

Executive Summary

This project report outlines the development of a **Library Management System (LMS)** built on **Salesforce**, a leading cloud-based platform widely used to manage data and automate business processes.

The primary objective of the system is to simplify and automate routine library operations that are traditionally handled manually. The LMS supports:

- Tracking book availability and issue status in real time
- Streamlining the book issuance and return process
- Automatically calculating fines for overdue returns
- Generating clear, interactive reports and dashboards to monitor library usage

Designed with user-friendliness in mind, the application is intuitive even for users with no prior Salesforce or technical experience. It functions like a smarter, automated version of familiar tools such as Excel or Google Sheets—minimizing manual errors, saving time, and making library management more efficient and organized for librarians, members, and administrators alike.

Table of Contents

1. Problem Statement
2. Goals of the Project
3. System Overview
4. Step-by-Step Implementation
 - 4.1: Custom Objects
 - 4.2: Object Relationships
 - 4.3: Validation Rules
 - 4.4: Automation
5. Roles and Access Management
6. Reports and Dashboards
7. Testing and Validation
8. Business Requirements
9. Tools
10. Testing Results
11. Execution
12. Deployment Plan
13. Release Notes
14. Conclusion

1. Problem Statement

Traditional library systems rely on manual entries in registers or spreadsheets to track books, users, and returns. This leads to:

- Human errors
- Delays in tracking available books
- Difficulty applying fines or generating reports

The goal is to digitize the system to reduce manual effort, improve accuracy, and provide real-time information to librarians and users.

2. Goals of the Project

- Replace manual book issuing and returning with a cloud-based system.
- Enable students and staff to view and request books online.
- Automate book availability and fine calculation.
- Provide secure, role-based access for Admins, Librarians, and Members.

3. System Overview

The LMS(Library Management System) includes the following modules:

- **Book Management:**

Stores book details like title of the book, author, category, and Quantity.

- **Member Management:**

Maintains user records like Members, librarians, and admin.

- **Issue & Return:**

Records each transaction like book issued, return date, and due date.

- **Fines:**

Calculates and tracks penalties for late book returns.

4. Step-by-Step Implementation

4.1: Custom Objects

Created four custom objects in Salesforce:

- **Book:** Book name, Author, Category, Book id, Quantity, Available quantity
- **Member:** Name, Email, ph no., id, active book issues (roll up summary with issue object)
- **Issue:** Issue Date, Issue id, Due days, is active, Member issue ref (master-detail with member object), Book issue ref (lookup with book object), Return Date, Status

- **Fine:** Amount, Status, member fine ref (lookup with member object), Member email, fine issue ref (master-detail with issue object)

Steps to create a custom object :

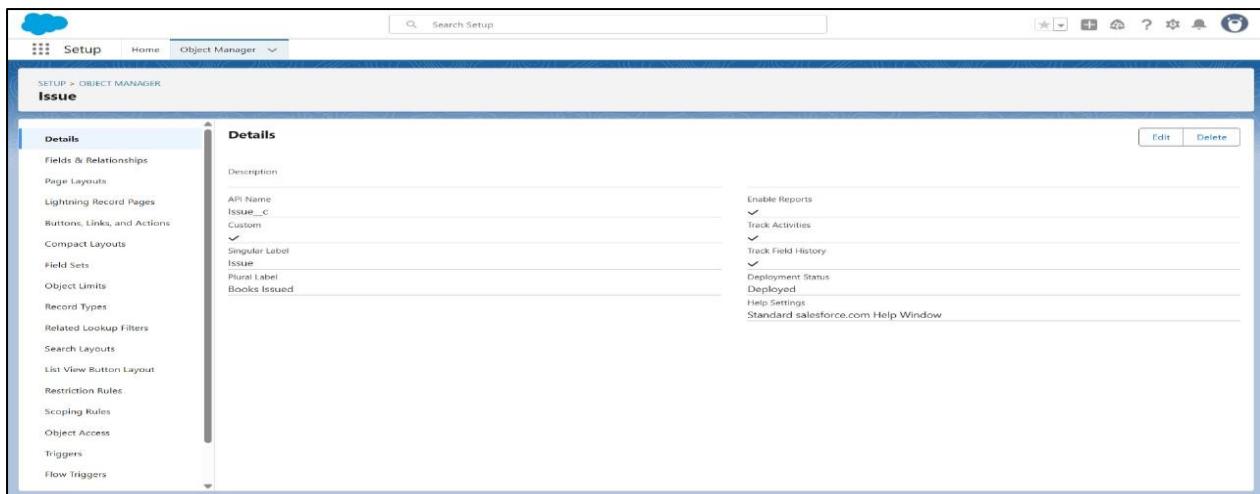
- a. Click on the gear icon on the top right side and click set up.
- b. Select the object manager tab and click the create button on the top right and select new custom object.
- c. Fill the details in the fields and check the boxes which are required and set the FLS and click save. (Your custom is now created)

The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. A new object named 'Book' is being created. The 'Details' section includes fields for API Name ('Book__c'), Singular Label ('Book'), and Plural Label ('Books'). Under 'Enable Reports', 'Track Activities', and 'Track Field History' are checked. Deployment status is set to 'Deployed'. The left sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Lightning Record Pages.

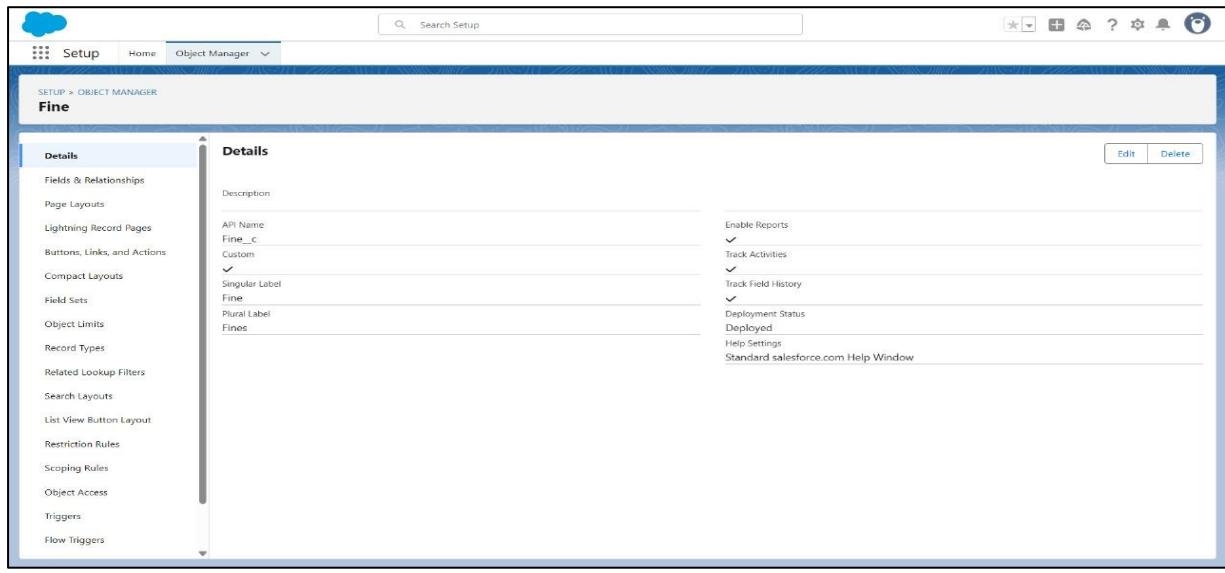
Book Custom Object

The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. A new object named 'Member' is being created. The 'Details' section includes fields for API Name ('Member__c'), Singular Label ('Member'), and Plural Label ('Members'). Under 'Enable Reports', 'Track Activities', and 'Track Field History' are checked. Deployment status is set to 'Deployed'. The left sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Lightning Record Pages.

Member Custom Object



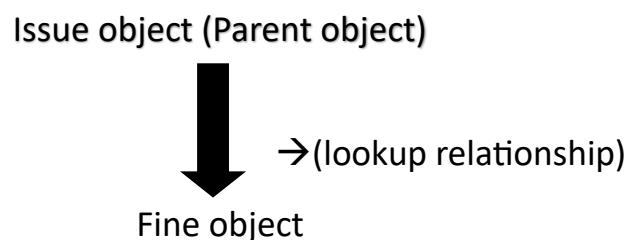
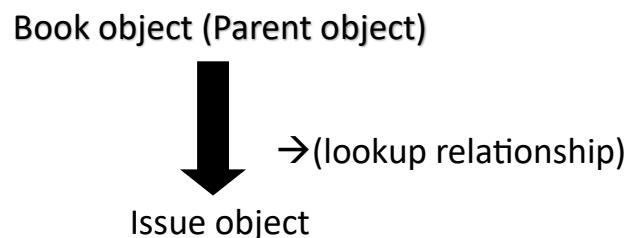
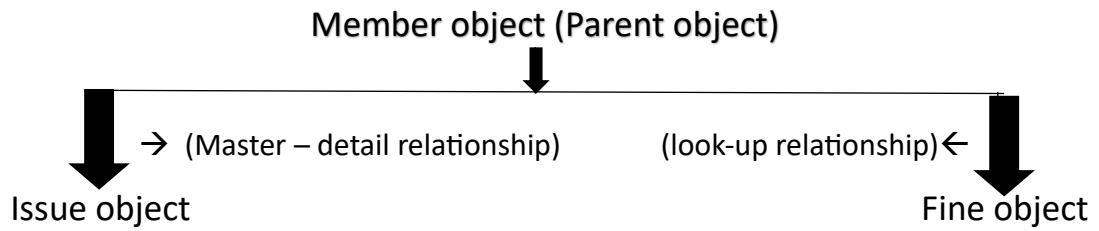
Issue Custom Object



Fine Custom Object

4.2: Object Relationships

- Each **Issue** is linked to a **Book** and a **Member**
- Each **Fine** is linked to a specific **Issue** and **member**
- Book availability is calculated from quantity and transactions
- **Member object** and **issue object** contains the master-detail relationship where member object is the master object (parent) and the issue object is the detail object (child)
- **Book** and **issue** have a lookup relation where book is parent and issue is child
- **Fine** also shares the lookup relationship with the **issue** and **member** objects , where both issue and member are parent objects and fine is a child object



4.3: Validation Rules

To prevent:

- Issuing books when out of stock (available quantity = 0)
- Duplicate issues for the same member-book pair
- Issuing more than allowed limit for a member (member can hold only upto 5 books)
- Return date is not less than the issue date
- Available quantity of a book is greater then the total quantity

Validation Rules					
RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY	
Book_Availability	Book_Issue_Ref	This book is currently unavailable (quantity = 0)	✓	salesforce batch10, 7/13/2025, 7:08 AM	
issuodate_returndate	Returned Date	Returned Date must be greater than Issued Date. dont enter the returned date manually , it will be calculated automatically	✓	salesforce batch10, 7/13/2025, 7:08 AM	
Limit_Maximum_Books_per_Member	Member_Issue_Ref	You cannot issue more than 5 books at a time.please do return a book to take a new book.	✓	salesforce batch10, 7/13/2025, 7:09 AM	

Validation Rules					
RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY	
Validate_Available_Quantity	Available_quantity	Available Quantity cannot be greater than Total Quantity. Pls update either available quantity or quantity	✓	salesforce batch10, 7/13/2025, 2:46 AM	

Validation Rules

Issue Validation Rule					
Rule Name	Book_Availability	Active	✓	Help for this Page ⓘ	
Error Condition Formula	book_Issue_Ref__r.Available_quantity__c <= 0	Error Location	Top of Page		
Error Message	This book is currently unavailable (quantity = 0)	Created By	salesforce batch10, 7/11/2026, 2:33 AM	Modified By	salesforce batch10, 7/11/2026, 2:33 AM

Validation Rule for Book Availability

The screenshot shows the Salesforce Setup interface under the Object Manager for the 'Issue' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Validation Rules. The main content area displays the 'Issue Validation Rule' detail page. The validation rule is named 'Limit_Maximum_Books_per_Member' and has the formula `Member_Issue_Ref__r.Active_Book_Issues__c >= 5`. The error message is 'You cannot issue more than 5 books at a time please do return a book to take a new book.' It is marked as Active and has the error location set to 'Top of Page'. The rule was created by 'salesforce batch10' on 7/11/2025, 3:23 AM, and modified by the same user on 7/11/2025, 9:53 AM.

Validation Rule for Maximum books for person

The screenshot shows the Salesforce Setup interface under the Object Manager for the 'Book' object. The main content area displays the 'Book Validation Rule' detail page. The validation rule is named 'Validate_Available_Quantity' and has the formula `Available_quantity__c > Quantity__c`. The error message is 'Available Quantity cannot be greater than Total Quantity. Pls update either available quantity or quantity'. It is marked as Active and has the error location set to 'Available_quantity'. The rule was created by 'salesforce batch10' on 7/13/2025, 12:33 AM, and modified by the same user on 7/13/2025, 2:46 AM.

Validation Rule for preventing available quantity less than total quantity

4.4: Automation

Using Salesforce automation tools:

- On issuing a book: available quantity decreases, availability updated
- On return: available quantity increases, status marked returned and deletes the particular issue
- Sending an email to the member as a remainder of 2 days, 1 day to return the book to avoid the late fines
- If returned late: Fine record created automatically and send a fine amount mail
- Updating the fine amount until the member pays his fine
- Auto deleting a fine record if the person paid his fine amount

These new features are available only in the Automation Lightning app:

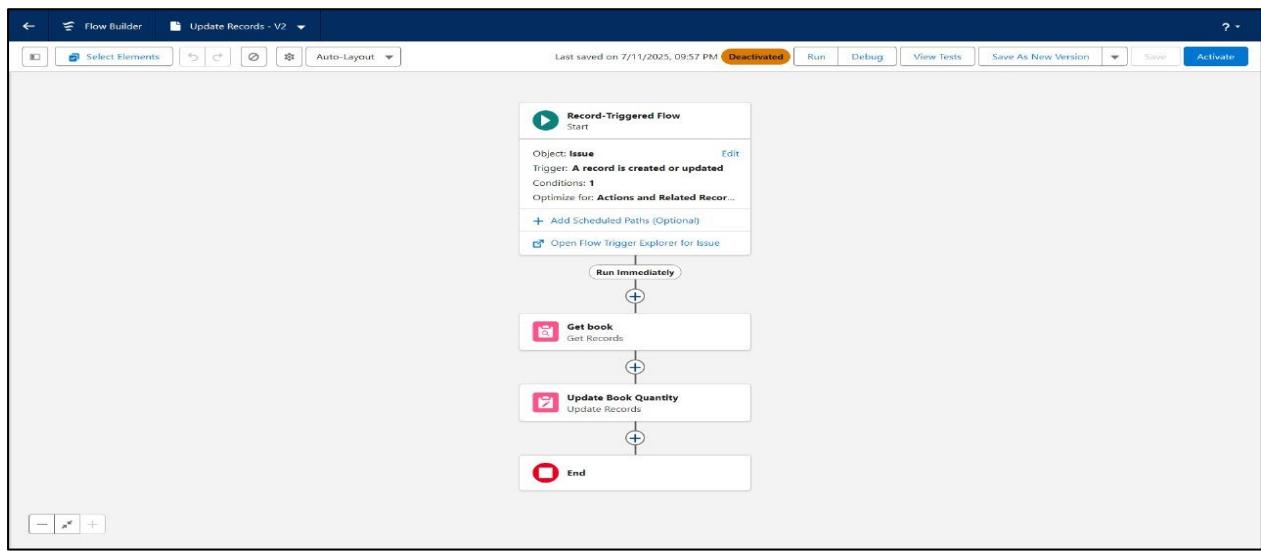
- Search for automations
- Sort your list view with more options
- Organize your automations with categories and subcategories

If you don't see the app in the App Launcher, check that Enable the Automation Lightning App is selected in Process Automation Settings.

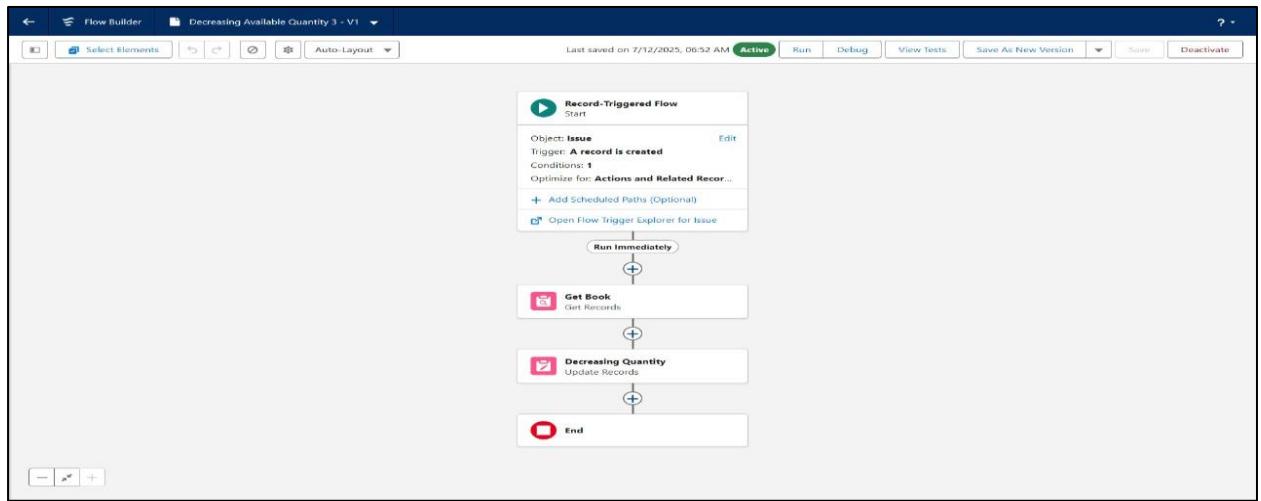
All Flows

Flow Label	Process Type	Actions	Te...	Package State	Pac...	Last Modified By	Last Modified Date
Book Return Reminder	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		salesforce batch10	7/12/2025, 1:48 AM
Fine Day	Autolaunched Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Unmanaged		salesforce batch10	7/12/2025, 12:39 AM
Returned Date	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		salesforce batch10	7/12/2025, 12:31 AM
Deleting Fine Record After Fine Paid	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		salesforce batch10	7/12/2025, 12:31 AM
Increasing Available Quantity when Returned	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		salesforce batch10	7/11/2025, 6:40 PM
Increasing Available Quantity when Returned	Autolaunched Flow	<input type="checkbox"/>	<input type="checkbox"/>	Unmanaged		salesforce batch10	7/11/2025, 6:45 PM
Decreasing Available Quantity when Issued	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		salesforce batch10	7/11/2025, 6:30 PM
Update Records	Autolaunched Flow	<input type="checkbox"/>	<input type="checkbox"/>	Unmanaged		salesforce batch10	7/11/2025, 6:27 AM
Duplicate Issue	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		salesforce batch10	7/11/2025, 8:00 AM
Review Approval Request	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed			
Process Simple Approval	Flow Approval Processes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed			
Approvals Workflow: Process Approval Submission	Screen Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-installed			
Approvals Workflow: Evaluate Approval Requests	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed			
Create Draft Flow Approval Process	Screen Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-installed			
Check Flow API Name	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-installed			

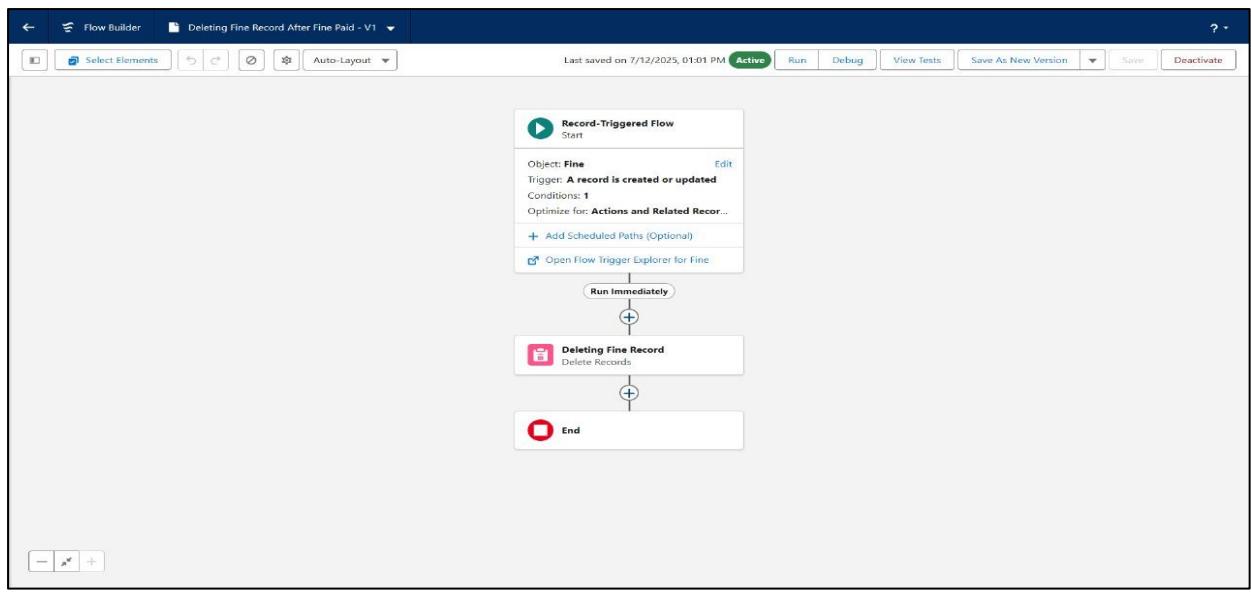
Flows



Flow to Update Records



Flow to decrease Available Quantity



Flow to delete Fine Record after fine paid

5. Roles and Access Management

<u>Profile</u>	<u>Access Rights</u>
• Admin	Full access to all records and settings
• Librarian	Manage books, issues, returns, and fines
• Member	View books, request issues, check fines only

This Roles and Access Management ensures users can only access what is relevant to their role.

6. Reports and Dashboards

- **Overdue Books Report:** Shows books that are not returned on time.
- **Most Issued Books Report:** Shows which books are in high demand.
- **Fine Collection Report:** Displays total fines collected monthly.
- **Library Dashboard**
 - Pie chart of popular categories
 - Bar chart of fines by month
 - List of members with overdue books
 - Total active issues

These insights help administrators make informed decisions.

Reports

library management... Home Members Books Books Issued Fines Dashboards Reports

Dashboard Library Admin Dashboard
Last refreshed 1 day ago. Refresh this dashboard to see the latest data.
As of Jul 15, 2025, 12:48 AM Viewing as salesforce batch10

Refresh Edit Subscribe

Active Issues Report Most Issued Book Categories Report Members with Overdue Books

Active Issues Report

Record Count

Category

Fantasy Story Books Technology Biography Mystery

Member: Name ↑ Issue: Issue Name Book_Issue_Ref: Book Name Return Date

aditya aditya - java (fine -mail) Java 7/13/2025

sudheer sudheer - wonders (day fine) wonder in world 7/7/2025

View Report (Active Issues R... As of Jul 15, 2025, 12:48 AM View Report (Overdue Books Report) As of Jul 15, 2025, 12:48 AM

Fines Collected by Month

Most Issued Books Report

Dashboards

7. Testing and Validation

Test Data: Sample records were created to simulate real operations.

Automation Testing: Flows and triggers were tested to ensure correct stock updates, status changes, and fine generation.

Role Testing: Different test users were created for Admin, Librarian, and Member to verify proper access and UI views.

All validations passed successfully.

8. Business Requirements

Functional Requirements:

1. Users should be able to search and view available books by title, author, or category.
2. Members should be able to request or borrow books (issue request).
3. Librarians should be able to issue and return books, and update book stock(Quantity).
4. The system should track due dates and return status for all borrowed books.
5. The system should automatically calculate fines based on the number of overdue days.
6. Admin should be able to view and manage all users, books, issues, and fines.
7. Reports should be available for:
 - o Overdue books
 - o Fine collections
 - o Most issued books
8. The system should prevent over-issuing (e.g., when book stock = 0).
9. Validation rules should ensure:
 - o A book is not issued multiple times to the same user before return
 - o Return date is not earlier than issue date

Non-Functional Requirements:

1. The system should respond to user actions within 2 seconds.
2. It should be available 24/7 via any modern web browser.
3. Interface should be user-friendly and require minimal training.
4. System should support at least 10 concurrent users without performance issues.

9. Tools Used

Salesforce Developer Edition – Used to build and test the application in a cloud-based development environment.

Flow Builder – Automated processes like updating stock on book issue/return and generating fines.

Report Builder – Created reports and dashboards for overdue books, fine summaries, and book demand analysis.

10. Testing Results

Test Case	Description	Expected Result	Actual Result	Status
TC1	Issue book with quantity > 0	Book issued	Book issued	Pass
TC2	Issue book with quantity = 0	Error shown	Error shown	Pass
TC3	Not returning book after due date	Fine generated	Fine generated	Pass

11. Execution

All components, including custom objects, fields, flows, validation rules, and permissions, were implemented directly in the Salesforce Developer Edition Org.

All development and testing activities were carried out within the same environment, ensuring a simplified process and smooth execution.

12. Deployment Plan

- Ensured all necessary components (custom objects, fields, flows, validation rules, profiles, etc.) were configured correctly in the Developer Org.
- Performed module-wise testing after each configuration step (e.g., book issuing, returning, fine generation).
- Verified user role access through test users for Admin, Librarian, and Member profiles.
- Conducted end-to-end testing to validate the functionality of automation and reporting.

13. Release Notes

Version 1.0 – Initial Release

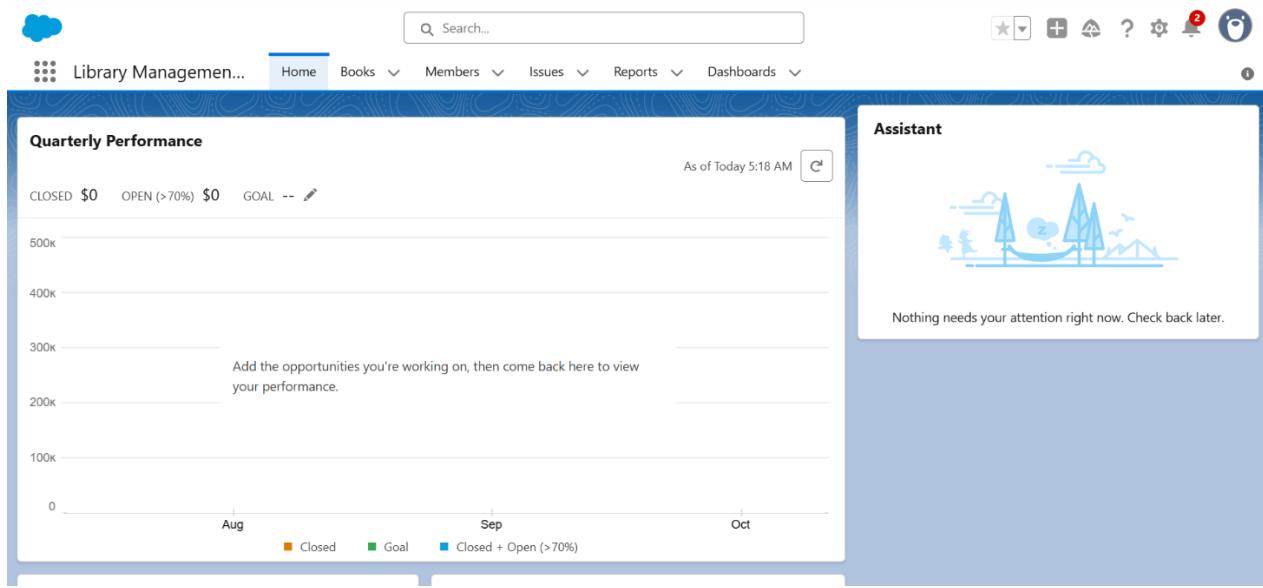
- Custom Objects created: Book, Member, Issue, Fine
- Implemented Flows for automation (stock updates, fine generation)
- Validation Rules added to enforce business logic
- Created Reports & Dashboards for insights
- Defined Profiles and Role-based Access Controls for Admins, Librarians, and Members
- Full functionality tested and verified in the Developer Org

14. Conclusion

The Salesforce-based Library Management System (LMS) effectively replaces manual processes with a modern, automated, cloud-driven solution. Built entirely in the Salesforce Developer Org using declarative tools like Flows, Validation Rules, and Custom Objects, it delivers real-world functionality with minimal code.

Key benefits include:

- **Efficiency** – Reduces manual errors and streamlines librarian workflows.
- **Accessibility** – Allows members to easily search books, track due dates, and view fines.
- **Automation** – Manages stock updates, fine calculations, and email alerts seamlessly.



Library Management System Application