Software Requirements Specification

Final Version

July 18, 2017

Automated Grading and Feedback Tool for Java

Harish Babu Achanta
Siva Reddy Mekapothula
Prasanthi Bhogaraju
Sunil Kumar Sangaraju
Venkatesh Katragadda
Vamshi Krishna Girikala
Madanamohan Reddy Govindu

Submitted in partial fulfillment
Of the requirements of
CSIS44-691 Graduate Directed Project 1

Table of Contents

1.0. INTRODUCTION

1.1. PURPOSE

The purpose of this document is to present a detailed description of the software application named Automated Grading and Feedback Tool for Java developed to run on any Operating System. It will explain the purpose and features of the Tool, the interfaces of the Tool, what the tool will do, the constraints under which it must operate and how the tool will react to users interaction. This document is intended for both the Client and Developers of the tool and will be proposed to the client for approval.

1.2 PROBLEM STATEMENT

The problem with current manual correction of java programming assignments is students are not getting their grades and the useful feedback in a timely manner. The instructors or teaching assistants need to put in a lot of effort and time in grading and testing them against various test cases. The current online system only involves online quizzes which are corrected automatically based on pre-defined test answers. When it comes to programming assignments, the files need to be uploaded by students and then the instructors or teaching assistants are required to download them and then to unzip the files and grade them manually which leads to a lot of time and effort consuming.

1.3. PROJECT SCOPE

This will be an web application for the users (Students or Instructors) within the university who would like test their java code and get a proper grade and Feedback (Grading procedures will be explained in detail in later part of this document). This application will be designed to maximize efficiency and save time by aiding instructors to automatically grade the Java coding assignments that eliminates the traditional process of grading the assignments. By understanding the users need this application will be productive and useful to the users.

The audience will be the prospective students or instructors. Students can use the application to submit their Java assignments in zipped format in the application and also view their grades and feedback in the same view once the instructor has released the grade changes. The instructor can automatically grade the assignment within a single click as the tool gives the flexibility to automatically extract the zip file compile the code for errors and grade the assignment.

1.4. GLOSSARY

Term	Definition
Source Code	General text having commands to be compiled into an executable computer program
Feedback	Responsive information as a result of a task performed which is used for future improvements.
Scanning	Reading the program for syntactical errors is scanning.
Compilation	Computer program that transforms source code into another computer program.
Grading	Giving marks based on the students' performance.
Code Evaluation	The code is compiled and checked for errors and proper program structure.
Rubric	A simple document having information/rules on how a particular code should be evaluated.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The automated grading and feedback tool for Java is new open source project that is intended to reduce the manual and traditional process of evaluating and grading java assignments for students in various schools. This product will automatically download the java source code from the repository and scan the code for errors and automatically grade the assignment based on the rubric that is available in the application for that particular assignment.

2.2 PRODUCT FEATURES

The following is the list of features that gives the outline and description of each and every feature associated with the product. The features list is divided into core and additional features.
The core features are those that are mandatory for the system to have and the additional features are those functionalities that will be implemented if time permits.

2.2.1 CORE FEATURES

1. User Login and Welcome page

- The web application will have a welcome page that gives a basic idea of what the application is about **for new users.**
- The welcome page will also have a login button for the existing users to access the application.
- The user can be a student or Instructor.
- The login page will have common login procedure for both Student and Instructor.

- The student and Instructors will have separate views which means they will have their own login credentials.

2. Student View
   - The student view page consists of an upload option for the students to upload their Java Assignments in zipped formats.
   - The students should be able to view their grades for each assignment
   - The student can also view the feedback for the corresponding assignment.
3. Instructor view
   - The instructor can view the list of students along with their corresponding assignments.
   - The instructor can select the specific assignment to grade.
   - The instructor can click on grade button to automatically grade the assignments.
   - The instructor can view the assignment feedback file in .CSV(comma separated value) format after the grading is done.

4. Test cases view
   - The instructor can upload the test cases document.
   - The test cases document consists of all the required test cases that should be executed against the code.
   - The test cases document also consists of the expected output for each test case in a specific format.

5. Validation file
   - The validation file is uploaded by the instructor that is only visible to the instructor.
   - The validation file consists of all the method names along with their arguments, variable names, number of classes and class names.
   - The validation file also consists of the rubric depending on which the grading is done.

2.2.2 ADDITIONAL FEATURES

1. Performance Analysis

   The instructor should be able to evaluate the student by student performance analysis and also section wise analysis based on the grades obtained by the student.
2. Visualization

   The instructor should be able to visually represent the individual performance for each assignment.

3. FUNCTIONAL REQUIREMENTS SPECIFICATION

The Main purpose of the project is build a software application that facilitates Automatic Grading of Java source code by scanning the source code for syntax errors and program structure.

- There are few sub parts in the entire project:
  - Student View
  - Instructor view
  - Code Evaluation in Instructor view
  - Grading and Feedback to the Student

The Following are the requirements of the project,

APPLICATION HOME PAGE

- The application should have a dedicated URL for the users to access the application through a browser.
- The users should be able to access the home page by opening the URL through any browser.
- The user can be a student or instructor, so they will have a common login.
- The home page view will have all the available features of the applications.

R3.1. LOGIN FUNCTIONALITY

- **R3.1.1.** The application will have a common login feature for both the instructor and student.
- **R3.1.2.** The student or the instructor should have separate views.
- **R3.1.3.** The users should be redirected to their respective pages based on the login credentials.
- **R3.1.4.** The user login should be have both user id and password as the mandatory fields.
- **R3.1.5** 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.
- **R3.1.6** If the user enters the invalid password then an error message "the password entered is incorrect" should be displayed.
- **R3.1.7.** If the user enters wrong user-id, then an error message "Incorrect user-id entered" should be displayed.
- **R3.1.8.** If the user enters invalid user id and password an error message "user-id and password mismatch" should be displayed.
- **R3.1.9**. If the login is successful, the user will be directed to their respective home page based on their valid login credentials.

R3.2. A STUDENT VIEW PAGE

The student page should have a page to view all the assignment for the student to select the intended assignment to view. Also the student page should have a grade page and feedback page for the student to view grades and feedback for a particular assignment.

R3.2.1. The student should be redirected to the student page based on the login credentials.
R3.2.2. The 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.
R3.2.4 if the file is uploaded successfully an alert message "successfully uploaded" should be displayed

R3.2.5. If the upload fails an error message "Upload Failed" should be displayed.

R3.3. THE STUDENT PAGE FEATURES

R3.3.1. The Student should have an option to view the grades for a particular assignment.
R3.3.2. The field where the grade is viewed should be non-editable.
R3.3.3. The Student should also have the feature to view the feedback once the assignment is graded by the instructor.
R3.3.4. The Student Feedback should be in the format of .CSV which consists of detailed view of the errors along with the marks deducted.
R3.3.5. The feedback file should be downloadable and non-editable and should only be viewed.

R3.4. INSTRUCTOR VIEW PAGE

The instructor view page will have a button "Upload Document" to upload the test cases and validation documents.

R3.4.1 The instructor should be redirected to the Instructor View page based on the Instructors login credentials.

R3.4.2. The instructor should be able to view all the students' java assignments in zipped format.

R3.4.3. The instructor should have an option to upload various test cases that needs to be executed against the students written java code.
R3.4.4. 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.
R3.4.5. The instructor also should be able to include the expected output for each test case in the test cases document in a specific format.
R3.4.6. The instructor should have an option to upload the validation file that needs to be compared with the source code while grading the assignment.
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.
R3.4.8. The Validation file should also consists of the rubric which consists of the mandatory functions and their corresponding marks.
R3.4.9. The instructor page should also have a field to view the grade for a corresponding assignment.
R3.4.10. The instructor page should have field to display the submission time that the student last submitted the assignment.
R3.4.11. The instructor should be able to select a specific student's assignment that needs to be graded.
R3.4.12. The instructor page should have a grade button which when clicked the assignment should be graded automatically.
3.4.13. The field that indicates the latest assignment submission time that's displayed in the instructors view should be greyed out which means it is non-editable.

R3.5.AUTOMATIC GRADING PROCESS

R3.5.1. The Instructor should be able to select a particular assignment to start the grading of that particular assignment.

R3.5.2. The Instructor should click on the grade button to grade the assignment automatically.

R3.5.3. On clicking the grade button, the zipped file should be automatically unzipped and extracted to separate folder and graded automatically.

R3.5.4. The grade will be picked from the instructor view and automatically reflected in the student view.

R3.5.5. The feedback that is recorded in the excel sheet will be attached in the student's view for each and every student.

R3.6. SUCCESS FULL CODE COMPILATION PROCESS

Step 1. The java class files present in the extracted folder should be compiled and executed.

Step2. The student's java program should be run against the test cases present in the test cases document that is uploaded by the instructor.

Step 3. The output of the source code should be captured in a separate file.

Step 4. The output of student code for each test case should be compared with that of the expected output present in the test cases document.

Step5. The marks are awarded/deducted based on the rubric present in the Validation document.

R3.7. CODE COMPILATION FAILURE AND SCANNING

Step1. If the code fails to compile, no output is generated and 50% of marks are deducted (the percentage of marks deducted can be customizable).

Step2. The program is not run against any more test cases.

Step3. Whether the program is **successfully compiled or not**, the program is scanned for the class names, method names and its arguments, Comments and program structure and compared with that of the heuristics written in the validation document.

Step4. If there are any mismatches found when compared with the rules in validation document, The corresponding marks are deducted as per the rubric present in the validation document.

Step 5. The data containing errors along with the deducted marks is recorded in to a .csv file which will be given as a feed back to the student

Step 6. The grade is automatically calculated by the total number of marks that are recorded in the feedback file.

Step 7. The Calculated grade is automatically reflected in grade field present on the instructor view.

R3.8 FEEDBACK DOCUMENT

- The feedback document consists of data regarding output mismatches.
- The feedback consists of the syntax errors, validation errors, output mismatch errors and corresponding marks deducted for all the errors.
- The feedback document in the form .csv format should also contain the information regarding number of test cases the student code has passed.
- This feedback file is displayed on the feedback field present on the instructor view.

R3.9. ADDITIONAL FUNCTIONAL REQUIREMENTS

- The student should get an immediate feedback on submission of assignment.
- The source code should be executed against the preliminary test cases and assignment should be graded and feedback should be provided to the student immediately.
- The java source should be evaluated for program structure like presence of comments for each method, indentation, and space between different methods.

4. DETAILED NON-FUNCTIONAL REQUIREMENTS

The Non-Functional requirements like privacy, security, performance, Flexibility are the important features that are important for any Software Application.

4.1 SECURITY

Security is more important from both the student perspective as well as Instructor perspective.
The test cases and expected output files that the instructor uses to evaluate the students code should not be seen by the student.

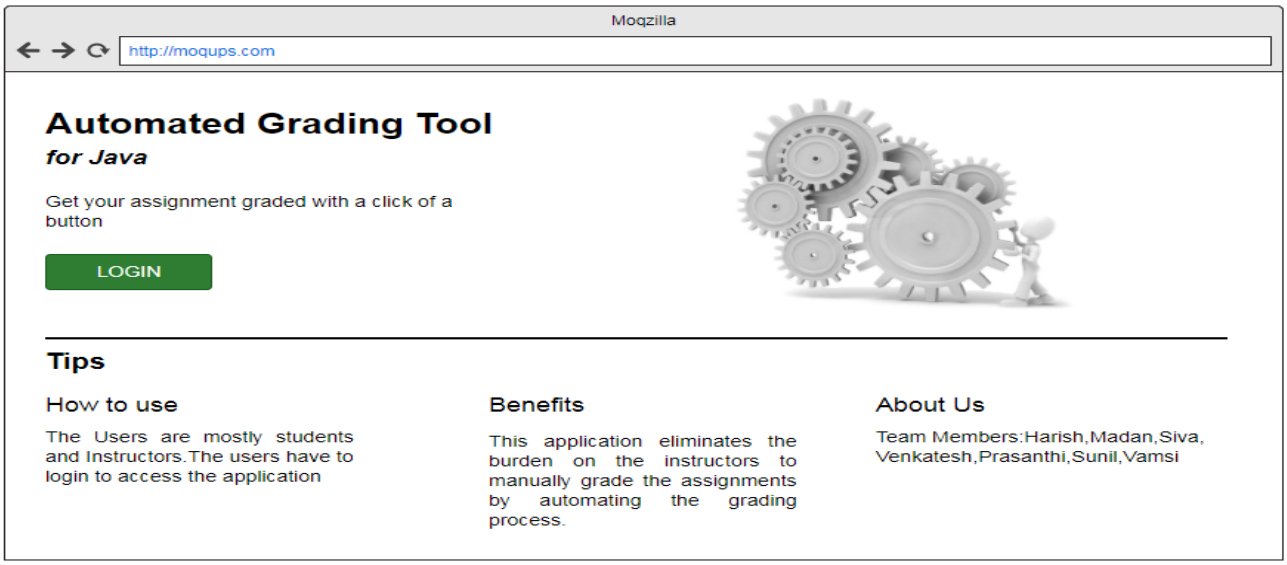
4.2. FLEXIBILITY

The System needs to be built in a modular fashion so that it is measurable and any additional features can be easily added and implemented.

5. EXTERNAL INTERFACE REQUIREMENTS

5.1 USER INTERFACE

5.1.1 Welcome Page



- This welcome page is for first time users who would like to browse the application
- This page provides a quick tutorial of what the application is about.
- The welcome page prompts the existing users to login to access the application
- The novice users can easily read and understand the basic idea of the application on browsing the home page.

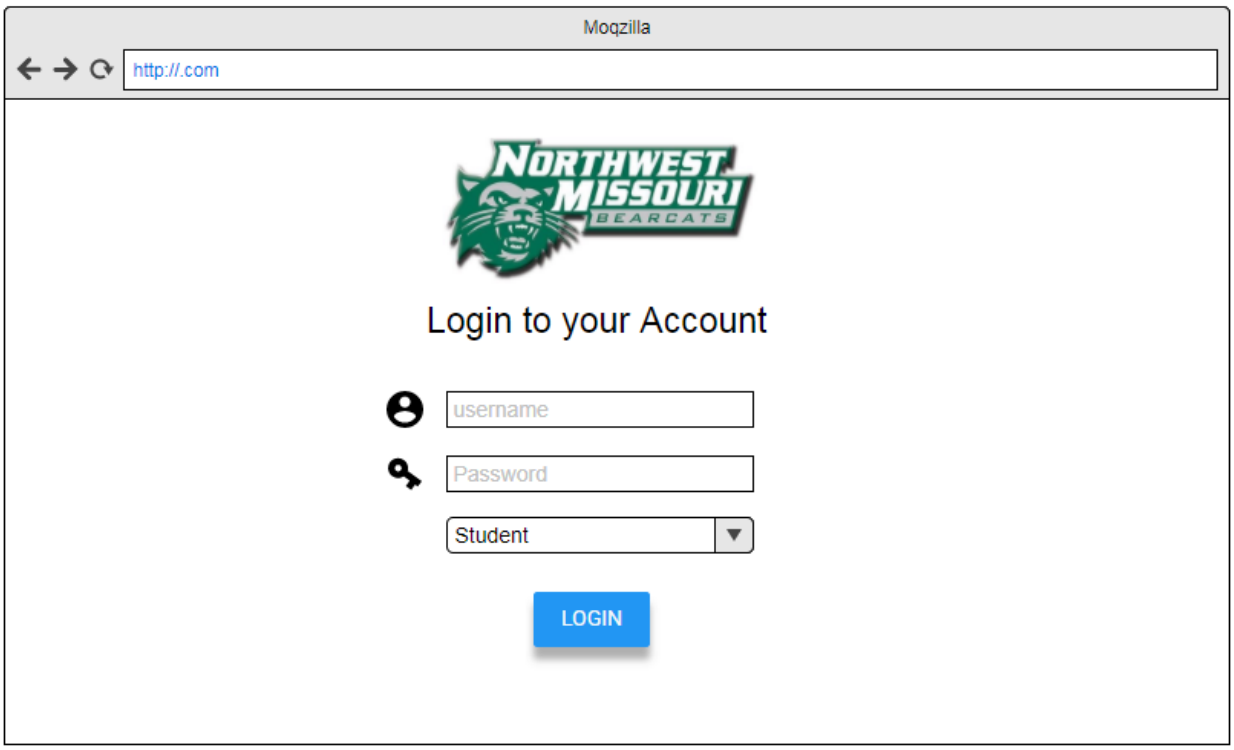
5.1.2 Login Page

- The Login page is used to keep the application secure from the intruders and only the registered users can access the application.
- The automated grading and feedback tool for java will have two types of users to login to the application.
- The users can only be either students or Instructor.
- Different views for student and instructor are displayed based on the user login credentials
- The instructor should enter valid user id and password and selects the role as instructor to view the application features of the instructor

Login Validation

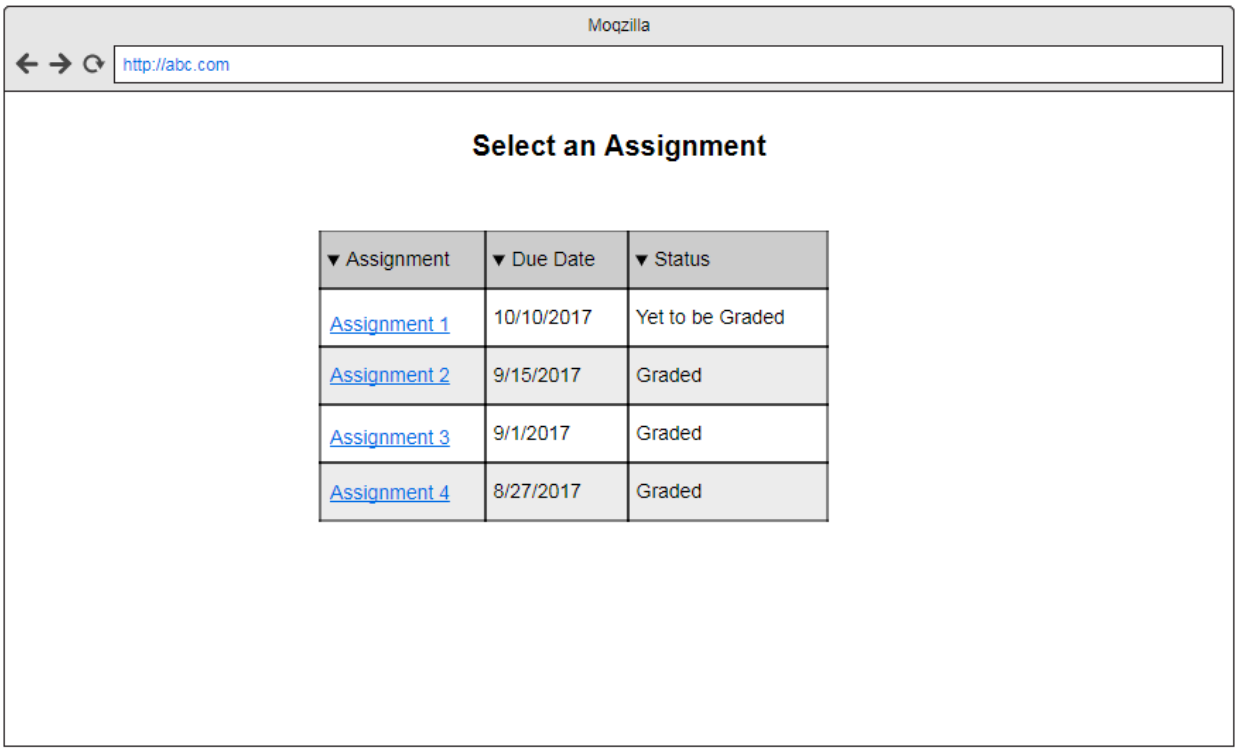
- The user will be validated against user id and password and the role selected by the user.

- The user id and corresponding password will be validated with that values that are stored in the database.
- The login credentials of the student user are different from that of the instructor.
- The student should enter the user id and password and select the role as student from the drop down.
- The default role for the login page will be student.

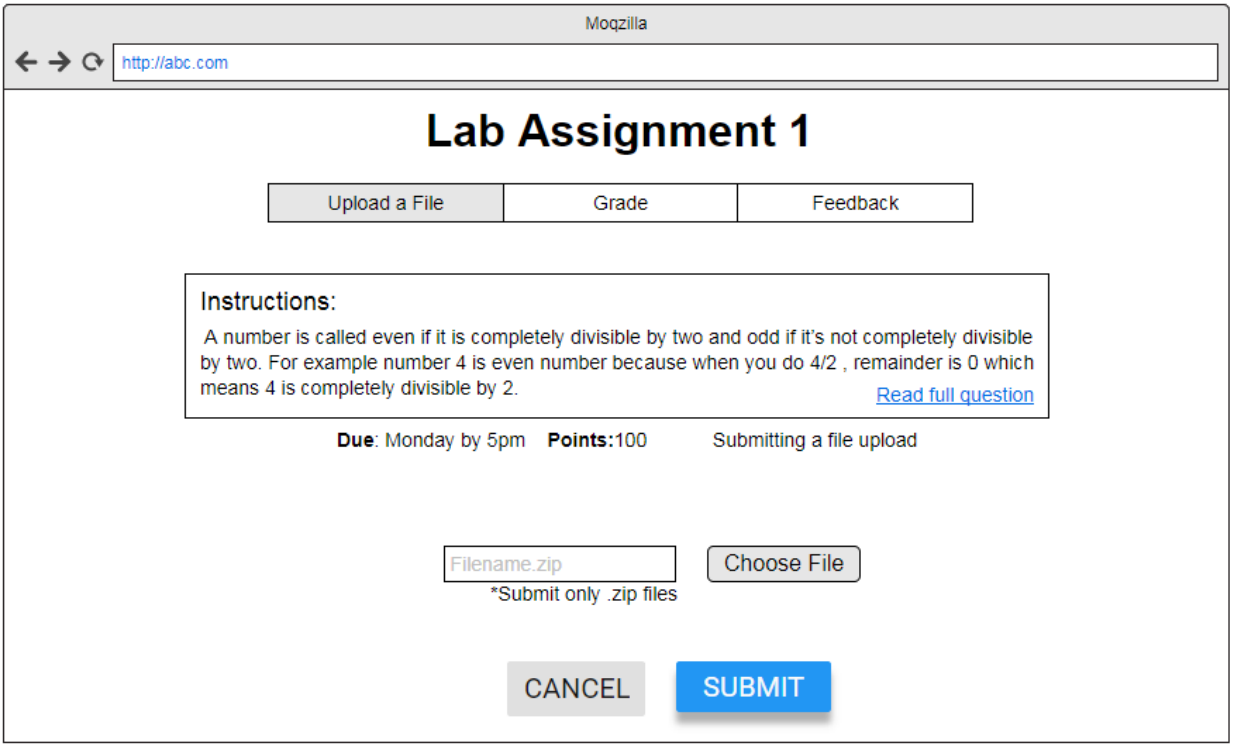


5.1.3 Student Assignments

- The assignments page contains of list of all the student assignments along with the due date and status for each assignment.
- The student can click on the assignment link to view the assignment details.
- The assignment due date is the date by which the assignment expires or the student is due to submit the assignment.
- The assignment status field gives the information about the assignment is graded or not so that the student can be cautious of what assignments are graded and what are not.

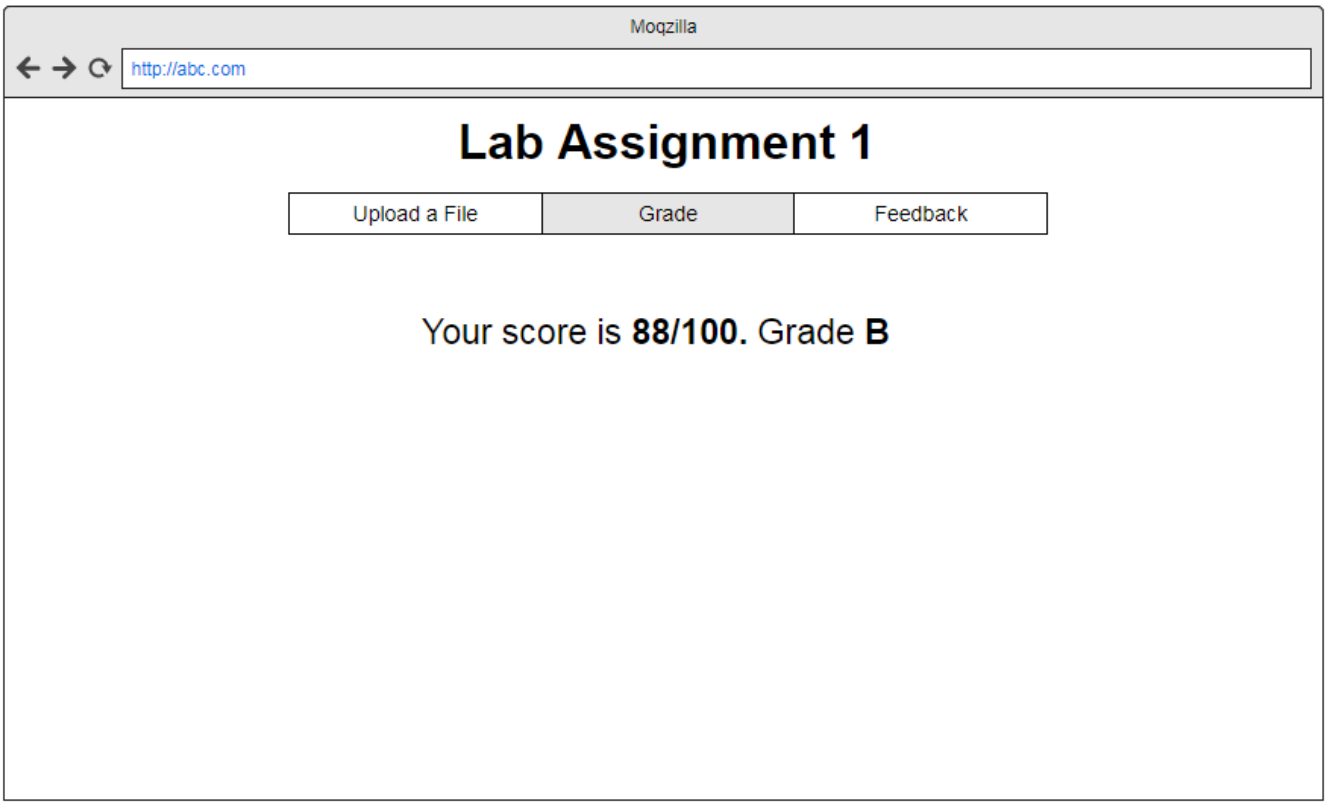


5.1.4 Upload an Assignment



- The user lands on this page by clicking on each individual assignment.
- The assignment page consists of details of the assignment like what is the assignment about.
- The Assignment question is in the form of a minimized format and user can click on the read full question to read the entire question.
- The student can see at what date and time the assignment is due to submission.
- The assignment also has an field that displays how many marks the question carries.
- There is an option on the individual assignment page for the student to upload the assignment.
- The assignment that is uploaded should be specifically of zipped format.
- The student clicks on the choose file to select the .zip file from the system path.
- The user clicks on the submit button to upload the assignment to the server.
- The user also can cancel the upload by clicking on the cancel button.

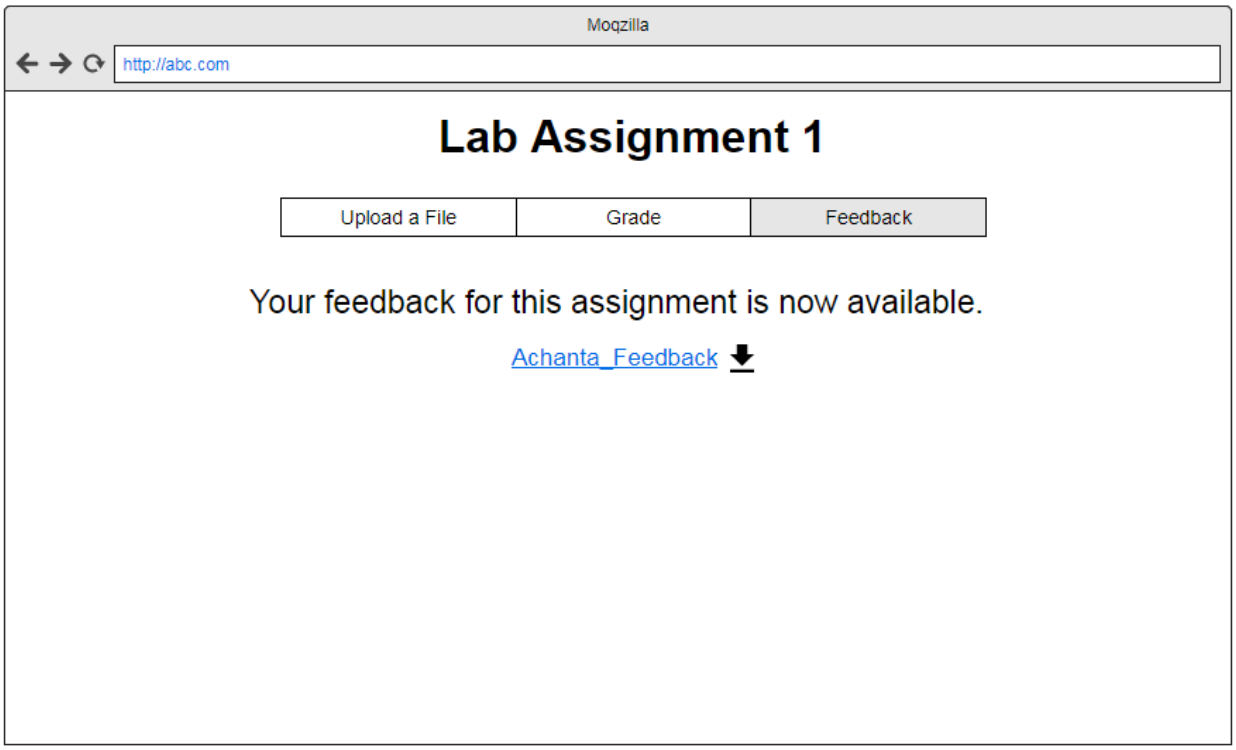
5.1.5. Grade Page



- The grade page is obtained by clicking on the grade button for a particular assignment.
- The grade is viewed only for that particular assignment.

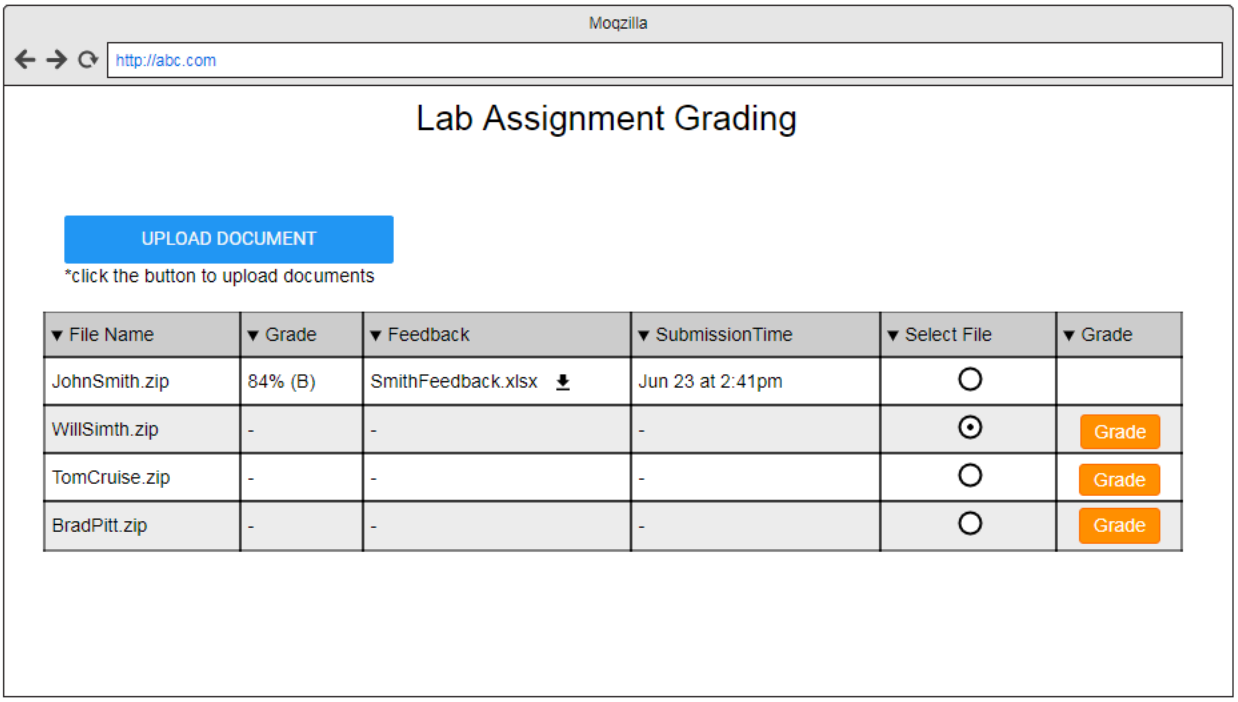
- The grade displayed is based on the score obtained by the student for that particular assignment.

5.1.6. Feedback



- The student clicks on the feedback tab to view the feedback for a particular assignment.
- The feedback will be of the .csv format that contains all the errors along with amount of marks deducted.
- The feedback is picked from the instructor view and displayed in the feedback tab of the student view.
- The feedback file is available as a downloadable link, so that the student can easily download the report and view it.
- The feedback file is protected or in read only mode so that student cannot manipulate the feedback and the file is secured.

5.1.7. Instructor View Page



- The instructor has to login to the application with his/her own login credentials and the role “instructor” should be selected for the application to know that the instructor is logging in.

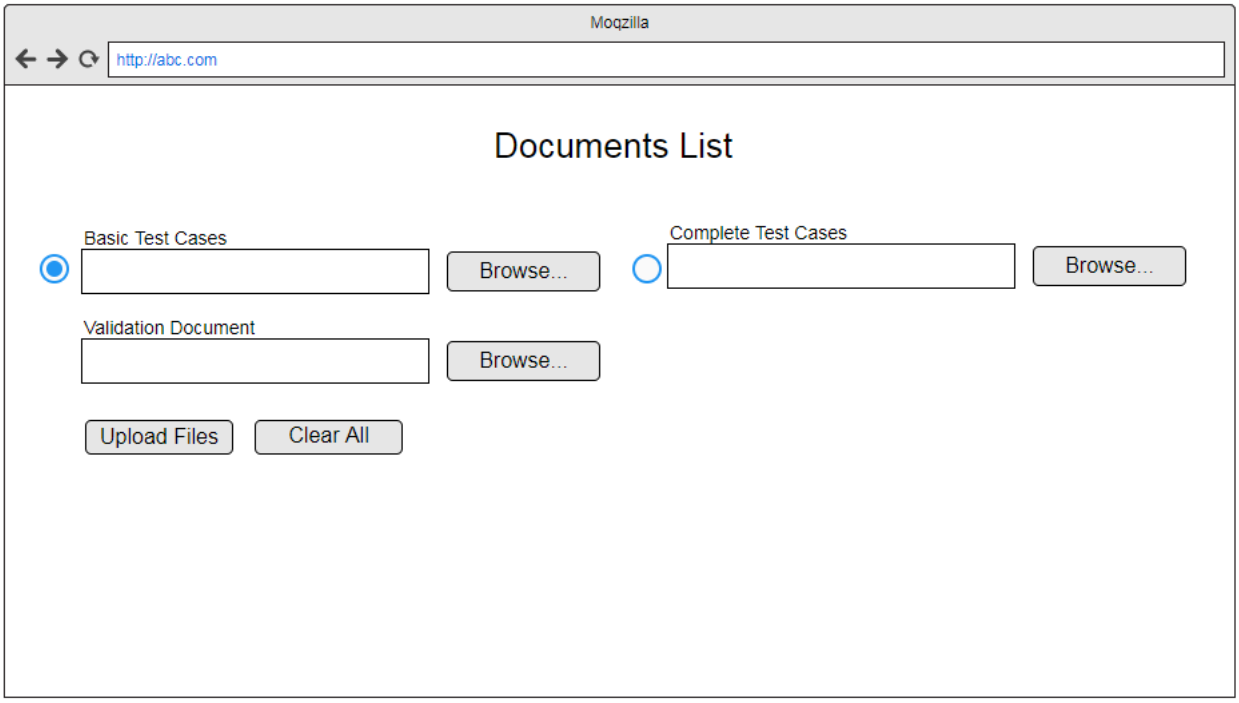
- The Login credentials should be validated with that of the login credentials present in the database.
- Once the Instructor successfully logs in , the instructor view page is displayed.
- The instructor page has an “Upload Document button” to upload the test cases and validation document.

The instructor view has options like,

- **Filename** that has to view all the student submitted assignments that are retrieved from the server and displayed on the screen.
- **Grade** field to view the score and associated grade for each and every corresponding assignment.
- **Feedback** field to view the feedback in a (.CSV format) of the graded assignment.
- The submission time option shows the latest time of when the student has submitted the assignment.
- The select file option is provided to the instructor to select a particular assignment to grade.

- The grade button is provided on the instructor page for the instructor to select an assignment and Click on the grade button to grade the assignment.
- The assignment will be automatically graded and the score will be reflected on the grade field on the instructor view page and feedback report will appear under the feedback field.

5.1.8. Documents Upload



- The instructor lands on the documents upload page when the instructor clicks on the Upload documents button on the instructor view page.
- The upload options for basic test cases is used when the instructor grades the assignment immediately once the student submits the assignment.
- The complete test cases document is uploaded by the instructor that has all the test cases the program should execute on.
- The Validation document is upload to check the class names and method names present in the program.
- Once the test cases and validation document are ready the instructor clicks on the upload files so that the files get uploaded successfully.

5.2 HARDWARE REQUIREMENTS

- Automated Grading tool for Java is expected to be a web based application that is in general built for windows platform and solely supported by all the available web browsers. The application is accessed with the help of an URL opened through the browser.
- The Xampp web server which is an open source cross platform web server is intended to be used as server for storing and processing of files.
- The Xampp server acts as an interface between student submitting the assignment and the assignment being reflected in the instructor's view. The Information will be sent using TCP/IP protocol and HTTP protocol.

5.3 SOFTWARE INTERFACES

- The user interface of Automated grading and feedback tool for java web application is developed using HTML, CSS, JavaScript. Bootstrap framework is used to make the application more responsive when accessed from various digital devices of various screen sizes.
- PHP is intended to be used as a server side scripting language and My SQL is the expected database to be used.

5.3.1 Incoming And Outgoing Items

- Outgoing data consist of the files that are uploaded by the student on to the server via from end screens.
- The test cases and the validation documents uploaded by the instructor on to the server.

- Incoming data consists of all the students submitted files that are retrieved from the server and displayed on UI screen.
- The Feedback document generated and stored on the server is also displayed on the instructors view

5.3.2 Services And Communications

- The automated grading tool web application relies mostly on the server push and pull protocols
- Server interaction will seldom occur in the following situations:
    - Whenever the student submits/Uploads the assignment in .zip format.
    - Whenever the server finishes extracting zipped files and saves the extracted folder on the server itself.
    - The application will interact with the server when the application should retrieve all the student submitted files.

5.4 COMMUNICATION INTERFACES

The Automated Grading tool for Java web application is intended to have a web-based network server developed using the PHP language. The server is present to retrieve grade information like total assignment marks scored and Assignment grade from the database that stores the transaction history and user interaction results. The PHP is compatible with the MySQL database and whenever a user opens the application the latest information is retrieved from both the server and the database and displayed on the UI screen.

6. KEY MILESTONES

Milestone	Deadline	Comments
SRS Document	7/17/2017	Software Requirements documented and Presented
Software Architecture	7/17/2017	Basic project architecture documented and reviewed
Project Management Plan	7/17/2017	Project plan is developed to track the work progress
Server & Database Setup	9/11/2017	This item will be setup during implementation.
Test Plan & Test Cases	7/17/2017	The actual results are recorded at beginning of test
Gantt chart	7/17/2017	Gantt chart describes the work division and resource responsible
Interface and Code Implementation	9/30/2017	This needs to be implemented 2 weeks a ahead of project completion to check for bugs
Project Completion	12/16/2017	The complete reliable project is submitted to the client