Book Store Management System: Project Documentation and Design Plan:

Table of Contents

1. Introduction

The Book Store Management System is a Command Line Interface (CLI) application designed to manage a bookstore's inventory. This project leverages Python's standard libraries to provide essential functionalities such as adding, viewing, searching, and removing books. The application ensures data persistence by saving book information to a file and loading it upon startup.

2. Project Overview

The primary goal of this project is to create a user-friendly and efficient bookstore management system that operates entirely within the terminal. The application is designed to be modular, with each feature encapsulated in separate functions or modules. This approach enhances code maintainability and reusability.

3. Features

- **Add Books**: Users can add new books with details such as title, author, ISBN, genre, price, and quantity in stock.
- **Prevent Duplicate Books**: The system ensures that no two books share the same ISBN.
- **View Books**: Displays all saved books in a well-organized format.
- **Save to File**: Automatically saves book information to a file upon addition or modification.
- **Load from File**: Loads previously saved book data when the program starts.
- **Remove Books**: Allows users to delete books by their ISBN.
- **Error Handling**: Provides meaningful error messages for invalid inputs.
- **Menu System**: An interactive menu for easy navigation.

4. File Structure

The project is organized into multiple Python files, each dedicated to specific functionalities:

```
Book-Store-Management-System/
|
├── Design Plan/
|   ├── plan_and_design.pdf      # Detailed plan and design documentation
|   ├── use_case_diagram.png     # Visual representation of use cases
|   ├── uml_diagram.png          # UML diagram illustrating class relationships
|   └── visual_representation.png # Additional visual representations
|
├── book.py                  # Contains the definition and management of book objects
├── book_manager.py          # Handles operations related to managing books
├── books.json               # JSON file storing the book inventory data
├── file_handler.py          # Utility functions for handling file operations
└── main.py                  # Main entry point of the application
```

5. Application Design

The application design is illustrated in the **Design Plan** directory, which includes detailed documentation and visual representations.

6. Modules and Functions

main.py

- **main_menu()**: Displays the main menu and handles user interactions.

book.py

- **Book()**: Class definition for book objects.
- **create_book()**: Creates a new book object.

book_manager.py

- **add_book()**: Adds a new book to the inventory.
- **view_books()**: Displays all books in the inventory.
- **search_book()**: Searches for a book by ISBN.
- **remove_book()**: Removes a book by ISBN.

file_handler.py

- **load_books()**: Loads book data from the JSON file.
- **save_books()**: Saves book data to the JSON file.

7. Error Handling

The application includes robust error handling to ensure meaningful error messages are displayed for invalid inputs. For example:

- **Invalid ISBN**: Prompts the user to enter a valid ISBN.
- **Negative Price or Quantity**: Informs the user that the price must be positive and the quantity must be non-negative.
- **File Not Found**: Handles the case where the book data file does not exist and prompts the user to start fresh.

8. Usage Instructions

1. **Start the Program**: Run **main.py** to start the application.

2. **Main Menu**: Use the main menu to navigate through the available options.
3. **Add Book**: Select option 1 to add a new book.
4. **View Books**: Select option 2 to view the list of books.
5. **Search Book**: Select option 3 to search for a book by ISBN.
6. **Remove Book**: Select option 4 to remove a book by ISBN.
7. **Exit**: Select option 5 to save all data and exit the program.

9. Future Enhancements

- **User Authentication**: Implement user authentication to restrict access to authorized users.
- **Advanced Search**: Add advanced search functionality to search books by title, author, or genre.
- **Inventory Management**: Enhance inventory management features to track book sales and restocking.
- **Graphical User Interface (GUI)**: Develop a GUI version of the application for better user experience.

10. Conclusion

The Book Store Management System is a comprehensive CLI application designed to manage a bookstore's inventory efficiently. The modular design and robust error handling ensure the application is maintainable and user-friendly. This project demonstrates the application of Python concepts learned throughout the course and provides a solid foundation for further enhancements.

Submitted by: Shakil Ahamed Riaz

Date: 06-03-2025

GitHub Repository: https://github.com/shakilahamedriaz/Book-Store-Management-System