**Phase 7: Testing and Validation (Library Book Borrowing System)**

**Goal**

To thoroughly test and validate the functionality, data integrity, and automation processes of the Library Book Borrowing System in Salesforce before final presentation and submission.

**Step 1: Objectives of Testing**

The purpose of this phase is to:

* Verify that all objects, fields, and relationships work as designed.
* Ensure all automations (flows, validation rules) function correctly.
* Validate reports and dashboards for accurate data representation.
* Confirm that user permissions and app navigation are properly configured.

*Screenshot: Library Management App homepage overview before testing.*

**Step 2: Functional Testing**

**Objective**

Ensure every function of the system performs as expected across the main objects: **Books, Students, and Borrow Records**.

**Steps**

1. Open the **Library Management App** in Salesforce.
2. Navigate through all tabs:
   * **Books** → Check book records display Title, Author, ISBN, and Status.
   * **Students** → Verify student records show Roll Number, Department, and Borrow History.
   * **Borrow Records** → Confirm relationships link correctly to both Book and Student.
3. Perform CRUD operations (Create, Read, Update, Delete) on each object to ensure no errors occur.
4. Check related lists:
   * Book detail page → should display linked Borrow Records.
   * Student detail page → should show all Borrowed Books.

**Expected Result:**  
All records and relationships load correctly without error.

*Screenshot: Borrow Record detail page showing Book and Student lookups.*

**Step 3: Validation of Automation (Optional if Phase 4 Skipped)**

**Objective**

Ensure that automation flows and rules (if implemented) trigger as expected.

**Steps**

1. Create a **Borrow Record** with Status = Borrowed.
   * Confirm that the related Book’s Status automatically changes to *Borrowed*.
2. Update the Borrow Record’s Status = Returned.
   * Confirm that the Book’s Status updates to *Available*.
3. (If Validation Rule enabled): Attempt to borrow a 4th book for the same Student.
   * System should display an error message.

**Expected Result:**  
All automation processes execute correctly and update related data without manual input.

*Screenshot: Flow Builder or successful book status update in record view.*

**Step 4: Data Validation**

**Objective**

Ensure data accuracy, consistency, and referential integrity.

**Steps**

1. Verify every **Borrow Record** has valid **Book** and **Student** references.
2. Confirm that all Picklist values (Status, Category, Department) appear correctly in dropdowns.
3. Check for duplicate records using Salesforce’s Duplicate Management tools.
4. Export sample records to CSV via **Data Export** and validate consistency.

**Expected Result:**  
All data is accurate, relationships are intact, and no duplicate entries exist.

*Screenshot: Data Export preview or Book list view.*

**Step 5: Reports and Dashboards Validation**

**Objective**

Ensure the analytical components from Phase 5 correctly display and summarize system data.

**Steps**

1. Open the **Library Overview Dashboard**.
2. Review each component:
   * **Borrowed vs Returned** → Matches number of Borrow Records.
   * **Top Borrowed Books** → Displays accurate top 5.
   * **Borrow Count per Student** → Reflects actual data.
   * **Overdue Borrow Records** → Matches Borrow Records with past Return Dates.
3. Cross-check numbers with underlying records for accuracy.
4. Refresh dashboard to confirm data updates dynamically.

**Expected Result:**  
Reports and Dashboard metrics match the live Borrow Record data.

*Screenshot: Library Overview Dashboard displaying verified charts.*

**Step 6: User Access Testing**

**Objective**

Ensure correct permissions and data visibility for all user profiles and roles.

**Steps**

1. Log in as each user type:
   * **Library Manager (Admin)** → Full access to all data and setup.
   * **Librarian** → Read/Edit access on Books and Borrow Records.
   * **Assistant Librarian** → Read-only access to Student and Book data.
2. Verify each user can:
   * Access assigned tabs.
   * Create or edit records based on permission level.
   * View only allowed data.

**Expected Result:**  
Each user can perform their designated tasks without unauthorized access.

*Screenshot: Profile permission settings or user login test view.*

**Step 7: Error Handling and Fixes**

**Objective**

Identify and correct any issues found during testing.

**Common Issues and Fixes**

| **Issue** | **Cause** | **Solution** |
| --- | --- | --- |
| Field not visible | Field-level security not set | Edit Profile/Permission Set → Enable field visibility |
| Flow not triggering | Record-trigger condition mismatch | Edit Flow → Adjust trigger condition and re-activate |
| Report data missing | Filters too narrow | Adjust report filters (e.g., Date Range = All Time) |
| Access denied | Profile lacks object permission | Update profile to include Read/Edit permission |

**Expected Result:**  
All issues identified are resolved before final deployment.

*Screenshot: Example of corrected flow or permission set screen.*

**Step 8: Final Validation Summary**

**Objective**

Summarize test outcomes to ensure the app meets requirements.

**Summary Table**

| **Test Area** | **Status** | **Notes** |
| --- | --- | --- |
| Object Functionality | ✅ Passed | CRUD and relationships working |
| Automation Flows | ✅ Passed / Skipped | Depends on Phase 4 implementation |
| Data Integrity | ✅ Passed | No duplicates or broken links |
| Reports/Dashboards | ✅ Passed | Data accuracy verified |
| Permissions & Access | ✅ Passed | Correct per profile |

**Step 9: Documentation for Evaluation**

1. Capture screenshots of successful test results.
2. Prepare a short Testing Summary Report (PDF) including:
   * Validation checklist.
   * Screenshot evidence.
   * Summary of fixes applied.
3. Attach this as part of your final project documentation.

*Screenshot: Testing Summary Report sample page.*

**Phase 7 Outcome**

* Verified that all functional, data, and automation components work as expected.
* Validated reports and dashboards for accuracy.
* Confirmed user permissions and accessibility.
* Ensured app is stable and ready for deployment and presenta