**Test Plan**

**Automated Grading and Feedback Tool for Java**

**Team C – Street Coders**

Table of Contents

[1.0 INTRODUCTION 3](#_Toc488151972)

[1.1 Objectives 3](#_Toc488151973)

[1.2 Team Members 3](#_Toc488151974)

[2.SCOPE 4](#_Toc488151975)

[2.1 Strategy 4](#_Toc488151976)

[2.2 Test Scenarios 5](#_Toc488151977)

[3. TEST APPROACH 9](#_Toc488151978)

[3.1 Test Automation 9](#_Toc488151979)

[4. RESPONSIBILITIES 9](#_Toc488151980)

**Team C- Street Coders**

**Test Plan**

# 1.0 INTRODUCTION

The Test Plan has been created to communicate the test approach to team members while developing the Automated Grading and Feedback Tool for Java. It includes the objectives, scope, schedule, risks, and approach during the development. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

## Objectives

We will be using Bugzilla for logging bugs in code which are found during the testing process. Every team member will be maintaining their own version of test suite whose details will be uploaded to Bugzilla once a day to track defects and the test results.

Testing will be done on a daily basis whenever the code is updated. The core feature testing will carry the highest priority for testing. Since every team member will be coding in this project the developer will choose one of the remaining five team members to test their code and log it in the bug tracker

## 1.2 Team Members

|  |  |
| --- | --- |
| **Resource Name** | **Role** |
| Siva Reddy Mekapothula | Primary contact |
| Harish Babu Achanta | Client management |
| Vamshi Krishna Girikala | Requirements management |
| Madanamohan Reddy Govindu | Data management |
| Venkatesh Katragadda | Issues management |
| Sunil Kumar Sangaraju | Quality and testing management |
| Prasanthi Rani Bhogaraju | Communications and documentation management |

# 

# 2.SCOPE

## 2.1 Strategy

The initial phase will include all ‘must have’ requirements. These and any other requirements that get included must all be tested. A tester must be able to:

1. Create a manual test with as many steps as necessary
2. Save it
3. Retrieve it and can view it when running the test
4. Enter results and appropriate comments
5. View results

The above-mentioned tester duties will be managed with the following control procedures

**Control Procedures:**

The team members work on different tasks and integrate the resulting code during milestones mostly working individually and collaborating at milestones.

**Reviews:**

We individually review others work and comment them for better enhancement.

**Review meetings:**

Review meetings every week to discuss, review and fix any errors occurred

**Change Request:**

We need to check with Client Requirements regularly whether we are developing the app accordingly or not, if not change them as per the client request.

**Defect Reporting:**

While using the app, if any defect occurs, Tester reports it in the bug tracking tool and assigns it to the concerned developer to fix it.

## 2.2 Test Scenarios

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement | Test Case |
| R 3.1.1 | The application will have a common login feature for both the instructor and student. | Verifying that application will have a common login feature for both the instructor and student. |
| R 3.1.2 | The student or the instructor should have separate views. | Verifying that student and the instructor should have separate views. |
| R 3.1.3 | The users should be redirected to their respective pages based on the login credentials. | Verifying that users should be redirected to their respective pages based on the login credentials. |
| R 3.1.4 | The user login should be having both user id and password as the mandatory fields. | Verifying that user login should be have both user id and password as the mandatory fields. |
| R 3.1.5 | If the user tries to login without entering the user-id and password, then an error message “the user id and password fields can’t be left blank” should be displayed. | Verifying that user tries to login without entering the user-id and password |
| R 3.1.6 | If the user enters the invalid password then an error message “the password entered is incorrect” should be displayed. | Verifying that user enters the invalid password |
| R 3.1.7 | If the user enters wrong user-id, then an error message “Incorrect user-id entered” should be displayed. | Verifying that user enters wrong user-id |
| R 3.1.8 | If the user enters invalid user id and password an error message “user-id and password mismatch” should be displayed. | Verifying that user enters invalid user id and password |
| R 3.1.9 | If the login is successful, the user will be directed to their respective home page based on their valid login credentials. | Verifying that login is successful |
| R 3.2.1 | The student should be redirected to the student page based on the login credentials. | Verifying that student should be redirected to the student page based on the login credentials. |
| R 3.2.2 | The student should have an option to upload the Java assignment. | Verifying that student should have an option to upload the Java assignment. |
| R 3.2.3 | The java assignment should be in the form of zipped folder. | Verifying that java assignment should be in the form of zipped folder. |
| R 3.2.4 | If the file is uploaded successfully an alert message “successfully uploaded” should be displayed | Verifying that the file is uploaded successfully |
| R 3.2.5 | If the upload fails an error message “Upload Failed” should be displayed. | Verifying that user fails to upload zipped folder |
| R 3.3.1 | The Student should have an option to view the grades for a particular assignment. | Verifying that student should have an option to view the grades for a assignment |
| R 3.3.2 | The field where the grade is viewed should be non-editable. | Verifying that field where the grade is viewed should be non-editable. |
| R 3.3.3 | The Student should also have the feature to view the feedback once the assignment is graded by the instructor. | Verifying that Student should also have the feature to view the feedback once the assignment is graded by the instructor. |
| R 3.3.4 | The Student Feedback should be in the format of .CSV which consists of detailed view of the errors along with the marks deducted. | Verifying that Student Feedback consists of detailed view of the errors along with the marks deducted. |
| R 3.3.5 | The feedback file should be downloadable and non-editable and should only be viewed. | Verifying that feedback file should be downloadable and non-editable and should only be viewed. |
| R 3.4.1 | The instructor should be redirected to the Instructor View page based on the Instructors login credentials. | Verifying that instructor should be redirected to the Instructor View page based on the Instructors login credentials. |
| R 3.4.2 | The instructor should be able to view all the students’ java assignments in zipped format. | Verifying that instructor should be able to view all the students’ java assignments in zipped format. |
| R 3.4.3 | The instructor should have an option to upload various test cases that needs to be executed against the students written java code. | Verifying that instructor should have an option to upload various test cases that needs to be executed against the students written java code. |
| R 3.4.4 | Once the input test cases document is uploaded the respective file should be displayed on the page which is an indication that file is uploaded successfully. | Verifying that input test cases document is uploaded the respective file should be displayed on the page. |
| R 3.4.5 | The instructor also should be able to include the expected output for each test case in the test cases document in a specific format. | 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. |
| R 3.4.6 | The instructor should have an option to upload the validation file that needs to be compared with the source code while grading the assignment. | 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. |
| R 3.4.7 | The 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. | 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. |
| R 3.4.8 | The Validation file should also consist of the rubric which consists of the mandatory functions and their corresponding marks. | Verifying that validation file should also consist of the rubric which consists of the mandatory functions and their corresponding marks. |
| R 3.4.9 | The instructor page should also have a field to view the grade for a corresponding assignment. | Verifying that instructor page should also have a field to view the grade for a corresponding assignment. |
| R 3.4.10 | The instructor page should have field to display the submission time that the student last submitted the assignment. | Verifying that instructor page should have field to display the submission time that the student last submitted the assignment. |
| R 3.4.11 | The instructor should be able to select a specific student’s assignment that needs to be graded. | Verifying that instructor should be able to select a specific student’s assignment that needs to be graded. |
| R 3.4.12 | The instructor page should have a grade button which when clicked the assignment should be graded automatically. | Verifying that instructor page should have a grade button which when clicked the assignment should be graded automatically. |
| R 3.4.13 | The 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. | 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. |
| R 3.5.1 | The Instructor should be able to select a particular assignment to start the grading of that particular assignment. | Verifying that Instructor should be able to select a particular assignment to start the grading of that particular assignment |
| R 3.5.2 | The Instructor should click on the grade button to grade the assignment automatically. | Verifying that instructor should click on the grade button to grade the assignment automatically. |
| R 3.5.3 | On clicking the grade button, the zipped file should be automatically unzipped and extracted to separate folder and graded automatically. | Verifying that on clicking the grade button, the zipped file should be automatically unzipped and extracted to separate folder and graded automatically |
| R 3.5.4 | The grade will be picked from the instructor view and automatically reflected in the student view. | Verifying that grade will be picked from the  instructor view and automatically reflected in the student view. |
| R 3.5.5 | The feedback that is recorded in the excel sheet will be attached in the student’s view for each and every student. | Verifying that feedback that is recorded in the excel sheet will be attached in the student’s view for each and every student. |

# **3.** TEST APPROACH

The project is using an agile approach, with weekly iterations. At the end of each week, the requirements identified for that iteration will be delivered to the team and will be tested.

Exploratory testing will play a large part of the testing as the team has never used this type of tool and will be learning as they go.

## 3.1 Test Automation

No scenarios for testing automation have been identified

# 4. RESPONSIBILITIES

|  |  |
| --- | --- |
| Test Plan | Sunil Kumar Sangaraju |
| Test Case | Sunil Kumar Sangaraju |
| Manual Testing | Siva Reddy Mekapothula, Harish Babu Achanta |

**Approvals**

|  |  |  |
| --- | --- | --- |
|  | **Project Manager** | **Quality & Testing Manager** |
| **Name** | Harish Babu Achanta | Sunil Kumar Sangaraju |
| **Signature** |  |  |