Code Team 6 //Milestone 3 & 4

by Wenjie Ye, Joseph Kois, Hira Afzal, Ivan Cebreros

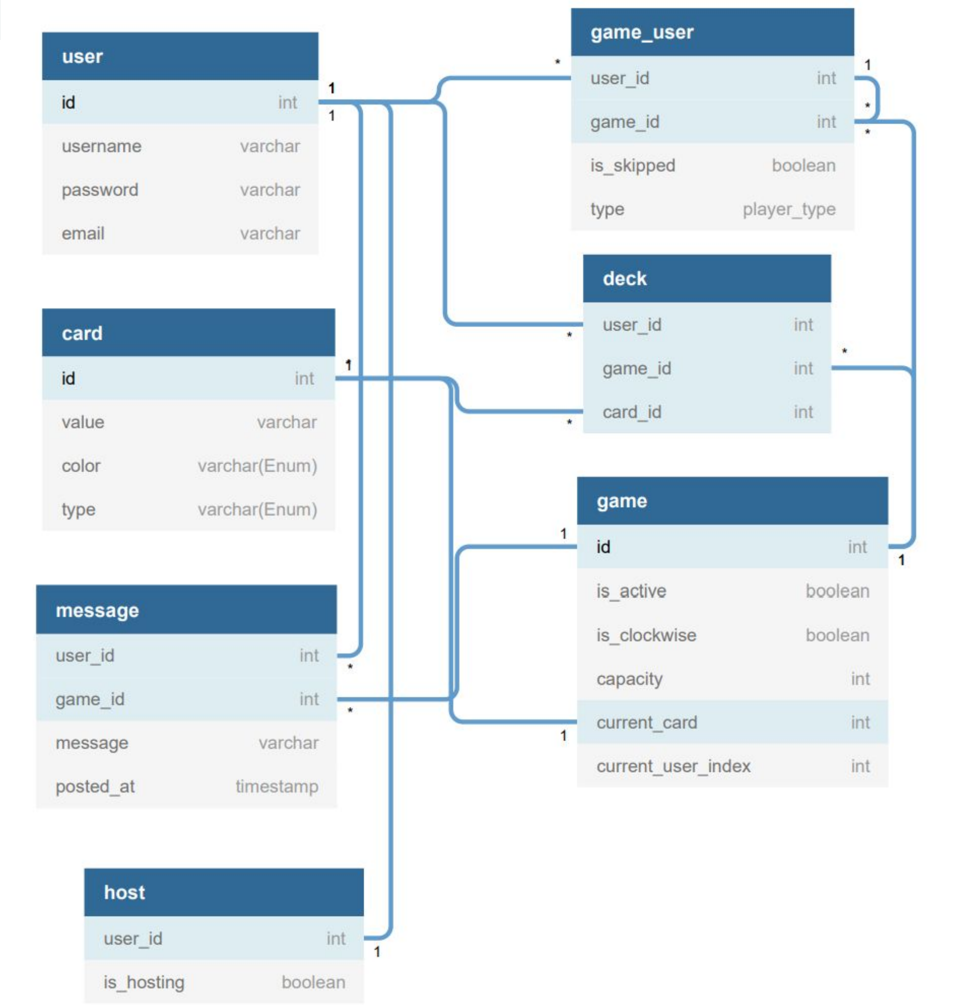
Table of Contents

- M3 //Entity Design
  - Entity Design
- M4 //Application Logic
  - Game
  - User
  - Deck
  - Message

M3

//Entity Design

Entity Design



user	
id	int
username	varchar
password	varchar
email	varchar

card	
id	int
value	varchar
color	varchar(Enum)
type	varchar(Enum)

message	
user_id	int
game_id	int
message	varchar
posted_at	timestamp

host	
user_id	int
is_hosting	boolean

game_user	
user_id	int
game_id	int
is_skipped	boolean
type	player_type

deck	
user_id	int
game_id	int
card_id	int

game	
id	int
is_active	boolean
is_clockwise	boolean
capacity	int
current_card	int
current_user_index	int

M4
//Application Logic

User Actions

- Login and Sign up
- Join multiple games
  - If game is active, user can only be able to observe the game
- Leave a game
- Create a new game
  - User who created the game will become the host for that game
  - Host is the only one who can start the game
- Send Messages

User Actions - In their turn

- Case 1: current card is +2 or +4
  - Has +2 or +4 card
    - Play the card and end the turn
  - Otherwise,
    - Draw 2 or 4 cards and end the turn
- Case 2: no cards can be played
  - Draw a card and play the card if it can be played, then end the turn
- Case 3:
  - Play a card
    - If this card is the last card, win the game

Game Events

- +2 card
  - Next player will draw 2 cards if not holding +2 cards and skip their turn
- +4 card (Wild draw 4 card)
  - Change the color
  - Next player must draw 4 cards and skip their turn
- Wild
  - Change the color
- Reverse
  - Reverse direction of play
- Skip
  - The next player is skipped

API

1. getCurrentUserCount
2. getGameList
3. getGameInfo
4. getPlayerList
5. getCardInfo
6. skipUser
7. createGame
8. leaveGame
9. getMessageList
10. getCardsOnHand
11. drawCard
12. updateColor
13. updateDirection
14. updateTurn
15. deleteGame

Game

1. GET /game
   - Description
     - To get a list of games in the lobby
   - Parameters
     - N/A
   - Responses
     - 200 - success
       Example Value: {
       “gameList” : [1, 2, 3, 5, 6]
       }
     - 403 - unauthenticated

2. GET /game/info?gameId={gameId}&userId={userId}
   - Description
     - To get the game infos
   - Parameters
     - gameId: 1
     - userId: 1
   - Responses
     - 200 - success
       Example Value: {
       “isClockwise” : true,
       “playerList” : [{“id”: 2, “name”: “John”, “hold”: 7}, {“id”: 3, “name”: “bob”, “hold”: 5}],
       “isActive”: false,
       “isSkipped: false,
       “currentPlayerId”: 2,
       }
     - 403 - unauthenticated

User

1. POST /login
   - Description
     - User logs in to the application
   - Parameters

     ```
     Example Value: {
         “email” : “john@mail.com”,
         “password”: “test123”
     }
     ```
   - Responses
     - 200 - success

       ```
       Example Value: {
           “userId”: 1
       }
       ```
     - 401 - invalid authentication credentials

2. POST /signup
   - Description
     - User signs up an account
   - Parameters

     ```
     Example Value: {
         “email” : “john@mail.com”,
         “username”: “JohnDoe”
         “password”: “john123”
     }
     ```
   - Responses
     - 201 - successfully created user

       ```
       Example Value: {
           “userId” :1,
       }
       ```

Deck

1. GET /deck/hold?userId={userId}&gameId={gameId}
   - Description
     - Fetch all the cards for a user in a specific game
   - Parameters
     - userId: 1
     - gameId: 1
   - Responses
     - 200 - success

       ```
       Example Value: {
           “hold”:[{  “id”:1
                  “value”: 8
                     “color”: “red”
                  “type”: “normal”
           }],
       }
       ```
     - 403 - unauthenticated

2. POST /deck/drawcard
   - Description
     - User draws cards
   - Parameters

     ```
     Example Value: {
         “userId” : 1,
         “gameId” : 1,
         “cardCount”: 2
     }
     ```
   - Responses
     - 201 - success

       ```
       Example Value: {
           “status”: 1,
           “success”: “user draw 2 cards”
       }
       ```
     - 403 - unauthenticated

Message

1. GET /message/{gameId}
   - Description
     - To get a list of messages in a specific game
   - Parameters
     - N/A
   - Responses
     - 200 - success

       Example Value: {
           “messages”: [{name: “John”, message: “Nice!”}]
       }
     - 403 - unauthenticated

2. POST /message
   - Description
     - To post a message in a game
   - Parameters

     Example Value: {
         “userId” : 1,
         “gameId”: 3,
         “message”: “Nice!”
     }
   - Responses
     - 201 - success

       Example Value: {
           “status”: 1
       }
     - 403 - unauthenticated