**Software Requirements Specification**

**for**

Scamazon

**Version 1.0 approved**

**Prepared by Keifer Hardy, Alexander Kellough, Brennan Sheckart, Steven Skeen, and Joseph Sorrell**

**Intro to Software Engineering Group 4**

**5 February 2024**

# **Introduction**

## **Purpose**

The purpose of this document is to provide a detailed description for the Scamazon e-Commerce Bookstore Version 1.0 project. It will explain the purpose and features of the overall system, how it interacts with external stimuli, what constraints it will operate under, and its interface as a whole. This document is intended for use by the stakeholders and developers during the lifetime of the project. It will outline the requirements needed for the project in CS 4214, including having a buyer, seller, and admin user that interacts with the bookstore.

## **Document Conventions**

This document is written following the SRS outline provided in the Intro-To-SE\_SRS\_Template. Each section is written to fulfill the suggested contents written in the template. No highlighting or special typographical standards are used. Each requirement statement in the document can have its own priority specified.

## **Intended Audience and Reading**

This document is intended for use by the developers and graders, both as a description of the scope and intent of the described software, as well as a general roadmap for the development process. The following introductory sections will give a generalized view of the scope of the software, the second section will generally describe intended functionality and interactions, and the rest of the document will serve as a detailed description and reference manual for the development process. Particularly, Section 4 will serve as documentation of requirements that may not easily be inferred from the description of features alone.

## **Product Scope**

Scamazon is a web application that is designed to enable buying and selling of books online. It is intended to allow users to sell or buy books depending on their needs. Buyers should be able to search, compare, purchase, and return their books. Sellers should be able to add, sell, and receive payments for their books. It will allow a secure login and logout of users and will support an admin user that can manage posts and users. The system will be available on the internet and is intended to be easy to use for all users.

## **References**

## Python Django Documentation

* Title: Django Documentation
* Author: Django Software Foundation
* Version Number: Version 5.0
* Date: 2/5/23
* Source/Location: [Django Documentation](https://docs.djangoproject.com/en/5.0/)

## HTML/CSS Documentation

* Title: MDN Web Docs (HTML/CSS)
* Author: Mozilla
* Version Number: HTML5 and CSS3
* Date: 2/5/23
* Source/Location: [HTML Doc](https://developer.mozilla.org/en-US/docs/Web/HTML) and [CSS Doc](https://developer.mozilla.org/en-US/docs/Web/CSS)

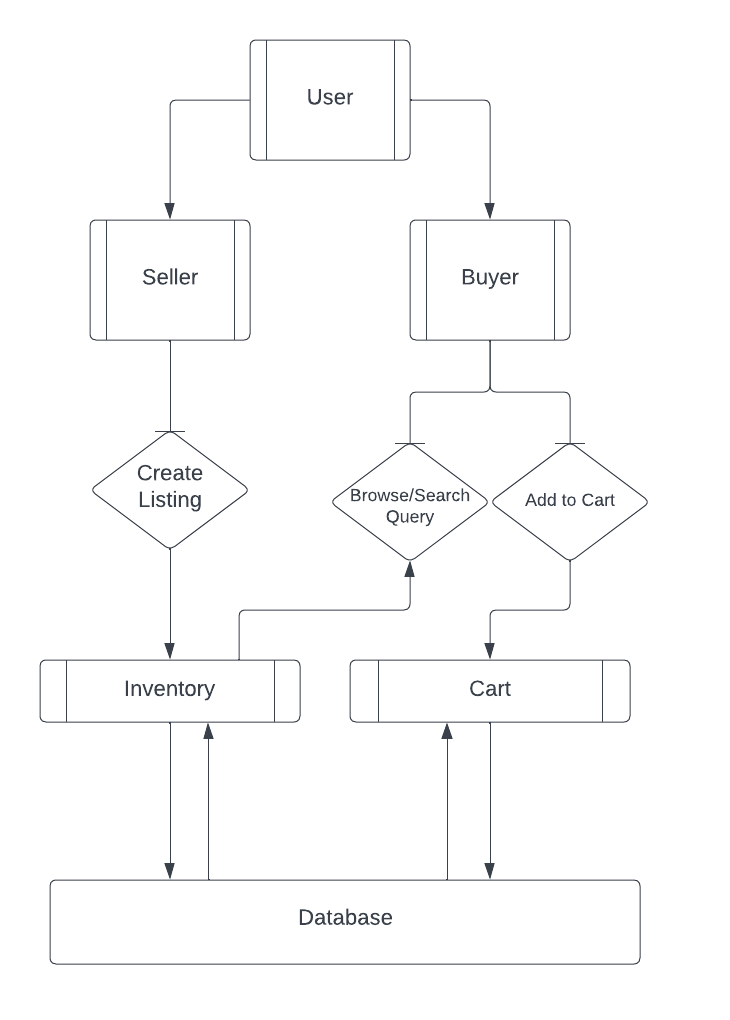
## SQL Documentation

* Title: MySQL Documentation
* Author: MySQL Global Development Group
* Version Number: MySQL 8.0
* Date: 2/5/23
* Source/Location: [MySQL Documentation](https://dev.mysql.com/doc/)

# **Overall Description**

## **Product Perspective**

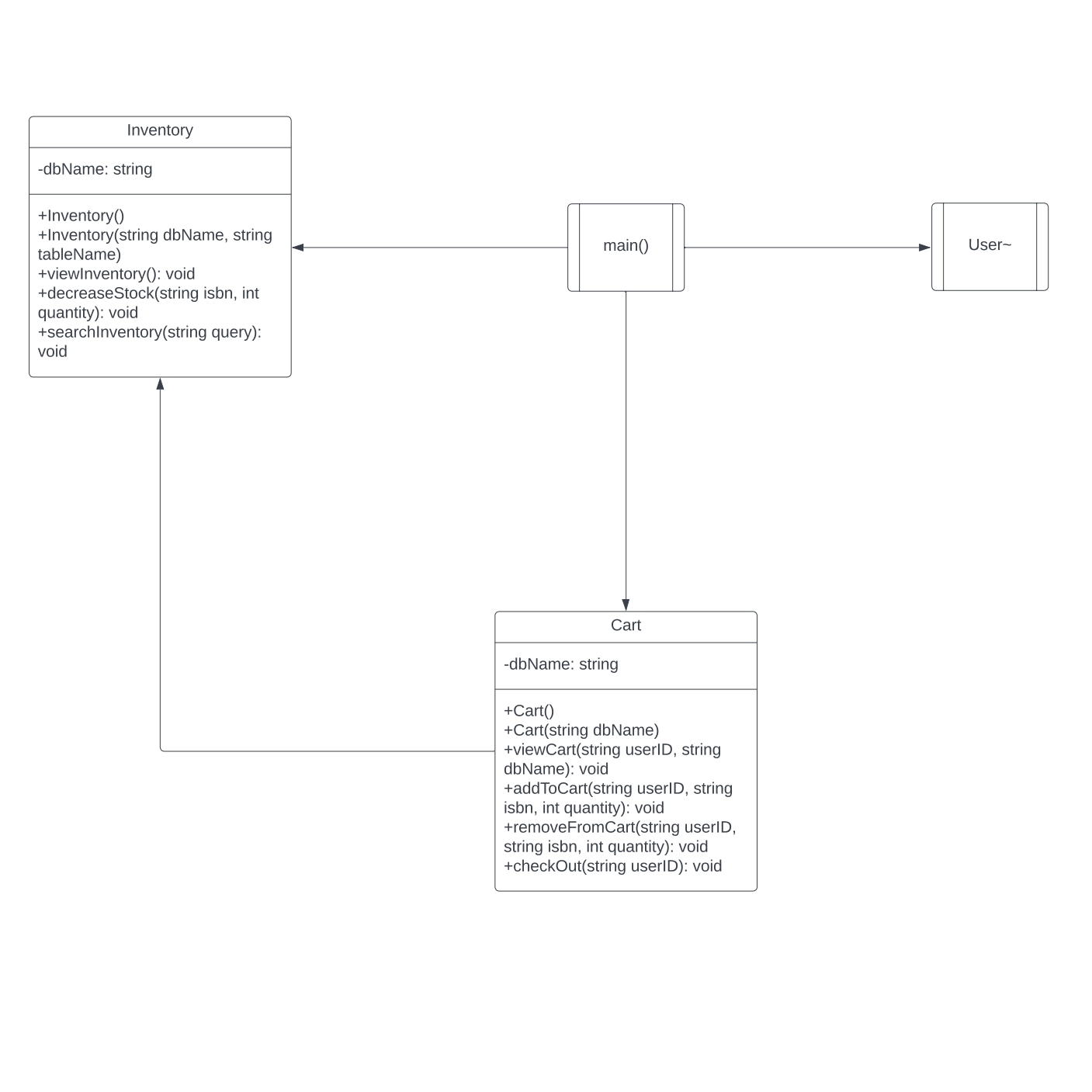
Scamazon is a new, self-contained, stand-alone e-commerce platform intended to be an alternative competitor to sites like Amazon and Ebay. The system as a whole requires sellers to be able to list and sell books to users that buy them. It should support a secure login and logout as well as supporting an admin that can block users, listings, and other actions. For relations between system components see the diagram below.

****

## **Product Functions**

The overall functions of the entire system are:

* Users can register for a new account
* Users can log in and out of their account
* Buyers can view and search for listings
* Buyers can add books to cart and checkout
* Buyers can return a book they previously purchased
* Sellers can post listings for books and manage those listings
* Sellers can receive mock payment for their purchased books
* Admin can manage user accounts
* Admins can audit and modify user listings

****

## **User Classes and Characteristics**

**2.3.1****Seller** A user logged in as a seller can use the website to create a listing where they can set the price, condition, and edition of the book. These web listings can be accessed and modified by the original lister and site admins. Once a seller creates a listing it becomes available for buyers to view and purchase.

**2.3.2 Buyer**

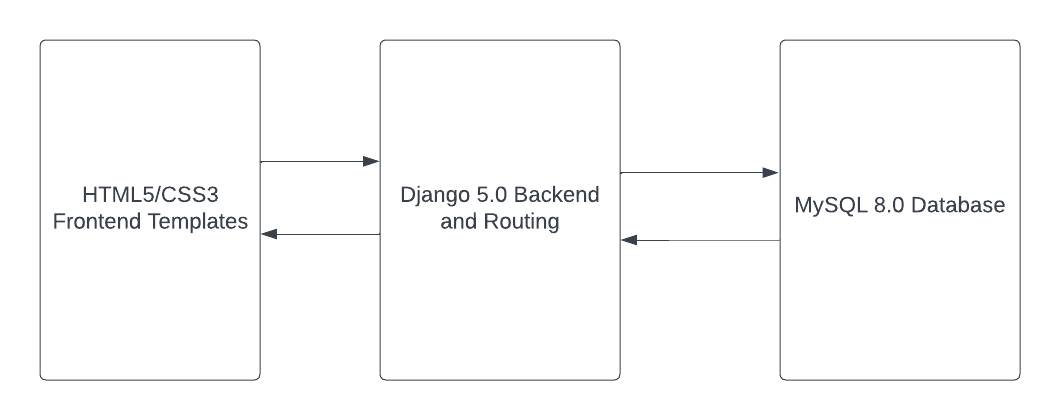
Buyers will be able to access the web app in order to search, view, and purchase existing listings. A user logged in as a buyer will be able to add items to their cart, view their cart, and checkout.   
  
**2.3.3 Admin**

A user logged in as an admin has access to higher privileges than other site users. Admins can view, modify, and (if necessary) remove existing listings and audit site changes. Admins can also block and approve user accounts. Because this class is only used by developers there is less frontend emphasis placed on satisfying this user.



## **Operating Environment**

Scamazon uses a HTML5/CSS3 Interface to interact with a private MySQL 8.0 database through Python Django 5.0 API calls.

**

## **Design and Implementation Constraints**

Given the indeterminate size of the site’s user base, Scamazon will need to be designed to handle significant user traffic without sacrificing performance or user security. As the platform’s backend is exclusively programmed in Python, performance and security concerns are thereby limited to the capabilities of the aforementioned language. In addition, Scamazon’s main server must be hosted on hardware capable of keeping up with the previous constraints. The development of the platform as a whole will be limited by the short development time that the development team has been alloted.

# **System Features**

## Sign-Up Screen

### 3.1.1 Description and Priority

The sign-up screen allows new users to create accounts within the system. This feature is of High priority as it is fundamental for user engagement and system utilization.

### 3.1.2 Stimulus/Response Sequences

* Stimulus:
  + User navigates to the sign-up screen.
  + User fills out the required fields (e.g., username, email, password).
  + User submits the sign-up form.
* Response:
  + System validates the user input.
  + If the input is valid, the system creates a new user account.
  + If there are errors, the system displays appropriate error messages.

### 3.1.3 Functional Requirements

REQ-1: User Registration

* The system shall provide fields for the user to input their username, email address, and password.
* The system shall verify that the email address provided by the user is unique.
* The system shall hash and securely store the user's password.

REQ-2: User Type Selection

* The user should be able to select their user type: either buyer or seller.
* This user type should be stored in the database and should redirect to the appropriate user screen depending on user type.

REQ-3: Validation and Error Handling

* The system shall validate the format of the email address provided by the user.
* If the email address is invalid, the system shall display an error message prompting the user to enter a valid email address.
* If any required fields are left empty, the system shall display appropriate error messages

REQ-4: User Feedback

* Upon successful registration, the system shall display a confirmation message to the user.
* In case of any errors during the sign-up process, the system shall provide error messages to guide the user in correcting their inputs.

## Login Screen

### 3.2.1 Description and Priority

The login screen allows registered users to authenticate themselves and access the system. This feature is of High priority as it is crucial for user access and system security.

### 3.2.2 Stimulus/Response Sequences

* Stimulus:
  + User navigates to the login screen.
  + User inputs their credentials (e.g., username/email and password).
  + User submits the login form.
* Response:
  + System validates the user input.
  + If the credentials are valid, the system grants access to the user's account.
  + If the credentials are invalid, the system displays appropriate error messages.

### 3.2.3 Functional Requirements

REQ-1: User Authentication

* The system shall provide fields for the user to input their username/email and password.
* The system shall verify the provided credentials against the stored user data.
* The system shall authenticate the user if the provided credentials match those stored in the system database.

REQ-2: Validation and Error Handling

* The system shall verify that both the username/email and password fields are not empty before allowing submission.
* If the username/email or password field is empty, the system shall display appropriate error messages indicating the missing information.
* If the entered username/email does not exist in the system database, the system shall display an error message indicating that the username/email is not recognized.
* If the entered password does not match the password associated with the provided username/email, the system shall display an error message indicating incorrect credentials.

REQ-3: User Feedback

* Upon successful login, the system shall redirect the user to the appropriate dashboard or landing page.
* In case of any errors during the login process, the system shall provide error messages to guide the user in correcting their inputs.

## Logout

### 3.3.1 Description and Priority

The logout feature allows authenticated users to terminate their current session and log out from the system. This feature is of Medium priority as it contributes to user convenience and security but may not be as critical as core functionalities.

### 3.3.2 Stimulus/Response Sequences

* Stimulus:
  + User initiates the logout action by clicking on the logout button or link.
  + User confirms the logout action, if necessary.
* Response:
  + The system terminates the user's session.
  + The system redirects the user to the login screen or a designated landing page.

### 3.3.3 Functional Requirements

REQ-1: Logout Process

* The system shall provide a logout button or link accessible from any page within the application.
* Upon clicking the logout button or link, the system shall prompt the user to confirm the logout action.
* If the user confirms the logout action, the system shall invalidate the user's session.
* After successful logout, the system shall redirect the user to the login screen or a designated landing page.

REQ-2: User Feedback

* Upon successful logout, the system shall display a confirmation message indicating that the user has been logged out.

## Book Search

### 3.4.1 Description and Priority

The book title search feature enables users to search for books by their titles within the system. This feature is of High priority as it is a fundamental aspect of the system's functionality and directly contributes to user experience and content discovery.

### 3.4.2 Stimulus/Response Sequences

* Stimulus:
  + User enters a book title into the search bar and initiates the search.
  + User submits the search query by pressing the search button or hitting Enter.
* Response:
  + The system processes the search query and retrieves relevant book titles based on the provided input.
  + If matching book titles are found, the system displays them in the search results.
  + If no matching book titles are found, the system notifies the user accordingly.

### 3.4.3 Functional Requirements

REQ-1: Search Functionality

* The system shall provide a search bar prominently displayed on the user interface for entering book titles.
* Upon submitting the search query, the system shall retrieve and display a list of book titles matching the search criteria.

REQ-2: Search Results

* The system shall display search results in a clear and organized manner, presenting relevant book titles along with additional information such as authors, publication dates, and genres.
* Each search result shall be clickable, allowing users to access detailed information about the corresponding book.

REQ-3: Error Handling

* If the search query yields no results, the system shall display a message indicating that no matching book titles were found.
* The system shall handle potential errors or exceptions during the search process gracefully, providing informative error messages to users when necessary.

## Book Browsing

### 3.5.1 Description and Priority

The book browsing feature allows users to explore and browse through a collection of books within the system. This feature is of Medium priority as it enhances user engagement and content discovery but may not be as critical as core functionalities.

### 3.5.2 Stimulus/Response Sequences

* Stimulus:
  + User navigates to the book browsing section.
  + User scrolls through the list of available books.
  + User clicks on a book to view detailed information.
* Response:
  + The system displays a grid or list of available books, showcasing cover images and brief details.
  + If the user clicks on a book, the system presents detailed information about the selected book.

### 3.5.3 Functional Requirements

REQ-1: Book Display

* The system shall present a visually appealing and organized display of books, including cover images, titles, authors, and brief descriptions.

REQ-2: Pagination and Infinite Scroll

* The system shall implement pagination or infinite scroll mechanisms to allow users to navigate through multiple pages of book listings without overwhelming the user interface.

REQ-3: Detailed Book Information

* When a user clicks on a book in the browsing interface, the system shall display detailed information about the selected book, including a larger cover image, author details, publication information, and a synopsis.

## Buying a Book

### 3.6.1 Description and Priority

Enabling buyers to purchase a book facilitates transactions within the system. This feature is of High priority as it directly supports the primary objective of the system, which is facilitating book transactions and generating revenue.

### 3.6.2 Stimulus/Response Sequences

* Stimulus:
  + User selects a book they wish to purchase.
  + User adds the selected book to their shopping cart.
  + User proceeds to checkout to complete the purchase.
* Response:
  + The system updates the shopping cart with the selected book.
  + Upon checkout, the system prompts the user for payment and shipping information.
  + After successful payment processing, the system confirms the purchase and provides the user with access to the purchased book.

### 3.6.3 Functional Requirements

REQ-1: Adding Books to Cart

* The system shall provide an "Add to Cart" button or similar functionality for users to add books to their shopping cart.
* Users shall be able to specify the quantity of books they wish to purchase before adding them to the cart.
* The system shall update the shopping cart in real-time to reflect the added books and their quantities.

REQ-2: Checkout Process

* The system shall provide a streamlined checkout process, guiding users through the steps required to complete the purchase.
* Upon checkout, the system shall prompt the user to provide payment information, including credit card details or alternative payment methods.
* The system shall also prompt the user to enter shipping information, including address details for physical book deliveries.

REQ-3: Order History and Tracking

* The system shall maintain a record of users' purchase history, including details of purchased books, transaction dates, and order statuses.
* Users shall be able to view their order history and track the status of their orders, including shipment tracking information if applicable.

REQ-4: Error Handling and Validation

* The system shall validate user input during the checkout process, ensuring that all required fields are filled out correctly.
* In case of errors or issues during payment processing, the system shall provide informative error messages to guide users in resolving the issue and completing the purchase successfully.

## Posting a Book to Sell

### 3.7.1 Description and Priority

Allowing users to post books for sale enables them to contribute to the marketplace and generate revenue. This feature is of High priority as it supports the primary objective of the system, which is facilitating book transactions and enhancing the diversity of available books.

### 3.7.2 Stimulus/Response Sequences

* Stimulus:
  + User navigates to the "Sell a Book" section or feature.
  + User provides details about the book they wish to sell, such as title, author, condition, price, and additional information.
  + User submits the book listing for sale.
* Response:
  + The system processes the submitted book listing and adds it to the marketplace.
  + Upon successful posting, the system confirms the listing and notifies the user.

### 3.7.3 Functional Requirements

REQ-1: Book Listing Creation

* The system shall provide a user-friendly interface for users to create and submit book listings for sale.
* Users shall be able to input relevant details about the book, including title, author, ISBN (International Standard Book Number), condition (e.g., new, like new, used), price, and optional descriptions.

REQ-2: Pricing and Negotiation

* Users shall have the flexibility to set the selling price for their books.

REQ-3: Listing Management

* Sellers shall have access to a dashboard or interface where they can manage their posted book listings.
* Sellers shall be able to edit, update, or remove existing listings as needed, including marking books as sold or unavailable.

REQ-4: Verification and Moderation

* The system shall implement verification mechanisms to ensure the accuracy and legitimacy of posted book listings.
* Listings may undergo moderation or review by administrators to enforce community guidelines and prevent fraudulent or inappropriate content.

REQ-5: Communication and Notification

* Sellers shall receive notifications about inquiries, offers, and purchases related to their book listings.

## Receiving Mock Payments for Sold Books

### 3.8.1 Description and Priority

The feature of receiving a mock payment for sold books simulates the payment process within the system. This feature is of High priority as it is crucial for testing and demonstrating the payment flow without actual financial transactions.

### 3.8.2 Stimulus/Response Sequences

* Stimulus:
  + User initiates the purchase of a book by adding it to the cart and proceeding to checkout.
  + User selects the mock payment option during checkout.
  + User confirms the purchase.
* Response:
  + The system simulates the payment process and generates a mock payment confirmation that notifies the seller.
  + Upon successful mock payment, the system marks the book as purchased.

### 3.8.3 Functional Requirements

REQ-1: Mock Payment Integration

* The system shall integrate a mock payment gateway or simulator to emulate the payment process.
* Users shall have the option to choose the mock payment method during checkout for testing purposes.

REQ-2: Payment Simulation

* Upon selecting the mock payment option, the system shall simulate the payment transaction without involving actual financial transactions.
* The system shall generate a mock payment confirmation and update the order status to reflect the successful payment.

REQ-3: Order Fulfillment

* Upon successful mock payment, the system shall fulfill the order by providing access to the purchased digital content or initiating the shipping process for physical books.

REQ-4: Payment Receivement by Seller

* Upon successful mock payment by a buyer, the book is mark as sold in the sellers UI and the seller is notified that they have received payment for the book.

## Admin Blocking New Users

### 3.9.1 Description and Priority

Admin Blocking New Users allows administrators to prevent certain users from registering in the system. This feature is of Medium priority as it contributes to user management and system security.

### 3.9.2 Stimulus/Response Sequences

* Stimulus:
  + Admin accesses the user management interface.
  + Admin identifies a user to be blocked and initiates the blocking action.
* Response:
  + The system processes the blocking request and updates the user's status to "Blocked."
  + Any attempts by the blocked user to register or access the system result in denial of access.

### 3.9.3 Functional Requirements

REQ-1: Admin Interface

* The system shall provide an interface accessible only to administrators for managing user accounts.
* The admin interface shall include options to view and manage user accounts, including the ability to block new users.

REQ-2: Blocking Action

* The admin shall have the ability to block specific users from registering in the system.
* When an admin blocks a user, the system shall update the user's status to "Blocked" in the user database.

REQ-3: Notification to User

* When an admin blocks a user, the system shall send a notification to the affected user, informing them of the blocking action.
* The notification shall include relevant information and instructions for resolving the issue, if applicable.

REQ-4: Unblock Functionality

* The admin interface shall include the option to unblock users, allowing previously blocked users to register and access the system.
* When an admin unblocks a user, the system shall update the user's status to "Active."

## Admin Blocking New Book Listings

### 3.10.1 Description and Priority

Admin Blocking New Book Listings allows administrators to prevent certain book listings from being published or displayed in the system. This feature is of Medium priority as it contributes to content moderation and quality control within the platform.

### 3.10.2 Stimulus/Response Sequences

* Stimulus:
  + Admin accesses the book management interface.
  + Admin identifies a book listing to be blocked and initiates the blocking action.
* Response:
  + The system processes the blocking request and hides the blocked book listing from public view.
  + Any attempts by users to access the blocked book listing result in an error or notification indicating that the listing is unavailable.

### 3.10.3 Functional Requirements

REQ-1: Admin Interface

* The system shall provide an interface accessible only to administrators for managing book listings.
* The admin interface shall include options to view and manage book listings, including the ability to block new listings.

REQ-2: Blocking Action

* The admin shall have the ability to block specific book listings from being published or displayed in the system.
* When an admin blocks a book listing, the system shall hide the listing from public view while retaining it in the system database.

REQ-3: Denial of Access

* Users shall be prevented from accessing blocked book listings through search results, category browsing, or direct links.

REQ-4: Notification to Users

* When an admin blocks a book listing, the system shall send a notification to the user who submitted the listing, informing them of the blocking action.

REQ-5: Unblock Functionality

* The admin interface shall include the option to unblock book listings, allowing previously blocked listings to be published or displayed again.
* When an admin unblocks a book listing, the system shall make the listing visible to users in the appropriate sections of the platform.

## Admin Overseeing Users Information

### 3.11.1 Description and Priority

Admin Overseeing Users Information allows administrators to access and manage user information within the system. This feature is of High priority as it facilitates user management, security, and compliance with privacy regulations.

### 3.11.2 Stimulus/Response Sequences

* Stimulus:
  + Admin accesses the user management interface.
  + Admin views a list of all users.
  + Admin selects a user to view or manage their information.
* Response:
  + The system retrieves and displays the user's information, including personal details, and account settings.
  + Admin can edit, update, or perform actions related to the user's account as needed.

### 3.11.3 Functional Requirements

REQ-1: Admin Interface

* The system shall provide an interface accessible only to administrators for overseeing user information.
* The admin interface shall include options to view and manage user accounts, as well as access user activity logs.

REQ-2: User Information Access

* Admins shall have access to comprehensive user profiles containing personal details, contact information, and account settings.

REQ-3: User Management Actions

* Admins shall be able to perform various actions on user accounts, such as account activation, deactivation, password resets, and privilege adjustments.

## Buyers Returning Books

### 3.12.1 Description and Priority

Allowing buyers to return books facilitates a smooth and customer-friendly return process within the system. This feature is of Medium priority as it directly impacts user experience and satisfaction, ensuring users have a reliable mechanism for returning unwanted or defective books.

### 3.12.2 Stimulus/Response Sequences

* Stimulus:
  + Buyer accesses the returns section.
  + Buyer selects the book they wish to return and initiates the return process.
* Response:
  + The system processes the return request.
  + The system updates the order status and initiates the refund process.

### 3.12.3 Functional Requirements

REQ-1: Return Policy

* The system shall enforce return policies, including eligibility criteria, timeframes, and conditions for returning books.
* Return policies shall be communicated to buyers during the purchasing process and available for reference in the user interface.

REQ-2: Refund Processing

* The system shall process refunds in accordance with the return policy.
* Refunds shall be issued to the original payment method used for the purchase within a reasonable timeframe.

REQ-3: Inventory Management

* The system shall update inventory levels and book availability based on returned items to ensure accurate stock management.

# **Other Nonfunctional Requirements**

## **Performance Requirements**

REQ-1: Speedy Response on All Forward Facing Functions:

* Functions that deal with the buyer or seller (non-admin users) should complete within 5 seconds in order to assure retention of users.
* An instant response should be displayed even if the function is loading in the backend

REQ-2: Handling Errors or Slow Performance:

* In the case of slow performance caused by outside circumstances, the system should display a “loading” message to the user to indicate that the system is attempting to complete the function even if it is taking a while.
* In the case of an error that causes a function to not complete, the system should display a message telling the user that something went wrong.

Note: This product has limited performance requirements. As long as all functions complete correctly and quickly enough that users don’t leave the page then the system should be considered adequate.

## **Safety Requirements**

REQ-1: User Privacy:

* Clearly outline and communicate the website's privacy policy, detailing how user data will be used and protected.
* Obtain explicit consent from users before collecting any personal information.

REQ-2: Content Accuracy:

* Ensure that the book information, including titles, authors, and descriptions, is accurate to prevent any misinformation that could lead to dissatisfaction or academic issues.

REQ-3: User Education:

* Provide clear instructions on how to use the website and make purchases to prevent user errors or misunderstandings.

REQ-4: Intellectual Property Rights:

* Ensure that the website respects copyright laws and does not infringe on the intellectual property rights of authors or publishers.

## **Security Requirements**

REQ-1: Data Security:

* All user data, including personal information and transaction details, must be stored securely to prevent unauthorized access.
* Store user passwords as hashed encrypted codes to prevent admins from reading and potentially leaking user passwords.

REQ-2: Transaction Security:

* Implement secure connections (HTTPS) to protect user data during the entire transaction process.

REQ-3: Secure Authentication:

* Implement strong authentication mechanisms to prevent unauthorized access to user accounts.

## **Software Quality Attribute**s

This project aims to be easy to use for a new user. There should not be a large learning curve for a new user. Flexibility and portability is also very important for this project as it is being developed in a short timespan.

Usability and Reusability:

* All code sections that are overly complex, verbose, or redundant should be contained inside functions for ease of access.
* All large functionality, especially those concerning communication between client and server, should be contained in an API or wrapper functions to abstract details away from the user.

Project Maintainability:

* All code and documentation will be constructed such that it is either self-explanatory, or accompanied by a brief explanation of its function.
* All functions and objects will be constructed so that their return values and parameters fit a common pattern that can be easily understood and predicted by engineers.
* All code edits and previous versions will be maintained and documented through the project's github.
  + All items pushed or edited on said github must be accompanied by a meaningful commit message that accurately describes changes.
  + Issues should be logged and documented through the github issues page for centralized, easy access. (

# **Other Requirements**

**Appendix A: Glossary**

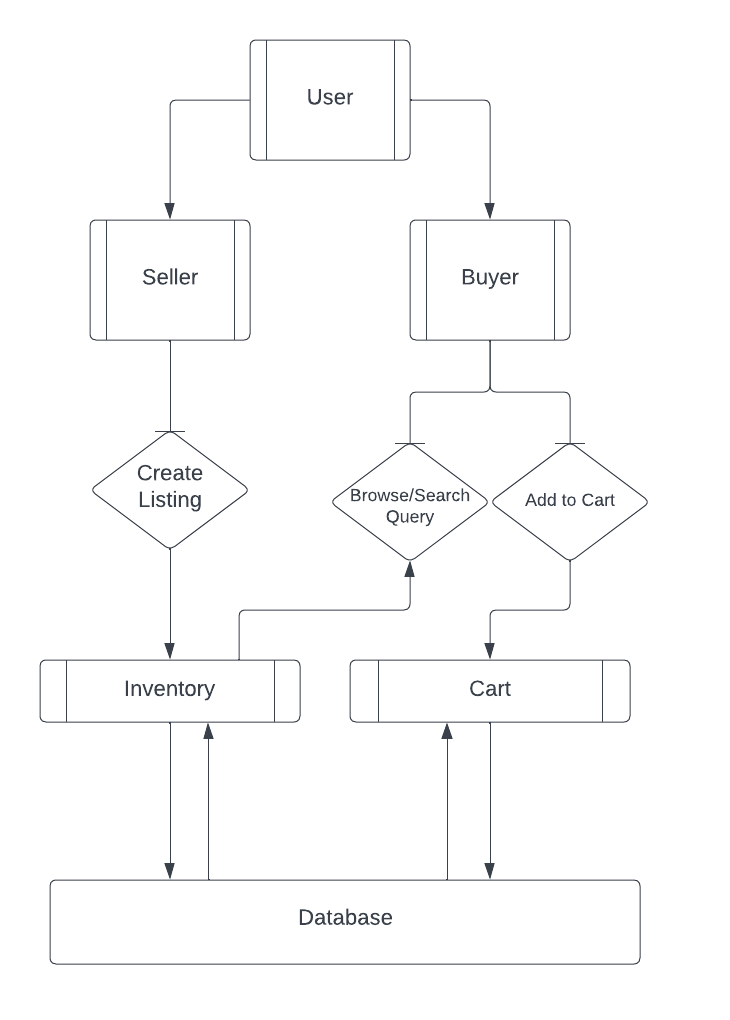
* Buyer - the user that intends on purchasing listed books on the site
* Seller - the user that intends on listing books onto the site for purchase by the buyer

**Appendix B: Analysis Models**

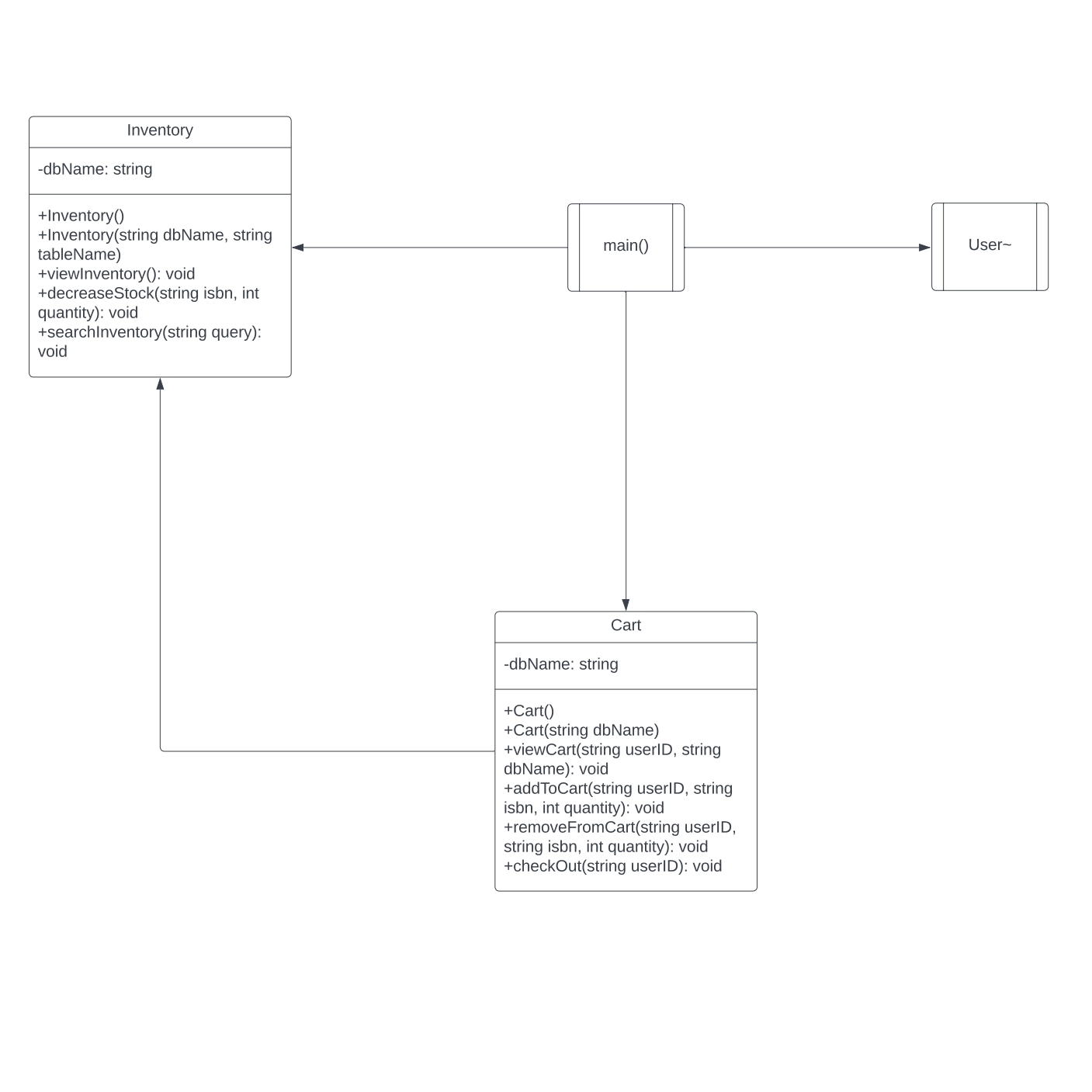
**B.1 User Classes Diagram**



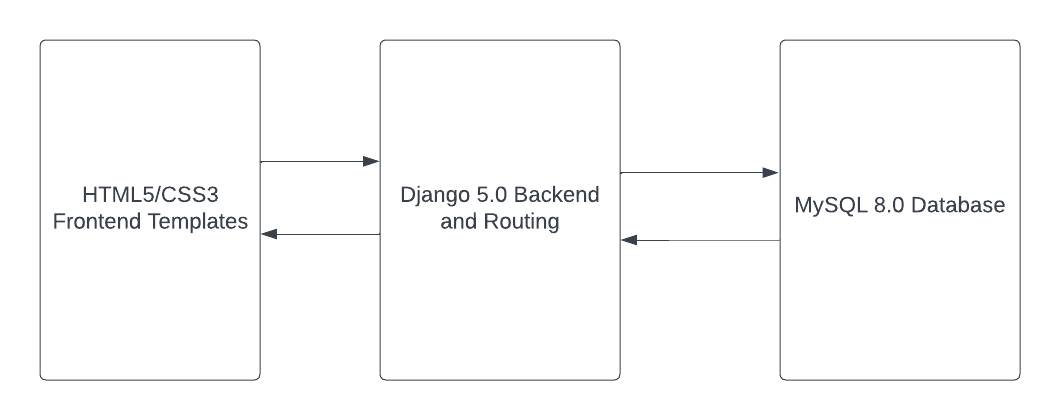
**B.2 Database Data Flow Chart**

****

**B.3 Class Interaction Flowchart**

****

**B.3 Systems Tech Relation Flowchart**



**Appendix C: To Be Determined List**

* *TBD*