Leanne Kate Suen Fa & Sharon Ku
CART 351 Networks & Navigation
November 10, 2021

Greenhouse Hub Prototype

PART A: PROJECT UPDATE

Summarizing Project Proposal

In times of isolation, we turn towards social media to remain close to our friends and family. However, the experience of communicating digitally is impeded by social media's ubiquitous notifications and its capitalist view of users as consumers.

To combat this, we designed Greenhouse Hub, a collaborative space where users nurture connections between their loved ones. The user cultivates a plant in other people's gardens and in doing so, deepens their relationship with them—these special people are called *podmates*. Through exchanges of caring notes, photos, videos, memes, gifs, and other creative media, the podmates engage in growing the plants together. The world is designed as a 2D space that houses a myriad of greenhouse-looking pods. Each user chooses a pod to live in. The users can visit any other player's pod and interact with their plants.

Plants evolve through various growth stages, allowing users to visibly witness their development, which mirrors their growing connection with their podmates. Our goal in designing this world is to empower users by providing them with a place of solace free of advertisements and free of manipulation. It is a happy, magical, and loving space in the digital realm.

In preparation for the development phase, we found three similar projects that inspired us or informed us of our design decisions.

1) Lovebox for Lovers, a messaging device shaped as a box, showed us one way to facilitate sending of pictures and messages among friends who are physically separated; however, we aim to do so accessibly, without the barrier of cost.

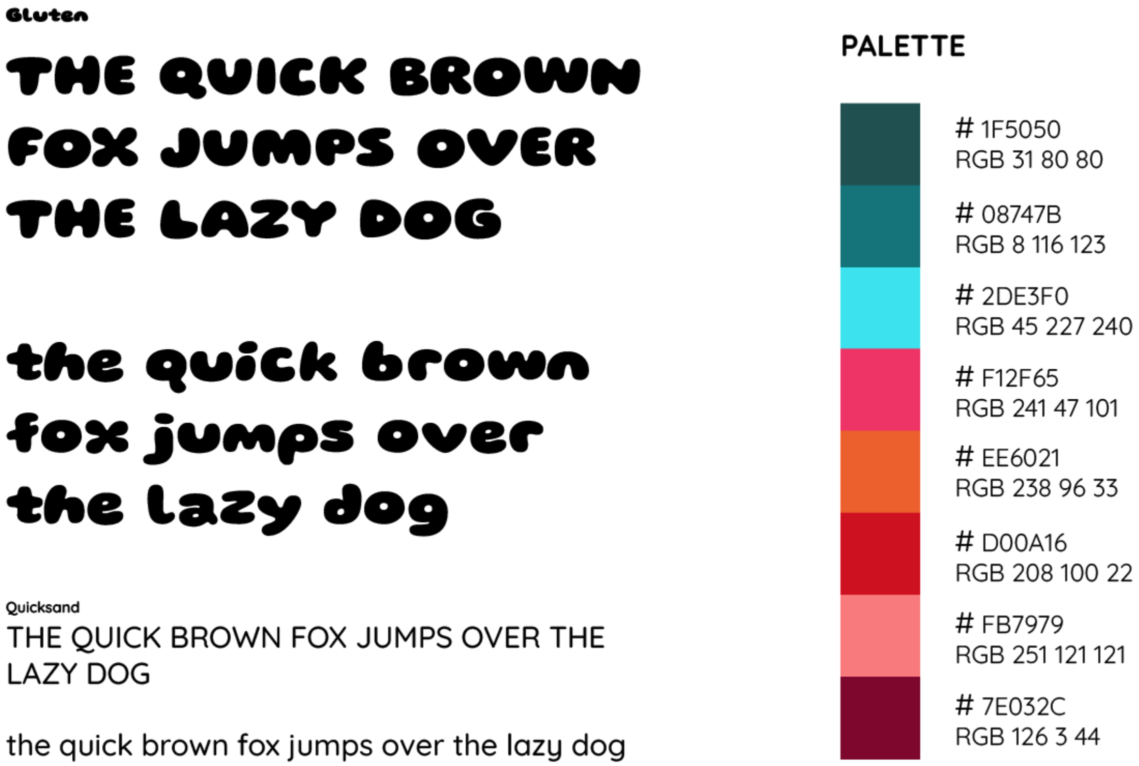
2) Habitica, a habit-building application that uses a social network for boosting motivation, presented a structured user interface. We want to differentiate from the traditional app interfaces by enhancing the navigation with visuals and sounds.

3) And finally, New Nature by Marpi, a digital petting zoo at Artechouse's art exhibition, exemplified the beautiful and immersive experiences that can emerge when humans interact with plants. We wish to imbed our plant designs with an equal amount of attentiveness to fabricate a similar atmosphere.

Update on Progress

WHAT HAS BEEN COMPLETED

Before even starting coding the prototype, we have mapped the steps of the user experience and produced mockups for nearly every scene of the project accordingly. We have also decided on the branding and produced a style guide consisting of a colour palette, typefaces for both headers and body text (Fig. 1). We have also produced most of the necessary icons in order to ensure consistent visual aesthetics (Fig. 2). Based on the same style guide, we have also completed the production of the necessary images to be used in the project, such as the pods, seeds, and plants (Fig. 3-4). All the visual components match the gentle, happy, and playful atmosphere with an aura of mystery we wanted to inspire.



Gluten

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

the quick brown fox jumps over the lazy dog

Quicksand

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

the quick brown fox jumps over the lazy dog

PALETTE

# 1F5050	RGB 31 80 80
# 08747B	RGB 8 116 123
# 2DE3F0	RGB 45 227 240
# F12F65	RGB 241 47 101
# EE6021	RGB 238 96 33
# D00A16	RGB 208 100 22
# FB7979	RGB 251 121 121
# 7E032C	RGB 126 3 44

Fig. 1: Style guide with choice of typefaces and colour palette

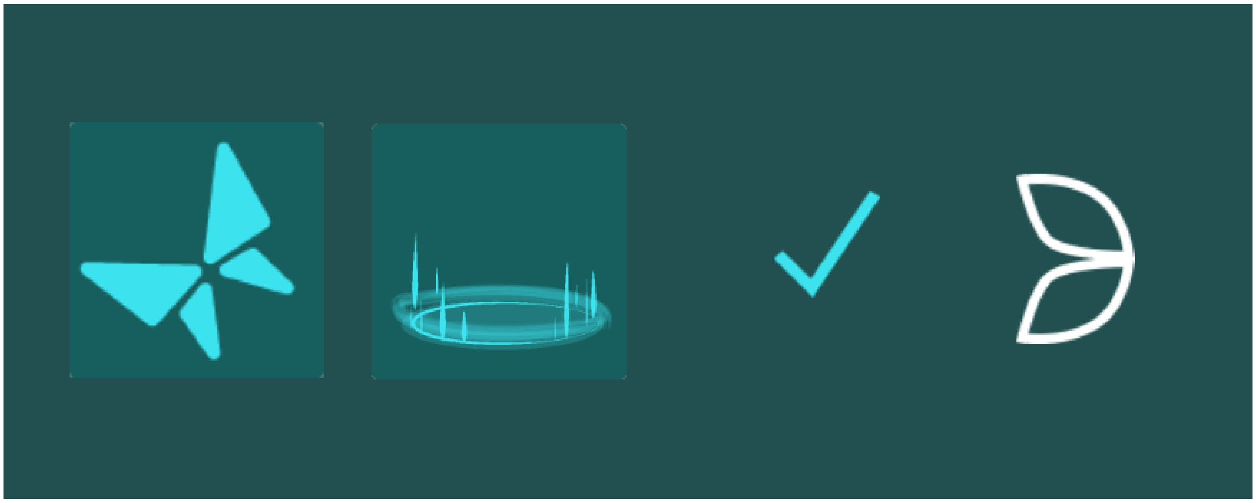


Fig. 2: Icons (from left to right) : send message, teleport, validate, submit

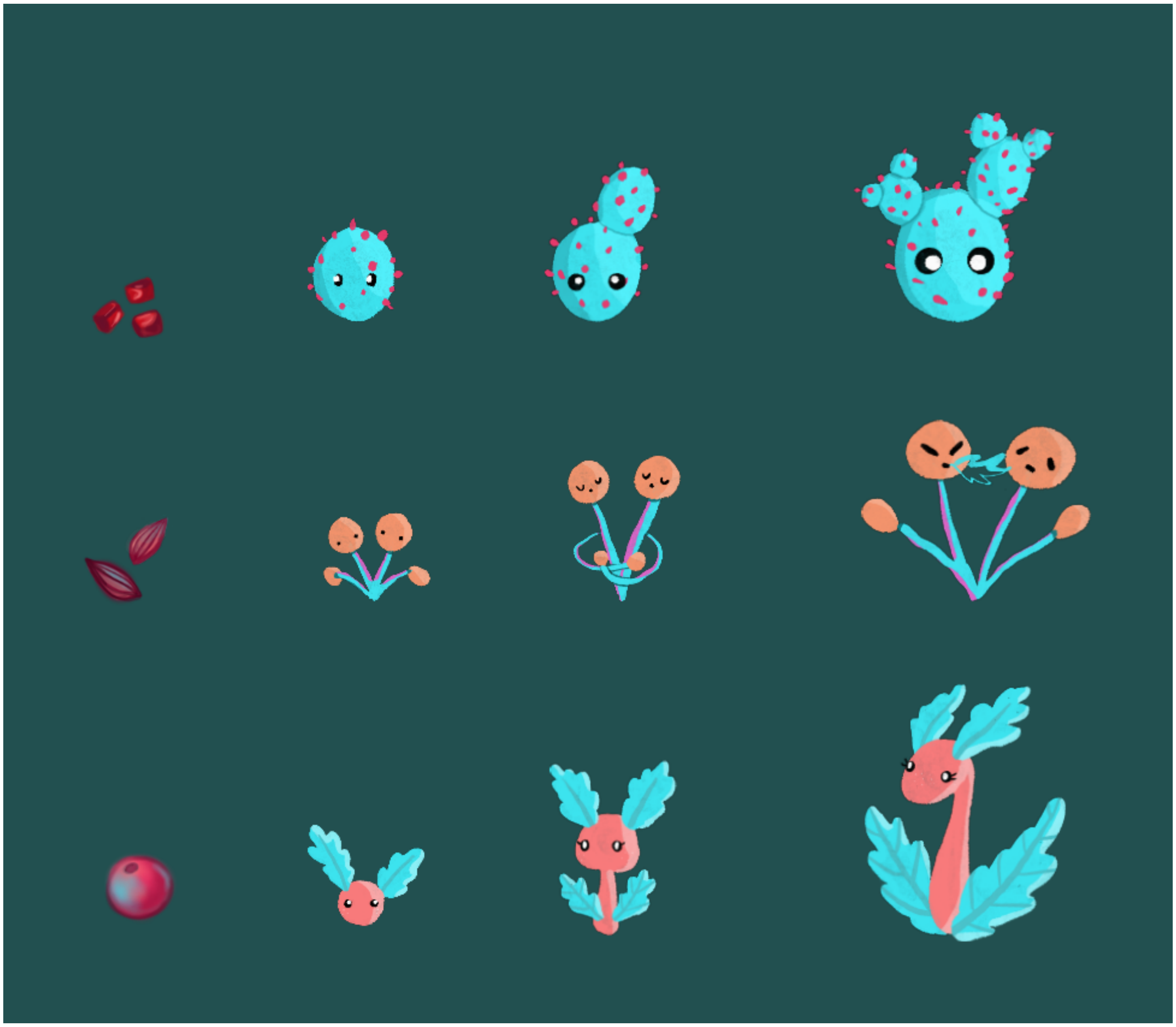


Fig. 3: Seeds and plant maturation stages

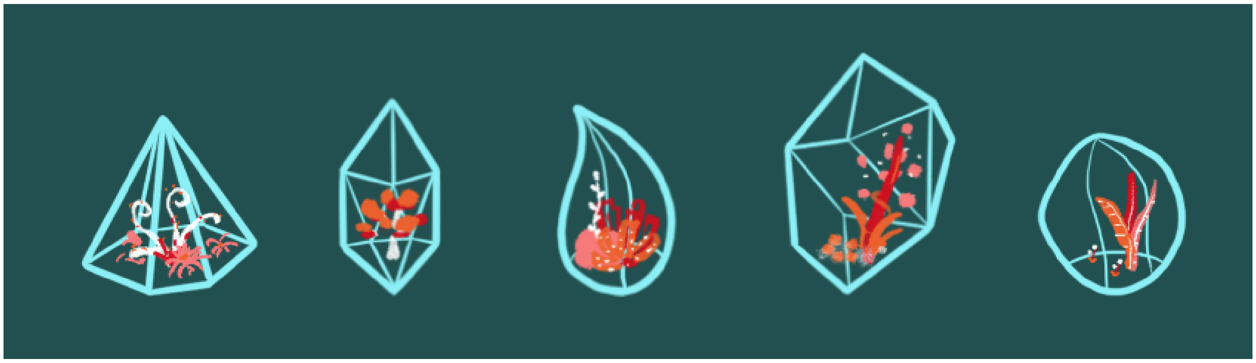


Fig. 4: Pod choices

We have started working on the prototype and so far the user is able to click their way through the whole game to navigate from one scene to the other and can choose 2 paths from the welcome screen either as a first time user or as a returning user. Following user testing, we also made the possible actions like clicking on an icon or on a pod more noticeable by adding some text on overlap or changing the size on overlap. We've mainly been focusing on client-side code by coding in the HTML, CSS, JS, and p5.js files. Currently, all the public-facing code is stored in a "public" folder.

WHAT IS LEFT TO BE DONE

- On the server side, we've set up the files required for working with node.js and installed the Express module, though we haven't installed socket.IO or started implementing socket connections yet. We watched The Coding Train tutorials on how to code these socket connections on the client and server side.
- We still need to work on the back-end interactions requiring the storage of data, i.e. the username/password storage, the messaging function, the display of plant settings after clicking on a plant, the storing of pod coordinates, and the planting of a seed.
- We also need to display plant properties when clicking on a plant inside the plant view screen.
- We still need to produce the background images for the welcome screen and the registration form.
- We want to incorporate music and sounds into our program.
- We need to program the growth of the plant.
- We need to create a database using SQLite to store all user data.

Two Diagrams of Overall System: Showing Navigation Links

DIAGRAM (1/2)

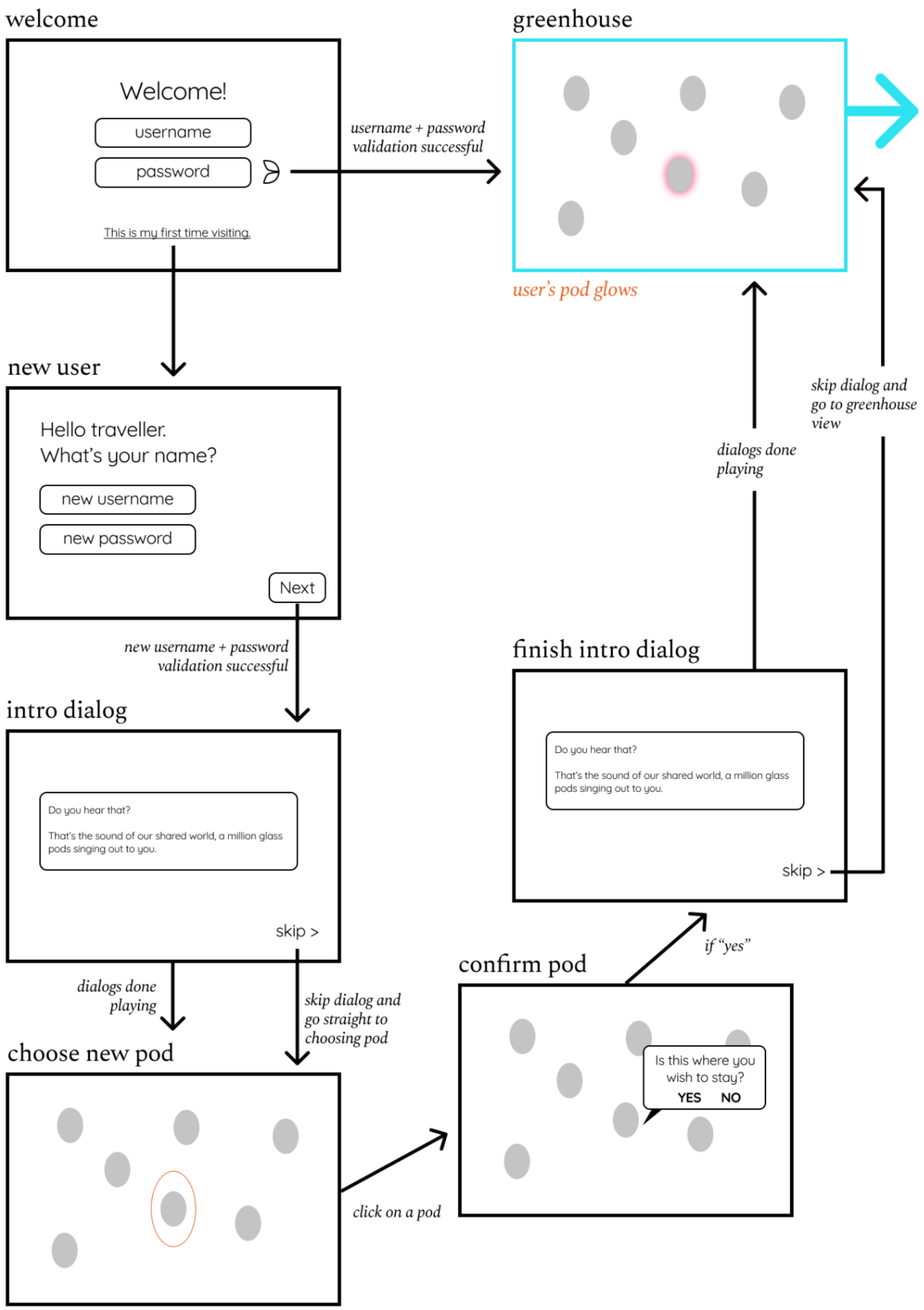
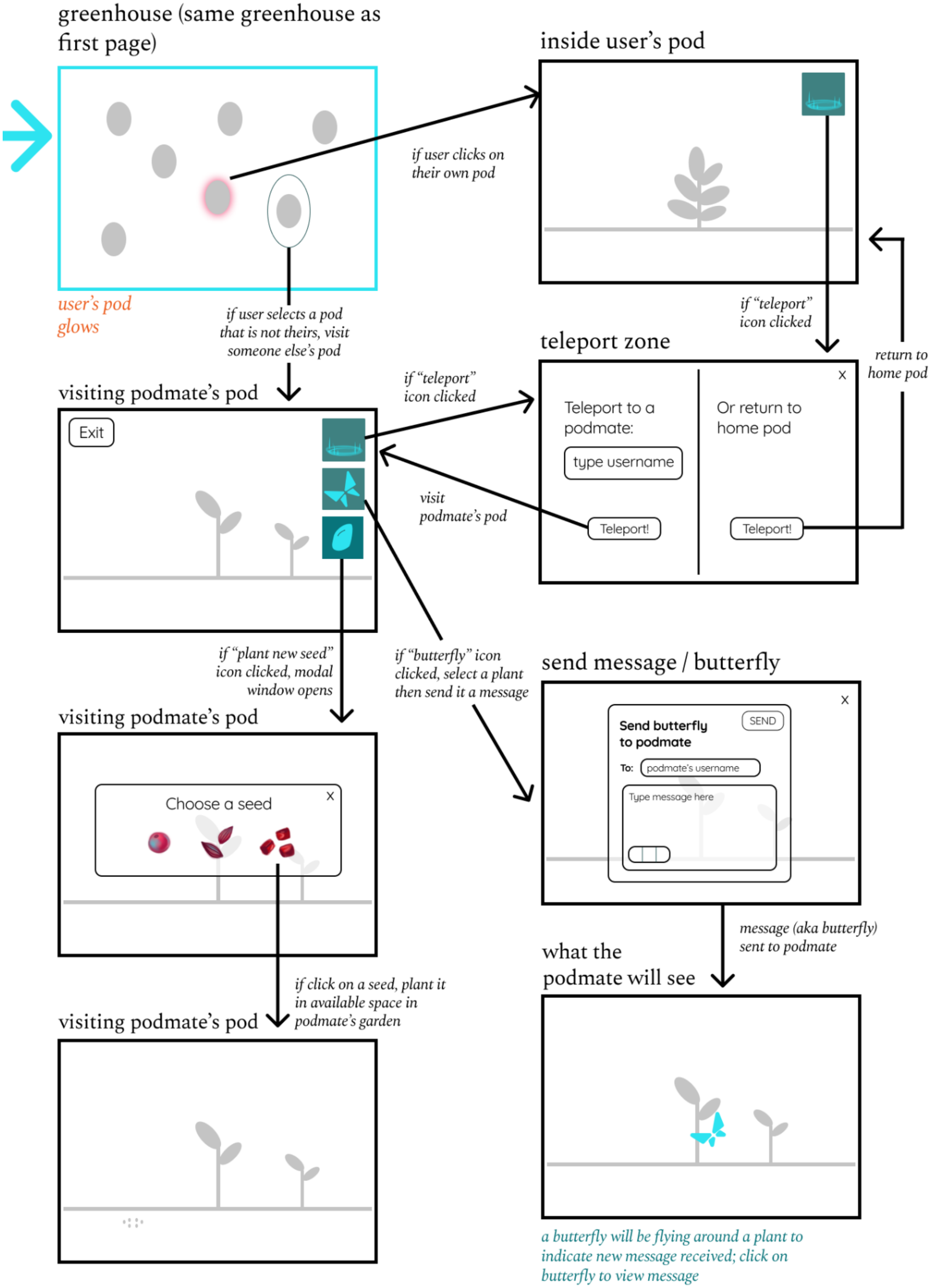

welcome
Welcome!
username
password
This is my first time visiting.
username + password validation successful
greenhouse
user's pod glows
new user
Hello traveller.
What's your name?
new username
new password
Next
new username + password validation successful
intro dialog
Do you hear that?
That's the sound of our shared world, a million glass pods singing out to you.
skip >
dialogs done playing
skip dialog and go straight to choosing pod
choose new pod
click on a pod
confirm pod
Is this where you wish to stay?
YES NO
if "yes"
finish intro dialog
Do you hear that?
That's the sound of our shared world, a million glass pods singing out to you.
skip >
dialogs done playing
skip dialog and go to greenhouse view

DIAGRAM (2/2)



greenhouse (same greenhouse as first page)

user's pod glows

if user clicks on their own pod

inside user's pod

if user selects a pod that is not theirs, visit someone else's pod

if "teleport" icon clicked

teleport zone

return to home pod

visiting podmate's pod

if "teleport" icon clicked

visit podmate's pod

if "plant new seed" icon clicked, modal window opens

if "butterfly" icon clicked, select a plant then send it a message

send message / butterfly

visiting podmate's pod

message (aka butterfly) sent to podmate

if click on a seed, plant it in available space in podmate's garden

what the podmate will see

visiting podmate's pod

a butterfly will be flying around a plant to indicate new message received; click on butterfly to view message

Description of each Component

- Login/Register
    - The welcome screen will offer 2 options to the user, either to login as a returning user or to register a new account:
        - If the user is a returning user, inputting their username and password will allow them to access the greenhouse page.
        - If the user is new, clicking on “This is my first time visiting” link will lead them to the registration form page which will ask them to create a username and password:
            - This data will be validated so that there are no 2 users with the same username and both fields will need to be filled in order to follow through.
            - After the validation, an introductory dialog sequence will welcome the user to the Greenhouse Hub. The user can skip this dialog and proceed to the next step.

- Select pod (for new user)
    - After the first part of the introductory dialog, the user will be led to the Greenhouse where a modal window will let them know that they need to choose an unoccupied pod among the ones on the page.
        - This modal window will only appear if the user is new.
    - Once the pod is chosen, the introductory dialog finishes. The user can skip this dialog.
    - After the dialog, the user is brought to the Greenhouse screen.

- Connecting with other podmates
    - Connecting with other players is an essential part of the game experience and is thus a necessary step in order to go forward. There are several ways the user can do this:
        - Searching with known username
            - A search bar will allow the user to type in the username of someone they are already acquainted with (see Visit friend > Teleport).
        - Entering a random pod
            - Clicking on a random pod while navigating the Greenhouse screen will allow them to enter it (see Visit friend > Navigate through map).
        - Display looking for friend banner (optional)
            - Clicking on “Looking for friend” icon will raise a flag above their home pod indicating that they are open to finding new friends.

- Visit friend
    - Teleport to friend
        - While inside their home pod, they can click on the “Teleport” icon then input the other player’s username to go directly to their greenhouse garden.

  - Navigate through map
    - Once they have clicked on a pod to visit it, they will see the inner greenhouse garden screen.

- Select seed (when visiting friend's pods)
  - When inside another player's garden, you can click on the "Plant new seed" icon to open the "Choose seed" modal window. There will be 3 seeds to choose from. Clicking on one of them to select the seed will make the modal window disappear and the player will be able to place it into the garden. The final plant is a mystery.

- Send/Receive butterfly messages (when visiting friend's pods)
  - Plant growth will depend on the number of times the seedling receives butterflies from other users. Visitors can choose to fertilise any plant they want by clicking on the "Butterfly" icon. Sending butterflies is the equivalent of messaging the person and clicking on it will lead to a form where they can input their message in the form of text, images or links.
    - Plant growth
      - There will be 3 stages of plant growth with 4 score thresholds for each life stage from seed to seedling to young plant to blooming plant.
    - How to know when a message was received:
      - If the user receives a message from a podmate, the plant that has been fertilised by the message will have a butterfly circling it.
      - If the user clicks on the butterfly, the message will pop up.

User Testing

Mariam's feedback

We tested our program with our friend Mariam. She mentioned that the choice of pod in the greenhouse screen was not clear, so we addressed this by making the greenhouse grow on hover. She also mentioned that she wasn't sure what the three buttons in the plant view were for. This makes sense since we haven't added their functions yet, though to improve the user interface, we added a glow to the back of the buttons and placed captions beside them on hover. She found the overall aesthetic cute, which we were glad to hear as it reflects the atmosphere we were going for. However, since we did not provide any background to what our program is about, Mariam told us that based on the interactions we've coded so far, she wasn't sure what the goal of Greenhouse Hub is. This helpful feedback made us realize the importance of the first-time user screen which we will have to complete in order to provide the users with our website's background story.

Amy's feedback

After playtesting with Mariam and implementing the changes, we playtested with another friend, Amy. The two main aspects that needed to be fixed are (1) the canvas resizing after the player goes from the inside of the pod screen to the greenhouse screen, and (2) the issue with superimposing greenhouses. We fixed the canvas resizing problem right away. For the superimposing positions, we will address that problem later on.

Just like Mariam, Amy found our program cute. She was happy to see the plant character. Amy had the intuition to click-and-drag on our greenhouse map, much like one would do on Google Maps. We haven't added that possible user action and might consider doing it in a future prototype.

Reflection: Accomplishments and Further Improvements

We've currently created a foundation for our code and laid out the different files in which we will store our code. For the new-user process, we coded up to the password input stage. For the returning user process, we coded the login page, the greenhouse screen (where users can scroll and view all the pods), and the plant view (where users can see and hover over the three buttons: "Butterfly", "Teleport", and "Return to pods", though only the "Return to pods" icon functions so far).

We are very proud of the aesthetic choices we have made so far. We are accomplished by the amount of code we were able to write with p5.js, given that we only learned how to program with this library last year. We found that using one html page to handle various screens was causing issues with the input forms inside our p5 canvas. So we were happy that we solved this issue quickly by using different html pages.

The new-user process of completing the form is still in development. The user interface can still be improved, in particular the hierarchy of text and the icon positioning. We also wish to optimize the responsivity of our program to different screen sizes. Right now, it's mainly programmed for computer screens. When we click on the plant, we wanted the plant information to be displayed but it's not appearing. Clicking lacks precision when image mode is set to CENTER but when on CORNER image increases in size from the corner.

We are excited to see the final product of our program and explore different technologies to get it to work. The experience of being able to visit other pods and send messages to each other will be extremely gratifying so we look forward to that.

PART B: INTERACTIVE PROTOTYPE

Mid Fidelity User Interface

Visit Greenhouse Hub

(still in development!)

https://hybrid.concordia.ca/k_sharon/cart351/prototype/public/

Visit https://hybrid.concordia.ca/k_sharon/cart351/prototype/ to see non-public files

Selection of APIs and 3rd Party Libraries

- p5.js: for drawing on canvas
- p5.sound: for adding sounds to program
- jQuery
- Node.js & Socket.IO: for communicating between web clients and servers.
- Sqlite: for storing information inside a database

Data Map of Data Storage and Data Sharing Requirements

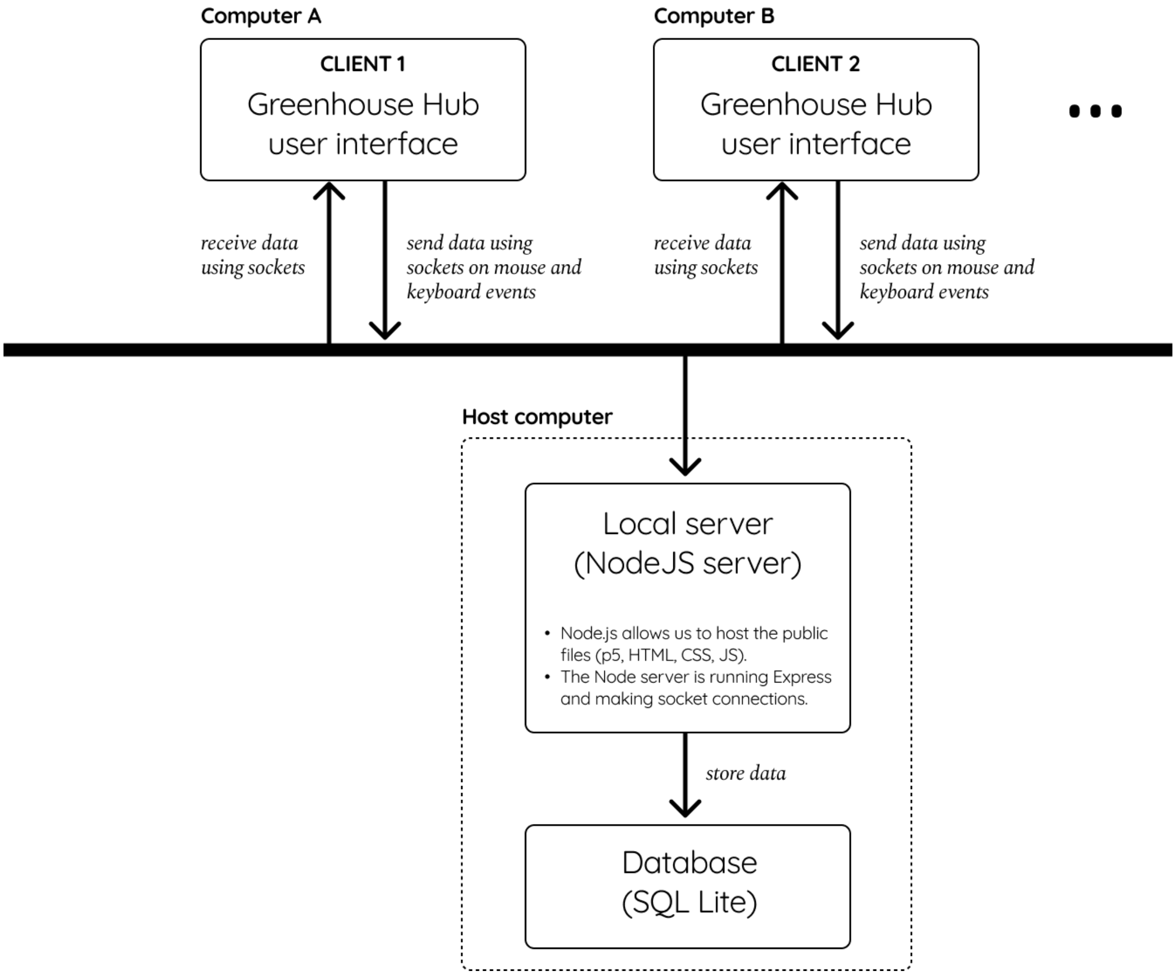

Computer A
CLIENT 1
Greenhouse Hub user interface
receive data using sockets
send data using sockets on mouse and keyboard events
Computer B
CLIENT 2
Greenhouse Hub user interface
receive data using sockets
send data using sockets on mouse and keyboard events
•••
Host computer
Local server (NodeJS server)
• Node.js allows us to host the public files (p5, HTML, CSS, JS).
• The Node server is running Express and making socket connections.
store data
Database (SQL Lite)

Fig. 5: Data map showing how data will be stored.

Socket ID is needed to identify the users; the ID is generated the first time the user connects to the website. Username, password, and pod ID are stored once the user creates an account for the first time. The three seeds are also randomly generated and stored at the creation of an account.

When the user sends a butterfly to a podmate, the database will store all the items listed under “messagesReceived.” The message’s “readState” is set to false by default, and only updates to true when the receiver opens the message by clicking on a butterfly.

Each user starts with three seeds. Every time the user visits a podmate and clicks on “Plant new seed” icon then selects a seed, the seed’s corresponding “plantName” has the “planted” value set to true (this value is false by default).

When a podmate visits the user’s pod and plants a seed, this plant is added to the database with all the entries listed under “plants.” A newly-planted seed has a growth stage of 1.

Database:

Each user has the following information stored:

- socketId (integer)
- username (string)
- password (need to find a way to encrypt) (string)
- podId (integer)
- messagesReceived
    - senderUsername (string)
    - readState (boolean - “true” if message has been read, else “false”)
    - associatedPlantNumber: indicates which plant received the message; the plant number corresponds to the order in which plants were grown (integer)
    - messageContent
        - strings
        - images
        - videos
- seeds: the 3 seeds that the user can plant in podmate’s pod
    - plantName: used for grabbing correct image of seed (string)
        - planted (boolean - “true” if seed has been planted, else “false”)
- plants
    - orderGrown: order in which plant was grown in garden (integer)
    - name: used for grabbing correct image of plant (string)
    - position: position in garden
        - x (number)
        - y (number)
    - growth stage (integer: 1, 2, 3, or 4)
    - numMessageNeededToGrow: number of messages plant must receive until next growth stage (integer)

In the “public” folder, we will have a JSON file that stores pods information (stays the same every time):

- pod id (integer, starting count from 0)
- positions
    - x (number)
    - y (number)
- image (string)
- size/scale (number)