CIT AI Tech Consulting

Department of AI and DS
Coimbatore Institute of Technology

Muti user Chat System (MCS)
Software Test Document

Version: ***1.0d***

Document Id: MCS-std-1.0d
Authors: Akshay Karthick M S
Bharath Kumar R
Dhanush Kumar K
Rohithraj N M
Reviewed By: Dr Valliappan Raman (Head Of Department – AI & DS)
Date: 08.12.2023

Contents

1. Purpose of Review

The purpose of this review serves two main objectives: to critically assess the Test Plan for the Multiple-user Chat System project in terms of its completeness and effectiveness, and to ensure that the testing strategy seamlessly aligns with the project's objectives and requirements. This review aims to accomplish the following:

1. **Verify Alignment with Project Objectives:** Evaluate whether the defined test objectives align with the broader goals of the MCS project and ensure that the testing approach is in sync with the project's functionalities and user expectations.

2. **Evaluate Comprehensive Coverage:** Thoroughly examine the scope of testing to ensure that it comprehensively covers all critical features and functionalities of the Multiple User Chat System and verify that both functional and non-functional aspects are adequately addressed.

3. **Assess Adequacy of Resource Planning:** Scrutinize the resource requirements to ensure that they are clearly defined and align with the scale and complexity of the project and ensure that the allocation of resources, including human resources, testing environments, and test data, is appropriate for the scope of testing.

4. **Enhance Risk Assessment and Mitigation:** Evaluate the effectiveness of the risk assessment section in identifying potential challenges and uncertainties and suggest improvements to categorize and prioritize risks for better mitigation planning.

5. **Review Clarity and Readability:** Assess the overall clarity and readability of the document to ensure that stakeholders, including testers and developers, can easily understand the testing strategy and objectives.

6. **Ensure Compliance with Industry Standards:** Confirm that the Test Plan adheres to industry best practices and any specific testing standards relevant to the nature of the project.

2. An Overview of the Findings

The test plan for MCS project demonstrates a commendable level of detail in addressing the testing requirements for the Multiple User Chat System. The review identified the following key aspects:

1. **Extensive Feature Coverage:** The test plan adequately covers a wide range of features of the Multiple User Chat System, including real-time messaging, user authentication, and group chat functionalities.

2. **User Scenarios and Use Cases:** The user scenarios and use cases are well-defined, contributing to a clear understanding of how the system will be tested under different user interactions and scenarios.

3. **Considerations for Load Testing:** The plan recognizes the importance of load testing, especially in the context of a chat system that supports multiple users simultaneously. However, more details are needed regarding how the load will be generated and specific success criteria.

4. **Security Testing:** The plan acknowledges the significance of security testing in a chat system. It outlines the intention to perform security assessments, but additional details are required regarding specific testing scenarios and tools to be used.

5. **User Interface Testing:** While functional testing is well-covered, there is a need for more emphasis on testing the user interface to ensure a seamless and intuitive user experience.

6. **Integration Testing:** Integration testing is mentioned, but it is important to outline specific integration points and test scenarios to ensure a thorough examination of how various components interact.

7. **Clarity in Documentation:** The document could benefit from improved clarity in certain sections, such as the test objectives, to avoid potential misinterpretations.

8. **Mobile Device Testing:** Considering the widespread use of mobile devices, it is essential to emphasize the importance of testing across various mobile devices, encompassing different operating systems and screen sizes. This inclusion will enhance the overall testing process and ensure optimal performance across all mobile platforms.

3. Brief Documentation of the Inspection Meeting

The inspection meeting for MCS project took place on 30/11/2023 at G-meet. The main objective of the meeting was to thoroughly examine the Test Plan for the Multiple User Chat System, focusing on its completeness and effectiveness.

Agenda:

1. **Clarification on Test Objectives:** Team members agreed that the test objectives needed further clarification. The discussion centered around the importance of having clear, measurable objectives that align with the project requirements. It was emphasized that explicit test objectives are essential for guiding the testing process effectively.

2. **Resource Allocation for Load Testing:** Concerns were raised regarding the lack of specific details regarding resource allocation for load testing. The team highlighted the critical nature of load testing in a chat system and stressed the

need for clearly defined resources, including servers and virtual users, to accurately simulate real-world scenarios.

3. **Suggestions for Improving Risk Assessment:** The risk assessment section received significant attention during the meeting. Team members recommended a more structured approach, which involved categorizing and rating identified risks. This approach would enable better prioritization and allow the team to address the most critical risks first.

Outcomes:

- The Test Objectives section will be revised to provide more explicit details, incorporating the feedback received during the meeting.
- A dedicated section specifying resources for load testing will be added to ensure proper allocation and planning for this crucial testing phase.
- The risk assessment section will be enhanced by categorizing and rating identified risks, providing a more comprehensive understanding of potential project challenges.

Next Steps:

- The updated Test Plan will be shared with all team members for their final approval.
- A follow-up meeting has been scheduled for 03/12/2023 to verify the successful implementation of the suggested modifications.

4. List of Identified Faults and Actions Taken to Correct

1. **Test Objectives Refinement:**
   Fault: The test objectives lack specificity and do not align with user expectations.
   Action: Revamp the test objectives to explicitly align with user expectations identified during UAT, ensuring that they are specific and measurable.

2. **Load Testing Resource Planning:**
   Fault: The resource allocation for concurrent user load testing lacks details.
   Action: Add a dedicated section that specifies the number of simulated users, network conditions, and server configurations for robust load testing based on findings from UAT.

3. **Functional Testing Coverage:**
   Fault: Some critical functional scenarios, identified during UAT, are missing from the test plan.
   Action: Enhance the Functional Testing section by incorporating scenarios discovered during UAT to ensure comprehensive coverage of all critical functional scenarios.

4. **User Authentication Testing:**
   Fault: The testing coverage for various user authentication scenarios is inadequate.

Action: Integrate additional test cases in the Functional Testing suite to validate user authentication under different conditions, addressing issues reported during UAT.

5. **Message Delivery Verification:**
   Fault: Instances of delayed message delivery were observed during UAT.
   Action: Implement specific test cases in Functional Testing to verify and ensure real-time message delivery reliability, addressing the issues identified during UAT.

6. **Mobile Application Performance:**
   Fault: UAT revealed responsiveness issues in the mobile application.
   Action: Include a dedicated section in Functional Testing to validate the responsiveness of the mobile application across different devices and screen sizes, ensuring that it meets user expectations.

7. **Security Vulnerability Testing:**
   Fault: Security vulnerabilities identified during UAT are not adequately addressed.
   Action: Enhance the Security Testing section to include testing for identified vulnerabilities and implement corrective measures to ensure that the application is secure.

5. Quantitative Summary of Faults

- Total Faults Identified: 7
- Critical Faults: 3
- Major Faults: 3
- Minor Faults: 1

Breakdown by Severity:
- Critical Faults Percentage: 42.86%
- Major Faults Percentage: 42.86%
- Minor Faults Percentage: 14.28%

6. Check List

1. **Test Plan Content:**
   - Clearly defined test objectives that meet user expectations.
   - Thorough scope of testing that covers all critical features.
   - Comprehensive test deliverables, including test cases, test summary reports, and defect reports.

2. **Resource Planning:**
   - Well-defined resource requirements for both Functional Testing and Load Testing.

- Sufficient allocation for load testing, taking into account concurrent user scenarios identified during UAT.

3. **Risk Assessment:**
   - Categorization and rating of risks based on UAT and FT findings.
   - Mitigation strategies for high-priority risks, particularly those that impact user experience.

4. **Functional Testing:**
   - Inclusion of functional scenarios identified during UAT.
   - Adequate coverage for user authentication, message delivery, and mobile application responsiveness.
   - Integration of security testing measures to address vulnerabilities discovered during UAT.

5. **Performance and Load Testing:**
   - Detailed plan for load testing scenarios, considering findings from UAT.
   - Evaluation of server response times under different loads.
   - Stress testing to identify system limitations and potential bottlenecks.

6. **User Acceptance Testing (UAT) Integration:**
   - Alignment of the test plan with UAT feedback and observations.
   - Incorporation of critical UAT scenarios into the overall testing strategy.

7. **Mobile Application Testing:**
   - Dedicated testing for mobile application responsiveness across various devices and screen sizes.
   - Verification of mobile-specific features such as push notifications and background processes.

8. **Regression Testing:**
   - Develop a strategy for regression testing to guarantee that any new features or fixes do not negatively affect the current functionality.

9. **Documentation:**
   - Keep the documentation up-to-date to reflect any modifications made during the review process.
   - Ensure that the documentation is accessible to all team members involved in testing.