Iteration 2 Worksheet

Paying off technical debt

Show **two** instances of your group paying off technical debt. For these two instances:

- Explain how your are paying off the technical debt.
- Show commits, links to lines in your commit where you paid off technical debt.
- Classify the debt, and justify why you chose that classification with 1-3 sentences.

Example of how to link to a commit with line number - on the top of the page, click "permalink", which will link to the most recent commit then, click the margin to choose a line: https://code.cs.umanitoba.ca/3350-summer2023/sample-project/-/blob/8e38ae9c3084d62adc4ac5fafa3b87d7d862dc26/.gitignore#L7

Make sure the path it not to main or their develop (or similar) branch, as those rapidly change.

ANSWER:

1) https://code.cs.umanitoba.ca/3350-summer2023/g3canada-3/-/commit/f5f71bdc1f2f5cba4a65dfd1b44648b04e50b05a#8ffdafa025108007fe216c32ad3867be21aafa62_16_16
    a. In this case, we are paying off the technical debt of not allotting photo information on recipes simply because we did not have the time to do so beforehand. In the linked commit, we are finally adding the info as well as the included rest of the file.
    b. This debt is deliberate and prudent. This is because we chose to leave this off, as well as knowing that this action will not cause harm to the rest of the application's functionality
2) https://code.cs.umanitoba.ca/3350-summer2023/g3canada-3/-/commit/8ff60e4ca8747426d40156474f6cd5fc7aff92c8#647c2e233cf478eeb93b68e16f8040edb57c4da0_0_16
    a. In this case, we are paying off the technical debt of repeating code that could be reusable. In the linked commit, we are finally paying off the debt by moving this functionality to another class that can handle it and hiding away the implementation its own corner.
    b. This debt is deliberate and reckless. This debt could've easily snowballed into multiple classes having the same functionality and needing extensive refactor, never mind the possibility of class specific additions to this class.

SOLID

Find a SOLID violation in the project of group `n+1` (group 13 does group 1). Open an issue in their project with the violation, clearly explaining the SOLID violation - specifying the type, provide a link to that issue. Be sure your links in the issues are to **specific commits** (not to `main`, or `develop` as those will be changed).

Provide a link to the issue you created here.

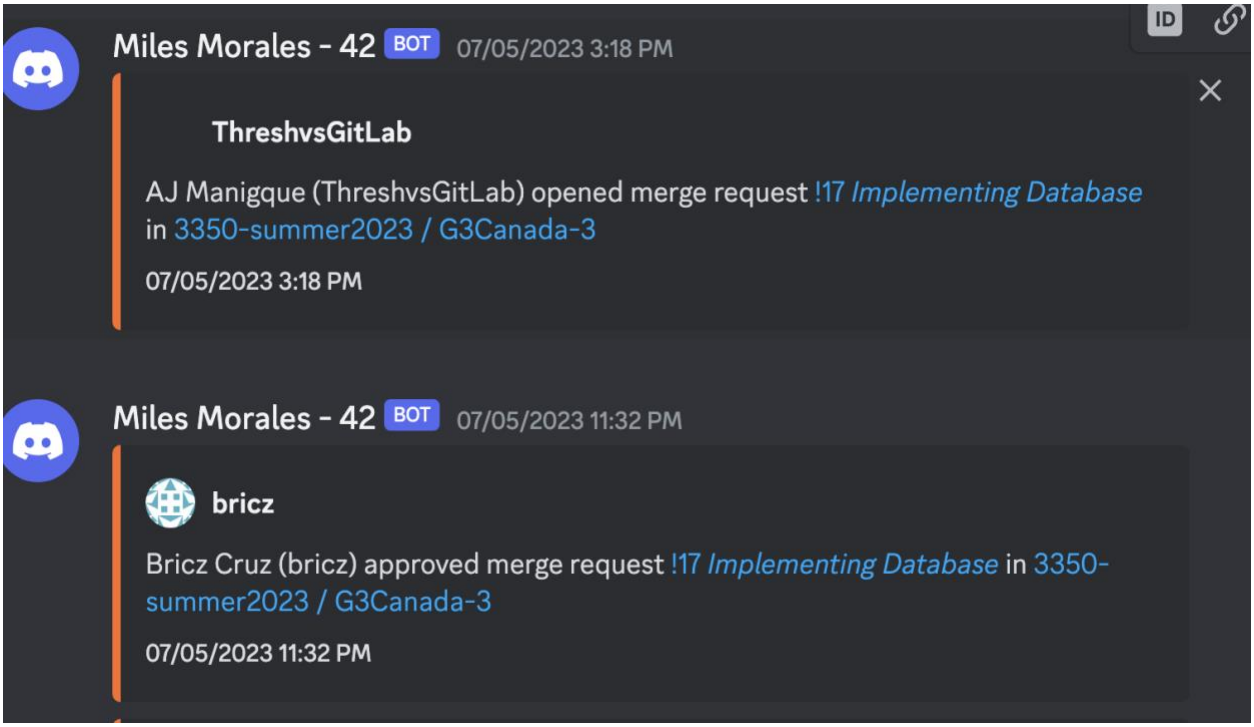
ANSWER:

https://code.cs.umanitoba.ca/3350-summer2023/funkyflamingos-4/-/issues/86

Retrospective

Describe how the retrospective has changed the way you are doing your project. Is there evidence of the change in estimating/committing/peer review/timelines/testing? Provide those links and evidence here - or explain why there is not evidence.

ANSWER:



We have decided to be a lot more communicative in terms of our thoughts and ideas, as well as setting hard deadlines to accomplish major tasks to avoid any unnecessary stressors. This has resulted in quicker resolve times as well as easier peer reviews and timelines so that expected project contributions can be time alotted.

Design patterns

Show links to your project where you use a well-known design pattern. Which pattern is it? Provide links to the design pattern that you used.

Note: Though Dependency Injection is a programming pattern, we would like to see a programming pattern other than Dependency Injection.

ANSWER:

Singleton Pattern
https://code.cs.umanitoba.ca/3350-summer2023/g3canada-3/-/commit/6e27b2c88d256709fa92a85b905af795966739cb#c51d79395c4078e3022f247d4bfe24593df6059b_10_12

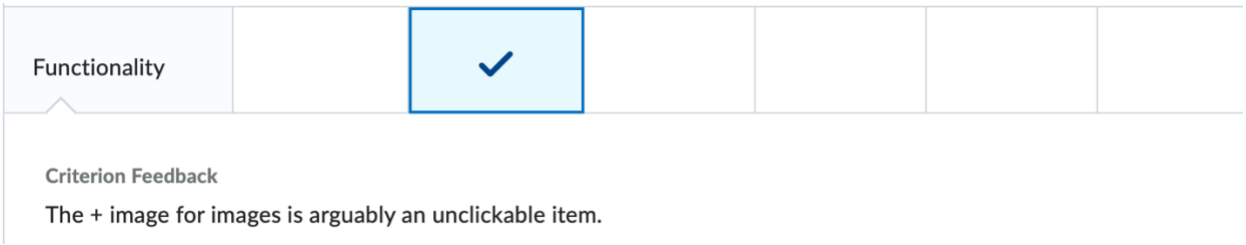
Iteration 1 Feedback fixes

Provide a link to an issue opened by the grader.

Explain what the issue was, and why it was flagged. Explain what you did to refactor or fix your code to address the issue. Provide links to the commits where you fixed the issue.

No issues? Show 1 more piece of technical debt being paid off. Or demonstrate how you are not adding debt by being prudent in your new additions.

ANSWER:



In this issue, we had an unimplemented UI that can be confusing to users. This is detrimental because this unfinished feature was shipped in the iteration even though it should not have been.

```
46
47  +     private ImageView recipeImage;
48  +
```

https://code.cs.umanitoba.ca/3350-summer2023/g3canada-3/-/commit/f5f71bdc1f2f5cba4a65dfd1b44648b04e50b05a#b3e232fe8c2cdc0da8685caa18d6dd518ac9850f_35_47

We resolved this issue by now adding an image view that can handle the functionality for this UI.