

Agent Web Interface

Term Report

Blake Park

December 13th, 2011

Summary

The project develops a web user interface and server that submits an UWAgent mobile agent to launch and track a user job remotely. This project will enable users to upload the job files and execute or modify in web application.

Presented Challenges

- The developer was not familiar with web development environment and language (html).
- Connect the NetBeans IDE database and store data from the user's input
- Share the object between Servlets
- Increment the int object in application scope
- Encrypt the password and decrypt it correctly with secret key generated from key generator function

Current states of the project

1. Users Registration: User is able to register with their ID, name, password and email. These data is going to be stored in database table(Users). Password will be stored as encrypted.
2. Users Login: User can login from the web with ID and Password, it checks the ID and Password match correctly. It Checks the session is new by retrieve user id from the hash table.
3. Users Logout: Users can logout by clicking logout button from the web and it remove the user id from the hash table and make session invalidate.

4. File upload: User can upload multiple files of job with appropriate information includes job name, CPU, memory and command line. It will be stored in local directory “C:\folder\user name\job name\”

Necessary changes for current project states

- Integrate job information and file upload in one page
- Add synchronize clause for hash table
- Remove http://local host8080: from all the URL
- declare the connection URL with one string
- change the query execution(first 3 lines of code for separate function)
- provide meaningful attribute name for Servlet context
- add synchronize clause the jobID
- verify how to work max_size_action variable in the upload servlet file
- give provision to user to modify existing upload file so user can modify job files
- restrict the file type when they upload file
- set admin with more admin function
- remove unused code or function

In terms of outlook for next quarter

- Ready to keep track of job files that has been submitted
- Switch the database and modify code.
- Connect with UW servers
- Set up the MASS software through the website

GOALS

For this quarter the project had the following goals:

- Set up the user login pages
 1. Check user's ID and Password matches correctly
 2. Store user's Id in hash table for session management
- Let user create account through registration
 1. Store user information(ID, Password, Email, First name, Last name) in the database
 2. Encrypt password before store in the database
- Let user to log out
 1. Delete user id from hash table and invalidate session
- Create job files uploading environment
 1. Store job file information(Job name, Command line, CPU, Memory) in the database
 2. Upload job files up to 5 for one job
 3. Create the directory under username and job name
- Let user to view the job files and modify
 1. display the job files from directory
 2. let user modify the job files(add, delete, replace)
- Execute the job that has been uploaded
 1. Execute the job files
- Design the web pages with some graphics
 1. Design header, footer and background of JSP pages for the project

STATUS

- Set up the user login pages
 1. Check user's ID and Password matches correctly –Completed: if it matches, redirect user to main page otherwise display error message
 2. Store user's Id in hash table for session management –Completed: store user id with session id in hash table
- Let user create account through registration
 1. Store user information(ID, Password, Email, First name, Last name) in the database –Completed: it connects the database and successfully stored the information
 2. Encrypt password before store in the database –Completed: The password encrypted by function in the password encrypt class and stored in database
- Let user to log out
 1. Delete user id from hash table and invalidate session –Completed: it successfully remove the user id hash table and invalidate the session by function call `session.invalidate()`;
- Create job files uploading environment
 1. Store job file information(Job name, Command line, CPU, Memory) in the database –Completed with information storing in the database but still need to put information submission page and file submission page together
 2. Upload job files up to 5 for one job –Completed: files could be uploaded up to 5
 3. Create the directory under username and job name –Completed: it construct the directory and upload file in the location `C:\folder\username\jobname\` but need to set this up in the server later
- Let user to view the job files and modify
 1. display the job files from directory –Need to be done: User should see all the uploaded job by themselves
 2. let user modify the job files(add, delete, replace) –Need to be done: User also should modify job files in web application. They might update, add or delete files
- Execute the job that has been uploaded

1. Execute the job files –Need to be done: User should run those job files they uploaded in web application
- Design the web pages with some graphics
 1. Design header, footer and background of JSP pages for the project –In progress: background, header and footer with some colors and picture but still need to be developed as project goes on

PROJECT INFORMATION

- JSPs
 1. Index.jsp – login page
 2. Main.jsp – main page after user login where display menu
 3. Registration.jsp – display user registration
 4. Jobsubmit.jsp – display job information submission
 5. FileUpload.jsp – display file upload(should be integrated with jobsubmit.jsp later)
 6. Logout.jsp
- Servlets and other java class files
 1. ControllerServlet – It handles login and log out (should change the name to LoginLogoutServlet)
 2. RegistrationServlet - handles registrtion
 3. UploadServlet – handles file upload
 4. JobSubmissionServlet – handles submitting job information
 5. Dal.java – handles all database connection
 6. PasswordEncryptor.java – java class used to encrypt and decrypt the password
 7. MultipartRequestParser.java – java class for uploadServlet

DIFFICULTIES AND RESOLUTIONS

The major challenge of the project in the beginning of this quarter was how to approach the concept of Servlet and get familiar with web application development environment with Netbeans IDE. For the first few weeks, I had to go through video tutorials how to create the web application project. The tutorial

helped me much but I still had to understand the concepts of Servlet and what does the Servlet capable of with html. I also had to learn how to use html for jsp pages as well. I spent some time to figure how to html page invoke the servlet to display web application on web browser.

Once I created the project, the first task was set up the connection with database for the registration and login pages. Since I actually had no experience with database as well, I had to learn how to create the database with appropriate tables and columns. Then I had to hard code the connection between the database and Servlet.

As the project goes on, the most difficult part was I had to learn every concept before I become able to write code. The next task was session management. I had to learn the concept of the session and how to handle session management. Sharing the hash table object problem was resolved by put couple of Servlets in one Servlet. For setting up the file upload environment I was able to get much help from the textbook.

Recently, I could resolve most of previous problems but I guess I will have more challenges on file display and execution. When I start to work with server next quarter, it might produce more hard challenges.