DESIGN DOCUMENT

Architecture:

I have used Django(Python) web programming framework.

A Django project builds a website using apps as components. Django is modeled around a *Model-View-Controller* (MVC) framework.
The model, which provides the interface with the database containing the application data.
The view, which decides what information to display to the user and collects information from the user.
The controller, which manages the logic for the application and acts as an interface between the model and the view.

Django uses slightly different terminology in its implementation of MVC - *Model-Template-View* (MTV)

1. Models(model.py)
The model layer in Django contains the database and the Python code that directly uses it. Django helps you write models that connect to the database tables. The default database in Django is sqlLite. I have used the MySQL database in my project.
From a model, we can query and update its objects.

2. Templates
The template is the interface between the user and your Django application. It decides how the data is presented.

3. Views(View.py)
The view layer handles the logic for the application. It contains the web HTML pages and the Python code that is used to render these pages.. A view sources data from your database and delivers it to a template. It can also collect information from the user.

Data Design:

I have used a python script to create an sql file containing commands to feed data into the MySQL database. It also normalizes the data provided.

Procedural Design:

When a user opens the library management system, he can search for a book. The value entered to search a book is sent to views.py and queries are written for the database. The results are then returned to the user.

Interface Design:

The interface is designed using html, css and javascript frameworks.

Assumptions:

1. The users of the system are Librarians and not book borrowers.

2. There is only one copy of each book.