Amnil Pharma Solution

Quality Assurance Test Plan(QATP)

31st Mar,2025

General Information	
Customer	Amnil Pharma Solution
Documented by	Bivek Sah
Preparation Date	2025/3/27
Version	

Revision History

Version	Date	Author	Approval	Change Summary
	Mar 31, 2025	Bivek Sah		

Introduction

Amnil Pharma Solution is a specialized pharmaceutical software designed to streamline and automate key processes within the pharmaceutical industry. It provides a comprehensive IT solution for managing order processing, procurement, inventory, quality control, and compliance with regulatory standards.

1. Objective

The objective of the displayed system appears to be the management of raw materials and packaging materials (RM/PM) in a pharmaceutical setting. The system allows users to track and manage shelves, raw material properties, warehouse locations, sections, racks, and the status of stored items. It is likely designed for inventory control, compliance tracking, and procurement recommendations within the pharmaceutical industry.

- Verify that all functional modules work as expected.
- Ensure user actions trigger the correct system responses.
- Identify and log defects for resolution.
- Ensure data consistency and validation.

2. Scope

This document is related to the testing that will be done for the Amnil Pharma Solution Project by the QA team. This document complements the project approach document for project management information. It is the focal point, which will direct the testing effort for the project. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate, and how the system will react to external stimuli. Defines the features, modules, and functionalities that will be tested and identifies testing constraints, assumptions, and objectives.

3. Work Plan

- The approach, activities, resources, and timeline required to ensure software quality through systematic testing
- Ensure raw materials meet quality standards by tracking COA (Certificate of Analysis) and stability data.
- Monitor shelf locations, track expiry & retest dates, and categorize materials as "Active" or "Inactive."

- Manage user roles, access controls, and audit logs to ensure data security and accountability.
- Generate reports on stock status, shelf assignments, and procurement records for decision-making.

4. Testing Approach

Test Types

5.1 Functional Requirements

5.1.1 Integration Testing
5.1.2 Regression Testing
5.1.3 UI Testing
5.1.4 Usability Testing
5.1.5 User Acceptance Testing(UAT)
5.1.6 Alpha Testing
5.1.7 Smoke Testing

5.2 Non-functional Requirements
5.2.1 Load Testing
5.2.2 Security Testing
5.2.3 Concurrency Testing
5.2.4 Cross-Browser Testing

5.3 Automation Testing

5.1 Functional Requirements

5.1.1 Integration Testing

Ensures that different modules or components work together correctly. Detects interface issues between interconnected systems.

1. Entry Criteria

- Unit Testing is Complete
- All individual modules/components have been successfully unit tested.
- Major bugs from unit testing are fixed.

2. Exit Criteria
    - All Functional Test Cases are Executed
    - Core functionalities such as login, data processing, CRUD operations, and UI actions are validated.
3. Test Suspension and Resumption Criteria
    - If severe defects block further execution, such as crashes, data corruption, or system unresponsiveness, and all blockers and high-severity issues are resolved and verified.
    - When the test environment (servers, databases, APIs) is unstable or unavailabl,e the integration environment is restored and stable for testing.

5.1.2 Regression Testing

Regression testing will be performed when defects have been fixed and whenever necessary. The impacted scenario will be fully executed as a Regression.

1. Entry Criteria
    - The defect of the application is successfully deployed.
2. Exit Criteria
    - All functional test cases have been completed unless otherwise noted, there will be on-hold test cases due to the unresolved issues at the time of test cycle completion.
    - All defects are entered in the Google sheet.
3. Test Suspension and Resumption Criteria
    - Showstoppers or high issues, which prevent the testing of major functionality, are encountered.
    - Functionality is unstable, i.e., too many non-reproducible defects are encountered.
    - Testing will be resumed when the showstoppers or high issues, which prevent the testing of major functionality, are fixed.

5.1.3 UI Testing

UI (User Interface) Testing ensures that the graphical interface of a desktop application is functioning correctly, is user-friendly, and meets design specifications. It focuses on elements such as buttons, text fields, navigation, responsiveness, consistency, and error handling.

1. Entry Criteria
    - A stable version of the desktop application is available.
    - UI components are integrated and functional.

2. Exit Criteria
    - Verifies UI components (buttons, menus, forms, error messages, etc.) function as expected.
    - UI elements render correctly across different resolutions and environments.
    - No broken links, misalignment, or layout issues remain.

3. Test Suspension and Resumption Criteria
    - Severe UI issues such as application crashes, broken navigation, missing UI elements, or unresponsive pages.
    - UI elements function correctly, and no blocking UI issues exist.
    - APIs, databases, and third-party services are stable and functional.
    - Slow or inaccessible UI due to server downtime, incorrect deployment, or network issues.

5.1.4 Usability Testing

Usability testing is software testing that evaluates how user-friendly and efficient a software application is by testing it with real users. The primary goal is to identify usability issues, improve user experience, and ensure the application is intuitive and easy to use.

1. Entry Criteria
    - The software should be functionally stable with no critical defects.
    - Clear usability scenarios and test cases should be prepared.
2. Exit Criteria
    - Any remaining issues should not significantly impact the user experience.
    - Metrics such as completion time, ease of navigation, and efficiency meet predefined expectations.

3. Test Suspension and Resumption Criteria
    - Severe navigation issues prevent users from accessing key functions.
    - UI elements (buttons, forms, links) are missing or unresponsive.
    - Navigation, forms, and key user interactions are fully functional.
    - No severe UI/UX issues blocking task completion.

5.1.5 User Acceptance Testing(UAT)

User Acceptance Testing (UAT) is the final phase of software testing where the end users validate whether the software meets business requirements and is ready for deployment. UAT ensures that the application is functional, user-friendly, and aligns with real-world business scenarios before going live.

1. Entry Criteria
    - Business requirements should be finalized and approved.
    - All functional, integration, and regression testing should be completed.
    - Major bugs must be fixed, and there should be no high-severity defects.
2. Exit Criteria
    - No critical defects remain that would impact business operations.
    - Business stakeholders formally approve the system for production.
    - Findings, feedback, and test results are documented and reviewed.
3. Test Suspension and Resumption Criteria
    - Major defects prevent key business processes from being executed.
    - Core functionalities fail, making it impossible for users to complete essential tasks.
    - Major blocking defects are fixed and retested successfully.
    - No high-priority business process failures exist.

5.1.6 Alpha Testing

Alpha testing is a type of software testing conducted at the development site by internal testers (developers, QA team, or employees) before the software is released to external users. It helps identify bugs and usability issues early in the development cycle. Identify bugs before releasing the product to end users.

1. Entry Criteria
    - Major features are implemented and integrated.

- Initial testing phases have ensured the application is functionally stable.

2. Exit Criteria
    - Critical and high-severity defects have been resolved.
    - The software runs smoothly without crashes or slowdowns.
3. Test Suspension and Resumption Criteria
    - Major software crashes or freezes prevent further testing.
    - Severe defects in core modules make it impossible to continue execution.
    - Major blocking issues are resolved, allowing further testing.
    - All core functionalities work without crashes.

5.1.7 Smoke Testing

Smoke Testing is a type of software testing that ensures the basic functionalities of an application are working before conducting more detailed testing. It is also known as "Build Verification Testing" (BVT) because it verifies whether a new software build is stable enough for further testing.

1. Entry Criteria
    - Developers have completed basic unit testing.
    - Essential test cases covering critical functionalities are documented.
2. Exit Criteria
    - No critical defects prevent further testing.
    - Core features (e.g., login, navigation, database connection) function as expected.
3. Test Suspension and Resumption Criteria
    - The software build fails to install or deploy correctly.
    - Missing dependencies or incorrect configurations prevent the application from launching.
    - A new, stable build is provided with all core functionalities intact.
    - Installation and deployment issues are resolved.
    - Major defects that blocked testing have been resolved.

5.2 Non-functional Requirements

5.2.1 Load Testing

Load testing is a type of performance testing that evaluates how a software application behaves under expected and peak user loads. The goal is to identify performance bottlenecks, ensure stability, and determine how the system performs under normal and high-traffic conditions. Measure system response time, throughput, and resource utilization.

1. Entry Criteria
   - No critical defects should exist that may impact performance testing.
2. Exit Criteria
   - The system meets expected response times, throughput, and resource utilization goals.
   - The application performs well under increasing load.
3. Test Suspension and Resumption Criteria
   - Performance testing is suspended if showstoppers are encountered during testing.
   - The system does not show any improvement in the scaled-up environment.
   - Testing will resume when the showstoppers are resolved.

5.2.2 Security Testing

Security testing is a type of software testing that identifies vulnerabilities, threats, and risks in a system to ensure its data and resources are protected from unauthorized access, breaches, and attacks. It helps ensure that security mechanisms are working as intended and that sensitive data remains confidential and intact. Verify authentication, authorization, encryption, and data integrity. Assess resilience against penetration attempts and cyberattacks.

1. Entry Criteria
   - The software should be free from critical functional defects.
   - Security policies, encryption standards, and compliance requirements should be established.
2. Exit Criteria
   - The software complies with security guidelines and regulatory requirements.
   - Simulated attacks confirm that security measures are effective.
   - Detailed reports with risk assessments and mitigation plans are documented.

3. Test Suspension and Resumption Criteria
    - Showstoppers or high issues, which prevent the testing of major functionality, are encountered.
    - Functionality is unstable, i.e., too many non-reproducible defects are encountered.
    - Testing will be resumed when the showstoppers or high issues, which prevent the testing of major functionality, are fixed.

5.2.3 Concurrency Testing

Concurrency testing is a non-functional testing technique used to evaluate how a system performs when multiple users or processes access it simultaneously. This ensures data integrity, performance, and system stability. Identify issues like deadlocks, data corruption, and performance degradation. Validate how the Pharmaceutical Management System handles multiple users modifying data simultaneously.

1. Entry Criteria
    - All required test cases are prepared and reviewed.
    - Test data is available.
    - Necessary access credentials and permissions are assigned.
2. Exit Criteria
    - All planned test cases have been executed.
    - No high-severity defects remain unresolved.
    - Test reports are reviewed and signed off.
    - The system meets the business and technical requirements.
3. Test Suspension and Resumption Criteria
    - The application crashes, freezes, or becomes unresponsiveness under concurrent load.
    - Database deadlocks or race conditions can cause inconsistent results.
    - Application stability is restored under concurrent load.
    - Database deadlocks and race conditions are fixed.

5.2.4 Cross-Browser Testing

Cross-browser testing will make sure the UI elements of the NCELL Customer Chatbot are being correctly displayed on each agreed-upon browser.

1. **Entry Criteria**
    - The functionality of the component is clearly announced, or changes to the components are mentioned in the release notes.

- All the required software, tools, and databases are installed and connected.
- The Developer Release Notes (DRN) and Unit Test Coverage and Results have been provided with the release.

2. **Exit Criteria:**
   - The smoke test is completed.
   - No showstoppers are in the system.
   - The defects found during the smoke test are tracked.
   - Any exceptions to the above are documented and agreed upon.
3. **Test Suspension and Resumption Criteria:**
   - Smoke testing will be suspended if showstoppers are encountered.
   - Testing will be resumed when the showstoppers are fixed.

5.3 Automation Testing

Automation testing is a software testing technique that uses specialized tools and scripts to execute test cases automatically, reducing manual effort and increasing efficiency. It helps in verifying software functionality, performance, and reliability by running pre-defined test scripts on an application.

1. Entry Criteria
   - The test environment must be fully configured and stable.
   - The required test data should be available.
2. Exit Criteria
   - All planned automated test cases have been executed.
   - All critical and high-severity defects are reported and closed.
   - Automated tests validate system performance within acceptable limits.
3. **Test Suspension and Resumption Criteria:**
   - Test scripts fail due to incorrect locators, outdated UI elements, or unhandled exceptions.
   - Frequent script failures with no clear resolution lead to unreliable test results.
   - Scripts are debugged, updated, and optimized for recent changes.
   - Automated tests execute without unexpected failures.

6. Out of Scope

The items below are out of scope for the QA team.

- Setting up the QA/UAT/Staging/Production environments
- Validation of Deployment Process
- Developer testing and Unit testing
- Testing of Integrated Third-party Applications
- Test Data validation

7. Bug Report

Bug reports are created to provide the development team and the project managers with exhaustive information about the discovered defects. They must be helpful in determining causes of the errors and correcting them.

7.4.1 Defect Severity can be classified into four categories:

A. **Critical (blocker) defects:** The defect causes complete system failure or major functionality breakdown, making the application unusable.

Examples:

- The application crashes or the system hangs.
- Data loss or corruption.
- Security vulnerabilities (e.g., unauthorized access, data breach).
- Payment failure in an e-commerce application.
- Login is not working for all users.
- Fix Priority: Immediate. Must be resolved before release.

B. **High defects:** A major functionality is not working correctly, but the system remains operational. No feasible workaround is available.

Examples:

- Incorrect calculations in financial transactions.
- API failure that affects core functionality.
- Unable to save or update critical records.
- Major performance degradation under normal conditions.

- Fix Priority: High. Needs urgent attention before product release.

C. **Medium defects:** The defect affects non-critical functionality or causes inconvenience, but there is a workaround.
 Examples:

- UI elements are not displaying properly.
- Certain reports or filters are not working correctly.
- Slow response times that don't break functionality.
- Minor database inconsistencies that don't affect overall functionality.
- Fix Priority: Should be fixed before release, but is not a showstopper.

D. **Low defects:** The defect has a minimal effect on functionality and is mostly related to UI, aesthetics, or minor usability concerns.
 Examples:

- Typographical errors (spelling mistakes, incorrect labels).
- Tooltip or help text inaccuracies.
- Minor responsiveness issues in some screen sizes.
- Fix Priority: Can be fixed in future updates, low priority.

8. Priority Scale

The Priority Scale indicates the urgency to fix a fault. Priority is decided by the Project Manager or an individual or a group appointed by him/her. Guidelines for priority are listed below:

High: Resolve Immediately

Medium: Resolve Pre-Release

Low: Desired but not urgent

9. Bug Defect

- **New**: A Defect is reported by a tester.
- **Assigned**: Assigned to a developer for fixing.
- **Open/In Progress**: Developer starts working on the fix.
- **Fixed**: Developer resolves the issue and marks it as "Fixed."
- **Retesting**: Tester verifies the fix.
- **Closed**: If the fix is successful, the defect is closed.

- **Reopened**: If the issue still exists, it is reopened.
- **Duplicate**: it has already been reported in the system.

10. Test Design Review Process

10.1 Peer Review

A collaborative review where test cases, test scripts, or documents are reviewed by other testers (colleagues) at the same level before submission to a lead. The goal is to catch errors, improve clarity, and ensure the logical flow of test cases. A tester writes test cases and shares them with a fellow QA team member. The peer checks for completeness, correctness, missing scenarios, and ambiguity. Suggestions and corrections are discussed. The tester updates the test cases based on feedback.

10.2 Lead Review

A formal review conducted by a QA Lead or Test Manager to validate the final test design before execution. The focus is on requirement traceability, feasibility, compliance with standards, and test strategy alignment. The updated test cases (post-peer review) are submitted to the QA Lead. The lead provides feedback, and the tester updates the document accordingly. Once all feedback is addressed, the test cases are approved for execution.

11. Hardware Requirements

Computers

12. Test Environment

Integration: Development Environment, all the development will happen here.
Model: QA Environment, all the manual testing will be performed here. Automation Environment
Staging: All the performance-related testing will happen here.

13. Resources

The following table lists the tools used by the QA team:

Purpose	Tools
Defect tracking tool	Google Sheet
Test Case tool	Google Sheet
Screenshots/Video Capture	Lightshot
Performance Testing	Jmeter
Project Management	Google Drive

14. Responsibilities of Test Team Members

A. QA Tech Lead

- Managing the QA team from a technical perspective.
- Analyzing the tasks and distributing them between team members.
- Communicating with the client team and discussing all issues, providing recommendations before an update or release.
- Experience in the participation of different SDLC models like Agile, Scrum, Kanban, Sequential, Iterative, and Incremental.
- Creating test documentation, including test cases, test plans, etc.
- Proposing best practices and tools for a project.

15. Deliverable

- Test Plan.
- Test Cases
- Bug reports and reports regarding the testing progress.
- User Manual
- Release Notes

16. Test Design

This section describes the test designing strategy the QA team would follow, identifying tests that need to be executed and designing identified test scenarios in standardized formats.

- Requirement gathering
- Requirement analysis
- Write test plans
- Prepare a test case according to the plan
- Test execution
- Prepare a test report
- Deliver a bug report

- Reverify bug
- Final testing
- QA sign off