
EE/CprE/SE 4920 Status Report #4

2/28/2025 – 3/13/2025

Group sdmay25-23

Project title: Code Critiquer System for the C Language and Embedded C

Client &/Advisor: Dr. Diane Rover

Team Members/Role:

James Joseph - Secure System Design, CPR E 2880 Liaison

Samuel Lickteig - Backend System Design

Alix Noble - Testing

Andrew Sand - Team Organization, CPR E 2880 Liaison

Owen Sauser - Client Interaction, Frontend System Design, CPR E 2880 Liaison

- **Status Summary**

During this status report period, the team had a critical deadline: the first demonstration to our project advisor. For the most part, the team focused on getting various essential components in a minimally operational state for the aforementioned demonstration. These functionalities included improving the analysis code to work better with CPRE 2880 lab projects, implementing basic placeholder graphs into the analytics page, finishing the ability for instructors to add students to the system, and integrating the Code Composer Studio compilation process into the static analysis. Additionally, the team is also looking ahead toward what the final deliverable will look like since the end of the semester will approach fast.

- **Project Milestones**

- Instructors can now import students into the system
- GitLab Issues have been overhauled to have Labels, Assignees, Descriptions, and Milestones
- Code Composer Studio Compilation is not integrated into the critiquer

- **Past Week Accomplishments**

- James Joseph:
 - Started work on making the analysis work with real lab code
- Samuel Lickteig:
 - Finished new student upload and small fixes
 - Added user types to flask session data for authentication

- Original code provided by previous team assumed there were only instructor accounts, so all account types could access any page
 - Working on student feedback collection and how the professor will be able to view that feedback
 - Once feature is fully implemented, student will be able to submit comments or questions on individual antipatterns found in their code
- Alix Noble:
 - Continued work on professor analytics page
 - Experimented with different Bokeh UI elements in an effort to make dynamic graphs
- Andrew Sand:
 - Overhauled the team's GitLab Issues
 - Created Milestones based on the team's priority list
 - Created Labels for Issues
 - Created, assigned, and wrote detailed descriptions for roughly twenty Issues based on the team's priority list and notes from meetings
 - This will help organize the tasks that need to be done for completion and what everyone is working on
 - Cleaned up some old GitLab Issues that have been dormant
 - Started researching final deliverable documentation
 - Looked over last semester's documentation and the final documentation from team24-34 in preparation of the next class meeting
- Owen Sauser:
 - Finalized the commands that the python code will create a Code composer studio project in the workspace.
 - Finalized the commands that they python codes use to run the project with the users code
 - Got code composer and the Tiva support files onto the server and modified the commands to run on the server.

- **Pending Issues** (If applicable)
 - James Joseph:
 - Ran into a problem where "#DEFINE GLOBAL (GLOBAL1 * GLOBAL2)" breaks the AST parsing. I will need to find a way around it or make a PR on the core library.
 - Samuel Lickteig:
 - N/A
 - Alix Noble:
 - Having difficulty with the graphing library. I would like to create more dynamic graphs, but am having troubles with it.
 - Andrew Sand:
 - N/A
 - Owen Sauser:
 - N/A

- **Individual Contributions** (Note: The "Hours Cumulative" section is independent from the members' cumulative hours from the Fall 2024 semester)

<u>NAME</u>	<u>Individual Contributions</u>	<u>Hours this Week</u>	<u>Hours Cumulative</u>
James Joseph	Fixing incompatibilities with 2880 code	10	33
Samuel Lickteig	Student upload, user session types, progress on student feedback collection	6	29
Alix Noble	Analytics page testing	5	14
Andrew Sand	Overhauled GitLab Issues and researched documentation	7	15
Owen Sauser	Code composer Studio Compilation on windows and linux	8	17

- **Comments and Extended Discussion** (Optional)
N/A

- **Plans for the Upcoming Week**

- James Joseph:
 - Fix AST code
- Samuel Lickteig:
 - Implement an admin account that will handle to creation of instructor accounts in the future
 - Do testing on features implemented this and next status report period to verify they are ready for demonstration
- Alix Noble:
 - I'm going to switch focus a bit and work on the student report that Dr. Rover requested.
- Andrew Sand:
 - Get a head start on documentation for final deliverable
 - Get a few last GitLab Issues and Merge Requests cleaned up that have been abandoned for quite some time
 - Look into another Issue to pick up
- Owen Sauser:
 - I will be making a toggle to be able to disable the code composer compilation. Both in case the compilation breaks or if they just want to speed it up.
 - Make documentation for how someone would set up the code composer compilation on Linux and Windows.

- **Summary of Weekly Advisor Meeting** (If applicable/optional)

3/10/2025:

- This meeting was scheduled as a time to demo that project thus far to our advisor, Dr. Rover
- Feedback from Dr. Rover:
 - Analytics: Should sit down at some point and discuss what filters should be for the graphs
 - Analytics: Filter options: instructor, semester, course, section, assignment, instances of specific antipatterns
 - General: Would like to get a report of some kind of which students submitted for each assignment (This could be a .txt, .pdf, .csv, etc.)
 - Analytics: Have graphs for count and percentage of submissions and another for count and percentage of students
 - Analytics: Instructors should only be able to view their sections

- Analytics: Do something with a timeline? Have a timeline of antipattern occurrences throughout a semester (Have the option to filter out different antipatterns from the timeline)
- CCS Parsing: What if we had another server running Windows that we ran CCS on it to get its output instead of installing it through the command line on Linux? (Look into pros and cons from this)
- Antipattern Feedback: Implement providing explanations or links to resources based on certain antipatterns (Have a default per antipattern, but allow instructors to customise this feedback)
- Student Feedback: Have this be per antipattern. Need to be able to have instructors/TAs view student submitted feedback
- General: Enter the bugs of the week into the system
- Look into starting to demo on March 31st