**CHAPTER-1**

**INTRODUCTION**

**1.1 Problem statement**

In daily life, proper management of time is required to do the tasks efficiently. An average businessman needs to remember all his appointments and it becomes a tedious task to manage all the meetings. In the current scenario, the host of the meeting invites the prospective invitees by either calling them on phone, or inviting them in person. It becomes more difficult if the meeting is cancelled and it is often quite late until this news reaches the person attending the meeting. This may result in wasting a lot of time and efforts of the persons involved in the meeting. Therefore, handling of the complications involved in this process makes it difficult to manage time effectively.

**1.2 Proposed System**

This is an android application which helps the user to decrease the time involved in planning and executing the meeting in a more interactive way. It provides features like creation of meeting in the phone and inviting the people from it. Every meeting will have a unique ID number and the person can confirm his/her absence. These features will help in methodical use of time and resources and decrease the burden from the user. This application comes with additional features to help the tasks of the user like Currency converter is used to convert which is used in determining the converted currency value, Automatic switch of user profile from General mode to Silent mode when the user is in the 50m radius of the meeting location. The user can send SMS from this application to the desired and saving the received message.

**1.3 Scope of the project**

This application will assist the users working in management departments, business executives and event management etc. In a company, this application will help in managing the core meetings and operations of the companies, the meetings can be managed easily by the user.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 List of Journals Referred**

* **Title:Android Based Mobile Aplication Devlopment and its security International Journal of Computer Trends and Technology- volume3Issue3- 2012**

**Description:**

In this paper, Android mobile platform for the mobile application development layered approach and the details of security information for Android is discussed.

* **Title:** **Advancement in Mobile Communication using Android ©2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 7**

**Description:**

This paper deals about the introduction of android in mobile devices. The recent trends in android give the future development in software development kit (SDK) which is ubiquitous in all mobile phones. With this use of android in our mobile phone the compatibility accessibility and the interface of the mobile phone version ones has been considerably increased. The latest of android allows the speech recognition facility in the mobile devices. With this further development of android allows us to install video or picture recognition with moving sensors in our mobile phones.

* **Title:Eclipse: A platform for integrating development tools Ibm Systems Journal, Vol 43, No 2, 2004**

**Description:**

The Eclipse Platform was created to address the issues by providing a common platform for diverse IDEbased products and facilitate their integration. The first part of this paper introduces and gives an historical perspective of IDEs, followed by a description of the technical aspects of the Eclipse Platform.

**2.2 List of Articles Referred**

## Title:Android Development with Android Studio or Eclipse ADT – Tutorial

**Description:**

This article describes how to create Android applications. It describes Studio usage of the ADT plug-in for Eclipse. It is based on Android 5.0.

# Title:Android Message: How to send/receive SMS using the built-in messaging application in Android

**Description:**

This article is focused on the practicalities of sending and receiving text and data messages within the Android system

**2.3List of Textbooks Referred**

* **Beginning Android 4 Application Development- Wei-Meng Lee**

**Description:**

This fast-paced introduction to the newest release of Android OS gives aspiring mobile app developers what they need to know to program for today's best Android smart phones and tablets.

* **An**[**droid Application Development All-in-one for Dummies**](http://erclk.about.com/?zi=0/3W%5Ba)

**Description:**

It explains how to download the Android SDK and work with Eclipse in order to get your Android app running. Beginning with the very basics of Android development, it also teaches you how to price your app and submit it to the Android Market.

**CHAPTER-3**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**3.1 Overall Description**

**3.1.1 Product Functions**

The major functionalities of this project are:

* Currency converter.
* Creating meetings and inviting the people.
* Change in mobile phone’s profile.

**3.1.2 User Characteristics**

* User can save time by multitasking.
* User should be dealing with currency conversions/stock market for work purpose.
* Highly recommended for business executives.

**3.1.3 Operating Environment**

Environment in which product operates is described here:

**Software Requirements**

Language: Java

Build Tool: Eclipse, Android SDK

Front end: XML

**Hardware Requirements**

Processor : Duel core or higher

Hard Disk : 20GB

RAM :512MB

Operating System : Windows 7

**3.2 External Interface Requirements**

**3.2.1 User Interfaces**

Operating System: Windows OS and Android Tools

**3.2.2 Hardware Interfaces**

Processor: Single / Dual Core Processor and above

RAM: 2GB RAM

Storage Capacity:10GB HDD

The application requires android OS device to communicate with other devices.

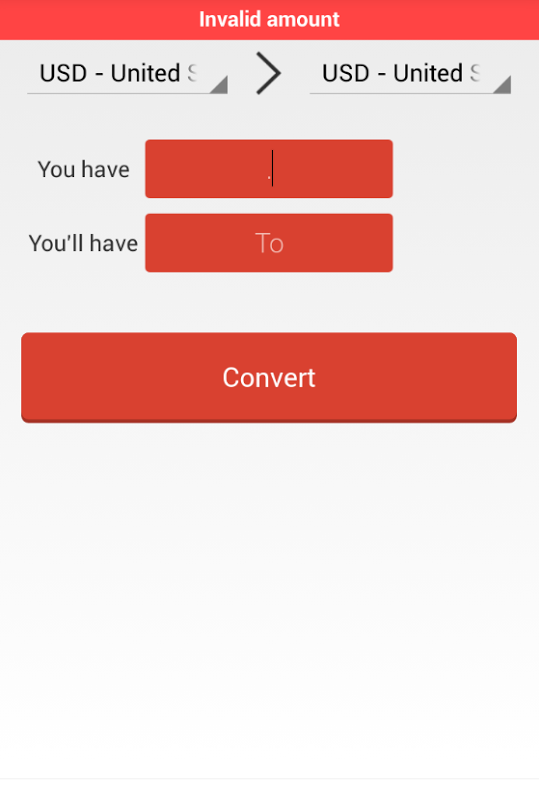
**3.2.3 Software Interfaces**

Programming platform: JDK 1.7.

IDE: Eclipse, Android development kit.

**3.2.4 Communication Interfaces**

**3.2.4.1 Currency converter**

****

**Fig:3.3.4.1.1 Sample UI**

Web Browser: Internet Explorer 5.5 or higher

E-mail Server: James Mail Server 2.3.1

Protocols: POP3, SMTP

**3.3 Functional Requirements**

**3.3.1 Currency converter**

The user inserts the value in a particular currency type and then requests to convert it into the desired currency type. The user can select from the options available and then enter the data to be converted, upon clicking the option as convert the value gets formatted depending on the command.

**3.3.2 Profile change**

The User when arrives within the 50m radius of the venue of the meeting, in order to not to disturb him in the course of the meeting, this system converts the profile of the phone from General(ringing state) to vibration(No sound) state. The position of the venue is recorded and when the user is within the proximity. Then changes the profile of the phone.

**3.3.3 Conduct meetings:**

Users can create meetings and invite different people to it. The user creates the meeting and adds the details to the meeting. Every meeting has its own unique ID to access it and the invited people can confirm his absence/presence from the meeting. The meeting details include location, agenda, number of invitees etc.

**3.4 Non-functional Requirements**

**3.4.1 Performance**

The product shall take initial load time depending on the internet connection strength which also depends on the media from which the product is run. The performance shall depend upon hardware components of the client/customer. System can withstand many number of clients requested the desired service. As Media world is an internet application it provides direct communication with the users of the system where they can get their queries or problems resolved.

**3.4.2 Reliability**

The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended.

**3.4.3 Availability**

The measure of time that the system is up and running correctly; the length of time between failures and the length of time needed to resume operation after a failure.

**3.4.4 Security**

Security Require Administrator is provided with secure login and password and thus cannot be misused.

•The user’s web browser shall never display a user’s password. It shall always be displayed with special characters representing typed characters.

•The user’s web browser shall never display a user’s private information after retrieving from the database.

•The system’s back-end servers shall only be accessible to administrator only.

**3.4.5 Maintainability**

The ability to change the system to deal with new technology or to fix defects

**3.4.6 Efficiency**

It refers to the level at which a software system uses scarce computational resources, such as CPU cycles, memory, disk space, buffers and communication channels.

**CHAPTER-4**

**DESIGN**

**4.1 UML Diagrams**

**4.1.1 Use Case Diagram**

****

Fig 4.1.1: use case diagram for the function

**4.1.2 Class Diagram**

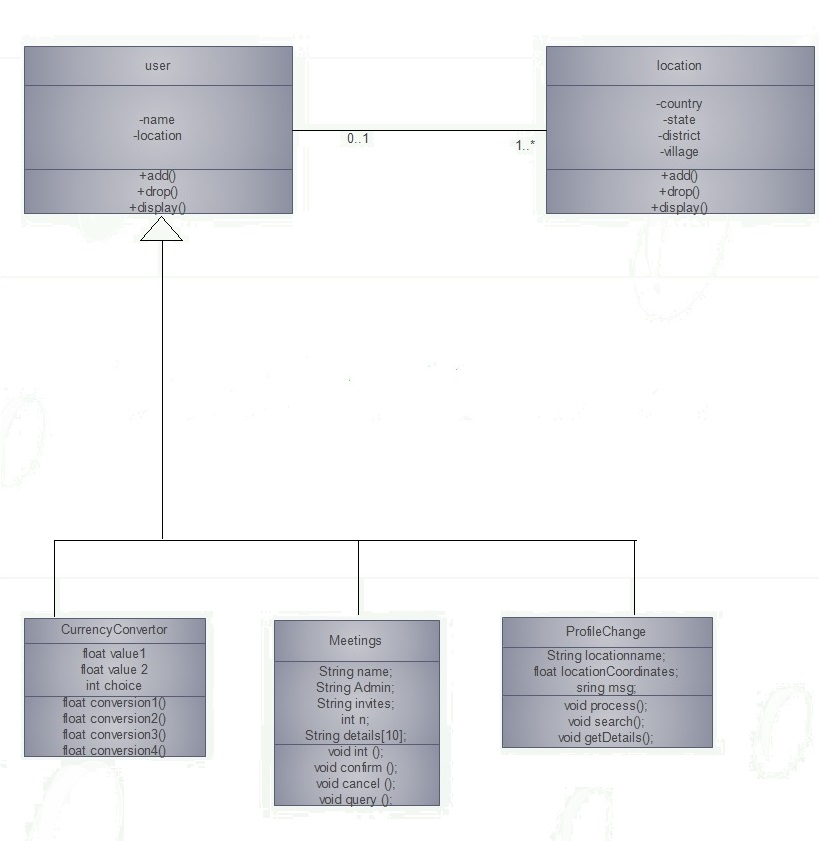
****

Fig 4.1.2: Class diagram

**4.1.3 Sequence Diagram**

**4.1.3.1 Sequence diagram for currency converter system**

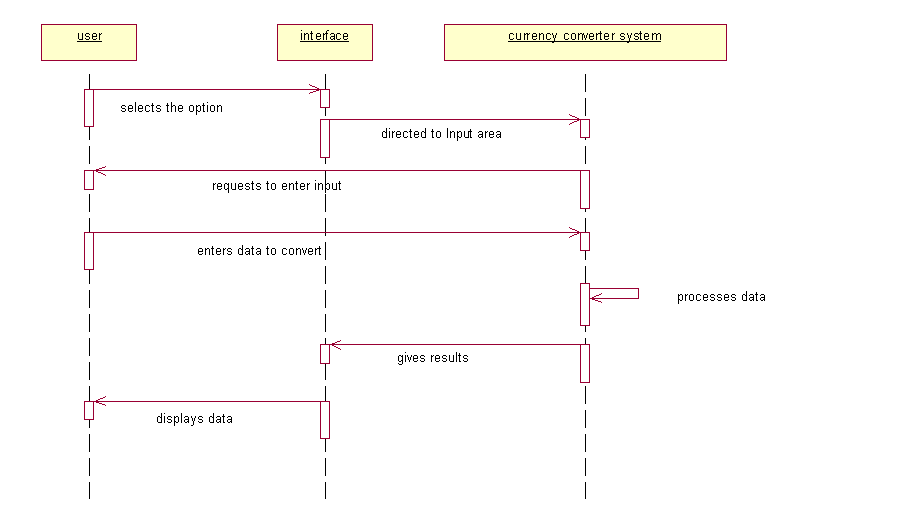
****

Fig 4.1.3.1 Currency converter system

Fig 4.1.3.2 Profile change

Fig 4.1.3.3 meetings system

**4.1.3.2 Sequence diagram for profile change system**

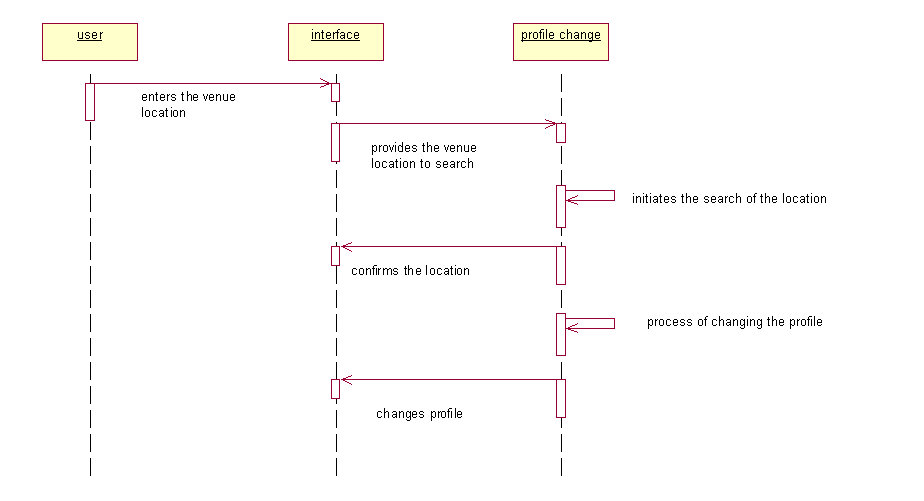
****

Fig 4.1.3.2 Profile change

**4.1.3.3 Sequence diagram for meeting system**

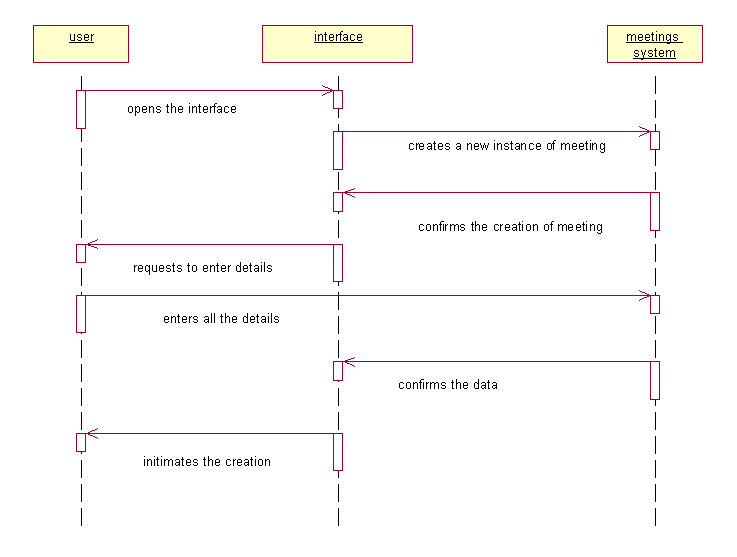
****

Fig 4.1.3.3 meetings system

**CHAPTER-5**

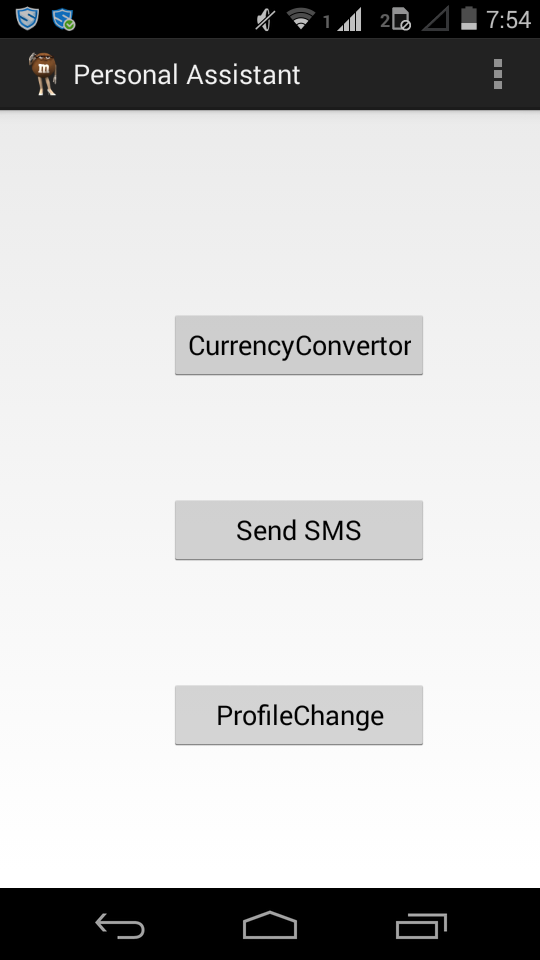
**IMPLEMENTATION**

|  |  |  |
| --- | --- | --- |
| **5.1 Pseudo code**  package com.aurora.personalassistant;  import android.app.Activity;  import android.content.Intent;  import android.os.Bundle;  import android.view.Menu;  import android.view.View;  public class PrincipleActivity extends Activity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_principle);  }  @Override  public boolean onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.principle, menu);  return true;  }  public void onClick(View v) {  switch (v.getId()) {  case R.id.button1:  startActivity(new Intent(PrincipleActivity.this,ConvertActivity.class));  break;  case R.id.button2:  startActivity(new Intent(PrincipleActivity.this,SMSActivity.class));  break;  case R.id.button3:  startActivity(new Intent(PrincipleActivity.this,ProfileActivity.class));  break;  default:  break;  }  }  }    package com.aurora.personalassistant;  import android.app.Activity;  import android.content.Context;  import android.media.AudioManager;  import android.os.Bundle;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.TextView;  import android.widget.Toast;  import android.widget.ToggleButton;  public class ProfileActivity extends Activity {  /\*\* Called when the activity is first created. \*/    ToggleButton tbt;  TextView txtview;    @Override  public void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_profile);    tbt = (ToggleButton) findViewById(R.id.togglebutton);  txtview = (TextView) findViewById(R.id.textview);  txtview.setText("Welcome to Profile Changer Application");  final AudioManager mobilemode = (AudioManager) this.getSystemService(Context.AUDIO\_SERVICE);    tbt.setOnClickListener(new OnClickListener() {    public void onClick(View v) {  // TODO Auto-generated method stub    if(tbt.getText().toString().equals("Switch to LOUD"))  {  mobilemode.setRingerMode(AudioManager.RINGER\_MODE\_SILENT);  txtview.setText("SILENT profile activated !");  Toast.makeText(getBaseContext(),"SILENT profile activated ",Toast.LENGTH\_LONG).show();  }  else if(tbt.getText().toString().equals("Switch to SILENT"))  {  mobilemode.setRingerMode(AudioManager.RINGER\_MODE\_NORMAL);  txtview.setText("LOUD profile activated !");  Toast.makeText(getBaseContext(),"LOUD profile activated !",Toast.LENGTH\_LONG).show();    }    }  });  }  }  package com.aurora.personalassistant;  import android.os.Bundle;  import android.app.Activity;  import android.app.PendingIntent;  import android.content.BroadcastReceiver;  import android.content.Context;  import android.content.Intent;  import android.content.IntentFilter;  import android.telephony.SmsManager;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.Toast;  public class SMSActivity extends Activity {  Button sendSMS;  EditText msgTxt;  EditText numTxt;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_sms);    sendSMS=(Button)findViewById(R.id.sendBtn);  msgTxt=(EditText)findViewById(R.id.message);  numTxt=(EditText)findViewById(R.id.numberTxt);  sendSMS.setOnClickListener(new View.OnClickListener()  {  public void onClick(View v)  {  String myMsg=msgTxt.getText().toString();  String theNumber=numTxt.getText().toString();  sendMsg(theNumber,myMsg);  Intent i = new Intent(android.content.Intent.ACTION\_VIEW);  i.putExtra("address", "5556; 5558");  // i.putExtra("address", "5556; 5558; 5560");  // here i can send message to emulator 5556,5558,5560  // you can change in real device  i.putExtra("sms\_body", "Hello my friends!");  i.setType("vnd.android-dir/mms-sms");  startActivity(i);  }  });  }  protected void sendMsg(String theNumber, String myMsg) {  String SENT="Message Sent";  String DELIVERED="Message Delivered";    PendingIntent sentPI= PendingIntent.getBroadcast(this, 0, new Intent(SENT), 0);  PendingIntent deliveredPI= PendingIntent.getBroadcast(this, 0, new Intent(DELIVERED), 0);    registerReceiver(new BroadcastReceiver()  {  public void onReceive(Context arg0, Intent arg1)  {  switch(getResultCode())  {  case Activity.RESULT\_OK:  Toast.makeText(SMSActivity.this,"SMS sent",Toast.LENGTH\_LONG).show();  break;  case SmsManager.RESULT\_ERROR\_GENERIC\_FAILURE:  Toast.makeText(getBaseContext(), "Generic Failure", Toast.LENGTH\_LONG).show();  break;  case SmsManager.RESULT\_ERROR\_NO\_SERVICE:  Toast.makeText(getBaseContext(),"No service", Toast.LENGTH\_LONG).show();  break;  }  }  },new IntentFilter(SENT));    registerReceiver(new BroadcastReceiver()  {  public void onReceive(Context arg0, Intent arg1)  {  switch(getResultCode())  {  case Activity.RESULT\_OK:  Toast.makeText(getBaseContext(),"SMS Delivered",Toast.LENGTH\_LONG).show();  break;  case Activity.RESULT\_CANCELED:  Toast.makeText(getBaseContext(), "SMS not delivered", Toast.LENGTH\_LONG).show();  break;  }  }  },new IntentFilter(DELIVERED));  SmsManager sms=SmsManager.getDefault();  sms.sendTextMessage(theNumber, null,myMsg,sentPI,deliveredPI);  }  };   |  | | --- | | * 1. Pseudo code   package com.aurora.personalassistant;  import android.app.Activity;  import android.content.Intent;  import android.os.Bundle;  import android.view.Menu;  import android.view.View;  public class PrincipleActivity extends Activity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_principle);  }  @Override  public boolean onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.principle, menu);  return true;  }  public void onClick(View v) {  switch (v.getId()) {  case R.id.button1:  startActivity(new Intent(PrincipleActivity.this,ConvertActivity.class));  break;  case R.id.button2:  startActivity(new Intent(PrincipleActivity.this,SMSActivity.class));  break;  case R.id.button3:  startActivity(new Intent(PrincipleActivity.this,ProfileActivity.class));  break;  default:  break;  }  }  }  package com.aurora.personalassistant;  import org.json.JSONException;  import org.json.JSONObject;  import com.actionbarsherlock.app.SherlockActivity;  import com.loopj.android.http.AsyncHttpClient;  import com.loopj.android.http.AsyncHttpResponseHandler;  import android.os.Bundle;  import android.util.Log;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import android.widget.Toast;  public class ConvertActivity extends SherlockActivity  {  private static final String URL = "http://openexchangerates.org/api/latest.json?app\_id=f2381be937714114b86274f5ea360b7d";  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_convert);    Button btnDo = (Button)findViewById(R.id.buttonDo);  final EditText usdValue = (EditText) findViewById(R.id.editTextUSD);  final TextView gbpValue = (TextView) findViewById(R.id.textViewGBP);  final TextView eurValue = (TextView) findViewById(R.id.textViewEURO);  final TextView inrValue = (TextView) findViewById(R.id.textViewINR);    btnDo.setOnClickListener(new OnClickListener() {  @Override  public void onClick(View arg0) {    if (! usdValue.getText().toString().equals("") ) {  AsyncHttpClient client = new AsyncHttpClient();  client.get(URL, new AsyncHttpResponseHandler() {  @Override  public void onFailure(Throwable arg0, String arg1) {  // TODO Auto-generated method stub  super.onFailure(arg0, arg1);  }  @Override  public void onFinish() {  // TODO Auto-generated method stub  super.onFinish();  }  @Override  public void onStart() {  // TODO Auto-generated method stub  super.onStart();  }  @Override  public void onSuccess(String response) {  Log.i("Currencyconvertor", "HTTP Sucess");  // TODO Auto-generated method stub  try {  JSONObject jsonObj = new JSONObject(response);  JSONObject ratesObject = jsonObj  .getJSONObject("rates");  Double gbpRate = ratesObject.getDouble("GBP");  Double eurRate = ratesObject.getDouble("EUR");  Double inrRate = ratesObject.getDouble("INR");  Log.i("Currencyconvertor", "GBP:" + gbpRate);  Log.i("Currencyconvertor", "EUR:" + eurRate);  Log.i("Currencyconvertor", "INR:" + inrRate);    Double usds = Double.valueOf( usdValue.getText().toString());  Double gbps = usds \* gbpRate;  Double euros = usds \* eurRate;  Double inrs = usds \* inrRate;    inrValue.setText("INR:" + String.valueOf(inrs));  gbpValue.setText("GBP:" + String.valueOf(gbps));  eurValue.setText("EURO:" + String.valueOf(euros));      }  catch (JSONException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }  });    }  else {  Toast.makeText(getApplicationContext(),  "please enter a USD value ",  Toast.LENGTH\_LONG).show();  }    }  });        }}    package com.aurora.personalassistant;  import android.app.Activity;  import android.content.Context;  import android.media.AudioManager;  import android.os.Bundle;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.TextView;  import android.widget.Toast;  import android.widget.ToggleButton;  public class ProfileActivity extends Activity {  /\*\* Called when the activity is first created. \*/    ToggleButton tbt;  TextView txtview;    @Override  public void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_profile);    tbt = (ToggleButton) findViewById(R.id.togglebutton);  txtview = (TextView) findViewById(R.id.textview);  txtview.setText("Welcome to Profile Changer Application");  final AudioManager mobilemode = (AudioManager) this.getSystemService(Context.AUDIO\_SERVICE);    tbt.setOnClickListener(new OnClickListener() {    public void onClick(View v) {  // TODO Auto-generated method stub    if(tbt.getText().toString().equals("Switch to LOUD"))  {  mobilemode.setRingerMode(AudioManager.RINGER\_MODE\_SILENT);  txtview.setText("SILENT profile activated !");  Toast.makeText(getBaseContext(),"SILENT profile activated ",Toast.LENGTH\_LONG).show();  }  else if(tbt.getText().toString().equals("Switch to SILENT"))  {  mobilemode.setRingerMode(AudioManager.RINGER\_MODE\_NORMAL);  txtview.setText("LOUD profile activated !");  Toast.makeText(getBaseContext(),"LOUD profile activated !",Toast.LENGTH\_LONG).show();    }    }  });  }  }  package com.aurora.personalassistant;  import android.os.Bundle;  import android.app.Activity;  import android.app.PendingIntent;  import android.content.BroadcastReceiver;  import android.content.Context;  import android.content.Intent;  import android.content.IntentFilter;  import android.telephony.SmsManager;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.Toast;  public class SMSActivity extends Activity {  Button sendSMS;  EditText msgTxt;  EditText numTxt;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_sms);    sendSMS=(Button)findViewById(R.id.sendBtn);  msgTxt=(EditText)findViewById(R.id.message);  numTxt=(EditText)findViewById(R.id.numberTxt);  sendSMS.setOnClickListener(new View.OnClickListener()  {  public void onClick(View v)  {  String myMsg=msgTxt.getText().toString();  String theNumber=numTxt.getText().toString();  sendMsg(theNumber,myMsg);  Intent i = new Intent(android.content.Intent.ACTION\_VIEW);  i.putExtra("address", "5556; 5558");  // i.putExtra("address", "5556; 5558; 5560");  // here i can send message to emulator 5556,5558,5560  // you can change in real device  i.putExtra("sms\_body", "Hello my friends!");  i.setType("vnd.android-dir/mms-sms");  startActivity(i);  }  });  }  protected void sendMsg(String theNumber, String myMsg) {  String SENT="Message Sent";  String DELIVERED="Message Delivered";    PendingIntent sentPI= PendingIntent.getBroadcast(this, 0, new Intent(SENT), 0);  PendingIntent deliveredPI= PendingIntent.getBroadcast(this, 0, new Intent(DELIVERED), 0);    registerReceiver(new BroadcastReceiver()  {  public void onReceive(Context arg0, Intent arg1)  {  switch(getResultCode())  {  case Activity.RESULT\_OK:  Toast.makeText(SMSActivity.this,"SMS sent",Toast.LENGTH\_LONG).show();  break;  case SmsManager.RESULT\_ERROR\_GENERIC\_FAILURE:  Toast.makeText(getBaseContext(), "Generic Failure", Toast.LENGTH\_LONG).show();  break;  case SmsManager.RESULT\_ERROR\_NO\_SERVICE:  Toast.makeText(getBaseContext(),"No service", Toast.LENGTH\_LONG).show();  break;  }  }  },new IntentFilter(SENT));    registerReceiver(new BroadcastReceiver()  {  public void onReceive(Context arg0, Intent arg1)  {  switch(getResultCode())  {  case Activity.RESULT\_OK:  Toast.makeText(getBaseContext(),"SMS Delivered",Toast.LENGTH\_LONG).show();  break;  case Activity.RESULT\_CANCELED:  Toast.makeText(getBaseContext(), "SMS not delivered", Toast.LENGTH\_LONG).show();  break;  }  }  },new IntentFilter(DELIVERED));  SmsManager sms=SmsManager.getDefault();  sms.sendTextMessage(theNumber, null,myMsg,sentPI,deliveredPI);  }  };  XML Code:  <?xml version=*"1.0"* encoding=*"utf-8"*?>  <manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*  package=*"com.aurora.personalassistant"*  android:versionCode=*"1"*  android:versionName=*"1.0"* >  <uses-sdk  android:minSdkVersion=*"8"*  android:targetSdkVersion=*"18"* />  <uses-permission android:name=*"android.permission.SEND\_SMS"* />  <uses-permission android:name=*"android.permission.RECEIVE\_SMS"* />  <uses-permission android:name=*"android.permission.INTERNET"*/>  <application  android:allowBackup=*"true"*  android:icon=*"@drawable/ic\_launcher"*  android:label=*"@string/app\_name"*  android:theme=*"@style/AppTheme"* >  <activity  android:name=*"com.aurora.personalassistant.PrincipleActivity"*  android:label=*"@string/app\_name"* >  <intent-filter>  <action android:name=*"android.intent.action.MAIN"* />  <category android:name=*"android.intent.category.LAUNCHER"* />  </intent-filter>  </activity>  <activity android:name=*"com.aurora.personalassistant.ConvertActivity"*  android:theme=*"@style/Theme.Sherlock"*  android:label=*"@string/app\_name"* >  </activity>    <activity android:name=*"com.aurora.personalassistant.ProfileActivity"*>  </activity>      <activity android:name=*"com.aurora.personalassistant.SMSActivity"* >  </activity>  <receiver android:name=*"com.example.sendsms.SMSReceiver"* >  </receiver>  </application>  </manifest>  <?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"* >  <TextView android:layout\_width=*"fill\_parent"*  android:layout\_height=*"wrap\_content"*  android:id=*"@+id/textview"*/>    <ToggleButton android:id=*"@+id/togglebutton"*  android:layout\_width=*"150px"*  android:layout\_height=*"100px"*  android:textOn=*"Switch to LOUD"*  android:textOff=*"Switch to SILENT"* />  </LinearLayout>  <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:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*".PrincipleActivity"* >  <Button  android:id=*"@+id/button1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_above=*"@+id/button2"*  android:layout\_alignParentRight=*"true"*  android:layout\_marginBottom=*"75dp"*  android:layout\_marginRight=*"58dp"*  android:onClick=*"onClick"*  android:text=*"CurrencyConvertor"* />  <Button  android:id=*"@+id/button2"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_above=*"@+id/button3"*  android:layout\_alignLeft=*"@+id/button1"*  android:layout\_alignRight=*"@+id/button1"*  android:layout\_marginBottom=*"75dp"*  android:onClick=*"onClick"*  android:text=*"Send SMS"* />  <Button  android:id=*"@+id/button3"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignLeft=*"@+id/button2"*  android:layout\_alignParentBottom=*"true"*  android:layout\_alignRight=*"@+id/button2"*  android:layout\_marginBottom=*"75dp"*  android:onClick=*"onClick"*  android:text=*"ProfileChange"* />  </RelativeLayout>  <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:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*".SMSActivity"* >  <TextView  android:id=*"@+id/numberTxt"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentTop=*"true"*  android:layout\_marginTop=*"62dp"*  android:hint=*"Enter number"*  android:textAppearance=*"?android:attr/textAppearanceMedium"* />  <TextView  android:id=*"@+id/message"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignLeft=*"@+id/textView1"*  android:layout\_below=*"@+id/textView1"*  android:layout\_marginTop=*"33dp"*  android:hint=*"enter message"*  android:textAppearance=*"?android:attr/textAppearanceMedium"* />  <Button  android:id=*"@+id/sendBtn"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignLeft=*"@+id/textView2"*  android:layout\_centerVertical=*"true"*  android:hint=*"Send"* />  </RelativeLayout>  <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:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*".ConvertActivity"* >  <EditText  android:id=*"@+id/editTextUSD"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignBaseline=*"@+id/textView1"*  android:layout\_alignBottom=*"@+id/textView1"*  android:layout\_toRightOf=*"@+id/textView1"*  android:ems=*"10"* >  <requestFocus />  </EditText>  <TextView  android:id=*"@+id/textView1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentLeft=*"true"*  android:layout\_alignParentTop=*"true"*  android:layout\_marginTop=*"28dp"*  android:text=*"USD:"*  android:textAppearance=*"?android:attr/textAppearanceMedium"* />  <TextView  android:id=*"@+id/textViewINR"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentLeft=*"true"*  android:layout\_below=*"@+id/editTextUSD"*  android:layout\_marginTop=*"50dp"*  android:text=*"INR:"*  android:textAppearance=*"?android:attr/textAppearanceLarge"* />  <TextView  android:id=*"@+id/textViewGBP"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignLeft=*"@+id/textViewEURO"*  android:layout\_below=*"@+id/textViewEURO"*  android:layout\_marginTop=*"39dp"*  android:text=*"GBP:"*  android:textAppearance=*"?android:attr/textAppearanceLarge"* />  <TextView  android:id=*"@+id/textViewEURO"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignLeft=*"@+id/textViewINR"*  android:layout\_below=*"@+id/textViewINR"*  android:layout\_marginTop=*"38dp"*  android:text=*"EURO:"*  android:textAppearance=*"?android:attr/textAppearanceLarge"* />  <Button  android:id=*"@+id/buttonDo"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_below=*"@+id/textViewGBP"*  android:layout\_centerHorizontal=*"true"*  android:layout\_marginTop=*"17dp"*  android:text=*"Convert"* />  <TextView  android:id=*"@+id/textView2"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentBottom=*"true"*  android:layout\_centerHorizontal=*"true"*  android:layout\_marginBottom=*"16dp"*  android:text=*"Currency Convertor"*  android:textAppearance=*"?android:attr/textAppearanceMedium"* />  </RelativeLayout>  <resources>  <!-- Default screen margins, per the Android Design guidelines. -->  <dimen name=*"activity\_horizontal\_margin"*>16dp</dimen>  <dimen name=*"activity\_vertical\_margin"*>16dp</dimen>  </resources>  <?xml version=*"1.0"* encoding=*"utf-8"*?>  <resources>  <string name=*"app\_name"*>Personal Assistant</string>  <string name=*"action\_settings"*>Settings</string>  <string name=*"hello\_world"*>Hello world!</string>  </resources>  <resources>  <!--  Base application theme, dependent on API level. This theme is replaced  by AppBaseTheme from res/values-vXX/styles.xml on newer devices.  -->  <style name=*"AppBaseTheme"* parent=*"android:Theme.Light"*>  <!--  Theme customizations available in newer API levels can go in  res/values-vXX/styles.xml, while customizations related to  backward-compatibility can go here.  -->  </style>  <!-- Application theme. -->  <style name=*"AppTheme"* parent=*"AppBaseTheme"*>  <!-- All customizations that are NOT specific to a particular API-level can go here. -->  </style>  </resources>  <menu xmlns:android=*"http://schemas.android.com/apk/res/android"* >  <item  android:id=*"@+id/action\_settings"*  android:orderInCategory=*"100"*  android:showAsAction=*"never"*  android:title=*"@string/action\_settings"*/>  </menu> | | SC  S |   **package** com.aurora.personalassistant;  **import** android.content.BroadcastReceiver;  **import** android.content.Context;  **import** android.content.Intent;  **import** android.os.Bundle;  **import** android.telephony.SmsMessage;  **import** android.widget.Toast;  **public** **class** SMSReceiver **extends** BroadcastReceiver {  @Override  **public** **void** onReceive(Context context, Intent intent) {  Bundle bundle=intent.getExtras();  SmsMessage[] messages=**null**;  String str="";    **if**(bundle!=**null**)  {  Object[] pdus=(Object[]) bundle.get("pdus");  messages=**new** SmsMessage[pdus.length];  //PDUS- protocol description unit    **for**(**int** i=0;i<messages.length;i++)  {  messages[i]=SmsMessage.*createFromPdu*((**byte**[])pdus[i]);  str += "Message from" + messages[i].getDisplayOriginatingAddress();  str +=":";  str += messages[i].getMessageBody().toString();  str +="\n";  }  //display message  Toast.*makeText*(context, str, Toast.*LENGTH\_LONG*).show();        }  } |

**CHAPTER-6**

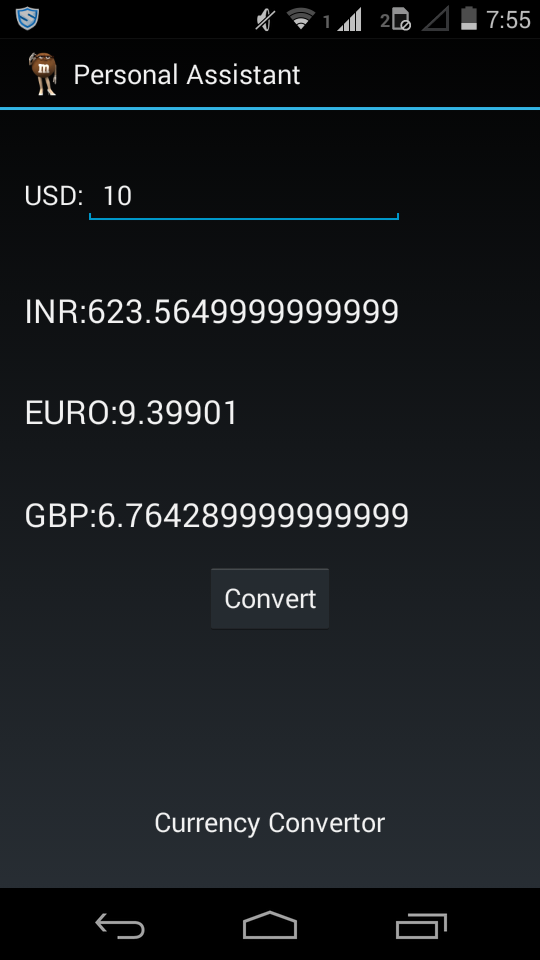
**Discussion of results**

**6.1 Screen shots**

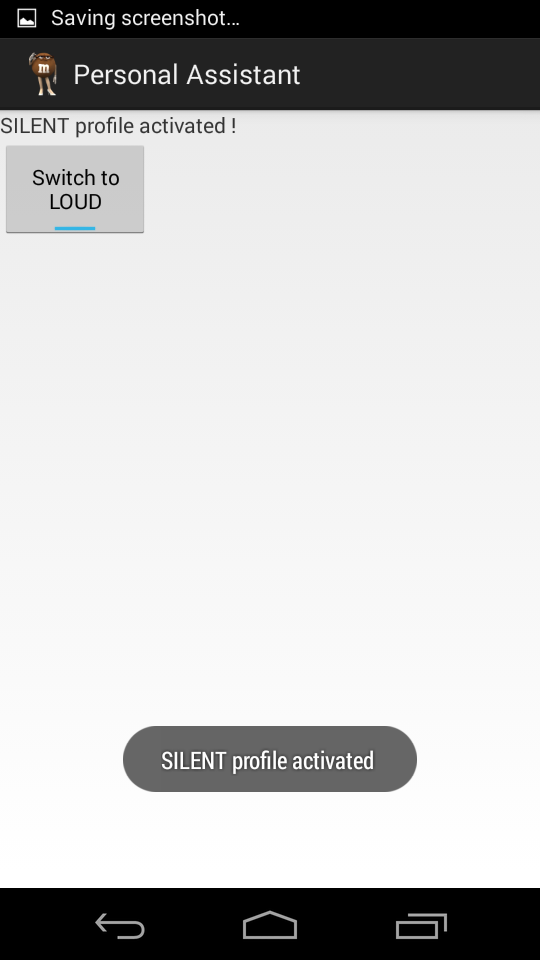
****

**Fig 6.1.1 Final User Interface**

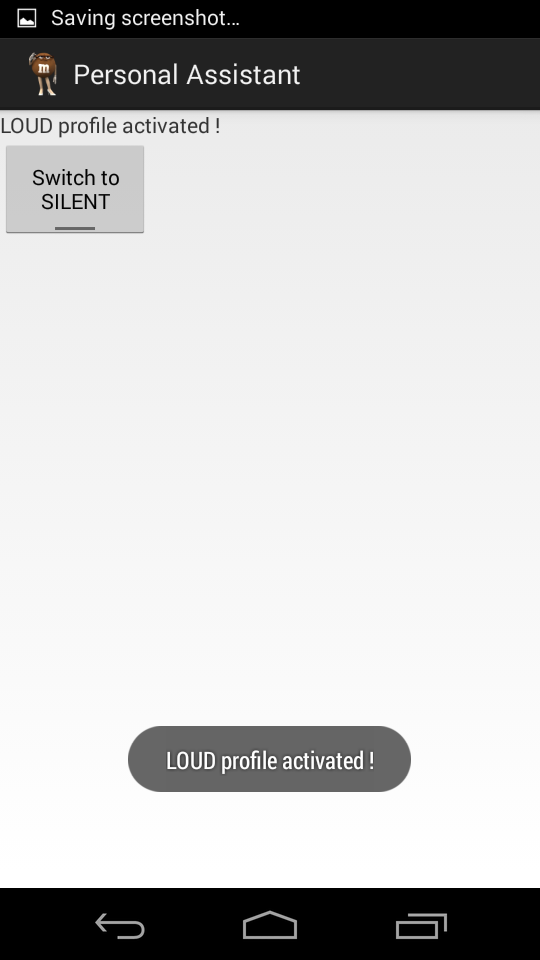
|  |
| --- |
|  |
|  |

****

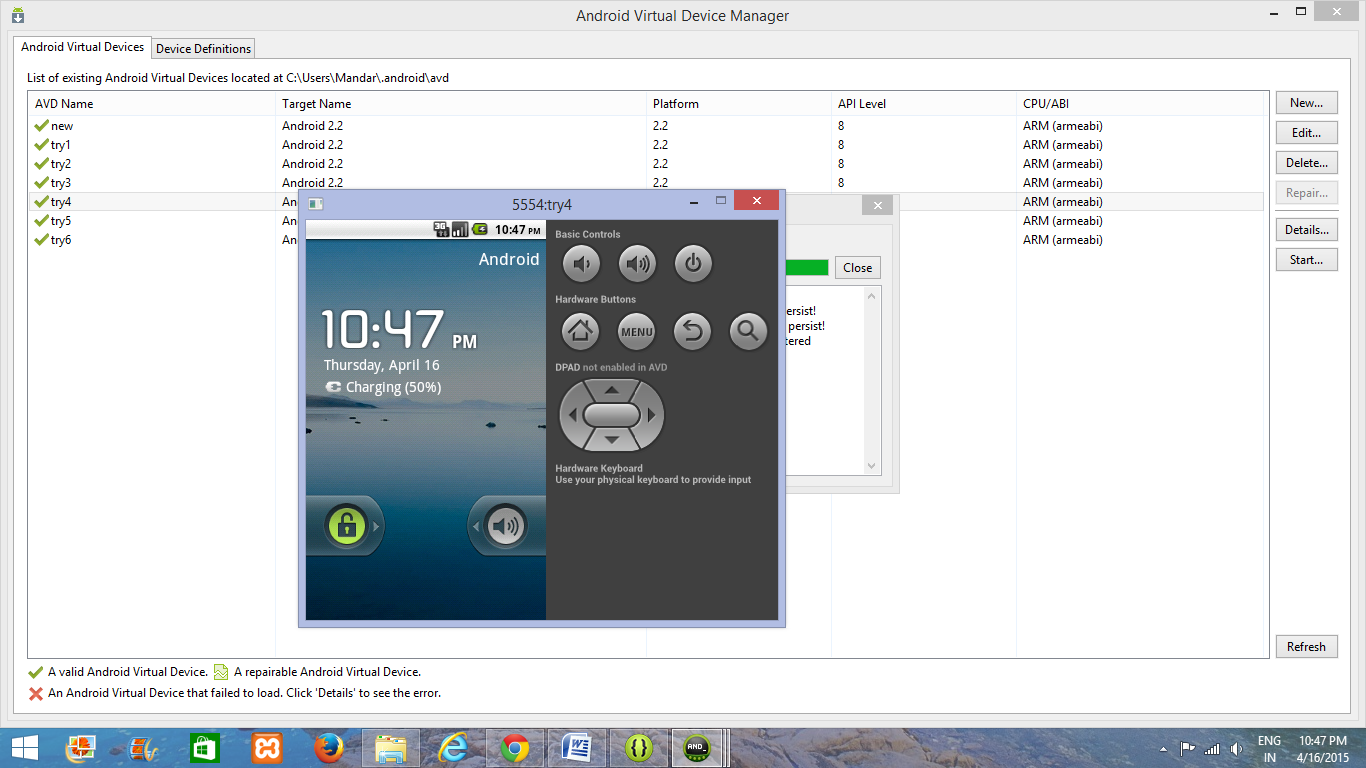
**Fig 6.1.2 Execution of Currency convertor**

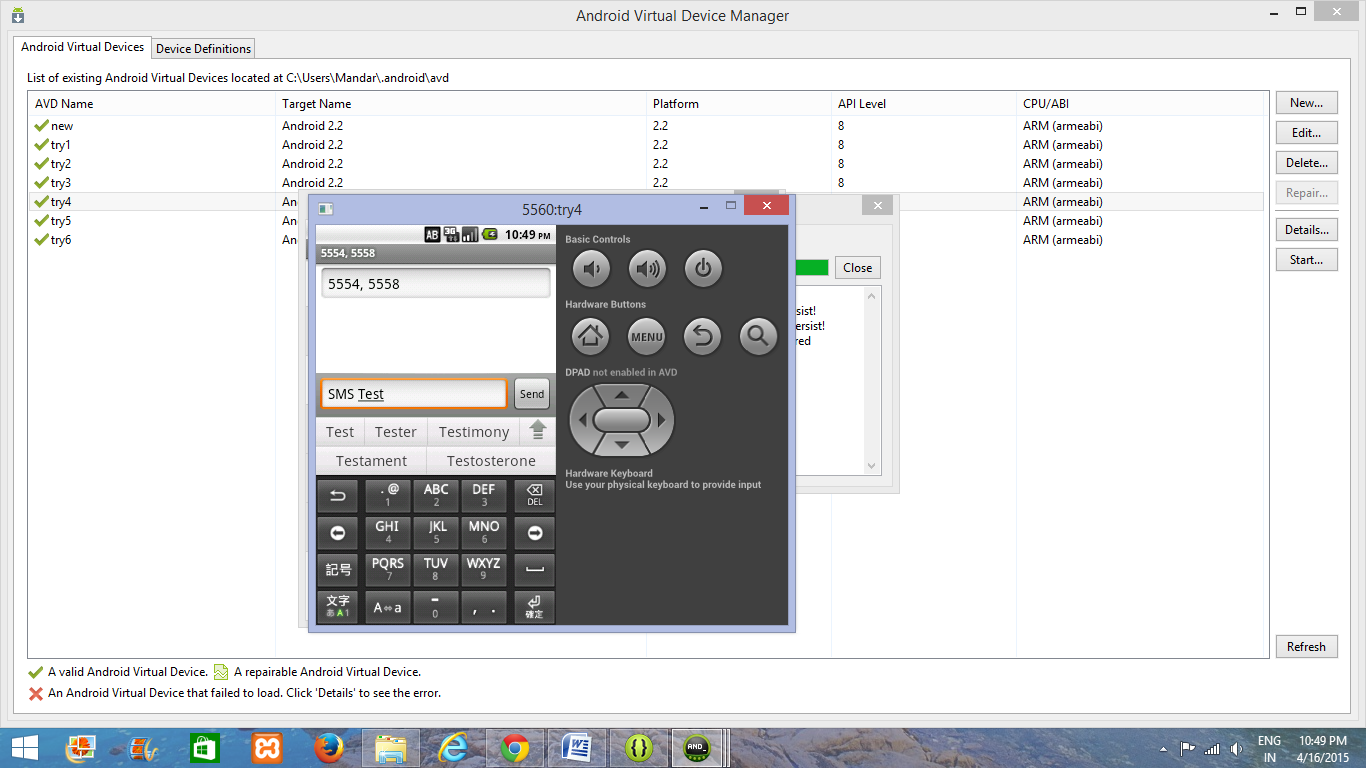


**Fig 6.1.3 Execution of profile change**

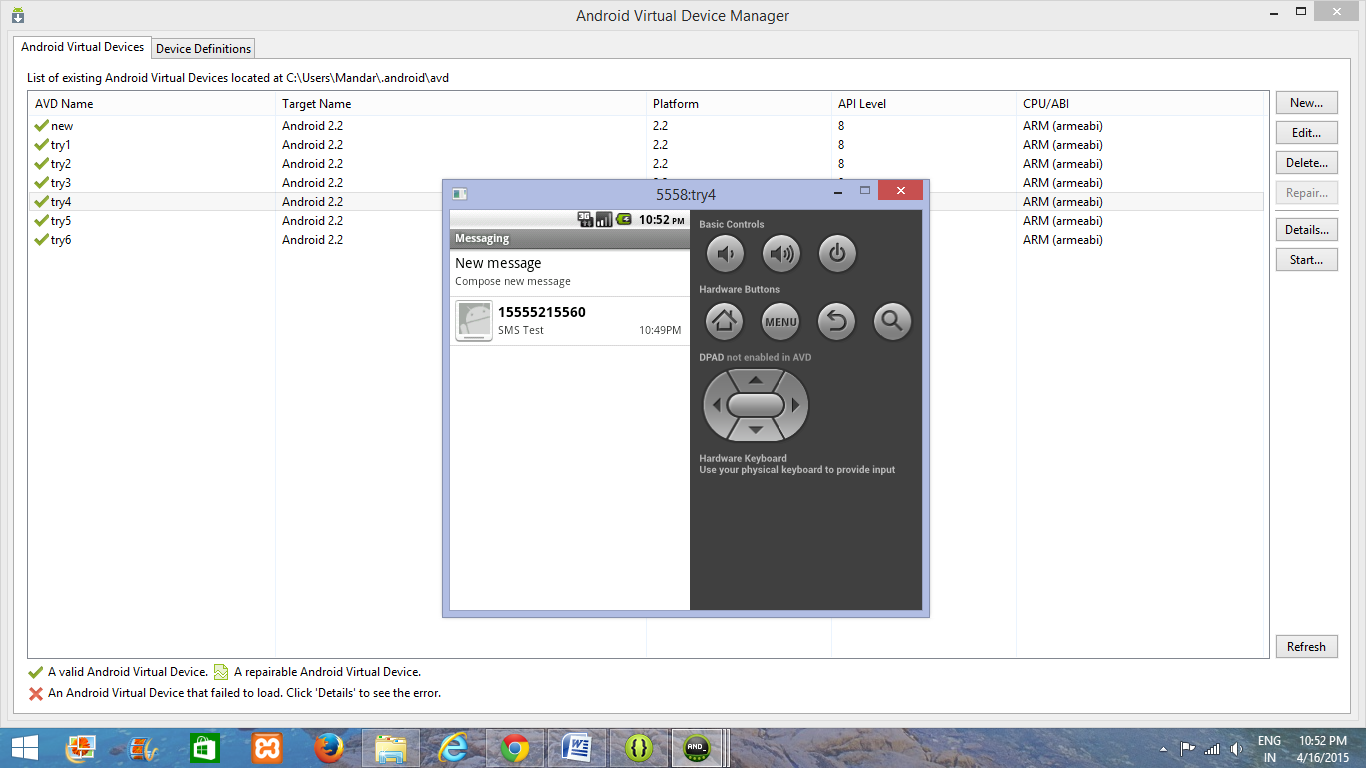


**Fig 6.1.4 Execution of profile change-vive versa**



**Fig:6.1.5 Emulator opened to send SMS**

**Fig:6.1.6 Sending SMS and marking receipients**



**Fig:6.1.7 SMS received in the emulator**

**CHAPTER-7**

**TESTING**

1. **Testing Mythologies**
   1. **Testing in the Iterative Lifecycle**

Testing is not a single activity, nor is it a phase in the project during which we assess quality. If developers are to obtain timely feedback on evolving product quality, testing must occur throughout the lifecycle: we can test the broad functionality of early prototypes: we can test the stability, coverage and performance of the architecture while there is still an opportunity to fix it; and we can test the final product to assess its readiness for delivery to customers.

* 1. **Dimensions of Testing**

To assess product quality, different kinds of tests, each one with a different focus, are needed. These tests can be categorized by several dimensions:

Quality dimension: The major quality characteristic or attribute that is the focus of test.

Stage of testing: The point in the lifecycle at which the test, usually limited to a single quality dimension.

Type of testing: The specific test objective for an individual test, usually limited to a single quality dimension.

* 1. **Stages of Testing**

Testing is not a single activity, executed all at once. Testing is executed against different types of targets in different stages of the software development. Test stages progress from testing small elements of the system, such as components (unit testing), to testing completed systems (system testing). The four stages have the following purposes:

**Unit test***:* The smallest testable elements of the system are tested individually, typically at the same time those elements are implemented.

**Integration test***:* The integrated units (or components or subsystems) are tested.

**System test**: the complete application and system (one or more applications) are tested.

**Acceptance test**: The complete application (or system) is tested by end users (or representatives) for the purpose of determining readiness for deployment.

These stages occur throughout the lifecycle, with varying emphasis. An early conceptual prototype user in the inception phase to assess the viability of the product vision will be subjected to acceptance tests. Architectural prototype developed during the elaboration phase be subjected to integration and system tests to validate architectural integrity and performance of key architectural elements.

* 1. **Types of Testing**

After a test plan has been developed, system testing begins by testing program modules separately, followed by testing “bundled” modules as a unit. A program module may function perfectly in isolation but fail when interfaced with other modules. The approach is to test each entity with successively larger ones, up to the system test level.

System testing consists of the following steps:

* Program(s) testing.
* String testing.
* System testing.
* User acceptance testing.

**Program testing**

A program represents the logical elements of system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. Achieving an error-free program is the responsibility of the programmer. Program testing checks for two types of errors: syntax and logic. A syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted key words are common syntax errors. These errors are shown through error messages generated by syntax errors. These errors are shown through error messages generated by the computer. A logic error, on the other hand, deals with incorrect data fields, out-of range items, and invalid combinations. Since diagnostics do not detect logic errors, the programmer must examine the output carefully for them.

When a program is tested, the actual output is compared with the expected output. When there is a discrepancy, the sequence of instructions must be traced to determine the problem. The process is facilitated by breaking the program down into self-contained portions, each of which can be checked at certain key points. The idea is to compare program values against desk-calculated values to isolate the problem.

**String Testing**

Programs are invariably related to one another and interact in a total system. Each program is tested to see whether it conforms to related programs in the system. Each portion of the system is tested against the entire module with both test and live data before the entire system is ready to be tested.

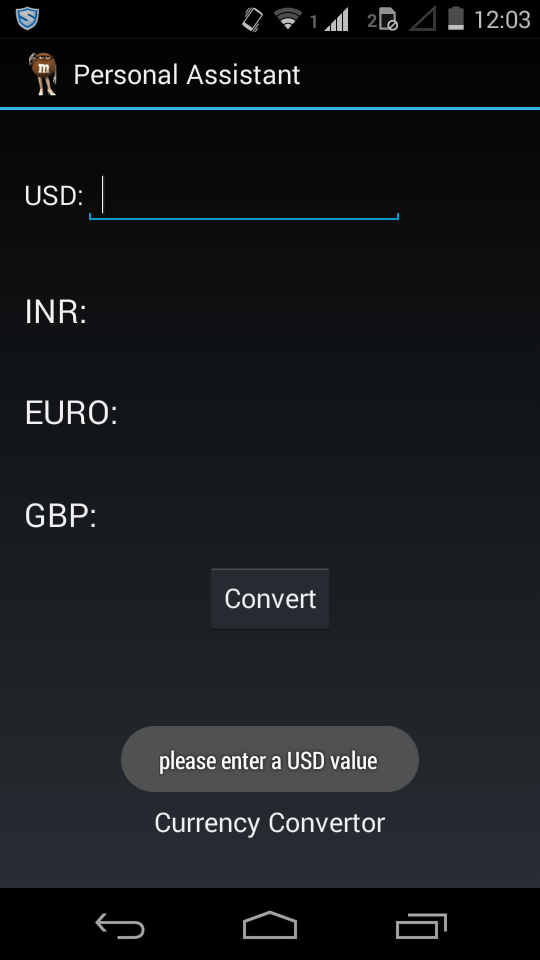
**System Testing**

System testing is designed to uncover weaknesses that were not found in earlier tests. This includes forced system failure and validation of the total system as it will be implemented by its user(s) in the operational environment. Generally, it begins with low volumes of transactions based on live data. The volume is increased until the maximum level for each transaction type is reached. The total system is also tested for recovery and fallback after various major failures to ensure that no data are lost during the emergency. All this is done with the old system still in operation. After the candidate system passes the test, the old system is discontinued.

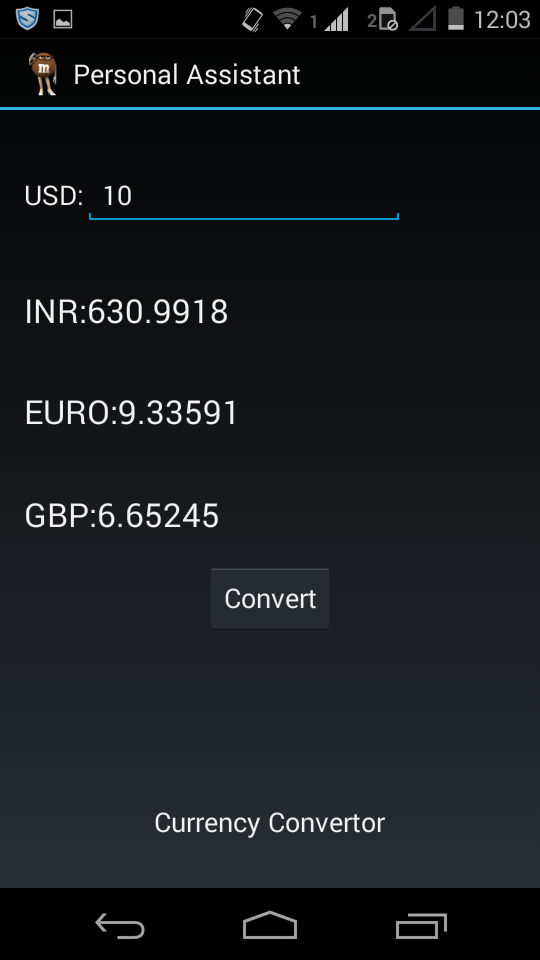
**User Acceptance Testing**

An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system’s procedures operate to system specifications and that the integrity of vital data is maintained. Performance of an acceptance test is actually the user’s show. User motivation and knowledge are critical for the successful performance of the system. Then a comprehensive test report is prepared. The report indicates the system’s tolerance, performance range, error rate, and accuracy.

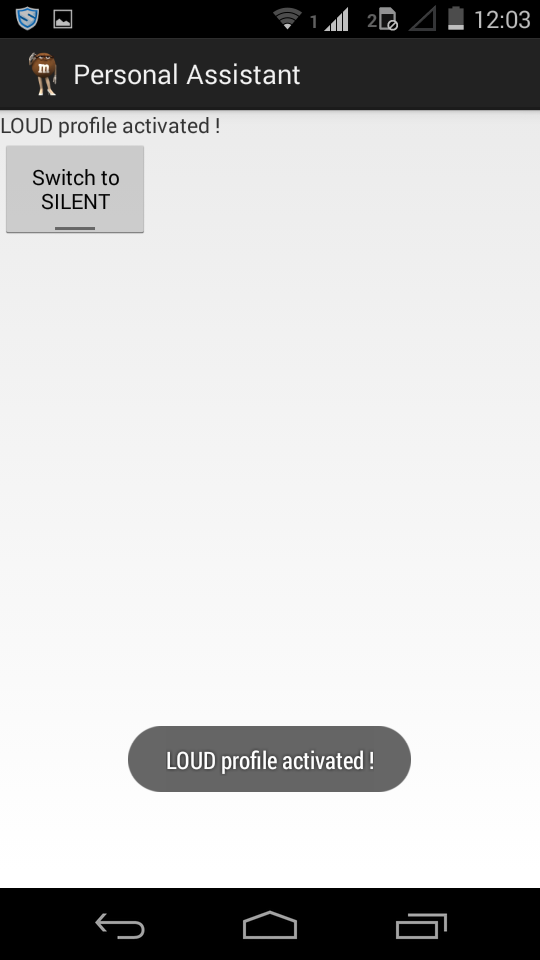
The above different kind of steps have led us to the following Test results.



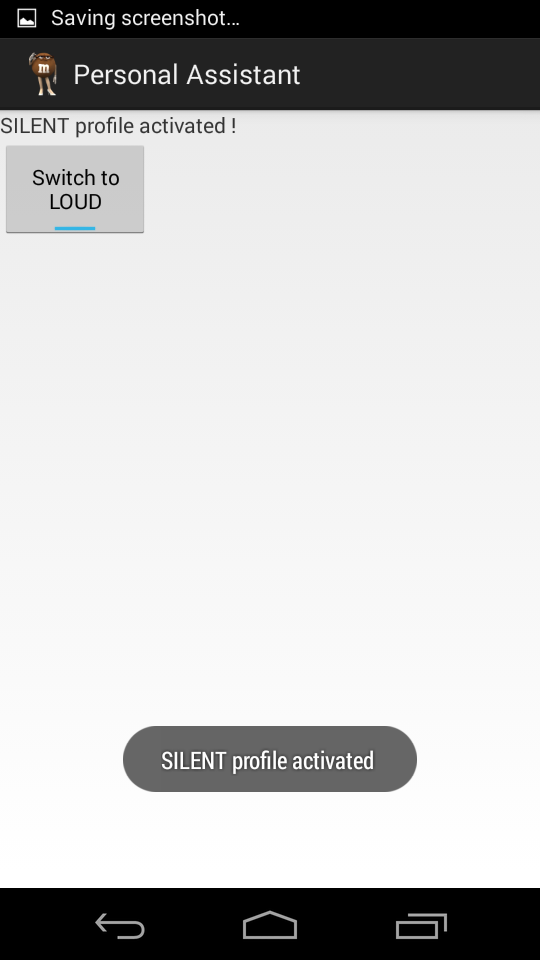
**Fig: 7.1.1 no value**

****

**Fig:7.1.2 legal value**

****

**Fig:** **7.1.3 profile change**

****

**Fig:** **7.1.4 profile change**

**7.2 Test case tables**

Test Case #1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Type** | **Test Case Description** | **Expected Value** | **Actual Value** | **Result** |
| 1. | Checking the internet connection. | We need to go to the network connections and get connected to the internet. | Internet connected. | Internet connected successfully. | Pass |
| 2. | Unable to start the internet services. | Internet is not getting connected due to some network problems. | Error in connecting internet. | No internet connection | Pass |

Test Case #2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Type** | **Test Case Description** | **Expected Value** | **Actual Value** | **Result** |
| 1. | No input | No value entered in the USD textbox | Toast message of,”please enter USD value” | Toast message of,”please enter USD value” | Pass |
| 2. | Input given | Entering numerical 10 for USD value | Relevant conversion to INR,Euro,GBP | Relevant conversion to INR,Euro,GBP | Pass |

Test Case #3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Type** | **Test Case Description** | **Expected Value** | **Actual Value** | **Result** |
| 1. | Clicking on Toggle button | Change of profile | Change of profile from general to silent | Change of profile from general to silent | Pass |
| 2. | Clicking on Toggle button | Change of profile | Change of profile from silent to general | Change of profile from silent to general | Pass |

Test Case #4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Type** | **Test Case Description** | **Expected Value** | **Actual Value** | **Result** |
| 1. | Multiple SMS | Multiple numbers added as recipients. | Message sent to all the numbers | Message sent to all the numbers | Pass |
| 2. | Multiple SMS | Multiple invalid numbers added as recipients. | Message not sent to all the numbers | Message not sent to all the numbers | Pass |

**CHAPTER-8**

**CONCLUSION AND FUTURE SCOPE**

**8.1 Conclusion:**

This Android Application is useful to the business executives and to mange task well, the target user will ideally be someone who wants to mange task well. Currency converter gives the exact conversion rate of the values. Profile change module helps in transformation of mobile phone’s profile from general to silent and vice versa. Group SMS module helps the user to send the SMS to multiple recipients from one device.

**8.2 Future Scope:**

This application will assist the users working in management departments, business executives and event management etc. In a company, this application will help in managing the core meetings and operations of the companies, the meetings can be managed easily by the user.

**8.3 References**

http://www.computerworld.com/article/2514892/app-development/how-to-build-an-android-application-step-by-step.html

https://developer.android.com/training/basics/firstapp/index.html

https://developer.android.com/training/index.html  
http://www.codeproject.com/Articles/628894/Learn-How-to-Develop-Android-Application

http://en.wikipedia.org/wiki/Android\_software\_development

http://www.tutorialspoint.com/android/

**APPENDICES**

APPENDIX-A

ANDROID

**Android** is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphonesand tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touchscreen input, it also has been used in game consoles, digital cameras, regular PCs (e.g. the HP Slate 21) and other electronics.

As of July 2013, the Google Play store has had over one million Android applications ("apps") published, and over 50 billion applications downloaded.[[11]](http://en.wikipedia.org/wiki/Android_%28operating_system%29#cite_note-phonearena1-12) A developer survey conducted in April–May 2013 found that 71% of mobile developers develop for Android.[[12]](http://en.wikipedia.org/wiki/Android_%28operating_system%29#cite_note-visionmobile1-13) At Google I/O 2014, the company revealed that there were over one billion active monthly Android users, up from 538 million in June 2013.[[13]](http://en.wikipedia.org/wiki/Android_%28operating_system%29#cite_note-14) As of 2015, Android has the largest installed base of all general-purpose operating systems.

Appendix-B

Eclipse tool

In computer programming, **Eclipse** is an integrated development environment (IDE). It contains a base workspace and an extensible [plug-in](http://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog, Python, [R](http://en.wikipedia.org/wiki/R_(programming_language)), Ruby(including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge.[[2]](http://en.wikipedia.org/wiki/Eclipse_%28software%29#cite_note-VisualAge-2) The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.