

BOOKSHELF

SOFTWARE REQUIREMENTS SPECIFICATION

<Version 1.0>

In fulfillment of CMSC 128 Laboratory requirements

Submitted by:

Keanna Louie Sun

Submitted on:

January 6, 2022

TABLE OF CONTENTS

	Page no.
I. Introduction	
1.1 Purpose	3
1.2 Intended Audience	3
1.3 Intended Use	3
1.4 Scope	3
1.5 Definitions and Acronyms	3
II. Overall Description	
2.1 User Needs	4
2.2 Assumptions and Dependencies	4
III. System Features and Requirements	
3.1 Functional Requirements	5
3.1.1 User Use Cases	5
3.2 External Interface Requirements	6
3.3 System Features	7
3.4 Nonfunctional Requirements	7

I. Introduction

1.1 Purpose

The purpose of this document is to specify all the functional and non-functional requirements of the software application Bookshelf. This document aims to help the reader have a comprehensive understanding of the features of Bookshelf, and also how it works in the back-end. This document is intended to be read by the stakeholders and developers and will be sent for approval.

1.2 Intended Audience

The developers and project managers of Bookshelf will have direct and unguarded access to this SRS document. Testers may read this document at their own preference and are not required to read this before testing the software application. Marketing teams are assumed to read the SRS document before creating any marketing strategy for this software application product.

1.3 Intended Use

This document should be used when clarifying and specifying the requirements of Bookshelf, especially for people and developers who were not part of the initial developments of the application. Developers can also use this document when looking over and tracking the details implemented to this system.

1.4 Scope

Bookshelf is a software application that can be accessed through the web. The objective of Bookshelf is to help people to keep track of their reading habits by giving them an application that can let them do this efficiently. Another intended use of this software application is to provide with and allow users to write their opinions and reviews of the books they've read. This software application also provides book summaries, gives book suggestions based on what the user has read, and informs users of the most read books and authors of the Bookshelf community monthly. Users can have access to all these features by creating a Bookshelf account while anonymous users will still be able to read content found on the application, but will not be allowed to post anything. The ultimate goal of Bookshelf is to compete with other similar websites, such as Goodreads, by having an average of 45 million visitors per month on average.

1.5 Definitions and Acronyms

Terms	Definition
-------	------------

Anonymous user	A user that has not made an account on Bookshelf
Database	The collection of information used by this software application; the database provided by Django
User	A person who uses Bookshelf with an account

II. Overall Description

2.1 User Needs

The primary users of Bookshelf are people who are looking for an online software application that can help them record the books they've read while keeping an online profile to interact with other users. This can be anyone who has internet connection and is familiar with using online software applications. We can assume that whoever visits the Bookshelf web application are either looking for an alternative to other popular book review websites, want an application that can track their reading goals, or are looking at reviews of a certain book and have come across Bookshelf.

The following specifications are what a user would want and need from the application:

- The web application should have clear and organized content
- The web application should allow the user to create a profile
- There should be a visible log out button on all the web app's pages
- It should allow a logged in user to post a book review
- It should be secure enough that a user's profile cannot be easily hacked
- The services and features of the web app should always be available and should perform consistently
- The web app should load quickly
- The design should be easy on the eyes and visually pleasing
- The content should be easily digestible and readable

2.2 Assumptions and Dependencies

The state and requirements of this system are based on a Windows framework, so it cannot be assumed that the requirements will be the same on a Mac or Linux framework. It is also assumed that the web application will only be accessed through a computer and not through a smartphone, which will not accurately display the intended design of Bookshelf. This web application is using Django as its framework and database, so the software application's operation and performance may be dependent on the version of Django available.

III. System Features and Requirements

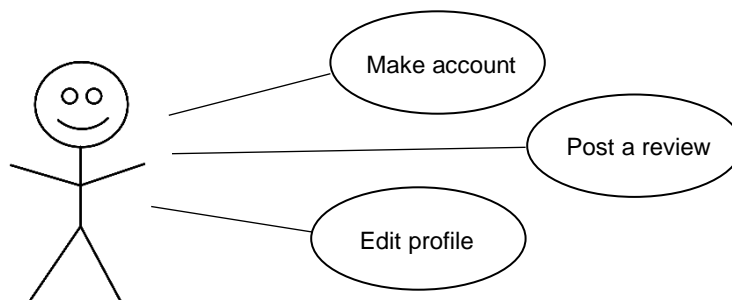
3.1 Functional Requirements

The following are the functional requirements that the developers have implemented that will allow users to accomplish the tasks they want to do in the system:

1. The user should provide their email and password when logging in.
2. The system automatically creates a profile for a user when they register for an account.
3. When the user does not move to a different page or doesn't do a task for 5 minutes, they will be automatically logged out due to session expiry.
4. The log out button available in every page of the website will allow the user to easily log out of their account at any time.
5. The 'post a review' button will allow the user to create and share their opinion and thoughts about a certain book.
6. The 'edit profile' button allows the user to update or change the information on their profile.

3.1.1 User Use Cases

The following are the use cases of the user:



Use case 1: Make account

Description: The user creates an account by filling up the sign-up form and submitting their information.

Step-by-Step Description

1. The user goes to the 'My Profile' page
2. The user fills up the information on the signup form
3. The users clicks on the 'submit' button

Use case 2: Post a review

Description: The user creates a book review post.

Step-by-Step Description

1. The user logs in to their account
2. The user goes to the 'Book Reviews' page
3. The user clicks on the 'Post a Review' button
4. The user fills up the required content
5. The user clicks the 'post' button

Use case 3: Edit Profile

Description: The user edits the information on their profile

Step-by-Step Description

1. The user logs in to their account
2. The user goes to the 'My Profile' page
3. The user clicks on the 'Edit Profile' button
4. The user can now change any information they want
5. The user clicks on the 'update' button

3.2 External Interface Requirements

The following are user interfaces that the software application has included:

1. The standard font style for the navigation bar is Merriweather, with a font size of 25 pixels
2. The standard font size of the content of the software application is either 25 pixels or 30 pixels, depending if the text is a header or not.
3. The fonts used within the software application should only be Merriweather, Nunito, Nanum Myeongjo, Georgia, or Roboto Slab.
4. The colors used for the software application should only be the following (in hex): 562c2c, f2542d, f5dfbb, 0e9594, 127475, any any other color that is similar to these colors.
5. An API should be used to convert text to speech for the visually impaired.

This software application does not interface with any other hardware or software interfaces.

The following are requirements for communication functions that the software application will need:

1. E-mail
2. Web browser

Since Bookshelf was built using the Django framework, its database is dependent on Django. To check whether the credentials entered by the user are correct when logging in (username and password), this data is authenticated by Django's `authenticate()` function, which checks the database for any user objects with the exact same data. If there exists a user, then they will be logged in but if ever the user enters the wrong username or password, they will not be logged in and will have to try again. When an anonymous user makes an account, Django also checks whether or not the username is already taken by getting a user object from the database with the same username. If there is no user object, then the registration is successful, otherwise, the anonymous user will have to choose a different username.

3.3 System Features

The following are the system features:

- The system will register an anonymous user if they provide valid credentials
- The system will log in the user when they provided the correct username and password pair.
- All users are able to view the book reviews and book summaries provided on the web application.
- Only registered users will be allowed to post a book review.
- Registered users are able to edit their profile.

3.4 Nonfunctional Requirements

3.4.1 Performance

Each page of software application should load within 3 seconds and each button should be able to execute its function within 200 to 300 milliseconds. Functions and features of Bookshelf should work at all times, unless there are technical difficulties with servers.

3.4.2 Security

The software application should keep the users' information safe and provide encryption to keep hackers from stealing information. An admin user should regularly check the state of the database for any attempts of hacking or breaching. The software should always be protected from unauthorized access and malware attacks.

3.4.3 Capacity

The software application should be able to handle at least 100,000 users at a time while still performing satisfactorily.

3.4.4 Manageability

The system should keep the information of the user for as long as they visit the website every year. If a user has not had any activity for a year, the software should send the user an email informing them that their account and information will be deleted unless they log in with the next 30 days.