# Voiced Controlled Music Player

BY

CHONG ZHENG KIT

Student ID: 18ACB04784

Supervised by

Ts Dr Ooi Chek Yee

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

# REPORT STATUS DECLARATION FORM

**Title:** **Voiced Controlled Music Player**

**Academic Session**: JAN 2022

I CHONG ZHENG KIT

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**:

56, Lorong Seri Setali 89,

Kuantan, Pahang

___________________________

Ooi Chek Yee

Supervisor's name

**Date**: 20 APRIL 2022

**Date**: 12 April 2022

| | | | |
|---|---|---|---|
| **Universiti Tunku Abdul Rahman** | | | |
| Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis** | | | |
| Form Number: **FM-IAD-004** | Rev No.: **0** | Effective Date: **21 JUNE 2011** | Page No.: **1 of 1** |

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date:20/4/2022

**SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS**

It is hereby certified that ***CHONG ZHENG KIT*** (ID No: ***18ACB04784*** ) has completed this final year project/ dissertation/ thesis* entitled "VOICE CONTROLLED MUSIC PLAYER" under the supervision of TS DR OOI CHEK YEE (Supervisor) from the Department of Computer and Communication Technology , Faculty of Computer and Communication Technology , and Dr Mogana a/p Vadiveloo (Co-Supervisor)* from the Department of Computer and Communication Technology , Faculty of Computer and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

______________________

(*CHONG ZHENG KIT*)

*Delete whichever not applicable

# DECLARATION OF ORIGINALITY

I declare that this report entitled "**Voiced Controlled Music Player**" 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 : Chong Zheng Kit

Date : 20/4/2022

# ACKNOWLEDGEMENTS

I would like to express my deepest appreciation and thanks to my supervisor, Dr Ooi Chek Yee who suggested me with this mobile application development title. Dr Ooi has given me a golden opportunity to involve in this voice control field which is currently trending technology. He also given me a lot of guidance to complete this project.

Other than that, there are also friends and family who helped me during this project development. Although they all have their own study or business to take care off, they are still willing to help me and support me when I face difficulties.

Once again, thanks for all the people who helped me during this project.

# Abstract

In this project, an android music player application will be developed. This music player's specialty is that the users can voice control the music player. However, users can use the music player as a normal music player too. This music player can scan the phone storage for the list of the songs and are consist of basic function that a music player possessed. Users can control the music player through the voice command button or using the interface of the music player. By speaking the voice command to the phone, the application will carry out its function based on the command such as play and pause. Users will be able to give out the voice command even if they are in the interface of the other application since the voice command button will stay on the screen if the user did not terminate the application. The interface of this application is very simple as it does not require time for the users to understand how the application works.

In this project the waterfall methodology is being implemented. There are 4 basic phases of this methodology, which is the Analysis Phase, Design Phase, Implementation Phase and lastly Testing phase. Through these 4 phases, the project can develop in the smoother pace.

## Table of Contents

## List of tables

## List of Figures

List of Figures

# Chapter 1 - Introduction

## 1.0 Problem Statement and Motivation

Problem statement:

- Users need to buy another device for voice command function to control the music player.
    - Majority of the users do not want to pay in order to enjoy the function.
- Users may need to suffer from the advertisement that keeps on pop out during the usage of the application.
    - Majority of the users feels annoy while advertisement suddenly pop out during the enjoyment of the music.
- Users may encounter application with messy interface.
    - Messy interface may lead to annoyance and cause dissatisfaction to the users.
- Users may need to pay to join the application community as a member in order to enjoy the full function of the application.
    - Some function is not unlocked if users did not pay the membership fees.

Humans are born with a body structure that allows us to carry out multitask. However, we also have our own limitations. We cannot surpass our human limitations, and this is where technologies play a major role. Although those problems can be solved with the help of others, there are also scenario where we cannot use that solution. For example, while we are cooking alone at home, music is playing with our phone, our hands only left 1 finger which is not wet. If we want to control the music player efficiently, we need to wipe our hand to dry which is quite troublesome since it will get wet again during cooking session. This is the time where voice command music player comes in handy by just having the users to use their one finger and speak out the command to control the music player.

The problem where human cannot surpass their limitation is a serious problem for us as a human being to accomplished the task more efficiently. Music is one of the necessary aspect for us to increase our work efficiency however we tend to ignore it since our hands are full with the works and are unable to control the music player. Based on research from Teresa Lesuik from Canada, she founds that music can show bring positive impact and quality of work were very low if the music were absence in the working environment [1].

## 1.1 Background Information

In this world of globalization, various technologies have been part of our life. These technologies have played an important role in our daily usage to make our life much convenient and comfortable. Among all those technologies, mobile technologies such as smartphone benefits in society communications. Nowadays, smartphone has become an inseparable gadget to almost all human on Earth. If we walk down on the street, we can observe the people around us are looking at the screen while they are walking along the path without paying much attention to their surroundings. A smartphone is a device that comes with multiple functions and available to multitask at the same moment. Of course, we all know that the main function of a smartphone is communicate with others. It is very convenient since there are many applications available in the online market that are able to carry out this function with free of charge or having a certain price such as the WhatsApp, WeChat or even Facebook Messenger. However, others function such as capturing photos, music player and gaming are available in this light and thin smartphone device. There are many benefits and limitations regarding the smartphones, but it's all depended on the consumers itself [2].

Wireless headset, earphones or Air pods are seen to be paired with their respective smart devices to allow the users to enjoy entertainment on demand. The entertainment could be music or even a video and these entertainments are all being powered by the 'Player' app. Moving back to the years of 1877, phonograph which is the first music player instrument that are able to record and play music are being created by Thomas Edison [3]. After that, music player continues to evolve as the times goes on such as CD players in 1982, MP3 players in 1998 and finally until nowadays which is the music streaming or wireless listening with the help of Bluetooth technology [3]. There are a lot of video or music players on the play store nowadays and each of them have their own design or specialties. They also have their own advantages or weaknesses while using it. By having a good music player app in our smartphone will be beneficial to all of the music lover in the world since music is one of an important aspect in their daily life.

In every smartphone that exist in the world nowadays are mainly function by having a strong operating system as their supports. Up until 2020, there are only 2 main operating system that exist which is the Android from Google and IOS from Apple. In the year of 2007, these two current leading companies took their first step in introducing their operating system to the world. Apple has launched their first iPhone in earlier 2007 while Google launched public beta of Android in November 2007 [4]. The main difference between the Android and IOS will be Android is an open-source operating system. By having an open-source operating systems means that anyone can modify the phone operating system to make the phone function according to the user's personal preferences. It breaks the tradition of all the others operating system of closed operating system. In this project, I will be developing the music player based on Google Android.

### 1.2 Objectives

- To list out all the song list that exist in the device.

  After the proposed application have scan the device for songs, the application will list out all the songs to let users to view. Song title and artist's name will be displayed.

- To play music that exist is the song list

  This is the main function of this application. Any song being scanned and include in the list will be able to play.

- To allow users to have music playing in the background

Users will be able to play music with this application even if the screen is off or using other applications.

- To allow users to voice command the application

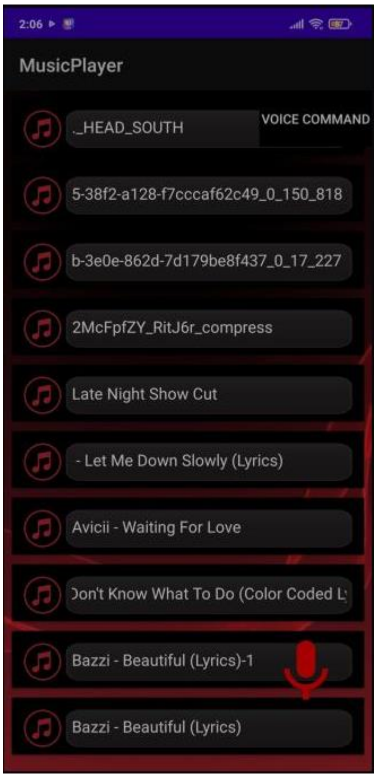
  Users will be able to voice command this music player by simply tap on the mic mutton. The music player will be able to function based on the command given.

- To allow users to voice command the application in background.

  Users will be still available to voice command even though the application is running on the background. The mic button will remain on the screen although the users have switch to other applications.

## 1.3 Highlight of what have been achieved

The application being developed is named as Voice Controlled Music Player. This application will be able to play mp3 file that being found in the phone storage. This application was being developed by using Android Studio.


2:06
MusicPlayer
_HEAD_SOUTH
VOICE COMMAND
5-38f2-a128-f7cccaf62c49_0_150_818
b-3e0e-862d-7d179be8f437_0_17_227
2McFpfZY_RitJ6r_compress
Late Night Show Cut
- Let Me Down Slowly (Lyrics)
Avicii - Waiting For Love
Don't Know What To Do (Color Coded L
Bazzi - Beautiful (Lyrics)-1
Bazzi - Beautiful (Lyrics)

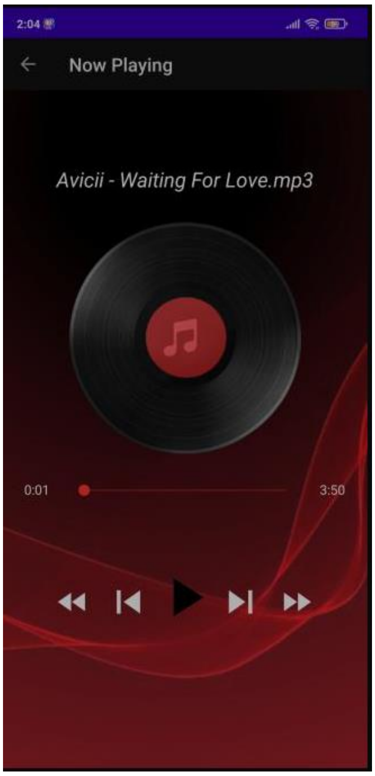
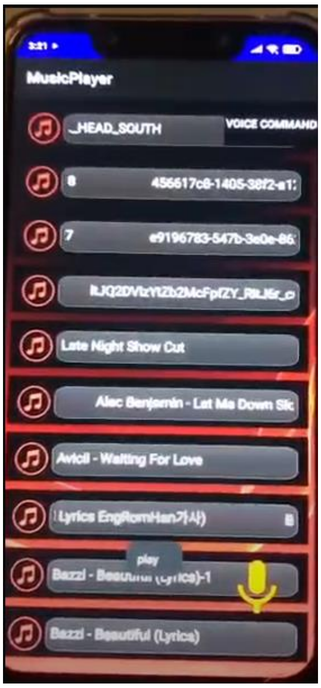


Figure 1.3.1 Main Page

Figure 1.3.2 Player



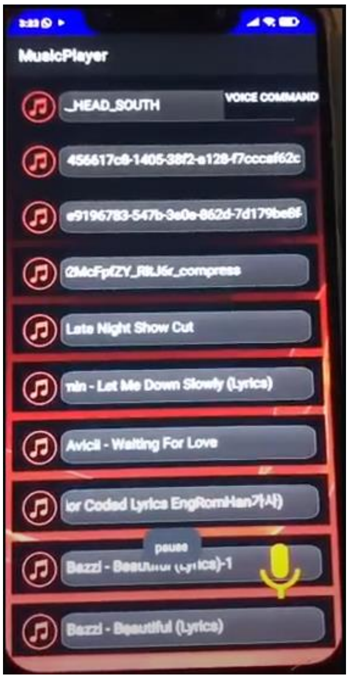

MusicPlayer
_HEAD_SOUTH
VOICE COMMAND
456617c8-1405-38f2-a128-f7cccaf62c
e9196783-547b-3e0e-862d-7d179be8f
2McFpfZY_RitJ6r_compress
Late Night Show Cut
min - Let Me Down Slowly (Lyrics)
Avicii - Waiting For Love
lor Coded Lyrics EngRomHan가사)
pause
Bazzi - Beautiful (Lyrics)-1
Bazzi - Beautiful (Lyrics)

Figure 1.3.3 VC Play

Figure 1.3.4 VC Pause

Figure 1.3.1 is showing the main page of this application. When user click on the voice command button, it will trigger the on and off the voice command button that exist at the bottom right corner. When user click on any song title that exist in the main page list, it will direct the user to the music player page which is the Figure1.3.2, Player Activity page. In Player Activity page, user will be able to control the music player like any others music player.

Figure 1.3.3 and Figure 1.3.4 is showing the voice command function. After user tap on the voice command button and speak the command, the application will recognize the voice command and carry out its function. For example, in Figure 1.3.3, “play” is being speak to the application and the application will recognize the command and carry out its function. This process will be the same for Figure 1.3.4.

# Chapter 2 – Literature Review

## 2.1 Technologies Involved

The problem that being reviewed in the introduction is that we as a human cannot surpass our own limits and wanted to make our life easier. As we all know, the main solution is the technologies that being developed as fast as lightning nowadays. However, technologies also being separated into many smaller domains such as the construction domain, working domain, entertainment domain and much more. In this project, I will be setting my foot on the entertainment domain which is developing a functional and useful music player. The main function I'll be including in this music player will be the voice command function. In the market right now, there are quite a number of technologies that are using the voice recognition and command. For examples will be the Siri from Apple, Cortana from Microsoft or Alexa from Amazon [5]. I will be separating this literature review into 2 parts. Information about the voice recognition or command will be discussed in the first part while similar application will be reviewed in the second part. All of the information I found from the internet or the application I have reviewed give me lots of benefits in my proposed project. However, nothing is perfect, some limitations and weaknesses also could be found from them.

Figure 2.1.1 and figure 2.1.2 showing Alexa and Siri which is current trend of voice command devices.

Figure 2.1.1 - Alexa

Figure 2.1.2 - Siri

### 2.1.1 Voice Recognition and Command

Let us talk about some history of the voice recognition. At the early stage of this technologies which is around 1950s to 1960s, words are not the main focused in this voice recognition but numbers only. However, there are 16 words are able to being recognize by the 'SHOEBOX' which introduced by the IBM. From 1960s to 1990s, this technology continues to make advancement little by little through increasing of words being recognized by the system. Few methods have been introduced in this period of time such as the 'Hidden Markov Model (HMM)' which are able to predict unknown sounds actually being word or voice portal (VAL) system which a voice recognition system that form a lots of phone tree systems nowadays. There was a huge growth happening in the 2000s because Google has launched Google Voice Search. Although voice recognition technology has reached close to 80% accuracy by the year of 2001, there were no functional applications that are able to apply this technology to the fullest until Google make their huge step. Many people can get their hands on this Google Voice Search function as long as they have google service in their devices. At the same time, Google has included 230 billion words from the user's searches into their voice search system. At the early part of 2010s, we can saw huge explosion of voice recognition applications. Apple being launched Siri in 2011, Amazon's Alexa and Google home being introduced to the market around that time too. We can see the users are becoming more and more comfortable communicating with the devices using their speech. [6]

The world now is full will a lot of smartphones, smart cars and smart appliances such as the Google home or Alexa. All of this devices or applications are able to perform smart function and one of them will be the Voice Recognition function. This technology is very hard to complete and needs a lot of time to processes it before it is ready to use. For example, imagine how a child learns a language. To make this function workable, a mic, recognition software and data model about the voice are essential. All of the voice recognition hardware or software works almost the same logic [6].

Figure 2.1.3 showing the cycle of voice input in a device.

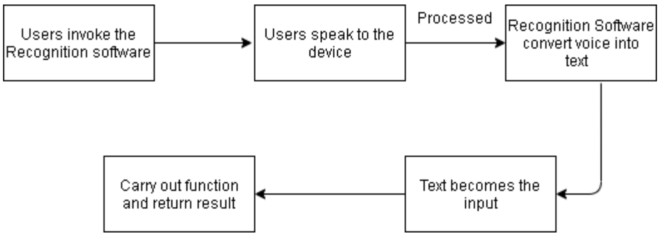

Users invoke the Recognition software
Users speak to the device
Processed
Recognition Software convert voice into text
Text becomes the input
Carry out function and return result

**Figure 2.1.3 Voice Input Cycle**

## 2.2 Existing software review

### 2.2.1 Youtube

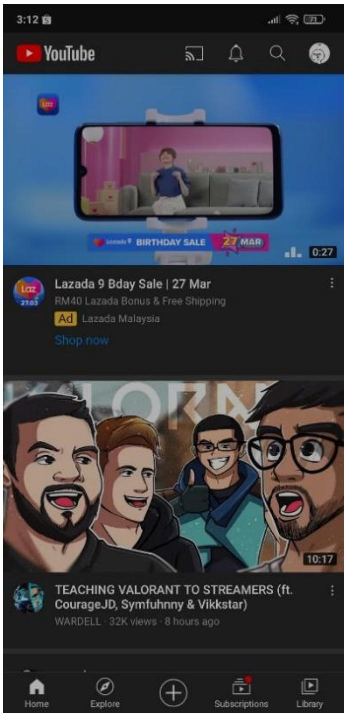
Without any doubt, Youtube is one of the biggest and strongest music streaming apps that exist in the world right now. This application support in any webpage, operating system and almost any technologies device in the world. In this report I will be reviewing only on the smartphone's Youtube. [7] When we first enter the application, we will be starting off at its homepage which consist of the logo, stream at other device button, search button and profile button at the top. Middle will be video suggestion and advertisement and bottom will be having the home page button, explore button, upload video button, user's subscription button and lastly library button.

The stream on other device button work just like the name which is streaming on other smart devices that are having the stream function which include smart TV and certain monitor. The notification button will allow user to view if any of the subscription channel have upload any new videos. When users click on the search button, a list of search history will appear and user can input keywords to search by using keyboard or voice input. Last button on the top will be the account button which lead users to the management account and setting page.

At the bottom side will be the home button first which is showing the homepage since the application has been started. At the home page, recommended video by the application and advertisement will be showed. Next will be the explore button which leads the user to select which categories of video that the user wanted to watch such as trending, gaming, music, news or sport. The upload button will allow the users to upload video or having a live stream session. The subscriptions button will show videos of user’s subscriptions only. Lastly will be the library button which show user’s history, playlist, downloads and some minor functions.

Not everything is perfect. Youtube is a commercial application which will include advertisement in the video and the homepage. Users have to watch the advertisement before or during the video playing. Users also need to pay for Youtube Premium to fully access its feature such as the playing video at the background.

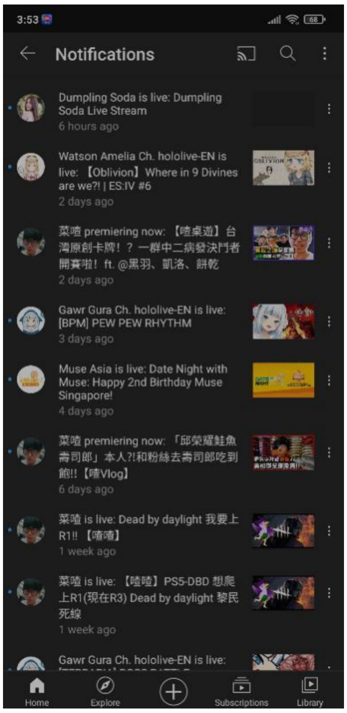
Figure 2.2.1.1 to Figure 2.2.1.4 is showing some screenshot of Youtube Application.



Lazada 9 Bday Sale | 27 Mar
RM40 Lazada Bonus & Free Shipping
Lazada Malaysia

TEACHING VALORANT TO STREAMERS (ft. CourageJD, Symfuhnny & Vikkstar)
WARDELL · 32K views · 8 hours ago

**Figure 2.2.1.1 Youtube Home Page**



Dumpling Soda is live: Dumpling Soda Live Stream
6 hours ago

Watson Amelia Ch. hololive-EN is live: 【Oblivion】 Where in 9 Divines are we?! | ES:IV #6
2 days ago

菜喳 premiering now: 【喳桌遊】台灣原創卡牌！？一群中二病發決鬥者開賽啦！ ft. @黑羽、凱洛、餅乾
2 days ago

Gawr Gura Ch. hololive-EN is live: [BPM] PEW PEW RHYTHM
3 days ago

Muse Asia is live: Date Night with Muse: Happy 2nd Birthday Muse Singapore!
4 days ago

菜喳 premiering now: 「邱榮耀鮭魚壽司郎」本人?!和粉絲去壽司郎吃到飽!!【喳Vlog】
6 days ago

菜喳 is live: Dead by daylight 我要上R1!! 【喳喳】
1 week ago

菜喳 is live: 【喳喳】PS5-DBD 想爬上R1(現在R3) Dead by daylight 黎民死線
1 week ago

Gawr Gura Ch. hololive-EN is live:

**Figure 2.2.1.2 Youtube Notification**

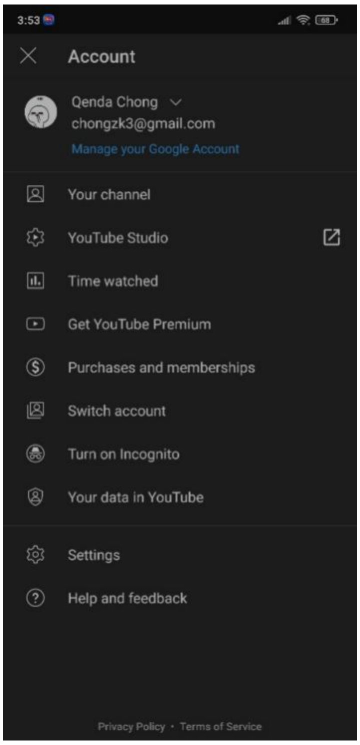
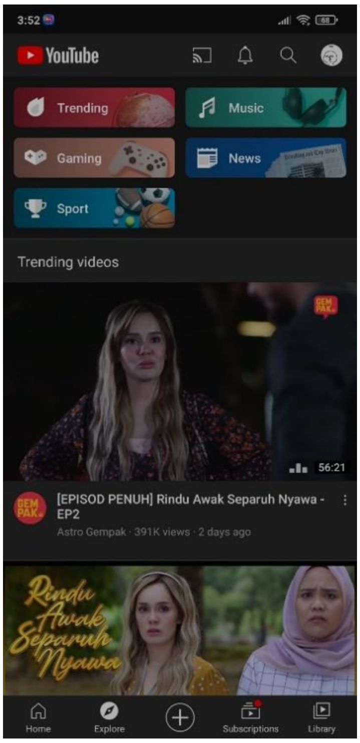


**Figure 2.2.1.3 Youtube Profile**



[EPISOD PENUH] Rindu Awak Separuh Nyawa - EP2
Astro Gempak · 391K views · 2 days ago

**Figure 2.2.1.4 Youtube Explore**

### 2.2.2 Spotify

‘Listening is everything’ this is the slogan of the Spotify and we all know that this application is a well-known music player and streaming application that is being widely used by the people in this continent. This application can be installed on Windows, IOS, Android and many more operating systems. As we are developing an android music player, I will be only reviewing on the Spotify app that are able to install on smart phones.

When we enter the Spotify app, we will be directed to the homepage. In the home page, there will be a lot of playlist being generated based on the history of users. There will be also playlist like Top 50 in Malaysia by obtaining the GPS information from the user’s device.

After that will be the browse page. There will be a lot of categories of songs provided by the app. Users can select any categories from the list to listen to the playlist that being auto generated by the app.

Next up will be the search page. In this page, users are able to search for the songs and playlist that is being generated by other users or the app itself. User s are also able to scan the QR code of the playlist being generated by the app to ease the searching process. The recent search will also being list out in this page.

Furthermore, will be the radio page. In this page, the app will provide few station for the users to join and play the songs in respective station. Last but not least will be the user’s library. In this library will be including user’s playlist, stations, songs, albums, artists, podcast, videos and recently played songs.

Although this application is being widely used by the people, there are also some limitations and weaknesses. This app is 100% available for free and use. However, users cannot utilize all the functions unless the users have become a premium user which require users to pay for it. Without premium account, user can only shuffle play the music in the playlist, having few minutes of advertisement per hour, audio quality being limited and unable to download the songs.

Figure 2.2.2.1 to Figure 2.2.2.5 is showing some screenshot of SpotifyApplication.

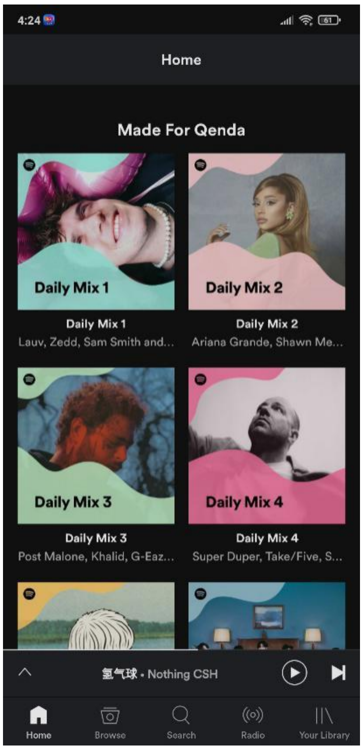


Figure 2.2.2.1 Spotify Home

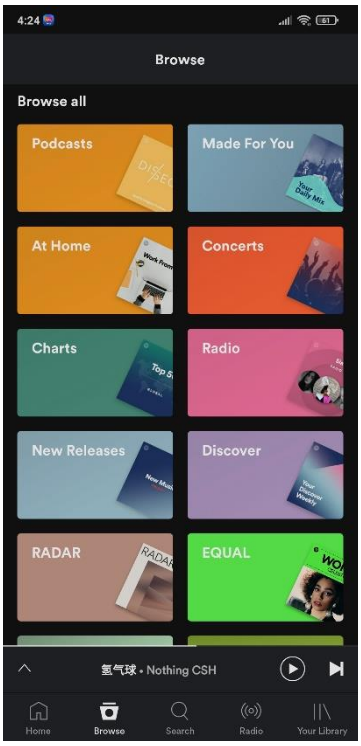


Figure 2.2.2.2 Spotify Browse

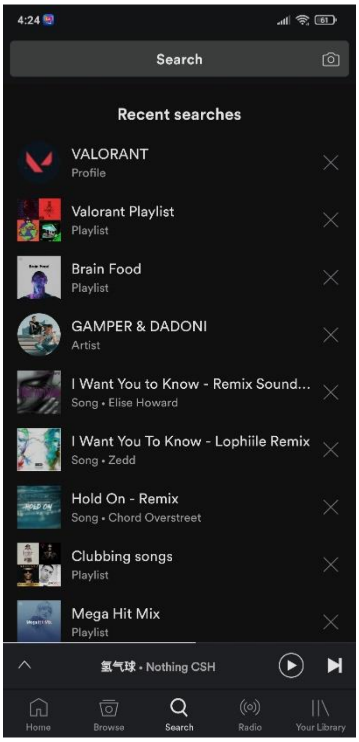


Figure 2.2.2.3 Spotify Search

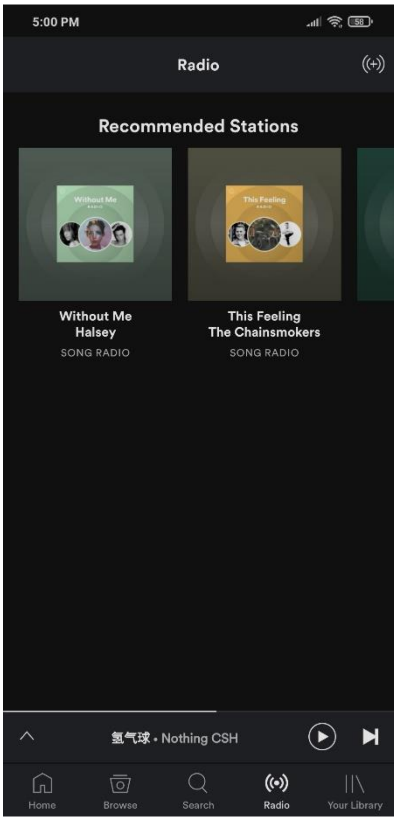


Figure 2.2.2.4 Spotify Radio

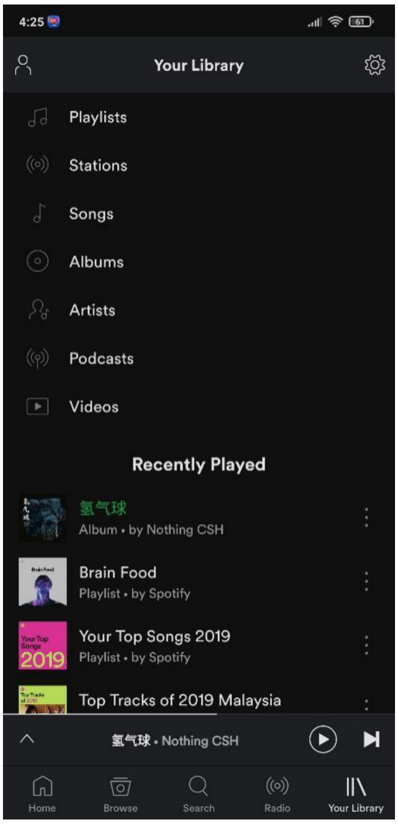


Figure 2.2.2.5 Spotify Library

## 2.3 Comparison between similar applications

Table 2.1 showing the comparison between the reviewed applications with the proposed application.

**Table 2.1: Comparison between similar applications**

| **Application / Criteria** | **Youtube** | **Spotify** | **Proposed Application** |
|---|---|---|---|
| **User Interface** | Average | Average | Simple |
| **Scan songs** | No | No | Yes |
| **Pop Up button** | No | No | Yes |
| **Background Play** | No | Yes | Yes |
| **Advertisement** | Yes | Yes | No |
| **Voice Command** | No | No | Yes |

# Chapter 3 – Proposed Method/ Approach

## 3.1 Design Specifications

### 3.1.1 Methodology

This project is being is being conducted under the methodology which is called the waterfall methodology. This methodology consists of many phases which is called initiation phase, analysis phase, design phase, development phase, testing phase and lastly deployment phase. This methodology is a sequential app development life cycle such that we cannot move on to the next phase unless the previous phase is completed [8].

Figure 4.1 showing the waterfall methodology model which found from the internet.

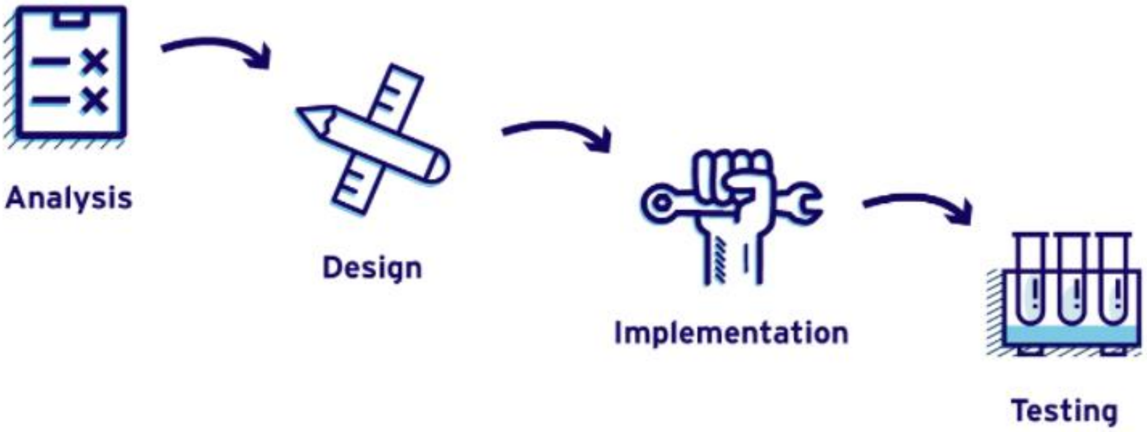


Figure 3.1 Waterfall Methodology Model

**Requirement Analysis Phase**

In this phase, the requirement of this project or app are being defined and analyse thoroughly.

There is no building of the application involve. This is a research phase to let the developer know the problem and its solutions [9].

**Design Phase**

This is the phase where developer design and develop the application to solve the problem mention in the analysis phase according to the requirements and data model. This phase will be producing a prototype to be used in later phases [9].

**Implementation**

This is the phase where the prototype is being installed. Usually this will be shortest phase since the preparation already done in the previous phases [9].

**Testing**

After the implementing phase where the prototype being installed, testing of this application will be conducted to check whether it will be work as intended [9].

### 3.1.2 Tools used

To develop this application, the tools or compiler I will be using is the Android Studio. It is an Integrated Development Environment (IDE) which designed specifically for android application development. This compiler was being developed by Google and Jetbrains which formed on Jetbrains' IntelliJ IDEA and it is written in Java, Kotlin and C++. Android studio is being supported in Windows, MacOS and linux based operating system and available for free in the official website [10].

Other than that, Visual Paradigm is also used to design diagrams such as use case diagram and flow chart. This application is available online or software that is downloadable.

Hardware:

1. Laptop
    - Processor: Intel(R) Core (TM) i7-7700HQ
    - RAM: 8GB
    - Graphic Card: NVIDIA GeForce GTX 1050 ti
    - Hard Disk Storage: 120GB SSD
    - Operating System: Window 10
2. Smartphone device
    - Chipset: Qualcomm SDM845 Snapdragon 845
    - RAM: 6GB
    - Phone Storage: 128 GB
    - Operating System: Android 10

### 3.1.3 User Requirements

Functional Requirements:

- As a user, I able to view the song list.
- As a user, I able to select from the song list.
- As a user, I able to play the song.
- As a user, I able to voice command the music player.

Non-Functional Requirements:

- As a user, I able to know functions of the application.
- As a user, I able to know the functions of the buttons.

### 3.1.4 System Performance Definition

1. Response Time

Any application developed need a fast response for the users to use it as comfortable as possible. It is better for the response time between 0.1-1 second and any time more than that may leads to user's happiness decrease.

2. Resource Usage

   It is better to have a low memory consumption application which will carry out the same function as other application. If the application consumes more resources, it may cause the application or the phone to run slower and drain more battery power.

## 3.2 System Design/ Overview

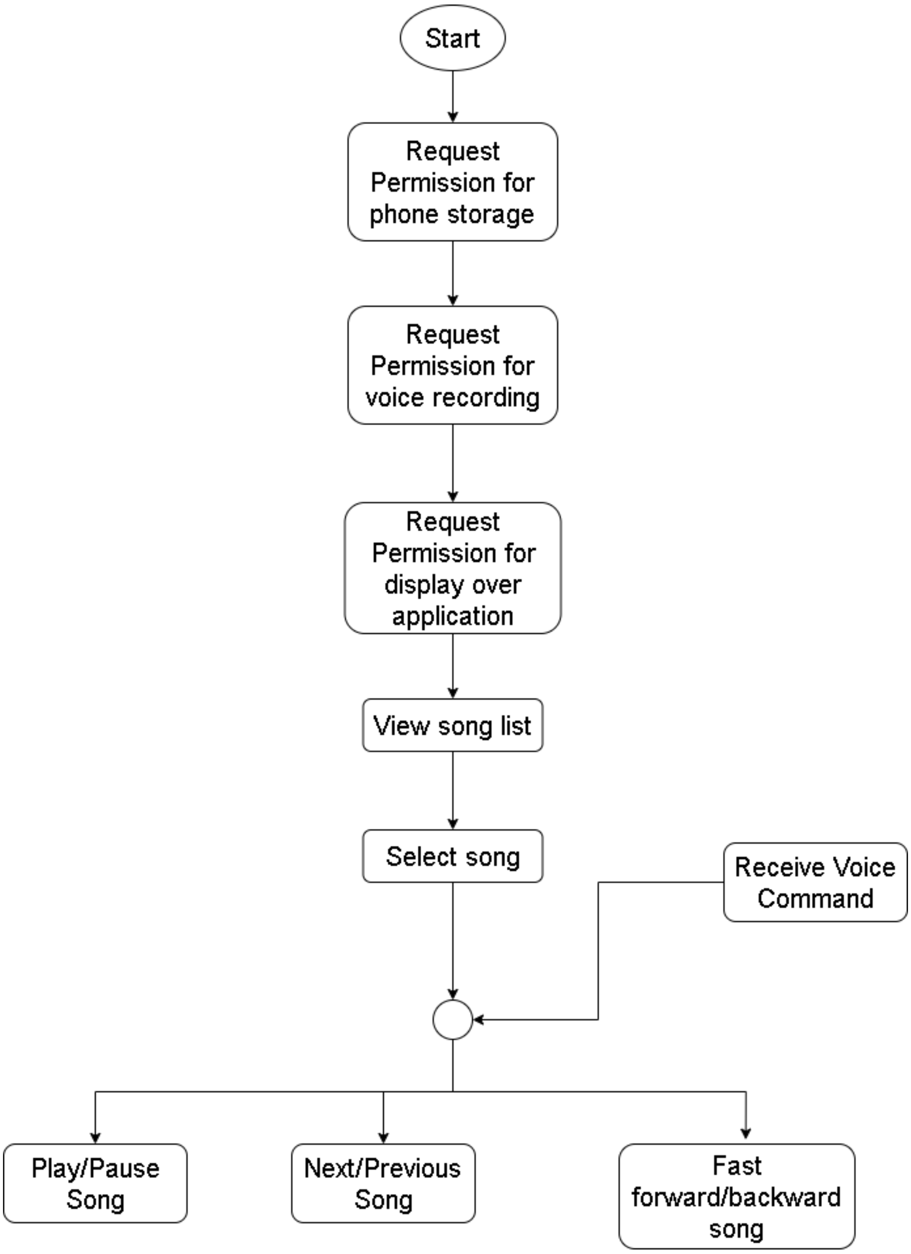

Start
Request Permission for phone storage
Request Permission for voice recording
Request Permission for display over application
View song list
Select song
Receive Voice Command
Play/Pause Song
Next/Previous Song
Fast forward/backward song

***Figure 3.2.1 - System flow-chart***

When the application started, it will prompt the permission for the user to scan phone's storage for the song file. After that the song is being listed out in the application. By selecting the song, user will be able to play the song which leads to song playing interface. In the song playing interface, users will be able to carry out function such as play and pause the song, play next or previous song, and lastly fast-forward or backward the song which is playing through the button in the interface. User s will be able to carry out above function by using voice command.

Use case diagram

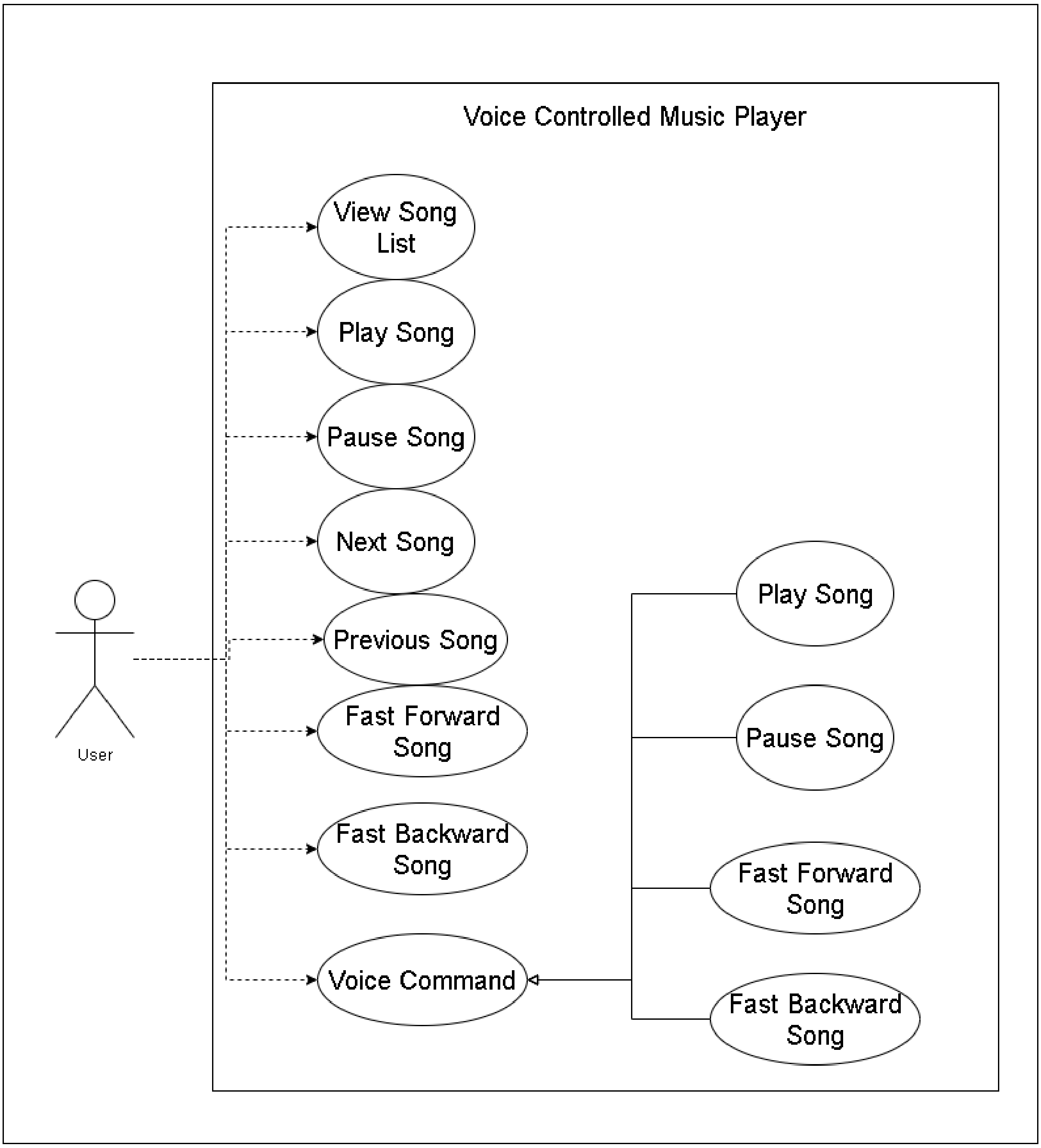


***Figure 3.2.2 Use case diagram***

Use Case Description

| | |
|---|---|
| Use Case Name: View Song List | ID: UC001 |
| Actor: User | |
| Description: Allow user to view the song list. | |
| Trigger: User start the application. | |
| Precondition: User store song in phone. | |
| Normal Flow of events:<br>1. User view the song list.<br>2. User scroll up and down. | |
| Alternate flow of events:<br>1a. No song available -> nothing shows in the list. | |

Table 3.2.1 Use case description of View Song List.

| | |
|---|---|
| Use Case Name: Play Song | ID: UC002 |
| Actor: User | |
| Description: Allow user to play the song list. | |
| Trigger: User select the song | |
| Precondition:<br>1a. Song is listed.<br>1b. Music player is paused. | |
| Normal Flow of events:<br>1a. User select the song.<br>1b. User click on the play button. | |

| |
|---|
| 2. Music player play the song. |
| Alternate flow of events:<br>1a. File corrupted.<br>1b. Unable to play the song. |

Table 3.2.2 Use case description of Play Song.

| | |
|---|---|
| Use Case Name: Pause Song | ID: UC003 |
| Actor: User | |
| Description: Allow user to pause the song. | |
| Trigger: User wish to pause the song | |
| Precondition: Song is playing. | |
| Normal Flow of events:<br>1. User click on the pause button.<br>2. Music player pause the song. | |
| Alternate flow of events:<br>1a. Song ended. | |

Table 3.2.3 Use case description of Pause Song.

| | |
|---|---|
| Use Case Name: Play Next Song | ID: UC004 |
| Actor: User | |
| Description: Allow user to play next the song. | |
| Trigger: User wish to play next song. | |
| Precondition: Music player is running. | |
| Normal Flow of events:<br>1. User click on the next button.<br>2. Music player play the next song. | |
| Alternate flow of events: - | |

Table 3.2.4 Use case description of Next Song.

| | |
|---|---|
| Use Case Name: Play Previous Song | ID: UC005 |
| Actor: User | |
| Description: Allow user to play previous the song. | |
| Trigger: User wish to play previous song. | |
| Precondition: Music player is running. | |
| Normal Flow of events:<br>1. User click on the previous button.<br>2. Music player play the previous song. | |
| Alternate flow of events: - | |

Table 3.2.5 Use case description of Previous Song.

| | |
|---|---|
| Use Case Name: Fast Forward Song | ID: UC006 |
| Actor: User | |
| Description: Allow user to fast forward the song. | |
| Trigger: User wish to fast forward the song. | |
| Precondition: Music player is playing. | |
| Normal Flow of events:<br>1. User click on the fast forward button.<br>2. Music player fast forward for 10 seconds. | |
| Alternate flow of events: - | |

Table 3.2.6 Use case description of Fast Forward Song.

| | |
|---|---|
| Use Case Name: Fast Rewind Song | ID: UC007 |
| Actor: User | |
| Description: Allow user to fast Rewind the song. | |
| Trigger: User wish to fast rewind the song. | |
| Precondition: Music player is playing. | |
| Normal Flow of events:<br>1. User click on the fast rewind button.<br>2. Music player fast rewind for 10 seconds. | |
| Alternate flow of events: - | |

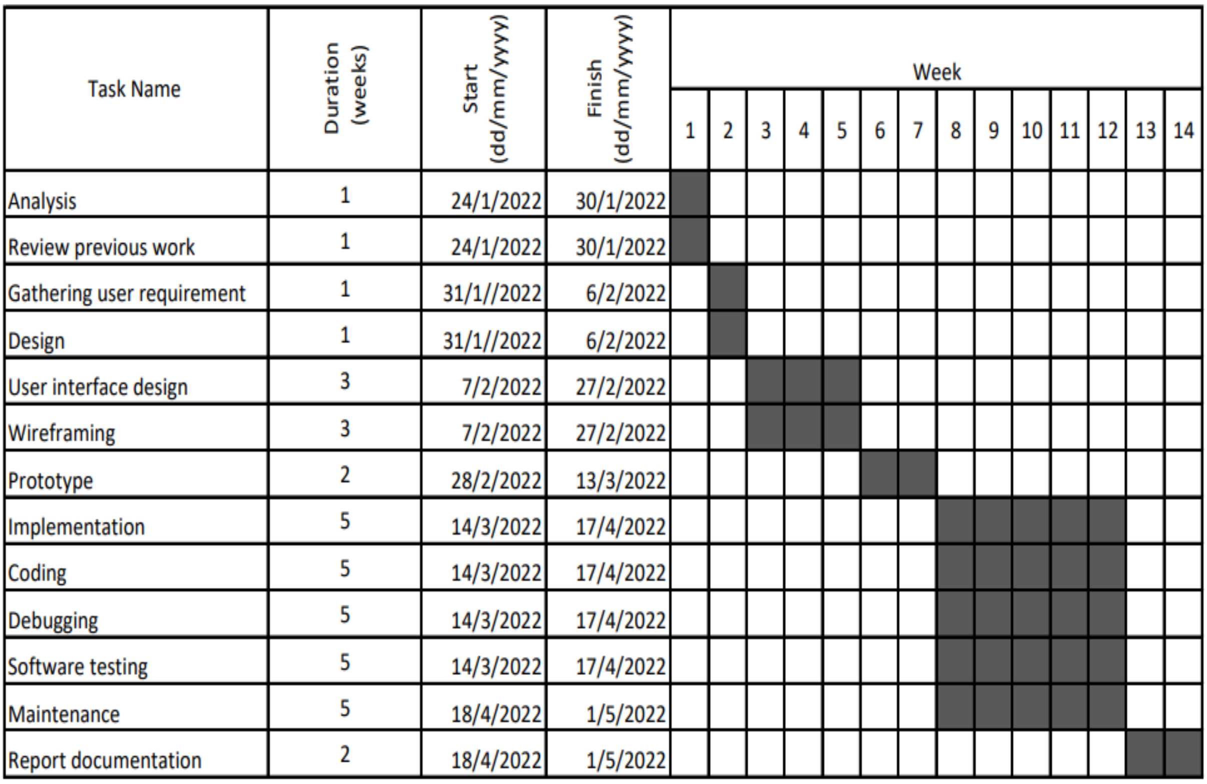
Table 3.2.7 Use case description of Fast Rewind Song.

| | |
|---|---|
| Use Case Name: Voice Command | ID: UC008 |
| Actor: User | |
| Description: Allow user to voice command the player. | |
| Trigger: User wish to voice command the player. | |
| Precondition: Music player is playing. | |
| Normal Flow of events:<br>1. User click on the voice command button.<br>2. User give command.<br>3. Player carry out function. | |
| Alternate flow of events:<br>2a. Unrecognized command.<br>2b. Player did not carry out the function. | |

Table 3.2.8 Use case description of Voice Command.

## 3.3 Analysis

### 3.3.1 Timeline



| Task Name | Duration (weeks) | Start (dd/mm/yyyy) | Finish (dd/mm/yyyy) | Week | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Analysis | 1 | 24/1/2022 | 30/1/2022 | ■ | | | | | | | | | | | | | |
| Review previous work | 1 | 24/1/2022 | 30/1/2022 | ■ | | | | | | | | | | | | | |
| Gathering user requirement | 1 | 31/1//2022 | 6/2/2022 | | ■ | | | | | | | | | | | | |
| Design | 1 | 31/1//2022 | 6/2/2022 | | ■ | | | | | | | | | | | | |
| User interface design | 3 | 7/2/2022 | 27/2/2022 | | | ■ | ■ | ■ | | | | | | | | | |
| Wireframing | 3 | 7/2/2022 | 27/2/2022 | | | ■ | ■ | ■ | | | | | | | | | |
| Prototype | 2 | 28/2/2022 | 13/3/2022 | | | | | | ■ | ■ | | | | | | | |
| Implementation | 5 | 14/3/2022 | 17/4/2022 | | | | | | | | ■ | ■ | ■ | ■ | ■ | | |
| Coding | 5 | 14/3/2022 | 17/4/2022 | | | | | | | | ■ | ■ | ■ | ■ | ■ | | |
| Debugging | 5 | 14/3/2022 | 17/4/2022 | | | | | | | | ■ | ■ | ■ | ■ | ■ | | |
| Software testing | 5 | 14/3/2022 | 17/4/2022 | | | | | | | | ■ | ■ | ■ | ■ | ■ | | |
| Maintenance | 5 | 18/4/2022 | 1/5/2022 | | | | | | | | ■ | ■ | ■ | ■ | ■ | | |
| Report documentation | 2 | 18/4/2022 | 1/5/2022 | | | | | | | | | | | | | ■ | ■ |

## 3.4 Design

### 3.4.1 System Flow Chart

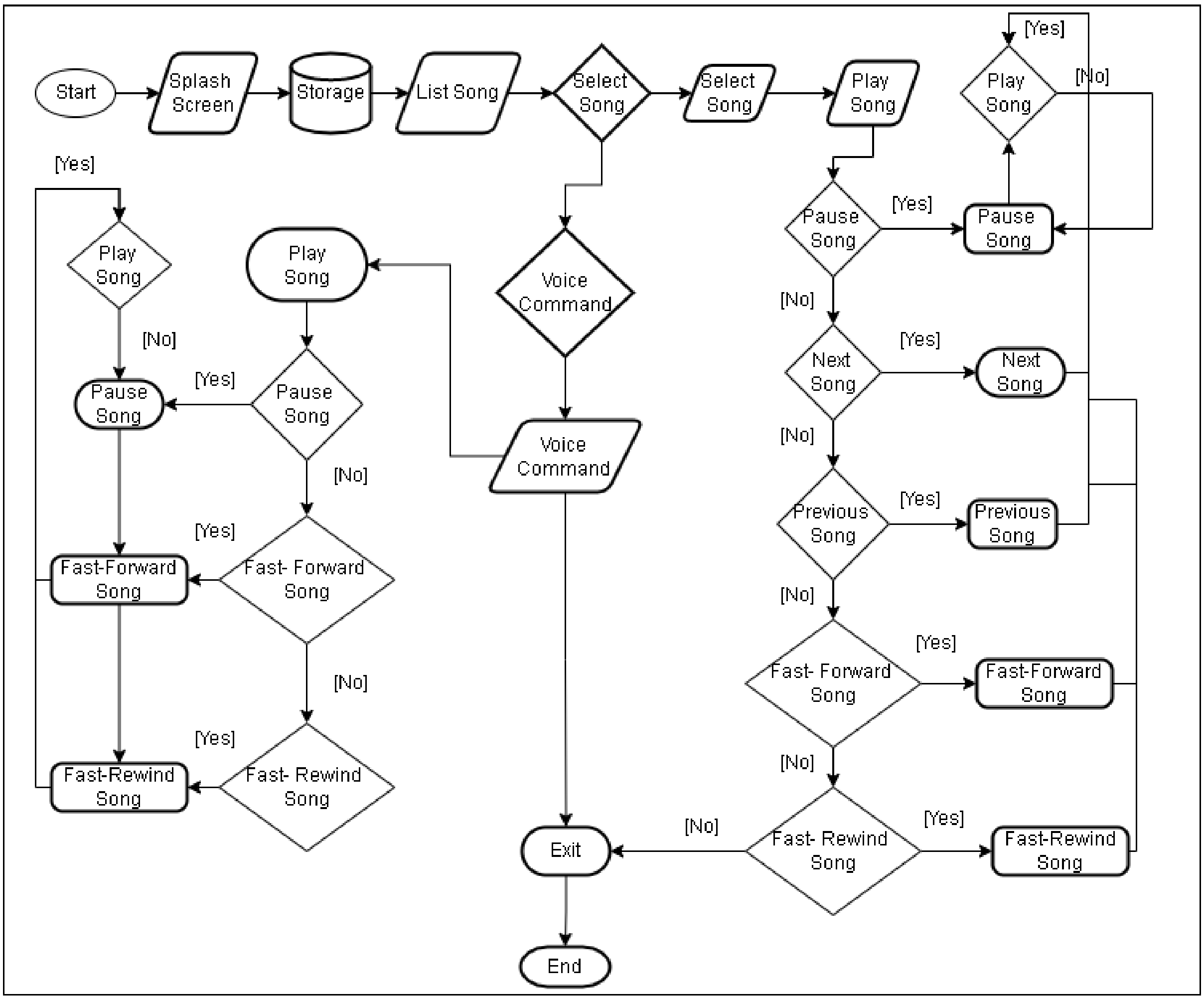


Figure 3.4.1 System Flow Chart

| **Process** | **Functionality** |
|---|---|
| Start | Start Application |
| Splash Screen | Display Splash Screen (Logo and instruction). |
| Storage | Search storage for mp3 file. |
| List Song | List all mp3 file found. |
| Select Song (decision) | Check decision for select song. |
| Voice Command (decision) | Check decision for voice command. |
| Select Song | Select song. |
| Play Song | Play selected song. |
| Pause Song (decision) | Check decision for pause song. |
| Next Song (decision) | Check decision for next song. |
| Previous Song (decision) | Check decision for previous song. |
| Play Song (decision) | Check decision for play song. |
| Pause Song | Pause the song. |
| Next Song | Play next song. |
| Previous Song | Play previous song. |
| Voice Command | Voice command the player. |
| Exit | Exit application. |
| End | End application |

Table 3.4.1: Explanation of flow chart functionality

### 3.4.2 Application User Interface Design

**Splash Screen**

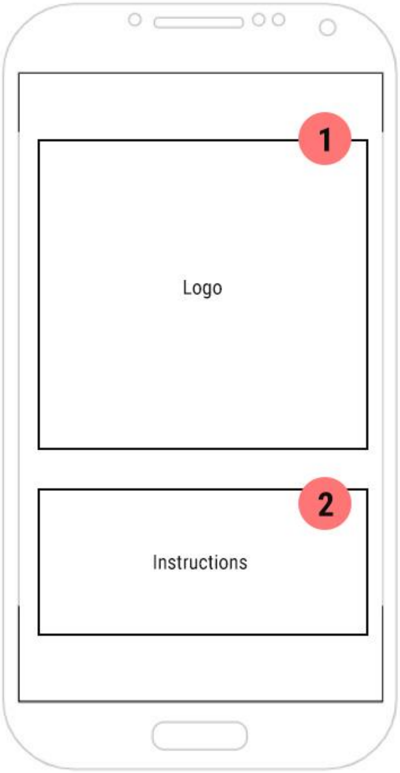


Figure 3.4.2.1 Splash Screen

1. Application Logo

2. Voice Command Instruction

**Main Page**

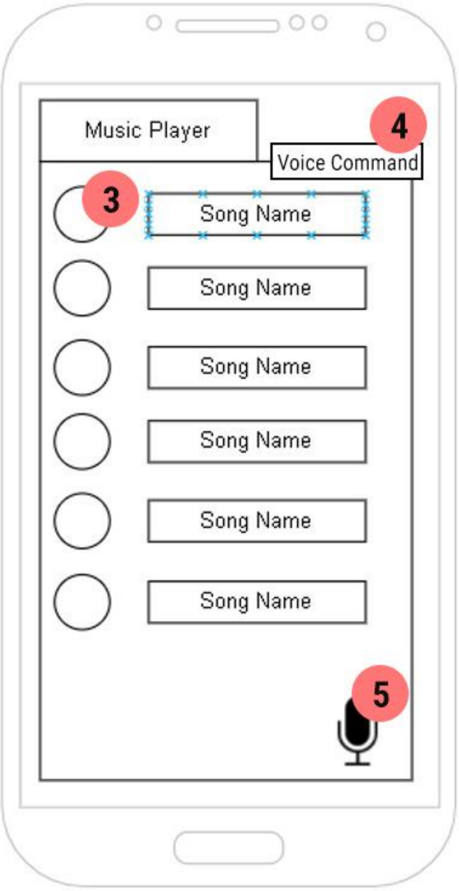


Figure 3.4.2.2 Main page

3. Song List

4. Voice command on/off button

5. Voice command mic button

**Player Page**

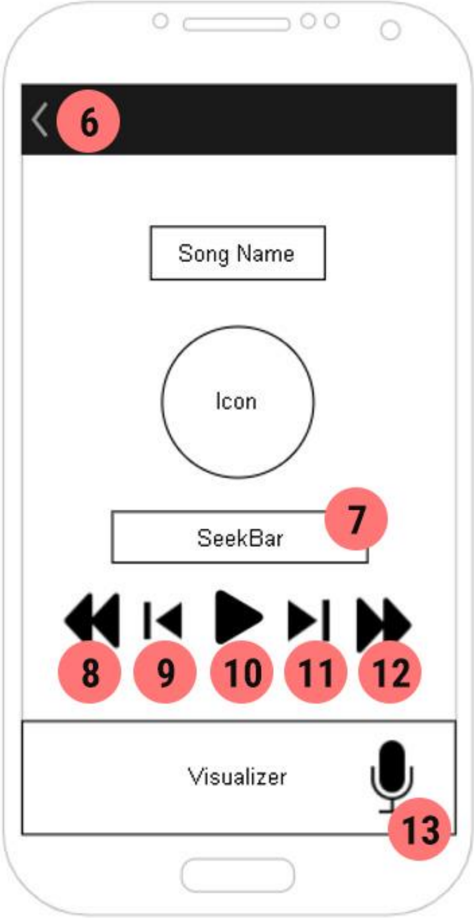


Figure 3.4.2.3 Player Page

6. Back button

7. Seek Bar

8. Fast Rewind button

9. Previous button

10. Play/Pause button

11. Next button

12. Fast Forward button

13. Mic button

## 3.5 WireFrame

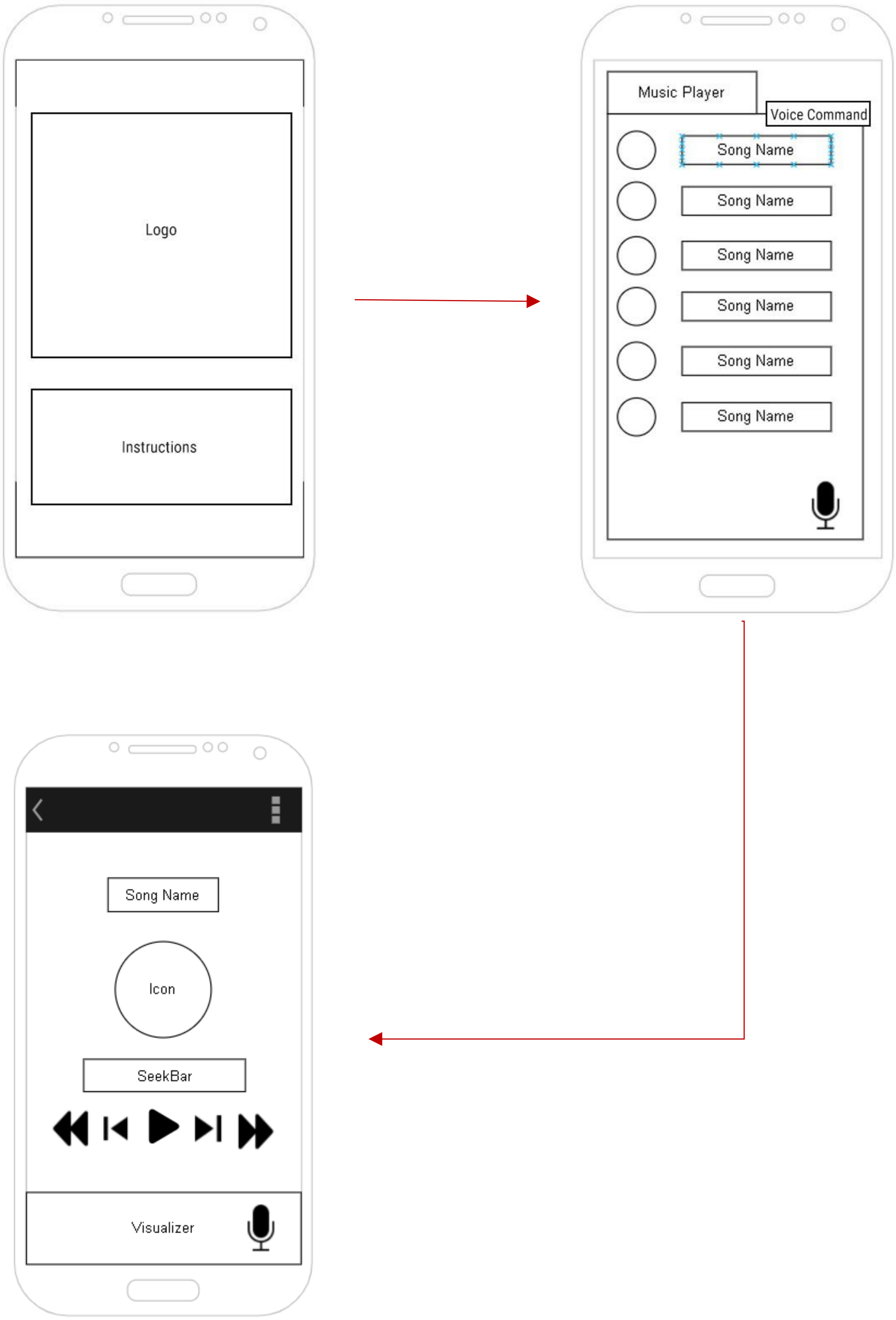


Figure 3.5.1 Wireframe of Proposed Application

## 3.6 System Component Design

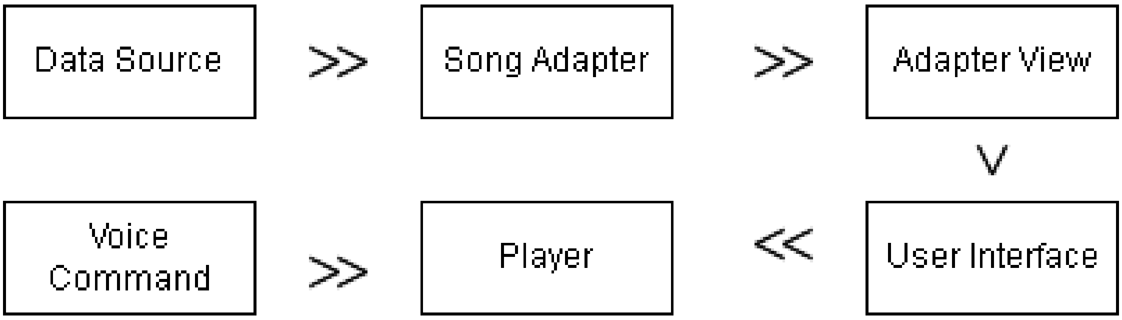


Figure 3.6.1 System Flow

Data source representing the data which is the mp3 file that exist in the database. Song adapter will hold the data and transfer it to adapter view. Adapter view will display the data in the user interface. User will be able to play the song and the player can control the audio play. Voice command is given by the user to control the music player.

## 3.7 Verification Plan

| Test Case Number | Test Case Name | Type of Input | Test Case Description | Expected Result |
|---|---|---|---|---|
| 1 | Mobile Application | Application | Check whether application can install and run-in actual device. | Mobile application can be installed and run. |
| 2 | UI Design | Source Code | Check whether the design is same as the emulator. | UI is same as the emulator. |
| 3 | VC button | Source Code | Check whether the on and off the voice command button is working | VC button is working by showing mic button when on and vice versa when off. |
| 4 | List Song | Source Code | Check whether the application can list the song exist in the device. | Application can read the storage and display the mp3 file exist in the device. |
| 5 | Select Song | Source Code | Check whether user can select the song to play form the list. | Player will play the selected song. |
| 6 | Play Song | Source Code | Check whether the user can play the song. | Player will play the selected song. |

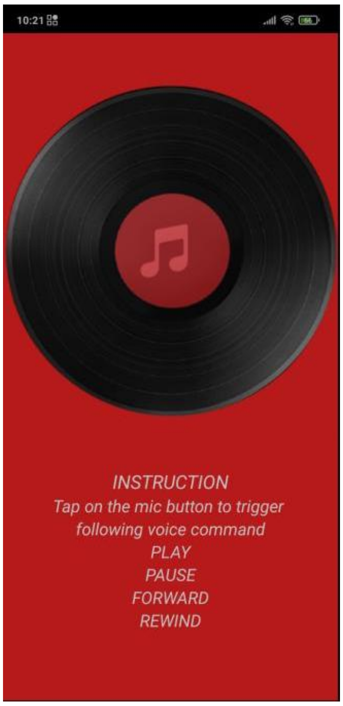
| | | | | |
|---|---|---|---|---|
| 7 | Pause Song | Source Code | Check whether the user can pause the song. | Player will pause the song. |
| 8 | Next Song | Source Code | Check whether the user can play the next song. | Player will play the next song. |
| 9 | Previous Song | Source Code | Check whether the user can play the previous song. | Player will play the previous song. |
| 10 | Fast Forward Song | Source Code | Check whether the user can fast forward the previous song. | Player will fast forward the song by 10 second. |
| 11 | Fast Rewind Song | Source Code | Check whether the user can fast rewind the previous song. | Player will fast rewind the song by 10 second. |
| 12 | Seek Bar | Source Code | Check whether the seek bar is same progress as the player. | The seek bar show same progress as the music playing. |
| 13 | Voice command | Source Code | Check whether the application can record the audio and carry out the function. | Audio can be recorded and carry out its respective function. |

| | | | | |
|---|---|---|---|---|
| 14 | Background Play | Source Code | Check whether the application can play song in the background. | Player can play songs in the background. |
| 15 | Background Voice command | Source Code | Check whether the application can carry out voice command in the background. | Application can receive voice command in the background. |

Table 3.7 Verification Plan

# Chapter 4: System Application Development

## 4.1 Result after done building



INSTRUCTION
Tap on the mic button to trigger following voice command
PLAY
PAUSE
FORWARD
REWIND

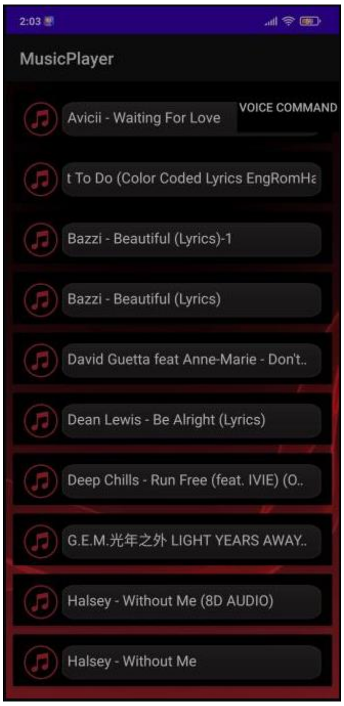
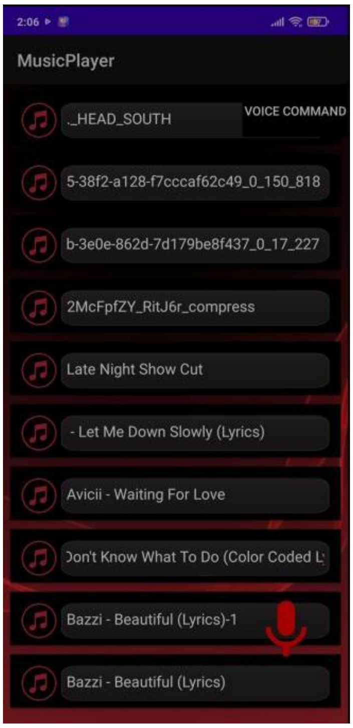

2:03
MusicPlayer
VOICE COMMAND
Avicii - Waiting For Love
t To Do (Color Coded Lyrics EngRomHa
Bazzi - Beautiful (Lyrics)-1
Bazzi - Beautiful (Lyrics)
David Guetta feat Anne-Marie - Don't..
Dean Lewis - Be Alright (Lyrics)
Deep Chills - Run Free (feat. IVIE) (O..
G.E.M.光年之外 LIGHT YEARS AWAY..
Halsey - Without Me (8D AUDIO)
Halsey - Without Me

Figure 4.1.1 Splash Screen

Figure 4.1.2 Main Page (Without VC button)


2:06
MusicPlayer
VOICE COMMAND
_HEAD_SOUTH
5-38f2-a128-f7cccaf62c49_0_150_818
b-3e0e-862d-7d179be8f437_0_17_227
2McFpfZY_RitJ6r_compress
Late Night Show Cut
- Let Me Down Slowly (Lyrics)
Avicii - Waiting For Love
Don't Know What To Do (Color Coded L
Bazzi - Beautiful (Lyrics)-1
Bazzi - Beautiful (Lyrics)

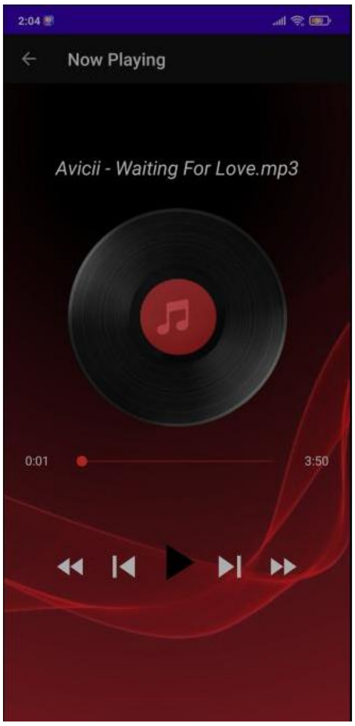
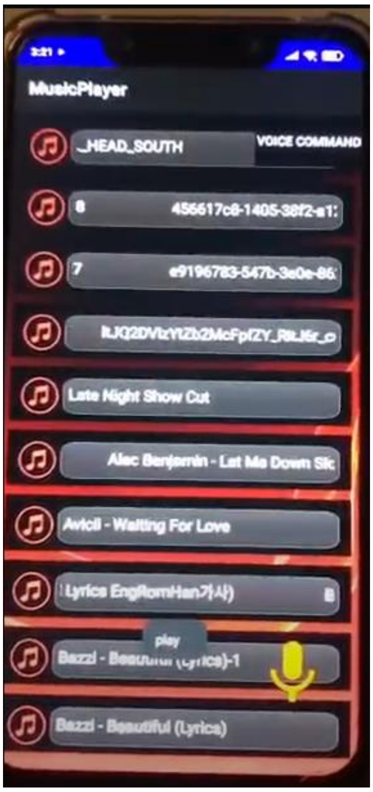


Figure 4.1.3 Main Page (With VC button)

Figure 4.1.4 Music Player



MusicPlayer

_HEAD_SOUTH

VOICE COMMAND

8 456617c8-1405-38f2-a1:

7 e9196783-547b-3e0e-86

ltJQ2DVtzYtZb2McFpfZY_RltJ6r_c

Late Night Show Cut

Alec Benjamin - Let Me Down Slc

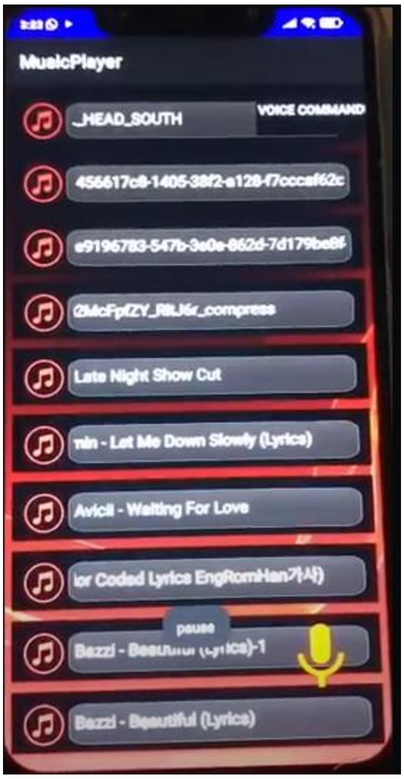
Avicii - Waiting For Love

Lyrics EngRomHan가사) 8

play

Bazzi - Beautiful (Lyrics)-1

Bazzi - Beautiful (Lyrics)



MusicPlayer

_HEAD_SOUTH

VOICE COMMAND

456617c8-1405-38f2-a128-f7cccaf62c

e9196783-547b-3e0e-862d-7d179be8f

2McFpfZY_RltJ6r_compress

Late Night Show Cut

min - Let Me Down Slowly (Lyrics)

Avicii - Waiting For Love

ior Coded Lyrics EngRomHan가사)

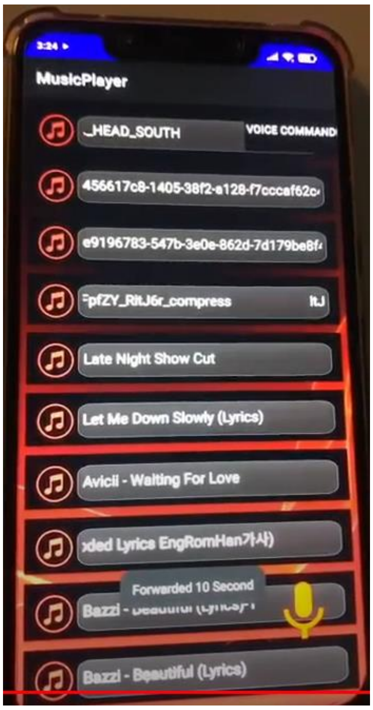
pause

Bazzi - Beautiful (Lyrics)-1

Bazzi - Beautiful (Lyrics)

Figure 4.1.5 Voice command(play)

Figure 4.1.6 Voice command(pause)


3:24
MusicPlayer
_HEAD_SOUTH
VOICE COMMAND
456617c8-1405-38f2-a128-f7cccaf62c
e9196783-547b-3e0e-862d-7d179be8f
pfZY_RItJ6r_compress
ItJ
Late Night Show Cut
Let Me Down Slowly (Lyrics)
Avicii - Waiting For Love
oded Lyrics EngRomHan가사)
Forwarded 10 Second
Bazzi - Beautiful (Lyrics)
Bazzi - Beautiful (Lyrics)

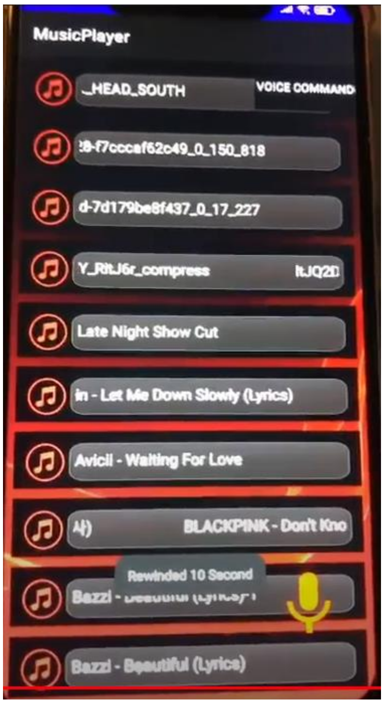


Figure 4.1.7 Voice command(forward)    Figure 4.1.8 Voice
command(rewind)

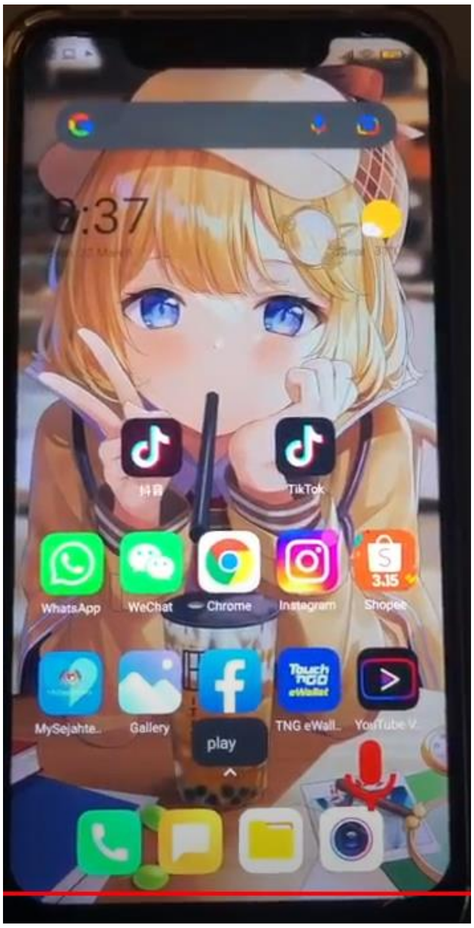


Figure 4.1.9 Voice command background (play)

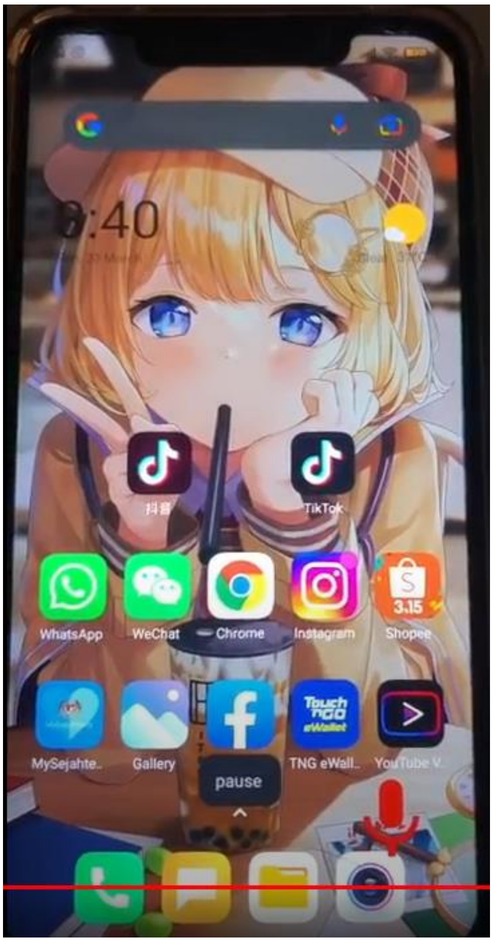


Figure 4.1.10 Voice command background(pause)

Figure 4.1.1 shows the splash screen before the application is fully loaded. In this splash screen contain logo and the instruction for using the voice command. After the splash screen will be the main page which is Figure 4.1.2, the main page. In the main page contain the song list which found in the local storage of the phone itself. After turning on the voice command it will appear a mic button at the bottom left corner of the screen which is just like Figure 4.1.3. After user select the song from the song list, it will move to music player interface which is Figure 4.1.4. In this interface user will be able to control the music player by clicking the respective button such as the play/pause button, next/previous song button and fast forward and backward button. Voice command include play/pause and fast forward and backward function which shown in Figure 4.1.5 to Figure 4.1.8. The voice command can also be used in the background, and it is shown in Figure 4.1.9 to Figure 4.1.10.

# Chapter 5 – Conclusion

## 5.1 Project Review

The limitations of a normal music player which user cannot command it by voice are solved through this project. The aim purpose of the design is to let all-age user feel comfortable using it, thus it is design as simple as possible with function intact. This project can help the user feel ease while listening to music.

## 5.2 Future Work

There are a lot of improvement can be made for this project. First, this project can implement cloud service. User can view and download the song from it and play it. Furthermore, playlist function can be included in it. User can add songs they want into the playlist and straight play from it rather than playing each song. Besides that, AI machine learning also can be included, so that the application can self-learn and recommend song at what location and what time.

## Reference

[1] Farris. B, “Does listening to music at work improve productivity? here's 8 compelling reasons why it does,” *Anchor Advisors*, 14-Jan-2019. [Online]. Available: https://anchoradvisors.com/listening-music-improve-productivity. [Accessed: 31-Mar-2022].

[2] R. I. Admin, “How smartphone changed our lives.,” *Social Diary Sunday Magazine*, 28-Jan-2019. [Online]. Available: http://socialdiary.pk/how-smartphone-changed-our-lives/#:~:text=Smartphones%20along%20with%20internet%20service,and%20they%20have%20an%20easy. [Accessed: 31-Mar-2022].

[3] “History of music machines: Evolution of music players,” *History of Music Machines | Evolution of Music Players*, 2021. [Online]. Available: https://www.volocars.com/blog/history-of-music-machines. [Accessed: 31-Mar-2022].

[4] Callaham J, “The history of Android: The evolution of the biggest Mobile OS in the world,” *Android Authority*, 28-Dec-2021. [Online]. Available: https://www.androidauthority.com/history-android-os-name-789433/#:~:text=The%20founding%20of%20Android,Chris%20White%2C%20and%20Andy%20Rubin. [Accessed: 31-Mar-2022].

[5] K. Vyas, “Everything you need to know about voice recognition technology - dzone AI,” *dzone.com*, 06-Mar-2018. [Online]. Available:

https://dzone.com/articles/everything-you-need-to-know-about-voice-recognitio. [Accessed: 31-Mar-2022].

[6] Sonix, “A brief history of speech recognition,” *Sonix*, 2021. [Online]. Available: https://sonix.ai/history-of-speech-recognition#:~:text=The%20first%20speech%20recognition%20systems,to%2016%20words%20in%20English. [Accessed: 31-Mar-2022].

[7] R. Bennett, “Review: YouTube app for iPhone and Android,” *[2021],* 09-Mar-2022. [Online]. Available: https://filmora.wondershare.com/youtube-video-editing/youtube-app-for-iphone-android.html. [Accessed: 31-Mar-2022].

[8] “Waterfall vs. Agile Methodology,” *Agile Introduction For Dummies,* 04-Jan-2008. [Online]. Available: https://agileintro.wordpress.com/2008/01/04/waterfall-vs-agile-methodology/. [Accessed: 31-Mar-2022].

[9] “MaRS Startup Toolkit. 2021. Product development: Waterfall Model | Software development methodology | Entrepreneur’s Toolkit,” *Product development: Waterfall Model | Software development methodology*. [Online]. Available: https://learn.marsdd.com/article/product-development-the-waterfall-methodology-model-in-software-development. [Accessed: 31-Mar-2022].

[10] “Meet android studio : Android developers,” *Android Developers*. [Online]. Available: https://developer.android.com/studio/intro. [Accessed: 31-Mar-2022].

# Appendices

## Appendix A: Source Code

AndroidManifest.xml

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


    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"></uses-
permission>
    <uses-permission
android:name="android.permission.SYSTEM_ALERT_WINDOW"></uses-permission>


    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:requestLegacyExternalStorage="true"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:largeHeap="true"
        android:hardwareAccelerated="false"
        android:theme="@style/Theme.MusicPlayer">
        <activity android:name=".PlayerActivity"></activity>
        <activity android:name=".MainActivity"></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>
        <service android:name=".WidgetService" android:enabled="true"></service>
    </application>


</manifest>
```

MainActivity.java

```
package com.example.musicplayer;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.provider.Settings;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.text.Layout;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.karumi.dexter.Dexter;
import com.karumi.dexter.MultiplePermissionsReport;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionDeniedResponse;
import com.karumi.dexter.listener.PermissionGrantedResponse;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.multi.MultiplePermissionsListener;
import com.karumi.dexter.listener.single.PermissionListener;


import java.io.File;
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    ListView listView;
    String[] items;
    Button buttonAddWidget;
    int OnandOff;


    WidgetService widgetService = new WidgetService();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        listView = findViewById(R.id.listViewSong);





        OnOff= 0;


        buttonAddWidget = (Button) findViewById (R.id.button_widget);
        getpermission();
        buttonAddWidget.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                if (!Settings.canDrawOverlays(MainActivity.this))
                {
                    getpermission();
                }
                else
                {
                    Intent intent = new Intent(MainActivity.this, WidgetService.class);

                    if(OnOff==0) {
                        startService(intent);
                        OnOff = 1;
                    }
                    else
                    {
                        if (OnOff==1)
                        {
                            stopService(intent);
                            widgetService.onDestroy();
                            OnOff=0;
                        }
                    }

                }

            }
        });




        runTimePermission();

    }



    public void getpermission() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M
&& !Settings.canDrawOverlays(this)) {
            if (!Settings.canDrawOverlays(MainActivity.this)) {
                Intent intent = new Intent
                        (Settings.ACTION_MANAGE_OVERLAY_PERMISSION,
Uri.parse("package:" +
                                getPackageName()));
                startActivityForResult(intent, 1);
            }

        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode==1)
            if(!Settings.canDrawOverlays(MainActivity.this))
            {
                Toast.makeText(this, "Permission denied by
user.",Toast.LENGTH_SHORT).show();
            }

    }

    //access storage
    public  void runTimePermission (){


Dexter.withContext(this).withPermissions(Manifest.permission.READ_EXTERNAL_STOR
AGE, Manifest.permission.RECORD_AUDIO)
                .withListener(new MultiplePermissionsListener() {
                    @Override
                    public void onPermissionsChecked(MultiplePermissionsReport
multiplePermissionsReport) {
```

```
                    displaySongs();
                }

                @Override
                public void onPermissionRationaleShouldBeShown(List<PermissionRequest>
list, PermissionToken permissionToken) {
                    permissionToken.continuePermissionRequest();
                }
            }).check();

    }

    //find song from external storage
    public ArrayList<File> findSong (File file)
    {
        ArrayList<File> arrayList = new ArrayList<>();
        File [] files = file.listFiles();

        for (File singlefile: files )
        {
            if (singlefile.isDirectory() && !singlefile.isHidden()) // check directory and whether is
hidden
            {
                arrayList.addAll(findSong(singlefile));
                Collections.sort(arrayList);
            }
            else
            {
                if (singlefile.getName().endsWith(".mp3") ||singlefile.getName().endsWith(".wav"))
                {
                    arrayList.add(singlefile);

                }
            }
        }

        return arrayList;
    }

    void displaySongs(){
        final ArrayList<File> mySongs =
findSong(Environment.getExternalStorageDirectory());
        items = new String[mySongs.size()];

        for (int i = 0 ; i < mySongs.size(); i ++ )
        {
            items[i] = mySongs.get(i).getName().toString().replace(".mp3",
"").replace(".wav","");
        }
        /*
        ArrayAdapter<String> myAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, items);
```

```
    listView.setAdapter(myAdapter);
    */


    customAdapter customAdapter = new customAdapter();
    listView.setAdapter(customAdapter);


    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int i, long id) {
            String songName = (String) listView.getItemAtPosition(i);
            startActivity(new Intent(getApplicationContext(),PlayerActivity.class)
                    .putExtra("songs", mySongs)
                    .putExtra("songname",songName)
                    .putExtra("pos",i));
        }
    });


}

class customAdapter extends BaseAdapter
{

    @Override
    public int getCount() {
        return items.length;
    }

    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
    public View getView(int i, View convertView, ViewGroup parent) {
        View myView = getLayoutInflater().inflate(R.layout.list_item, null);
        TextView textsong = myView.findViewById(R.id.txtsongname);
        textsong.setSelected(true);
        textsong.setText(items[i]);

        return myView;
    }
}


private static int OnOff;
```

```
    public void setOnOff(int newV){
        OnOff = newV;
    }
}
```

PlayerActivity.java

```
package com.example.musicplayer;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.animation.AnimatorSet;
import android.animation.ObjectAnimator;
import android.content.Intent;
import android.graphics.PorterDuff;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;
import com.gauravk.audiovisualizer.visualizer.BarVisualizer;
import java.io.File;
import java.util.ArrayList;

public class PlayerActivity extends AppCompatActivity {
    Button btnplay, btnnext,btnprev,btnff,btnfr;
    TextView txtsname, txtsstart,txtsstop;
    SeekBar seekmusic;
    BarVisualizer visualizer;
    ImageView imageView;

    String sname;
    public static final String EXTRA_NAME = "song_name";
    static MediaPlayer mediaPlayer;
    int position=0;
    ArrayList<File> mySongs;
    Thread updateseekbar;

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == android.R.id.home)
        {
            onBackPressed();
        }
        return super.onOptionsItemSelected(item);
    }
```

```
@Override
protected void onDestroy() {
    if(visualizer != null)
    {
        visualizer.release();
    }
    super.onDestroy();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_player);

    getSupportActionBar().setTitle("Now Playing");
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setDisplayShowHomeEnabled(true);

    btnprev = findViewById(R.id.btnprev);
    btnnext = findViewById(R.id.btnnext);
    btnplay = findViewById(R.id.playbtn);
    btnff = findViewById(R.id.btnff);
    btnfr = findViewById(R.id.btnfr);
    txtsname = findViewById(R.id.txtsn);
    txtsstart = findViewById(R.id.txtsstart);
    txtsstop = findViewById(R.id.txtsstop);
    seekmusic = findViewById(R.id.seekbar);
    visualizer = findViewById(R.id.blast);
    imageView = findViewById(R.id.imageview);


    if (mediaPlayer != null)
    {
        mediaPlayer.stop();
        mediaPlayer.release();
    }
    Intent i = getIntent();
    Bundle bundle = i.getExtras();

    mySongs = (ArrayList)bundle.getParcelableArrayList("songs");
    String songName = i.getStringExtra("songname");
    position = bundle.getInt("pos", 0);
    txtsname.setSelected(true);
    Uri uri = Uri.parse(mySongs.get(position).toString());
    sname = mySongs.get(position).getName();
    txtsname.setText(sname);

    mediaPlayer = MediaPlayer.create(getApplicationContext(),uri);
    mediaPlayer.start();


    updateseekbar = new Thread()
    {
      @Override
      public void run() {
        int totalDuration = mediaPlayer.getDuration();
        int currentposition = 0;

        while(currentposition < totalDuration)
        {
          try {
            sleep(500);
            currentposition = mediaPlayer.getCurrentPosition();
            seekmusic.setProgress(currentposition);
          }
          catch (InterruptedException | IllegalStateException e)
          {
            e.printStackTrace();
          }
        }
      }
    };


    seekmusic.setMax(mediaPlayer.getDuration());
    updateseekbar.start();

seekmusic.getProgressDrawable().setColorFilter(getResources().getColor(R.color.colorPrima
ry), PorterDuff.Mode.MULTIPLY);
    seekmusic.getThumb().setColorFilter(getResources().getColor(R.color.colorPrimary),
PorterDuff.Mode.SRC_IN);

    seekmusic.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
      @Override
      public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {

      }

      @Override
      public void onStartTrackingTouch(SeekBar seekBar) {

      }

      @Override
      public void onStopTrackingTouch(SeekBar seekBar) {
        mediaPlayer.seekTo(seekBar.getProgress());

      }
    });

    String endTime = createTime(mediaPlayer.getDuration());
    txtsstop.setText(endTime);

    final Handler handler = new Handler();
    final int delay = 1000;
```

```
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        String currentTime = createTime(mediaPlayer.getCurrentPosition());
        txtsstart.setText(currentTime);
        handler.postDelayed(this, delay);

    }
},delay);


btnplay.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (mediaPlayer.isPlaying())
        {
            btnplay.setBackgroundResource(R.drawable.ic_play);
            mediaPlayer.pause();
        }
        else
        {
            btnplay.setBackgroundResource(R.drawable.ic_pause);
            mediaPlayer.start();
        }
    }
});
//next listener

mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
    @Override
    public void onCompletion(MediaPlayer mp) {
        btnnext.performClick();
    }
});

int audiosessionId = mediaPlayer.getAudioSessionId();
if (audiosessionId != -1 )
{
    visualizer.setAudioSessionId(audiosessionId);
}


btnnext.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        playnext();
    }
});

btnprev.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        playprev();
```

```
            }
        });

        btnff.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ff();
            }
        });
        btnfr.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                fr();
            }
        });


    }
    public void playnext()
    {
        mediaPlayer.stop();
        mediaPlayer.release();
        position = ((position +1%mySongs.size()));
        Uri u = Uri.parse(mySongs.get(position).toString());
        mediaPlayer = MediaPlayer.create(getApplicationContext(), u);
        sname = mySongs.get(position).getName();
        txtsname.setText(sname);
        mediaPlayer.start();
        btnplay.setBackgroundResource(R.drawable.ic_pause);
        startAnimation(imageView);

        int audiosessionId = mediaPlayer.getAudioSessionId();
        if (audiosessionId != -1 )
        {
            visualizer.setAudioSessionId(audiosessionId);
        }

    }

    public void playprev()
    {
        mediaPlayer.stop();
        mediaPlayer.release();
        position = ((position-1)< 0 )?(mySongs.size()-1):(position -1 );

        Uri u = Uri.parse(mySongs.get(position).toString());
        mediaPlayer= MediaPlayer.create(getApplicationContext(),u);
        sname = mySongs.get(position).getName();
        txtsname.setText(sname);
        mediaPlayer.start();
        btnplay.setBackgroundResource(R.drawable.ic_pause);
        startAnimation(imageView);
        int audiosessionId = mediaPlayer.getAudioSessionId();
        if (audiosessionId != -1 )
        {
            visualizer.setAudioSessionId(audiosessionId);
        }
    }

    public void ff()
    {
        if (mediaPlayer.isPlaying())
        {
            mediaPlayer.seekTo(mediaPlayer.getCurrentPosition()+ 10000);// ff 10second
        }
    }

    public void fr()
    {
        if (mediaPlayer.isPlaying())
        {
            mediaPlayer.seekTo(mediaPlayer.getCurrentPosition() - 10000);// ff 10second
        }
    }


    public void startAnimation(View view)
    {
        ObjectAnimator animator = ObjectAnimator.ofFloat(imageView, "rotation", 0f, 360f);
        animator.setDuration(1000);
        AnimatorSet animatorSet = new AnimatorSet();
        animatorSet.playTogether(animator);
        animator.start();
    }

    public String createTime(int duration)
    {
        String time = "";
        int min = duration/1000/60;
        int sec = duration/1000%60;
        time+= min+ ":";
        if (sec<10)
        {
            time+= "0";
        }
        time +=sec;

        return time;
    }
}
```

SplashActivity.java

```
package com.example.musicplayer;
```

```
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;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class SplashActivity extends AppCompatActivity {
    private ImageView logo;
    private static int splashScreenTimeOut = 5000;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splashscreen);
        logo = findViewById(R.id.logo);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(SplashActivity.this, MainActivity.class);
                startActivity(intent);
                finish();
            }
        }, splashScreenTimeOut);

    }
}
```

WidgetService.java

```
package com.example.musicplayer;

import android.app.Service;
import android.content.Intent;
import android.graphics.PixelFormat;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import android.speech.RecognitionListener;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;

import androidx.annotation.Nullable;

import java.io.File;
import java.net.URISyntaxException;
import java.util.ArrayList;
import java.util.Locale;


public class WidgetService extends Service {
    int LAYOUT_FLAG;
    View mFloatingView;
    WindowManager windowManager;
    ImageView imageClose;
    private SpeechRecognizer speechRecognizer;
    private Intent speechRecognizerIntent;
    private String keeper = "";
    PlayerActivity player = new PlayerActivity();

    Button micWidget;
    float height, width;

    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            LAYOUT_FLAG =
WindowManager.LayoutParams.TYPE_ACCESSIBILITY_OVERLAY;
            LAYOUT_FLAG =
WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY;
        } else {
            LAYOUT_FLAG = WindowManager.LayoutParams.TYPE_PHONE;
        }

        mFloatingView = LayoutInflater.from(this).inflate(R.layout.layout_widget, null);

        WindowManager.LayoutParams layoutParams = new
WindowManager.LayoutParams(WindowManager.LayoutParams.WRAP_CONTENT,
                WindowManager.LayoutParams.WRAP_CONTENT,
                LAYOUT_FLAG,
                WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
                PixelFormat.TRANSLUCENT);
        layoutParams.gravity = Gravity.BOTTOM | Gravity.RIGHT;
        layoutParams.x = 0;
        layoutParams.y = 100;

        //layout parama for close button
        WindowManager.LayoutParams imageParams = new
WindowManager.LayoutParams(140,
                140,
                LAYOUT_FLAG,
                WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
                PixelFormat.TRANSLUCENT);
        imageParams.gravity = Gravity.BOTTOM | Gravity.CENTER;
        imageParams.y = 100;

        windowManager = (WindowManager) getSystemService(WINDOW_SERVICE);
        imageClose = new ImageView(this);
        imageClose.setImageResource(R.drawable.close_white);
        imageClose.setVisibility(View.INVISIBLE);
        windowManager.addView(imageClose, imageParams);
        windowManager.addView(mFloatingView, layoutParams);
        mFloatingView.setVisibility(View.VISIBLE);

        height = windowManager.getDefaultDisplay().getHeight();
        width = windowManager.getDefaultDisplay().getWidth();

        micWidget = (Button) mFloatingView.findViewById(R.id.mic_widget);



        speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this);
        speechRecognizerIntent = new
                Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);


speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,Recogn
izerIntent.LANGUAGE_MODEL_FREE_FORM);
        speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
                Locale.getDefault());

        speechRecognizer.setRecognitionListener(new RecognitionListener() {
            @Override
            public void onReadyForSpeech(Bundle bundle) {

            }

            @Override
            public void onBeginningOfSpeech() {

            }

            @Override
            public void onRmsChanged(float v) {

            }
            @Override
            public void onBufferReceived(byte[] bytes) {

            }

            @Override
            public void onEndOfSpeech() {

                PlayerActivity.mediaPlayer.start();
            }

            @Override
            public void onError(int i) {
                Toast.makeText(getApplicationContext(), "Unrecognized Command",
Toast.LENGTH_LONG).show();

            }

            @Override
            public void onResults(Bundle bundle) {


                ArrayList<String> matchesFound =
                        bundle.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
                if(matchesFound != null) {
                    keeper = matchesFound.get(0);
                    if (keeper.equals("pause the song") || keeper.equals("pause song") ||
                            keeper.equals("pause")) {
                        PlayerActivity.mediaPlayer.pause();
                        Toast.makeText(getApplicationContext(), keeper,
                                Toast.LENGTH_LONG).show();
                    } else if (keeper.equals("play the song") || keeper.equals("play song") ||
                            keeper.equals("play")) {
                        PlayerActivity.mediaPlayer.start();
                        Toast.makeText(getApplicationContext(), keeper,
                                Toast.LENGTH_LONG).show();
                    } else if (keeper.equals("fast forward") || keeper.equals("forward")) {
                        player.ff();
                        Toast.makeText(getApplicationContext(), "Forwarded 10 Second",
                                Toast.LENGTH_LONG).show();
                    } else if (keeper.equals("fast rewind") || keeper.equals("rewind")) {
                        player.ff();
                        Toast.makeText(getApplicationContext(), "Rewinded 10 Second",
                                Toast.LENGTH_LONG).show();
                    }
                }
            }

            @Override
            public void onPartialResults(Bundle bundle) {

            }
```

```
            @Override
            public void onEvent(int i, Bundle bundle) {

            }
        });

        micWidget.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                    PlayerActivity.mediaPlayer.pause();
                    speechRecognizer.startListening(speechRecognizerIntent);

                }
        });



        return START_STICKY;
    }



        @Override
        public void onDestroy(){
            super.onDestroy();
            if (mFloatingView != null) {
                windowManager.removeView(mFloatingView);
            }
            if (imageClose != null) {
                windowManager.removeView(imageClose);
            }
        }
    }
```

activity_main.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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/black"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/listViewSong"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/background"
        android:divider="@android:color/transparent"
        android:dividerHeight="10.0sp"
        android:padding="8dp"></ListView>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/button_widget"
        android:layout_width="120dp"
        android:layout_height="60dp"
        android:text = "Voice Command"
        android:background="@android:color/black"
        android:textColor="@color/white"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true">

    </androidx.appcompat.widget.AppCompatButton>

  </RelativeLayout>
```

<u>activity_player.xml</u>

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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/background"
    android:orientation="vertical"
    android:weightSum="10"
    tools:context=".PlayerActivity">


    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="7"
        android:gravity="center"
        android:orientation="vertical"
        >
        <TextView
            android:id="@+id/txtsn"
            android:layout_margin="20dp"
            android:ellipsize="marquee"
            android:marqueeRepeatLimit="marquee_forever"
            android:padding="10dp"
            android:singleLine="true"
            android:text="Song Name"
            android:textColor="#FFF"
            android:textSize="22sp"
            android:textAlignment="center"
            android:textStyle="italic"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            ></TextView>
        <ImageView
            android:id="@+id/imageview"
            android:layout_marginBottom="8dp"
            android:layout_width="250dp"
            android:layout_height="250dp"
            android:src="@drawable/disc">

        </ImageView>
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="60dp"
            >
            <SeekBar
                android:id="@+id/seekbar"
                android:layout_centerInParent="true"
                android:layout_alignParentBottom="true"
                android:layout_margin="20dp"
                android:layout_marginBottom="40dp"
                android:layout_width="250dp"
                android:layout_height="wrap_content"
                >

            </SeekBar>
            <TextView
                android:id="@+id/txtsstart"
                android:layout_toLeftOf="@+id/seekbar"
                android:layout_centerInParent="true"
                android:layout_alignParentLeft="false"
                android:layout_marginLeft="20dp"
                android:text="0:10"
                android:textColor="#FFF"
                android:textSize="14sp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                >

            </TextView>
            <TextView
                android:id="@+id/txtsstop"
               android:layout_toRightOf="@+id/seekbar"
                android:layout_centerInParent="true"
                android:layout_alignParentRight="false"
                android:layout_marginRight="20dp"
                android:text="4:10"
                android:textColor="#FFF"
                android:textSize="14sp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                >

            </TextView>


        </RelativeLayout>
     </LinearLayout>
     <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="3">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">
            <androidx.appcompat.widget.AppCompatButton
                android:id="@+id/playbtn"
                android:layout_centerHorizontal="true"
                android:background="@drawable/ic_pause"
                android:layout_width="70dp"
                android:layout_height="70dp"
    ></androidx.appcompat.widget.AppCompatButton>

            <androidx.appcompat.widget.AppCompatButton
                android:id="@+id/btnnext"
                android:layout_toRightOf="@+id/playbtn"
                android:layout_marginTop="15dp"
                android:background="@drawable/ic_next"
                android:layout_width="50dp"
                android:layout_height="50dp"
    ></androidx.appcompat.widget.AppCompatButton>

            <androidx.appcompat.widget.AppCompatButton
                android:id="@+id/btnprev"
                android:layout_toLeftOf="@+id/playbtn"
                android:layout_marginTop="15dp"
                android:background="@drawable/ic_prev"
                android:layout_width="50dp"
                android:layout_height="50dp"
    ></androidx.appcompat.widget.AppCompatButton>
            <androidx.appcompat.widget.AppCompatButton
                android:id="@+id/btnff"
                android:layout_toRightOf="@id/btnnext"
                android:layout_marginTop="20dp"
                android:layout_marginLeft="15dp"
                android:background="@drawable/ic_fast_forward"
                android:layout_width="40dp"
                android:layout_height="40dp"
    ></androidx.appcompat.widget.AppCompatButton>
            <androidx.appcompat.widget.AppCompatButton
                android:id="@+id/btnfr"
                android:layout_toLeftOf="@id/btnprev"
                android:layout_marginTop="20dp"
                android:layout_marginRight="15dp"
                android:background="@drawable/ic_fast_rewind"
                android:layout_width="40dp"
                android:layout_height="40dp"
    ></androidx.appcompat.widget.AppCompatButton>
            <com.gauravk.audiovisualizer.visualizer.BarVisualizer
                xmlns:custom="http://schemas.android.com/apk/res-auto"
                android:id="@+id/blast"
                android:layout_width="match_parent"
                android:layout_height="70dp"
                android:layout_alignParentBottom="true"
                custom:avDensity="0.5"
                custom:avType="outline"
                custom:avWidth="4dp"
                custom:avColor="#FF362E"
                custom:avSpeed="normal"/>

        </RelativeLayout>

     </LinearLayout>

    </LinearLayout>
```

activity_splashscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
   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"
   android:background="@color/colorAccent">

   <ImageView
      android:id="@+id/logo"
      android:layout_width="389dp"
      android:layout_height="512dp"
      android:src="@drawable/disc"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent"></ImageView>

   <TextView
      android:id="@+id/VC_command"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="0dp"
      android:layout_marginTop="0dp"
      android:padding="0dp"
      android:singleLine="true"
      android:text="INSTRUCTION"
```

```
    android:textAlignment="center"
    android:textColor="#FFF"
    android:textSize="22sp"
    android:textStyle="italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/logo" />
  <TextView
    android:id="@+id/VC_command_1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="0dp"
    android:layout_marginTop="0dp"
    android:singleLine="true"
    android:text="Tap on the mic button to trigger "
    android:textAlignment="center"
    android:textColor="#FFF"
    android:textSize="20sp"
    android:textStyle="italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/VC_command" />
  <TextView
    android:id="@+id/VC_command_2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="0dp"
    android:layout_marginTop="0dp"
    android:singleLine="true"
    android:text="following voice command"
    android:textAlignment="center"
    android:textColor="#FFF"
    android:textSize="20sp"
    android:textStyle="italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/VC_command_1" />

  <TextView
    android:id="@+id/VC_command_3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="0dp"
    android:layout_marginTop="0dp"
    android:singleLine="true"
    android:text="PLAY"
    android:textAlignment="center"
```

```
        android:textColor="#FFF"
        android:textSize="20sp"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/VC_command_2" />
    <TextView
        android:id="@+id/VC_command_4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="0dp"
        android:layout_marginTop="0dp"
        android:singleLine="true"
        android:text="PAUSE"
        android:textAlignment="center"
        android:textColor="#FFF"
        android:textSize="20sp"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/VC_command_3" />
    <TextView
        android:id="@+id/VC_command_5"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="0dp"
        android:layout_marginTop="0dp"
        android:singleLine="true"
        android:text="FORWARD"
        android:textAlignment="center"
        android:textColor="#FFF"
        android:textSize="20sp"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/VC_command_4" />
    <TextView
        android:id="@+id/VC_command_6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="0dp"
        android:layout_marginTop="0dp"
        android:singleLine="true"
        android:text="REWIND"
        android:textAlignment="center"
        android:textColor="#FFF"
        android:textSize="20sp"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/VC_command_5" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

layout_widget.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">

    <RelativeLayout
        android:id="@+id/layout_widget"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_margin="30dp">

        <Button
            android:id="@+id/mic_widget"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:background="@drawable/mic"
            android:gravity="center" />
    </RelativeLayout>


</RelativeLayout>
```

list_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_marginEnd="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginStart="8dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="8dp"
        android:background="@color/black">
        // android:background="@drawable/list_bg" >
        <ImageView
            android:id="@+id/imgsong"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:layout_marginStart="5dp"
            android:layout_marginTop="5dp"
            android:src="@drawable/music_icon"
            android:layout_width="40dp"
            android:layout_height="40dp">


        </ImageView>

        <TextView
            android:id="@+id/txtsongname"
            android:layout_marginTop="5dp"
            android:layout_alignParentEnd="true"
            android:layout_marginStart="5dp"
            android:layout_marginEnd="5dp"
            android:padding="6dp"
            android:textColor="@color/white"
            android:text="Song Name"
            android:textSize="17sp"
            android:singleLine="true"
            android:marqueeRepeatLimit="marquee_forever"
            android:ellipsize="marquee"
            android:scrollHorizontally="true"
            android:layout_toEndOf="@+id/imgsong"
            android:layout_width="wrap_content"
            android:layout_height="40dp"
            android:background="@drawable/musicname_frame">

        </TextView>
    </RelativeLayout>

</androidx.cardview.widget.CardView>
```

## Appendix B: Poster

Chong Zheng Kit
18ACB04784

# QendaMP

This music player's specialty is that the users are able to **voice control** the music player. However, users are able to use the music player as a normal music player too. This music player is able to scan the phone storage for the list of the songs and are consist of basic function that a music player possessed.

## FEATURES

**To list out all the song list that exist in the device.**

After the proposed application have scan the device for songs, the application will list out all the songs to let users to view. Song title and artist's name will be displayed.

**To allow users to voice command the application**

Users will be able to voice command this music player by simply tap on the mic mutton. The music player will be able to function based on the command given.

**To play music that exist in the song list**

This is the main function of this application. Any song being scanned and include in the list will be able to play.

**To allow users to voice command the application in background.**

Users will be able to voice command this music player by simply tap on the mic mutton. The music player will be able to function based on the command given.

**To allow users to play music in the background**

Users will be able to play music with this application even if the screen is off or using other applications.

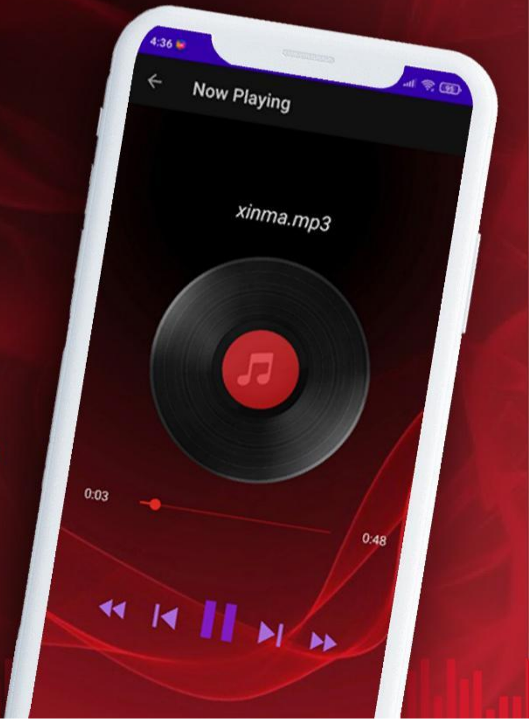
## METHODOLOGY

**Identification ▶ Design ▶**

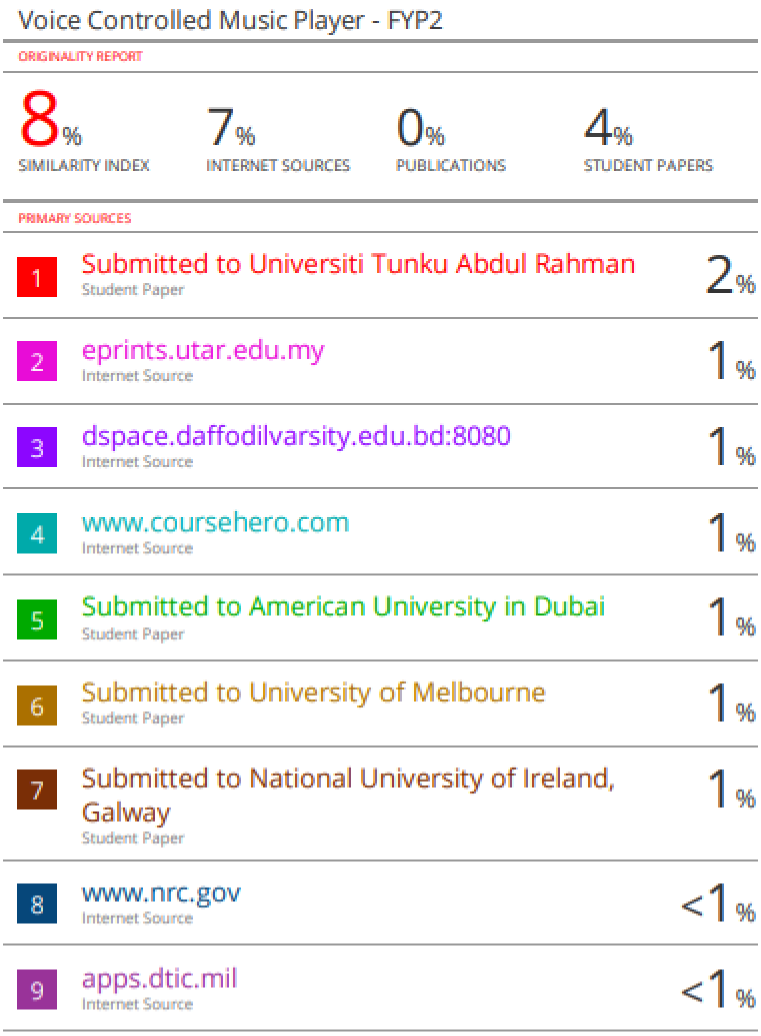
**Development ▶ Prototyping ▶**

**Testing ▶ Deployment ▶**

**Maintenance**



## Appendix C: Plagiarism Check Result



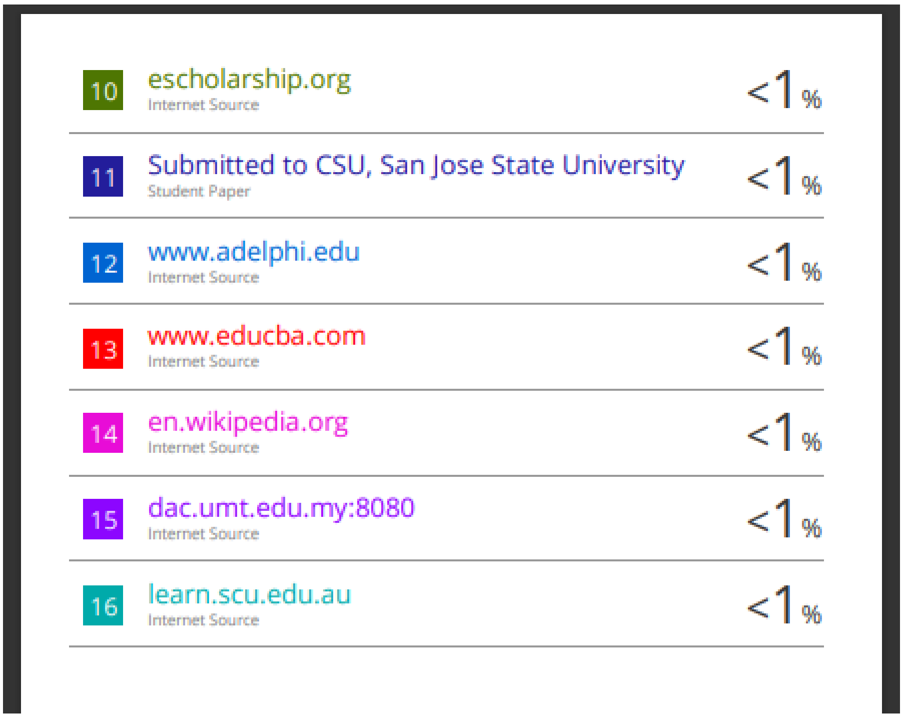
Voice Controlled Music Player - FYP2

ORIGINALITY REPORT

| 8% | 7% | 0% | 4% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

|  |  |  |
|---|---|---|
| 1 | Submitted to Universiti Tunku Abdul Rahman<br>Student Paper | 2% |
| 2 | eprints.utar.edu.my<br>Internet Source | 1% |
| 3 | dspace.daffodilvarsity.edu.bd:8080<br>Internet Source | 1% |
| 4 | www.coursehero.com<br>Internet Source | 1% |
| 5 | Submitted to American University in Dubai<br>Student Paper | 1% |
| 6 | Submitted to University of Melbourne<br>Student Paper | 1% |
| 7 | Submitted to National University of Ireland, Galway<br>Student Paper | 1% |
| 8 | www.nrc.gov<br>Internet Source | <1% |
| 9 | apps.dtic.mil<br>Internet Source | <1% |



|  |  |  |
|---|---|---|
| 10 | escholarship.org<br>Internet Source | <1% |
| 11 | Submitted to CSU, San Jose State University<br>Student Paper | <1% |
| 12 | www.adelphi.edu<br>Internet Source | <1% |
| 13 | www.educba.com<br>Internet Source | <1% |
| 14 | en.wikipedia.org<br>Internet Source | <1% |
| 15 | dac.umt.edu.my:8080<br>Internet Source | <1% |
| 16 | learn.scu.edu.au<br>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)** | Chong Zheng Kit |
| **ID Number(s)** | 18ACB04784 |
| **Programme / Course** | Bachelor of Computer Science |
| **Title of Final Year Project** | Voice Controlled Music Player |

| **Similarity** | **Supervisor's Comments**<br>**(Compulsory if parameters of originality exceeds the limits approved by UTAR)** |
|---|---|
| **Overall similarity index: 8 %**<br>**Similarity by source**<br>Internet Sources: 7%<br>Publications: 0%<br>Student Papers: 4 % | 4% from the student paper is due to index. This can be ignored. |
| **Number of individual sources listed** of more than 3% similarity: 0 | |
| **Parameters of originality required and limits approved by UTAR are as Follows:**<br>**(i) Overall similarity index is 20% and below, and**<br>**(ii) Matching of individual sources listed must be less than 3% each, and**<br>**(iii) Matching texts in continuous block must not exceed 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

Signature of Co-Supervisor

Name: Ts Dr Ooi ChekYee

Date 12 April 2022

Name: Dr Mogana a/p Vadiveloo

Date:

## Appendix D: FYP2 Checklist

# UNIVERSITI TUNKU ABDUL RAHMAN

## FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

### CHECKLIST FOR FYP2 THESIS SUBMISSION

| | |
|---|---|
| Student Id | 18ACB04784 |
| Student Name | CHONG ZHENG KIT |
| Supervisor Name | TS DR OOI CHEK YEE |

| TICK (√) | DOCUMENT ITEMS<br>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 Plastic Cover (for hardcopy) |
| √ | Title Page |
| √ | Signed Report Status Declaration Form |
| √ | Signed FYP Thesis Submission Form |
| √ | 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) |
| √ | Weekly Log |
| √ | Poster |
| √ | Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005) |
| √ | I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report. |

*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:

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 1** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

| |
|---|
| **1. WORK DONE** |
| **2. WORK TO BE DONE**<br>**Analysis requirements**<br>**Review previous works**<br>**Design**<br>**Coding**<br>**Debugging**<br>**Prototype**<br>**Report documentation** |
| **3. PROBLEMS ENCOUNTERED**<br>**-** |
| **4. SELF EVALUATION OF THE PROGRESS**<br>**Everything is fine.** |

________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 2** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

| |
|---|
| **1. WORK DONE**<br>**Analysis requirements** |
| **2. WORK TO BE DONE**<br>**Review previous works**<br>**Design**<br>**Coding**<br>**Debugging**<br>**Prototype**<br>**Report documentation** |
| **3. PROBLEMS ENCOUNTERED**<br>**-** |
| **4. SELF EVALUATION OF THE PROGRESS**<br>**Everything is fine.** |

________________________ __ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 3** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

| |
|---|
| **1. WORK DONE**<br>**Analysis requirements**<br>**Review previous works** |
| **2. WORK TO BE DONE**<br>**Design**<br>**Coding**<br>**Debugging**<br>**Prototype**<br>**Report documentation** |
| **3. PROBLEMS ENCOUNTERED**<br>**-** |
| **4. SELF EVALUATION OF THE PROGRESS**<br>**Everything is fine.** |

_________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 4** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

| |
|---|
| **1. WORK DONE**<br>**Analysis requirements**<br>**Review previous works**<br>**Design** |
| **2. WORK TO BE DONE**<br>**Coding**<br>**Debugging**<br>**Prototype**<br>**Report documentation** |
| **3. PROBLEMS ENCOUNTERED**<br>**-** |
| **4. SELF EVALUATION OF THE PROGRESS**<br>**Everything is fine.** |

________________________ __ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 5** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

**1. WORK DONE**

**Analysis requirements**

**Review previous works**

**Design**

**2. WORK TO BE DONE**

**Coding**

**Debugging**

**Prototype**

**Report documentation**

**3. PROBLEMS ENCOUNTERED**

**-**

**4. SELF EVALUATION OF THE PROGRESS**

**Everything is fine.**

________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 6** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

| |
|---|
| **1. WORK DONE**<br>**Analysis requirements**<br>**Review previous works**<br>**Design** |
| **2. WORK TO BE DONE**<br>**Coding**<br>**Debugging**<br>**Prototype**<br>**Report documentation** |
| **3. PROBLEMS ENCOUNTERED**<br>**-** |
| **4. SELF EVALUATION OF THE PROGRESS**<br>**Everything is fine.** |

________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 7** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

**1. WORK DONE**

**Analysis requirements**

**Review previous works**

**Design**

**2. WORK TO BE DONE**

**Coding**

**Debugging**

**Prototype**

**Report documentation**

**3. PROBLEMS ENCOUNTERED**

**-**

**4. SELF EVALUATION OF THE PROGRESS**

**Everything is fine.**

________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 8** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

**1. WORK DONE**

**Analysis requirements**

**Review previous works**

**Design**

**Coding**

**Debugging**

**2. WORK TO BE DONE**

**Prototype**

**Report documentation**

**3. PROBLEMS ENCOUNTERED**

**-**

**4. SELF EVALUATION OF THE PROGRESS**

**Everything is fine.**

________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 9** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

| |
|---|
| **1. WORK DONE**<br>**Analysis requirements**<br>**Review previous works**<br>**Design**<br>**Coding**<br>**Debugging**<br>**Prototype** |
| **2. WORK TO BE DONE**<br>**Report documentation** |
| **3. PROBLEMS ENCOUNTERED**<br>**-** |
| **4. SELF EVALUATION OF THE PROGRESS**<br>**Everything is fine.** |

_________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 10** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

| |
|---|
| **1. WORK DONE**<br>**Analysis requirements**<br>**Review previous works**<br>**Design**<br>**Coding**<br>**Debugging**<br>**Prototype** |
| **2. WORK TO BE DONE**<br>**Report documentation** |
| **3. PROBLEMS ENCOUNTERED**<br>**-** |
| **4. SELF EVALUATION OF THE PROGRESS**<br>**Everything is fine.** |

________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Y3S3** | **Study week no.: 11** |
| **Student Name & ID: Chong Zheng Kit 18ACB04784** | |
| **Supervisor: Ts Dr. Ooi Chek Yee** | |
| **Project Title: Voice Controlled Music Player** | |

**1. WORK DONE**

**Analysis requirements**

**Review previous works**

**Design**

**Coding**

**Debugging**

**Prototype**

**Report documentation**

**2. WORK TO BE DONE**

**3. PROBLEMS ENCOUNTERED**

**-**

**4. SELF EVALUATION OF THE PROGRESS**

**Everything is done.**

________________________ _ Ooi Chek Yee

Student's signature Supervisor's signature