# Library Book Management System - Documentation

## 1. Stakeholder Interview Summary

A meeting was held with a school librarian, who explained the need for a system that records book details like title, author, edition number, number of copies, classification, and shelf location. He emphasized tracking active borrowings: who borrowed what, and when.  
He noted the importance of setting a borrowing limit depending on student level and enabling alerts for overdue returns. Weekly reports such as borrowed books count and names of overdue students were requested.  
  
In a separate meeting with a school administration representative, they expressed interest grade, and late students.  
They stressed the importance of ease of use, Arabic language support, access control between librarian/admin, fast search, and printable reports.

## 2. Functional Requirements

1. Record book details: title, author, edition, number of copies, classification, shelf.  
2. Record student borrowing (name, grade, book title, borrow date).  
3. Set a borrowing limit per student.  
4. Alert if a student is late returning a book.  
5. Generate weekly/monthly reports (late students, top books, etc.)  
6. Differentiate roles: Librarian vs Admin.  
7. Enable fast search and book status view.  
8. Support printing reports.

## 3. Non-Functional Requirements

1. Easy and intuitive UI.  
2. Responsive speed when loading/searching.  
3. Support Arabic.  
4. Printable outputs.  
5. Secure login for each role.

## 4. Visual Design & Tools

This section outlines the visual modeling used to plan the system.

### A. Use Case Diagram

Helps understand:  
- Who are the users (Actors)?  
- What do they do in the system (Use Cases)?  
Useful to present a clear overview in presentations.

A diagram of a book

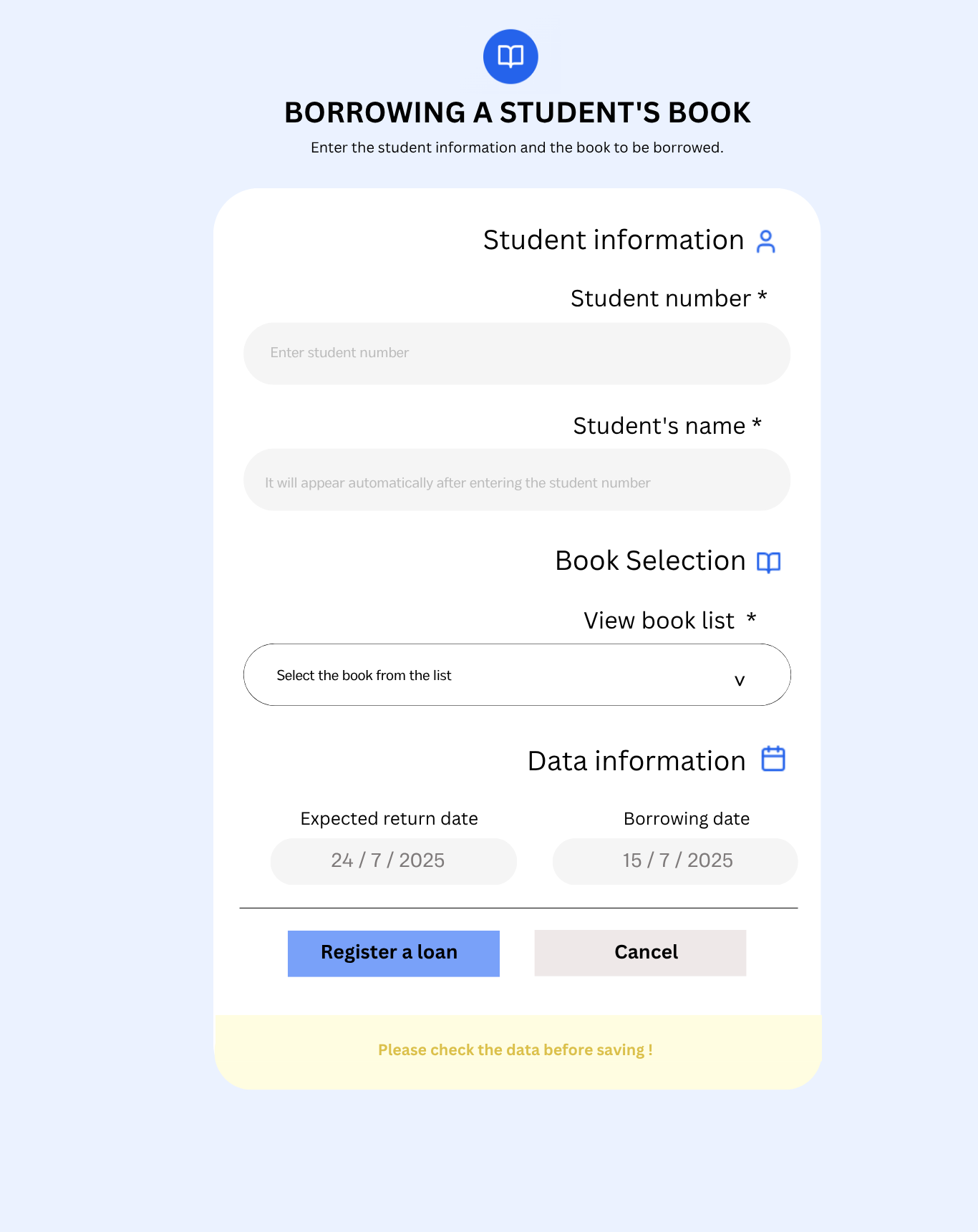
AI-generated content may be incorrect.

### B. ERD (Entity-Relationship Diagram)

Helps understand:  
- What tables exist in the database.  
- How they relate (One-to-Many relationships, etc.)  
Useful for planning the database schema.

### A screenshot of a computer AI-generated content may be incorrect.C. Wireframe (UI Mockup)

Helps visualize:  
- Book registration page.  
- Student borrowing interface.  
Useful for showing how the final system might look.



## Tools Used

- Use Case Diagram: draw.io / diagrams.net  
- ERD: dbdiagram.io  
- Wireframe: Canva

## 5. Programming (Implementation):

In this phase, designs are converted into real code using languages ​​such as C# or Python.

We did not actually implement this phase because the goal of the project is to learn and focus on planning and analyzing the system, not building it entirely programmatically.

## 6. Testing:

After programming, the system is tested to ensure it functions correctly and is error-free.

Types of testing include:

• Unit Test

• System Test

• User Acceptance Test

No actual testing was conducted because the system has not yet been programmed.

## 7. Deployment:

In this phase, the system is delivered to the school or uploaded to a client device or server.

This phase was not implemented because the project is purely analysis and design, and has not yet reached the technical delivery stage.

## 8. Maintenance:

After the system is delivered, the follow-up phase begins, such as:

• Modifying or fixing errors that appear later

• Improving functionality according to the school's needs

We have not reached this stage because it follows the actual operation of the system.

## Conclusion:

This project focuses on the first basic phases (requirements, analysis, design), which are the foundation upon which any successful system is built.

As for the subsequent phases, they have been mentioned and their roles explained, but they have not been implemented because the project is still in the design and study phase, not in the actual implementation phase. However, the project will be fully completed and operational soon.