**TABLE OF CONTENTS**

**Topic Page #s**

1. SRS 3 - 5
   1. Project Background 3
   2. Mission Statement 3
   3. Glossary 3
   4. Behavior Requirements 4
   5. Error and Warning Requirements 4
   6. Status Requirements 4
   7. Graphical User Interface Requirements 4 - 5
   8. Analysis, Design and Implementation Requirements 5
2. Use Case Diagram 6 - 8
   1. Use Case Description 6 - 8
3. Activity Diagram 9 - 12
4. Basic Class Diagram & CRC 13 - 14
5. MVC 15
6. Sequence Diagram 16 - 18
7. Detailed Class Diagram 19
8. Source Code 20 - 43
   1. Text\_to\_speechActivity.java 20 - 28
   2. SavedAudio.java 29 - 35
   3. Share.java 36 - 38
   4. Imptwindow.java 39 - 40
   5. MyBaseAdapter.java 41 - 42
   6. FileInformation.java 43
9. Layouts Code 44 - 59
   1. Main2.xml 44 - 48
   2. Saved\_audio.xml 49 - 51
   3. Importwindow.xml 52 - 54
   4. Share.xml 55 - 57
   5. Custom\_view\_row.xml 58
   6. Buttonshape.xml 59
   7. Textview.xml 59
   8. Textshape.xml 59
   9. Headphones\_selector.xml 59
   10. Keyboard\_selector.xml 59
10. Snapshots 60 - 64
    1. Main Menu 60
    2. Share Screen 61
    3. Import Screen 62
    4. Share by Email 63
    5. Import from SMS 64

**I PROJECT BACKGROUND**

The National Transportation Safety Board urged all U.S. states to ban drivers from using electronic devices while driving, including text messaging. The NTSB issued the recommendation after several investigations that found texting to be the cause of deadly accidents. According to NHTSA [National Highway Traffic Safety Administration], more than 3,000 people lost their lives last year in distraction-related accidents. It also causes 600,000 crashes every year.

SpeakEasy is an application that helps reading text messages received while driving. It uses the built in Android “text-to-speak” engine. Instead of reading the messages directly, the application shall import the text message and read for the users. Ideally the application should listen for text messages to arrive and automatically attempt to play back the audio “text-to-speech” translation of that text message. However, this will be classified as a “nice-to-have” function and is not part of the original design. The users can also save the audio files so they can listen to the audio later. There shall be an import function that imports text from other applications like email and SMS. Another function that is classified as nice to have is importing text from Facebook and Twitter. However, the difficulty here is that the Facebook and Twitter apps do not allow copying of text. Lastly, there shall be a Share feature that will allow advertising of the application with simple text posts via Facebook, Twitter, and Email. It would also be “nice” if the application can post/send the Saved Audio to Facebook, Twitter, or via Email. This would be a combination of the Saved Audio feature and the Share feature. This is classified as a nice to have feature, and is not required for the final project.

**II MISSION (GOAL) STATEMENT**

The project is required to develop text reading application that can be used for people in moving vehicles, people with sore throats, and as an entertainment tool that can speak for you.

**III GLOSSARY**

1) NTSB: National Transportation Safety Board is a federal investigatory board headquartered in Washington, D.C., whose mandate is to ensure safe public transportation.

2) NHTSA: The National Highway Traffic Safety Administration is an agency of the Executive Branch of the [U.S. government](http://en.wikipedia.org/wiki/United_States_Government), part of the [Department of Transportation](http://en.wikipedia.org/wiki/United_States_Department_of_Transportation).

3) API: An application programming interface is a [source code](http://en.wikipedia.org/wiki/Source_code)-based specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for [routines](http://en.wikipedia.org/wiki/Subroutine), [data structures](http://en.wikipedia.org/wiki/Data_structure), [object classes](http://en.wikipedia.org/wiki/Class_(computer_programming)), and variables.

4) Google: Google Inc. is an American [multinational](http://en.wikipedia.org/wiki/Multinational_corporation) Internet and software corporation specialized in [Internet search](http://en.wikipedia.org/wiki/Internet_search), [cloud computing](http://en.wikipedia.org/wiki/Cloud_computing), and advertising technologies.

5) Facebook: Facebook is a [social networking service](http://en.wikipedia.org/wiki/Social_networking_service) and website launched in February 2004, operated and privately owned by [Facebook Inc.](http://en.wikipedia.org/wiki/Facebook_Inc.).

6) Twitter: Twitter is an online [social networking service](http://en.wikipedia.org/wiki/Social_networking_service) and [micro blogging](http://en.wikipedia.org/wiki/Microblogging) service that enables its users to send and read text-based posts of up to 140 [characters](http://en.wikipedia.org/wiki/Character_(computing)), known as "tweets".

7) SMS: Short Message Service (SMS) is a [text messaging](http://en.wikipedia.org/wiki/Text_messaging) service component of phone, web, or mobile communication systems, using standardized [communications protocols](http://en.wikipedia.org/wiki/Communications_protocols) that allow the exchange of short text messages between [fixed line](http://en.wikipedia.org/wiki/Fixed_line) or [mobile phone](http://en.wikipedia.org/wiki/Mobile_phone) device.

8) TTS: Text To Speech is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software and hardware. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

**V.1 Behavior Requirement**

There will be four options at the bottom of the application: Main, Saved, Share and Import. Each will behave differently according to their specifications. When an application is opened, user is shown the Main screen to type the text in the dialog box. User will also be provided with Speak button at the top-right corner and when implemented, text will be translated into Speech. Moreover, on the bottom-right corner of the text input box, there shall be a Save button, offered to users to save the audio in the archive. There shall be a keyboard toggle button that will allow users to toggle the keyboard ON/OFF for convenience. There shall be a Clear Text button that will clear the screen of any text the user typed so that the user does not have to hit the backspace button – for convenience. There shall be a saved button, which leads to the Saved screen. “Saved” screen is very simple and elegantly implemented in a way, when user saves the audio, it pushes the data to the Saved screen and users have the option to play or delete the file. Share screen is offered for users to advertise the application on Twitter, Facebook or E-mail. An Import screen is available on every part of the application. For example, when user has saved the audio and decides to Import data from SMS or E-mail, the Import button does the trick. It takes users to SMS or E-mail, allowing them to copy the text. As a “nice-to-have” feature, the application should automatically return to the Speak/Main screen allowing the user to paste in the text. Same procedure can be implemented with the E-mails. Lastly, share screen will implement three buttons (Facebook, Twitter, and Email), which shall post/send text via those media channels.

**V.2 Error and Warning Requirement**

Speak-Easy should give pop-up warnings to the user if he invokes an action not appropriate for the system in its current state. An example of this would include the user hitting the Save button when the user did not input any text. The application should warn the user that he should input the text before attempting to save.

### V.3 Status Requirements

The possible status of Speak-Easy is idle. Speak-Easy will work by allowing the user to import text into the application, and Speak-Easy will read it out-loud. In the 2nd phase of this project, Speak-Easy “might” be able to read texts automatically as they come in. This is not a requirement, and is classified as a "nice-to-have" feature. The user should have to turn on a setting to read texts automatically as they come in. This should allow the application to continue working in the background. This is not guaranteed because unlike the iPhone, Androids have a limitation in having applications run in the background. Most applications, except for native apps, are shut down when you hit the back or exit button. If possible, we will try to work around this limitation, but again this is not guaranteed, and is not considered as part of SRS. This is an extra feature that should be classified as OUT or Maybe.

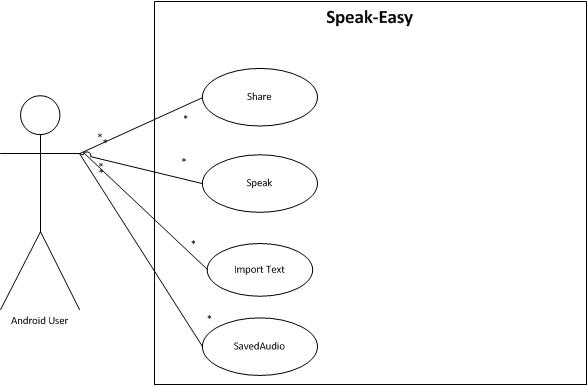
### V.4 Graphical User Interface Requirements

1. A Speak button shall be displayed. If the user presses the Speak button, it will check if there is any entered text – if so, then it will speak – if not, it will display an error message.
2. A button shall be displayed allowing the user to save the spoken text. It will save as an audio file.
3. A textbox shall be displayed allowing users to type in the text they would like to be spoken.
4. A Saved button shall be displayed allowing the user to navigate to the Saved screen.
5. A keyboard button shall be displayed allowing the user to toggle the keyboard.
6. A clear text button shall be displayed allowing the user to clear any text in the textbox.
7. A Progress Dialog shall be displayed upon touching the Speak button as it takes a few seconds for the Text-To-Speech engine to initialize.
8. There shall be a navigation menu including Main, Saved, Shared, and Import buttons/labels.
9. The Share screen shall display the three possible sharing options including Email, Twitter, and Facebook.
10. The Saved Audio screen shall display a List View allowing the user to play previously saved audio.
11. The Saved Audio screen shall have buttons allowing the user to delete selected audio as well as all audio.
12. The import screen shall display two choices for the user: SMS or E-mail.

### V.5 ANALYSIS, DESIGN AND IMPLEMENTATION REQUIREMENTS

Object-Oriented Analysis (OOA), Object-Oriented Design (OOD) and Object-Oriented Programming (OOP) are required. Use case diagrams, class diagrams, and activity diagrams are required. An OOP language such as Java is required. CASE tools should be used for OOA, OOD and OOP: Microsoft Visio and UMLDiagrammer (or other UML drawing tools) can be used for OOA and OOD .

**2. Use Case**



**2.1 Use Case description**

Import Text

User imports the text from an Email or Text message as input for the text-to-speech feature.

|  |  |  |
| --- | --- | --- |
| **Scenario** | Import text from Email or Text Messenger | |
| **Triggering event** | User hits the Import button, and then chooses Email or Text | |
| **Actors** | Android Users | |
| **Related use cases** | NA | |
| **Stakeholders** | Android Users, Application Developers | |
| **Pre-condition** | Application needs to be running, and user needs to press the import button. | |
| **Post-condition** | NA | |
| **Flow of events** | **Actor** | **System** |
| 1. User presses Import button | Import Text |
| **Exception** | * NA | |

Listen to Saved Audio

Allows the user to listen to audio that the user saved past time.

|  |  |  |
| --- | --- | --- |
| **Scenario** | Listen saved audio files | |
| **Triggering event** | Hits the Saved button, and chooses audio files. | |
| **Actors** | Android Users | |
| **Related use cases** | NA | |
| **Stakeholders** | Android Users, Application Developers | |
| **Pre-condition** | Application needs to be running, and user needs to first hit the Saved button, and then user needs to choose audio file to play. | |
| **Post-condition** | NA | |
| **Flow of events** | **Actor** | **System** |
| 1. User hits Saved button 2. User chooses audio files | Saved  Play audio files |
| **Exception** | * NA | |

Share

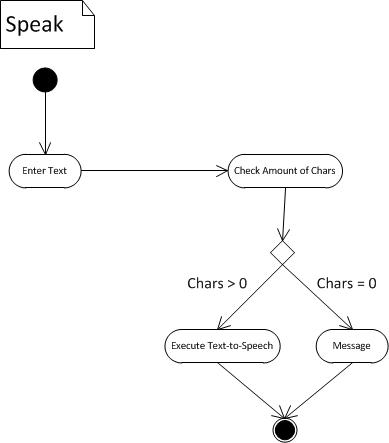
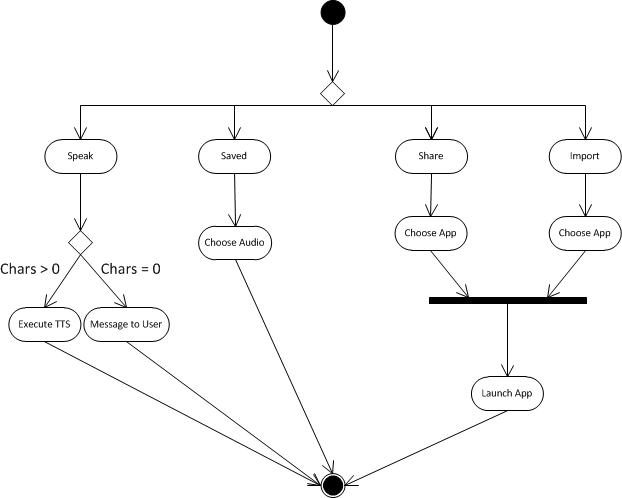
Allows the user to share this application via email and social media.

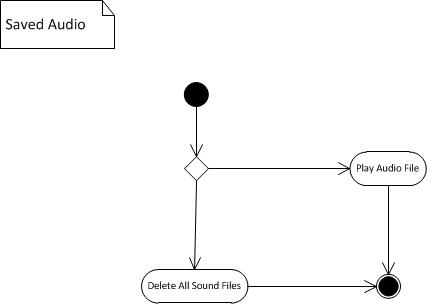
|  |  |  |
| --- | --- | --- |
| **Scenario** | Email or post on social media a brief advertisement of the application. | |
| **Triggering event** | User hits the Share button. | |
| **Actors** | Android Users | |
| **Related use cases** | NA | |
| **Stakeholders** | Android Users, Application Developers | |
| **Pre-condition** | Application needs to be running. | |
| **Post-condition** | NA | |
| **Flow of events** | **Actor** | **System** |
| 1. Hit the Share button | Share |
| **Exception** | * NA | |

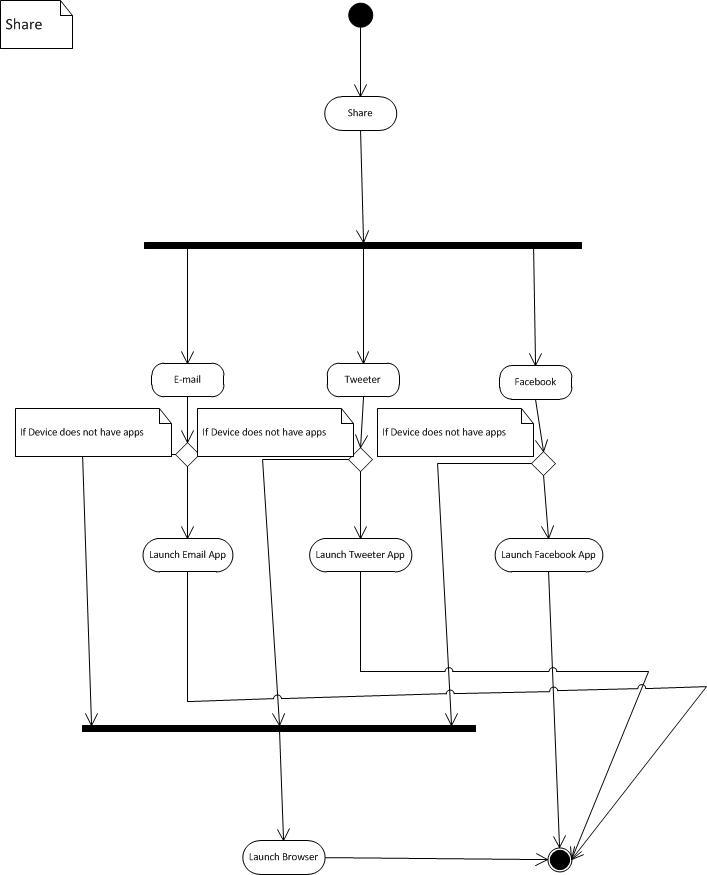
Speak

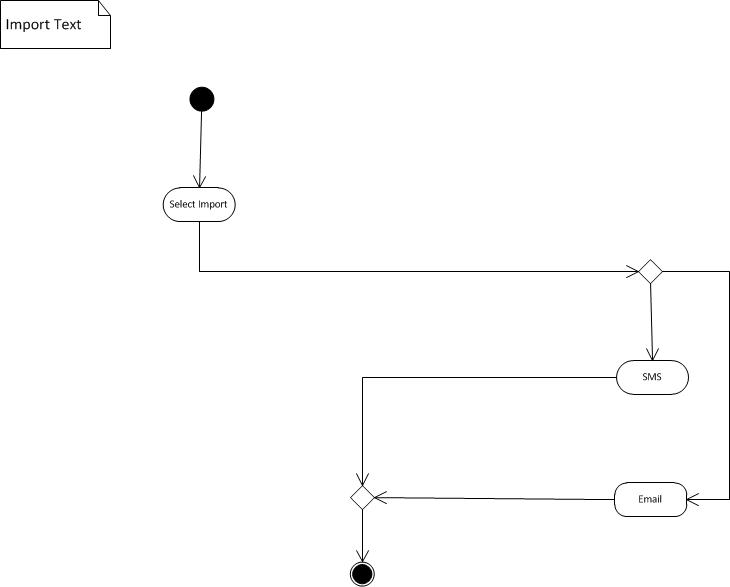
Speaks out loud the text entered by the user.

|  |  |  |
| --- | --- | --- |
| **Scenario** | The phone will read the text input and speak it out loud. | |
| **Triggering event** | The user hits the Speak button. | |
| **Actors** | Android Users | |
| **Related use cases** | NA | |
| **Stakeholders** | Android Users, Application Developers | |
| **Pre-condition** | Application is running, and user either typed the text manually or imported it. | |
| **Post-condition** | System will have a sound file saved for future playback. | |
| **Flow of events** | **Actor** | **System** |
| 1. User hits the Speak button | Speak |
| **Exception** | * There is no input. User gets a warning. | |

**3. Activity Diagram**

****

****

****

**4. CRC cards**

|  |  |
| --- | --- |
| **Main** | **Collaborators**   * Saved * Import * FileInformation |
| **Responsibilities**   * onCreate() * onClick() * onTouch() * onInit() * speakWords() * onActivityResult() * saveAudioToFile() * toggleKeyboard() * speakUtterance() * GetFileInfo() * saveArrayToDisk() * findDups() * onDestroy() * onPause() |

|  |  |
| --- | --- |
| **Saved** | **Collaborators**   * Main * MyBaseAdapter * FileInformation |
| **Responsibilities**   * onCreate() * onClick() * playFile() * displayList() * deleteAudioFile() * removeIndividualFile() * saveArrayToDisk() * deleteAllFiles() * onDestroy |

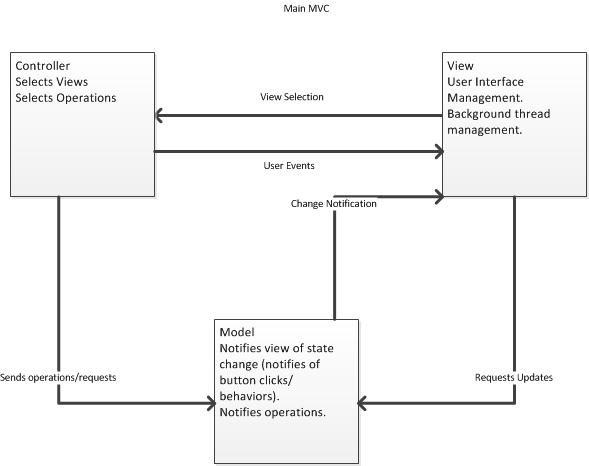
|  |  |
| --- | --- |
| **MyBaseAdapter** | **Collaborators**   * SaveAudio * FileInformation |
| **Responsibilities**   * MyBaseAdapter() * getCount() * getItem() * getItemId() * getView() |

|  |  |
| --- | --- |
| **FileInformation** | **Collaborators**   * Main * SavedAudio * MyBaseAdapter |
| **Responsibilities**   * FileInformation() * setPrevName() * getPrevName() * setDestinationFileName() * getDestinationFileName() |

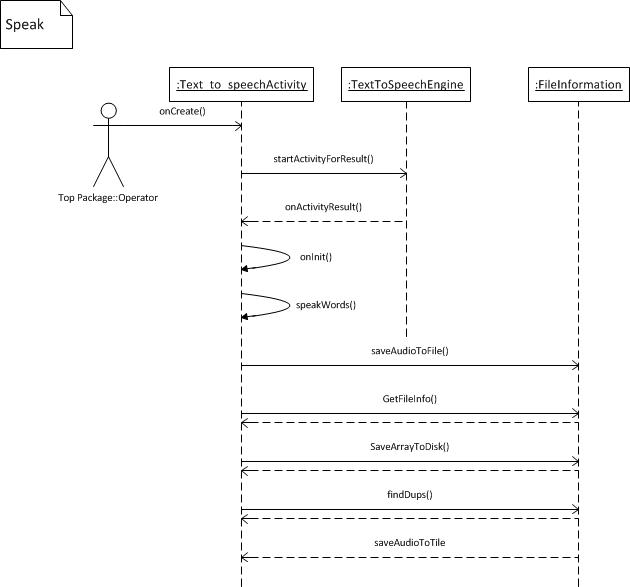
|  |  |
| --- | --- |
| **Share** | **Collaborators** |
| **Responsibilities**   * onCreate() * onClick() * appInstalledOrNot() |

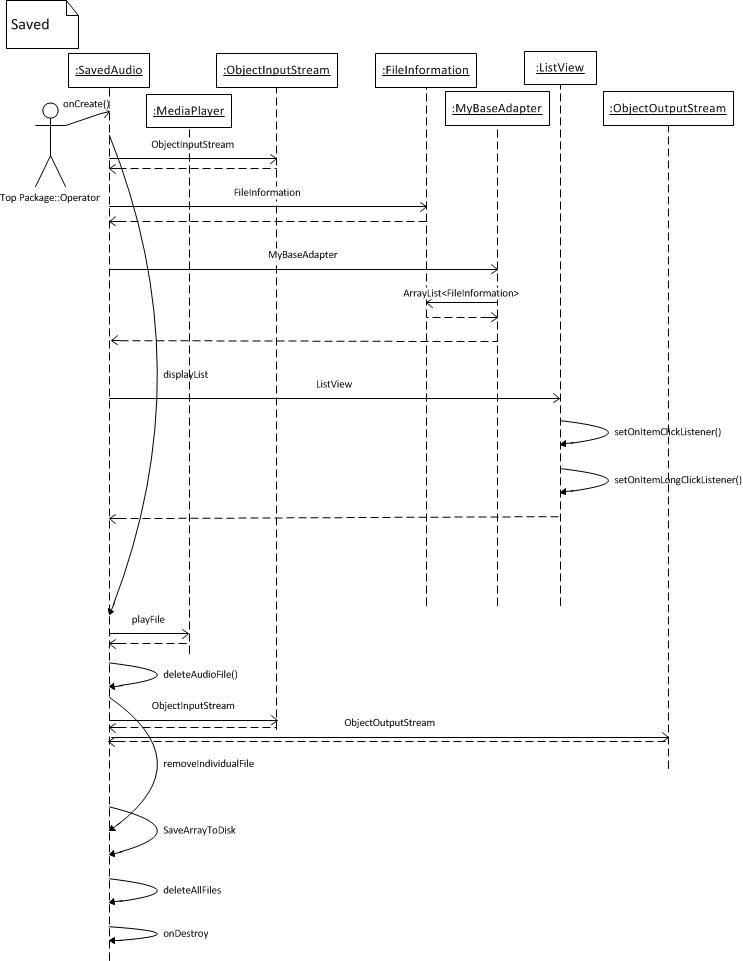
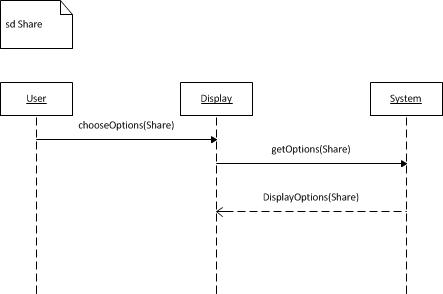
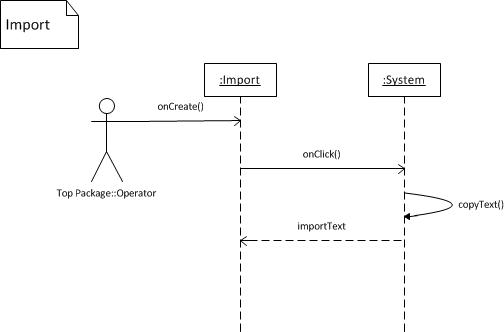
|  |  |
| --- | --- |
| **Import** | **Collaborators**   * Main |
| **Responsibilities**   * onCreate() * onClick() |

**5. MVC**

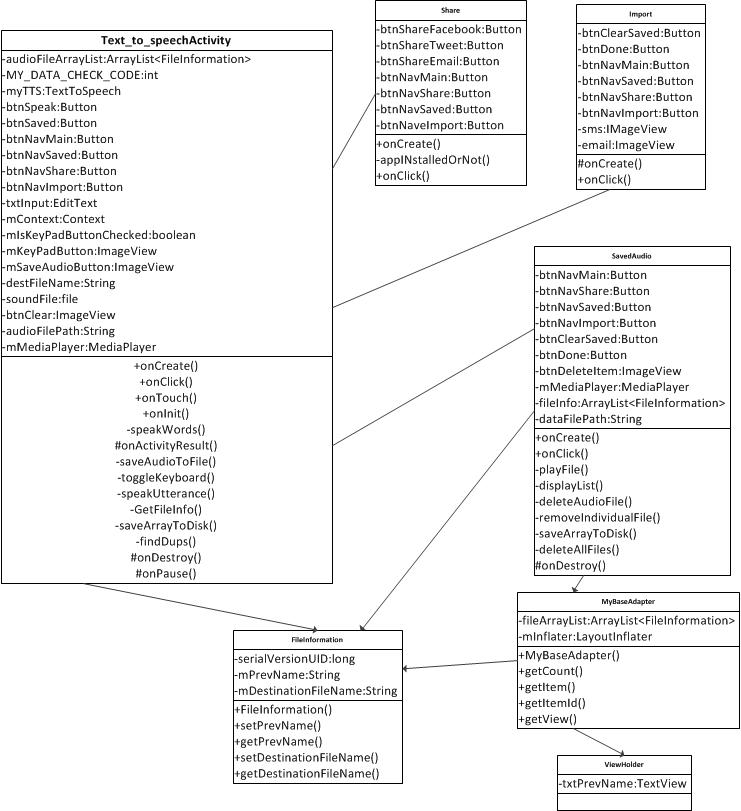
****

**6. Sequence Diagram**

****

****

**7. Detailed Class Diagram**

****

**8. Source Code**

**Text\_to\_speechActivity.java**

package com.comp380.texttospeech;

import android.app.Activity;

import android.app.AlertDialog;

import android.app.ProgressDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.media.AudioManager;

import android.media.MediaPlayer;

import android.media.MediaPlayer.OnCompletionListener;

import android.os.AsyncTask;

import android.os.Bundle;

import android.os.Environment;

import android.speech.tts.TextToSpeech;

import android.view.MotionEvent;

import android.view.View.OnClickListener;

import android.view.View.OnTouchListener;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.Toast;

import android.view.View;

import android.view.inputmethod.InputMethodManager;

import android.widget.EditText;

import android.speech.tts.TextToSpeech.OnInitListener;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.OptionalDataException;

import java.io.StreamCorruptedException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Locale;

import java.util.Random;

public class Text\_to\_speechActivity extends Activity implements OnClickListener, OnTouchListener, OnInitListener {

/\*\* Called when the activity is first created. \*/

public ArrayList<FileInformation> audioFileArrayList = null;

private int MY\_DATA\_CHECK\_CODE = 0;

private TextToSpeech myTTS;

Button btnSpeak, btnSaved, btnNavMain, btnNavSaved, btnNavShare, btnNavImport;

EditText txtInput;

Context mContext;

boolean mIsKeyPadButtonChecked = false;

ImageView mKeyPadButton, mSaveAudioButton, btnClear;

String destFileName, audioFilePath;

MediaPlayer mMediaPlayer = null;

File soundFile = null;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main2);

mContext = this;

btnSpeak = (Button) findViewById(R.id.btnSpeak2);

btnSpeak.setOnClickListener(this);

btnSaved = (Button) findViewById(R.id.btnSaved2);

btnSaved.setOnClickListener(this);

btnNavMain = (Button) findViewById(R.id.btnNavMain2);

btnNavMain.setOnClickListener(this);

btnNavSaved = (Button) findViewById(R.id.btnNavSaved2);

btnNavSaved.setOnClickListener(this);

btnNavShare = (Button) findViewById(R.id.btnNavShare2);

btnNavShare.setOnClickListener(this);

btnNavImport = (Button) findViewById(R.id.btnNavImport2);

btnNavImport.setOnClickListener(this);

mSaveAudioButton = (ImageView) findViewById(R.id.headphones\_down);

mSaveAudioButton.setOnClickListener(this);

mKeyPadButton = (ImageView) findViewById(R.id.keyboard);

mKeyPadButton.setOnClickListener(this);

btnClear = (ImageView) findViewById(R.id.btnClear);

btnClear.setOnClickListener(this);

txtInput = (EditText) findViewById(R.id.txtInput2);

txtInput.setOnTouchListener(this);

Intent checkTTSIntent = new Intent();

checkTTSIntent.setAction(TextToSpeech.Engine.ACTION\_CHECK\_TTS\_DATA);

startActivityForResult(checkTTSIntent, MY\_DATA\_CHECK\_CODE);

}

// touching the text input sets the boolean mIsKeyPadButtonChecked to True

// symbolizing that the Keyboard is showing

public boolean onTouch(View v, MotionEvent event) {

mIsKeyPadButtonChecked = true;

return false;

}

// Click listener for this class

public void onClick(View v)

{

switch(v.getId())

{

case R.id.btnSpeak2 : speakUtterance();

break;

case R.id.btnSaved2 : startActivity(new Intent(this, SavedAudio.class));

break;

case R.id.btnNavSaved2 : startActivity(new Intent(this, SavedAudio.class));

break;

case R.id.btnNavShare2 : startActivity(new Intent(this, share.class));

break;

case R.id.btnNavImport2 : startActivity(new Intent(this, importwindow.class));

break;

case R.id.btnClear : txtInput.setText("");

break;

case R.id.keyboard : toggleKeyboard();

break;

case R.id.headphones\_down : saveAudioToFile();

break;

}

}

// Called to signal the completion of the TextToSpeech engine initialization.

public void onInit(int initStatus)

{

if(initStatus == TextToSpeech.SUCCESS)

myTTS.setLanguage(Locale.US);

else if(initStatus == TextToSpeech.ERROR)

Toast.makeText(this, "Sorry! Text To Speech failed...", Toast.LENGTH\_LONG).show();

}

// Speak words

private void speakWords(String speech)

{

myTTS.speak(speech, TextToSpeech.QUEUE\_FLUSH, null);

}

// Called when an activity onActivityForResult(checkTTSIntent) exits

protected void onActivityResult(int requestCode, int resultCode, Intent data)

{

if(requestCode == MY\_DATA\_CHECK\_CODE)

{

if(resultCode == TextToSpeech.Engine.CHECK\_VOICE\_DATA\_PASS)

{

myTTS = new TextToSpeech(this, this);

}

else

{

Intent installTTSIntent = new Intent();

installTTSIntent.setAction(TextToSpeech.Engine.ACTION\_INSTALL\_TTS\_DATA);

startActivity(installTTSIntent);

}

}

}

private void saveAudioToFile()

{

String dataFilePath = Environment.getExternalStorageDirectory().getAbsolutePath() +

File.separatorChar + "FileInformationFile.dat";

ArrayList<FileInformation> fileInfo = null;

final String words = txtInput.getText().toString();

final int maxFileNameLength = 25;

String saveText, previewName;

int spacePosition;

if(words.length()>0)

{

saveText = words;

if(saveText.length() > maxFileNameLength)

previewName = saveText.substring(0, maxFileNameLength) + "...";

else

previewName = saveText;

spacePosition = saveText.indexOf(" ");

if(spacePosition>0) {

if(Environment.MEDIA\_MOUNTED.equals(Environment.getExternalStorageState()))

destFileName = Environment.getExternalStorageDirectory().getAbsolutePath() +

File.separatorChar + saveText.substring(0, spacePosition) + ".wav";

}

else {

if(Environment.MEDIA\_MOUNTED.equals(Environment.getExternalStorageState()))

destFileName = Environment.getExternalStorageDirectory().getAbsolutePath() +

File.separatorChar + saveText + ".wav";

}

Random rand = new Random();

if(findDups(destFileName, dataFilePath))

{

if(spacePosition>0) {

if(Environment.MEDIA\_MOUNTED.equals(Environment.getExternalStorageState()))

destFileName = Environment.getExternalStorageDirectory().getAbsolutePath() +

File.separatorChar + saveText.substring(0, spacePosition) + rand.nextInt(20) + ".wav";

}

else {

if(Environment.MEDIA\_MOUNTED.equals(Environment.getExternalStorageState()))

destFileName = Environment.getExternalStorageDirectory().getAbsolutePath() +

File.separatorChar + saveText + rand.nextInt(20) + ".wav";

}

}

HashMap<String, String> myHashRender = new HashMap();

myHashRender.put(TextToSpeech.Engine.KEY\_PARAM\_UTTERANCE\_ID, previewName);

try{

if(myTTS.synthesizeToFile(saveText, myHashRender, destFileName) == TextToSpeech.SUCCESS)

{

//make a suitable path to a file on the external storage

String state = Environment.getExternalStorageState();

if(Environment.MEDIA\_MOUNTED.equals(state))

{

//create some data and fill up array

audioFileArrayList = GetFileInfo(previewName, destFileName);

ObjectInputStream in = null;

try {

in = new ObjectInputStream(new FileInputStream(dataFilePath));

} catch (StreamCorruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// if the ObjectInputStream is empty, create a new OutputStream and write ArrayList to disk

if(in == null)

{

saveArrayToDisk(audioFileArrayList, dataFilePath);

Toast.makeText(this.getBaseContext(), "New data file created. Saved successfully!", Toast.LENGTH\_SHORT).show();

}

else

{

try {

fileInfo = (ArrayList<FileInformation>) in.readObject();

} catch (OptionalDataException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

fileInfo.addAll(audioFileArrayList);

saveArrayToDisk(fileInfo, dataFilePath);

Toast.makeText(getBaseContext(), "Text-To-Speech Saved!", Toast.LENGTH\_SHORT).show();

}

}

}

else

Toast.makeText(getBaseContext(), "Oops! Something happened. Your text was not saved.", Toast.LENGTH\_SHORT).show();

} catch(Exception e) {

Toast.makeText(getBaseContext(), "synthesizeToFile Failed", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

}

else

Toast.makeText(getBaseContext(), "Sorry! You have to type text in order to save.", Toast.LENGTH\_SHORT).show();

}

private void toggleKeyboard()

{

mIsKeyPadButtonChecked = !mIsKeyPadButtonChecked;

//mKeyPadButton.setActivated(mIsKeyPadButtonChecked); //Android 4.0

if(mIsKeyPadButtonChecked)

{

mKeyPadButton.postDelayed(new Runnable()

{

public void run()

{

InputMethodManager keyboard = (InputMethodManager) getSystemService(INPUT\_METHOD\_SERVICE);

keyboard.showSoftInput(txtInput , 0);

}

}, 200);

}

else

{

mKeyPadButton.postDelayed(new Runnable()

{

public void run()

{

InputMethodManager keyboard = (InputMethodManager) getSystemService(INPUT\_METHOD\_SERVICE);

keyboard.hideSoftInputFromWindow(txtInput.getWindowToken() , 0);

}

}, 200);

}

}

// If Speak button is clicked, executes speakWords method and shows ProgressBar

private void speakUtterance()

{

final String words = txtInput.getText().toString();

try {

if(words.length()>0)

{

AsyncTask<Void,Void,Void> lTask = new AsyncTask<Void,Void,Void>(){

ProgressDialog lDialog;

protected void onPreExecute(){

lDialog = ProgressDialog.show(mContext, "","Your msg",true);

}

protected Void doInBackground(Void... params){

speakWords(words);

return null;

}

protected void onPostExecute(Void result){

Thread myThread = new Thread(){

public void run()

{

try {

sleep(1000);

} catch(InterruptedException e) {

e.printStackTrace();

} finally {

lDialog.dismiss();

}

}

};

myThread.start();

}

};

lTask.execute();

}

else

Toast.makeText(getApplicationContext(), "You have to type words first!", Toast.LENGTH\_SHORT).show();

} catch(Exception e) {

Toast.makeText(getParent(), "Something else went wrong!", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

}

private ArrayList<FileInformation> GetFileInfo(String strPrevName, String strDestName) {

ArrayList<FileInformation> file\_info = new ArrayList<FileInformation>();

FileInformation fi = new FileInformation(strPrevName, strDestName);

file\_info.add(fi);

return file\_info;

}

private void saveArrayToDisk(ArrayList<FileInformation> myArray, String myPath)

{

try {

// write array to disk

File file = new File(myPath);

if(file.exists())

file.delete();

ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(myPath));

out.writeObject(myArray);

out.flush();

out.close();

}

catch(FileNotFoundException e)

{

Toast.makeText(getBaseContext(), "File Not Found.", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

catch(IOException e)

{

Toast.makeText(getBaseContext(), "IO Exception.", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

catch(Exception e)

{

Toast.makeText(getBaseContext(), "Exception!", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

}

private boolean findDups(String myAudioFilePath, String myDataFilePath)

{

ArrayList<FileInformation> myList = null;

ObjectInputStream in = null;

try {

in = new ObjectInputStream(new FileInputStream(myDataFilePath));

} catch (StreamCorruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

if(in != null)

{

try {

myList = (ArrayList<FileInformation>) in.readObject();

} catch (OptionalDataException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

for(int i=0;i<myList.size();i++)

{

if(myAudioFilePath.equals(myList.get(i).getDestinationFileName()))

return true;

}

}

return false;

}

@Override

protected void onDestroy() {

// TODO Auto-generated method stub

super.onDestroy();

}

@Override

protected void onPause() {

// TODO Auto-generated method stub

super.onPause();

}

}

**SavedAudio.java**

package com.comp380.texttospeech;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.OptionalDataException;

import java.io.StreamCorruptedException;

import java.util.ArrayList;

import com.comp380.texttospeech.R;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.media.AudioManager;

import android.media.MediaPlayer;

import android.media.MediaPlayer.OnCompletionListener;

import android.os.Bundle;

import android.os.Environment;

import android.os.Parcelable;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.AdapterView.OnItemLongClickListener;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.ListView;

import android.widget.Toast;

public class SavedAudio extends Activity implements OnClickListener {

Button btnClearSaved, btnDone, btnNavMain, btnNavSaved, btnNavShare, btnNavImport;

ImageView btnDeleteItem;

MediaPlayer mMediaPlayer;

ArrayList<FileInformation> fileInfo = null;

String dataFilePath = Environment.getExternalStorageDirectory().getAbsolutePath() +

File.separatorChar + "FileInformationFile.dat";

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.saved\_audio);

btnClearSaved = (Button) findViewById(R.id.btnClearSaved);

btnClearSaved.setOnClickListener(this);

btnDone = (Button) findViewById(R.id.btnDoneSaved3);

btnDone.setOnClickListener(this);

btnNavMain = (Button) findViewById(R.id.btnNavMain3);

btnNavMain.setOnClickListener(this);

btnNavSaved = (Button) findViewById(R.id.btnNavSaved3);

btnNavSaved.setOnClickListener(this);

btnNavShare = (Button) findViewById(R.id.btnNavShare3);

btnNavShare.setOnClickListener(this);

btnNavImport = (Button) findViewById(R.id.btnNavImport3);

btnNavImport.setOnClickListener(this);

displayList();

}

public void onClick(View arg0) {

switch (arg0.getId()) {

case R.id.btnDoneSaved3:

finish();

break;

case R.id.btnNavMain3:

startActivity(new Intent(this, Text\_to\_speechActivity.class));

break;

case R.id.btnNavShare3:

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

break;

case R.id.btnNavImport3:

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

break;

case R.id.btnClearSaved : deleteAllFiles();

break;

}

}

private void playFile(String strFileName)

{

try {

mMediaPlayer = new MediaPlayer();

mMediaPlayer.setDataSource(strFileName);

mMediaPlayer.setAudioStreamType(AudioManager.STREAM\_RING);

mMediaPlayer.prepare();

mMediaPlayer.start();

mMediaPlayer.setOnCompletionListener(new OnCompletionListener() {

public void onCompletion(MediaPlayer mp) {

mMediaPlayer.stop();

}

});

} catch (Exception e) {

Toast.makeText(getBaseContext(), "Sorry! Can't play file", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

}

private void displayList()

{

ObjectInputStream in = null;

try {

in = new ObjectInputStream(new FileInputStream(dataFilePath));

} catch (StreamCorruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

if(in != null)

{

try {

fileInfo = (ArrayList<FileInformation>) in.readObject();

} catch (OptionalDataException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

final ListView lv = (ListView) findViewById(R.id.srListView);

lv.setAdapter(new MyBaseAdapter(this, fileInfo));

lv.setOnItemClickListener(new OnItemClickListener()

{

@Override

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

Object o = lv.getItemAtPosition(position);

FileInformation fullObject = (FileInformation) o;

playFile(fullObject.getDestinationFileName());

//Toast.makeText(SavedAudio.this, "You have chosen: " + " " + fullObject.getPrevName(), Toast.LENGTH\_LONG).show();

}

});

lv.setOnItemLongClickListener(new OnItemLongClickListener()

{

@Override

public boolean onItemLongClick(AdapterView<?> arg0, View arg1, final int arg2, long arg3) {

Object o = lv.getItemAtPosition(arg2);

final FileInformation fullObject = (FileInformation) o;

//deleteFile(fullObject.getDestinationFileName());

//deleteAudioFile(fullObject.getDestinationFileName());

final DialogInterface.OnClickListener dialogClickListener = new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

switch (which){

case DialogInterface.BUTTON\_POSITIVE:

deleteAudioFile(fullObject.getDestinationFileName());

removeIndividualFile(arg2);

//Toast.makeText(getBaseContext(), "You clicked YES!", Toast.LENGTH\_SHORT).show();

break;

case DialogInterface.BUTTON\_NEGATIVE:

//Toast.makeText(getBaseContext(), "You clicked NO!", Toast.LENGTH\_SHORT).show();

break;

}

}

};

AlertDialog.Builder builder = new AlertDialog.Builder(SavedAudio.this);

builder.setMessage("Are you sure you want to delete " + "\"" + fullObject.getPrevName() + "\"?")

.setPositiveButton("Yes", dialogClickListener)

.setNegativeButton("No", dialogClickListener).show();

return false;

}

});

}

}

private void deleteAudioFile(String strFilePath)

{

File mfile = new File(strFilePath);

mfile.delete();

}

private void removeIndividualFile(int position)

{

ObjectInputStream in = null;

try {

in = new ObjectInputStream(new FileInputStream(dataFilePath));

} catch (StreamCorruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

if(in == null)

{

fileInfo.remove(position);

Toast.makeText(getBaseContext(), "File deleted!", Toast.LENGTH\_SHORT).show();

saveArrayToDisk();

}

else

{

try {

fileInfo = (ArrayList<FileInformation>) in.readObject();

} catch (OptionalDataException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

fileInfo.remove(position);

Toast.makeText(getBaseContext(), "File deleted!", Toast.LENGTH\_SHORT).show();

saveArrayToDisk();

}

displayList();

}

private void saveArrayToDisk()

{

try {

// write array to disk

File file = new File(dataFilePath);

if(file.exists())

file.delete();

ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(dataFilePath));

out.writeObject(fileInfo);

out.flush();

out.close();

}

catch(FileNotFoundException e)

{

Toast.makeText(getBaseContext(), "File Not Found.", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

catch(IOException e)

{

Toast.makeText(getBaseContext(), "IO Exception.", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

catch(Exception e)

{

Toast.makeText(getBaseContext(), "Exception!", Toast.LENGTH\_SHORT).show();

e.printStackTrace();

}

}

private void deleteAllFiles()

{

DialogInterface.OnClickListener dialogClickListener = new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

switch (which){

case DialogInterface.BUTTON\_POSITIVE:

//Yes button clicked

int stringLength, arrayLength;

String fileName, fileNameSubstring;

File mfile = new File(Environment.getExternalStorageDirectory().getAbsolutePath() + File.separatorChar);

File[] list=mfile.listFiles();

arrayLength = list.length;

// Delete everything

for(int j=0;j<arrayLength;j++)

{

fileName = list[j].getName();

stringLength = fileName.length();

fileNameSubstring = fileName.substring(stringLength-3,stringLength);

if(fileNameSubstring.equalsIgnoreCase("wav"))

list[j].delete();

}

String dataFilePath = Environment.getExternalStorageDirectory().getAbsolutePath() +

File.separatorChar + "FileInformationFile.dat";

File myFile = new File(dataFilePath);

myFile.delete();

fileInfo.clear();

saveArrayToDisk();

displayList();

Toast.makeText(getBaseContext(), "Files deleted!", Toast.LENGTH\_SHORT).show();

break;

case DialogInterface.BUTTON\_NEGATIVE:

//No button clicked

Toast.makeText(getBaseContext(), "No files deleted!", Toast.LENGTH\_SHORT).show();

break;

}

}

};

File mfile1 = new File(Environment.getExternalStorageDirectory().getAbsolutePath() + File.separatorChar);

File[] list1=mfile1.listFiles();

if(list1.length>1)

{

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setMessage("Are you sure you want to delete all your saved files?")

.setPositiveButton("Yes", dialogClickListener)

.setNegativeButton("No", dialogClickListener).show();

}

else

Toast.makeText(getBaseContext(), "The queue is empty. No files to delete!", Toast.LENGTH\_SHORT).show();

}

@Override

protected void onDestroy() {

// TODO Auto-generated method stub

super.onDestroy();

}

}

**Share.java**

package com.comp380.texttospeech;

import java.util.List;

import android.app.Activity;

import android.content.ActivityNotFoundException;

import android.content.ComponentName;

import android.content.Context;

import android.content.Intent;

import android.content.pm.ActivityInfo;

import android.content.pm.ApplicationInfo;

import android.content.pm.PackageManager;

import android.content.pm.ResolveInfo;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.Toast;

public class share extends Activity implements OnClickListener{

Button btnShareEmail, btnShareTweet, btnShareFacebook, btnNavMain, btnNavSaved, btnNavShare, btnNavImport;

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.share);

btnShareEmail = (Button) findViewById(R.id.btnShareEmail);

btnShareEmail.setOnClickListener(this);

btnShareTweet = (Button) findViewById(R.id.btnShareTweet);

btnShareTweet.setOnClickListener(this);

btnShareFacebook = (Button) findViewById(R.id.btnShareFacebook);

btnShareFacebook.setOnClickListener(this);

btnNavMain = (Button) findViewById(R.id.btnNavMain);

btnNavMain.setOnClickListener(this);

btnNavSaved = (Button) findViewById(R.id.btnNavSaved);

btnNavSaved.setOnClickListener(this);

btnNavShare = (Button) findViewById(R.id.btnNavShare);

btnNavShare.setOnClickListener(this);

btnNavImport = (Button) findViewById(R.id.btnNavImport);

btnNavImport.setOnClickListener(this);

}

private boolean appInstalledOrNot(String uri)

{

PackageManager pm = getPackageManager();

boolean app\_installed = false;

try {

pm.getPackageInfo(uri, PackageManager.GET\_ACTIVITIES);

app\_installed = true;

}

catch (PackageManager.NameNotFoundException e) {

app\_installed = false;

}

return app\_installed;

}

public void onClick(View v) {

if(v.getId() == R.id.btnNavMain)

startActivity(new Intent(this, Text\_to\_speechActivity.class));

else if(v.getId() == R.id.btnNavSaved)

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

else if(v.getId() == R.id.btnNavImport)

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

Intent intent = null;

PackageManager manager = null;

boolean installed = false;

String adMessage = "This is Advertisement! Tell your friends about Speak Easy! =)";

switch (v.getId()) {

//if app is installed -> launch app

//else -> browser (facebook URL)

case R.id.btnShareFacebook:

installed = appInstalledOrNot("com.facebook.katana");

if(installed){

intent = new Intent(Intent.ACTION\_SEND);

manager = getPackageManager();

intent = manager.getLaunchIntentForPackage("com.facebook.katana");

intent.addCategory(Intent.CATEGORY\_LAUNCHER);

startActivity(intent);

break;

}

else {

intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("http://www.facebook.com"));

startActivity(intent);

break;

}

//if app is installed -> launch app

//else -> browser (twitter URL)

case R.id.btnShareTweet:

installed = appInstalledOrNot("com.twitter.android");

if(installed){

intent = new Intent(android.content.Intent.ACTION\_SEND);

intent.setType("text/plain");

intent.putExtra(android.content.Intent.EXTRA\_TEXT, adMessage);

manager = v.getContext().getPackageManager();

List<ResolveInfo> activityListT = manager.queryIntentActivities(intent, 0);

for (final ResolveInfo app : activityListT) {

if ("com.twitter.android.PostActivity".equals(app.activityInfo.name)) {

final ActivityInfo activity = app.activityInfo;

final ComponentName name = new ComponentName(activity.applicationInfo.packageName, activity.name);

intent.addCategory(Intent.CATEGORY\_LAUNCHER);

intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK |Intent.FLAG\_ACTIVITY\_RESET\_TASK\_IF\_NEEDED);

intent.setComponent(name);

v.getContext().startActivity(intent);

break;

}

}

break;

}

else {

intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("http://www.twitter.com"));

startActivity(intent);

break;

}

//sending email with advertising message

case R.id.btnShareEmail:

intent = new Intent(Intent.ACTION\_SEND);

intent.putExtra(Intent.EXTRA\_TEXT, adMessage);

intent.setType("application/twitter");

startActivity(intent);

break;

}

}

}

**Importwindow.java**

package com.comp380.texttospeech;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

public class importwindow extends Activity implements OnClickListener {

Button btnClearSaved, btnDone, btnNavMain, btnNavSaved, btnNavShare,

btnNavImport;

ImageView sms, email;

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.importwindow);

btnNavMain = (Button) findViewById(R.id.btnNavMain);

btnNavMain.setOnClickListener(this);

btnNavSaved = (Button) findViewById(R.id.btnNavSaved);

btnNavSaved.setOnClickListener(this);

btnNavShare = (Button) findViewById(R.id.btnNavShare);

btnNavShare.setOnClickListener(this);

btnNavImport = (Button) findViewById(R.id.btnNavImport);

btnNavImport.setOnClickListener(this);

sms = (ImageView) findViewById(R.id.sms);

sms.setOnClickListener(this);

email = (ImageView) findViewById(R.id.email);

email.setOnClickListener(this);

}

public void onClick(View v) {

// TODO Auto-generated method stubIntent intent = new

if (v.getId() == R.id.btnNavSaved)

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

else if (v.getId() == R.id.btnNavMain)

startActivity(new Intent(this, Text\_to\_speechActivity.class));

else if (v.getId() == R.id.btnNavShare)

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

else if (v.getId() == R.id.sms) {

Intent sms = new Intent(Intent.ACTION\_MAIN);

sms.addCategory(Intent.CATEGORY\_LAUNCHER);

sms.setClassName("com.android.mms",

"com.android.mms.ui.ConversationList");

startActivity(sms);

}

else if (v.getId() == R.id.email) {

Intent intent = getPackageManager().getLaunchIntentForPackage(

"com.android.email");

startActivity(intent);

}

}

}

**MyBaseAdapter.java**

package com.comp380.texttospeech;

import java.util.ArrayList;

import com.comp380.texttospeech.R;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.TextView;

public class MyBaseAdapter extends BaseAdapter {

private static ArrayList<FileInformation> fileArrayList;

private LayoutInflater mInflater;

public MyBaseAdapter(Context context, ArrayList<FileInformation> file\_info) {

fileArrayList = file\_info;

mInflater = LayoutInflater.from(context);

}

public int getCount() {

return fileArrayList.size();

}

public Object getItem(int position) {

return fileArrayList.get(position);

}

public long getItemId(int position) {

return position;

}

public View getView(int position, View convertView, ViewGroup parent) {

ViewHolder holder;

if(convertView == null) {

convertView = mInflater.inflate(R.layout.custom\_row\_view, null);

holder = new ViewHolder();

holder.txtPrevName = (TextView) convertView.findViewById(R.id.PrevName);

//holder.txtDestinationFileName = (TextView) convertView.findViewById(R.id.DestinationFileName);

convertView.setTag(holder);

} else {

holder = (ViewHolder) convertView.getTag();

}

holder.txtPrevName.setText(fileArrayList.get(position).getPrevName());

//holder.txtDestinationFileName.setText(fileArrayList.get(position).getDestinationFileName());

return convertView;

}

static class ViewHolder {

TextView txtPrevName;

//TextView txtDestinationFileName;

}

}

**FileInformation.java**

**package** com.comp380.texttospeech;

**import** java.io.Serializable;

**public** **class** FileInformation **implements** Serializable {

**private** **static** **final** **long** *serialVersionUID* = 1L;

**private** String mPrevName = "";

**private** String mDestinationFileName = "";

**public** FileInformation(String prevName, String destinationFileName)

{

mPrevName = prevName;

mDestinationFileName = destinationFileName;

}

**public** **void** setPrevName(String previewName) {

**this**.mPrevName = previewName;

}

**public** String getPrevName() {

**return** mPrevName;

}

**public** **void** setDestinationFileName(String destFileName) {

**this**.mDestinationFileName = destFileName;

}

**public** String getDestinationFileName() {

**return** mDestinationFileName;

}

}

**9. Layouts Code**

**Main2.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

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

android:id=*"@+id/relativeLayout0"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:background=*"@drawable/background"* >

<TextView

android:id=*"@+id/textView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_centerHorizontal=*"true"*

android:text=*"@string/lblMainMenu"*

android:textSize=*"25dp"* />

<ImageView

android:id=*"@+id/imageView4"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"1dp"*

android:layout\_below=*"@+id/textView1"*

android:layout\_centerHorizontal=*"true"*

android:background=*"#FFFFFF"* />

<Button

android:id=*"@+id/btnSaved2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/txtInput2"*

android:layout\_below=*"@+id/imageView4"*

android:layout\_marginTop=*"5dp"*

android:text=*"@string/btnSaved"* />

<Button

android:id=*"@+id/btnSpeak2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignBaseline=*"@+id/btnSaved2"*

android:layout\_alignBottom=*"@+id/btnSaved2"*

android:layout\_alignParentRight=*"true"*

android:layout\_marginRight=*"5dp"*

android:text=*"@string/btnSpeak"* />

<EditText

android:id=*"@+id/txtInput2"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignBottom=*"@+id/progressBar1"*

android:layout\_alignParentLeft=*"true"*

android:layout\_below=*"@+id/imageView5"*

android:layout\_marginLeft=*"5dp"*

android:layout\_marginRight=*"5dp"*

android:layout\_marginTop=*"5dp"*

android:gravity=*"left"*

android:hint=*"Type text here"*

android:inputType=*"textMultiLine"*

android:textColor=*"#808080"* />

<RelativeLayout

android:id=*"@+id/relativeLayout1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentBottom=*"true"*

android:layout\_alignParentLeft=*"true"*

android:layout\_alignParentRight=*"true"*

android:layout\_below=*"@+id/txtInput2"* >

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentBottom=*"true"*

android:layout\_alignParentLeft=*"true"*

android:background=*"@drawable/nav\_layout\_background\_sm"* />

<RelativeLayout

android:id=*"@+id/relativeLayout2"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_above=*"@+id/imageView1"*

android:layout\_alignParentLeft=*"true"*

android:background=*"#000000"* >

<Button

android:id=*"@+id/btnNavMain2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentLeft=*"true"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginLeft=*"21dp"*

android:background=*"#000000"*

android:clickable=*"true"*

android:text=*"@string/btnNavMain"*

android:textColor=*"#FFFFFF"* />

<Button

android:id=*"@+id/btnNavSaved2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginLeft=*"43dp"*

android:layout\_toRightOf=*"@+id/btnNavMain2"*

android:background=*"#000000"*

android:text=*"@string/btnNavSaved"*

android:textColor=*"#FFFFFF"* />

<Button

android:id=*"@+id/btnNavShare2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginLeft=*"43dp"*

android:layout\_toRightOf=*"@+id/btnNavSaved2"*

android:background=*"#000000"*

android:text=*"@string/btnNavShare"*

android:textColor=*"#FFFFFF"* />

<Button

android:id=*"@+id/btnNavImport2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentRight=*"true"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginRight=*"21dp"*

android:background=*"#000000"*

android:text=*"@string/btnNavImport"*

android:textColor=*"#FFFFFF"* />

</RelativeLayout>

<ImageView

android:id=*"@+id/imageView3"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"10dp"*

android:layout\_above=*"@+id/relativeLayout2"*

android:layout\_alignParentLeft=*"true"*

android:background=*"#000000"* />

<ImageView

android:id=*"@+id/imageView2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_centerHorizontal=*"true"*

android:layout\_centerVertical=*"true"*

android:src=*"@drawable/speakeasy"* />

</RelativeLayout>

<ImageView

android:id=*"@+id/imageView5"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"1dp"*

android:layout\_alignParentLeft=*"true"*

android:layout\_below=*"@+id/btnSaved2"*

android:layout\_marginTop=*"5dp"*

android:background=*"#FFFFFF"* />

<!--

<ToggleButton

android:id="@+id/button11"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/headphones\_down"

android:layout\_centerHorizontal="true"

android:textOff="SHOW"

android:textOn="HIDE" />

<ImageButton

android:id="@+id/imageButton12"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/button11"

android:layout\_toRightOf="@+id/btnSaved2"

android:src="@drawable/keyboard\_32x32" />

-->

<ProgressBar

android:id=*"@+id/progressBar1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_centerHorizontal=*"true"*

android:layout\_centerVertical=*"true"*

android:visibility=*"invisible"* />

<!--

<ImageView

android:id="@+id/play"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignRight="@+id/textView1"

android:layout\_alignTop="@+id/headphones\_down"

android:src="@drawable/play\_32x32" />

<ImageView

android:id="@+id/btnListItems"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/btnClear"

android:layout\_marginLeft="15dp"

android:layout\_toRightOf="@+id/btnSaved2"

android:src="@drawable/list\_32x32" />

<ImageView

android:id="@+id/btnDeleteList"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/play"

android:layout\_toLeftOf="@+id/btnSpeak2"

android:src="@drawable/delete\_list\_32x32" />

-->

<ImageView

android:id=*"@+id/headphones\_down"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignBottom=*"@+id/txtInput2"*

android:layout\_alignRight=*"@+id/txtInput2"*

android:layout\_marginBottom=*"16dp"*

android:layout\_marginRight=*"16dp"*

android:background=*"@drawable/headphones\_selector"* />

<ImageView

android:id=*"@+id/btnClear"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_above=*"@+id/progressBar1"*

android:layout\_alignLeft=*"@+id/headphones\_down"*

android:src=*"@drawable/clear\_32x32"* />

<ImageView

android:id=*"@+id/keyboard"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_above=*"@+id/btnClear"*

android:layout\_alignLeft=*"@+id/btnClear"*

android:layout\_marginBottom=*"14dp"*

android:background=*"@drawable/keyboard\_selector"* />

</RelativeLayout>

**saved\_audio.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

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

android:id=*"@+id/relativeLayout1"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:background=*"@drawable/background"* >

<TextView

android:id=*"@+id/textViewSavedAudio"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_centerHorizontal=*"true"*

android:text=*"@string/lblSavedAudio"*

android:textSize=*"25dp"* />

<ImageView

android:id=*"@+id/imageViewHL2"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"1dp"*

android:layout\_alignParentLeft=*"true"*

android:layout\_below=*"@+id/textViewSavedAudio"*

android:background=*"#FFFFFF"* />

<Button

android:id=*"@+id/btnClearSaved"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignRight=*"@+id/srListView"*

android:layout\_below=*"@+id/imageViewHL2"*

android:layout\_marginLeft=*"5dp"*

android:layout\_marginTop=*"5dp"*

android:layout\_marginBottom=*"5dp"*

android:text=*"@string/btnClearSaved"* />

<Button

android:id=*"@+id/btnDoneSaved3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignBaseline=*"@+id/btnClearSaved"*

android:layout\_alignBottom=*"@+id/btnClearSaved"*

android:layout\_alignLeft=*"@+id/srListView"*

android:text=*"@string/btnDoneSaved"* />

<ImageView

android:id=*"@+id/imageViewHL"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"1dp"*

android:layout\_alignBottom=*"@+id/btnDoneSaved3"*

android:layout\_alignParentLeft=*"true"*

android:background=*"#FFFFFF"* />

<ListView

android:id=*"@+id/srListView"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"324dp"*

android:layout\_above=*"@+id/ImageView02"*

android:layout\_alignParentLeft=*"true"*

android:layout\_below=*"@+id/btnClearSaved"*

android:layout\_marginLeft=*"5dp"*

android:layout\_marginRight=*"5dp"*

android:layout\_marginTop=*"5dp"*

android:gravity=*"left"*

android:textColor=*"#808080"* >

</ListView>

<ImageView

android:id=*"@+id/imageView3"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"10dp"*

android:layout\_alignParentBottom=*"true"*

android:layout\_alignParentLeft=*"true"*

android:background=*"#000000"* />

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_above=*"@+id/imageView3"*

android:layout\_alignParentLeft=*"true"*

android:background=*"@drawable/nav\_layout\_background\_sm"* />

<ImageView

android:id=*"@+id/ImageView01"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"10dp"*

android:layout\_above=*"@+id/imageView1"*

android:layout\_alignParentLeft=*"true"*

android:background=*"#000000"* />

<ImageView

android:id=*"@+id/ImageView02"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"10dp"*

android:layout\_above=*"@+id/ImageView01"*

android:layout\_alignParentLeft=*"true"*

android:background=*"#000000"* />

<RelativeLayout

android:id=*"@+id/relativeLayout3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/srListView"*

android:layout\_alignTop=*"@+id/ImageView01"* >

<Button

android:id=*"@+id/btnNavMain3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentLeft=*"true"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginLeft=*"21dp"*

android:background=*"#000000"*

android:clickable=*"true"*

android:text=*"@string/btnNavMain"*

android:textColor=*"#FFFFFF"* />

<Button

android:id=*"@+id/btnNavSaved3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginLeft=*"43dp"*

android:layout\_toRightOf=*"@+id/btnNavMain3"*

android:background=*"#000000"*

android:text=*"@string/btnNavSaved"*

android:textColor=*"#FFFFFF"* />

<Button

android:id=*"@+id/btnNavShare3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginLeft=*"43dp"*

android:layout\_toRightOf=*"@+id/btnNavSaved3"*

android:background=*"#000000"*

android:text=*"@string/btnNavShare"*

android:textColor=*"#FFFFFF"* />

</RelativeLayout>

<Button

android:id=*"@+id/btnNavImport3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignTop=*"@+id/ImageView01"*

android:layout\_alignRight=*"@+id/srListView"*

android:background=*"#000000"*

android:layout\_marginRight=*"10dp"*

android:text=*"@string/btnNavImport"*

android:textColor=*"#FFFFFF"* />

</RelativeLayout>

**importwindow.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

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

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:background=*"@drawable/background"*

android:orientation=*"vertical"* >

<LinearLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"Import"*

android:textSize=*"25dp"* />

</LinearLayout>

<ImageView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"1dp"*

android:background=*"#FFFFFF"* />

<TextView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginBottom=*"5dp"*

android:layout\_marginLeft=*"5dp"*

android:layout\_marginRight=*"5dp"*

android:layout\_marginTop=*"5dp"*

android:gravity=*"center"*

android:text=*"Copy text from SMS or e-mail and paste the text on the main screen."* />

<ImageView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"1dp"*

android:background=*"#FFFFFF"* />

<Gallery android:id=*"@+id/gallery1"*

android:layout\_width=*"fill\_parent"*

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

<LinearLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"352dp"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<ImageView

android:id=*"@+id/sms"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"66dp"*

android:src=*"@drawable/sms"* />

<ImageView

android:id=*"@+id/email"*

android:layout\_width=*"48dp"*

android:layout\_height=*"52dp"*

android:src=*"@drawable/email"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:background=*"#000000"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_weight=*"25"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/btnNavMain"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:background=*"#000000"*

android:text=*"@string/btnNavMain"*

android:textColor=*"#FFFFFF"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_weight=*"25"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/btnNavSaved"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:background=*"#000000"*

android:text=*"@string/btnNavSaved"*

android:textColor=*"#FFFFFF"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_weight=*"25"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/btnNavShare"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:background=*"#000000"*

android:text=*"@string/btnNavShare"*

android:textColor=*"#FFFFFF"* />

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_weight=*"25"*

android:gravity=*"center"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/btnNavImport"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:background=*"#000000"*

android:text=*"@string/btnNavImport"*

android:textColor=*"#FFFFFF"* />

</LinearLayout>

</LinearLayout>

<ImageView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"42dp"*

android:layout\_weight=*"0.06"*

android:background=*"#000000"* />

</LinearLayout>

**share.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

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

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:orientation=*"vertical"*

android:background=*"@drawable/background"* >

<LinearLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"*

android:gravity=*"center"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/lblShare"*

android:textSize=*"25dp"*/>

</LinearLayout>

<ImageView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"1dp"*

android:background=*"#FFFFFF"* />

<TextView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/lblShareMsg"*

android:layout\_marginRight=*"5dp"*

android:layout\_marginLeft=*"5dp"*

android:layout\_marginTop=*"5dp"*

android:layout\_marginBottom=*"5dp"*

android:gravity=*"center"*/>

<ImageView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"1dp"*

android:background=*"#FFFFFF"* />

<Button

android:id=*"@+id/btnShareEmail"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/btnShareEmail"*

android:layout\_marginTop=*"10dp"*

android:layout\_marginLeft=*"5dp"*

android:layout\_marginRight=*"5dp"*/>

<Button

android:id=*"@+id/btnShareTweet"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/btnShareTweet"*

android:layout\_marginTop=*"5dp"*

android:layout\_marginLeft=*"5dp"*

android:layout\_marginRight=*"5dp"*/>

<Button

android:id=*"@+id/btnShareFacebook"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/btnShareFacebook"*

android:layout\_marginTop=*"5dp"*

android:layout\_marginLeft=*"5dp"*

android:layout\_marginRight=*"5dp"*

android:layout\_marginBottom=*"173dp"*/>

<ImageView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"10dp"*

android:background=*"#000000"* />

<LinearLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"*

android:gravity=*"center"*

android:background=*"#000000"*>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"*

android:gravity=*"center"*

android:layout\_weight=*"25"* >

<Button

android:id=*"@+id/btnNavMain"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/btnNavMain"*

android:textColor=*"#FFFFFF"*

android:background=*"#000000"*/>

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"*

android:gravity=*"center"*

android:layout\_weight=*"25"* >

<Button

android:id=*"@+id/btnNavSaved"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/btnNavSaved"*

android:textColor=*"#FFFFFF"*

android:background=*"#000000"*/>

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"*

android:gravity=*"center"*

android:layout\_weight=*"25"* >"

<Button

android:id=*"@+id/btnNavShare"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/btnNavShare"*

android:textColor=*"#FFFFFF"*

android:background=*"#000000"*/>

</LinearLayout>

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"*

android:gravity=*"center"*

android:layout\_weight=*"25"* >

<Button

android:id=*"@+id/btnNavImport"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/btnNavImport"*

android:textColor=*"#FFFFFF"*

android:background=*"#000000"*/>

</LinearLayout>

</LinearLayout>

<ImageView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:background=*"#000000"* />

</LinearLayout>

**custom\_view\_row.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

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

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:orientation=*"vertical"*

android:paddingBottom=*"10dip"*

android:paddingLeft=*"10dip"*

android:paddingRight=*"10dip"*

android:paddingTop=*"10dip"* >

<TextView

android:id=*"@+id/PrevName"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:textColor=*"#C20000"*

android:layout\_marginLeft=*"33dp"*

android:textSize=*"18sp"*

android:textStyle=*"bold"*

android:text=*"test"* />

<!--

<TextView

android:id="@+id/DestinationFileName"

android:layout\_width="wrap\_content"

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

-->

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignBottom=*"@+id/PrevName"*

android:layout\_alignParentLeft=*"true"*

android:layout\_marginLeft=*"1dp"*

android:src=*"@drawable/audio\_wav"* />

</RelativeLayout>

**buttonshape.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<shape xmlns:android=*"http://schemas.android.com/apk/res/android"* android:shape=*"rectangle"*>

<gradient android:startColor=*"#808080"* android:endColor=*"#808080"* android:angle=*"270"* />

<corners android:bottomRightRadius=*"7dp"* android:bottomLeftRadius=*"7dp"* android:topLeftRadius=*"7dp"* android:topRightRadius=*"7dp"* />

</shape>

**textview.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

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

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:padding=*"6dp"*

android:textSize=*"24dp"* />

**textshape.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<shape xmlns:android=*"http://schemas.android.com/apk/res/android"* android:shape=*"rectangle"*>

<gradient android:startColor=*"#FFFFFF"* android:endColor=*"#FFFFFF"* android:angle=*"270"* />

<corners android:bottomRightRadius=*"10dp"* android:bottomLeftRadius=*"10dp"* android:topLeftRadius=*"10dp"* android:topRightRadius=*"10dp"* />

</shape>

**headphones\_selector.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<selector xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<!-- pressed -->

<item android:state\_pressed=*"true"* android:drawable=*"@drawable/headphones\_down\_darker"* />

<!-- default/unchecked -->

<item android:drawable=*"@drawable/headphones\_down"* />

</selector>

**keyboard\_selector.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<selector xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<!-- pressed -->

<item android:state\_pressed=*"true"* android:drawable=*"@drawable/keyboard\_32x32\_darker"* />

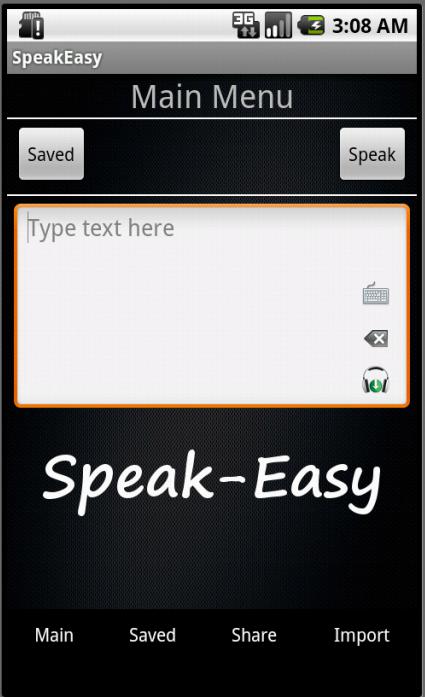
<!-- default/unchecked -->

<item android:drawable=*"@drawable/keyboard\_32x32"* />

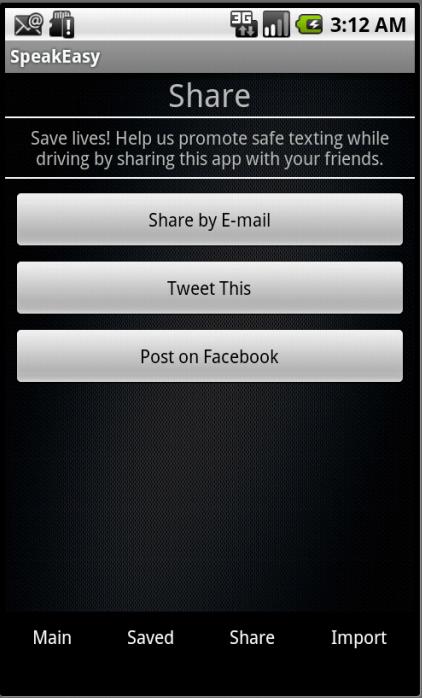
</selector>

**10. Diagrams**

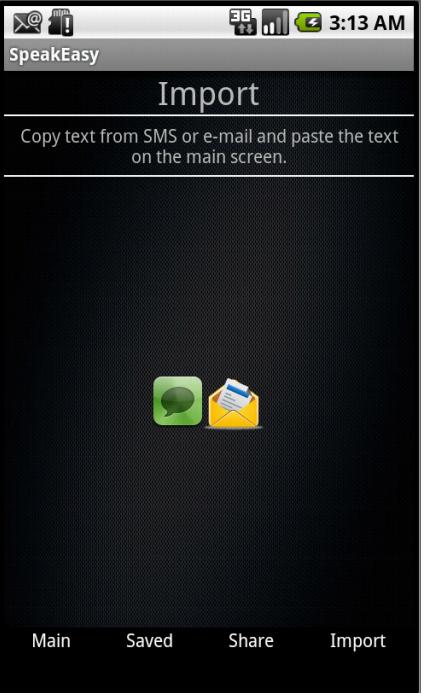
**Main Menu**

****

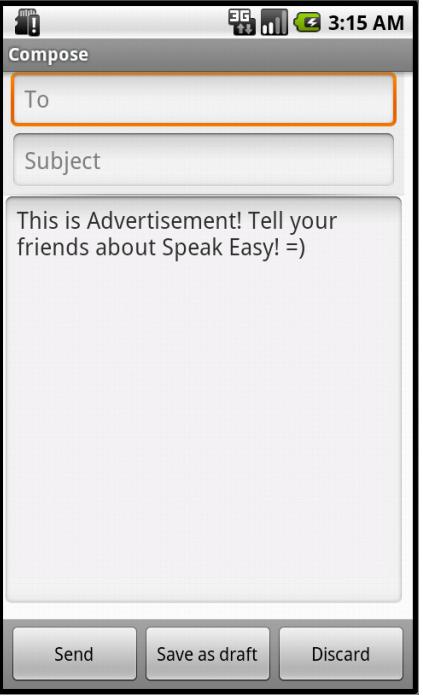
**Share Screen**

****

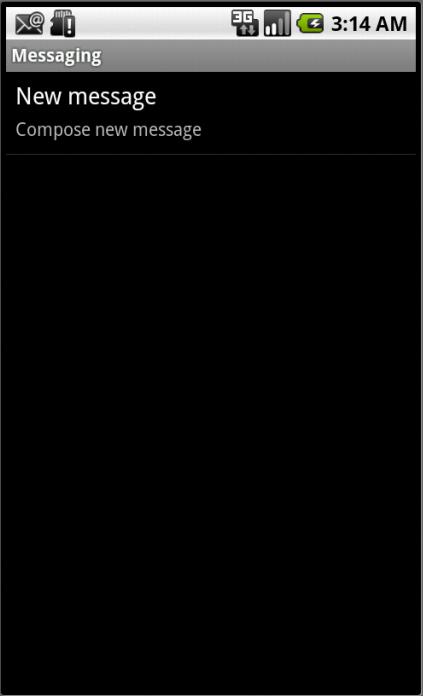
**Import Screen**

****

**Share by Email screen**

****

**Import from SMS**

****