Criterion B: Solution Overview

Inputs & Outputs

Inputs

Input	Data Type	Example
Date (dd/mm/yyyy)	Date class	13/02/2019
Time Spent Walking (hrs)	double	0.5
Time Spent Biking (hrs)	double	1.0
Time Spent Swimming (hrs)	double	1.5
Week Number	integer	3
Current Weight of Client (kg)	double	86.2
Weekly "How I Felt" scale	ComboBox	4

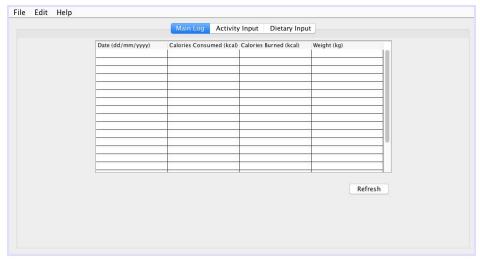
Outputs

Output	Data Type	Example
--------	-----------	---------

Date (dd/mm/yyyy)	Date class	14/03/2019
Hours Walked (hrs)	double	0.5
Hours Biked (hrs)	double	1.0
Hours Swam (hrs)	double	1.5
Week Number	integer	4
Weight of Client Week by Week (kg)	double	84.3
Weekly "How I Felt" Scale	integer	6
Previous Week's "How I Felt" Score	integer	5
Recommended Hours to Walk for the Following Week	double	0.4
Recommended Hours to Bike for the Following Week	double	1.6
Recommended Hours to Swim for the Following Week	double	1.0

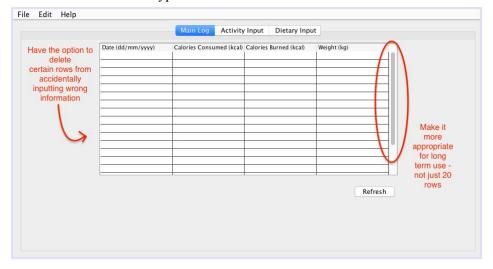
Prototype Development

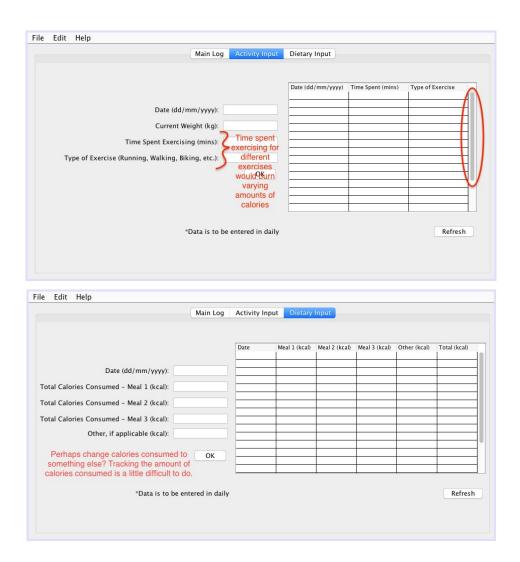
Original Prototype



	Main Log	Activity Input	Dietary	Input			
			Date (dd	/mm/yyyy)	Time Spent (min	s) Type of	Exercise
Date (dd/r	mm/vvvv):						
Current W					7		
Time Spent Exercisi	ng (mins):						
Type of Exercise (Running, Walking, Bik	king, etc.):						
		ОК			-		
	Data is to he	entered in daily	v				Refresh
Edit Help	Main Log	Activity Input	Dietary	Input			
Edit Help	Main Log	Activity Input	t Dietary	Input			
Edit Help	Main Log		Dietary Meal 1 (kcal)	Meal 2 (kcal)	Meal 3 (kcal)	Other (kcal)	Total (kcal)
Edit Help Date (dd/mm/yyyy):	Main Log				Meal 3 (kcal)	Other (kcal)	Total (kcal)
Date (dd/mm/yyyy):	Main Log				Meal 3 (kcal)	Other (kcal)	Total (kcal)
Date (dd/mm/yyyy): al Calories Consumed – Meal 1 (kcal):	Main Log				Meal 3 (kcal)	Other (kcal)	Total (keal)
Date (dd/mm/yyyy): al Calories Consumed – Meal 1 (kcal): al Calories Consumed – Meal 2 (kcal):	Main Log				Meal 3 (kcal)	Other (kcal)	Total (kcal)
Date (dd/mm/yyyy): al Calories Consumed – Meal 1 (kcal): al Calories Consumed – Meal 2 (kcal):	Main Log				Meal 3 (kcal)	Other (kcal)	Total (kcal)
Date (dd/mm/yyyy): al Calories Consumed – Meal 1 (kcal): al Calories Consumed – Meal 2 (kcal): al Calories Consumed – Meal 3 (kcal):	Main Log				Meal 3 (kcal)	Other (kcal)	Total (kcal)
Date (dd/mm/yyyy): al Calories Consumed – Meal 1 (kcal): al Calories Consumed – Meal 2 (kcal): al Calories Consumed – Meal 3 (kcal):					Meal 3 (kcal)	Other (kcal)	Total (kcal)

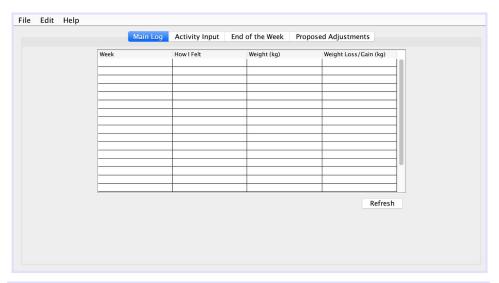
Prototype with Comments from the Client

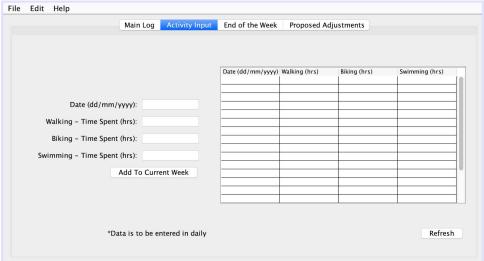


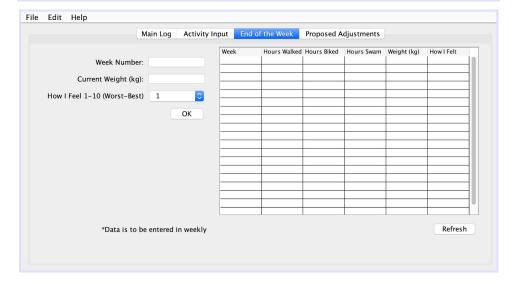


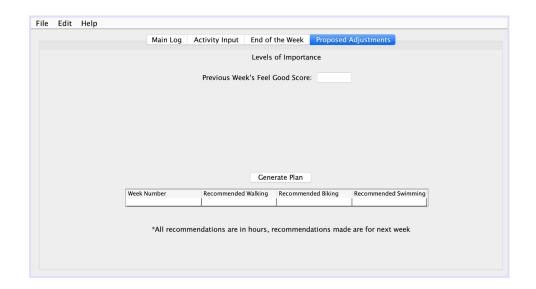
Interview 2: Proposed Improvements (NOT INCLUDED IN WORD COUNT)

The client made several comments regarding user friendliness and what should be included in the final features of the program. Some of the features that the client mentioned includes having the option to delete or edit cells in any displayed tables and having the program be more appropriate for long-term use. The client also suggested that it might be simpler for the program to track something else other than the total calories consumed. Perhaps a revamped third tab for the program would be appropriate.









UML Class Diagram Exercise Day + date: Date + timeWalked: double + timeBiked: double MainGUI has + timeSwam: double + exerciseDays: ArrayList + exerciseWeeks: ArrayList + ExerciseDay(date: Date, timeWalked: + walkingHrs: jTextField + bikingHrs: jTextField + swimmingHrs: jTextField double, timeBiked: double, timeSwam: double) + importanceScale: jComboBox + setDate(date: Date) + setTimeWalked(timeWalked: double) + setTimeBiked(timeBiked: double) + enterButtonAction + refreshButtonAction + setTimeSwam(timeSwam: double) Exercise Week & Analysis + getDate: Date + weekNumber: int + getTimeWalked: double + getTimeBiked: double + totalHoursW: double + totalHoursB: double + getTimeSwam: double + totalHoursS: double + currentWeight: double + feelGoodScore: int + makeProposal(): void uses + ExerciseWeek(weekNumber: int, totalHoursW: double, totalHoursB: uses double, totalHoursS: double, currentWeight: double. feelGoodScore: int) + setWeekNumber(weekNumber: int) SavingAndOpening + setTotalHoursW(totalHoursW: double) + setTotalHoursB(totalHoursB: + saveExercises(exercises[0...*]) + readExercises() + setTotalHoursS(totalHoursS: double) + setCurrentWeight(currentWeight: double) + setFeelGoodScore(feelGoodScore: int) + getWeekNumber: weekNumber + getTotalHoursW: totalHoursW + getTotalHoursB: totalHoursB + getTotalHoursS: totalHoursS + getCurrentWeight: currentWeight + getFeelGoodScore: feelGoodScore

Chronological Development Plan

Creation of the GUI (2 weeks)

- I. GUI after the **initial interview**
 - A. Create the first tabbed pane as an overall data log
 - 1. Is viewable as a table
 - B. Second pane for activity input
 - 1. Includes text fields for the input of new data
 - 2. Includes a table to display inputted data
 - C. Third pane for dietary input
 - 1. Includes text fields for the input of new data
 - 2. Includes a table to display inputted data
- II. Reworking of the GUI after having had a **second interview** with the client
 - A. Revamped third tab by taking out dietary input options and including weekly inputs, such as weight and "how I felt", as well as making calculations to total hours spent in each physical activity.
 - B. Added a fourth tab which makes suggestions for next week, in accordance to how the client felt, the amount of time spent in each activity within the particular week, and weight loss.

Template Classes (1 week)

- I. Create Main
- II. Create inheritance classes from the main ExerciseDay and ExerciseWeek

Saving and Reading (1 week)

- I. Creation of a **Save and Read** class
 - A. Read exercise data method
 - B. Save exercise data method

Word Count: 189