Criterion B- Solution Overview

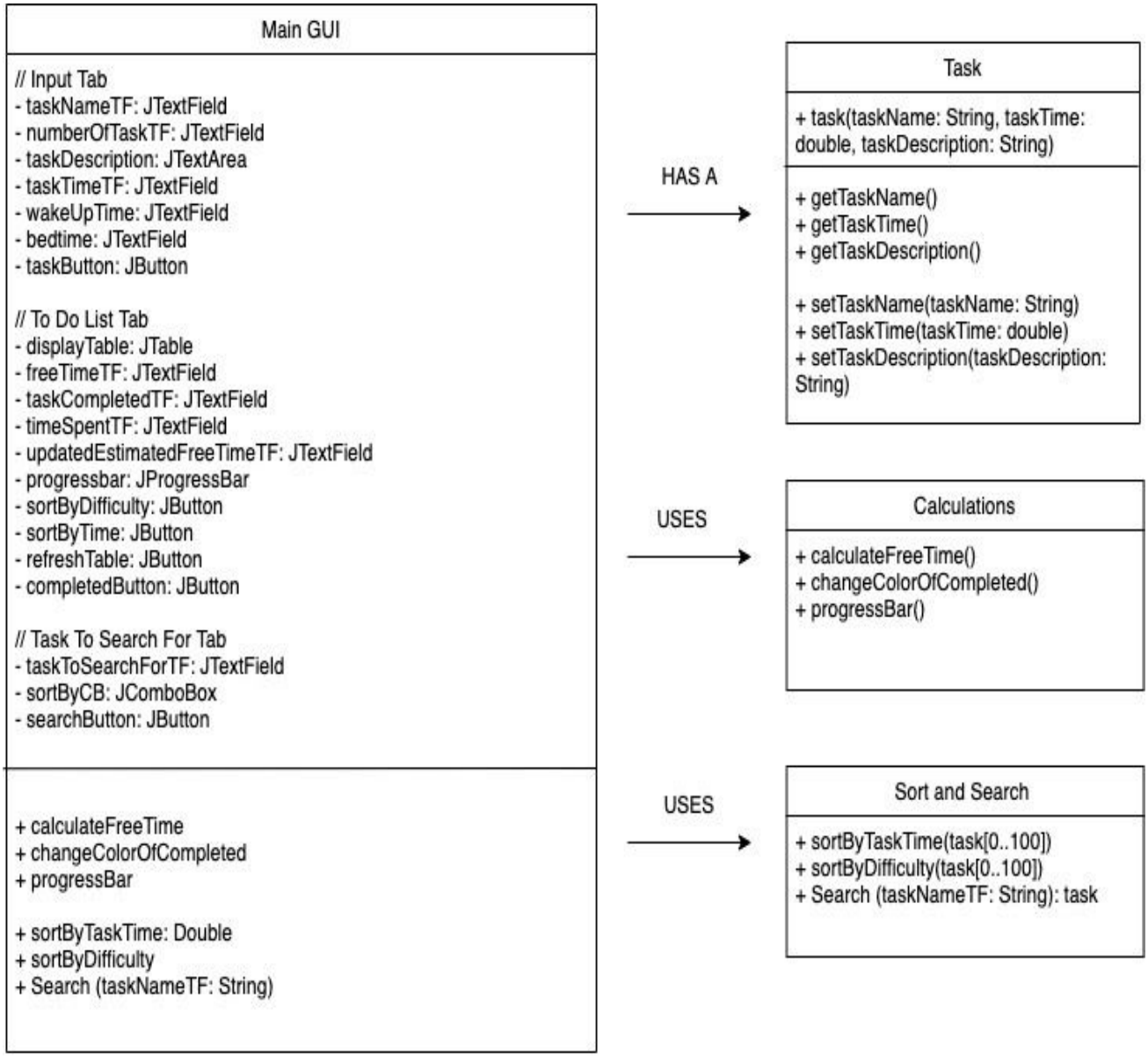
Input

Input	Data Type	Normal Range	Example
Name of task	String	N/A	Reading
Detail Description of task	String	N/A	Read a book for English class
Time to spend on task	Int	<10 hours	1
Difficult Level	Int	1-10	4
What task has been completed	String	N/A	Reading
How long did the client spend on the completed task	Int	1-10 hours	2

Output

Output	Data Type	Normal Range	Example
List of all the tasks within a given day including the description, difficulty level, and time	Array	0-10	Read | finish book | 5 | 3
List of tasks showing which tasks are completed and the actual time spent on the task	Array	0-10	Read | completed | 2
List of completed tasks (Progress bar of completed tasks)	GUI progress bar	N/A	A visual representation of all of the completed tasks the client has done

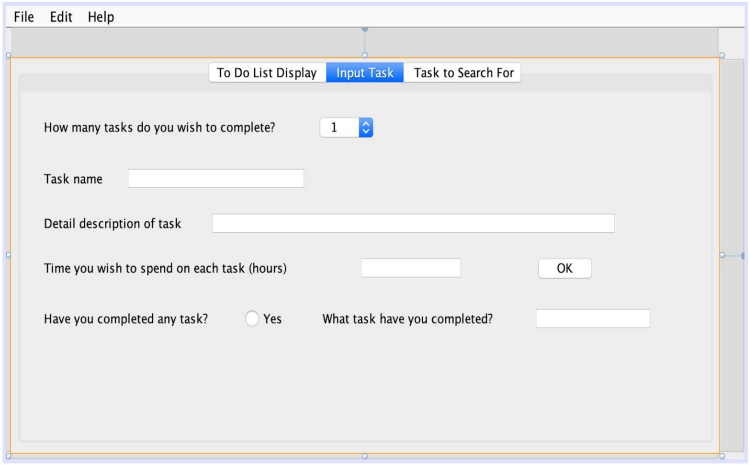
Class Diagrams



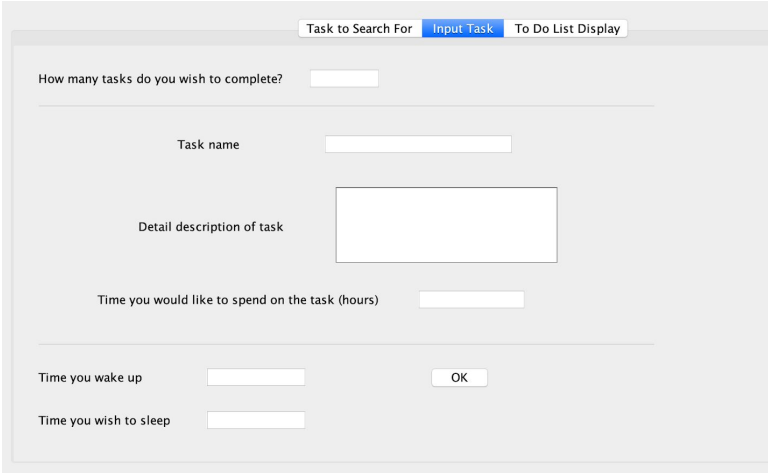
Final Prototype

The original prototype was edited based on the client's feedback in the second interview, thus numerous small changes were made regarding Mr. Eng's preferences. The final prototype was altered from the edited to fit his criterias better.

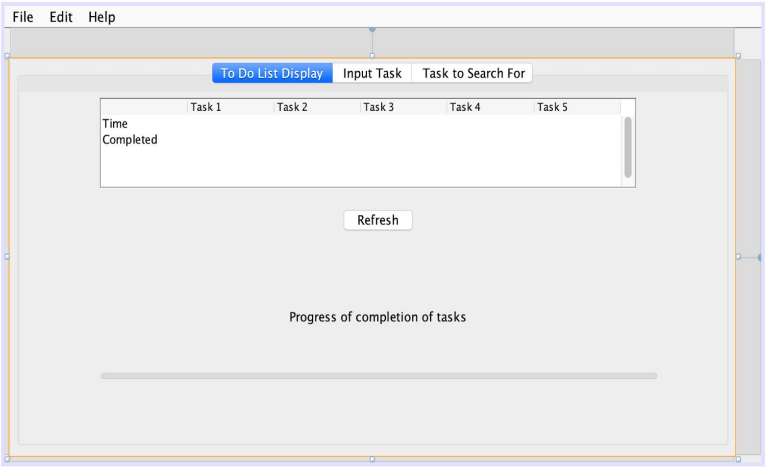
Original GUI Prototype

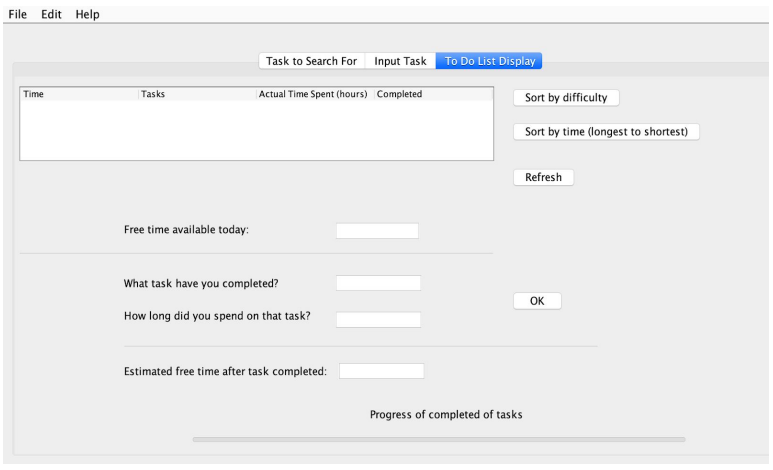


Edited Prototype



Mr. Eng stated[1] that he would like to have another input asking if he has completed the task in the To Do List Display tab instead of the Input Task Tab.





The data in the table can be sorted in two different ways in the edited prototype. And the additional factor of how long the client initially desired to spend on a task versus the actual time it took for a more efficient time allocation in the future[2].

---

[1] Andrew Eng, interview #2 by Kaila Eng, Bangkok, November 23, 2019, audio recording 02:55, Appendix
[2] Andrew Eng, interview #2 by Kaila Eng, Bangkok, November 23, 2019, audio recording 05:30, Appendix

Word Count: 120

Chronological Development Plan

Preparation (1 hour)

- ➔ Have all appropriate variable names
- ➔ Add in basic MouseReleased events button

Create MainGUI (4 hours)

- ➔ Do handling code for all four tabs (OK buttons)
- ➔ Make a code for refreshing the table display
- ➔ Handling code for ToDoListDisplay (Refresh buttons)
    - Loop through tasks using the for loop
    - Code for the sort buttons on ToDoListDisplay (bubble sorting algorithm)

Make appropriate classes (2 hours)

- ➔ Make a task class (template class)
- ➔ Establish gets and sets in task class
- ➔ Make a bubble sorting class
    - ◆ Sort through task class
    - ◆ Choose a type of bubble sorting algorithm

Progress bar coding (30 minutes)

- ➔ Create booleans for functioning progress bar
- ➔ Must have a while loop (while taskCompleted == true)
- ➔ Input basic math equations (number of completed / total number of tasks) x 100

Word Count: 124

Testing Plan

Input	Normal	Border	Abnormal	Extreme	Handling error
Task Name	Math Homework	Characters such as “a” Short names are okay, but no name → re-entering	Numbers	More than 100 characters	If the task name box is empty, and the OK button was pressed, then the program will not take in any of the other

					information.
Detail description of task	Complete the class handout and work on pages 123 questions 1abc, 2bca Or no description	No description More than 500 characters	N/A	More than 1000 characters	If there are more than 1000 characters, then the client would have to scroll along the display table in order to read it (inconvenient).
Time to spend on the task	3	0	Negative numbers	Large numbers such as hundreds, thousands, etc.	If there are large numbers, the program will still take in all of the information.
Difficulty Level	4	1 & 10	No answer (using combo box)	N/A (using a combo box)	The program will continue to take in the information even without any chosen difficulty level.
What task have you completed	Reading	N/A	Misspelling the task name, using integers	Inputting a task that was never created	If there is a misspelling of the task there should be a popup where it notifies a suggested task for the client.
How long did you spend on the task	2	0	Negative numbers	Large numbers such as hundreds, thousands, etc.	Program will take in the information and continue as usual if there are large numbers.