MP3 PLAYER USING ANDROID STUDIO

A MINI PROJECT REPORT

Submitted by

GOKULAKRISHNAN K J (310620104045)AKASH

SARVESH M (310620104008)

DEVENDRAA KODAL (310620104036)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

EASWARI ENGINEERING COLLEGE

CHENNAI-600089

(Autonomous Institution)

Affiliated to

ANNA UNIVERSITY: CHENNAI - 600025

MAY 2023

EASWARI ENGINEERING COLLEGE, CHENNAI

(Autonomous Institution)

Affiliated to

ANNA UNIVERSITY: CHENNAI - 600025

BONAFIDE CERTIFICATE

Certified that this project report **"MP3 PLAYER USING ANDROID STUDIO"** is the bonafide work of"**GOKULAKRISHNAN (310620104045) AKASH SARVESH (310620104008) AND DEVENDRAA KODAL (310620104036)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Internal	**Head of the Department**
Mr.K.SHANKAR	Dr.G.S.ANANDHA MALA
Assisstant Professor	Professor
Department of Computer	Department of Computer
Science and Engineering	Science and Engineering
Easwari Engineering College,	Easwari Engineering College,
Ramapuram, Chennai 600089	Ramapuram, Chennai 600089

Submitted for Semester Mini-Project viva-voce examination held on _____

INTERNAL EXAMINER. **EXTERNAL EXAMINER**

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We hereby place our deep sense of gratitude to our beloved Founder Chairman of the institution**, Dr.T.R.Pachamuthu, B.Sc., M.I.E.,** for providing us with the requisite infrastructure throughout the course. We would also like to express our gratitude towards our Chairman **Dr.R.Shivakumar, M.D., Ph.D.** for giving the necessary facilities.

We convey our sincere thanks to **Dr.R.S.Kumar, M.Tech., Ph.D.** Principal Easwari Engineering College, for his encouragement and support. We extend our hearty thanks to **Dr.V.Elango, M.E., Ph.D.**, Vice Principal (academics) and **Dr.S.Nagarajan, M.E, Ph.D**, Vice Principal (admin), Easwari Engineering College, for their constant encouragement.

We take the privilege to extend our hearty thanks to **Dr.G.S.Anandha Mala, M.S, M.E, Ph.D.,** Head of the Department, Computer Science and Engineering, Easwari Engineering College for her suggestions, support and encouragement towards the completion of the project with perfection.

We would like to express our gratitude to our Mini Project Coordinator, **Mr. K.Shankar, M.Tech., (Ph.D)** Assistant Professor, Department of Computer Science and Engineering, Easwari Engineering College, for his constant support and encouragement.

Finally, we wholeheartedly thank all the faculty members of the Department of Computer Science and Engineering for warm cooperation and encouragement.

ABSTRACT

The MP3 player application is a software program that enables users to play and manage their music files on their electronic devices. This application provides an easy-to-use interface that allows users to browse their music library, create playlists, and customize playback settings. The application is designed to support a wide range of audio formats, including MP3, WAV, and FLAC. It also features advanced playback controls, such as shuffle, repeat, and crossfade, that enhance the listening experience. Additionally, the MP3 player application includes features such as equalizer presets and lyrics display, which add to the user's overall enjoyment of their music. With its intuitive interface and versatile functionality, the MP3 player application is an essential tool for any music lover.

CHAPTER 1

INTRODUCTION

1.1 Overview of the project

The project is a Simple MP3 Player Application is a user-friendly software program thatallows users to play and manage their music files on their electronic devices. The application provides a straightforward and easy-to-use interface that makes it accessible to all users. It supports popular audio formats, including MP3, WAV, and FLAC, and allows users to browse their music library, create playlists, and customize playback settings. The application also includes standard playback controls such as play, pause, stop, next, and previous, making it simple for users to control their music playback. Additionally, the application features an equalizer for adjusting the sound output and theability to display lyrics while playing a song. Overall, the Simple MP3 Player Application is an excellent choice for those who want a basic and uncomplicated music player that gets the job done.

Application features

Music library: The application allows users to browse their music library and organize their music by albums, artists, or playlists.

Playback controls: The application includes standard playback controls such as play, pause, stop, next, and previous for easy control of the music playback.

Audio formats: The application supports various audio formats, including MP3, WAV, and FLAC, among others.

Overall, an MP3 player application provides an intuitive and user-friendly interface that makes it easy for users to play and manage their music collection.

Following are the user interface components that will be shown in our application

There will be two drop down menu to choose the start date and end date in order to calculate the time interval and a “calculate” button is present when pressed it generates the time interval.

There will be a “FORWARD” button available below the grid view to generate five seconds forward

There will be a “BACKWARD” button available below the grid view to generate five seconds reverse.

There will be a “PAUSE” button available below the grid view to pause the song.

1.2 Aim of the project

The aim of a MP3 Player Application is to provide a user-friendly software program that enables users to play and manage their music files easily. The primary focus of a Simple MP3 Player Application is on simplicity and ease of use, with a straightforward interface that allows users to navigate their music library, create playlists, and customizeplayback settings with ease.

CHAPTER 2
REQUIREMENT SPECIFICATION

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behaviour of the system.

2.1 Functional requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. The various methods used in this project are as follows:-

Emulator to perform and display the functionality of the project. Android studio to create, design, test, debug and run the android project.Mouse to navigate through the emulator.

Keyboard to give inputs to the project.

2.2 Non-functional requirements

These are the constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-functional requirements often apply to the system as a whole.

Non-Functional Requirements are as follows:-

2.2.1. Performance: The application should respond quickly to user input and should not have any noticeable lag or delay when performing calculations.

2.2.2. Usability: The user interface should be intuitive and easy to use, with clear instructions and error messages if necessary.

2.2.3. Security: The application should not store any sensitive user data, and should ensure that user inputs are validated and sanitized to prevent any security vulnerabilities.

2.2.4. Compatibility: The application should be compatible with a wide range of Android devices, screen sizes, and operating system versions.

2.2.5. Accessibility: The application should be designed to be accessible to users with disabilities, including support for accessibility features like screen readers and high-contrast mode.

2.2.6. Maintainability: The code should be well-structured, documented, and easy to maintain, with clear separation of concerns and reusable components.

2.3 Details of the software

Here, the coding, creating, designing, testing, debugging and running of our project is done in Android Studio.

2.3.1 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS

and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++,and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that varyby platform version." External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

2.4 Software and Hardware requirement

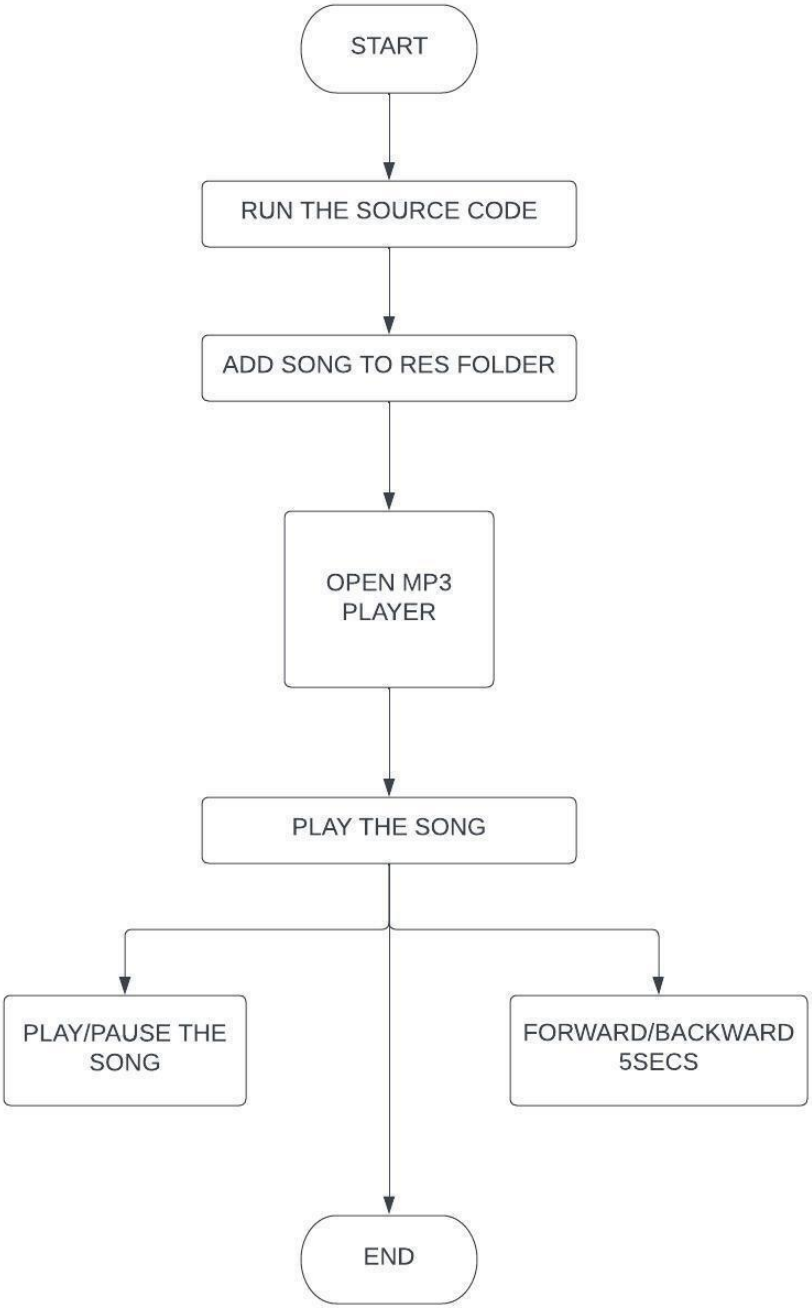
Basic system requirements for Android Studio

	Microsoft Windows	Mac	Linux
Operating System Version	Microsoft 庐 Windows 庐 7/8/10 (32- or 64-bit) The Android Emulator only supports 64-bit Windows.	Mac 庐 OS X 庐 10.10 (Yosemite) or higher, up to 10.14 (macOS Mojave)	GNOME or KDE desktop Tested on gLinux based on Debian (4.19.67-2rodete2).
Random Access Memory (RAM)	4 GB RAM minimum; 8 GB RAM recommended.		
Free digital storage	2 GB of available digital storage minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for 韃 Android SDK 韃 and emulator system image).		
Minimum required JDK version	Java Development Kit 韃 8		
Minimum screen resolution	1280 x 800		

CHAPTER 3

SYSTEM DESIGN

Data flow design is as shown below - covering the flow of the data in the system. It describes the relation between user input and the system behavior



CHAPTER 4

IMPLEMENTATION

To implement the Current system we have used different functions of our project which are as follows:

MainActivity.java

```
package com.example.media_player;

import androidx.appcompat.app.AppCompatActivity;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.SeekBar;

import java.nio.channels.SeekableByteChannel;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private Button btnp,btnb,btnf;
    private MediaPlayer player;
    private SeekBar sk;
    private Runnable runnable;
    private Handler handler;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnp=findViewById(R.id.btnplay);
        btnb=findViewById(R.id.btnbackward);
        btnf=findViewById(R.id.btnforward);
        sk=findViewById(R.id.seekbar);
        handler= new Handler();
        btnp.setOnClickListener(this);
        btnb.setOnClickListener(this);
```

```
    btnf.setOnClickListener(this);
    player=MediaPlayer.create(this,R.raw.song);
    player.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mediaPlayer) {
            sk.setMax(player.getDuration());
            player.start();
            changeSeekbar();
        }
    });
    sk.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
            if(b)
            {
                player.seekTo(i);
            }
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });
}

private void changeSeekbar() {
    sk.setProgress(player.getCurrentPosition());
    if(player.isPlaying())
    {
        runnable = new Runnable() {
```

```
            @Override
            public void run() {
                changeSeekbar();
            }
        };
        handler.postDelayed(runnable, 1000);
    }
}

@Override
public void onClick(View view) {
    switch (view.getId()){
        case R.id.btnplay:
            if(player.isPlaying())
            {
                player.pause();
                btnp.setText(">");
            }
            else
            {
                player.start();
                btnp.setText("||");
                changeSeekbar();
            }
            break;
        case R.id.btnforward:
            player.seekTo(player.getCurrentPosition()+5000);
            break;
        case R.id.btnbackward:
            player.seekTo(player.getCurrentPosition()-5000);
            break;
    }
}
}
```

Activity_main.xml

```
<?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:gravity="center"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="501dp"
        android:src="@drawable/ghani" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center">
    <Button
        android:id="@+id/btnbackward"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="Backward"/>
    <Button
        android:id="@+id/btnplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="||"/>
    <Button
        android:id="@+id/btnforward"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="Forward"/>
    </LinearLayout>
    <SeekBar
        android:id="@+id/seekbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>


</LinearLayout>
```

CHAPTER 5

SNAPSHOTS

CHAPTER 6
CONCLUSION

In conclusion, the age calculator app is a useful tool for calculating the age of an individual accurately. By using Joda Time library, the app is able to handle different dateformats and time zones, making it versatile and reliable. The app's user interface is user-friendly and easy to navigate, allowing users to input dates easily and get quick results. The project fulfilled its aim of creating an app that accurately calculates age and providesan intuitive user experience. With further improvements such as adding additional features and optimizing performance, the age calculator app can be even more useful and widely adopted.

CHAPTER 7

REFERNCES:

Websites

www.Tutorialslinks.com

www.Github.com

www.wikipedia.com

www.Youtube.com

www.stackoverflow.com