Quality Requirements

1. Performance Requirements

A. Availability

Definition

The tool should always be available for all users and any unavailability communicated.

Acceptance Criteria

- The website and application should have an uptime of **90%**
- Any planned mantainance will occure outside of normal university hours
- Planned downtime will be indicated or communicated with clear UI changes.

B. Scalability

Definition

The product should be at first able to scale horizontally to accommodate for typical traffic that would be expected by an institutions during peak times, then this should extend out to multiple institutions per region.

Acceptance Criteria

- Autoscalling of the microservices used will be able to accommmodate changing traffic on a regular basis.
- Different instances of the back-end will be deployed per major reagon to decouple scaling, allowing for different regions and resources to scale up accordingly.

C. Perfomance

Definition

The different clients of the product should be fast and higly responsive.

Acceptance Criteria

End-to-End testing will be conducted regulary to test server response times and ensure that a user can quickly perfom each use case or multiple use cases in an acceptable time frame.

2. Usability

Definition

Users should be able to navigate and interact with the main features of the tool with ease and intuitively.

Acceptance Criteria

- We will conduct regular tests with each UI implementation and ask our user to rate their experience
- The reviews will be conducted anonymously using a google form for data collection and rating
- Seperate aspects will be scored accordingly
- The user experience should have a **80% satisfactory rate**

3. Security

Definition

User data should be secured at rest and in transition in order to prevent unothorized access and comply with local laws and regulations.

Acceptance Criteria

- User data will be Encrypted in transition and Encrypted at rest.
- Users will be grouped into user pools during sign up
- This will provide limited access to data modification and access as per use case.

4. Maintainability

Definition

The software should be easily mantainable by both internal, or new external party(s), with clear concise documentation and code.

Acceptance Criteria

- Code linters and standard linting rules will be apllied to ensure code uniformality.
- A 3-person review procedure will be used to ensure that all solutions provided are effecient and employ the best procedures.
- New features implemented will be explained thouroghly with each **Pull Request**, which will also be tested and evaluated by at least 3 parties prior to deployment.
- The application will be a serverless in it's core, which means that other than optimizing implementation, there won't be a server to mantain.

5. Reliability

Definition

The software should be stable and present reliable and varified data, with low downtime and errors.

Acceptance Criteria

- Lectures will be asked to confirm their scheduled events prior to upload
- Students will be asked to confirm their courses after selection
- Error logging will be perfomed so that the nature can be assessed and prevented in the future.