95-702 Distributed Systems

Project 1

Assigned: Friday, September 7, 2012

Due: Friday, February September 21, 11:59:59 PM

This project has six objectives:

First, you are introduced to GlassFish. GlassFish is an open source application server that implements the latest JEE specification. This tool is used throughout the course. The NetBeans integrated development environment is introduced and is used to build source code and interact with GlassFish.

Second, you build your first set of distributed systems. These are four small web applications using Servlets and Java Server Pages.

Third, you are introduced to the Android simulator. In this simple project, you will be using the simulator's browser capabilities. The simulator runs stand-alone, or within the Eclipse IDE.

Fourth, you are introduced to simple mobile device awareness and adapting content to be suitable for either desktop or mobile devices.

Fifth, you are introduced to the MVC pattern if you have not used it before.

And finally, as in all projects this semester, you should reflect on the functional and non-functional characteristics (e.g. security, scalability, failure handling, interoperability) of your solutions. There will be questions on the midterm and final exam concerning these characteristics. You should be able to demonstrate a nuanced comprehension of course content and be able to explain the technical aspects in relation to potential real-world applications.

For each project task, software documentation is required. The software that you write (HTML files, Java files and so on) must contain comments that describe what each significant piece of code is intended to accomplish. Points will be deducted if code is not well documented. Read the documentation-related links provided on the course schedule (for class #1) to understand what is expected.

Be sure to consult the rubric linked from the course schedule for details on grading.

For each task below, you must submit screenshots that demonstrate your programs running. These screenshots will aid the grader in evaluating your project.

Task 1

Use the NetBeans Project Name: Project1Task1

You do not need to use an MVC framework for this project.

Write an index.jsp page that asks the user to enter a string of text data, and to make a choice of two hash functions using radio buttons. The hash function choices should be MD5 and SHA-1, with MD5 being the default. When the submit button is pressed a servlet is executed. The servlet must be named ComputeHashes.java. The servlet will compute the appropriate cryptographic hash value from the text transmitted by the browser. You will need to employ the Java crypto API to compute the MD5 or SHA-1 hash of the text. The original text will be echoed back to the browser along with the name of the hash, and the hash value. The hash values sent back to the browser should be displayed in two forms: as hexadecimal text and as base 64 notation. We will discuss the use of such hash values later in the course.

To compute the MD5 and SHA-1 hashes, use these standard java packages:

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
```

To compute the Base64 encoding, use the following package:

```
import sun.misc.BASE64Encoder;
```

The BASE64Encoder class is an internal non-documented class. BASE64Encoder objects have a method with the signature String encode(byte[]). It returns a base 64 string encoding of an array of bytes.

To compute the hexadecimal representation of a byte array, use the following code:

```
// From the web site "Real's How To"
public String getHexString(byte[] b) throws Exception {
   String result = "";
   for (int i=0; i < b.length; i++) {
      result += Integer.toString((b[i] & 0xff) + 0x100, 16).substring( 1 );
   }
   return result;
}
```

Be sure to provide a user friendly and attractive user interface.

So that you may test your program, here are example hashes.

Hashes of the string "Hello":

SHA-1 (Hex):F7FF9E8B7BB2E09B70935A5D785E0CC5D9D0ABF0

SHA-1 (Base 64): 9/+ei3uy4Jtwk1pdeF4MxdnQq/A=

MD5: (Hex): 8B1A9953C4611296A827ABF8C47804D7

MD5: (Base 64): ixqZU8RhEpaoJ6v4xHgE1w==

Task 2

Use the NetBeans Project Name: Project1Task2

You do not need to use an MVC framework for this project.

Later in the semester when we are studying the RSA algorithm, it will be useful to be able to do math operations on some arbitrarily large integers. Therefore Task 2 is to create a useful calculator app for this purpose.

Write a simple web application that allows a user to perform one of six operations on two, possibly very large, integers (x, y). The operations will include

1. Addition (x+y)
2. Multiplication (x*y)
3. An operation to determine if x and y are relatively prime
4. Modulo (x mod y)
5. A modular inverse (x^{-1} mod y)
6. Raise x to the power of y (i.e. x^y)

A JSP page will present three input fields to the user. The first two will be used to collect the two integers, x and y. The third will be used to collect the operation type. The operations supported will be "add", "multiply", "relativelyPrime", "mod", "modInverse", and "power". Use drop down boxes in XHTML. A submit button will be provided and when it is hit a servlet will be visited. The servlet will be named BigCalc.java and will use the BigInteger class to perform the conversions from strings and the appropriate computation. The servlet will return the result to the browser marked up in HTML. You need to validate both integers and the operation. In the case of invalid input return an error message to the browser - but don't crash the server.

The BigInteger class has multiply, add, modInverse, mod, and pow methods to use. For the operation that determines if the two integers are relatively prime use the gcd() method of the BigInteger class. If the greatest common divisor of the two integers is one then the two integers are relatively prime. And finally, note that the exponent for the pow method takes an integer, not a BigInteger.

Be sure to provide a user friendly and attractive user interface.

Task 3

Use the NetBeans Project Name: Project1Task3

You do not need to use an MVC framework for this project.

Write another web application using NetBeans. This application will determine if a string entered into a browser is a palindrome. A string is a palindrome if it is empty, has a single character, or reads the same when reading from left to right or from right to left. Name your servlet Palin.java. Use an appropriate doctype for an Android mobile.

Download and install the Android simulator from Google. Use the browser on the simulator to visit this web application.

Produce a screen shot showing the simulator working on your web application.

Note:

- You will not be able to connect to the servlet from the Android simulator using the IP address of “localhost” because "localhost" will refer to the Android device itself (not your laptop). Android provides a loopback address of 10.0.2.2 that refers to the system that the simulator is running on.
- For judging a palindrome, only consider letters. Disregard case, punctuation and white space. For example "Madam I'm Adam" is a palindrone.

Task 4

Use NetBeans Project Name: Project1Task4

You **MUST** use an MVC framework for this project.

For task 4, you will build an Olympic picture application. (This application will be demonstrated in class on Monday.)

Olympic.org, the official web site of the Olympic movement, has a section of photos. You are to build an application that allows the user to choose a sport, and it returns a photo of athletes in that sport.

In more detail:

1. The user is presented with a screen with instructions: "Choose a sport to display a picture of an Olympic athlete and a menu of sports to choose from."
    - You should include at least 12 sports for which there are pictures.
2. Upon Submit, a servlet will search for a picture of the sport at Olympic.org.
3. The response to the user should be "Here is picture of an athlete playing *insert the sport*", and the picture.
    - Olympic.org has smaller and larger photo sizes. You should use the larger ones.

- The user should also be able to "Choose another sport." with an appropriate menu and Submit button.

Exceptions:

4. If a photo cannot be found, then an appropriate graceful error message should be provided, as well as the ability to " Choose another sport?".
5. If there is a failure in connecting to Olympic.org, then an appropriate graceful error message should be provided, as well as the ability to " Choose another sport".

Your application need only work with a desktop browser (not the Android simulator).

Finding maps:

- Olympic.org is apparently hosted in Italy, and requests have high latency. Don't be surprised if each request takes substantial time.
- For this task you should screen scrape Olympic.org. (It does not provide an API anyway.)
- Your solution should not pre-select the choice of an image. Rather, you should find the picture programmatically. In this way, if Olympic.org adds a new picture, then your images may change.
- Olympic.org does provide a search capability. Experiment with the searches to see how the search query works. Note that using a "sport=" of a number, and the "search=" parameter empty will search for all images of that sport.
- Additional hints will be discussed in class.

Produce screen shots of your application displaying pictures of 5 different sports.

Questions:

Questions should be posted to the Blackboard Discussion Board, under the Project 1 Discussion Forum.

Summary:

There should be four projects in Netbeans.

The Netbeans projects will be named as follows:

- Project1Task1
- Project1Task2
- Project1Task3
- Project1Task4

You should also have four screenshots folders:

- Project1Task1 Screenshots
- Project1Task2 Screenshots
- Project1Task3 Screenshots
- Project1Task4 Screenshots

Copy all your NetBeans project folders and screenshot folders into a folder named with your andrew id.

Zip that folder, and submit it to Blackboard.

The submission should be a single zip file.