ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 5

Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your Android project as follows:
   a. From eclipse, choose *"Export…"* from the File menu.
   b. From the Export window, choose *General* then *File System*. Click *Next*.
   c. Make sure that your Android project for this assignment is selected. Make sure that all of its subfolders are also selected.
   d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
   e. When exporting make sure you select *Create directory structure for files*.
   f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
   a. When you submit the assignment, compress your exported Android project into a single zip file. The format of compressed file name is InClassAssignment#.zip
   b. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 5 (100 Points)

In this assignment you will make simple HTTP requests and will parse simple XML data.

Notes:

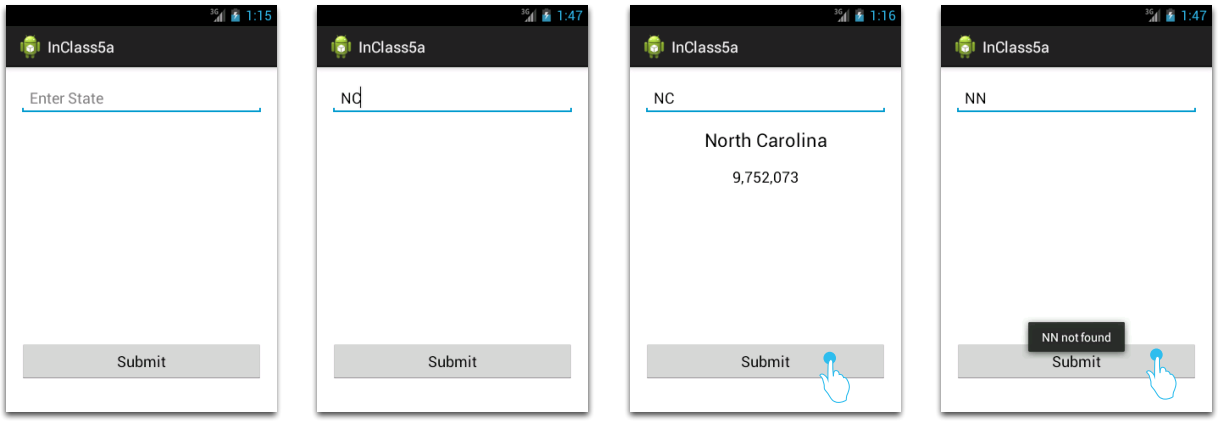
1. The recommended Android Virtual Device (AVD) should have minimum SDK version set to 11 and target SDK at least 17. The app should display correctly on 3.2" QVGA (ADP2) (320x480: mdpi).

Initial Setup and API Description

API for getting the population of a specific state (text):

- *Endpoint*: http://cci-webdev.uncc.edu/~mshehab/api-rest/states/index.php
- *Arguments*: (GET method)
  - *method*: should be set to *getStatePopulation*.
  - *state* : should be set to the 2 letter state abbreviation.
- Description: This API will return the population of the state provided. A successful output will be as follows: For example, submitting a request to get the population of NC: http://cci-webdev.uncc.edu/~mshehab/api-rest/states/index.php?method=getStatePopulation&state=NC. Note that the result is the state name and the population delimited by ";".

```
North Carolina;9,752,073
```



(a) Launching screen | (b) State abbreviation entered | (c) State population retrieved and displayed | (d) Retrieving an incorrect state entry

Figure 1, Application User Interface (Part 1)

Part 1 (50 Points): Using Simple HTTP Requests and String Response

The interface should be created to match the user interface (UI) presented in Figure 1. You will be using layout files, and strings.xml to create the user interface. Perform the following tasks:

- The EditText that will hold the state abbreviation should be limited to accept at

most 2 characters. See Figure 1(b).

- You should use a separate thread to perform the retrieval and data retrieval from the server. Do not use the Main Thread to perform these tasks. Use an AsyncTask or a Thread/Handler. The recommended method for calling the APIs is through Apache's **HttpClient** class (See the "Get Params" example in the class notes). Use the API described in t
- When the "Submit" button is tapped, the corresponding GET request should be sent to the server to retrieve the state population data. If the population data is returned correctly then display the state name and the population in TextViews as indicated in Figure 1(c). If the request returns an error, generate a toast to indicate this error as indicated in Figure 1(d).

XML API for getting the population of a specific state:

- *Endpoint*: http://cci-webdev.uncc.edu/~mshehab/api-rest/states/index.php
- *Arguments*: (GET method)
  - *method*: should be set to *getStatePopulation*.
  - *state* : should be set to the 2 letter state abbreviation.
  - *format*: should be set to XML.
- Description: This API will return the population of the state provided. A successful output will be as follows: For example, submitting a request to get the population of NC: http://cci-webdev.uncc.edu/~mshehab/api-rest/states/index.php?method=getStatePopulation&state=nc&format=XML

```
<result error="">
    <state name="North Carolina">
        <population>9,752,073</population>
    </state>
</result>
```

On Error:
http://cci-webdev.uncc.edu/~mshehab/api-rest/states/index.php?method=getStatePopulation&state=NNC&format=XML

```
<result error="NNC Not Found" />
```

Part 2 (50 Points): Using Simple HTTP Requests and XML Response

Implement the app in Part 1 but instead use the above XML API to retrieve the state population and name.