Bookstore Inventory System

Requirements Specifications

# Introduction

## Purpose

The purpose of this document is to define the requirements for the Book Inventory System (BIS) project. This project will demonstrate the basics of the Software Development Lifecycle in the context of a bookstore inventory management application. It will incorporate a list of functional requirements and a small number of use cases for this version of the project.

## Intended Audience

The intended audience of this document is the sole project developer and tester, David Wilson.

## Intended Use

The intended use of this document is to write specifications for and define use cases of the proposed book inventory system.

## Product Scope

The product will enable management of inventory and sales for a small bookstore with a single location. Its initial version will permit viewing, adding, and removing inventory; searching inventory for specific ISBNs, titles, and authors; setting store prices for inventory; making and tracking sales; and tracking customers who are also members of the store’s loyalty program. This product supports only physical inventory, as opposed to e-books. It also assumes all sales are made in person at the store’s single, physical location.

## Definitions and Acronyms

The following is a list of acronyms and definitions expected to be used as part of this product, broken down by the type of record they are used to describe within the system. Many of these terms also refer to datapoints that make up a specific type of record.

### Book Records

* **Author**: The writer of a given book. Some books have multiple authors. In these cases, authors will be listed alphabetically, with the first author given as the primary author.
* **Cover Price**: The value established for the book by its publisher, so-called because it is typically listed somewhere on the book’s cover or, in the case of hardcover, inside the book’s dust jacket.
* **Edition**: A single version of a given book with similar typesetting, cover art, and contents, usually generated from the same print run or a reprinting of a specific, prior print run. They are usually identified by number starting at “1” or as “first edition,” “second edition,” etc.
* **Format**: The nature of the book’s physical binding and materials given as either “paperback” or “hardcover.”
* **International Standard Book Number (ISBN)**: A unique identifying number assigned to a specific edition and format of a given book. This is the primary identifier for a book record.
* **Stock**: The number of copies of books matching a given ISBN that the store has in stock.
* **Store Price**: The value at which the book is sold inside the store.
* **Title**: The name given to a book by its author and shared by all editions of that book. If certain editions include a subtitle exclusive to that edition, it will not be considered part of the book’s title for the purposes of this system.

### Order Records

* **Contents**: A list of books sold in a given order. The content of an order is organized by ISBN. Each item is also associated with a quantity and a price per item based on the store price of the book at the time of the sale.
* **Employee ID**: The identifying number of the employee who made the sale.
* **Customer ID**: The identifying number of the customer purchasing the books in the sale, assuming they are a member of the store’s loyalty program. If the customer is not a member, this value is set to “0.”
* **Order Number**: A unique identifier assigned to a specific order.
* **Order**: Refers to a transaction in which the store sells a book to a customer.

### Customer Records

* **Customer Number**: A unique, ten-digit identifier assigned to a specific customer, starting from 0000000001.
* **Customer**: Customers may become members of the store’s loyalty program to receive newsletters and special offers. In exchange, the store is able to obtain more information about individual book sales for the purposes of analytics. An order may contain multiple books, but it can be associated with only one customer. Each record will include the member’s contact information, including personal name or business name, mailing address, phone number, and email address. Not all customers are members of this program.

### Employee Records

* **Employee Number**: A unique, ten-digit identifier assigned to a specific employee, starting from 0000000001.
* **Employee**: A member of store personnel responsible for making sales. Each employee record includes that employee’s name, mailing address, phone number, and email address.

# Overall Description

This project will create a new inventory and sales management program to assist independent, single location bookstores in tracking their stock, sales, and customers. It will enable the store to search for specific stock by ISBN, title, or author; make and track sales; and maintain records of individual employees and customers who are loyalty program members.

## User Needs

The intended end-user is the manager of a small bookstore. The program should be intuitive enough to use with minimal training. It should enable the user to search their stock for specific products, make sales, and add records for employees and customers.

## Assumptions and Dependencies

For the purposes of this product, we are assuming inexpensive modern technology. The target system would run Windows, but the requirements will be very low. We are assuming that all products are physical, so the only formats needed to track are hardcover and paperback. We are assuming that the user has a separate method of validating book information, as this product is not intended to validate details like specific ISBNs. We are assuming the user has separate software to track their costs, including costs for product, which are not addressed in this product.

# System Features and Requirements

## Functional Requirements

### FR1 – Manipulate a Record

ID: FR1

Title: Manipulate a Record

Description: A user updates the collection of Book, Customer, Employee, or Order records in one of the following ways:

* Add a new record, representing a new ISBN, customer, employee, or sale added to the existing collection of the same.
* Remove an existing record, representing an ISBN, customer, employee, or sale being removed from the existing collection of the same.
* Edit a feature of an existing record to correct a flaw in the entry or, in the case of a book, alter the quantity of books in stock without creating a new sale.

This FR holds dependencies only in those instances where the user must enter the key data member manually, in which case it references FR3 to determine if a record with that data member already exists.

Dependencies: FR3 – Search for a Record.

### FR2 – Print List of Records

ID: FR2

Title: Print List of Records

Description: A user prints a formatted list of all records from one of the following types: Books, Customers, Employees, or Orders. They are listed in ascending order based on the following fields:

* Books: ISBN
* Customers: Customer Number
* Employees: Employee Number
* Orders: Order Number

Dependencies: None.

### FR3 – Search for a Record

ID: FR3

Title: Search for a Record

Description: User searches for a Book, Customer, Employee, or Order record with one of a handful of specified fields for each. If more than one record matches the search, all applicable records are returned. The search fields are as follows:

* Books: ISBN, Title, Author
* Customer: Customer Number
* Employee: Employee Number
* Orders: Order Number

Dependencies: None.

## External Interface Requirements

### User Interface Requirements

The product should be usable through a clean, intuitive Graphical User Interface. Buttons should be clearly labeled by function.