

Bookstore Management System Requirements Specification

May 23, 2024

This is the documentation for the Bookstore Management System project. This documentation aims to comprehensively detail the requirements and specifications necessary for the successful implementation and functionality of a user-friendly interface designed to enhance the experience of both customers and staff in a modern setting. The primary objective of this system is to create a seamless and engaging environment where customers can easily purchase books and interact with services, while staff members can efficiently manage other important operations.

This documentation is intended to provide a clear and detailed description of all requirement specifications for the system. It serves as a guide for developers, stakeholders, and other relevant parties involved in the project's development and implementation. By outlining the functional and non-functional requirements, this document aims to ensure a shared understanding of the project's goals, scope, and deliverables.

Bookstore System Documentation

Table of Contents

1. EXECUTIVE SUMMARY

- 1.1 PROJECT OVERVIEW
- 1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION

2. PRODUCT/SERVICE DESCRIPTION

- 2.1 PRODUCT CONTEXT
- 2.2 USER CHARACTERISTICS
- 2.3 ASSUMPTIONS
- 2.4 CONSTRAINTS
- 2.5 DEPENDENCIES

3. REQUIREMENTS

- 3.1 FUNCTIONAL REQUIREMENTS
- 3.2 NON-FUNCTIONAL REQUIREMENTS
 - 3.2.1 *Product Requirements*
 - 3.2.1.1 User Interface Requirements
 - 3.2.1.2 Usability
 - 3.2.1.3 Efficiency
 - 3.2.1.3.1 Performance Requirements
 - 3.2.1.3.2 Space Requirements
 - 3.2.1.4 Dependability
 - 3.2.1.5 Security
 - 3.2.2 *Organizational Requirements*
 - 3.2.2.1 Environmental Requirements
 - 3.2.2.2 Operational Requirements
 - 3.2.2.3 Development Requirements
 - 3.2.3 *External Requirements*
 - 3.2.3.1 Regulatory Requirements
 - 3.2.3.2 Ethical Requirements
 - 3.2.3.3 Legislative Requirements
 - 3.2.3.3.1 Accounting Requirements
 - 3.2.3.3.2 Security Requirements
- 3.3 DOMAIN REQUIREMENTS

4. USER SCENARIOS/USE CASES

- 4.1 DIAGRAMS
 - 4.1.1 USE CASE DIAGRAM
 - 4.1.2 ENTITY RELATIONSHIP DIAGRAM (ERD)
 - 4.1.3 BUSINESS PROCESS MODEL AND NOTATION (BPMN)
 - 4.1.4 DATA FLOW DIAGRAM (DFD)
 - 4.1.5 UNIFIED USE CASE AND CLASS DIAGRAM
 - 4.1.6 STATE DIAGRAM
 - 4.1.7 STATE DIAGRAM
 - 4.1.8 INTERACTION OVERVIEW DIAGRAM

5. USER INTERFACE

6. TESTING

Executive Summary

Project Overview

This project aims to develop a user-friendly bookstore management system that incorporates functionalities for both bookselling and internal bookstore operations. The target audience for this project includes the customers, the staff of the bookstore (managers and librarians) and administrators.

Customers: The primary audience includes users who can browse and purchase books, track their orders, access detailed book information and many other included functionalities which will be unveiled later in this document, that will enhance the customer's overall experience.

Staff: This system equips managers with the tools to monitor librarian performance and have access to statistics regarding books sold and bought and provides librarians with a simple and modern interface for managing book sales and committing bills.

Administrator: The system provides an administrator with complete control over the platform. They can add, modify, or delete staff member data (librarians and managers) and oversee all aspects of librarian and manager functionalities.

Purpose and Scope of this Specification

This specification outlines the functionalities, user roles, and technical considerations for the development of a multi-functional bookstore management system with e-commerce capabilities. This document will guide the development process and provide system users with a clear understanding of the platform's potentials.

Intended audience: Project Developers (providing them with a detailed understanding of the system requirements and functionalities to guide the development process), Project Managers, System Testers (to ensure the system meets its intended goals) and System Users (Librarians, Managers, Customers).

In Scope:

- E-commerce platform functionalities for book browsing, purchasing, and order tracking.
- Detailed book information and search capabilities.
- Chatbot integration for customer support.
- Review system for customer feedback.
- Discount code functionality for customer benefit.
- Staff login with differentiated access levels:
- Librarian access for managing book sales and manager access for monitoring librarian performance and sales data.
- Administrator access for complete control over staff data and system functionalities.

Out of Scope:

Physical Hardware and Network Infrastructure: The focus here is on the software functionalities of the bookstore management system.

Third-Party Integrations Beyond Defined Scope: Integrations with external services beyond those explicitly mentioned are not covered in detail.

Unmentioned Functionalities: This specification outlines a core set of functionalities for the bookstore management system. However, future revisions may incorporate additional features based on user feedback, evolving needs, or project scope adjustments. These functionalities will be documented as they are added to the system.

Product/Service Description

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

Product Context

This multi-functional bookstore management system is not entirely independent and self-contained. It interacts with various related systems and external interfaces to function comprehensively.

Internal System Components:

User Interface is the primary interface for users (customers, librarians, managers, administrators) to interact with the system. It provides functionalities for browsing books, managing purchases, managing staff data, etc. In addition, the system stores user information, book data, purchase history, and other relevant data in a central **database**. The **E-commerce Engine** handles online transactions, including secure payment processing and order fulfillment processes, while the **Security System** incorporates security measures to protect user information, financial transactions, and system data integrity.

External Interface: The system relies on internet connectivity for users to access the platform, process online transactions (if applicable), and potentially communicate with external systems.

User Characteristics

Students:

- **Type:** Student
- **Experience:** Limited experience with online bookstore systems.
- **Technical Expertise:** Familiar with using online platforms and mobile devices for various purposes.
- **Other Characteristics:** Price sensitive. May prioritize used books or discount options. Actively seek out new releases and popular titles. May be interested in features like online reviews and recommendations.

General Public:

- **Type:** Community member, Book enthusiast
- **Experience:** Varying levels of experience with online libraries. They may be new or experienced users.
- **Technical Expertise:** Moderate comfort level with technology expected. May require a user-friendly interface with clear instructions.
- **Other Characteristics:** May prioritize browsing by genre, author, or topic. Interested in features like customer reviews and may be attracted to special features the system provides.

Librarians:

- **Type:** Librarian (part of Staff)
- **Experience:** Experienced with bookstore operations and potentially management systems.
- **Technical Expertise:** High technical expertise expected. Comfortable navigating system functionalities and managing user accounts.
- **Other Characteristics:** Need efficient tools for managing book sales and processing transactions through bill printing.

Managers:

- **Type:** Manager (part of Staff)
- **Experience:** Experienced with bookstore operations and potentially management systems.
- **Technical Expertise:** High technical expertise expected.
- **Other Characteristics:** Need functionalities for adding new books in the system, monitoring librarian performance, analyzing sales data and statistics regarding books sold and bought. May require features for generating reports and making data-driven decisions.

Administrators:

- **Type:** Bookstore administrator with full system access
- **Experience:** High level of experience with IT systems and management systems.
- **Technical Expertise:** Expert-level technical expertise expected. Comfortable managing user accounts and system security.
- **Other Characteristics:** Need complete control over user access, system configuration, and data security.

Assumptions

We assume the system will be accessible through a web browser on various devices (desktops, laptops, tablets, etc.) with an internet connection. However, specific functionalities or layouts may need adjustments to optimize the user experience across different screen sizes and device capabilities.

We also take in consideration users will have a moderate comfort level with technology and using online platforms. The system should be designed with a user-friendly interface and clear instructions to be used by customers with varying technical skills.

Lastly, we assume integration with a secure payment gateway, considering the system's e-commerce functionality, which allows users to browse and purchase books.

These assumptions directly influence the technical specifications and functionalities documented in the system requirements. Any changes in the mentioned assumptions will necessitate revisions to the system requirements to ensure a successful implementation. *Example:* if the assumption about user device compatibility changes from primarily desktops to a wider range of mobile devices, the UI design might need adjustments for optimal usability on smaller screens.

Constraints

Access, Management, and Security: The system will require a robust access control system to manage user roles and permissions. Secure login protocols and data encryption will be crucial to protect sensitive user data and financial transactions.

System Resource Constraints: While specific limitations are unknown at this stage, the design should consider potential constraints on disk space, processing power, and memory usage. This may influence decisions regarding data storage optimization and code efficiency.

Other Design Constraints: Depending on the bookstore's or development team's established practices, the design may be constrained by specific choices for programming languages, frameworks, or coding standards.

Bookstore System Documentation

Dependencies

The system requires a robust Database Management System (such as MySQL or MongoDB) to store and manage all data related to books, users, sales, reservations, and reviews. The database schema and setup must be completed before developing any modules that require data storage and retrieval (e.g., user management, book inventory).

A web server (such as Apache) is required to host the Bookstore Management System and handle user requests.

The development of the system will utilize frameworks and languages such as:

Frontend example: HTML, CSS, JavaScript

Backend example: PHP.

An authentication service (such as OAuth) is needed to manage user login and access control for different user roles (customers, staff, admin). The user authentication module must be developed and tested before implementing role-based access control for different user functionalities.

The system needs to integrate with a payment gateway (such as PayPal) to process book purchases securely. Payment gateway integration needs to be completed and tested before the book purchase functionality can be made available to customers.

A service for sending email or SMS notifications to customers regarding book availability and other updates and an integration with a chatbot framework or service to provide customer support and handle queries are also part of the software dependencies of the system. The chatbot must be integrated and configured before offering customer support through the system.

Hardware Dependencies: The system needs to have reliable server infrastructure to host the web server, database, and other backend services. In addition, the system must be compatible with various client devices, including desktops, laptops, tablets, and smartphones.

Requirements

Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd
BR_01	The system will be composed of the following key components: Home Page, Review Page, Search Bar, Add to Cart and Us.	These functionalities provide a starting point for user interaction and lay the groundwork for further development of more advanced features based on user needs and future requirements.	1	18/03/2024
BR_02	Librarians will be able to print bills for each book sold to the customer, allowing for accurate record-keeping of transactions.	The ability for librarians to print bills for each book sold was established as a core requirement during the initial phase of project planning.	1	18/03/2024
BR_03	Librarians will be able to edit the bill if he/she has made a mistake while committing it.	None	2	18/03/2024

Bookstore System Documentation

Req#	Requirement	Comments	Priority	Date Rwd
BR_04	Managers shall have the capability to add new books to the system, including relevant details such as title, author, genre and quantity.	By enabling managers to directly add new books, the system streamlines the process of managing the inventory and facilitates the addition of new resources to meet user needs.	1	27/03/2024
BR_05	Managers shall have the capability to delete or modify data of books in the system.	The system creates a more efficient and organized way to manage the available books	1	27/03/2024
BR_06	Managers must be able to monitor the performance of librarians, including tracking the number of bills printed, data for books sold and total revenue on a daily basis.	The inclusion of librarian performance monitoring functionalities was established as a requirement during the initial project planning phase.	2	27/03/2024
BR_07	Managers will have access to statistics regarding books sold and bought, including information about name of the book, quantity and price.	By analyzing sales data, managers can identify popular titles, genres in high demand, and potential areas for expanding the collection to better cater to user needs.	3	02/04/2024
BR_08	Admins will have the authority to manage employee accounts, including the ability to add, delete or modify employee data as needed.	None	1	02/04/2024
BR_09	Allow users of the system to view customer reviews for books listed on the system.	Users gain valuable insights from other readers' perspectives, including writing style, content quality, and suitability for their interests.	2	02/04/2024
BR_10	Allow customers to submit support queries or reach a human representative.	Providing multiple support options demonstrates the bookstore's commitment to user satisfaction and its willingness to address user concerns effectively.	3	02/04/2024
BR_11	Allow customers to apply a discount code to their order in the checkout page.	None	3	02/04/2024
BR_12	Allow users to reset their password during authentication	Users can recover their accounts independently without requiring staff intervention. This provides a convenient and efficient solution for forgotten passwords.	2	02/04/2024
BR_13	Users receive notifications when a book they want to purchase is available in the stock.	None	3	02/04/2024

Bookstore System Documentation

Req#	Requirement	Comments	Priority	Date Rwd
BR_14	Users can check if a book is available to buy or not.	Real-time inventory information builds trust with users by ensuring they have accurate and up-to-date information about book availability.	2	02/04/2024
BR_15	Users can navigate through the inventory of the bookstore	None	2	04/04/2024

Non-Functional Requirements

3.2.1 Product Requirements

These requirements specify that the delivered product must behave in a particular way, such as execution speed, reliability, etc.

3.2.1.1 User Interface Requirements

The user interface must be intuitive and easy to navigate.

The interface must include a responsive design to adapt to various screen sizes (desktops, laptops, tablets, and smartphones).

Menus should be logically structured with clear categories such as Home, Search, Cart, User Profile, and Admin Panel.

The system should provide clear, actionable error messages for user errors and system failures.

The system should support multiple languages, allowing users to select their preferred language for the interface.

3.2.1.2 Usability

Learnability:

The system should be easy to learn, with minimal training required for new users.

It should provide an onboarding tutorial for first-time users to guide them through key features.

User Documentation and Help:

The user documentation and help should be complete and context-sensitive, explaining how to achieve common tasks.

An online help center should be available with FAQs, user guides, and video tutorials.

Ease of Use:

The system should be designed with a user-friendly interface, including tooltips and guided tutorials to accommodate users with varying technical skills.

Users should be able to complete basic tasks, such as searching for books or placing orders, within three clicks from the home page.

3.2.1.3 Efficiency

3.2.1.3.1 Performance Requirements

- The system should handle up to 500 concurrent users without performance degradation.
- 95% of transactions shall be processed in less than 1 second.
- Page load times should not exceed 2 seconds under normal usage conditions.
- The system should support up to 10,000 active book listings without impacting search performance.

3.2.1.3.2 Space Requirements

- The system should have a maximum storage capacity of 1TB for book data and user information, scalable based on future needs.
- Each user account should have a limit of 100MB for storing personal data, including purchase history and preferences.
- The system should support efficient data archiving to minimize storage usage for older transaction records.

3.2.1.4 Dependability

Availability

The system must be available 99.9% of the time, with downtime limited to scheduled maintenance periods.

Scheduled maintenance should occur during off-peak hours and should be communicated to users at least 48 hours in advance.

Unscheduled downtime should be resolved within 2 hours, and users should be notified immediately.

Reliability

The system should include health monitoring, error detection, and logging functionalities to notify administrators of potential issues.

Data backup should be performed daily, with the ability to restore data within 4 hours in the event of a failure.

Monitoring:

Health Monitoring - The system should include real-time monitoring of critical components such as server performance, database health, and network connectivity. Implement dashboards that display key performance indicators (KPIs) like CPU usage, memory usage, disk I/O, and response times.

Failure Conditions - Define and document failure conditions with performance thresholds. Set up alerts for system administrators when thresholds are exceeded.

Error Detection - Implement input validation, null value checks, and exception handling. Use automated testing (unit, integration, and system tests).

Logging - Log significant events: user actions, errors, security breaches, and performance metrics. Timestamp and include details like user IDs and error codes. Use centralized logging for easy search and analysis.

Error Correction - Provide automated error correction mechanisms (e.g., retry failed transactions). Ensure clear and actionable error messages. Maintain an error reporting and tracking system.

Maintenance

- **Modularity**

The system should be designed with a modular architecture, separating core functionalities into distinct, manageable modules (e.g., user management, book inventory, transaction processing, reporting).

Each module should have well-defined interfaces, allowing for independent updates and maintenance without affecting other parts of the system.

- **Code Complexity**

The system's codebase should be kept as simple as possible, avoiding unnecessary complexity. This includes using clear, consistent coding standards and practices.

Complex algorithms or processes should be documented thoroughly, including comments within the code to explain intricate logic and flow.

- **Interface Design**

The administrative interface should include tools for monitoring system health and performance, such as dashboards displaying key metrics (e.g., server load, transaction rates, error rates).

Provide an interface for system administrators to manage configurations and settings without requiring direct code changes or database manipulation.

3.2.1.5 Security

The system must comply with industry-standard security protocols to protect user data. This includes using encryption for data storage and transmission, implementing secure login mechanisms, and conducting regular security audits.

Implement multi-factor authentication (MFA) for administrator and manager logins.

The system should include data integrity checks to ensure accurate data storage and retrieval. Regular integrity checks should be performed to prevent and detect data corruption.

User passwords should be stored using secure hashing algorithms, such as bcrypt or Argon2.

Activity logs should be maintained for all administrative actions, with retention for at least 1 year.

3.2.2 Organizational Requirements

Requirements which are a consequence of organizational policies and procedures e.g. process standards used, implementation requirements, etc

Environmental Requirements

The system should operate on standard web browsers (Chrome, Firefox, Safari, Edge). Ensure compatibility with the latest versions of these browsers and perform regular testing for updates.

The system should be hosted on servers with adequate capacity and redundancy to handle peak loads. Ensure that the hosting environment complies with relevant environmental regulations and standards, such as ISO 14001.

Operational Requirements

The system should be maintained and updated regularly to ensure smooth operation. Schedule regular maintenance windows and update cycles to address bugs and implement new features. A dedicated support team should be available 24/7 to handle technical issues and user queries. The system should include automated monitoring tools to detect and alert on performance issues.

Development Requirements

The development process should follow Agile methodologies, using iterative development, regular sprint reviews, and incorporating user feedback throughout the development cycle. All code should be peer-reviewed and adhere to established coding standards to ensure maintainability and quality. Use continuous integration/continuous deployment (CI/CD) pipelines to automate testing and deployment processes.

3.2.3 External Requirements

Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Regulatory Requirements

The system must comply with GDPR for handling user data if operating within the EU. Ensure that data handling processes are in place for data access, correction, and deletion requests. Comply with local data protection laws and regulations in all regions where the system is available. Regularly review and update privacy policies to reflect changes in regulations and ensure transparency with users.

Ethical Requirements

The system should ensure user privacy and ethical handling of data. Implement policies for ethical data use, including anonymizing user data where possible. Obtain user consent before collecting or using personal data, and provide clear options for users to manage their data preferences. Ensure that all advertising and promotional activities within the system are ethical and transparent.

Legislative Requirements:

Accounting Requirements

- The system must comply with relevant accounting standards for transaction records. Ensure accurate and auditable transaction records for financial reporting.
- Implement audit trails for all financial transactions, including timestamps and user identification.
- Provide exportable financial reports in standard formats, such as CSV or PDF, for accounting purposes.

Security Requirements

- The system must adhere to PCI DSS standards for handling payment information. Regularly review and update security measures to maintain PCI compliance.
- Conduct regular security assessments and penetration testing to identify and mitigate vulnerabilities.
- Ensure that all third-party payment processors used by the system are compliant with relevant security standards.

Domain Requirements

Bookstore Management

The system should support the addition, deletion, and modification of book records, including details such as title, author, genre, ISBN, and quantity.

Enable categorization of books into various genres and subgenres for easier browsing and management.

User Roles and Permissions

Distinct user roles: Customer, Librarian, Manager, and Administrator, each with specific permissions and access levels are defined.

Each role has access only to the functionalities relevant to their responsibilities (e.g., librarians can manage book sales, managers can analyze sales data).

Book Inventory Management

Provide real-time tracking of book inventory, including alerts for low stock and automatic updates when books are sold or restocked.

Order and Transaction Management

Support the complete lifecycle of book transactions, from browsing and purchasing to payment processing and order fulfillment.

Allow users to view their order history and track the status of their current orders.

Reporting and Analytics

System generates detailed reports on book sales, inventory levels, user activity, and system performance. System provides analytics tools for managers to identify trends, popular genres, and user preferences.

User Scenarios/Use Cases

1. Add New Book

Ensure the inventory is always current and complete by allowing managers to quickly and efficiently add new books to the inventory. Managers can input details of new books, including title, author, genre, ISBN, and quantity, into the system.

Problem: New books need to be added promptly to keep the collection up-to-date and meet customer demands.

Actors: Manager

STEPS

Manager logs into the system.

Navigates to the "Add New Book" section.

Fills in the book details.

Submits the form to add the book to the inventory.

Outcome: The new book is added to the inventory and becomes available for customers to view and borrow or purchase.

2. Remove book

Bookstore System Documentation

Maintain a relevant and up-to-date collection by removing outdated or damaged books. Managers use the interface to remove books from the system when they are outdated, damaged, or no longer needed.

Problem: Books that are no longer relevant or in poor condition need to be removed from the inventory.

Actors: Manager

STEPS

Manager logs into the system.

Navigates to the "Remove Book" section.

Searches for the book to be removed.

Confirms the removal of the book.

Outcome: The book is removed from the inventory, and customers can no longer view or borrow it.

3. Edit Book Characteristics

Keep book information accurate and up-to-date. Managers can update details of books in the inventory to correct errors or reflect changes.

Problem: Book details such as author, genre, or quantity may need to be updated due to errors or changes.

Actors: Manager

STEPS

Manager logs into the system.

Navigates to the "Edit Book Characteristics" section.

Searches for the book to be edited.

Updates the necessary details.

Submits the changes.

Outcome: The book details are updated in the system, ensuring accurate information is available to customers.

4. Check Book Statistics

Analyze the popularity and circulation of books to inform inventory decisions and bookstore services. Managers can view statistics about book checkouts, revenue generated etc.

Problem: Librarians and managers need access to statistics about book checkouts and returns to make data-driven decisions.

Actors: Managers, Administrator

STEPS

User logs into the system.

Navigates to the "Check Book Statistics" section.

Selects the book or range of books to view statistics for.

Review the displayed statistics.

Outcome: Statistics page works as intended.

5. Check Warehouse Inventory

Ensure that the bookstore knows its stock levels to manage ordering and inventory effectively. Administrators and managers use the interface to check warehouse inventory. This way, they can check the current stock levels of books in the warehouse and manage reordering efficiently.

Problem: Accurate information about current stock levels is needed to avoid overstocking or running out of books.

Actors: Managers, Administrator

STEPS

User logs into the system.

Bookstore System Documentation

Navigates to the "Check Warehouse Inventory" section.

Searches for specific books or categories.

Review the stock levels displayed.

Outcome: Users can make informed decisions about ordering and stock management based on current inventory data.

6. Commit bills

Ensure accurate and timely processing of customer transactions. Librarians use the system functionalities to commit bills and finalize transactions for book purchases. This way, accurate billing records are ensured while the number of bills committed per day is recorded.

Problem: Bills need to be committed and recorded accurately to maintain financial integrity.

Actors: Librarians

STEPS

Librarian logs into the system.

Navigates to the "Commit Bills" section.

Review the bill details.

Confirms and commits the bill.

Outcome: The bill is finalized, and the transaction is recorded in the system.

7. About us

Customers can view information about the bookstore, including its history, mission, services, and contact details.

Problem: Customers need to understand the services of the bookstore and get information about the business.

Actors: Customers

STEPS

Customer navigates to the "About Us" page.

Customer reads the provided information.

Outcome: Customers gain a better understanding of the bookstore and its services.

8. Home page

Serve as the main landing page, a user-friendly and informative entry point that provides easy access to key features and information. The home page serves as the main navigation hub, displaying featured books, announcements, and quick access links to other sections.

Problem: Users need a central point to navigate the LMS and access its features.

Actors: Customers

STEPS

User accesses the home page.

Navigates to desired sections using the available links.

Outcome: Users can efficiently access various features and information.

9. Edit bills

Allow corrections and updates to billing records to ensure accuracy. The main objective is to ensure billing records are accurate and up-to-date.

Problem: Occasionally, bills may need to be edited to correct errors or update information.

Objectives: Ensure billing records are accurate and up-to-date.

Actors: Librarians

STEPS

User logs into the system.

Navigates to the "Edit Bills" section.

Bookstore System Documentation

Searches for the bill to be edited.

Make the necessary changes and save the updated bill.

Outcome: The bill is updated

10. Security and compliance

Ensure the system is secure and complies with relevant regulations and standards. Administrators can also manage security settings and enforce role-based access control.

Problem: The system must protect sensitive information and adhere to legal requirements.

Actors: Administrators

STEPS

Admin logs into the system.

Navigates to the "Security and Compliance" section.

Reviews and updates security settings.

Ensures compliance with relevant standards.

11. Check performance

Monitor and evaluate the performance of librarians and overall bookstore operations. Provide managers with tools to track performance reports and analytics related to librarian activities and bookstore operations.

Problem: Managers need access to performance metrics to ensure efficient operations and identify areas for improvement.

Actors: Managers

STEPS

Manager logs into the system.

Navigates to the "Check Performance" section.

Reviews performance metrics and reports.

Outcome: Managers can assess performance and make informed decisions to improve bookstore operations.

12. Reservation notification system

Inform customers via email or SMS when reserved books become available (the system sends automated notifications to customers).

Problem: Customers need timely notifications about the availability of reserved books to enhance their user experience.

Actors: Customers

STEPS

Customer reserves a book that is currently out of stock.

The system monitors the inventory.

When the book becomes available, the system sends a notification to the customer.

Outcome: Customers receive timely notifications and can promptly pick up or purchase their reserved books.

13. Search and browse by various criteria

Provide customers with the ability to find books quickly and easily using various search criteria such as title, author, genre, and keywords, and browse the catalog.

Problem: Customers need efficient search and browsing tools to find books of interest.

Actors: Customers

STEPS

Customer navigates to the search page.

Enters search criteria or selects browsing filters.

Bookstore System Documentation

Reviews search results or browse categories.

Outcome: Customers find books of interest quickly and efficiently.

14. Encryption of sensitive information

Protect sensitive information from unauthorized access. The system encrypts sensitive data stored and transmitted (such as customer data, transaction details, and employee records).

Problem: Sensitive data must be encrypted to ensure security and compliance with regulations.

Actors: System (non-functional requirement)

STEPS

Sensitive data is entered or transmitted.

The system automatically encrypts the data.

Encrypted data is stored or transmitted securely.

Outcome: Sensitive information is protected from unauthorized access.

15. Role based access control

Ensure that users only have access to information and functions relevant to their roles. The main objective is to manage user access effectively to enhance security and operational efficiency. Administrators assign roles to users, and the system enforces access controls based on these roles.

Problem: Access control must be managed to prevent unauthorized actions and data access.

Actors: Administrators, Librarians, Managers, Customers

STEPS

User logs into the system.

Roles and permissions are assigned to the users.

Users access the system with permissions based on their roles.

Outcome: Users have appropriate access levels, preventing unauthorized actions.

16. View book stock

Provide librarians/customers with up-to-date information on book availability and stock levels.

Problem: Customers and staff need to know the availability of books to make informed decisions.

Actors: Customers, Librarians

STEPS

User logs into the system.

Searches for a book or navigates to the book's details page.

Views the current stock level for the book.

Outcome: Customers or Librarians can see if a book is available and make decisions accordingly.

17. Password reset option

Allow users to reset their passwords securely and conveniently. The main objective is to provide a secure and user-friendly way to reset passwords.

Problem: Users may forget their passwords and need a way to regain access to their accounts.

Actors: Customers, Librarians, Managers, Administrators (all users)

STEPS

User navigates to the login page.

Click on the "Forgot Password" link.

Enters their registered email address (where they receive a password reset link)

Click the link and set a new password.

Outcome: Users can regain access to their accounts securely and promptly.

18. Modify data of an employee

Bookstore System Documentation

Keep employee information accurate and up-to-date. Admins use the admin interface to modify employee data (such as contact information, roles, and employment status).

Problem: Employee details may need to be updated due to changes in roles, contact information, or other factors.

Actors: Administrators

STEPS

Admin logs into the system.

Navigates to the "Modify Employee Data" section.

Searches for the employee to be updated.

Updates and saves the necessary details.

Outcome: Employee records are updated, ensuring accurate information is maintained.

19. Fine management (for overdue items)

Manage and process fines by the librarians for the customers of the bookstore

Problem: Fines need to be applied and collected for any infraction of rules of the bookstore.

Actors: Librarians

STEPS

Librarian logs into the system.

Navigates to the "Fine Management" section.

Reviews applicable fines and applies fines to the relevant customer accounts.

Outcome: Fines are managed effectively

20. Monthly and total revenue

Track the library's revenue to understand financial performance and make informed decisions. The primary objective is to provide detailed insights into monthly and total revenue.

Problem: Accurate revenue tracking is essential for financial planning and management.

Actors: Managers

STEPS

User logs into the system.

Navigates to the "Revenue Reports" section.

Selects the desired time period (monthly or total) to review the revenue report.

Outcome: Users gain insights into the library's financial performance.

21. Backup the database

Ensure data is protected against loss or corruption by maintaining regular backups. It is crucial to provide a reliable mechanism for backing up and restoring data.

Problem: Data loss or corruption can have significant impacts on library operations.

Actors: System (non-functional requirement)

STEPS

Administrator logs into the system.

Navigates to the "Backup Database" section.

Schedules regular backups or initiates a manual backup.

Outcome: Data is regularly backed up, ensuring it can be restored in case of loss or corruption.

22. View order history

Bookstore System Documentation

Allow customers to access their order history for reference and tracking purposes. Customers access their order history through their account.

Problem: Customers need visibility into their past transactions and orders.

Actors: Customers

STEPS

Customers log into their account.

Under the “Orders” section the user must navigate to “Past Orders”

Views the list of past orders and transactions.

Outcome: Customers can easily track their past orders for reference.

23. View customer reviews

Enable customers to read and evaluate book reviews from other users.

Problem: Customers rely on reviews to make informed decisions about book purchases.

Actors: Customers

STEPS

Customer navigates to the book details page.

Scrolls down to the reviews section.

The system displays the reviews.

The user can filter the reviews based on a scale of 1-5 and read through them.

Outcome: Customers gain insights into the quality and popularity of a book through peer reviews.

24. Leave a book review

Enable customers to share their opinions and experiences by leaving book reviews. The main goal is to encourage customer engagement and participation.

Problem: Customers want to contribute feedback and help other users make informed decisions.

Actors: Customers

Metrics: Number of reviews submitted, review engagement metrics.

Use Case

Description: Customers can leave reviews for books they have read and share their opinions with other users.

STEPS

Customer navigates to the book details page.

Click on the "Leave a Review" button.

Rates the book and writes a review.

The system saves the review in the database.

The review section on the product page is updated, displaying the review saved as the most relevant one.

Outcome: Customer reviews contribute to the overall feedback and evaluation of books in the system.

25. Contact support

Provide customers with a way to reach out for assistance with inquiries, issues, or feedback.

Problem: Customers may encounter issues or have questions that require assistance from support staff.

Actors: Customers

STEPS

The customer clicks on the chatbot pop up on the website.

The customer submits their issue as a query.

The chatbot generates a response based on the query or connects the user to a human customer service representative

Bookstore System Documentation

Outcome: Support staff address customer inquiries and issues promptly and effectively.

26. Apply discount code to checkout page

Allow customers to apply discount codes to their orders for promotional purposes.

Problem: Customers may have discount codes that they want to use for savings on their purchases.

Actors: Customers

STEPS

Customers add items to their cart and proceed to checkout.

Enters the discount code in the appropriate field.

The system verifies if the code is valid, in that case the code is applied to the total order amount.

The customer completes the purchase.

Outcome: Customers receive discounts on their orders as per the applied discount codes.

27. Add employee

Expand the library staff efficiently by adding new employees. Administrators use the admin interface to add new employee profiles.

Problem: New employees need to be registered in the system to contribute to library operations.

Actors: Administrators

STEPS

Admin logs into the system.

Navigates to the "Add Employee" section.

Enters employee details such as name, contact information, and role.

Saves the new employee profile.

Outcome: New employees are added to the system.

28. Check for availability of a book

Allow customers to quickly determine if a book is available for purchase.

Problem: Customers need to know if a book is in stock before planning to purchase it.

Actors: Customers

STEPS

Customer navigates to the book details page.

Check the availability status displayed.

Outcome: Customers know if the book is available and can proceed with purchasing accordingly.

29. Daily revenue

Track daily revenue to monitor financial performance. Members of staff (managers) use the interface to view daily revenue reports.

Problem: Daily revenue data is needed for financial reporting and decision-making.

Actors: Managers, Administrators

STEPS

User logs into the system.

Navigates to the "Revenue" section (with daily filter applied).

Selects the desired date for the revenue report.

Reviews the daily revenue report.

Outcome: Users gain insights into daily revenue generation.

Bookstore System Documentation

30. Deleting an employee

Manage staff changes by removing employee profiles when necessary.

Problem: Employee profiles may need to be deleted due to various reasons.

Actors: Manager

STEPS

Admin logs into the system.

Navigates to the "Delete Employee" section.

Selects the employee profile to be deleted.

Confirms the deletion action.

Outcomes: Employee profiles are removed from the system securely and efficiently.

Diagrams

Here's a summary of our work for the Bookstore Management System:

Use Case Diagrams and Templates:

Modeled 30 different use cases.

Filled template tables for each use case.

Divided work among team members, with each member handling 5 requirements.

Entity-Relationship Diagram (ERD):

Created an ERD to model the database structure.

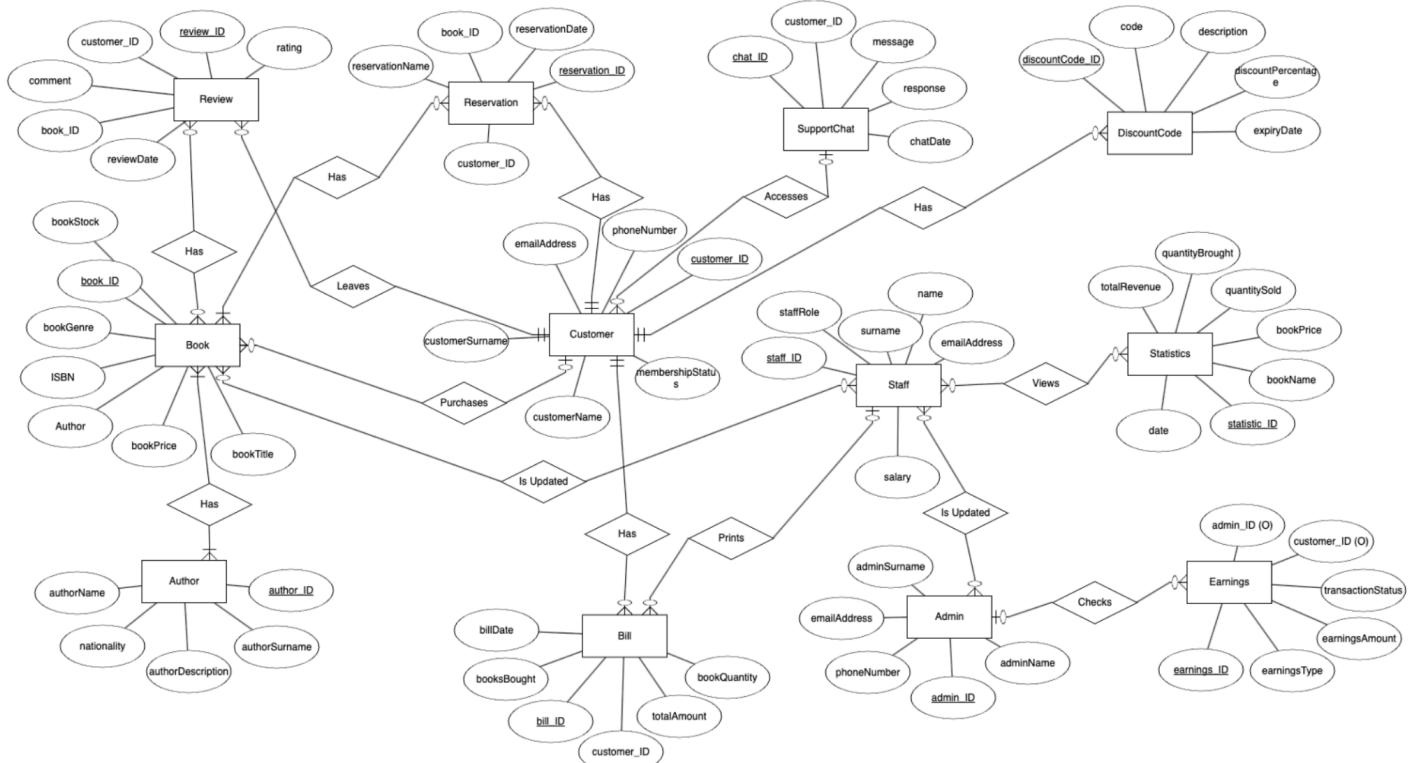
An Entity-Relationship Diagram is a graphical representation used to model the data structure of a database. It visually outlines the entities (or objects) within a system and the relationships between these entities. Here are the main components of an ERD:

Entities: represent real-world objects or concepts that have a distinct existence within the domain being modeled. Examples include a Book, Author, or Admin.

Attributes: properties or details that describe an entity. For example, the Book entity has attributes like book_ID, bookStock or ISBN.

Relationships: they illustrate how entities are connected or interact with each other. For instance, each Customer may purchase a Book.

Cardinality: it specifies the number of instances of one entity that can be associated with instances of another entity. Common cardinalities include one-to-one, one-to-many, and many-to-many.



Bookstore System Documentation

The Entity-Relationship Diagram that we have designed illustrates the database schema for our online bookstore system. The key entities are Customer, Book, Review, Reservation, SupportChat, Author, Bill, Staff, Admin, DiscountCode, Statistics, and Earnings. The relationships between these entities are clearly defined, supporting the necessary operations of the bookstore system.

Customer

Attributes: customer_ID, customerName, customerSurname, emailAddress, phoneNumber, membershipStatus

Relationships: Has many Reservations, leaves many Reviews, purchases many Books, has one Bill, has many SupportChats and is Updated by one Admin

Book

Attributes: book_ID, bookTitle, bookGenre, ISBN, Author, bookPrice, bookStock

Relationships: Has many Authors, has many Reviews, is Reserved in many Reservations and is Purchased by many Customers

Review

Attributes: review_ID, rating, comment, reviewDate, book_ID, customer_ID

Relationships: Written by one Customer

Reservation

Attributes: reservation_ID, reservationName, reservationDate, book_ID, customer_ID

Relationships: Made by one Customer

SupportChat

Attributes: chat_ID, message, response, chatDate, customer_ID

Relationships: Accessed by one Customer

Author

Attributes: author_ID, authorName, authorSurname, authorDescription, nationality

Relationships: Writes many Books

Bill

Attributes: bill_ID, billDate, booksBought, bookQuantity, totalAmount, customer_ID

Relationships: Issued to one Customer

Staff

Attributes: staff_ID, name, surname, emailAddress, salary, staffRole

Relationships: Views many Statistics and is Updated by one Admin

Admin

Attributes: admin_ID, adminName, adminSurname, emailAddress, phoneNumber

Relationships: Updates many Customers, checks many Earnings

DiscountCode

Attributes: discountCode_ID, code, description, discountPercentage, expiryDate

Relationships: Has many Bookings

Statistics

Attributes: statistic_ID, totalRevenue, quantitySold, quantityBrought, bookPrice, bookName, date

Relationships: Viewed by many Staff

Earnings

Attributes: earnings_ID, transactionStatus, earningsAmount, earningsType, customer_ID (optional), admin_ID (optional)

Relationships: Checked by one Admin and related to one Customer

Relationships

- Customer and Reservation: A customer can have multiple reservations (one-to-many relationship).
- Customer and Review: A customer can leave multiple reviews (one-to-many relationship).
- Customer and Book: A customer can purchase multiple books (many-to-many relationship).
- Customer and Bill: A customer has one bill (one-to-one relationship).
- Customer and SupportChat: A customer can have multiple support chats (one-to-many relationship).
- Customer and Admin: A customer can be updated by an admin (many-to-one relationship).
- Book and Author: A book can be written by multiple authors, and an author can write multiple books (many-to-many relationship).
- Book and Review: A book can have multiple reviews (one-to-many relationship).
- Book and Reservation: A book can be reserved multiple times (one-to-many relationship).
- Review: Each review is linked to one book and one customer (many-to-one relationship).
- Reservation: Each reservation is linked to one book and one customer (many-to-one relationship).
- SupportChat: Each support chat is linked to one customer (many-to-one relationship).
- Bill: Each bill is linked to one customer (one-to-one relationship).
- Staff and Statistics: Staff can view multiple statistics (one-to-many relationship).
- Admin and Customer: An admin can update multiple customers (one-to-many relationship).
- Admin and Earnings: An admin checks multiple earnings (one-to-many relationship).
- DiscountCode: Discount codes can be applied to bookings (one-to-many relationship).

Bookstore System Documentation

Business Process Model and Notation (BPMN):

Designed BPMN diagram to represent business processes.

Business Process Model and Notation is a graphical representation used to model business processes. It provides a standardized method to visualize and document the steps involved in a business process from start to finish. We tried to design BPMN to be easily understood by the team members and supposed stakeholders.

Flow Objects: Represent something that happens. Events are circles and can be of different types (e.g., start, intermediate, end).

Start Event: Marks the beginning of a process.

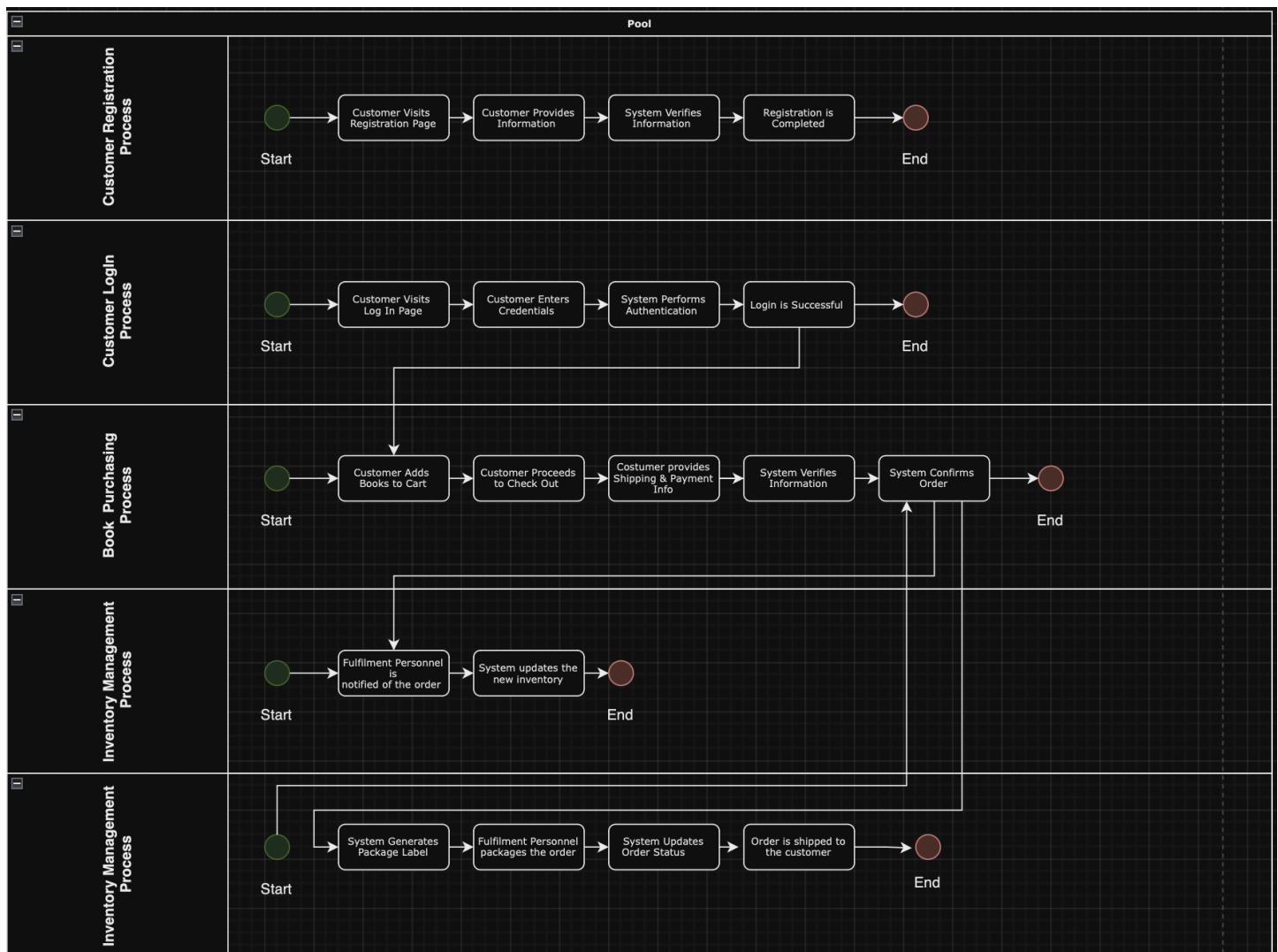
Intermediate Event: Occurs between the start and end events.

End Event: Marks the completion of a process.

Activities: Represent tasks or work that needs to be performed. Activities are rectangles with rounded corners (a task is a single unit of work and a sub-process is a complex activity that can be broken down into smaller steps)

Connecting Objects: The connection of objects can be done using sequence flows (solid lines with arrows that show the order in which activities are performed), message flows (dashed lines with arrows that show communication between different entities or processes) or associations (dotted lines used to link text to flow objects).

Swimlanes: pools represent a major participant in a process, such as a company or department and lanes subdivide pools to represent roles, responsibilities, or departments within the participant.



Bookstore System Documentation

The Business Process Model and Notation diagram we have created for our project illustrates all the processes involved in the system. The diagram is divided into four main processes: Customer Registration, Customer Login, Book Purchasing, and Inventory Management.

Customer Registration Process

Start: The process begins when the customer visits the registration page.

Customer Provides Information: The customer inputs their personal details required for registration.

System Verifies Information: The system checks the provided information for correctness and completeness.

Registration is Completed: Once verification is successful, the registration process is completed.

End: The process ends after successful registration.

Customer Login Process

Start: The process starts when the customer visits the login page.

Customer Enters Credentials: The customer inputs their login credentials (username and password).

System Performs Authentication: The system verifies the credentials to authenticate the user.

Login is Successful: If the credentials are correct, the login process is successful.

End: The process ends after the customer successfully logs in.

Book Purchasing Process

Start: The process begins when the customer adds books to their cart.

Customer Adds Books to Cart: The customer selects the books they want to purchase and adds them to the cart.

Customer Proceeds to Check Out: The customer moves to the checkout process.

Customer Provides Shipping & Payment Info: The customer inputs their shipping address and payment information.

System Verifies Information: The system verifies the provided shipping and payment details.

System Confirms Order: Once verification is successful, the system confirms the order.

End: The process ends after the order is confirmed.

Inventory Management Process

This process has two subprocesses: updating inventory and packaging orders.

Subprocess 1: Updating Inventory

Start: The process begins when the fulfillment personnel is notified of the order.

Fulfillment Personnel Notified of the Order: Notification is sent to the fulfillment team about the new order.

System Updates the New Inventory: The system updates the inventory to reflect the ordered items.

End: The process ends after the inventory is updated.

Subprocess 2: Packaging and Shipping

Start: The process starts after the inventory is updated.

System Generates Package Label: The system generates a label for the package.

Fulfillment Personnel Packages the Order: The fulfillment team packages the ordered items.

System Updates Order Status: The system updates the status of the order to indicate it has been packaged.

Order is Shipped to the Customer: The package is shipped to the customer.

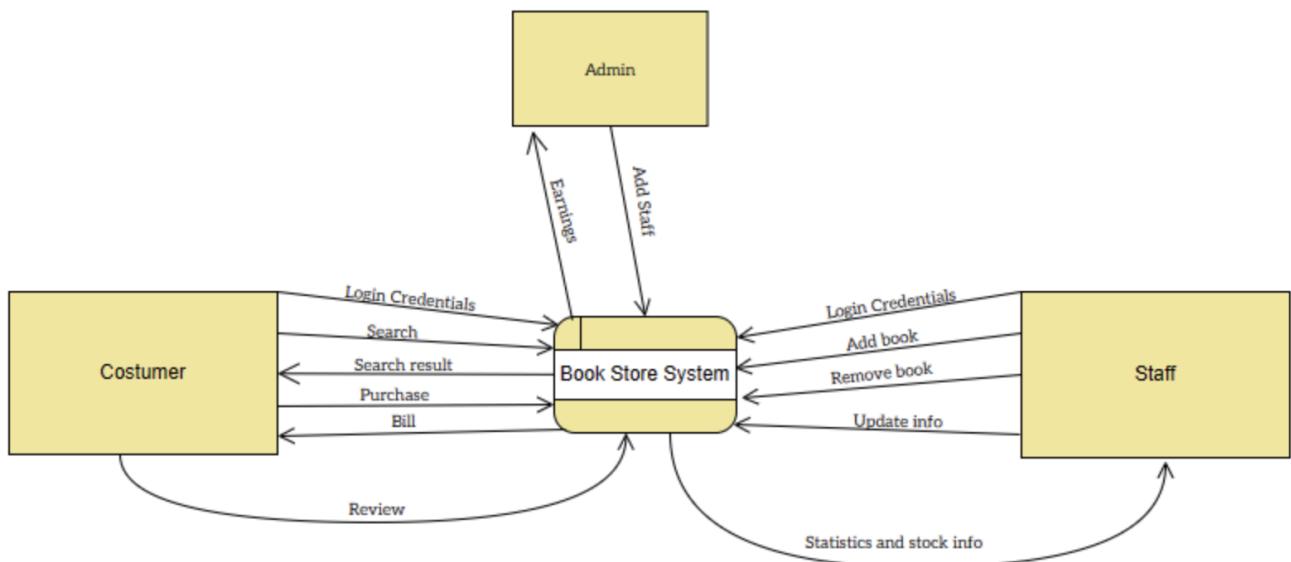
End: The process ends after the order is shipped.

Bookstore System Documentation

Data Flow Diagrams (DFD):

Created Level 0, Level 1, and Level 2 DFDs to show data flow within the system. A Data Flow Diagram is a graphical representation of the flow of data within a system. It illustrates how data is processed by a system in terms of inputs and outputs.

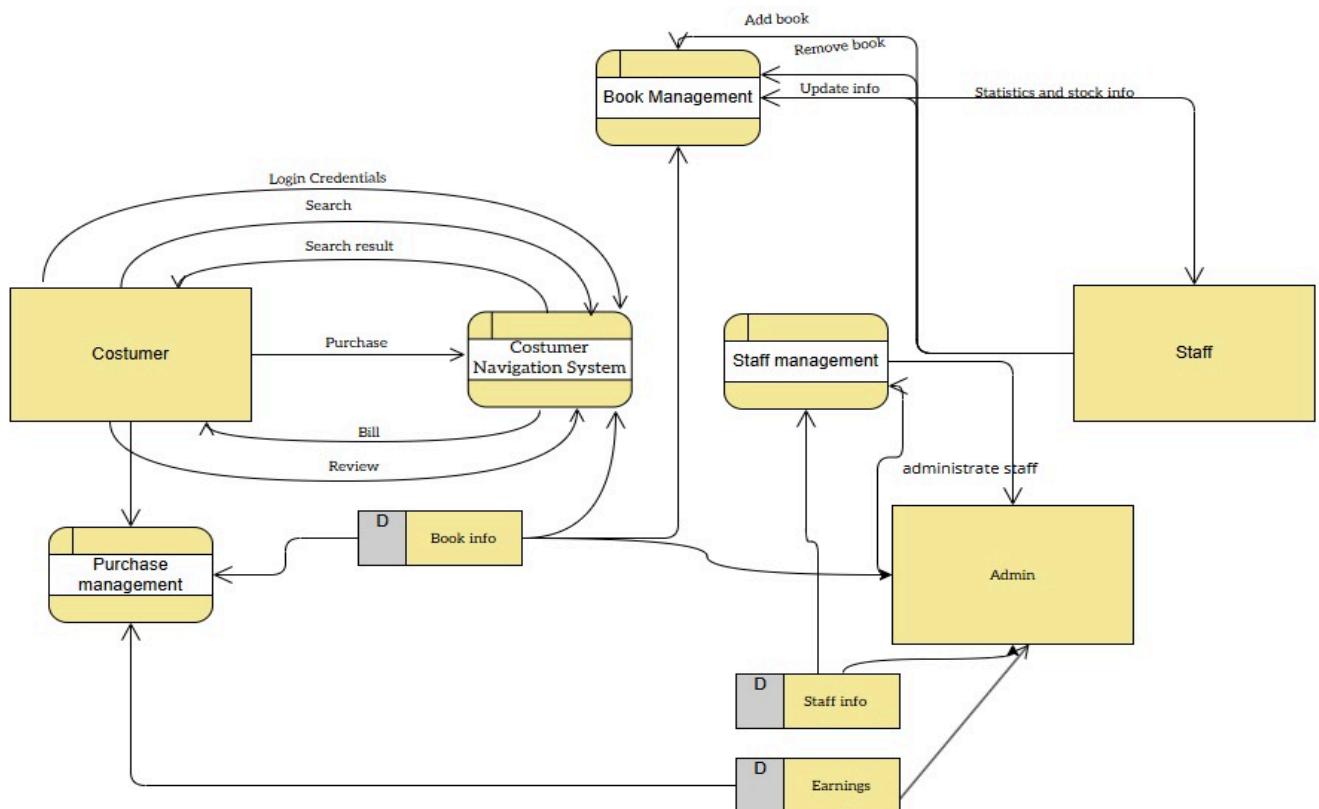
A Level 0 DFD, also known as a Context Diagram, provides a high-level overview of the entire system. It shows the overall system as a single process (usually in the center) with external entities that interact with the system and major data flows which show the input and output data to/from the system.



A Level 1 DFD breaks down the single process of the Level 0 DFD into sub-processes. This provides a more detailed view of the major functions within the system. Each sub-process in the Level 1 DFD represents a major system function.

Data Flows: Show the flow of data between sub-processes and external entities.

Data Stores: Represent places where data is stored within the system.

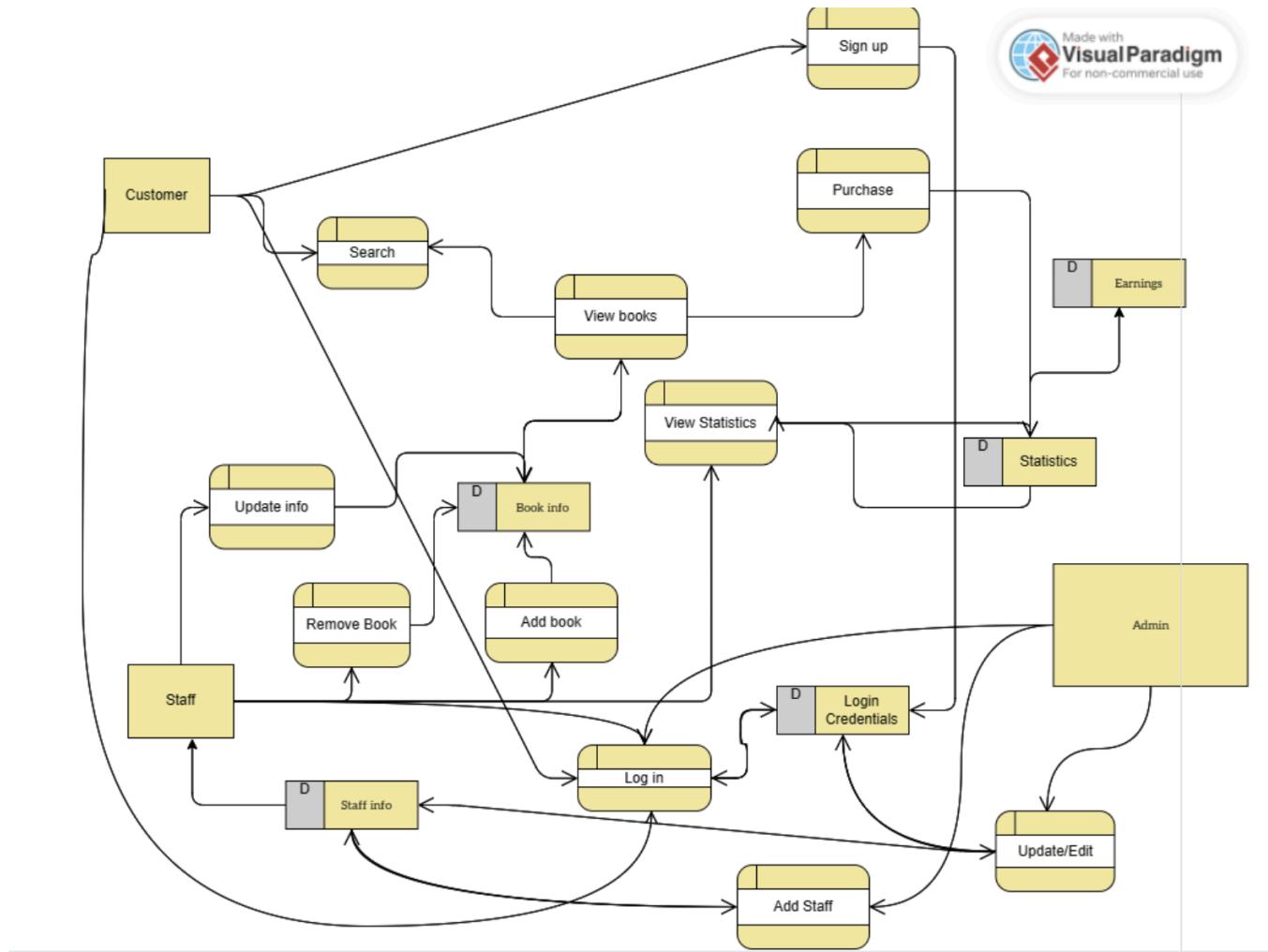


Bookstore System Documentation

A Level 2 DFD provides even more detail by breaking down each Level 1 sub-process into its own set of sub-processes. This further decomposition continues until the desired level of detail is achieved. Level 2 DFDs show specific operations within each Level 1 sub-process.

Data Flows: Show the flow of data between more granular sub-processes.

Data Stores: Same as in previous levels, showing where data is stored.



Unified Use Case and Class Diagram:

Collaboratively created one comprehensive Use Case Diagram representing the entire system.
Developed one Class Diagram for the whole system.

A Use Case Diagram is a visual representation of the functional requirements of a system. It shows the interactions between users (actors) and the system, highlighting what the system does from the perspective of the user.

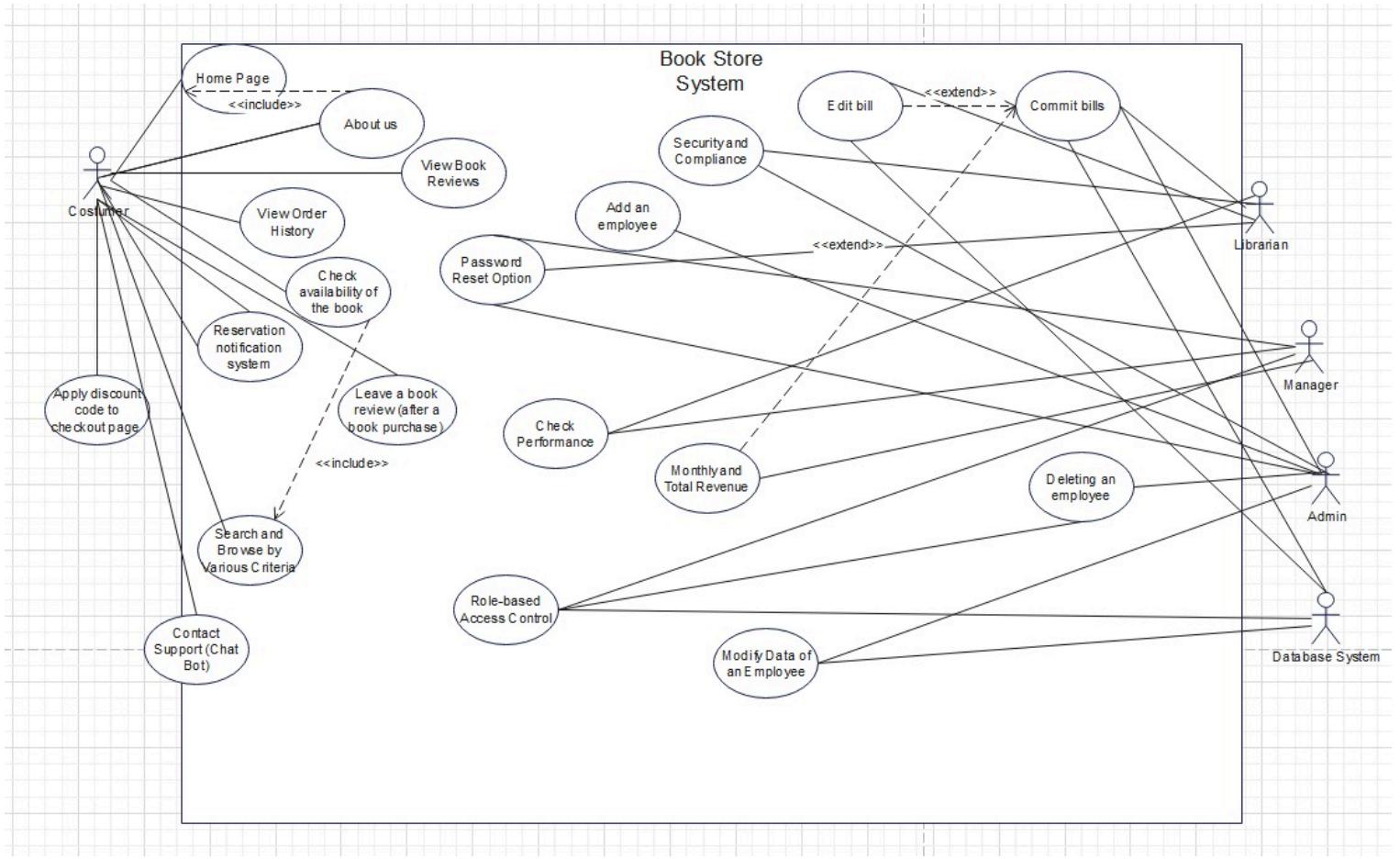
Actors: represent entities (people, organizations, external systems) that interact with the system, represented as stick figures.

Use Cases: represent specific functionalities or services provided by the system to the actors, represented as ovals.

System Boundary: defines the scope of the system, showing what is inside (part of the system) and what is outside (actors), represented as a rectangle enclosing the use cases.

Relationships: Associations (lines connecting actors to use cases, showing interaction), Include (represents mandatory use case behavior included within another use case), Extend (represents optional use case behavior that extends the base use case) and Generalization (represents inheritance between actors or use cases)

Bookstore System Documentation



The use case diagram provided represents the functionalities of a Book Store System, illustrating the interactions between different actors and the system's use cases. Here's a detailed explanation:

Actors:

Customer: The end-user who interacts with the bookstore system to see books to purchase.

Librarian: The person responsible for managing the book inventory and processing bills.

Manager: The person who oversees the overall operations, including performance checks and revenue tracking.

Admin: The administrator who manages employee data and access control.

Database System: The backend system that supports data management for the bookstore.

Use Cases for Customer:

Home Page: The starting point for the customer.

View Order History: Customers can see their past orders.

Check Availability of the Book: Customers can check if a specific book is available.

Reservation Notification System: Notify customers when a reserved book becomes available.

Search and Browse by Various Criteria: Customers can search for books based on different criteria.

Apply Discount Code to Checkout Page: Use discount codes during the checkout process.

Contact Support (Chat Bot): Interact with customer support via a chat bot.

View Book Reviews: Read reviews left by other customers.

Leave a Book Review (after a book purchase): Customers can leave reviews for books they have purchased.

About Us: Provides information about the bookstore.

Password Reset Option: Allows customers to reset their passwords.

Use Cases for Librarian:

Commit Bills: Process and finalize billing transactions.

Edit Bill: Modify billing details if necessary (extends Commit Bills).

Security and Compliance: Ensuring that billing processes comply with security and regulatory standards.

Use Cases for Manager:

Check Performance: Monitor the performance metrics of the bookstore.

Daily, Monthly and Total Revenue: Track and analyze revenue data.

See statistics with daily, monthly and total filter.

Use Cases for Admin:

Add an Employee: Add new employees to the system.

Modify Data of an Employee: Update employee information.

Deleting an Employee: Remove employees from the system.

Role-based Access Control: Manage access permissions for different roles.

Relationships:

Include Relationship:

- Home Page includes About Us, View Book Reviews, View Order History, Check Availability of the Book, Search and Browse by Various Criteria, and Password Reset Option.
- Leave a Book Review (after a book purchase) includes View Book Reviews.

Extend Relationship:

- Edit Bill extends Commit Bills.
- Security and Compliance extends Commit Bills.

A Class Diagram is a static structure diagram that describes the structure of a system by showing its classes, their attributes, methods, and the relationships among objects.

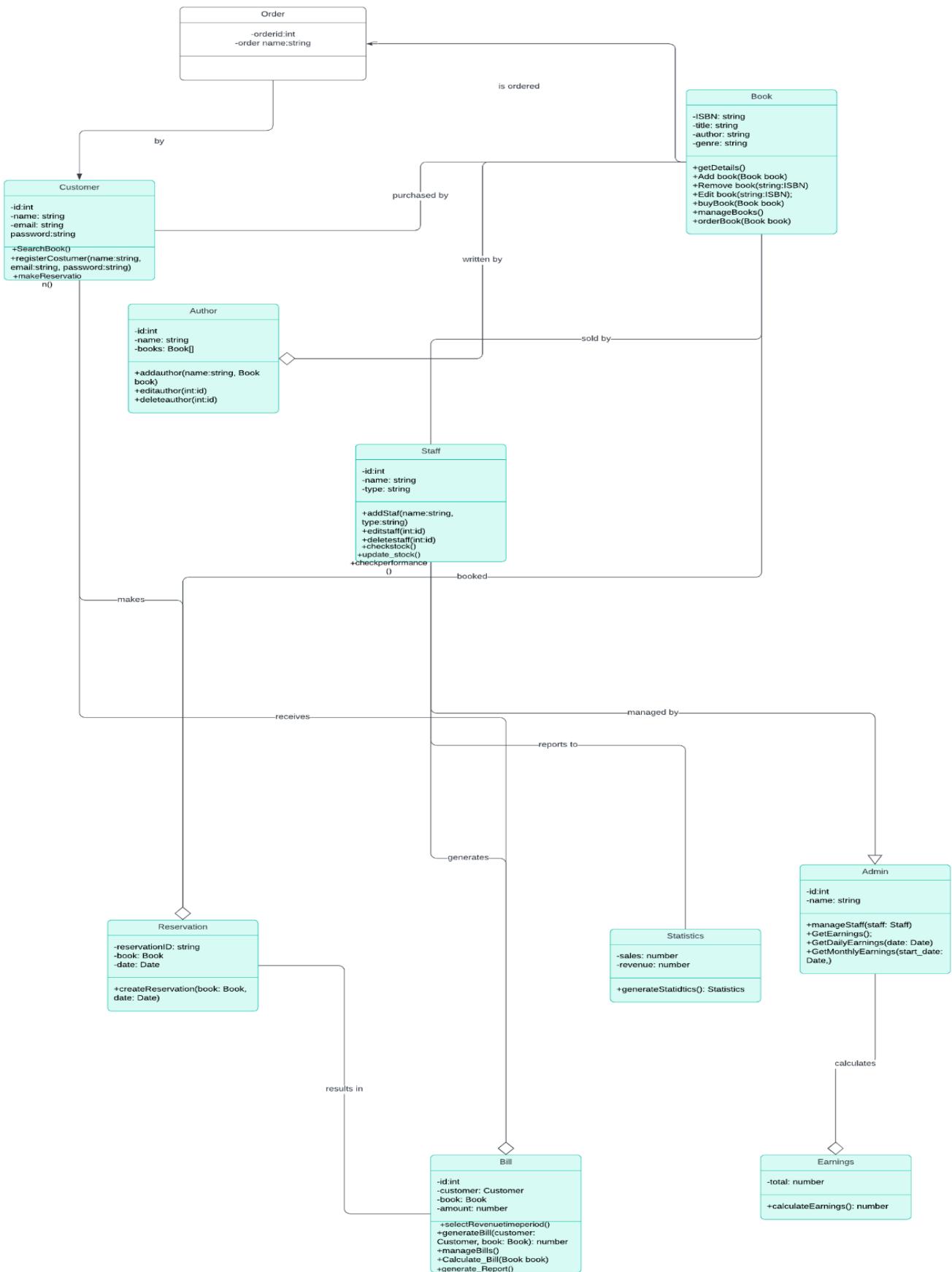
Classes: represent objects or concepts within the system, encapsulating data and behavior, represented as rectangles divided into three sections (Class Name, Attributes, Methods).

Attributes: represent the properties or data held by a class, represented in the second section of the class rectangle.

Methods: represent the operations or functions that the class can perform, represented in the third section of the class rectangle.

Relationships: Association (represents a relationship between classes (lines connecting classes), Aggregation (represents a whole-part relationship - hollow diamond at the whole end), Composition (represents a stronger whole-part relationship where parts cannot exist independently - filled diamond at the whole end), Inheritance/Generalization (represents an "is-a" relationship - triangle pointing to the parent class) and Multiplicity (indicates how many instances of a class are related to one instance of another class).

Bookstore System Documentation



Bookstore System Documentation

State Diagrams:

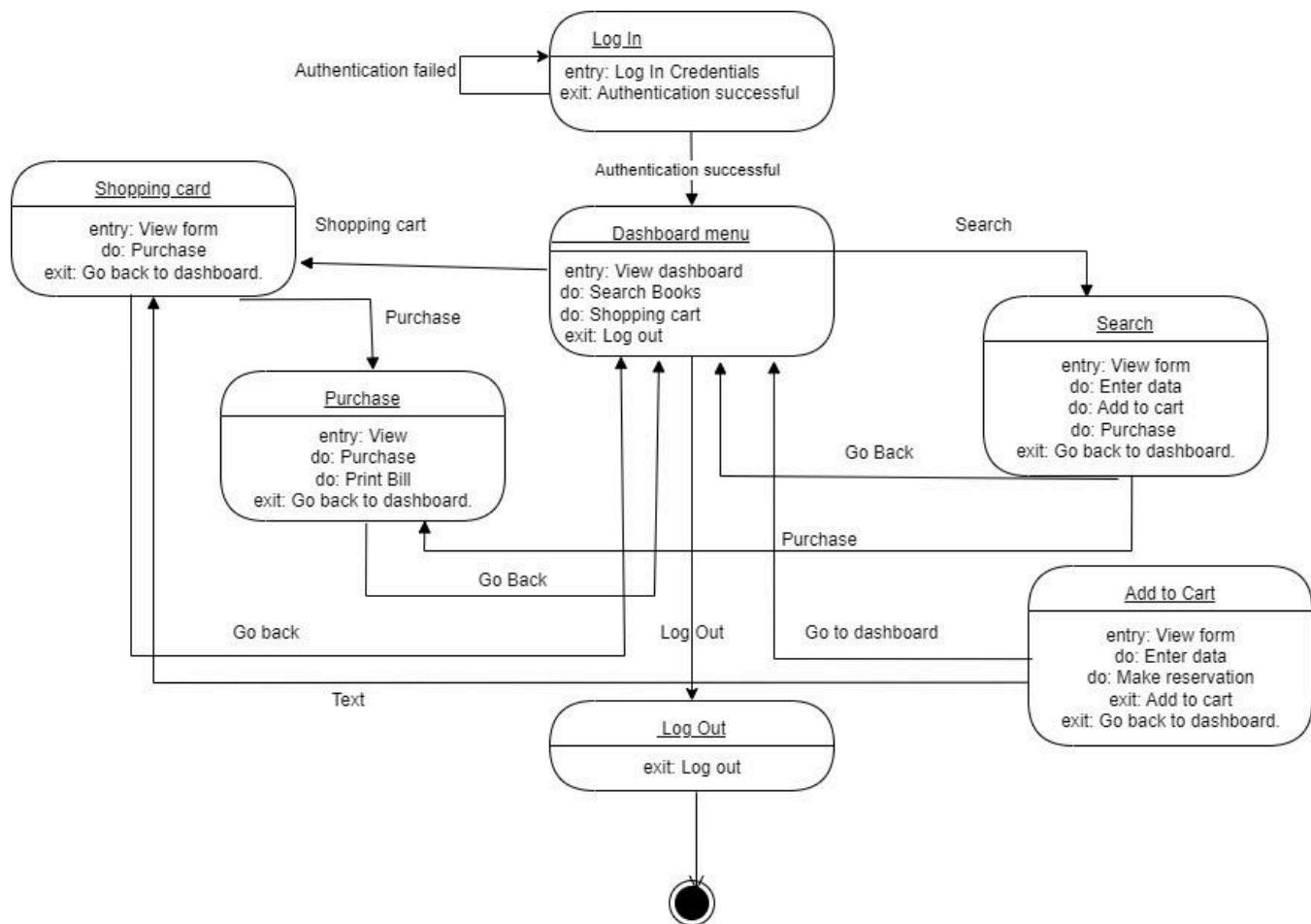
Created state diagrams for each interface.

Developed state diagrams for each user level.

A state diagram, also known as a state machine diagram or state chart diagram, is a type of behavioral diagram in UML (Unified Modeling Language) used to model the behavior of a system or component over time. It represents the possible states of an object or system and the transitions between those states triggered by events.

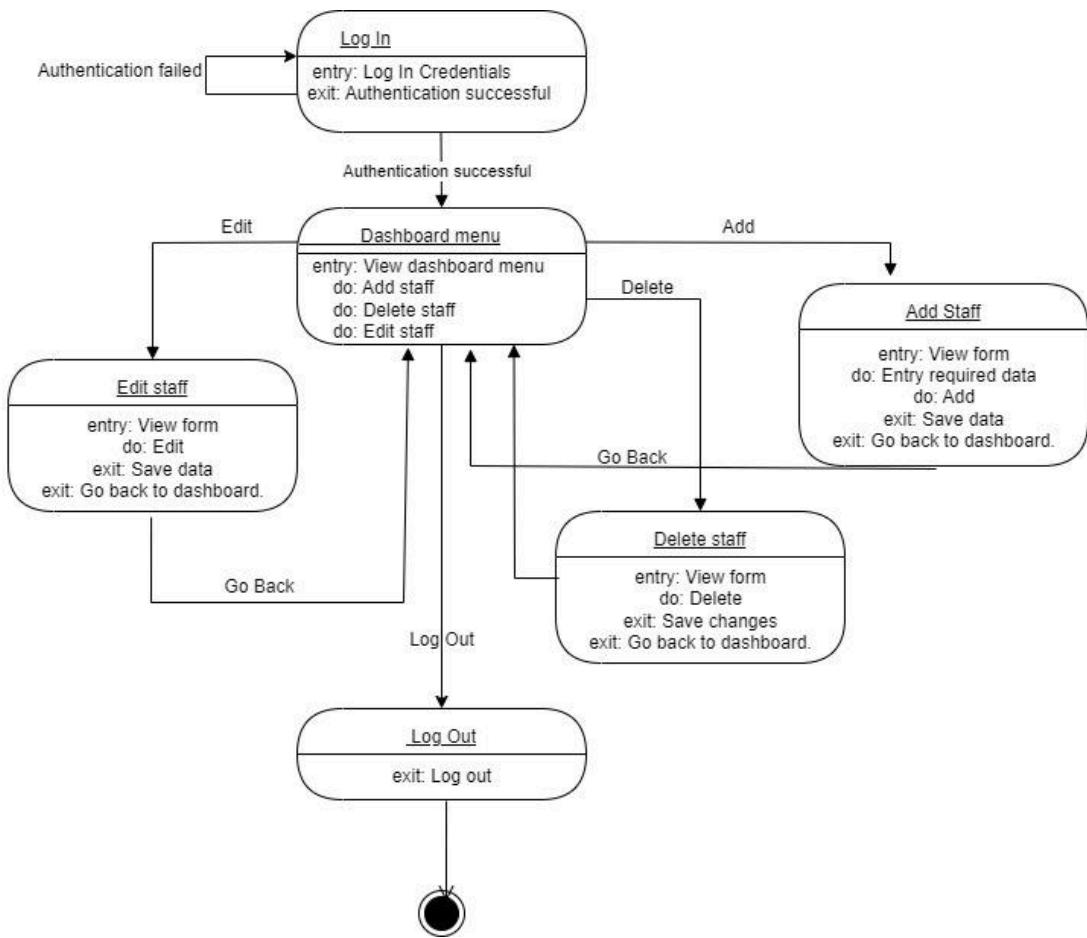
Each user level interacts with the system to transition the order state according to their respective roles and permissions. By dividing the state diagram into different user levels, we can clearly define the responsibilities and actions of each user type within the Bookstore Management System.

Customer Level:

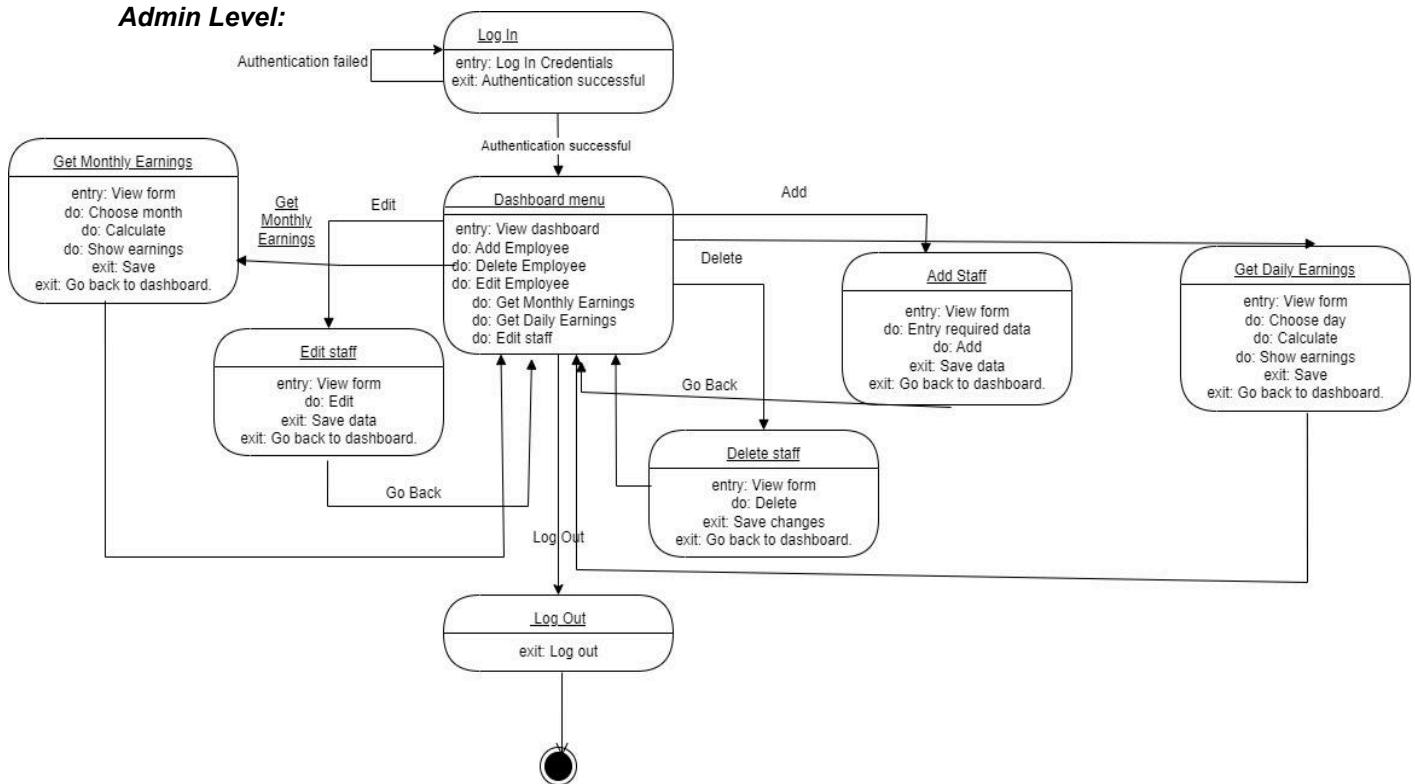


Bookstore System Documentation

Staff Level:



Admin Level:



Bookstore System Documentation

Individual Work on Activity, Sequence, and Collaboration Diagrams:

Each team member individually worked on:

- Sequence Diagrams to show the interaction sequences.

A Sequence Diagram is a type of interaction diagram in UML that shows how objects interact in a particular sequence of time. It focuses on the order of messages exchanged between objects.

Lifeline: represents an object or an actor, represented as a vertical dashed line.

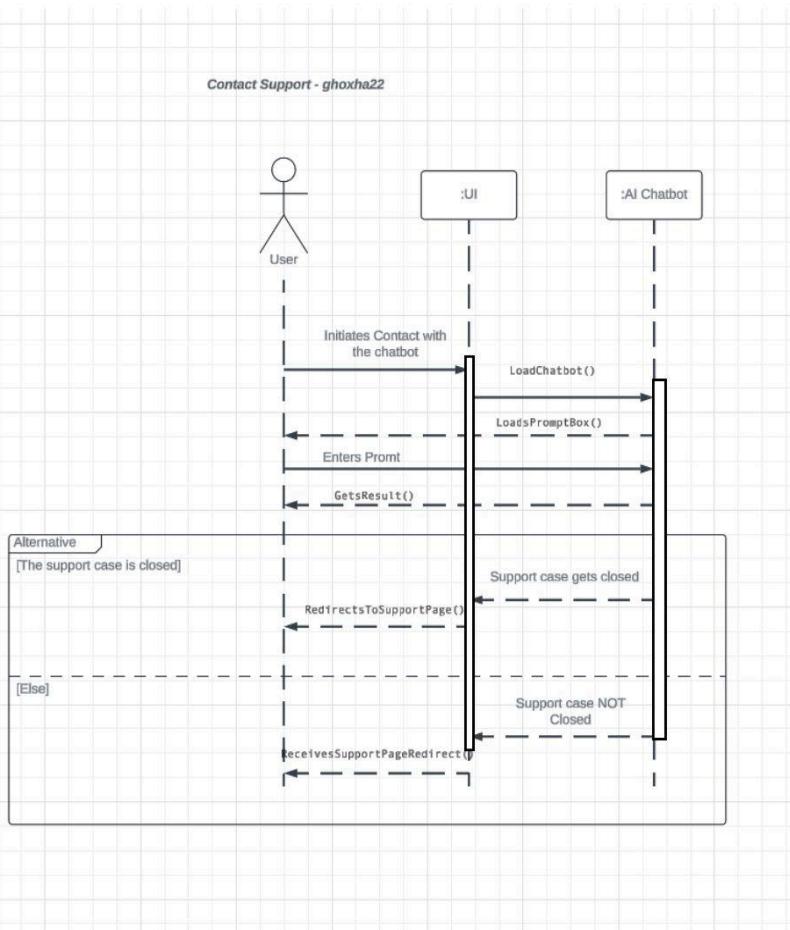
Activation Bar: indicates the period during which an object is performing an action, represented as a thin vertical rectangle on a lifeline.

Message: represents communication between objects, represented as arrows.

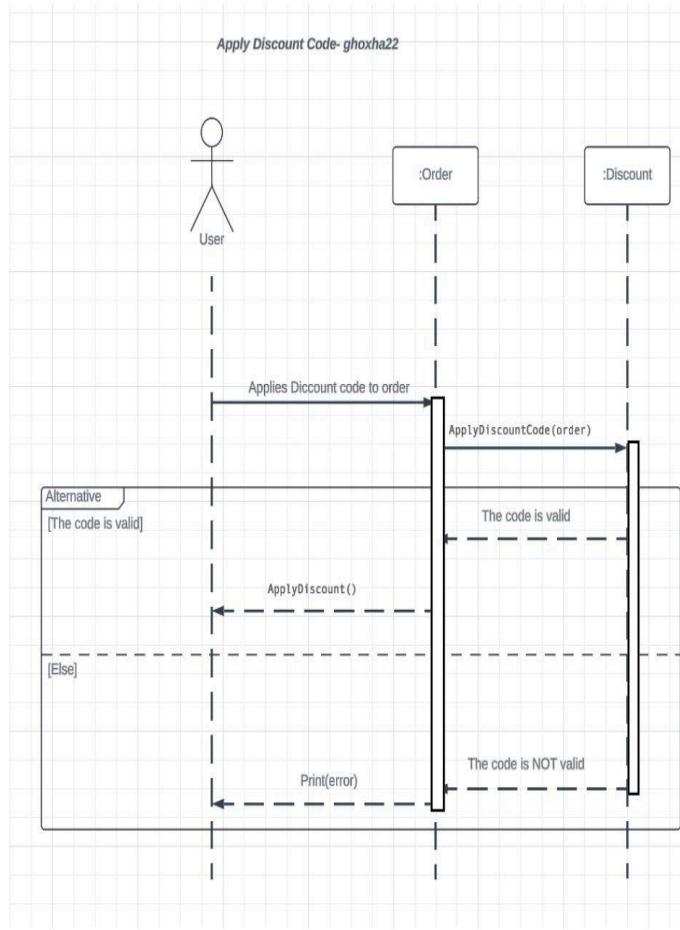
Synchronous Message: Line with a filled arrowhead - **Asynchronous Message:** Line with an open arrowhead - **Return Message:** Dotted line with an open arrowhead.

Actor: represents an external entity that interacts with the system, represented as a stick figure.

Contact Support - ghoxha22

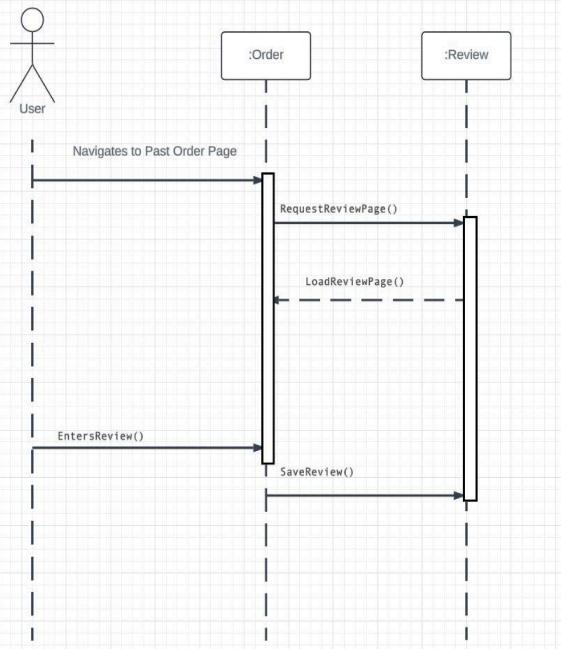


Apply Discount Code- ghoxha22

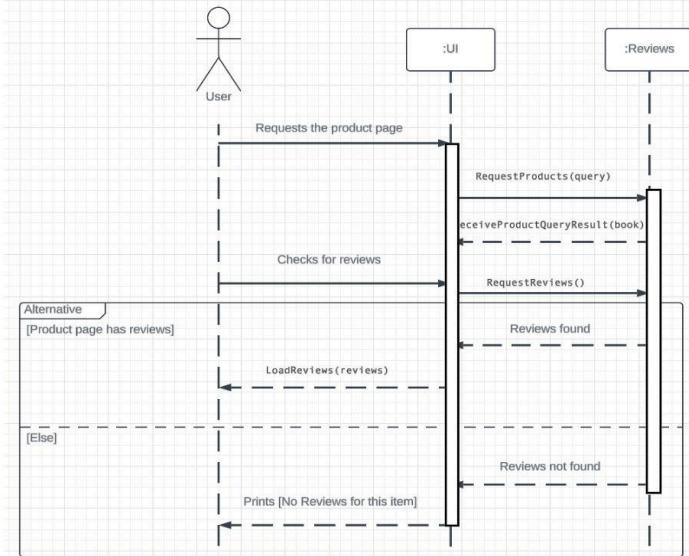


Bookstore System Documentation

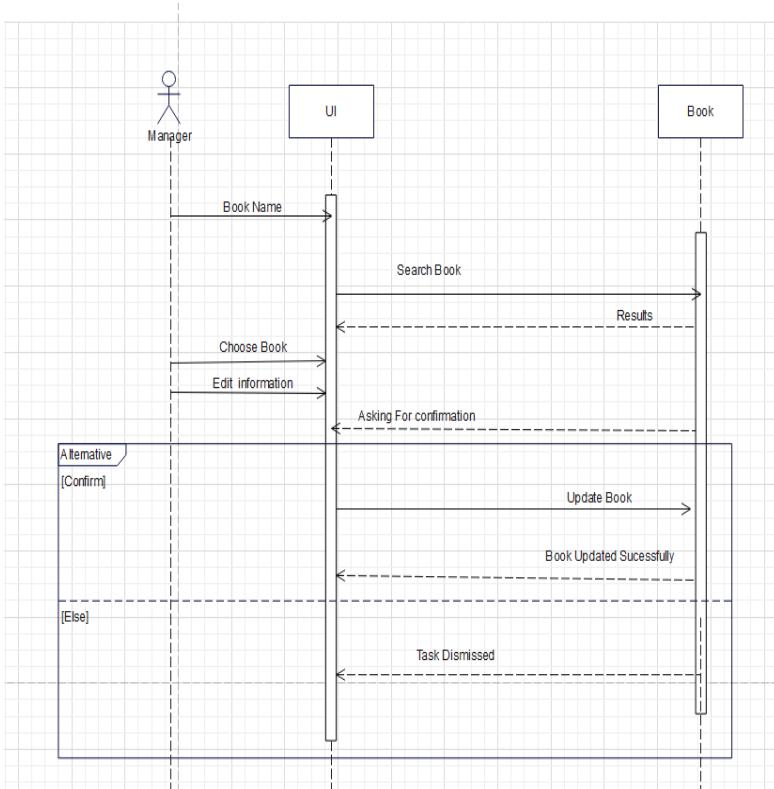
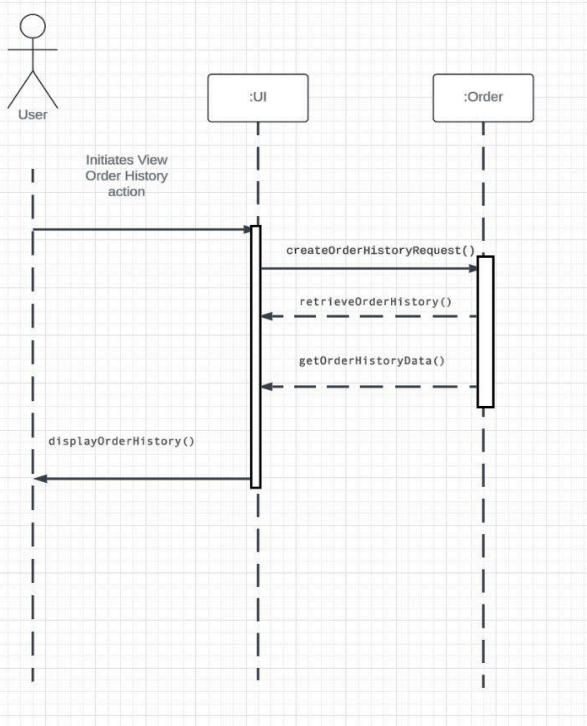
Leave a review - ghoxha22



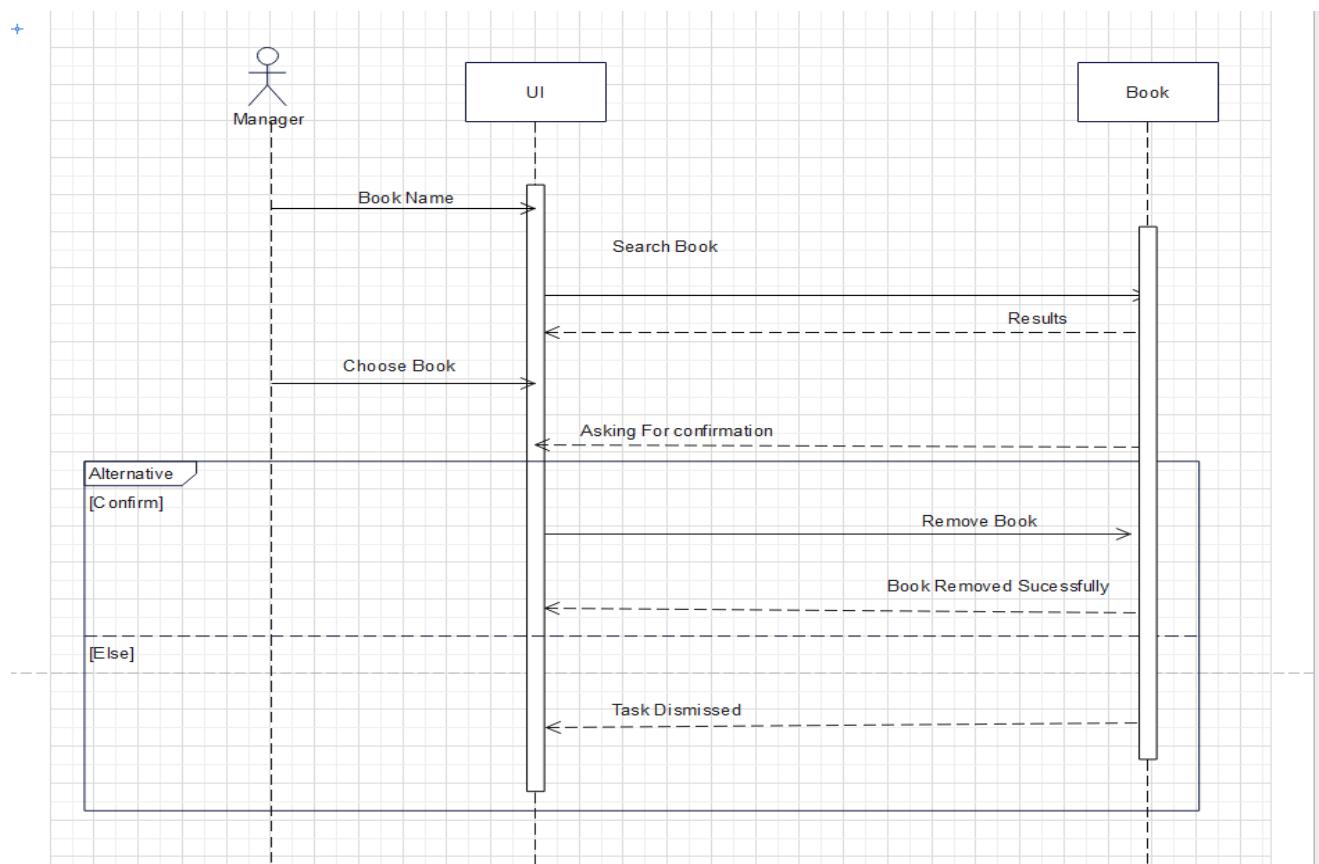
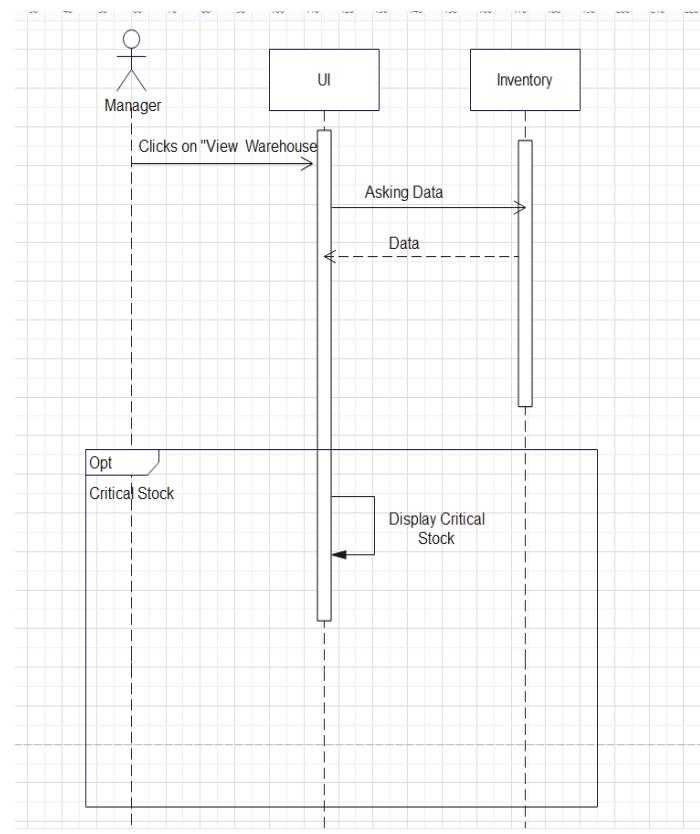
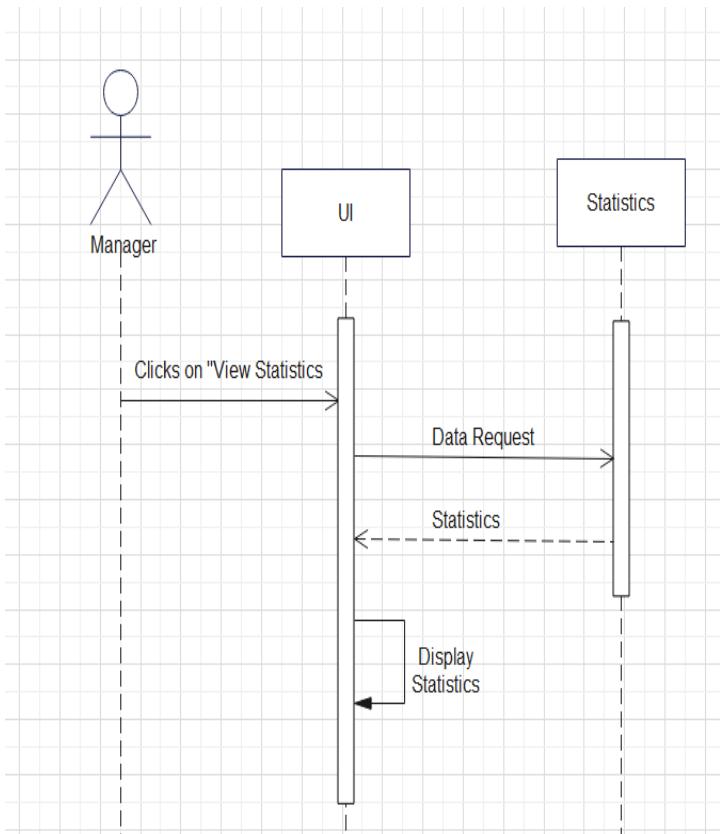
View Customer Reviews - ghoxha22



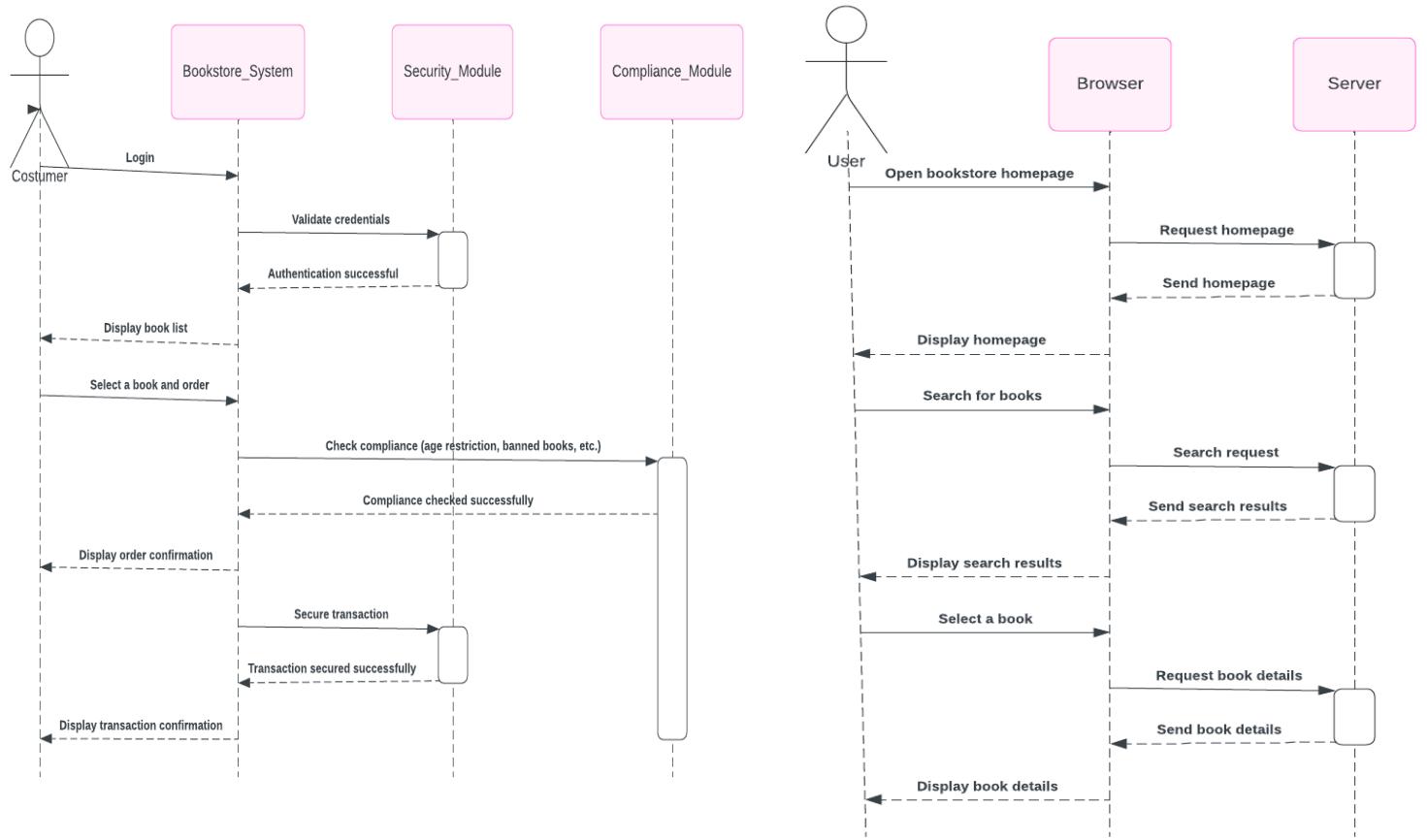
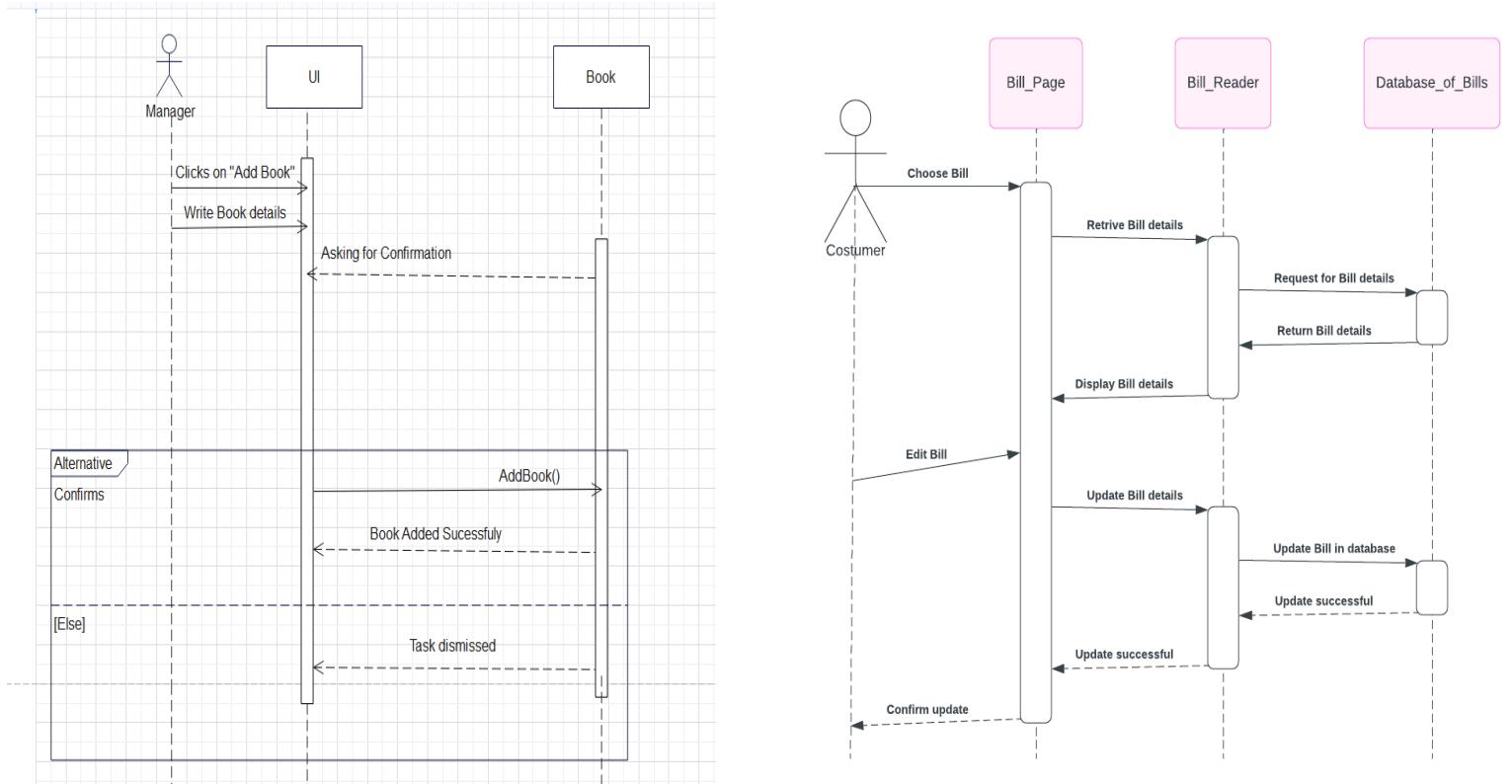
View Order History - ghoxha22

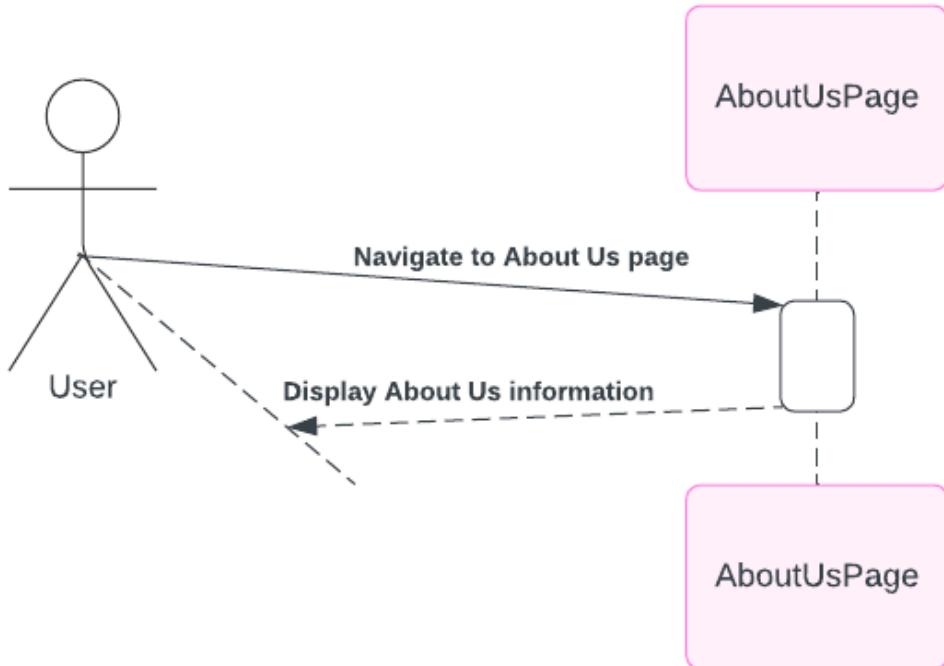


Bookstore System Documentation



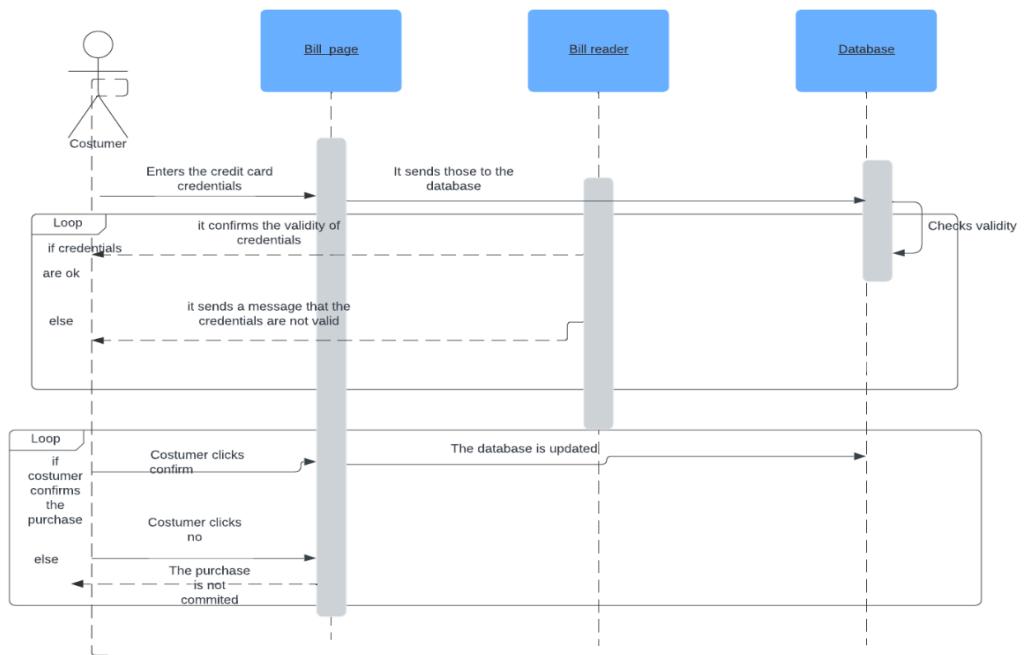
Bookstore System Documentation



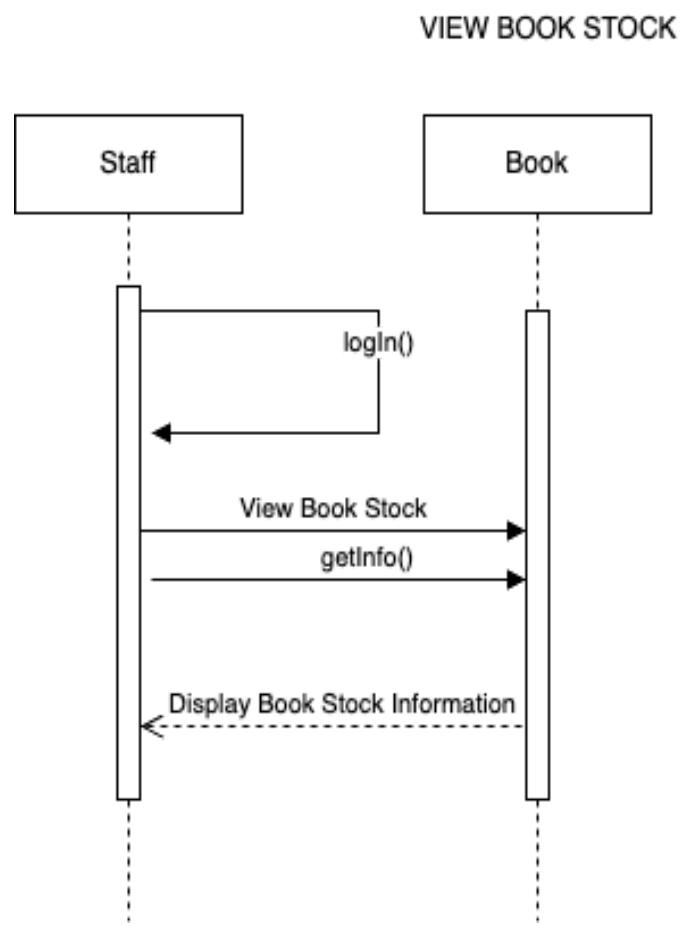
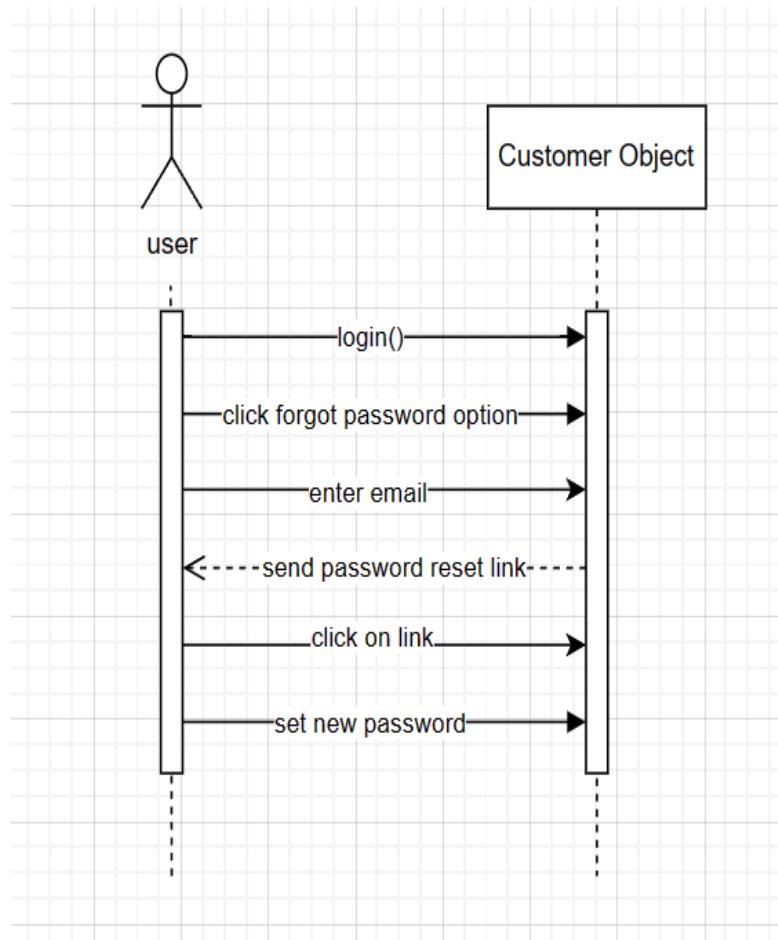
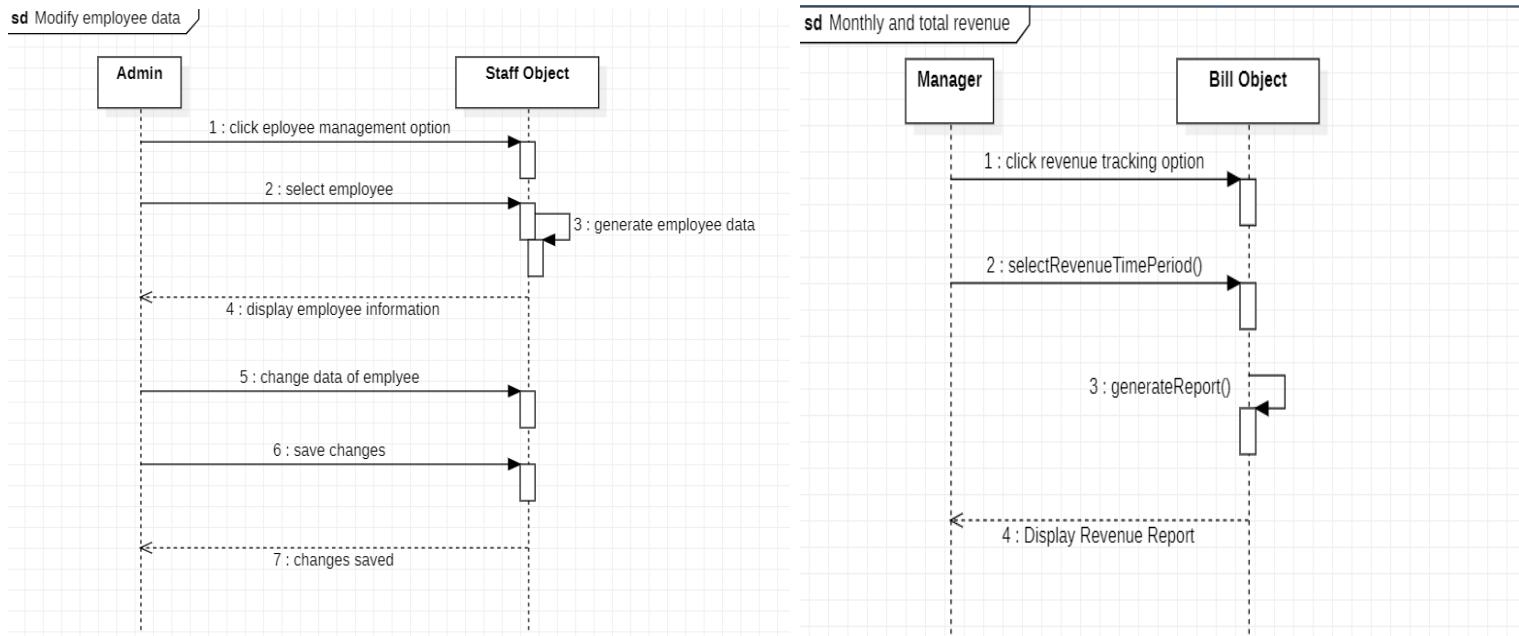


Sequence diagrams

Create a sequence diagram to model the logic for a sophisticated procedure, function, or operation. Watch a basic tutorial and read more in this help center article.

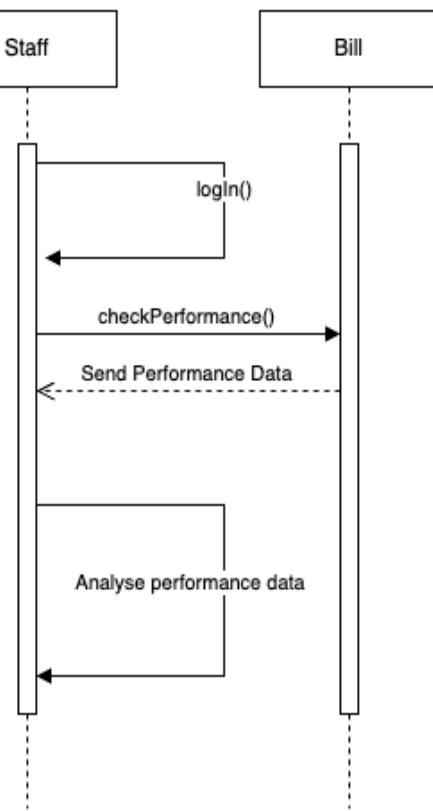


Bookstore System Documentation

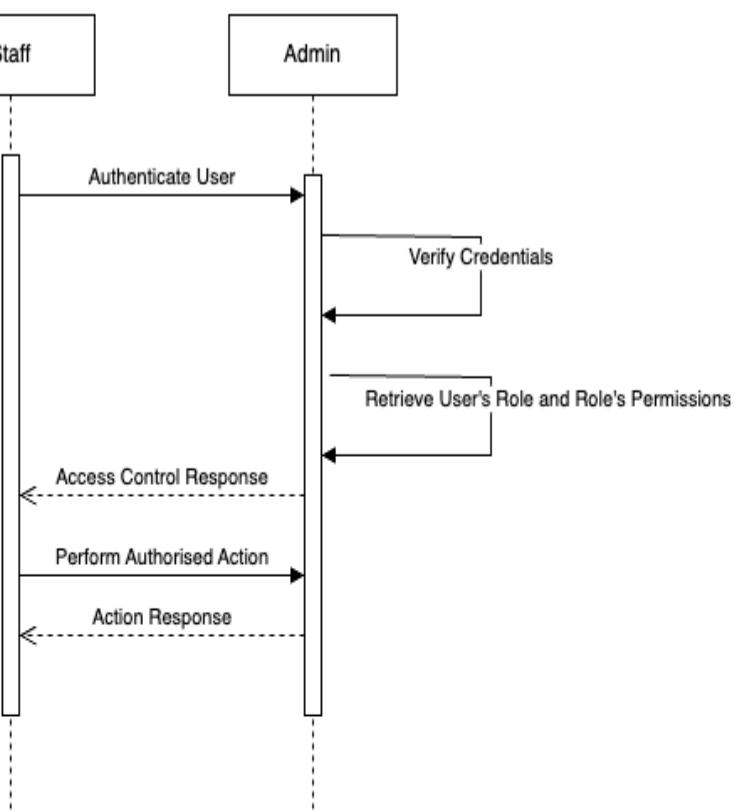


Bookstore System Documentation

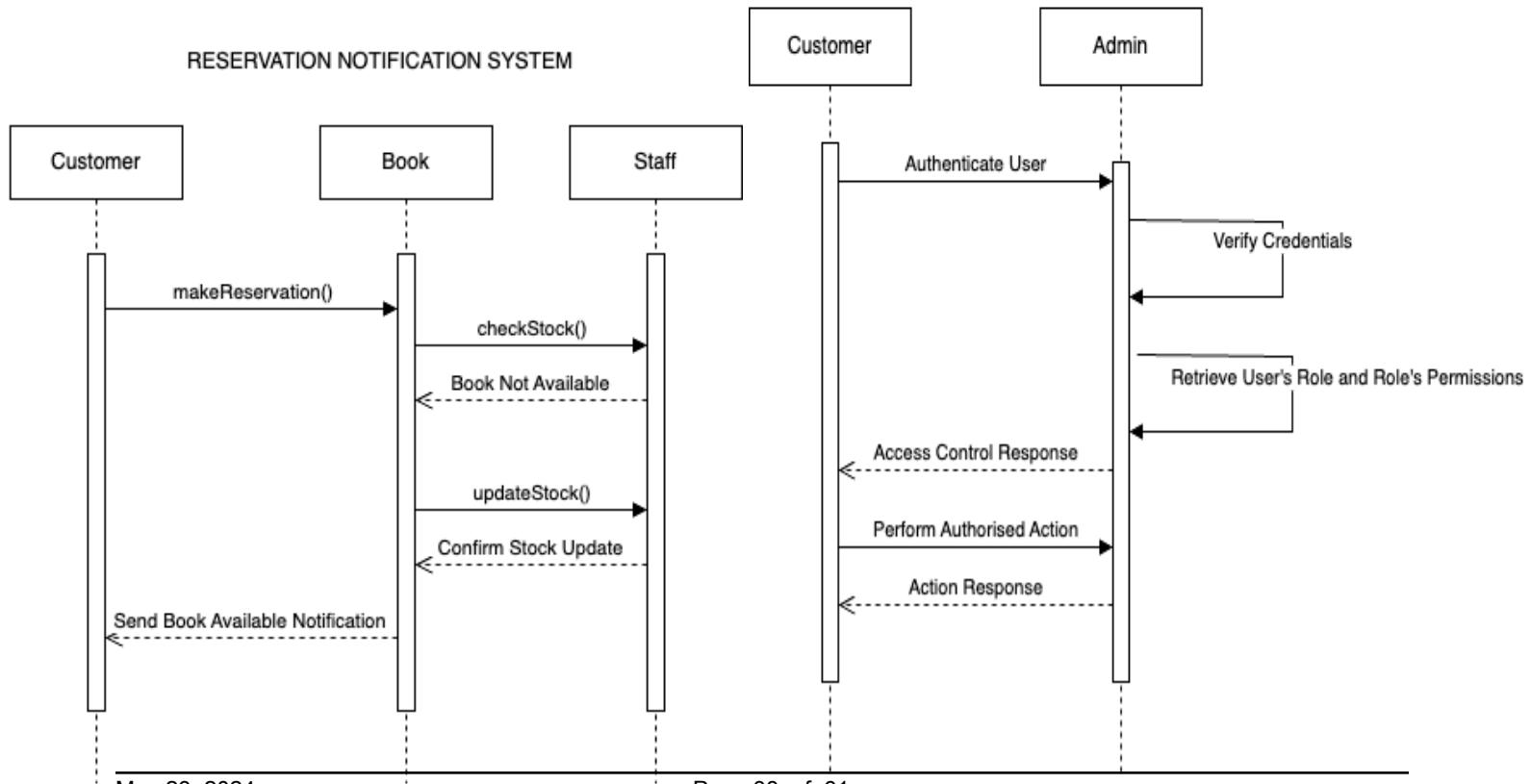
CHECK PERFORMANCE



ROLE-BASED ACCESS CONTROL

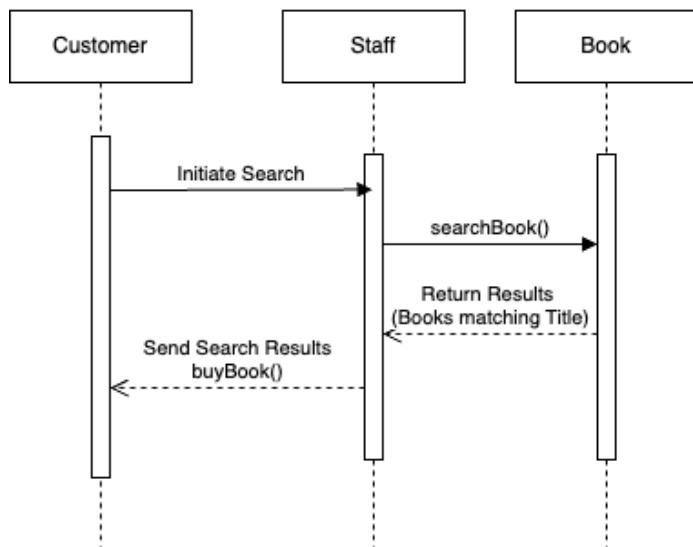


RESERVATION NOTIFICATION SYSTEM

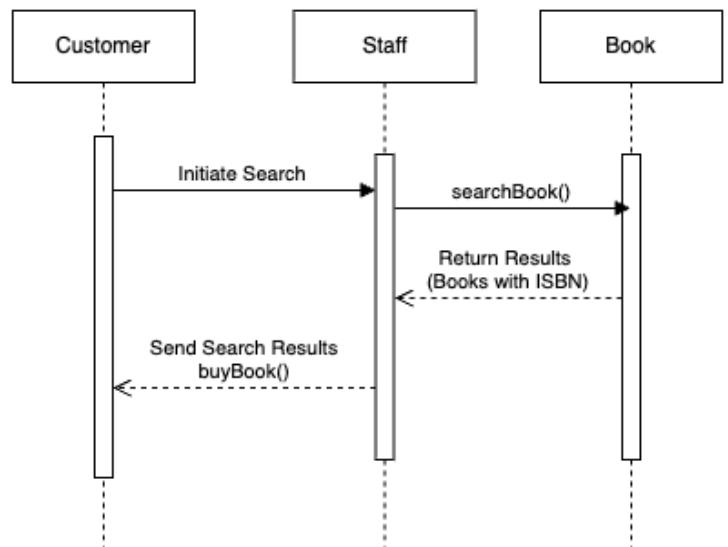


Bookstore System Documentation

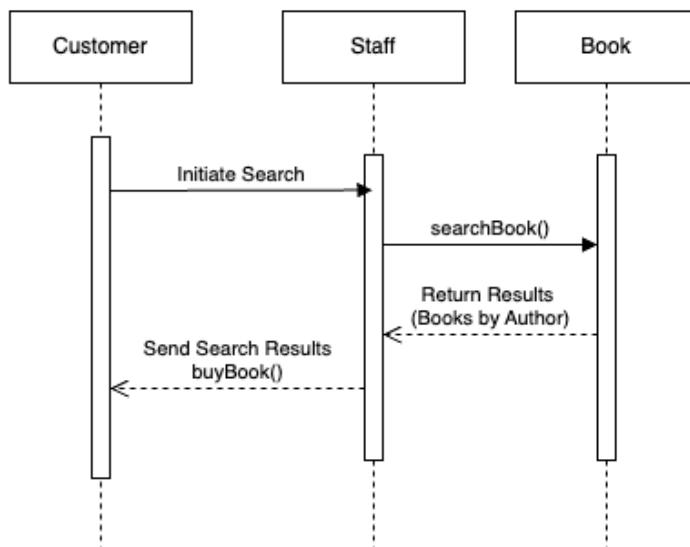
SEARCH AND BROWSE BY TITLE CRITERIA



SEARCH AND BROWSE BY ISBN CRITERIA

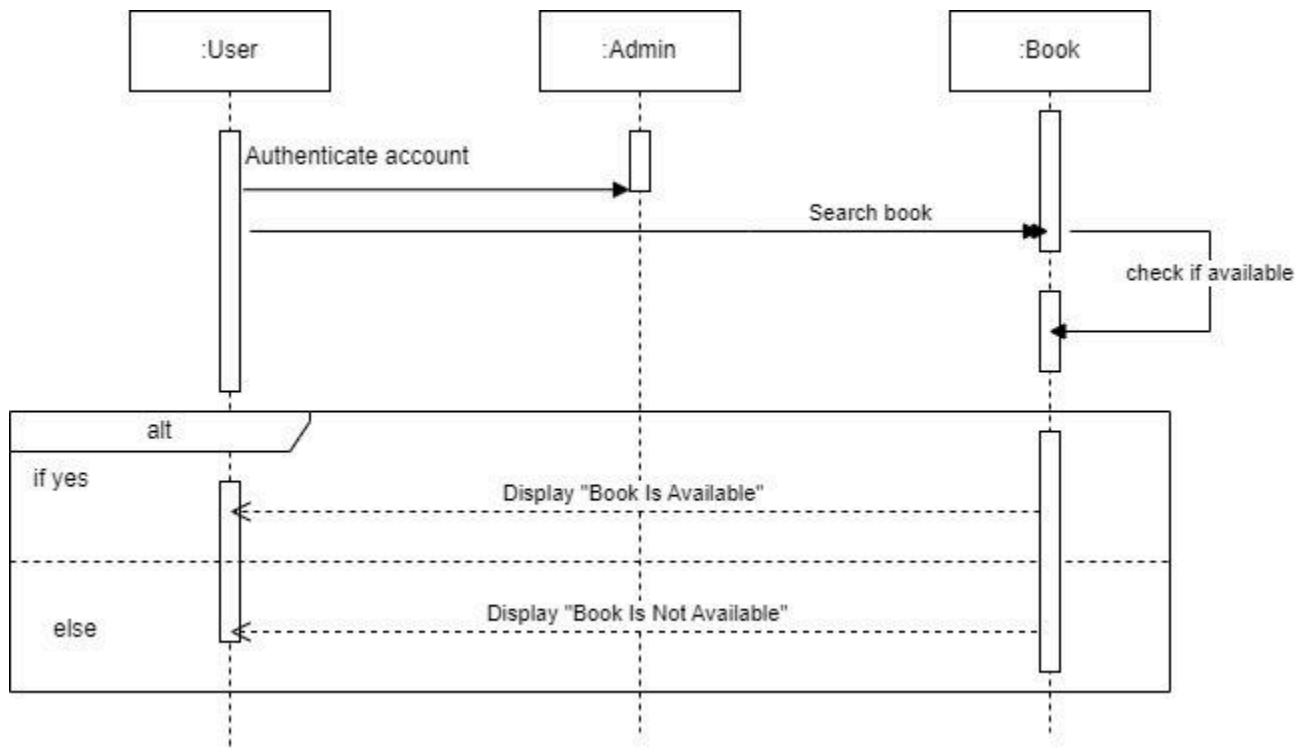


SEARCH AND BROWSE BY AUTHOR CRITERIA

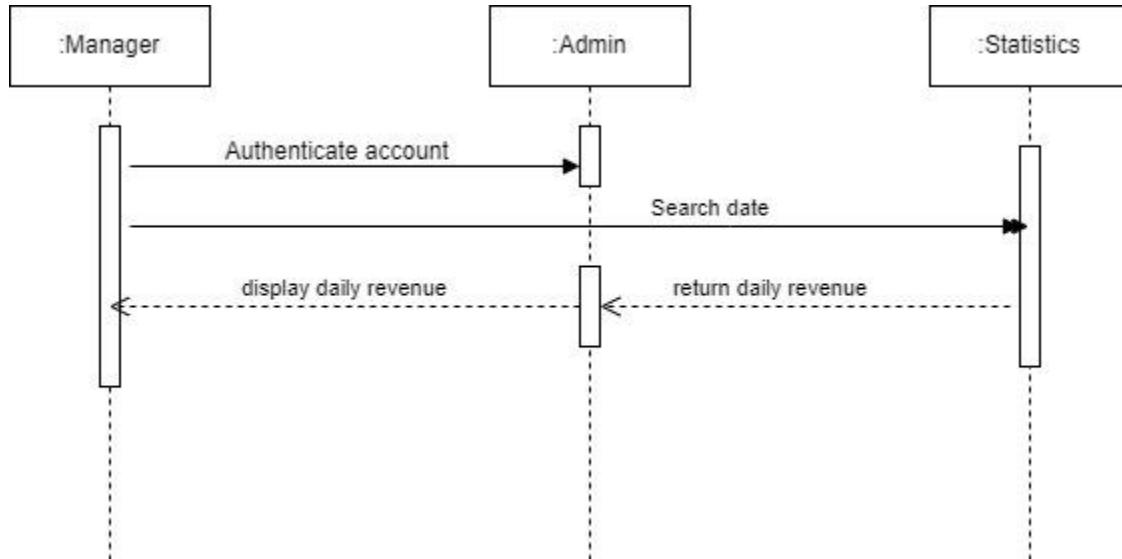


Bookstore System Documentation

Check if a book is available

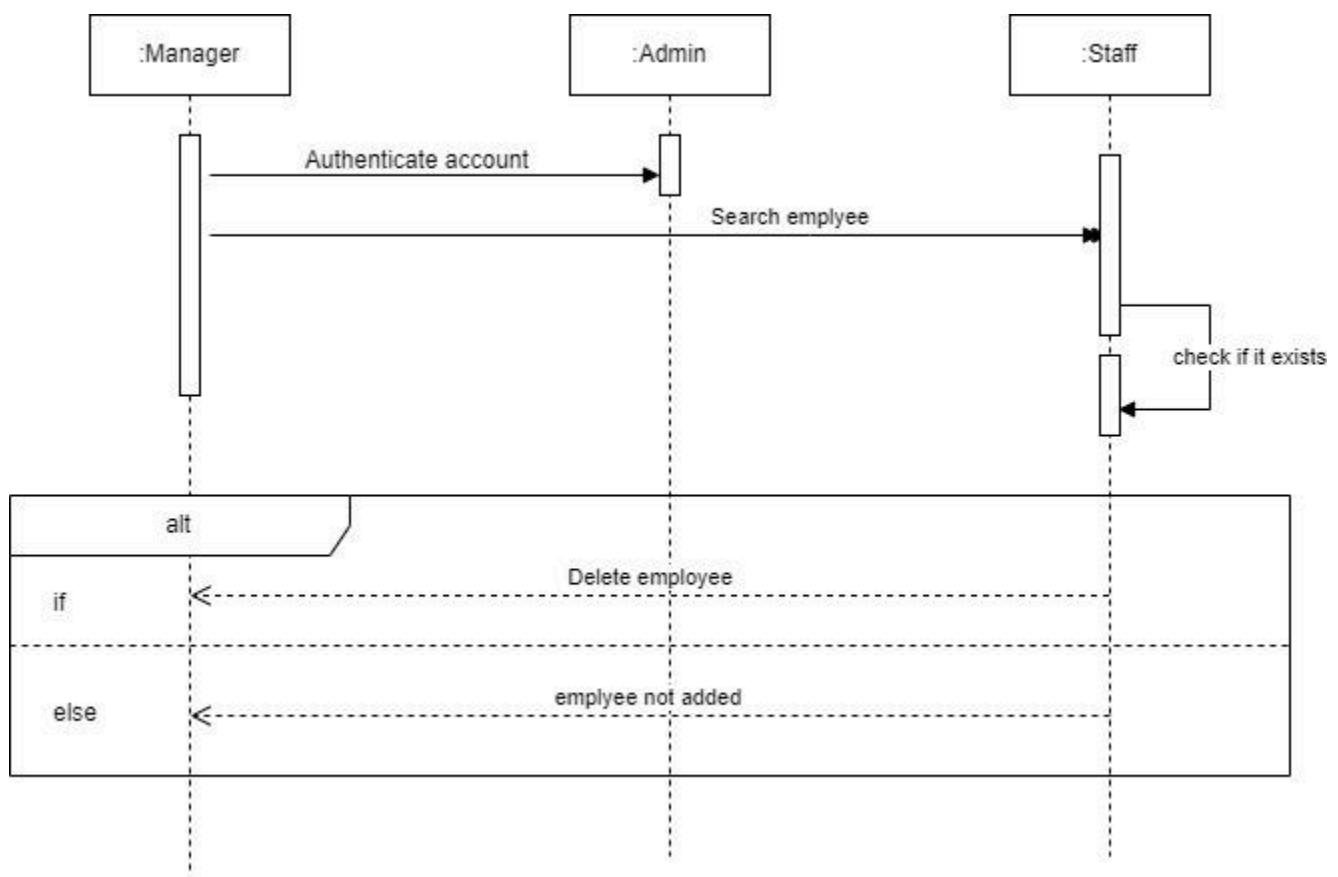


Daily revenue

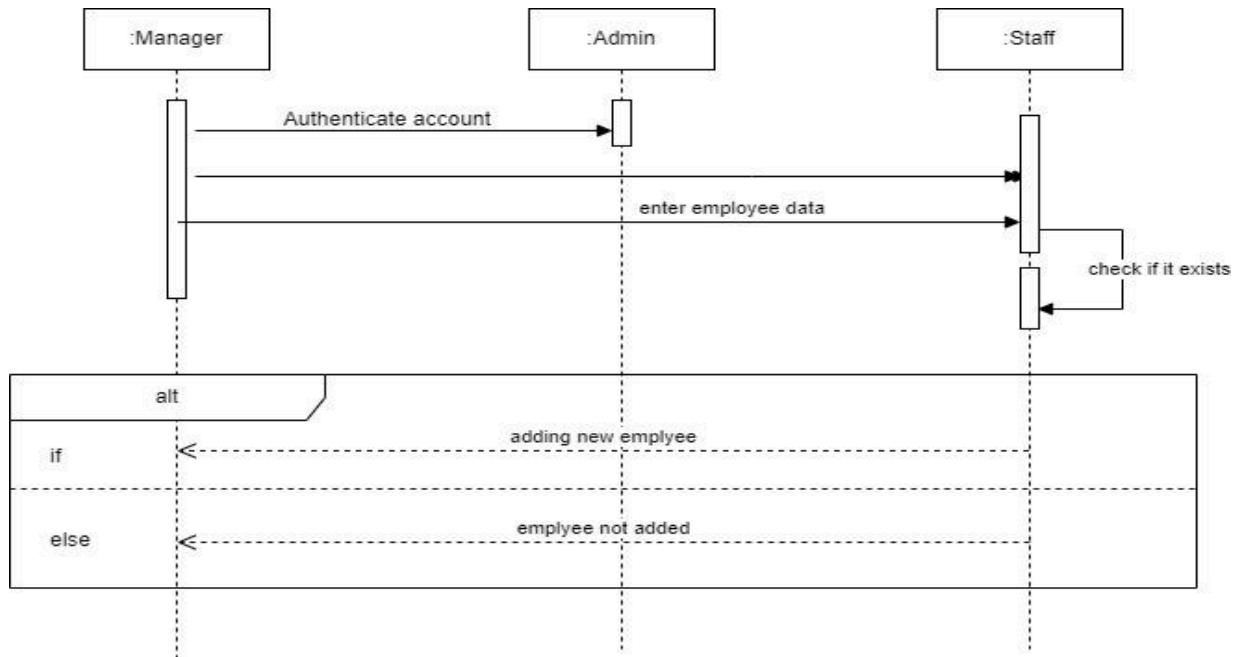


Bookstore System Documentation

Delete employee



Add employee



- Activity Diagrams for the requirements they represented.

An Activity Diagram is a behavioral diagram in UML that represents the flow of activities within a system. It is used to model the workflow or business processes and shows the sequence of activities, including parallel and conditional activities.

Activity: represents a task or action within the workflow, represented as a rounded rectangle.

Start (Initial) Node: marks the beginning of the workflow, represented as a filled circle.

End (Final) Node: marks the end of the workflow, represented as a filled circle with a border.

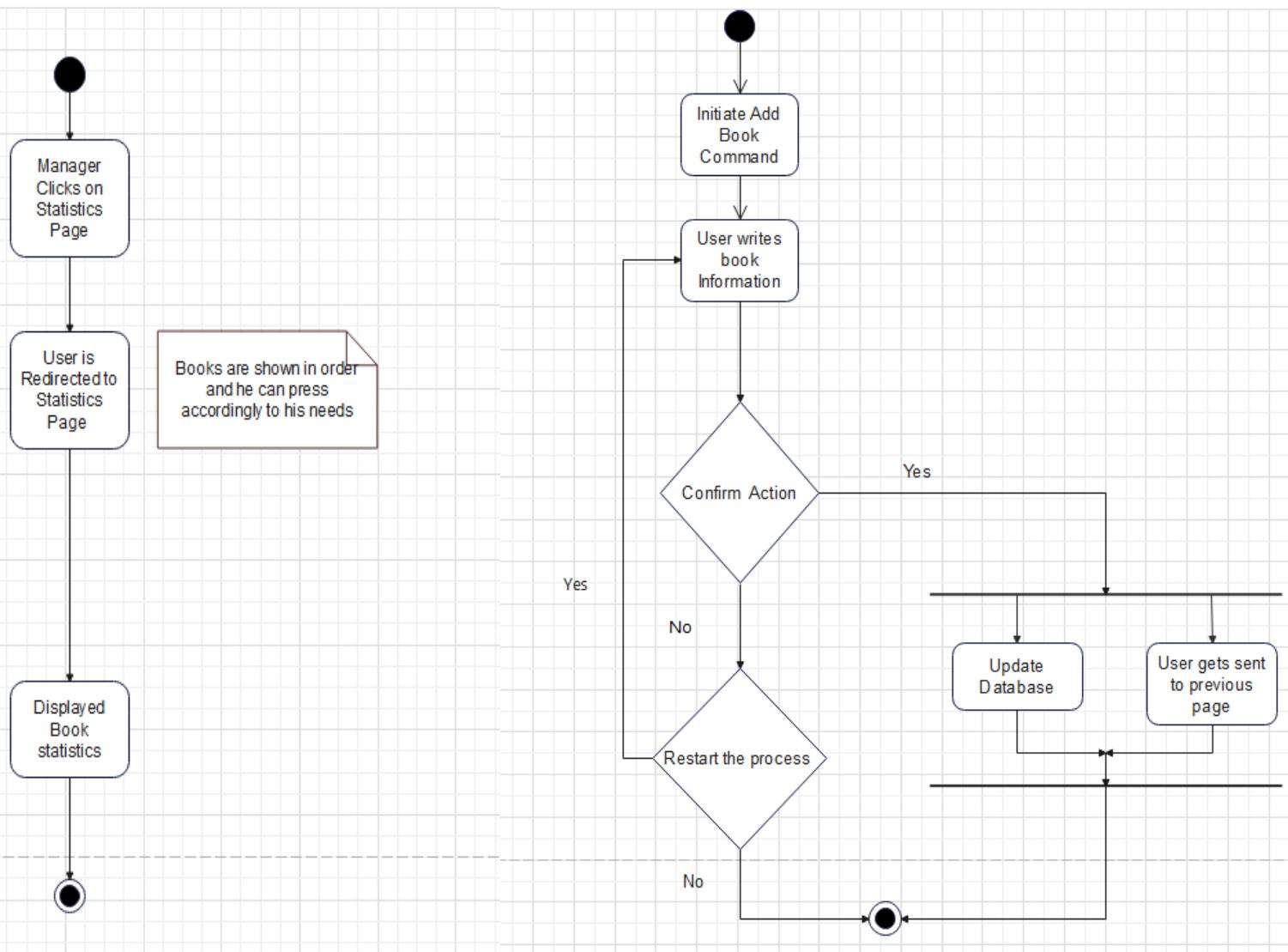
Decision Node: represents a branching point where a decision is made, represented as a diamond with outgoing arrows labeled with conditions.

Merge Node: combines multiple paths into a single path, represented as a diamond.

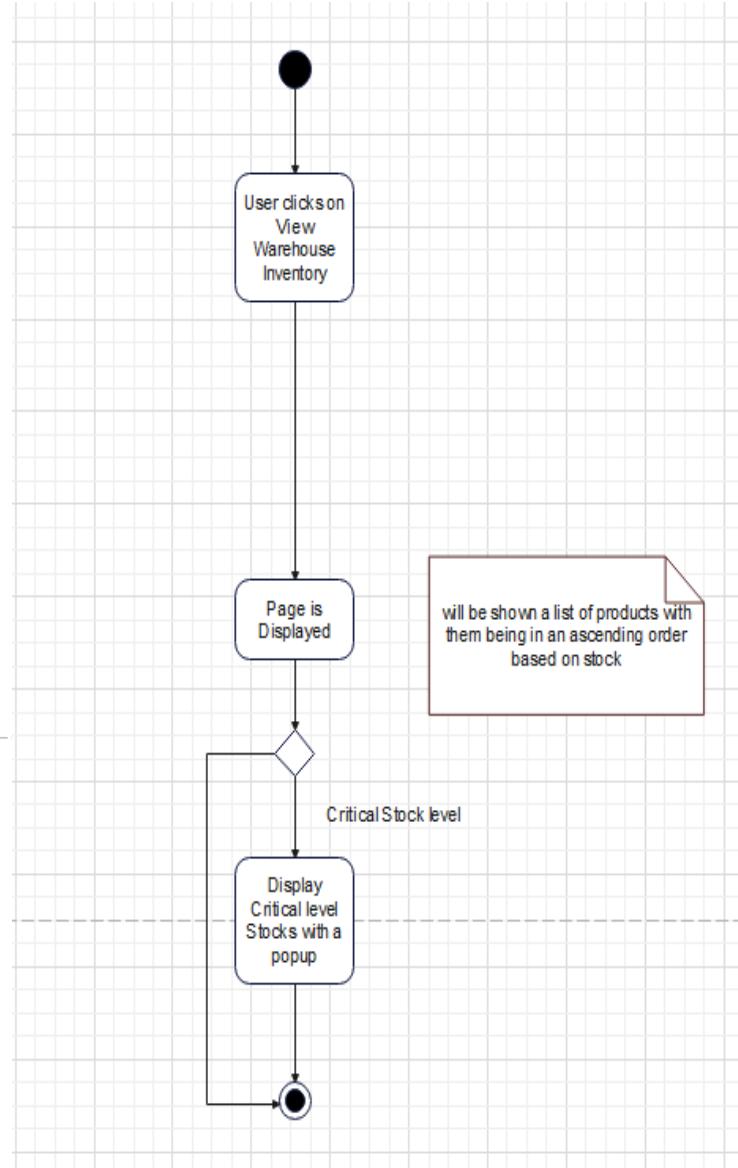
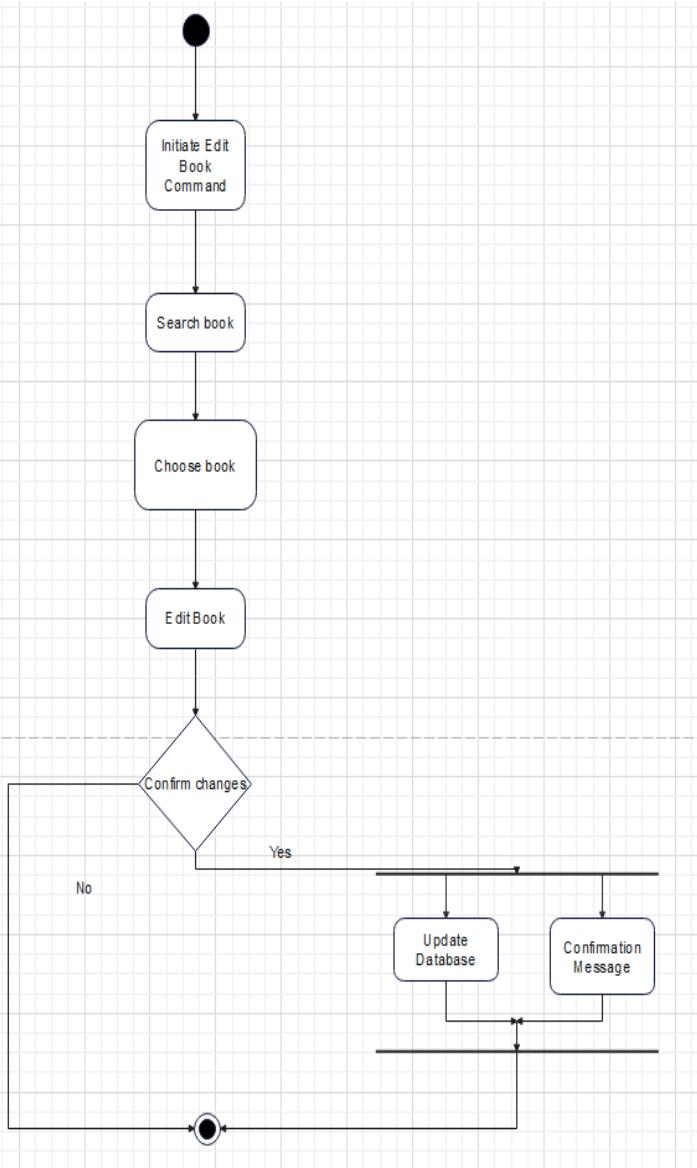
Fork Node: splits a single flow into multiple parallel flows, represented as a horizontal or vertical bar.

Join Node: joins multiple parallel flows into a single flow, represented as a horizontal or vertical bar.

Swimlanes: divide the diagram into lanes, representing different actors or departments.

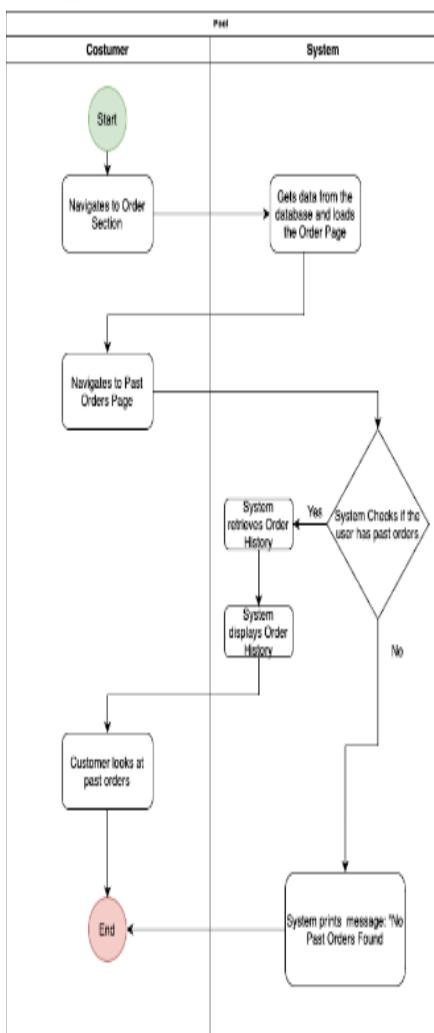


Bookstore System Documentation

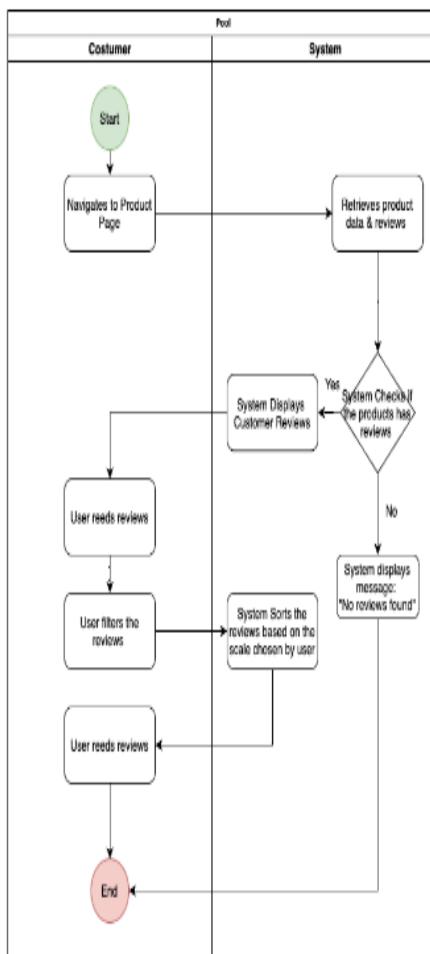


Bookstore System Documentation

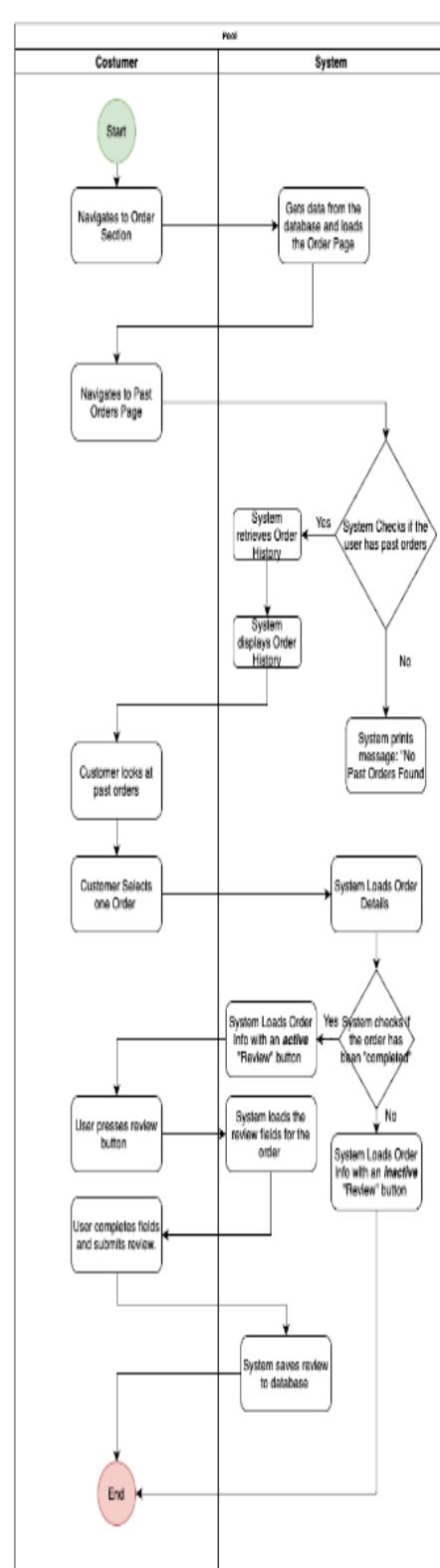
UC: View Past Orders



UC: View Customer Reviews

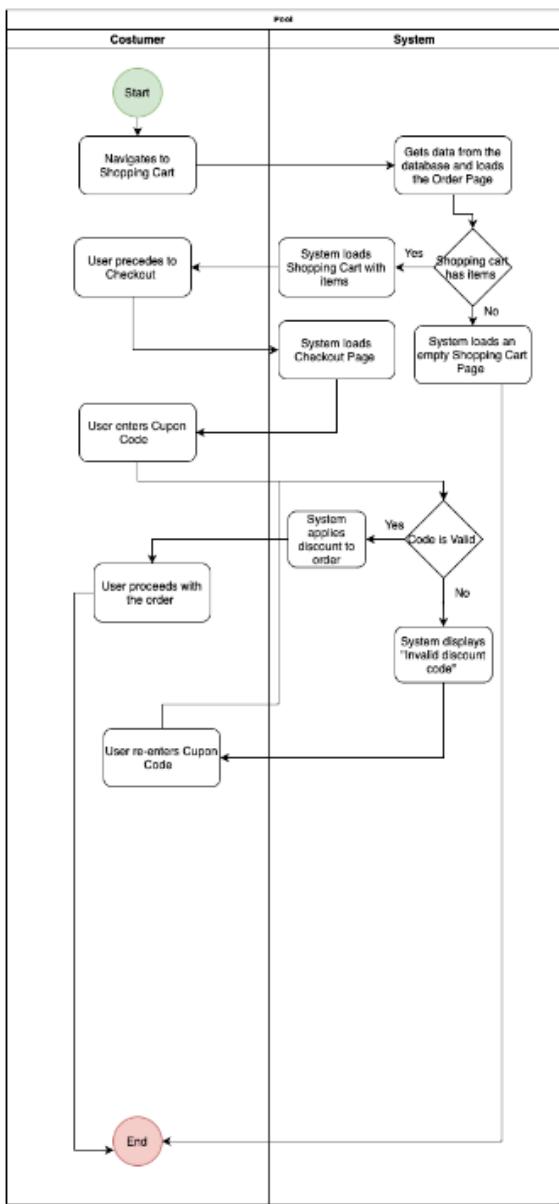


UC: Leave a review

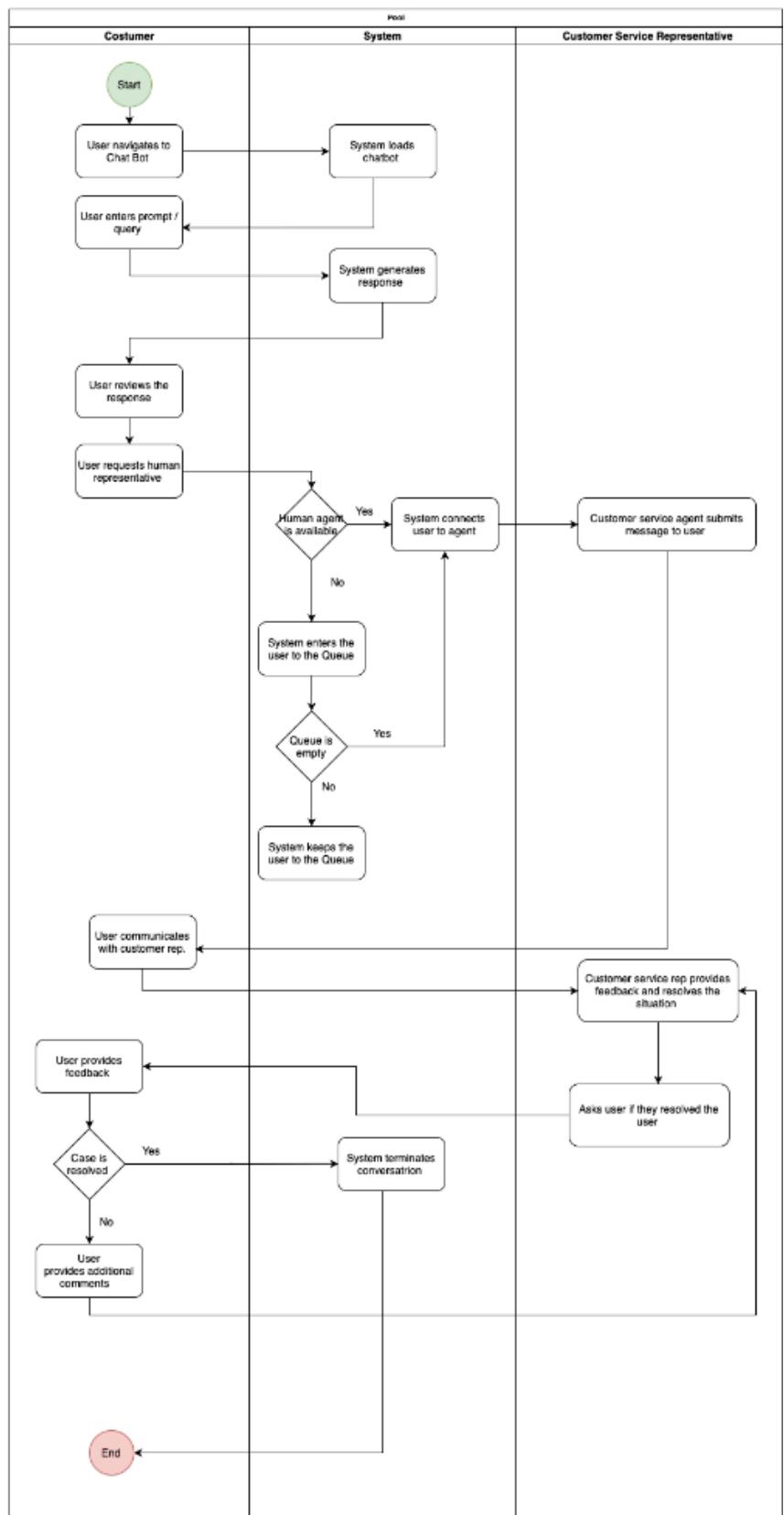


Bookstore System Documentation

UC: Apply Discount Code

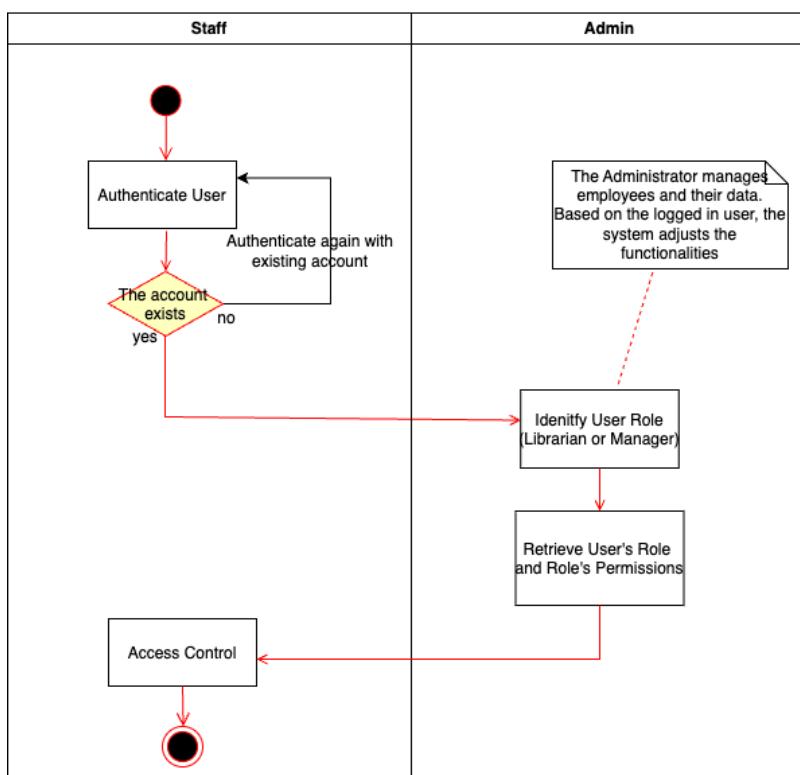


UC: Support ChatBot

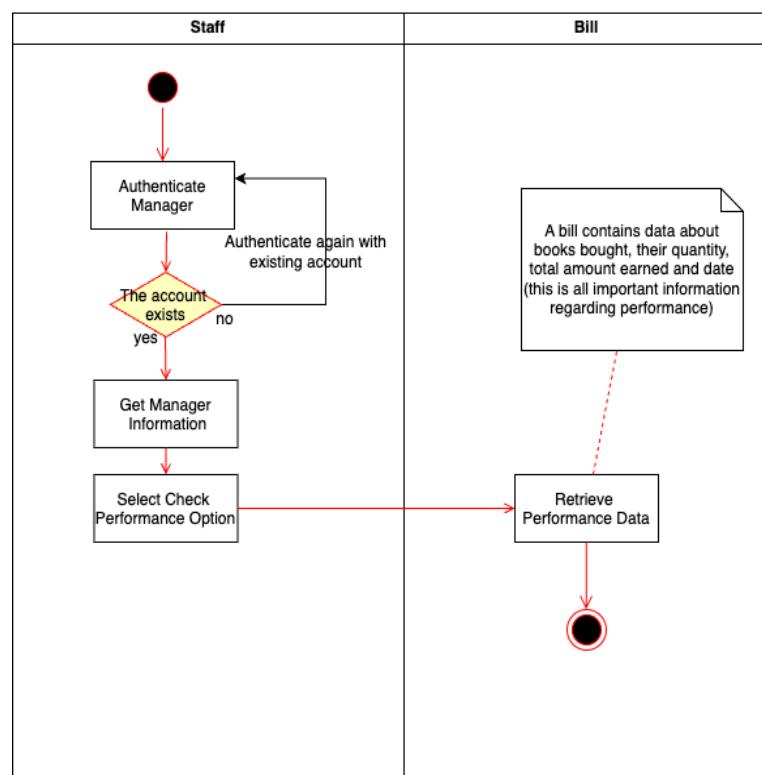


Bookstore System Documentation

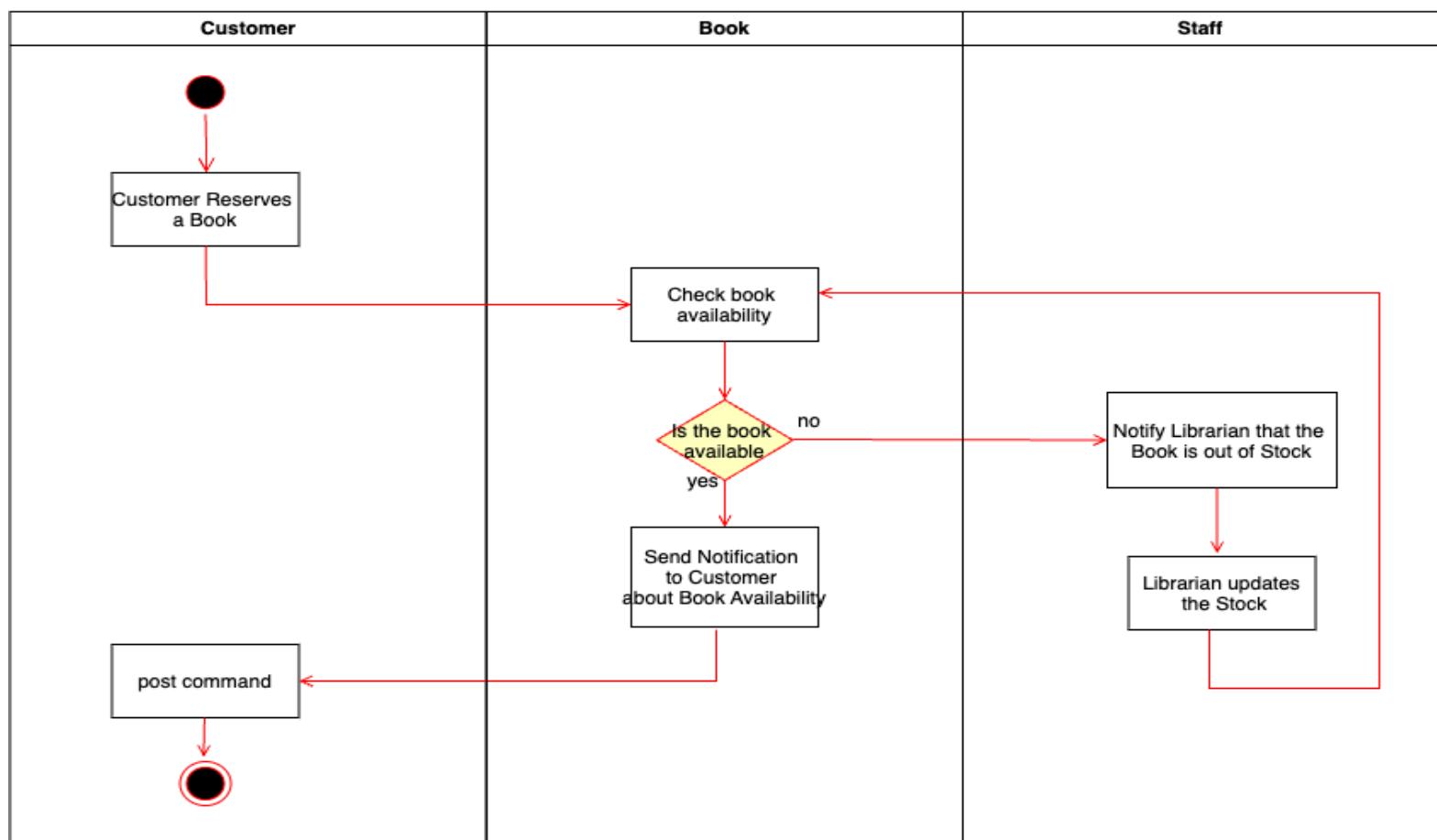
ROLE-BASED ACCESS CONTROL



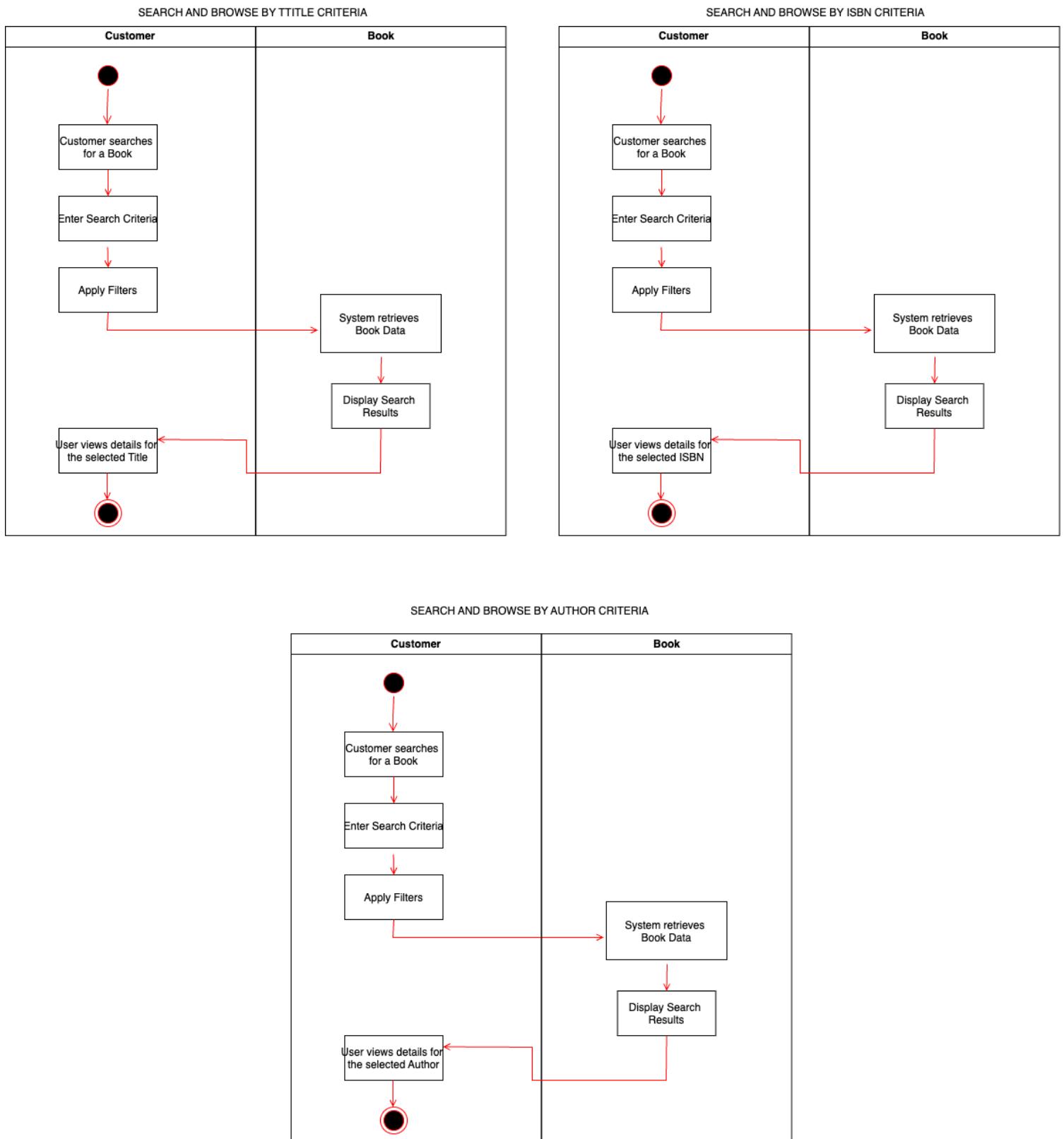
CHECK PERFORMANCE



RESERVATION NOTIFICATION SYSTEM

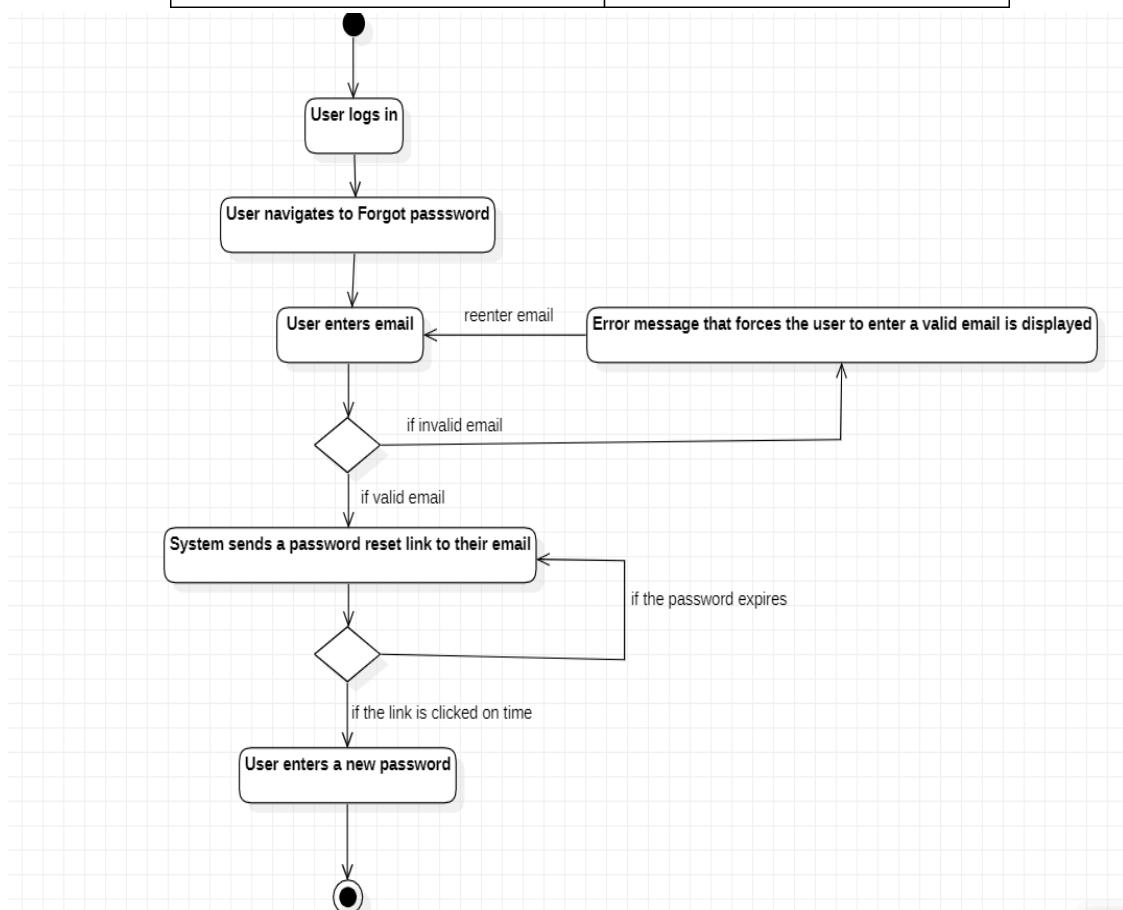
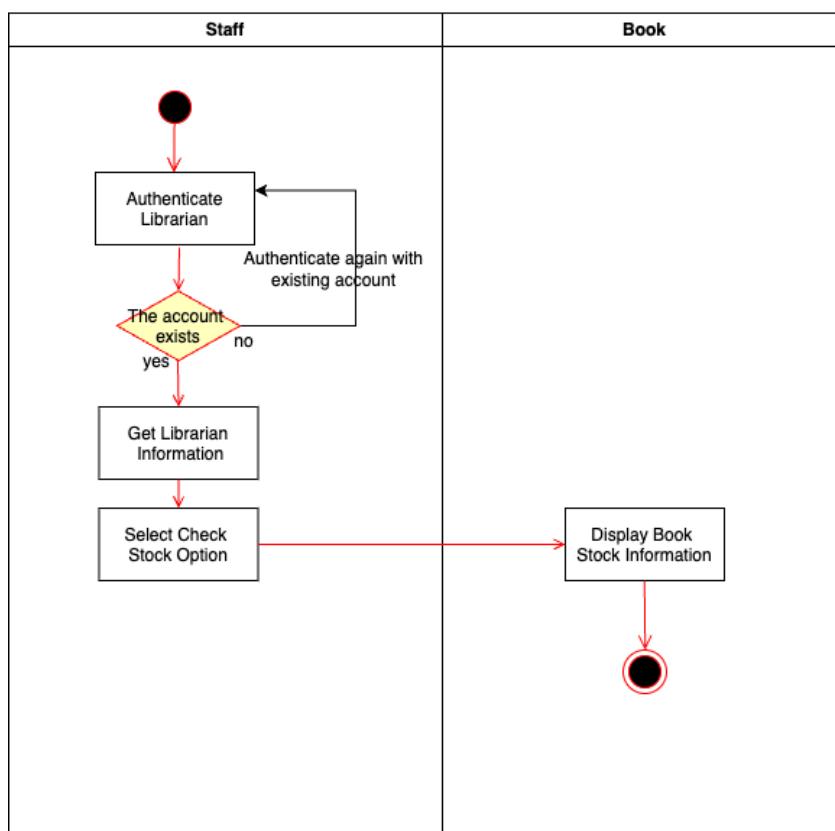


Bookstore System Documentation

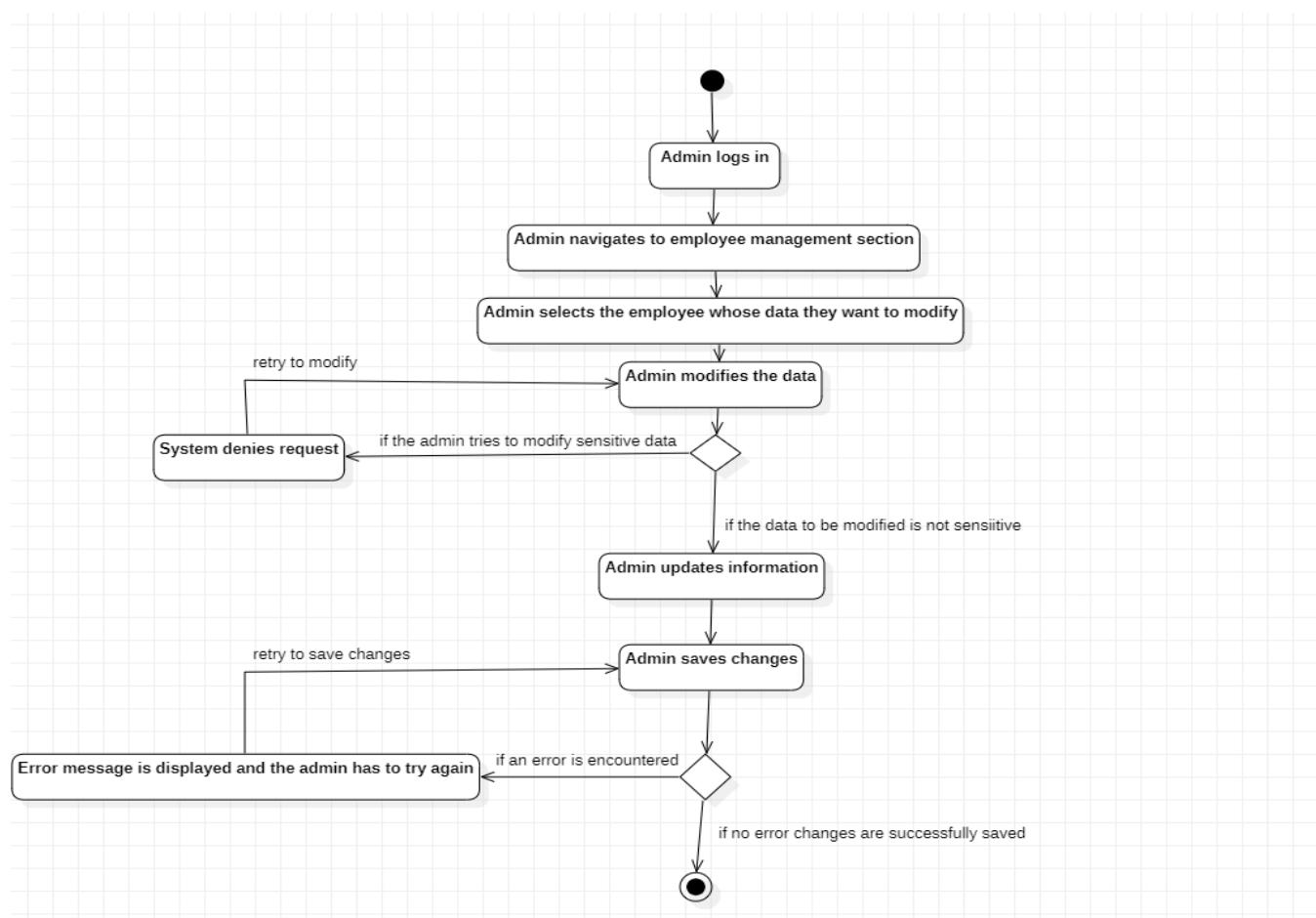
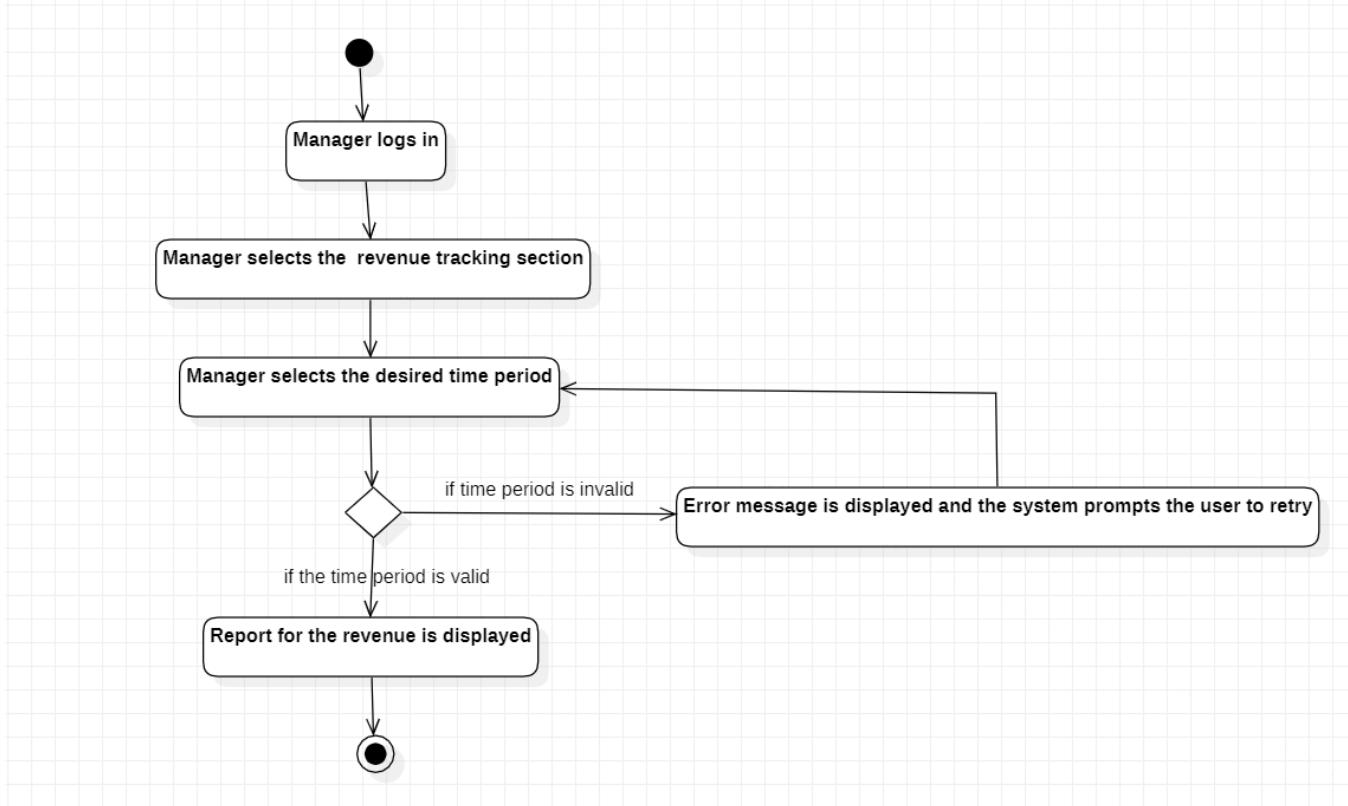


Bookstore System Documentation

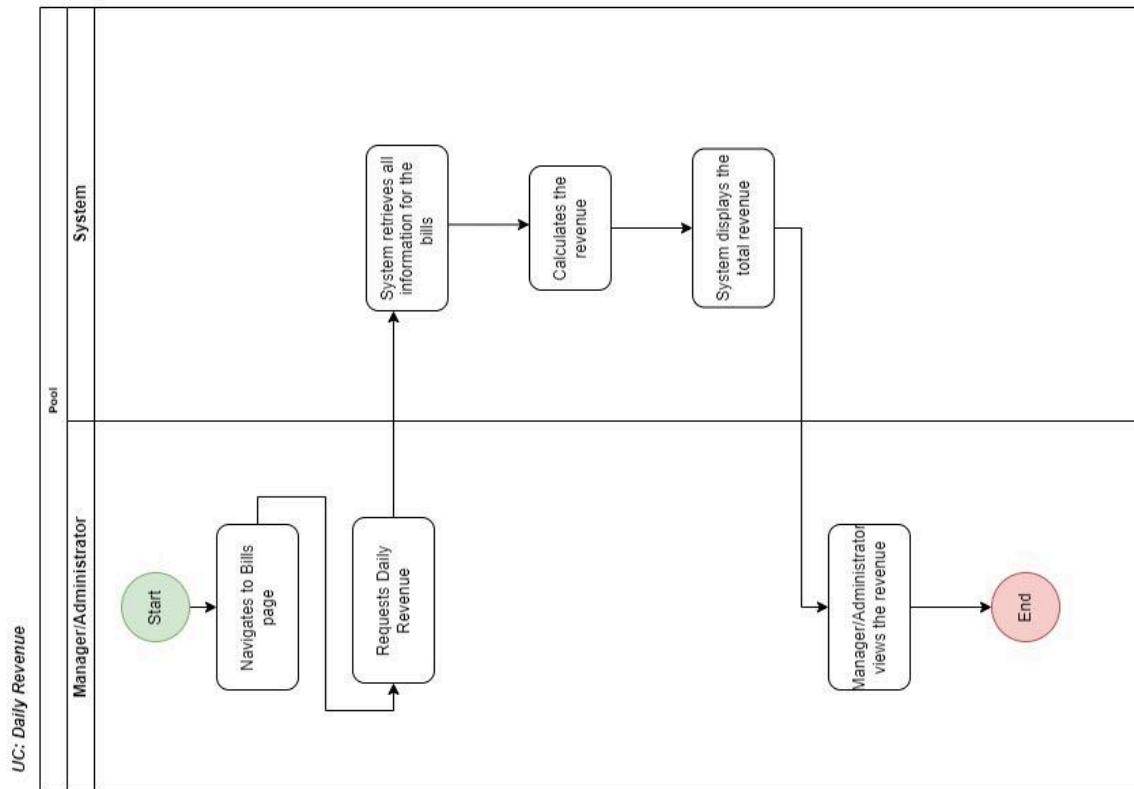
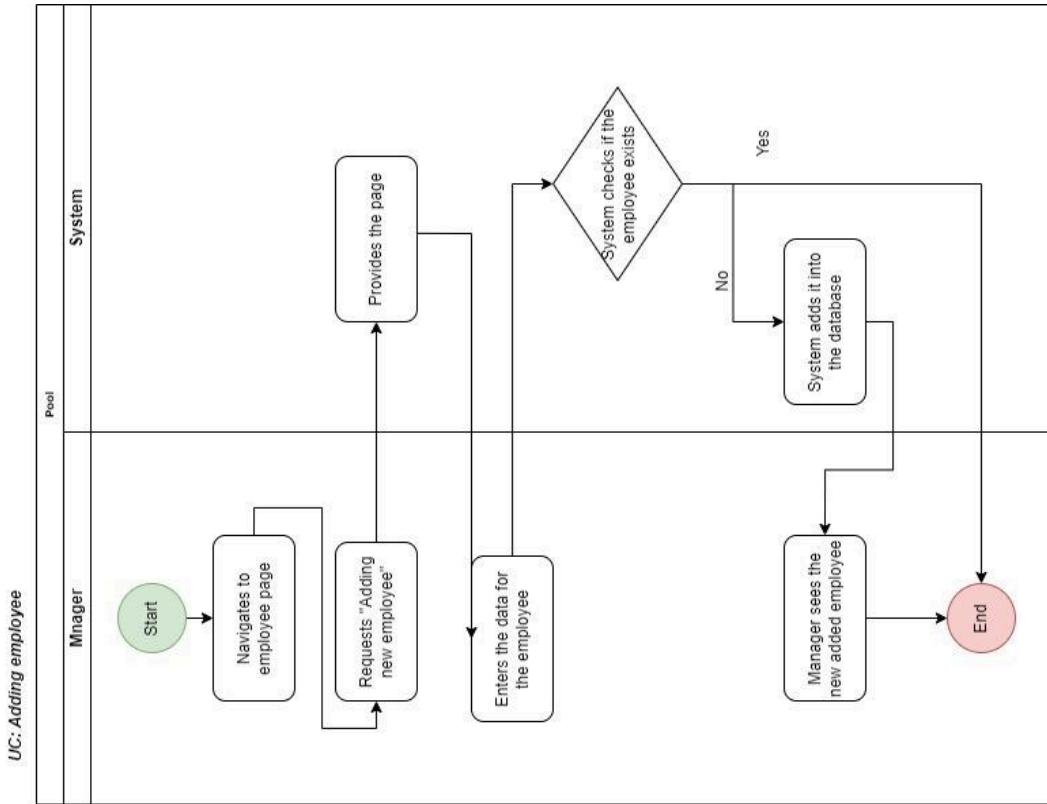
VIEW BOOK STOCK



Bookstore System Documentation



Bookstore System Documentation



Bookstore System Documentation

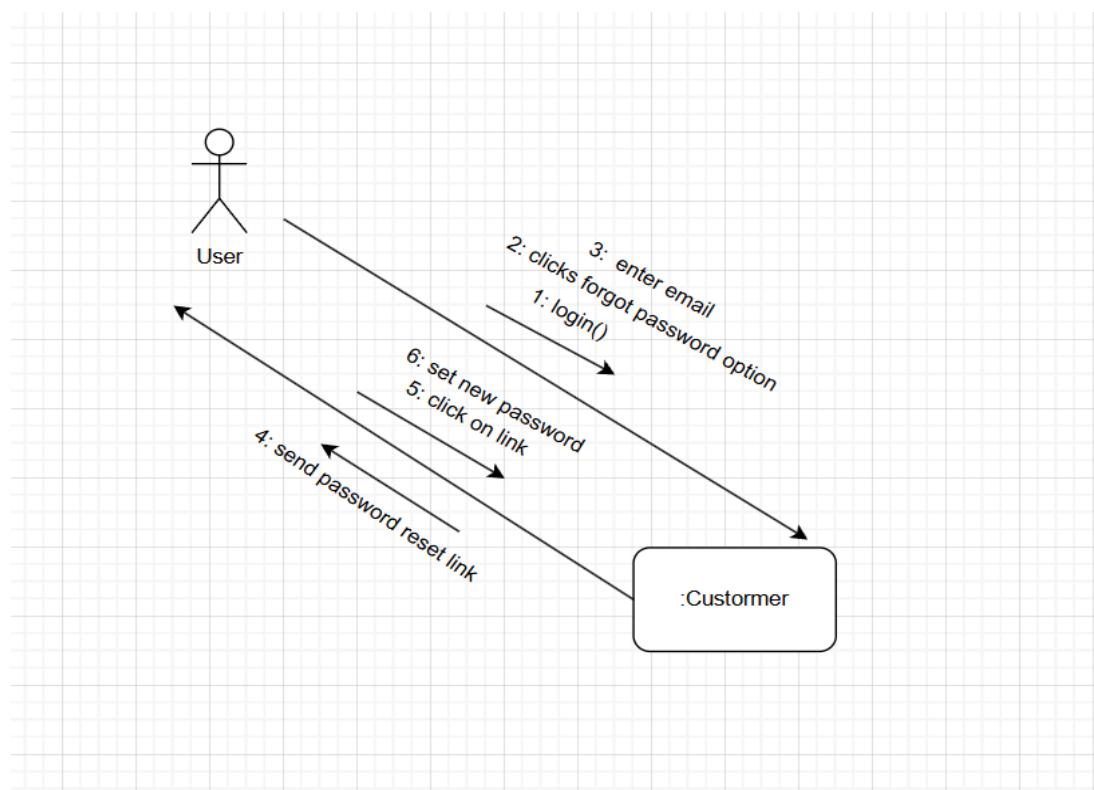
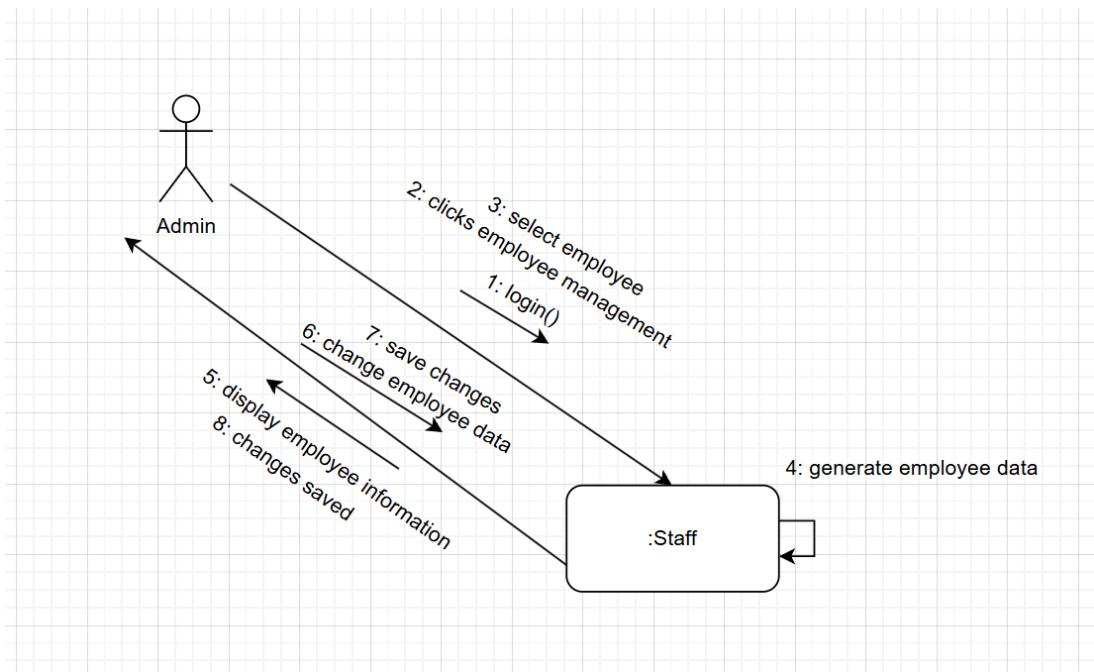
- Collaboration Diagrams to illustrate the collaborations between objects.

A Collaboration Diagram (or Communication Diagram) is a type of interaction diagram in UML that shows how objects interact in terms of messages. It emphasizes the structural organization of objects and their links rather than the time sequence.

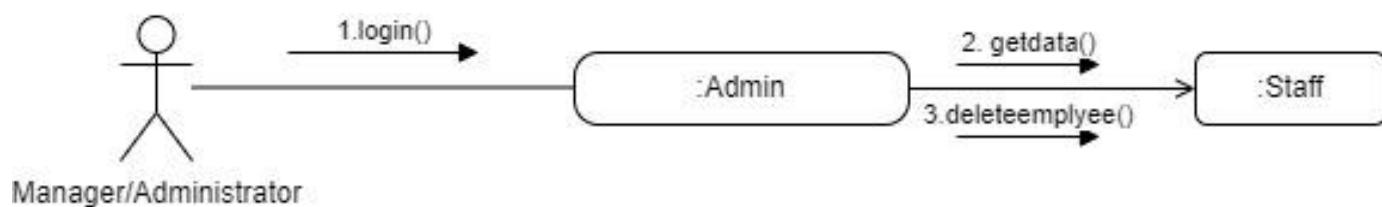
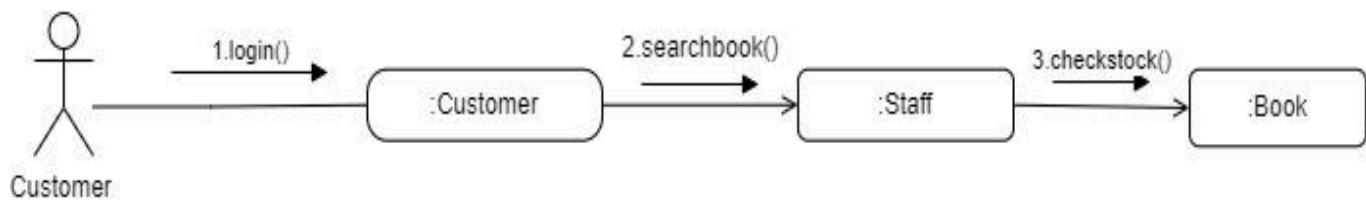
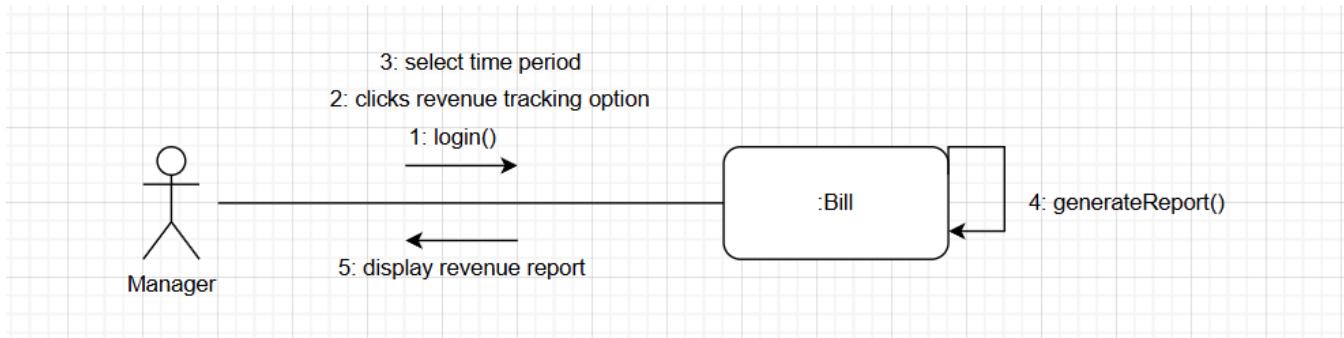
Objects: represent the entities that interact, represented as rectangles with underlined names.

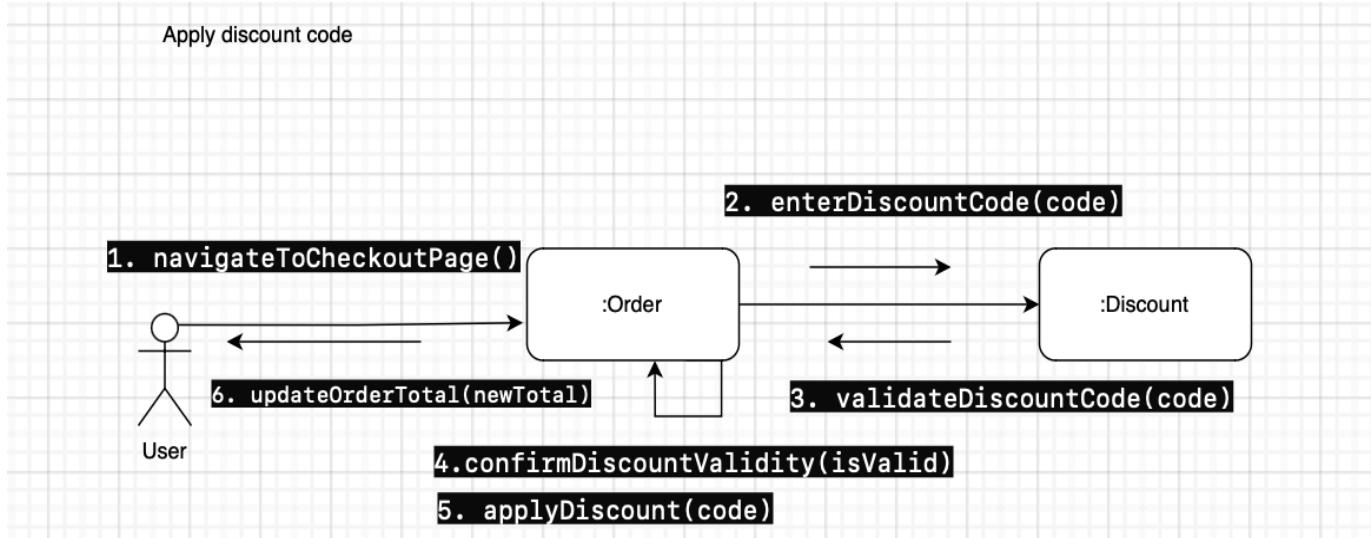
Links: represent connections between objects, represented as lines connecting objects.

Messages: represent communication between objects, represented as arrows with sequence numbers indicating the order of messages.

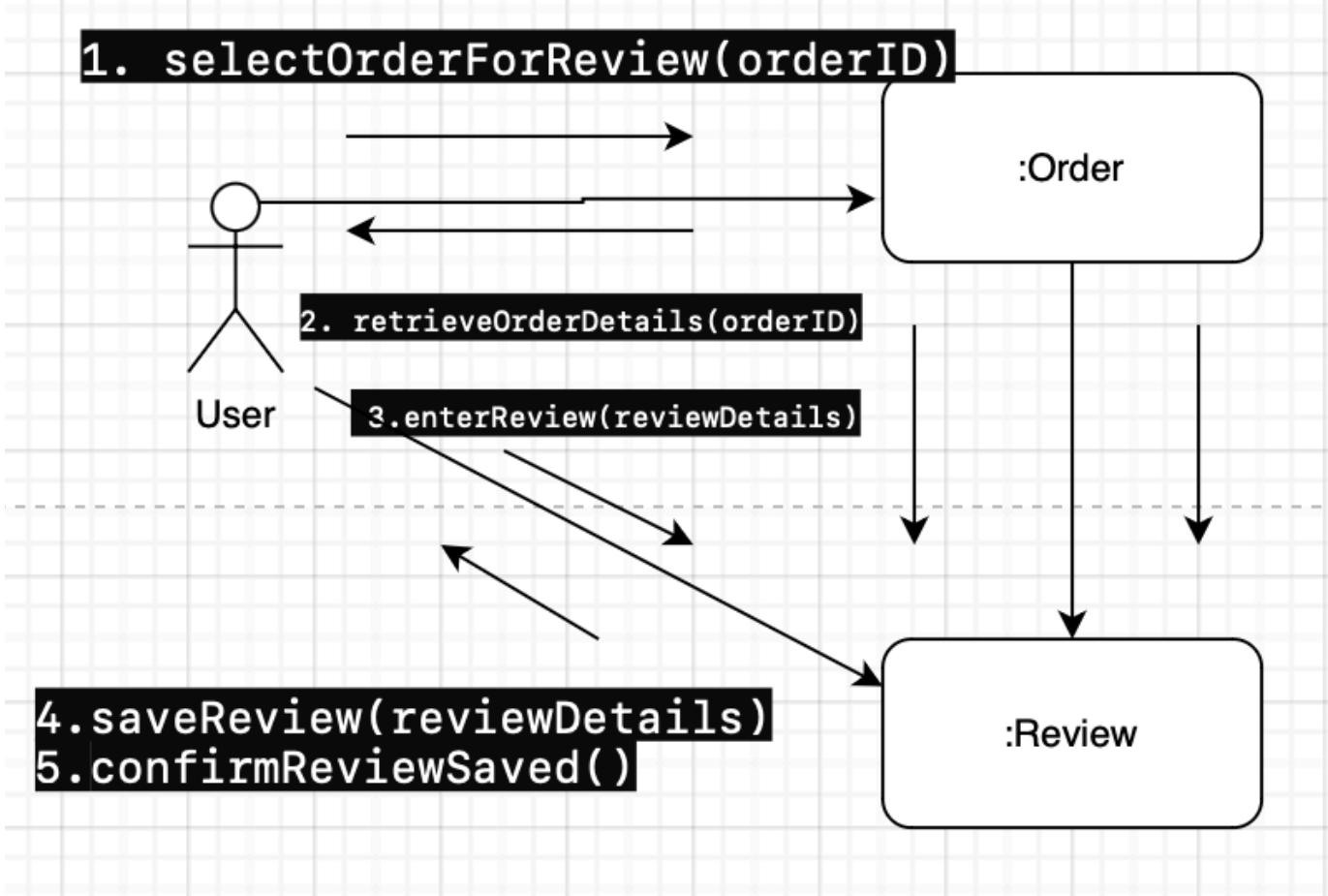


Bookstore System Documentation

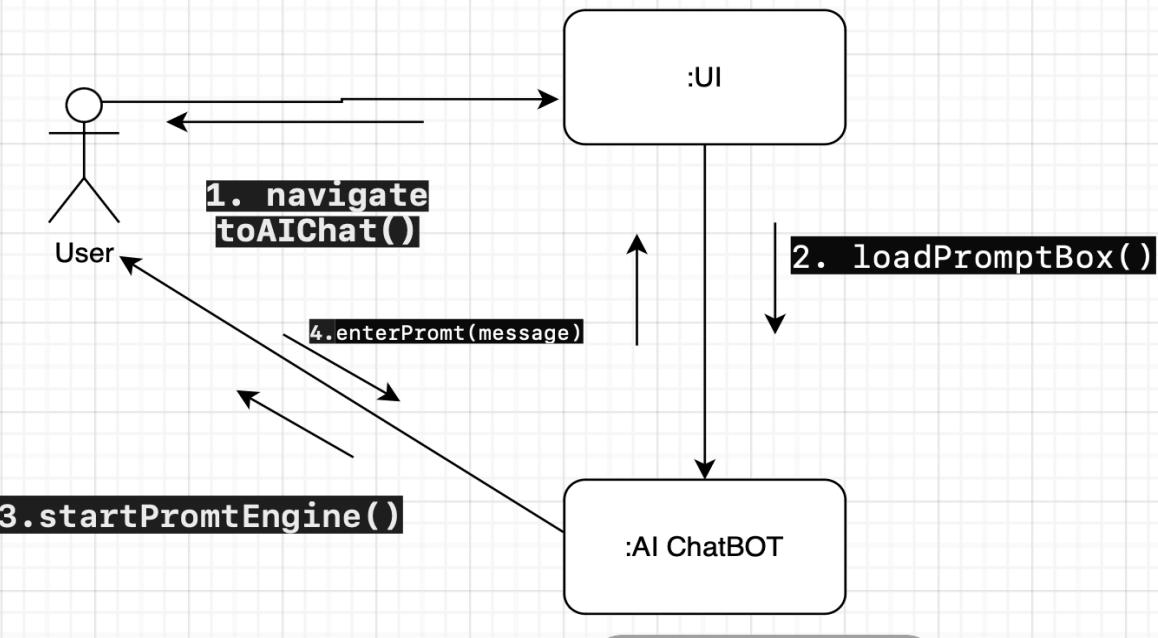




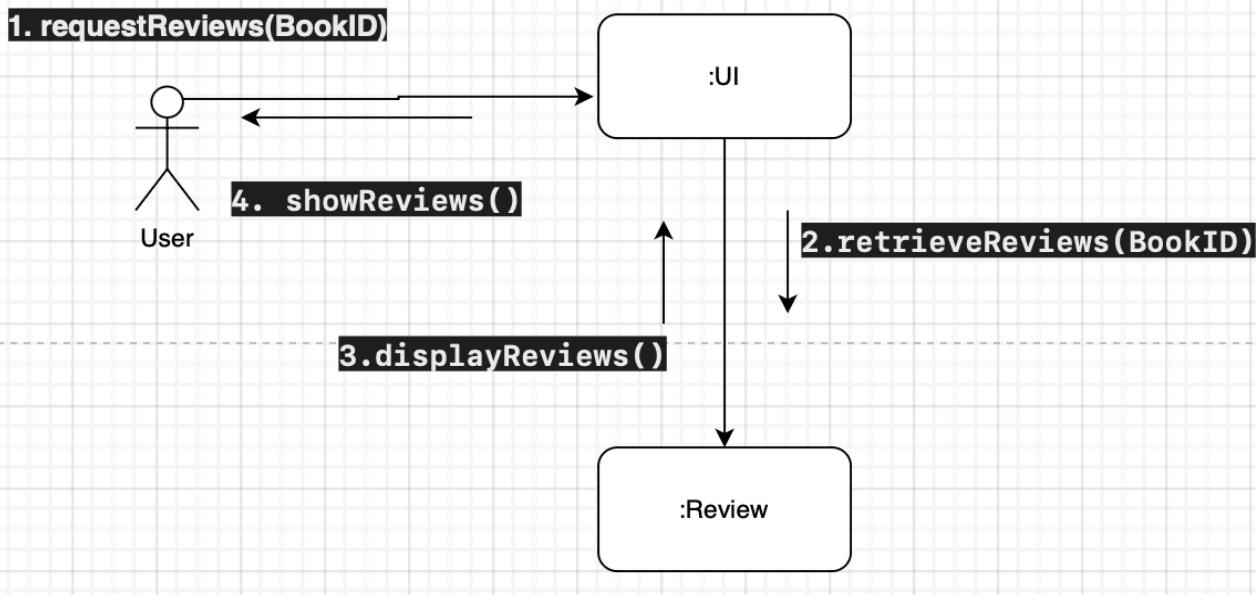
Add Review



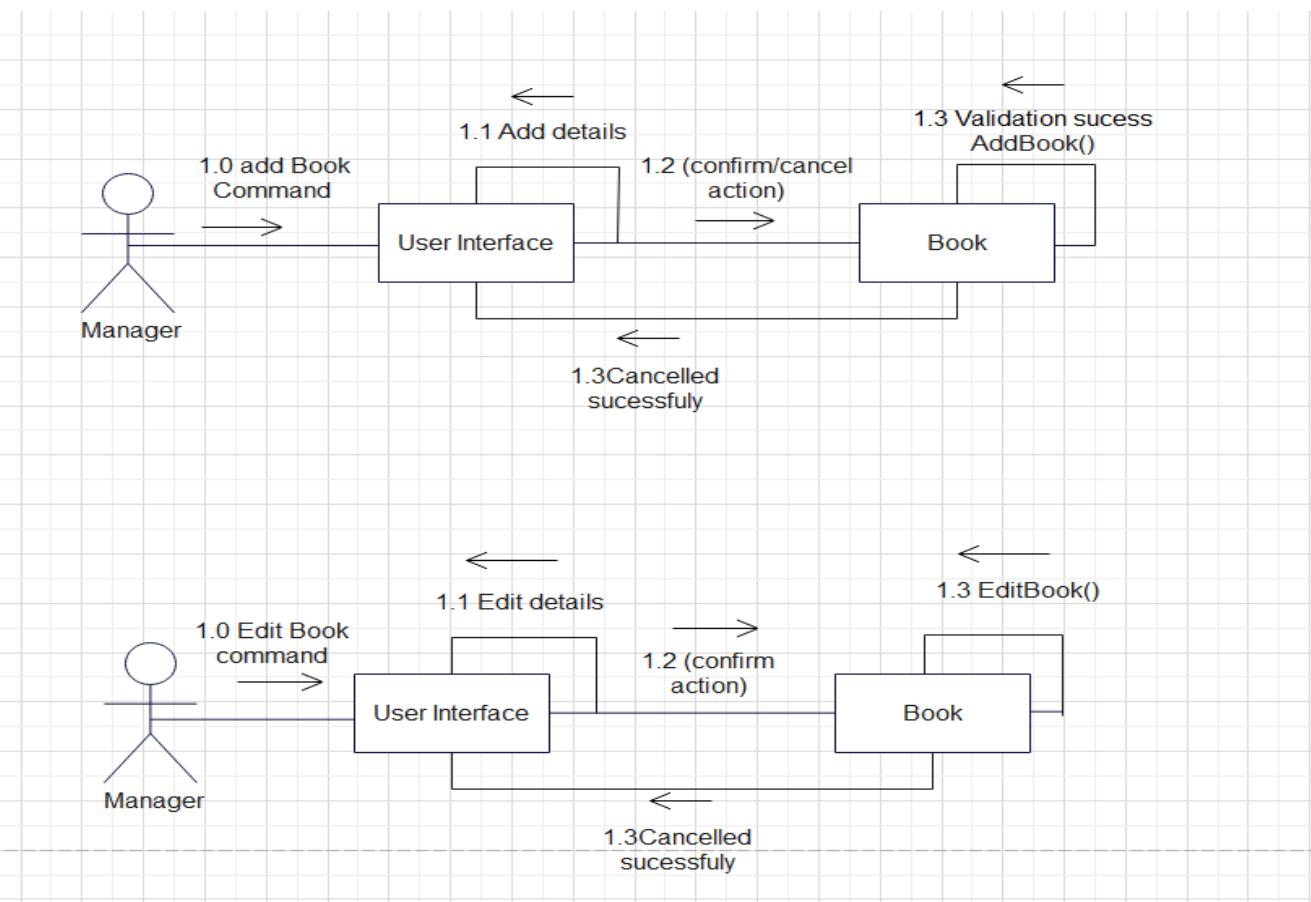
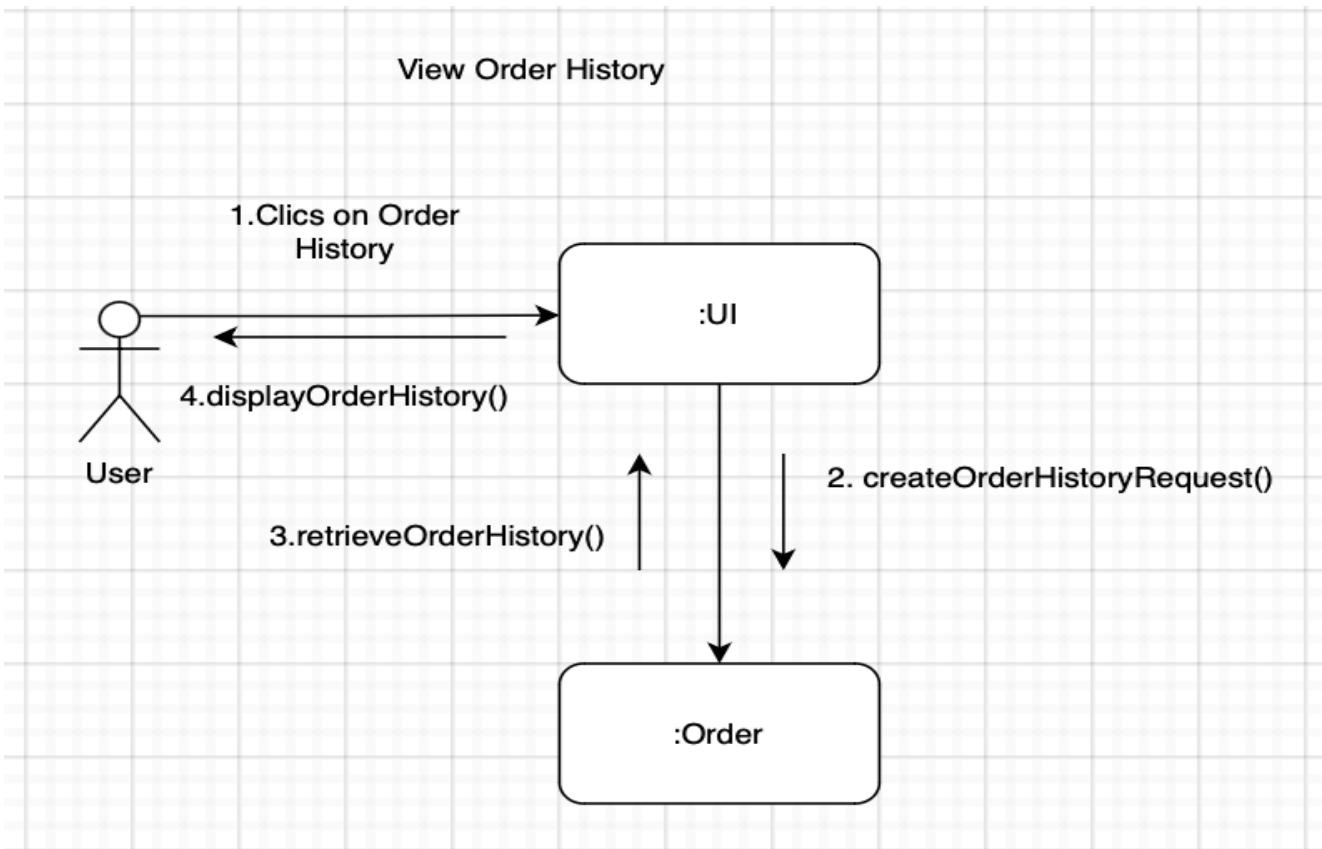
Contact Support Chatbot



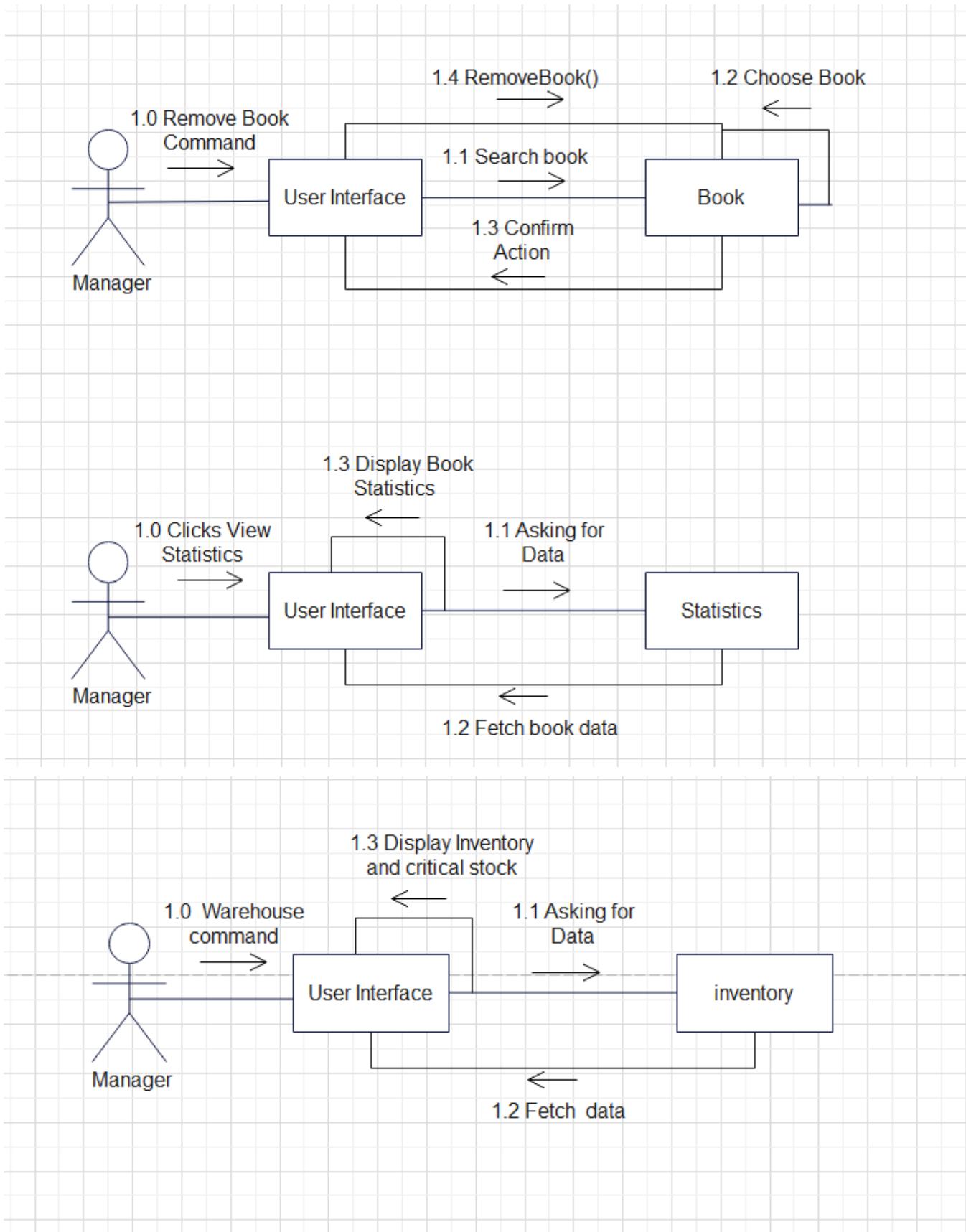
View Customer Reviews



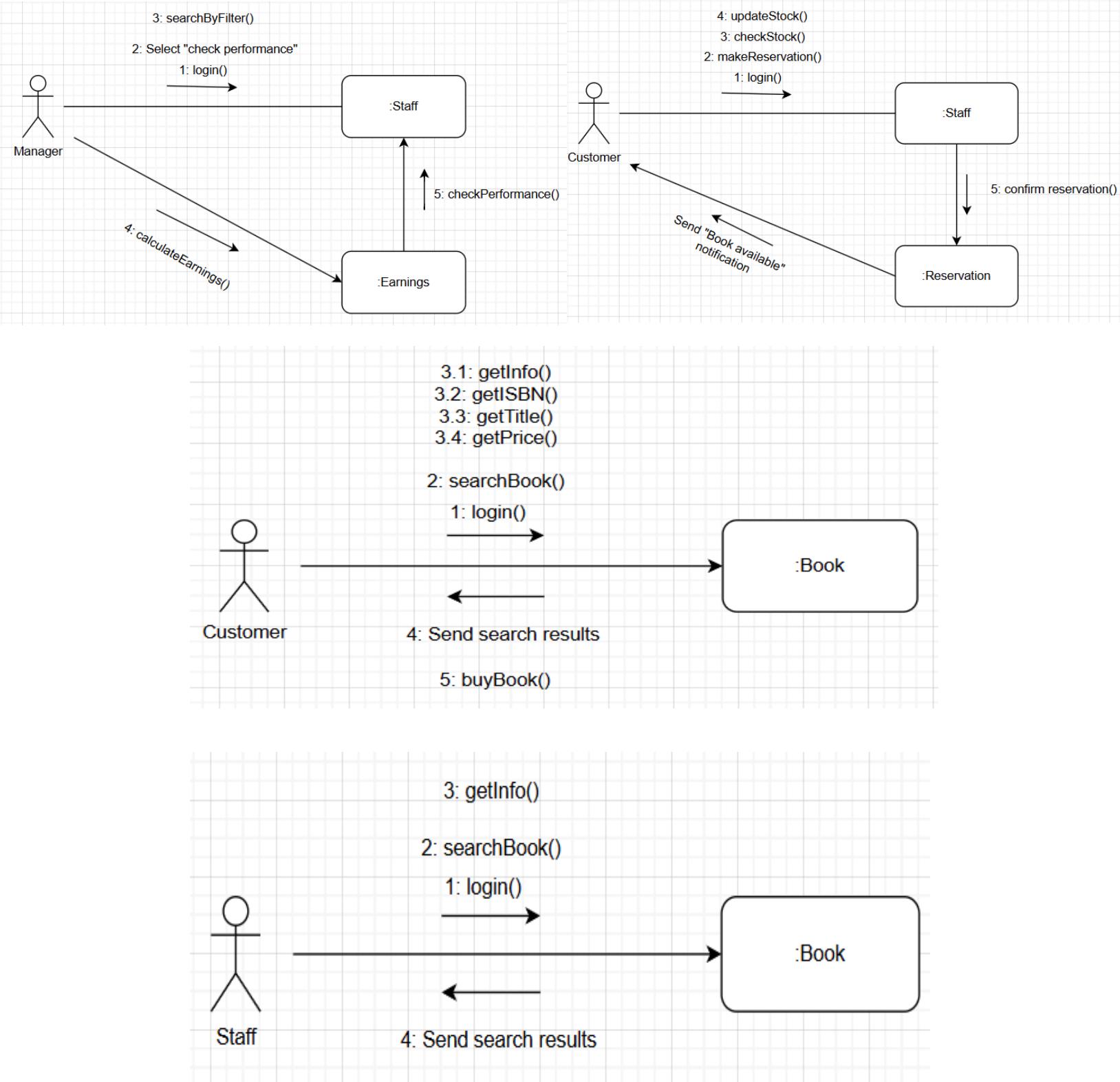
Bookstore System Documentation



Bookstore System Documentation



Bookstore System Documentation



Bookstore System Documentation

Interaction Overview Diagram: (BONUS)

An Interaction Overview Diagram is a type of diagram used in software and systems modeling, specifically within the UML. It combines elements from both activity diagrams and sequence diagrams to provide an overview of the control flow of interactions within a system.

Interaction Overview Diagrams are used to depict the control flow of different interactions, which could be sequences of messages or other forms of interaction between objects in a system. They help in understanding how different interactions are orchestrated together to achieve a larger workflow or process. These diagrams incorporate elements from both activity diagrams (such as control nodes and flows) and sequence diagrams (such as lifelines).

Activity Nodes: Represent actions or interactions.

Initial Node: The starting point of the diagram.

Final Node: The ending point of the diagram.

Decision Nodes: Points where a decision is made, leading to different branches of flow.

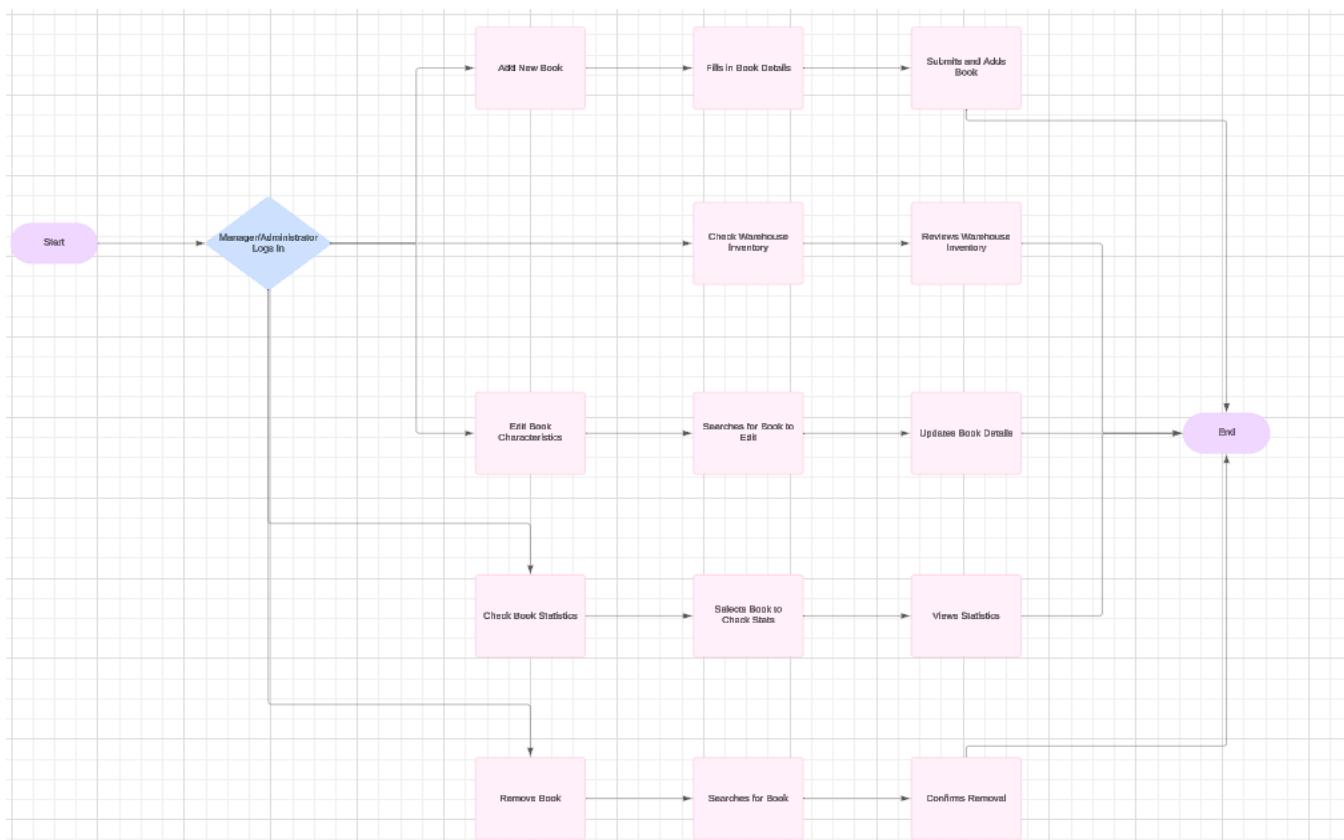
Merge Nodes: Points where different branches of flow merge back together.

Fork Nodes: Points where a single flow splits into multiple concurrent flows.

Join Nodes: Points where multiple concurrent flows join back into a single flow.

Interaction Occurrences: Represent specific sequences of interactions, typically shown using a reference to a sequence diagram or another interaction diagram.

Control Flows: Arrows that connect the nodes, showing the direction of control flow.



DESCRIPTION

Start: The process begins with the start node.

Decision Node: The flow requires a login from either a Manager or Administrator.

Add New Book Flow:

Add New Book: Initiate the process to add a new book.

Fill in Book Details: Enter details of the new book.

Submit and Add Book: Finalize and add the book to the inventory.

Check Warehouse Inventory:

Check Warehouse Inventory: Initiate the process to check current inventory.

Bookstore System Documentation

Review Warehouse Inventory: Review the details of the inventory.

Edit Book Characteristics:

Edit Book Characteristics: Start editing an existing book's details.

Search for Book to Edit: Find the book that needs editing.

Update Book Details: Make the necessary changes and update the book details.

Check Book Statistics:

Check Book Statistics: Start checking statistics for a book.

Select Book to Check Stats: Choose a specific book to view its statistics.

View Statistics: View the selected book's statistics.

Remove Book:

Remove Book: Initiate the process to remove a book.

Search for Book: Find the book that needs to be removed.

Confirm Removal: Confirm the removal of the book from the inventory.

End: The process ends here.

User interface

Home Page:

Header with logo, navigation menu, and login/register buttons.

Search bar with categories dropdown (allows users to specify the category or type of products they want to search within. Example - categories might include genres or authors).

Featured books section.

Footer with links to important pages (e.g., About Us, Contact Us).

Product Listing Page:

Grid or list view of books with cover image, titles, authors, and prices.

Filter and sort options (e.g., by genre, author, price).

Pagination for browsing through multiple pages of results.

Product Details Page:

Larger book cover image.

Detailed book information (title, author, description, price, etc.).

Add to Cart button.

Turn on notifications for reservation of a book (if the stock is zero).

Reviews and ratings section

Cart Page:

List of items in the cart with thumbnails, titles, quantities, and prices.

Subtotal, shipping options, and total price.

Proceed to checkout and continue shopping buttons.

Checkout Page:

Customer information form (if not logged in).

Shipping and billing address fields.

Payment options.

Order summary with itemized list and total cost.

Place Order button.

Customer dashboard (when logged in):

Display the customer's account information, including recent orders, order history, and saved addresses/payment methods (if applicable).

Search for books and browse categories, similar to the functionality available on the main website.

View and manage items in the cart, proceed to checkout, and complete purchases.

Track the status of current orders and view details such as shipping information and estimated delivery dates.

Integrate a chatbot feature where customers can ask for help, get answers to frequently asked questions, or request assistance from customer support representatives (the chatbot can provide automated responses to common queries related to order status, product information, shipping, and returns. Customers can also escalate to a live customer support agent if their query cannot be resolved by the chatbot).

Bookstore System Documentation

Provide a way for customers to leave reviews or suggestions for improving the bookstore's products or services.
Update personal information, change password and logout option.

Manager Dashboard:

Overview of sales, orders, and inventory.

Check Librarian's Performance option.

Manage Customers: View customer information and order history.

View Statistics with daily, monthly and total filters (in addition, here are included: sales trends or top-selling products displayed with charts and graphs)

Manage Products: Add, edit, or delete books.

Manage Orders: View and process customer orders.

Admin Dashboard:

Overview section displaying key metrics such as total sales, number of orders, and current inventory status.

Navigation menu with options to access different sections:

Manage Employees: Add, remove, or edit data about employees (librarians, managers).

Settings section for administrative tasks such as managing roles and permissions.

Librarian Dashboard:

Book Details: Access detailed information about each book in the inventory including title, author, genre, ISBN, price, and current stock level.

View Stock: Display a list of all available books along with their current stock levels.

Change Stock: Ability to update the stock level of individual books (e.g., add new stock, remove stock due to damage).

Print Bills: Generate and print bills for each transaction made in the physical store, including details of items sold, quantities, prices, and total amount.

Search functionality to quickly find specific books or orders within the system.

Bookstore System Documentation

Testing Strategy Documentation

Throughout the development of our Bookstore Management System, ensuring its quality and reliability through rigorous testing is crucial. In this section, we have outlined our approach to testing, including test plans, test cases, testing methodologies, and how we will document test results and findings.

Test Plans:

We have prepared detailed test plans that outline our testing approach, objectives, and scope. These plans include the testing environment description, testing schedule, responsibilities of team members, identification of risks, and sign-off criteria for each testing phase.

Test Cases:

Unit Test Cases: These focus on testing individual units or components to ensure they function correctly.

Integration Test Cases: These verify the interactions and integration between different modules of our system.

System Test Cases: These comprehensively test the entire system to ensure it meets our requirements.

User Acceptance Test (UAT) Cases: These involve end-users to validate that our system meets their expectations before deployment.

Testing Methodologies:

Unit Testing: Testing each unit in isolation to verify its functionality.

Integration Testing: Testing the integration between different components to ensure they work together seamlessly.

System Testing: Validating the system as a whole to ensure it meets all requirements.

User Acceptance Testing (UAT): Involving end-users to validate the system against their requirements before deployment.

Test Results and Findings:

Our plan is to maintain test execution logs to record test case status and any defects encountered, document defects found during testing, including detailed descriptions, steps to reproduce, severity, and priority and provide summarized test summary reports to provide an overview of testing activities.

By following this comprehensive testing strategy, we will ensure that our Bookstore Management System is thoroughly tested and meets the required quality standards before deployment. Regular monitoring of test results will help identify and address any issues early in the development lifecycle, minimizing the risk of defects in the final product.