FLASH CARD REPORT

Author

RAJ SIRUVANI
21F1005673
21f1005673@student.onlinedegree.iitm.ac.in
A Final Year Undergrad Student in Aerospace Engineering, likes Physical Activities and Music
And always tends to have a positive Attitude for every situation encountered in life.

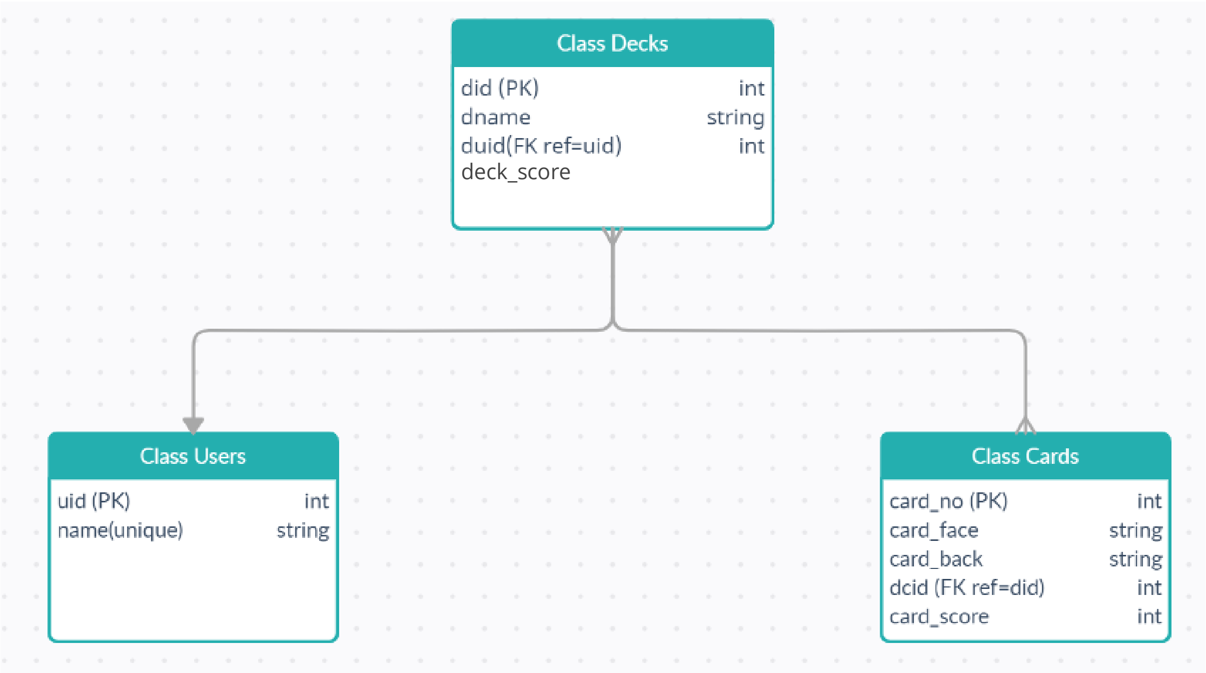
Description

Basic Idea is to let User add/Update/Delete the Flashcards so that the User could have their Own customized application for memory training, so this could be done by having 3 Tables Storing information about USERS, DECKS, CARDS and connected by relations (Backend).

Technologies used

Packages Used: flask, flask_restful, flask_sqlalchemy.
Flask used as a Web framework by Python Packages, used for developing Web Applications easily (It's a good place to start for beginners.
flask_restful is used for building REST API's of our flask Application and also testing them with appropriate inputs
flask_sqlalchemy is used for Database Linking and querying the database to retrieve information from the database and present to the View

DB Schema Design



Class Decks	
did (PK)	int
dname	string
duid(FK ref=uid)	int
deck_score	

Class Users	
uid (PK)	int
name(unique)	string

Class Cards	
card_no (PK)	int
card_face	string
card_back	string
dcid (FK ref=did)	int
card_score	int

A many-to- many relationship is established between Users and Decks, connected by uid so That each and every deck created could be uniquely identified by any user.

All Cards are stored in Cards Database and each card is uniquely identified by card_no and Also each card has a deck_id (Which would let us identify each card's parent deck and in turn we would also know which user created that card.

API Design

User -> POST (API for Creating a new User)

Deck -> GET(Retrieves All the Decks created by a User)

->POST(Creating a New Deck)

->PUT(Updating the Name of a Particular Deck)

->DELETE(Deleting the Deck& all the Cards inside that Deck)

Cards->GET(Gets all the Cards inside a deck created by a user)

- ➔ POST(Adding new Cards to a deck by some User)
- ➔ PUT(Updating a Particular card by card_no by a User)
- ➔ DELETE(Deleting a card in a deck by the card number)

Score -> GET (Retrieves the Average Score of the Deck(Last visited Score)

Architecture and Features

Each User can have different(unique) Decks and could perform CRUD operations on all the Cards inside the deck, all the Controllers, DB Models and API are in the same 'main.py'

All the HTML documents are stored inside the templates folder.

Around 15 html Webpages have been designed for carrying out various operations in the flashcard with the help of 10 @app.route() controllers(Starts from line 206)

Features Implemented: User Login, Deck Management (CRUD operations successfully working), Review, APIs for Interaction to the Database, (Input Validation also done)

Video

Project Overview (Please watch in 1.25x or 1.5x) As Accidently the Duration has been exceeded by 3min.