A LIBRARY MANAGMENT SYSTEM

SE112

2023-2024 Academic Semester

GROUP NO: 30

TEAM MEMBERS: Aleyna Dikmen 23244710049 Ahmet Kağan Doğan 20244710043

Çağatay Uğurlu 23244710040

CONTENTS

[1.PROJECT PROPOSAL 3](#_Toc165844394)

[2.PROJECT MANAGMENT 4](#_Toc165844395)

[3.METHODOLOGY 6](#_Toc165844396)

[4.MOSCOW SCHEME 8](#_Toc165844397)

[5.USER-STORY 9](#_Toc165844398)

[6.CASE DİAGRAM 10](#_Toc165844399)

[7.CLASS DİAGRAM 11](#_Toc165844400)

[8.SEQUENCE DİAGRAM 12](#_Toc165844401)

[9.MOCK-UPS 13](#_Toc165844402)

# 1.PROJECT PROPOSAL

Traditional library management systems often encounter limitations in terms of accessibility and user engagement. Our project aims to overcome these challenges by introducing groundbreaking features enabled by innovative technologies. By offering intuitive interfaces, personalized recommendations, and streamlined processes, we seek to transform the library experience, making it more efficient, interactive, and enjoyable for all stakeholders.

# 2.PROJECT MANAGMENT

2.1 PROJECT PLAN

Library Management System

Project Overview:

Develop a web-based library management system incorporating innovative technologies to enhance accessibility, efficiency, and user experience.

Project Objectives:

Create a modern, user-friendly interface for library staff and patrons.

Implement features such as personalized recommendations and advanced search capabilities.

Utilize innovative technologies (e.g., machine learning, cloud computing) to enhance functionality.

Ensure scalability, reliability, and security of the system

2.2 RISK PLAN

Risk 1: Integration Complexity

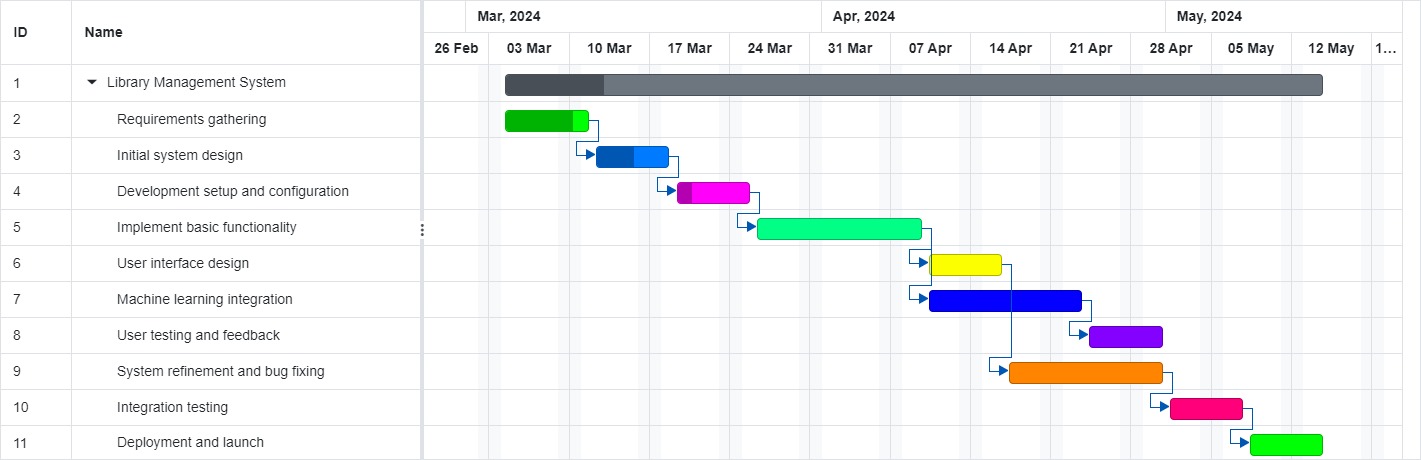
B-Plan: Develop a modular architecture to facilitate easier integration of innovative technologies. Prioritize the identification and resolution of integration challenges during the development phase. Establish clear communication channels with technology providers or experts to seek timely assistance in resolving integration issues. Maintain a contingency budget and timeline buffer to accommodate unforeseen delays or complications related to technology integration. Regularly review and adjust the integration strategy based on emerging challenges and feedback from stakeholders.

Risk 2: User Adoption and Training

B-Plan: Conduct comprehensive user training sessions prior to the system launch to familiarize library staff and patrons with the new system. Develop user-friendly interfaces with intuitive navigation to minimize the learning curve for users. Implement feedback mechanisms to gather user insights and address usability issues promptly. Provide ongoing support and assistance to users through helpdesk services, online resources, and user community forums. Collaborate with library administrators to promote the benefits of the system and encourage active user participation through incentives and rewards programs.

# 3.METHODOLOGY

To achieve our objectives, we will employ an agile software development methodology. Agile methodologies emphasize adaptability, collaboration, and iterative development, allowing us to respond swiftly to evolving requirements and stakeholder feedback. By breaking the project into manageable sprints, we can prioritize key features, maintain a high level of transparency, and deliver continuous value to users throughout the development process. This approach ensures that our final product meets the evolving needs and expectations of the library community.



# 4.MOSCOW SCHEME

|  |  |  |
| --- | --- | --- |
| Functional Reqs ID | Explanation | MOSCOW priority (M, S, C, W) |
| FR1 | Users should be able to search for books by title, author, or category | M (Must-have) |
| FR2 | Users should be able to view detailed information about each book, including summary, availability, and location. | M (Must-have) |
| FR3 | Librarians should be able to add new books to the system, including title, author, category, and other relevant details. | M (Must-have) |
| FR4 | Users should be able to reserve books online, with notifications sent upon availability. | M (Must-have) |
| FR5 | Librarians should be able to update book availability status and location in real-time. | S (Should-have) |
| FR6 | Users should be able to view their borrowing history and pending reservations. | S (Should-have) |
| FR7 | Librarians should be able to generate reports on book usage, popular titles, and overdue books | S (Should-have) |
| FR8 | Users should be able to receive personalized book recommendations based on their reading preferences | C (Could-have) |
| FR9 | Librarians should be able to manage user accounts, including registration, suspension, and deletion. | C (Could-have) |
| FR10 | Users should be able to renew borrowed books online, if no reservations are pending. | W (Won't-have) |
| Non-Functional Reqs ID | Explanation | MOSCOW priortiy  (M, S, C, W) |
| NFR1 | The system should have a response time of less than 2 seconds for book searches. | M (Must-have) |
| NFR2 | The system should be accessible from multiple devices, including desktop computers, tablets, and smartphones | S (Should-have) |
| NFR3 | The system should comply with industry-standard security protocols to protect user data and transactions | C (Could-have) |

# 5.USER-STORY

User Story: Managing Library Operations

Title: Managing Library Operations

As a: User (Student or academic staff), Library\_Admin

I want to: Perform various tasks related to managing library operations.

So that: I can effectively utilize library resources and ensure smooth library functioning.

Acceptance Criteria:

As a user, I can search for books by title, author, or category.

As a user, I can view detailed information about each book, including its availability status and location.

As a user, I can borrow books from the library.

As a user, I can renew the borrowed books online if no reservations are pending and if the renewal limit has not been reached.

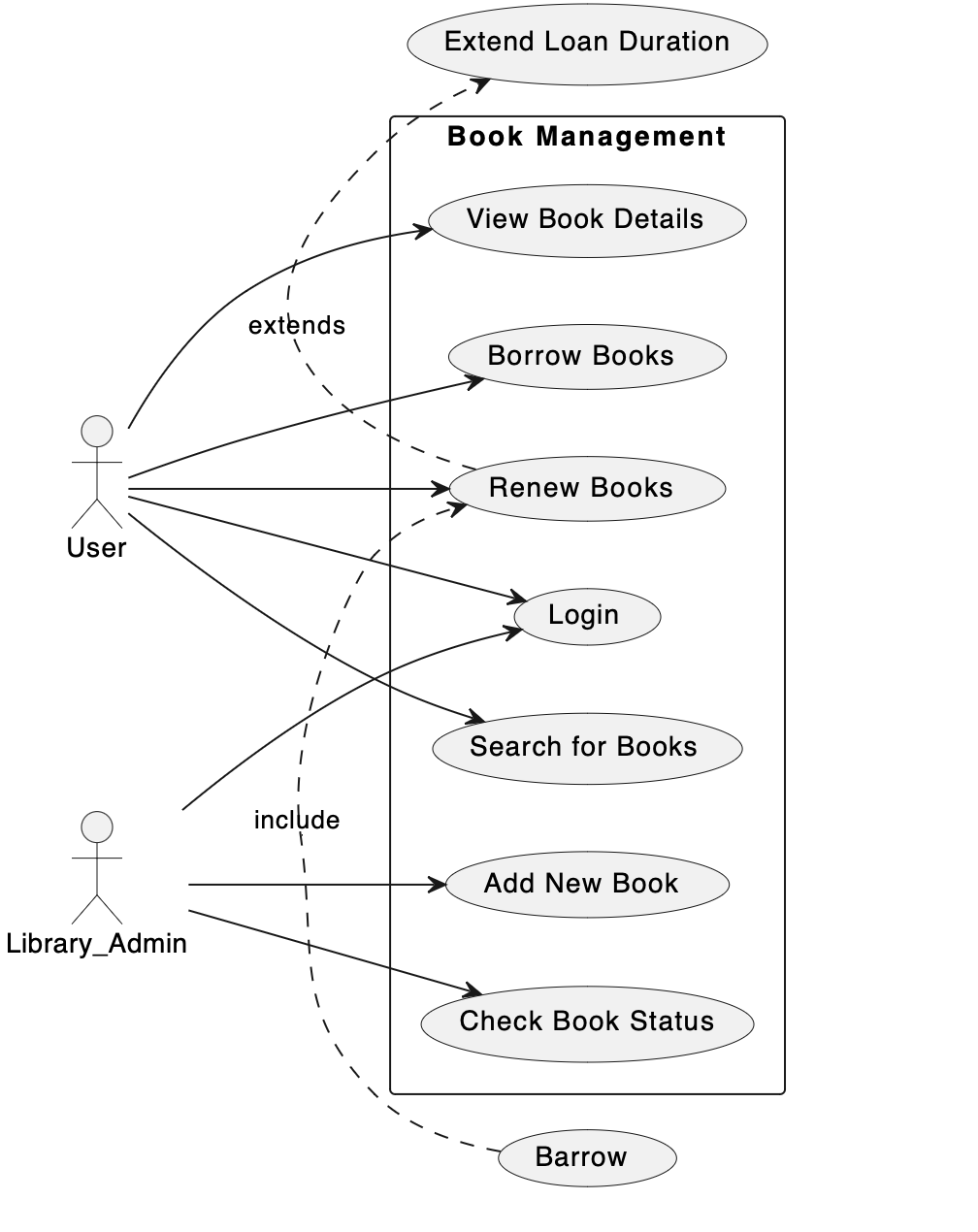
As a Library\_Admin, I can check the status of a book.

As a Library\_Admin, I can call back a book to the library if necessary.

All users (User, Library\_Admin) must log in to the library system before using it.

Academic\_Staff and Library\_Admin can extend the duration of book loans online.

# 6.CASE DİAGRAM



@startuml

left to right direction

actor User as U

actor Library\_Admin as LA

rectangle "Book Management" {

usecase "Search for Books" as Search

usecase "View Book Details" as View

usecase "Borrow Books" as Borrow

usecase "Renew Books" as Renew

usecase "Check Book Status" as Check

usecase "Add New Book" as AddBook

usecase "Login" as Login

U --> Search

U --> View

U --> Borrow

U --> Renew

LA --> Check

LA --> AddBook

U --> Login

LA --> Login

}

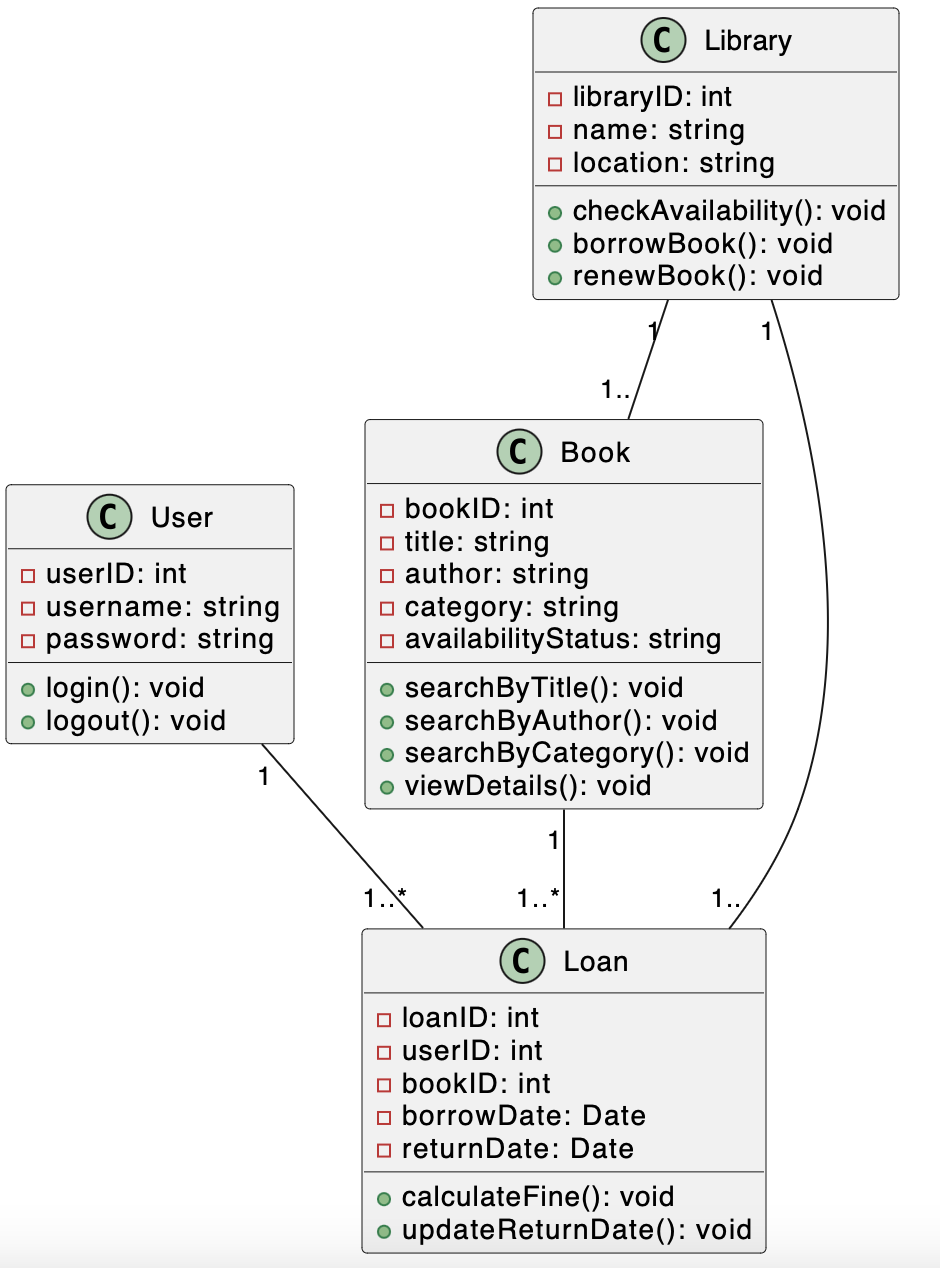
(Barrow) .> (Renew) : include

(Renew) .> (Extend Loan Duration) : extends

@enduml

# 

# 7.CLASS DİAGRAM



@startuml

class User {

- userID: int

- username: string

- password: string

+ login(): void

+ logout(): void

}

class Book {

- bookID: int

- title: string

- author: string

- category: string

- availabilityStatus: string

+ searchByTitle(): void

+ searchByAuthor(): void

+ searchByCategory(): void

+ viewDetails(): void

}

class Library {

- libraryID: int

- name: string

- location: string

+ checkAvailability(): void

+ borrowBook(): void

+ renewBook(): void

}

class Loan {

- loanID: int

- userID: int

- bookID: int

- borrowDate: Date

- returnDate: Date

+ calculateFine(): void

+ updateReturnDate(): void

}

User "1" -- "1..\*" Loan

Book "1" -- "1..\*" Loan

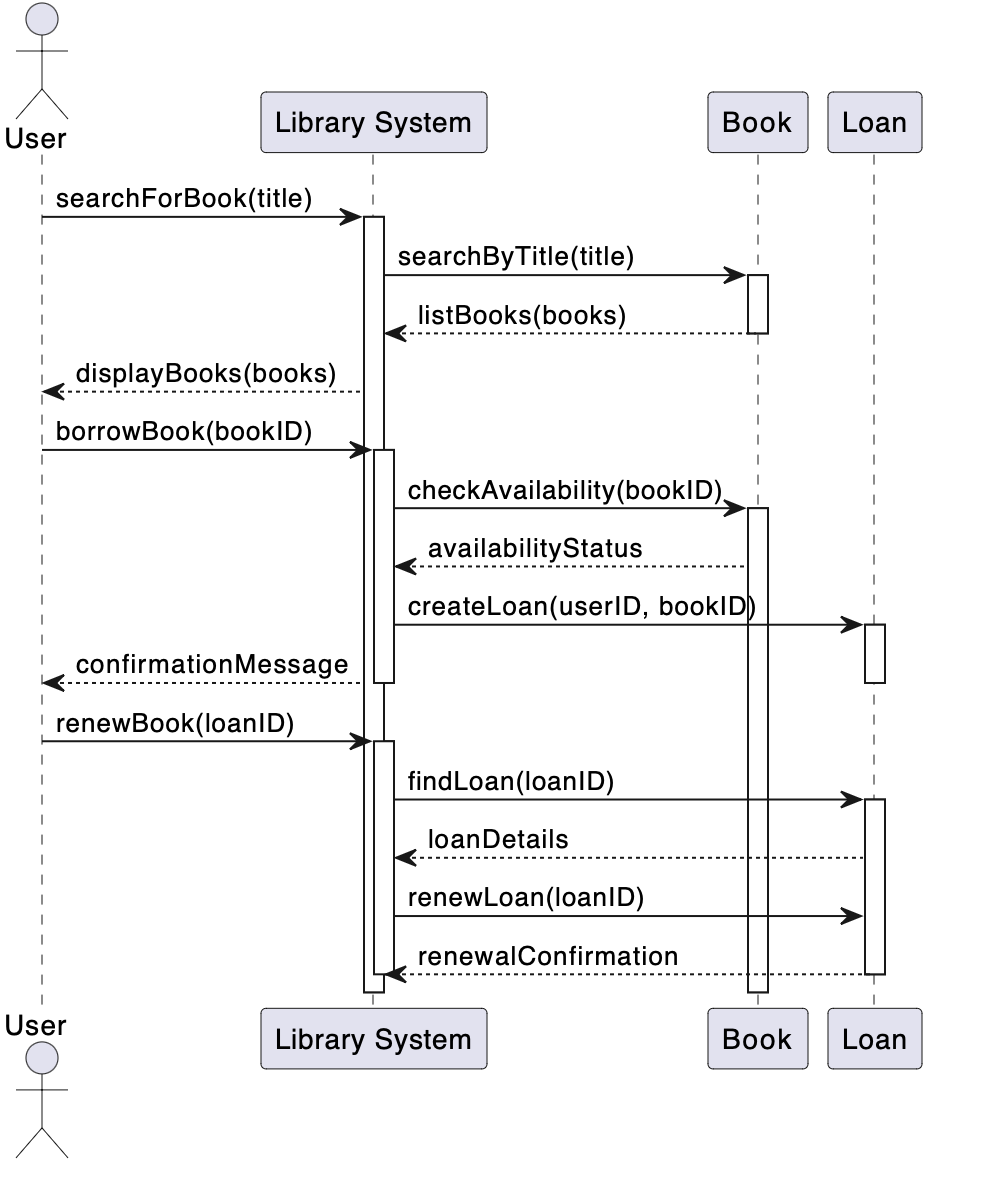
Library "1" -- "1.." Book

Library "1" -- "1.." Loan

@enduml

# 

# 8.SEQUENCE DİAGRAM



@startuml

actor User

participant "Library System" as LS

participant "Book" as B

participant "Loan" as L

User -> LS: searchForBook(title)

activate LS

LS -> B: searchByTitle(title)

activate B

B --> LS: listBooks(books)

deactivate B

LS --> User: displayBooks(books)

User -> LS: borrowBook(bookID)

activate LS

LS -> B: checkAvailability(bookID)

activate B

B --> LS: availabilityStatus

LS -> L: createLoan(userID, bookID)

activate L

LS --> User: confirmationMessage

deactivate L

deactivate LS

User -> LS: renewBook(loanID)

activate LS

LS -> L: findLoan(loanID)

activate L

L --> LS: loanDetails

LS -> L: renewLoan(loanID)

L --> LS: renewalConfirmation

deactivate L

deactivate LS

@enduml

# 9.MOCK-UPS

