

Library Management System Report

This report provides an overview of the implementation, challenges faced, and additional features included in the library management system project.

Objective

To design and implement a simplified library management system using Python, demonstrating understanding of OOP concepts such as classes, objects, encapsulation, abstraction, and inheritance.

Overview

The system should allow for adding books to the library, checking out books, returning books, and tracking the availability of books. This project will be broken down into several components, each focusing on different OOP concepts

Implementation Details

❖ Book Class

- The 'Book' class represents a book in the library and includes attributes such as title, author, and ISBN.
- The availability attribute is encapsulated as a private attribute with getter and setter methods.
- The class includes a method to display book information.

❖ Library Class

- The 'Library' class manages a collection of books and stores them in a list.
- It provides methods to add a book, remove a book, and list all books in the library.
- The class includes a method to search for a book by title or author.

❖ Borrowing System

- The borrowing system is part of the 'Library' class and allows users to check out and return books.
- Checking out a book updates the book's availability, and error checking prevents borrowing unavailable books.

❖ DigitalBook Class

- The 'DigitalBook' class inherits from the 'Book' class and includes additional attributes such as file size and format.
- The class includes a method to simulate downloading the digital book by printing a message to the console.

❖ User Interaction

- The script includes a simple text-based user interface for interacting with the library management system.
- Users can add books, list all books, borrow and return books, and search for books through input prompts.

Challenges and Solutions

- ✓ **User Interface Design:** Creating a simple and effective text-based interface was a challenge. Iterating on the design and testing different scenarios helped improve the user experience.
- ✓ **Error Handling:** Handling errors such as attempting to borrow an unavailable book was addressed by adding appropriate checks and feedback.

Additional Features

- ✓ **Search Functionality:** Added a method to search for books by title or author, enhancing the usability of the system.
- ✓ **Download Method for Digital Books :** Included a method in the 'DigitalBook' class to simulate downloading digital books, adding an extra layer of functionality.

This report summarizes the key aspects of the library management system implementation and provides insights into the challenges faced and the additional features added to enhance the system.