Project Handover Document 5

Group U

July 1, 2024

Authors

Muhammad Bilal Saeed, Ali Hyder, Tanya Tom

Introduction

This document provides a comprehensive overview of the Learning Buddy project, focusing on the recent bug fixes implemented as part of Exercise 4. This handover is intended to facilitate a smooth transition for future development and maintenance of the project.

Project Overview

Learning Buddy is an educational application designed to facilitate learning through quizzes and access to general information. The project aims to provide users with an engaging platform to enhance their knowledge in various subjects.

Initial Implementation

The initial implementation of Learning Buddy featured a Command Line Interface (CLI) in Java. Users interacted with the application through text-based commands, selecting quiz categories and difficulty levels.

Key Features:

- Quiz functionality with category and difficulty level selection.
- Command-line interaction for a simple and direct user experience.
- Database integration for managing quiz questions and categories.

Evolution

Significant enhancements were made to transition from a text-based CLI to a more user-friendly Graphical User Interface (GUI) using JavaFX. This change improved the application's accessibility and usability.

Enhancements:

- **Improved CLI Code:** Refactored for readability and efficiency, with added error handling and database integration.
- **New GUI Code:** Implemented with JavaFX, featuring visual feedback, enhanced interactivity, and better user experience.

Testing

Both black box and white box testing methodologies were employed to ensure the functionality and reliability of the application.

Black Box Testing:

- Focused on user interface interactions, navigation flow, and overall functionality.
- Validated various user actions, such as starting quizzes, reading general information, and exiting the application.

White Box Testing:

- Included unit tests for individual classes and components.
- Verified internal logic, input validations, and database operations.

Bug Fixes

1. Fix for 8th White Box Test Case:

- **Issue:** The 8th white box test case, which tested setting whitespace strings using setters, failed.
- **Solution:** The setters for the Category class were updated to correctly handle and reject whitespace strings. An `IllegalArgumentException` is now thrown if a whitespace string is provided.
- **Result:** This fix was verified by re-running the test case, which now passes successfully.

2. Runtime Error Fix:

- **Issue:** An error occurred while running the program, causing unexpected termination.
- **Solution:** The root cause of the runtime error was identified as a null pointer exception during the database connection setup. The initialization process was revised to ensure that all required resources are available before proceeding.
- **Result:** The program now runs without errors, verified through multiple test runs and scenarios.

Summary of Fixes

The following test cases were specifically verified after the fixes:

White Box Test Cases:

- **TC-08:** Fixed handling of whitespace strings.
  - **Precondition:** Create a Category object with initial values.
  - **Steps:**
    1. Attempt to set whitespace strings using setters.
    2. Verify that `IllegalArgumentException` is thrown.
    3. Verify that getters return initial values.
  - **Expectation:** `IllegalArgumentException` is thrown for whitespace strings, and getters return initial values.
  - **Observation:** Pass

Additional Verifications:

- **Database Initialization:**
  - **Precondition:** Ensure the database is clean and accessible.
  - **Steps:**
    1. Initialize the database.
    2. Verify no null pointer exceptions occur.
  - **Expectation:** Database initializes without errors.
  - **Observation:** Pass
- **Program Execution:**
  - **Precondition:** Application setup with all dependencies.
  - **Steps:**
    1. Run the application.
    2. Navigate through the main features: start quiz, read general information, and exit.
  - **Expectation:** Application runs smoothly without crashes.
  - **Observation:** Pass

Recommendations

To ensure continued stability and improvement of the Learning Buddy application, the following steps are recommended:

1. **Regular Code Reviews:** Conduct frequent code reviews to identify potential issues early and ensure adherence to best practices.
2. **Automated Testing:** Implement automated tests for both unit and integration testing to streamline the validation process for future changes.

3. **User Feedback:** Gather user feedback regularly to understand their needs and identify areas for improvement.
4. **Documentation:** Maintain thorough documentation of all changes, enhancements, and bug fixes to facilitate future development and troubleshooting.

Conclusion

The identified bugs in the Learning Buddy application have been successfully fixed, improving the overall stability and functionality of the program. Continuous testing and adherence to recommended practices will ensure the application remains robust and user-friendly.

References

- Detailed test cases and results are documented in the attached `test_description.pdf`.
- For further maintenance procedures, refer to the maintenance documentation provided with the previous handover documents.