



Riyadh University of Arts

Requirements Specification Document

Library Management System (LIB.MS)

November 2025

Contents

1	Introduction.....	4
1.1	Purpose of the Document.....	4
1.2	Project Background	4
1.3	Objectives	4
1.4	Stakeholders	4
2	Project Scope	5
2.1	In-Scope Deliverables	5
2.2	Out-of-Scope	5
3	Functional Requirements	6
3.1	Core LIB.MS Modules	6
3.1.1	Cataloging	6
3.1.2	Circulation	6
3.1.3	Patron Management	6
3.1.4	Acquisitions	6
3.1.5	Online Public Access Catalog (OPAC) / Discovery Layer	6
3.2	Advanced Modules	7
3.2.1	Electronic Resource Management.....	7
3.2.2	Course Reserves	7
3.2.3	Institutional Repository	7
3.2.4	Reporting & Analytics Module	7
4	Product Requirements.....	8
4.1	User Personas and Needs	8
4.1.1	Students	8
4.1.2	Faculty	8
4.1.3	Librarians	8
4.1.4	IT/System Administrators	8
4.1.5	Finance and Procurement	8
5	Sample User Stories	9
5.1	Student.....	9
5.2	Faculty	9
5.3	Librarian	9
5.4	Admin/IT	9
6	Example Use Cases.....	10
6.1	Use Case: Borrowing a Book	10
6.2	Use Case: Faculty Course Reserve Request	10
6.3	Use Case: Repository Submission.....	10
7	Technical Requirements.....	11
7.1	Hosting Environment	11
7.2	Data Residency	11
7.3	Security and Compliance	11
7.4	Language Support	11
7.5	APIs & Extensibility	11

7.6	Mobile Application	11
7.7	Self-Service Kiosks.....	11
7.8	User Notifications.....	11
8	Integration Requirements	12
8.1	Student Information System (SIS) Integration.....	12
8.2	ERP Integration	12
8.3	SSO Integration	12
8.4	Learning Management System (LMS) Integration	12
9	Non-Functional Requirements	13
9.1	Performance and Availability	13
9.2	Scalability and Reliability	13
9.3	Data Migration	13
9.4	Support and Training.....	13
10	Sample Requirements Traceability Matrix (RTM)	15
10.1	RTM Table	15
11	System Architecture Diagram (Conceptual).....	16
12	Sample Workflows	17
12.1	Circulation Workflow	17
12.2	Acquisitions Workflow.....	17
12.3	ERM Workflow	17
12.4	Institutional Repository Workflow.....	17

1 Introduction

1.1 Purpose of the Document

This Requirement Specifications Document defines the expanded product, functional, and technical specifications for the new Library Management System (LIB.MS) for Riyadh University of Arts (RUA). It builds upon the baseline requirements provided and transforms them into a more comprehensive specifications document suitable for RFP preparation, vendor evaluation, architectural planning, and implementation governance.

1.2 Project Background

RUA aims to create a holistic, digital-first library ecosystem supporting its diverse community of learners, artists, faculty researchers, and administrative staff. The LIB.MS will serve as the digital backbone for all library services across cataloging, circulation, acquisitions, electronic resource management, institutional repository functions, and discovery interfaces.

1.3 Objectives

- Deliver a future-ready, cloud-native LIB.MS aligned with RUA's digital transformation.
- Provide seamless, bilingual (Arabic/English) user experiences across web and mobile.
- Support RUA's creative ecosystem with multimedia repository capabilities.
- Reduce operational overhead through workflow automation and analytics.
- Enable system-wide integration across Student Information System (SIS), Enterprise Resource Platform (ERP), Single Sign-on (SSO), and Learning Management System (LMS).
- Ensure compliance with Personal Data Protection Law (PDPL) and internal IT governance.

1.4 Stakeholders

- Library Services Department (Primary Owner)
- IT & Digital Transformation Office
- Academic Affairs & Faculty
- Undergraduate & Graduate Students
- Finance & Procurement
- Legal & Compliance
- Executive Leadership

2 Project Scope

2.1 In-Scope Deliverables

- Cloud-hosted SaaS Library Management Platform
- Core functional and advanced modules (Cataloging, Circulation, Acquisitions, Patron Management, OPAC, ERM, Course Reserves, Institutional Repository)
- Identity, academic, and financial system integrations
- Mobile apps (iOS/Android)
- Notifications, self-service, and discovery features
- Data migration from legacy sources
- Training, documentation, and deployment support

2.2 Out-of-Scope

- On-premises deployment
- Custom-built library software
- Physical hardware procurement (kiosks, scanners, beacons)
- Non-library digital content systems unrelated to RUA's repository

3 Functional Requirements

3.1 Core LIB.MS Modules

3.1.1 Cataloging

- Must support the creation, import, and editing of bibliographic records using the MARC 21 standard.
- Must support batch import and export of MARC records.
- Must allow for the creation of item records for physical assets (books, journals, etc.) and the linking of digital assets.
- Must support authority control for names, subjects, and series.

3.1.2 Circulation

- Must manage all circulation transactions: check-out, check-in, renewal, and holds (requests).
- Must allow for the configuration of flexible loan policies based on user type (e.g., undergraduate, faculty, staff) and item type (e.g., general collection, short-term loan).
- Must automatically calculate and manage fines for overdue items.
- Must provide staff with a clear, real-time view of an item's status and a patron's borrowing history.

3.1.3 Patron Management

- Must maintain a comprehensive database of all library users.
- Must support the automatic creation and updating of patron records via integration with the university's Student Information System (SIS) (see Section 5.1).
- Must allow staff to manage patron accounts, including updating contact information and managing borrowing privileges.

3.1.4 Acquisitions

- Must manage the entire lifecycle of ordering new materials, from creating purchase orders to receiving items.
- Must support budget management, allowing librarians to track spending against different fund codes.
- Must integrate with the university's Enterprise Resource Planning (ERP) system for financial processing.

3.1.5 Online Public Access Catalog (OPAC) / Discovery Layer

- Must provide a modern, intuitive, and user-friendly web interface for students and faculty to search the library's entire collection (physical and digital).
- Search functionality must include features such as faceted search (filtering by author, date, format, etc.), spell-check, "did you mean?" suggestions, and the display of book cover art.
- Must allow authenticated users to log in to view their account, renew items, and place holds.

3.2 Advanced Modules

3.2.1 Electronic Resource Management

- The LIB.MS must include an integrated ERM module to manage the library's collection of digital subscriptions (databases, e-journals, e-books).
- Must manage license terms, access rights, and subscription renewal dates.
- Must provide a central knowledge base to manage the library's electronic holdings and ensure they are discoverable in the OPAC.

3.2.2 Course Reserves

- The system must provide functionality for managing course reserves.
- Faculty should be able to request that specific physical or digital items be placed on reserve for their courses.
- Students should be able to easily search for and access materials for the courses they are enrolled in.

3.2.3 Institutional Repository

- The system must offer a module or a seamlessly integrated solution for creating and managing an institutional repository.
- This will be used to host, preserve, and provide access to the scholarly and creative output of the RUA community, such as faculty research, student theses, and digital art projects.

3.2.4 Reporting & Analytics Module

- The system must provide built-in dashboards for circulation trends, catalog usage, patron activity, fines, and popular items.
- Provide analytics for ERM usage, including database access, click-throughs, and license compliance.
- Enable OPAC/search analytics, such as top queries, zero-result searches, item-level engagement, and discovery-to-checkout conversion.
- Allow exporting of reports in multiple formats (CSV, Excel, PDF).
- Offer automated scheduled reports for library leadership and academic departments.
- Provide API access for integration with RUA's BI and institutional analytics platforms.

4 Product Requirements

4.1 User Personas and Needs

4.1.1 Students

- Search and access digital and physical resources.
- Manage loans, renewals, fines.
- Access course reserves.
- Receive notifications and updates.
- Use mobile devices for self-service.

4.1.2 Faculty

- Request course reserves.
- Deposit research or creative outputs to the repository.
- Access subscription-based digital resources.
- Monitor item availability for assignments.

4.1.3 Librarians

- Perform cataloging and metadata management.
- Manage circulation operations.
- Track and manage acquisitions and budgets.
- Maintain digital resource licenses.
- Support students and faculty through helpdesk tools

4.1.4 IT/System Administrators

- Manage identities, access, and SSO.
- Monitor system uptime and APIs.
- Oversee integrations.

4.1.5 Finance and Procurement

- Manage budgets, POs, and invoices.
- Ensure alignment between LIB.MS acquisitions and ERP.

5 Sample User Stories

5.1 Student

- As a student, I want to search the library catalog so I can find books and digital resources quickly.
- As a student, I want to place holds so I can reserve a book that is currently checked out.
- As a student, I want to access digital journals using my university login, so I don't need multiple accounts.
- As a student, I want to receive reminders when my items are due, so I avoid fines.
- As a student, I want to access course readings directly from Blackboard.

5.2 Faculty

- As a faculty member, I want to request course reserves so students can access assigned materials.
- As a faculty member, I want to deposit my research into the repository for preservation.
- As a faculty member, I want to view usage analytics for course materials.

5.3 Librarian

- As a librarian, I want to import MARC records in bulk to reduce manual work.
- As a librarian, I want to configure loan rules per user group so borrowing policies are consistent.
- As a librarian, I want to track budgets for multiple fund codes for transparency.

5.4 Admin/IT

- As an IT admin, I want SSO enabled so users can log in with their RUA credentials.
- As an IT admin, I want API logs and monitoring tools for troubleshooting.

6 Example Use Cases

6.1 Use Case: Borrowing a Book

Actor: Student

Precondition: Student has an active account synced from SIS.

Flow:

1. Student searches OPAC.
2. Student locates item and checks availability.
3. Student visits circulation desk or kiosk.
4. System validates loan rules.
5. Book is issued and notification sent.

6.2 Use Case: Faculty Course Reserve Request

Actor: Faculty

Flow:

1. Faculty submits reserve request through LIB.MS.
2. Librarian reviews and approves.
3. Item is flagged as "Course Reserve."
4. Items appear for students enrolled in the course.

6.3 Use Case: Repository Submission

Actor: Student/Faculty

Flow:

1. User logs into repository module.
2. Uploads document/media.
3. Metadata entered.
4. Librarian reviews and publishes.

7 Technical Requirements

7.1 Hosting Environment

- The LIB.MS must be a fully cloud-hosted Software-as-a-Service (SaaS) solution. No on-premises hardware or software installation will be considered. The vendor is responsible for all system maintenance, updates, and backups.

7.2 Data Residency

- The solution must comply with all data residency regulations of the Kingdom of Saudi Arabia, including the Personal Data Protection Law (PDPL). The vendor must guarantee that all patron data and university-owned data is hosted on servers located within KSA.

7.3 Security and Compliance

- The system must adhere to modern security standards, including data encryption at rest and in transit. It must support role-based access control (RBAC) to ensure staff can only access functions relevant to their duties.

7.4 Language Support

- The system must provide full, native support for both Arabic and English in all user-facing and staff-facing interfaces. This includes right-to-left (RTL) text rendering and proper sorting and searching of Arabic metadata.

7.5 APIs & Extensibility

- The system must provide a well-documented set of APIs to allow for future custom integrations or extensions as the university's needs evolve.

7.6 Mobile Application

- The vendor must provide a native mobile app for iOS and Android that allows users to search the catalog, manage their account, place holds, and access digital resources.

7.7 Self-Service Kiosks

- The system must support integration with self-service hardware. Specifically, it must have a standard protocol (e.g., SIP2) to communicate with self-checkout kiosks.

7.8 User Notifications

- The system must be able to automatically send email and/or push notifications to users for key events, such as hold pickup notices, due date reminders, and overdue alerts.

8 Integration Requirements

8.1 Student Information System (SIS) Integration

- The LIB.MS must integrate with the university's SIS (e-Register) via a robust API.
- This integration must automatically synchronize patron data, ensuring that new students/faculty are added, and departing ones are deactivated, in the LIB.MS in near real-time.

8.2 ERP Integration

- The LIB.MS must integrate with the university's ERP (Oracle) via API.
- This integration must support the automated transfer of financial data from the Acquisitions module to the ERP for budget tracking and invoice processing.

8.3 SSO Integration

- The LIB.MS must support authentication via the university's central identity provider using standard protocols (e.g., SAML 2.0 or OpenID Connect). This will allow all users to log in with their standard RUA credentials.

8.4 Learning Management System (LMS) Integration

- The system must offer an integration or plugin (e.g., LTI) for the university's Learning Management System (e.g. Blackboard). This would allow faculty to embed library resources and search functionality directly into their online courses.

9 Non-Functional Requirements

9.1 Performance and Availability

- Vendor must guarantee a minimum uptime of 99.9 percent, with clear SLA penalties for violations.
- All mission-critical actions (search, check-out, check-in, renewals) must complete in under 2 seconds under normal load.
- All user-facing pages, especially OPAC search results, must achieve a Google PageSpeed Insights score of 85 or higher on both mobile and desktop.
- System must auto-scale to handle peak load periods such as exam season, registration periods, and semester openings.
- Background processes (batch jobs, indexing, large imports) must not degrade live user performance.

9.2 Scalability and Reliability

- System must support seamless vertical and horizontal scaling to accommodate seasonal spikes.
- Application components must be containerized or designed for distributed deployment.
- Indexing, search services, and repository services must scale independently without impacting core circulation.
- Failover mechanisms must be in place to avoid service interruption.
- System must include automated health checks and self-recovery mechanisms

9.3 Data Migration

- Vendor must provide full migration services covering bibliographic records, item records, patron data, loan history, fines, acquisition budgets, and digital asset metadata.
- Must support ingestion from MARC, CSV, Excel, and legacy proprietary formats.
- Vendor must provide cleansing, deduplication, and validation tools to ensure data accuracy.
- A test migration cycle must be conducted before production migration.
- Vendor must provide a migration plan including timelines, responsibilities, risk mitigation, and rollback strategy.

9.4 Support and Training

- Vendor must provide multilingual support (Arabic/English) with SLAs defining response and resolution times.
- Critical issues must be addressed within 4 hours; non-critical within 24–48 hours.
- Training must include librarian-focused modules for cataloging, circulation, acquisitions, ERM, and repository.
- Admin training must cover SSO, APIs, integrations, reporting, and role management.
- Vendor must provide ongoing access to user manuals, video tutorials, and updated documentation.
- Optional advanced training sessions must be available for new feature releases or staffing changes.

9.5 Accessibility Requirements

- The system must fully comply with KSA Digital Government Authority (DGA) accessibility standards, aligned with WCAG 2.1 Level AA.
- All public-facing and staff interfaces must support keyboard navigation, screen readers, and non-visual access technologies.

- OPAC, mobile app, and staff tools must use proper semantic HTML, ARIA labels, and consistent heading structures.
- Color contrast ratios must meet WCAG guidelines to ensure readability for visually impaired users.
- Multimedia content in the Institutional Repository must support captions, transcripts, and alternative text where applicable.
- Accessibility testing must be included in UAT, including screen reader testing (NVDA/VoiceOver) and automated audits.

10 Sample Requirements Traceability Matrix (RTM)

The RTM ensures every requirement is mapped to user stories, use cases, and system components.

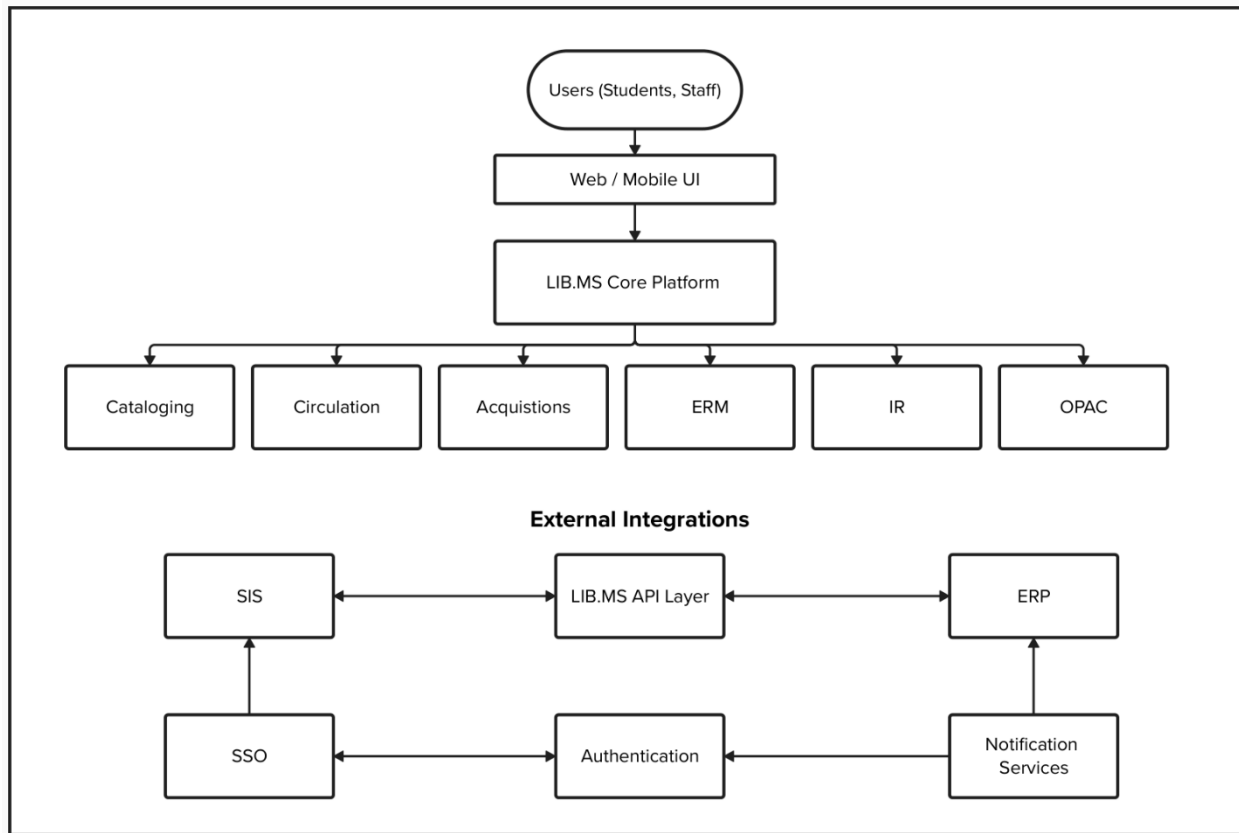
10.1 RTM Table

Requirement ID	Requirement Description	Type	Source	User Story Link	Use Case Link	Module	Priority
FR-001	Search library catalog	Functional	Library Dept	STU-01	UC-01	OPAC	Must Have
FR-002	Place holds/requests	Functional	Students	STU-02	UC-01	Circulation	Must Have
FR-003	MARC21 cataloging	Functional	Librarians	LIB-01	UC-04	Cataloging	Must Have
FR-004	SIS integration for patron sync	Integration	IT	ADM-02	UC-05	Patron Mgmt.	Must Have
FR-005	ERP integration for acquisitions	Integration	Finance	LIB-03	UC-06	Acquisitions	Must Have
FR-006	Course reserves	Functional	Faculty	FAC-01	UC-02	Reserves	Should Have
FR-007	Repository submission	Functional	Faculty/Students	FAC-02	UC-03	IR	Should Have
NFR-001	99.9 percent uptime	Non-Functional	IT Governance	—	—	Platform	Must Have

(More rows can be added as the requirements evolve.)

11 System Architecture Diagram (Conceptual)

A high-level architecture representation of how the LIB.MS integrates within RUA's digital ecosystem.



12 Sample Workflows

Below are process workflows for core modules.

12.1 Circulation Workflow

Student → Search OPAC → Check Availability → Visit Desk/Kiosk → LIB.MS Validates Rules → Item Issued → Notification Sent

12.2 Acquisitions Workflow

Librarian Creates PO → LIB.MS Sends Financial Data → ERP Validates Budget → Vendor Delivery → LIB.MS Receives Items → Cataloging Links Records

12.3 ERM Workflow

Library Acquires e-Resource → License Terms Added → Access Rights Configured → Knowledge Base Updated → OPAC Displays Digital Resource → Usage Analytics Generated

12.4 Institutional Repository Workflow

User Uploads Content → Metadata Entered → Librarian Reviews → Repository Stores Asset → OPAC/Discovery Index Updated → Public Access Enabled

