GUI Coursework 1

**Overview:**

The application is designed to give the user an easy experience to manage their fitness statistics.

**The Layout:**

**Initial design:**

Tabs - You:

This tab merges photos, which the user uploads when creating either an exercise or a diet change. In the program below, the user has simply recorded a record of his chest, for a personal goal. Based on feedback I found that users wanted a welcome message and recent photos for a personal touch.

The users BMI is calculated using the below algorithm:

Macintosh HD:Users:Harry:Dropbox:Computer Science:G52GUI:BMI.pngMacintosh HD:Users:Harry:Dropbox:Computer Science:G52GUI:BMI equation.png

The program only needs a regular update of the users weight (mass) because their height shouldn’t change rapidly, although the option is there for change.

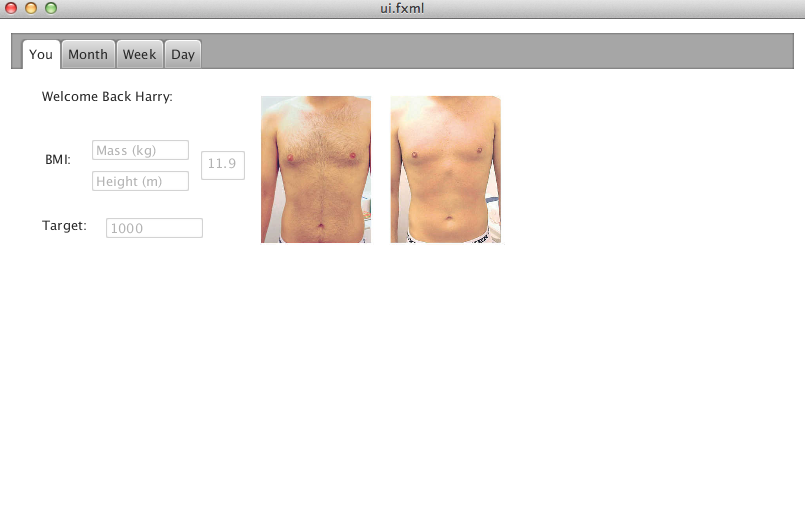
To try and encourage the users to reach goals and targets, the target is shown on every tab and is easily manageable from the home screen.

Based on user feedback a simple design is most effective otherwise with many tabs and buttons the user is over complicated with new methods to learn so it becomes less effective.

**BMI**

OnEdit: BMI will be automatically calculated and displayed in the box (11.9).

Reason: User feedback showed that an additional button click was unnecessary.



**Target**

OnEdit: Update the users target on each tab.

Reason: Should only be able to change their target occasionally, but they might want a new target for a single day.

**Photo**

OnClick: Move to the Day tab focused on the date the photo was taken.

Reason: User feedback showed that they wanted to click to see stats on that day and display when the photo was taken.

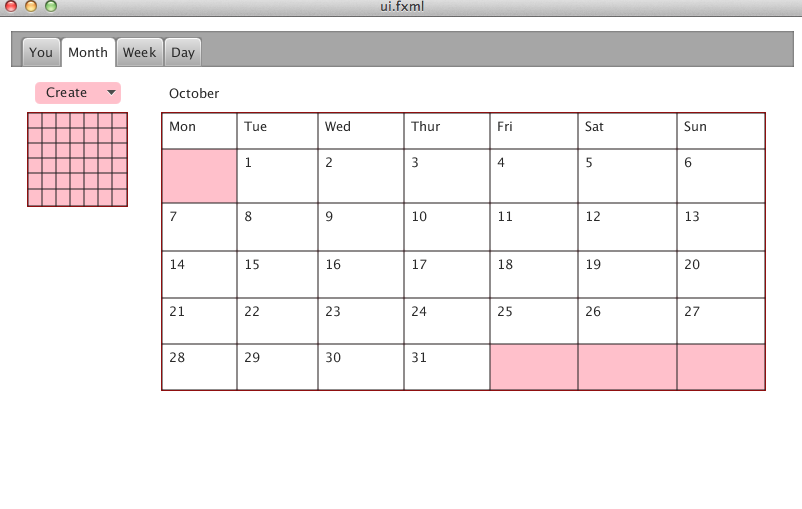
Figure 1. Main tab.

Tabs - Month:

From the requirements a Month display was created. This features a calendar where each month you can tell if you have reached your target or not.

Below the calendar there is a total of Kilojoules you have burnt and KJ eaten including the average amount of KJ eaten/burnt per day.

The function, totalKJ() will run when you switch to this tab, because the total will need to be updated every time you check as the user can create tasks in multiple tabs.

Figure 2. The month Tab.

**Day**

OnHover: Shows more detail statistics of:

* Target met
* KJ eaten
* KJ Burnt

Reason: Detailed information was a must have in a calendar view.

**Grid**

OnClick: Move to the Week tab focused on the date the user clicked.

Reason: User feedback showed that they wanted to click on a smaller version of the calendar which was on every screen.

The user must be able to record exercises or Dietary statistics where ever possible, so based on user feedback the Create drop down menu is on each tab page.

**Create**

OnClick: The user selects either an Exercise or a Dietary addition to the system.

Reason: Multiple choices are easier made before a complex screen is presented to the user.

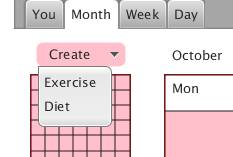


Figure 3. Create Menu

Create Dropdown Menu – Exercise Creator:

The dropdown menu is exactly the same as the Diet window, this means less complication for the user. The only difference is the title bar will read Exercise or Diet and the inline text for the input boxes will be relevant to the type.

This window uses the addExercise() method to add an exercise to the system.

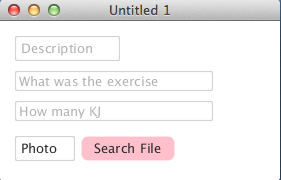
**Description**

OnEdit: A list of common descriptions will be displayed.

* Running
* Cycling

Added to the list will be the most recent exercise based on the time that you add this event.

Reason: Time is added and can be changed right of the description.



**12:15**

Figure 4. Create Exercise.

Create Dropdown Menu – Diet Creator:

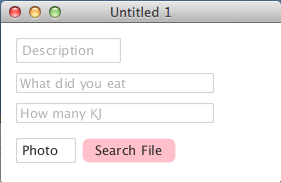
Visually similar to the Exercise creator but uses addDiet() instead.

The food type (what did you eat) is unique to Description because the actually type of food is used as a catalogue instead of the description, for example when editing the box, recent foods, popular foods will appear from this and the user can select and choose a recent one.

**Time**

OnClick: You can freely edit the time as requested by a user. Although the time gets the clock rounded to 15 minutes.

Reason: Most people don’t immediately plot actions, so previous time was needed.



**Photo Search File**

OnClick: The standard native file selection window appears and the user selects a photo for the exercise.

Reason: This is simple to try and cut down the user learning for this program.

**12:15**

Figure 5. Create Diet.

Tabs - Week:

A week view displays if the target has been met for each day, so the user can clearly see what days they have not reached their target KJ.

The date is clearly shown above the grid to reflect what days are being displayed.

**Week Grid**

OnHover: The grid is coloured a lighter shade of pink and the information about that day is displayed.

OnClick: Switches to the Day tab where all the information is detailed.

Reason: The users felt it was more natural to simply click on the day they wanted rather than switch tabs, then select the day.

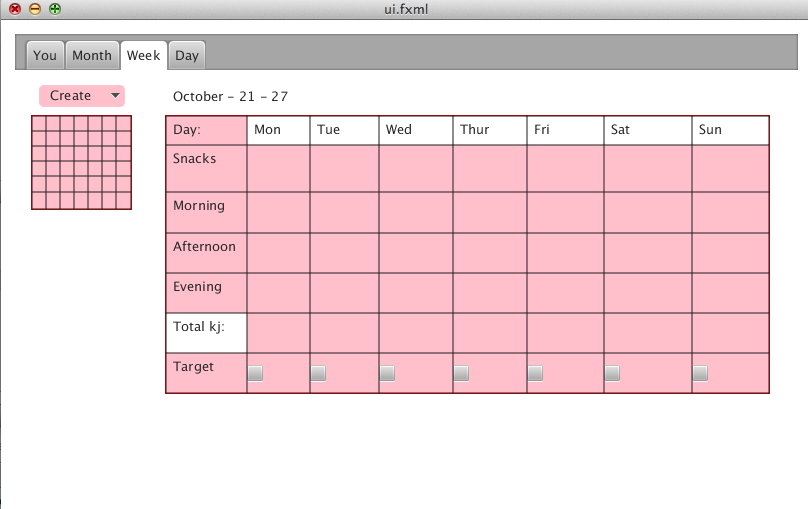


Figure 6. Week Tab.

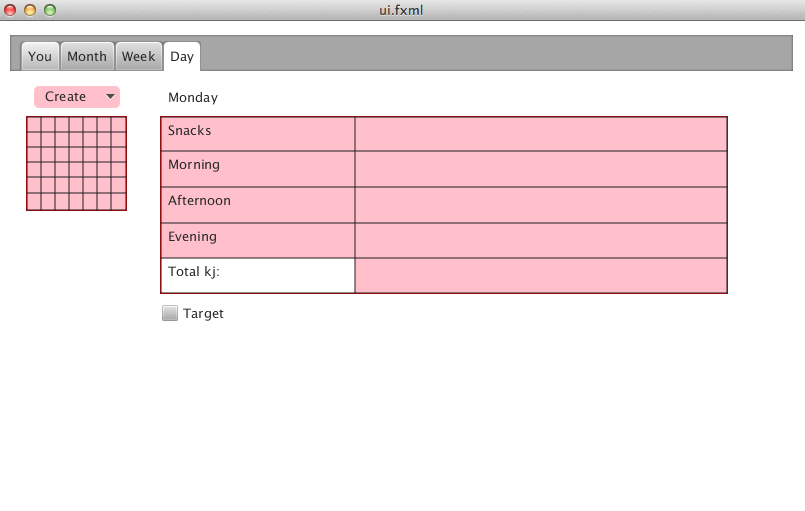
Tabs - Day:

Finally the day tab, lists detailed information about each day, including a list of foods that the user entered, the KJ they burnt/ate and what part of the day they used them.

**Day Grid**

OnClick: The user opens the create drop down menu, to select either an exercise or a diet addition.

Reason: This makes it even easier to add exercises or diets to the system.



**Time periods**

Morning: 3am – 12pm

Afternoon: 12pm – 6pm

Evening: 6pm – 3am

Reason: Normally the user eats at given times (e.g. dinner in evening) and exercises in these periods, so it was sensible to group these.

Figure 7. Day Tab

**The Data Model:**

Below is the initial data model diagram for the data in the database. I started with a clever 2 class technique where ‘Type” was both diet and exercise information. Where the isExercise Bool reflected whether it was an exercise or diet addition. Although reviewing this model from user feedback I found that a 2 class diagram would be insufficient because if the user wanted to adapt the program to find all the diet information they could but having a isExercise variable in a supposed diet class is not good practice.

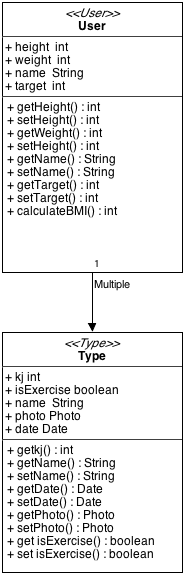


Figure 8. UML model Data only, first revision.

The below revised version contains an additional class separating the Exercise and Food for a simple, easy to view contained format.

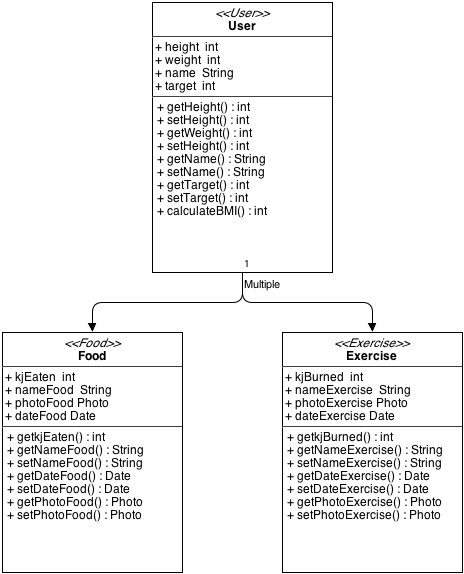


Figure 9. UML model Data only, revised version.

The UML model with all the GUI classes shows the process of the system, where the information flows and how each tab is navigated by the user.

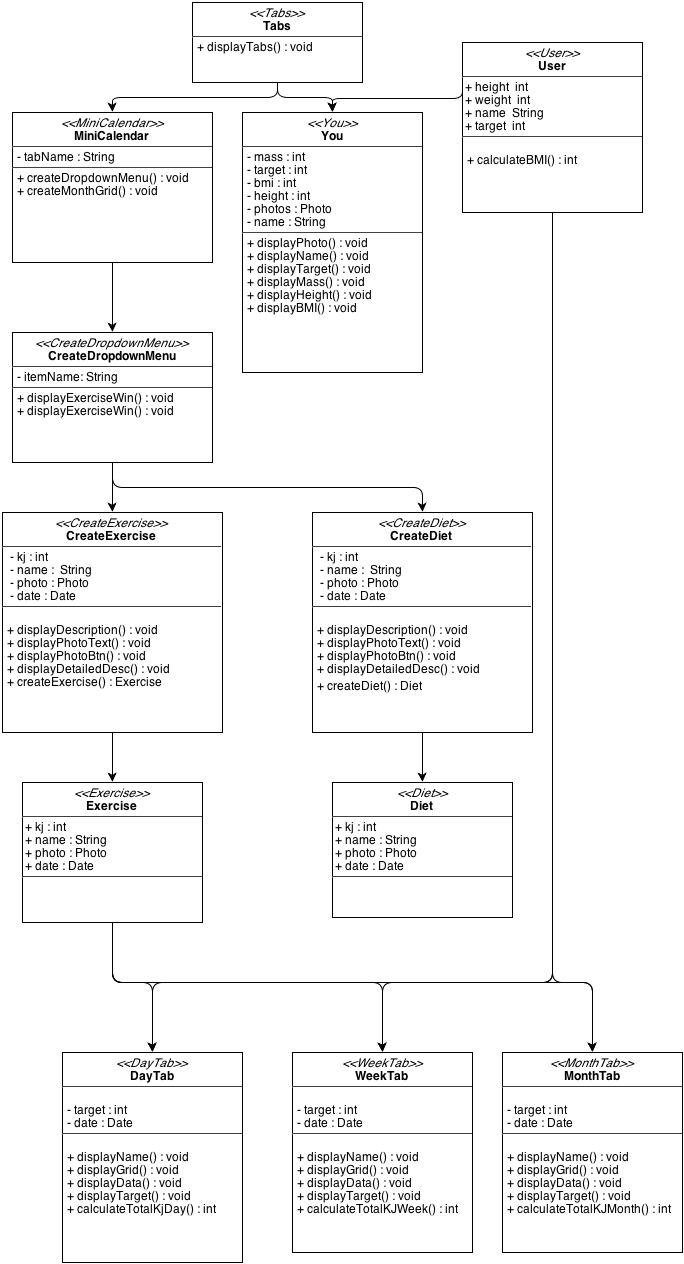


Figure 10. UML model.

**The ‘use’ cases:**

1. The user Inputs their Characteristics

Weight, height, name and target is added by the user when the program first starts, this means the BMI calculation can perform using this data to create a profile.

* OnEdit: for each box, the relevant variables are updated based on the boxes, the BMI calculation is run when both boxes have integers in them.
* OnClick: for each photo, the focus moves to the day tab for which date the photo was taken on.

1. The user either:
   1. Exercises

The description of the Exercise and what was performed can be entered in a simple box that the user can easily navigate. The number of KJ burnt is included in this window. When this is created the information is included in the system and displayed when the other tabs are viewed.

* + OnKeyPress(return): The data is saved to a csv file with the exercises added.
  + OnButtonPress(Photo): The default native file searching window appears and the user selects a photo from their computer.
  1. Eats food

The description of the food and what was actually eaten can be entered in a simple box that the user can easily navigate. The number of KJ eaten is included in this window. When this is created the information is included in the system and displayed when the other tabs are viewed.

* + Same as Exercises, just using the Food object.

1. The user reads the records

Clicking on each tab displays the total amount of KJ the user eaten minus the number they ate, this calculation on every tab relevant to each tab. Ex: Month tab calculates KJ per month average.

* OnEdit: clicking on any calendar on either tab, brings up the [Exercise,Diet] drop down menu, selecting this brings up the create window.
* OnHover: information about the day is highlighted when hovering over a day.
* OnClick: the relevant day is viewed by focusing on the day tab.

The use case diagram:

The user:

1. Views records of their information.
2. Inputs basic characteristics, i.e. change of weight or target
3. Exercises
   1. Inputs the exercise and/or photos into the system
4. Eats food
   1. Inputs the dietary information and/or photos into the system

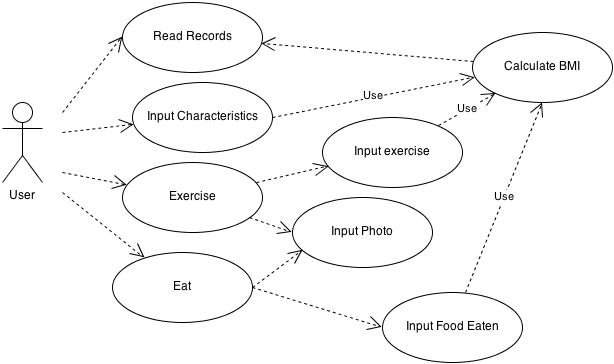


Figure 11. Use case diagram.

**User Feedback:**

I produced a demo of the application describing the main requirements and asked the audience to give helpful feedback for improvement. This was the response I got:

“Ability to edit the time on creation”

“Ability to create tasks by clicking on anywhere on the calendar”

“Remove the button to calculate the BMI, should do this automatically”

“Clicking on a photo should bring the application to the day it was taken”

“Always have a calendar available on screen, so can easily edit from any screen”

“Screens are too complex, need a more simple design”

**Conclusion:**

The pre designs were changed dramatically based from user feedback, and I feel this helped sculpt the application improving its overall ability.