**CM1202 – Developing Quality Software – Group 1**

**Task 1 - Functional Requirements and Assessment Criteria**

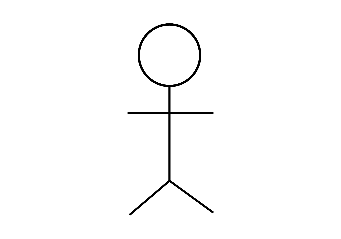
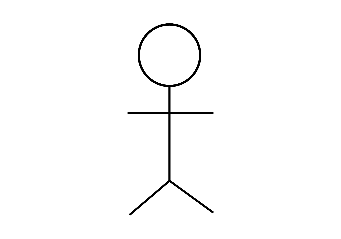
Education Software Test System

* **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. (should)**
  + **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:** 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 14. (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 15. (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 16. (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 automatically show the student their marks. This should work 100% of the time
* **Requirement 17. (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 student’s results.
* **Requirement 18. (must)**
* **Requirement:** Teachers should be able to edit created tests.
* **Assessment Criteria:**
* The system must show teachers all created tests.
* The teacher must be able to navigate to the test they want to edit without guidance and edit it.
* Upon editing teachers should be able to change all questions and all answers, as well as the title, the subject, and the students to publish the test to.
* **Requirement 19. (must)**
  + **Requirement:** Teachers using the system can monitor a single student’s progress and results
  + **Assessment Criteria:**
    - Teachers can select any student, and the system must show the user that student’s previous test results and statistics.
    - This must show data for the correct student 99% of the time.
* **Requirement 20. (must)**
  + **Requirement:** Formative tests should not record the individual student performance.
  + **Assessment Criteria:**
    - 100% of completed formative tests should not be available for post completion viewing by a teacher.
    - Only summative tests should be able to be individually viewed post completion by a teacher
* **Requirement 21. (must)**
  + **Requirement:** During a formative test, students should be provided feedback after submitting their answer for one question.
  + **Assessment Criteria:**
    - After entering an answer for a question, a user acting as a student should receive instant feedback 9/10 times.
* **Requirement 22. (should)**
  + **Requirement:** Whilst taking a formative test, students should be limited to attempting the test a maximum of 3 times.
  + **Assessment Criteria:**
    - 100% of all attempts to answer a single question for the 4th time should result in error and not count another attempt
    - The user acting as a student should be informed that no more attempts are allowed.
* **Requirement 23. (could)**
  + **Requirement:** During a formative test, when answering a question incorrectly the user could be prompted information about where they could learn more about the subject or given a hint.
  + **Assessment Criteria:**
    - After entering an incorrect answer to a question, a user acting as a student should, on 9/10 attempts, be prompted with a hint or a link.
    - This link should give relevant information about the question’s topic.

**Task 2 – Use Case Diagrams**

Education Software Test System



Student

Lecturer

<<Extend>>

**Task 3 – Use Case Descriptions**

|  |  |  |
| --- | --- | --- |
| Use Case No: 1 | Use Case Name: Login Screen | Rating Must Have |
| Purpose: Allow the user to login in to their unique account and direct them to an appropriate screen. | | |
| Main Actor: All users will need to use this feature. | | |
| Description:  When the user opens the program, they should be taken to the login screen. The user will then be prompted to enter their username and their password. When this is entered the user will press an enter button and then the system will validate the username and password combination against one on a secure database. When validating if the information is correct then it will get the users level of access and then depending on if the user is a lecturer or a student then it will direct them to the correct screen.  Non-Functional Requirements:   * 1: Security, The users must all have unique Logins and a minimum complexity of password to login. * 2: Validation, All the storage of passwords for the users must be stored in an encrypted format * 3: Access, The level of access that the user should have will be stored in the same area as the password. * 4: Ease of Use, It must be clear to the user where the information should be input. * 5: Response, The time between the user pressing enter and the validation being complete must be less than 1 minute. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case No: 2 | Use Case Name: Create Test | | Rating: Must Have |
| Purpose: Customer should be able to create a specified type of test and add questions for the students to answer | | | |
| Main Actor: Lecturer | | Secondary Actors: | |
| Description: Upon choosing to start a new test, the teacher must be prompted to select a type of test from formative or summative. The teacher must then add a title to the test. The teacher must then be prompted to add at least 1 question to at most 10 questions. The accepted answers must also be added. The lecturer must then post the test. The system will then notify the students that they have a new test for completion and it will appear on their own screens  Non-functional requirements:   * Ease of use: the lecturer should be able to select the type of test and add questions to create the test without challenges * Accuracy: Upon selecting the type of test, the features of formative/summative must apply 100% of the time. Each test must always only contain the questions added by the lecturer, and their corresponding answers. The system can be more flexible and accept answers with misprints * Security: only lecturers must be able to post tests * Response: the system should carry out all operations individually, within 30 seconds | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case No: 3** | **Use Case Name: Completing a test** | | ***Rating must have*** |
| Purpose: Customer should be able to answer the assigned questions on a test then submit the test. | | | |
| Main Actor: Customer | | *Secondary Actors:* | |
| *Description:*  The customer acting as a student must have the chosen quiz ready to fill out displayed to them. They will then proceed to answer the questions they wish. For formative assessments, as the student enters the answers to each question the system will notify them if their answer is correct/incorrect. If their answer is incorrect they will have two more attempts at the question (max. 3 attempts) and a hint could be displayed. For summative tests, the student must submit their test and are only allowed one attempt. The system will give them their marks after submission.  Non-Functional Requirements:   * Ease of use: The customer should be able to answer the test without instruction or training. * Consistency: The final answers submitted by the students should be the answers taken for the final mark and comparison against correct solutions 100% of the time. * Security: The customer should only be able to submit tests for their own account. Any errors regarding student scores being mixed should be avoided 100% of the time. * Response: The customer should be notified when they have either submitted or saved a test 10 seconds after completing the test. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case No:  4 | Use Case Name: View test statistics | | Rating: Must Have |
| Purpose:  Customer should be able to view the statistics and results of tests that have been completed. | | | |
| Main Actor: customer | | Secondary Actors: | |
| Description:  The customer acting as a teacher should only be provided with the option of viewing summative tests results that have had their due date passed. Once the customer has clicked on one of the summative tests, they should have the option as to how they want to view the test’s results e.g. Chronological, Alphabetical. The customer should also be provided with an option to view the statistics in several different ways (average mark, the most common mark, the highest mark and the lowest mark.)  The customer shouldn’t be able to view the individual test results of formative tests (each individual students test results), they should only be able to view the test statistics of formative tests.  Non-Functional requirements   * Security: The customer must be logged into the system in order to view the results of tests * Response: The system should provide the correct statistics and results of the system within 1 minute of the statics/results being requested. * Accuracy: The system should provide the correct statistics to the customer. * Consistency: The system should work 100% of the time. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case No: 5 | Use Case Name: Giving feedback on tests | | Rating: should have |
| Purpose: after the student submit the test the system will generate a feedback for the student. | | | |
| Main Actor: Lecturer | | Secondary Actors: Student | |
| Description:  After finishing every attempt for the formative assessment students should be provided with a feedback on their performance (the answered that they got correctly or incorrectly). Also, they can be guided to a lecture to help them with the answer that they got wrong. However, in summative assessment the feedback will be available to student after the deadline.  Non-Functional requirements:   * Security: student shouldn’t be able to modify anything from the feedback. * Ease of use: the user will be able to use the system without needing any training. * Response: the time between the user finishing the test and giving the feedback should be less than 30 second. | | | |