Software Requirements Specification

Version 1.0

June 28, 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** 3](#_Toc486430969)

[**1.1. Purpose** 3](#_Toc486430970)

[**1.2 Problem Statement** 3](#_Toc486430971)

[**1.3. Scope of Project** 3](#_Toc486430972)

[**2. Functional Requirements Specification** 3](#_Toc486430973)

[**2.1 Requirements** 4](#_Toc486430974)

[**R1. A student View Page** 4](#_Toc486430975)

[**R2.The Student Page Features** 4](#_Toc486430976)

[**R3. Instructor View Page** 4](#_Toc486430977)

[**R4. Code compilation failure** 5](#_Toc486430978)

[**R5. Successful Code Compilation** 5](#_Toc486430979)

[**R6. Scanning** 5](#_Toc486430980)

[**R7. Grading Process** 6](#_Toc486430981)

[**3. Detailed Non-Functional Requirements** 7](#_Toc486430982)

[**3.1 Security** 7](#_Toc486430983)

[**3.2. Flexibility** 7](#_Toc486430984)

# **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. Scope of Project**

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.

## 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 |  |
| Compilation |  |
| Grading |  |
| Code Evaluation |  |
| Rubric |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **2. 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

## **2.1 Requirements**

The Following are the requirements of the project,

### **R1. A student View Page**

**R1.1.** Student view page includes the following functionalities. That includes,

* + Student Login

**R1.2.** The student login will be validated against User Id and Password validation.

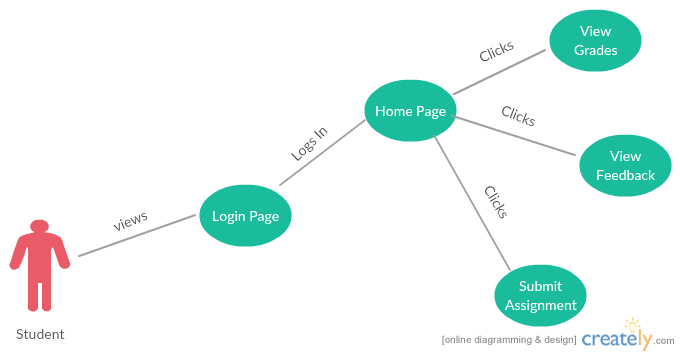
### **R2.The Student Page Features**

The student view page consists of File upload option to upload the java assignment file.

**R2.1.** The Student has an option to view the grades that are given by the instructor.

**R2.2.** The Student can also view the Feedback from the instructor.

**R2.3.** The Student Feedback consists of detailed view of the errors along with the marks deducted.



**Fig1: Flow chart of Student Interaction**

### **R3. Instructor View Page**

**R3.1.** The Instructor view will be having separate login credentials to login to the application.

**R3.2.** The instructor logs in to the application to view all the students’ assignments

**R3.3.** The instructor needs to upload various test cases that needs to be executed against the students written java code.

**R3.4**.The instructor also includes the expected output for each test case in the test cases document in a specific format.

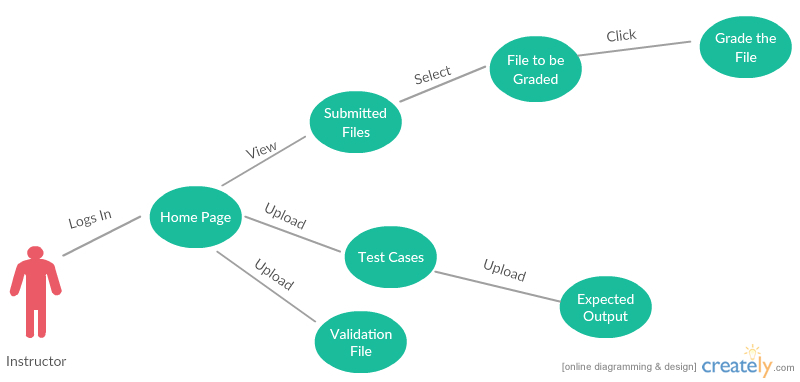
**R3.5.** The instructor should be able to select a student’s assignment that needs to be graded and click on Grade button.

**R3.6.** As soon as the instructor clicks the grade button,

**R3.7.** The zip file gets extracted to a folder automatically

**R3.8.**The input to the program is fetched from the file that has all the test cases uploaded.

**R3.9.** The program is compiled and executed for each of the test cases



**Fig2: Flow chart of Faculty Interaction**

### **R4. Code compilation failure**

**R4.1**. 50% of marks will be deducted from total if the program fails to compile.

**R4.2.** The java source code needs to be validated with the rules written in the validation files for proper function names, variable names and comments for each method.

**R4.3.** If there are mismatches or errors in the source code written by the student, the marks should be automatically deducted based on the error.

**R4.4.**The Rubric for deduction of the marks is also included in the Validation file.

**R4.5.** The data containing errors along with the deducted marks is recorded in to an excel sheet which will be given as a feed back to the student.

### **R5. Successful Code Compilation**

**R5.1.** The program compiles and executes successfully.

**R5.2.** The output is compared with the expected output.

**R5.3.** In case of any output mismatches the marks are deducted based on the rubric present in the validation file uploaded by the instructor.

**R5.4.** The program is again scanned for proper program structure.

**R5.5.** The program is scanned for presence of comments for each method and proper indentation.

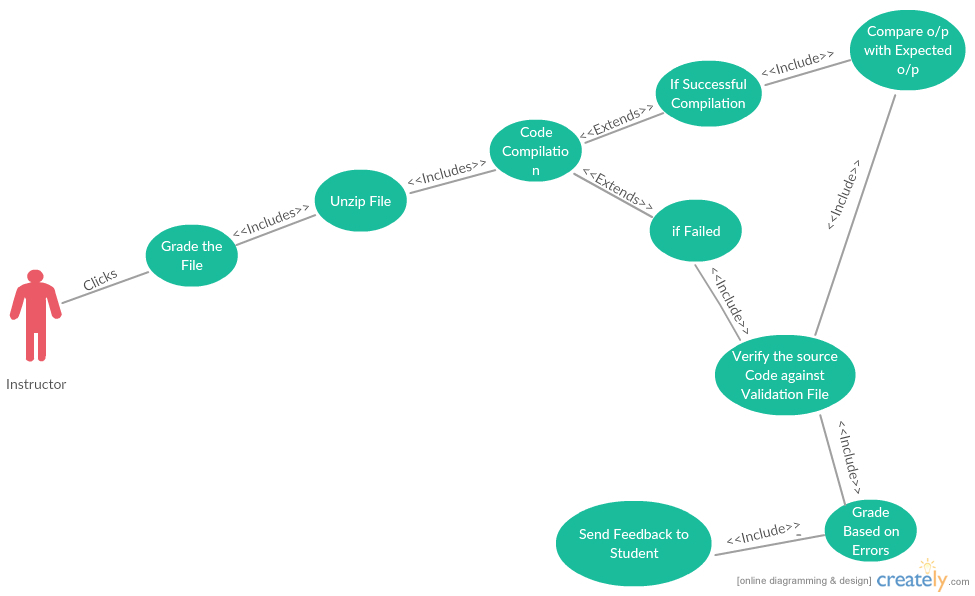
**R6. Scanning**

The syntax scanning of java source code is done in accordance with the rules written in the validation file.

**R6.1.** The validation file contains all the required method names, variable names, method arguments, constant names.

**R6.2.** If the students source code doesn’t comply with validation rules written by the instructor, the score is deducted automatically based on the rules specified in the validation file.

**R6.3.** The data containing errors along with the deducted marks is recorded in to an excel sheet which will be given as a feed back to the student.

****

**Fig3: Flow chart of grading process**

### **R7. Grading Process**

**R7.1.** The Grade is given to the student based on the rubric that is formulated in the validation file.

**R7.2.** The grade will be picked from the instructor view and automatically reflected in the student view.

**R7.3.** The feedback that is recorded in the excel sheet will be attached in the student’s view for each and every student.

# **3. Detailed Non-Functional Requirements**

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

## **3.1 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.

## **3.2. 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.