

# Library Management System

## Design Document (Milestone-1)

Name: Vaibhav Tyagi (VXT200018)

Course: CS 6360 Database Design

## **Introduction:**

Many individuals visit libraries every day to read books. Libraries are a crucial component of all colleges and universities and are found in each of them. Numerous individuals visit libraries to read and borrow books in order to learn new things. Therefore, it would be practically difficult for an individual to keep track of all the transactions due to the sheer number of books and individuals. Therefore, this application is made to help developers handle all of these aspects much more effectively.

## **Design Overview:**

This application is a website that can operate on local web server within a library. The front-end of the website is created using HTML and CSS along with Bootstrap and JavaScript. The backend is built in python using Django which is a MVC web framework. MySQL database is used as a DBMS.

## **Programming Languages and Libraries used:**

Frontend: HTML, CSS, Bootstrap, JavaScript

Backend: Django, PyMySQL, mysqlclient

Database: MySQL

## **Database design:**

With a few simple changes, the schema is same as what is described in the description. To track the availability of books, the "Book" schema has an additional field called "Available." It should be noted that this could be accomplished without the inclusion of this column by determining if the value of the column "Date\_In" in the "Book\_Loans" table is NULL, but doing so would result in an additional 2 queries that would need to be executed. Further since it is not included in the Schema or requirement specifications, the ISBN13 information is not used.

## **Architecture:**

Django is a MVC architecture - Your data is modeled or represented by the model(M). It is an interface to the data, not the data itself. You may use the model to extract data from your database without being familiar with its technical details. In order to apply the same model across different databases, the model often includes an abstraction layer with your database.

What you see is the view(V). It's your model's presentation layer. The view on your computer is what you see in a browser when using a Web app or the user interface when using a desktop software. The view offers a user input interface as well.

The information that moves from the model to the view is managed by the controller (C). It sets the rules as to what data is provided to the view and what data is fetched from the

database via the model using programmed logic. Additionally, it implements business logic by either updating the view, altering data through the model, or doing both.