**TEST SUITE**

**Automated Grading and Feedback Tool for Java**

**Team C – Street Coders**

## **Introduction**

The Test Suite has been created in conjunction to the test plan for Automated Grading and Feedback tool for Java. This document will clearly identify what the test deliverables will be and the desired results and results obtained

## **Test Cases and Results Log:**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements ID | Description | Test Case | Final Output |
| R 3.1.2 | Verifying that student and the instructor should have separate views. | * The student logs in with student credentials and selects role as student   **Expected output**  **Student views only student page**   * If the student logs in with student credentials and selects role as instructor   **Expected output**  **Error Message: “Incorrect role selected”**   * If The instructor logs in with instructor credentials and selects role as student.   **Expected output**  **Error Message: “Incorrect role selected”** |  |
| R 3.1.3 | Verifying that users should be redirected to their respective pages based on the login credentials. | * If the student logs in with student credentials and selects role as instructor   **Expected output**  **Error Message: “Incorrect role selected”**   * If The instructor logs in with instructor credentials and selects role as student.   **Expected output**  **Error Message: “Incorrect role selected”** |  |
| R 3.1.4 | Verifying that user login should be have both user id and password as the mandatory fields. | * If the student and instructor doesn’t enter any login credentials and tries to login   **Expected output**  **Error Message: Please enter the Username and Password.** |  |
| R 3.1.5 | Verifying that user tries to login without entering the user-id and password | * If the student and instructor doesn’t enter any login credentials and tries to login   **Expected output**  **Error Message: Please enter the Username and Password.** |  |
| R 3.1.6 | Verifying that user enters the invalid password | * If user tries to login with invalid password   **Expected output**  **Error Message: The password entered is incorrect** |  |
| R 3.1.7 | Verifying that user enters wrong user-id | * If user tries to login with invalid User-id   **Expected output**  **Error Message: Incorrect user-id entered** |  |
| R 3.1.8 | Verifying that user enters invalid user id and password | * If student and instructor enters wrong credentials and tries to login   **Expected output**  **Error Message: user-id and password mismatch** |  |
| R 3.1.9 | Verifying that login is successful | * If user enters valid credentials   **Expected output**  **Alert Message: You are logged in successfully** |  |
| R 3.2.2 | Verifying that student should have an option to upload the Java assignment. | * If student uploads empty zip file   **Expected output**  **Error Message: We don’t find any java files in your zip file** |  |
| R 3.2.3 | Verifying that java assignment should be in the form of zipped folder. | * If user upload a non-zipped folder   **Expected output**  **Error Message: The file format is incorrect.** |  |
| R 3.2.4 | Verifying that the file is uploaded successfully | * If user uploads zipped folder   **Expected output**  **Alert Message: Successfully uploaded** |  |
| R 3.2.5 | Verifying that user fails to upload zipped folder | * If user uploads incorrect zip folder   **Expected output**  **Error Message: The folder has incorrect files.** |  |
| R 3.3.2 | Verifying that field where the grade is viewed should be non-editable. | * The student tries to edit the existing grade of the particular assignment   **Expected output**  **Error Message: The grade is non-editable.** |  |
| R 3.3.3 | Verifying that Student should also have the feature to view the feedback once the assignment is graded by the instructor. | * The student tries to edit the feedback file for particular assignment   **Expected output**  **Error Message: Feedback file is non-editable** |  |
| R 3.3.4 | Verifying that Student Feedback consists of detailed view of the errors along with the marks deducted. | * All the error messages should be recorded into file   **Expected output**  **“All the error messages are recorded** |  |
| R 3.3.5 | Verifying that feedback file should be downloadable and non-editable and should only be viewed. | * **The user tries to download the feedback file**   **Expected output**  **Alert Message: The report is successfully downloaded.** |  |
| R 3.4.2 | Verifying that instructor should be able to view all the students’ java assignments in zipped format. | * If instructor doesn’t able to see zipped folders   **Expected output**  **Error Message: The assignment folder is not available** |  |
| R 3.4.3 | Verifying that instructor should have an option to upload various test cases that needs to be executed against the students written java code. | * The uploaded test case document is incorrect format   **Expected output**  **Error Message: Unable to read to the test case**   * If test case document is empty   **Expected output**  **Error Message: The file is empty** |  |
| R 3.4.4 | Verifying that input test cases document is uploaded the respective file should be displayed on the page. | * The instructor uploads test case document   **Expected output**  **Message:** Uploaded successfully   * The program reads the test case document that is unavailable   **Expected output**  **Error message: File not found** |  |
| R 3.4.5 | Verifying that instructor also should be able to include the expected output for each test case in the test cases document in a specific format. | * The test case document doesn’t have expected output document   **Expected output**  **Error Message: The expected output is empty** |  |
| R 3.4.6 | Verifying that instructor should have an option to upload the validation file that needs to be compared with the source code while grading the assignment. | * The instructor uploads the validation file with incorrect format.   **Expected output**  **Error Message:** File format is incorrect |  |
| R 3.4.7 | Verifying that validation file should consist of the class names, method names, method argument names and constants that needs to be checked for in the student submitted code. | * If validation file contains no validations   **Expected output**  **Error Message: Your file doesn’t contain any validations** |  |
| R 3.4.8 | Verifying that validation file should also consist of the rubric which consists of the mandatory functions and their corresponding marks. | * If validation file doesn’t have rubric   **Expected output**  **Error Message: Your file doesn’t contain any validations** |  |
| R 3.4.9 | Verifying that instructor page should also have a field to view the grade for a corresponding assignment. | * The instructor able to view grade for a corresponding assignment   **Expected output**  **Instructor able to see grade for a corresponding assignment** |  |
| R 3.4.10 | Verifying that instructor page should have field to display the submission time that the student last submitted the assignment. | * The instructor able see submission time of the assignment   **Expected output**  **Instructor able to see submission time** |  |
| R 3.4.11 | Verifying that instructor should be able to select a specific student’s assignment that needs to be graded. | * The instructor able to select specific student’s assignment that needs to be graded   **Expected output**  **Instructor have a chance to grade a specific student assignment** |  |
| R 3.4.12 | Verifying that instructor page should have a grade button which when clicked the assignment should be graded automatically. | * The instructor clicks on grade button for particular assignment   **Expected output**  **Message: Assignment is graded successfully** |  |
| R 3.4.13 | Verifying that field that indicates the latest assignment submission time that’s displayed in the instructor’s view should be greyed out which means it is non-editable. | * The user tries to edit the latest submission   **Expected output**  **Error Message: The submission time is none editable.** |  |
| R 3.5.1 | Verifying that Instructor should be able to select a particular assignment to start the grading of that particular assignment | * The instructor should be able to select a particular assignment to start the grading of that particular assignment   **Expected output**  **The instructor should able to select particular assignment for grading** |  |
| R 3.5.2 | Verifying that instructor should click on the grade button to grade the assignment automatically. | * The instructor should click on the grade button to grade the assignment automatically.   **Expected output**  **The instructor able to click on the grade button to grade the assignment automatically.** |  |
| R 3.5.3 | Verifying that on clicking the grade button, the zipped file should be automatically unzipped and extracted to separate folder and graded automatically | * The instructor clicking the grade button, the zipped file should be automatically unzipped and extracted to separate folder and graded automatically.   **Expected output**  **Message: Successfully graded assignment** |  |
| R 3.5.4 | Verifying that grade will be picked from the  instructor view and automatically reflected in the student view. | * The grade will be picked from the instructor view and automatically reflected in the student view.   **Expected output**  **The grade will be picked from the instructor view and automatically reflected in the student view.** |  |
| R 3.5.5 | Verifying that feedback that is recorded in the excel sheet will be attached in the student’s view for each and every student. | * The feedback that is recorded in the excel sheet will be attached in the student’s view for each and every student.   **Expected output**  **The feedback is recorded in excel sheet and it is available in the student page.** |  |