**Libarary Management system**

**Report Summary**

# Introduction

The Library Management System (LMS) is a critical component in modern information environments, providing an efficient means to organize, manage, and access a vast array of resources. This assessment delves into the pivotal phase of the software testing lifecycle: the execution of test cases for the LMS. Having meticulously crafted a comprehensive test plan in the preceding assessment, the focus now shifts to the practical implementation and evaluation of these test cases.

The Library Management System, being a cornerstone in educational and organizational settings, demands robust and error-free functionality. The test cases designed aim to scrutinize various facets of the system, ensuring that it not only meets but exceeds the expectations of its stakeholders. This assessment specifically emphasizes the design, execution, and reporting of test cases, employing industry-standard practices to guarantee the reliability and performance of the LMS.

# Test Case Design Specification Document.

The initial element of this assessment involves the creation of a meticulous Test Case Design Specification Document. Each test case is meticulously curated to cover diverse scenarios, conditions, and input data. The structured format, as articulated in the provided Test Case Template, encompasses essential details such as Test Case ID, description, prerequisites, steps, expected results, and status. This document serves as the blueprint for the subsequent testing phases, ensuring a systematic and comprehensive approach to validating the Library Management System.

## Objectivies and outcomes

Assuring the functionality and dependability of the Library Management System (LMS) is the main goal of the testing procedure. This entails confirming that every aspect of the LMS functions as intended and accurately. Validating user authentication is a major focus of the testing; in particular, the testing verifies that the login procedure—which entails entering the password and username and clicking the login button—functions flawlessly, guaranteeing safe access to the system. The testing also tries to evaluate the user interface and experience, making sure that the LMS's navigation and user interaction are clear and error-free.

The anticipated results include a user successfully logging in with legitimate credentials, a breakdown of the system's strong points and areas for development, and verification that the LMS satisfies the functional criteria for data access and user administration. Ensuring that the Learning Management System (LMS) is a useful and easy-to-use tool in educational or organisational settings requires a thorough testing procedure.

# Test Cases

A thorough set of test cases was created throughout the implementation of our quality assurance procedures with the goal of carefully verifying the operation of our Library Management System (LMS). Our goal was to build a comprehensive set of test cases as many as we can. Every test case was meticulously created to incorporate both the expected and actual results, a detailed description, and step-by-step instructions. This arrangement was essential for evaluating the system's reaction. The purpose of the pass/fail criteria was to unbiasedly assess how well the system performed in comparison to its specifications.

## Defects Tracking Overview

I have implemented a thorough defect tracking mechanism while the Library Management System is being tested. To guarantee the LMS's resilience and dependability, this procedure is essential for locating, recording, and fixing system abnormalities. Every defect has a unique ID and is recorded with a thorough description, procedures for reproducing the issue, the expected and actual system behaviour, a priority rating, and a severity level. To represent the work being made towards their resolution, the current status of these

## Defects Prioritization and Severity Ratings

High Priority: Flaws that need to be fixed right away because they have a major impact on how well the system functions or how users operate.

Medium Priority: Problems that should be fixed quickly and have a moderate effect on the system.

Low Priority: Flaws that can be fixed during regular maintenance periods but have little effect.

Defects are classified as "Critical" or "Major" depending on their severity:

Critical: These are serious flaws that need immediate attention because they cause system failures or major functional interruptions.

Major: These flaws indicate problems that impact the system's functioning but do not result in total failure, even if they are nonetheless significant.A total of seventy-five faults have been found in the defect records. Fifty of these problems have been rated as critical, indicating their highest priority and seriousness. There are also 14 flaws that are classified as major, which indicate serious issues that need to be addressed. Additionally, out of all the problems in the defect management process, 11 have been categorised as low priority, meaning they are less urgently problematic.

# Testing

In this part, i proceeded to the coding stage after obtaining the test cases. Here, the test cases served as a roadmap for writing code that would precisely test the scenarios that were specified.

To ensure that every component of the LMS was completely tested in accordance with the intended scenarios, this coding was specifically designed to duplicate the actions and conditions specified in my test cases.

## Testing Procedures and Results Documentation:

I did run these tests after coding, noting the results as you went. This stage was essential for contrasting the actual outcomes with the anticipated ones specified in your test cases.

I provided a clear indication of the system's performance and reliability by carefully noting whether each test passed or failed.

## Pass and Fail Results:

Pass Results: These demonstrated that, in some circumstances, the LMS performed as anticipated, confirming the system's dependability and efficiency in those domains.

Failure Results: These were equally significant because they indicated areas in which the LMS fell short of expectations. I can concentrate on improving and fine-tuning the system by identifying these failures, which are essential for ongoing improvement.

## MemberDAOTEST

For a MemberDAO class in a Java application, I have implemented a number of JUnit tests in this code. Managing member-related database operations is apparently the responsibility of the MemberDAO class.

I verified that the addMember function functions properly by testing the addition of members with different properties (such as name, email, and address). I confirmed the database's ability to retrieve members, both individual members and all members. I have verified that the database can be used to update member information. I experimented with removing members from the database. I made tests to see how the system responds to different input scenarios, such as empty fields, invalid emails, names including special characters, and different phone and address formats.

## BookDAO

I have created JUnit tests in this code to verify the BookDAO class's ability to handle book-related activities.

Initialization: Before every test, I create a Book object and a BookDAO instance.

CRUD Tests: I practiced entering, editing, retrieving, and removing books from the database.

Validation of Input: I evaluated how the system manages several book attributes, including as titles, authors' names, ISBNs, publishing dates, genres, publisher names, conditions, and availability statuses. Particularly, you evaluated edge scenarios such as fields that include unusual characters or are empty.

Pass/Fail Conditions: Tests are considered successful when the result of the operation

Top of Form

# Conclusion

In summary, evaluating the effectiveness of the Library Management System has been made possible by my thorough testing of it. The methodical procedure, which included creating comprehensive test plans and carrying out a variety of test scenarios, has demonstrated the system's strengths and weaknesses. My practical application of these test scenarios—especially through automation—has greatly improved testing phase efficiency, enabling quick identification and cataloguing of system flaws.

Testing every aspect of the LMS, from data administration to user login, made sure that functionality matched user needs. Tracking problems ranging from small bugs to serious system malfunctions was made possible by the defect logs, which were arranged according to severity and impact. This allowed for both quick correction and future improvement planning.

Important stage in the lifecycle of the LMS is reached with the completion of this testing phase. Every test carried out represented a step closer to a robust and user-friendly system. My work has raised the bar for system dependability and opened the door for further advancements to guarantee that the LMS is a reliable resource in the ever-evolving information management industry. This demonstrates not only a well-implemented testing plan but also a dedication to quality and ongoing progress.