

**YILDIRIM BEYAZIT UNIVERSITY**

**FACULTY OF ENGINEERING & NATURAL SCIENCES**

**SUMMER PRACTICE REPORT**

|  |  |
| --- | --- |
| Student Information | |
| Department | Computer Engineering |
| Name, Surname | Nuh Yılmaz ŞENTÜRK |
| Student ID Number | 1205011047 |
| The Name of the Company | IPA SAVUNMA |
| The Name of the Department | Ar-Ge |

**Notes for the practice report:**

1. This report will be written in English on computer and printed copy of it will be delivered to Faculty Secretarial. *If the company where the summer practice was held asks for the Turkish version of the report*, the report must also be translated to Turkish and the Turkish copy will be signed by the responsible engineer. Please note that this is *NOT* mandatory and will be applied *if asked by the company*.
2. Pages will be printed one-sided and in black & white color.
3. It is allowed to replicate weekly work schedules and other pages when needed.
4. The soft copy of the report will be stored for at least one year and given to Commission of Summer Practice of your department, when required.

**Weekly Work Schedule I**

|  |  |  |  |
| --- | --- | --- | --- |
| Work schedule between 18.07.2016 and 22.07.2016 | | | |
|  | The work performed | The page number | Hours worked |
| Monday | Installing Android Studio & Java JDK | 01 | 8 hours |
| Tuesday | Android Project Structure | 02 | 8 hours |
| Wednesday | First Application | 03 | 8 hours |
| Thursday | Chancing App's Icon and Background Color | 04 | 8 hours |
| Friday | Repeat Java Programming | 05 | 8 hours |
|  |  |  |  |
| Section: 1 | | Total Hours: 40 | |

**Weekly Work Schedule II**

|  |  |  |  |
| --- | --- | --- | --- |
| Work schedule between 25.07.2016 and 29.07.2016 | | | |
|  | The work performed | The page number | Hours worked |
| Monday | Linear Layout & Relative Layout | 06 | 8 hours |
| Tuesday | Toasts | 07-08 | 8 hours |
| Wednesday | Ardunio Uno | 09 | 8 hours |
| Thursday | Buttons | 10-11 | 8 hours |
| Friday | Button Design | 12-13 | 8 hours |
|  |  |  |  |
| Section: 2 | | Total Hours: 40 | |

**Weekly Work Schedule III**

|  |  |  |  |
| --- | --- | --- | --- |
| Work schedule between 01.08.2016 and 05.08.2016 | | | |
|  | The work performed | The page number | Hours worked |
| Monday | TextView & Styles | 14 | 8 hours |
| Tuesday | Translation and ScrollView | 15 | 8 hours |
| Wednesday | EditText and Types | 16-17 | 8 hours |
| Thursday | Activities and Intent | 18-19 | 8 hours |
| Friday | Phone Call Application | 20-21 | 8 hours |
|  |  |  |  |
| Section: 3 | | Total Hours: 40 | |

**Weekly Work Schedule IV**

|  |  |  |  |
| --- | --- | --- | --- |
| Work schedule between 08.08.2016 and 12.08.2016 | | | |
|  | The work performed | The page number | Hours worked |
| Monday | Website Application | 22-23 | 8 hours |
| Tuesday | Share Application | 24-25 | 8 hours |
| Wednesday | Mail Application | 26-27 | 8 hours |
| Thursday | SMS Send Application | 28-29 | 8 hours |
| Friday | GPS Application | 30-31-32 | 8 hours |
|  |  |  |  |
| Section: 4 | | Total Hours: 40 | |

|  |  |  |
| --- | --- | --- |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 18.07.2016 | | Page Number: 01 |
| Work Performed: Installing Android Studio & Java JDK  I will work on Android during this summer practice. I downloaded the necessary programs for Android Programming. These programs as follows:   * Android Studio 2.1 * JDK (Java SE Development Kit) - 8u92   After installation of the programs, Android SDK installed plug-ins. This process took several hours. I viewed Android studio interface. Checked out the XML structures. Android Studio interface looks like:  C:\Users\NOAH\Desktop\Screenshot_1.jpg  I connect my phone to Android Studio for test applications. The phone is Galaxy S5 also has Android Version 6.0.1. An empty android application was created. The program tested with the phone. The program worked properly. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 19.07.2016 | | Page Number: 02 |
| Work Performed: Android Project Structure  Today, I learned structure of android projects. I researched them all.    **AndroidManifest.xml**   * overall project config and settings   **src/java/...**   * source code for your Java classes   **res/...** = resource files (many are XML)   * drawable/ = images * layout/ = descriptions of GUI layout * menu/ = overall app menu options * values/ = constant values and arrays * strings = localization data * styles = general appearance styling   **Gradle**   * a build/compile management system * **build.gradle** = main build config file | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 20.07.2016 | | Page Number: 03 |
| Work Performed: First Application  I want to do my first application today. I researched examples on internet. I created a new project. The project interface edited look like:    Project contains the following objects:   * 3 TextView (Output text, Name title, Surname title) * 2 EditText (Name, Surname) * 1 Button (For write name and surname to TextView)   These objects defined in Java code:  **final** TextView tv =(TextView) findViewById(R.id.***textView***); **final** Button btn = (Button) findViewById(R.id.***button***); **final** EditText text1 =(EditText) findViewById(R.id.***editText***); **final** EditText text2 =(EditText) findViewById(R.id.***editText2***);  After that, Write this code for button:  btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  tv.setText(text1.getText()+**" "**+text2.getText());  } });  I installed the application on the phone. Application worked perfectly. When the user presses the button, "Text" are changing with “Name + Surname”. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 21.07.2016 | | Page Number: 04 |
| Work Performed: Chancing App's Icon and Background  I worked on application customization today. I researched how to change the application icon. Icon images are stored in the project file. Location of the icon is stated in AndroidManifest.xml. In this way:  **android:icon="@mipmap/ic\_launcher"**  There is icon files (png) of different sizes. These pictures are for devices with different screen resolution. These icon sizes are as follows:  C:\Users\NOAH\Desktop\Screenshot_4.jpg   |  |  |  |  | | --- | --- | --- | --- | | **Icon Type** | **Abbreviation** | **Resolution** | **Size** | | High density | hdpi | ~160 dpi | 48 x 42 px | | Medium density | mdpi | ~240 dpi | 72 x 72 px | | Extra High density | xhdpi | ~320 dpi | 96 x 96 px | | XX High density | xxhdpi | ~480 dpi | 144 x 144 px | | XXX High density | xxxhdpi | ~640 dpi | 192 x 192 px |   I found a picture for my application’s icon. I reduced the sizes like this. Project icons have changed with them. I saved this application on my computer as apk. I want to install with a new way. I installed the app and worked. Application icon has changed. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 22.07.2016 | | Page Number: 05 |
| Work Performed: Repeat Java Programming  I repeated JAVA programming all day. I remembered things I forgot. I've studied this topics:   * Data Types and Variables * Arrays * Control Structures (If-else, switch-case etc.) * Loops (For, While etc.) * Methods * Classes, Objects * Polymorphism * Inheritance   I learned definition an array on XML file.  <**string-array name="iller"**>  <**item**>ANKARA</**item**>  <**item**>TRABZON</**item**>  <**item**>İSTANBUL</**item**> </**string-array**>  Also, I learned how to use this Array in JAVA.  String[] some\_array = getResources().getStringArray(R.array.iller) | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 25.07.2016 | | Page Number: 06 |
| Work Performed: Linear Layout & Relative Layout  I did work and practical examples on Linear Layout & Relative Layout today. These are one of the most important building blocks android.  [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout.html) is a view group that aligns all children in a single direction, vertically or horizontally. You can specify the layout direction with the android:orientation attribute.  [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout.html) is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center).  Relativelayout was created and customized in java. Also I did research on delay. Added a button, after clinking button; TextView was changed for a second range. For this, used handler;  btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  tv.setText(**"Text"**);  Handler handler1=**new** Handler();  **for** (**int** a = 1; a<=10 ;a++) {  handler1.postDelayed(**new** Runnable() {  @Override  **public void** run() {  tv.setText(tv.getText()+**"\*"**);  }  }, 1000 \* a);  }  } });  Finally, I researched TableLayout today. TableLayout is a ViewGroup that displays child View elements in rows and columns. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 26.07.2016 | | Page Number: 07 |
| Work Performed: Toasts  I did some research on Toast Messages today. I made examples of them.  A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. For example, navigating away from an email before you send it triggers a "Draft saved" toast to let you know that you can continue editing later. Toasts automatically disappear after a timeout. Toast simply works like this:  C:\Users\Noah\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Screenshot_5.jpgContext context = getApplicationContext(); CharSequence text = "Hello toast!"; int duration = Toast.LENGTH\_SHORT; Toast.makeText(context, text, duration).show();  Also we can dynamically configure it. I changed the toast message to a picture in my example. In addition, I've prepared a dynamic message. Application looks like: | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 26.07.2016 | | Page Number: 08 |
| Work Performed: Toasts  Java Codes of Example:  **public class** MainActivity **extends** AppCompatActivity {  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  Button btn1 =(Button)findViewById(R.id.***button***);  Button btn2 =(Button)findViewById(R.id.***button2***);  Button btn3 =(Button)findViewById(R.id.***button3***);  btn1.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  Toast.*makeText*(MainActivity.**this**,**"Toast Message"**,Toast.***LENGTH\_SHORT***).show();  }  });  btn2.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  ImageView cat = **new** ImageView(getApplicationContext());  cat.setImageResource(R.drawable.***cat***);  Toast toast1 =**new** Toast(getApplicationContext());  toast1.setView(cat);  toast1.setDuration(Toast.***LENGTH\_LONG***);toast1.show();  }  });  btn3.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  TextView text =**new** TextView(getApplicationContext());  text.setText(**"Bla bla bla..."**);  text.setBackgroundColor(Color.***BLUE***);  text.setTextColor(Color.***WHITE***);  text.setPadding(14,14,14,14);  Toast toast =**new** Toast(getApplicationContext());  toast.setView(text);  toast.setDuration(Toast.***LENGTH\_LONG***);toast.show();  }  });  } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 27.07.2016 | | Page Number: 09 |
| Work Performed: Ardunio Uno  Today I try to connect the Arduino UNO to android phone. USB OTG-3 cable was purchased from the internet. I connect the phone with OTG USB-3 cable. But the phone did not give any response. I did some research on the internet on it. And I realized that the need additional software. These applications:   * ArduinoCommander * ArduinoDroid   After install two applications, First, we have to start ArduinoCommander, now a blue window with 4 options should appear. As we want to connect via USB we tap on "USB-Device". Now tap on "AutoDetect", the Android device will scan for Arduino(s) and show them on the screen. We will pic our Arduino. We have to tap it.  After application recognized Arduino. We can open ArduinoDroid. This app is an IDE, a compiler and an uploader for Arduino, it also uses the host mode to upload sketches, which you can write directly in the app. After you wrote a code, you can compile it, that needs some time, but works good. I wrote a small blink code for LEDs on Arduino, I tested. I used the sample code for this test application. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 28.07.2016 | | Page Number: 10 |
| Work Performed: Buttons  I did research about the buttons for use more effective. My research results show me, importance of **android:onClick** command. I decided to implement it. OnClickListener has been implement, it has connected five buttons. I wrote the command for each button with the switch-case in a method. Method name is butttonclick.  **package** com.ipadefence.app004;  **import** ...  **public class** MainActivity **extends** AppCompatActivity{  Button **btn1**,**btn2**,**btn3**,**btn4**,**btn5**;   **public void** buttonclick(View v){  **switch** (v.getId()){  **case**(R.id.***button***):  Toast.*makeText*(getApplicationContext(),**"1"**,Toast.***LENGTH\_LONG***).show();  **break**;  **case**(R.id.***button2***):  Toast.*makeText*(getApplicationContext(),**"2"**,Toast.***LENGTH\_LONG***).show();  **break**;  **case**(R.id.***button3***):  Toast.*makeText*(getApplicationContext(),**"3"**,Toast.***LENGTH\_LONG***).show();  **break**;  **case**(R.id.***button4***):  Toast.*makeText*(getApplicationContext(),**"4"**,Toast.***LENGTH\_LONG***).show();  **break**;  **case**(R.id.***button5***):  Toast.*makeText*(getApplicationContext(),**"5"**,Toast.***LENGTH\_LONG***).show();  **break**;  }  }  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 28.07.2016 | | Page Number: 11 |
| Work Performed: Buttons  Then I edited my activity\_main file like this:  *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="com.ipadefence.app004.MainActivity"**>   <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="New Button"  android:id="@+id/button"  android:layout\_alignParentTop="true"  android:layout\_centerHorizontal="true"  android:onClick="buttonclick"** />   <**Button ...** />   <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="New Button"  android:id="@+id/button3"  android:layout\_centerVertical="true"  android:layout\_centerHorizontal="true"  android:onClick="buttonclick"** />  <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="New Button"  android:id="@+id/button4"  android:layout\_below="@+id/button3"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="56dp"  android:onClick="buttonclick"** />   <**Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="New Button"  android:id="@+id/button5"  android:layout\_alignParentBottom="true"  android:layout\_centerHorizontal="true"  android:onClick="buttonclick"** />  </**RelativeLayout**> | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 29.07.2016 | | Page Number: 12 |
| Work Performed: Button Design  I made my own button design today. Also made button with using transparent photos. I found suitable pictures on internet for button design.  Screenshot_8Screenshot_7  I moved to this pictures the drawable folder. I wrote selector for this button. Selector file name: selector2.xml. Codes are:  *<?***xml version="1.0" encoding="utf-8"***?>* <**selector xmlns:android="http://schemas.android.com/apk/res/android"**> <**item android:drawable="@drawable/buton02"  android:state\_pressed="true"**/> <**item android:drawable="@drawable/buton01"** /> </**selector**>  Button01 is first picture, button2 is second picture. When user open the application, button looks like first picture. When user clicking the button, button will change to second picture. Due to this command: **android:state\_pressed="true"**/> After that, I created the button in activity\_main.xml. Like this:  <**Button  android:layout\_width="100dp"  android:layout\_height="100dp"  android:id="@+id/button2"  android:background="@drawable/selector2"  android:layout\_marginTop="41dp"  android:layout\_alignParentTop="true"  android:layout\_centerHorizontal="true"** />  I set the background of button with my selector. I designed the second button just with xml. I created two shape for this. Also I create a new selector for these shapes. Selector codes are:  *<?***xml version="1.0" encoding="utf-8"***?>* <**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  <**item android:state\_pressed="true" android:drawable="@drawable/buton\_tasarim"** />  <**item android:drawable="@drawable/buton\_tasarim2"** /> </**selector**>  Buton\_tasarim and buton\_tasarim2 are my shape files. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 29.07.2016 | | Page Number: 13 |
| Work Performed: Button Design  Buton\_tasarim.xml is like this:  *<?***xml version="1.0" encoding="utf-8"***?>* <**shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle"**>  <**gradient  android:startColor="#4DD0E1"  android:endColor="#006064"  android:angle="90"  android:type="linear"** />  <**corners  android:radius="20dp"** />  <**stroke  android:color="#ffffff"  android:width="3dp"**/> </**shape**>  Buton\_tasarim2.xml is just changing colors for click effect. I created the second button in activity\_main.xml. Like this:  <**Button  android:layout\_width="150dp"  android:layout\_height="wrap\_content"  android:text="Buton"  android:id="@+id/button"  android:background="@drawable/buton\_selector"  android:layout\_centerVertical="true"  android:layout\_centerHorizontal="true"** />    I tested the application buttons. They worked nicely. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 01.08.2016 | | Page Number: 14 |
| Work Performed: TextView & Styles  Today I researched in detail of TextViews. I made entrance to the first topic Styles for this. A style is a simple resource that is referenced using the value provided in the name attribute (not the name of the XML file). As such, you can combine style resources with other simple resources in the one XML file, under one <resources> element. For example:  *<?***xml version="1.0" encoding="utf-8"***?>* <**resources**> <**style name="CodeFont" parent="@android:style/TextAppearance.Medium"**>  <**item name="android:layout\_width"**>fill\_parent</**item**>  <**item name="android:layout\_height"**>wrap\_content</**item**>  <**item name="android:textColor"**>#00FF00</**item**>  <**item name="android:typeface"**>monospace</**item**> </**style**> </**resources**>  We can use this style for all textviews like this:  <**TextView style="@style/CodeFont" android:text="@string/hello"** />  To allow users to copy some or all of the TextView's value and paste it somewhere else, set the XML attribute android:textIsSelectable to "true" or call setTextIsSelectable(true). The textIsSelectable flag allows users to make selection gestures in the TextView, which in turn triggers the system's built-in copy/paste controls. Also android:autoLink : Controls whether links such as urls and email addresses are automatically found and converted to clickable links. The default value is "none", disabling this feature. Other constants: web, email, phone, map, all. We can change the font of textview with android:typeface. Typeface (normal, sans, serif, monospace) for the text. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 02.08.2016 | | Page Number: 15 |
| Work Performed: Translation and ScrollView  Today, Turkish contracts were translated into English for the company. This took a lot of my time. I did not do much except this. I have reviewed the use of ScrollView.  When an app has layout content that might be longer than the height of the device and that content should be vertically scrollable, then we need to use a ScrollView. Layout container for a view hierarchy that can be scrolled by the user, allowing it to be larger than the physical display. A ScrollView is a FrameLayout, meaning you should place one child in it containing the entire contents to scroll; this child may itself be a layout manager with a complex hierarchy of objects. A child that is often used is a LinearLayout in a vertical orientation, presenting a vertical array of top-level items that the user can scroll through. I made an application about it. I added a TextView and added a text on TextView like out of bonders. Added ScrollView to read properly. Worked Nicely.  *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="com.ipadefence.app006.MainActivity"**>   <**ScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_above="@+id/linearLayout"**>    <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/hello"  android:textSize="16sp"** />   </**ScrollView**>   <**LinearLayout  android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_alignParentBottom="true"  android:layout\_alignParentRight="true"  android:layout\_alignParentEnd="true"  android:id="@+id/linearLayout"**>   <**Button ...** />   <**Button ...** />  </**LinearLayout**> </**RelativeLayout**> | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 03.08.2016 | | Page Number: 16 |
| Work Performed: EditText and Types  I studied in detail EditText today. Viewed use InputType for editText. Every text field expects a certain type of text input, such as an email address, phone number, or just plain text. So it's important that you specify the input type for each text field in your app so the system displays the appropriate soft input method (such as an on-screen keyboard). You should always declare the input method for your text fields by adding the android:inputType attribute to the <EditText> element.I used these inputTypes on my application: textPersonalName, textPassword, numberPassword, textEmailAddress, phone, textPostalAddress, textMultiLine, time, date, number, numberSigned, numberDecimal.  Also I used AutoCompleteTextView on my application. An editable text view that shows completion suggestions automatically while the user is typing. The list of suggestions is displayed in a drop down menu from which the user can choose an item to replace the content of the edit box with.  MainActivity.java file for AutoCompleteTextView items:  **package** com.ipadefence.app009;  **import** ...  **public class** MainActivity **extends** AppCompatActivity {   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   AutoCompleteTextView act = (AutoCompleteTextView) findViewById(R.id.***act***);    String[] cihazlar = getResources().getStringArray(R.array.***Cihazlar***);  ArrayAdapter<String> myarrayadapter =**new** ArrayAdapter<String>(getApplicationContext(),android.R.layout.***simple\_dropdown\_item\_1line***,cihazlar);  act.setAdapter(myarrayadapter);   } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 03.08.2016 | | Page Number: 17 |
| Work Performed: EditText and Types  Activity\_main.xml and Interface:  *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="com.ipadefence.app009.MainActivity"**>  <**ScrollView  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"  android:id="@+id/scrollView"** >  <**LinearLayout  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_alignParentTop="true"  android:layout\_centerHorizontal="true"**>  <**EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText"  android:focusable="true"  android:layout\_gravity="center\_horizontal"** />  **//OTHER EditText codes HERE.**  <**EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:inputType="numberDecimal"  android:ems="10"  android:id="@+id/editText13"  android:layout\_gravity="center\_horizontal"  android:hint="number(decimal)"** />  <**AutoCompleteTextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:completionThreshold="1"  android:id="@+id/act"  android:hint="AutoComplete"** />  </**LinearLayout**>  </**ScrollView**> </**RelativeLayout**> | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 04.08.2016 | | Page Number: 18 |
| Work Performed: Activities and Intent  An Activity is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map. Each activity is given a window in which to draw its user interface. The window typically fills the screen, but may be smaller than the screen and float on top of other windows.  An Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use-cases: To start an activity, to start a service, to deliver a broadcast. An Activity represents a single screen in an app. You can start a new instance of an Activity by passing an Intent to startActivity(). The Intent describes the activity to start and carries any necessary data. I made application for use intents and activities. There are three activities in my application. There is also data transfer from the main page to third page.    MAIN ACTIVITY SECOND ACTIVITY THIRD ACTIVITY | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 04.08.2016 | | Page Number: 19 |
| Work Performed: Activities and Intent  On MainActivity.java, I added a method for switching between activities:  **public void** on\_click(View view){  **if**(view.getId()==R.id.***button***){  *//2.ekran* Intent intent=**new** Intent(getApplicationContext(),ikinciEkran.**class**);  startActivity(intent);  }  **else if**(view.getId()==R.id.***button2***){  *//3.ekran* CharSequence charSequence = **et**.getText();  **if**(TextUtils.*isEmpty*(charSequence)){  Toast.*makeText*(getApplicationContext(),**"Text Boş.."**,Toast.***LENGTH\_SHORT***).show();  }  **else** {  Intent intent = **new** Intent(getApplicationContext(), Ucuncuekran.**class**);  intent.putExtra(**"anahtar"**, charSequence);  startActivity(intent);  }  } }  Also I added onClick to my buttons on activity\_main.xml like this: **android:onClick="on\_click"**  Third page java codes for show the text coming from main page:  **public class** Ucuncuekran **extends** AppCompatActivity {  TextView **tv**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_ucuncuekran***);  **tv** =(TextView) findViewById(R.id.***textView***);  Bundle gelenveri=getIntent().getExtras();  CharSequence charSequence=gelenveri.getCharSequence(**"anahtar"**);  **tv**.setText(charSequence);  } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 05.08.2016 | | Page Number: 20 |
| Work Performed: Phone Call Application  Today, I made application to call number from editText. For this, I wrote the code to allow call on manifest file.  AndroidManifest.xml:  *<?***xml version="1.0" encoding="utf-8"***?>* <**manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.ipadefence.app014"**>  <**uses-permission android:name="android.permission.CALL\_PHONE"** />  <**application ...** </**application**> </**manifest**>  Also, beginning in Android 6.0 (API level 23), users grant permissions to apps while the app is running, not when they install the app. This approach streamlines the app install process, since the user does not need to grant permissions when they install or update the app. It also gives the user more control over the app's functionality; for example, a user could choose to give a camera app access to the camera but not to the device location. The user can revoke the permissions at any time, by going to the app's Settings screen. For this, I used this code in my application for permission:  ActivityCompat.*requestPermissions*(**this**,  **new** String[]{Manifest.permission.***CALL\_PHONE***},3);  I use the intent make calls. Application design like this: | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 05.08.2016 | Page Number: 21 | |
| Work Performed: Phone Call Application  My MainActivity.java:  **package** com.ipadefence.app014; **import** android.Manifest; **import** android.app.Activity; **import** android.content.Intent; **import** android.net.Uri; **import** android.support.v4.app.ActivityCompat; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.text.TextUtils; **import** android.view.View; **import** android.widget.EditText; **import** android.widget.ImageView; **import** android.widget.Toast;  **public class** MainActivity **extends** AppCompatActivity {  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **final** EditText editText =(EditText) findViewById(R.id.***editText***);  ImageView call= (ImageView) findViewById(R.id.***imageView***);   ActivityCompat.*requestPermissions*(**this**,  **new** String[]{Manifest.permission.***CALL\_PHONE***},3);   call.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  **if**(TextUtils.*isEmpty*(editText.getText())){  Toast.*makeText*(getApplicationContext(),**"Numara Giriniz.."**,Toast.***LENGTH\_SHORT***);  }  **else** {  Intent intent = **new** Intent(Intent.***ACTION\_CALL***);  intent.setData(Uri.*parse*(**"tel:"** + editText.getText().toString()));  startActivity(intent);  }  }  });  } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 08.08.2016 | Page Number: 22 | |
| Work Performed: Website Application  Today, I made application to open website on android browser from editText. The application opens the website in editText. You do not need a permission to open website.  My activitymain.xml:  *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="com.ipadefence.app015.MainActivity"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Open the Website"  android:textSize="40dp"  android:background="#f45959"  android:padding="5dp"  android:gravity="center"  android:id="@+id/tv"  android:layout\_marginTop="37dp"  android:layout\_below="@+id/editText"  android:layout\_alignParentLeft="true"  android:layout\_alignParentStart="true"** />   <**EditText  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:inputType="textPersonName"  android:text="www.example.com"  android:ems="10"  android:id="@+id/editText"  android:layout\_alignParentTop="true"  android:layout\_alignParentLeft="true"  android:layout\_alignParentStart="true"** /> </**RelativeLayout**> | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 08.08.2016 | Page Number: 23 | |
| Work Performed: Website Application  My MainActivity.java:  **package** com.ipadefence.app015;  **import** android.Manifest; **import** android.content.Intent; **import** android.net.Uri; **import** android.support.v4.app.ActivityCompat; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.TextView;  **public class** MainActivity **extends** AppCompatActivity {   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  TextView textView= (TextView) findViewById(R.id.***tv***);  **final** EditText editText=(EditText) findViewById(R.id.***editText***);  textView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  Intent intent = **new** Intent(Intent.***ACTION\_VIEW***);  intent.setData(Uri.*parse*(**"http://"**+editText));  startActivity(intent);  }  });   } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 09.08.2016 | Page Number: 24 | |
| Work Performed: Share Application  I researched how to share text to other applications. I decided to make application about this. And I designed my application like this:    In this application, User can write anything to EditText. When user click the button, Application will open share panel for share another app. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 09.08.2016 | Page Number: 25 | |
| Work Performed: Share Application  I wrote my MainActivity.java like this:  **package** com.example.noah.shareapplication;  **import** android.content.Intent; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText;  **public class** MainActivity **extends** AppCompatActivity {   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **final** EditText editText = (EditText) findViewById(R.id.***editText***);  Button button = (Button) findViewById(R.id.***button***);   button.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  CharSequence mesaj = editText.getText();  ShareMassage(mesaj);  }  });  }   **private void** ShareMassage(CharSequence mesaj){  String shareBody = mesaj.toString();  Intent sharingIntent = **new** Intent(Intent.***ACTION\_SEND***);  sharingIntent.setType(**"text/plain"**);  *//sharingIntent.putExtra(Intent.EXTRA\_SUBJECT, "Başlık");* sharingIntent.putExtra(Intent.***EXTRA\_TEXT***, shareBody);  startActivity(Intent.*createChooser*(sharingIntent, **"PAYLAŞ!!"**));  }   } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 10.08.2016 | Page Number: 26 | |
| Work Performed: Mail Application  I decided to make e-mail send application today. To send an email from your application, you don’t have to implement an email client from the beginning, but you can use an existing one like the default Email app provided from Android, Gmail, Outlook, K-9 Mail etc. For this purpose, we need to write an Activity that launches an email client, using an implicit Intent with the right action and data. In this example, we are going to send an email from our app by using an Intent object that launches existing email clients.  I need e-mail address, subject and text for this application. For that, I designed my application like this: | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 10.08.2016 | Page Number: 27 | |
| Work Performed: Mail Application  I used constant type on intent for send e-mail. Like this: **message/rfc822.** Here is my MainActivity.java:  **package** com.ipadefence.app017;  **import** android.content.Intent; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText;  **public class** MainActivity **extends** AppCompatActivity {   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **final** EditText mailaddress =(EditText) findViewById(R.id.***editText***);  **final** EditText mailkonu=(EditText) findViewById(R.id.***editText2***);  **final** EditText mailicerik=(EditText) findViewById(R.id.***editText5***);  Button buton =(Button) findViewById(R.id.***button***);   buton.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  mailGondor(mailaddress.getText().toString(),mailkonu.getText().toString(),mailicerik.getText().toString());  }  });   }  **private void** mailGondor(String adress, String konu, String icerik) {  Intent mailintent=**new** Intent(Intent.***ACTION\_SEND***);  mailintent.setType(**"message/rfc822"**);  mailintent.putExtra(Intent.***EXTRA\_EMAIL***,**new** String[]{adress});  mailintent.putExtra(Intent.***EXTRA\_SUBJECT***,konu);  mailintent.putExtra(Intent.***EXTRA\_TEXT***,icerik);  startActivity(mailintent);  } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 11.08.2016 | Page Number: 28 | |
| Work Performed: SMS Send Application  I researched how to send SMS directly on application. Android has SMSManager.  You can use SMSManager API or devices Built-in SMS application to send SMS's. I used a EditText and Button on design. I wrote the code to allow send sms on manifest file.  AndroidManifest.xml:  *<?***xml version="1.0" encoding="utf-8"***?>* <**manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.noah.smssendapp"**>  <**uses-permission android:name="android.permission.SEND\_SMS"**/>  <**application ...** </**application**> </**manifest**>  My MainActivity.java is (I wrote my phone number for test):  **package** com.example.noah.smssendapp; **import** android.Manifest; **import** android.support.v4.app.ActivityCompat; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.telephony.SmsManager; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.Toast; **public class** MainActivity **extends** AppCompatActivity {   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **final** EditText editText = (EditText) findViewById(R.id.***editText***);  Button button = (Button) findViewById(R.id.***button***);  ActivityCompat.*requestPermissions*(**this**,  **new** String[]{Manifest.permission.***SEND\_SMS***}, 1);  button.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  String mesaj = editText.getText().toString();  SendSMS(mesaj);  }  });  }  **private void** SendSMS(String mesaj){   SmsManager smsManager = SmsManager.*getDefault*();  smsManager.sendTextMessage(**"+905370000000"**, **null**, mesaj, **null**, **null**);  Toast.*makeText*(getApplicationContext(),**"wow"**,Toast.***LENGTH\_LONG***).show();  } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 11.08.2016 | Page Number: 29 | |
| Work Performed: SMS Send Application  My App’s interface, Also It works perfectly:      The application sends SMS automatically. You don’t need to be confirm. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 12.08.2016 | Page Number: 30 | |
| Work Performed: GPS Application  For this application, I wrote the code to allow access the GPS location on manifest file.  AndroidManifest.xml:  *<?***xml version="1.0" encoding="utf-8"***?>* <**manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.noah.gpsapplication"**>  <**uses-permission android:name="android.permission.SEND\_SMS"**/>  <**uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"**/>  <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"**/>  <**application ...** </**application**> </**manifest**>  Also I used my send SMS app’s codes for send GPS location myself. I created a class with implement LocationListener. With this class, I can access my Latitude and Longitude.  Mylocation class:  **class** mylocation **implements** android.location.LocationListener{  @Override  **public void** onLocationChanged(Location location) {  location.getLatitude();  location.getLongitude();  }   @Override  **public void** onStatusChanged(String s, **int** i, Bundle bundle) {  }  @Override  **public void** onProviderEnabled(String s) {  }  @Override  **public void** onProviderDisabled(String s) {  } }  I have only a button on my application to do everything. | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 12.08.2016 | Page Number: 31 | |
| Work Performed: GPS Application  My mainactivity.java:  **package** com.example.noah.gpsapplication; **import** ... **public class** MainActivity **extends** AppCompatActivity {   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  Button button = (Button) findViewById(R.id.***button***);  button.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  GPS();  }  });  }  **private void** GPS() {  LocationManager locationManager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  mylocation mylocation = **new** mylocation();   **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  Toast.*makeText*(getApplicationContext(), **"permission ERROR"**, Toast.***LENGTH\_LONG***).show();  **return**;  }  locationManager.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***, 0, 0, mylocation);  Location a = locationManager.getLastKnownLocation(LocationManager.***GPS\_PROVIDER***);  a.getLatitude();  a.getLongitude();  String myLocation = **"Enlem: "** + a.getLatitude() + **"Boylam: "** + a.getLongitude();  SendSMS(myLocation);   }  **private void** SendSMS(String mesaj){   SmsManager smsManager = SmsManager.*getDefault*();  smsManager.sendTextMessage(**"+905370000000"**, **null**, mesaj, **null**, **null**);  Toast.*makeText*(getApplicationContext(),**"wow"**,Toast.***LENGTH\_LONG***).show();  } } | | |
| Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences, Practice Report | | |
| Date: 12.08.2016 | Page Number: 32 | |
| Work Performed: GPS Application  Application worked: | | |