

Queue

Assignment 3
CSSE1001/7030
Semester 2, 2018

Version 1.0.1
25 marks

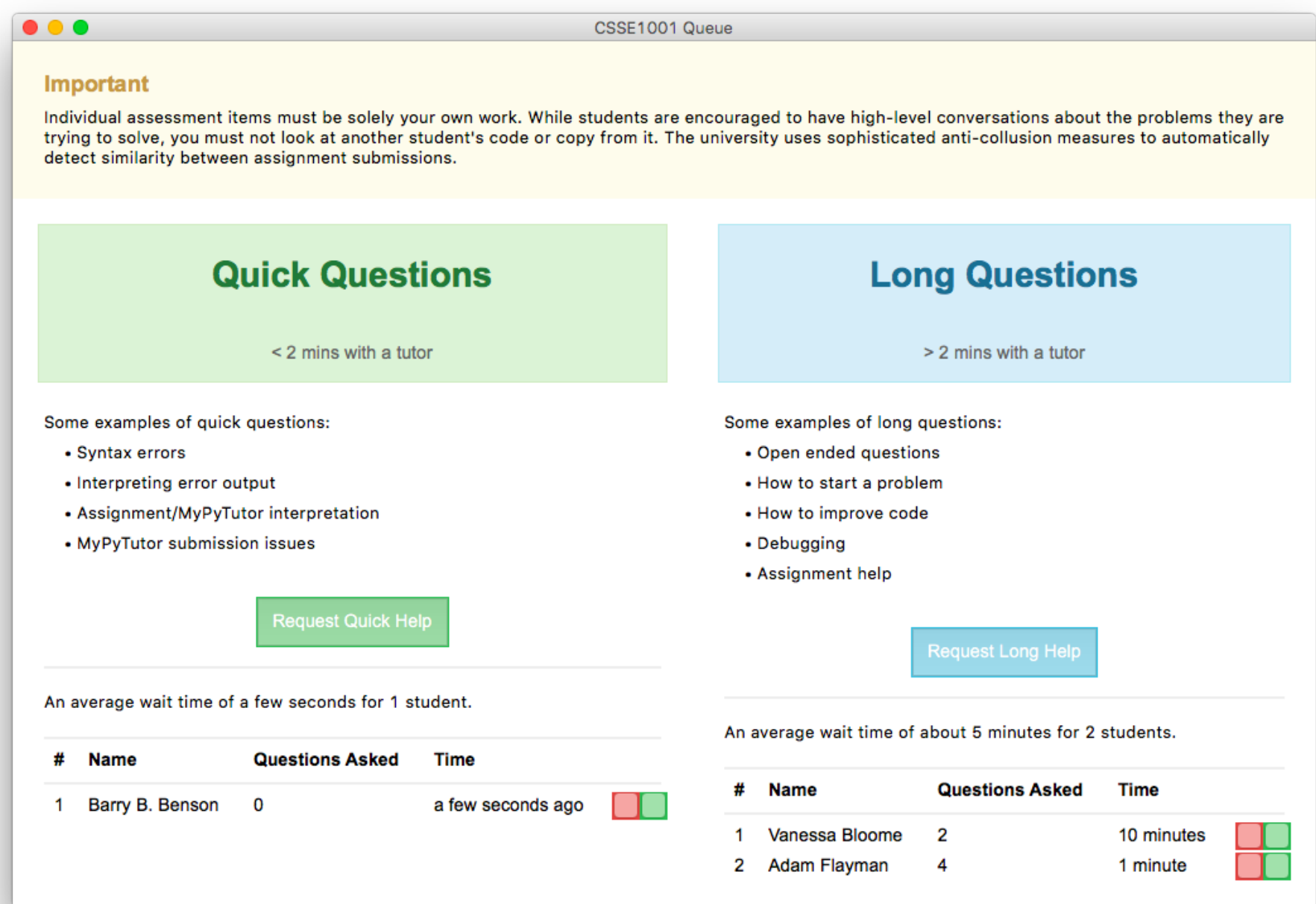
Due Monday 22nd October, 2018, 20:30

1. Introduction

This assignment will give you an opportunity to create a Graphical User Interface (GUI) and to apply the techniques for designing good software that you have been learning in class.

Part 1 of the assignment requires you to create a queueing application to mimic the one that is used currently for CSSE1001/7030 pracs.

An example of what this queue should look like is below



Basic GUI Example

Unlike the existing web queue, however, the application you write is to be a desktop application, rather than a web application. It must be written entirely in python using tkinter. You are not required to make the application work with multiple users over a network.

The GUI in your software is to have the same labels, buttons and text as the existing web application, but the "Request Help" buttons will have slightly different functions. When a user presses either the "Request Quick Help" button or "Request Long Help" button, the application must display a dialog box which asks the user for their name. Once they have entered their name, the application should join the user to the queue, display their name on the screen, and regularly update the amount of time they have been waiting on the queue

(roughly once every ten seconds). Further, the "Questions Asked" heading really refers to the questions answered since opening the application, and is tracked separately for quick & long help.

Note that mac and linux users may experience issues using the background/bg keyword with buttons. Your assignment only needs to work on the windows computers in the lab. Interested students can investigate highlightbackground as an alternative.

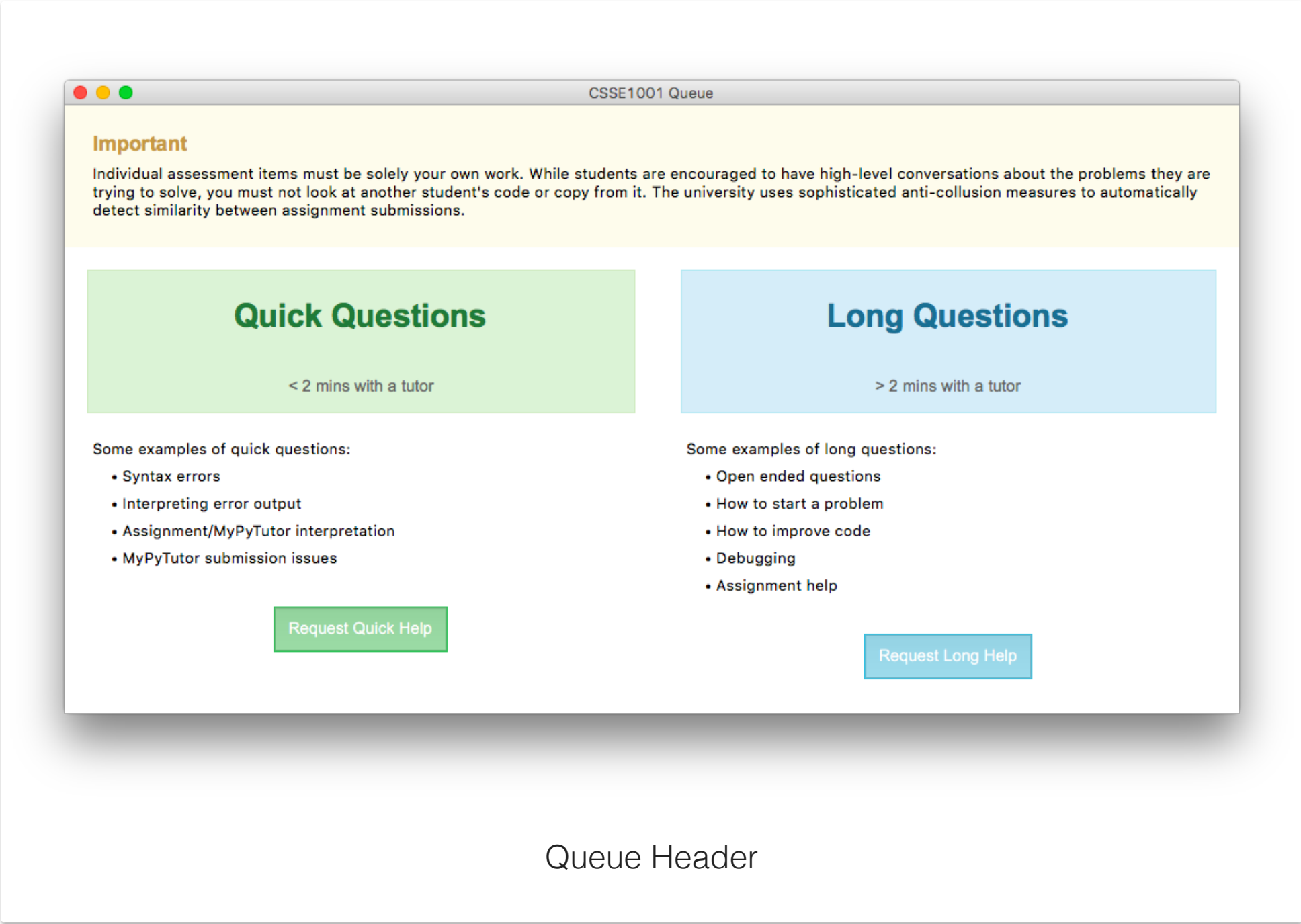
When a student is added to the queue, they must be added in such a way that no student who has asked a greater number of questions is placed ahead of them. A student may be removed from the queue by either pressing the red button - voluntarily cancelling their request, or by pressing the green button - marking the request as accepted.

2. Part One

2.1. Queue Header

The first task when implementing part 1 of the assignment is to create a GUI for the top half of the queue. The half includes the yellow notice at the top, the blue and green headers for each type of question, a list of example questions and buttons for requesting help. The text for each section must match exactly.

An example of this is shown below.

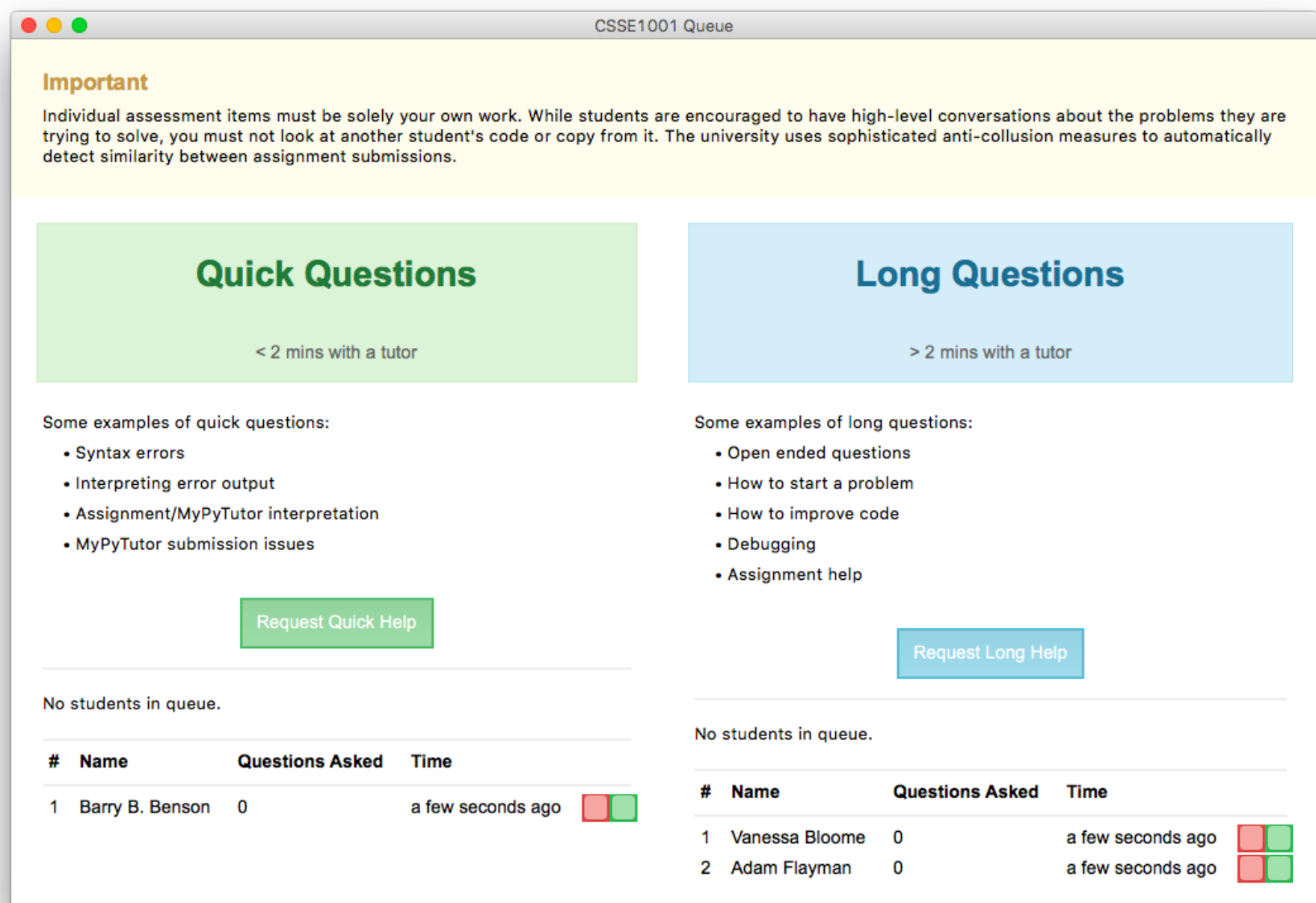


2.2. Student Queue

The second task is to implement a table that lists students in a queue. This section should have the base functionality of being able to add students to the queue by clicking the request help button. It should also have the ability to remove students from the queue by clicking either the red button or the green button.

To receive marks for this section you do not need to implement the functionality listed in the functionality section. This means that the "Questions Asked" and "Time" columns may contain the placeholders of "0" and "a few seconds" respectively.

The screen output from performing this task is shown below.



Student Queue

2.3. Full Queue Functionality

The final task of part 1 is to implement the full functionality of the queue (minus networking). This functionality is documented below.

- Keep track of how many times a question has been asked by the same student since opening the application. This number should be tracked separately for each queue and should not be updated if they were removed by clicking the red button.
- Keep track of the approximate time a student has been waiting on the queue according to the following table. Where **x** is used in the display, the value should be rounded down - i.e. a wait time of 13:37 should be displayed as "13 minutes ago".

Wait Time	Display
< 1 minute	a few seconds ago
< 2 minutes	a minute ago
< 1 hour	x minutes ago
< 2 hours	1 hour ago
>= 2 hours	x hours ago

- Prevent the same student from joining multiple times
- Display the average wait time for the queue, e.g. An average wait of about 12 minutes for 2 students.
- Order the queue first by "Questions Asked" in descending order, then by "Time" in ascending order. I.e. Students who have asked fewer questions should be placed to the top. If two students have asked the same number of questions, the one who has waited longer should be placed to the top.

In order to keep track of how long students have been waiting on the queue, you should record a timestamp using the `time` module. You must not base the accuracy of your timing on assumed time waited by tkinter's `after` method - `.after(10000, ...)` is only guaranteed to wait **at least** ten seconds.

3. Part Two: Additional Features

Part 2 of the assignment requires you to add one or more new features to the software developed in Part 1. To obtain a 7 in CSSE1001/7030, you must obtain more than 50% of the marks for Part 2 of the Assignment. It is up to you which additional feature(s) you add, but it must be substantial enough to demonstrate your expertise in writing software. Some suggested example features are:

1. A python internet based queue
2. Provision of a game which the user can play while they are waiting to have their questions answered

All additional features must be written in Python. To appraise the suitability of a given additional feature, you are advised to consult with tutors or post a private query on Piazza.

For Part 2 of the Assignment you must also submit a pdf document which describes the additional feature(s) you have implemented, a brief user manual on the how to interact with your additional feature software, and the approach you took to writing the code.

4. Masters Task

Masters students (i.e. CSSE7030 students) must do an additional task. They must implement a new top level window to display statistics about the history of questions that have been asked.

The style is left up to the discretion of the student and your window will be marked on the sophistication of the styling of the window. The window must display at least the following information:

- A leader board of students who have asked the most questions for both types of questions. It should display a minimum of 10 students for each.
- Mean wait time for both types of questions.
- Median wait time for both types of questions.
- Mode wait time for both types of questions.

Part marks can be earned for the style of the window without having it implemented correctly.

5. Criteria

Criteria	Mark
Queue Header	10
Header Mimics Look	8
Style & Documentation	2
Student Queue	5
Queue Mimics Look	4
Style & Documentation	1
Full Queue Functionality	6
Queue Functions As Expected	4
Style & Documentation	2
Extension	4
Sophistication of Style	2
Functionality of Window	2
Total	/25

* The masters task will be worth 20% of their overall grade. Undergraduates can implement the masters task but will receive no marks.

6. Assignment Submission

Your assignment must be submitted via the assignment three submission link on Blackboard. You must submit a file, `a3.zip`, containing your submission for this assignment. The zip file must contain a file called `a3.py` which, when run, will display your application.

Late submission of the assignment will **not** be accepted. Do not wait until the last minute to submit your assignment, as the time to upload it may make it late. Multiple submissions are allowed, so ensure that you have submitted an almost complete version of the assignment well before the submission deadline of 8:30pm. Your latest, on time, submission will be marked. Ensure that you submit the correct version of your assignment. In the event of exceptional circumstances, you may submit a request for an extension. See the [course profile](#) for details of how to apply for an extension.

Requests for extensions must be made **no later** than 48 hours prior to the submission deadline. The expectation is that with less than 48 hours before an assignment is due it should be substantially completed and submittable. Applications for extension, and any supporting documentation (e.g. medical certificate), must be submitted via [my.UQ](#). You must retain the original documentation for a *minimum period* of six months to provide as verification should you be requested to do so.

Change Log

Version 1.0.1 - October 2nd

- Changed due date to the correct day of the week.
- Clarified that the GUI only needs to work on the lab computers.
- Fixed screenshot to include average wait times.