Project Milestone 4

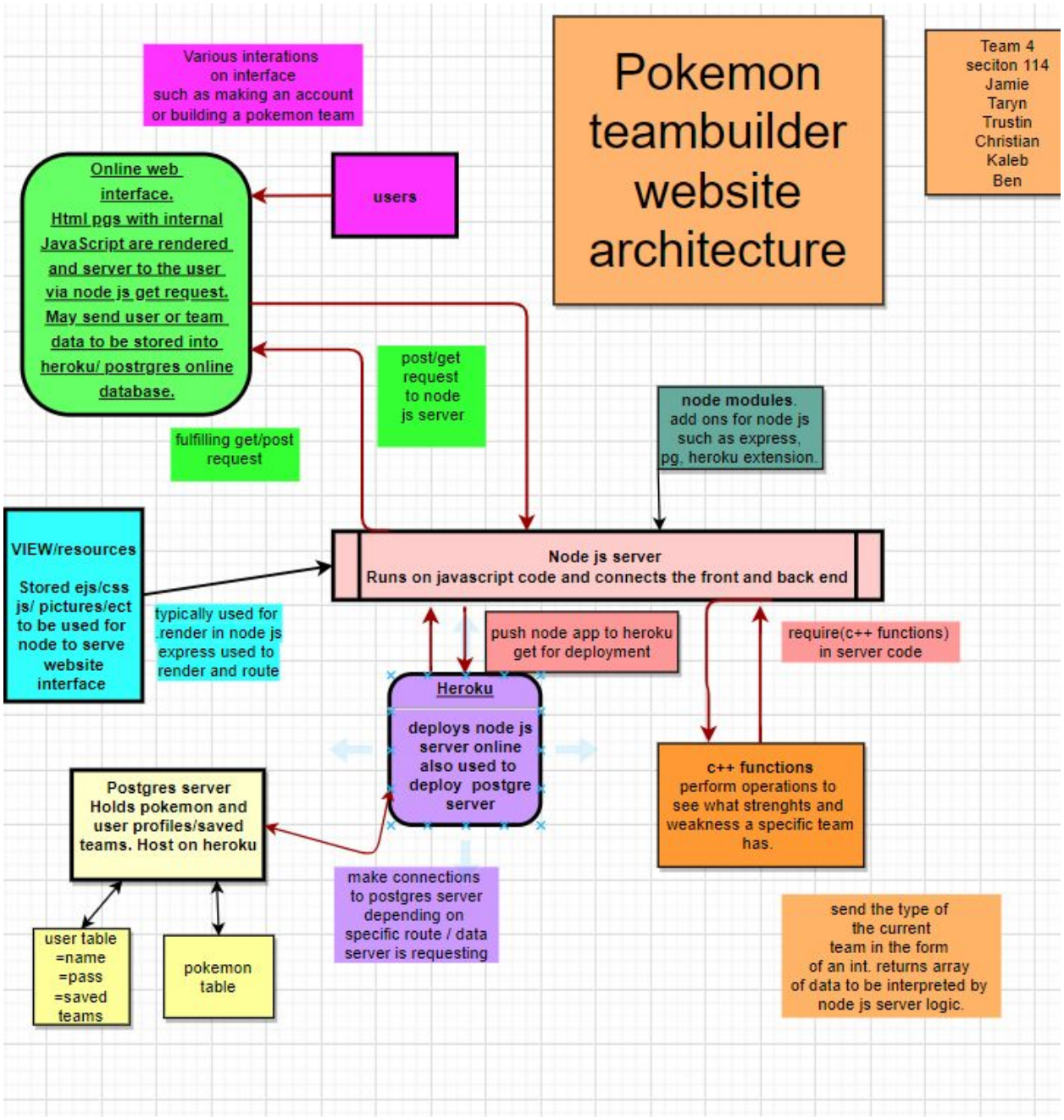
Group 114-4

Revised Features

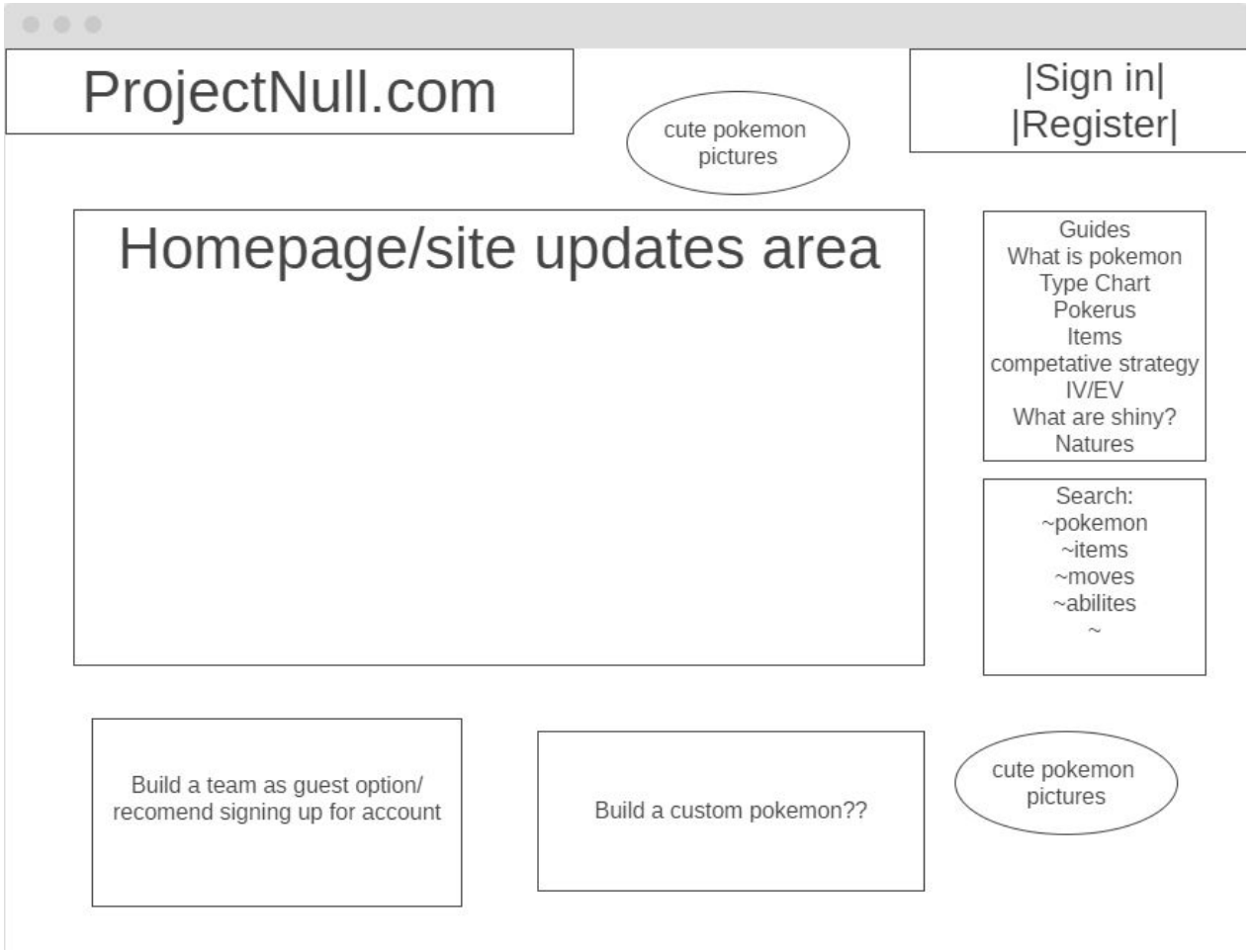
Highest Priority

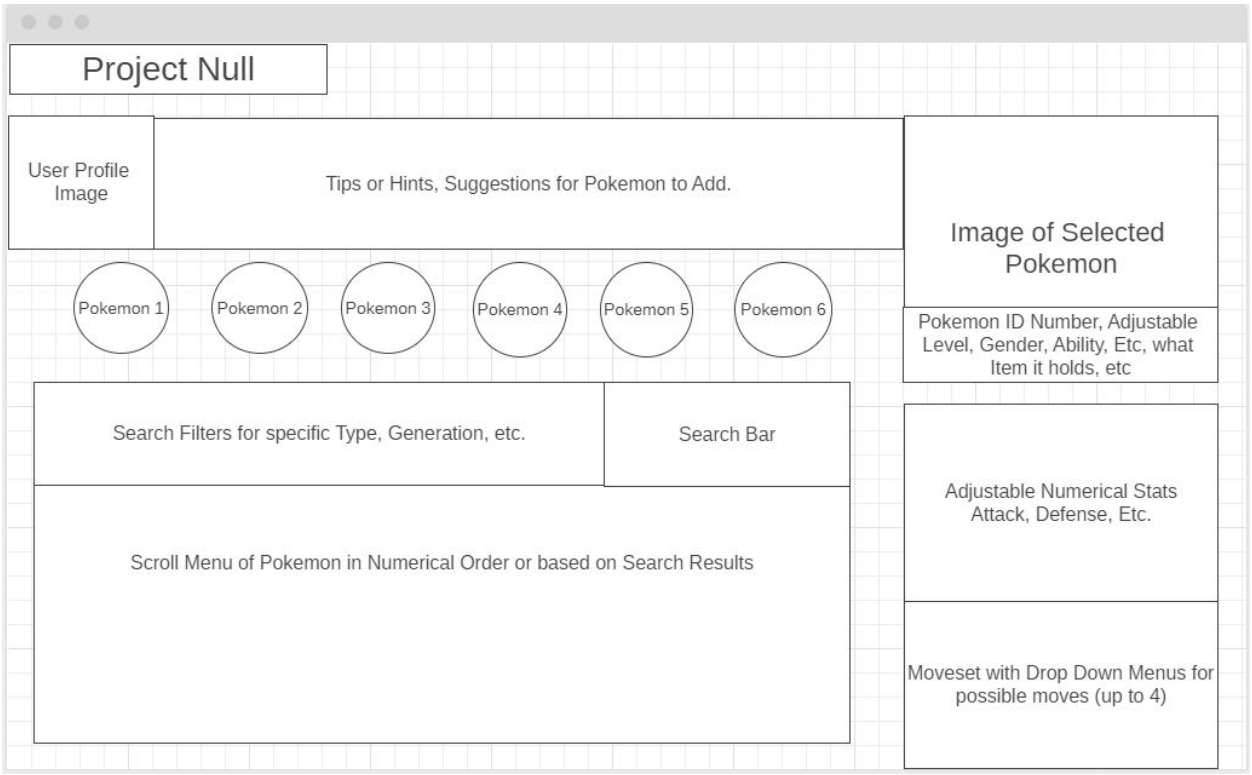
- Pokemon item/stat/img/type: page
- Team builder:
    - On a team builder page, users will be able to create Pokemon teams
    - Users will be able to search for the pokemon they want
- User Authentication
    - Last time we only had a database for users set up, however, we began working on the code to create user accounts and be able to sign up and log in.
- Dropping Team Suggestions
    - We figured out that team suggestions would need either machine learning or we can change it to make it really basic with just pokemon type suggestions.
- Drop friend functionality
    - We decided to drop friend functionality, simply because of our time constraint. It's not of a high priority at the moment, and we believe we can dedicate our time to better features that will further improve user experience.

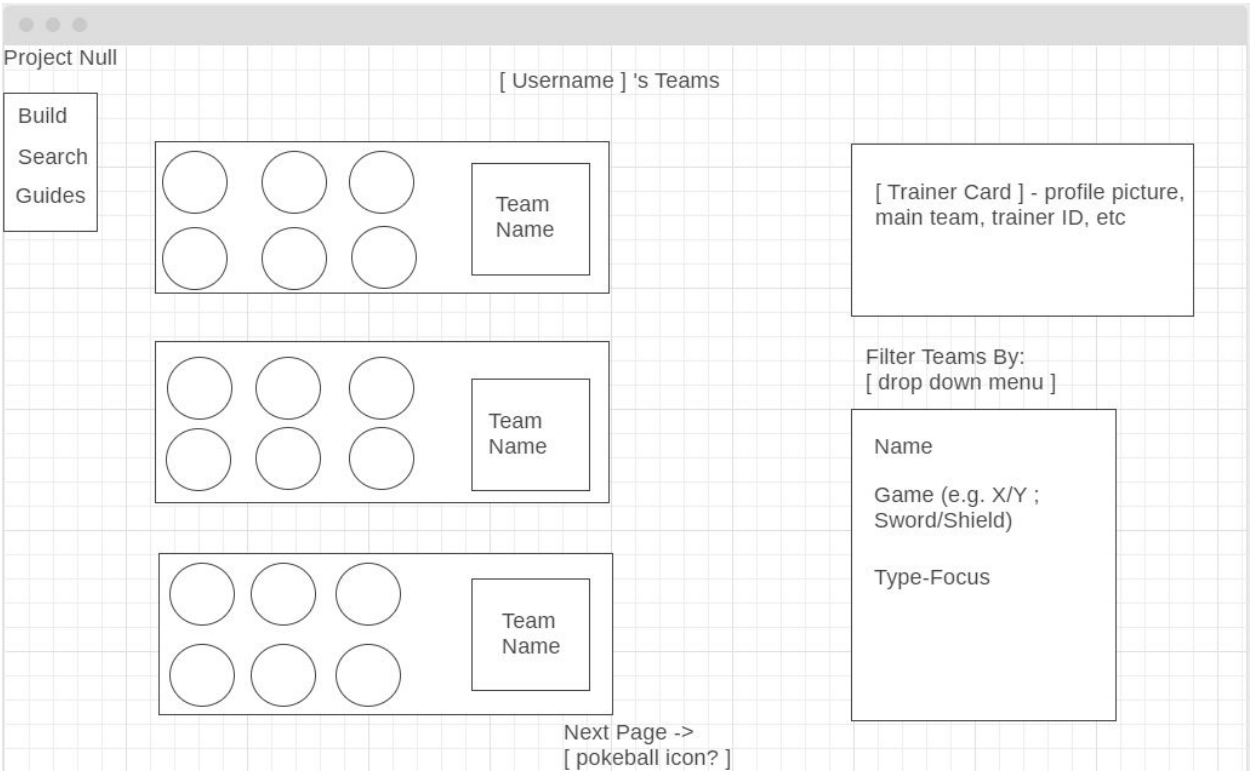
Architecture Diagram



Front End Design



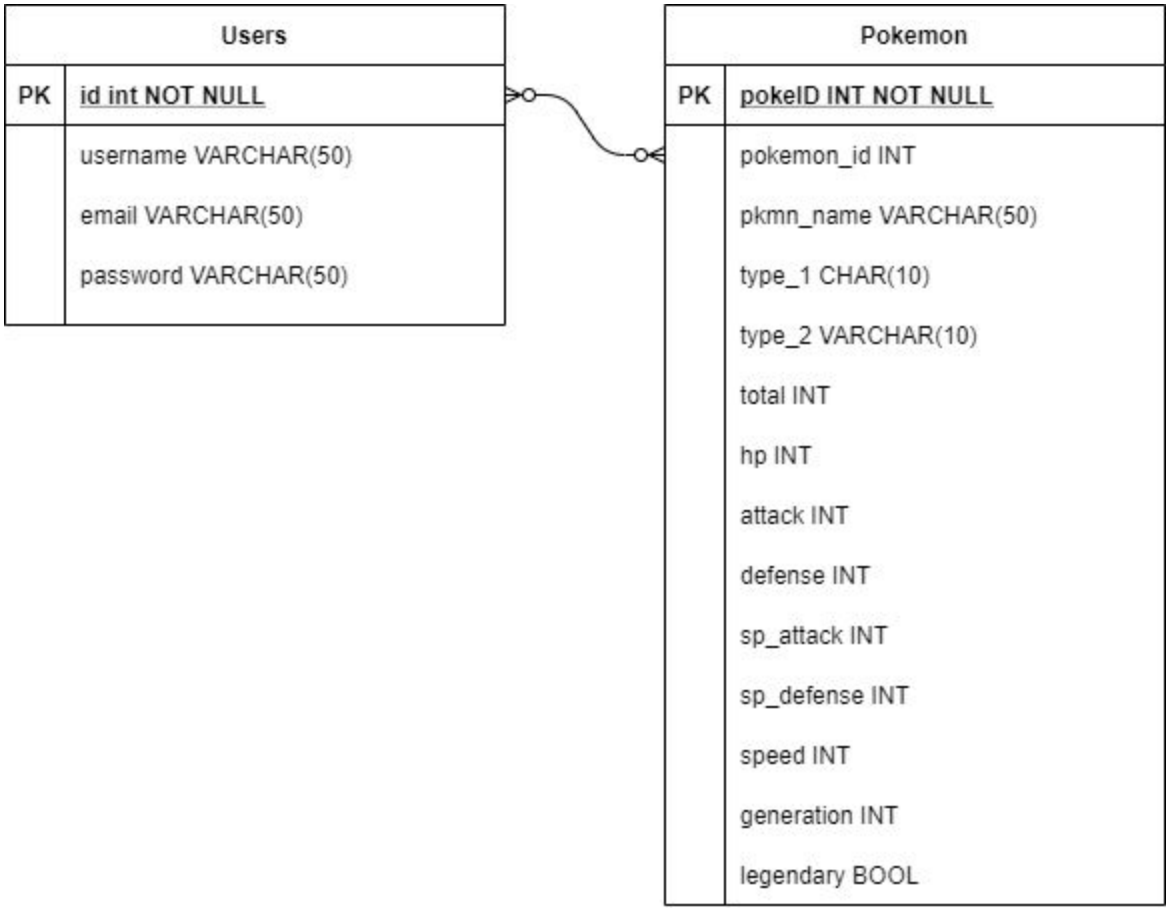

Project Null
User Profile Image
Tips or Hints, Suggestions for Pokemon to Add.
Image of Selected Pokemon
Pokemon 1
Pokemon 2
Pokemon 3
Pokemon 4
Pokemon 5
Pokemon 6
Pokemon ID Number, Adjustable Level, Gender, Ability, Etc, what Item it holds, etc
Search Filters for specific Type, Generation, etc.
Search Bar
Adjustable Numerical Stats Attack, Defense, Etc.
Scroll Menu of Pokemon in Numerical Order or based on Search Results
Moveset with Drop Down Menus for possible moves (up to 4)


Project Null
[Username] 's Teams
Build
Search
Guides
Team Name
Team Name
Team Name
[Trainer Card] - profile picture, main team, trainer ID, etc
Filter Teams By:
[drop down menu]
Name
Game (e.g. X/Y ; Sword/Shield)
Type-Focus
Next Page ->
[pokeball icon?]

Web Service Design

- Using Heroku web hosting to host our Database and Node server.

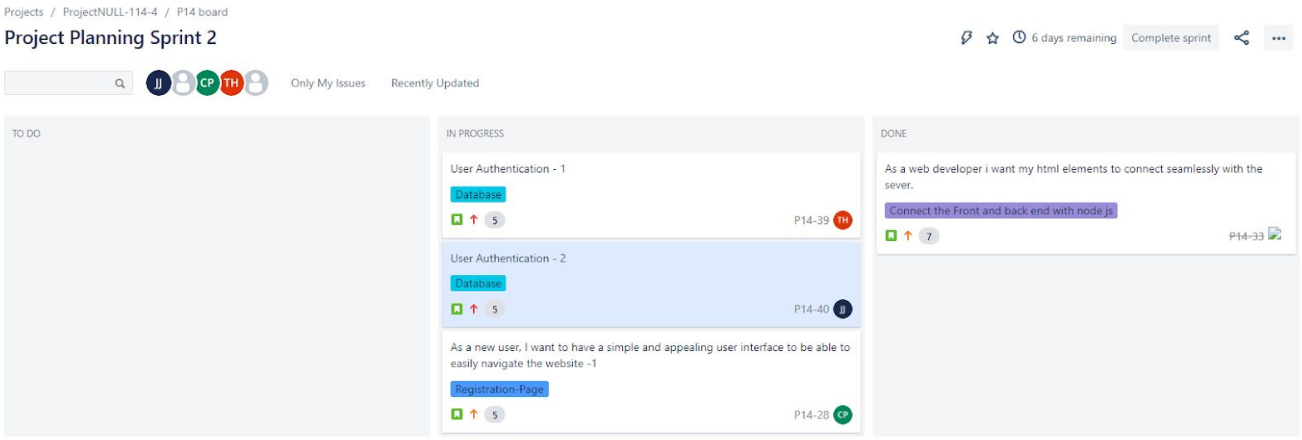
Database Design



Individual Contributions

- **Ben:** I have made a working node server that locally runs all the html/js/css we have so far. Have been mostly researching how to connect with heroku online postgres database. Working on a “list of pokemon page” which will request pokemon from the database through node and have them injected into the live ejs files be rendered to the interface. I also made the server architecture diagram.
- **Christian:** working with trustin, in finishing the UI. currently working on having a working search function, connecting the database and the UI making it possible to search through the database and return the pokemon fitting the descriptions and filters. Will then move to having a team be saved to the users.
- **Jamie:** Working with Taryn, we continued to make sure the database was functional and connected with the rest of the coding that Kaleb created. We worked on building up the user database by adding in an authentication system, which we still need to clean up, but should be able to implement soon working with Ben using Node.js.
- **Kaleb:** Working on making C++ functions into a node js addon for further integration with the website. This would allow us to make calls to the database from our code, perform calculations with C++ and return data to our webpage. Working with Ben in order to integrate this into our node.js server.
- **Taryn:** Together with Jamie, we continued to work on database functionality, architecture, and implementation with the other facets of the code. With the server and database connected, we’re able to access the database externally now (and not just from my local machine). As far as node.js implementation, we will work with Ben to ensure everything connects. The main feature that we’re adding is the user authentication functionality using Node.js, which is currently being revised and implemented soon.
- **Trustin:** Working with christian to finish the HTML. My part was to make a sign up page. I will also be working with the backend team to hook the sign up page up to a SQL database so users can be stored into the database along with the team they created and eventually be connected to other users on the website.

Jira Screenshot:



Github Link: https://github.com/CSCI-3308-CU-Boulder/114-4.git

Challenges

- A significant challenge has been connecting all of our components that we've made up until now. With various front-end and back-end components complete, connecting everything has proved to be difficult. It has definitely required more communication and possibly some revising in the future. Once we have the basics connected, however, it will most likely be much easier to work on the rest of the project.
- Another challenge was trying to figure out the user authentication system by combining many of the lessons we've learned from recent classes. While we understand how to work with Node.js, for example, it was a challenge for us to try to create a user account system with our limited resources and experience.
- Learning heroku, postgres, node js, bootstrap, internal js, has been hard. I think most of us don't have any experience with any of these development tools.
- Managing the time to work on the project and balancing the lab, HW and other classes has been tough and a consistent schedule for meeting times with an ever changing workload.