95-702 Distributed Systems
Project 1
Assigned: Saturday, September 3, 2011
Due: Friday, September 16, 11:59:59 PM

This project has five objectives:

First, the student is introduced to GlassFish. GlassFish is an open source application server that implements the JEE 5 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, the student builds his or her first set of distributed systems. The student builds four small web applications using Java Server Pages and servlets.

Third, the student is 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, the student is introduced to mobile device awareness and adapting content to be suitable for either desktop or mobile devices.

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 relative to 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.

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

For each of the four tasks below, submit a documented servlet and an index.jsp page. The documentation will include your name, a description of each piece of code and well-chosen variable names. There is an example linked on the course schedule showing what we mean by "good documentation".
In addition, you must submit screenshots that demonstrate your programs running. These screenshots will aid the grader in evaluating your project.

**Task 1**

Use NetBeans Project Name: Project1Task1

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

Write an index.jsp page that requests a user to enter a string of text data. Provide a submit button. When the submit button is pressed a servlet is executed. The servlet must be named ComputeHashes.java. The servlet will compute two cryptographic hash values from the text transmitted by the browser. One hash value will be computed using MD5 and the other using SHA-1. You will need to employ the Java crypto API to compute the MD5 and SHA1 hashes of the text. The original text will be echoed back to the browser along with the two hash values. The hash values will also be sent back to the browser and will appear as hexadecimal text and as text in 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:

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

To compute the Bas64 encoding, use the following package:

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

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

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

```java
// 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.toHexString((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, an example execution is provided.
Computing hashes of Hello of length 5

SHA-1 (Hex): F7FF9E8B7BB2E09B70935A5D785E0CC5D9D0ABF0
SHA-1 (Base 64): 9/+ei3uy4Jtwk1pdeF4MxdnQq/A=
MD5: (Hex): 8B1A9953C4611296A827ABF8C47804D7
MD5: (Base 64): ixqZU8RhEpaoJ6v4xHgE1w==

Task 2
Use 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. A modular inverse (x\(^{-1}\) mod y).
5. 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", "modInverse", "mod", 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.

Be sure to provide a user friendly and attractive user interface.
Task 3

Use 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.

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.

Notes:

- You will not be able to connect to “localhost” from your Android simulator. One way to solve this is to find the IP address of your machine and use that instead. A simplier approach is to use Android’s Loopback Address of 10.0.2.2.

Task 4

Use NetBeans Project Name: Project1Task4
You **MUST** use an MVC framework for this project.

For task 4, build a web application that will ask the user for a letter, and then reply with the name of an animal that begins with that letter, and a video that shows the animal.

In more detail:

1. User is presented with a screen with instructions: “Welcome to the Animal Letter Videos game. This simple game will map letters to animals, such as "A is for Alligator" and will show you a video that shows the animal selected from Vimeo.com. Select the letter you would like to play with.
2. User types in a letter a-z, or A-Z, and Submit.
3. If input is valid (e.g. “b”) then the user is presented with a reply such as:
4. This screen should also allow the user play again by typing in another letter and hitting submit.

Exceptions:
3a. If the input is not a valid letter, then the user should be presented with a reply that indicates that the input was not a valid letter (e.g. “7 is not a letter, please type in a letter from a-z or A-Z”) and allows them to choose again.
3b. If there is no video available for a given animal, then the user should be informed of that (the server should not just crash, nor return nothing).

Device awareness and content adaptation:
• Your application should work on both a desktop browser, and an Android browser, and should be device aware and adapt content appropriately.
• Most importantly, you should choose an appropriate video size for the screen size. A suitably large video window should be used if the user is on a desktop/laptop browser. A suitably small video window should be used if the user is on an Android phone.

Finding videos:
• Videos should be found dynamically from Vimeo.com. That is, you should not pre-choose the videos to use, but should search for them when needed. (This will keep the game fresh and, occasionally, changing.)
• An example of a useful search to use, where search is your animal name is: http://vimeo.com/categories/nature/animals/videos/search:nature%20"+search+/sort:relevant"
• This search usually provides a list of videos that mostly have something to do with the animal you are looking for. It is not perfect. Joe played around with a lot of search options to find what seemed to work best. You are welcome to improve on this search.
• In the future, retrieving a video URL from a service like Vimeo will be very easy using web services. For now, we will “screenscape” the reply from Vimeo. Screenscraping means that your web application will read the HTML reply stream from Vimeo and search in it for the information that you want.
• By deciphering the Vimeo search results page, the video you want to embed in your application's response should have a URL of the form: http://vimeo.com/10197057, where 10197057 is the unique number of the video. You need to find that unique video number by parsing the result of your Vimeo search.
• What is shown in your browser as the result of a search is not what the browser receives from its first GET. The browser goes back out and fetches more
information to build the page. So "view source" will not give you an accurate record of what the search first returns. Rather, print the results to the GlassFish console to be able to see the true result.

- Hint: After examining the Vimeo search results, I found that the best way to extract the video number of the first video in the search results is to:
  - Search for "thumbnail_format"
  - Continuing from that point, search for "href=""
  - What follows this string will be the video number, which you can extract and use to build the video URL given above.

Mapping of letters to animals
- It is acceptable to have a fixed mapping of letters to animals. That is, “a” can always be for “alligator”, “b” for “baboon”, etc.
- (In the future, once we have the power of using web services, we may search for alternatives for each letter.)
- Note: You must search for a video to correspond to the animal. You should not pre-choose the video for each animal.

Device awareness and content adaptation:
- For this task, you can take a simple approach and just check if the user-agent is an Android device or not.
- You need to have an appropriate DOCTYPE string (for desktop/laptop or mobile) defined as the first element of your HTML reply.

Produce screen shots of:
- Your application displaying videos of 3 letters in a desktop/laptop browser
- Your application displaying videos of 3 different letters in the Android simulator.

Summary:
There should be four projects in Netbeans.
The Netbeans projects will be named as follows:
- Project1Task1
- Project1Task2
- Project1Task3
- Project1Task4

You should also have two screenshots folders:
- 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.