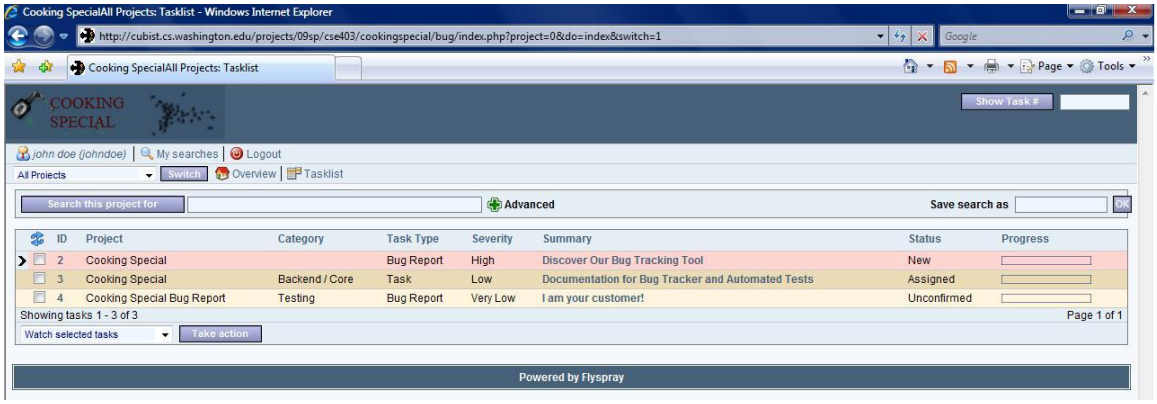
Bug Tracking

How our bug tracking system works?

We use a web based bug tracking software currently installed on http://cubist.cs.washington.edu/projects/09sp/cse403/cookingspecial/bug/ . Both our team members and customers submit bugs through this software. However, we have two different projects under this software. "Cooking Special" project is only for team use. For any anonymous user, this project is invisible. We assign tasks and submit bugs using this section. "Cooking Special Bug Report" project is for customers or anonymous visitors to submit the bugs they have found or to request the features they want to see in our project.

Here is a screenshot of our bug tracking system.



1) How does a developer file a new bug.

"Cooking Special" Project on http://cubist.cs.washington.edu/projects/09sp/cse403/cookingspecial/bug/ is for our developers to submit the bugs they have found. They log into the system by typing their usernames and passwords. This section is visible only to developers. You can check this section by typing "johndoe" as your username and password.

When a developer submits a new bug, he enters the category of the bug such as Data Gathering, Frontend or Backend, current status of the bug, severity of the bug, and priority. He can assign the bug to a person who is responsible from that category or add some other people to watch the process of handling this bug. Therefore the probability of missing any bug is minimized with this method.

2) How does a customer file a new bug.

If a customer goes to our bug tracking website, he won't be able to see the submitted bugs by our developers, but he will be able to submit new bugs by creating an account or even anonymously. Customer can either submit a bug in the system, or he can request a new feature. He also enters the category of the bug so that the submitted bug is directed to the correct person who is responsible from that category. He can also specify the severity and priority of the bug. So developers can take immediate actions if there is a big problem with the system or some tasks can be prioritized over others if they are more severe. Customers can also specify their email address to watch the process of his submitted bug. He is mailed immediately in case of any process.

Please make sure All Projects
http://cubist.cs.washington.edu/projects/09sp/cse403/cookingspecial/bug/index.php?project=0&do=index&switch=1 is selected when searching for existing bugs. Moreover customers file bugs under "Cooking Special Bug Report" while devs under "Cooking Special" (from the drop down list box near top left).

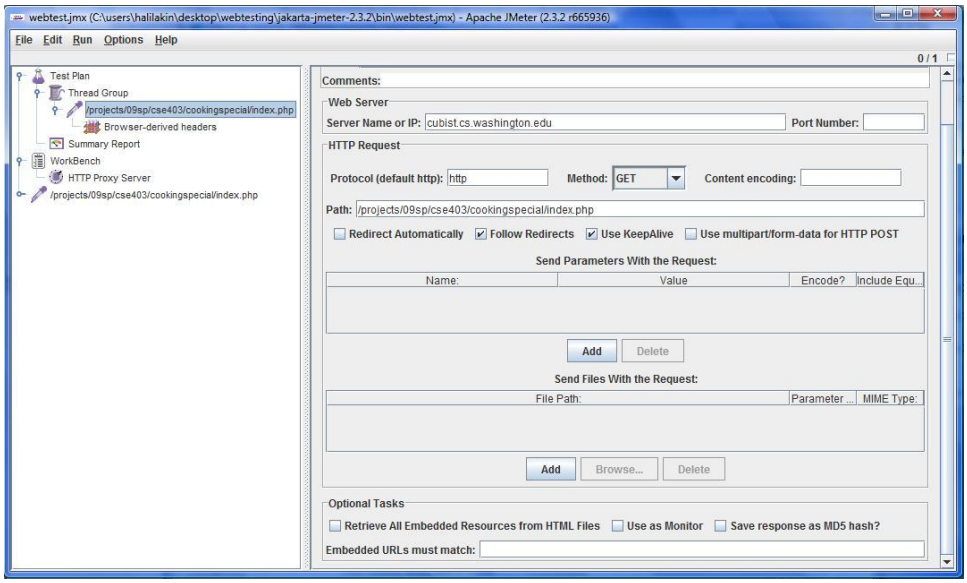
Filing a bug guideline: PLEASE FOLLOW THIS

1. Please try to fill all the fields on our website to your best knowledge
2. **Summary**: Please describe the bug in approximately 60 or fewer characters.
3. In the **Details** Field, please include the following in the with appropriate heading.
   Description: The details of your problem report, including:
   Overview: More detailed restatement of summary.
   Steps to Reproduce: Minimized, easy-to-follow steps that will trigger the bug. Include any special setup steps.
   Actual Results: What the application did after performing the above steps. The application crashed.
   Expected Results: What the application should have done, were the bug not present. (AND/Or, at least, the application should not crash.)
   Build Date & Platform: Date and platform of the build in which you first encountered the bug.
   Additional Builds and Platforms: Whether or not the bug takes place on other platforms (or browsers, if applicable).
   Additional Information: Any other useful information.
4. **Add an attachment**: You can attach relevant files to a bug report. Debugging information more than 20 lines long should be supplied this way. Also, if you have an HTML file that demonstrates the bug, you should attach that. You can only attach one file during initial submission so if your demonstration needs more, revisit the newly filed bug to do this part. Attach any subsidiary files (such as images) first and then edit the HTML file to point to the new URLs of the attached files before uploading, so the demo is self-contained. Ask before attaching more than five files.

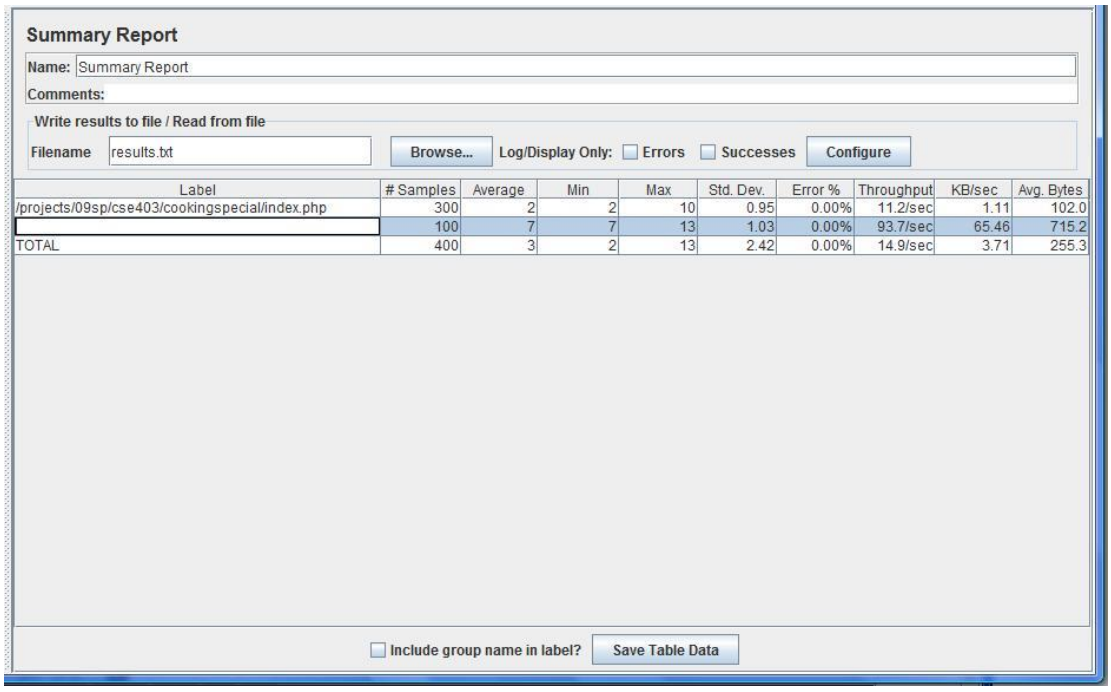
This above information was mainly taken from the Mozilla developer page at https://developer.mozilla.org/index.php?title=En/Bug_writing_guidelines&action=print which is also available on our server at http://cubist.cs.washington.edu/projects/09sp/cse403/cookingspecial/bug_report/bug_guidelines.htm

How our testing system works?

We use JMeter for testing. JMeter is a desktop application written in Java, which allows users to load, test and analyze HTTP, HTTPS websites and even databases. We can query a web page and the database with different input strings in a loop. Here is the screenshot of our first tests.



The test scripts are saved in .jmx files. As we build up our system, our testing mechanism will be pretty much writing and running these jmx files. Since we currently have a single index.php page in our host, we tested this page and here is a sample of our running testing system.


Summary Report
Name: Summary Report
Comments:
Write results to file / Read from file
Filename results.txt
Browse...
Log/Display Only: Errors Successes
Configure
Label # Samples Average Min Max Std. Dev. Error % Throughput KB/sec Avg. Bytes
/projects/09sp/cse403/cookingspecial/index.php 300 2 2 10 0.95 0.00% 11.2/sec 1.11 102.0
100 7 7 13 1.03 0.00% 93.7/sec 65.46 715.2
TOTAL 400 3 2 13 2.42 0.00% 14.9/sec 3.71 255.3
Include group name in label?
Save Table Data