

Abstract

The Tenant Management System is developed to support shopping center tenants by facilitating financial transactions, notification, employee management, coupon scanning, issue reporting, and access to mall announcements through secure web and mobile platforms.

As the system involves critical business operations and financial data requiring high accuracy, software testing is essential to ensure system quality, correctness, and stability prior to deployment.

Throughout the development process, Manual Testing was conducted to validate user requirements, verify core system functionalities (Functional Testing), and ensure data accuracy. Defects were systematically identified and reported to prevent potential issues in real-world usage. This testing process reduces operational risks and supports the delivery of a stable, secure, and reliable system.

Introduction

The Tenant Management System supports shopping center business operations. Given its involvement in financial transactions and operational workflows, ensuring system accuracy and reliability is essential prior to production deployment.

The software testing process followed the Software Testing Life Cycle (STLC), beginning with requirement analysis and environment setup. Test cases were then designed and executed to evaluate system functionality against defined requirements. Defects were documented and communicated to the development team, followed by retesting and regression testing to ensure system stability.

Through this process, system readiness is improved, production risks are minimized, and overall confidence is strengthened.

Test Case Design Approach

Test Case Design Techniques

- **Equivalence Partitioning:** Inputs such as employee data, and form fields were grouped into valid and invalid partitions to reduce redundant test cases while ensuring coverage. [1]
- **Boundary Value Analysis:** Minimum, maximum, and boundary values were tested in form validations to detect edge-case defects. [1]

Testing Principles Applied

- **Exhaustive Testing is Impossible:** Testing focused on high-risk features such as financial transactions and user management instead of all possible combinations. [2]
- **Defect Clustering:** Greater attention was given to complex modules, including payment processing and Web Form management. [2]
- **Testing is Context-Dependent:** Testing strategies were selected based on business impact and operational risk. [2]

Technology

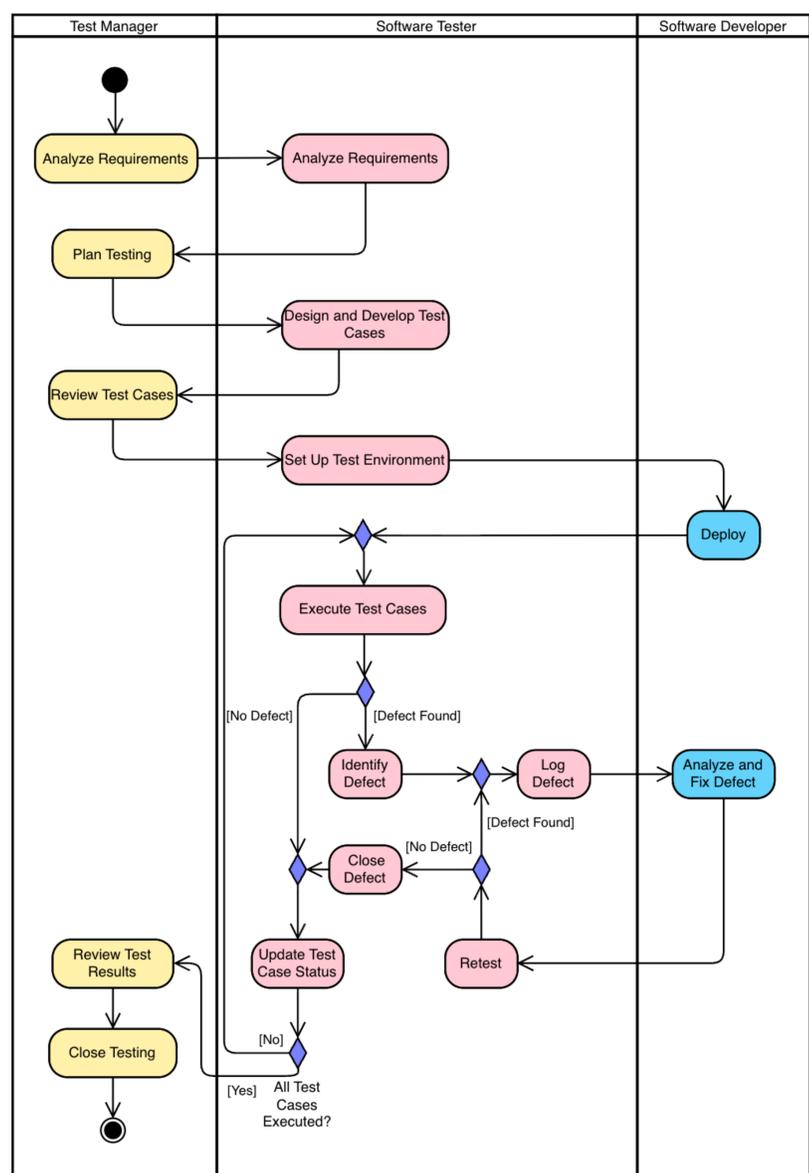
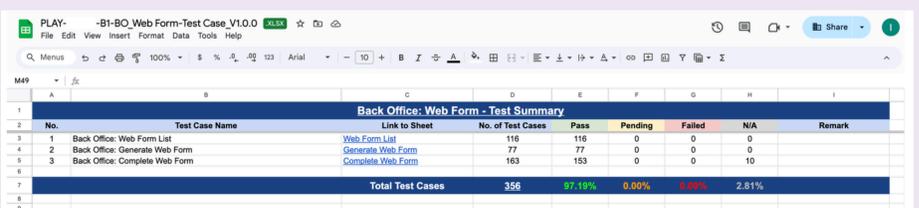


Test Results

A total of 566 test cases were designed, with 1,596 test executions performed across Back Office and application features. During testing, approximately 200 defects were identified, reported, and resolved. All identified defects were successfully retested and closed.

The feature achieved a Pass Rate of 97.19%, as shown in Figure 1, demonstrating improved system stability after defect resolution. Figure 2 illustrates a test case that initially failed due to a detected defect. After the issue was fixed, the same test case was retested and successfully passed, as presented in Figure 3, confirming the effectiveness of the defect resolution process.

Testing Process

No.	Test Case Name	Link to Sheet	No. of Test Cases	Pass	Pending	Failed	N/A	Remark
1	Back Office: Web Form List	Web_Form_List	116	116	0	0	0	
2	Back Office: Generate Web Form	Generate_Web_Form	77	77	0	0	0	
3	Back Office: Complete Web Form	Complete_Web_Form	163	153	0	0	10	
Total Test Cases			356	356	0.00%	0.00%	2.81%	

Figure 1: Test Summary Report Showing 97.19% Pass Rate

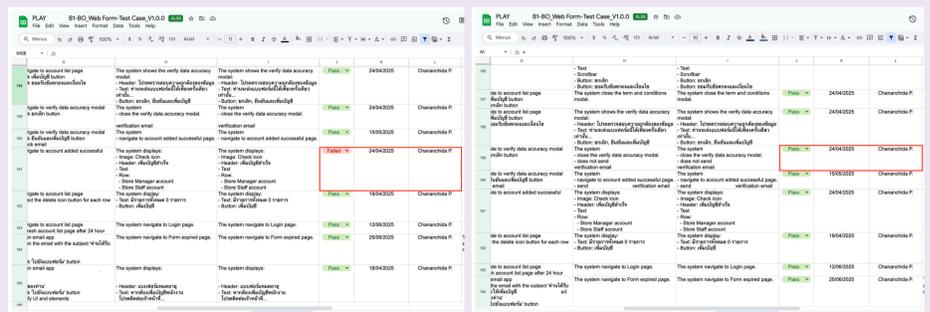


Figure 2: Failed Test Case Result (Before Fix)

Figure 3: Passed Test Case Result (After Retesting)

Overall, the results indicate that the system is stable, reliable, and ready for deployment.

Conclusion

The manual testing process for the Tenant Management System successfully identified and resolved critical defects prior to production deployment. Given the system's involvement in financial transactions and operational workflows, ensuring accuracy and reliability was essential.

By following the Software Testing Life Cycle (STLC), including structured test case design, execution, defect management, retesting, and regression testing, the system was systematically verified.

The results demonstrate improved stability and reduced production risk, indicating that the system is reliable and ready for deployment.

References:

- [1] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*, 3rd ed. Hoboken, NJ, USA: Wiley, 2011, pp. 40–61.
- [2] ISTQB®, "Seven Testing Principles," ASTQB. [Online]. Available: <https://astqb.org/istqb-foundation-level-seven-testing-principles/>. [Accessed: 10-Feb-2026].