

Executive Summary

Business Solution

I wanted to implement a spreadsheet through which someone could track their eating habits. Oftentimes I think that people start tracking their eating habits but it is cumbersome and they give up easily—at least that’s what I do. I wanted to create a system that allowed users to easily enter food into a database, track food for individual days, and search for food in the database to add to a day. Also, because goals are so important, I implemented a goal feature whereby a user can track on a day-by-day basis how close (or far away) they are from hitting their goal.

System Overview

The system has the following capabilities:

1. **Insert Day.** The user can insert a new day to track their food for that day.
 - a. **Error Checking.** Two days with the same date cannot be created.
 - b. **Quick date entry.** The user can use today’s date quickly by simply leaving the input field blank and pressing OK.
 - c. **Chronological Date Entry.** When inserting a new day, it will find the right place chronologically to insert the data
2. **Clear All.** The user can clear all of the data on the spreadsheet.
3. **Add food items to a day.** The user can add food items to the day. The totals for each nutritional category are calculated and placed in the “Totals Row” of the day.
 - a. Will validate that the date exists in the spreadsheet already.
 - b. If date does not exist will exit the procedure.
 - c. Search feature allows for user to enter all or part of the food description. Once the food is found the form auto-populates with information from the database.
 - d. See subroutine in the insertFoodForm named “save_Click()” for code detail
 - e. Updates Remaining row to help user see how close to their daily goals they are.
4. **Add food to database.** The user can add a new food to the food database found on the “Food Database” worksheet of the workbook.
 - a. Finds the next empty row in the database and inputs the information

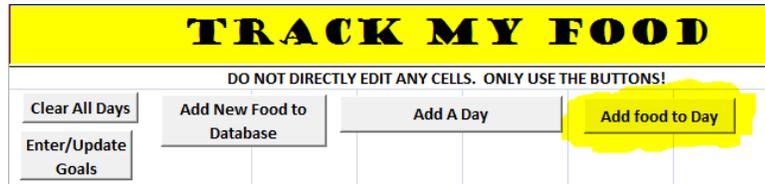
5. **Enter goals.** The user can enter daily goals and then track their progress towards meeting those goals on a day by day basis.
 - a. The purpose of the goals button is to help the user track their daily progress towards reaching their nutrition goals.
 - b. After a food is added, the total of all the foods is added up for that day and subtracted from the values in the “Goals” tab.

Implementation

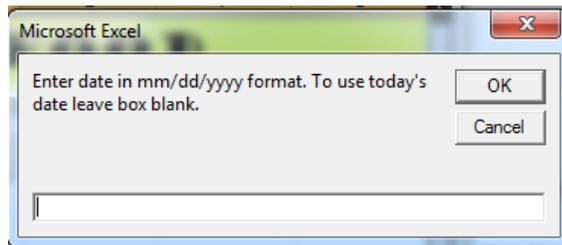
The system has the following capabilities:

Insert Day

- **Purpose.** Allows the user to add a new day to their spreadsheet. This new day can then be used to track food information for the day.
- **Process**
 - User initiates use by pressing the “Add food to Day” button (see below)



- User then enters the date they wish to enter in the text box (see below). If the user does not specify a date, the current date is used.



- The date is inserted along with a “Daily Totals:” row and a “Remaining” row. The remaining row is used to show how much more a user needs to fulfill their daily goals.

	10/6/2011	Calories	Serving Size	Fat	Cholesterol	Sodium	Carbohydrate
Daily Totals:							
Remaining:							

- **Features**
 - **Error Checking.** Two days with the same date cannot be created.
 - **Quick date entry.** The user can use today’s date quickly by simply leaving the input field blank and pressing OK.
 - **Chronological Date Entry.** When inserting a new day, it will find the right place chronologically to insert the data
 - See sub “addNewDay” in Module1 for code details

Clear All Button

- **Purpose.** Clears all the days of the worksheet.
- **Process**
 - User clicks on “Clear All Days” button



-
- All days are cleared on the worksheet.

- **Features**

- See sub “clearAll” in module1 for code details

Add food items to day.

- **Purpose.** Brings up a form through which you can search for and add food to a particular day
- **Process.**

- User clicks on the “Add food to day” button



-
- A form pops up:

-
- A key thing to point out is that this form serves two purposes:
 - Either the user can manually fill out all the boxes regarding the nutrition information.
 - OR the user can use the search function to search for a food in the database. When the search finds a food, it auto-populates all of the nutrition information.
 - The user then clicks the “Add food” button to save the food to the day entered.
 - The daily totals for each category are update.
 - The Remaining column is updated to indicate how much of a particular category the user needs to eat to meet their goal.

- **Features**

- Will validate that the date exists in the spreadsheet already.
- If date does not exist will exit the procedure.

- Search feature allows for user to enter all or part of the food description. Once the food is found the form auto-populates with information from the database.
- See subroutine in the insertFoodForm named “save_Click()” for code detail
- Updates Remaining row to help user see how close to their daily goals they are.

Add New Food to Database

- **Purpose.** Allows the user to enter a new food into the food database (the food database is located on the “Food Database” tab of the worksheet)
- **Process.**
 - User clicks the “Add New Food to Database” button as shown below



- It brings up the add new food form

- User enters the nutrition information and hits the “Save” button.
- This saves the information in the form to the “Food Database” tab (example below)

Description	Calories	Serving Size	Fat	Cholesterol	Sodium	Total Carbohydrate	Protein
Quaker Oats	150	4	2	5	32	4	4
Granola Bars	150	1	2	3	34	67	4
Banana	100	1	2	3	50	15	9

- **Features.**
 - Finds the next empty row in the database and inputs the information

“Enter Goals” button

- **Purpose.** Brings up a form that users can use to update their goals. The information that they enter is saved on the “Goals” worksheet.
- **Process**
 - The user clicks the “Enter/Update Goals” button

TRACK MY FOOD

DO NOT DIRECTLY EDIT ANY CELLS. ONLY USE THE BUTTONS!

Clear All Days

Add New Food to Database

Add A Day

Add food to Day

Enter/Update Goals

- A form to enter goals in will pop up:

The screenshot shows a dialog box titled "UserForm1" with a close button (X) in the top right corner. The main title of the dialog is "Enter Your Goals". Below the title, there are six input fields arranged in two columns and three rows. The first row contains "Calories" and "Protein". The second row contains "Fat" and "Carbohydrate". The third row contains "Cholesterol" and "Sodium". Below these fields is a button labeled "Save Goals".

- The user enters the numbers and then clicks the "Save Goals" button.
- The data is written to the "Goals" tab an example of which is shown below:

DAILY GOALS					
Calories	Fat	Cholesterol	Sodium	Total Carbohydrate	Protein
2000	2	2	2	2	150

- **Features**

- The purpose of the goals button is to help the user track their daily progress towards reaching their nutrition goals.
- After a food is added, the total of all the foods is added up for that day and subtracted from the values in the "Goals" tab.

Learning Difficulties

I learned the following:

- Excel can do just about anything you want it to if you are good with vba.
- On a form, to change the order that focus goes to in regards to textboxes, set the tabOrder property in the textbox to the desired position. For example, if I want a particular textbox to be focused on when the form is brought up I would set the value of its tabOrder property to 1.
- I implemented a complicated method to shift rows down by copying values from one row to another row. It took as an argument the row that you wanted shifted down as well as an integer argument that specified the number of rows to shift down. This of course is unnecessary

as I could have just as easily inserted new rows...this was a classic case of over-thinking the problem.

- Getting the data to be in right format (String vs Double) etc was difficult sometimes but I was able to work through that.
- Merged cells can really mess up selecting columns. If a column contains a merged cell and you select it in vba code, it will also select all the other columns which was messing me up pretty badly. I unmerged the cells and everything was ok. This was the last big problem I had to figure out.
- Overall, I was able to implement all features from my proposal and learned a great deal. This has been a great class—thanks for everything.