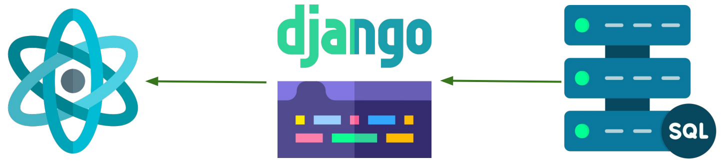
Technical Presentation BookShop

Challenge

Giuseppe Calcagno

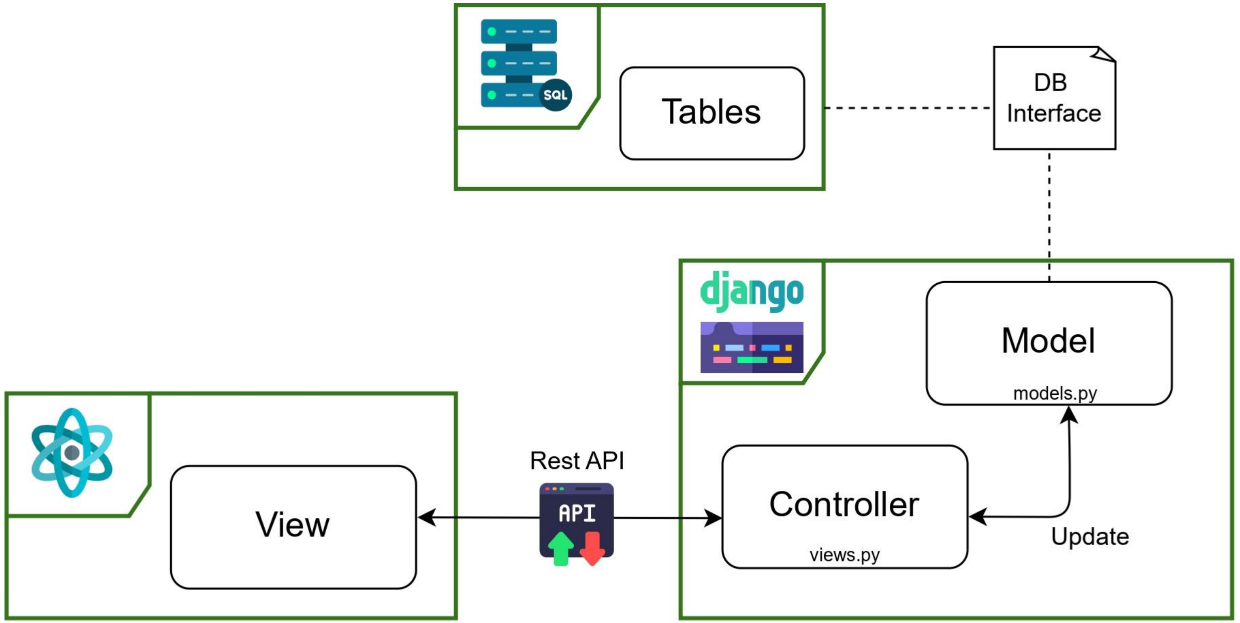
Proposed Solution



Technologies Used

- **DBMS: sqlite3**
  IT is a lightweight, embedded database. It is cross-platform, ACID-compliant, and ideal for small to medium-sized applications.
- **Back-End: Django**
  It is a high-level web framework for Python,
- **Front-End: React Js**
  React.js is a powerful JavaScript library for building dynamic component-based and interactive user interfaces.

Proposed Solution - Architecture



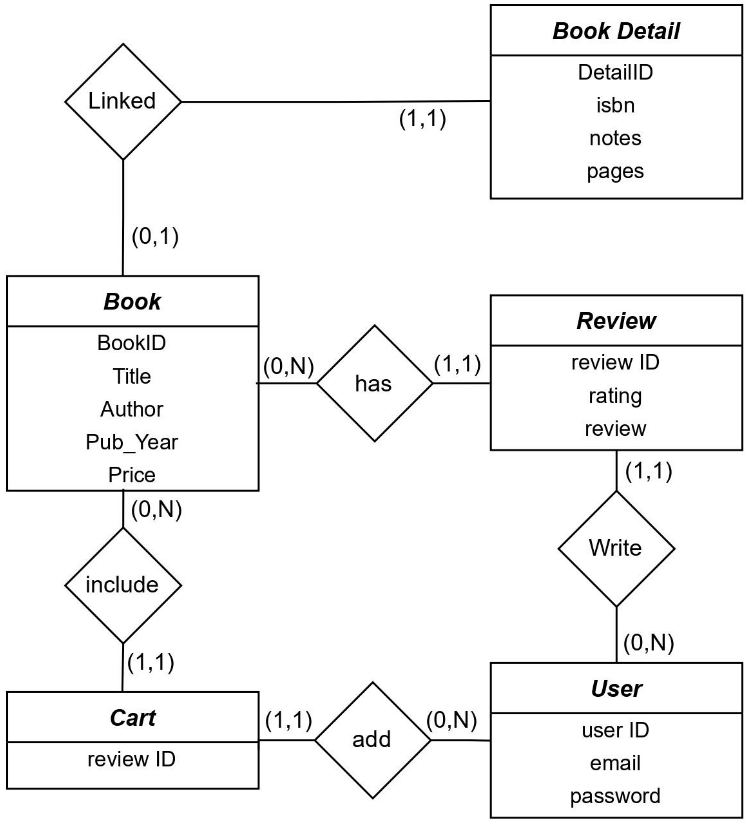
MVC Patter

Model:
Manages data and business logic

Controller:
Handles user input and updates the Model and View accordingly

View:
Deals with the presentation and user interface

Proposed Solution - BD



Relevant Design Choice

- **Cart Entity**
  This entity is designed for the simple addition of multiple instances of the same book to the cart. Nonetheless, this design is future-proof, allowing the addition of other attributes (such as the adding date) to the entity.

- **Book Detail Entity**
  This entity is not included as attributes in the book for straightforward future management. In the database, necessary book information is separated from additional details. This prevents numerous null entries and enables agile modifications of additional information

Proposed Solution - BackEnd

REST API Overview

- **Book API**
  GET: Retrieves a list of books and filter them based on pars..
  POST: Adds a new book.
  PUT: Updates an existing book.
  DELETE: Deletes a book.

- **Book Detail API**
  GET: Retrieves detailed information about a specific book.
  POST: Adds a description for a book.
  PUT: Updates the description of a book.

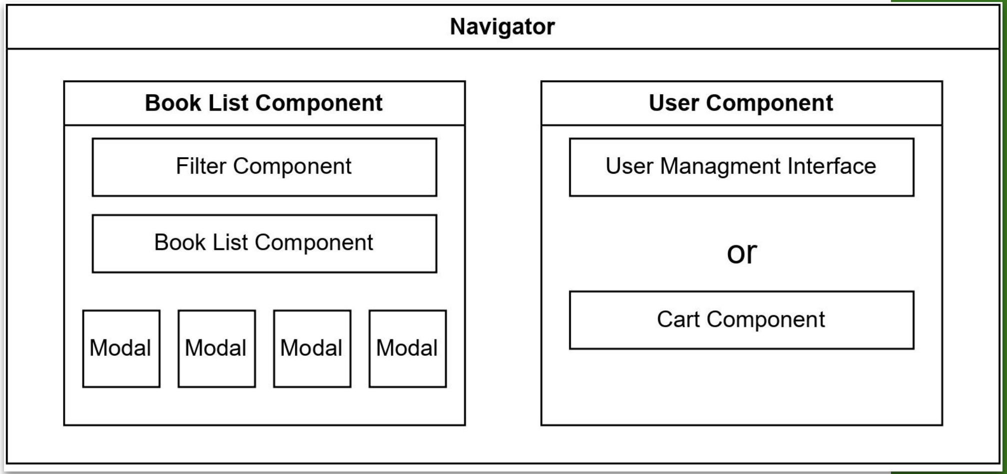
- **Cart API**
  GET: Retrieves the books in the user's cart.
  POST: Adds a book to the user's cart.
  DELETE: Removes a book from the user's cart.

- **Review API**
  GET: Retrieves reviews for a specific book.
  POST: Adds a review for a specific book.

- **Librarian API**
  GET: get information about a book using AI service.

- **User Management APIs**

Proposed Solution - FrontEnd



Developed in React JS

- Reusable Components
- Bootstrap Framework