Lab Test 5

By: Sathira Williams

**Implemented User Stories:**

Create an account and login - Isaiah

Pre-selected library of books - Sathira

Comment on book – Vince

Citation - Sathira

Searching for a book – Jet

Ratings and reviews – Abbey

**Manual Test**:

On startup click on the profile icon to login to the account:

Graphical user interface, application, Word

Description automatically generated

After logging in click the home button on the sidebar and open any book, and then press “Open Comments”.

Graphical user interface, application, Word

Description automatically generated

Problem Descriptions:

Problem Description 1: After logging in and entering a comment section of the book, posting a comment does not display the user who posted the comment.

Suggested Fix: Adding a JLabel to each of the comments that displays the user who posted the comment.

Problem Description 2: After logging in and entering a comment section of the book, posting a comment is not differentiated per book, instead there is only one global comment section for all the books instead of being separated by book.

Suggested Fix: Add an identifier for each comment in the DB that associates with the book, and then only display the comments that match with the book for the comment section.

Problem Description 3: After logging in and entering a comment section of the book, the comment section GUI has a dark background not matching with the rest of the GUI, and should be centered horizontally for design consistency.

Suggested Fix: Adjust the Miglayout for the comment section GUI components and set the background color to white for the components.

**Code Review:**

Problem Descriptions 1: The getComments() method in the Forum class does not close the ResultSet object after it has finished reading from it, which can cause a resource leak if the method is called multiple times.

Component Name: Forum.java

Suggested Fix: To avoid the resource leak, the ResultSet object should be closed using the close() method after it has been read from in the getComments() method.

Problem Descriptions 2: The Rating class does not perform any input validation or error handling when setting the rating value, which can cause unexpected behavior if the value is set to an invalid input.

Component Name: Rating.java

Suggested Fix: To ensure that the rating value is within a valid range, the setRating() method should perform input validation to ensure that the rating value is between 0 and 5. If an invalid rating value is provided, the method should throw an IllegalArgumentException.

Problem Descriptions 3: The editMessage() method in the Review class does not perform any input validation or error handling when setting the new message value, which can cause unexpected behavior if the value is set to an invalid input.

Component Name: Review.java

Suggested Fix: To ensure that the new message value is not empty, the editMessage() method should perform input validation to ensure that the new message value is not an empty string. If an empty string is provided, the method should throw an IllegalArgumentException.