https://www.pes.edu/wp-content/uploads/2019/09/pes_logo.png

**FINAL SEMESTER ASSESSMENT (FSA) B.TECH. (CSE)**

**VI SEMESTER**

**UE18CS355 – OBJECT ORIENTED ANALYSIS AND DESIGN WITH SOFTWARE ENGINEERING LABORATORY**

**PROJECT REPORT**

**ON**

**E-BOOK MANAGEMENT SYSTEM**

SUBMITTED BY

|  |  |
| --- | --- |
| **NAME** | **SRN** |
| 1. **Afzal Mukhtar** | **PES2201800675** |
| 1. **Hritika Rahul Mehta** | **PES2201800024** |
| 1. **Meghana I.** | **PES2201800028** |

**JANUARY – MAY 2021  
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**EC CAMPUS,**

**BENGALURU – 560100, KARNATAKA, INDIA**

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | |
| **Sl.No** | **TOPIC** | **PAGE No** |
|  | [ABSTRACT](#_Abstract) | 1 |
|  | [SOFTWARE REQUIREMENTS SPECIFICATION](#_SRS_1) | 2 |
|  | [PROJECT PLAN](#_Project_Plan) | 13 |
|  | [DESIGN DIAGRAMS](#_Design_Diagrams) | 20 |
|  | [MODULE DESCRIPTION](#_Modules_1) | 31 |
|  | [TEST CASES](#_Test_Cases) | 32 |
|  | [SCREEN SHOTS OF OUTPUT](#_Output_Screenshots) | 40 |

# Abstract

This project is aimed at building a simple E-Book Management system, one much like an “online library” or a “virtual library”. The book management portal will provide a user-friendly interface to store and maintain details of members or users and the books. It will also facilitate easy access to books. Functionalities like “login and register”, “search for book”, “update library”, and “borrow book” will be added for ease of usage.

The purpose of this E-Book Management System is to provide to the target users, the access and subscription to the required books either free or at a prescribed cost for a defined amount of time. The user interface allows finding books based on his/her preferences and subscribing to the same according to the applicable rules. The application also allows the administrator to keep track of the available books and filter them based on various criteria thereby making it more user-friendly.

# 

# SRS

## Introduction

### Purpose

The primary aim of the document is to represent the requirements of the project, E-Book Management System. This document gives the detailed description of both functional and non- functional requirements. The main purpose is to provide a user-friendly environment to maintain the details of the users, books and enable easy and convenient access to books.

### Intended Audience

The document is proposed for the system developers, University and public Librarians, administration faculty of the Education Department and to whosoever is interested.

* 1. **Product Scope**

E - Book Management System can also be referred to as an ‘Online Library’ or a ‘Virtual Library’. The purpose is to provide to the target users, the access and subscription to the required books either free or at a prescribed cost for a defined amount of time. The user interface allows finding books based on his/her preferences and subscribing to the same according to the applicable rules.

The application also allows the administrator to keep track of the available books and filter them based on various criteria thereby making it more user-friendly.

### References

Books:

* + - “Software Engineering: Principles and Practice”, Hans van Vliet 3rd edition Wiley India 2010
    - “Software Requirements and Specifications: A Lexicon of Practice, Principles”, Michael J. Jackson and Michael Jackson

Websites:

* + - [https://www.slideshare.net/](https://www.slideshare.net/ToseefHasan2/srs-for-library-management-system)
    - [https://www.krazytech.com](https://www.cse.msu.edu/~chengb/RE-491/Papers/SRS-BECS-2007.pdf)

## Overall Description

### Product Perspective

E - Book Management System is a virtual bookstore/library application which mainly gains popularity because of its online feature and it supports a number of functions for both the users and the stores. It can be proposed as a follow-on member of existing Online E-Book systems with a few additional features which are lacking in the current versions.

### Product Functions

The Product Functions include the following interfaces:

### The ‘Library’

* + - Searching for books
    - Search Refinement by Categories, language, publication, availability etc
* On clicking a book
  + Look into ratings/reviews
  + Borrow the book
  + Return the book

### Bookshelf - Books you ‘borrowed’

* + List of the books
  + Book Info (Title, Author, Ratings, Review)

### Book Interface

* + Bookmarking

### Account Info

* + User info

### User Classes and Characteristics

* + - Libraries
      * Providing the books
    - Users
      * “Borrowing” the books
      * Leaving Reviews & Ratings
      * Book Interface
        + Bookmarking

### Operating Environment

The designed application would operate well in all the existing operating system (Mac OS).

The hardware configuration includes PROCESSOR: Pentium(R) Dual Core-CPU

HARD DISK: 40 GB, basic I/O devices like keyboard, mouse, monitor screen.

Access to good internet connection is one of the vital requirements for the application.

#### Design and Implementation Constraints

* + - Protection of the application against threats and vulnerabilities.
    - Maintenance of the confidentiality and integrity of the data.
    - Timely updates about the books and users must be made.

### Assumptions and Dependencies

The assumptions include:

* + - User-friendly application.
    - Data about users and books must be recorded and accessible by the users according to their levels of access.
    - Confidentiality, Integrity and Availability of data is preserved.
    - Ubiquitous, high storage, fast and on-demand access to the applications’ resources.
    - Constantly running and no failure data server and warehouse. The dependencies include:
      * The specific hardware and software requirements for the application to run.
      * The users should have a proper understanding of the features in order to use it.
      * Timely addition, updating and deletion feature.
      * The application should have a general stored report.

## External Interface Requirements

### User Interfaces

There are two levels of interfaces:

#### ADMINISTRATOR MODE:

* + - Register user
    - Enter Book Details
    - Issue a Book
    - Resolve Conflicts

#### USER MODE:

A user can be previously registered or can register on spot A user can:

* + - Check the availability of books
    - Borrow a book
    - Return a book

### Software Interfaces

Front end development requirements:

* + - Python Tkinter

Back-end requirements:

* + - Python
    - SQLite Database

Operating System Requirements:

* + - Mac OS 10.15 or above (Development)

### Communication Interfaces

*HTTP for accessing the books to be downloaded onto the User’s device.*

### Hardware Interfaces

PROCESSOR: Pentium(R) Dual Core-CPU HARD DISK: 40 GB

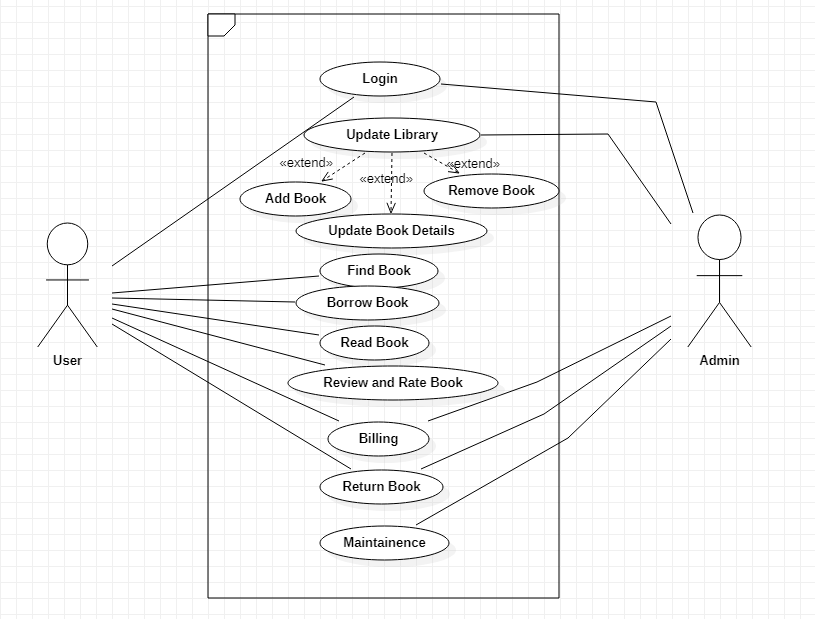
RAM: 256 MB or more

OPERATING SYSTEM: Mac 10.15 & above

## Analysis Models

### USE CASE DIAGRAM: -

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved

.

## System Features

### User Authentication

1. **Description and Priority**

User authentication is a security process that covers all of the human-to-computer interactions that require the user to register and log in. Said more simply, authentication asks each user, “who are you?” and verifies their response.

Priority: High

1. **Stimulus/Response Sequences**

User Authentication is common to both modes of interface.

A new user provides valid credentials which would lead to their registration. Registered users can login using their username & password.

1. **Functional Requirements**

**REQ-1:** New users to the application would have to register themselves by providing the required information asked for and they are validated in order to prevent conflicts within the same.

**REQ-2:** Registered users can login successfully by providing valid credentials and would lead the user to the Homepage of the application.

**REQ-3:** Unsuccessful login with invalid credentials would lead to an error message and another attempt for the same.

### Finding Books

1. **Description and Priority**

A user can search through the online catalogue available to them and find books pertaining to their interests.

Priority: Medium

1. **Stimulus/Response Sequences**

A search box to enable the user to enter a particular title and find it. A filter option to refine the search results.

1. **Functional Requirements**

**REQ-1:** A search box is implemented where the user types in key terms pertaining to their book, and the search delivers its response as a set of results.

**REQ-2:** The search can be filtered by various criteria such as genre, publications, price range, language etc., according to the user’s preference.

### Subscribing Plans

1. **Description and Priority**

A user can have a subscription plan that will allow them to borrow books from the library for a set period of time.

Priority: High

1. **Stimulus/Response Sequences**

Page containing features/information of the Subscription Plan including

* Plan Description
* Monthly Validity
* Price

The user can choose a subscription plan and subscribe to a book.The book would be available to read once the user subscribes to a plan.

1. **Functional Requirements**

**REQ-1:** Choice of subscription plans (according to the book)

**REQ-4:** To choose a subscription plan, and upon success, the books are made available to the user for reading.

### Reading the Book

1. **Description and Priority**

This feature allows the users to read the book in their native book readers.

Priority: Low

1. **Stimulus/Response Sequences**

Bookmarking – Saving the last page read, for user to keep track

Reading book in native reader of choice

1. **Functional Requirements**

**REQ-1:** The user is allowed to read the book and perform any functions provided by the native reader.

**REQ-3:** The user can bookmark pages to look into later.

## Other Non-functional Requirements

### Performance Requirements

* + - The performance of the system must be fast and accurate.
    - The management system must be able to handle expected and unexpected errors, Eg. Inbuilt error testing to identify invalid username/password.
    - The system should be able to handle large amounts of data. It must accommodate a high number of books and users without faults.
    - The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal.
    - The system shall be allowed to take more time when doing large processing jobs.

### Safety Requirements

* + - Maintaining the database is one of the key responsibilities of the proposed E-Book Management System.
    - The database may crash at any point of time and reason could be virus or operating system failure. Having a database backup and recovery system is vital.
    - Proper UPS / inverter facility should also be there in case of power supply failure.

### Security Requirements

* + - System uses a secured database.
    - System will have different types of users and every user has access constraints.
    - Proper user authentication should be provided.
    - There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

### Software Quality Attributes

* + - Admins will have the right to create changes to the system, but the members or other users cannot do changes.
    - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database.
    - The user is able to easily download and install the system.

### Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision or infer new data from existing data. This includes the rules and regulations that the application users must abide by. May also include the cost and the discount prices offered. Neither the user nor the admin should involve in illegal activities by crossing the rules.

## Other Requirements

The additional requirements include:

* Well documented legalized business rules.
* Always on server in the data centers.
* High storage and fast access database.

## Appendix A: Glossary

1. E-Book – Electronic book - book publication made available in digital form, consisting of text, images, or both, readable on electronic devices.
2. Bookmark – record the address of (a website, file, etc.) to enable quick access in future.
3. Personalization / Customization – the act of designing for individual requirements.
4. Confidentiality – the state of keeping or being kept secret or private.
5. Integrity – the quality of being honest and having strong moral principles.
6. Availability – the quality of being able to be used or obtained.
7. Dependency – the degree to which each program module relies on each one of the other modules.
8. Ubiquitous – existing or being everywhere at the same time.
9. Authentication – Ability of the system to confirm the identity of the user.
10. Stimulus – something that provokes or causes an action or response.
11. Credentials – testimonials or certified documents showing that a person is entitled to credit or has a right to exercise official power
12. Subscription – an arrangement to receive something, typically a publication, regularly by paying in advance.
13. Licensing – a business arrangement in which one company gives another company permission to manufacture its product for a specified payment.
14. Patenting – a form of intellectual property that gives its owner the legal right to exclude others from making, using, or selling an invention for a limited period of years in exchange for publishing an enabling public disclosure of the invention.

## Appendix B: Field Layouts

An Excel sheet containing field layouts and properties/attributes and report requirements.

### Sample sheet with the information required to register the user: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Length** | **Data Type** | **Description** | **Mandatory** |
| Sl. No | 4 | Numeric | Incremental Number for records purpose | Y |
| Customer Name | 50 | String | Name of User | Y |
| E-Mail ID | 30 | Alphanumeric | Mail ID of user | Y |
| Unique ID | 5 | Numeric | Unique Identifier assigned to each user | Y |
| Mobile No | 10 | Numeric | Mobile number of users | N |
| Age | 2 | Numeric | Age of user | N |
| Account Username | 50 | Alphanumeric | Account information | Y |

### Sample Report Requirements: -

|  |  |
| --- | --- |
| **Registration Report** | **Subscription Report** |
| Sl. No | Unique ID |
| Customer Name | Subscription Plan |
| E-Mail ID | List of books currently subscribed to |
| Unique ID | Amount Payable |
| Mobile No | Subscription Timeline |
| Age | Transaction Details |
| Account Username | Remarks |

## Appendix C: Requirement Traceability Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No** | **Requirement ID** | **Brief Description of Requirement** | **Architecture Reference** | **Design Reference** | **Code File Reference** | **Test Case ID** | **System Test Case ID** |
| 1 | REQ#1 | Updating details of a book in the system | 2.2.3 | 2.4.1, 4.2 & 4.5 | Database | U\_46 – U\_52 & U\_55 – U\_58 | U\_46 – U\_52 & U\_55 – U\_58 (GUI) |
| 2 | REQ#2 | Deleting an existing book in the system | 2.2.3 | 2.4.1, 4.2 & 4.5 | Database | U\_32 – U\_35 | U\_32 – U\_35 (GUI) |
| 3 | REQ#3 | Adding a new book to the catalogue | 2.2.3 | 2.4.1, 4.2 & 4.5 | Database | U\_16 – U\_24 | U\_16 – U\_24 (GUI) |

# Project Plan

## 1: Identify the lifecycle to be followed for the execution of your project and justify why u have chosen the model.

The lifecycle selected is an***Iterative*** approach.

An Iterative approach has been selected as it is most fitting for our project requirements and objectives. Some of the factors involved in deciding the Lifecycle approach has been listed in the table below along with how well each of the approach fares pertaining to a specific factor.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FACTORS** | **Waterfall** | **V-**  **Shaped** | **Prototyping** | **Iterative** | **Incremental** | **Agile** |
| **Unclear User Requirement** | Poor | Poor | Good | Good | Poor | Excellent |
| **Reliable System** | Good | Good | Poor | Good | Good | Good |
| **Short Time Schedule** | Poor | Poor | Good | Excellent | Excellent | Poor |
| **Cost Limitations** | Poor | Poor | Poor | Excellent | Good | Excellent |
| **Visibility of**  **Stakeholders** | Good | Good | Excellent | Good | Excellent | Good |
| **Component**  **Reusability** | Good | Good | Poor | Excellent | Good | Poor |
| **Documentation** | Excellent | Excellent | Good | Excellent | Excellent | Poor |

Our project is a simple implementation of an online book management system with many reusable components, almost clear-cut user requirements and well-defined user-interface. These factors allow for the product to be built in multiple iterations wherein each iteration can incorporate the user’s feedback and requirements. This is a cost effective and easy method to test and debug in each of the smaller iterations.

After considering all of the above factors, an iterative approach seemed to be the best fit for our project implementation as this approach caters to all of the project needs.

## 2: Identify the tools which u want to use it throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.

The following tools are thought of to be used throughout the lifecycle:

1. **Planning Tools:**
2. **Design Tools:**
3. **Version Control Tools:**
4. **Development Tools:**
5. **Bug Tracking Tools:**
6. **Testing Tools:**

## 3: Determine all the deliverables and categorize them as reuse/build components and justify the same.

CBSE processes are software processes that support component-based software engineering. They take into account the possibilities of reuse and the different process activities involved in developing and using reusable components. There are two types of CBSE processes:

* + **CBSE for reuse** is concerned with developing components or services that will be reused in other applications. It usually involves generalizing existing components.
  + **CBSE with reuse** is the process of developing new applications using existing components and services.

For this project, we have focused on **CBSE with reuse** wherein widely available and popularly used components are ‘reused’ as per project requirements.

The following are the deliverables for this project in terms of components:

### USER AUTHENTICATION:

|  |  |  |
| --- | --- | --- |
| **TASK** | **CATEGORY** | **JUSTIFICATION** |
| Sign Up Form | Reuse | Generic components like textboxes, dropdown lists and checkboxes needed for a sign-up form are widely used and easily available. |
| Login Page | Reuse | Generic components like textboxes, dropdown lists and checkboxes needed for a login page are widely used and easily available. |

### READING THE BOOK:

|  |  |  |
| --- | --- | --- |
| **TASK** | **CATEGORY** | **JUSTIFICATION** |
| Navigating through book | Build | Switching to a specific page as well as page-to-page navigation requires components like pagination, toggle switch, breadcrumb, etc which have to be built as per project requirements. |
| Customization | Build | Changing font size and toggling between dark/light reading modes require components like toggle switch, font size slider, etc which have to be built as per project requirements. |

### BOOK DETAILS:

|  |  |  |
| --- | --- | --- |
| **TASK** | **CATEGORY** | **JUSTIFICATION** |
| Display | Reuse | Displaying of basic information about the book including description, reviews, ratings need generic reusable components like textboxes, rating icons, etc. |
| User Interaction | Build | Submitting a rating/review requires customized components to promote better user interface. |

### BORROW BOOK:

|  |  |  |
| --- | --- | --- |
| **TASK** | **CATEGORY** | **JUSTIFICATION** |
| Billing | Reuse | Billing requires user credentials, payment method details, secure gateway etc. which are features used in most online ventures and hence easily available to reuse. |
| Adding  Book to List | Build | Adding user-interested books to cart would require components like cart icon, books list table, etc which have to be built to incorporate better UI. |

### SEARCH BOOK:

|  |  |  |
| --- | --- | --- |
| **TASK** | **CATEGORY** | **JUSTIFICATION** |
| Matching & Displaying Queries | Reuse | Search boxes, query retrieval code, etc are widely used and easily available components |
| Filtering | Build | Filtering according to user interests would require customization of components. |
| Curating List | Build | A list of curated books aligning with users’ preference requires customized components. |

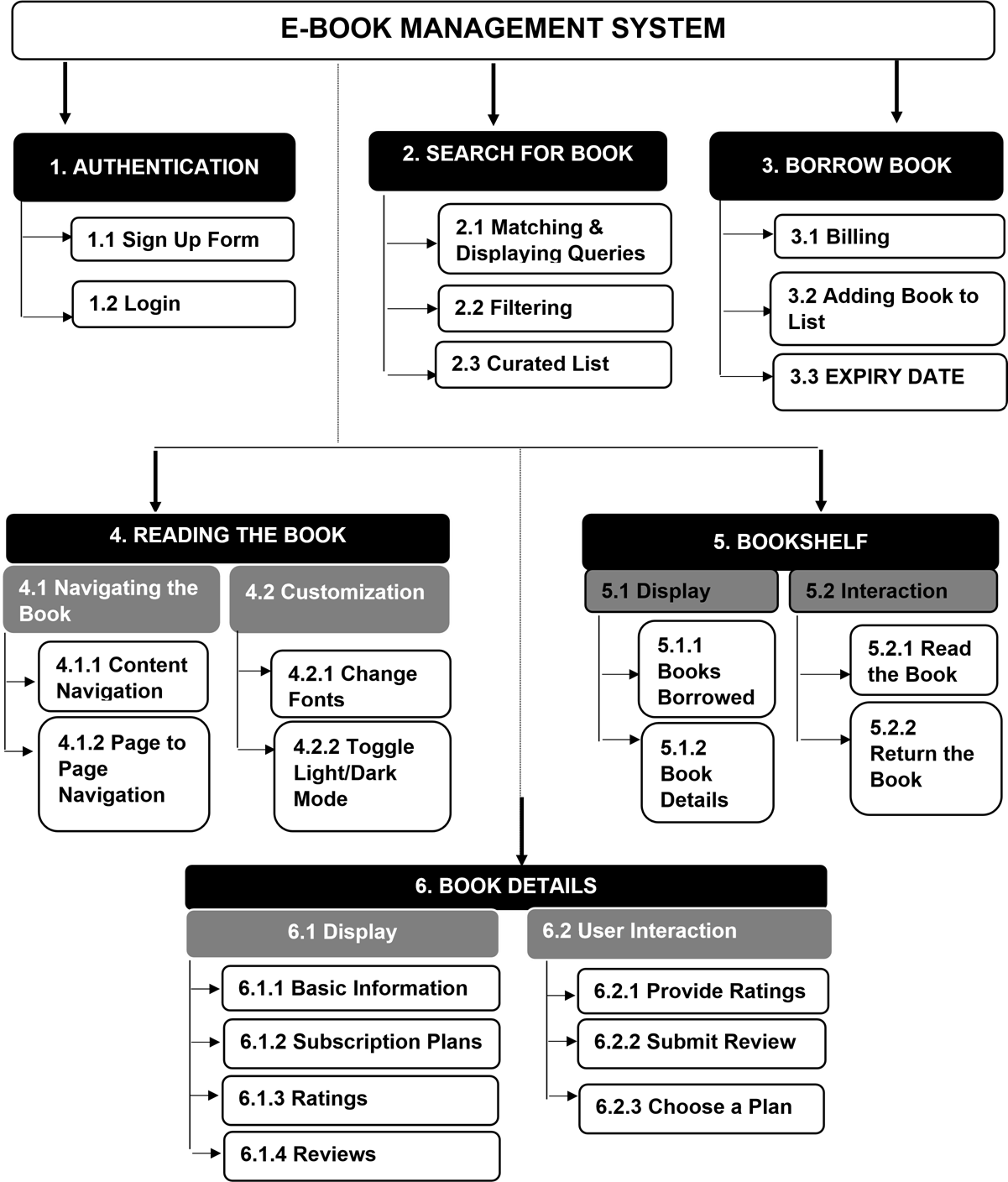
### BOOKSHELF:

|  |  |  |
| --- | --- | --- |
| **TASK** | **CATEGORY** | **JUSTIFICATION** |
| Display | Reuse | Displaying details about borrowed books need generic components like textboxes, etc. |
| Interaction | Build | Reading and eventually returning the books need special components to track date, time, etc and hence need to be built. |

## Create a WBS for the entire functionalities in detail.

**WBS – Work Breakdown Structure**

A work-breakdown structure in project management and systems engineering, is a deliverable oriented breakdown of a project into smaller components. A work breakdown structure is a key project deliverable that organizes the team's work into manageable sections.



## 5: Do a rough estimate of effort required to accomplish each task in terms of person months.

Based on the above WBS, we calculate the effort required to accomplish each task in terms of person months. The procedure we have used is as follows:

1. Each task is divided into subtasks.
2. Each task including its subtasks is expected to have **900 lines of code** i.e., **0.9KLOC** (Kilo Lines of Code) on an average.
3. Also, assuming the project to be of **Organic type** due to small team size and well understood problem, the value of constants is **ab=2.4 and bb=1.05.**
4. The formula to calculate the **Effort in Person Months** for an Organic Project is given by:

**Effort E = ab \* (KLOC)bb**

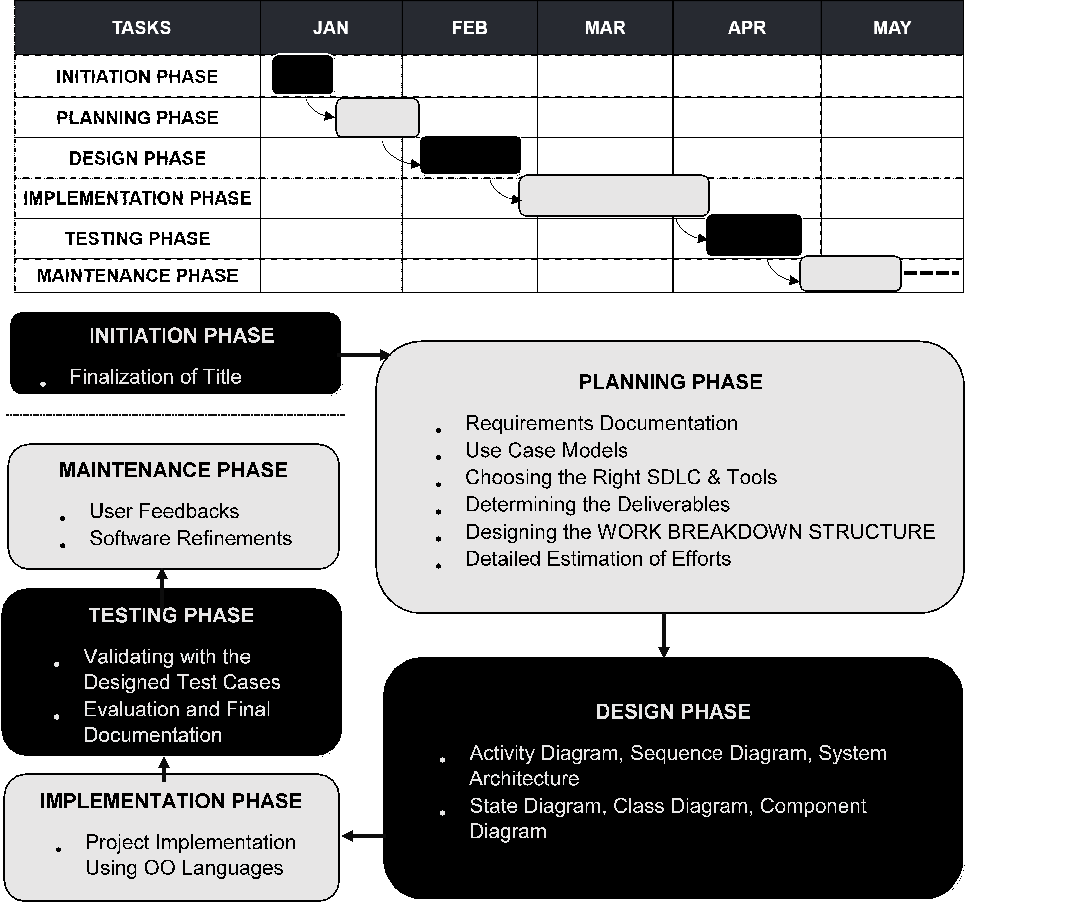
**Effort E = 2.4 \* (KLOC)1.05**

|  |  |  |
| --- | --- | --- |
| **TASKS** | **KILO LINES OF CODE** | **EFFORT E in PERSON MONTHS** |
| **1. Authentication** | **0.9** | **2.2** |
| **2. Search for Book** | **0.9** | **2.2** |
| **3. Borrow Book** | **0.9** | **2.2** |
| **4. Reading the Book** | **0.9** | **2.2** |
| **5. Bookshelf** | **0.9** | **2.2** |
| **6. Book Details** | **0.9** | **2.2** |
|  |  | **13.2 ~ 13** |

The sum of the calculation of Effort for all the individual tasks add up to **13.2 Person Months ~ 13 Person Months.**

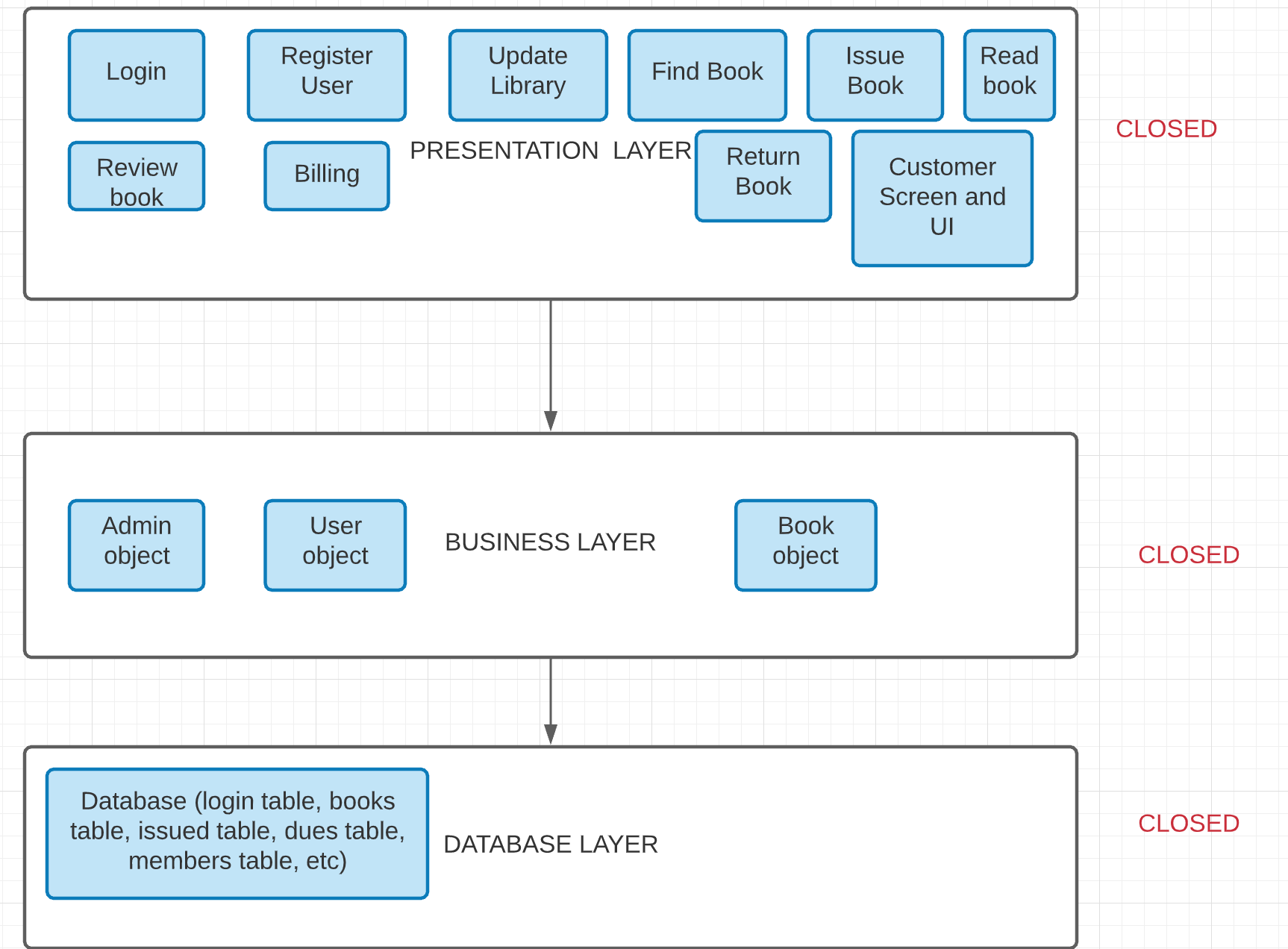
## Create the Gantt Chart for scheduling.

### Timeline



# Design Diagrams

## Software Architecture Diagram- Layered



We identify distinct classes of services that can be arranged hierarchically in a series of horizontal layers. Each layer performs a specific role within the application.

**ROLES OF EACH LAYER:**

**DATABASE LAYER:**

Simple Databases expanding up to SANs (Storage Area Networks)

**BUSINESS LAYER:**

Business logic is the programming that manages communication between an end user interface and a database.

**PRESENTATION LAYER:**

Presentation of web pages, end user interacting APIs, and functions delivered.

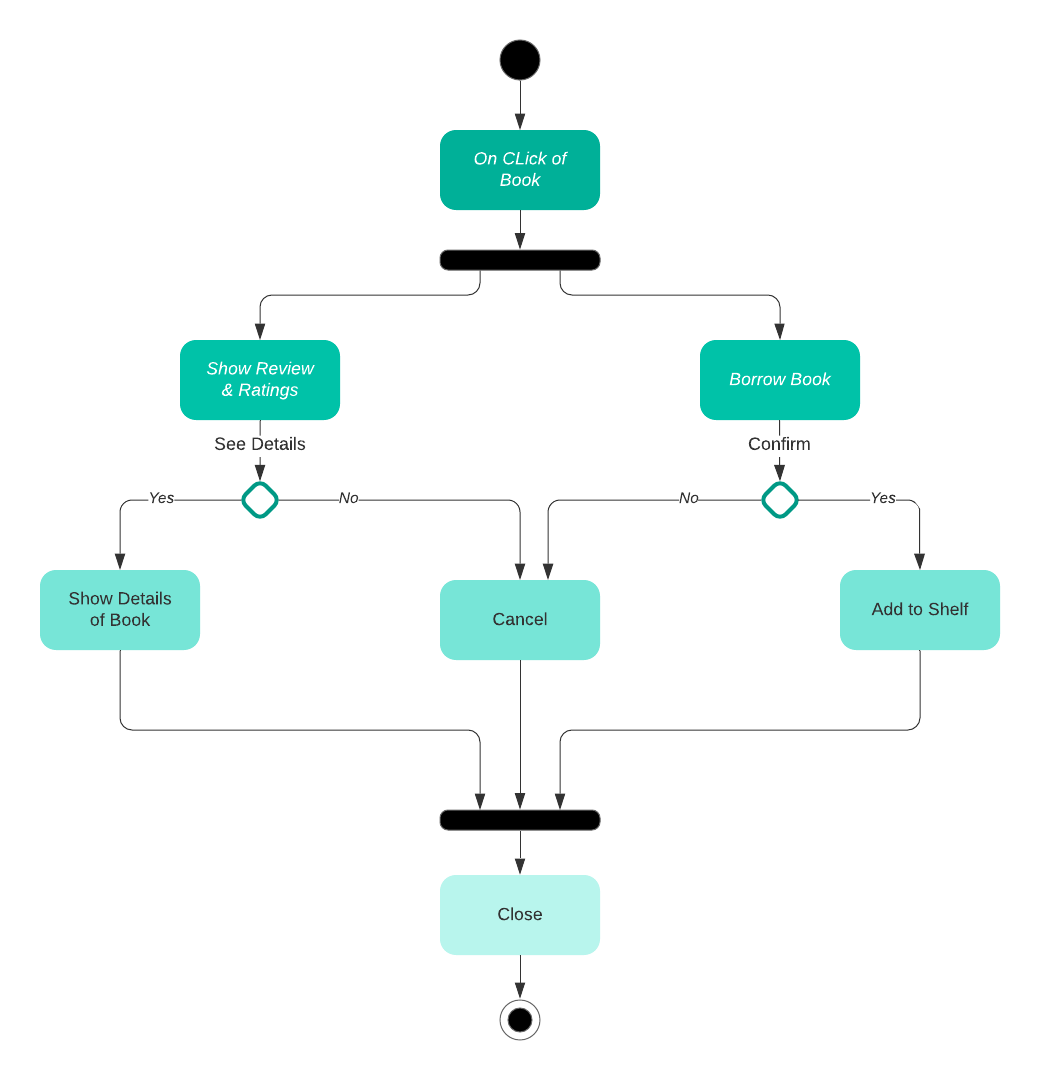
## Activity Diagrams

**What’s an activity diagram?**

Activity diagram is a behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

**Activity Diagram for ClickBook functionality:**

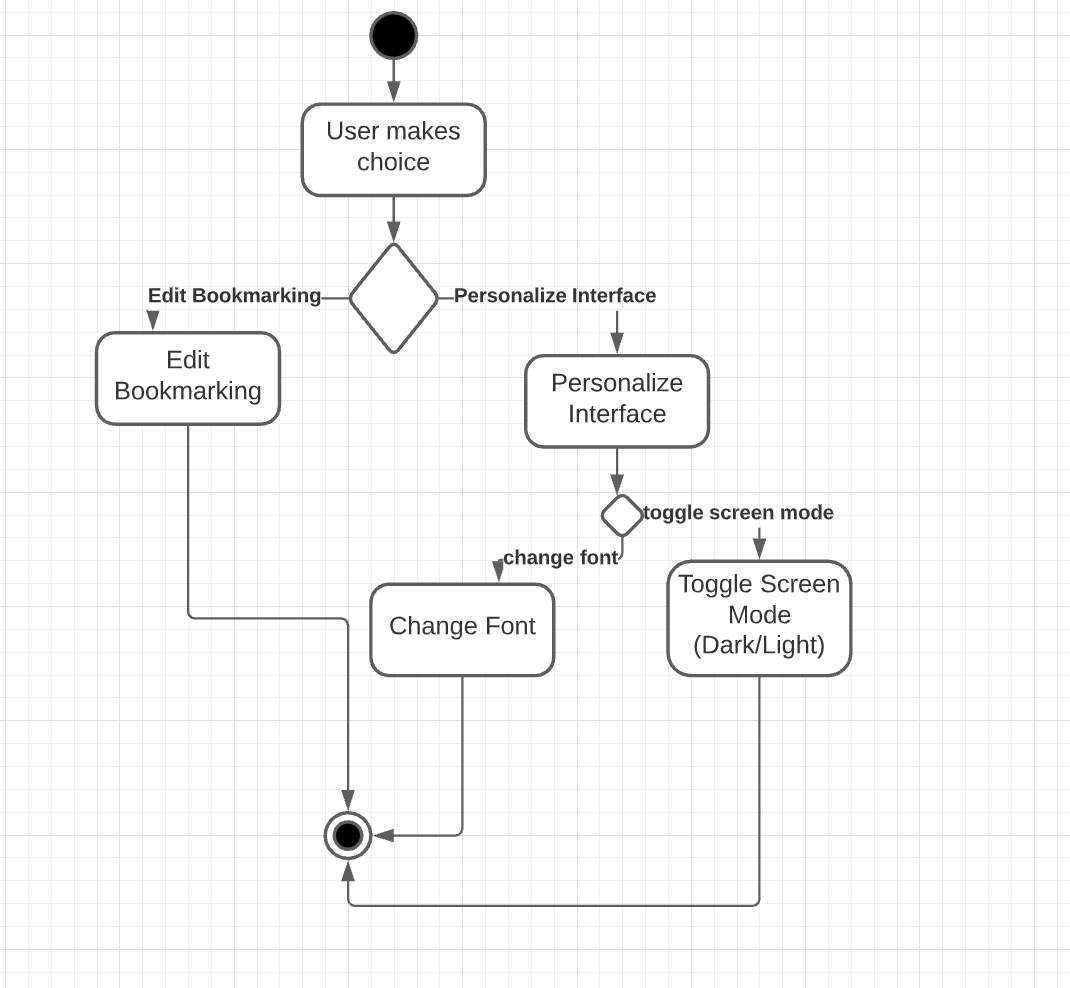
User may view reviews and ratings, details of book, or borrow book and add it to shelf on click of book.



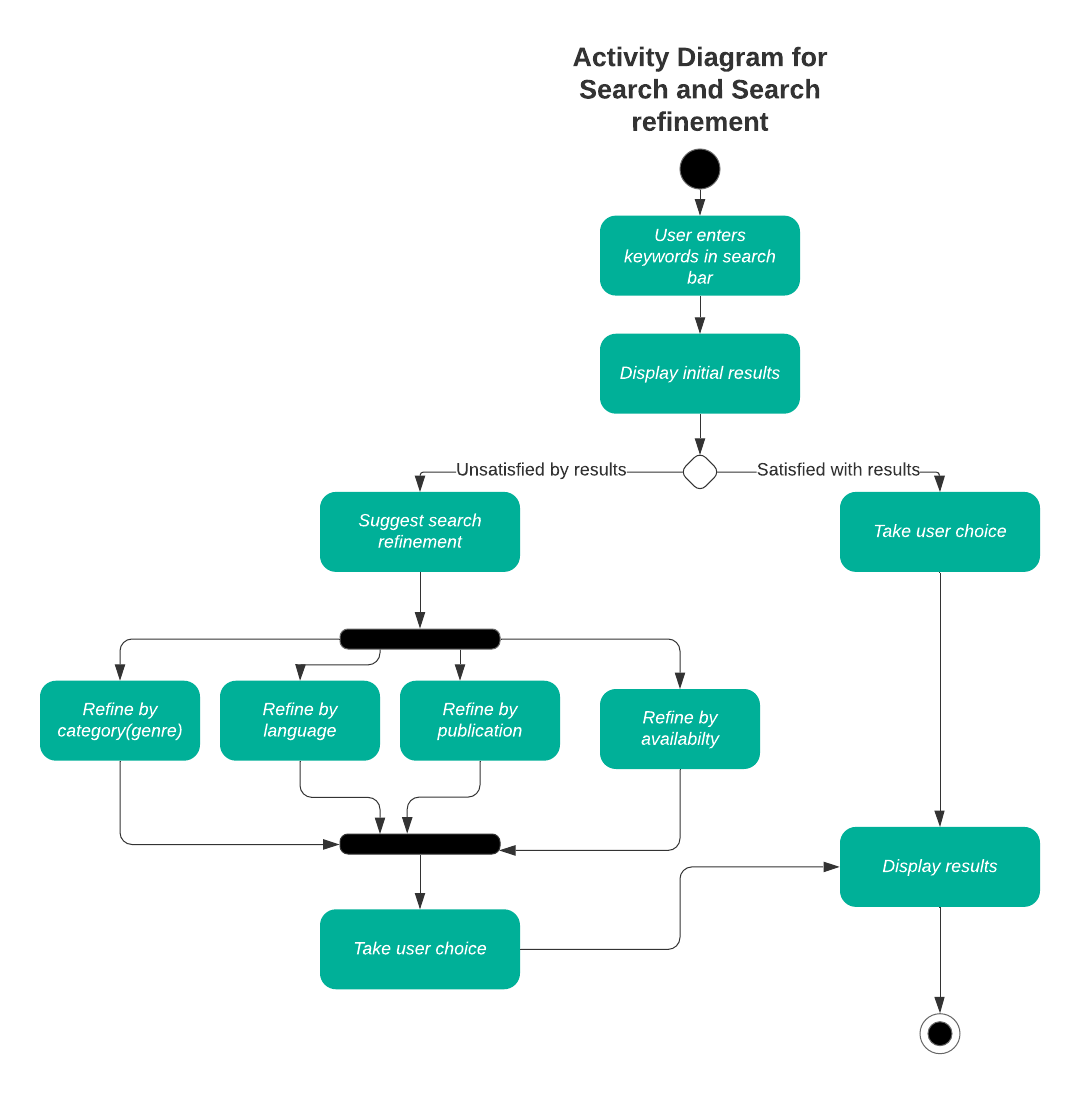
**Activity diagram for Personalization and Bookmarking:**

User may choose to edit bookmarking or personalize interface (change font or toggle screen mode).

NOTE: The Personalize Interface functionality falls beyond the scope of implementation of this project.



**Activity Diagram for Search and Search Refinement:**   
User may search for book based on selected refinements.



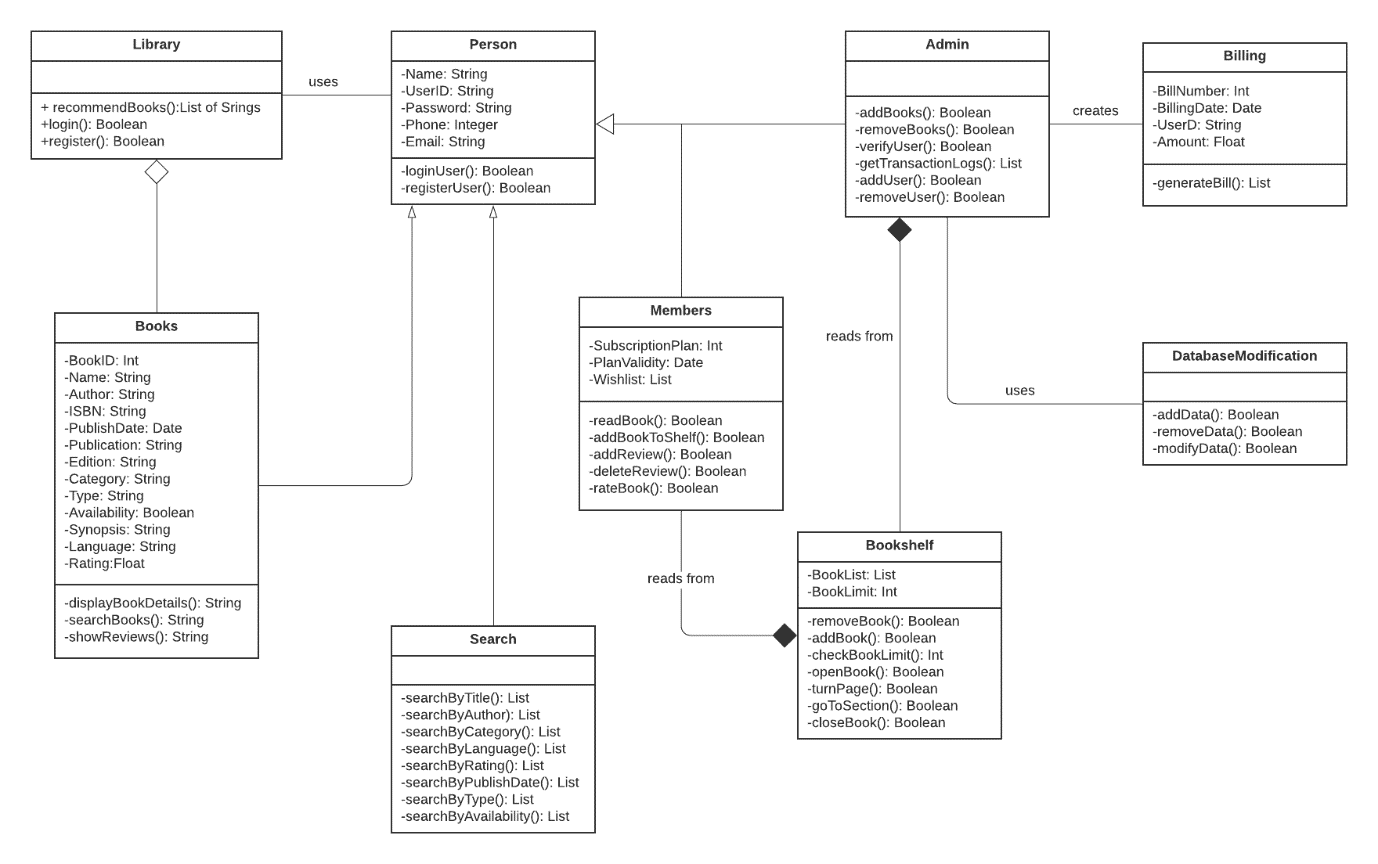
## Class Diagrams

**What is a class diagram?**  
The UML Class diagram is a graphical notation used to construct and visualize object-oriented systems. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's:

* classes,
* their attributes,
* operations (or methods),
* relationships among objects.

**Class Diagram for E-Book Management System:**

Depicts the various classes that comprise the system and the relationships between them.



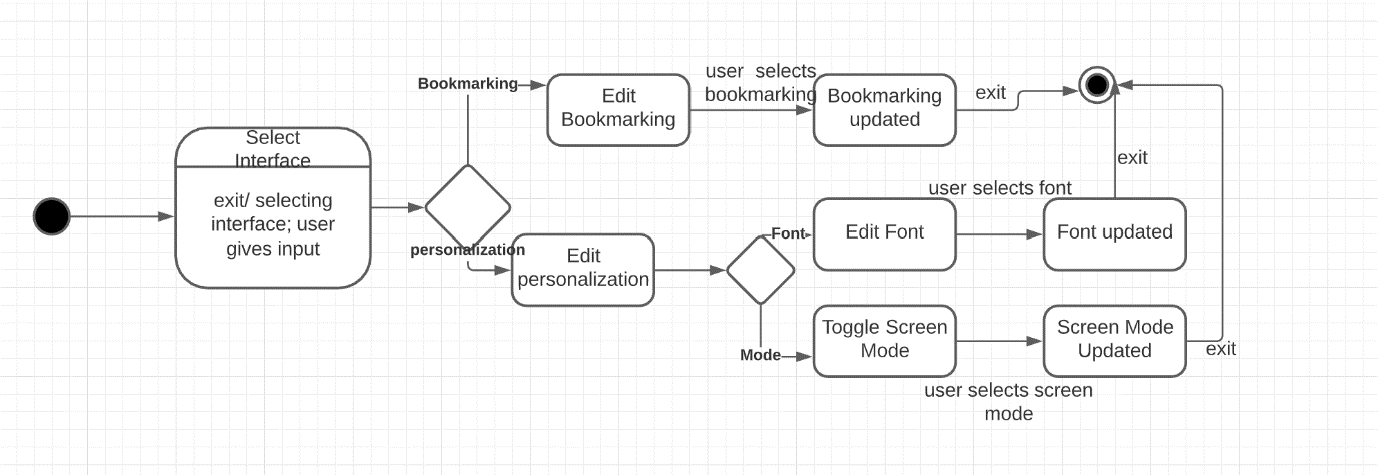
## State Diagrams

What is a state diagram?  
A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions.

**State Diagram for Personalization and Bookmarking:**

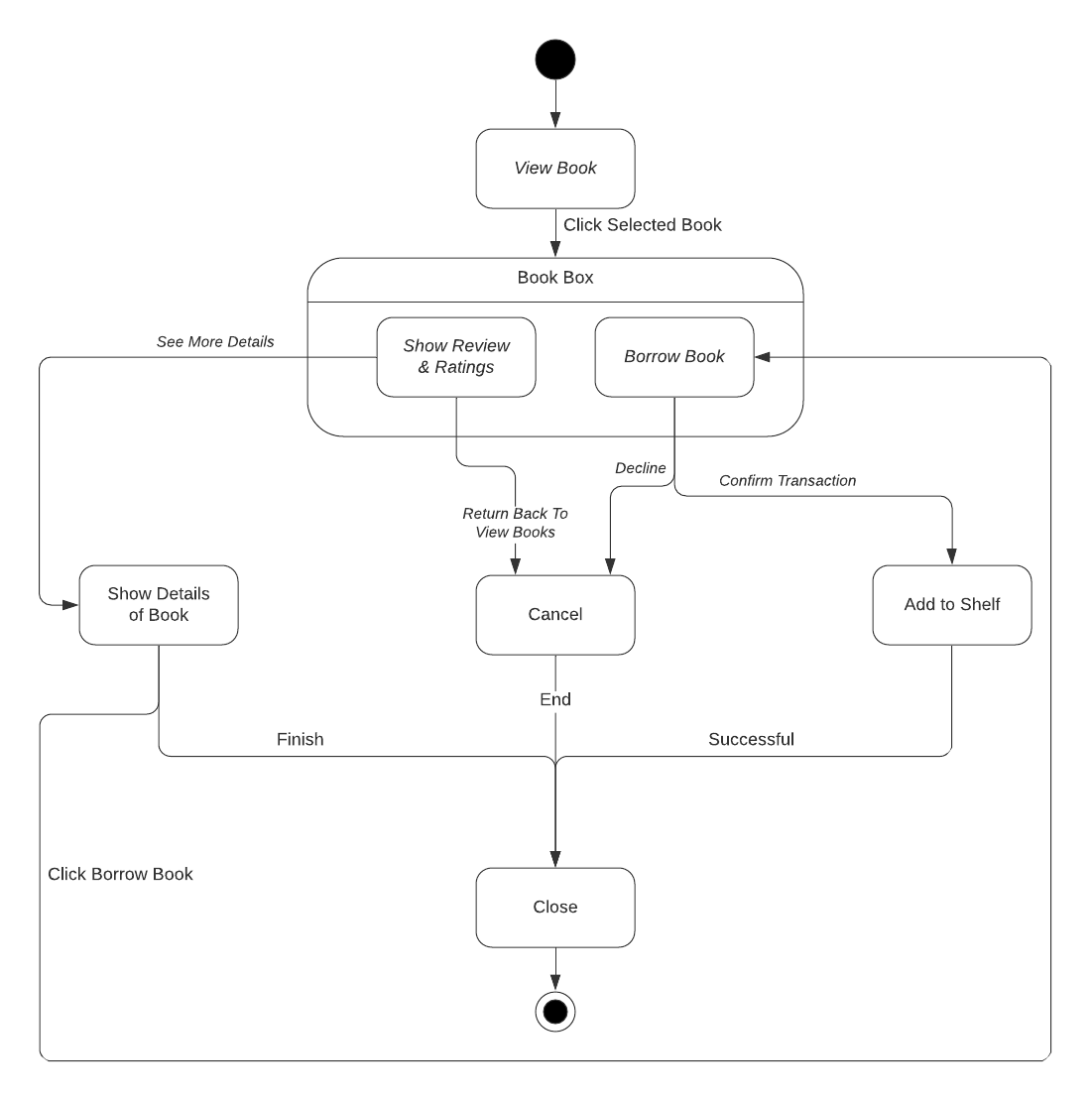
User may choose to edit bookmarking or personalize interfaces.

NOTE: The personalization functionality falls beyond the scope of implementation of this project.



**State Diagram for ClickBook functionality:**

User may view reviews and ratings, details of book, or borrow book and add it to shelf on click of book.



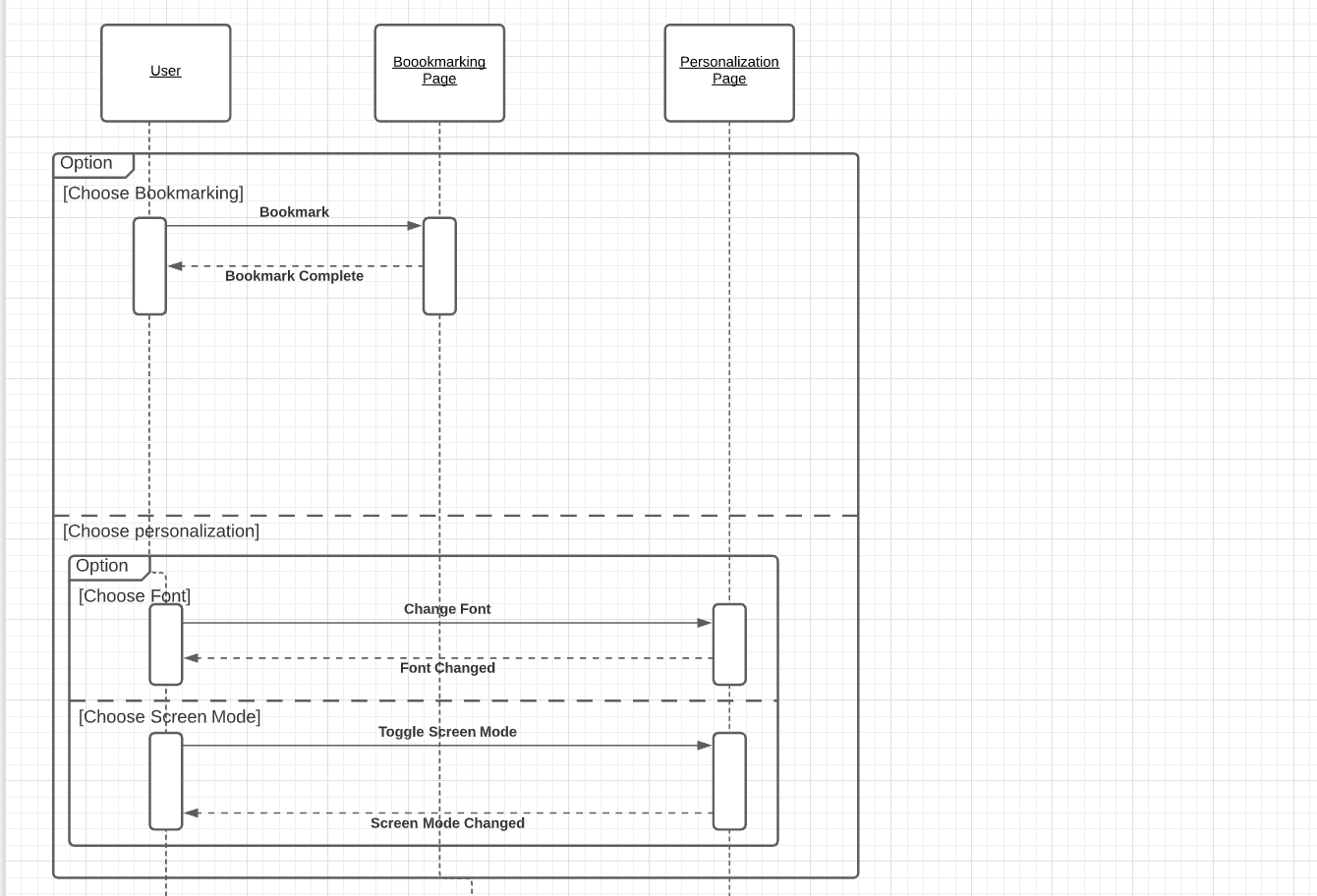
## Sequence Diagrams

What is a sequence diagram?  
A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place.

**Sequence Diagram for Personalization and Bookmarking:**

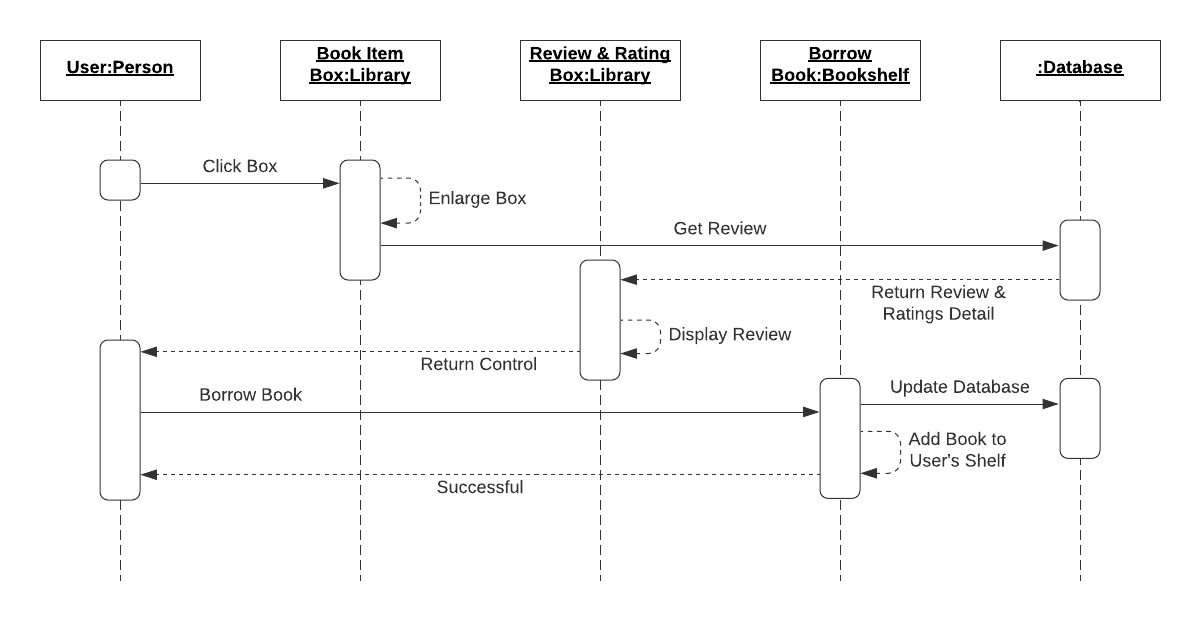
User may choose to edit bookmarking or personalize interfaces.

NOTE: The personalization functionality falls beyond the scope of implementation of this project.

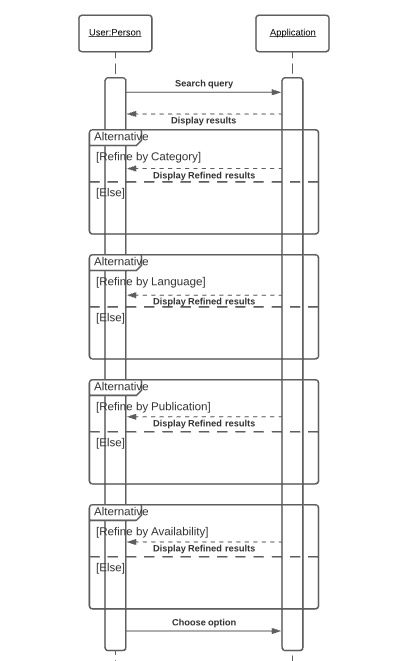


**Sequence Diagram for ClickBook functionality:**

User may view reviews and ratings, details of book, or borrow book and add it to shelf on click of book.



**Sequence Diagram for Search and Search Refinement:**   
User may search for book based on selected refinements.



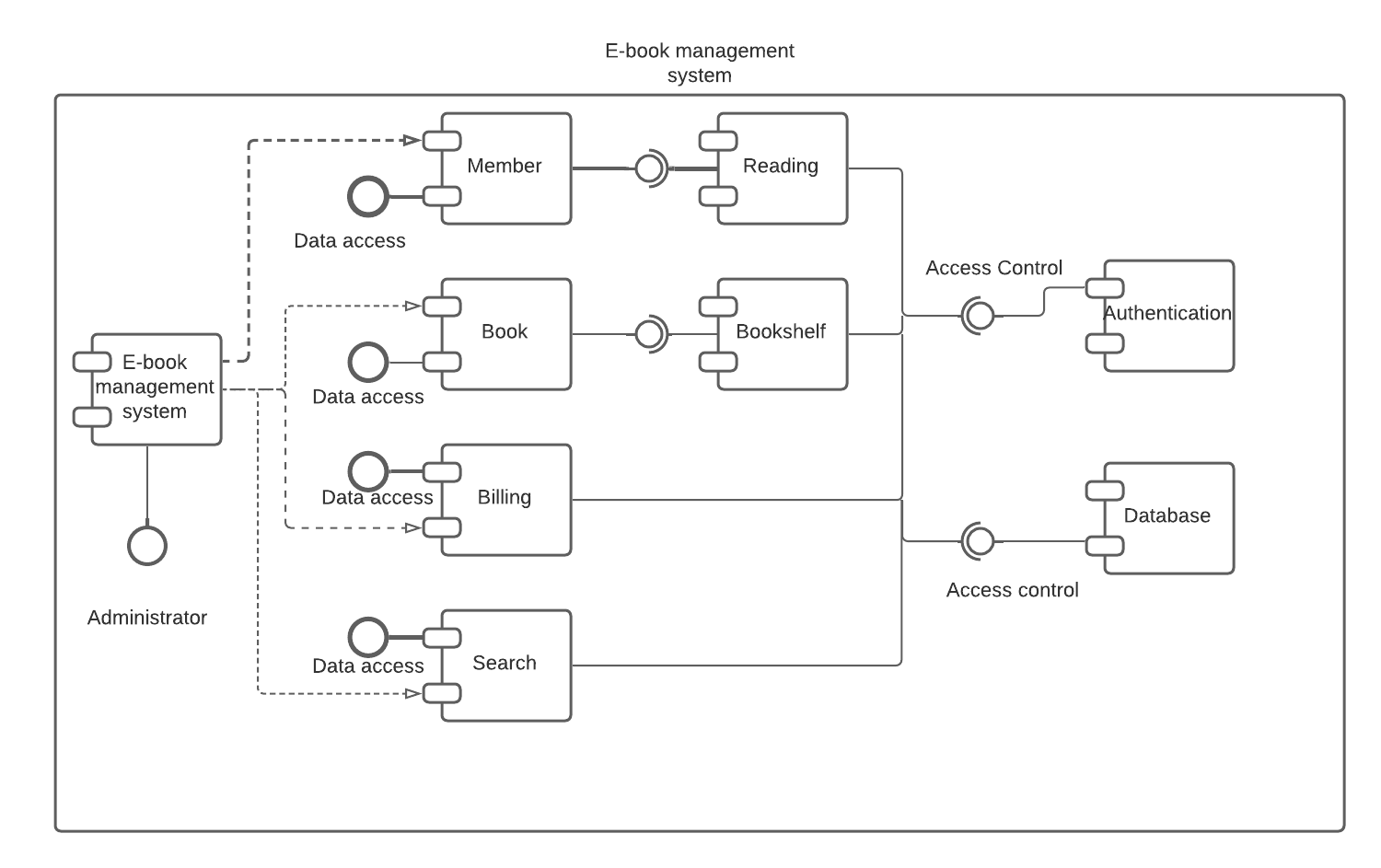
## Component Diagrams

What’s a component diagram?

In Unified Modeling Language (UML), a component diagram depicts how components are wired together to form larger components or software systems.

**Component Diagram of E-Book Management System:**

Shows the different components that build the system and how they are wired together to form the system as a whole.



# 

# Modules

|  |  |
| --- | --- |
| Module Name | Description |
| LoginRegister | This is a module that contains all the login and registration functionalities of the eBook Management System. It includes various use cases like authentication, registering a new user, and login to the user account. |
| Add Book (Admin) | An admin has the privileges to add new books to the database, which will be available to all the other users to borrow and read. |
| Remove Book (Admin) | An admin has the privilege of removing books from the library, but deleting it from the database. This removes the book from library and makes it unavailable to anyone trying to borrow this book from the library, if not already in possession. |
| Borrow Book | A user can request to borrow a book from the library. If the book is available, the users can add it to their own bookshelf and have a copy of that book. |
| Return Book | A user can return the book they possess, to the library. The library performs the required transaction process for the user for returning the book. |
| Find Book | Find Book interface is common to both the users and the admin. They can search for books based on just title, or many other refinements to get a specific search result. |
| Read Book | A user who has borrowed a book, will have access to read the book from their bookshelf. The book will be opened in the user’s choice of document viewer. |
| Review and Ratings | A user can leave a review and rating for each book they’ve read, either having it borrowed from the library or not. This helps other users to borrow the book based on their interest. |
| Maintenance | Admins can perform maintenance checks by directly going over the database or via GUI. |

# Test Cases

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Module Name | Test Case Description | Pre-conditions | Test Steps | Test Data | Expected Result | Actual Result | Test Result (Pass/Fail) |
| UT\_01 | LoginRegis ter | Test the register functionality | Open app | Enter name, email, username, and password | Name = John Doe Username = johndoe1928 Password = password [Email=jondoe@email.com](mailto:Email=jondoe@email.com) | Registration successful. | Register and login to user / admin page | Pass |
| UT\_02 | Name = Jane Doe  Username = janedoe  Password = password [Email=jondoe@email.com](mailto:Email%3Djondoe@email.com) | User already exists. | Show Error that user exists | Pass |
| UT\_03 | Enter name, email, and username | Name = Jane Doe  Username = janedoe [Email=jondoe@email.com](mailto:Email%3Djondoe@email.com) | Password Empty | Shows error for empty password field | Pass |
| UT\_04 | Enter name, email, username, and password | Name = John Doe Username = johndoe1920 Password = password | Password doesn’t meet guidelines. | Shows error message that password is invalid | Pass |
| UT\_05 | Enter name, username, and password | Name = John Doe Username = johndoe1920 Password = password | Email Empty | Shows error message that Email can’t be empty | Pass |
| UT\_06 | Enter email, username, and password | [Email=jondoe@email.com](mailto:Email%3Djondoe@email.com) Username = johndoe1920 Password = password | Name Empty | Shows error message that Name is Empty | Pass |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UT\_07 | LoginRegis ter | Test the register functionality | Open App | Enter name, and email | Name = John Doe [Email=jondoe@email.com](mailto:Email%3Djondoe@email.com) | Empty Fields | Shows error messages pertaining to  empty fields | Pass |
| UT\_08 | Test login functionality | Enter name, email, username, and password | Name = John Doe Username = johndoe1920 Password = p | Password doesn’t meet guidelines. | Shows error message that password is invalid | Pass |
| UT\_09 | Enter username, and password. | Username = janedoe Password = password | Login successful! | Redirects to user/admin’s page | Pass |
| UT\_10 | Enter wrong  password | Username = janedoe Password = fakepassword | Invalid Username or Password | Shows error message as invalid username or password | Pass |
| UT\_11 | Enter wrong  username | Username = johndoesPassword = password |
| UT\_12 | Enter  non-existent username | Username = johndoe7878 Password = password |
| UT\_13 | Enter only username | Username = janedoe Password = | Fields Cannot be empty | Shows message to enter login credentials | Pass |
| UT\_14 | Enter only password | Username =  Password = password | Pass |
| UT\_15 | Both Blank | Username= Password = | Pass |
| UT\_16 | Admin | Test add book functionality | Open app. Logged in as valid administrator | Go to Add Books Enter book ID, title, author name, and category | Bookid = 1029282 Title= Software Engineering Explained Author = John Doe Category = Education | Book added successfully | Book reflects in the database, shows message | Pass |
| UT\_17 | Admin | Test add book functionality | Open app. Logged in as valid administrator. | Go to Add Books. Enter book ID, title, author, category | Bookid = 1029282  Title= Software Engineering Explained Author = John Doe Category = Education | Book already exists. | No updates to database, shows message book exists | Pass |
| UT\_18 | Go to Add Books. Enter book ID | Bookid = | Missing Data | Shows error for missing data | Pass |
| UT\_19 | Go to Add Books. Enter category | Category = Education | Pass |
| UT\_20 | Go to Add Books. Enter book ID and title | Bookid = 1029282  Title= Software Engineering Explained | Pass |
| UT\_21 | Go to Add Books. Enter book ID author, and category | Bookid = 1029282  Author = John Doe Category = Education | Pass |
| UT\_22 | Go to Add Books. Enter book ID, title, author | Bookid = 1029282 Title= Software Engineering Explained Author = John Doe | Pass |
| UT\_23 | Go to Add Books. Blank |  | Pass |
| UT\_24 | Go to Add Books.  Enter book ID(different), and rest same | Bookid = 1029284 Title= Software Engineering Explained  Author = John Doe Category = Education | Book added successfully | Book reflects in the database, show message | Pass |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UT\_25 | Admin | Test find book functionality | Open app. Logged in as valid administrator. | Go to Search Enter bookID / availability | BookID=10238202 | Book found. | Displays Book details | Pass |
| UT\_26 | Go to Search. Enter bookID / availability | BookID=102382022 | Book Doesn't Exists | Shows message book doesn’t exist | Pass |
| UT\_27 | Go to Search. Enter title / availability | Title=Cooking With Me | Display list of all relevant titles | Displays list of books with matching titles | Pass |
| UT\_28 | Go to Search. Enter category / availability | Category = fiction | Display list of books with matching fiction | Displays list of books with matching fiction | Pass |
| UT\_29 | Go to Search. Enter Language / availability | Language = Greek | Display list of books with matching language | Displays list of books with matching language | Pass |
| UT\_30 | Go to Search. Enter Availability | Availability = available | Display list of all available books | Displays list of all available books | Pass |
| UT\_31 | Go to Search. Enter Publication / Availability | Publication = McGraw Hill | Display list of all books of that publication | Displays list of books of that publication | Pass |
| UT\_32 | Test remove book functionality | Go to Remove Books Enter book id to remove. | Book id=1929329 | Book removed. | Removes book from database | Pass |
| UT\_33 | Admin | Test remove book functionality | Open app. Logged in as valid administrator. | Go to Remove Books  Enter book id to remove. | Book id=19dfc34 | Book not found. | Displays error message | Pass |
| UT\_34 | Go to Remove Books  Enter book name, book id to remove. | Bookid = 1029284 Title= Software Engineering Explained | Book removed | Displays message book removed | Pass |
| UT\_35 | Go to Remove Books Enter book name to remove | Title= Software Engineering Explained | Enter Book ID | Displays message to enter book id | Pass |
| UT\_36 | User | Test add book functionality | Open app. Logged in as a valid user. | Go to Browse books. Search for relevant book | Bookid = 1029282  Title= Software Engineering Explained Author = John Doe Category = Education | Book to be added to user’s shelf | Book is added to the user’s shelf | Pass |
| UT\_37 | Bookid = 1029282  Title= Software Engineering Explained Author = John Doe Category = Education | Book Present in user’s shelf | Shows that book is already present | Pass |
| UT\_38 | Test search book functionality | Go to Search. Enter bookID / availability. | BookID=10238202 | Book found. | Display book/s to user | Pass |
| UT\_39 | BookID=102382020 | Show it is Invalid book. | Displays message for invalid book | Pass |
| UT\_40 |  | Go to Search. Enter Title / availability | Title = Cooking with Me | Display list of all relevant titles | Displays the list of books of the same title | Pass |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UT\_41 | User | Test search book functionality |  | Go to Search. Enter Category / availability | Category = fiction | Display list of books of relevant Category | Displays list of books of the same  category | Pass |
| UT\_42 | Go to Search. Enter Language / availability | Language = Greek | Display list of books of relevant language | Displays books of that language | Pass |
| UT\_43 | Go to Search. Enter availability | Availability = available | Display list of all available books | Displays all books available | Pass |
| UT\_44 | Go to Search. Enter Publication / availability | Publication = McGraw Hill | Display list of books of relevant publication | Displays all the books from that  publisher | Pass |
| UT\_45 | Go to Search. Enter Author / availability | Author = Jondoe | Display list of books of relevant author | Displays all the books from that author | Pass |
| UT\_46 | Test remove book functionality | Go to Bookshelf Click on book Click on remove book | Book id=1929329 | Book to be remove from user’s shelf | Book removed from user’s shelf | Pass |
| UT\_47 | Go to Bookshelf Click on book Click on remove book | Book id=19dfc34 | Book not present in user’s shelf | Book not in user’s shelf not shown | Pass |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UT\_48 | User | Test the add reviews and ratings. | Open app; logged in as valid user. | Click on Book, Add Review Enter review and rating | Bookid=19832029 Review=”excellent read!” Rating=5 | Display other reviews and add the new review and rating. | Displays all the old review, and adds the new review and rating | Pass |
| UT\_49 | Bookid=19832026 Review=”excellent read!” Rating=5 | Unavailable book should not be shown | Unavailable book not displayed, no review section | Pass |
| UT\_50 | Click on Book, Add Review Enter rating | Bookid=19832029 Rating=5 | Display other reviews and add the new rating. | Displays all the old review, and adds the new review and rating | Pass |
| UT\_51 | Click on Book, Add Review Enter review | Bookid=19832029 Review=”excellent read!” | Display other reviews and add the new review and rating. | Displays all the old review, and adds the new review and rating | Pass |
| UT\_52 | Click on Book, Add Review Blank Entry |  | Show Error, no blanks allowed | Displays error message that blank reviews cannot be submitted | Pass |
| UT\_53 | Test the read book functionality | Click book on bookshelf | Bookid=198320 | Book should open to read | Book displayed to be read | Pass |
| UT\_54 | Bookid=1980102929 | Book not in user’s shelf should not be shown | Book isn’t shown | Pass |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UT\_55 | User | Test the edit bookmarking functionality | Open app; logged in as valid user. | Click book on bookshelf Click on Edit Bookmarking | Bookid=19802392 Page=5 | Bookmarking edited successfully. | Bookmark modified | Pass |
| UT\_56 | Enter bookID Click on Edit Bookmarking | Bookid=198293829 Page=6 | Book not in user’s shelf should not be displayed | Book unavailable in user’s shelf | Pass |
| UT\_57 | Test the show reviews and ratings, and display book details functionality | Go to Search Book/Bookshelf (Enter Book id) Click on Book | Bookid=19832029 | Display all previous ratings and reviews, and also displays book details. | Displays book info and all reviews and ratings | Pass |
| UT\_58 | Bookid=19119822 | Book not found (On entering ID) | Displays book not found message (On entering ID) | Pass |
| UT\_59 |  | Test add book functionality | Go to Search Book Enter Search details Click on Book Click on add book | Bookid=19803902 | Book successfully added to shelf | Adds book to user’s shelf | Pass |
| UT\_60 |  | Go to Search Book Enter Search details Click on Book Click on add book | Bookid=1980390123 | Book not found | Displays message book not found | Pass |

# Output Screenshots

|  |  |
| --- | --- |
|  |  |
| Login Page | Login Page – Error |
|  |  |
| Registeration Page | Registeration Page – Error |

|  |
| --- |
|  |
| Admin – GUI (Main Page, Add Books, Remove Books, Transaction Logs) |

|  |
| --- |
|  |
| User – GUI (Browse, Bookshelf) |

|  |
| --- |
|  |
| User – GUI (Subscription Page) |

|  |
| --- |
|  |
| Admin/User – GUI (Search) |