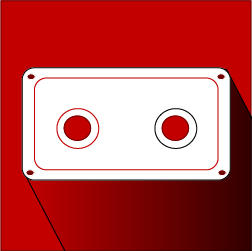
****

**TRAXter**

**TRAXter**

**A Mini-Project Report**

**Under**

**Implementation of Technology**

***Submitted by***

**Darshan Shah**

**Chris Pinto**

**Krish Shetty**

**Kuldeep Sonani**

**NAME OF THE CANDIDATE**

***Under The Guidance Of***

**Prof. Ratnesh Chaturvedi**

***in partial fulfillment for the award of the degree***

***of***

**Bachelor of Technology**

**IN**

**Computer Engineering**

**at**

**Mukesh patel school of technology management and engineering, nmims**

**MONTH & YEAR**

# CERTIFICATE

This is to certify that the project entitled “TRAXter” is the bonafide work carried out by Darshan Shah, Chris Pinto, Krish Shetty and Kuldeep Sonani of B.Tech (Computer Engineering), MPSTME (NMIMS), Mumbai, during the IV semester of the academic year 2014-2015, inpartial fulfillment of the requirements for the award of the Degree of Bachelors of Technology as per the norms prescribed by NMIMS. The mini-project work has been assessed and found to be satisfactory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Ratnesh Chaturvedi

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examiner 1 Examiner 2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dean

Dr. Sharad Mhaiskar

# DECLARATION

We, Darshan Shah, Chris Pinto, Krish Shetty and Kuldeep Sonani, Roll No. E029, E030, E041, E045 respectively, of B.Tech (Computer Engineering), IV semester understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. (Source: IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of the Students:

Name: Darshan Shah, Chris Pinto , Krish Shetty and Kuldeep Sonani.

Roll No. E029, E030, E041, E045 (respectively)

Place:

Date:

# Acknowledgement

In Performing our project, we have taken the help and guidance of some respected people, who deserve our greatest gratitude. The completion of our project gave us much pleasure. We would like to show our gratitude to Prof. Ratnesh Chaturvedi, Course Instructor, MPSTME for providing us with guidelines for our project throughout numerous consultations. We would like to expand our deepest gratitude to all those who have directly or indirectly guided us in completing our app.

We would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

# Table of contents

**CHAPTER NO. TITLE PAGE NO.**

List of Figures i

Abbreviations iii

Abstract iv

1. INTRODUCTION

1.1 Project Overview

1.2 Hardware Specification

1.3 Software Specification

2. REVIEW OF LITERATURE

3. ANALYSIS& DESIGN

4. METHODS IMPLEMENTED

5. RESULTS & DISCUSSION

6. CONCLUSION & FUTURE SCOPE

REFERENCES

APPENDIX

# List of Figures

**CHAPTER NO. TITLE PAGE NO.**

1. INTRODUCTION

No figures.

2. REVIEW OF LITERATURE

No figures.

3. ANALYSIS and DESIGN

Fig3.1-Phase 1 Main Interface

Fig3.2-Phase 1 Tracklist

Fig3.3-Phase 1Notification Bar

Fig3.4-Phase 1Application Information

Fig3.5-Phase 2 Main Interface

Fig3.6-Lock Screen Widget

Fig3.7-Main Screen Widget

Fig3.8-Genre 1 Tracklist

Fig3.9-Genre 2 Tracklist

4. METHOD OF IMPLEMENTATION

Fig4.1-Show the Android project view

Fig4.2- Project Build Files.

Fig4.3- Android Studio project structure

\*\*\*\*\*\*\*\*\*\*

***NOTE:***

* *Figures must be properly explained in the text*
* *Figure No. and caption of the figure must be below the figure in all the chapters*

# Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| ADB | Android Debugging Bridge |
| ADT | Android Development Tools |
| Android SDK | Android Software Development Kit |
| ART | Android Runtime |
| CPU | Central Processing Unit |
| CR | Class Representative |
| GPU | Graphic Processing Unit |
| HoD | Head of Department |
| IPS | In-Plane Switching |
| LCD | Liquid Crystal Display |
| RAM | Random AccessMemory |
| SQL | Structured Query Language |
| SR | Student Representative |
| XML | Extensible Markup Language |

# Introduction

TRAXter is a music player for media files that supports various music formats.

Current TRAXter features include:

* Play various format music files from your Library.
* Display information on the songs through meta-data reading.
* Display songs in an organized view.
* Create static playlists by dragging and dropping from the Library view.
* Create automatic playlists from criterias.
* Search for songs in the sources list like Library or Playlists.

## Project Overview

The Android operating system has grown immensely , it has taken the spot of the world’s most popular OS, since it can be tailored to suit various individual needs. As Android continues to grow beyond smartphones, it will become the brains behind invisible, ubiquitous computing.

A simple audio player with basic controls like play, pause, forward, backward, next, previous, playlist and seekbar has already been introduced since the inception of Android.

So then what is ‘TRAXter’ all about? What was the thought behind the conception of the idea?

TRAXter targets the traveler in each individual. Nothing says India like its Public Transport and its Pedestrians. So what we have tried to develop is an app that can/will be used by the average Indian passenger. With its simple design and easy to use interface we are hopeful of a large user base.

## Hardware Specification

TRAXter is not a hardware intensive application, thus requires very basic level of hardware, as its application ranges across all devices, low end to high end, phones to tablets. Thus, the following are the minimum hardware specifications required to run the application:

* Processor:
  + 1 GHz ARM Cortex A8 based CPU core
  + PowerVR SGX 540 GPU.
* Memory:
  + 512 MB of RAM
  + 1 GB of Flash Memory
  + Micro-SD card slot (Optional)
* Screen:
  + 3.5-inch LCD display
  + Capacitive or Resistive touch

During Development of the application, OnePlus One was used to develop the application on mobiles and ASUS Google Nexus 7 was used to develop the application on a tablet. Following are the specifications:

* OnePlus One:
  + Processor:
    - 2.5 GHzQualcomm Snapdragon 801
    - Adreno 330 GPU
  + Memory:
    - 3GB of RAM
    - 64GB of Flash Memory
  + Screen:
    - 5.5-inch IPS LCD display
* ASUS Google Nexus 7:
  + Processor:
    - 1.2 GHz NVIDIA Tegra 3 CPU
    - ULP GeForce GPU
  + Memory:
    - 1 GB of RAM
    - 16GB of Flash Memory
  + Screen:
    - 7-inch IPS LCD display

## Software Specifications:

TRAXter has been developed on Android SDK API 22, which is primarily for android 5.0 Lollipop. This also has backward compatibility to previous APIs. TRAXter is designed to work on Android 4.0, Ice Cream Sandwich and above.

Apart from Android 4.0 and above, the application, like most android applications can run on the following Operating Systems:

* Blackberry OS 11
* Sailfish OS
* Chrome OS
* Color OS
* INUI OS
* YUN OS
* Nokia X mobile Platform

Some of these are based on android, while some (Bbos 11, Sailfish OS) are made compatible to run android applications.

Following is the list of Android versions on which TRAXter has been tested:

* Android 4.0 (Ice Cream Sandwich)
* Android 4.2 (Jellybean)
* Android 4.3 (Jellybean)
* Android 4.4 (KitKat)
* Android 5.0 (Lollipop)
* Android 5.0.2 (Lollipop)
* Android 5.1 (Lollipop)

This gives the application a broad platform, as these operating systems are in majority of smart phones being used by prospective users.

# Review of Literature

When analyzing the possibilities of the application, following points came to light for the usability of TRAXter:

On long as well as short distance railway routes passengers often take to music as their pass time or just to add to the thrill of the journey. Often falling asleep as they do so, failing to wake up from the musical trance.

Falling asleep with music on, leads to battery drainage, curtailing the duration of passive playback of music would lead to energy conservation.

While walking freedom to raise the volume of head phones to the maximum proves hazardous , hence limiting that maximum turnout seems necessary.

# Analysis and Design

Objectives :

* To limit volume turnout for walking/ride mode
* To integrate the sleep alarm & timer
* Voice Control (Future Scope)

During the conception of the application, the requirements were mapped out first. This included analyzing the need of this application, its features and figuring out a list of functions it would perform during operation. This largely consisted of the base of the analysis.

After performing a complete analysis, the design of the application was pursued. TRAXter is designed using Material Design elements. Material Design was introduced very recently with the launch of Android 5.0 (Lollipop).

## Features:

* Music sleep timer and alarm
* Limited voice in walking/ride mode
* Voice Control (Future Scope)

Implementation (Interfaces)

Implementation is the realization, application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other system. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming.

The undertaking began with two phases the alpha and the beta phase:

Phase Alpha:

The alpha phase started with the basic functionality implementation. This phase dealt with the basic playback of music on the android phone. Beginning from scratch using data and open source codes from the internet we devised and developed a basic music player.

This phase of the app saw implementation of segregation of music on basis of varied criteria such as artist, album, genre and song names per say.

The Alpha phase app had various added features such as additional details on long press of the album art.

Main Interface:

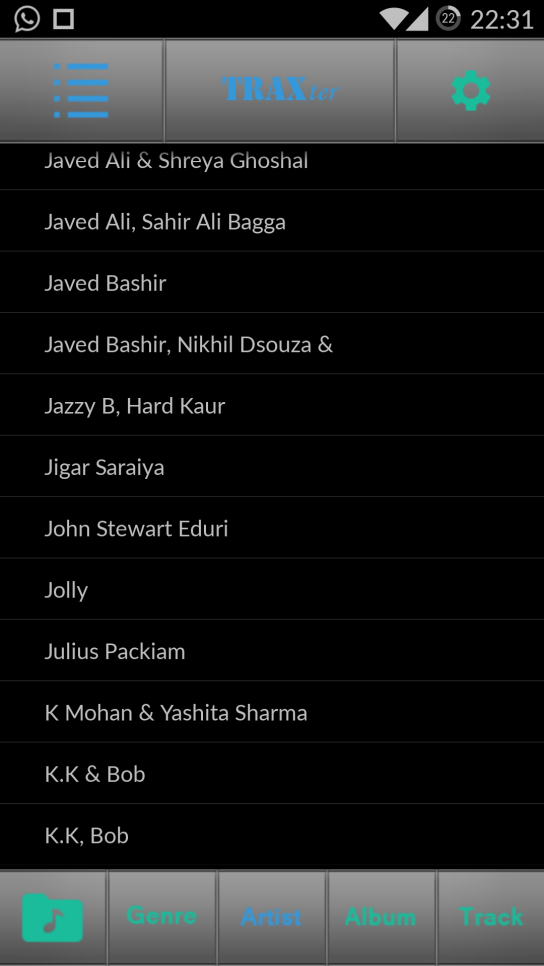


**Figure 3.1** Main Interface

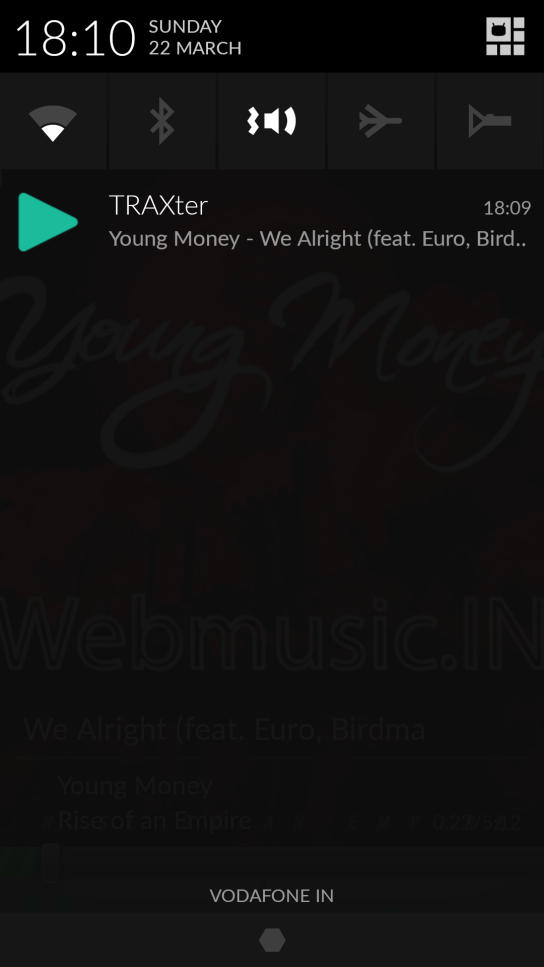
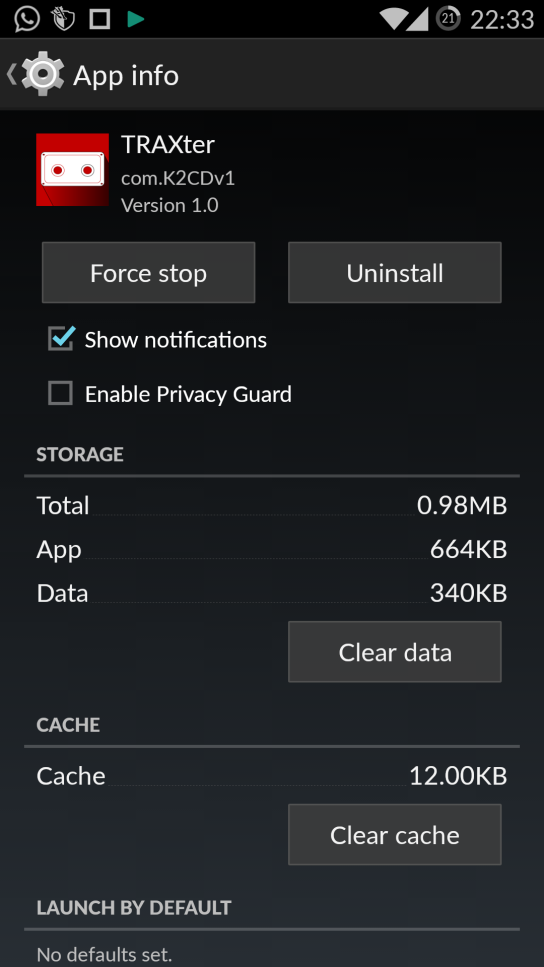
Buttons:

1. Pause
2. Next Song
3. Previous Song
4. Shuffle
5. Menu Tab
6. Settings

Tracklist:



**Figure 3.2** Tracklist.

**Figure 3.3** Notification Bar **Figure 3.4**  Application information

The Alpha phase app is well equipped with all the basic features of a music player. This was an important and time consuming phase of the application development.

The Second phase was the beta phase of the app development:

This phase targets the online streaming of music. This phase marked the change in design for our app. The Material design format for the app was implemented in this phase of the app development.

Using Android Studio Fragments this app took shape :

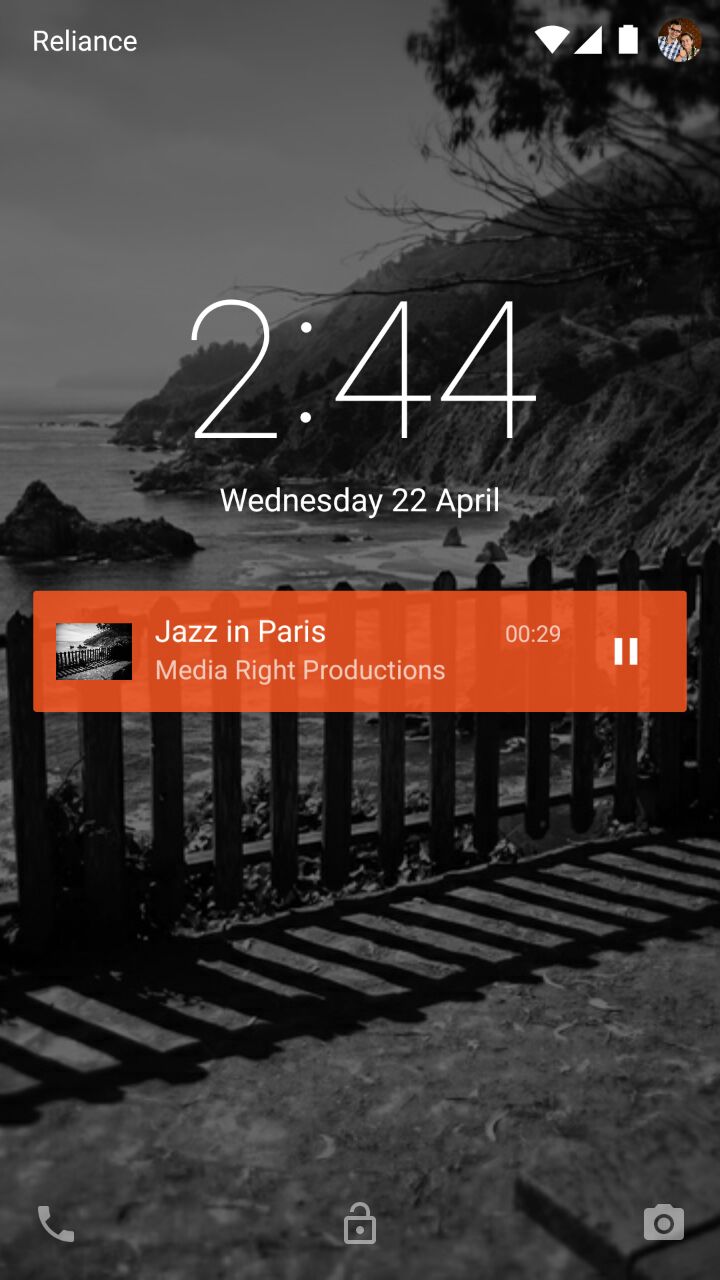
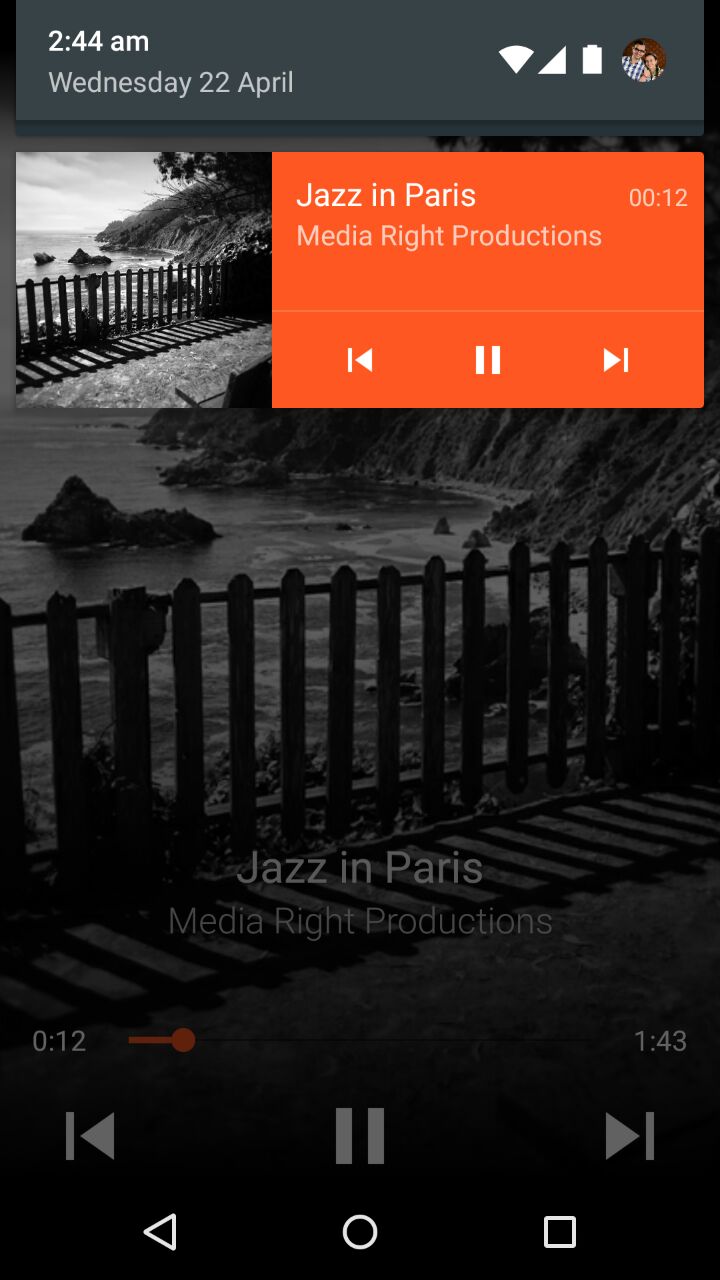
Beta Phase Interface:



**Figure 3.5** Player Interface

Buttons:

1. Play/pause
2. Next Song
3. Previous Song
4. Back

**Figure 3.6** Lock Screen Widget **Figure 3.7** Main Screen Widget

The beta app encompasses easy access capabilities with widgets such as the lock screen widgets and the main screen widget.

The lock screen widget buttons include:

1. Play/Pause

The Main Screen Widget buttons include:

1. Play/Pause
2. Next Song
3. Previous Song

# IMG_7490.JPG IMG_7487.JPG

**Figure 3.8**  Genre tracklist 1 **Figure 3.9**  Genre tracklist 2

The beta app also implements segregation on the basis of genre as of now .

# Methods Implemented

Android Studio:

Android Studio is the official IDE for Android application development, based on[IntelliJ IDEA](https://www.jetbrains.com/idea/). On top of the capabilities you expect from IntelliJ, Android Studio offers:

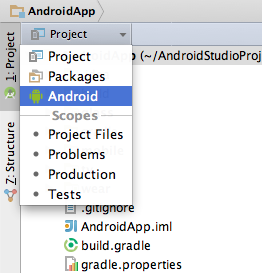
1. Flexible Gradle-based build system
2. Build variants and multiple apk file generation
3. Code templates to help you build common app features
4. Rich layout editor with support for drag and drop theme editing
5. lint tools to catch performance, usability, version compatibility, and other problems
6. ProGuard and app-signing capabilities
7. Built-in support for [Google Cloud Platform](http://developers.google.com/cloud/devtools/android_studio_templates/), making it easy to integrate Google Cloud Messaging and App Engine

Project and File Structure:

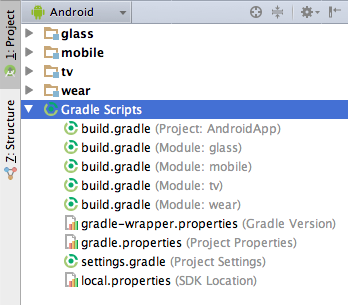
Android Project View:

By default, Android Studio displays your profile files in the Android project view. This view shows a flattened version of your project's structure that provides quick access to the key source files of Android projects and helps you work with the [Gradle-based build system](http://developer.android.com/sdk/installing/studio-build.html). The Android project view:

* Groups the build files for all modules at the top level of the project hierarchy.
* Shows the most important source directories at the top level of the module hierarchy.
* Groups all the manifest files for each module.
* Shows resource files from all Gradle source sets.
* Groups resource files for different locales, orientations, and screen types in a single group per resource type.



**Figure 4.1** Show the Android project view.



**Figure 4.2** Project Build Files.

The Android project view shows all the build files at the top level of the project hierarchy under Gradle Scripts. Each project module appears as a folder at the top level of the project hierarchy and contains these three elements at the top level:

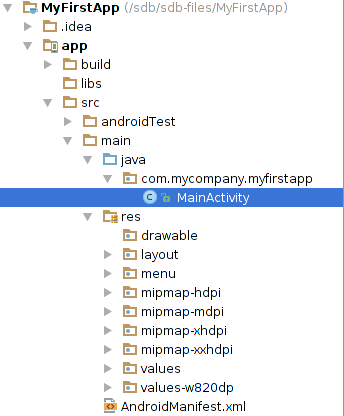
* java/ - Source files for the module.
* manifests/ - Manifest files for the module.
* res/ - Resource files for the module.

For example, Android project view groups all the instances of the ic\_launcher.png resource for different screen densities under the same element.

**Note:** The project structure on disk differs from this flattened representation. To switch to back to the segregated project view, select Project from the Project drop-down.

Android Studio Project and Directory Structure

When you use the Project view of a new project in Android Studio, you should notice that the project structure appears different than you may be used to in Eclipse. Each instance of Android Studio contains a project with one or more application modules. Each application module folder contains the complete source sets for modules including src/mainand src/androidTest directories, resources, build file and the Android manifest. For the most part, you will need to modify the files under each module's src/main directory for source code updates, the gradle.build file for build specification and the files under src/androidTest directory for test case creation.



**Figure 4.3**  Android Studio project structure

Creating new files:

You can quickly add new code and resource files by clicking the appropriate directory in the Project pane and pressing ALT + INSERT on Windows and Linux or COMMAND + N on Mac. Based on the type of directory selected, Android Studio offers to create the appropriate file type.

For example, if you select a layout directory, press ALT + INSERT on Windows, and select Layout resource file, a dialog opens so you can name the file (you can exclude the .xml suffix) and choose a root view element. The editor then switches to the layout design editor so you can begin designing your layout.

Android Build System :

The Android build system is the toolkit you use to build, test, run and package your apps. The build system can run as an integrated tool from the Android Studio menu and independently from the command line. You can use the features of the build system to:

* Customize, configure, and extend the build process.
* Create multiple APKs for your app with different features using the same project and modules.
* Reuse code and resources across source sets.

The flexibility of the Android build system enables you to achieve all of this without modifying your app's core source files. To build an Android Studio project, see [Building and Running from Android Studio](http://developer.android.com/tools/building/building-studio.html). To configure custom build settings in an Android Studio project, see [Configuring Gradle Builds](http://developer.android.com/tools/building/configuring-gradle.html)

Android Testing Tools:

Testing is a critical software development activity because it helps you improve the quality of your apps, ensure better user satisfaction, and reduce overall development time spent on fixing defects.The following sections describe tools that help you test your mobile apps for the Android platform :

* [Android Testing Support Library](http://developer.android.com/tools/testing-support-library/index.html)

This library provides a set of APIs that allow you to quickly build and run test code for your apps, including JUnit 4 and functional user interface (UI) tests. The Android Testing Support Library includes the following test automation tools:

* [AndroidJUnitRunner](http://developer.android.com/tools/testing-support-library/index.html#AndroidJUnitRunner): JUnit 4-compatible test runner for Android
* [Espresso](http://developer.android.com/tools/testing-support-library/index.html#Espresso): UI testing framework; suitable for functional UI testing within an app
* [UI Automator](http://developer.android.com/tools/testing-support-library/index.html#UIAutomator): UI testing framework; suitable for cross-app functional UI testing across system and installed apps
* [Monkey](http://developer.android.com/tools/help/monkey.html)

This tool runs on your emulator or device and generates pseudo-random streams of user events such as clicks, touches, or gestures, as well as a number of system-level events. You can use the Monkey tool to stress-test applications that you are developing, in a random yet repeatable manner.

* [monkeyrunner](http://developer.android.com/tools/help/monkeyrunner_concepts.html)

This testing system provides an API for writing programs that control an Android device or emulator from outside of Android code

Installation

Installation (or setup) of a program (including drivers) is the act and the effect of putting the program in a system so that it can be executed. Most programs are supplied in a condensed form intended for sale and distribution. In order to be used, they must be 'unpacked' and the relevant information placed correctly on the computer, taking account of variations between devices, and any customized settings required by the user. During installation, various tests are made of system suitability, and the device is configured to store the relevant files and any necessary settings required for that program to operate correctly. One file installation (Traxter.apk), an executable file that’s synonymous for installation can be made for other applications as well.

APK:

Android application package (APK) is the [package](http://en.wikipedia.org/wiki/Package_format) [file format](http://en.wikipedia.org/wiki/File_format) used to distribute and install [application software](http://en.wikipedia.org/wiki/Application_software) and [middleware](http://en.wikipedia.org/wiki/Middleware) onto [Google](http://en.wikipedia.org/wiki/Google)'s [Android](http://en.wikipedia.org/wiki/Android_(operating_system)) operating system. Certain other operating systems and devices, such as BlackBerry devices with the operating system version 10 or higher, also support APK packages.

APK files are analogous to other [software packages](http://en.wikipedia.org/wiki/Software_package_(disambiguation)) such as [MSI packages](http://en.wikipedia.org/wiki/Windows_Installer) in [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) or  [Deb packages](http://en.wikipedia.org/wiki/Deb_(file_format)) in [Debian](http://en.wikipedia.org/wiki/Debian)-based operating systems like Ubuntu. To make an APK file, a program for Android is first compiled, and then all of its parts are packaged into one file. An APK file contains all of that program's code (such as [.dex](http://en.wikipedia.org/wiki/.dex) files), resources, assets, certificates, and [manifest file](http://en.wikipedia.org/wiki/Manifest_file). As is the case with many file formats, APK files can have any name needed, provided that the file name ends in ".apk".

APK files are a type of [archive file](http://en.wikipedia.org/wiki/Archive_file), specifically in [zip format](http://en.wikipedia.org/wiki/Zip_(file_format)) packages based on the [JAR file format](http://en.wikipedia.org/wiki/JAR_(file_format)), with .apk as the [filename extension](http://en.wikipedia.org/wiki/Filename_extension). The [MIME type](http://en.wikipedia.org/wiki/Internet_media_type) associated with APK files is application /vnd.android.package-archive.

An APK file is an [archive](http://en.wikipedia.org/wiki/Archive_file) that usually contains the following directories:

* META-INF directory:
* MANIFEST.MF: the [Manifest file](http://en.wikipedia.org/wiki/Manifest_file)
* CERT.RSA: The certificate of the application.
* CERT.SF: The list of resources and [SHA-1](http://en.wikipedia.org/wiki/SHA-1) digest of the corresponding lines in the MANIFEST.MF file;
* lib: the directory containing the compiled code that is specific to a software layer of a processor, the directory is split into more directories within it:
* armeabi: compiled code for all [ARM](http://en.wikipedia.org/wiki/ARM_architecture) based processors only
* armeabi-v7a: compiled code for all ARMv7 and above based processors only
* x86: compiled code for [x86](http://en.wikipedia.org/wiki/X86) processors only
* mips: compiled code for [MIPS](http://en.wikipedia.org/wiki/MIPS_architecture) processors only
* res: the directory containing resources not compiled into resources.arsc (see below).
* assets: a directory containing applications assets, which can be retrieved by AssetManager.
* AndroidManifest.xml: An additional Android manifest file, describing the name, version, access rights, referenced library files for the application. This file may be in Android [binary XML](http://en.wikipedia.org/wiki/Binary_XML) that can be converted into human-readable plaintext XML with tools such as [AXMLPrinter2](http://code.google.com/p/android4me/downloads/list), [android-apktool](http://code.google.com/p/android-apktool/), or [Androguard](http://code.google.com/p/androguard/wiki/Usage#Androaxml).
* classes.dex: The classes compiled in the [dex file format](http://en.wikipedia.org/wiki/DEX_(file_format)) understandable by the [Dalvik virtual machine](http://en.wikipedia.org/wiki/Dalvik_(software))
* resources.arsc: a file containing precompiled resources, such as binary XML for example.

**Results And Discussion**

Testing

Software Testing is the process used to help identify the correctness, completeness,security, and quality of developed computer software. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a 'criticism' or comparison

that compares the state and behaviour of the product against a specification. There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following routine procedure. One definition of testing is "the process of questioning a product in order to evaluate it", where the "questions" are operations the tester attempts to execute with the product, and the product answers with its behavior in reaction to the probing of the tester.

Although most of the intellectual processes of testing are nearly identical to that of review or inspection, the word testing is connoted to mean the dynamic analysis of the product—putting the product through its paces. Some of the common quality attributes include capability, reliability, efficiency, portability, maintainability, compatibility, and usability. A good test is sometimes described as one which reveals an error; however,more recent thinking suggests that a good test is one which reveals information of interest to someone who matters within the project community. For testing our project we followed Black Box technique.

Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The testdesigner selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.

Test Results

\*\* test were performed for various files with different sampling rate, different bitrate, different file size. And results were satisfying. Quality of sound output was also acceptable.

\*\*\* In testing its found that the project is incomplete with respect to specified objectives

**Conclusion and Future Scope**

We accepted an impossible task and never gave up during it’s conception. A lot of hurdles were successfully corrected, but some were not. The work we’ve put in has been valuable and we’ve come very close to making a great and useful music player.

This is a work in progress with a lot of potential and scope for the everyday Indian.

We will work our very best to accomplish the voice recognition, which is even posing as a challenge for the google team.

The alarm feature for the app is a few months away from reality.

The inactivity feature is very unstable and caused the app to crash, the feature however is already available.

Criticisms:

1. App instability for certain features.
2. Incompletion as per said deadline.

**REFERENCES:**

1. Android Developers (<https://developer.android.com/training>)
2. Tutorials (<http://www.tutorialspoint.com/android/android_mediaplayer.htm>)
3. Turtle Player Open Source Code (<http://www.turtle-player.co.uk/>)