

Feature: Bingo Board

Developer: Diego Garcia

Date submitted: 04/12/2024

Reviewer: Julie Reyes

Date Reviewed: 04/16.2024

Major positives

- Nice design that isn't over complicated, nice and efficient grabbing the data it needs to load the data. Nice detail as well in the designs over what is happening with each line.
- Has failure cases to display the errors that can occur with his designs for each of his success cases. Makes sure to accurately describe what is happening and the result/return the failure causes.
- Has a database ERD to show connection to the UserAccount information, and displays all the columns of data he has on the gig table. Not a super complicated table but has right good implementation not setting posterUsername which is a reference to primary key which then allows more than 1 gig per username. Important detail to make sure the functionality of his feature is correct.
- Code implementations he has shown in git has very well implemented code that builds and runs well. Did a good job implementing it into the current codebase seamlessly and made sure to use tokens with his fetch calls. Still has ways to go but shows a lot of promise with his overall implementation of his design even with such little documentation which would help the process go faster.

Major negatives

- Does not make sure to include codelines in his sequence diagram lines all the time. Would make implementation easier to integrate into the codebase and make it more readable if it were to be intended for another developer to implement.
- Does not have full implementation of backend tests. Only has text file for DataAccessLayer not in service up to frontend. Even the DAL test file only has one simple test method that doesn't cover the full scope of what he should be testing for a function let alone a whole class/file.
- Has no designs for the functionality/flow of the front end. How is the user supposed to interact with the feature, how will they access/move across the data we load?

Unmet Requirements

- Services and frontend testing (only has DAL test)

- Further frontend design flow documentation

Design Recommendations

- Spend more time designing sequence diagrams with actual function/method calls to make implementation easier. A little more time designing would work out all the logic you need to code in.
- Front end design looks really good but a little more designing/documentation of the flow will help you flesh out some of the questions/ concerns you might have with your UI. Things like how the user will interact with the interface, most logical flow, handling user error, etc.

Test Recommendations

- Testing as you implement your designs through each layer would help get code working and running properly and have tests done once you finish the frontend, instead of having to go back to add them all in. This would also speed up the time it would take you to debug your fixes as you will properly be able to build your code up testing step by step always knowing what layer is causing your error. Make sure to test all cases, not just passing running cases. You need to make sure your code can properly catch and handle all exception/error scenarios, as well as requirements like load time.