Smart Library Management System

Table of Contents

Introduction

The **Smart Library Management System (SLMS)** is an object-oriented Python application designed to automate and streamline library operations. It helps manage book inventories, user accounts, borrowing and returning of books, and sends notifications for due or overdue books.

Features

- Add, edit, and remove books with detailed metadata
- Register and manage users (Members and Librarians)
- Issue and return books with due date tracking
- Search for books by title, author, or category
- Notifications for upcoming due dates and overdue returns
- Generate reports on book circulation and user activity
- Role-based access control (Librarians vs. Members)

System Architecture

The system follows an **Object-Oriented Programming (OOP)** design with the following core components:

- **Book:** Represents individual book records.
- **User:** Base class for all users; extended by Member and Librarian classes.
- **Library:** Manages collections of books and users, and handles transactions.
- **NotificationManager:** Sends reminders and alerts.
- **ReportGenerator:** Generates summary reports for administrative purposes.

Classes and Design

Class	Responsibility	Key Methods
`Book`	Holds book data and availability status	`update_info()`, `is_available()`

Class	Responsibility	Key Methods
`User`	Base class with user profile info	`update_profile()`, `authenticate()`
`Member` (inherits `User`)	Library members who borrow books	`borrow_book()`, `return_book()`
`Librarian` (inherits `User`)	Manage books, users, and library operations	`add_book()`, `remove_book()`, `manage_users()`
`Library`	Central system to manage books, users, and transactions	`add_user()`, `issue_book()`, `search_books()`
`NotificationManager`	Handles sending notifications and reminders	`send_due_reminder()`, `send_overdue_alert()`
`ReportGenerator`	Creates reports on library usage and inventory	`generate_report()`

Installation

1. Make sure you have Python installed in your computer.
2. Clone the repository:

   ```
   git clone https://github.com/App-Factory-USIU/smart-library-management.git
   cd smart-library-management
   ```

3. Create and activate a virtual environment:

   ```
   python -m venv env
   source env\Scripts\activate    # On Linux or MacOS use `env/bin/activate`
   ```

4. Install required dependencies (if any):

   ```
   pip install -r requirements.txt
   ```

Usage

Run the main program:

```
python main.py
```

Documented By: Alice Jeremoki