Android Mobile Application Build on Eclipse

Garima Pandey*, Diksha Dani**

* Department of Computer Science & Engg, Mahamaya Technical University, Noida
** Department of Computer Science & Engg,, Inderprastha Engineering College, Ghaziabad
India

Abstract- Android is now the most used mobile operating system in the world. Android now has more users, more phones and more tablets worldwide than any other mobile operating system. The Google Play app store has been growing at breakneck speed and with almost as many apps as the Apple app store. This, for entrepreneurs and developers, is the chance of a lifetime to make even more money and reach an even broader audience base.
This paper gives a complete knowledge of how to start working on eclipse and develop an application and get it run on emulator.

Index Terms- Android SDK, ADT plug-in, AVD manager, Eclipse-IDE, java/c++, Android Apps Development.

I. INTRODUCTION

Android is a Linux-based, **open source** mobile operating system developed by Open Handset Alliance led by Google to develop apps for Android devices. To start with we use a set of tools that are included in the Android SDK. Once we have downloaded and installed the SDK, we can access these tools right from our Eclipse IDE, through the ADT plug-in, or from the command line. Developing with Eclipse is the preferred method because it can directly invoke the tools that we need while developing applications..

The basic steps for developing applications are shown in Figure 1. The development steps encompass four development phases, which include:

- Setup: During this phase we install and set up our development environment. We also create Android Virtual Devices (AVDs) and connect hardware devices, on which we can install our applications.
- Development: During this phase we set up and develop our Android project, which contains all of the source code and resource files for our application.
- Debugging and Testing: During this phase we build our project into a debug gable .apk package that we can install and run on the emulator.
- Publishing: During this phase we configure and build our application for release and distribute our application to users.

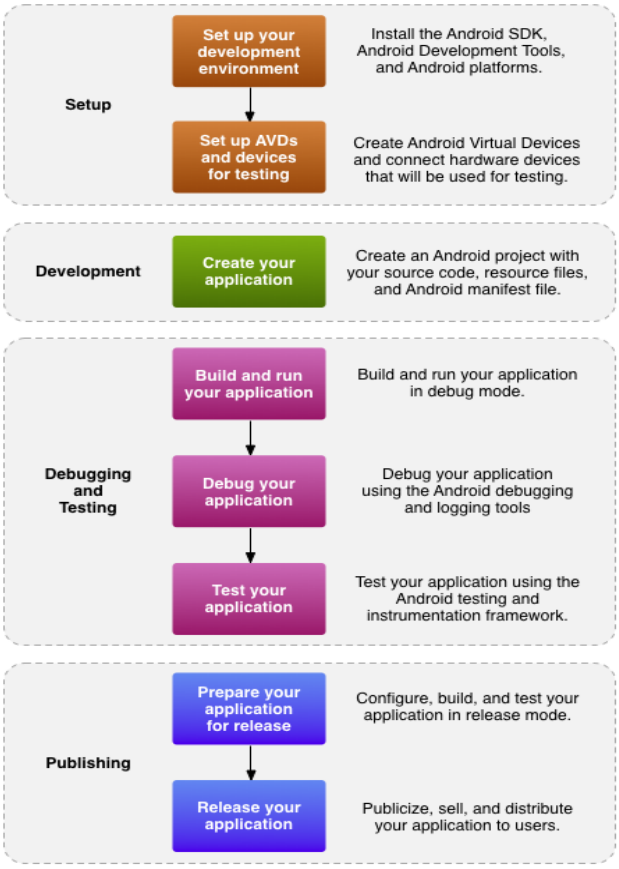

Figure 1. Steps for Application Development

This paper is distributed in following sections:

1) Abstract
2) Introduction
3) Eclipse
4) Tools and environment
5) Development of an app
6) Conclusions

II. ECLIPSE

Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications in Java.
The initial codebase originated from IBM. 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.
Released under the terms of the Eclipse Public License, Eclipse SDK is free and open source software (Table 1).

Codename	Date	Platform version	Projects
N/A	June 2004	3.0[14]	
N/A	June 2005	3.1	
Luna	June 2014 (planned)	4.4	Luna projects
Kepler	June 2013	4.3	Kepler projects
Juno	June 2012	4.2[15]	Juno projects
Indigo	June 2011	3.7[Notes 1]	Indigo projects
Helios	June 2010	3.6	Helios projects
Ganymede	June 2008	3.4	Ganymede projects
Galileo	June 2009	3.5	Galileo projects
Europa	June 2007	3.3	Europa projects
Callisto	June 2006	3.2	Callisto projects

Table 1. Eclipse Releases

III. TOOLS AND ENVIRONMENT

Here we will discuss installation details for our software.

a. Android SDK Tools, revision 20 or newer.

b. SDK Platform Android 3.0 (API 11).
The minimal platform supported by Java API is Android 2.2 (API 8).But for successful compilation the target platform should be set to Android 3.0 (API 11) or higher. It will not prevent them from running on Android 2.2.
Figure 2 gives the look of the window, where we have to select our required android settings.

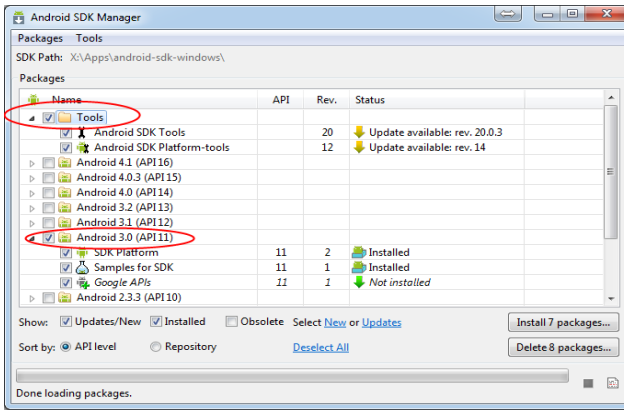

Figure 2. Android SDK Manager

c. Eclipse IDE
There is a list of Eclipse versions that are compatible with the Android SDK. In this paper we are using Eclipse 3.7 (Indigo).

d. ADT plug-in for Eclipse
Android Development Tools (ADT) is a plug-in for the Eclipse IDE that is designed to give us a powerful, integrated environment in which to build Android applications.
ADT extends the capabilities of Eclipse to let us quickly set up new Android projects, create an application UI, add packages based on the Android Framework API, debug applications using the Android SDK tools, and even export signed (or unsigned) .apk files in order to distribute the application.
Developing in Eclipse with ADT is highly recommended and is the fastest way to get started. With the guided project setup it provides, as well as tools integration, custom XML editors, and debug output pane, ADT gives us an incredible boost in developing Android applications.
Following steps are used to download and install the ADT plug-in:

- Start Eclipse, then select Help ‣ Install New Software.
- Click Add (in the top-right corner).

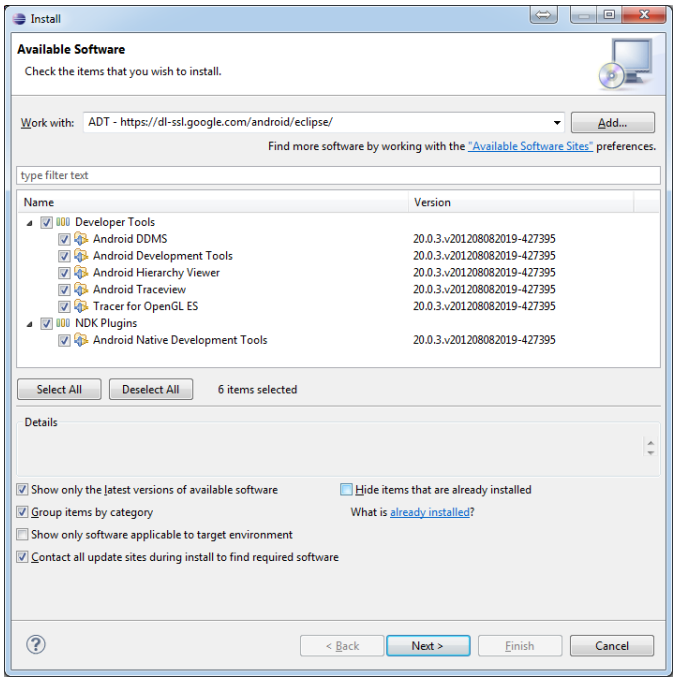

Figure 3. ADT-Plugin/Tool Installation

.

- In the Add Repository dialog that appears, enter “ADT Plug-in” for the Name and the URL refer to Figure 3.
- Click OK (System must be connected to internet).

- In the Available Software dialog, select the checkbox next to Developer Tools and click Next.
- In the next window, we'll see a list of the tools to be downloaded. Click Next.
- Read and accept the license agreements, then click Finish.
- When the installation completes, restart Eclipse.

e. AVD Manager

The AVD Manager provides a graphical user interface in which we can create and manage Android Virtual Devices (AVDs), which are required by the Android Emulator.

- For emulation, we need to define a device.
- Select Window -> Android AVD Manager from the menu (Figure 4).

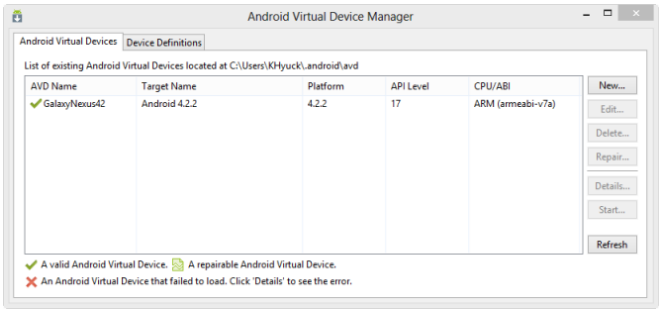

Figure 4. AVD Manager

f. Importing Project

- Open the import Dialog
- Select File > Import ... to open the import dialog.
- Import the "MyProject" project
- In the import dialog, expand the General node and select Existing Projects into Workspace, then click Next to move to the Import Projects step. Make sure that Select root directory is selected, then click the Browse... button.
- In the Browse for Folder dialog, locate the "MyProject" folder, select it and click OK. Then, click Finish to import the project. The project now shows up in the Package Explorer.
- Launch the "MyProject" project
- Right click the "MyProject" in the Package Explorer window, and then select Run As > Android Application from the menu.

g. Deleting Project

Here is the project-wise solution. Right click the "MyProject" in the Package Explorer window, and then select Delete from the menu. In the dialog that appears, ensure that delete project contents on disk is not selected if we want to use the project's folders in workspace. If not, we can check it before be click OK.

IV. DEVELOPMENT OF AN APP

This app will start with a homepage, here we will setup background image and set a mp3 sound and welcome text. Then an automatic intent will fire and app will go on to next page, on 2nd page we will set a counter which will contain 3 buttons: ADD , SUB , EXIT.

Follow these simple 10 steps carefully.

1. Go to file>new>android application project.
2. Open project go to res>layout>activity_main.xml
   Here we can start layout design:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="horizontal"
   android:layout_width="fill_parent"
    android:layout_height="fill_parent">

   <Button
      android:id="@+id/badd"
      android:layout_width="wrap_content"
      android:layout_height="78dp"
      android:text="ADD"
      android:layout_gravity="center"/>

   <Button
      android:id="@+id/bsub"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="SUB"
      android:layout_gravity="center" />

      <TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="YOUR TOTAL IS 0"
      android:gravity="center"
      android:id="@+id/tvDisplay"
      android:textSize="25sp"
     >
      </TextView>
</LinearLayout>
```

3. For creating action to these buttons, in project Go to src>app.com>MainActivity.java

```
package myapp.com;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends Activity {

int counter;
```

```
Button add,sub;
TextView display;

@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);
   counter = 0;
   add = (Button) findViewById(R.id.badd);
   sub = (Button) findViewById(R.id.bsub);
   display = (TextView) findViewById(R.id.tvDisplay);
   add.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
  // TODO Auto-generated method stub
      counter++;
      display.setText("your total is " + counter);
}
});

sub.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
   // TODO Auto-generated method stub
   counter--;
   display.setText("your total is " + counter);
}
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
   // Inflate the menu; this adds items to the action bar if it is
present.
   getMenuInflater().inflate(R.menu.main, menu);
   return true;
}
}
```

4. On desktop copy one sound file with mp3 extension and one image for setting background (rename it in small letter, example aaa.jpg,splash.mp3).
5. Right click on res>layout>new>folder.
   Save mp3 file here.
6. Click res>drawable-hdpi.
   Save image file here.
7. Right click on layout>new>others>example.xml
   Here we create new xml file for our welcome page and we will set our background image.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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/splash_backgroung">
   </LinearLayout>
```

8. Here we will create a new class for the new xml page
   here we will use mp3 song and set an intent for 5 seconds. This will cause the page automatic exit after 5 seconds
   Click on Src>app.com(right click)>new>class

```
package myapp.com;
import android.app.Activity;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Bundle;

public class Splash extends Activity {
MediaPlayer oursong;
@Override
protected void onCreate(Bundle savedInstanceState) {
// TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.splash);
oursong = MediaPlayer.create(Splash.this,R.raw.splashsound);
oursong.start();
Thread timer = new Thread(){
public void run(){
try{
sleep(5000);
}catch(InterruptedException e){
e.printStackTrace();
}finally{
Intent n = new Intent("myapp.com.MainActivity");
startActivity(n);
}
}
};
timer.start();
}
@Override
protected void onPause() {
// TODO Auto-generated method stub
super.onPause();
oursong.release();
}
}
```

9. Set Android Manifest file
   Click AndroidManifest.xml
   Copy Activity area and set one action class as LAUNCHER and another class as DEFAULT.

```
<activity
         android:name="myapp.com.Splash"
         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="myapp.com.MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
        <action android:name="myapp.com.MainActivity" />
         <category
android:name="android.intent.category.DEFAULT" />
         </intent-filter>
      </activity>
```

10. Right click on project>Run As>1 Android Application

The above steps will help us in System Environment, Developing Application and get it run on emulator.
These are all basic steps of android programming. Here main work is carried out in following sections:

- xml layout
- java class
- string class
- manifest files

Android apps have reached a new level of construction where it is working on the concept of Apple's SIRI application, voice command apps, home security etc.
In this paper basic structure for building an android app is covered.

V. CONCLUSION

Our objective behind this paper presentation was to discuss all basic details to start android application and to overcome the technical jargons which come as a big constraint on the way of beginner programmer.

Simplicity was the major factor in explaining all installation process of eclipse and a simple android application, which will give a bust to all aspiring android developers.

ACKNOWLEDGMENT

I express my gratitude to my guide Professor Diksha Dani, department of computer science and engineering, IPEC Ghaziabad, for giving the opportunity and facilities to carry out this application development. I must also express my sincere thanks to Mr. Rohit Pandey, STMicroelectronics, Greater Noida, for his invaluable guidance and support that have added a great deal to substance of this paper.

REFERENCES

[1] Shyam Bhati, Sandeep Sharma, Karan Singh "*Review On Google Android a Mobile Platform*" IOSR Journal of Computer Engineering(IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727Volume 10, Issue 5 (Mar. - Apr. 2013), PP 21-25

[2] Khawlah A. AI-Rayes, Aise Zulal Sevkli, Hebah F. AI-Moaiqel, Haifa M. AI-Ajlan, Khawlah M. AI-Salem, Norah I. AI-Fantoukh "*A Mobile Tourist Guide for Trip Planning*" IEEE MULTIDISCIPLINARY ENGINEERING EDUCATION MAGAZINE, VOL. 6, NO. 4, DECEMBER 2011

[3] Priyanka Shah, Ruta Gadgil, Neha Tamhankar "*Location Based Reminder Using GPS For Mobile (Android)*" ARPN Journal of science and Technology ©2011-2012. VOL. 2, NO. 4, May 2012

[4] Sumaiya Patel, Darshana Thakur, Sujit Sekhar. Priyanks Dhamane "Lockme - Android Security Application" IJCER, VOL. 3, Issue. 3

[5] http://developer.android.com

AUTHORS

First Author – Garima Pandey , M. Tech, Mahamaya Technical University, Noida, garima.pandey@gmail.com.
Second Author – Diksha Dani, M. Tech, Inderprastha Engineering College, Ghaziabad.