

MY TRAVEL GUIDE USING ANDROID APPLICATION DEVELOPMENT

**Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in computer science and
Engineering**

by

L.PRANEETH (REG.NO.37110400)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
ACCREDITED WITH GRADE "A" BY NAAC**

**JEPPIAR NAGAR,RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU.**

MARCH 2021

SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of
Y.MADHU GANESH(REG.NO.37110416) AND L.PRANEETH(REG.NO.37110400)
who carried out the project entitled "VOICE CONTROL MUSIC PLAYER" under
my supervision from August 2020 to March 2021

Internal Guide

Dr. S.MURAGAN, M.E., Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Y.MADHU GANESH AND L.PRANEETH** hereby declare that the Project Report entitled "VOICE CONTROL MUSIC PLAYER" is done by me under the guidance of DR.S.MURAGAN,M.E.,PH.D.,Dept of COMPUTER SCIENCE AND ENGINEERING at Sathyabama Institute of science and technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean, school of computing** and **Dr. S. VIGNESWARI, M.E., Ph.D., Head of the Department, Dept. of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of the gratitude to my project guide to **Dr.S.MURAGAN M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT:

Android OS makes up majority of the smartphone market. a recent study concluded that, the number of people using an android smartphone had surpassed 2 billion worldwide. various developers and organizations are frequently trying to develop music applications with improved functionalities and the latest technologies which will give the user a better experience. android based voice-controlled music player is an application that tries to make the user's life better. there are times when listening to music on any regular android application, we cannot change the song or give certain commands like pause, resume, next, previous and so on. for example, while jogging, weightlifting or in a crowded place where it becomes very difficult to operate our mobile phones and give commands to the player. in such situations, it becomes privilege to have a music player which is easy to interact with and which has a feature of voice recognition, so that the commands can be given by just telling the player what to do. once the application is open, a button click will activate the voice recognition. the user can now speak out what he wants the application to do, like play a certain song, pause, resume, previous and next, the music player provides many more features that the application can perform on the user's command. the music player will be created using Java. so, instead of using a music player which has to be operated with physical touches over and over it is better to have a music player which acts as an artificial intelligence simplifying our problem and make it easy to operate.

TABLE OF CONTENTS

Chapter No.	TITLE	page no:
1.	INTRODUCTION	
2.	LITREATURE SURVEY	
	2.1 Control Media Player with Voice Commands	
	2.2 Implementation and analysis of mood-based music	
	2.3 Music player based on emotion	
3.	LIFECYCLE	
	3.1 SDLC(umbrella model)	
	3.2 Stages of SDLC	
4.	SOFTWARE REQUIREMENTS	
	4.1 SOFTWARE REQUIREMENT SPECIFICATION	
5.	DESIGN	
	5.1 UML DIAGRAM	
6.	IMPLEMENTATION SMAPLE CODE	
	6.1 METHOD	
7.	RESULTS AND DISCUSSION	
8.	CONCLUSION	

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1	UMBRELLA DIAGRAM	4
2	REQUIREMENTS STAGE	5
3	ANALYSIS STAGE	6
4	DESIGNING STAGE	6
5	CODING STAGE	7
6	TEST STAGE	8
7	INSTALLATION STAGE	9
8	UML CLASS DIAGRAM	14
9	UML USE CASE DIAGRAM	15
10	SEQUENCE DIAGRAM	16
11	COMPONENT DIAGRAM	17
12	DEPLOYMENT DIAGRAM	18

1.INTRODUCTION:

Today, it is nearly not possible for people to live and carry out some operations without technology. People have developed and used technology every day for their own benefit. Today it is very easy to control any application with software. Nowadays, many applications can be controlled with software so that people can live more comfortably. People can see examples of such practices in every aspect of their lives. To facilitate the social life of people in studies in this field in the literature: Different voice recognition algorithms and command sets were used on java. With different voice recognition algorithms, “On TV”, “Off TV”, “Volume Up”, “Volume Down” and “Channel One” command sets were tried separately for male and female users. It has been tried by establishing different algorithms on a phone simulation. The results obtained were found to vary according to the way the voice is spoken.. Over 80% success was achieved in voice recognition on the letters “a”, “e” and “i”. In a different study, separate tests were performed on male and female users with different algorithms . Using artificial intelligence techniques, a voice recognition system independent of text and speaker was developed on the Turkish language. The syllable based Turkish word recognition system was developed using different voice recognition algorithms . In the simulation environment performed on java, the successful recognition rate for 10 people was found to be 99% . They performed music and speech recognition. Successful results were obtained in the study which performed 40 commands . It was controlled by voice commands of a remote controlled car . It has been tried to determine the English pronunciation of the numbers 0-9. In this study; an application was developed to control the media player to listen to music with voice commands over the computer. The application was implemented using the SpeechRecognitionEngine Class in the System.Speech library in the C # .Net Framework. If there is a match in voice commands, the operations that can be done with mouse and keyboard are executed.

2.LITERATURE SURVEY:

2.1 An Application to Control Media Player with Voice Commands

Using technology today is of great importance in terms of making people's lives easier. It has become very easy to run some applications with technology. In this study, an application that provides media player control with voice commands was developed. This application was developed to address the needs of people who cannot listen to music on their own due to any disability. The application was implemented in java language. In order to manage the media player with voice commands, voice recognition libraries were first used. In the developed application, operations with keyboard and mouse can be done with voice commands. Voice commands can be sent with the wireless headset from anywhere in the shooting area.

A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions

Automated analysis of human affective behavior has attracted increasing attention from researchers in psychology, computer science, linguistics, neuroscience, and related disciplines. However, the existing methods typically handle only deliberately displayed and exaggerated expressions of prototypical emotions despite the fact that deliberate behaviour differs in visual appearance, audio profile, and timing from spontaneously occurring behaviour. To address this problem, efforts to develop algorithms that can process naturally occurring human affective behaviour have recently emerged. Moreover, an increasing number of efforts are reported toward multimodal fusion for human affect analysis including audiovisual fusion, linguistic and paralinguistic fusion, and multi-cue visual fusion based on facial expressions, head movements, and body gestures. This paper introduces and surveys these recent advances. We first discuss human emotion perception from a psychological perspective. Next we examine available approaches to solving the problem of machine understanding of human affective behavior, and discuss important issues like the collection

and availability of training and test data. We finally outline some of the scientific and engineering challenges to advancing human affect sensing technology.

2.2 An emotion based music player for Android

Music plays a very important role in human's daily life and in the modern advanced technologies. Usually, the user has to face the task of manually browsing through the playlist of songs to select. Here we are proposing an efficient and accurate model that would generate a playlist based on current emotional state and behavior of the user. Existing methods for automating the playlist generation process are computationally slow, less accurate and sometimes even require use of additional hardware like EEG or sensors. Speech is the most ancient and natural way of expressing feelings, emotions and mood and its processing requires high computational, time, and cost. This proposed system based on real-time extraction of facial expressions as well as extracting audio features from songs to classify into a specific emotion that will generate a playlist automatically such that the computation cost is systemrelatively low.

Implementation and analysis of mood-based music recommendation system

Music plays a very important role in human's daily life and in the modern advanced technologies. Usually, the user has to face the task of manually browsing through the playlist of songs to select. Here we are proposing an efficient and accurate model, that would generate a playlist based on current emotional state and behavior of the user. Existing methods for automating the playlist generation process are computationally slow, less accurate and sometimes even require use of additional hardware like EEG or sensors. Speech is the most ancient and natural way of expressing feelings, emotions and mood and its processing requires high computational, time, and cost. This proposed system based on real-time extraction of facial expressions as well as extracting audio features from songs to classify into a specific emotion that will generate a playlist automatically such that the computation cost is relatively low.

2.3 Music player based on emotion recognition of voice signals

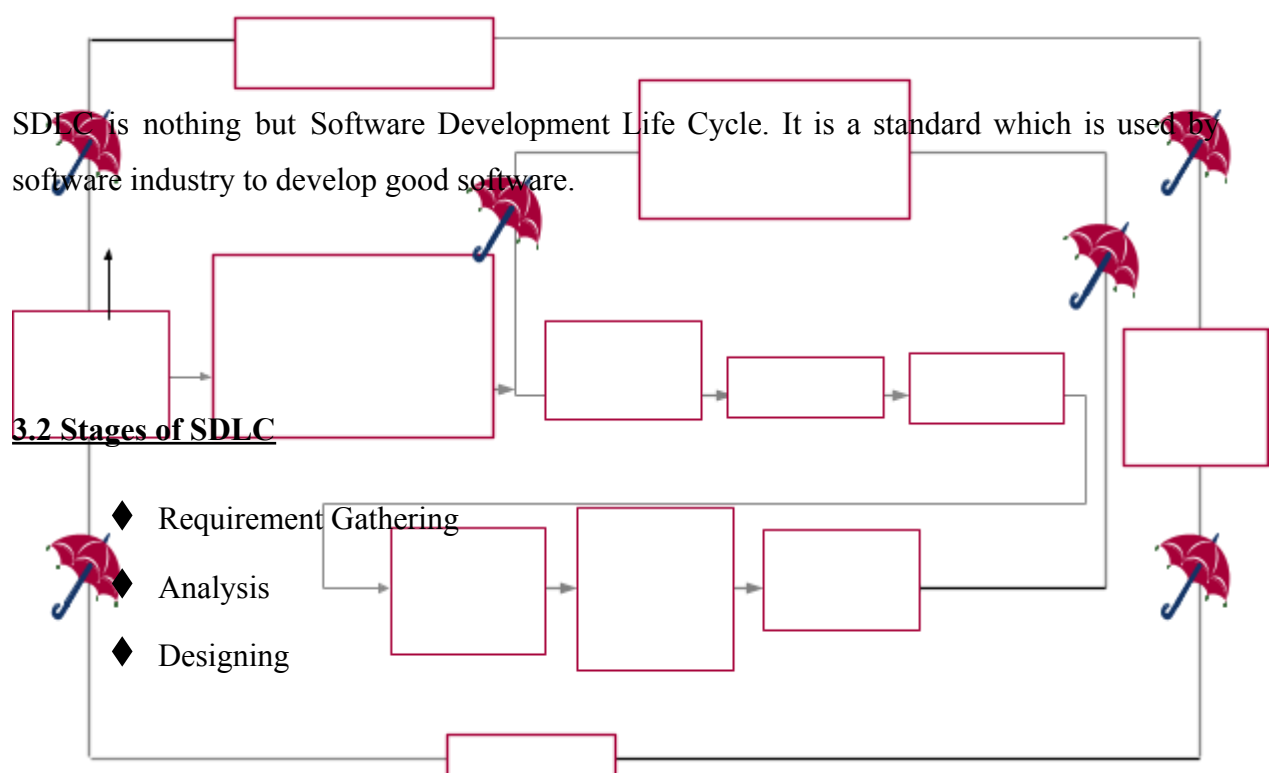
In this paper, a smart music system is designed by recognizing the emotion using voice speech signal as an input. The objective of the speech emotion recognition (SER) system is to determine the state of emotion of a human being's voice. This study recognizes five emotions-anger, anxiety, boredom, happiness and sadness. The important aspects in implementing this SER system includes the speech processing using the Berlin emotional

database, then extracting suitable features and selecting appropriate pattern recognition or classifier methods to identify the emotional states. Once the emotion of the speech is recognized, the system platform automatically selects a piece of music as a cheer up strategy from the database of song playlist stored. The analysis results show that this SER system implemented over five emotions provides successful emotional classification performance of 76.31% using GMM model and an overall better accuracy of 81.57% with SVM model.

3.LIFECYCLE:

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system the software development process.

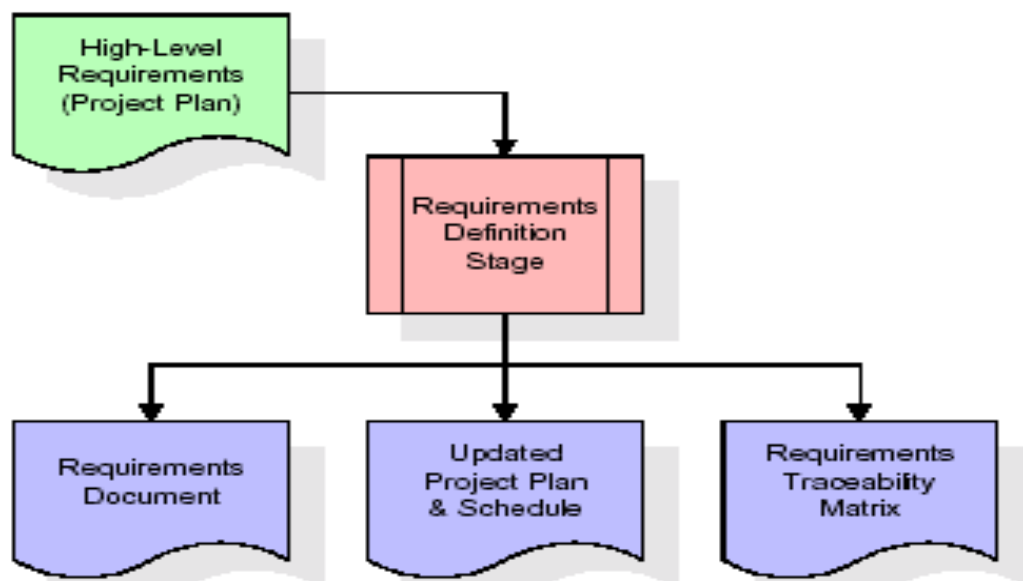
3.1 SDLC (Umbrella Model):



- ◆ Coding
- ◆ Testing
- ◆ Maintenance

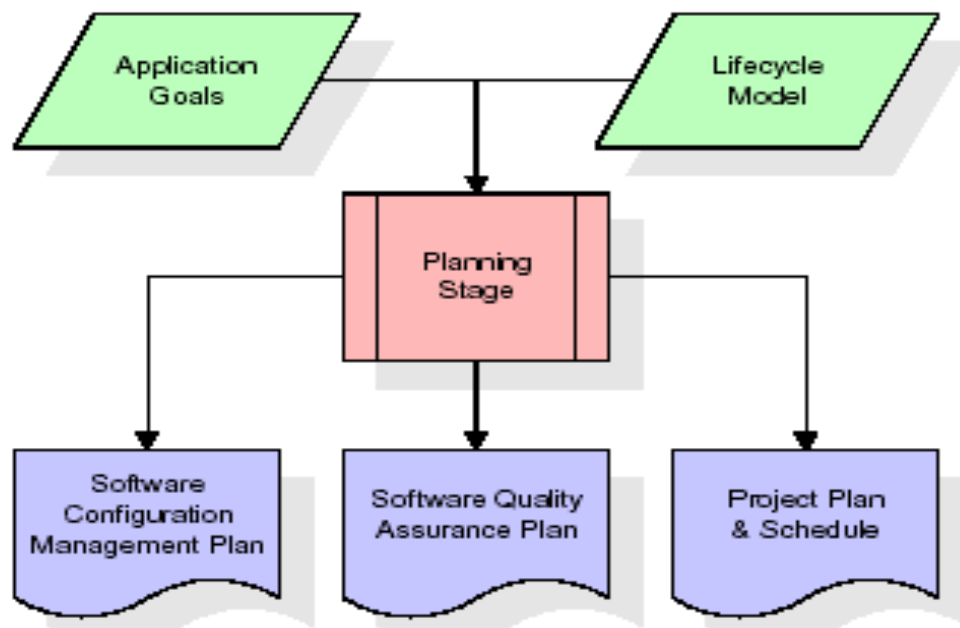
Requirements Gathering stage:

These necessities gather processing tookbe information objectives recognized within elevated prerequisites segment about venture method. Every objective will a lot about at least necessities. These necessities characterize the significant elements of the expected application, characterize operational information zones and reference information territories, and characterize underly data substances. Significant capacity incorporates basics procedure toward overseen, just as strategic information sources, yieldas well as documents.



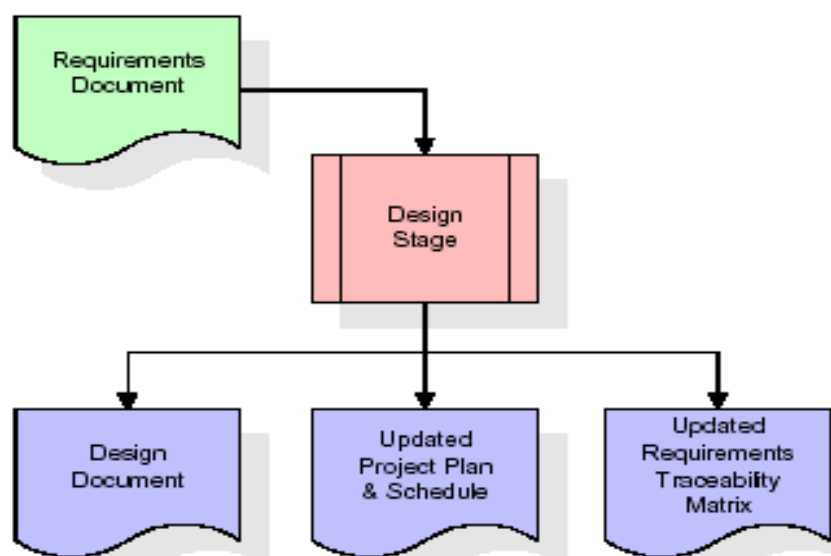
Analysis Stage:

The arranging stage builds up an elevated perspective on the planned programming thing, and usages this to set up initial endeavor framework, survey feasibility as well as risk related through endeavor, delineate fitting organization along with specific approaches.



The more fundamental zone about errand design be posts about critical level essentials, similarly implied targets. The total about items needs toward be made while necessities definitions level stream at any rate anyone about targets

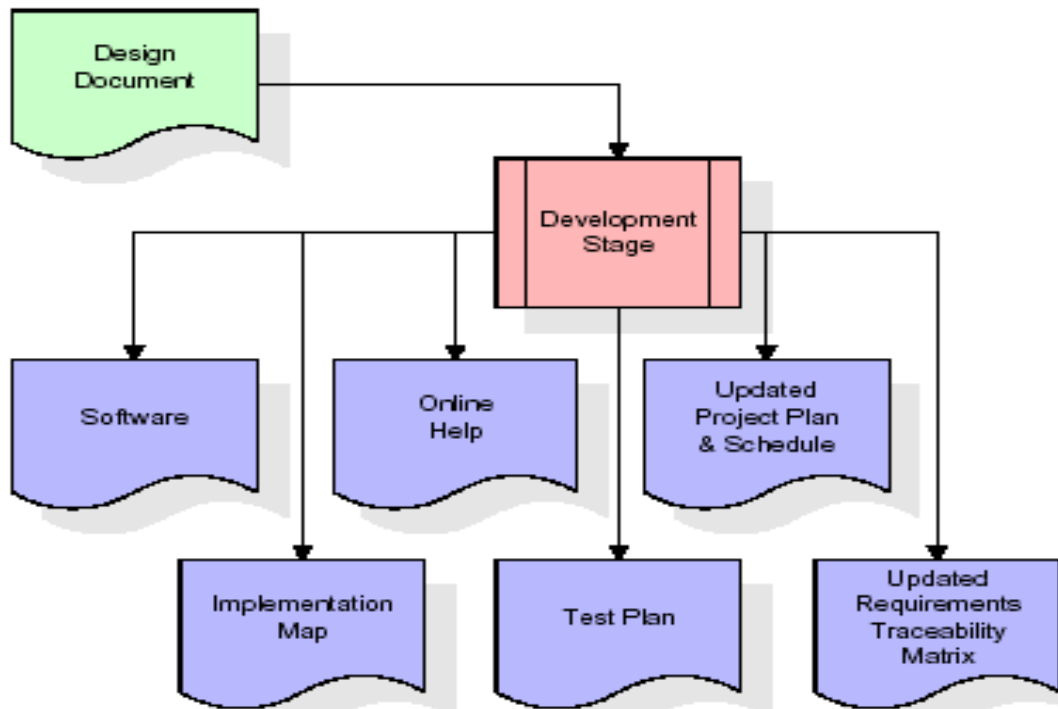
Designing Stage:



At point while structure records be settled as well as acknowledged, RTM be refreshed toward displays this every plan component be officially connected through particular prerequisite. The yield about structure stage be framework reports, a refresh RTM, be refreshed under method.

Development (Coding) Stage:

The improvement state took be fundamental data arrangement segments delineated within confirmed structure reports. Intended to every arrangements segment, the deals about in any event single programming relics should conveyed. Programming relics join but no obliged toward menus, talked, and information administrators structures, information itemizing positions, thought strategy and limits.



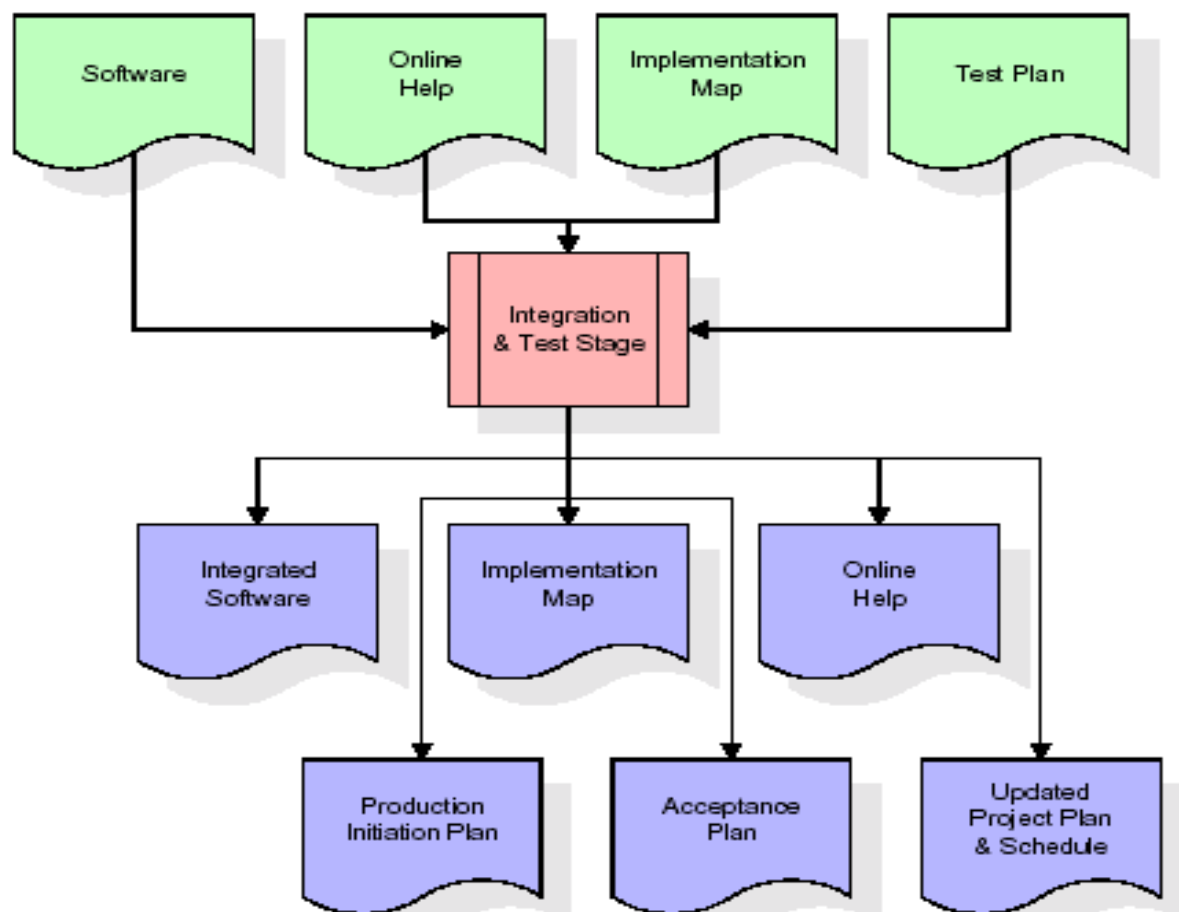
The RTM be toward show so as to every makes old irregularity be associates through specific arrangement segment, as well as to every developed artifact have at any rate looking at explore thing. Rightnow, the RTM be with structure. The yields about implement stage fuse

totally utilitarian course about action of programming so as to satisfies requirements along structure parts as of late recorded, this depicts action about item, an execution maps this perceives fundamental code area centers to total noteworthy works, a test plansbe portrays investigations toward utilized to support rightness along finish about item, an invigorated RTM, as well as a revived endeavor design.

Integration & Test Stage:

While joining as well as test stage, the item doodads, online help, as well as test informationbe migratesthrough implement condition toward an alternate test conditions. Here, total tests be rushed toward affirm the precision as well as satisfyabout test.

The yields
about
joining as
well as
test state

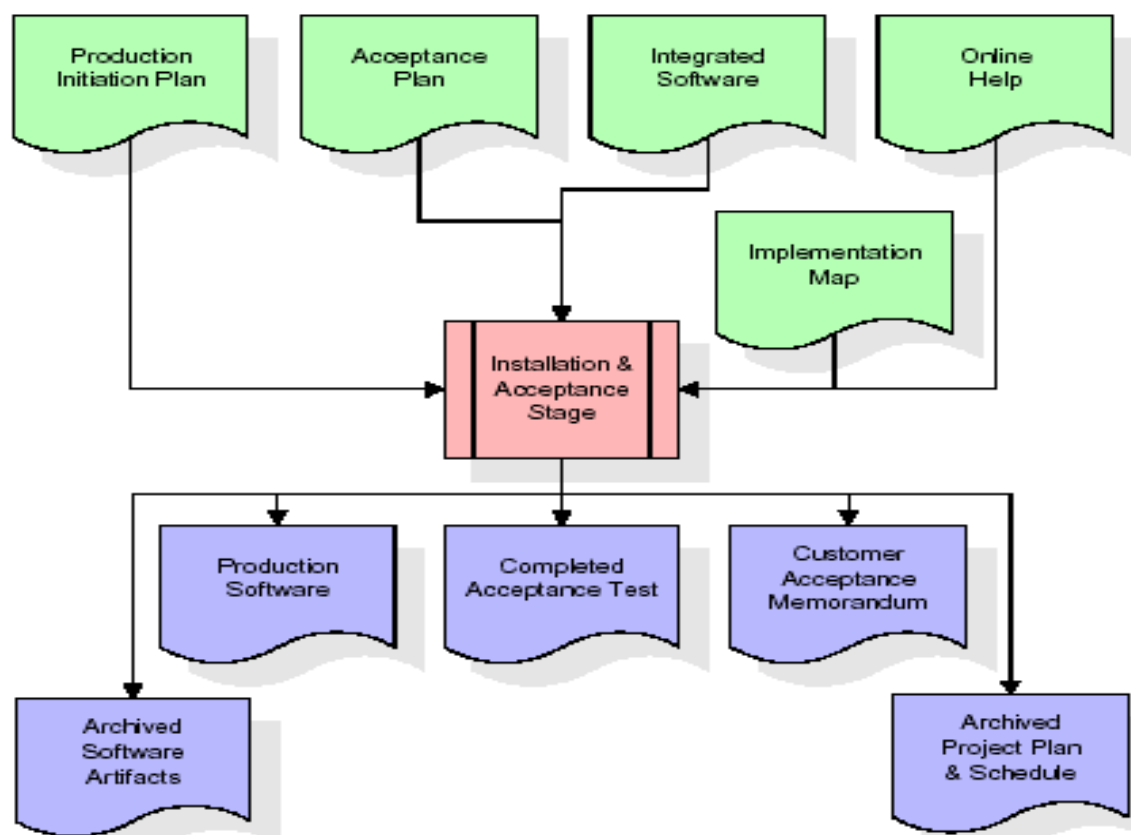


consolidate the organized game plan about programming, utilization map, a creation origin

design so as to depicts referenced information along creation customers, an affirmation design having last set-up of trials, along revived endeavor design

◆ Installation & Acceptance Test:

Then client staff has affirmed so as to basic creation information loads be corrects as well as test be executes through acceptable consequences, the client authoritatively recognizes transport about product.



Maintenance:

Outsides square shapes addresses backing about undertaking, Management should starts through essential assessment, perception about document later agents should be assigned works along that should encounters getting ready upon so as particulars consigned classes. Intended to that life cycle there does not have any conclusions.

4 Software Requirements:

Android is a Linux-based working framework planned fundamentally for touchscreen cell phones, for example, cell phones and tablet computers. Initially created by Android, Inc., which Google upheld monetarily and later purchased in 2005. Android is open source and Google delivers the code under the Apache License. This open source code and lenient permitting permits the product to be uninhibitedly adjusted and circulated by gadget makers, remote transporters and fan engineers. Moreover, Android has a huge local area of engineers composing ("applications") that broaden the usefulness of gadgets, composed fundamentally in an altered variant of the Java programming language.

Android programming advancement is the interaction by which new applications are made for the Android working framework. Applications are typically evolved in the Java programming language utilizing the Android Software Development Kit. ADT (Android Development Tools) is the product used to create android applications. It fundamentally encases Eclipse IDE, which is a multi-language Integrated improvement climate (IDE) including a base workspace and an extensible module framework for redoing the climate.. The most recent adaptation accompanies ADT module preinstalled and packaged to the IDE.

Application programming interface (API) indicates how some product segments ought to cooperate with one another. By and by in the vast majority of the cases an API is a library that normally incorporates detail for schedules, information structures, object classes, and factors. An API determination can take numerous structures, including an International Standard like POSIX, seller documentation like the Microsoft Windows API, the libraries of a programming language, e.g., Standard Template Library in C++ or Java API. Google APIs can be downloaded from Google Code, Google's website for designer instruments, APIs and specialized assets. The Google Data API] permit software engineers to make applications that peruse and compose information from Google administrations. As of now, these incorporate APIs for Google Apps, Google Analytics, Blogger, Google Base, Google Book Search, Google Calendar, Google Code Search, Google Earth, Google Spreadsheets, Google Notebook, and Picasa Web Albums.

SDK (Software Development Kit or "devkit") is normally a bunch of programming advancement instruments that considers the formation of utilizations for a specific programming package, software structure, equipment stage, PC framework, computer game comfort, working framework, or comparable improvement stage. It could be something as basic as an application programming interface (API) as certain records to interface to a specific programming language or incorporate complex equipment to speak with a specific installed framework. Regular apparatuses incorporate troubleshooting helps and different utilities frequently introduced in a coordinated advancement climate (IDE). In the most recent form of ADT, the android SDK adds on to the IDE naturally when you unfasten and load the IDE. 6.

4.1 SOFTWARE REQUIREMENT SPECIFICATION

Overall Description

A Software Requirements Specification (SRS) – an essentials specific intended to an item structure be completed delineation about lead about work toward be made. This consolidates a ton about usage case so as to depictst all associations customers should have through item. Despite use cases, the SRS also having non-helpful needies. Nonfunctionals essentials are necessities which power goals on the arrangement or else use, (intended to instance, operational effectiveness requirements, qualities measures, structure impediments).

ECONOMIC FEASIBILITY

A framework can be grown in fact and that will be utilized whenever introduced should at present be a wise speculation for the association. In the prudent plausibility, the advancement cost in making the framework is assessed against a definitive advantage got from the new frameworks. Budgetary advantages must rise to or surpass the expenses. The framework is financially achievable. It doesn't require any expansion equipment or programming. Since the interface for this framework is created utilizing the current assets and advancements accessible at NIC, There is ostensible use and practical attainability beyond a shadow of a doubt

OPERATIONAL FEASIBILITY

Proposed projects are advantageous just in the event that they can be transformed out into data framework. That will meet the association's working prerequisites. Operational practicality parts of the undertaking are to be taken as a significant piece of the venture execution. This framework is focused to be as per the previously mentioned issues. In advance, the administration issues and client necessities have been mulled over. So there is no doubt of opposition from the clients that can sabotage the conceivable application benefits. The all around arranged plan would guarantee the ideal use of the PC assets and would help in the improvement of execution status.

TECHNICAL FEASIBILITY

Prior no framework existed to oblige the requirements of 'Secure Infrastructure Implementation System'. The current framework created is in fact attainable. It is an electronic UI for review work process at NIC-CSD. Consequently it gives a simple admittance to the clients. The data set's motivation is to make, set up and keep a work process among different substances to encourage all concerned clients in their different limits or jobs. Consent to the clients would be allowed dependent on the jobs indicated. Subsequently, it gives the specialized assurance of precision, dependability and security.

5 . DESIGN

5.1 UML diagrams

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

- **User Model View**

- i. This view represents the system from the users perspective.
- ii. The analysis representation describes a usage scenario from the end-users perspective.

- **Structural Model view**

- i. In this model the data and functionality are arrived from inside the system.
- ii. This model view models the static structures.

- **Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- **Implementation Model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

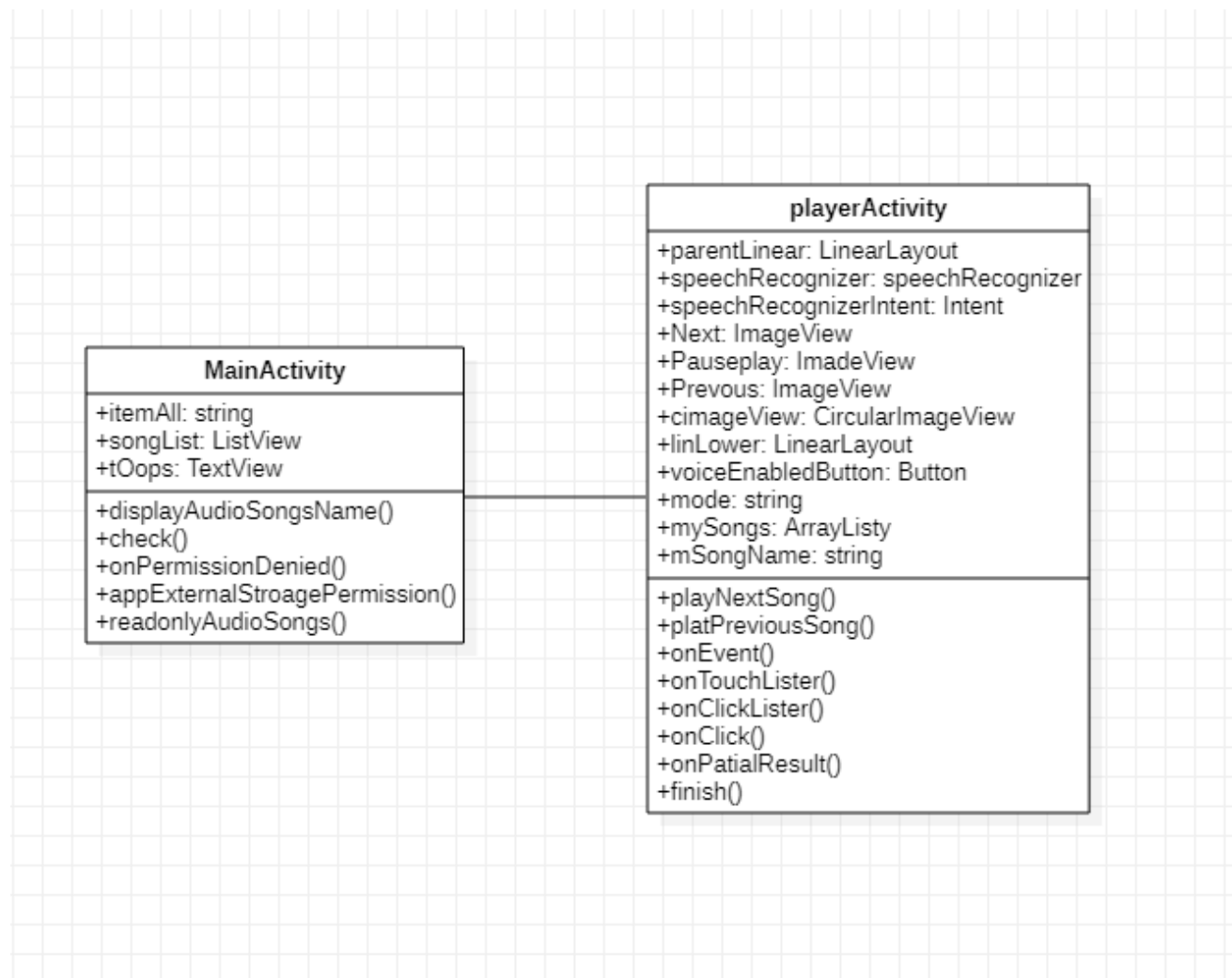
- **Environmental Model View**

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

5.2 Class diagram:

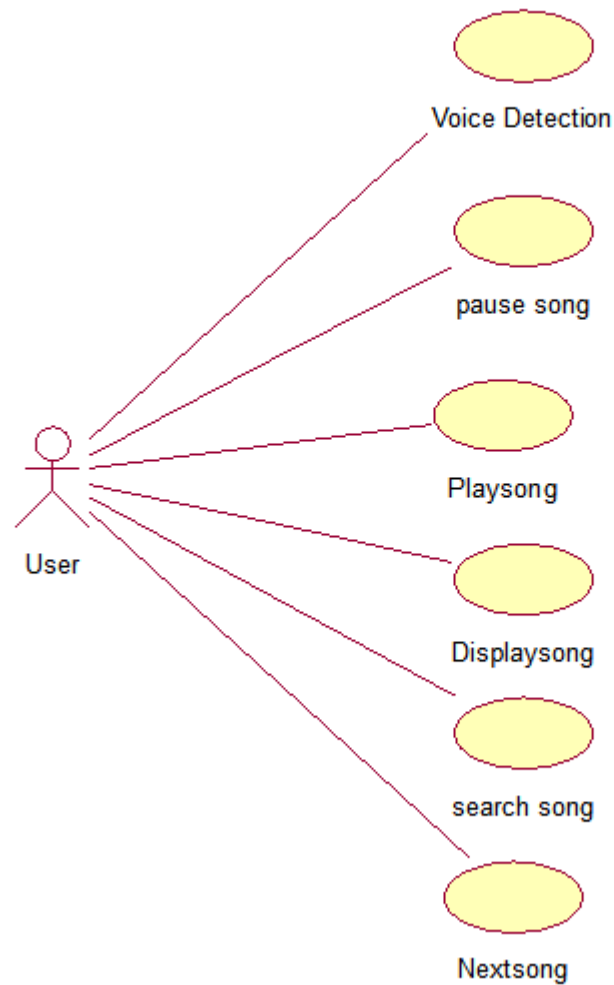
The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

- The initial part have the class name
- The second section have the attributes
- The third section contains the methods



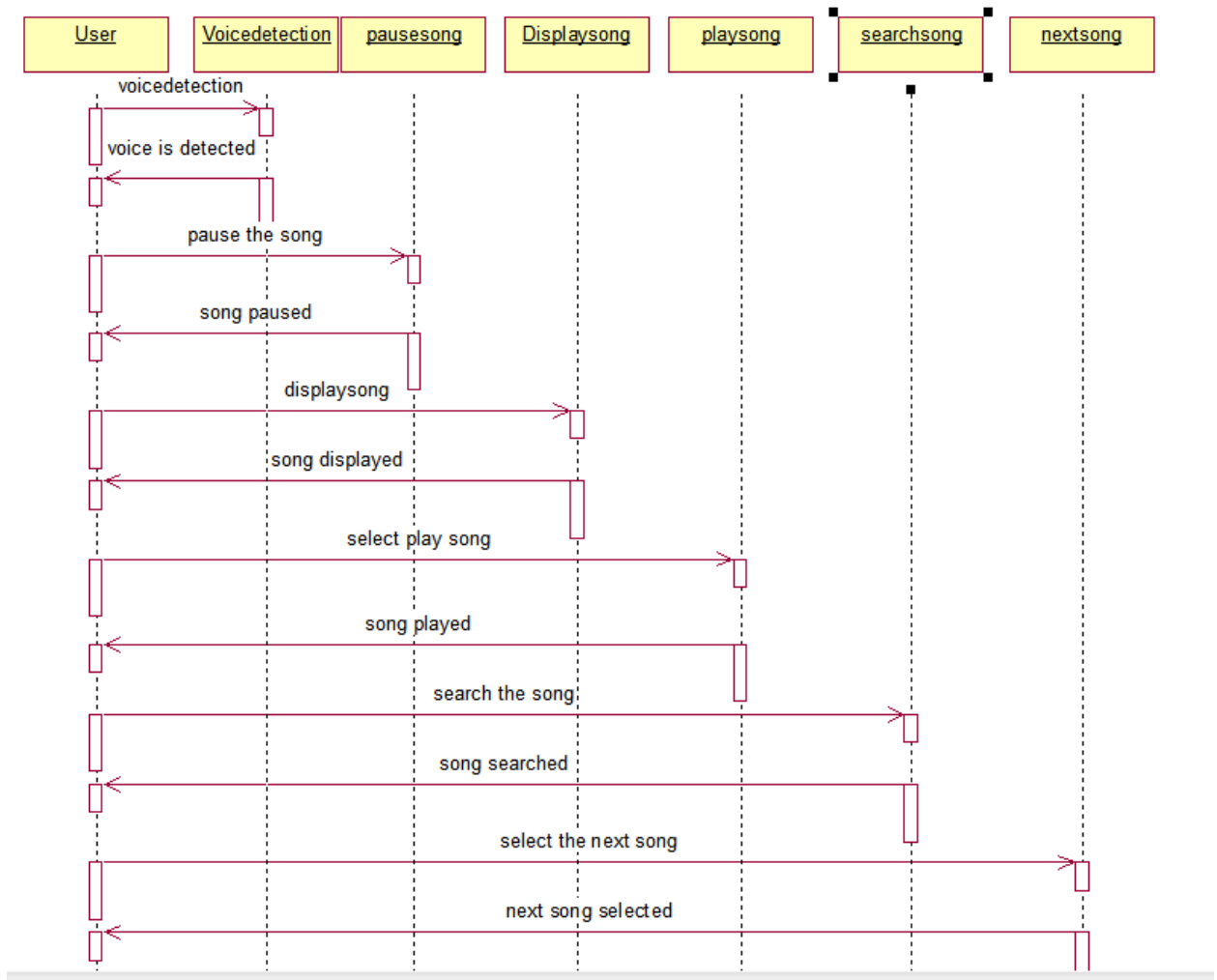
Use case Diagram:

A use case diagram at its simplest is a representation of a user's interaction can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



Sequence diagram:

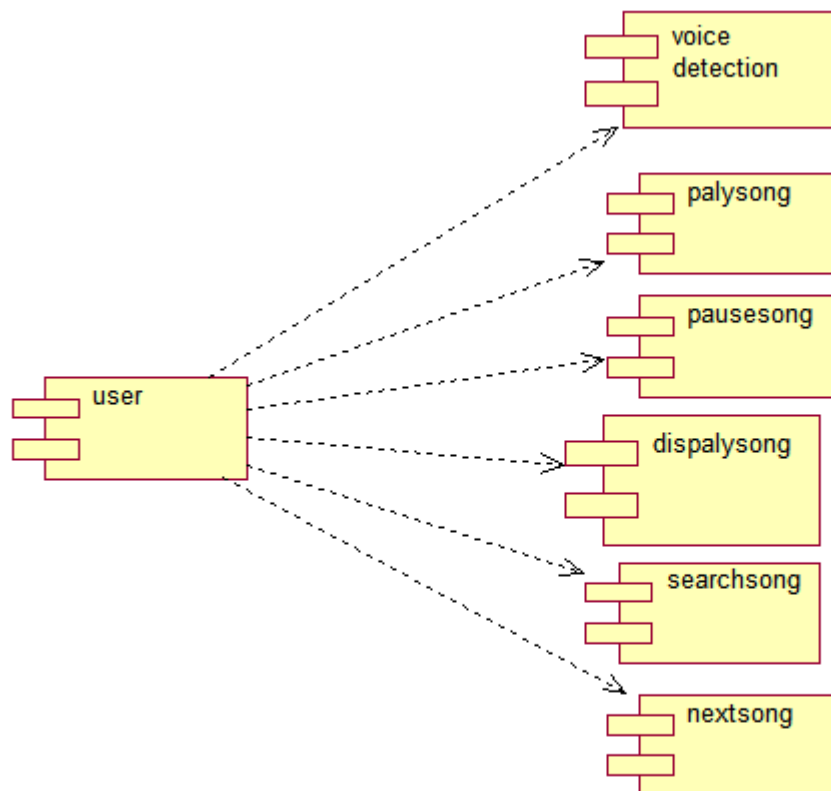
A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.



Component Diagram:

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

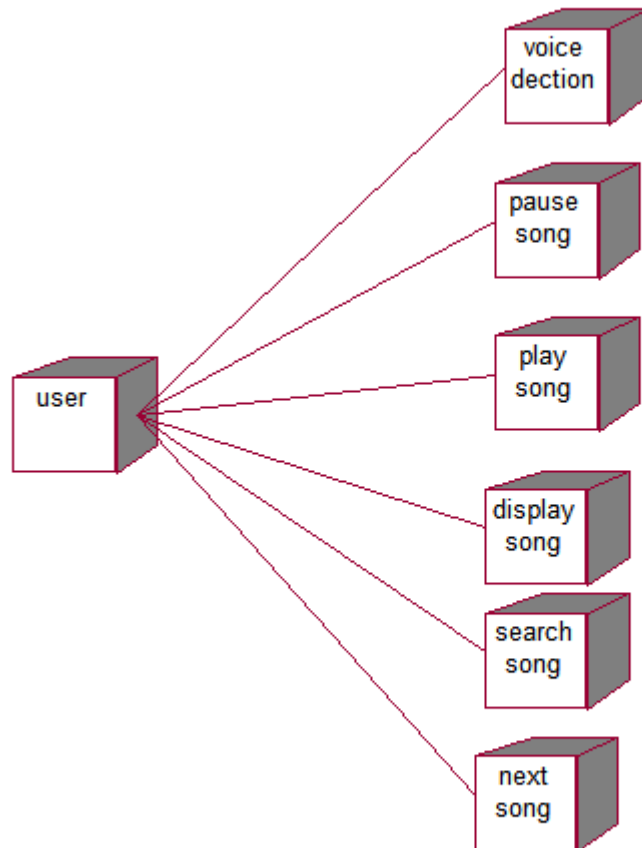
Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



Deployment Diagram:

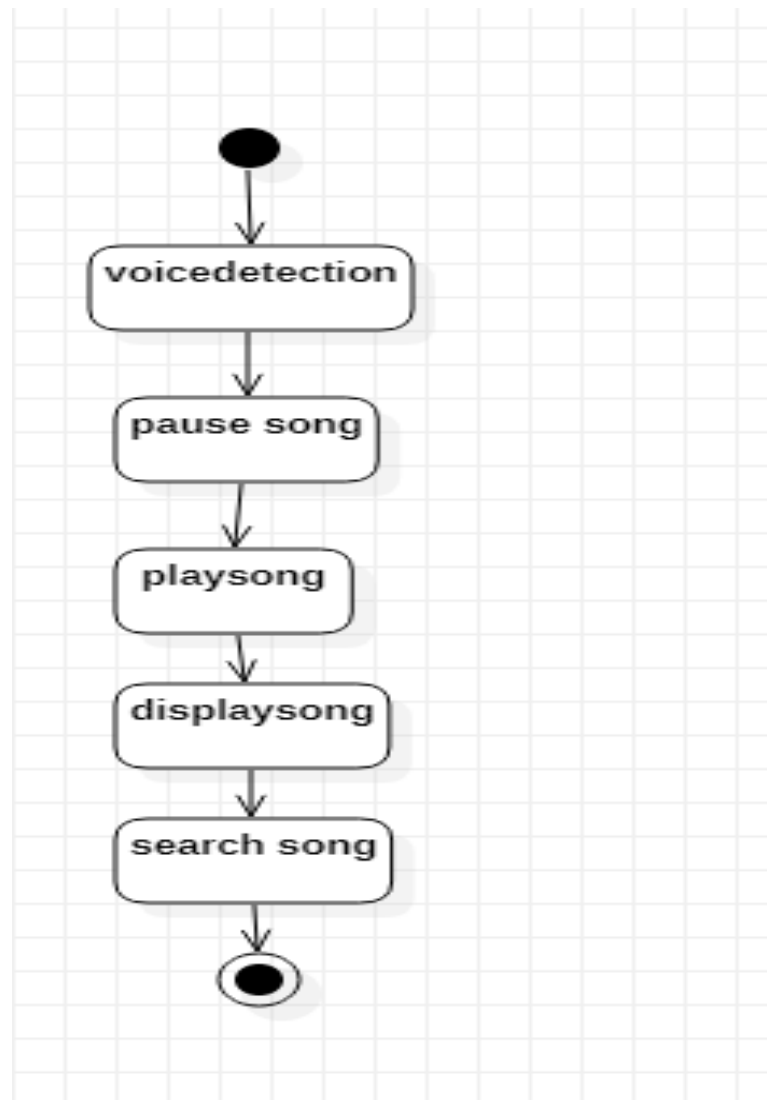
A deployment diagram in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



Activity Diagram:

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.



Data Flow Diagram:

Data stream diagrams outline how information be set up through structure similar toward wellsprings of informations as wella as yields. Information streams diagrams able toward part with a from about framework. The method initiates through general picture about business as well as.

6 IMPLEMENTATION:

The implementation stage be less innovative than framework plan. It is basically worried about client preparing, and record change. The framework might be requiring broad client preparing. The underlying boundaries of the framework ought to be adjusts because of a programming. A basic working strategy is given so the client can comprehend the various capacities obviously and rapidly. The various reports can be acquired either on the inkjet or spot grid printer, which is accessible at the removal of the client. The proposed framework is not difficult to execute. As a rule execution is utilized to mean the way toward changing over another or reconsidered framework plan into an operational one.

Sample Code:

```
package com.example.voicebasedmusicplayer;

import android.Manifest;
import android.content.Intent;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
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 android.widget.TextView;

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

import java.io.File;
```

```
import java.util.ArrayList;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    private String[] itemsAll;
```

```
    private ListView songsList;
```

```
    private TextView tvOops;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        songsList = findViewById(R.id.songsList);
```

```
        tvOops = findViewById(R.id.tv_oops);
```

```
        appExternalStorageStoragePermission();
```

```
    }
```

```
    public void appExternalStorageStoragePermission() {
```

```
        Dexter.withActivity(this)
```

```
            .withPermission(Manifest.permission.READ_EXTERNAL_STORAGE)
```

```
            .withListener(new PermissionListener() {
```

```
                @Override
```

```
                public void onPermissionGranted(PermissionGrantedResponse response) {
```

```
                    displayAudioSongsName();
```

```
                }
```

```
                @Override
```

```
                public void onPermissionDenied(PermissionDeniedResponse response) {
```

```
                }
```

```
            }
```

```

        public void onRequestPermissionsResult(int requestCode,
        permission, PermissionToken token) {
            token.continuePermissionRequest();
        }
    }).check();
}

public ArrayList<File> readOnlyAudioSongs(File file) {
    ArrayList<File> arrayList = new ArrayList<>();
    File[] allFiles = file.listFiles();
    for (File individualFile : allFiles) {
        if (individualFile.isDirectory() && !individualFile.isHidden()) {
            arrayList.addAll(readOnlyAudioSongs(individualFile));
        } else {
            if (individualFile.getName().endsWith(".mp3") ||
individualFile.getName().endsWith(".aac") || individualFile.getName().endsWith(".wav") ||
individualFile.getName().endsWith(".wma")) {
                arrayList.add(individualFile);
            }
        }
    }
    return arrayList;
}

private void displayAudioSongsName() {
    final ArrayList<File> audioSongs =
readOnlyAudioSongs(Environment.getExternalStorageDirectory());
    itemsAll = new String[audioSongs.size()];

    if (audioSongs.isEmpty()) {
        tvOops.setVisibility(View.VISIBLE);
        songsList.setVisibility(View.GONE);
    } else {
        tvOops.setVisibility(View.GONE);
    }
}

```

```

songsList.setVisibility(View.VISIBLE);
for (int songCounter = 0; songCounter < audioSongs.size(); songCounter++) {
    itemsAll[songCounter] = audioSongs.get(songCounter).getName();
}

ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(MainActivity.this,
android.R.layout.simple_list_item_1, itemsAll);
songsList.setAdapter(arrayAdapter);

songsList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int i, long l) {
        String songName = songsList.getItemAtPosition(i).toString();
        Intent intent = new Intent(MainActivity.this, PlayerActivity.class);
        intent.putExtra("song", audioSongs);
        intent.putExtra("name", songName);
        intent.putExtra("position", i);
        startActivity(intent);
    }
});
}

}

package com.example.voicebasedmusicplayer;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Build;
import android.provider.Settings;

```

```
import android.speech.RecognitionListener;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.mikhaellopez.circularimageview.CircularImageView;

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

public class PlayerActivity extends AppCompatActivity {
    private LinearLayout parentLinear;
    private SpeechRecognizer speechRecognizer;
    private Intent speechRecognizerIntent;
    private String keeper = "";

    private ImageView PausePlay, Next, Previous;
    private CircularImageView cimageView;
    private TextView SongName;
    private LinearLayout linLower;
    private Button VoiceEnabledButton;

    private String mode = "ON";
```

```

private MediaPlayer myMediaPlayer;
private int position;
private ArrayList<File> mySongs;
private String mSongName;

@SuppressLint("ClickableViewAccessibility")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);

    checkVoiceCommandPermission();

    PausePlay = findViewById(R.id.btn_PlayPause);
    Next = findViewById(R.id.btn_Next);
    Previous = findViewById(R.id.btn_Previous);
    SongName = findViewById(R.id.songName);
    linLower = findViewById(R.id.Lin_Lower);
    cimageView = findViewById(R.id.logo);
    VoiceEnabledButton = findViewById(R.id.btn_VoiceEnabled);

    parentLinear = findViewById(R.id.parentLinear);
    speechRecognizer = SpeechRecognizer.createSpeechRecognizer(PlayerActivity.this);
    speechRecognizerIntent = new
Intent(Intent.ACTION_RECOGNIZE_SPEECH);
    speechRecognizerIntent.putExtra(Intent.EXTRA_LANGUAGE_MODEL,
Intent.LANGUAGE_MODEL_WEB_SEARCH);
    speechRecognizerIntent.putExtra(Intent.EXTRA_CALLING_PACKAGE,
getPackageName());
    speechRecognizerIntent.putExtra(Intent.EXTRA_LANGUAGE,
Locale.getDefault());
    speechRecognizerIntent.putExtra(Intent.EXTRA_MAX_RESULTS, 3);

    validateReceiveValueAndStartPlaying();

```

```
cimageView.setBackgroundResource(R.drawable.logo);
```

```
speechRecognizer.setRecognitionListener(new RecognitionListener() {
```

```
    @Override
```

```
    public void onReadyForSpeech(Bundle params) {
```

```
    }
```

```
    @Override
```

```
    public void onBeginningOfSpeech() {
```

```
    }
```

```
    @Override
```

```
    public void onRmsChanged(float rmsdB) {
```

```
    }
```

```
    @Override
```

```
    public void onBufferReceived(byte[] buffer) {
```

```
    }
```

```
    @Override
```

```
    public void onEndOfSpeech() {
```

```
    }
```

```
    @Override
```

```
    public void onError(int error) {
```

```
    }
```

```
    @Override
```

```

public void onResults(Bundle bundle) {
    ArrayList<String> matchesFound =
bundle.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);

    if (matchesFound != null) {
        if (mode.equals("ON")) {
            keeper = matchesFound.get(0);

            if (keeper.equals("pause the song") || keeper.equals("pause
song"))||keeper.equals("hold the song")||keeper.equals("hold song")) {
                playPauseSong();
                Toast.makeText(PlayerActivity.this, "Command = " + keeper,
Toast.LENGTH_LONG).show();
            } else if (keeper.equals("play the song") || keeper.equals("play song")) {
                playPauseSong();
                Toast.makeText(PlayerActivity.this, "Command = " + keeper,
Toast.LENGTH_LONG).show();
            } else if (keeper.equals("play next song") || keeper.equals("play the next
song")) {
                playNextSong();
                Toast.makeText(PlayerActivity.this, "Command = " + keeper,
Toast.LENGTH_LONG).show();
            } else if (keeper.equals("play previous song") || keeper.equals("play the
previous song")) {
                playPreviousSong();
                Toast.makeText(PlayerActivity.this, "Command = " + keeper,
Toast.LENGTH_LONG).show();
            }
            else{
                Toast.makeText(PlayerActivity.this, "Didn't recognize the keyword..!Try
saying ", Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

        else{
            Toast.makeText(PlayerActivity.this, "Didn't recognize the keyword..!Try saying
", Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

@Override
public void onPartialResults(Bundle partialResults) {

}

```

```

@Override
public void onEvent(int eventType, Bundle params) {

}

});

```

```

parentLinear.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                speechRecognizer.startListening(speechRecognizerIntent);
                keeper = "";
                break;

            case MotionEvent.ACTION_UP:
                speechRecognizer.stopListening();
                break;
        }
        return false;
    }
});

```

```

VoiceEnabledButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (mode.equals("ON")) {
            mode = "OFF";
            VoiceEnabledButton.setText("Voice Enabled Mode - OFF");
            linLower.setVisibility(View.VISIBLE);
        } else {
            mode = "ON";
            VoiceEnabledButton.setText("Voice Enabled Mode - ON");
            linLower.setVisibility(View.GONE);
        }
    }
});

```

```

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

```

```

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

        if (myMediaPlayer.getCurrentPosition() > 0) {
            playPreviousSong();
        }
    }
});

```

```

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

```

```

        if (myMediaPlayer.getCurrentPosition() > 0) {
            playNextSong();
        }
    }
});
}

```

```

private void validateReceiveValueAndStartPlaying() {
    if (myMediaPlayer != null) {
        myMediaPlayer.stop();
        myMediaPlayer.release();
    }
}

```

```

Intent intent = getIntent();
Bundle bundle = intent.getExtras();

```

```

mySongs = (ArrayList) bundle.getParcelableArrayList("song");
mSongName = mySongs.get(position).getName();
String songName = intent.getStringExtra("name");

```

```

SongName.setText(songName);
SongName.setSelected(true);

```

```

position = bundle.getInt("position", 0);
Uri uri = Uri.parse(mySongs.get(position).toString());

```

```

myMediaPlayer = MediaPlayer.create(PlayerActivity.this, uri);
myMediaPlayer.start();
}

```

```

private void checkVoiceCommandPermission() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {

```

```

        if (!(ContextCompat.checkSelfPermission(PlayerActivity.this,
Manifest.permission.RECORD_AUDIO) == PackageManager.PERMISSION_GRANTED))
    {
        Intent intent = new
Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS, Uri.parse("package:" +
getPackageName()));
        startActivity(intent);
        finish();
    }
}
}

```

```

private void playPauseSong() {
    cimageView.setBackgroundResource(R.drawable.four);

    if (myMediaPlayer.isPlaying()) {
        PausePlay.setImageResource(R.drawable.play);
        myMediaPlayer.pause();
    } else {
        PausePlay.setImageResource(R.drawable.pause);
        myMediaPlayer.start();
        cimageView.setBackgroundResource(R.drawable.five);
    }
}

```

```

private void playNextSong() {
    myMediaPlayer.pause();
    myMediaPlayer.stop();
    myMediaPlayer.release();

    position = ((position + 1) % mySongs.size());
    Uri uri = Uri.parse(mySongs.get(position).toString());
    myMediaPlayer = MediaPlayer.create(PlayerActivity.this, uri);
    mSongName = mySongs.get(position).toString();
}

```

```

        SongName.setText(mSongName);
        myMediaPlayer.start();
        cimageView.setBackgroundResource(R.drawable.three);

        if (myMediaPlayer.isPlaying()) {
            PausePlay.setImageResource(R.drawable.pause);
        } else {
            PausePlay.setImageResource(R.drawable.play);
            cimageView.setBackgroundResource(R.drawable.five);
        }
    }

    private void playPreviousSong() {
        myMediaPlayer.pause();
        myMediaPlayer.stop();
        myMediaPlayer.release();

        position = ((position - 1) < 0 ? (mySongs.size() - 1) : (position - 1));
        Uri uri = Uri.parse(mySongs.get(position).toString());
        myMediaPlayer = MediaPlayer.create(PlayerActivity.this, uri);
        mSongName = mySongs.get(position).toString();
        SongName.setText(mSongName);
        myMediaPlayer.start();
        cimageView.setBackgroundResource(R.drawable.two);

        if (myMediaPlayer.isPlaying()) {
            PausePlay.setImageResource(R.drawable.pause);
        } else {
            PausePlay.setImageResource(R.drawable.play);
            cimageView.setBackgroundResource(R.drawable.five);
        }
    }
}

```

```

@Override
protected void onStop() {
    super.onStop();
    myMediaPlayer.pause();
    myMediaPlayer.stop();
}
}

```

6.1 METHOD

the voice recorded in the system. Once the voice is recorded, it can go through various processes and be processed. The voice is digitized to perform these operations. The voice is first filtered and then sampled for digitization. The voice wave that forms the sound has two important features. These properties are amplitude and frequency. Frequency, while determining the soundness and quiver characteristics of voice; amplitude determines the intensity of the voice and the energy it carries. Equation 1 is given for the Total Amplitude (TG) calculation. $TG = \sum_{t=1}^n x(t)$ (1) In this equation $x(t)$; amplitude at time t ; In other words, it expresses the energy carried by the voice wave at the moment t . If the sum of the total amplitude value calculated by this method is above a certain value, then the meaning of sound, that is, speech, is started. Filters are used for two purposes in the processing of voice. These are the separation of the voice signal and the correction of the voice signal. Digital filters are FIR (Finite Impulse Response) filter and IIR (Infinite Impulse Response) filter. In FIR filters, the input signal forms the output y_n , which is the weighted sum of the current and previous inputs versus x_n . The mathematical expression of this filter is given by Equation 2. $y_n = b_0x_n + b_1x_{n-1} + b_2x_{n-2} + \dots + b_qx_{n-q}$ (2) In this equation y_n is the result of the filter output. In IIR filters, the input signal constitutes the output y_n , which represents the weighted sum of the previous outputs, together with the weighted sums of the current and previous inputs versus x_n . In this model, together with the x_n input, the weighted sum of the previous

p

outputs gives the filter output y_n . After digitizing the voice, the voice is encoded and the voice recognition process is completed. The following libraries should first be added to the system for voice recognition.

7.RESULTS



15:34 



Voice Based Music Player



/storage/emulated/0/Download/[iSongs.info]
01 - Maguva Maguva.mp3

Command = play the next song

DISCUSSION:

Using technology today is of great importance in terms of making people's lives easier. It has become very easy to run some applications with technology. In this study, an application that provides media player control with voice commands was developed. This application was developed to address the needs of people who cannot listen to music on their own due to any disability. The application was implemented in java language. In order to manage the media player with voice commands, voice recognition libraries were first used. In the developed application, operations with keyboard and mouse can be done with voice commands. Voice commands can be sent with the wireless headset from anywhere in the shooting area.

8. CONCLUSION:

As the pronunciation of the voice command becomes more difficult and the number of letters in it increases, the level of accurate voice recognition decreases. 100% accurate recognition can be achieved by using short words and words with full pronunciation when making voice definitions. With misrecognition, the voice command performs the function linked to the nearest voice command. No action can be taken with inability to identify. In this study, test procedures were performed with 20 people. In some word tests, more than one test was performed over the same person's voice. The following in Table 2 shows the results of the experimental studies. As can be seen from the results, it was more difficult to identify words with a high number of words and difficult to pronounce. The application has a coding that can do everything we do about daily media player with voice commands. It is thought that the application will be useful for people who cannot use the computer for any reason (bedridden, elderly, disabled, etc.). Specially developed to facilitate the lives of the visually impaired. With the application you can meet your daily music listening needs without being connected to anyone. In this study, media player control was provided to listen to music by remote voice commands. Voice commands can be sent from any point with a wireless or wired headset. Media player was managed with voice commands without using mouse and keyboard. In addition, this study will enable people with disabilities, elderly or bedridden patients to meet

their listening needs. The application developed in java using the Speech.dll library was tested with different voice commands.

REFERENCES:

[1]. Karakaş M., “Computer Based Control Using Voice Input”, Master Thesis, Dokuz Eylül University, (2010).

[2]. Muda L., Begam M., Elamvazuthi I., "Voice Recognition Using Mel Frequency Cepstral Coefficient and Dynamic Time Warping Techniques", Journal of Computing, 2(3):138-143, (2010).

[3]. Baygün M. K., Yaldir A. K., “Linear Predictive Coding ve Dynamic Time Warping Teknikleri Kullanılarak Ses Tanıma Sistemi Geliştirilmesi”, Pamukkale University, (2009).

[4]. Öztürk B., Çakar T., “Gerçek Zamanlı Ses Tanıma”, Graduation Project, Istanbul University Faculty of Engineering Department of Electrical/Electronics Engineering, (2007).

[5]. Demirci M. D.,” Bilgisayar Destekli Ses Tanıma Sistemi Tasarımı”, Master Thesis, Istanbul University Institute of Science and Technology, (2005).

[6]. “Bilgisayar destekli bir dil programı” [Online]. Available: <https://docplayer.biz.tr/16256021-Bilgisayar-destekli-birdil-programi-turkce-konusma-tanima-sistemi.html> [Accessed: 19-Sep-2019].

[7]. Aşlıyan R., Günel K., Yakhno T., ”Dinamik Zaman Bükmesi Yöntemiyle Hece Tabanlı Konuşma Tanıma Sistemi”, Çanakkale Onsekiz Mart University, Academic Informatics, (2008).

[8]. Meral O., “Doğrusal Öngörülü Kodlama ve Adaptif Algoritma Tabanlı Konuşmacı Tanıma”, Master Thesis, Istanbul University Institute of Science and Technology, (2008).

[9]. Dede G., Sazlı,M.H., “Biyometrik Sistemlerin Örüntü Tanıma Perspektifinden İncelenmesi ve Ses Tanıma Modülü Simülasyonu”, Institute of Defense Sciences.

[10].Bolat B., Küçük Ü., Yıldırım T., “Aktif Öğrenen PNN ile Konuşma/Müzik Sınıflandırma”, Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu, (2004).