

# Design Document

## Architecture:

Framework used: Django (Python).

Django uses MVC framework as *Model-Template-View* (MTV) during implementation.

### 1. Models (models.py)

Django helps us to write models that connect to the database tables. From a model, we can query and update its objects.

### 2. Templates

It is an interface between the user and your Django application. It decides how the data is presented.

### 3. Views (views.py)

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

## Design decisions:

Python script creates a 'library.sql' file containing SQL commands (that dumps data into the MySQL database 'library'), and normalize the data.

Example: When searching for a book:

When a user runs the project, he can search for a book. The value entered to search a book is sent to views.py and the database is then queried. The results are then returned to the user.

The interface is designed using HTML, CSS and JavaScript.

## Assumptions:

1. The users of the system are Librarians and not book borrowers.
2. There is only one copy of each book in the library.