**Functional Analysis Document**

Project: University Library Management System

Author: Functional Analyst Demo (Ramiro Funes)

Date: 2023-05-24

Version: 1.0

# Table of Contents

1. Business Context

2. Business Objectives

3. Stakeholders

4. High-Level Requirements

5. Use Cases

6. User Stories

7. Non-Functional Requirements

8. Data Model

9. Acceptance Criteria

10. Risks & Assumptions

11. Glossary

# Business Context

The University currently manages library operations manually with spreadsheets and paper logs. This causes long waiting times, limited visibility, and inconsistent reporting. The new system must provide a modern, web-based platform to manage catalog, borrowing, returns, and reporting.

# Business Objectives

• Reduce average checkout time under 2 minutes.  
• Provide real-time availability.  
• Automate overdue fines.  
• Weekly/monthly reporting.  
• Improve student satisfaction (>80% after 6 months).

# Stakeholders

Librarians: Manage catalog and daily tasks.  
Students: Borrowers and main end-users.  
IT Department: Maintains system, integrations.  
Finance: Tracks fines and payments.  
Management: Consumes reports for decision-making.

# High-Level Requirements

1. Search books by multiple criteria.  
2. Borrow/return with due date.  
3. Librarian catalog management.  
4. Automatic fine calculation.  
5. Reports (usage, fines).  
6. Student directory integration.  
7. Responsive design.

# Use Cases

UC-01: Search Books  
UC-02: Borrow Book  
UC-03: Return Book  
UC-04: Pay Fine  
UC-05: Manage Catalog  
UC-06: Generate Reports

# User Stories

As a Student, I want to search books to find resources quickly.  
As a Student, I want to borrow books with clear due dates.  
As a Student, I want to pay fines online.  
As a Librarian, I want to manage the catalog.  
As a Librarian, I want to generate reports.  
As Management, I want to see most borrowed books.

# Non-Functional Requirements

• Performance: results under 3s.  
• Scalability: 500 concurrent users.  
• Security: SSO.  
• Usability: WCAG 2.1 AA.  
• Availability: 99.5% uptime.

# Data Model (simplified)

Entities: Book, Student, Loan, Payment.  
Relationships: Student borrows Book → Loan; Payment linked to Student.

# Acceptance Criteria

• Borrow updates availability.  
• Fine = $1/day overdue.  
• Max 3 loans per student.  
• Reports exportable to PDF/Excel.

# Risks & Assumptions

• Risk: Student directory API limited.  
• Risk: Payment gateway delays.  
• Assumption: Hosting and SSL provided.

# Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Loan | Temporary lending of a book to a student. |
| Overdue Fine | Penalty for late return. |
| Catalog | Database of all managed books. |