Smart Library Management System

# Table of Contents

 Smart Library Management System  Table of Contents

 Introduction  Features

 System Architecture  Classes and Design  Installation

 Usage

# 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**

update\_profile(),

User Base class with user profile info

authenticate()

Member (inherits User) Library members who borrow books borrow\_book(), return\_book()

Librarian (inherits

User) Library

NotificationManager

Manage books, users, and library operations

Central system to manage books, users, and transactions

Handles sending notifications and reminders

Creates reports on library usage and

add\_book(), remove\_book(), manage\_users()

add\_user(), issue\_book(), search\_books()

send\_due\_reminder(), send\_overdue\_alert()

ReportGenerator

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

1. Create and activate a virtual environment:

python -m venv env

source env\Scripts\activate # On Linux or MacOS use `env/bin/activate`

1. Install required dependencies (if any):

pip install -r requirements.txt

# Usage

Run the main program:

python main.py

*Documented By: Alice Jeremoki*