COLLEGE OF ENGINEERING AND RURAL TECHNOLOGY, MEERUT

“ **MUZE APP”**



**SUBMITTED BY-**

**NAME:**

**ROLL NO:**

**NAME:**

**ROLL NO:**

**SESSION:**

**UNDER THE GUIDANCE OF**

**Mr. Varun Kaushik**

**(HOD, CS Dept.)**

**SUBMITTED TO**

**Dr. A.P.J.ABDUL KALAM TECHNICAL UNIVERSITY,LUCKNOW**

# COLLEGE OF ENGINEERING & RURAL TECHNOLOGY

**“Muze App”**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS**

**FOR THE AWARD OF THE DEGREE OF**

**BACHELOR OF TECHNOLOGY (CS)**



**SESSION:**

**SUBMITTED BY-**

**NAME:**

**NAME:**

**ROLL NO:**

**TABLE OF CONTENTS**

**DECLARATION ………………………………………………………………………5**

**CERTIFICATES ………………………………………………………………………6**

**ACKNOWLEDGEMENT ………………………………………………………….......7**

**ABSTRACT ..…………………………………………………………………………….8**

**LIST OF TABLES FIGURES …………………………………………………………..9**

**CHAPTER 1 INTRODUCTION ………………………………………………………10**

**1.1 OBJECTIVE ………………………………………………………………..11 1.2 CURRENT SYSTEM ………………………………………………………12 1.3 PROPOSED SYSTEM ……………………………………………………..13**

**CHAPTER 2 FEASIBILITY STUDY ………………………………………………..14**

**2.1 ECONOMICAL FEASIBILITY …………………………………………...15**

**2.2 TECHNICAL FEASIBILITY ……………………………………………..16**

**2.3 OPERATIONAL FEASIBILITY …………………………………………..17**

**2.4 BEHAVIORAL FEASIBILITY …………………………………………...17**

**CHAPTER 3 PROJECT PLANNING ……………………………………………….18**

**3.1 PROJECT GOAL …………………………………………………………..18**

**3.2 PROJECT DELIVERABLE ………………………………………………..18 3.3 SUPPORTING PLANS …………………………………………………….18-19**

**CHAPTER 4 REQUIREMENT ANALYSIS ………………………………………..20**

**CHAPTER 5 DEVELOPMENT PLATFORM REQUIREMENTS ………………21**

**5.1 SOFTWARE REQUIREMENTS ………………………………………….21 5.2 HARDWARE REQUIREMENTS …………………………………………21**

**CHAPTER 6 SYSTEM DESIGN ……………………………………………………...22**

**6.1 BLOCK DIAGRAM …………………………………………………………22**

**6.2 USE-CASE DIAGRAM ……………………………………………………...23**

**CHAPTER 7 DETAILED DESIGN ………………………………………………….24**

**7.1 DATA FLOW DIAGRAM ………………………………………………...24-29 7.2 E-R DIAGRAM …………………………………………………………….30-31**

**CHAPTER 8 COADING ……………………………………………………………….32-45**

**CHAPTER 9 TESTING ………………………………………………………………..46-54**

**CHAPTER 10 SNAPSHOT ……………………………………………………………..55-63**

**CHAPTER 11 MAINTAINANCE ………………………………………………………64**

**CHAPTER 12 CONCLUSION ………………………………………………………….65**

**CHAPTER 13 FUTURE SCOPE ……………………………………………………….66**

**CHAPTER 14 DEPENDENCIES AND LIMITATIONS ……………………………...67**

**CHAPTER 15 APPENDIXES …………………………………………………………..68-71**

**APPENDIX – A …………………………………………………………………….68**

**APPENDIX – B …………………………………………………………………….69**

**APPENDIX – C …………………………………………………………………….70-71**

**BIBLOGRAPHY …………………………………………………………………………....72**

**REFERENCES ……………………………………………………………………………..73**

**DECLARATION**

We hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been the text.

NAME:

ROLL NO:

Date

NAME

ROLL NO:

Date

**CERTIFICATE**

This is to certify that this Project Report entitled **“MUZE”** which is submitted by XXXXXXXXX having Roll No**,xxxxxxxxxxxxxx XXXXXXX**having Roll NO **xxxxxxxx,** in the partial fulfillment, for the award of degree of **Bachelor of Technology** in Department of Computer Science & Engineering, of **COLLEGE OF ENGINEERING & RURAL TECHNOLOGY, MEERUT**, affiliated to **A.P.J ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW**; is carried out by him/her under my supervision.

The matter embodied in this Project Work has not been submitted earlier for award of any degree or diploma in any university/institution to the best of our knowledge and belief.

**Project Guide Head Of Department**

(Mr.Varun Kaushik) (Mr. Varun Kaushik )

**ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the Project Work, undertaken during B. Tech. Final Year. We owe special debt of gratitude to **Mr. Rizwan (Lecturer)**, Department of Computer Science & Engineering, College Of Engineering & Rural Technology, Meerut for her constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of **Mr. Varun Kaushik (H.O.D)**, Department of Computer Science & Engineering, College Of Engineering & Rural Technology, Meerut for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

)

**ABSTRACT**

In order to solve the problem of complex functions and large required memory of mobile phone music player on the current market, a new music player of simple, convenient, less required memory as well as user-friendly is developed. Based on the Android technology, using the Java language and Eclipse programming tools lead to design and coding of music player. The new design mainly realizes six core functions including main play interface, playlists, menus, play settings, file browsing and song search. This player has merits of high performance, simple operation, and run independently on the Android mobile devices. At the same time, the player can also browse and access files in mobile phones.

Music Player is a windows and android application that guide the user easily to find all the music and play all the music in phone. It can play the songs by user control such as playing next song, previous song from the list. It also facilitates the fast forwarding and fast back warding off song. MUZE is an advance music player which provides not only the features of a basic music player but also provide enhanced features.

In an in built Music player application of android phones, there is no option to download

songs from internet through the application but our proposed music player application

facilitate downloading songs from internet via application. It provides reliability.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Figure Name** | **Page No.** |
| 1 | BLOCK DIAGRAM | 22 |
| 2 | USE CASE DIAGRAM | 23 |
| 3 | DATA FLOW DIAGRAM | 24-29 |
| 4 | E-R DIAGRAM | 30-31 |
| 5 | SNAPSHOTS | 55-63 |

**CHAPTER: 1**

**INTRODUCTION**

**“In contemporary business and science a project refers to a collaborative enterprise, involving research or design that is carefully planned to achieve a particular aim.”**

Music Player is a windows and android application that guide the user easily to find all the music and play all the music in phone. It can play the songs by user control such as playing next song, previous song from the list. It also facilitates the fast forwarding and fast back warding off song. MUZE is an advance music player which provides not only the features of a basic music player but also provide enhanced features.

In an in built Music player application of android phones, there is no option to download songs from internet through the application but our proposed music player application facilitate downloading songs from internet via application. It provides reliability. The layout and user interface of the application proposed by us will be much better than the current application. Users can interact with our application with excellent experience. Our application will also give the facility of changing the songs from playlist by shaking the handset. User can enjoy the song with the lyrics too. She/he can share the songs via Bluetooth through the application. Songs or playlist of songs can be posted on facebook or twitter via our proposed application “MUZE”. User can cut a specific clip of his/hers desired song and can play as a separate clip.

Browsing, playing songs, recording etc operations can all be completed easily by flicking with a thumb. User can enjoy the music of his/her desire with lots of fun.

**1.1 Objective of the project**

The main objective of this android application is to raise up the functionality of existing system. The Current Music player has very few features with less interactive interface and limited functionality. Purposed System will have a better and interactive interface for users. Proposed application provides advance features like sharing songs via Bluetooth to others, she/he can cut & save a specific part of song etc. It has customized background, user can change the current song of playlist by shaking the phone and so on.

In current application user can not post the songs at facebook and twitter but our proposed application “MUZE” will give the functionality of posting songs of user’s desire at facebook and twitter. User can enjoy hers/his favorite songs with the lyrics via our proposed application. She/he can cut a specific part of a particular song and can play it as a separate clip. User can change the background of the application according to hers/his choice.

In current application the browsing option is not available, so our aim is to provide a better and fast browsing facility in our proposed application.

The main objective of our proposed application is to provide an attractive and interactive application to user by which user can enjoy the world of music with much better performance and reliability than the current application.

**1.2 THE CURRENT SYSTEM**

The current system is a basic music player which allows users to play desired songs. This application also gives view of songs in three different manners which are:

* As a list of artists of the songs.
* As a list of genres of the songs.
* As a list of albums of songs.
* As a list of the names of songs.

The current music player also allows making multiple lists of songs according to user’s requirements which is called playlist. Current application also provides shuffling of songs that is playing random song from list. In this application user can not post hers/his songs on facebook and twitter. Interface of the current application is not much attractive and reliable.

**LIMITATIONS OF CURRENT SYSTEM :**

* We can’t share the songs via Bluetooth in over music player.
* Does not provide cutting the specific part of the song.
* Does not have a better search option for finding the required song.
* Does not have the option of changing the background of the application.
* Does not have attractive interface.
* The current application is less interactive.
* User can not list of her/his favorite songs on facebook and twitter.
* User can not download songs from internet through the current application.

**1.3 THE PROPOSED SYSTEM**

**“MUZE”** is an android application that uses certain programs to facilitate technical function of the music player. With this application user can play desired songs .The main features includes Bluetooth sharing , music scissors , shake and play , attractive interface and so on .This application helps you to enjoy the music by easy and fast accessing . This application will also play two or more than two songs as a single track. This application makes the background beautiful by providing multiple background images as per the requirements of user.

**FEATURES OF PROPOSED SYSTEM :** Following are the features of proposed application which make the application much better than the current application and make it more reliable and attractive.

* **Bluetooth Sharing: -** Application provides shearing of music app via Bluetooth efficiently.
* **Shake and Play:-**User can change the present song of playlist by shaking the phone.
* **Network Information:**-This application provides the information about the song using internet connection.
* **Better Browsing Option**:-This application gives fast browsing for finding songs.

**CHAPTER: 2**

**FEASIBILITY STUDY**

To design Android mobile phone music player as long as a computer has the Android development and the application development of Android is free. In addition, mobile phone music player is basic needs for public. The information that which functions are necessary form all the consumers , which functions are needed for some people, and which features are seldom to use is easy to understand. And a lot of research is eliminated, thus saved the spending. Therefore, the whole process of development doesn’t need to spend any money that is economic feasibility.

The key consideration in feasibility analysis is:

**2.1. Economical Feasibility:**

**2.2. Technical Feasibility:**

**2.3. Operational Feasibility:**

**2.4. Behavioral Feasibility:**

**2.1. Economical Feasibility :**

It looks at the financial aspects of the project. It determines whether the management has enough resources and budget to invest in the proposed system and the estimated time for the recovery of cost incurred. It also determines whether it is worthwhile to invest the money in the proposed project. Economic feasibility is determines by the means of cost benefit analysis. The proposed system is economically feasible because the cost involved in purchasing the hardware and the software are within approachable. The personal cost like salaries of employees hired are also nominal, because working in this system need not required a highly qualified professional. The operating-environment costs are marginal. The less time involved also helped in its economical feasibility. It was observed that the organization has already using computers for other purpose, so that there is no additional cost to be incurred for adding this system to its computers.

The backend required for storing other details is also the same database that is Sql. The computers in the organization are highly sophisticated and don’t needs extra components to load the software. Hence the organization can implement the new system without any additional expenditure. Hence, it is economically feasible.

* **Software Cost :**

|  |  |
| --- | --- |
| SDK Toolkit | Free |
| Android Studio | Free |

* **Manpower Cost :**

|  |  |
| --- | --- |
| Team Cost | **40,000/-** |

**Total Cost : 40,000**/**-**

**2.2. Technical Feasibility :**

It is a measure of the practically of a specific technical solution and the availability of technical resources and expertise

* The proposed system uses xml and java as front-end and SQLight as back-end tool.

SQLight is a popular tool used to design and develop database objects such as table views, indexes.

* The above tools are readily available, easy to work with and widely used for developing commercial application.

Hardware used in this project are- p4 processor 2.4GHz, 128 MB RAM, 40 GB hard disk, floppy drive. These hardware were already available on the existing computer system. The software like MS-Access 2003, Web logic Server, Thin Driver, visual studio and .net framework and operating system WINDOWS-XP’ used were already installed On the existing computer system. So no additional hardware and software were required to purchase and it is technically feasible. The technical feasibility is in employing computers to the organization. The organization is equipped with enough computers so that it is easier for updating. Hence the organization has not technical difficulty in adding this system.

**Tools Used :**

1) Android SDK

2) Android Studio

3) API 17

**Duration of Project:-**

|  |  |
| --- | --- |
| For Study | 15 days |
| For Designing | 12 days |
| For Development | 40 days |

**Total time 67 days**

**2.3. Operational Feasibility :**

The system will be used if it is developed well then be resistance for users that undetermined

* No major training and new skills are required as it is based on DBMS model.
* It will help in the time saving and fast processing and dispersal of user request and applications.
* New product will provide all the benefits of present system with better performance.
* Improved information, better management and collection of the reports.
* User support.
* User involvement in the building of present system is sought to keep in mind the user specific requirement and needs.

**2.4. Behavioral Feasibility**

People are inherent to change. In this type of feasibility check, we come to know if the newly developed system will be taken and accepted by the working force i.e. the people who will use it.

## CHAPTER: 3

## PROJECT PLANING

## Project planning is the first step when we making any project. Project planning is the part of the project management, which relate to the use of schedules such as the Gantt chart to plan and subsequently report progress within the project environment.

## Project planning requires an in depth analysis and structuring of the following.

## 3.1 Setting the project goal:-

## Provide a platform to play audio file.

## Provide different interfaces for different level of users.

## Support playlist file playlist.

## Provide the feasibility of shake and play and also include audio recording.

## Provide the feasibility of pause the music via sensor.

## Provide the theme changer.

## Project Deliverable:-

## Create the GUI (Graphical user interface) 2 week

## Fetch the songs from SD card 3 days

## Theme setting work 1 week

## Automatic play music with shaking 1 week

## Audio recording 5 days

## Pause the song with sensor 3 week

## Supporting Plans

## Support planning can be defined as the process which helps people to set their own aim, then secure the support and care that are needed to achieve them. It is the key that unlocks person centered, coordinated

## 

## Human resource plan :

## Identify, by name, the individuals with a leading role in the project.

## 

## Risk Management plan :

Risk Management is the identification, assessment and prioritization of risk followed by coordinated and economical application of resource to minimize monitor and control the probability and impact of unfortunate events.

## CHAPTER: 4

## Requirement Analysis

## Requirements analysis is the process of determining user expectations for new or modified product. These features called requirement must be quantifiable, relevant and detailed. In software engineering such requirements are often called function specifications. Requirements analysis is an important aspect of project management. Requirements analysis Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to mold user expectations to fit the requirements. Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

**4.1 User Requirements:-**

* File and folder manager for the browsing of the music library
* 'Now playing' music console (pause, play, next etc.)
* Basic audio files codec support.
* Volume adjuster (equalizer optional)
* Song duration with song completion status bar.
* Album art display compatibility (highly recommended)
* Playlist support and Playlist creation.
* Artist, all songs, album etc. Organization
* Status bar support

**CHAPTER: 5**

**DEVLOPMENT REQUIRMENT ANALYSIS**

**5.1 Software Requirements:-**

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

**System Specification**

* SDK
* I.D.E. Eclipse/Android Studio
* API Level 21

**Mobile Specification**

*Android* The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardwarerequirements list is often accompanied by a hardwar**e** compatibility list (HCL), especially in case of operating systems.

* version Minimum 4.2(Jellybean)

**5.2 Hardware Requirements :-**

**System Specification**

* Microprocessor :- Pentium-4 class Processor , 450 MHz
* RAM :- 256 MB
* Hard Disk :- 40 GB on installation Drive ,which include 500 MB on System Drive.

**Mobile Specification**

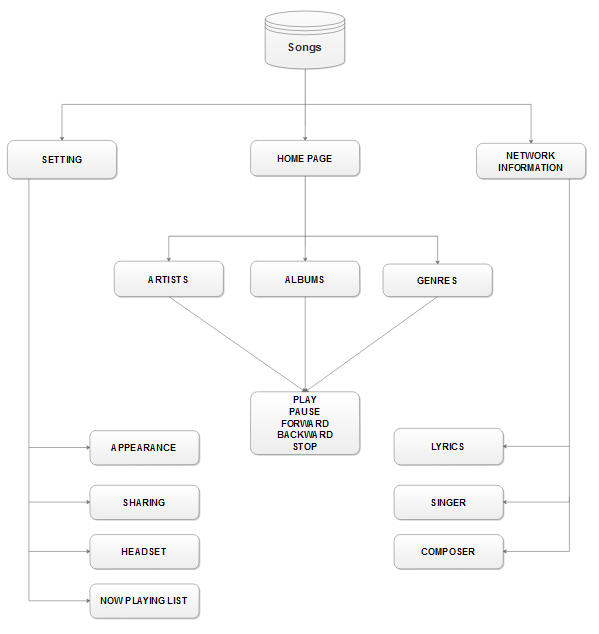
* RAM 256 MB
* Memory 50MB
* Headphone jack
* Speaker and Bluetooth

**CHAPTER: 6**

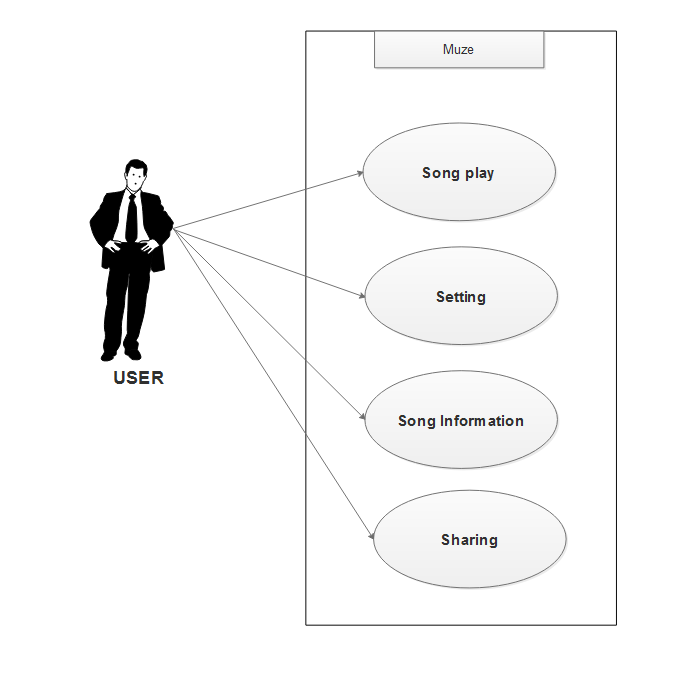
**SYSTEM DESIGN**

Systems design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. The actual power of software development lies in modular design of software. The logical separable parts of the project are called modules. Modular design not only supports easy development but also brings the clarity in the software. It is a good programming approach.

**6.1 Block Diagram:-**

****

**6.2 Use-Case Diagram:-**

****

**CHAPTER:7**

**DETAILED DESIGN**

**7.1 Data Flow Diagram:-**

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destination. The data flow diagram is analogous to a road map. It is a network model of all possibilities with different detail shown on different hierarchical levels. This processes of representing different details level is called “leveling” or “partitioning” by some data flow diagram advocates. Like a road map, there is no starting point or stop point, no time or timing, or steps to get somewhere. We just know that the data path must exist because at some point it will be needed. A road map shows all existing or planned roads because the road is needed.

Details that is not shown on the different levels of the data flow diagram such as volumes, timing, frequency, etc. is shown on supplementary diagrams or in the data dictionary. For example, data store contents may be shown in the data dictionary.

Data Flow Diagram (DFD) uses a number of symbols to represent the systems. Data Flow Diagram also known as ‘Bubble Chart’ is used to clarify system requirements and identifying the major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the level of details

**Terms used in DFD :**

**1. Process :**

A process transforms data values. The lowest level processes are pure functions without side effects. An entire data flow graphics high level process.

**Graphical Representation: **

**2. Data flows :**

A data flow connects the output of an object or process to input of another object or process. It represents the intermediate data value within a computation. It is represented by an arrow and labeled with a description of data, usually its name or type.

**Graphical Representation:**

****

**3. Actors :**

An actor is active object that drives the data flow graph by producing or consuming values.

**4. Data store :**

A data store is a passive object with in a data flow diagram that stores data for later access.

**Graphical Representation: **

****

**5. External Entity**

A rectangle represents an external entity such as a librarian, a library member.

**Graphical Representation:**

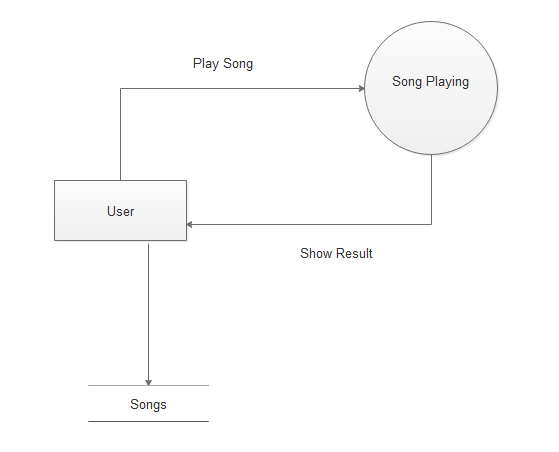
****

**6. Output Symbol**

This box represented data production during human computer interaction.

**Graphical Representation:**

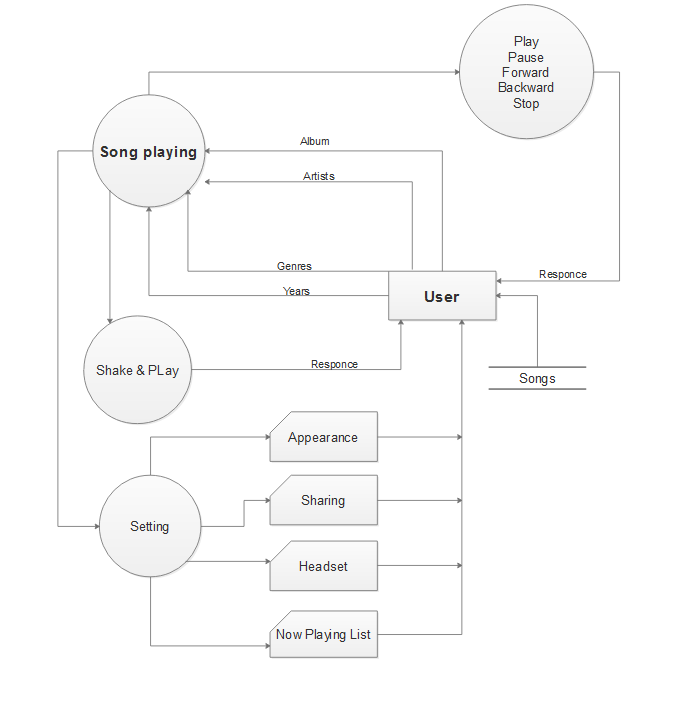
**Context Level DFD :**

****

# 1st Level DFD :

# C:\Users\joy\Desktop\Diagrams\prdfd1.PNG

# 2nd Level DFD :



**7.2 E - R DIAGRAM :**

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destination.

The data flow diagram is analogous to a road map. It is a network model of all possibilities with different detail shown on different hierarchical levels. This processes of representing different details level is called “leveling” or “partitioning” by some data flow diagram advocates. Like a road map, there is no starting point or stop point, any time or timing, or steps to get somewhere. We just know that the data path must exist because at some point it will be needed. A road map shows all existing or planned roads because the road is needed.

Details that is not shown on the different levels of the data flow diagram such as volumes, timing, frequency, etc. is shown on supplementary diagrams or in the data dictionary. For example, data store contents may be shown in the data dictionary.

Data Flow Diagram (DFD) uses a number of symbols to represent the systems. Data Flow Diagram also known as ‘Bubble Chart’ is used to clarify system requirements and identifying the major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the level of details

# C:\Users\joy\Desktop\Diagrams\httpatomoreillycomsourceoreillyimages234935.png

**CHAPTER :8**

**CODING**

**ActivityMainMenu.java**

package com.example.muze.muze.activities;

import java.util.ArrayList;

import android.content.Context;

import android.content.Intent;

import android.os.AsyncTask;

import android.os.Bundle;

import android.os.Handler;

import android.preference.PreferenceManager;

import android.util.Log;

import android.view.View;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.Toast;

import com.example.muze.muze.MainActivity;

import com.example.muze.muze.NotificationMusic;

import com.example.muze.muze.R;

import com.example.muze.muze.helpers.SingleToast;

import com.example.muze.muze.kMP;

/\*\*

\* First screen that the user sees - the Main Menu.

\*

\* Must listen for clicks so we can change to the other

\* sub menus (Activities).

\*

\* Thanks for providing a basic ListView navigation layout:

\* http://stackoverflow.com/q/19476948

\*/

public class ActivityMenuMain extends ActivityMaster

implements OnItemClickListener {

/\*\*

\* All the possible items the user can select on this menu.

\*

\* Will be initialized with default values on `onCreate`.

\*/

public static final ArrayList<String> items = new ArrayList<String>();

// Adapter that will convert from Strings to List Items

public static ArrayAdapter<String> adapter = null;

/\*\*

\* List that will be populated with all the items.

\*

\* Look for it inside the res/layout xml files.

\*/

ListView listView;

/\*\*

\* ID we'll use when calling the settings window.

\* It'll say if the user changed theme or not.

\*

\* @see onActivityResult()

\*/

static final int USER\_CHANGED\_THEME = 1;

// These variables are used to allow user to

// press twice to exit the program

// (showing a message when pressing the first time).

private boolean backPressedOnce = false;

private Handler backPressedHandler = new Handler();

/\*\* How long to wait to disable double-pressing to quit \*/

private static final int BACK\_PRESSED\_DELAY = 2000;

/\*\* Action that actually disables double-pressing to quit \*/

private final Runnable backPressedTimeoutAction = new Runnable() {

@Override

public void run() {

backPressedOnce = false;

}

};

/\*\*

\* Called when the activity is created for the first time.

\*/

@Override

protected void onCreate(Bundle seventhSonOfASeventhSon) {

// We need to load the settings right before creating

// the first activity so that the user-selected theme

// will be applied to the first screen.

//

// Loading default settings at the first time the app;

// is loaded.

PreferenceManager.setDefaultValues(this, R.xml.preferences, false);

kMP.settings.load(this);

super.onCreate(seventhSonOfASeventhSon);

setContentView(R.layout.activity\_main\_menu);

// Adding all possible items on the main menu.

items.add(getString(R.string.menu\_main\_music));

//items.add(getString(R.string.menu\_main\_settings));

items.add(getString(R.string.menu\_main\_shuffle));

if (kMP.mainMenuHasNowPlayingItem)

items.add("Now playing");

// ListView to be populated with the menu items

listView = (ListView)findViewById(R.id.activity\_main\_menu\_list);

// Thing that converts the menu items to the ListView

adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1, items);

// Filling teh list with all the items

listView.setAdapter(adapter);

// We'll get warned when the user clicks on an item.

listView.setOnItemClickListener(this);

// Initializing the main program logic.

kMP.initialize(this);

scanSongs(false);

}

/\*\*

\* Starts the background process of scanning the songs.

\*

\* @param forceScan If we should scan again. You should set

\* this to true if you want to scan again

\* the database.

\* Otherwise, leave it `false` so we don't

\* rescan the songs when this Activity

\* is created again for some reason.

\*/

void scanSongs(boolean forceScan) {

// Loading all the songs from the device on a different thread.

// We'll only actually do it if they weren't loaded already

//

// See the implementation right at the end of this class.

if ((forceScan) || (! kMP.songs.isInitialized())) {

SingleToast.show(ActivityMenuMain.this,

getString(R.string.menu\_main\_scanning),

Toast.LENGTH\_LONG);

new ScanSongs().execute();

}

}

/\*\*

\* Will react to the user selecting an item.

\*/

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

// Gets the string value of the current item and

// compares to all possible items.

String currentItem = listView.getItemAtPosition(position).toString();

if (currentItem.equals(getString(R.string.menu\_main\_music))) {

startActivity(new Intent(this, ActivityMenuMusic.class));

}

else if (currentItem.equals(getString(R.string.menu\_main\_settings))) {

// Let's start the settings screen.

// While doing so, we need to know if the user have

// changed the theme.

// If he did, we'll refresh the screen.

// See `onActivityResult()`

Intent settingsIntent = new Intent(this, MainActivity.class);

startActivityForResult(settingsIntent, USER\_CHANGED\_THEME);

}

else if (currentItem.equals(getString(R.string.menu\_main\_shuffle))) {

// Can only jump to shuffle all songs if we've

// scanned all the songs from the device.

if (! kMP.songs.isInitialized()) {

SingleToast.show(this,

getString(R.string.menu\_music\_proceed\_error),

Toast.LENGTH\_LONG);

return;

}

// Shuffle all songs

kMP.nowPlayingList = kMP.songs.getSongs();

Intent nowPlayingIntent = new Intent(this, ActivityNowPlaying.class);

nowPlayingIntent.putExtra("sort", "random");

nowPlayingIntent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(nowPlayingIntent);

}

else if (currentItem.equals(getString(R.string.menu\_main\_now\_playing))) {

// Jump to Now Playing screen

startActivity(new Intent(this, ActivityNowPlaying.class));

}

else {

}

}

/\*\*

\* When destroying the Activity.

\*/

@Override

protected void onDestroy() {

super.onDestroy();

if (backPressedHandler != null)

backPressedHandler.removeCallbacks(backPressedTimeoutAction);

// Need to clear all the items otherwise

// they'll keep adding up.

items.clear();

// Cancel all thrown Notifications

NotificationMusic.cancelAll(this);

kMP.stopMusicService(this);

}

/\*\*

\* We're overriding the default behavior for when the

\* user presses the back button.

\*

\* This way, it will show "Please click BACK again to exit"

\* and if the user presses again it will quit.

\*

\* Thanks, guys at StackOverflow:

\* http://stackoverflow.com/a/13578600

\*/

@Override

public void onBackPressed() {

/\* if (this.backPressedOnce) {

// Default behavior, quit it

super.onBackPressed();

kMP.forceExit(this);

return;

}

this.backPressedOnce = true;

SingleToast.show(this, getString(R.string.menu\_main\_back\_to\_exit), Toast.LENGTH\_SHORT);

backPressedHandler.postDelayed(backPressedTimeoutAction, BACK\_PRESSED\_DELAY);\*/

super.onBackPressed();

Intent intent = new Intent(ActivityMenuMain.this, MainActivity.class);

intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(intent);

}

/\*\*

\* Activity is about to become visible - let's start the music

\* service.

\*/

@Override

protected void onStart() {

super.onStart();

kMP.startMusicService(this);

}

@Override

protected void onPause() {

super.onStart();

//kMP.startMusicService(this);

items.clear();

}

// HELPER METHODS

/\*\*

\* Does an action on another Thread.

\*

\* On this case, we'll scan the songs on the Android device

\* without blocking the main Thread.

\*

\* It gives a nice pop-up when finishes.

\*

\* Source:

\* http://answers.oreilly.com/topic/2699-how-to-handle-threads-in-android-and-what-you-need-to-watch-for/

\*/

class ScanSongs extends AsyncTask<String, Integer, String> {

/\*\*

\* The action we'll do in the background.

\*/

@Override

protected String doInBackground(String... params) {

try {

// Will scan all songs on the device

kMP.songs.scanSongs(ActivityMenuMain.this, "external");

return ActivityMenuMain.this.getString(R.string.menu\_main\_scanning\_ok);

}

catch (Exception e) {

Log.e("Couldn't execute background task", e.toString());

e.printStackTrace();

return ActivityMenuMain.this.getString(R.string.menu\_main\_scanning\_not\_ok);

}

}

/\*\*

\* Called once the background processing is done.

\*/

@Override

protected void onPostExecute(String result) {

super.onPostExecute(result);

SingleToast.show(ActivityMenuMain.this,

result,

Toast.LENGTH\_LONG);

}

}

/\*\*

\* Adds a new item "Now Playing" on the main menu, if

\* it ain't there yet.

\*/

public static void addNowPlayingItem(Context c) {

if (kMP.mainMenuHasNowPlayingItem)

return;

ActivityMenuMain.items.add(c.getString(R.string.menu\_main\_now\_playing));

kMP.mainMenuHasNowPlayingItem = true;

// Refresh ListView

adapter.notifyDataSetChanged();

}

}

**activiti\_main.xml**

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/swipe"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity"

android:orientation="vertical"

android:background="@drawable/back"

android:layout\_gravity="center\_horizontal">

<ListView

android:id="@+id/song\_list"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" />

</LinearLayout>

**ActvityListAlbum.java**

package com.example.muze.muze.activities;

import java.util.ArrayList;

import android.app.ActionBar;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import com.example.muze.muze.R;

import com.example.muze.muze.kMP;

/\*\*

\* Shows a list of albums from a specified artist.

\*/

public class ActivityListAlbums extends ActivityMaster

implements OnItemClickListener {

/\*\*

\* List of songs that will be shown to the user.

\*/

private ListView songListView;

/\*\*

\* Raw items that will be shown on ListView.

\*/

private ArrayList<String> items;

private String currentArtist;

@Override

protected void onCreate(Bundle tableSex) {

super.onCreate(tableSex);

setContentView(R.layout.activity\_list\_songs);

songListView = (ListView)findViewById(R.id.activity\_list\_songs\_list);

// We expect an extra with the artist name

Intent intent = getIntent();

Bundle bundle = intent.getExtras();

if (bundle == null)

throw new RuntimeException("Expected Artist Name");

// This is the artist that we'll display the albums.

currentArtist = (String)bundle.get("artist");

if (currentArtist == null || currentArtist.isEmpty())

throw new RuntimeException("Expected Artist Name");

this.setTitle(currentArtist);

// Connects the song list to an adapter

// (thing that creates several Layouts from the song list)

items = kMP.songs.getAlbumsByArtist(currentArtist);

// Let's prepend all the albums with this label.

// Then, when selecting the item, we'll need to

// subtract one.

items.add(0, getString(R.string.all\_songs));

// Adapter that will convert from Strings to List Items

final ArrayAdapter<String> adapter = new ArrayAdapter<String>

(this, android.R.layout.simple\_list\_item\_1, items);

// Filling teh list with all the items

songListView.setAdapter(adapter);

songListView.setOnItemClickListener(this);

// This enables the "Up" button on the top Action Bar

// Note that it returns to the parent Activity, specified

// on `AndroidManifest`

ActionBar actionBar = getActionBar();

actionBar.setDisplayHomeAsUpEnabled(true);

}

/\*\*

\* Will react to the user selecting an item.

\*/

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

// We can only handle the user choice from now on

// if we've successfully scanned the songs from the

// device.

if (! kMP.songs.isInitialized())

return;

// Let's switch to a song list.

Intent intent = new Intent(this, ActivityListSongs.class);

// This is the special case - the user selected "All Albums".

// We'll show all songs from this artist, then.

if (position == 0) {

kMP.musicList = kMP.songs.getSongsByArtist(currentArtist);

intent.putExtra("title", currentArtist);

}

// Regular case, user selected a specific album.

// Show all songs from that album.

else {

String selectedAlbum = items.get(position);

kMP.musicList = kMP.songs.getSongsByAlbum(selectedAlbum);

intent.putExtra("title", selectedAlbum);

}

startActivity(intent);

}

}

**ActivityMenuSetting.java**

**package com.example.muze.muze.activities;**

**import java.util.ArrayList;**

**import android.app.ActionBar;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.view.View;**

**import android.widget.AdapterView;**

**import android.widget.AdapterView.OnItemClickListener;**

**import android.widget.ArrayAdapter;**

**import android.widget.ListView;**

**import com.example.muze.muze.R;**

**import com.example.muze.muze.kMP;**

**/\*\***

**\* Shows a list of albums from a specified artist.**

**\*/**

**public class ActivityListAlbums extends ActivityMaster**

**implements OnItemClickListener {**

**/\*\***

**\* List of songs that will be shown to the user.**

**\*/**

**private ListView songListView;**

**/\*\***

**\* Raw items that will be shown on ListView.**

**\*/**

**private ArrayList<String> items;**

**private String currentArtist;**

**@Override**

**protected void onCreate(Bundle tableSex) {**

**super.onCreate(tableSex);**

**setContentView(R.layout.activity\_list\_songs);**

**songListView = (ListView)findViewById(R.id.activity\_list\_songs\_list);**

**// We expect an extra with the artist name**

**Intent intent = getIntent();**

**Bundle bundle = intent.getExtras();**

**if (bundle == null)**

**throw new RuntimeException("Expected Artist Name");**

**// This is the artist that we'll display the albums.**

**currentArtist = (String)bundle.get("artist");**

**if (currentArtist == null || currentArtist.isEmpty())**

**throw new RuntimeException("Expected Artist Name");**

**this.setTitle(currentArtist);**

**// Connects the song list to an adapter**

**// (thing that creates several Layouts from the song list)**

**items = kMP.songs.getAlbumsByArtist(currentArtist);**

**// Let's prepend all the albums with this label.**

**// Then, when selecting the item, we'll need to**

**// subtract one.**

**items.add(0, getString(R.string.all\_songs));**

**// Adapter that will convert from Strings to List Items**

**final ArrayAdapter<String> adapter = new ArrayAdapter<String>**

**(this, android.R.layout.simple\_list\_item\_1, items);**

**// Filling teh list with all the items**

**songListView.setAdapter(adapter);**

**songListView.setOnItemClickListener(this);**

**// This enables the "Up" button on the top Action Bar**

**// Note that it returns to the parent Activity, specified**

**// on `AndroidManifest`**

**ActionBar actionBar = getActionBar();**

**actionBar.setDisplayHomeAsUpEnabled(true);**

**}**

**/\*\***

**\* Will react to the user selecting an item.**

**\*/**

**@Override**

**public void onItemClick(AdapterView<?> parent, View view, int position, long id) {**

**// We can only handle the user choice from now on**

**// if we've successfully scanned the songs from the**

**// device.**

**if (! kMP.songs.isInitialized())**

**return;**

**// Let's switch to a song list.**

**Intent intent = new Intent(this, ActivityListSongs.class);**

**// This is the special case - the user selected "All Albums".**

**// We'll show all songs from this artist, then.**

**if (position == 0) {**

**kMP.musicList = kMP.songs.getSongsByArtist(currentArtist);**

**intent.putExtra("title", currentArtist);**

**}**

**// Regular case, user selected a specific album.**

**// Show all songs from that album.**

**else {**

**String selectedAlbum = items.get(position);**

**kMP.musicList = kMP.songs.getSongsByAlbum(selectedAlbum);**

**intent.putExtra("title", selectedAlbum);**

**}**

**startActivity(intent);**

**}**

**}**

**CHAPTER: 9**

**TESTING**

**9.1.SYSTEM TESTING :**

Here the System testing involved is the most widely used testing process consisting of five stages as shown in the figure. In general, the sequence of testing activities is component testing, integration testing, and then user testing. However, as defects are discovered at any one stage, they require program modifications to correct them and this may require other stages in the testing process to be repeated.

System testing

Sub-system testing

Acceptance testing

Module testing

Unit testing

**(Component testing) (Integration testing) (User testing**)

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of the software. The results of testing are used later on during maintenance also.

Testing is vital to the success of the system. System testing makes a logical assumption that if the parts of the system are correct, the goal will be successfully achieved. In adequate testing or non-testing leads to errors that may not appear until months or even years later (Remember the New York three day power failures due to a misplaced ‘Break’ statement).

This creates two problems:

1. The time lag between the cause and the appearance of the problem.
2. The time interval effect of the system errors on files and the records on the system.

A small error can conceivably explode into a much larger problem. Effective testing early in the process translates directly into long term cost savings from a reduced number of errors.

Another reason for system testing is it’s utility as a user oriented vehicle before implementation. The best program is worthless if it does not meet the user requirements. Unfortunately, the user’s demands are often compromised by efforts to facilitate program or design efficiency in terms of processing time or design efficiency.

Thus in this phase we went to test the code we wrote. We needed to know if the code compiled with the design or not? Whether the code gave the desired outputs on given inputs? Whether it was ready to be installed on the user’s computer or some more modifications were needed?

Through the web applications are characteristically different from there software counterparts but the basic approach for testing these web applications is quite similar. These basic steps of testing have been picked from software engineering practices.

The following are the steps, we undertook:

1. The content of the Intranet site is reviewed to uncover Content

Errors. Content Errors covers the typographical errors, grammatical errors, errors in content .

1. The design model of the web application is reviewed to uncover the navigation errors. Use cases, derived as a part of the analysis activity allows a web designer to exercise each usage scenario against the architectural and navigational design. In essence these non-executable tests help to uncover the errors in navigation.
2. When web applications are considered the concept of unit changes. Each web page encapsulates content navigation links, content and processing elements (Forms, Scripts, and JSP’s as in our case). It is not always possible to test each of these individually. Thus is the base of the web applications the unit to be considered is the web page. Unlike the testing of the algorithmic details of a module the data that flows across the module interface, page level testing for web applications is driven by content, processing and links encapsulating the web page.
3. The Assembled web application is tested for overall functionality and content delivery. The various user cases are used that test the system for errors and mistakes.
4. The Web application is tested for a variety of environmental settings and is tested for various configurations and upon various platforms.

The modules are integrated and integration test are conducted.

1. Thread based testing is done to monitor the regression tests so that the site does not become very slow is a lot of users are simultaneously logged on.
2. A controlled and monitored population of end users tests Intranet application, this all comprises of the User Acceptance Testing.

Because web applications evolve continuously, the testing process is an ongoing activity, conducted by web support staff in our case the Organization’s IS people who will finally update and manage the application.

**9.2.PSYCHOLOGY OF TESTING :**

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn’t work. Testing is the process of executing a program with the intent of finding errors.

**9.3.TESTING OBJECTIVES :**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

* Testing is a process of executing a program with the intent of finding an error.
* A successful test is one that uncovers an as yet undiscovered error.
* A good test case is one that has a high probability of finding error, if it exists.
* The tests are inadequate to detect possibly present errors.
* The software more or less confirms to the quality and reliable standards.

**9.4.LEVELS OF TESTING :**

In order to uncover the errors present in different phases, we have the concept of levels of testing. The basic levels of testing are

Client Needs Acceptance Testing

Requirements System Testing

Design Integration Testing

**Unit testing :**

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications, testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. There are so many modules like Login, HR Department, Interviewer Section, etc. Each module has been tested by giving different set.

**Integration Testing :**

After unit testing, we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this project the main system is formed by integrating all the modules. When integrating all the modules I have checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration.

**SYSTEM TESTING :**

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements.

Here entire ‘HRRP’ has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not.

**ACCEPTANCE TESTING :**

Acceptance Testing is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized.

Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

**WHITE BOX TESTING :**

This is a unit testing method, where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors.

I tested step wise every piece of code, taking care that every statement in the code is executed at least once. The white box testing is also called Glass Box Testing.

I have generated a list of test cases, sample data, which is used to check all possible combinations of execution paths through the code at every module level.

White-box test focuses on the program control structure. Test cases are derived to ensure that all statement in the program control structure. Test cases are derived to ensure that all statement in the program control structure. Test cases are derived to ensure that all statement in the program has been executed at least once during testing and that all logical conditions have been exercised. Basis path testing, a white box technique, makes use of program graphs (or graph matrices) to derive the set of linearly independent test that will ensure coverage. Condition and data flow testing further exercising degrees of complexity.

**BLACK BOX TESTING :**

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a block that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

Black-box test are designed to uncover errors functional requirement without regard to the internal workings of a program. Black-box testing techniques focus on the information domain of the software, deriving test cases by partitioning the input and output domain of a

program in manner that provides through test coverage. The black-box test is used to demonstrate that software functions are operational, that input is properly produced, and that the integrity of external information are maintained. A black-box test examines some fundamental aspect of a system with little or no regard for the integral logical structure of the software.

**9.5.TEST INFORMATION FLOW :**

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vortex of the spiral and, concentrates on each unit, component of the software as implemented in source code.

Considering the process from a procedural point of view, testing within the context of software engineering is actually a series of four steps that are implemented sequentially. The steps are shown in Figure. Initially, tests focus on each component individually, ensuring that it functions properly as unit. Hence, the name unit testing. Unit testing makes heavy use of white-box testing techniques, exercising specific paths in module’s control structure to ensure complete coverage and maximum error detection.

***System Testing***

***System Engineering***

***Validation Testing***

***Integration Testing***

***Design***

***Validation testing***

***Code***

##### 6.3 INFORMATION FLOW OF DATA FOR TESTING

Software Configuration

Test Results

**Error**

Error Rate Data

Expected Results

Test Configuration

Predicated Reliability

Correction

# CHAPTER: 10

# 

# SNAPSHOTS

* **Loading activity**

****

* **HomePage Activity**

# D:\Project\Document\Dfd\Screen shots\IMG_20160509_234349_573.JPG

* **Playing Screen Activity**

# D:\Project\Document\Dfd\Screen shots\IMG_20160509_234235_614.jpg

* **Genres Activity**

# D:\Project\Document\Dfd\Screen shots\IMG_20160510_002205_447.JPG

* **Albums Activity**

# D:\Project\Document\Dfd\Screen shots\IMG_20160510_002217_983.JPG

* **More Options Activity**

# D:\Project\Document\Dfd\Screen shots\IMG_20160509_234642_862.JPG

# Setting Activity

# D:\Project\Document\Dfd\Screen shots\IMG_20160509_234709_296.JPG

* **Record Sound Activity**

# D:\Project\Document\Dfd\Screen shots\IMG_20160509_234737_809.JPG

* **About Activity**

# D:\Project\Document\Dfd\Screen shots\IMG_20160509_234833_149.JPG

**CHAPTER :11**

**MAINTENANCE**

Maintenance of the project is very easy due to its modular design and concept any modification can be done very easily. All the data are stored in the software as per user need & if user wants to change he has to change that particular data, as it will be reflected in the software every where. Some of the maintenance applied is: -

1. **BREAKDOWN MAINTENANCE: -**

The maintenance is applied when an error occurs & system halts and further processing cannot be done .At this time user can view documentation or consult us for rectification & we will analyze and change the code if needed. Example: - If user gets a error “report width is larger than paper size” while printing report & reports can not be generated then by viewing the help documentation & changing the paper size to ‘A4’ size of default printer will rectify the problem.”

1. **PREVENTATIVE MAINTENANCE: -**

User does this maintenance at regular intervals for smooth functioning

(operation) of software as per procedure and steps mentioned in the manual.

Some reasons for maintenance are: -

1. Error Correction: - Errors, which were not caught during testing, after the system has, been implemented. Rectification of such errors is called corrective maintenance.
2. New or changed requirements: - When Organization requirements changes due to changing opportunities.
3. Improved performance or maintenance requirements: -Changes that is made to improve system performance or to make it easier to maintain in the future are called preventive maintenance. Advances in technology (Adaptive maintenance): - Adaptive maintenance includes all the changes made to a system in order to introduce a new technology.

# CHAPTER: 12

# CONCLUSION

In this project we make a music player application that is android based. In this music player there are many advance feature that make this application more advance than other music application, the advance feature of this music player is that we can automatically change the song by shaking, and also pause the song with simple hand sensing. There is also a feature is that we can share the app without going to file manager means we can directly share this app. In this application we can also record the audio.

The most interesting thing is that we can also change the theme of the music player with existing default theme in the smart phone. This application provide a user interactive Graphical User Interface. Most of the requirements of the users are meet in this application.

**CHAPTER: 13**

**FUTURE SCOPE**

* The Mp3 player is the most recent in an evolution format that have helped users to enjoy their tones.
* This application can be used by any android user to play and share them with others
* This application will be very helpful in editing songs.
* This application will provide easy access to user efficiently.
* User can enjoy music with better reliability and more features.
* In future we can also able to find out the lyrics of songs.

**CHAPTER: 14**

**DEPENDENCIES AND LIMITATION**

**11.1 Dependencies**

* This application depends on the API Level of the mobile.
* Android Phone is required for the application.
* Although this software is architecturally independent but following Hardware resources are required. For i.e. any Android phone ,bluetooth,sensors and relevant software.

**11.2 Limitations**

Since, every Application has some limitations so our proposed system is also not untouchable in this regard.

Though it was planned for this Application to be absolutely perfect but everything as such has some limitations, so does the Application. Following may be the drawback in this Application.

* This application can execute only on Smart Phone.
* The smart phone must be android OS, because this application cannot be execute on windows and iphone.
* The API Level of the android phone may be Jellybean or Higher Level
* This application does not provide network browsing option for songs.

**APPENDIX-A**

**This application uses various technology like Android, Java, XML, Sqligt(DataBase).**

* **Android**
* **JAVA**
* **XML**
* **DATABASE**

**Android** is a mobile operating system based on linux kernel and currently developed by Google. Initially Android was founded by Andy Rubin in 2003.

**Android Platform Differences**

Android is hailed as “the first complete, open, and free mobile platform.”

**Complete:** The designers took a comprehensive approach when they developed the Android platform.They began with a secure operating system and built a robust software framework on that allows for rich application development opportunities.

**Open:** The Android platform is provided through open source licensing.Developers have unprecedented access to the handset features when developing applications.

**Free:** Android applications are free to develop.There are no licensing or royalty fees to develop on the platform. No required membership fees. No required testing fees. No required signing or certification fees. Android applications can be distributed and commercialized in a variety of ways.

**Android: A Next Generation Platform**

Although Android has many innovative features not available in existing mobile platforms, its designers also leveraged many tried-and-true approaches proven to work in the wireless world. It’s true that many of these features appear in existing proprietary platforms, but Android combines them in a free and open fashion, while simultaneously addressing many of the flaws on these competing platforms.

**Free and Open Source**

Android is an open source platform. Neither developers nor handset manufacturers pay royalties or license fees to develop for the platform. The underlying operating system of Android is licensed under GNU General Public License Version 2 (GPLv2), a strong “copyleft” license where any third-party improvements must continue to fall under the open source licensing agreement terms.The Android framework is distributed under the Apache Software License (ASL/Apache2), which allows for the distribution of both open and closed source derivations of the source code.

**Android Versions** Aestro, Blendes, CupeCake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, IcecreamSandwich, Jellybeans, Kitkat, Lollipop, Marshmallow .

**APPENDIX-B**

**Introduction to Java**

Java is a high level, third-generation programming language, like C, Fortran, Perl and many others. It is a platform for distributed computing – a development and run-time environment that cointains built-in support for the World Wide Web.

**History of Java**

Java development began at Sun Microsystems in 1991, the same year the World Wide Web was conceived. Java’s creator, James Gosling did not design java for the Internet. His Objective was to create a common development environment for consumer electronic devices which was easily portable from one device to another.

This effort evolved into a language, code named Oak and later renamed Java that retains much of the syntax and power of c++ , but is simpler and more platform independent.

**Java Features**

**Some of the important features of Java are as follows:**

* **Simplicity**
* **Orientation**
* **Platform Independence**
* **Security**
* **High Performance**
* **Multi Threading**
* **Dynamic linking.**
* **Garbage Collection.**

One of the most important features of Java is Platform Independence which makes it famous and suitable language for World Wide Web.

**APPENDIX-C**

**Data base** is the most important thing in this universe as data base gives identity to a thing without data base existence of a thing is impossible while working on a project first step is to design a database.

**What is data base?**

Data Base is a collection of tables and table is a collection of records in a tabular form i.e. in row and columns.

**Data Base can be divided into two parts:-**

1. **RDBMS.**
2. **DBMS.**

We will be using RDBMS (Relational Database Management System) in our project i.e. oracle 9i Enterprise edition.

**Why we are using (RDBMS)?**

**Some of the merits of using (RDBMS) are as under:-**

* Centralization of database.
* Client Server Technology.
* Security.
* Normalization of Data Base.
* Relationship.
* Transaction Processor.

**What is SQLite?**

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is the one database, which is zero-configured, that means like other database you do not need to configure it in your system.

**Why SQLite?**

 SQLite does not require a separate server process or system to operate (serverless).

 SQLite comes with zero-configuration, which means no setup or administration needed.

 A complete SQLite database is stored in a single cross-platform disk file.

 SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.

 SQLite is self-contained, which means no external dependencies.

 SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.

 SQLite supports most of the query language features found in the SQL92 (SQL2) standard.

 SQLite is written in ANSI-C and provides simple and easy-to-use API.

 SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

TUTORIALS POINT Simply Easy Learning

**BIBILIOGRAPHY**

**BOOKS**

- Android App Development - by Donn Felker

- Fundamentals Of Database System - by Navathe

- Java 2 - by Herbert Schidt

- Beginning Android - by Mark L. Murphy

- Software Testing - Ron Patton

-Android.Application.Development.For.Dummies

**REFERENCES**

* <http://www.developer.android.com>
* http://[www.developers.com](http://www.developers.com)
* [http://www.androidpolice.com](http://www.s.com)
* <http://www.android.com>
* <http://www.google.com>
* <http://www.wikipedia.com>
* <http://www.tutorial.com>
* <http://stackoverflow.com>
* <http://www.quora.com>
* <http://www.androidhive.com>
* <http://www.coderzheaven.com>
* <http://www.youtube.com>
* <http://www.codeproject.com>
* http://android-er.blogspot.in
* <http://wptrafficanalyzer.in/blog>
* <http://code.tutsplus.com>