SHILLONG COLLEGE

BOYCE ROAD, LAITUMKHRAH
SHILLONG - 793003, MEGHALAYA

ESTD - 1956

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF BACHELOR OF COMPUTER SCIENCE AND APPLICATIONS

Submitted By,

Student Name : ALDAREEN KHYRIEM
Roll No : P1400003
Reg.No : 14519 of 2013 – 2014

Under the Guidence of

Sir B.Mukhim
Assistant Professor
Department of Computer Science And Applications

NORTH EASTERN HILLS UNIVERSITY

CERTIFIED THAT THIS IS A BONAFIDE RECORD OF THE PROJECT

ENTITLED

NOTES TAKING

Submitted for the partial fulfillment for the award of degree of
Bachelor of Computer Applications

By,

Student Name : ALDAREEN KHYRIEM

Roll No : P1400003

Reg.No : 14519 of 2013 – 2014

Project Guide	Head Of Department	Examiner
Sir Bantei Mukhim	Miss AiomMitri	

Viva Voce held on:_____

ACKNOWLEDGEMENT

Doing this project, writing and compiling it into a project report was not an individual effort. Without the guidance, encouragement and help of many people this endeavor of mine would not have been successful.

First of all I convey all my thanks to the Great Almighty, the author of knowledge and wisdom, for his countless love and blessings.

I would like to express my sincere thanks to the Principal of ShillongCollege, Shillong, Dr. K. D. Ramsiej for giving me an opportunity to conduct this project and making it a successful one. I extend my deepest gratitudeand thanks to my guide, Sir B. Mukhimfor guiding me in different matters and solutions regarding the topic and for encouraging me from time to time till I complete my project.I would also like to thank and appreciate the support provided by our Head of Department Miss A. Mitri who has been a very helpful person.

Last but not the least my heartiest thanks goes to my friends and family members for their constant help and support that they rendered all the time till I could complete my fruitful work.

THANK YOU

CONTENTS

SYNOPSIS

PROJECT NAME : Notes Taking

OBJECTIVE : Notes Taking is a simple app. It gives users a quick and simple Notepad editing experience when user write notes and to-do lists. Taking notes with Notes Taking is easier than any other app.

FEATURES :

- The text option allows for as many characters as we are willing to type. Once saved, we can edit or delete the note through our device's menu button.
- Get things done : make to-do's and checklists.
- Notes Taking allows users to convert audio note to text by using google's voice recognition.
- It also has support for voice recording, so that important speech can be recorded.
- Audio note can be save on external SD card depends on different format (mp3 and amr).

Existing System & Proposed System

At present there are plenty of note-taking apps for android like Evernote, Google Keep, Shoutout Loud, etc. As far as it is concerned the existing Apps has both the advantages and disadvantages. Some apps have audio note but no voice note while some others have voice note but no audio note. It may be mentioned that as per the existing system more than one apps is needed by the user and this is found to be space consuming.

To overcome this problem a new proposed system could be adopted in which a user can use only one app which could make available all features like text note, audio note and voice note only in a single app. This would be very helpful and friendly to a user.

Hardware & Software Requirements

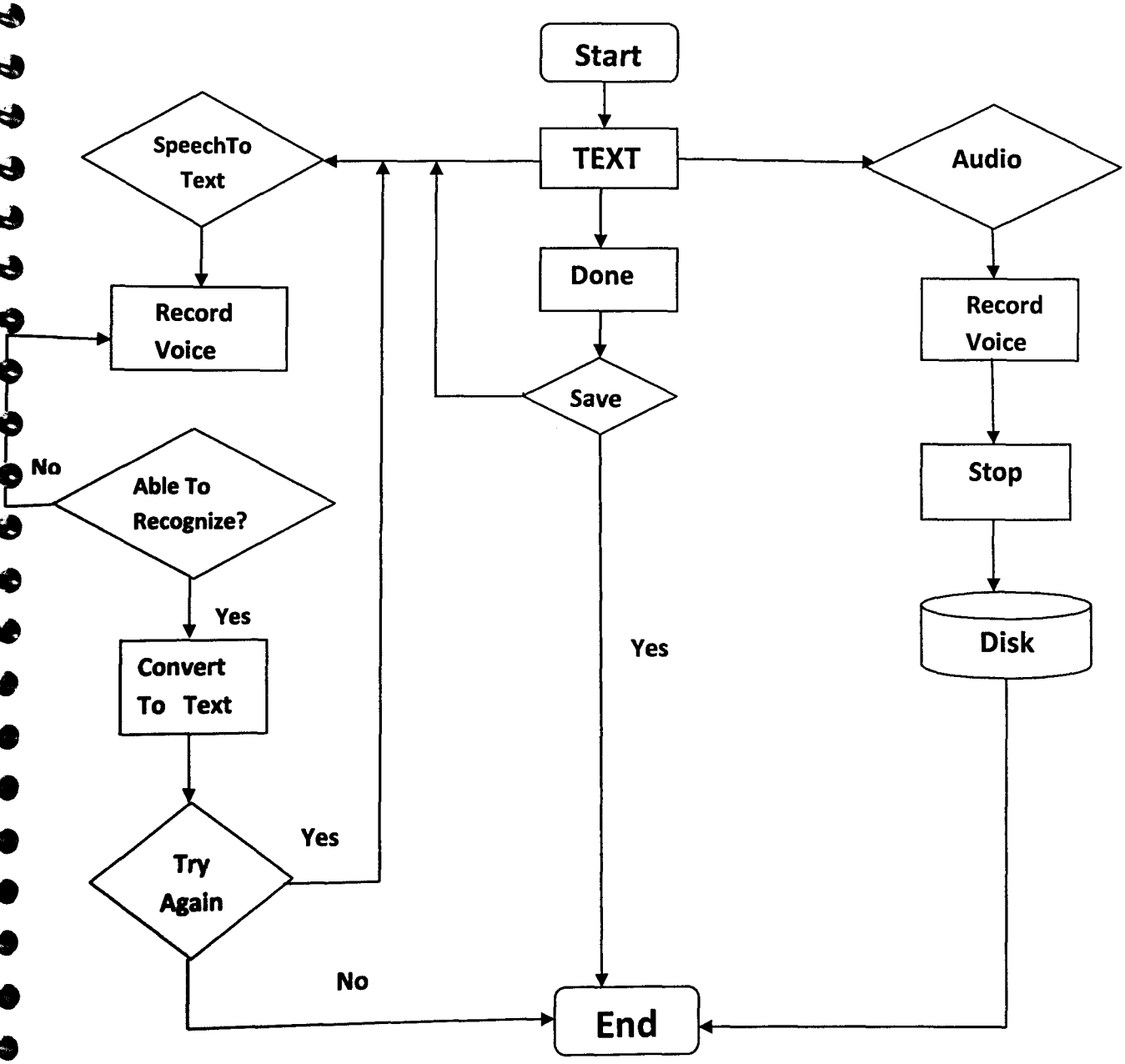
Software Requirements

- Android Studio
- Microphone
- JDK

Hardware Requirement

- RAM : 512 MB
- CPU : 800Mhz

FLOWCHART


Start
TEXT
SpeechTo Text
Audio
Done
Record Voice
Record Voice
Save
No
Able To Recognize?
Stop
Yes
Disk
Convert To Text
Yes
Yes
Try Again
No
End

SOURCE CODE

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.aldareen.notestaking"
android:versionCode="1"
android:versionName="1.0">

<uses-sdk
android:minSdkVersion="15"
android:targetSdkVersion="22"/>

<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.WAKE_LOCK" />

<application
android:allowBackup="true"
android:icon="@mipmap/img19"
android:label="Notes Taking"
android:theme="@style/AppTheme" >

<activity
android:screenOrientation="portrait"
android:name=".MainFragmentActivity"
android:label="@string/app_name" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.DEFAULT" />
</intent-filter>
</activity>

<activity
android:screenOrientation="portrait"
android:name=".SplashActivity"
android:label="@string/title_activity_splash" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>

<activity
android:screenOrientation="portrait"
android:name=".ListText"
```

```
android:uiOptions="splitActionBarWhenNarrow" />

<activity
android:screenOrientation="portrait"
android:name=".ViewNote"
android:uiOptions="splitActionBarWhenNarrow" />

<activity
android:screenOrientation="portrait"
android:name=".ReminderEditActivity"
android:label="@string/edit_reminder_title"/>

<receiver android:name=".OnBootReceiver">
<intent-filter>
<action android:name="android.intent.action.BOOT_COMPLETED" />
</intent-filter>
</receiver>

<receiver android:name=".OnAlarmReceiver" />
<service android:name=".ReminderService" />

</application>
</manifest>
```

SplashActivity.java

```
package com.example.aldareen.notestaking;

/**
 * Created by Aldareen on 2/29/2016.
 */

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.TextView;

public class SplashActivity extends Activity {
    ImageView photo;
    TextView display;
    Animation anim,animfade;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

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

        Thread timer=new Thread(){
public void run(){
try{
sleep(6000);
                }catch(InterruptedException e){
                    e.printStackTrace();
                }
finally{
                    Intent i=new
Intent(SplashActivity.this,MainFragmentActivity.class);
                    startActivity(i);
                }
            }
        };
        timer.start();

photo=(ImageView)findViewById(R.id.ivPhoto);
display=(TextView)findViewById(R.id.tvDisplay);


anim= AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.fade_in_bounce);
photo.setVisibility(View.VISIBLE);
photo.startAnimation(anim);

animfade= AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.fade);
display.setVisibility(View.VISIBLE);
display.startAnimation(animfade);

    }

@Override
protected void onPause() {
super.onPause();
        finish();
    }
}
```

res/anim/fade.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/accelerate_interpolator">

<alpha
android:duration="2000"
android:fromAlpha="0.0"
android:toAlpha="1.0"
/>

<alpha
```

```
android:startOffset="2000"
android:duration="2000"
android:fromAlpha="1.0"
android:toAlpha="0.0"
/>
</set>
```

res/anim/fade_in_bounce.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:fillAfter="true"
android:interpolator="@android:anim/bounce_interpolator">

<alpha
android:duration="5000"
android:fromAlpha="0.0"
android:toAlpha="1.0"
android:interpolator="@android:anim/accelerate_interpolator"/>

<scale
android:duration="5000"
android:fromXScale="1.0"
android:fromYScale="0.0"
android:toXScale="1.0"
android:toYScale="1.0"/>
</set>
```

res/layout/activity_splash.xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
android:background="@drawable/img5">

<ImageView
android:layout_width="120dp"
android:layout_height="120dp"
android:id="@+id/ivPhoto"
android:src="@drawable/img3"
android:layout_above="@+id/TextView"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true"
android:layout_marginBottom="42dp" />

<TextView
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="NOTES TAKING  APP"
android:id="@+id/tvDisplay"
android:textColor="#111fff"
android:textSize="30dp"
android:textStyle="bold|italic"
android:layout_alignParentTop="true"
android:layout_alignRight="@+id/ivPhoto"
android:layout_alignEnd="@+id/ivPhoto"
android:layout_marginRight="71dp"
android:layout_marginEnd="71dp"
android:textIsSelectable="false" />

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Loading...."
android:id="@+id/TextView"
android:textColor="#f3f40b"
android:textStyle="bold|italic"
android:textSize="30dp"
android:layout_marginBottom="44dp"
android:layout_alignParentBottom="true"
android:layout_alignRight="@+id/ivPhoto"
android:layout_alignEnd="@+id/ivPhoto" />

</RelativeLayout>
```

OUTPUT :

MainFragmentActivity.java

```
package com.example.aldareen.notestaking;

/**
 * Created by Aldareen on 2/29/2016.
 */

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.TextView;

public class MainFragmentActivity extends FragmentActivity {

    MainLayout mLayout;
private ListView lvMenu;
private String[] lvMenuItems;
    Button btMenu,todo;
    TextView tvTitle;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

mLayout=(MainLayout)this.getLayoutInflater().inflate(R.layout.activi
ty_main,null);
        setContentView(mLayout);
lvMenuItems=getResources().getStringArray(R.array.menu_items);
lvMenu=(ListView)findViewById(R.id.menu_listview);
lvMenu.setAdapter(new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, lvMenuItems));

lvMenu.setOnItemClickListener(new OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                onMenuItemClick(parent, view, position, id);
            }
        });

btMenu=(Button)findViewById(R.id.button_menu);
btMenu.setOnClickListener(new OnClickListener() {
@Override
```


```
public void onClick(View v) {
                toggleMenu(v);
            }
        });

tvTitle=(TextView)findViewById(R.id.activity_main_content_title);
        FragmentManager
fm=MainFragmentActivity.this.getSupportFragmentManager();
        FragmentTransaction ft=fm.beginTransaction();
        Text fragment=new Text();
        ft.add(R.id.activity_main_content_fragment,fragment);
        ft.commit();

todo=(Button)findViewById(R.id.bTodo);
todo.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
                Intent iTodo=new
Intent(MainFragmentActivity.this,ReminderEditActivity.class);
                startActivity(iTodo);
            }
        });
    }

public void toggleMenu(View v){
mLayout.toggleMenu();
    }

private void onMenuItemClick(AdapterView<?> parent,View view,int
position,long id){
        String selectedItem=lvMenuItems[position];
        String currentItem=tvTitle.getText().toString();
if(selectedItem.compareTo(currentItem)==0){
mLayout.toggleMenu();
return;
        }

        FragmentManager
fm=MainFragmentActivity.this.getSupportFragmentManager();
        FragmentTransaction ft=fm.beginTransaction();
        Fragment fragment=null;
if (selectedItem.compareTo("Text Note")==0){
            fragment=new Text();
        }
else if (selectedItem.compareTo("Audio Note")==0){
            fragment=new Audio();
        }
else if (selectedItem.compareTo("Speech To Text Note")==0){
            fragment=new SpeechToText();
        }
else if (selectedItem.compareTo("About")==0){
            fragment=new About();
        }
else if (selectedItem.compareTo("Exit")==0){
            AlertDialog.Builder alert=new AlertDialog.Builder(this);
            alert.setMessage("Do You Want To Exit?");
```

```
            alert.setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
                    finish();
                }
            });
            alert.setNegativeButton("No", new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
                    dialog.cancel();
                }
            });
            alert.create().show();
        }

if (fragment!=null){

ft.replace(R.id.activity_main_content_fragment,fragment);
            ft.commit();
tvTitle.setText(selectedItem);
        }
mLayout.toggleMenu();
    }

public void onBackPressed(){
if(mLayout.isMenuShown()){
mLayout.toggleMenu();
        }else {
super.onBackPressed();
        }
    }
}
```

MainLayout.java

```
package com.example.aldareen.notestaking;

/**
 * Created by Aldareen on 2/29/2016.
 */

import android.content.Context;
import android.os.Handler;
import android.util.AttributeSet;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.view.animation.Interpolator;
import android.widget.LinearLayout;
import android.widget.Scroller;

public class MainLayout extends LinearLayout{
```

```
private static final int SLIDING_DURATION=500;
private static final int QUERY_INTERVAL=16;
int mainLayoutWidth;
private View menu;
private View content;
private static int menuRightMargin=15;
private enum MenuState{
HIDING, HIDDEN, SHOWING, SHOWN,
    };

private int contentXoffset;
private MenuState currentMenuState= MenuState.HIDDEN;
private Scroller menuScroller = new Scroller(this.getContext(),new
EaseInInterpolator());
private Runnable menuRunnable=new MenuRunnable();
private Handler menuHandler=new Handler();
int prevX=0;
boolean isDragging=false;
int lastDiffX=0;

public MainLayout(Context context,AttributeSet attrs){
super(context,attrs);
    }

public MainLayout(Context context){
super(context);
    }

protected void onMeasure(int widthMeasureSpec,int
heightMeasureSpec){
super.onMeasure(widthMeasureSpec, heightMeasureSpec);
mainLayoutWidth=MeasureSpec.getSize(widthMeasureSpec);
menuRightMargin=mainLayoutWidth*10/100;
    }

protected void onAttachedToWindow(){
super.onAttachedToWindow();
menu=this.getChildAt(0);
content=this.getChildAt(1);
content.setOnTouchListener(new OnTouchListener() {
@Override
public boolean onTouch(View v, MotionEvent event) {
return MainLayout.this.onContentTouch(v,event);
            }
        });
menu.setVisibility(View.GONE);
    }

protected void onLayout(boolean changed,int left,int top,int
right,int bottom){
if (changed){
            LayoutParams
contentLayoutParams=(LayoutParams)content.getLayoutParams();
            contentLayoutParams.height=this.getHeight();
            contentLayoutParams.width=this.getWidth();
            LayoutParams
```

```
menuLayoutParams=(LayoutParams)menu.getLayoutParams();
            menuLayoutParams.height=this.getHeight();
            menuLayoutParams.width=this.getWidth()-menuRightMargin;
        }

menu.layout(left, top, right - menuRightMargin, bottom);
content.layout(left + contentXoffset, top, right + contentXoffset,
bottom);
    }

public void toggleMenu(){
if (currentMenuState == MenuState.HIDING ||
currentMenuState == MenuState.SHOWING)
return;

switch (currentMenuState){
case HIDDEN:
currentMenuState = MenuState.SHOWING;
menu.setVisibility(View.VISIBLE);
menuScroller.startScroll(0,0,menu.getLayoutParams().width,0,SLIDING_
DURATION);
break;
case SHOWN:
currentMenuState= MenuState.HIDING;
menuScroller.startScroll(contentXoffset,0,-
contentXoffset,0,SLIDING_DURATION);
break;

default:
break;
        }
menuHandler.postDelayed(menuRunnable,QUERY_INTERVAL);
this.invalidate();
    }

protected class MenuRunnable implements Runnable{
public void run(){
boolean isScrolling=menuScroller.computeScrollOffset();
            adjustContentPosition(isScrolling);
        }
    }

private void adjustContentPosition(boolean isScrolling){
int scrollerXoffset=menuScroller.getCurrX();
content.offsetLeftAndRight(scrollerXoffset-contentXoffset);
contentXoffset=scrollerXoffset;
this.invalidate();
if (isScrolling)
menuHandler.postDelayed(menuRunnable,QUERY_INTERVAL);
else
            this.onMenuSlidingComplete();
    }

private void onMenuSlidingComplete(){
switch (currentMenuState){
case SHOWING:
```

```
currentMenuState= MenuState.SHOWN;
break;
case HIDING:
currentMenuState= MenuState.HIDDEN;
menu.setVisibility(View.GONE);
break;
default:
return;
        }
    }

protected class EaseInInterpolator implements Interpolator{
public float getInterpolation(float t){
return(float) Math.pow(t-1,5)+1;
        }
    }

public boolean isMenuShown(){
return currentMenuState== MenuState.SHOWN;
    }

public boolean onContentTouch(View v,MotionEvent event){
if (currentMenuState == MenuState.HIDING ||
currentMenuState == MenuState.SHOWING)
return false;
int curX=(int)event.getRawX();
int diffX=0;

switch (event.getAction()){
case MotionEvent.ACTION_DOWN:
prevX=curX;
return true;
case MotionEvent.ACTION_MOVE:
if (!isDragging){
isDragging=true;
menu.setVisibility(View.VISIBLE);
                }
                diffX=curX-prevX;
if (contentXoffset+diffX<=0){
                    diffX=-contentXoffset;
                }
else if (contentXoffset+diffX>mainLayoutWidth-menuRightMargin){
                    diffX=mainLayoutWidth-menuRightMargin-
contentXoffset;
                }
content.offsetLeftAndRight(diffX);
contentXoffset+=diffX;
this.invalidate();
prevX=curX;
lastDiffX=diffX;
return true;
case MotionEvent.ACTION_UP:
                Log.d("MainLayout.java onContentTouch()","Up
lastDiffX "+lastDiffX);
if (lastDiffX>0){
currentMenuState= MenuState.SHOWING;
```

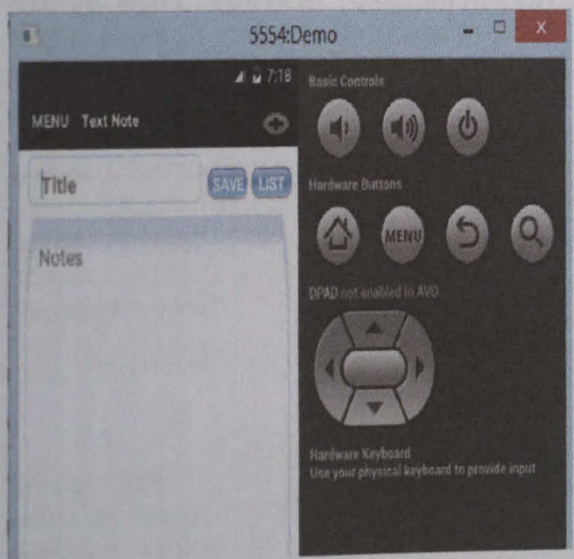
```
menuScroller.startScroll(contentXoffset,0,menu.getLayoutParams().wid
th-contentXoffset,0,SLIDING_DURATION);
                }else if(lastDiffX<0){
currentMenuState= MenuState.HIDING;
menuScroller.startScroll(contentXoffset,0,-
contentXoffset,0,SLIDING_DURATION);
                }
menuHandler.postDelayed(menuRunnable,QUERY_INTERVAL);
this.invalidate();
isDragging=false;
prevX=0;
lastDiffX=0;
return true;
default:
break;
        }
return false;
    }
}
```

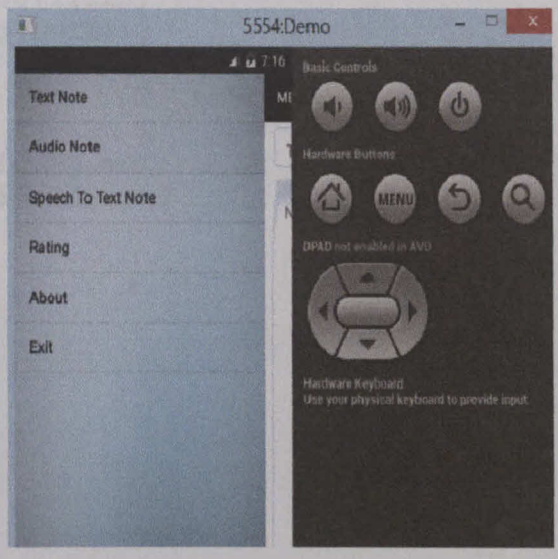
res/layout/activity_main.xml

```
<com.example.aldareen.notestaking.MainLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">

<ListView
android:id="@+id/menu_listview"
android:layout_width="fill_parent"
android:layout_height="match_parent"
android:background="@drawable/img4"
android:cacheColorHint="#00000000">
</ListView>
</LinearLayout>

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@android:color/black"
android:orientation="horizontal">
```

```
<Button
android:layout_width="70dp"
android:layout_height="wrap_content"
android:onClick="toggleMenu"
android:background="@android:color/black"
android:text="Menu"
android:textColor="@android:color/white"
android:id="@+id/button_menu"/>

<TextView
android:layout_width="0dp"
android:layout_height="wrap_content"
android:text="Text Note"
android:gravity="left"
android:textColor="@android:color/white"
android:id="@+id/activity_main_content_title"
android:layout_weight="1"
android:textSize="15dp"
android:textStyle="bold" />

<Button
android:layout_width="50dp"
android:layout_height="30dp"
android:gravity="right"
android:background="@android:drawable/ic_menu_add"
android:id="@+id/bTodo"/>
</LinearLayout>

<FrameLayout
android:id="@+id/activity_main_content_fragment"
android:layout_width="match_parent"
android:layout_height="match_parent">
</FrameLayout>

</LinearLayout>
</com.example.aldareen.notestaking.MainLayout>
```

OUTPUT :





Text.java

```
package com.example.aldareen.notestaking;

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

/**
 * Created by Aldareen on 2/29/2016.
 */

public class Text extends Fragment implements View.OnClickListener {

private long rowID;
private EditText title_edit;
private EditText note_edit;
private static final String TITLE="title";
private static final String NOTE="note";
    Button save,list;

public View onCreateView(LayoutInflater inflater,ViewGroup
container,Bundle savedInstanceState){
        View view=inflater.inflate(R.layout.add_note,null);
title_edit=(EditText)view.findViewById(R.id.titleEdit);
note_edit=(EditText)view.findViewById(R.id.noteEdit);
        Bundle extras=getActivity().getIntent().getExtras();
if(extras!=null){
rowID=extras.getLong("row_id");
title_edit.setText(extras.getString(TITLE));
note_edit.setText(extras.getString(NOTE));
        }


save=(Button)view.findViewById(R.id.bSave);
list=(Button)view.findViewById(R.id.bList);
save.setOnClickListener(this);
list.setOnClickListener(this);
return view;
    }



@Override
public void onClick(View v) {
switch (v.getId()){
```

```
case R.id.bSave:
if(title_edit.getText().length()!=0){
                    saveNote();
                    Toast.makeText(getView().getContext(),
"Successfully Saved!", Toast.LENGTH_LONG).show();
                }
else{
                    AlertDialog.Builder alert=new
AlertDialog.Builder(getView().getContext());
                    alert.setTitle("Title is required");
                    alert.setMessage("Put in a title for this
note");
                    alert.setPositiveButton("Okay",null);
                    alert.show();
                }
break;
case R.id.bList:
                Intent listmenu=new
Intent(getView().getContext(),ListText.class);
                startActivity(listmenu);
break;
        }
    }

private void saveNote(){
        DatabaseConnector dbConnector=new
DatabaseConnector(getView().getContext());
if(getActivity().getIntent().getExtras()==null){
            dbConnector.InsertNote(title_edit.getText().toString(),
note_edit.getText().toString());
        }
else{
            dbConnector.UpdateNote(rowID,
title_edit.getText().toString(), note_edit.getText().toString());
        }
    }

}
```

DatabaseConnector.java

```
package com.example.aldareen.notestaking;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

import java.sql.SQLException;

/**
 * Created by Aldareen on 11/7/2015.
 */

public class DatabaseConnector {
```

```
private static final String DB_NAME="MyNotes";
private static final String TABLE_NAME="tablenotes";
private static final String TITLE="title";
private static final String ID="_id";
private static final String NOTE="note";
private static final int DATABASE_VERSION=1;
private SQLiteDatabase database;
private DatabaseHelper dbOpenHelper;
public DatabaseConnector(Context context){
dbOpenHelper=new
DatabaseHelper(context,DB_NAME,null,DATABASE_VERSION);
    }

public void open()throws SQLException{
database=dbOpenHelper.getWritableDatabase();
    }

public void close(){
if(database!=null){
database.close();
        }
    }

public void InsertNote(String title,String note){
        ContentValues newCon=new ContentValues();
        newCon.put(TITLE,title);
        newCon.put(NOTE,note);
try {
            open();
        } catch (SQLException e) {
            e.printStackTrace();
        }
database.insert(TABLE_NAME,null,newCon);
        close();
    }

public void UpdateNote(long id,String title,String note){
        ContentValues editCon=new ContentValues();
        editCon.put(TITLE,title);
        editCon.put(NOTE,note);
try {
            open();
        } catch (SQLException e) {
            e.printStackTrace();
        }
database.update(TABLE_NAME, editCon, ID + "=" + id, null);
        close();
    }

public void DeleteNote(long id){
try {
            open();
        } catch (SQLException e) {
            e.printStackTrace();
        }
database.delete(TABLE_NAME, ID + "=" + id, null);
```

```
        close();
    }

public Cursor ListAllNotes(){
return database.query(TABLE_NAME,new
String[]{ID,TITLE},null,null,null,null,TITLE);
    }

public Cursor GetOneNote(long id){
return database.query(TABLE_NAME,null,ID + "="
+id,null,null,null,null);
    }
}
```

DatabaseHelper.java

```
package com.example.aldareen.notestaking;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

/**
 * Created by Aldareen on 11/7/2015.
 */

public class DatabaseHelperextends SQLiteOpenHelper {
private static final String DB_NAME = "MyNotes";
private static final String TABLE_NAME = "tablenotes";
private static final String TITLE = "title";
private static final String NOTE = "note";
public DatabaseHelper(Context context,String
name,SQLiteDatabase.CursorFactory factory,int version){
super(context,DB_NAME,factory,version);
    }
public void onCreate(SQLiteDatabase db){
        String createQuery="CREATE TABLE " + TABLE_NAME + " (_id
integer primary key autoincrement," + TITLE + ", " + NOTE + ")";
        db.execSQL(createQuery);
    }

public void onUpgrade(SQLiteDatabase db,int oldVersion,int
newVersion){
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME );
        onCreate(db);
    }
}
```

ListText.java

```
package com.example.aldareen.notestaking;

import android.app.ListActivity;
```


```
import android.content.Intent;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.CursorAdapter;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;

import java.sql.SQLException;

public class ListText extends ListActivity {
public static final String ROW_ID="row_id";
private static final String TITLE="title";
private ListView noteListView;
private CursorAdapter noteAdapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
noteListView = getListView();
noteListView.setOnItemClickListener(viewNoteListener);
        String[] from = new String[]{TITLE};
int[] to = new int[]{R.id.ViewTitleNotes};
noteAdapter = new SimpleCursorAdapter(ListText.this,
R.layout.list_note, null, from, to);
        setListAdapter(noteAdapter);
    }
    OnItemClickListener viewNoteListener=new OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
long arg3) {
            Intent viewnote = new Intent(ListText.this,
ViewNote.class);
            viewnote.putExtra(ROW_ID, arg3);
            startActivity(viewnote);
        }
    };

@Override
protected void onResume() {
super.onResume();
new GetNotes().execute((Object[])null);
    }

@Override
protected void onStop() {
        Cursor cursor=noteAdapter.getCursor();
if(cursor!=null)
            cursor.deactivate();
noteAdapter.changeCursor(null);
super.onStop();
    }
```

```
private class GetNotes extends AsyncTask<Object,Object,Cursor>{
        DatabaseConnector dbConnector=new
DatabaseConnector(ListText.this);

@Override
protected Cursor doInBackground(Object... params) {
try {
dbConnector.open();
            } catch (SQLException e) {
                e.printStackTrace();
            }
return dbConnector.ListAllNotes();
        }

@Override
protected void onPostExecute(Cursor result) {
noteAdapter.changeCursor(result);
dbConnector.close();
        }
    }
}
```

ViewNote.java

```
package com.example.aldareen.notestaking;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.widget.TextView;

import java.sql.SQLException;

/**
 * Created by Aldareen on 11/7/2015.
 */

public class ViewNote extends AppCompatActivity {
private long rowID;
private TextView TitleTv;
private TextView NoteTv;
private static final String TITLE="title";
private static final String NOTE="note";

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.view_note);
TitleTv=(TextView)findViewById(R.id.TitleText);
NoteTv=(TextView)findViewById(R.id.NoteText);
        Bundle extras=getIntent().getExtras();
rowID=extras.getLong(ListText.ROW_ID);
    }

@Override
protected void onResume() {
super.onResume();
new LoadNotes().execute(rowID);
    }

private class LoadNotes extends AsyncTask<Long,Object,Cursor>{
        DatabaseConnector dbConnector=new
DatabaseConnector(ViewNote.this);

@Override
protected Cursor doInBackground(Long... params) {
try {
dbConnector.open();
            } catch (SQLException e) {
                e.printStackTrace();
            }
return dbConnector.GetOneNote(params[0]);
        }

@Override
protected void onPostExecute(Cursor result) {
super.onPostExecute(result);
            result.moveToFirst();
int TitleIndex=result.getColumnIndex(TITLE);
int NoteIndex=result.getColumnIndex(NOTE);
TitleTv.setText(result.getString(TitleIndex));
NoteTv.setText(result.getString(NoteIndex));
            result.close();
dbConnector.close();
        }
    }

@Override
public boolean onCreateOptionsMenu(Menu menu) {
        menu.add("Edit
Note").setOnMenuItemClickListener(this.EditButtonClickListener).setS
howAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);
        menu.add("Delete
Notes").setOnMenuItemClickListener(this.DeleteButtonClickListener).s
etShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);
return super.onCreateOptionsMenu(menu);
    }

   OnMenuItemClickListener EditButtonClickListener=new
OnMenuItemClickListener() {
@Override
public boolean onMenuItemClick(MenuItem item) {
```

```
            Intent addeditnotes=new Intent(ViewNote.this,
com.example.aldareen.notestaking.MainFragmentActivity.class);
            addeditnotes.putExtra(ListText.ROW_ID,rowID);
            addeditnotes.putExtra(TITLE,TitleTv.getText());
            addeditnotes.putExtra(NOTE, NoteTv.getText());
            startActivity(addeditnotes);
return false;
        }
    };

    OnMenuItemClickListener DeleteButtonClickListener=new
OnMenuItemClickListener() {
@Override
public boolean onMenuItemClick(MenuItem item) {
            DeleteNote();
return false;
        }
    };

private void DeleteNote(){
        AlertDialog.Builder alert=new
AlertDialog.Builder(ViewNote.this);
        alert.setTitle("Delete Item");
        alert.setMessage("Do you really want to delete this note?");
        alert.setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int button) {
final DatabaseConnector dbConnector = new
DatabaseConnector(ViewNote.this);
                AsyncTask<Long, Object, Object> deleteTask = new
AsyncTask<Long, Object, Object>() {

@Override
protected Object doInBackground(Long... params) {
dbConnector.DeleteNote(params[0]);
return null;
                    }

@Override
protected void onPostExecute(Object result) {
                        finish();
                    }
                };
                deleteTask.execute(new Long[]{rowID});
            }
        });
        alert.setNegativeButton("No",null).show();
    }
}
```

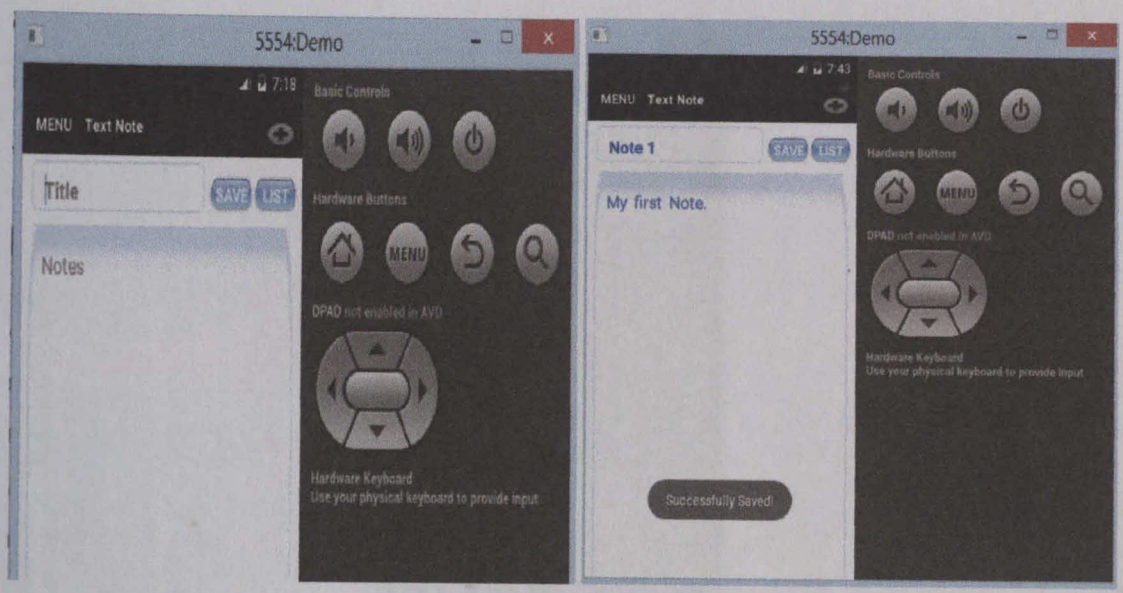
res/layout/add_note.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:layout_weight="1">

<LinearLayout
android:id="@+id/linearLayout"
android:orientation="vertical"
android:layout_width="match_parent"
android:background="@android:color/white"
android:layout_height="fill_parent"
android:padding="5dp">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">

<EditText
android:id="@+id/titleEdit"
android:layout_width="wrap_content"
android:background="@drawable/big_field"
android:layout_height="40dp"
android:layout_weight="1"
android:hint="@string/titleEdit"
android:paddingLeft="15.0dip"
android:paddingTop="7.0dip"
android:layout_marginLeft="5.0dip"
android:layout_marginRight="5.0dip"
android:textStyle="bold"
android:textColor="#1e2eff"/>

<Button
android:layout_width="50dp"
android:layout_height="30dp"
android:background="@drawable/format"
android:text="Save"
android:gravity="center"
android:textColor="@android:color/white"
android:id="@+id/bSave"/>

<Button
android:layout_width="50dp"
android:layout_height="30dp"
android:background="@drawable/format"
android:text="List"
android:gravity="center"
android:textColor="@android:color/white"
android:id="@+id/bList"/>
</LinearLayout>

<EditText
android:id="@+id/noteEdit"
android:layout_width="match_parent"
android:background="@drawable/big_field"
android:layout_height="600dp"
android:layout_weight="1"
```

```
android:layout_marginTop="10dp"
android:paddingLeft="15.0dip"
android:gravity="top"
android:hint="@string/noteEdit"
android:paddingTop="20dp"
android:inputType="textMultiLine"
android:layout_marginLeft="3.0dip"
android:layout_marginRight="5.0dip"
android:textColor="#1e2eff"/>
</LinearLayout>
</ScrollView>
```

OUTPUT :

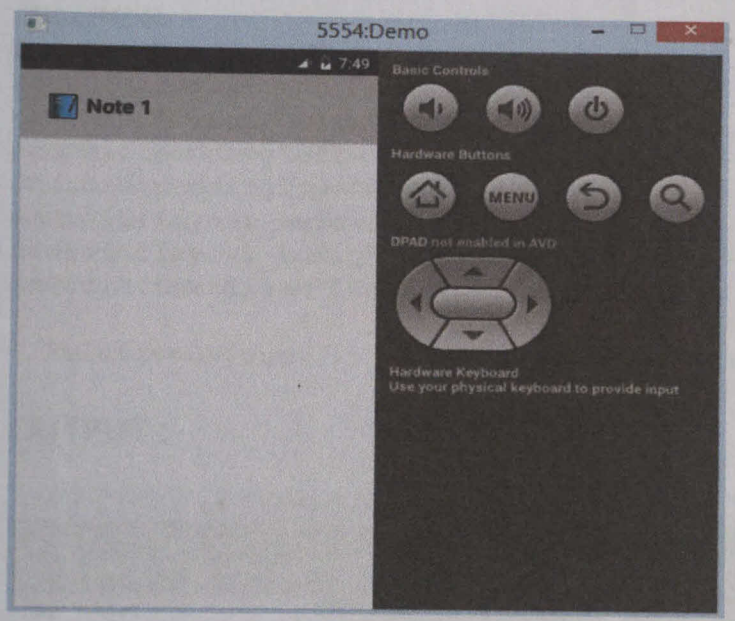


res/layout/list_note.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="horizontal"
android:background="@android:color/darker_gray"
android:padding="20dip">

<ImageView
android:id="@+id/icon"
android:layout_width="30dp"
android:layout_height="25dp"
android:layout_marginLeft="2dp"
android:layout_marginRight="5dp"
android:layout_marginTop="5dp"
android:src="@drawable/note"/>
```

```
<TextView
android:id="@+id/ViewTitleNotes"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textStyle="bold"
android:layout_marginTop="3dp"
android:lines="1"
android:textSize="20dp"
android:textColor="#010609"/>

</LinearLayout>
```

OUTPUT :

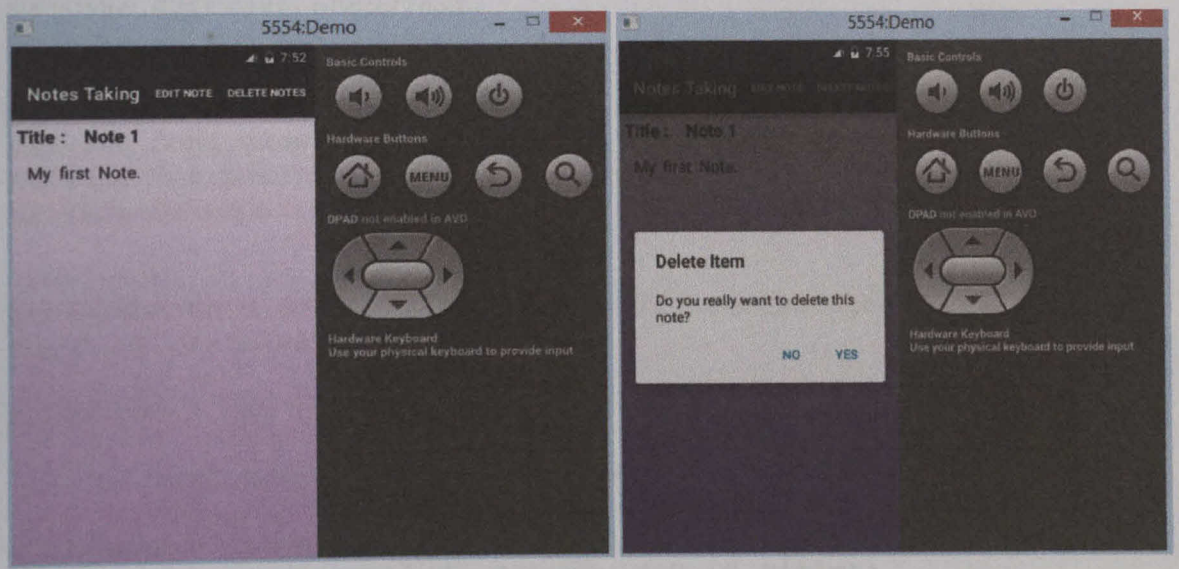


res/layout/view_note.xml

```
<?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:background="@drawable/b"
android:padding="5dp">

<TextView
android:id="@+id/Yourtitle"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/YourTitle"
android:textStyle="bold"
android:textSize="20dp"/>

<TextView
android:id="@+id/TitleText"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_toRightOf="@+id/Yourtitle"
android:paddingLeft="20.0dip"
android:paddingBottom="10dp"
android:textSize="20dp"
android:textStyle="bold"/>


<TextView
android:id="@+id/YourNotes"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/TitleText"
android:text="@string/YourNotes"/>

<TextView
android:id="@+id/NoteText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:paddingLeft="10.0dip"
android:paddingTop="3.0dip"
android:layout_below="@+id/TitleText"
android:layout_toRightOf="@+id/YourNotes"
android:textSize="18dp"/>

</RelativeLayout>
```

OUTPUT :



ReminderEditActivity.java

```
package com.example.aldareen.notestaking;

import android.app.Activity;
```

```
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TimePicker;
import android.widget.Toast;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class ReminderEditActivity extends Activity {

private static final int DATE_PICKER_DIALOG = 0;
private static final int TIME_PICKER_DIALOG = 1;

private static final String DATE_FORMAT = "yyyy-MM-dd";
private static final String TIME_FORMAT = "kk:mm";
public static final String DATE_TIME_FORMAT = "yyyy-MM-dd kk:mm:ss";

private EditText mTitleText;
private EditText mBodyText;
private Button mDateButton;
private Button mTimeButton;
private Button mConfirmButton;
private Long mRowId;
private RemindersDbAdapter mDbHelper;
private Calendar mCalendar;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

mDbHelper = new RemindersDbAdapter(this);

        setContentView(R.layout.reminder_edit);

mCalendar = Calendar.getInstance();
mTitleText = (EditText) findViewById(R.id.title);
mBodyText = (EditText) findViewById(R.id.body);
mDateButton = (Button) findViewById(R.id.reminder_date);
mTimeButton = (Button) findViewById(R.id.reminder_time);

mConfirmButton = (Button) findViewById(R.id.confirm);

mRowId = savedInstanceState != null ?
savedInstanceState.getLong(RemindersDbAdapter.KEY_ROWID)
```

```
                                        : null;

        registerButtonListenersAndSetDefaultText();
    }

private void setRowIdFromIntent() {
if (mRowId == null) {
        Bundle extras = getIntent().getExtras();
mRowId = extras != null ?
extras.getLong(RemindersDbAdapter.KEY_ROWID)
                            : null;
      }
   }

@Override
protected void onPause() {
super.onPause();
mDbHelper.close();
    }

@Override
protected void onResume() {
super.onResume();
mDbHelper.open();
       setRowIdFromIntent();
      populateFields();
    }

@Override
protected Dialog onCreateDialog(int id) {
switch(id) {
case DATE_PICKER_DIALOG:
return showDatePicker();
case TIME_PICKER_DIALOG:
return showTimePicker();
       }
return super.onCreateDialog(id);
    }

private DatePickerDialog showDatePicker() {
      DatePickerDialog datePicker = new
DatePickerDialog(ReminderEditActivity.this, new
DatePickerDialog.OnDateSetListener() {

@Override
public void onDateSet(DatePicker view, int year, int monthOfYear,
int dayOfMonth) {
mCalendar.set(Calendar.YEAR, year);
mCalendar.set(Calendar.MONTH, monthOfYear);
mCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
           updateDateButtonText();
        }
      }, mCalendar.get(Calendar.YEAR),
mCalendar.get(Calendar.MONTH),
mCalendar.get(Calendar.DAY_OF_MONTH));
return datePicker;
```

```
    }

private TimePickerDialog showTimePicker() {
       TimePickerDialog timePicker = new TimePickerDialog(this, new
TimePickerDialog.OnTimeSetListener() {

@Override
public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
mCalendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
mCalendar.set(Calendar.MINUTE, minute);
            updateTimeButtonText();
         }
      }, mCalendar.get(Calendar.HOUR_OF_DAY),
mCalendar.get(Calendar.MINUTE), true);

return timePicker;
   }
private void registerButtonListenersAndSetDefaultText() {

mDateButton.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View v) {
            showDialog(DATE_PICKER_DIALOG);
         }
      });

mTimeButton.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View v) {
            showDialog(TIME_PICKER_DIALOG);
         }
      });

mConfirmButton.setOnClickListener(new View.OnClickListener() {
public void onClick(View view) {
              saveState();
              setResult(RESULT_OK);
               Toast.makeText(ReminderEditActivity.this,
getString(R.string.task_saved_message), Toast.LENGTH_SHORT).show();
               finish();
           }

        });

        updateDateButtonText();
         updateTimeButtonText();
   }

private void populateFields()  {
if (mRowId != null) {
            Cursor reminder = mDbHelper.fetchReminder(mRowId);
            startManagingCursor(reminder);
mTitleText.setText(reminder.getString(
```

```
reminder.getColumnIndexOrThrow(RemindersDbAdapter.KEY_TITLE)));
mBodyText.setText(reminder.getString(

reminder.getColumnIndexOrThrow(RemindersDbAdapter.KEY_BODY)));


            SimpleDateFormat dateTimeFormat = new
SimpleDateFormat(DATE_TIME_FORMAT);
            Date date = null;
try {
            String dateString =
reminder.getString(reminder.getColumnIndexOrThrow(RemindersDbAdapter
.KEY_DATE_TIME));
            date = dateTimeFormat.parse(dateString);
mCalendar.setTime(date);
         } catch (ParseException e) {
            Log.e("ReminderEditActivity", e.getMessage(), e);
         }
        } else {
           SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(this);
           String defaultTitleKey =
getString(R.string.pref_task_title_key);
           String defaultTimeKey =
getString(R.string.pref_default_time_from_now_key);

           String defaultTitle = prefs.getString(defaultTitleKey,
null);
           String defaultTime = prefs.getString(defaultTimeKey,
null);

if(defaultTitle != null)
mTitleText.setText(defaultTitle);

if(defaultTime != null)
mCalendar.add(Calendar.MINUTE, Integer.parseInt(defaultTime));
        }
        updateDateButtonText();
        updateTimeButtonText();
    }

private void updateTimeButtonText() {
        SimpleDateFormat timeFormat = new
SimpleDateFormat(TIME_FORMAT);
        String timeForButton =
timeFormat.format(mCalendar.getTime());
mTimeButton.setText(timeForButton);
   }

private void updateDateButtonText() {
        SimpleDateFormat dateFormat = new
SimpleDateFormat(DATE_FORMAT);
        String dateForButton =
dateFormat.format(mCalendar.getTime());
mDateButton.setText(dateForButton);
   }
```

```
@Override
protected void onSaveInstanceState(Bundle outState) {
super.onSaveInstanceState(outState);
//    outState.putLong(RemindersDbAdapter.KEY_ROWID, mRowId);
}

private void saveState() {
        String title = mTitleText.getText().toString();
        String body = mBodyText.getText().toString();

        SimpleDateFormat dateTimeFormat = new
SimpleDateFormat(DATE_TIME_FORMAT);
       String reminderDateTime =
dateTimeFormat.format(mCalendar.getTime());

if (mRowId == null) {

long id = mDbHelper.createReminder(title, body, reminderDateTime);
if (id >0) {
mRowId = id;
            }
        } else {
mDbHelper.updateReminder(mRowId, title, body, reminderDateTime);
        }
new ReminderManager(this).setReminder(mRowId, mCalendar);
    }
}
```

OnAlarmReceiver.java

```
package com.example.aldareen.notestaking;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.pm.ComponentInfo;
import android.util.Log;

public class OnAlarmReceiver extends BroadcastReceiver {

private static final String TAG =
ComponentInfo.class.getCanonicalName();

@Override
public void onReceive(Context context, Intent intent) {
      Log.d(TAG, "Received wake up from alarm manager.");

long rowid =
intent.getExtras().getLong(RemindersDbAdapter.KEY_ROWID);

      WakeReminderIntentService.acquireStaticLock(context);

      Intent i = new Intent(context, ReminderService.class);
```

```
        i.putExtra(RemindersDbAdapter.KEY_ROWID, rowid);
        context.startService(i);
    }
}
```

OnBootReceiver.java

```
package com.example.aldareen.notestaking;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.pm.ComponentInfo;
import android.database.Cursor;
import android.util.Log;

import java.text.SimpleDateFormat;
import java.util.Calendar;

public class OnBootReceiver extends BroadcastReceiver {
private static final String TAG =
ComponentInfo.class.getCanonicalName();

@Override
public void onReceive(Context context, Intent intent) {

        ReminderManager reminderMgr = new ReminderManager(context);
        RemindersDbAdapter dbHelper = new RemindersDbAdapter(context);
dbHelper.open();
        Cursor cursor = dbHelper.fetchAllReminders();

if(cursor != null) {
          cursor.moveToFirst();
int rowIdColumnIndex =
cursor.getColumnIndex(RemindersDbAdapter.KEY_ROWID);
int dateTimeColumnIndex =
cursor.getColumnIndex(RemindersDbAdapter.KEY_DATE_TIME);
while(cursor.isAfterLast() == false) {
            Log.d(TAG, "Adding alarm from boot.");
            Log.d(TAG, "Row Id Column Index - " + rowIdColumnIndex);
            Log.d(TAG, "Date Time Column Index - " +
dateTimeColumnIndex);

            Long rowId = cursor.getLong(rowIdColumnIndex);
            String dateTime = cursor.getString(dateTimeColumnIndex);

            Calendar cal = Calendar.getInstance();
            SimpleDateFormat format = new
SimpleDateFormat(ReminderEditActivity.DATE_TIME_FORMAT);
try {
               java.util.Date date = format.parse(dateTime);
               cal.setTime(date);

               reminderMgr.setReminder(rowId, cal);
            } catch (java.text.ParseException e) {
```

```
                Log.e("OnBootReceiver", e.getMessage(), e);
            }
            cursor.moveToNext();
        }
        cursor.close() ;
    }
dbHelper.close();
    }
}
```

ReminderManager.java

```
package com.example.aldareen.notestaking;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;

import java.util.Calendar;

public class ReminderManager {
private Context mContext;
private AlarmManager mAlarmManager;

public ReminderManager(Context context) {
mContext = context;
mAlarmManager =
(AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
    }

public void setReminder(Long taskId, Calendar when) {

        Intent i = new Intent(mContext, OnAlarmReceiver.class);
        i.putExtra(RemindersDbAdapter.KEY_ROWID, (long)taskId);
        PendingIntent pi = PendingIntent.getBroadcast(mContext, 0,
i, PendingIntent.FLAG_ONE_SHOT);
mAlarmManager.set(AlarmManager.RTC_WAKEUP, when.getTimeInMillis(),
pi);
    }
}
```

RemindersDbAdapter.java

```
package com.example.aldareen.notestaking;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
```

```
public class RemindersDbAdapter {

private static final StringDATABASE_NAME = "data";
private static final StringDATABASE_TABLE = "reminders";
private static final int DATABASE_VERSION = 3;

public static final StringKEY_TITLE = "title";
public static final StringKEY_BODY = "body";
public static final StringKEY_DATE_TIME = "reminder_date_time";
public static final StringKEY_ROWID = "_id";

private static final StringTAG = "ReminderDbAdapter";
private DatabaseHelper mDbHelper;
private SQLiteDatabase mDb;

private static final StringDATABASE_CREATE =
"create table " + DATABASE_TABLE + " ("
+ KEY_ROWID + " integer primary key autoincrement, "
+ KEY_TITLE + " text not null, "
+ KEY_BODY + " text not null, "
+ KEY_DATE_TIME + " text not null);";

private final Context mCtx;

private static class DatabaseHelper extends SQLiteOpenHelper {
        DatabaseHelper(Context context) {
super(context, DATABASE_NAME, null, DATABASE_VERSION);
        }

@Override
public void onCreate(SQLiteDatabase db) {
            db.execSQL(DATABASE_CREATE);
        }

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
            Log.w(TAG, "Upgrading database from version " +
oldVersion + " to "
+ newVersion + ", which will destroy all old data");
            db.execSQL("DROP TABLE IF EXISTS " + DATABASE_TABLE);
            onCreate(db);
        }
    }

public RemindersDbAdapter(Context ctx) {
this.mCtx = ctx;
    }

public RemindersDbAdapter open() throws SQLException {
mDbHelper = new DatabaseHelper(mCtx);
mDb = mDbHelper.getWritableDatabase();
return this;
    }
```

```
public void close() {
mDbHelper.close();
    }

public long createReminder(String title, String body, String
reminderDateTime) {
        ContentValues initialValues = new ContentValues();
        initialValues.put(KEY_TITLE, title);
        initialValues.put(KEY_BODY, body);
        initialValues.put(KEY_DATE_TIME, reminderDateTime);

return mDb.insert(DATABASE_TABLE, null, initialValues);
    }


public boolean deleteReminder(long rowId) {

return mDb.delete(DATABASE_TABLE, KEY_ROWID + "=" + rowId, null) >0;
    }

public Cursor fetchAllReminders() {

return mDb.query(DATABASE_TABLE, new String[] {KEY_ROWID, KEY_TITLE,
KEY_BODY, KEY_DATE_TIME}, null, null, null, null, null);
    }

public Cursor fetchReminder(long rowId) throws SQLException {

        Cursor mCursor =

mDb.query(true, DATABASE_TABLE, new String[] {KEY_ROWID,
KEY_TITLE, KEY_BODY, KEY_DATE_TIME}, KEY_ROWID + "=" + rowId, null,
null, null, null, null);
if (mCursor != null) {
            mCursor.moveToFirst();
        }
return mCursor;

    }

public boolean updateReminder(long rowId, String title, String body,
String reminderDateTime) {
        ContentValues args = new ContentValues();
        args.put(KEY_TITLE, title);
        args.put(KEY_BODY, body);
        args.put(KEY_DATE_TIME, reminderDateTime);

return mDb.update(DATABASE_TABLE, args, KEY_ROWID + "=" + rowId,
null) >0;
    }
}
```

ReminderService.java

```
package com.example.aldareen.notestaking;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.util.Log;

public class ReminderService extends WakeReminderIntentService {

public ReminderService() {
super("ReminderService");
        }

@Override
void doReminderWork(Intent intent) {
      Log.d("ReminderService", "Doing work.");
      Long rowId =
intent.getExtras().getLong(RemindersDbAdapter.KEY_ROWID);

      NotificationManager mgr =
(NotificationManager)getSystemService(NOTIFICATION_SERVICE);

Intent notificationIntent = new Intent(this,
ReminderEditActivity.class);
      notificationIntent.putExtra(RemindersDbAdapter.KEY_ROWID,
rowId);

      PendingIntent pi = PendingIntent.getActivity(this, 0,
notificationIntent, PendingIntent.FLAG_ONE_SHOT);

      Notification note=new
Notification(android.R.drawable.stat_sys_warning,
getString(R.string.notify_new_task_message),
System.currentTimeMillis());
      note.setLatestEventInfo(this,
getString(R.string.notify_new_task_title),
getString(R.string.notify_new_task_message), pi);
      note.defaults |= Notification.DEFAULT_SOUND;
      note.flags |= Notification.FLAG_AUTO_CANCEL;

int id = (int)((long)rowId);
      mgr.notify(id, note);
   }
}
```

WakeReminderIntentService.java

```
package com.example.aldareen.notestaking;

import android.app.IntentService;
import android.content.Context;
```

```
import android.content.Intent;
import android.os.PowerManager;

public abstract class WakeReminderIntentService extends
IntentService {
abstract void doReminderWork(Intent intent);

public static final String
LOCK_NAME_STATIC="com.example.aldareen.task.Static";
private static PowerManager.WakeLock lockStatic=null;

public static void acquireStaticLock(Context context) {
getLock(context).acquire();
   }

synchronized private static PowerManager.WakeLock getLock(Context
context) {
if (lockStatic==null) {
        PowerManager
mgr=(PowerManager)context.getSystemService(Context.POWER_SERVICE);
lockStatic=mgr.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK,
LOCK_NAME_STATIC);
lockStatic.setReferenceCounted(true);
      }
return(lockStatic);
   }

public WakeReminderIntentService(String name) {
super(name);
   }

@Override
final protected void onHandleIntent(Intent intent) {
try {
         doReminderWork(intent);
      }
finally {
getLock(this).release();
      }
   }
}
```

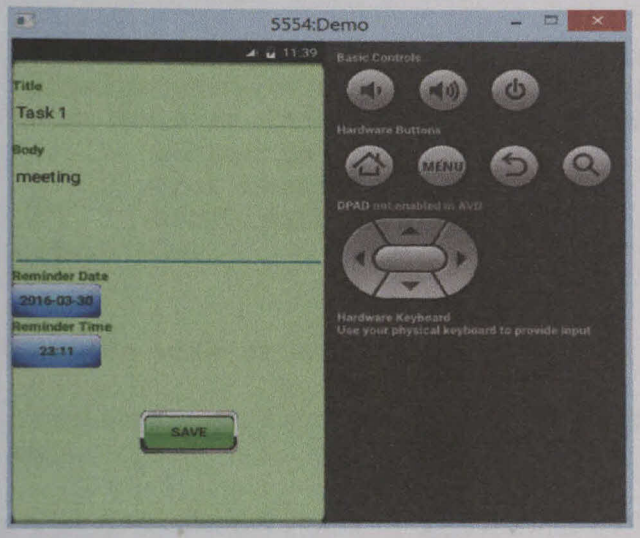
res/layout/reminder_edit.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:background="@drawable/b4"
android:layout_height="fill_parent">

<LinearLayout
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
```

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:paddingTop="20dp"
android:text="@string/title"
android:textStyle="bold"
android:textColor="#1e600b" />

<EditText
android:id="@+id/title"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
/>

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:paddingTop="10dp"
android:text="@string/body"
android:textStyle="bold"
android:textColor="#1e600b" />

<EditText
android:id="@+id/body"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:minLines="5"
android:scrollbars="vertical"
android:gravity="top" />

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/date"
android:textStyle="bold"
android:textColor="#1e600b" />

<Button
android:id="@+id/reminder_date"
android:layout_height="40dp"
android:layout_width="90dp"
android:background="@drawable/remindr"
/>

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/time"
android:textStyle="bold"
android:textColor="#1e600b" />

<Button
android:id="@+id/reminder_time"
android:layout_height="40dp"
android:layout_width="90dp"
```

```
android:background="@drawable/remindr"
/>

<Button
android:id="@+id/confirm"
android:text="@string/confirm"
android:layout_width="100dp"
android:layout_height="50dp"
android:layout_marginTop="50dp"
android:layout_marginLeft="130dp"
android:background="@drawable/s"/>
</LinearLayout>
</ScrollView>
```

OUTPUT :



Audio.java

```
package com.example.aldareen.notestaking;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.os.Environment;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

import java.io.File;
import java.io.IOException;
```

```
/**
 * Created by Aldareen on 3/29/2016.
 */

public class Audio extends Fragment {

private static final String AUDIO_RECORDER_FILE_EXT_AMR_NB=".amr";
private static final String AUDIO_RECORDER_FILE_EXT_MP3=".mp3";
private static final String AUDIO_RECORDER_FOLDER="AudioRecorder";
private MediaRecorder recorder=null;
private int currentFormat=0;

private int output_formats[]={MediaRecorder.OutputFormat.AMR_NB,
            MediaRecorder.OutputFormat.DEFAULT};

private String file_exts[]={AUDIO_RECORDER_FILE_EXT_MP3,
AUDIO_RECORDER_FILE_EXT_AMR_NB};
    View view;

public View onCreateView(LayoutInflater inflater,ViewGroup
container,Bundle savedInstanceState){
view=inflater.inflate(R.layout.fragment_audio,null);
        setButtonHandlers();
        enableButtons(false);
        setFormatButtonCaption();
return view;
    }

private void setButtonHandlers(){

((Button)view.findViewById(R.id.btnStart)).setOnClickListener(btnCli
ck);

((Button)view.findViewById(R.id.btnStop)).setOnClickListener(btnClic
k);

((Button)view.findViewById(R.id.btnFormat)).setOnClickListener(btnCl
ick);
    }

private void enableButton(int id,boolean isEnable){
        ((Button)view.findViewById(id)).setEnabled(isEnable);
    }

private void enableButtons(boolean isRecording){
        enableButton(R.id.btnStart,!isRecording);
        enableButton(R.id.btnFormat,!isRecording);
        enableButton(R.id.btnStop, isRecording);
    }

private void setFormatButtonCaption(){
        ((Button)view.findViewById(R.id.btnFormat)).setText
                (getString(R.string.audio_format) + "(" +
file_exts[currentFormat] + ")");
    }
```


```
private String getFilename(){
        String filepath=
Environment.getExternalStorageDirectory().getPath();
        File file=new File(filepath,AUDIO_RECORDER_FOLDER);
if(!file.exists()){
            file.mkdirs();
        }
return (file.getAbsolutePath() +"/" +System.currentTimeMillis()
+file_exts[currentFormat]);
    }

private void startRecording(){
recorder=new MediaRecorder();
recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder.setOutputFormat(output_formats[currentFormat]);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
recorder.setOutputFile(getFilename());
recorder.setOnErrorListener(errorListener);
recorder.setOnInfoListener(infoListener);
try{
recorder.prepare();
recorder.start();
        }catch (IllegalStateException e){
            e.printStackTrace();
        }catch (IOException e){
            e.printStackTrace();
        }
    }

private void stopRecording(){
if(null!=recorder){
recorder.stop();
recorder.reset();
recorder.release();
recorder=null;
        }
    }

private void displayFormatDialog(){
        AlertDialog.Builder builder=new
AlertDialog.Builder(getView().getContext());
        String formats[]={"MP3","AMR_NB"};

builder.setTitle(getString(R.string.choose_format_title)).setSingleC
hoiceItems(formats, currentFormat, new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
currentFormat=which;
                setFormatButtonCaption();
                dialog.dismiss();
            }
        }).show();
    }

private MediaRecorder.OnErrorListener errorListener=new
```

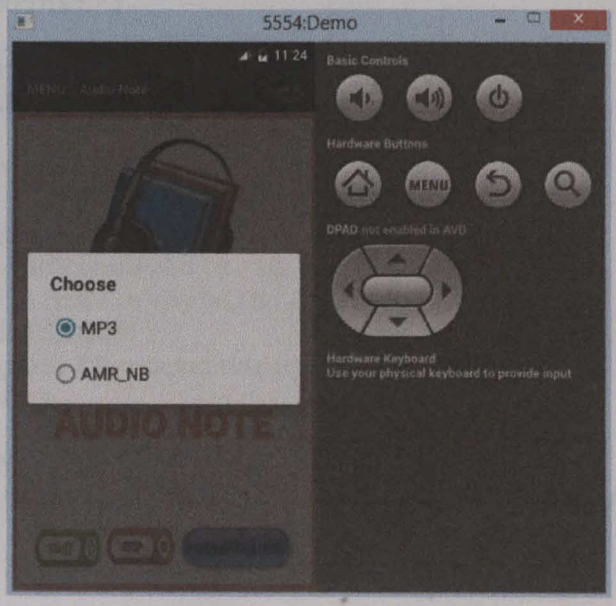
```
MediaRecorder.OnErrorListener(){
public void onError(MediaRecorder mr,int what, int extra){
            Toast.makeText(getView().getContext(), "Error : "
+ what + ", " + extra, Toast.LENGTH_SHORT).show();
        }
    };

private MediaRecorder.OnInfoListener infoListener=new
MediaRecorder.OnInfoListener(){
public void onInfo(MediaRecorder mr,int what, int extra){
            Toast.makeText(getView().getContext(), "Warning : "
+ what +", "+extra, Toast.LENGTH_SHORT).show();
        }
    };

private View.OnClickListener btnClick=new View.OnClickListener(){
public void onClick(View v){
switch (v.getId()){
case R.id.btnStart:{
                    Toast.makeText(getView().getContext(), "Start
Recording", Toast.LENGTH_SHORT).show();
                    enableButtons(true);
                    startRecording();
break;
                }
case R.id.btnStop:{
                    Toast.makeText(getView().getContext(), "Stop
Recording", Toast.LENGTH_SHORT).show();
                    enableButtons(false);
                    stopRecording();
break;
                }
case R.id.btnFormat:{
                    displayFormatDialog();
break;
                }
            }
        }
    };
}
```

res/layout/fragment_audio.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:padding="20dip"
android:background="@drawable/bac">

<ImageView
android:layout_width="200dp"
android:layout_height="200dp"
```

```
android:src="@drawable/au"
android:layout_gravity="center"
android:scaleType="fitCenter" />

<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/app_info"
android:layout_weight="1.0"
android:textSize="40dip"
android:textStyle="bold"
android:gravity="center"
android:textColor="#ff1707"/>

<LinearLayout
android:orientation="horizontal"
android:layout_width="fill_parent"
android:layout_height="wrap_content">

<Button
android:layout_width="100dp"
android:layout_height="50dp"
android:id="@+id/btnStart"
android:background="@drawable/start"
android:layout_weight="1.0" />

<Button
android:layout_width="100dp"
android:layout_height="50dp"
android:id="@+id/btnStop"
android:background="@drawable/stop"
android:layout_weight="1.0" />

<Button
android:layout_width="150dp"
android:layout_height="50dp"
android:id="@+id/btnFormat"
android:background="@drawable/format"
android:text="Format\n(mp3)"
android:layout_weight="1.0"/>
</LinearLayout>
</LinearLayout>
```

OUTPUT :



SpeechToText.java

```
package com.example.aldareen.notestaking;

import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognitionListener;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.CompoundButton;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.ToggleButton;

import java.util.ArrayList;

/**
 * Created by Aldareen on 3/16/2016.
 */

public class SpeechToText extends Fragment implements
RecognitionListener {


private TextView returnedText;
private ToggleButton toggleButton;
private ProgressBar progressBar;
```

```
private SpeechRecognizer speech = null;
private Intent recognizerIntent;
private String LOG_TAG = "SpeechToText";

public View onCreateView(LayoutInflater inflater,ViewGroup
container,Bundle savedInstanceState){
        View
view=inflater.inflate(R.layout.fragment_speechtotext,null);

returnedText = (TextView)view.findViewById(R.id.textView1);
progressBar = (ProgressBar)view.findViewById(R.id.progressBar1);
toggleButton = (ToggleButton)view.findViewById(R.id.toggleButton1);
progressBar.setVisibility(View.INVISIBLE);
speech = SpeechRecognizer.createSpeechRecognizer(getActivity());
speech.setRecognitionListener(this);
recognizerIntent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
recognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_PREFERENCE
, "en");
recognizerIntent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE,
getActivity().getPackageName());
recognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
recognizerIntent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 3);
toggleButton.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
@Override
public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
if (isChecked) {
progressBar.setVisibility(View.VISIBLE);
progressBar.setIndeterminate(true);
speech.startListening(recognizerIntent);
                } else {
progressBar.setIndeterminate(false);
progressBar.setVisibility(View.VISIBLE);
speech.stopListening();
                }
            }
        });

return view;
    }

@Override
public void onResume() {
super.onResume();
    }

@Override
public void onPause() {
super.onPause();
if (speech != null) {
//speech.destroy();
Log.i(LOG_TAG, "destroy");
        }
```

```
    }

@Override
public void onBeginningOfSpeech() {
        Log.i(LOG_TAG, "onBeginningOfSpeech");
progressBar.setIndeterminate(false);
progressBar.setMax(10);
    }

@Override
public void onBufferReceived(byte[] buffer) {
        Log.i(LOG_TAG, "onBufferReceived " + buffer);
    }

@Override
public void onEndOfSpeech() {
        Log.i(LOG_TAG, "onEndOfSpeech");
progressBar.setIndeterminate(true);
toggleButton.setChecked(false);
    }

@Override
public void onError(int error) {
        String errorMessage = getErrorText(error);
        Log.d(LOG_TAG, "FAILED " + errorMessage);
returnedText.setText(errorMessage);
toggleButton.setChecked(false);
    }

public void onEvent(int arg0, Bundle arg1) {
        Log.i(LOG_TAG, "onEvent");
    }

public void onPartialResults(Bundle arg0) {
        Log.i(LOG_TAG, "onPartialResults");
    }

public void onReadyForSpeech(Bundle arg0) {
        Log.i(LOG_TAG, "onReadyForSpeech");
    }

public void onResults(Bundle results) {
        Log.i(LOG_TAG, "onResults");
        ArrayList<String> matches =
results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
        String text = "";
for (String result : matches)
            text += result + "\n";
returnedText.setText(text);
    }

@Override
public void onRmsChanged(float rmsdB) {
        Log.i(LOG_TAG, "onRmsChanged : " + rmsdB);
progressBar.setProgress((int) rmsdB);
    }
```

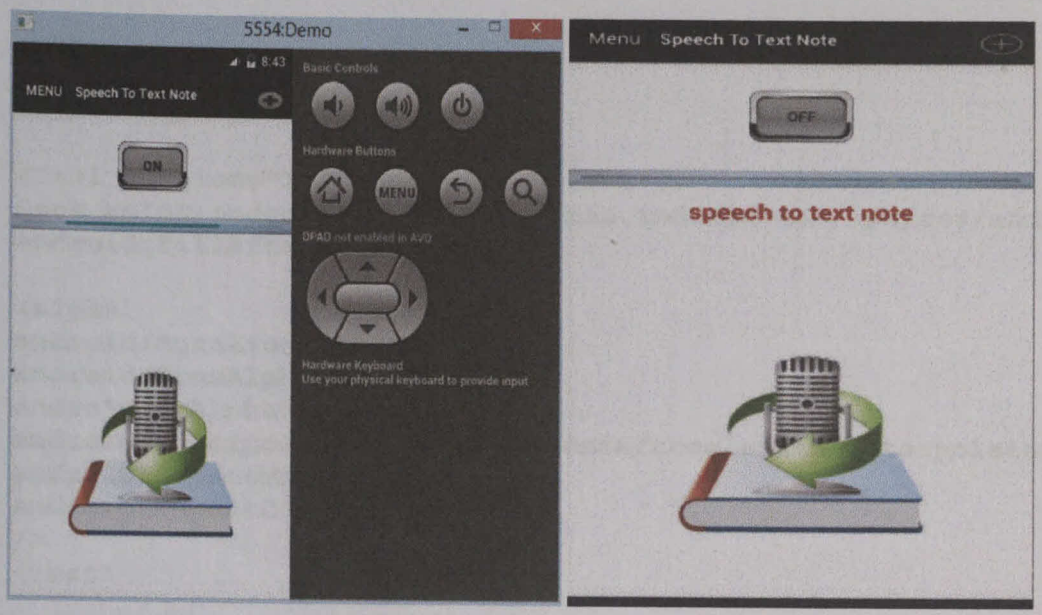
```
public static String getErrorText(int errorCode) {
        String message;
switch (errorCode) {
case SpeechRecognizer.ERROR_AUDIO:
                message = "Audio recording Error";
break;
case SpeechRecognizer.ERROR_CLIENT:
                message = "Client Side Error";
break;
case SpeechRecognizer.ERROR_INSUFFICIENT_PERMISSIONS:
                message = "Insufficient Permissions";
break;
case SpeechRecognizer.ERROR_NETWORK:
                message = "Network Error";
break;
case SpeechRecognizer.ERROR_NETWORK_TIMEOUT:
                message = "Network Timeout";
break;
case SpeechRecognizer.ERROR_NO_MATCH:
                message = "No match";
break;
case SpeechRecognizer.ERROR_RECOGNIZER_BUSY:
                message = "Recognition Service Busy";
break;
case SpeechRecognizer.ERROR_SERVER:
                message = "Error From Sever";
break;
case SpeechRecognizer.ERROR_SPEECH_TIMEOUT:
                message = "No Speech Input";
break;
default:
                message = "Didn't Understand please try again";
break;
        }
return message;
    }
}
```

res/layout/fragment_speechtotext.xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:background="@drawable/bc1">

<ToggleButton
android:id="@+id/toggleButton1"
android:layout_width="80dp"
android:layout_height="50dp"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
```

```
android:layout_marginTop="26dp"
android:background="@drawable/boff"
android:text="ToggleButton"
android:checked="false" />

<ProgressBar
android:id="@+id/progressBar1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
style="?android:attr/progressBarStyleHorizontal"
android:layout_alignParentLeft="true"
android:layout_below="@+id/toggleButton1"
android:layout_marginTop="28dp"
android:paddingLeft="10dp"
android:paddingRight="10dp"
android:background="@drawable/prog"/>

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/progressBar1"
android:textSize="20dp"
android:textStyle="bold"
android:textColor="@android:color/holo_red_dark"
android:layout_centerHorizontal="true"
android:layout_marginTop="10dp"/>

<ImageView
android:id="@+id/imageView1"
android:layout_width="200dp"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true"
android:src="@drawable/stt"/>
</RelativeLayout>
```

OUTPUT :



About.java

```
package com.example.aldareen.notestaking;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;

/**
 * Created by Aldareen on 2/29/2016.
 */
public class Aboutextends Fragment {

    Animation anim;
    ImageView img;

public View onCreateView(LayoutInflater inflater,ViewGroup
container,Bundle savedInstanceState){
        View view=inflater.inflate(R.layout.fragment_about,null);
img=(ImageView)view.findViewById(R.id.ivImg);
anim=
AnimationUtils.loadAnimation(getActivity().getApplicationContext(),
R.anim.blink);
img.setVisibility(View.VISIBLE);
img.startAnimation(anim);
return view;
```

```
    }
}
```

res/anim/blink.xml

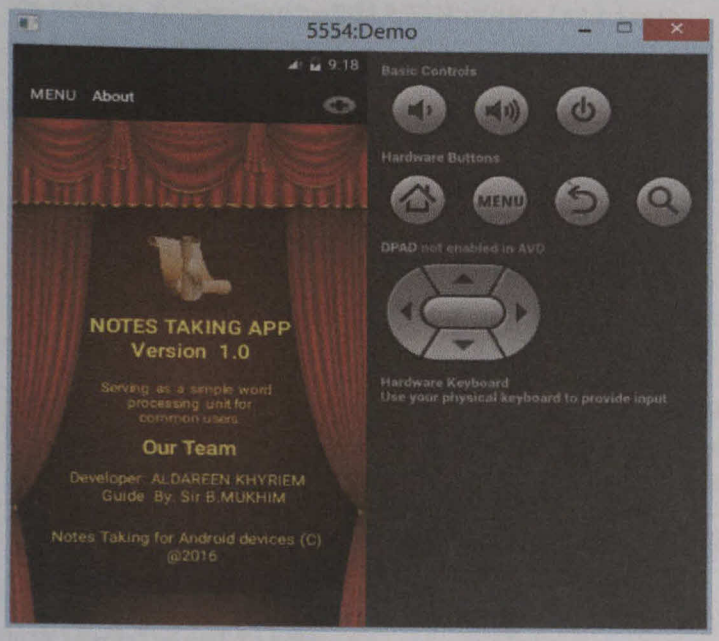
```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:fillAfter="true">

<alpha
android:duration="600"
android:fromAlpha="0.0"
android:toAlpha="1.0"
android:interpolator="@android:anim/accelerate_interpolator"
android:repeatMode="reverse"
android:repeatCount="infinite"
/>
</set>
```

res/layout/fragment_about.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:isScrollContainer="true"
android:background="@drawable/img15"
xmlns:android="http://schemas.android.com/apk/res/android">

<LinearLayout
android:orientation="vertical"
android:layout_width="fill_parent"
android:padding="20dip"
android:layout_height="fill_parent"
>
<ImageView
android:id="@+id/ivImg"
android:layout_width="80dp"
android:layout_height="80dp"
android:src="@drawable/img19"
android:layout_gravity="center"
android:layout_marginTop="90.0dp" />

<TextView
android:textSize="20.0sp"
android:textStyle="bold"
android:gravity="center"
android:layout_gravity="center"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginTop="0.0dip"
android:layout_marginBottom="8.0dip"
```

```
android:text="NOTES TAKING APP\nVersion  1.0"
android:textColor="#f6ff74"
/>

<ImageView
android:layout_width="fill_parent"
android:layout_height="wrap_content"  />


<TextView android:gravity="center"
android:textSize="13.0sp"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginTop="6.0dip"
android:text="Serving  as  a  simple  word  \n processing  unit for
\n common users."
android:textColor="#f6ff74" />

<TextView android:textSize="20.0sp"
android:textStyle="bold"
android:gravity="center"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginTop="6.0dip"
android:text="Our Team"
android:textColor="#f6ff74" />

<TextView android:gravity="center"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginTop="6.0dip"
android:text="Developer: ALDAREEN KHYRIEM \n  Guide  By: Sir
B.MUKHIM"
android:textColor="#f6ff74"
/>

<TextView
android:gravity="center"
android:id="@+id/textView6"
android:paddingBottom="3.0dip"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginTop="4.0dip"
android:text="\nNotes Taking for Android devices (C) \n @2016"
android:textColor="#f6ff74" />
</LinearLayout>
</ScrollView>
```

OUTPUT :



res/values/menu_items.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string-array name="menu_items">
<item >Text Note</item>
<item >Audio Note</item>
<item >Speech To Text Note</item>
<item >About</item>
<item >Exit</item>
</string-array>
</resources>
```

res/values/strings.xml

```
<resources>
<string name="title_activity_splash">Notes Taking</string>
<string name="app_name">Notes Taking</string>

<string name="app_info">AUDIO NOTE</string>
<string name="start_recording">START</string>
<string name="stop_recording">STOP</string>
<string name="audio_format">FORMAT</string>
<string name="choose_format_title">Choose</string>

<string name="text_label_1">Rate</string>
<string name="button_label">Submit</string>
<string name="text_label_2">Result :</string>
```

```
<string name="titleEdit">Title</string>
<string name="noteEdit">Notes</string>
<string name="YourTitle">Title :</string>
<string name="YourNotes"></string>

<string name="edit_reminder_title">Task Reminder - Edit</string>
<string name="no_reminders">No Reminders Yet</string>
<string name="menu_insert">Add Reminder</string>
<string name="menu_delete">Delete Reminder</string>
<string name="menu_settings">Settings</string>
<string name="title">Title</string>
<string name="body">Body</string>
<string name="confirm">Save</string>
<string name="edit_reminder">Edit Reminder</string>
<string name="date">Reminder Date</string>
<string name="time">Reminder Time</string>

<string name="task_saved_message">Task has been saved</string>

<string name="notify_new_task_message">A task needs to be
reviewed!</string>
<string name="notify_new_task_title">Task Reminder!</string>

<string
name="pref_category_task_defaults_key">task_default_category</string
>
<string name="pref_category_task_defaults_title">Task Title
Default</string>
<string name="pref_task_title_key">default_reminder_title</string>
<string name="pref_task_title_dialog_title">Default Reminder
Title</string>
<string name="pref_task_title_message">The default title for a
reminder.</string>
<string name="pref_task_title_summary">Default title for
reminders.</string>
<string name="pref_task_title_title">Default Reminder Title</string>

<string
name="pref_category_datetime_key">date_time_default_category</string
>
<string name="pref_category_datetime_title">Date Time
Defaults</string>
<string
name="pref_default_time_from_now_key">time_from_now_default</string>
<string name="pref_default_time_from_now_dialog_title">Time From
Now</string>
<string name="pref_default_time_from_now_message">The default time
from now that a new reminder should be set to.</string>

</resources>
```

CONCLUSION

Before this project, I had no idea about how the android apps are being made. But after completion of the project I have gathered a huge knowledge of android and how to develop its app.

The main aim of the project is to provide an easy to use application for taking notes by typing, recording and voice note. For voice note the developer can also build using API with their own database to improve the quality context and to make this app a little bit faster.

There are many improvement that are to be done in this application but as of now whatever I have come up is performing as per standards and is upto the mark. The application has been implemented and tested on real device.

Hope this project would be a helpful one to the user.

BIBLIOGRAPHY

- Android Studio Development Essentials,
  Neil Smyth

- Professional Android Application Development,
  Reto Meier

- Internet,
  - Android: Start audio recording with intent of.htm
  - How to create an SpeechToText App for Android
  - Android SQLite database Tutorial.htm
  - Android : Simple Sliding Menu example.htm
  - Tutorialsbuzz.com
  - How to Store file to external storage directory in android.htm