

# **MILESTONE 3: API DESIGN**

By Team Project2 (Letter J)

# Game Overview

Our project is an online multiplayer UNO game where players can create or join game rooms, draw and play cards, and compete to get rid of all their cards first.

Each turn, players can match a card by color or number, or play special cards such as Skip, Reverse, Draw Two, Wild, and Wild Draw Four.

The game follows standard UNO rules without variations.



# List of all endpoints

- **Game management endpoints**
  - **Create Lobby:** Creates a new lobby for players to join and ready up before game begins
  - **Join Lobby:** Let logged-in players enter an open lobby and appear as a player
  - **Start Game:** Triggered by lobby host to create game instance: shuffle and deal cards and move all players into game room
  - **Get Game State:** Retrieve current state for player
- **Game action endpoints**
  - **Draw Card:** Allows players to draw from the deck
  - **Play Card:** Plays a card from the hand and applied effects
  - **End Turn:** Passes turn to next player ID
  - **End Game:** When players hand reaches zero
  - **Send Chat Message:** Posts a text message to all participants in the lobby or game room

# **DEEP DIVE INTO 2 COMPLEX ENDPOINTS**

# Draw Card Endpoint

1. **Endpoint Name:** Draw Card
2. **HTTP Method & Route:** POST  
`/api/games/:game_id/draw`
3. **Purpose:** Current player draws card from the draw pile
4. **Authorization:**
  - a. User must be authenticated
  - b. User must be a participant in the game
  - c. User must be the current player
5. **Request Body:** —
6. **Validation Checks:**
  - Game exists and user is playing
  - Caller is `current_player_id`
  - Draw pile available
7. **State Updates:**
  - Move top card to caller's hand
  - Update player's `hand_count`
  - Advance turn
8. **Success Response:** 202 Accepted
9. **Error Cases:**
  - 403 not your turn
10. **Socket.io Events:**
  - `game:draw:completed` → private to drawer (cards drawn)
  - `game:hand:update` → private to drawer (new hand)
  - `game:state:update` → game room (pile)

# Send Chat Message Endpoint

1. **Endpoint Name:** Send Chat Message
2. **HTTP Method & Route:** POST  
`/api/games/:game_id/chat`
3. **Purpose:** Send a chat message to all players in the game room.
4. **Authorization:**
  - a. User must be authenticated
  - b. User must be a player in this game
5. **Request Body:**

```
{ "text": "string" }
```
6. **Validation Checks:**
  - Game exists
  - Caller is participant
  - text present and within max length
7. **State Updates:**
  - Insert into chat (game\_id, user\_id, text)
  - Update game chat for all
8. **Success Response:** 202 Accepted
9. **Error Cases:**
  - 400 missing text
  - 403 not in game
  - 404 game not found
10. **Socket.io Events:**
  - chat:message:new → game room

# Socket.io events summary

**Game domain:** Handles all in-game actions and updates, including when players draw or play cards, when the turn changes, and when the discard pile or top card updates. It manages the real-time flow of gameplay for everyone in the match.

**Lobby domain:** Manages all pre-game coordination, such as when players join or leave the lobby, toggle their ready status, and when the game officially starts.

**Chat domain:** Handles real-time messaging within a lobby or game room, ensuring that every player immediately sees new messages sent by others.