SOFTWARE ANALYSIS AND DESIGN

for

Book Shop Automation Software

Version 1.0 approved

Presented by

Rameshwar Bhaskaran (14CS30027)
Aditya Bhagwat (14CS30002)
Group 55

IIT Kharagpur
March 27, 2016

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to document the detailed design specifications of the BAS. It contains analysis and dissection of the requirements with the help of diagrams.

1.2 Analysis of stakeholders

1.2.1 Customer

The main focus of this software is to satisfy customer's user experience in finding their book of choice. The customer is given least number of priveleges as he cannot control the database.

1.2.2 Employee

The employee is a generalised form of any person associated with the shop who has read/write permissions on the database. There can be multiple employees in the shop. The employee can update the inventory when new stock has arrived or when books are defective/sold.

1.2.3 Sales Clerk

The sales clerk is the employee who directly handles transactions with the customer. His main job is to receive the book's details(ISBN code) and appropriately generate the receipt, apart from the normal employee's priveleges.

1.2.4 Manager

The manager controls the requests and decides whether a particular book has to be procured from a vendor ,depending on the demand for the book, apart from the normal tasks a sales clerk can do.

1.2.5 Owner

The owner of the book shop is given the highest privelege. Apart from the tasks a manager can do, the owner can view books which have fallen below the threshold level and order them. There is only one owner in the shop.

1.3 Relevant Study

A number of related softwares were studied and one common feature which was found to enhance user experience multifold was the use of a "cart" wherein the user can add items to it. This prevents a highly impractical situation where the user has to bill one book at a time.

1.4 Alternatives to used tools

The software is built with the Java as the main language and is designed to work on Linux and Windows. Linux will be the most preferred platform however. Swing is used for building the GUI and the database vendor used is MySQL. JavaFX is also a good alternative as it has good animations and transitions. Also the PDF generation library used is iText library. The alternatives were considered also:

Swing

- JavaFX

MySQL

- ODBMS
- ORDBMS

iText

- PdfBox
- PdfClown

1.5 Criteria to evaluate alternatives

- Swing provides a stable user interface and is the industry's standard.Further, the support and documentation are in Swing's favour. JavaFX has a sleek interface and better looks , hence the software can also be built in JavaFX to improve user interface look.
- MySQL was preferred over ORDBMS and ODBMS because of its superior user community and popularity.
- iText library is the most popular Pdf generation library. As it had a richer number of features and a superior community,this was chosen

1.6 Unusual circumstances

As is the case with any software, the user session will be destroyed if the software is closed unexpectedly, like in the case of power shutdown etc. However the database is not lost as it is a persistent data. User session loss wouldn't affect the software much as no valuable data would be lost.

2 UML Diagrams

2.1 Use Case Diagrams

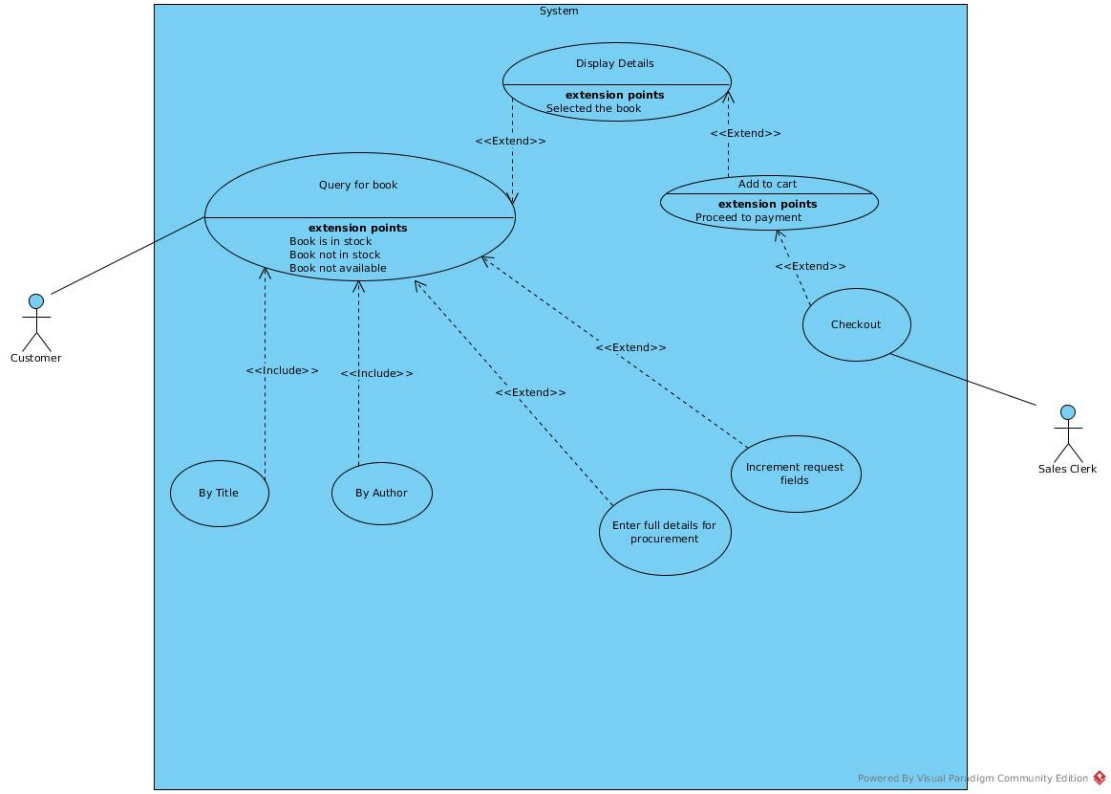


Figure 2.1: Customer related use cases.

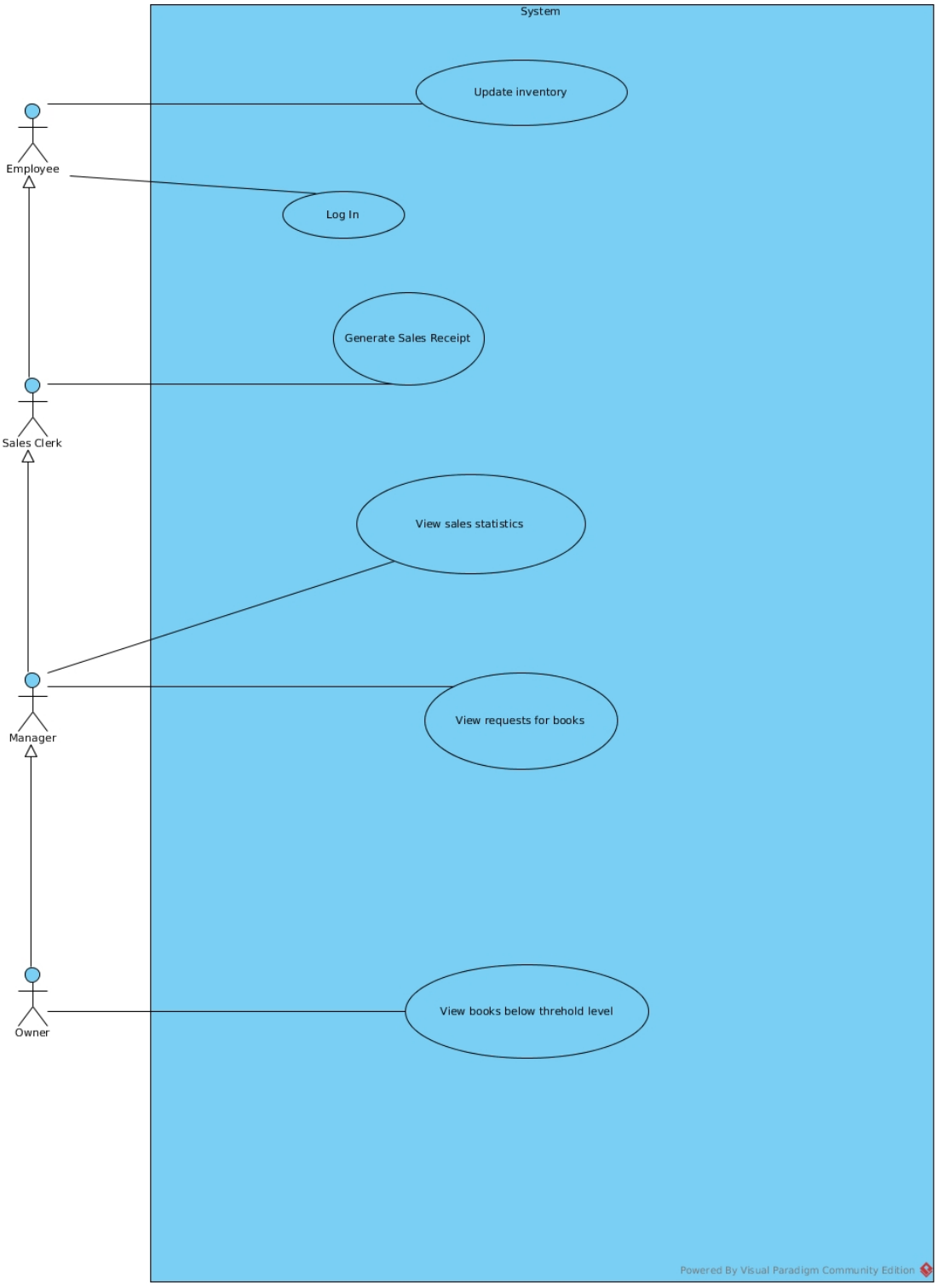


Figure 2.2: Employee related use cases.

2.2 Class Diagram

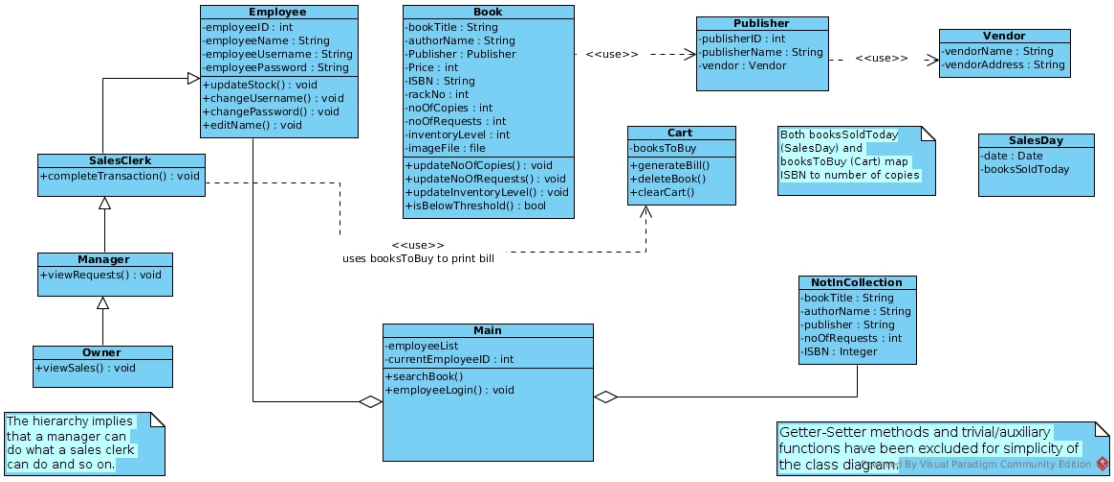


Figure 2.3: Class Diagram.

2.3 Sequence Diagrams

2.3.1 Book query and transactions

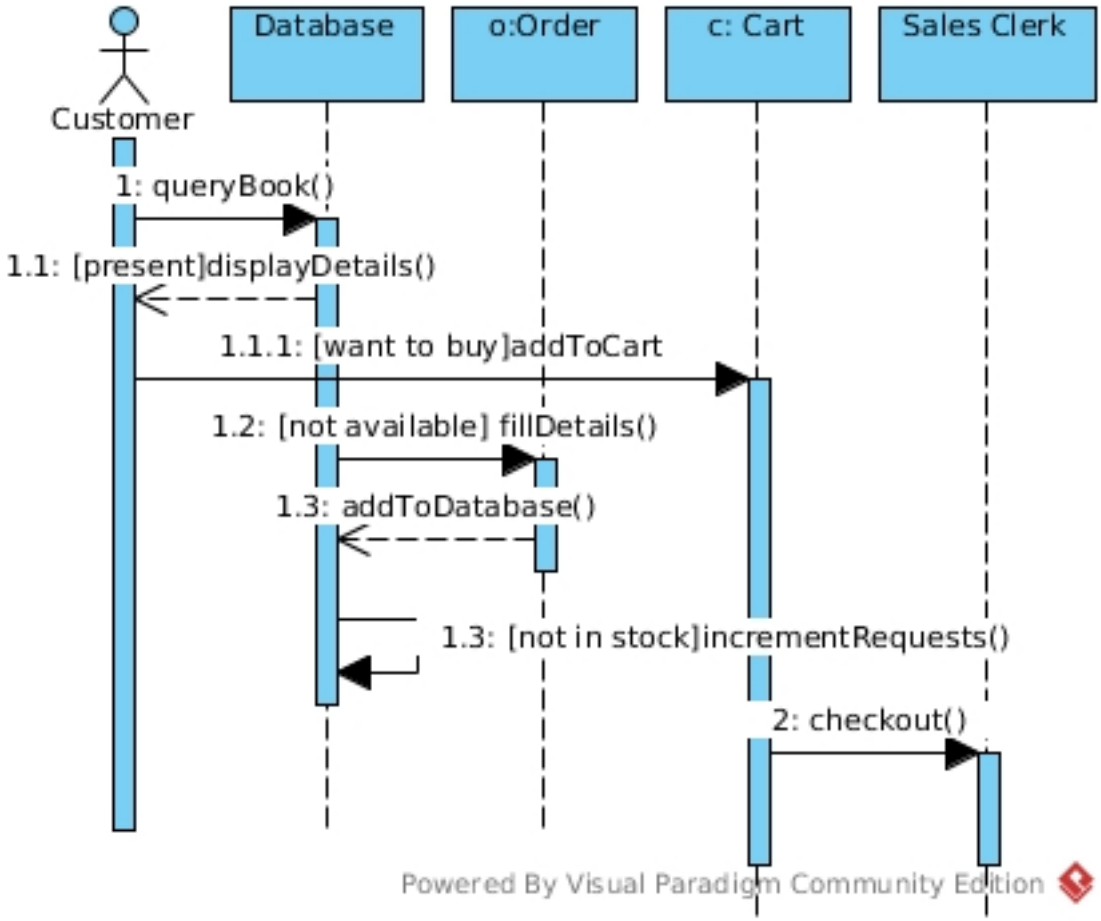


Figure 2.4: Book related transactions.

2.3.2 Generate receipt

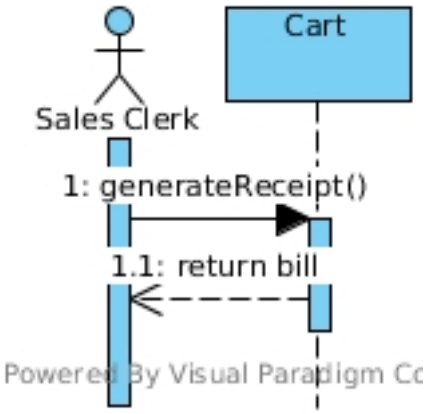


Figure 2.5: Sequence for generating receipt.

2.3.3 Procurement of books

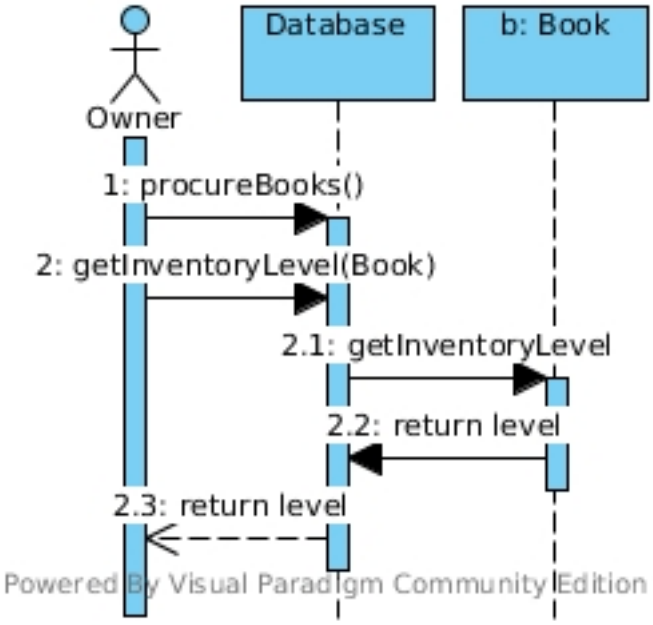


Figure 2.6: Sequence for procuring the book.

2.3.4 Update inventory

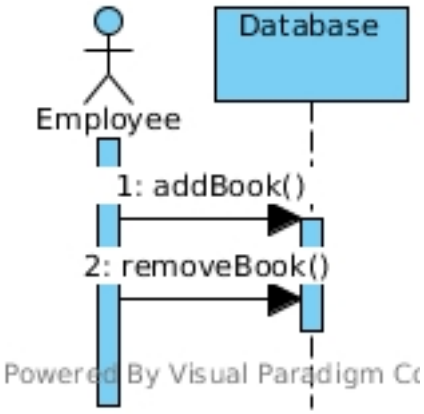


Figure 2.7: Sequence for updating the inventory.

2.3.5 View requests

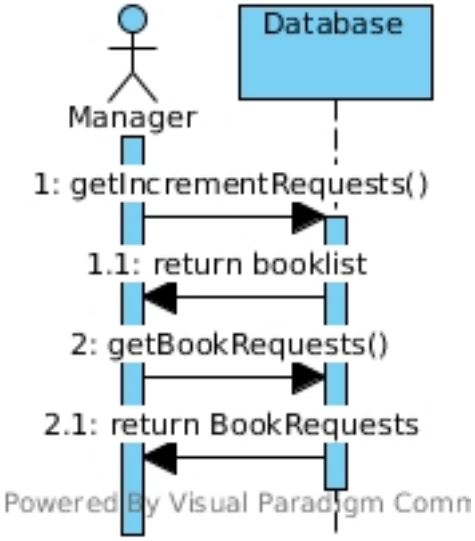
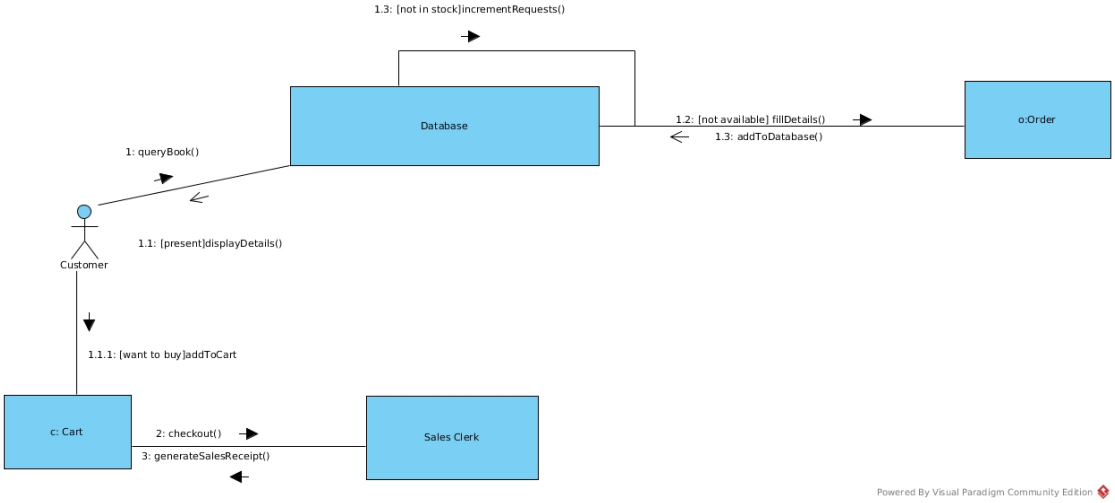


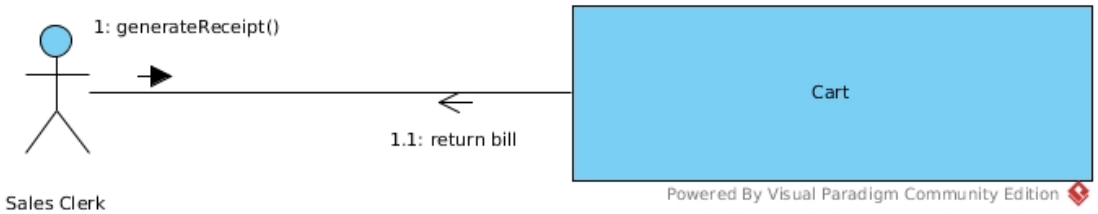
Figure 2.8: Sequence for viewing requests.

2.4 Communication Diagrams

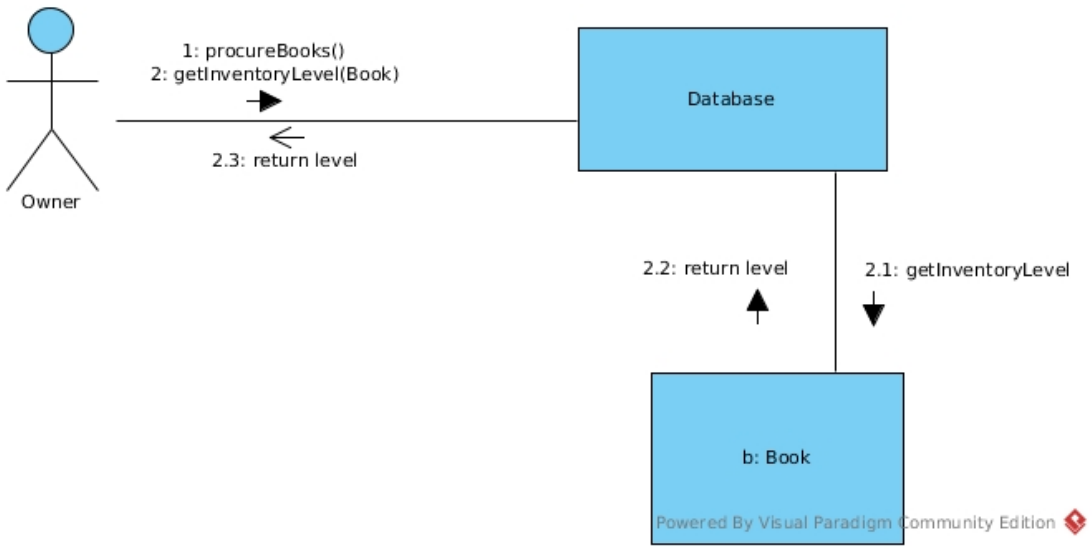
2.4.1 Book query and transactions



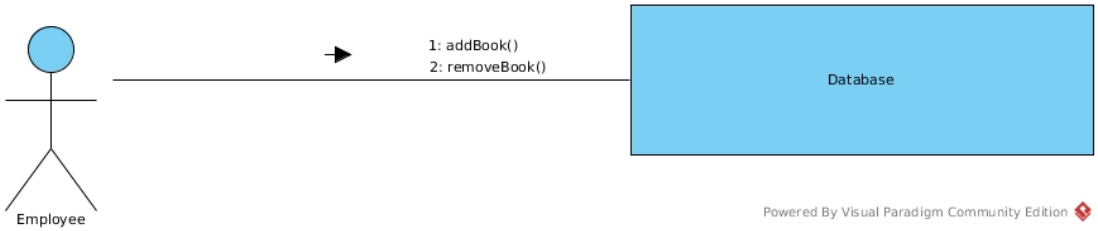
2.4.2 Generate receipt



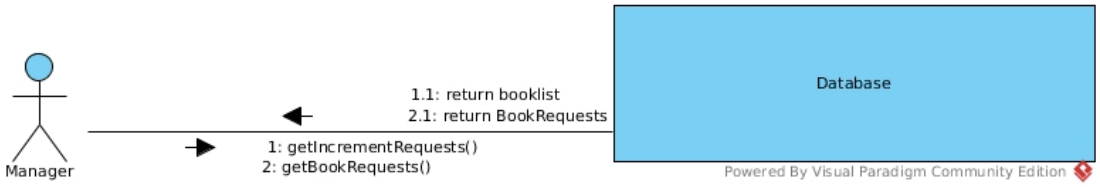
2.4.3 Procure books



2.4.4 Update inventory

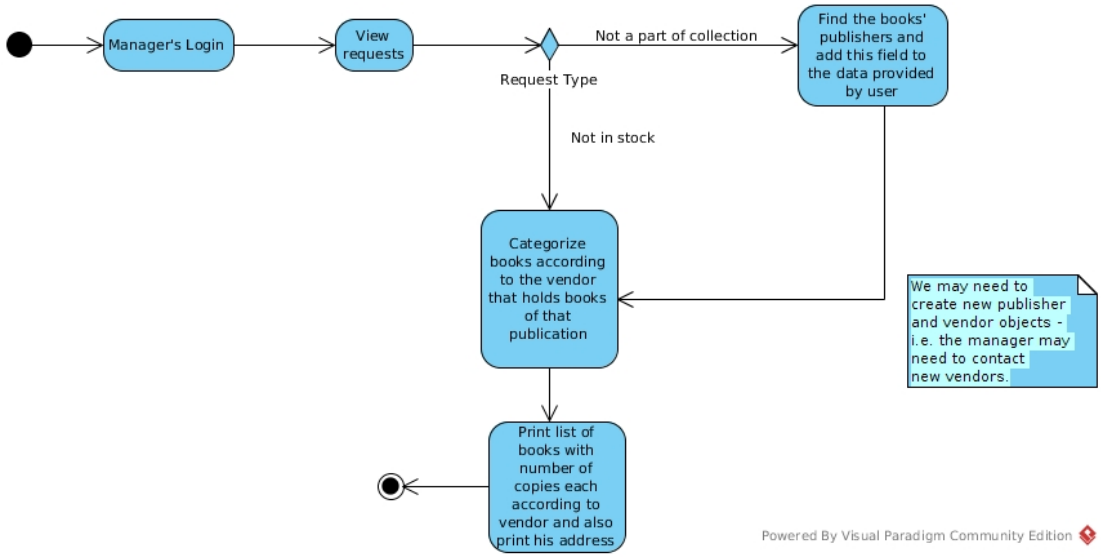


2.4.5 View requests

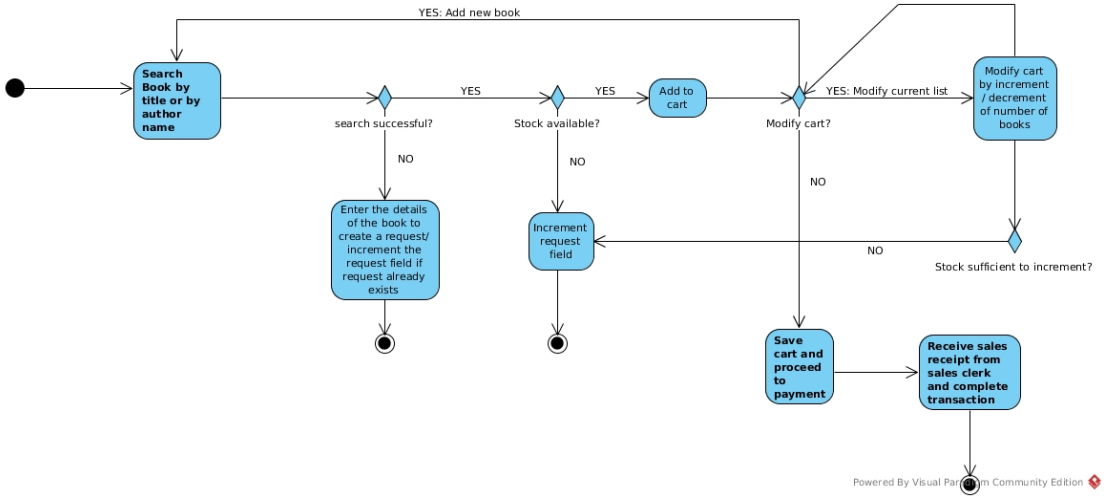


2.5 Activity Diagrams

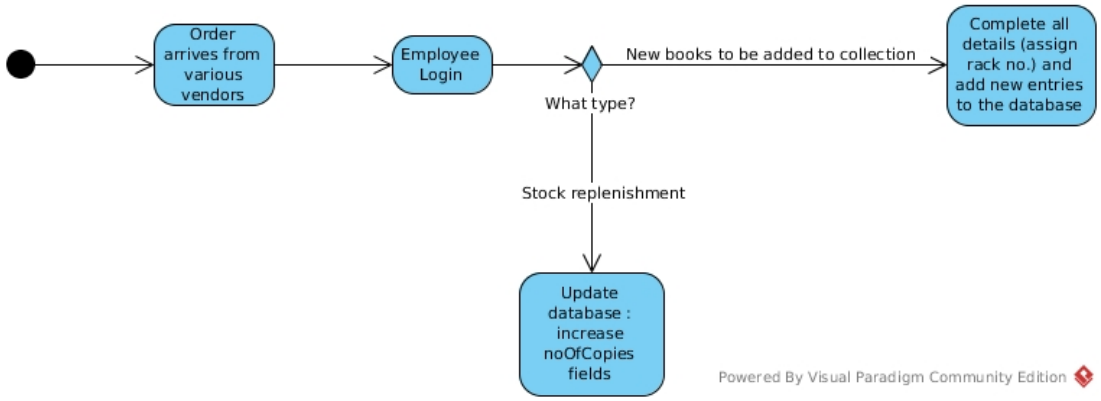
2.5.1 Manager's activity



2.5.2 Book query

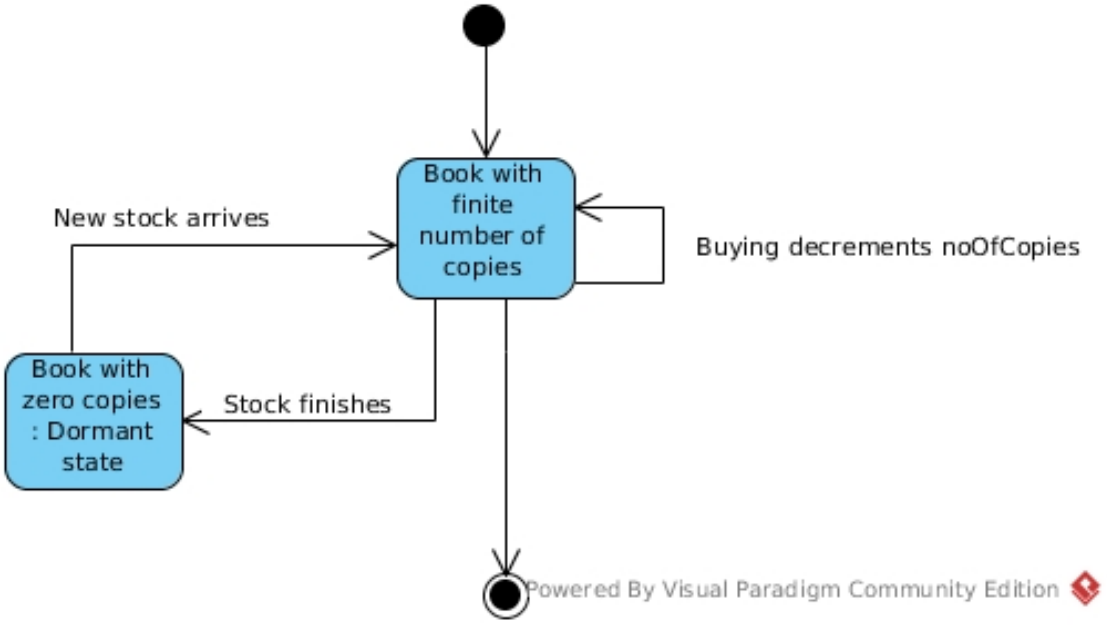


2.5.3 Stock arrival

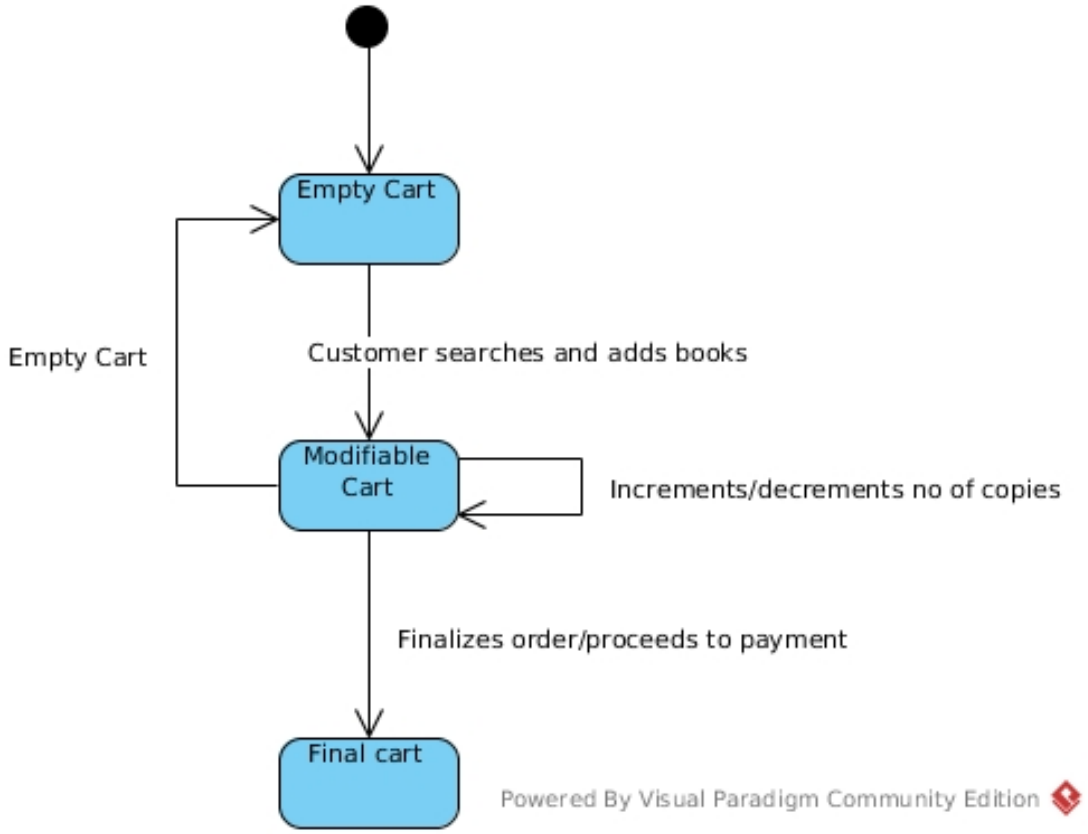


2.6 State diagrams

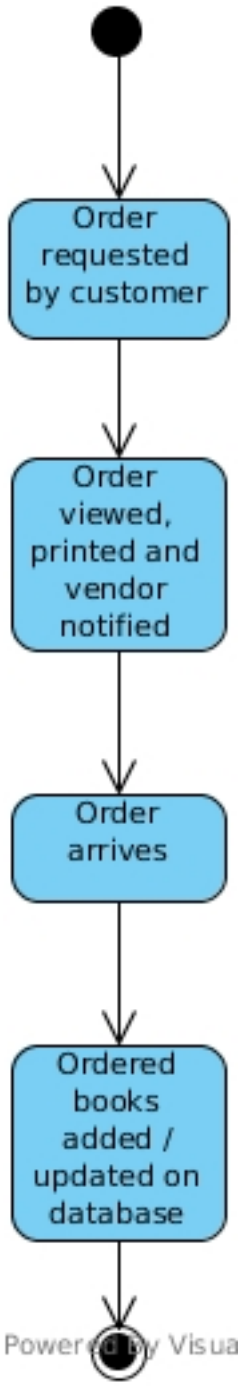
2.6.1 Book



2.6.2 Cart


Empty Cart
Empty Cart
Customer searches and adds books
Modifiable Cart
Increments/decrements no of copies
Finalizes order/proceeds to payment
Final cart
Powered By Visual Paradigm Community Edition

2.6.3 Order


Order requested by customer
Order viewed, printed and vendor notified
Order arrives
Ordered books added / updated on database

3 System analysis

3.1 System Parameters

The software is built in Java and using Swing library for GUI. The database used is MySQL and JPA/Hibernate framework is used for ORM mapping. The main external library used is the iText library for generating the sales receipt in pdf format and as mentioned before. The external hardware necessary is the printer to enable printing of the sales receipt. The software does not need networking in this version of the software as a local database is used.

3.2 Software Analysis

3.2.1 Level 0

The design flow of the diagram is shown below.

It provides the functionality of the application and shows interactions between the main players and the BAS. The main parts of the system are :

- **BAS :**
  This is the software which handles books in the shop. It has many features like query book, viewing statistics etc.
- **Employee :**
  He/she is the basic employee of the shop. He can update the inventory whenever needed.
- **Customer :**
  This stakeholder is the main person who the software caters for. However the least privilege is also given to the same.
- **Sales Clerk :**
  The sales clerk is a superior employee who handles transactions directly with the customer. He is in-charge of generating appropriate receipt and handle monetary transactions.
- **Manager :**
  The manager is a main employee who is in-charge of the functioning of the shop. He is as powerful as the owner when it comes to normal shop adminstration. However ,as expected the owner has obviously more privileges than the employee.

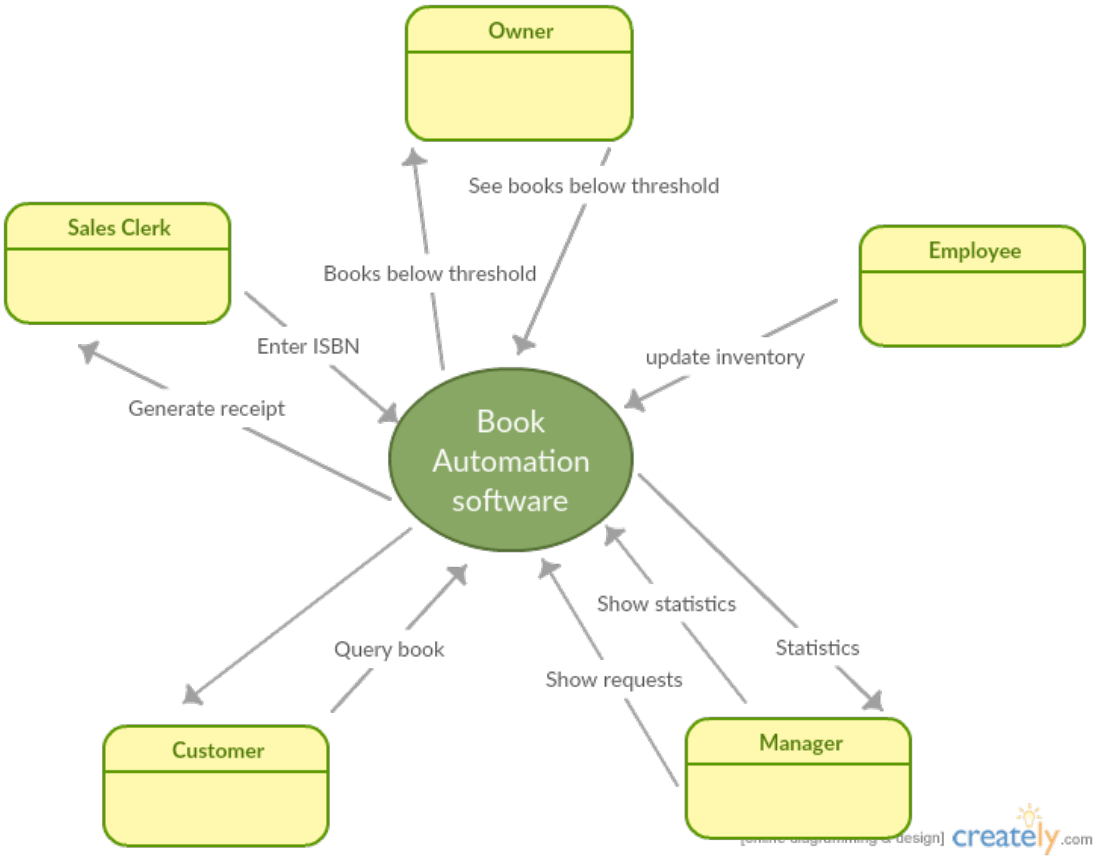


Figure 3.1: Context Diagram

- **Owner**
  The owner of the shop is given the highest privileges.He can do all that a manager can do , and also view books which have fallen below a threshold value and procure them when required.

  All operations are time efficient as the database operations are already optimised by the database vendor(MySQL).

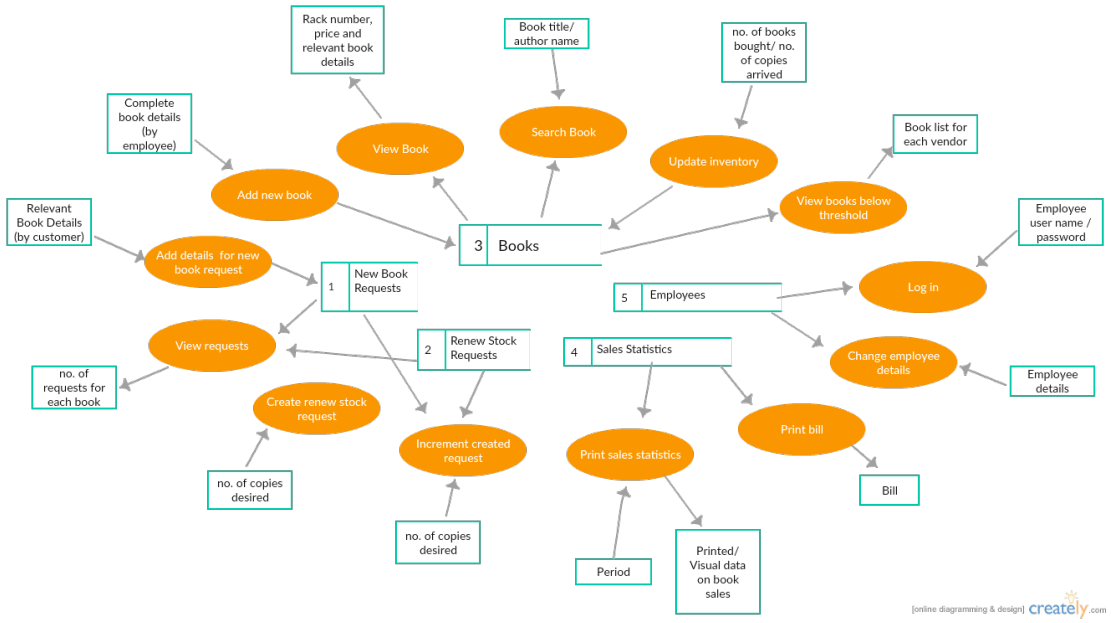


Figure 3.2: Data flow diagram

3.2.2 Level 1

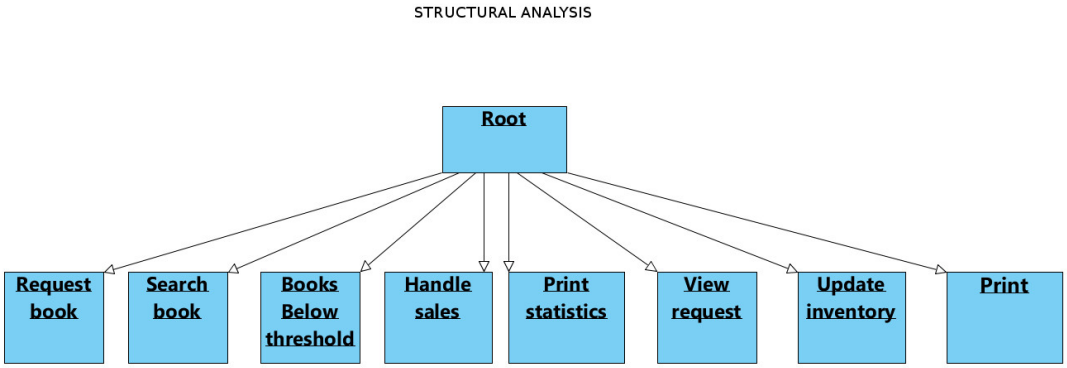
As is the case of software analysis , Level 1 gives an in-depth view of the software and intends to give a thorough analysis of the same. The above shows the dataflow diagram.

3.3 Database Design

The software needs to store the following details:

- Books
- Employee details
- Login details
- Sales invoices/details for the day
- Increment Requests
- Requests for new books

The database design will have to include each point as a table. A foreign key can be used between the Employee table and Login details table.



3.4 Structural analysis

The above diagram describes the basic structural flow of the BAS.

3.5 Global system architecture

The software is built using Object Oriented Programming concepts and the Model-View-Controller model is used for working and organisation of classes.

3.5.1 Class Organisation

As depicted in the class diagram, the classes have been organised to model the real-life situation, as is the philosophy of OOP.
The **Book** class is the class that stores Book attributes. The employee relations are modelled by the below diagram

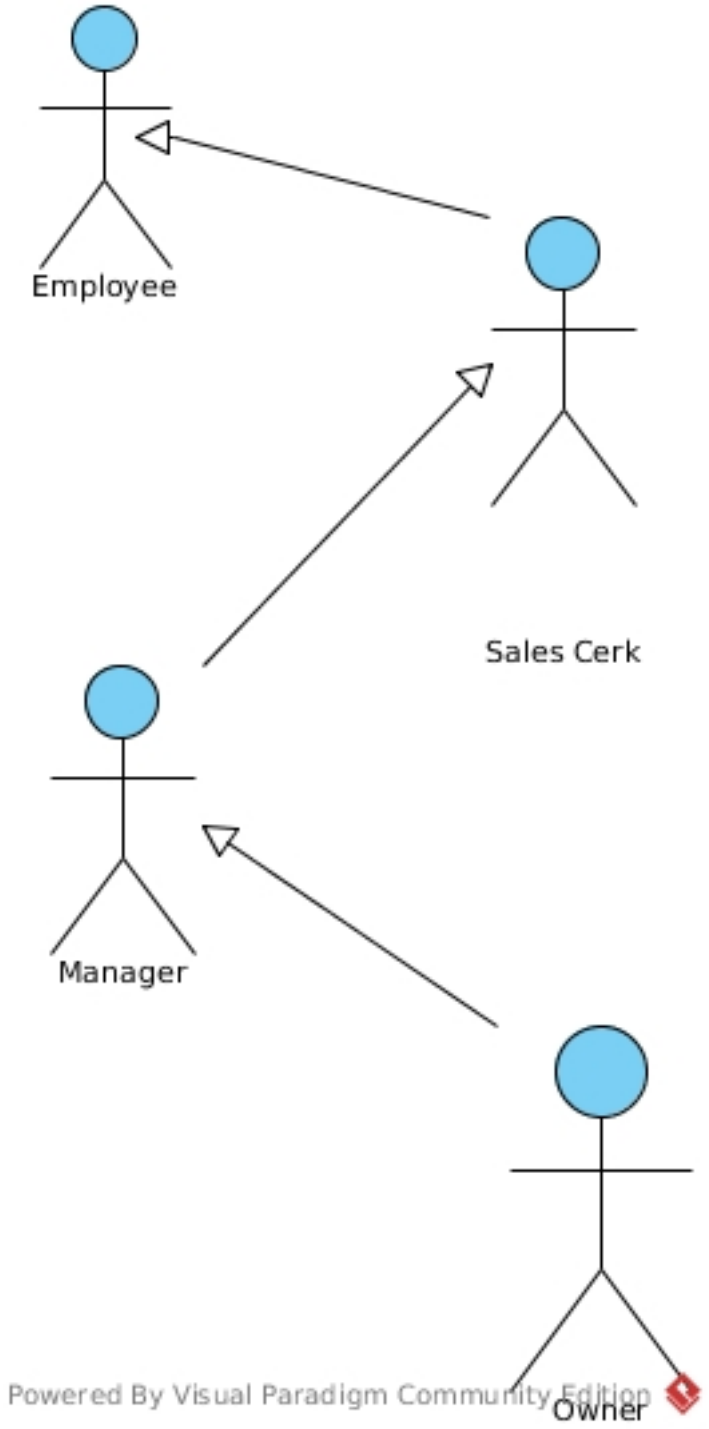


Figure 3.3: The relationship between employees

The **employee** related classes have

- **Employee**
- **Sales Clerk**
- **Manager**
- **Owner**

The software doesn't however need customer details as it is a small book shop and the process of collecting customer details may hinder user experience. However to implement a mailing system, we require the customer's email address and other minor details for notification.

The **requests** field have two types and uses three classes :

- **RequestOrder** *(abstract)*
- **NotInStock**
- **NotInCollection**

3.6 Other features/implementations

To give a good recommendation system and to correct spelling mistakes, the edit distance algorithm is employed. Also, to give proper queries, string matching algorithms are used to improve the recommendation engine.

3.7 Coding style

The following style is used for the coding this software.

- Class names begin with a capital letter
- Function names start with a small letter and then follow camel case in case of multiple words *i.e* ***getRequestsFromCustomer()*** etc.
- 4 spaces are used for a single indent
- Every non-trivial function is preceded by a line of comment describing its use/-function.
- All variable names and function names follow proper semantics.
- In case of comparison between integers, the style used is $0==n$ and not $n==0$.
- Constants are all in capital letters, like *PLANCK_CONSTANT*

3.8 Limitations and dependencies

This software is built considering that the book is a small book shop and contains around 10000 books. When the number of books are much more, the software may lag. Also, the present version of the software uses a local database to store data, which is a noticeable limitation. The BAS doesn't have any dependency to external software as such.