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

1 author:

Şenay Kocakoyun
Istanbul Gedik University
11 PUBLICATIONS   **357** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Infographics View project

Development View project

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

Şenay KOCAKOYUN[1]

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 been influenced by these developments. Practices used in 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

[1]*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).
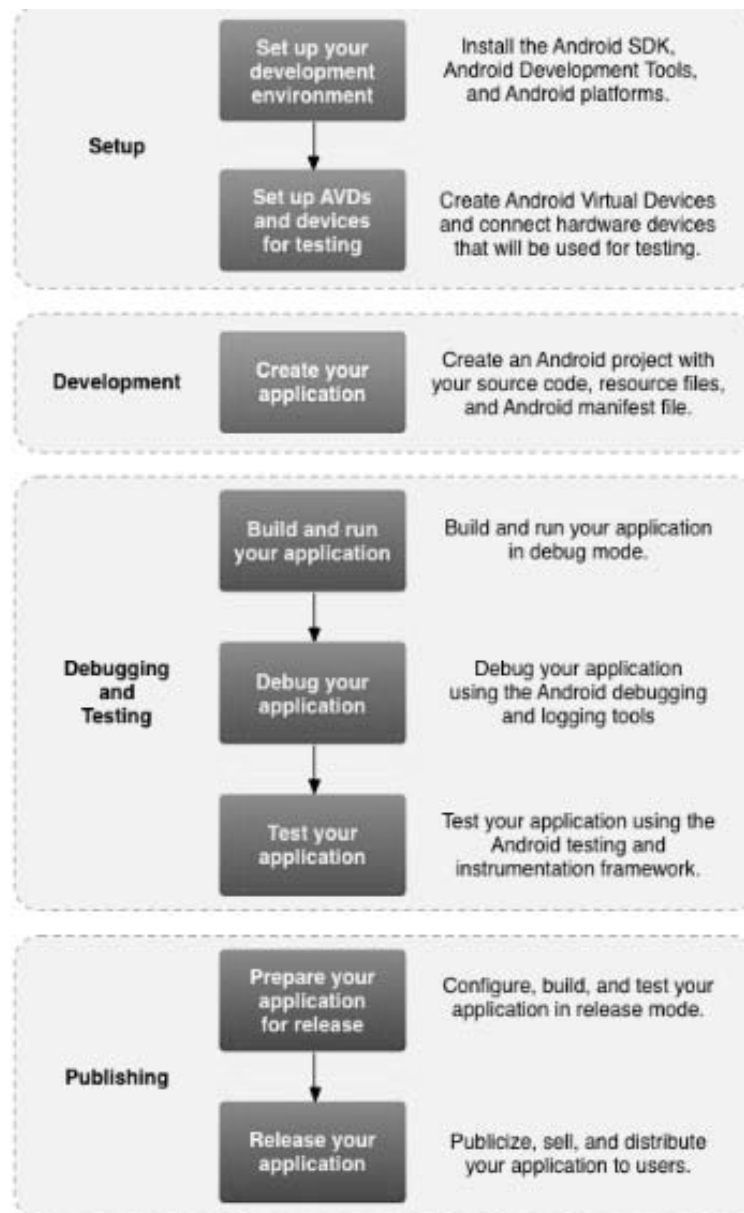

Figure 1. Steps for Application Development [2]

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 Java SE Development Kit (JDK) can be downloaded 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).
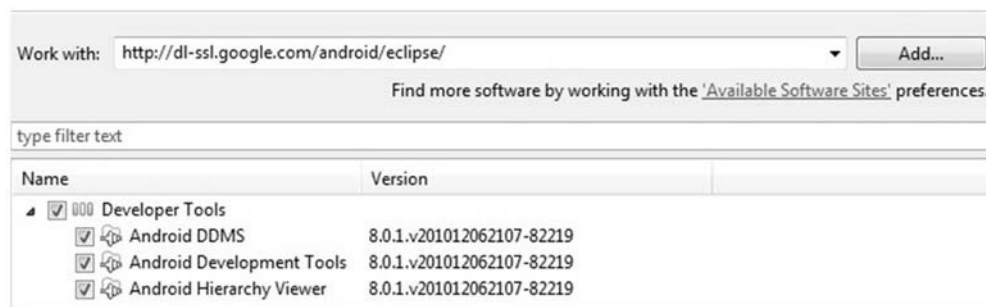

Figure 2. Eclipse ADT setup

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).
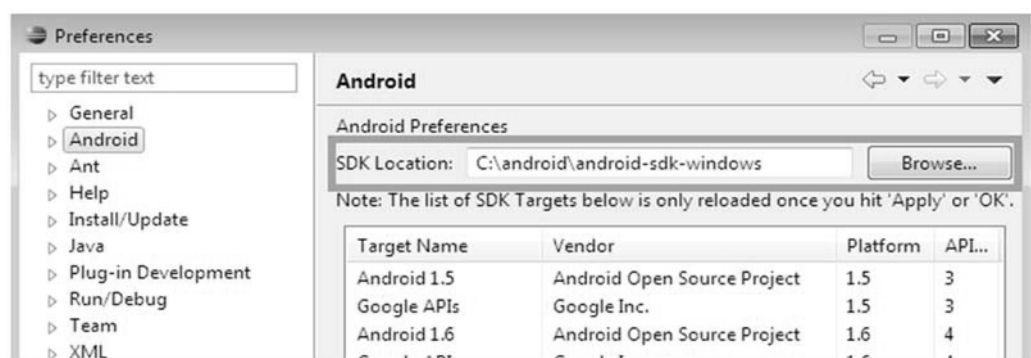

Figure 3. Android SDK Directory settings

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).
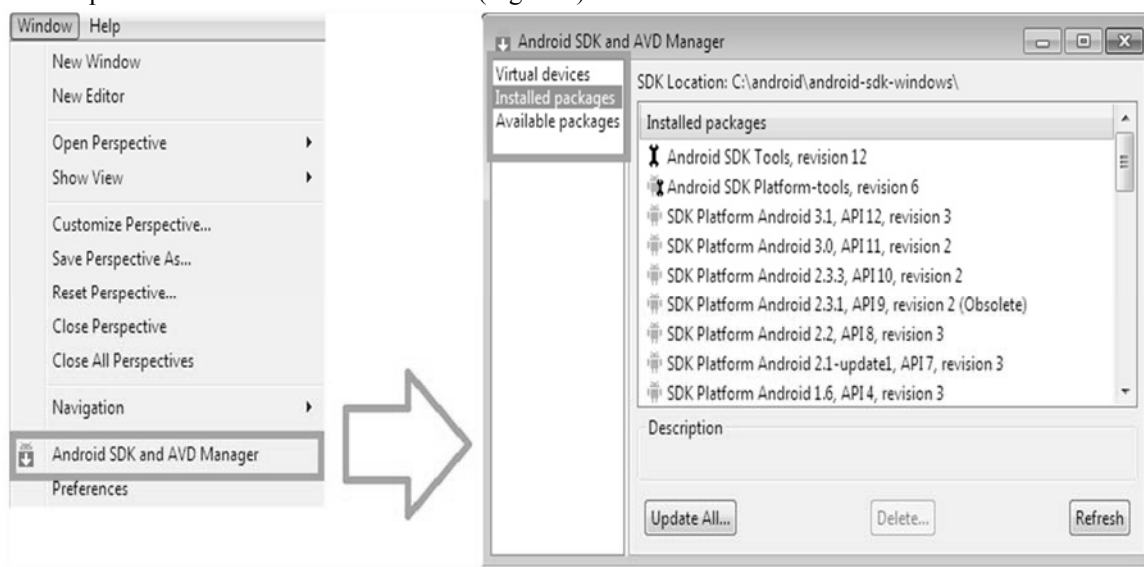

Figure 4. Android SDK and AVD Manager settings

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).
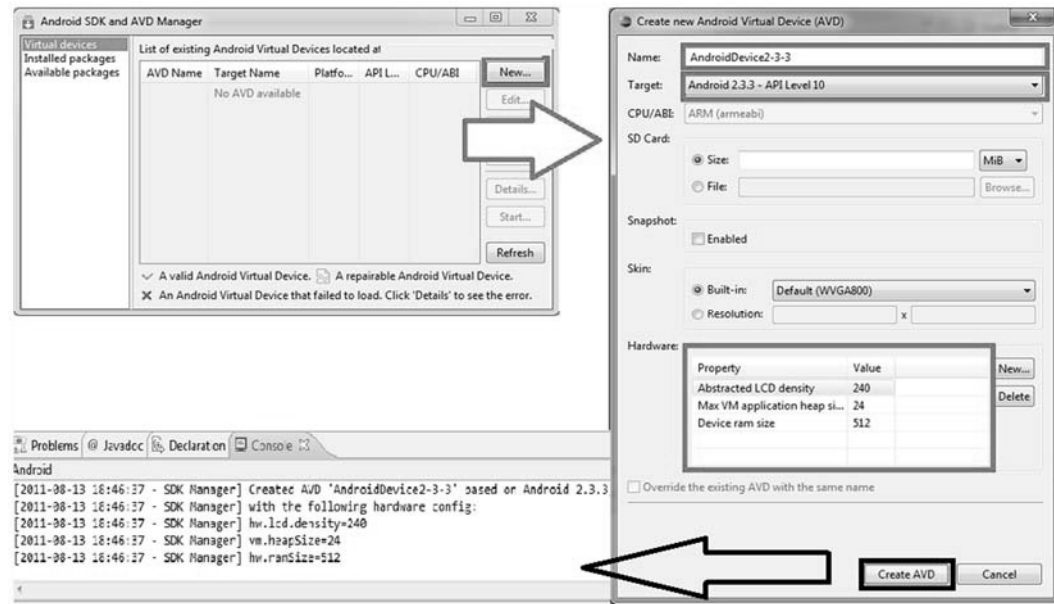
Figure 5. Virtual Device setup for Android

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).

Figure 6. Hierarchical view of Android project files

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).
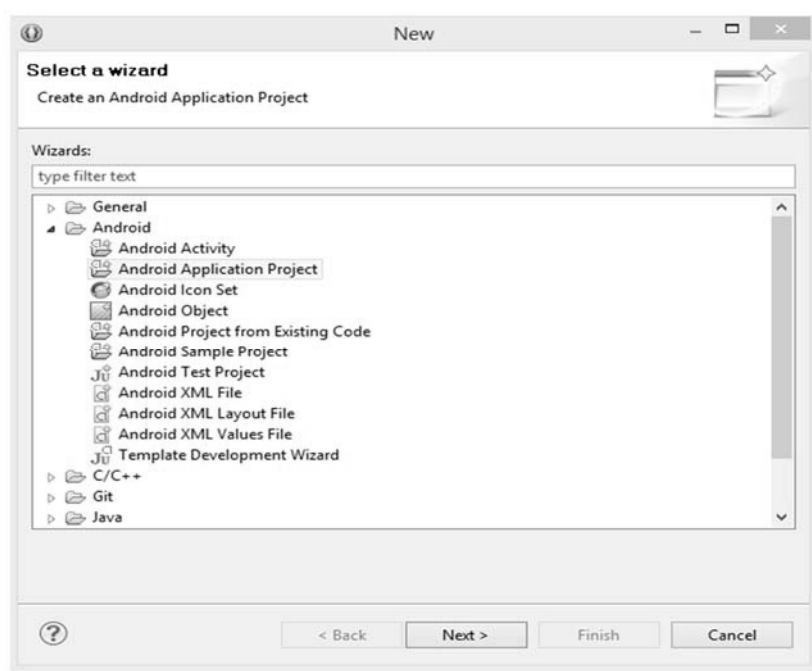

Figure 7. Android Project Creation Screen-I

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.
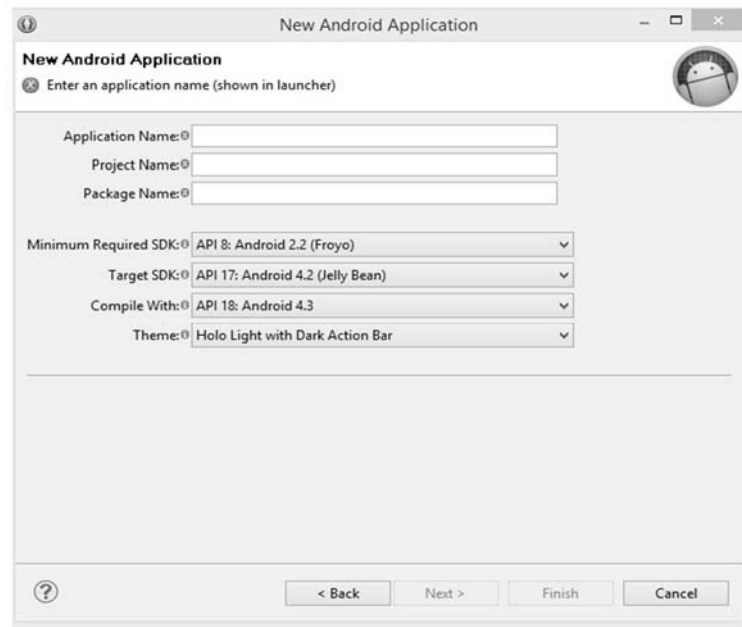
Şenay KOCAKOYUN

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.
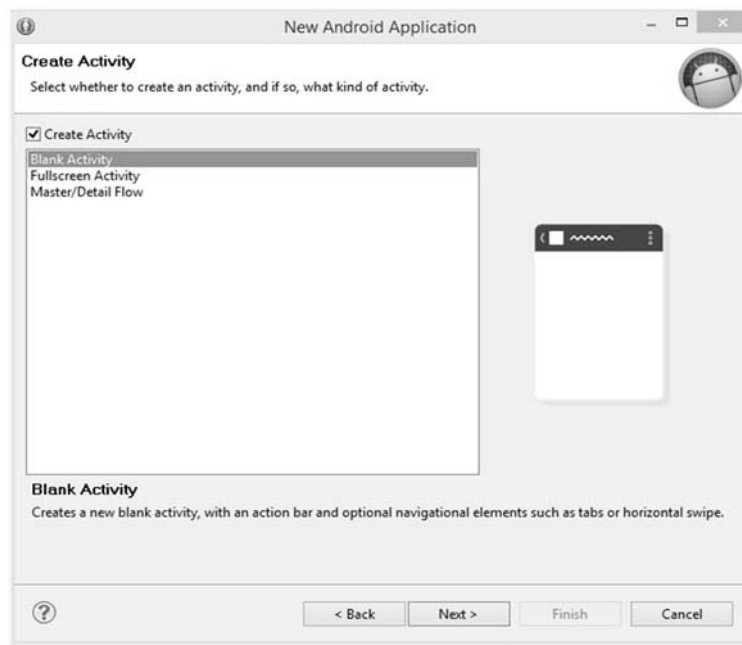

Figure 9. Android Project Creation Screen-III

   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).
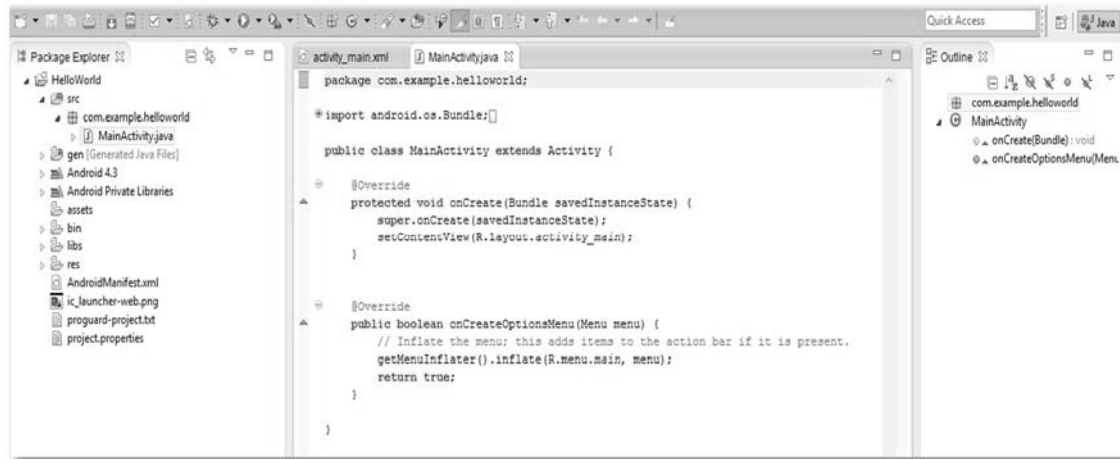

Figure 10. Android Project Creation Screen-IV

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

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

   To create NEU-CEIT Android application, the following designs and coding are done.
1.   Select *File> New*
2.   Go to the Android Application project.
3.   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).
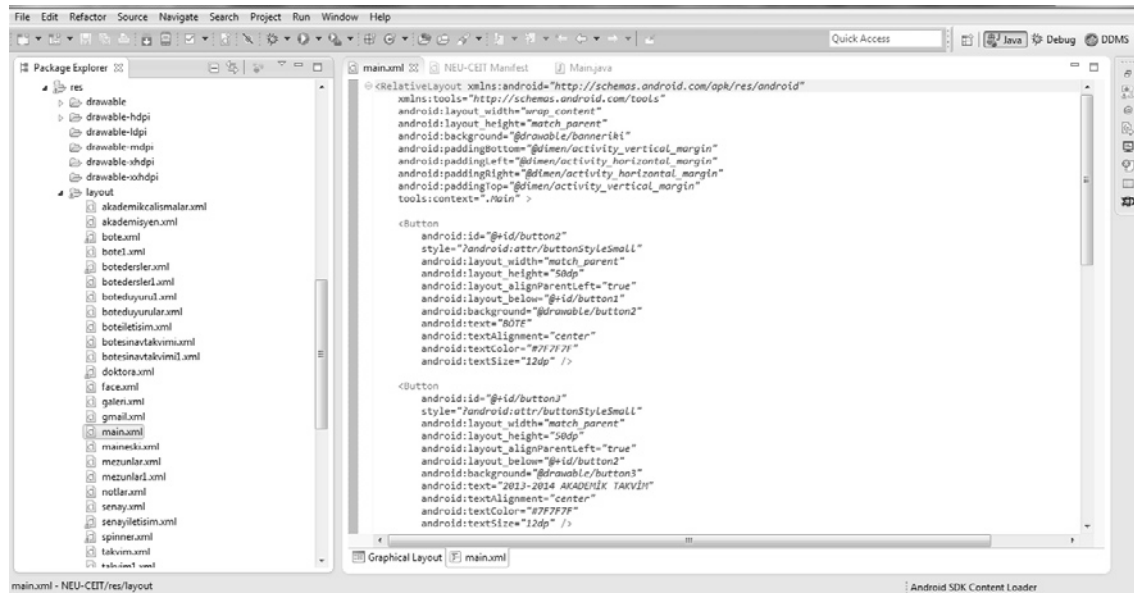
Figure 12. NEU-CEIT res>layout> main.xml page

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
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="wrap_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
    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
    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"
```

```
    android:background="@drawable/button3"
    android:text="2013-2014 AKADEMİK TAKVİM"
    android:textAlignment="center"
    android:textColor="#7F7F7F"
    android:textSize="12dp" />
<Button
    android:id="@+id/button4"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_below="@+id/button3"
    android:background="@drawable/button4"
    android:text="ÖĞRENCİ BİLGİ SİSTEMİ"
    android:textAlignment="center"
    android:textColor="#7F7F7F"
    android:textSize="12dp" />
<Button
    android:id="@+id/button5"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_below="@+id/button4"
    android:background="@drawable/button5"
    android:text="NEU-UZEM (Uzaktan Eğitim Merkezi)"
    android:textAlignment="center"
    android:textColor="#7F7F7F"
    android:textSize="12dp" />
<Button
    android:id="@+id/button6"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/button5"
    android:background="@drawable/button6"
    android:text="İLETİŞİM"
    android:textAlignment="center"
    android:textColor="#7F7F7F"
    android:textSize="12dp" />
<Button
    android:id="@+id/button1"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button2"
    android:layout_alignParentRight="true"
    android:layout_marginTop="162dp"
    android:background="@drawable/button1"
    android:shadowColor="#002200"
    android:text="YAKIN DOĞU ÜNİVERSİTESİ  (1988&apos;den Bugüne Yakın Doğu)"
    android:textAlignment="center"
    android:textColor="#7F7F7F"
    android:textSize="12dp" />
</RelativeLayout>
```

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).
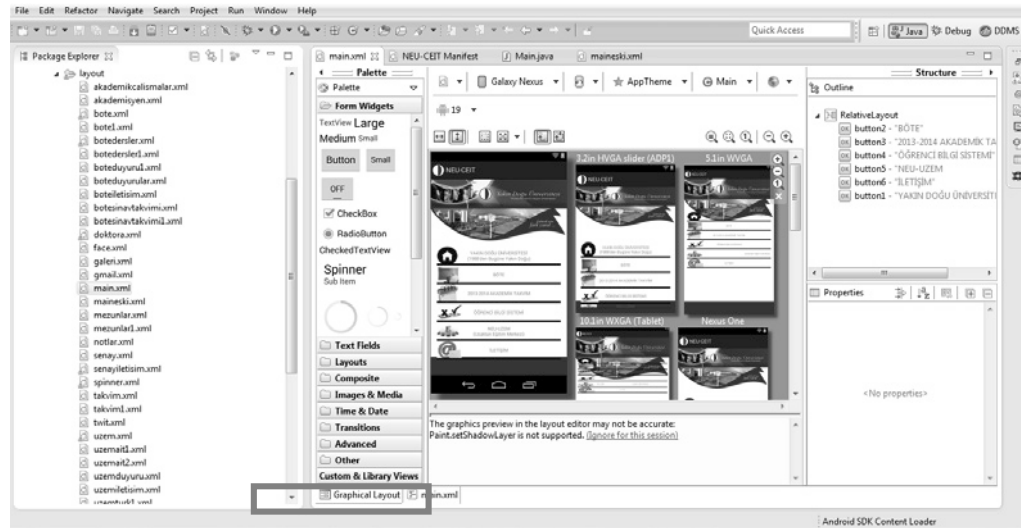
Figure 13. Main.xml graphical layout screenshot

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).

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 {
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
Button buton1=(Button) findViewById(R.id.button1);
buton1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
                startActivity(new Intent("com.bolume.ceit.YDU"));   } });
Button buton2=(Button) findViewById(R.id.button2);
buton2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
                startActivity(new Intent("com.bolume.ceit.LISTVIEW")); } });
Button buton3=(Button) findViewById(R.id.button3);
buton3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
                startActivity(new Intent("com.bolume.ceit.TAKVIM"));    } });
Button buton4=(Button) findViewById(R.id.button4);
buton4.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
                startActivity(new Intent("com.bolume.ceit.NOTLAR"));  } });
Button buton5=(Button) findViewById(R.id.button5);
buton5.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
                startActivity(new Intent("com.bolume.ceit.UZEM")); } });
        Button buton6=(Button) findViewById(R.id.button6);
        buton6.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
        startActivity(new Intent("com.bolume.ceit.SENAYILETISIM")); }});
}
}
}
```

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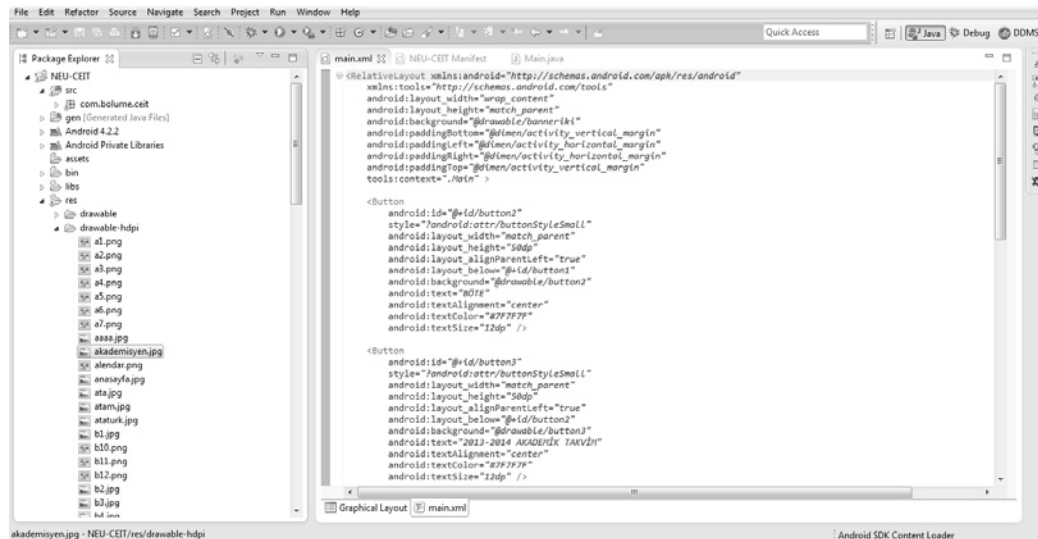
Figure 15. NEU-CEIT res>drawable-hdpi Picture files

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.

6. 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).
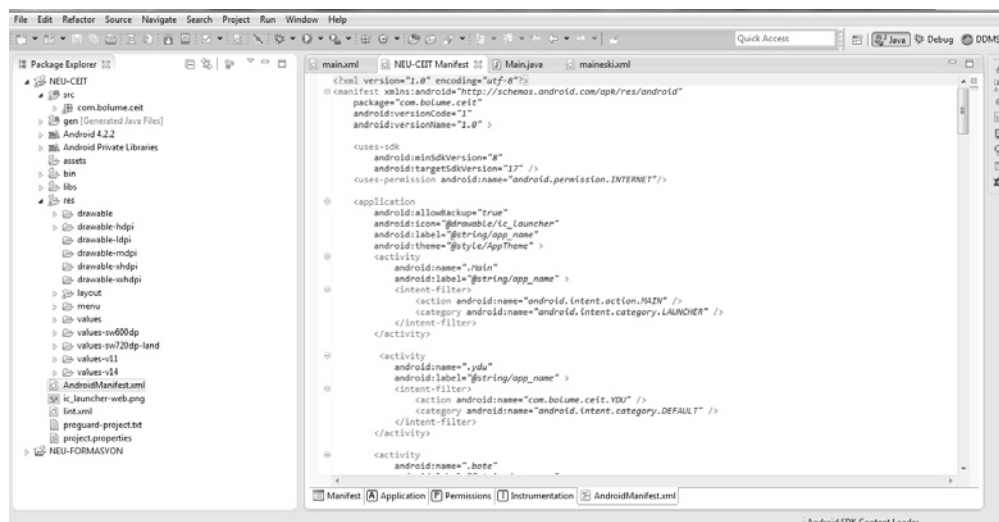

Figure 16. NEU-CEIT Manifest.xml page

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>
```

```xml
<activity
    android:name=".veritabaniekrani"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="com.bolume.ceit.VERITABANIEKRANI" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity
    android:name=".veritabani"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="com.bolume.ceit.VERITABANI" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

<activity
    android:name=".spinner"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="com.bolume.ceit.ACILIRLISTE" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
</application>
</manifest>
```

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:

```java
//veritabanı.java

package com.bolume.ceit;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class Veritabani extends SQLiteOpenHelper {
    private static final String VERITABANI ="ogrenciler.db";
    private static final int SURUM = 1;
    public Veritabani(Context con){
            super(con,VERITABANI,null,SURUM); }

    @Override
    public void onCreate(SQLiteDatabase db) {
            // TODO Auto-generated method stub
    db.execSQL("CREATE TABLE ogrenciisim(id INTEGER PRIMARY KEY
AUTOINCREMENT, isim TEXT, soyad TEXT);");    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
            // TODO Auto-generated method stub
            db.execSQL("DROP TABLE IF EXIST ogrenciisim");
            onCreate(db);
    }
}
```

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:
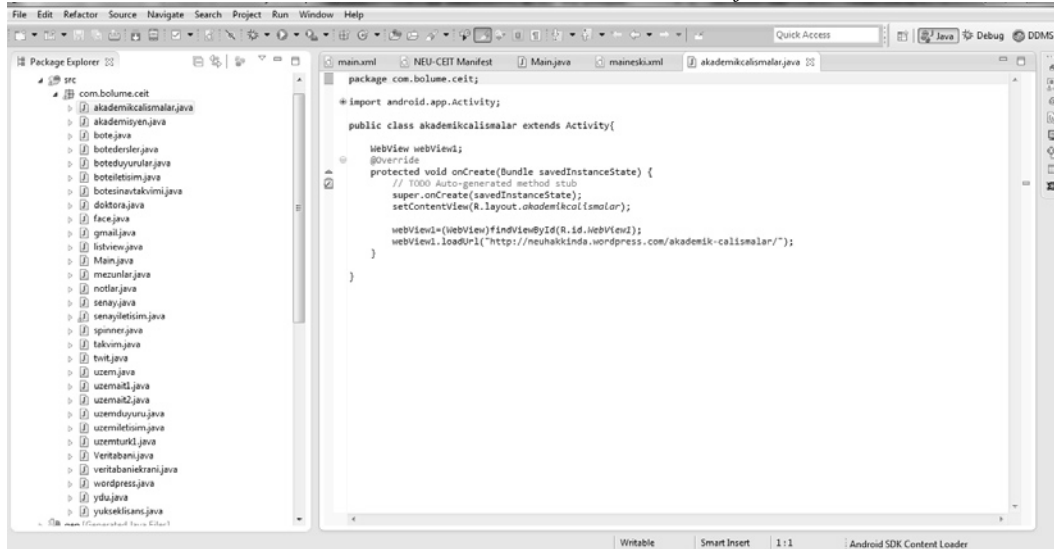
Figure 17. akademikcalismalar.java page

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

```
// akademisyen.java
package com.bolume.ceit;
import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebView;
public class akademisyen extends Activity{

        WebView webView1;

        @Override
        protected void onCreate(Bundle savedInstanceState) {

        // TODO Auto-generated method stub
                super.onCreate(savedInstanceState);
                setContentView(R.layout.akademikcalismalar);
                webView1=(WebView)findViewById(R.id.WebView1);
                webView1.loadUrl("http://neuhakkinda.wordpress.com/ogretim-uyeleri/");
        }
}
```

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

```
//mezunlar.java
package com.bolume.ceit;
import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebView;
public class mezunlar extends Activity{
        WebView webView1;
```

```
@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:

```
// senay.java
package com.bolume.ceit;
import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebView;
public class senay  extends Activity{
    WebView webView1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
            // TODO Auto-generated method stub
            super.onCreate(savedInstanceState);
            setContentView(R.layout.senay);
            webView1=(WebView)findViewById(R.id.WebView1);
            webView1.getSettings().setJavaScriptEnabled(true);
            webView1.loadUrl("http://senaykocakoyun.wordpress.com/yazilarim/");
    }
}
```

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)
            {
                    startActivity(new Intent("com.bolume.ceit.DERSPROGRAMI"));}
            else if(position==1)
            {
                    startActivity(new Intent("com.bolume.ceit.WORDPRESS")); }
            else if(position==2)
            {
                    startActivity(new Intent ("com.bolume.ceit.AKADEMISYEN")); }
```

```
                        else if(position==3)
                        {
                                startActivity(new Intent ("com.bolume.ceit.BOTE")); }
                        else if(position==4)
                        {
                                startActivity(new Intent ("com.bolume.ceit.YUKSEKLISANS")); }
                        else if (position==5)
                        {
                                startActivity(new Intent("com.bolume.ceit.DOKTORA")); }
                        else if (position==6)
                        {
                                startActivity(new Intent("com.bolume.ceit.BOTESINAV"));}
                        else if (position==7)
                        {
                        startActivity(newIntent("com.bolume.ceit.AKADAMIKCALISMALAR"));}
                        else if (position==8)
                        {
                                startActivity(new Intent("com.bolume.ceit.MEZUNLAR")); }
                        else if (position==9)
                        {
                                startActivity(new Intent("com.bolume.ceit.BOTEDUYURULAR")); }
                        else if (position==10)
                        {
                                startActivity(new Intent("com.bolume.ceit.BOTEILETISIM")); }
                        super.onListItemClick(l, v, position, id);
                        }
                @Override
                protected void onCreate(Bundle savedInstanceState) {
                        // TODO Auto-generated method stub
                        super.onCreate(savedInstanceState);
                        setListAdapter(new
        ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,stringarray));
                }
        }
```

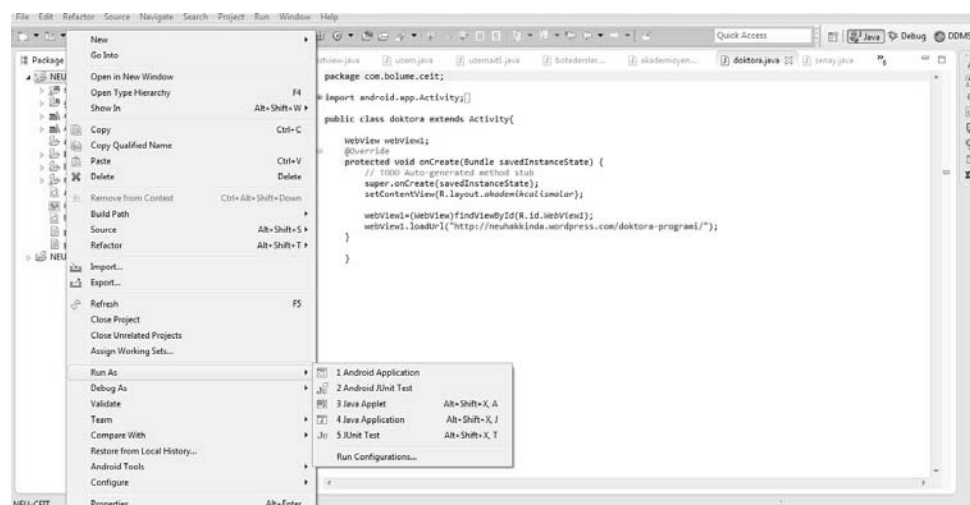10. Right click on *project>Run As>1 Android Application* (Figure 18).

Figure 18. Running the application

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.
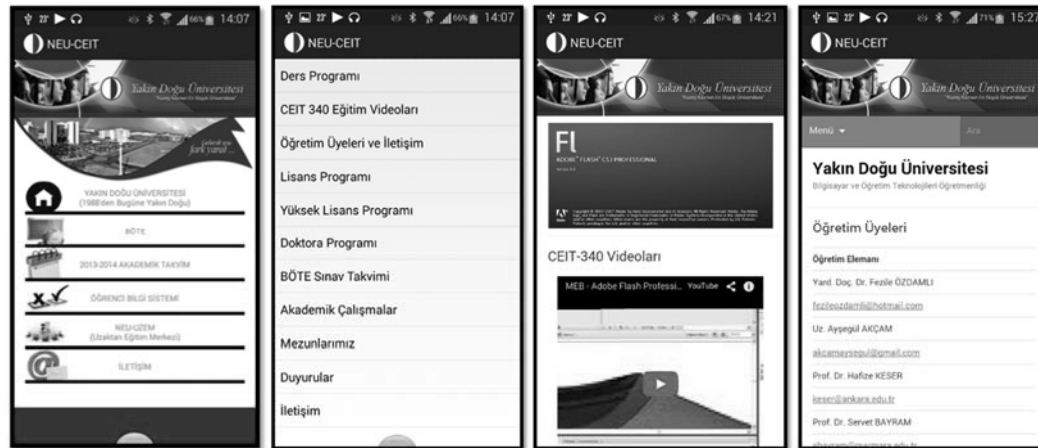

Figure 19. NEU-CEIT Screen Views

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

[1]   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-to-furnish-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.