1 Overview

In this lab, you are going to build a simple web service. Specifically, you are going to build a web service that allows a user to look up a member of the Luther community in the online directory.

You will need to implement the web service, as well as a client to talk to the service. You can write your client in either Java or Perl. I’m guessing you will opt for Java, but just in case you are interested in Perl, checkout the package SOAP::Lite for a simple Perl interface to SOAP. The client can provide a command line user interface, a simple servlet page, or a graphical swing interface. (It’s not as hard as you might think if you checkout the JEditorPane class).

Your client should accept a string to use in searching the directory. The string should be a name or partial name, or someone’s email or partial email address. In either case the search string should be passed to the web service as a parameter.

Your web service should allow the end user to enter a search string and should print out all the matches for the string. I will provide you with a class called DirectoryList, that talks to the Luther directory service, so you don’t have to worry about screen scraping the web page. You can use this object to implement the server side of the web service. DirectoryList has a constructor that takes your search string. There is an accessor function to allow you to get an iterator over some number of LutherUser objects. LutherUser is a simple java bean that contains the user’s name, email address, and campus phone number.

For example, this main method illustrates the use of the DirectoryList object. You should use the DirectoryList class as part of your server side code.

```java
public static void main(String[] args) {
    DirectoryList myD = new DirectoryList(args[1]);
    Iterator it = myD.getTheList().iterator();
    while (it.hasNext()) {
        LutherUser mu = (LutherUser)it.next();
        System.out.println(mu.getName() + "::" + mu.getEmail() + "::" + mu.getPhone());
    }
}
```

Your job is to create a web service that responds to client requests to look up a name in the directory. The advantage to this web service approach is that we can centralize the screen scraping part of the code on a server. That way we only need to change one place when LIS changes the directory web page (or allows us access to their user database.) There are many paths to success.
Learning how to adapt example code to do what you want is one of the most important skills that you will use as you start programming professionally. It’s important to remember that it’s OK if you don’t understand every nuance of the example programs right away. If you can get them to work for your own purpose, then you can spend the time figuring out the intricate details as you really need to know them.

**example1** A trivial web service to make sure that axis is installed and working

**example2** The calculator example we’ve looked at in class.

**example3** A simple example to illustrate the deploy.wsdd file

**example4** An example of adding a log handler to your project

**example5** Shows how you can generate a wsdl file from an existing service and use that file to create a client.

**example6** Shows how you can start from an existing class/interface and generate both the client and the server side code to create a full blown web page.

If you are careful and go through these examples. Particularly 5 and 6, writing the lab in a very structured way that allows for good object oriented expansion will be a breeze. In other words, your goal should be to figure out how to have your service return an array of LutherUser objects. If you follow this path, You will only need to write about 10 lines of java. (The system will generate the rest for you.) The server side is about the same, you can generate everything on the server automatically except for creating and calling and iterating over the objects returned by the DirectoryList object.

Even if you don’t make it through 5 and 6, you should be able to hack something together after working through the first 3 examples. If you choose this method you will probably end up returning everything from the server as a big string. That works, but is ugly, requires more hand coding, is a lot more susceptible to errors.
2 Grading

This lab is due 5:00pm Friday December 12th. There will be NO extensions granted for this last assignment.

The lab is worth 100 points. You will get 75–80 points if you hack something together that works. If you are able to create a working service that is fully structured and makes use of the code that is generated for you by axis you should get at least 95. When was the last time you got more points for writing less code?

When you turn in the program you should send me email letting me know where and how to run your client. You should leave your web service up and running so that all I need to do is run your client. I will also want to look at your source files.