My Very Own GUI Life Coach

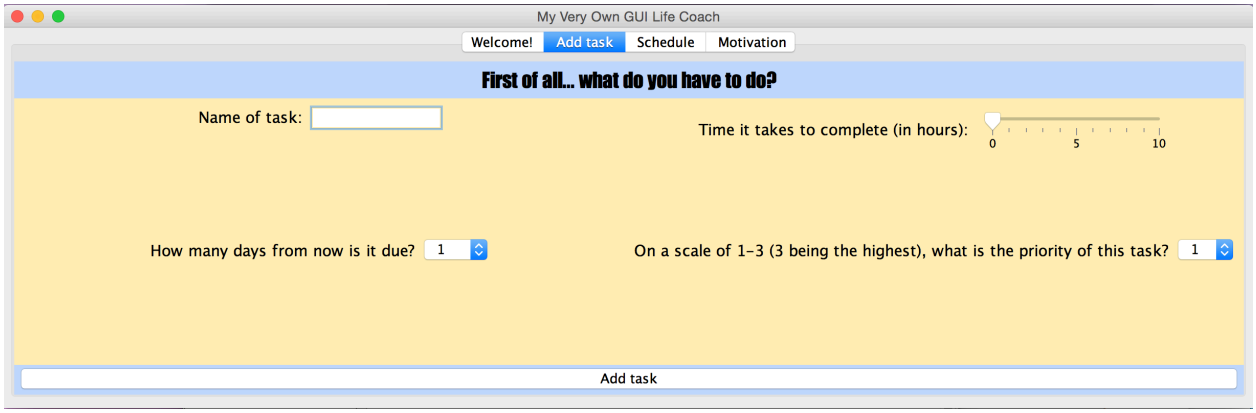
CS 230 Spring 2017 Final Project

Julia Klugherz, Sara Vannah, Andjela Stojkovic

1. User Manual

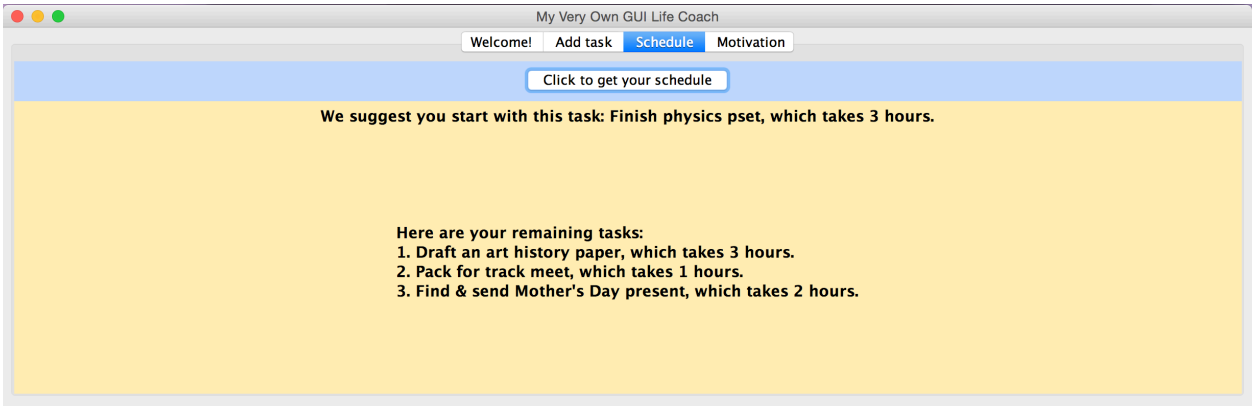
The modern American woman has a hectic array of tasks to juggle and keep track of. We want to help. Our application allows users to input and cross-off tasks as they arise and are completed. Upon opening the program, the user is greeted by a tabbed GUI pane titled “Welcome”.

Clicking on a second “Add Task” panel allows the user to input their tasks. A textfield is provided for the name of the task. A slider is used to input the number of hours required to complete the task; drop-down menus are provided to input the number of days until the task is due and the priority on a scale from 1-3, with 3 being the highest priority. After the information about the task is entered, the user clicks the “Add task” button to add the task to her to-do list. More tasks can be added by changing the input fields and again clicking “Add task”. The program uses the information the user has provided about the task to determine an order for the tasks to be completed.

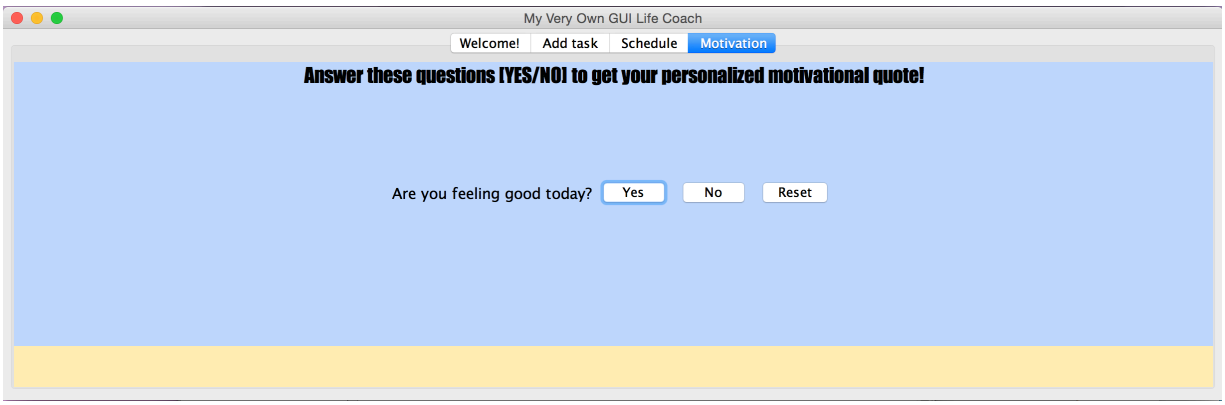


Within the code, the tasks are held in a LinkedList. When a task is added, a weight is determined for the task given by the sum of time in hours required to complete the task and the priority factor given by the user, minus the number of days until the task is due. The weight is then used to determine where in the LinkedList the new task should be inserted - corresponding to when the task should be completed.

To view the customized schedule, the user clicks on the "Schedule" tab. This tab displays the task that should be completed first, as well as the time needed to complete the task. In addition, the remaining tasks (up to 10) are shown on this tab with the time required to complete them, to help the user plan her day. To see your tasks, click the "Click to get your schedule" button.



The last tab, titled "Motivation," will provide the user with a specific motivational quote based on their answers to questions in a binary decision tree. The program implements a LinkedBinaryTree of Strings, which hold the questions. The user answers Yes/No questions and the program presents the user with a motivational quote catered to their circumstance. The program will end here, with the user seeing their schedule properly formatted and their time beautifully managed, and a perfect motivational quote to get them started.


My Very Own GUI Life Coach
Welcome!
Add task
Schedule
Motivation
Answer these questions [YES/NO] to get your personalized motivational quote!
Are you feeling good today?
Yes
No
Reset

2. Technical description

a) ADT's:

A LinkedList holds the tasks in order of calculated priority, using variables like priority factor, due date of task, and length of time it takes to complete to sort.

A Binary Decision Tree implementation of LinkedBinaryTree is used to decide which motivational quote the program should present to the user.

b) Classes:

1. Task.java → instance variables: String name, int timeItTakes, int dueDate, int PriorityFactor
2. ToDoList.java → uses LinkedList to hold tasks, calculates a weight for each task based on user-input parameters to sort incoming tasks
3. Quotes.java → uses LinkedBinaryTrees to determine which motivational quote to present to the user
4. LifeCoachDriver.java → GUI driver class
5. GUI Panels:
    a. WelcomePanel.java → welcome panel
    b. InputTasks.java → panel where users input task information
    c. SchedulePanel.java → panel where up to 10 most important tasks are listed in the order they should be completed, with the time required to complete them
    d. MotivatePanel.java → uses Binary Decision Tree to present user with specific quote
6. Javafoundations
    a. BTNode.java
    b. BinaryTree.java
    c. LinkedBinaryTree.java

c) Actions:

1. addTask() – Add task to proper LinkedList in Vector depending on priority
2. Getters/setters for all instance variables in addTask
3. orderTasks() – Calculate the order in which tasks should be completed in each list
4. toString() methods for Task and toDoList
5. findTask() – allows search for task
6. getFirstTask() – gets first task in ToDoList
7. getNextTasks() – gets next tasks (up to task #10) from ToDoList other than first task
8. getRightString() – gets the right String from the right LinkedBinaryTree (can be another question or the motivational quote)
9. getLeftString() – gets the left String from the left LinkedBinaryTree (can be another question or the quote)
10. getRootQuestion() – get the question at the root of the tree
11. resetTree() – resets the tree to the initial question, used when the user presses reset in the GUI