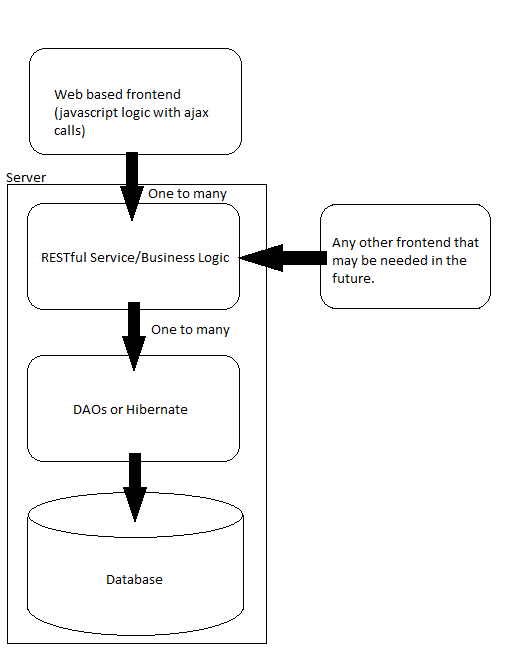
**An Introduction to TheBookCloud**

TheBookCloud is a web application that creates the easiest (and prettiest) way to read literature that is in the public domain. It provides a rich and comprehensive user interface for selecting books/short stories to read. The project is broken into two logical parts: The front-end for rendering the material and the RESTful backend for database access. It requires no software other than a browser and displays the material in a much more aesthetic way (realistic page turns and two pages displayed side by side) than other online readers such as Google books. This application is for anyone who is an avid reader and wants to keep tabs on their reading through all devices. Because the backend is RESTful it is accessible for other applications. So if another programmer wanted to grab books or chapters of books or author information, the material would be available for them to do so. This application could hold extra special value in a scholastic environment where funding is tight.



**Technology stack:**

Data base: MySql

ORM (jpa implementation): Hibernate

IoC container: Spring 3 Core

Restful framework: Spring MVC (using @RequestMapping and @PathVariable annotations)

JavaScript Library: jQuery & jQuery UI

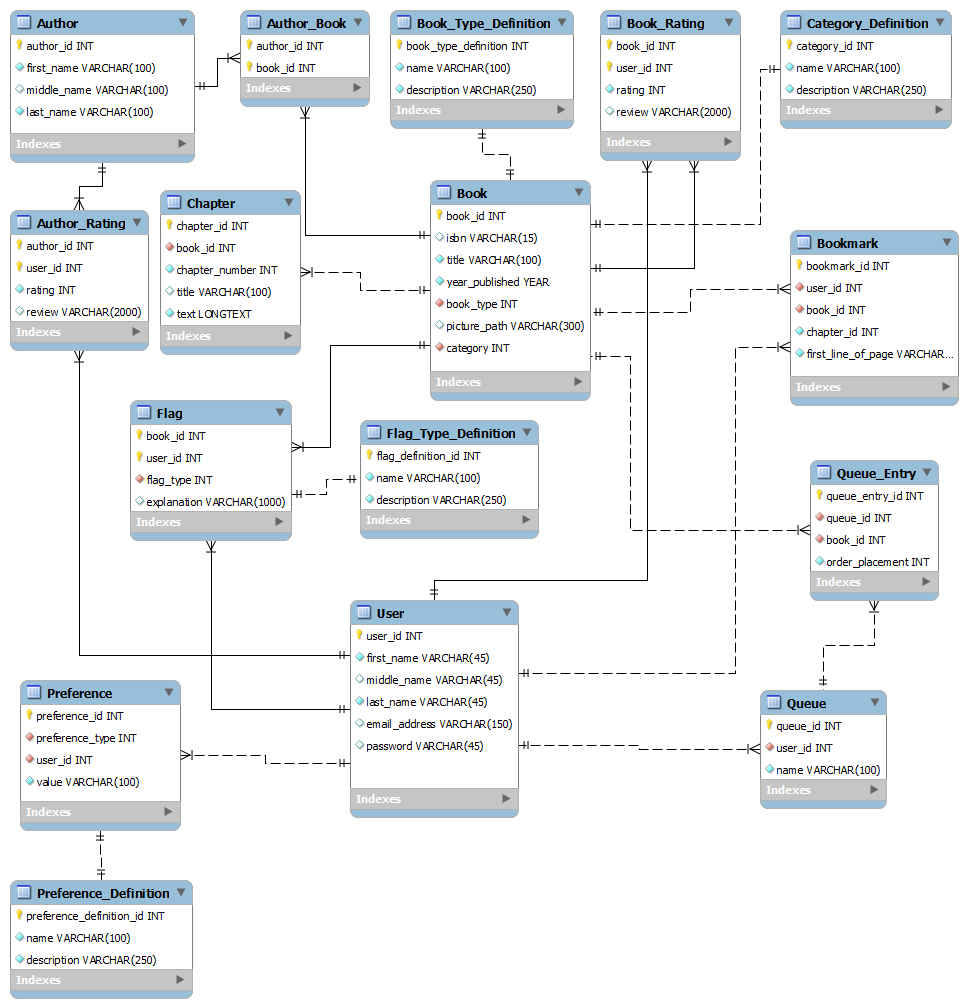
JavaScript FrameWork: BackBone.js or none (TBD)

**Things users can do:**

1. Sign up.
   * This will take minimal input from the user. First name, last name, email, password
2. Log In.
3. Log in with Face book.
4. Log in with Google.
5. Like us on Facebook.
6. Upload an ePub.
   * The site will keep an absolute minimum amount of material. Items to be kept are: title, author(s), and the chapters of the book with minimal formatting (epubs are natively HTML).
7. Search for a book.
8. Search for an author
9. Browse for a book.
10. Browse for an author.
11. Flag copyright infringement.
12. Flag inaccurate text.
13. Flag inappropriate material.
14. Rate a book.
15. Rate an author.
16. Read a book.
17. Set a bookmark.
18. Remove a bookmark.
19. Start reading from a bookmark.
20. Create a queue.
21. Delete a queue.
22. Add a book to a queue.
23. Remove a book from a queue.
24. Sort a queue (to include custom sorting).

**The Data Base:**

Given the user requirements, this should be a suitable entity relationship diagram.



**A List of Anticipated Classes:**

1. Pojos
   1. Author
   2. AuthorRating
   3. Book
   4. Bookmark
   5. BookRating
   6. Chapter
   7. Flag
   8. Preference
   9. Queue
   10. QueueEntry
   11. User