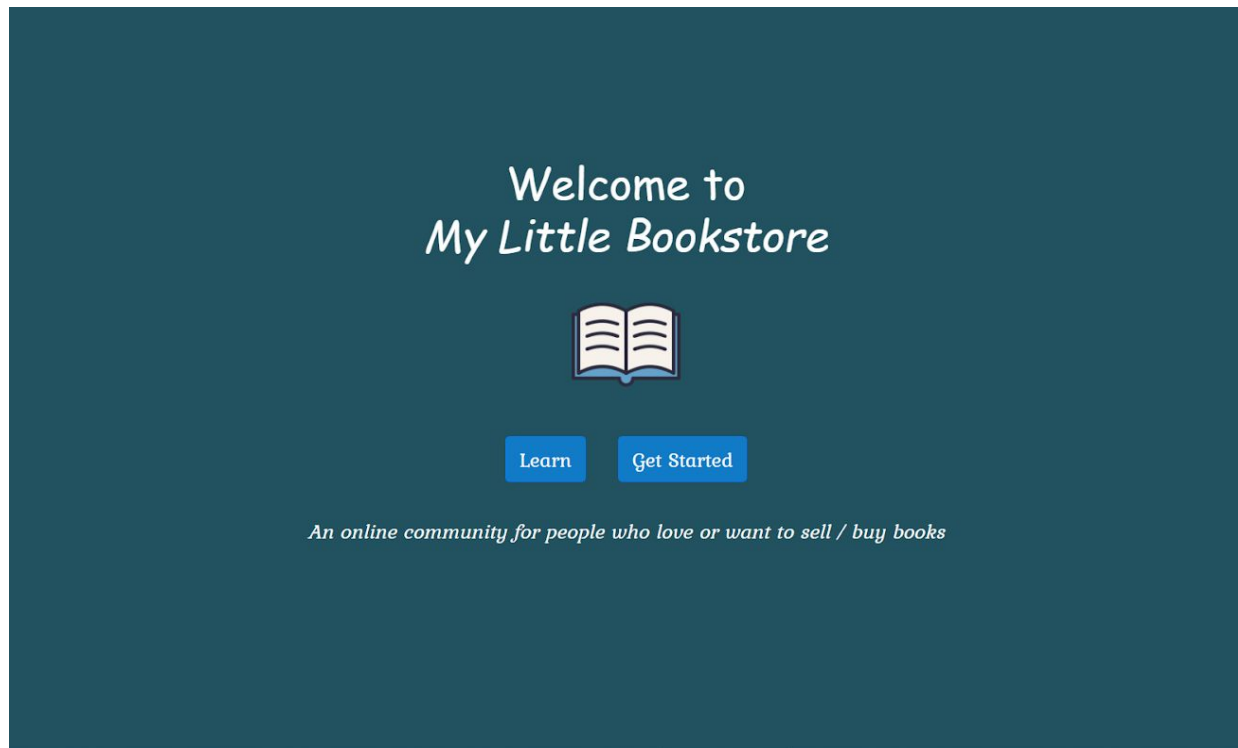


# *My Little Bookstore*

## Software Design Document



### Team Traders

Team Members: JongSun Park (Jongsun.Park@stonybrook.edu)

Daye Eun (Daye.Eun@stonybrook.edu)

Hyeon Joon Lee (Hyeonjoon.Lee@stonybrook.edu)

Donghun Kim (DongHun.Kim@stonybrook.edu)

# Table of Contents

<b>1. Introduction</b>	<b>4</b>
1.1 Product Description	4
1.2 Scope	5
1.3 Users	5
1.4 User Feedback	6
1.5 Existing alternatives	6
1.5.1 Aladin Second-Hand Bookstore	6
1.5.2 Daangn Market	6
1.5.3 Buy My Book	7
1.6 Definitions	7
1.7 References	7
<b>2. Requirements</b>	<b>7</b>
2.1 Functional Requirements	7
2.2 Use Case Diagram	9
2.3 Use Cases	10
2.3.1 Add Books Use Case	10
2.3.2 Add eBooks Use Case	10
2.3.3 Delete Books Use Case	11
2.3.4 Edit Book information Use Case	12
2.3.5 Book Request Use Case	12
2.3.6 Handle a Book Request Use Case	13
2.3.7 Browse All Books Use Case	13
2.3.8 View Recommended Books Use Case	14
2.3.9 Rate a Book Use Case	15
2.3.10 Rate a Book Use Case	15
2.3.11 Book Review Use Case	16
2.3.12 Read a Book Review Use Case	16
2.3.13 Write a Discussion about a Book Use Case	17
2.3.14 Chat Use Case	18
2.5 User Interface	18
2.4 Non-functional Requirements	18
2.4.1 Performance Requirements	18
2.4.2 Reliability	19
2.4.3 Availability	19
2.4.4 Security	19
2.4.5 Maintainability	19
2.4.6 Legal	20
2.4.7 Operating constraints	20

<b>3. System Architecture</b>	<b>20</b>
3.1 Overview	20
3.2 UML Class Diagrams	22
3.3 UML Sequence Diagrams	23
3.3.1 Add Books	23
3.3.2 Browse Books & Book Request	24
3.4 Deployment	25
3.5 Code Conventions	25
<b>4. Schedule</b>	<b>25</b>
<b>5. Contributions</b>	<b>25</b>

# 1. Introduction

## 1.1 Product Description

People like to read books for personal growth, education, and entertainment. For people who love books, it is always a joy to buy books and create a collection of their favorite books. Nevertheless, it is not easy to determine whether a book is a perfect fit unless you read it through. These books that are purchased but do not belong to are nothing but problems since they collect dust while taking space. Simply throwing them out can be an option, but it is heart-breaking since you are a book lover, and after all, they might work out for someone else. If you knew someone interested in those books, you might gladly hand them over. However, it is unlikely to find such a person among your friends.

As a solution, our team suggests “My Little Bookstore,” a web application that aims to create a community by allowing users to upload the books they would like to sell and hand them over in person. People will also be able to view information about books provided by other users to help make the right decision. Our customers include anyone who loves or cares about books and also those looking for specific second-hand books. We want to provide a variety of services to support their reading experience.

Our software will offer a bookstore function for individuals, which will let the user list all the books they would like to sell to others with the price they are willing to receive. Users can upload their books by either searching the ISBN of the book on our application, which employs Google Books API, or manually typing in the relevant information. Others will be able to view the user’s bookstore and decide whether they want the book or not, based on the price and reviews. If people decide to buy the book, they can request to buy the book and work out the details for an in-person sale with the bookstore owner.

Our software also supports a book review function, letting users post reviews about a book. This function is crucial for both users who are selling the book and buying the book. As buyers can easily gain information about the book they wish to purchase, we expect that this function will stimulate active book trading while also letting people freely share their thoughts. Reviews will be provided on the “Book Information” page, which can be accessed via clicking the title of the book.

In conclusion, we would like to create a place where book lovers can feel safe that all books find a proper owner. They will have a place not only to sell books but also to share knowledge gained and thoughts, ultimately fostering the joy of reading. We hope that “My Little Bookstore” will create a flow of books via the trade of used books and knowledge sharing so that anyone who loves books can enjoy the benefit of reading.

## 1.2 Scope

In addition to encouraging the trade of books, our web application will form a community in which people can share their book experiences. Not limited to book lovers who often read and own lots of books, people who are merely looking for second-hand books will find our application helpful.

As most people are familiar with selling platforms, our application sets a very low entry barrier. Also, configured to be used with any book genre, “My Little Bookstore” will accommodate users with diverse fields of interest and thus is considerably generalizable. Users will be able to access our applications’ functionalities, regardless of their specialties or prior knowledge. Although our application supports bookselling, it will not support an online payment system. All financial transactions will be left to the users’ hands, out of the application’s scope.

“My Little Bookstore” follows a responsive web design, which automatically adjusts the scaling of the webpage’s layouts based on a hardware device’s screen size. Thus, the application will run on any device that supports web browsing; users will be able to access every function of the application via PC, mobile phones, tablets, etc.

## 1.3 Users

Users of “My Little Bookstore” can be divided into two groups: those who love but wish to sell their books and those looking for specific second-hand books. Both groups’ age will be over nineteen, presuming that they have their own books that they wish to sell or have enough money to purchase others’. Inferring from this age, the general education level of users will be high school graduation.

Our formal users, book lovers, are interested in selling books they no longer need and enjoy discussing thoughts gained from reading books. Since they like books and frequently read them, this group’s knowledge level will likely be higher than the general population. Also, they will be familiar with all the terms related to books, such as ISBN. They also care about books’ conditions, along with detailed information about each book, such as the publisher and translator.

The other group of users is the ones looking for cheap second-hand books. They might be less familiar with terms related to books and care less about the detailed information. They tend to look for exact books or search for specific categories in their interests. They have less interest in reading reviews and might tend to look for cheaper books instead of looking at the book’s condition.

## 1.4 User Feedback

User feedback was most useful for determining the priority of functionalities. It helped determine the overall design while making sure we were developing a beneficial product. We had received some feedback by interviewing a few book lovers and are planning to conduct additional surveys and interviews after initial deployment to determine required changes in functionalities.

Some potential users have asked to support eBooks since they are getting popular these days. Therefore, our team decided to include the eBook uploading functionality, although only the copyright-free ones will be allowed due to legal issues. As such eBooks are already free, we will not allow users to charge them.

One another issue brought up by the interviewees was the need for verification when writing reviews. Since some people might write fake reviews about a book without reading it to prejudice others, there should be some means to verify whether they had actually read it. From this particular feedback, we learned that book lovers not only care about the number of reviews but also their quality and reliability. As it is infeasible to check every single review's reliability, our team prioritized those provided by users who own the book in their bookstore.

## 1.5 Existing alternatives

### 1.5.1 Aladin Second-Hand Bookstore

“Aladin” is the biggest second-hand bookstore in South Korea, which is available both online and offline. One of its most significant merits is that a seller can sell their books directly to the store and doesn't have to wait for a buyer. The disadvantages are that some books are not accepted and that the margin is relatively low. We incorporated the idea of recommending books along with the general concept of a bookstore like “Aladin” itself, as it is one of the most popular and successful bookstores.

### 1.5.2 Daangn Market

“Daangn Market” is a popular direct selling platform in South Korea. It allows users to upload items they desire to sell. Others can view such items based on their regions and contact the seller if they want to buy them. This platform's strength is that it is easy to upload and sell various products. The weaknesses of this application, for book lovers, are that it is too general for selling books and that it does not support easy posting of books. Furthermore, it does not provide sufficient information about each

book. We followed the idea of direct selling and referred to some UI design of this application while adding specialty as a bookselling platform.

### 1.5.3 Buy My Book

“Buy My Book” is an application that lets users, mostly students, buy or sell academic books, including textbooks and study materials, along with non-academic books. Students can upload their books with the prices of their choice, and other students nearby can buy them or get them for free. This application has similar goals and functionalities with “My Little Bookstore” but has a narrower range of users since it mainly aims for students. Targeting book lovers, we added review and discussion features, which are crucial for providing useful information.

## 1.6 Definitions

**ISBN** - International Standard Book Number (ISBN) is a numeric commercial book identifier intended to be unique for each book. We will use the term ISBN to refer to ISBN-13.

## 1.7 References

“Aladin.” *Aladin*, [www.aladin.co.kr](http://www.aladin.co.kr). Accessed 17 Oct. 2020.

“Buy My Book - Apps on Google Play.” *Buy My Book - Apps on Google Play*, [play.google.com/store/apps/details?id=com.TAGIdeas.BMB&hl=en](https://play.google.com/store/apps/details?id=com.TAGIdeas.BMB&hl=en). Accessed 17 Oct. 2020.

“Daangn Market.” *Daangn Market*, [www.daangn.com](http://www.daangn.com). Accessed 17 Oct. 2020.

## 2. Requirements

### 2.1 Functional Requirements

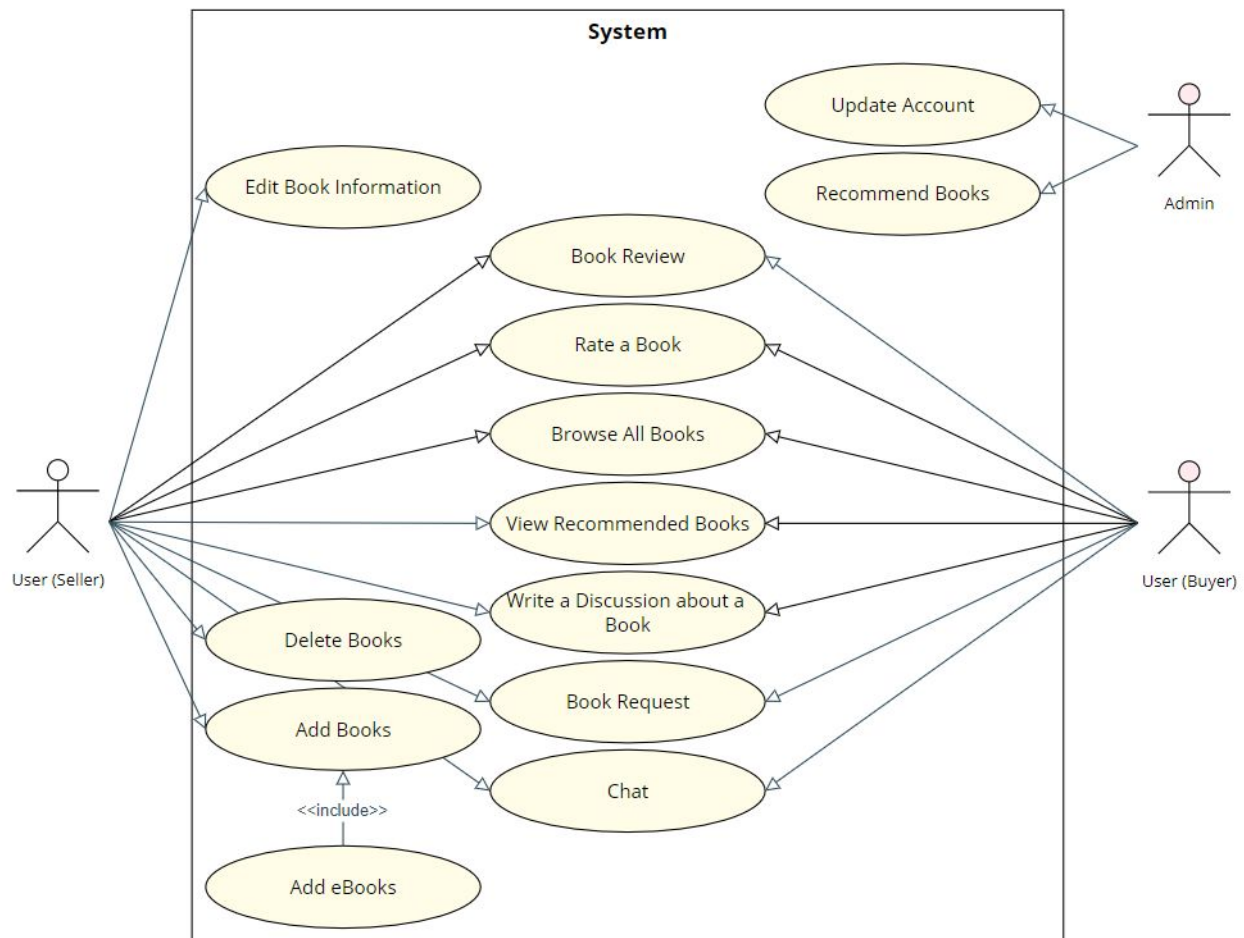
- **Sign Up (Authentication):** Users can sign up for the system by providing personal information (including user location and book genre preference) and confirming to join. An empty bookstore with the name “<User name>’s Bookstore” will be automatically created, accessible on the “My Bookstore” page. An empty bookstore will not appear when a user searches for a book.

- **Login/Logout:** Registered users can log in to or log out from their account at any time.
- **Edit Profile:** Users can edit their profile on the “Profile” page by modifying information, including their location, book genre preference, and contact information.
- **Add Books:** Sellers can upload their books by clicking the “Add Books” button on the “My Bookstore” page. Sellers can search for the ISBN on the pop up modal and retrieve corresponding book information. Alternatively, they can manually type in the relevant information. Sellers also have to provide a price for the book that they are willing to receive.
- **Add eBooks:** Sellers can also add an eBook to their bookstores by clicking the “Add Books” button and providing the digital file. Sellers can only add eBooks that are copyright-free and are not allowed to charge them.
- **Delete Books:** Sellers can delete books if they wish to remove them from their sales list. Such cases include the seller no longer willing to sell a book or had already sold the book. Sellers can achieve this by clicking the trash bin icon on the “My Bookstore” page.
- **Edit Book Information:** Sellers can edit book information by clicking the edit icon on the “My Bookstore” page.
- **Book Request:** Buyers can purchase a book from another bookstore by initially searching the book on the “Browse” page. After clicking the desired book among the resulting books, the buyer proceeds to the “Individual Book” page, in which he/she can click the “Request” button and send a request to the book owner.
- **Browse All Books:** Users can view all books uploaded by accessing the “Browse” page. Filtering those books by genre or the user’s location, as well as sorting by rating, price, or alphabetical order, is supported.
- **View Recommended Books:** Users can view recommended books on the main page. Such recommendations are decided according to the user’s preferred book genres and location, which can be initialized when signing up and edited on the “Profile” page. If users are not logged in or have not provided personal preferences, books with the highest user ratings in all regions will be shown.
- **Rate a Book:** Sellers can rate books by clicking the star icons on the “My Bookstore” page. Buyers, on the other hand, can rate books on the “Book Information” page.
- **Write Reviews about a Book:** Users can write reviews by proceeding to the “Book Information” page and clicking the “Add Review” button.
- **Read Reviews about a Book:** Users can read reviews about each book by accessing the “Book Information” page. By clicking the “Read More” button, they can read the full content of a review.



- **Write a Discussion about a Book:** Users can participate in an ongoing discussion about a book by clicking a discussion box on the “Discussions” page. Users can also start his/her own discussion by clicking the “Add Discussion” button. Users must tag the book title when creating a new discussion.
- **Chat:** Users can communicate with other users via the chatting system. Users can send and receive messages and share information with others.

## 2.2 Use Case Diagram



**Figure 1.** Use Case Diagram

## 2.3 Use Cases

### 2.3.1 Add Books Use Case

<b>Use Case:</b>	Add Books
<b>Primary Actor:</b>	Seller
<b>Priority:</b>	Essential
<b>Goal in Context:</b>	The seller wishes to upload books to his/her bookstore page.
<b>Preconditions:</b>	The seller is viewing the “My Bookstore” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The seller clicks the “Add Books” button below the list of books he/she had uploaded previously.</li> <li>2. The system presents a modal with a search box.</li> <li>3. The seller types the ISBN of the book he/she wishes to upload in the search box.</li> <li>4. The seller clicks the “Confirm” button to retrieve the corresponding data.</li> <li>5. The seller clicks the “Save” button to finalize adding a book.</li> </ol>
<b>Extensions:</b>	<p>2a. The seller clicks the “Manual Entry” tab on the top of the modal.</p> <p>2a.1 The modal displays empty input boxes instead of a search box, and prompts the seller to type in relevant information, such as title and author.</p>
<b>Postconditions:</b>	The seller is viewing the updated list of books on the “My Bookstore” page.

### 2.3.2 Add eBooks Use Case

<b>Use Case:</b>	Add eBooks
------------------	------------

<b>Primary Actor:</b>	Seller
<b>Priority:</b>	low
<b>Goal in Context:</b>	The seller wishes to upload eBooks to his/her bookstore page.
<b>Preconditions:</b>	The seller is viewing the “My Bookstore” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The seller clicks the “Add Books” button below the list of books he/she had uploaded previously.</li> <li>2. The system presents a modal with a search box.</li> <li>3. The seller clicks the “Upload eBook” tab on the top of the modal.</li> <li>4. The seller uploads a digital file and fills in the required fields.</li> <li>5. The seller clicks the “Save” button to finalize adding a book.</li> </ol>
<b>Postconditions:</b>	The seller is viewing the updated list of books on the “My Bookstore” page.

### 2.3.3 Delete Books Use Case

<b>Use Case:</b>	Delete Books
<b>Primary Actor:</b>	Seller
<b>Priority:</b>	Essential
<b>Goal in Context:</b>	The seller wishes to delete a book from his/her bookstore page.
<b>Preconditions:</b>	The seller is viewing the “My Bookstore” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The seller clicks the trash bin icon next to the book that he/she had uploaded before and wishes to delete.</li> <li>2. System asks the seller to confirm the deletion.</li> <li>3. The seller clicks the “Delete” button.</li> </ol>
<b>Postconditions:</b>	The seller is viewing the updated list of books on the “My Bookstore” page.

### 2.3.4 Edit Book information Use Case

<b>Use Case:</b>	Edit Book Information
<b>Primary Actor:</b>	Seller
<b>Priority:</b>	Essential
<b>Goal in Context:</b>	The seller wishes to change the book information such as price, title, author and rate.
<b>Preconditions:</b>	The seller views the list of books he/she uploaded.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The seller clicks the edit icon next to the book panel.</li> <li>2. The seller views a modal containing book information in the editable input boxes.</li> <li>3. The seller clicks the price, title and rate input box on the modal and changes the book information.</li> <li>4. The seller clicks the “Save” button on the bottom of the modal.</li> </ol>
<b>Postconditions:</b>	The seller views the list of books with updated information on the "My Bookstore" page.

### 2.3.5 Book Request Use Case

<b>Use Case:</b>	Book Request
<b>Primary Actor:</b>	Buyer
<b>Priority:</b>	Essential
<b>Goal in Context:</b>	The buyer wishes to purchase a book from another user.
<b>Preconditions:</b>	The buyer clicks on a specific book which he/she wants to purchase and proceeds to the corresponding “Book Information” page.

<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. Among the list of sellers under the title “Bookstore Offers,” the buyer selects a person from whom he/she will purchase the book.</li> <li>2. The buyer clicks the “Buy” button next to a seller on the list.</li> <li>3. System displays a modal saying that a request has been sent to the seller.</li> </ol>
<b>Postconditions:</b>	The buyer views the detailed information of the book and the list of sellers, along with reviews about the book.

### 2.3.6 Handle a Book Request Use Case

<b>Use Case:</b>	Handle a Book Request
<b>Primary Actor:</b>	Seller
<b>Priority:</b>	Essential
<b>Goal in Context:</b>	The seller wishes to accept or decline a book request from a buyer.
<b>Preconditions:</b>	The seller is viewing the “My Bookstore” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The seller views the list of received requests under the title “Received Requests.”</li> <li>2. The seller either clicks the “Accept” button or the “Decline” button.</li> </ol>
<b>Extensions:</b>	2a. The “Status” column, which displays “Pending” by default, either displays “Accepted” or “Declined” following the seller’s decision.
<b>Postconditions:</b>	The seller views the updated list of received requests.

### 2.3.7 Browse All Books Use Case

<b>Use Case:</b>	Browse All Books
------------------	------------------

<b>Primary Actor:</b>	User
<b>Priority:</b>	Essential
<b>Goal in Context:</b>	The user wishes to view all the books on the website.
<b>Preconditions:</b>	The user is on the “Browse” page of the website.
<b>Scenario:</b>	1. The user views the list of books that are currently uploaded by other users and sorted by rating.
<b>Extensions:</b>	1a. The user can sort the books by price and alphabetical order. 1b. The user can filter the books by genre and location.
<b>Postconditions:</b>	The user is viewing the sorted list of books.

### 2.3.8 View Recommended Books Use Case

<b>Use Case:</b>	View Recommended Books
<b>Primary Actor:</b>	User
<b>Priority:</b>	Essential
<b>Goal in Context:</b>	The user wishes to view the recommended books.
<b>Preconditions:</b>	The user is on the main page
<b>Scenario:</b>	1. The user views the list of books that are filtered based on the user’s preferred genre and location, both of which can be edited in the “Profile” page.
<b>Postconditions:</b>	The use remains on the main page.

### 2.3.9 Rate a Book Use Case

<b>Use Case:</b>	Rate a Book
<b>Primary Actor:</b>	Buyer
<b>Priority:</b>	Moderate
<b>Goal in Context:</b>	The buyer wishes to rate a specific book.
<b>Preconditions:</b>	The buyer is viewing the “Book Information” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The buyer clicks the “Rate” button next to the star icons.</li> <li>2. The system presents a modal with a selection box.</li> <li>3. The buyer chooses a rating option, ranging from one to five stars.</li> <li>4. The buyer clicks the “Confirm” button to finalize the rating.</li> </ol>
<b>Postconditions:</b>	The buyer is viewing the updated “Book Information” page.

### 2.3.10 Rate a Book Use Case

<b>Use Case:</b>	Rate a Book
<b>Primary Actor:</b>	Seller
<b>Priority:</b>	Moderate
<b>Goal in Context:</b>	The seller wishes to rate a specific book.
<b>Preconditions:</b>	The seller is viewing the “My Bookstore” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. By clicking the star icons next to each book on the list, the user chooses a rating option, ranging from one to five.</li> </ol>
<b>Postconditions:</b>	The user is viewing the updated “My Bookstore” page.

### 2.3.11 Book Review Use Case

<b>Use Case:</b>	Book Review
<b>Primary Actor:</b>	User
<b>Priority:</b>	Moderate
<b>Goal in Context:</b>	The user wishes to write a review about a specific book.
<b>Preconditions:</b>	The user is viewing the “Book Information” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user clicks the “Add Review” button below the list of existing reviews.</li> <li>2. The system presents a modal with an input text box, in which the seller can write a review.</li> <li>3. The seller clicks the “Save” button to finalize writing the review.</li> </ol>
<b>Extensions:</b>	2a. The user can edit or delete his/her past reviews at any time.
<b>Postconditions:</b>	The user is viewing the updated list of reviews on the “Book Information” page.

### 2.3.12 Read a Book Review Use Case

<b>Use Case:</b>	Read a Book Review
<b>Primary Actor:</b>	User
<b>Priority:</b>	Moderate
<b>Goal in Context:</b>	The user wishes to read a review about a specific book.
<b>Preconditions:</b>	The user is viewing the “Book Information” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user views a list of brief heads of reviews on the bottom-right side of the page, under the title “Reviews.”</li> </ol>



<b>Extensions:</b>	1a. The user clicks the “Read More” button next to the review he/she wants to read in detail.
<b>Postconditions:</b>	The user is viewing the list of reviews on the ‘Book Information’ page.

### 2.3.13 Write a Discussion about a Book Use Case

<b>Use Case:</b>	Write a Discussion about a Book
<b>Primary Actor:</b>	User
<b>Priority:</b>	Moderate
<b>Goal in Context:</b>	The user wishes to discuss a book.
<b>Preconditions:</b>	The user is viewing a list of discussions on the “Discussions” page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user chooses a specific discussion he/she feels interested in.</li> <li>2. The user clicks the comment icon to view the list of all existing comments on that particular discussion and an input text box at the bottom.</li> <li>3. The user leaves a comment or any discussion in the box and clicks the “Comment” button.</li> </ol>
<b>Extensions:</b>	<p>1a. The user can either start a new discussion by clicking the “Add Discussion” button or participate in a pre-existing discussion.</p> <p>1b. The user can filter the discussions by searching for a specific book in the search box.</p> <p>1c. The user can sort the discussions by the number of likes, the number of comments, or the date.</p> <p>3a. The user can click the “edit” or “delete” buttons to edit or delete his/her previous comments at any time.</p>
<b>Postconditions:</b>	The user is viewing an updated list of discussions on the “Discussion” page.

### 2.3.14 Chat Use Case

<b>Use Case:</b>	Chat
<b>Primary Actor:</b>	User
<b>Priority:</b>	Low
<b>Goal in Context:</b>	The user wishes to start a chat with another user.
<b>Preconditions:</b>	The user views another user's ID on the "Discussion," "My Bookstore," or "Book Information" page.
<b>Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user clicks another user's ID.</li> <li>2. The system presents a modal (chatbox) with a text input field.</li> <li>3. The user types the message he/she would like to send in the input field.</li> <li>4. The user clicks the "Send" button to send the message.</li> </ol>
<b>Postconditions:</b>	The modal closes, and the user remains on the same page.

## 2.5 User Interface

The "My Bookstore" page has been updated to support the handling of received requests on the same page, without interactions involving modals. Changes in source codes are uploaded to the Github repository below.

[https://github.com/CSE416-Traders/CSE416\\_Group\\_Project](https://github.com/CSE416-Traders/CSE416_Group_Project)

## 2.4 Non-functional Requirements

### 2.4.1 Performance Requirements

Under both normal and peak workload conditions, the application does not generate any substantial delays in users' tasks. At least 90% of tasks that involve database modification, such as creating a library or uploading a book, shall be processed in less than 3s, regardless of the hardware used;

those that generate a delay of over 10s will be automatically terminated, prompting a user to reload the page. Other tasks that do not involve database modification, such as login and browsing books or libraries, shall be processed in less than 1s. The application supports more than 1,000 terminals simultaneously.

### 2.4.2 Reliability

The web application should work reliably in terms of recovery. The application should handle an unexpected termination of a session and allow users to continue with the previous tasks without substantial data loss. Furthermore, countermeasures for primary database crash, corruption, or loss are achieved through daily database backups.

### 2.4.3 Availability

Our application is always available except for the regular maintenance period. In cases of system failures, additional maintenance periods will be introduced. Subsequently, users will receive a notification regarding the failure and its causes and will be asked to reload or re-enter the website after a certain period. The recovery time varies depending on the severity of the failure.

### 2.4.4 Security

Utilizing GraphQL as the database query language, our application may expose vulnerabilities against attacks like deeply nested queries, leading to network quality disruption. The system will incorporate appropriate protection techniques, such as depth limiting, to respond to such threats.

To prevent unauthorized access, it is also necessary to have a set of security measures, such as a complexity requirement for passwords, a restriction on the number of login trials, and automatic logout. Any password should be longer than eight characters and include both alphabets and digits. In case a user types in a false userID or password and fails to login three times consecutively, the system will forbid further login trials for thirty minutes. A user session that has been idle for ten minutes will be automatically terminated.

### 2.4.5 Maintainability

Consistency of the source codes that build up our web application is crucial for the ease of maintenance, as it ensures a uniform environment for possible administrators or developers. Modular programming approaches and appropriate uses of interfaces also allow the ease of maintenance by reducing the source codes' complexity.

### 2.4.6 Legal

All the private information users supply should be protected. When a user creates an account, his/her private information, including email address, phone number, and address, is collected. Such information is guaranteed not to be distributed to a third party for commercial benefit. Prior to signing up, users must agree to the collection and use of his/her personal information.

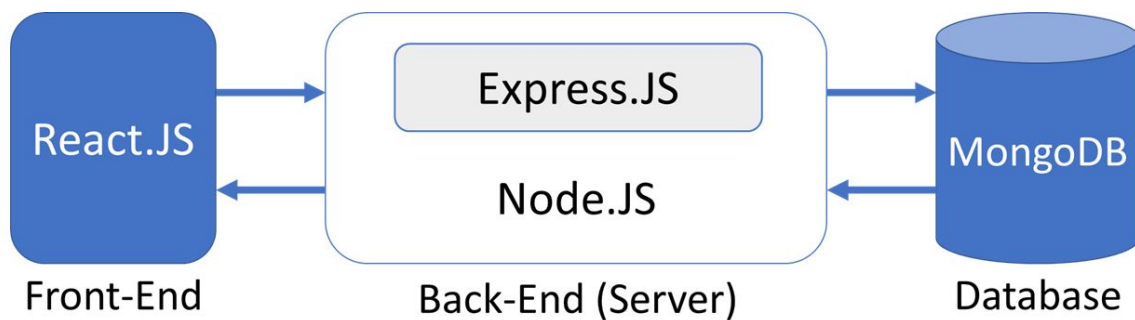
eBooks uploaded by users should be copyright-free; unauthorized ebooks will be filtered out during the upload, using the Google Books API.

### 2.4.7 Operating constraints

Since our application runs on a web platform, it requires internet connection and web browser software, including Google Chrome, Safari, Firefox, etc.

## 3. System Architecture

### 3.1 Overview



**Figure 2.** MERN Stack Overview

Our team will do MERN stack development to implement the application. MERN stands for MongoDB, Express.js, React.js, and Node.js, which allows developers to create a full-stack application. As illustrated in the above diagram, while React.js is a JavaScript framework used for front-end, Node.js serves as a web server and Express.js is a Node.js web framework. MongoDB, a NoSQL database, handles the database part. We choose to use MERN because it is quick and easy to implement a project. Also, MERN architecture can be constructed entirely using JavaScript and JSON.

For the front-end, we will use React.js v16.13.1, which is the most recent stable version available. For the server tier, we will use Node.js v14.9.0 and Express 4 which are both the most recent stable versions available. For the database, we will use MongoDB v4.0.10, which is the most recent stable version. We decided to use the most recent stable versions of each tier since they have the most features while being verified.

We will follow the Model-View-Controller design pattern, which is a widely used architectural pattern for web applications. For deployment, we decided to use Heroku since it is a cloud-based platform that supports the management of applications and it is easy to deploy a MERN application.

An additional framework we plan to use is React Bootstrap v1.3.0 since it lessens the work of designing the components and makes it look professional. We are going to use GraphQL as a query language since it allows the app to be fast and stable while retrieving relevant data. Also, Our project plans to use Google Books API to bring accurate book information and create consistency.

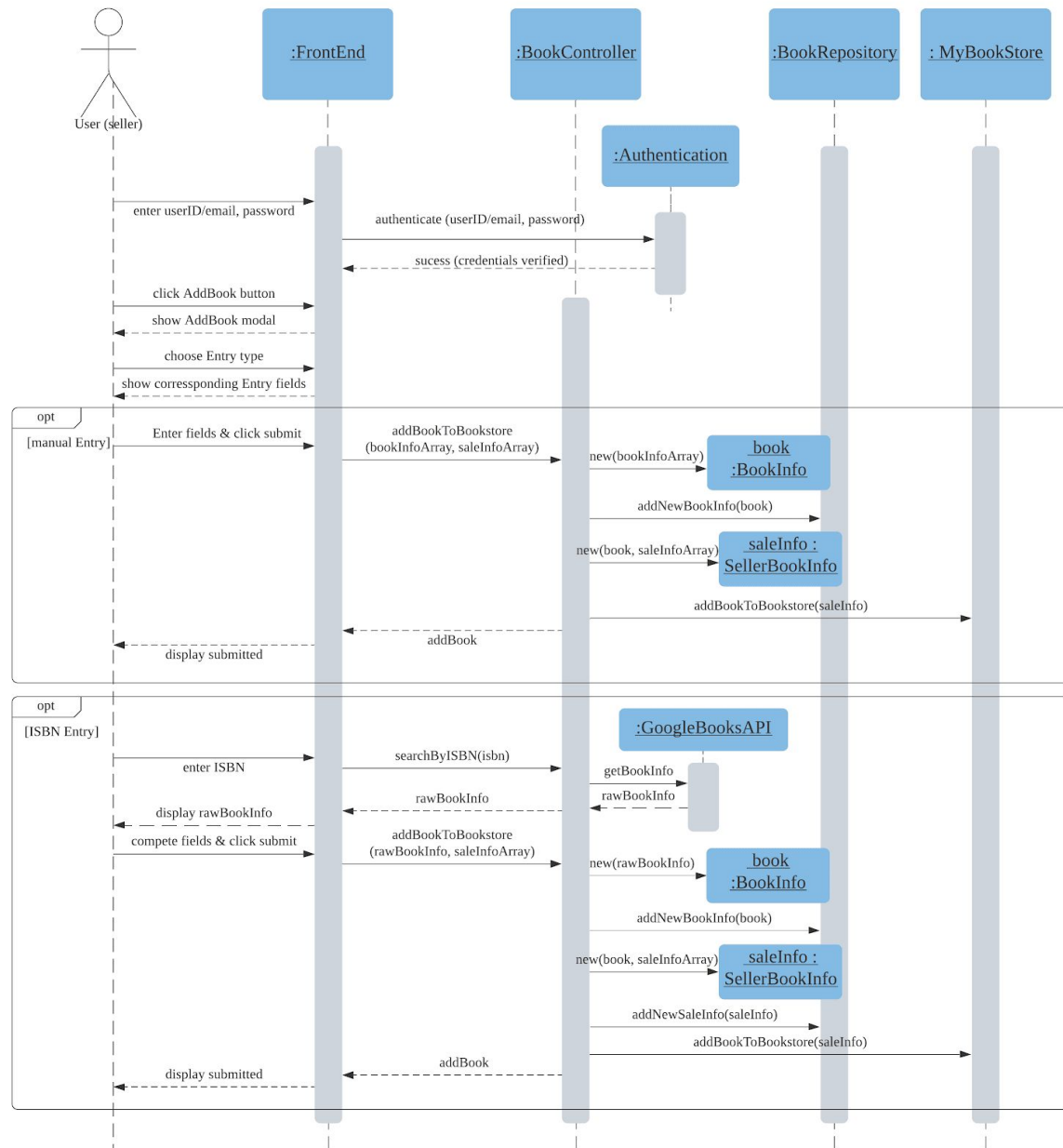


### 3.3 UML Sequence Diagrams

Link to Sequence Diagrams:

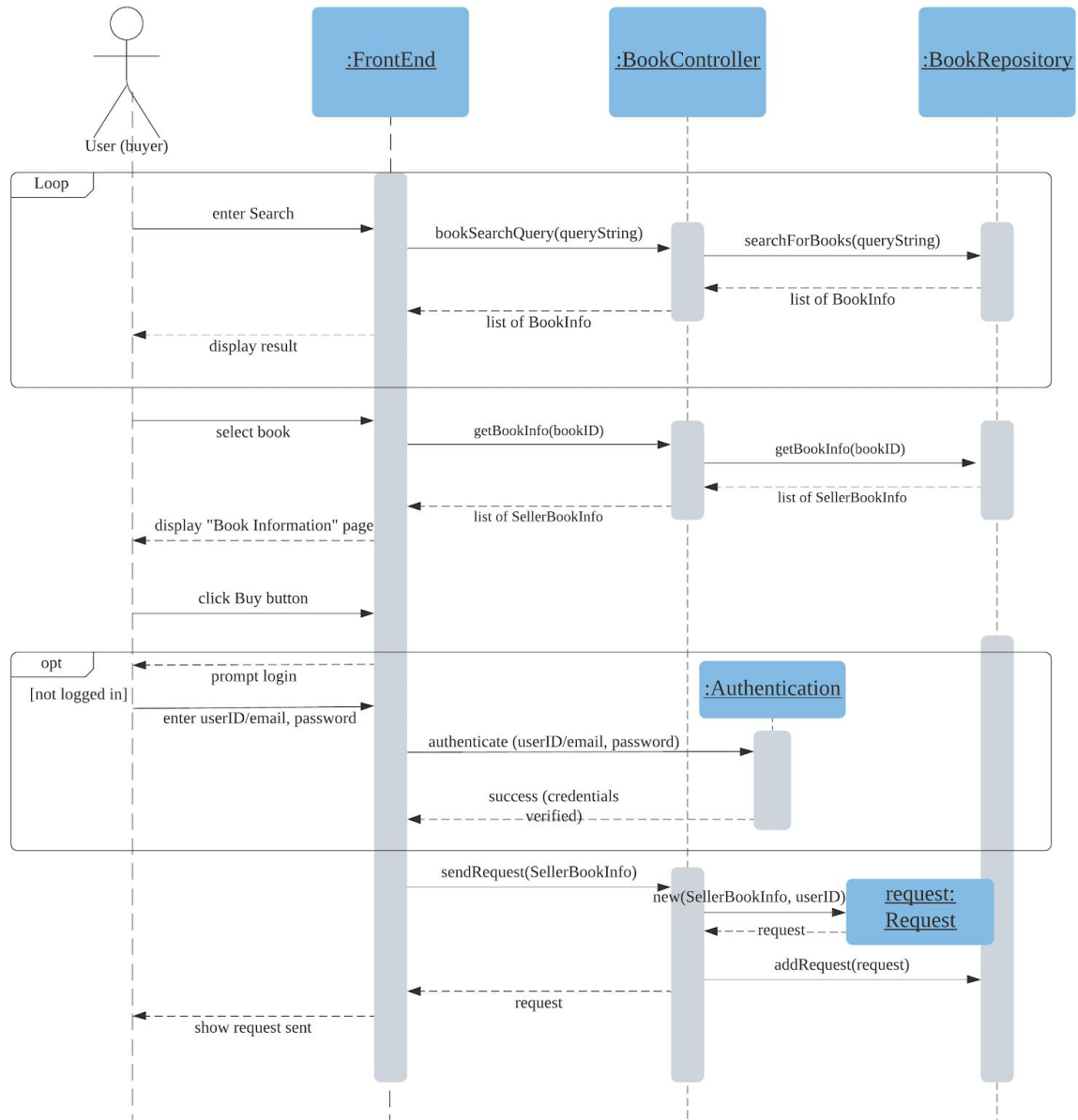
<https://app.lucidchart.com/invitations/accept/43256f9e-aa49-45b4-a7e5-6703364e6e03>

#### 3.3.1 Add Books



**Figure 4.** UML Sequence Diagram - Add Books

### 3.3.2 Browse Books & Book Request



**Figure 5.** UML Sequence Diagram - Browse Books & Book Request

### 3.4 Deployment

We decided to use Heroku for deploying our server. Heroku is a platform as a service that allows users to rent virtual computers on which to run their own computer applications. Moreover, we will use Github to push our code to Heroku.



## 3.5 Code Conventions

As we are using MERN stack, which comprises MongoDB, Express, React and Node.js, to build our web application, our application will mainly consist of JavaScript language. Consequently, we will carefully follow JavaScript code conventions to enhance our codes' maintainability, readability, and clarity. The link below describes the code convention that we will follow.

<https://google.github.io/styleguide/jsguide.html>

## 4. Schedule

We created our schedule in Trello and plan to update according to future amendments.

Trello Invitation Link

<https://trello.com/invite/b/MZ9xZ6oU/63c0d6860c46c25bdcac63a3db58d6e7/my-littleBookstore>

## 5. Contributions

**Jongsun Park:** 3.4 Deployment, 3.5 Code Convention, 4 Schedule

**Daye Eun:** 3.1 Overview

**Donghun Kim:** 3.4 Deployment, 3.5 Code Convention, 4 Schedule

**Hyeon Joon Lee:** 1 Introduction, 2 Requirements

We all equally contributed to 3.2 UML Class Diagrams, 3.3 UML Sequence Diagrams