Booked: A Book Club Application
Software Requirements Specification

CSE 682 Software Engineering

Syracuse University

Winter 2024

Hannah Ashton

Sarah Welvaert

Nicholas Maurer

Giselle Razo

Table Of Contents

Preface

This document specifies the detailed requirements to build an interactive book club application for book lovers to use. The Software Requirements Specification (SRS) is the official statement of what the system should implement.

Introduction

This application is inspired by the need for a more interactive book reading application for book lovers to not only review books, but to also engage with other readers in the process. The application described in this SRS will emphasize the social networking/discussion forum aspect of a book club. This design was favored because of the prevalence of applications developed for readers that feature book tracking and reviewing functionality. However, the majority of book related applications lack the ability for users to discuss books with groups within the application, and to connect at a scheduled time for more meaningful conversations and social interaction. The application outlined below will also have a discussion forum about the books where readers can connect with others with similar literary interests to them. Finally, the application will have a shared calendar for users to organize discussion forums where users can invite other registered users for a more in depth discussion about literature. In addition to these innovative features, the application is inspired by the success of related applications and will have a virtual bookshelf so readers can keep track of books they are interested in reading at any moment of time.

Document Conventions

In this document, the verbs “shall” and “should” are used to indicate which requirements are necessary versus recommended. Requirements that the system shall meet are considered necessary for the minimum functionality. Requirements that the system should meet are considered recommended but not necessary.

Each requirement is numbered in the following fashion such that the user requirements and their corresponding system requirements align. A strikethrough indicates that the requirement is no longer relevant.

UR 4: Sample user requirement.

SR 4.1: First sample system requirement that corresponds to UR 4.

SR 4.2: Second sample system requirement that corresponds to UR 4.

The functional requirements are also divided into four sections (user account, book database, book club discussions, and calendar) based on the general feature they relate to.

Each user requirement includes a priority rating from 1 to 4. The highest priority is 1, meaning that the requirement should be implemented first. The priority level 2 will be implemented next, and so on.

User Requirements Definition

Functional User Requirements

Section 1: User Account

UR 1: The system shall allow new users to create a new account. (Priority: 2)

UR 2: The system shall allow existing users to sign into their account. (Priority: 2)

Section 2: Bookshelves

UR 3: The system shall allow users to have a bookshelf of saved books. (Priority: 1)

UR 4: The system shall allow the user to categorize books in a read, to-be-read, or currently reading bookshelf. (Priority: 1)

UR 5: The system shall allow users to create custom bookshelf categories. (Priority: 4)

UR 6: The system shall allow users to store book reviews. (Priority: 4)

UR 7: The system shall allow users to store all notes related to individual books. (Priority: 4)

Section 3: Book Club Discussions

UR 8: The system shall allow users to create groups with other registered users. (Priority: 3)

UR 9: The system shall allow users to post text and data on group/book club page channels. (Priority: 3)

UR 10: The system shall keep a record of all of the messages and data made by users on a per book club basis. (Priority: 3)

UR 11: The system shall notify users within their respective book club when a new message is received. (Priority: 3)

Section 4: Calendar

UR 12: The system shall have a calendar feature. (Priority: 4)

UR 13: The system shall provide the user with the ability to create club meetings. (Priority: 4)

Non-Functional User Requirements

UR 14: The system shall be secure.

UR 15: The system shall be reliable and responsive.

UR 16: The system shall limit the capacity to a manageable size.

System Requirements Specification

Functional System Requirements

Section 1: User Account

UR 1: The system shall allow new users to create a new account.

~~SR 1.1: The system shall welcome the user and ask them to complete a registration form.~~

~~SR 1.2: The system shall accept the user's first and last name, email address, and password for this account.~~

~~SR 1.3: The system shall check to see if the user's email is already associated with an account; if it is, the system shall prompt the user to sign in instead.~~

~~SR 1.4: The system shall allow the user to submit their information once the form is completed.~~

~~SR 1.5: Once the user's information has been submitted, the system shall add the user's information to the data access system.~~

SR 1.6: The system shall bring the user to their new account page after account creation.

SR 1.7 The system shall prompt new users to log in via a Github account.

UR 2: The system shall allow existing users to sign into their account.

~~SR 2.1: The system shall allow the user to input their email address and password and click "Submit".~~

~~SR 2.2: The system shall compare the email address and password with the data access system.~~

~~SR 2.3: If the user's credentials do not match the database values, the system shall notify the user and allow them to try again.~~

~~SR 2.4: The system shall have a "Forgot Password" option for the user to reset their password.~~

SR 2.5: If the user's credentials are validated, the system shall open the user's account.

SR 2.6: The system shall prompt existing users to enter their Github username and password.

Section 2: Bookshelves

UR 3: The system shall allow users to have a bookshelf of saved books.

SR 3.1: The system shall store title, author name, ISBN, publishing date for each book.

SR 3.2: The system shall store data pertaining to the author including name, books published, and biography.

SR 3.3: The system shall allow for the search of data pertaining to books including title, ISBN, and author and the ability to save the book in a category.

SR 3.4: The system shall notify users of an invalid search query if no matches are found.

SR 3.5: The system shall allow users to add a new book by inputting the title, author name, ISBN, and publishing date.

SR 3.6: The system shall allow users to edit existing books that they have added.

SR 3.7: The system shall allow users to delete existing books that they have added.

UR 4: The system shall allow the user to categorize books in a read, to-be-read, or currently reading bookshelf

SR 4.1: The system shall allow users to store a book in any category.

SR 4.2: The system shall allow users to move books to different categories.

UR 5: The system shall allow users to create, view, update, and delete custom bookshelf categories.

SR 5.2: The system shall allow users to create a new bookshelf and give it a custom title.

SR 5.3: The system shall allow users to update the title of their custom bookshelves.

SR 5.4: The system shall allow users to add and remove books from their custom bookshelves once created.

SR 5.5: The system shall allow users to have the same book in multiple bookshelf categories.

SR 5.6: The system shall allow users to delete their custom bookshelves.

UR 6: The system shall store all user book reviews.

SR 6.1: The system shall allow the user to create new book reviews.

SR 6.2: The system shall allow the user to modify their previously created reviews.

SR 6.3: The system shall allow users to delete their previously created reviews.

SR 6.4: The system shall allow users to view all reviews on any book.

SR 6.5: The system shall associate each review with a specific individual account according to who created it.

SR 6.6: The system shall display the date and time that the review was created.

SR 6.7: The system shall have the ability to remove reviews based on community guidelines and store the number of infractions against a user.

UR 7: The system shall store all notes related to individual books.

SR 7.1: The system shall display the date and time of the note.

SR 7.2: The system shall display the page number of the book associated with the note.

SR 7.3: The system shall give the user the option to set a note to private or public.

Section 3: Book Club Discussions

UR 8: The system shall allow users to create groups with other registered users.

SR 8.1: The system shall allow users to create new groups and register them as public (visible in searches and open to new members) or private (by invitation only).

SR 8.2: The system shall provide an interface of groups that a user has joined or created, with the ability to enter these group pages.

SR 8.3: The system shall provide a separate interface to allow users to browse public groups.

SR 8.4: The system shall allow users to request to join any public group.

SR 8.5: The system shall maintain an association between individual users and their groups.

SR 8.6: The system should allow groups to have moderators who have the additional privileges of being able to remove group members and edit the discussion channel.

SR 8.7: The system shall allow users to delete any group they have created or are the moderator of.

SR 8.8: The system shall allow users to leave any group they have previously joined.

SR 8.9: The system shall allow users to update the name of any group they have created or are the moderator of.

UR 9: The system shall allow users to post text and data on channels of users grouped into book clubs.

SR 9.1: The system shall create a discussion forum upon book club creation.

SR 9.2: The system shall allow for text and graphics to be posted to the discussion forum via keyboard and touchscreen inputs.

SR 9.3: The system shall limit the visibility of the messages to only the members of the group where the message was posted.

UR 10: The system shall keep a record of all of the messages and data made by users on a per book club basis.

SR 10.1: The system shall allow users to view messages previously posted to the discussion forums.

SR 10.2: The system shall maintain a record of all chats in the discussion forums in the database.

SR 10.3: The system shall allow users to flag harmful or disruptive chats posted to the discussion forums.

SR 10.4: The system should provide security to limit access to the database to only users with specified access privileges. (Non-functional)

UR 11: The system shall notify users within their respective book club when a new message is received.

SR 11.1: The system shall prompt the user to allow the app to send them push notifications.

SR 11.2: The system shall display a brief preview of the chats received to all members of the group including the name of the group the chat was received in, the name of the member who posted the message, and the first 20 characters of the message.

SR 11.3: The app should display a typing icon when a member of the chat is in the process of typing a message in the group discussion forum.

Section 4: Calendar

UR 12: The system shall have a calendar feature accessible to users.

SR 12.1: The system shall provide the user to access a calendar for the user to coordinate book club events.

UR 13: The system shall provide the user with the ability to create club meetings.

SR 13.1: The system shall allow the user to add recipients to meetings.

SR 13.2: The app shall allow the user to add a book club topic to the meetings.

SR 13.3: The app shall provide the invitees of book club meetings to RSVP events.

SR 13.4: The app shall provide the user with the ability to modify newly created meetings.

Non-Functional System Requirements

UR 14: The system shall be secure.

SR 14.1: The system shall log the user off if a user is inactive for 10 minutes.

SR 14.2: The system shall use encryption for users' credentials and store them hashed with additional salting.

SR 14.3: The system shall allow users to delete their account, removing all customer's data from the system.

SR 14.4: The system shall allow access only to authorized users using role-based access control.

UR 15: The system shall be reliable and responsive.

SR 15.1 : The system shall not crash more than once per month for any given user.

SR 15.2: The system should be responsive to the user's touch/click within 250 ms.

SR 15.3: For all users, the average number of help requests should not exceed 2 per month.

SR 15.4: Maintenance outages for the system shall not exceed 4 hours total per month.

UR 16: The system shall limit the capacity to a manageable size.

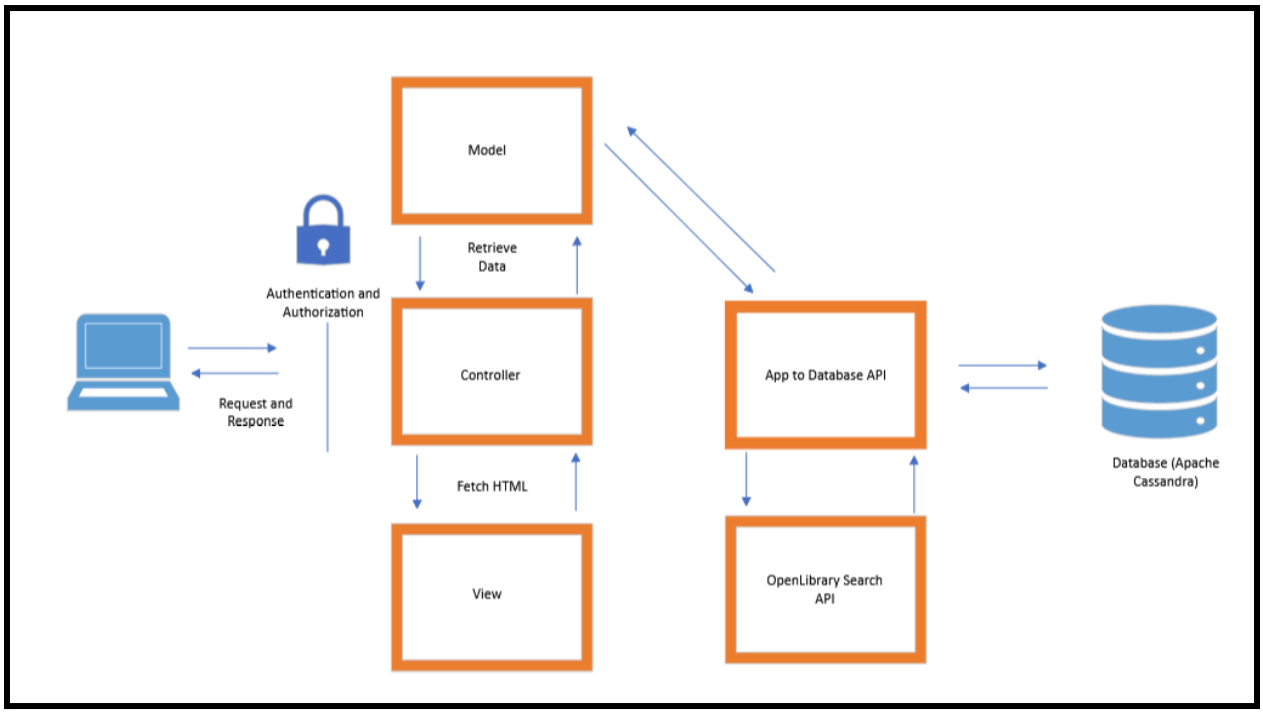
SR 16.1 : The system shall limit discussion posts to a maximum of 1,000 characters.

SR 16.2: The system shall notify the user when the book database is at maximum capacity (5 TB) and no more books can be added.

SR 16.3: The system shall allow 10,000 logged-in users at once without performance degradation.

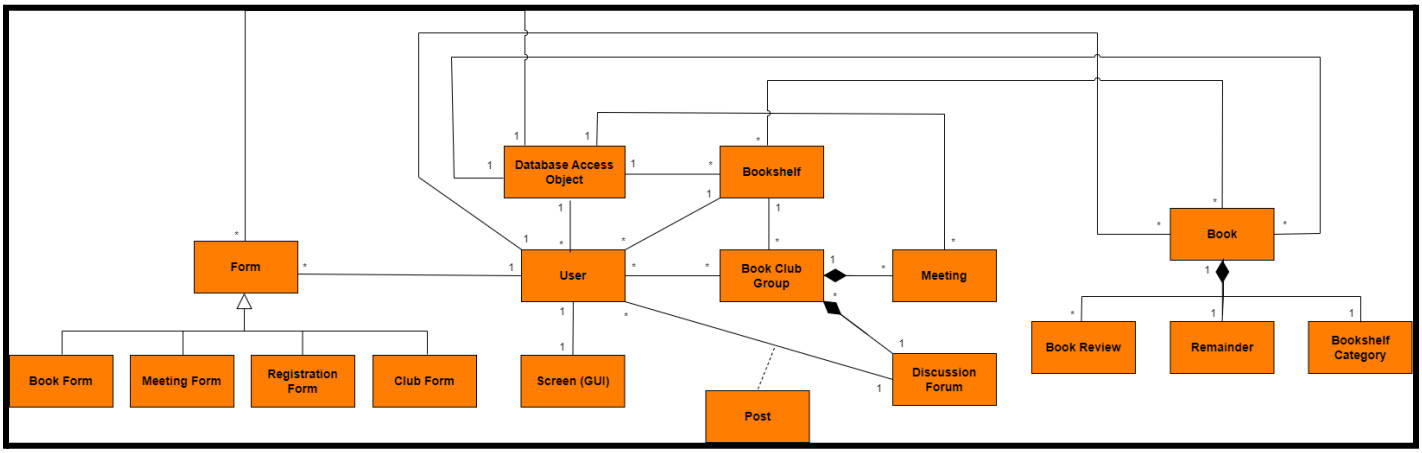
System Models

System Architecture

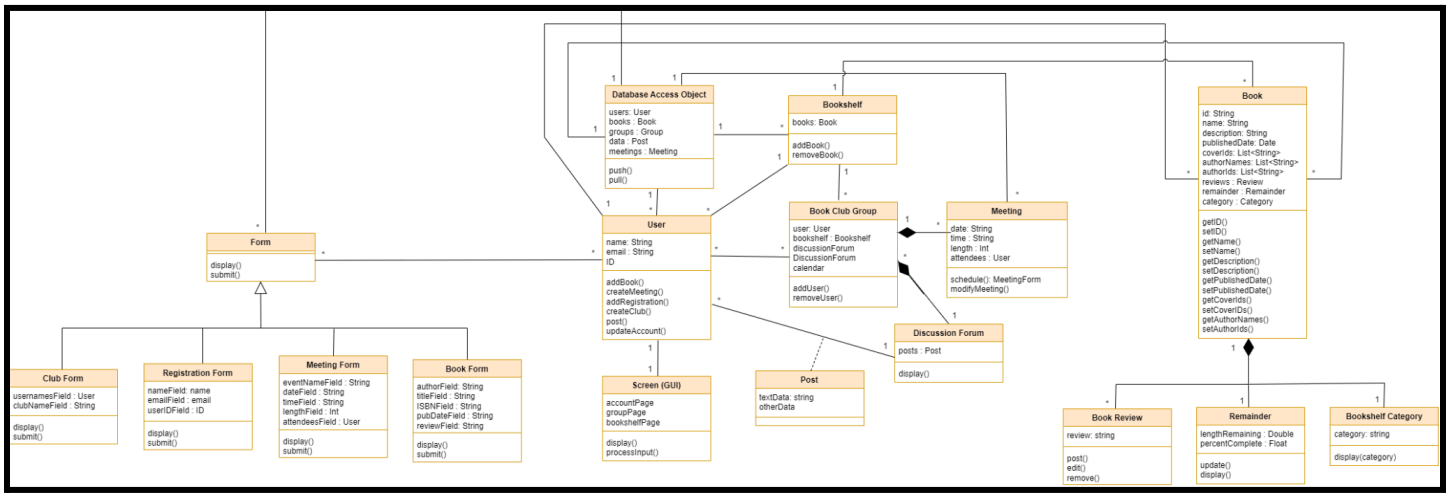


Similar to many other web applications, the Booked system will be structured as a Model-View-Controller architecture, since the layers are clear components and can be maintained and tested easily. The root of this design is the Apache Cassandra database, which will hold the Booked data in a distributed system. This database is connected to the Model-View-Controller structure via an App to Database API from the OpenLibrary Search API. Within the system, the Model layer of the architecture is responsible for knowing what types of data are contained in the database and retrieving these from the API and database. The Controller layer then converts the types of data into the usable information to be presented through the View layer, which contains the user interface. Authentication and authorization ensure the system is accessed securely, and requests and responses allow the user to engage.

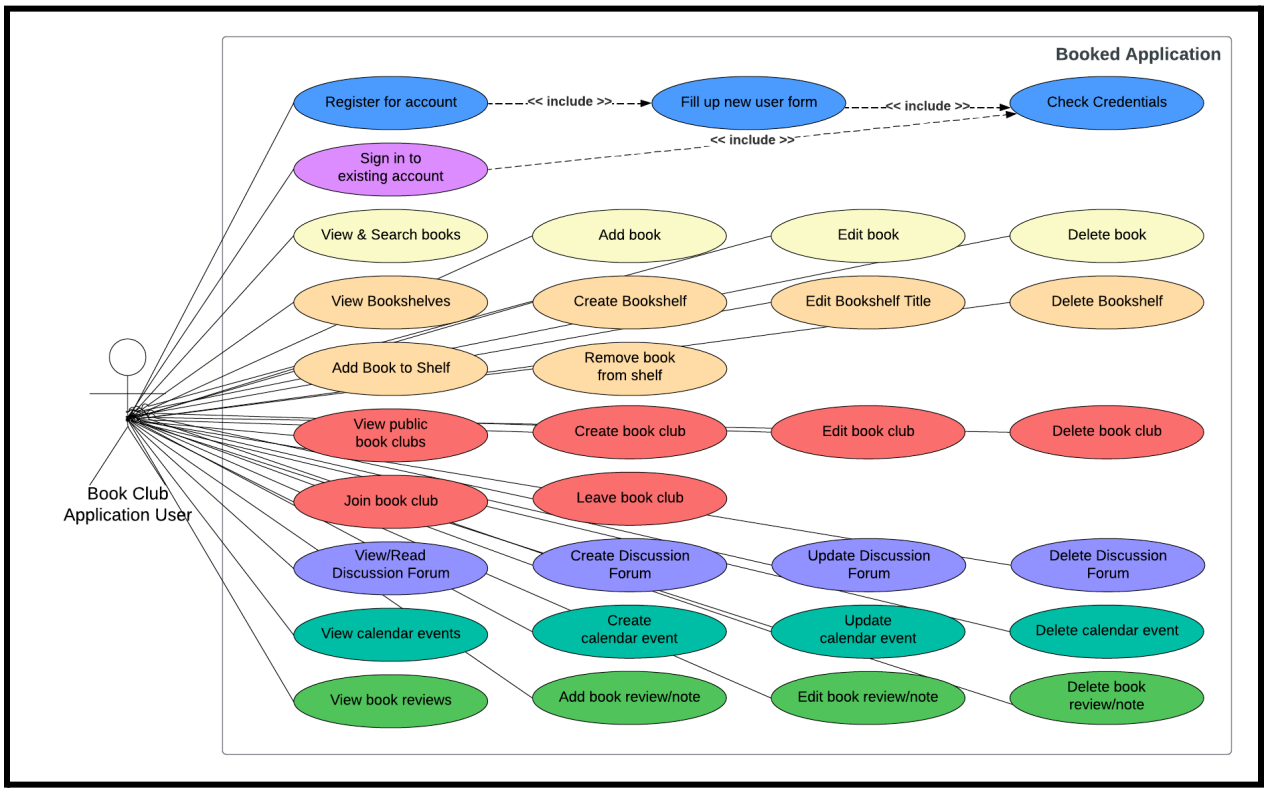
High-Level Class Diagram



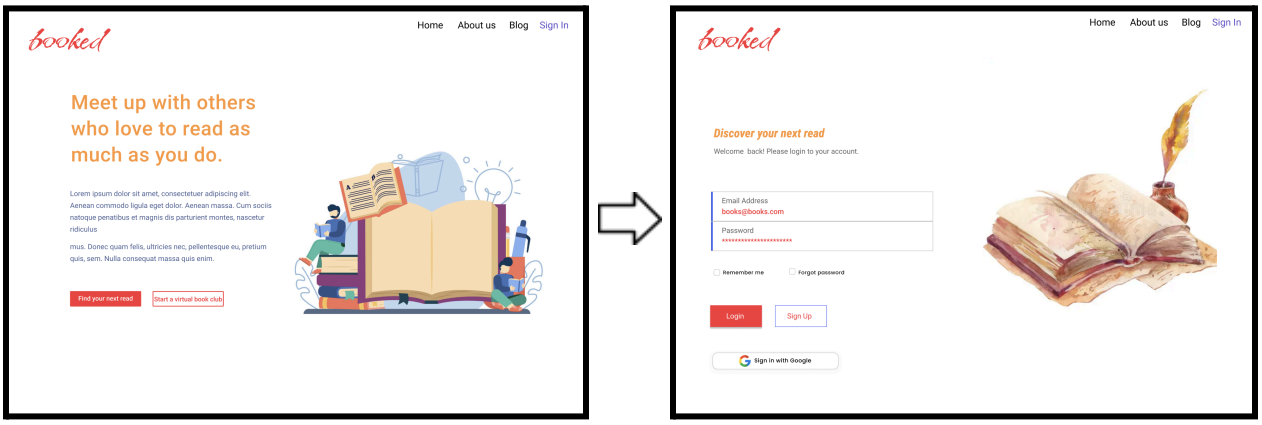
Low-Level Class Diagram



High-Level Use Case Diagrams



Application Prototype



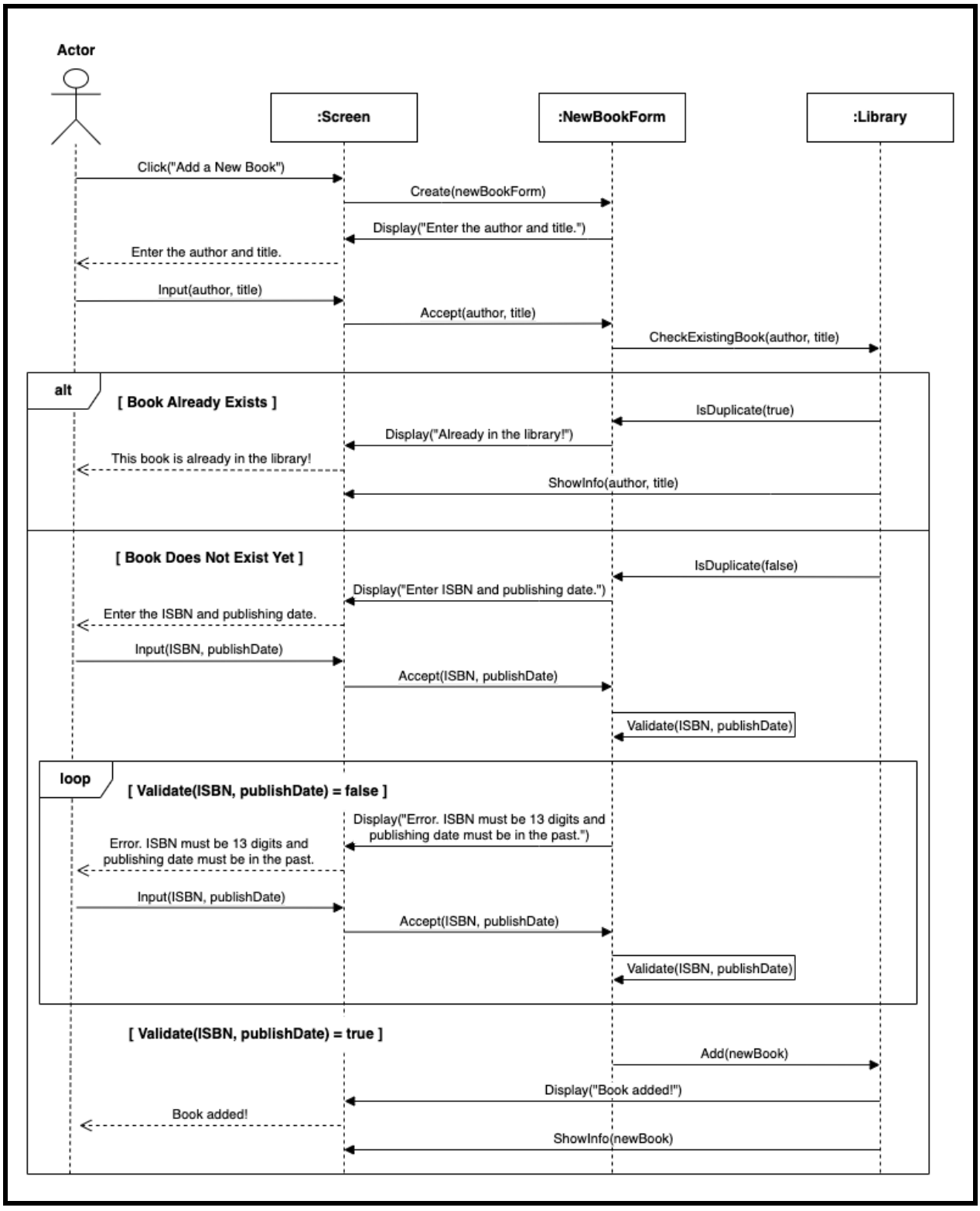
Use Cases and Sequence Diagrams

Add Book: Detailed Use Case

Title:	Add Book
Description:	The goal and context of this use case is to describe the process of a user adding a new book to the Booked system. The user inputs the book's details, including the title, author name, ISBN, and publishing date. The book is then added to the system's book collection for any user to access.
Actors:	System user.
Stimulus (Trigger):	A system user selects "Add a New Book" from the Book Search page.
Preconditions:	The user has logged into their account and navigated to the Book Search page.
Postconditions:	The user is brought to the information page for the book they have just added. The new book, as entered by the user, is now available for any user to view, review, discuss, or save to a shelf.
Main Success Scenario:	1. On the Book Search page, the user clicks "Add a New Book". 2. The system then provides the user with a blank form, prompting the user to enter the author name and book title. 3. The user enters the author and title, and then clicks "Add Book". 4. If there is not a book with the same author and title, then the system provides additional fields and prompts the user for the ISBN and publishing date. 5. If the ISBN and publishing date are valid, then the system adds the book's information to the system's library and informs the user that the book was added. 6. The user is then brought to the information page for the book they have just added.
Extensions:	Extension of step 4: If the author name and book title both match an existing record in the library, the user is informed that the book is already in the system and is brought to the information page for that book.

	Extension of step 5: If the ISBN is not a 13 digit number or the publishing date is not a valid date in the past, the user is informed there is an error and is prompted to reenter the ISBN and publishing date until they are valid.
Priority:	Priority 3

Add Book: Sequence Diagram

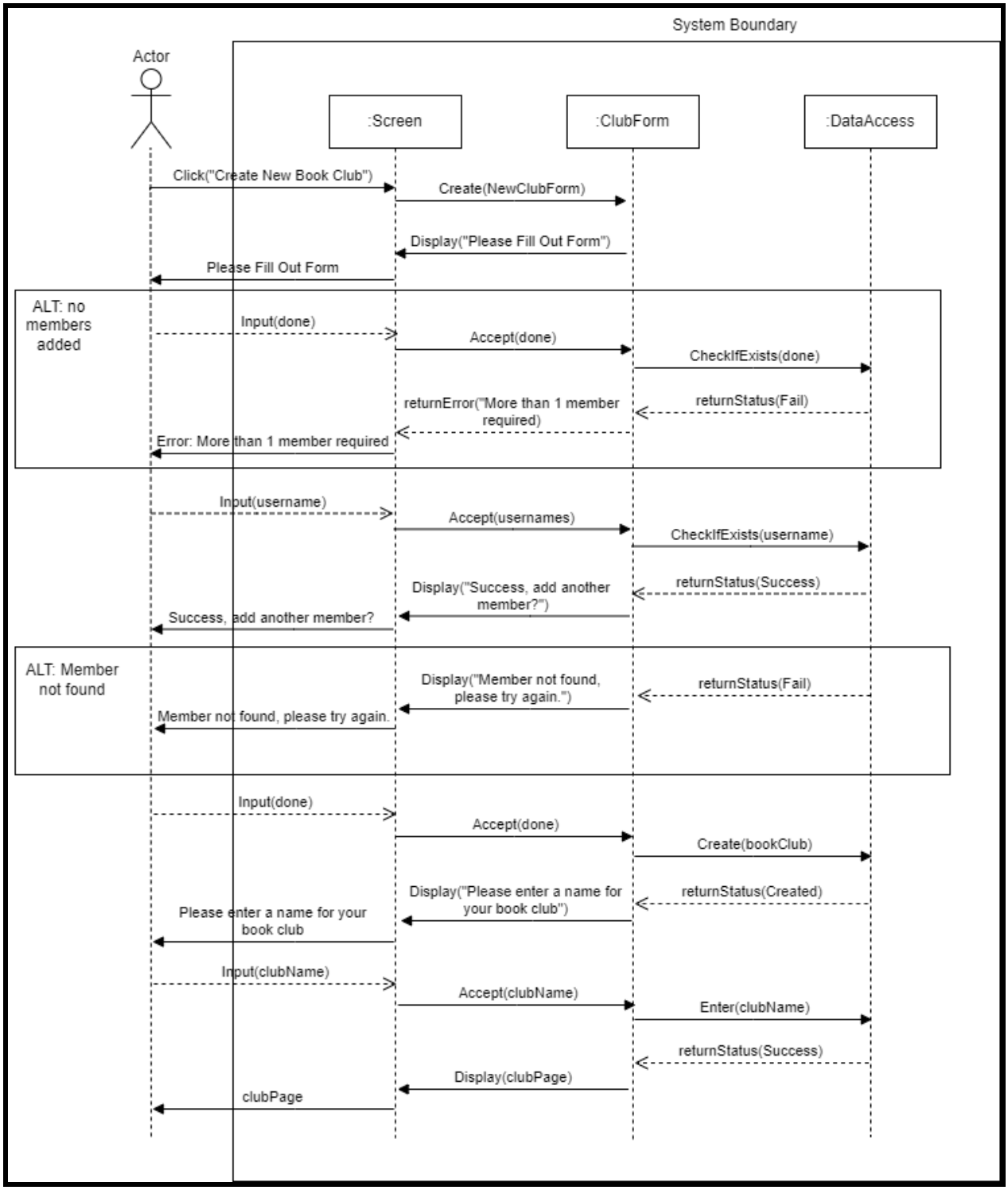


Create Book Club: Detailed Use Case

Title:	Create Book Club
Description:	The goal and context of this use case is to describe the process of a user creating a new book club group using the Booked system. A book club group is a set of users who all have access to the book club group page, and its features such as a discussion forum, bookshelf, and calendar that they can interact with to enjoy book club activities.
Actors:	System user.
Stimulus (Trigger):	The user selecting the 'Create Book Club' menu option in the book clubs page of the GUI triggers the Create Book Club use case.
Preconditions:	The user has logged into the system successfully and is at the book clubs page of the GUI.
Postconditions:	After all the events in this use case have taken place, the system has created the book club according to the user's specifications, is showing the new book club group page on the GUI, has sent invitations to the invited members, and is awaiting further input.
Main Success Scenario:	1. A user navigates to the 'Book Clubs' page in the GUI and selects the 'Create Group' menu option. 2. After this is selected, they are taken to a new page where they are prompted for the names of the users that they would like to add. 3. The user enters the names of other registered users one at a time. For each name, an icon shows up with a brief preview of that user's account, and the user who is creating the group confirms that is the intended member. 4. Once they have finished selecting users, the system displays a confirmation message to the user saying "Invitation(s) sent!". 5. The user is taken to a new page in the GUI that represents the new book club group and is prompted to enter a name for the club. 6. Once this is entered by the user, the system displays the book club group under the specified name in the GUI.

Extensions:	Extension of step 3: If the desired user is not a registered member, the system displays a “User not found” message and a prompt to return the user to the search page. Extension of step 3: If the user attempts to create a group without adding any members, the system shall present an error message notifying the user that at least 1 other user must be added to create a new book club.
Priority:	3

Create Book Club: Sequence Diagram

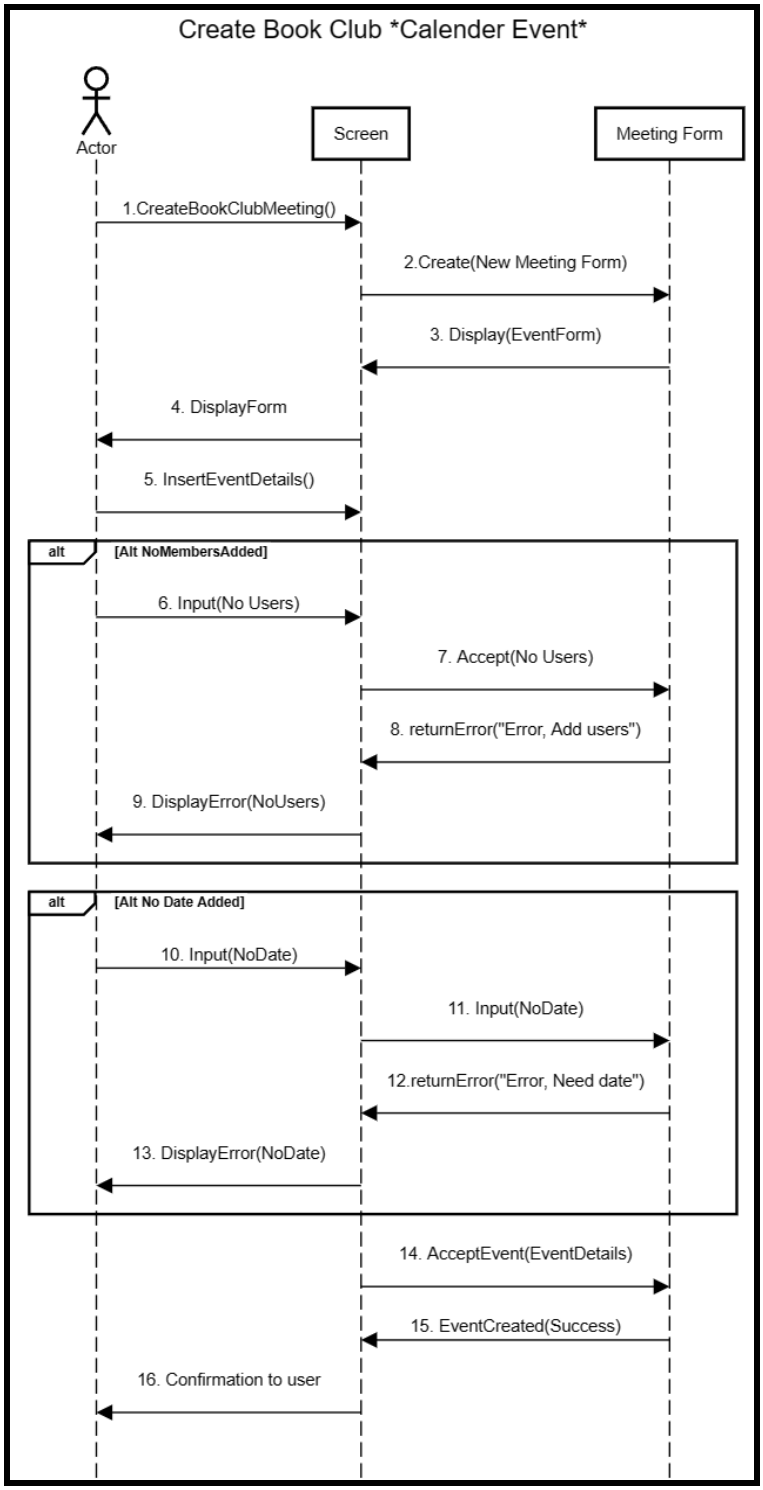


Create Calendar Event: Detailed Use Case

Title:	Create Book Club Calendar Event
Description:	The goal and context of this use case is to describe the process of a user creating a new book club group meeting on the calendar side of the application.
Actors:	System user.
Stimulus (Trigger):	The user selects the 'Create Book Club Meeting' menu option in the book clubs page of the GUI which triggers the Create Book Club Meeting use case.
Preconditions:	The user has logged into the system successfully and is at the book clubs section of the website.
Postconditions:	After all the events in this use case have taken place, the system will show the user their new book club meeting on the GUI and awaits further input. The event invite is also sent to invitees.
Main Success Scenario:	1. User initiates request to start a new event 2. A user navigates to the 'Book Clubs' page in the GUI and selects the 'Create Event' menu option. 3. After this is selected, they are taken to a new page where they are prompted for the names of the event, date and time, and invitees for the meeting. 4. The form is displayed to the user. 5. The user inputs the information in the form and clicks "Create Event" option. 6. Alt: NoMembers Added The user inputs no invitees in the event invite. 7. The event with no invitees is submitted. 8. An Error Message is sent to the GUI. 9. The Error Message is10. Alt: No Date Added The user inputs no date in the event invite 10. The event with no date is submitted 11. The error Message indicating the user needs to establish a date for the event is sent to GUI. 12. The error Message is displayed to the user.

	13. The system records the new event created. 14. The event is saved in the background database. 15. The event created success message is sent to the screen. 16. The confirmation of the event created is displayed to the user.
Extensions:	Extension of step 5: If the user attempts to create an event without an actual date the system shall present an error message notifying the user to add a date. Extension of step 5: If the user attempts to create a calendar event without adding any members, the system shall present an error message notifying the user.
Priority:	4

Create Calendar Event: Sequence Diagram

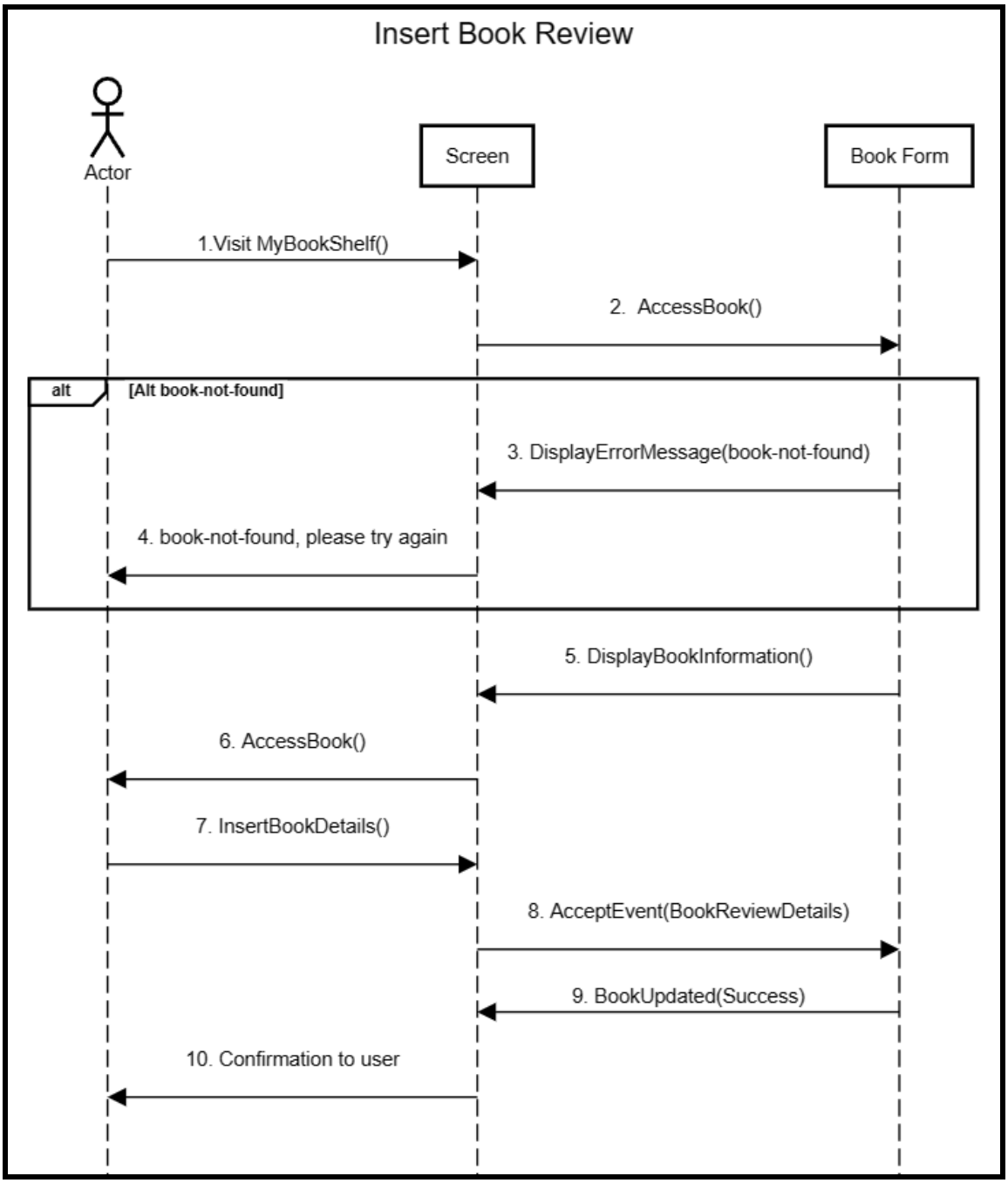


Insert Book Review: Detailed Use Case

Title:	Insert Book Review
Description:	The goal and context of this use case is to describe the process of a user accessing their virtual bookshelf and writing (or updating) a review for a particular book.
Actors:	System user.
Stimulus (Trigger):	The user selects the 'Add Book Review' option in the book clubs page of the GUI which triggers the Insert Book Review Detailed Used Case.
Preconditions:	The user has logged into the system successfully and is at his virtual bookshelf page. Additionally the user can query for a book and insert a review.
Postconditions:	After all the events in this use case have taken place, the system will update the user's book review and save it in the database.
Main Success Scenario:	1. User accesses the virtual bookshelf to add a book review for a particular book. 2. The book is searched via the GUI. 3. The error message book-not-found is sent to the GUI. 4. The error message is displayed to the user.The GUI sends information to the screen to be shown to the user. 5. The Book details are shown to the user. 6. The user has the option to add a review or modify the review that is already there. 7. The Book review details are sent to the meeting form. 8. The save event details are forwarded to the database. 9. The book updated message is returned to to the screen 10. The user is shown a confirmation message for his/her added book review.
Extensions:	Steps 3 and 4 are alternate scenarios when a book is not found in the database

Priority:	4

Insert Book Review: Sequence Diagram



System Evolution

Updated Requirements

User Requirements 1 and 2, regarding account creation, have been modified to maintain the relevance of this Software Requirements Specification while promoting the usability of the system. Instead of requiring the user to create a new username and password, Booked now grants users access to their account via a Github log in.

Use cases have also been added to allow the user to create, read, update, and delete the key system elements, including books, shelves, reviews, clubs, and meetings. These additions allow for a more comprehensive system that promotes the user's autonomy and control over each element.

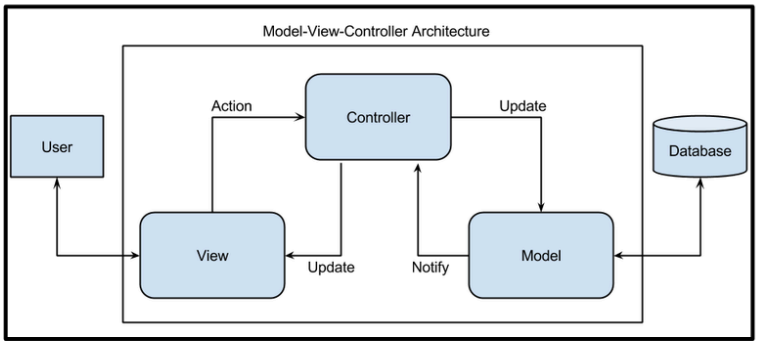
Implemented Priorities 1 & 2

The initial prioritization of the system has also evolved through the first implementation. For feasibility, this first implementation focused on priorities 1 and 2, namely the bookshelves and the user accounts.

Within the bookshelf category, some facets have been deprioritized to later iterations; for instance, UR 5 (creating custom bookshelves) has been deemed non-essential at this point in development and thus was updated to priority 4.

Priorities 3 and 4 remain to be implemented in future iterations.

Appendices



Frontend	Application Layer Technology	Data Layer Technology
• HTML • Thymeleaf	• Spring Boot • Spring Security • Github	• Apache Cassandra • Open Library

Prerequisites

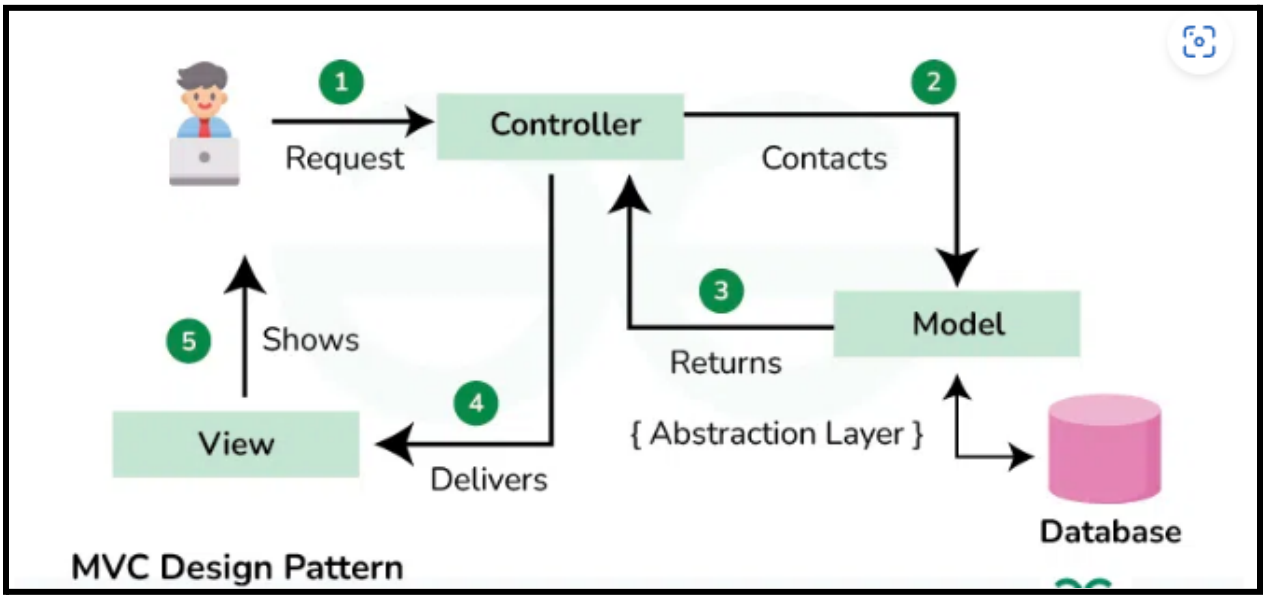
In order to run Booked users need to have:

1. JDK
2. Github account
3. Go to BookedApp.Java and hit run
4. Go to http://localhost:8080/

Backend

For the backend of the application a Model, View, Controller Implementation will be applied to the architecture. In this architecture the application is separated into three components to aid in the maintenance of the codebase. Each of the three components is responsible for a specific aspect of the application.

1. Model: Handles the data and business logic of the application.
2. View: Handles the displaying of data from the Model to the user. Receives data from the Model and sends user input to the controller.
3. Controller: Intermediary between the Model and the View. Takes the user input and updates the model so changes are reflected.



Spring MVC: Will aid in the HTML display to the users. Spring security will manage security of the page (login, password information)

Database

For Booked we are utilizing Astra DB , a cloud database built on NoSQL data engine which is known for its horsepower for our database.

OpenLibrary was used to populate the book database because of its large amount of data stored.

The database and backend will be connected through Spring Data , a common interface to connect data sources.

The data will be preloaded only one time through an API data dump.