

IEMS 5722
Mobile Network Programming and Distributed Server Architecture
2014-2015 Semester 2

Assignment 1: User Interaction in Android

Due Date: 29th January, 2014 (Thursday)

Notes:

- i.) See the instructions at the end of this assignment, follow them to submit your files for marking
- ii.) Late submissions will receive 30% mark penalty

1. Aim

To understand how Android apps handles user input and how to display output to communicate with the user.

2. Objectives

- Create an Android app to translate inputted text
- Handle user input from text fields and buttons
- Translate the input text and display the results back to the user
- Notify the user through toasts and dialogs

3. Instructions

3.1 App set up

Download the **translateapp.zip** from the following Github URL and add the project to your workspace. The app contains the basic hello world app, and a **WordDictionary.java** which serves as the translate service in this exercise.

URL: <https://github.com/IEMS5722-Spring2015/TranslateApp>

The translate dictionary is hardcoded as a Java **HashMap** for simplicity. The dictionary allows you to translate English numbers to Chinese characters. For example, “three” becomes “三”. Read Java references to learn how to use a hashmap in case you have not used it before. The dictionary hashmap is shown below (excerpt from WordDictionary.java):

```
public Map<String, String> wordDict = new HashMap<String, String>();
public WordDictionary() {
    wordDict.put("zero", "0");
    wordDict.put("one", "一");
    wordDict.put("two", "二");
    wordDict.put("three", "三");
    wordDict.put("four", "四");
    wordDict.put("five", "五");
}
```

4. Tasks

4.1 Create the User Interface

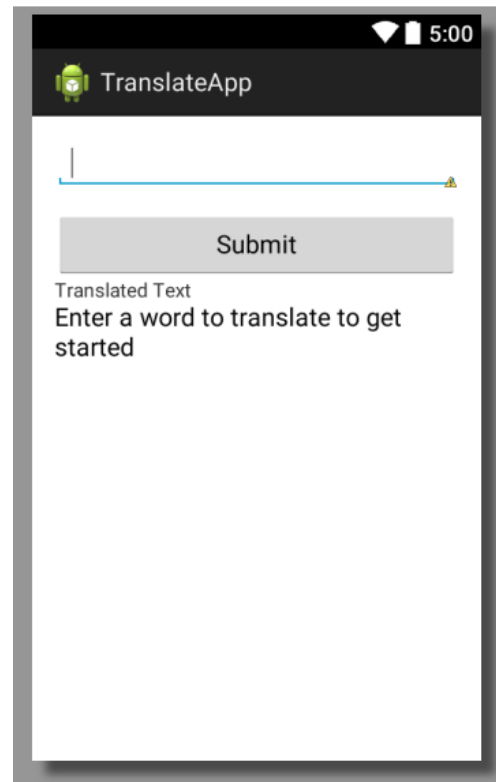
The **graphical user interface (GUI)** handles interaction from the user such as scrolling, clicking and typing. The app can also respond to the user by showing a message for example.

For this app, the UI **must contain** (at a minimum)

- Text field for user input (EditText)
- Button for the user to submit the form (Button)
- Text field showing the translated text (TextView)

An example can be seen on the right.

Hint - Use the graphical layout to see what items can be placed on the form, then use the XML view to modify everything else. (You are advised to write XML codes to create the UI so as to familiarize yourself with the Android layout.)



4.2 Handle User Input

The app must be notified when a user interacts with it in order for it to do something. For example, an app can set an **OnClickListener** on a **Button**, which will then call a function whenever that button is clicked.

Hint - Your activity will need to find a reference to the button first. It will also need a reference to the input text to get the value the user entered. You can do this in the **onCreate()** function of the activity.

4.3 Give Feedback to the User

After your translation has been completed, the output should be displayed to the user. For our app to be more useful, the user should also be notified when something has not worked. For example, the user may not have entered any text or the translation function failed.

For this app, the feedback should:

- **Show the translated text** in an output text field
- **Toast the user a message** if they have not entered anything

As a **bonus**, a **dialog box** with an error message should be displayed if the text cannot be translated.

4.4 Sharing information (Bonus Goal)

Modern apps rarely work in isolation and users have an expectation that information can be shared between apps. In Android, the communication between apps is handled via **Intents**, which are attributed with an **Action**. The Android OS knows which of the installed apps are able to handle that Action, and the user is then prompted to choose which app to use if there is more than one. Intents can optionally be loaded with text or other arbitrary binary data among others.

In our app, we would like to **share the translated text** to other apps using the **Intent** method. This will also require an extra **Share** button.

5. Submission

To submission your assignment, create a folder with a name in the following format:

<your_student_id>_assgn1

Copy the follow materials into the folder you created:

- The **src** folder (including all Java source code files)
- The **res** folder (including all the sub-folders and files)
- The **AndroidManifest.xml** file

Compress the folder into a .zip file, and submit it in the CUHK eLearning System online:

<https://elearn.cuhk.edu.hk/>

6. Resources

Java Programming

- The Java Tutorial
<http://docs.oracle.com/javase/tutorial/>
- Java HashMap Tutorial
http://www.tutorialspoint.com/java/java_hashmap_class.htm

Android Programming

- Layouts:
<http://developer.android.com/guide/topics/ui/declaring-layout.html>
- Input Controls:
<http://developer.android.com/guide/topics/ui/controls.html>
- Input Events:
<http://developer.android.com/guide/topics/ui/ui-events.html>
- Toast Messages:
<http://developer.android.com/guide/topics/ui/notifiers/toasts.html>
- Dialog Messages:
<http://developer.android.com/guide/topics/ui/dialogs.html>
- Intent and Sharing:
<http://developer.android.com/training/sharing/send.html>