

DESIGN DOCUMENT

1. Introduction

The Library Management System is designed to facilitate the efficient management of books, borrowers, loans, and fines in a library setting. The system employs a client-server architecture, utilizing HTML, CSS, Python Flask, and MySQL technologies.

2. System Architecture

2.1 Frontend

The frontend is implemented using HTML and CSS for the graphical user interface. Flask templates are used to dynamically generate web pages based on user interactions. The interface includes search functionality, book checkout, borrower management, and fine handling.

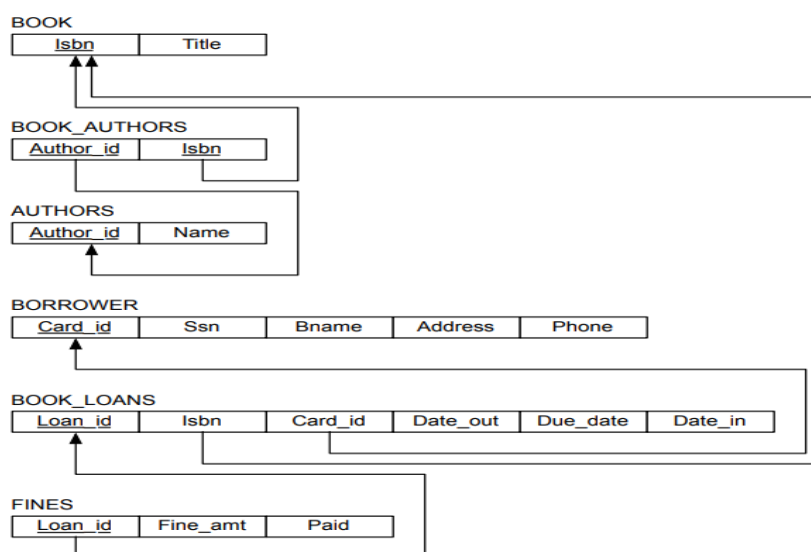
2.2 Backend

The backend is powered by Python Flask, providing a web server and handling user requests. The Flask application interfaces with the MySQL database using the Python MySQL Connector. This allows seamless communication between the user interface and the database for performing CRUD (Create, Read, Update, Delete) operations.

2.3 Database

The MySQL database stores information about books, borrowers, book loans, and fines. The schema includes tables such as BOOKS, AUTHORS, BOOK_AUTHORS, BORROWERS, BOOK_LOANS, and FINES, with appropriate relationships established through foreign keys.

Database schema :



2.4 Libraries and Tools

Flask: Web framework for building the application.

Python MySQL Connector: Facilitates communication between Python and MySQL.

HTML and CSS: Used for frontend development.

3. Functionalities Implementation

3.1 Book Search and Availability (Search books)

The system supports searching for books based on ISBN. The search interface is implemented using a single text field, providing case-insensitive substring matching.

3.2 Book Loans

Checking Out Books: (check-out)

1. Librarians can select books from search results or provide an ISBN directly.
2. Borrower information is collected, and a new entry is created in the BOOK_LOANS table.
3. Constraints ensure a maximum of 3 active loans per borrower and prevent checkout if a book is already checked out.

Checking In Books:

1. Librarians can locate loan records by searching on ISBN, card_no, or borrower name substring.
2. Selected entries can be checked in, updating the date_in in the corresponding BOOK_LOANS tuple.

3.3 Borrower Management

Librarians can create new borrower accounts, ensuring that all required attributes (name, SSN, address) are provided. The system generates unique card_no values for each new borrower, and duplicate SSNs are rejected.

3.4) Fines (Check fines, update fines, pay fines)

1. Fines are automatically assessed at a rate of \$0.25 per day.
2. Librarians can update/refresh entries in the FINES table.
3. Fines are calculated based on returned and late books.

4. GUI Design

The graphical user interface is designed for intuitive usability, featuring search fields, buttons, and menu items for various functionalities.

5. Assumptions and Constraints

The database connection is assumed to be secure and properly configured.

All users are expected to have basic knowledge of library operations.