

## Library Book Borrowing System (Education Domain)

 **Goal: Build a Salesforce-based Library Management solution to track books, borrowing, and due dates, automate reminders, and provide actionable reports.**

---

### Phase 1: Problem Understanding & Industry Analysis

 **Goal: Understand what we're building and why.**

#### Requirement Gathering

- Talk to stakeholders (Librarian, Assistant Librarians, Students)
- Example Requirements:
  - Track all books with availability status (Available / Borrowed)
  - Allow librarian to create borrow records for students
  - Restrict students to max 3 borrowed books
  - Send return date reminders automatically
  - Generate reports for overdue books and most borrowed books

#### Stakeholder Analysis

- **Admin:** You (system setup, configuration)
- **Librarian:** Creates & manages borrow records
- **Assistant Librarian:** Helps with returns & book updates
- **Library Manager:** Monitors overdue books & generates reports
- **Students:** Borrow and return books (data tracked in Salesforce)

#### Business Process Mapping

Flow:

Student requests → Librarian checks availability → Borrow Record created → Due date reminder sent → Book returned → Record marked as Returned → Reports generated

#### Industry-specific Use Case Analysis

Libraries face book loss, missed deadlines, and manual tracking issues → need automation to reduce effort, improve compliance, and track popular books.

## **AppExchange Exploration**

Checked Library Apps on AppExchange, decided to build a **custom solution** to match college-specific needs and learn Salesforce end-to-end.

---

## **Phase 2: Org Setup & Configuration**

### **👉 Goal: Prepare Salesforce Environment**

#### **Salesforce Editions**

- Use **Developer Edition Org** (free dev org)

#### **Company Profile Setup**

- Company Info: Library Name, Address
- Timezone: Local
- Currency: INR/USD

#### **Business Hours & Holidays**

- Library Working Hours: 9AM – 6PM
- Holidays: Public/College holidays

#### **Fiscal Year Settings**

- Standard fiscal year (Jan – Dec) → for reporting borrowed books per month

#### **User Setup & Licenses**

- Users: Librarian, Assistant Librarian, Manager
- Assign Standard User licenses

#### **Profiles**

- Librarian Profile: Create/Edit Books & Borrow Records
- Manager Profile: View all borrow records & reports

#### **Roles**

- Manager (top) → Librarian (below) → ensures visibility rolls up

#### **Permission Sets**

- “Report Access” permission set → allows Librarian to view dashboards

## OWD (Org-Wide Defaults)

- Book: Public Read-Only (everyone can see books)
- Borrow Record: Private (only owner/manager can see)

## Sharing Rules

- Share borrow records with manager and assistant librarians if needed

## Login Access Policies

- Restrict librarian logins to library hours

## Dev Org Setup / Sandbox Usage

- Build in developer org (acts like sandbox)
- In real projects, test in sandbox then deploy to production

## Deployment Basics

- Use **Change Sets** to move configuration to production
- 

## Phase 3: Data Modeling & Relationships

👉 **Goal: Build data structure.**

### Standard & Custom Objects

- Standard: Contact (for Students if needed)
- Custom: Book\_\_c, Borrow\_Record\_\_c, Student\_\_c

### Fields

- **Book:** Title, Author, ISBN, Status (Picklist), Category
- **Student:** Name, Roll Number, Email
- **Borrow Record:** Borrow Date, Return Date, Status (Borrowed/Returned)

### Record Types

- Borrow Record → “Short Term Borrow” vs “Long Term Borrow”

### Page Layouts

- Book page → related list of Borrow Records

- Student page → list of borrowed books

### **Compact Layouts**

- Show Book Title, Author, Availability on record cards

### **Schema Builder**

- Visualize relationship: Borrow Record → Lookup to Book and Student

### **Lookup vs Master-Detail**

- Lookup Relationship (Borrow Record doesn't own Book)

### **Junction Objects**

- Not needed unless multiple books per borrow record (can be added for complexity)

### **External Objects**

- Could integrate external database (ex: digital library) in future
- 

## **Phase 4: Process Automation (Admin)**

👉 **Goal: Automate tasks.**

### **Validation Rules**

- Prevent more than 3 active borrow records per student

### **Workflow Rules / Process Builder**

- (Legacy) Auto-update Book Availability → replaced with Flow

### **Approval Process**

- Approval for extending borrow period beyond 15 days

### **Flow Builder**

- Record-Triggered Flow: Send email reminder 2 days before Return Date
- Screen Flow: Librarian Borrow Form

### **Email Alerts**

- Send email to student with book details and due date

### **Field Updates**

- Automatically mark Book as “Available” after return

### **Tasks**

- Create a follow-up task for librarian for overdue books

### **Custom Notifications**

- In-app notification to librarian when a book becomes overdue
- 

## **Phase 5: Apex Programming (Developer)**

👉 **Goal: Add advanced logic.**

### **Classes & Objects**

- BorrowService class → reusable logic for borrow process

### **Apex Triggers**

- On Borrow Insert → Check limit & mark book as unavailable
- On Borrow Update → Update Book availability when returned

### **Trigger Design Pattern**

- Use handler class for clean code

### **SOQL & SOSL**

- SOQL to count active borrow records for student

### **Collections**

- Use List & Map for bulk updates

### **Control Statements**

- IF active borrow count > 3 → throw error

### **Batch Apex**

- Nightly batch to mark overdue books and send reminders

### **Queueable Apex**

- Async process for bulk borrow record updates

### **Scheduled Apex**

- Daily morning → send librarian overdue report

### **Future Methods**

- (Optional) Notify external system like college ERP

### **Exception Handling**

- Show proper error message when borrow limit is exceeded

### **Test Classes**

- Create data, insert borrow records, verify trigger logic (>85% coverage)
- 

## **Phase 6: User Interface Development**

👉 **Goal: Make it user-friendly.**

### **Lightning App Builder**

- Create “Library Management” App

### **Record Pages**

- Book page → shows availability & borrow history

### **Tabs**

- Books, Students, Borrow Records

### **Home Page Layouts**

- Dashboard showing overdue book count & most borrowed books

### **Utility Bar**

- Quick Action → “Create Borrow Record”

### **LWC (Lightning Web Component)**

- Component to search books by title/author
- Show results in a table

### **Apex with LWC**

- Imperative call to fetch available books

### **Events in LWC**

- Child → Parent event when book is selected

### **Wire Adapters**

- Get list of available books dynamically

### **Navigation Service**

- After creating borrow record → navigate to record page
- 

## **Phase 7: Integration & External Access**

 **Goal: Connect with outside systems.**

### **Named Credentials**

- Configure for SMS API (optional)

### **External Services**

- Could connect to external catalog in future

### **Platform Events**

- Publish event when overdue book is returned

### **Change Data Capture**

- Notify external system when borrow record is updated

### **Salesforce Connect**

- Display external e-book data inside Salesforce
- 

## **Phase 8: Data Management & Deployment**

 **Goal: Manage data and move changes.**

### **Data Import Wizard**

- Import sample books and student data

### **Data Loader**

- Import bulk borrow records for testing

### **Duplicate Rules**

- Prevent duplicate ISBN entries

### **Data Export & Backup**

- Weekly export of library data

### **Change Sets**

- Move objects, flows, dashboards from sandbox to production

### **VS Code & SFDX**

- Retrieve and deploy metadata, use version control
- 

## **Phase 9: Reporting, Dashboards & Security Review**

👉 **Goal: Monitor library performance & secure data.**

### **Reports**

- Most Borrowed Books
- Overdue Books
- Student Borrow History

### **Report Types**

- Custom report type: Borrow Record + Book + Student

### **Dashboards**

- Pie chart: Borrowed vs Returned
- Bar chart: Top 5 Students with Most Borrows

### **Dynamic Dashboards**

- Manager sees all data, Librarian sees own data

### **Sharing Settings**

- Private borrow records, public books

### **Field Level Security**

- Hide student email from unauthorized users



## **Audit Trail**

- Track changes to due dates
- 

## **Phase 10: Final Presentation & Demo Day**

👉 **Goal: Deliver like a real project.**

### **Pitch Presentation**

Problem → Solution → Benefits

### **Demo Walkthrough**

Show: Borrow record creation → email reminder → dashboard updates

### **Handoff Documentation**

Admin guide + user guide

### **LinkedIn/Portfolio Project Showcase**

Publish as a Salesforce Project with screenshots

---