Software Requirements Specification

Version 1.6

<<Annotated Version>>

June 20, 2020

Book Store System

Hazar Al-Abdullah

Lilas Al-Shwaity

Aya Shullar

Wessam Al-Mahmoud

Submitted in full fulfillment

Of the requirements of

Software Engineering course

<<Any comments inside double brackets such as these are *not* part of this SRS but are comments upon this SRS example to help the reader understand the point being made.

Refer to the SRS Template for details on the purpose and rules for each section of this document. >>

# Table of Contents

[Table of Contents i](#_Toc43583530)

[List of Figures ii](#_Toc43583531)

[1.0. Introduction 3](#_Toc43583532)

[1.1. Purpose 3](#_Toc43583533)

[1.2. Scope of Project 3](#_Toc43583534)

[1.3. Glossary 4](#_Toc43583535)

[1.4. References 4](#_Toc43583536)

[1.5. Overview of Document 4](#_Toc43583537)

[2.0. Overall Description 5](#_Toc43583538)

[2.1 System Environment 5](#_Toc43583539)

[2.2 Functional Requirements Specification 6](#_Toc43583540)

[2.2.1 Use case: View all books 6](#_Toc43583541)

[2.2.2 Use case: Search for a book 7](#_Toc43583542)

[2.2.3 Use case: Add a book 8](#_Toc43583543)

[2.2.4 Use case: Update a book 9](#_Toc43583544)

[**Xref:** Section 3.2.4, Update a book 9](#_Toc43583545)

[2.2.5 Use case: Delete a book 10](#_Toc43583546)

[Xref: Section 3.2.5, Delete a book 10](#_Toc43583547)

[2.3 User Characteristics 11](#_Toc43583548)

[2.4 Non-Functional Requirements 11](#_Toc43583549)

[3.0. Requirements Specification 12](#_Toc43583550)

[3.1 External Interface Requirements 12](#_Toc43583551)

[3.2 Functional Requirements 13](#_Toc43583552)

[3.2.1 View all books 13](#_Toc43583553)

[3.2.2 Search for a book 13](#_Toc43583554)

[3.2.3 Add a book 14](#_Toc43583555)

[3.2.4 Update a book 14](#_Toc43583556)

[3.2.5 Delete a book 14](#_Toc43583557)

[3.3 Detailed Non-Functional Requirements 15](#_Toc43583558)

[3.3.1 Logical Structure of the Data 15](#_Toc43583559)

[3.3.2 Security 17](#_Toc43583560)

# 

# List of Figures

[Figure 1 - System Environment 4](#_Toc43583497)

[Figure 4 - Logical Structure of the Article Manager Data 14](#_Toc43583498)

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the Book Store System. It will explain the purpose and features of the system, what the system will do, the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system and will be proposed to Mrs. Nahla Saad Eldeen.

## 1.2. Scope of Project

This software system will be a Book Store System for a local author. This system will be designed to maximize the author’s productivity by providing tools to assist in automating the books publishing process. By maximizing the author’s work efficiency and production the system will meet the author’s needs while remaining easy to understand and use.

More specifically, this system is designed to allow an author to manage and publish books to a public website. The system also contains a relational database containing a list of books.

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Database | Collection of all the information monitored by this system. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Author. |

## 1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# 2.0. Overall Description

## 2.1 System Environment

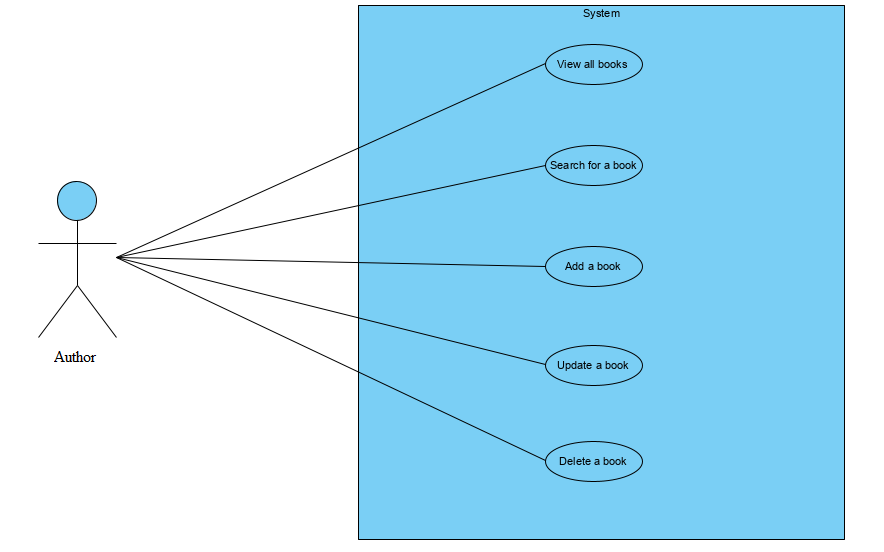


Figure 1 - System Environment

The Book Store System has one active actor.

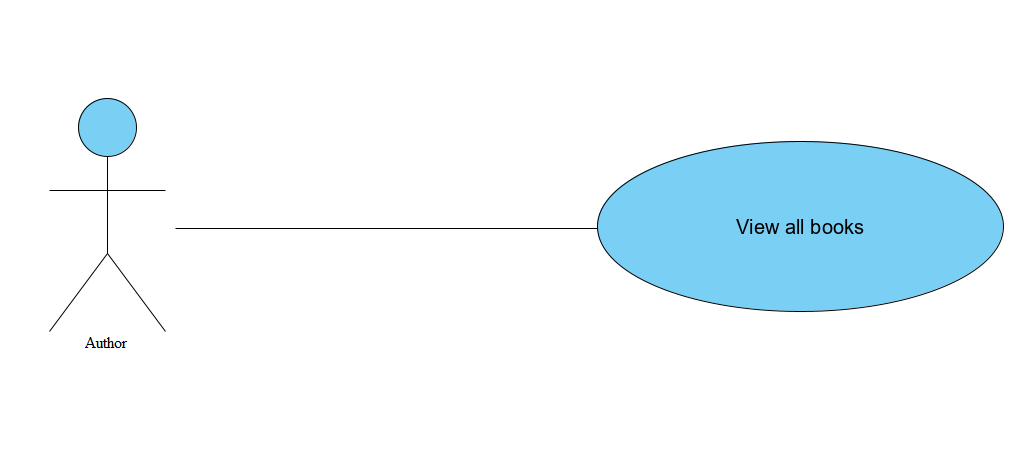
The Author accesses the website through the Internet.

## 2.2 Functional Requirements Specification

This section outlines the use cases for the author.

### 2.2.1 Use case: View all books

**Diagram:**



**Brief Description**

The Author accesses the website, the system automatically shows all the books in the database.

**Initial Step-By-Step Description**

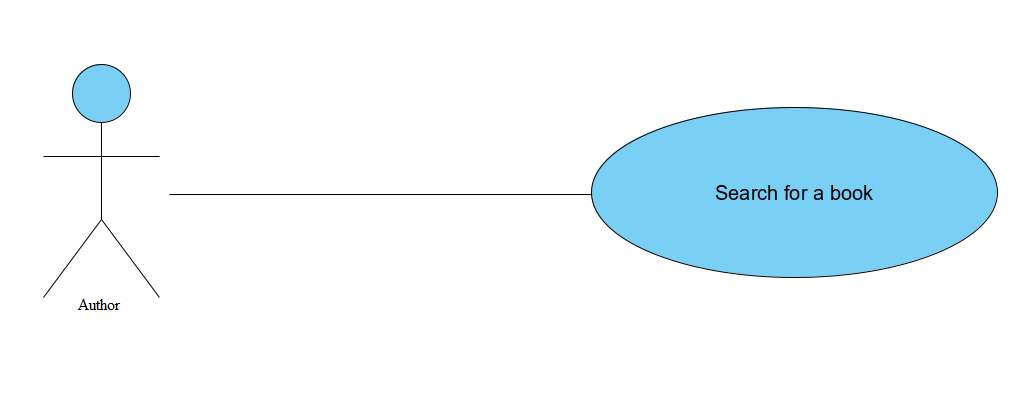
Before this use case can be initiated, the Author has already accessed the website.

1. The Author access the website.
2. The System then fetches all the books in the database and shows it to the author.

**Xref:** Section 3.2.1, View all books

### 2.2.2 Use case: Search for a book

**Diagram:**



**Brief Description**

The author searches for a certain book in the website.

**Initial Step-By-Step Description**

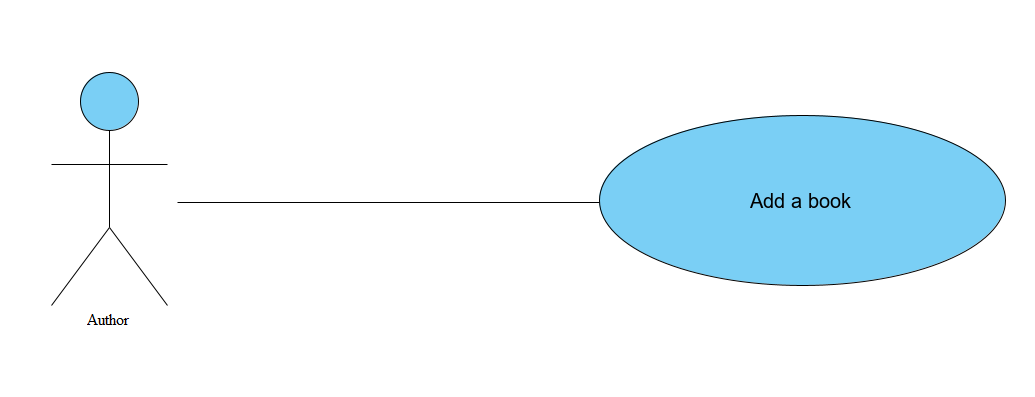
Before this use case can be initiated, the Author has already connected to the website.

1. The Author enters the book’s title or tag or keywords in the search bar.
2. The System search’s the database for a matching book.
3. The System displays a list of books related to the authors input.

**Xref:** Section 3.2.2, Search for a book

### 2.2.3 Use case: Add a book

**Diagram:**

****

**Brief Description**

The Author adds a new book to the website.

**Initial Step-By-Step Description**

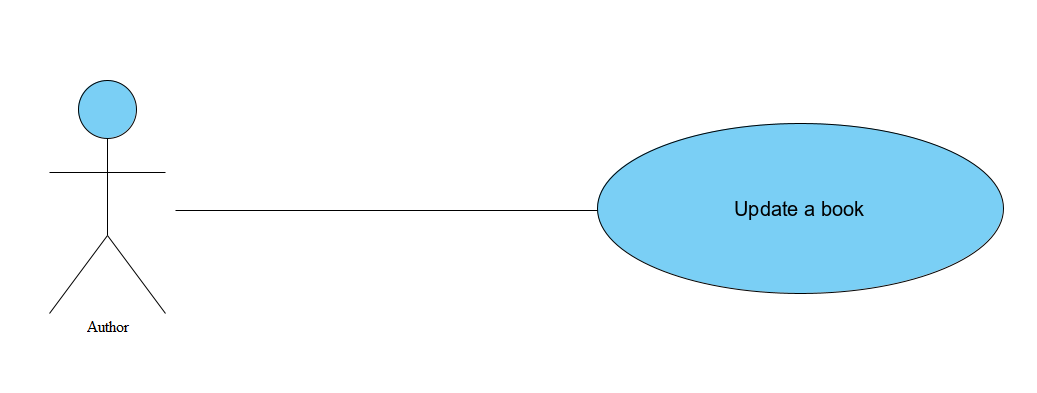
Before this use case can be initiated, the Author has already connected to the website.

1. The Author chooses to add a new book.
2. The System directs the Author to the addition page.
3. The Author selects the book to be uploaded.
4. The System adds the book to the database.

**Xref:** Section 3.2.3, Add a book

### 2.2.4 Use case: Update a book

**Diagram:**

****

**Brief Description**

The Author updates an existing book’s title, tag, content or keywords in the database.

**Initial Step-By-Step Description**

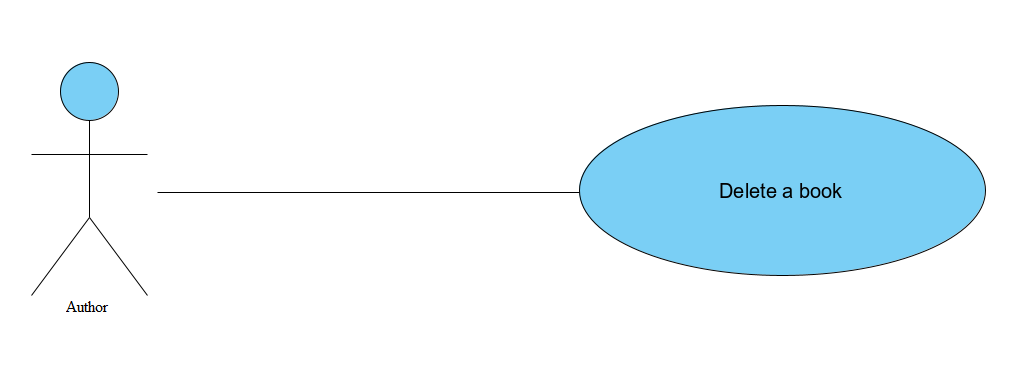
Before this use case can be initiated, the Author has already accessed the website.

1. The Author clicks the update a book button.
2. The System directs the Author to the updating page.
3. The Author selects the book to be updated.
4. The System opens the online editor so the Author can update a book’s title, tag, content or keywords.
5. The System saves the changes made into the database.

### **Xref:** Section 3.2.4, Update a book

#### 2.2.5 Use case: Delete a book

**Diagram:**

****

**Brief Description**

The Author permanently deletes a book from the database.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Author has already accessed the website.

1. The Author clicks the delete a book button.
2. The System directs the Author to the deletion page.
3. The Author selects a book to be deleted.
4. The System deletes the selected book from the database forever.

#### Xref: Section 3.2.5, Delete a book

## 2.3 User Characteristics

The Reader is expected to be Internet literate and be able to use a search engine. The main screen of the Online Journal Website will have the search function and a link to “Author/Reviewer Information.”

The Author and Reviewer are expected to be Internet literate and to be able to use email with attachments.

The Editor is expected to be Windows literate and to be able to use button, pull-down menus, and similar tools.

The detailed look of these pages is discussed in section 3.2 below.

## 2.4 Non-Functional Requirements

The Online Journal will be on a server with high speed Internet capability. The physical machine to be used will be determined by the Historical Society. The software developed here assumes the use of a tool such as Tomcat for connection between the Web pages and the database. The speed of the Reader’s connection will depend on the hardware used rather than characteristics of this system.

The Article Manager will run on the editor’s PC and will contain an Access database. Access is already installed on this computer and is a Windows operating system.

# 3.0. Requirements Specification

## 3.1 External Interface Requirements

The only link to an external system is the link to the Historical Society (HS) Database to verify the membership of a Reviewer. The Editor believes that a society member is much more likely to be an effective reviewer and has imposed a membership requirement for a Reviewer. The HS Database fields of interest to the Web Publishing Systems are member’s name, membership (ID) number, and email address (an optional field for the HS Database).

The *Assign Reviewer* use case sends the Reviewer ID to the HS Database and a Boolean is returned denoting membership status. The *Update Reviewer* use case requests a list of member names, membership numbers and (optional) email addresses when adding a new Reviewer. It returns a Boolean for membership status when updating a Reviewer.

## 3.2 Functional Requirements

The Logical Structure of the Data is contained in Section 3.3.1.

### 3.2.1 View all books

|  |  |
| --- | --- |
| **Use Case Name** | View all books |
| **XRef** | Section 2.2.1, View all books |
| **Trigger** | The Author assesses the website |
| **Precondition** | The Author has added books to the database. |
| **Basic Path** | 1. The Author access the website. 2. The System then fetches all the books in the database and shows it to the author. |
| **Alternative Paths** | None |
| **Postcondition** | All of the books in the database are shown to the author. |
| **Exception Paths** | None |
| **Other** | None |

### 3.2.2 Search for a book

|  |  |
| --- | --- |
| **Use Case Name** | Search for a book |
| **XRef** | Section 2.2.2, Search for a book |
| **Trigger** | The author inputs a tag or a keyword or any text in the search bar and hits search. |
| **Precondition** | The author has already accessed the website. |
| **Basic Path** | 1. The Author enters the book’s title or tag or keywords in the search bar. 2. The System search’s the database for a matching book. |
| **Alternative Paths** | None |
| **Postcondition** | A list of books related to the author’s input is shown |
| **Exception Paths** | None |
| **Other** | None |

### 3.2.3 Add a book

|  |  |
| --- | --- |
| **Use Case Name** | Add a book |
| **XRef** | Section 2.2.3, Add a book |
| **Trigger** | The Author clicks the add new book button. |
| **Precondition** | The Author has accessed the website. |
| **Basic Path** | 1. The Author chooses to add a new book. 2. The System directs the Author to the addition page. 3. The Author selects the book to be uploaded. |
| **Alternative Paths** | None |
| **Postcondition** | The System adds the book to the database. |
| **Exception Paths** | None |
| **Other** | None |

### 3.2.4 Update a book

|  |  |
| --- | --- |
| **Use Case Name** | Update a book |
| **XRef** | Section 2.2.4, Update a book |
| **Trigger** | The Author clicks the update a book button. |
| **Precondition** | The Author has accessed the website. |
| **Basic Path** | 1. The Author clicks the update a book button. 2. The System directs the Author to the updating page. 3. The Author selects the book to be updated. 4. The System opens the online editor so the Author can update a book’s title, tag, content or keywords. |
| **Alternative Paths** | None |
| **Postcondition** | The System saves the changes made into the database. |
| **Exception Paths** | None |
| **Other** | None |

### 3.2.5 Delete a book

|  |  |
| --- | --- |
| **Use Case Name** | Delete a book |
| **XRef** | Sec 2.2.5 Delete a book |
| **Trigger** | The Author clicks the delete a book button. |
| **Precondition** | The Author has accessed the website. |
| **Basic Path** | 1. The Author clicks the delete a book button. 2. The System directs the Author to the deletion page. 3. The Author selects a book to be deleted. |
| **Alternative Paths** | None |
| **Postcondition** | The System deletes the selected book from the database forever |
| **Exception Paths** | None |
| **Other** | None |

## 3.3 Detailed Non-Functional Requirements

### 3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the internal Article Manager database is given below.

Review

Reviewer

Article

Author

writes

sent to

writes

has

Figure 4 - Logical Structure of the Article Manager Data

The data descriptions of each of these data entities is as follows:

**Author Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of principle author |  |
| Email Address | Text | Internet address |  |
| Article | Pointer | Article entity | May be several |

**Reviewer Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of principle author |  |
| ID | Integer | ID number of Historical Society member | Used as key in Historical Society Database |
| Email Address | Text | Internet address |  |
| Article | Pointer | Article entity of | May be several |
| Num Review | Integer | Review entity | Number of not returned reviews |
| History | Text | Comments on past performance |  |
| Specialty | Category | Area of expertise | May be several |

**Review Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Article | Pointer | Article entity |  |
| Reviewer | Pointer | Reviewer entity | Single reviewer |
| Date Sent | Date | Date sent to reviewer |  |
| Returned | Date | Date returned; null if not returned |  |
| Contents | Text | Text of review |  |

**Article Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Article |  |
| Author | Pointer | Author entity | Name of principle author |
| Other Authors | Text | Other authors is any; else null | Not a pointer to an Author entity |
| Reviewer | Pointer | Reviewer entity | Will be several |
| Review | Pointer | Review entity | Set up when reviewer is set up |
| Contents | Text | Body of article | Contains Abstract as first paragraph. |
| Category | Text | Area of content | May be several |
| Accepted | Boolean | Article has been accepted for publication | Needs Copyright form returned |
| Copyright | Boolean | Copyright form has been returned | Not relevant unless Accepted is True. |
| Published | Boolean | Sent to Online Journal | Not relevant unless Accepted is True. Article is no longer active and does not appear in status checks. |

The Logical Structure of the data to be stored in the Online Journal database on the server is as follows:

**Published Article Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Article |  |
| Author | Text | Name of one Author | May be several |
| Abstract | Text | Abstract of article | Used for keyword search |
| Content | Text | Body of article |  |
| Category | Text | Area of content | May be several |

### 3.3.2 Security

The server on which the Online Journal resides will have its own security to prevent unauthorized *write*/*delete* access. There is no restriction on *read* access. The use of email by an Author or Reviewer is on the client systems and thus is external to the system.

The PC on which the Article Manager resides will have its own security. Only the Editor will have physical access to the machine and the program on it. There is no special protection built into this system other than to provide the editor with *write* access to the Online Journal to publish an article.