

CSCI 4448 Project Part 4

1. Team:

- a. Taylor Jesse
- b. Samuel Reed
- c. Spencer Hanson

2. Title: Decision Tree Visualization and Learning Tool

3. Project Vision: Our project's goal is to provide a web service to facilitate building a decision tree. When a user creates an account and logs in, we plan on providing tools to help create a survey that will focus on a specific decision.

Project Description: This will involve a list of questions and possible answers regarding situational attributes. Once a user creates a survey, we plan on generating a link so that the user can easily distribute the survey in order to accumulate sample data. Once there is a sufficient amount of sample data, the user can create a decision tree visualization of the result. We'll apply basic entropy information theory on the sample data in order to show the user what the decision pathway looks like from the individuals who took the survey. The ultimate goal is to provide an easy way to model decision making processes and show an informative visualization.

4. Part 2 Class Diagram

****See Updated Class Diagram and Completed Class Diagram Appended to Back****

5. Updated Part 2 Class Diagram

****See Updated Class Diagram and Completed Class Diagram Appended to Back****

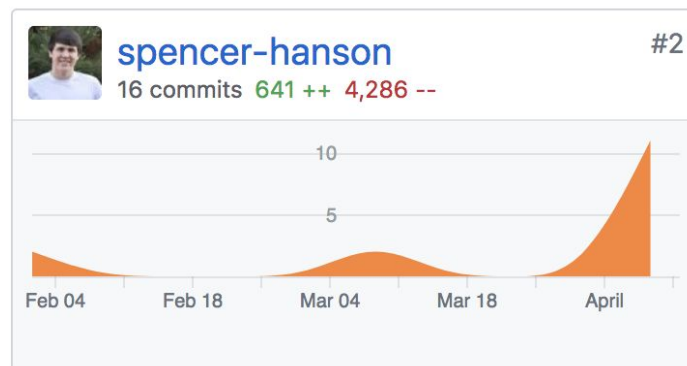
6. Summary: During this sprint, for the project part 4, we implemented code for converting our inputs for the decision tree. Also we set up a Tomcat server, to serve the Java Server Pages. We implemented some basic HTML for our interface, and converted them to JSP pages, with some basic functionality.

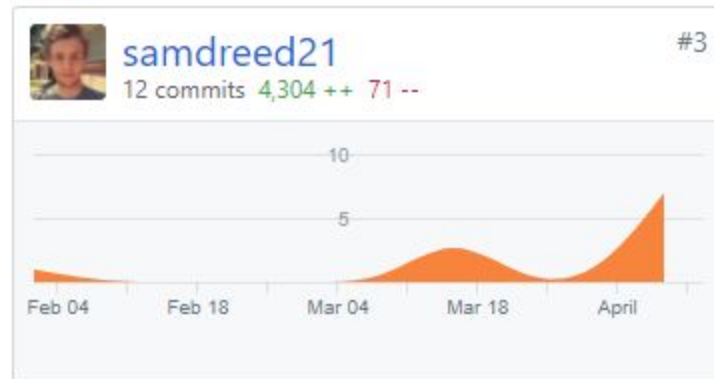
7. Breakdown:

- a. Spencer Hanson
 - i. Set up TomCat server
 - ii. Converted html pages to jsp pages
 - iii. Wrote ANT script for building WAR file to deploy to TomCat
- b. Taylor Jesse

- i. Wrote the html pages (now jsp pages) for the Login Page, Signup Page, Survey Editor Page, Take Survey Page, and Survey Completion Page.
 - ii. Wrote conversion functions in the survey class such as createKeys() and convertResponses() which create unique numerical keys for each answer in order to feed into the decision tree model.
- c. Sam Reed
- i. Wrote html pages for Main Menu and View Survey
 - ii. Wrote python script for creating decision tree
 - iii. Wrote java code to write output of converted responses to text file
 - iv. Wrote Method to call python script for buildTree

8. GitHub Graphs:





9. Estimate Remaining Effort: We estimate that our project is roughly 75% done at this point. The other 25% will involve connecting our backend code to the UI via the controller class. This involves incorporating the Observer design pattern in order for our Controller class to receive updates from the View class which will take some time. After that, we will need to do some testing of the application to make sure our requirements are met. We have basic tests that test the functionality of the backend by generating a survey, unique keys, and responses.

10. Design Patterns: We are incorporating the Observer design pattern into our project. Our View class will be the “Subject” and the Controller class will be the “Observer.” Like the Observer pattern, the View class uses composition on the Controller class. Our update() method in View is synonymous to notify() in the Subject class of the Observer pattern and our receiveInput() in the Controller coincides with the update() method in the Observer class. Effectively, our update() will notify the Controller class that something in the View class has changed (via user input) and the receiveInput() will handle and update our backend model with those changes. Since there’s exactly one Controller and one View, we do not implement attach() and detach() because there is only one and always one observer (Controller class). We also do not incorporate the abstract classes of Observer because it is not necessary for only one instance of an observer and one instance of a subject.

11. Show the portion of your class diagram with each design pattern implemented:

Show the class diagram for the actual design pattern

Show the portion of your class diagram that implements this design pattern

Designate the participants in your own diagram

View subject, controller observer

12. Final Iteration: In the next iteration of our Visualization Decision Tree project, we plan on hooking up the Java Server Pages to our code, making the interface functional. Also a minor cleanup to the UI is needed, since some of the HTML isn't very pretty.

****Attached below are the class diagrams. The first class diagram is the corrected class diagram from part 2 of the project, the second class diagram is the class diagram reflecting the work we have done so far****

