VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Santhibastawad Road, Machhe

Belagavi - 590018, Karnataka, India

MOBILE APPLICATION DEVELOPMENT PROJECT (18CSMP68) REPORT

ON

"MEDIA PLAYER APPLICATION"

Submitted in the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

For the Academic Year 2022-2023

Submitted by

Namratha Prakash	**1JS20IS054**
Neha Subrahmanya Hegde	**1JS20IS056**

Under the Guidance of

Mrs. Nagashree S
Assistant Professor
Dept. of ISE, JSSATEB

Ms. Sukrutha C Basappa
Assistant Professor
Dept. ISE, JSSATEB

2022-2023

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060.

JSS MAHAVIDYAPEETHA, MYSURU

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that MOBILE APPLICATION DEVELOPMENT MINI PROJECT (18CSMP68) Report entitled **"Media Player Application"** is a Bonafide work carried out by **Namratha Prakash [1JS20IS054]**, **Neha Subrahmanya Hegde [1JS20IS056]** in partial fulfillment for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University Belagavi during the year 2022- 2023.

Signature of the Guide	**Signature of the Guide**	**Signature of the HOD**
Mrs. Nagashree S	**Ms. Sukrutha C Basappa**	**Dr. Rekha PM**
Assistant Professor	**Assistant Professor**	**Professor & HOD**
Dept. of ISE	**Dept. of ISE**	**Dept. of ISE**
JSSATEB	**JSSATEB**	**JSSATEB**

External Viva

Name of the Examiners **Signature and Date**

1.

2.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So with gratitude, we acknowledge all those whose guidance and encouragement crowned our efforts with success.

First and foremost, we would like to thank his **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Bhimasen Soragaon**, Principal, JSSATE, Bangalore for providing an opportunity to carry out the Project Work as a part of our curriculum in the partial fulfilment of the degree course.

We express our sincere gratitude for our beloved Head of the department, **Dr. Rekha P.M**, for her co-operation and encouragement at all the moments of our approach.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guide **Mrs. Nagashree S**, Assistant Professor and **Ms. Sukrutha C Basappa,** Assistant Professor for the constant encouragement, valuable help and assistance in every possible way.

We would like to thank all **ISE Department Teachers** and **Non-teaching staff** for providing us with their valuable guidance and for being there at all stages of our work.

Namratha Prakash [1JS20IS054]
Neha Subrahmanya Hegde [1JS20IS056]

ABSTRACT

The Media Player Application project aims to develop a versatile and user-friendly media player application for Android devices using the Android Studio development environment. The application allows users to organize and play the music collections, providing a seamless and enjoyable music listening experience.

The project utilizes the Android Studio IDE, which offers a powerful platform for developing Android applications and offers a range of functionalities. The application incorporates the User Interface Design, an intuitive and visually appealing user interface, designed to enhance the overall user experience. It includes features such as Playlist display, song information, playback controls and Volume controls.

This Application provides a range of playback controls, allowing users to play, pause, forward, rewind and repeat songs. It also includes a seek bar for precise control over the current playback position. The application also integrates an Volume control, enabling users to adjust the sound output to their preferences. Users can modify the sound output by manually adjusting the volume level.

The development of the Media Player Application involves utilizing Android Studio's development tools, including the Android Software Development Kit (SDK), Java programming language, and various Android APIs and libraries. The project also focuses on optimizing performance, ensuring smooth audio playback, and adhering to design principles for a cohesive user interface.

Overall, the Media Player Application project aims to deliver a feature-rich and robust Media player application for Android devices. The application strives to provide users immersive music listening experience, enhancing their enjoyment and convenience on their mobile devices.

CHAPTER 1 INTRODUCTION

OVERVIEW

The media player application developed in Java using Android Studio brings an enhanced music playback experience to Android devices. With its sleek and user-friendly interface, the app offers a comprehensive list of songs for users to explore. Upon selecting a song, they are seamlessly transported to a dedicated player page where they can effortlessly control their music. The player page showcases album artwork, song information, and provides easy-to-use controls for playback. The next and previous buttons enable smooth navigation between tracks, while the volume seek feature allows for personalized audio levels. Additionally, the app offers a convenient looping function, allowing users to enjoy their favorite songs on repeat. Whether it's a catchy chorus or a soothing instrumental, users can keep the music playing in an endless loop, creating a truly personalized listening experience. Whether commuting, exercising, or relaxing, this media player provides an immersive and customizable music listening experience that caters to the diverse preferences of Android users.

PROBLEM STATEMENT

The development of a Media Player Application aims to address several challenges and requirements to provide a user-friendly and immersive music listening experience. The challenges include incorporating volume and seek controls for adjusting audio levels and seeking to specific song positions. Additionally, intuitive play, pause, forward, and backward controls are needed to navigate through the music library effortlessly. The application should also provide a looping feature, allowing users to repeat favorite songs or sections. Displaying song duration, singer, movie, and song artwork enhances the user experience with contextual information. Furthermore, the application should support playlist management, enabling users to view and play the songs in playlist. These features aim to enhance user control, convenience, and customization in their music playback experience.

CHAPTER 2

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- RAM: 8 GB RAM minimum
- Processor: Intel 5 or AMD 5
- Storage: approximately 10GB storage
- Internet Connectivity

SOFTWARE REQUIREMENTS

- Operating System: Android OS version 5.0 (Lollipop) or later.
- IDE: Android Studio
- Programming Language: JAVA
- Android Software Development Kit (SDK)
- API: Java Development Kit (JDK)

CHAPTER 3

SYSTEM DESIGN AND FUNCTIONALITIES

SYSTEM FUNCTIONALITIES

1. **List of songs**: The system initializes the media player by creating a list of songs along with the play button and displaying it on the screen using the Android Studio interface.
2. **User Input:** The system captures touch gestures from the user, allowing them to control the song selection.
3. **Play/Pause**: The player activity includes a play button that starts or pauses the playback of the selected song. The button toggles between the play and pause icons.
4. **Next and Previous**: Users can navigate through the playlist by clicking the next and previous buttons. These buttons increment or decrement the current song index, allowing seamless switching between songs.
5. **Volume Control**: The application provides a seek bar for controlling the volume. Users can adjust the volume level by dragging the seek bar thumb, resulting in immediate changes to the audio output.
6. **Song Looping**: A toggle button is available to enable or disable song looping. When enabled, the currently playing song repeats continuously until the user chooses to disable the loop functionality.
7. **Fast-rewind:** This feature enables users to quickly navigate backward within the song, allowing them to easily revisit or skip sections as desired.
8. **Fast-forward:** This feature enables users to quickly navigate forward within the song, allowing them to easily skip sections as desired.
9. **Time-Duration:** This functionality enhance the user experience by giving them accurate information about the remaining time.
10. **Song-seek:** With song seek bar functionality, users can interact with the seek bar to navigate to specific positions within the song.

XML DESIGN

The XML design for the Media Player Application encompasses the visual layout and structure of the user interface, defining how different components and elements are arranged and displayed. The XML files play a crucial role in specifying the design and appearance of the application's screens and views.

Here is a description of the key XML design elements in the Media Player Application:

- **Activity Layouts:** The main activity layout file defines the overall structure of the application's main screen. It typically includes a root ViewGroup, such as a RelativeLayout or ConstraintLayout, that acts as a container for other UI elements.

- **Buttons and Controls**: The XML design includes various buttons and controls for user interaction. These can include play, pause, next, and previous buttons, which allow users to control the music playback. These elements are typically implemented using <Button>, <ImageButton>, <ToggleButton> or similar elements.

- **Seekbar**: The application incorporates a SeekBar element to provide users with a visual representation of the current playback position within a song. The SeekBar can be customized with attributes such as progress, thumb appearance, and seek functionality.

- **ImageViews:** ImageViews are used to display song artwork or other graphical elements associated with songs. These elements are implemented using the <ImageView> element and can be customized with attributes such as scale type, padding, and tinting.

- **TextViews:** TextViews are utilized for displaying text-based information, such as song titles, artist names, and song durations. They are implemented using the <TextView> element and can be customized with attributes related to text appearance, alignment, and formatting.

- **Other XML Elements:** The XML design may also incorporate other elements and attributes as per the application's requirements, such as LinearLayouts or CardViews for arranging and styling different UI elements.

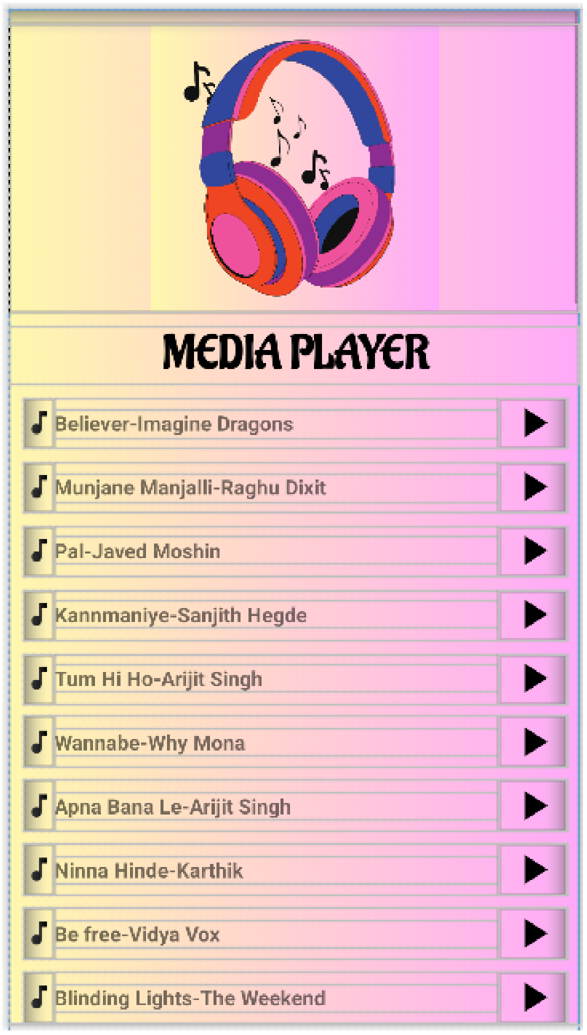


Fig 3.1 XML Design of Playlist

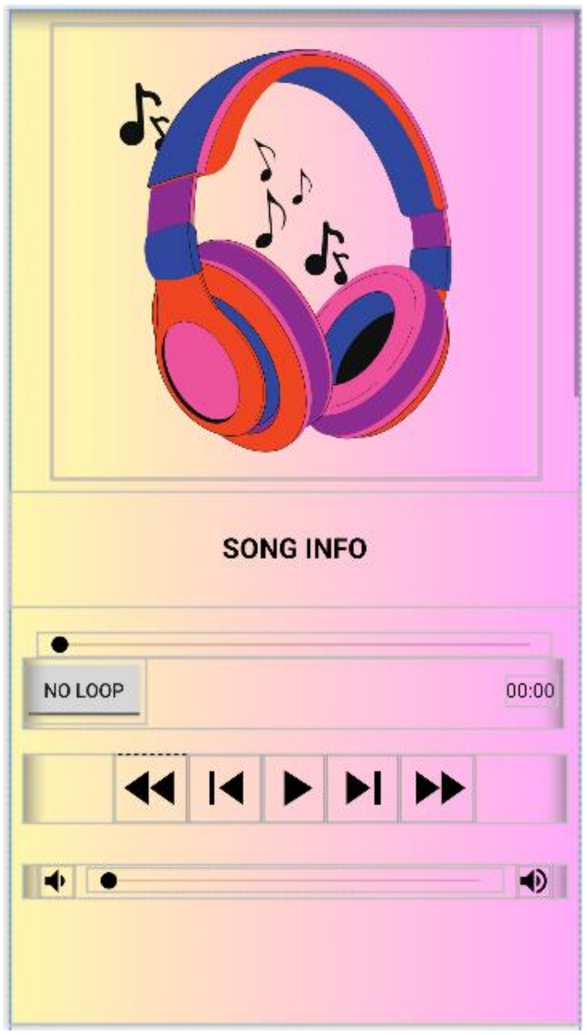


Fig 3.2 XML Design of Music Player

Overall, the XML design in the Media Player Application defines the structure and appearance of the user interface, incorporating various UI elements, layouts, and views. Through a combination of XML layouts and corresponding item layouts, the design creates a visually appealing and user-friendly interface for users to navigate, interact with, and enjoy their music collection. The inclusion of a SeekBar allows for precise control and visualization of the playback position within songs, enhancing the overall music playback experience

CHAPTER 4

SYSTEM IMPLEMENTATION

IMPLEMENTATION

1. Set up the Android Studio development environment and create a new project.

2. **User Interface Design:**
   - Create layouts for the main activity and player activity.
   - Design the song list view using a scroll view and linear layout.
   - Include image views and text views to display song information.

3. **Song List Implementation:**
   - Create an ArrayList to store the songs.
   - Create a **Raw folder** in the resource directory to store the music files. The "raw" folder is typically used to hold audio files in their original format, which can then be accessed and played by the Media Player Application.
   - Populate the song list with song resources or file paths.
   - Dynamically generate the song list view with the title and associated image.

4. **Player Activity Implementation:**
   - Pass the current song index from the main activity to the player activity using an intent.
   - Retrieve the song index in the player activity.
   - Initialize the media player with the selected song.
   - Handle play/pause functionality using a play button.
   - Display the song title and associated image.

5. **Navigation and Control:**
   - Implement next and previous buttons to switch between songs.
   - Update the media player with the new song when navigating.
   - Update the displayed song title and image accordingly.

6. **Volume Control:**
   - Add a seekbar to control the volume.
   - Retrieve the current volume level using the AudioManager.
   - Set the seekbar progress and update the volume when it changes.

7. **Song Looping:**
   - Include a toggle button for song looping.
   - Set the media player's looping property based on the toggle button's state.
   - Enable or disable song looping accordingly.

8. **Time Duration Display:**
   - Calculate the remaining time of the song by subtracting the current playback position from the total duration.
   - Format the remaining time into a readable format (e.g., minutes and seconds).
   - Update a text view to display the remaining time.

9. **Fast Forward Functionality:**
   - Implement a button for fast forwarding.
   - When the button is clicked, check if the media player is playing.
   - If it is, use the seekTo method of the media player to move the playback position forward by a specified duration (e.g., 10 seconds).

10. **Fast Rewind Functionality:**
    - Implement a button for fast rewinding.
    - When the button is clicked, check if the media player is playing.
    - If it is, use the seekTo method of the media player to move the playback position backward by a specified duration (e.g., 10 seconds).

11. **Song Seek Functionality:**
    - Implement a seekbar to display and control the current playback position.
    - Set the maximum value of the seekbar to the total duration of the song.
    - Update the seekbar progress based on the current playback position of the media player.

- When the user interacts with the seekbar, use the seekTo method of the media player to set the playback position to the selected position.

SOURCE CODES

MainActivity.java

```
package com.example.musicccc;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;


public class MainActivity extends AppCompatActivity  {
    ImageView
playyl,playy2,playy3,playy4,playy5,playy6,playy7,playy8,playy9,playy10,playy1
1,playy12,playy13,playy14,playy15;
    ImageView playy16,playy17,playy18,playy19,playy20;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        playy1=findViewById(R.id.play1);
        playy2=findViewById(R.id.play2);
        playy3=findViewById(R.id.play3);
        playy4=findViewById(R.id.play4);
        playy5=findViewById(R.id.play5);
        playy6=findViewById(R.id.play7);
        playy7=findViewById(R.id.play8);
        playy8=findViewById(R.id.play9);
        playy9=findViewById(R.id.play10);
        playy10=findViewById(R.id.play11);
        playy11=findViewById(R.id.play12);
        playy12=findViewById(R.id.play13);
        playy13=findViewById(R.id.play14);
        playy14=findViewById(R.id.play15);
        playy15=findViewById(R.id.play16);
        playy16=findViewById(R.id.play17);
        playy17=findViewById(R.id.play18);
        playy18=findViewById(R.id.play19);
        playy19=findViewById(R.id.play20);
        playy20=findViewById(R.id.play21);


        playy1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy1))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",0);
                    startActivity(it);
                }
            }
        });
        playy2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy2))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",1);
                    startActivity(it);

                }
            }
        });
        playy3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy3))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",2);
                    startActivity(it);

                }
            }
        });
        playy4.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy4))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",3);
                    startActivity(it);

                }
            }
        });
        playy5.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy5))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",4);
                    startActivity(it);

                }
            }
        });
        playy6.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy6))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",5);
                    startActivity(it);

                }
            }
        });
        playy7.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy7))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",6);
                    startActivity(it);

                }
            }
        });
        playy8.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy8))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",7);
                    startActivity(it);

                }
            }
        });
        playy9.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy9))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",8);
                    startActivity(it);

                }
            }
        });
        playy10.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy10))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",9);
                    startActivity(it);

                }
            }
        });
        playy11.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy11))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",10);
                    startActivity(it);

                }
            }
        });
        playy12.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy12))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",11);
                    startActivity(it);

                }
            }
        });
        playy13.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy13))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",12);
                    startActivity(it);

                }
            }
        });
        playy14.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy14))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",13);
                    startActivity(it);

                }
            }
        });
        playy15.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy15))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",14);
                    startActivity(it);

                }
            }
        });
        playy16.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy16))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",15);
                    startActivity(it);

                }
            }
        });
        playy17.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy17))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",16);
                    startActivity(it);

                }
            }
        });
        playy18.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy18))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",17);
                    startActivity(it);

                }
            }
        });
        playy19.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy19))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",18);
                    startActivity(it);

                }
            }
        });
        playy20.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (view.equals(playy20))
                {
                    Intent it;
                    it = new Intent (MainActivity.this,PlayerActivity.class);
                    it.putExtra("currentIndex",19);
                    startActivity(it);

                }
            }
        });
```

PlayerActivity.java

```
package com.example.musicccc;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.content.Context;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.View;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.ToggleButton;
import java.util.ArrayList;

public class PlayerActivity extends AppCompatActivity {
    ImageView play, prev, next, fastf, fastr,imageView;
    TextView songTitle, textstop;
    SeekBar mSeekBarTime, mSeekBarVol;
    static MediaPlayer mMediaPlayer;
    private Runnable runnable;
    private ToggleButton loopButton;
    private AudioManager mAudioManager;
    int currentIndex = 0;
    ArrayList<Integer> songs;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_player);
        mAudioManager = (AudioManager)
getSystemService(Context.AUDIO_SERVICE);
        play = findViewById(R.id.play);
        prev = findViewById(R.id.prev);
        next = findViewById(R.id.next);
        fastf = findViewById(R.id.fstf);
        fastr=findViewById(R.id.fstr);
        songTitle = findViewById(R.id.songTitle);
        imageView = findViewById(R.id.imageView);
        mSeekBarTime = findViewById(R.id.seekBarTime);
        mSeekBarVol = findViewById(R.id.seekBarVol);
        textstop=findViewById(R.id.txtsstop);
        loopButton=findViewById(R.id.loopb);

        songs = new ArrayList<>();
        songs.add(0, R.raw.believer);
        songs.add(1, R.raw.munjanemanjalli);
        songs.add(2, R.raw.pal);
        songs.add(3, R.raw.kannmaniye);
        songs.add(4, R.raw.tumhiho);
        songs.add(5, R.raw.wannabe);
        songs.add(6, R.raw.apnabanale);
        songs.add(7, R.raw.babyily);
        songs.add(8, R.raw.befree);
        songs.add(9, R.raw.blindinglights);
        songs.add(10, R.raw.dancemonkey);
        songs.add(11, R.raw.dandelions);
        songs.add(12, R.raw.doobey);
        songs.add(13, R.raw.fairy);
        songs.add(14, R.raw.galliyan);
        songs.add(15, R.raw.rkm);
        songs.add(16, R.raw.hello);
        songs.add(17, R.raw.rnjh);
        songs.add(18, R.raw.inkem);
        songs.add(19, R.raw.jeena);
        songs.add(20, R.raw.uify);

        currentIndex=getIntent().getIntExtra("currentIndex",0);
        // intializing mediaplayer
        mMediaPlayer = MediaPlayer.create(getApplicationContext(),
songs.get(currentIndex));
        mMediaPlayer.setOnPreparedListener(new
MediaPlayer.OnPreparedListener(){
            public void onPrepared(MediaPlayer mp){
                int totalDuration=mMediaPlayer.getDuration();
                mSeekBarTime.setMax(totalDuration);
                String
totalTime=String.format("%02d:%02d",totalDuration/60000,(totalDuration%60000)
/1000);
                textstop.setText(totalTime);
                mMediaPlayer.setLooping(loopButton.isChecked());
                updateSeekbar();
            }
        });

        // seekbar volume
        int maxV =
mAudioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC);
        int curV = mAudioManager.getStreamVolume(AudioManager.STREAM_MUSIC);
        mSeekBarVol.setMax(maxV);
        mSeekBarVol.setProgress(curV);

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

                mAudioManager.setStreamVolume(AudioManager.STREAM_MUSIC,
progress, 0);
            }
            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {

            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {

            }
        });
        loopButton.setOnClickListener(new View.OnClickListener(){
            public void onClick(View v){
                boolean loopEnabled=loopButton.isChecked();
                mMediaPlayer.setLooping(loopEnabled);
            }
        });
        //above seekbar volume
        fastr.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (mMediaPlayer.isPlaying())
                {
                    mMediaPlayer.seekTo(mMediaPlayer.getCurrentPosition()-
10000);
                }
            }
        });
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mSeekBarTime.setMax(mMediaPlayer.getDuration());
                if (mMediaPlayer != null && mMediaPlayer.isPlaying()) {
                    mMediaPlayer.pause();
                    play.setImageResource(R.drawable.ic_play);

                } else {
                    mMediaPlayer.start();
                    play.setImageResource(R.drawable.ic_pause);

                }

                songNames();

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

        prev.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if (mMediaPlayer != null) {
                    play.setImageResource(R.drawable.ic_pause);
                }
```

```
                    currentIndex--;
                if(currentIndex<0) {
                    currentIndex = songs.size() - 1;
                }
                if (mMediaPlayer.isPlaying()) {
                    mMediaPlayer.stop();
                }
                mMediaPlayer = MediaPlayer.create(getApplicationContext(),
songs.get(currentIndex));
                mMediaPlayer.start();
                songNames();
            }
        });
        fastf.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (mMediaPlayer.isPlaying())
                {

mMediaPlayer.seekTo(mMediaPlayer.getCurrentPosition()+10000);
                }
            }
        });
    }
    private void updateSeekbar(){
        int totalDuration=mMediaPlayer.getDuration();
        int remainingTime=totalDuration-mMediaPlayer.getCurrentPosition();
        String
remainingTimeString=String.format("%02d:%02d",remainingTime/60000,(remainingT
ime%60000)/1000);
        textstop.setText(remainingTimeString);
        mSeekBarTime.setProgress(mMediaPlayer.getCurrentPosition());
        handler.postDelayed(updateRunnable,100);
    }
    protected Runnable updateRunnable=new Runnable() {
        @Override
        public void run() {
            updateSeekbar();
        }
    };
    private void songNames() {
        switch(currentIndex){
          case 0:
            songTitle.setText("BELIEVER-IMAGINE DRAGONS");
            imageView.setImageResource(R.drawable.believer);
            break;
          case 1:
            songTitle.setText("MUNJANE MANJALLI-RAGHU DIXIT\nMovie:Just Math
Mathalli");
            imageView.setImageResource(R.drawable.jmm);
              break;
          case 2:
            songTitle.setText("PAL-JAVED MOSHIN\nMovie:Jalebi");
            imageView.setImageResource(R.drawable.pal);
              break;
          case 3:
            songTitle.setText("KANNMANIYE-SANJITH HEGDE\nMovie:Pailwan");
            imageView.setImageResource(R.drawable.kannmaniye);
              break;
          case 4:
            songTitle.setText("TUM HI HO-ARIJITH SINGH\nMovie:Aashiqui 2");
            imageView.setImageResource(R.drawable.tumhiho);
                break;
          case 5:
            songTitle.setText("WANNABE-WHY MONA");
            imageView.setImageResource(R.drawable.wann);
                break;
          case 6:
            songTitle.setText("APNA BANA LE-ARIJITH SINGH\nMovie:Bhediya");
            imageView.setImageResource(R.drawable.apn);
                break;
          case 7:
            songTitle.setText("NINNA HINDE-KARTHIK\nMovie:Maanikya");
            imageView.setImageResource(R.drawable.bbyil);
                break;
          case 8:
            songTitle.setText("BE FREE-VIDYA VOX");
            imageView.setImageResource(R.drawable.befree);
                break;
          case 9:
            songTitle.setText("BLINDING LIGHTS-THE WEEKEND");
            imageView.setImageResource(R.drawable.bl);
                break;
          case 10:
            songTitle.setText("DANCE MONKEY-TONES AND I");
            imageView.setImageResource(R.drawable.dm);
                break;
          case 11:
            songTitle.setText("DANDELIONS-RUTH B.");
            imageView.setImageResource(R.drawable.ddn);
                break;
          case 12:
            songTitle.setText("DOOBEY-OAFF\nMovie:Gehraiyaan");
            imageView.setImageResource(R.drawable.doob);
                break;
          case 13:
            songTitle.setText("FAIRYTALE-ALEXANDER RYBAK");
            imageView.setImageResource(R.drawable.ft);
                break;
          case 14:
            songTitle.setText("GALLIYAN-ANKIT TIWARI\nMovie:Ek Villain");
            imageView.setImageResource(R.drawable.gall);
                break;
          case 15:
            songTitle.setText("RAKKAMMA-NAKASH AZIZ,SUNIDHI
CHAUHAN\nMovie:Vikrant Rona");
            imageView.setImageResource(R.drawable.rkm);
                break;
          case 16:
            songTitle.setText("HELLO MISTER-NEETI MOHAN\nMovie:Kotigobba 2");
            imageView.setImageResource(R.drawable.hellomm);
                break;
          case 17:
            songTitle.setText("RANJHA-JASLEEN ROYAL\nMovie:SHERSHAAH");
            imageView.setImageResource(R.drawable.rnjh);
                break;
          case 18:
            songTitle.setText("INKEM INKEM-SID SRIRAM\nMovie:Geeta
Govindam");
            imageView.setImageResource(R.drawable.inkmm);
```

```
                break;
            case 19:
              songTitle.setText("JEENA JEENA YA-SHAAN\nMovie:Maanikya");
              imageView.setImageResource(R.drawable.jeenaa);
                break;
            case 20:
              songTitle.setText("UNTIL I FOUND YOU-STEPHAN SANCHEZ, EM
BEIHOLD");
              imageView.setImageResource(R.drawable.uify);
                break;
        }

        // seekbar duration
        mMediaPlayer.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
            @Override
            public void onPrepared(MediaPlayer mp) {
                mSeekBarTime.setMax(mMediaPlayer.getDuration());
                mMediaPlayer.start();
            }
        });
        mSeekBarTime.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress,
boolean fromUser) {
                if (fromUser) {
                    mMediaPlayer.seekTo(progress);
                    mSeekBarTime.setProgress(progress);

                    updateSeekbar();
                }
            }
            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {

            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {

            }
        });

        mMediaPlayer.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {
            @Override
            public void onCompletion(MediaPlayer mp) {
                     playNextSong();
            }
        });


        new Thread(new Runnable() {
            @Override
            public void run() {
                while (mMediaPlayer != null) {
                    try {
                        if (mMediaPlayer.isPlaying()) {
                            Message message = new Message();
                            message.what = mMediaPlayer.getCurrentPosition();
                            handler.sendMessage(message);
                            Thread.sleep(1000);
                        }
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
        }).start();
    }
    private void playNextSong() {
    currentIndex++;
        if (currentIndex >=songs.size()) {
        currentIndex = 0;
            }

            if (mMediaPlayer.isPlaying()) {
            mMediaPlayer.stop();
        }

        mMediaPlayer
MediaPlayer.create(getApplicationContext(),songs.get(currentIndex));
     mMediaPlayer.start();
     songNames();
    }

    @SuppressLint("Handler Leak") Handler handler = new Handler () {
        @Override
        public void handleMessage  (Message msg) {

            mSeekBarTime.setProgress(msg.what);
        }
    };
    @Override
    protected void onDestroy(){
        super.onDestroy();;
        if (mMediaPlayer != null) {
            mMediaPlayer.release();
            mMediaPlayer=null;
        }
    }
}
```

CHAPTER 5

RESULTS AND DISCUSSIONS

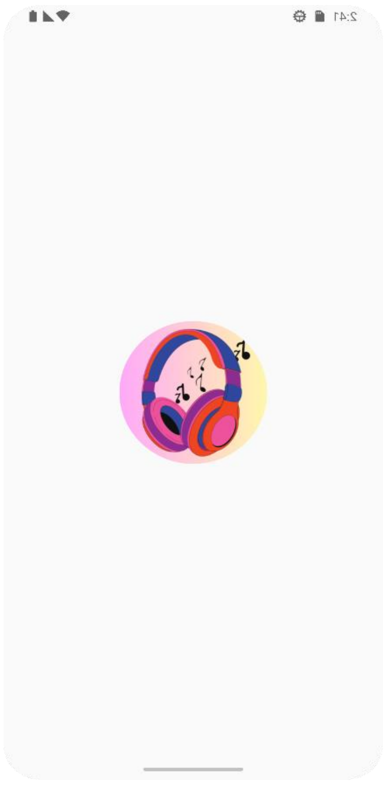
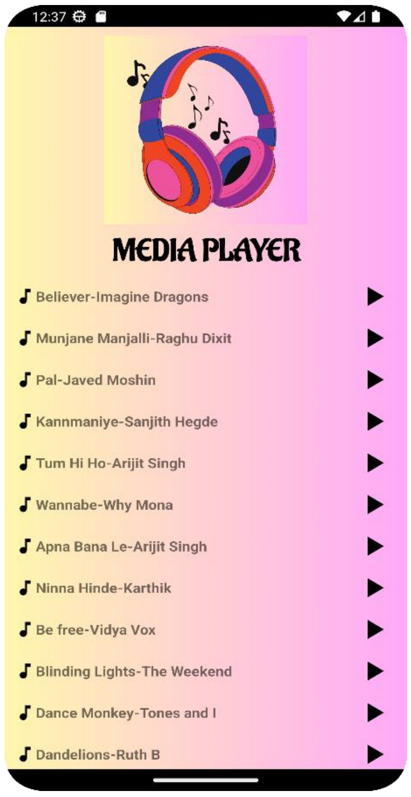


Fig 5.1: Splash Screen



12:37

MEDIA PLAYER

- Believer-Imagine Dragons
- Munjane Manjalli-Raghu Dixit
- Pal-Javed Moshin
- Kannmaniye-Sanjith Hegde
- Tum Hi Ho-Arijit Singh
- Wannabe-Why Mona
- Apna Bana Le-Arijit Singh
- Ninna Hinde-Karthik
- Be free-Vidya Vox
- Blinding Lights-The Weekend
- Dance Monkey-Tones and I
- Dandelions-Ruth B

Fig 5.2: Playlist

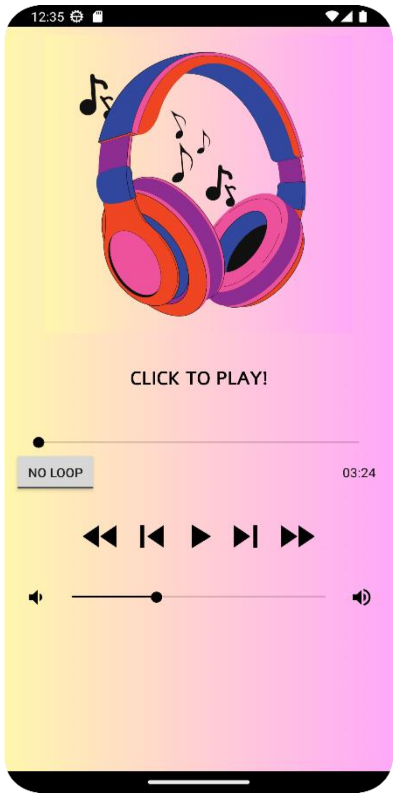
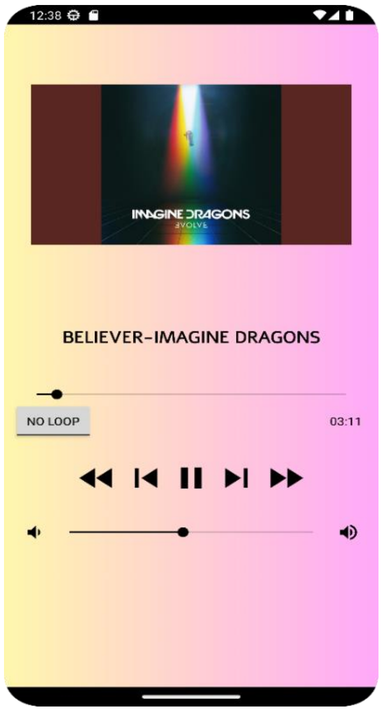
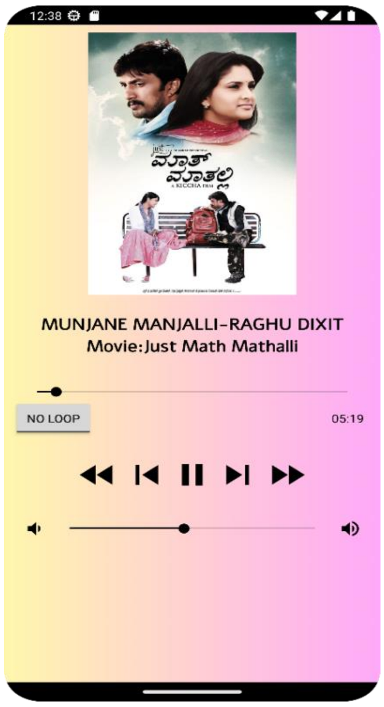


Fig 5.3: Play Music

Fig 5.1 represents the Splash screen of the Media Player Application. It is the initial screen that appears when the Media Player Application is launched. It typically displays the application's logo and provides a brief visual representation while the application is loading. Fig 5.2 represents the Playlist Screen. It typically consists of a list of Songs that showcases the playlist names and associated artwork or icons with additional information of each song. Fig 5.3 is the play music screen is where the user clicks play button to play the song.





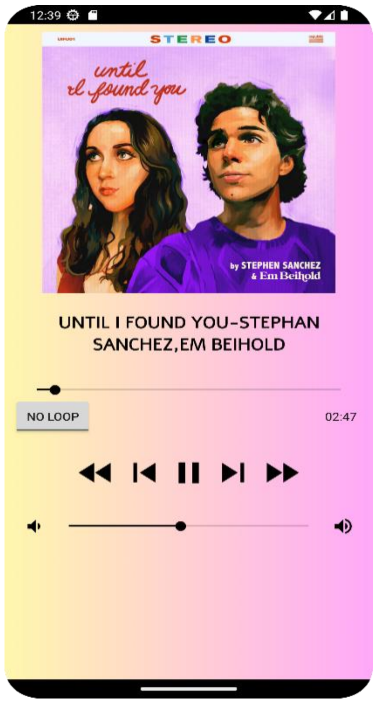
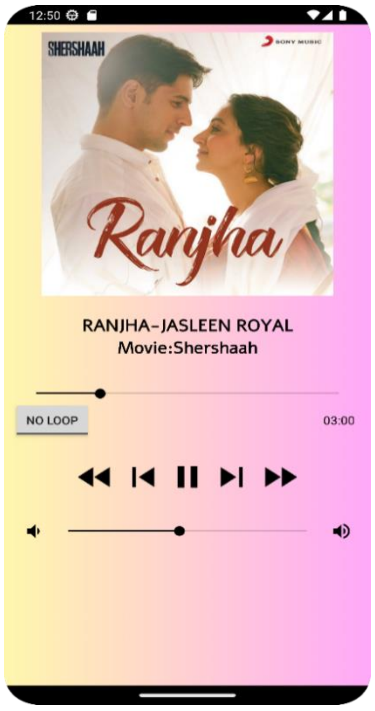
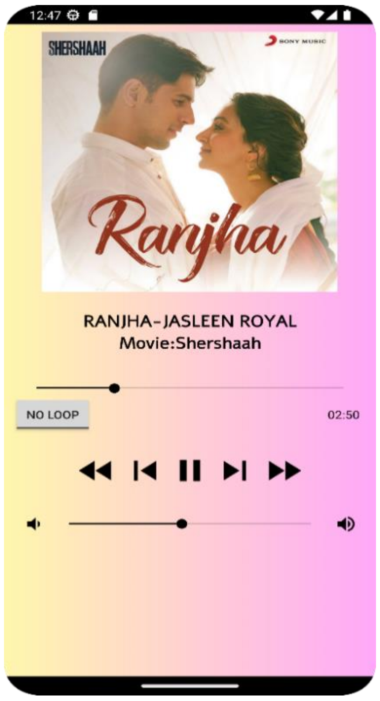


Fig 5.4 First Song **Fig 5.5 Next Song** **Fig 5.6 Previous Song**

The Fig 5.4 illustrates the user interface when the first song in the playlist is being played. Fig 5.5 showcases the interface during the playback of the next song, presenting updated song information and controls. The Fig 5.6 demonstrates the interface while playing the previous song of the First Song, allowing users to easily switch back to the previous track on clicking previous icon button.





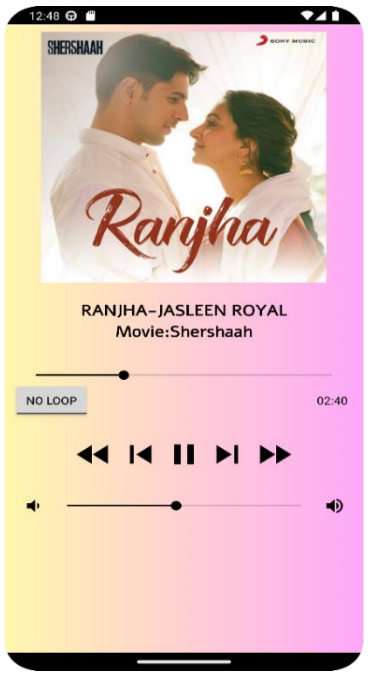


Fig 5.7: Fast Rewind **Fig 5.8: Current Playback** **Fig 5.9: Fast Forward**

The Fig 5.9 depicting the fast forward functionality showcases the user interface when users activate the feature to skip forward in the current song. Similarly, the Fig 5.7 illustrating the fast rewind functionality displays the interface when users activate the option to skip backward in the song. The Fig 5.8 representing the current playback position demonstrates the user interface indicating the current position in the song.

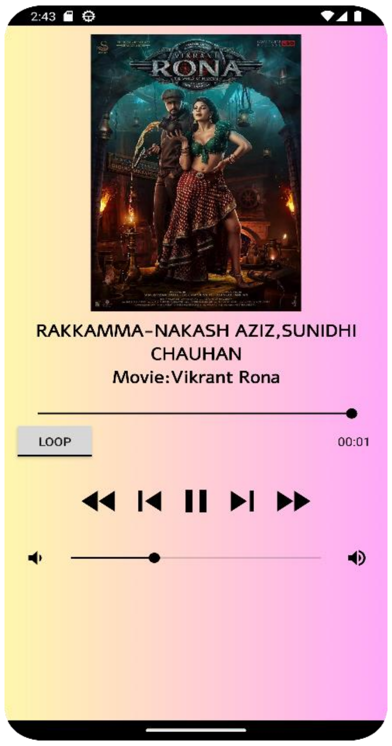


Fig 5.10: Loop Activated

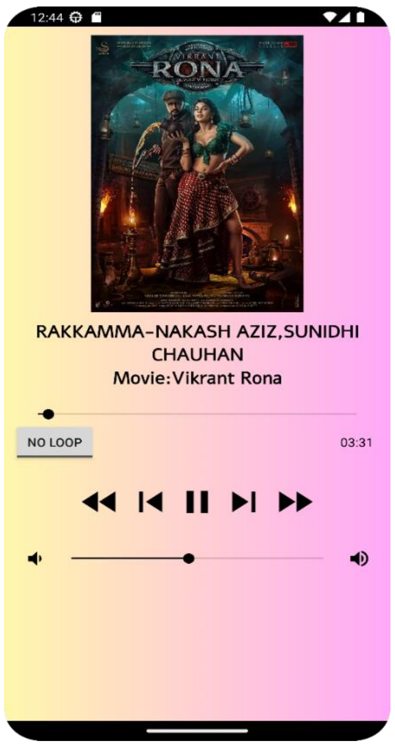


Fig 5.11: Loop Deactivated

The Fig 5.10 showcases the user interface of the Media Player Application when the loop functionality is activated, featuring a button that users can toggle on or off. This enables them to continuously repeat the current song. The Fig 5.11 provides a visual representation of the looped section during song playback. This visual cue informs users that the selected section of the song will repeat indefinitely until the loop mode is deactivated.

CHAPTER 6
CONCLUSION

In conclusion, the Android Media player developed in Java using Android Studio incorporates essential features for playing music. This Application developed using Android Studio aims to provide users with an enhanced and immersive music listening experience on their Android devices. By incorporating features such as volume and seek control, play/pause/fast-forward/fast-rewind functionalities, loop capability, song duration display, song information, and playlist management, the application offers a comprehensive set of tools to customize and enjoy music playback.

Through an intuitive and user-friendly interface, the application enables users to easily navigate their music library, select and play songs, adjust volume levels, and control playback position. The inclusion of looping functionality allows users to repeat their favorite songs or sections, while the display of song details and artwork enhances the visual and contextual experience.

The development of the Media Player Application relies on Android Studio's development environment, utilizing Java programming language, Android SDK, and various libraries and APIs. Adherence to design principles and optimization techniques ensures smooth performance, responsiveness, and a seamless user interface.

In conclusion, the Media Player Application enriches the music listening experience on Android devices, offering a versatile and user-friendly platform for users to enjoy and manage their music collections. By combining all the above features, the media player provides an enjoyable and customizable music listening experience to Android users.