9-2015

Developing Java Programs on Android Mobile Phones Using Speech Recognition

Santhrushna Gande
California State University - San Bernardino, 004580341@coyote.csusb.edu

DEVELOPING JAVA PROGRAMS ON ANDROID MOBILE PHONES USING SPEECH RECOGNITION

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Santhrushna Gande

September 2015

DEVELOPING JAVA PROGRAMS ON ANDROID MOBILE PHONES USING SPEECH RECOGINITION

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Santhrushna Gande

September 2015

Approved by:

ABSTRACT

Nowadays Android operating system based mobile phones and tablets are widely used and had millions of users around the world. The popularity of this operating system is due to its multi-tasking, ease of access and diverse device options.

"Java Programming Speech Recognition Application" is an Android application used for handicapped individuals who are not able or have difficultation to type on a keyboard. This application allows the user to write a compute program (in Java Language) by dictating the words and without using a keyboard. The user needs to speak out the commands and symbols required for his/her program. The program has been designed to pick up the Java constant keywords (such as 'boolean', 'break', 'if' and 'else'), similar to the word received by the speech recognizer system in the application.

The "Java Programming Speech Recognition Application" contains external plug-ins such as programming editor and a speech recognizer to record and write the program. These plug-ins come in the form of libraries and pre-coded folders which have to be attached to the main program by the developer.

ACKNOWLEDGEMENTS

I am indebted to my advisor, mentor and supporter Dr. Tong Lai Yu for providing me not only with guidance and knowledge during this project, but with invaluable advice. I also would like to thank Dr. Owen J. Murphy and Dr. Haiyan Qiao for their valuable advice and unlimited support.

I owe special thanks to my family and friends for their beautiful advice and support which encourage me to stand stronger in the road of life. For having them I am extremely grateful.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Background

Mobile applications are software applications developed for mobile devices such as smart phones and tablets. These applications extend the functionality of the device and allow user to use the phone or tablet for different purposes and in a user friendly environment. Mobile applications are often written in Java Programming Language, different frameworks will make them functional for different operating systems such as Android Operating System and Apple Operating System.

Android Operating System is a mobile platform and is based on the Linux Kernel developed by Google and a consortium of companies. Android was primarily designed for touch screen mobile devices which were a revolution in the technology industry. Nowadays, Android is widely used by different mobile, tablet and computer companies. Due to the popularity of this operating system, millions of applications have been developed for its users around the world.

Speech Recognition Application is an Android Application, which converts spoken programming commands and sentences into text. This application can help programmers to develop their program faster and also give the chance to handicapped individuals who are unable to type, write their own program.

Speech Recognizer recognizes Java commands and libraries and compiles the program accurately.

Purpose

Android has various applications which convert speech to text, as well as different editors which allow users to write a computer program in a user friendly environment. An example of Speech converter is Google speech recognizer which is used in google voice application. Deuter IDE, Code Peeker and AIDE are examples of programming editors for android operating system.

Existing System

Problems in the Existing System

A typical existing speech recognition system converts speech to text irrespective of the content. This means that the content can be anything other than programming commands and syntaxes. For writing the program the user must follow the particular format and syntax, which is not possible through the existing application.

Improved version of this application should recognize Java syntaxes and do not allow any other words outside the language libraries. Also the program must allow a user to leave a comment, by dictating “comment” followed by the user comment.

CHAPTER TWO

SYSTEM ANALYSIS

Proposed System

The Speech Recognition Application allows users to develop their own Java program by following the particular Java syntax and dictating the code to the application. The application will convert the spoken words to text and after saving the program, it will import the file to a compiler and execute the program.

While using this application the user has to start the program with class name followed by elements and methods declaration. The user can import packages and libraries at any stage of his/her program.

The Speech Recognition Application makes the process of programming even easier by its prepared statements and blocks. For instance, the user doesn't need to dictate every single basic component of the main method, but only needs to say the word "main". The application will open a basic "main class" including brackets and empty spaces for arguments. After the program is complete, the user can dictate the word "commit" which will save and compile the program.

System Requirement Specification

Hardware Requirements

- Android mobile of version 2.2 minimum.
- Processor should not be less than 500MHZ.

- RAM should not be less than 170MB.
- SD card of minimum 512 MB.
- Device should be enabled for USB debugging.

Software Requirements

- Android Mobile Operating System of version 2.2 or later.
- IDE tools: Eclipse or Android Studio.
- User Interface :XML.
- Code Behind: JAVA and XML.
- Internet: Yes.

CHAPTER THREE

SELECTED SOFTWARE

What is Android?

Android was founded in Palo Alto, California in October 2003 by Andy Rubin, Rich Miner, Nick Sears and Chris White. As Google acquired Android on August 17, 2005, people thought that Google was entering the mobile phone market. Google produced a mobile device which was powered by Linux kernel, by team Rubin. Google promised to provide a flexible, upgradable system platform for the users, and that’s when Android got updated and reached todays advancement.

Android is a mobile operating system designed for touch screen mobile phones, tablets and computers. It is also used in game consoles, regular PCs and other electronics. Android is the most widely used and even highest selling OS overall. According to 2013 sales, android devices, sold more than Microsoft Windows, iOS and Mac OS. As of July 2013, Google play store had published over one million Android applications and downloaded over 50 billion applications. According to the survey conducted in May 2013, they found that 71\% of mobile developers are developing for android. Currently, there are over one billion active Android users [12].

The software tools used for developing android devices are open source and proprietary software developed by Google. Android has become popular because of its zero-cost features and customizable operating system for high-tech devices. Android’s open-source nature allows developers to use the open-source code as a base for community-driven projects. They can even add features for advanced users.

Google Speech Recognition

<u>Definition of Voice Recognition</u>

“Speech Recognition (SR) by machine, which translate spoken words into text has been a goal of research for more than six decades. It is also known as automatic speech recognizer (ASR), computer speech recognition, or simply speech to text (STT). The research in speech recognition by machine involves a lot of disciplines, including signal processing, acoustics, pattern recognition, communication and information theory, linguistics, physiology, computer science and psychology” [11].

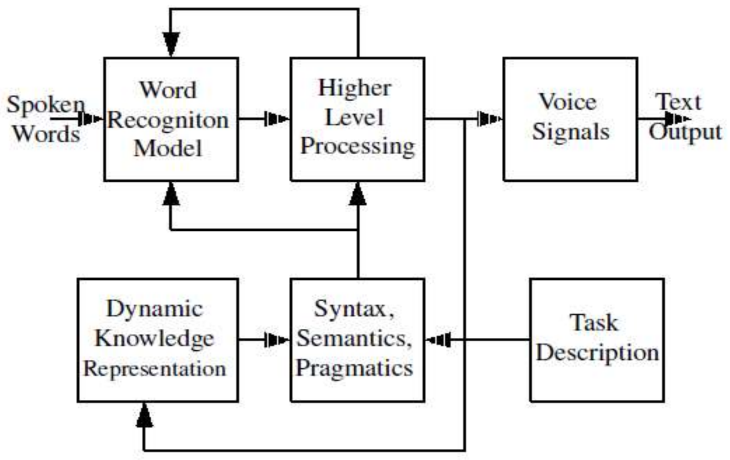


Figure 1. Typical Speech Recognition System [11].

At the present time, mobile products of Speech recognition (SR) are pervasive. There are numerous third party SR apps that support android. We have chosen Google Voice Recognition (GVR), which is preinstalled in many android device as our recognition engine [11].

“GVR makes use of neural network algorithm to convert human audio speech to text and works for a number of major languages but we only need to use English in our application. A neural network consists of many processors working in parallel, mimicking a virtual brain”. In real time environment for better operation and more computing power parallel processors are used. Neural network is unique because of its ability to adapt and acquire based on existing data. In general, no particular algorithm will be used by neural network to achieve its own task, it acquires from the example of alternative data. Though GVR may work in some Android phones offline, it normally accesses through Internet its huge database for voice recognition experimented by former users. It also glance at earlier google search queries, it helps to voice recognition engine to estimate which phrases are frequently used than others [11].

Generally, a neural network can learn from two main classification of learning methods supervised or self-organized. In supervised training, an external teacher contribute labelled data and the required output. Simultaneously, self-organized network holds enabled data and finds groups, patterns in the data on its own. GVR learns from its individual database over the self-organized method [11].

Since Android client involves a few tasks, including interfacing to the user, accepting text from the GVR engine, and communicating with the server.   The GVR itself also has to connect to the Internet through Wi Fi to interact with Google cloud database. The execution time for each task is never a constant. In particular, the bandwidth of Wi Fi communication can fluctuate widely, depending on the traffic of the environment [11].

Introduction to Libraries

Google voice recognizer is an application developed by Google Company. This application has been widely used in Google PC products and Android Operating Systems. This application enables us to type with our voice, meaning by dictating the words, the application will receive the input, analyze it and convert it to text.

Google voice recognizer was a revolutionary application at its own time and has brought many new functionalities to Android systems. For example, the device user can dictate a sentence for google search and the application will receive the input, convert it to text, perform google search and show the result to the user.

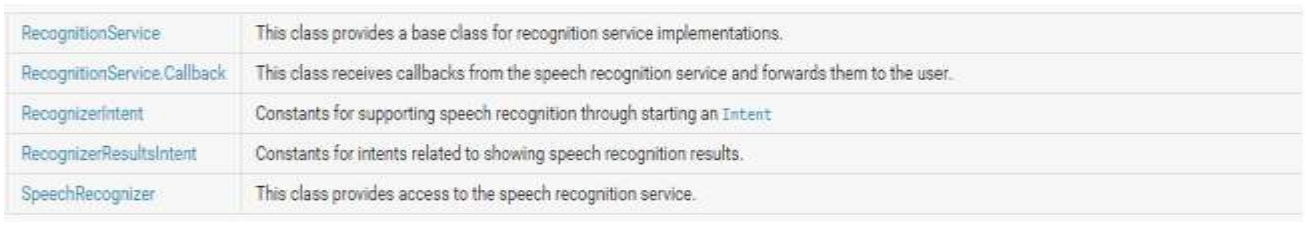
Nowadays many application developers use speech recognizer in their applications for more functionality. To make it easier for the application developers, android added the pre-defined APR for speech recognizer into its library. In this way, the developers only need to add the library into their application and call the right function and method in their java class.

In order to write an offline voice recognizer which recognize the dictated word and convert it to text, importing below JAR file into the java class is recommended [2].

```
import android.speech.RecognitionListener;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.ToggleButton;
```

Package Definitions

Android.speech package. This package contains one interface (RecognitionListener) and five classes (RecognitionService, RecognitionService.Callback, RecognizerIntent, RecognizerResultsIntent and SpeeckRecognizer) [2].



RecognitionService	This class provides a base class for recognition service implementations.
RecognitionService.Callback	This class receives callbacks from the speech recognition service and forwards them to the user.
RecognizerIntent	Constants for supporting speech recognition through starting an `Intent`
RecognizerResultsIntent	Constants for intents related to showing speech recognition results.
SpeechRecognizer	This class provides access to the speech recognition service.

Figure2. Android.speech Classes

Android.app package. "This package contains high-level classes encapsulating the overall Android application model. An application can be defined by using one or more Android's four core application components. Activity and Service components are defined in this package. An activity can start other activities, including activities that live in separate application. A Service is an application component that provides a screen with which user can interact in order to perform and action. For instance, a service can handle playing music, network connection or work with browser without the user being aware of the work going on" [2].

Android.Widger package. "To create your own widget, extend View or a subclass. To use your widget in layout XML, there are two additional files for you to create. Here is a list of files you'll need to create to implement a custom widget" [4] [2].

Java Implementation file. "This is the file that implements the behavior of the widget. If you can instantiate the object from layout XML, you will also have to code a constructor that retrieves all the attribute values from the layout XML file" [4] [2].

XML Definition file. “An XML file in res/values/ that defines the XML element used to instantiate your widget, and the attributes that it supports. Other applications will use this element and attributes in their in another layout XML” [4] [2].

Layout XML. “An optional XML file inside res/layout/ that describes the layout of your widget. You could also do this in code in your Java file. API Demos sample application has an example of creating a custom layout XML tag, Label View. See the following files that demonstrate implementing and using a custom widget” [4] [2].

Label View java class: The Implementation file.

Res_values_attrs.xml—Definition file.

Re/layout/custom_view_1.xml—Layout file.

Android.view package. “Provides classes that expose basic user interface classes that handle screen layout and interaction with the user” [4] [2].

Principle and Algorithm of Google Speech Recognizer

Modern Speech Recognition system algorithms are based on two important part:

- Acoustic modeling
- Language modeling

Below algorithms are the basic algorithms used in a speech recognition system:

Hidden Markov Models (HMMs). Modern speech recognition systems use Hidden Markov Models which is the statistical model that output a collection of symbols or quantities. These models are used in speech Recognition systems because it can receive a speech signal as a piecewise stationary short signal or a short-time stationary signal and approximate it as a stationary process in a very short time period (around 10 milliseconds) [7].

Another reason that Hidden Markov models are used in speech recognizers is that they can learn automatically and are easy to use [7].

Dynamic Time Wrapping (DTW)-based Speech Recognition. Dynamic Time Warping (DTW) is a type of algorithm used for measuring between two time process which may vary in time or speed. This system was traditionally used in speech recognition system but recently in new systems it has been replaced with a successful HMM [7].

Neural Networks. Neural networks is a type of algorithm used for estimating or approximating functions that can depend in a large number of inputs [7].

Deep Natural Networks and other Deep Learning Models. A DNN is an artificial neural network with different layers which are hidden. These layers are between the user input and system output. Deep Natural Networks can model complicated non-linear relationships [7].

CHAPTER FOUR

SYSTEM DESIGN

Data Flow Diagrams

A Data Flow Diagram (DFD) is a graphical method to present the data flow in a program. DFD represents the overview of the system and the structure analysis. This method gives the detailed description of the system components and also can provide a high level system overview. Previously, programming architectures would use flow charts and pseudo code to represent the system. A DFD explains the function which must be performed in a program and to the data functions they need them.

UML Diagrams

Use Case Diagram

This diagram shows the four most important steps in this Java Developing application. In this short scenario, the user will receive some Speech Hints from the application, these hints are the result of data check in the voice recorder system in android. If the voice recorder has no result yet, the application will start giving some speech hints.

The next step is when the user will start giving the commands. By starting to speak, the voice recorder will be activated automatically and will start receiving inputs from the microphone.

Upon receiving some results from the voice to text converter (voice to text converter is a predefined and coded program by google Speech Recognition. The current application uses libraries and extensions to use this API) the program will start searching for the match word in its Java library and web search. If any match was found, we will see the resulted match on the screen.

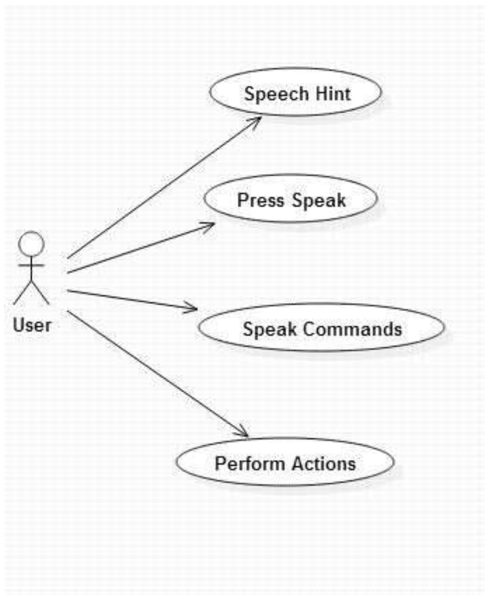


Figure 3. Use Case Diagram

Sequence Diagram

This sequence diagram shows the activity in the MainActivity class where all the process flow is placed. In order to execute a command, the application should complete 6 steps and flow from one method to the next method.

onCreate() method is the step where the Adapter is created and if there has been an Adapter created in the last cycle, it will clear it. Next stage is when the application checks if the Voice Recognizer has started and works properly.

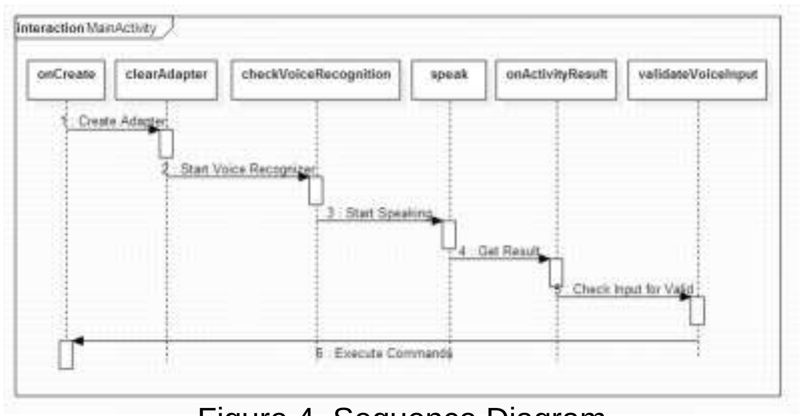


Figure 4. Sequence Diagram

Class Diagram

The class Diagram shows the number of classes that exist in this application and their existing methods. It also it shows the relationship between different classes in different packages.

MainActivity and JavaConstants are two essential classes in this application where we stored the constants of java programming (for word guessing) and complete process of application flow.

MainActivity contains the main variable, methods and objects. This uses the data in JavaConstants class in order to check for validation of received data from Voice Recorder.

Database Helper is another class in different package which connects to different libraries and pass the result to the JavaConstants class.

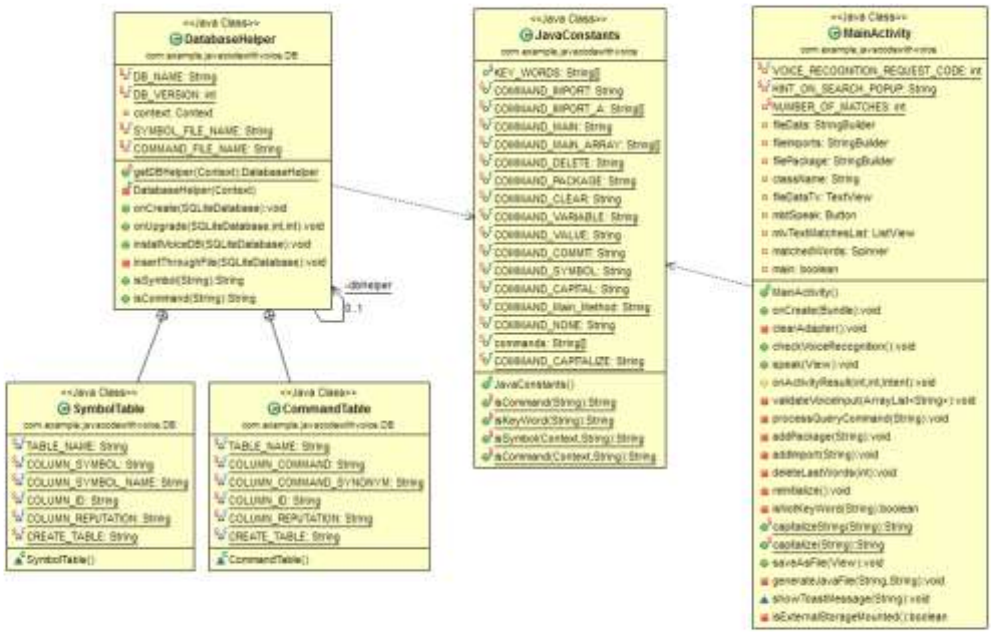

<<Java Class>>
DatabaseHelper
com.example.javacodewithvoice.DB
DB_NAME: String
DB_VERSION: int
context: Context
SYMBOL_FILE_NAME: String
COMMAND_FILE_NAME: String
getDBHelper(Context):DatabaseHelper
DatabaseHelper(Context)
onCreate(SQLiteDatabase):void
onUpgrade(SQLiteDatabase,int,int):void
installVoiceDB(SQLiteDatabase):void
insertThroughFile(SQLiteDatabase):void
isSymbol(String):String
isCommand(String):String
-dbHelper
0..1
<<Java Class>>
SymbolTable
com.example.javacodewithvoice.DB
TABLE_NAME: String
COLUMN_SYMBOL: String
COLUMN_SYMBOL_NAME: String
COLUMN_ID: String
COLUMN_REPUTATION: String
CREATE_TABLE: String
SymbolTable()
<<Java Class>>
CommandTable
com.example.javacodewithvoice.DB
TABLE_NAME: String
COLUMN_COMMAND: String
COLUMN_COMMAND_SYNONYM: String
COLUMN_ID: String
COLUMN_REPUTATION: String
CREATE_TABLE: String
CommandTable()
<<Java Class>>
JavaConstants
com.example.javacodewithvoice
KEY_WORDS: String[]
COMMAND_IMPORT: String
COMMAND_IMPORT_A: String[]
COMMAND_MAIN: String
COMMAND_MAIN_ARRAY: String[]
COMMAND_DELETE: String
COMMAND_PACKAGE: String
COMMAND_CLEAR: String
COMMAND_VARIABLE: String
COMMAND_VALUE: String
COMMAND_COMMIT: String
COMMAND_SYMBOL: String
COMMAND_CAPITAL: String
COMMAND_Main_Method: String
COMMAND_NONE: String
commands: String[]
COMMAND_CAPITALIZE: String
JavaConstants()
isCommand(String):String
isKeyWord(String):String
isSymbol(Context,String):String
isCommand(Context,String):String
<<Java Class>>
MainActivity
com.example.javacodewithvoice
VOICE_RECOGNITION_REQUEST_CODE: int
HINT_ON_SEARCH_POPUP: String
NUMBER_OF_MATCHES: int
fileData: StringBuilder
fileImports: StringBuilder
filePackage: StringBuilder
className: String
fileDataTv: TextView
mbtSpeak: Button
mlvTextMatchesList: ListView
matchedWords: Spinner
main: boolean
MainActivity()
onCreate(Bundle):void
clearAdapter():void
checkVoiceRecognition():void
speak(View):void
onActivityResult(int,int,Intent):void
validateVoiceInput(ArrayList<String>):void
processQueryCommand(String):void
addPackage(String):void
addImport(String):void
deleteLastWords(int):void
reinitialize():void
isNotKeyWord(String):boolean
capitalizeString(String):String
capitalize(String):String
saveAsFile(View):void
generateJavaFile(String,String):void
showToastMessage(String):void
isExternalStorageMounted():boolean

Figure 5. Class Diagram

Activity Diagram:

An Activity Diagram is a simple diagram which shows the user's journey from the time he/she starts the application until the time he finishes a complete successful cycle. This journey gives the big picture of how the Main Activity class flow the process.

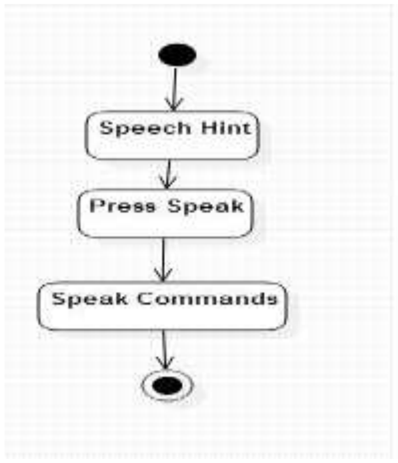


Figure 6. Activity Diagram

CHAPTER FIVE

APPLICATION CODES

Java Code with Voice

Java Code with Voice is the main folder of this application containing the below sub-folders:

- Android 5.1.1 Library
- Android Dependencies
- Android Private Library
- Src
- Gen[Generated Java File]
- Assets
- Bin
- Res
- Libs

Android 5.1.1 Library

This is a set of codes provided by Android for developer who wish to develop an Android application. Each time a developer wants to develop an android application, must add this library to his application.

Basically, Android support library is a set of codes in the form of JAR file. This library package provides backward-compatible version of Android framework APIs as well as features that are only available through the library

APIs. In order to use the Android support library for an Android application, it is important to know which feature OS supports in that particular library.

This library has a wide use in different Android applications but one of the major uses of this library in this application is the android.speech package. This package acts as the speech recognizer in this application and provides access to the speech recognizer service in Android.

Android.speech package consist of Constants(CONFIDENCE SCORES, ERROR_ AUDIO , ERROR, CLIENT and etc), public methods (static SpeechRecognizer, static Boolean and etc) and inherits methods from java.lang.object.

Including the Android support library in this project allow the application to be compatible in Android 5.1.1 and below versions as we have used Android 5.1.1 library.

Android Dependencies

Android Dependencies is a virtual folder where android projects properties are stored in the form of JAR files. When an application references more than one libraries that each required its own JAR file, the build system has to detect and resolve the duplication. Android dependencies libraries would associate each JAR file with a full qualified name and a version number to figure out which version has to be used.

Android Private Library

This library consist of two JAR files:

- Android-support-v7-appcompat.jar
- Android_ support-v4

These JAR files are the android supporting libraries used at the time of running the application on lower version than android 5.1.1.

For instance if the application is using Android API 19 as the targeted library for compiling, it may need the android_ support-v4 library to make it runnable on lower Android APR devices.

The application has two different support JAR files, v7 and v4. The difference between these two is that v7 supports Android version 3.0 and above, and v4 supports Android version 2.0 and above.

Android Private Libraries and the Android Dependencies are not actual folders and they will be created by the development tools such as Eclipse. The difference between Android Private Libraries and Android Dependencies Libraries is that the Private Libraries just has the reference to the support libraries and Dependencies Libraries tells the application, which appcompat_ v7_ x is the project referencing or using.

<u>Src Folder</u>

src Folder contains the main java classes which were developed by the application developer. The hierarchy of src Folder is as per below:

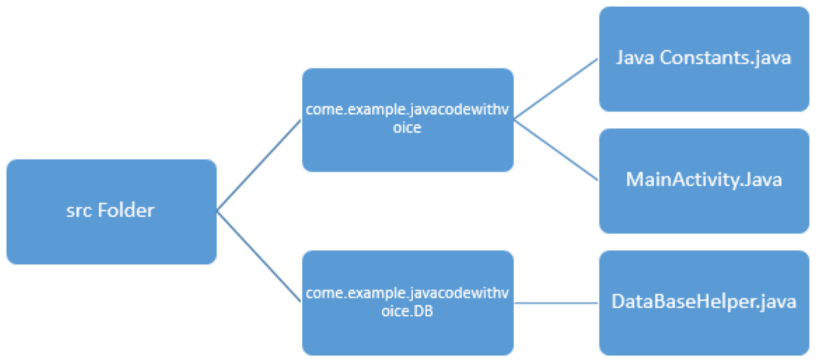


Figure 7. src Folder

Java Constants.java

“Java Constants” is a class which clarify java programming constants such as “abstract”, “assert”, “Boolean”, “break” and etc. In this class we are creating a structure for the text received by speech recorder. If the word that has been dictated is anything between these words, the application will take the action and create the requested field of programming. This class also give other probability of word guess by the speech recognizer. At some cases the word that is dictated and the word that is guessed by the Speech Recognizer can have slight different in spelling.

For example, if the user dictates the word “main”, the correct guess of the “main” word should appear but sometimes due to the user accent, distance from microphone, or other reasons, the Speech Recognizer guesses the wrong word (for example “mean” or “mein”).

The “Java Constants” class has predicted a few options of wrong word guess by the Speech Recognizer and created a condition for them. Have a look at below:

```
public static final String COMMAND_MAIN = "main";
public static final String COMMAND_MAIN_ARRAY[] = { "main", "maine",
"mean","meen","mein" };
```

These values will ask the main method to recognize other words such as the words mentioned in COMMAND_ MAIN_ ARRAY and accept it as the “main”

Other methods used in the java class are “isKeyword” method, “Symbol” method and “IsCommand” method which check the database, choose the right keyword, symbol or command and return it to the main method.

```
public static String isCommand(Context context, String commandI) {
                DatabaseHelper dbHelper =
DatabaseHelper.getDBHelper(context);
                String command = dbHelper.isCommand(commandI);
                dbHelper.close();
                dbHelper = null;
                return command;
        }
```

MainActivity.java

MainActivity is a class which extends Activity class and represents the main activities and link all parts of the program together. Below is the method flowchart of the MainActivity Class:

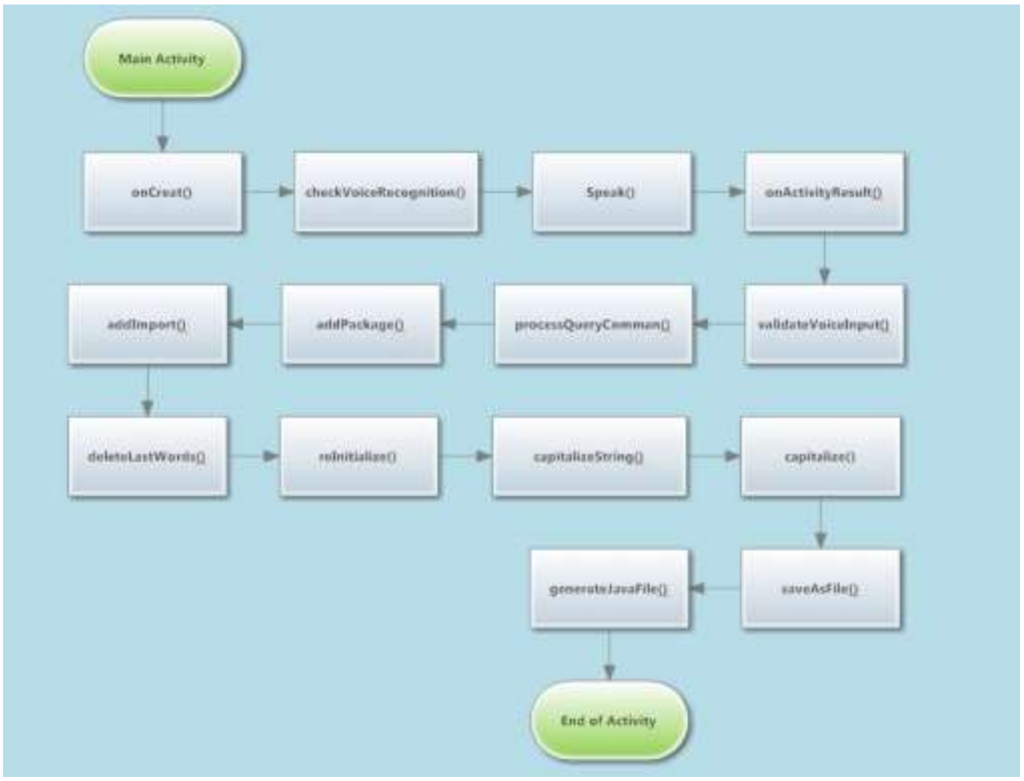


Figure 8. Main Activity Method Flow

Oncreate() method. onCreate() is where we initialize the activities and static setup. This method also provides the Bundle containing the activityâ€™s previous frozen state, if there is one.

CheckVoiceRecognition() method. This method will check if the voice recognition is present by using PackageManager class which is a Class for retrieving various kinds of information related to the application packages that are currently installed on the device. This class is located in getPackageManager().

The method checks if any speck has been prompted via ‘ACTION_ RECOGNIZE_ SPEECH’ and by checking the ‘activities’ array list. Here is when the if condition comes to the picture. The ‘if’ condition will check if the size of the ‘activities’ array is equal to zero.

If ‘yes’, then the program display a message that “voice recognizer not present” and if ‘NO’ the compiler moves to the next method which is speak () method.

```
public void checkVoiceRecognition() {
PackageManager pm = getPackageManager();
List<ResolveInfo> activities = pm.queryIntentActivities(new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);
if (activities.size() == 0) {
mbtSpeak.setEnabled(false);
fileDataTv.setText(filePackage.toString() + "\n"+ fileImports.toString() + "\n"
+ "Voice recognizer not present");
}
}
```

Speak() method. The result of this method is starting the voice recognizer activity. The journey of this method starts by specifying the calling package in order to identify the application. At the next step, the application will display a hint to the user about what he should say, also it gives the recognizer a hint about what the user want to say.

There are 2 types of language models available:

1. LANGUAGE_ MODEL_ WEB_ SEARCH (this model is used for short phrases)
2. LANGUAGE_ MODEL_ FREE_ FORM (this model is based on free-form speech recognition and used if the recognizer is not sure about the word or phrase)

EXTRA_ MAX_ RESULTS will specify how many results the user wants to receive. The result will be stored. The result will be listed starting with the most confidence result.

StartActivityForResult will star the Voice Recognizer activity and if VOICE_ RECOGNITION_ REQUEST_ CODE get some results (=>0) this code will move to the next method which is onActivityResult().

```
public void speak(View view) {

System.out.println(getFilesDir().getPath());

Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE, getClass()
.getPackage().getName());
intent.putExtra
(RecognizerIntent.EXTRA_PROMPT,HINT_ON_SEARCH_POPUP);intent.putExt
ra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,RecognizerIntent.ACTION_R
ECOGNIZE_SPEECH);
intent.putExtra
(RecognizerIntent.EXTRA_MAX_RESULTS,NUMBER_OF_MATCHES);
startActivityForResult(intent, VOICE_RECOGNITION_REQUEST_CODE);
}
```

OnActivityResult () method. This method checks if the voice recognition is successful by using RESULT_ OK API which stands for successful operations. In case of positive result and the first match contains the “search” word, the application will start searching for the correct word match in the web search until the match word is found. If the process was successful and the match was found, the else condition will pass the compiler to the next method which is validateVoiceInput() method.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
if (requestCode == VOICE_RECOGNITION_REQUEST_CODE)
if (resultCode == RESULT_OK) {
ArrayList<String> textMatchList =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
```

```
if (!textMatchList.isEmpty()) {
if(textMatchList.get(0).contains("search")) {
String searchQuery = textMatchList.get(0);
searchQuery = searchQuery.replace("search", "");
Intent search = new Intent(Intent.ACTION_WEB_SEARCH);
search.putExtra(SearchManager.QUERY, searchQuery);
startActivity(search);
} else {
validateVoiceInput(textMatchList);
}
}
```

In this method we have identified various error cases and tried to prepare the constructive action for them. For example if an audio error was encountered “Audio Error” message will be displayed: If there is a generic client error, “Client Error” will be shown. Other examples of errors are “Network Error”, “No Match Error” and “Server Error”.

```
else if (resultCode == RecognizerIntent.RESULT_AUDIO_ERROR) {
                                showToastMessage("Audio Error");
} else if (resultCode == RecognizerIntent.RESULT_CLIENT_ERROR) {
                                showToastMessage("Client Error");
} else if (resultCode == RecognizerIntent.RESULT_NETWORK_ERROR) {
                    showToastMessage("Network Error");
            }
else if (resultCode ==RecognizerIntent.RESULT_NO_MATCH) {
                    showToastMessage("No Match");
} else if (resultCode == RecognizerIntent.RESULT_SERVER_ERROR) {
                    showToastMessage("Server Error");
            }
}
```

ValidateVoiceInput() method. This method has an array list (textMatchList) which is the result of the successful word match in onActivityResult() method. The entire code in this method checks the components in the array list and with the help of these component, it write a correct Java Programming sentence. This method checks the ClassName, symbols and non-constant words.

CHAPTER SIX

SYSTEM TESTING

Testing is the software quality assurance and an important part of the project. Testing represents the overall review of specification, design and coding. It is also one of the step in software engineering process.

Testing involves software test case design methods into series of steps which results in successful development of software. Testing is planned before and executed systematically.

Strategic Approach to Software Testing

The software engineering process can be considered as spiral pattern. Firstly, the developer must define the role of software and then lead to software requirement analysis. At the second step, the developer should design the software and then finally do the coding. Unit testing begins at initial stage of the spiral model and focus on each unit of the software implemented in source code. Then next comes the integration testing which focuses on the design and construction of software. Validation testing also comes across for testing the software constructed; and at last, system testing begins which involves testing every element of the system and one general test of the whole unit.

Unit Testing

Testing is a process of examining every unit of a software. The objective of testing is to ensure that the entire software works properly and the software is being responsive to all the possible data inputs. Testing helps the developer to find the weaknesses and bugs in the application and correct those problems before a bad user experience occur.

Testing has different levels:

- Unit Testing
- Module Testing
- Integration Testing
- System Testing
- User Acceptance Testing

Unit Testing

Unit testing is for testing the logic and design of the application or software. This level of testing ensure that the application/software is functioning properly and the code flow is smooth. In another word, the application flow from one method to the next method in the main class without failing to identify even one step of user input.

Module Testing

Module testing involves testing each component of an application/software. Meaning the tester must check every function separately. For instance if we are

Module testing a microwave, we should examine the clock, Timer, cooking control, oven control and etc. separately and make sure each unit function properly.

Integration Testing

Integration testing is when we developer should understand every possible failure situation and test it with his/her software and make sure that the software has a backup plan to cover that situation. This type of testing gives the developer the vision to contemplate all possible errors.

System Testing

System testing makes sure that the software or application meets the system requirement meaning it examines all the predictable requirements of a user. System testing does not require specific knowledge about coding or logics behind the code.

User Acceptance Testing

User Acceptance testing is an important level of testing which requires the developer to have significant participation with the end user and ensure that the software/application meets the user functional requirements and it satisfies the user ease of access and control in every single part of the application/software.

Testing Performed For My Application

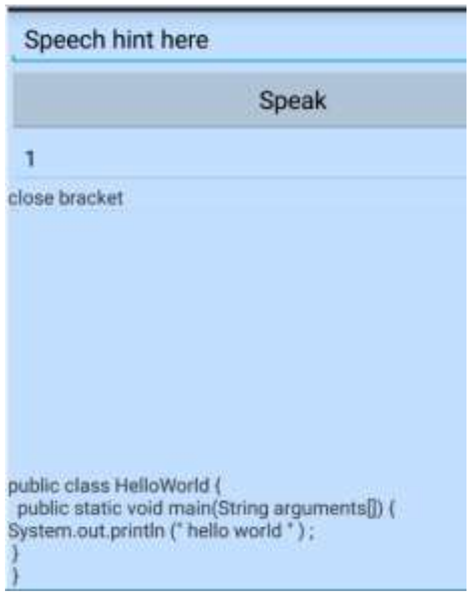
All the above testing's are performed in my project whether to know my application is working accurately or not. The application has also been tested to see whether it meets the user requirements or not. First, I performed unit testing to check whether the flow of my program is smooth going or not. Then I have tested each function separately to know its performance which is module testing. After completion of writing the whole code we test whether the application meets the user requirement i.e. we test whether the application is only accepting the java code. Finally, we take the application to the user and test whether it has all the requirements which needed.

I have tested the application multiple times by considering different cases like giving wrong input and null input. Whenever I gave the wrong input the application did not accept and also shown an error message to the user properly. I have also tried to declare main method twice but the application did accept the main method instead it displayed an appropriate message to the user saying that main method already exists. In this way I had performed all types of testing at different levels on my application to perform better.

CHAPTER SEVEN

OUPUT SCREENSHOTS

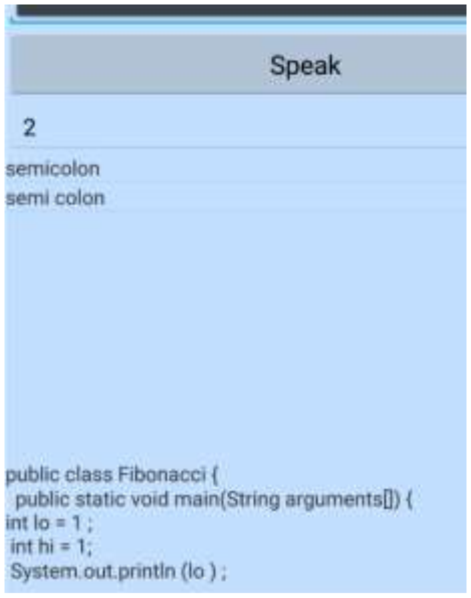
This project has been tested with different level of testing including unit testing, module testing, integration testing, system testing and user acceptance testing and passed successfully with no defect encounter. Below are the screen shots taken during the testing process :



1
close bracket

```
public class HelloWorld {
  public static void main(String arguments[]) {
System.out.println (" hello world " ) ;
}
}
```

Figure 9. Screenshot of Hello World Program



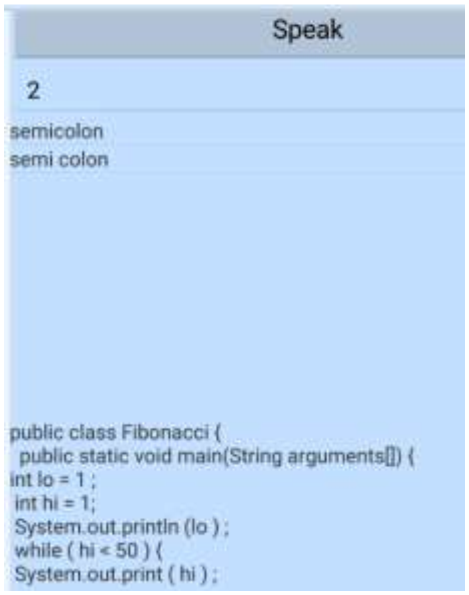
2
semicolon
semi colon

```
public class Fibonacci {
  public static void main(String arguments[]) {
int lo = 1 ;
 int hi = 1;
 System.out.println (lo ) ;
```

Figure 10. Screenshot Explaining the Declaring of Variables

Figure 9: In the above screenshot we could see that for the special character “semicolon if we say semicolon it displays the “;” instead of the word. Here we could see that the simple hello world program is written by using the application.

Figure 10: The above screenshot explains the declaring the integer variables and assigning values to the variables.

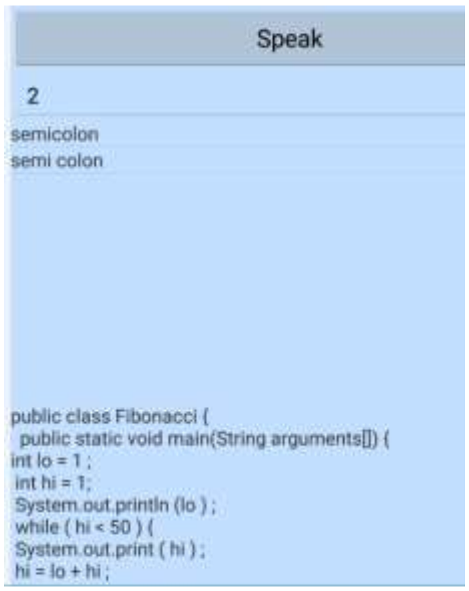


2

semicolon
semi colon

```
public class Fibonacci {
 public static void main(String arguments[]) {
int lo = 1 ;
 int hi = 1;
 System.out.println (lo ) ;
 while ( hi < 50 ) {
 System.out.print ( hi ) ;
```

Figure 11. Screenshot Showing Usage of While Loop

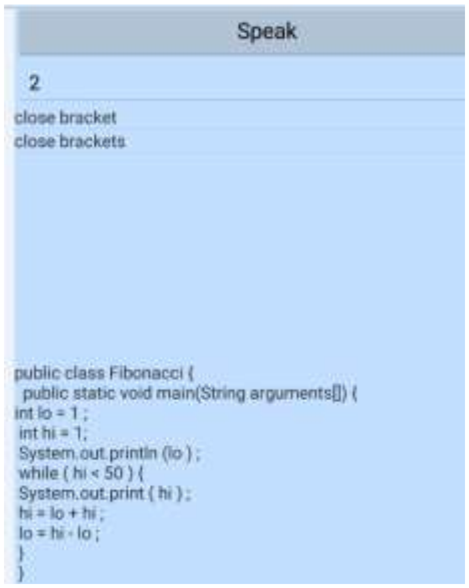


2

semicolon
semi colon

```
public class Fibonacci {
 public static void main(String arguments[]) {
int lo = 1 ;
 int hi = 1;
 System.out.println (lo ) ;
 while ( hi < 50 ) {
 System.out.print ( hi ) ;
 hi = lo + hi ;
```

Figure 12. Screenshot Showing Usage of Operators

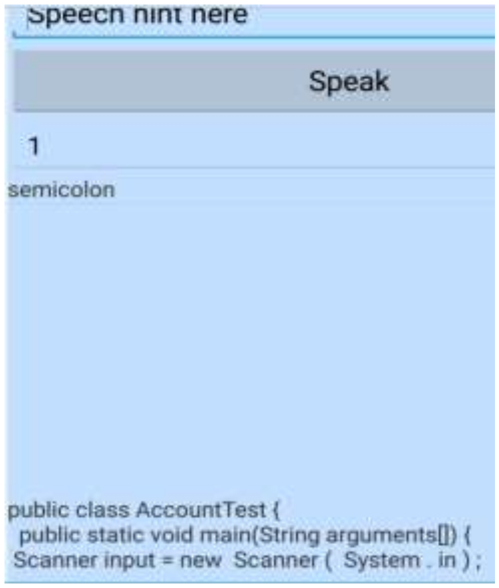


2

close bracket
close brackets

```
public class Fibonacci {
 public static void main(String arguments[]) {
int lo = 1 ;
 int hi = 1;
 System.out.println (lo ) ;
 while ( hi < 50 ) {
 System.out.print ( hi ) ;
 hi = lo + hi ;
 lo = hi - lo ;
 }
 }
```

Figure 13. Screenshot of Fibonacci Program

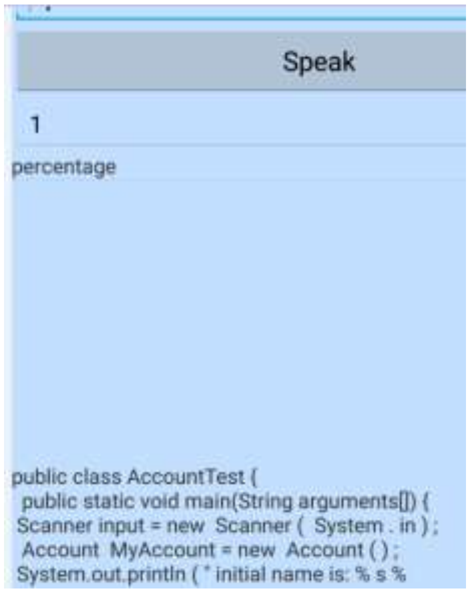


1

semicolon

```
public class AccountTest {
 public static void main(String arguments[]) {
 Scanner input = new  Scanner (  System . in ) ;
```

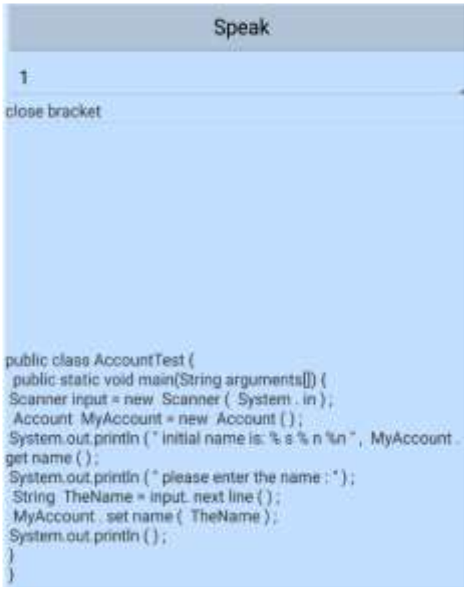
Figure 14. Screenshot Explaining the Usage of Class



percentage

```
public class AccountTest {
 public static void main(String arguments[]) {
Scanner input = new  Scanner (  System . in ) ;
 Account  MyAccount = new  Account ( ) ;
System.out.println ( " initial name is: % s %
```

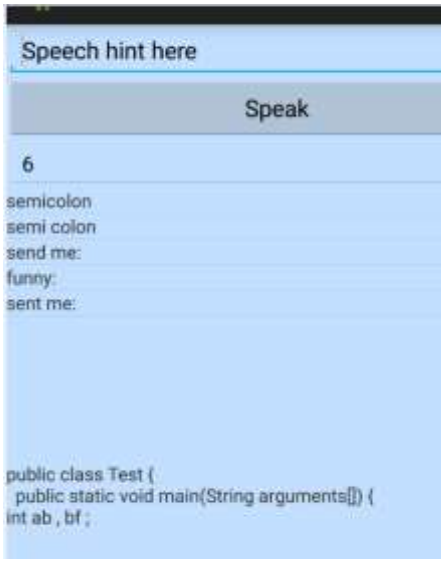
Figure 15.Screenshot Explaining the Usage of Special Characters



close bracket

```
public class AccountTest {
 public static void main(String arguments[]) {
Scanner input = new  Scanner (  System . in ) ;
 Account  MyAccount = new  Account ( ) ;
System.out.println ( " initial name is: % s % n %n " ,  MyAccount .
get name ( ) ;
System.out.println ( " please enter the name : " ) ;
 String  TheName = input. next line ( ) ;
 MyAccount . set name (  TheName ) ;
System.out.println ( ) ;
}
}
```

Figure 16. Screenshot Explaining the Usage of Creating Object and Calling the Function



semicolon
semi colon
send me:
funny:
sent me:

```
public class Test {
 public static void main(String arguments[]) {
int ab , bf ;
```

Figure 17. Screenshot Explaining the Declaration of Typical Integer Variables.

Figure 16: Illustrates the usage of creating the object and also calling the function getname() and illustrates the complete usage of AcoountTest class.

CHAPTER EIGHT

FUTURE ENHANCEMENT

Functionality enhancement in the Speech Recognition project could be the next step of upgrading the application. For example, the adding components such as external libraries or importing other internal projects classes.

Other improvement for this application could be voice detection. This application is using google voice detection and any enhancement in google voice, could be beneficial for this project. The sensitivity of the voice and accent in current version of google can be a challenging aspect for non-English speaker. Therefore, widening the range of word guessing in google voice application can result a positive update for “Speech Recognition” Application.

CHAPTER NINE

CONCLUSION

The Speech Recognition Application is an application for the physically handicapped individuals and helps them to develop their own java programs. The application user can dictate the words and programming command, and by the help of speech recognizer, the application converts the spoken words to text. This system can improve the productivity of disabled intelligent individuals who always wish to write their own program.

Speech Recognition Application can be accessed via devices with Android Operating System. This application has been specifically developed for Android OS and uses Java Programming Language as the development language.

REFERENCES

[1] Developers/Develop/ Reference/ Activity. http://developer.android.com/reference/android/app/Activity.html. (Visited on 05/20/2015).

[2] Android libraries. http://android-libraries.com/. (Visited on 05/20/2015).

[3] Application fundamentals / android developers. http://developer.android.com/guide/components/fundamentals.html. (Visited on 05/20/2015).

[4] Dealing with dependencies in android projects - android tools project site.http://tools.android.com/recent/dealingwithdependenciesinandroid projects. (Visited on 05/20/2015).

[5] The java tutorials. https://docs.oracle.com/javase/tutorial/. (Visited on 05/20/2015).

[6] Recognizer intent / android developers. http://developer.android.com/ reference/android/speech/RecognizerIntent.html,(Visited on 05/20/2015).

[7] Speech recognition - Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Speechrecognition.(Visited on 05/20/2015).

[8] Speechrecognitionengine class (microsoft.speech.recognition). http://msdn.microsoft.com/en-us/library/microsoft.speech.recognition. Speechrecognitionengine.aspx. (Visited on 05/20/2015).

[9] Android (Operating System) - Wikipedia, the free encyclopedia. http://http://en.wikipedia.org/wiki/Android. (Visited on 05/20/2015).

[10] Suma Swamy. *An Efficient Speech Recognition System. Chennai*.2013.

[11] Tong Lai Yu, Santhrushna Gande, and Ronald Yu. *Google Voice Recognition - An Open-Source Based Speech Recognition Android Application for Helping Handicapped Students Writing Programs.* WORLDCOMP'15, July 11-13, Las Vegas, Nevada, USA.

[12] Ronald Yu and Tong Lai Yu. *Ultra High Video Data Compression for Android Devices Using OpenCV and other Open-Source Tools*, Las Vegas, Nevada, USA.ICWN 2014,p.144-150.