Objective: Use your requirements document to develop a comprehensive Entity-Relationship (ER) model that accurately represents the data requirements and relationships for your database project. This model will serve as a blueprint for your database design, capturing the essential entities, relationships, and constraints based on the requirements you have gathered. No formal template is required, but the following sections should be included in a requirement document (the *italic* parts are subsections).

Introduction [5 points].

Introduction:
Project Overview: The purpose of this database is to store a collection of different books, magazines, movies able to be rented out by different users.

- **Scope:**
    - The purpose of a Library Management System is to make a useful, scalable, and educational tool that works like a real library. As part of the project, we will be able to work directly with all aspects of database development, from conceptual modeling to logical design to physical implementation. We will also add ways to make sure that borrowing rules are followed and useful reports will be made to help with operational decisions, collection growth, and member participation. At the end of the day, this project will be both a useful way to learn and show how well-designed database systems can help solve difficult information management issues.

- **Glossary:**
    - ISBN: International Standard Book Number; is a unique, 13-digit numerical identifier for a specific book edition and format, used by libraries, bookstores, and online retailers to catalog, track, and sell books globally.
    - SQL: Structured Query Language
    - Functional Requirement: specify what a system must do, detailing the specific actions, functions, and features it needs to perform to meet user and business needs
    - Non-Functional Requirement: specify what a system must do, detailing the specific actions, functions, and features it needs to perform to meet user and business needs

Identify ER Modeling Components [15 points].

- *Data Entities:*
  - *Items (Shero)*
    - Title
    - Genre
    - Release Date
    - Availability
    - Number of Checkouts
    - CopyID
    - Due Date
  - Book (Subclass of Items)
    - Author/creator
    - ISBN
  - Magazine (Subclass of Items)
    - Issue number
    - Publisher
  - Movies (Subclass of Items)
    - Rating
    - Director
  - User (Jaiden)
    - First Name
    - Last Name
    - Phone Number
    - Email Address
    - Home Address
      - House Number
      - Street Name
      - Apt Number
      - Zipcode
      - City
      - State
    - Which items has checked out
    - Account Standing (good standing if no late fees)
    - Client (Subclass of User)
      - ID
    - Librarian (Subclass of User)
      - Employee Id

Checkout/Transaction (AMRIT)

- TransactionID
- CopyID (FK → ItemCopy)
- ClientID (FK → Client)
- LibrarianID (FK → Librarian, optional)
- Checkout Date
- Due Date
- Return Date
- Late Fee (calculated or stored)

Fee (Luis)

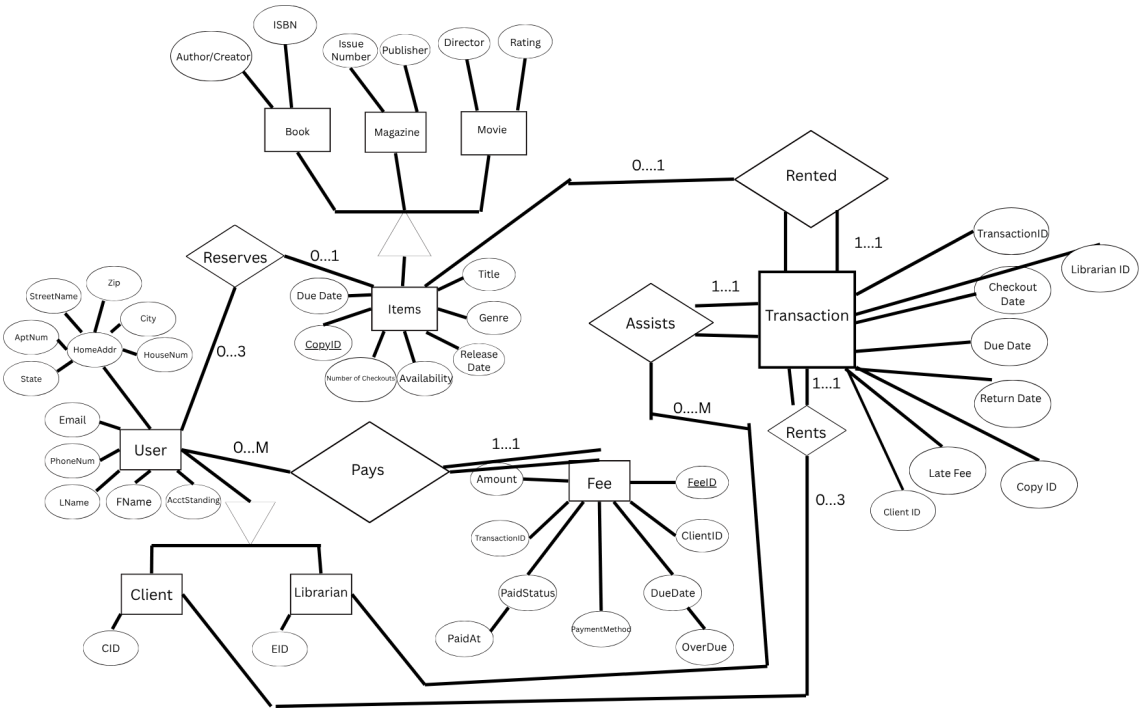
- FeeID (Primary Key)
- ClientID (FK → Client)
- TransactionID (FK → Checkout/Transaction) (Used to see which items caused the fee)
- Amount (Decimal Value)
- PaidStatus (Can be “Paid”, “Unpaid”, or “Waived”)
    - PaidAt (DateTime)
- DueDate (DateTime)
    - OverDue (Boolean)
- PaymentMethod (Can be “Cash”, “Card” or “Online”)

Relationships (Eric)

- Client (0…3) - Reserves - (0..1) Items
- User (0..M) - Pays - - (1..1) Fee
- Client (0…3) - Rents - - (1..1) Items

- Librarian (0…M) - Assists - (1..1) Items
- Items (0…1) - Rented - - (1..1) Transaction

Create the ER Model [30 points]: Use a diagramming tool (such as drawio.com



Links to an external site.

or Lucidchart.com

Links to an external site.

, Visual Paradigm

Links to an external site.

, or GetMind

Links to an external site.

, or any other ER diagramming tool) to create your ER model. Ensure that your model includes:

- All identified entities and their attributes
- Primary keys for each entity
- All relevant relationships between entities with appropriate cardinality in min..max format
- Any additional constraints or notes that are relevant (but not directly presented in the ER model)

Review your ER diagram to ensure it accurately reflects the requirements. Check for completeness and consistency. Make sure all entities, attributes, and relationships are clearly represented.

Appendices. You are welcome to use appendices to provide additional information, e.g., your design choices, explain why you chose certain entities, how you determined the relationships, and any assumptions you made during the modeling process.

GitHub Repository Management. Continue to maintain all project artifacts in your GitHub repository. The team leader should then submit the repository URL for this project part on Canvas.

https://github.com/JaidenTGreen/EECS-447-Project/tree/main