# Library System

## Description

You have been asked to build an application in Java that manages the library system. The library consists of books that can be borrowed by members. The system will track book availability, manage borrow and return transactions, calculate late fees, and also track which member currently has which book.

## Entities and Attributes

|  |  |
| --- | --- |
| Library | Manages the collection of books and members. It also tracks borrow and return transactions. |
| Book | Each book has a title, author, ISBN, status (available/borrowed), and publication year. |
| Member | Each member has a unique ID, name, and a list of borrowed books. |
| BorrowTransaction | A transaction that tracks when a book is borrowed, its due date, and whether it's returned on time. |
| Librarian | Manages the overall system, adds and removes books, and assists with member registration. |

## Book Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TITLE** | **AUTHOR** | **ISBN** | **STATUS** | **PUBLICATION YEAR** |
| "1984" | George Orwell | 123-456789 | Available | 1949 |
| "The Hobbit" | J.R.R. Tolkien | 987-654321 | Borrowed | 1937 |
| "Harry Potter" | J.K. Rowling | 456-789123 | Available | 1997 |

## Assignment 1

Build an application that tracks the process of borrowing and returning books.

**The application should be able to:**

* Allow members to borrow available books.
* Keep track of return dates and late fees.
* Show a list of currently available books and borrowed books.

## Assignment 2

**Create a method that handles overdue books:**

* If a member has a book past its due date, the system should calculate the fine (€0.50 per day late).
* Create a method that returns the amount due for overdue books.

## Assignment 3

**Handle member borrowing limits:** A member can borrow a maximum of 3 books at a time. If a member tries to borrow more than 3 books, throw a TooManyBooksException.

## Assignment 4

**Create a book reservation system:** A member can reserve a book that is currently borrowed by another member. Once the book is returned, the reservation system should notify the member.

## Assignment 5

**Write Unit tests:**

* Write tests for the book borrowing process, including the handling of overdue books and borrowing limits.
* Test the reservation system functionality.

Book Shop System  
Description

You have been asked to build an application in Java for managing a bookshop. The system will manage the inventory of books, customer orders, and payments. Books can be added or removed from the inventory, and customers can place orders online or in-store.

## Entities and Attributes

|  |  |
| --- | --- |
| Bookshop | Manages the inventory of books and customer orders. |
| Book | Each book has a title, author, price, and stock availability. |
| Customer | Each customer has a name and a unique ID. They can place orders and track their purchases. |
| Order | An order tracks the list of books purchased by a customer, the total price, and the order status (pending/fulfilled). |

## Book Details

|  |  |  |  |
| --- | --- | --- | --- |
| **TITLE** | **AUTHOR** | **PRICE (€)** | **STOCK** |
| "1984" | George Orwell | 15 | 10 |
| "The Hobbit" | J.R.R. Tolkien | 25 | 5 |
| "Harry Potter" | J.K. Rowling | 20 | 0 |

## Assignment 1

## **Create a method for processing customer orders:**

## The system should check the availability of books and update stock once an order is placed.

## If a book is out of stock, the order cannot be fulfilled, and an error message should be displayed.

## Assignment 2

**Create a method to manage discounts:**

## Books on promotion have a discount. Create a method that applies discounts to certain books during checkout.

## The discount percentage is specified when the promotion is created.

## Assignment 3

**Handle out-of-stock scenarios:** If a book is not available, it should not be added to the order. If a customer tries to order a book with zero stock, throw an OutOfStockException.

## Assignment 4

## **Create a payment processing system:**

## After an order is placed, the system should process the payment.

## The payment method can be cash, card, or online payment. Implement different transaction methods for each.

## Assignment 5

**Write Unit tests:**

* Test the order placement functionality, including stock management.
* Test the discount application on books and the payment system.