

Quality Standards - Online Tutoring Platform - 12.11.2024

1. Introduction

This document outlines the quality standards for the Online Tutoring Platform to ensure it delivers reliable, secure, and user-friendly services. These standards will guide development, testing, and deployment processes.

2. Quality Goals

- **Reliability:** Achieve 99.9% uptime to ensure uninterrupted learning experiences.
- **Performance:** Ensure fast response times and scalable infrastructure.
- **Security:** Protect user data and comply with GDPR regulations.
- **Usability:** Deliver a seamless and accessible user interface for all users.
- **Maintainability:** Ensure easy updates and debugging.

3. Quality Attributes and Metrics

Attribute	Description	Metric
Performance	Fast and responsive platform under all conditions.	- Page load time < 2 seconds for 95% of requests.
Reliability	High system availability and graceful error handling.	- Uptime > 99.9%. - Recovery time < 10 minutes for critical failures.
Usability	Intuitive, consistent, and accessible UI.	- Achieve a 4.5/5 user rating in usability surveys. - Full compliance with WCAG 2.2 guidelines .
Security	Secure data storage and communication.	- 100% data encryption (in transit and at rest). - No critical vulnerabilities identified during release security scans.
Maintainability	Modular, well-documented codebase to support future growth.	- Maintain 60% unit test coverage. - Resolve critical bugs within 24 hours of detection.

4. Testing Strategy

1. Unit Testing

- Validate individual components using Jest (React) and JUnit (Spring Boot).
- Coverage Target: 60%.

2. Integration Testing

- Test interaction between front-end, back-end, and database layers.
- Tools: Postman, Selenium, and Mock APIs.

3. Load Testing

- Simulate user load to evaluate performance.
- Tools: Apache JMeter, Locust.
- Target: Support up to 1,000 concurrent users with no more than 5% error rate.

4. Security Testing

- Conduct penetration tests and vulnerability scans on each release.
- Tools: SonarCloud.

5. User Acceptance Testing (UAT)

- Collect feedback from alpha testers in December.
-

5. QA Processes

1. Version Control and CI/CD

- Use GitHub Actions for automated builds, testing, and deployment.
- Each merge request requires successful build and test execution.

2. Code Reviews

- Peer review all code contributions before merging.
- Check for adherence to coding standards (e.g., PEP8, Java conventions).

3. Bug Tracking

- Use Jira for issue tracking and prioritization.
- Categorize bugs as Critical, Major, Minor, or Trivial.

4. Monitoring (not implemented)

- Set up real-time monitoring using Prometheus and Grafana.
 - Log errors and performance metrics using ELK stack (Elasticsearch, Logstash, Kibana).
-

6. Compliance

- **Data Privacy:** Full compliance with GDPR and local regulations.
- **Accessibility:** Ensure the platform meets WCAG 2.2 AA standards for accessibility.