

Syeda Reeha Quasar

14114802719

Abstract

Create a test plan document for Library Management System.

Experiment - 5

Software Testing and Quality Assurance

# **EXPERIMENT – 5**

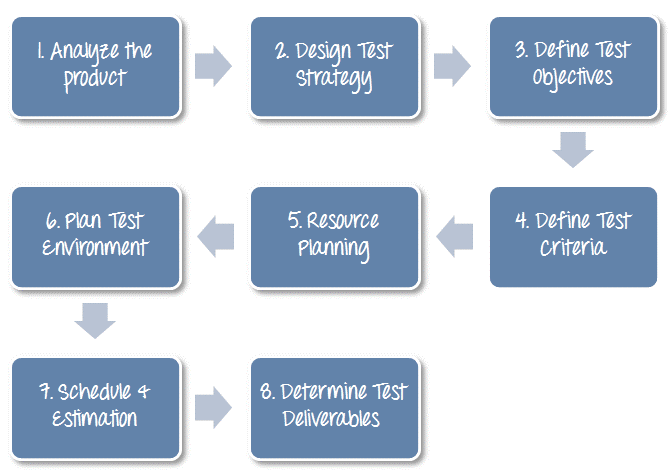
## **Aim:**

Create a test plan document for Library Management System.

## **Theory:**

A Test Plan is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product. A test Plan helps us determine the effort needed to validate the quality of the application under test. The test plan serves as a blueprint to conduct software testing activities as a defined process, which is minutely monitored and controlled by the test manager.

**How to write a test plan:**



## **Solution:**

**What are the requirements of the library management system?**

* Users can register and log in.
* The user can search for the added books and check in or out.
* The user can pay the fine or extend the duration of the borrowed period.
* The user can change the password and other profile details.
* The user can add books.
* The user can place the holds and modify existing holds.
* The user can manage the inventory of the books.

These are some of the common features expected from the library management system. So you have to write a test case for library management system to check for. In addition to these test scenarios, you have GUI-based software to check for bugs, usability, and functionality.

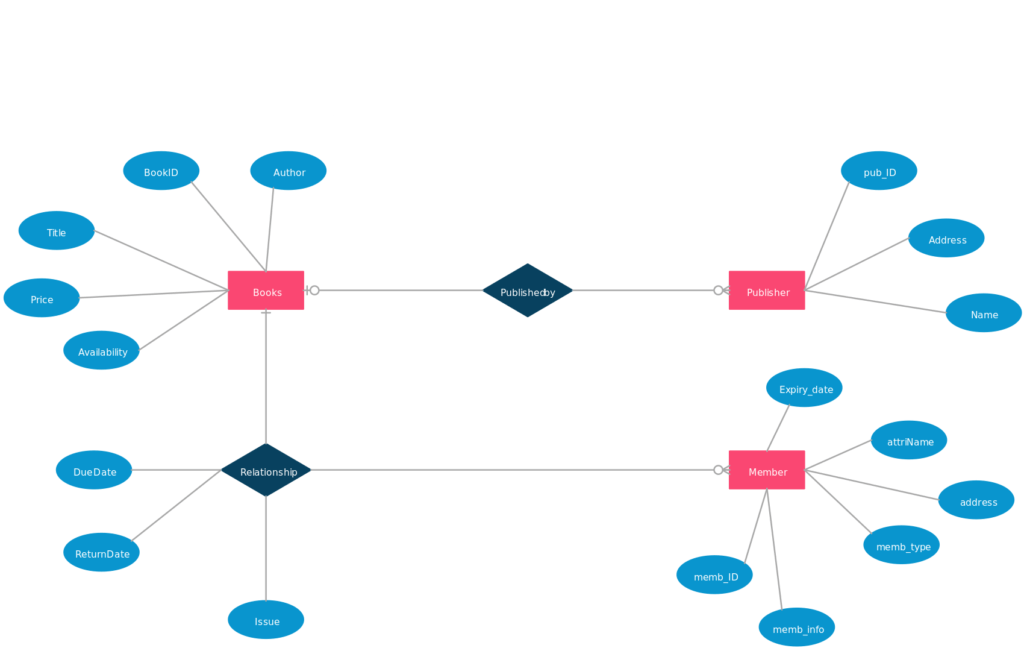
* Test Scenario For Hospital Management System
* Test Case For HR Payroll System
* Test Case For Computer Keyboard
* Test Case For Mouse
* Test Case For New Folder
* Test Case For Notepad

**What is Library Management System?**

The purpose of a library management system is to operate a library with efficiency and reduce costs. The system is entirely automated and streamlines all the tasks involved in the operations of the library. The activities of book purchasing, cataloging, indexing, circulation recording, and stock checking are done by the Library Management System software. Such software eliminates the need for repetitive manual work and minimizes the chances of errors.

The library management system software helps in reducing operational costs. Managing a library manually is labor intensive and an immense amount of paperwork is involved. An automated system reduces the need for manpower and stationery. This leads to lower operational costs.

The system saves time for both the user and the librarian. With just a click the user can search for the books available in the library. The librarian can answer queries with ease regarding the availability of books. Adding, removing, or editing the database is a simple process. Adding new members or canceling existing memberships can be done with ease.



## **Decision Table:**

## **Result:**

Test plan for Library management system was created and verified.

# **Viva Questions**

**Q1. Why we use decision tables?**

The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs. However, if different combinations of inputs result in different actions being taken, this can be more difficult to show using equivalence partitioning and boundary value analysis, which tend to be more focused on the user interface. The other two specification-based tech-niques, decision tables and state transition testing are more focused on business logic or business rules. A decision table is a good way to deal with combinations of things (e.g. inputs). This technique is sometimes also referred to as a 'cause-effect' table. The reason for this is that there is an associated logic diagramming technique called 'cause-effect graphing' which was sometimes used to help derive the decision table.

**Q2 What are the disadvantages of Decision Table testing?**

The main disadvantage is that when the number of inputs increases the table will become more complex.

**Q3. Why Decision Table Testing is Important?**

Decision Table Testing is Important because it helps to test different combinations of conditions and provides better test coverage for complex business logic. When testing the behavior of a large set of inputs where system behavior differs with each set of inputs, decision table testing provides good coverage and the representation is simple so it is easy to interpret and use.

In Software Engineering, boundary value and equivalent partition are other similar techniques used to ensure better coverage. They are used if the system shows the same behavior for a large set of inputs. However, in a system where for each set of input values the system behavior is different, boundary value and equivalent partitioning technique are not effective in ensuring good test coverage.In this case, decision table testing is a good option.

This technique can make sure of good coverage, and the representation is simple so that it is easy to interpret and use.

This table can be used as the reference for the requirement and for functionality development since it is easy to understand and cover all the combinations.

**Q4. What are the advantages of Decision Table Testing?**

1. When the system behavior is different for different inputs and not the same for a range of inputs, both equivalent partitioning, and boundary value analysis won’t help, but a decision table can be used.
2. The representation is simple so that it can be easily interpreted and is used for development and business as well.
3. This table will help to make effective combinations and can ensure better coverage for testing
4. Any complex business conditions can be easily turned into decision tables
5. In a case we are going for 100% coverage typically when the input combinations are low, this technique can ensure the coverage.

**Q5: Why is Decision Table also called a Cause-Effect table?**

Decision table testing is a software testing technique used to test system behavior for different input combinations. This is a systematic approach where the different input combinations and their corresponding system behavior (Output) are captured in a tabular form. That is why it is also called as a **Cause-Effect** table where Cause and effects are captured for better test coverage.