

# SOFTWARE ANALYSIS AND SOFTWARE DESIGN FOR BAS

Prudhvi Dharmana 12CS10037, Sumit Goswami 12CS10051

## Table of Contents

---

Introduction .....	2
1.1 Purpose .....	2
1.2 Analyze Stake Holders.....	2
1.3 Relevant Study .....	3
1.4 Alternatives .....	3
1.5 Criteria to Evaluate Alternatives .....	4
1.6 Unusual Circumstances.....	4
2. Software Analysis.....	5
2.1 Level 0 .....	5
2.2 Level 1 .....	7
Structured Analysis .....	8
3.1 Structured Diagram.....	8
3.2 Global System Architecture and Platform .....	8
3.3 Class Design.....	9
Book .....	9
Section .....	9
3.4 Database Design .....	11
3.5 Conclusion.....	11

## Introduction

---

### 1.1 Purpose

The purpose of this document is to provide detailed design specifications of Bookshop Automation Software (BAS). It analyses the problem and explains different design viewpoints using various diagrams and elaborates the planned implementations that needs to be carried out.

### 1.2 Analyze Stake Holders

#### CUSTOMER

**Description**

Customer is same as the customer for the bookshop. He can search, request for increments and request for new copies.

**Design Concerns**

There can be many customers and their data is kept only when they request books for reservation.

**Access Rights**

The customer has least access rights in BAS. He can only have a limited access of viewing the books from the store.

#### EMPLOYEE

**Description**

Employee is the one which is at the most generalized form of users who can control the data entered into BAS. He can add and edit books from the bookstore but cannot delete them.

**Design Concerns**

There can be many employees and all of them has a common interface thus following a **Factory Design Pattern**. The Employee should work with the database of the book. So he must have a login to edit the data. And each and every edit is stored into the bookstore with the corresponding employee.

**Access Rights**

The employee has higher access rights than the customer but cannot access any of statistics and revenue of the overall business done in the store.

## MANAGER AND OWNER

**Description**

Manager is higher than the employee and is almost equivalent to owner in BAS. Owner has one additional feature that he can also see which books have fallen below threshold so that he can procure those copies within time.

**Design Concerns**

There is only one manager and only one owner in BAS. Therefore it follows a **Singleton** Design Pattern and as employees they also have a common interface for login.

**Access Rights**

They have the highest access rights to BAS. They can do edit, delete and access all features that BAS provides.

### 1.3 Relevant Study

One of the major problems of any bookstore is organizing books into various sections. This is a standard problem that occurs to every bookstore. One of the popular solutions to this is to use internationally accepted **BISAC** separation of sections. This was implemented in BAS. At the start manager can choose which sections from BISAC correspond to what racks and in this way the statistics of books can easily be done with respect to sections. This also helps BAS to automatically know the rack place whenever a new book is arrived. So **instead of employees feeding data to BAS, BAS suggests employees to put the book in the particular rack.**

### 1.4 Alternatives

We are using **ODBMS (Object Database Management System) with JPA**. Other alternative to this might be using

1. **RDBMS (Relational Database Management System)** with JDBC in Java. But this has disadvantages like this is not close Object Oriented Design though it is faster. The basic idea of abstraction is lost and we work with native SQL queries.
2. **With Object serialization** though we can do it with this and it has as good abstraction but one major dis-advantage is we can't use this with large data. And it is much slower in running cost as well as implementing is difficult since we have to algorithms for each and everything unlike in SQL in which we can query.

So from these alternatives it is always better to use ODBMS and so we have chosen to use it.

### 1.5 Criteria to Evaluate Alternatives

The main criteria to evaluate alternatives come from the basic principles of software engineering i.e. quality of software, reusability and speed of development.

- In both the alternatives the development time is much higher
- For RDBMS reusability reduces gradually as the size of code increases.
- For serialization the quality of software becomes exponentially worse as the software usage increases and this should never be the property of an ideal software.

### 1.6 Unusual Circumstances

As for any practical problem unusual circumstances do arise in case of every software. One of the most common thing is power loss for which we can do nothing as the computer itself requires power. Another unusual circumstance is sudden crashing of our software either due to lack of memory from the operating system or suddenly the process is ended. **In such cases we don't have any data loss because we use persistence and not object serialization** and we may have a session loss and this has practically no significance and so can be neglected.

## 2. Software Analysis

### 2.1 Level 0

The Design flow of the whole software can be schematically described by the following diagram

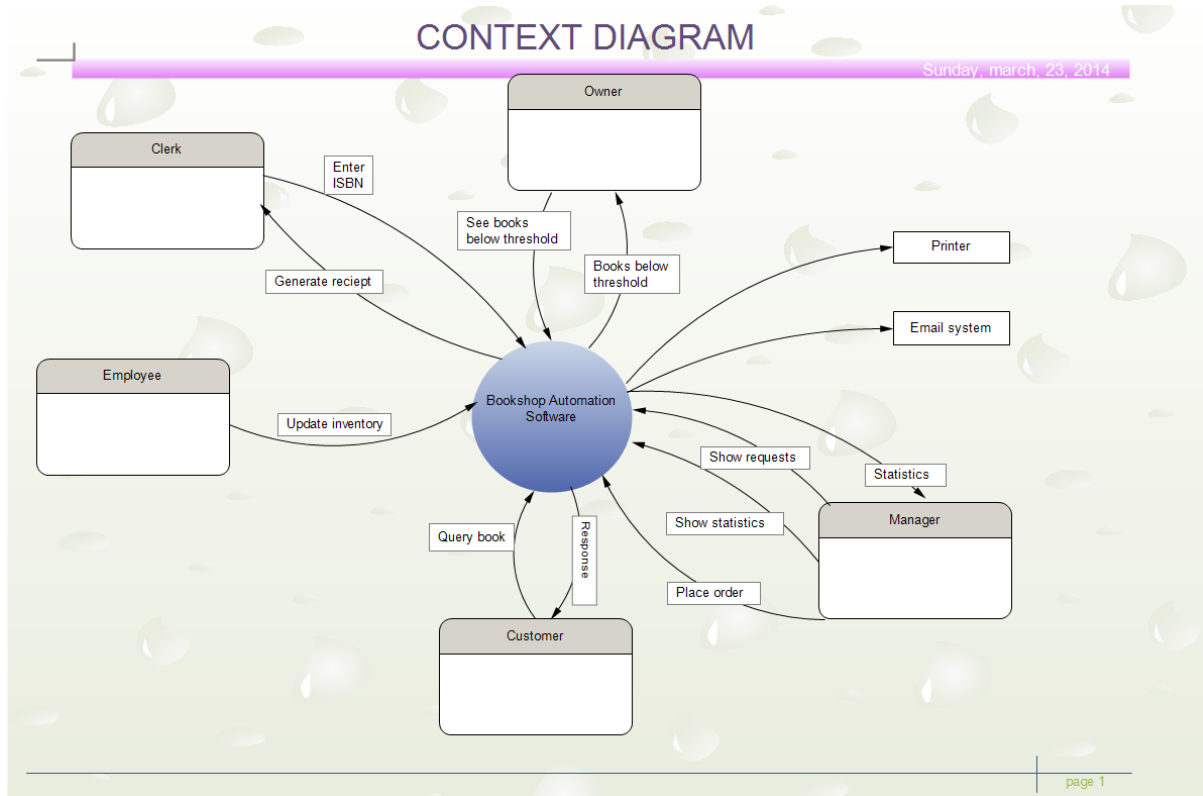


Figure 2.1

Fig. 2.1 provides an overview of the functionality of the whole software in a very crude term. It shows the distinct parts of the software and it interacts to effectively put the whole system working. The various parts of the system are explained below:

**BAS:** The main software we have been discussing. All the features like query book, show request statistics and place order have been explained in SRS in very detailed way.

**Employee, Owner, Clerk, Manager:** All these form the basis of updating and entering data into the BAS. All of these have been explained in SRS.

**Printer:** BAS is capable of developing a pdf format so that it can be sent to the printer and then receipt can be generated

**Mail System:** With a working internet connection BAS is capable of sending mails to customer notifying them about information of their reserved book data and their arrival.

All the above queries are time efficient as BAS uses persistence as the main database system and even the job of pdf printing is efficient as it used pdfBox one of the leading open source software and libraries for pdf printing from Java. Since implanting in Java is easier for GUI applications the development time is highly reduced.

## 2.2 Level 1

The flow of each of the operation mentioned above can be very well understood by studying the data flow diagrams for each of the following diagram:

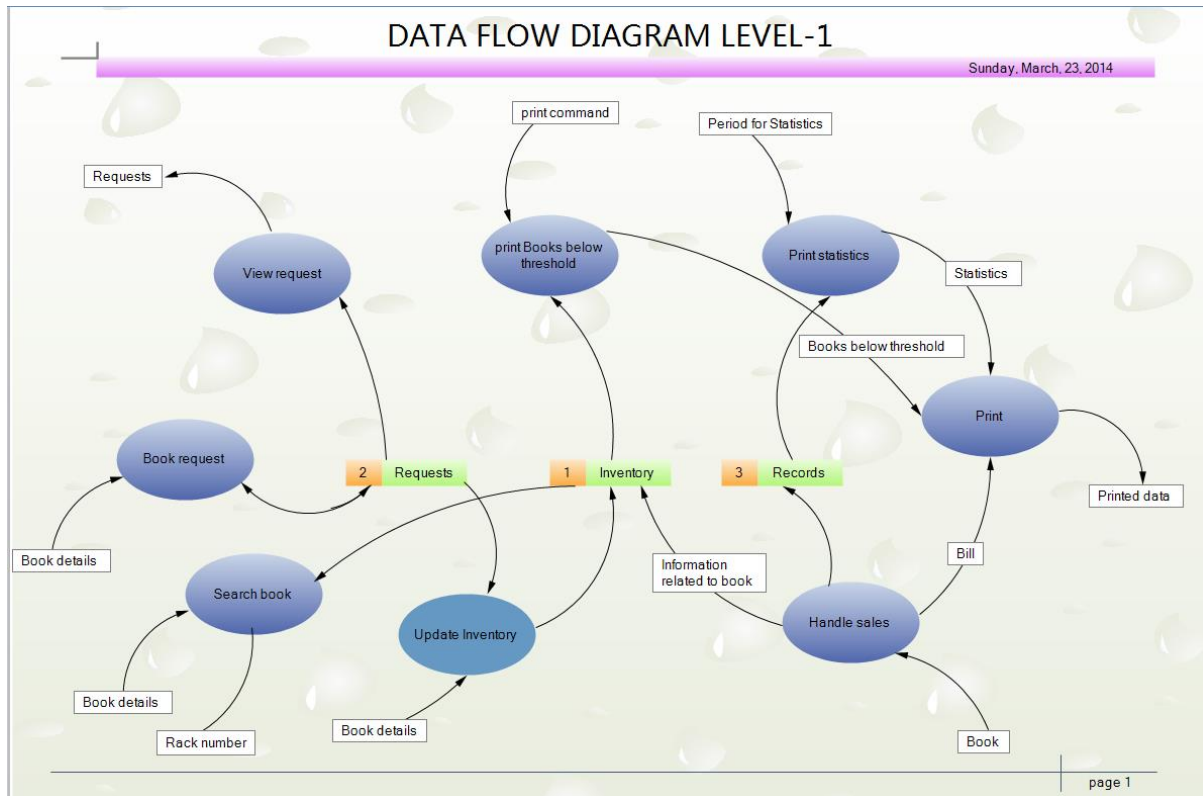


Figure 2.2

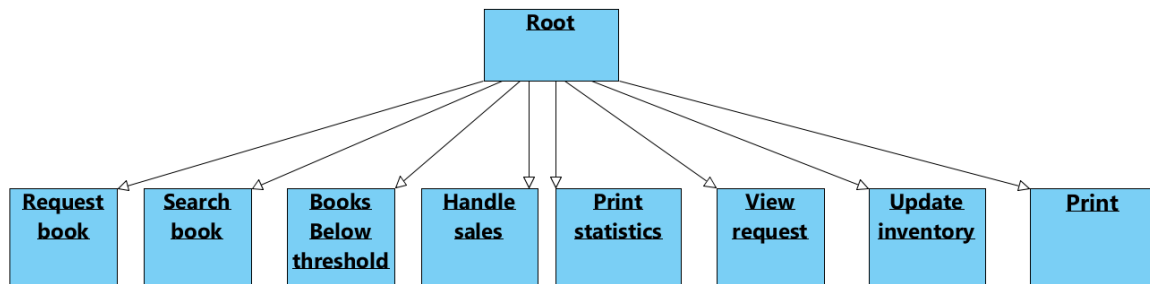
This is a level 1 data flow diagram that explains level 0 in a much deeper way.



## 3. Structured Analysis

### 3.1 Structured Diagram

The structured diagram is shown in the figure below.

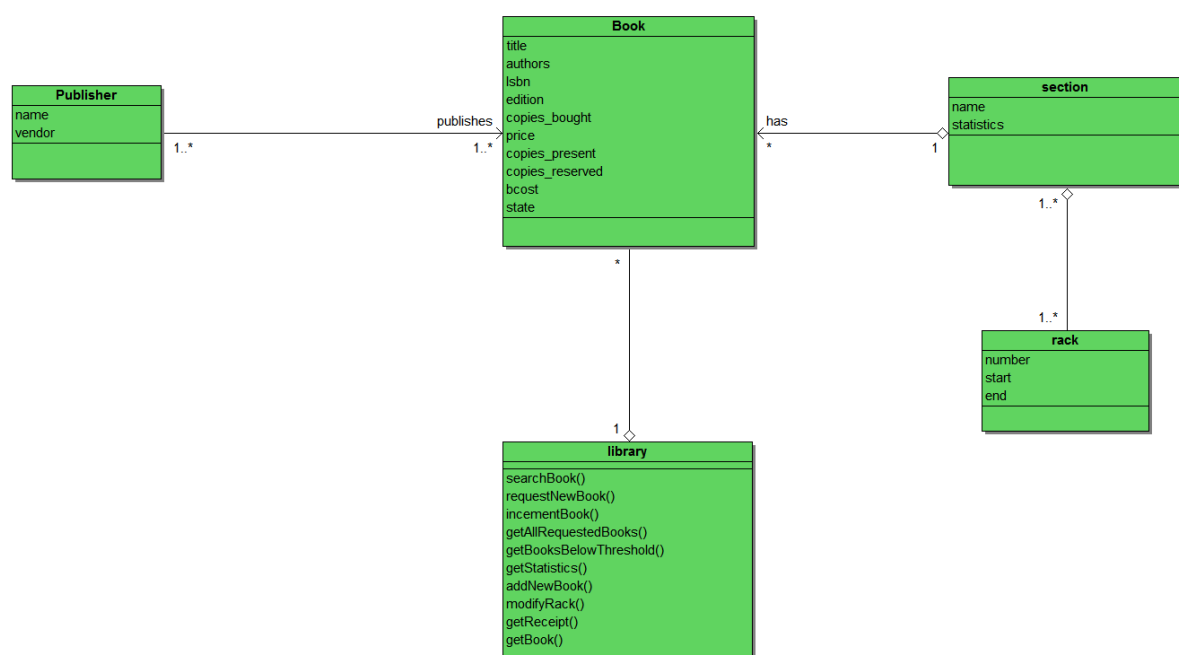


Structured analysis takes advantage of information hiding through successive decomposition (or top down) analysis. This allows attention to be focused on pertinent details and avoids confusion from looking at irrelevant details. As the level of detail increases, the breadth of information is reduced. The result of structured analysis is a set of related graphical diagrams, process descriptions, and data definitions. So from here we can greatly decompose the features into easier implementation

### 3.2 Global System Architecture and Platform

The global system architecture is Object Oriented architecture and implanted in Java language. So obviously Java works in JVM environment on any operating system. The software uses networking for e-mail sending and it uses Java inbuilt library for that. The advantages of using Java is it has many libraries available for doing tasks like printing and e-mail sending Java hibernate with JPA is extraordinary with its ODBMS capability. So we have chosen Java as our platform.

### 3.3 Class Design



The class design of maintain books can be clearly seen and well understood from this UML diagram.

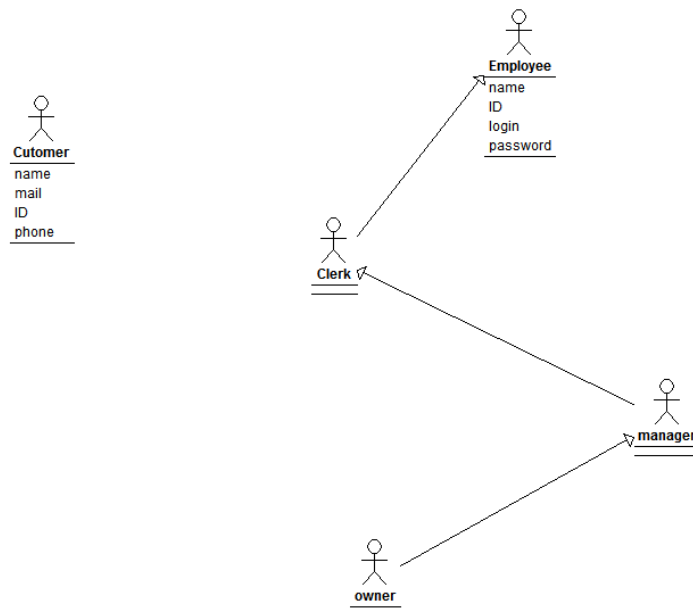
It allows books to have many publisher and a section can be contained in many racks and vice versa.

**Book**

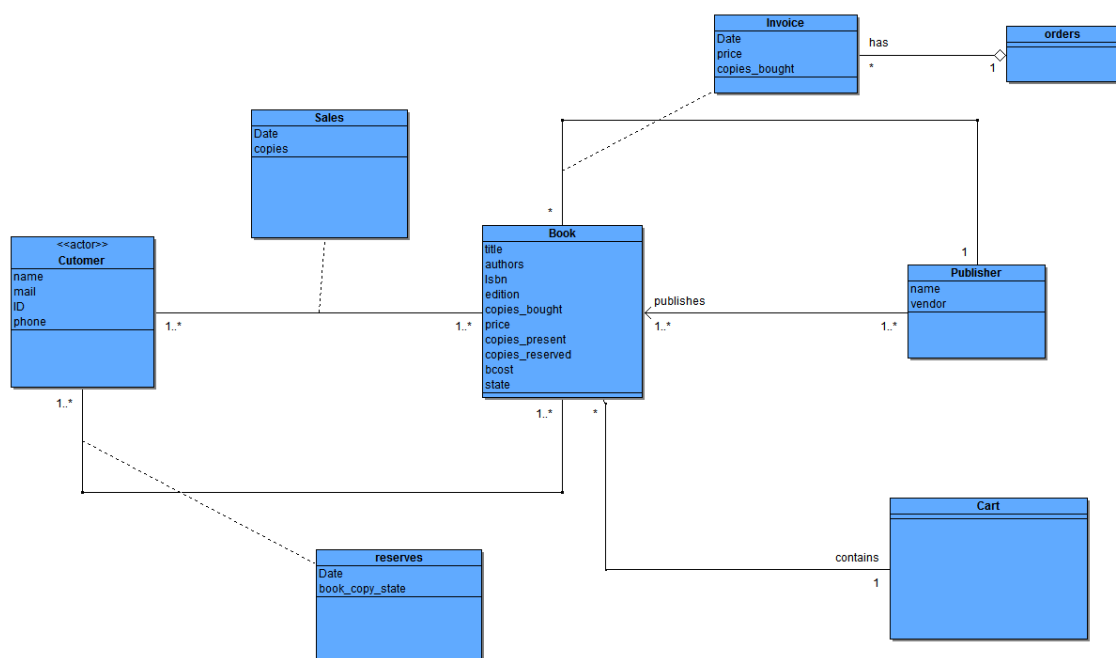
It denotes all books with same ISBN number i.e. all copies of the same book is denoted by this class.

**Section**

It denotes the BISAC sections and the rack denotes the sections which it contains.



The above diagram shows the hierarchy of classes for employees as the generalized person and manager can do all the things an employee can do. And this is a better design rather than taking each and every person as a specialized form of employee.



And this diagram shows the associations of all classes involved in BAS.

### 3.4 Database Design

We can keep ***reserves, orders, cart, sales, and library as tables*** in ODBMS with their classes they are associated as foreign keys.

### 3.5 Conclusion

The SA/SD document extending upon the BAS, SRS and the various function in forms of class diagrams. Attached with it a document containing all class diagrams, use case diagrams, sequence diagrams, communication diagrams, and state and activity diagrams.