**Functional Requirements and Assessment Criteria**

Education Software Test System

* **Requirement 1. (must)**
  + **Requirement:** The system must allow the customer the ability to create short assessments in the form of tests.
  + **Assessment Criteria:**
    - Demonstration of a user being able to create a short assessment using the software, without prior training.
    - Once the creation of the test is finalised, it should be pushed to the system and made available to students.
* **Requirement 2. (must)**
  + **Requirement:** Customers should be able to choose whether they want the assessment to be formative or summative.
  + **Assessment Criteria:**
    - When creating a test, before publishing, but after assigning all questions, the acting teacher should be able to select whether the test is formative or summative with a simple selection tool.
    - That test should appear and behave as the assigned type to all students 100% of the time.
    - The system should only allow the lecturer to create two different types of tests, formative tests and summative tests.
* **Requirement 3. (must)**
  + **Requirement:** The software should record the number of times a student answered a question incorrectly and allow the user acting as a teacher to view this data.
  + **Assessment Criteria:**
    - A teacher can view a completed test, and for each question will see the number of times that question was answered incorrectly.
* **Requirement 4. (should)**
  + **Requirement:** The user acting as a student should get feedback after each attempt at a test, but only provided with answers following the final attempt.
  + **Assessment Criteria:**
    - When taking a summative test, after each completion of a test, the student should be given information detailing how many answers were correct, and how many incorrect.
    - After completing the final attempt of a test, the student must be able to view the completed test, and shown which answers were correct, and which incorrect, with the correct answers being given.
* **Requirement 5. (must)**
  + **Requirement:** The system should allow the acting teachers to obtain statistics about the tests such as class score, questions most answered incorrectly.
  + **Assessment Criteria:**
    - 100% of users acting as teachers should be able to use the system to access a statistics page based of a single test.
    - That page will contain and display statistics gathered from students who completed that test.
* **Requirement 6. (must)**
  + **Requirement:** Teachers can view and request result reports from any student’s completed test
  + **Assessment Criteria:**
    - Demonstration of a teacher account pulling up the results from 3 different students.
    - Demonstration of a teacher looking at a particular student and then looking at results from their last 2 tests
* **Requirement 7 (must)**
  + **Requirement:** The lecturer should be able to view results for any one test taken by a particular student
    - Demonstration of the teacher account pulling up the results from the recent test of a particular student.
    - Demonstration of the teacher pulling up the first test taken by a different particular student.
* **Requirement 8 (should)**
  + **Requirement:** There should be a teacher login and a student login to allow for students to answer, teachers to create tests etc….
    - Demonstration of the teacher account logging into a window and then trying to creating a test.
    - Demonstration of a student account logging into the window and trying to make a test.
* **Requirement 9 (should)**
  + **Requirement:** Each student much have a unique ID number and login to be used as a unique identifier (key) within the database.
    - 2 demonstrations of logging into 2 different student account with different unique ID’s
* **Requirement 10 (should)**
  + **Requirement:** The system should be stored on a server, to make sure that students cannot edit any source code, making the tests easier or finding answers.
    - Demonstration of opening the source code of the python file and showing the processing and validation is happening on an external server.
* **Requirement 11. (must)**
  + **Requirement:** There should be a teacher login and a student login (to allow for students to answer, teachers to create tests)
  + **Assessment Criteria:**
    - Users should be able to input a username and a password
    - If an invalid username/password is input
    - Upon entering a correct username and password – the system should add the features that the teacher can use.
* **Requirement 12. (must)**
  + **Requirement:** Each user must have a unique ID number and login to be used as a unique identifier (key) within the database.
  + **Assessment Criteria:**
    - Students should always be redirected to their “homepage”
    - Upon entering their details, users should only ever be able to access the account corresponding to their own details
* **Requirement 13. (must)**
  + **Requirement:** The system should be stored on a server, to make sure that students cannot edit any source code, making the tests easier or finding answers.
  + **Assessment Criteria:**
    - Students should not be able to access neither the answers or the source code
* **Requirement 14. (must)**
  + **Requirement:** When taking answers from students allow spelling mistakes or variations of the answer to be correct
  + **Assessment Criteria:**
    - The system should be flexible and allow for spelling mistakes
    - Upon entering an answer to a test, any minor error should be overlooked, and the answer taken to be correct
* **Requirement 15. (must)**
  + **Requirement:** Users should be taken to different home screens based on whether they are students or teachers
  + **Assessment Criteria:**
    - Upon entering their correct details, the user should be taken to their home screen 99% of the time.
* **Requirement 16. (must)**
  + **Requirement:** Students should only be allowed access to their personal results, and the tests they are able to take
  + **Assessment Criteria:**
    - Each student should have a page unique to them based on their details.
    - Any two students should not be able to view each other’s results, or the tests they have taken or awaiting tests.
* **Requirement 17. (must)**
  + **Requirement:** The system should work reliably and without fail 99% of the time
  + **Assessment Criteria:**
* The system should be able to allow users to access their accounts – depending on their account; create a test; answer a test; view history; view data and only fail 99% of the time.
* **Requirement 28. (must)**

**Requirement:** Summative Tests – The mark for the test taken should be provided immediately after the test is finished.

* **Assessment Criteria:**
* After finishing the summative test, the software will provide students with their marks.
* **Requirement 29. (must)**
* **Requirement:** Summative Tests – lecturers should be able to view and retrieve results for all students after the test deadline.
* **Assessment Criteria:**
* After the summative tests deadline the lecturer will be able to view and retrieve students results.
* **Requirement 30. (should)**
* **Requirement:** Should run on windows 7/10 PC.
* **Assessment Criteria:**
* The programme must run without any error on windows 7/10 PC.
* **Requirement 31. (could)**
* **Requirement:** Questions should appear to students in random order.
* **Assessment Criteria:**
* When starting the test every student should have a different form with different question order.
* **Requirement 32. (must)**
* **Requirement:** Teachers should be able to edit created tests.
* **Assessment Criteria:**
* The system should allow the lecturer to edit tests any time after creating them.