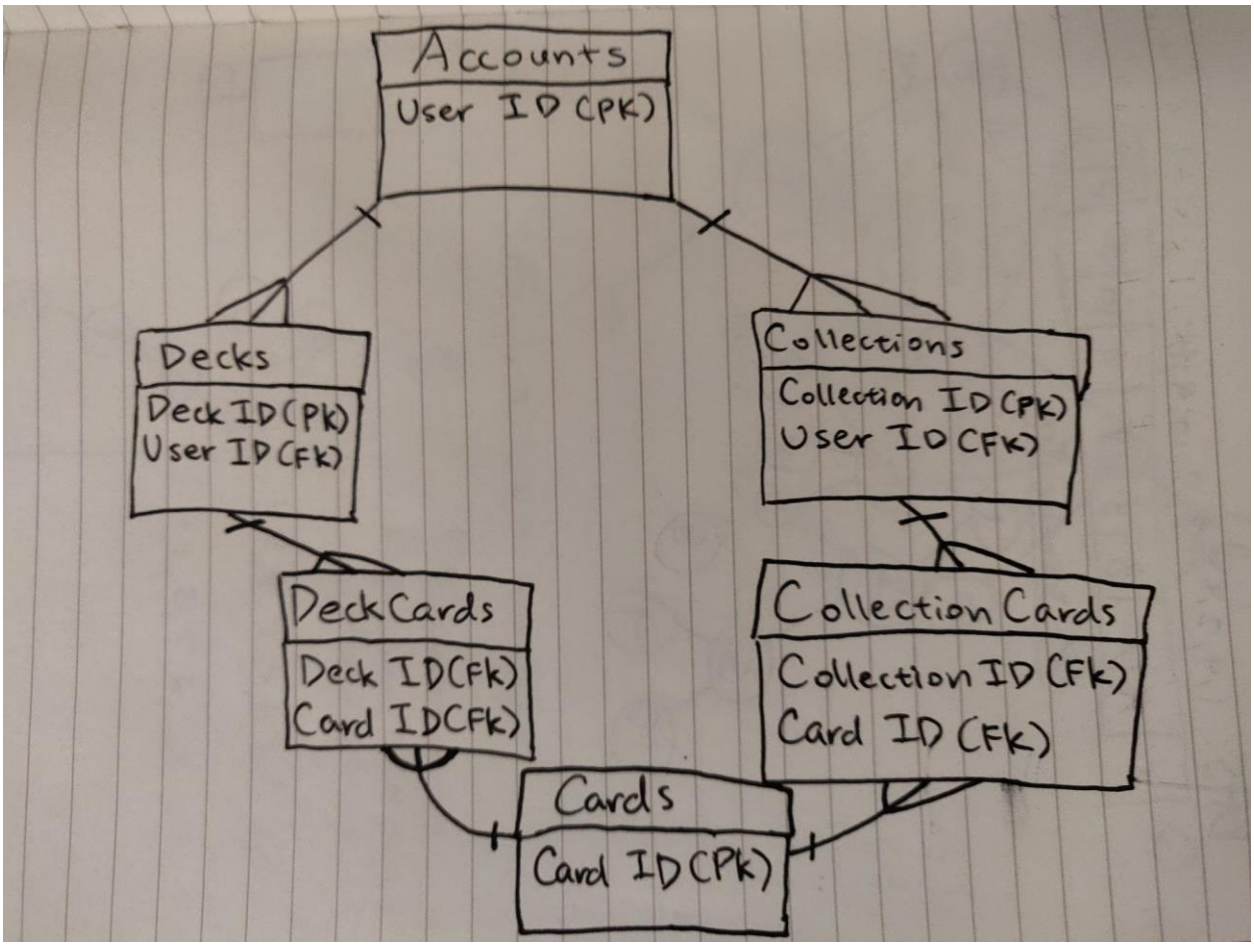
Johnny Chong Project Step 4 Draft Version: Design HTML Interface
Eric Lester Trading Cards Finance Group #47

flip1.engr.oregonstate.edu:4746

From the peer reviews the primary criticism relates to the previous review done by the TA saying that CardDetails relates too much to Cards and should be combined. The only reason this table was originally added was to add more kinds of relationships in the table but was redundant and now is not a part of our database.

Parts of the review that we did not implement included a misrepresentation of what Cards was meant to be in the database. Cards is meant to define a card someone would own without any information defined by the user. The user has the option to add predefined cards to either their collection or their decks. Adding new cards to the database would be a backend responsibility and not a capability of the user. Updating these cards in real time would be a maintenance issue for further development that goes outside the scope of this class(including a possible web scraper to identify and average card prices over multiple websites). I think the confusion over Cards comes from the ForSale variable that is meant to be an extension to an alternative site that would sell cards and notify a user if the card is for sale.

Changes that were made from our own discretion were adding a variable for quantity of cards to CollectionCards and DeckCards since this would reduce the entries in the table since multiple of the same cards could be represented as a single entry. Also to add names to both cards and collections to allow users to identify them.



- Accounts

Accounts is used to store user information and relate ownership of other data tables.

  - userID: int, auto_increment, unique, not NULL, PK
  - userName: varchar, not NULL, unique
  - userPassword: varchar, not NULL
  - userEmail: varchar, not NULL , unique
  - Relationship: 1:M relationship between Accounts and Collections is implemented with userID as a FK inside of Collections
  - Relationship: 1:M relationship between Accounts and Decks is implemented with userID as a FK inside of Decks

- Collections

Collections is a table used to store a user's collection of cards.

  - collectionID: int, auto_increment, unique, not NULL, PK
  - userID: int, not NULL, FK

    - collectionName: varchar, not NULL
    - Relationship: M:1 relationship between Collections and Collections Cards is implemented with collectionID as a FK inside CollectionCards, CollectionCards is the intersection table creating a M:M relationship between Collections and Cards.

- Decks

Decks are smaller versions of collections that have specific domains and have cards that are not a part of a user's collection.

    - deckID: int, auto_increment, unique, not NULL, PK
    - userID: int, not NULL, FK
    - deckName: varchar, not NULL
    - Relationship: M:1 relationship between Decks and DeckCards is implemented with deckID as a FK inside DeckCards, DeckCards is the intersection table creating a M:M relationship between Decks and Cards.

- Cards

Cards are the entities that will be a part of Account's Collections and Decks, and describe specifics about the card.

    - cardID: int, auto_increment, unique, not NULL, PK
    - cardName: varchar, not NULL, unique
    - cardDescription: varchar, not NULL
    - cardPrice: decimal
    - cardForSale: Bool
    - Relationship: M:1 relationship between Cards and CollectionCards is implemented with cardID as a FK inside CollectionCards, CollectionCards is the intersection table creating a M:M relationship between Collections and Cards.
    - Relationship: M:1 relationship between Cards and DeckCards is implemented with cardID as a FK inside DeckCards, DeckCards is the intersection table creating a M:M relationship between Decks and Cards.
    - Relationship: 1:1 relationship between Cards and CardDetails is implemented with cardID as a FK inside of CardDetails

- CollectionCards

CollectionCards is the intersection table creating a M:M relationship between Collections and Cards.

    - collectionCardsID: int, auto_increment, unique, not NULL
    - collectionID: int, not NULL, FK
    - cardID: int, not NULL, FK
    - cardQuantity: int not NULL

- DeckCards

DeckCards is the intersection table creating a M:M relationship between Decks and Cards.

- deckCardsID: int, auto_increment, unique, not NULL
- deckID: int, not NULL, FK
- cardID: int, not NULL, FK
- cardQuantity: int not NULL

Schema