CSC 667/867 Internet App Design - Term Project

UNO Multiplayer Card Game

Milestone 3: Web App Entity Design

Submitted: October 19, 2020

Developed by:

Akhil Gandu, Allen Sun, Katie Kennedy, & Roberto Herman Valdivia

Prof. John Roberts, SFSU, Fall 2020

GitHub Repository

Our source code for this project can be found at this link:
https://github.com/sfsu-csc-667-fall-2020-roberts/term-project-akhil-allen-katie-roberto

Database Entities

players

The players table has one entry per unique account. Each user must have an account to play Uno on our application. A user is eligible to enter a multiplayer game of Uno, but is not necessarily currently playing an active game.

game

Each entry in the game table contains the live data necessary for a single active game, including users playing, score, status of the deck, cards in players' "hands," players' turns, and direction of turn rotation.

game_players

This table is a subset of four users who are playing against each other in a live game. The game_players table is a member of the game table. Information about a single player's hand (e.g. the cards they are holding) and other gameplay-relevant data is stored in each of the four entries in this table.

deck

Each entry in the deck table is a card. Because Uno is played with only one deck, the card makeup of the deck is important to keep data on. The deck table also allows for random shuffling with the start of each new game.
Solution to storing cards in the database.

- Solution 1:
  Create a Class in the application named like CardDistribution, containing an array of 108 elements. Each card has an id of.
  When we read from the database, we deserialize the CardDistribution JSON into a javascript object and use it. When we want to update the deck after every move, we serialize the CardDistribution object into JSON and persist it in a column in the Game table.
- Solution 2:
  Create a Stack with 108 cards in random order in each single game. For each move, we just pop up a card from the Stack.

card

Each card in the deck has either a numerical value and color (red, green, yellow, or blue), has game instruction assigned to it, or is a colorless “wild” card.

game_move

The game_move table contains a list of entries required for the execution of a game player’s single turn. Each entry is a possible next move based on the data in the game table during an active game.

play_type

For each game_move, a player can either draw cards until a playable one is found, or play already in hand.

message

Each entity in the message table is a single chat message sent by one play to the others during an active game.

result

The result table records the outcome of each game by rank with 1 being the game winner, and the 3 others as tiered “losers” based on how many cards each still held at the end of the game.

lobby

The lobby table contains entries of subgroups of players on the app, including an entry of itself with all players currently waiting to play or are currently playing on the app. These subgroups are referred to as rooms. Each room can contain a group of players actively playing a game, or players waiting to play together before starting the game.

Diagram of Entity Relationship Design