**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.

**Login Page Code**

<form>

<td><input type=”text”, name=”userid”></td>

<td><input type=”Password”, name=”password”></td>

<td><input type=”submit”, name=”login”></td>

</form>

**// validating the userid and Password**

<script>

function validateUserId (userid) {

var letters= /^[0-9A-Za-z]$/;

if(userid.value.match(letters)){

// verifying the user id contains characters and numbers

}

else{

// an error message will be displayed

}

}

function passwordValidation(password){

var maxNumberofLetters=15;

var minNumberofLetters=8;

var letters=/^[0-9A-Za-z@!#%$&^\*&]{8,15}$/;

if(password.length>maxNumberofLetters || password.length<minNumberofLetters){

// password length is checking whether it satisfying the minimum and maximum length conditions

}

If(password.value.match(letters)){

// We are checking password contains special characters, numbers, capital letters, small letters

}

else{

// An error message will be displayed

}

}

</script>

**//Verifying the userId and password**

<?php

function verifyingUserIdPassword(){

if($\_POST[‘login’]){

//After click on login button, application will execute the queries and verify the both username and password.

}

else{

//Both username and password are incorrect.

}

}

**// Student submitting the assignment**

function uploadDocument(){

$upload\_dir=’/ C:/uplaod’;

//Here we are checking student uploads the .zip file or not.

}

**// Down loading the feedback document**

function downloadingDocument($name){

if(file\_exists($name)){

// we will download the file and stored in required location

}

else{

// file doesn’t exist. Please verify it.

}

}

?>