

# CPSC 304 Project Cover Page

Milestone #:3

Date: October 27, 2023

Group Number: 111

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Neil Li	22681720	a0y6f	neillyx26@gmail.com
Kevin Liu	94200474	z4f4p	kevinliu2002@yahoo.ca
Dane Urban	67510214	g1v5g	z5363301@ad.unsw.edu.au

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

## **Summary:**

Our team intends on creating a deck building database for a card game for our project, in which players can look up decks they can build, and find the cards necessary for that deck. They can also see the listings for those cards and where they are sold, as well as who are the artists that created that card.

## **Project Breakdown:**

From a high-level perspective, we plan on approaching the project by splitting it into 2 problem domains; frontend and backend. We plan to utilize a React frontend with a Java Spring backend and implement RESTful API architecture for the separate domains to communicate with each other. By approaching the problem in this manner, we can ensure that frontend and backend tasks remain divided and have greater ease when breaking down problems into tickets.

## **Features:**

- Get Cards (Parameters: Search: String, Type: Enum, Set: Enum, Artist: Enum)
  - Returns a list of all cards fitting the parameter criteria
  - We want this to show the minimal amount of information about all the cards
    - Additional information can be collected from GetCard
  - Note: Parameters can be empty or null.
- Get Decks (Parameters: Search: String, Form: Enum)
  - Returns a list of all decks fitting the parameter criteria
    - Have return maybe be id, photo, form and name. We can then click on this to get more in depth info about the deck
    - We want this to show the minimal amount of information about all decks
  - Note: Parameters can be empty or null
- Get Card (Parameter: Id)
  - Returns all information we want to display about a card
- Get Deck (Parameter: Id)
  - Returns all information about a deck we want to display
- Create Account (Parameter: username, password)
  - Creates an account for a user in which they can create decks
- Login Account (Parameter: username, password)
  - Logs a user into an account
- Modify Deck
  - Modifies a deck. User must be logged in and own the deck they are modifying
- Modify Card
  - Modifies a card. User must be logged in and have perms to modify the card
- Delete Card
  - Deletes a card. User must be logged in and have perms to modify the card

- Delete Deck
  - Deletes a Deck. User must be logged in and own the deck they are deleting
- Create Card
  - Adds a new card to database. User must be logged in and have perms to modify the card.
- Create Deck
  - Creates a new deck. Users must be logged in to create a new deck.
- Get Listings
  - Gets all listings in database
- Get Card Listings
  - Gets all listings for a given card
- Get Artist
  - Gets artist details
- Get Creator
  - Gets creator details and list of ids of decks created by the given creator
- 

### Frontend Tasks:

- Create an interactable interface to insert new card data and post (/cards/upload)
  - This should be done in a form-like manner
- Create an interactable interface for creating a deck
  - This will be done by allowing a user to add a quantity of a card to add to a deck
  - Cards can only be added if they have been previously uploaded
  - By default, all cards appear (perhaps add pagination), however searching functionality will be provided to filter
- Create an interface for viewing cards
  - Add the ability to select cards and look at the details of individual cards
  - Shows all (paginated) cards by default
  - Ability to filter
- Create an interface for viewing decks
  - This should show details about the deck as well as all the cards in the deck
- Create an interface for viewing the creator
  - This should show details about a creator as well as all the decks they have created
- Create an interface for viewing artist
  - This should show details about an artist as well as all of their card art
- Create an interface for viewing card listings
  - Shows all (paginated) listings on all cards by default
  - Can filter listings based on properties such as card name, art, artist

### Backend Tasks:

- Create POST request for inserting new cards (/cards/upload)
- Create POST request creating new decks (/decks/upload)
- Create POST request for adding new listings (/listings/upload)

- Create GET request for getting cards (/cards)
- Create GET request for gettings decks (/decks)
- Create GET request for getting cards in deck (/decks/deck)
- Create GET request for getting listings (/listings)
- Create GET request for getting listing for certain card (/listings/card)
- Create GET request filter system (/cards/filter)
  - Users can sort cards using a filter based on properties such as type, stat-able, art, set
- Create GET request for searching for card (/cards/card)
  - This returns all versions of the card including those with different artwork
- Create Get request for getting artist
- Create Get request for getting Creator
- Create Delete request for deleting card (/cards/card)
- Create Delete request for deleting deck (/decks/deck)

Task	deadline	responsibility
<ul style="list-style-type: none"> <li>- Create POST request creating new decks (/decks/upload)</li> <li>- Create GET request for gettings decks (/decks)</li> <li>- Create Get request for getting Creator</li> <li>- Create Delete request for deleting deck (/decks/deck)</li> </ul>	November 8th	Neil
<ul style="list-style-type: none"> <li>- Create POST request for inserting new cards (/cards/upload)</li> <li>- Create GET request for getting cards (/cards)</li> <li>- Create Get request for getting artist</li> <li>- Create Delete request for deleting card (/cards/card)</li> </ul>	November 8th	Kevin
<ul style="list-style-type: none"> <li>- Create GET request for getting cards in deck (/decks/deck)</li> </ul>	November 15th	Neil
<ul style="list-style-type: none"> <li>- Create an interactable interface to insert new card data and post (/cards/upload)</li> <li>- Create an interactable interface for creating a deck</li> <li>- Create an interface for viewing cards</li> <li>- Create an interface for viewing decks</li> </ul>	November 12th	Dane

<ul style="list-style-type: none"> <li>- Create POST request for inserting new cards (/cards/upload)</li> <li>- Create POST request creating new decks (/decks/upload)</li> <li>- Create GET request for getting listings (/listings)</li> <li>- Create GET request filter system (/cards/filter)</li> </ul>	November 22nd	Neil
<ul style="list-style-type: none"> <li>- Create GET request for searching for card (/cards/card)</li> <li>- Create GET request for getting listing for certain card (/listings/card)</li> </ul>	November 15th	Kevin
<ul style="list-style-type: none"> <li>- Create an interface for viewing decks</li> <li>- Create an interface for viewing the creator</li> <li>- Create an interface for viewing artist</li> <li>- Create an interface for viewing card listings</li> </ul>	November 22nd	Dane'
Connect front and back end	November 29th	Dane, Kevin, Neil