

Introduction:

The purpose of this document is to provide a comprehensive overview of the testing process required to ensure the fulfillment of all non-functional requirements for the Workplace Organization Application. It serves as a guide for stakeholders and developers who may be involved in the project beyond the initial development period of 5 months. The document covers the types of tests to be conducted, their objectives, test coverage, acceptance criteria, testing environment, testing automation, and the expected outcome of a test report containing the testing results.

To guarantee the successful implementation of the Workplace Organization Application, a variety of tests should be performed. These tests are designed to assess the application's compliance with non-functional requirements as outlined in the "ANALYSIS DOCUMENT - WORKPLACE ORGANIZATION APPLICATION.pdf." The technical design, tech stack, and structure of the application are detailed in the "TECHNICAL DESIGN DOCUMENT- WORKPLACE ORGANIZATION APPLICATION.pdf."

Testing Types and Acceptance criteria:

Unit Testing:

The approach for unit testing is to use the Jest unit test framework in order to test the service and controller levels of each of the microservices. Jest provides a seamless integration into a JavaScript/Typescript environment and provides an easy unit test execution with a single command.

Acceptance criteria:

To make sure unit testing is considered successful and valuable it should cover all service functions and all controller functions while passing all the tests.

Integration Testing:

Integration testing will feature a Cypress integration which will test the simpler operations (like the basic crud endpoints and button operations) within the application while the more complex integration testing operations (like user registration and user email verification) will be performed manually by the developer.

Acceptance criteria:

To make sure the integration testing is performed right and completely there should be a test for each type of functionality of the application regardless if it is covered by Cypress automated testing or manual testing. Furthermore, all tests should pass.

Quality Testing:

Quality testing will be covered by static code analysis via SonarQube which will ensure the code quality of the application and a bug free experience.

Acceptance criteria:

The analysis should conclude with SonarQube bug counter at 0. If any false positives are present they should be marked as such as to not arise again on further scans.

Performance Testing:

Performance testing will be conducted after the application is deployed. Apache JMeter is a good tool to test the performance of the application once it is deployed.

The test should consist of 1000 users making requests every second to both the auth-api and the postfeed-api. This will be the equivalent of 2000 requests per second.

Acceptance criteria:

If the pods get too loaded they should scale horizontally to distribute the load. If the Cluster VM's CPU usage goes above the alert threshold during the test an alert should be triggered to the developer's email address.

Security Testing:

Security testing will be covered by OWASP ZAP. OWASP ZAP is a dynamic application security testing [DAST] tool which provides a possibility of CI/DC pipeline integration. This test should be performed after deployment of the application.

Acceptance Criteria:

No high-risk vulnerabilities should be present after the scan. If there are false-positives they should be marked as such as not to arise during future scans.

Testing Environment and Automation:

Unit testing, integration testing and quality testing should be performed before the deployment process during a ci/cd pipeline run. The testing environment should trigger each test automatically and evaluate the results before moving to the next type of test. Static code analysis should provide a manual acceptance prompt which would give the developer time to check the results of the scan before allowing the push request and pipeline run to complete.

After the code has ran through the CI/CD pipeline the application should be redeployed on the cloud and the security testing should be performed. After that the performance testing should begin and it should be able to be visualized along with any monitoring that is available on the environment.