# Software Requirements Specification

## for

# HootAndTrade

**Version 3.0 approved**

**Prepared by Andres Dominguez, Hober Granados, Alvaro Daniel Guardado, Levy Ocampo-Rivera, Vincent Rodriguez, and Jose Andres Zamora**

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| Template | 11/26/2023 | Initial Version of Document | 1.0 |
| First Draft | 11/27/2023 | Rough Draft for First Demo | 2.0 |
| Final Draft | 12/6/2023 | Final Draft for Submission | 3.0 |

# 1. Introduction

## 1.1 Purpose

This document will identify the requirements for the features implemented in the HootAndTrade website. In general, HootAndTrade is a website where users can upload books to be available to all other users for download. Some of the features implemented include search, shopping cart, and comment/message writing features.

## 1.2 Intended Audience and Reading Suggestions

This document is written for and to be read by a project lead, developers, and users of this software. The following sections are intended for each audience:

- Project leads should read Sections 2 - 5 to fully understand the project to properly plan development.
- Developers should read Sections 3 - 5 to know the requirements for successful completion of the services.
- Users should read Sections 1 & 5 to understand what the website is meant for and the policies it follows.

## 1.3 Product Scope

This software is a website intended to act as a platform for the sale and distribution of various books. Users will be allowed to list their products and view other user's listings. Users can rate each other's material and also utilize a search bar feature to assist users in finding what they are looking for. Lastly, there is a page with information about the development team for users who are curious about the team behind this software. Through these features, our web application strives to make textbooks more accessible to students, and the pursuit of knowledge more accessible.

## 1.4 Definitions, Acronyms, and Abbreviations

- Django: a high-level, open-source web framework for building web applications using the Python programming language.
- MTV Architecture: Model-View-Template is the architectural pattern used by the Django web framework.
- CSS: Cascading Style Sheets
- HTML: Hyper Text Markup Language

## 1.5 References

- "Group 4 Software Design Specification"

# 2. Overall Description

## 2.1 System Analysis

Books, as a product, offer the unique quality of providing enjoyment on multiple occasions during ownership. Nevertheless, many individuals find themselves with an abundance of books, whether stored on a bookshelf or within a personal library. A book exchange platform presents a valuable opportunity for owners to lend or sell books they are willing to part with, benefiting both the owners and potential customers. This system facilitates a practical and mutually beneficial solution for individuals seeking to share their literary collections.

The development team has encountered difficulty during the following efforts:

- Installing and connecting a shared project and creating branches from main.
- Adding items to the shopping cart and having all items appear in the shopping cart page.

The decision to employ a website as the hosting platform for this software stems from its capacity to provide convenient accessibility to a diverse user base comprising both sellers and buyers. This choice enables concurrent access by multiple users, facilitating swift communication with servers for efficient data storage of listings.

## 2.2 Product Perspective

This software is a standalone application without any relation to any other systems. It is based on previous e-commerce websites such as eBay, and Amazon along with specific book companies such as Barnes & Noble and local libraries. Unlike existing companies that sell books, this software allows users to user sale of books without a middleman and also allows sellers to resell any literature they wish. This is similar to e-commerce platforms with the main difference being a focus on book exchange.

## 2.3 Product Functions

- About Us: Gives more info on the website and developers.
- Search bar: Allows users to search for a specific book in our database.
- Navbar: Allows users to access other pages.
- Rating: Allows users to rate a particular book.
- Shopping Cart: Allows users to pick books and purchase them all at once.
- Comments: Allows users to submit feedback to the developers.
- Favorite List: Allows users to add purchased books to a separate list of favorites.
- Future implementation:
  - Filtering
  - Wishlists
  - Book Recommendations

## 2.4 User Classes and Characteristics

There are two user classes for this software:

- Public users
  - Public users will be able to view the homepage, about page, and browse the book listings. To access other features such as rating or making a purchase the user is required to make an account.
- Account holders
  - Account holders can access every feature public users have access to along with features oriented around the sale and purchase of books. To verify their status they are required to log in to their account.

## 2.5 Operating Environment

This application will be able to run on any device with access to a web browser and connection to the internet. This includes laptops, desktops, mobile devices, etc.

## 2.6 Design and Implementation Constraints

- Web Framework: Django was used for this software's web framework along with a built-in template language.
- Python: Python is the language to construct the project and Anaconda and Pycharm were used as well
- HTML/CSS: Web programming language used to create a website along with Django

## 2.7 User Documentation

No other documentation will be provided to the user.

## 2.8 Assumptions and Dependencies

- A user must have a device with the capability to use a web browser
- A user must have access to a suitable internet connection
- A user is assumed to be able to read English (no translation options will be provided)

## 2.9 Apportioning of Requirements

NA

# 3. External Interface Requirements

## 3.1 User Interfaces

This software is accessible by users through the Internet. Public users will be shown a login page prompting them to sign into an existing account or create a new account. Access to pages outside of the login is restricted only to account holders.

- User interface will present the home page with a login on first access
- This page will prompt the user to create a new account or sign into an existing account
  - A new account will create a new user and their username and password inside Django
  - If a user successfully signs in they will be given access to the rest of the software and be allowed to view their account info and listing info
- The look of every page is built upon a base to create a consistent appearance across every page
- A logo and banner will be on top of each web page for public users and account holders. Along with this, a navbar is on top to allow users to navigate through the web pages for account holders.

Accessibility: Images will have alternate text options and font sizes will adjust with screen size for easy readability. For those who are fully or partially blind, the assumption is made that they are utilizing features built into their operating system or external support to access this software.

## 3.2 Hardware Interfaces

No hardware is utilized

## 3.3 Software Interfaces

- 3rd party payment processing will be implemented in a real-world application to complete each transaction
- 3rd party shipping companies will be utilized to deliver products between users along with giving users an option for shipping methods(overnight, free shipping, etc.)

## 3.4 Communications Interfaces

- This software is accessible with any web browser including mobile devices.
- The client-server model will be utilized with HTTP Secure to communicate between the user and the webpage’s server
- Email notifications will be implemented in a real-world example.

# 4. Requirements Specification

## 4.1 Functional Requirements

4.1.1 Login

4.1.1.1 The system shall receive the user’s inputted username

4.1.1.2 The system shall receive the user’s inputted password

4.1.1.3 The system shall prompt the user to register if the username and password are not recognized

4.1.2 About Us

4.1.2.1 The system shall display a description of the website and the team of developers

4.1.3 Post Book

4.1.3.1 The system shall accept a title inputted by the user

4.1.3.2 The system shall accept an author inputted by the user

4.1.3.3 The system shall accept an image uploaded by the user as the book cover

4.1.4 Display Books

4.1.4.1 The system shall display the cover, title, and author of each of the books in the database so far.

4.1.4.2 The system shall take the user to a page with all of the details of a book upon clicking on a book.

4.1.5 Search Bar

4.1.5.1 The system shall accept text inputted by the user

4.1.5.2 The system shall find books whose details match the text inputted by the user in the search bar.

4.1.6 Navbar

4.1.6.1 The system shall provide links to the other pages of the website in a navigation bar

4.1.7 My Books

4.1.7.1 The system shall find the books in the database that have been purchased by the user and display them.

4.1.8 Rating

4.1.8.1 The system shall allow the user to submit a rating out of five on a particular book.

4.1.9 Shopping Cart

4.1.9.1 The system shall keep track of the books that the user sent to the shopping cart

4.1.10 Favorite List

4.1.10.1 The system shall find the books in our database that the user has favorited and display

## 4.2 External Interface Requirements

4.2.1 User Interface

4.2.1.1 Front End design through HTML and CSS & Django

4.2.1.2 Back End design through Python & Django

## 4.3 Logical Database Requirements

4.3.1 User Data

4.3.1.1 Username

4.3.1.2 Password

4.3.2 Book Data

4.3.2.1 ID

4.3.2.2 Title

4.3.2.3 Author

4.3.2.4 Price

4.3.2.5 Cover Image

4.3.3 My Books

4.3.3.1 List of BookItems

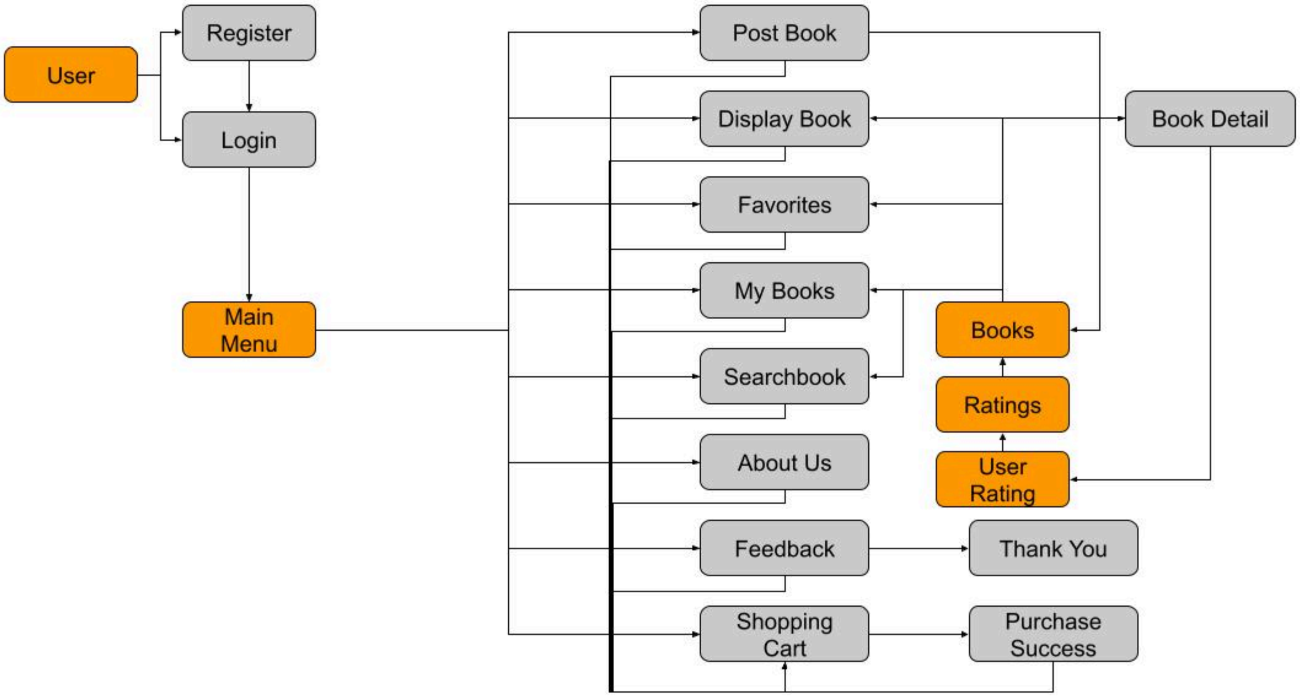
4.3.3.1.1 Book ID

4.3.3.1.2 isFavorite

4.3.4 Shopping Cart

4.3.4.1 List of BookID

4.3.5 Data entities and their relationships



## 4.4 Design Constraints

4.4.1 Python

4.4.1.1 Django itself is written in Python. The core components, including the request/response handling, URL routing, and database models, are all implemented in Python.

4.4.2 HTML & CSS

4.4.2.1 Django uses templates to generate HTML dynamically. These templates are HTML files with embedded template tags and variables. Template tags are enclosed in {% %} and allow for the inclusion of logic, control structures, and dynamic data. HTML pages can be styled with CSS.

4.4.3 Django

4.4.3.1 Django provides an admin interface for managing site content and user data. This feature is automatically generated based on the data models, saving development time and providing an out-of-the-box solution for content management.

4.4.4 Google SMTP

4.4.4.1 This service allows users to send emails using their Gmail or Google Workspace email addresses through Google's SMTP servers.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

The website is expected to operate seamlessly with a reliable internet connection, ensuring that all server requests are processed within one second. Every page, hyperlink, and function should be navigable and interconnected, creating a cohesive and efficient user experience.

## 5.2 Safety Requirements

The website is committed to ensuring the security of client information through robust measures. To safeguard user credentials, passwords will be hashed using strong, one-way hashing algorithms coupled with unique salts. This ensures that even in the unlikely event of unauthorized access, sensitive password information remains irreversibly encrypted.

## 5.3 Security Requirements

Furthermore, the website will implement Two-Factor Authentication (2FA) using Google's SMTP service. With 2FA, users will undergo an additional layer of verification beyond their passwords. This involves a temporary code sent to their registered email address through Google's secure SMTP servers. By leveraging this combination of advanced password protection and 2FA, the website aims to fortify the integrity of client information, providing a secure and trustworthy user experience.

## 5.4 Software Quality Attributes

Django's admin page greatly aids in the maintainability of the website by offering an intuitive and powerful interface. It enables efficient content and database management, allowing authorized users to easily add, edit, or delete content without direct database access. The admin page also facilitates user management, customization, and extensibility, providing flexibility to accommodate specific project requirements. With automatic form generation, built-in data validation, and consistent UI design, administrators can perform routine maintenance tasks with efficiency and accuracy. The audit trail feature ensures traceability of actions, enhancing accountability and simplifying the identification and reversal of unintended changes. Overall, Django's admin page streamlines the ongoing maintenance and management of the website, contributing to a secure, organized, and user-friendly system.

## 5.5 Business Rules

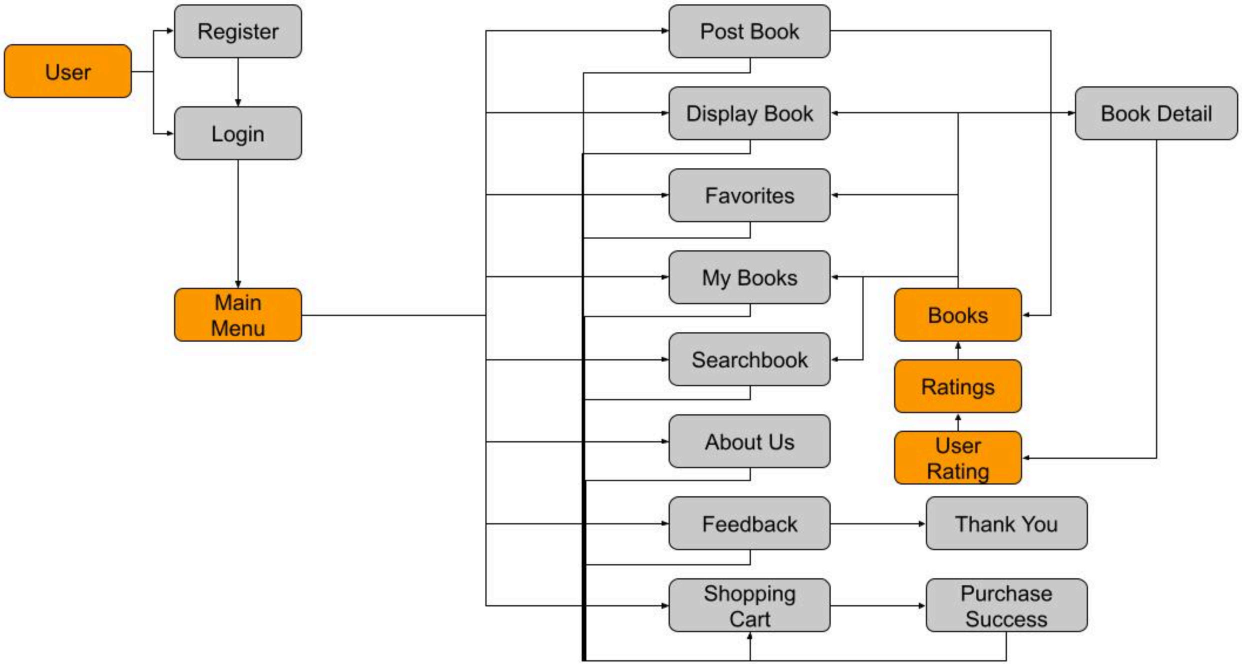
Maintenance of the website can only be done by those with access to the admin page which can not be reached through any of the public webpages. Thus public users will not be able to change any of the existing information in the database but merely add to it.

# 6. Legal and Ethical Considerations

- Piracy and Counterfeit Copies:
    - The sale of pirated or counterfeit copies of books raises ethical concerns. It not only deprives authors and publishers of rightful earnings but also undermines the integrity of the publishing industry.
- Privacy and Data Security:
    - Online platforms that facilitate book sales and exchanges may handle user data. Ensuring the privacy and security of users' personal information is essential to prevent unauthorized access, data breaches, or misuse of sensitive data.
- Fair Compensation for Authors:
    - Ethical concerns may arise if authors are not fairly compensated for the resale of their books in secondary markets. Ensuring that authors receive due royalties and acknowledging their intellectual property rights is essential for ethical book exchanges.
- Censorship and Content Restrictions:
    - Platforms involved in book sales and exchanges must navigate issues related to censorship and content restrictions. Striking a balance between freedom of expression and preventing the sale of potentially harmful or illegal content is a complex ethical consideration.

# Appendix A: Analysis Models

## Entity Relationships


User
Register
Login
Main Menu
Post Book
Display Book
Favorites
My Books
Searchbook
About Us
Feedback
Shopping Cart
Book Detail
Books
Ratings
User Rating
Thank You
Purchase Success