

**TOBB ETU**

**Economy & Technology University**

**BIL 481**

**Software Requirements Specification (SRS)**

**Agile-ish**

**Kaan BEHZETOĞLU**

**Mehmet Fatih AKAY**

**Duygu AKMAN**

**Ahmet Taha ÖZCAN**

### ***CONTENTS***

### ****1. Introduction****

1.1 Purpose  
1.2 Scope  
1.3 Definitions, Acronyms, and Abbreviations  
1.4 References  
1.5 Overview

### ****2. Overall Description****

2.1 Product Perspective  
2.2 Product Functions  
2.3 User Characteristics  
2.4 Assumptions and Dependencies

### ****3. Specific Requirements****

3.1 Functional Requirements  
3.2 External Interface Requirements  
3.3 Performance Requirements  
3.4 Design Constraints  
3.5 Other Requirements

### ****4. Appendices****

### ****5. References****

# 1. Introduction

## Purpose

## The purpose of this document is to define the **functional and non-functional requirements** of the **Libman Project** in detail. This document will serve as a **reference point** for both the **project team** (developers and testers) and the **stakeholders** (course instructor).

## Scope

## This document covers the **user roles** of the Libman system (Admin, User), its **core functionalities** (inventory management, borrowing/returning), **external interfaces**, **performance expectations**, and **quality attributes**. Elements that are **out of scope** (such as e-book reading and budget management) are specified in the **Project Definition document**.

## Definitions, Acronyms, Abbreviations

## ****SRS:**** Software Requirements Specification The document that defines the functional and non-functional requirements of the software system.

**Admin:** Library Administrator  
The staff member responsible for managing the library’s inventory, users, and borrowing/returning processes.

**User:** Students and Faculty Members  
The end users of the system who can search for, borrow, and return books.

**FR:** Functional Requirement  
A specific functionality or behavior that the system must provide.

## References

## PA1\_481.pdf

Project Definition.docx

Project Plan.docx

## Overview

## The remainder of this document provides a detailed description of the **overall system overview (Section 2)** and the **specific requirements (Section 3)**.

## 2. Overall Description

## Product Perspective

## **Libman** is a standalone **web-based system** designed to replace the manual processes currently used in our university library.

## Product Functions

## The core functions of the system include **inventory management** (adding/removing books) for the **Admin**, and **catalog browsing**, **viewing book availability**, and **borrowing/returning books** for the **User**.

## User Characteristics

## ****User (Student/Faculty Member):**** Individuals with basic web literacy who will use the system to search for and borrow books.

**Admin (Library Staff):** Personnel with basic computer skills responsible for managing the inventory and overseeing borrowing and returning operations.

## Constraints

## The project must be developed by a team of four students within the time constraints defined by the course. The system will be built using **free and open-source technologies** (such as PostgreSQL, MySQL, Python, or Java). The system must operate on **modern web browsers** running on **personal computers** .

## Assumptions and Dependencies

## It is assumed that a **server environment** will be available for **system deployment**. It is also assumed that **users (students)** possess a **valid university email address** required for system registration.

## 3. Specific Requirements

## Functional Requirements

## ****1. Authentication****

1. The system shall authenticate users with the **“User”** role (Students/Faculty Members) using a **username and password**.
2. The system shall authenticate individuals with the **“Admin”** role (Library Staff) using a **privileged username and password**.

#### ****2. User Functions****

1. The system shall allow users to **search the library catalog** by **Book Title**, **Author Name**, or **ISBN number**.
2. The system shall display the **real-time availability status** of each book in the search results as either **“Available (On Shelf)”** or **“Borrowed”**.
3. The system shall allow users to request to **borrow** a book that is currently **available**.
   * Upon successful borrowing, the book’s status shall automatically change to **“Borrowed.”**
4. The system shall allow users to **return** a book they have borrowed.
   * Upon successful return, the book’s status shall automatically change to **“Available.”**
5. The system shall allow users to **view their current borrowed books** and their **due dates** on their profile page.
6. The system shall allow users to **view their borrowing history**, showing the books they have previously borrowed .

#### **3. Admin Functions**

1. The system shall allow the **Admin** to **add a new book** to the system, including details such as **Title, Author, ISBN, and Quantity**.
2. The system shall allow the **Admin** to **remove a book** from the system (e.g., in cases of loss or damage).
3. The system shall allow the **Admin** to **view a list of all registered user** showing which user has borrowed which book.
4. Admin manages user’s book related requests.

## External Interface Requirements

#### ****1. User Interface (UI)****

#### The system shall provide a **web-based graphical user interface (GUI)**. The interface shall be **clean, clear, and intuitive**, ensuring ease of use for both Admin and User roles.

## Performance Requirements

#### **1. Performance**

The system shall respond to search queries within **2 seconds**, even when operating on a database containing **1000 books**.

#### **2. Usability**

The interface shall be **user-friendly**, and the process of borrowing a book shall not require more than **three clicks** from the user.

#### **3. Compatibility**

The web application shall operate smoothly on the **latest versions of modern web browsers**, including **Google Chrome** and **Mozilla Firefox**.

#### **4. Maintainability**

The codebase shall be **easily understandable and extensible**, allowing new features to be added in the future through a **modular design structure**.

## Design Constraints

## The system shall be developed using **free and open-source technologies**.

For **backend development**, either **Python** or **Java (Spring Boot)** shall be used.

The **database** shall be implemented using either **PostgreSQL** or **MySQL**.

**Git** shall be used as the **version control system**, and **GitHub** shall serve as the **code repository**.

## Other Requirements

Include additional requirements not covered above.

**Effort Estimations**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Description** | **Estimated Effort (Person-Hours)** |
| **Phase 1:** Requirements Gathering and Planning | Collecting requirements, defining project scope, and planning tasks. | 20 Person-Hours |
| **Phase 2:** Design (UI/UX and Database) | Designing the user interface and database schema. | 16 Person-Hours |
| **Phase 3:** Development (Coding) | Implementing system features, backend, and frontend. | 60 Person-Hours |
| **Phase 4:** Testing | Conducting unit, integration, and user acceptance tests. | 20 Person-Hours |
| **Phase 5:** Deployment and Documentation | Deploying the system and preparing final documentation. | 12 Person-Hours |
| **Total Estimated Effort** | — | **128 Person-Hours** |

**Task Assignments**

**Ahmet Taha ÖZCAN:** Project Manager (PM) & Backend Developer  
Responsible for overall project coordination, backend implementation, and ensuring timely progress.

**Duygu AKMAN:** Requirements Analyst & Frontend/UI Developer  
Responsible for gathering and analyzing requirements, designing the user interface, and implementing frontend components.

**Mehmet Fatih AKAY:** Lead Developer (Backend & Database)  
Responsible for backend logic, database design and management, and ensuring overall code quality and integration.

**Kaan BEHZETOĞLU:** Designer (UI/UX) & Tester (QA)  
Responsible for creating user interface designs, improving user experience, and conducting testing and quality assurance activities.

|  |  |  |
| --- | --- | --- |
| **Functional Requirements (FR)** | **Responsible Developer(s)** | **Responsible Tester** |
| **Authentication (FR1, FR2)** | Fatih | Kaan |
| **Catalog Search (FR3)** | Ahmet | Kaan |
| **View Availability Status (FR4)** | Duygu (Frontend), Fatih (Backend) | Kaan |
| **Borrowing Process (FR5)** | Duygu (Backend) | Kaan |
| **Return Process (FR6)** | Ahmet (Backend) | Kaan |
| **User Profile (FR7, FR8)** | Duygu (Frontend), Fatih (Backend) | Kaan |
| **Admin – Add Book (FR9)** | Fatih | Kaan |
| **Admin – Delete Book (FR10)** | Duygu | Kaan |
| **Admin – Reporting (FR11)** | Ahmet | Kaan |
| **Admin – Request Management (FR12)** | Fatih | Kaan |
| **UI/UX Design** | Kaan & Duygu | Ahmet & Fatih |

## 4. Appendices

## 5. References

PA1\_481.pdf

Project Definition.docx

Project Plan.docx

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasks \ Members | Duygu AKMAN | Ahmet Taha ÖZCAN | Mehmet Fatih AKAY | Kaan BEHZETOĞLU |
| Introduction | X |  |  |  |
| Overall Description |  |  | X | X |
| Functional Requirements | X |  |  |  |
| Performance, Constraints, Quality Attributes |  | X |  | X |
| Effort Estimations |  | X |  |  |
| Task Assignments |  |  | X |  |
| Review | X | X | X | X |