BoiChokro

A Hand Book Community

Software Engineering and Information System Design Lab
Course Code: CSE - 3104
Submitted by: Team OrbitBEE

Submitted by:

Mahmudul Hasan Noman (22CSE001)
Lazmi Rahman Ayman (22CSE004)
Redwanul Haque Rafe (22CSE030)
Tanim Ahmed (22CSE032)
Amin Bhuiyan (22CSE046)
Most. Maliha Akter (22CSE049)

Supervised by:

Md Samsuddoha
Assistant Professor
Department of Computer Science and Engineering
University of Barishal

Submission Date: 19-01-2026

Team Contribution

Team Member	Member Contribution
Mohammad Redwanul Haque Rafe (22CSE030)	Inception, Elicitation & Use-Case Diagram
Most. Maliha Akter (22CSE049)	Activity & Swimlane Diagram
Mahmudul Hasan Noman (22CSE001)	Use Case Scenario, ER Diagram & Schema
Lazmi Rahman Ayman (22CSE004)	Class Based Modeling
Amin Bhuiyan (22CSE046)	State Transition & Write Full Documentation
Tanim Ahmed (22CSE032)	Data Flow Diagram & Sequence Diagram

Table of Contents

BoiChokro - A Community System in Hand Book

BoiChokro is an open-source web-based community-based project that develops a local and sustainable ecosystem to share physical books. Some people appreciate the experience of reading hardcopy books. Nevertheless, it is becoming expensive to acquire new books and there is no arranged local market where a used book can be acquired hence making it hard to access these books by the students and book lovers. BoiChokro attempts to solve this issue by allowing users to purchase, sell, swap, and even donate books in a particular local area, community or university campus, which avoids the expense of delivering them and makes them reusable. Role-based access control registers a user, libraries, and administrators in the BoiChokro system. Every user is registered with secure authentication schemes. Role-based access system guarantees that the users, libraries and administrators will access only features they are supposed to in their line of duty.

Once the user has logged into the system, he or she is able to proceed to search the books based on the area and also based on categories. The system shows the information about the books such as the availability status, condition, type of listing (sale, swap, or donation) and location. When a user wants a certain book but it is not readily available, the user can put it on his/her wish list. The system keeps track of book listings continuously and gets notifications automatically whenever a matching book is available in the same area. Any user may post their own books either to sell or swap. The administrator can approve or reject all the listings and donation requests posted recently before they become visible to the community.
It also provides a library module which gives the ability to add libraries to your platform, either a public or university one. The libraries are able to deny or accept user request to book certain books, keep track of the lending records and to maintain library patrons. On what a user requests a library book, the library will check on the request and by the time of approval issues the book, the system will update automatically to the status and lending records.

BoiChokro will have an inbuilt transaction and payment system to facilitate safe book purchase. Upon making a purchase, the system will make payment using an inbuilt payment gateway and update the transaction status of the user. In the case of swaps and donations, this system monitors the completion of handover without any monetary flow. In order to make the community more involved, BoiChokro offers a community discussion platform where users are allowed to leave book reviews, book recommendations, and book requests with priority tags like NeedBook. Users have an opportunity to like, comment, and filter posts, and report inappropriate content. Moreover, a chat module is also available in real time.

BoiChokro has a special characteristic of an environmental impact dashboard, which encourages the idea of sustainability. The system monitors reused and donated books and reuses environmental benefits including paper saved and trees saved as a result of book reuse. Such statistics are introduced in the form of visual dashboards, which encourage users to make more active contributions to protecting the environment. Administrators are at the main center of upkeep of the platform. They administer users and libraries, access control, accept new content, filter community interactions, and create system reports.

In general, BoiChokro integrates the practice of sharing the books, social engagement, and integrating the library into the community, and environmental conservation using one platform. The system would provide access to and affordability of reading, as well as community building (nothing like community bonding like sharing a book) and environmental sustainability (one book at a time).

1. Inception

1.1 Introduction:

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project.

To establish the ground work we have worked with the following factors related to the inception phases:

1. Identifying Stakeholders

2. Recognizing multiple viewpoints

3. Working towards collaboration

4. Asking the First Questions

1.2 Identifying Stakeholders:

A stakeholder refers to any individual or group who is directly or indirectly affected by the development, deployment, and operation of the system. Stakeholders include users who interact with the system regularly, as well as organizations and individuals who influence system decisions, maintenance and policy enforcement.

To identify the stakeholders of the BoiChokro – A Hand Book Community system, we analyzed the objectives of the platform and considered the following questions:

a. Who will use the system regularly?
b. Who will manage and control system operations?
c. Who will benefit from the outcomes of the system?
d. Who will support financial transactions and technical maintenance?
e. Whose activities will be affected after system deployment?

1. **End Users (Readers / Students / Book Lovers):**
   End users are the primary stakeholders of the system. They directly interact with BoiChokro to buy, sell, swap, and donate books within their local community. They also participate in community discussions, use wishlist and alert features, and view environmental impact statistics.

2. **Library Authorities (Public Libraries, University Libraries):**
   Library authorities use the system to register libraries, manage available books, approve user bookings, maintain lending history, and manage library members. They play a key role in ensuring book availability and organized circulation.

3. **System Administrator:**
   The system administrator is responsible for managing the overall platform. This stakeholder controls user roles, approves book listings and donations, moderates community content, manages access control, and generates system-wide reports.

4. **Donors (Individuals and Institutions):**
   Donors contribute books to the platform for donation purposes. They are interested in a transparent and efficient donation process and expect to see the social and environmental impact of their contributions.

5. **Educational Institutions (Schools and Community Libraries):**
   Educational institutions receive donated books through the system. They benefit from easy coordination with donors and libraries and rely on the system for fair and organized book distribution.

6. **Payment Service Provider:**
   The payment service provider facilitates secure online transactions for book purchases. This stakeholder ensures payment integrity, transaction security, and smooth integration with the platform.

7. **Development and Maintenance Team:**
   Developers are responsible for designing, implementing, maintaining, and upgrading the system. They handle technical issues, system updates, and ensure long-term scalability and reliability.

1.3 Recognizing Multiple Viewpoints:

Each stakeholder group has different expectations and concerns regarding the BoiChokro system. Recognizing these viewpoints helps in resolving conflicts and aligning system requirements with stakeholder needs.

1.3.1 End User Viewpoints

a. Affordable access to physical books within nearby locations.
b. Easy options to buy, sell, swap, and donate books.
c. Automated wishlist alerts for desired books.
d. Safe and interactive community space for discussions and recommendations.
e. Awareness of environmental impact through visual dashboards.

1.3.2 Library Authority Viewpoints

a. Ability to register libraries and manage book inventories.
b. Control over approving user bookings.
c. Accurate maintenance of lending history.
d. Increased visibility through geo-location based search.
e. Access to analytics on book usage and demand.

1.3.3 System Administrator Viewpoints

a. Full administrative control over users, libraries, and book listings.
b. Approval authority for new listings and donations.
c. Tools for moderating community posts and chat content.
d. Generation of reports and performance analytics.
e. Enforcement of role-based access control.

1.3.4 Donor Viewpoints

a. Simple and reliable book donation process.
b. Transparency regarding where donated books are distributed.
c. Clear visibility of social and environmental contributions.

1.3.5 Educational Institution Viewpoints

a. Easy receipt of donated books for students and readers.
b. Smooth coordination with donors, libraries, and administrators.
c. Fair and organized distribution of books.

1.3.6 Payment Service Provider Viewpoints

a. Secure and reliable payment processing.
b. Smooth integration of payment APIs.
c. Protection against transaction failures and fraud.

1.3.7 Developer’s viewpoints:

a. Easy to develop.
b. No ambiguous requirement.

1.4 Working Towards Collaboration:

Since stakeholders may have overlapping or conflicting requirements, collaboration is necessary to reach a balanced and practical solution. To achieve this, the following steps were followed:

1. Identification of common and conflicting requirements.
2. Categorization of requirements based on priority.
3. Evaluation of feasibility and system constraints.
4. Finalization of requirements through consensus.

Common Requirements

a. User-friendly and efficient system.
b. Web-based access.
c. Secure authentication and role-based access.
d. Easy book discovery and management.

Conflicting Requirements

a. Maximum feature set vs. limited budget.
b. Open access vs. controlled access.
c. Advanced features vs. system simplicity.

After analysis and prioritization, the final requirements were agreed upon.

Final Requirements

a. Secure user registration with email OTP verification.
b. Role-based access control for users, libraries, and administrators.
c. Book posting for sale, swap, and donation.

d. Approval mechanism for listings and donations.
e. Area-based and category-based book search.
f. Wishlist and automated availability alerts.
g. Library registration and lending management.
h. Community discussion and chat system.
i. Integrated payment system.
j. Environmental impact tracking and visualization.
k. Administrative reporting and moderation.

Asking the First Questions

During the inception phase, a set of context-free questions was used to understand the project goals, stakeholder expectations, and system constraints. These questions helped clarify:

a. The core problem to be solved.
b. Expected benefits of the system.
c. Possible alternatives to the proposed solution.
d. Stakeholder perceptions of success.
e. Communication effectiveness among stakeholders.

The answers to these questions guided the requirement analysis and shaped the overall system vision.

Conclusion

The inception phase of the BoiChokro project established a clear understanding of the system's purpose, scope, and stakeholders. It helped identify key requirements, resolve conflicting viewpoints, and define a shared vision for the platform. This phase laid a strong foundation for subsequent requirement analysis, design, and development activities, ensuring that BoiChokro effectively addresses community needs, promotes sustainable book reuse, and strengthens local engagement.

2. Elicitation

2.1 Introduction

Elicitation is a task that helps stakeholders define what is required from a software system. During the elicitation phase, several challenges may arise, such as problems of scope, problems of requirement volatility, and problems of understanding among stakeholders. Identifying and managing these issues at an early stage is essential to ensure that the system meets real user needs. However, elicitation is not an easy task. To overcome these challenges, we conducted the requirements elicitation activities for the BoiChokro (A Hand Book Community) system in a structured, organized, and systematic manner. This approach helped us clearly understand stakeholder expectations and transform them into well-defined system requirements**.**

2.2 Eliciting Requirements

Unlike the inception phase, where a Question and Answer (Q&A) approach is primarily used, the elicitation phase employs a requirements elicitation format that combines elements of problem solving, elaboration, negotiation, and specification. This process requires close cooperation between end users, system administrators, and developers to successfully elicit accurate and complete requirements.

To elicit requirements for the BoiChokro system, we carried out the following four activities:

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenarios
4. Elicitation Work Products

2.3 Collaborative Requirements Gathering

Various approaches to collaborative requirements gathering have been proposed, each utilizing different interaction scenarios. For the BoiChokro project, we followed a collaborative approach involving discussions and analysis with potential stakeholders.

The following steps were completed during the collaborative requirements gathering process:

a. Meetings and informal discussions were conducted with potential end users such as students and book readers to understand their expectations from a local book-sharing platform.
b. Library representatives were consulted to identify their requirements regarding book management, user bookings, lending history, and library membership management.
c. Discussions were held regarding the challenges faced in acquiring affordable physical books and managing unused books within local communities.
d. Feedback was collected on community interaction features, donation processes, and environmental impact awareness.
e. Finally, based on these discussions and observations, the final requirement list for the Boichokro system was selected and refined.

2.4 Quality Function Development

Normal Requirements

These are essential system functionalities. Absence of these would make the system unacceptable.

a. User, Library, and Admin Core Management.
b. Create and update user personal information.
c. Secure password hashing.
d. Role-based access control (Admin, Library, User).
e. Email OTP verification and password recovery.
f. Book & Library Management.
g. Post books for sale, swap, or donation.
h. Register libraries and available books.
i. Manage book listings and availability.
j. Approve new listings and donations.
k. Manage all user, book, and library data.
l. Moderate community posts.

m. Manage access control.
n. Communication & Notifications.

Expected Requirements

These improve user satisfaction proportionally—the better they work, the happier the users are.

a. User Experience & Transactions.
b. View transaction history.
c. Maintain wishlist.
d. Automated alerts for book availability.
e. Track reused and donated books.
f. Review and rating system.
g. Library Operations.
h. Approve user bookings.
i. Geo-location search for books and libraries.
j. Maintain lending history.
k. Manage library membership.
l. Community Interaction.
m. Participate in community discussions.
n. Post requests for books.
o. Like, comment, and filter community posts.
p. Report inappropriate content.
q. Generate reports (Admin).
r. Message alerts in chat module.

Exciting Requirements

These are not expected by users but significantly enhance satisfaction and system uniqueness.

a. Environmental Impact Features.
b. View environmental impact statistics.
c. Calculate paper and trees saved.
d. Visual dashboards showing contribution impact.
e. Integrated payment system.

2.5 User Stories

2.5.1 User Stories – End User

1. **As a user**, I want to create an account with email verification, so that my account remains secure.
2. **As a user**, I want to buy used books within my locality, so that I can avoid delivery costs.
3. **As a user**, I want to sell my old books I no longer need, so that others can benefit from them.
4. **As a user**, I want to add books to my wishlist, so that I get notified when they become available.
5. **As a user**, I want to chat with other users, so that I can negotiate or discuss books easily.
6. **As a user**, I want to post or see book reviews and recommendations, so that I can help other readers and get help from others.

2.5.2 User Stories – Library

1. **As a library**, I want to register and list available books, so that users can find them easily.
2. **As a library**, I want to approve or reject user bookings, so that books are properly managed.
3. **As a library**, I want to maintain lending history, so that records remain transparent.
4. **As a library**, I want analytics on book usage, so that I can improve library services.

2.5.3 User Stories – Administrator

1. **As an admin**, I want to manage users and roles, so that access control is enforced.
2. **As an admin**, I want to approve book listings and donations, so that platform quality is maintained.

3. **As an admin**, I want to moderate community posts, so that inappropriate content is removed.
4. **As an admin**, I want to generate system reports, so that performance and usage can be analyzed.

2.5.4 User Stories – Donor

1. **As a donor**, I want to donate books to schools and libraries, so that underprivileged students benefit.
2. **As a donor**, I want to see impact statistics, so that I feel motivated to donate more.

2.5.5 User Stories – Payment System

1. **As a payment system**, I want to securely process payments, so that transactions are safe and reliable.

3. Scenario Based Modeling

3.1 Use Case Scenarios

Level–0 (System)	Level–1 (Module)	Level–2 (Use Case)	Actors
Book Reuse & Sharing System	Authentication	Sign Up	User, Library
		Email OTP Verification	User, Library
		Sign In	User, Library, Admin
		Sign Out	User, Library, Admin
		Change Password	User, Library, Admin
		Password Recovery	User, Library

	User Management	Create Personal Profile	User
		Update Personal Information	User
		Role-Based Access Control	Admin
	Book Management	Post Book (Sell / Swap / Donate)	User
		Upload Book Details	User
		Edit Book Listing	User, Library
		Delete Book Listing	User, Library
		Approve Book Listings	Admin
	Library Management	Register Library	Library
		Manage Available Books	Library
		Approve User Bookings	Library
		Maintain Lending History	Library
	Search & Transactions	Search Books	User, Library
		Filter Books (Category / Location)	User
		Geo-location Based Search	User
		Request Book	User
	Wishlist & Alerts	Transaction History	User
		Add Book to Wishlist	User
		Book Availability Alerts	User

	Community Module	Post Community Discussion	User
		Request Books	User
		Like / Comment on Posts	User
		Filter Community Posts	User
		Report Inappropriate Content	User
		Moderate Community Posts	Admin
	Chat Module	Send Message	User
		Receive Message Alerts	User
		Real-Time Chat	User
		Chat Moderation	Admin
	Notifications	Dashboard Notifications	User, Library
		Email Notifications	User, Library
	Environmental Impact Dashboard	Track Reused Books	User, Admin
		Track Donated Books	User, Admin
		Calculate Paper Saved	System
		View Contribution Statistics	System
	Payments	Make Payment	User
		Payment Verification	User
	Administration	Manage Users	Payment Gateway
		Manage Libraries	Admin
		Manage Books	Admin
		Generate Reports	Admin
		Manage Access Control	Admin

3.2 Use Case Description

01. Authentication

i. Use Case: Sign Up

Primary Actors: User, Library

Goal in Context: To register a new user or library into the system.

Preconditions:

a. Enters System provides a registration interface.

b. User/Library is not already registered

Trigger: User/Library wants to access the system.

Main Scenario:

a. Actor opens the Sign Up page.

b. Required details (name, email, role, password).

c. System validates inputs.

d. System sends OTP to registered email.

e. Actor verifies OTP.

f. System confirms registration.

Exceptions:

a. Invalid OTP
b. email already registered.

Priority: Essential

When Available: First increment

ii. Use Case: Sign In

Primary Actors: User, Library, Admin

Goal in Context: To access the system.

Preconditions: Actor must be registered.

Trigger: Actor wants to log in.

Main Scenario:

a. Actor enters email and password.
b. System authenticates credentials.
c. Dashboard is displayed.

Exceptions:

a. Invalid credentials.
b. Blocked account.

Priority: Essential

When Available: First increment

iii. Use Case: Sign Out

Primary Actors: User, Library, Admin

Goal in Context: To securely exit the system.

Preconditions: Actor must be logged in.

Trigger: Actor selects Sign Out.

Main Scenario:

a. System ends session.
b. Redirects to home page.

Priority: Essential

iv. Use Case: Password Recovery / Change Password

Primary Actors: User, Library, Admin

Goal in Context: To reset or update password.

Preconditions: Actor has a registered email.

Trigger: Actor forgets or wants to change password.

Main Scenario:

a. Actor requests password reset/change.

b. OTP/link sent to email.

c. Actor sets new password.

d. System confirms update.

Exceptions:

a. Weak password,
b. Invalid OTP.

Priority: Essential

2. User Management

i. Use Case: Create Personal Profile

Primary Actors: User

Goal in Context: To create a personal profile.

Preconditions: User must be logged in.

Trigger: User accesses profile section.

Main Scenario:

a. User enters profile details.

b. System saves profile.

Priority: Expected

ii. Use Case: Role-Based Access Control

Primary Actors: Admin

Goal in Context: To manage permissions.

Preconditions: Admin logged in.

Main Scenario:

a. Admin selects user/library.
b. Assigns or updates role.
c. System updates access rights.

Priority: Essential

3. Book Management

i. Use Case: Post Book (Sell / Swap / Donate)

Primary Actors: User

Goal in Context: To list a book for reuse.

Preconditions: User logged in.

Main Scenario:

a. User enters book details and mode (sell/swap/donate).
b. Uploads images.
c. Submits listing.
d. Admin approves listing.

Exceptions: Invalid details.

Priority: Essential

ii. Use Case: Edit / Delete Book Listing

Primary Actors: User, Library

Goal in Context: To maintain accurate listings.

Preconditions: Listing exists.

Main Scenario:

a. Actor selects listing.

b. Updates or deletes details.

c. System confirms action.

Priority: Essential

4. Search & Transactions

i. Use Case: Search Books

Primary Actors: User, Library

Goal in Context: To find available books.

Preconditions: System has active listings.

Main Scenario:

a. Actor enters keywords or filters.

b. System displays results.

Exceptions: No results found.

Priority: Essential

ii. Use Case: Request Book

Primary Actors: User

Goal in Context: To request a listed book.

Preconditions: Book is available.

Main Scenario:

a. User sends request.

b. Owner/Library receives notification.

c. Request approved or rejected.

Priority: Essential

5. Community Module

i. Use Case: Post Community Discussion

Primary Actors: User

Goal in Context: To interact with the community.

Preconditions: User logged in.

Main Scenario:

a. User creates a post.
b. Others like/comment.
c. Admin moderates content if needed.

Priority: Expected

6. Chat Module

i. Use case: real-time chat

Primary Actors:User

Goal in Context: To communicate with other users.

Preconditions: Both users logged in.

Main Scenario:

a. User sends message.
b. Receiver gets notification.

Priority: Expected

7. Environmental Impact Dashboard

i. Use Case: Track Reused / Donated Books

Primary Actors: User, Admin

Goal in Context: To measure environmental impact.

Preconditions: Transactions completed.

Main Scenario:

a. System logs reused/donated books.

b. Calculates paper saved.

c. Displays statistics.

Priority: Expected

8. Payments & Administration

i. Use Case: Make Payment

Primary Actors: User

Goal in Context: To complete paid transactions.

Preconditions: Payment gateway available.

Main Scenario:

a. User selects payment option.

b. Payment verified.

c. Transaction recorded.

Priority: Essential

ii. Use Case: Manage Users / Libraries / Reports

Primary Actors: Admin

Goal in Context: To administer the system.

Preconditions: Admin logged in.

Main Scenario:

a. Admin views system data.

b. Manages users, libraries, books.

c. Generates reports.

Priority: Essential

4. Use Case Diagram

Level: 0
Name: *BoiChokro*
Primary Actor: User, Library, Administration
Secondary Actor: Database, Payment Gate way, Email

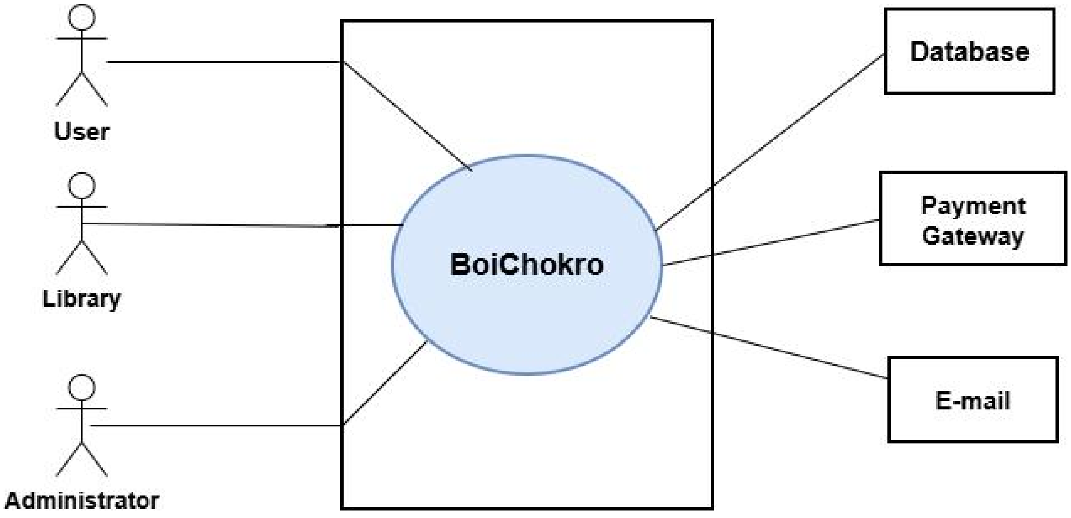


Figure: Level 0 - BoiChokro

Level: 1

Name: User-Book-Library management, Administration, Payment etc.

Primary Actor: User, Library/Shop, Admin

Secondary Actor: System

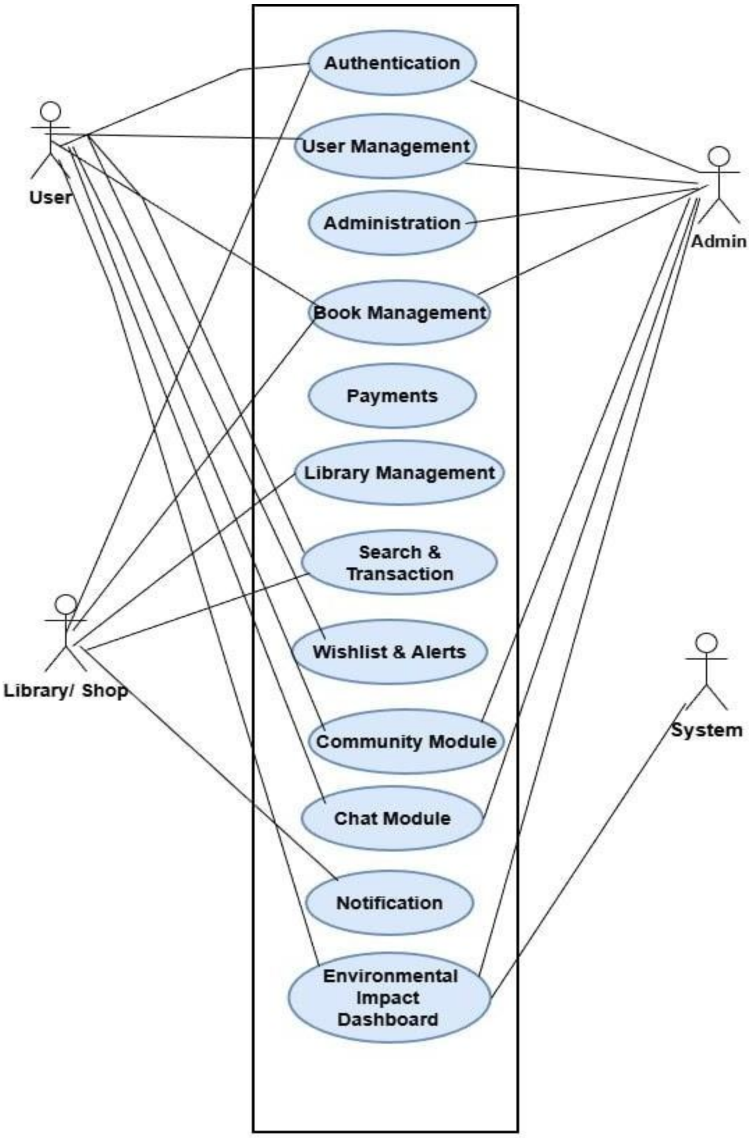


Figure: Level 1 - BoiChokro

Level: 2.1
Name: Authentication
Primary Actor: User, Library/Shop, Admin
Secondary Actor:

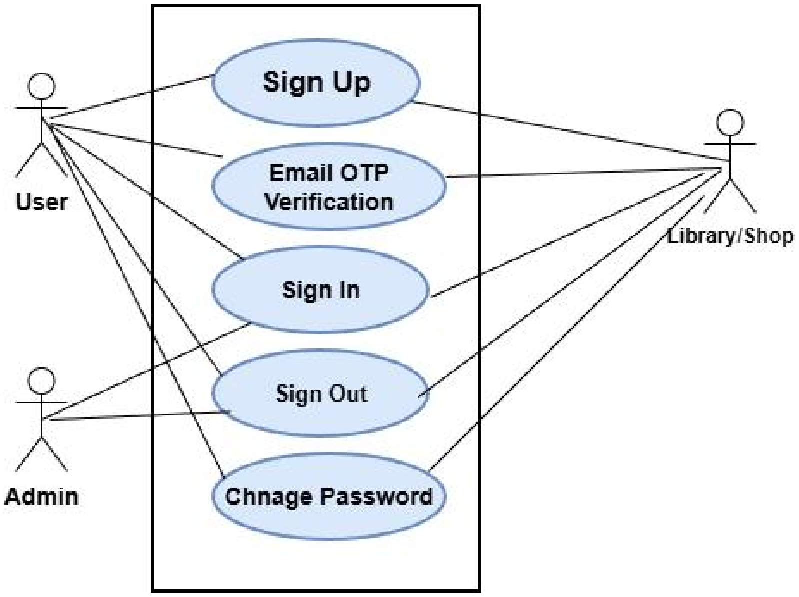


Figure: Level 2.1 - Authentication

Level: 2.2
Name: User management
Primary Actor: User, Admin

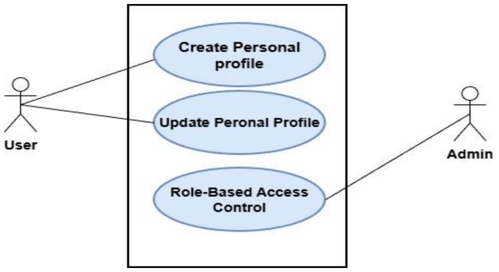


Figure: Level 2.2 - User Management

Level: 2.3
Name: Book management
Primary Actor: User, Library, Admin

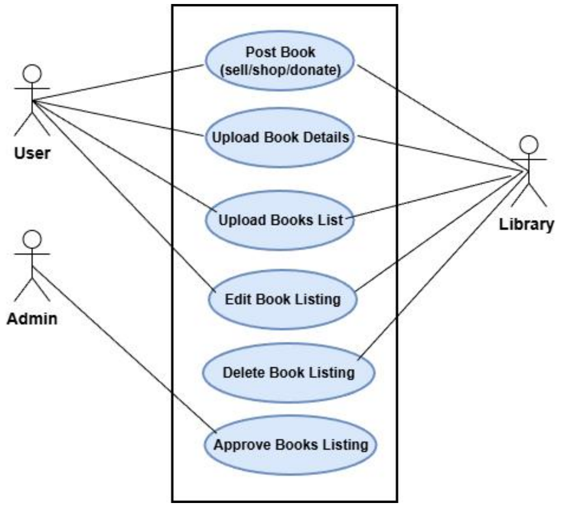


Figure: Level 2.3 - Book Management

Level: 2.4
Name: Library/Shop management
Primary Actor: Library/Shop

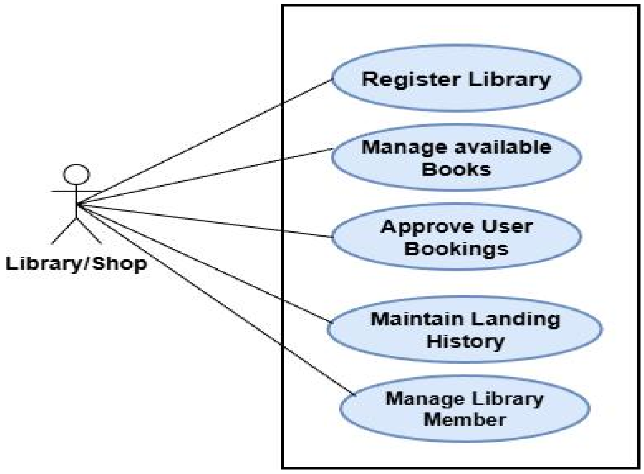


Figure: Level 2.4 - Library/Shop Management

Level: 2.5
Name: Search & transaction
Primary Actor: User, Library

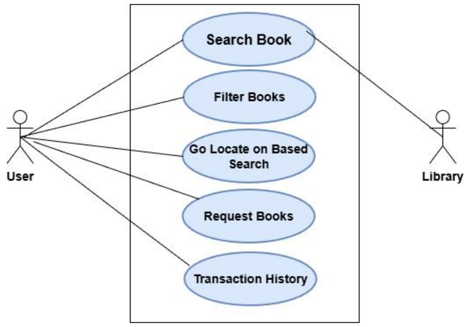


Figure: Level 2.5 - Search & Transaction

Level: 2.6
Name: Wishlist & Alerts
Primary Actor: User

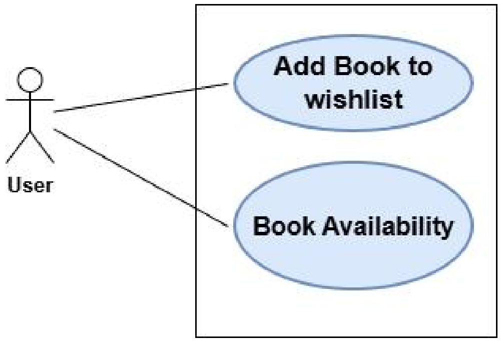


Figure: Level 2.6 - Wishlist &Alerts

Level: 2.7
Name: Community Module
Primary Actor: User, Admin

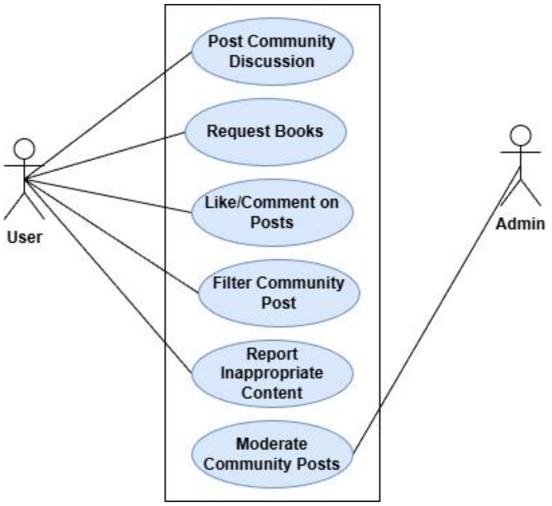


Figure: Level 2.7 - Community Model

Level: 2.8
Name: Chat Module
Primary Actor: User, Admin

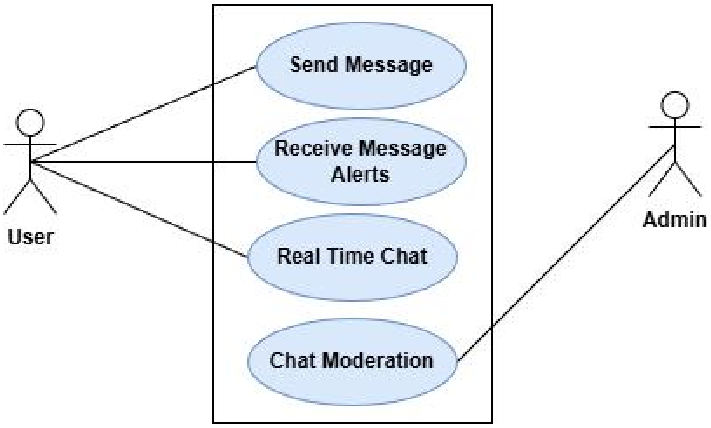


Figure: Level 2.8 - Chat Module

Level: 2.9
Name: Notification Module

Primary Actor: User, Library

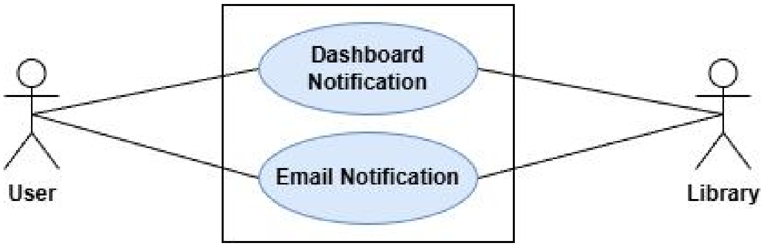


Figure: Level 2.9 - Notification Module

Level: 2.10
Name: Environmental Impact Dashboard
Primary Actor: User, Admin
Secondary Actor: System

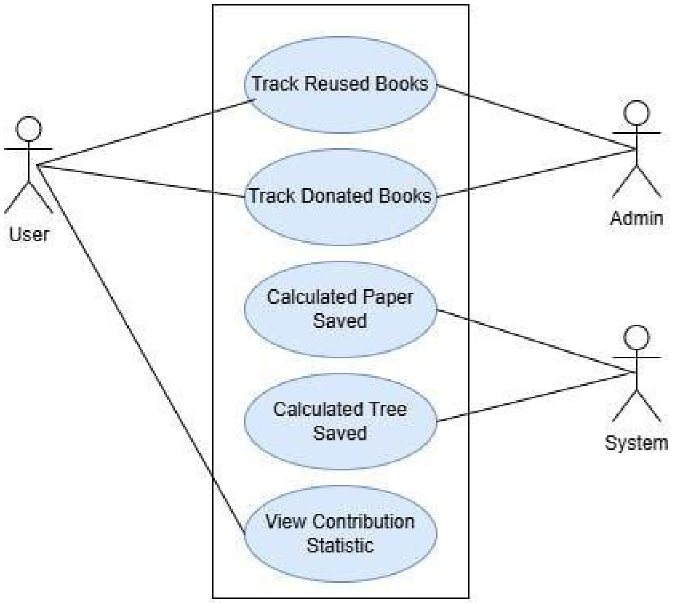


Figure: Level 2.10 - Environmental Impact Dashboard

Level: 2.11
Name: Payments
Primary Actor: User
Secondary Actor: Payment Gateway

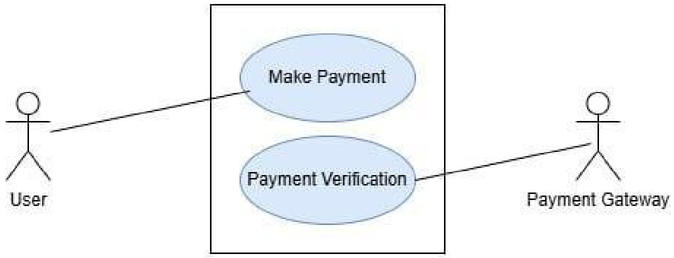


Figure: Level 2.11 - Payments

Level: 2.12
Name: Administration
Primary Actor: Admin

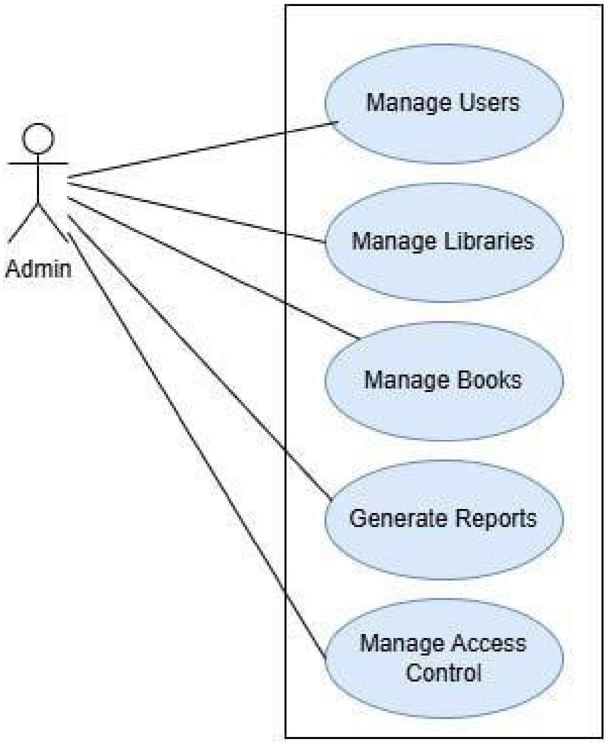


Figure: Level 2.12 - Administration

5. Activity & Swimlane Diagram

Use case 1: Authentication
Activity Diagram:

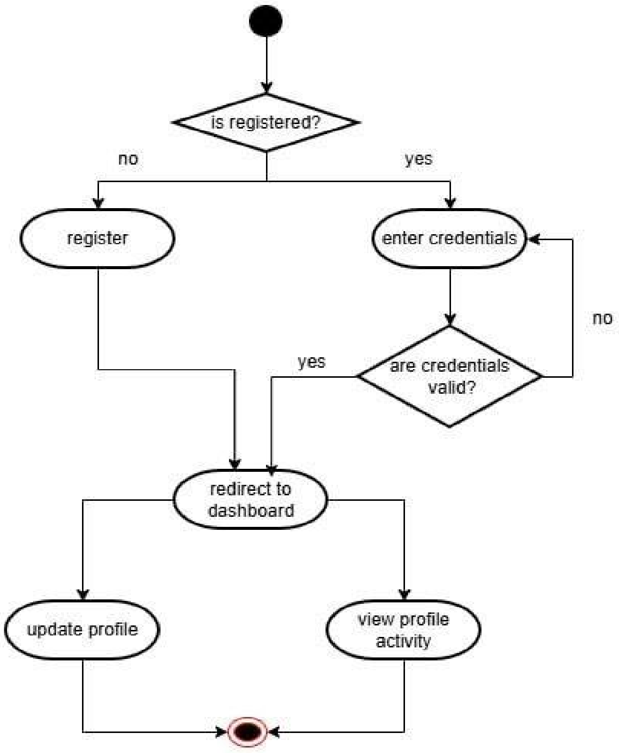


Figure: Authentication

Swimlane Diagram:

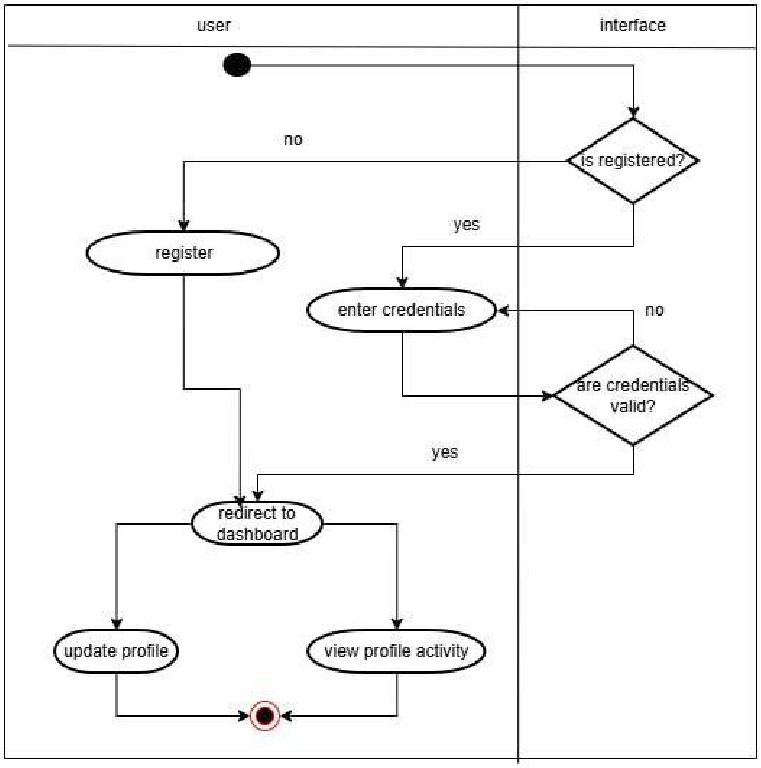


Figure: Authentication

The following diagram shows the user log-in process, which runs verification of registration and credential validity and allows access to the dashboard functionality, such as profile updates and activities.

Name: User Management

Reference: Use Case 2.2

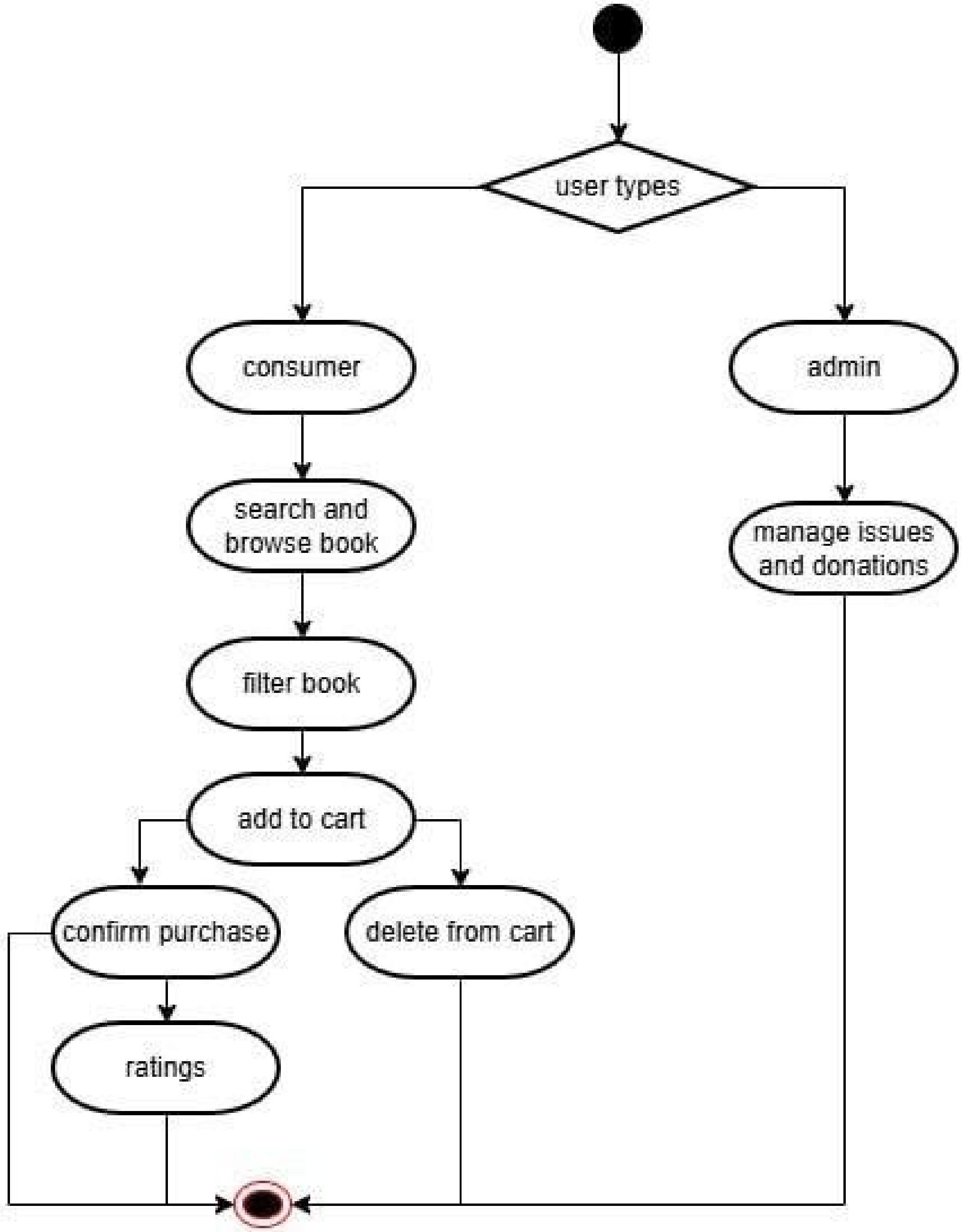


Figure: User Management

Swimlane Diagram:

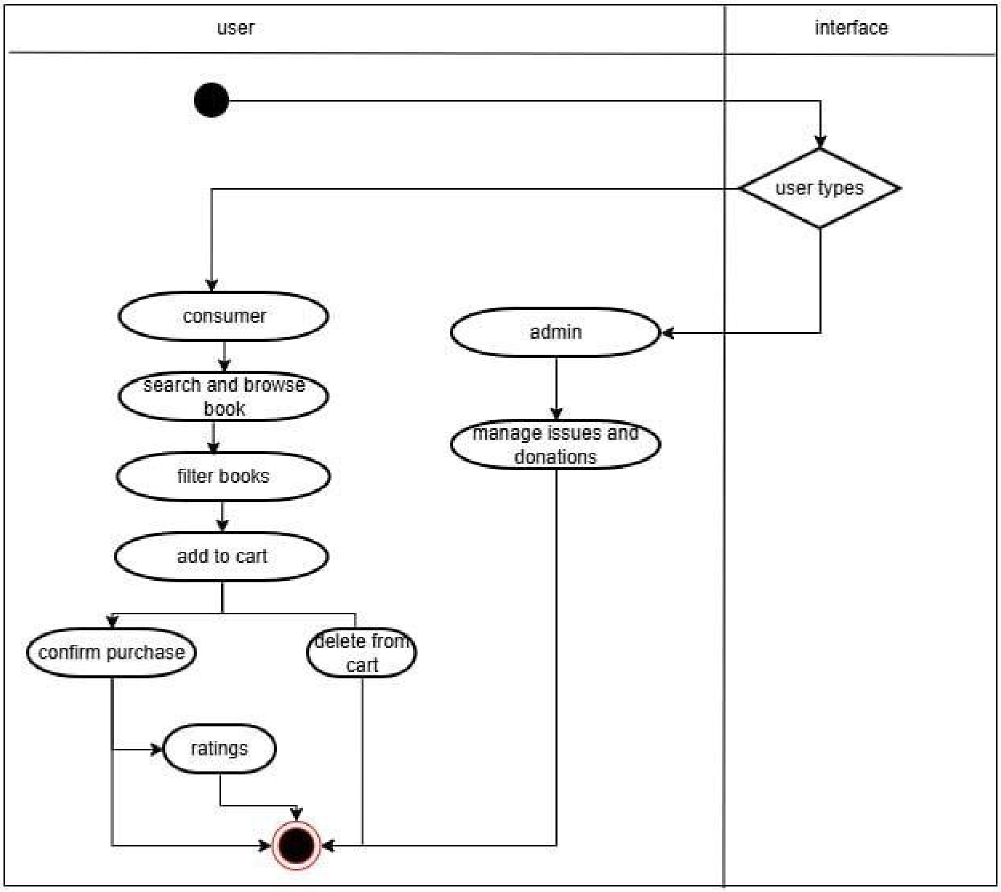


Figure: User Management

It differentiates the consumer and the admin role: consumers have the ability to view, filter, and buy books, whereas the admins deal with problems and donation, both culminating in a specific exit point.

Name: Book Management

Reference: Use Case 2.3

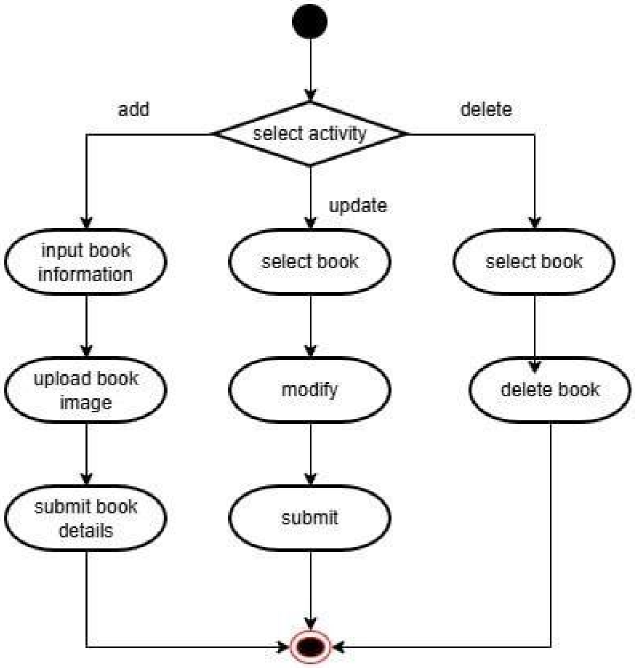


Figure: Book Management

Swimlane Diagram

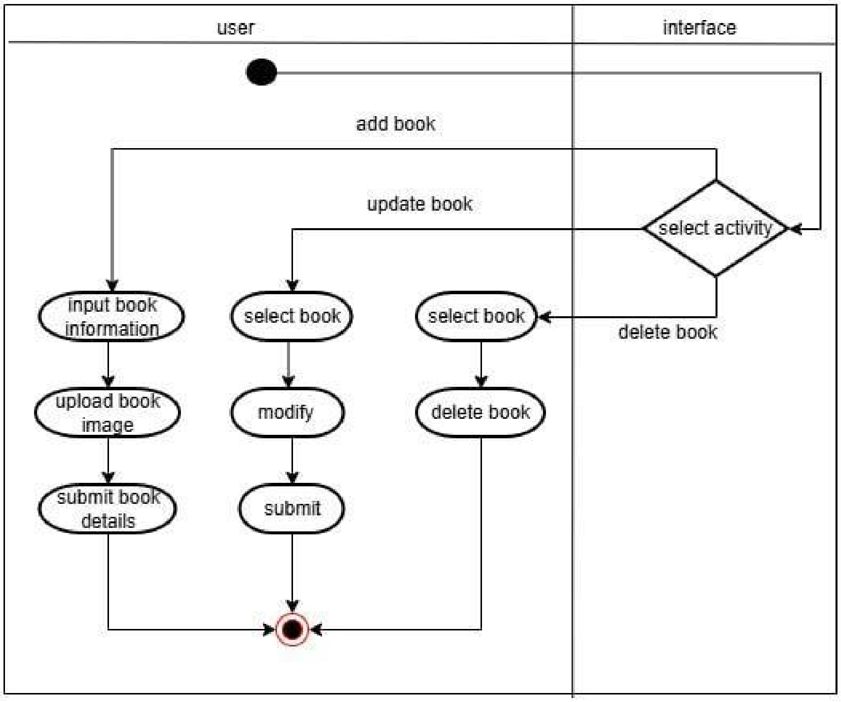
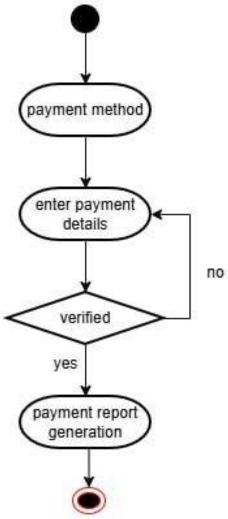


Figure: Book Management

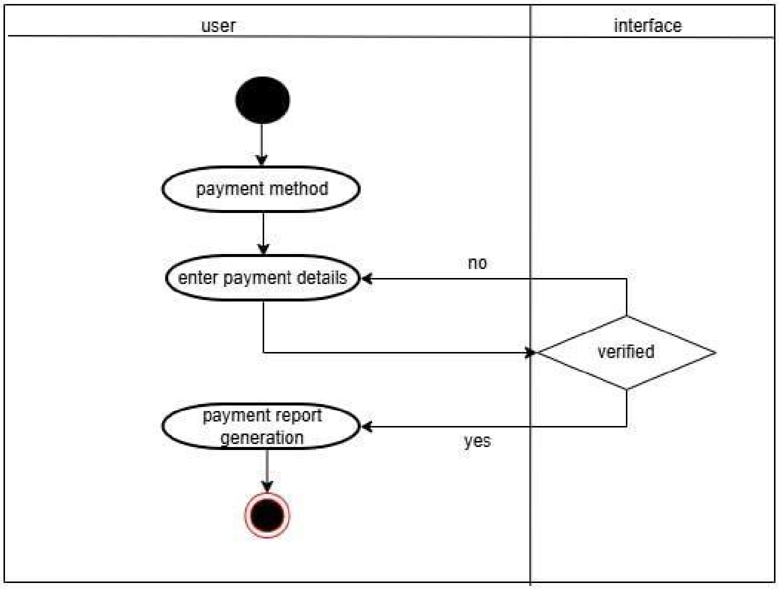
The flow supports the process of adding, updating, or deleting books and lets the user choose an activity and execute specific steps to complete the process that leads to a single process termination.

Name: Payment

Reference: Use Case 2.11

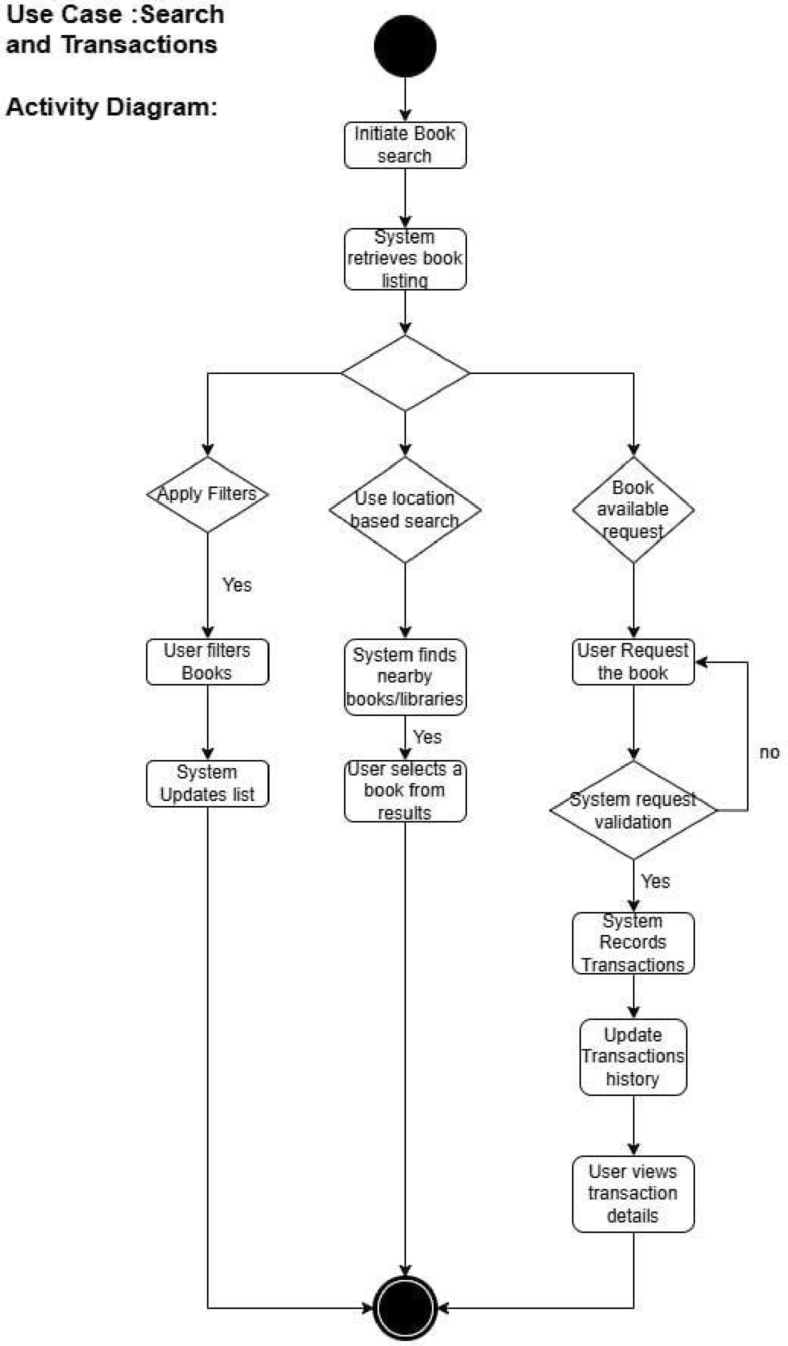


Swimlane Diagram:

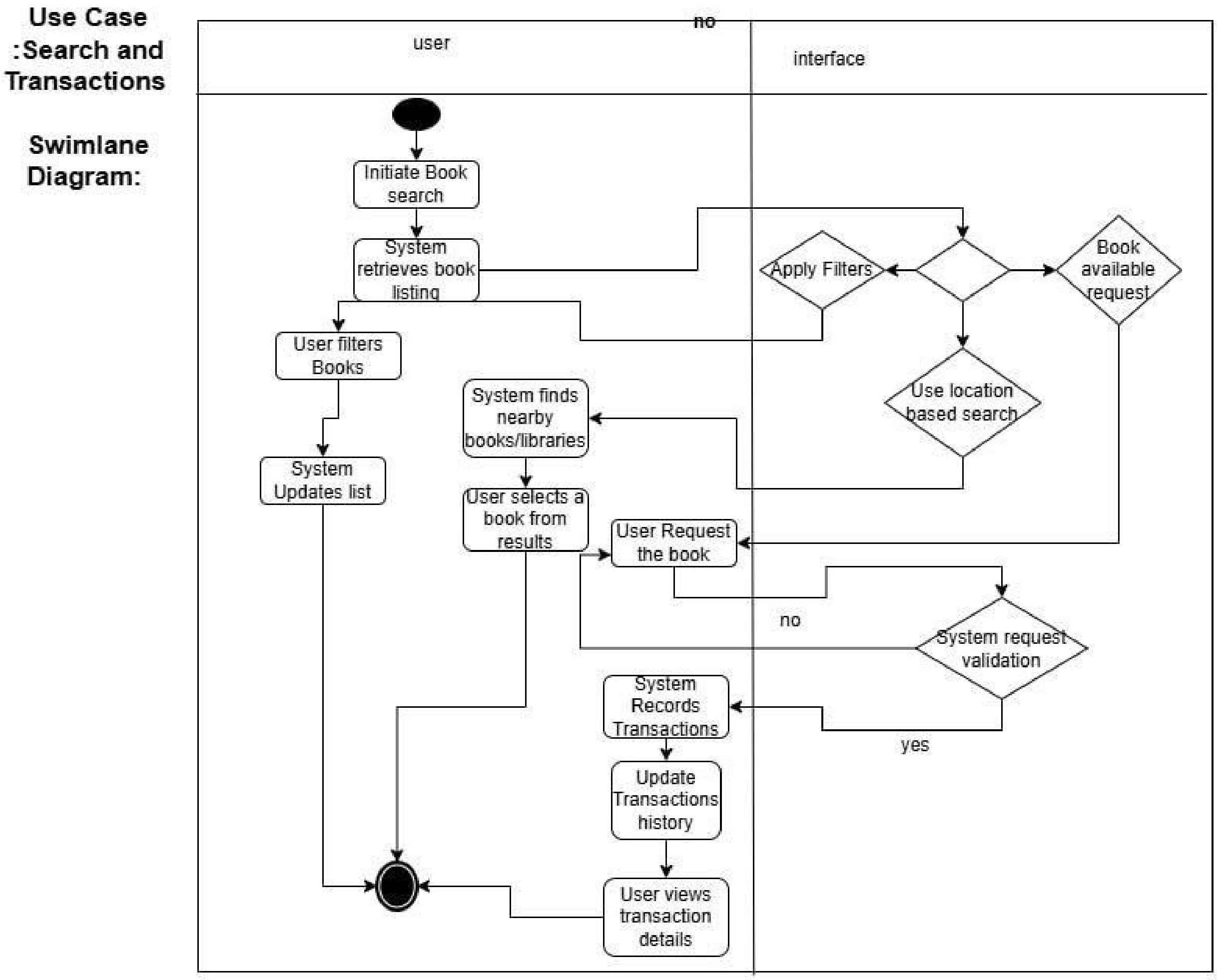


This diagram serves to represent the payment workflow, when a user selects a method, fill out the details and goes to report generation in a case of successful verification.

Name: Search & Transaction
Reference: Use Case 2.5


Use Case :Search and Transactions
Activity Diagram:
Initiate Book search
System retrieves book listing
Apply Filters
Use location based search
Book available request
Yes
User filters Books
System finds nearby books/libraries
User Request the book
Yes
no
System Updates list
User selects a book from results
System request validation
Yes
System Records Transactions
Update Transactions history
User views transaction details

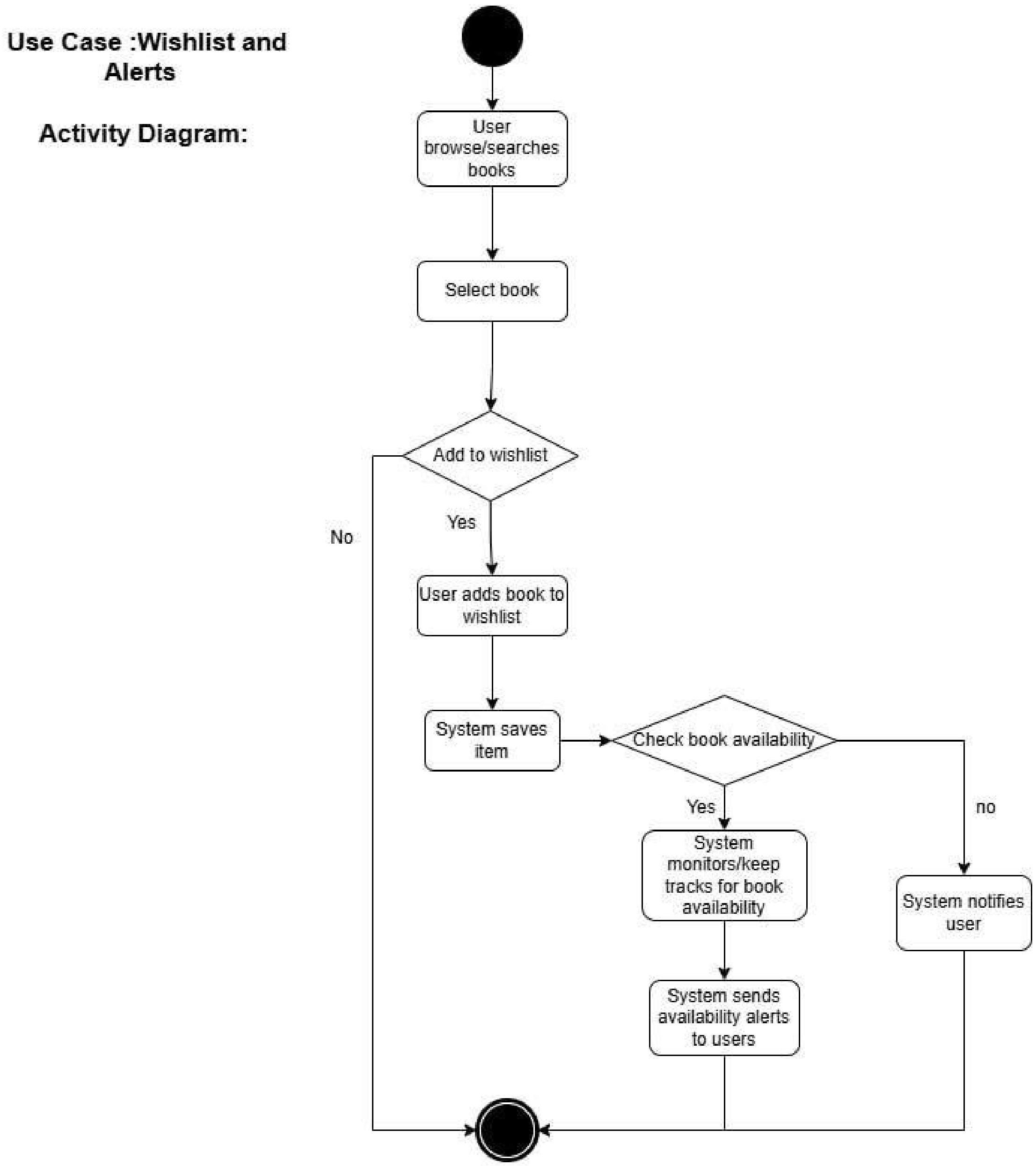
Swimlane Diagram


Use Case
:Search and
Transactions
Swimlane
Diagram:
no
user
interface
Initiate Book search
System retrieves book listing
Apply Filters
Book available request
User filters Books
Use location based search
System finds nearby books/libraries
System Updates list
User selects a book from results
User Request the book
no
System request validation
System Records Transactions
yes
Update Transactions history
User views transaction details

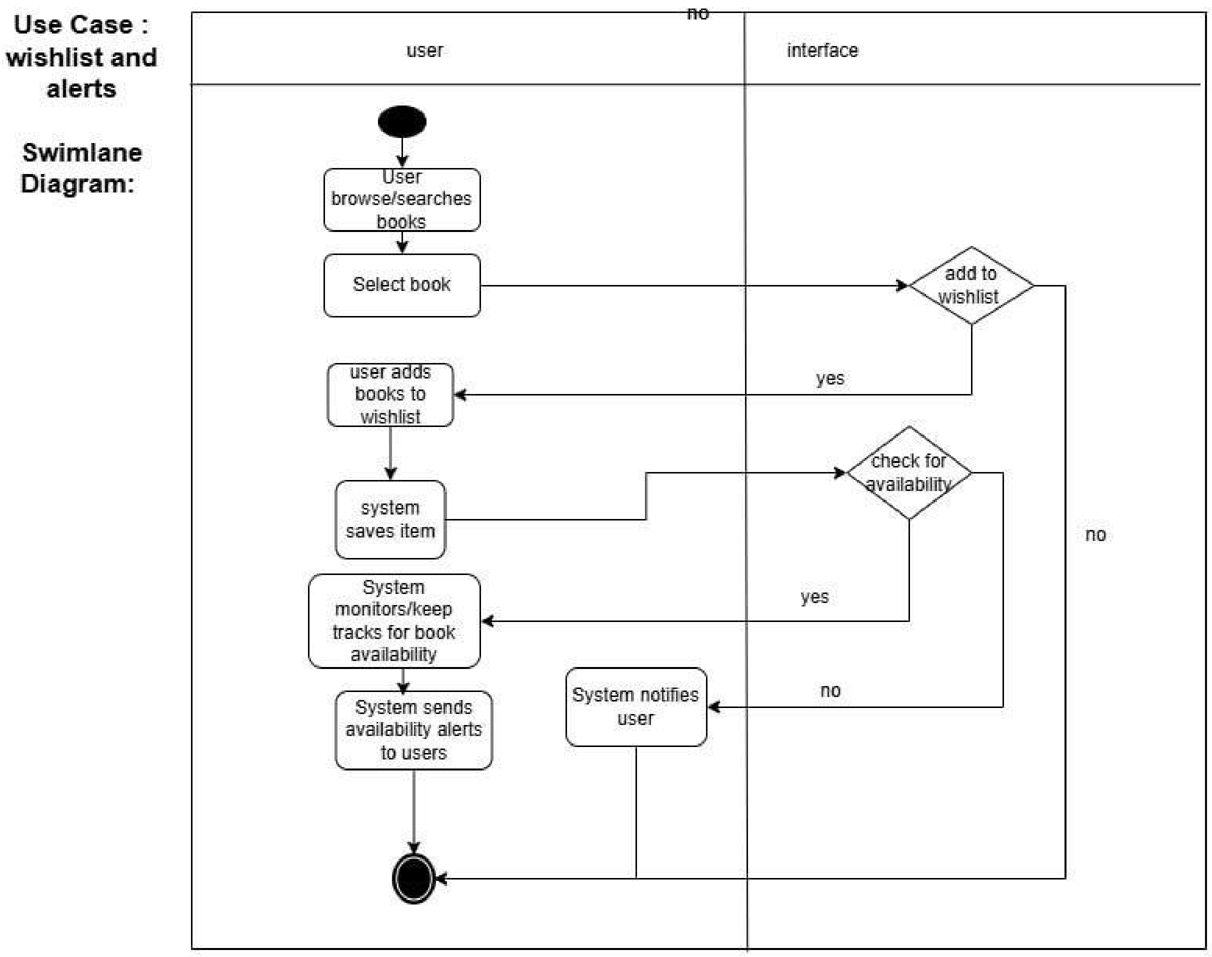
The diagram illustrates the search and transaction process of the book search and transaction, including filtering, location search, availability check, and recording of transactions, and finally viewing of the transaction details.

Name: Wishlist & Alerts

Reference: Use Case 2.6


Use Case :Wishlist and Alerts
Activity Diagram:
User browse/searches books
Select book
Add to wishlist
No
Yes
User adds book to wishlist
System saves item
Check book availability
Yes
no
System monitors/keep tracks for book availability
System notifies user
System sends availability alerts to users

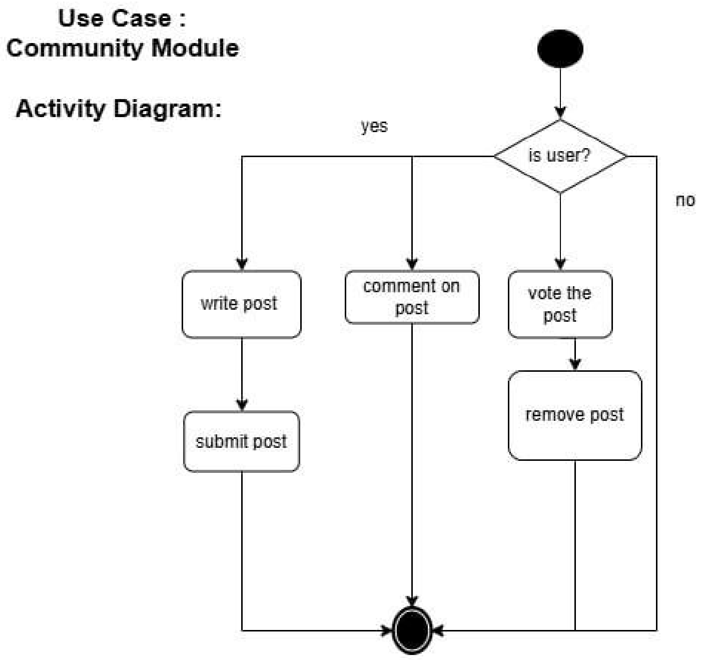
Swimlane Diagram


Use Case : wishlist and alerts
Swimlane Diagram:
no
user
interface
User browse/searches books
Select book
add to wishlist
yes
user adds books to wishlist
check for availability
system saves item
no
yes
System monitors/keep tracks for book availability
no
System sends availability alerts to users
System notifies user

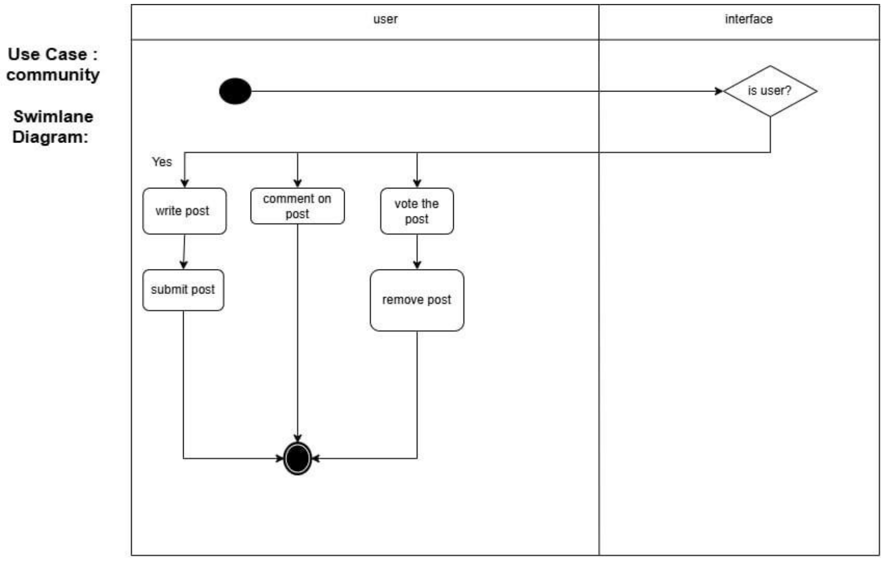
It represents that users add the books to their wishlist, and the system monitors the availability of the book sending alerts or notifications depending on the status of the book.

Name: Community Model

Reference: Use Case 2.7

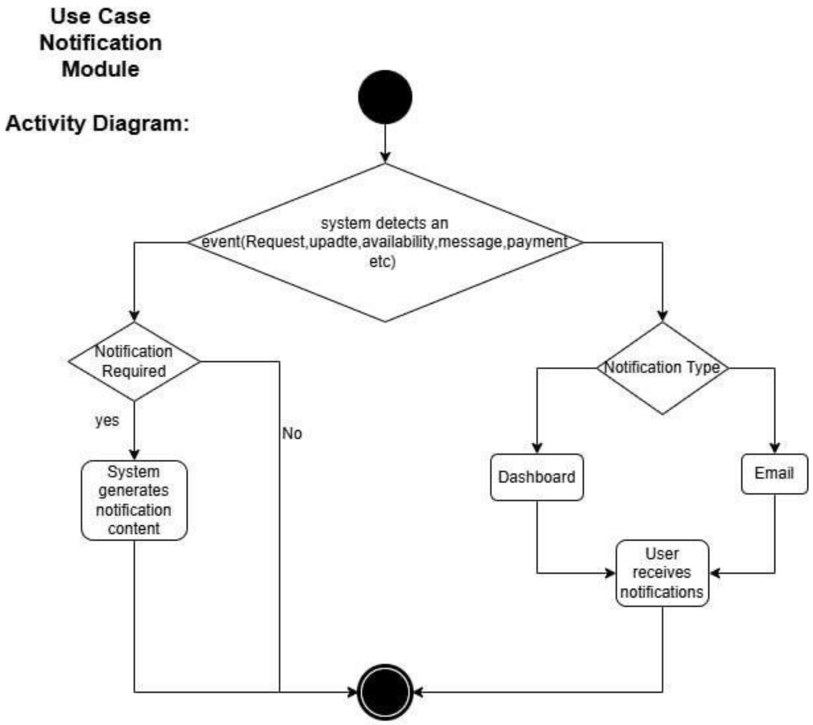


Swimlane Diagram

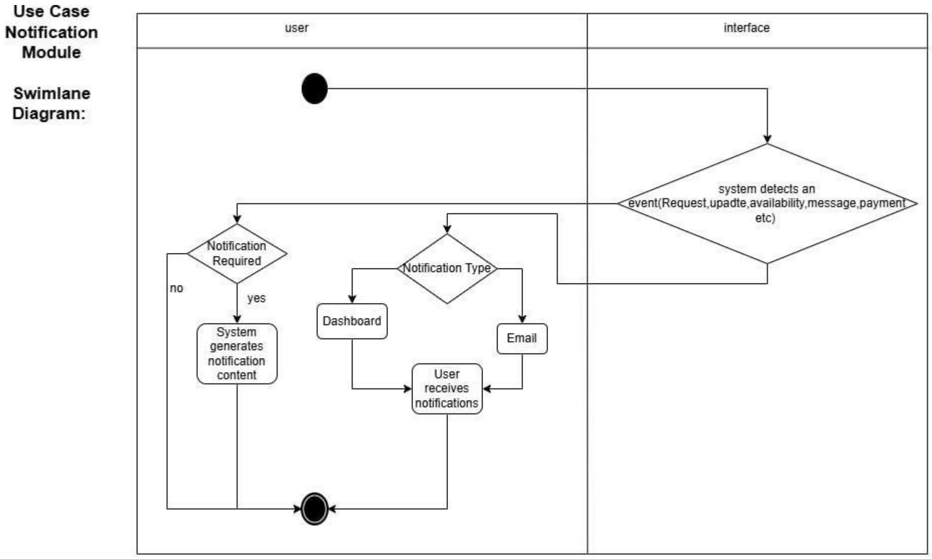


The flow enables authenticated users to write, comment, vote and remove posts and unauthenticated users are unable to interact.

Name: Notification Module
Reference: Use Case 2.9

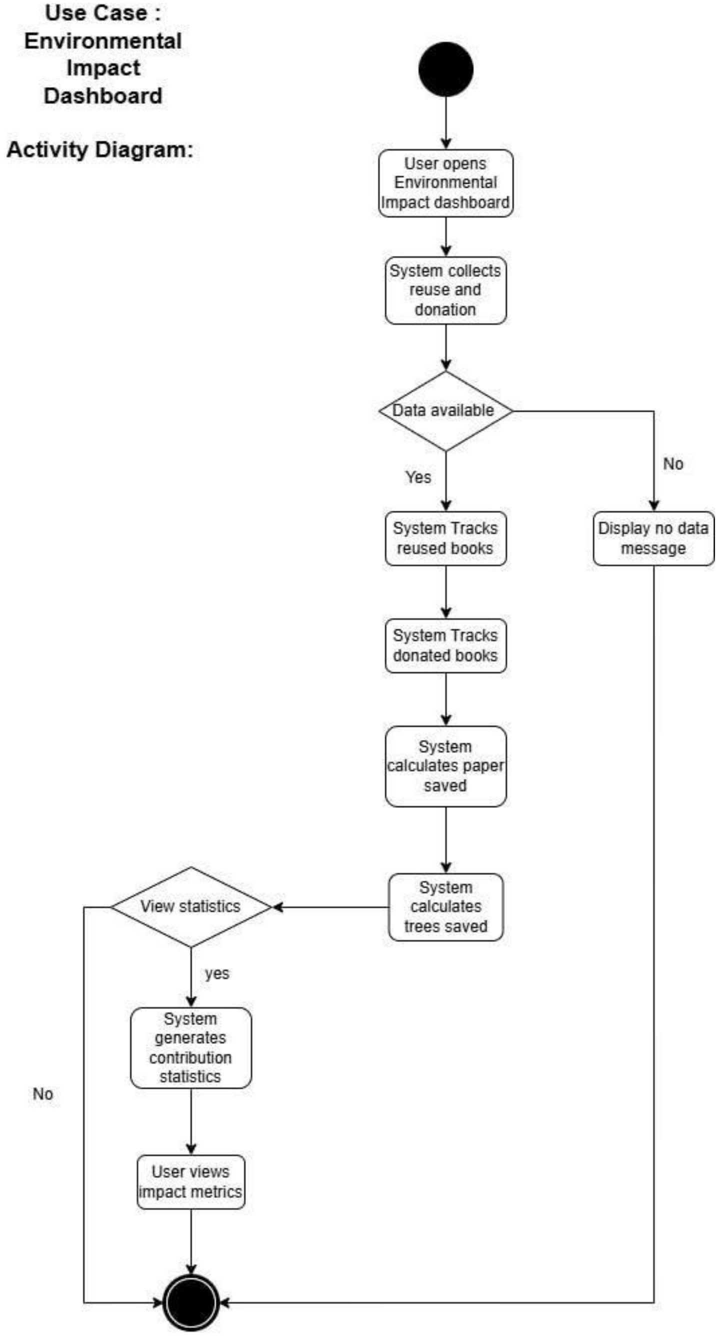


Swimlane Diagram:

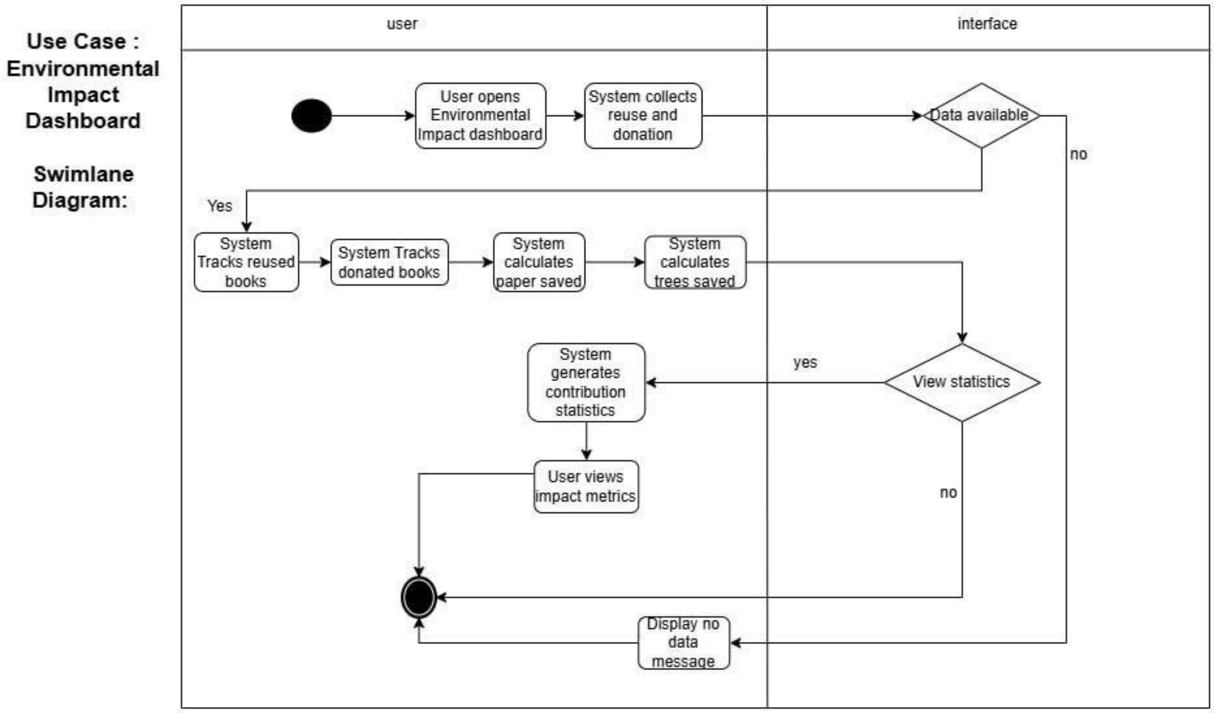


This flow chart describes the process by which the system identifies events, determines whether or not the notifications are required, creates the content, and transmits the content through a dashboard or e-mail.

Name: Environmental Impact Dashboard
Reference: Use Case 2.10


Use Case :
Environmental Impact Dashboard
Activity Diagram:
User opens Environmental Impact dashboard
System collects reuse and donation
Data available
Yes
No
System Tracks reused books
Display no data message
System Tracks donated books
System calculates paper saved
System calculates trees saved
View statistics
yes
No
System generates contribution statistics
User views impact metrics

Swimlane Diagram:


Use Case :
Environmental
Impact
Dashboard
Swimlane
Diagram:
user
interface
User opens Environmental Impact dashboard
System collects reuse and donation
Data available
no
Yes
System Tracks reused books
System Tracks donated books
System calculates paper saved
System calculates trees saved
yes
View statistics
System generates contribution statistics
User views impact metrics
no
Display no data message

This diagram enables the system to gather data regarding the reuse and donation data and show the impact metrics to the user in case the data is present.

6. Data Modeling

6.1 Entity Relation (ER) Diagram

Step 1: Identified Entities

The following entities are identified from the system model:

a) User
b) Role
c) Library
d) Book
e) BookListing
f) Transaction
g) Payment
h) Wishlist
i) CommunityPost
j) Comment
k) ChatMessage
l) Notification
m) EnvironmentalImpact

Step 2: Attributes and Primary Keys of Each Entity

User	user_id name password phone role_id (FK) is_verified, created_at)
Role	role_id role_name
Library	library_id user_id (FK) library_name email

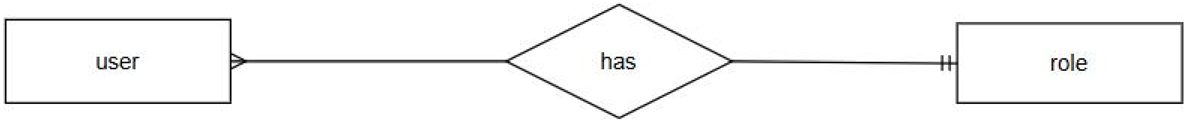
	phone address status created_at
Book	book_id title author category isbn condition
BookListing	listing_id book_id (FK) user_id(FK) library_id(FK) listing_type price availability_status approved_by_admin created_at
Transaction	transaction_id listing_id (FK) buyer_id(FK) transaction_type transaction_date status
Payment	payment_id transaction_id(FK) amount payment_method payment_status payment_date
Wishlist	wishlist_id user_id (FK) book_id(FK) created_at

CommunityPost	post_id user_id (FK) content post_type created_at status
Comment	comment_id post_id(FK) user_id(FK) comment_text created_at
ChatMessage	message_id sender_id(FK) receiver_id(FK) message_text sent_at status
Notification	notification_id user_id(FK) message, notification_type, is_read, created_at
EnvironmentalImpact	impact_id user_id(FK) reused_books donated_books paper_saved trees_saved, last_updated

Cardinality and Relationship:

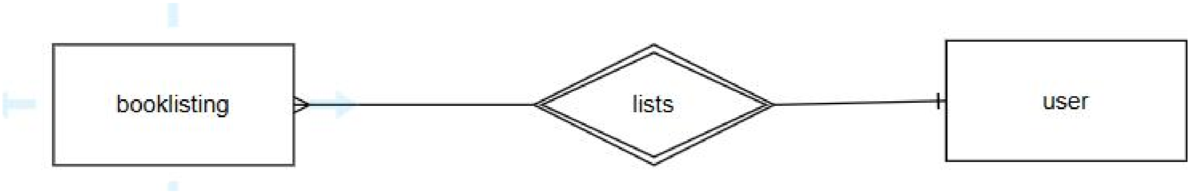
1. Role – User

- **Cardinality:** One-to-Many (1 : M)



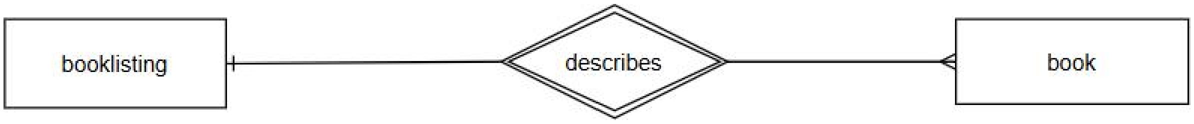
2. User – BookListing

- **Cardinality: One-to-Many (1 : M)**



3. Book – BookListing

- **Cardinality:** One-to-Many (1 : M)



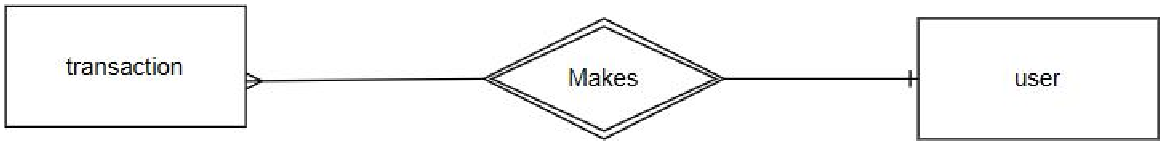
.

4. BookListing – Transaction

- **Cardinality:** One-to-Many (1 : M)

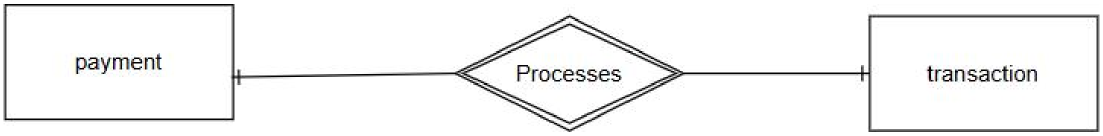
5. User – Transaction

- **Cardinality:** One-to-Many (1 : M)



6. **Transaction – Payment**

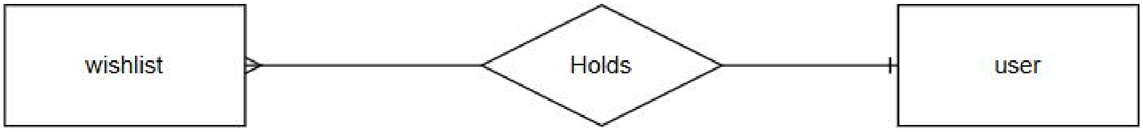
- **Cardinality:** One-to-One (1 : 1)



.

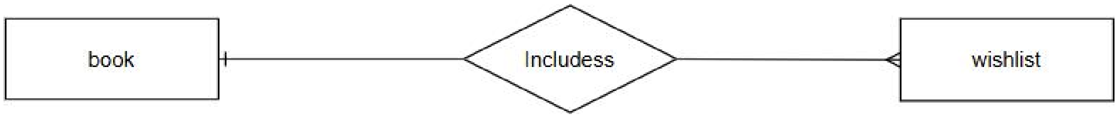
7. **User – Wishlist**

- **Cardinality:** One-to-Many (1 : M)



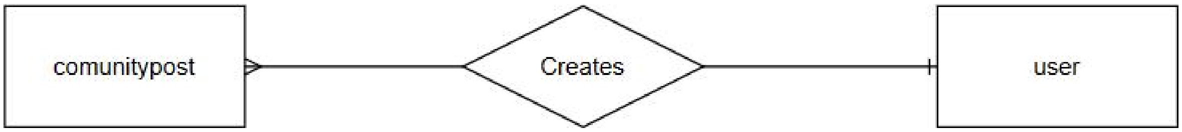
8. **Wishlist – Book**

- **Cardinality:** Many-to-One (M : 1)



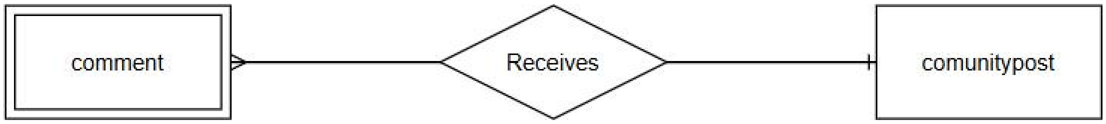
9. **User – CommunityPost**

- **Cardinality:** One-to-Many (1 : M)



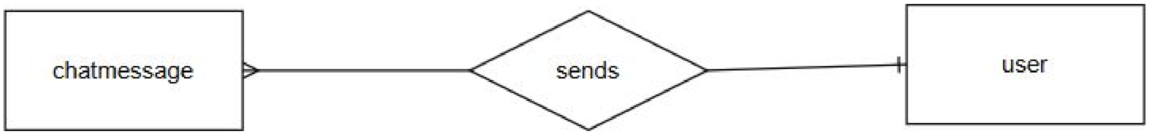
10. **CommunityPost – Comment**

- **Cardinality:** One-to-Many (1 : M)



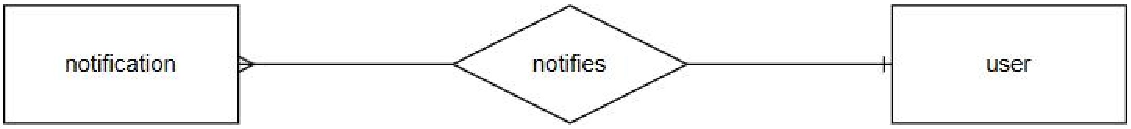
11. User – ChatMessage

- **Cardinality:** One-to-Many (1 : M)



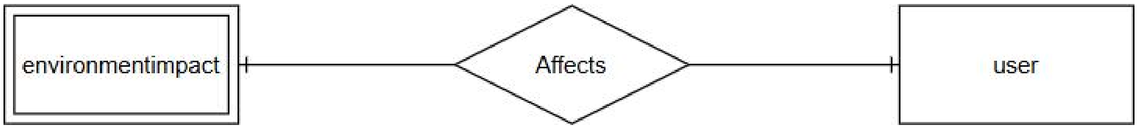
12. User – Notification

- **Cardinality:** One-to-Many (1 : M)



13. User – EnvironmentaI impact

- **Cardinality:** One-to-One (1 : 1)



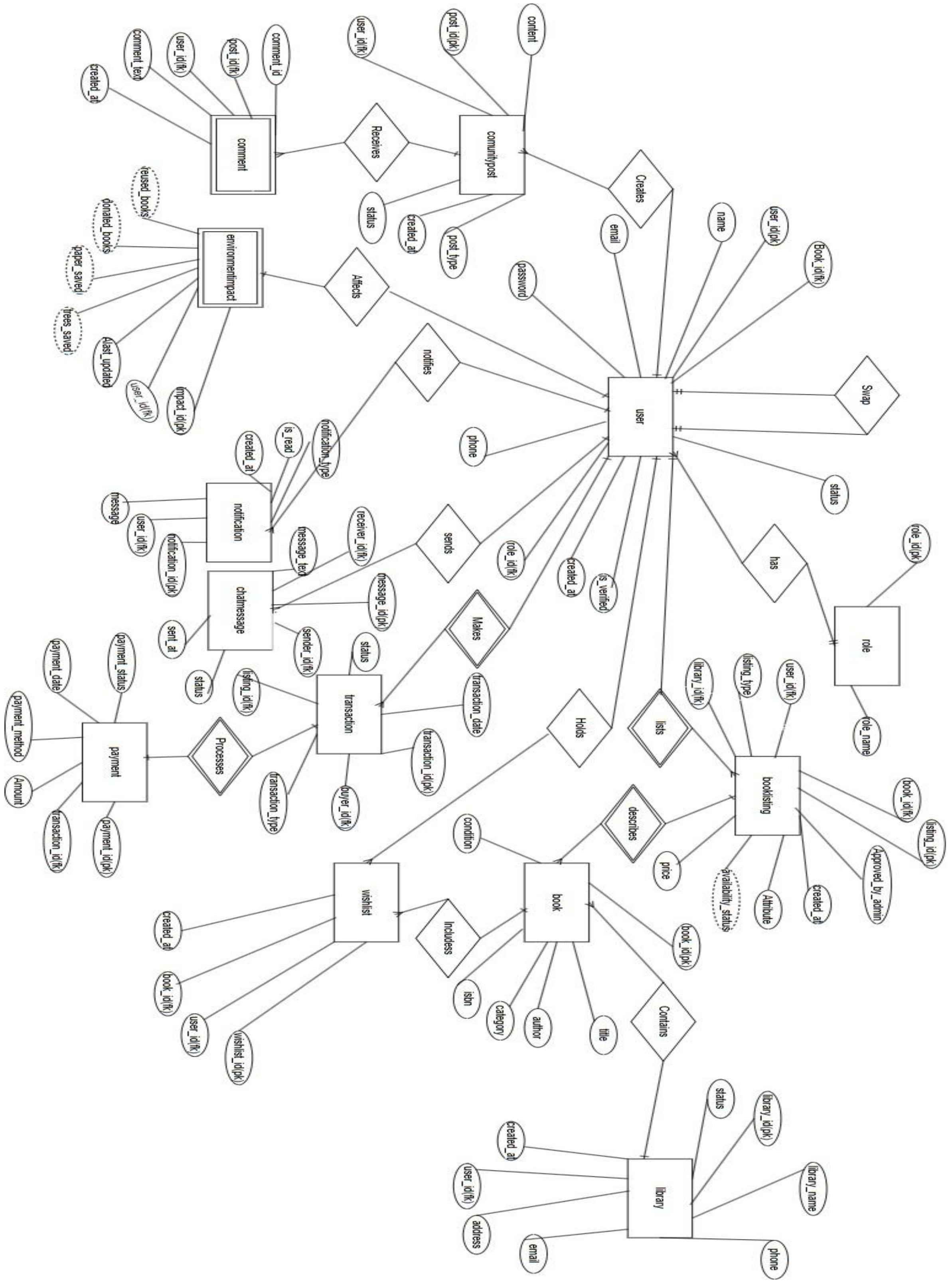


Figure: ER diagram For BoiChokro

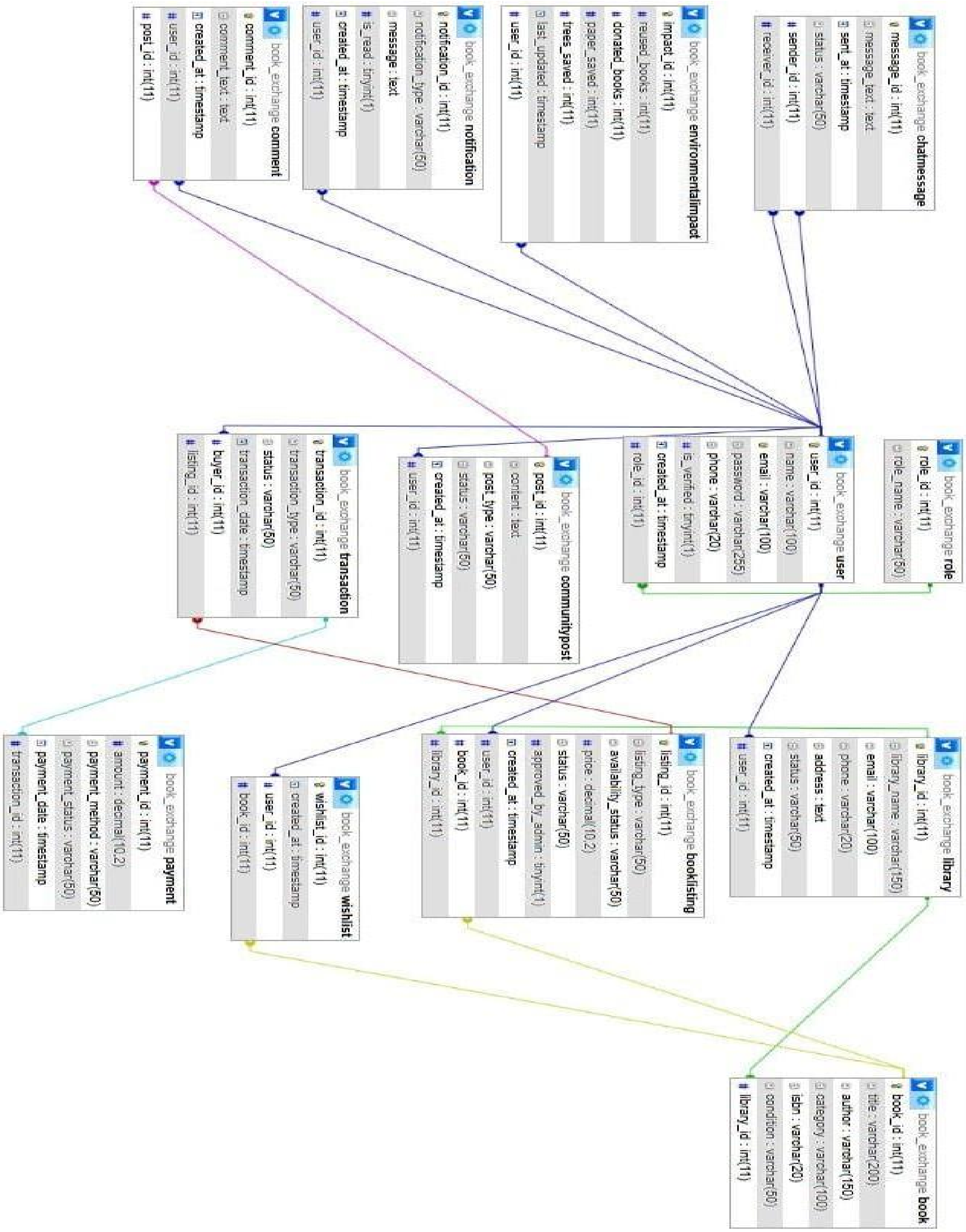


Figure: Schema Diagram for BoiChokro

7. Class-Based Modeling

Class-based modeling identifies the classes, attributes and operations required to represent the structure of the system. It helps transform requirements into an object-oriented design by identifying system entities and defining their responsibilities.

7.1 Identify Class and Attributes

Step-1: Identifying and categorize all nouns

All nouns were extracted from the problem statement, requirements, and use-case descriptions of the Boichokro system and categorized as follows:

External Entities : User, Library, Admin, Payment Gateway, School, Dono**r.**

Things : Book, Book Listing, Wishlist, Notification, Message, Review, Rating, Donation, Transaction.

Occurrences / Events : Register, Login, Logout, Post Book, Approve Listing, Search, Request Book, Payment, Chat, Donate, Borrow, Return, Generate Report.

Roles : Admin, Library Authority, Reader, Student.

Organizational Units: Library, Community, Educational Institution.

Places : Book Location, Library Location.

Structures : System, Server, Database, Web Interface.

Step-2: Selection of Potential Classes

The identified nouns were evaluated using the following selection characteristics:

1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Table: Final Class Selection

Potential Class	Characteristic Number That Applies
User	Accepted
Admin	Accepted
Library	Accepted
Book	Accepted
Book Listing	Accepted
Transaction	Accepted
Donation	Accepted
Wishlist	Accepted
Community Post	Accepted
Chat Message	Accepted
Notification	Accepted
Payment	Accepted
Environmental Impact	Accepted
Server	Rejected (fails 3)
Interface	Rejected (fails 1,4,5)
Button	Rejected (fails 1,3,5,6)
Location	Rejected (fails 3)
Search	Rejected (fails 3)
Report	Rejected (fails 1,3,6)

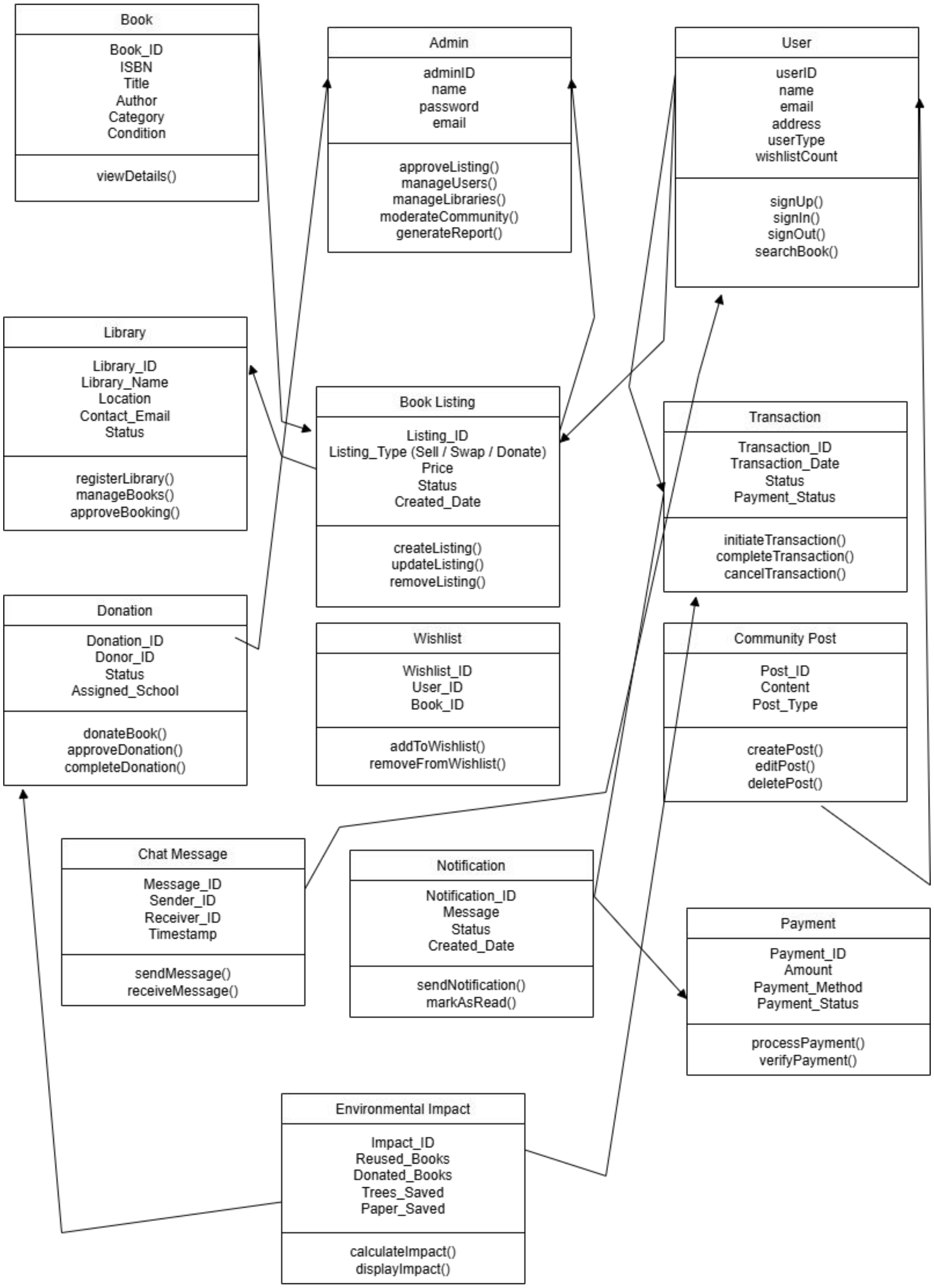
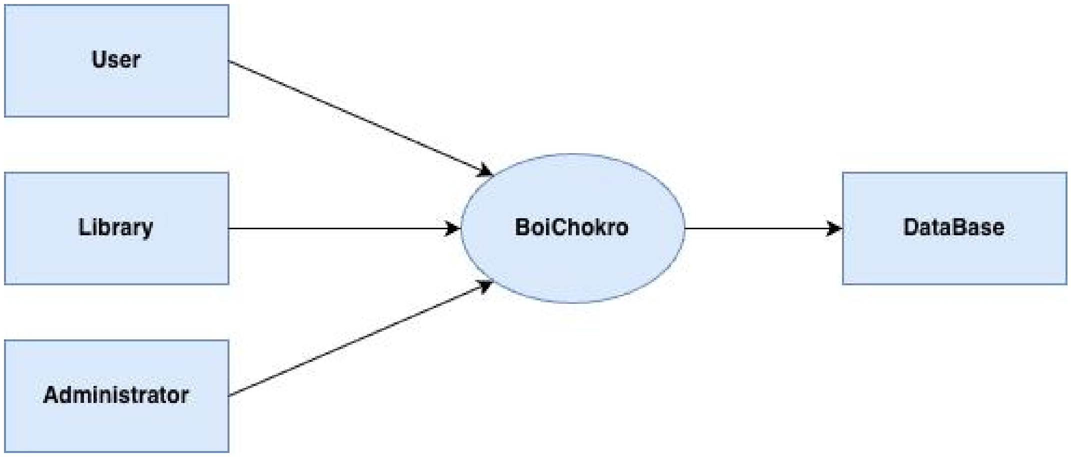


Figure: Class Diagram for Boichokro

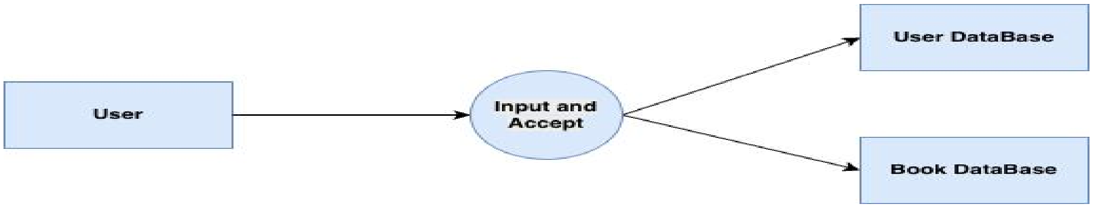
8. Flow-Oriented Modeling

8.1 Data Flow Diagram



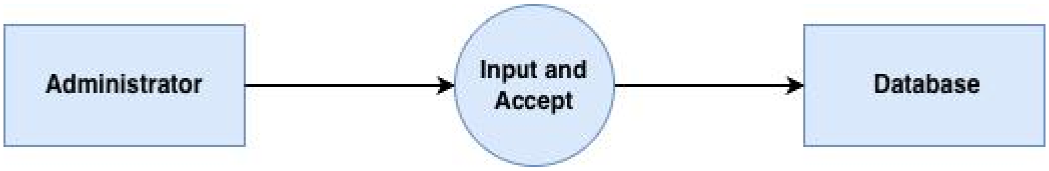
Level 0 for BoiChokro

This schematic illustrates the interaction point of users, libraries and administrators with the central BoiChokro system, which receives the inputs of the users and stores the data in a single database



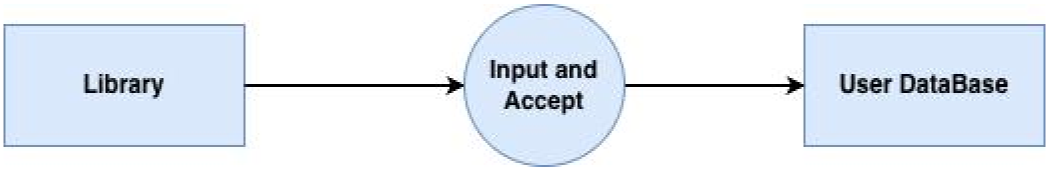
Level 1.1(user) For BoiChokro

It shows the process of accepting user data into the system and directing it to User and Book databases to be stored and processed.



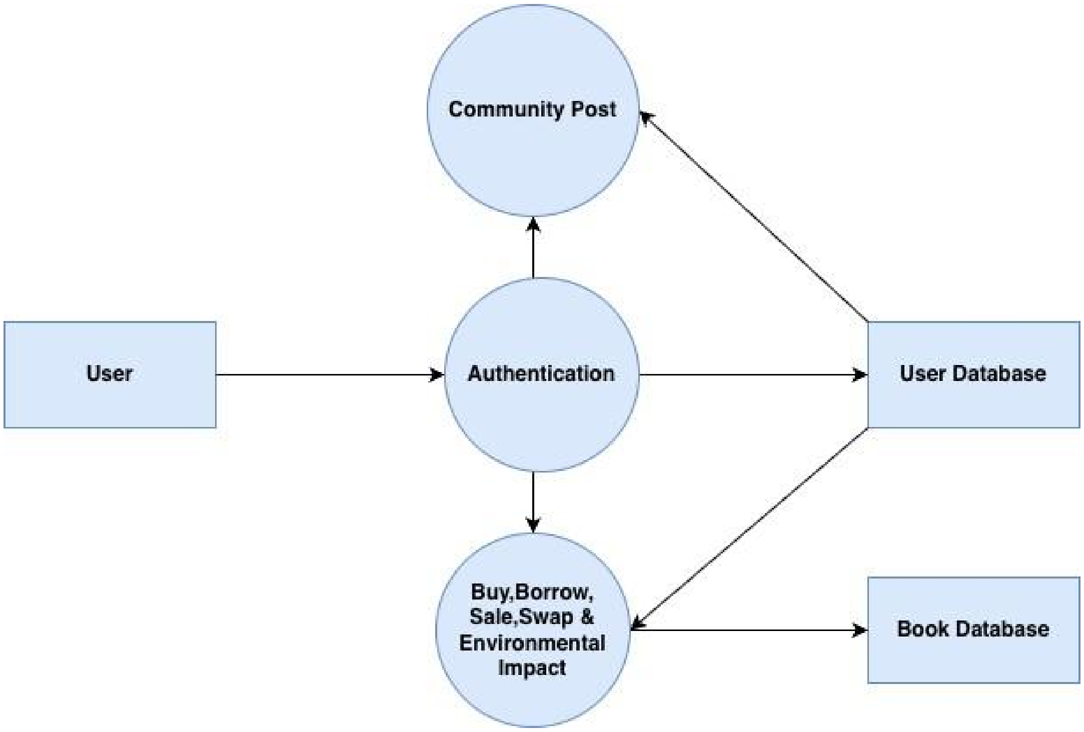
Level 1.2(Administrator) For BoiChokro

The flow indicates the entry of data by the administrators into the system, and the information will further be processed and stored in the central database.



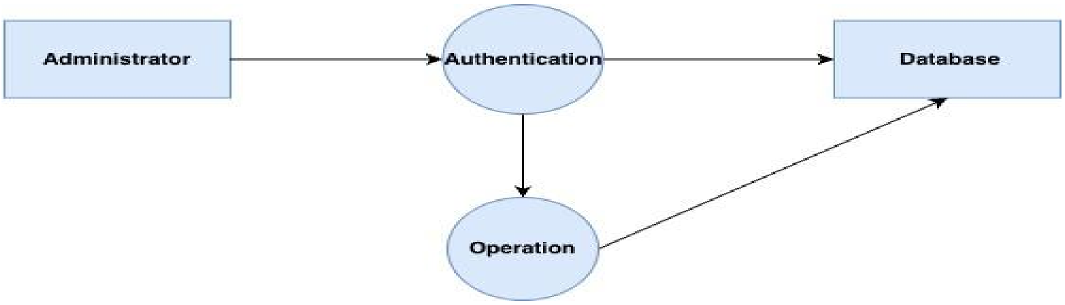
Level 1.3 (Library) For BoiChokro

The diagram describes how the libraries provide their information to the system via the input module and this information is collected by the system and reflected in the User database.



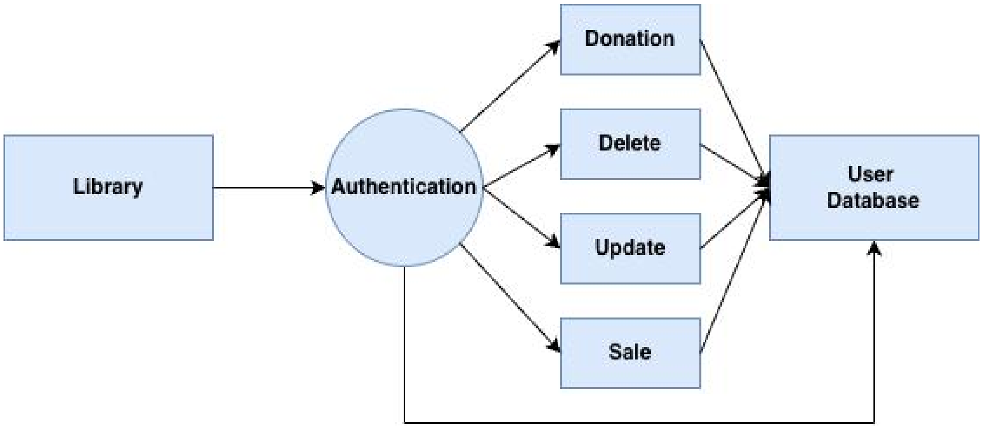
Level 2.1(User) For BoiChokro

This chart indicates how the registered users will use posts in the community and access the facilities of books such as purchase, loan, and environmental management, and the data flow will be between the user and the book databases.



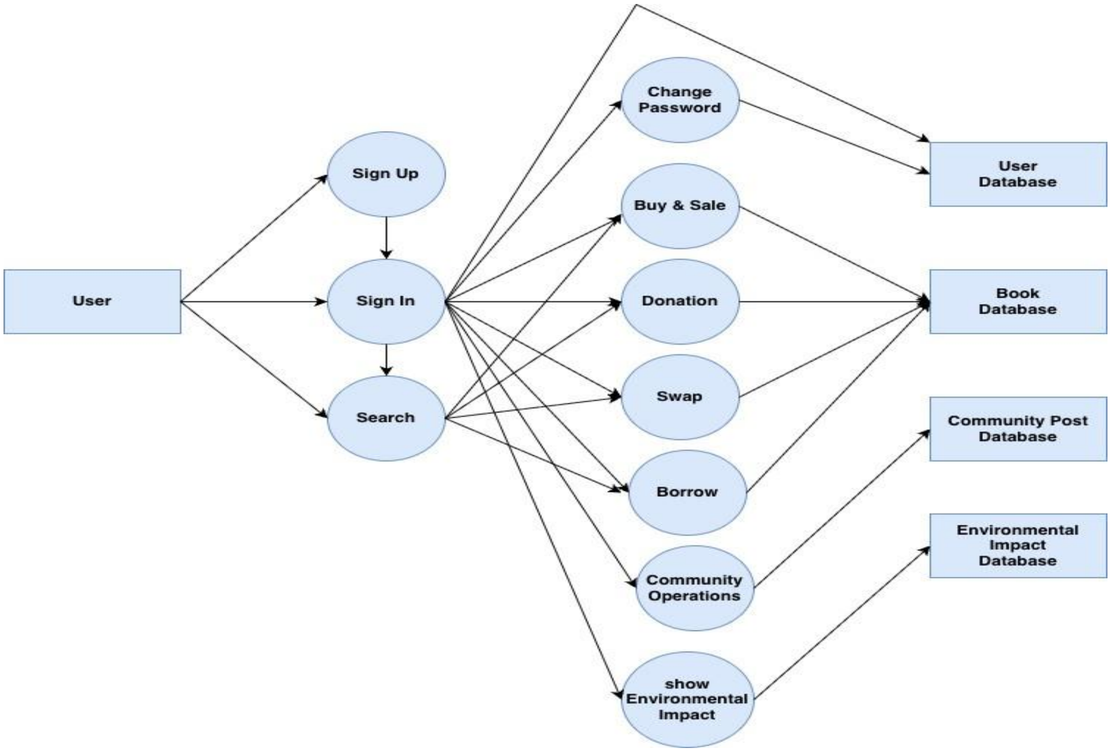
Level 2.2 (Administrator) For BoiChokro

This describes the type of activity that the administrators perform to gain access to the system database, conducting operations and interacting with it in order to perform administrative activities.



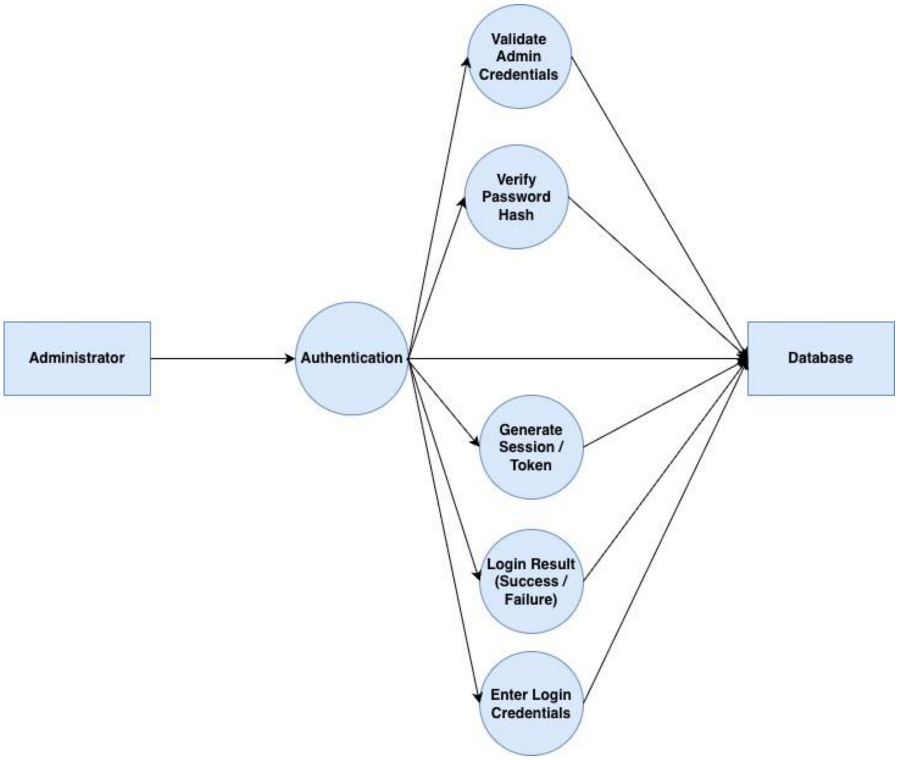
Level 2.3 (library) For BoiChokro

Here, libraries verify and execute operations such as donation, deletion, update and sale, which all update user database.



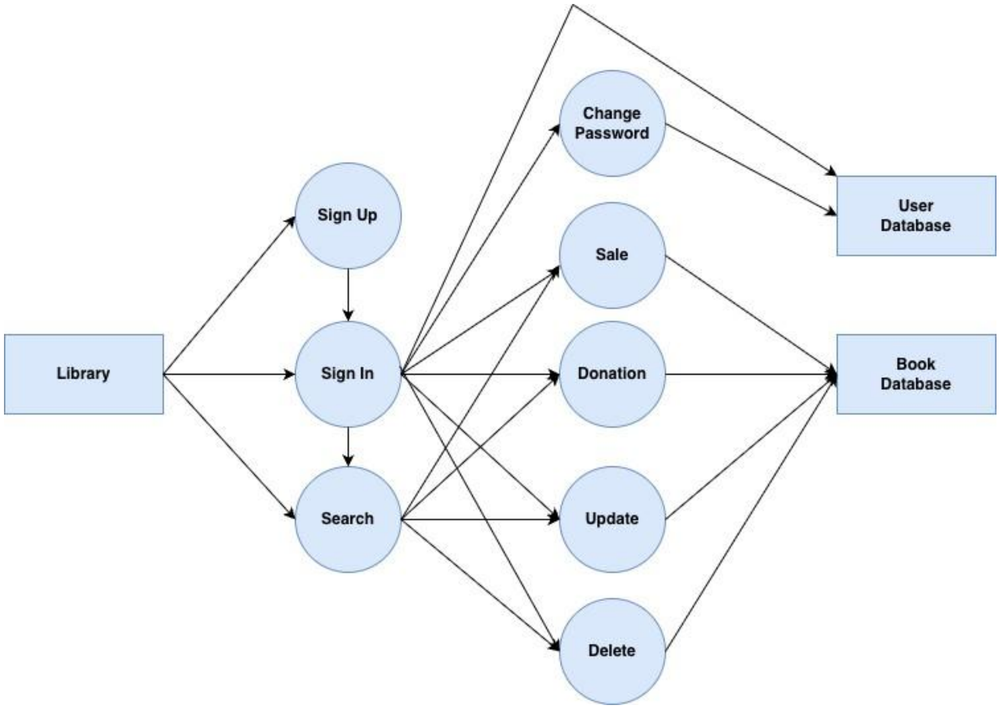
Level 3.1 (User) For BoiChokro

This chart outlines user interaction during the sign-up and sign-in to accessing functions such as booking transactions, community posts, and environmental impact with the data entering into databases accordingly.



Level 3.2 (Administrator) For BoiChokro

It elaborates on the process of authentication of the administrator, credential validation, password hashing and generation of a session, which are recorded in the system database.



Level 3.3 (Library) For BoiChokro

This flow depicts the operations of authenticating and performing such operations as sale, donation, update, and deletion, which interact with users and book database.

9. Behavioral Modeling

9.1 State Transition Diagram

Name: User Authentication

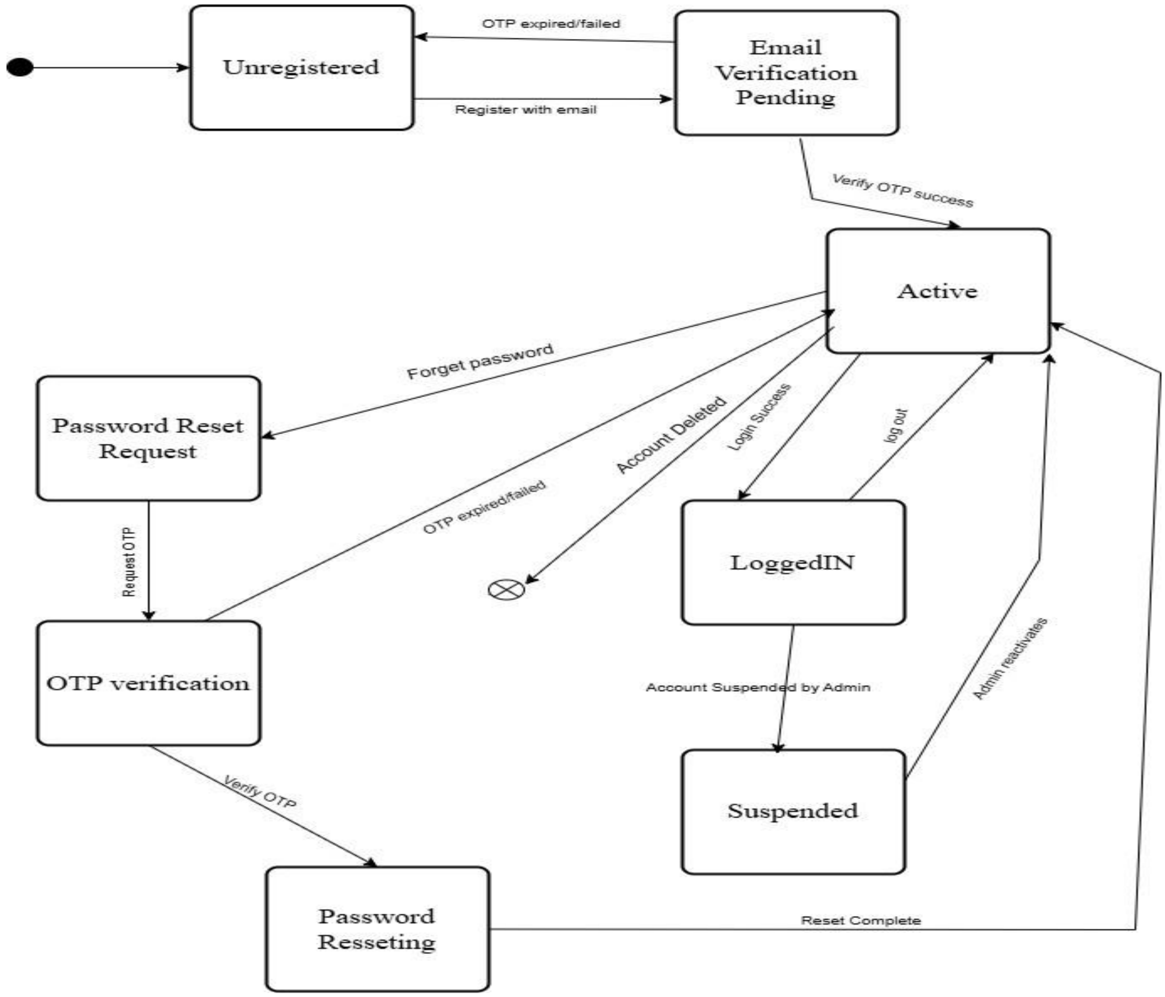


Figure: User Authentication

The diagram is an overview of the user states between registration and login, password reset, suspension, and reactivation, with the transitions between states being initiated by actions such as OTP verification, and an abuse decision made by the administration.

ID: 02

Name: User Logged In

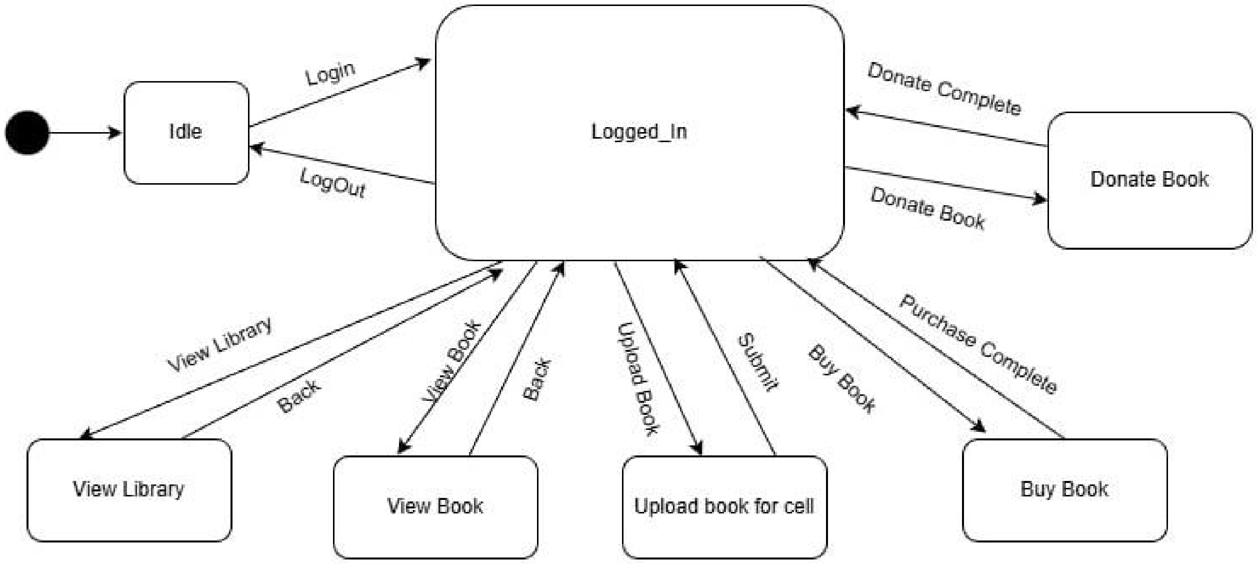


Figure: User Logged In

It describes behavior of the user after the login process and the user is able to view, upload, donate, and purchase books, with each process returning to the logged-in state.

ID: 03

Name: Library

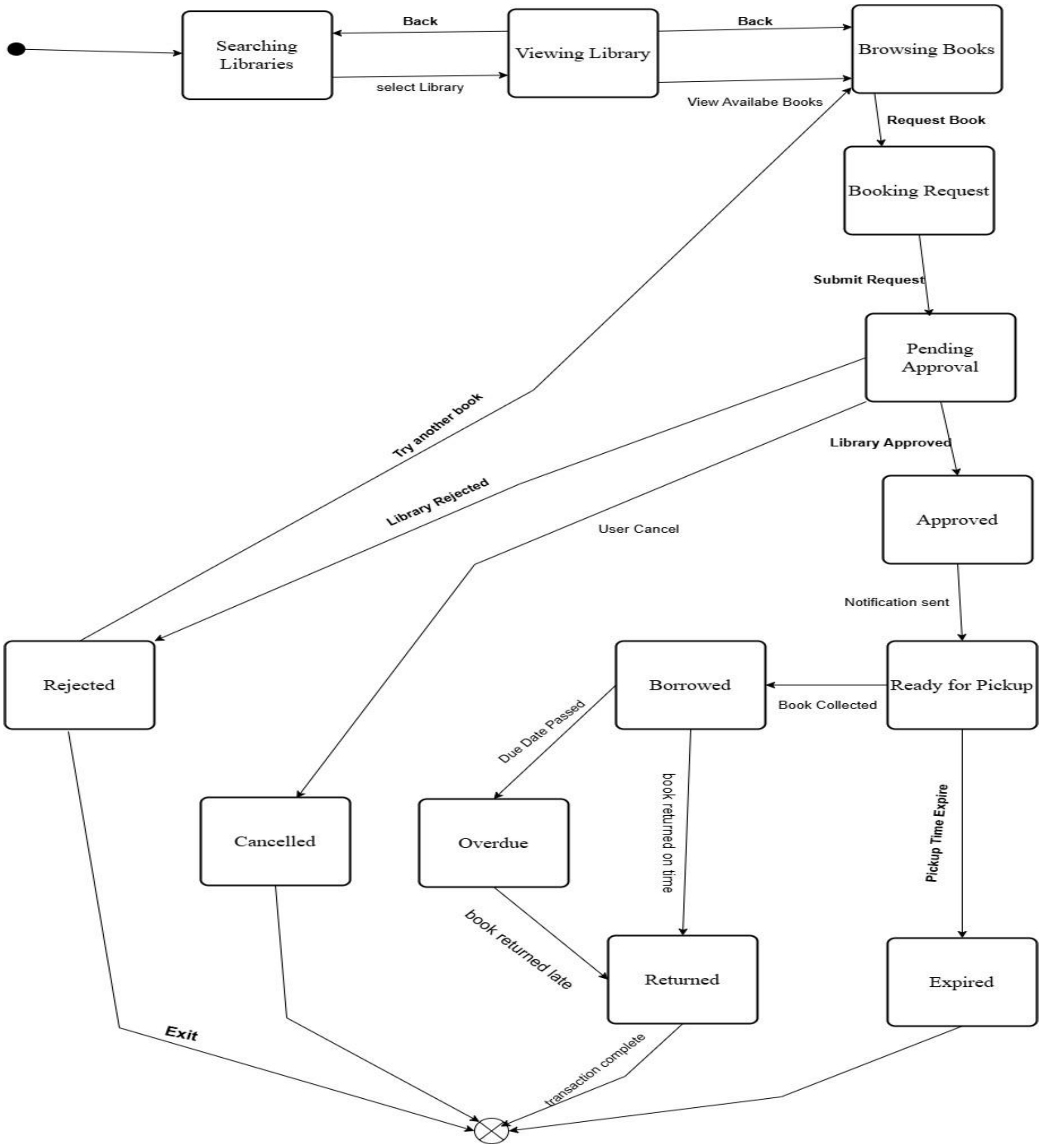


Figure: Library

This flow illustrates the way in which the users search libraries, request books and navigate approval, rejection, cancellation, and overdue situations and finally complete the transaction.

ID: 04

Name: Wishlist & Alerts

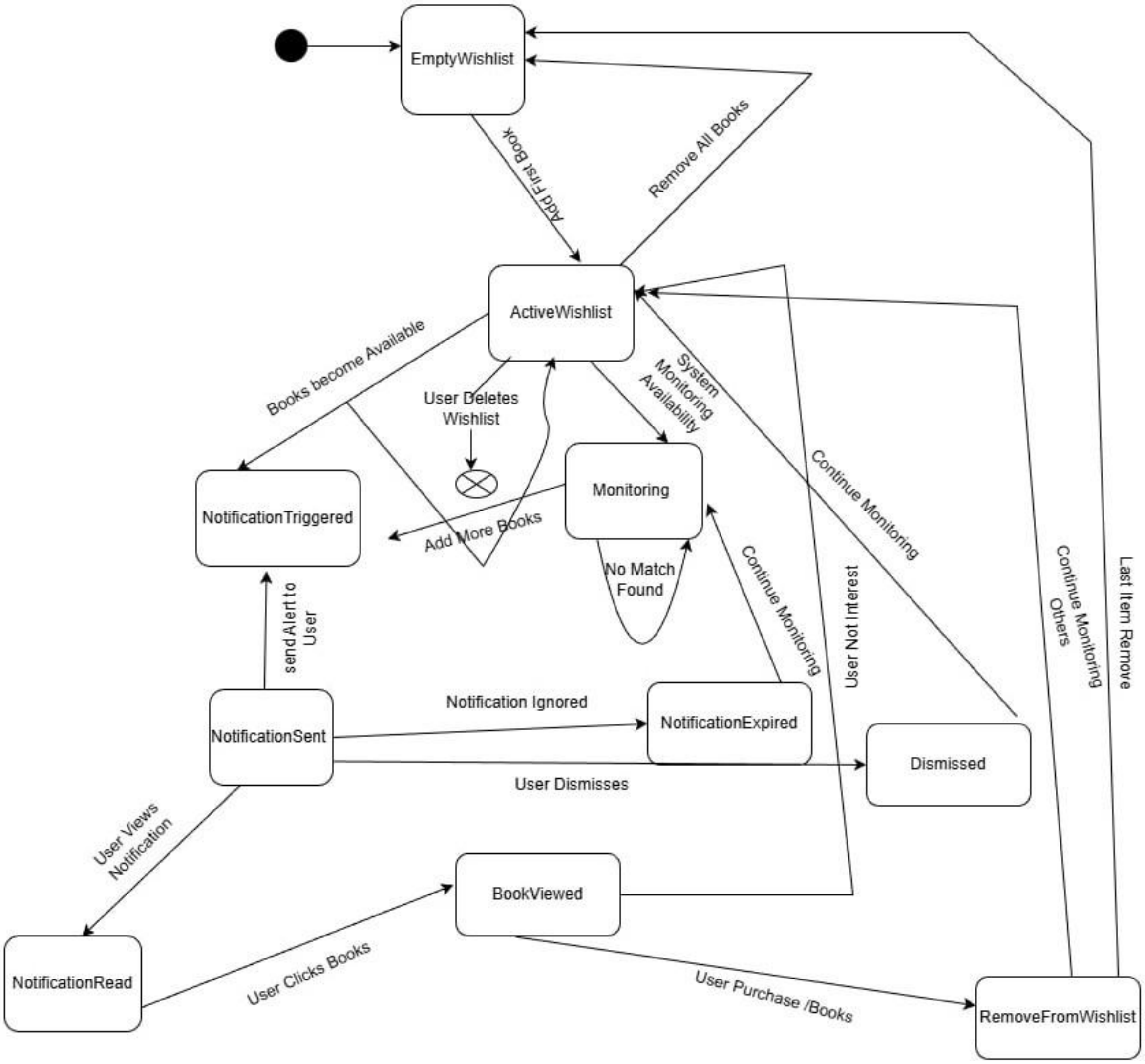


Figure: Wishlist & Alerts

This illustration follows the process of the wishlist operation of adding books to the track of the availability, sending a notification, seeing the books, and deleting the items, being dynamic as the user becomes involved in its management

ID: 05

Name: Payments

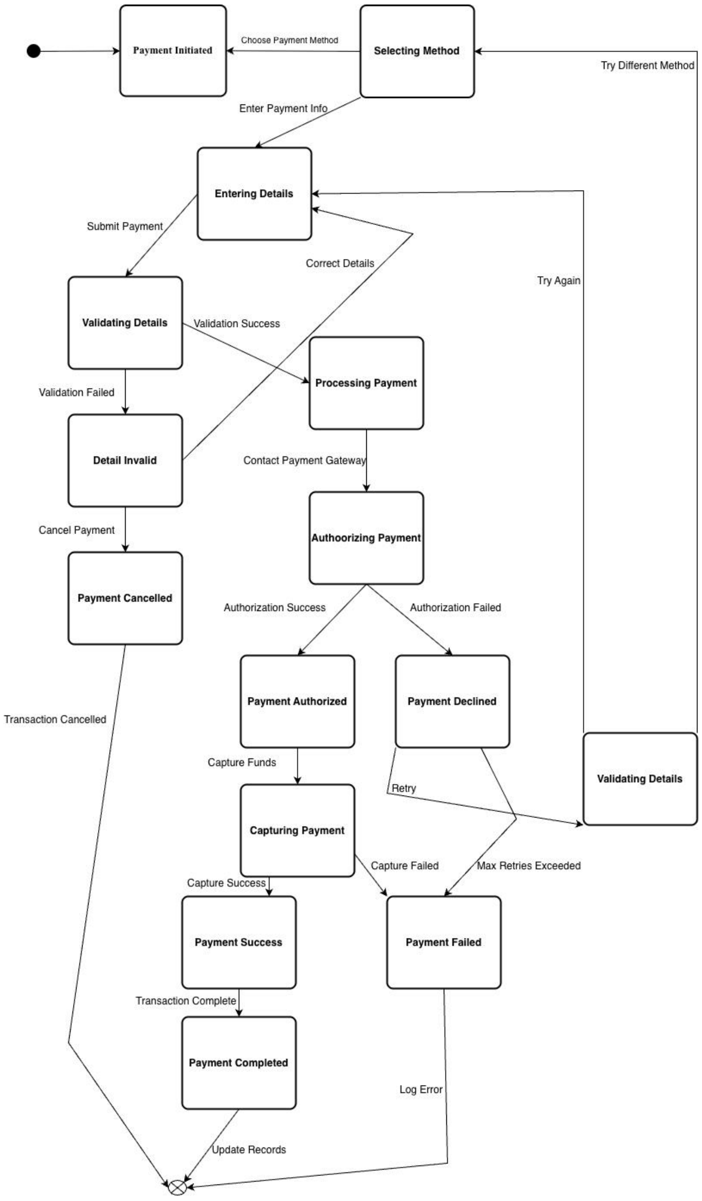


Figure: Payments

The diagram covers the entire payment process, including the choice of method, the validation of the detail, the authorization, the capture of the funds, the completion of the transaction, and the errors and the resilience strategies.

ID: 06

Name: Community Post

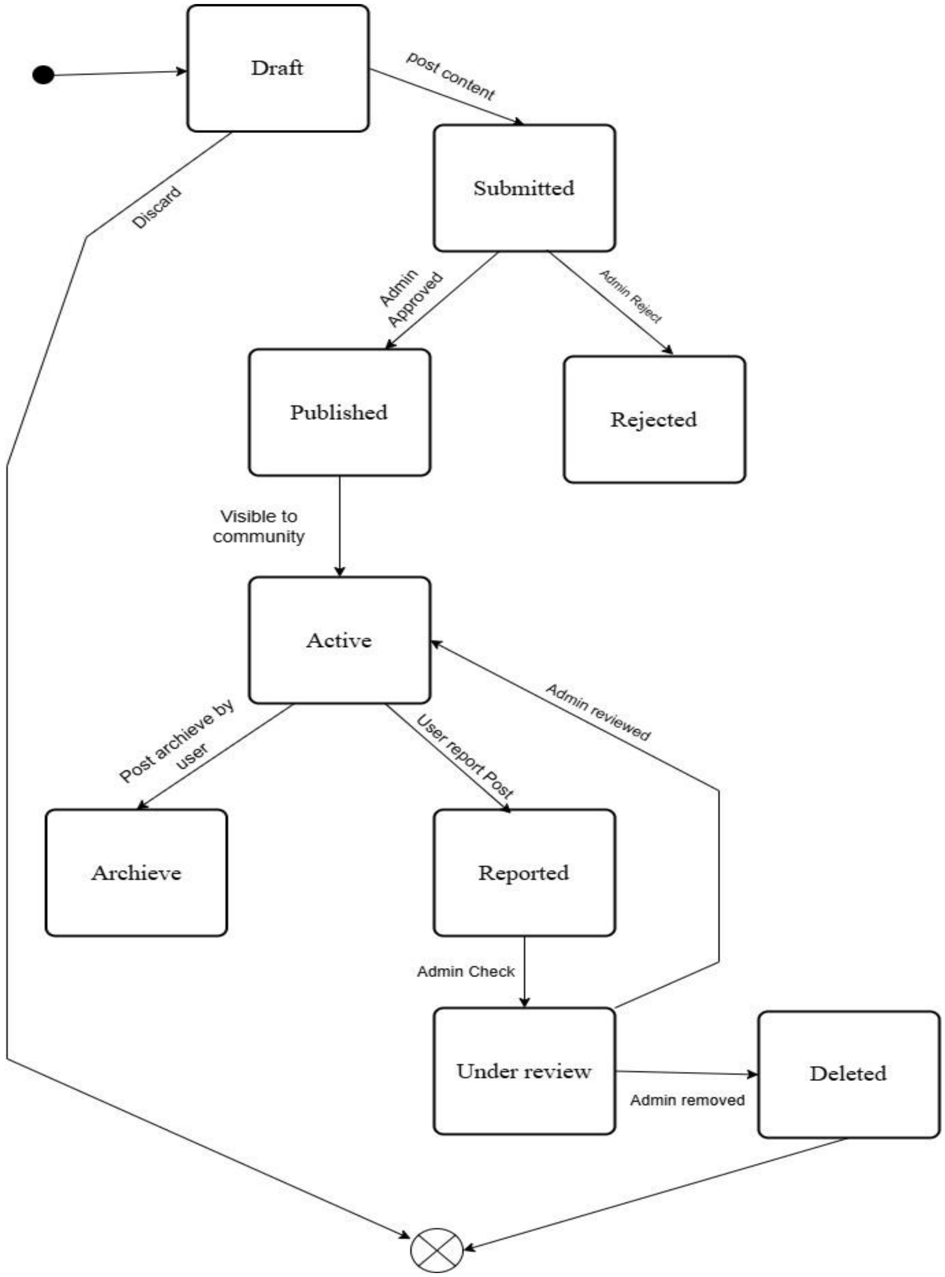


Figure: Community Post

This follows the path of a post between draft and publication, through the steps of approval by the administration, report on the post by users, review and the potential possibility of the result being an archive or the post being deleted.

ID: 07

Name: Environmental Dashboard

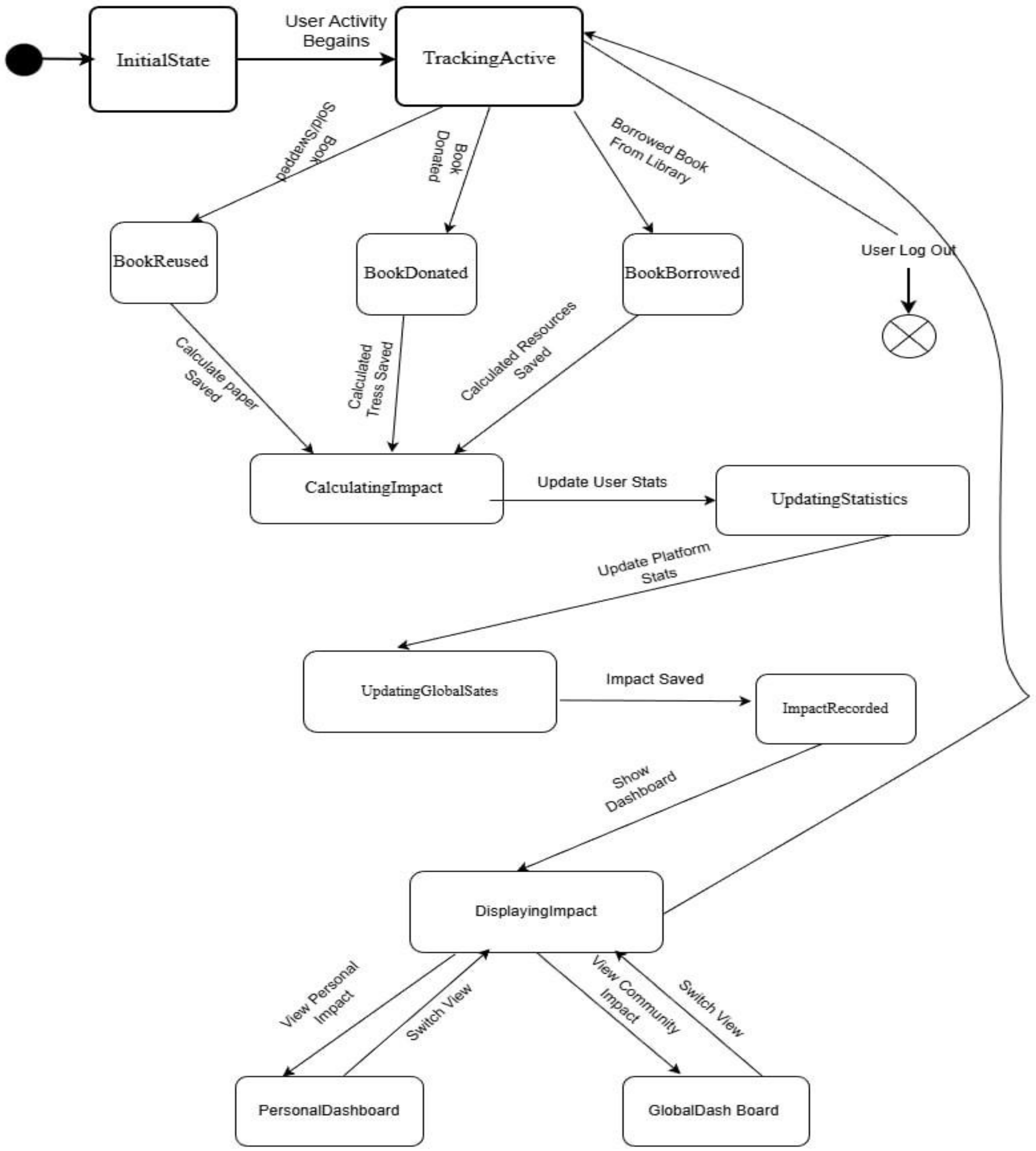


Figure: Environmental Dashboard

This flow demonstrates how user actions such as reuse, donation, and borrowing processes are monitored, computed, and figures updated and eventually, personal and global dashboard interfaces are displayed.

9.2 Sequence Diagram

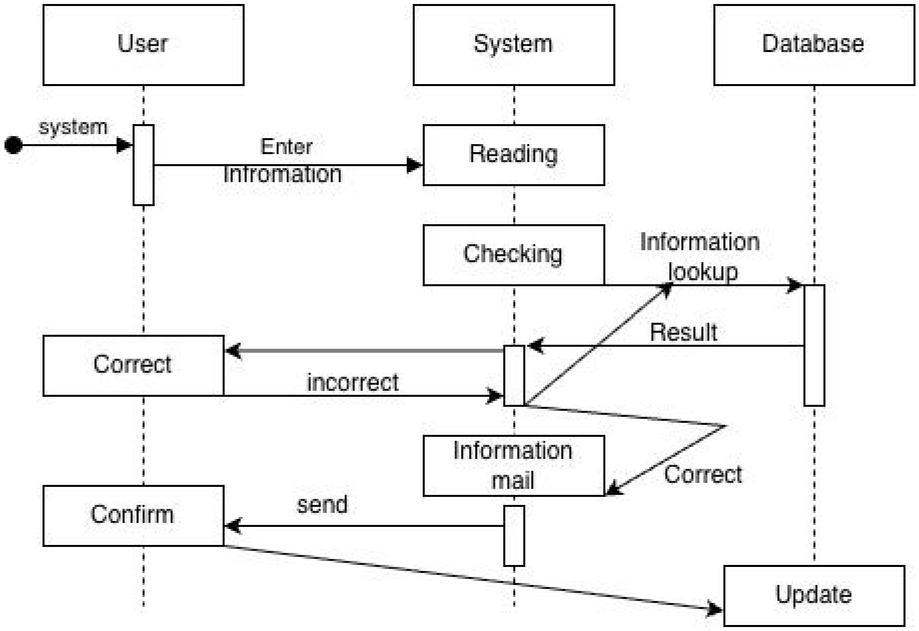


Figure: Registration

It demonstrates the procedure in which the user inputs the registration data into the system, the system checks the data against the database and approves or rejects the data before the final procedure of updating the system is complete.

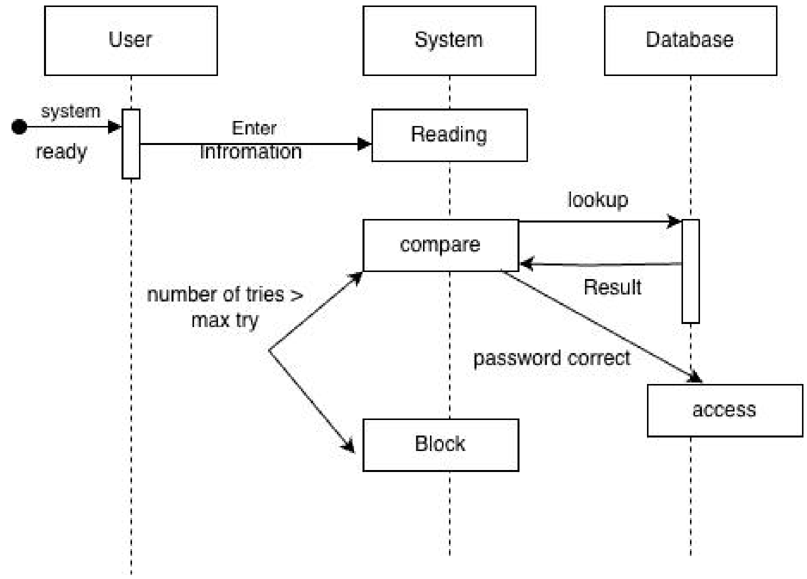


Figure: Sign In

The following sequence diagram shows how the user signs in: the system will read the credentials entered by the user, compare them to the entries in the database and either allow the user in when correct or lock out the user after a self-imposed number of attempts. It entails interactions between the User, System and Database.

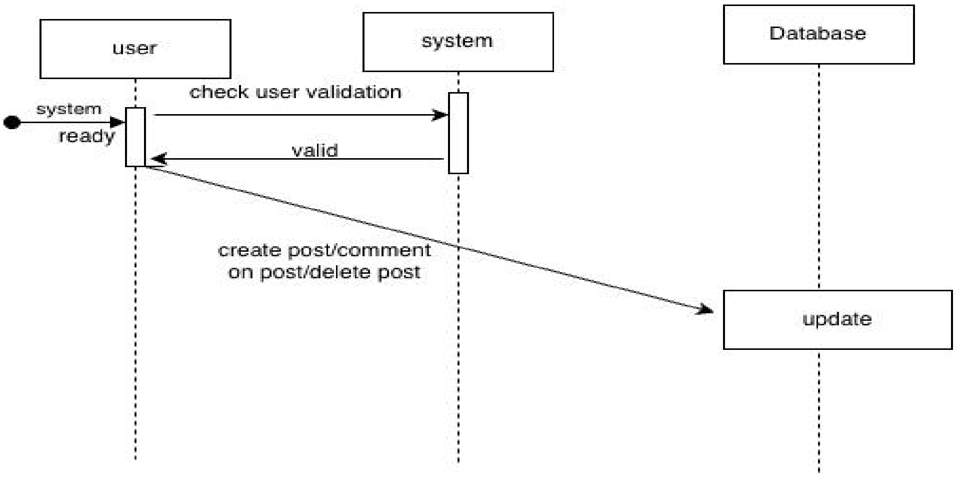


Figure: Community

It describes the interaction of a verified user with the system to create, comment or delete posts, and the updates will be reflected in the database.

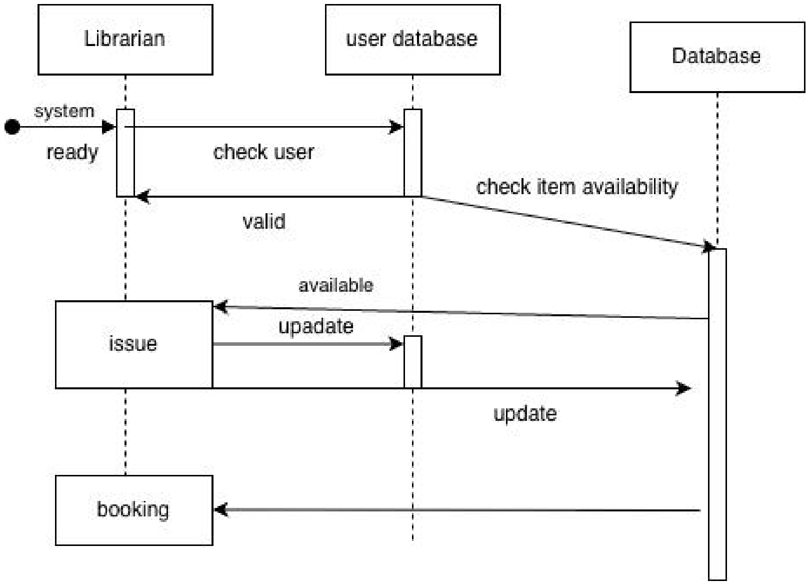


Figure: Borrow

This flow will signify the role of the librarian that involves checking availability of users and items, updating records and booking it out by the coordinated exchange of data with databases.

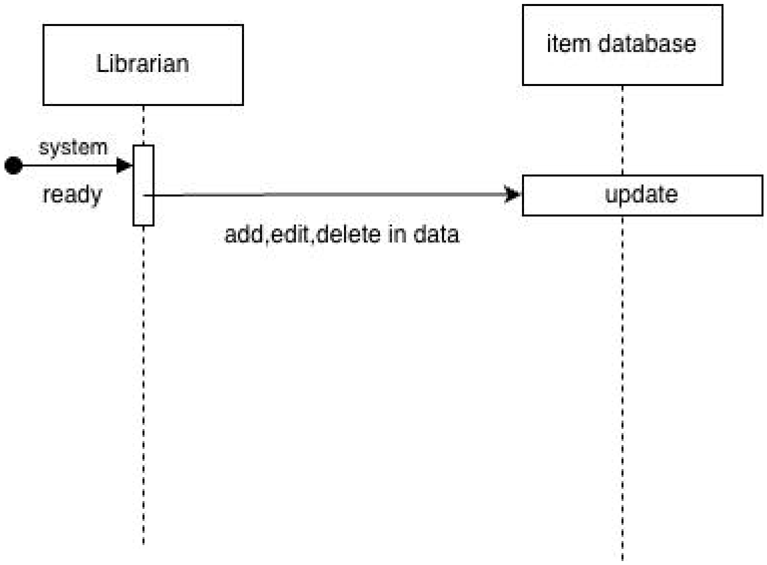


Figure: Update

It depicts the process by which the librarian is able to add, edit or delete information which gets processed by the system and will be reflected in the item database.

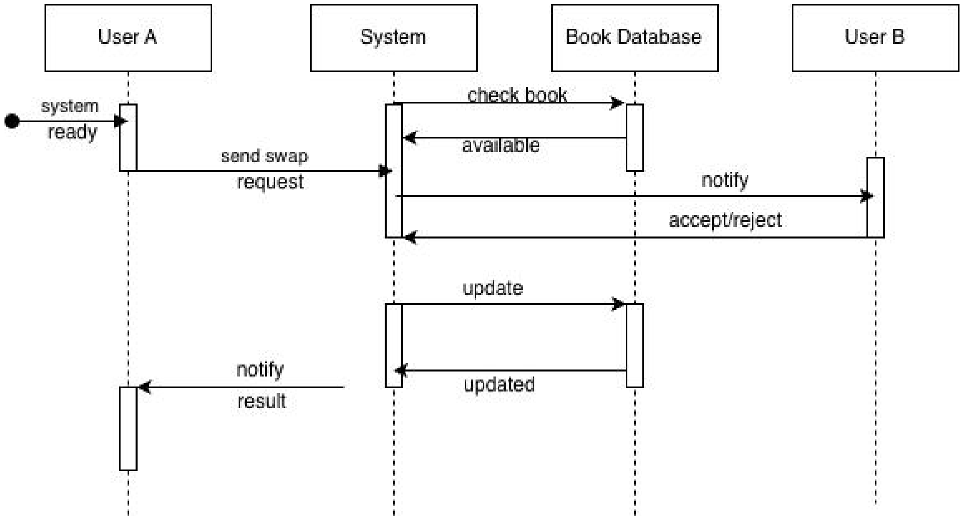


Figure: Book Swapped

This diagram illustrates how User A initiates a book swap request, the system checks availability, notifies User B, and updates the database based on User B’s response. It highlights the interaction flow between users and system components during a swap.

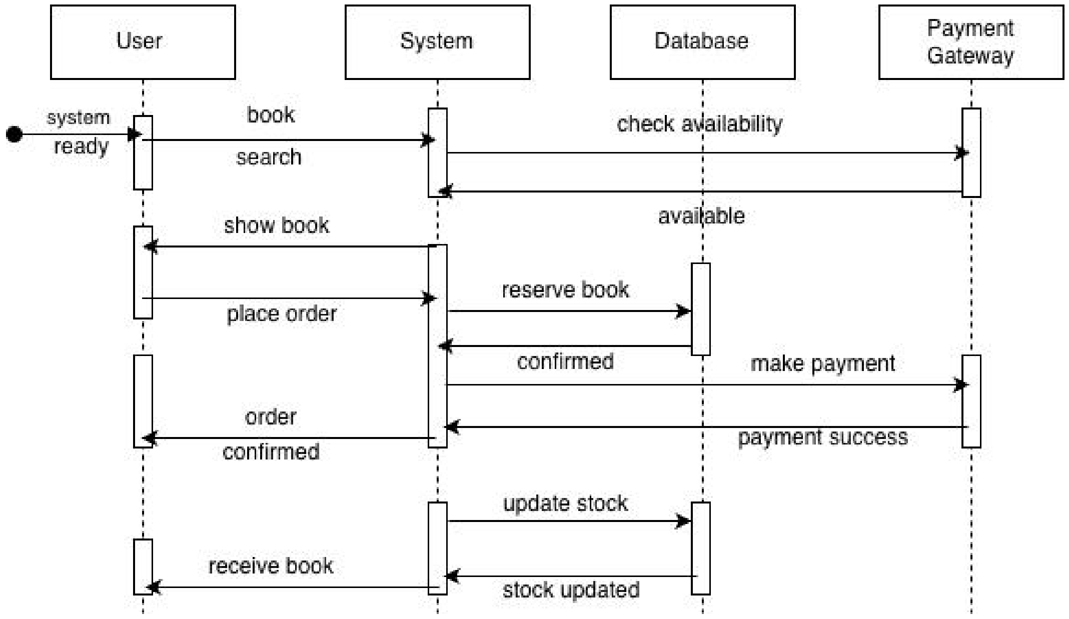


Figure: Book Selling

This diagram shows the process of selling a book, starting from the user's search to placing an order, checking availability, reserving the book, processing payment, and confirming delivery. It emphasizes coordination between the system, database, and payment gateway.