Otay Ranch Community Book Exchange Database Design and Architecture

Karl Moreno
Github: karlMoreno

Project Description

Project Overview:
In the Otay Ranch community, a noticeable absence of free library stands in the local parks has been identified. Our project aims to bridge this gap by establishing free library stands for the exchange of books, thereby fostering a stronger sense of community and encouraging a culture of reading and knowledge sharing. To complement these physical library stands, we plan to create a dedicated Discord server. This server will serve as a digital meeting place where community members can engage, discuss, and connect over their shared interests in literature and community activities.

Objectives:

Establish Free Library Stands: Set up and maintain free library stands in various parks around the Otay Ranch community. These stands will operate on the principle of free book exchange, allowing community members to borrow, return, and donate books conveniently. Through the exchange volunteers such as high schools students with an interest in computer science will attain on the job experience that includes maintained Databases in the SQL Query programming language and contributed bot commands in Python scripting programming languages. As well as the other tasks of organization, moderation, and community involvement that goes into a project like this.

Create a Community Discord Server: Develop an online platform on Discord where residents can interact, discuss books, share recommendations, and engage in community-building activities. This server will also provide updates on the latest additions to the library stands and upcoming community events.

Foster Community Engagement: Use both the physical library stands and the digital Discord server to bring people together, creating a sense of belonging and community spirit. Encourage discussions, book clubs, reading challenges, and other community-driven initiatives.

Promote Reading and Literacy: Enhance the accessibility of books to all age groups in the community, promoting reading habits, and contributing to the overall literacy and education of the community.

Facilitate Book Sharing and Accessibility: Ensure that a diverse range of books is available at the library stands, catering to various interests and age groups, making reading material more accessible to everyone.

Community Empowerment: Empower community members to take an active role in managing and maintaining the library stands, contributing to the library's collection, and participating in the Discord server's activities.

Project Impact:

By combining the tangible aspect of physical library stands with the dynamic interaction of a Discord server, this project aims to create a well-rounded community experience. It's not just about sharing books; it's about building connections, sharing knowledge, and nurturing a vibrant community culture. The introduction of these library stands, coupled with the digital platform, will provide a comprehensive approach to community engagement, learning, and shared experiences in the Otay Ranch area.

Use Cases

Use Case	Field Inventory Management
Actors	Emily (Community Administrator), Books, Authors, Genres, Library Stands in Parks
Description	As a Community Administrator, Emily is responsible for the hands-on management of book inventories across the Otay Ranch park library stands. Her role involves conducting regular visits to each park, physically verifying and cataloging the books present, and ensuring the accuracy of the library's digital database. During her rounds, Emily updates book details, including assigning unique IDs, linking books to their authors and genres, and noting the location of each book at specific park stands. She pays close attention to the condition of books, updates author and genre information as needed, and coordinates with volunteers for restocking and maintenance. Additionally, Emily engages with community members to gather feedback and encourage participation in the library's Discord server, playing a key role in enhancing the community's reading culture and overall library experience. Her thorough and interactive approach ensures that the park library stands are not only well-maintained but also serve as vibrant hubs for community engagement and learning.

Use Case	Overseeing Library User Feedback and Engagement

Actors	Library Users, Reviews and Ratings, Feedback and Suggestions
Description	Liam, responsible for user engagement and feedback, utilizes the system to monitor and manage reviews, ratings, and feedback provided by library users. He ensures that each review and rating is correctly linked to the respective user and book, maintaining a record of each user's ID, the reviewed book's ID, the rating, and the text of the review. Additionally, Liam collects and organizes feedback and suggestions from users, each tagged with a unique ID and linked to the user's ID. His role is crucial in gathering user insights and feedback, which aids in improving the library's services and collection.

Use Case	Library System Administration and Oversight
Actors	Community Administrators, Libraries, Library Users, Books
Description	Alex, serving as a Community Administrator and Discord Moderator, is integral to the operational excellence and community engagement of the neighborhood library system. Through the Free Library Database System, Alex's role encompasses dual responsibilities: moderating the library's online community on Discord and physically verifying the book inventory across various library locations.

Use Case	Community-Driven Book Exchange via Discord
Actors	New Library User, Discord Server (Bot Commands), Book Donation System
Description	In this use case, a new library user, Alex, engages in the Otay Ranch park library stands' book exchange program using a Discord server. Alex starts by donating a

	book through a Discord bot command, providing details like the title and author, which the bot then registers in the database along with Alex's Discord username as the donor. When borrowing a book, Alex uses another bot command to locate and borrow available titles. After reading, Alex's curiosity about the book's previous owners is satisfied using a command that retrieves the book's history, respecting privacy norms. Alex can also leave feedback or a review for the book via the bot, enhancing community interaction. The system optionally notifies donors when their donated books are borrowed, creating a cycle of community contribution and connection. This streamlined process via Discord bot commands not only facilitates book donations and borrowing but also fosters a rich community engagement and a shared reading experience within the Otay Ranch neighborhood.

Functional Requirements

Books:
a. A Book shall have only one ID.
b. A Book shall have only one Title.
c. A Book shall have one or many authors.
d. A Book shall have at least one genre or category.
e. A Book shall belong to at least one library location.
f. A Book may have multiple authors.
g. A Book may belong to multiple genres. (If a book can have more than one genre)

Library Users:
a. A User shall have an ID.
b. A User shall have one discord username.
c. A User shall have a record of borrowed books.
d. A user shall have a record of donated books
e. A User may have a record of reviews and ratings. (To link users with their reviews and ratings)

Libraries:
a. A Library shall have one ID.
b. A Library shall have a name.
c. A Library shall have one location.
d. A Library shall have a list of available books.
e. A Library may receive book donations. (To link libraries with book donations)

Borrowing Records:
a. A Record shall have one ID.
b. A Record shall have one User ID.
c. A Record shall have one Book ID.
d. A Record shall have a borrow date.
e. A Record shall have a return date.

Reviews and Ratings:
a. A Review shall have one ID.

b. A Review shall have one User ID.
c. A Review shall have one Book ID.
d. A Review shall have one rating.
e. A Review shall have one text description.

Community Administrators:
a. An Administrator shall have one ID.
b. An Administrator shall have one first name.
c. An Administrator shall have one last name.
d. An Administrator shall have one role or position.
e. An Administrator shall have one discord username

Book Donations:
a. A Donation shall have one ID.
b. A Donation shall have one Donor ID.
c. A Donation shall list one or many Books.
d. A Donation shall have a donation date.

Feedback and Suggestions:
a. A Feedback shall have one ID.
b. A Feedback shall have one User ID.
c. A Feedback shall have one content description.
d. A Feedback shall have a submission date.

Authors:
a. An Author shall have one ID.
b. An Author shall have a first name.
c. An Author shall have a last name.
d. An Author may have affiliations.

Genres:
a. A Genre shall have one ID.
b. A Genre shall have a name.
c. A Genre may have a description.

ERD

Books:
Book_ID (Primary Key)
Title
Genre_ID (Foreign Key)
Library_ID (Foreign Key)

Library Users:

User_ID (Primary Key)
Discord_Username
Borrow_Record_ID (Foreign Key)
Donation_Record_ID (Foreign Key)
Review_ID (Foreign Key)

Libraries:

Library_ID (Primary Key)
Name
Location

Borrowing Records:

Record_ID (Primary Key)
User_ID (Foreign Key)
Book_ID (Foreign Key)
Borrow_Date
Return_Date

Reviews and Ratings:

Review_ID (Primary Key)
User_ID (Foreign Key)
Book_ID (Foreign Key)
Rating
Text_Description

Community Administrators:

Administrator_ID (Primary Key)
First_Name
Last_Name
Role

Book Donations:

Donation_ID (Primary Key)
Donor_ID (Foreign Key)
Book_ID (Foreign Key)
Donation_Date
User_ID (Foreign Key)

Feedback and Suggestions:

Feedback_ID (Primary Key)
User_ID (Foreign Key)
Content_Description
Submission_Date

Authors:

Author_ID (Primary Key)
First_Name
Last_Name

Genres:

Genre_ID (Primary Key)
Genre_Name
Description

Libraries:

Library_ID (Primary Key)
Name
Location

Junction Tables (for many-to-many relationships):

Authorship (Book_Authors) (Book_ID, Author_ID)
Classification (Book_ID, Genre_ID)

Constraints Description Table

Table	Foreign Key	On Delete	On Update	Comment
Books	Author_ID	SET NULL	CASCADE	If an author is deleted, the author's reference in the book should be nullified to preserve the book record. Updates to an author should cascade to all their books.
Books	Genre_ID	SET NULL	CASCADE	If a genre is deleted, the genre's reference in the book should be nullified to preserve the book record. Updates to a genre should cascade to all books in that genre.
Books	Library_ID	SET NULL	CASCADE	If a library is deleted, the library's reference in the book should be nullified to preserve the book record. Updates to a library should cascade to all books in that library.
Borrowing Records	User_ID	SET NULL	CASCADE	If a user is deleted, their

				associated borrowing records will not be deleted but the User_ID will be set to NULL. Any updates to the user's information will cascade to the borrowing records.
Borrowing Records	Book_ID	SET NULL	CASCADE	If a book is deleted, the Book_ID in borrowing records will be set to NULL, indicating that the book is no longer in the system. Any updates to the book's information will cascade to the borrowing records.
Reviews and Ratings	User_ID	CASCADE	CASCADE	If a user is deleted, their reviews and ratings should also be deleted to remove their personal data and contributions. Updates to the user's information should cascade to their reviews and ratings.
Reviews and Ratings	Book_ID	SET NULL	CASCADE	If a book is deleted, the

				book's reference in the reviews should be set to NULL, as the reviews could still be valuable for historical analysis or future reference. Updates to the book's information should cascade to the reviews and ratings associated with the book.
Book Donations	Donor_ID	SET NULL	CASCADE	If a donor is deleted, their identifier in the donation records should be set to NULL, preserving the donation record without the donor's information. Updates to the donor's information should cascade to the donation records.
Book Donations	Book_ID	RESTRICT	CASCADE	Deletion of a book should be restricted if it has been donated to prevent loss of donation history. Updates to the book's information should cascade to the donation

				records.
Feedback and Suggestions	User_ID	SET NULL	CASCADE	If a user is deleted, their identifier in the feedback records should be set to NULL, allowing the feedback to remain for quality assurance and historical reference. Updates to the user's information should cascade to their feedback records.
Authorship	Book_ID	CASCADE	CASCADE	If a book is deleted, its associations with authors should also be removed (CASCADE). Updates to a book's ID should cascade to the Authorship table.
Authorship	Author_ID	CASCADE	CASCADE	If an author is deleted, their associations with books should also be removed (CASCADE). Updates to an author's ID should cascade to the Authorship table.
Classification	Book_ID	CASCADE	CASCADE	If a book is deleted, its

				genre classifications should also be removed (CASCADE). Updates to a book's ID should cascade to the Classification table.
Classification	Author_ID	CASCADE	CASCADE	If a genre is deleted, all book associations with that genre should also be removed (CASCADE). Updates to a genre's ID should cascade to the Classification table.

Relationship Summary

13/22
Books

Many-to-Many with Authors: Each book can have many authors, and each author can write many books.
Many-to-Many with Genres: Each book belongs to many genres, and each genre can include many books.
Many-to-One with Libraries: Each book is stored in one library, but each library can store many books.
One-to-Many with Borrowing Records: Each book can have many borrowing records.
One-to-Many with Reviews and Ratings: Each book can have many reviews and ratings.
One-to-Many with Book Donations: Each book can be donated multiple times.

Library Users

One-to-Many with Borrowing Records: Each user can have many borrowing records.
One-to-Many with Book Donations: Each user can make many book donations.
One-to-Many with Reviews and Ratings: Each user can write many reviews and ratings.
One-to-Many with Feedback and Suggestions: Each user can provide many pieces of feedback and suggestions.

Libraries

One-to-Many with Books: Each library can contain many books.
One-to-Many with Borrowing Records: Each library can have many borrowing records associated with it.
One-to-Many with Book Donations: Each library can receive many book donations.

Borrowing Records

Many-to-One with Library Users: Each borrowing record is associated with one library user.
Many-to-One with Books: Each borrowing record is associated with one book.

Reviews and Ratings

Many-to-One with Library Users: Each review and rating is written by one library user.

Many-to-One with Books: Each review and rating is associated with one book.

Book Donations

Many-to-One with Library Users (Donors): Each book donation is made by one library user.

Many-to-One with Books: Each donation record is associated with one or more books.

Feedback and Suggestions

Many-to-One with Library Users: Each feedback or suggestion is provided by one library user.

Authors

One-to-Many with Books: Each author can write many books.

Genres

One-to-Many with Books: Each genre can include many books.

Community Administrators

One-to-Many with Feedback and Suggestions: An administrator may address or respond to multiple feedback entries or suggestions.

Non-Functional Database Requirements

Performance Requirements:

The system should be able to handle simultaneous access by multiple users without significant lag or downtime.

Response time for querying the database, such as searching for books or updating book records, should be within a few seconds.

Availability and Reliability:

The system should be available for use at all times, with minimal downtime for maintenance.

There should be a high degree of reliability in data storage, with a minimal risk of data loss or corruption.

Scalability:

The database should be scalable to accommodate the growing number of books, users, and transaction records over time.

It should be capable of scaling up to handle peak usage times without performance degradation.

Security:

The system must ensure the security and privacy of user data, including personal details and transaction histories.

Proper authentication and authorization measures should be in place to prevent unauthorized access.

Data Backup and Recovery:

Regular backups of the database should be conducted to prevent data loss.

There should be a robust data recovery plan in case of system failures or disasters.

User Interface:

The user interface for accessing the database, particularly for non-technical users, should be intuitive and user-friendly.

Accessibility features should be included to cater to users with different abilities.

Interoperability:

The system should be able to integrate or communicate with other systems or platforms, such as the community’s Discord server or other library management systems.

Compliance and Standards:

The system should comply with relevant data protection and privacy laws.

Adherence to industry standards for database design and management is essential.

Maintenance and Support:

The system should be easy to maintain and update without significant disruptions.

Support should be available for addressing technical issues or user queries.

Environmental Sustainability:

The system's design and operation should consider environmental sustainability, minimizing energy consumption and utilizing eco-friendly hosting options.