**1. Introduction**

This test plan outlines the testing approach for the "Product Name Resolution Pro It Able to Meeting Management System." The primary goal is to ensure the system meets the required functionality, performance, security, and usability standards. Testing will be conducted using various tools such as Selenium (for UI testing), Manual Testing, Rest Assured (for API testing), K6 (for load testing), Postman (for API testing), Cucumber with Gherkin syntax (for BDD), and Excel/Word for reporting and documentation.

**2. Test Objectives**

The objectives of this test plan are:

1. To validate the functionality of the meeting management system.
2. To ensure the correct resolution of product names and other system-related functionalities.
3. To verify API endpoints for proper communication and data exchange.
4. To conduct performance and load testing to ensure system scalability.
5. To check the usability of the system across various devices and platforms.
6. To verify the security of the application and ensure compliance with security standards.

**3. Scope of Testing**

* **Functional Testing**: Verification of the system's features (e.g., meeting creation, scheduling, attendee management, notifications, and product name resolution).
* **API Testing**: Ensure that all API endpoints are functional and return expected responses using Rest Assured and Postman.
* **Performance Testing**: Use K6 for load testing to simulate multiple users and evaluate the system's performance under stress.
* **User Interface Testing**: Use Selenium to automate UI interactions and manual testing for usability and user experience.
* **Security Testing**: Verify the security of user data, authentication, and authorization protocols.
* **Regression Testing**: Ensure that new code changes don't break existing functionality.

**4. Testing Tools**

1. **Selenium**: Automated UI testing for all web-based interactions and workflows.
2. **Manual Testing**: For exploratory and usability testing of UI/UX features.
3. **Rest Assured**: For API testing to ensure correct data exchange between the client and server.
4. **K6**: For load and performance testing to simulate user traffic.
5. **Postman**: For API testing, especially useful for manual inspection of API responses.
6. **Excel/Word**: For test reporting, documenting test cases, and logging issues.
7. **Cucumber with Gherkin**: For Behavior-Driven Development (BDD) to write tests in a human-readable format.

**5. Test Methodology**

* **Test Strategy**: This test will follow a combination of automated and manual testing.
* **Test Approach**:
  + **Functional Testing**: Using Selenium for automated test cases and manual testing for complex user journeys.
  + **API Testing**: Validate all REST API endpoints using Rest Assured and Postman. Ensure correct HTTP methods, status codes, and response data.
  + **Performance Testing**: Use K6 to simulate high load and measure response times, throughput, and error rates.
  + **Regression Testing**: Full regression suite using Selenium, ensuring that new changes do not impact existing features.
  + **Security Testing**: Perform security tests to validate user access control, encryption, and data protection mechanisms.

**6. Test Plan**

**6.1 Functional Testing**

* **Test Cases**:
  + Create a meeting with valid details and check for successful creation.
  + Edit and update meeting details.
  + Verify product name resolution in meeting context.
  + Add attendees to the meeting and verify notification triggers.
  + Reschedule a meeting and check for updated notifications.
  + Validate user permissions and access control (Admin, User roles).
* **Tools**: Selenium for automated testing, manual verification for edge cases.

**6.2 API Testing**

* **Test Cases**:
  + Test GET, POST, PUT, DELETE API methods for meeting management.
  + Validate API responses (status codes, response body, headers).
  + Test authentication and authorization mechanisms via API.
  + Test product name resolution API endpoints for correctness.
* **Tools**: Rest Assured for automated testing, Postman for manual API testing.

**6.3 Performance Testing**

* **Test Cases**:
  + Simulate 1000 concurrent users using K6 to test the system’s load capacity.
  + Measure response time and latency under heavy traffic.
  + Stress test the system to identify any performance bottlenecks.
* **Tools**: K6 for load and performance testing.

**6.4 UI/UX Testing**

* **Test Cases**:
  + Verify that all UI elements are correctly aligned and functional across different browsers (Chrome, Firefox, etc.).
  + Ensure the user interface is responsive on different screen sizes (mobile, tablet, desktop).
  + Test the accessibility of the application (keyboard navigation, screen reader compatibility).
* **Tools**: Selenium for automated UI testing, Manual testing for usability.

**6.5 Security Testing**

* **Test Cases**:
  + Test user authentication and authorization.
  + Check for vulnerabilities such as SQL injection, XSS, and CSRF.
  + Validate data encryption for sensitive information.
* **Tools**: Manual testing, security scanners, penetration testing.

**7. Test Deliverables**

1. **Test Cases**: Written test cases in Excel or Word.
2. **Test Results**: Logs and reports of test execution.
3. **Bug Reports**: Detailed documentation of issues found during testing (Excel or Word).
4. **Test Execution Summary**: Summary of passed/failed test cases.
5. **Performance Reports**: Load testing results (from K6).
6. **Security Reports**: Vulnerability assessment and test results.
7. **User Documentation**: Information for users regarding test procedures, features, and any known issues.

**8. Test Schedule**

| **Task** | **Start Date** | **End Date** |
| --- | --- | --- |
| Test Planning | 2025-04-28 | 2025-04-29 |
| Test Case Creation | 2025-04-30 | 2025-05-02 |
| Functional Testing | 2025-05-03 | 2025-05-10 |
| API Testing | 2025-05-03 | 2025-05-06 |
| Performance Testing | 2025-05-07 | 2025-05-10 |
| UI/UX Testing | 2025-05-05 | 2025-05-08 |
| Security Testing | 2025-05-09 | 2025-05-12 |
| Regression Testing | 2025-05-13 | 2025-05-15 |
| Final Reporting and Documentation | 2025-05-16 | 2025-05-17 |

**9. Entry and Exit Criteria**

* **Entry Criteria**:
  + The application should be feature-complete and stable enough for testing.
  + The testing environment should be set up and ready.
  + Test data should be prepared.
* **Exit Criteria**:
  + All planned tests should be executed.
  + Critical and high-severity defects should be resolved.
  + Test reports should be reviewed and shared with stakeholders.

**10. Risk and Mitigation**

* **Risk**: Unstable test environment could affect the accuracy of test results.
  + **Mitigation**: Ensure all dependencies and environments are set up before starting tests.
* **Risk**: Incomplete API documentation may lead to missed test cases.
  + **Mitigation**: Collaborate with the development team to ensure complete API documentation.
* **Risk**: High load testing may not reflect real-world user behavior.
  + **Mitigation**: Use real-world scenarios to simulate load and traffic.

**11. Approval**

The test plan and its deliverables will be reviewed and approved by the project stakeholders, including the QA manager, project manager, and relevant developers.