



Library System

Under Supervisor :

Dr: Osama Hefny

Content

- ❖ Introduction.
- ❖ Scope.
- ❖ Requirements.
- ❖ Use case diagram.
- ❖ Use case template explanation.
- ❖ Use case template.
- ❖ Class diagram.
- ❖ Sequence diagram.

Introduction

A Library System project focuses on creating a software solution for efficient library management, encompassing tasks like book cataloging, user management, and check-in/check-out processes. The goal is to streamline operations, improve resource accessibility, and offer accurate information on book availability.

Scope

The objective of this project involves creating an internet-accessible library system with a central server. Users interact through a web browser, managing secure accounts, searching for books, and contacting administrators. The website streamlines e-book purchases for user convenience, detailed in processes and requirements section.

Requirements

1. User Registration and Authentication : " Functional "

- Allow users to register for Library by entering (first name, last name, date of birth, password).
- Authenticate user identity during login.

2. Catalog Management : " Functional "

- Add new books to the library catalog.
- Update book info.
- Delete books from the catalog.

3. Search and Retrieval : " Functional "

- Display availability and location.
- Implement advanced search.
- Allow users to search for book.

4. Borrowing and Returning : " Functional "

- Facilitate book checkout and record borrowing details.
- Manage due dates and notify users of upcoming returns.
- Handle book returns and update availability status.

5. User Management : " Functional "

- Create and manage user accounts.
- Update user information.

6. Reporting and Analytic : " Functional "

- Generate reports on book Provide.
- analytic on each books category.

7. Favorites section : " Functional "

- Enables users to add books to his favorites.
- Adding favorites section which enables user to see his favorites.

8. Book Recommendations : " Functional "

- Provide personalized book recommendations based on user preferences, reading history, or popular trends.

9. Payment : " Functional "

- Allow users to pay for borrowing.
- Manage discount codes.

10. Discount codes : " Functional "

- Allow users to borrow books at a discount through a variety of means Unless requirements.

11. Reading History : " Functional "

- Maintain a reading history for users.
- allowing them to track the books they have borrowed and read in the past.

12. Book Reviews and Ratings : " Functional "

- Allow only book borrowers to write reviews and provide ratings.
- facilitating community engagement and aiding in book selection.

13. Social Media Integration : " Functional "

- Enable sharing book recommendations, reviews, or borrowing activities on social media platforms.

14. Notifications : " Functional "

- Send notifications to users when their reserved books are available for pickup.
- Notify users when requested books become available.

15. Copyrights : " Non-Functional "

- Attaching the edition of the book, its author, and the publishing house for all borrowed books.

16. Book Availability Status : " Functional "

- Display real-time availability status for each book in the library catalog.
- indicating whether a book is currently available for borrowing or if it's on loan.

17. Shopping cart : " Functional "

- Allow users to select and reserve books.
- User can add and remove books as he wish.

18. Reading in the application : " Functional "

- Enable users to read their books directly in the application without the need of downloading.

19. Taking notes : " Functional "

- Provide the feature of taking note which enables user the take his own note and save it .

20. Cancel booking : " Functional "

- Allow user to cancel reservation within the first 24 hours of booking.

21. Performance : " Non-Functional "

- Ensure fast response times for searches and operations.
- Handle concurrent user requests efficiently.

22. Security : " Non-Functional "

- Use secure communication protocols.
- Enforce access controls to prevent unauthorized access.

23. Reliability : " Non-Functional "

- Minimize system failures and downtime.

24. Usability : " Non-Functional "

- Provide an intuitive user interface.
- Offer clear instructions and assistance for system navigation.

25. Compatibility : " Non-Functional "

- Support different devices and operating systems.
- Enable access from desktops, laptops, tablets, and mobile devices.

26. Scalability : " Non-Functional "

- Accommodate a growing number of books, users, and transactions.
- Scale up or down based on library needs .

27. Maintainability : " Non-Functional "

- Ensure clean and modular code.
- Provide documentation and support tools for maintenance.

28. Protection : " Non-Functional "

- Protect user data and privacy.

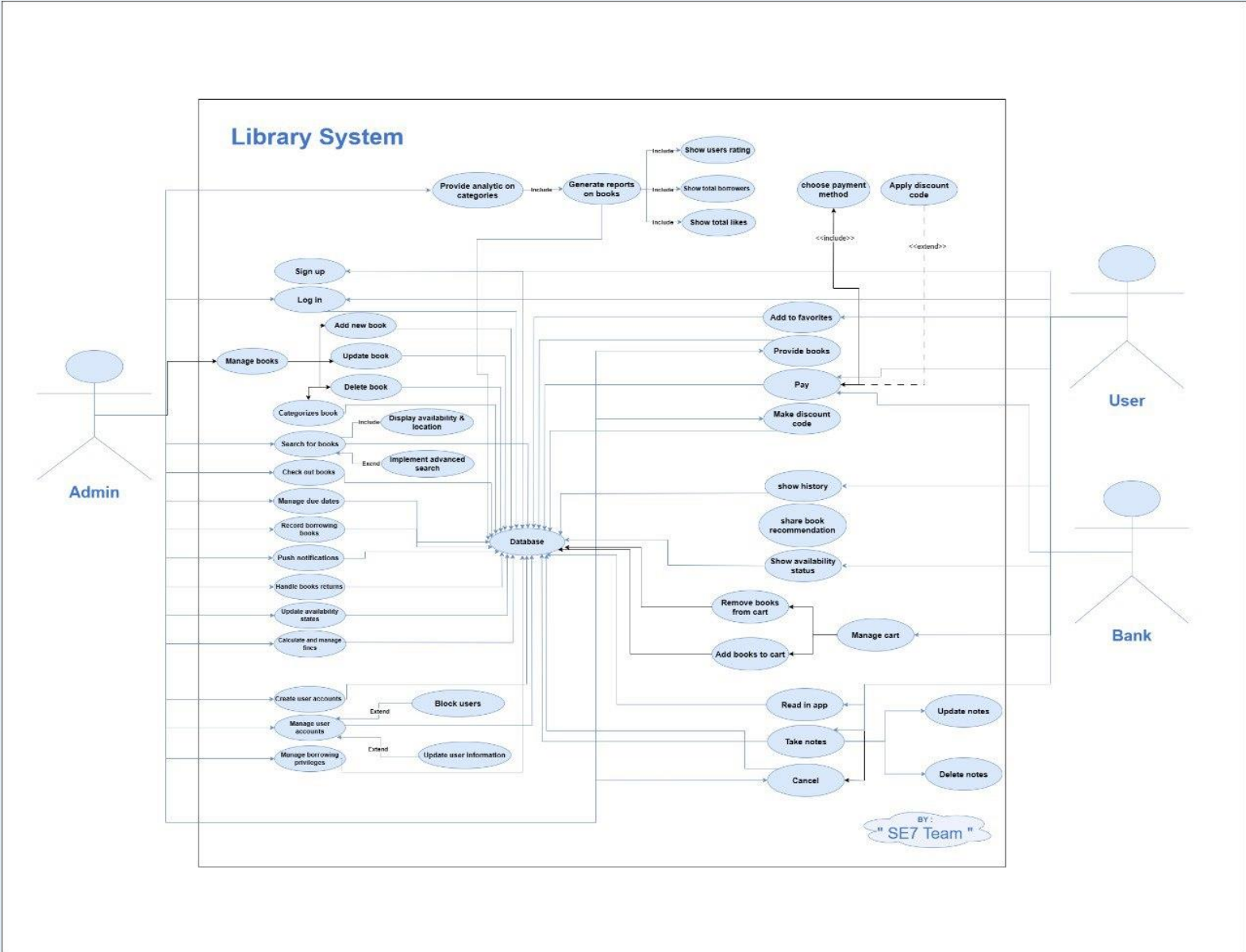
29. Integration with external services : " Non-Functional "

- Integrate with external services to retrieve additional book information, such
- as book summaries, reviews, or cover images.

30. Accessibility Features : " Non-Functional "

- Implement accessibility features to ensure the library system can be used by individuals with disabilities, such as screen reader compatibility or adjustable font sizes.

Use case diagram :



Use case template explanation

- **Actors:** represent the entities, whether human or system, that interact with the system to achieve a specific goal.
- **Description:** provides a brief overview of the use case.
- **Trigger:** identifies the event that initiates the execution of the use case.
- **Pre-conditions:** specify the conditions that must exist before the use case is executed.
- **Post-conditions:** describe the expected outcomes of the system after successful execution of the use case.

- Normal Flow: represents the most common and expected scenario.
- Alternative Flow: represent variations to the primary flow and explain how the system should behave in those situations.
- Exceptions: documents any unexpected events that may occur during the execution.
- Includes: lists other use cases that are included by the current use case.
- Frequency of Use: specifies how often the use case is expected to be executed within a given time frame.

- Assumptions: list any assumptions made during the definition of the use case.
- Special Requirements: highlight any specific non-functional requirements associated with the use case.
- Issues: document any known concerns associated with the use case.

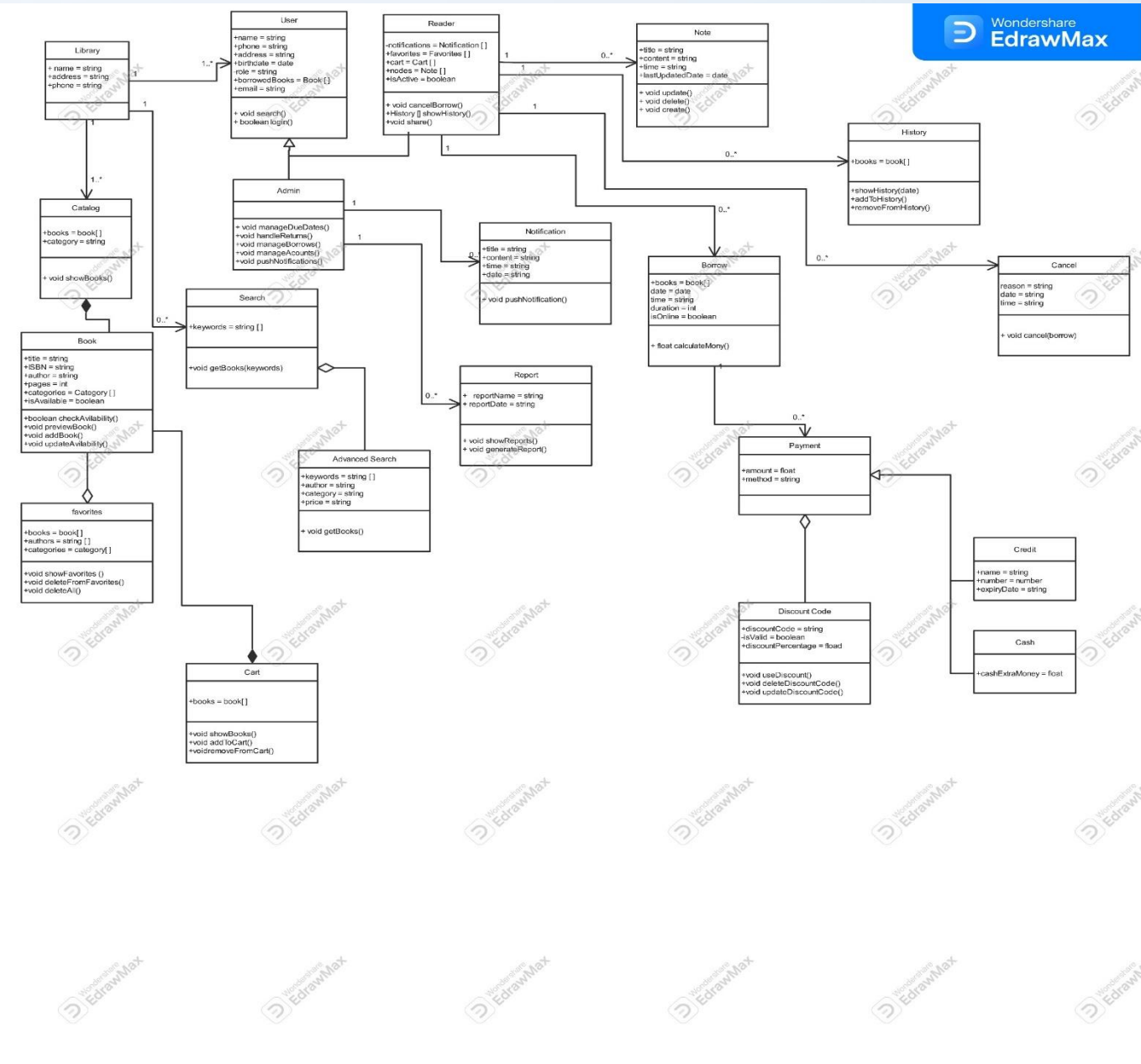
Use case template :

Use Case ID:	UC003		
Use Case Name:	Manage books		
Created By:	Nadia Helal Hassan	Last Updated By:	Nadia Helal Hassan
Date Created:	Nov 9 th ,2023	Last Revision Date:	Nov 16 th ,2023
Actors:	Admin (primary actor).		
Description:	The "Manage Books" use case involves admins adding, updating, or removing books from the library system.		

Page 6

Trigger:	admins trigger the 'Manage Books' module, select actions such as 'Add \new book', 'Update Book Details' or 'Remove Book'.
Preconditions:	1. User has librarian or administrator privileges. 2. Library system is operational.
Post conditions:	1. Book database reflects changes. 2. Users can access updated book information.
Normal Flow:	1. Admin logs in. 2. Navigates to "Manage Books" section. 3. Adds, updates, or removes books.
Alternative Flows:	Step "3", in case a book not Found: Admin encounters a book not in the system, initiates adding new book details.
Exceptions:	Database Error: Technical issues preventing updates.
Includes:	"Add new book" , "Update books" , "Delete books" and "Categorize" use cases.
Frequency of Use:	Frequent.
Special Requirements:	Book information accuracy.
Assumptions:	Admins have necessary training.
Notes and Issues:	Regular database backups.

Class diagram :



Sequence diagram :

