

# **LIBRARY MANAGEMENT**

A library management system is essentially a relational database customized for use in running a library and supporting its operations. As we have seen, databases underlie many organizational applications of computers.

## **Requirements:**

### **Cataloging:**

The software should allow librarians to create and maintain a comprehensive catalog of all the materials in their collection. This includes the ability to add new items, edit existing records, and delete obsolete ones. It should also support a range of metadata standards, such as MARC (Machine-Readable Cataloging) and Dublin Core.

### **Circulation:**

The software should provide a system for managing circulation activities, including check-in and check-out of materials, renewals, holds, and reservations. It should also allow librarians to generate reports on circulation activities, such as borrowing trends and overdue items.

### **Patron management:**

The software should allow librarians to manage patron accounts, including registration, renewal, and deletion. It should also provide features for tracking patron activities, such as borrowing history and fines.

### **Acquisitions:**

The software should provide tools for managing the acquisition process, including the ability to create purchase orders, track order status, and receive and process invoices. It should also support integration with external vendors and other library systems.

### **Reporting:**

The software should provide a range of reporting features, including circulation statistics, inventory reports, patron activity reports, and financial reports.

### **Search and discovery:**

The software should provide a user-friendly interface for searching and discovering library resources. This includes features such as advanced search, faceted search, and relevancy ranking.

### **Interlibrary loan:**

The software should provide tools for managing interlibrary loan activities, including the ability to request and track items from other libraries.

**Integration:**

The software should support integration with other library systems, such as online public access catalogs (OPACs), discovery layers, and digital repositories.

**Security:**

The software should provide robust security features, including user authentication and authorization, data encryption, and access control.

**Scalability:**

The software should be scalable to support libraries of all sizes, from small school libraries to large research libraries. It should also support growth and expansion over time.

**Risks:****Data loss or corruption:**

If the Library Management software is not designed to handle data properly, there is a risk that data could be lost or corrupted, leading to inaccurate or incomplete records of library resources and patron activities.

**Integration issues:**

Library Management software often needs to integrate with other library systems, such as discovery layers and digital repositories. If these integrations are not properly configured, they could result in errors, data inconsistencies, or other issues.

**Training and support:**

Library Management software can be complex, and staff members may require training and support to use it effectively. If this training and support are not provided, staff members may be unable to use the software to its full potential, leading to errors, inefficiencies, or other issues.

**Vendor lock-in:**

Some Library Management software vendors may make it difficult or expensive for organizations to switch to another system. This could limit the organization's flexibility and options for upgrading or changing their software in the future.

**Cost:**

Library Management software can be expensive to purchase and maintain, particularly for smaller libraries or organizations with limited budgets. If the costs of the software are not properly managed, they could become a financial burden for the organization.

## Configure JIRA for managing the project to solve the identified problem.

### STEP 1:

Click on the project button – Your created projects will be displayed.


#### Create project

You can change these details anytime in your project settings.

\*Indicates a required field.

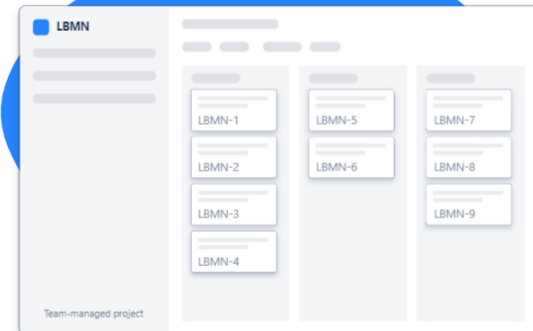
Name\*  
LBMN

Template [More templates](#)

**Scrum**  
Jira Software  
Sprint toward your project goals with a board, backlog, and roadmap.

[Show more](#)

[Back](#) [Create project](#)



### STEP 2:

Create an issue.

**Create issue**

Project\*  
LBMN (LBMN)

Issue type\*  
Task

Status  
To Do

Summary\*  
Update library ID

Description  
Update customer's library ID to database

☐ Create another issue

[Cancel](#) [Create](#)

### STEP 3:

Click on Board and drag the issues to its respective fields.

The screenshot shows the Jira Software interface for the LBMN project. The left sidebar contains navigation options: LBMN Software project, PLANNING (Roadmap, Backlog, Board), and DEVELOPMENT (Code, Project pages, Add shortcut, Project settings). The 'Board' option is selected. The main area displays 'All sprints' with a search bar and filters for Epic and Sprint. The board is divided into three columns: 'TO DO 2 ISSUES', 'IN PROGRESS 1 ISSUE', and 'DONE'. The 'TO DO' column contains two issues: 'update book id' (LBMN-4) and 'arrange the books' (LBMN-5). The 'IN PROGRESS' column contains one issue: 'Update library ID' (LBMN-1). The 'DONE' column is empty. A 'Quickstart' button is visible in the bottom right corner.

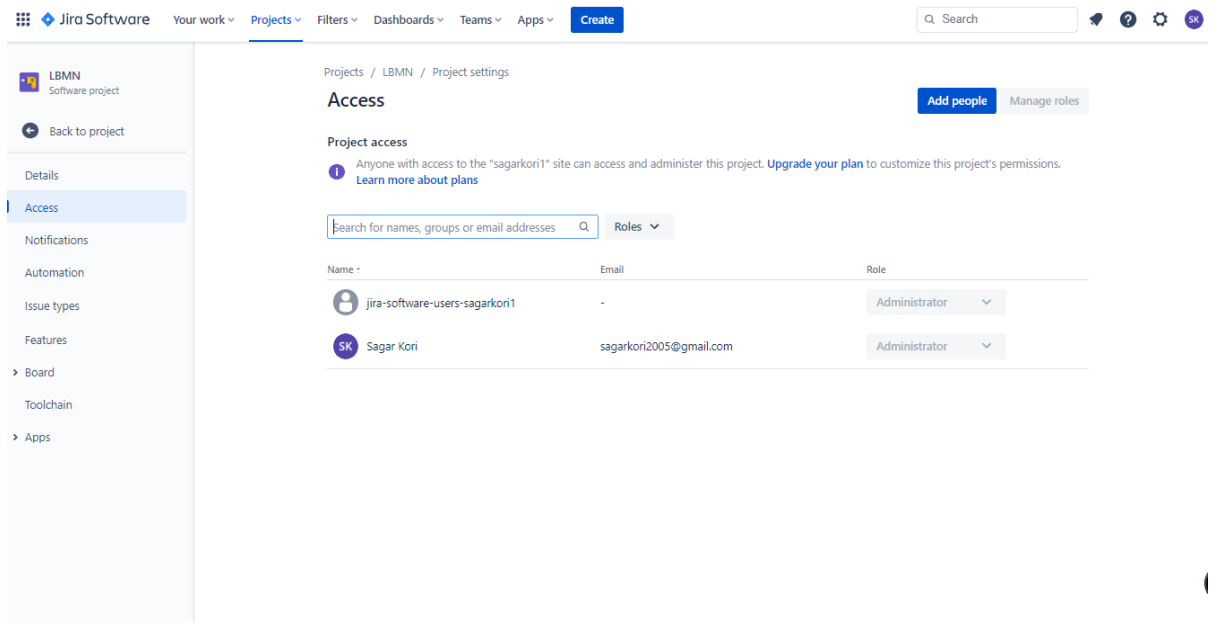
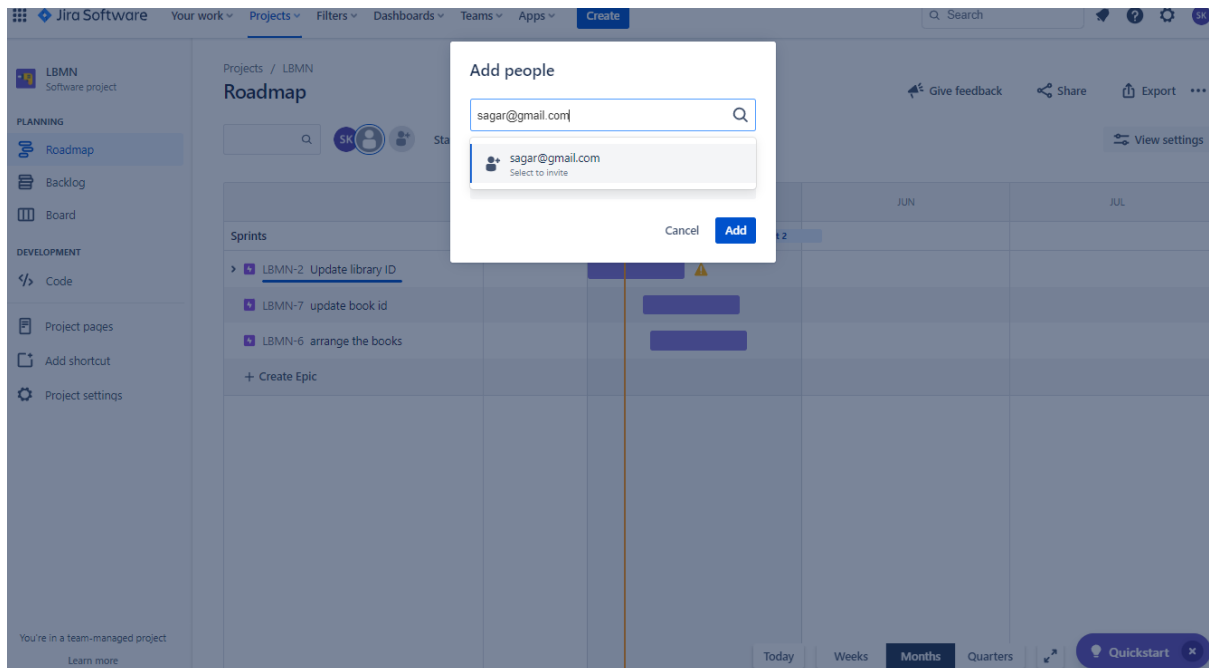
### STEP 4:

Click on Roadmap and create the Epic and child processes.

The screenshot shows the Jira Software interface for the LBMN project in the Roadmap view. The left sidebar is the same as in Step 3, but the 'Roadmap' option is selected. The main area displays the 'Roadmap' with a search bar and filters for Status category and Epic. The roadmap is a timeline view showing sprints and issues. The timeline is divided into columns for MAY, JUN, and JUL. The 'Sprints' section shows 'LBMN Sprint 1, LBMN Sprint 2' spanning from early May to early June. Below the sprints, three issues are listed: 'LBMN-2 Update library ID' (spanning from early May to early June), 'LBMN-7 update book id' (spanning from early May to early June), and 'LBMN-6 arrange the books' (spanning from early May to early June). A '+ Create Epic' button is visible at the bottom left of the roadmap area. A 'Quickstart' button is visible in the bottom right corner.

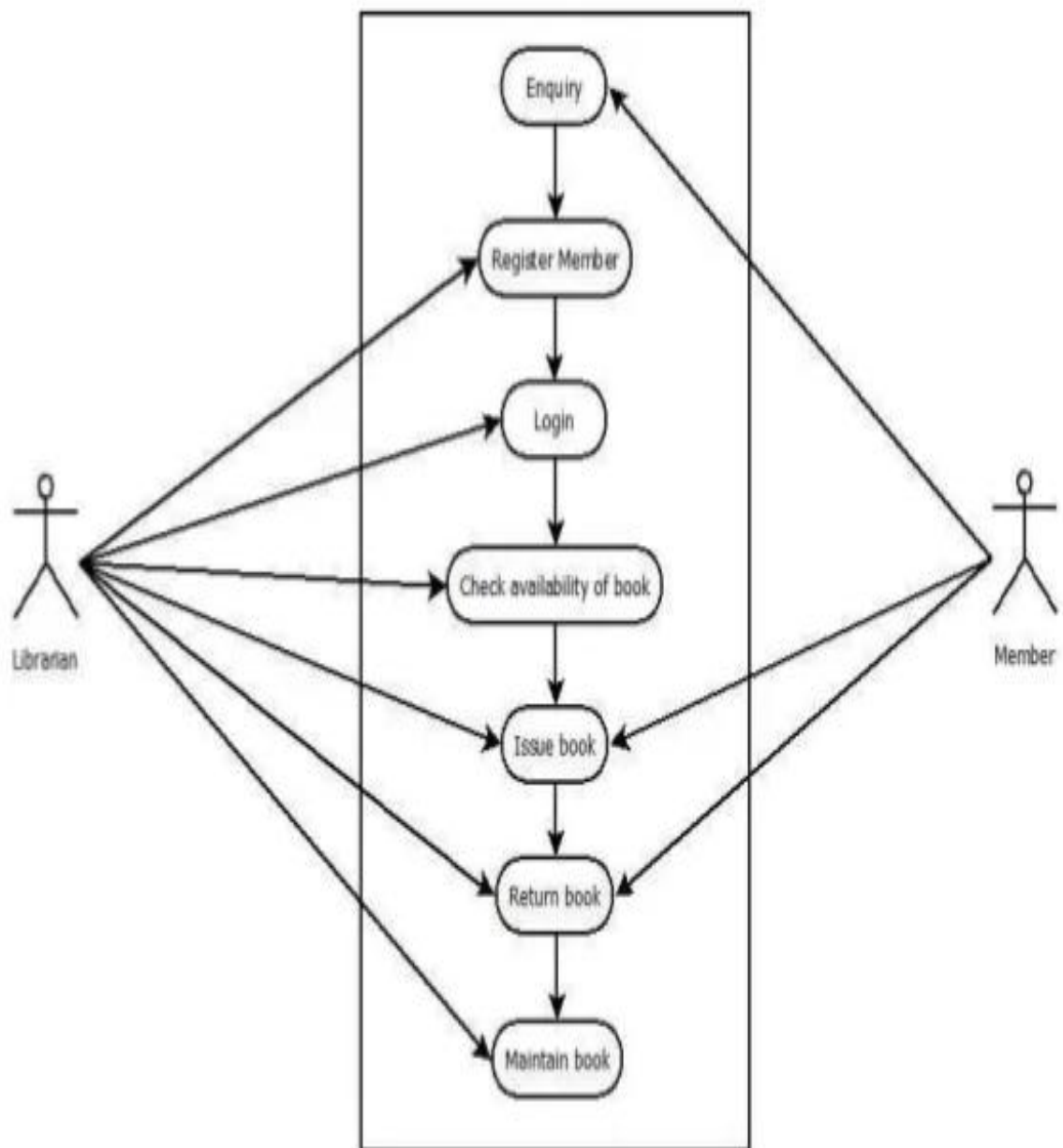
## STEP 5:

Share the Project to the users.



UML Diagram:

# USE CASE DIAGRAM



User stories of a library management system:

<b>TITLE:</b> Librarian	<b>Priority:</b> High	<b>Estimate:</b> 2
<b>User Story-</b> Librarian As a Librarian I want to add new books to the library's collection, Remove books from the library's collection, Search for books by title, author, or subject and Manage patron accounts (create, update, and delete) So that I can Library patrons can check out the latest titles, The library's inventory is up-to-date, Patrons can find the books they are looking for and the library can keep track of who has checked out which books		
<b>Acceptance Criteria:</b> Given that the Librarian wants to add new books to library collection, So that every new book is registered and maintained properly		

<b>TITLE:</b> Patron	<b>Priority:</b> High	<b>Estimate:</b> 2
<b>User Story-</b> Patron As a Patron I want to add Patron Search for books by title, author, or subject, check out books, renew books, return books and Place holds on books that are currently checked out. So that I can find books that match their interests, Take the books home to read, Keep the books longer if needed, allow other patrons to check out the books and reserve the books for when they become available		
<b>Acceptance Criteria:</b> Given that the Patron search for books by title, author, or subject, So that he can manage all the books properly and keep correct record of the books		

# Test plan suite for above user stories using Jira

## Step 1:

Install Zephyr Scale – Test Management for Jira.

The screenshot shows the Jira Software Marketplace interface. The top navigation bar includes 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'Teams', 'Apps', and a 'Create' button. A search bar is on the right. The left sidebar lists 'Apps', 'ATLASSIAN MARKETPLACE', 'Find new apps', 'Manage apps', 'App requests', 'Promotions', and 'OAuth credentials'. The main content area displays the 'Zephyr Scale - Test Management for Jira' app by SmartBear. It features a 4-star rating (307 reviews) and a 'CLOUD FORTIFIED' badge. A 'Success' toast message states: 'Zephyr Scale - Test Management for Jira was added.' Below the app title, there's a section titled 'A scalable, performant test management solution' with a video player showing a person at a screen. The video player has a play button and a 'Platinum Marketplace Partner' badge.

The screenshot shows the Zephyr Scale project page in Jira. The top navigation bar is the same as the previous screenshot. The left sidebar lists 'LBMN Software project', 'PLANNING', 'Roadmap', 'Backlog', 'Board', 'DEVELOPMENT', 'Code', 'Project pages', 'Zephyr Scale' (highlighted), 'Add shortcut', and 'Project settings'. The main content area displays the 'Zephyr Scale' project. A blue banner at the top states: 'Zephyr Scale is currently disabled for the project LBMN (LBMN). Click here to enable'. Below this, there's a section titled 'Getting started with Zephyr Scale' with links for 'Getting Started', 'Documentation', 'Importing Test Cases', and 'Videos'. An illustration of a person standing next to a screen is on the right. The bottom right corner shows a notification bell and a '2' badge.



## Step 2:

### Create Test Suite (case) for your project

The screenshot shows the Jira Software interface for the 'LBMN' project. The left sidebar contains navigation options: PLANNING (Roadmap, Backlog, Board), DEVELOPMENT (Code), Project pages (Zephyr Scale, Add shortcut, Project settings), and a footer note 'You're in a team-managed project'. The main area is titled 'Zephyr Scale' and shows the 'Test Cases' tab. A table of test cases is displayed with columns for P, Key, V, Name, and Status. One test case is listed: LBMN-T1, 1.0, Check the book ID, with a status of APPROVED. The bottom of the page shows '1 - 1 of 1' and 'Archived test cases'.

## Step 3:

### Add the details for your test suite. (case)

The screenshot shows the 'Create Test Case' form in Jira Software. The left sidebar is the same as in Step 2. The main area is titled 'LBMN / Test Cases' and shows the 'Create Test Case' form. The 'Details' tab is active, showing fields for Name, Objective, Precondition, and Details. The 'Details' section includes Status (Approved), Priority (High), Component (None), Owner (Sagar Kori), and Estimated Time (03:00). The 'Labels' section shows a label 'test' and a link to 'Click to add labels'.