

Office Hours Waiting List

Requirements and Specification Document

2023-02-26, version 1.0

Project Abstract

Office hours can get crowded and overwhelming for certain classes. When students arrive at office hours to get help, they sometimes have to wait aimlessly and in a room packed with students for an indefinite amount of time before they are able to receive the help they came for. This is stuffy and stress-inducing for students and teachers alike.

Office Hours Waiting List is designed to combat these difficulties by ensuring the most systematic and efficient waiting process as possible.

The goal of our web app is to allow students to be able to “check-in” to a waiting list and have the flexibility to leave and come back when they are notified of their turn. They will also be able to see their position in the queue in real-time, or leave the queue if necessary.

Document Revision History

Rev. 1.0 2023-02-27: initial version (project abstract, customer, user requirements)

Customer

The customer for this app is for professors or teachers hosting office hours, and students attending those office hours. The teacher can start a waiting list for students attending the office hours. The students then enter this virtual waiting list when they arrive there. This virtual waiting list will remove the clutter of students congregating in the room waiting for their turn to get help. The customer has the ability to join the queue, look up their place in the queue, and leave the queue. They could also enter what they came to get help for, so teachers can see that in advance.

extension idea: perhaps a student can see what other students came in for help for, and if they came for the same thing, they could add their name with someone else and get help together so that it is more efficient for everyone

User Requirements

1. User opens the application for the first time
 - a. they are prompted to “start” or “join” an office hours waiting list
 - i. start a waiting list
 - ii. join a waiting list
2. Home Page: The waiting list
 - a. the host view should be able to see:
 - i. see the code to join the waiting list
 - ii. delete their waiting list
 - iii. students names in FIFO order
 - iv. description of help needed
 - v. ability to “call” on someone and remove their name from the list
 - b. the student view should be able to see:
 - i. students name in FIFO order
 - ii. ability to leave the queue
 - iii. add or edit description for reason they need help

Use Cases

This specific project involves numerous use cases for two different users. The first user will be the student and the second will be a TA or an instructor. We will first cover the student’s use cases. Firstly, a student will have the functionality to add or remove their name from the waiting list. They should also have the ability to add a description of the

reason they need help, which can be read by the TA or instructor. Now, we will cover the TA or instructor's use cases. A TA or instructor should have the ability to start their own waiting list, which any student can join. This waiting list will require a code to join. The TA or instructor will also have the ability to delete their waiting list. The instructor or TA will have the ability to see students' names in the order they were queued as well as the description of the reason the student needs help. They will also have the ability to call someone off the list and remove their name from the list.

User Interface Requirements

The user interface is going to be made up of three separate entities. There is going to be a landing page that prompts the user with two separate choices. These options include starting a waiting list or joining a waiting list. After creating a waiting list, the host view will include some depiction of the code required to join the list. It will also show the queue in FIFO order which includes the name of the student and the description of the reason they need help. They will then have a button they can use to call/remove a student from the list. The student's view will also include the list in FIFO order, but this time only includes the names. They will also have a button to leave the queue, and the ability to add/edit their description of the reason they need help.

Security Requirements

There are a few security requirements in this application.

- 1) Ensure the student cannot remove anyone but themselves from the queue
- 2) Only the instructor has ability to remove anyone from the queue other than in case 1

System Requirements

- The system will depend on a computer that has our server
- We will need to target Unix and window because students and staff may use both
- Required memory would need to be 8 gigabytes as that is pretty standard
- A system that can contain a database will be required

- The speed requirement would need to be very fast considering people need to know when the TA are available as soon as it happens

Specification

Frontend design: <https://www.figma.com/file/kShFcOf3OsKrwunc7MnY5E/Office-hours-waiting-list-frontend-design?node-id=0%3A1&t=uHjuieZ7fMZyjcon-0>

Database design

Table 1 - **room_creators**

column name	data type
room_code (unique to table)	string
creator_name	string
time_created	datetime
time_destroyed	datetime

Table 2 - **room_visitors**

column name	data type
visitor_id (unique to table)	int
room_code	string
visitor_name	string
time_entered	datetime
time_left	datetime
is_waiting	boolean

notes:

with this design, we do not

- remove people from the table once they leave the waiting room

