**INTRODUCTION**

**Purpose:**

The purpose of this software functional specification requirement document is to present the functional signatures that will be used in developing the code in the future semester for GDP II for the project “**A**utomatic **G**rading and **F**eedback **T**ool for **J**ava”. This document helps team members to not to miss the important functionality of code requirements and helps in understanding the operation of the code by providing comments. This document also eases the process of development of the code.

**Scope:**

The scope of the project “**A**utomatic **G**rading and **F**eedback **T**ool for **J**ava” is to develop and deliver an automatic java assignment grading tool with operations including automatic unzip the assignment file, retrieving the .java file, compiling and executing the file and then tested with different test cases uploaded by the instructor or teaching assistant. Finally, the student’s data on each assignment will be stored in a database which used for providing useful information for the instructor. This project enhances the instructor’s efficiency by assisting them by providing tools which will meet all grading needs. This system includes scripts to automate the process, database and user interface developed for easy user interaction.

**Pages of web application:**

* Welcome page
* Login page
* Assignments overview pages
* Upload document page
* Assignment Grades page
* Assignment Feedback view page
* Instructor grading page
* Upload test cases page

**Modules**

|  |  |  |
| --- | --- | --- |
| **Initial view** | **Student view** | **Instructor view** |
| Welcome page | Assignments overview page | Instructor grading page |
| Login page | Upload document page | Upload test cases page |
|  | Assignment grades page |  |
|  | Assignment Feedback view page |  |

Based on the functionality, the pages were divided into three views. Along with the user interface pages, these modules contain their respective controllers in order to communicate with their respective databases.

As mentioned in the above table initial view module manages the welcome and login page. It also consists of many controllers in order to communicate with the database for the storage and retrieval of data. The user database has the user details. The controllers in this module are used to check the username and password format, and also used to encrypt the password.

Student view module manages the assignments overview page, upload document page, assignment grades page and assignment feedback view page. This module consists of controllers that communicate with the upload document page and assignment feedback page. The assignment grades is a page where the student can check their respective grades and the assignment feedback view is the page where the students can view the feedback given by the instructor.

The instructor view module consists of the instructor grading page and upload test cases page. In the instructor grading page the instructor will upload the grades of the students in the page. In the upload test cases page, the instructor uploads the test cases so that the assignment can pass the test cases and can be graded accordingly.

// This LoginPage class contains methods which are used for validating and verifying the user identification details of user of the application.

**Class LoginPage**{

Public function validation\_UserId(String userId): boolean {

// During registration process, username is checked and compared to the names stored in database. If not matches then user is granted permission to use that name for further login purpose.

}

Public function validation\_PasswordCharacters(String password): boolean {

// Method to validate passwords for minimum length, special characters, numbers,Uppercase letters, and lowercase letters

}

Public function validation\_PasswordLengths (String password): boolean {

// Entered new password should follow the constraints implemented for that function like minimum length. For example, if the constraint is minimum of 9 characters then entered password length must be >= 9 characters.

}

Public function verify\_UserId(String userId): boolean {

// Method to verify the userid for user authentication

}

Public function verify\_Password(String password): boolean {

// During login process, the entered credentials like username and password should match the details present in the database. If not the access should be denied.

}

// This class is core part of the project. Here we will evaluate the assignment, grading of assignment and giving the feedback.

**Class Instructor\_View**{

Public function checking\_Role(String role):String{

// This method is to check the role of the user. During registration process, each person is given some permissions either user is a student or instructor. Based on stored details, application will automatically identify if the user is either instructor or not.

}

Public function fileExists(String fileName):String{

// This subroutine is used to identify whether a student submitted a file or not. It is done by comparing the student list of the students submitted and stored in the database. This process will be done by verification method by using looping concept.

}

Public function file\_Submission(String file): boolean{

// Method to process the file upload after retrieving the file from a path in the computer system. This method is called when the student wants to submit the document to the application.

}

Public function file\_UnZip(String file, String location):String{

// Method to unzip the file if the student submits a zipped file and separate the source code from it

}

Public function retrive\_JavaFile(String fileName, String location): String{

// Method to retrieve the zipped file from the location. This method retrieves the desired file having a fileName from a location which is basically a path to retrieve the file.

}

Public function testcases\_Upload(ArrayList<String> testcases): ArrayList<String>{

//Method to upload the test cases

}

Public function compliation(String java\_file): ArrayList<String>{

//Method to process the compilation of source code. The source code is checked for syntax errors and method signatures

}

Public function assignment\_Evaluating(ArrayList<String> assignment ): ArrayList<String>{

//Method to evaluate an assignment based on test case validation

}

Public function assignment\_Grading(ArrayList<String> evaluation): int{

//Method to update the grades to the record based on the evaluation

}

Public function assignment\_Feedback(ArrayList<String> report): ArrayList<String>{

//Method to generate the feedback based on the assignment evaluation and grades. The feedback is stored in the form of an excel sheet along with test case number, description and the score for each test case.

}

Public function result (String result): String {

//method for sending the results to the student after grading them. The results contain a score associated with a grade for that score interval.

}

}

//This class contains the code for the student operations like document uploading, downloading feedback, role checking either student or instructor or teaching assistant, and to view the grades.

**Class StudentView** {

Public function roleChecking(String role):boolean{

//This method is to check the role of the user. During registration process, each person is given some permissions either user is a student or instructor. Based on stored details, application will automatically identify if the user is either student or not.

}

Public function document\_Uploading (String document\_Name, String location): String{

//Method to choose the file from the user system and upload it to the application. Parameter document\_Name specifies the name of the document and parameter location specifies the location path of the file to be uploaded

}

Public function document\_Downloading(String documentName):String{

// This method includes the code for downloading the feedback sheet to the student’s personal system.

}

Public function viewingGrade(String assignmen\_tName): int{

// This method contains the code to process and make the grades available to the correspond student.

}

Public function view\_Feedback(String comment): ArrayList<String>{

// This method contains the code process the feedback and make downloadable text document and make it available for the student to review it.

}

}