

# Cooking to Goal Analysis

## Table of Contents:

[Client](#)  
[Project](#)  
[Program Features](#)  
[Program Usability](#)  
[Project Prototype](#)  
[Project Technical Specifications](#)  
[Our Comments & Concerns](#)  
[Original Document](#)

## Client:

- grocery delivery company

[top](#)





## Project:






- a meal planning and purchasing tool for grocery clients

[top](#)

## Program Features:




(legend in comment)

-  create a week's worth of menus that are guaranteed to meet their dietary goals
  - one week is default planning period
  - user enters # of meals of each type for planning period
  - user can adjust # of persons for any / all meals
  - a checklist of menu suggestions is presented for the user to select for each meal type
    - may include oven-ready (TV dinner) meals
    - may include cook from scratch meals the user prepares
    - system must come pre-loaded with a reasonable number of choices for each type of meal
-  From the menu, a weekly shopping list of items to buy is created
-  a setup screen to establish the default planning assumptions for a client
  - default number of persons served at a meal & their age, gender, and nutritional goals.
  - basis for calculating the quantity of ingredients to be purchased for any planning period
  - must meet their nutritional goals
-  Recipe entry screen

- user enters their own meal choices
- stored in the system as meal options
  - must accept, store, classify, and display customer-entered recipes as meal choices
-  ability to print 3x5 recipe cards
-  Quantity Calculator (available once weekly menu is fixed)
  - adjusts portion size to meet the guaranteed weekly nutritional limits
  - assumes what's bought is eaten
    - assume meals are fully consumed
    - reduce the remaining weekly total
    - scale the amounts of the other meals to keep within the limit
    - recipe display and the shopping lists derived from them must scalable up and down to meet a nutritional target
-  a way to enter and view statistics for tracking progress
  - e.g. weights, measurements, blood sugar readings, etc.
  - track progress vs. time (line graph?)
-  product & recipe rating system
  - use ratings to adjust future choices presented to a customer
  - remove a choice for some period of time if it has recently been selected
  - add selections that depend on the season.
-  (“can do without” feature example)

[top](#)

### Program Usability:

- users for the system have limited familiarity with computers:
  - can open and close an application
  - can use the keyboard and mouse
  - can (maybe) remember to save a file
  -  protect them from their own mistakes and have lots of help tips that guide them without cluttering the interface
- User manual not necessary
  - A Quick Start Guide would be helpful -- one sheet, front and back.
-  graphical view of how user selections are affecting the weekly limit as they make selections.
  - is in addition to tables and lists of the detailed quantities
-  Detailed ingredients or nutritional panels in the appropriate screens.
- think of your users as being used to seeing restaurant menus, recipes, shopping lists and nutritional panels on foods -- an interface that has the same flavor would be a comfort

[top](#)

### **Project Prototype:**

- as quickly as possible, even if it means limiting it to some simple cases
  - e.g. breakfasts, one-pot dinners, meal-ready entrees.
- interested in getting feedback from users on its ease of use than comprehensive calculation capabilities.
- want to see designs for the three most important printed reports:
  - a shopping list
  - a scaled recipe
  - the weekly menu

[top](#)

### **Project Technical Specifications:**

- use only Java SE 6 classes
  - ones that you provide with source code and test cases
- Code documentation is very important due to follow-on project
- Follow-on will make the application web based so...
  - base logic is only in classes that do not depend on the interface implementation
- system can load and save data from simple files
  - the files need to be structured in a logical way that can be converted to database tables in a follow-on project

[top](#)

### **Our Comments & Concerns:**

- Do we need to / How are we going to, get nutritional information for every product in the program? (- Drew)
- Where will the initial pre-loaded recipes come from? (-Ryan)
- Our customer needs to define the expected nutritional limits and goals. (-Ryan)
- (insert your comments / concerns here)

[top](#)



## Original Document

- (portions of document already analyzed are grayed out)

[top](#)

---

## Cooking to Goal

Our grocery delivery company wants to offer a meal planning and purchasing tool that our clients can use to create a weeks worth of menus that are guaranteed to meet their dietary goals. From the menu, a weekly shopping list of items to buy is created that we hope the customer will forward to us to supply, thus increasing our volume of sales. Since our delivery routes are in place, any incremental sales to existing customers are essentially free and any new customers are added to existing routes at only a small cost due to the time of the extra stop during a route. Our average mark-up on food purchased is about 50%, so more volume has a significant impact to our bottom line. We also hope to promote this service through health clubs, nutritionists, etc. to increase our customer base by 50% as well.

[top](#)

There needs to be as a setup screen to establish the default planning assumptions for a client: things like number of persons served at a meal, their age, gender, and nutritional goals. This is the basis for calculating the quantity of ingredients to be purchased for any planning period -- we guarantee that what is purchased will meet their nutritional goals. It would be nice to also offer the client a way to enter and view for tracking their progress: weights, measurements, blood sugar readings, etc. so that they can see their progress vs. time with the program.

[top](#)

After a client is set up, we need a way for them to easily indicate the number of meals of each type to be served during the planning period (one week by default), and a way to adjust the number of persons served up or down for any or all of the meals. Next we need to present a checklist of menu suggestions for each type of meal that the customer can select. Note that the selections can include oven-ready entrees (i.e. TV dinners) from us or cook-from-scratch meals they prepare.

[top](#)

The system must come pre-loaded with a reasonable number of choices for each type of meal, but a vital feature of our system is the ability for the customer to be able to enter additional choices of their own through a recipe entry screen. Once entered, these choices remain part of the selection options. This implies that the system must be able to accept, store, classify, and display recipes for customer-entered choices. It would also be nice to be able to print recipes onto 3"x5" cards.

[top](#)

I can see some value to offering a rating system for products and recipes selected and consumed, and then using the ratings to adjust future choices presented to a customer during the selection entry. (We might also want to remove a choice for some period of time if it has recently been selected, or add selections that depend on the season.)

[top](#)

Finally the heart of the system is the quantity calculator once the weekly menu has been fixed. In short, this adjusts portion size to meet the guaranteed weekly nutritional limits. It works on the assumption that what's bought is eaten, so if the customer selects two oven-ready entrees for one meal, assume they are fully consumed, reduce the remaining weekly total, and scale the amounts of the other meals to keep within the limit. This means that the recipe display and the shopping lists derived from them must scalable up and down to meet a nutritional target.

[top](#)

The users for the system have limited familiarity with computers, not a deep understanding. They can open and close an application, use the keyboard and mouse and (maybe) remember to save a file, but otherwise we need to protect them from their own mistakes and have lots of help tips that guide them without cluttering the interface. A detailed user guide isn't necessary, they don't have time to read it anyway. A Quick Start Guide would be helpful if you can keep it to one sheet, front and back.

[top](#)

Users will need to see a graphical view of how their selections are affecting the weekly limit as they make selections. This is in addition to tables and lists of the detailed quantities. Detailed ingredients or nutritional panels need to be available in the appropriate screens. You can think of your users as being used to seeing restaurant menus, recipes, shopping lists and nutritional panels on foods -- an interface that has the same flavor would be a comfort.

[top](#)

We'd like an initial version of the application as quickly as possible, even if it means limiting it to some simple cases e.g. breakfasts, one-pot dinners, meal-ready entrees. For that early delivery, I'm more interested in getting feedback from users on its ease of use than I am comprehensive calculation capabilities. We'll also want to see your designs for the three most important printed reports -- a shopping list, a scaled recipe and the weekly menu.

[top](#)

We want the initial project to use only Java SE 6 classes and ones that you provide with source code and test cases. This is a work for hire and time is of the essence. Code documentation is very important because we will let another contract for a follow-on project as soon as we're sure your project delivery is working. The follow-on project will also make the application web-

based, so we need to be sure that the base logic is only in classes that do not depend on the interface implementation. Your system can load and save data from simple files initially, but the files need to be structured in a logical way that can be converted to database tables in a follow-on project.

[top](#)

We look forward to seeing your detailed proposal for this project.

[top](#)