**Section 1: Introduction**

**Team Name & Mascot**

**Team Captain:** Marshall Huckins

**Team Name:** *The Book Explorers*  
**Mascot:** *"Whiskers the Wise Owl"*

**Why this mascot?**

* Owls symbolize **wisdom, knowledge, and reading**, making them perfect for a book-related project.
* "Whiskers the Wise Owl" encourages **exploration and learning**, just like WhatABook helps users discover books.

|  |  |  |
| --- | --- | --- |
| Name | Role | Bio |
| Marshall Huckins | Team Captain | Marshall Huckins is a web development student with a keen interest in both front-end and back-end technologies, including HTML, CSS, JavaScript, and Node.js. He enjoys problem-solving and optimizing code to create seamless user experiences. Marshall is always looking to deepen his understanding of development principles and explore new frameworks. |
| Dua Hasan | Team Member | Dua Hasan is a web development student with a focus on both front-end and back-end technologies like HTML, CSS, JavaScript, and Node.js. Dua is passionate about building dynamic, user-friendly websites and constantly learning new tools and frameworks. In her free time, Dua enjoys exploring tech trends and working on personal projects. |
| Sara George | Team Member | Sara is a web design student with a long background in graphic design. Sara first started her journey in web development by experimenting with the creation of custom user profiles for sites like DeviantArt, Myspace, and Tumblr as well as basic Angelfire pages by using simple CSS and JavaScript. She has worked in graphic design and digital art from a young age, and is looking to implement those skills in the graphical design of unique and eye-catching webpages. |

**1.1 Purpose**

The WhatABook project aims to develop a **Python-based application** that connects to a **MongoDB database** to provide users with an intuitive book browsing and wishlist management experience. This application will enable customers to:

* Browse in-store book listings
* Search for books by title, genre, author, and book ID
* Add books to a personal wishlist
* View their wishlist
* Access a user-friendly interface tailored for middle-aged users with minimal computer experience

The primary goal of this project is to create a simple and accessible **console-based application** that facilitates easy interaction with the MongoDB database while meeting **business requirements** set by WhatABook.

**1.2 User Personas**

**Persona 1: Emily Carter**

* **Name:** Emily Carter
* **Distinguishing Characteristics:** Avid reader, enjoys fantasy novels, prefers dark mode UI
* **Photo/Illustration/Icon:** 📖
* **Features Used Most Often:** Search for books, add books to favorites, leave reviews
* **Features Rarely Used:** Administrative functions, book purchase options
* **Needs Met by Application:** Personalized book recommendations, an easy-to-navigate library
* **Computer Skill Level:** Intermediate
* **Additional Elements:** Prefers mobile apps, uses voice search, reads e-books, participates in online book clubs, frequently shares book reviews on social media, prefers dark mode for reading at night

**Persona 2: James Anderson**

* **Name:** James Anderson
* **Distinguishing Characteristics:** Bookstore manager, responsible for inventory
* **Photo/Illustration/Icon:** 📚
* **Features Used Most Often:** Manage book inventory, add/edit book details
* **Features Rarely Used:** Book recommendations, personal reviews
* **Needs Met by Application:** A simple dashboard for managing book listings
* **Computer Skill Level:** Advanced
* **Additional Elements:** Prefers desktop access, needs data export functionality, uses reports to track best-selling books, often trains new employees on inventory management, wants a bulk upload feature for adding books quickly

**Persona 3: Sarah Lee**

* **Name:** Sarah Lee
* **Distinguishing Characteristics:** College student, uses the library for research
* **Photo/Illustration/Icon:** 📝
* **Features Used Most Often:** Search books by category, add to reading list
* **Features Rarely Used:** Manage inventory, admin functions
* **Needs Met by Application:** Quick search with filters, citation generation
* **Computer Skill Level:** Beginner
* **Additional Elements:** Prefers web-based solutions, needs bookmarking feature, likes to create reading lists for different subjects, wants text-to-speech functionality for accessibility, uses cloud storage to sync notes across devices

**1.3 User Stories**

|  |  |  |
| --- | --- | --- |
| **User Story ID** | **Persona** | **User Story** |
| **US1** | Emily Carter | As a book lover, I want to search for books by title or genre, so that I can easily find books that interest me. |
| **US2** | Emily Carter | As a book lover, I want to add books to my favorites list, so that I can keep track of books I want to read. |
| **US3** | Emily Carter | As a book lover, I want to receive personalized book recommendations, so that I can discover new books based on my interests. |
| **US4** | Emily Carter | As a book lover, I want to leave reviews on books I have read, so that I can share my thoughts with other readers. |
| **US5** | Emily Carter | As a book lover, I want to filter books by rating, so that I can find highly rated books quickly. |
| **US6** | James Anderson | As a bookstore manager, I want to add new books to the inventory, so that customers can see updated book listings. |
| **US7** | James Anderson | As a bookstore manager, I want to edit book details, so that I can ensure the catalog is accurate. |
| **US8** | James Anderson | As a bookstore manager, I want to remove books that are out of stock, so that customers only see available books. |
| **US9** | James Anderson | As a bookstore manager, I want to generate reports on book sales, so that I can track inventory trends. |
| **US10** | James Anderson | As a bookstore manager, I want to set discounts for certain books, so that I can run promotions. |
| **US11** | Sarah Lee | As a student, I want to search for academic books by subject, so that I can find relevant research material. |
| **US12** | Sarah Lee | As a student, I want to add books to a reading list, so that I can keep track of books I need for my studies. |
| **US13** | Sarah Lee | As a student, I want to view book summaries, so that I can quickly determine if a book is useful for my research. |
| **US14** | Sarah Lee | As a student, I want to generate citations for books, so that I can properly reference them in my assignments. |
| **US15** | Sarah Lee | As a student, I want to bookmark specific pages in an e-book, so that I can easily revisit important information. |

**1.4 Story Decomposition & Tasking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID** | **Persona** | **User Story Description** | **Task #** | **Task Description** |
| **US1** | Emily Carter | Search for books by title or genre | T1 | Define UI requirements |
| **US1** | Emily Carter | Search for books by title or genre | T2 | Implement backend functionality |
| **US1** | Emily Carter | Search for books by title or genre | T3 | Connect UI to backend |
| **US1** | Emily Carter | Search for books by title or genre | T4 | Develop database schema |
| **US1** | Emily Carter | Search for books by title or genre | T5 | Test and optimize functionality |
| **US2** | Emily Carter | Add books to favorites list | T1 | Define UI requirements |
| **US2** | Emily Carter | Add books to favorites list | T2 | Implement backend functionality |
| **US2** | Emily Carter | Add books to favorites list | T3 | Connect UI to backend |
| **US2** | Emily Carter | Add books to favorites list | T4 | Develop database schema |
| **US2** | Emily Carter | Add books to favorites list | T5 | Test and optimize functionality |
| **US3** | Emily Carter | Receive personalized book recommendations | T1 | Define UI requirements |
| **US3** | Emily Carter | Receive personalized book recommendations | T2 | Implement backend functionality |
| **US3** | Emily Carter | Receive personalized book recommendations | T3 | Connect UI to backend |
| **US3** | Emily Carter | Receive personalized book recommendations | T4 | Develop database schema |
| **US3** | Emily Carter | Receive personalized book recommendations | T5 | Test and optimize functionality |
| **US4** | Emily Carter | Leave reviews on books | T1 | Define UI requirements |
| **US4** | Emily Carter | Leave reviews on books | T2 | Implement backend functionality |
| **US4** | Emily Carter | Leave reviews on books | T3 | Connect UI to backend |
| **US4** | Emily Carter | Leave reviews on books | T4 | Develop database schema |
| **US4** | Emily Carter | Leave reviews on books | T5 | Test and optimize functionality |
| **US5** | Emily Carter | Filter books by rating | T1 | Define UI requirements |
| **US5** | Emily Carter | Filter books by rating | T2 | Implement backend functionality |
| **US5** | Emily Carter | Filter books by rating | T3 | Connect UI to backend |
| **US5** | Emily Carter | Filter books by rating | T4 | Develop database schema |
| **US5** | Emily Carter | Filter books by rating | T5 | Test and optimize functionality |
| **US6** | James Anderson | Add new books to inventory | T1 | Define UI requirements |
| **US6** | James Anderson | Add new books to inventory | T2 | Implement backend functionality |
| **US6** | James Anderson | Add new books to inventory | T3 | Connect UI to backend |
| **US6** | James Anderson | Add new books to inventory | T4 | Develop database schema |
| **US6** | James Anderson | Add new books to inventory | T5 | Test and optimize functionality |
| **US7** | James Anderson | Edit book details | T1 | Define UI requirements |
| **US7** | James Anderson | Edit book details | T2 | Implement backend functionality |
| **US7** | James Anderson | Edit book details | T3 | Connect UI to backend |
| **US7** | James Anderson | Edit book details | T4 | Develop database schema |
| **US7** | James Anderson | Edit book details | T5 | Test and optimize functionality |
| **US8** | James Anderson | Remove books from inventory | T1 | Define UI requirements |
| **US8** | James Anderson | Remove books from inventory | T2 | Implement backend functionality |
| **US8** | James Anderson | Remove books from inventory | T3 | Connect UI to backend |
| **US8** | James Anderson | Remove books from inventory | T4 | Develop database schema |
| **US8** | James Anderson | Remove books from inventory | T5 | Test and optimize functionality |
| **US9** | James Anderson | Generate reports on book sales | T1 | Define UI requirements |
| **US9** | James Anderson | Generate reports on book sales | T2 | Implement backend functionality |
| **US9** | James Anderson | Generate reports on book sales | T3 | Connect UI to backend |
| **US9** | James Anderson | Generate reports on book sales | T4 | Develop database schema |
| **US9** | James Anderson | Generate reports on book sales | T5 | Test and optimize functionality |
| **US10** | James Anderson | Set discounts for books | T1 | Define UI requirements |
| **US10** | James Anderson | Set discounts for books | T2 | Implement backend functionality |
| **US10** | James Anderson | Set discounts for books | T3 | Connect UI to backend |
| **US10** | James Anderson | Set discounts for books | T4 | Develop database schema |
| **US10** | James Anderson | Set discounts for books | T5 | Test and optimize functionality |
| **US11** | Sarah Lee | Search for academic books | T1 | Define UI requirements |
| **US11** | Sarah Lee | Search for academic books | T2 | Implement backend functionality |
| **US11** | Sarah Lee | Search for academic books | T3 | Connect UI to backend |
| **US11** | Sarah Lee | Search for academic books | T4 | Develop database schema |
| **US11** | Sarah Lee | Search for academic books | T5 | Test and optimize functionality |
| **US12** | Sarah Lee | Add books to reading list | T1 | Define UI requirements |
| **US12** | Sarah Lee | Add books to reading list | T2 | Implement backend functionality |
| **US12** | Sarah Lee | Add books to reading list | T3 | Connect UI to backend |
| **US12** | Sarah Lee | Add books to reading list | T4 | Develop database schema |
| **US12** | Sarah Lee | Add books to reading list | T5 | Test and optimize functionality |
| **US13** | Sarah Lee | View book summaries | T1 | Define UI requirements |
| **US13** | Sarah Lee | View book summaries | T2 | Implement backend functionality |
| **US13** | Sarah Lee | View book summaries | T3 | Connect UI to backend |
| **US13** | Sarah Lee | View book summaries | T4 | Develop database schema |
| **US13** | Sarah Lee | View book summaries | T5 | Test and optimize functionality |
| **US14** | Sarah Lee | Generate citations for books | T1 | Define UI requirements |
| **US14** | Sarah Lee | Generate citations for books | T2 | Implement backend functionality |
| **US14** | Sarah Lee | Generate citations for books | T3 | Connect UI to backend |
| **US14** | Sarah Lee | Generate citations for books | T4 | Develop database schema |
| **US14** | Sarah Lee | Generate citations for books | T5 | Test and optimize functionality |
| **US15** | Sarah Lee | Bookmark pages in e-books | T1 | Define UI requirements |
| **US15** | Sarah Lee | Bookmark pages in e-books | T2 | Implement backend functionality |
| **US15** | Sarah Lee | Bookmark pages in e-books | T3 | Connect UI to backend |
| **US15** | Sarah Lee | Bookmark pages in e-books | T4 | Develop database schema |
| **US15** | Sarah Lee | Bookmark pages in e-books | T5 | Test and optimize functionality |

**1.4 Work Estimation (story points)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID** | **Persona** | **User Story Description** | **Task #** | **Task Description** | **Work Hours** | **Story Points** |
| US1 | Emily Carter | Search for books by title or genre | T1 | Define UI requirements | 3 | 1 |
| US1 | Emily Carter | Search for books by title or genre | T2 | Implement backend functionality | 4 | 2 |
| US1 | Emily Carter | Search for books by title or genre | T3 | Connect UI to backend | 5 | 3 |
| US1 | Emily Carter | Search for books by title or genre | T4 | Develop database schema | 6 | 5 |
| US1 | Emily Carter | Search for books by title or genre | T5 | Test and optimize functionality | 8 | 8 |
| US2 | Emily Carter | Add books to favorites list | T1 | Define UI requirements | 3 | 1 |
| US2 | Emily Carter | Add books to favorites list | T2 | Implement backend functionality | 4 | 2 |
| US2 | Emily Carter | Add books to favorites list | T3 | Connect UI to backend | 5 | 3 |
| US2 | Emily Carter | Add books to favorites list | T4 | Develop database schema | 6 | 5 |
| US2 | Emily Carter | Add books to favorites list | T5 | Test and optimize functionality | 8 | 8 |
| US3 | Emily Carter | Receive personalized book recommendations | T1 | Define UI requirements | 3 | 1 |
| US3 | Emily Carter | Receive personalized book recommendations | T2 | Implement backend functionality | 4 | 2 |
| US3 | Emily Carter | Receive personalized book recommendations | T3 | Connect UI to backend | 5 | 3 |
| US3 | Emily Carter | Receive personalized book recommendations | T4 | Develop database schema | 6 | 5 |
| US3 | Emily Carter | Receive personalized book recommendations | T5 | Test and optimize functionality | 8 | 8 |
| US4 | Emily Carter | Leave reviews on books | T1 | Define UI requirements | 3 | 1 |
| US4 | Emily Carter | Leave reviews on books | T2 | Implement backend functionality | 4 | 2 |
| US4 | Emily Carter | Leave reviews on books | T3 | Connect UI to backend | 5 | 3 |
| US4 | Emily Carter | Leave reviews on books | T4 | Develop database schema | 6 | 5 |
| US4 | Emily Carter | Leave reviews on books | T5 | Test and optimize functionality | 8 | 8 |
| US5 | Emily Carter | Filter books by rating | T1 | Define UI requirements | 3 | 1 |
| US5 | Emily Carter | Filter books by rating | T2 | Implement backend functionality | 4 | 2 |
| US5 | Emily Carter | Filter books by rating | T3 | Connect UI to backend | 5 | 3 |
| US5 | Emily Carter | Filter books by rating | T4 | Develop database schema | 6 | 5 |
| US5 | Emily Carter | Filter books by rating | T5 | Test and optimize functionality | 8 | 8 |
| US6 | James Anderson | Add new books to inventory | T1 | Define UI requirements | 3 | 1 |
| US6 | James Anderson | Add new books to inventory | T2 | Implement backend functionality | 4 | 2 |
| US6 | James Anderson | Add new books to inventory | T3 | Connect UI to backend | 5 | 3 |
| US6 | James Anderson | Add new books to inventory | T4 | Develop database schema | 6 | 5 |
| US6 | James Anderson | Add new books to inventory | T5 | Test and optimize functionality | 8 | 8 |
| US7 | James Anderson | Edit book details | T1 | Define UI requirements | 3 | 1 |
| US7 | James Anderson | Edit book details | T2 | Implement backend functionality | 4 | 2 |
| US7 | James Anderson | Edit book details | T3 | Connect UI to backend | 5 | 3 |
| US7 | James Anderson | Edit book details | T4 | Develop database schema | 6 | 5 |
| US7 | James Anderson | Edit book details | T5 | Test and optimize functionality | 8 | 8 |
| US8 | James Anderson | Remove books from inventory | T1 | Define UI requirements | 3 | 1 |
| US8 | James Anderson | Remove books from inventory | T2 | Implement backend functionality | 4 | 2 |
| US8 | James Anderson | Remove books from inventory | T3 | Connect UI to backend | 5 | 3 |
| US8 | James Anderson | Remove books from inventory | T4 | Develop database schema | 6 | 5 |
| US8 | James Anderson | Remove books from inventory | T5 | Test and optimize functionality | 8 | 8 |
| US9 | James Anderson | Generate reports on book sales | T1 | Define UI requirements | 3 | 1 |
| US9 | James Anderson | Generate reports on book sales | T2 | Implement backend functionality | 4 | 2 |
| US9 | James Anderson | Generate reports on book sales | T3 | Connect UI to backend | 5 | 3 |
| US9 | James Anderson | Generate reports on book sales | T4 | Develop database schema | 6 | 5 |
| US9 | James Anderson | Generate reports on book sales | T5 | Test and optimize functionality | 8 | 8 |
| US10 | James Anderson | Set discounts for books | T1 | Define UI requirements | 3 | 1 |
| US10 | James Anderson | Set discounts for books | T2 | Implement backend functionality | 4 | 2 |
| US10 | James Anderson | Set discounts for books | T3 | Connect UI to backend | 5 | 3 |
| US10 | James Anderson | Set discounts for books | T4 | Develop database schema | 6 | 5 |
| US10 | James Anderson | Set discounts for books | T5 | Test and optimize functionality | 8 | 8 |
| US11 | Sarah Lee | Search for academic books | T1 | Define UI requirements | 3 | 1 |
| US11 | Sarah Lee | Search for academic books | T2 | Implement backend functionality | 4 | 2 |
| US11 | Sarah Lee | Search for academic books | T3 | Connect UI to backend | 5 | 3 |
| US11 | Sarah Lee | Search for academic books | T4 | Develop database schema | 6 | 5 |
| US11 | Sarah Lee | Search for academic books | T5 | Test and optimize functionality | 8 | 8 |
| US12 | Sarah Lee | Add books to reading list | T1 | Define UI requirements | 3 | 1 |
| US12 | Sarah Lee | Add books to reading list | T2 | Implement backend functionality | 4 | 2 |
| US12 | Sarah Lee | Add books to reading list | T3 | Connect UI to backend | 5 | 3 |
| US12 | Sarah Lee | Add books to reading list | T4 | Develop database schema | 6 | 5 |
| US12 | Sarah Lee | Add books to reading list | T5 | Test and optimize functionality | 8 | 8 |
| US13 | Sarah Lee | View book summaries | T1 | Define UI requirements | 3 | 1 |
| US13 | Sarah Lee | View book summaries | T2 | Implement backend functionality | 4 | 2 |
| US13 | Sarah Lee | View book summaries | T3 | Connect UI to backend | 5 | 3 |
| US13 | Sarah Lee | View book summaries | T4 | Develop database schema | 6 | 5 |
| US13 | Sarah Lee | View book summaries | T5 | Test and optimize functionality | 8 | 8 |
| US14 | Sarah Lee | Generate citations for books | T1 | Define UI requirements | 3 | 1 |
| US14 | Sarah Lee | Generate citations for books | T2 | Implement backend functionality | 4 | 2 |
| US14 | Sarah Lee | Generate citations for books | T3 | Connect UI to backend | 5 | 3 |
| US14 | Sarah Lee | Generate citations for books | T4 | Develop database schema | 6 | 5 |
| US14 | Sarah Lee | Generate citations for books | T5 | Test and optimize functionality | 8 | 8 |
| US15 | Sarah Lee | Bookmark pages in e-books | T1 | Define UI requirements | 3 | 1 |
| US15 | Sarah Lee | Bookmark pages in e-books | T2 | Implement backend functionality | 4 | 2 |
| US15 | Sarah Lee | Bookmark pages in e-books | T3 | Connect UI to backend | 5 | 3 |
| US15 | Sarah Lee | Bookmark pages in e-books | T4 | Develop database schema | 6 | 5 |
| US15 | Sarah Lee | Bookmark pages in e-books | T5 | Test and optimize functionality | 8 | 8 |

**Section 2: Process Design**

**2.1 Sitemap**

The sitemap for **WhatABook** is designed to provide an intuitive and structured navigation experience for users. Below is a hierarchical representation of the sitemap:

* **Home**
  + Browse Books
    - View Book Details
    - Search by Title
    - Search by Genre
    - Search by Author
  + Wishlist
    - Add to Wishlist
    - Remove from Wishlist
    - View Wishlist
  + Reviews
    - Add a Review
    - View Reviews
  + Account
    - Login
    - Register
    - Profile Settings
  + Admin (Only for Bookstore Managers)
    - Add/Edit Books
    - Remove Books
    - Manage Discounts
    - Generate Reports
  + Help & Support
    - FAQ
    - Contact Us

This sitemap ensures that users can easily navigate the application while distinguishing between **regular users (book lovers and students)** and **admin users (bookstore managers)** to provide the appropriate level of access and functionality.

**2.2 Prototypes**

The **prototype** for the WhatABook application consists of **hand-drawn sketches** outlining the core functionalities and user interface. These sketches are designed to align with **usability principles** and best practices for an intuitive experience.

**Key UI Principles Used:**

1. **Consistency** - The interface follows a uniform layout with standardized navigation elements across all pages.
2. **Simplicity** - The design minimizes cognitive load by presenting only essential features to the user.
3. **Accessibility** - Ensures proper font sizes, color contrast, and easy navigation for users of different skill levels.
4. **Mobile-First Approach** - The design prioritizes responsiveness, ensuring usability across desktop and mobile devices.

**Sketches Overview:**

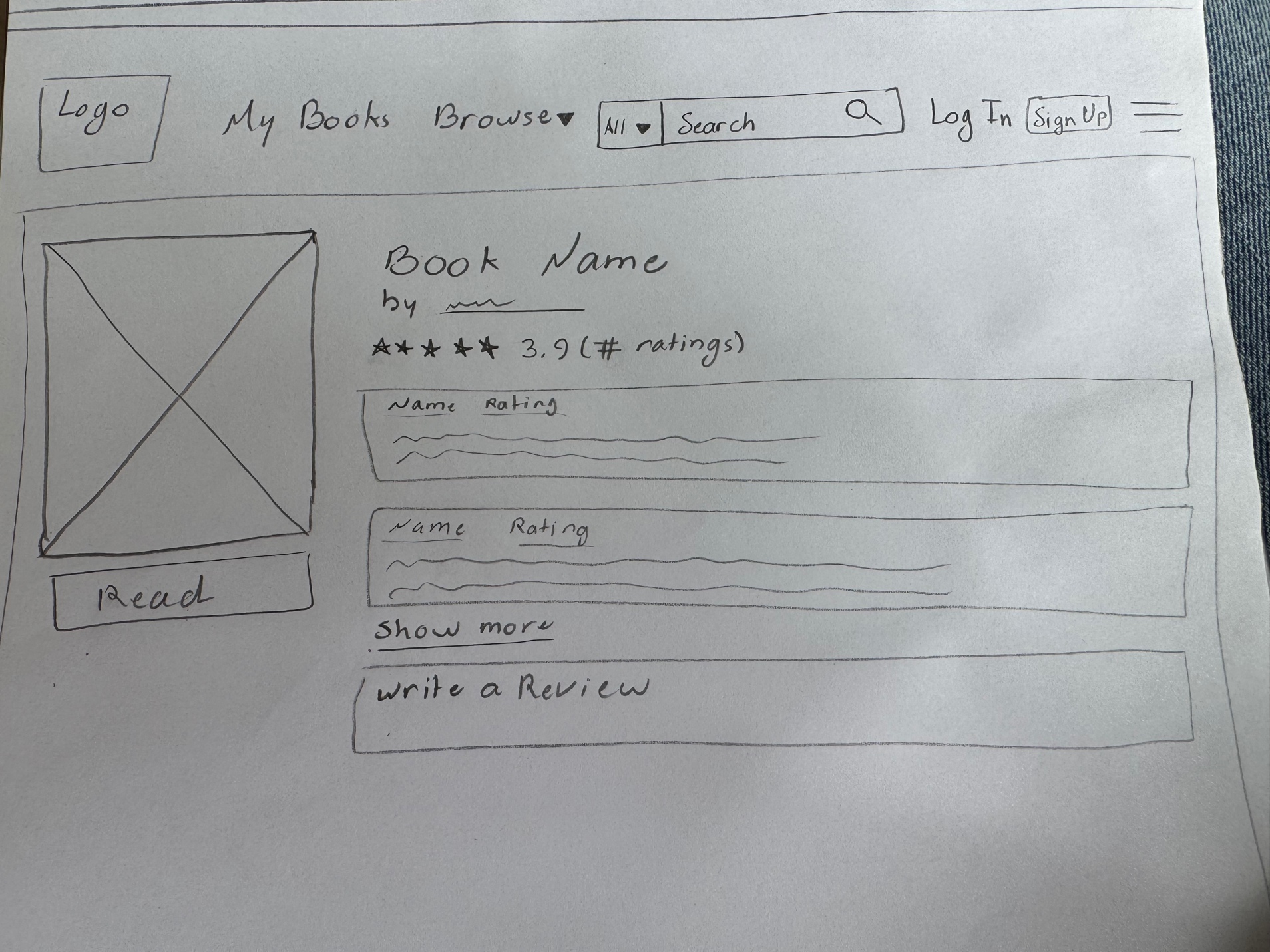
* **Home Page:** Displays featured books, search bar, and navigation menu.
* **Book Details Page:** Shows book information, ratings, reviews, and options to add to wishlist.
* **Wishlist Page:** Allows users to view and manage their saved books.
* **Review Section:** Provides functionality to add/view user reviews.
* **Admin Dashboard:** Contains options for inventory management, reporting, and discount settings.

**A piece of paper with writing on it

Description automatically generated**

**A close-up of a web page

Description automatically generatedA close-up of a book page

Description automatically generatedA sketch of a website

Description automatically generated**

**2.3 Object-Relational Diagram (ORD)**

**A diagram of a diagram

Description automatically generated with medium confidence**

**2.4 NoSQL Data Structure**

**1. customers Collection - Stores customer information.**

**Schema:**

{

"\_id": "ObjectId",

"firstName": "String",

"lastName": "String"

}

**Example Document:**

{

"\_id": ObjectId("60d5ec49d6e5a2b5f8a5a001"),

"firstName": "John",

"lastName": "Doe"

}

**2. books Collection - Stores book details.**

**Schema:**

{

"\_id": "ObjectId",

"title": "String",

"genre": "String",

"author": "String"

}

**Example Document:**

{

"\_id": ObjectId("60d5ec49d6e5a2b5f8a5a002"),

"title": "The Great Gatsby",

"genre": "Classic",

"author": "F. Scott Fitzgerald"

}

**3. wishlistItems Collection -Tracks which customers have added books to their wishlist.**

**Schema:**

{

"\_id": "ObjectId",

"customerId": "ObjectId (FK to customers.\_id)",

"bookId": "ObjectId (FK to books.\_id)"

}

**Example Document:**

{

"\_id": ObjectId("60d5ec49d6e5a2b5f8a5a003"),

"customerId": ObjectId("60d5ec49d6e5a2b5f8a5a001"),

"bookId": ObjectId("60d5ec49d6e5a2b5f8a5a002")

}

**Relationships:**

1:M → A customer can have multiple wishlist items (one customer, many wishlist books).

1:M → A book can appear in multiple customers’ wishlists.

**Section 3: Retrospective**

**3.1 Team Challenges**

One of the main challenges our team faced was coordinating work in an online course setting. Since team members were in different time zones or had different schedules, it was difficult to align meeting times and collaborate effectively. Additionally, working with MongoDB and structuring NoSQL queries posed a learning curve for some team members, requiring extra research and troubleshooting.

Another challenge was ensuring consistent documentation and integration of different parts of the project. With multiple people working on different sections, keeping track of progress and avoiding redundant work required clear communication.

**3.2 What Would You Do Differently?**

If we were to approach this project again, we would implement a structured meeting schedule early on to define clear milestones and check-ins. Additionally, assigning more defined roles and responsibilities at the start would streamline work and reduce duplication of effort.

To address the challenges with MongoDB, having more structured practice sessions before diving into the main project could improve efficiency and confidence with NoSQL queries. Lastly, establishing a document version control system, such as using GitHub more effectively, would enhance collaboration and prevent documentation inconsistencies.