Quality Assurance
Test Plan

TABLE OF CONTENTS

Document Revisions

Date	Version	Description	Author

Document Approval

Signature	Signature
Date: <Date>	Date: <Date>
Name: <Name>	Name: <Name>

1 Introduction

1.1 Purpose

Purpose of the document

1.2 Scope of the Document

1.3 Overview

1.4 Definitions, Acronyms, and Abbreviations

Term	Description
API	Application Program Interface
BA	Business Analyst
UI	User Interface

1.5 References

2 Scope of Testing

2.1 Product Overview

2.2 Product Risks

2.3 Test Coverage

Mention all the testing types used in the project.

Test Type	Applicable (Yes/No)	Description
Smoke testing	Yes	Ensure basic functionalities work in each QA build.

2.4 Functional Requirements

List down all the major functional requirements that will be tested.

Use Case No.	Functional Requirement	Iteration
F1	User Authentication	Initial
F2	User Registration	Initial
F3	User Login/Logout	Initial
F4	Password Recovery	Initial
F5	Self-Registration	Initial
F6	Dashboard Functionality	Iteration 1
F7	Help and Support Links	Iteration 1
F8	Room Access from Dashboard	Iteration 2
F9	Global Chat Functionality	Iteration 2
F10	Adding/Removing Contacts	Iteration 2
F11	Private Messaging	Iteration 2
F12	User Profile Management	Iteration 3
F13	Calendar View	Iteration 3
F14	Event Management in Calendar	Iteration 3
F15	My Profile Management	Iteration 4
F16	Contacts and Messages Management	Iteration 4

F17	Room Lists Functionality	Iteration 4
F18	Recordings Functionality	Iteration 5
F19	Presentation Room Testing	Iteration 5
F20	Conference Room Testing	Iteration 6
F21	Interview Room Testing	Iteration 6

2.5 Non-functional Requirements

S.N.	Non Functional Requirement	Criticality
1	***Performance Efficienc***y: Ensures Apache OpenMeeting runs at optimum speed and responsiveness, meeting user expectations under varying workloads. Users benefit from high performance efficiency.	High
2	***Usability:*** An emphasis is placed on the user-friendliness of Apache OpenMeeting, emphasizing intuitive interfaces and easy navigation. Platform usability ensures that users can efficiently and effectively use its features.	High
3	***Security:*** Provides protection for data and user information in Apache OpenMeeting. Users' data is protected against unauthorized access and data breaches with high-level security measures.	High
4	***Reliability***: Maintains Apache OpenMeeting's consistency and reliability. In addition to providing users with stable and trustworthy platforms, high reliability means minimal downtime, system failures, or errors.	High
5	***Load Factor:*** *The platform must be able to handle concurrent users and data processing efficiently due to load factor considerations.*	Medium

2.6 Out of Scope

- **Hardware-related Implementation**: Specific hardware configurations or requirements are not covered in the project.
- **External Plugin Support**: Out-of-scope tasks include extensive support for external plugins that are not provided by the free community.
- **Specialized Mobile Functionality**: Additional features, such as full functionality for small-sized mobile devices, are not included in the current version.
- **Advanced UI Technologies**: Full mobile compatibility will be achieved after the user interface is ported to HTML5/WebRTC.

3 Test Deliverables and Schedule

Lists all the major QA deliverables with their due dates.

Deliverable	Description	Date
Test Data Request	Describes a different set of data required to complete Functional testing successfully	1 week before each test execution cycle

Once the project schedule is finalized, the above delivery dates will be updated.

4 Test Design

Mention the test design strategies or methods that use to formulate the test cases.

4.1 Test Case Review Process

Mention how the test review process is going to be handled.

5 Test Execution

5.1 Approach

Mention the test execution approaches with Objective, entry criteria, exit criteria for each.

5.1.1 Smoke Testing

5.1.1.1 Objective

5.1.1.2 Entry Criteria

5.1.1.3 Exit Criteria

5.2 Test Data Requirements

5.3 Test Reporting

Report	Objective/Content	Owner	Audience	Frequency
Test Execution Report				
Test Status Report				

5.4 Defect Management

Mention the defect management of the project

6 Test Team

6.1 QA Team Organization

In general, the testing structure within an Apache openmeeting can be represented as follows:

1 QA Manager/Director

The QA Manager/Director is in charge of overseeing every aspect of a software product's or system's quality assurance process. They establish and enforce quality assurance policies and procedures, manage the QA team, and ensure all quality standards are met by the software.

1.1 Test Lead/Coordinator
1.2

The Test Lead/Coordinator is in charge of planning, organizing, and executing the software testing process. Test Lead collaborates with the QA team to create test plans and schedules, assign test cases to testers, and monitor testing progress.

1.2.1 Automation Engineer

The Automation Engineer is in charge of creating and updating automated test scripts. They collaborate with the QA team to find test cases that can be automated, and they create and maintain scripts that can be used to automate these tests.

1.2.2 Manual Testers

Manual testers are in charge of manually executing test cases. They collaborate with the QA team to identify test cases that must be performed manually, and they execute these tests in accordance with the test plan.

1.2.3 Performance Testing Engineer

The Performance Testing Engineer is in charge of testing the performance of a software product or system. They collaborate with the QA team to discover performance bottlenecks and build and execute test cases to measure the software's performance.

1.2.4 Security Testing Engineer

The Security Testing Engineer is in charge of testing a software product's or system's security. They collaborate with the QA team to detect security flaws and build and execute test cases to evaluate the software's security.

1.2.5 Usability Testing Specialist

The Usability Testing Specialist is in charge of testing a software product's or system's usability. They collaborate with the QA team to uncover usability concerns and build and run test cases to evaluate the software's usability.

1.2.6 Test Analyst

The Test Analyst is in charge of analyzing and documenting software product or system requirements. They collaborate with the QA team to identify test cases that must be conducted and document testing findings.

6.2 Roles and Responsibilities

Role	Name(s)	Responsibilities
QA Manager/Director		• Oversees the entire quality assurance process. • Specifies the QA strategy and policies. • Coordination with other teams and stakeholders is required.
Test Lead		• Manages the QA team's day-to-day operations. • Schedules tests and assigns duties to team members. • Reports progress and issues to the Quality Assurance Manager.
Automation Engineer		• Creates and updates automated test scripts. • Collaboration with developers to incorporate automated tests into the build process. • Performs automated tests and analyses the findings.
Manual Testers		• Performs manual testing. • Creates and executes test cases on project specifications. • Tracks and logs defects, as well as communicating with developers.
Performance Testing Engineer		• Creates and conducts performance tests to validate the system's scalability. • Identifies and fixes bottlenecks in performance.
Security Testing Engineer		• Performs security audits to discover vulnerabilities. • Addresses security vulnerabilities in collaboration with developers.
Usability Testing Specialist		• User experience and interface design are prioritized. • Usability testing is carried out to ensure that the program is user-friendly.
Test Analyst		• Creates test strategies by analyzing requirements and specifications. • Works with the development team to comprehend the design and architecture.

7 Test Environment

Mention the test environment requirements and justify them.

7.1 Software and Hardware

Details of the software and hardware needs.

No.	Machine	Purpose	Software Requirements	Hardware Configuration	Quantity
	Server	Hosting OpenMeetings	Linux OS, Apache Tomcat, JDK 17, Apache Maven 3.6.0 or higher, Git, MySQL	CPU: Intel Core i7 or equivalent, RAM: 16GB, Storage: 500GB SSD Gigabit Ethernet	1
	Client	Test the Apache OpenMeetings application	Linux OS, Web browser (Chrome, Firefox), Java Runtime Environment	CPU: Intel Core i3 or equivalent RAM: 4GB Storage: 128GB SSD Webcam and microphone	Multiple
	Client	Test the Apache OpenMeetings application	Windows OS, Web browser (Chrome, Firefox), Java Runtime Environment	CPU: Intel Core i3 or equivalent RAM: 4GB Storage: 128GB SSD Webcam and microphone	Multiple
	Client	Test the Apache OpenMeetings application	Mac OS, Web browser (Chrome, Firefox, Safari), Java Runtime Environment	CPU: Intel Core i3 or equivalent RAM: 4GB Storage: 128GB SSD Webcam and microphone	Multiple
	Network	Connect server and clients	Gigabit Ethernet	Network switch, cables	
	Database Server	Store OpenMeetings data.	Windows Server 2016 or 2019 MySQL 5.7 or later	CPU: Intel Core i3 or equivalent RAM: 8GB Storage: 128GB SSD Gigabit Ethernet	1

Here is a justification for the above hardware and software test requirements for Apache OpenMeetings.

Server Requirements

Hardware:

- **CPU**: Intel Core i7 or equivalent - This ensures that the server can handle the processing load of hosting OpenMeetings.
- **RAM**: 16GB - This ensures that the server has enough memory to run OpenMeetings smoothly.
- **Storage**: 500GB SSD - This ensures that the server has enough storage to store OpenMeetings data and logs.
- **Gigabit Ethernet**: This ensures that the server has a fast enough network connection to handle the traffic from multiple clients.

Software:

- Linux OS: This is a stable and reliable operating system that is well-suited for running OpenMeetings.
- Apache Tomcat: This is an open-source web application server that is required to run OpenMeetings.
- JDK 17: This is the Java Development Kit that is required to run OpenMeetings.
- Apache Maven 3.6.0 or higher: This is a build automation tool that is required to build and deploy OpenMeetings.
- Git: This is a version control system that is used to manage the OpenMeetings source code.
- MySQL: This is an open-source relational database management system that is required to store OpenMeetings data.

Client Requirements

Hardware:

- CPU: Intel Core i3 or equivalent - This ensures that the client computer can run the OpenMeetings application smoothly.
- RAM: 4GB - This ensures that the client computer has enough memory to run the OpenMeetings application.
- Storage: 128GB SSD - This ensures that the client computer has enough storage to store OpenMeetings data and logs.
- Webcam and microphone: These are required for video and audio conferencing.

Software:

- Linux OS, Windows OS, or Mac OS: These are all supported operating systems for running the OpenMeetings application.
- Web browser (Chrome, Firefox, or Safari): These are all supported web browsers for running the OpenMeetings application.
- Java Runtime Environment: This is required to run the OpenMeetings application.

Network Requirements

Hardware:

- Gigabit Ethernet: This ensures that the network can handle the traffic from multiple clients.
- Network switch, cables: These are required to connect the server and clients to the network.

Database Server Requirements – Optional (Database can be setup on Main server also)

Hardware:

- CPU: Intel Core i3 or equivalent - This ensures that the database server can handle the processing load of storing OpenMeetings data.
- RAM: 8GB - This ensures that the database server has enough memory to run MySQL smoothly.
- Storage: 128GB SSD - This ensures that the database server has enough storage to store OpenMeetings data.
- Gigabit Ethernet: This ensures that the database server has a fast enough network connection to handle the traffic from the application server.

Software:

- Windows Server 2016 or 2019: This is a supported operating system for running MySQL.
- MySQL 5.7 or later: This is a supported version of MySQL for running OpenMeetings.

7.2 *Tools*

Lists the tools used by the QA team:

No.	Purpose	Tool
1	Defect Management	JIRA
2	CI/CD	Jenkins
3	Test automation	Selenium WebDriver
4	Version Control	Git
5	Performance testing	JMeter
6	Security testing	Burp Suite
7	Test case management	TestLink
8	Load testing	Locust
9	Communication	Slack, Confluence

Here is a justification for the above tools that can be used by the QA team for Apache OpenMeetings.

Defect Management: JIRA is a popular and reliable defect management solution. It provides a centralized platform for tracking, managing, and prioritizing software issues, guaranteeing effective communication between QA and development teams.

CI/CD : Jenkins provides continuous integration and continuous delivery (CI/CD) possible. Automation of the build and deployment processes guarantees that code changes are immediately merged, tested, and deployed, resulting in shorter development cycles and earlier issue exploration.

Test automation : Selenium WebDriver is an extremely effective tool for automating web application testing. Because Apache OpenMeetings is a web-based platform, Selenium allows for the construction of automated test scripts to validate the application's functionality across several browsers and systems.

Version Control: Git is a popular distributed version control system. It enables QA teams to manage and track changes to source code, engage with developers, and maintain a dependable codebase while maintaining version control and traceability.

Performance Testing: JMeter is a powerful performance testing software. It enables the QA team to simulate multiple users and evaluate the performance, scalability, and reliability of Apache OpenMeetings under various load scenarios, assisting in the identification and resolution of performance bottlenecks.

Security testing : Burp Suite is a comprehensive tool for checking the security of online applications. Given the sensitivity of data in Apache OpenMeetings, using Burp Suite assists in identifying and addressing potential security vulnerabilities while assuring the overall security posture of the application.

Test case management : TestLink is a free and open-source test management application for organising test cases, test plans, and test execution data. It serves as a centralised repository for test artefacts, encouraging collaboration and assuring thorough test coverage.

Load Testing : Locust is a scalable and easy-to-use load testing tool. It enables the QA team simulate thousands of concurrent users, allowing them to evaluate the system's performance under high stress and ensure its ability to handle a large user base.

Communication : Slack and Confluence are key tools for communication and collaboration. They allow for effective communication within the QA team, documenting of test plans, sharing of test results, and general team collaboration.

8 Assumptions