Software Requirements Specification

for

Final Project

**Version 3.0 approved**

**Prepared by Aimee Valladares,**

**Ricardo Zuniga, and Andrew Wilson**

**Webio**

**23 April 2021**

# Table of Contents

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 2

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 4**

3.1 User Interfaces 4

3.2 Hardware Interfaces 4

3.3 Software Interfaces 4

3.4 Communications Interfaces 5

**4. System Features 5**

4.1 Product Management (System Feature 1) 6

4.2 Security Measures (System Feature 2) 6

4.3 User Interface(System Feature 3) 6

4.4 Database(System Feature 4) 7

**5. Other Nonfunctional Requirements 7**

5.1 Performance Requirements 7

5.2 Safety Requirements 8

5.3 Security Requirements 8

5.4 Software Quality Attributes 8

**6. Other Requirements 9**

**7. System Requirements Chart 10**

**Appendix A: Glossary 12**

**Appendix B: Analysis Models 12**

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Aimee Valladares, Ricardo Zuniga, Andrew Wilson | 4/4/2021 | Introduction Report for SRS document | 1.0 |
| Aimee Valladares,  Ricardo Zuniga | 4/13/2021 | Overall Description for SRS document - First Draft | 2.0 |
| Aimee Valladares,  Ricardo Zuniga,  Andrew Wilson | 4/14/2021 | Overall Description for SRS document - Final Draft | 2.5 |
| Aimee Valladares | 4/21/2021 | Final SRS Report - First Draft | 3.0 |
| Aimee Valladares,  Ricardo Zuniga,  Andrew Wilson | 4/22/2021 | Final SRS Report - Final Draft | 3.5 |

# **Introduction**

## **Purpose**

The product is software that helps manage the product inventory of Reader’s Nook in real time after every change in the organization's product stock. The SRS document will describe the whole system, focusing on the key subsystems for the software that affect the functionality and usability of the product. Specifically, this document will focus on the product scope, purpose, and description. This document focuses on the version 1.0 SRS.

## **Document Conventions**

The document will be using Times New Roman font with the paragraphs being font size 12, subheaders being font size 14, and all major headers being font size 18. Bold and italic stylization is used throughout the document for emphasis or highlighting important terminology and definitions that are significant. The document will be reviewed at the very least once every two weeks in case there are any changes to the requirements. Each of the requirement statements will have a designated priority level as well indicate if the requirement is related to another requirement statement for functionality.

## **Intended Audience and Reading Suggestions**

The document is intended for the developers, project managers, and anyone else that is involved with the development of the system including Aimee Valladares, Andrew Wilson, and Ricardo Zuniga. The rest of the document will contain the overall description of the product and its requirements.

This document will affect further documentation for the software product including project design documentation testing documents, training documents, and user manual. This rest of this SRS document contains the five headings that breaks down and organizes the product scope, features, interfaces, and systems.

## **Product Scope**

The software being specified is the programming, database, and the user interface for the product. The software’s purpose is to create a system that manages the product inventory of the Reader’s Nook company after every transaction. The software must be compatible with the current company software, namely the website and software that handles the day-to-day transactions that scans the product and The goal of the software is to apply in-depth software requirement engineering processes to accurately collect the requirements.

The software will have a database that contains key information regarding the product, user interface that is easy-to-understand for various company employees, and performance that has the product inventory update after each transaction and accessibility. For more information regarding the product scope refer to the Phase II - Project Elicitation document.

## **References**

The reference format will be utilizing MLA 8 format.

* Valladares, Aimee, Zuniga, Ricardo, and Wilson, Andrew. *Phase II - Project Elicitation*. CANVAS, 26 Mar. 2021.
* “TechTerms.” *The Tech Terms Computer Dictionary*, techterms.com/.

# **Overall Description**

## **Product Perspective**

This software is an addition to the organization’s existing system, specifically the organization’s current website. The software is meant to be a self-contained product that is only meant to add greater functionality to the organization’s system. The software must be compatible with the existing system without much modification to the organizations system like the company website.

## **Product Functions**

1. The product will have an easy-to-use and easy-to-understand user interface (UI).
2. The product must help manage the organization’s inventory by updating the system after every change in the company’s product stock.
3. The update must occur in real time. Specifically, the system must update after *every* transaction.
4. The product must utilize a database for storing and accessing information.
5. The product must only be accessible to employees.
6. The product must be compatible with the current company software.

## **User Classes and Characteristics**

The primary users that will use this product are company employees. However, the users can be separated into two categories: employees with little to no technical expertise (customer representative, manager, financial advisor, etc.) and employees with more technical knowledge (IT specialist, developers, etc.). Both classes will have a login though employees with greater technical expertise will have greater security access and clearance. To help ensure that the employees with limited technical knowledge can still use the product because sufficient documentation will be provided namely a user manual and training. The important difference between the user classes is that employees with limited technical knowledge primarily need to understand *how to use* the product whereas employees with technical expertise *also* understand *how the product works.*

## **Operating Environment**

The program must be able to run on any desktop or laptop running a Windows 10. The program must also be able to run alongside any background applications that run on the computer on a regular basis (e.g. antivirus software).

## **Design and Implementation Constraints**

One design and implementation constraint on the product is the programming languages are used during the software development. For the front-end development or the product the programming languages, HTML, CSS, and JavaScript will be used. The database for the product will be created using SQL or EXCEL. The product interface and coding must be compatible with the company’s current website and software.

Another design and implementation constraint on the product is the security and performance of the software. The product must utilize security measures so that it is only accessible to employees as well as protect any information stored within the product. The product must have enough memory and processing power to immediately update the system after every transaction.

## **User Documentation**

The user documentation that will be delivered is training documents, testing documents, and a user manual. Training documents will be required to allow the employees with limited technical knowledge to use the software. The user manual will be documentation that will be provided as reference in case any situations occur.

## A**ssumptions and Dependencies**

Software must work with the website that the company is currently using. Information on the products being sold will be stored in a database that will be accessed from different locations, such as stores and warehouses, by employees. Software is a high priority for the company, this could cause some risk in occuring once software is deployed. It must be noted that errors may occur in the development of the product due to utilizing the assumptions and dependencies incorrectly, changing, or not being shared.

# **External Interface Requirements**

## **User Interfaces**

The most important interface between the user and the program is the **graphical user interface (GUI)**. The GUI should be easily readable and unintrusive. The GUI must first have a password login portal that has the employee either enter their information and password. For this access portal we must at least include a forgot password link and a login button. The GUI will have a minimum of three buttons: View, Submit, Save. All of the buttons would follow the same layout of having a black border with bolded font. The GUI would have a similar layout to a form as it would have text boxes to either automatically and/or manually enter the product (book) information. Finally, the interface should have two alert messages, one that states the information has been successfully entered and the update has occurred and another that states there was an error in reading the information.

## **Hardware Interfaces**

The necessary hardware for the GUI and software is a monitor, keyboard, touch screen, and scanner (for user inputs) for all kinds of setups including more portable ones like a tablet. The hardware devices needed. Though the product does have security and performance requirements it is not necessary to extensively change the current hardware or install new hardware. This is because the company already has enough **RAM**, memory, and processing power for the company website and software that can be used concurrently by multiple users at different locations like the company store and warehouse. Any more information detailing the user interface design will be recorded in future documentation.

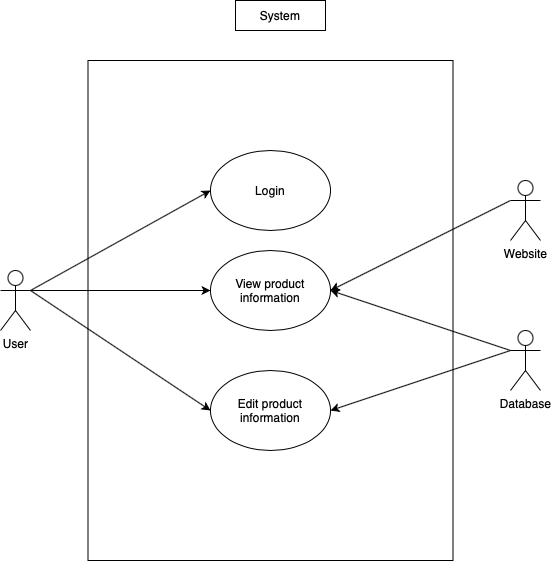
## **Software Interfaces**

The software will be using a database to store information on the books that the company is selling. It will have information on the books and if they are in stock either at the store or the warehouse. The software system will also be connected to the company’s website and will update the amount of stock left in real time. The software will also have to be compatible to current software that the company is already using.

## **Communications Interfaces**

The product does essentially utilize an electronic form that will send information to the database which can be accessed in various locations like in the company store or in the warehouse. The product will also employ communication security specifically encryption so the data can be safely accessed in various locations as well as protect employee login information. The product should be able to follow the web browser and network server protocols that are already in place by the organization’s current software.

# **System Use Cases**



This here is the overall use case diagram.

## Product Management

1. **Description and Priority** - The system will keep track of all the books and its information such as the ISBN number and if the books are in stock by using the database. The priority is *high* since this is an important part of the system.
2. **Stimulus/Response Sequences** - This feature will retrieve information of the books that are in the database and display it to the user using the UI.
3. **Functional Requirements** -

REQ-1: The software must be able to retrieve information from the database

REQ-2: The software must be able to display the information of the products to the user using the UI.

REQ-3: Must automatically update the information for the company product stock after sufficient information is entered.

## Security Measures

1. **Description and Priority** - The product should have security measures that ensure the data stored and accessed is safe. The priority for this system feature is *medium* because while the security measures ensure safety and protection, this system feature does not greatly impact the functionality of the software product as a whole.
2. **Stimulus/Responses Sequences** - The system will implement password protection and encryption so that only employees and the development team can access the software.
3. **Functional Requirements** -

REQ-1: Must utilize a login portal to access the product.

REQ-2: The login portal must be password protected and recognize each unique password belonging to each employee.

REQ-3: Must encrypt the data of the database and employee login to ensure that the data is safe.

REQ-4: The security measures must be consistent throughout the entire network as well as at various locations like the company store and the warehouse.

## User Interface

1. **Description and Priority** - The priority for this system feature is *medium* because while the product should be easy-to-use and understandable but it does not employ extensive graphics and complex functionality.
2. **Stimulus/Response Sequences** - The system will display information retrieved from the database and display it in a more readable manner to the users.
3. **Functional Requirements** -

REQ-1: The user interface must be easy-to-use and easy-to-understand with a simple layout and clear instructions.

REQ-2: Must display the form that has the user submit the product information.

REQ-3: The user interface will include multiple web pages. Each web page will have a different layout corresponding with different functionality.

REQ-4: The GUI must at least include the login portal, form, and database.

REQ-5: Must include buttons to enable the user to enter and access information.

REQ-6: Must display the information of the product that the user has searched for.

## Database

1. **Description and Priority** – The product should have a database that stores and accesses all of the company’s product (book) information. The priority for this system feature is *high* due to it being an integral part of the product.
2. **Stimulus/Response Sequences** – The system will allow the user to store their information as well as handle the update in the company’s stock.
3. **Functional Requirements**

REQ-1: The database should allow the user to enter certain queries that can add, modify, view and/or remove information stored in it.

REQ-2: The database should include the following product information: book’s 9-digit ISBN, the product quantity, an indicator of whether an item is in stock (at the store), and an indicator of whether an item is in store and/or at the warehouse. The database must also contain key information regarding the book, specifically the title, author, and publisher.

REQ-3: The user interface must be able to connect to the database.

REQ-4: The database must accurately reflect the company’s stock, both in-store as well as in the warehouse.

# **Other Nonfunctional Requirements**

## **Performance Requirements**

The performance requirements for the product are meant to ensure that the product can update every change in the system and operate as quickly as the stakeholders require. One performance requirement is that the system can handle multiple users. Specifically, the system must be able to handle at the very least 50 concurrent users.

Another performance requirement is that the system handles modifications in a timely manner. For instance, the system must respond to user interactions on a page within 0.1 seconds. It must load each new page within 1 second. The system must load and be usable within 10 seconds of user login. Finally, the system should probably have consistent response times.

## **Safety Requirements**

Due to the product handling important information about the company’s stocks and products, we must implement the safety requirement that ensures the data is protected. Notably, in the case of a system failure, a backup of all of the data on the system will be created. A backup will be created every 3 hours and uploaded to a hard disk and to the cloud. Each backup will be kept for a month. When the system is repaired, the data in one of the backups can be reuploaded into the repaired system.

## **Security Requirements**

One security requirement is that the software will be only accessible via user authentication. Specifically, each user of the system will have a username and password that is stored in the database. Additionally, each password in the database will be encrypted, salted, and hashed. Another security requirement is that the software will employ encryption when accessing and sending information across the network.

## **Software Quality Attributes**

The key software quality attributes that are important to this product and ensuring that it is up to business-ready status is correctness, availability, maintainability, interoperability, and reusability.

**Correctness:**

* The software must keep correct data about the inventory of products.
* The software must only allow the user to enter the system if they enter a correct username **AND** password.
* The backups must contain correct data about the inventory of products.

**Availability:**

* The software must be usable during the times when the store is open.
* The software must be able to operate continuously for at least 6 months without maintenance (outside of abnormal circumstances).

**Maintainability:**

* The software must be able to be repaired or upgraded easily without affecting the data stored in the software.

**Interoperability:**

* Whenever a user enters a username and password into the user interface, it must check if the username and password match any account in the database.
* Whenever a transaction occurs, it must update the data in the database to maintain the correct stock of each item in the store.

**Reusability:**

* Certain components of this software could be reused in future projects.

**Robustness:**

* When an item that is not sold at the store is scanned, the system must give the user a message stating that the item is not sold at the store.

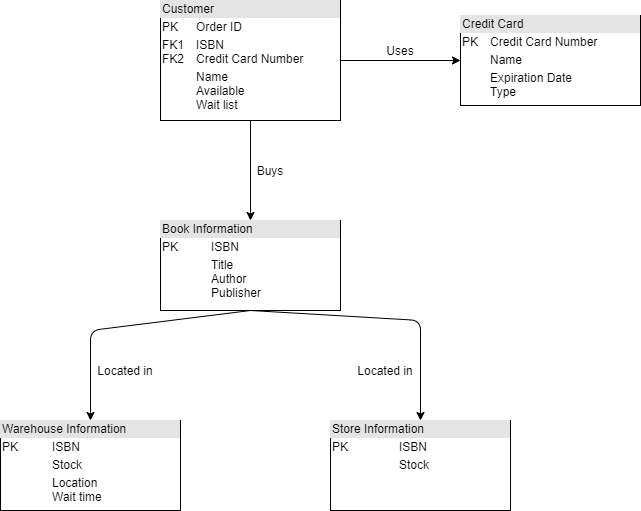
**Reliability:**

* The system must consistently send information about the inventory of products to the database after each transaction.

# **Other Requirements**

The database that the system is going to be using will be using SQL or EXCEL. Must be able to handle changes made through the system in real time as well as having multiple people access the database at the same time. The database must also be able to handle queries based on the database relational diagram shown below.

The database would essentially contain at least 5 table components: Customer, Credit Card, Book Information, Warehouse Information, and Store Information. *Customer* would contain information about the customer and also display any possible issue the customer may have had. For instance, the table should illustrate if the book was available and if it was not then we're waitlisted. The *Credit Card* table is meant to illustrate the form of payment. The *Book Information* table should illustrate key, identifiable information about the book. The *Warehouse Information* table displays whether or not a book is in stock in the warehouse. Finally, the *Store Information* table displays if the book is in stock.



# **System Requirements Chart**

The System Requirement Chart illustrates all of the functional and non-functional requirements for the software product.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Priority | Type | Source | In Use Case(s) | Description |
| PM\_REQ-1 | High | Functional | Customer | Product Management | 4.1.3 |
| PM\_REQ-2 | High | Functional | Customer | Product Management | 4.1.3 |
| PM\_REQ-3 | High | Functional | Customer | Product Management | 4.1.3 |
| SM\_REQ-1 | High | Functional | Customer | Security Measures | 4.2.3 |
| SM\_REQ-2 | High | Functional | Customer | Security Measures | 4.2.3 |
| SM\_REQ-3 | High | Functional | Team | Security Measures | 4.2.3 |
| SM\_REQ-4 | High | Functional | Team | Security Measures | 4.2.3 |
| UI\_REQ-1 | Med | Functional | Customer | User Interface | 4.3.3 |
| UI\_REQ-2 | Med | Functional | Customer | User Interface | 4.3.3 |
| UI\_REQ-3 | Med | Functional | Team | User Interface | 4.3.3 |
| UI\_REQ-4 | Med | Functional | Team | User Interface | 4.3.3 |
| UI\_REQ-5 | Med | Functional | Stakeholder | User Interface | 4.3.3 |
| UI\_REQ-6 | Med | Functional | Stakeholder | User Interface | 4.3.3 |
| DB\_REQ-1 | High | Functional | Team | Database | 4.4.3 |
| DB\_REQ-2 | High | Functional | Team | Database | 4.4.3 |
| DB\_REQ-3 | High | Functional | Team | Database | 4.4.3 |
| DB\_REQ-4 | High | Functional | Team | Database | 4.4.3 |
| Performance | High | Non-functional | Team | N/A | 5.1 |
| Safety | High | Non-functional | Team | N/A | 5.2 |
| Security | High | Non-functional | Team | N/A | 5.3 |
| Correctness | High | Non-functional | Team | N/A | 5.4 |
| Availability | High | Non-functional | Stakeholder | N/A | 5.4 |
| Maintainability | Med | Non-functional | Team | N/A | 5.4 |
| Interoperability | High | Non-functional | Team | N/A | 5.4 |
| Reusability | Low | Non-functional | Team | N/A | 5.4 |
| Robustness | High | Non-functional | Team | N/A | 5.4 |
| Reliability | High | Non-functional | Team | N/A | 5.4 |

**Appendix A: Glossary**

* **Database** - A collection of electronically stored data
* **User Interface** - Visual elements that allow users to control and interact with a software.
* **CSS** - Cascading Style Sheets, a style sheet language that is used to modify the appearance of a markup language such as HTML
* **HTML** - Hypertext Markup Language, a markup language that is used to structure web pages
* **SQL** - Structured Query Language, a language that is used to manage data in a DBMS.
* **DBMS** - Database Management System, software designed to manage and control data in a database.
* **ISBN** - International Standard Book Number, a number used to identify books.
* **RAM** - Random Access Memory, hardware component that determines the memory that online applications can use.

**Appendix B: Analysis Models**

Diagram 1: Entity-Relationship Diagram

