1. What goal will your website be designed to achieve?

Entertainment/interactive experience for younger ages, revamping an old website that provides a nostalgic experience for older players. Fundamentally the goal is just for fun but if the player can learn something about being responsible for a pet along the way, that is a plus.

1. What kind of users will visit your site? In other words, what is the demographic of your users?

Younger users age range 8-14, perhaps also those that played Neopets/Tamagotchi when they were kids now in age range of 25-35. I foresee most users being at that age where they get their first computer and would lose interest somewhere around high school age.

1. What data do you plan on using? You may have not picked your actual API yet, which is fine, just outline what kind of data you would like it to contain.

There will be user data so that each user has control of their own accounts and cannot access another person’s account. There will also be the virtual pet’s data so hunger levels, entertainment levels, that will change as time passes. Also data to store a user’s inventory/purchases. The big data point is tracking the last time an action was taken on hunger, for example, and tracking the next time a user logs in, compares the two, and adjusts hunger accordingly.

1. In brief, outline your approach to creating your project (knowing that you may not know everything in advance and that these details might change later). Answer questions like the ones below, but feel free to add more information:

a. What does your database schema look like?

1. There should be a user with: username, password, email, standard user data, virtual currency amount
2. Virtual pet data: user id, hunger, satisfaction, pet details, etc, creation date time, last activate date time, pet appearance
3. Inventory: user id, list of items
4. Items: specific info on items, such as effects and price, item category for food, play, non-essential, item appearance

b. What kinds of issues might you run into with your API?

Relying on outside API for pets/items might run into some issues if the API is down and don’t have a proper back-up image for what those things should look like. Might rely on outside API for realtime events/updating virtual pet’s vitals, it needs to be accurate, as any gaming platform no one likes a buggy experience.

c. Is there any sensitive information you need to secure?

Just username/password/email standard info that should be private to a user.

d. What functionality will your app include?

Playing with a pet, feeding a pet, purchasing items, random events to occur that alter for example, the pet’s description, or reduces the player’s money. There will be functionality that is not random to increase the user’s money so that they can purchase more items. User is ‘responsible’ for the vitality of this virtual pet.

e. What will the user flow look like?

First step would be to create an account, choose a pet from a set of options, and set-up this pet that the user would interact with go forward. Perhaps eventually a user would have more than one pet to interact with. User would log-in, view their pet info, the amount of currency they collected, last log-in. perhaps view other user’s pet infos, interact with pet on a single page, shop view on a different page.

f. What features make your site more than CRUD? Do you have any stretch goals?

The main feature of feeding/playing/interacting with the pet still involves CRUD but would be mostly done through JS so as not to redirect/refresh the page every time an action is taken, a departure from how Neopets originally worked. Eventually having some kind of animations would be cool, custom pet avatars, more functionality and different things for users to do, multiple shops, venues, mini-games, things that would contribute to more of a full interactive experience.