

A COMPREHENSIVE SOFTWARE MODELLING AND DESIGN PROJECT

BOOKSTORE MANAGEMENT SYSTEM



Selma Aksoy - Saadet Yilmaz - Rei Xhemo - Klevis Biraci-
Stefano Beli - Segeraldo Doci - Klavio Katiraj

23 May , 2024

OVERVIEW

OUR GOAL

to create a system that efficiently manages the operations of a bookstore, including tasks such as book inventory management, order processing, customer account management, and sales tracking.



PURPOSE & SCOPE

- aims to streamline the operations of a bookstore by managing books, customers, orders, and inventory efficiently.
- covers various user roles including customers, manager, and administrators, each with specific functionalities to enhance the bookstore experience.





KEY USER REQUIREMENTS

- The software shall keep track of the books supplied to the BookStore and their cost
- There must be different access levels for different user roles.
- The customer shall be able to browse books by their categories , genres , authors etc.
- The customer shall be able to browse books by their categories , genres , authors etc.

These requirements guided our functional and non-functional specifications.

MAJOR FUNCTIONAL REQUIREMENTS

Each functionality was designed to meet the specific needs of users and ensure a smooth operational flow.

➤ Administrators should have the ability to define permissions, restrictions, and access rights for each user role in the system.

➤ Customers can select desired books from search results or browse categories

➤ The system shall maintain a centralized inventory database that updates in real-time to reflect changes in stock levels.

➤ The system shall provide textboxes where the customer can submit their feedback.

➤ The system shall incorporate a recommendation engine to suggest books to customers based on their browsing history, preferences, and past purchases.

➤ The system shall support the addition of new categories and classifications by administrators, with the ability to assign books to multiple categories.

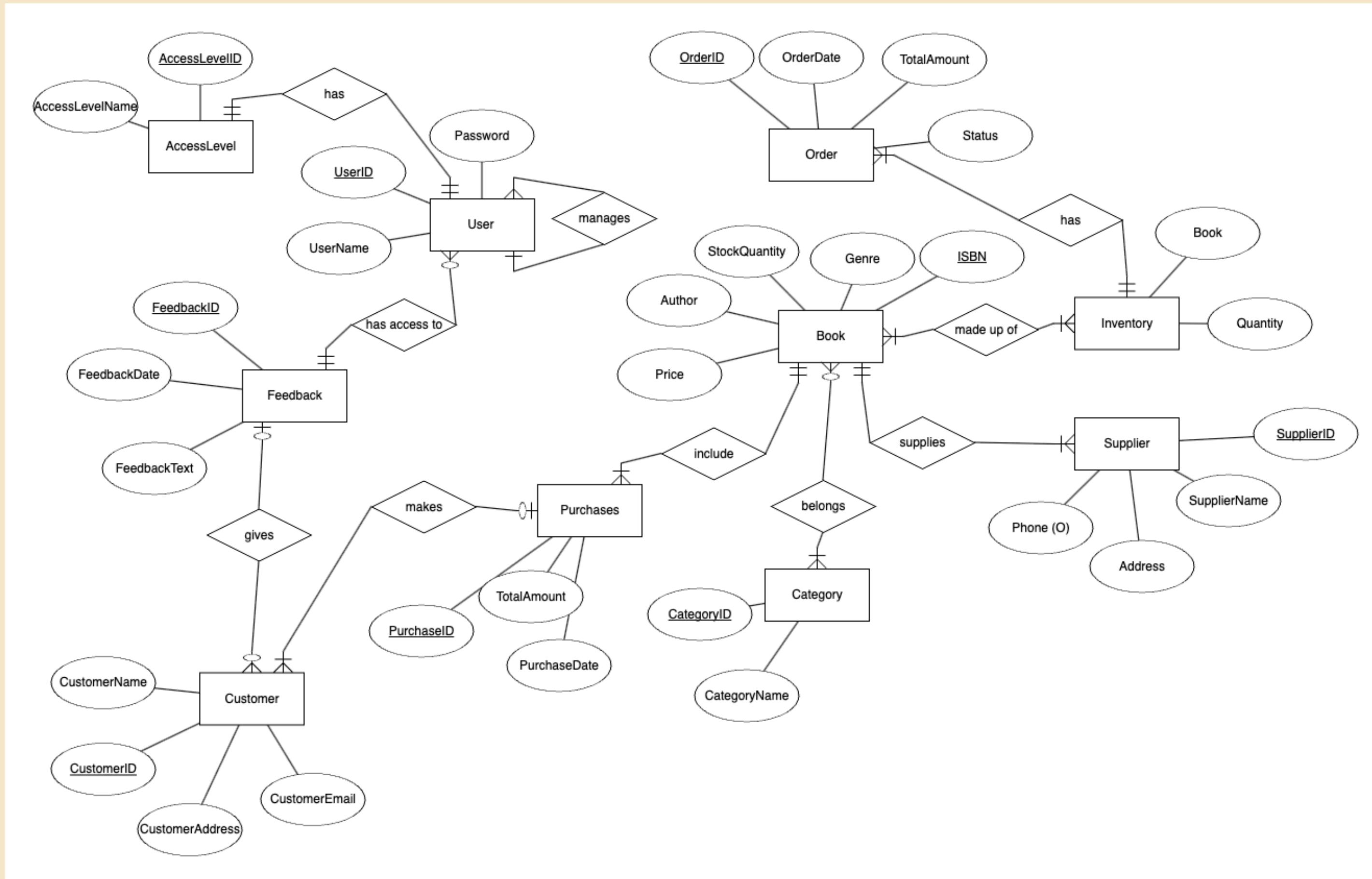
NON-FUNCTIONAL REQUIREMENTS

Our non-functional requirements focus on performance, security, usability, and scalability.

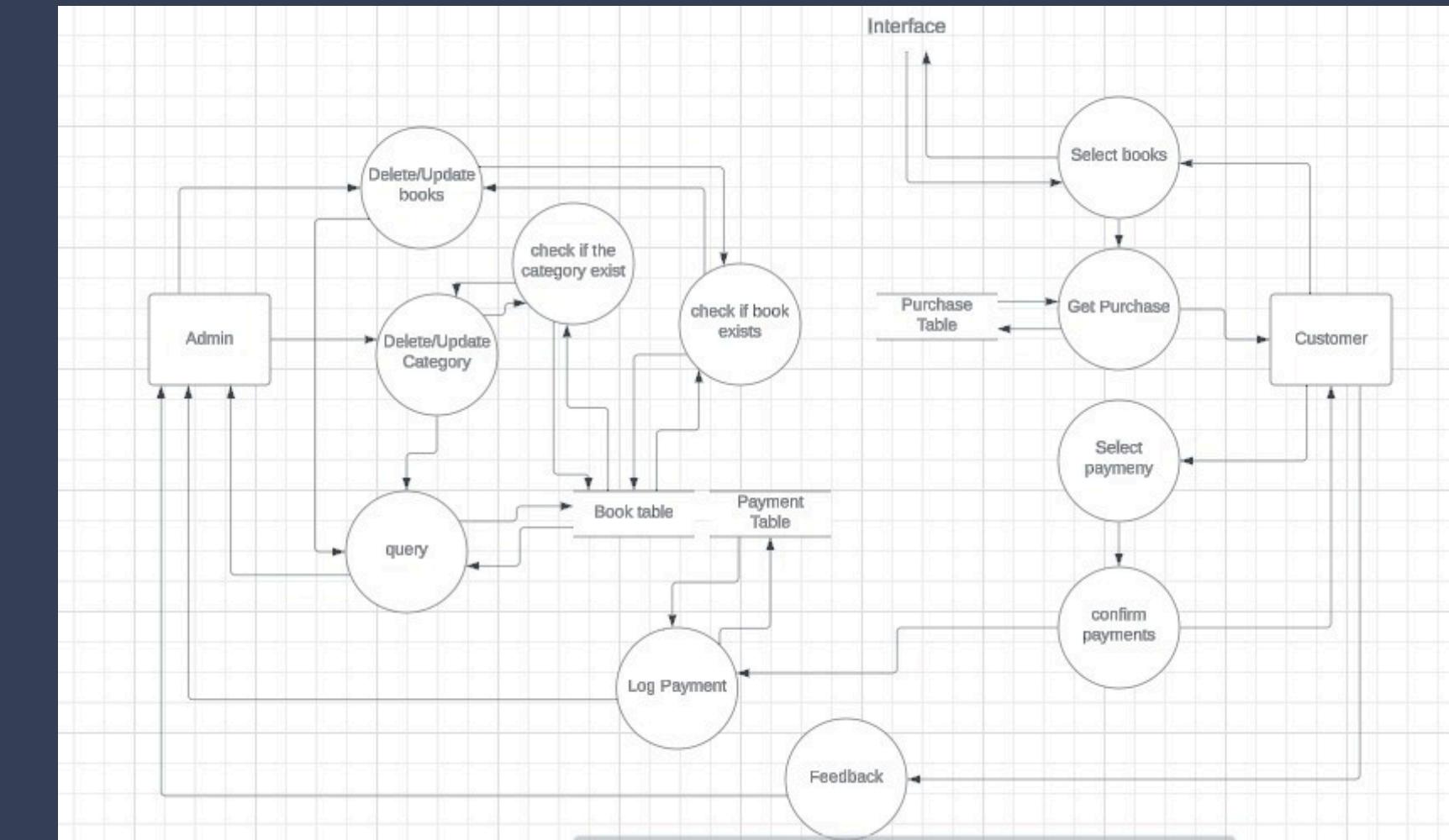
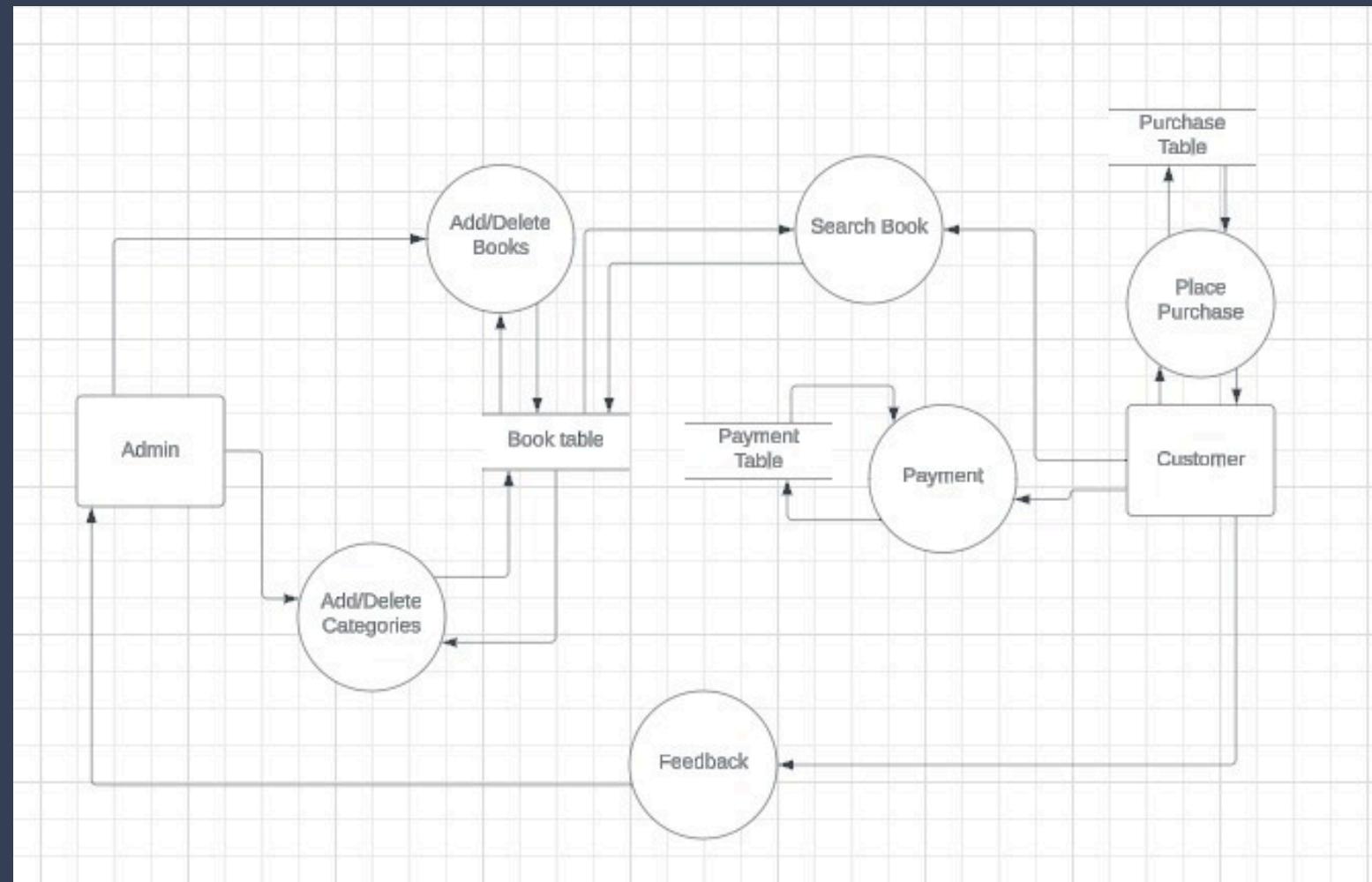
- The user interface should be intuitive and user-friendly, requiring minimal training for bookstore staff and customers to navigate and use the system.
- Access to user data and administrative features shall be restricted based on role-based access controls to prevent unauthorized access.
- Customers receive confirmation of their purchase after completing the checkout process. The system generates an order summary and provides a confirmation email to the customer.
- Data integrity shall be ensured through regular backups and redundancy measures to prevent data loss in case of system failures.

ERD DIAGRAM

The ERD we created outlines the major entities and their relationships

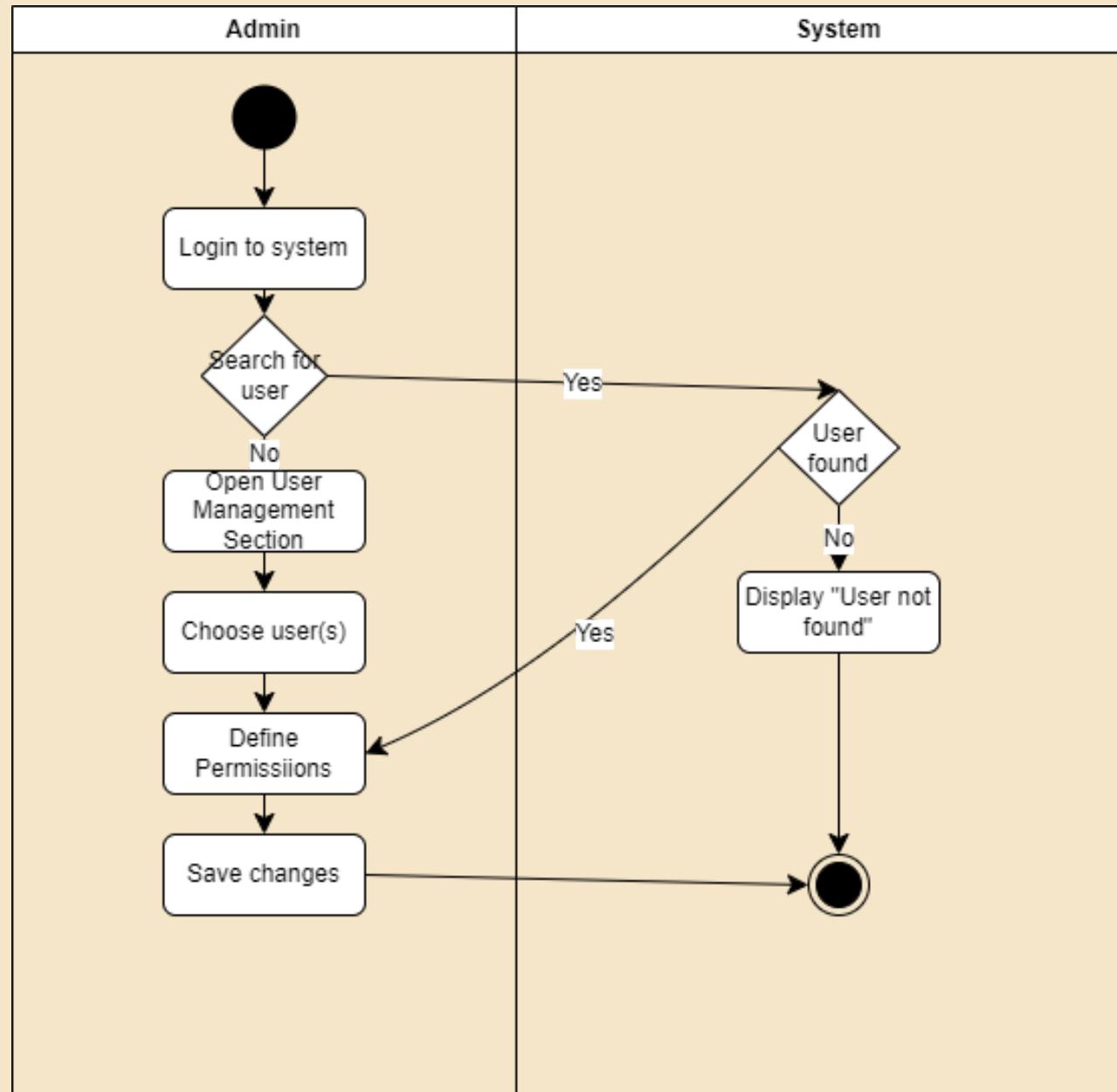


DFD LEVEL 0 - 1 - 2

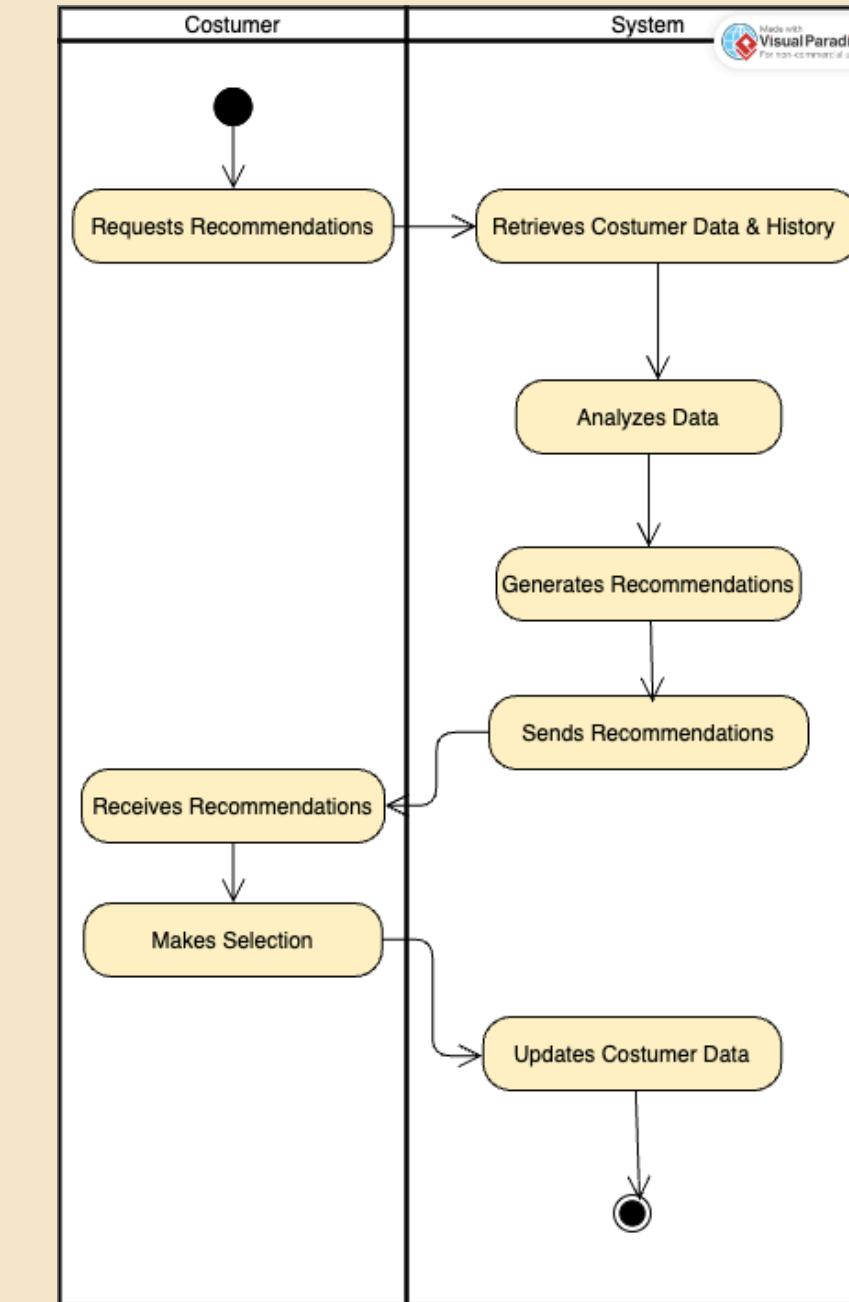


ACTIVITY DIAGRAMS

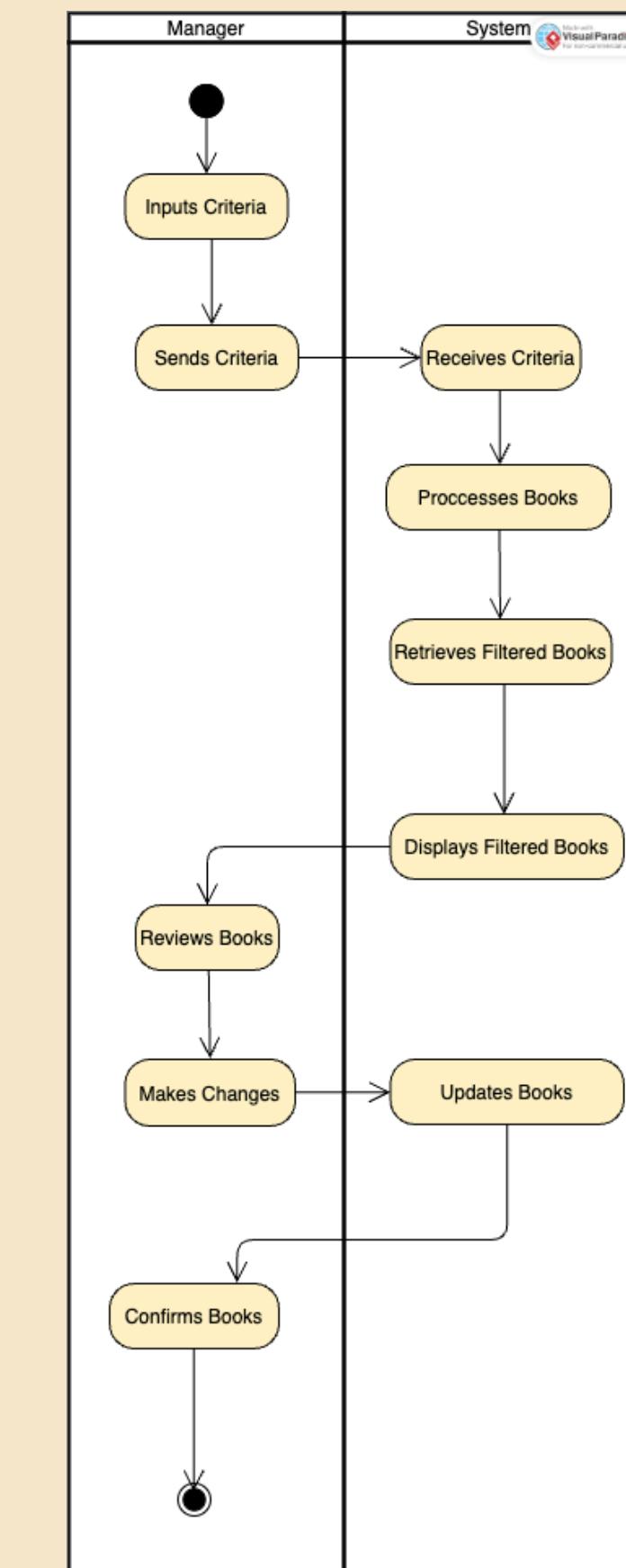
Activity diagrams were created for key system requirements to depict the flow of activities.



defining permission



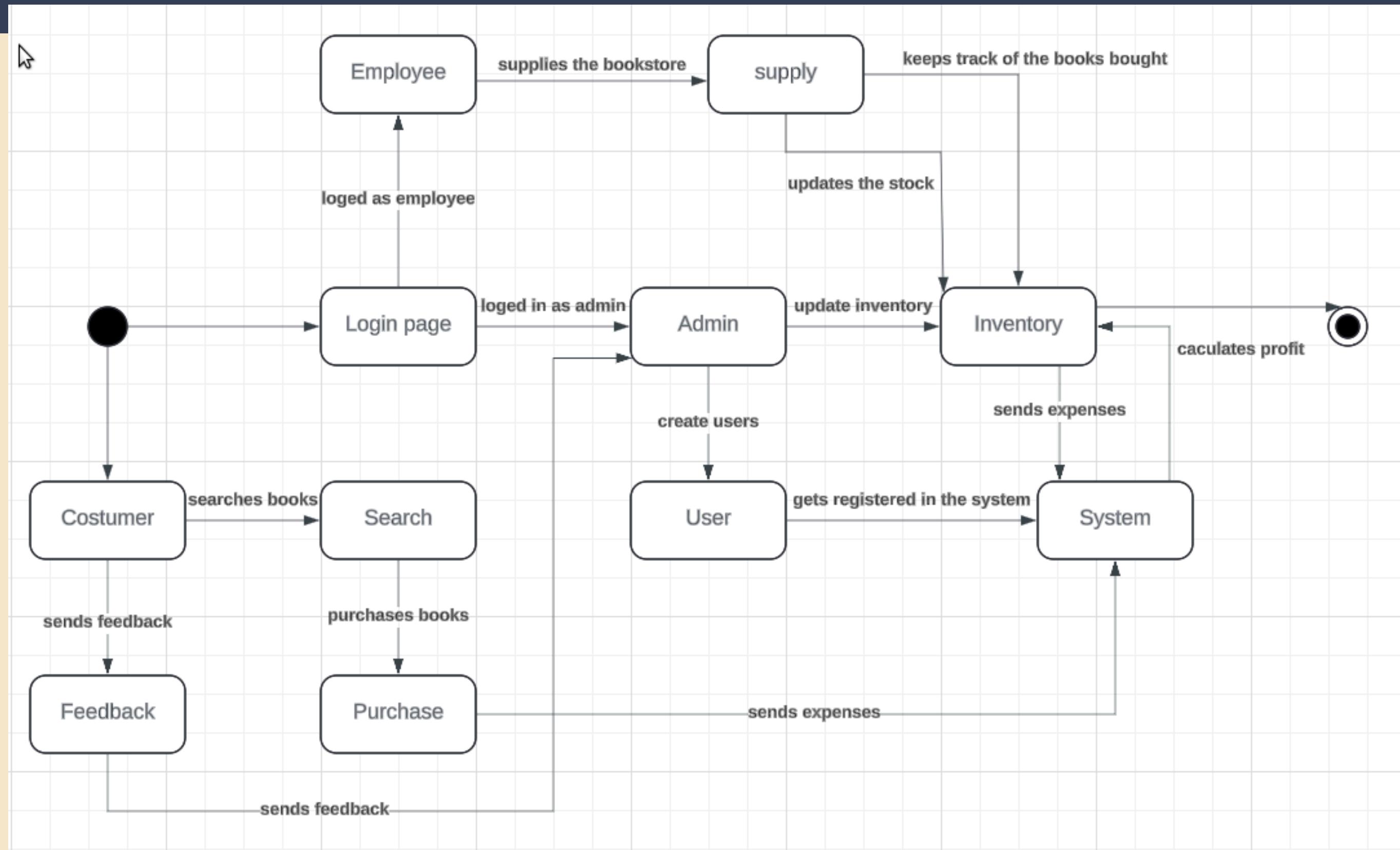
recommendation engine



filtering books

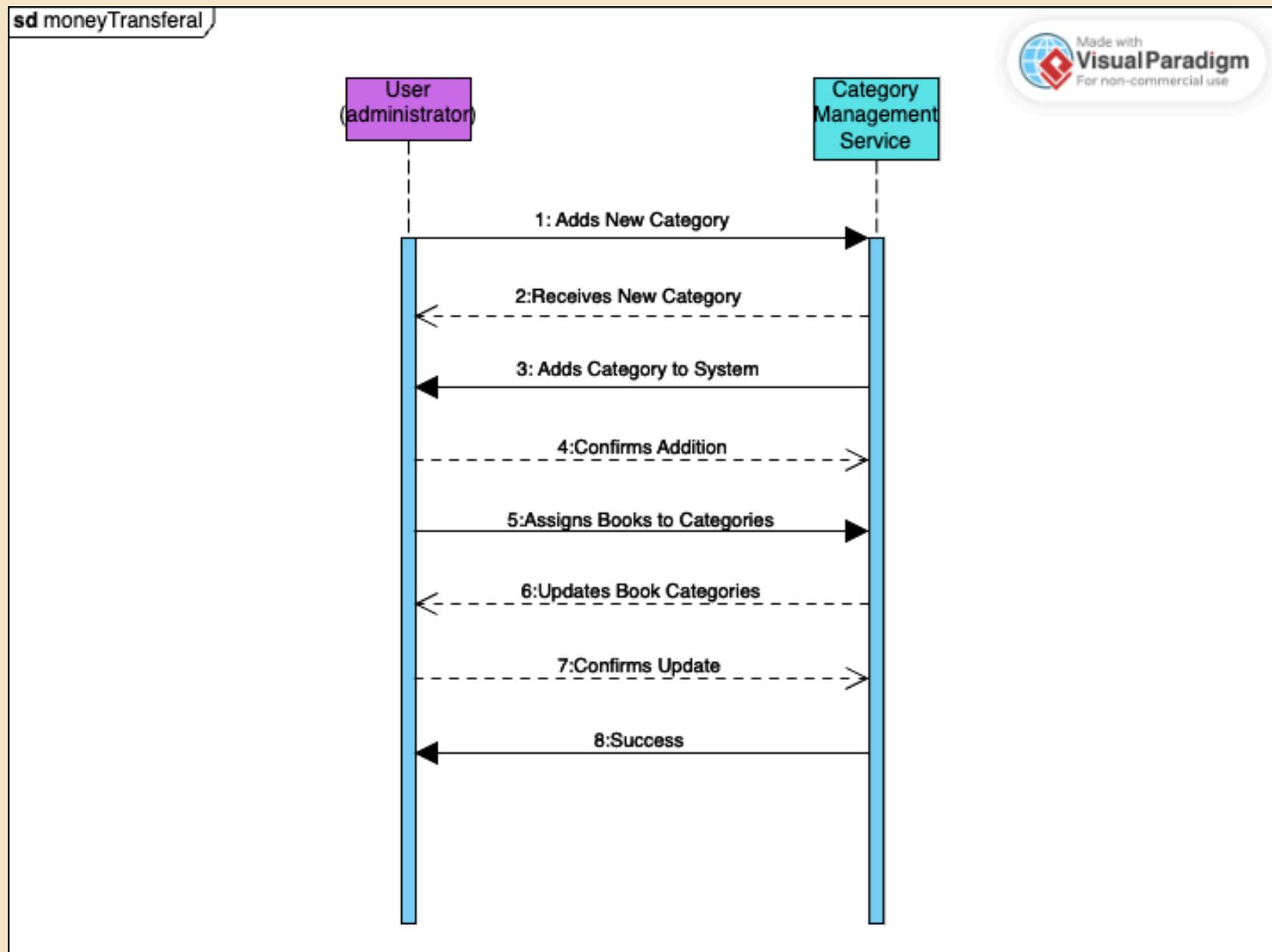
STATE-MACHINE DIAGRAM

This helped in visualizing state transitions based on user actions.



SEQUENCE

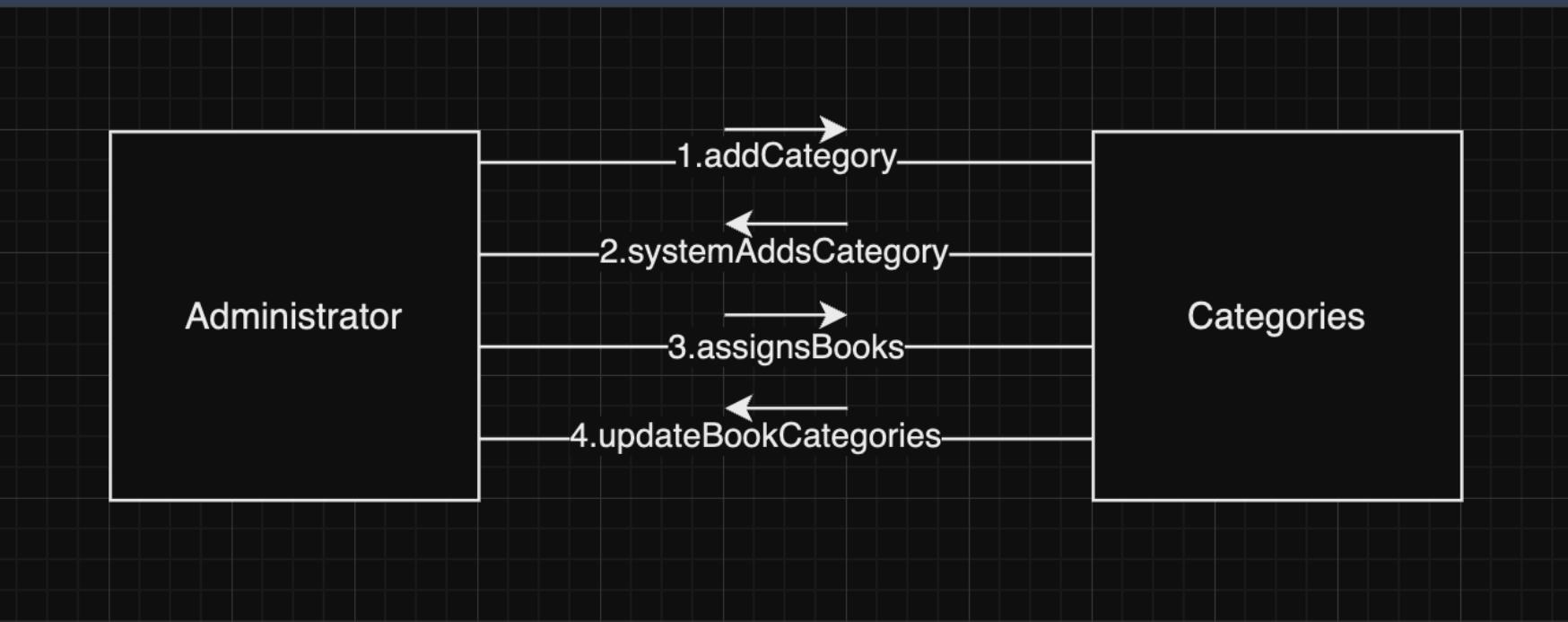
helped in understanding the interaction flow and timing.



adding new category

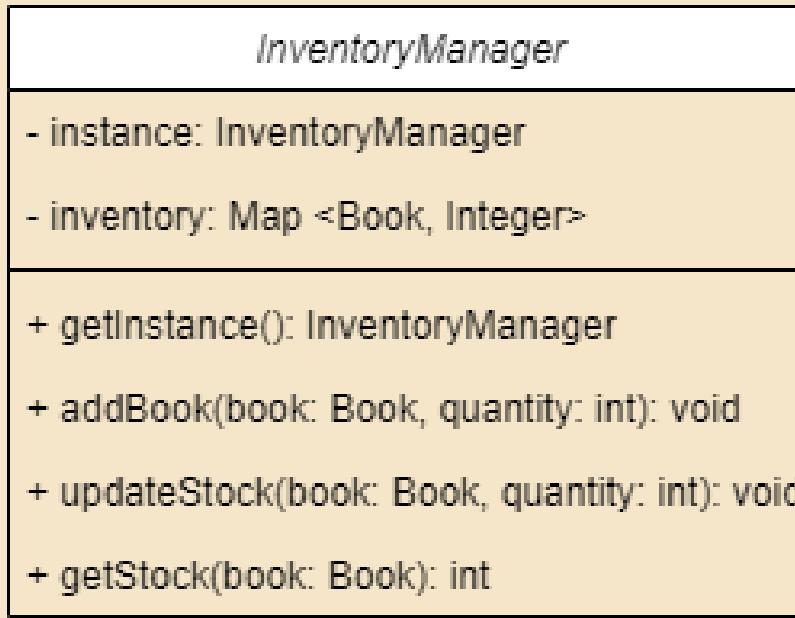
COLLABORATION

It complemented sequence diagrams by focusing on object roles and interactions.



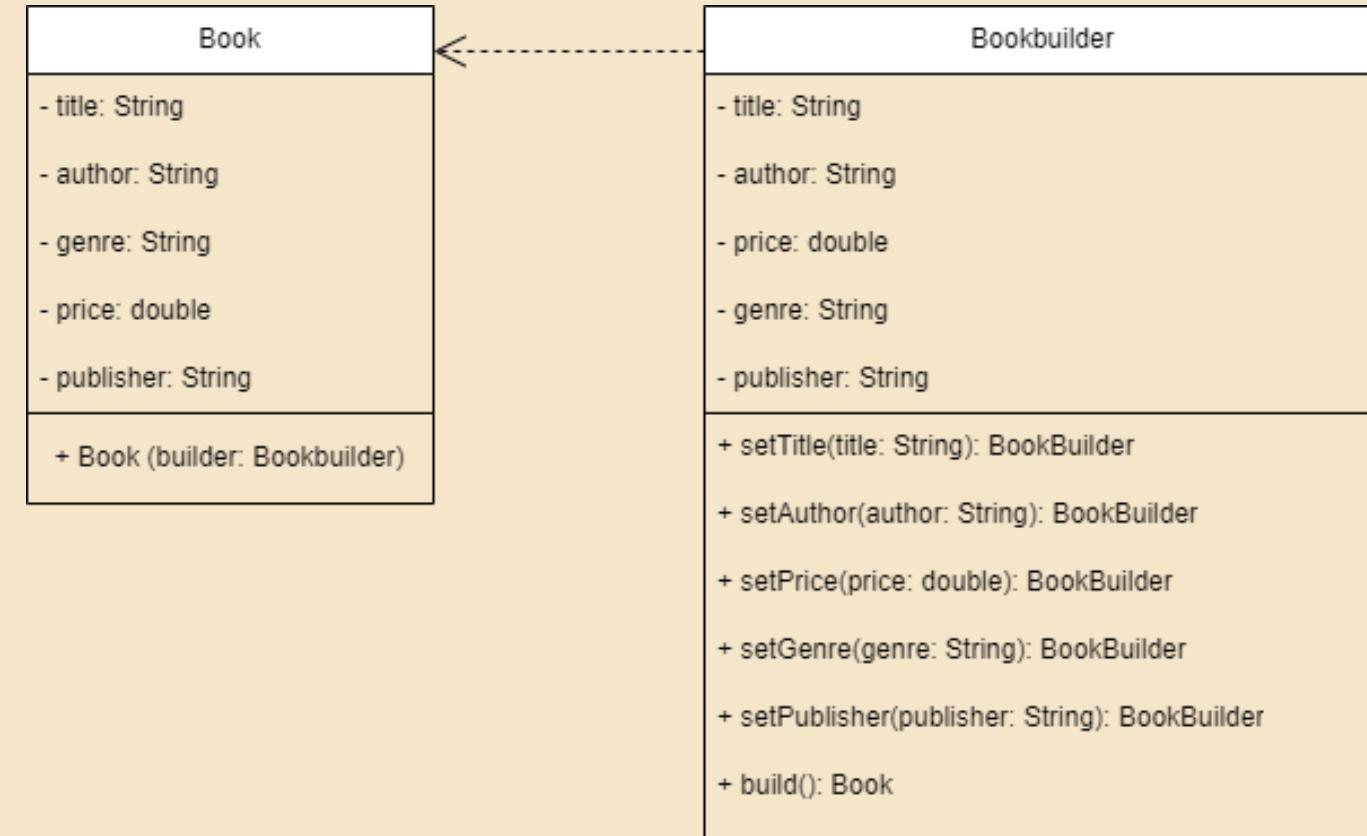
DESIGN PATTERNS

➤ SINGLETON



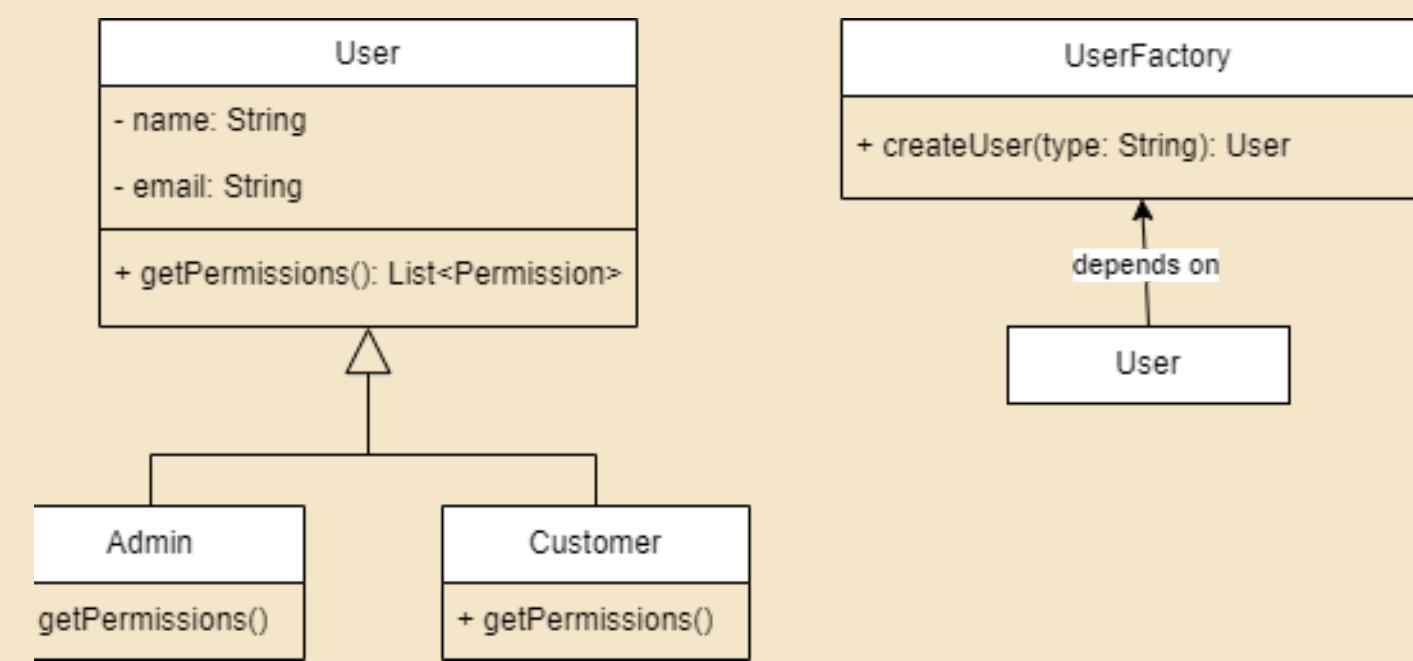
This is useful for managing shared resources like a configuration object or connection pool in our Book Store Management System, ensuring consistency and avoiding the overhead of multiple instances.

➤ BUILDER



In our system, the Builder pattern is particularly useful for creating complex objects like customer profiles or order details with various optional attributes

➤ FACTORY



The Factory Method pattern defines an interface for creating an object but allows subclasses to alter the type of objects that will be created

SUMMARY

In summary, our project involved defining user and system requirements, creating detailed diagrams, and applying design patterns to build a robust Book Store Management System. Each phase was crucial in ensuring the system's functionality and efficiency.

[githublink](#)



THANK YOU !

githublink