



Received:10.12.2016 Accepted:23.01.2017

Developing of Android Mobile Application Using Java and Eclipse: An Application

Şenay KOCAKOYUN¹

Abstract - In this work, an article has prepared which can be a guide to researchers who want to start developing Android-based applications. Android is known as an operating system that comes face to face with more than one million applications. While Android sets the ground for different applications, people are aiming for unlimited entertainment for their lives and learning while they are having fun. Mobile application development steps are described for those who want to develop a mobile app for their own business, blog, service or product, but have limited resources for it. The installation steps of the necessary software are mentioned. Adroid SDK and Eclipse ADT installation, Android SDK directory settings, Android SDK and AVD manager settings, It provides detailed information on setting up a virtual device for Android, the hierarchical view and functions of Android project files, creating an Android project from scratch and developing the application. Designs for building NEU-CEIT Android application are written in detail along with the structures in xml and java files. This work will guide the researchers who want to develop applications in the Android Eclipce system environment, without developing the application and running in the emulator.

Keywords: Android SDK, AVD manager, Eclipse-IDE, java/c++, Android Apps Development

1.Introduction

Android, one of the mobile market leaders, has to a billion apps on the Google Play Store [1]. It's a Linux-based, open source mobile operating system developed by the Open Handset Alliance, which is managed by Google to develop applications for Android, Linux devices [2]. The day-to-day Android operating system in the electronics market has become increasingly popular, especially in the smartphone market. It is open source and preferred by researchers due to the fact that some of the development tools are free [3]. This has been a great source of inspiration for using the Android system for researchers. In addition, it allows developers to use a very suitable hardware platform for less effort to realize their ideas. That's why the number of researchers working on it has improved the further development of Android [4-5]. Along with these developments, hardware and application forms have seriously changed the way education, learning, information access and information are presented. Although most of these hardware and applications have not been developed to provide educational-instructional support, education and training environments have contributed to their own purposes according to each new technology and have made use of them as much as possible [6].

Mobile devices such as laptops, personal digital assistants and mobile phones have become a learning tool with great potential both in classrooms and outdoors [7-8]. In this context, the use of mobile devices in learning environments can be seen as a factor that can increase the efficiency in the learning-teaching process. In this process, mobile learning environments are expected to be developed in order to successfully achieve the educational integration of mobile learning technologies and mobile applications [9]. However, the development of mobile application software is weak and the methodologies deemed relevant to the progress of such mobile applications are not sufficient. There is still a shortage of research methods and at the same time lack of understanding and analysis of the concerns and difficulties that may arise in the mobile application development process [10]. There are steps to be followed 'Anadolu BIL Vocational School of Higher Education, Computer Programming Department, Istanbul Aydın University, İstanbul, Turkey senaykocakoyun@aydin.edu.tr

when developing applications. The basic steps for developing applications are shown in Figure 1. the development steps include four developmental stages (Figure 1).

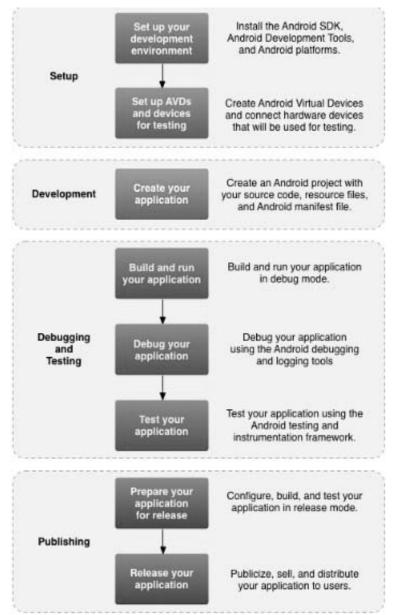


Figure 1. Steps for Application Development [2]

- 1. **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.
- 2. **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.
- 3. **Debugging and Testing:** During this phase you build your project into a debuggable.apk package(s) that you can install and run on the emulator or an Android-powered device.
- 4. **Publishing:** During this phase we configure and build our application for release and distribute our application to users.

Java, Eclipse platform, Android ADT and Android SDK are available from the most basic programming languages for mobile application development on Android operating system. These systems have a nice interface and smooth operation. These applications provide a great user experience [11]. In addition to being popular with the Android operating system, applications based on the Android SDK are also attracting more attention.

In this work, a number of tools were used in Eclipse when developing applications for Android. First, a number of tools were used in the Android SDK. After downloading and installing the SDK, these tools are accessed directly from the Eclipse IDE, through the ADT plug-in, or from the command line. Developing with Eclipse is the preferred method because we can directly search for the tools we need while developing applications [2]. Now let's talk about the installation of the Android application development tools.

2. Installation of Android Application Development Tools

Here are the details of the installation of the software.

- Java Development Kit (JDK): Java Software Development Tools
- Android SDK: Android Software Development Tools
- Eclipse SDK: Advanced Software Development Interface

The first installation you need is the Java SE Development Kit (JDK). JRE which is the regular Java version installed on computers is only used to run Java applications. JDK allows to develop applications in Java language.

The SE downloaded Java Development Kit (JDK) can be from http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html. The next most common tool in Java programming that needs to be installed is the installation of the Adroid SDK (Software Development Kid). The Android SDK is a development tool that provides tools and APIs for developing applications on the Android platform with the Java programming language [12-13]. In order to download Android SDK tools, installation files must be downloaded from http://developer.android.com/sdk/index.html. The program can be downloaded from the Eclipse IDE for Java Developers at http://www.eclipse.org/downloads/. In this page you need to choose the operating system that suits you. Eclipse can be installed to the desired location. Now let's look at the Eclipse ADT installation.

2.1. Eclipse ADT Setup

Eclipse is an integrated development environment (IDE). It includes a core workspace and an extensible plugin system for customization of the environment. The Eclipse software development kit (SDK), which contains Java development tools, is for Java developers [14]. Users can extend their capabilities by writing extensions, such as development toolsets for other programming languages written for the Eclipse Platform, and they can create and contribute to their own applications [14-15]. The Eclipse SDK is free and open source software. After opening the Eclipse program for installation, click *Help> Install New Software* (Figure 2).

Work with:	http://dl-ssl.google.com/andr	oid/eclipse/	-	Add
		Find more software by working with th	e <u>'Available Software Si</u>	tes' preferences.
type filter te	ot			
Name		Version		
	Developer Tools Android DDMS Android Development Tools Android Hierarchy Viewer	8.0.1.v201012062107-82219 8.0.1.v201012062107-82219 8.0.1.v201012062107-82219		

Figure 2. Eclipse ADT setup

Work with: This field will be http://dlssl.google.com/android/eclipse/. The installation will take a long time after all attachments have been selected. After installation, the Eclipse program needs to be restarted.

2.2. Android SDK Directory Settings

After selecting *Window> Preferences on* the Eclipse SDK, the "*Android*" tab is selected from the left menu of the window. The location of the installed Android SDK should then be shown (Figure 3).

Preferences				
type filter text	Android		<⊳ ▼ ○	÷ • •
 General Android 	Android Prefere	nces		
⊳ Ant	SDK Location:	C:\android\android-sdk-windows	Bro	wse
 Help Install/Update 	Note: The list of	SDK Targets below is only reloaded on	ce you hit 'App	y' or '0
Java	Target Name	Vendor	Platform	API
Plug-in Development	Android 1.5	Android Open Source Project	1.5	3
Run/Debug	Google APIs	Google Inc.	1.5	3
Team	Android 1.6	Android Open Source Project	1.6	4
XML	C 1 ADI	C 11	10	

Figure 3. Android SDK Directory settings

2.3. Android SDK and AVD Manager Settings

After completing the directory settings, click *Window> Android SDK and AVD Manager*. Afterwards, the package installation process for Android should be done (Figure 4).

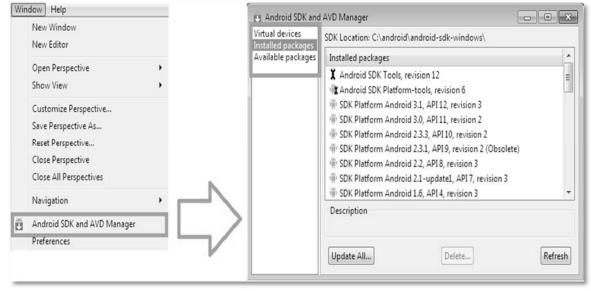


Figure 4. Android SDK and AVD Manager settings

2.4. Virtual Device Setup for Android

AVD Manager provides a graphical user interface where we can create and manage Android Virtual Devices (AVD) as required by the Android Emulator. You can configure the virtual device from "Virtual Devices" in the Window> Android SDK and AVD Manager screen. In this section, you will need to specify the properties of your own phone (Figure 5).

Android SDK and	AVD Manager				-	- 0 22	Create ne	ew Android Virtual I	Device (AVD)		×
Virtual devices Installed packages	List of existing	g Android Virtual De	vices located	al			Name	AndroidDevice2-3	1.3		
Available packages	AVD Name	Target Name	Platfo A	PIL	CPU/ABI	New	Target:	Android 2.3.3 - AP			
		No AVD available				Edit	Targett	Android 23.3 - AP	1 Level 10		•
							CPU/ABE	ARM (armeabi)			· · · ·
							SD Card:				
								Size:			MiB 💌
						Details	1	© File:			Browse
						Start					
						- Startes	Snapshot:	Enabled			
						Refresh		E chaolea			
	✓ A valid Ar	ndroid Virtual Device	e. 🔛 A repai	rable An	droid Virtua	I Device.	Skin:				
	X An Andro	id Virtual Device that	t failed to loa	d. Click	'Details' to :	see the error.			Default (WVGA	3 00)	•
								C Resolution:		x	
							Hardware:				
								Property		Value	New
								Abstracted LCD	density	240	Delete
								Max VM applicat		24	Dereit
(a.	(E	(2				-1	Device ram size		512	
Problems @ Javad	Icc Bb Declara	ton 🖵 Console i-	2.5								
hdroid								e the existing AVD w			
2011-08-13 18:46:						ased or Android 2.	3.3.	e the ensuing AVD W	with the same i	same	
2011-08-13 18:46:				nor und	ie contra	k≣l					
2011-08-13 18:46:								1			
2011-08-13 18:46:	37 - SDK Mar	nager] hw.ranSiz	a=512							County AVE	Cancel
4									_	Create AVD	Cancel

Figure 5. Virtual Device setup for Android

2.5. Hierarchical View and Functions of Android Project Files

The first time an Android project is created, the hierarchical structure of folders and files will be as follows (Figure 6).

🛱 Package Explorer 😂	
A SS NEU-CEIT	
▲ ﷺ src	
> 🖽 com.bolume.ceit	
p 😕 gen [Generated Java Files]	
> 🛋 Android 4.2.2	
Android Private Libraries	
😂 assets	
> 📴 bin	
b 📴 libs	
a 🛃 res	
b 🗁 drawable	
drawable-hdpi	
🗁 drawable-ldpi	
🗁 drawable-mdpi	
🗁 drawable-xhdpi	
🗁 drawable-xxhdpi	
> 🔂 layout	
> 🗁 menu	
b 🗁 values	
b 🗁 values-sw600dp	
b 🗁 values-sw720dp-land	
b 🗁 values-v11	
> > > values-v14	
AndroidManifest.xml	
ic_launcher-web.png	
d lint.xml	
proguard-project.txt	
project.properties	

Figure 6. Hierarchical view of Android project files

Developing of Android Mobile Application Using Java and Eclipse: An Application

The pages of Android applications are created with the *xml* files in the *layout* folder. The management of these *xml* image files is provided by the Java files in the *src* folder. The description of these files and folders is as follows:

- src Folder: The folder in which the Java packages and files are located.
- **res Folder:** The folder where the views of the application are located.
- **drawable Folders:** The icon of the application and the images to be included in the application are contained in these folders.
- layout Folder: contains the xml files that the application images are in.
- AndroidManifest.xml File: This file contains project and coordination information.

2.6. Create an Android Project

To create an Android project, select "Android Application Project" from the list with File> New> Other (Figure 7).

	1	lew		- 🗆 ×
Select a wizard Create an Android Appli	cation Project			\$
Wizards:				
type filter text				
문 Android Sam J한 Android Test 값 Android XML 값 Android XML 값 Android XML 값 Android XML	lication Project Set ect ect from Existing Code ple Project Project File Layout File			~
	< Back	Next >	Finish	Cancel

Figure 7. Android Project Creation Screen-I

On the next screen, various properties related to the project should be entered (Figure 8). Here are the descriptions of the options:

- Application Name: Name of the application
- **Project Name:** The name of the Project
- **Package Name:** The name of the package in which the application's files will be stored. It is very important that you identify your app on your Google Play account.
- Build SDK: Specifies the version of Android the project will compile.
- Minimum Required SDK: Specifies the lowest version of the Android operating system on which the project will run.

Iew Android Applica Enter an application nation			· ·
i enter an application has	ne (snown in launcher)		a
Application Name:®			
Project Name:@			
Package Name:0			
finimum Required SDK:0	API 8: Android 2.2 (Froyo)	~	
Target SDK:0	API 17: Android 4.2 (Jelly Bean)	~	
Compile With:0	API 18: Android 4.3	~	
Theme:0	Holo Light with Dark Action Bar	~	

Figure 8. Android Project Creation Screen-II

Next will come up with options for the icons of the application when we move on to the next step. Here you can download the icon from an image file or you can create it yourself. Once you have made the necessary adjustments, you can proceed to the next step (Figure 9). At this stage, information about the design of the application will be requested. If you have an application targeting tablets, the *"Master Detail Flow"* type should be selected. This way, the normal application is displayed on the smartphone while a navigation menu is displayed on the tablet devices on the left.

Ð	Ne	w Android Ap	plication		-	• ×
Create Activity Select whether to cre	ate an activity, and if so,	what kind of activ	ity.			0
Create Activity						
Blank Activity						
Fullscreen Activity Master/Detail Flow						
waster/Detail Flow						
				(mm	:	
Blank Activity						
	activity, with an action ba	wand ontional na	vigational elemen	tr cuch as tabs or	horizon	tal cuine
creates a new blank	activity, with an action be	and optional ne	vigational elemen	its such as tabs of	10112011	tai swipe.
2		< Back	Next >	Finish	0	ancel

Figure 9. Android Project Creation Screen-III

Detailed information such as "Activity Name", "Layout Name", "Navigation Type", "Hierarchical Parent", "Title" will be asked for the screen to be created on the next screen. Once the required information has been filled in, the project will be displayed on the left menu of Eclipse after clicking "Finish" (Figure 10).

🛱 Package Explorer 🛛 📄 🛱 🎽 🗖	这 activity_main.xml 🛛 MainActivityjava 🕴	 E Outline 🕄	- 0
 ¹ HelloWorld ¹ ¹ erc ¹ ¹ ¹ cremexample.belloworld ¹ ¹ ¹ ¹ ¹ ¹ ¹	<pre>package com.example.helloworld; # import android.cs.Bundle;] public class MainActivity extends Activity {</pre>	 □ ↓³₂ ≷ ⊕ com.example. ▲ ⊕ MainActivity ▲ onCreate 	ار کی ہے کہ اسلام کی br>الایک الایک الایک کی ا

Figure 10. Android Project Creation Screen-IV

Right-click on the project and select Run As> 1 Android Application option (Figure 11).

Run As 🕨 🕨	I Android Application
Debug As Profile As Validate Team Compare With	J ^① 2 Android JUnit Test 図 3 Java Applet て第X A 図 4 Java Application て第X J Ju 5 JUnit Test て第X T
Restore from Local History Android Tools	Run Configurations
Properties #1 Resource Configurations	pi

Figure 11. Android Project Creation Screen-V

Restart Eclipse when installation is complete. After this point, a virtual device suitable for running the application is running and the application is started.

3. Development of an Application

I

To create NEU-CEIT Android application, the following designs and coding are done.

- 1. Select *File> New*
- 2. Go to the Android Application project.
- You can start designing by opening the res> layout> main.xml page from the open folders. In the Eclipse IDE environment, the directory structure and contents of your project are displayed in the Package Explorer on the left side (Figure 12).

	 电 ③ * [99 ⑤ 約 *] 2 * 2 * 2 * 5 6 * 3 * 1 4	Quick Access	9 000
Package Explorer 😒 📃 😂 🔤	 main.aml 😫 🗟 NEU-CEIT Manifest 🛛 🗵 Main.java		
 ret dravable-hőpi dravable-hőpi dravable-hőpi dravable-hőpi dravable-nöhöji dravable-nöhöji dravable-nöhöji skademikalismalarami akademikalismalarami botekami grasilami grasilami mainekami mainekami mainekami sensyittimami sensyittimami sensyittimami takomami takomami 	<pre>GREATISTICLESULT: ADJUST /pre>		*

Figure 12. NEU-CEIT res>layout> main.xml page

res/layout directory: An *.xml* file is created that determines the view for each activity (window) in the program. All of these files are in this directory. *Res> layout* directory, the *main.xml* file defines the elements that determine the appearance of the main activity in the program. The code snippets in the *main.xml* file are given below:

//main.xml
<pre></pre> <pre></pre> <pre></pre> <pre></pre> <pre></pre> <pre>//main.xmi</pre> <pre></pre>
xmlns:tools="http://schemas.android.com/tools"
android:layout width="wrap content"
android:layout_widih="wap_content" android:layout_height="match_parent"
android:background="@drawable/banneriki"
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=".Main"> <button< th=""></button<>
android:id="@+id/button2"
style="?android:attr/buttonStyleSmall"
android:layout_width="match_parent"
android:layout_height="50dp"
android:layout_alignParentLeft="true"
android:layout_below="@+id/button1"
android:background="@drawable/button2"
android:text="BÖTE"
android:textAlignment="center"
android:textColor="#7F7F7F"
android:textSize="12dp" />
<button< th=""></button<>
android:id="@+id/button3"
style="?android:attr/buttonStyleSmall"
android:layout_width="match_parent"
android:layout_height="50dp"
android:layout_alignParentLeft="true"
android:layout_below="@+id/button2"



In the Eclipse IDE environment, if you click on the *main.xml* file under the *res> layout* directory in the Package Explorer on the left side of the screen below, the contents of the text of the file are displayed on the right side. If you click on the *Graphical Layout* tab shown on the right side, the activity is graphically displayed on the screen (Figure 13).

INTERNATIONAL JOURNAL OF ELECTRONICS, MECHANICAL AND MECHATRONICS ENGINEERING Vol.7 Num.1 - 2017 (1335-1354)

Şenay KOCAKOYUN

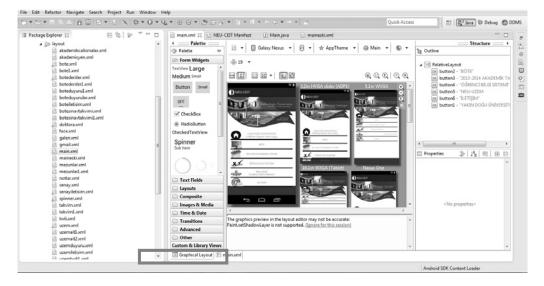


Figure 13. Main.xml graphical layout screenshot

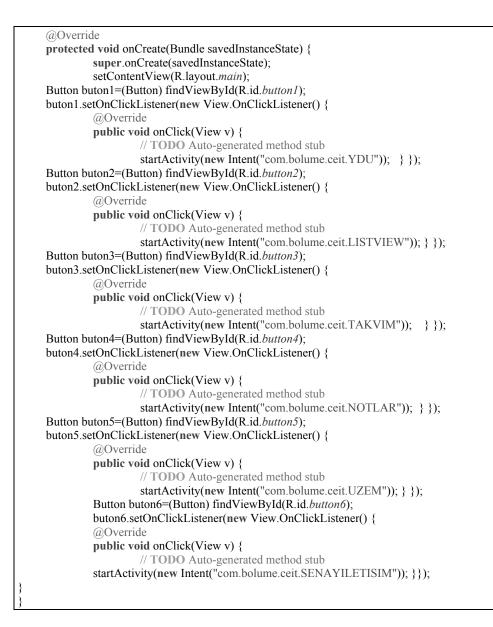
4. The project *src* > *com.bolume.ceit* > *main.java* page has been worked on to create an action for the buttons for the prepared buttons (Figure 14).

ckage Explorer 😫 📄 🕾	🗸 🗖 🔄 main.aml 🔄 NEU-CEIT Manifest 🕖 Main.java 😫	
3 NEU-CEIT	A package controlune.ceit;	
4 29 src	· package interesting;	
a 唐 com.bolume.ceit	<pre># import android.os.Bundle;[]</pre>	
 J akademikcalismalarjava 		
 [J] akademikcalismalargava [J] akademisyen.java 		
 J akademisyenjava J bote.java 	public class Main extends Activity {	
 J boteljava J botedersler, java 	Bovernide	
 J botedersier.java J boteduyurular.java 	protected void onCreate(Bundle savedInstanceState) {	
 J boteobyuruargava J boteiletisim.java 	super.onCreate(savedInstanceState);	
	<pre>setContentView(R.layout.moin);</pre>	1
 Dotesinavtakvimi.java 		
 J doktora.java 	<pre>Button buton1=(Button) findViewById(R.id.button1); buton1.setOnClickListener(new View.OnClickListener() {</pre>	
I facejava	anony reconcrete free Arenover research f	
 J gmeiljeve 	e governide	
 Istviewjava 	public void onClick(View v) {	
J Main.java	// TODO Auto-generated method stub	
 I mezunlar.java 	<pre>startActivity(new Intent("com.bolume.ceit.YDU"));</pre>	-
I notlar.java)); ⁷	
J senayjava	177	
I senayiletisim.java	<pre>Button buton2=(Button) findViewById(R.id.button2);</pre>	
J spinner.java	buton2.setOnClickListener(new View.OnClickListener() {	
 J tekvim.jeva 		
 I twitjava 	BOverride public void onClick(View v) (
 J uzem.java 	// TODO Auto-generated method stub	
J uzemaitl.java	<pre>startActivity(new Intent(*com.bolume.ceit.LISTVIEW"));</pre>	
J uzemait2.java	}	
b II uzemduyuru.java	>>>	
J uzemiletisim.java		
D uzemturki.java	Button buton3=(Button) findViewById(R.id.button3); buton3.setOnClickListener(new View.OnClickListener() {	
 J Veritabani.java 	butons.secontractistener (new view.oncraticitistener () {	
I veritabaniekrani.java	BOvernide	
J wordpress.java	public void onClick(View v) {	
 J ydu.java 	// TODO Auto-generated method stub	-
. In underabligane inca	• (3

Figure 14. NEU-CEIT src>com.bolume.ceit>main.java page

src directory: All *.java* source files created for the Android program will be in this directory. The *main.java* file is the source file with the *.java* extension of the main activity (window) created for the Android program. The contents of this file are shown below:

//Main.java
package com.bolume.ceit;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
public class Main extends Activity {



5. Click res> drawable-hdpi, all image files used in the project are saved here (Figure 15).

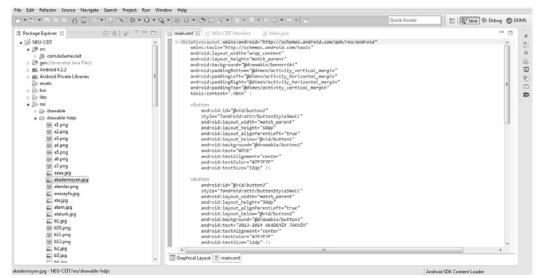


Figure 15. NEU-CEIT res>drawable-hdpi Picture files

res directory: Contains the resources required for the program. An Android application includes the program source code, as well as images and sound files, and anything that relates to the visual appearance of the app. Sources are defined separately from the program code. At this point, visual characteristics of the program can be changed without changing the program code.

 Within the program NEU-CEIT *Manifest.xml* file; the minimum SDK version required to run the program, the SDK version the program was created in, and a single activity definition between the application tags (Figure 16).

File Edit Refactor Source Navigate	e Search Project Run Wir	ndow Help				
ロ・ロ・ヨ トロ [2] 0 [6] 日 ロ ・ (5] 10 + 0 + 0 + 0 + 0 + 0 + 0 + 10 + 10 +						
18 Package Explorer 33	888 - 0	🗟 main.xml	🗟 NEU-CEIT Manifest 🗱 🕢 Main,java 😒 maineski.aml		° 0	
Sign NDL-CET Sign NDL-CET Sign NDL-CET Sign NDL-CET Sign NDL-CET Sign Adversarial (see TRee] Sign Adversarial (see TRee] Sign Adversarial (see TRee] Sign Adversarial (see TRee) Sign			<pre>ruture::::::::::::::::::::::::::::::::::</pre>			
				14	Android SDK Content Loader	

Figure 16. NEU-CEIT Manifest.xml page

The *AndroidManifest.xml* file is clicked to set up the Android Manifest file. The Copy Activity field and the Action class are set to LAUNCHER and another class is set to DEFAULT. A DEFAULT must be created for each class. 30 Manifest files have been created for this application. As an example, a few examples of activity are given below:

```
//NEU-CEIT Manifest.java
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.bolume.ceit"
  android:versionCode="1"
  android:versionName="1.0">
  <uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="17" />
  <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=".Main"
      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=".uzem"
      android:label="@string/app name">
      <intent-filter>
         <action android:name="com.bolume.ceit.UZEM" />
         <category android:name="android.intent.category.DEFAULT" />
      </intent-filter>
    </activity>
     <activity
      android:name=".listview"
      android:label="@string/app name">
      <intent-filter>
         <action android:name="com.bolume.ceit.LISTVIEW" />
         <category android:name="android.intent.category.DEFAULT" />
      </intent-filter>
    </activity>
     <activity
      android:name=".akademikcalismalar"
      android:label="@string/app_name">
      <intent-filter>
         <action android:name="com.bolume.ceit.AKADAMIKCALISMALAR" />
         <category android:name="android.intent.category.DEFAULT" />
      </intent-filter>
    </activity>
     <activity
      android:name=".doktora"
      android:label="@string/app name">
      <intent-filter>
         <action android:name="com.bolume.ceit.DOKTORA" />
         <category android:name="android.intent.category.DEFAULT" />
      </intent-filter>
    </activity>
```

Şenay KOCAKOYUN



7. To make a database connection; *src> com.bolume.ceit>* the code that needs to be written into the *database.java* file is given below:

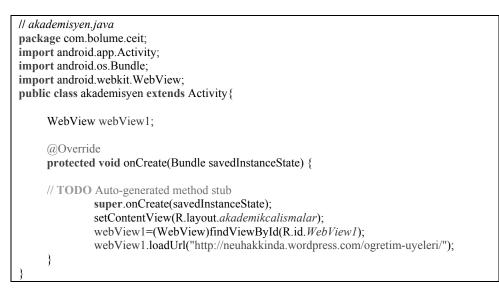
//veritaba	ni.java
import ar import ar import ar public cla priv priv	com.bolume.ceit; adroid.content.Context; adroid.database.sqlite.SQLiteDatabase; adroid.database.sqlite.SQLiteOpenHelper; ass Veritabani extends SQLiteOpenHelper { rate static final String VERITABANI ="ogrenciler.db"; rate static final int SURUM = 1; lic Veritabani(Context con) { super(con,VERITABANI,null,SURUM); }
pub db.e	<pre>verride lic void onCreate(SQLiteDatabase db) { // TODO Auto-generated method stub xecSQL("CREATE TABLE ogrenciisim(id INTEGER PRIMARY KEY CREMENT, isim TEXT, soyad TEXT);"); }</pre>
\bigcirc	<pre>verride lic void <u>onUpgrade</u>(SQLiteDatabase db, int oldVersion, int newVersion) { // TODO Auto-generated method stub db.execSQL("DROP TABLE IF EXIST ogrenciisim"); onCreate(db);</pre>

- 8. NEU-CEIT Embedded in *View* for Android application The following codes are required for opening WordPress pages in the application as an example.
 - Academic studies which direct to academic akademikcalismalar.java file:

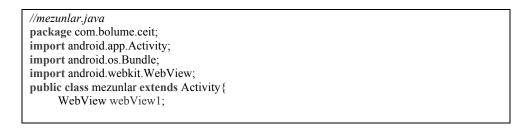
. R. N. S. C. B. B. S. S. K. S. O.	ביום פיום פיום פיום פיום פיום פיום אין	Debug 🌚 D
ackage Explorer 🛛 📄 🕸 🔍 🗖 🗇	🗴 main.xml 🖞 NEU-CEIT Manifest 🕧 Main.java 🖞 maineski.xml 🗍 akademikcalismalar.java 🕸	
ackage Explorer 32 A 10 arc A	<pre>main.xml</pre>	
> (1) springrive > (2) > (2) tottjina > (2) tottjina > (2) ucenjan > (2) ucentitjina > (2) ucentitjina > (2) ucentutjina > (2) ucentutjina > (2) vetitabaničkani jina		

Figure 17. akademikcalismalar.java page

• akademisyen.java file which makes the transition to lecturers and communication page:



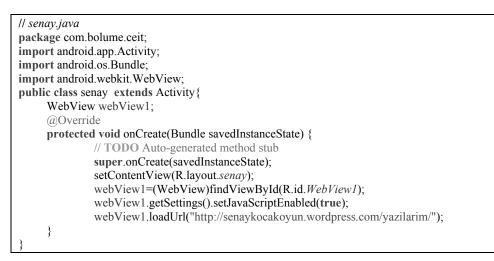
• The *mezunlar.java* file which provides the transition to the page of the students graduated from the CEIT program:



Şenay KOCAKOYUN

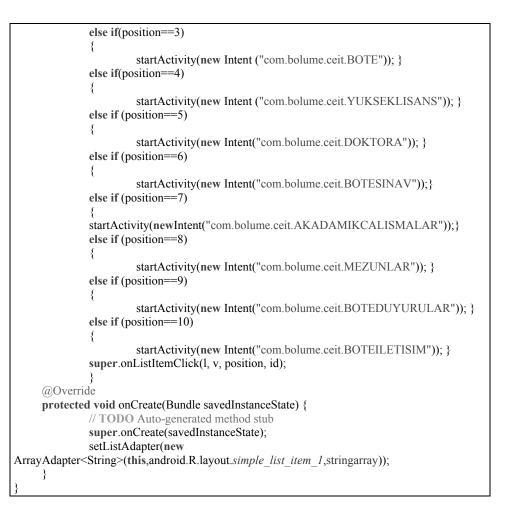
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mezunlar);
        webView1=(WebView)findViewById(R.id.WebView1);
        webView1.loadUrl("http://neuhakkinda.wordpress.com/mezunlar/");
    }
}
```

• senay.java file created to communicate with the applicant prepares:



9. The code snippets that are placed in *listview.java* to make a list view menu are as follows:

//Listview.java
package com.bolume.ceit;
import android.app.ListActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
public class listview extends ListActivity {
String stringarray[]={"Ders Programı","CEIT 340 Eğitim Videoları","Öğretim Üyeleri ve
İletişim", "Lisans Programı", "Yüksek Lisans Programı", "Doktora Programı", "BÖTE Sınav
Takvimi","Akademik Çalışmalar","Mezunlarımız","Duyurular","İletişim"};
@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
// TODO Auto-generated method stub
if(position==0)
{
<pre>startActivity(new Intent("com.bolume.ceit.DERSPROGRAMI"));}</pre>
else if(position==1)
{
<pre>startActivity(new Intent("com.bolume.ceit.WORDPRESS")); }</pre>
else if(position==2)
{
<pre>startActivity(new Intent ("com.bolume.ceit.AKADEMISYEN")); }</pre>



10. Right click on project>Run As>1 Android Application (Figure 18).

1 • E •	New			0.00	4 . 1	医骨骨 医骨骨	• * • • • • •	* *	Quick Access	1 B	Java 🌣 Debu	0	DDMS
I Package	Go Into	Go Into		stview.java [) uzemjeve i	[]] idemaitLjava	(]) botedersler	[]] skademisyen	🕑 doktorajava 🖂	(I) sensyjaca	»,	- 0	1
· 通 NEU · 四 · 四 · 四 · 二 · 二		New Window rpe Hierarchy	F4 Alt+Shift+W + Ctrl+C	package com.bolume.ceit; # import android.app.Activity;[] public class doktora extends Activity(0. 0 E
NE SALAN AND AND AND AND AND AND AND AND AND A	Delete		Ctrl+V Belete Ctrl+Alt+Shift+Dexm Alt+Shift+S Alt+Shift+T+	ii @Overrid protecte // 1 supe setC	<pre>webles; potentide potentide // TOD Autogenerated method itsu isgor.ok/rate(seveCinitenceSiste) { istContention(.isgor.isdorEcol(ission); webles:(_iebvies)find/isbyId(R.id.iebvics); webles:(_iebvies)find/isbyId(R.id.iebvics); } }</pre>								0,11
		oject srelated Projects Vorking Sets	15										
	Run As Debug A Validate Team Compan Restore f Android	e With from Local History	•	 1 Android Ap 2 Android JU 3 Java Applet 4 Java Applic Ju S JUnit Test Run Configu 	nit Test	Alt=Shift=X, A Alt=Shift=X, J Alt=Shift=X, T							
	Configur			4									
U-CEIT	Propertie	es	Alt-Enter										

The above steps will help researchers who want to develop applications in the Eclipse system environment, without developing the application and running it in the emulator. All of these are the basic steps of Android programming.

Xml layout, java class, string class, manifest files are developed with application partitions between files. Note that the names of the xml and java files are the same in the study. Each xml page must have a java page. Figure 19 shows screenshots of some pages of the application.

ψ 27 ► Ω 🛛 🕹 \$ 🗑 📶 665 🏦 14:07	학 🖬 27 🍉 🙃 👘 🕸 🧃 14:07	± 27 ► Ω 8 \$ 🖲 "4675∰ 14:21	
NEU-CEIT	NEU-CEIT	NEU-CEIT	NEU-CEIT
Yakin Dogu Universitesi	Ders Programi	Yakin Doğu Üniversitesi	Yakan Doğu Üniversitesi
Very Grow in Aust Descentor	CEIT 340 Eğitim Videoları	Nume Conner (1) Raya Ordernadow	
Julius State	Öğretim Üyeleri ve İletişim		Menû 👻 Ara
YAAN DOĞU ÜNİVERSİTESİ (1988'den Bugüne Yakın Doğu)	Lisans Programi	www.th	Yakın Doğu Üniversitesi Bigisayar ve Öğretim Teknolojileri Öğretmenliği
the second secon	Yüksek Lisans Programı	Not imported 4 800 1007 Abits Index Antenna en el la basera el la b	Öğretim Üyeleri
8012	Doktora Programi		
2013-2014 AKADEMIK TAKM	ošreo	CEIT-340 Videolari	Öğretim Elemanı
	BÖTE Sınav Takvimi	MEB - Adobe Flash Professi You'tube < 0	Yard. Doç. Dr. Fezile ÖZDAMLI
CONDICI BILG SISTEM	Akademik Çalışmalar		fezileozdaminilhotmai.com
NEU-CZEM (Uzaktav Egitan Merkezi)		1	Uz. Ayşegül AKÇAM
	Mezunlarımız		akcamayanou/@umail.com
C ILETIŞÎN	Duyurular		Prof. Dr. Hafize KESER
	İletişim		kenerillankara.edu.te
	neuşim		Prof. Dr. Servet BAYRAM
		(max (abauram Timarman ada k

Figure 19. NEU-CEIT Screen Views

In this article, the basic structure for building an Android application is discussed. It is thought that application will be an example for developers, as different applications take their place in our lives.

5. Conclusion

As interest in mobile applications grows, mobile application development has become a complex process. The software development process of the developed applications is of great importance. In order to control these processes and get the best results, it is necessary to explain the application development stages in detail. The vast majority of studies in this field give the results of the operation of software development processes. The lack of studies describing how these applications are developed is noteworthy. However, it is known that the development of mobile application software is weak and the accepted statements about the progress of mobile applications are not enough. It is necessary to draw attention to the fact that there is still shortage in the research methods and at the same time to determine the concerns and difficulties that may arise in the mobile application development process. Because software development usually brings abstract suggestions, it is necessary to reflect the development process of the software as it is.

In this study, NEU-CEIT, the developmental process of the Android mobile application, was developed for use in the education of students in the Department of Teaching Computer and Instructional Technology at Near East University. While developing the application in the study, the steps to be followed are described and the tools to be loaded are mentioned one by one.

The Eclipse platform, Android ADT and Android SDK can be used to develop mobile applications for the Android operating system. These systems have a nice interface and smooth operation. Java on the Eclipse IDE is now a preferred choice for software development. Researchers of Software Engineering have developed programs for various purposes in this environment and went on to analyze them. These applications provide a great user experience. In addition to being popular with the Android operating system, Android SDK based applications have attracted more attention. In this work, a number of tools were used in Eclipse while developing the application for Android. First, the tools in the Android SDK are used. After downloading and installing the SDK, these tools are accessed directly from the Eclipse IDE, through the ADT plug-in, or from the command line. After installing Eclipse ADT, Android SDK directory settings and Android SDK and AVD manager settings have been made. With virtual device setup for Android, the application is displayed on virtual devices. After describing the hierarchical appearance and functions of the Android project files in detail, an Android project creation phase has been given. The NEU-CEIT Android application has been designed and coded in the development phase. Designed on res> layout> main.xml page. An .xml file has been created that determines the view for each activity in the program. All of these files are located in the res> layout directory. res> layout directory, the elements that determine the appearance of the main activity in the program have been defined. The code snippet inside the *main.xml* file is given. In order to create an action for the prepared buttons, the coding on project src>com.bolume.ceit>main.java page is given in detail.

Any *.java* extension source files created for the Android program will be in the src directory. The *main.java* file is the *.java* extension source file for the main activity created for the Android program. The content of this file is given in terms of being an example to the researcher. All image files used in the project are saved in the drawable-hdpi area. An Android application stores program source code, as well as images and audio files, and anything that is related to the visual appearance of the application in the res directory. Sources are defined separately from the program code. This allows you to change the visual properties of the program without changing the program code.

Within the program NEU-CEIT *Manifest.xml* file; The minimum SDK version required to run the program, the SDK version the program was created for, and a single activity definition between the application tags. The *AndroidManifest.xml* file has been opened to set up the Android Manifest file. In general, the action class is set to LAUNCHER and another class is set to DEFAULT. A DEFAULT must be created for each class. 30 Manifest files have been created for this application. Several examples of activity have been given as examples in the study. To make a database connection; *src>com.bolume.ceit>veritabani.java* The code that should be written in the file is clearly written.

NEU-CEIT the code required to open WordPress pages that are embedded in View for Android application in the application is given as an example to guide some pages. The *akademikcalismalar.java* file that directs academic studies is one of them. The code snippets for *listview.java* have been written to make a list view menu. After all the clarifications, the NEU-CEIT Android mobile application has been launched and has successfully performed a screen-image transfer.

In this work, NEU-CEIT Android mobile application development process is described. The applications that need to be set up during this development process are explained. The aim of this work is to guide the beginner to the Android application, to explain the initial steps of the application, to explain the coding between the pages in detail, to give the code that passes the pages, and all the steps necessary for the researcher to overcome all problems while developing the application. This study will be an example for a researcher who wants to develop an Android-based application. In this context, it is thought that the research and development techniques used may help the researchers and software development teams who will work in similar contexts.

References

- Sokolova, Karina, Charles Perez, and Marc Lemercier. "Android application classification and anomaly detection with graph-based permission patterns." *Decision Support Systems* 93 (2017): 62-76.
- [2] Appstedltd. Received in 10.09.2016 from http://appstedltd.blogspot.com.tr/2014/07/post-development-steps-tofurnish-your.html.
- [3] Ma, Li, Lei Gu, and Jin Wang. "Research and development of mobile application for android platform." *International Journal of Multimedia and Ubiquitous Engineering* 9.4 (2014): 187-198.
- [4] M. Butler, "Android: Changing the Mobile Landscape", Pervasive Computing, (2011), pp. 4-7.
- [5] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev and C. Glezer, "Google Android: A Comprehensive Security Assessment", Security & Privacy, (2010), pp. 35 – 44.
- [6] Arslan, A. (2015). Eğitim ve Öğretimde Sosyal Medyanın Kullanımı. A. Büyükaslan, & A. M. Kırık (Ed), Sosyalleşen olgular sosyal medya 2 (191-219). Konya: Çizgi Kitapevi.
- [7] Sönmez, F. (2010). Mobil cihazların uzaktan eğitime olumlu katkıları. ABMYO Dergisi,(17), 31-36.
- [8] Sung, Y. T., Chang, K. E., & Liu, T. C. (2016). The effects of integrating mobile devices with teaching and learning on students' learning performance: A meta-analysis and research synthesis. Computers & Education, 94, 252-275.
- [9] Sönmez, F., Gülpınar, V., & Yıldırım, H.(2014). Affirmative contributions of mobile devices to distance education. International Conference on Science, Technology, Engineering and Management, DOI: 10.13140/RG.2.1.2576.6245
- [10] Kumar, N. A., Krishna, K. H., & Manjula, R. (2016). Challenges and Best Practices in Mobile Application Development. Imperial Journal of Interdisciplinary Research, 2(12).
- [11] Ma, Li, Lei Gu, and Jin Wang. "Research and development of mobile application for android platform." *International Journal of Multimedia and Ubiquitous Engineering* 9.4 (2014): 187-198.
- [12] Bing, Y. (2012). Analysis and Research of System Security Based on Android, *Fifth International Conference on Intelligent Computation Technology and Automation*, 581-584, Hunan/China (2012).
- [13] Pandey, G., & Dani, D. (2014). Android mobile application build on eclipse. International Journal of Scientific and Research Publications, 4(2), 1-2.
- [14] Chawla, D., Aggarwal, R., & Aggarwal, A. (2016). Scientific Research and Development of Mobile Application for Android Platform. IITM Journal of Management and IT, 7(1), 125-131.
- [15] Wu, F., Clarke, D., Jiang, J., Baba, A., & Buford, S. (2016). The Digital Age of Campus Maps on Mobile Devices. Journal of Computer and Communications, 4(07), 22.