Dev Mountain

Foundations Capstone Plan

David Lay

August 2023

## Summary

For my Capstone Project, I will be recreating a simplified version of a Star Wars tabletop game, *Epic Duels.* The app will allowusers to select a character and battle an AI opponent by playing attack and defense cards.



## Front End

Single Page:

* + Heading
  + Character/card area
  + Action area
  + Buttons/forms
    - Select character
    - Start Game
    - Play a card
    - Log game
    - Start a new game

Styling

* + Ids for each char with custom style
  + Include uniform style for other page elements, mimic Star Wars theme
  + Background images using various stock
  + Hover style on buttons coded sections
  + See Wireframe below for rough layout

## Data Tables

* + Characters
    - (name, starting health, image, status)
  + Cards
    - Each entry represents a character card
    - Values: (id, **CharacterID**, attack, defend, status)
  + Stats
    - Active game info will append Stats table at game end

## Key Features (back end)

* + Select Character (GET)
  + Start game (POST)
    - Get random AI opponent
    - Get a hand (5 random cards)
  + Play A Card (POST)
    - Request
      * User selects a card, which modifies the active game data
    - Response
      * Select a random CPU card
      * resolve the outcome
      * Updates the health values on character table
      * Send health outcome back to user
      * Send an “End of Game” status/rule as needed
  + Log/ Save game stats (PUT)
    - If the game is over, the user can send the steps
    - TABLE of just stats, history
  + End Game (DELETE)
    - User can select Clear the current game table

## 

## Wireframe - main page

## 

## Interactions

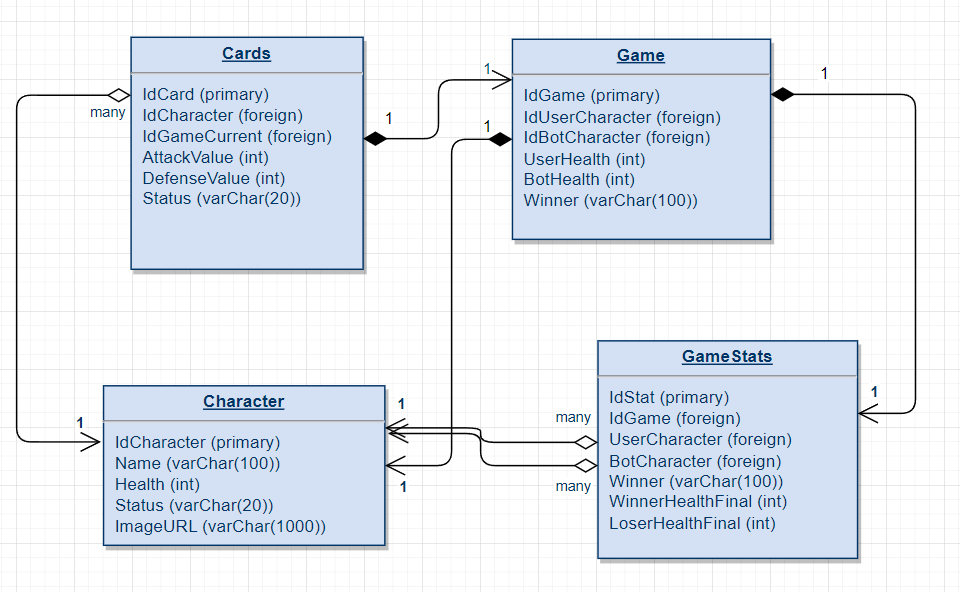
## 

## 

## Database schema

Requirements adjusted slightly. “Game” table was dropped. Instead, and object for active game is housed on the server.

Most of the values are consistent; see seed.sql file for final.



## Requirements Matrix

Indicates planned elements and self-evaluation.

| **list** | **Category** | **Item** | **Req** | **Description** | **Planned** | **points** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Planning | Wireframe | N | project includes a wireframe for each view | yes | 5 |
| 2 | Planning | MVP list | Y | project includes a list of MVP features \* | yes | 5 |
| 3 | Planning | Data Model | N | project includes a data model | yes | 5 |
| 4 | MVP | 3 Features | Y | app has at least 3 main features \* | yes | 5 |
| 5 | MVP | Request Handlers | N | front end makes a request to the server and handles the response \* | yes | 5 |
| 6 | MVP | Interactive | Y | front end is interactive \* | yes | 5 |
| 7 | MVP | Styling | Y | app has custom styling \* | yes | 5 |
| 8 | Front End | 5 Semantic Tags | Y | app has at least 5 semantic tags \* | yes | 5 |
| 9 | Front End | 1 view | Y | app includes 1 view \* | yes | 5 |
| 11 | Front End | Flexbox | N | styling includes flexbox | Yes | 5 |
| 14 | Server | GET | Y | app includes a GET endpoint and handler function \* | yes | 5 |
| 15 | Server | POST | Y | app includes a POST endpoint and handler function \* | yes | 5 |
| 17 | Server | DELETE | N | app includes a DELETE endpoint and handler function | yes | 5 |
| 18 | Server | Sequelize | N | app utilizes Sequelize | yes | 5 |
| 19 | Server | Controler file | N | project includes at least 1 controller file | yes | 5 |
| 20 | Database | Seed | N | project includes a seed file or function | yes | 5 |
| 21 | Database | Table | N | app uses 1 table | yes | 5 |
| 22 | Database | Additional Table | N | app uses 1+ additional tables | yes | 5 |
| 23 | Database | Join and Foreign Key | N | app uses a foreign key and join | yes | 5 |
| 24 | Presentation | Purpose and Demo | Y | discusses project purpose and demonstrates MVP \* | yes | 5 |
| 25 | Presentation | Does not present bugs | N | does not discuss broken/unimplemented features | yes | 5 |
| 26 | Presentation | 2 to 3 min | N | recording is between 2-3 minutes | yes | 5 |
|  |  |  |  |  | Total attempted | 110 |

### 