IEMS 5722

Mobile Network Programming and Distributed Server Architecture

2014-2015 Semester 2

Assignment 4: Implementing Your Own Server

Due Date: 2nd April, 2015 (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 the interactions between clients and servers in a mobile context by implementing a Web service that serves an Android app

2. Objectives

- Implement the translation Web service using either **Python** or **Node.js**
- Modify the **translation app** user interface to make it like an instant messaging app
- Extend the translation app so that it can receive input from other **third party apps**

3. Instructions

You are encouraged to use your app submitted in Assignment 2 as the starting point for this this assignment. If you do not wish to, you may use the solution to Assignment 2 from GitHub at https://github.com/IEMS5722-Spring2015/A2Solution

4. Tasks

4.1. Server Implementation

In the previous assignments, the server was a black box service. This time, you will need to implement the server by yourself. You may use either **Python** or **Node.js** and connection can be through either **HTTP** or **TCP**. The following is a list of requirements of your server program.

1. Your server must be able to translate the words as given in the WordDictionary from Assignment 1. In addition, this implementation must be able to handle **multiple space-separated words**. For example “one two three” should be translated to “一 二 三”.

2. The server should communicate with the client using messages encoded in **JSON**. The server response will feature two fields: “output” and “message”. For successful translations, the output will be the translated text and the message should say “OK”. For failed translations, the output will be an empty string and the message should give the reason for the failure

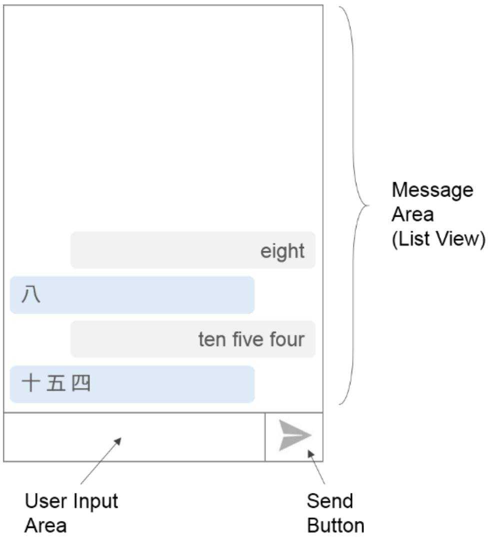
3. Your Android app should be updated to make use of the extra information. If the output is non-empty, your app should show the result of the successful translations. If the output is empty, then your app should alert the user using the message provided by the server.

4. Your server can either receive requests from the client via **HTTP GET** or by listening for **TCP** connection from the client to receive the input. You only need to implement one method.

4.2. User Interface

You should modify the user interface of the translation app and make it into something like a chat or instant messaging interface. The figure on the right shows an example.

The user interface should have a **list view** for displaying the inputs of the user and the translation responses from the server. User inputs should be right-aligned, while server responses should be left-aligned. At the bottom, there should be an “Edittext” for user input and a button for the user to send the input to the server. You should implement the interface such that **no action** should be taken when the user input is **empty**.



4.3. Receiving Data from Other Apps

In addition, you should create an **intent filter** for your activity, such that it will be able to receive textual content shared from another app in the smartphone. Once your app is launched by the Android system, you should extract the text sent from the other app, and send a request to the server for a translation.

Hint: You should create an intent filter that checks for the **ACTION_SEND** intent, which filters for “**text/plain**” MIME type messages.

5. Bonus Goal

To challenge yourself, build a translation **cache** in your app, so that when the user inputs a text that has already been translated before, the translation will be retrieved from the cache, and **NO requests** will be sent to the server. You can implement this by creating a **SQLite database** in your app to store input-output pairs, and query this database before a request is made to the server.

6. Submission

To submission your assignment, create a folder with a name in the following format:
<your_student_id>_assgn4

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
- A folder containing your **server-side source codes**

Compress the folder into a .zip file, and submit it in the CUHK eLearning System online:
https://elearn.cuhk.edu.hk/

7. References and Resources

Python

- Dive into Python (Free e-book)
  http://www.diveintopython.net/
- Python Programming on Wikibooks
  http://en.wikibooks.org/wiki/Python_Programming
- Socket Server
  https://docs.python.org/2/library/socketserver.html
- Flask (A Python Web Framework)
  http://flask.pocoo.org/

Node.js

- Node School (Tutorials of Node.js)
  http://nodeschool.io/
- Express (A Web Framework for Node.js)
  http://expressjs.com/

Using SQLite Database in Android

- Saving data in SQL databases in Android
  http://developer.android.com/training/basics/data-storage/databases.html