1. INTRODUCTION

- The **Online Bookstore System** is a simple and efficient software solution designed to manage and sell books through a digital platform.

- It provides a smooth experience for both **admin users** and **customers**, allowing them to interact through one centralized system.

- The system allows the **admin** to add, update, and manage book listings, including details like book title, author, category, price, description, and book cover images.

- This makes it easier to organize and maintain the entire book collection.

- Users can easily browse through all available books and use powerful **search and filter options**, such as category, author, and price range, to quickly find the books they need.

- The system ensures a fast, simple, and user-friendly book selection experience.

- Customers can view book details, add items to the cart, place orders, and track their purchases through the user panel.

- The system also includes features for handling inquiries, order management.

- The **admin panel** helps the administrator manage books, orders, users, and categories, improving overall efficiency and reducing manual work.

- By automating these processes, the Online Bookstore System reduces errors, saves time, and improves both store management and customer satisfaction.

- This project aims to offer a **secure, reliable, and easy-to-use platform** for buying and managing books online, suitable for both small bookstores and large digital book-selling businesses.

2.1 Scope

1. Book Management

- Admin can add, update, or remove books from the system.
- Details like title, author, category, price, description, and images can be managed easily.

2. User-Friendly Browsing

- Customers can search and filter books by category, author, or price.
- Helps users quickly find the books they want.

3. Shopping and Orders

- Users can view book details, add books to cart, and place orders online.
- Customers can track their orders from the system.

4. Admin Control Panel

- Admin can manage books, users, categories, role, city, feeback and orders in one place.
- Makes store management fast and organized.

5. Automation and Efficiency

- Reduces manual work and errors by automating tasks.
- Saves time and improves customer satisfaction.

6. Scalability

- Can be used by small bookstores or large online book stores.
- Secure and reliable for handling multiple users and orders.

7. Customer Support

- System can handle inquiries or messages from customers efficiently.

2.2 Objectives

1. Simplify Book Management for Admin

- Allow the admin to easily add, update, or delete books.
- Manage all book details like title, author, category, price, description, and images from one place.
- Reduce manual work and errors in maintaining the book collection.

2. Provide a Smooth Shopping Experience for Customers

- Let users browse books easily with search and filter options.
- Customers can view book details, add books to cart, and place orders online.
- Enable customers to track their orders and manage their account.

3. Improve Efficiency and Automation

- Automate repetitive tasks like order processing and inventory management.
- Save time for both admin and customers.
- Reduce mistakes caused by manual handling.

4. Enhance Customer Satisfaction

- Offer a fast, secure, and user-friendly platform.
- Make it easy to find, order, and receive books.
- Handle customer inquiries efficiently.

5. Support Business Growth

- Suitable for small bookstores and large online book stores.
- Allow scalability for adding more books, users, and categories.
- Enable admins to manage the store more effectively as it grows.

2.3 Constraints

2.3.1 H/W (Hardware) Constraints

Hardware constraints refer to the physical components of the system that limit how the software can run efficiently. For the Online Bookstore System, the possible hardware constraints are:

1. **Server Requirements**

   - The server hosting the system must have enough storage and memory to handle book data, images, and user requests.
   - Example: Minimum 8 GB RAM and 500 GB storage for smooth performance.

2. **Client Device Requirements**

   - Users (customers/admin) should access the system from devices that support modern web browsers.
   - Example: Desktop, laptop, tablet, or smartphone with a stable internet connection.

3. **Processing Power**

   - Both server and client devices should have sufficient CPU performance to process requests quickly.
   - If the hardware is weak, the system may be slow or unresponsive.

4. **Network Dependency**

   - The system requires a stable internet connection for real-time access, searching books, and placing orders.
   - Poor network speed can affect system performance and user experience.

2.3.2 S/W (Software) Constraints

Software constraints refer to the limitations imposed by the software environment needed to run the system. For the Online Bookstore System, the constraints are:

1. **Operating System Compatibility**

   - The system must run on servers with supported OS like Windows Server or Linux.
   - Client devices should support browsers like Chrome, Firefox, Edge, or Safari.

2. **Database Requirements**

   - The system relies on a database (like SQL Server, MySQL, or PostgreSQL) to store book and user data.
   - Database size, structure, and query efficiency can limit system performance if not properly managed.

3. **Browser Compatibility**

   - Web application must be tested to work correctly on all modern browsers.
   - Older browsers may not support some features like responsive design or dynamic scripts.

4. **Security Software**

   - Proper firewall and antivirus software are needed to prevent unauthorized access.
   - Security constraints may limit some system actions to protect sensitive data (like user credentials and payment info).

5. **Software Dependencies**

   - The system may require additional software or frameworks like .NET, Angular, or PHP depending on the technology used.
   - Incompatibility or missing dependencies can prevent the system from running properly.

2.4 Advantages

1. **Easy Book Management**

   - Admin can add, update, or remove books quickly without manual record-keeping.
   - Reduces errors in maintaining the book collection.

2. **Convenient for Customers**

   - Users can search, filter, and browse books easily.
   - Books can be purchased from anywhere, anytime using the web.

3. **Time-Saving and Efficient**

   - Automation of order processing and inventory management saves time for both admin and customers.
   - Reduces repetitive manual tasks.

4. **Enhanced Customer Satisfaction**

   - Fast, secure, and user-friendly platform ensures a smooth shopping experience.
   - Features like cart, order tracking, and home delivery improve convenience.

5. **Scalable and Flexible**

   - Can be used by small bookstores or large online shops.
   - Supports adding new books, users, and categories easily.

6. **Better Business Control**

   - Admin panel allows managing books, users, orders, categories, roles, cities, and feedback in one place.
   - Helps in better reporting, auditing, and store management.

7. **Customer Support and Communication**

   - Handles customer inquiries and feedback efficiently.
   - Email notifications and order updates improve communication.

2.5 Limitations

1. **Internet Dependency**

- System requires a stable internet connection to access and use.
- Poor connectivity may slow down or prevent access.

2. **Initial Setup Cost**

- Setting up the system may require hardware, software, and server costs.

3. **Technical Knowledge Required**

- Admin needs basic computer and software knowledge to manage the system efficiently.

4. **Browser and Device Compatibility**

- Some older browsers or devices may not support all features properly.

5. **Limited Offline Functionality**

- Cannot perform tasks like order placement or inventory management without internet.

6. **Security Risks**

- As with any online system, there is a risk of hacking or data breaches if security measures are weak.

3.1 Hardware and Software Requirements

➢ Hardware Requirements :-

Processor	Intel Core i5
RAM	8 GB
Hard Disk	51.4 GB
Disk Space	118 GB Minimum

Software Requirements :-

Operating System	Microsoft Windows 11
Frontend	Net bean-24
Backend	Net bean-24
DataBase	MySQL v- 8.0.43

1.4 Project Profile[puchvanu]

Project Title	Online Book Store
Front-End	Java Server Faces
Back-End	MySQL 8.0.43
Submitted By	Ms.Patel Khushi Manojbhai Ms.Udhanawala Yesha Hirenkumar

3.2 Development Tools

The Online Bookstore System was developed using a combination of frontend, backend, and database tools. These tools helped to build a secure, scalable, and efficient web application. The following development tools and technologies were used:

1. Integrated Development Environment (IDE)

NetBeans IDE

- The entire project (frontend + backend) was developed using NetBeans, a powerful Java-based IDE.
- NetBeans provides features such as code completion, debugging, project management, and integration with application servers.
- It supports Java EE technologies like JSF, EJB, and JPA, making it suitable for enterprise-level applications.

2. Frontend Technologies

JSF (JavaServer Faces)

- JSF was used to create the user interface for both customer and admin modules.
- Provides reusable components for forms, buttons, tables, and input fields.
- Helps create a clean MVC-based structure, making the UI more organized.
- Supports backing beans for connecting UI with backend logic.

3. Backend Technologies

EJB (Enterprise JavaBeans)

- EJB was used for handling business logic of the application.
- Useful for tasks like:
- Processing orders
- Handling user authentication
- Managing books and categories
- Managing cart and feedback
- Provides security, transaction management, and scalability.

RESTful Web Services (JAX-RS)

- REST APIs were created to handle communication between frontend and backend.
- These APIs support operations like:
- Fetching book details
- Adding items to cart
- Placing orders
- Retrieving feedback and user data
- Provides lightweight and fast communication using JSON.

4. Database Technology

MySQL Database

- MySQL was used as the main database to store all application data such as:
- Books
- Users
- Categories
- Orders
- Order items
- Feedback
- MySQL Workbench was used to design and manage the database.

JPA (Java Persistence API)

- JPA was used for Object–Relational Mapping (ORM).
- Entities were created to map Java classes to database tables.
- Simplifies database operations like insert, update, delete, and select.
- Makes the data layer clean and easy to maintain.

5. Application Server

GlassFish / WildFly (if used)

- The project can be deployed on an enterprise application server like GlassFish or WildFly.
- Supports Java EE technologies such as EJB, JSF, and JPA.

6. API Testing Tools

Postman

- Used to test REST APIs for correct request–response handling.

4.1 Feasibility Study

Feasibility study evaluates whether the Online book store system is practical and possible to develop. It includes several dimensions:

1. Technical Feasibility

- Checks if the required technology (JSF, EJB, REST, Client,MySQL, NetBeans) is available.
- Confirms that your team can develop the system with existing tools.

2. Operational Feasibility

- Checks if users (admin, customers) can use the system easily.
- Confirms that the system solves the real problem (bookstore automation).

3. Economic Feasibility

- Analyzes cost vs. benefits.
- Cheaper than maintaining manual records.
- Helps increase sales and customer satisfaction.

4. Time Feasibility

- Confirms that the project can be completed within the available time.

4.2 Software Engineering Model

For this project, the **Iterative Waterfall Model** (or Modified Waterfall) is suitable.

Why Iterative Waterfall?

- Requirements were clear from the beginning.
- Structured phase-by-phase development.

Phases Used :-

1. Requirement Analysis

- Gathering all requirements for the Book Store System.
- Understanding user roles, workflows, book management needs, and client-server interactions.

2. System Design

- Designing the UI structure for JSF pages.
- Creating ER Diagrams for the MySQL database.
- Preparing Class Diagrams for EJB and JPA entities.
- Designing RESTful API structure.

3. Implementation

- Developing front-end using JSF.
- Writing business logic using EJB.
- Building REST API services.
- Creating entities using JPA.
- Using NetBeans IDE for development.
- Integrating the system with MySQL database.

4. Testing

- Performing Unit Testing of EJB classes.
- Testing REST APIs using tools.
- Manual testing of JSF pages and workflows.
- Fixing issues and refining modules.

5. Deployment

- Deploying the application on GlassFish/Payara Server.
- Configuring database resources and application settings.

6. Maintenance

- Fixing bugs after deployment.
- Improving system efficiency and adding additional features.
- Updating database or UI based on feedback.

4.3 Risk Analysis

This section lists the possible risks during the project and how you will handle them.

Risk	Type	Impact	Mitigation (Solution / Preventive Action)
JSF / EJB configuration errors	Technical Risk	High	Proper configuration, follow setup documentation, test each module separately
API communication failure between JSF, EJB, REST	Technical Risk	High	Test APIs using Postman, enable logging, ensure correct JSON parsing
Database connection failure (MySQL)	Technical Risk	High	Use stable connection pool, add error handling, monitor DB server status
Users not understanding the UI	Operational Risk	Medium	Create simple UI, add tooltips, provide user instructions
Miscommunication between developers	Operational Risk	Medium	Conduct regular meetings, maintain documentation, use project tracking tools
Data leakage (sensitive user/book/order data)	Security Risk	High	Apply HTTPS, secure API endpoints, restrict admin/user privileges
SQL Injection attacks	Security Risk	High	Use prepared statements, JPA parameter binding, validate inputs
Delay in development or testing	Schedule Risk	Medium	Weekly progress review, assign tasks clearly, maintain timeline
Team member absence	Schedule Risk	Medium	Distribute responsibilities, maintain documentation so others can continue work
Lack of tools or required hardware	Financial/Resource Risk	Medium	Use open-source tools (NetBeans, MySQL), ensure backups, request tools early

4.4 Project Table

4.4.1 Task Dependency

Task	Depends on
Requirement Analysis	-
Designing	Requirements
Database Creation	Design
Backend Development	Database
Frontend Pages	Backend APIs
Testing	Backend + Frontend
Deployment	Testing

4.4.2 Timeline Chart

Task	Week 1	Week 2	Week 3	Week 4	Week 5
Requirements	■				
Design		■			
Database			■		
Backend Dev			■		
Frontend Dev				■	
Testing					■

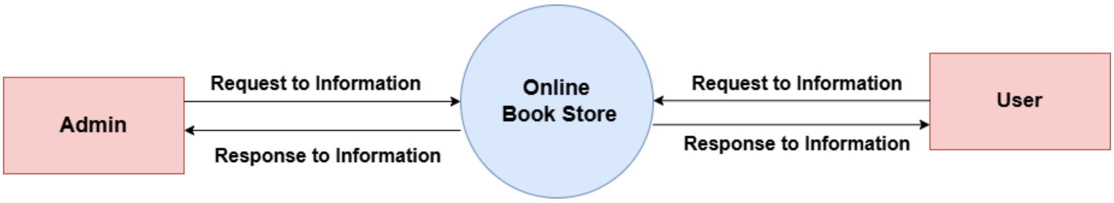
4.4.3 Project Table[baki]

Task	Description	Start Date	End Date	Status
Requirement Collection	Understanding features	01-01-2025	05-01-2025	Completed
UI Design	Designing JSF pages	06-01-2025	10-01-2025	Completed
Backend Dev	EJB + REST APIs	11-01-2025	25-01-2025	In Progress
Testing	Checking errors	26-01-2025	30-01-2025	Pending

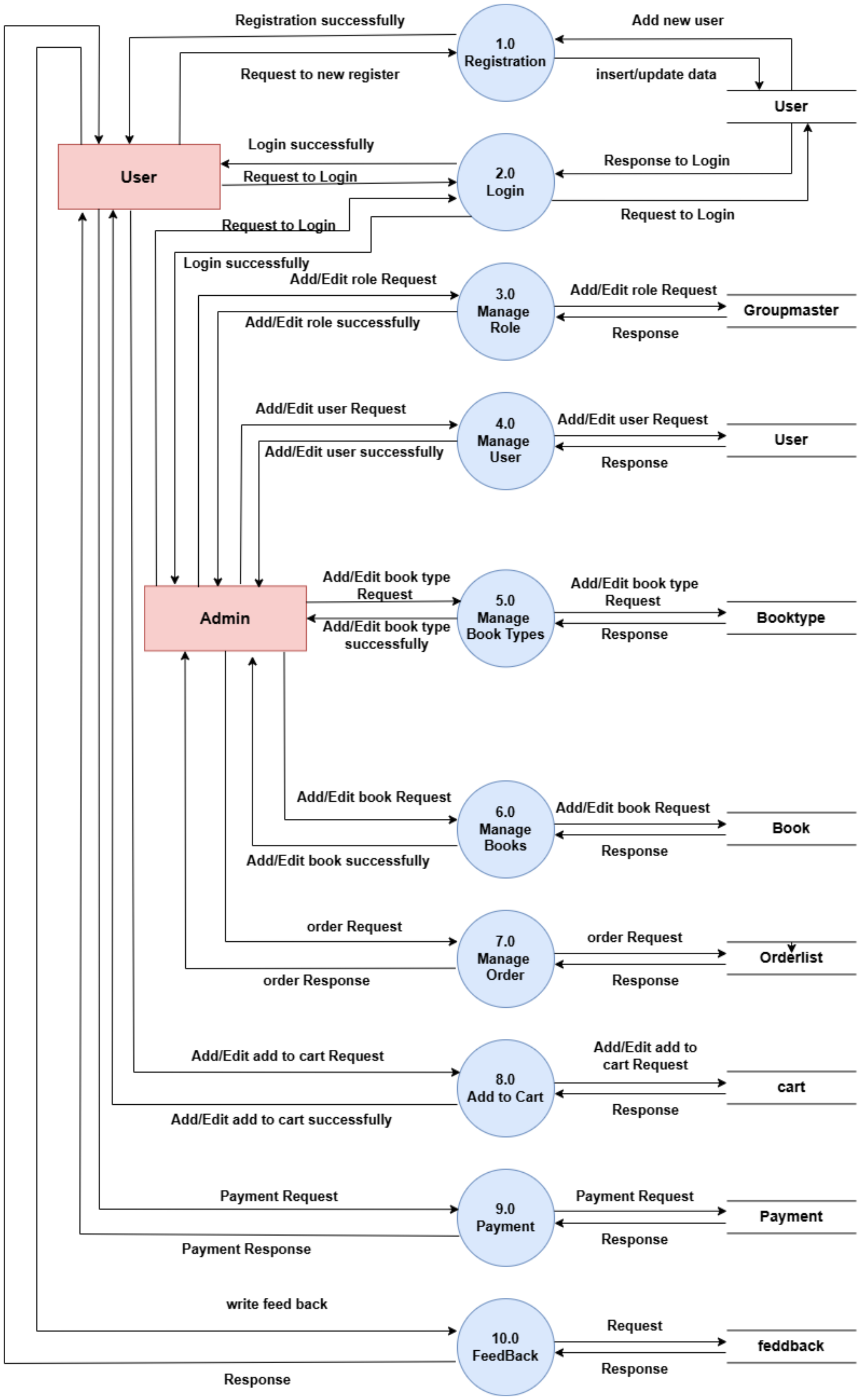
5. Detail Planning

5.1 Data Flow Diagram

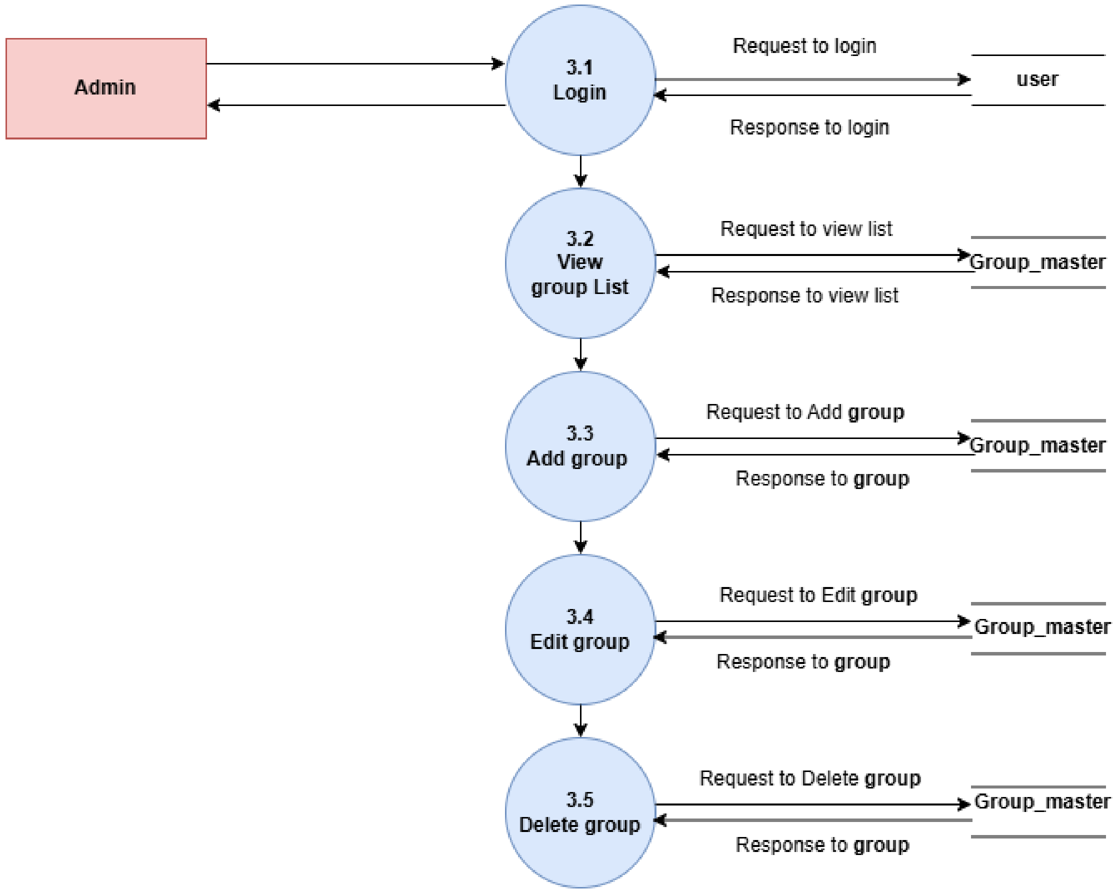
- Online Book store: **Zero level DFD**



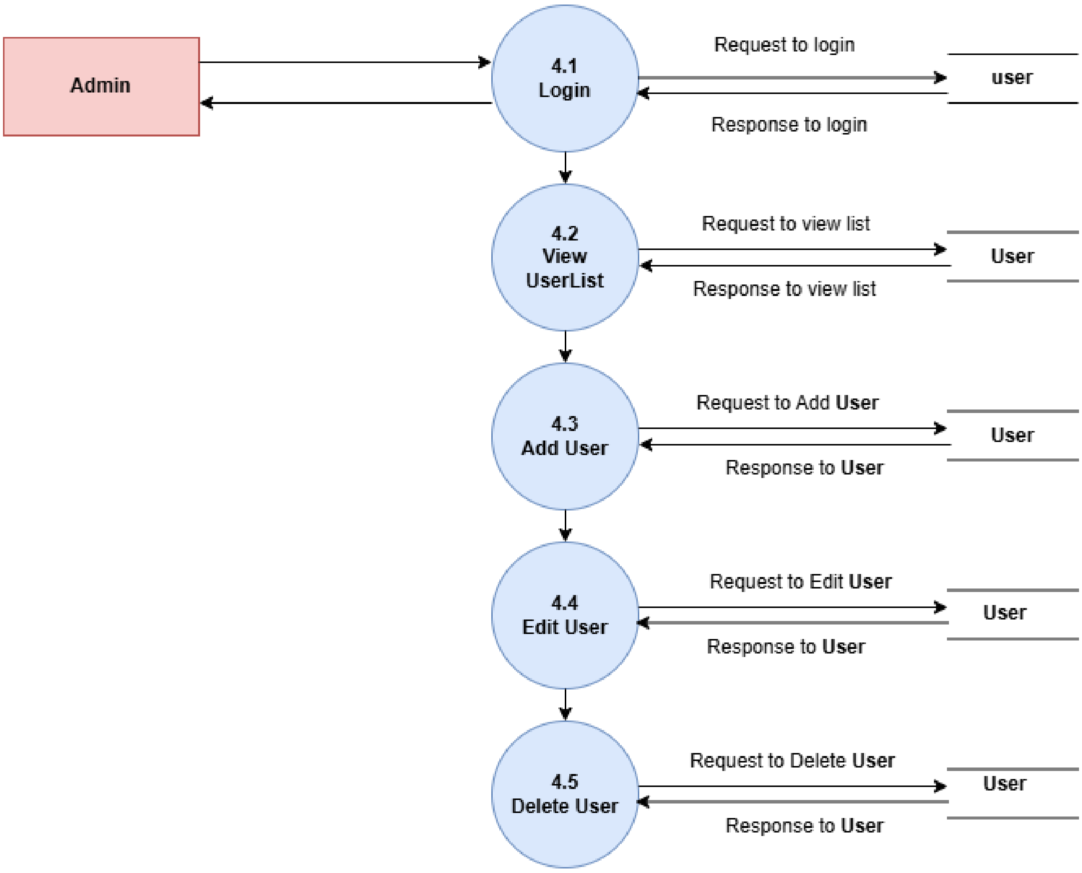
➢ One level DFD



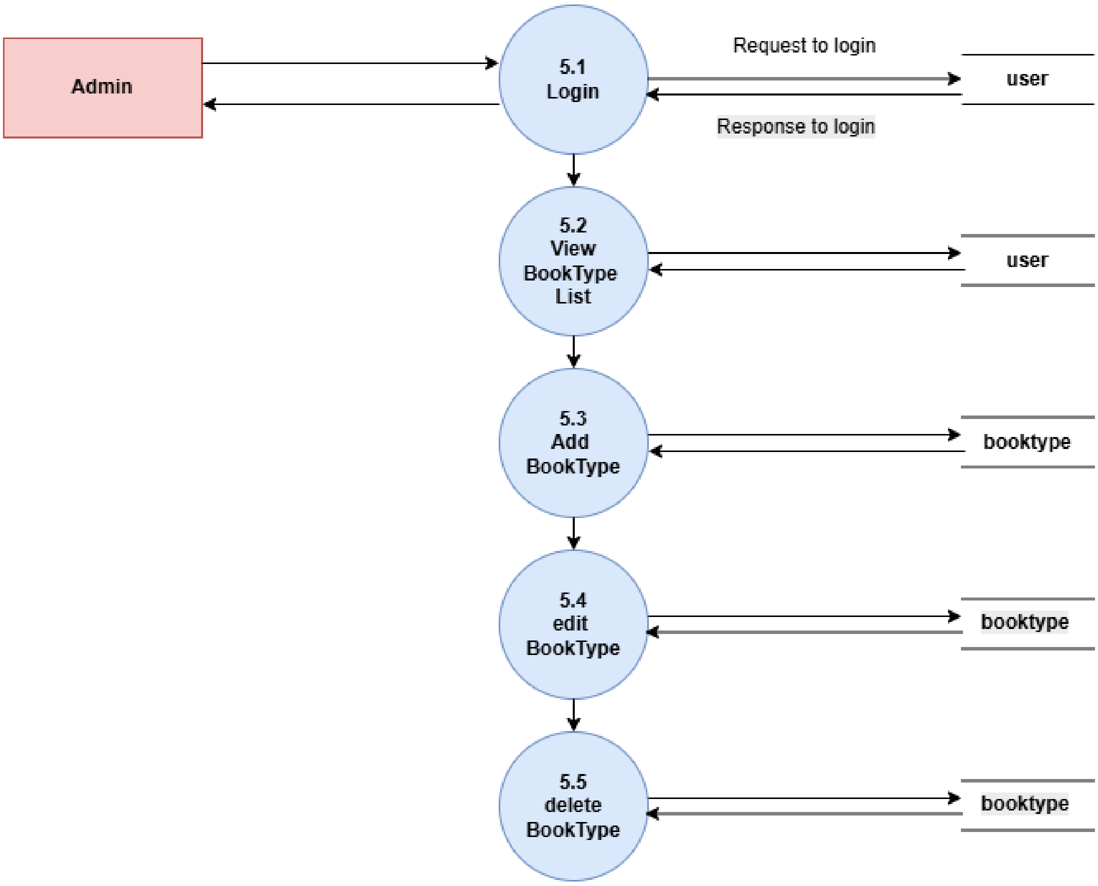
➢ Admin 2 level Groupmaster



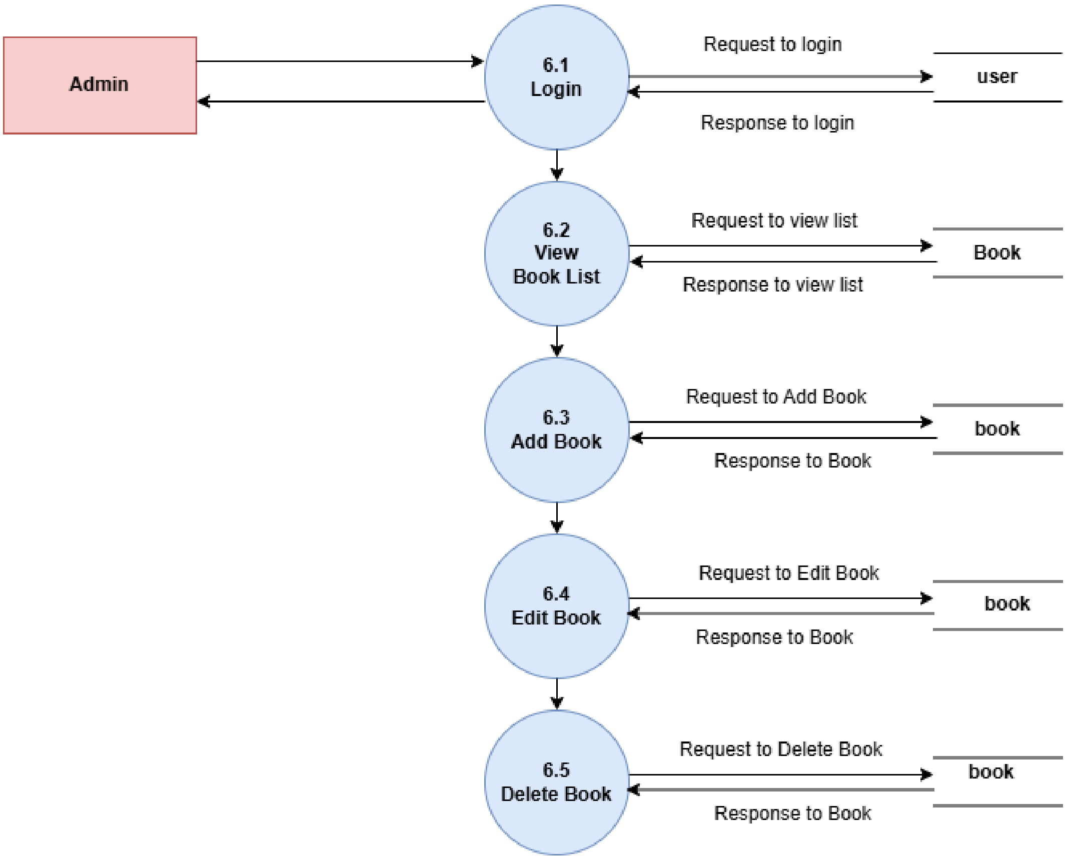
➢ Admin : 2 level User Management



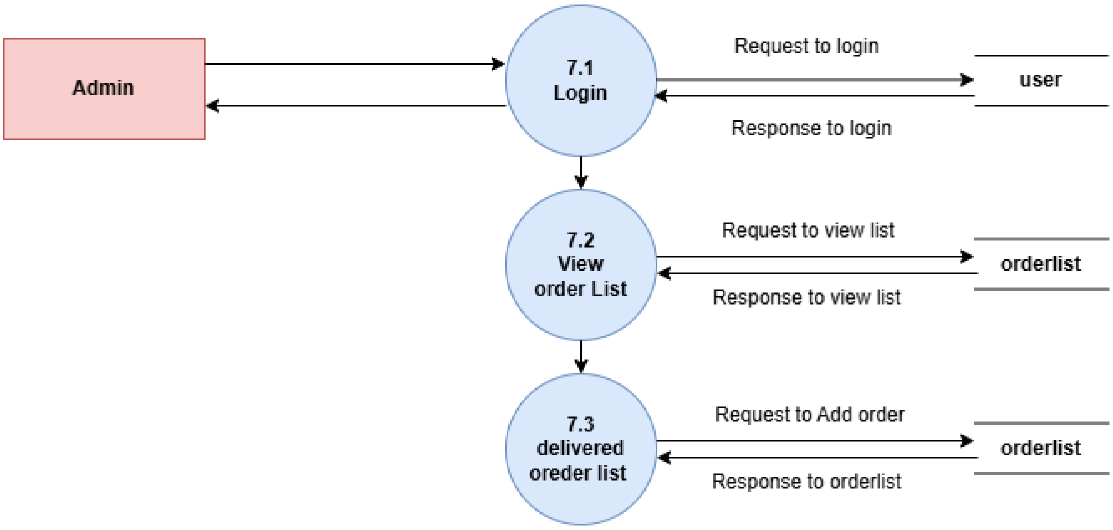
➢ Admin : 2 level Book Type Management



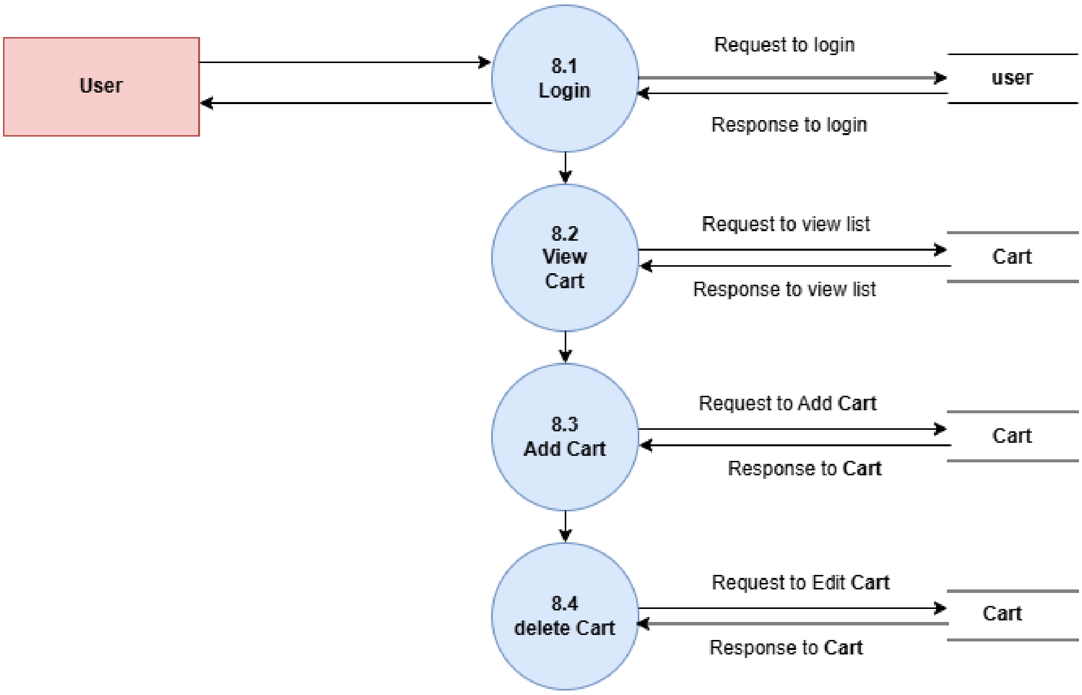
- **Admin : 2 level Books Management**



➢ Admin : 2 level Order Management

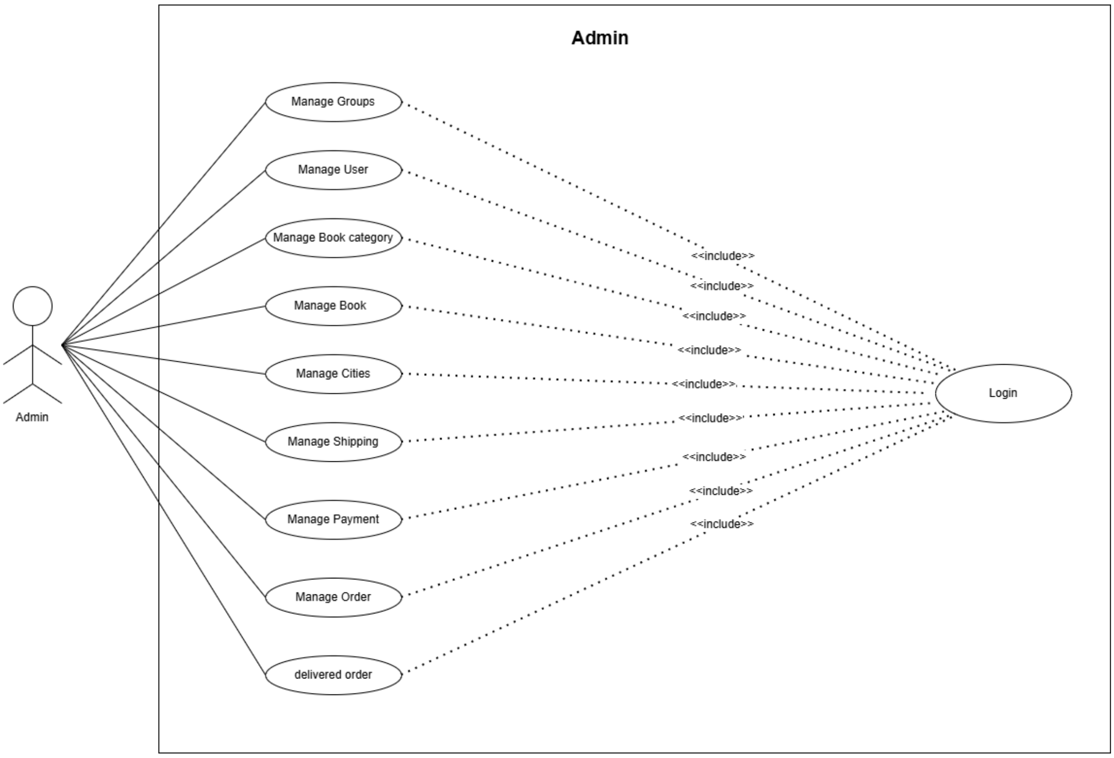


➢ User: 2 level Add to cart Management

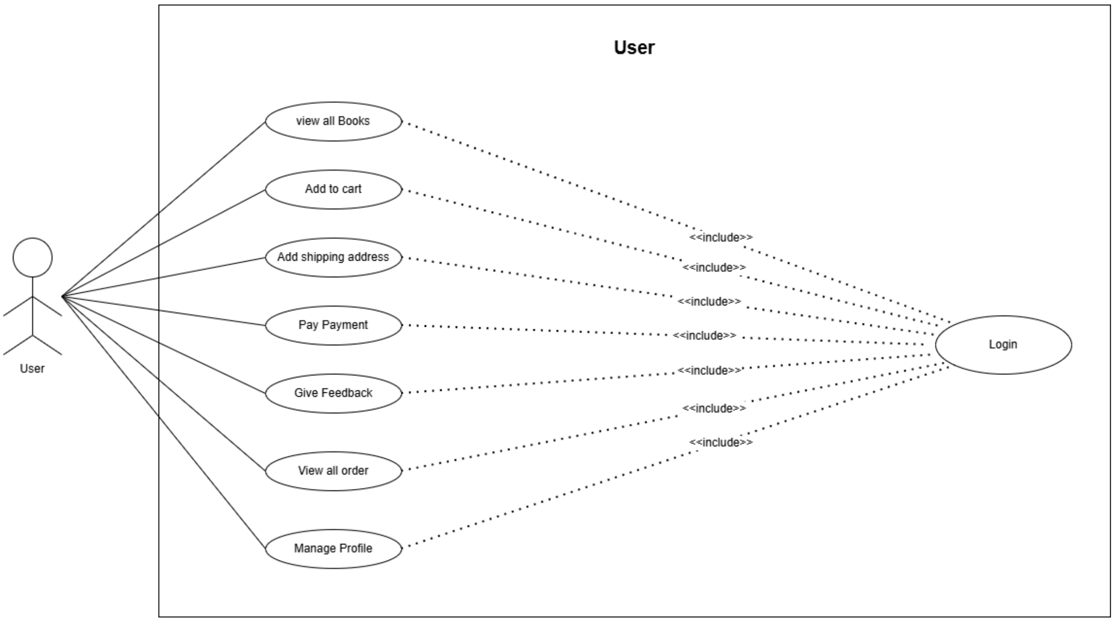


5.2 UML Use Case Diagram

➢ Admin

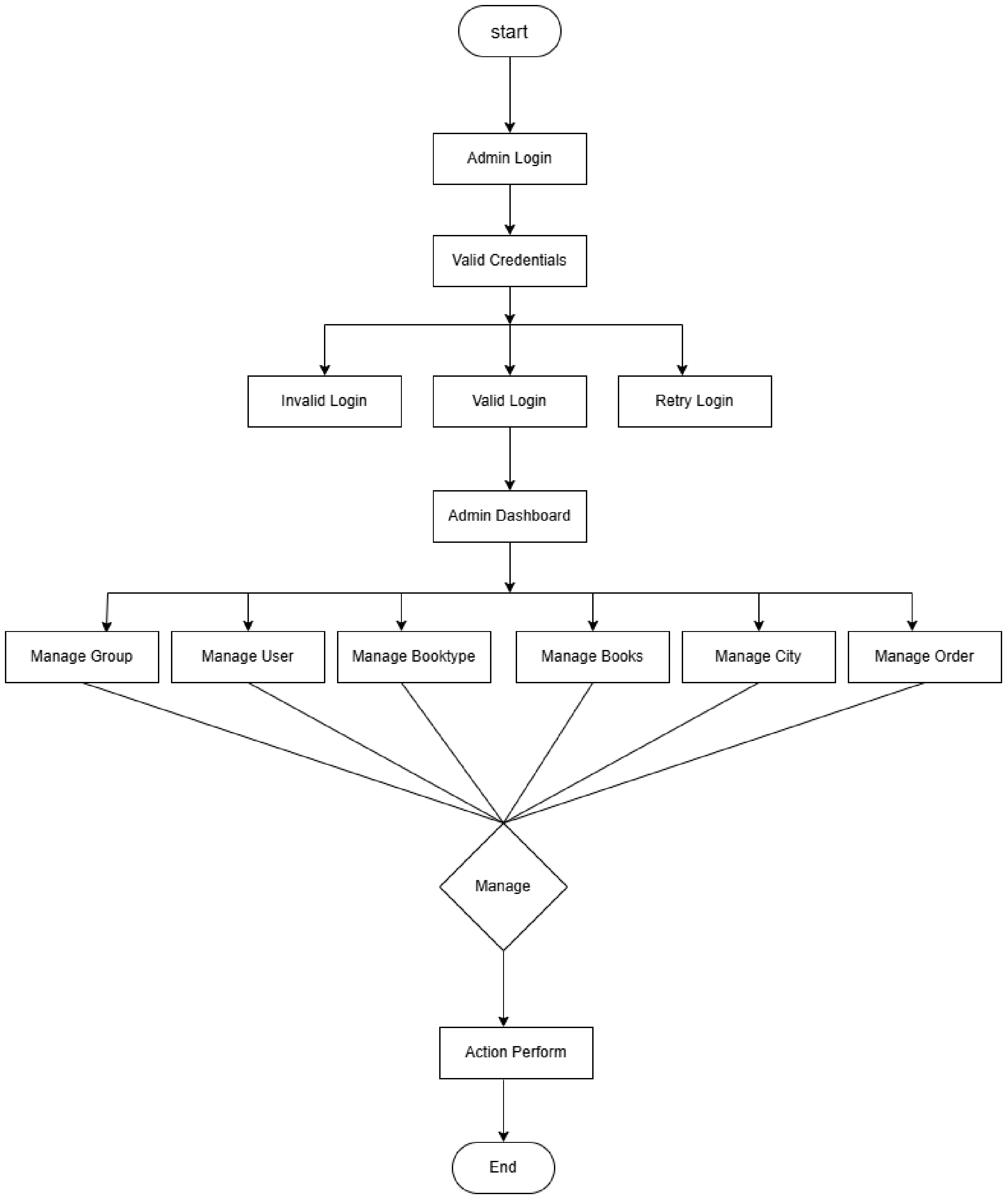


➢ User

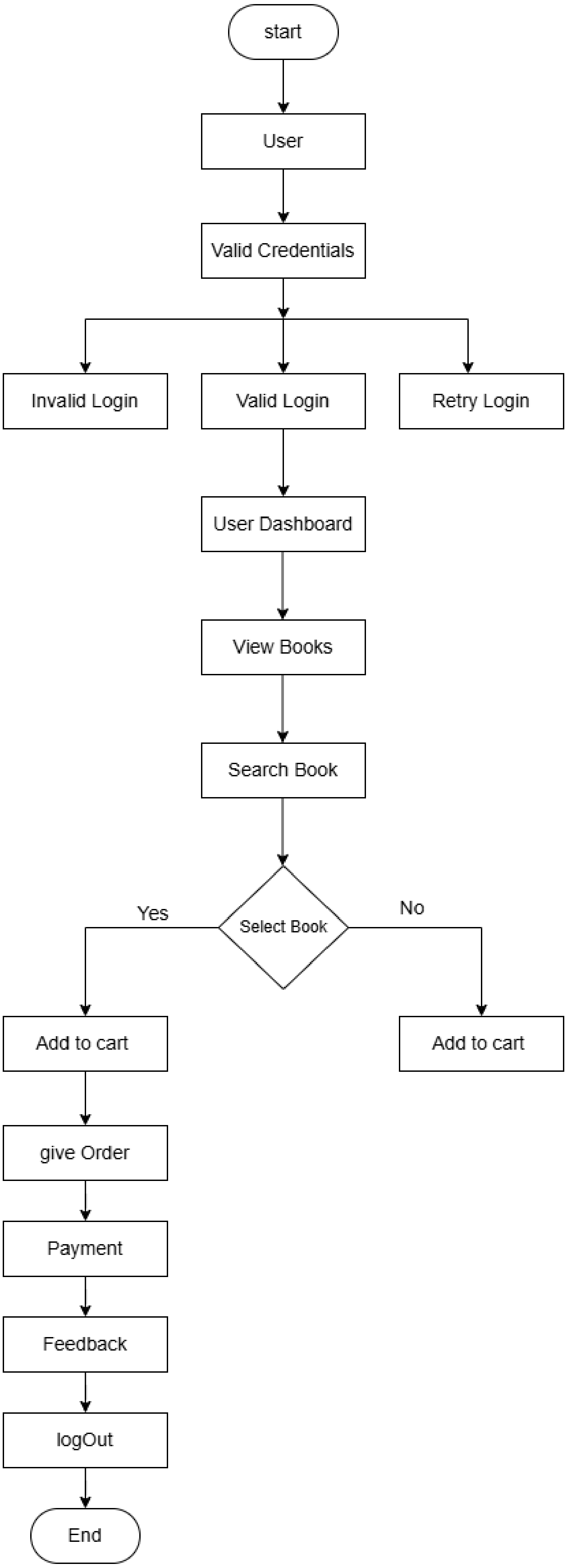


5.3 UML Activity Diagram

- **Admin**

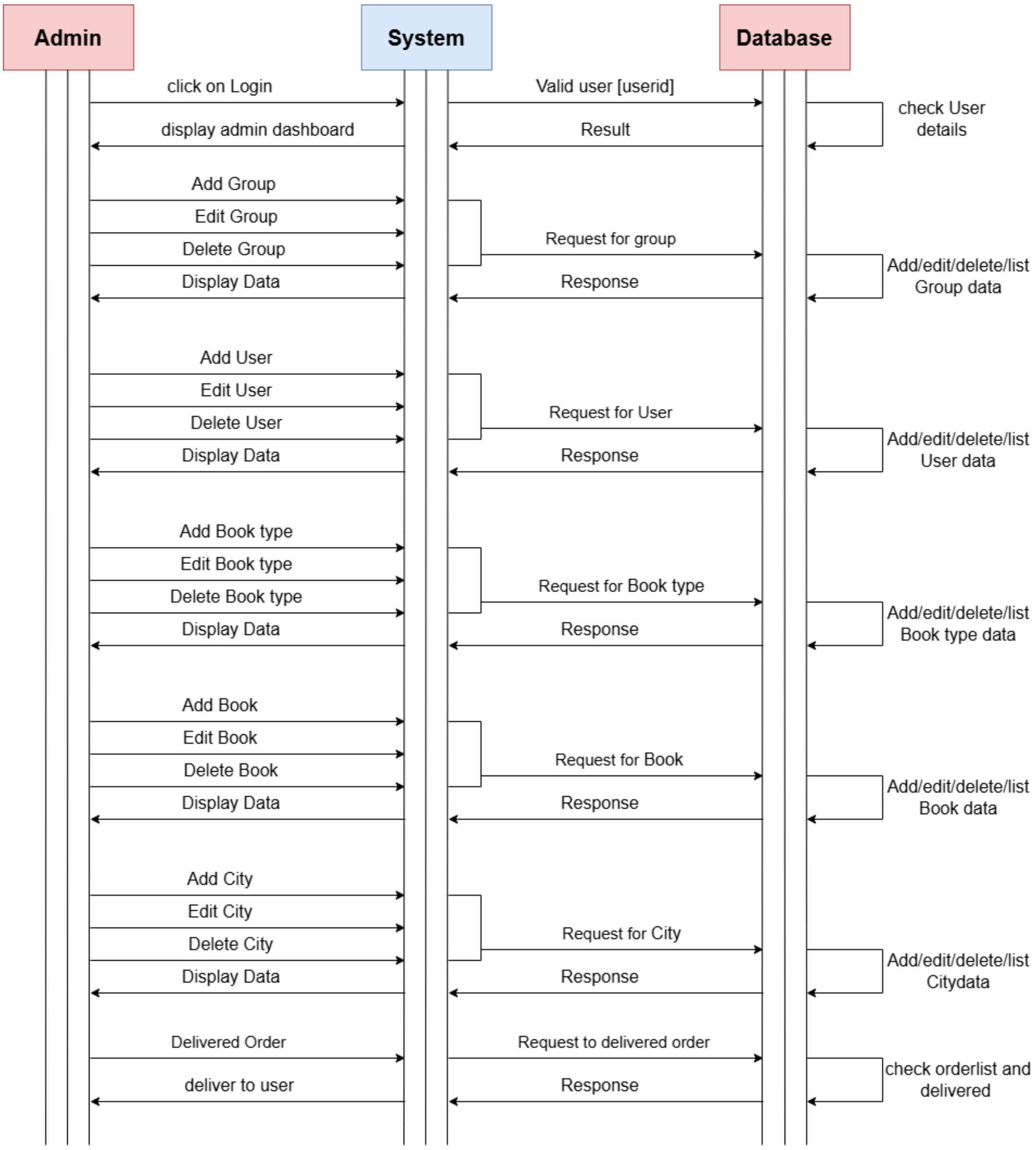


➢ User

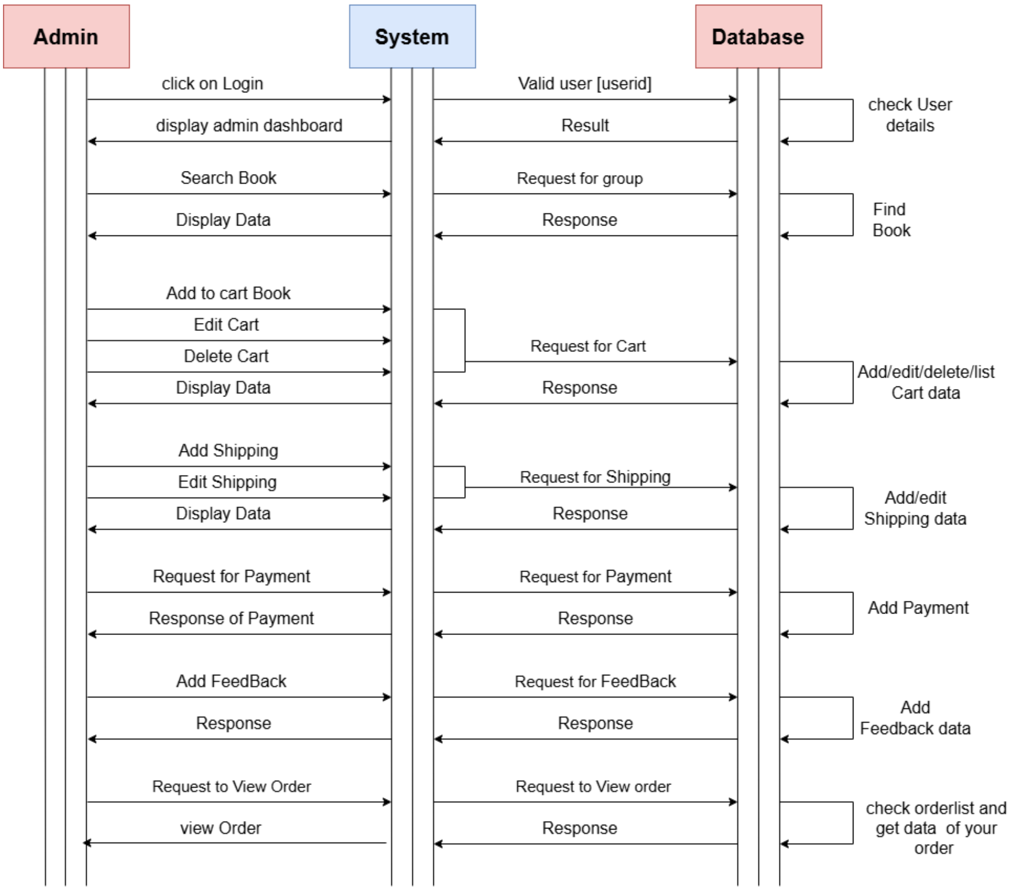

start
User
Valid Credentials
Invalid Login
Valid Login
Retry Login
User Dashboard
View Books
Search Book
Select Book
Yes
No
Add to cart
Add to cart
give Order
Payment
Feedback
logOut
End

5.4 Sequence Diagram

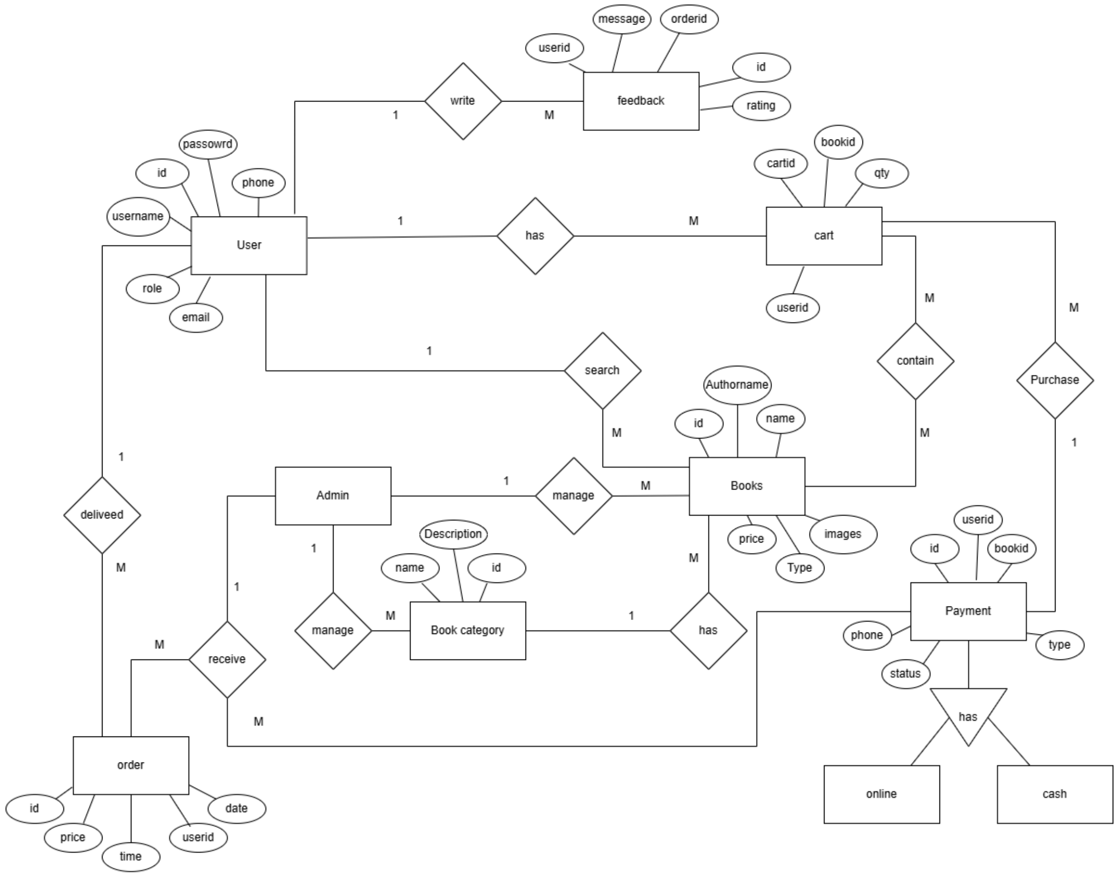
➢ Admin



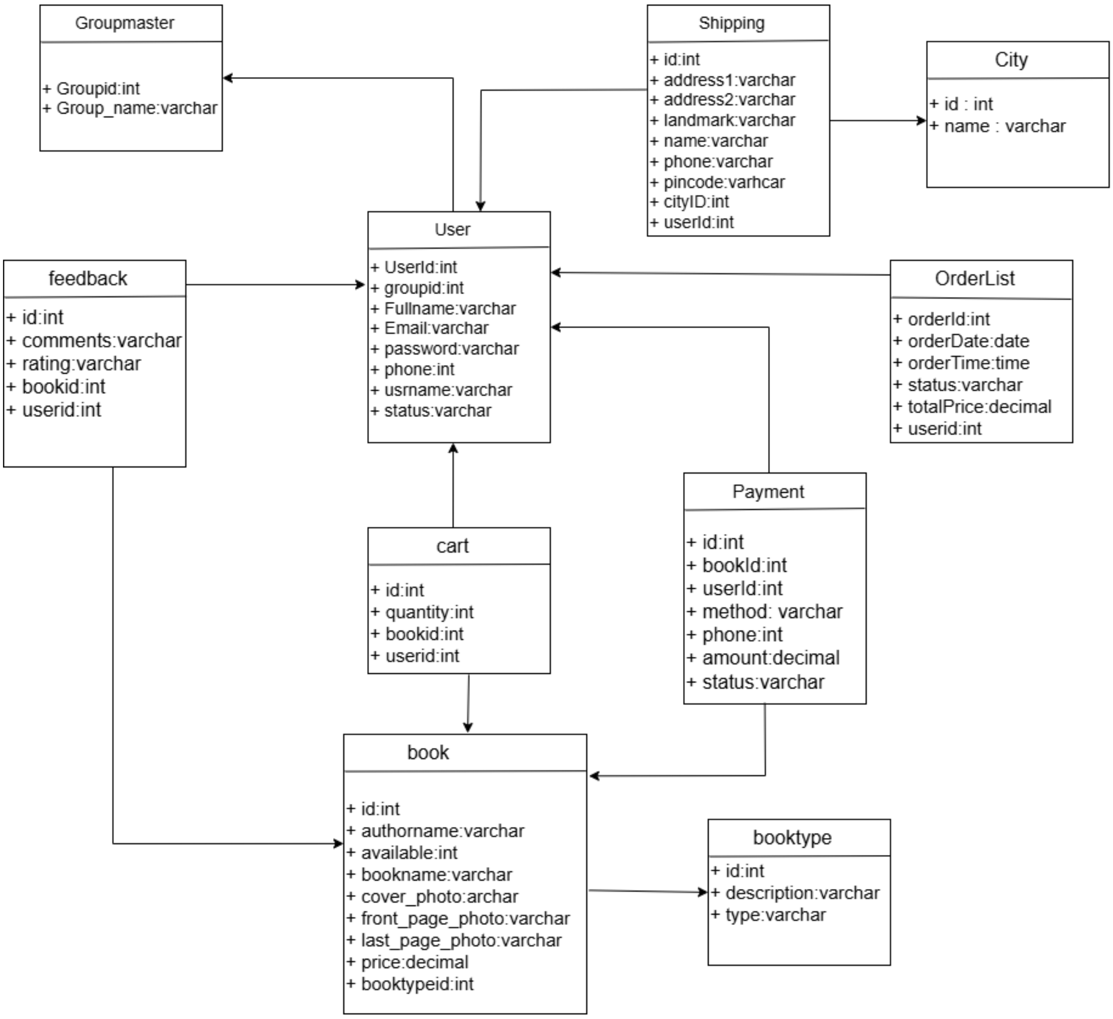
➢ User


Admin
System
Database
click on Login
display admin dashboard
Valid user [userid]
Result
check User details
Search Book
Display Data
Request for group
Response
Find Book
Add to cart Book
Edit Cart
Delete Cart
Display Data
Request for Cart
Response
Add/edit/delete/list Cart data
Add Shipping
Edit Shipping
Display Data
Request for Shipping
Response
Add/edit Shipping data
Request for Payment
Response of Payment
Request for Payment
Response
Add Payment
Add FeedBack
Response
Request for FeedBack
Response
Add Feedback data
Request to View Order
view Order
Request to View order
Response
check orderlist and get data of your order

5.5 ER Diagram



5.6 Table Relation



5.7 Data Dictionary

- **Group_master**

Column_name	Data Type	Key	Length
Group_id	int	Primary key	-
Group_name	varchar	-	255

- **User:-**

Column_name	Data Type	Constrants	Length
id	int	Primary key	-
Group_id	int	Foreign key	-
Fullname	varchar	-	255
phone	varchar	-	10
Username	varchar	-	255
Password	varchar	-	255
Email	varchar	-	255
Status	varchar	-	255

- **BookType:-**

Column_name	Data Type	Constrants	Length
id	int	Primary key	-
Type	varchar	-	255
Description	varchar	-	255

- **Book**

Column_name	Data Type	Constrants	Length
id	int	Primary key	-
Booktype	int	Foreign key	-
Book_name	varchar	-	255
Autho_rname	varchar	-	255
Price	decimal	-	38,0
Available	tinyint	-	1
Cover_photo	varchar	-	255
front_page_photo	varchar	-	255
last_page_photo	varchar	-	255

- **Cart**

Column_name	Data Type	Constrants	Length
id	int	Primary key	-
Userid	int	Foreign key	-
Bookid	int	Foreign key	-
Quantity	int	-	-

- **City**

Column_name	Data Type	Constrants	Length
id	int	Primary key	-
Name	varchar		255

- **Orderlist:-**

Column_name	Data Type	Constrants	Length
Orderid	int	Primary key	-
Userid	int	Foreign key	-
Time	time	-	-
date	date	-	-
Status	varchar	-	255
Total_price	decimal	-	38,0

- **Shipping**

Column_name	Data Type	Constrants	Length
id	int	Primary key	-
Userid	int	Foreign key	-
cityid	int	Foreign key	-
Name	varchar	-	255
phone	int	-	10
Address1	varchar	-	255
Address2	varchar	-	255
Landmark	vachar	-	255
Pincode	int	-	10

- **Payment:-**

Column_name	Data Type	Constrants	Length
id	int	Primary key	-
Bookid	int	Foreign key	-
Userid	int	Foreign key	-
Paymentmethod	varchar	-	255
Phone	int	-	10
Status	decimal	-	38,0

- **Feedback**

Column_name	**Data Type**	**Constrants**	**Length**
id	int	Primary key	-
Userid	int	Foreign key	-
Bookid	int	Foreign key	-
Quantity	int	-	-
Rating	int	-	-
Comments	varchar	-	

6. Input Design

7. System Testing

- **Login**

Sr no	Description	Input value		Expected Result	Actual Result	Paas/ Fail
		usernam	password			
1.	Check for validation	john123	john	Redirect to admin dashboard	Redirect to admin dashboard	Pass
2.	Check for validation	John123	Null	Error:password is required.	Error: password is required.	Fail
3.	Check for validation	Null	Null	Error:Both required message	Error:Both required message	Fail

- **Registration**

Sr No.	Description	Input Values	Expected Result	Actual Result	Pass/Fail
1	Check input detail validation	If all fields are filled properly and clicked on sign up button	Redirect to Login Page	Redirect to Login Page	Pass
2	Password validation	Check for password strength. Must contain 1 alphabet, 1 number and 1 special character	Enter Valid Password	Enter Valid Password	Fail
3	Email validation	Check for valid email.	Enter Valid Email	Enter Valid Email	Fail
4	Check for validation	If all details are not filled	Error: Please enter details	Error: Please enter details	Fail

- **Book Type**

Sr No.	Description	Input Values	Expected Result	Actual Result	Pass/Fail
1	Check input detail validation	If all fields are filled properly and clicked on save button	Redirect to Show group List page	Redirect to Show group List Page	Pass
2	Book type name	If null booktype name	Error : book type is required.	Error : book type is required.	Fail
3	Description	Check for Description	Enter Description	Enter Description	Pass

8. Future Enhancement

- **Mobile Application Integration**
  A dedicated Android/iOS mobile app can be developed so users can browse and purchase books conveniently from their smartphones.

- **Digital E-Book Support**
  The system can be expanded to offer **e-books**, allowing customers to buy and read digital formats directly from the platform.

- **Multi-Language Support**
  Adding support for multiple languages (English, Hindi, Tamil, etc.) will make the platform accessible to a wider audience.

- **Customer Chat Support / Chatbot**
  A built-in AI chat assistant or helpdesk system can be added to provide instant customer support and resolve queries quickly.

- **Auto Email & SMS Alert System**
  Notifications for order confirmation, shipping updates, out-of-stock alerts, and promotional messages can be automated.

- **Cloud Deployment**
  Migrating the system to cloud platforms (AWS, Azure, Google Cloud) will ensure better scalability, performance, and reliability.

- **Author Panel (New Enhancement)**

A dedicated author panel can be introduced where authors can:

  - Create and manage their own book listings
  - Update book details and pricing
  - View book sales statistics
  - Respond to customer questions
  - Upload new book editions or covers
  - This will make the system more interactive and useful for authors and publishers.

9. Glossary

The glossary explains the important words and terms used in this Online Bookstore System project. It helps readers understand both technical and non-technical terms that appear throughout the document. In this section, simple meanings are provided for common concepts like Admin, User Interface (UI), Database, Entity, REST API, JSF, EJB, Cart, Order, Category, and more. These explanations make it easier for anyone—whether they are students, developers, or general readers—to clearly understand how the system works. The glossary ensures that all readers can follow the project details without confusion and understand the functions, features, and technologies used in the online bookstore.