GitHub Enterprise Users managed by North Carolina State University

Edit New issue

Team B Code Review #114

Open

wrstickl_ncstate opened 1 hour ago · edited by wrstickl_ncstate Edits

Viewing (#90) for both frontend and backend implementation of UC2

Backend Files: All edited items*.java files, controller, entity, dto, server

Frontend Files: ListItemsComponent.jsx

Create sub-issue

wrstickl_ncstate 1 hour ago · edited by wrstickl_ncstate Edits Author

Going through Guiding Questions.

Code inspection of Tiger's 1B implementation.

- This inspection looked at the pull requests with numbers Ttphians/uc2 implementation #90 for the backend & frontend of UC2, which is for adding items.

Code Inspection Checklist

Is the code Javadoced with meaningful comments that help the reader understand the method?

- Code has proper javadoc for all classes and methods, could use some more elaboration in class comments.

Are the comments up to date with the current implementation of the method?

- Yes, they are up to date.

Are there ways to simplify the implementation or reduce redundancy?

- I don't think so. The every addition to the frontend adds to the user experience, and all the backend methods have their uses and are implemented efficiently.

Is the functionality at the right location in the architecture or should it be moved to a different layer?

- Yes, the functionality is at the right location.

Is there appropriate input validation? (Just b/c the front end has a check for invalid inputs, doesn't mean there isn't a way to get invalid input to the backend.)

- The front end does have checks for some invalid inputs, rest are handled in the backend with api responses.

Does the implementation meet the requirements?

- The implementation works great and looks very good. The frontend extends on Bennett's design very well and shows Tiger's own style well.

Is there missing functionality?

- No.

Are there error paths that aren't handled?

- Invalid inputs are handled by frontend and some invalid integers are handled in the backend.

Are the variable names meaningful?

- Yes, they are meaningful.

Test Inspection Checklist

Are the test ids or test methods unique?

- The test methods unique.

Are descriptions, repeatable, specific, and unambiguous?

- The descriptions fit the need for UC2.

Are expected results clearly stated?

- The expected results are clearly stated.

Are preconditions defined for each test?

- Yes, preconditions are defined for each test.

Is the test behavior tied to a requirement?

- Yes, the test behavior is tied to a requirement.

Does each test accurately verify a specific requirement?

- Each test accurately verifies a specific requirement.

Does the test evaluate the right results to know that it passed?

- Yes.

Do the tests sufficiently cover the requirements? Are more tests needed?

- 3 test are enough to sufficiently cover all requirements.

Overall Summary of Tiger's Code

- Tiger has done a very good job with the implementation of both the backend and frontend. The implementation used the skeleton classes made for the design well and the frontend extends on Bennett's new design very well and fits with the products overall theme. Great job!

wrstickl_ncstate added a sub-issue 22 minutes ago

Frontend #115

wrstickl_ncstate removed a sub-issue 22 minutes ago

Frontend #115

wrstickl_ncstate 10 minutes ago Author

Frontend Files

ListItemsComponent.jsx

- Fits very well with product theme and rest of UI components
- Good comments throughout
- Could use overall javadoc comment at top of class detailing frontend

wrstickl_ncstate 5 minutes ago · edited by wrstickl_ncstate Edits Author

Backend Files

ItemServiceImpl.java

- Good use of original class.
- Consistent javadoc throughout file.

ItemControllerTest.java

- I think we should try to implement some invalid tests for creating items.
- Also think some tests are needed to test more of the HTTP errors when calling with invalid values.

ItemServiceTest.java

- Good use of empty fields for testing errors.
- As in controller test, more incorrect and boundary value tests could be added.

Add a comment

Write Preview

Use Markdown to format your comment

Paste, drop, or click to add files

Close issue Comment

Metadata

Assignees

No one - Assign yourself

Labels

No labels

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

Create a branch for this issue or link a pull request.

Notifications

Unsubscribe

You're receiving notifications because you're subscribed to this thread.

Participants

Transfer issue

Duplicate issue

Lock conversation

Pin issue