Dietary Menu Planning Using Constraint Logic Programming

GRANADA, CEDRIC G., Ateneo de Davao University
PATENIO, CAMILLE SHARMAINE A., Ateneo de Davao University

One of the problems people face today is to finding or knowing the appropriate meal that contains the right amount of calories that their bodies need. This study is about generating or suggesting a daily menu based on the user's physical information specifically their Recommended Calorie Intake (RCI), their allergies and their medical history. The study will be using the Constraint Logic Programming (CLP) approach. CLP is a type of logic programming that combines constraint solving and logic programming specification power. CLP is also used for solving constraint satisfaction problems (CSP), design, assigning, planning, scheduling and time-tabling problems. The researchers considered the user's physical information, allergies, medical history and the diet or the user's preference as the data to be used. The researchers also considered labeling (depth-first search) and optimization (branch and bound method) in finding the optimal solution to the problem.

General Terms: Menu-planning, recommended calorie intake, constraint logic programming, constraint satisfaction problem

Additional Key Words and Phrases: depth-first search, branch and bound

1. INTRODUCTION

1.1 Background of the study

One of the problems people face today is to finding or knowing the appropriate meal that contains the right amount of calories that their bodies need. People nowadays do not eat the right kind of food, which is why they do not have enough nutrients in their bodies and some have inappropriate weight in correspondence to their height that can lead them to have a certain kind of disease.

This study is about generating or suggesting a menu based on the user's physical information specifically their Recommended Calories Intake (RCI), medical history, allergies, and the user's preferences and this would help solve the problems mentioned.

There are proponents have found some of the approaches to menu planning. The computer-based menu planning which modeled the problem-specific knowledge by mapping data information from different sources, including food composition data, and dietary reference values, is one of the approaches to menu planning. Another approach is the expert system approach which defines, creates and manages meals according to user preferences regarding a diet plan, preferred foods, preferred meal preparation options, and meal times. Another approach is the web-based decision support system and also the multi-dimensional 0/1 knapsack problem using genetic algorithm.

The proponents are going to approach the study with the use of Constraint Logic Programming. CLP tries to provide a solution to a problem by combining constraint solving algorithms with the Logic Programming specification power. CLP solves constraint satisfaction problem and are often used in design, planning and scheduling problems.

The proponents decided to pursue dietary menu planning using CLP since there has been no study conducted using this approach. With this reason, the proponents hope that it would produce a more functional and more flexible dietary menu planning system.

1.2 Problem Statement

The study sought to suggest or generate a dietary menu that is based on the user's physical information specifically their recommended calorie intake, allergies, medical history and user's preferences using the Constraint Logic Programming (CLP). Moreover, the study wishes to answer the following questions:

The specific problems of the study are as follows:

(1) What are the data to be used in solving dietary menu planning?
(2) What meals can be created given the user's physical information?
(3) What are the constraints of the dietary menu planning?
(4) How can the dietary menu planning be solved using CLP?

1.3 Objectives

The study intended to suggest or generate a dietary menu using CLP. Specifically, the study wishes to accomplish the following objectives:

(1) To know what data to be used in solving dietary menu planning
(2) To know the meals that can be created given the user's physical information.
(3) Find out what are the constraints of the dietary menu planning.
(4) Explain how the dietary menu planning can be solved using CLP.

1.4 Significance of the Study

The significance of the study is that it helps people to be more aware of the food that they are eating and that it would also to maintain their calorie intake per day. It is significant because our society today is very health conscious; some people calculate the calories of the food that they are going to eat to maintain their weight to be fit. This system would help people to be fit and this system would be very efficient to those who calculate the food calories. This kind of system can also help dieticians minimize their workload. Dietary menu plan can help people with diseases such as malnutrition, which can help them gain or maintain their weight. Since there have been no study in dietary menu planning using the CLP approach, it would contribute significantly to the still growing area of designing and assigning of the CLP.

1.5 Scope and Limitations

The study of dietary menu planning would not only cover the user's physical information as its basis, it would also consider the user's medical history, allergies and preferences. The dietary menu would consist of 5 meals; 3 heavy meals and 2 light meals. The study would use the food composition data, which is a Filipino cuisine based, provided by the nutritionist the researchers have consulted. For the medical history, the researchers only considered the following health risks: diabetes, cancer, gallbladder disease, hyperlipidemia, atherosclerosis, hypertension, stroke, heart disease and

osteoarthritis. As for the allergies, the proponents considered the most common allergies of the Filipinos which are the; egg, chicken, seafood and eggplant. The study will be implemented using the Constraint Logic Programming approach.

2. REVIEW OR RELATED LITERATURE

2.1 Dietary Menu Planning

A dietary menu plan would generate a menu based on the user's preference, physical information, medical history and allergies. The purpose of dietary menu planning is to meet individual food preferences and dietary guidelines of the user. This kind of system can also help dieticians minimize their workload. Dietary menu plan can help people with malnutrition, which can help them gain weight.

2.2 Approaches to Dietary Menu Planning

2.2.1 Computer-based Approach

The study of [Seljak, 2009], is an example of dietary menu planning using the computer-based approach. The study used the food data composition of the EuroFIR platform. Aside from the food database, the system also requires the following data and information: personal data, dietary references, meal formats and cuisine rules. Dietary references are standard amounts of each nutrient needed and this is necessary so that it can maintain good health. Cuisine rules are important because every nation, ethnic group and even individual has its own food. The formulation used in this system is the multidimensional knapsack problem, a combinatorial optimization problem which is widely used. The multidimensional knapsack problem finds the most valuable combination of foods that would fit in a "knapsack" of fixed volumes given the different values and volumes of the food. The values that are defined are: food quality, cost, comprising taste, consistency, color, temperature, shape and method of preparation. The volume, on the other hand is defined by dietary recommendations and guidelines. The algorithm used to solve the multidimensional knapsack problem is the evolutionary computation technique, which is a subfield of artificial intelligence that involves numerical and combinatorial optimization problem, to solve the menu-planning.

2.2.2 Expert System Approach

[Schwarzberg et al.,2009] used Expert System in creating their system which define, create and manage meals according to user preferences regarding a diet plan, preferred foods, preferred meal preparation options, and meal times. The system uses information about an individual's diet preferences and related goals, to develop a personalized plan including a complete meal plan and to generate a meal suggestion messages for each meal which is then transmitted to users' portable devices at their preferred meal times. If a user accepts a meal suggestion, nutritional information regarding the accepted meal is tracked and considered in determining the user's progress toward a goal. If a user rejects a meal suggestion, the system generates a meal substitution suggestion.

Another study that used the expert system approach is the study of [Balazs, G., Istvan, V. et al., 2006]. By giving questionnaires to people, the researchers were able to gather data for the personalized requirements which are: age, sex, BMI, type of work, allergies, family diseases and medical history. The study also has nutritional constraints,

harmony(variety, contrast, color, appeal), and scheduling(meal by meal basis, daily basis, or weekly basis) in addition to the personalized requirements. This study also used the EuroFIR food composition database. The assessment of the generated menu plan was based on: penalty function which assesses nutritional content, domain ontology which describes dietician knowledge and rules to avoid non-matching meal combinations. The system uses the Evolutionary Divide and Conquer method and the Genetic Algorithm for near-optimum search to solve the menu-planning problem.

2.2.3 Web-based Decision Support System Approach

The study of [Hsu, C., Huang, L., Chen, T., et al., 2010] used Web-based Decision Support System for their dietary analysis. The study used and modified the Composition of Foods used in Taiwan databases and menu-based food composition database established by Hu et al. Personal data, body weight and dietary management, dietary records and intake, and personalized dietary recommendations and nutrition information are needed for creating the dietary recommendations. Before the menu is generated, first the user's information will be inputted, then it will search the menu which is appropriate to the given information of the users, next is the selection of the user's preferences and then the system calculates the total energy per meal, breakfast, lunch and dinner, and compares it with the recommended daily nutrient quantity.

2.2.4 Kashima, T., Matusmoto, S. and Ishii, H.

[Kashima et al. 2009] developed a menu planning system based on multi-dimensional 0/1 knapsack problem and is solved using the genetic algorithm. The system used the standard tables of food composition in Japan. The user's physical information (gender, age, height, weight) is an important input so that the menu will be generated. Along with the user's physical information, user's preferences are also needed. The menu planning problem is formulated as:

Maximize

$$\sum_{i=1}^{n} c_i x_i, \qquad (1)$$

Subject to

$$\sum_{j=1}^{m} a_{ij}^{l} x_i \leq b_j, \qquad (2)$$

$$\sum_{k=1}^{l} a_{ik}^{h} x_i \geq g_j, \qquad (3)$$

$$x_i \in \{0,1\}, i = 1,2,\cdots,n,$$
$$B = \{b_j | 1,2,\cdots,m\}, G = \{g_k | k = 1,2,\cdots,l\},$$

Figure 1: Formula for the menu planning.

where i is dish number for the total number of dishes n, j is negative and k is positive nutritional component number. c_i is satisfaction value of i, a^l_{ij} is intake of j in i such as salt, calorie, and fat, and b_i is limit value of recommended daily intake of j. Similarly, a^h_{ik} is intake of k such as vitamin, calcium, and fiber.

2.3 Meal Patterns

According to [Narciso, M.H, 2005] meal patterns deal with what makes up a meal. Meal patterns is also related to the foods that include in the meals, the person in charge of the meals, the way the meals are prepared and served and the users of the meal and the dining environment. Meal pattern is also called dietary patterns, food habits, food ways or food culture. Meal pattern has two components: the material which is described as food availability and accessibility factors, and the cultural component which consist of food related beliefs and attitude of a particular group.

The Philippines' geography, multi-cultural history and culture are the reasons for the Filipino meal patterns. Filipino meal pattern is somewhat similar to the neighboring countries since they have relatively common ancestry and has the same natural food resources.

Rice products made up the largest part of the daily Filipino diet followed by the vegetables then the fish. One of the Filipino meal patterns is that a meal should have rice, otherwise, it is not considered as a meal. There are three Filipino meals in a day; *agahan* or *almusal* (breakfast), *tanghalian* (lunch) and *hapunan*(dinner). The typical components of Filipino meals are shown in Figure 2.

Agahan	*Tanghalian or Hapunan*
Fish, meat or egg	Fish or meat
Bread or rice	Vegetables
Fruit	Rice
Coffee with milk and sugar	Fruit or Dessert

Figure 2: Components of Filipino Meals

Aside from the meals mentioned before, Filipinos have two extra smaller meals which is the snacks or locally called *merienda.* The *merienda* usually consists of sweets, pastries and *kakanin*.

2.4 Classical Large Scale Combinatorial Optimization (LSCO) Structure

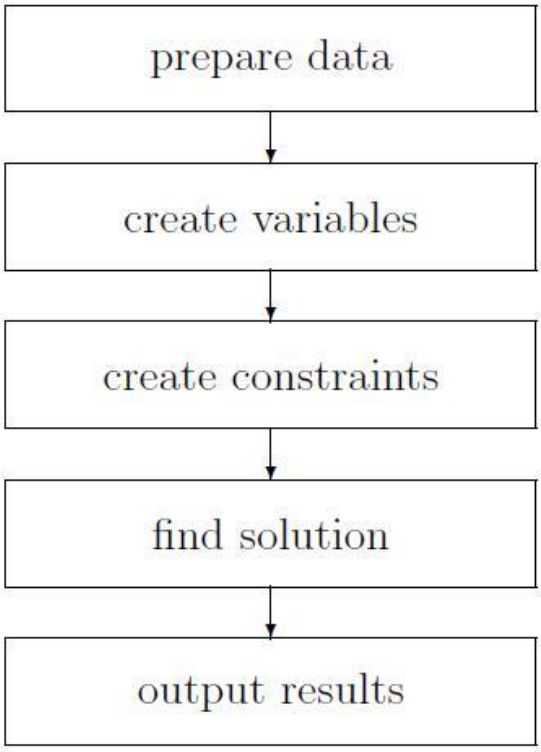

Figure 3: LSCO Application Structure

[Simonis, H. 1992] presented in figure 1 a flow analysis that the RiskWise follows. The data are read and prepare the data structures that are required for the problem solver. Variables are created that will be used in the solver. These variables will simply be placed in slots already provided in the data structure. Then we create constraints between these variables. After that, we find solution with all constraints stated. Finally we have to take the solution and produce the output results.

3. METHODOLOGY

3.1 Conceptual Framework

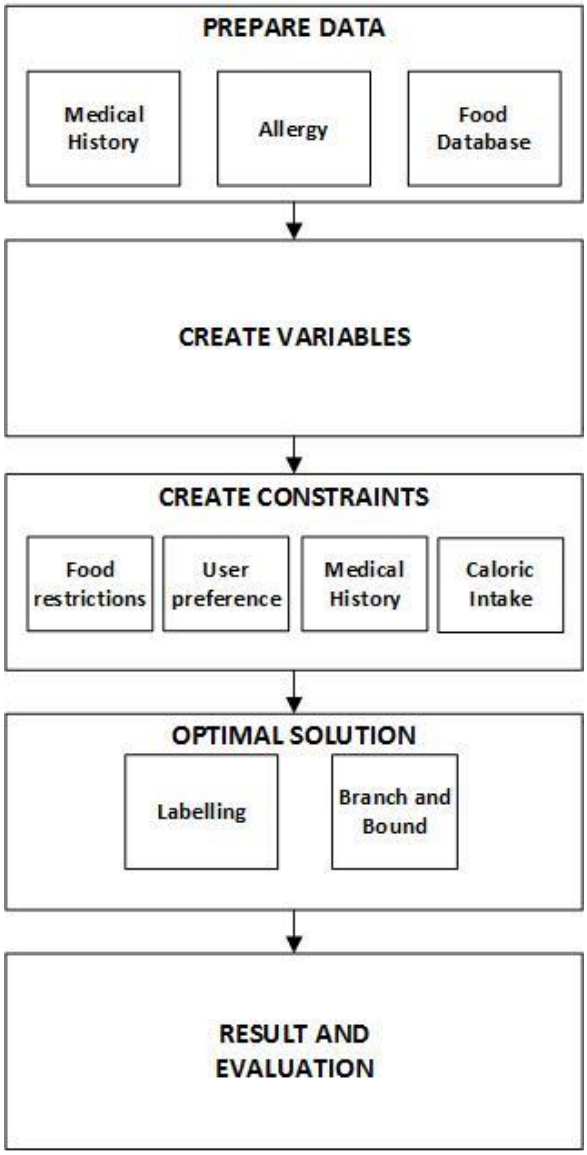

Figure 4: Diagram to create a meal suggestion using CLP

3.2 Methodology

In this study of suggesting or creating a meal based on ingredients using Constraint Logic Programming, the phases are as follows:

1. Prepare data
   1.1 Food Database
   1.2 Medical History
   1.3 Allergy
2. Create variables
3. Create constraints from variables

3.1 Preferences
3.2 Illnesses
3.3 Allergy
3.4 Caloric Intake
4. Find optimal solution
4.1 Labeling
4.2 Optimization
4.2.1 Branch and Bound Method
5. Result and Evaluation

3.2.1 Phase 1: Prepare Data

The proponents gathered the following data: medical history, allergy and the food database. For the physical information, the proponents gathered the BMI levels and the type of work so to be able to come up with the Recommended Calorie Intake. The formula for calculating the Recommended Calorie Intake and the food database were gathered from the nutritionist that the researchers consulted since the proponents did not find a legit Filipino food database with nutritional facts online and also according to the nutritionist, the Food and Nutrition Research Institute (FNRI) was not updated that was why the proponents did not have a good food database. However, the nutritionist gave us a booklet which consists of food exchange lists for meal planning and that was what the researchers used as food database. Here is an example of the food database:

```
food(158,"Puto, Bumbong",rice_starch_b,"2 pcs",100).
food(179,"Jelly Roll",rice_starch_b,"1 slice",100).
food(200,"Tenderloin, Beef",meat,"3 slice",123).
food(210,"Chicken Leg",meat,"3 slice",123).
food(233,"Tilapya",meat,"4 pcs",82).
food(239,"Alimango, Laman",meat,"1 cup",164).
food(257,"Bangus, Tinapa",meat,"1 pc",164).
food(274,"Ham",meat,"1 slice",122).
food(275,"Bacon",meat,"4 pc",180).
food(284,"Pastillas, Durian",sugar,"5 pcs",100).
food(285,"Pastillas, Gatas",sugar,"5 pcs",100).
```

Figure 5: Sample of the food database.

The parameters of the “food” are the id number, followed by the name of the food, then the category, then the measurement of the food and lastly, the calories of the food. The proponents also collected the food restriction for each illness in the medical history and for the allergies. The proponents searched the food restrictions for the medical history then asked the nutritionist to verify the researched data. The food restrictions were saved in another file, an example of the food restriction file is shown in figure 6.

```
ingredients(1, "white flour", [171,172,173,,174,175,176,177,
        178,179,180,181,182,183,184,185,186,194,196]).
ingredients(2, "bacon", [275]).
ingredients(3, "beans", [252]).
ingredients(4, "processed foods",[253,254,255,256,257,258,259,270,271,272,273,274,275]).
ingredients(5, "fatty meats", [236,237,238,239,240,241,263]).
```

Figure 6: Food restrictions

```
illness("diabetes", [1, 2]).
illness("cancer", [1,3,4,7]).
illness("gallbladder disease",[5]).
illness("hyperlipidemia", [5]).
illness("atherosclerosis", [8]).
illness("hypertension", [8]).
illness("stroke", [3,5,9]).
illness("heart disease", [5,8]).
illness("osteoarthritis", [4,6,7]).
```

Figure 7: Medical history restriction.

The parameters of the "ingredients" are the id number, the name of the food restriction and a list of id numbers, of the "food" input file, that are included in the food that are to be avoided. For the "illness" input file, which is for the medical history, the parameters are the illness name and a list of the "ingredients" id numbers which the illness are restricted to. The illnesses included in the medical history are the most common illnesses in the country and it is also verified by the nutritionist.

3.2.2 Phase 2: Create Variables

From the data that were gathered in the previous phase, variables would be generated. Variables are necessary in order to create constraints.

3.2.3 Phase 3: Create Constraints from Variables

Constraints are generated from the variables that are available. The constraints would contain the restrictions or rules in generating the dietary menu. The proponents gathered Filipino meal pattern, food restriction for a specific health risk and allergies and user's preferences for the constraints. Another constraint that the proponents included was the plate method which means that every heavy meal should contain starch, vegetable, meat and fruit.

3.2.4 Phase 4: Find Optimal Solution

Given the constraints, the optimal solution was generated using the labeling and branch and bound method. The optimal solution has priorities, the one with the higher priority are the ones included in the user's preference. The menu that was generated was optimized by the user's preferences and the recommended calorie intake.

$$Cost = \begin{cases} \sum_{j=1}^{N}\sum_{k=1}^{P} X_{jk} \cdot C + \sum_{i=1}^{M} C_i & if \quad C_{min} \leq \sum_{i=1}^{M} C_i \leq C_{max} \\ 0 & otherwise \end{cases}$$

Figure 8: Formula for the Objective Function

In figure 8, the formula for the objective function is shown, where N is the preference number of the user, P is the preference number of food, C is equal to 10,000 which is constant and M is the calorie of the food. C_{min} is calculated as RCI * 9/10 while the C_{max} = RCI.

3.2.5 Phase 5: Results and Evaluation

The proponents consulted a nutritionist/dietician for the evaluation of the output. The evaluation part was very crucial because that would determine if the study was successful or not.

4. THEORETICAL BACKGROUND

4.1 Logic Programming

According to [Simonis, H. 1992], the basic idea of logic programming is the separation of logic and control. Logic programming is concerned in the correct statement of the problem and the efficiency of the program.

Logic programming is applied in the following areas:

- Relational Databases
- Natural Language Interfaces
- Expert Sytems
- Symbolic Equation Solving
- Planning
- Prototyping
- Simulation
- Programming Language Implementation

4.2 Constraint Logic Programming

[Fruhwirth, et al. 1992] described Constraint Logic Promgramming as an attempt to overcome the difficulties of logic programming by enhancing a Prolog-like language with constraint solving mechanisms. It combines advanced techniques from Operations Research and finite mathematics. It is also particularly used for solving constraint satisfaction problems. CLP has been used in many different application areas such as engineering, planning, scheduling, layout, fault diagnosis and constrained search problems. Also, CLP is best suited in the combinatorial problems such as in Operations Research, hardware design, financial decision making, or even in Biology (DNA sequencing).

4.3 Branch and Bound

Suppose that we are interested in finding a solution with the minimal value of the cost function. During the search, one maintains the currently best value of the cost function in a variable bound. Each time a solution with a smaller cost is found, this value is recorded in bound. This method of the appropriate modification of the backtracking search is called the Brach and bound [Apt, K & Wallace, M. 2006].

5. RESULTS AND DISCUSSION

5.1 Discussion of Codes

```
build_foodGroup([], []) :- !.
build_foodGroup([FoodListH|FoodListT], [FoodGroupH|FoodGroupT]) :-
    arg(3, FoodListH, FoodGroupH),
    build_foodGroup(FoodListT, FoodGroupT).
```

This builds a list of all the third arguments of the food file which are vegetables, fruits, milk, starch, meat and sugar.

```
get_min_max_each([], _, []) :- !.
get_min_max_each([food(ID, _, Category, _, _, _)|FoodT], GetCategory, [ID|Rest]) :-
    Category = GetCategory,
    get_min_max_each(FoodT, GetCategory, Rest).
get_min_max_each([food(_, _, _, Category, _, _)|FoodT], GetCategory, List) :-
    Category \= GetCategory,
    get min max each(FoodT, GetCategory, List).
```

This method checks the food file then returns a nested list of the minimum and maximum ids of each food group or category.

```
create_meals([Veg, Fru, Mil, RicA, RicB, Mea, Sug], Meal, MealList) :-
    Meal = [bFast(V1, F1, M, R1, M1), amSnack(Su1, St1),
            lunch(V2, F2, R2, M2), pmSnack(Su2, St2),
            dinner(V3, F3, R3, M3)],
    V1 #:: Veg,
    V2 #:: Veg,
    V3 #:: Veg,
    F1 #:: Fru,
    F2 #:: Fru,
    F3 #:: Fru,
    M #:: Mil,
    R1 #:: RicA,
    R2 #:: RicA,
    R3 #:: RicA,
    M1 #:: Mea,
    M2 #:: Mea,
    M3 #:: Mea,
    St1 #:: RicB,
    St2 #:: RicB,
    ( Sug = [] ->
        Su1 = 0,
        Su2 = 0,
    MealList = [V1, F1, M, R1, M1, St1,
            V2, F2, R2, M2, St2, V3, F3, R3, M3]
    ;
    Su1 #:: Sug,
    Su2 #:: Sug,
    MealList = [V1, F1, M, R1, M1, Su1, St1,
            V2, F2, R2, M2, Su2, St2, V3, F3, R3, M3]
    ).
```

V1 #:: Veg, and so on, means that V1 will have any value in the list Veg which is built in the get_min_max_each method. The Sug = [] means that if the sugar list is empty, which is a result of the constraints, then all the sugar in the meal would be 0.

```
remove_food_res(Res, Ing, NewRes) :-
    build_remove(Res, Ing, Temp1),
    extend_remove(Temp1, Temp2),
    remove_duplicate(Temp2, NewRes).

remove_food_mh(Mh, Ill, Ing, NewMh) :-
    all_ill(Mh, Ill, Temp1),
    extend_remove(Temp1, Temp2),
    remove_duplicate(Temp2, Temp3),
    quick_sort2(Temp3, Res),
    build_remove(Res, Ing, Temp4),
    extend_remove(Temp4, Temp5),
    remove_duplicate(Temp5, NewMh).

combine_remove(IDList, Res, Mh, FinalList) :-
    append(Res, Mh, Temp1),
    lists:flatten(Temp1, Temp2),
    remove_duplicate(Temp2, Temp3),
    quick_sort2(Temp3, ResMh),
    remove_food(IDList, ResMh, FinalList).
```

This would append the two lists, the food restrictions (allergies) and the medical history, and creates a new list, then flattens and removes all the duplicates of the new list.

```
remove_food(List, [], List) :- !.
remove_food([IDFoodH|IDFoodT], [ResH|ResT], NewList) :-
    IDFoodH = ResH,
    remove_food(IDFoodT, ResT, NewList).
remove_food([IDFoodH|IDFoodT], [ResH|ResT], [IDFoodH|NewListT]) :-
    IDFoodH \= ResH,
    remove food(IDFoodT, [ResH|ResT], NewListT).
```

This would eliminate the foods that are included in the list of restricted food. This is important so that the menu that would be created would not include the food that the users should not eat or are restricted to.

```
new_food(_, [], _) :- !.
new_food([food(ID, Name, Category, Weight, Measurement, Calorie)|FoodT],
         [ID|Rest],
         [food(ID, Name, Category, Weight, Measurement, Calorie)|NewFoodT]) :-
    new_food(FoodT, Rest, NewFoodT).
new_food([food(ID, _, _, _, _, _)|FoodT], [First|Rest], NewFood) :-
    ID \= First,
    new_food(FoodT, [First|Rest], NewFood).
```

After passing the list of all food that should be removed, this trims down the food file to a new food database which does not include the removed foods.

```
to_calorie([], _, [], []) :- !.
to_calorie([MealH|MealT], FoodFile, [CalorieH|CalorieT], [SumH|SumT]) :-
    functor(MealH, _, Arity),
    NewArity is Arity + 1,
    term_arguments(MealH, 1, NewArity, List),
    to_calorie_each(List, FoodFile, CalorieH),
    ic_global:sumlist(CalorieH, SumH),
    to_calorie(MealT, FoodFile, CalorieT, SumT).
```

This gets the food calorie of the food id in the lists, which are the meals and then sums it all up. This would set the calorie of the menu.

```
to_pref([], _, []) :- !.
to_pref([MealH|MealT], PrefFile, [PrefH|PrefT]) :-
    functor(MealH, _, Arity),
    NewArity is Arity + 1,
    term_arguments(MealH, 1, NewArity, List),
    to_pref_each(List, PrefFile, PrefH),
    to_pref(MealT, PrefFile, PrefT).\
```

This function is the same as the to_calorie, but instead of calorie, it gets the preferences. Also in this function, there is no summed list; this means this only gets the id list of the preference. The output of this function will be used for the Cost.

```
to_pref_each([], _, []) :- !.
to_pref_each([EaH|EaT], Pref, [EaPH|EaPT]) :-
    (EaH = 0 ->
        EaPH = 0,
        to_pref_each(EaT, Pref, EaPT)
    ;
        get_pref(Pref, EaH, EaPH),
        to_pref_each(EaT, Pref, EaPT)).

get_pref(Pref, ID, CatID) :-
    once(member(preferences(ID, _, CatID), Pref)).

check_pref_all([BFast, Am, Lun, Pm, Din], Pref, PrefTot) :-
    split(Pref, PrefLB, PrefLLD, PrefS),
    check_pref(BFast, PrefLB, BTemp),
    check_pref(Lun, PrefLLD, LTemp),
    check_pref(Din, PrefLLD, DTemp),
    check_pref(Am, PrefS, ATemp),
    check_pref(Pm, PrefS, PTemp),
    lists:flatten([BTemp, LTemp, DTemp, ATemp, PTemp], Temp),
    ic_global:sumlist(Temp, PrefTot).
```

This checks all preferences from the chosen meal's preferences then sums it up. This is now the weight of the user's preferences and the preferences combined. The sum will be used for the Cost.

```
get_cal_range(CalNeeded, Min) :-
    Temp is CalNeeded * 9,
    Min is Temp / 10.

get_cost(MinCal, CalNeeded, CalTotal, PrefOfMealWCost, Cost) :-
    MinCal =< CalTotal,
    CalTotal =< CalNeeded,
    Temp is PrefOfMealWCost * 10000,
    Cost is Temp + CalTotal.
get_cost(MinCal, _CalNeeded, CalTotal, _PrefOfMealWCost, Cost) :-
    MinCal > CalTotal,
    Cost is 0.
get_cost(_MinCal, CalNeeded, CalTotal, _PrefOfMealWCost, Cost) :-
    CalTotal > CalNeeded,
    Cost is 0.
```

The code above is the formula for the cost of the objective function which can be seen in figure 8.

5.2 User input

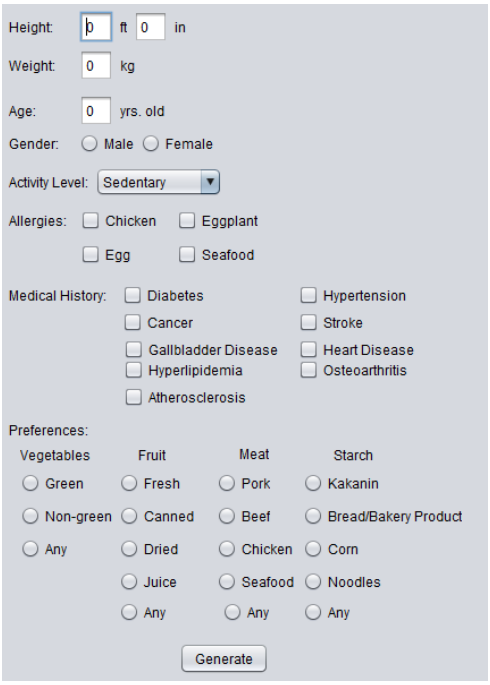

Figure 9: GUI of the system

Before the menu will be generated, the user must input their height, weight, gender, age and level of activity so that the Recommended Calorie Intake will be calculated. The user will also input their allergies, medical history and as well as the food preference. Eclipse would receive the following inputs from the user; RCI, list of allergies which is converted to a list of the id number of the food, the list of medical history, and the list of the user preference which has the same format as the food allergies. Shown in the figure 10 is an example of the input that eclipse should receive.

```
(1500, [10,11,12,13],["diabetes", "cancer"], [1,2,2,4])
```

Figure 10: Sample input

1500 is the RCI; the list [10, 11, 12, 13] corresponds to the food id of the allergies which are eggplant, seafood, chicken and egg. After the allergies is the medical history, which is a list of string. And the list [1, 2, 2, 4] parallels to the user's food preference. The first element of the food preference list is the vegetable, green is 1, non-green is 2 and any is 0. The second element is the fruit, fresh is 1, canned is 2, dried is 3, juice is 4 and any is 0. The third element is meat, pork is 1, beef is 2, chicken is 3, seafood is 4 and any is 0. Last element is the starch which is for the snacks, kakanin is 1, bread is 2, corn is 3, noodles is 4 and any is 0.

5.3 Results

The proponents were not able to get the most optimized menu because it takes a lot of time to get it. The program takes too long to produce the optimized menu because it would match all food to create the menu and the proponents have 292 food items in the food database. However, the proponents were able to get this result from a 30-hour continuous run of the program: cost = -82478, menu = [["Carrots", "Pineapple, crushed", "Fresh carabao's

milk", "Cassava", "Beef Plate"], ["Maraschino cherries", "Pasensiya"], ["Acelgas", "Dates, pitted", "Rice, cooked", "Hito"], ["Bukayo", "Tupig"], ["Bamboo shoot", "Sweetened Mango", "Ube", "Beef Brisket"]], amount = [["1 cup", "1 cup", "1 cup", "1 pc", "3 slices"], ["10", "22 pcs"], ["1 cup", "6 pcs", "1 cup", "4 pcs"], ["10 pc", "1/2 pc"], ["1 cup", "1 glass", "2 pc", "3 slices"]]. The inputs of the result above were: [2500, [], [], [0,0,0,0]], which means that the RCI was 2500, no allergies and medical history and the food preferences were any. Since the program takes so long to generate the optimized output, the proponents were not able to sync the front-end and the back-end. To test the program, the proponents manually input the inputs.

The proponents found a solution, which was to put a time-out in the query, so that it would still output a menu. The query with timeout is shown on figure 11.

```
branch_and_bound:bb_min(generate_meal(['food.ecl', 'ingredients.ecl',
'illnesses.ecl', 'preferences.ecl'], RCI, [RES], [MH], [PREF], X, Y, Z),
X, bb_options(continue, -1.0Inf, 0, 1.0, 1.0, 60, _, _, _, _)).
```

Figure 11: Query for generating the menu

The proponents found out that using time-out, the program would still create an optimized menu but within the time limit the proponents set. After many tests, the proponents got different menu but with the same time limit and inputs. That was because every run of the program, it would match random food items to create a menu. With these findings, the proponents expected that the program would produce the same menu with the same inputs if the run will completely be finished.

The researchers tested the program with different inputs but with the same time-out. The time that the proponents used to test the program was 30 seconds.

Input	**Output**
1500,[],[],[0,0,0,0]	X = -1499
	Y = [["Baguio Beans", "Cashew", "Lite, low-fat milk", "Lugaw, thick consistency", "Heart, Chicken"], ["Pastillas, Langka", "Espasol"], ["Lettuce(raw)", "Peach halves", "Cassava", "Shells, Kuhol"], ["Tira-tira", "Kalamay, Latik"], ["Bataw pods", "Kamachile", "Lugaw, medium consistency", "Hito"]]
	Z = [["1 cup", "1 pc", "1 tetra-brick", "1 cup cooked rice + 2 cups water", "1 cup"], ["1 pc", "2 pcs"], ["1 cup", "6 pcs", "1 pc", "1 cup shelled"], ["1 pc", "1 pc"], ["1 cup", "7 pods", "1 cup cooked rice + 3 cups water", "4 pcs"]]
1500,[],[],[1,0,0,0]	X = -31498
	Y = [["Lettuce(raw)", "Watermelon", "Lite, low-fat milk", "Sweet Potatoes", "Octopus"], ["Banan cue", "Puto, Pula"], ["Stringbean Leaves", "Strawberry", "Lugaw, thin consistency", "Small Intestine, Carabeef"], ["Tira-tira", "Espasol"], ["Bataw pods", "Marang", "Ube", "Bangus"]]
	Z = [["1 cup", "1 slice", "1 tetra-brick", "1 pc", "1 cup"], ["1 pc", "3 pcs"], ["1 cup", "1 1/4 cups", "1 cup cooked rice + 5 cups water", "1 cup"], ["1 pc", "2 pcs"], ["1 cup", "1/2 pc", "2 pc", "3 slices"]]
1500,[],[],[0,3,0,2]	X = -31498
	Y = [["Eggplant", "Prunes, seedless", "Powdered, low-fat milk", "Ube", "Tilapya"], ["Caramel", "Pan Amerikano"], ["Balbalulang(seaweeds)", "Marang", "Cassava", "Heart, Pork"], ["Pastillas, Langka", "Masapudrida"], ["Singkamas pods", "Mango, ripe", "Rice, cooked", "Round, Carabeef"]];
	Z = [["1 cup", "6 pcs", "1/4 cup or 4 Tbsp", "2 pc", "4 pcs"], ["7 pcs", "2 pcs"], ["1 cup", "1/2 pc", "1 pc", "1 cup"], ["1 pc", "1 pc"], ["1 cup", "1 slice (medium)", "1 cup", "3 slice"]]
1500,[],[],[1,2,2,0]	X = -81452

	Y = [["Upo", "Papaya, ripe", "Long-life, skimmed milk", "Rice, cooked", "Sapsap, Daing"], ["Banan cue", "Jelly Roll"], ["Balbalulang(seaweeds)", "Mango, green", "Lugaw, medium consistency", "Round, Carabeef"], ["Tira-tira", "Puto, Seko Bilog"], ["Cabbage", "Mango, Paho", "Cassava", "Shoulder, Carabeef"]]
	Z = [["1 cup", "1 slice", "1 cup", "1 cup", "8 pcs"], ["1 pc", "1 slice"], ["1 cup", "1 slice (medium)", "1 cup cooked rice + 3 cups water", "3 slice"], ["1 pc", "3 pcs"], ["1 cup", "9 small", "1 pc", "3 slice"]]
1500,[],[],[2,4,4,4]	X = -61493
	Y = [["Kangkong", "Sweetened Mango", "Fresh cow's milk", "Gabi", "Pigeon pea seeds, Dried"], ["Tira-tira", "Tamales"], ["Pepper leaves", "Mangosteen", "Lugaw, medium consistency", "Shells, Kuhol"], ["Pastillas, Langka", "Spaghetti"], ["Singkamas tuber(lamang ugat)", "Rambutan", "Cassava", "Dilis"]]
	Z = [["1 cup", "1 glass", "1 cup", "4 pcs", "1 cup"], ["1 pc", "2 pcs"], ["1 cup", "3 pcs", "1 cup cooked rice + 3 cups water", "1 cup shelled"], ["1 pc", "4 cups"], ["1 cup", "8 pcs", "1 pc", "1 cup"]]
1500,[],["diabetes", "cancer"],[0,0,0,0]	X = -1500
	Y = [["Baby corn", "Unsweetened Orange", "Fresh carabao's milk", "Sweet Potatoes", "Heart, Pork"], [0, "Suman, Cassava"], ["Lima bean,pods(patani)", "Dates, pitted", "Lugaw, medium consistency", "Liver, Beef"], [0, "Puto, Bumbong"], ["Alugbati leaves", "Datiles", "Ube", "Shrimps, Suwahe"]]
	Z = [["5 pcs", "1 glass", "1 cup", "1 pc", "1 cup"], [0, "1/2 pc"], ["1 cup", "6 pcs", "1 cup cooked rice + 3 cups water", "1 cup"], [0, "2 pcs"], ["1 cup", "1 cup", "2 pc", "10 pcs"]]
1500,[],["diabetes", "cancer"],[1,0,0,0]	X = -31500
	Y = [["Cabbage", "Sineguelas", "Lite, low-fat milk", "Sweet Potatoes", "Bangus"], [0, "Tupig"], ["Cauliflower", "Pineapple", "Cassava", "Heart, Chicken"], [0, "Kutsinta"], ["Chayote leaves", "Lanzones", "Ube", "Dalagang Bukid"]]
	Z = [["1 cup", "10 pcs", "1 tetra-brick", "1 pc", "3 slices"], [0, "1/2 pc"], ["1 cup", "2 slices", "1 pc", "1 cup"], [0, "1 pc"], ["1 cup", "14 pcs", "2 pc", "4 pc"]]
1500,[],["diabetes", "cancer"],[0,3,0,2]	X = -31482
	Y = [["Sigarilyas", "Prunes, seedless", "Buttermilk, Powdered", "Ube", "Lean, Beef"], [0, "Puto, Puti"], ["Talinum", "Dikyam", "Lugaw, thick consistency", "Talangka"], [0, "Kalamay, Ube"], ["Alegaw leaves", "Dates, pitted", "Lugaw, thin consistency", "Shells, Kuhol"]]
	Z = [["1 cup", "6 pcs", "1/4 cup or 4 Tbsp", "2 pc", "3 slice"], [0, "1 slice"], ["1 cup", "6 pcs", "1 cup cooked rice + 2 cups water", "75 pcs"], [0, "1 slice"], ["1 cup", "6 pcs", "1 cup cooked rice + 5 cups water", "1 cup shelled"]]
1500,[],["diabetes", "cancer"],[1,2,2,0]	X = -91489
	Y = [["Camote leaves", "Tiesa", "Yoghurt", "Lugaw, thin consistency", "Liver, Carabeef"], [0, "Corn pudding"], ["Asparagus tips", "Sineguelas", "Cassava", "Beef Flank"], [0, "Suman, Ibos"], ["Pepper leaves", "Grapes", "Sweet Potatoes", "Round, Beef"]]
	Z = [["1 cup", "1/4 pc", "1/2 cup", "1 cup cooked rice + 5 cups water", "1 cup"], [0, "1 slice"], ["1 cup", "10 pcs", "1 pc", "3 slices"], [0, "1 pc"], ["1 cup", "15 pcs", "1 pc", "3 slice"]]
1500,[],["diabetes", "cancer"],[2,4,4,4]	X = -71492
	Y = [["Bamboo shoot", "Sweetened Mango", "Fresh cow's milk", "Rice, cooked", "Talangka"], [0, "Bibingka, Malagkit"], ["Garlic leaves", "Unsweetened Pineapple", "Cassava", "Lobster"], [0, "Baby corn"], ["Rimas", "Buko water", "Gabi", "Tilapya"]]
	Z = [["1 cup", "1 glass", "1 cup", "1 cup", "75 pcs"], [0, "1 slice"], ["1 cup", "1 glass", "1 pc", "6 Tbsp"], [0, "1 cup"], ["1 cup", "1 cup", "4 pcs", "4 pcs"]]
1500,[11,13],["diabetes", "cancer"],[0,0,0,0]	X = -1500
	Y = [["Unsoy", "Strawberry", "Long-life, skimmed milk", "Sweet Potatoes", "Shank, Beef"], [0, "Biko"], ["Mustard leaves", "Mango, ripe", "Rice, cooked", "Cenerloin, Beef"], [0, "Baby corn"], ["Lima bean,pods(patani)", "Mangosteen", "Cassava", "Beef Flank"]]
	Z = [["1 cup", "1 1/4 cups", "1 cup", "1 pc", "3 slice"], [0, "1 slice"], ["1 cup", "1 slice (medium)", "1 cup", "3 slice"], [0, "1 cup"], ["1 cup", "3 pcs", "1 pc",

	"3 slices"]]
1500,[11,13],["diabetes", "cancer"],[1,0,0,0]	X = -31499
	Y = [["Banana heart", "Prunes, seedless", "Lite, low-fat milk", "Cassava", "Chicken Leg"], [0, "Suman, Ibos"], ["Chayote leaves", "Champoy, salted", "Sweet Potatoes", "Small Intestine, Beef"], [0, "Espasol"], ["Bataw pods", "Unsweetened Prune", "Potato", "Chicken Meat"]]
	Z = [["1 cup", "6 pcs", "1 tetra-brick", "1 pc", "3 slice"], [0, "1 pc"], ["1 cup", "8 pcs", "1 pc", "1 cup"], [0, "2 pcs"], ["1 cup", "1 glass", "4 pcs", "3 slice"]]
1500,[11,13],["diabetes", "cancer"],[0,3,0,2]	X = -91500
	Y = [["Kaluray flowers", "Melon Kastila", "Yoghurt", "Cassava", "Sirloin Steak, Beef"], [0, "Suman, Marwekos"], ["Talinum", "Watermelon", "Lugaw, medium consistency", "Round, Carabeef"], [0, "Corn flakes"], ["Asparagus tips", "Duhat", "Rice, cooked", "Beef Plate"]]
	Z = [["1 cup", "1 slice", "1/2 cup", "1 pc", "3 slice"], [0, "2 pcs"], ["1 cup", "1 slice", "1 cup cooked rice + 3 cups water", "3 slice"], [0, "1 cup"], ["1 cup", "20 pcs", "1 cup", "3 slices"]]
1500,[11,13],["diabetes", "cancer"],[1,2,2,0]	X = -91500
	Y = [["Kaluray flowers", "Melon Kastila", "Yoghurt", "Cassava", "Sirloin Steak, Beef"], [0, "Suman, Marwekos"], ["Talinum", "Watermelon", "Lugaw, medium consistency", "Round, Carabeef"], [0, "Corn flakes"], ["Asparagus tips", "Duhat", "Rice, cooked", "Beef Plate"]]
	Z = [["1 cup", "1 slice", "1/2 cup", "1 pc", "3 slice"], [0, "2 pcs"], ["1 cup", "1 slice", "1 cup cooked rice + 3 cups water", "3 slice"], [0, "1 cup"], ["1 cup", "20 pcs", "1 cup", "3 slices"]]
1500,[11,13],["diabetes", "cancer"],[2,4,4,4]	X = -51494
	Y = [["Singkamas pods", "Datiles", "Buttermilk, Powdered", "Ube", "Shoulder, Carabeef"], [0, "Corn, boiled"], ["Carrots", "Sweetened Pine-grapefruit", "Lugaw, thin consistency", "Chicken Meat"], [0, "Tupig"], ["Pigeon pea pods(kadyos)", "Unsweetened Prune", "Cassava", "Liver, Pork"]]
	Z = [["1 cup", "1 cup", "1/4 cup or 4 Tbsp", "2 pc", "3 slice"], [0, "1 pc"], ["1 cup", "1 glass", "1 cup cooked rice + 5 cups water", "3 slice"], [0, "1/2 pc"], ["1 cup", "1 glass", "1 pc", "1 cup"]]
1750,[11,13],[],[0,0,0,0]	X = -1750
	Y = [["Bataw pods", "Apple", "Buttermilk, Powdered", "Lugaw, medium consistency", "Ham Sausage"], ["Pastillas, Durian", "Pan Amerikano"], ["Acelgas", "Mangosteen", "Cassava", "Alimango, Laman"], ["Bukayo", "Kalamay, Ube"], ["Alugbati leaves", "Marang", "Lugaw, thin consistency", "Lapu-lapu, Daing"]]
	Z = [["1 cup", "1 pc", "1/4 cup or 4 Tbsp", "1 cup cooked rice + 3 cups water", "3 pcs"], ["5 pc", "2 pcs"], ["1 cup", "3 pcs", "1 pc", "1 cup"], ["10 pc", "1 slice"], ["1 cup", "1/2 pc", "1 cup cooked rice + 5 cups water", "1 pc"]]
1750,[11,13],[],[1,0,0,0]	X = -31750
	Y = [["Chayote leaves", "Suha", "Lite, low-fat milk", "Gabi", "Chicken Head"], ["Tira-tira", "Corn flakes"], ["Ampalaya leaves", "Fruit Cocktail", "Lugaw, medium consistency", "Quail's Egg"], ["Banan cue", "Masapudrida"], ["Squash leaves", "Sineguelas", "Lugaw, thin consistency", "Baloko"]]
	Z = [["1 cup", "3 segments", "1 tetra-brick", "4 pcs", "2 heads"], ["1 pc", "1 cup"], ["1 cup", "1 cup", "1 cup cooked rice + 3 cups water", "9 pcs"], ["1 pc", "1 pc"], ["1 cup", "10 pcs", "1 cup cooked rice + 5 cups water", "3 slices"]]
1750,[11,13],[],[0,3,0,2]	X = -41668
	Y = [["Alugbati leaves", "Champoy, salted", "Lite, low-fat milk", "Gabi", "Heart, Chicken"], ["Maraschino cherries", "Marie"], ["Upo", "Melon Kastila", "Cassava", "Chicken Meat"], ["Maruya", "Pan de leche"], ["Spinach", "Dates, pitted", "Lugaw, thin consistency", "Heart, Pork"]]
	Z = [["1 cup", "8 pcs", "1 tetra-brick", "4 pcs", "1 cup"], ["10", "22 pcs"], ["1 cup", "1 slice", "1 pc", "3 slice"], ["1 pc", "1 pc"], ["1 cup", "6 pcs", "1 cup cooked rice + 5 cups water", "1 cup"]]
1750,[11,13],[],[1,2,2,0]	X = -91748

	Y = [["Pokpoklo(seaweed)", "Tamarind", "Powdered, low-fat milk", "Sweet Potatoes", "Liver, Beef"], ["Bukayo", "Tupig"], ["Baguio Beans", "Kamachile", "Lugaw, thick consistency", "Shank, Carabeef"], ["Caramel", "Sotanghon"], ["Sweet pea pods(sitsaro)", "Banana", "Rice, cooked", "Heart, Beef"]]
	Z = [["1 cup", "4 pcs", "1/4 cup or 4 Tbsp", "1 pc", "1 cup"], ["10 pc", "1/2 pc"], ["1 cup", "7 pods", "1 cup cooked rice + 2 cups water", "3 slice"], ["7 pcs", "3 cups"], ["1 cup", "1 pc", "1 cup", "1 cup"]]
1750,[11,13],[],[2,4,4,4]	X = -81732
	Y = [["Baby corn", "Buko water", "Buttermilk, Powdered", "Lugaw, medium consistency", "Alimasag, Laman"], ["Caramel", "Spaghetti"], ["Pokpoklo(seaweed)", "Unsweetened Pineapple", "Cassava", "Shells, Kuhol"], ["Pastillas, Durian", "Suman, Marwekos"], ["Saluyot", "Unsweetened Prune", "Ube", "Shrimps, Suwahe"]]
	Z = [["5 pcs", "1 cup", "1/4 cup or 4 Tbsp", "1 cup cooked rice + 3 cups water", "1 cup"], ["7 pcs", "4 cups"], ["1 cup", "1 glass", "1 pc", "1 cup shelled"], ["5 pc", "2 pcs"], ["1 cup", "1 glass", "2 pc", "10 pcs"]]
1750,[11,13],["diabetes", "cancer"],[0,0,0,0]	X = -1750
	Y = [["Onion bulb", "Sweetened Mango", "Buttermilk, Powdered", "Ube", "Alamang, Tagunton"], [0, "Kalamay, Ube"], ["Patola", "Lychees", "Cassava", "Alimango, Laman"], [0, "Bihon"], ["Stringbean Leaves", "Sweetened Pine-orange", "Lugaw, medium consistency", "Quail's Egg"]]
	Z = [["1 cup", "1 glass", "1/4 cup or 4 Tbsp", "2 pc", "16 Tbsp"], [0, "1 slice"], ["1 cup", "10 pcs", "1 pc", "1 cup"], [0, "2 cups"], ["1 cup", "1 glass", "1 cup cooked rice + 3 cups water", "9 pcs"]]
1750,[11,13],["diabetes", "cancer"],[1,0,0,0]	X = -31750
	Y = [["Alugbati leaves", "Sweetened Mango", "Buttermilk, Liquid", "Ube", "Alimango, Aligue"], [0, "Suman, Cassava"], ["Langka, hilaw", "Mango, green", "Lugaw, thin consistency", "Liver, Carabeef"], [0, "Ampaw Pinipig"], ["Saluyot", "Grapes", "Sweet Potatoes", "Pork, Leg(pata)"]]
	Z = [["1 cup", "1 glass", "1 cup", "2 pc", "6 Tbsp"], [0, "1/2 pc"], ["1 cup", "1 slice (medium)", "1 cup cooked rice + 5 cups water", "1 cup"], [0, "3 pcs"], ["1 cup", "15 pcs", "1 pc", "3 slices"]]
1750,[11,13],["diabetes", "cancer"],[0,3,0,2]	X = -31623
	Y = [["Acelgas", "Raisins, seedless", "Long-life, skimmed milk", "Sweet Potatoes", "Talangka"], [0, "Puto, Bumbong"], ["Kaluray flowers", "Mango chips", "Cassava", "Cenerloin, Beef"], [0, "Puto, Seko Bilog"], ["Camote leaves", "Dates, pitted", "Gabi", "Shank, Carabeef"]]
	Z = [["1 cup", "4 Tbsp", "1 cup", "1 pc", "75 pcs"], [0, "2 pcs"], ["1 cup", "6 pcs", "1 pc", "3 slice"], [0, "3 pcs"], ["1 cup", "6 pcs", "4 pcs", "3 slice"]]
1750,[11,13],["diabetes", "cancer"],[1,2,2,0]	X = -91745
	Y = [["Camote leaves", "Tiesa", "Lite, low-fat milk", "Rice, cooked", "Beef Brisket"], [0, "Espasol"], ["Upo", "Lanzones", "Gabi", "Sirloin Steak, Beef"], [0, "Corn flakes"], ["Celery", "Dalanghita", "Lugaw, thin consistency", "Porterhouse Steak, Beef"]]
	Z = [["1 cup", "1/4 pc", "1 tetra-brick", "1 cup", "3 slices"], [0, "2 pcs"], ["1 cup", "14 pcs", "4 pcs", "3 slice"], [0, "1 cup"], ["1 cup", "5 pcs", "1 cup cooked rice + 5 cups water", "3 slice"]]
1750,[11,13],["diabetes", "cancer"],[2,4,4,4]	X = -91695
	Y = [["Pepper", "Unsweetened Pineapple", "Fresh carabao's milk", "Sweet Potatoes", "Squid"], [0, "Ampaw Pinipig"], ["Kangkong", "Unsweetened Prune", "Gabi", "Dilis"], [0, "Espasol"], ["Banana heart", "Unsweetened Orange", "Cassava", "Shrimps, Sugpo"]]
	Z = [["1 cup", "1 glass", "1 cup", "1 pc", "8 pcs"], [0, "3 pcs"], ["1 cup", "1 glass", "4 pcs", "1 cup"], [0, "2 pcs"], ["1 cup", "1 glass", "1 pc", "5 pcs"]]
2000,[10,11,13],["gallbladder disease", "atherosclerosis", "stroke", "heart disease", "osteoarthritis"],[0,0,0,0]	X = -1999
	Y = [["Pepper leaves", "Makopa", "Lite, low-fat milk", "Ube", "Tenderloin, Pork"], [0, "Puto, Bumbong"], ["Pigeon pea pods(kadyos)", "Sweetened Apple", "Potato", "Beef Brisket"], [0, "Corn, boiled"], ["Onion bulb", "Apple Sauce", "Sweet Potatoes", "Sirloin Steak, Beef"]]
	Z = [["1 cup", "8 pcs", "1 tetra-brick", "2 pc", "3 slice"], [0, "2 pcs"], ["1 cup", "1 glass", "4 pcs", "3 slices"], [0, "1 pc"], ["1 cup", "1 cup", "1 pc", "3 slice"]]
2000,[10,11,13],["gallbladder disease", "atherosclerosis",	X = -31999

"stroke", "heart disease", "osteoarthritis"],[1,0,0,0]	Y = [["Squash leaves", "Sweetened Mango", "Powdered, low-fat milk", "Potato", "Beef Brisket"], [0, "Biko"], ["Pokpoklo(seaweed)", "Apple Sauce", "Gabi", "Chicken Meat"], [0, "Bihon"], ["Celery", "Mango, Paho", "Lugaw, thick consistency", "Meat, Carabeef"]]
	Z = [["1 cup", "1 glass", "1/4 cup or 4 Tbsp", "4 pcs", "3 slices"], [0, "1 slice"], ["1 cup", "1 cup", "4 pcs", "3 slice"], [0, "2 cups"], ["1 cup", "9 small", "1 cup cooked rice + 2 cups water", "3 slice"]]
2000,[10,11,13],["gallbladder disease", "atherosclerosis", "stroke", "heart disease", "osteoarthritis"],[0,3,0,2]	X = -1999
	Y = [["Pepper leaves", "Sweetened Mango", "Lite, low-fat milk", "Potato", "Rump, Carabeef"], [0, "Puto, Pula"], ["Cabbage", "Strawberry", "Rice, cooked", "Beef Brisket"], [0, "Bihon"], ["Kaluray leaves", "Fruit Cocktail", "Gabi", "Porterhouse Steak, Beef"]]
	Z = [["1 cup", "1 glass", "1 tetra-brick", "4 pcs", "3 slice"], [0, "3 pcs"], ["1 cup", "1 1/4 cups", "1 cup", "3 slices"], [0, "2 cups"], ["1 cup", "1 cup", "4 pcs", "3 slice"]]
2000,[10,11,13],["gallbladder disease", "atherosclerosis", "stroke", "heart disease", "osteoarthritis"],[1,2,2,0]	X = -91960
	Y = [["Mustard leaves", "Rambutan", "Powdered, low-fat milk", "Potato", "Beef Flank"], [0, "Bihon"], ["Sigarilyas", "Tamarind", "Rice, cooked", "Rump, Carabeef"], [0, "Sotanghon"], ["Talinum", "Lychees", "Lugaw, thin consistency", "Beef Plate"]]
	Z = [["1 cup", "8 pcs", "1/4 cup or 4 Tbsp", "4 pcs", "3 slices"], [0, "2 cups"], ["1 cup", "4 pcs", "1 cup", "3 slice"], [0, "3 cups"], ["1 cup", "10 pcs", "1 cup cooked rice + 5 cups water", "3 slices"]]
2000,[10,11,13],["gallbladder disease", "atherosclerosis", "stroke", "heart disease", "osteoarthritis"],[2,4,4,4]	X = -61980
	Y = [["Singkamas pods", "Unsweetened Prune", "Long-life, skimmed milk", "Lugaw, thick consistency", "Chicken Leg"], [0, "Bihon"], ["Spinach", "Sweetened Mango", "Lugaw, medium consistency", "Beef Flank"], [0, "Bibingka, Galapong"], ["Carrots", "Sweetened Pine-orange", "Gabi", "Shank, Carabeef"]]
	Z = [["1 cup", "1 glass", "1 cup", "1 cup cooked rice + 2 cups water", "3 slice"], [0, "2 cups"], ["1 cup", "1 glass", "1 cup cooked rice + 3 cups water", "3 slices"], [0, "1 slice"], ["1 cup", "1 glass", "4 pcs", "3 slice"]]
2250,[10,11,12,13],["diabetes", "cancer"],[0,0,0,0]	X = -2240
	Y = [["Onion bulb", "Mango chips", "Powdered, low-fat milk", "Lugaw, thick consistency", "Porterhouse Steak, Beef"], [0, "Corn pudding"], ["Chayote leaves", "Pineapple, sliced", "Ube", "Beef Plate"], [0, "Golden corn, canned"], ["Kaluray flowers", "Tamarind", "Rice, cooked", "Beef Brisket"]]
	Z = [["1 cup", "6 pcs", "1/4 cup or 4 Tbsp", "1 cup cooked rice + 2 cups water", "3 slice"], [0, "1 slice"], ["1 cup", "5 slices", "2 pc", "3 slices"], [0, "1 cup"], ["1 cup", "4 pcs", "1 cup", "3 slices"]]
2250,[10,11,12,13],["diabetes", "cancer"],[1,0,0,0]	X = -32244
	Y = [["Langka, hilaw", "Pineapple, crushed", "Fresh carabao's milk", "Lugaw, thick consistency", "Tenderloin, Beef"], [0, "Cassava Cake"], ["Singkamas tuber(lamang ugat)", "Dikyam", "Ube", "Shoulder, Carabeef"], [0, "Golden corn, canned"], ["Talinum", "Mango chips", "Cassava", "Beef Flank"]]
	Z = [["1 cup", "1 cup", "1 cup", "1 cup cooked rice + 2 cups water", "3 slice"], [0, "1 slice"], ["1 cup", "6 pcs", "2 pc", "3 slice"], [0, "1 cup"], ["1 cup", "6 pcs", "1 pc", "3 slices"]]
2250,[10,11,12,13],["diabetes", "cancer"],[0,3,0,2]	X = -22184
	Y = [["Langka, hilaw", "Pineapple, crushed", "Fresh carabao's milk", "Lugaw, thick consistency", "Tenderloin, Beef"], [0, "Cassava Cake"], ["Singkamas tuber(lamang ugat)", "Dikyam", "Ube", "Shoulder, Carabeef"], [0, "Golden corn, canned"], ["Talinum", "Mango chips", "Cassava", "Beef Flank"]]
	Z = [["1 cup", "1 cup", "1 cup", "1 cup cooked rice + 2 cups water", "3 slice"], [0, "1 slice"], ["1 cup", "6 pcs", "2 pc", "3 slice"], [0, "1 cup"], ["1 cup", "6 pcs", "1 pc", "3 slices"]]
2250,[10,11,12,13],["diabetes", "cancer"],[1,2,2,0]	X = -82135
	Y = [["Gabi leaves", "Lychees", "Fresh cow's milk", "Potato", "Beef Plate"], [0, "Cassava Cake"], ["Cauliflower", "Mabolo", "Rice, cooked", "Lean, Beef"], [0, "Puto, Seko Bilog"], ["Camote leaves", "Sweetened Apple", "Lugaw, thick consistency", "Beef Brisket"]]

	Z = [["1 cup", "10 pcs", "1 cup", "4 pcs", "3 slices"], [0, "1 slice"], ["1 cup", "1 pc", "1 cup", "3 slice"], [0, "3 pcs"], ["1 cup", "1 glass", "1 cup cooked rice + 2 cups water", "3 slices"]]
2250,[10,11,12,13],["diabetes", "cancer"],[2,4,4,4]	X = -62065
	Y = [["Tomato", "Unsweetened Prune", "Lite, low-fat milk", "Lugaw, thick consistency", "Pork, Leg(pata)"], [0, "Bihon"], ["Sigarilyas", "Sweetened Apple", "Gabi", "Rump, Carabeef"], [0, "Kalamay, Latik"], ["Squash", "Sweetened Pine-orange", "Ube", "Tenderloin, Beef"]]
	Z = [["1 cup", "1 glass", "1 tetra-brick", "1 cup cooked rice + 2 cups water", "3 slices"], [0, "2 cups"], ["1 cup", "1 glass", "4 pcs", "3 slice"], [0, "1 pc"], ["1 cup", "1 glass", "2 pc", "3 slice"]]
2500,[10],["gallbladder disease", "atherosclerosis", "stroke", "heart disease", "osteoarthritis"],[2,4,4,4]	X = -12269
	Y = [["Celery", "Lychees", "Yoghurt", "Lugaw, thick consistency", "Beef Plate"], [0, "Corn/rice curls"], ["Balbalulang(seaweeds)", "Pineapple, crushed", "Ube", "Beef Brisket"], [0, "Bihon"], ["Kamansi", "Pineapple, sliced", "Potato", "Lean, Beef"]]
	Z = [["1 cup", "10 pcs", "1/2 cup", "1 cup cooked rice + 2 cups water", "3 slices"], [0, "2 cups"], ["1 cup", "1 cup", "2 pc", "3 slices"], [0, "2 cups"], ["1 cup", "5 slices", "4 pcs", "3 slice"]]

Table 1: Test run

Table 1 shows the tests made by the proponents. In the output column, shows the results of the program after the inputs were inputted, where X is the cost, Y is the menu generated which is a nested list of the meals, and Z is the amount of the food which is parallel to Y. The cost can be calculated using the formula in figure 8. The first number in the rightmost part of the cost represents the preference while the rest of the numbers in the leftmost part of the cost is the total calorie of the menu that was generated.

The table above was shown by the proponents to the nutritionist for the evaluation. According to the nutritionist, the system that the proponents made was able to generate a promising result; however some of the combination of the foods were mismatched. The nutritionist suggests to the researches to develop the system and to make it more personalized and specific so that this would be a helpful tool to the nutritionist and the dietician.

6. CONCLUSION AND RECOMMENDATION

After all the studies conducted, the proponents discovered that using constraint logic programming (CLP) as an approach is one possible way of solving a dietary meal planning. This was because constraints were easily applied while the solutions are being backtracked.

One recommendation for future studies is the total change of the current facts or the food database. Though the current facts, are legit, the main problem is that the food are more of an ingredient than a recipe. Also, the calories that are placed on the current facts are fixed to one certain food. If possible, the food should have three (3) different sizes: small, medium and large. With this, the choice of food would be larger and more flexible.

Another thing to add is the rule of rice. The problem behind the proponents system is that rice was only limited to once per day, which adopted the same rule as the other food. If possible, the system may become better if rice is used throughout the day, because as what the proponents have researched, majority of the Filipino meal had rice.

The system also needs more rules unto how the meals are being combined. This may significantly decrease run time and also produce more accurate solutions.

The result of the study may be used by dieticians and nutritionists on giving their patients one of the best meal plan for the day. The system still cannot be used by the public or the general, the reason being the facts and lack of rules. Unlike the dieticians and nutritionists, the public does not know whether the meal combination is perfect or not. This system may become a tool for the public once perfected. And as one of the proponent's dietician said, "When this system is perfected, we will have no jobs".

REFERENCES

Kashima, T., Matsumoto, S., Ishii, H., Evaluation of Menu Planning Capability Based on Multidimensional 0-1 Knapsack Problem of Nutritional Management System. IAENG International Journal of Applied Mathematics 39, IJAM_39_04, 2009.

Seljak, B. K., Computer-Based Dietary Menu Planning: How to Support It by Complex Knowledge?, Computer System Department, Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia, 2010.

Schwarzberg, R., Zabinski, M., Melton, R. and Dion, T.J., System and Method for Automatically Defining, Creating and Managing Meals, Standley Law Group LLp, 6300 Riverside D rive Dublin, OH 43017 (US), 2009.

Vassanyi, I., Kosa, I., Pinter, B. and Gaal, B., Personalized Dietary Counseling System Using Harmony Rules in Tele-Care, University of Pannonia, Medica Informatics RD Center, Veszprem, Hungary, 2014.

Chien-Yeh Hsu Ph.D., Li-Chieh Huang M.S., Tzuo Ming Chen, Ph.D., Li-Fu Chen, M.S., and Jane C.-J. Chao, Ph.D., A Web-Based Decision Support System for Dietary Analysis and Recommendations, Graduate Institute of Medical Informatics, Taipei Medical University, Taipei, Taiwan, 2011.

Narciso, M.H., Filipino Meal Patterns in the Unites States of America, The Graduate School, University of Wisconsin Stout Menomonie, WI 54751, 2005.

Simonis, H. Developing Applications with ECLiPSe. n.d.

Simonis, H. Scheduling and Planning with Constraint Logic Programming. COSYTEC SA, 1995.

Simonis, H. Constraint Logic Programming. COSYTEC SA, Orsay, France, 2008.

Apt, K.R., Wallace, M., Constraint Logic Programming Using Eclipse, Cambridge University Press New York, NY, USA, 2007.

Fruhwirth, T., Herold, A., Kuchenhoff, V., Provost, T.L., Lim, P., Monfroy, E. and Wallace, M., Constraint Logic Programming – An Informal Introduction, Springer Berlin Heidelberg, 1992.