Thierry's Minions/Team25

Deliverable 4

CSCC01 Fall 2018

Verification & Code Review

Submitted To:
Saba Kiaei
Teaching Assistant
Computer Science
Department

Submitted By :
Rishabh Kaant Sharma
Joseph Sokolon
Balaji Badu
Jayden Arquelada
Edgar Sarkisian

Contents

1 Code Review Strategy

- Every task is created on its own branch. Once a member has completed their work for the task they open up a pull request to merge to the feature branch.
- One team member reviews the code, and if they approve they accept the merge request. If the member doesn't approve they send it back to the developer and request them to make the necessary changes.
- Once all the tasks have been completed, reviewed and merged into the feature branch a final PR will be created, to merge the feature into master.
- One or more team members will test the feature in its entirety and ensure that it meets the requirement. Then they will merge the feature into master. This effectively "releases the feature".
- During a code review members will follow the code review guidelines.
  - First, read through the task and what its supposed to accomplish. Then read the title and comments of the PR, to get a high level overview of what they did.
  - Second, go to the files changed section of the PR and read through all the changes made. Members should look for and point out any of the following issues:
    1. Changes made that are irrelevant to the task.
    2. Changes made that could be implemented better with a design pattern.
    3. Changes made that have or could introduce bugs to the program.
  - Thirdly the reviewer should checkout the branch locally and test themselves to see if the task was implemented correctly.
  - Finally if everything went well they can approve the changes and accept the merge request.

2 Code Review Summary

2.1 Balaji

Task/Feature	Feature-Normalization
Pull Request:	`https://github.com/CSCC01/Team25/pull/21`
Comments:	Use of Verifier interface is good for scalability in adding more verifiers in the future. Possibly add more exception classes as to have custom error messages suited for each verifier. Avoid embedded try catch blocks if possible. Overall design is working and is maintainable and scalable. Add more comments and javaDocs in various verifier classes.

2.2 Edgar

Task/Feature	Last Upload Feature
Pull Request:	`https://github.com/CSCC01/Team25/pull/15`
Comments:	Feature was tested and works. Adds foundation for adding more UI tabs in the future. Weird glitch was observed that changes the UI when you hover your mouse over the other tab. Tried fixing with the team but after an hour we decided to merge the feature and raise an Issue (`https://github.com/CSCC01/Team25/issues/16`), since the glitch doesn't directly impede the workflow.

2.3 Joey

Task/Feature	Generate Report - Task C
Pull Request:	`https://github.com/CSCC01/Team25/pull/17`
Comments:	Code is very simple and effective. Comments are good. Server.js is getting very large, violates single responsibility principle.

Task/Feature	Data Normalization - Task C
Pull Request:	`https://github.com/CSCC01/Team25/pull/8`
Comments:	The refactoring on controllers is good. They now only return what we need. The refactoring on server is good too. Keep consistent with design to keep database stuff in database.js

2.4 Rishabh

2.5 Jayden