CSE 360

Team Project Phase 3 Report

Team <Wednesday 34>

Team Member Names:

1. Eliza Chook
2. Anas Mohamed Riyaz
3. Nathaniel Teo
4. Isabella Tunon-Robinson
5. Tharun Vimalaadhithan
6. Zakia Mohammadi

1. The List of User Stories to be Implemented

Isabella Tunon-Robinson:

1. As a reviewer, I can create reviews of potential answers so the rest of the class can benefit from my experiences.
2. As a reviewer, I can update a review based on inputs received and new insights gained, and others can see my new review that contains a link to the previous review so others can see both my original and updated review and appreciate the change.

Eliza Chook:

1. As a student, I can request to be given the role of a reviewer so my classmates can benefit from my experience and reduce the number of potential answers they need to read. An instructor must approve this request, and the role is assigned upon approval.

Nathaniel Teo:

1. As a student, I can establish and manage a list of reviewers I trust and assign each a weight, so my curated list of answers serves me well.
2. As a student, I can read the reviews of potential answers to questions and add promising reviewers to my potential trusted reviewers list so I can establish and manage a list of trusted reviewers.

Anas Mohamed Riyaz:

1. As a student, I can see when a reviewer in my list of trusted reviewers updates a review, so I can quickly benefit from the update.

Zakia Mohammadi:

1. As a reviewer, I can establish and maintain a Reviewer Profile, which includes details about my experience, a list of all the reviews I have provided, and feedback I have received from students who have used my reviews.

Tharun Vimalaadhithan:

1. As a student, I can provide private feedback to a reviewer of a potential answer so the reviewer can provide better reviews.
2. As a reviewer, I can access a list of my reviews and see the number of private feedback messages for each so I can manage and improve my reviews.

2. Implementation Plan

2.1. Allocated Work

2.1.1. Isabella Tunon-Robinson

- Create Review Function; April 3rd, 2025; March 30, 2025
- Update Review Function; April 3rd, 2025; March 30, 2025

2.1.2. Eliza Chook

- Request Reviewer Role Function; April 2nd 2025; April 1st 2025

2.1.3. Nathaniel Teo

- Weighted Reviewer Function; April 3rd 2025; April 3rd 2025
- Trusted Reviewers Function; April 3rd 2025; April 3rd 2025

2.1.4. Anas Mohamed Riyaz

- Notification on Reviewer Update System; April 3rd 2025; April 3rd 2025

2.1.5. Zakia Mohammadi

- Reviewer Profile; April 3rd 2025; March 28th 2025

2.1.6. Tharun Vimalaadhithan

- Private Feedback Function; April 3rd 2025; April 3rd 2025
- View List of Reviews and Messages Function; April 3rd 2025; April 3rd 2025

2.2. Schedule of Standup Meetings and Notes

2.2.1. <Week 1 23rd - 29th March>

- ***21th March 2025; 10.30am - 12.00pm***
  - Eliza Chook; Attended meeting for the test case distribution for HW3 and wrote down summary of the meeting
  - Anas Mohamed Riyaz; Forgot to attend but read through the summary and view recording of the meeting and understood the discussion
  - Zakia Mohammadi; Attended meeting for the test case distribution for HW3
  - Nathaniel Teo; Attended meeting for the test case distribution for HW3
  - Isabella Tunon-Robinson; Attended meeting for the test case distribution for HW3
  - Tharun Vimalaadhithan; Attended meeting for the test case distribution for HW3

2.2.2. <Week 2 23rd - 29th March>

- ***26th March 2025; 10.30am - 12.00pm***
  - Eliza Chook; Attended meeting in the discussion of plan of action for phase 3 and wrote down summary of the meeting
  - Anas Mohamed Riyaz; Forgot to attend but read through the summary and view recording of the meeting and understood the discussion

– Zakia Mohammadi; Attended meeting in the discussion of plan of action for phase 3
– Nathaniel Teo; Could not attend due to clash in schedule but read through the summary and view recording of the meeting and understood the discussion
– Isabella Tunon-Robinson; Attended meeting in the discussion of plan of action for phase 3
– Tharun Vimalaadhithan; Forgot to attend but read through the summary and view recording of the meeting and understood the discussion

2.2.3. <Week 3 30th March - 4th April>

- ***2nd April 2025; 10.30am - 12.00pm***
  – Eliza Chook; Attended meeting in updating group on phase 3 progress and wrote down summary of the meeting
  – Anas Mohamed Riyaz; Attended meeting in updating group on phase 3 progress
  – Zakia Mohammadi; Attended meeting in updating group on phase 3 progress
  – Nathaniel Teo; Could not attend due to clash in schedule but read through the summary and view recording of the meeting and understood the discussion
  – Isabella Tunon-Robinson; Attended meeting in updating group on phase 3 progress
  – Tharun Vimalaadhithan; Forgot to attend but read through the summary and view recording of the meeting and understood the discussion

3. JUnit Test to be Implemented

3.1. Junit Test 1: Create new Review; Assigned to: Isabella Tunon-Robinson

The purpose of this test is to create a new review of an answer and save it to test the functionality of adding a new review onto the review list.

3.2. Junit Test 2: See latest review for answer; Assigned to: Isabella Tunon-Robinson

The purpose of this test is to check that any role, other than just the reviewer, can see the latest reviews

3.3. Junit Test 3: Update review; Assigned to: Isabella Tunon-Robinson

The purpose of this test is to check that the update/overwrite review functionality works

3.4. Junit Test 4: Update review to see previous review; Assigned to: Isabella Tunon-Robinson

The purpose of this test is to check that the update/overwrite review functionality works and that it links to the previous review list.

3.5. Junit Test 5: View old review from new review's previousReviewId; Assigned to: Isabella Tunon-Robinson

The purpose of this test is for non-reviewers to check that they can read through old reviews

3.6. Junit Test 6: Update nonexistent review (negative test); Assigned to: Isabella Tunon-Robinson

The purpose of this test is to prove that nothing would happen if you were to update a nonexistent review. It would stay empty.

3.7. Junit Test 7: Duplicate review for same answer is allowed; Assigned to: Isabella Tunon-Robinson

The purpose of this test is to check that a duplicate review for the same answer is allowed. I don’t think having dupes is a big deal as different reviewers can think similar

3.8. Junit Test 8: Test Request Reviewer Role; Assigned to: Eliza Chook

The purpose of this test is to test if a user can request for the reviewer role. It verifies if the request is in the database once requested.

3.9. Junit Test 9: Test Accept Review Request; Assigned to: Eliza Chook

The purpose of this test is to test if the instructor can accept requests. It verifies if the user has the reviewer role after action is done and also tests if request is removed once action is done.

3.10. Junit Test 10: Test Deny Review Request; Assigned to: Eliza Chook

The purpose of this test is to test if the instructor can deny requests. It verifies if the user does not have the reviewer role after action is done and also tests if request is removed once action is done.

3.11. Junit Test 11: Test Review Request Removed After Action; Assigned to: Eliza Chook

The purpose of this test is to test if requests are removed after the instructor accepts or denies requests. It verifies this by accepting and denying requests and checks if the requests are still in the database.

3.12. Junit Test 12: Test No Duplicate Review Requests; Assigned to: Eliza Chook

The purpose of this test is to make sure that users cannot have duplicate requests. It verifies this by getting a student to send a request twice and checks the database if only one request from the same student is in there.

3.13. Junit Test 13: Test Reviewer Profile Creation; Assigned to: Zakia Mohammadi

This tests a reviewer profile object that is created with given parameters. Its purpose is that all fields like ID, username, experience, and createdDate are correctly set while instantiating.

3.14. Junit Test 14: Test Reviewer Role Assignment; Assigned to: Zakia Mohammadi

This test is to confirm whether a user with role “reviewer” has the role correctly stored and is correctly returned by getRole(). Basically confirming that the role assignment logic works correctly and a reviewer is recognized by the system.

3.15. Junit Test 15: Test Created Date Not Null; Assigned to: Zakia Mohammadi

The purpose of this test is to check if the createdDate field is not null in the ReviewerProfile after a profile is created. This is to validate that the creation date is correctly captured and stored.

3.16. Junit Test 16: Test Reviewer Username Matches; Assigned to: Zakia Mohammadi

This test is to compare the expected reviewer username with the actual one that is returned by the ReviewerProfile object. Its purpose is to ensure that the profile is linked to the correct user and there is no mismatch.

3.17. Junit Test 17: Test Multiple Role In User; Assigned to: Zakia Mohammadi

This is to verify that the roles list in User supports multiple roles and the reviewer role is working correctly.

3.18. Junit Test 18: Test Show Recent Non-Updated Review; Assigned to: Anas Mohamed Riyaz

Verifies that a single, standard review posted recently (within 7 days) by a reviewer on the student's trusted list is correctly retrieved and displayed by the student home page logic.

3.19. Junit Test 19: Test Show Recent Updated Review; Assigned to: Anas Mohamed Riyaz

Checks if a review that has been updated recently by a trusted reviewer is correctly retrieved, displayed, and specifically marked as an "Updated Review" by the student home page logic.

3.20. Junit Test 20: Test Show Multiple Trusted Updates & Ignore Untrusted;Assigned to: Anas Mohamed Riyaz

Validates that the system logic correctly aggregates recent updates from multiple trusted reviewers and crucially ensures that recent updates from reviewers not on the student's trusted list are ignored.

3.21. Junit Test 21: Test Ignore Old Reviews from Trusted Reviewers; Assigned to: Anas Mohamed Riyaz

Tests the time boundary condition, ensuring that reviews from trusted reviewers that are older than the defined cutoff period are correctly filtered out by the logic and not displayed as recent updates.

3.22. Junit Test 22: Test Show No Updates When No Trusted Reviewers;Assigned to: Anas Mohamed Riyaz

Verifies the behavior when the student has not added any reviewers to their trusted list. It ensures that no updates are fetched or displayed by the student home page logic in this scenario, even if reviewers have recent activity.

3.23. Junit Test 23: testAddFeedbackValid; Assigned to: Tharun Vimalaadhithan

This test creates a Feedback object for a specific review using a valid student username and message.

It adds the feedback to the database and retrieves an auto-generated feedback ID.

The test asserts that the generated feedback ID is greater than zero, confirming successful insertion.

Finally, it retrieves the list of feedbacks for the review and verifies the inserted feedback is present.

3.24. Junit Test 24: testGetFeedbackCount; Assigned to: Tharun Vimalaadhithan

This test inserts three separate feedback entries for a designated review.

It executes a database query to count all feedback records associated with that review.

The test asserts that the retrieved count exactly equals three.

This confirms that the counting functionality correctly reflects multiple feedback insertions.

3.25. Junit Test 25: testRetrieveFeedbacksForReview; Assigned to: Tharun Vimalaadhithan

This test starts by inserting a single feedback entry for a given review ID.

It retrieves the list of all feedback entries for that review from the database.

The test asserts that the list is not null and contains at least one feedback record.

It further validates that the feedback message matches the expected content.

3.26. Junit Test 26: testAddUserRole; Assigned to: Nathaniel Teo

This test inserts a user and a reviewer profile into the database.

It updates the reviewer profile to have a weight of 1, thus making it a trusted reviewer.

The test asserts that the first element of the list of trusted reviewers is not empty, thus ensuring that the user has been properly added.

3.27. Junit Test 27: testModifyReviewerWeightSuccess; Assigned to: Nathaniel Teo

This test modifies the weight of a reviewer (trust level).

It updates the reviewer profile to have a weight of 10, which is within the bounds of a valid input.

The test asserts that the first element of the list of trusted reviewers has a weight of 10, thus ensuring that the weight of the user has been properly modified.

3.28. Junit Test 28: testModifyReviewerWeightPositiveFail; Assigned to: Nathaniel Teo

This test modifies the weight of a reviewer (trust level).

It updates the reviewer profile to have a weight of 11, which is outside the maximum bound of a valid input.

The test asserts that the first element of the list of trusted reviewers must not have a weight of 11, thus ensuring that the input handling is done correctly.

3.29. Junit Test 29: testModifyReviewerWeightNegativeFail; Assigned to: Nathaniel Teo

This test modifies the weight of a reviewer (trust level).

It updates the reviewer profile to have a weight of -1, which is below the minimum bound of a valid input.

The test asserts that the first element of the list of trusted reviewers must not have a weight of -1, thus ensuring that the input handling is done correctly.

3.30. Junit Test 30: testRemoveReviewer; Assigned to: Nathaniel Teo

This test removes the reviewer from the trusted list.

It updates the reviewer profile to have a weight of 0, which is the requirement to make a reviewer removed from the trusted list.

The test asserts that the first element of the list of trusted reviewers must be empty, thus ensuring that the user has been properly removed.

4. Implementation

4.1. URL and access path to the source code and the architecture and design documents

Link to doc:
https://docs.google.com/document/d/1bxBHZXi3PO9ROpl20H6npaTsNBeUG0jRVbWMEdOTu4g/edit?usp=sharing

4.2. URL and access path to a folder of Javadoc HTML Files

Link: https://github.com/nteo/CSE360Wed34/tree/main/JavaDocs

5. GitHub Repository

5.1. The URL to access the GitHub Repository with a ReadMe file

Link: https://github.com/nteo/CSE360Wed34

5.2. Access to the Team’s Solution for TP3

Link to UML:
https://docs.google.com/document/d/1bxBHZXi3PO9ROpl20H6npaTsNBeUG0jRVbWMEdOTu4g/edit?tab=t.0

Link to JavaDocs: https://github.com/nteo/CSE360Wed34/tree/main/JavaDocs

5.3. Access to Screencasts

5.3.1. Screencast for how the new code works, how it was implemented, how it was validated using JUnit testing, and how it works from the various role-players that would use the application

Screencast Link:

https://drive.google.com/file/d/1xbEu0DXDZ12seGdOTFnQ_clQokQMr-rf/view?usp=sharing

5.3.2. Screencast showing and explaining the alignment from the stakeholders’ needs as described in the product vision through the architecture and detailed design to the validation of the code using JUnit.

Screencast Link:
https://drive.google.com/file/d/1X5rulL4CDqNjHOR6IA-ChvR2Uwp-D2vQ/view?usp=sharing

5.3.3. Screencasts from each Standup Meeting

Meeting 1 (26th March):
https://drive.google.com/file/d/1jfdlmjyoDprNpayw__Bdk_ZAOqHmef40/view?usp=drive_link

Meeting 2 (2nd April):

https://drive.google.com/file/d/1gev14yxGPyX1CXAt56bg9FtK8EbeUm8V/view?usp=drive_link

6. Appendix A: Credit Sheet

This appendix lists the members of the team and includes a description of the contribution that team member has made to this submission. Each team member must provide text for their contribution at a team meeting just prior to the submission of this deliverable and the entire team must agree. If a team member fails to provide this information and/or does not participate in the agreement process to fill out this table, that member will receive no credit for the submission of this deliverable.

Be very brief with your description. It should be no more than six lines of text that can fit in this space, so the entire table fits on one page.

Team Member Name	Contributions
Isabella Tunon-Robinson	Created reviews and implemented so that you can update them and everyone (not just reviewers) can look at reviews Implemented test cases Record for screencast 1 and 2 Edited the screencasts and merged them all together Made TP2 UML diagram Filled in documentations
Eliza Chook	Record the meetings conducted Filled in documentations Implement request reviewer role related functions Implemented test cases Record for screencast 1 and 2 Made TP3 UML diagram
Nathaniel Teo	Implemented weighted reviewer list related functions Implemented trusted reviewer list related functions Implemented test cases Record for screencast 1 and 2 Did the Javadoc HTML
Anas Mohamed Riyaz	Implemented notification of reviewer update system Implemented test cases Debugged reviewer profile issues Record for screencast 1 and 2
Zakia Mohammadi	Implemented reviewer profile and related functions Implemented test cases Record for screencast 1 and 2 Did the Javadoc HTML
Tharun Vimalaadhithan	Implemented private feedback function between students and reviewers Implemented review list and private messaging function for reviewer Implemented test cases Record for screencast 1 and 2

7. Appendix B: Team Norms

No changes to team norms has been made as of now