

SRS ON

**LIBRARY MANAGEMENT SYSTEM**

**Course code: CSE236**

**Course Title: Software project II**

**Submitted To**

Mr.Golam Rabbany

**Department of CSE**

**Daffodil International University**

**Submitted by**

**Prodip Sarker**

**ID:221-15-5850**

**Section:61\_U**

**Daffodil International University**

**LIBRARY MANAGEMENT**

**SYSTEM**

Table of Contents

[**1.1** **PURPOSE** 5](#_Toc152518380)

[**1.2** **SCOPE** 5](#_Toc152518381)

[**1.3** **REFERENCES** 5](#_Toc152518382)

[2. FUNCTIONAL REQUIREMENTS 7](#_Toc152518383)

[3. NON-FUNCTIONAL REQUIREMENTS 7](#_Toc152518384)

[4. CONTROL FLOW DIAGRAM OF LIBRARY MANAGEMENT SYSTEM 9](#_Toc152518385)

[5. USE CASE ANALYSIS 9](#_Toc152518386)

[6. UML Diagram 11](#_Toc152518387)

[7. SDLC Methodologies 12](#_Toc152518388)

[8. Testing 17](#_Toc152518389)

[9. Pricing: 19](#_Toc152518390)

[10. Payment Terms & Condition: 19](#_Toc152518393)

[11. Responsibility : 20](#_Toc152518394)

[12. Contact Us 20](#_Toc152518397)

1. **INTRODUCTION**

### **PURPOSE**

* + - The purpose of this project is to provide a friendly environment to maintain the details of books and library members.
    - The main purpose of this project is to maintain easy circulation system using computers and to provide different reports.
    - The Library System is a package to be used by Libraries to improve the efficiency of Librarians, Library employees and Users.
    - The system provides books catalogue and information to members and helps them decide on the books to borrow from the library.
    - The Librarian can keep the books catalogue updated all the time so that the members (students and the professors) get the updated information all the time.

### **SCOPE**

* + - The document only covers the requirements specifications for the Library Management System.
    - This document does not provide any references to the other component of the Library Management System.
    - All the external interfaces and the dependencies are also identified in this document.
    - The overall scope of the feasibility study is to provide sufficient information to allow a decision to be made as to whether the Library Management System project should proceed and if so, its relative priority in the context of other existing Library Management Technology

# FUNCTIONAL REQUIREMENTS

The Library Management System must have the following functional requirements:

1. The LMS should store all information about librarian and other users (students and faculty members) – their login info, books issued, etc.
2. The LMS should store all information about the books and users in two separated databases.
3. The LMS should allow searching books/journals by author, title, keywords or availability.
4. The LMS should generate request’s reports for librarian, upon which he/she could make decisions about accepting/rejecting the requests.
5. The LMS should allow users to view their personal information and status (numbers of books issued, days left, etc.)
6. The LMS should provide modules to search, request and renew books.
7. The users should be able to view their recent check-ins/checkouts, calculate and clear dues, request/recommend more books, etc.
8. The librarian must be able to add/remove books, manage users, process dues, view recommendations from users, etc.

Use case analysis of the different functionalities has been provided in section 5 for better understanding of the system

# NON-FUNCTIONAL REQUIREMENTS

#### Usability Requirements

The user interface should be interactive, simple and easy to understand. The system should prompt for the user and administrator to login to the application for proper input criteria.

#### Error Handling

Library management system shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period.

#### Security Requirements

The LMS should provide databases’ modification only for the librarian after proper authorization.

The system shall accommodate high number of books and users without any fault.

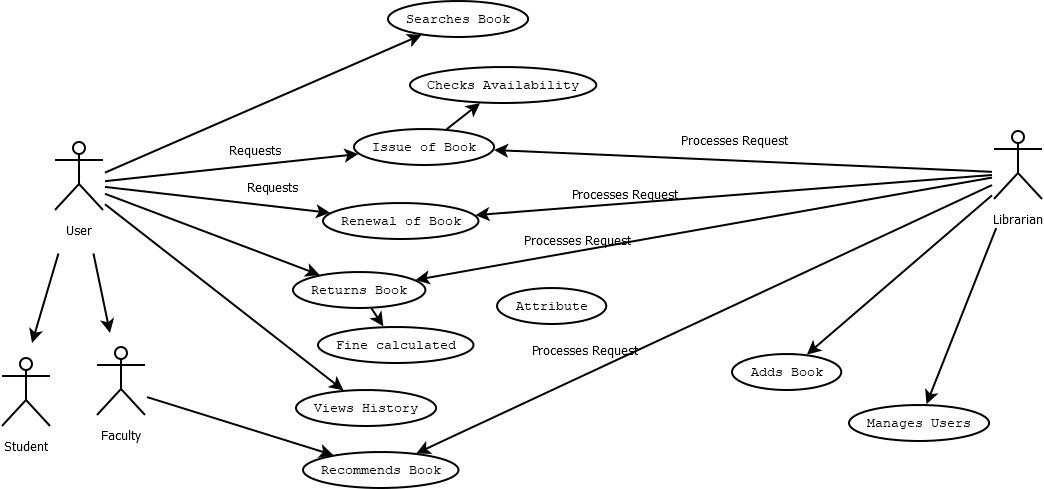
#### Software Requirements

* + 1. Database - MYSQL is used as database as it easy to maintain and retrieve records by simple queries which are in English language which are easy to understand and easy to write.
    2. Development tools and Programming language- HTML is used to write the whole code and develop webpages with CSS, JavaScript for styling work and Python Django for backend development.

#### Hardware Requirements

* + 1. Intel core i3 and above is required for a stable experience and fast retrieval of data.
    2. 2 GB RAM is required as it will provide fast reading and writing capabilities which will in turn result in better performance time.

# CONTROL FLOW DIAGRAM OF LIBRARY MANAGEMENT SYSTEM



# USE CASE ANALYSIS

|  |  |
| --- | --- |
| Name | Login |
| Actors | Student/Faculty/Admin |
| Preconditions | Existence of database and correctly installed web-based server. |
| Description | When program is opened using browser the user sees logon page. |

|  |  |
| --- | --- |
| Name | Manage Users |
| Actors | Admin |
| Preconditions | Existence of database and correctly installed web-based server. |
| Description | Manages the users and their actions. |

|  |  |
| --- | --- |
| Name | Add/remove Books to the library |
| Actors | Admin |
| Preconditions | Existence of database and correctly installed web-based server. |
| Description | Manages the books in the library. |

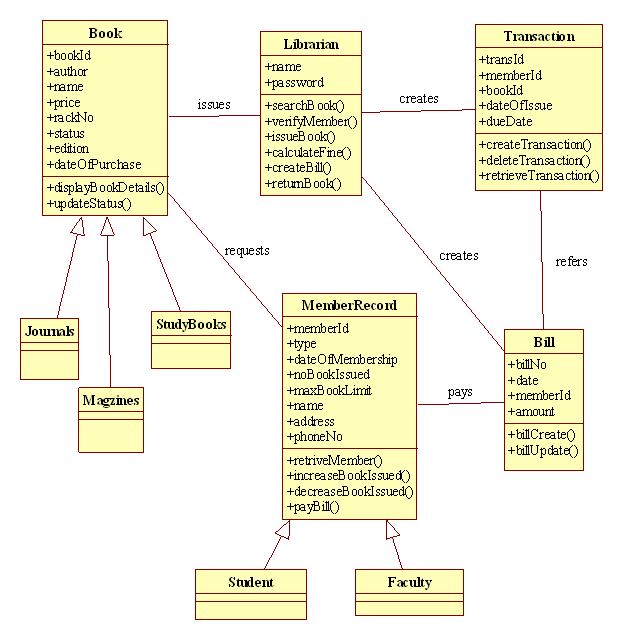
|  |  |
| --- | --- |
| Name | Processing issues |
| Actors | Admin |
| Preconditions | Existence of database and correctly installed web-based server. |
| Description | Manages the request of issues and requests by users/faculty. |

|  |  |
| --- | --- |
| Name | Request issue/return |
| Actors | Student/Faculty |
| Preconditions | Existence of database and correctly installed web-based server. |
| Description | Student/Faculty can request for books issuing or can return the issued books. |

|  |  |
| --- | --- |
| Name | Search books |
| Actors | Student/Faculty |
| Preconditions | Existence of database and correctly installed web-based server. |
| Description | Student or Faculty can search for the required books. |

|  |  |
| --- | --- |
| Name | Recommend books |
| Actors | Faculty |
| Description | Faculty can recommend the new required books. |
|  |  |

# UML Diagram



# SDLC Methodologies

What are SDLC Methodologies?

The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time. The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands. The SDLC defines and outlines a detailed plan with stages, or phases, that each encompass their own process and deliverables.

Types of SDLC Methodologis :

There are commonly 6 types of SDLC Methodologis.

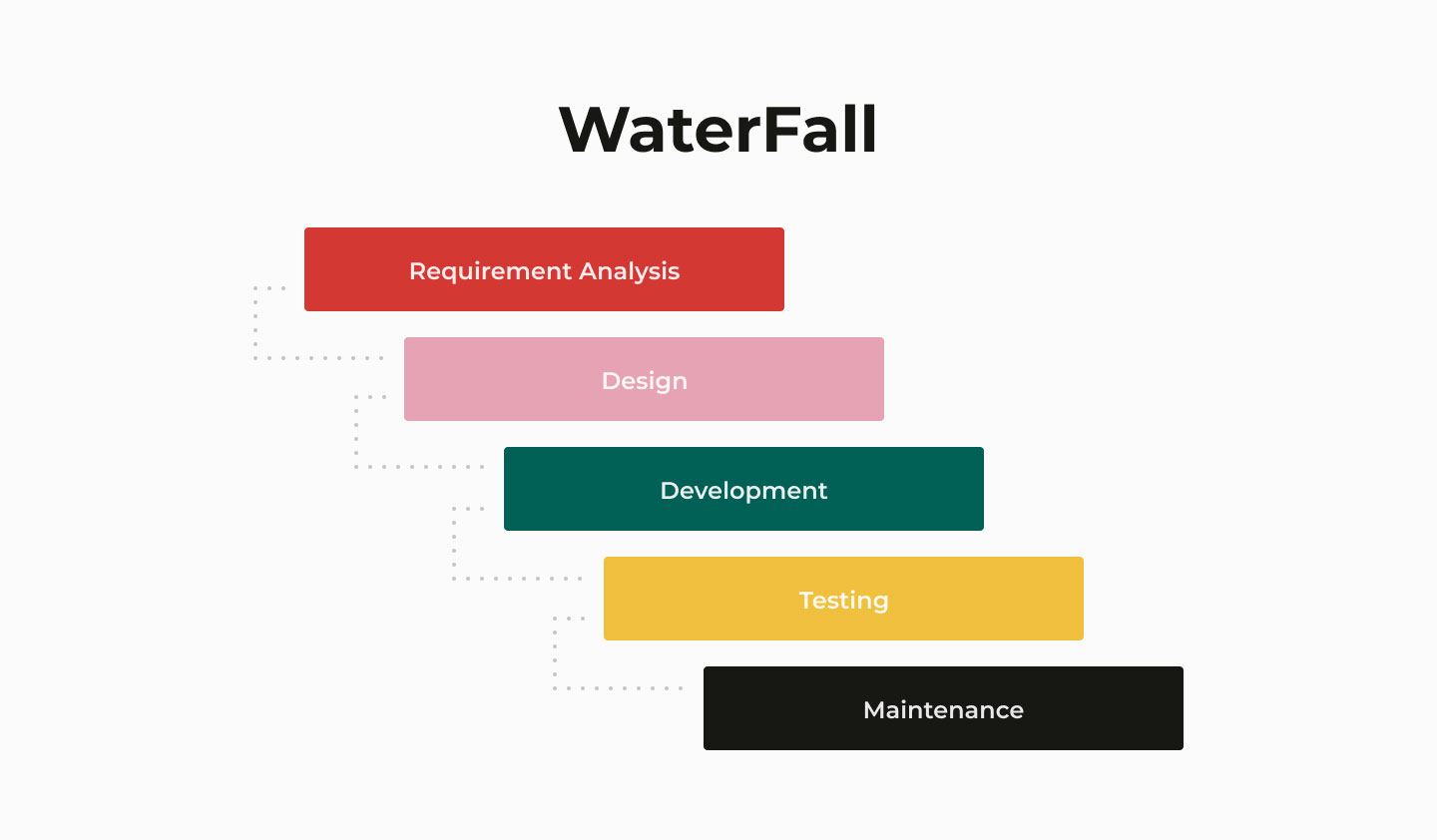
1. Waterfall
2. Agile
3. Lean
4. Iterative
5. Spiral
6. V-Shaped
7. RAD

I think Agile Model is best for me. Agile is a software development life cycle method that focuses on what really matters. Instead of wasting hours or days to come up with a detailed plan that may or may not be suitable for the project's execution, agile teams can start right away with a small portion of work.

Advantages of Agile Methodology

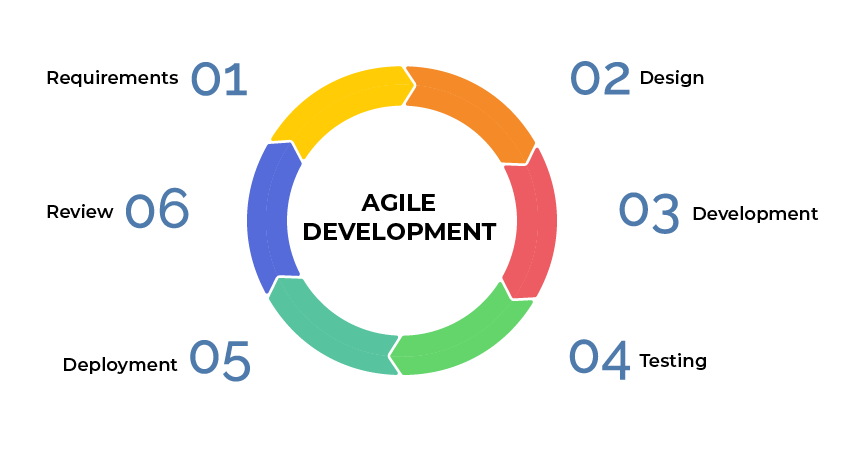
1. Customer satisfaction is rapid, continuous development and delivery of useful software.
2. Customer, Developer, and Product Owner interact regularly to emphasize rather than processes and tools.
3. Product is developed fast and frequently delivered (weeks rather than months.)
4. A face-to-face conversation is the best form of communication.
5. It continuously gave attention to technical excellence and good design.
6. Daily and close cooperation between business people and developers.
7. Regular adaptation to changing circumstances.
8. Even late changes in requirements are welcomed

**Waterfall:**



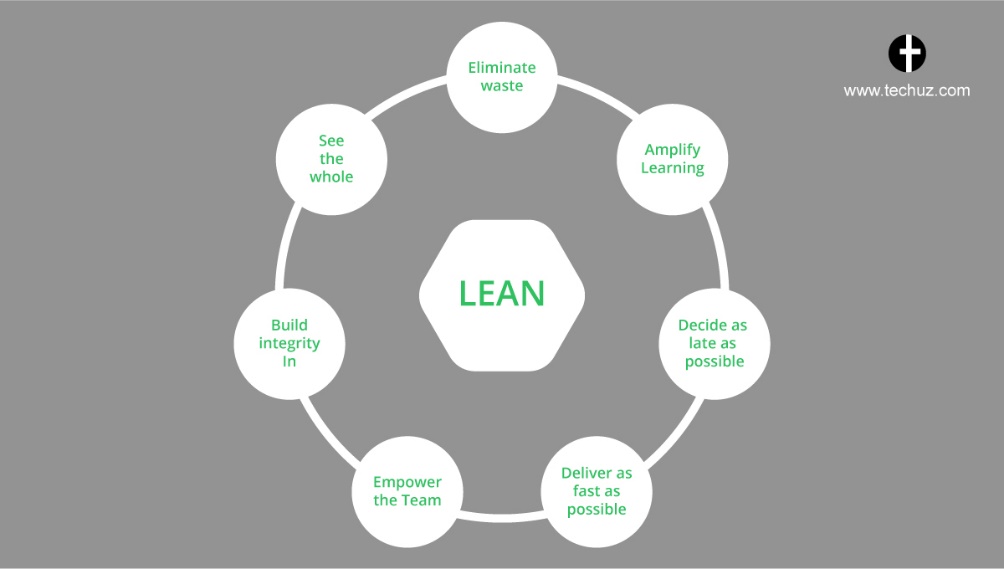
Waterfall represents the oldest, simplest, and most structured methodology. Each phase depends on the outcome of the previous phase, and all phases run sequentially. This model provides discipline and gives a tangible output at the end of each phase. However, this model doesn’t work well when flexibility is a requirement. There is little room for change once a phase is deemed complete, as changes can affect the cost, delivery time, and quality of the software.

**Agile:**



The agile methodology produces ongoing release cycles, each featuring small, incremental changes from the previous release. At each iteration, the product is tested. The agile model helps teams identify and address small issues in projects before they evolve into more significant problems. Teams can also engage business stakeholders and get their feedback throughout the development process.

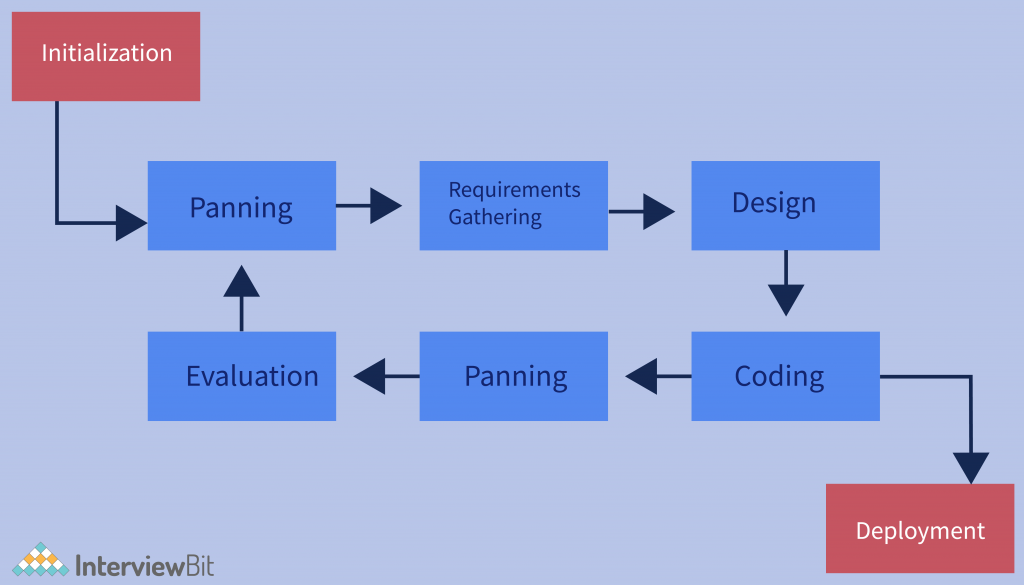
**Lean:**



The software development is inspired by lean manufacturing practices and principles. The lean principles encourage creating better flow in work processes and developing a continuous improvement culture. The seven lean principles are:

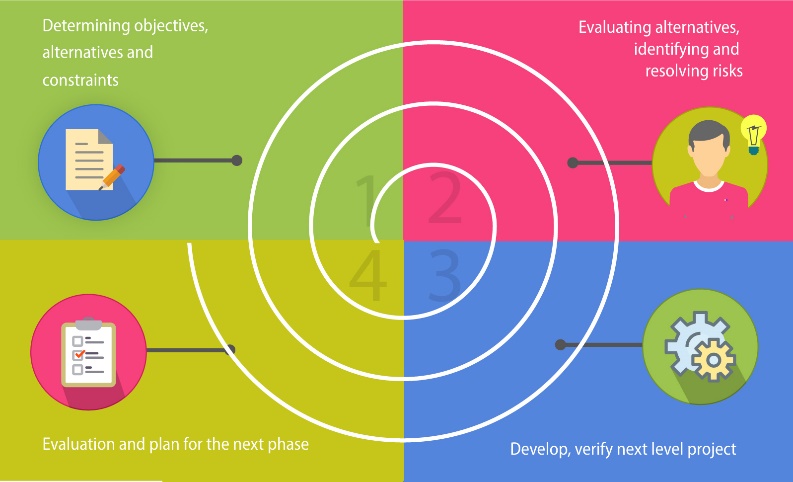
* Eliminate waste
* Amplify learning
* Make decisions as late as possible
* Deliver as fast as possible
* Empower your team
* Build integrity in
* Build holistically

**Iterative:**



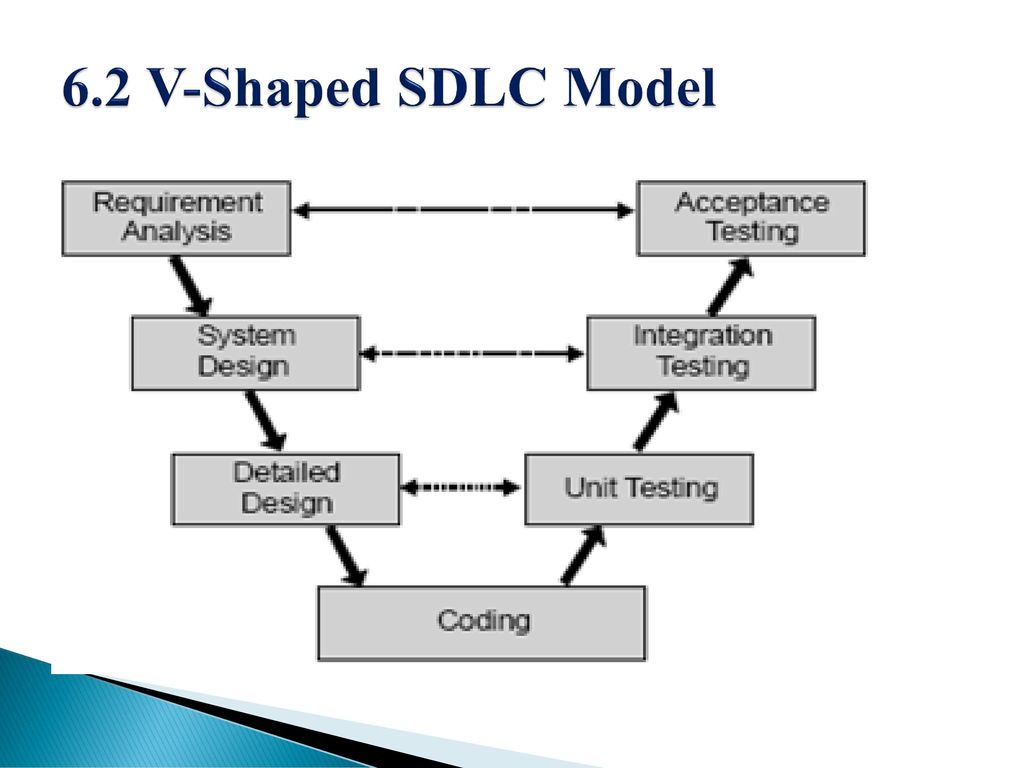
In the iterative process, each development cycle produces an incomplete but deployable version of the software. The first iteration implements a small set of the software requirements, and each subsequent version adds more requirements. The last iteration contains the complete requirement set.

**Spiral:**



In the spiral development model, the development process is driven by the unique risk patterns of a project. The development team evaluates the project and determines which elements of the other process models to incorporate.

**V-Shaped:**



In the V-shaped model, verification phases and validation phases are run in parallel. Each verification phase is associated with a validation phase, and the model is run in a V-shape, where each phase of development has an associated phase of testing.

RAD:



[Rapid Application Development](http://blog.capterra.com/what-is-rapid-application-development/) is an SDLC methodology to accelerate the development with rapid prototyping of the product.It is an iterative model that focuses more on functional software and prototyping and less on planning.There are four main phases of RAD namely, requirement planning, user design, construction and cutover. The first two being the most important as they go through several iterations until the product is fully refined as per the user requirements.

# Testing

# 

Following features will be used for testing

* This Application will be tested with Agile model
* Application will be tested by PHPUnit
* Application will also be tested by Server Expert with MySQL.
* **Testing Approach:**

The best software testing approach for an library management systems project is a combination of black-box testing and white-box testing. Black-box testing focuses on the external behavior of the system without considering its internal structure, while white-box testing focuses on the internal structure of the system to ensure that it is implemented correctly.

**Black-box testing** techniques such as equivalence partitioning, boundary value analysis, and use case testing are well-suited for testing library management systems because they can uncover a wide range of bugs and defects. These techniques are also relatively easy to apply, even for testers who are not familiar with the internal workings of the system.

**White-box testing** techniques such as unit testing, integration testing, and code coverage can also be used to test library management systems. These techniques are particularly important for ensuring that the system's code is well-designed, bug-free, and secure.

In addition to black-box and white-box testing, other important testing approaches for library management systems include:

* **Performance testing:** This type of testing ensures that the system can handle a large number of users and transactions without crashing or slowing down.
* **Security testing:** This type of testing identifies and fixes vulnerabilities that could allow attackers to steal data or compromise the system.
* **Usability testing:** This type of testing ensures that the system is easy for users to navigate and understand.
* Testing types, techniques and tactics:

**1. Functional Testing:** Functional testing verifies that the system's functionalities operate as intended according to the specified requirements. It encompasses various techniques, including:  
   
 **a. Unit Testing:** Unit testing focuses on individual units of code, ensuring they function

correctly in isolation.

**b. Integration Testing:** Integration testing validates the interaction between different modules or components of the system.

**c. System Testing:** System testing assesses the entire system as a whole, ensuring it meets all functional requirements.

**2.Non-functional Testing:** Non-functional testing extends beyond the core functionalities, evaluating aspects like performance, usability, security, and compatibility. It includes techniques such as:

**a. Load Testing:** Load testing evaluates the system's ability to handle increasing user loads and maintain performance under pressure.

**b. Stress Testing:** Stress testing pushes the system beyond its expected capacity to identify performance bottlenecks and breaking points.

**c. Usability Testing:** Usability testing assesses the user interface's ease of use, intuitiveness, and overall user experience.

**d. Penetration Testing**: Penetration testing involves ethical hacking attempts to identify and exploit security vulnerabilities.

* Testing tools  
   1.Functional Testing Tools:
* **Selenium:**

**Purpose:** Automated web browser testing.

**Features:** Supports various programming languages, browsers, and operating systems.

* **Cypress:**

**Purpose:** End-to-end testing for modern web applications.

**Features:** Fast execution, real-time reloading, and easy setup.

* **TestComplete:**

**Purpose:** Comprehensive automated testing across desktop, web, and mobile applications.

**Features:** Record and playback, script-free testing, and support for various scripting languages.

**2.Performance Testing Tools:**

* **Apache JMeter:**

**Purpose:** Load testing and performance measurement.

**Features:** Supports a variety of applications, protocols, and provides graphical analysis of performance reports.

* **LoadRunner:**

**Purpose:** Load testing for applications and websites.

**Features:** Simulates virtual users, measures system performance, and identifies bottlenecks.

* **Gatling:**

**Purpose:** Open-source load testing framework based on Scala.

**Features:** Highly scalable, supports scripting in Scala, and provides detailed performance metrics.

**3.Security Testing Tools:**

* **OWASP ZAP (Zed Attack Proxy):**

**Purpose:** Automated security testing for web applications.

**Features:** Scanning for common security vulnerabilities, automated scanners, and various tools for manual testing.