



**Software Engineering**

**FALL 2024**

**PROJECT TITLE**

***E-BOOK LIBRARY SITE***

**PARTICIPANTS**

*Eren YÜKSEL – 200316068*

*Berkay ARI – 190316066*

*Tural MAMMADOV – 190316001*

*Mert KÖSA – 200316018*

*Batuhan DURAN – 200315087*

*İbrahim Ata ALIŞ - 190316014*

# Feasibility Report

## 1. Executive Summary

The project aims to develop an online library management system where users can browse, search, and read books online. Users can also add books to their library, track their reading progress, and save last reading page from where they left off. This system will enhance the user experience for book enthusiasts by providing a digital library solution accessible anytime and anywhere.

## 2. Objectives

- Develop a user-friendly library application using .NET Core MVC.
- Provide features for searching, viewing, and reading books online.
- Allow users to track and save their reading progress.
- Ensure scalability, security, and maintainability of the system.

## 3. Technical Feasibility

- **Technology Stack:**
  - Backend: .NET Core MVC Framework
  - Frontend: Razor Views, HTML, CSS, JavaScript
  - Database: PostgreSQL
  - Tools: Visual Studio, Visual Studio Code, Github
  - Hosting: Local
- **Skills Required:**
  - Proficiency in C# and .NET Core.
  - Knowledge of Entity Framework Core for database handling.
  - Understanding of MVC architecture.

#### 4. Operational Feasibility

- **Target Users:** Students, professionals, and book enthusiasts who prefer digital platforms.
- **Ease of Use:** A simple and intuitive user interface will ensure usability for all age groups.
- **Maintenance:** Regular updates and backups will keep the system reliable and operational.

#### 5. Economic Feasibility

- **Initial Investment:**
  - Development tools: Free for individual developers using Visual Studio Community Edition.
- **Potential Revenue:**
  - Advertisement placements for monetization.
- **Cost vs. Benefit Analysis:**
  - Reduced costs for physical library maintenance.
  - Increased user base and reach.

#### 6. Legal Feasibility

- **Copyright Compliance:** Ensure all books are legally sourced or public domain.
- **Terms of Use:** Define acceptable use policies for users.

## PLANNING PROGRAM

Phase	Task	Duration
Planning & Requirement	Define scope, gather requirements	2 Weeks
System Design	Create architecture and UI wireframes	2 Weeks
Development	Code application (backend & frontend)	2 Weeks
Testing	Perform unit, integration, and UAT	1 Week
Deployment	Deploy to production environment	1 Week
Maintenance	Ongoing updates and support	Continuous

## RISK MANAGEMENT

Risks	Mitigation Strategies
<ul style="list-style-type: none"><li>• Data Breach</li></ul>	<ul style="list-style-type: none"><li>• Use encryption for sensitive user data.</li></ul>
<ul style="list-style-type: none"><li>• Performance Issues</li></ul>	<ul style="list-style-type: none"><li>• Optimize database queries and use caching.</li></ul>
<ul style="list-style-type: none"><li>• Delays in Development</li></ul>	<ul style="list-style-type: none"><li>• Break tasks into smaller sprints with reviews.</li></ul>
<ul style="list-style-type: none"><li>• Legal Issues with Book Content High</li></ul>	<ul style="list-style-type: none"><li>• Verify legal status before adding books.</li></ul>

## 1. Data Security Risks

- **Risk:** Unauthorized access to sensitive user or library data (e.g., user personal details, book inventory).
- **Solution:**
  - Implement encryption for data storage and communication (e.g., SSL/TLS).
  - Use role-based access controls and strong authentication (e.g., two-factor authentication for admins).
- **Alternative Scenario:**
  - If data breaches occur, have an incident response plan to quickly isolate the issue and notify stakeholders while resolving vulnerabilities.

## 2. Performance Issues with High User Traffic

- **Risk:** System performance degrades during peak usage (e.g., many users searching or reserving books simultaneously).
- **Solution:**
  - Use load balancing and scalable server infrastructure.
  - Optimize database queries and implement caching for frequently accessed data.
- **Alternative Scenario:**
  - If the system slows down, provide a fallback “read-only mode” or queued access for users until resources are restored.

## 3. User Adoption Challenges

- **Risk:** Users (especially admins) find the system difficult to use or do not adopt it fully.
- **Solution:**
  - Conduct user training sessions.
  - Use intuitive design principles in the user interface.

- **Alternative Scenario:**

- Collect feedback and implement iterative improvements to address user complaints or usability issues.

#### **4. Legal Issues with Book Content High**

- **Risk:** The library system may unintentionally host or distribute books with copyrighted material, plagiarized content, or content that violates laws (e.g., hate speech, defamation). This could lead to lawsuits, fines, or reputational damage.
- **Solution:**
  - Delete copyrighted book from the database .

### **Functional requirements:**

- The system must allow users to register or log in.
- The system should establish a system for users to save and read books.
- There should be a section with a list of books available in the system.
- There should be a search page for users to search and save books to library.
- The system should allow the users to keep track of the last page they read.

### **Non-Functional requirements:**

#### **Performance-->**

- Page loading time should be 2 seconds
- Must support 1000 active users simultaneously

#### **Security-->**

- Passwords should be stored securely
- Unauthorized persons should not be able to access the system

#### **Usability-->**

- The interface should be understandable and fast

### **Reliability-->**

-Pirate ads should not be allowed on the site

## **User Requirements:**

- Users must be able to create an account and log in.
- Users should be able to search for books quickly.
- Users should be able to save books to the list they want to read online.
- Users should be able to view and delete the books they saved in the bookshelf.
- Users should be able to read the books they saved in the bookshelf.
- Users should be able to see their reading progress.

## **System Requirements:**

### **Account Management-->**

A user authentication module (password encryption, password reset) that allows users to register and log in.

### **Search System-->**

A search engine that filters books by title, author, or genre.

### **Reservation Module-->**

Users can reserve an existing book online and manage reservation status and issue refunds a system.

### **Rental and Return Tracking-->**

A panel showing the details and return dates of the books rented by users.

### **Notification System-->**

An email or push notification system that reminds users of booking confirmation and return dates.

### **Admin Panel-->**

An administrative interface for the administrator that includes functions for adding, editing and deleting books.

## **System Stakeholders:**

**Primary Stakeholders:** Individual users, System owners, Developers

**Secondary Stakeholders:** IT support team, book suppliers, parents/educators.

**Tertiary Stakeholders:** Legal institutions, society and software developers.

## **Usability Requirements:**

- First-time users should be able to navigate the website and perform basic tasks.
- The system should be easy to use for users.
- User errors should be minimized.
- A help section or FAQs should be accessible from all pages, providing guidance on common

## **Requirements Engineering Processes:**

### **1. Elicitation**

This stage involves gathering and identifying the requirements from all stakeholders.

#### **Key Activities:**

- **Stakeholder Interviews:**
  - Conduct interviews with Scrum master, developer team and customer to understand their needs and expectations.
- **Workshops and Brainstorming:**
  - Organize collaborative workshops to discuss desired features such as book search, book reading and database.
- **Observation:**
  - Study the current manual or digital system to identify pain points and improvements.
- **Questionnaires and Surveys:**
  - Distribute surveys to gather input from a wider audience, including library users and potential users.



## 2. Analysis

Once requirements are gathered, they must be structured, prioritized, and evaluated for feasibility.

### Key Activities:

- **Requirement Categorization:**
  - Divide requirements into functional (e.g., book search) and non-functional (e.g., system scalability).
- **Feasibility Analysis:**
  - Evaluate technical, financial, and legal feasibility of the requested features.
- **Requirement Prioritization:**
  - Identify critical requirements that need immediate attention and ensure they align with the overall project objectives, balancing user needs and resource availability.
- **Modeling:**
  - Create use case diagrams, sequence diagrams, and data flow models to visualize requirements.

## 3. Validation

Validation ensures that the requirements meet stakeholder needs and are free of errors or ambiguities.

### Key Activities:

- **Requirement Reviews:**
  - Conduct walkthroughs with stakeholders to verify requirements' correctness and completeness.
- **Prototyping:**
  - Develop low-fidelity prototypes of key features (e.g., search functionality) to get feedback from users.
- **Test Case Development:**
  - Create preliminary test cases to ensure requirements can be tested effectively.

- **Consistency Checks:**

- Ensure there are no conflicting or overlapping requirements.

#### **4. Management**

This stage ensures that the requirements remain relevant and manageable throughout the project lifecycle.

##### **Key Activities:**

- **Requirement Documentation:**

- Maintain a central repository (e.g., a requirements specification document) to track all requirements.

- **Change Management:**

- Define a process for handling new or changing requirements, including impact analysis.

- **Traceability:**

- Map requirements to their respective implementation and testing phases to ensure alignment.

- **Periodic Reviews:**

- Regularly revisit requirements to verify they remain valid as the project

#### **Scenarios:**

**Scenario 1:** A user wants to search for a book and read-->

##### **Tasks:**

1. Finding the search bar on the home page after logging in.
2. Do not type the name or author of the book into the search bar.
3. Finding the book in search results.
4. Checking whether the book is available by opening its detail page.

**Scenario 2:** A user wants to borrow a book by reserving it online-->

##### **Tasks:**

1. Log in to the account or open a new account if there is no account.
2. Find the book using the search bar.
3. Clicking the "Reserve" button on the detail page of the book.

4. Confirm the reservation and see the success message.

**Scenario 3:** A user wants to view and manage borrowed books-->

**Tasks:**

1. Login to account.
2. Locating the "Borrowed Books" tab in the user panel.
3. Review the books listed and their return dates.
4. If necessary, submit a request for early return of the book.

**Scenario 4:** An admin wants to add a new book to the library system-->

**Tasks:**

1. Login to the admin panel.
2. Finding the "Add New Book" tab.
3. Filling in the book's information (title, author, genre, ISBN).
4. Adding the book to the system and seeing the success message.

## **Structured Specifications:**

**Scenario 1:** A user wants to search for a book and read

**Description:** The user wants to search for a particular book and check if it is available.

**Preconditions:**

- The user must be able to access the site.
- The book database must be up to date.

**Actors:**

User

**Flow of Events:**

1. The user finds the search bar on the home page.
2. Writes the name or author of the book in the search bar.
3. The system shows the list of matching books.
4. The user clicks on the detail page of the book he wants.
5. The system displays the details of the book available.

**Postconditions:**

The user gets to know the current status of the book.

**Exceptions:**

If the searched book is not found, the message "Search results not found" is displayed.

**Scenario 2:** A user wants to borrow a book by reading it online-->

**Description:** The user wants to read a book online.

**Preconditions:**

- The user must be logged in to the site.
- The book must be available for read.

Actors:

User

System

**Flow of Events:**

1. After logging in, the user finds the book using the search bar.
2. Goes to the book's detail page and clicks the "Save" button.
3. The system processes the request and displays the confirmation message.

**Postconditions:**

The book reservation is completed successfully.

**Exceptions:**

If the book is not in database, the message "The book is not found" is displayed.

**Scenario 3:** A user wants to view and manage saved books-->

**Description:** The user wants to see the books he saved and read.

**Preconditions:**

- The user must be logged in.
- Saved books must be registered in the system.

**Actors:**

User

System

**Flow of Events:**

1. The user opens the "Saved Books" tab on the profile page.
2. The system shows the user the list of saved books and last page read.
3. If the user deems it necessary, he/she delete the books from library.

**Scenario 4:** Admin Adds a New Book to the Library System-->

**Description:**

An administrator wants to add a new book to the library system. Information such as book title, author name, genre and ISBN must be recorded in the system.

**pre conditions:**

- The administrator must have an active account and the necessary authorizations.
- The library system must be online and operational.

**Actors:**

Executive

Library System

**Flow of Events:**

1. The administrator logs into the admin panel:
2. The administrator goes to the login page and enters the valid username and password.

The system verifies the administrator credentials and provides access to the admin panel.

The admin finds the "Add New Book" tab:

3. The administrator clicks the "Add New Book" button on the panel.

The system displays a form where you can enter book information.

The administrator fills in the book information:

4. The administrator enters the book title, author, genre, ISBN, and other additional information (e.g., year of publication, number of copies).

The system checks the accuracy of the entered information and the completion of mandatory fields.

The administrator adds the book to the system:

5. The administrator clicks on the "Add Book" button.

The system saves the information in the database and updates the catalogue.

The administrator sees the success message:

6. The system says "Book added successfully!" displays the message.

The newly added book appears in the book list in the admin panel.

### **Exceptions:**

#### **Invalid Entry:**

If any required fields are left blank or data in an incorrect format is entered, the system displays an error message (e.g. "ISBN format is incorrect.").

The administrator must correct the information and resubmit.

#### **Duplicate Registration:**

If a book with the same ISBN is already registered in the system, the system displays a warning message (e.g. "This ISBN is already registered in the system.").

#### **System Error:**

If the system cannot add the book due to a technical problem, it displays an error message (e.g. "There was an error adding the book. Please try again.").

## **Requirement Checking:**

**Requirement 1:** Administrator, admin panel; The user must be able to log in to their account.

**Validity:**

The requirement is valid; In order to manage the system securely, only authorized persons must be able to log in.

**Consistency:**

Does not conflict with other security requirements. For example, the login process may require multi-factor authentication; complies with this requirement.

**Completeness:**

No other verification methods other than username and password are mentioned for login. These details may be added for completeness.

**Realism:**

An authentication mechanism (e.g. username and password) is a feasible and common method today.

**Verifiability:**

Testable; A username and password can be entered to check whether the system is working correctly.

**Requirement 2:** The user must be able to use the system where he can rent and return books.

**Validity:**

This requirement is valid; The user must find the relevant section in an accessible manner for book reservation and return transactions.

**Consistency:**

It is compatible with book data, when the user adds a book to his list, another book will not be shown.

**Completeness:**

The book information chosen by the user is given in full.

**Realism:**

Nowadays, renting books online is a common and realistic situation.

**Verifiability:**

The accuracy of the book purchased by the user can be determined by the ISBN code.

**Requirement 3:** There should be a section with a list of books available in the system.

**Validity:**

The requirement is valid. Directly supports the system's objectives.

**Consistency:**

The requirement is consistent. Aligns with related features; real-time sync is crucial to avoid conflicts.

**Completeness:**

The requirement feels incomplete. Needs details on displayed fields and user interactions (filtering, etc.).

**Realism:**

Technically feasible with current tools and design strategies.

**Verifiability:**

Testable with functional and user interface testing.

**Requirement 4:** There must be a panel where the administrator side can handle their work.

**Validity:**

This requirement is valid; The administrator must find the relevant section in an accessible manner for operations.

**Consistency:**

It complies with user interface design principles and does not conflict with other admin operations.

**Completeness:**

Additional fields may need to be included in the book information, such as publication year, number of pages, or language. These must be added completely.

**Realism:** Nowadays, it is a practical necessity for everyone to be able to access books online. Such sites can meet this need.



**Verifiability:**

Testable; The accuracy of the information entered and whether the book information has been filled in completely can be checked.

**System Boundaries:****System Inside (Internal Boundaries):****Book Database----->**

Database where books and related information are stored.

Access to tables containing book data such as ISBN, title, author, genre.

Adding, updating and deleting books.

**User Library Interface----->**

Search, view and filter books.

Book reservation, borrowing and return transactions.

Users' access and update of membership information.

**Feedback and Error Messages System----->**

Error conditions, success messages and user information.

Providing feedback to the user when adding or updating a book (e.g. "Book added successfully!", "This ISBN already exists!").

**System Outside (External Boundaries)****Users----->**

Library members accessing the system.

People with whom administrators and users interact through the interface.

## Integrations with Other Systems (Optional)----->

-Notification Systems: Sending notifications to users (for example, when a new book is added).

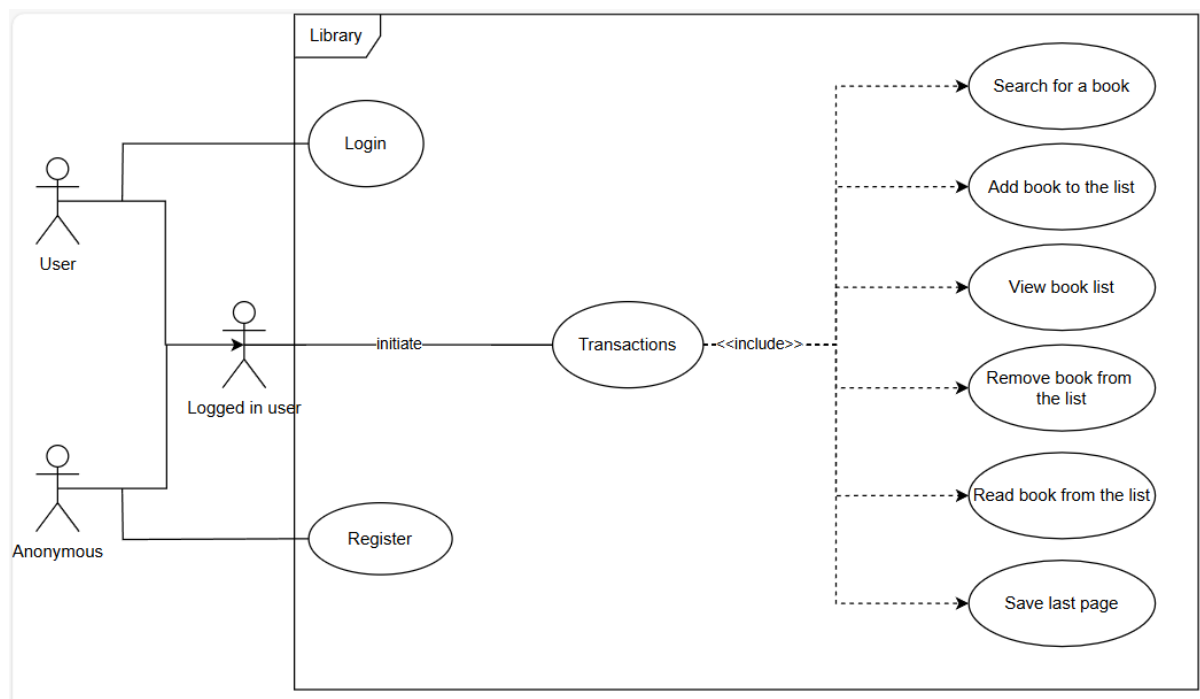
-External Library Resources: Integration into other library systems or databases (e.g. ISBN database control).

## Hardware and Software Infrastructure----->

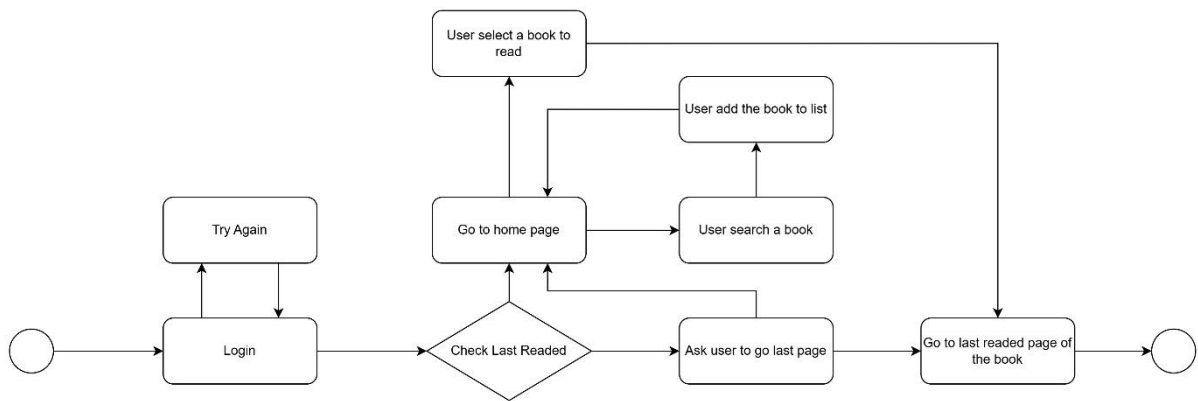
Server and Database: Infrastructure that stores and manages system data.

User Devices: Devices that access the library system (computers, mobile devices, etc.).

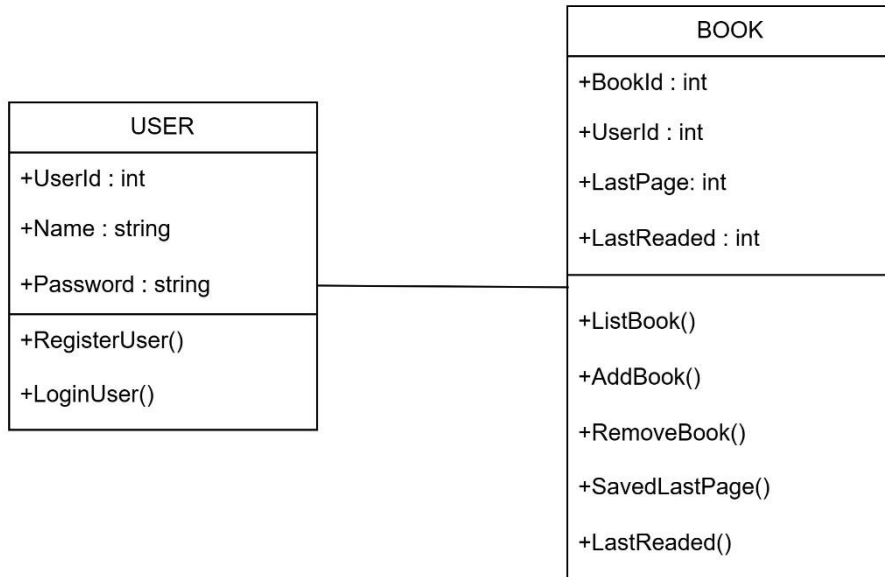
## Use Case Diagram:



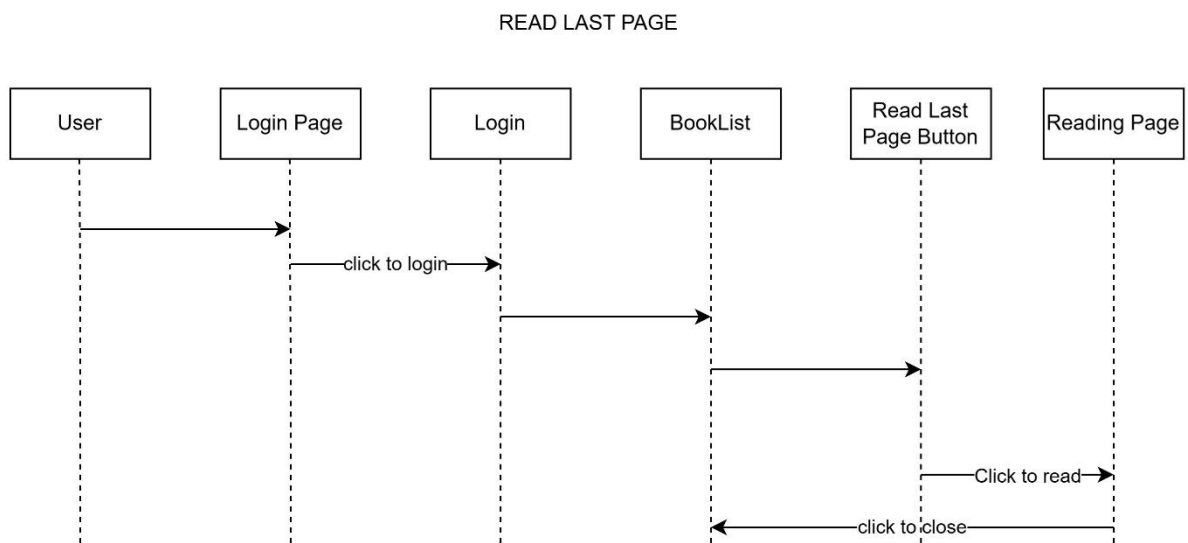
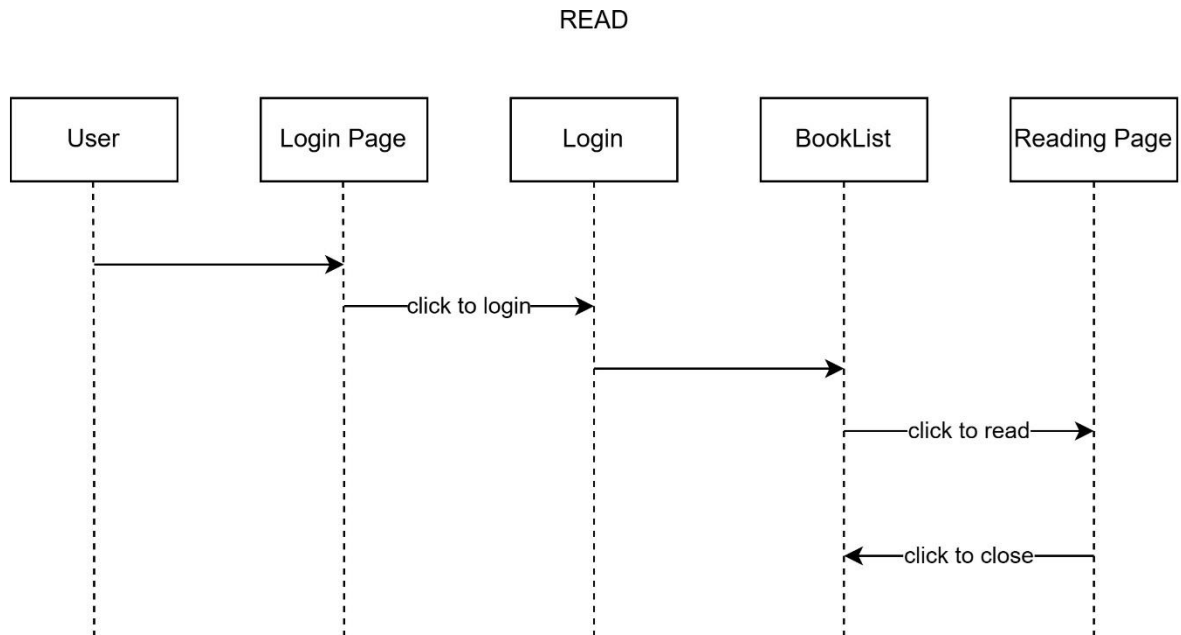
## State Diagram:



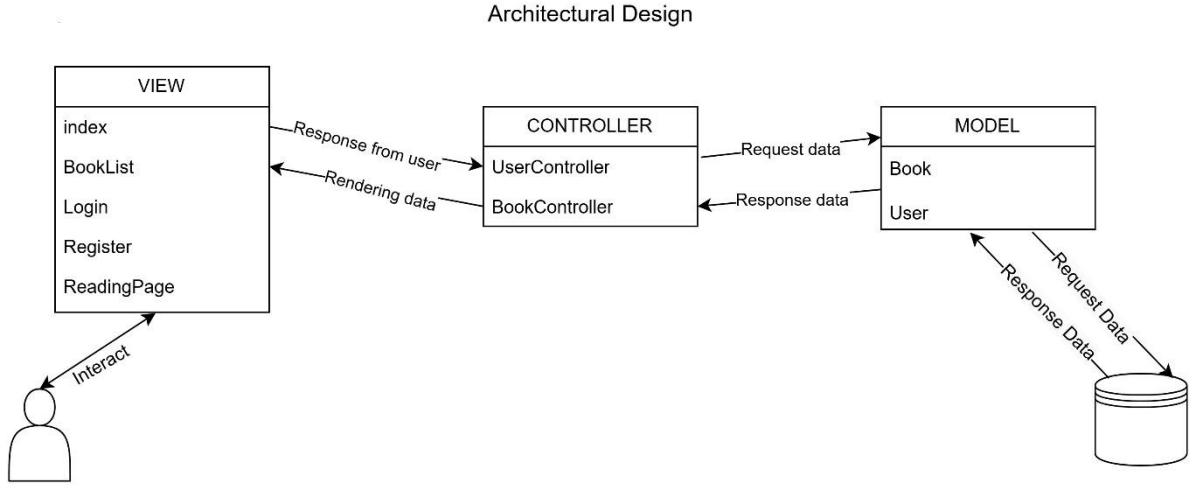
## Class Diagram:



## Sequence Diagrams:



# Architectural Design:



## UNIT TESTING RESULT

The screenshot shows the Visual Studio IDE with the following details:

- Explorer**: Shows the project structure with folders like SWEP, Models, Services, Views, and MyProject.Tests.
- Code Editor**: Displays the `GutenbergServiceTests` class with a `SearchBooksAsync` method and a `SendAsync` method.
- Terminal**: Shows the command `dotnet test` and the output of the test run.

**Test Results:**

```
PS C:\Users\mertk\OneDrive\Masadustu\SWEP\MyProject.Tests> dotnet test
Geri yüklenmek için tüm projeler belirleniyor...
Geri yükleme için tüm projeler güncel.
MyLibrarySite -> C:\Users\mertk\OneDrive\Masadustu\SWEP\MyLibrarySite\bin\Debug\net8.0\MyLibrarySite.dll
MyProject.Tests -> C:\Users\mertk\OneDrive\Masadustu\SWEP\MyProject.Tests\bin\Debug\net8.0\MyProject.Tests.dll
Geri yüklem için tüm projeler güncel.
MyLibrarySite -> C:\Users\mertk\OneDrive\Masadustu\SWEP\MyLibrarySite\bin\Debug\net8.0\MyLibrarySite.dll
MyProject.Tests -> C:\Users\mertk\OneDrive\Masadustu\SWEP\MyProject.Tests\bin\Debug\net8.0\MyProject.Tests.dll
MyLibrarySite -> C:\Users\mertk\OneDrive\Masadustu\SWEP\MyLibrarySite\bin\Debug\net8.0\MyLibrarySite.dll
MyProject.Tests -> C:\Users\mertk\OneDrive\Masadustu\SWEP\MyProject.Tests\bin\Debug\net8.0\MyProject.Tests.dll
C:\Users\mertk\OneDrive\Masadustu\SWEP\MyProject.Tests\bin\Debug\net8.0\MyProject.Tests.dll (.NETCoreApp,Version=v8.0) için test çalıştırması
Microsoft (R) Test Yürütmesi Komut Satırı Aracı Sürüm 17.8.0 (x64)
Microsoft (R) Test Yürütmesi Komut Satırı Aracı Sürüm 17.8.0 (x64)
Telif Hakkı (c) Microsoft Corporation. Tüm hakları saklıdır.

Test yürütmesi başlatılıyor, lütfen bekleyin...
Toplam 1 test dosyası belirlenen desenle eşleşti.

Başarılı - Başarısız: 0, Başarılı: 3, Atlanan: 0, Toplam: 3, Süre: 213 ms - MyProject.Tests.dll (net8.0)
PS C:\Users\mertk\OneDrive\Masadustu\SWEP\MyProject.Tests>
PS C:\Users\mertk\OneDrive\Masadustu\SWEP\MyProject.Tests>
PS C:\Users\mertk\OneDrive\Masadustu\SWEP\MyProject.Tests>
```

<b>Date:</b>	15.11.2024
<b>Location:</b>	Discord
<b>Duration:</b>	2 Hours
<b>Participants:</b>	Eren YÜKSEL – 200316068 Berkay ARI – 190316066 Tural MAMMADOV – 190316001 Mert KÖSA – 200316018 Batuhan DURAN – 200315087 İbrahim Ata ALIŞ - 190316014
<p>We discussed about to which project we do.</p> <p>Discussed about how to do a library site.</p>	

<b>Date:</b>	22.11.2024
<b>Location:</b>	Discord
<b>Duration:</b>	2 Hours
<b>Participants:</b>	Eren YÜKSEL – 200316068 Berkay ARI – 190316066 Tural MAMMADOV – 190316001 Mert KÖSA – 200316018 Batuhan DURAN – 200315087 İbrahim Ata ALIŞ - 190316014
Discussed about requirements and scenarios.	

<b>Date:</b>	29.11.2024
<b>Location:</b>	Discord
<b>Duration:</b>	4 Hours
<b>Participants:</b>	Eren YÜKSEL – 200316068 Berkay ARI – 190316066 Tural MAMMADOV – 190316001 Mert KÖSA – 200316018 Batuhan DURAN – 200315087 İbrahim Ata ALIŞ - 190316014
<p>Drawed all the diagrams</p> <p>Completed report writing</p>	

## Product Backlog:

- Creating a GitHub repository for the project and defining roles.
- Developing the user login and registration modules.
- Finding and developing the API for the database structure.
- Developing the home page and book search engine.
- Implementing the save book module and designing the bookshelf.
- Implementing the read book module and designing the reading page.
- Performance optimization (improving database queries and caching).
- Conducting load testing and error reporting.
- Front-end design additions (html, css).



## SPRINT 1

### SPRINT BACKLOG:

- Creating a GitHub repository for the project and defining roles.
- Developing the user login and registration modules.
- Finding and developing the API for the database structure.

SPRINT 1	DATE: 02.12.2024
Participants:	Eren YÜKSEL – 200316068 Berkay ARI – 190316066 Tural MAMMADOV – 190316001 Mert KÖSA – 200316018 Batuhan DURAN – 200315087 İbrahim Ata ALIŞ - 190316014
Berkay-> Creating a GitHub repository for the project and defining roles. Mert, Eren, Tural -> Developing the user login and registration modules. Everyone -> Finding and developing the API for the database structure.	

## SPRINT 2

### SPRINT BACKLOG:

- Developing the home page and book search engine.
- Implementing the save book module and designing the bookshelf.
- Implementing the read book module and designing the reading page.

<b>SPRINT: 2</b>	<b>DATE: 14.12.2024</b>
<b>Participants:</b>	Eren YÜKSEL – 200316068 Berkay ARI – 190316066 Tural MAMMADOV – 190316001 Mert KÖSA – 200316018 Batuhan DURAN – 200315087 İbrahim Ata ALIŞ - 190316014
Everyone -> Developing the home page and book search engine. Berkay, Mert -> Implementing the save book module and designing the bookshelf. Mert, Eren, Tural -> Implementing the read book module and designing the reading page.	

## SPRINT 3

### SPRINT BACKLOG:

- Performance optimization (improving database queries and caching).
- Conducting load testing and error reporting.
- Front-end design additions (html, css).

<b>SPRINT: 3</b>	<b>DATE: 27.12.2024</b>
<b>Participants:</b>	Eren YÜKSEL – 200316068 Berkay ARI – 190316066 Tural MAMMADOV – 190316001 Mert KÖSA – 200316018 Batuhan DURAN – 200315087 İbrahim Ata ALIŞ - 190316014
Everyone -> Performance optimization (improving database queries and caching). Mert -> Unit testing and error reporting. Batuhan, İbrahim-> Front-end design additions (html, css).	