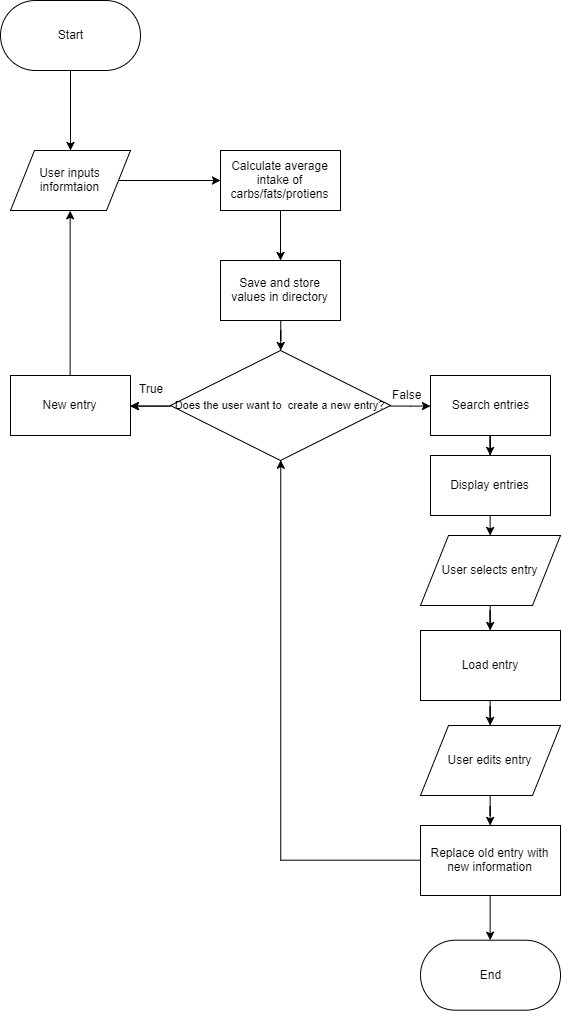
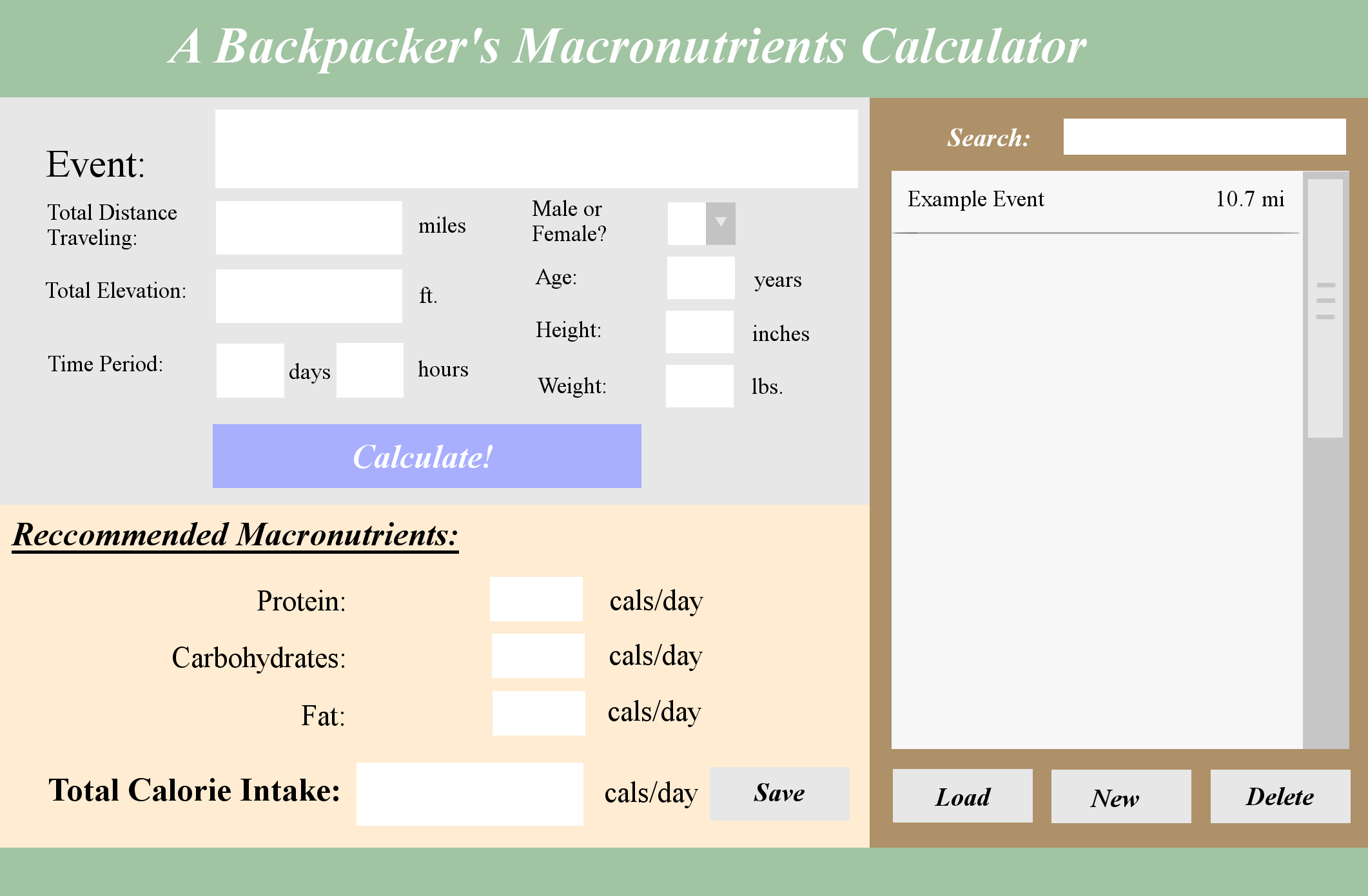
**Criterion B: Design**

The final product will be a calculator that takes the inputs of a scout’s age, weight, and height, and the location’s total distance, elevation, and time period to end up with a calculated result of total calorie intake along with a macronutrient ratio of fats, proteins, and carbohydrates needed for a backpacking/hiking trip in units of calories per day.

**Flow Diagram**

**GUI**



List of saved entries

Main Panel (Input/Output for entries)

**Data Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Description | Data Type | Sample Data |
| Event | Custom Class | |  | | --- | | Event | | -String: location  -int: distance  -int: elevation  -int: time | | +getLocation()  +setLocation(String:l)  +getDistance()  +setDistance(int:d) +getElevation()  +setElevation(int:e) +getTime()  +setTime(int:t) | | Event:  location: “Shenandoah national Park”  distance: 9 miles  elevation: 2,355 ft  time: 450 minutes |
| Person | Custom Class | |  | | --- | | Person | | -char: gender  -int: age  -int: height  -int: weight | | +getGender()  +setGender(boolean:s)  +getAge()  +setAge(int: a)  +getHeight()  +setHeight(int:h)  +getWeight()  +setWeight(int:w) | | Person:  gender: “M”  age: 18  height: 66 inches  weight: 123 lbs. |
| Events | Store a list of events | Array of combined Event and Person objects | n/a |
| List | inputFile | File | new File(“list.txt”) |
| FindTotalCal | Calculates total caloric intake using Person and Event data. | int | 16,000 kcal |
| FindRatio | Applies constant macronutrient ratio to FindTotalCal | int | (Ratio 30/40/30)  4,800 kcal of proteins  6,400 kcal of carbs  4,800 kcal of fat |

**Test Plan**

|  |  |  |
| --- | --- | --- |
| **Step Description** | **Input** | **Expected Outcome** |
| User opens the program | n/a | The program opens with available text field inputs ready. |
| User creates a new entry with “New” button | Button click | The form’s textfields are cleared and prompts the user to type in new information |
| User receive their calculated results with the “Calculate!” button | Button Click | The program takes data from the input fields (From items Event and Person) and calculates the total calories burned and applies a recommended ratio. These values are visible. |
| User saves entry with “Save” button | Button Click | Information on the form is saved onto a list of entries. |
| User selects entry from list | Entry Click | The selected entry is the only entry that is highlighted in the list. |
| User loads entry with “Load” button | Button Click | The program takes all the information from the selected entry and replaces the text fields in the main input panel. |
| User deletes entry with “Delete” button | Button Click | The user will be asked if they would like to delete the selected entry. The entry will be deleted if the user agrees. |
| User searches for an entry | Typed into an input text field | The program searches the current list of entries and highlights entries with matching keywords as the input. |
| User exits program | Button Click | The program will ask if the user would like to save any information currently inputted. |

**Pseudocode**

**Objects:**

Custom Class Person

Create Person()

Create constructors

Create getters and setters for char and int values

Method Int FindBMR (ints of Person):

Calculate Basal Metabolic Rate(BMR)

Custom Class Event

Create Event()

Create constructors

Create getters and setters for String and int values

Method Int FindMoreCal (ints of Events):

Calculate additional calories burned from backpacking using the int values

**Main Program:**

Class RunProgram

(#1)Method Main:

Create file for “list.txt”

Scanner inputFile

ArrayList entries

Print FindTotalCal

Print FindRatio

(#2)Method FillArrayList

While scanner.hasNextLine() :

Take values from objects Person and Events

Add to ArrayList1 of Person

Add to ArrayList2 of Events

Combine ArrayList1 and ArrayList2 into one ArrayList

return combined ArrayList

(#3)Method Save (input: ArrayList/ Output: ArrayList)

Take combined ArrayList events and put them into a larger ArrayList entries.

(#4)Method SearchandLoad (Input: String / Output: Entry):

String searched = Prompt user to search

String control = searched

Loop through first characters each getLocation Strings in the object Event

If (first chars of String searched == first chars of getLocation Strings)

Get the event of that matched chars

return event

(#5)Method FindTotalCal (Input: int FindBMR and int FindMoreCal/ Output: int) {

Takes FindBMR //calories burned while resting

and FindMoreCal //calories burned during activity

and combines them to get total calories

return total calories

(#6)Method int FindRatio (int of FindTotalCal)

\*If BMR or properties of Event object are a certain value, use a certain ratio

Apply that ratio to int FindTotalCal

Int protein

Int fats

Int carbohydrates

return protein, fats, and carbohydrates

**Complexity**

* Java Libraries: File, GUI libraries
* ArrayLists: Keeps a list of entries for the file, list.txt
* Sorting: Sorts the entries in accordance to distance
* File I/O: Creates a file for reading and writing of the list of entries.
* If/Else Statements: Used for male/female calculations and their differences
* Searching: The user can search for specific entries.
* Looping: Loops will be used for the search function.
* User-defined methods: The program is organized into different methods.
* User-defined objects: Person and Event objects are created for the calculations in the entries.