CS420 - Spring 2006

Team Assignment 5
The Geospace Project (*http://geospace.vancouver.wsu.edu*) – Iteration 5
Due Date: April 28, 11:59 p.m.
Points: 200 points

Objectives:

1. Submit a Gant Chart showing your planned activities. (Hopefully you create this before you start with this iteration of the project.)
2. Both teams should meet and agree upon requirements presented in class as well as in this assignment. Both teams should work out an agreed upon design and a remote interface based upon the agreed upon requirements.
3. Develop and **thoroughly test** the deployed system *http://geospace.vancouver.wsu.edu*.
4. Use the Subversion repository on bitterroot to track all documents and code.

Iteration Description and Requirements:

Currently we have a system that allows users to:

1. log into the system, modify their profile, and logout.
2. interact with a yahoo map, save the map, and add and delete artifacts (notes, markers, images, and **paths**) on the map.
3. publish maps (no domain restriction)

This iteration will address the following:

1. Traceability between design and code.
2. Scalability.
3. The addition of several features that will enhance marketability.

Deliverables for Both Teams:

1. Gant chart and proposal. The proposal should outline team objectives for the next 2 -3 weeks. **The chart and proposal are due April 20**. (10 pts)
2. Create Activity, Sequence, State Chart, and Class diagrams that describe the implemented system. Behavioral diagrams are often the most useful. It is up to you to choose a combination of diagrams that best describe the system. The code should contain comments that reference the UML diagrams. **The objective of this deliverable is to provide excellent documentation for new developers.** (50 pts).
3. Each team member must review their peers using the peer review forms. Peer reviews are kept private and must be submitted via the class website.
4. Use an ant file to build and test.
5. Deploy to the class server machine: geospace.vancouver.wsu.edu. Each team should select 1 team member to be responsible for deployment.

Deliverable Specific to the Client Team: (140 pts)

1. Users may not want all articles to be displayed at the world level. Allow the user to specify bounding values for the zoom range on articles in a map.
2. The server team will be setting an account storage limit. Before users can upload documents, their account limit must be checked. If the account limit has been violated, a friendly reminder must be displayed to the user.
3. There should be an email icon in the map menu that allows users to email the map URL to an address of their choosing (we should record the addresses in their account, so we can retrieve them later for marketing purposes).
4. Other document types besides images should be uploadable/downloadable (however, we don't have time to address viewer issues).
5. When you publish a map ask the user if they would like to password protect the map, if so allow them to enter a password (twice) for this map. When published maps are viewed (that are password protected), the user must enter the recorded password to view. When maps are unpublished the password is lost and the default is no password.

Deliverable Specific to the Server Team: (140 pts)

1. Update the test suite to cover new functionality.
2. Add to EJBs to provide functionality for storage limit enforcement.
3. Add to EJBs to provide functionality for zoom range storage per article.
4. Add to EJBs to provide for storage of documents in the database (this includes images as well as other types).
5. Add to EJBS to provide password protection of published maps.
6. Scale the system by deploying to two machines using the JBOSS/MYSQL built in scalability functionality.

Submission and Grading:

1. All submitted documents will be graded on content as well as grammar. All documents must be in an html format, so they can be posted on a website if necessary.
2. Create a plain text README file that describes how to deploy and run the system. This should be placed in the docs directory and labeled (README_TH5).
3. Submit all documents and code by checking them into subversion repository. (I will checkout a version when the assignment is due.)
4. Each team must submit extra documents containing the following artifacts:
    a. Group meeting time(s) and duration. (post this on your website). You can use a password using htaccess.
    b. A description of who worked on what (place this in your peer review forms).
5. Each student must submit (via the website) a peer-review for each of their teammates. Use one-peer review form, zip it, and submit it to the class website.