TeamVas Testplan

**Introduction**

I want to ensure software quality through various testing methods and continuous integration/continuous deployment (CI/CD) processes. The following testing methods will be applied:

* Unit testing
* Integration testing
* End-to-end testing
* Performance testing
* Code quality analysis.
* User acceptance tests

**Test Environment**

For TeamVas I will be using .NET 7, even though it has a standard term support, converting to a higher .NET will be easier since it will include the latest features. I expect TeamVas to grow, therefore I will be using a MySQL database since growth in data can be efficiently managed in MySQL. I will use a C# asp.net backend with a React framework-based frontend. I will use GitHub for version control and GitHub actions for my CI/CD.

**Test Strategy**

Unit Testing with MSUnit: Testing individual components for correctness.

Integration Testing: Testing combined parts of the application to ensure they work together.

End-to-End Testing: Testing the entire application flow from start to finish. To ensure that the application works on from start to finish.

Performance Testing: Testing the application’s performance under load.

Static Code Analysis with SonarQube: Ensuring code quality and adherence to standards. Make sure high parts of code are covered by tests.

Secure web development using OWASP: By looking at the OWASP top 10, I will see what the most critical security risks are to web applications. I will then take these risks into account in my web application.

**CI/CD Integration**

Setup a build pipeline that compiles the code and runs unit tests on every check-in. Build a test pipeline which executes integration, end-to-end and performance tests. Integrate SonarQube analysis in the CI/CD pipeline for better software quality. Automate building docker containers inside the CI/CD pipeline.

**Unit test cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Preconditions** | **Steps** | **Expected result** |
| **UTC-01** | A teacher should be able to make courses for the students to attend. | Logged in as a teacher. | -Navigate to the courses page.  -Press the button at the top right to create a course.  -Fill in course details and press confirm. | A course is successfully created. |
| **UTC-02** | A user should be able to send private messages to other users. | User is logged in with the correct credentials | -Navigate to the profile of another user you want to message.  -Click on send message.  -Type the message and press send. | The message should be sent to the other user. |
| **UTC-03** | A student should be able to hand in assignments of work to a specific course. | Student is logged in with the correct credentials. | -Navigate to the courses page.  -Press the button at the top right to hand in an assignment.  -Upload the assignment and press confirm. | The assignment should now be submitted. |

**Integration test cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Preconditions** | **Steps** | **Expected result** |
| **ITC-01** | A user should be able to chat/call in group chats. | User is logged in and inside a group chat with 1 or more members. | -Navigate to the group page you want to chat/call in.  -Type a message and press send or press the call button in the top right corner. | The message or call should be sent by the user and received by others in the group chat. |
| **ITC-02** | A user should be able to mute his microphone or not show his camera. | The user is logged in and inside of a call. | -Navigate the mute button or camera off button.  -Press either button. | User should now be muted or have a turned off camera. |
| **ITC-03** | A teacher should be able to grade the work of a student. | Teacher is logged in and has a submitted assignment of a student. | -Navigate to the course and press top right button to view assignments.  -Click on the grade you want to give the student along with a note.  -Press confirm. | The teacher gave a grade to the student. |

**Test Type Reference Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Functionality** | **Unit Tests** | **Integration Tests** | **End-to-End Testing** | **User Acceptance Testing** |
| Teacher makes courses for students. | Yes | No | Yes | Yes |
| User should be able to send private messages. | Yes | Yes | Yes | Yes |
| Student should be able to hand in assignments of work to a specific course. | Yes | Yes | Yes | Yes |
| A user should be able to chat/call in a group chat. | No | Yes | Yes | Yes |
| A user should be able to mute his microphone/not show his camera. | No | Yes | Yes | Yes |
| A teacher should be able to grade the work of a student. | Yes | Yes | Yes | Yes |

**User acceptance testing**

For the user acceptance tests to be most successful, it would be preferred to be tested by users with different experience in the field. Hence TeamVas will be tested by a various number of users. These testers will go over the applications flow and test all its features, giving feedback at the end.