Jude Rizzo, Manfred Tan, Jason Zheng

Jacket Games

1. The first page a user will see is a login page
    - This login page will be standard, and it will prompt the user for username and password, which will be stored in a user database with
    - If an account already exists, it will give the standard prompt
2. When the user logs in, there will be 3 options. This will be considered the “Main/Profile” Page, and of the three options, two of them will be large, center buttons that go in the center. The first button will be the option to “Create a New Deck”. The second option will be the option to play a new game given the selected decks.
3. On pressing the “Create New Deck”, the user will be lead to another page, where there will be a large table with four enteries that will have rows that look like the following:

Character	Type	Stats	Add/Remove from deck

    - In doing this, we will use bootstrap table and button features
    - These will come from a database of all of the characters in the game, their respective type (there will be NHL players and Pokemon), as well as some stats within their field (these stats will be used in a later described random algorithm for how to win the game.)
    - Each deck can only have one of each character, and there will be hundreds to choose from. [Extra - Make a deck]
        - Each deck will have 20 characters
    - The names of the characters in the deck will appear on the right side of the website, and once the deck is made, they will be stored as a dictionary in the userinfo database, (with the key being the deck name, and the items being a respective “deck”, aka a list of strings, with each string being the character name”
4. When the deck is finished, one may click a button at the top of the page (which will only be activated - it will be disabled with the bootstrap feature if the deck contains less than 20 characters)
5. Then the user, after creating their new deck will be sent back to the homepage
6. The “play” feature will always be against a computer, which will do the following.
    - The user will first be sent to a temporary page, where they will then be given the option to pick one of the decks they made. This will be coming from the database which involves all of their decks. Once they pick, and press submit, they are sent to the next page.

- First the computer will randomly select a category. This category will randomly determine the "type" of the card that will want to be played. From this, the user, of the three cards that will be randomly chosen from their deck to be put into their hand.
- The page will be split into three sections (top, middle and bottom). The top section will be the computers console. At the start of each round (there will be 5 cards), the computer will pick a type randomly (either Pokemon or NHL), and after that is made clear, the user will pick a card of that type from their hand. If they don't have any, then they will be able to send up a card of the other type and will automatically lose in that category. The cards that the user has in their hand will show their player name, and their stats on the bottom.
- Once the player chooses their card, the computer will randomly choose a card of that category from it's database. This will show up in the top third of the screen, and in the middle the winner will be decided. The score will be shown in the middle of the board, and will simply be dictated by how many wins there are. The first player (user or computer) to get to 5 wins will win the entire game, and it will show a brief victory or defeat screen and return back to the regular screen.
- The winner of each round will be computed by a ratio that takes statistics into account. It will be slightly different:

  For pokemon: We sum up all the 6 stats for each pokemon. Then we add them together. Next, we take a random number less than the sum. If it is less than the sum of the first pokemons stats, the first Pokemon wins, otherwise the second Pokemon wins.

  For NHL Players:

  You do the same thing for Pokemon, but for each player, their "stat" is considered their save% * 20 + goals. (this will account for scores, assists, and goals)

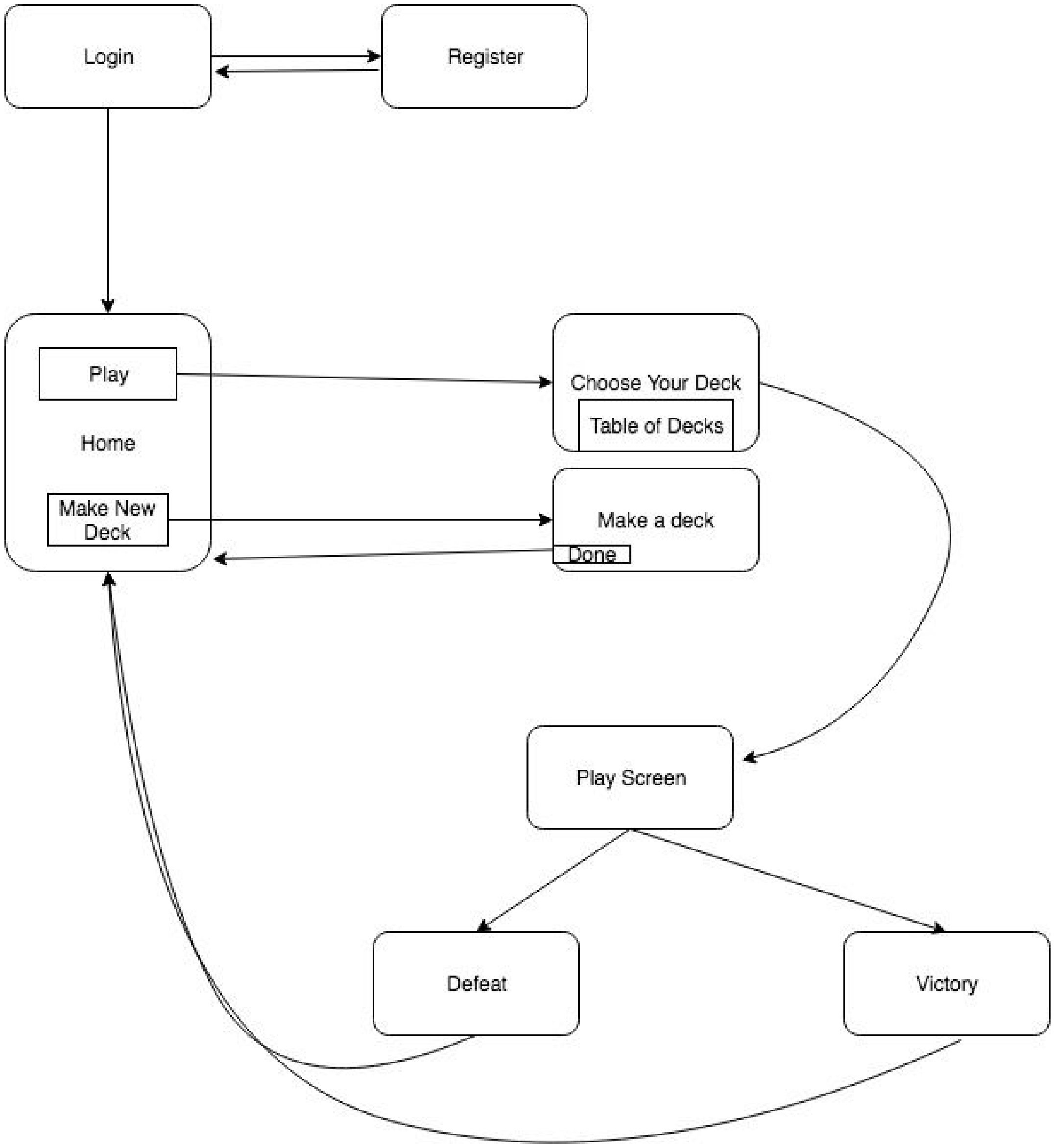
7. Minimum Viable Product:
    - First, we will need a successful login/register mechanism. We will store the user information into the user database.
    - Next we will need to have the characters input into the database. We will call these methods when starting up the website and permanently store the users into the characters database.
    - We also need a page to make a deck. The user will see the characters from the database and be able to choose 20.
    - We will create preset computer decks that the user will play against. We will hard code one deck that the computer will use. After successfully creating the game, we will allow the computer to have more decks.

    - We will need to create a method that determines which card wins when two cards are inputted into that method.
8. Extra Features:
    - Win loss ratios on each account showing
    - Pre-made decks for users if they don't want to make one and want to play
    - Game modes with 7 and 10 rounds
    - Different difficulties
    - Different background for the playing field or back of cards

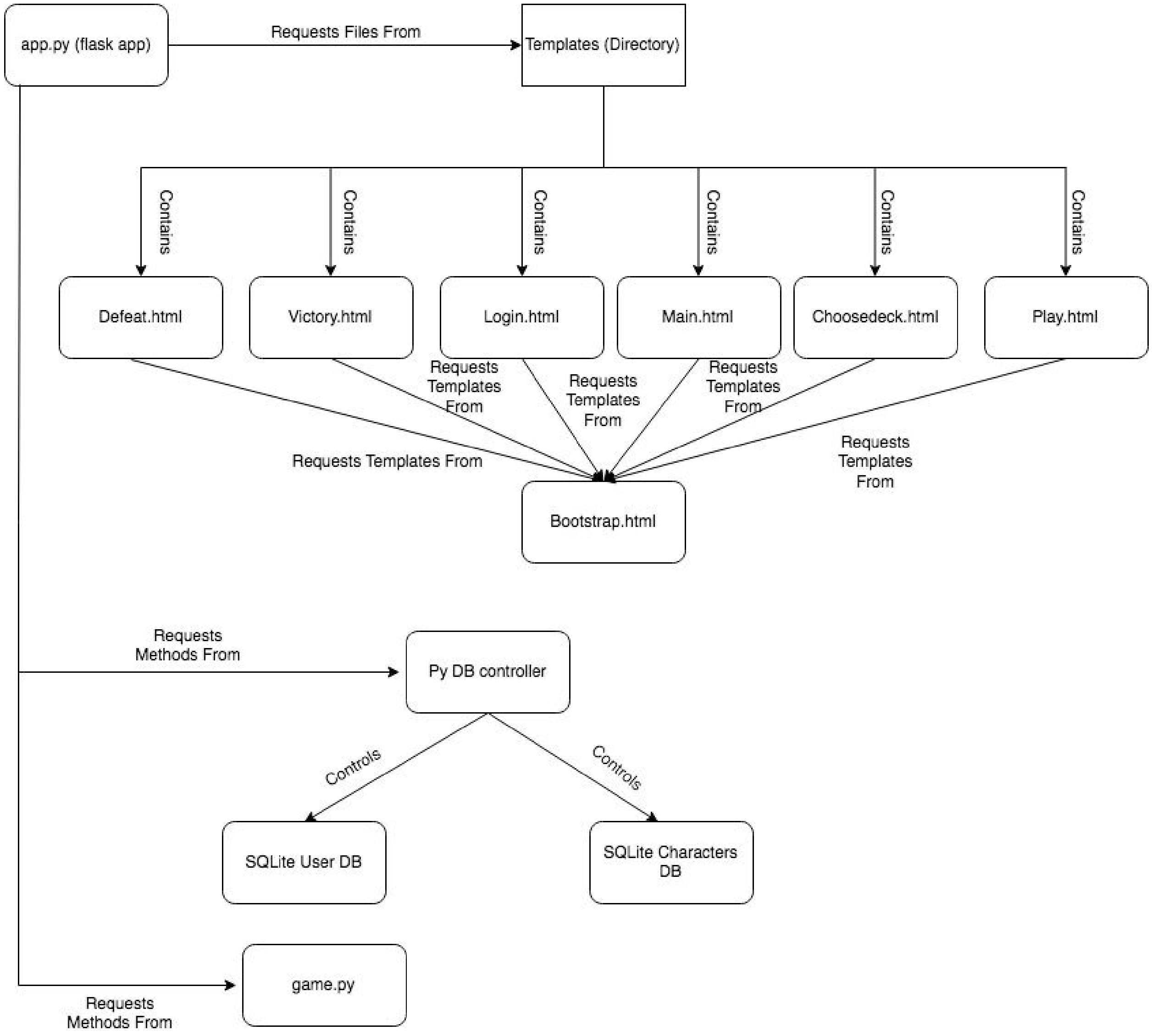
What We'll Need:

- Flask app with a minimum of 7 routes:
    - Login Page, Register Page, Main page, Create Deck, Play, Victory, Defeat, etc.
- HTML/Jinja templating for front end
- CSS + Bootstrap with the HTML/Jinja
- Database with 2 tables
    - Login Table storing Usernames and Passwords
    - Table storing all the characters and their information (as shown above)
- Python program to facilitate adding to database tables

Site Map:


Login
Register
Play
Home
Make New Deck
Choose Your Deck
Table of Decks
Make a deck
Done
Play Screen
Defeat
Victory

Component Diagram:



Site Deisgn:

Login Page

Jacket Games

Login:

Username: _________

Password: __________

Create Account Here

Register Page

Jacket Games

Create Account:

Username: _________

Password: __________

Confirm Password: _________

Login Here

Home Page

Home Jacket Games Log Out

Create a New Deck

Play

Create New Deck

Home Jacket Games Log Out

Save

CHARACTER	TYPE	STATS	
character 1	ex1	ex1	Add
character 2	ex2	ex2	Add
character 3	ex3	ex3	Remove

Play Page

Home Jacket Games Log Out

[ENEMY CARD HERE]

[YOUR CARD HERE]

End Page

Home Jacket Games Log Out

VICTORY / DEFEAT

Return to Home Screen

Database Layout:

Users

UserID	username	password	Decks

Character

Character Name	Character Type	Winning Percentage	API / origin

List of APIs:

Deck of Cards API:

- No quota
- Used to shuffle decks, make decks, add to decks etc

NHL API:

- No quota
- Stats on hockey players that will be converted into in game stats

PokeAPI:

- Limit of 100 API requests per IP address per minute
- Stats on Pokemon that will be converted into in game stats

Roles:

Jude Rizzo: Flask routes mentioned above, users must be logged in, and some error messages
Manfred Tan: Databases that store the decks user makes, and info from APIs
Jason Zheng: CSS on all the flask routes and on the playing cards