

Design Phase/Installation and Operation Phase

Book Book (Group 12)

The new chapter for secondhand books



Presented to

Assoc. Prof. Nakornthip Prompoon

Project Members

6532125821 Peneik Sitthimongkhon

6531337321 Yuatyong Chaichana

6531335021 Puvamin Tanakittanon

6531322921 Pranesh Ingkanunt

6530363421 Wasutha Sukkrasanti

6530313021 Peemdey Chancharoen

6530186721 Tapaneeya Odmung

6530128321 Nutthapong Dissanont

6530105921 Naphon Tangtrongpairoj

2110335 Software Engineering I

1st Semester, 2024

Department of Computer Engineering Faculty of Engineering,
Chulalongkorn University

Table of Content

Introduction	1
Overview of the To-be System	1
Project Objective	3
Reference Documents	3
Objective of the Design Documents.....	4
Design Criteria & Design Activity	5
Design Criteria.....	5
Coupling	6
Consideration of Coupling's Type	7
Cohesion	9
Design Activity	10
Adding Specifications	10
Restructuring the Design	11
Optimizing the Design.....	12
Introduction of an Overview of System Design Modeling	13
System Design Constraint.....	14
Operational constraint	14
Performance constraint.....	14
Security constraint	14
Cultural and Political Constraint.....	14
Use Case Diagram	15
Class Diagram	16
Component Diagram.....	17
Method Specification	18
Edit Request Post	18
Approve Refund	19
Verifying and Validating Class and Method Design.....	21
Edit Post	21

Approve Refund	21
User Interface Design Principles	23
Layout	23
Content Awareness.....	25
Aesthetics.....	26
User Experience	27
Consistency.....	28
Minimal User Effort.....	29
Non-functional Requirements and Their Effect on System Design.....	30
Detail Real Use Case Description	33
Register as Buyer	33
Search Books	38
User Interface Design.....	41
Register as Buyer	41
Search Books	48
Non-functional Requirements and their Effect on Physical Architecture Design	51
Overview of the System Infrastructure Design	53
Network Diagram	53
Deployment Diagram	54
Hardware and Software Specification	54
Verifying and Validating the Physical Architecture Layer	56
Network diagram and deployment diagram	56
Hardware and Software Specification	59
The Impact of the Analysis and Design Solution	60
Impact on global context	60
Impact on the society context.....	60
Impact on the economic context.....	60
Impact on the environment context.....	60
Conclusion of the Design Process	61

System Installation and Operation	61
Migration Plan	61
1. Conversion Strategy	61
2. Change Management Plan	62
Post-implementation Activities.....	63
System Support.....	63
System Maintenance	63
Project Assessment	64
List of Teammate Contributions	65

List of Figures

Figure 1 Overview of the To-be System	1
Figure 2 Example of Interaction Coupling	7
Figure 3 Example of Inheritance Coupling	8
Figure 4 Example of Method Cohesion	9
Figure 5 Example of Class Cohesion.....	9
Figure 6 Example of Generalization/Specialization Cohesion	10
Figure 7 Payment Package with Added Specifications	11
Figure 8 User Package with Factoring in ECB pattern	12
Figure 9 Use Case Diagram	15
Figure 10 Class Diagram.....	16
Figure 11 Component Diagram.....	17
Figure 12 Comparing Method “editPost()” Between Class Diagram and Method Specification	21
Figure 13 Comparing Method “editPost()” Between Class Diagram and Method Specification	22
Figure 14 User Interface Layout.....	24
Figure 15 Content Awareness in User Interface	25
Figure 16 Aesthetics in User Interface	26
Figure 17 User Experience.....	27
Figure 18 Consistency in User Interface	28
Figure 19 Minimal User Effort	29
Figure 20 Register	41
Figure 21 Register - Filled-in information.....	42
Figure 22 Register - The account is already registered	42
Figure 23 Register - Fill in buyer details.....	43
Figure 24 Register - Please complete all required fields.....	43
Figure 25 Register - Verify email with OTP.....	44
Figure 26 Register - Resend OTP.....	45
Figure 27 Register - Terms and Conditions of Use	45
Figure 28 Register - Please agree to the Terms of Service	46
Figure 29 Register - Register with purchase access.....	46
Figure 30 Register - Request OTP by phone number.....	47
Figure 31 Register - The OTP is incorrect.....	47
Figure 32 Search books - Search with Special Conditions.....	48
Figure 33 Search books - Search found.....	49
Figure 34 Search books - Search not found	49

Figure 35 Search books - Sort search results	50
Figure 36 Search books - System Error	50
Figure 37 Network Diagram.....	53
Figure 38 Deployment Diagram.....	54
Figure 39 Validating Between Deployment Diagram and Network Diagram.....	57
Figure 40 Validating Between Deployment Diagram and Network Diagram.....	58
Figure 41 Validating Between Software Hardware Specification and Deployment Diagram	59

List of Tables

Table 1 Coupling Level Between Packages	6
Table 2 Non-functional Requirements and Effect on User Interface Design	32
<i>Table 3 Non-functional Requirements and Effect on Physical Architecture Design ...</i>	<i>52</i>
Table 4 Hardware and Software Specifications.....	55

Project Name: Book Book

The new chapter for secondhand books.

Introduction

The two most popular channels for trading secondhand books in Thailand are Facebook Group Platform and Shopee e-Commerce. Currently, both present significant challenges. Facebook Group Platform offers flexibility allowing sellers and buyers can interact freely without intermediaries, but they lack security and structure.

Shopee e-Commerce provides an intermediary system for safer transactions but is not specifically designed for secondhand book trading which limits its flexibility and user experience. Our goal is to create a dedicated platform that combines the strengths of both systems. By incorporating the security of an intermediary system with features specifically designed for secondhand books we aim to reduce trading risks, improve customer satisfaction and establish sustainable revenue streams for long-term success.

Overview of the To-be System

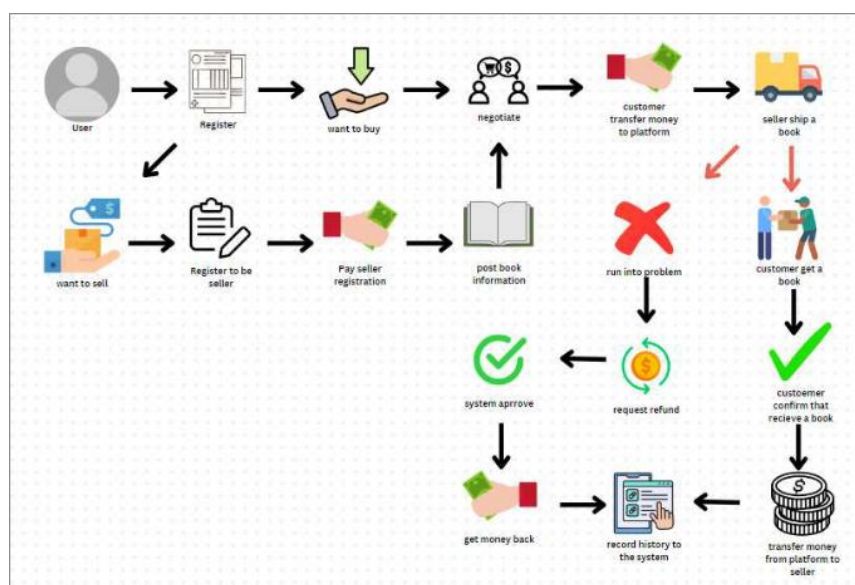


Figure 1 Overview of the To-be System

1. When opening the platform, the users will have to log in to the system with their username and password, otherwise they have to sign up with some basic information.

1.1 In case that user wants to sign up as a seller, they need to go through an extra step by registering their mobile number through OTP and sending ID card pictures to the system then the admin will approve their seller account to join.

2. There are multiple actions for the users to do in the system.

2.1 users can buy secondhand books through searching or browsing in recommendation feeds. Users can select tags to sort the type of the book, such as genre, number of pages, book condition, or special feature.

2.1.1 In case that there is no current selling book that the user is interested they can make a request post with the book detail that use the same description system as in selling post(2.2)

2.2 Users that register as sellers can sell their book. by creating a post to sell the book. The post shall contain the book name, author name, and book condition. Users can add a special tag to identify the type of book to make their book easy to find. If that book is already in the system database. The system will add further info about the book and show the first hand book price to be a tool for second hand price estimation with current book condition. Otherwise, the system will request more information, and system administrators will check the validity of the book and add it to the database.

3. If the desire to sell and buy is matching. Users can chat with each other through the chat box. Users can ask for further information, negotiate the price, or talk about shipping methods. The system can recommend shipping methods with different prices but can top up with extra packaging prices from the seller. The price for the book including the shipment is up to both parties dealing.

4. After negotiation customers will have to transfer money to the system and the book the owner will deliver the book. If delivery is through the standardized method that can be accessed by the platform, The application will provide customers with tracking service. If not, sellers need to provide the buyer with the shipment tracking themselves.

5. After book owners ship the book there will be 3 possible cases

5.1. The customer receives the book. Ensure that the book's condition on the website is correct. Customers will have to confirm in the system, then the payment will be forward to seller accounts and record that transaction as completed.

5.2. The customer receives the book, but the books have a problem as stated in policy, for instance, the book's condition is not in the same state as declared by the seller. Customers will have to report the problem within the time limit after receiving the book to the system. Afterward the platform will approve a return request sending the book back to the owner within five calendar days. And the payment would be refunded back to the customer accounts. If the inspection said that it is a false seller. Seller will receive a penalty.

5.3. If the customer did not receive the book. system will transfer money back to the customer. And the investigation will be launch to determine the cause of the undelivered book and may result in refund and punishment to the seller if not provided with reasonable cause.

Project Objective

The objective is to develop web-based applications for matching people who want to buy and sell the same book. This application provides a system where users can either choose books that they have and create a post to sell them, or they can search and find secondhand books by name or other specification, For instance, author signature, book version, or special book cover. To look for the price and book condition and negotiate with the seller for reasonable price and inquire more information about the book. This application could also prevent buyers from buying overpriced secondhand books and helping sellers sell their books at a reasonable price.

Reference Documents

1. Project Milestone 2: Book Book's Project Proposal
To serve as a guide for 3.
2. Project Milestone 3: Book Book's System/Software requirements specifications
To serve as a guide for this document as some of its information can be applied.
3. Recommended Milestone 4 Template: Final Report
To serve as a reference and example for this document

Objective of the Design Documents

The document's goals include outlining the to-be system design in a web application to ensure alignment with user expectations. Additionally, it aims to offer a comprehensive understanding of different aspects of the software system, minimizing confusion in subsequent software development stages. Furthermore, the document seeks to enhance collaboration, foster a more efficient design process, and aid in the early identification of potential issues. The objectives can be listed as follows:

1. To outline the criteria for design, such as Coupling and Cohesion, and describe the design activities within the project.
2. To identify any design constraints for the system.
3. To present Use Case and Class diagrams, incorporating Package diagrams.
4. To create a component diagram for the entire system.
5. To define method specifications
6. To address user interface design principles.
7. To provide detailed descriptions of real-use cases, including input validation and relevant messages.
8. To design the user interface, refer from Detail Real Use Case Description, Non-Functional Requirement and Functional requirement and follow the design principle.
9. To explore nonfunctional requirements and their impact on the physical architecture design.
10. To present an overview of the system infrastructure design, including network and deployment diagrams
11. To specify hardware and software requirements.
12. To verify and validate the physical architecture layer.
13. To analyze and discuss the societal, environmental, economic, and global impacts of the proposed system.

Design Criteria & Design Activity

Design Criteria will aid in assessing the design which includes Coupling, and Cohesion which make it easier to maintain while Design Activity will help in expanding the description of partitions, layers, and classes. Our class diagram is designed by using Entity-Control-Boundary Pattern. We grouped classes from our class diagram into 9 packages as below.

1. Review
2. Book
3. Report
4. Admin
5. User
6. Post
7. Chatting
8. Payment
9. Refund

Design Criteria

The system should have low coupling as each package should be independent as much as possible so that changing in one module doesn't heavily impact other modules. Likewise, the system should also have high cohesion to make the module do a single thing.

Coupling

Coupling refers to how interdependent the modules or packages are, while close coupling means that changes in one part of the design may require changes in another part. The coupling level table between our packages is shown below.

	User	Chatting	Post	Book	Review	Payment	Refund	Report
User	N/A	1	1	1	1	1	1	1
Chatting		N/A	0	0	0	0	0	0
Post			N/A	1	0	0	0	0
Book				N/A	1	1	0	0
Review					N/A	0	0	0
Payment						N/A	1	0
Refund							N/A	0
Report								N/A

Table 1 Coupling Level Between Packages

Each level signifies the following:

- **Level 0 (No Direct Coupling):** The methods do not relate to one another.
- **Level 1 (Data):** The calling method passes a variable to the called method. If the variable is a composite variable, it will be used fully in the called method.
- **Level 2 (Stamp):** The calling method passes a composite variable to the called method, but the called method uses only a portion of the passing variables.
- **Level 3 (Control):** The calling method passes control variables whose value will control the execution of the called method.
- **Level 4 (Common or Global):** The method refers to the global data area that is outside the individual objects.
- **Level 5 (Content or Pathological):** The method accesses the private variable or method of another class, which violates encapsulation.

Consideration of Coupling's Type

1. Interaction Coupling: Interaction coupling deals with coupling among methods and objects through message passing.

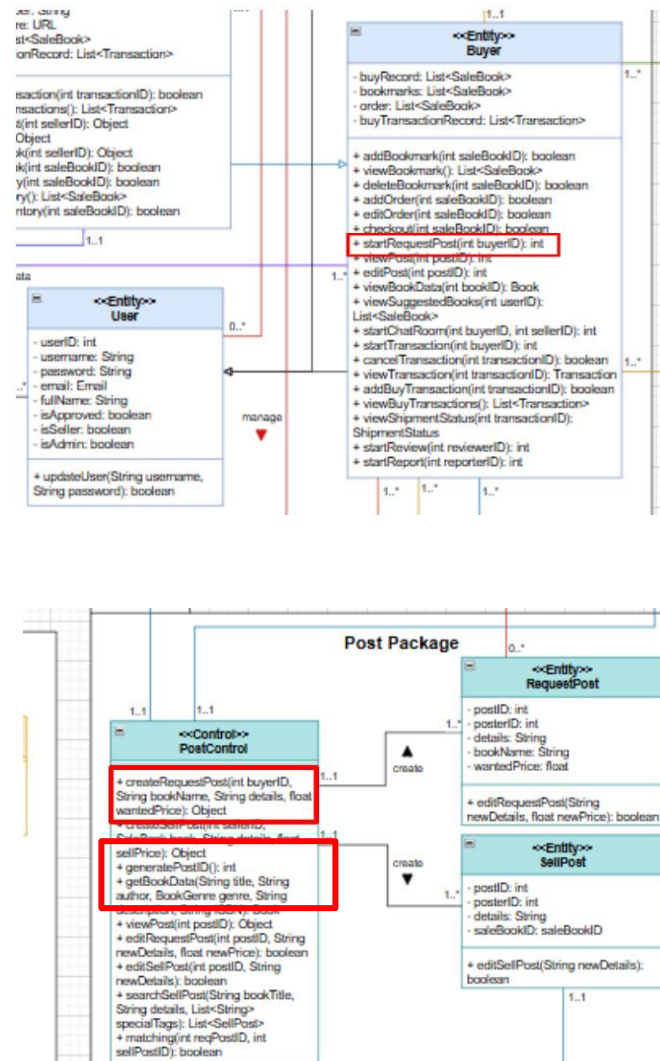


Figure 2 Example of Interaction Coupling

From the figure above, the method “startRequestPost()” in the user package will call “createRequestPost()” in the post package and pass the variable “userID”. The variables would be passed and fully used in the method, making the coupling level between user package and post package is 1.

2. Inheritance Coupling: Inheritance coupling deals with coupling among classes in an inheritance hierarchy.

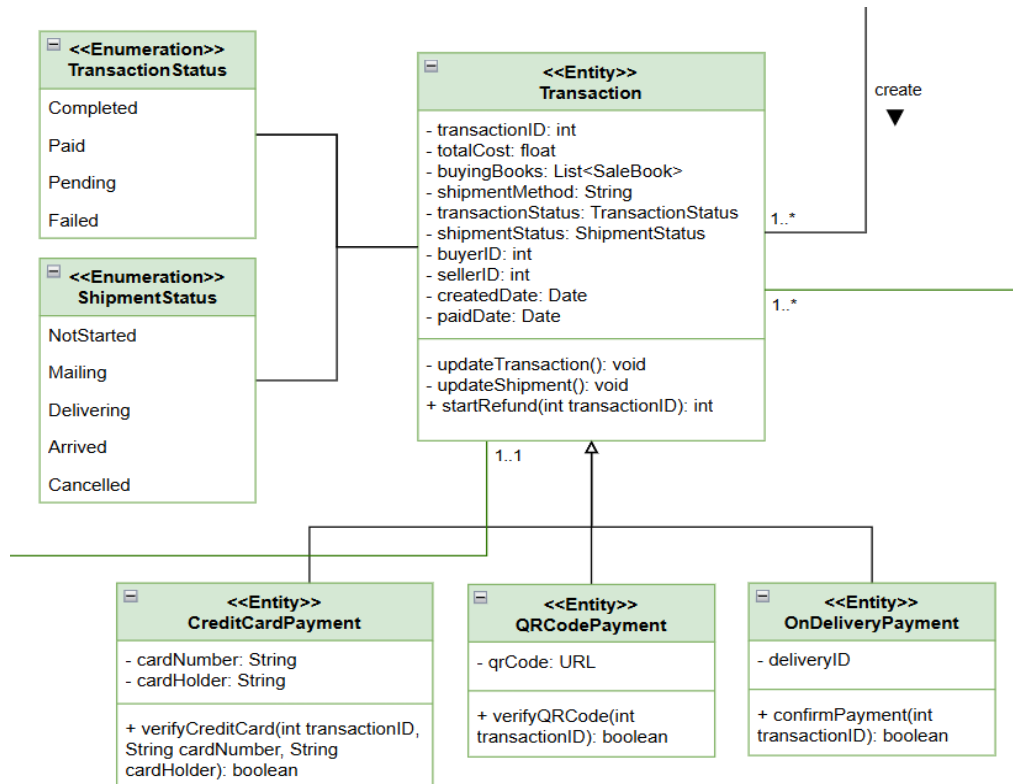


Figure 3 Example of Inheritance Coupling

From the figure above, class “CreditCardPayment” has “verifyCreditCard()” which has parameter “transactionID” inherited from class “Transaction”. In addition, class “QRCodePayment” and “OnDeliveryPayment” have methods “verifyQRCode()” and “confirmPayment()” respectively, which also have parameter “transactionID” inherited from class “Transaction”. All three subclasses have no method that calls superclass’ method, making there not much to be concerned about.

Cohesion

1. Method Cohesion: Each method of the class in our class diagram does only one thing.

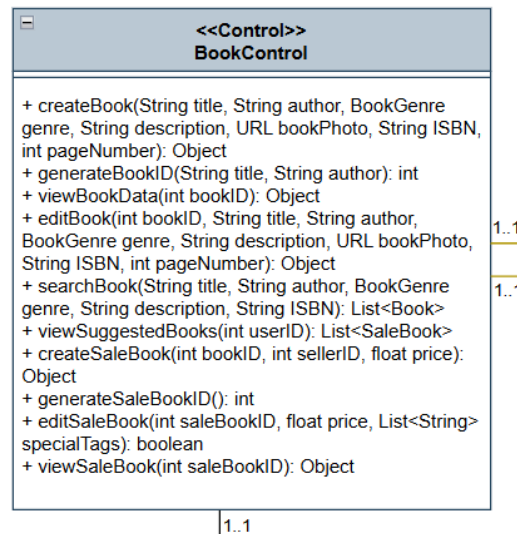


Figure 4 Example of Method Cohesion

From the figure above, the method named “editBook()” is the only method that can change the attributes of class “Book”, yielding high cohesion and the module also does a single thing.

2. Class Cohesion: Each class in our class diagram represents only one thing.

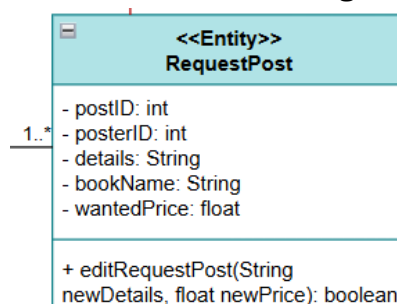


Figure 5 Example of Class Cohesion

From the figure above, the class named “RequestPost” is the only class that represents requesting posts made by buyers, yielding high cohesion and the module also represents a single object.

3. Generalization/Specialization Cohesion: Classes in a hierarchy are shown as a-kind-of relationship. While association and aggregation are not sensible to represent as a class hierarchy.

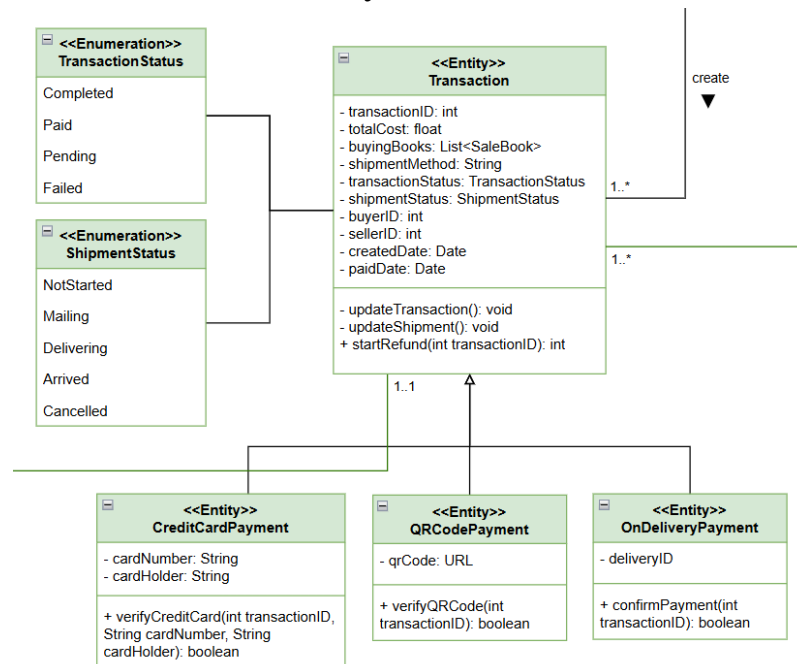


Figure 6 Example of Generalization / Specialization Cohesion

From the figure above, the class “Transaction” has subclasses “CreditCardPayment”, “QRCodePayment” and “OnDeliveryPayment” where their classes’ name makes sense for the type of message in real world, yielding high cohesion.

Design Activity

Design Activities are an extension of analysis and evolution activities to develop and improve class diagrams.

Adding Specifications

First, our team reviews the current set of analysis models. We found that the methods, attributes, and relationships of every class are both sufficient and necessary when compared with the requirements. Second, we examine the visibility of each class in the class diagram. We ensure that methods or attributes not intended for other class access are set to private, and we set them to public if access by other classes is required. Next, we added the method signatures (argument to pass to this method and type of return value(s) of each method) to every class.

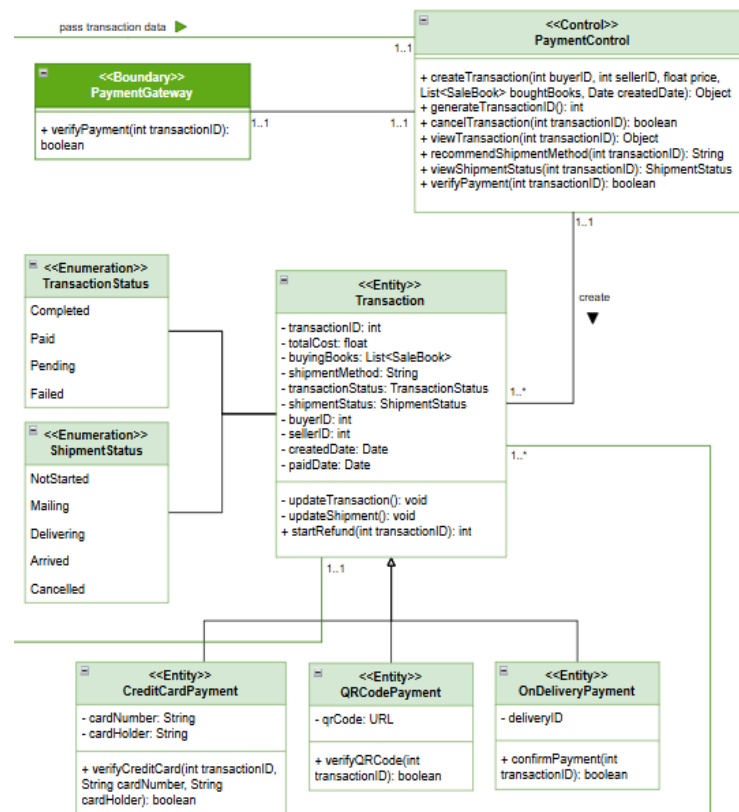


Figure 7 Payment Package with Added Specifications

From the figure above, every class in payment package contributes to the requirements of the system. For example, all 3 subclasses of Transaction class refer to 3 methods of payment that the system shall support.

Then, every variable and method that should not be accessed by other classes are set to private, such as “updateTransaction()” and “updateShipment()”.

Lastly, every method is assigned its signatures or arguments if needed. For example, “startRefund()” requires “transactionID” to create a refund.

Restructuring the Design

1. **Factoring:** Factoring means separating aspects from a class to simplify the design. Our team uses the Entity-Control-Boundary (ECB) design pattern, which already factorizes and organizes each class according to its responsibilities. Then, our team re-examines each class to simplify all classes. As a result, each class is now simplified.
2. **Normalization:** Normalization converts association classes to normal classes, and associations and aggregations to attributes. Since our design does not have special associations that need to be considered, the design is considered normalized.

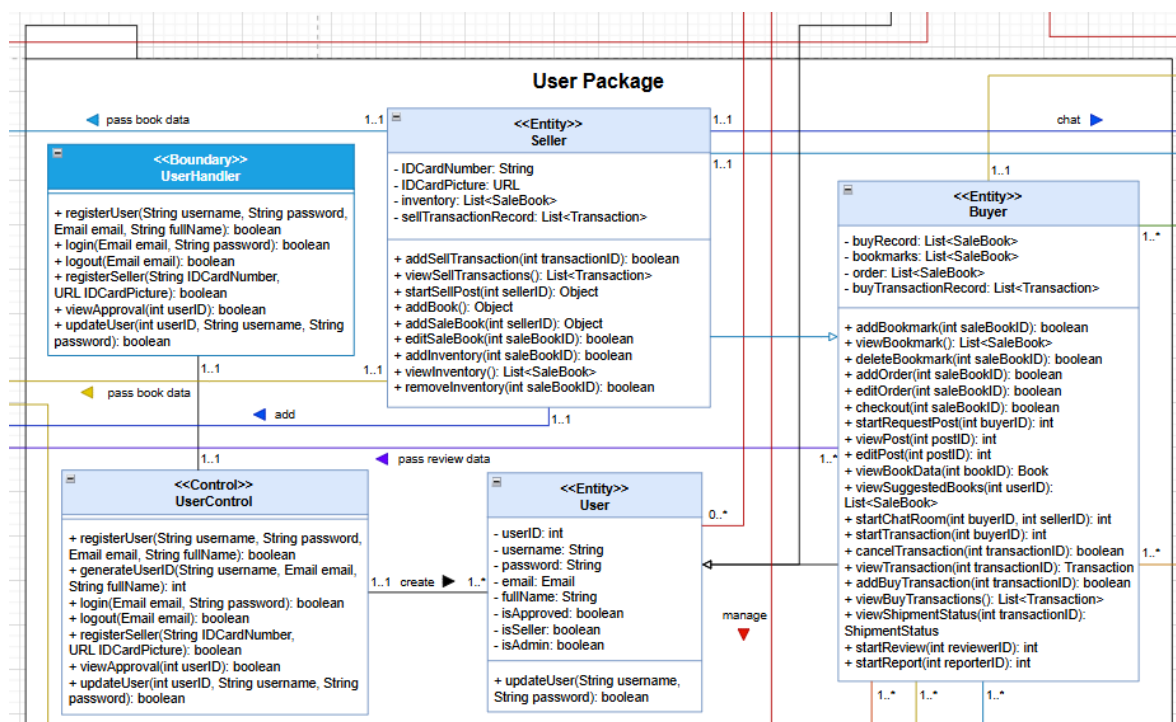


Figure 8 User Package with Factoring in ECB pattern

Optimizing the Design

The last design activity is optimizing the design in order to balance understandability with efficiency. There are 6 steps to achieve this goal.

1. Review access paths between objects: If the path is long and the message is sent frequently, it is possible to add an attribute to the calling object to store the direct connection. In the class diagram, we identify paths that are excessively long, especially for frequently sent messages. Our goal is to ensure no path exceeds 3-4 steps, which is considered optimal.
2. Review each attribute of each class: If the only methods using an attribute are read and update methods and these are exclusively used by a single class, we will move the attribute to the calling class. According to our class diagram, every attribute is updated or read in its own class.
3. Review direct and indirect fan-out of each method: Because our team already does factoring activity, the number of fan-out is appropriate.
4. Consider execution order statements in often-used methods: This step will help in arranging methods for efficiency. The class diagram of our team already has optimal execution. Upon review there are no specific methods that were identified that required reordering or optimization in terms of execution order.

5. Avoid re-computation: This step is about creating derived attributes and triggers. From the team's examination, there aren't any attributes that needed to adjust as derived attributes to avoid re-computation.
6. Consider combining classes that form a one-to-one association: there are several one-to-one relationships, but it appears that combining these classes is not practical due to their distinct responsibilities and the nature of the ECB (Entity-Control-Boundary) pattern.

For example, from figure 8 above, when "registerSeller()" is called from UserHandler class, the data is passed to UserControl class, and then User class which creates a new Seller entity. Therefore, the length of data path for registering new seller is 3 steps, which is still considered optimal.

Introduction of an Overview of System Design Modeling

We decided to use OOSAD methodology to design the To-be system in the document. About the agreement of software and system design, the system should be high cohesion and low coupling, because we want the system to be easy to maintain and improve overtime according to our selected SDLC methodology. The system is divided into objects which show the relationship between the objects to improve team communication. And to show the level of abstraction in software development, we use UML diagrams with model format, for instance, use-case diagram, deployment diagram and component diagram. The benefits of UML are

1. It visualizes the model of the system and organizes the model efficiently.
2. It enhances collaboration and provides a more efficient design process which also helps identify potential problems early in the design process.
3. It helps developers save time by automating some of the design processes.
4. It helps developers avoid the potential errors in the program

System Design Constraint

Operational constraint

1. The system operates on both desktop and mobile platforms.
2. It is compatible with Windows, MacOS, Android, and iOS.
3. The system connects to an ISBN database.
4. It records users' previous transactions.
5. Real-time consistency is maintained across all components, including transactions, messaging, and updates.

Performance constraint

1. The system retrieves the data of any book in 2 seconds or less.
2. Adding new data to any book is completed in 2 seconds or less.

Security constraint

1. The system authenticates users using a username-password mechanism.
2. Each user's transactions are secured privately.
3. Users can request permission from admins to modify book data.
4. Access to admin data is restricted from general users.

Cultural and Political Constraint

1. The system operates in Thai.
2. Transactions use Thai Baht as the currency.
3. The system complies with Thailand's Computer-related Crime Act 2017 and Personal Data Protection Act 2019.

Use Case Diagram

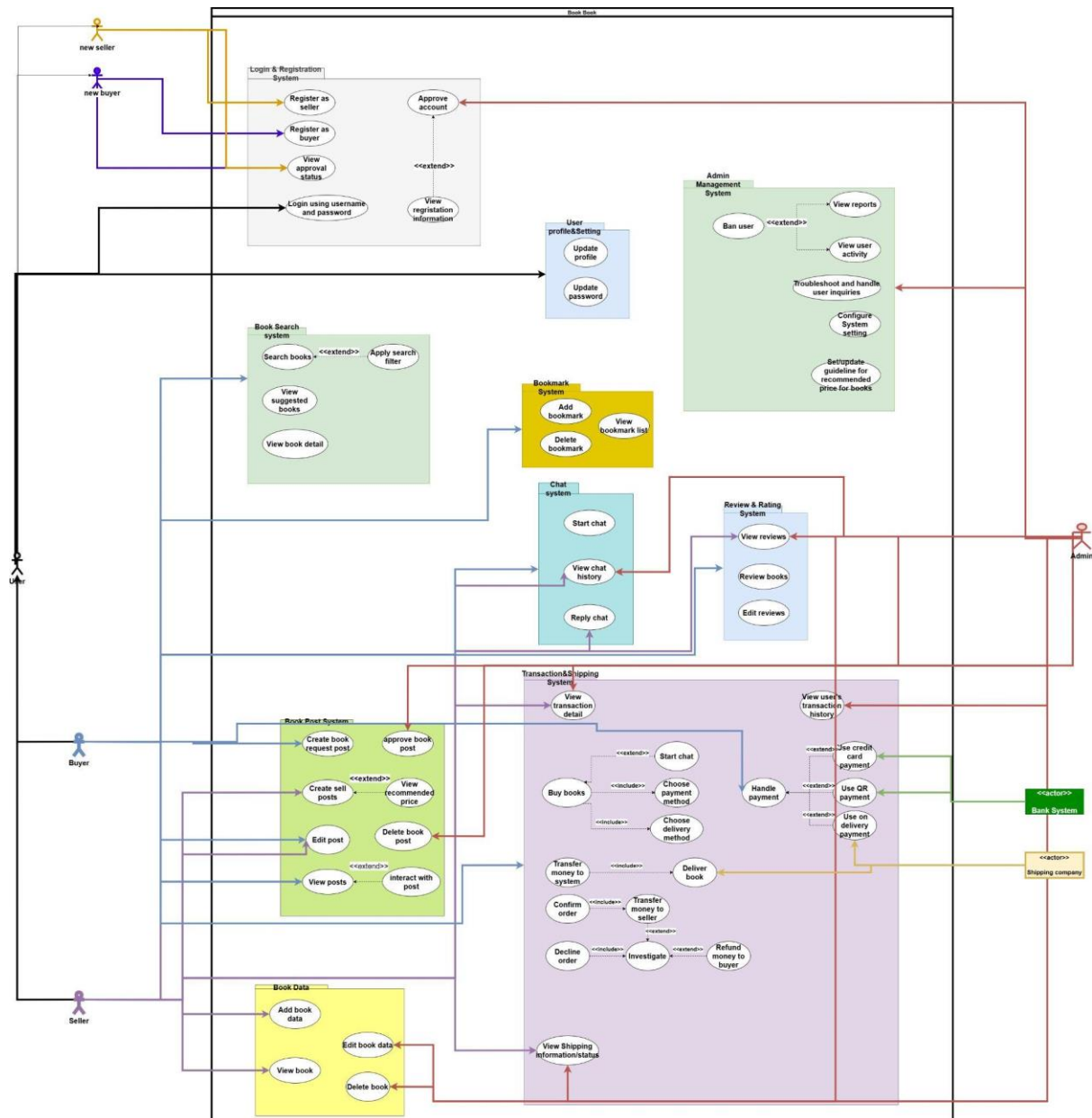


Figure 9 Use Case Diagram

https://drive.google.com/file/d/1pA6-jOpbwy_rX033AppkjJVfIyqpjwoC/view?usp=sharing

Class Diagram

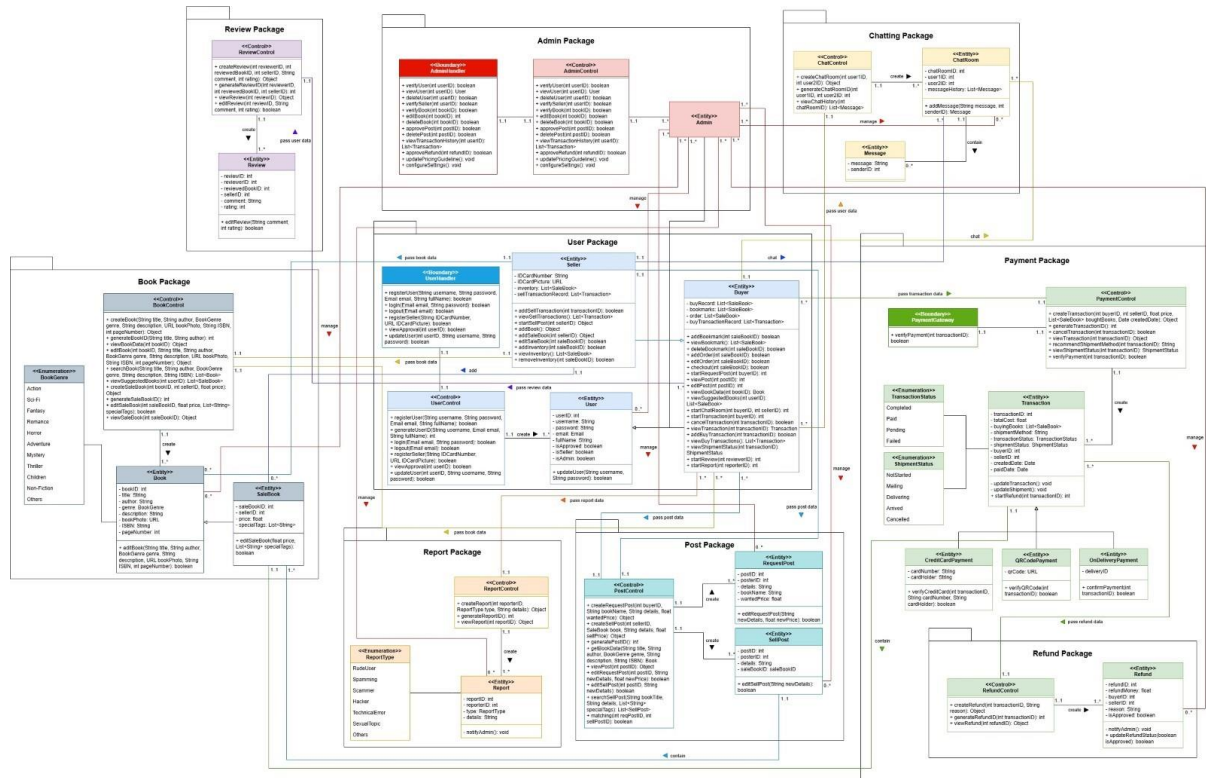


Figure 10 Class Diagram

<https://drive.google.com/file/d/1CNM0uZIRaw2KDN6OMatrW0eBXSv-hRO/view?usp=sharing>

Component Diagram

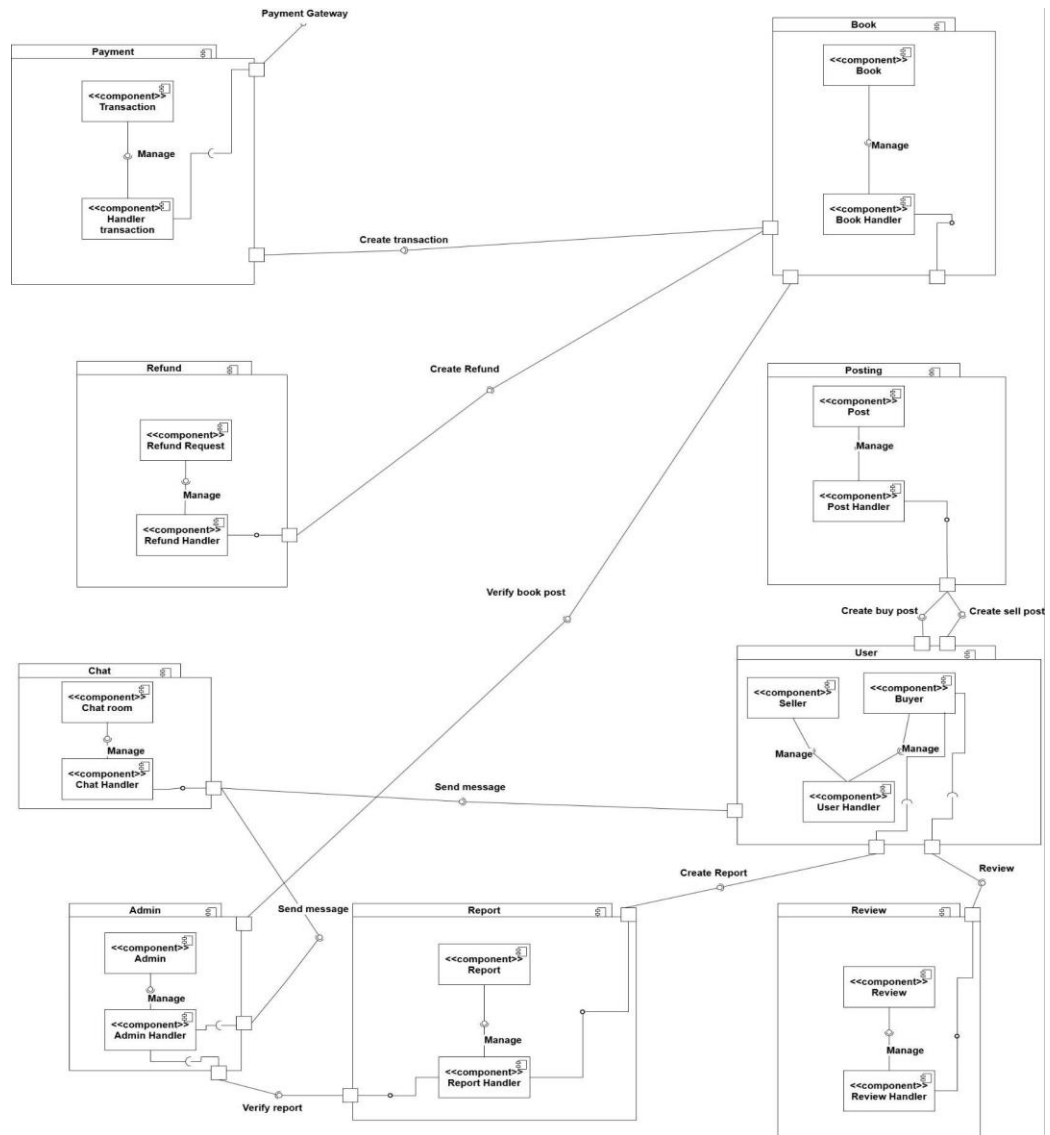


Figure 11 Component Diagram

<https://drive.google.com/file/d/11HsyxHPqyzLtBGySwfQojlA0AqHKslre/view?usp=sharing>

Method Specification

Edit Request Post

MethodName: editRequestPost	Class Name: PostControl	ID: 01
Contract ID: 01	Programmer: TOM	Date Due: 1/12/25
Triggers/Events: User wants to edit a post		
Arguments Received: Data Type:	Note:	
int: postID	postID of the post	
String: newDetails	details of the post	
float: newPrice	wanted price of the book	
Messages Sent & Arguments Passed & Return Data Type: ClassName.MethodName:	Argument : Data Type:	Return Data Type:
PostControl.editRequestPost()	int: postID String: newDetails Float: newPrice	boolean
RequestPost.editRequestPost()	String: newDetails Float: newPrice	boolean
Arguments Returned: Data Type:	Note:	
boolean	Truth value that shows if the post is found and edited	

<p>Algorithm Specification:</p> <p>IF postID is found</p> <p> View post class</p> <p> IF Post is RequestPost</p> <p> Change post detail to newDetail</p> <p> Change post wantedPrice to newPrice</p> <p> Return True</p> <p> ELSE Return False</p> <p>ELSE Return False</p>
Misc. Notes: None

Approve Refund

Method Name: approveRefund	Class Name: AdminControl	ID: 02
Contract ID: 02	Programmer: PSK	Date Due: 1/12/25
Triggers/Events: Admin wants to approve or disapprove refund		
Arguments Received:		Note:
Data Type:		
int: refundID		refundID of the refund
Messages Sent & Arguments Passed & Return Data Type:	Argument : Data Type:	Return Data Type:
ClassName.MethodName:		
RefundControl.viewRefund()	int: refundID	Object
Refund.updateRefundStatus()	boolean: isApproved	boolean

Arguments Returned: Data Type:	Note:
boolean	Refund is successfully approved or disapproved by admin
Algorithm Specification: IF refundID found Admin views refund IF Admin approves refund Refund.updateRefundStatus(True) ELSE Refund.updateRefundStatus(False) Return True ELSE Return False	
Misc. Notes: None	

Verifying and Validating Class and Method Design

Edit Post

In the figure below, it is evident that the methods' name ("editPost()"), class name and pass method are the same in both Class Diagram and Method Specification.

Messages Sent & Arguments Passed & Return Data Type: ClassName.MethodName:	Argument : Data Type:	Return Data Type:
PostControl.viewPost()	int: postID	Object
PostControl.editPost()	int: postID String: details float: wantedPrice	void

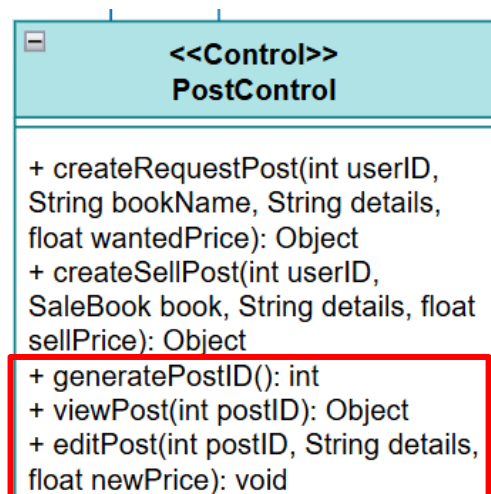


Figure 12 Comparing Method “editPost()” Between Class Diagram and Method Specification

Approve Refund

In the figure below, it is evident that the methods' name ("approveRefund()"), class name and pass method are the same in both Class Diagram and Method Specification.

Messages Sent & Arguments Passed & Return Data Type: ClassName.MethodName:	Argument : Data Type:	Return Data Type:
RefundControl.viewRefund()	int: refundID	Object
AdminController.approveRefund()	int: refundID	boolean

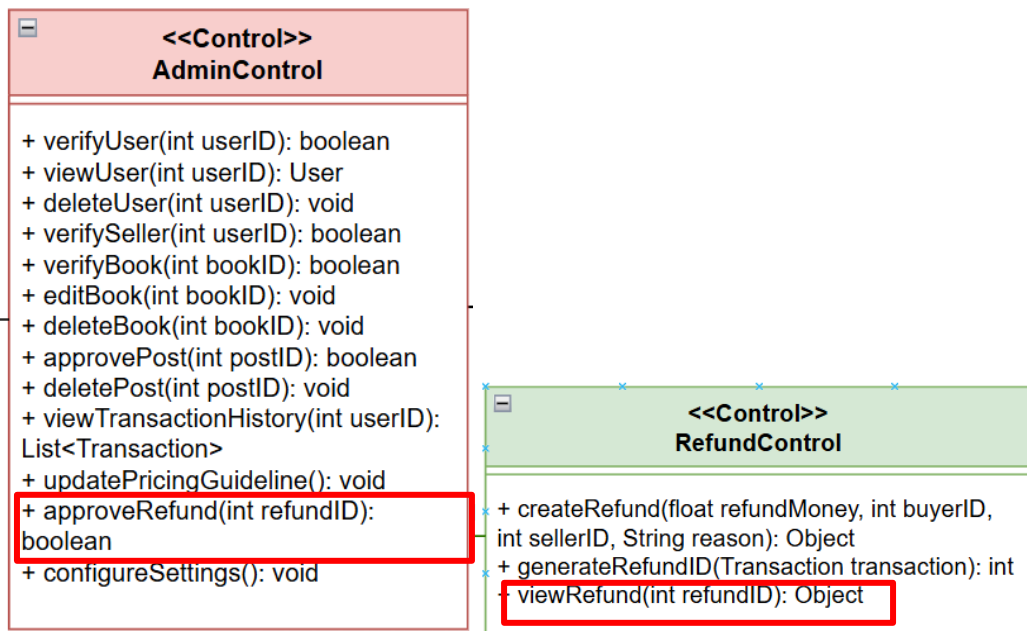


Figure 13 Comparing Method “editPost()” Between Class Diagram and Method Specification

User Interface Design Principles

Layout

User interface split into 2 sections:

1. Header Section:

- a. Contains the logo, navigation links (ค้นหาหนังสือ, ประวัติการสั่งซื้อ, ติดต่อเรา), and the user profile picture.
- b. Split this into two parts: the left side for the logo and the right side for the navigation and profile picture.

2. Content Section:

Divided into two main rows:

- a. Row 1 (Top Row):
 - This row will contain the **search functionality** with:
 1. A search bar where users can type in the book title.
 2. Filter buttons like "ค้นหาด้วยเงื่อนไขพิเศษ" and "ค้นหาด้วยรายละเอียดเพิ่มเติม."
 - This row acts as the user control area, allowing them to refine their search.
- b. Row 2 (Bottom Row):
 - This row will contain the **book listing** in a grid format.
 - The grid will display each book in a card layout, including details like title, author, price, description, and action buttons (ดูข้อมูล and เพิ่มใส่ตะกร้า).
 - To ensure responsiveness, you can design the grid to show a maximum of three book cards per row on larger screens, reducing to one or two cards per row on smaller screens.

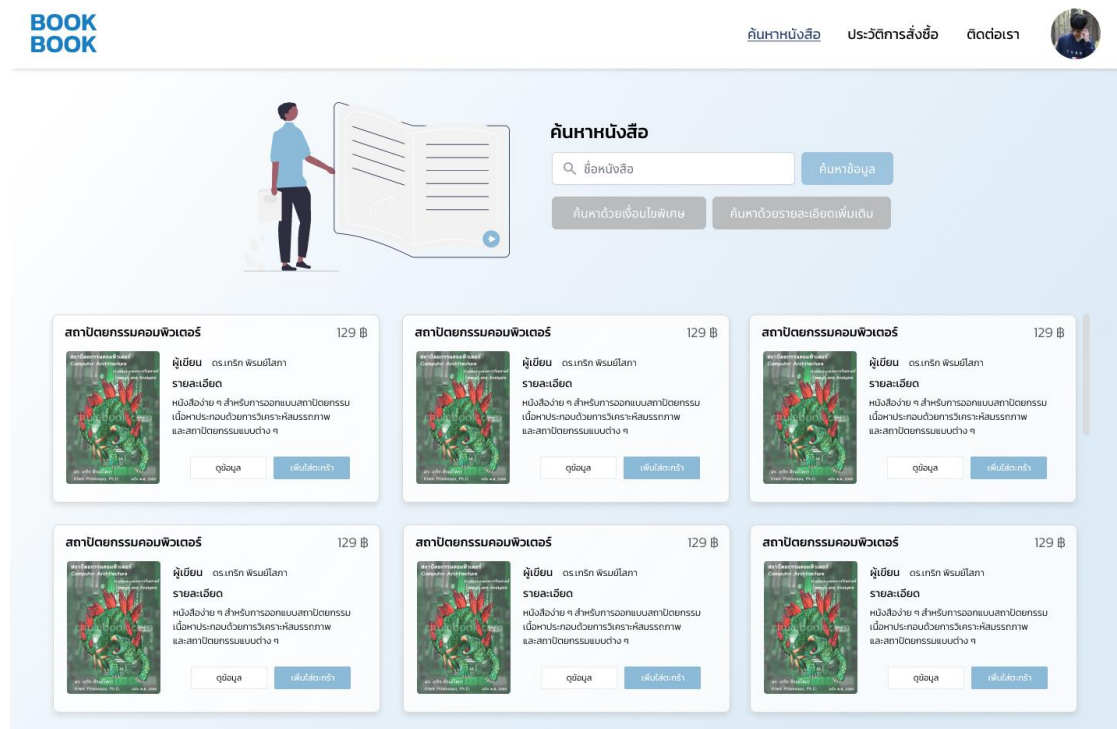


Figure 14 User Interface Layout

Content Awareness

The website's content awareness is strong, with a prominent search area guiding users to find books easily. The labeled fields for book title, author, publisher, and ISBN provide clear entry points for specific searches. Buttons for advanced search options allow users to refine their results, enhancing usability. The layout emphasizes the search functionality, making the page's purpose immediately clear. Spacing and color contrasts improve interaction with each element, ensuring a user-friendly experience.

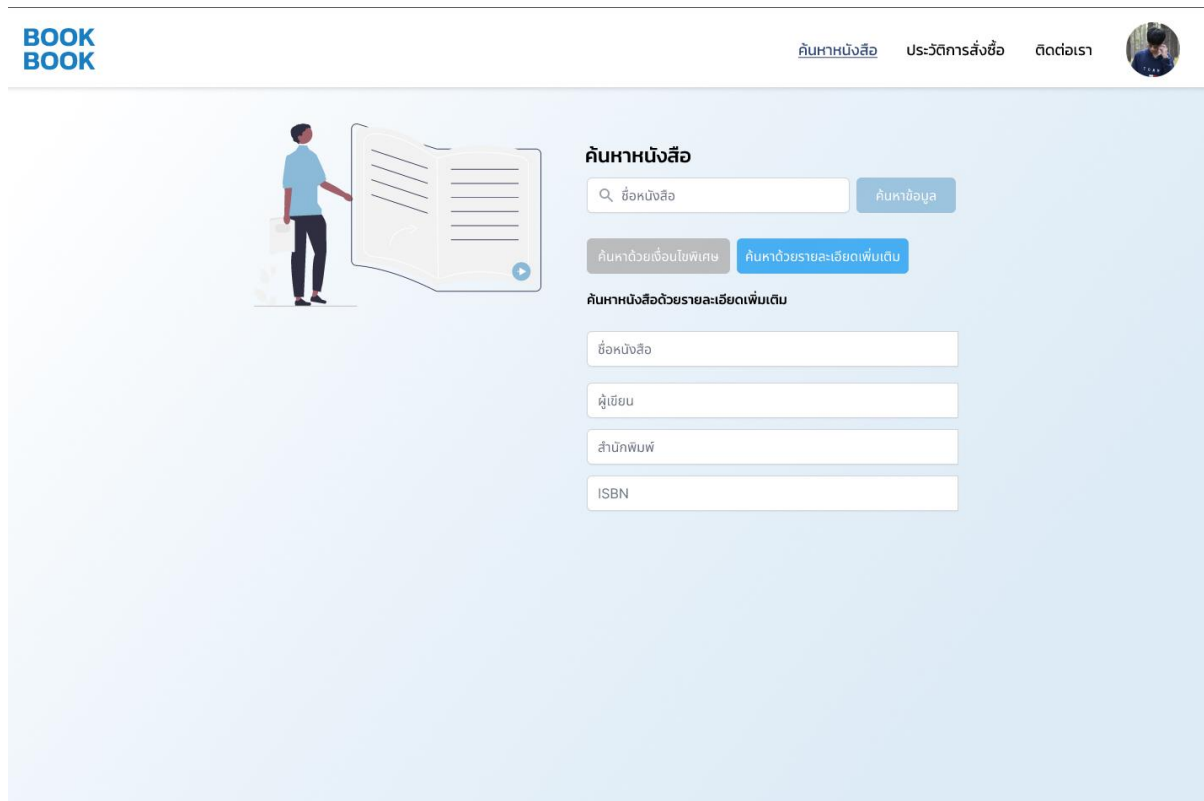


Figure 15 Content Awareness in User Interface

Aesthetics

The aesthetics of this design are clean and minimal, creating a professional and user-friendly interface. The color scheme, primarily light blue with accents of white and gray, gives a calming and approachable feel, making it easy for users to focus on the content without distractions. The layout is well-organized, with ample spacing around elements, enhancing readability and visual clarity. The use of consistent fonts and subtle icons reinforces a cohesive visual identity. Overall, the design balances simplicity with functionality, creating a visually pleasing experience that feels intuitive and inviting.

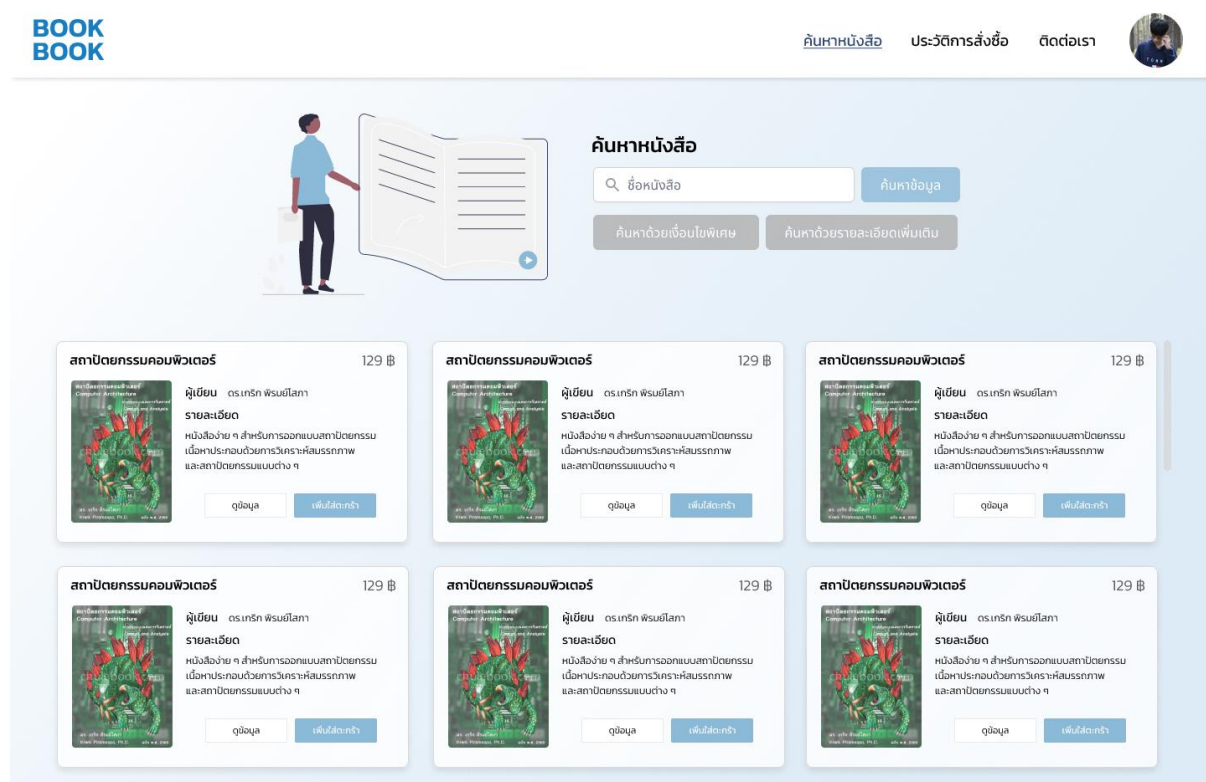


Figure 16 Aesthetics in User Interface

User Experience

The user experience on these two pages is straightforward and intuitive, with clearly separated search options. The first page focuses on detailed search fields (e.g., title, author, publisher), while the second page provides checkbox filters for special conditions like second-hand books. The design offers clear visual feedback, such as active button colors and selected checkboxes, enhancing user confidence. Consistent layout and well-spaced elements create a clean, accessible interface, making it easy for users to customize their search without confusion.

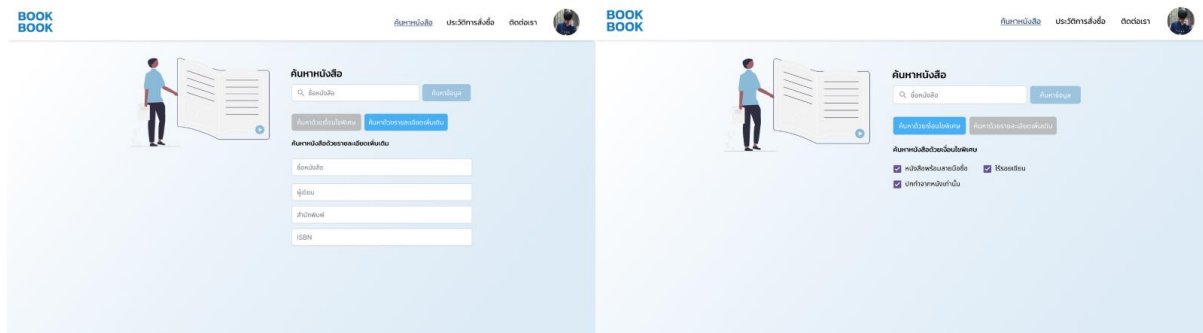


Figure 17 User Experience

Consistency

The system has a **design system** in place that defines colors, fonts, font sizes, and component creation. Each component has **variances** to allow for easy editing and design adjustments, ensuring that every **card** has a consistent appearance. The overall theme of the system aligns in a unified direction, with no use of colors outside the predefined palette and no unusual fonts or inappropriate font sizes.

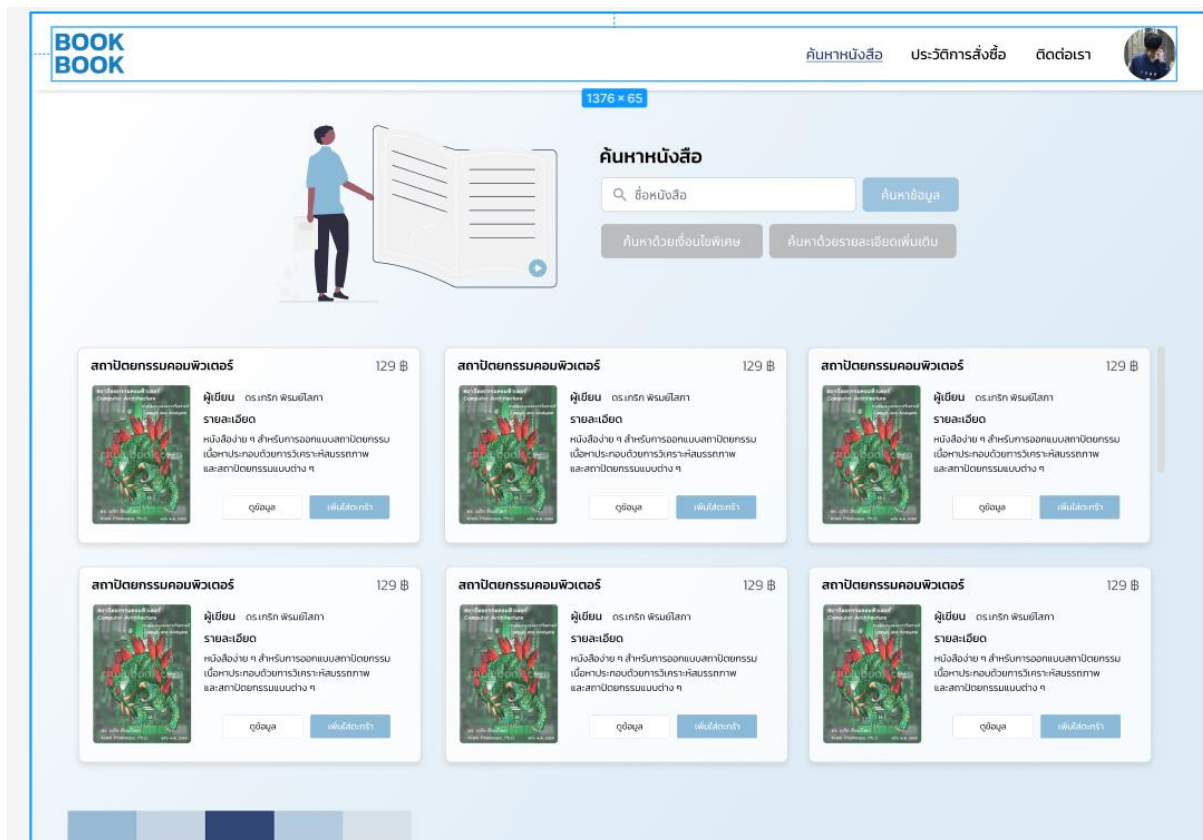


Figure 18 Consistency in User Interface

Minimal User Effort

The design minimizes user effort by keeping search filters and results accessible side-by-side. Users can quickly scan book details in consistent card layouts on the left, while applying filters on the right without excessive navigation. Checkboxes for special conditions make filter adjustments easy, reducing cognitive load. Clear labels and consistent colors guide users to interactive elements, simplifying the process. Overall, the design reduces clicks and scrolling, providing a smooth and efficient book-searching experience.

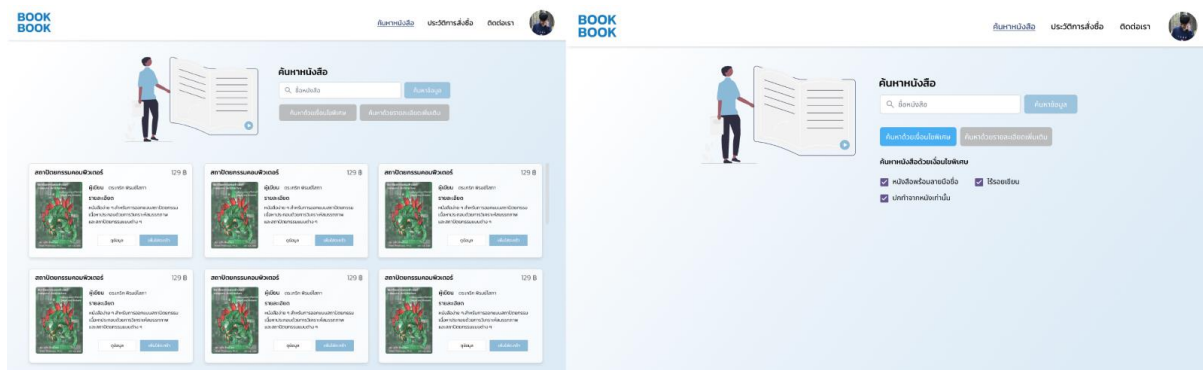


Figure 19 Minimal User Effort

Non-functional Requirements and Their Effect on System Design

Type of Non-Functional Requirement	Requirement	Effect for user interface design
Operational requirement	The system shall operate on desktop and mobile.	Design a responsive, adaptive UI layout that optimizes display and interaction for both desktop and mobile devices.
	The system shall operate on Windows, MacOS, Android and iOS	Use a cross-platform development framework or web-based solution to ensure compatibility across multiple operating systems.
	The system shall be connected to ISBN database	Implement API integration to the ISBN database, ensuring seamless data exchange with proper error handling and security protocols for real-time data access.
	The system shall be able to record users' previous transactions.	Design a database schema to log and retrieve user transaction histories with indexing for efficient data retrieval, ensuring scalability and data integrity.
Performance requirements	The system shall be able to retrieve the data of any book in 2 seconds or less.	Use caching for frequently accessed book data, optimize database queries, and reduce network latency to ensure quick data retrieval.

	The system shall be able to add new data to any book in 2 seconds or less.	Optimize database inserts and implement efficient indexing strategies for fast data entry, ensuring minimal processing delay.
	The system shall be able to retrieve the data of any post in 2 seconds or less.	Use query optimization and caching for posts, ensuring that retrieval operations are efficient and within the required time frame.
	The system shall be able to add new data to any post in 2 seconds or less.	Utilize efficient insert techniques and ensure that database updates on posts are optimized for performance.
	The system shall receive and confirm payment after the payment is made by the buyer in 1 second or less.	Handle payment confirmation asynchronously to keep server response times fast..
	The system shall be able to notify any user about critical transactions in the system after a transaction is made in less than 1 seconds or less.	Use real-time notifications to inform users immediately about key transactions.
Security Requirements	The system shall authenticate users using username-password.	Implement a secure authentication mechanism with hashed passwords, and enforce password complexity policies.
	The system shall be able to secure every user's transaction in private.	Use end-to-end encryption for transaction data and secure data storage practices to protect user transaction privacy.
	The system shall allow only administrators to modify data of books and posts.	Limit data modification to admins with access control mechanisms.

	The system shall prevent users from accessing the admin's data and authorization.	Implement strict access rules to separate admin data from user data, ensuring confidentiality.
Cultural and political requirements	The system shall operate in Thai.	Enable internationalization techniques to support Thai language content across the application, ensuring that text is correctly formatted and displayed in Thai.
	The system shall use Thai Baht currency	Integrate currency localization to display monetary values in Thai Baht, ensuring accurate conversions and formatting for Thai currency.
	The system shall follow Thailand's Computer-related Crime Act 2017 and Personal Data Protection Act 2019.	Implement privacy-by-design principles and data protection measures to comply with Thailand's legal requirements, including data minimization, consent handling, and secure data storage.

Table 2 Non-functional Requirements and Effect on User Interface Design

Detail Real Use Case Description

Register as Buyer

Use Case Name: Register as buyer	ID: 5	Importance Level: High
Primary Actor: new buyer	Use Case Type: Detail, Real	
Stakeholders and Interests: Buyer registers on the website to gain access to search, view, and purchase books.		
Brief Description: This use case describes how a new buyer registers on the website to create an account, allowing them to interact with the platform and view the book feed.		
Trigger: Buyer accesses the website for the first time and initiates the registration process. Type: External		
Relationships: Association: new buyer, Register as buyer Include: - Extend: - Generalization: -		
Normal Flow of Event: 1. Buyer accesses the website for the first time. 2. The buyer selects the "Register" option to initiate the registration process.		

3. The buyer is required to provide their email, password, and username.

If the buyer wishes to register for full access to book purchases, execute sub-flow **S-1**.

If the buyer doesn't complete all of the required fields or fill the invalid data, execute exceptional-flow S-X2.

If the user provides an email or username that is already registered, execute exceptional-flow S-X1.

4. The buyer reviews the provided information and selects the "Confirm" button to submit the registration.
5. The system sends OTP to the buyer's email.
6. The buyer is redirected to the OTP confirmation page and has two minutes to enter the OTP before it expires. But if it expired, the system will send a new OTP.
7. The buyer enters the OTP. and then clicks at the confirm button to continue the registration process.

If the buyer requests a new OTP, they can select the "Resend OTP" button, initiating sub-flow **S-2**.

If the OTP is incorrect, execute exceptional-flow S-X4..

8. If the OTP is correct, the system redirects the buyer to the terms and agreements page, containing the Terms of Service, Rules, and PDPA statements.
9. The buyer reads and agrees to the Terms of Service by checking the agreement box, then clicks "Confirm" to proceed.

If the doesn't agrees to the Terms of Service, execute exceptional-flow S-X3.

10. The system completes the registration process, creating a new account for the buyer.
11. Upon successful registration, the buyer is redirected to the feed page of the website, where they can begin interacting with the platform.

Subflows:

S-1: Register with purchase access

1. The buyer provides additional information, including their full name, address, and phone number.

If the buyer doesn't complete all of the required fields or fill the invalid data, execute exceptional-flow S-X2.

2. After email verification, the system sends an OTP to the buyer's phone number via SMS.
3. The system redirects the buyer to the OTP confirmation page. They have 2 minute to fill the OTP until the OTP is invalid. But if it expired, the system will send a new OTP.
4. The buyer enters the OTP.

If the buyer requests a new OTP, they can select the "Resend OTP" button, initiating sub-flow **S-3**.

If the OTP is incorrect, execute exceptional-flow S-X4.

5. If the OTP is correct, the system redirects the buyer to the terms and agreements page, containing the Terms of Service, Rules, and PDPA statements.
6. The buyer reads and agrees to the Terms of Service by checking the agreement box, then clicks "Confirm" to proceed.

If the doesn't agrees to the Terms of Service, execute exceptional-flow S-X3.

7. The system completes the registration process, creating a new account for the buyer.
8. Upon successful registration, the buyer is redirected to the feed page of the website, where they can begin interacting with the platform.

S-2: Resend OTP in email confirm page

1. When the buyer selects the "Resend OTP" button, the previous OTP is immediately invalidated, and a new OTP is sent to the buyer's email. The buyer has two minutes to enter the OTP before it expires. But if it expired, the system will send a new OTP.

2. The buyer enters the OTP.

If the OTP is incorrect, execute exceptional-flow S-X4.

3. If the OTP is correct, the system continues the registration process.

S-3: Resend OTP in phone number confirm page

1. When the buyer selects the "Resend OTP" button, the previous OTP is immediately invalidated, and a new OTP is sent to the buyer via SMS. The buyer has two minutes to enter the OTP before it expires. . But if it expired, the system will send a new OTP.

2. The buyer enters the OTP.

If the OTP is incorrect, execute exceptional-flow S-X4.

3. If the OTP is correct, the system continues the registration process.

Alternate/Exceptional Flows:

S-X1: The system displays the error message: "The account is already registered."

1. The buyer tries to register with a username, email, or phone number that is already registered.
2. The system displays an error message, "The account is already registered," above the "Confirm" button. The buyer remains on the registration page with all previously entered information retained.
3. Once the buyer enters a unique username, email, and phone number, they can select the "Confirm" button to continue with the registration process.

S-X2: The system displays the error message: “Please complete all required fields.”

1. Required fields are missing or contain invalid data.
2. The system displays an error message, “Please complete all required fields,” above the "Confirm" button. The buyer remains on the registration page with all previously entered information retained.
3. Once the buyer fills in all required fields with valid data, they can select the "Confirm" button to continue with the registration process.

S-X3: The system displays the error message: “Please agree to the Terms of Service.”

1. The buyer fails to check the "Terms of Service" checkbox.
2. The system displays an error message, “Please agree to the Terms of Service,” above the "Confirm" button. The buyer remains on the registration page with all previously entered information retained.
3. Once the buyer checks the "Terms of Service" checkbox, they can select the "Confirm" button to continue with the registration process.

S-X4: The system displays the error message: “The OTP is incorrect.”

1. The buyer enters an incorrect OTP.
2. The system displays an error message, “The OTP is incorrect,” above the "Confirm" button, and the OTP field resets to blank. The buyer remains on the OTP confirmation page.
3. Once the buyer enters a correct OTP, they can select the "Confirm" button to continue with the registration process.

Search Books

Use Case Name: Search books	ID: 4	Importance Level: High
Primary Actor: Buyer	Use Case Type: Detail, Real	
Stakeholders and Interests: Buyer-want to find a book post with book detail.		
Brief Description: This use case describes how buyers can search for a book post with book detail, book special description.		
Trigger: Buyer-want to find a book post with book detail. Type: External		
Relationships: Association: Buyer, Book post Include: - Extend: Apply search filter Generalization: -		
Normal Flow of Event: 1. Buyer taps the search bar and types in the book name. If the buyer wants to search with additional details, he or she can tap navigate to the “Details” column and execute S-1: Search with Additional Details . If the buyer wants to search with special conditions, he or she can navigate to the “Special” column and execute S-2: Search with Special Conditions .		

2. Buyer taps the search button, redirecting them to the book post page displaying the search results.

If the book is found, execute **S-3**.

If the book is not found, execute **S-4** .

If the buyer wants to categorize the book post, they can navigate to the “Sort Search Result” column and execute **S-5: Sort Search Results**.

If the buyer taps the search button without filling the book name in the search bar, execute exceptional-flow S-X1.

3. Buyer can use the scroll bar to navigate up and down to view more posts

Subflows:

S-1: Search with Additional Details

1. Buyer select the option to fill in details
 - a. For author, publisher, and ISBN, buyers have to type in the details, default is an empty field indicating no filter over this field.
 - b. For content and language, buyers have to choose the available choices, default is “any” indicating no filter over this field.
 - c. For publication date, buyers can choose a range of dates by typing in 2 dates, or choose anytime, default for both fields is “any” indicating no date constraints.
 - d. For the price range, buyers can choose a range of prices by typing in 2 numbers, default for both fields is “any” indicating no date constraints.

S-2: Search with Special Conditions

1. Execute use case : Search with special conditions

S-3: Search found

1. The system shows the search result

S-4: Search not found

1. The system notify the buyer “Book not found”
2. The buyer begins the new search.

S-5: Sort search results

1. The buyer taps on the “Sort” button.
2. The system shows ways of sorting for the buyer to choose
 - a. Price: High to Low
 - b. Price: Low to High
 - c. Latest
 - d. Most Popular
3. The buyer taps on the option he or she wants
4. The system rearrange the search results and sorts

Alternate/Exceptional Flows:

S-X1: The system shows an error message “Please Enter Book Name”

1. Buyer taps the search button without entering a book name in the search bar.
2. The search bar turns red, indicating an error.
3. Buyer can click on the search bar, which resets the error indicator (removes the red highlight) and allows them to enter a book name.
4. Once the buyer enters a valid book name, they can tap the search button again, redirecting them to the book post page with relevant search results.

User Interface Design

Register as Buyer

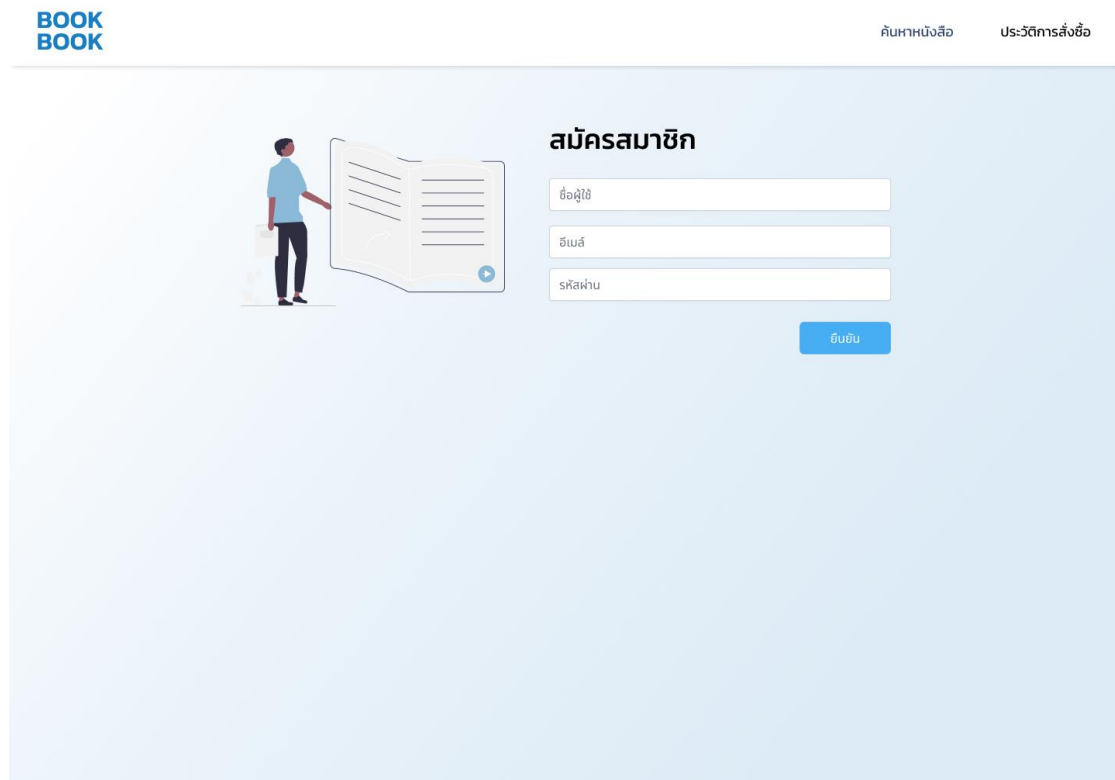
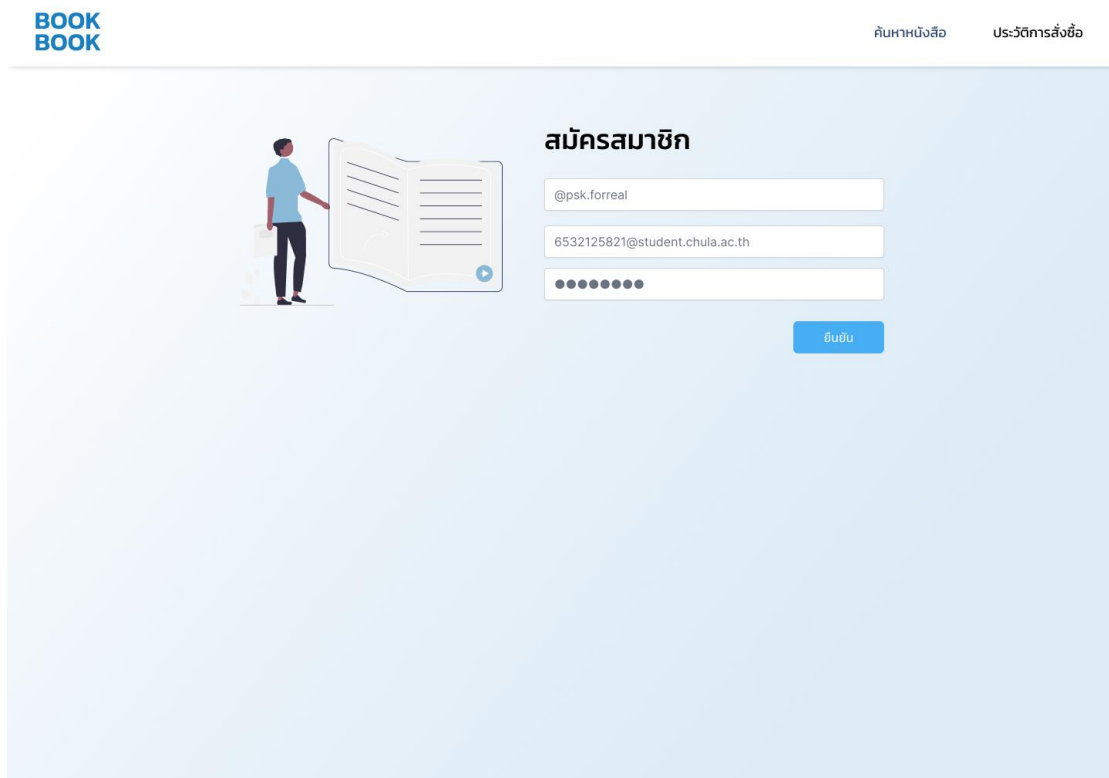
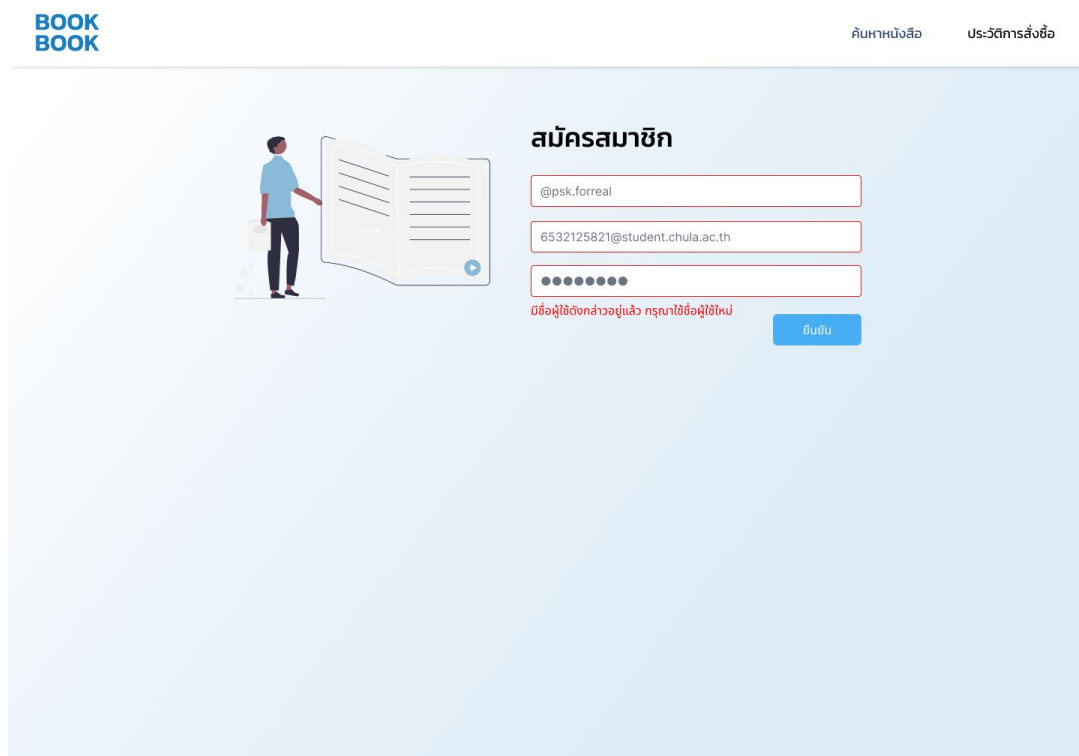


Figure 20 Register



The screenshot shows the 'BOOK BOOK' registration page. The header includes the 'BOOK BOOK' logo on the left and navigation links 'ค้นหาหนังสือ' and 'ประวัติการสั่งซื้อ' on the right. The main content area features an illustration of a person standing next to a large open book on the left. To the right, under the heading 'สมัครสมาชิก', there are three input fields: the first contains '@psk.forreal', the second contains '6532125821@student.chula.ac.th', and the third is a password field with eight dots. A blue 'ยืนยัน' button is positioned below the password field.

Figure 21 Register - Filled-in information



This screenshot shows the same registration page as Figure 21, but with an error message. The input fields contain the same information: '@psk.forreal', '6532125821@student.chula.ac.th', and a password field with eight dots. Below the password field, a red error message reads 'มีชื่อผู้ใช้ดังกล่าวอยู่แล้ว กรุณาใช้ชื่อผู้ใช้ใหม่'. The blue 'ยืนยัน' button remains at the bottom right.

Figure 22 Register - The account is already registered

BOOK BOOK

ค้นหาหนังสือ ประวัติการสั่งซื้อ

กรอกข้อมูลผู้ซื้อ

ชื่อ นามสกุล

เบอร์โทร

ที่อยู่

[ยืนยันอีเมล](#)

Figure 23 Register - Fill in buyer details

BOOK BOOK

ค้นหาหนังสือ ประวัติการสั่งซื้อ

กรอกข้อมูลผู้ซื้อ

ชื่อ นามสกุล

เบอร์โทร

ที่อยู่

กรุณากรอกข้อมูลให้ครบถ้วน

[ยืนยันอีเมล](#)

Figure 24 Register - Please complete all required fields

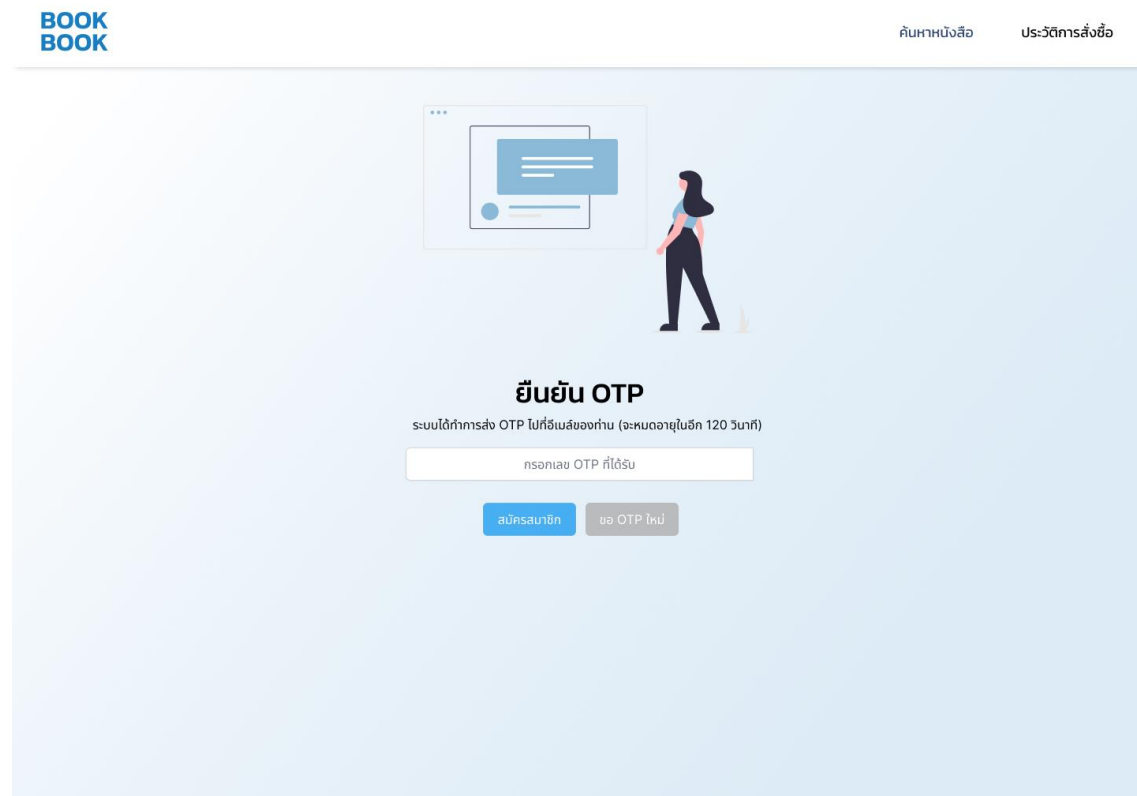


Figure 25 Register - Verify email with OTP

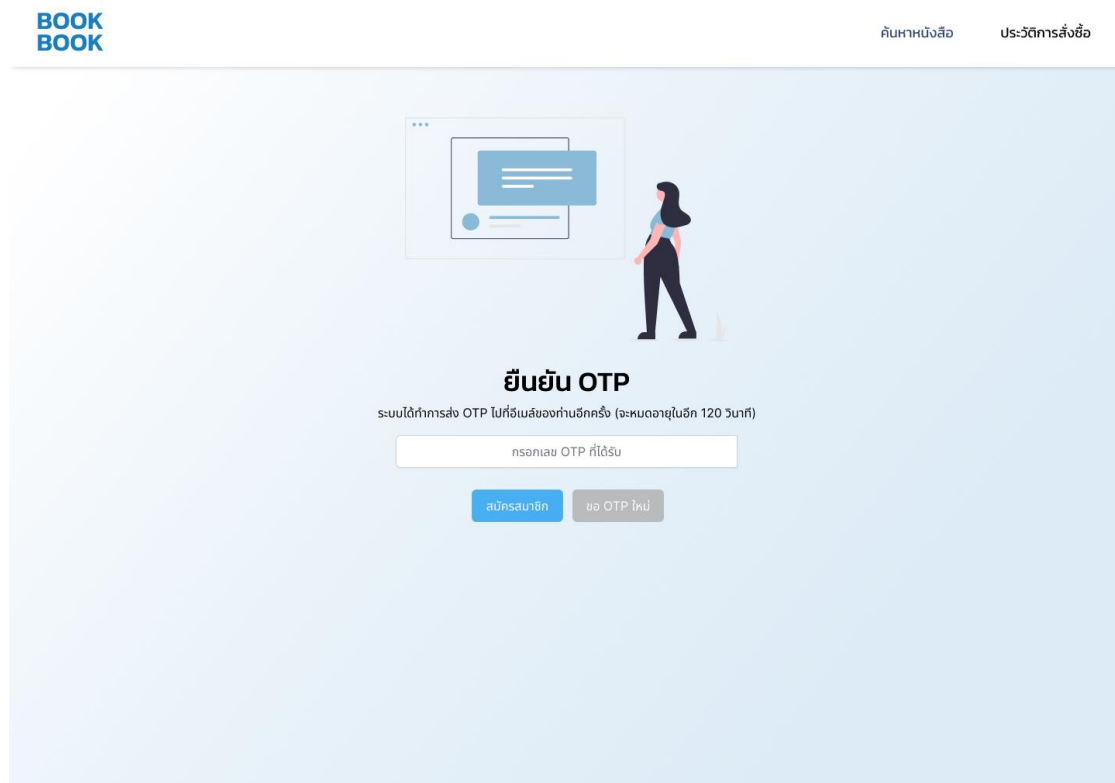


Figure 26 Register - Resend OTP

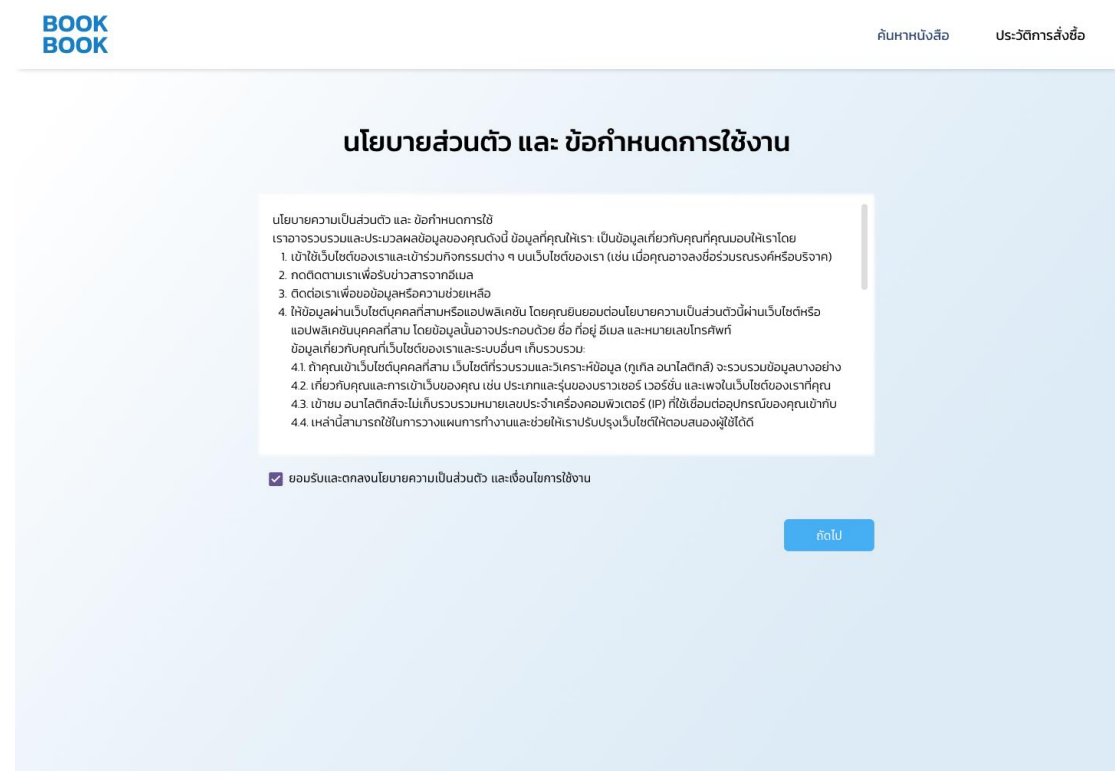


Figure 27 Register - Terms and Conditions of Use

BOOK BOOK

ค้นหาหนังสือ ประวัติการสั่งซื้อ

นโยบายส่วนตัว และ ข้อกำหนดการใช้งาน

นโยบายความเป็นส่วนตัว และ ข้อกำหนดการใช้

เราขอรวบรวมและประมวลผลข้อมูลของคุณดังนี้ ข้อมูลที่คุณให้เรา: เป็นข้อมูลเกี่ยวกับคุณที่คุณมอบให้เราโดย

1. เข้าใช้เว็บไซต์ของเราและเข้าร่วมกิจกรรมต่าง ๆ บนเว็บไซต์ของเรา (เช่น เมื่อคุณอาจลงชื่อสมัครใช้หรือบริจาค)
2. กดติดตามเราเพื่อรับข่าวสารรายสัปดาห์
3. ติดต่อเราเพื่อขอข้อมูลหรือความช่วยเหลือ
4. ให้ข้อมูลผ่านเว็บไซต์บุคคลที่สามหรือแอปพลิเคชัน โดยคุณยินยอมต่อนโยบายความเป็นส่วนตัวนี้ผ่านเว็บไซต์หรือแอปพลิเคชันบุคคลที่สาม โดยข้อมูลนี้อาจประกอบด้วย ชื่อ ที่อยู่ อีเมล และหมายเลขโทรศัพท์

ข้อมูลเกี่ยวกับคุณที่เว็บไซต์ของเราและระบบอื่นๆ เก็บรวบรวม:

- 4.1 ถ้าคุณเข้าเว็บไซต์บุคคลที่สาม เว็บไซต์ที่รวบรวมและวิเคราะห์ข้อมูล (กูเกิล ออราไดคัล) จะรวบรวมข้อมูลบางอย่าง
- 4.2 เกี่ยวกับคุณและการเข้าเว็บไซต์ของคุณ เช่น ประเภทและรุ่นของเบราว์เซอร์ เวอร์ชัน และเพจในเว็บไซต์ของเราที่คุณ
- 4.3 เข้าชม ออราไดคัลจะไม่เก็บรวบรวมหมายเลขประจำเครื่องคอมพิวเตอร์ (IP) ที่ใช้เชื่อมต่ออุปกรณ์ของคุณเข้ากับ
- 4.4 เหล่านี้สามารถใช้ในการวางแผนการทำงานและช่วยให้เราปรับปรุงเว็บไซต์ให้ตอบสนองผู้ใช้ได้ดี

☐ ยอมรับและตกลงนโยบายความเป็นส่วนตัว และเงื่อนไขการใช้งาน


กรุณายอมรับและตกลงนโยบายความเป็นส่วนตัวและเงื่อนไขการใช้งาน

ถัดไป

Figure 28 Register - Please agree to the Terms of Service

BOOK BOOK

ค้นหาหนังสือ ประวัติการสั่งซื้อ



ยืนยัน OTP

ระบบได้ทำการส่ง OTP ไปที่เบอร์มือถือของท่าน (จะหมดอายุในอีก 120 วินาที)

กรอกเลข OTP ที่ได้รับ

สมัครสมาชิก

ขอ OTP ใหม่

Figure 29 Register - Register with purchase access

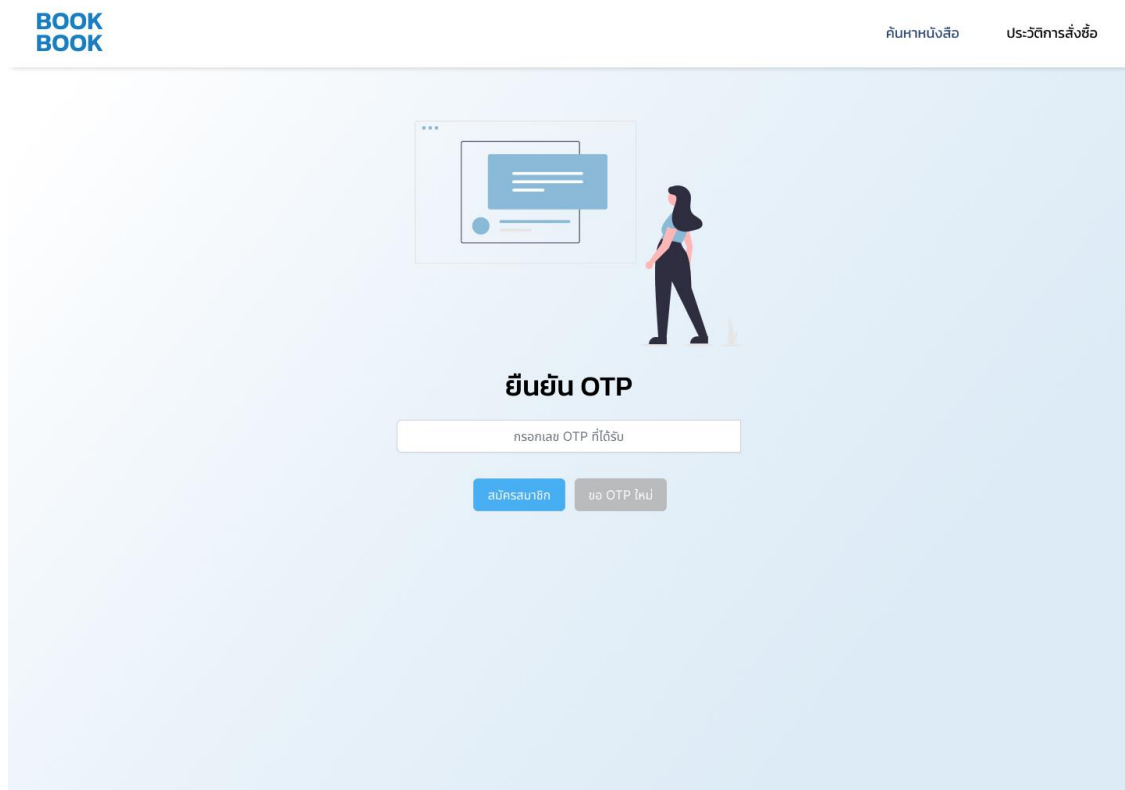


Figure 30 Register - Request OTP by phone number

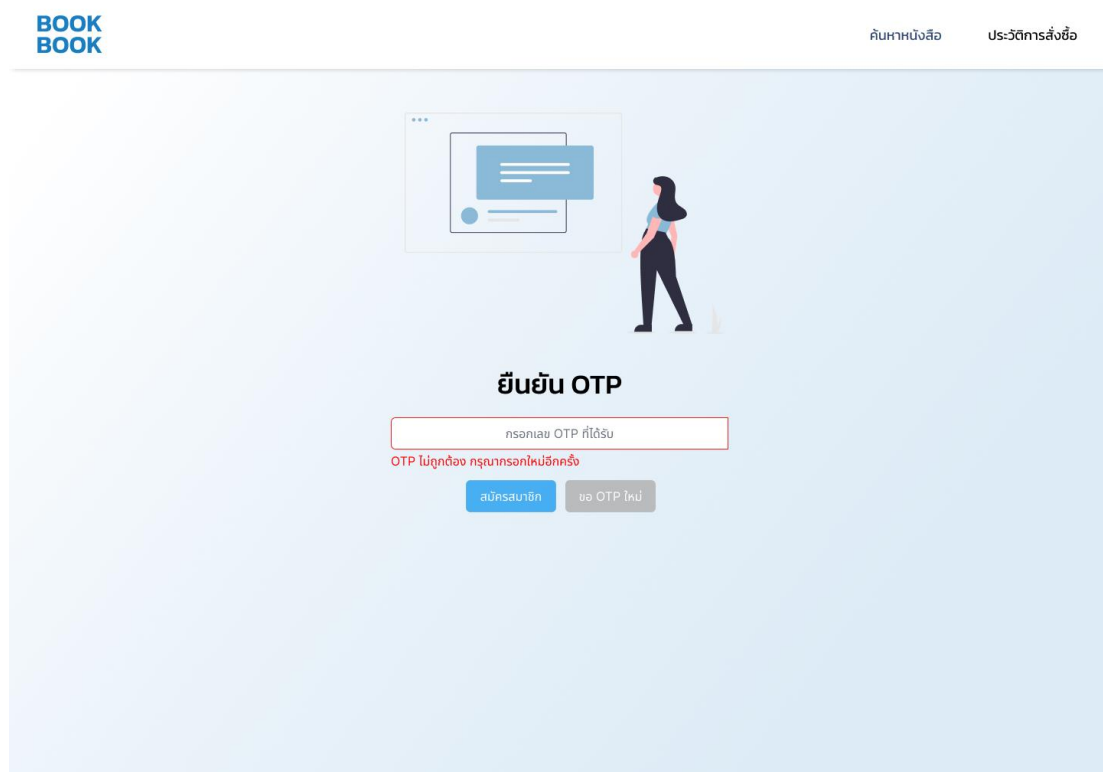


Figure 31 Register - The OTP is incorrect

Search Books

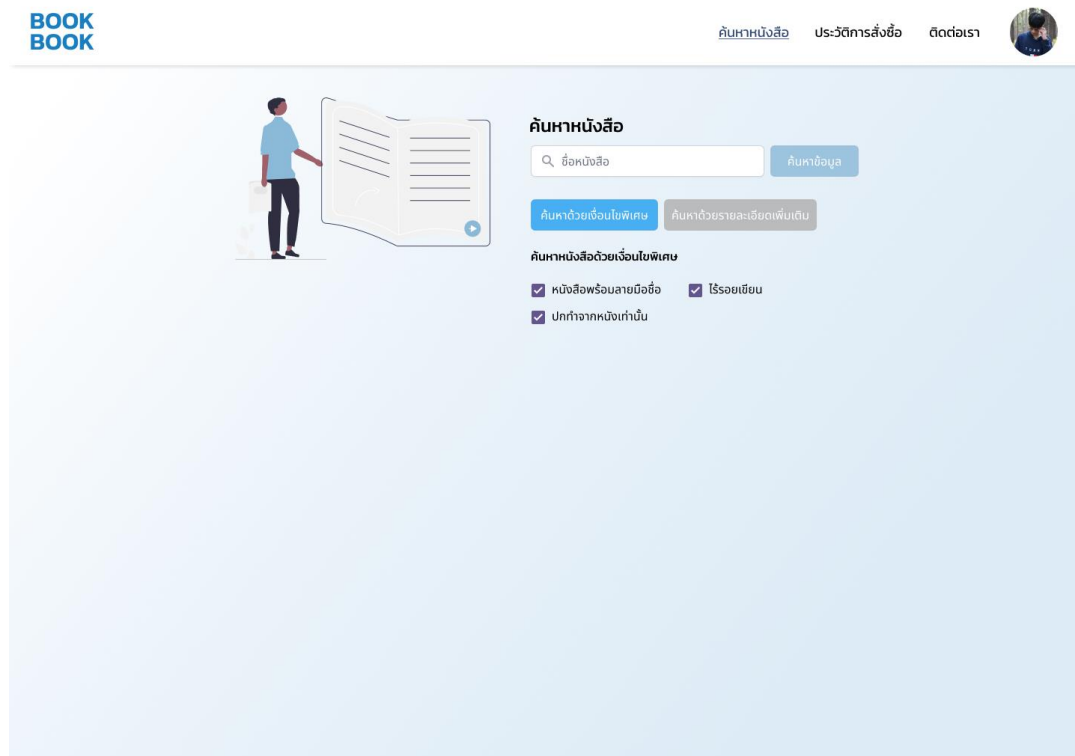


Figure 32 Search books - Search with Special Conditions

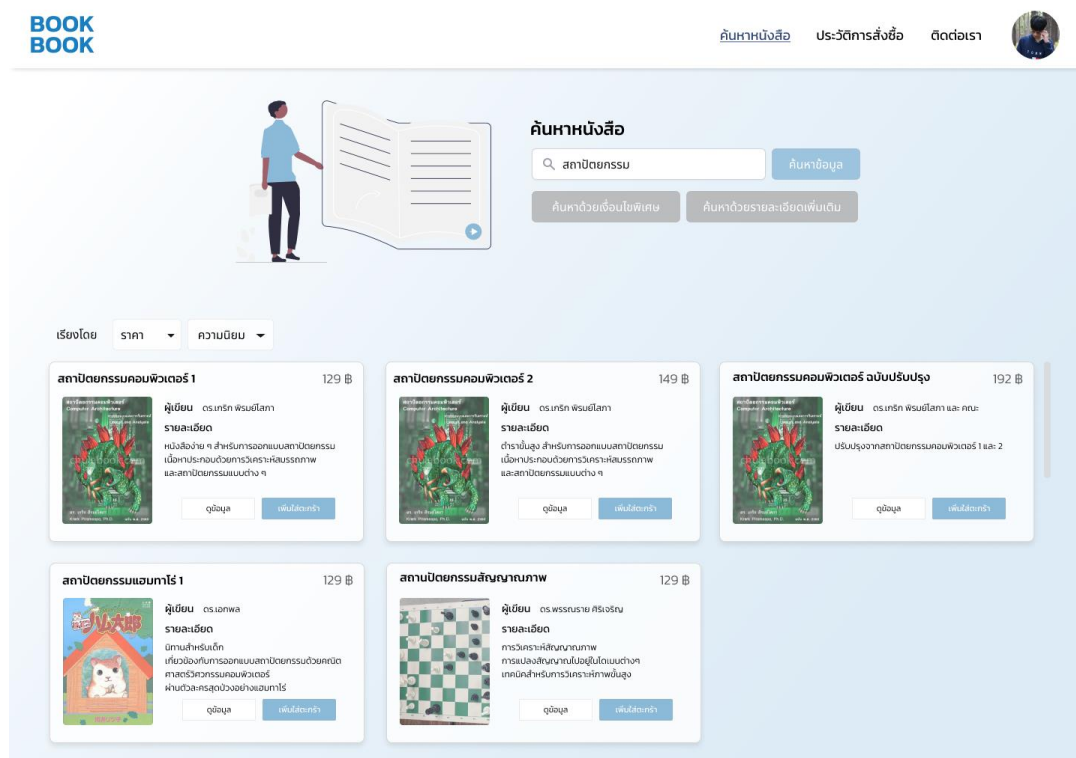


Figure 33 Search books - Search found

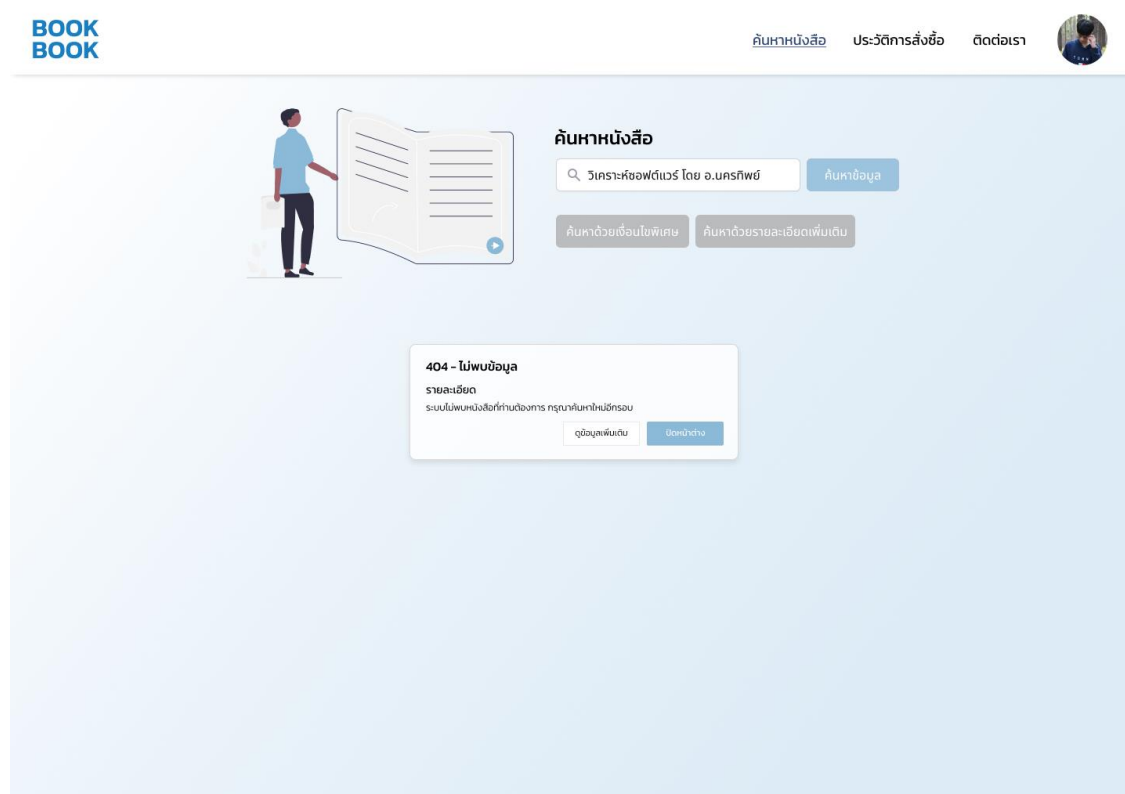


Figure 34 Search books - Search not found

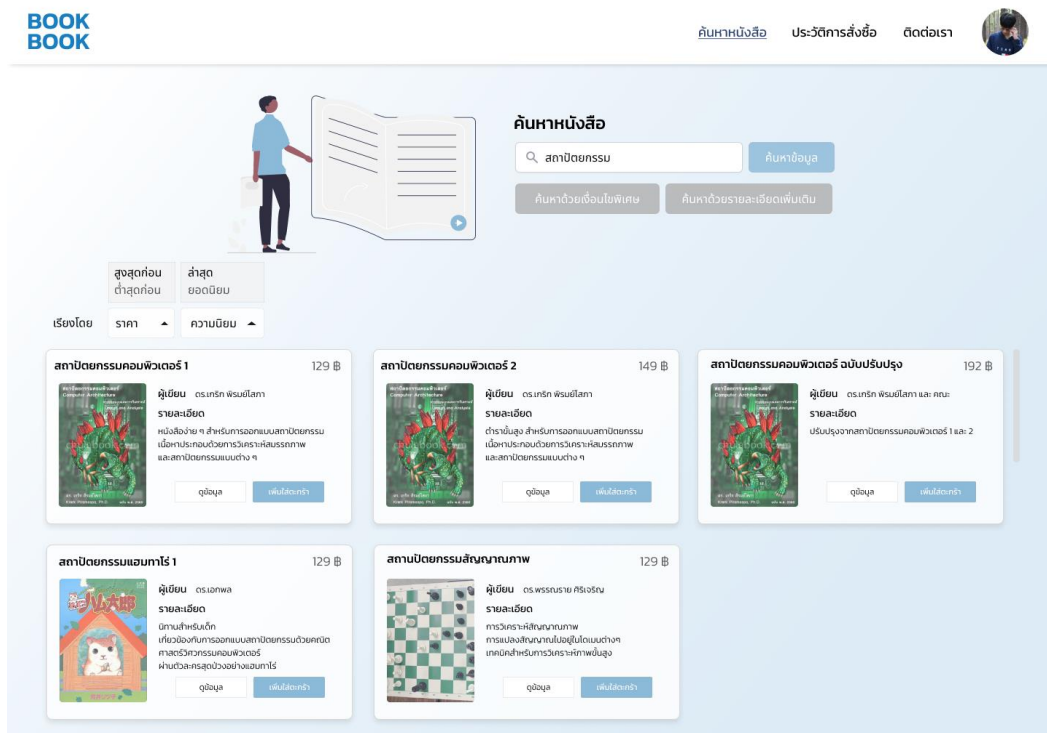


Figure 35 Search books - Sort search results

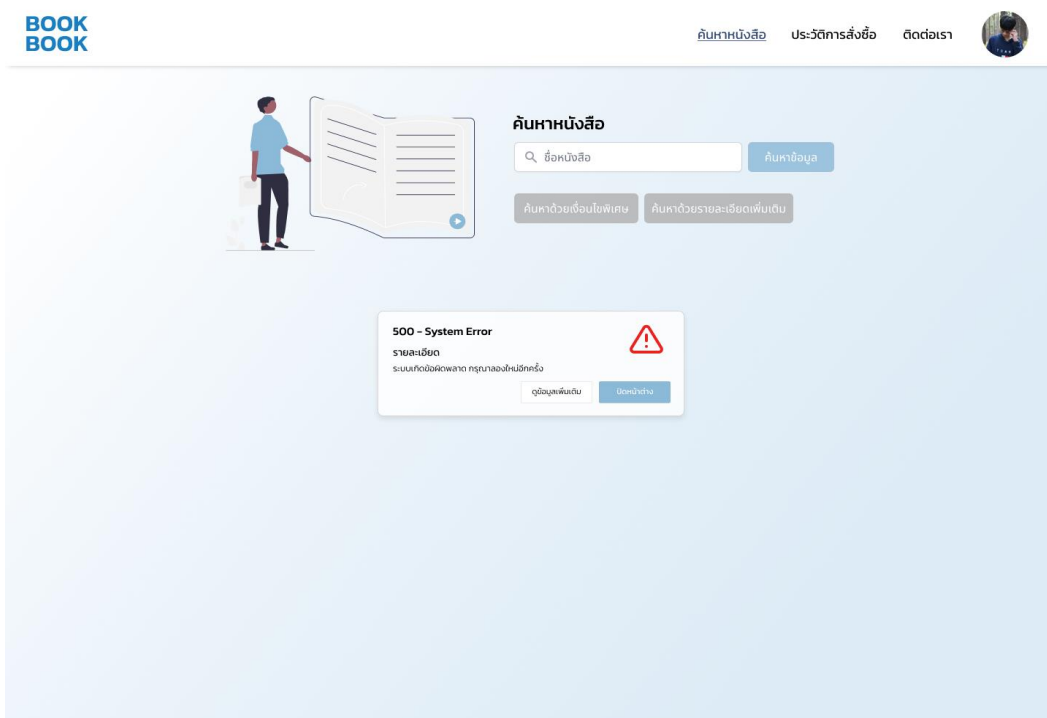


Figure 36 Search books - System Error

Non-functional Requirements and their Effect on Physical Architecture Design

Type of Nonfunctional Requirement	Requirement	Effect for designing architectural model
Operational requirements	The system shall operate on desktop and mobile.	Client tier supports both platforms
	The system shall operate on Windows, MacOS, Android and iOS	Client tier supports a wide range of operating systems, both desktop and mobile.
	The system shall be connected to ISBN database	Requires optimizing database queries, indexing, and possibly sharding the database.
	The system shall be able to record users' previous transactions.	Implement database and logging systems.
	The system shall be able to retrieve the data of any book in 2 seconds or less.	Utilize load balancer to distribute traffic between 3 web servers.
	The system shall be able to add new data to any book in 2 seconds or less.	Employ a replica database via TCP/IP for further data reliability.
	The system shall be able to retrieve the data of any post in 2 seconds or less.	
	The system shall be able to add new data to any post in 2 seconds or less.	LAN directly connecting the web servers and main database
	The system shall receive and confirm payment after the payment is made by the buyer in 1 second or less.	

	The system shall be able to notify any user about critical transactions in the system after a transaction is made in less than 1 seconds or less.	
Security Requirements	The system shall authenticate users using username-password.	HTTPS encrypted communication.
	The system shall be able to secure every user's transaction in private.	Firewall monitoring incoming and outgoing traffic.
	The system shall allow only administrators to modify data of books and posts.	3rd party auth system, with an additional factored authentication. For enhanced user's security.
	The system shall prevent users from accessing the admin's data and authorization.	Trustworthy 3rd party payment system.
Cultural and political requirements	The system shall operate in Thai.	System and data security, using trustworthy authentication systems and internet firewalls.
	The system shall use Thai Baht currency	
	The system shall follow Thailand's Computer-related Crime Act 2017 and Personal Data Protection Act 2019.	Low data sharing, only the authentication system is a 3rd party service.

Table 3 Non-functional Requirements and Effect on Physical Architecture Design

Overview of the System Infrastructure Design

Network Diagram

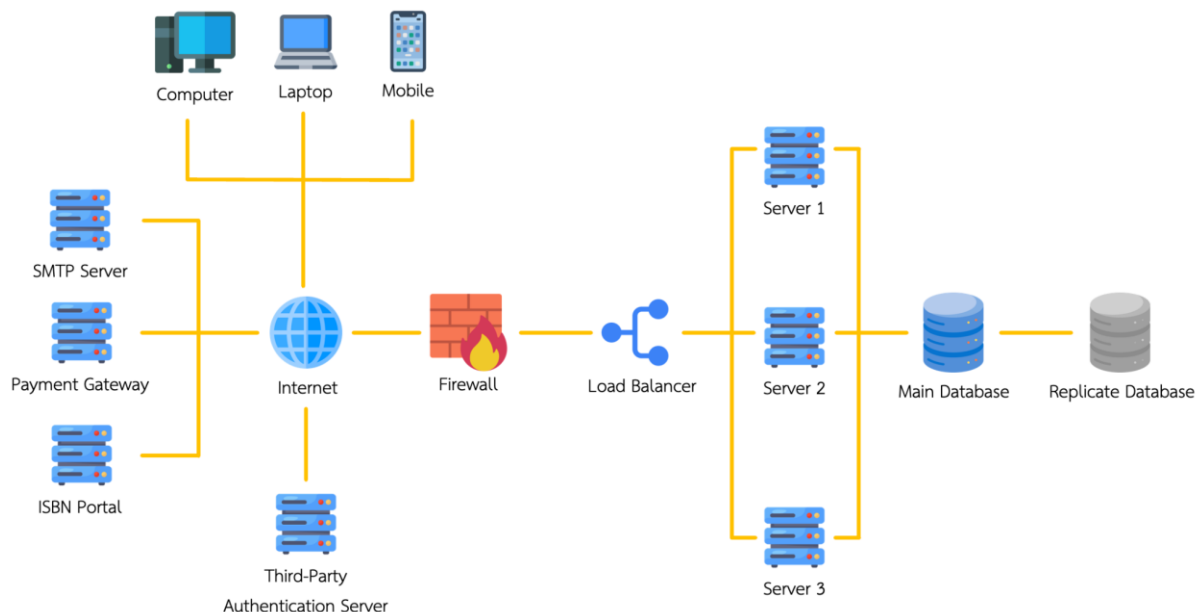


Figure 37 Network Diagram

https://chula-my.sharepoint.com/:p/g/personal/6530363421_student_chula_ac_th/EUKYWiVisbVPqldlksZGm7IBecXKggWGwePjoGTguKyTvA?e=k5nuwq&nav=eyJzSWQjI2NSwiY0lkIjo3OTczODg3NjJ9

The client, Payment Gateway, SMTP, ISSN, and Book Book system are interconnected via the internet.

Book Book system employs a Three-Tiered Architecture (Client-Server) that incorporates balance processing and boasts high scalability. The system is fortified with a fire wall and a load balancer to enhance security and avert server overloads. The load balancer efficiently distributes incoming requests among cloud servers, ensuring scalability as needed. In terms of databases, there is a main database and a backup replica database, which can be employed in case of main database failure, ensuring data integrity and reliability.

Deployment Diagram

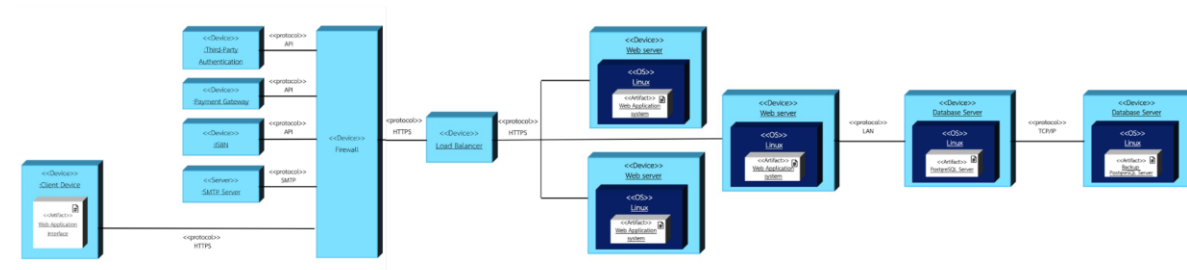


Figure 38 Deployment Diagram

https://chula-my.sharepoint.com/:p:/g/personal/6530363421_student_chula_ac_th/EUKYWiVisbVPqldlksZGm7IBecXKggWGwePjoGTguKyTvA?e=aoinpB&nav=eyJzSWQiOiI4NywiY0lkIjo1NDA5OTgzMDR9

Deployment diagram has 12 main nodes. Showing inner working detail extending the network diagram. External devices include 3rd party authentication, payment gateway, ISSN system, and user's devices. After the firewall and load balancer, 3 redundant linux web servers are available to handle incoming traffic. They're subsequently connected to postgresql databases, one is the main, and another one is replica for backup.

Hardware and Software Specification

From the deployment diagram, we used a 3-tier architecture which can be divided into 3 sections which are client, web server, and database server. The table below provides the minimum requirements of hardware and software specification that allows the system to function without a crash and for the interaction with the application to be used smoothly. The table has been created with many factors considered, such as hardware performance, cost of maintenance, etc.

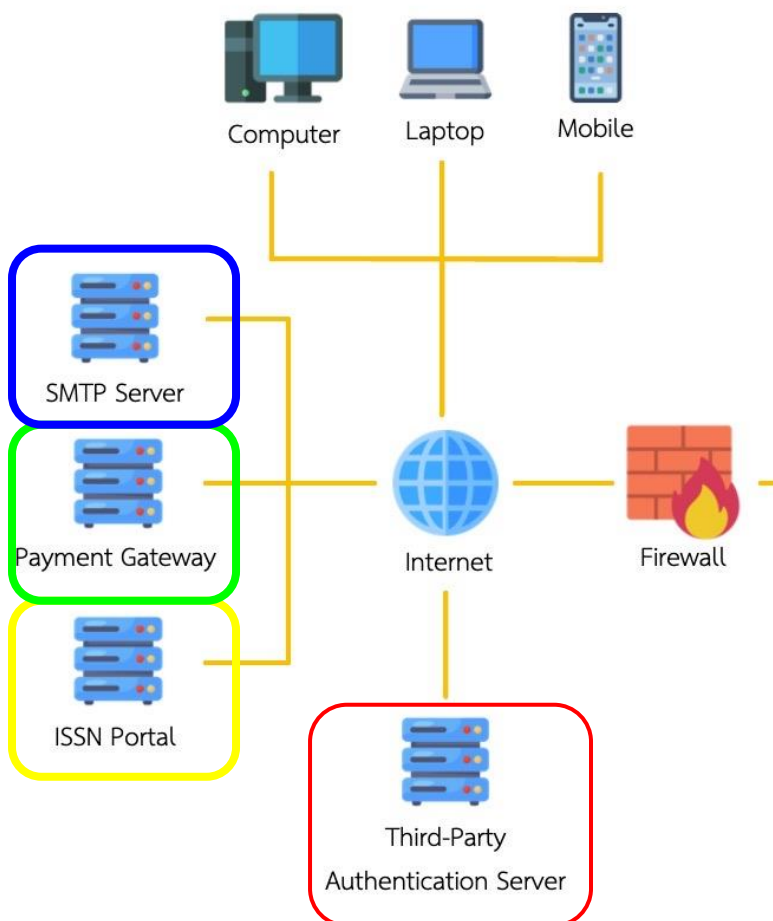
Category / Components	Client tier	Application tier	Data tier
Operating System	<ul style="list-style-type: none"> - Windows - MacOS - Linux - Android - IOS 	<ul style="list-style-type: none"> - Linux 	<ul style="list-style-type: none"> - Linux
Software	<ul style="list-style-type: none"> - Google chrome - Microsoft edge - Firefox - Safari - Opera 	<ul style="list-style-type: none"> - Apache HTTP server - Node.js? 	<ul style="list-style-type: none"> - PostgreSQL
Hardware	<ul style="list-style-type: none"> - Dual-core CPU - 4 GBs RAM 	<ul style="list-style-type: none"> - Quad-core CPU - 32 GBs RAM - 1 TB storage size 	<ul style="list-style-type: none"> - Quad-core CPU - 128 GBs RAM - 4 TBs storage size - 8 TBs backup storage size
Network	<ul style="list-style-type: none"> - Ethernet - WiFi - Cellular network 	<ul style="list-style-type: none"> - High-speed Ethernet 	<ul style="list-style-type: none"> - High-speed Ethernet

Table 4 Hardware and Software Specifications

Verifying and Validating the Physical Architecture Layer

Network diagram and deployment diagram

From the network diagram and deployment diagram, it can be seen that the parts connected to the system consist of five main components: the client, the payment gateway, and the third-party authentication server ,the ISSN portal and the SMTP server.



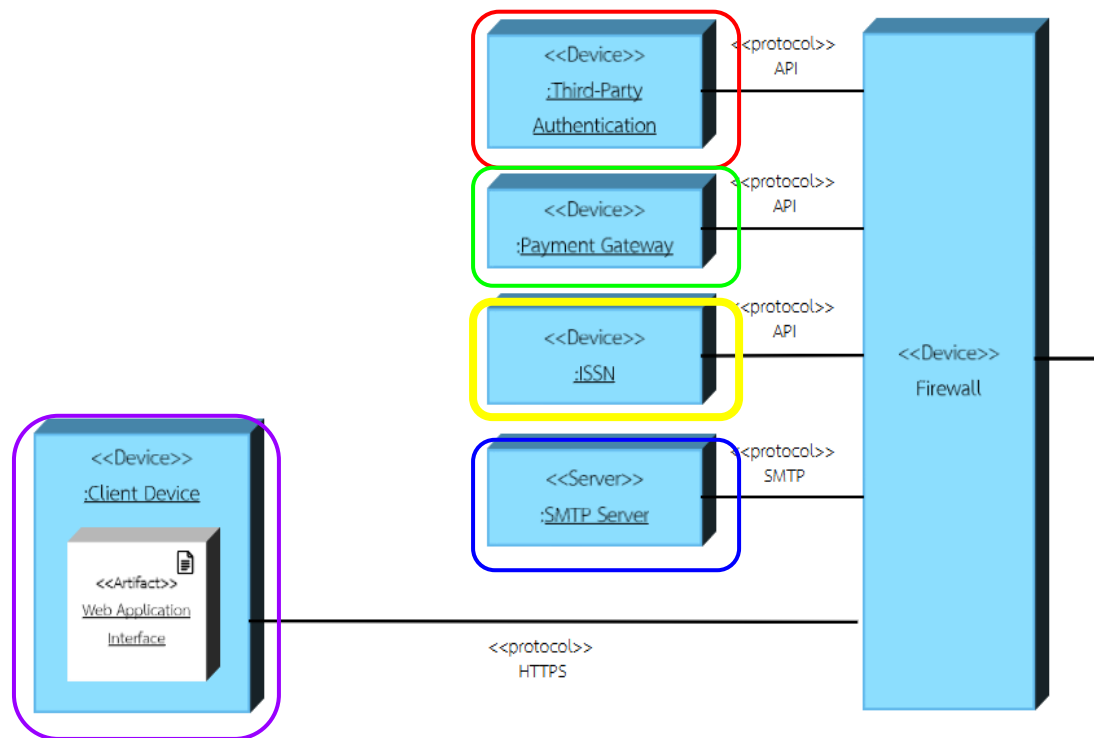


Figure 39 Validating Between Deployment Diagram and Network Diagram

On the internal side, the system consists of the load balancer, servers, the main database, and the replicated database, as shown in the network diagram and deployment diagram.

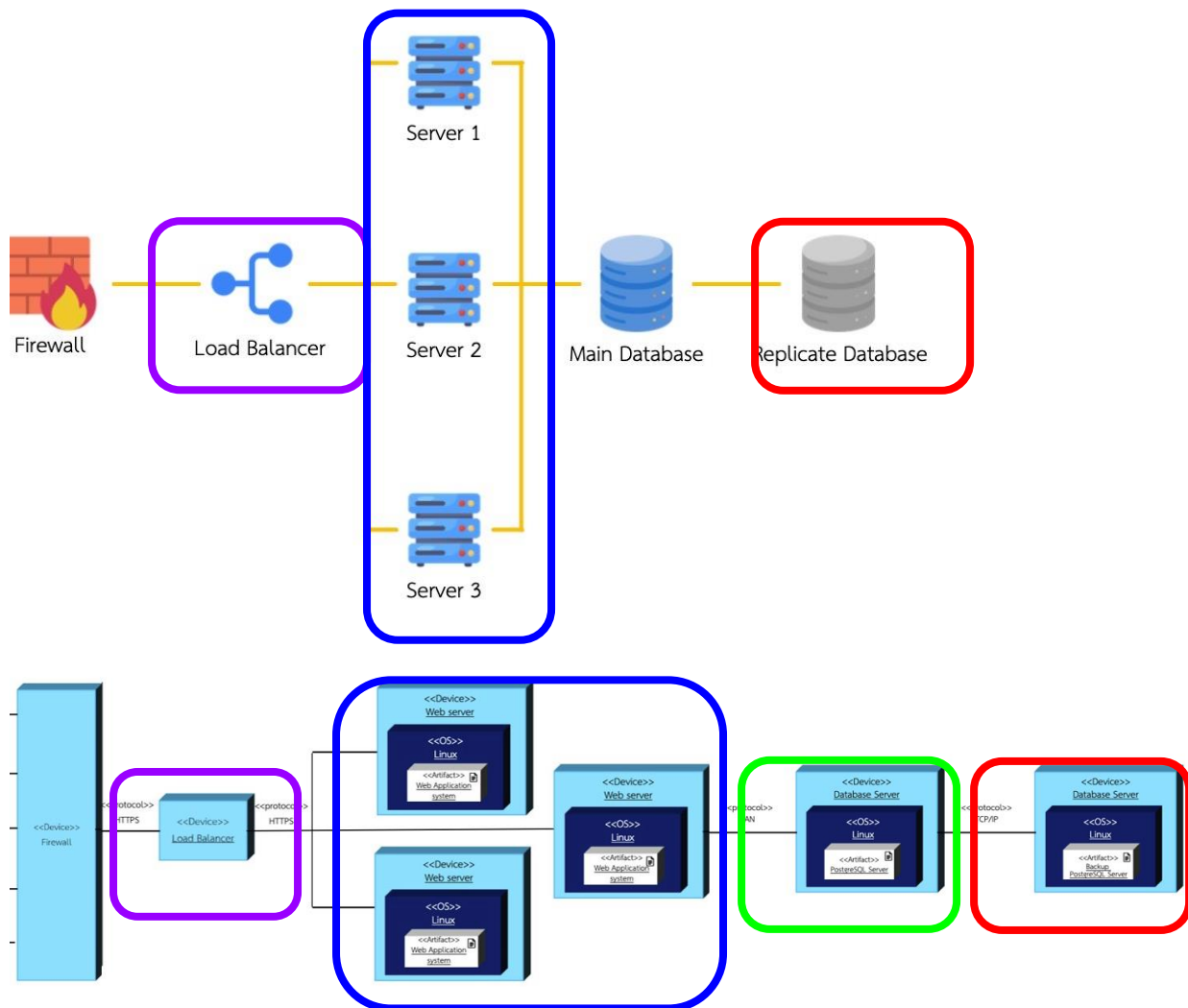


Figure 40 Validating Between Deployment Diagram and Network Diagram

Hardware and Software Specification

Category / Components	Client tier	Application tier	Data tier
Operating System	<ul style="list-style-type: none"> - Windows - MacOS - Linux - Andriod - IOS 	<ul style="list-style-type: none"> - Linux 	<ul style="list-style-type: none"> - Linux
Software	<ul style="list-style-type: none"> - Google chrome - Microsoft edge - Firefox - Safari - Opera 	<ul style="list-style-type: none"> - Apache HTTP server - Node.js? 	<ul style="list-style-type: none"> - PostgreSQL
Hardware	<ul style="list-style-type: none"> - Dual-core CPU - 4 GBs RAM 	<ul style="list-style-type: none"> - Quad-core CPU - 32 GBs RAM - 1 TB storage size 	<ul style="list-style-type: none"> - Quad-core CPU - 128 GBs RAM - 4 TBs storage size - 8 TBs backup storage size
Network	<ul style="list-style-type: none"> - Ethernet - WiFi - Cellular network 	<ul style="list-style-type: none"> - High-speed Ethernet 	<ul style="list-style-type: none"> - High-speed Ethernet

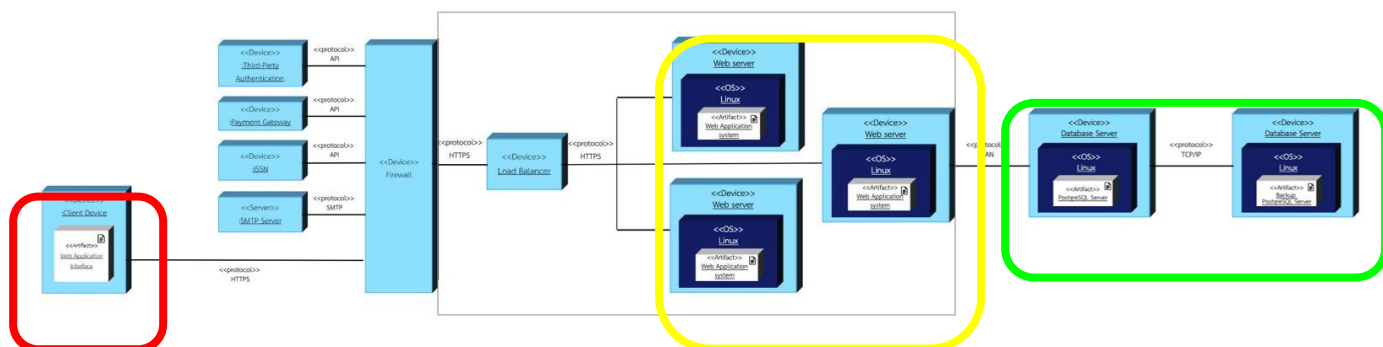


Figure 41 Validating Between Software Hardware Specification and Deployment Diagram

The Impact of the Analysis and Design Solution

Impact on global context

Positive

- The system will make a wide variety of books more accessible across Thailand, including rural areas with limited bookstores which led to growth in the second hand book market.
- The system might create more opportunities to share culture between countries because of a variety of second hand books from different countries.

Negative

- Could increase carbon emissions due to more shipping within the country.

Impact on the society context

Positive

- Connects buyers and sellers, building a community of readers.

Negative

- May result in fewer people visiting local bookstores.

Impact on the economic context

Positive

- Creates extra income for people who sell their used books.
- Boosts demand for shipping and logistics services in Thailand.

Negative

- Price competition may force sellers to lower prices, reducing profits.

Impact on the environment context

Positive

- Encourages book reuse, saving paper and reducing waste.

Negative

- Increased shipping within Thailand may lead to more packaging waste and higher carbon emissions.
- Data storage uses a lot of energy, which can raise carbon emissions.

Conclusion of the Design Process

In this document, we have reviewed the design criteria, system design modeling, and the system design constraint to verify the project's scope and to design the system properly.

We also constructed class diagrams and a package diagram on top of the class diagrams. We also constructed component diagrams to illustrate the relations between the components. These will serve as a rough layout of how the system works.

Then, to give finer details of the methods in the system, we created method specifications, two of which were shown in this report. We also verified class and method design later to ensure consistency in the diagrams we have created.

Following the method specifications, a detailed real use case description demonstrates the details and the flow of events. We then design the user interface of the flow of events. We follow the user interface design principles mentioned in this report strictly to ensure high quality, user-friendly design.

Lastly, we visited the non-functional requirements to design a physical architecture model that satisfies all requirements of the system. With a final validation to ensure the integrity of the components and requirements of the system.

System Installation and Operation

Migration Plan

1. Conversion Strategy

a. Conversion Plan

Hardware Installation

Our system needed 6 servers as 3 servers for backend 2 for databases and 1 for load balancing and needed space with good air circulation, security and maintainer.

Software Installation

Since our project is a web application, users need only a web browser to use our platform but for servers there will be some preparation for suitable operating system and software environments.

Data conversion

Our system is a new system and operates independently of any data from the existing system. Consequently, there will be no need for any data conversion.

b. Conversion Dimension

Conversion Style

Direct conversion style is chosen because Book Book is a new platform and is relatively new to the market. Therefore, there is no need to compare the system to the others.

Conversion Location

Pilot conversion is chosen for our application due to providing low risk to the system. One location is converted first for easier moderation. Once the system passes the test, the application will be installed at the remaining locations using phased conversion. This process takes a longer time to perform, but ensures the system's functionality.

Conversion Module

Whole-System conversion is chosen for our application because the to-be system is entirely separated from the as-is system. Full deployment is necessary for the proper functioning of the entire system. We believe that this approach will not only improve user experience but also contribute to a clearer understanding of the system, as opposed to modular conversion.

2. Change Management Plan

a. Revising Management Policy

The development team might need to hire new employees or relocate some personnels to work as the admin of the system, as well as the help desk and support staff.

b. Assessing Costs and Benefits

The cost and benefits assessment is performed on two perspectives. First, for the organization, a cost-benefit analysis is performed on both the as-is system and the to-be system in order to compare their benefits, risks and feasibility. Second, from the potential adopters' perspective, we will analyze personal cost-benefits for adopters from all levels.

c. Motivating Adoption

1) Informational Strategy

Our strategy is presenting data and analytics. This will highlight the benefits of our to-be system and how it can solve

the problems from the as-is system. Hosting live demonstrations can also help users to see the system in real-time action for clearer understanding.

2) Political Strategy

Our political strategy focuses on emphasizing the importance of digital sovereignty and the ability to control the digital infrastructure of second-hand book business. This will make users see the impacts and benefits of our system, and provide strong motivation to adopt the new system. Another possible strategy is to implement policies that encourage or mandate the use of the software.

d. Conduct Training

To ensure clear understanding of software and system operation, a comprehensive classroom training will be held to educate users. The training will include user interface navigation to perform basic tasks, and advanced features, including assessment of external events, user reports evaluation, and security measures. Hands-on practice will also be included with interactive exercises to reinforce learning, discussion and collaboration between users.

Post-implementation Activities

System Support

1. The system provides comprehensive online support, featuring detailed documentation and frequently asked questions (FAQs), via email address or in web pages, offering users accessible resources for troubleshooting and guidance. Also the system offers tutorials guiding users through website usage for enhanced ease-of-use.
2. Our Help Desk offers expert guidance across three levels: Level 1 support for general assistance, Level 2 support by experienced professionals where users can report specific issues, and Level 3 for cases where identified problems transition into bugs, facilitating change requests for resolution.

System Maintenance

1. Weekly team gatherings to evaluate the website's performance collectively.
2. Monthly feedback sessions involving users and managers to gather insights and suggestions for improvement.
3. Assessing the effectiveness of the application in solving problems and meeting business needs.
4. Tracking change requests to identify areas for application enhancement.

5. The team will deliberate on potential new features for the application and strategize ways to enhance or modify the existing features.
6. We will collect data from the first day until the end of 3 months, then we will develop a new version that meets your needs and fixes any errors that may occur.

Project Assessment

Project Team Review

1. Upon the immediate installation of the system, a thorough project team review will be conducted.
2. Team members will engage in collaborative reflection, revisiting and discussing individual and collective learnings from the project, acknowledging both mistakes and successful strategies.
3. The project manager will assume the responsibility of synthesizing and compiling the insights generated during the discussion into a comprehensive “lessons learned” document.
4. The project team review serves as a valuable tool not only for individual improvement but also contributes to the overall proficiency of the team, enhancing their collaborative effectiveness in future projects.

System Review

1. Post the system installation’s completion, comprehensive reviews are slated at the 3-month and 6-month milestones.
2. These assessments seek a meticulous comparison between actual costs and benefits against the earlier estimated figures from the feasibility analysis.
3. The findings from this comparative analysis will inform the calculation of baseline costs for upcoming projects, ensuring more accurate projections and informed decision-making.

List of Teammate Contributions

Name	Responsibility	Level of Achievement
Peneik Sitthimongkhon (Faro)	<ul style="list-style-type: none"> Reference Documents User interface Design Formatting 	3
Yuatyong Chaichana (Yuat)	<ul style="list-style-type: none"> System Design Constraints Component Diagram Hardware and software specification 	3
Puvamin Tanakittanon (Peet)	<ul style="list-style-type: none"> Objective of design document Overview of system modeling. Use Case Diagram 	3
Peemdej Chancharoen (Peem)	<ul style="list-style-type: none"> User interface design principles NFR on user interface design Real usecase description 	3
Tapaneeya Odmung (Nack)	<ul style="list-style-type: none"> Method Specification Verifying and Validating Non-functional Requirements on Physical Architecture Design 	3

Name	Responsibility	Level of Achievement
Pranesh Ingkanunt (Pooh)	<ul style="list-style-type: none"> • Design Criteria • Design Activities • Class Diagram 	3
Wasutha Sukkrasanti (Book)	<ul style="list-style-type: none"> • To-be System • Overview of the To-be System • Overview of Infrastructure Design 	3
Naphon Tangtrongpairroj (Mon)	<ul style="list-style-type: none"> • To-be System • Project Objective • Constrains 	3
Nutthapong Dissanont (Tontoey)	<ul style="list-style-type: none"> • As-is System • Introduction • Non-functional Requirements 	3