

CS420 - Spring 2006

Team Assignment 2

The Geospatial Registration Project – Iteration 2

Due Date: February 28, 2:50 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 test a prototype that provides end to end connectivity (with minimal functionality).
4. Deploy to the class project website: geospace.vancouver.wsu.edu.
5. Use the Subversion repository on bitterroot to track all documents and code.

Iteration Description:

Currently we have a system that allows users to log in to the system, modify their profile, and logout. This iteration will provide basic persistent mapping functionality. Geo Registered content implies that we will be using maps to associate user data with location. We will use the Yahoo Maps API (<http://developer.yahoo.net/maps/>) to generate the geospatial content. We will allow users to add their own data to the maps and allow them to save that data and the associated locations for future viewing.

Project Requirements:

1. The system must be scalable in order to perform well under heavy user load.
2. The system must be secure and reliable.
3. The system must use open source tools, but ownership of the created content including code and documentation must be retained by the University, The Instructor, and the Students.

Iteration 2 Use-Cases:

6. Display User Homepage

Actor: Web User

Pre-Condition: Use-Case 3 User logs into system (successfully)

Main Scenario:

1. The user should be presented with a both a file interface and a map interface.
2. The user chooses either
 - a. Interacting with the File Interface (See Use-Case 7)
 - b. Interacting with the Map/Image Interface (See Use-Case 8)
 - c. Logout (See Use-Case 5)
 - d. Modify Profile (Write this Use-Case) (Should be able to navigate back to Display User Homepage Use-Case 6)

7. Interacting with the File Interface

Actor: Web User

Pre-Condition: Use-Case 6 (Display User Homepage)

Main Scenario:

1. The user is presented with a file menu that lists all saved documents that can be viewed in the Map/Image Interface. Each document can be either a map or a map article (something that can be displayed on a map). A directory/file relationship should serve as a metaphor for a map/map article. Map articles can have symbolic links pointing to them from other maps. Thus one Map Article may be represented in several map views. The user interface should convey the relationship between maps and map articles.
2. The user chooses either to:
 - a. Upload a document. (the user can choose to associate a map and location) **(UI Team Must Write This Use-Case).**
 - b. Delete a document.
 - c. Refresh the file view.
 - d. View a document in the map/image view.

8. Interacting with the Map/Image Interface

Actor: Web User

Pre-Condition: Use-Case 6 (Display User Homepage)

Main Scenario:

1. The first time the user logs into the homepage the map interface should provide a world map. Additional logins should display the last map used by the user.
2. The user can chooses either to:
 - e. Pan and Zoom
 - f. Go to a location
 - g. Save the current view (The user should be queried for a name and it should show up as a map on the File View) (**UI Team Must Write This Use-Case**).
 - h. Add a Map Article (e.g. photograph) from the file interface to a location on the current map.

Deliverables for Both Teams:

1. Gant chart and proposal. The proposal should outline team objectives for the next 2 weeks. **The char and proposal is due Feb 21.**
2. Based on our discussions in class and assignment requirements, add to iteration 1's SRS. Provides enough detail to carry out this iteration (description, requirements, use-cases, etc). The UI team will need to add new Use-Cases.
3. Use the requirements to create a simple UML design of the system. An Activity Diagram and Class diagram are sufficient. (Use Rational or Visio to create the UML – both teams must agree upon tool.)
4. A test plan (not necessary JUNIT tests) must be created and executed.
5. 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.
6. Use an ant file to build and test.
7. 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:

1. Create a web accessible user interface for the project that complies with the requirements and Use-Cases. This should include JSPs to server up dynamic functionality. The content should be deployable into the JBOSS application server using a "war" file.
2. Use the EJB's provided by the server team to save and retrieve information.

Deliverable Specific to the Server Team:

1. Create one or more EJBs that provide functionality for saving, deleting, editing, and searching, user data information as described by the requirements and use-cases. The content should be deployable into the JBOSS application server using a "jar" file.
2. Create a database that stores user information and is amicable to searches and retrieval.

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_TH1).
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.