

# Book Shop Automation Software

Software Requirements  
Specification



Indian Institute of Information Technology  
Ranchi

**Presented By:-**

**Suraj Kumar (2019UGCS027R)**

**Nitish Kumar (2019UGCS023R )**

**Mani Shankar Choudhary (2019UGCS006R )**

**Abhishek Kumar Kushwaha (2019UGCS035R)**

**Guided By:-**

**Dr. Jayadeep Pati**

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Scope . . . . .	2
1.3	Definitions,Acronyms,and Abbreviations . . . . .	2
1.4	References . . . . .	2
<b>2</b>	<b>overall Description</b>	<b>3</b>
2.1	Product Perspective . . . . .	3
2.2	product function . . . . .	3
2.2.1	Query for books: . . . . .	3
2.2.2	Request for book: . . . . .	3
2.2.3	Update stock and inventory: . . . . .	3
2.2.4	View request . . . . .	3
2.2.5	Generate sales receipt: . . . . .	3
2.2.6	Generate sales statistics: . . . . .	4
2.2.7	print the list of books to be bought depending on the inventory level: . . . . .	4
2.3	User characteristics . . . . .	4
2.3.1	Book . . . . .	4
2.3.2	Employee . . . . .	4
2.3.3	Sales Day . . . . .	4
2.3.4	Not In Collection . . . . .	4
2.3.5	Cart . . . . .	5
2.3.6	Main . . . . .	5
2.3.7	Other auxiliary classes : . . . . .	5
2.4	Constraints . . . . .	5
2.5	Assumption and Dependencies . . . . .	5
<b>3</b>	<b>Specific Requirements</b>	<b>5</b>
3.1	External Interfaces . . . . .	5
3.1.1	User interfaces . . . . .	5
3.1.2	Hardware Interfaces . . . . .	6
3.1.3	Software Interfaces . . . . .	6
3.1.4	Communications Interfaces . . . . .	6
3.2	Functions Requirements . . . . .	7
3.2.1	Check for availability of book: . . . . .	7
3.2.2	View request: . . . . .	7
3.2.3	Purchase Book: . . . . .	7
3.2.4	Update stock: . . . . .	8
3.2.5	Generate Sales Statistics: . . . . .	8
3.3	performance Requirements . . . . .	9
3.4	Logical Database requirements . . . . .	9
3.5	Design Constraints . . . . .	9
3.5.1	Menu driven . . . . .	9

3.5.2	Confirmation message . . . . .	9
3.5.3	Error Detection . . . . .	9
3.6	Software system quantity Attributes . . . . .	9
3.7	Object oriented Models . . . . .	9

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Book Shop Automation Software” (BAS) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications

## 1.2 Scope

In today’s world, there arises a need to get things done quickly and efficiently. Customers find it difficult at times to find a book from book shops close to them . At the same time, the book shop owner wouldn’t want to take a risk of losses due to stagnant stock. This issue is resolved by a neat mutually beneficial way by the software. It strives to be a fully independent entity with all necessary features to handle transactions and queries in a book shop. Also a need to automate tasks which need to be performed daily arises. In this case, the software reduces the risk and need of a third party to manually perform these mundane tasks.

## 1.3 Definitions,Acronyms,and Abbreviations

- BAS Book Shop Automation Software
- SRS Software Requirements Specifications
- ISBN International Standard Book Number is a unique 13 digit code for each book. Contains information related to Title, Publisher and Group etc. Different Editions will have different ISBN and ISBN for different copies of same book will be same.

## 1.4 References

- IEEE standard template for IEEE standard
- Fundamentals of Software Engineering by R.Mall

## **2 overall Description**

### **2.1 Product Perspective**

The BAS is an independently functioning system that automates manual tasks in a Book Shop like answering customer queries, billing, making sales statistics, planning for further orders etc. The system handles and stores all necessary records of books in a database. The user interface is simple for any user to understand when he/she uses it for the first time itself . The software also analyses data and presents it in a useful form for any important stakeholder like the owner and his business associates to take important decisions.

### **2.2 product function**

The book-shop automation system provides the following facilities and services:

#### **2.2.1 Query for books:**

The customer can query for availability of the book by entering its title or the author's name. On availability , the BAS shows the rack number and the number of copies in stock.

#### **2.2.2 Request for book:**

If the query for the book fails, then the customer has an option of request- ing for the book shop to order the book in question by entering the full details of the book (Name, Author's name, ISBN code,Publisher's name ). Depending on the number of requests, the Manager may decide to order the book.

#### **2.2.3 Update stock and inventory:**

When the customer confirms the book, the BAS updates the inventory. Also in case of new arrivals/defective pieces, the BAS updates accordingly.

#### **2.2.4 View request**

The Manager can view the number of requests for the books and decide on the course of action to be taken.

#### **2.2.5 Generate sales receipt:**

To complete the transaction, the BAS generates a sales receipt for endorsing the transaction.

### **2.2.6 Generate sales statistics:**

The BAS generates sales statistics based on transactions between any given period. This feature is used by the owner and any other authorised person. This also helps in finding the inventory level.

### **2.2.7 print the list of books to be bought depending on the inventory level:**

Every day the book shop owner would give a command for the BAS to print the books which have fallen below the threshold and the number of copies to be procured along with the full address of the stockist.

## **2.3 User characteristics**

Main class has the access to information as it contains aggregations of many class objects and is the most privileged in the sense of access. Cart is used very frequently in conjunction with Sales Clerk to buy books and print receipts. This is the most general function of the software.

### **2.3.1 Book**

This class is central to this project. It stores those fields that characterize a Book object such as it's name, author's name, price, publisher, which rack it is kept on and so on. This implies that there can only be one book object per name - there will be one instance of the Book object "The Alchemist" irrespective of the number of copies of that book available. It's member functions give access to it's fields and allow us to change some mutable fields like number of copies available. This is the biggest class and holds the most amount of information.

### **2.3.2 Employee**

This class is derived multiple times to create sub-classes like sales clerk, manager and owner . These sub-classes allow us to implement the organizational hierarchy of the book shop.

### **2.3.3 Sales Day**

- This class allows us to maintain an account of all the sales that take place during working day.
- This is a very important class that stores the sales statistics for the day. Useful to get an overall picture of the sales statistics of a period and derive useful information from it.

### **2.3.4 Not In Collection**

Not In Collection is the order request that contains the details of the book as specified by the customer.

### **2.3.5 Cart**

A customer may choose more than one book to buy in one transaction (and more than one copy per book). The cart maintains the list of books he/she wishes to buy.

### **2.3.6 Main**

This is the main application class that binds everything together. It contains the arrays of Book objects and registered employees. It also contains information regarding the books that have fallen below the threshold stock level and the books that are in demand but not a part of the collection. The sales statistics are also maintained in this class.

### **2.3.7 Other auxiliary classes :**

Publisher and vendor classes have been introduced to better implement the Book class. Each vendor may house books of multiple publishers. Many books may have the same publisher.

## **2.4 Constraints**

As the BAS needs to handle large number of books, that may lead to lagging of the system to handle data from database. A linear time algorithm may itself lose efficiency in such cases. Logarithmic time algorithms can be used ,but they will compromise on the memory used.

## **2.5 Assumption and Dependencies**

We have assumed that the computer used in the book shop is perfectly working and has a perfectly working Java runtime environment and JVM installed. As the software is built using Java, it is safely assumed that appropriate software is installed.

Coming to the problem statement, in order to give a more general perception to the problem in hand, there are other types of employees other than the Manager and Sales Clerk.

# **3 Specific Requirements**

## **3.1 External Interfaces**

### **3.1.1 User interfaces**

User interface is used to provide communication between users and system. When users look at the interface, they should understand which pane is used for which purpose. Each task of an interface should be specified clearly and users should use them correctly. For example, when users press to any button on

interface, they should know which operations are done by pressing this button. The user interface should be easy to learn. When users use the user interface, they should know which element is used to which operations. If the user interface is very hard to learn by the user then teaching the interface activity would take longer time and hence there will be an extra cost for teaching the user interface of the product to the user. The interface actions and elements should be consistent. When users press any button, required actions should be done by the system. The screen layout and colour of the user interface should be appealing. When users look at the screen, it will have a nice vision. Colours will be selected clearly, thus eyes of users won't feel tired. It has been required that every form's interface should be user friendly and simple to use.

### **3.1.2 Hardware Interfaces**

The hardware interface for the user would be any PC having a configuration of P-IV and above 2GB HDD for loading any OS so that BAS could interact with the system without any problem. The main interface would be monitor, keyboard and mouse.

### **3.1.3 Software Interfaces**

Book Shop Automation Software will use database for storing and management of records. So an access to the database management system is required. When such an event occurs the system establishes connection to the database management system, once the connection is created; the client program can communicate with the database management system. A standard called Java Database Connectivity (JDBC) provides an application programming interface (API), which allows client-side programs to call database management system, as long as PC has the necessary software installed. Most database management system vendors provide JDBC drivers for their systems. A user can actually connect to several database management system and send query and transaction requests using the Java Database Connectivity (ODBC) API, which are then processed at the server side. Any query results are sent back to user, which can process or display the result as needed.

### **3.1.4 Communications Interfaces**

For communications sockets on TCP shall be used. A client program creates a socket on its end of the communication and attempts to connect that socket to a server. When the connection is made, the server creates a socket object on its end of the communication. The client and server can now communicate by writing to and reading from the socket. The `java.net.Socket` class represents a socket, and the `java.net.ServerSocket` class provides a mechanism for the server program to listen for clients and establish connections with them. The following steps occur when establishing a TCP connection between two computers using sockets:

## 3.2 Functions Requirements

### 3.2.1 Check for availability of book:

When customer selects this option he is required to enter book title or the author name of book. The system would search the books in the books register based on the keywords. After making the search the system should output the details of all the books based on the details given.

3.2.1.1	Select query book availability option. <b>Input:</b> "query book availability" option is clicked. <b>Output :</b> User prompted to enter the key words.
3.2.1.2	Search for book name and display result. <b>Input:</b> Book title or the author name. <b>Output:</b> Display details of all the books, no. of copies available and the rack no. where the book is located.
Processing	Search the books in the book register based on the key words, if the book is not currently being sold by book-shop, then the customer is asked to enter full details of the book for procurement of the book in future. If a book is in stock, the exact number of copies available and rack number in which the book is located should be displayed. If a book is not in stock, the query for book is used to increment a request field for the book.

### 3.2.2 View request:

Once the manager selects this option, the system displays the current demand for different books.

Precondition	Manager is logged in
3.2.2.1	Select view request option. <b>Input:</b> "Find Trends" option is clicked. <b>Output:</b> Display the current demand of different books.
Processing	Displays the books list if any requests are present otherwise it gives no pending requests

### 3.2.3 Purchase Book:

Once the customer selects this option the system will ask to enter the ISBN no. of books sold. And the system will take the prices of books from inventory and generates the bills and updates the stock and generate the sales receipt for the book.



3.2.3.1	Select purchase option . <b>Input:</b> “purchase “option is clicked. <b>Output:</b> Prompt message to the salesclerk to enter the ISBN number of selected book.
3.2.3.2	Generate sales receipt. <b>Input:</b> Enter the ISBN number of selected book. <b>Output:</b> Gets the price from inventory and prints the sales receipt and updates the stock.
Processing	Generates the bill based on the ISBN number of book selected by customer.

#### 3.2.4 Update stock:

Once the employee selects this option,he would be asked to enter the list of new book name.

Precondition	Employee is logged in.
3.2.4.1	Select update inventory option. <b>Input:</b> “update inventory” option is clicked. <b>Output:</b> Employee will be prompted to enter the details of books.
3.2.4.2	Updating the inventory. <b>Input:</b> Employee will enter the book details that is procured for the first time. <b>Output:</b> Generates the ISBN number for the new books.
Processing	Updates the inventory by generating ISBN numbers of the books and updating the other details such as rack number, publisher, book name, and price.

#### 3.2.5 Generate Sales Statistics:

Once the manager selects this option, he will be informed about exact business done over any period of time. It also calculates inventory level required for various books.

Precondition	Manager is logged in
3.2.5.1	Select generate sales statistic option. <b>Input:</b> Select “generate sales statistic” option <b>Output:</b> System will generate the sales statistics
Processing	System calculates the inventory level required for a book which is equals to number of copies of book sold over a period of two weeks multiplied by average number of days it takes to procure the book from its publisher. And generate statistic according to it. Display it.

### **3.3 performance Requirements**

The software is supposed to be designed for a small to medium sized book shop. The book shop is assumed to house a few thousand books (or less) at any given point in time. Hence search results are supposed to be almost instantaneous. This goes for all the operations that a customer may perform, including updating the cart and requesting for a new book. The same can be said about printing sales receipt and viewing order requests. Displaying the sales statistics is the only computationally heavy operation and may potentially take a few minutes but only if the period entered is too large, say 10 years.

### **3.4 Logical Database requirements**

### **3.5 Design Constraints**

#### **3.5.1 Menu driven**

The system is based on menu driven interfaces. Menu selection will be done by using the mouse and the key board keys.

#### **3.5.2 Confirmation message**

Confirmation messages on taken actions, input acceptance and error conditions will be displayed after each input.

#### **3.5.3 Error Detection**

Error messages will be displayed at the time of detection of input errors and the system errors.

### **3.6 Software system quantity Attributes**

The software must include user authentication for all types of operation besides operation by the customer. Different classes of employees have different accessibility privileges and this must be strictly upheld for the data to remain safe.

### **3.7 Object oriented Models**