

Bookstore Inventory System Software Requirements Specification

Version 1.0

Revision History

Date	Version	Description	Author
9 Oct 2010	0.1	Vision Document Initial Draft	Ho Nam Ho Gerson Recinos Jemar Miller David Altum
10 Oct 2010	0.2	Vision Document Database, features reviewed	Ho Nam Ho Gerson Recinos Jimar Miller Adam Wurtzel David Altum Finan Bariagabr Francisco Diaz
16 Oct 2010	0.3	Vision Document Rough draft	Ho Nam Ho Gerson Recinos Jimar Miller Adam Wurtzel David Altum Finan Bariagabr Francisco Diaz
23 Oct 2010	0.4	Vision Document Final Draft	Ho Nam Ho Gerson Recinos Jimar Miller Adam Wurtzel David Altum Finan Bariagabr Francisco Diaz
21 Nov 2010	1.0	Inception Milestone Final Draft	Ho Nam Ho Gerson Recinos Jimar Miller Adam Wurtzel David Altum Finan Bariagabr Francisco Diaz

Table of Contents

- 1. Introduction 4
 - 1.1 Purpose..... 4
 - 1.2 Scope..... 4
 - 1.3 Definitions, Acronyms and Abbreviations 4
 - 1.4 References..... 4
 - 1.5 Overview..... 5

- 2. Overall Description 5
 - 2.1 Use-Case Specification 5
 - 2.1.1 Actors 6
 - 2.1.2 Use-Cases..... 6
 - 2.1.3 Use-Case Risk List. 7
 - 2.1.4 Use-Case Specifications..... 7
 - 2.1.4.1 Low/Over/Out of Stock Alert. 7
 - 2.1.4.2 Login. 8
 - 2.1.4.3 Search Book. 9
 - 2.1.4.4 Add Book/Category. 11
 - 2.1.4.5 Delete Book/Category..... 13
 - 2.1.4.6 Update Quantity..... 17
 - 2.1.4.7 Update Customer Info..... 19
 - 2.1.4.8 Record transaction into Transaction Database. 23
 - 2.1.4.9 Rented Book Returned..... 24
 - 2.1.4.10 Return Rental 26
 - 2.2 Assumptions and Dependencies 27

- 3. Specific Requirements 27
 - 3.1 Functionality 27
 - 3.2 Usability..... 27
 - 3.3 Reliability..... 28
 - 3.4 Performance 28
 - 3.5 Supportability..... 28
 - 3.6 Design Constraints 28
 - 3.7 Online User Documentation and Help System Requirements 28
 - 3.7.1 User Manual..... 28
 - 3.7.2 Online Help 28
 - 3.7.3 Installation, Guides, Configuration and Read Me File..... 28
 - 3.7.4 Labeling and Packaging..... 28
 - 3.7.5 Environmental Requirements 28
 - 3.7.6 Platform Requirements..... 28
 - 3.7.7 System Requirements 28
 - 3.8 Purchased Components 29
 - 3.9 Interfaces..... 29
 - 3.9.1 User Interfaces..... 29
 - 3.9.2 Hardware Interfaces 29
 - 3.9.3 Software Interfaces 29
 - 3.9.4 Communications Interfaces..... 29
 - 3.10 Licensing Requirements..... 29
 - 3.11 Legal, Copyright and Other Notices 29
 - 3.12 Applicable Standards 29

- 4. Product Acceptance Criteria..... 29
 - 4.1 Functionality available in version 1.0..... 29

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to define the high level requirement of the Bookstore Inventory Software System.

1.2 Scope

This program will be used as an all-around back-end bookstore inventory system. Some of the key features of the system are the following. The software will enable the client (business) to have real time statistics of sales and book inventory, automatically produce End of Day sales reports and End of Day low-inventory notices (purchase order suggestions).

It will also enable the users (customers) to view the real time inventory and extract book information, enable users to place orders online and pick up books in-store and access to a personalized account profile, order history, etc.

Lastly, the program will enable vendors to have access to part of the system – it will allow any vendor to update book information, add/remove new books to/from the inventory and updated prices.

1.3 Definitions, Acronyms and Abbreviations

This is a comprehensive list of all terms used in this vision document.

<i>POS (Point of Sale)</i> –	An electronic terminal that handles all credit/cash transactions.
<i>Vendor</i> –	A company/person who is in the business of selling products and goods to businesses.
<i>Inventory</i> –	a detailed list of goods and materials that are in stock.
<i>User</i> –	a person who can interact with the software – can be an employee or end user (customer).
<i>Client</i> –	The UNLV bookstore.
<i>Book Inventory</i> –	The detailed list of books in stock.
<i>Database (DB)</i> –	An organized (structured) body of related information.
<i>End of Day Report</i> –	A report that is done after business hours are over. Typically, it includes sales and inventory.
<i>Sales</i> –	The overall money transaction during a specified time interval.
<i>Transaction</i> –	The exchange of goods or services for legal tender.

1.4 References

None

1.5 Overview

In the following sections we outline the software product in higher detail. We will start with defining the key features that will be implemented. Next, we will discuss the constraints that will be imposed upon the software and the quality ranges, in other words, the robustness, fault tolerance and usability of the software product amongst other things. In the precedence and priority section we will comment on the most important functionalities that the software product must have and the integrity of the sales system.

In the following sections will discuss all other product requirements, such as, performance requirements, platform requirements and environmental requirements. Lastly, we will comment on the documentation requirements, such as, user manuals, online help & support, installation and packaging.

2. Overall Description

2.1 Use-Case Specification

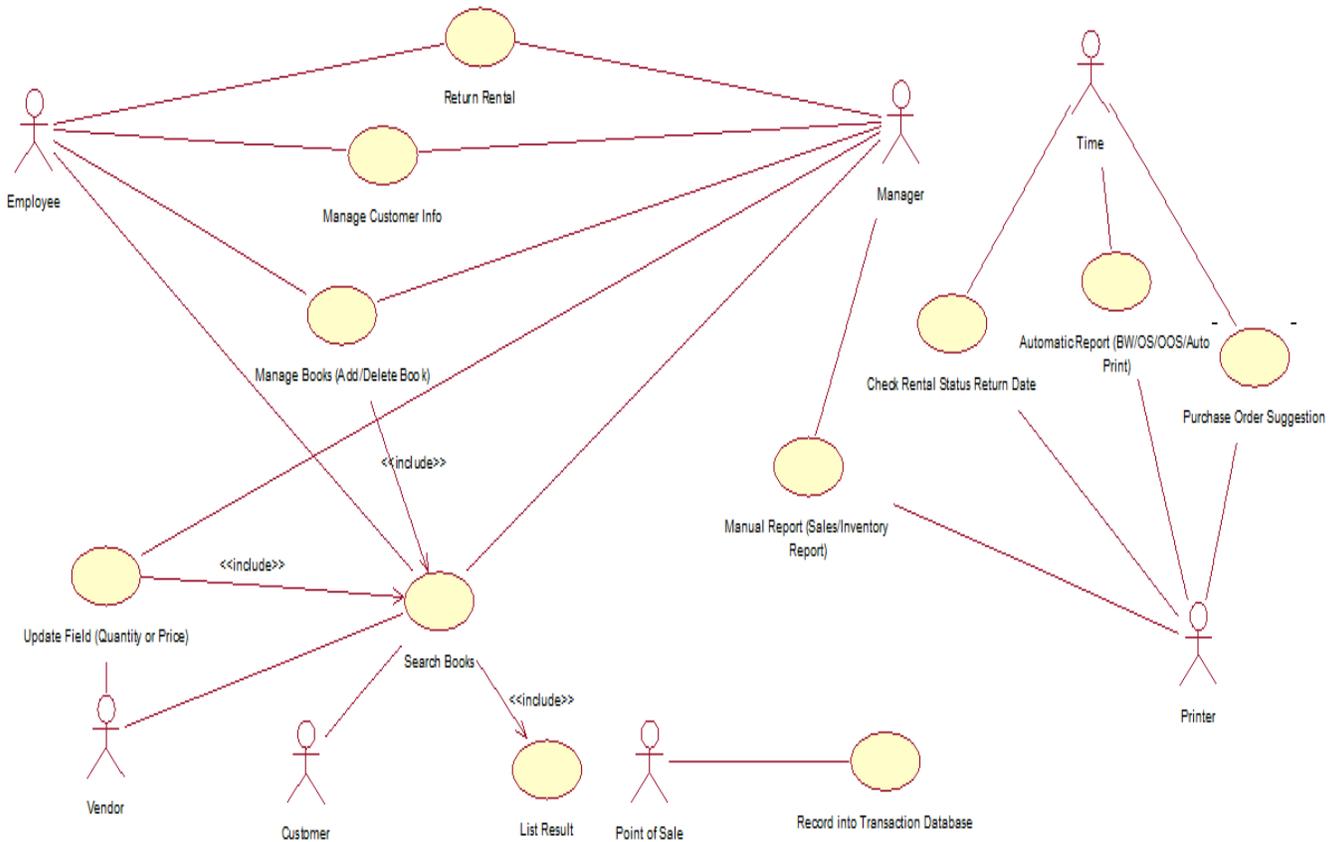


Figure 1 Use Case Diagram

2.1.1 *Actors*

Manager -	A manager of the bookstore
Employee -	An employee of the bookstore
Customer -	A customer of the bookstore
Vendor -	A vendor that supplies the bookstore
Printer -	A printer that the daily and requested reports prints
Point of Sale (POS) -	The sale transaction devices in the bookstore
Time -	The time that initiates the automatic systems of the software product

2.1.2 **Use-Cases**

Low/Over/Out of Stock Alert - This use-case describes the how alerts are handled when book-store items are in a low stock, out of stock and over stock alert state.

Login - This use-case describes the process of verifying identity to access a certain feature. In this case, it is being used to limit access of managing bookstore book prices to only managers and vendor book prices to only vendors.

Search Book - This use-case describes the process by which the system look thru the book database and find a list of books that fulfilled the parameters given. (Authors, ISBN, subjects, etc...)

Add Book/Category - The purpose of the add books/categories . This ensures that employees sell books that customers need for their classes. The system grants extra privileges to managers where they can create categories for any book in the database. This use-case describes the process of adding books or categories.

Delete Book/Category - This use-case describes the process by which the system deletes a book record in the database. This use-case also describes the process if a manager wants to delete a category of books.

Update Quantity - The purpose of this use-case is for the system to increase or decrease the number of books in the book database

Update Customer Info - The purpose of this use-case is for the system to update the customer information based on the input from the POS

Record Transaction into Transaction Database - In this use case the system records a transaction into the transaction database.

Rented Book Returned - In this use case, the system searches the customer database for all rented books that are due at a specified date. Checks whether the book(s) has been returned.

Return Returned - In this use case the system updates the customer information in the database after a customer return a book from the POS

2.1.3 Use Case Risk List

Use Case Risk are ranked from 1 - 10 with 1 being the highest risk use case and 10 being the lowest risk use case.

Record Transaction to Transaction Database	High
Delete Book/Category	
Add Book/Category	
Update Quantity	
Update Customer Info	
Return Rented	
Rented Book Returned	
Login	
Low/Over/Out of Stock Alert	
Search Book	Low

2.1.4 Use-Case Specifications

2.1.4.1 Low/Over/Out of Stock Alert

- **Brief Description**

- This use-case describes the how alerts are handled when book-store items are in a low stock, out of stock and over stock alert state.

- **Actors**

Book-store System

- **Dependencies**

Daily Report

- **Basic Flow of Events: Low/Over/Out of Stock Alert occurs**

1. The use-case begins automatically at the end of the day when the Book-store system checks the book database searching for three alert types which are low stock, out of stock, or overstock books.
2. For each low/out of/overstock book found the system writes into the daily report the books information such as book name, ISBN, alert type, and quantity.
3. The system creates an alert message that will be seen on next Manager Login that the daily report has new alerts.
4. The use-case ends.

- **Alternate Flow of Events #1: Low/Over/Out of Stock Alert does not occur.**

1. The use-case begins automatically at the end of the day when the Book-store system checks the book database searching for three alert types which are low stock, out of stock, or overstock books.
2. No alerts are found within the book database.
3. The system writes into the daily report that no alerts were found.
4. The use-case ends.

2.1.4.2 Login

- **Brief Description**

- This use-case describes the process of verifying identity to access a certain feature. In this case, it is being used to limit access of managing bookstore book prices to only managers and vendor book prices to only vendors.

- **Actors**

Managers, Vendors, Employees

- **Dependencies**

Add book, Delete book, Update quantity, update price, sales/refund, request sales report, request inventory report (Pretty much everything that request you to login before hand)

- **Basic Flow of Events: Login, No error**

1. This use-case begins when an actor attempts to manage book prices.
2. The use-case prompts the user for a login and password information.
3. The use-case processes this information after the user presses "enter".
4. The use-case verifies the login information provided is correct.
5. The use-case gives access either to the bookstore prices or the vendor prices depending on the login information.
6. The use-case ends.

- **Alternate Flow of Events #1: Login, incorrect password/username**

This flow of events describes the process of making changes to previous selection. It follows the basic flow of events up to step three (3)

1. This use-case begins when an actor attempts to manage book prices.
2. The use-case prompts the user for a login and password information.
3. The use-case processes this information after the user presses "enter".
4. The use-case attempts to verify the provided login information is correct.
5. The use-case displays an appropriate message indicating the provided information is incorrect and how many attempts left.
6. The use-case restarts the process from step one if the number of attempts left is not zero.
7. The use-case locks access to this feature if number of tries left is zero.
8. The use-case ends.

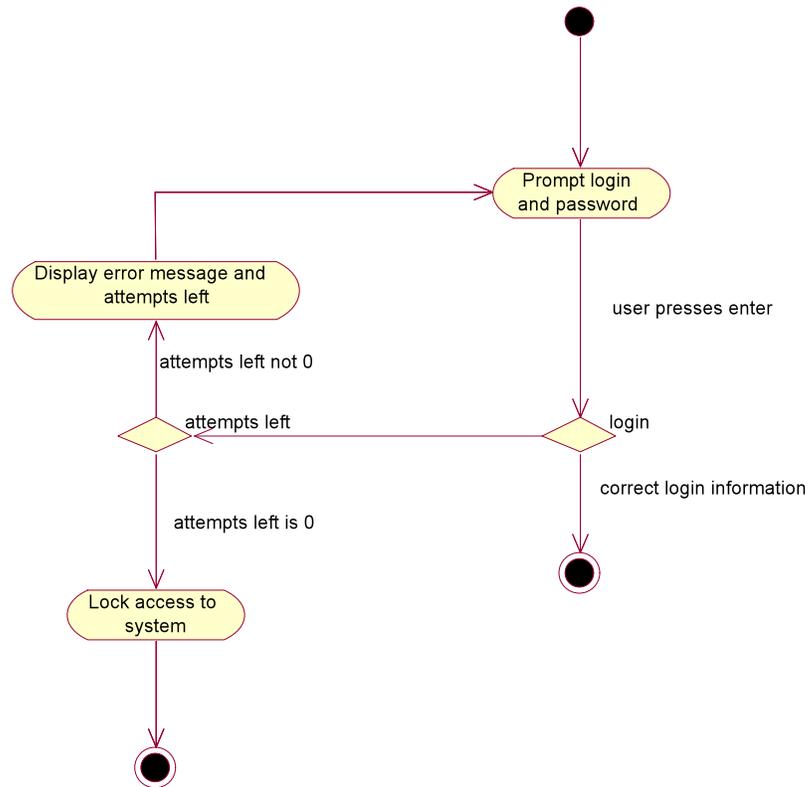


Figure - Login Use-Case Diagram

2.1.4.3 Search Book

- **Brief Description**

- This use-case describes the process by which the system look thru the book database and find a list of books that fulfilled the parameters given. (Authors, ISBN, subjects, etc...)

- **Actors**

Book database, customer, managers, employee, vendor, website

- **Dependencies**

List Result.

- **Basic Flow of Events: Search for book/s, found the book/s**

1. The use-case begins when an actor requests for a search of book(s).
2. The use-case prompts the user for a search parameter
 - i. Search by ISBN
 - ii. Search by Authors
 - iii. Search by publishers
 - iv. Search by Subjects
 - v. Search by Etc...
3. The use-case prompts the user for appropriate information based on the choice of method (author name, department name and/or course number, etc...)

4. The user presses “search book” on screen.
 - i. The use-case processes the provided information for a search.
 - ii. The use-case displays a summary of requested parameters for finding the book/s
5. The use-case generates and returns the list from the found books using “List Result” use-case.
6. The use-case ends.

• **Alternate Flow of Events #1: Search for book/s, did not find specific book/s**

This flow of events describes the process of making changes to previous selection. It follows the basic flow of events up to step three (3)

1. The use-case begins when an actor requests for a search of book(s).
2. The use-case prompts the user for a search parameter;
 - i. Search by ISBN
 - ii. Search by Authors
 - iii. Search by publishers
 - iv. Search by Subjects
 - v. Search by Etc...
3. The use-case prompts the user for appropriate information based on the choice of method (author name, department name and/or course number, etc...)
4. The use-case attempts to process the provided information
5. The use-case displays an appropriate message indicating the provided information is incorrect and what information is required to continue with the search.
6. The use-case restarts the process from step three;
 - o The use-case can restart from step one if the user wishes to change the search parameter
7. The use-case ends.

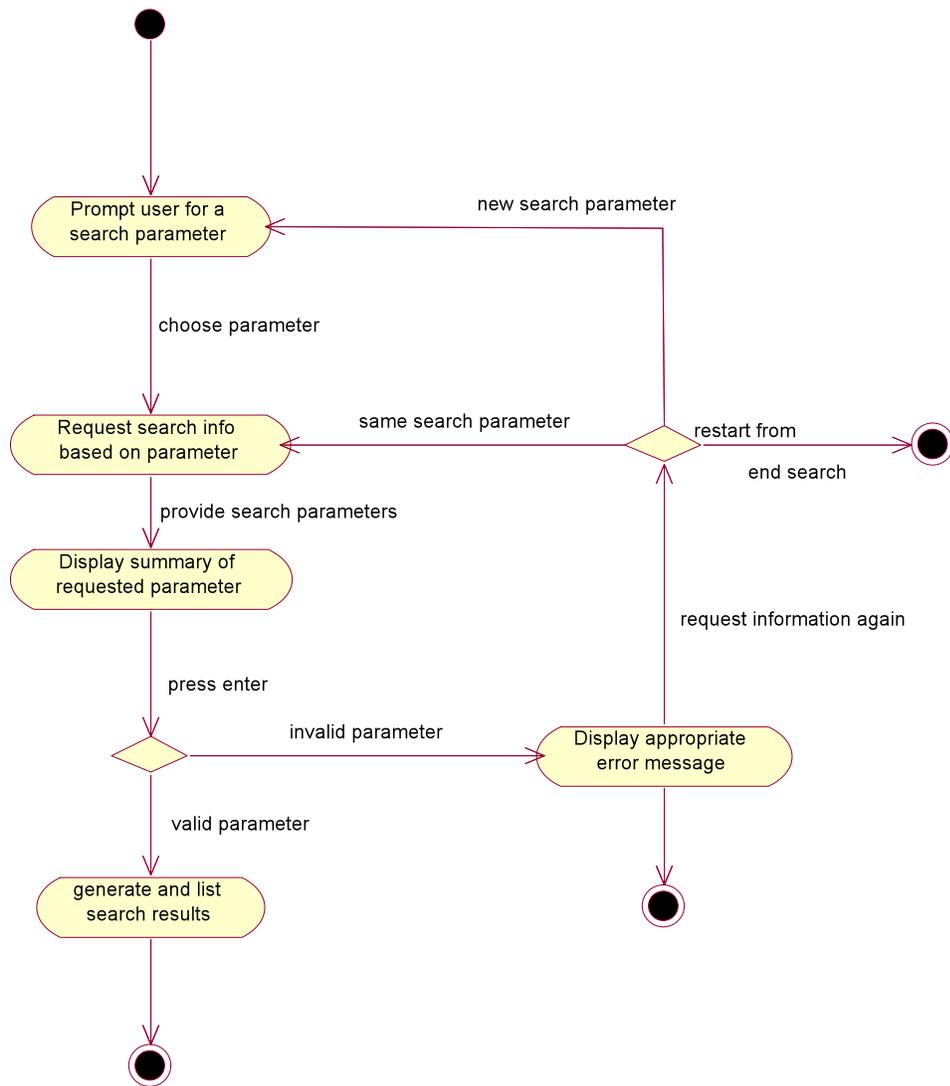


Figure - Search Book Use-Case Diagram

2.1.4.4 Add Book/Category

- **Brief Description**

- The purpose of the add books/categories . This ensures that employees sell books that customers need for their classes. The system grants extra privileges to managers where they can create categories for any book in the database. This use-case describes the process of adding books or categories.

- **Actors**

Managers, Employees

- **Basic Flow of Events: Add book**

1. The use-case begins when either a manager or employee choose to add a book.
2. The employee or manager scan the ISBN.
3. The system fetches all pertinent information to populate our database.
4. The system inserts the information from the Library of Congress into the database.
5. The system has added the scanned book into the database.
6. The use-case ends.

- **Alternate Flow of Events #1: Add category**

This flow of events describes the steps taken if a manager wants to add a category:

1. The use-case begins when a manager chooses to add a book category.
2. The manager navigates to the location where the category is to be added.
3. The system displays a dialog box for the new category.
4. The manager types the book category name.
5. The system verifies that a duplicate category does not exist in the destined location.
6. The system informs the manager that the addition of a book category succeeded.
7. The use-case ends.

- **Alternate Flow of Events #2: Add book, but ISBN is not found**

This flow of events describes the steps taken if an employee or manager add a book, when the ISBN is not found from the Library of Congress:

1. The use-case begins when an employee or manager chooses to add a book.
2. The employee or manager scan the ISBN.
3. The system fails to locate the ISBN from the Library of Congress.
4. The system informs the employee or manager that the ISBN was not found and did not add this book to the database.
5. The employee or manager acknowledges this message.
6. The employee or manager gathers all required information for the database from the book.
7. The employee or manager inputs all required data into the database.
8. The system informs the employee or manager that the book was added to the database.
9. The employee or manager acknowledges the message.
10. The use-case ends.

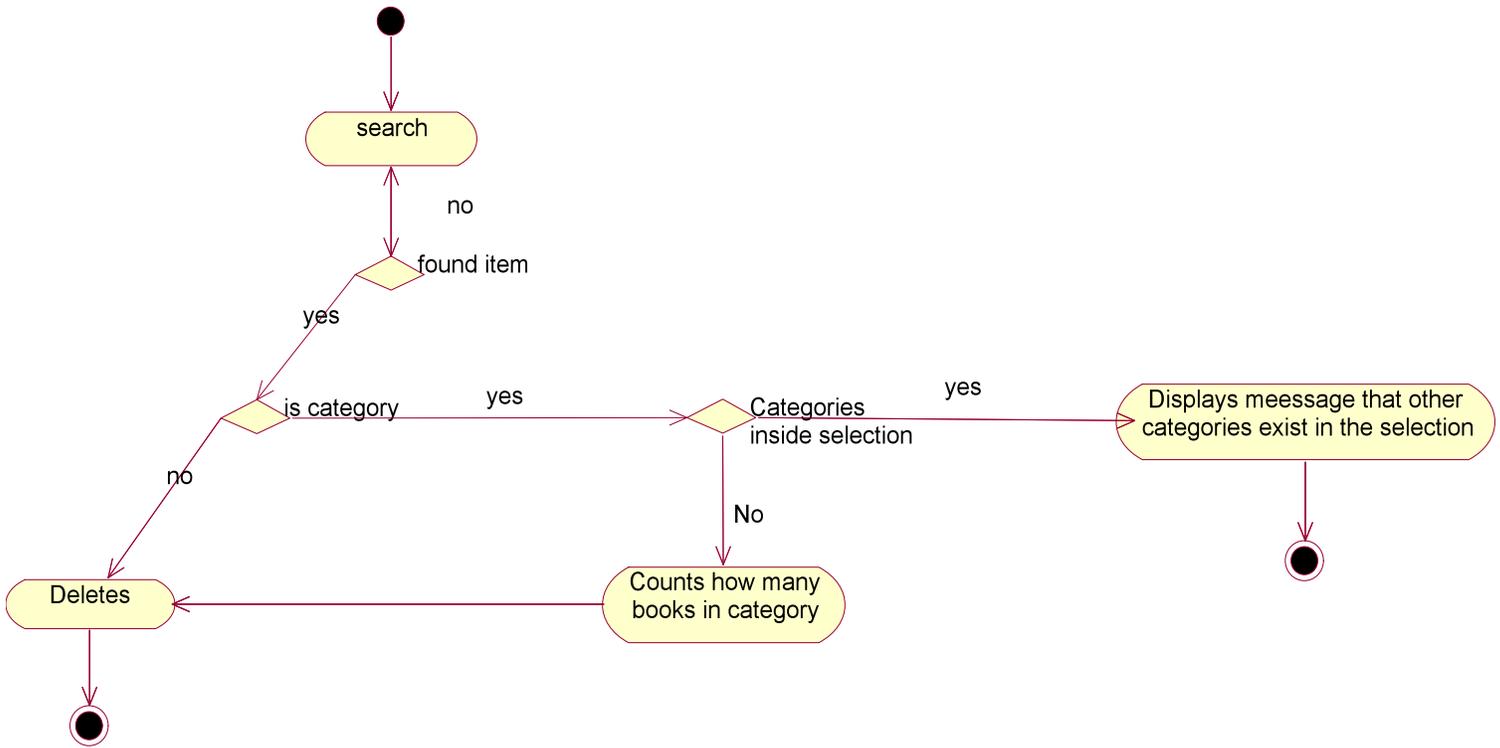


Figure - Add Book/Category Use-Case Diagram

2.1.4.5 Delete Book/Category

- **Brief Description**

This use-case describes the process by which the system deletes a book record in the database. This use-case also describes the process if a manager wants to delete a category of books.

- **Actors**

Managers, Employees

- **Basic Flow of Events: Delete book**

1. The use-case begins when a manager or employee utilizes the search function.
2. The employee or manager employs the search with any of the search choices listed by the system.
3. The employee or manager activates the search as long as they selected a search choice.
4. The system locates and highlights the desired book.

5. The employee or manager deletes the book by clicking delete on the system menu.
6. The system displays a message to the employee or manager to confirm the deletion transaction
7. The use-case ends.

- **Alternate Flow of Events #1: Delete book or category when there is nothing to delete**

1. The use-case begins when a manager or employee utilizes the search function.
2. The employee or manager employs the search with any of the search choices listed by the system.
3. The employee or manager activates the search as long as they selected a search choice.
4. The system displays an error message to the employee or manager that the book or category does not exist.
5. The employee or manager acknowledges this message.
6. The use-case ends.

- **Alternate Flow of Events #2: Delete Category**

This flow of events describes the steps if a manager wants to delete a category

1. The use-case begins when a manager wants to delete a book category.
2. The manager navigates to the category location that is to be deleted.
3. The manager highlights/selects a book category.
4. The manager selects delete on the system menu.
5. The system displays a message to the manager to confirm the deletion transaction.
6. The manager either confirms or denies the transaction.
7. The use-case ends.

- **Alternate Flow of Events #3: Delete Category with books in category**

This flow of events describes the steps if a manager wants to delete a category that contains books

1. The use-case begins when a manager wants to delete a book category.
2. The manager highlights/selects a book category.
3. The manager selects delete on the system menu.
4. The system counts how many books reside in the category that the manager wants to delete.
5. The system displays a message of how many books are in the category.
6. The manager must acknowledge and confirm the amount of books in the category.
7. The system displays a message to the manager to confirm the deletion transaction.
8. The manager either confirms or denies the transaction.
9. The use-case ends.

- **Alternate Flow of Events #4: Delete Category with books and other categories in a category**

This flow of events describes the steps if a manager wants to delete a category that contains books and other categories

1. The use-case begins when a manager wants to delete a book category.
2. The manager highlights/selects a book category.
3. The manager selects delete on the system menu.
4. The system detects if categories reside in the category that the manager wants to delete.
5. The system displays a message that informs the manager that other book categories exist in the category the manager wants to delete.
6. The manager must acknowledge this message from the system.
7. The system displays "Please try again" message when the other categories move to a different location or have been deleted.
8. The manager acknowledges the message.
9. The use-case ends.

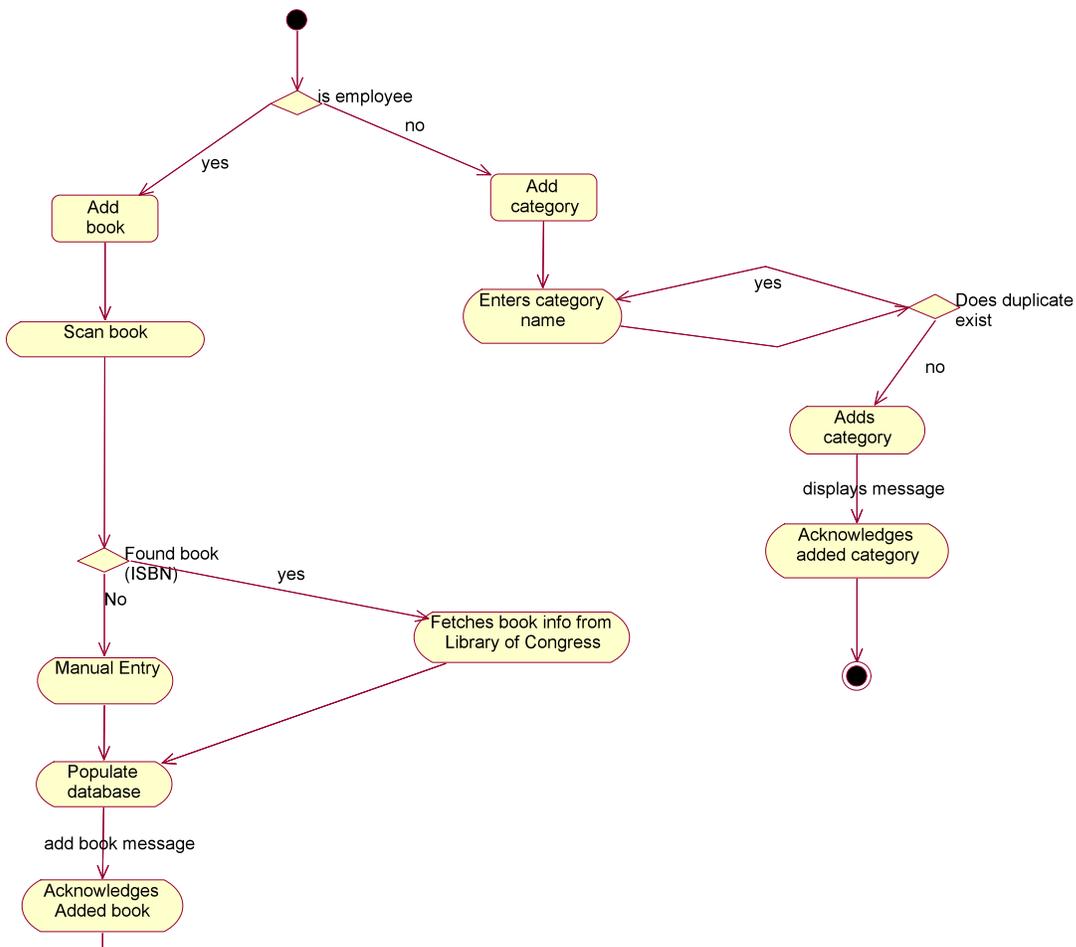


Figure - Delete Book/Category Use-Case Diagram

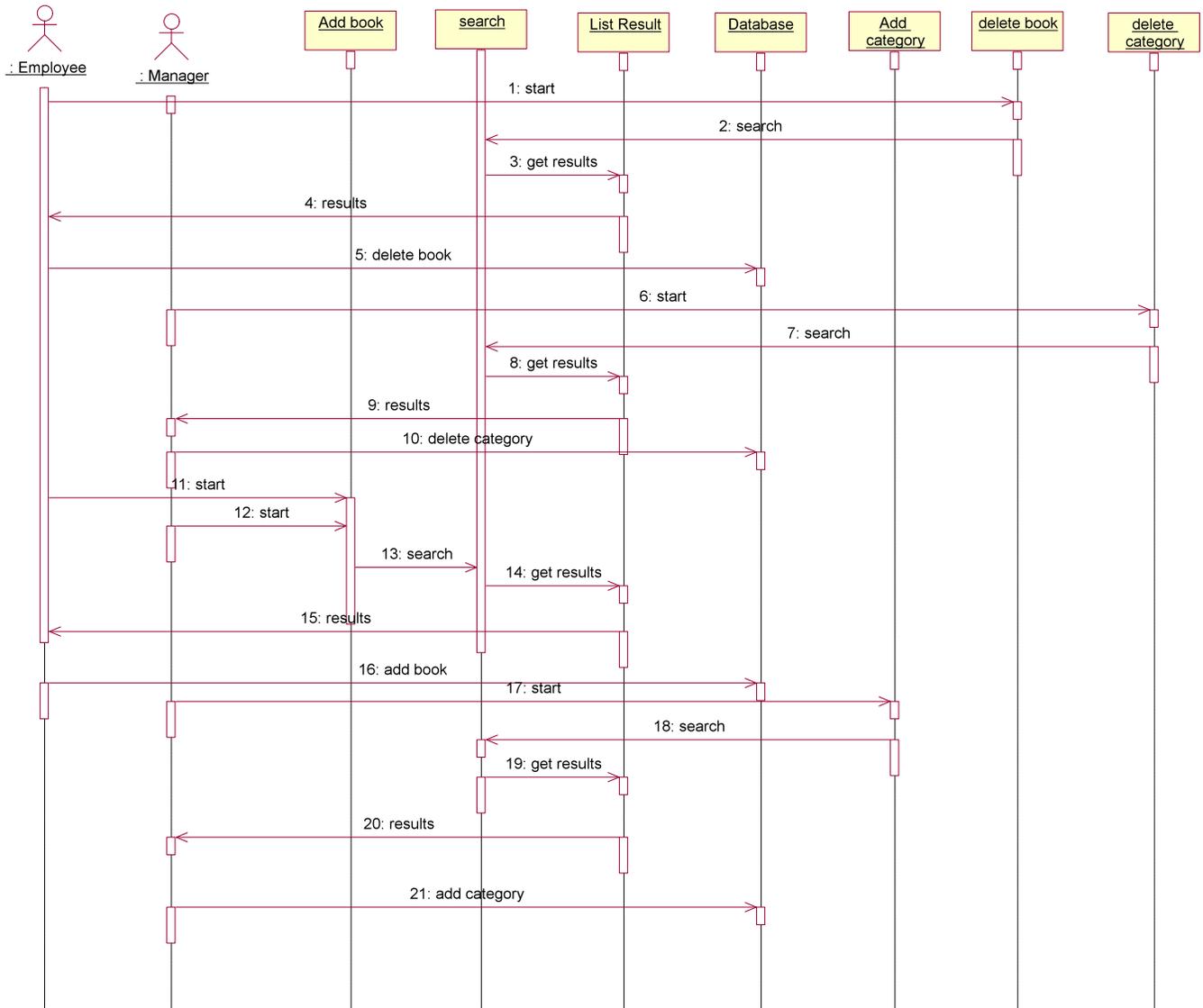


Figure – Manage Books Sequence Diagram

2.1.4.6 Update Quantity

- **Brief Description**

- The purpose of this use-case is for the system to increase or decrease the number of books in the book database

- **Actors**

POS, Book Inventory Database

- **Basic Flow of Events:**

1. This use case begins with the system receiving input from the POS.
2. The system receives the ISBN number of the book sold and a code for refund or sale.
3. The system searches the book database (by ISBN) for the sold book.

4. If the code indicates that the book was sold it decrements the book quantity. If the code indicates a refund, it will increment the book count by one.
5. End of use case.

• **Alternate Flow of Events #1: Attempts to decrease the number of books, when quantity is zero.**

1. This use case begins with the system receiving input from the POS.
2. The system receives the ISBN number of the book sold and a code for refund or sale.
3. The system searches the book database for the specified book.
4. The quantity field is decremented and the book count becomes -1.
5. The system then prompts staff that the quantity is below 0.
6. This use case ends

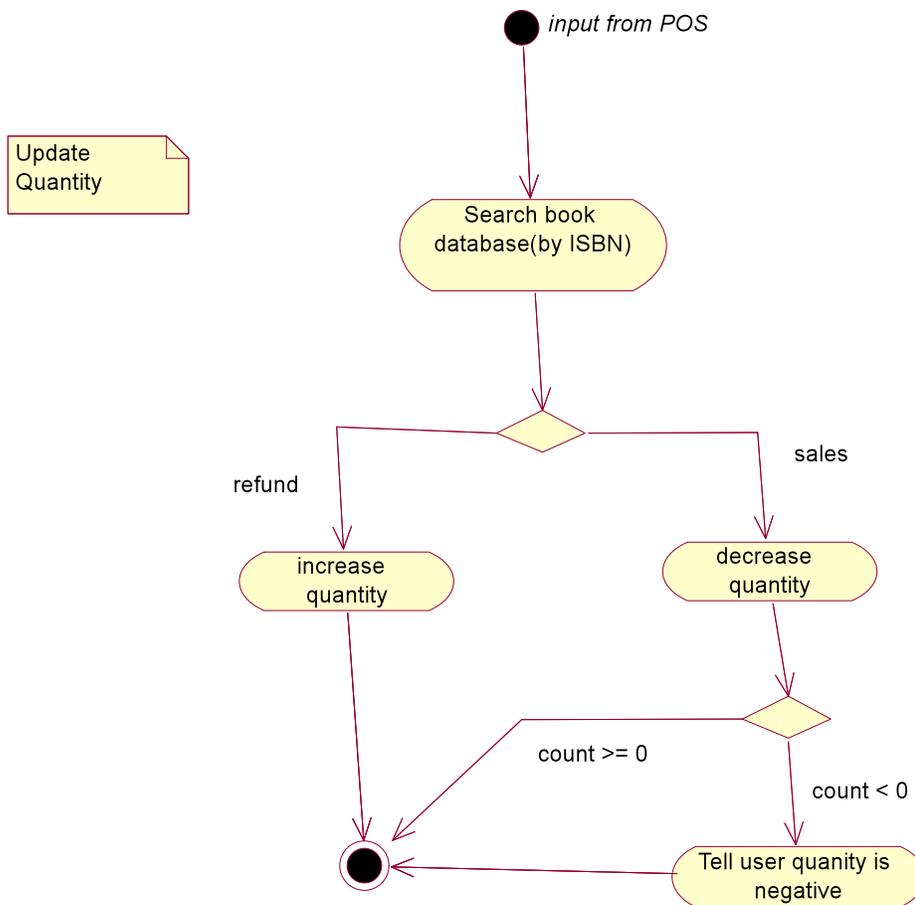
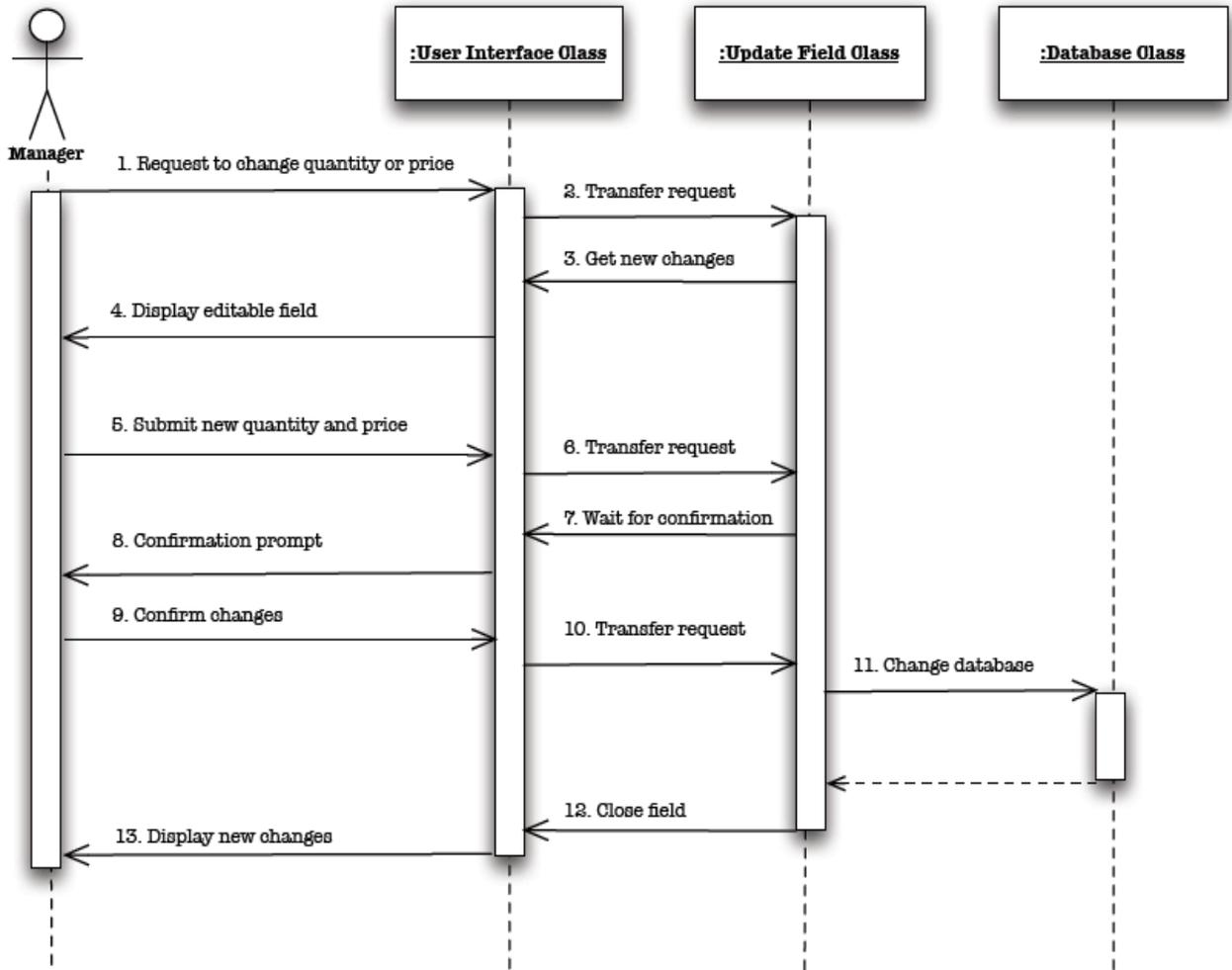


Figure - Update Quantity Use-Case Diagram



Sequence diagram for a manager successfully updating the quantity and price field for a book.

2.1.4.7 Update Customer Info

- **Brief Description**

- The purpose of this use-case is for the system to update the customer information based on the input from the POS

- **Actors**

POS, Customer Info Database

- **Basic Flow of Events:**

1. This use case begins with the system receiving input from the POS.
2. The system receives the customer name and retrieves the customer info from the customer database. If it is a new customer, then the system adds a new entry to the database.
3. The system records the customer's name, address and telephone number into the database.

4. If the customer has rented a book it will also be recorded along with the due date of the book(s) rented.
 5. End of use case
- **Alternate Flow of Events #1: The system does not find a customer in the customer database - adds a new entry. This use case begins with the system receiving input from the POS.**
 1. This use case begins with the system receiving input from the POS.
 2. The system receives the customer name, telephone number, address and searches the customer database for the given customer.
 3. The system does not find the customer.
 4. The system creates a new entry and records the customer details into the database.
 5. This use case ends.
 - **Alternate Flow of Events #2: The system finds a match for a customer by name, but other details are different. Customer responds verifies information. The system updates the customer info.**
 1. This use case begins with the system receiving input from the POS.
 2. The system receives the customer name, telephone number and address. The system searches the customer database by name and finds a match. Other details for that customer are different.
 3. The system sends a prompt to the POS system to determine if the customer's details have changed.
 4. If the POS returns yes, then the details are updated.
 5. The use-case ends
 - **Alternate Flow of Events #3: The system finds a match for a customer by name, but other details are different. Customer is unable to verify the information. New entry is added to the database.**
 1. This use case begins with the system receiving input from the POS.
 2. The system receives the customer name, telephone number and address. The system searches the customer database by name and finds a match. Other details for that customer are different.
 3. The system sends a prompt to the POS system to determine if the customer's details have changed.
 4. The POS returns no, a new entry is added into the customer info database.
 5. This use case ends.

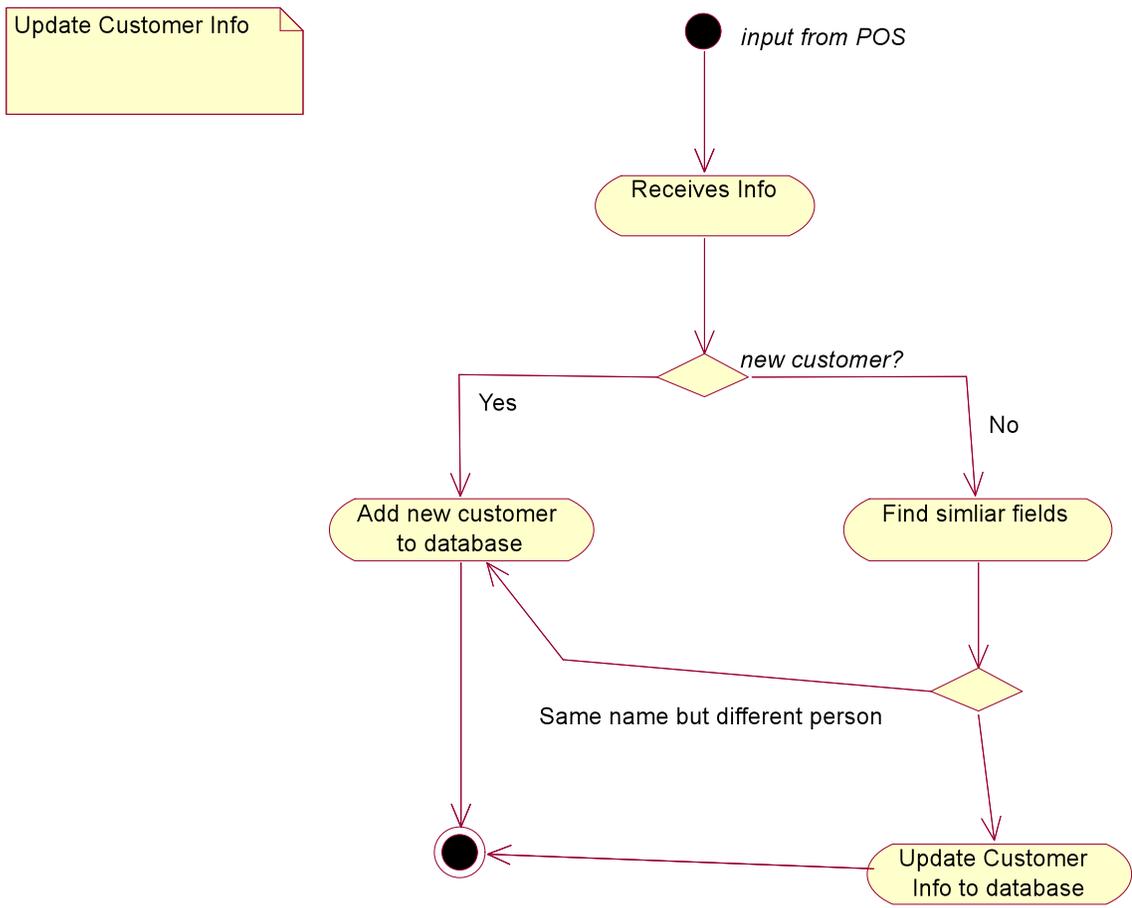


Figure - Update Customer Info Use-Case Diagram

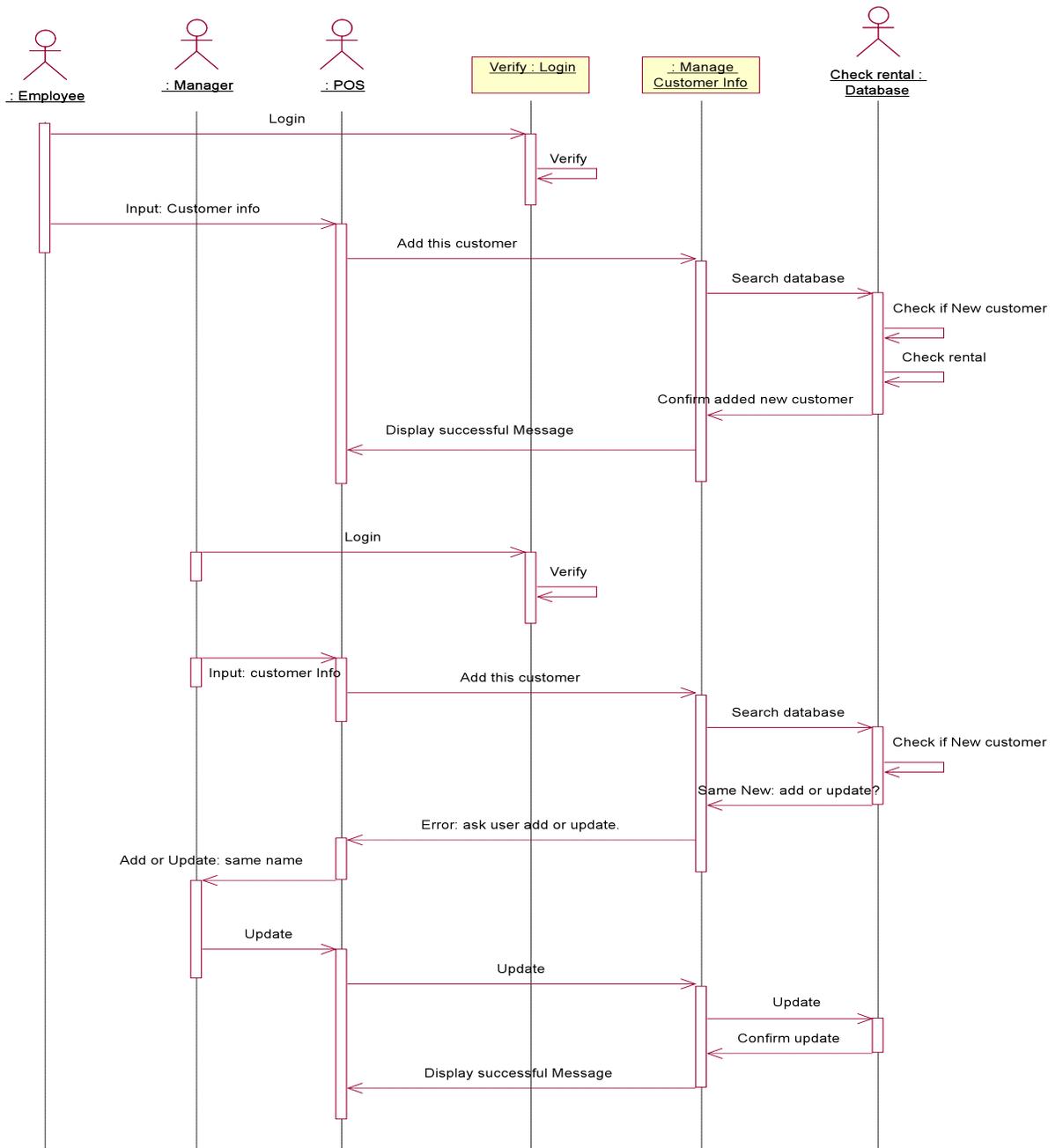


Figure – Manage Customer Info Sequence Diagram

2.1.4.8 Record Transaction into Transaction Database

- **Brief Description**

- In this use case the system records a transaction into the transaction database.

- **Actors**

POS, Transactions Database

- **Basic Flow of Events:**

1. This use case begins with the system receiving input from the POS of a new sale.
2. The system creates a new entry in the database
3. The system records the date, time, transaction ID, name of customer, number of items and purchase amount into the database.
4. End of use case.

- **Alternate Flow of Events: The system receives notification of a transaction – refund**

1. This use case begins when the system receives input from the POS of a refund.
2. The system creates a new entry in the database.
3. The system records the date, time, transaction id, name of customer, number of items and purchase amount to the database.
4. The system sets a flag to record that the transaction was a refund.
5. End of use case.

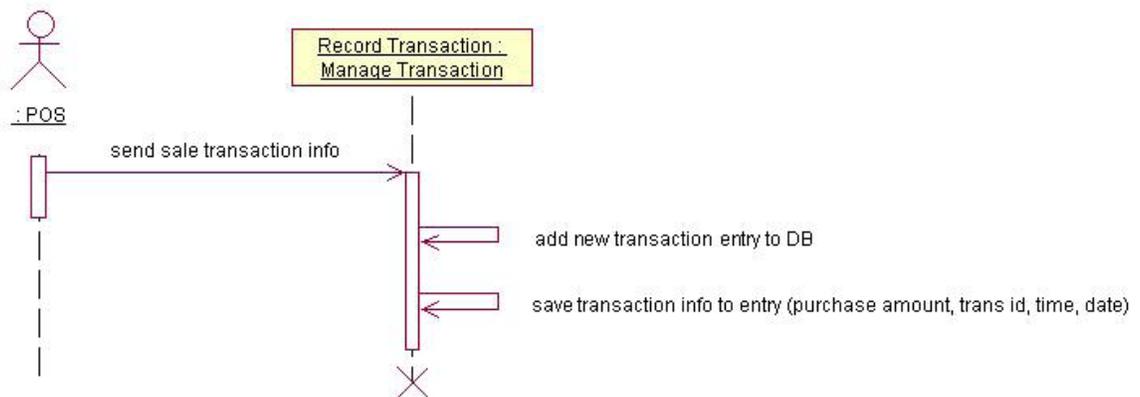


Figure - Record Transaction to Transaction Database Use-Case Diagram

2.1.4.9 Rented Book Returned

- **Brief Description**

- In this use case, the system searches the customer database for all rented books that are due at a specified date. Checks whether the book(s) has been returned.

- **Actors**

Employee, Customer Info Database, Printer

- **Basic Flow of Events:**

1. This use case begins when a staff member requests the report for book rentals due.
2. The system prompts the user for a specified date.
3. The database iterates through the customer info database and looks for a flag that shows a customer has a rented book.
4. The system finds the field and checks whether a rented books is due on the provided date.
5. The system confirms that a book is due on the given date.
6. The system checks whether the customer has returned the book.
7. The system verified that the book(s) have been returned.
8. End of use case.

- **Alternate Flow of Events #1: The system find that a rented book has not been returned.**

1. This use case begins when a staff member requests the report for book rental due.
2. The system prompts the user for a date.
3. The system iterates through the customer database and looks for a rented books flag.
4. The system finds the rented books flag and checks whether the given date matches the due date.
5. The system confirms that a books is due on the given date.
6. The system checks whether the book(s) has been returned.
7. The system is unable to confirm that the book has been returned.
8. The system prints out the customer information.
9. This use case ends.

- **Alternate Flow of Events #2: The system finds that a customer does not have book rental due**

1. This use case begins when a staff member request the report for book rentals due.
2. The staff member enters a specific date to check.
3. The system iterates through the customer info database and looks for a “Rented Books” flag.
4. The system does not find the rented books flag for the customer.
5. The system continues to the next customer in the database.
6. This use case end

- **Alternate Flow of Events #3: The system finds that a customer has a rented book (or rented books) in their possession, but they are not due at the specified date.**

1. This use case begins when a staff member requests the report for book rentals due.
2. The system prompts the user for a specific date.

3. The system iterates through the customer info database and looks for a flag that show a customer has a rented book.
4. The system finds the flag and compares the specified date with the date in the database.
5. The system determines that they are not the same.
6. The system moves on to the next customer.
7. This use case ends.

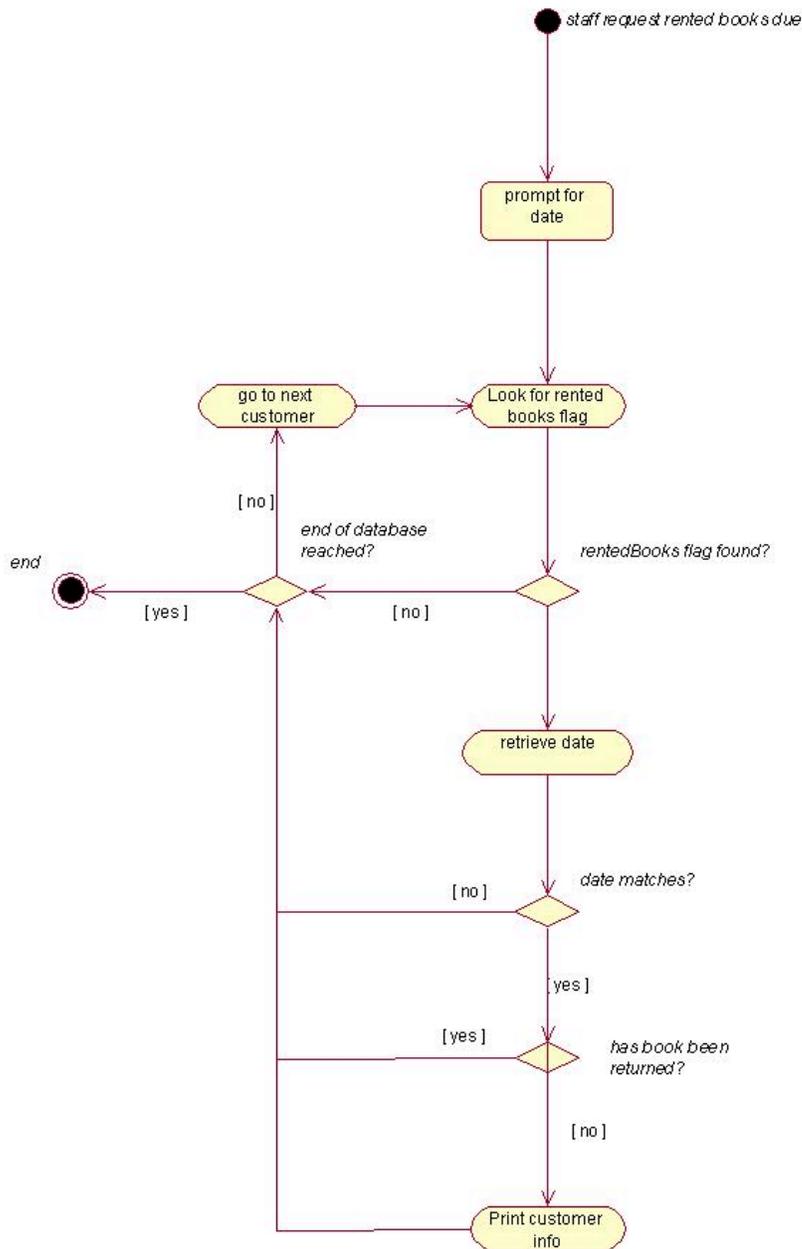


Figure - Rented Book Returned Use-Case Diagram

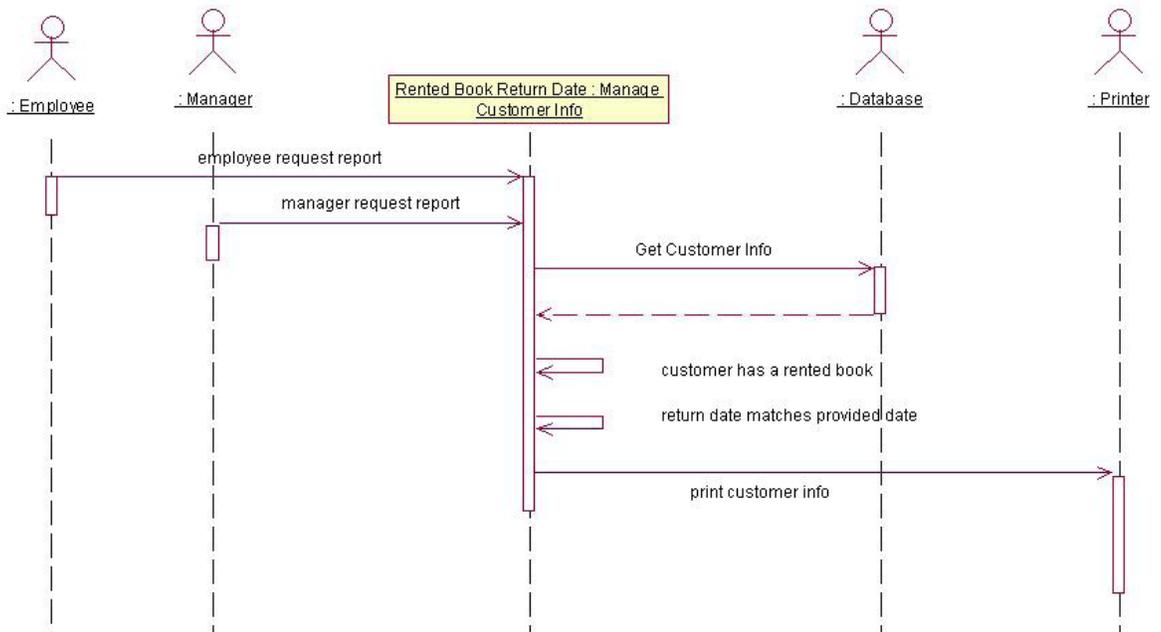


Figure – Rented Book Returned Sequence Diagram

2.1.4.10 Return Rental

- **Brief Description**

- In this use case the system updates the customer information in the database after a customer return a book from the POS

- **Actors**

POS, Customer Info Database

- **Basic Flow of Events:**

1. This use case begins with the system receiving input from the POS.
2. The system receives the customer name and book returned, and then retrieves the customer info from the customer database.
3. The system verifies if book returned is the same book in record.
4. The system confirms the customer has returned the book.
5. End of use case.

- **Alternate Flow of Events #1: The system does not find the rental book in the customer info database.**

1. This use case begins with the system receiving input from the POS.
2. The system receives the customer name and book returned, and then retrieves the customer info from the customer database.
3. The system could not verify the book returned is the same book in record.
4. The system displays error at the interface to prompt the user.
5. This use case ends.

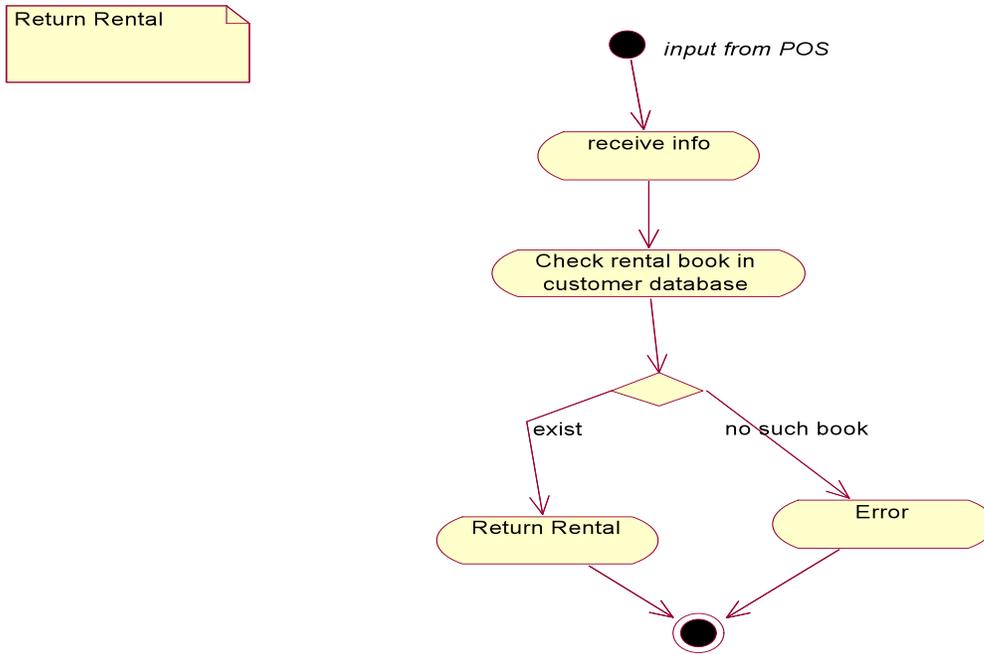


Figure - Return Rental Use-Case Diagram

2.2 Assumptions and Dependencies

None

3. Specific Requirements

3.1 Functionality

The system shall be designed such that a user can query information stored in the databases. For example, query a book by author, title, ISBN, publisher, etc.

The system shall be designed for easy navigation. Also each separate database can be modified to tailor a client's needs. The system shall be able to provide all the system features as outline in previous sections.

3.2 Usability

The system shall enable the users to navigate and perform operations in an intuitive and easy manner. The user interface must allow the user to search for the books in a timely manner. Also, vendors must access the system in a secure manner and the book information. However, they

should have this capability during the business day but not affect the business operations of the bookstore.

3.3 Reliability

- The system shall be designed to be robust and be able to handle a large amount of traffic.
- The system must be able to handle improper user input.
- The system shall be designed in a manner that is free of security flaws.

3.4 Performance

The system shall be designed to handle a large amount of traffic.

3.5 Supportability

The system shall be designed for easy maintenance and upgradability. The system will be designed to handle a large and diverse load at peak times at the start of each semester. The system must continue operating at a constant pace with a maximum load.

3.6 Design Constraints

None

3.7 Online User Documentation and Help System Requirements

3.7.1 User Manual

A user manual will provided as a pdf document to staff. It will outline, in detail, the system's functions.

3.7.2 Online Help

System documentation will also be available online, it will include the user manual, FAQ section and an email support system.

3.7.3 Installation, Guides, Configuration and Read Me File

The installation guide and configuration guidelines will be provided as part of the Read Me file which is included as a .txt file.

3.7.4 Labeling and Packaging

The software will be packaged for the unix environment running Apache, SQL and PHP. Labeling will include all icons, copyright information and any applicable trademarks.

3.7.5 Environmental Requirements

The environment of the system is in the student book store at room temperature and is to be self-contained.

3.7.6 Platform Requirements

In order to maintain easy upgradability and robustness, the software product will be developed for the UNIX platform.

3.7.7 System Requirements

The product must have an interface with the printer/printers of the store in order to print the necessary report.

3.8 Purchased Components

None

3.9 Interfaces

3.9.1 User Interfaces

The actors employee, manager, customer, and vendor will each have an interface platform to the software product. The platform for manager and vendor will be protected by login and password since the platform allows the user to make some critical changes that can have propagating effects in the system. Customer and employee platforms don't make such critical changes, so they're not protected by login and password. If a customer decides to make change to personal information, then this platform will be protected by login and password.

3.9.2 Hardware Interfaces

This product will have an interface with some external devices in order to have a complete function as designed. The bookstore sale transaction devices (such as registers and scanners) will need to have an interface to record transactions to the transaction database and update book quantity in the book database. These devices are represented in the system as an actor called "*POS*".

The product will also interface with a local printer to print the daily and requested reports prints. This device is represented in the system as an actor called "*Printer*".

3.9.3 Software Interfaces

None

3.9.4 Communications Interfaces

None

3.10 Licensing Requirements

None

3.11 Legal, Copyright and Other Notices

None

3.12 Applicable Standards

TCP/IP, UL and ISBN standard.

4. Product Acceptance Criteria

4.1 Functionality available in version 2.0

None