Project Report

Objectives

The main objective of our project was to tie together various technologies which were new to us, in order to create a simple game application. We intentionally chose the simple game of tic-tac-toe so that we could focus on learning the new technologies rather than on implementation of difficult game logic.

Research Problems Addressed

There technically was not a research problem addressed, as we were not trying to solve a problem which had never been solved before. Rather, we were addressing the educational goals of the individual team members: to gain more experience working on a SCRUM team, to explore several open source technologies, and to learn in a practical way how to apply some of the technologies discussed in class lectures.

Approach

The main approach we used was that of linking several small web services together through one main gateway project. The main gateway project was written in java using java servlets and jsp files. This gateway was added to an Apache Tomcat server, using localhost:8080.

When a user opens a browser and navigates to localhost:8080/gateway/, since they are not logged in, the LoginCheckFilter redirects them to the Login.html page. On this page, the user has the ability to either log in to the application or to register. If they click the ‘Register’ button, then the href takes them to the Register.html page. There they will fill out their email, password, and name, and then click the “Submit” button. The submit button calls the RegisterUser() function in the register.js javascript file.

**Methods for Evaluation of Results**

**Architecture and Functional Components of Prototype System**

Because we are using various webservers, to run our prototype system, we need to take several steps to fire up the various servers. We tried to standardize our install locations and database locations throughout our team so that we could all start up the system in the same way.

First, the MongoDb server needs to be started so that we have a database to store our player and game data. With the architecture we used, this can be accomplished by opening a command prompt, and navigating to c:\program files\MongoDB\server\3.2\bin, and typing mongod.exe –dbpath c:\temp\db.

Second, the node.js server needs fired up so that we can use the sockets.io library to handle our chat application. To do this, one needs to open a cmd prompt, navigate to where the chat files are in our GitHub directory (for example, c:\users\yourname\Documents\GitHubDownloads\project\Chat). Once you are in the correct directory, you simply have to type node index.js to start the node server on port 3000.

Third, the Authentication service needs to be started in your java IDE. In Eclipse, this is done for our project by right-clicking on the ‘AuthenticationServerApplication’ and selecting ‘run as Java application.’ It will run on port 9001.

Fourth, the GameServer needs to be started, also run as a Java application. It will run on port 9002.

Finally, the gateway application needs to be added onto an Apache Tomcat server, and that server needs to be started.

**Three Most Interesting Contributions of Project Design and/or Implementation**

Summarize:

-- what you have learned through the hand-on experience of doing this project

-- what concepts and techniques you learned in class are used in the current project design

-- what concepts and techniques you learned in class can be considered for extension of your current project

expect the report to be well written and documented with references