

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. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Export your Android project and create a zip file which includes all the project folder and any required libraries.
6. Submission details:
 - a. Only a single group member is required to submit on moodle for each group.
 - b. The file name is very important and should follow the following format:
Group#_InClass05.zip
 - c. You should submit the assignment through Moodle: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 5 (100 Points)

In this assignment you will get familiar with Android Concurrency, HTTP connections and XML parsing. You will build a simple weather application.

Initial Setup and API Description

You should use the OpenWeatherMap api (<http://openweathermap.org>) for getting the weather information. The api of interest is the Current Weather Data api which is based on the city name and country name. For information related to the api please check <http://openweathermap.org/forecast>. The API details is as follows:

- Endpoint: <http://api.openweathermap.org/data/2.5/forecast>
- Arguments (GET Method)
 - q: this is the city name and country name for example, Charlotte,US.
 - units: should either be set to imperial or metric.
 - cnt: should be set to 8, for eight 3-hour forecasts in a full day.
 - mode: should be set to xml.

For example to retrieve the weather for Charlotte in imperial format, the url should be setup as follows:

[http://api.openweathermap.org/data/2.5/forecast?
q=Chicago&mode=xml&cnt=8&units=imperial](http://api.openweathermap.org/data/2.5/forecast?q=Chicago&mode=xml&cnt=8&units=imperial)

The response will be as follows:

```
<weatherdata>
  <location>
    <name>Chicago</name>
    <type/>
    <country>US</country>
    <timezone/>
    <location altitude="0" latitude="41.850029" longitude="-87.650047" geobase="geonames" geobaseid="0"/>
  </location>
  <credit/>
  <meta>
    <lastupdate/>
    <calctime>0.0128</calctime>
    <nextupdate/>
  </meta>
  <sun rise="2015-09-27T11:43:08" set="2015-09-27T23:39:20"/>
  <forecast>
    <time from="2015-09-27T00:00:00" to="2015-09-27T03:00:00">
      <symbol number="802" name="scattered clouds" var="03n"/>
      <precipitation/>
      <windDirection deg="107" code="ESE" name="East-southeast"/>
      <windSpeed mps="8.23" name="Fresh Breeze"/>
      <temperature unit="imperial" value="63.03" min="63.03" max="69.56"/>
      <pressure unit="hPa" value="1011.32"/>
      <humidity value="57" unit="%"/>
      <clouds value="scattered clouds" all="32" unit="%"/>
    </time>
    <time from="2015-09-27T03:00:00" to="2015-09-27T06:00:00">...</time>
    <time from="2015-09-27T06:00:00" to="2015-09-27T09:00:00">...</time>
    <time from="2015-09-27T09:00:00" to="2015-09-27T12:00:00">...</time>
    <time from="2015-09-27T12:00:00" to="2015-09-27T15:00:00">...</time>
    <time from="2015-09-27T15:00:00" to="2015-09-27T18:00:00">...</time>
    <time from="2015-09-27T18:00:00" to="2015-09-27T21:00:00">...</time>
    <time from="2015-09-27T21:00:00" to="2015-09-28T00:00:00">...</time>
  </forecast>
</weatherdata>
```

Figure 1, XML Response

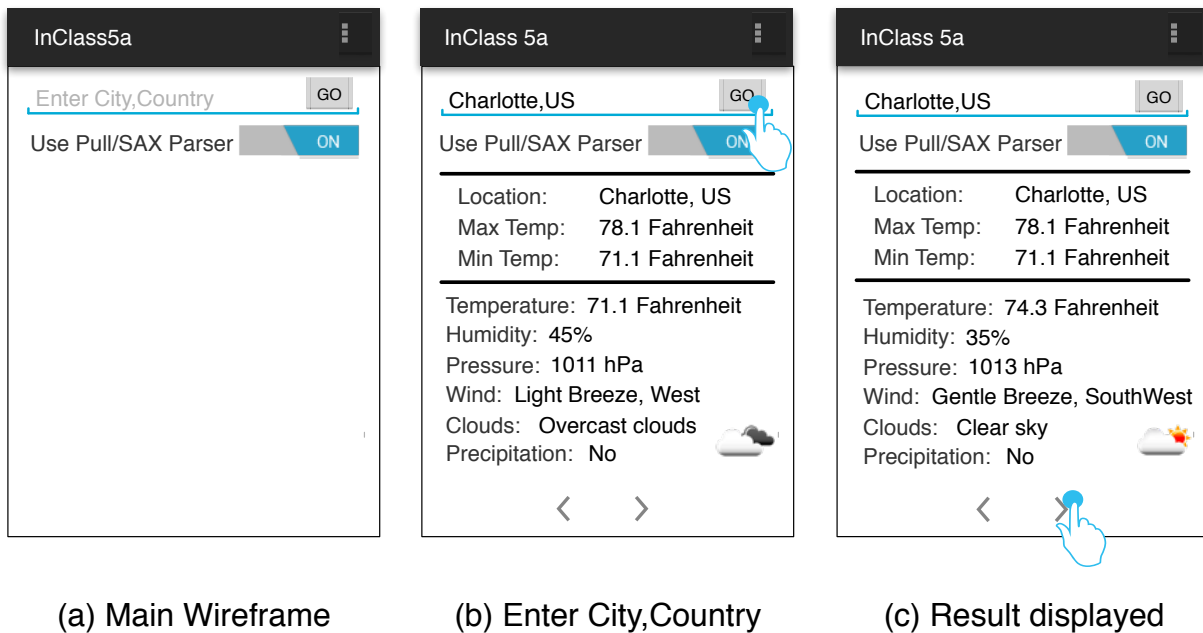


Figure 2, Main Activity Wireframe

Part 1: Parsing XML Document using XML SAX Parser (80 Points)

The Activity UI should match the UI presented in Figure 2. Below are the requirements:

1. Create a new android project called "In Class 5".
2. The EditText holds the query values separated by comma. See Figures 2(a) and 2(b).
3. You should use a separate thread to perform data retrieval from the server and data parsing. Do not use the Main Thread. Use an AsyncTask or a Thread/Handler.
4. The "Use Pull/SAX Parser" switch decides the method of parsing to use: if ON then the parser should be set to SAX.
5. When the "GO" button is pressed, the corresponding query request should be sent to the server to retrieve the xml document stream. If multiple cities are returned in the XML, only retrieve the first city. Note that what is returned is the forecast for the entire day, separated in eight (8) 3-hour time blocks.
6. Create a weather class containing string variables temperature, humidity, pressure, windSpeed, windDirection, clouds, precipitation, and symbol.
7. Implement an XML SAX Parser and pass the document stream to the parser. Parse the weather forecast information and store each forecast in a weather object.
8. The location presented in the TextView under the switch should be retrieved from the received XML from the location element.
9. Calculate the maximum and minimum temperature over all forecast data, and display them along with the retrieved location, as seen in Figure 2(b). Also, display the weather information for the first 3-hour block in TextViews as shown in Figure 3(b).
10. Display the corresponding weather icon, using the "symbol" string that is retrieved from the XML attribute "var." Note that weather icon can be downloaded as a bitmap by constructing a URL using the "var" string and appending it to the image URL as

follows: “<http://openweathermap.org/img/w/XXX.png>” in place of the XXX.

11. Clicking on the arrows should change the forecast displayed to the next time block. The retrieved location, max temperature, and min temperature should not change.

Part 2: Parsing XML Document using XML Pull Parser (20 Points)

This part is similar to the part 1, but instead use the XML Pull Parser to parse and display the city’s weather information when the switch is set to OFF.