CS 473 - MDP
Mobile Device Programming

Maharishi International University

CS 473 - MDP
Mobile Device Programming

MS.CS Program

Department of Computer Science

Renuka Mohanraj , Ph.D.

Maharishi International University

CS 473 – MDP

Mobile Device Programming

Lesson 12
Localization

Maharishi International University

Wholeness of the lesson

An android application can run on many devices in many different regions. In order to make your application more interactive, your application should handle text, numbers, files etc., in ways appropriate to the locales where your application will be used. *Similarly in Cosmic consciousness, established in inner fullness like localization in android, one acts to fulfill the needs of others and society with greater effectiveness.*

Localization

- Localization is the process of making an app available in multiple languages.
- To sell an application worldwide successfully, you must target it to a broad and diverse audience.
- English enjoys a broad acceptance and practice, and therefore you can expect that an English language application will sell well globally, but why give up on sales opportunities to the broader non-English-speaking Android market?
- In its simplest approach, this means translation into multiple languages. But there's more! Beyond language translation, an application needs to properly handle dates and times, number and currency formats, and for some applications unit of measure.
- The reasons for localizing an application are manifold. Your application may be bound for some cultural reasons to a specific region.

String resource

- Keeping your labels and other bits of text outside the main source code of your application is generally considered to be a very good idea. [strings.xml]
- It helps with Internationalization (I18N) and localization (L10N).
- Even if you are not going to translate your strings to other languages, it is easier to make corrections if all the strings are in one spot instead of scattered throughout your source code.

Android Way

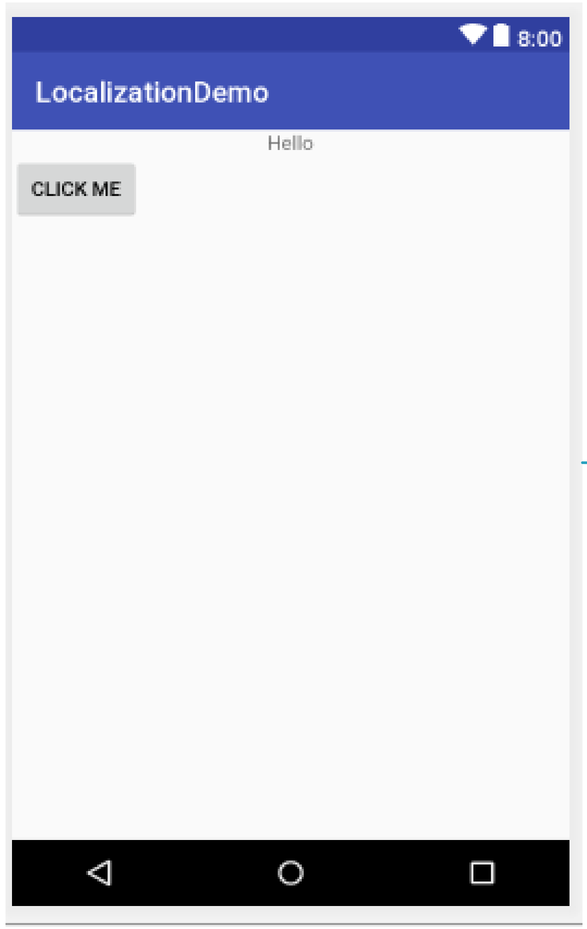
- The way Android currently handles this is by having multiple resource directories, with the criteria for each embedded in their names.
- Suppose, for example, you want to support strings in both English and Spanish.
- Normally, for a single-language setup, you would put your strings in a file named res/values/strings.xml.
- To support both English and Spanish, you would create two folders, res/values-en/ and res/values-es/, where the value after the hyphen is the ISO 639–1two-letter code for the language you want.
- Your English-language strings would go in res/values-en/strings.xml and the Spanish ones in res/ values-es/strings.xml. Android will choose the proper file based on the user’s device settings.

Android Way

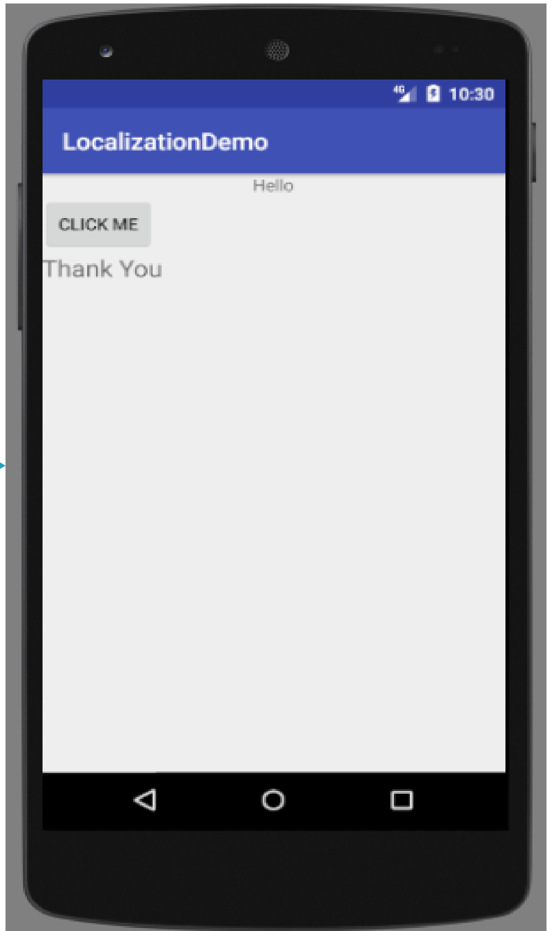
- An even better approach is for you to consider some language to be your default and put those strings in res/values/strings.xml.
- Then, create other resource directories for your translations (e.g., res/values-es/strings.xml for Spanish).
- Android will try to match a specific language set of resources; failing that, it will fall back to the default of res/values/strings.xml.
- This way, if your app winds up on a device with a language that you do not expect, you at least serve up strings in your chosen default language.
- Otherwise, if there is no such default, you will wind up with a ResourceNotFoundException, and your application will crash.

Hands on Example(English)

This for the default Language(English), you have store everything in String Resources. To make localization here we use French.

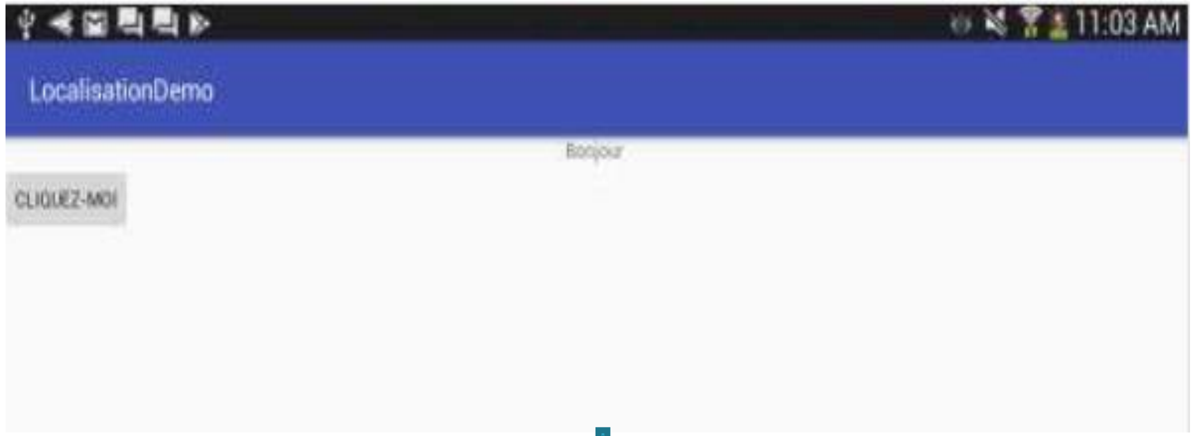


After CLICK ME



Hands on Example(France)

Once you changed the Language settings to France, you will get the following output

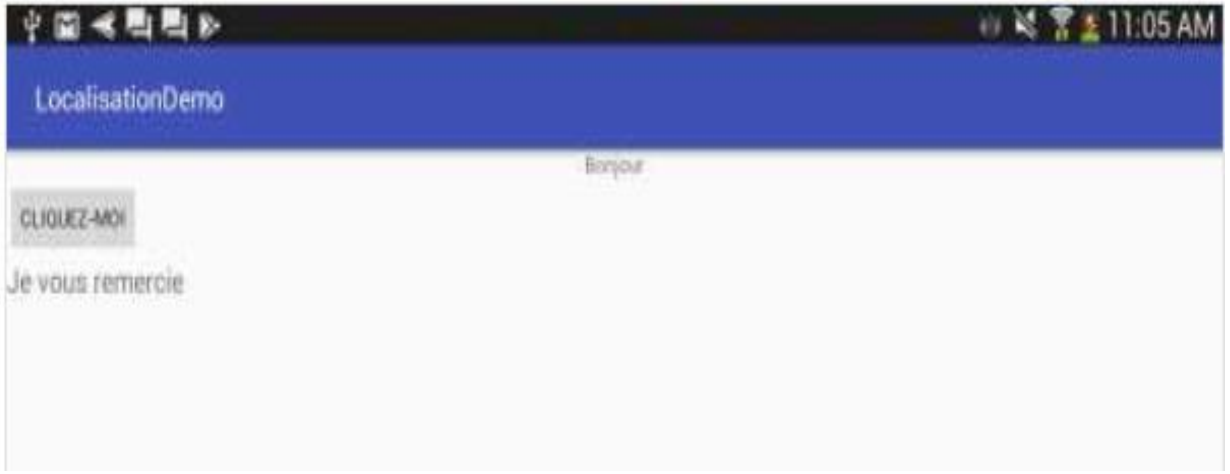


11:03 AM

LocalisationDemo

Bonjour

CLIQUEZ-MOI



11:05 AM

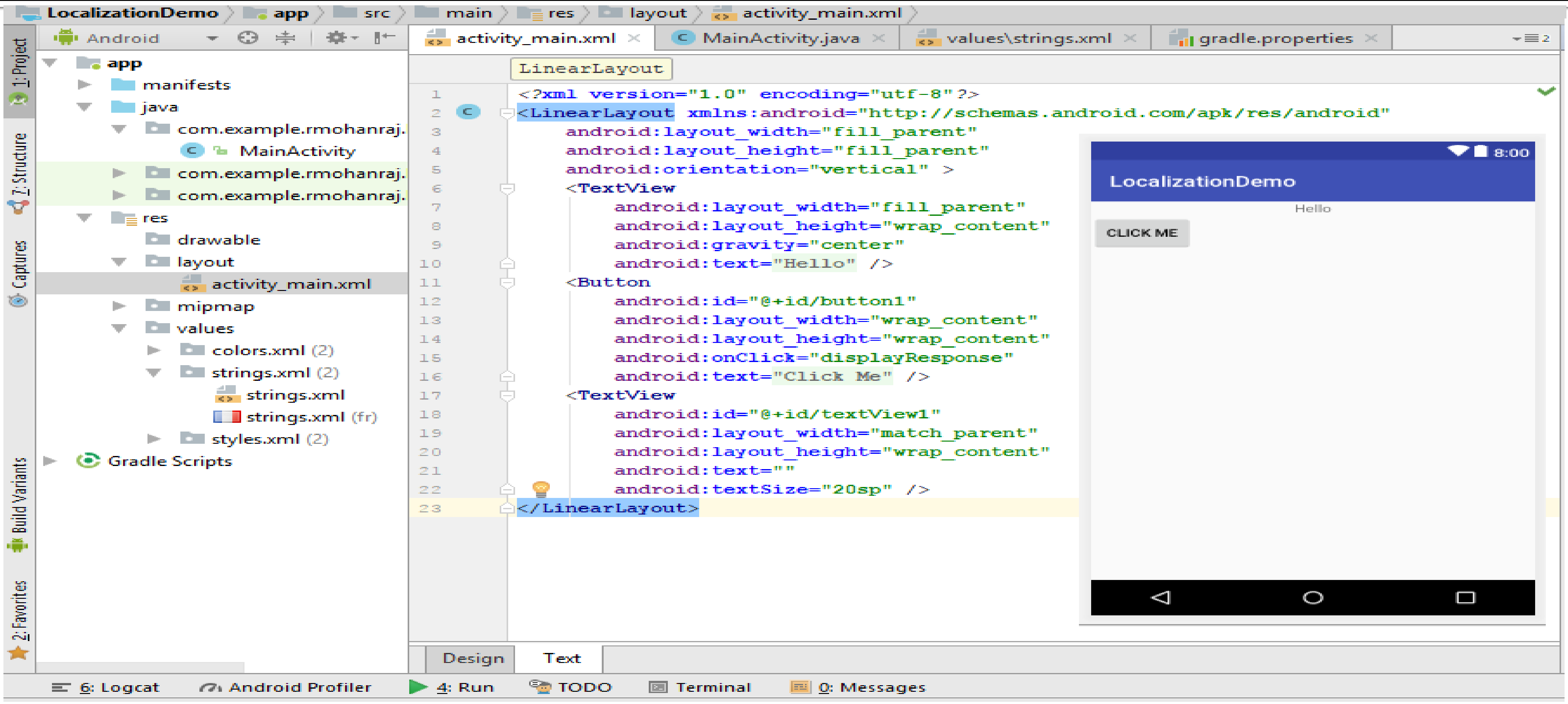
LocalisationDemo

Bonjour

CLIQUEZ-MOI

Je vous remercie

activity_main.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Hello" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="displayResponse"
        android:text="Click Me" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="20sp" />
</LinearLayout>
```

String Resources

strings.xml

```
<resources>
    <string name="app_name">LocalizationDemo</string>
    <string name="title">Localization</string>
    <string name="hello">Hello </string>
    <string name="button"> Click Me</string>
    <string name="response">Thank You</string>
</resources>
```

strings.xml(fr)

```
<resources>
    <string name="app_name">LocalisationDemo</string>
    <string name="title">Localisation</string>
    <string name="hello">Bonjour </string>
    <string name="button"> Cliquez-moi</string>
    <string name="response">Je vous remercie</string>
</resources>
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void displayResponse(View v) {
          textview1.setText(R.string.response);
    }
}
```

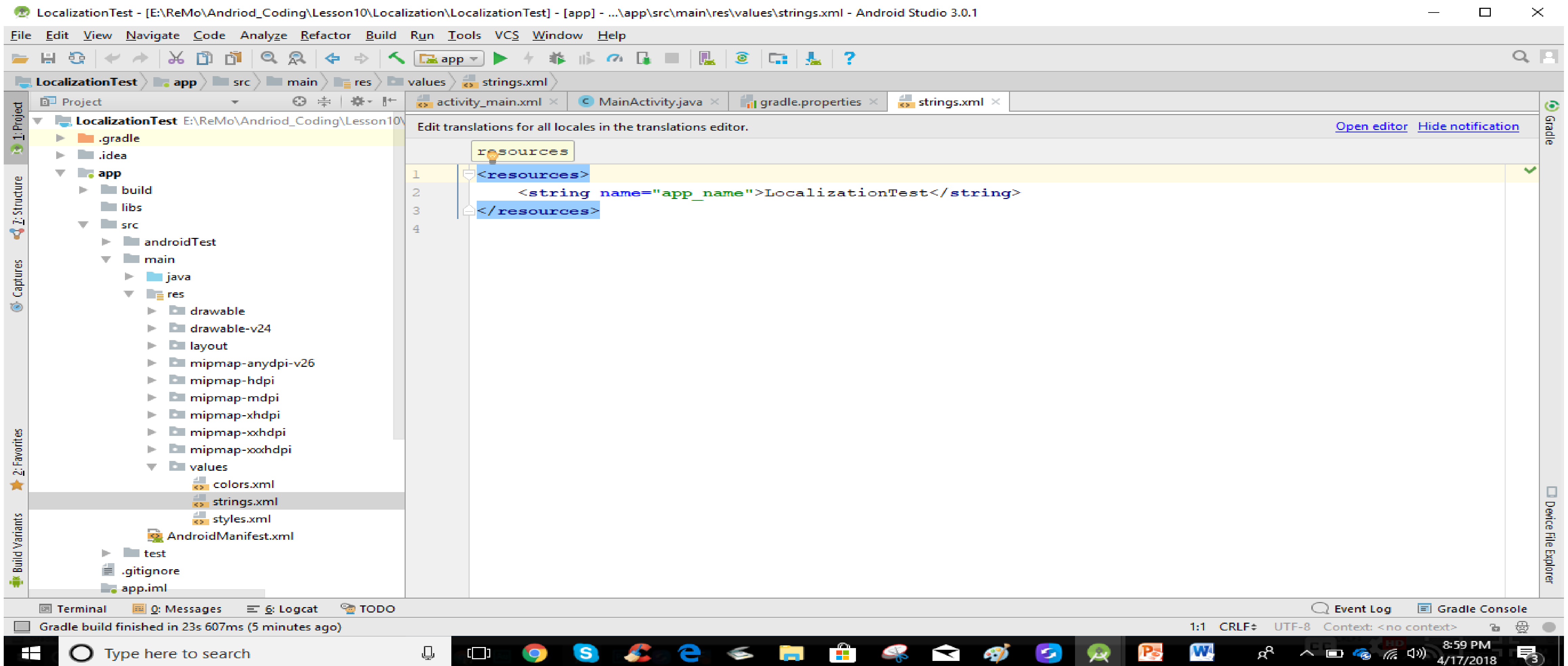
Demo Code: DaysApp, LocalizationDynammic

Main Point 1

Localization is the process of making an app available in multiple languages for the benefit of everyone. ***Science of Consciousness:*** *In the state of refined cosmic consciousness, flow of universal love for everyone and everything*

Localization Step by Step Screens

Method 1 : Enter Data into String resource directly.



LocalizationTest - [E:\ReMo\Andriod_Coding\Lesson10\Localization\LocalizationTest] - [app] - ...\app\src\main\res\values\strings.xml - Android Studio 3.0.1

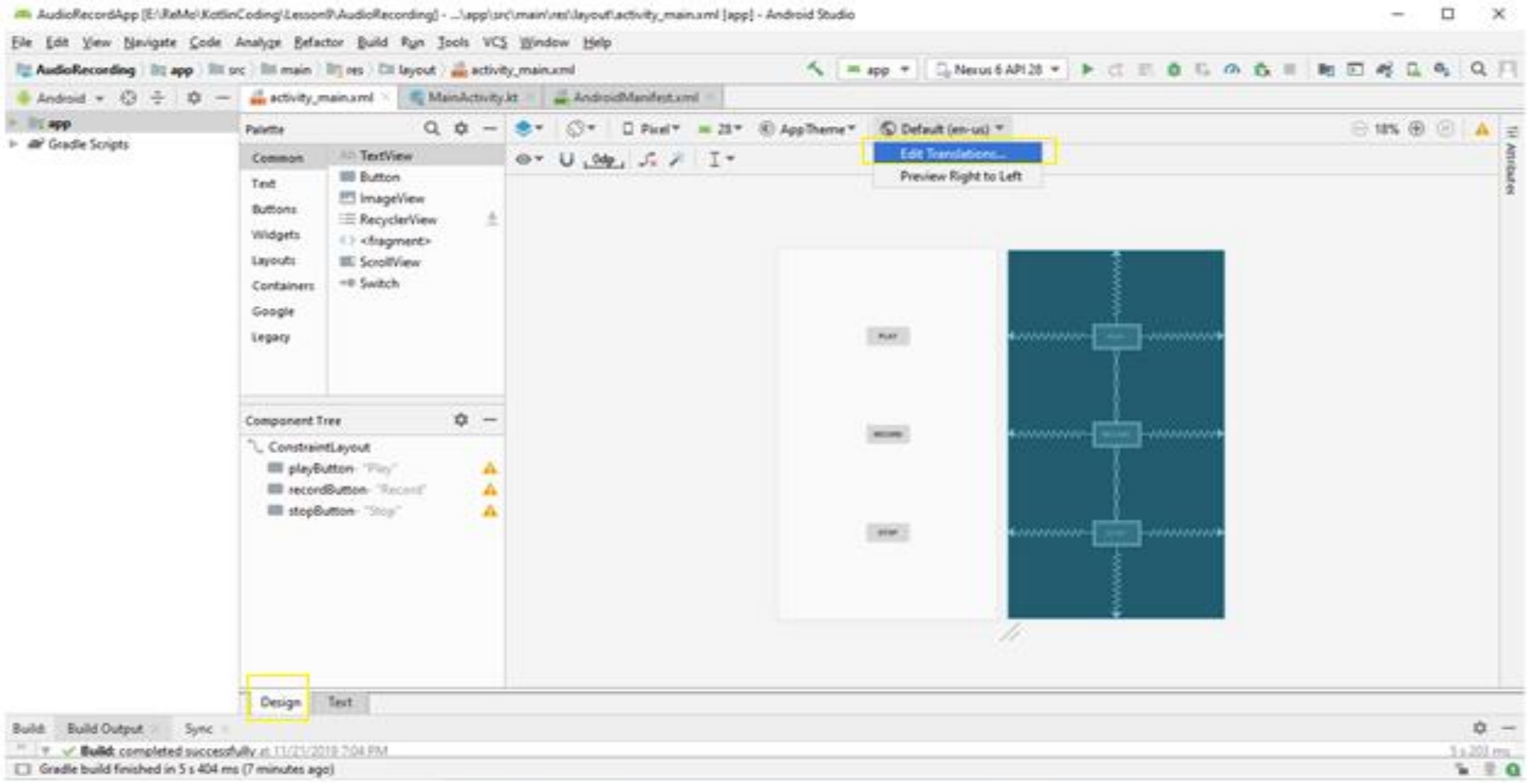
Edit translations for all locales in the translations editor.

```
<resources>
    <string name="app_name">LocalizationTest</string>
</resources>
```

Gradle build finished in 23s 607ms (5 minutes ago)

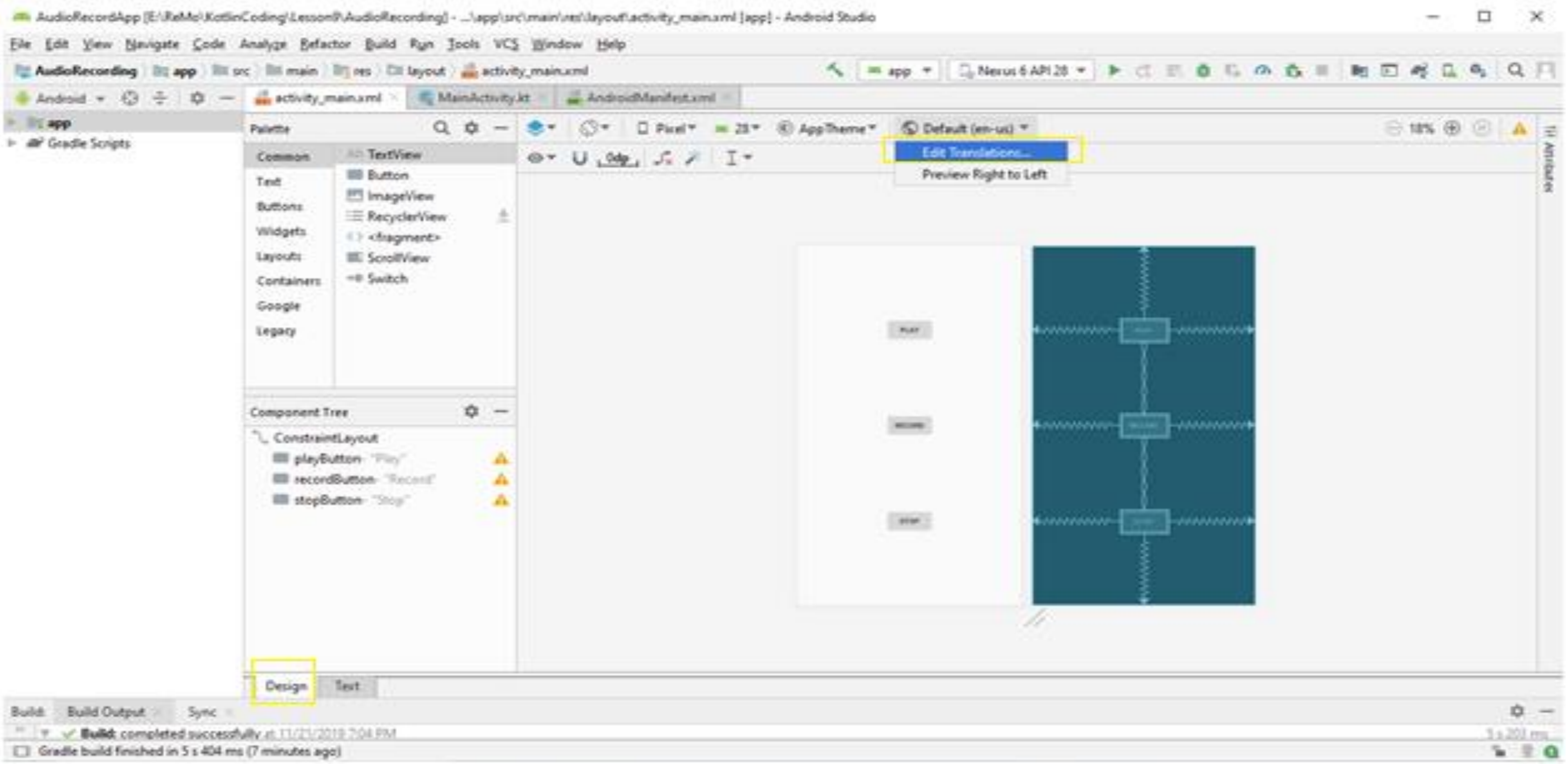
Localization Step by Step Screens

Method 2 : Step 1 : Go to your Layout Design View and Select Default(en-us) → Edit Translation



Localization Step by Step Screens

Step 2 :Click + to add more String inputs as a Key/Value Pair into your resource.

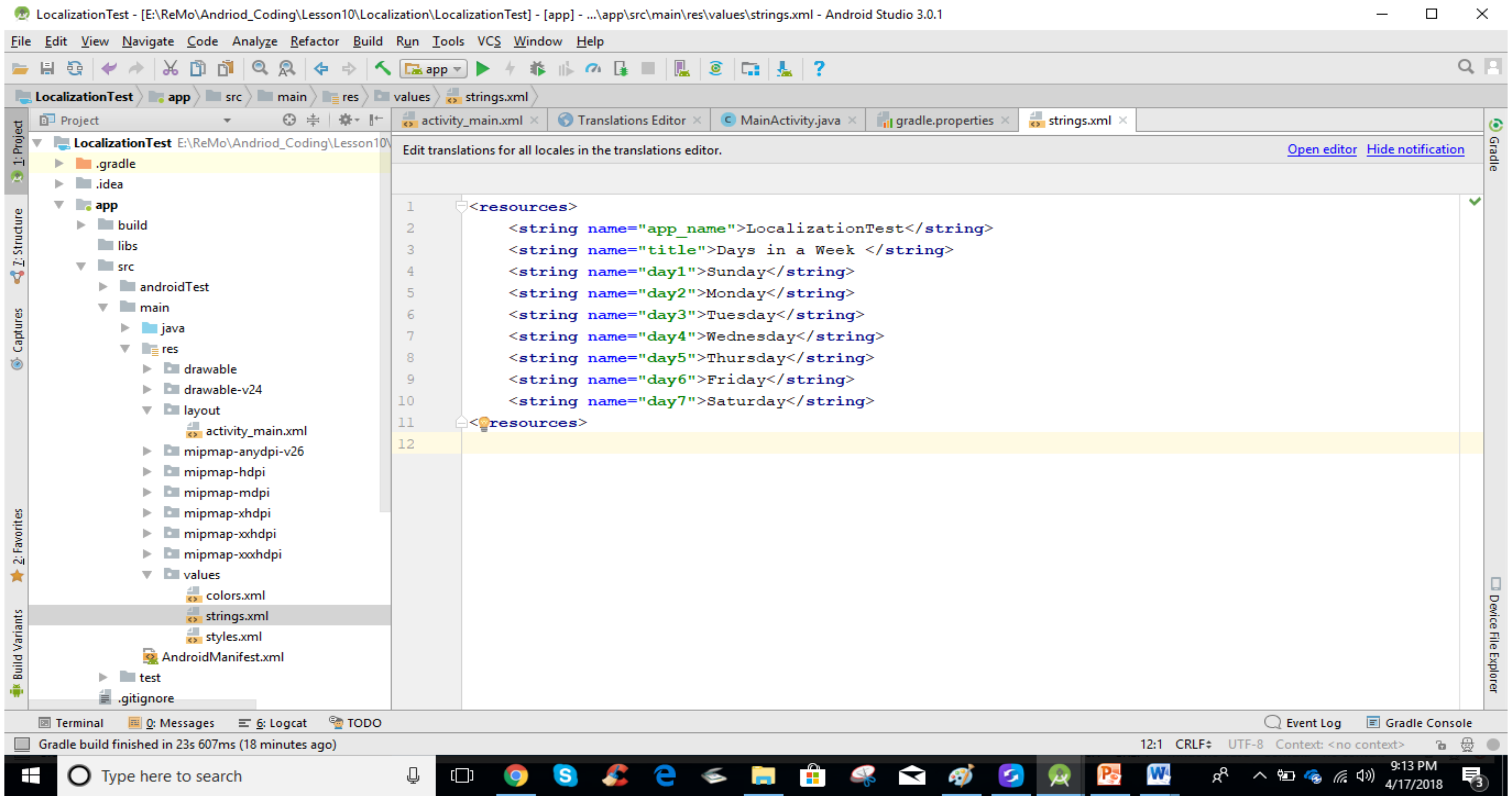


Build: completed successfully at 11/21/2019 7:04 PM

Gradle build finished in 5 s 404 ms (7 minutes ago)

Localization Step by Step Screens

Step 3 :If you click on your strings.xml, you will notice with all the values.



LocalizationTest - [E:\ReMo\Andriod_Coding\Lesson10\Localization\LocalizationTest] - [app] - ...\app\src\main\res\values\strings.xml - Android Studio 3.0.1

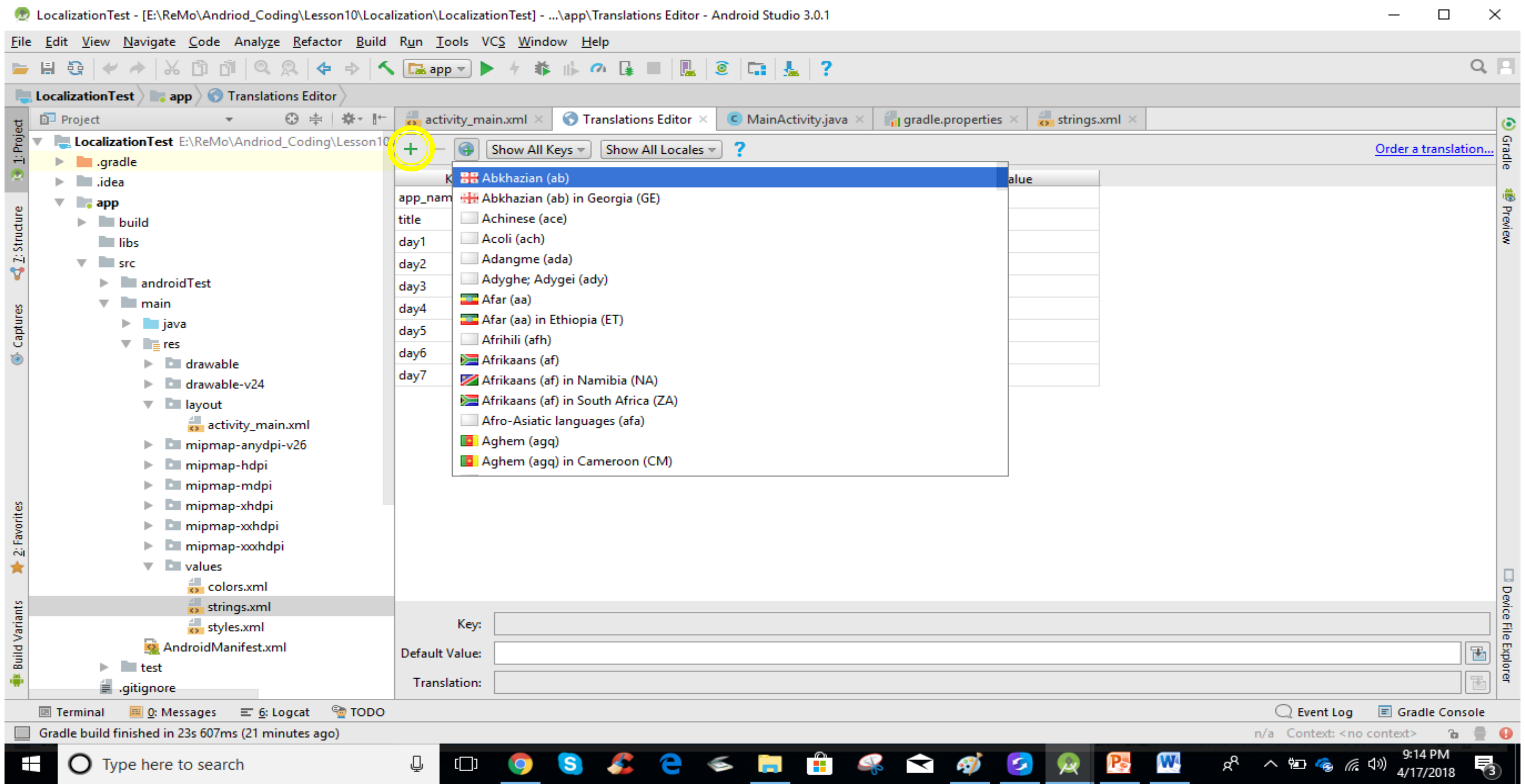
Edit translations for all locales in the translations editor.

```
<resources>
    <string name="app_name">LocalizationTest</string>
    <string name="title">Days in a Week </string>
    <string name="day1">Sunday</string>
    <string name="day2">Monday</string>
    <string name="day3">Tuesday</string>
    <string name="day4">Wednesday</string>
    <string name="day5">Thursday</string>
    <string name="day6">Friday</string>
    <string name="day7">Saturday</string>
</resources>
```

Gradle build finished in 23s 607ms (18 minutes ago)

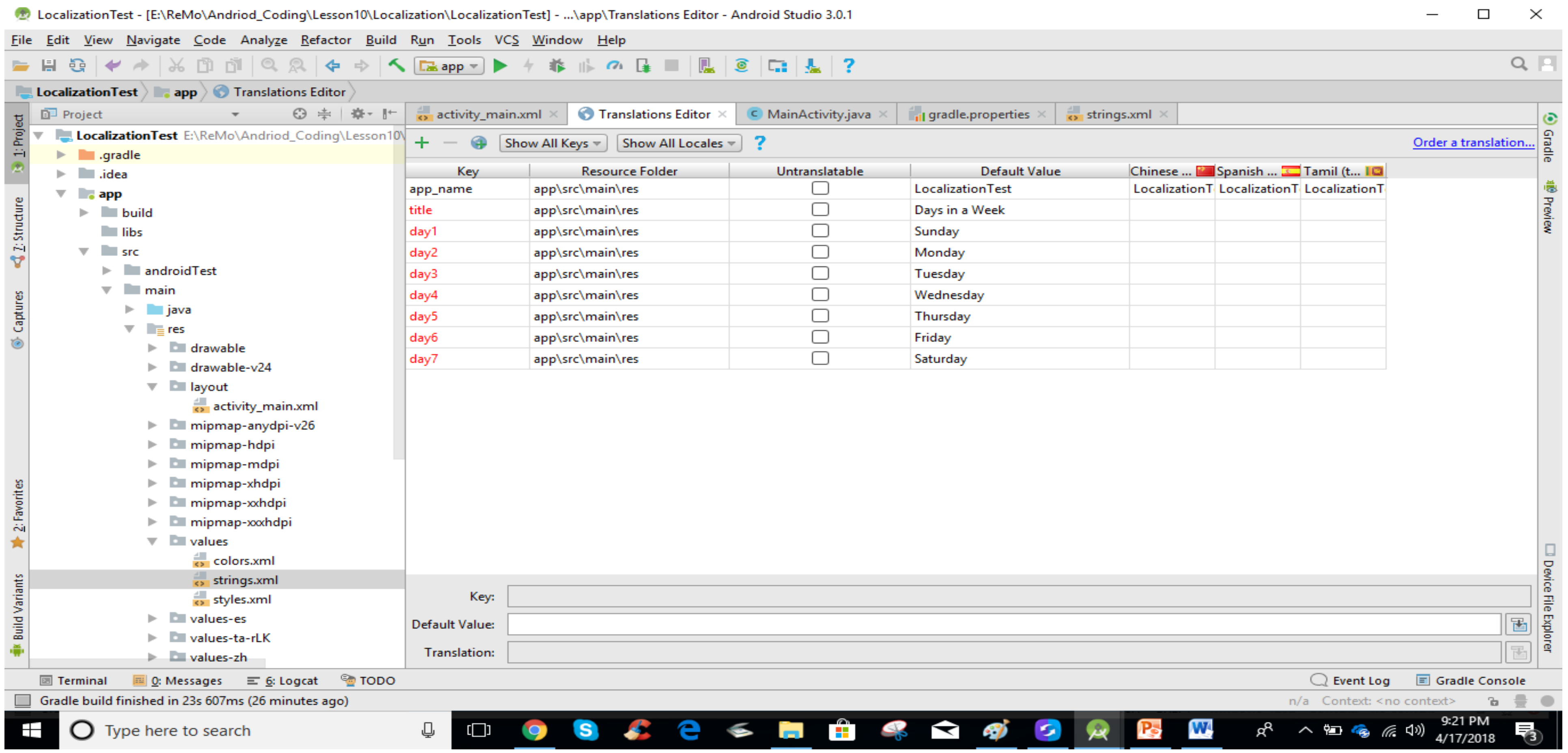
Localization Step by Step Screens

Step 4 : To add Locale support, go back to your Translation Editor and click the (+) highlighted icon and choose the desired language.


LocalizationTest - [E:\ReMo\Andriod_Coding\Lesson10\Localization\LocalizationTest] - ...\app\Translations Editor - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LocalizationTest app Translations Editor
activity_main.xml Translations Editor MainActivity.java gradle.properties strings.xml
Project
LocalizationTest E:\ReMo\Andriod_Coding\Lesson10
.gradle
.idea
app
build
libs
src
androidTest
main
java
res
drawable
drawable-v24
layout
activity_main.xml
mipmap-anydpi-v26
mipmap-hdpi
mipmap-mdpi
mipmap-xhdpi
mipmap-xxhdpi
mipmap-xxxhdpi
values
colors.xml
strings.xml
styles.xml
AndroidManifest.xml
test
.gitignore
Show All Keys
Show All Locales
Order a translation...
app_nam
title
day1
day2
day3
day4
day5
day6
day7
Abkhazian (ab)
Abkhazian (ab) in Georgia (GE)
Achinese (ace)
Acoli (ach)
Adangme (ada)
Adyghe; Adygei (ady)
Afar (aa)
Afar (aa) in Ethiopia (ET)
Afrihili (afh)
Afrikaans (af)
Afrikaans (af) in Namibia (NA)
Afrikaans (af) in South Africa (ZA)
Afro-Asiatic languages (afa)
Aghem (agq)
Aghem (agq) in Cameroon (CM)
Key:
Default Value:
Translation:
Terminal 0: Messages 6: Logcat TODO Event Log Gradle Console
Gradle build finished in 23s 607ms (21 minutes ago)
n/a Context: <no context>
Type here to search
9:14 PM 4/17/2018

Localization Step by Step Screens

Step 5 : After Choosing the Language, you will get screen looks like below. Red color indicates that need to provide terms for each Locale.



LocalizationTest - [E:\ReMo\Andriod_Coding\Lesson10\Localization\LocalizationTest] - ...\app\Translations Editor - Android Studio 3.0.1

```
LocalizationTest  E:\ReMo\Andriod_Coding\Lesson10\
  .gradle
  .idea
  app
    build
    libs
    src
      androidTest
      main
        java
        res
          drawable
          drawable-v24
          layout
            activity_main.xml
          mipmap-anydpi-v26
          mipmap-hdpi
          mipmap-mdpi
          mipmap-xhdpi
          mipmap-xxhdpi
          mipmap-xxxhdpi
          values
            colors.xml
            strings.xml
            styles.xml
          values-es
          values-ta-rLK
          values-zh
```

Key	Resource Folder	Untranslatable	Default Value	Chinese ...	Spanish ...	Tamil (t...
app_name	app\src\main\res	☐	LocalizationTest	LocalizationT	LocalizationT	LocalizationT
title	app\src\main\res	☐	Days in a Week			
day1	app\src\main\res	☐	Sunday			
day2	app\src\main\res	☐	Monday			
day3	app\src\main\res	☐	Tuesday			
day4	app\src\main\res	☐	Wednesday			
day5	app\src\main\res	☐	Thursday			
day6	app\src\main\res	☐	Friday			
day7	app\src\main\res	☐	Saturday			

Key:	
Default Value:	
Translation:	

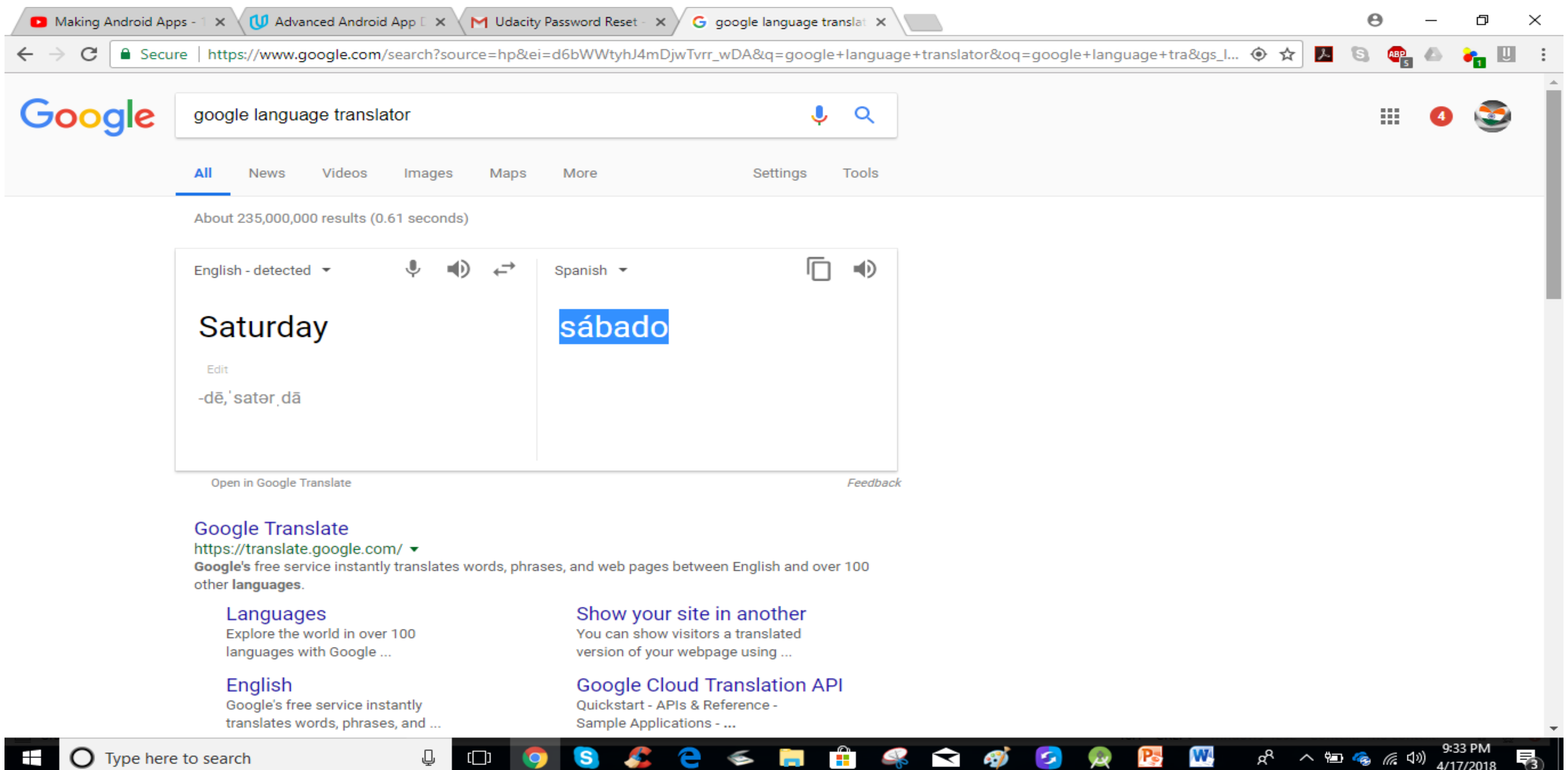
Gradle build finished in 23s 607ms (26 minutes ago)

n/a Context: <no context>

9:21 PM
4/17/2018

Localization Step by Step Screens

Step 6 :Use Google Language Translator to fill out the values in each locale or use https://translate.google.com/



Making Android Apps - | Advanced Android App | Udacity Password Reset | google language translat

Secure | https://www.google.com/search?source=hp&ei=d6bWWtyhJ4mDjwTvrr_wDA&q=google+language+translator&oq=google+language+tra&gs_l...

google language translator

All News Videos Images Maps More Settings Tools

About 235,000,000 results (0.61 seconds)

English - detected | Spanish

Saturday | sábado

Edit

-dē,ˈsatər͵dā

Open in Google Translate | Feedback

Google Translate
https://translate.google.com/
Google's free service instantly translates words, phrases, and web pages between English and over 100 other **languages**.

Languages
Explore the world in over 100 languages with Google ...

Show your site in another
You can show visitors a translated version of your webpage using ...

English
Google's free service instantly translates words, phrases, and ...

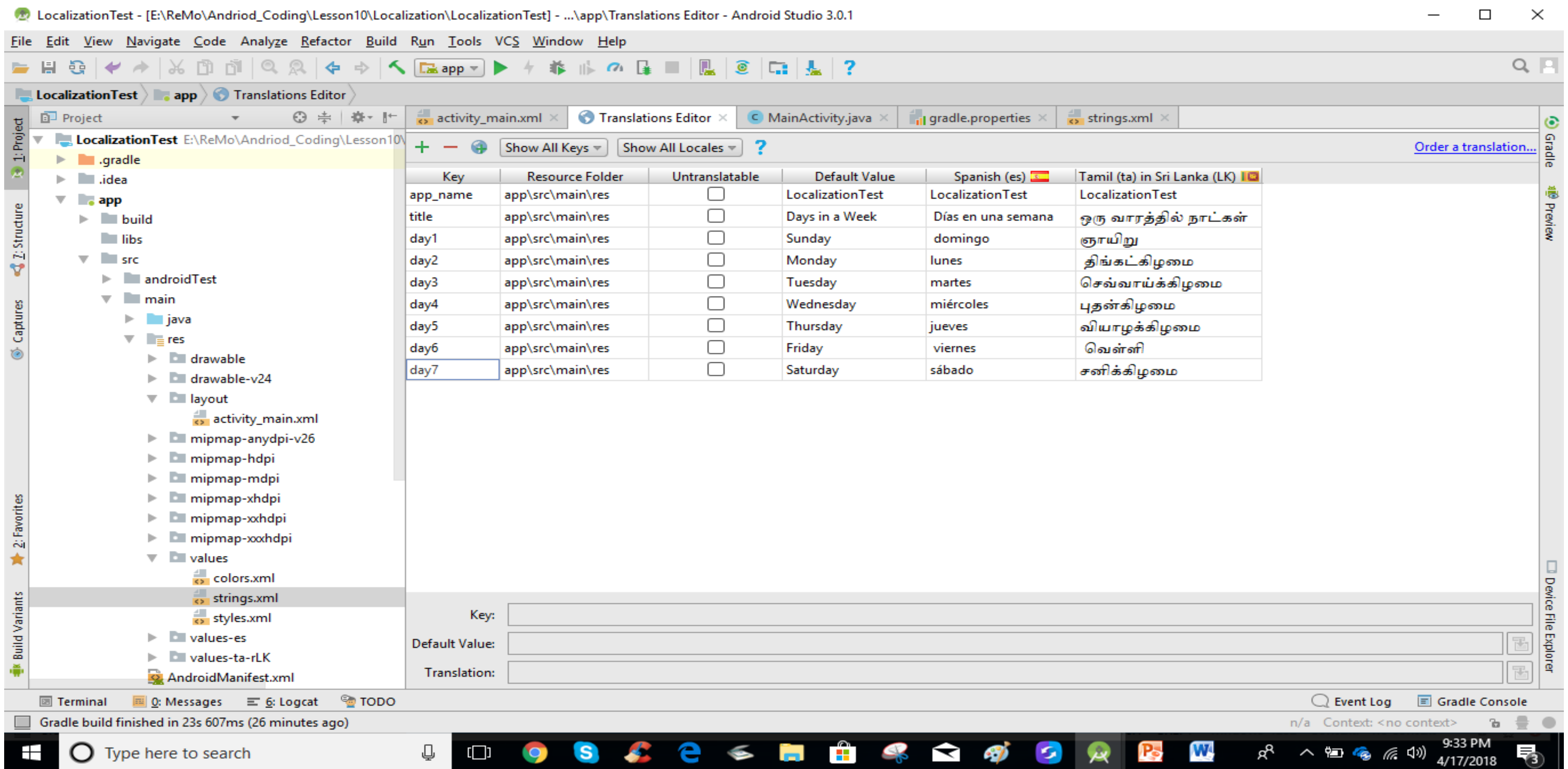
Google Cloud Translation API
Quickstart - APIs & Reference - Sample Applications - ...

Type here to search

9:33 PM
4/17/2018

Localization Step by Step Screens

Step 7 :Once everything is done, the screen looks below.



LocalizationTest - [E:\ReMo\Andriod_Coding\Lesson10\Localization\LocalizationTest] - ...\app\Translations Editor - Android Studio 3.0.1

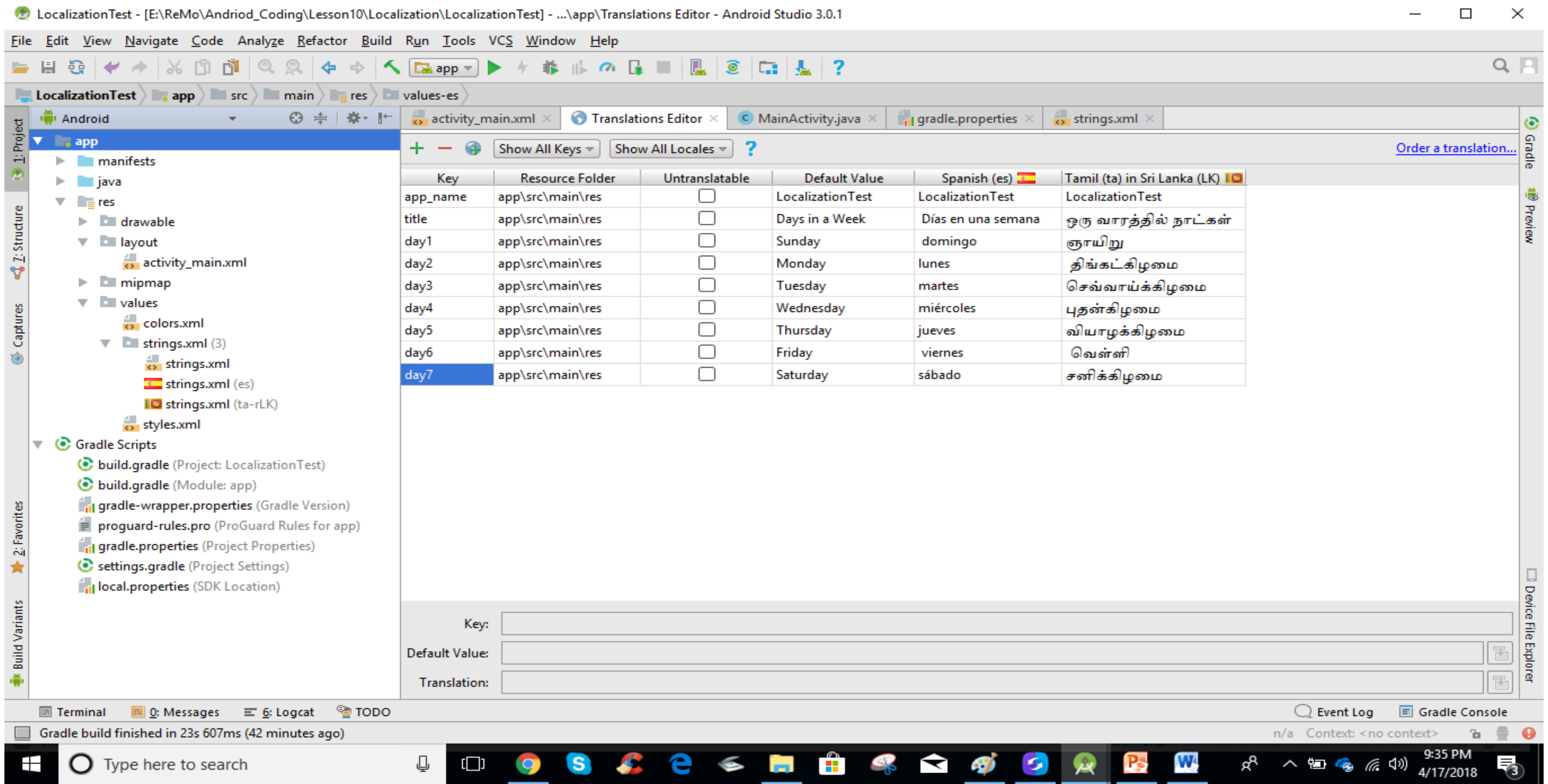
Key	Resource Folder	Untranslatable	Default Value	Spanish (es)	Tamil (ta) in Sri Lanka (LK)
app_name	app\src\main\res	☐	LocalizationTest	LocalizationTest	LocalizationTest
title	app\src\main\res	☐	Days in a Week	Días en una semana	ஒரு வாரத்தில் நாட்கள்
day1	app\src\main\res	☐	Sunday	domingo	ஞாயிறு
day2	app\src\main\res	☐	Monday	lunes	திங்கட்கிழமை
day3	app\src\main\res	☐	Tuesday	martes	செவ்வாய்க்கிழமை
day4	app\src\main\res	☐	Wednesday	miércoles	புதன்கிழமை
day5	app\src\main\res	☐	Thursday	jueves	வியாழக்கிழமை
day6	app\src\main\res	☐	Friday	viernes	வெள்ளி
day7	app\src\main\res	☐	Saturday	sábado	சனிக்கிழமை

Key:
Default Value:
Translation:

Gradle build finished in 23s 607ms (26 minutes ago)

Localization Step by Step Screens

Step 8 : Open your res -> values->strings.xml, the screen as looks as below

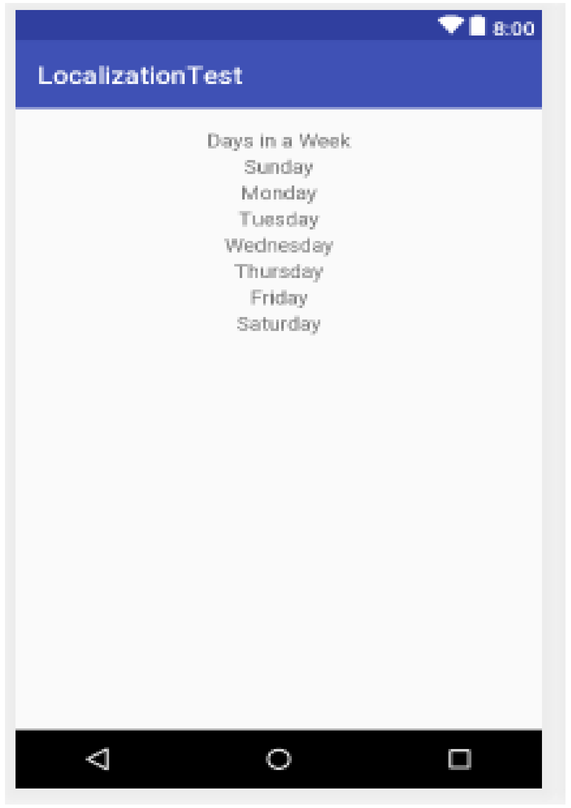

LocalizationTest - [E:\ReMo\Andriod_Coding\Lesson10\Localization\LocalizationTest] - ...\app\Translations Editor - Android Studio 3.0.1

Key	Resource Folder	Untranslatable	Default Value	Spanish (es)	Tamil (ta) in Sri Lanka (LK)
app_name	app\src\main\res	☐	LocalizationTest	LocalizationTest	LocalizationTest
title	app\src\main\res	☐	Days in a Week	Días en una semana	ஒரு வாரத்தில் நாட்கள்
day1	app\src\main\res	☐	Sunday	domingo	ஞாயிறு
day2	app\src\main\res	☐	Monday	lunes	திங்கட்கிழமை
day3	app\src\main\res	☐	Tuesday	martes	செவ்வாய்க்கிழமை
day4	app\src\main\res	☐	Wednesday	miércoles	புதன்கிழமை
day5	app\src\main\res	☐	Thursday	jueves	வியாழக்கிழமை
day6	app\src\main\res	☐	Friday	viernes	வெள்ளி
day7	app\src\main\res	☐	Saturday	sábado	சனிக்கிழமை

Gradle build finished in 23s 607ms (42 minutes ago)

Localization Step by Step Screens

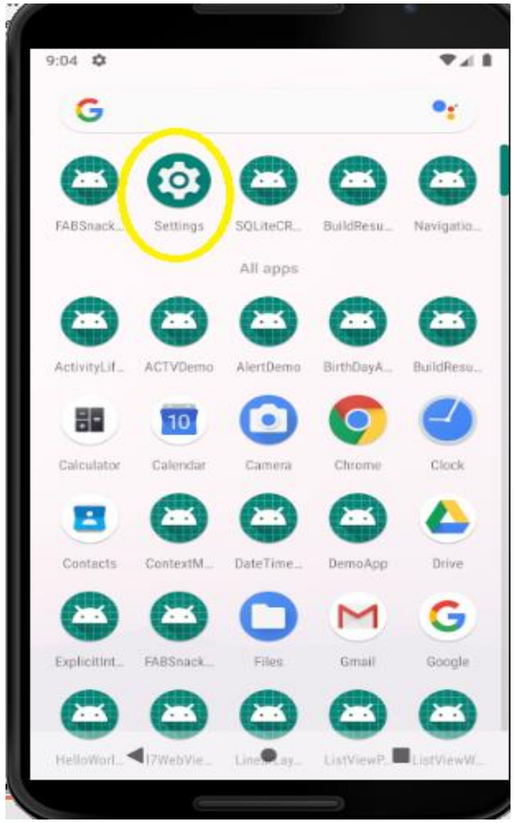
Step 9 :Go to your activity_xml and design layout as shown below. Add eight TextView components and make all component text attribute reference with its string resource.

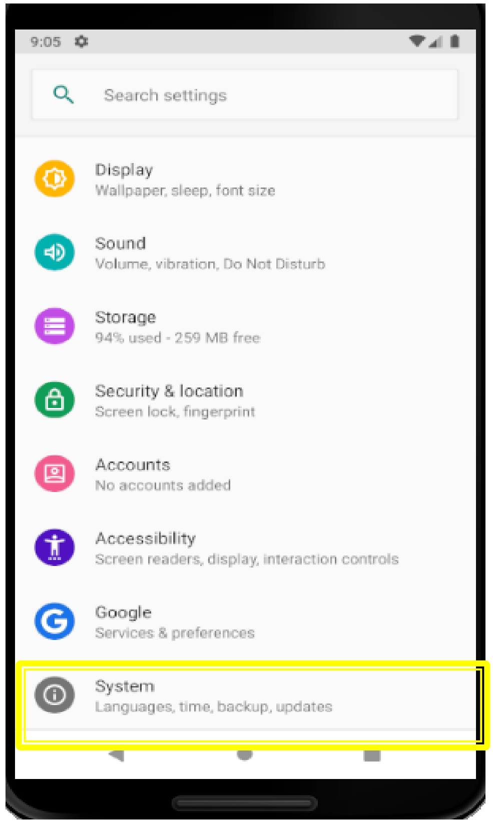


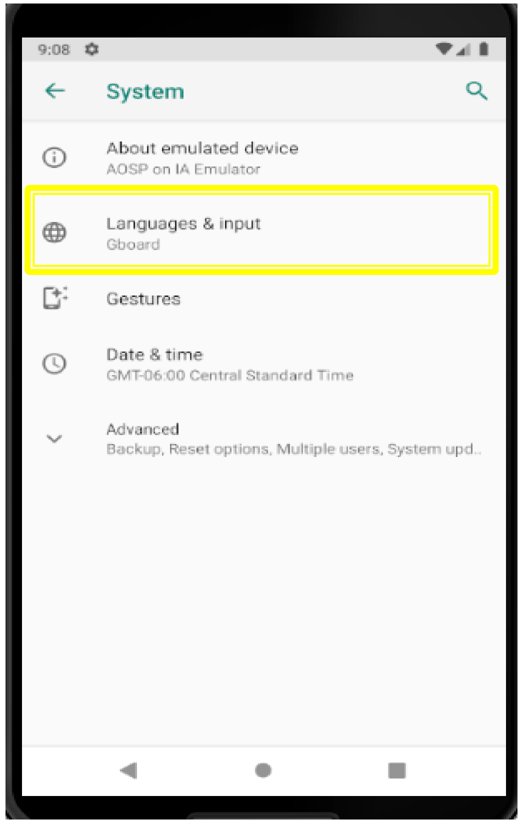
```
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:text="@string/title"
        android:textSize="16sp"/>
    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:text="@string/day1"
        android:textSize="16sp"/>
```

How to change the Language Settings

From your emulator or real device,
Choose Settings → System → Language & Input

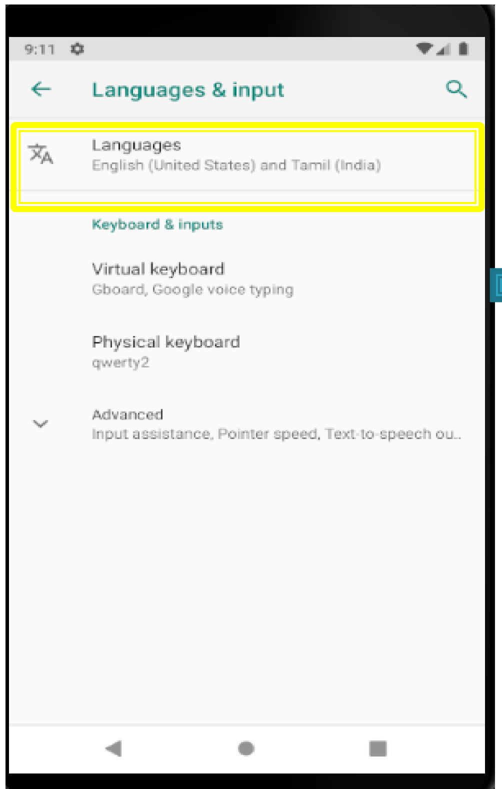






How to change the Language Settings

Continuation of Previous slide Screenshots,
Choose Languages, English is the first priority now. Drag your preferences and click Add a language to insert more languages. Make sure that, you should have the String resources for the selected language in your app.



9:11

Languages & input

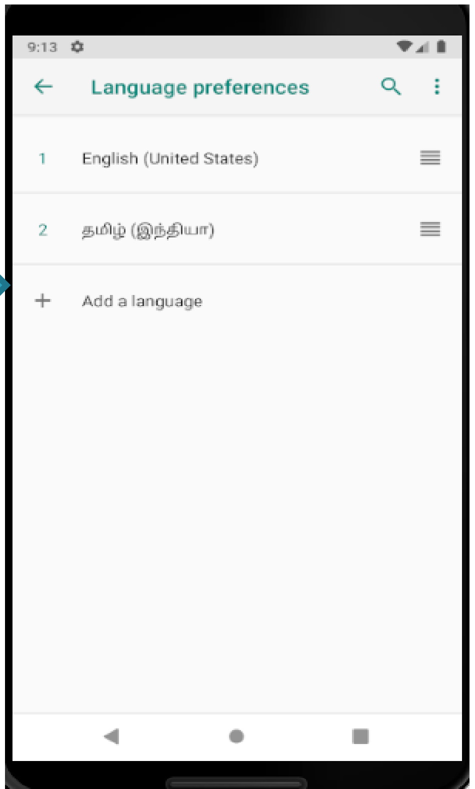
Languages
English (United States) and Tamil (India)

Keyboard & inputs

Virtual keyboard
Gboard, Google voice typing

Physical keyboard
qwerty2

Advanced
Input assistance, Pointer speed, Text-to-speech ou..



9:13

Language preferences

1 English (United States)

2 தமிழ் (இந்தியா)

+ Add a language

Online tools to get multilanguage strings.xml resource file

- Automatic String Resource Translation
- Upload the original strings.xml file and get the desired language.

- https://asrt.gluege.boerde.de/

Change Language settings Programmatically

- To get localized data one should use Resources with the desired Locale set in their Configuration.
- Update the resources via updateConfiguration with a config that includes the desired locale display metrics.
- Recreate your activity

Code

```
// get the resource's current configuration.
 val config = resources.configuration
 //Construct a locale from language and country.
        val locale = Locale(code)
        Locale.setDefault(locale)
        config.locale = locale
  // Update new locale settings
   resources.updateConfiguration(config, resources.displayMetrics)
        recreate()
```

Main Point 2

In order to localize the strings used in your application, make a new folder under res with name of values-local where local would be the replaced with the region. Start by adding values directories to your resources so that Android knows you have localized your app completely to your broad range of audience. ***Science of Consciousness:*** *Similarly in Cosmic consciousness one can feel the completeness of inner and outer life*. The experience that activity is going on by itself, with the doer, the self-uninvolved.

UNITY CHART

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

Outer depends on inner

1. To build multilingual Android apps you need to collect the texts into resource files and translate them.
2. You can provide any resource type that is appropriate for the language and culture of your users. The right thought at the right time for full effectiveness without strain.

---

3. **Transcendental Consciousness**: TC is the home of all knowledge. The Upanishads declare "Know that by which all else is known" – this is the field of pure consciousness.
4. ***Impulses within the Transcendental field:*** *The infinite diversity of these impulses within the transcendental field, is possible due to the infinite creativity and intelligence.*
5. ***Wholeness moving within Itself****: In Unity Consciousness, one sees that the "key" to accessing complete knowledge of any object is the infinite value of that object, pure consciousness, which is known in this state to be one's own Self.*