CS 3380 - 13525
Team 12
Checkpoint 1-Domain Knowledge
02/06/25

Checkpoint 1: Understanding our Library Domain

1. Acquire Domain Knowledge

a. **Scenario:**
A library lends out various types of items (books, media, devices, etc.) to two categories of borrowers: **students** and **faculty**. We need to model:

- Different **borrowing limits** (number of items allowed at once).
- Different **loan periods** (how many days each type of borrower can keep an item).
- Handling **fines** for overdue items.
- Handling **requests/holds** for items currently unavailable or checked out.
- Maintaining an inventory of **multiple copies** of the same item.

b. **Key Points:**

1. **Users/Privilege:**
- Student
- Faculty (it only means that you are a teacher or something and you can borrow more stuff and borrow it for a longer period. This does not mean you are a librarian)
- Librarian
- Database Admin

2. **Borrowers:**
- **Student** borrowers
- **Faculty** borrowers
- Each of them has unique borrowing constraints. For example: a **student** can only borrow 5 items, but **faculty** can borrow 10 items.
- Each has a different loan period allowed. For example: a **student** can only borrow an item for **14 days**, but **faculty** can borrow it for **28 days**.

3. **Items:**
- There are 3 types of items (books/media/devices)
- Each of these items has an item ID (ISBN or IMDB_ID or Model Number)
- There can be multiple copies of the same item
- Each item has a trait to tell whether they are currently available, loaned out, or reserved for the person in the front of the queue (3 states total)

4. **Transaction History:**

- Reserve Time
- Check Out Time
- Return Time
- Due Date = Check Out Time + 14 or 28 based on Borrower.Type

5. **Fines (per user) :**

- Checkout Time
- Days the item has been out for.
- If System Time > Borrower.Item.Due Date && Borrower.Item.Return Time == NULL:

  Fines = 5 * (System Time - Borrower.Item.Due Date)

6. **Triggers/Reminders:**

- Before the due date (3 days before) email
- If System Time = Borrower.Item.Due_Date + 14:

  Borrower.Status = LIMITED (can only return now, no more borrowing)
- After the due date (every day apply 5 dollar fine)
- Waitlist Got Out Congratulation Email

7. **Waitlist (Hold/Requests):**

- When that item is all loaned out, users can join a queue (a waitlist) to put a hold on the next time that item is available again (so if that item is available again, the user at the front of the queue has priority to request or borrow that item)
- After a certain amount of time (e.g 1 - 3 days (need to decide on this)) and that person has not requested/picked up that item yet, pop that person from the queue and give access to the next user in the queue

2. Mini-World

a. **Entities:**

i. **User**

- Represents anyone with access to the system. Each User has a **Role** (Student, Faculty, Librarian, or Database Admin).
- Only **Students** and **Faculty** borrow items; **Librarians** and **Database Admins** primarily perform administrative functions.

ii. **Item**

- Represents a catalog entry in the library.
- There are three primary Item categories: **Book**, **Media**, and **Device**.
- An Item is identified by a **unique real-world ID** (e.g., ISBN for books, IMDB_ID for media, or model number for devices).

iii. **Copy**

- Represents a **physical copy** of an Item.
- Multiple Copies can exist for the same Item.
- Each Copy can be in one of three states:
    1. **Available** (ready for checkout),
    2. **Loaned out** (currently with a borrower),
    3. **Reserved** (on hold for the next person in the queue).

iv. **Transaction History**

- There are many transactions in a transaction history table
- Represents a **borrowing-related action**, including reserve, check-out, and return events.
- Key timestamps include **Reserve Time**, **Check-Out Time**, **Return Time**, and a **Due Date** (which depends on the Borrower type).

v. **Waitlist(Hold/Requests)**

- Represents the **queue** for an Item when no Copies are available.
- Users on a Waitlist get priority access in a **first-come, first-served** manner once a Copy is returned.

vi. **Fines** (Optional as a separate entity or tracked within User/Transaction)

- Represents monetary penalties for overdue items.
- Can be tracked per Transaction or aggregated at the Borrower (User) level.

b. **Properties for Each Entity:**

1. **User:**
    - Email (Unique Primary Key)
    - First Name
    - Last Name
    - Role (Student, Faculty, Librarian, Database Admin)
    - Status (Active, Limited, etc. based on fines or overdue items)

2. **Item**:

    a. Books:
    - ISBN (unique primary key)
    - Author
    - Title
    - Genre
    - isAvailable (are there copies to be loaned out)

    b. Media:

- IMDB ID (unique primary key)
- Author
- Title
- Genre
- isAvailable (are there copies to be loaned out)

c. Devices:

- Model Number (unique primary key)
- Name
- isAvailable (are there copies to be loaned out)

3. **Copy of Item**
    - CopyID (unique primary key)
    - ItemID (foreign key referencing the item it is a copy of)
    - State (available, reserved, loaned out)

4. **Transaction**
    - TransactionID (primary key)
    - UserID (foreign key referencing user)
    - CopyID (which copy is this transaction for) (foreign key)
    - Reserve_Time (NULL if not reserved in advance)
    - CheckOut_Time (NULL if not yet checked out)
    - Due_Date Computed based on CheckOutTime + Borrower.Type
    - Return_Time (null until return)

5. **Waitlist:**
    - UserID (who are on that waitlist) (composite key)
    - WaitlistID (composite key)
    - ItemID (what is requested for that waitlist) (composite key)
    - PosInQueue
    - Waitlist_Joined_Date

6. **Fines:**
    - *FineID* (primary key)
    - *UserID* (who owes the fine)
    - *TransactionID* (the relevant transaction)
    - *AmountDue*
    - *DateAssessed*
    - *isPaid*

c. **Relationships**

i. **Borrower**

  1. **Items: (one to many)**
  - A **Borrower** can check out and check in multiple items at a time (one-to-many relationships)

ii. **Items:** Books/Media/Devices can be checked out and checked in by users.

  1. Book Genre (Many-to-One) A book belongs to one genre.
  2. Book Copies (One-To-Many) A book can have multiple copies available for borrowing
  3. Book Author(One-To-Many) An Author can write multiple books

iii. **User**

  1. **Transaction: (one to many)**
  - A single User (Student or Faculty) can have multiple Transactions; each Transaction is tied to exactly one borrowing User.

~~**Users:** Users can check in and out the books and return them, they can also see their transaction history.~~

  - **Transaction - Copy (one to one)**
  - Each transaction involves only one single unique copy.

iv. **Transaction History** keeps a record of the items that are still check-out (that have not yet been turned in)

v. **Waitlist:** the waitlist keeps track of the people trying to reserve or take hold of a specific item.

  1. User (one to one)
     a. Each entry in the waitlist has one associated email (user)
  2. Copy (one to many)
     a. A user is waitlisted for all copies of a given item
  3. Item (one to one)
     a. Only one item can be waitlisted per entry

vi. **Fines:** Fines work alongside the transaction history to keep track of the days the items have been checked and past the due date limit

  1. User (one-to-one)

        a. Each User has one instance of “fines” which acts as the accumulation of fines for all overdue books under the user’s account.

    2. Transaction

        a. Fines depend on “Transaction” to calculate how long the book is overdue for to calculate how large the fine should be

d. Constraints

- i. Students and Faculty cannot borrow more than the maximum amount allowed
- ii. Students and Faculty can only borrow within their borrowing period (borrowers can only borrow that copy for a specific period based on their type of borrower (student/faculty)
- iii. A Borrower cannot checkout an item (book, movie, device) if that specific item is already checked out
- iv. Item due dates cannot have past dates, all dates must be a future date
- v. Borrowers cannot check out any items if they have unpaid fines on their account
- vi. Users must have either a “@cougarnet.uh.edu” or “@uh.edu” email to register
- vii. Only users with accounts can check out/reserve items
- viii. ISBNs must be 13-digit long and Item titles/names must be a non-empty string
- ix.
- **x. States of Copies:**
    - A copy can only be “Reserved” if len(waitlist) > 0 and the user at the front of the queue has been **notified**
    - A copy can only be “Available” if it has been returned and there is no hold on that item
    - A copy can only be “Checked Out” if there is a check-out transaction history for that copy and the return date for that transaction history is NULL.
- **xi. Waitlist:**
    - There is only a waitlist when all copies of that item have been “Checked Out” or “Reserved”
    - The user at the front of the queue has priority access when a copy is available
    - If the user does not act (checkout) within a specified window (1-3 days), the user is popped from the queue (lose priority access) and the next one in the queue is given priority access and notified

CS 3380 - 13525
Team 12
Checkpoint 1-Domain Knowledge
02/06/25