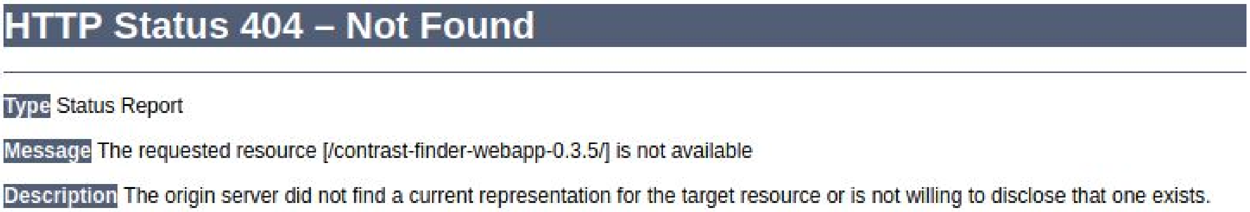
Deliverable 1

Tanaguru

Our first choice for the project was the Tanaguru Contrast Finder. We chose this due to its interesting concept and decent documentation at first glance. We extensively tried to get this project to build but were ultimately unsuccessful. While the documentation looks fine at first glance, using it to build the system was a different beast entirely. The build and installation instructions were vague and completely outdated to the current iteration of the project. While efforts were made to use this information for usage on the current build, little success was found. Previous versions of java, ubuntu, and tomcat were also tested to see if it was a compatibility issue to little avail. We were able to figure out some workarounds for some of our issues, but we also looked into STEM, our second choice, to see how we would fare.



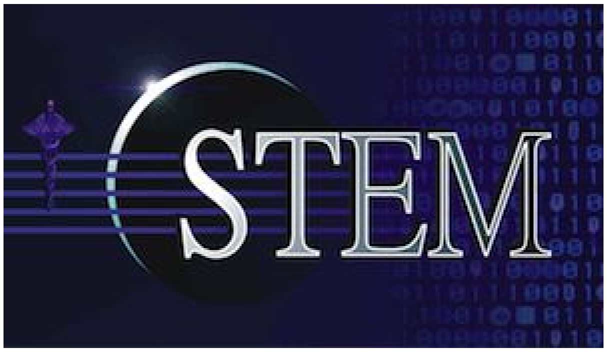
HTTP Status 404 – Not Found

Type Status Report

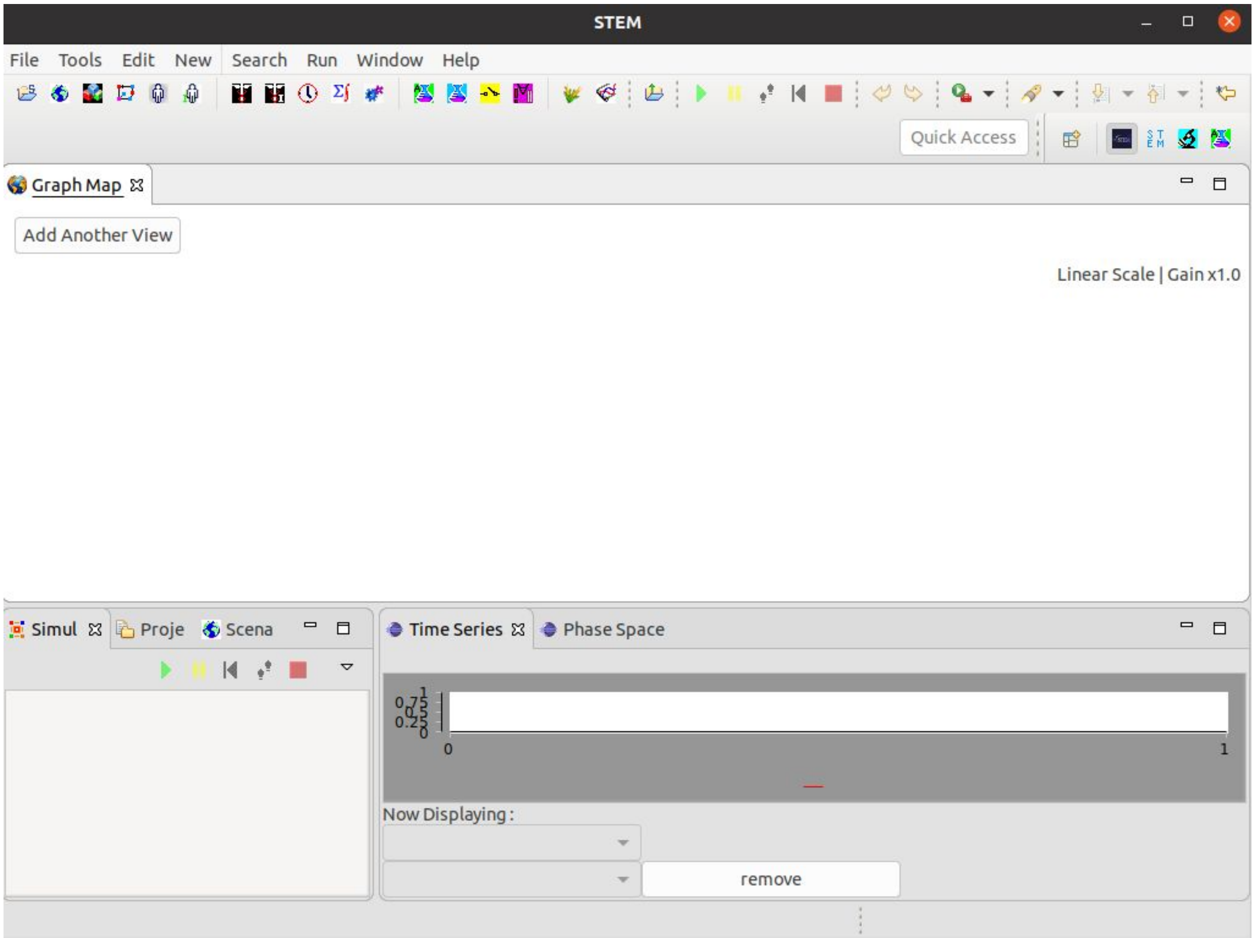
Message The requested resource [/contrast-finder-webapp-0.3.5/] is not available

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

STEM



While Tanaguru had the initial appearance of being well documented, there was no mistaking how extensive the documentation for STEM was. The build and installation instructions for the project were step-by-step and precise. This led to a much easier experience getting the project to build.



There is a reason documentation for this project must be much more precise, however. This project is leagues larger and more complex. Because of this complexity, if using the code instead of the application, it is only able to run with its specific plugins through the Eclipse IDE. This could be a major problem due to the requirement to run our tests through the terminal. Due to the complexity of the entire project, we will have to look into these issues to see if they can be resolved. Due to the unfamiliarity of Maven and Eclipse this could lead to issues seeing as we were unable to get the built-in tests running. Despite this, the extensive work and documentation that has been done on this project gives us hope that these issues will be resolved.

After deliberating between the two, the ultimate decision was made to go with Tanaguru. Despite the issues we had getting the project to deploy, for the purpose of our project we only need to test specific parts of the code, not the deployment of the software. Tanaguru is a better choice in this regard due to the extensive requirements STEM needs for its code to work, which could cause problems in our testing if we were to choose it.