User Stories – LMS

# US-01: Search Books

Description:

As a Student, I want to search for books by title or author so that I can quickly locate study resources.

Acceptance Criteria:

• Given I am logged in, when I enter a keyword (title/author/ISBN/category), then the system must return results within 3 seconds.  
• Results must display: Title, Author, ISBN, Availability.  
• If no matches are found, system displays 'No results found'.  
• Results can be filtered by category and availability.

Associated Tasks:

* Design search API query (developer)
* Connect UI search box to backend service
* Validate response time < 3s (tester)
* Verify no empty/null records appear (tester)

# US-02: View Availability

Description:

As a Student, I want to see if a book is available or checked out, so I don’t waste time going to the library unnecessarily.

Acceptance Criteria:

• Book search results must include availability status: 'Available', 'Checked Out', or 'On Hold'.  
• If checked out, the system must show expected return date.  
• Status updates in real time after borrowing/returning.

Associated Tasks:

* Update DB schema with availability column (developer)
* Add availability status in UI cards (frontend)
* Write test cases for status transitions (tester)

# US-03: Borrow Book

Description:

As a Student, I want to borrow a book with a clear due date, so I know when to return it.

Acceptance Criteria:

• Students can borrow max 3 books at once.  
• When borrowing, the system sets due date = Today + 14 days.  
• Confirmation message/email includes book details and due date.  
• Borrow button must be disabled if student already has 3 active loans.

Associated Tasks:

* Implement borrowing logic in Loan table (developer)
* Add confirmation modal with details (frontend)
* Create email notification template (developer)
* Write UAT cases for max-limit validation (tester)

# US-04: Pay Fine Online

Description:

As a Student, I want to pay fines online, so I don’t need to wait in line at the counter.

Acceptance Criteria:

• System must display outstanding fines at login.  
• Payment via credit card or online gateway.  
• After successful payment, fine is marked 'Paid' and finance is notified.  
• Failed transactions must not update fine status.

Associated Tasks:

* Integrate payment gateway API (developer)
* Build payment UI form (frontend)
* Confirm success/failure callback handling (developer)
* Test positive/negative payment flows (tester)

# US-05: Add Books to Catalog

Description:

As a Librarian, I want to add new books to the catalog, so students always have updated information.

Acceptance Criteria:

• All required fields (Title, Author, ISBN) must be validated.  
• ISBN must be unique.  
• New book status = 'Available' by default.

Associated Tasks:

* Add validation rules in backend service (developer)
* UI form for adding/editing books (frontend)
* Negative tests for duplicate ISBN (tester)

# US-06: Remove Books

Description:

As a Librarian, I want to remove outdated or damaged books, so the catalog reflects reality.

Acceptance Criteria:

• Only librarians with admin role can remove a book.  
• Removed books are flagged 'Inactive' (not deleted).  
• System must log all removals with timestamp and librarian ID.

Associated Tasks:

* Implement soft-delete in DB (developer)
* Add audit logging (developer)
* Test restricted access (tester)

# US-07: Generate Overdue Reports

Description:

As a Librarian, I want to generate overdue reports, so I can contact students.

Acceptance Criteria:

• Report must list: Student ID, Book ID, Days Overdue, Fine Amount.  
• Report exportable to PDF/Excel.  
• Generation time must be <10s for up to 10,000 records.

Associated Tasks:

* Write SQL queries for overdue loans (developer)
* Add export-to-PDF/Excel button (frontend)
* Test large dataset performance (tester)

# US-08: Borrowing Trends Report

Description:

As University Management, I want to view monthly borrowing trends, so I can plan acquisitions.

Acceptance Criteria:

• Report must group by month and category.  
• Display total loans per month for last 12 months.  
• Graphical chart view available in UI.

Associated Tasks:

* Write SQL aggregation for monthly data (developer)
* Build chart visualization (frontend)
* Validate aggregation logic with test dataset (tester)

# US-09: Most Borrowed Categories

Description:

As University Management, I want to know the most borrowed categories, so I can allocate budgets.

Acceptance Criteria:

• Report must list Top 5 categories by loan count.  
• Data must be filterable by semester/year.  
• Results exportable to Excel.

Associated Tasks:

* Create aggregation query by category (developer)
* Add filter UI elements (frontend)
* Test category ranking accuracy (tester)