

CSE542 - Software Engineering Concepts GUI Submission Requirements

Each team's submission must include:

- **Java source code** for the GUI program the team wrote. Your program should contain all of the GUI functionality the program requires. It should **display the current queue state** and include the option for users to **add an entry to the queue, remove an entry from the queue, and "pause" and "unpause" individual entries.**
- **When adding an entry, your program should require the user to enter their name, email address, and a "session password". Entries should always be added to the end of the queue.** Because this is only a prototype, **you can assume that the email address is valid and is not already on the queue.**
- **When removing an entry, your program should require the user enter their "session password".** Because this is only a prototype, **always assume that the "session password" is correct.** For this early phase, **advance all lower entries, including "paused" entries, when an entry is removed.**
- **Allow the user to "pause" and "unpause" an entry.** When this happens, **your program should require the user enter their "session password".** Because this is only a prototype, **always assume that the "session password" is correct.** Your program must **clearly identify which entries in the queue are currently "paused".**
- Your program should **begin with a random number of entries in the queue** (the number of entries should be **between 0 – 4**). Each entry will need to have **a name and "e-mail address" and be "unpaused".** You should use a preset list of names and email addresses to populate these entries.

Anything that uses work by non-team members MUST include proper attribution and be used in accordance with the licensing requirements of the site where it was found. Note that under United States law, work is automatically copyrighted with no permission for use provided unless EXPLICITLY stated otherwise.

Each team must submit a ZIP file containing their Java source files in AutoLab.

CSE542 - Software Engineering Concepts

Grade Sheet – GUI

GUI Application

Initial Display

Excellent (20-18)	Acceptable (17-10)		Bad (0)
Populates the queue as required -AND- Displays the queue of entries -AND- Ways to pause, add, and remove entries are obvious -AND- Order of entries in the queue is clear and obvious -AND- Clearly indicates when the queue is empty -AND- Code to do this works correctly	Populates the queue as required -AND- Displays the queue of entries -AND- Includes ways to pause, add, and remove entries -AND- Code to do this works correctly		Does not populate the queue as required -OR- Does not display the queue of entries -OR- Does not include ways to pause, add, and remove entries -OR- Code to do this contains 1 or more bugs

Adding Entries

Excellent (20-18)	Excellent (17-10)	Poor (9 - 6)	Bad (5 - 0)
Requires the user to enter a name, email address, and "session password" to add an entry -AND- Data are collected in a manner that is easy to use and navigate -AND- Adds the entry to the end of the queue so that only the name is displayed -AND- Code to do this works correctly	Requires the user to enter a name, email address, and "session password" to add an entry -AND- Data are collected in a manner that is easy to use and navigate -AND- Adds the entry to the end of the queue -AND- Code to do this works correctly	Requires the user to enter a name, email address, and "session password" to add an entry -AND- Adds the entry to the queue -AND- Code to do this works correctly	Cannot add entries to the queue -OR- Code to add an entry to the queue contains 1 or more bugs

Removing Entries

Excellent (20-18)	Excellent (17-10)	Poor (9 - 6)	Bad (5 - 0)
<p>The option to remove an entry is available only when an entry in the queue is selected -AND-</p> <p>Requires the user enter a "session password" to remove the selected entry -AND-</p> <p>Data are collected in a manner that is easy to use and makes clear the entry being removed -AND-</p> <p>Removes the selected entry and advances the entries below it in the queue -AND-</p> <p>Code to do this works correctly</p>	<p>Requires the user to enter an email address and "session password" to remove an entry -AND-</p> <p>Data are collected in a manner that is easy to use and navigate -AND-</p> <p>Removes the entry with the specified email address and advances the entries below it in the queue -AND-</p> <p>Displays an appropriate error message in a noticeable location when there are no entries in the queue with that email address -AND-</p> <p>Code to do this works correctly</p>	<p>Requires the user to enter an email address and "session password" to remove an entry -AND-</p> <p>Removes the entry with that email address from the queue -AND-</p> <p>Code to do this works correctly</p>	<p>Cannot remove entries from the queue -OR-</p> <p>Code to remove an entry from the queue contains 1 or more bugs</p>

Pausing Entries

Excellent (15-14)	Excellent (13-8)	Poor (7 - 4)	Bad (3 - 0)
<p>The option to pause an entry is available only when an unpaused entry in the queue is selected -AND-</p> <p>Requires the user enter a "session password" to pause the selected entry -AND-</p> <p>Data are collected in a manner that is easy to use and makes clear the entry being paused -AND-</p> <p>Updates the display to show the selected entry is paused -AND-</p> <p>Code to do this works correctly</p>	<p>Requires the user to enter an email address and "session password" to pause an entry -AND-</p> <p>Data are collected in a manner that is easy to use and navigate -AND-</p> <p>Updates the display to show the entry with that email address is paused -AND-</p> <p>Displays an appropriate error message in a noticeable location when there are no entries in the queue with that email address or that entry is already paused -AND-</p> <p>Code to do this works correctly</p>	<p>Requires the user to enter an email address and "session password" to pause an entry -AND-</p> <p>Updates the display to show the entry with that email address is paused -AND-</p> <p>Code to do this works correctly</p>	<p>Cannot pause entries in the queue -OR-</p> <p>Code to pause an entry in the queue contains 1 or more bugs</p>

Unpausing Entries

Excellent (15-14)	Excellent (13-8)	Poor (7 - 4)	Bad (3 - 0)
<p>The option to unpause an entry is available only when a paused entry in the queue is selected -AND-</p> <p>Requires the user enter a "session password" to unpause the selected entry -AND-</p> <p>Data are collected in a manner that is easy to use and makes clear the entry being unpaused -AND-</p> <p>Updates the display to remove the changes which showed the entry with that email address was paused -AND-</p> <p>Code to do this works correctly</p>	<p>Requires the user to enter an email address and "session password" to unpause an entry -AND-</p> <p>Data are collected in a manner that is easy to use and navigate -AND-</p> <p>Updates the display to remove the changes which showed the entry with that email address was paused -AND-</p> <p>Displays an appropriate error message in a noticeable location when there are no entries in the queue with that email address or that entry is already unpaused -AND-</p> <p>Code to do this works correctly</p>	<p>Requires the user to enter an email address and "session password" to unpause an entry -AND-</p> <p>Updates the display to remove the changes which showed the entry with that email address was paused -AND-</p> <p>Code to do this works correctly</p>	<p>Cannot unpause entries in the queue -OR-</p> <p>Code to unpause an entry in the queue contains 1 or more bugs</p>

Professionalism

Excellent (10)	Acceptable (9 - 8)	Poor (7 - 6)	Bad (5 - 0)
<p>GUI application is easy to use -AND-</p> <p>All classes are defined by their attribute(s) and the operation(s) which act on these attribute(s) -AND-</p> <p>Listeners are created as independent classes so they can be tested</p>	<p>GUI application is easy to use -AND-</p> <p>Most classes are defined by their attribute(s) and the operation(s) which act on these attribute(s)</p>	<p>GUI application requires minimal explanation before using it -AND-</p> <p>Many classes are defined by their attribute(s) and the operation(s) which act on these attribute(s)</p>	<p>GUI application requires substantial explanation in order to use it -OR-</p> <p>Few or no classes are defined by their attribute(s) and the operation(s) which act on these attribute(s)</p>

(For grading purposes, a Listener class that does not need any attributes to respond to its specific event will be score as if it were "defined by its attribute(s) and the operation(s) which act of these attribute(s)".)

Total Score (sum of individual scores)
