G5: Midterm Presentation

WIP [Wanda Kuang, Ivy Zhou, & Patrick Ho]

Problem Space

A common issue faced by our target audience is that **studying can be an isolating and unenjoyable experience.**

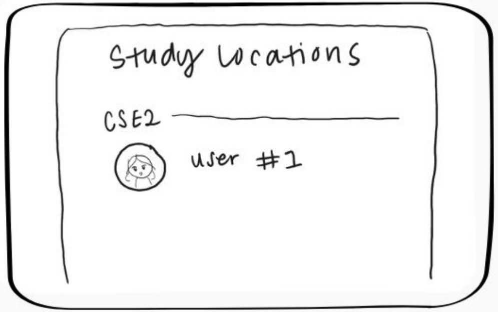
We aim to improve the process of finding study partners and study locations to foster collaboration between students and make studying a more enjoyable activity.

Target Audience: Students in introductory level courses at the University of Washington

Storyboard #1: Finding Others to Study With

???
??
Practice Exam

User #1 is studying for the midterm exam for their introductory CSE course, but they are struggling on a practice exam question.



User #1 visits the Discord server for the class, and tags their study location as CSE2.

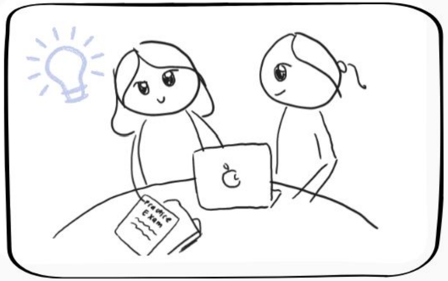
Storyboard #1: Finding Others to Study With

???
???
lecture slides
binary trees

User #2 is also studying for the midterm. They are struggling to understand a concept from lecture.

study locations
CSE2
user #1

User #2 visits the Discord server and sees that User #1 is working at CSE2.

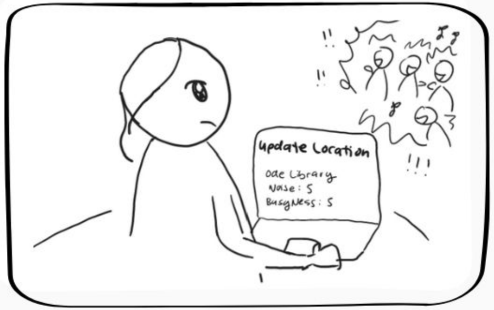


User #2 joins User #1 and they help each other.

User #1 and User #2 both feel ready for the exam now.

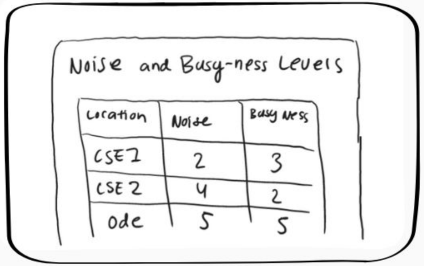
Storyboard #2: Deciding *Where* to Study

User #1 is studying at Ode Library. The library becomes very crowded and User #1 cannot concentrate due to the high noise level.



User #1 uses the Discord bot to update the noise and busy-ness levels for Ode Library.

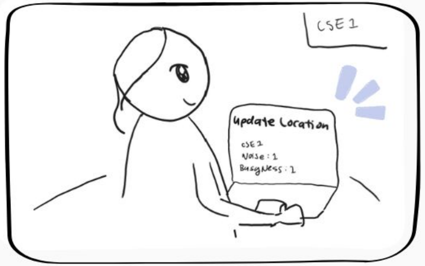
Storyboard #2: Deciding *Where* to Study



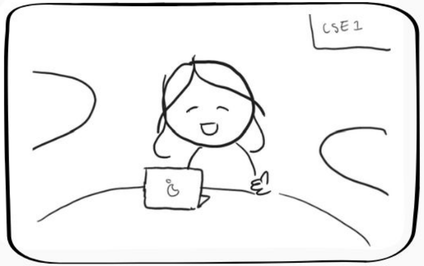
Noise and Busy-ness Levels

Location	Noise	Busy Ness
CSE1	2	3
CSE 2	4	2
Ode	5	5

In the Discord server, User #1 looks for an alternative study space with lower noise and busy-ness levels. They decide to move to CSE1.



After User #1 relocates to CSE1, they notice that the location is less noisy and busy than the current reported levels, so they update the scores.



User #1 is able to concentrate and complete their assignment.

Other students on the Discord server also see the updates made by User #1 and join the quiet study location to complete their work.

Planned Features

Primary Features

- Users can self-assign any study location roles that apply to them
- Users can append new study locations to the current list of study locations
- Users can communicate with other server members that are also in the same study location as them
- Users can update the status of their study locations (noise and busy-ness levels)
- Users can request to view a list of all possible bot commands

Secondary Features

- Bot will notify users who are tagged at a location when one of their classmates tags themselves at the same location
    - Users can tell the bot when to stop notifying them once they have a large enough study group
- Users can see the historical noise and busy-ness levels of a particular location (as reported by other users)
- Users can set a duration for how long they hold a location role for

Current State

Tech Stack: `discord.py`

- API wrapper library for Discord that leverages Python’s `await` / `async` syntax.
- Code spec completed (next 2 slides)

Implementation

- Working prototype of commands that do not manage roles or channels

Code Spec

init()

- Sets up the bot to begin functioning in a server by creating a location role selection message and a preset study location for CSE2 Lab 1.

help(cmd=”)

- Prints a help message describing the bot’s commands and usage in the channel that /help is typed by a user.
- Optional argument cmd allows users to specify a command to get more information (arguments, example usage, etc.) about. A cmd value of ” will result in a general help message being printed.

add_location(location)

- Adds a new location for location, creating an associated role, updating the location role selection message, and private text channel.
- tag_location(location, duration=1)
- Assigns the role for location to the user. After duration hours, remove the role from the user.
  - “Successfully assigned location role! This role will be removed in duration hours.”
- Can overwrite duration by calling the command again.
- Tagging yourself at a location means that you’re opting in and willing to work with others.
- Throws an error if location does not exist.
- Only accepts specific building “prefixes”: CSE1, CSE2, Ode
- Optional argument duration allows the user to specify how long they want to hold the location role for.

Code Spec (cont.)

list_people(location)

- Prints a message with a list of users currently holding the role of location. Does **not** mention the users.

list_info(location=”)

- Prints a message with the noise and busy-ness levels of all locations tracked by the bot.
- Optional argument location allows the user to get the noise and busy-ness levels of location for the past week.

update_noise(location, noise_level)

- Updates the noise level for a location.

update_busy(location, busy_level)

- Updates how busy a location is.

check_user(user)

- Sends a message to user to check whether they are still at a location after three hours of holding a location role and no text channel activity.

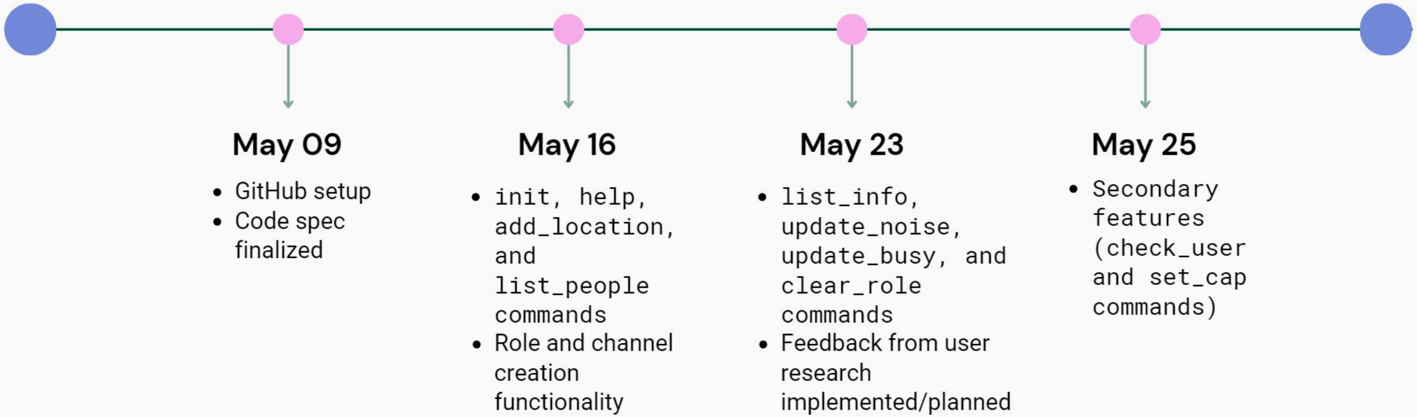
clear_role(user, location)

- Removes the location role for user.

set_cap()

- Allows the user to tell the bot to stop pinging people who are already in a location when new people join.

Projected Timeline



May 09

- GitHub setup
- Code spec finalized

May 16

- init, help, add_location, and list_people commands
- Role and channel creation functionality

May 23

- list_info, update_noise, update_busy, and clear_role commands
- Feedback from user research implemented/planned

May 25

- Secondary features (check_user and set_cap commands)

Feedback Questions

- What might incentivise you to use the study bot more often?
- What additional features would you like to see to help students find study locations near/around campus or to promote collaboration with peers?

Thank You!