**Team 71: Annmarie Geiger, Krystal Lu  
Project Title: Book Tracking Management System**

**Project Step 2 Draft**

### **Feedback**

We received four different reviews of our draft.

Adrianna Hoffman:

“Does the overview describe what problem is to be solved by a website with DB back end? Does the overview list specific facts?

Yes, the overview describes a website that helps people search up and keep track of different books they are reading/have read. People will have the ability to rate books, keep track of their reading progress, and share their read books with other users. They will also be able to search for books by author and genre for more efficient browsing. It also specifies that this website is currently designed for around 2,500 active users but has the ability to grow as its number of users grows. Overall, it’s a very well thought out management system.

Are at least four entities described, and does each one represent a single idea to be stored as a list? Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities? Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

Yes, they have created 5 entities: Users, Books, UserBooks, Authors, Genre, as well as an interaction table GenreofBooks. All entities have descriptive attributes, data types, and constraints. I particularly like the use of the UserBook entity to allow users to pull in books from the Books entity and manipulate the status and ratings a user has for a particular book. There are several 1:M relationships, and one M:M relationship in that books can have multiple genres and genres can have multiple books. The only thing I would consider is that some books can have multiple authors, which would make the relationship M:M instead of M:1.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

For the most part, I think the names are consistent and are clear and easy to understand. The few discrepancies are in the plural vs singular aspect. The Genre entity should be Genres, and the attributes idUsers and idBooks are plural when they should be singular - idUser and idBook.”

Emily Ho:

“Hi Annmarie & Krystal,

Great project draft so far! I love the idea of users being able to easily track books that they’re interested in and/or have already read.

**Does the overview describe what problem is to be solved by a website with DB back end? If yes, summarize. If not, what changes would better support describing the problem to be solved?**

The overview describes the problem that it is solving is to create a database for an existing website from a startup company that will enable users to review books they’ve read, share their book collections, and monitor reading progress. Users’ personal information, book details, and interactions with the 1,000 books that are available for browsing will all be stored in this database. For a personalized user experience, the database will include features like showing the books they have read, adding books to their wishlist and removing books that the user is no longer interested in. For user browsing efficiency and usability, the database will also allow the website to show sorted books by author and genre.

**Does the overview list specific facts? If yes, summarize what the facts illustrate about the proposed DB solution. If not, what facts would better support illustrating the scope and scale of the proposed DB solution?**

The overview includes specific facts like over 2,500 active users annually and 1,000 books to browse in the database as well as the ability to handle 1000s of users and books per year. This gives the stakeholder an idea of how much traffic the website receives along with the expectation to grow.

It may be helpful to provide information about how the current website is operating. What are its current capabilities (since it already has over 2,500 active users) without a backend, or does the website currently have an insufficient backend that the company wants rebuilt? You can mention the current challenges Booktique is facing so the stakeholders have a better idea of how your book tracking management system will improve the website.

**Are at least four entities described, and does each one represent a single idea to be stored as a list? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

There are 5 entities described in the outline: Users, UserBooks, Books, Authors, and Genre. GenreofBooks is appropriately labeled as an intersection table. I may recommend to update the name to GenreOfBooks to make it easier to read.

**Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

1. Users: This entity stores information about the users of the Booktique website. This includes appropriately useful attributes like idUsers, email, userName, and password. Users have a 1:M relationship with UserBooks.

The attribute idUsers could be updated to idUser to make it singular if following the professor’s preferred method of making attributes singular and entity tables plural.

2. UserBooks: The description says it “stores details of books available in the database that users can add to wishlist, check off that they have read, drop, and rate books.” This is all in the present tense. And the relationships described mention “tracking activity” and being able to “track” a book.

I believe this entity stores details about the current status of the book (favorited/wishlist, read, dropped, and rated) and its rating by the user. I’m curious if this may actually mean that Users and Books have a M:N relationship implemented with UserBooks as the intersection table. Each user can have many books that they’re tracking. And each book can be tracked by many users.

But in case I'm not interpreting your database structure correctly, here are notes about the current description. It could be helpful to add that idUser FK is from the Users entity and that idBook is from the Books entity. For attributes, I believe the recommendation is usually to use camelcase and start it in lowercase and to be consistent with the other attributes you have. You could possibly update UserBooksID to follow this.

3. Books: This entity stores information about the books available in the database. It has the attributes idBooks, bookTitle, bookDescription, bookYear, Authors\_idAuthor.

For the attribute bookYear, it might be better to use VARCHAR or INT if you’d like the table to just record the year (since the DATE datatype is in the format YYYY-MM-DD). The professor mentioned user VARCHAR if you want more flexibility (like companies that use fiscal year 2025Q1, etc), and to use INT when you know for sure there will be nothing but numbers.

Alternatively, if you want to use DATE, maybe the attribute could be renamed to something like bookPublishedDate.

It may be helpful to add that idAuthor is a FK from the Authors entity. I would consider making idBooks singular.

The outline says that there is a M:1 relationship with Authors as many books are written by only one author. I believe if M:1 is the relationship, then the appropriate explanation is that each book can only have 1 author associated with it. So maybe you just have to explain that the frontend or backend database limitation is to only be able to associate each book with 1 author. And if certain books have multiple authors, then it could to be recorded through a separate attribute (or just not recorded at all).

4. Authors: This entity stores information about the authors of the books in the database. It includes the attributes idAuthor and authorName with the appropriate datatypes.

5. Genre: This entity stores information about the genres of books. The attributes are idGenre and genreName, and have the appropriate datatypes.

It could be recommended to make Genre plural.

There is a M:N relationship that links Genres and Books since each genre can be associated with many books.

**Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

The M:N relationship is between Books and Genres implemented through the intersection table GenreOfBooks.

I had a question above about the relationships relating to UserBooks.

In terms of Authors and Books, I think you might just have to determine whether you want it to be a M:N relationship or stick to 1:M and then attribute that to a system limitation or that Booktique only has books in its database where there is 1 author.

The ERD shows the relationships appropriately constructed and presents a logical view of the database. It includes the entities, their primary keys, and associated foreign keys.

**Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

There are some primary keys that could be changed to singular (idUsers, UserBooksID, idBooks). The entities (Users, UserBooks, Books, Authors, Genre) use PascalCase naming convention, and they’re all plural except for the Genre entity. The attributes mostly use camelCase naming convention but there are some inconsistencies (Ex. UserBooksID likely should start in lowercase).”

Gregory Preiss:

“**Does the overview describe what problem is to be solved by a website with DB back end? If yes, summarize. If not, what changes would better support describing the problem to be solved?**

The overview does describe the problem. Booktique is a start-up company that wants to have a website that helps users track their reading progress, share their book collection, and rate books they have read. Because they are a start-up, they do not currently have this feature, so Team 71 will provide Booktique with the management system they desire.

**Does the overview list specific facts? If yes, summarize what the facts illustrate about the proposed DB solution. If not, what facts would better support illustrating the scope and scale of the proposed DB solution?**

It does list specific facts. Booktique has over 2,500 active users annually. They also have about 1k books to browse. This illustrates that Booktique definitely needs a website with a database that can seamlessly track all this information. This would be very, very difficult without one!

**Are at least four entities described, and does each one represent a single idea to be stored as a list? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

They have five entities and each represents a single idea to be stored as a list. The Users entity is for storing user details. The UserBooks entity is for users to add books to a wishlist, check off the books they have already finished reading, get rid of books, and to give books ratings. The Books entity is simply for the details of each book in their collection. The Authors entity is for details of each author. The Genres entity is for storing details of each genre.

**Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

The outline does describe the details of each entity and its purpose. It does list attribute datatypes and constraints. It does list the relationships, but they are a tiny bit ambiguous. I think for the relationships, it would be helpful to just make it more explicit how the relationship is tied together. For example: Users to UserBooks is a 1:M relationship and is implemented with Users\_iDUsers as the FK inside UserBooks. Pointing out the FK and which table it is inside helps the reader understand exactly how they are related. With that said, they did a good job explaining how the relationships will work on the website.

**Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

The 1:M relationships are mostly correct. I only spotted one that needed more clarification. Under the Books entity, the relationship from Books to Authors is said to be M:1 because “many books are written by only one author”. They should establish if all their books only have one author or not. Because if any of their books have multiple authors, this is no longer a M:1 relationship.

There is at least one M:M relationship, and it is correct. With that said, I would clarify what the intersection table is and which FKs are being used to implement the M:M relationship between entities.

The ERD does present a logical view of the database, and is easy to follow.

**Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

Unfortunately, the draft is a bit inconsistent throughout.

They are capitalizing and pluralizing all the entities, but the Genre entity is singular. So that needs to be made plural.

The attributes are a complete mix of singular and plural. For example, idUsers is plural, but the rest of the attributes in Users are singular. The same for idBooks under Books. They then have idAuthor being singular under Authors. These are just a few examples. This inconsistency needs to be fixed.

They need to also decide whether to use camelCase or underscores for their attributes, and stick with it. They are mixing them throughout. They also need to decide whether to capitalize the beginning of each attribute, or not capitalize. These inconsistencies made it quite difficult to navigate each attribute.

Overall, this is a great start to the project, and I think if they quickly fix the inconsistencies and relationship descriptions, they have an excellent project outline :)”

Alena Makarova (TA):

“Hi Team, great job!

However, there are a few areas that could be refined. First, there is inconsistency in naming conventions across the database schema. For example, *idUsers* and *Users\_idUsers* use different naming patterns, which could create confusion during implementation. Standardizing these conventions would improve clarity and maintainability.

Additionally, some fields could benefit from more specificity or constraints. For instance, the *userBookStatus* field might require predefined values (e.g., "wishlist," "read," "dropped") to maintain data consistency. Similarly, the *userBooksRating* field should include constraints to define the valid range of ratings.”

### **Actions**

The feedback we received was very thorough and we learned a lot from it. The most important pieces were about the inconsistencies in our entity/attributes portions, more detail in our overview segment, and our Books entity’s relationship with Authors.

Krystal immediately edited the proper formats like plural entities and singular attributes, camelCase, proper capitalization, and clarity to some of our entities descriptions. She also changed the *userBookStatus* attribute type to ENUM() to give predefined values (e.g., ‘wishlist’, ‘read’, ‘dropped’) to maintain data consistency. Range constraints were also included in the *userBooksRating* to maintain data consistency.We also added foreign keys to the necessary entities and where they are referenced.

Annmarie supplied more information to the overview giving a proper background as to why the Booktique company needed a proper database and backend than the one that they are currently working with.

We originally had the Books and Authors relationship as 1:M, but moved forward with the feedback to make an M:M relationship between them, as books can have multiple authors. This was something that we weren’t too sure about, so hearing this advice helped solidify this choice to change the relationship. Krystal recreated the ERD to show the change and the addition of the intersection table that is required for the new relationship.

### **Upgrades to Draft**

* **Books**
  + bookDescription: Data Type changed from varchar(255) to TEXT.
  + bookPublishDate: Added NOT NULL.
* **AuthorsofBooks**
  + Conversion to 2NF: Added a primary key called ‘authorBookID’.
* **GenresofBooks**
  + Conversion to 2NF: Added a primary key called ‘genreBookID’.

### 

### **Overview**

Booktique is an ambitious, highly-driven start-up company with over 2,500 active users annually. We are seeking assistance in developing a database for our website to help users track their reading progress, share their book collection, and rate books they have read. The website's database will store users' personal data, book details, and their interactions with the 1,000 available books to browse. Features such as: displaying the books they have read, putting books on their wishlist, and dropping books they are no longer interested in are key for creating a personalized user experience. Another requirement is the need to categorize books by author and genre to make browsing our website efficient and easy for our users. This system is designed to handle user interactions on a scale of thousands of users and books per year, with expectations for outstanding growth.

Currently, our system meets basic functionality, but our backend does not meet these growing demands of our user base. Data organization and scaling issues limit support for our website’s needs. This is where you come in! Implementing the above requirements will position Booktique to greater success in the market and among its user base.

### **Database Outline**

**Users:** Stores the details of the users on the website.

* userID: INT, AUTO\_INCREMENT, UNIQUE, NOT NULL, PK
* userName: VARCHAR(100), UNIQUE, NOT NULL
* password: VARCHAR(255), NOT NULL
* email: VARCHAR(255), UNIQUE, NOT NULL
* Relationships:
  + 1:M relationship from Users to UserBooks: A single user can track multiple books.
  + M:N relationship between Users and Books: A user can read one or more books.

**Books:** Stores the details of books available in the database.

* bookID: INT, AUTO\_INCREMENT, UNIQUE, NOT NULL, PK
* bookTitle: VARCHAR(100), NOT NULL
* bookDescription: TEXT, NOT NULL
* bookPublishDate: date, NOT NULL
* Relationships:
  + M:N relationship between Books and Authors: A book can be written by one or more Authors.
  + M:N relationship between Books and Users: A book can be read by one or more Users.
  + M:N relationship with Genres as each genre can have many books related to it and each book can have many genres.

**Authors:** Stores the details in relation to authors of books.

* authorID: INT, AUTO\_INCREMENT, UNIQUE, NOT NULL, PK
* authorName: VARCHAR(100), NOT NULL
* Relationship:
  + M:N relationship with Books as one author can write many books.

**Genres:** Stores the details of genres.

* genreID: INT, AUTO\_INCREMENT, UNIQUE, NOT NULL, PK
* genreName: VARCHAR(45), NOT NULL
* Relationships:
  + M:N with Books as each book can have many genres and each genre can have many books related to it.

**UserBooks:** Stores the details of books available in the database that users can add to wishlist, check off that they have read, dropped, and/or rate books.

* userBookID: INT, AUTO\_INCREMENT, UNIQUE, NOT NULL, PK
* userID: INT, NOT NULL, FK from Users
* bookID: INT, NOT NULL, FK from Books
* userBookStatus: ENUM('wishlist', 'read', 'dropped'), NOT NULL
* userBookRating: TINYINT(1), NULL, CONSTRAINT CHECK(userBookRating IS NULL OR ‘userBookRating > 0 and ‘userBookRating’ <= 5)
* Relationships:
  + M:1 relationship from UserBooks to Users as multiple entries in UserBooks can belong to one user for tracking activity.
  + M:1 relationship from UserBooks to Books as many users can track the same book.
  + An intersection table that was created to handle the M:N relationship between Users and Books. It is also an entity since it includes additional attributes.

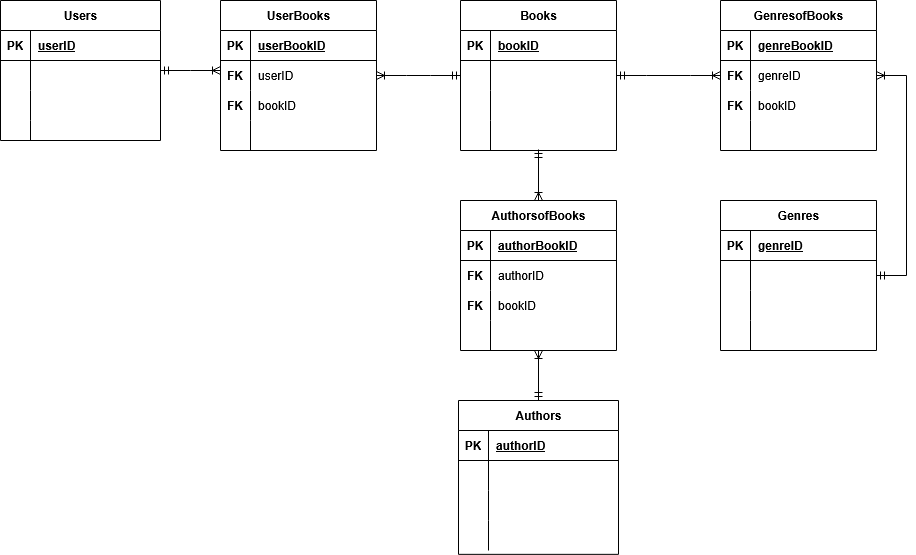
**AuthorsofBooks:** An intersection table to handle an M:N relationship between Books and Authors.

* authorBookID INT, AUTO\_INCREMENT, UNIQUE, NOT NULL, PK
* authorID: INT, NOT NULL, FK from Authors
* bookID: INT, NOT NULL, FK from Books
* Relationship:
  + M:1 relationship between AuthorofBooks and Authors.
  + M:1 relationship between AuthorofBooks and Books.

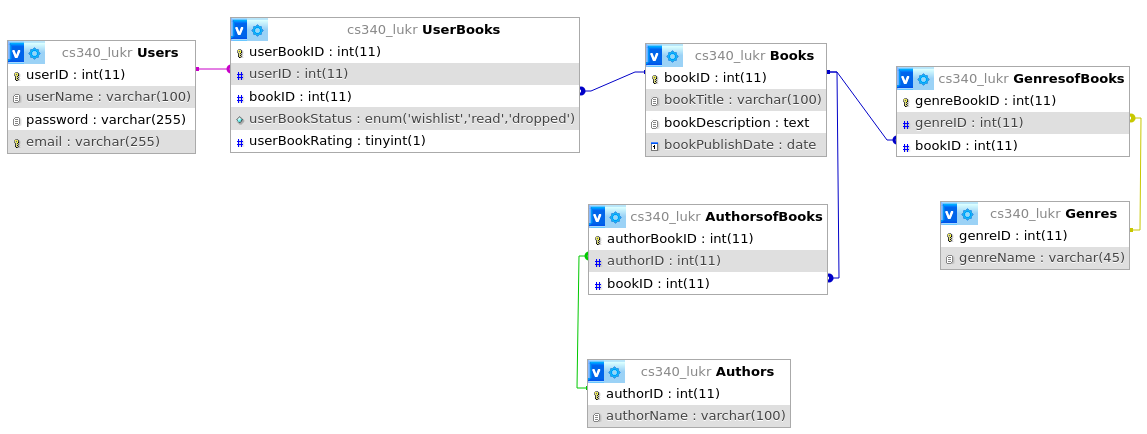
**GenresofBooks:** An intersection table to handle an M:N relationship between Books and Genres.

* genreBookID INT, AUTO\_INCREMENT, UNIQUE, NOT NULL, PK
* genreID: INT, NOT NULL, FK from Genres
* bookID: INT, NOT NULL, FK from Books
* Relationship:
  + M:1 relationship between GenreofBooks and Genres.
  + M:1 relationship between GenreofBooks and Books.

### **Entity-Relationship Diagram**

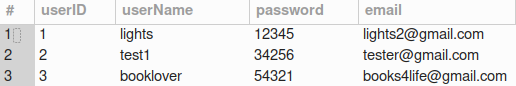
****

**Schema**

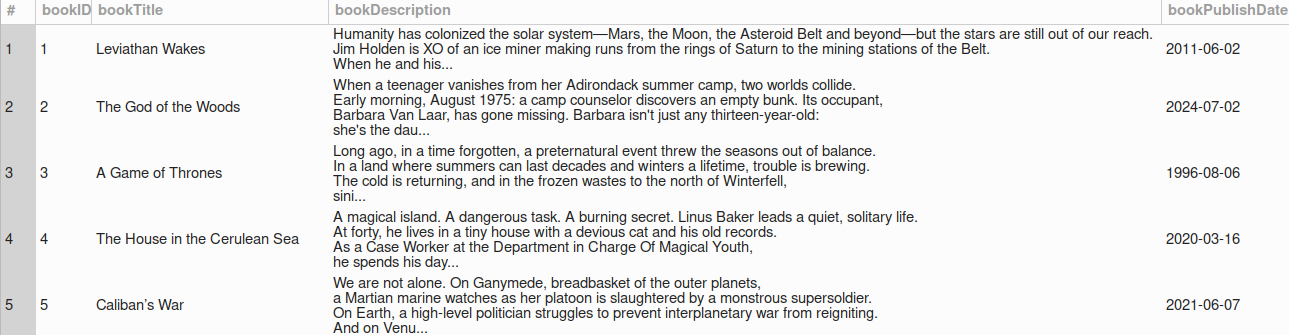
****

**Example Data**

Users:



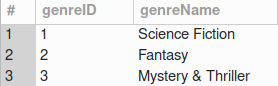
Books:



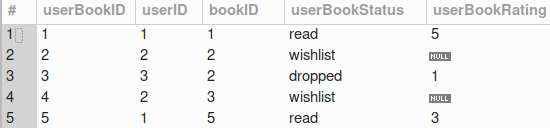
Authors:



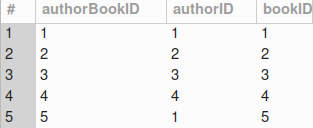
Genres:



UserBooks:



AuthorsofBooks:



GenresofBooks:



**Data References:**

**Goodreads. "Caliban’s War Summary." *Goodreads*, 2012,** [**https://www.goodreads.com/book/show/12591698-caliban-s-war**](https://www.goodreads.com/book/show/12591698-caliban-s-war)**.**

**Goodreads. "A Game of Thrones Summary." *Goodreads*, 1996,** [**https://www.goodreads.com/book/show/13496.A\_Game\_of\_Thrones**](https://www.goodreads.com/book/show/13496.A_Game_of_Thrones)**.**

**Goodreads. "The God of the Woods Summary." *Goodreads*, 2024,** [**https://www.goodreads.com/book/show/199698485-the-god-of-the-woods**](https://www.goodreads.com/book/show/199698485-the-god-of-the-woods)**.**

**Goodreads. "The House in the Cerulean Sea Summary." *Goodreads*, 2020,** [**https://www.goodreads.com/book/show/45047384-the-house-in-the-cerulean-sea**](https://www.goodreads.com/book/show/45047384-the-house-in-the-cerulean-sea)**.**

**Goodreads. "Leviathan Wakes Summary." *Goodreads*, 2011,** [**https://www.goodreads.com/book/show/8855321-leviathan-wakes**](https://www.goodreads.com/book/show/8855321-leviathan-wakes)**.**