Colin Knebl, Capstone Database Design

App: QuizMe, A flashcard based quiz application
Database: MongoDB

MongoDB was chosen as the database because the amount of data required is low, there are few relationships among the data, and I am more familiar with NoSQL databases such as MongoDB.

Number of data models: 3

- *User*: to contain the user's data (email, name, password)
- *Deck*: contains the deck title, if the deck is public (visible to other users of the app), and the cards contained in the deck
- *Card*: the text content on each side of the card, and if the card is "marked" (will allow the user to filter between reviewing only "marked" or only un-"marked" cards.

Number of collections: 1

- There will be a single collection will all documents embedded. There are two reasons for using the single collection and embedding all of the documents
    1. This design will optimize DB queries. When a user logs in, there will only be one network query to retrieve all user data, decks, and cards
    2. I originally thought I would split the user's decks into a separate collection, and the User.deck would be an array of deck IDs. I thought this approach would be easier to share decks between users. However, if a user 'clones' another user's deck and makes changes to it, the original deck owner should not get those changes (they may not want them). For that reason, all decks and cards belong solely to the owning user, and it makes sense to embed these documents within the User model.

The User document will have the flashcard Decks embedded, and the Deck document will contain the Card documents embedded.

Colin Knebl, Capstone Database Design

```
User:
{
    id: string;
    email: string;
    firstName: string;
    lastName: string;
    password: string;
    decks: Deck[];
}

Deck:
{
    id: string;
    title: string;
    public: boolean;
    cards: Card[];
}

Card:
{
    id: string;
    sideA: string;
    sideB: string;
    marked: boolean;
}
```