



CE370 SOFTWARE ENGINEERING

ANALYSIS REPORT

Asst. Prof. Dr. Abdullahi Abdu Ibrahim

- Bekir ÇAKIR 210502084
- Bartın Taha BAKICI 210502090
- Hasan Hüseyin ÇELİK 210502089
- Ceylin Sena SOYER 220502128

# 1. Introduction

## 1.1 Purpose

This document serves as the analysis report for the AULibrary System, developed as a part of the CE370 Software Engineering course. It aims to define the functional and non-functional requirements of the proposed system, identify the challenges associated with traditional library operations, and demonstrate how AULibrary addresses these issues. This document acts as a formal agreement between the development team and the stakeholders, providing a comprehensive understanding of the system before the design and implementation phases.

The AULibrary Management System can be accessed through the following link:

<https://aulibrary.com.tr/Showcase/Index/>

## 1.2 Background

University libraries play a critical role in academic success by enabling access to knowledge, research materials, and information services. However, many institutions still rely on manual or outdated systems that hinder efficiency and user engagement. In response to these limitations, the AULibrary System has been designed to digitally transform the library operations at Altınbaba University, enabling streamlined processes for students, staff, and administrators alike. It is built as a web-based platform using modern software development tools and frameworks.

## 1.3 Motivation

### 1.3.1 Statement of Problems with the Existing System

The previous library processes were heavily dependent on manual tracking and paper-based records, leading to several inefficiencies:

- Users could not check their book status or borrowing history online.
- Penalty tracking for late returns was performed manually, increasing the risk of errors.
- There was no direct communication channel between library staff and users.
- Book management (cataloging, updating, deleting) required physical intervention.
- Visitors had no online access to browse or inquire about books.

### 1.3.2 The New System

AULibrary introduces a centralized, modular, and fully digital system that addresses these challenges through the following features:

- Online registration and secure login for users
- Borrowing and returning books with automated due date and penalty calculation
- Admin interface for managing books, authors, users, categories, and employees
- Publicly accessible showcase page displaying available books and project information
- Integrated announcement and messaging system for communication
- Visual dashboards for statistics and financial summaries
- Gallery and weather integration modules for extended functionality

This modern architecture ensures higher accessibility, greater control for administrators, and an improved user experience.

## 1.4 Structure of the Document

This document is organized into the following sections:

- **Functional Requirements:** Description of the system's capabilities and how different types of users interact with it.
- **Specific Requirements:** Includes system use cases, their priorities, and detailed specifications.
- **Non-functional Requirements:** Outlines performance, security, and usability constraints.
- **System Models:** Provides class diagrams, interface navigation structures, and design sketches.
- **Definitions and Glossary:** Explains technical terms, acronyms, and references used throughout the document.

## 2. Functional Requirements

### 2.1 Description of the System Functionalities

The AULibrary system is designed to fulfill the core operational needs of a university library by integrating three main modules: User Panel, Admin Panel, and Showcase Page. Each module includes a series of functionalities targeted at specific user groups.

#### Showcase Panel

The Showcase Panel is the publicly accessible interface of the Library Information System, designed to provide users with general information about the library and enable them to explore its resources without requiring authentication. This panel plays a key role in promoting transparency, user engagement, and accessibility. It is structured around several core components:

- **Dynamic Book Display:** The panel retrieves and displays a list of books directly from the database, ensuring that users always view the most current collection available in the library.
- **About Us Section:** This section presents comprehensive information about the library's mission, services, and background, offering users insight into the institution's role and objectives.
- **Contact Form:** Users can fill out a form with their name, surname, and email address to get in touch with the library. Submitted data is stored in the database and can be accessed by administrators to manage inquiries and feedback.
- **User Registration and Login:** A clearly accessible "Sign Up" button redirects users to a registration page. Returning users who log in are automatically redirected to their personalized dashboard for a more tailored experience.

These features make the Showcase Panel a vital entry point into the system, enhancing usability and supporting effective communication between the library and its users.

#### User Panel

The User Panel is a personalized interface within the Library Information System that allows registered members to interact directly with the library's services. This panel is accessible after successful login and is designed to enhance user autonomy and engagement by providing access to account-specific functionalities. The system ensures that users can manage their interactions with the library in an efficient and organized manner. Key features of the User Panel include:

- **Member Login and Registration:** Users can create an account via the registration page or log in using existing credentials. Upon login, they are directed to their personalized dashboard for immediate access to user-specific functions.
- **Book Borrowing and Returning:** Members can borrow and return books through the system. Each transaction is recorded with details such as the book name, the associated librarian, issue date, expected return date, actual return date, and any delay.
- **Borrowing History Tracking:** Users have access to a detailed history of all the books they have borrowed, including timestamps and overdue details, promoting transparency and accountability.
- **Notifications and Penalties:** The system automatically tracks due dates and generates notifications to remind users of upcoming or missed deadlines. In cases of late returns, a penalty of 1 TL per overdue day is calculated and recorded in the user's profile, contributing to the library's financial ledger.

By integrating these features, the User Panel bridges communication between the individual user and the library, providing a seamless and data-driven experience that improves both satisfaction and operational flow.

## Admin Panel

The Admin Panel is the core management interface of the Library Information System, designed exclusively for authorized personnel to oversee and control all operational, financial, and content-related aspects of the library. This panel offers a comprehensive dashboard and a wide range of administrative tools that allow real-time data management, advanced statistical monitoring, and seamless communication between system components. It plays a critical role in ensuring the sustainability, efficiency, and strategic growth of the digital library platform. Key functionalities of the Admin Panel include:

- **Dashboard Overview:** Displays real-time statistics and activity logs through interactive LINQ-based data cards. Metrics include total number of books, current borrowings, highest-read books, top-performing categories, most active members, top publishers, and the most efficient staff members.
- **CRUD Operations:** Enables full Create, Read, Update, Delete operations for books, authors, categories, members, and library staff, streamlining resource and personnel management.
- **Borrowing and Returning Management:** Provides tools to issue and receive books from members, with detailed tracking of transaction IDs, user and staff details, lending/return dates, and overdue status.
- **Penalty Calculation and Revenue Tracking:** Automatically calculates a fine of 1 TL per day for overdue returns. The amount is directly added to the user's debt and reflected in the Finance module, which includes a complete financial summary and transaction logs.
- **Weather Monitoring:** Displays a one-week weather forecast for selected cities, enhancing the utility of the admin dashboard through real-time external data integration.

- **Gallery Management:** Allows administrators to upload and manage local image files from their own system. These images are displayed on the Showcase Panel, enabling dynamic visual updates to the library's front-facing interface.
- **Statistical LINQ Cards:** Presents advanced, dynamically computed statistics such as total funds in the finance section, most borrowed books, library inventory count, total messages received, and today's borrowing activities.
- **Panel Navigation:** Includes quick access to the Showcase Panel, Settings, Help, and Logout functionalities, ensuring smooth administrative navigation and system control.

By integrating these features, the Admin Panel provides a centralized, intelligent, and responsive environment for managing all facets of the digital library infrastructure.

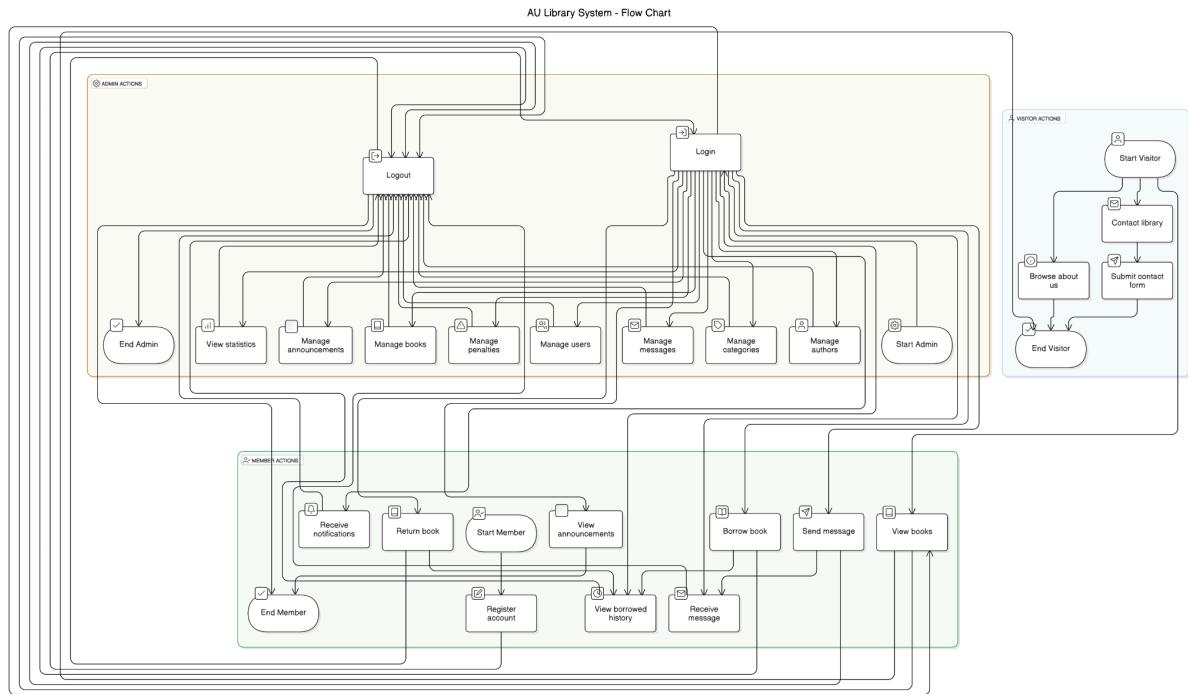
## 2.2 Description of the System Users

The AULibrary platform supports multiple user roles, each with specific access rights and interface experiences:

- **Guest (Unauthenticated Visitor):** Guests can view the Showcase Page, explore the book gallery, read about the project, and send messages via the contact form. No login is required.
- **Registered User (Library Member):** After signing up and logging in, users gain access to their personal dashboard where they can:
  - View borrowing history
  - Monitor outstanding penalties
  - Send/receive messages
  - Read announcements
  - Borrow and return books
- **Administrator (Library Staff):** Admins access an advanced panel where they can:
  - Manage users, employees, books, authors, and categories
  - Record borrowing and return operations
  - Issue announcements and view weather updates
  - Review and respond to messages
  - Manage the Showcase content
  - Monitor overall system statistics

## 2.3. Specific Requirements

### 2.3.1 Use Case Diagram



### Use Case Diagram Explanation

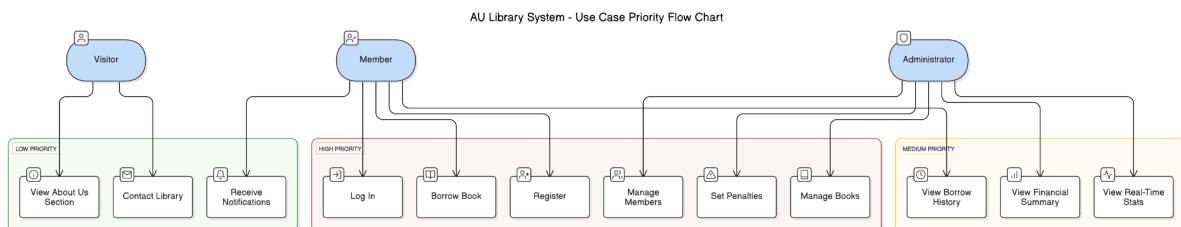
The Use Case Diagram above illustrates the interactions between different types of users (actors) and the key functionalities provided by the AU Library Management System. It identifies three main actors: Visitor, Member, and Admin. Each actor has distinct roles and permissions, allowing access to specific use cases depending on their authorization level.

- **Visitors can:**
  - View general announcements and books in the system.
  - Their actions may lead to administrative extensions like managing books, users, or announcements, performed by admins behind the scenes.
  
- **Members (registered users) are allowed to:**
  - Search for books.
  - View their own borrowed books and penalties (charges).

- The "View Borrowed Books" functionality includes access and validation from the Admin, ensuring data integrity and permission control.
- The system is also designed to extend member functions like viewing charges, by integrating backend management through administrators.
- **Admins have full access to system operations and perform the following:**
  - Manage categories, books, announcements, and users.
  - Handle system-wide message broadcasting.
  - Maintain author records and statistical reports.
  - The "Send Messages" use case is marked as an inclusion within "Manage Messages", emphasizing a modular structure that promotes code reusability.

This diagram serves as a high-level blueprint of the system's functional requirements, emphasizing clarity in user roles and the modular design of the system's architecture.

### 2.3.2 Use Case Priority List



Below is the Use Case Priority List for the Library Information System, organized by the significance of each functionality. This list serves to prioritize which features should be developed and implemented first, based on their importance for system operation.

#### High Priority

- **Register (Member):** The registration process is essential for users to gain access to all subsequent system functionalities, such as borrowing books and viewing personal accounts.
- **Log in (Member):** Login functionality enables members to access their personal information and interact with the system, making it a critical part of the user experience.
- **Borrow Book (Member):** Borrowing books is the core functionality of the library system, allowing members to interact with the library's resources effectively.
- **Manage Books (Administrator):** This feature allows administrators to add, update, and remove books from the library's catalog, essential for maintaining an up-to-date system.
- **Manage Members (Administrator):** Administrators can manage user registrations, update member information, and remove members, ensuring proper user management.
- **Set Penalties (Administrator):** Administrators can configure the penalty system for late book returns, ensuring that penalties are calculated and applied appropriately.

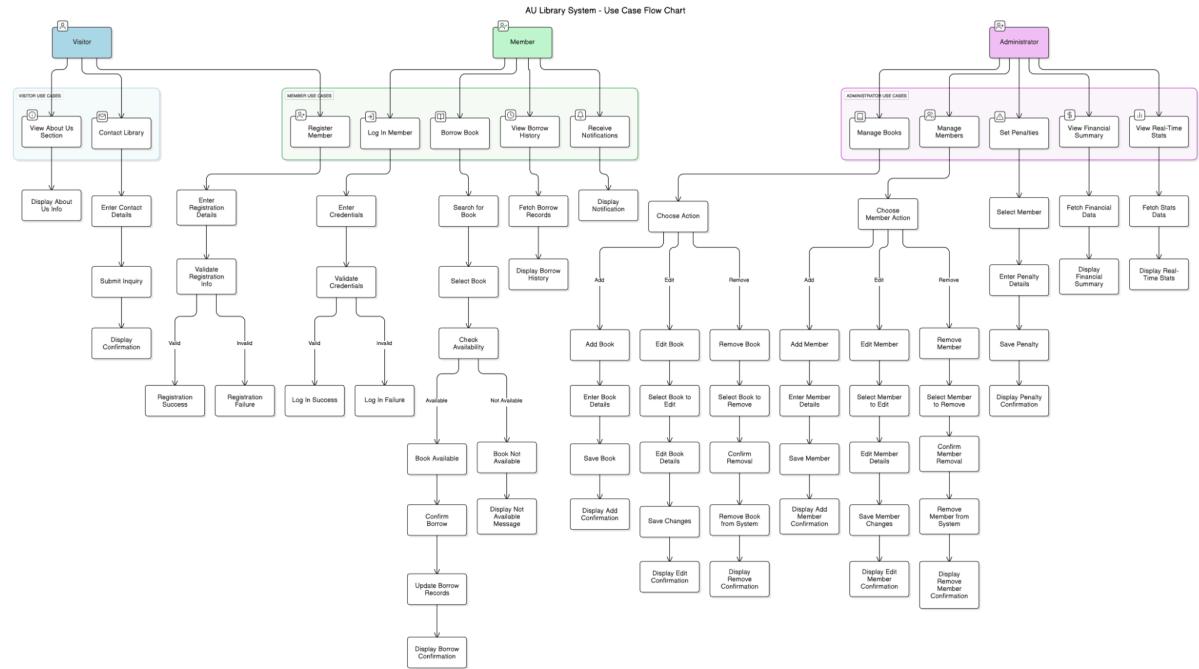
#### Medium Priority

- **View Borrow History (Member):** While not a critical feature, it enhances user experience by allowing members to track their borrowing history.
- **View Financial Summary (Administrator):** Administrators can view financial summaries, such as revenue from penalties and book sales, to assess the financial health of the library.
- **View Real-Time Stats (Administrator):** This functionality allows administrators to monitor key statistics, like active users and current book borrowing data, though it is not essential to the system's core operation.

## Low Priority

- **View "About Us" Section (Visitor):** A helpful but non-essential feature, allowing visitors to learn about the library's services and mission.
- **Contact Library (Visitor):** Allows visitors to contact the library, although this is more of an additional convenience than a core system feature.
- **Receive Notifications (Member):** Notifications about due dates, penalties, and other library-related updates, although important, are less critical compared to core functionalities like borrowing books.

### 2.3.3 Use Case Specifications



#### Register (Member)

The registration process enables visitors to become members of the library system. It allows new users to access library services such as borrowing books and tracking their activities.

- **Inputs:**
  - Full name (first and last)
  - Email address
  - Password
  - Contact details (e.g., phone number, address)
- **Process Flow:**
  - A visitor navigates to the "Sign Up" page.
  - The visitor provides personal details and selects a password.
  - The system checks if the email is already registered and ensures data integrity.
  - Upon verification, the system creates a new member profile and stores the information in the database.
  - A confirmation email is sent to the user, verifying their registration.
- **Outputs:**
  - The member's profile is created in the system.
  - A confirmation email is sent to the member's email address.
  - The user is directed to the login page to proceed with the next steps.

### **Log In (Member)**

The login functionality allows members to access their personalized accounts and engage with the library system's various services, such as borrowing books and viewing their borrowing history.

- **Inputs:**
  - Username (email)
  - Password
- **Process Flow:**
  - The member navigates to the login page.
  - The member enters their registered email and password.
  - The system validates the credentials.
  - If successful, the member is redirected to the user panel, where they can access their personal information and interact with available services.
- **Outputs:**
  - The member gains access to their personal dashboard.
  - If login fails, an error message is displayed prompting the user to correct their details.

### **Borrow Book (Member)**

This feature allows registered members to borrow books from the library. It is a key operation in the system as it enables members to interact with library resources.

- **Inputs:**
  - Book title or identifier (e.g., ISBN)
  - Member ID

- Borrow date
- **Process Flow:**
  - The member browses the available books on the system and selects the book they wish to borrow.
  - The system checks the availability of the selected book.
  - If the book is available, the member proceeds with the borrowing process.
  - The system assigns a due date and records the transaction in the member's borrowing history.
  - The system sends a notification to the member confirming the successful borrowing of the book and details about the return date.
- **Outputs:**
  - The book's status is updated to "borrowed."
  - A confirmation notification is sent to the member.
  - The borrowing history is updated.

## Manage Books (Administrator)

This functionality allows administrators to maintain the library's catalog by adding new books, editing book details, or removing outdated books. It is crucial to ensure that the library's collection remains current.

- **Inputs:**
  - Book details (title, author, ISBN, category, availability)
  - Action (add, update, delete)
- **Process Flow:**
  - The administrator logs into the admin panel.
  - The administrator selects whether to add a new book, update an existing book, or remove a book from the system.
  - The system validates the entered information (e.g., unique ISBN) and processes the action.
  - If a book is removed, it is deleted from the catalog, and its availability status is updated.
- **Outputs:**
  - The catalog is updated with the new or edited book information.
  - A confirmation of the action (add/update/delete) is displayed.

## Manage Members (Administrator)

Administrators can manage the membership data, including registering new members, updating member profiles, and removing inactive or non-compliant members. This is critical for maintaining a user-friendly and organized library system.

- **Inputs:**
  - Member information (e.g., name, email, membership status)
  - Action (add, update, delete)
- **Process Flow:**
  - The administrator accesses the "Manage Members" section.
  - The administrator can add new members, update existing member details, or remove members from the system.
  - The system verifies the entered information for consistency and stores the changes.
- **Outputs:**
  - The member list is updated, reflecting the new, updated, or deleted members.
  - A confirmation message is displayed for each successful action.

### **Penalties (Administrator)**

Administrators can configure penalties for late book returns, ensuring that overdue fines are calculated and applied correctly, which is essential for maintaining library operations.

- **Inputs:**
  - Penalty rate (e.g., 1 TL per day)
  - Duration of the overdue period
- **Process Flow:**
  - The administrator navigates to the "Penalty Management" section.
  - The administrator sets or updates the penalty rate (e.g., 1 TL/day).
  - The system automatically applies the penalty for overdue books, updating member accounts accordingly.
- **Outputs:**
  - The penalty system is updated with the new rate.
  - The system applies penalties to overdue books and updates the member's debt record.

### **View Borrow History (Member)**

This feature enables members to view a history of the books they have borrowed, promoting transparency and helping members manage their borrowing activities.

- **Inputs:**
  - Member ID
- **Process Flow:**
  - The member accesses their account dashboard.
  - The system fetches and displays the history of borrowed books.
  - The member can view details such as borrowing dates, return dates, and penalties if applicable.
- **Outputs:**
  - A detailed list of the books borrowed by the member, along with their statuses.

## **View Financial Summary (Administrator)**

This functionality allows administrators to view a summary of the library's financial status, including revenue generated from fines and book sales.

- **Inputs:**
  - Financial data (penalty revenues, book sale revenues)
- **Process Flow:**
  - The administrator navigates to the "Finance" section.
  - The system generates a financial summary based on the available data.
  - The administrator can analyze and download the report as necessary.
- **Outputs:**
  - A financial summary report showing the revenue from penalties and other library operations.

## **View Real-Time Stats (Administrator)**

This feature allows administrators to monitor key statistics in real time, such as active users, current book borrowings, and penalties, aiding in quick decision-making.

- **Inputs:**
  - Data on active users, borrowed books, penalties
- **Process Flow:**
  - The administrator logs into the admin panel and accesses the "Statistics" dashboard.
  - The system dynamically updates the statistics in real time.
  - The administrator reviews the real-time data for insights.
- **Outputs:**
  - Visual representation of key statistics on a dashboard.

## **View "About Us" Section (Visitor)**

This feature provides information about the library, including its mission, services, and history, offering useful insights for visitors.

- **Inputs:**
  - None (view-only)
- **Process Flow:**
  - A visitor navigates to the "About Us" section.
  - The section displays pre-configured content about the library.
- **Outputs:**
  - Text content providing information about the library.

### Contact Library (Visitor)

Allows visitors to contact the library for inquiries or support, adding a layer of user interaction with the library.

- **Inputs:**
  - Name, email, and message from the visitor
- **Process Flow:**
  - The visitor fills in the contact form on the "Contact Us" page.
  - The system stores the contact message in the database and notifies the administrator.
- **Outputs:**
  - A notification is sent to the administrator with the visitor's message.

### Receive Notifications (Member)

This feature sends notifications to members regarding due dates, overdue penalties, and system updates, improving user engagement.

- **Inputs:**
  - Notification settings (due date reminders, penalty warnings)
- **Process Flow:**
  - The system checks for upcoming due dates and overdue books.
  - The system sends automated notifications to the member based on their preferences.
- **Outputs:**
  - A notification is delivered to the member's inbox or via push notification.

## 3. Non-functional Requirements

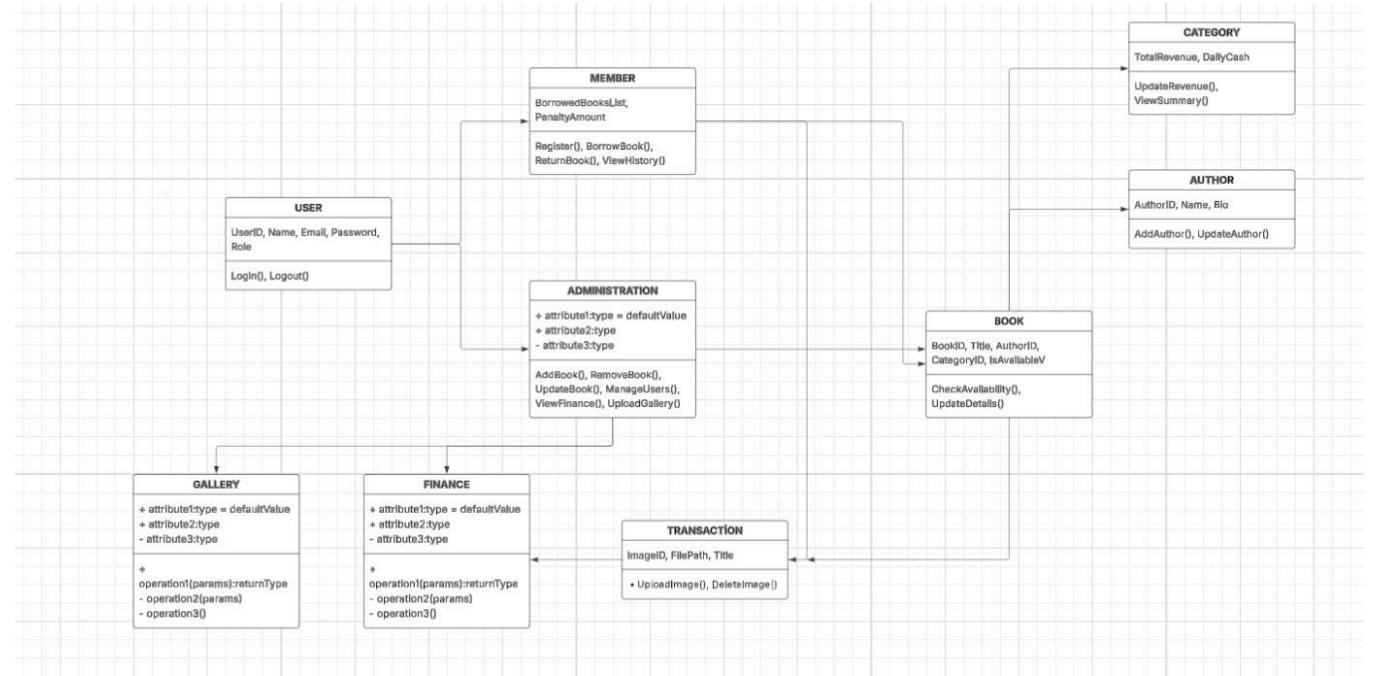
### 3.1 Volere Template for AULibrary System

Requirement Type	Description
Product Performance	The system should respond to user actions (clicks, queries) within 2 seconds under normal load.
Reliability	The system must operate with at least 99.5% uptime excluding maintenance windows.

<b>Security</b>	<p>All passwords must be encrypted in the database.</p> <p>Role-based access control must be enforced for Admin, User, and Guest roles.</p> <p>Only authenticated users can access personal and transactional data.</p>
<b>Usability</b>	<p>The interface should be intuitive and require less than 15 minutes of training for a new user.</p> <p>All buttons, links, and navigation must be responsive and mobile-friendly.</p>
<b>Scalability</b>	<p>The system should support at least 1,000 concurrent users without performance degradation.</p>
<b>Maintainability</b>	<p>The system should be developed with a modular architecture to ensure that individual components (e.g., Book, User, Message) can be updated or replaced independently.</p>
<b>Availability</b>	<p>The system should be accessible 24/7, with automatic backups every 24 hours.</p>
<b>Portability</b>	<p>The system should be deployable on any server that supports ASP.NET MVC5 and MS SQL Server.</p>
<b>Auditability</b>	<p>All transactions (borrowing, returns, user updates) should be logged with timestamps and retrievable by the admin.</p>
<b>Localization</b>	<p>The system interface must be prepared for future multilingual support (English, Turkish).</p>
<b>Compliance</b>	<p>The system should follow university data privacy policies and standard web security practices (e.g., HTTPS).</p>

## 4. System Models

### 4.1 Object and Class Model



### Core Classes and Descriptions

- **User**
  - *Attributes:* UserID, FullName, Email, Password, RegistrationDate
  - *Relationships:* One-to-many with BorrowRecord, Message
- **Admin**
  - *Attributes:* AdminID, Name, Email, Password
  - *Role:* System-level access and CRUD operations
- **Book**
  - *Attributes:* BookID, Title, ISBN, CategoryID, AuthorID, Image, Status
  - *Relationships:* One-to-many with BorrowRecord
- **Author**
  - *Attributes:* AuthorID, Name, Biography
  - *Relationships:* One-to-many with Book
- **Category**
  - *Attributes:* CategoryID, Name
  - *Relationships:* One-to-many with Book
- **BorrowRecord**
  - *Attributes:* RecordID, UserID, BookID, BorrowDate, DueDate, ReturnDate, Penalty
  - *Relationships:* Many-to-one with Book and User
- **Message**
  - *Attributes:* MessageID, SenderID, ReceiverID, Subject, Body, Date
  - *Relationships:* Many-to-one with User and Admin
- **Announcement**
  - *Attributes:* AnnouncementID, Title, Content, DatePosted, Viewable by all users.

- **WeatherReport**
  - *Attributes:* ReportID, Temperature, Description, Date
- **GalleryImage**
  - *Attributes:* ImageID, ImagePath, UploadDate
  - *Relationships:* One User can have many BorrowRecords and Messages. One Book can only be borrowed by one User at a time. One Admin can post many announcements. Books are categorized under one Category and have one Author.

## 4.2 User Interface – Navigational Paths and Screen Mock-ups

### Navigation Paths

The AULibrary system is composed of three main interface flows: the public-facing Showcase Page, the User Panel, and the Admin Panel. Each flow is designed with clean navigation and consistent UI elements.

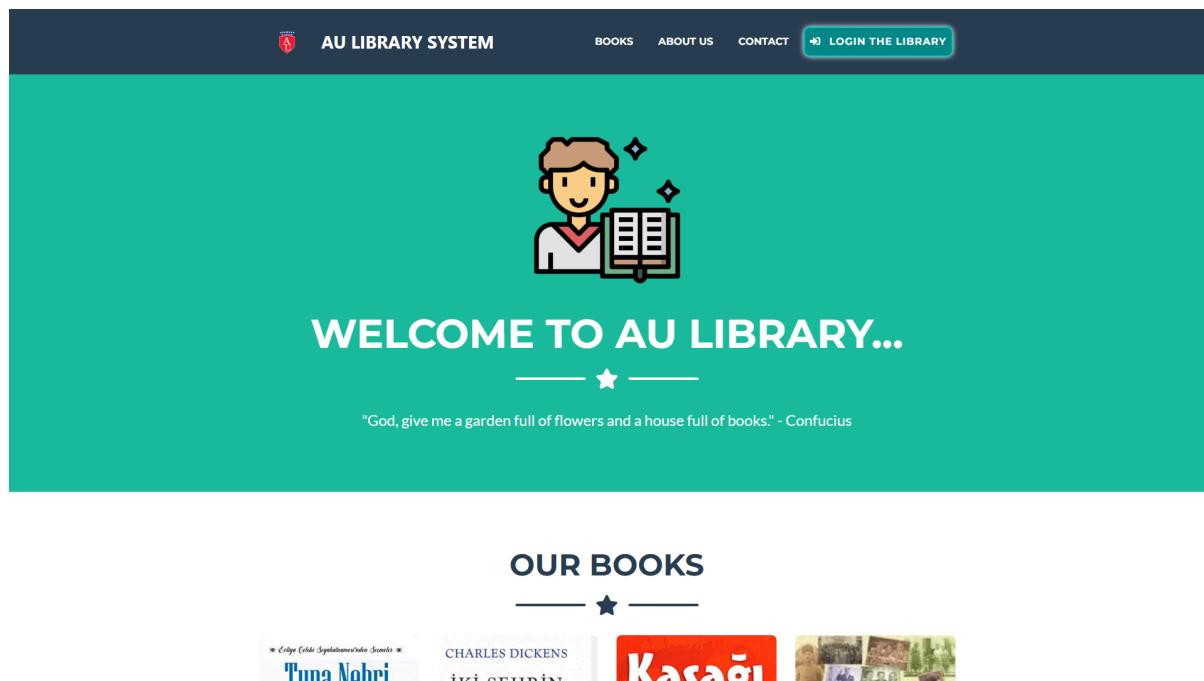
- Visitor (Showcase) Navigation Paths:
  - Home → Showcase → Book List
  - Home → Showcase → About Us
  - Home → Showcase → Contact Us → Submit Form
  - Home → Showcase → Register
  - Home → Showcase → Login
  - Home → Showcase → Weather Forecast
- Member Navigation Paths:
  - Login → Member Dashboard → Profile
  - Login → Member Dashboard → Borrow Book
  - Login → Member Dashboard → Return Book
  - Login → Member Dashboard → Borrowing History
  - Login → Member Dashboard → View Notifications
  - Login → Member Dashboard → Logout
- Admin Navigation Paths:
  - Admin Login → Dashboard → Books → Add / Edit / Delete
  - Admin Login → Dashboard → Authors → Add / Edit / Delete
  - Admin Login → Dashboard → Categories → Add / Edit / Delete
  - Admin Login → Dashboard → Members → Manage / Delete / View Details
  - Admin Login → Dashboard → Staff → Add / Remove / View
  - Admin Login → Dashboard → Transactions → Borrow Book
  - Admin Login → Dashboard → Transactions → Return Book
  - Admin Login → Dashboard → Statistics (Linq Cards)
  - Admin Login → Dashboard → Financial Overview → Cash Register (Kasa)

- Admin Login → Dashboard → Financial Overview → Logs
- Admin Login → Dashboard → Gallery → Upload Local Files
- Admin Login → Dashboard → Settings
- Admin Login → Dashboard → Help
- Admin Login → Dashboard → Logout

- Additional Functional Paths (System-Wide):
  - Dashboard → Real-Time Statistics
  - Dashboard → Most Read Book
  - Dashboard → Most Active Member
  - Dashboard → Best Publisher
  - Dashboard → Most Borrowed Author
  - Dashboard → Today's Loans
  - Dashboard → Total Messages
  - Dashboard → Category Count
  - Dashboard → Total Book Count
  - Dashboard → Currently Borrowed Books
  - Dashboard → View All Penalties / Debts

## Screen Mockups



AU LIBRARY SYSTEM
BOOKS
ABOUT US
CONTACT
LOGIN THE LIBRARY

CHARLES DICKENS  
İKİ ŞEHİRİN HİKAYESİ
Kaşağı  
ÖMER SEYFETTİN
geçtiğim günlerden  
Hasan Ali Yıldız

John Agard  
KİTAP
COMPUTER ENGINEERING  
Edited by Alan Poole
My Name is The Sun  
By Adrienne Body
HENRY DAVID THOREAU  
Yalnızlık

STEVE KLEIN

İSTANBUL  
A U
Altınbaba University Library
İSTANBUL  
A U

USER MAIL
PASSWORD

Forgot Password?
Remember Me

LOGIN

ADMIN PANEL

Don't have an account? [Register](#)
← Return To Showcase

© CE370 Software Engineering | Design by BÇ-CSS-HHQ-BTB


**AU Library**

[Menu](#)

[About Us](#)
[Settings ▾](#)

### Library User Profile Information

**Mail Address:** beathabestofficial@gmail.com

ID	24
NAME	Bartin Taha
SURNAME	Bakıcı
USERNAME	BeaTahaBest
PASSWORD	BARTIN_2001
SCHOOL	Altınbaba University
PHOTO	
TELEPHONE	05313373826

[Update Your Information](#)

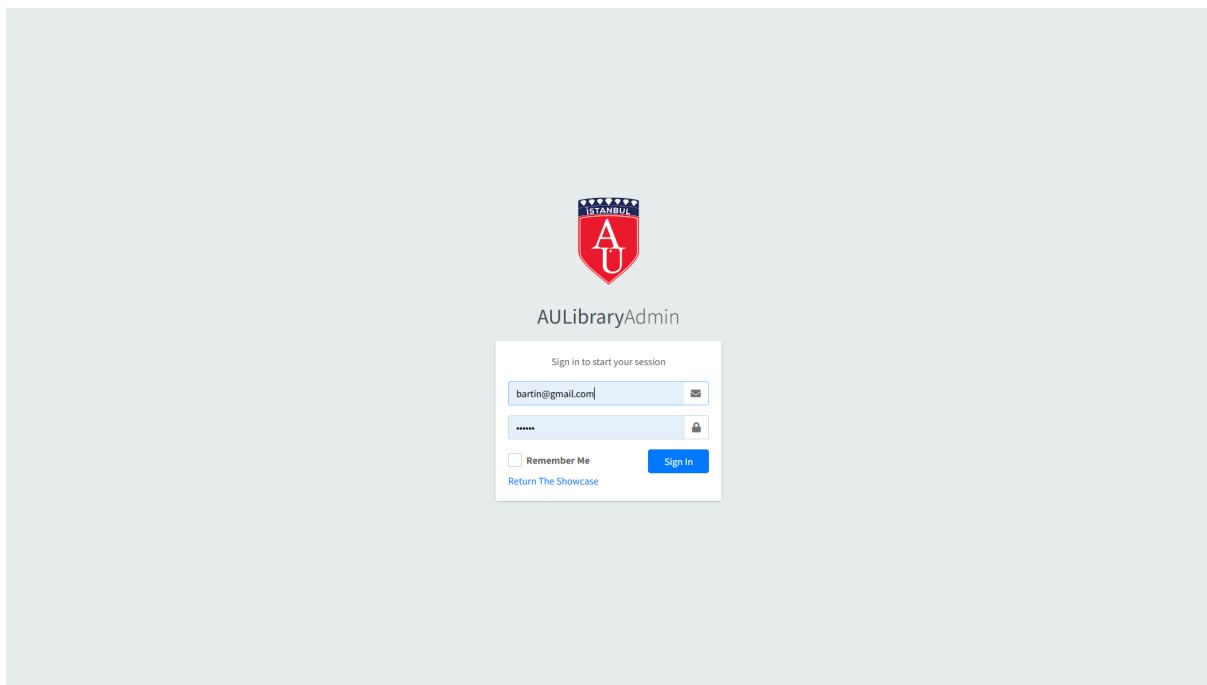

**AU Library**

[Menu](#)

[About Us](#)
[Settings ▾](#)

### Announcements

Category	Date	Content
New Book	18.04.2025	You can browse our new books in the book list and borrow them.
Don't Delay Returning Books	17.04.2025	Please return the books you borrowed on time, or you may be subject to a penalty.
Contact	16.04.2025	You can contact us through the "Contact" section on our website.
Penalty Fee	15.04.2025	Please make sure to pay any penalty fees within the semester.
Online Books	14.04.2025	Our online library will be available to you soon.



Altınbaba University Library

AU Library

Categories

Category ID	Category Name	Delete	Update
1	NOVEL	<a href="#">Delete</a>	<a href="#">Update</a>
2	ADVENTURE	<a href="#">Delete</a>	<a href="#">Update</a>
3	SCIENCE FICTION	<a href="#">Delete</a>	<a href="#">Update</a>
4	FANTASTIC	<a href="#">Delete</a>	<a href="#">Update</a>
5	POEM	<a href="#">Delete</a>	<a href="#">Update</a>
6	THEATRE	<a href="#">Delete</a>	<a href="#">Update</a>
7	CHILDREN'S BOOKS	<a href="#">Delete</a>	<a href="#">Update</a>
8	RUSSIAN CLASSICS	<a href="#">Delete</a>	<a href="#">Update</a>
9	STORIES	<a href="#">Delete</a>	<a href="#">Update</a>
10	COMEDY	<a href="#">Delete</a>	<a href="#">Update</a>
11	PERSONAL DEVELOPMENT	<a href="#">Delete</a>	<a href="#">Update</a>
12	BIOGRAPHY	<a href="#">Delete</a>	<a href="#">Update</a>
30	FUTURE	<a href="#">Delete</a>	<a href="#">Update</a>

[Add New Category](#)



Altınbaba University Library

AU Library

Book Page

Enter the book name to search:

ID	Name	Category	Author	Year of Publication	Publishing House	PAGE	Situation	Delete	Update
1	Tuna Nehri Buz Tutunca	Jules Verne	NOVEL	2015	Cicekk	250	True	<a href="#">Delete</a>	<a href="#">Update</a>
2	İki Şehrin Hikayesi	Charles Diskens	NOVEL	2018	Gunes	350	True	<a href="#">Delete</a>	<a href="#">Update</a>
3	Kaşağı	Ömer Seyfettin	RUSSIAN CLASSICS	2012	Yildiz	63	True	<a href="#">Delete</a>	<a href="#">Update</a>
5	Geçtiğim Günlerden	Hasan Ali Yücel	BIOGRAPHY	2011	Ay	210	True	<a href="#">Delete</a>	<a href="#">Update</a>
6	Kitap	Jules Verne	NOVEL	2018	Gunes	63	True	<a href="#">Delete</a>	<a href="#">Update</a>
7	TÜL	Jules Verne	NOVEL	2012	Ay	210	True	<a href="#">Delete</a>	<a href="#">Update</a>
9	My Name Is TheSun	Gogol White	SCIENCE FICTION	1990	Butter	120	True	<a href="#">Delete</a>	<a href="#">Update</a>
11	Yalnız	Ömer Seyfettin	STORIES	1997	Baş	190	True	<a href="#">Delete</a>	<a href="#">Update</a>
12	Ömer'in Çocukluğu	Jules Verne	NOVEL	2012	Yildiz	63	True	<a href="#">Delete</a>	<a href="#">Update</a>
13	Sihir Dükkanı	Jules Verne	NOVEL	2018	Baş	63	True	<a href="#">Delete</a>	<a href="#">Update</a>
21	Software Engineering	Gogol White	PERSONAL DEVELOPMENT	2010	Moon	690	True	<a href="#">Delete</a>	<a href="#">Update</a>
26	Don't Make Me Think	Steve Krug	NOVEL	2015	Pegasus	250	True	<a href="#">Delete</a>	<a href="#">Update</a>

[Add New Book](#)





Altinbas  
University  
Library

AU Library

Employees
Add New Employee

EMPLOYEE ID	EMPLOYEE NAME	DELETE	UPDATE
1	Bekir Çakır	<button style="background-color: #e67e22; color: white; border: none; padding: 2px 5px;">Delete</button>	<button style="background-color: #28a745; color: white; border: none; padding: 2px 5px;">Update</button>
2	Bartin Taha Bakıcı	<button style="background-color: #e67e22; color: white; border: none; padding: 2px 5px;">Delete</button>	<button style="background-color: #28a745; color: white; border: none; padding: 2px 5px;">Update</button>
3	Hasan Hüseyin Çelik	<button style="background-color: #e67e22; color: white; border: none; padding: 2px 5px;">Delete</button>	<button style="background-color: #28a745; color: white; border: none; padding: 2px 5px;">Update</button>
4	Ceylin Sena Soyer	<button style="background-color: #e67e22; color: white; border: none; padding: 2px 5px;">Delete</button>	<button style="background-color: #28a745; color: white; border: none; padding: 2px 5px;">Update</button>
22	Ali Şafak	<button style="background-color: #e67e22; color: white; border: none; padding: 2px 5px;">Delete</button>	<button style="background-color: #28a745; color: white; border: none; padding: 2px 5px;">Update</button>



Altinbas  
University  
Library

AU Library

Borrow Book
Lend the Book

User ID
Ahmet Aslan

Book ID
Tuna Nehri Buz Tutunca

Employee ID
Bartin Taha Bakıcı

Purchase Date
23.04.2025

Return Date
30.04.2025

Lend the Book

List of Returned Books							AU Library
Process ID	Book Name	Member	Employee	Purchase Date	User Return Date	Details	
5	Kaşığı	Bekir Çakır	Hasan Hüseyin Çelik	08.04.2025	09.04.2025	<a href="#">Details</a>	
6	TÜL	Zeynep Şeker	Ceylin Sena Soyer	08.04.2025	09.04.2025	<a href="#">Details</a>	
11	Tuna Nehri Buz Tutunca	Bekir Çakır	Bekir Çakır	09.04.2025	09.04.2025	<a href="#">Details</a>	
21	Geçtiğim Günlerden	Bekir Çakır	Bartın Taha Bakıcı	09.04.2025	09.04.2025	<a href="#">Details</a>	
22	Tuna Nehri Buz Tutunca	Bekir Çakır	Bekir Çakır	09.04.2025	09.04.2025	<a href="#">Details</a>	
23	Tuna Nehri Buz Tutunca	Ahmet Aslan	Bekir Çakır	09.04.2025	09.04.2025	<a href="#">Details</a>	
24	Tuna Nehri Buz Tutunca	Emel Sunal	Bekir Çakır	09.04.2025	09.04.2025	<a href="#">Details</a>	
25	Tuna Nehri Buz Tutunca	Zeynep Şeker	Bekir Çakır	09.04.2025	09.04.2025	<a href="#">Details</a>	
26	Kitap	Tanık Kemal	Bekir Çakır	09.04.2025	15.04.2025	<a href="#">Details</a>	
27	Tuna Nehri Buz Tutunca	Ahmet Aslan	Bekir Çakır	09.04.2025	09.04.2025	<a href="#">Details</a>	
28	Sihir Dükkanı	Berk Yılmaz	Bekir Çakır	09.04.2025	09.04.2025	<a href="#">Details</a>	
29	Tuna Nehri Buz Tutunca	Ahmet Aslan	Bekir Çakır	10.04.2025	10.04.2025	<a href="#">Details</a>	
30	Tuna Nehri Buz Tutunca	Ahmet Aslan	Bekir Çakır	11.04.2025	11.04.2025	<a href="#">Details</a>	
31	Tuna Nehri Buz Tutunca	Ahmet Aslan	Bekir Çakır	13.04.2025	13.04.2025	<a href="#">Details</a>	
32	Tuna Nehri Buz Tutunca	Ahmet Aslan	Bekir Çakır	13.04.2025	15.04.2025	<a href="#">Details</a>	
33	Tuna Nehri Buz Tutunca	Ahmet Aslan	Bekir Çakır	14.04.2025	14.04.2025	<a href="#">Details</a>	
34	Tuna Nehri Buz Tutunca	Bekir Çakır	Bekir Çakır	14.04.2025	14.04.2025	<a href="#">Details</a>	
35	Tuna Nehri Buz Tutunca	Zeynep Şeker	Bekir Çakır	14.04.2025	14.04.2025	<a href="#">Details</a>	
36	İki Şehrin Hikayesi	Ahmet Aslan	Bekir Çakır	14.04.2025	15.04.2025	<a href="#">Details</a>	

User Punishments							AU Library
ID	User	Start	Expiry	Money	Action	Delete	
3	1	10.04.2025	24.04.2025	7,00	36	<a href="#">Delete</a>	
4	4	12.04.2025	26.04.2025	7,00	36	<a href="#">Delete</a>	
8	2	18.04.2025	03.05.2025	8,00	38	<a href="#">Delete</a>	
9	1	19.04.2025	04.05.2025	8,00	37	<a href="#">Delete</a>	
23	19	18.04.2025	03.05.2025	15,00	47	<a href="#">Delete</a>	
25	11	01.01.0001	01.01.0001	0,00	48	<a href="#">Delete</a>	

**Altınbaba University Library**

**Statistics Page**

**Home Page**

**Categories**

**Books**

**Authors**

**Users**

**Employees**

**Borrow Book**

**Return Book**

**Actions**

**Late Return Charges**

**Statistics**

**Announcements**

**AU Weather Report**

**AU Weather Cards**

**Gallery**

**Showcase**

**Messages**

**Admin Setting**

**Help**

**Exit**

**\$ 45,00 TL**  
Amount in the case

**23**  
Total Users

**12**  
Number of Books

**0**  
Books Purchased

Month	Books Borrowed
January	60
February	55
March	85
April	78
May	92
June	48
July	98

**Daily Admin Checklist**

- Call for publisher.
- Check the books received.
- Check fines and payments.
- Check out the book lists.
- Enter new books into the system.

**World Map**

Legend: 40% (Dark Blue), 12% (Medium Blue), 11% (Light Blue), 10% (Yellow), 18% (White)

**System Reports**

Report	Value
Satisfaction	96%
Reliability	87%
Weekly Feedback	76%

Library users' satisfaction, reliability and weekly feedback scores are checked in the admin panel and evaluated every day.

Announcements				
Announcement ID	Category	Date	DELETE	Content
1	New Book	18.04.2025	<button>Delete</button>	<button>Detail</button>
2	Don't Delay Returning Books	17.04.2025	<button>Delete</button>	<button>Detail</button>
3	Contact	16.04.2025	<button>Delete</button>	<button>Detail</button>
4	Penalty Fee	15.04.2025	<button>Delete</button>	<button>Detail</button>
5	Online Books	14.04.2025	<button>Delete</button>	<button>Detail</button>

[Add New Announcement](#)

[Actions](#)

[Statistics](#)

[Announcements](#)

[AU Weather Report](#)

[AU Weather Cards](#)

[Gallery](#)

[Showcase](#)

[Showcase Messages](#)

[Admin Setting](#)

[Help](#)

[Exit](#)

**AU Istanbul Weather**

14°C Ankara Broken cloud sky  
1.5 m/s 78% 758 mmHg

23 Apr 18°C 24 Apr 17°C 25 Apr 19°C 26 Apr 20°C 27 Apr 15°C 28 Apr 14°C 29 Apr 13°C

Istanbul 23 Apr 18°C 24 Apr 17°C 25 Apr 19°C 26 Apr 20°C 27 Apr 15°C 28 Apr 14°C 29 Apr 13°C

ALTINBAS UNIVERSITESI

Statistics Announcements AU Weather Report AU Weather Cards

Gallery Showcase Showcase Messages

Admin Setting Help Exit

**AU Weather Report**

Istanbul, Turkey 23.04.2025

19 °C

3 °C

WED 17 °C

THU 16 °C

FRI 16 °C

AU Weather Report

Statistics Announcements AU Weather Report AU Weather Cards

Gallery Showcase Showcase Messages

Admin Setting Help Exit

<https://aulibrary.com.tr/Statistics/WeatherCard/>

## 5. Definitions, Acronyms and Abbreviations

Term / Acronym	Definition
<b>AULibrary</b>	The name of the web-based library automation system developed for Altınbaş University.
<b>MVC</b>	<i>Model-View-Controller</i> : A software architectural pattern used to separate concerns in application structure.
<b>CRUD</b>	<i>Create, Read, Update, Delete</i> : Basic operations for managing data in a database.
<b>LINQ</b>	<i>Language Integrated Query</i> : A Microsoft .NET component that adds native data querying capabilities.
<b>EF (Entity Framework)</b>	An Object-Relational Mapper (ORM) for .NET that allows developers to work with databases using .NET objects.
<b>SQL</b>	<i>Structured Query Language</i> : A standardized programming language for managing relational databases.
<b>ASP.NET MVC5</b>	A framework developed by Microsoft for building web applications using the MVC pattern in .NET.
<b>Admin Panel</b>	The backend control interface accessible only by administrators to manage users, books, categories, etc.
<b>User Panel</b>	The interface available to registered users for managing personal information, borrowing books, and tracking activity.

<b>Penalty</b>	A monetary fine applied to a user account when a borrowed book is returned past its due date.
<b>Borrow Record</b>	A record of a transaction where a user borrows a book, including borrow and return dates.
<b>Weather API</b>	An external interface used to fetch and display real-time weather data within the Admin Panel.
<b>Checklist</b>	A task management tool within the Admin Panel to help staff follow daily or weekly operational procedures.
<b>Mock-up</b>	A visual prototype or static representation of a user interface used for design and planning.
<b>UML</b>	<i>Unified Modeling Language</i> : A standardized modeling language for specifying, visualizing, and documenting software system design.
<b>Session</b>	A temporary user interaction period with the system that ends when the user logs out or the session times out.
<b>Bootstrap</b>	An open-source CSS framework used for responsive and mobile-first front-end web development.
<b>JS (JavaScript)</b>	A scripting language used to create dynamic and interactive web pages.

## 6. Glossary & References

### 6.1 Glossary

Term	Explanation
<b>Borrowing</b>	The process of a registered user taking a book from the library for a specific period.
<b>Penalty Fee</b>	A daily fine imposed when a user fails to return a borrowed book on time.
<b>Admin</b>	A privileged user with full access to all system modules, including book, user, and transaction management.
<b>Announcement</b>	A message or notification created by an admin to inform users about news or updates.
<b>Contact Form</b>	A publicly accessible form that allows visitors to send messages to the admin through the Showcase page.
<b>Weather Report</b>	A section in the admin panel used to display weather updates for informational purposes.
<b>Gallery</b>	A module where administrators can upload and display images related to the library.
<b>Checklist</b>	A built-in tool for admins to manage and track operational tasks.
<b>Dashboard</b>	The main landing page after login for both users and admins, showing relevant functions and statistics.

## 6.2 References

1. **ASP.NET MVC5 Official Documentation**  
<https://learn.microsoft.com/en-us/aspnet/mvc/overview>
2. **Microsoft Entity Framework Documentation**  
<https://learn.microsoft.com/en-us/ef/>
3. **Bootstrap CSS Framework**  
<https://getbootstrap.com/>
4. **OpenWeatherMap API (Used for Weather Data Integration)**  
<https://openweathermap.org/api>
5. **C# Programming Guide by Microsoft**  
<https://learn.microsoft.com/en-us/dotnet/csharp/>
6. **Visual Studio Community 2022 – Development Environment**  
<https://visualstudio.microsoft.com/>
7. **CE370 Software Engineering Course Slides & Lecture Notes**  
**(Altınpaş University, Spring 2025)**



CE370 SOFTWARE ENGINEERING

DESIGN REPORT

Asst. Prof. Dr. Abdullahi Abdu Ibrahim

• Bekir ÇAKIR	210502084
• Bartın Taha BAKICI	210502090
• Hasan Hüseyin ÇELİK	210502089
• Ceylin Sena SOYER	220502128

# 1. Introduction

## 1.1 Purpose of the Document

This design document provides a comprehensive technical description of the AU Library Management System (AULibrary). It serves as a bridge between the analysis report and the implementation phase, ensuring that all design components are clearly defined for the development team. The document outlines the system architecture, component-level designs, class diagrams, user interface structures, and the data model, allowing developers to implement the system efficiently and consistently.

The AULibrary Management System can be accessed through the following link:

<https://aulibrary.com.tr/Showcase/Index/>

## 1.2 Purpose of the System

The AULibrary System is a web-based library management platform developed for the course CE370 Software Engineering. Its primary purpose is to digitalize and optimize the operations of a university library—particularly Altınbaş University. It replaces the traditional, manual book management processes with an interactive, user-friendly, and role-based system.

The system serves three main types of users:

- **Visitor:** Can browse the public-facing showcase, view announcements, access the "About Us" section, and send contact messages.
- **Member (User):** Can register, log in, borrow and return books, view borrowing history and penalty charges, and receive system notifications.
- **Administrator:** Has access to the admin dashboard to manage books, users, employees, authors, categories, weather info, statistics, announcements, gallery, and contact messages.

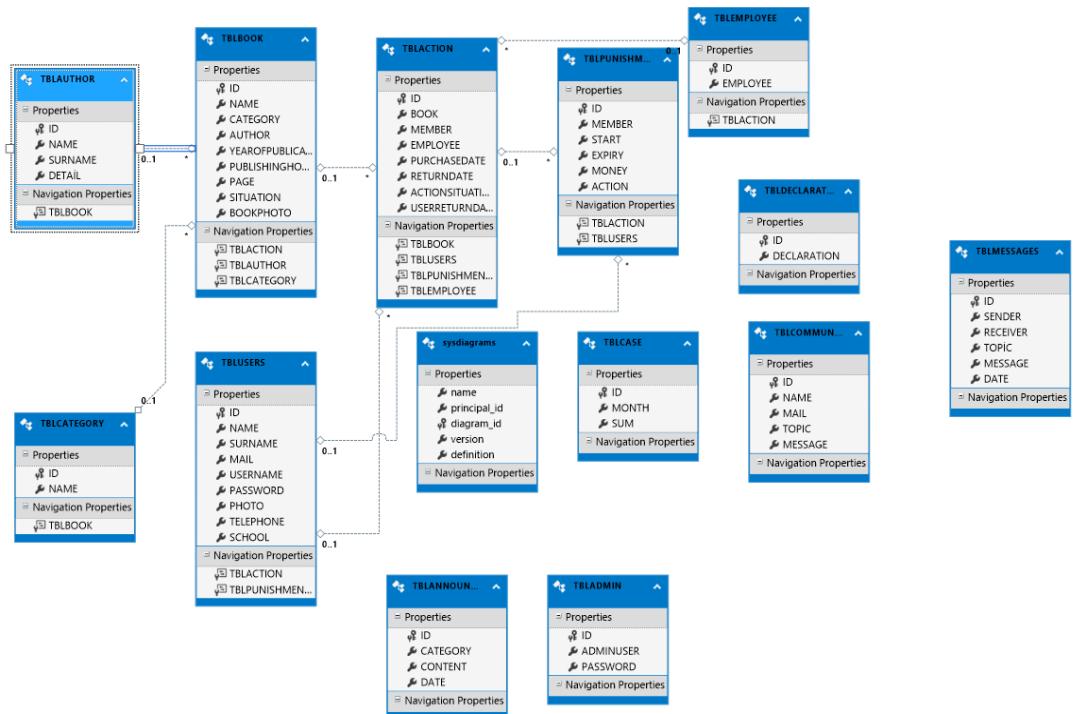
### Key Features Provided:

- Modular login panels based on user roles (Admin, Member)
- Real-time borrowing/returning system with penalty tracking
- Dynamic statistics dashboard for administrators
- Interactive weather and gallery modules
- Public showcase for book browsing and communication

## 1.3 Structure of the Document

This design report is systematically structured to provide a clear and complete understanding of the AU Library Management System's architecture and internal design. It begins with the System Architecture section, which outlines the general structure of the application, emphasizing the layered MVC model, data flow between components, and the technologies used such as ASP.NET MVC5, C#, Entity Framework, and MS SQL Server. The Component Design part breaks the system into major functional modules including User Management, Book Management, Borrow/Return, and Communication Services, explaining the purpose and interactions of each. The Database Design section presents the entity-relationship model of the system, showing how data is structured, which fields are primary or foreign keys, and how data consistency is ensured. The User Interface Design section includes navigation flow diagrams and mock-ups for different user roles—visitor, member, and admin—supporting the frontend development process. Class Diagrams are also included to describe the object-oriented structure of the system through its classes and associations. Sequence Diagrams demonstrate the runtime behavior of the system in specific scenarios such as login, borrowing books, or sending messages. The report concludes by summarizing how the proposed design ensures scalability, maintainability, and alignment with the requirements defined in the analysis phase.

## 2. Detailed Design Class Diagram



The Detailed Design Class Diagram of the AU Library Management System represents the refined structure of the software, transitioning from conceptual analysis into implementable design. This diagram reflects the relational database schema and outlines the system's principal entities (classes), their attributes, and how they interrelate through associations, compositions, and multiplicities. The aim is to provide a definitive guide for developers to implement and integrate the components of the system efficiently.

## 2.1 Class Descriptions and Relationships

### TBLBOOK (Book Class)

Represents the core resource of the system.

- Attributes: ID, Name, Category, Author, YearOfPublication, PublishingHouse, Page, Situation, BookPhoto
- Relationships:
  - Associated with TBLAUTHOR and TBLCATEGORY through foreign keys (many books can have one author and one category).
  - Related to TBLACTION (borrowing records), where a single book can have multiple transactions.
- Multiplicity:
  - Book ↔ Author: *many-to-one*
  - Book ↔ Category: *many-to-one*
  - Book ↔ Action: *one-to-many*

### TBLAUTHOR (Author Class)

Stores metadata about authors.

- Attributes: ID, Name, Surname, Detail
- Relationships: One author may write multiple books.
- Multiplicity: *One-to-many* (Author ↔ Books)

### TBLUSERS (User Class)

Represents library members.

- Attributes: ID, Name, Surname, Mail, Username, Password, Photo, Telephone, School
- Relationships:
  - A user may be involved in many TBLACTION records (borrowed books).
  - A user can also have multiple TBLPUNISHMENT entries.
- Multiplicity: *One-to-many*

## **TBLADMIN (Admin Class)**

Represents authorized system administrators.

- Attributes: ID, AdminUser, Password
- No direct associations shown in the database schema; interaction is logical rather than relational.

## **TBLACTION (Transaction Class)**

Stores data related to book borrow and return operations.

- Attributes: ID, Book, Member, Employee, PurchaseDate, ReturnDate, ActionSituation, UserReturnDate
- Relationships:
  - Linked with TBLBOOK, TBLUSERS, and TBLEMPLOYEE.
  - May relate to TBLPUNISHMENT in cases of late returns.
- Multiplicity: *Many-to-one* for all related entities

## **TBLPUNISHMENT (Penalty Class)**

Stores penalties applied to users for overdue returns.

- Attributes: ID, Member, Start, Expiry, Money, Action
- Relationships:
  - Related to one user and one action record.
- Multiplicity: *Many-to-one*

## **TBLEMPLOYEE (Employee Class)**

Employees handling physical transactions.

- Attributes: ID, Employee
- Relationships: Employees are involved in multiple TBLACTION records.
- Multiplicity: *One-to-many*

## **TBLCATEGORY (Category Class)**

Classifies books by category.

- Attributes: ID, Name
- Relationships: One category can classify many books.
- Multiplicity: *One-to-many*

## **TBLDECLARATION (Simple Announcement Class)**

Stores plain declaration messages for public or internal use.

- Attributes: ID, Declaration
- Not directly linked, assumed to be displayed in the Admin or Showcase interface.

## **TBLANNOUNCEMENT (Rich Announcement Class)**

Used for more structured announcements.

- Attributes: ID, Category, Content, Date
- Categorized display of library-wide updates and news.

## **TBLMESSAGES (Message Class)**

Handles internal communication.

- Attributes: ID, Sender, Receiver, Topic, Message, Date
- Relationships: Many messages per user or admin.
- Multiplicity: *Many-to-one* for sender/receiver

## **TBLCOMMUNICATION (Contact Message Class)**

Captures messages from external users (visitors).

- Attributes: ID, Name, Mail, Topic, Message
- Typically submitted via the contact form on the showcase page.

## **TBLCASE (Financial Record Class)**

Tracks monthly income or penalties.

- Attributes: ID, Month, Sum
- Used in the Admin panel to monitor revenue.

## **Sysdiagrams**

System-internal structure used by SQL Server for ERD metadata.

- Not utilized in logic or UI layers.

### 3. Dynamic models

#### 3.1 Sequence Diagrams:

The sequence diagram illustrates the core interaction flow for three critical user activities in the AU Library System: Register, Login, and Borrow Book. It shows how a visitor interacts with the system to become a member, log in, and borrow a book.

The process begins when a visitor opens the registration page, enters their details, and the system checks for email availability before creating a new user profile. Upon successful registration, the user logs in with valid credentials. Once authenticated, the member can search for books, check availability, and initiate a borrowing process. The system records the transaction and confirms the borrowing with a return date.

This diagram effectively represents how the user, website interface, and database interact to fulfill the system's primary functionalities, ensuring data validation and error handling at each step.

#### 3.2 State Diagram:

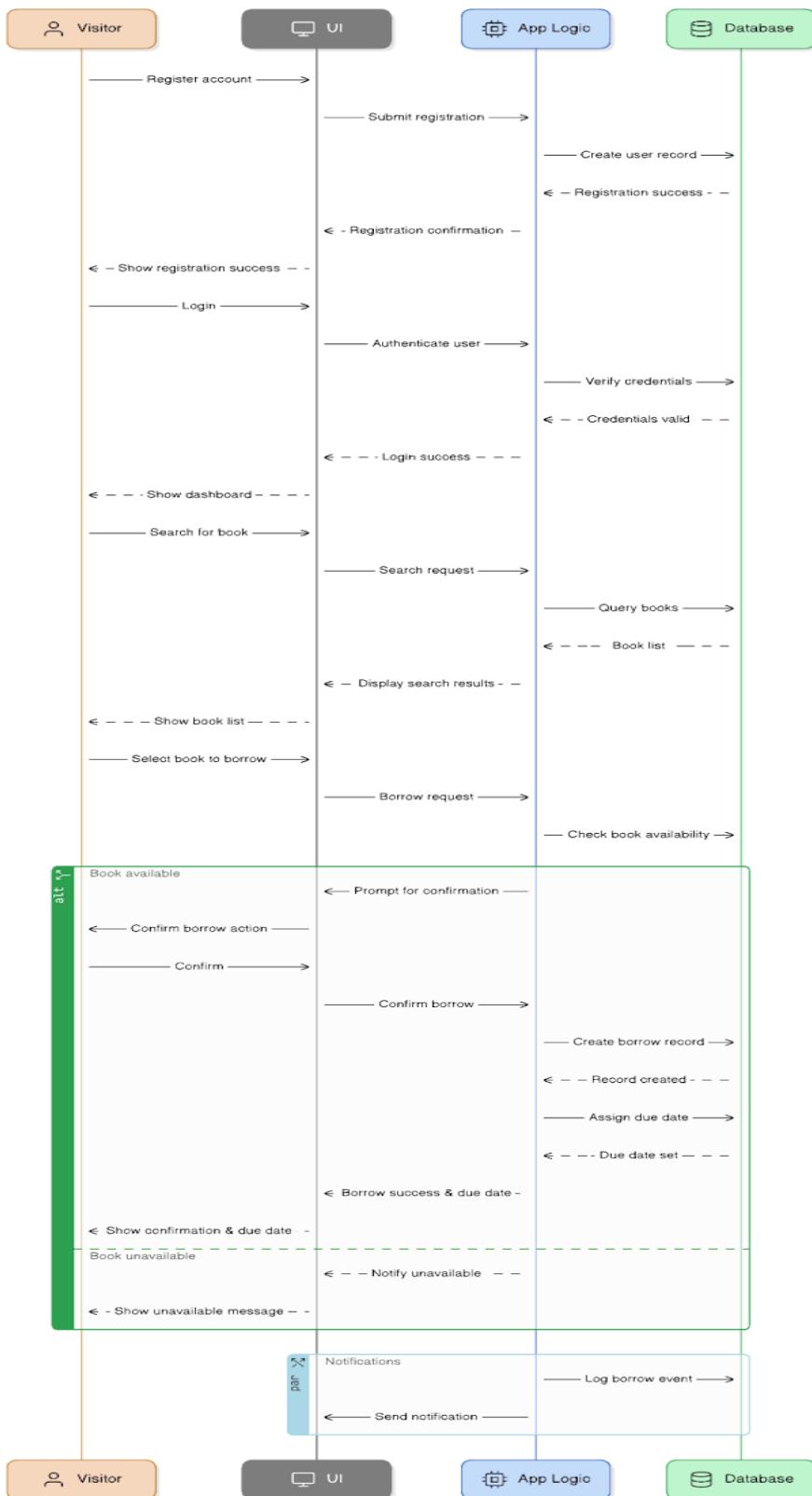
The state diagram of the AU Library System visually represents the dynamic behavior of the system during the book borrowing process. It illustrates how a member transitions between different system states based on their actions and system responses. Beginning with the system's idle state, the user initiates a book search, selects a title, and the system then verifies its availability. Depending on the result, the user can either proceed with the borrowing process or return to searching.

If the book is available, the system transitions through states such as borrow confirmation, successful borrowing, and book usage. Upon returning the book, the system checks the return date to determine whether the return is on time or late. If late, a penalty state is triggered and a fine is calculated and assigned. The diagram provides a clear and systematic flow from the start to the end of a book transaction, emphasizing system validations, conditional branches, and the importance of timely interactions. This state-based modeling enhances understanding of system behavior and supports accurate implementation of control logic in the codebase.

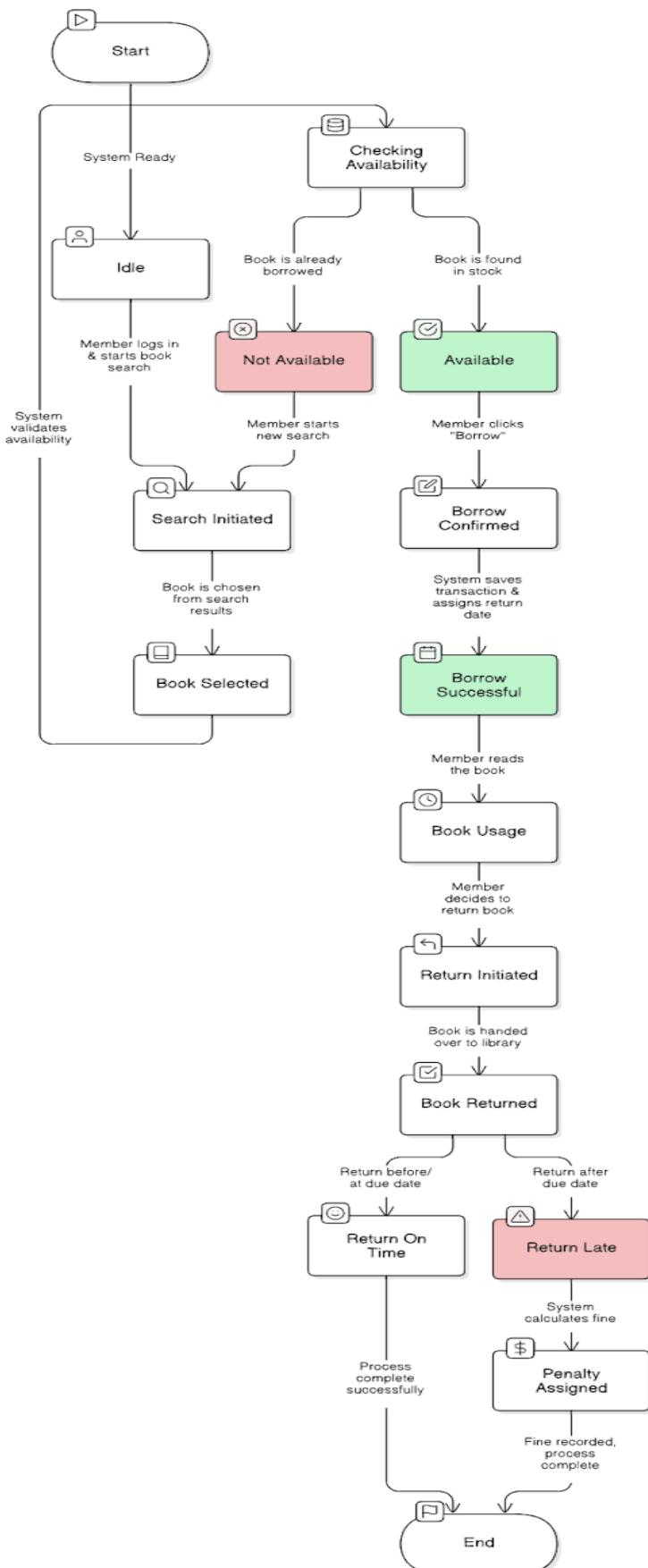
#### 3.3 Activity Diagram:

The activity diagram illustrates the complete workflow of the book borrowing process within the AU Library System, starting with the user logging in and searching for a book. The flow then diverges based on whether the book is found and available. If the book is found and available, the user proceeds to borrow it, prompting the system to create a borrow record, assign a due date, and confirm the transaction. If the book is not found or is already borrowed, the system guides the user to perform a new search, thereby restarting the process. This diagram clearly visualizes decision points, conditional flows, and sequential operations, helping developers and analysts understand the logic of user interactions and system responses to ensure a smooth, intuitive, and reliable borrowing experience.

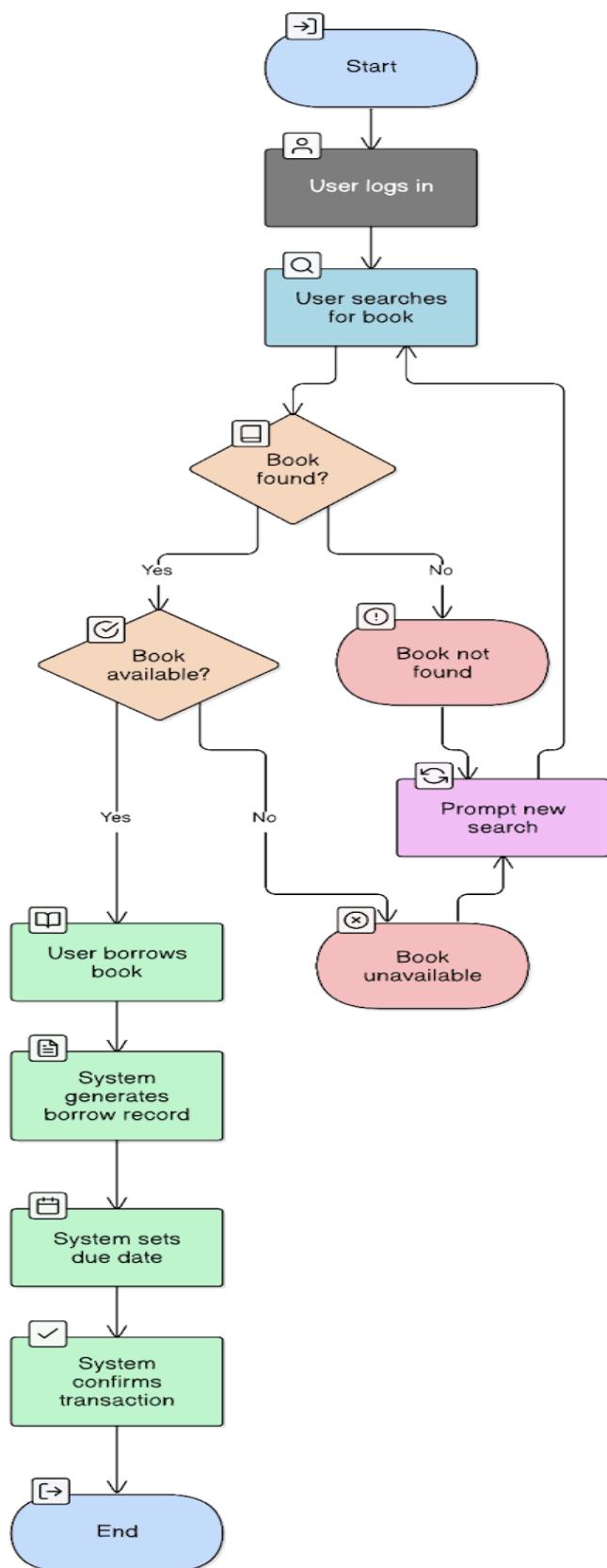
## Sequence Diagrams



## State Diagram



## Activity Diagram



## 4. Software architecture

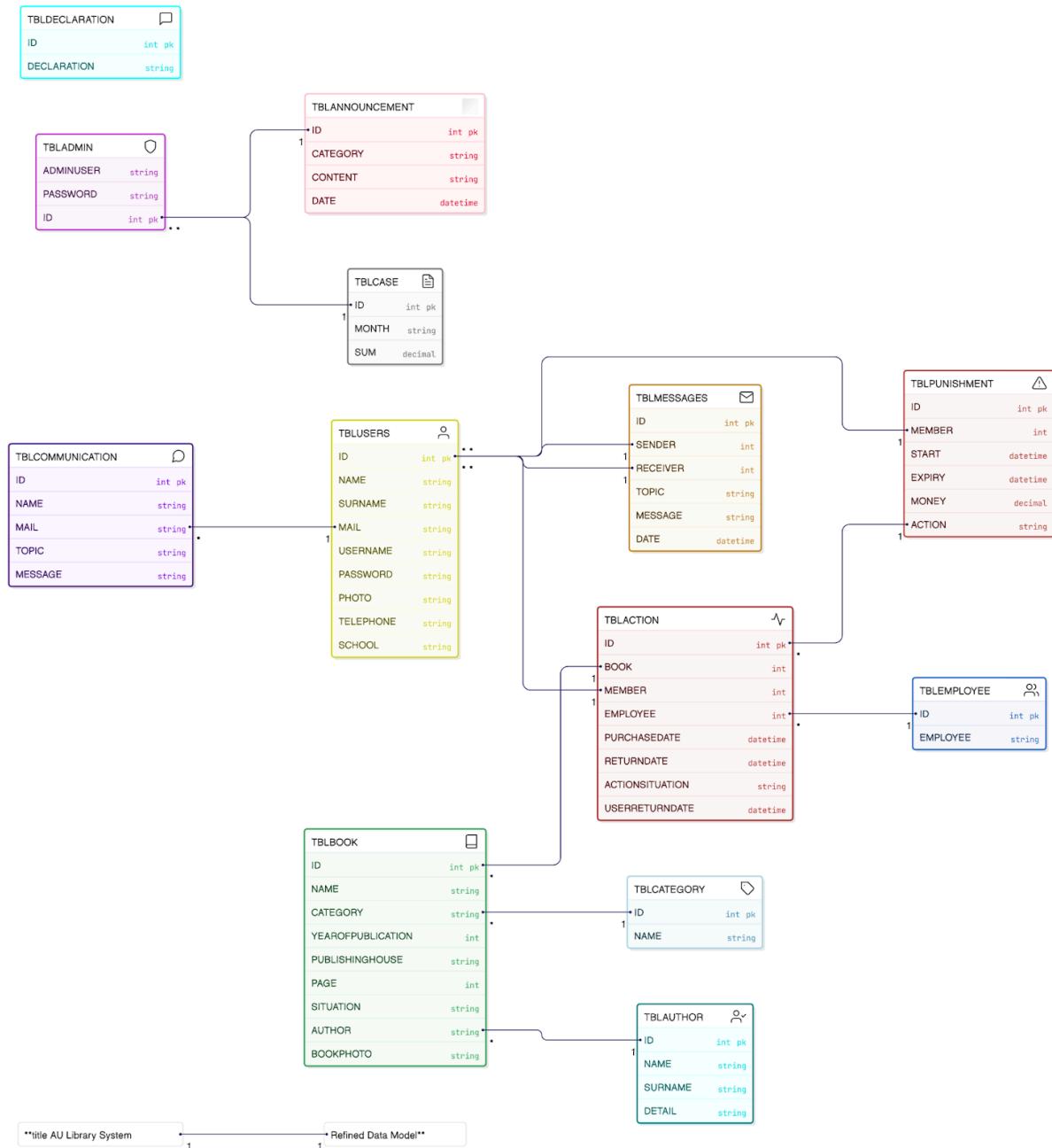
### 4.1 UML Package Diagram

The UML Package Diagram provides a modular view of the AU Library System by organizing classes into logical packages based on functionality. Each package represents a subsystem, such as User Management, Book Management, Transactions, Communication, Finance, and Core Services. Dependencies between packages are indicated by directed arrows to show which modules rely on others. This abstraction helps visualize the high-level structure of the system and ensures that responsibilities are separated, supporting scalability and maintainability of the application.

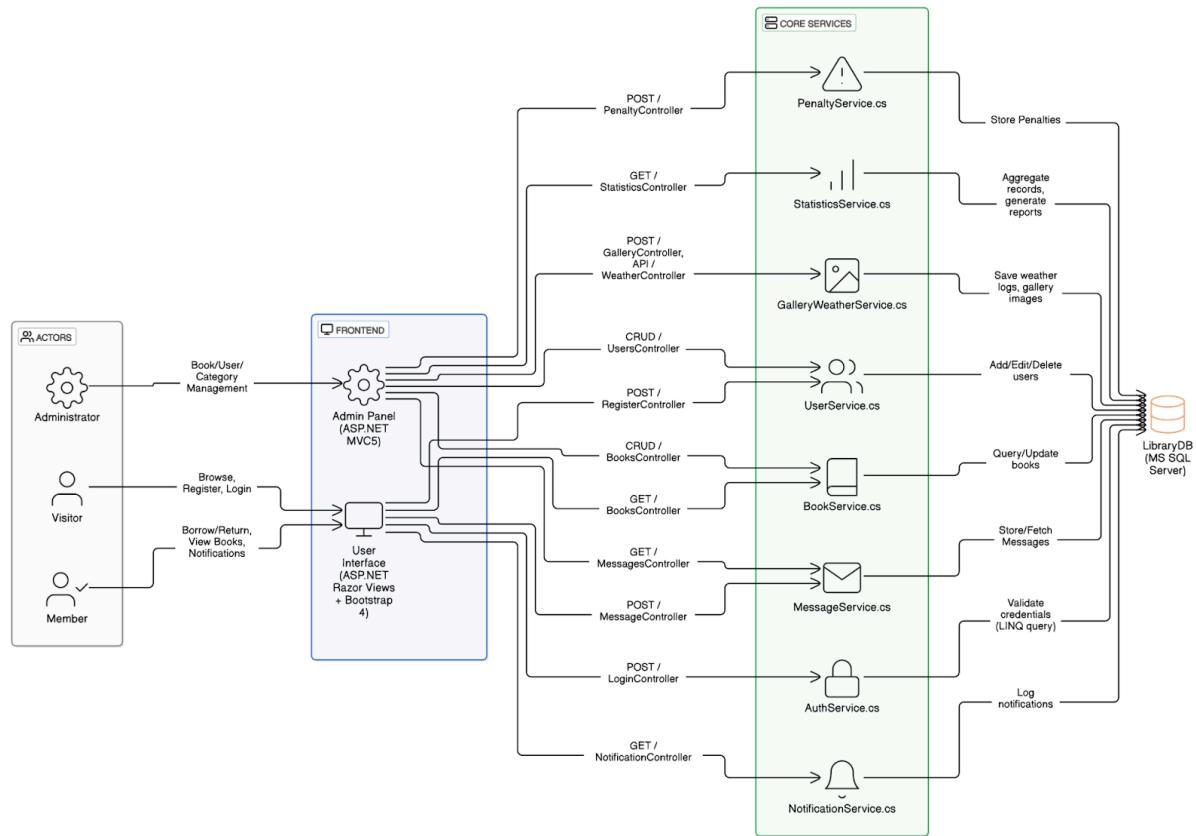
### 4.2 UML Component Diagram

The UML Component Diagram illustrates the physical and logical components that make up the AU Library System and how they interact with each other. It identifies user interfaces, backend services, and the database, showing their responsibilities and communication paths. Each component, such as the Book Management Module or Borrow/Return Service, represents a deployable part of the system implemented in ASP.NET MVC5 with C# and Entity Framework. Connectors between components represent data or control flow. This diagram is essential for understanding the technical architecture and the runtime interaction between independently developed modules.

## UML Package Diagram



## UML Component Diagram



## 5. Entity Relationship Diagram

The Entity Relationship Diagram (ERD) provides a high-level data model that defines the structure of the database used in the AU Library Management System. It outlines the system's primary entities, their attributes, and the relationships between them.

### Key Entities:

- **User**

*Attributes:* UserID, FullName, Email, Password, RegistrationDate

*Relationships:* One-to-many with BorrowRecord, Message

- **Admin**

*Attributes:* AdminID, Name, Email, Password

*Relationships:* One-to-many with Announcement, Message, Book, Category, ChecklistTask

- **Book**

*Attributes:* BookID, Title, ISBN, CategoryID, AuthorID, Status, ImagePath

*Relationships:* Many-to-one with Category and Author; one-to-many with BorrowRecord

- **Author**

*Attributes:* AuthorID, Name, Biography

*Relationships:* One-to-many with Book

- **Category**

*Attributes:* CategoryID, Name

*Relationships:* One-to-many with Book

- **BorrowRecord**

*Attributes:* RecordID, UserID, BookID, BorrowDate, DueDate, ReturnDate, Penalty

*Relationships:* Many-to-one with User and Book

- **Message**

*Attributes:* MessageID, SenderID, ReceiverID, Subject, Body, Date

*Relationships:* Many-to-one with User and Admin

- **Announcement**

*Attributes:* AnnouncementID, Title, Content, DatePosted

*Relationships:* One-to-many from Admin

- **WeatherReport**

*Attributes:* ReportID, Temperature, Description, Date

Managed by Admin; not directly linked to users.

- **GalleryImage**

*Attributes:* ImageID, ImagePath, UploadDate

Uploaded by Admin

- **ChecklistTask**

*Attributes:* TaskID, Title, Description, IsCompleted

Used internally by Admin to track daily tasks

Each relationship is implemented using **primary keys** and **foreign keys** to ensure referential integrity.

The ERD is crucial for the database development phase as it guides table creation and query

structuring.

## 6. Glossary & References

### Glossary

- **Entity:** A real-world object or concept represented in the database (e.g., Book, User).
- **Attribute:** A property or characteristic of an entity (e.g., Title, Email).
- **Relationship:** The association between entities (e.g., a User borrows a Book).
- **Foreign Key:** A field that links one entity to another, ensuring relational consistency.
- **Primary Key:** A unique identifier for each record in a table.
- **CRUD:** Create, Read, Update, Delete – basic operations in data management.
- **BorrowRecord:** A log capturing the book borrowing transaction for each user.
- **Admin Panel:** The administrative backend used to manage users, books, and content.
- **Penalty:** A monetary charge applied for late returns.
- **Gallery:** A module that displays library-related images uploaded by admins.

### References

1. **ASP.NET MVC5 Official Documentation**  
<https://learn.microsoft.com/en-us/aspnet/mvc/overview>
2. **Microsoft Entity Framework Documentation**  
<https://learn.microsoft.com/en-us/ef/>
3. **Bootstrap Framework**  
<https://getbootstrap.com/>
4. **OpenWeatherMap API**  
<https://openweathermap.org/api>
5. **C# Language Guide – Microsoft Docs**  
<https://learn.microsoft.com/en-us/dotnet/csharp/>
6. **CE370 Software Engineering Lecture Notes – Altınbaş University**