University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

Weight Management Patient Portal for Filipinos and Training Nutritionists

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Francis B. Cabalo

August 2017

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled "Weight Management Patient Portal for Filipinos and Training Nutritionists" prepared and submitted by Francis B. Cabalo in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Ma. Sheila A. Magboo, M.Sc.
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Gregorio B. Baes, Ph.D. (*cand.*)	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Richard Bryann L. Chua, Ph.D.(*cand.*)	_____	_____
4. Perlita E. Gasmen, M.Sc. (*cand.*)	_____	_____
5. Marvin John C. Ignacio, M.Sc. (*cand.*)	_____	_____
6. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Ma. Sheila A. Magboo, M.Sc.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences
and Mathematics

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences
and Mathematics

Leonardo R. Estacio Jr., Ph.D.
Dean
College of Arts and Sciences

Abstract

Monitoring and keeping record of a case study is not done daily in the training of a student nutritionist, which compromises the effectiveness of the treatment of the patient. The Weight Management Patient Portal for Filipinos and Training Nutritionist, is web-based application that implements standard procedure of DTI-FNRI in calculating meal plan and sample menu and also serves as a record database and communication medium for a student nutritionist and patient for student's case study.
Keywords: Meal Planning, weight management, food exchange list

Contents

List of Figures

List of Tables

I. Introduction

A. Background of the Study

Body Mass Index (BMI) is an estimate of an individuals tissue mass (muscle, fat, and bone) [1]. It is computed as in weight in kilograms divided by the square of height in meters. It is used to categorize a person as either underweight, normal, overweight, or obese. In the Philippines the BMI range for these 4 categories can be seen in Figure 1

BODY MASS INDEX FOR FILIPINOS *

Body Mass Index = (Weight [kilograms])/Height [meters]2)

Actual Weight kg	lbs	Height, m (in.) 1.27 (50.0)	1.30 (51.0)	1.32 (52.0)	1.34 (53.0)	1.37 (54.0)	1.40 (55.0)	1.42 (56.0)	1.45 (57.0)	1.47 (58.0)	1.50 (59.0)	1.52 (60.0)	1.55 (61.0)	1.57 (62.0)	1.60 (63.0)	1.62 (64.0)	1.65 (65.0)	1.68 (66.0)	1.70 (67.0)	1.73 (68.0)	1.75 (69.0)	1.78 (70.0)	1.80 (71.0)	1.83 (72.0)	1.85 (73.0)
23	51	14.3	13.6	13.2	12.8	12.2																			
25	56	15.5	14.8	14.4	13.9	13.3	12.8	12.5																	
27	60	16.8	16.0	15.5	15.0	14.4	13.8	13.5	12.9	12.5	12.0														
29	64	18.0	17.2	16.7	16.1	15.4	14.8	14.5	13.8	13.4	12.9	12.6	12.1												
31	68	19.2	18.3	17.8	17.2	16.5	15.8	15.5	14.8	14.4	13.8	13.4	12.9	12.6	12.1										
33	73	20.5	19.5	19.0	18.3	17.6	16.8	16.5	15.7	15.3	14.7	14.3	13.8	13.4	12.9	12.6	12.1								
35	77	21.7	20.7	20.1	19.4	18.6	17.9	17.5	16.7	16.2	15.6	15.2	14.6	14.2	13.7	13.4	12.9	12.4	12.1						
37	81	23.0	21.9	21.3	20.6	19.7	18.9	18.5	17.6	17.1	16.4	16.0	15.4	15.0	14.4	14.1	13.6	13.1	12.8	12.4	12.1				
39	86	24.2	23.1	22.4	21.7	20.7	19.9	19.5	18.6	18.1	17.3	16.9	16.2	15.9	15.2	14.9	14.3	13.8	13.5	13.0	12.7	12.3	12.0		
41	90	25.5	24.3	23.6	22.8	21.8	20.9	20.5	19.5	19.0	18.2	17.8	17.1	16.7	16.0	15.6	15.1	14.5	14.2	13.7	13.4	12.9	12.7	12.2	12.0
43	95	26.7	25.4	24.7	23.9	22.9	21.9	21.5	20.5	19.9	19.1	18.6	17.9	17.5	16.8	16.4	15.8	15.2	14.9	14.4	14.1	13.6	13.3	12.8	12.6
45	99	28.0	26.6	25.9	25.0	23.9	23.0	22.5	21.4	20.9	20.0	19.5	18.8	18.3	17.6	17.2	16.5	16.0	15.6	15.0	14.7	14.2	13.9	13.4	13.2
47	103	29.2	27.8	27.0	26.1	25.0	24.0	23.5	22.4	21.8	20.9	20.4	19.6	19.1	18.4	17.9	17.3	16.7	16.3	15.7	15.4	14.8	14.5	14.0	13.7
49	108	30.4	29.0	28.2	27.2	26.1	25.0	24.5	23.3	22.7	21.8	21.2	20.4	19.9	19.1	18.7	18.0	17.4	17.0	16.4	16.0	15.4	15.1	14.6	14.3
51	112	31.1	29.6	28.7	28.3	27.1	26.0	25.5	24.3	23.6	22.7	22.1	21.2	20.7	19.9	19.5	18.8	18.1	17.6	17.1	16.7	16.1	15.7	15.2	14.9
53	117	32.3	30.8	29.9	29.4	28.2	27.0	26.5	25.2	24.5	23.6	22.9	22.1	21.5	20.7	20.2	19.5	18.8	18.3	17.7	17.3	16.7	16.4	15.8	15.5
55	121	33.5	32.0	31.0	30.6	29.2	28.1	27.5	26.2	25.5	24.4	23.8	22.9	22.4	21.5	21.0	20.2	19.5	19.0	18.4	18.0	17.4	17.0	16.4	16.1
57	126	34.8	33.1	32.2	31.7	30.3	29.1	28.5	27.1	26.4	25.3	24.7	23.8	23.2	22.3	21.8	21.0	20.2	19.7	19.1	18.6	18.0	17.6	17.0	16.7
59	130	36.0	34.3	33.3	32.8	31.4	30.1	29.5	28.1	27.3	26.2	25.5	24.6	24.0	23.0	22.5	21.7	20.9	20.4	19.7	19.3	18.6	18.2	17.6	17.2
61	134	37.3	35.5	34.5	33.9	32.4	31.1	30.5	29.0	28.2	27.1	26.4	25.4	24.8	23.8	23.3	22.4	21.6	21.1	20.4	19.9	19.2	18.8	18.2	17.8
63	139	38.5	36.7	35.6	35.0	33.5	32.1	31.5	30.0	29.2	28.0	27.3	26.2	25.6	24.6	24.0	23.2	22.3	21.8	21.1	20.6	19.9	19.4	18.8	18.4
65	143	39.8	37.9	36.8	36.1	34.6	33.2	32.5	31.0	30.1	28.9	28.1	27.1	26.4	25.4	24.8	23.9	23.0	22.5	21.7	21.2	20.5	20.1	19.4	19.0
67	147	41.0	39.1	37.9	37.2	35.6	34.2	33.5	31.9	31.0	29.8	29.0	27.9	27.2	26.2	25.6	24.6	23.8	23.2	22.4	21.9	21.1	20.7	20.0	19.6
69	152	42.2	40.2	39.1	38.3	36.7	35.2	34.5	32.9	31.9	30.7	29.9	28.8	28.0	27.0	26.3	25.4	24.5	23.9	23.1	22.5	21.8	21.3	20.6	20.2
71	156	43.5	41.4	40.2	39.4	37.8	36.2	35.5	33.8	32.9	31.5	30.7	29.6	28.9	27.7	27.1	26.1	25.2	24.6	23.7	23.2	22.4	21.9	21.2	20.8
73	161	44.7	42.6	41.4	40.6	38.8	37.2	36.5	34.8	33.8	32.4	31.6	30.4	29.7	28.5	27.9	26.8	25.9	25.2	24.4	23.8	23.0	22.5	21.8	21.3
75	165		43.8	42.5	41.7	39.9	38.3	37.5	35.7	34.7	33.3	32.5	31.2	30.5	29.3	28.6	27.6	26.6	26.0	25.1	24.5	23.6	23.1	22.4	21.9
77	169		45.0	43.7	42.8	41.0	39.3	38.5	36.7	35.6	34.2	33.3	32.1	31.3	30.1	29.4	28.3	27.3	26.6	25.8	25.2	24.3	23.8	23.0	22.5
79	174			44.8	43.9	42.0	40.3	39.5	37.6	36.6	35.1	34.2	32.9	32.1	30.8	30.2	29.0	28.0	27.3	26.4	25.8	24.9	24.4	23.6	23.1
81	178				45.0	43.1	41.3	40.5	38.6	37.5	36.0	35.1	33.8	32.9	31.6	30.9	29.8	28.7	28.0	27.1	26.5	25.6	25.0	24.2	23.7
83	183					44.1	42.3	41.5	39.5	38.4	36.9	35.9	34.6	33.7	32.4	31.7	30.5	29.4	28.7	27.8	27.1	26.2	25.6	24.8	24.3
85	187						43.4	42.5	40.5	39.4	37.8	36.8	35.4	34.6	33.2	32.4	31.2	30.1	29.4	28.4	27.8	26.8	26.2	25.4	24.8
87	191						44.4	43.5	41.4	40.3	38.7	37.7	36.2	35.4	34.0	33.2	32.0	30.8	30.1	29.1	28.4	27.4	26.9	26.0	25.4
89	196							44.5	42.4	41.2	39.6	38.5	37.1	36.2	34.8	34.0	32.7	31.6	30.8	29.8	29.1	28.1	27.5	26.6	26.0
91	200								43.3	42.1	40.4	39.4	37.9	37.0	35.5	34.7	33.4	32.3	31.5	30.4	29.7	28.7	28.1	27.2	26.6
93	205								44.3	43.1	41.3	40.2	38.8	37.8	36.3	35.5	34.2	33.0	32.2	31.1	30.4	29.3	28.7	27.8	27.2
95	209									44.0	42.2	41.1	39.6	38.6	37.1	36.3	34.9	33.7	32.9	31.8	31.0	30.0	29.3	28.4	27.8
97	213									44.9	43.1	42.0	40.4	39.4	37.9	37.0	35.7	34.4	33.6	32.4	31.7	30.6	29.9	29.0	28.4
99	218										44.0	42.8	41.2	40.2	38.7	37.8	36.4	35.1	34.2	33.1	32.4	31.2	30.6	29.6	28.9
101	222										44.9	43.7	42.1	41.1	39.4	38.5	37.1	35.8	34.9	33.8	33.0	31.9	31.2	30.1	29.5
103	227											44.6	42.9	41.9	40.2	39.3	37.9	36.5	35.6	34.5	33.7	32.5	31.8	30.7	30.1
105	231												43.8	42.7	41.0	40.1	38.6	37.2	36.3	35.1	34.3	33.1	32.4	31.3	30.7
107	236												44.6	43.5	41.8	40.8	39.3	37.9	37.0	35.8	35.0	33.8	33.0	31.9	31.3
109	240													44.3	42.6	41.6	40.1	38.7	37.7	36.5	35.6	34.4	33.6	32.5	31.9
110	242													44.7	43.0	42.0	40.4	39.0	38.1	36.8	36.0	34.7	34.0	32.8	32.2

COLOR CODES		
Color	Interpretation	BMI(kg/m2)
Aquamarine	Underweight	<18.5
Green	Normal	18.5 - 24.9
Blue	Overweight	25 - 30
Red	Obese	>30

Figure 1: BMI Table for Filipinos

A study published in The Lancet, reveals that the Philippines is among the countries with the highest rate of underweight men and women. Analysing statistical data from 1975 to 2014, the study revealed that Philippines ranked ninth in terms of underweight men (3.6 million) and eighth in terms of underweight women (4.4 million). [2] In terms of prevalence, or rate of underweight population, the Philippines is at 14.3% for women, or 16th in the world. For men, it is 12.3 percent, which is the 40th globally [3]. Nutritional deficiency has been a problem in the Philippines. According to Lawrence Haddad, nutrition expert and lead author of the 2015 Global Nutrition Report, the country is lagging behind other poor countries in addressing malnutrition, not only in adults but children as well. Citing the 2013 Food and Nutrition Research Institutes National Nutrition Survey, which showed that 19.9 percent of children under five years are underweight, 30.3 percent stunted and 7.9 percent wasted, [2]

On the other hand, three of 10 Filipino adults are now overweight or obese, according to the National Nutrition Council (NNC). Citing the National Nutrition Survey of the Food and Nutrition Research Institute (FNRI), it is noted that the number of overweight or obese Filipino adults increased by 14.5 percentage points, from 16.6% in 1993 to 31.1% in 2013 [4]. The occurence of obesity is most prevalent in the 40-49.9 age group and least prevalent in people 70 years old and up, the report said. It also noted that there are more females who are obese than males. Specifically, android obesity or apple-shaped obesity is more common among women than men, according to FNRI[5] the rise of obesity in the country is attributed to the influx of processed foods and accessibility to fast foods, lack of physical activities, the change of dietary intakes and lifestyle, child undernutrition, and poor breastfeeding practices.[6]

These clearly show that the current nutritional status here in the Philippines lies in the extremes. These two extreme conditions are risk factors to chronic diseases and even mortality.[7] Achieving and maintaining of a normal BMI is one of the primary

concerns of DOH to the public. According to Asec. Gerardo Bayugo, Anything in excess is not advisable and all these processed meat somehow they contain chemicals and preservatives and all of these things are carcinogenic in excess. The best thing that a person can do is to make sure that his or her plate contains a balanced diet meat, carbohydrates, fruits and vegetables and water. [8]

BMI is the primary indicator of ones nutritional status. Though it is not sufficient to tell whether a person is underweight, normal or overweight, its simple computation is enough to check any signs of abnormality in the nutritional status. [1] BMI is also the primary input to compute for the ideal body of the person. If a person's weight is either below or above the normal range, changing the meal pattern is an effective therapy to attain a desirable weight. [9] The weight management therapy is done with a supervision of a nutritionist, a dietician, a doctor or a nurse. The nutritionist designs a meal plan to be followed by the patient in order to improve their weight. Through the use of food exchange lists as their primary tool to design the meal pattern, varieties on food items is available while the process of improving one's nutritional status is still attainable.

A meal plan is a guide to help you plan daily meals and snacks. It allows you to eat foods that you enjoy and that provide a good balance of nutrients for your health. Each meal plan gives an example of one day's food intake using the food exchange system. [10] Food exchange system was originally developed to facilitate the computation and planning of diabetic diets. In the Food Exchange Lists, commonly used foods are divided into seven groups, namely, (1) Vegetables, (2) Fruits, (3) Milk, (4) Rice, (5) Meat and Fish, (6) Fat, (7) Sugar (Table 2). [11] These food groups are based on the DOST-FNRI diet manual, which is the guidebook of student nutritionist as well as a professional dietitian. The entire process of designing the meal plan is direct forward for the initial steps, then it will be subjective since meal preference vary from one person to another. Being knowledgeable about food is not sufficient

to provide the basis for planning balanced diet. This process is foundation skill to be taught on nutritionist. [12] As a primary skill students of different schools (i.e. Philippine Normal University, UP Diliman) are trained in a real situation. In this practice students are either assigned by their teacher to a community to monitor, or they find their own patient with a specific condtion to aid them for their nutritional needs. [13]

B. Statement of the Problem

In the training of a student nutrionist, it is a fundamental requirement to have case study which aims to treat a non-normal BMI patient. Monitoring the daily caloric intake of the patient is a simple but its the regular recording of daily intake, and the proper planning of what food a person should take is a big factor in order to achieve or maintain a healthy weight. The computation of right amount of macronutrients and its distribution to the daily menu of a person is a task requiring broad knowledge of what nutrients each food contains.

C. Objectives of the Study

This research aims to create a Web portal for weight management with the following functionalities:

1. Allows the student nutritionist to
    (a) Add patient's profile
    (b) Edit his/her own profile
    (c) Auto-generate a meal plan for the patient
    (d) Create a menu for the patient
    (e) Update patient's height and weight

(f) View patient's daily food intake

(g) Give feedback to patient

2. Allows the patient to

(a) Edit his/her own profile

(b) View sample menu

(c) Check a food item's nutritional value based on food exchange list

(d) Input daily food intake

(e) Receive and view feedback from nutrition student

3. Allows the teacher dietitian to

(a) Add student's profile

(b) Edit his/her own profile

(c) Reassign patient to other student

(d) Reassign student to another teacher

(e) Add new item in the food exchange list

4. Allows the administator to

(a) Add teacher's profile

(b) Deactivate patient, student and teacher profile

D. Significance of the Project

This system aims to aid the training of a nutrition student, who's soon to be professionals, rather than a peer-to-peer diet advisory. This is done through the provided auto-generating meal plan for their patients, but still letting their creativity to come-up for a well-balanced menu. It also provides a medium for the nutrition student and

his/her patient to interact through monitoring patient's dietary consumption and giving feedbacks. This also contains a database of food exchange list set by DOST-FNRI which allows users an easy access and query for a food item. The meal planning method also accounts the 3 macrontrients (carbohydrates, proteins, and fats), which further details the needed nutrients of an individual, aside from counting the number of calorie intake.

E. Scope and Limitations

1. Items in the food exchange list will cover food items present in the Food Exchange Lists for Meal Planning (2012 reprint)

2. The process of computing meal plan follows the standard procedure of DOST-FNRI using the 65-15-20 distribution

3. Auto-generated meal plan covers only underweight, normal, overweight or obese patients. Designing the meal plan for other diseases (i.e. diabetes, renal disease, specific food allergy, etc.) and special conditions (i.e. pregnanat, lactating, etc.) should be done manually.

F. Assumptions

1. The generated sample menu is for one person.

2. The patient records his food intake daily

3. The patient and student are face-to-face when creating the patient's profile

4. Only the student nutritionist is to be in-charge of the patient

5. The student monitors the food intake record of the patient daily

6. The student updates the height and weight of the patient during weekly meetings.

7. Patient's height must be between 122 cm to 244 cm

II. Review of Related Literature

The goal of meal planning is to create a balanced meal given the patient's condition as parameter. Height, weight, age, medical conditions, type of activity are the usual information that a dietitian needs to provide a well-suited meal plan. Taking all of these in consideration, more restrictions will rise during the process of creating the meal plan. Once all of the constraints determined, it should be followed. Thus meal planning can be considered as both constraint satisfaction problem and multi-objective optimization problem [14]

Numerous IT systems and approaches are available today for dietetic applications, and, primarily, for menu construction and dietary analysis.[15] For example, the FML-Based System[16] developed for Japanese diet, is based on the employment of tags which enable to model different parts of the fuzzy controller in a taxonomic way. FML is used to describe the knowledge base and rule base of the diet domain. The dietician first defines the nutrient facts of the collected food from food guidebook and Internet. Then, the involved subject records his/her personal information and daily meals for a constant period. Wang et al. [17] ised FML to describe the knowledge base and the rule base of the constructed fuzzy logic system. Additionally, ontology stores the necessary knowledge base and rule base of the adopted fuzzy inference mechanism. The proposed mechanism operates as follows: (i) Dieticians first define the nutrient facts of the collected food items. (ii) The involved subjects record their personal information, eaten food items, and daily physical activity. Through the constructed FML-based personal profile ontology, the estimated energy requirement and the estimated unit requirement for each food group are obtained after inference. (iii) Finally, one-day dietary healthy level is inferred by the proposed fuzzy inference mechanism.

Fister et al. [18] present automatic generation of optimal eating plans for ath-

letes. The automatic generation of the eating plans is introduced as an optimization problem, where particle swarm optimization is taken as the problem solver. Inputs for the proposed particle swarm optimization algorithm are generated training plan and list of the potential meals, while the output of the algorithm represents a list of meals that should be consumed by the athletes.

Dholvitayakhun et al. [19] proposed an efficient local search (LS) technique for solving optimal assignment of food exchange lists problem under these constraints, energy and quantity limits. The proposed LS is improved by backtracking mechanism (BT). The objective function and its constraints were explained and conducted how to map into a real value for LS to minimize subjected to constraints Dholvitayakhun *et al* [20] proposed Modified Local Search, a basic heuristic search to generate a meal plan. The procedures of the MLS is summarized as follows: (1) Parameter initialization. (2) Generate and evaluate initial solution. (3) Set current solution as the best solution. (4) Generate and evaluate neighborhood solution. (5) Choose the one of the neighborhoods to be the best neighborhood. (6) Update the best solution by comparing the current best solution to the best neighborhood. If the best is better, set it as the best solution, otherwise no changing. (7) If the best solution has not been updated for a long time, the additional mechanism, called backtracking mechanism (BT) will be invoked to help the search algoritm escaping the trap by jumping to an visited solution before. (8) Stop searching if one of termination criterions, time limit and solution quality criteria, is met otherwise go back to step 4). The process of evaluating the solution in MLS is similar to the manual process

Mamat *et al*[21] proposed Fuzzy Linear Programming that uses Raffensperger's proposed Minimize Cost Diet Problem (MCDP) [22] model as following below:

Minimize $\sum_{j=1}^{n} c_j x_j$

Subject to $\sum_{j=1}^{n} a_{ij} x_j \geq b_i,$

$\sum_{j=1}^{n} a_{ij} x_j \leq d_i,$

$$i = 1, 2, ..., m, x_j \geq 0$$

where:

- j : food eaten per day
- c_j: amount of carbohydrate or fat nutrient,
- x_j: 100 g of food j eaten per day,
- a_{ij}: the amount of nutrient i in 100 g of food j,
- b_i: the required daily amount of nutrient i,
- d_i: the maximum daily amount of nutrient i,
- m: the number of nutrients and
- n: the number of food.

The amount of carbohydrate or fat nutrients is uncertain and it is assume as fuzzy numbers. Therefore, this fuzzy amount of nutrients is approached by linear programming with fuzzy objective coefficients[21]. $\bar{c}_j = (c_j^-/c_j/c_j^+)$ are triangular fuzzy numbers that assign the uncertain carbohydrate or fat nutrient of foods. The output covers not only the macronutrient component of the meal plan, but also it includes the micronutrients such as, Vitamins A, C and E, Calcuim, Iron, etc.

Recommender systems are needed to find food items of ones interest. El-Dosuky et al. [23] reviews recommender systems and recommendation methods, then propose a food personalization framework based on adaptive hypermedia and extend Hermes framework with food recommendation functionality. Healthy heuristics and standard food database are incorporated into the knowledgebase. Elsweiler et al. [24] incorporate nutrition into the recommender problem by examining the feasibility of algorithmically creating daily meal plans for a sample of user profiles, combined with

a diverse set of food preference data collected in a natural setting. The analyses demonstrate it is possible to recommend plans for a large percentage of users which meet the guidelines set out by international health agencies.

The Case-Based approach of Kovasznai *et al*[15] filters the collection of available food products, considering the data stored in the health records of clients or patients. Health records include information about food intolerance, allergy, and diet. Case-based reasoning solves problem by remembering previous problems that are similar to the current one. It used ripple down rules (RDR)[25] to subdivide the problem into smaller partitions which will be then compared to previous problem to find similarities [15] The system produces RDR tree which uses conditions as the node and false link as a leaf. These tree are used by the domain expert to construct the knowledge base for the system. As all the cases are bound to end at a leaf, satisfying the rules given, it will then be classified to a specific case. The solution for each case

Hsiao et. al. [26] Smart Diet Application used decision variables based on constraints of the diet to come up with the meal plan. Nutritional goal is set and is represented by

$$\begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} e_{11} & \cdots & e_{1m} \\ \vdots & \ddots & \vdots \\ e_{n1} & \cdots & e_{nm} \end{bmatrix}$$

where f_i denotes the i-th available food in the area adjacent to user's current location, and e_{ij} denotes the j-th nutrition element in food f_i. Budget goal tries to keep the cost to a minimum and meal favor goal ranks the food item to address user preference

Tom et al. [27] proposed DIligenS - Dietary Intelligence System to provide ubiquitous intelligence for automating the diet formulation and monitoring. DIligenS has a seven layered architecture consisting of knowledge base and inference engine, and necessary database. The meals scheduling, and diet formulation including consumption of food away from home is to be carried out within many constraints of budget,

time, health, and individual preferences. This study investigates the applicability of Fuzzy Multicriteria Decision Making (FCDM) following the Prioritised Fuzzy Constraint Satisfaction Approach to computerise the inherently imprecise and vague user preferences for meals selection

Different approaches can be used together to develop a much efficient way to solve a constraint satisfaction problem [14]. There are many different ways to automate the meal plan generation. All of it uses constraints as a guide to find the well-suited answer. But meal plans are not exact, meaning solutions should vary from time to time. The rules may or may not changed depending on the patient or the domain expert.

III. Theoretical Framework

A. Calculating Diets

Estimate the desirable body weight(DBW), sometimes referred to as reference, ideal or standard body weight is the primary procedure to process already given data. This can be done in four ways[11]:

A. Use of Standard Tables-FNRI Standard Weight for Filipinos

B. Tanhauser's(Broca) Method

C. Use the derived formula based on Body Mass Index

D. Use NDAP formula which gis close approximation of desitable BMI as well as midpoint of FNRI's range of reference weights

Second is to determine the total energy allowance by multiplying the DBW with the following values, according to:

Table 1: Corresponding Energy Allowance Rate Base on Type of Activity

Activity	kcal/kg DBW/day
Bed rest but mobile (hospital patient)	27.5
Sedentary (mostly sitting)	30.0
Light (tailor, nurse, physician, jeepney driver)	35.0
Moderate (carpenter, painter, heavy housework)	40.0
Very Active (swimming, lumberman)	45.0

Last, determine the carbohydrate (CHO), protein (PRO), and fat by:

A. Percentage distribution

Carbohydrates: 55-70 % of TEA

Proteins: 10-15 % of TEA

Fats: 20-30 % of TEA

B. Calculate number of grams of CHO, PRO, FAT by dividing the calories for each nutrient by the corresponding physiological value (4 kcal for CHO and PRO and 9 kcal for FAT per gram). It final result will be diet prescription

B. Designing Meal Pattern and Menu

Using the diet prescription, the following procedures are used to translate it into food exchanges (see Table 2)

Table 2: Composition of Food Exchanges

List	Food	Measure	CHO(g)	PRO(g)	FAT(g)	Energy(kcal)
I.A	Veg. A	1 cup raw	-	-	-	-
	Veg. A	2 cups raw	3	1	-	16
I.B	Veg. B	$\frac{1}{2}$ cup raw	3	1	-	16
II	Fruit	varies	10	-	-	40
III	Milk(Whole)	varies	12	8	10	170
	Milk(Low Fat)	4 tbsp	12	8	5	125
	Milk(Skimmed)	varies	12	8	tr	80
IV	Rice	varies	23	2	-	100
V	Meat(Low Fat)	varies	-	8	1	41
	Meat(Med. Fat)	varies	-	8	6	86
	Milk(High Fat)	varies	-	8	10	122
VI	Fat	1 tsp	-	-	5	45
VII	Sugar	1 tsp	5	-	-	20

1. List all the foods furnishing carbohydrates with the exceptio of rice

   (a) It is desirable to allow 2-3 exchanges of List 1. A & B vegetable per day

   (b) Unless there is need for drastic restriction of simple carbohydrates 3-4 exchanges of fruits per day are reasonable allowance

   (c) The amount and type of milk allowed depends upon th patient's needs, food habits and other economic considerations

   (d) Allow 5-9 teaspoons of sugar per dat unless contra-indicated

2. To determine how many rice exchanges:

   (a) Add the CHO from vegetables, fruit, milk, sugar

   (b) Subtract this sum from prescribed CHO

   (c) Divide the difference by 23 (g CHO furnished by 1 rice exchange)

   (d) The nearest whole quotient is the number of rice exchange allowed

3. To determine how many meat and fish exchanges are allowed:]

   (a) Add the PRO furnished by the food groups already listed

   (b) Subtract this sum from the prescribed PRO

   (c) Divide the difference by 8 (g PRO per meat and fish exchange)

   (d) The nearest whole quotient is the number if meat and fish exchange allowed

   (e) When calculating grams of fat per meat and fish exchange, use the fat value that best represents the patuent's usual intake. There is no need to add or subtract fat exchanges if the appropriate meat and fish category is used in the computatation of food allowance

4. Follow the same procedure for fat, using 5 as the divisor since one fat exchange contains 5 g of fat

An allowance of ± 5 grams the prescribed amount for PRO, CHO and FAT and ± 50 kcal for energy are given so the fractions of servings are avoided

Distribute the food allowance into breakfast, lunce supper and snacks, depending on the patient's eating habits

Figure 2: Sample meal plan and menu

1500 kcal 245-55-35			
Vegetable A: 2 exchanges		Sugar: 5 exchanges	
Vegetable B: 1 exchange		Rice: 7 exchanges	
Fruit: 4 exchanges		Meat and Fish: 4 exchanges	
Milk: 1 exchange		Fat: 3 exchanges	
Food Exchanges	No. of Exchanges	Sample Menu	Approximate Size per Serving
Breakfast			
Fruit	1	Ripe Papaya	1 slice or $\frac{3}{4}$ cup
Meat and Fish	1	Baked Hm Sausage	3 (9 cm dia x .3 cm thick)
Rice or substitute	1	Pan Amerikano	2 slices
Milk for Coffee	$\frac{1}{4}$	Evaporated Milk	2 tbsp
Sugar	2	White Sugar	2 tsp
Mid-A.M. Snack			
Meat and Fish or substitute	1	Cottage Cheese	$\frac{1}{3}$ cup
Rice or substitute	1	Pan de Limon	1 piece
Lunch			
Soup		Clear broth from Chicken Tinola	
Meat and Fish	1	Chicken Tinola	1 small leg
Vegetable A	2	Green Papaya and Sili Leaves	1 cup
Rice	$1\frac{1}{2}$	Boiled Rice	$\frac{3}{4}$ cup
Fruit	1	Watermelon	1 slice
Fat	1	Cooking Oil for Tinola	2 tsp
Mid-P.M. Snack			
Rice or substitute	1	Pan de Monay	1 piece
Fat	1	Cream Cheese	1 tbsp
Fruit	1	Pineapple Juice	$\frac{1}{3}$ cup (undiluted)
Sugar	2	White Sugar	1 tsp
Dinner			
Soup		Parseleyed Beef Broth	
Meat and Fish	1	Broiled Bangus w/ Kalamansi	1 slice
Vegetable B	1	Sauteed Squash	$\frac{1}{2}$ cup
Rice	$1\frac{1}{2}$	Boiled Rice	$\frac{3}{4}$ cup
Fruit	1	Lakatan	1 piece
Fat	1	Cooking Oil for Squash	1 tsp
Bed-Time Snack			
Rice or substitute	1	Galyetas de Patatas	10 pcs.
Milk	$\frac{3}{4}$	Evaporated Milk	$\frac{1}{3}$ cum (undiluted)
Sugar	2	White Sugar	2 tsp

C. Patient Portal

Electronic patient portals have received increasing interest with meaningful Use criteria and the move toward patient-centered medical homes.Portals typically allow patients and physicians to send electronic messages to one another in a secure environment.[28] Other functions might include renewing prescriptions, making appointments, and viewing portionsof ones own medical record. Despite potential challenges and need for revised workflow, there are scant data regarding attitudes of faculty and resident physicians at academic medical centers toward electronic patient portals integrated with an electronic medical record (EMR).[29]

D. Information System

An information system can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization. In addition to supporting decision making, coordination, and control, information systems may also help managers and workers analyze problems, visualize complex subjects, and create new products. [30]

Information system has been defined in terms of two perspectives: one relating to its function; the other relating to its structure. From a functional perspective; an information system is a technologically implemented medium for the purpose of recording, storing, and disseminating linguistic expressions as well as for the supporting of inference making. From a structural perspective; an information system consists of a collection of people, processes, data, models, technology and partly formalized language, forming a cohesive structure which serves some organizational purpose or function [31]

Three activities in an information system produce the information that organizations need to make decisions, control operations, analyze problems, and create

new products or services. These activities are input, processing, and output. Input captures or collects raw data from within the organization or from its external environment. Processing converts this raw input into a more meaningful form. Output transfers the processed information to the people who will use it or to the activities for which it will be used. Information systems also require feedback, which is output that is returned to appropriate members of the organization to help them evaluate or correct the input stage. [32]

E. Model-View-Controller

Model-View-Controller (MVC) pattern is used to separate application's concerns. Model represents an object carrying data. It can also have logic to update controller if its data changes. View represents the visualization of the data that model contains. Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate. [33] MVC pattern doesnt only used in component construction, but also it used in large-scale object-oriented system design, such as e-commerce system. [34]

F. Structured Query Language

SQL or standard query language is a standardized query language for requesting information from database. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. The scope of SQL includes data insert, query, update and delete, schema creation and modication, and data access control. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL Server use SQL as standard database language. [35] SQL is commonly used for Web database development and management. Though SQL is now considered to be a standard language, there are still a number of variations of it, such as mSQL and mySQL. By using a scripting language like PHP, SQL

commands can be executed when a Web page loads. This makes it possible to create dynamic Web pages that can display different information each time they load. [35]

IV. Design and Implementation

From the problem definition, you identify key techniques and approaches you feel are necessary to successfully find a resolution. You can further motivate the approaches by outlining how they were used in other contexts. This is an extension of the Literature review but with emphasis on how you intend to apply the methodology/ies to your problem. The ERDs, DFDs and technical architecture are placed here.

A. Use Case Diagram

Figure 3 shows the use case diagram of the system. It has 4 actors and 14 use cases. The four actors are admin, student, teacher, patient. The use cases are add teacher, add student, add patient, log-in, add food item, generate patient record statistics, create meal plan, create sample menu, update patient record, view patient's food intake, give feedback, check food item sustitute, view feedback, view sample menu, and aprrove meal plan and menu. All use cases are mutually exclusive except for the log-in. the Data Flow Diagram illustrates how use cases work together. .

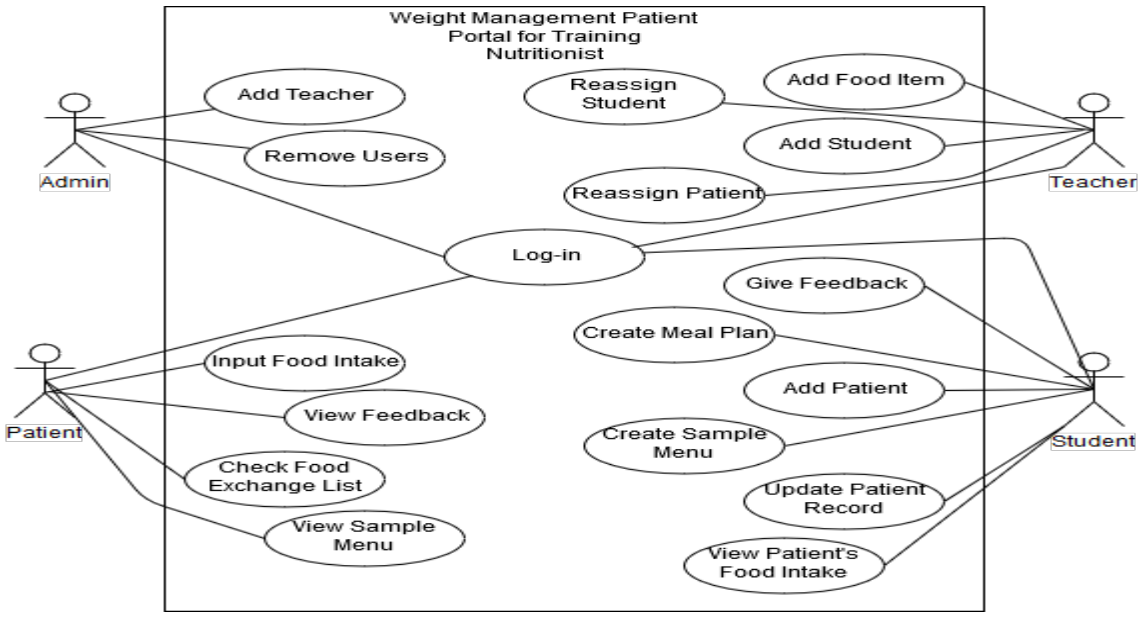

Figure 3: Use-Case Diagram, Weight Management Patient Portal for Filipinos and Training Nutritionist

B. Entity Relationship Diagram

Figure 4 iluustrates the relationship of all the roles in the system. It shows *Teacher, Student, Patient* dependency from a superior entity. *Weekly Report* contains the weekly changes on the patient.*Meal Plan* contains meal plan for the week of the patient and the *Sample Menu* is menu pattern for a specific day on that week

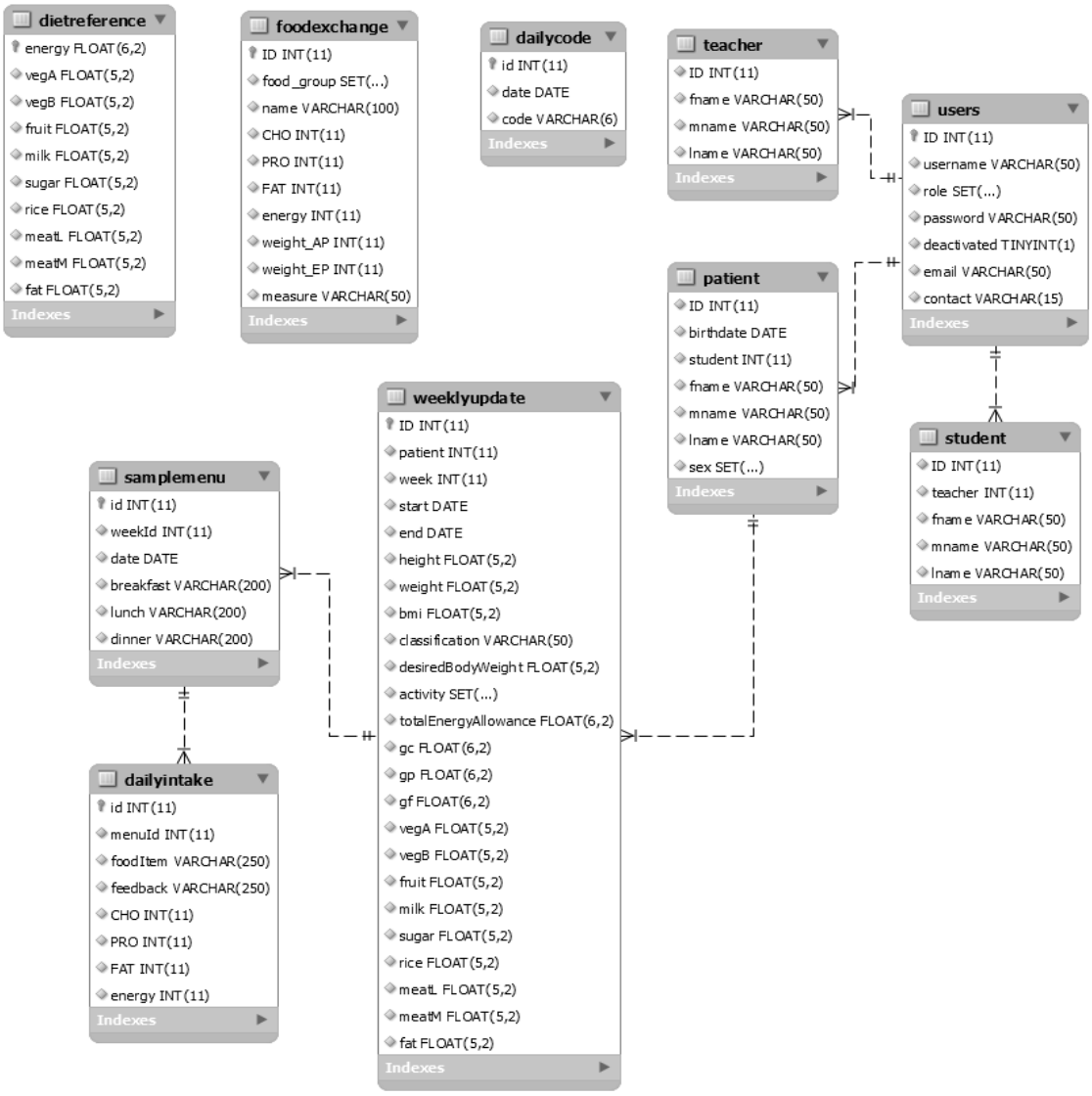

Figure 4: Entity Relationship Diagram Focused on the Ralationship of All Users

C. Class Diagram

Figure 5 shows the inheritance of different food groups and its specialized components

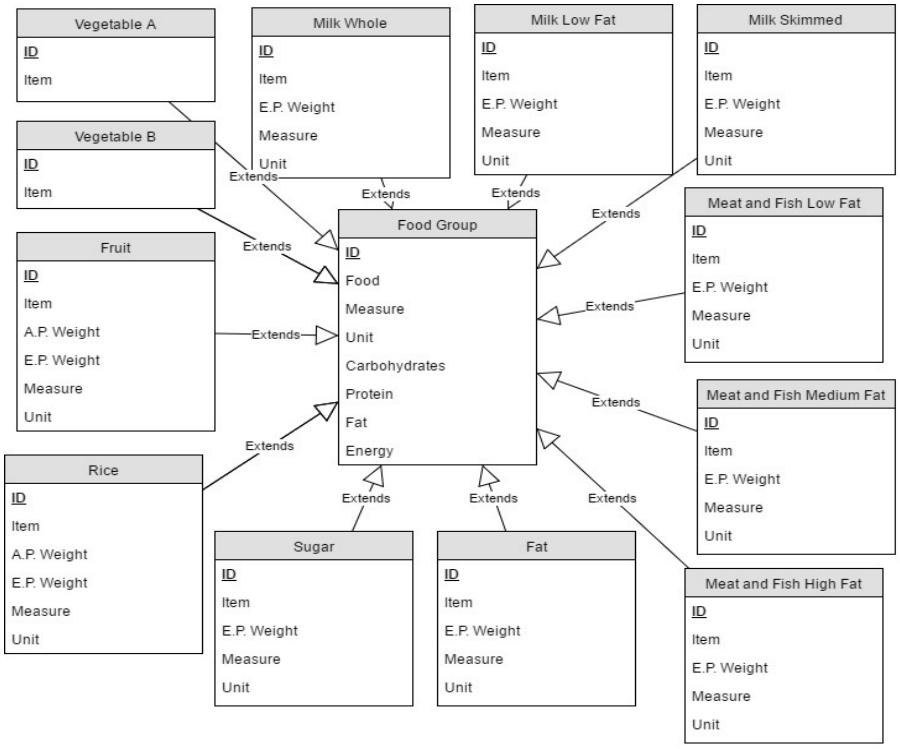

Figure 5: Class Diagram for Food Groups

D. Data Flow Diagram

Figure 6 illustrates the overview log-in process. It has 4 external entities, the admin, the teacher, ths student and the patient, which will be redirected to their corresponding pages.

Figure 7 focuses on the process can be done by the sytem administrator. It has add teacher and remove user as its processes, which access teacher, student and patient database.

Figure 8 shows the add student, reassign student or patient, and add food item processes which can be done by the student nutritionist. These processes access student, patient and food exchange database.

In figure 9, the diagram focuses on the capabilities of the student nutritionist.

It shows that he/she can add student, update height and weight of the patient and generate meal plan on a weekly basis, create sample menu, view patient's daily intake and givefeedback to it.

The figure 10 illustrates the patient's fuctionalities which includes veiw the sample menu, input daily food intake, view nutritionist's feedback on his intake and check a food item nutritional value in the food exchange list.

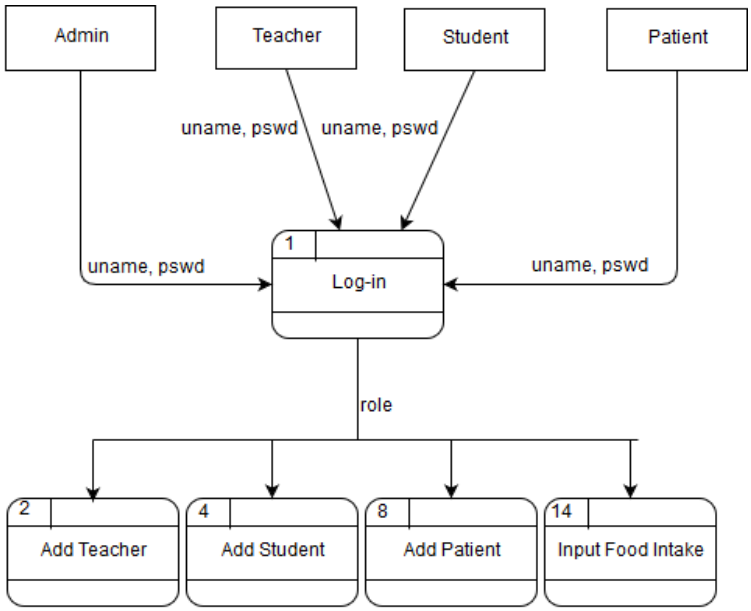

Figure 6: Level 1 Data Flow Diagram of the System Focusing on Log-in

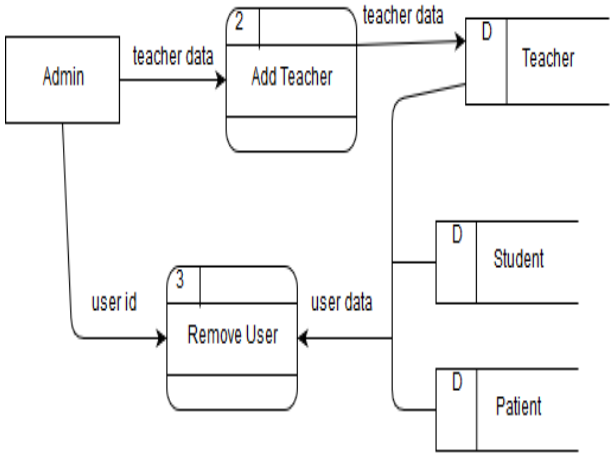

Figure 7: Level 1 Data Flow Diagram of the System Focusing on Administrator

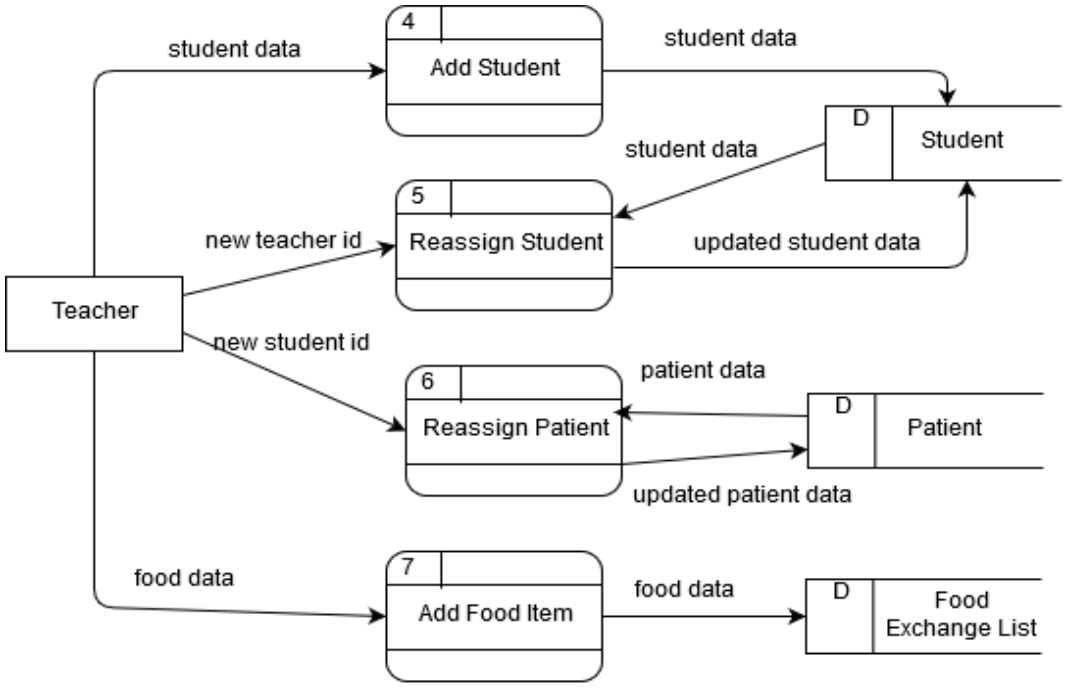

Figure 8: Level 1 Data Flow Diagram of the System Focusing on Teacher Dietitian

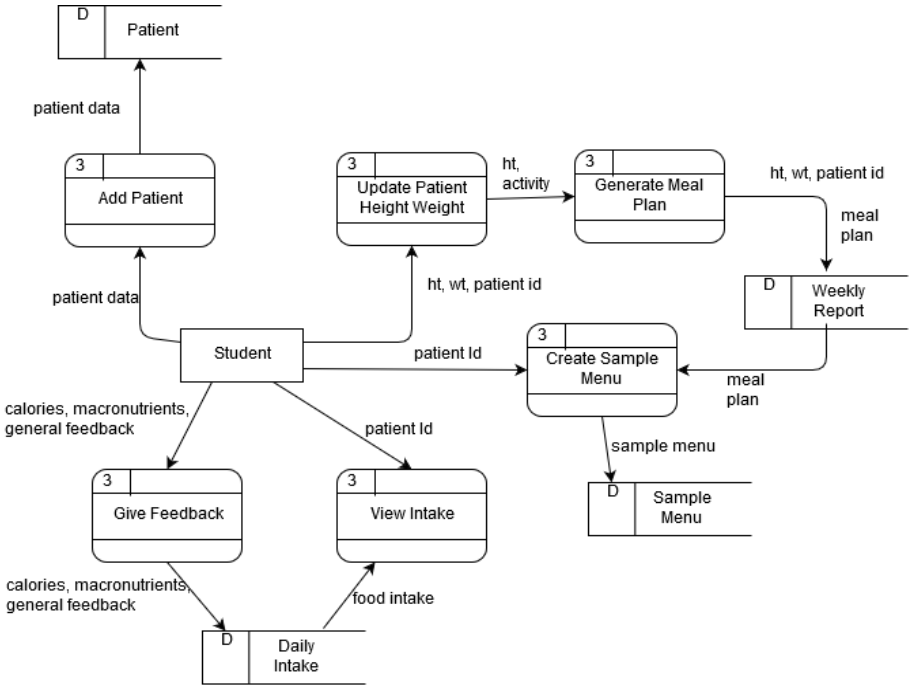

Figure 9: Level 1 Data Flow Diagram of the System Focusing on Student Nutritionist

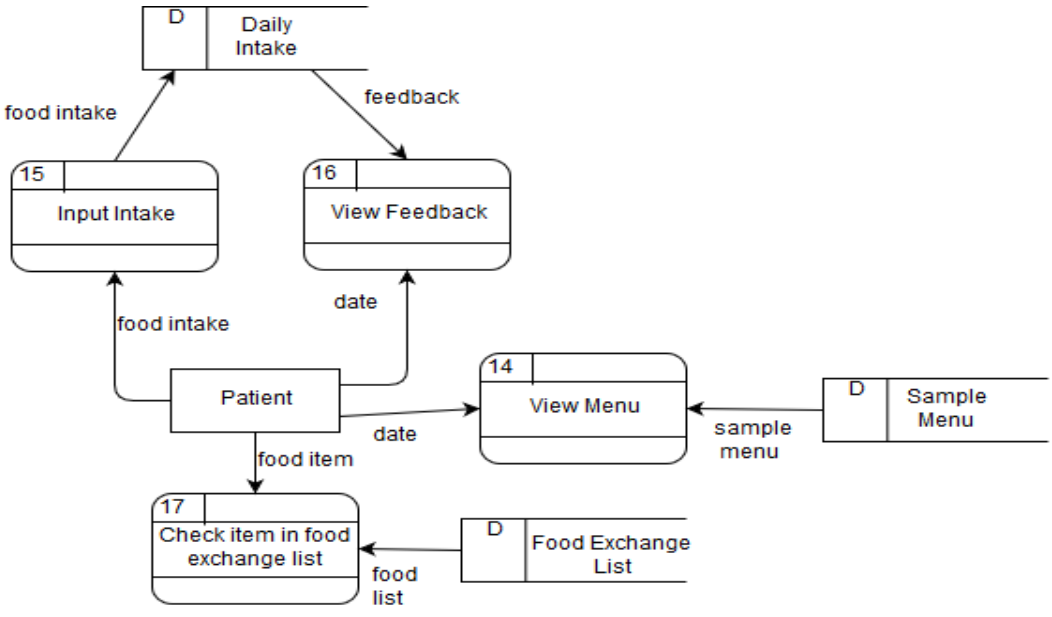

Figure 10: Level 1 Data Flow Diagram of the System Focusing on Patient

E. System Architecture

Weight Management Patient Portal for Filipinos and Training Nutritionist will be imlemented in Java Spring framework that uses the MVC(Model-View-Control) architecture. It will run on a Tomcat-7 server and requires Java 1.7 or later The part visible to the users is the system views and JSP, HTML, CSS, jQuery and Javascript is used for the interface. The system interacts with the MySQL database using models. The controller is the bridge between the model and the views.

F. Technical Architecture

Hardware Requirements for the server:

1. PC with 1 GHz processor or higher
2. Up to 2 GB of free disk space
3. 512 GB of RAM or higher

Hardware Requirements for the client:

1. PC with Intel Pentium 4 or later
2. 100 MB of free disk space
3. 128 MB of RAM or higher

V. Results

The system that has been developed to serve as portal for student nutritionists, patients, and teacher dietitians. The portal will serve as a web based comminication medium for the students to interact with their own patients, as well as keeping records of the patient's intake, nutritional status through out the case study. Only the log-in page will be accessible to all users, while the rest will be role specific.

The log-in page as shown in figure 11 requires the username and the password of the user. Once log-in he or she will be redirected to their corresponding default pages according to their roles.

If a user forgets his or her password, he or she may click the forget password link which will redirect to a page with a form requiring him or her to enter his or her email address already present in the users database. This form can be seen on figure 12. Once the email is entered, it will send an email with the new password.

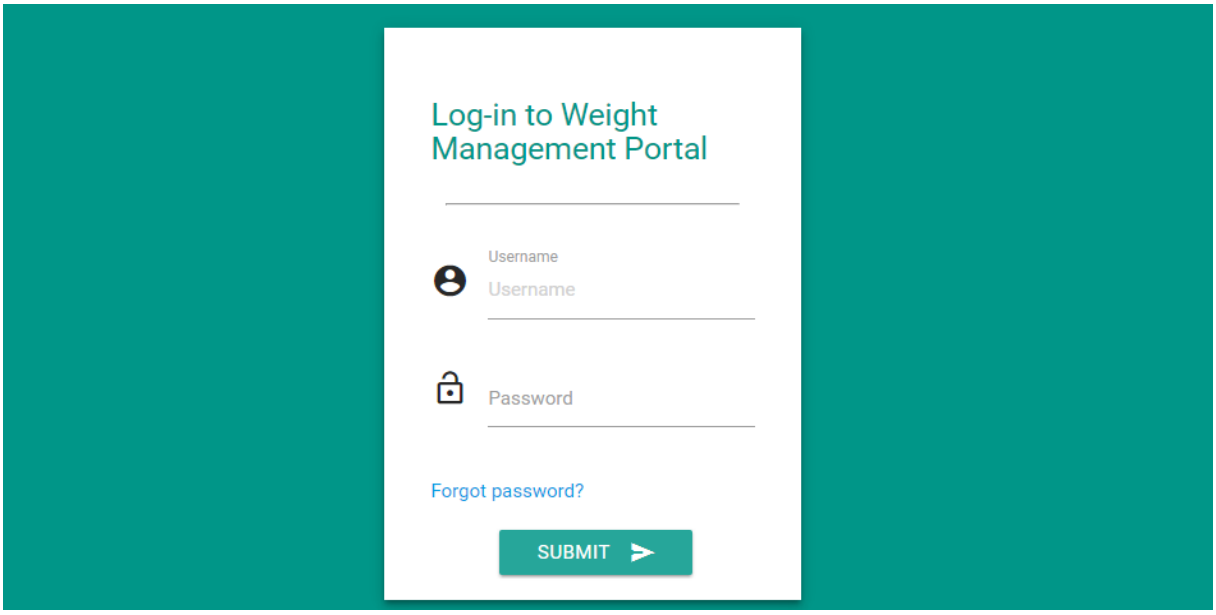

Figure 11: Log-in Page for users

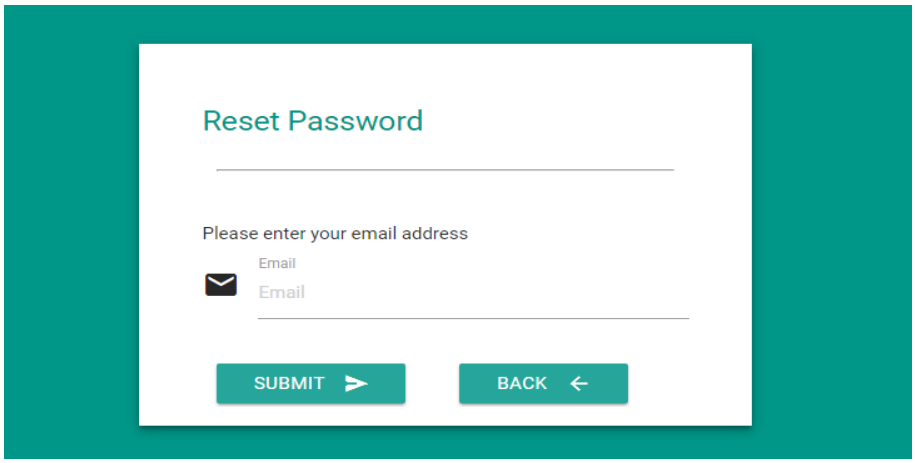

Figure 12: Reset Password Form

Once log-in, the system administrator will be redirected to add teacher's page, as seen on figure 13. On the admin's sidebar, he can click the links to teacher list, student list and patient list. The user can log-out by clicking the arrow inside a square icon. In the teacher list, figure 14, student list, figure 15, and patient list, figure 16, enables the administrator to deactivate one's account.

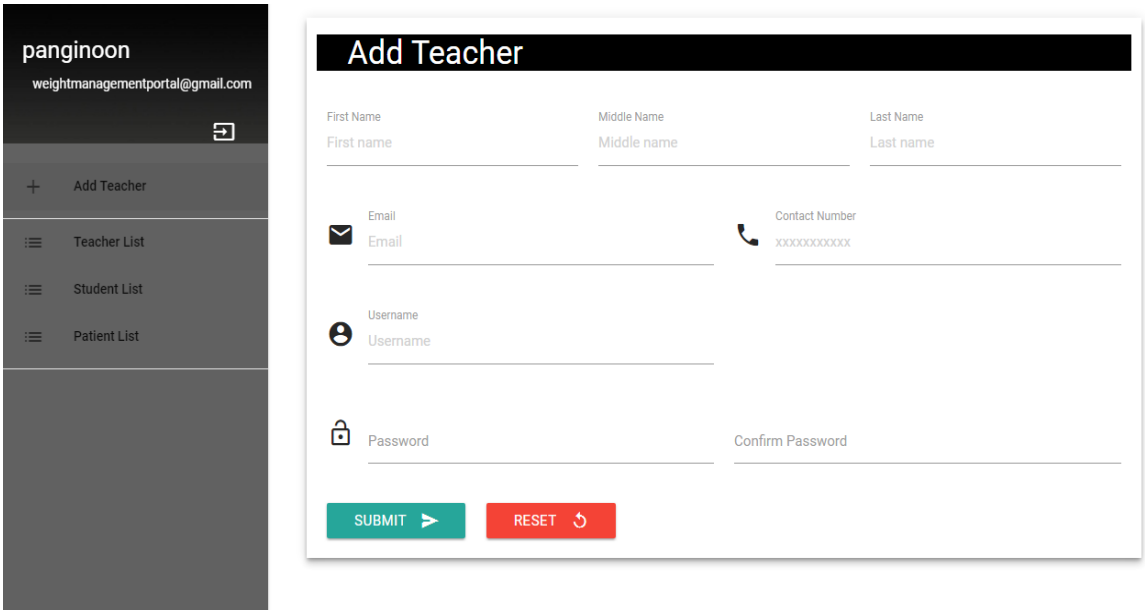

Figure 13: Add Teacher Form

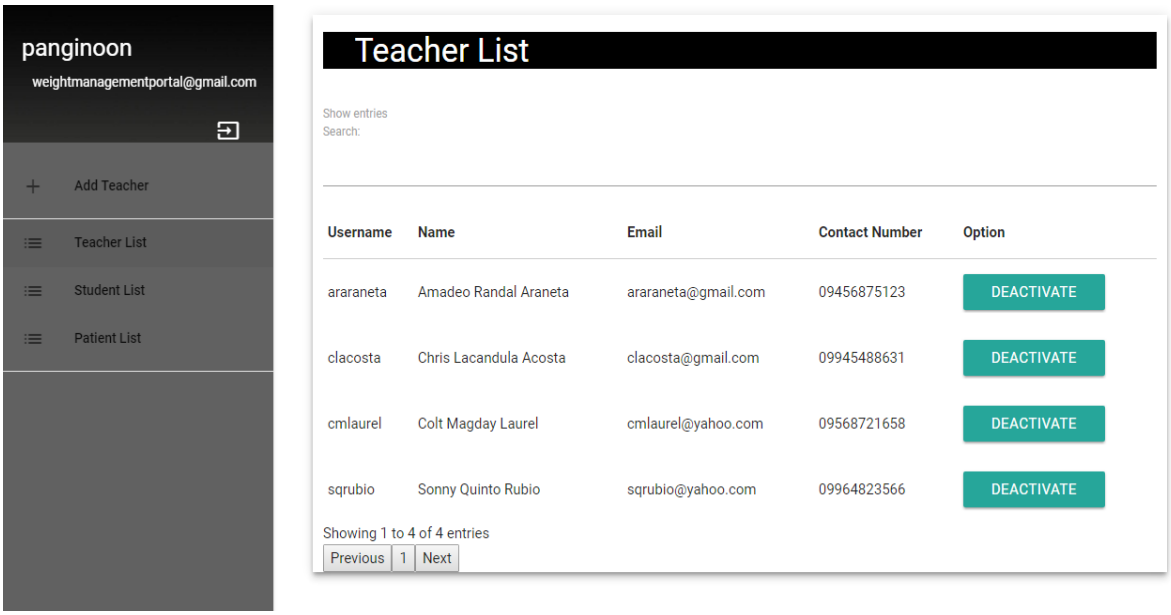

Figure 14: Teacher List Page

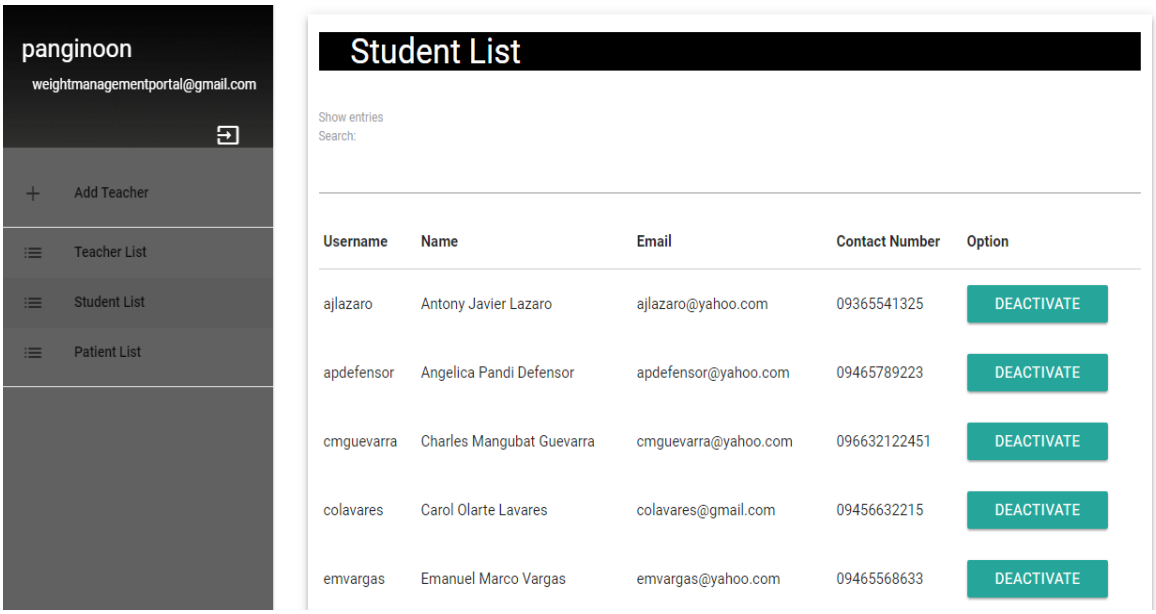

Figure 15: Student List Page

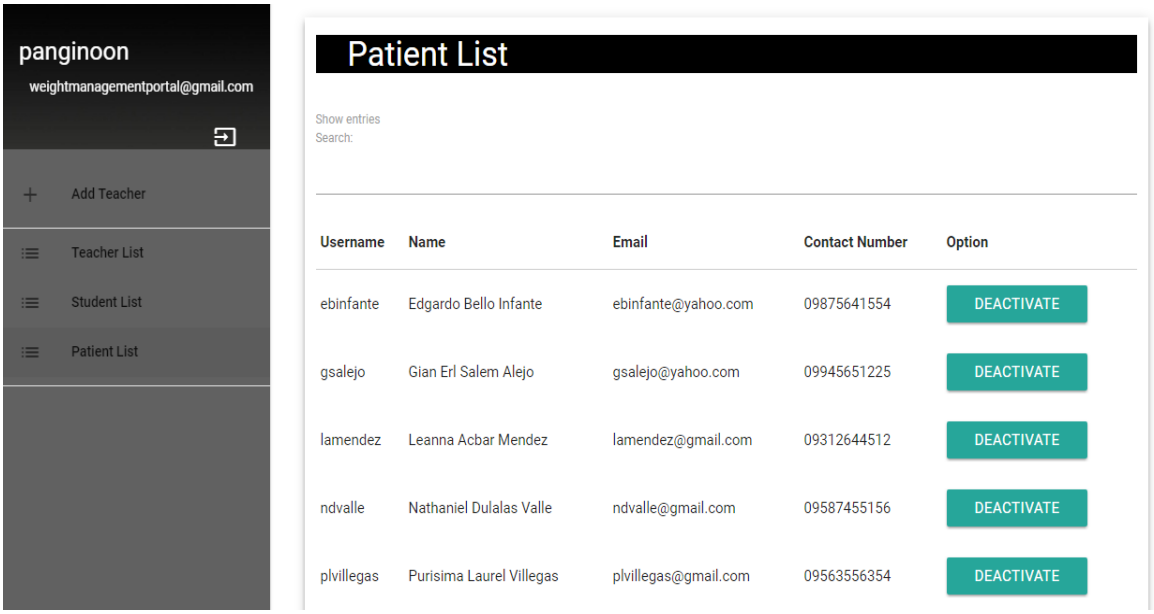

Figure 16: Patient List Page

The teacher dietitian will be redirected to add student page as show in figure 17. On his sidebar he can click on his username to edit his profile, which includes username, email, contact number and password, seen in figure 18. On the middle part of the sidebar he can navigate to reassign student, reassign patient, add food item to food exchange list and at the bottom most part he can see the daily code which act as the approval key for student's meal plan, sample menu and feedback.

The teacher can reassign the student or patient by changing the designated teacher or student respectively, (see figure 19 and figure 20). The teacher can also add new food item to the food exchange list, as shown in figure 21, which can be seen by a patient.

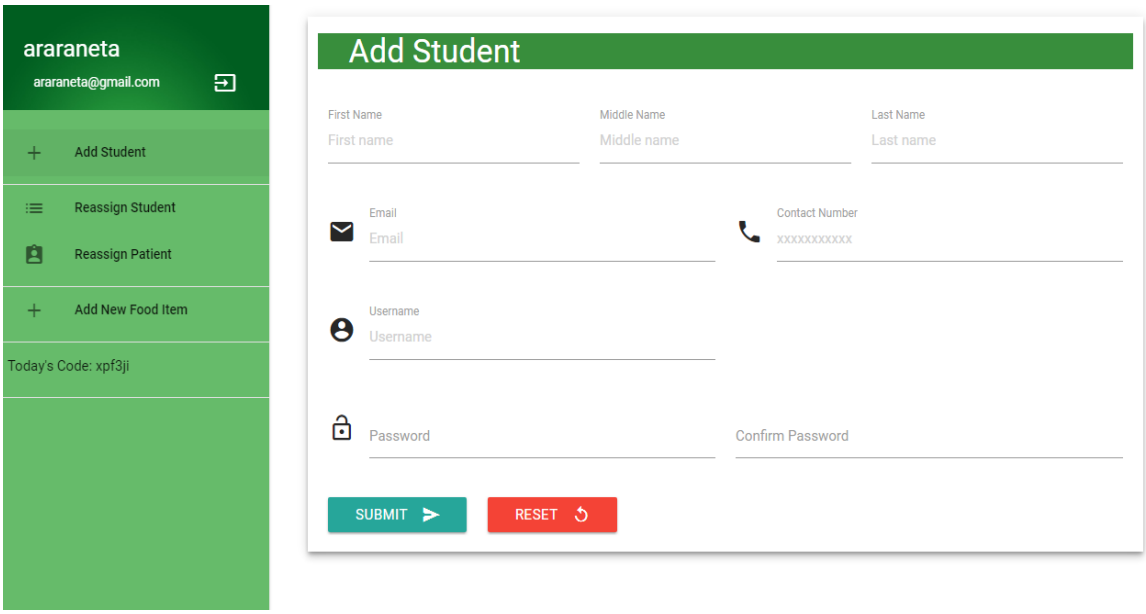

Figure 17: Add Student Form

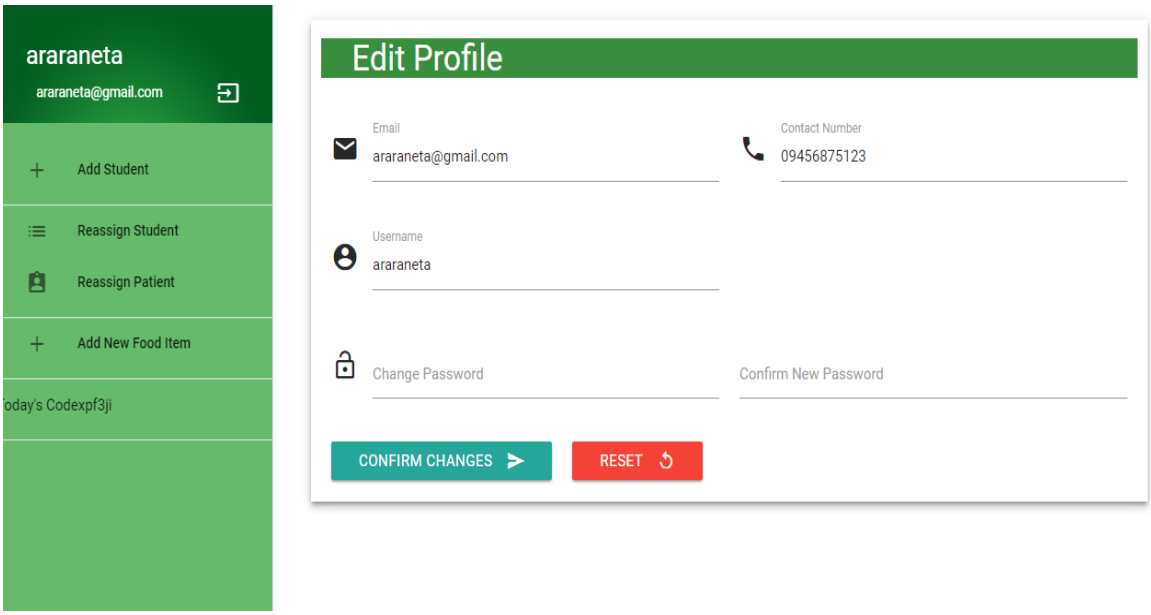

Figure 18: Teacher's Edit Profile Page

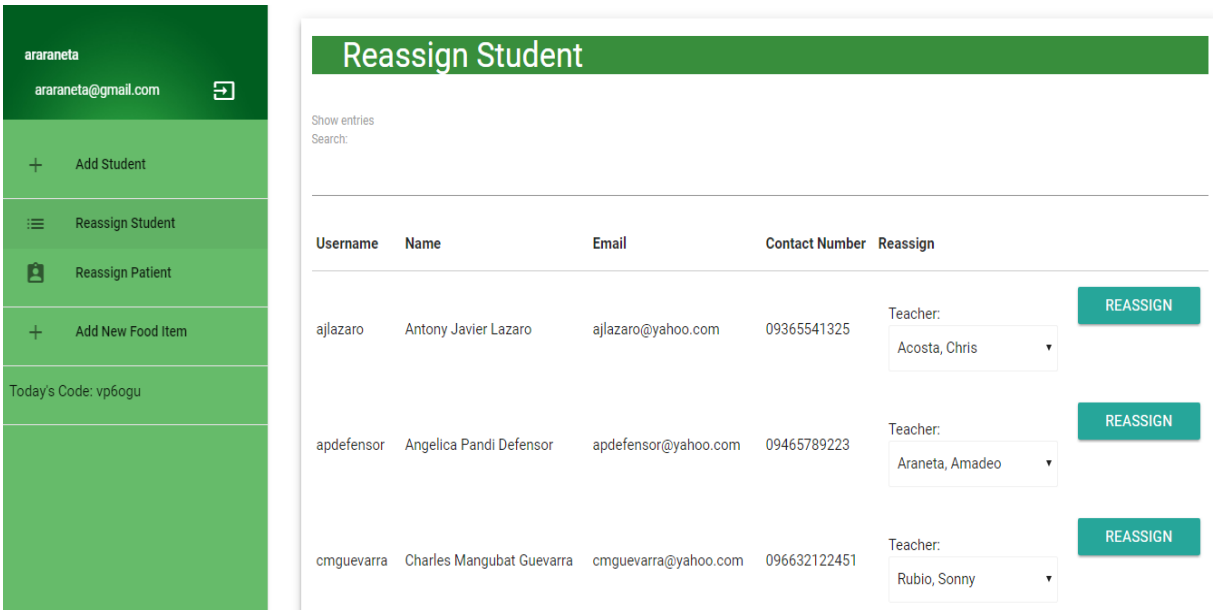

Figure 19: Reassign Student Page

Figure 20: Reassign Patient Page

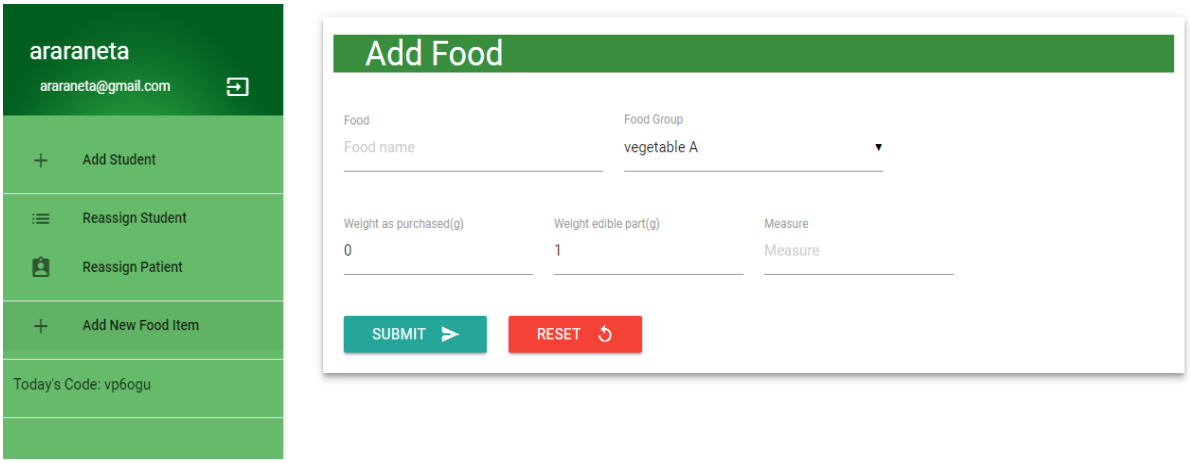

Figure 21: Reassign Patient Page

The student nutritionist user, once log-in, will be redirected to add patient page

that can be seen on figure 22. He can click on his username to edit his profile23. On the side he can click the patient list link which will show the patients designated to the logged-in student as shown on figure 24. Here he can calculate diet, create menu or view the intake of a patient. If he clicks the calculate diet link of the patient he will be redirected to calculate diet page, see figure 25. Once submitted meal plan will be automatically generated as seen on figure 26. Once the teacher approves the diet prescription, the student is now able to create a sample menu for the patient, see figure 27. This sample menu is viewable to the patient once it is approved by the teacher.

If a patient already input their daily intake, the student can then view the intakes and give feedback to the patient, which can be seen in figure 28

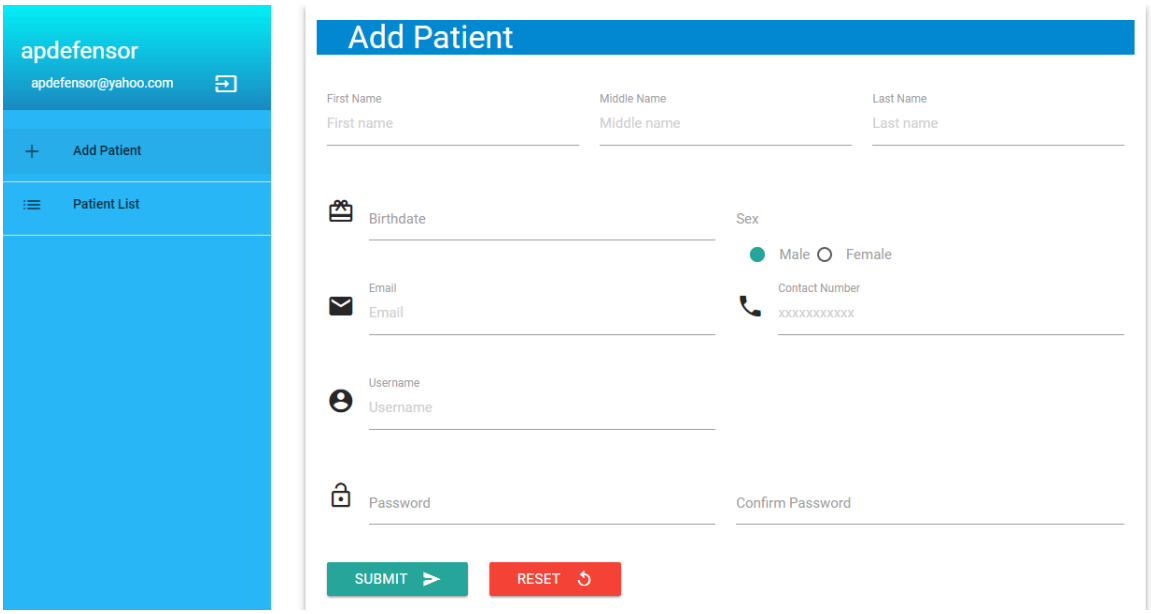

Figure 22: Add Patient Page

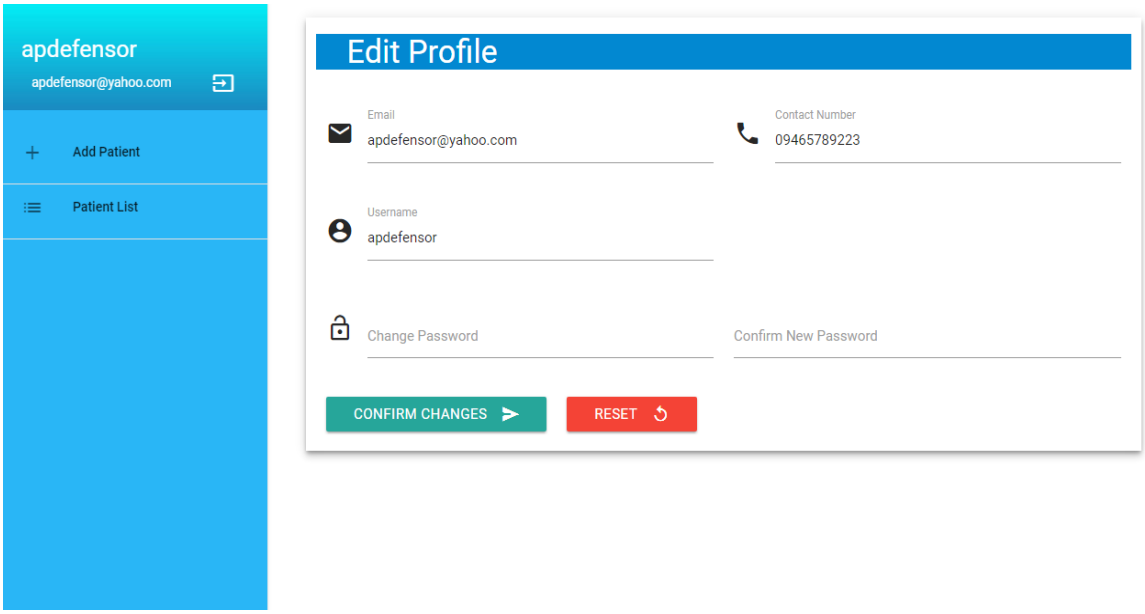

Figure 23: Student's Edit Profile Page

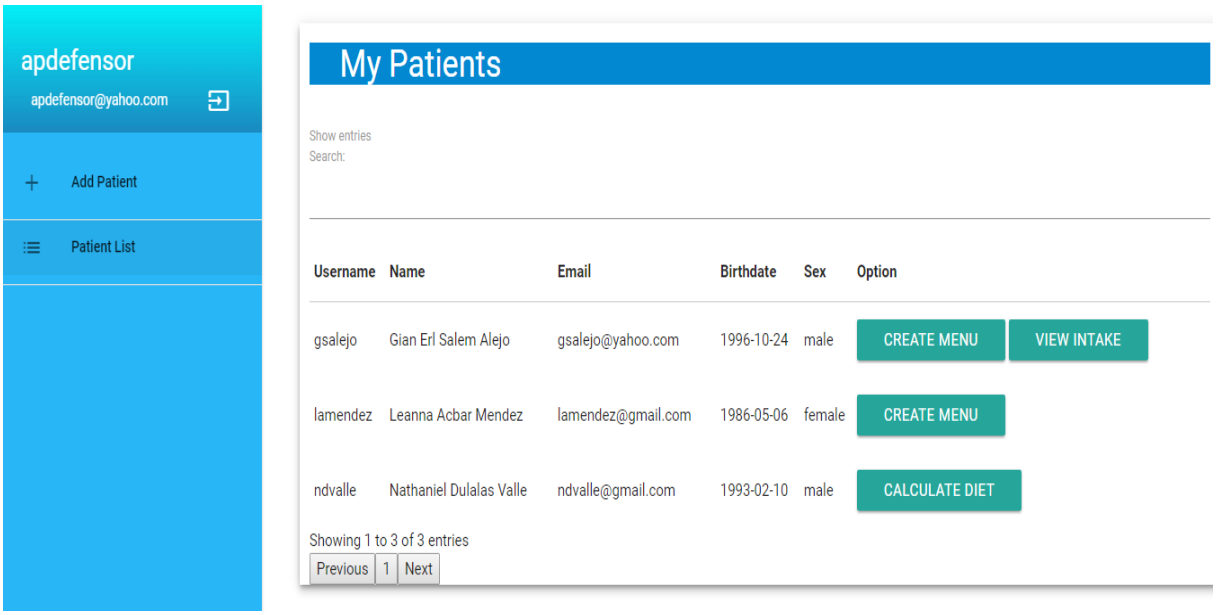

Figure 24: Student's Patient List Page

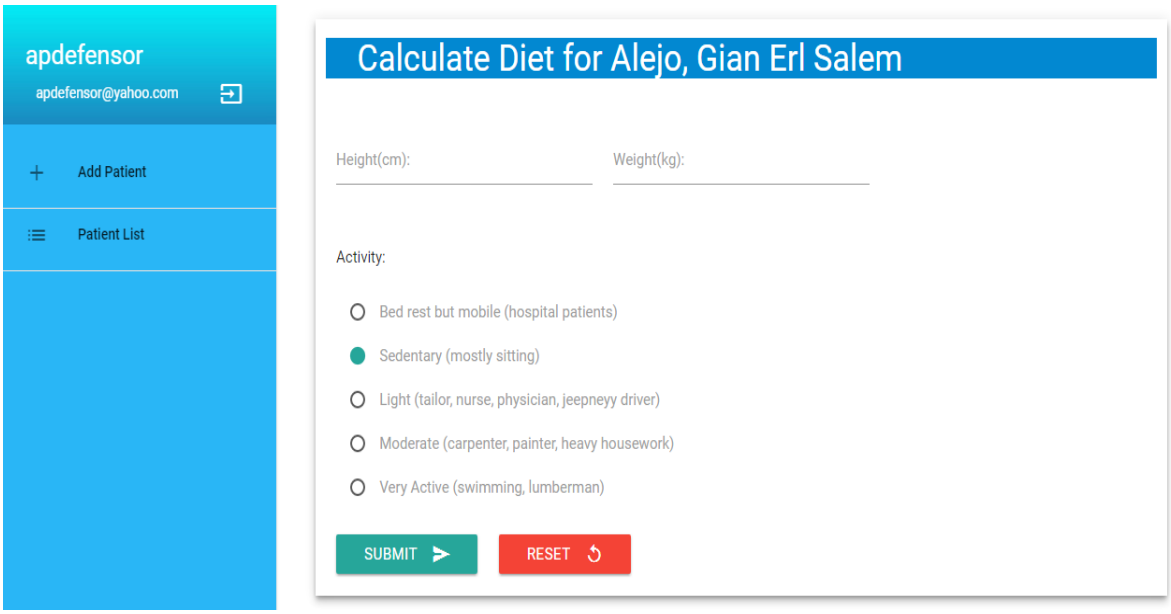

Figure 25: Calculate Diet Page

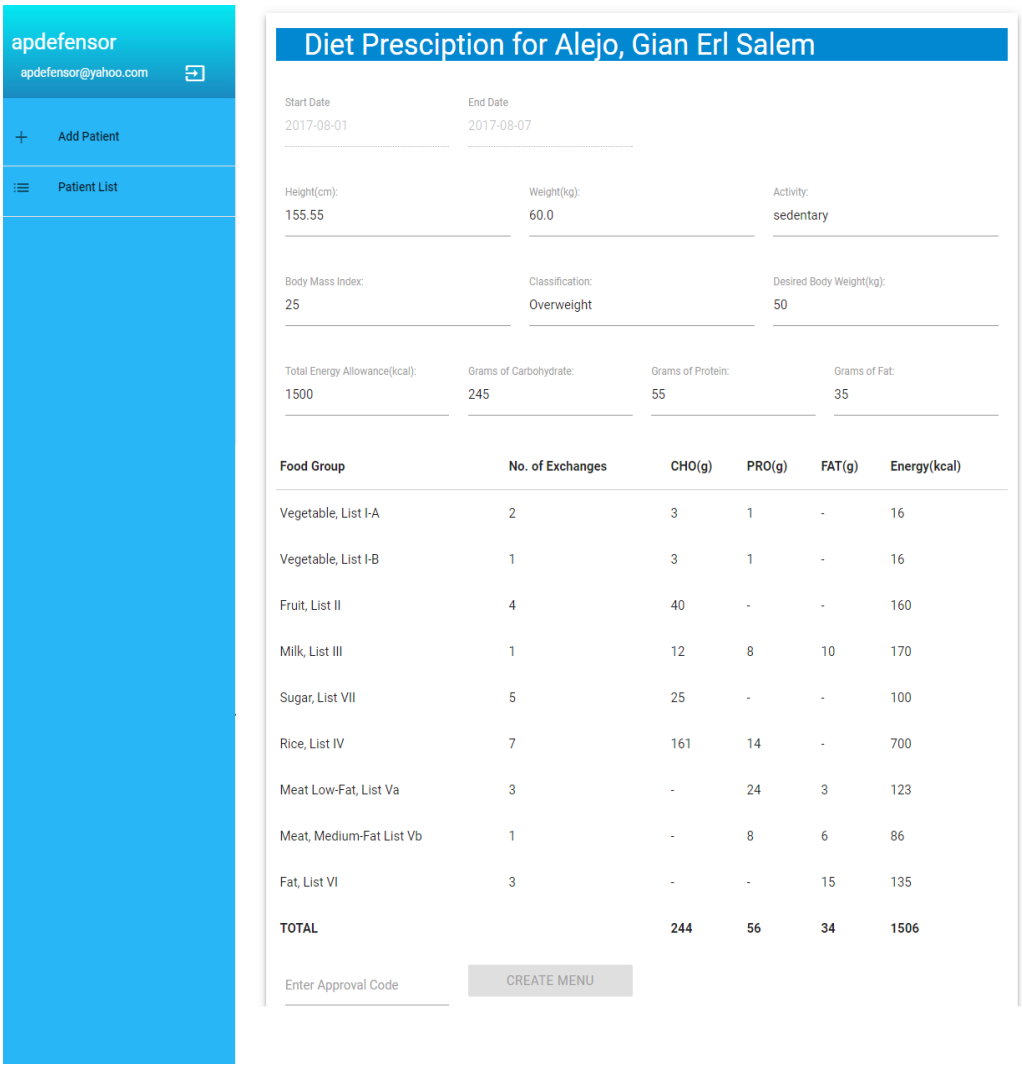

Figure 26: Diet Prescription

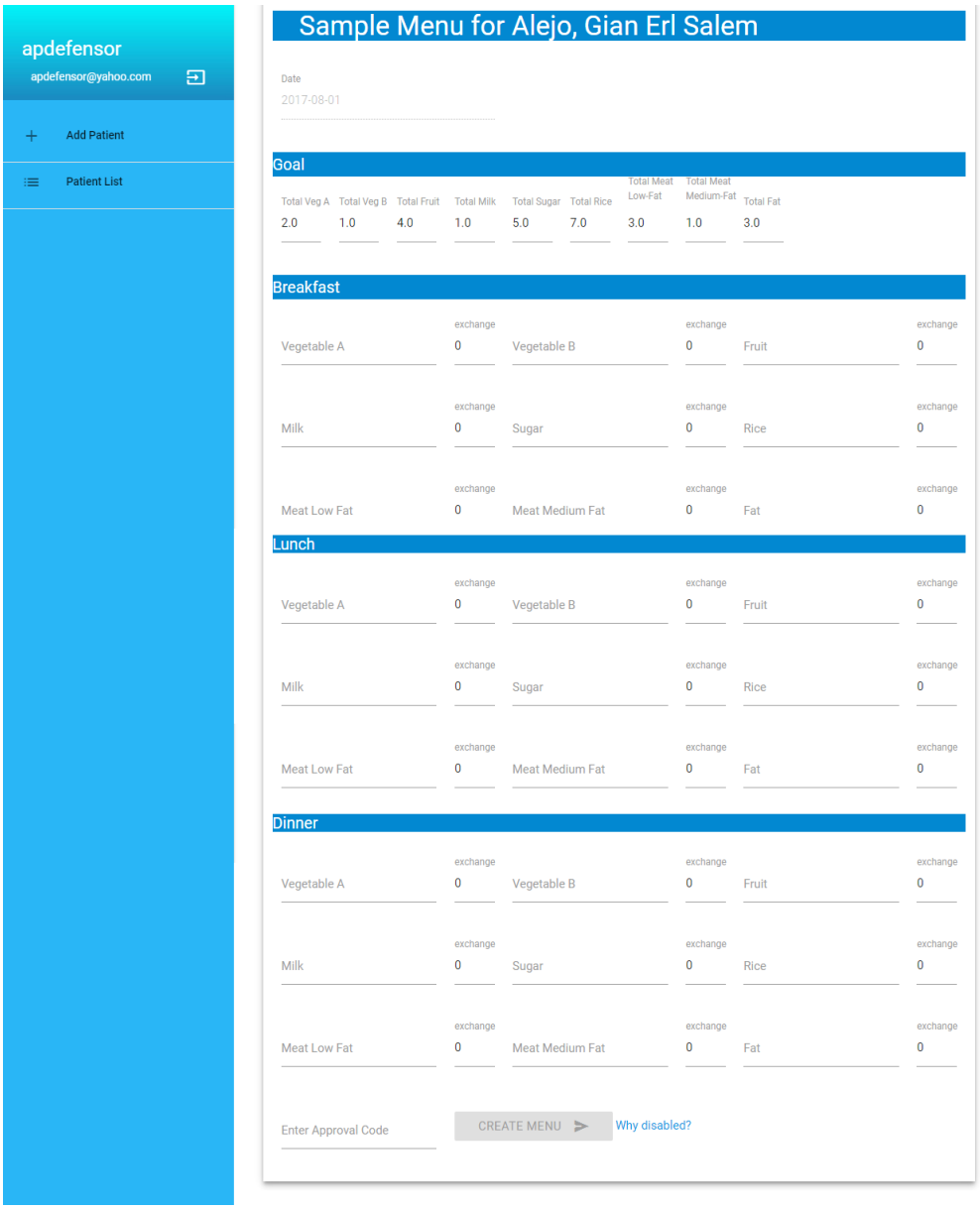

Figure 27: Sample Menu Form

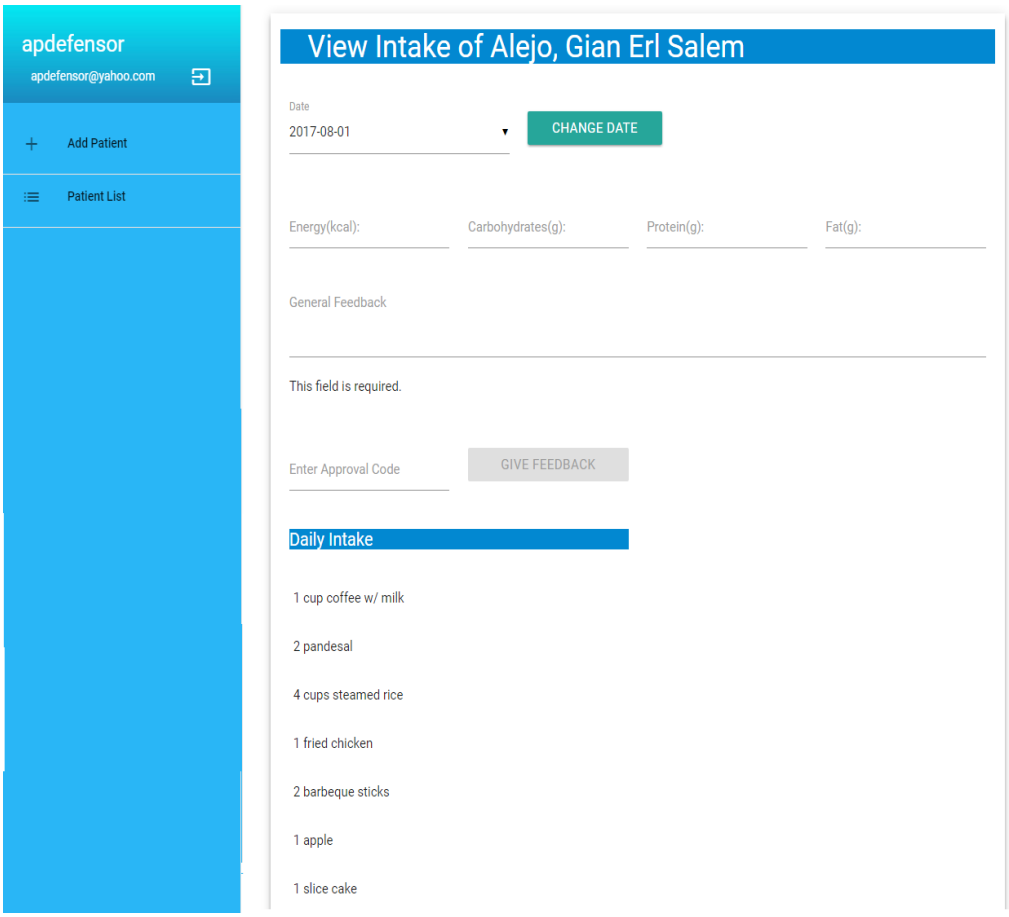

Figure 28: Feedback Form and Intake View

For a patient user, he will be redirected to add food intake page as shown on figure 29, which also shows the sample menu, as seen on 31, at the bottom of the page. He can also click on his user name to edit his own profile30. Once a feedback is given, the patient can now view it on th view feedback, see figure 32. He can change the date on it to view the coressponing sample menu, intake as well as the feedback given on that particular date. Lastly he can check the macronutrient of a food item listed in the food exchange list, shown on figure 33.

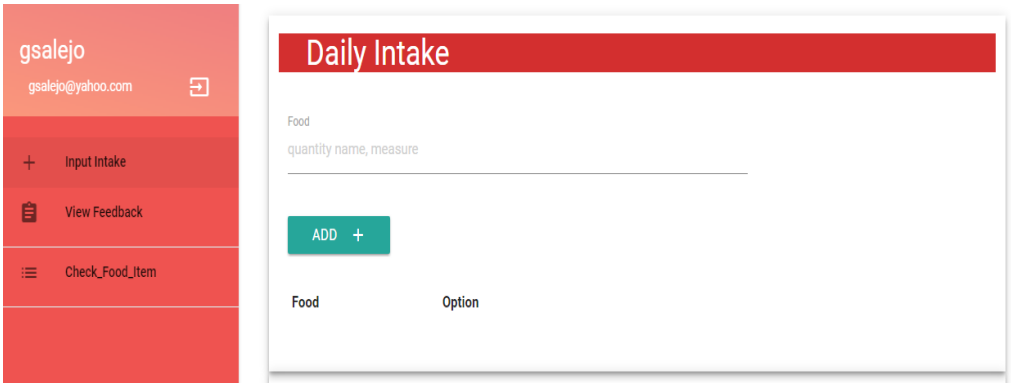

Figure 29: Daily Intake Form

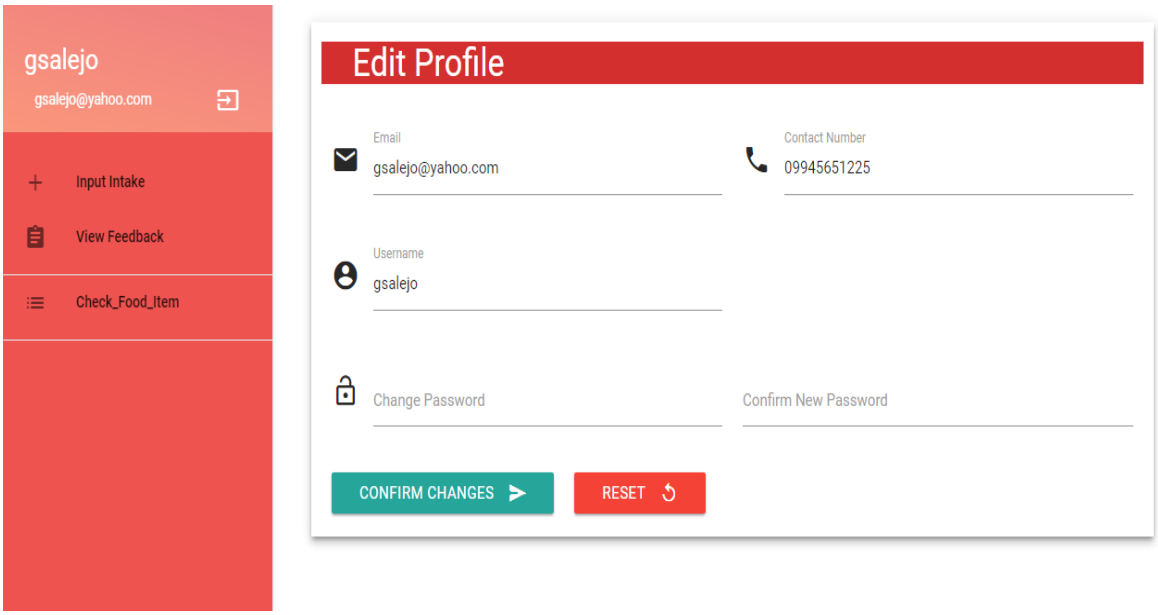

Figure 30: Patient's Edit Profile Page

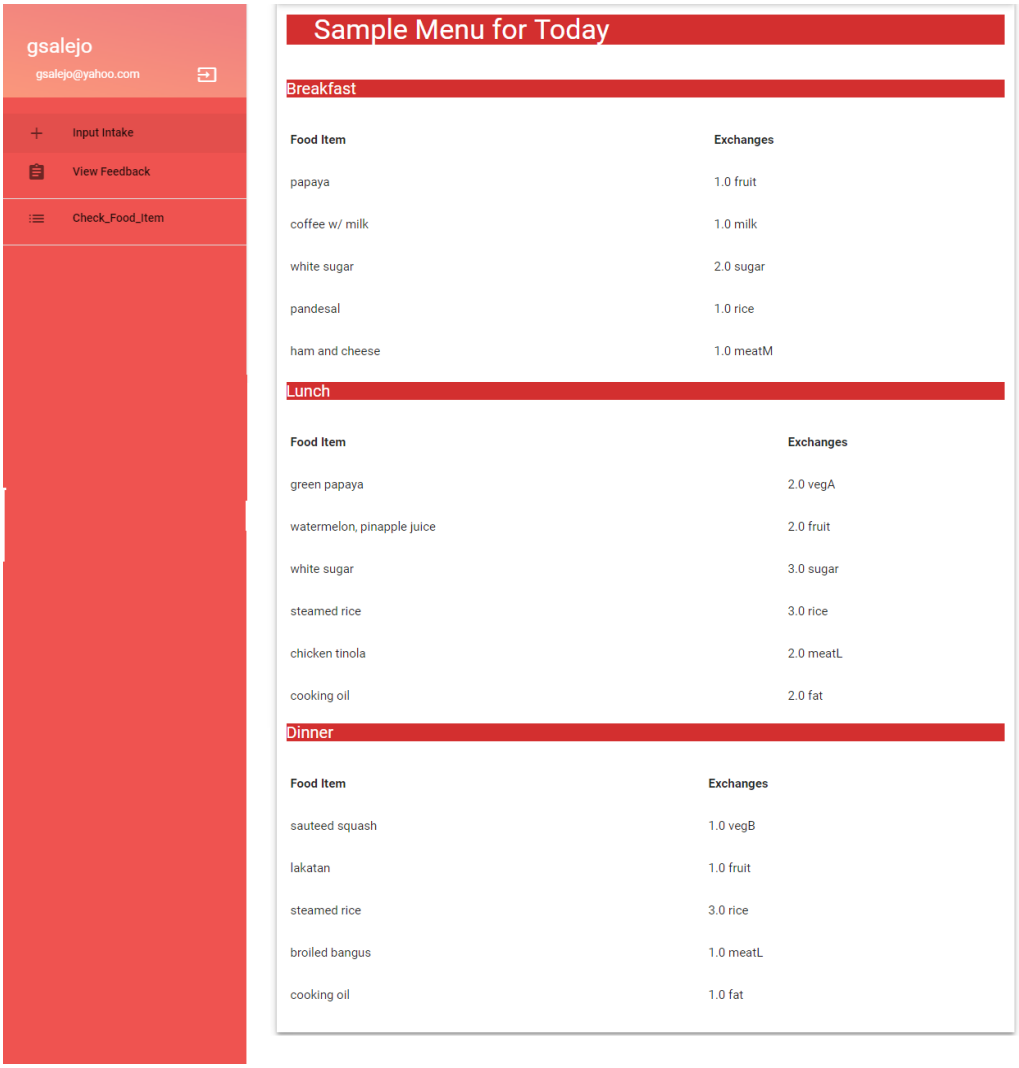

Figure 31: View Menu

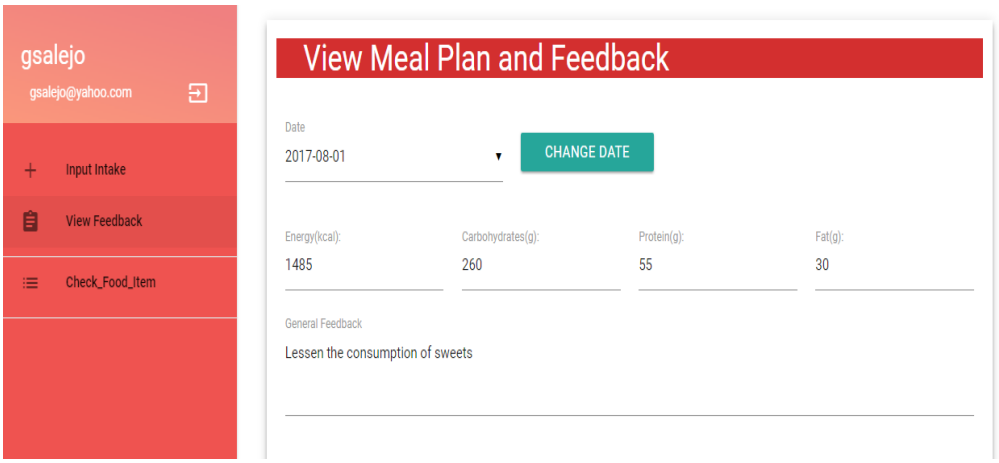

Figure 32: View Feedback

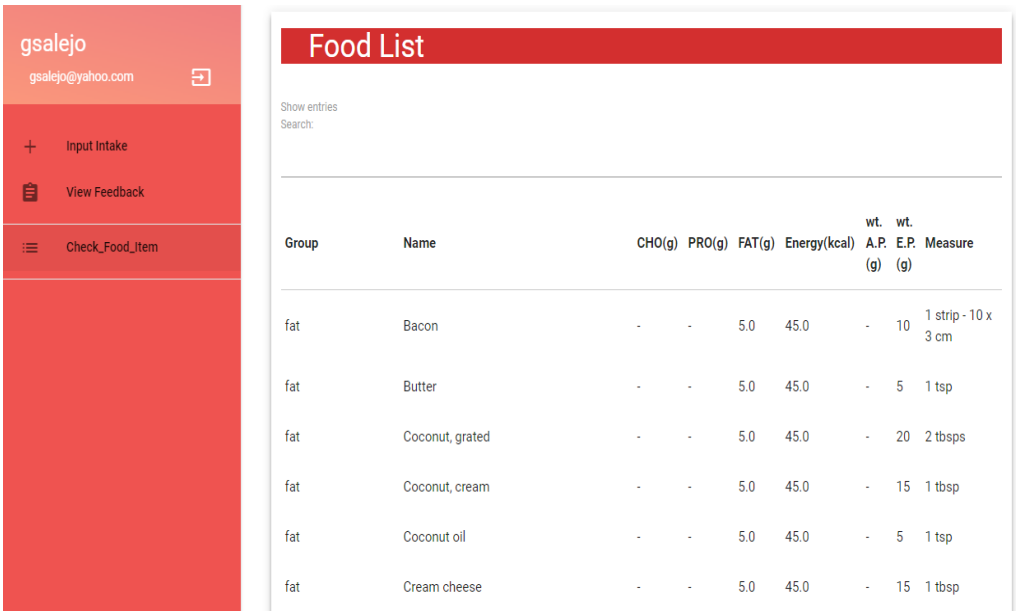

Figure 33: Food Exchange List Page

VI. Discussions

The Weight Management Patient Portal for Filipinos and Training Nutritionists is a web-based Java application which aims to provide a communication medium to the student nutritionists and their patients during the case study by which the student's goal is to make the patient approach or achieve a normal BMI. This also provides a tool for the student to create their meal plan and sample menu which is then stored electronically. The systems also enables the teacher dietitian to ensure the treatment of each student nutritionist by allowing them to approve critical processes such as the creation of meal plan, sample menu as well as the giving of feedback.

The system administrator first create set of teacher accounts, who will then create the student account and who will then create the patient accounts. The system administrator has the capaility to deactivate a user. If the system adminsitrator deactivate a teacher account, the students under that teacher will be on hang, so to prevent this, the teacher should first reassign the students to another teachers before proceeding to the deactivation of the account. This is also applied to the realtionship of students and patients, before a student account is deactivated, ateacher should reassign the patient to another student.

The process of auto-generating the meal plan strictly followed the standard procedure set by DOST-FNRI. This algorithm takes only in cosiderations cases of non-normal BMI patients and not those with special conditions such as pregnant or lactating, and diet resticting diseases such as renal problems, diabetes, etc. The creation of sample menu is done manually by the student but strictly follows the meal plan generated by the system.

The process of creating of sample menu should be done at least a day prior to the date it is followed, because the student will be unable input his intake for that date. And also, the inputing of the daily intake should be strictly followed since the student

cannot give feedback if the patient has not record his consumption for that specific date. This is done to have integrity on the data, for which the student intake shoud somehow follow the sample menu created by the nutritionist and for the student to give a sensible feedback for the patient. Since the objective of this system is to aid the student nutritionist in his treatment of a non-normal NMI patient, the procedures stated must be followed for the benefit of both parties.

Since the food exchange list can change or new food items can be added, a teacher dietitian can add new food items in the food exchange list database. This electronic database is can be seen by the patient so that he can see how much calories, carbohydrates, proteins and fats each food item have.

Eatthismuch is an application that auto-generates sample menu for a given amount of calories divided into a given amount of meals. It also provides recipe or even a delivery service for the food items the patient will need. These features are great for the patients but the primary goal of this project is to help the students in their case-study and it has a supervision of a professional.

VII. Conclusions

The Weight Management Patient Portal for Filipinos and Training Nutritionists is developed to create an online commuication medium between the student nutritionists and their patients. This system also automates the process of creating the meal plan and serves as an electronic database for the patient records throughout the case study. The main users are the student nutritionists and the patients. The teacher user is also added to supervise the treatment process. The student nutritionist can generate the meal plan, create a sample menu for his patient. The patient in turn can view this meal plan and menu which he will use as guide for his meal for that day. Once his done he inputs his intake, and in turn the student gives feedback to the patient

The system includes an electronic database of the weekly meal plan, patient's record, the daily sample menu, intakes and feedbacks, and food exchange list which allows patients to check the macronutrients of a food item.

VIII. Recommendations

The Weight Management Patient Portal for Filipinos and Training Nutritionists can be further improved by having an auto-generated sample menu as well, if it is intended for a non-treatment purposes. A sample menu database can also be added which makes a combination of food items from different food groups to create a complex food. As of now all data of the food exchange list came from the 2012 reprint provided by the DOST-FNRI. A patient query module can also be added for research purposes. The addition of meal planning procedures for other diet restricting diseases and special conditions can be benificial since it can cover more types of patients, and this will go hand in hand with the patient query, essentially needing a new set of users, the researchers.

IX. Bibliography

[1] "Body mass index (bmi) for adults." http://www.webmd.com/a-to-z-guides/body-mass-index-bmi-for-adults. Accessed: 2017-04-18.

[2] A. C. Vila, "Philippines among countries with most underweight people." http://www.philstar.com/health-and-family/2016/04/05/1569657/study-philippines-among-countries-most-underweight-people, 2016. Accessed: 2017-03-28.

[3] A. Villafania, "Does ph have among the most underweight people?." http://news.abs-cbn.com/lifestyle/04/01/16/does-ph-have-among-the-most-underweight-people. Accessed: 2017-03-28.

[4] R. Ramirez, "Number of obese, overweight pinoys increasing." http://www.philstar.com/headlines/2015/07/02/1472373/number-obese-overweight-pinoys-increasing. Accessed: 2017-03-29.

[5] T. Macas, "3 out of 10 filipino adults are overweight, obese." http://www.gmanetwork.com/news/lifestyle/healthandwellness/384493/3-out-of-10-filipino-adults-are-overweight-obese-report/story/, 2014. Accessed: 2017-03-29.

[6] B. Lagsa, "Obesity most prevalent among the richest in ph." http://www.rappler.com/move-ph/issues/hunger/99229-obesity-national-public-health-concern-philippines, 2015. Accessed: 2017-03-29.

[7] V. S. Malik, W. C. Willett, and F. B. Hu, "Global obesity: trends, risk factors and policy implications," *Nature Reviews Endocrinology*, vol. 9, no. 1, pp. 13–27, 2013.

[8] S. Crisostomo, "Doh reminds public to observe balanced diet." http://www.philstar.com/headlines/2015/10/30/1516352/doh-reminds-public-observe-balanced-diet, 2015. Accessed: 2017-04-3.

[9] V. S. Claudio, O. Dirige, and A. J. Ruiz, *Basic Nutrition for Filipinos*. Merria & Webter Bookstore, Inc., 5 ed., 2004.

[10] L. B. Bobroff, *Healthy Meal Plans*. University of Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, EDIS, 2001.

[11] *Food Exchange List For Meal Planning*. Food and Nutrition Research Institute, Department of Science and Technology, 11 2012.

[12] C. Tanchoco and A. J. Ruiz, *Diet Manual Recommended for Use in the Phillippines*. The Nutritionist-Dietitians' Association of the Philppines, 5 ed., 2010.

[13] "Nutrition and dietetics curriculum."

[14] B. Crawford, C. Castro, E. Monfroy, R. Soto, W. Palma, and F. Paredes, "Dynamic selection of enumeration strategies for solving constraint satisfaction problems," *Romanian Journal of Information Science and Technology*, vol. 15, no. 2, pp. 106–128, 2012.

[15] G. Kovasznai, "Developing an expert system for diet recommendation," in *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on*, pp. 505–509, IEEE, 2011.

[16] K. Kurozumi, S.-T. Lan, M.-H. Wang, C.-S. Lee, M. Kawaguchi, S. Tsumoto, and H. Tsuji, "Fml-based japanese diet assessment system," in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pp. 1–6, IEEE, 2013.

[17] M.-H. Wang, K. Kurozumi, M. Kawaguchi, C.-S. Lee, H. Tsuji, and S. Tsumoto, "Healthy diet assessment mechanism based on fuzzy markup language for japanese food," *Soft Computing*, vol. 20, no. 1, pp. 359–376, 2016.

[18] D. Fister, I. Fister, S. Rauter, and I. Fister, "Generating eating plans for athletes using the particle swarm optimization," in *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 000193–000198, Nov 2016.

[19] A. Dholvitayakhun and J. Kluabwang, "Application of local search for optimal assignment of food exchange lists problem," *International Journal of Computer Theory and Engineering*, vol. 6, no. 2, p. 189, 2014.

[20] A. Dholvitayakhun and J. Kluabwang, "Design of food exchange list for obesity using modified local search," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2016 13th International Conference on*, pp. 1–5, IEEE, 2016.

[21] M. Mamat, N. F. Zulkifli, S. K. Deraman, and N. M. M. Noor, "Fuzzy linear programming approach in balance diet planning for eating disorder and disease-related lifestyle," *Applied Mathematical Sciences*, vol. 6, no. 103, pp. 5109–5118, 2012.

[22] J. F. Raffensperger, "The least-cost low-carbohydrate diet is expensive," *Nutrition research*, vol. 28, no. 1, pp. 6–12, 2008.

[23] M. El-Dosuky, M. Rashad, T. Hamza, and A. El-Bassiouny, "Food recommendation using ontology and heuristics," in *International Conference on Advanced Machine Learning Technologies and Applications*, pp. 423–429, Springer, 2012.

[24] D. Elsweiler and M. Harvey, "Towards automatic meal plan recommendations for balanced nutrition," in *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 313–316, ACM, 2015.

[25] P. Compton, G. Edwards, B. Kang, L. Lazarus, R. Malor, P. Preston, and A. Srinivasan, "Ripple down rules: Turning knowledge acquisition into knowledge maintenance," *Artificial Intelligence in Medicine*, vol. 4, no. 6, pp. 463–475, 1992.

[26] J. H. Hsiao and H. Chang, "Smartdiet: A personal diet consultant for healthy meal planning," in *2010 IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 421–425, Oct 2010.

[27] M. Tom, "Computational intelligence using fuzzy multicriteria decision making for diligens: Dietary intelligence system," in *2012 IEEE International Conference on Fuzzy Systems*, pp. 1–7, June 2012.

[28] L. E. Keplinger, R. J. Koopman, D. R. Mehr, R. L. Kruse, D. S. Wakefield, B. J. Wakefield, and S. M. Canfield, "Patient portal implementation," *Family medicine*, vol. 45, no. 5, pp. 335–40, 2013.

[29] R. W. Grant, J. S. Wald, E. G. Poon, J. L. Schnipper, T. K. Gandhi, L. A. Volk, and B. Middleton, "Design and implementation of a web-based patient portal linked to an ambulatory care electronic health record: patient gateway for diabetes collaborative care," *Diabetes technology & therapeutics*, vol. 8, no. 5, pp. 576–586, 2006.

[30] V. Zwass, "information system." https://www.britannica.com/topic/information-system. Accessed: 2017-04-15.

[31] S. Fuad, "Information systems - definitions and concepts." http://www.uotechnology.edu.iq/ce/Lectures/SarmadFuad-MIS/MIS_Lecture_3. Accessed: 2017-04-15.

[32] A.-W. Scheer, *Architecture of integrated information systems: foundations of enterprise modelling*. Springer Science & Business Media, 2012.

[33] "Design patterns - mvc pattern." https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm. Accessed: 2017-04-18.

[34] Y. H. Ding, C. H. Liu, and Y. X. Tang, "Mvc pattern based on java," in *Applied Mechanics and Materials*, vol. 198, pp. 537–541, Trans Tech Publ, 2012.

[35] "Structure query language-sql." https://techterms.com/definition/sql. Accessed: 2017-04-18.

[36] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, 1997.

[37] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, USA: Springer-Verlag New York, Inc., 1995.

[38] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, pp. 1464–1480, 1990.

[39] T. Kohonen, J. Hynninen, and J. Kangas, "Som_pak: Self-organizing map program package," *Journal of Computer Science*, 1995.

X. Glossary

- student nutritionist - also refers to student or nutritionist who directly handles the patient
- teacher dietitian - also referns to teacher or dietitian who handles the student nutritionist

XI. Appendix

A. Source Code

```
package dbutils;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

import models.FoodListModel;
import models.IntakeModel;
import models.MealPlanModel;
import models.MenuModel;

public class DBDataExtract {
        /**
         * This function returns the role of the user given their user ID from the Database
         * @param id     User's ID
         * @return role User's Role (0-admin, 1-teacher 2-student 3-patient)
         */
        public static MealPlanModel getDietReference(double tea){
                MealPlanModel mealPlan = new MealPlanModel();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM dietreference WHERE energy <= '" + tea + " ORDER BY energy AS
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                tea = rs.getDouble("energy");
                                mealPlan.setExVegA(rs.getDouble("vegA"));
                                mealPlan.setExVegB(rs.getDouble("vegB"));
                                mealPlan.setExFruit(rs.getDouble("fruit"));
                                mealPlan.setExMilk(rs.getDouble("milk"));
                                mealPlan.setExSugar(rs.getDouble("sugar"));
                                mealPlan.setExRice(rs.getDouble("rice"));
                                mealPlan.setExMeatL(rs.getDouble("meatL"));
                                mealPlan.setExMeatM(rs.getDouble("meatM"));
                                mealPlan.setExFat(rs.getDouble("fat"));
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return mealPlan;
        }

        public static int getLastWeek(int patientId){
                int week = 1;
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT week FROM weeklyupdate WHERE patient = '" + patientId + "';";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                week += 1;
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return week;
        }

        public static Date getLastDay(int patientId){
                int week = getLastWeek(patientId) - 1;
                Date date = null;
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT `date` FROM `weeklyupdate` INNER JOIN `samplemenu` ON samplemenu.wee
                                        + "WHERE patient = '" + patientId + "' AND week = '"+ week +"'
ORDER by date DESC ;";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        if(rs.next()){
                                Timestamp ts = rs.getTimestamp("date");
                                date = new Date(ts.getTime());
```

```
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return date;
}

public static Date getLastSampleDate(int patientId){

        int week = getLastWeek(patientId) - 1;
        Date date = null;
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT end FROM weeklyupdate WHERE patient = '" + patientId + "' AND week =
                                + week + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                if(rs.next()){
                        Timestamp ts = rs.getTimestamp("end");
                        date = new Date(ts.getTime());
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return date;
}

public static boolean isComplete(int patientId){
        int week = getLastWeek(patientId) - 1;
        Date date = null;
        Date date2 = null;
        boolean isSame = false;
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT date, end FROM weeklyupdate INNER JOIN samplemenu ON weekId = weekly
                                + "WHERE patient = '" + patientId + "' AND week = '"
                                + week + "' ORDER by date DESC ;";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                if(rs.next()){
                        Timestamp dts = rs.getTimestamp("date");
                        date = new Date(dts.getTime());
                        Timestamp ets = rs.getTimestamp("end");
                        date2 = new Date(ets.getTime());
                        if(date.equals(date2)){
                                isSame = true;
                        }
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return isSame;
}

public static Date getStartDay(int patientId){
        int week = getLastWeek(patientId) - 1;
        Date date = null;
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT start FROM weeklyupdate WHERE patient = '" + patientId + "' AND week
                                + week + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                if(rs.next()){
                        Timestamp ts = rs.getTimestamp("start");
                        date = new Date(ts.getTime());
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return date;
}

public static ArrayList<MealPlanModel> getWeeklyReport(int id){
        ArrayList<MealPlanModel> weeklyMealPlans = new ArrayList<MealPlanModel>();
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM weeklyupdate"
                                + " WHERE patient = '" + id + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        MealPlanModel mealPlan = new MealPlanModel();
                        mealPlan.setId(rs.getInt("id"));
                        mealPlan.setPatientId(rs.getInt("patient"));
                        mealPlan.setWeek(rs.getInt("week"));
                        Timestamp tsStart = rs.getTimestamp("start");
                        Timestamp tsEnd = rs.getTimestamp("end");
```

```
                        mealPlan.setStart(new Date(tsStart.getTime()));
                        mealPlan.setEnd(new Date(tsEnd.getTime()));
                        mealPlan.setHeight(rs.getDouble("height"));
                        mealPlan.setWeight(rs.getDouble("weight"));
                        mealPlan.setBmi(rs.getDouble("bmi"));
                        mealPlan.setClassification(rs.getString("classification"));
                        mealPlan.setDbw(rs.getDouble("desiredBodyWeight"));
                        mealPlan.setActivity(rs.getString("activity"));
                        mealPlan.setTea(rs.getDouble("totalEnergyAllowance"));
                        mealPlan.setgCHO(rs.getDouble("gc"));
                        mealPlan.setgPRO(rs.getDouble("gp"));
                        mealPlan.setgFAT(rs.getDouble("gf"));
                        mealPlan.setExVegA(rs.getDouble("vegA"));
                        mealPlan.setExVegB(rs.getDouble("vegB"));
                        mealPlan.setExFruit(rs.getDouble("fruit"));
                        mealPlan.setExMilk(rs.getDouble("milk"));
                        mealPlan.setExSugar(rs.getDouble("sugar"));
                        mealPlan.setExRice(rs.getDouble("rice"));
                        mealPlan.setExMeatL(rs.getDouble("meatL"));
                        mealPlan.setExMeatM(rs.getDouble("meatM"));
                        mealPlan.setExFat(rs.getDouble("fat"));
                        weeklyMealPlans.add(mealPlan);
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return weeklyMealPlans;
}

public static MealPlanModel getReport(int id, int week){
        MealPlanModel mealPlan = new MealPlanModel();
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM weeklyupdate WHERE patient = '" + id + "' AND week = '" + wee
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        mealPlan.setId(rs.getInt("id"));
                        mealPlan.setPatientId(rs.getInt("patient"));
                        mealPlan.setWeek(rs.getInt("week"));
                        mealPlan.setHeight(rs.getDouble("height"));
                        mealPlan.setWeight(rs.getDouble("weight"));
                        mealPlan.setBmi(rs.getDouble("bmi"));
                        mealPlan.setClassification(rs.getString("classification"));
                        mealPlan.setDbw(rs.getDouble("desiredBodyWeight"));
                        mealPlan.setActivity(rs.getString("activity"));
                        mealPlan.setTea(rs.getDouble("totalEnergyAllowance"));
                        mealPlan.setgCHO(rs.getDouble("gc"));
                        mealPlan.setgPRO(rs.getDouble("gp"));
                        mealPlan.setgFAT(rs.getDouble("gf"));
                        mealPlan.setExVegA(rs.getDouble("vegA"));
                        mealPlan.setExVegB(rs.getDouble("vegB"));
                        mealPlan.setExFruit(rs.getDouble("fruit"));
                        mealPlan.setExMilk(rs.getDouble("milk"));
                        mealPlan.setExSugar(rs.getDouble("sugar"));
                        mealPlan.setExRice(rs.getDouble("rice"));
                        mealPlan.setExMeatL(rs.getDouble("meatL"));
                        mealPlan.setExMeatM(rs.getDouble("meatM"));
                        mealPlan.setExFat(rs.getDouble("fat"));
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return mealPlan;
}

public static MealPlanModel getMealPlan(int id){
        MealPlanModel mealPlan = new MealPlanModel();
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM weeklyupdate WHERE ID = '" + id + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        mealPlan.setId(rs.getInt("id"));
                        mealPlan.setPatientId(rs.getInt("patient"));
                        mealPlan.setWeek(rs.getInt("week"));
                        mealPlan.setHeight(rs.getDouble("height"));
                        mealPlan.setWeight(rs.getDouble("weight"));
                        mealPlan.setBmi(rs.getDouble("bmi"));
                        mealPlan.setClassification(rs.getString("classification"));
                        mealPlan.setDbw(rs.getDouble("desiredBodyWeight"));
                        mealPlan.setActivity(rs.getString("activity"));
                        mealPlan.setTea(rs.getDouble("totalEnergyAllowance"));
                        mealPlan.setgCHO(rs.getDouble("gc"));
                        mealPlan.setgPRO(rs.getDouble("gp"));
                        mealPlan.setgFAT(rs.getDouble("gf"));
                        mealPlan.setExVegA(rs.getDouble("vegA"));
                        mealPlan.setExVegB(rs.getDouble("vegB"));
                        mealPlan.setExFruit(rs.getDouble("fruit"));
```

```
                        mealPlan.setExMilk(rs.getDouble("milk"));
                        mealPlan.setExSugar(rs.getDouble("sugar"));
                        mealPlan.setExRice(rs.getDouble("rice"));
                        mealPlan.setExMeatL(rs.getDouble("meatL"));
                        mealPlan.setExMeatM(rs.getDouble("meatM"));
                        mealPlan.setExFat(rs.getDouble("fat"));
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return mealPlan;
}

public static ArrayList<FoodListModel> getFoodList(){
        ArrayList<FoodListModel> foodList = new ArrayList<FoodListModel>();
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM `foodexchange` ORDER BY `ID` ASC";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        FoodListModel food = new FoodListModel();
                        food.setId(rs.getInt("id"));
                        food.setGroup(rs.getString("food_group"));
                        food.setName(rs.getString("name"));
                        food.setgCHO(rs.getDouble("CHO"));
                        food.setgPRO(rs.getDouble("PRO"));
                        food.setgFAT(rs.getDouble("FAT"));
                        food.setWtAP(rs.getInt("weight_AP"));
                        food.setWtEP(rs.getInt("weight_EP"));
                        food.setEnergy(rs.getDouble("energy"));
                        food.setMeasure(rs.getString("measure"));
                        foodList.add(food);
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return foodList;
}

public static MenuModel getSampleMenu(int id, Date date){
        MenuModel menu = null;
        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT samplemenu.id, date, weekId, breakfast, lunch, dinner"
                                + " FROM weeklyupdate INNER JOIN samplemenu ON weekId = weeklyupdate.ID "
                                + "WHERE patient = '" + id + "' ORDER by date DESC ;";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        String day1 = "";
                        String day2 = "";
                        day1 = dateFormat.format(date);
                        Timestamp ts = rs.getTimestamp("date");
                        day2 = dateFormat.format(ts.getTime());
                        if(day1.equals(day2)){
                                menu = new MenuModel();
                                menu.setId(rs.getInt("id"));
                                menu.setDate(date);
                                menu.setWeekId(rs.getInt("weekId"));
                                menu.setBreakfast(rs.getString("breakfast"));
                                menu.setLunch(rs.getString("lunch"));
                                menu.setDinner(rs.getString("dinner"));
                        }
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return menu;
}

public static boolean hasSubmittedIntake(int menuId){
        boolean hasSubmitted=false;

        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT menuId FROM dailyintake WHERE menuId = '" +menuId+ "' ;";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                if(rs.next()){
                        hasSubmitted = true;
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return hasSubmitted;
```

```
        }

        public static IntakeModel getIntakes(int id, Date date){
                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                String date1 = dateFormat.format(date);
                String date2 = "";
                IntakeModel intake = new IntakeModel();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT dailyintake.id, menuId, foodItem, feedback, CHO, PRO, dailyintake.FA
                                        + " FROM dailyintake, samplemenu, weeklyupdate "
                                        + "WHERE patient = '" + id + "' AND menuId = samplemenu.id AND weekId=weekl
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                Timestamp ts = rs.getTimestamp("date");
                                date2 = dateFormat.format(ts.getTime());
                                if(date1.equals(date2)){
                                        intake.setId(rs.getInt("id"));
                                        intake.setMenuId(rs.getInt("menuId"));
                                        intake.setFoodItem(rs.getString("foodItem"));
                                        intake.setFeedback(rs.getString("feedback"));
                                        intake.setgCHO(rs.getInt("CHO"));
                                        intake.setgPRO(rs.getInt("PRO"));
                                        intake.setgFAT(rs.getInt("FAT"));
                                        intake.setEnergy(rs.getInt("energy"));
                                }
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return intake;
        }

        public static ArrayList<Date> getAllDates(int id){
                ArrayList<Date> dates = new ArrayList<Date>();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT date"
                                        + " FROM dailyintake, samplemenu, weeklyupdate "
                                        + "WHERE patient = '" + id + "' AND menuId = samplemenu.id AND weekId=weekl
                                                        + " ORDER BY date DESC;";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                Timestamp ts = rs.getTimestamp("date");
                                Date date = new Date(ts.getTime());
                                dates.add(date);
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return dates;
        }

        public static String getCode(Date date){
                String code = "ERROR";
                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                String date1 = dateFormat.format(date);
                String date2 = "";
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT *"
                                        + " FROM dailycode;";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                Timestamp ts = rs.getTimestamp("date");
                                date2 = dateFormat.format(ts.getTime());
                                if(date1.equals(date2)){
                                        code = rs.getString("code");
                                }
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return code;
        }
}


package dbutils;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.Timestamp;
import java.util.ArrayList;
import java.util.Date;
```

```
import org.apache.commons.codec.digest.DigestUtils;

import models.PatientModel;
import models.StudentModel;
import models.TeacherModel;
import models.UserModel;

public class DBUserExtract {
        /**
         * This function returns the role of the user given their user ID from the Database
         * @param id     User's ID
         * @return role User's Role (0-admin, 1-teacher 2-student 3-patient)
         */
        public static UserModel getUserDetails(int id){
                UserModel user = new UserModel();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM users WHERE id = '" + id + "';";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                user.setId(rs.getInt("id"));
                                user.setUsername(rs.getString("username"));
                                user.setRole(rs.getInt("role"));
                                user.setDeactivated(rs.getBoolean("deactivated"));
                                user.setEmail(rs.getString("email"));
                                user.setContact(rs.getString("contact"));
                                user.setPassword(rs.getString("password"));
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return user;
        }

        public static ArrayList<TeacherModel> getTeachers(){
                ArrayList<TeacherModel> teachers = new ArrayList<TeacherModel>();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM users JOIN teacher ON users.ID = teacher.ID";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                int id = rs.getInt("id");
                                String username = rs.getString("username");
                                int role = rs.getInt("role");
                                boolean deactivated = rs.getBoolean("deactivated");
                                String email = rs.getString("email");
                                String fname = rs.getString("fname");
                                String mname = rs.getString("mname");
                                String lname = rs.getString("lname");
                                String contact = rs.getString("contact");
                                teachers.add(new TeacherModel(id, username, email, "", role, deactivated, fname, mn
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return teachers;
        }

        public static ArrayList<StudentModel> getStudents(){
                ArrayList<StudentModel> students = new ArrayList<StudentModel>();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM users JOIN student ON users.ID = student.ID";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                int id = rs.getInt("id");
                                String username = rs.getString("username");
                                int role = rs.getInt("role");
                                boolean deactivated = rs.getBoolean("deactivated");
                                String email = rs.getString("email");
                                int teacher = rs.getInt("teacher");
                                String fname = rs.getString("fname");
                                String mname = rs.getString("mname");
                                String lname = rs.getString("lname");
                                String contact = rs.getString("contact");
                                students.add(new StudentModel(id, username, email, "", role, deactivated, teacher,
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return students;
        }

        public static ArrayList<PatientModel> getPatients(){
                ArrayList<PatientModel> patients = new ArrayList<PatientModel>();
```

```
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM users JOIN patient ON users.ID = patient.ID";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        int id = rs.getInt("id");
                        String username = rs.getString("username");
                        int role = rs.getInt("role");
                        boolean deactivated = rs.getBoolean("deactivated");
                        String email = rs.getString("email");
                        int student = rs.getInt("student");
                        String fname = rs.getString("fname");
                        String mname = rs.getString("mname");
                        String lname = rs.getString("lname");
                        String contact = rs.getString("contact");
                        String sex = rs.getString("sex");
                        Timestamp ts = rs.getTimestamp("birthdate");
                        Date birthdate = new Date(ts.getTime());
                        patients.add(new PatientModel(id, username, email, "", role, deactivated, student,
                                        mname, lname, contact, sex, birthdate));
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return patients;
}

public static int getId(String username){
        int id = 0;
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM users WHERE username = '" + username + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        id = rs.getInt("ID");
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return id;
}

/**
 * This function verifies if the username and password matches, and if the user account is deactivated or n
 * If the username and password matches and the account is active, the function returns the user's ID
 * @param username      User's Username
 * @param password      User's Password
 * @return id           User's ID
 */
public static int verifyUser(String username, String password){
        int id = 0;
        String retPswd = "";
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM users WHERE username = '" + username + "' AND deactivated = '
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        retPswd = rs.getString("password");
                        if(DigestUtils.sha1Hex(password).equals(retPswd)){
                                id = rs.getInt("id");
                        }
                }
                //System.out.println("Success: " + id);
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return id;
}

/**
 * This function checks if the username and email is already existing
 * returns true if either the username or email already exists and false otherwise
 * @param username      User's Username
 * @param email         User's Email
 * @return exists       true-user is existing, false-user is not registered
 */
public static boolean isUserExisting(String username, String email){
        return DBUserExtract.isEmailExisting(email) || DBUserExtract.isUsernameExisting(username);
}

public static boolean isEmailExisting(String email){
        boolean exists = true;
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM users WHERE email = '" + email + "';";
                Statement stm = conn.prepareStatement(query);
```

```
                        ResultSet rs = stm.executeQuery(query);
                        if(rs.next()){
                                exists = true;
                        } else{
                                exists = false;
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return exists;
        }

        public static int getIdUsingEmail(String email){
                int id = -1;
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM users WHERE email = '" + email + "';";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        if(rs.next()){
                                id = rs.getInt("id");
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return id;
        }

        public static boolean isUsernameExisting(String username){
                boolean exists = true;
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM users WHERE username = '" + username + "';";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        if(rs.next()){
                                exists = true;
                        } else{
                                exists = false;
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return exists;
        }

        public static boolean isEmailTheSame(int id, String email){
                boolean same = false;
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM users WHERE email = '" + email + "';";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        if(rs.next()){
                                same = (rs.getInt("id") == id);
                        } else{
                                same = false;
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return same;
        }

        public static boolean isUsernameTheSame(int id, String username){
                boolean same = true;
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM users WHERE username = '" + username + "';";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        if(rs.next()){
                                same = (rs.getInt("id") == id);
                        } else{
                                same = false;
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return same;
        }

}

package dbutils;
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.Timestamp;
import java.util.ArrayList;
import java.util.Date;

import org.apache.commons.codec.digest.DigestUtils;

import models.MealPlanModel;


public class DBInsert {
	public static boolean addTeacher(String username, String password, String email, String role,
			String fname, String mname, String lname, String contact){
		try{
			Connection conn = DBUtil.getConnection();
			String query = "INSERT INTO users (username, email, contact, password, role) VALUES "
					+ "(?,?,?,?,?);";
			PreparedStatement pstm = conn.prepareStatement(query);
			pstm.setString(1, username);
			pstm.setString(2, email);
			pstm.setString(3, contact);
			pstm.setString(4, DigestUtils.sha1Hex(password));
			pstm.setString(5, role);
			pstm.executeUpdate();

			query = "INSERT INTO teacher (ID, fname, mname, lname) VALUES "
					+ "(?,?,?,?);";
			//System.out.println("Contact: " + contact);
			PreparedStatement pstm2 = conn.prepareStatement(query);
			pstm2.setString(1, DBUserExtract.getId(username) + "");
			pstm2.setString(2, fname);
			pstm2.setString(3, mname);
			pstm2.setString(4, lname);
			pstm2.executeUpdate();

			DBUtil.closeConnection(conn);
			return true;
		} catch(Exception e){
			e.printStackTrace();
			return false;
		}
	}

	public static boolean addStudent(String username, String password, String email, String role,
			String fname, String mname, String lname, String contact, int teacherId){
		try{
			Connection conn = DBUtil.getConnection();
			String query = "INSERT INTO users (username, email, contact, password, role) VALUES "
					+ "(?,?,?,?,?);";
			PreparedStatement pstm = conn.prepareStatement(query);
			pstm.setString(1, username);
			pstm.setString(2, email);
			pstm.setString(3, contact);
			pstm.setString(4, DigestUtils.sha1Hex(password));
			pstm.setString(5, role);
			pstm.executeUpdate();

			query = "INSERT INTO student (ID, fname, mname, lname, teacher) VALUES "
					+ "(?,?,?,?,?);";
			//System.out.println("Contact: " + contact);
			PreparedStatement pstm2 = conn.prepareStatement(query);
			pstm2.setString(1, DBUserExtract.getId(username) + "");
			pstm2.setString(2, fname);
			pstm2.setString(3, mname);
			pstm2.setString(4, lname);
			pstm2.setInt(5, teacherId);
			pstm2.executeUpdate();

			DBUtil.closeConnection(conn);
			return true;
		} catch(Exception e){
			e.printStackTrace();
			return false;
		}
	}

	public static boolean addPatient(String username, String password, String email, String role,
			String fname, String mname, String lname, String contact, int studentId, Timestamp birthdat
		try{
			Connection conn = DBUtil.getConnection();
			String query = "INSERT INTO users (username, email, contact, password, role) VALUES "
					+ "(?,?,?,?,?);";
			PreparedStatement pstm = conn.prepareStatement(query);
			pstm.setString(1, username);
			pstm.setString(2, email);
			pstm.setString(3, contact);
			pstm.setString(4, DigestUtils.sha1Hex(password));
			pstm.setString(5, role);
			pstm.executeUpdate();
```

```
                query = "INSERT INTO patient (ID, fname, mname, lname, student, birthdate, sex) VALUES "
                        + "(?,?,?,?,?,?,?);";
                //System.out.println("Contact: " + contact);
                PreparedStatement pstm2 = conn.prepareStatement(query);
                pstm2.setString(1, DBUserExtract.getId(username) + "");
                pstm2.setString(2, fname);
                pstm2.setString(3, mname);
                pstm2.setString(4, lname);
                pstm2.setInt(5, studentId);
                pstm2.setTimestamp(6, birthdate);
                pstm2.setString(7, sex);
                pstm2.executeUpdate();
                DBUtil.closeConnection(conn);
                return true;
        } catch(Exception e){
                e.printStackTrace();
                return false;
        }
}

public static boolean addWeeklyReport(MealPlanModel mealPlan){
        try{
                //System.out.println(dateFormat.format(mealPlan.getStart().getTime()));
                Connection conn = DBUtil.getConnection();
                Timestamp tsStart = new Timestamp(mealPlan.getStart().getTime());
                Timestamp tsEnd = new Timestamp(mealPlan.getEnd().getTime());
                String query = "INSERT INTO `weeklyupdate` (`patient`, `week`, `height`, `weight`, `bmi`, "
                        + "`classification`, `desiredBodyWeight`, `activity`, `totalEnergyAllowance
                        + "`gp`, `gf`, `vegA`, `vegB`, `fruit`, `milk`, `sugar`, `rice`, `meatL`, `
                        + "`start`, `end`) "
                        + "VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?);";
                PreparedStatement pstm = conn.prepareStatement(query);
                pstm.setInt(1, mealPlan.getPatientId());
                pstm.setInt(2, mealPlan.getWeek());
                pstm.setDouble(3, mealPlan.getHeight());
                pstm.setDouble(4, mealPlan.getWeight());
                pstm.setDouble(5, mealPlan.getBmi());
                pstm.setString(6, mealPlan.getClassification());
                pstm.setDouble(7, mealPlan.getDbw());
                pstm.setString(8, mealPlan.getActivity());
                pstm.setDouble(9, mealPlan.getTea());
                pstm.setDouble(10, mealPlan.getgCHO());
                pstm.setDouble(11, mealPlan.getgPRO());
                pstm.setDouble(12, mealPlan.getgFAT());

                pstm.setDouble(13, mealPlan.getExVegA());
                pstm.setDouble(14, mealPlan.getExVegB());
                pstm.setDouble(15, mealPlan.getExFruit());
                pstm.setDouble(16, mealPlan.getExMilk());
                pstm.setDouble(17, mealPlan.getExSugar());
                pstm.setDouble(18, mealPlan.getExRice());
                pstm.setDouble(19, mealPlan.getExMeatL());
                pstm.setDouble(20, mealPlan.getExMeatM());
                pstm.setDouble(21, mealPlan.getExFat());
                pstm.setTimestamp(22, tsStart);
                pstm.setTimestamp(23, tsEnd);
                pstm.executeUpdate();

                DBUtil.closeConnection(conn);
                return true;
        } catch(Exception e){
                e.printStackTrace();
                return false;
        }
}

public static boolean submitFoodItem(int menuId, ArrayList<String> foods){
        String foodItem = "";
        boolean first = true;
        for(String food : foods){
                if(first){
                        first = false;
                        foodItem += food;
                }
                foodItem += "||" + food;
        }
        try{
                Connection conn = DBUtil.getConnection();
                String query = "INSERT INTO dailyintake (menuId, foodItem) VALUES "
                        + "(?,?);";
                PreparedStatement pstm = conn.prepareStatement(query);
                pstm.setInt(1, menuId);
                pstm.setString(2, foodItem);
                pstm.executeUpdate();

                DBUtil.closeConnection(conn);
                return true;
        } catch(Exception e){
                e.printStackTrace();
                return false;
        }
}
```

```
public static boolean addSampleMenu(int week, Date date, String breakfast, String lunch, String dinner){
        try{
                Connection conn = DBUtil.getConnection();
                String query = "INSERT INTO `samplemenu` (`weekId`, `date`, `breakfast`, `lunch`, `dinner`)
                        + "VALUES (?,?,?,?,?);";
                PreparedStatement pstm = conn.prepareStatement(query);
                pstm.setInt(1, week);
                Timestamp ts = new Timestamp(date.getTime());
                pstm.setTimestamp(2, ts);

                pstm.setString(3, breakfast);
                pstm.setString(4, lunch);
                pstm.setString(5, dinner);

                pstm.executeUpdate();

                DBUtil.closeConnection(conn);
                return true;
        } catch(Exception e){
                e.printStackTrace();
                return false;
        }
}

public static boolean addFoodItem(String foodName, String foodGroup, int wtAP, int wtEP, String measure,
                int cho, int pro, int fat, int energy){
        try{
                Connection conn = DBUtil.getConnection();
                String query = "INSERT INTO `foodexchange` (`food_group`, `name`, `CHO`, `PRO`, `FAT`, `ene
                        + "`weight_AP`, `weight_EP`, `measure`) "
                        + "VALUES (?,?,?,?,?,?,?,?,?);";
                PreparedStatement pstm = conn.prepareStatement(query);
                pstm.setString(1, foodGroup);
                pstm.setString(2, foodName);

                pstm.setInt(3, cho);
                pstm.setInt(4, pro);
                pstm.setInt(5, fat);
                pstm.setInt(6, energy);
                pstm.setInt(7, wtAP);
                pstm.setInt(8, wtEP);
                pstm.setString(9, measure);

                pstm.executeUpdate();

                DBUtil.closeConnection(conn);
                return true;
        } catch(Exception e){
                e.printStackTrace();
                return false;
        }
}

public static boolean addIntake(int menuId, String foodItem){
        try{
                Connection conn = DBUtil.getConnection();
                String query = "INSERT INTO `dailyintake` (`menuId`, `foodItem`) "
                        + "VALUES (?,?);";
                PreparedStatement pstm = conn.prepareStatement(query);
                pstm.setInt(1, menuId);
                pstm.setString(2, foodItem);

                pstm.executeUpdate();

                DBUtil.closeConnection(conn);
                return true;
        } catch(Exception e){
                e.printStackTrace();
                return false;
        }
}

public static boolean addDailyCode(Date date, String code){
        try{
                Connection conn = DBUtil.getConnection();
                String query = "INSERT INTO `dailycode` (`date`, `code`) "
                        + "VALUES (?,?);";
                PreparedStatement pstm = conn.prepareStatement(query);
                Timestamp ts = new Timestamp(date.getTime());
                pstm.setTimestamp(1, ts);
                pstm.setString(2, code);

                pstm.executeUpdate();

                DBUtil.closeConnection(conn);
                return true;
        } catch(Exception e){
                e.printStackTrace();
                return false;
        }
}
```

```
}

package dbutils;

import java.sql.Connection;
import java.sql.PreparedStatement;

import org.apache.commons.codec.digest.DigestUtils;

public class DBUpdate {

	public static boolean deactivateUser(int userId){
		boolean success = false;
		try{
			Connection conn = DBUtil.getConnection();
			String query = "UPDATE users SET deactivated = '1' WHERE id = ?;";
			PreparedStatement stm = conn.prepareStatement(query);
			stm.setInt(1, userId);
			stm.execute();
			success = true;
			DBUtil.closeConnection(conn);
		} catch(Exception e){
			e.printStackTrace();
			success = false;
		}
		return success;
	}

	public static boolean giveFeedback(int id, String feedback, int gCHO, int gPRO, int gFAT, int energy){
		boolean success = false;
		try{
			Connection conn = DBUtil.getConnection();
			String query = "UPDATE dailyintake SET feedback = ?, CHO = ?, PRO = ?, FAT = ?, energy = ? "
			PreparedStatement stm = conn.prepareStatement(query);
			stm.setString(1, feedback);
			stm.setInt(2, gCHO);
			stm.setInt(3, gPRO);
			stm.setInt(4, gFAT);
			stm.setInt(5, energy);
			stm.setInt(6, id);
			stm.execute();
			success = true;
			DBUtil.closeConnection(conn);
		} catch(Exception e){
			e.printStackTrace();
			success = false;
		}
		return success;
	}

	public static boolean reassignStudent(int userId, int teacherId){
		boolean success = false;
		try{
			Connection conn = DBUtil.getConnection();
			String query = "UPDATE student SET teacher = ? WHERE id = ?;";
			PreparedStatement stm = conn.prepareStatement(query);
			stm.setInt(1, teacherId);
			stm.setInt(2, userId);
			stm.execute();
			success = true;
			DBUtil.closeConnection(conn);
		} catch(Exception e){
			e.printStackTrace();
			success = false;
		}
		return success;
	}

	public static boolean reassignPatient(int userId, int studentId){
		boolean success = false;
		try{
			Connection conn = DBUtil.getConnection();
			String query = "UPDATE patient SET student = ? WHERE id = ?;";
			PreparedStatement stm = conn.prepareStatement(query);
			stm.setInt(1, studentId);
			stm.setInt(2, userId);
			stm.execute();
			success = true;
			DBUtil.closeConnection(conn);
		} catch(Exception e){
			e.printStackTrace();
			success = false;
		}
		return success;
	}

	public static boolean resetPassword(int userId, String password){
		boolean success = false;
		try{
			Connection conn = DBUtil.getConnection();
			String query = "UPDATE users SET password = ? WHERE id = ?;";
```

```
                        PreparedStatement stm = conn.prepareStatement(query);
                        stm.setString(1, DigestUtils.sha1Hex(password));
                        stm.setInt(2, userId);
                        stm.execute();
                        success = true;

                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                        success = false;
                }
                return success;
        }

        public static boolean editProfile(int userId, String username, String contact,
                        String email, String password){
                boolean success = false;
                try{
                        Connection conn = DBUtil.getConnection();
                        if(password == ""){
                                String query = "UPDATE users SET username = ?, email = ?, contact = ? WHERE id = ?;
                                PreparedStatement stm = conn.prepareStatement(query);
                                stm.setString(1, username);
                                stm.setString(2, email);
                                stm.setString(3, contact);
                                stm.setInt(4, userId);
                                stm.execute();
                                success = true;
                        } else{
                                String query = "UPDATE users SET username = ?, email = ?, contact = ?, password = ?
                                PreparedStatement stm = conn.prepareStatement(query);
                                stm.setString(1, username);
                                stm.setString(2, email);
                                stm.setString(3, contact);
                                stm.setString(4, DigestUtils.sha1Hex(password));
                                stm.setInt(5, userId);
                                stm.execute();
                                success = true;
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                        success = false;
                }
                return success;
        }
}
```

```
package dbutils;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

/**
 *
 * @author PNB01
 *
 * This class opens and closes connection to the database
 *
 */
public class DBUtil {
        private static String host_url = "";
        private static String username = "";
        private static String password = "";
        private static String driver = "";

        private final static void configConnection() throws IOException{
                Properties properties = new Properties();
                properties.load(Thread.currentThread().getContextClassLoader().getResourceAsStream("config.properti
                host_url = properties.getProperty("jdbc.host");
                username = properties.getProperty("jdbc.username");
                password = properties.getProperty("jdbc.password");
                driver = properties.getProperty("jdbc.driver");
        }

        public static Connection getConnection(){
                Connection conn = null;

                try{
                        configConnection();
                        Class.forName(driver);
                        conn = DriverManager.getConnection(host_url, username, password);
                } catch(Exception e){
                        e.printStackTrace();
                }

                return conn;
        }
```

```
        public static void closeConnection(Connection conn) {
                try {
                        conn.close();
                } catch (SQLException ex) {
                        ex.printStackTrace();
                }
        }
}

package dbutils;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

import models.FoodListModel;
import models.IntakeModel;
import models.MealPlanModel;
import models.MenuModel;

public class DBDataExtract {
        /**
         * This function returns the role of the user given their user ID from the Database
         * @param id     User's ID
         * @return role User's Role (0-admin, 1-teacher 2-student 3-patient)
         */
        public static MealPlanModel getDietReference(double tea){
                MealPlanModel mealPlan = new MealPlanModel();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT * FROM dietreference WHERE energy <= '" + tea + " ORDER BY energy AS
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                tea = rs.getDouble("energy");
                                mealPlan.setExVegA(rs.getDouble("vegA"));
                                mealPlan.setExVegB(rs.getDouble("vegB"));
                                mealPlan.setExFruit(rs.getDouble("fruit"));
                                mealPlan.setExMilk(rs.getDouble("milk"));
                                mealPlan.setExSugar(rs.getDouble("sugar"));
                                mealPlan.setExRice(rs.getDouble("rice"));
                                mealPlan.setExMeatL(rs.getDouble("meatL"));
                                mealPlan.setExMeatM(rs.getDouble("meatM"));
                                mealPlan.setExFat(rs.getDouble("fat"));
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return mealPlan;
        }

        public static int getLastWeek(int patientId){
                int week = 1;
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT week FROM weeklyupdate WHERE patient = '" + patientId + "';";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                week += 1;
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return week;
        }

        public static Date getLastDay(int patientId){
                int week = getLastWeek(patientId) - 1;
                Date date = null;
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT `date` FROM `weeklyupdate` INNER JOIN `samplemenu` ON samplemenu.wee
                                        + "WHERE patient = '" + patientId + "' AND week = '"+ week +"'
ORDER by date DESC ;";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        if(rs.next()){
                                Timestamp ts = rs.getTimestamp("date");
                                date = new Date(ts.getTime());
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
```

```
                e.printStackTrace();
        }
        return date;
}

public static Date getLastSampleDate(int patientId){

        int week = getLastWeek(patientId) - 1;
        Date date = null;
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT end FROM weeklyupdate WHERE patient = '" + patientId + "' AND week =
                                + week + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                if(rs.next()){
                        Timestamp ts = rs.getTimestamp("end");
                        date = new Date(ts.getTime());
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return date;
}

public static boolean isComplete(int patientId){
        int week = getLastWeek(patientId) - 1;
        Date date = null;
        Date date2 = null;
        boolean isSame = false;
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT date, end FROM weeklyupdate INNER JOIN samplemenu ON weekId = weekly
                                + "WHERE patient = '" + patientId + "' AND week = '"
                                + week + "' ORDER by date DESC ;";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                if(rs.next()){
                        Timestamp dts = rs.getTimestamp("date");
                        date = new Date(dts.getTime());
                        Timestamp ets = rs.getTimestamp("end");
                        date2 = new Date(ets.getTime());
                        if(date.equals(date2)){
                                isSame = true;
                        }
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return isSame;
}

public static Date getStartDay(int patientId){
        int week = getLastWeek(patientId) - 1;
        Date date = null;
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT start FROM weeklyupdate WHERE patient = '" + patientId + "' AND week
                                + week + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                if(rs.next()){
                        Timestamp ts = rs.getTimestamp("start");
                        date = new Date(ts.getTime());
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return date;
}

public static ArrayList<MealPlanModel> getWeeklyReport(int id){
        ArrayList<MealPlanModel> weeklyMealPlans = new ArrayList<MealPlanModel>();
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM weeklyupdate"
                                + " WHERE patient = '" + id + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        MealPlanModel mealPlan = new MealPlanModel();
                        mealPlan.setId(rs.getInt("id"));
                        mealPlan.setPatientId(rs.getInt("patient"));
                        mealPlan.setWeek(rs.getInt("week"));
                        Timestamp tsStart = rs.getTimestamp("start");
                        Timestamp tsEnd = rs.getTimestamp("end");
                        mealPlan.setStart(new Date(tsStart.getTime()));
                        mealPlan.setEnd(new Date(tsEnd.getTime()));
                        mealPlan.setHeight(rs.getDouble("height"));
```

```
                        mealPlan.setWeight(rs.getDouble("weight"));
                        mealPlan.setBmi(rs.getDouble("bmi"));
                        mealPlan.setClassification(rs.getString("classification"));
                        mealPlan.setDbw(rs.getDouble("desiredBodyWeight"));
                        mealPlan.setActivity(rs.getString("activity"));
                        mealPlan.setTea(rs.getDouble("totalEnergyAllowance"));
                        mealPlan.setgCHO(rs.getDouble("gc"));
                        mealPlan.setgPRO(rs.getDouble("gp"));
                        mealPlan.setgFAT(rs.getDouble("gf"));
                        mealPlan.setExVegA(rs.getDouble("vegA"));
                        mealPlan.setExVegB(rs.getDouble("vegB"));
                        mealPlan.setExFruit(rs.getDouble("fruit"));
                        mealPlan.setExMilk(rs.getDouble("milk"));
                        mealPlan.setExSugar(rs.getDouble("sugar"));
                        mealPlan.setExRice(rs.getDouble("rice"));
                        mealPlan.setExMeatL(rs.getDouble("meatL"));
                        mealPlan.setExMeatM(rs.getDouble("meatM"));
                        mealPlan.setExFat(rs.getDouble("fat"));
                        weeklyMealPlans.add(mealPlan);
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return weeklyMealPlans;
}

public static MealPlanModel getReport(int id, int week){
        MealPlanModel mealPlan = new MealPlanModel();
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM weeklyupdate WHERE patient = '" + id + "' AND week = '" + wee
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        mealPlan.setId(rs.getInt("id"));
                        mealPlan.setPatientId(rs.getInt("patient"));
                        mealPlan.setWeek(rs.getInt("week"));
                        mealPlan.setHeight(rs.getDouble("height"));
                        mealPlan.setWeight(rs.getDouble("weight"));
                        mealPlan.setBmi(rs.getDouble("bmi"));
                        mealPlan.setClassification(rs.getString("classification"));
                        mealPlan.setDbw(rs.getDouble("desiredBodyWeight"));
                        mealPlan.setActivity(rs.getString("activity"));
                        mealPlan.setTea(rs.getDouble("totalEnergyAllowance"));
                        mealPlan.setgCHO(rs.getDouble("gc"));
                        mealPlan.setgPRO(rs.getDouble("gp"));
                        mealPlan.setgFAT(rs.getDouble("gf"));
                        mealPlan.setExVegA(rs.getDouble("vegA"));
                        mealPlan.setExVegB(rs.getDouble("vegB"));
                        mealPlan.setExFruit(rs.getDouble("fruit"));
                        mealPlan.setExMilk(rs.getDouble("milk"));
                        mealPlan.setExSugar(rs.getDouble("sugar"));
                        mealPlan.setExRice(rs.getDouble("rice"));
                        mealPlan.setExMeatL(rs.getDouble("meatL"));
                        mealPlan.setExMeatM(rs.getDouble("meatM"));
                        mealPlan.setExFat(rs.getDouble("fat"));
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return mealPlan;
}

public static MealPlanModel getMealPlan(int id){
        MealPlanModel mealPlan = new MealPlanModel();
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM weeklyupdate WHERE ID = '" + id + "';";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        mealPlan.setId(rs.getInt("id"));
                        mealPlan.setPatientId(rs.getInt("patient"));
                        mealPlan.setWeek(rs.getInt("week"));
                        mealPlan.setHeight(rs.getDouble("height"));
                        mealPlan.setWeight(rs.getDouble("weight"));
                        mealPlan.setBmi(rs.getDouble("bmi"));
                        mealPlan.setClassification(rs.getString("classification"));
                        mealPlan.setDbw(rs.getDouble("desiredBodyWeight"));
                        mealPlan.setActivity(rs.getString("activity"));
                        mealPlan.setTea(rs.getDouble("totalEnergyAllowance"));
                        mealPlan.setgCHO(rs.getDouble("gc"));
                        mealPlan.setgPRO(rs.getDouble("gp"));
                        mealPlan.setgFAT(rs.getDouble("gf"));
                        mealPlan.setExVegA(rs.getDouble("vegA"));
                        mealPlan.setExVegB(rs.getDouble("vegB"));
                        mealPlan.setExFruit(rs.getDouble("fruit"));
                        mealPlan.setExMilk(rs.getDouble("milk"));
                        mealPlan.setExSugar(rs.getDouble("sugar"));
                        mealPlan.setExRice(rs.getDouble("rice"));
```

```
                        mealPlan.setExMeatL(rs.getDouble("meatL"));
                        mealPlan.setExMeatM(rs.getDouble("meatM"));
                        mealPlan.setExFat(rs.getDouble("fat"));
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return mealPlan;
}

public static ArrayList<FoodListModel> getFoodList(){
        ArrayList<FoodListModel> foodList = new ArrayList<FoodListModel>();
        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT * FROM `foodexchange` ORDER BY `ID` ASC";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        FoodListModel food = new FoodListModel();
                        food.setId(rs.getInt("id"));
                        food.setGroup(rs.getString("food_group"));
                        food.setName(rs.getString("name"));
                        food.setgCHO(rs.getDouble("CHO"));
                        food.setgPRO(rs.getDouble("PRO"));
                        food.setgFAT(rs.getDouble("FAT"));
                        food.setWtAP(rs.getInt("weight_AP"));
                        food.setWtEP(rs.getInt("weight_EP"));
                        food.setEnergy(rs.getDouble("energy"));
                        food.setMeasure(rs.getString("measure"));
                        foodList.add(food);
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return foodList;
}

public static MenuModel getSampleMenu(int id, Date date){
        MenuModel menu = null;
        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT samplemenu.id, date, weekId, breakfast, lunch, dinner"
                                + " FROM weeklyupdate INNER JOIN samplemenu ON weekId = weeklyupdate.ID "
                                + "WHERE patient = '" + id + "' ORDER by date DESC ;";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                while(rs.next()){
                        String day1 = "";
                        String day2 = "";
                        day1 = dateFormat.format(date);
                        Timestamp ts = rs.getTimestamp("date");
                        day2 = dateFormat.format(ts.getTime());
                        if(day1.equals(day2)){
                                menu = new MenuModel();
                                menu.setId(rs.getInt("id"));
                                menu.setDate(date);
                                menu.setWeekId(rs.getInt("weekId"));
                                menu.setBreakfast(rs.getString("breakfast"));
                                menu.setLunch(rs.getString("lunch"));
                                menu.setDinner(rs.getString("dinner"));
                        }
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return menu;
}

public static boolean hasSubmittedIntake(int menuId){
        boolean hasSubmitted=false;

        try{
                Connection conn = DBUtil.getConnection();
                String query = "SELECT menuId FROM dailyintake WHERE menuId = '" +menuId+ "' ;";
                Statement stm = conn.prepareStatement(query);
                ResultSet rs = stm.executeQuery(query);
                if(rs.next()){
                        hasSubmitted = true;
                }
                DBUtil.closeConnection(conn);
        } catch(Exception e){
                e.printStackTrace();
        }
        return hasSubmitted;
}

public static IntakeModel getIntakes(int id, Date date){
```

```
                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                String date1 = dateFormat.format(date);
                String date2 = "";
                IntakeModel intake = new IntakeModel();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT dailyintake.id, menuId, foodItem, feedback, CHO, PRO, dailyintake.FA
                                        + " FROM dailyintake, samplemenu, weeklyupdate "
                                        + "WHERE patient = '" + id + "' AND menuId = samplemenu.id AND weekId=weekl
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                Timestamp ts = rs.getTimestamp("date");
                                date2 = dateFormat.format(ts.getTime());
                                if(date1.equals(date2)){
                                        intake.setId(rs.getInt("id"));
                                        intake.setMenuId(rs.getInt("menuId"));
                                        intake.setFoodItem(rs.getString("foodItem"));
                                        intake.setFeedback(rs.getString("feedback"));
                                        intake.setgCHO(rs.getInt("CHO"));
                                        intake.setgPRO(rs.getInt("PRO"));
                                        intake.setgFAT(rs.getInt("FAT"));
                                        intake.setEnergy(rs.getInt("energy"));
                                }
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return intake;
        }

        public static ArrayList<Date> getAllDates(int id){
                ArrayList<Date> dates = new ArrayList<Date>();
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT date"
                                        + " FROM dailyintake, samplemenu, weeklyupdate "
                                        + "WHERE patient = '" + id + "' AND menuId = samplemenu.id AND weekId=weekl
                                                        + " ORDER BY date DESC;";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                Timestamp ts = rs.getTimestamp("date");
                                Date date = new Date(ts.getTime());
                                dates.add(date);
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return dates;
        }

        public static String getCode(Date date){
                String code = "ERROR";
                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                String date1 = dateFormat.format(date);
                String date2 = "";
                try{
                        Connection conn = DBUtil.getConnection();
                        String query = "SELECT *"
                                        + " FROM dailycode;";
                        Statement stm = conn.prepareStatement(query);
                        ResultSet rs = stm.executeQuery(query);
                        while(rs.next()){
                                Timestamp ts = rs.getTimestamp("date");
                                date2 = dateFormat.format(ts.getTime());
                                if(date1.equals(date2)){
                                        code = rs.getString("code");
                                }
                        }
                        DBUtil.closeConnection(conn);
                } catch(Exception e){
                        e.printStackTrace();
                }
                return code;
        }
}


package models;

public class FoodListModel {
        private int id;
        private String group;
        private String name;
        private double gCHO;
        private double gPRO;
        private double gFAT;
        private double energy;
        private int wtAP;
```

```
private int wtEP;
private String measure;

public FoodListModel(){
        this.id = 0;
        this.group = "";
        this.name = "";
        this.gCHO = 0;
        this.gPRO = 0;
        this.gFAT = 0;
        this.wtAP = 0;
        this.wtEP = 0;
        this.energy = 0;
        this.measure = "";
}

public FoodListModel(int id, String group, String name, double gCHO, double gPRO, double gFAT,
                double energy, int wtAP, int wtEP, String measure){
        this.id = id;
        this.group = group;
        this.name = name;
        this.gCHO = gCHO;
        this.gPRO = gPRO;
        this.gFAT = gFAT;
        this.wtAP = wtAP;
        this.wtEP = wtEP;
        this.energy = energy;
        this.measure = measure;
}


public int getId() {
        return id;
}
public void setId(int id) {
        this.id = id;
}
public String getGroup() {
        return group;
}
public void setGroup(String group) {
        this.group = group;
}
public String getName() {
        return name;
}
public void setName(String name) {
        this.name = name;
}
public double getgCHO() {
        return gCHO;
}
public void setgCHO(double gCHO) {
        this.gCHO = gCHO;
}
public double getgPRO() {
        return gPRO;
}
public void setgPRO(double gPRO) {
        this.gPRO = gPRO;
}
public double getgFAT() {
        return gFAT;
}
public void setgFAT(double gFAT) {
        this.gFAT = gFAT;
}
public double getEnergy() {
        return energy;
}
public void setEnergy(double energy) {
        this.energy = energy;
}
public int getWtAP() {
        return wtAP;
}
public void setWtAP(int wtAP) {
        this.wtAP = wtAP;
}
public int getWtEP() {
        return wtEP;
}
public void setWtEP(int wtEP) {
        this.wtEP = wtEP;
}
public String getMeasure() {
        return measure;
}
public void setMeasure(String measure) {
        this.measure = measure;
}
```

```
}

package models;

import java.util.ArrayList;
import java.util.Date;
import java.util.StringTokenizer;

public class IntakeModel {
        private int id;
        private int menuId;
        private ArrayList<String> foodItem;
        private String feedback;
        private int gCHO;
        private int gPRO;
        private int gFAT;
        private int energy;

        public IntakeModel(){
                this.id = 0;
                this.menuId = 0;
                this.foodItem = null;
                this.feedback = "";
                this.gCHO = 0;
                this.gPRO = 0;
                this.gFAT = 0;
                this.energy = 0;
        }

        public IntakeModel(int id, int menuId, String foodItem, String feedback,
                        int gCHO, int gPRO, int gFAT, int energy){
                this.id = id;
                this.menuId = menuId;
                setFoodItem(foodItem);
                this.feedback = feedback;
                this.gCHO = gCHO;
                this.gPRO = gPRO;
                this.gFAT = gFAT;
                this.energy = energy;
        }

        public int getId() {
                return id;
        }

        public void setId(int id) {
                this.id = id;
        }

        public void setFoodItem(String foodItem) {
                StringTokenizer st = new StringTokenizer(foodItem, "||");
                this.foodItem = new ArrayList<String>();
                while (st.hasMoreElements()) {
                        this.foodItem.add(st.nextToken());
                }
        }

        public int getMenuId() {
                return menuId;
        }

        public void setMenuId(int menuId) {
                this.menuId = menuId;
        }

        public ArrayList<String> getFoodItem() {
                return foodItem;
        }

        public String getFeedback() {
                return feedback;
        }

        public void setFeedback(String feedback) {
                this.feedback = feedback;
        }

        public int getgCHO() {
                return gCHO;
        }

        public void setgCHO(int gCHO) {
                this.gCHO = gCHO;
        }

        public int getgPRO() {
                return gPRO;
        }

        public void setgPRO(int gPRO) {
                this.gPRO = gPRO;
```

```
        }

        public int getgFAT() {
                return gFAT;
        }

        public void setgFAT(int gFAT) {
                this.gFAT = gFAT;
        }

        public int getEnergy() {
                return energy;
        }

        public void setEnergy(int energy) {
                this.energy = energy;
        }
}


package models;

import java.util.Date;

public class MealPlanModel {
        private int id;
        private int patientId;
        private int week;
        private Date start;
        private Date end;
        private double height;
        private double weight;
        private double bmi;
        private String classification;
        private double dbw;
        private String activity;
        private double tea;
        private double gCHO;
        private double gPRO;
        private double gFAT;

        public int getId() {
                return id;
        }

        public void setId(int id) {
                this.id = id;
        }

        public int getPatientId() {
                return patientId;
        }

        public void setPatientId(int patientId) {
                this.patientId = patientId;
        }

        public double getHeight() {
                return height;
        }

        public void setHeight(double height) {
                this.height = height;
        }

        public double getWeight() {
                return weight;
        }

        public void setWeight(double weight) {
                this.weight = weight;
        }

        public double getBmi() {
                return bmi;
        }

        public void setBmi(double bmi) {
                this.bmi = bmi;
        }

        public String getClassification() {
                return classification;
        }

        public void setClassification(String classification) {
                this.classification = classification;
        }

        public double getDbw() {
                return dbw;
        }
```

```
public void setDbw(double dbw) {
        this.dbw = dbw;
}

public String getActivity() {
        return activity;
}

public void setActivity(int activity) {
        switch(activity){
                case 1: this.activity = "bed_rest";
                        break;
                case 2: this.activity = "sedentary";
                        break;
                case 3: this.activity = "light";
                        break;
                case 4: this.activity = "moderate";
                        break;
                case 5: this.activity = "very_active";
                        break;
        }
}

public void setActivity(String activity) {
        this.activity = activity;
}

public double getTea() {
        return tea;
}

public void setTea(double tea) {
        this.tea = tea;
}

public double getgCHO() {
        return gCHO;
}

public void setgCHO(double gCHO) {
        this.gCHO = gCHO;
}

public double getgPRO() {
        return gPRO;
}

public void setgPRO(double gPRO) {
        this.gPRO = gPRO;
}

public double getgFAT() {
        return gFAT;
}

public void setgFAT(double gFAT) {
        this.gFAT = gFAT;
}
private double exVegA;
private double exVegB;
private double exFruit;
private double exMilk;
private double exSugar;
private double exRice;
private double exMeatL;
private double exMeatM;
private double exFat;

public MealPlanModel(double exVegA, double exVegB, double exFruit, double exMilk, double exSugar,
                                        double exRice, double exMeatL, double exMeatM, double exFat){
        this.exVegA = exVegA;
        this.exVegB = exVegB;
        this.exFruit = exFruit;
        this.exMilk = exMilk;
        this.exSugar = exSugar;
        this.exRice = exRice;
        this.exMeatL = exMeatL;
        this.exMeatM = exMeatM;
        this.exFat = exFat;
}

public MealPlanModel(int id, int patientId, int week, Date start, Date end, double height, double weight,
                double bmi, String classification, double dbw, String activity, double tea,
                double gCHO, double gPRO, double gFAT,
                double exVegA, double exVegB, double exFruit, double exMilk, double exSugar,
                double exRice, double exMeatL, double exMeatM, double exFat){
        this.id = id;
        this.patientId = patientId;
        this.week = week;
        this.start = start;
        this.start = end;
```

```
        this.height = height;
        this.weight = weight;
        this.bmi = bmi;
        this.classification = classification;
        this.dbw = dbw;
        this.activity = activity;
        this.tea = tea;
        this.gCHO = gCHO;
        this.gPRO = gPRO;
        this.gFAT = gFAT;
        this.exVegA = exVegA;
        this.exVegB = exVegB;
        this.exFruit = exFruit;
        this.exMilk = exMilk;
        this.exSugar = exSugar;
        this.exRice = exRice;
        this.exMeatL = exMeatL;
        this.exMeatM = exMeatM;
        this.exFat = exFat;
}

public MealPlanModel(){
        this.id = 0;
        this.patientId = 0;
        this.week = 0;
        this.start = new Date();
        this.end = new Date();
        this.height = 0;
        this.weight = 0;
        this.bmi = 0;
        this.classification = "";
        this.dbw = 0;
        this.activity = "";
        this.tea = 0;
        this.gCHO = 0;
        this.gPRO = 0;
        this.gFAT = 0;

        this.exVegA = 0;
        this.exVegB = 0;
        this.exFruit = 0;
        this.exMilk = 0;
        this.exSugar = 0;
        this.exRice = 0;
        this.exMeatL = 0;
        this.exMeatM = 0;
        this.exFat = 0;
}

public double getExVegA() {
        return exVegA;
}
public void setExVegA(double exVegA) {
        this.exVegA = exVegA;
}
public double getExVegB() {
        return exVegB;
}
public void setExVegB(double exVegB) {
        this.exVegB = exVegB;
}
public double getExFruit() {
        return exFruit;
}
public void setExFruit(double exFruit) {
        this.exFruit = exFruit;
}
public int getWeek() {
        return week;
}

public void setWeek(int week) {
        this.week = week;
}

public Date getStart() {
        return start;
}

public void setStart(Date start) {
        this.start = start;
}

public Date getEnd() {
        return end;
}

public void setEnd(Date end) {
        this.end = end;
}

public double getExMilk() {
```

```
                return exMilk;
        }
        public void setExMilk(double exMilk) {
                this.exMilk = exMilk;
        }
        public double getExSugar() {
                return exSugar;
        }
        public void setExSugar(double exSugar) {
                this.exSugar = exSugar;
        }
        public double getExRice() {
                return exRice;
        }
        public void setExRice(double exRice) {
                this.exRice = exRice;
        }
        public double getExMeatL() {
                return exMeatL;
        }
        public void setExMeatL(double exMeatL) {
                this.exMeatL = exMeatL;
        }
        public double getExMeatM() {
                return exMeatM;
        }
        public void setExMeatM(double exMeatM) {
                this.exMeatM = exMeatM;
        }
        public double getExFat() {
                return exFat;
        }
        public void setExFat(double exFat) {
                this.exFat = exFat;
        }


}

package models;

import java.util.ArrayList;
import java.util.Date;
import java.util.StringTokenizer;

public class MenuModel {
        private int id;
        private Date date;
        private int weekId;
        private ArrayList<String> breakfast;
        private ArrayList<String> lunch;
        private ArrayList<String> dinner;

        public MenuModel(){
                this.id = 0;
                this.date = null;
                this.weekId = 0;
                this.breakfast = null;
                this.lunch = null;
                this.dinner = null;
        }

        public MenuModel(int id, Date date, int weekId, String breakfast, String lunch, String dinner){
                this.id = id;
                this.date = date;
                this.weekId = weekId;
                setBreakfast(breakfast);
                setLunch(lunch);
                setDinner(dinner);
        }

        public int getId() {
                return id;
        }

        public void setId(int id) {
                this.id = id;
        }

        public Date getDate() {
                return date;
        }

        public void setDate(Date date) {
                this.date = date;
        }

        public int getWeekId() {
                return weekId;
        }
```

```
        public void setWeekId(int weekId) {
                this.weekId = weekId;
        }

        public ArrayList<String> getBreakfast() {
                return breakfast;
        }

        public void setBreakfast(String breakfast) {

                StringTokenizer st = new StringTokenizer(breakfast, "||");
                this.breakfast = new ArrayList<String>();
                while (st.hasMoreElements()) {
                        this.breakfast.add(st.nextToken());
                }
        }

        public ArrayList<String> getLunch() {
                return lunch;
        }

        public void setLunch(String lunch) {
                StringTokenizer st = new StringTokenizer(lunch, "||");
                this.lunch = new ArrayList<String>();
                while (st.hasMoreElements()) {
                        this.lunch.add(st.nextToken());
                }
        }

        public ArrayList<String> getDinner() {
                return dinner;
        }

        public void setDinner(String dinner) {
                StringTokenizer st = new StringTokenizer(dinner, "||");
                this.dinner = new ArrayList<String>();
                while (st.hasMoreElements()) {
                        this.dinner.add(st.nextToken());
                }
        }
}

package models;

import java.util.Date;

public class PatientModel extends UserModel{
        private int student;
        private String fname;
        private String mname;
        private String lname;
        private String sex;
        private Date birthdate;


        public String getSex() {
                return sex;
        }

        public void setSex(String sex) {
                this.sex = sex;
        }

        public Date getBirthdate() {
                return birthdate;
        }

        public void setBirthdate(Date birthdate) {
                this.birthdate = birthdate;
        }

        public int getStudent() {
                return student;
        }

        public void setStudent(int student) {
                this.student = student;
        }

        public PatientModel(){
                super();
                student = 0;
                fname = "";
                mname = "";
                lname = "";
        }

        public PatientModel(int id, String username, String email, String password,
                        int role, boolean deactivated, int student, String fname, String mname, String lname, Strin
                        String sex, Date birthdate){
                super(id, username, email, contact, password, role, deactivated);
```

```
                this.student = student;
                this.fname = fname;
                this.mname = mname;
                this.lname = lname;
                this.sex = sex;
                this.birthdate = birthdate;
        }

        public String getFname() {
                return fname;
        }

        public void setFname(String fname) {
                this.fname = fname;
        }

        public String getMname() {
                return mname;
        }

        public void setMname(String mname) {
                this.mname = mname;
        }

        public String getLname() {
                return lname;
        }

        public void setLname(String lname) {
                this.lname = lname;
        }
}


package models;

public class StudentModel extends UserModel{
        private int teacher;
        private String fname;
        private String mname;
        private String lname;

        public int getTeacher() {
                return teacher;
        }

        public void setTeacher(int teacher) {
                this.teacher = teacher;
        }

        public StudentModel(){
                super();
                teacher = 0;
                fname = "";
                mname = "";
                lname = "";
        }

        public StudentModel(int id, String username, String email, String password,
                        int role, boolean deactivated, int teacher, String fname, String mname, String lname, Strin
                super(id, username, email, contact, password, role, deactivated);
                this.teacher = teacher;
                this.fname = fname;
                this.mname = mname;
                this.lname = lname;
        }

        public String getFname() {
                return fname;
        }

        public void setFname(String fname) {
                this.fname = fname;
        }

        public String getMname() {
                return mname;
        }

        public void setMname(String mname) {
                this.mname = mname;
        }

        public String getLname() {
                return lname;
        }

        public void setLname(String lname) {
                this.lname = lname;
        }
}
```

```
package models;

public class TeacherModel extends UserModel{
        private String fname;
        private String mname;
        private String lname;

        public TeacherModel(){
                super();
                fname = "";
                mname = "";
                lname = "";
        }

        public TeacherModel(int id, String username, String email, String password,
                        int role, boolean deactivated, String fname, String mname, String lname, String contact){
                super(id, username, email, contact, password, role, deactivated);
                this.fname = fname;
                this.mname = mname;
                this.lname = lname;
        }

        public String getFname() {
                return fname;
        }

        public void setFname(String fname) {
                this.fname = fname;
        }

        public String getMname() {
                return mname;
        }

        public void setMname(String mname) {
                this.mname = mname;
        }

        public String getLname() {
                return lname;
        }

        public void setLname(String lname) {
                this.lname = lname;
        }
}


package models;

public class UserModel {
        private int id;
        private String username;
        private String email;
        private String password;
        private String contact;
        private int role;
        private boolean deactivated;

        public UserModel(){
                this.id = 0;
                this.username = "";
                this.email = "";
                this.contact = "";
                this.password = "";
                this.role = 5;
                this.deactivated = true;
        }

        public UserModel(int id, String username, String email, String contact,
                        String password, int role, boolean deactivated){
                this.id = id;
                this.username = username;
                this.email = email;
                this.contact = contact;
                this.password = password;
                this.role = role;
                this.deactivated = deactivated;
        }

        public String getContact() {
                return contact;
        }

        public void setContact(String contact) {
                this.contact = contact;
        }

        public int getId() {
                return id;
        }
```

```
        public void setId(int id) {
                this.id = id;
        }

        public String getUsername() {
                return username;
        }

        public void setUsername(String username) {
                this.username = username;
        }

        public String getEmail() {
                return email;
        }

        public void setEmail(String email) {
                this.email = email;
        }

        public String getPassword() {
                return password;
        }

        public void setPassword(String password) {
                this.password = password;
        }

        public int getRole() {
                return role;
        }

        public void setRole(int role) {
                this.role = role;
        }

        public boolean isDeactivated() {
                return deactivated;
        }

        public void setDeactivated(boolean deactivated) {
                this.deactivated = deactivated;
        }
}

package operations;

public class AlertMessage {
        private boolean type;
        private String head;
        private String subhead;
        private String title;
        private String redirect;

        /**
         * Creates an alert message
         *
         * @param type          true or false
         * @param head          main header for alert
         * @param subhead       message
         * @param title         html title
         * @param redirect      link to redirect
         */

        public AlertMessage(boolean type, String head, String subhead, String title, String redirect){
                this.type = type;
                if(head == null || head.length() == 0){
                        this.head = "ERROR! ";
                        this.subhead = "Returning to Log-in";
                } else{
                        this.head = head;
                        this.subhead = subhead;
                }
                this.title = title;
                if(redirect == null || redirect.length() == 0){
                        this.redirect = "3;url=/MealPlanning/Login";
                } else{
                        this.redirect = redirect;
                }
        }

        public boolean isType() {
                return type;
        }
        public void setType(boolean type) {
                this.type = type;
        }
        public String getHead() {
                return head;
        }
        public void setHead(String head) {
```

```
                this.head = head;
        }
        public String getSubhead() {
                return subhead;
        }
        public void setSubhead(String subhead) {
                this.subhead = subhead;
        }
        public String getTitle() {
                return title;
        }
        public void setTitle(String title) {
                this.title = title;
        }
        public String getRedirect() {
                return redirect;
        }
        public void setRedirect(String redirect) {
                this.redirect = redirect;
        }
}

package operations;

import dbutils.DBDataExtract;
import models.MealPlanModel;

public class DietCalculation {
        private double desired_body_weight;
        private double body_mass_index;
        private String classification;
        private double total_energy_allowance;
        private double wtCHO;
        private double wtPRO;
        private double wtFAT;

        private double exVegA;
        private double exVegB;
        private double exFruit;
        private double exMilk;
        private double exSugar;
        private double exRice;
        private double exMeatL;
        private double exMeatM;
        private double exFat;
        private double computedEnergy;
        private double computedCHO;

        private double height;
        private double weight;
        private String activity;
        private double computedPRO;
        private double computedFAT;

        public double getComputedCHO() {
                return computedCHO;
        }

        public double getComputedPRO() {
                return computedPRO;
        }

        public double getComputedFAT() {
                return computedFAT;
        }

        public DietCalculation(double height, double weight, int activity, int pdCHO, int pdPRO, int pdFAT){
                this.height = height;
                this.weight = weight;
                calculateBMI(height, weight);
                this.desired_body_weight = calculateDBW(height);
                this.total_energy_allowance = calculateTEA(activity);
                calculateNutrientWeight(pdCHO, pdPRO, pdFAT);
                calculateExchanges();
        }

        public double getHeight(){
                return this.height;
        }

        public double getWeight(){
                return this.weight;
        }

        public String getActivity(){
                return this.activity;
        }

        public double getDBW(){
                return desired_body_weight;
        }
```

```
public double getTEA(){
        return total_energy_allowance;
}

public double getComputedEnergy() {
        return computedEnergy;
}

public double getExVegA() {
        return exVegA;
}

public double getExVegB() {
        return exVegB;
}

public double getExFruit() {
        return exFruit;
}

public double getExMilk() {
        return exMilk;
}

public double getExSugar() {
        return exSugar;
}

public double getExRice() {
        return exRice;
}

public double getExMeatL() {
        return exMeatL;
}

public double getExMeatM() {
        return exMeatM;
}

public double getExFat() {
        return exFat;
}

public double getWtCHO() {
        return wtCHO;
}

public double getWtPRO() {
        return wtPRO;
}

public double getWtFAT() {
        return wtFAT;
}

private static double calculateDBW(double height){
        return roundToHalf((height - 100) * 0.9);
}


private static double roundToHalf(double d) {
    return Math.round(d * 2) / 2.0;
}

private static double roundToFive(double d) {
    return Math.round(d / 5) * 5;
}

private double calculateTEA(int activity){
        double total_energy_allowance = 0;
        switch(activity){
                case 1:
                        total_energy_allowance = (this.desired_body_weight * 27.5);
                        this.activity = "bed rest";
                        break;
                case 2:
                        total_energy_allowance = (this.desired_body_weight * 30.0);
                        this.activity = "sedentary";
                        break;
                case 3:
                        total_energy_allowance = (this.desired_body_weight * 35.0);
                        this.activity = "light";
                        break;
                case 4:
                        total_energy_allowance = (this.desired_body_weight * 40.0);
                        this.activity = "moderate";
                        break;
                case 5:
                        total_energy_allowance = (this.desired_body_weight * 45.0);
                        this.activity = "very active";
```

```
                        break;
                }
                return Math.round(total_energy_allowance);
        }

        private void calculateNutrientWeight(int pdCHO, int pdPRO, int pdFAT){
                this.wtCHO = roundToFive(Math.round(this.total_energy_allowance * pdCHO / 400));
                this.wtPRO = roundToFive(Math.round(this.total_energy_allowance * pdPRO / 400));
                this.wtFAT = roundToFive(Math.round(this.total_energy_allowance * pdFAT / 900));
        }

        private void calculateBMI(double height, double weight){
                this.body_mass_index = roundToHalf(weight/((height/100) * (height/100)));
                if(this.body_mass_index < 18.5){
                        this.classification = "Underweight";
                }else if(this.body_mass_index >= 18.5 && this.body_mass_index < 25){
                        this.classification = "Normal";
                }else if(this.body_mass_index >= 25 && this.body_mass_index <= 30){
                        this.classification = "Overweight";
                }else{
                        this.classification = "Obese";
                }
        }

        public double getBMI() {
                return body_mass_index;
        }

        public String getClassification() {
                return classification;
        }

        private void calculateExchanges(){
                MealPlanModel mealPlan = DBDataExtract.getDietReference(this.total_energy_allowance);
                this.exVegA = mealPlan.getExVegA();
                this.exVegB = mealPlan.getExVegB();
                this.exFruit = mealPlan.getExFruit();
                this.exMilk = mealPlan.getExMilk();
                this.exSugar = mealPlan.getExSugar();

                double tempCHO = this.wtCHO;
                double tempPRO = this.wtPRO;
                double tempFAT = this.wtFAT;
                tempCHO -= 6;
                tempPRO -= 2;
                tempCHO -= (this.exFruit * 10);
                tempCHO -= (this.exMilk * 12);
                tempPRO -= (this.exMilk * 8);
                tempFAT -= (this.exMilk * 10);
                tempCHO -= (this.exSugar * 5);
                this.exRice = roundToHalf(tempCHO/23);
                tempPRO -= (this.exRice * 2);
                double tempMeat = roundToHalf(tempPRO/8);
                this.exMeatL = mealPlan.getExMeatL();
                this.exMeatM = tempMeat - this.exMeatL;
                tempFAT -= (this.exMeatL * 1);
                tempFAT -= (this.exMeatM * 6);
                this.exFat = roundToHalf(tempFAT/5);

                this.computedEnergy = 32 + (this.exFruit * 40) + (this.exMilk * 170) + (this.exSugar * 20) +
                                (this.exRice * 100) + (this.exMeatL * 41) + (this.exMeatM * 86) + (this.exFat * 45);
                this.computedCHO = 6 + (this.exFruit * 10) + (this.exMilk * 12) + (this.exSugar * 5) +
                                (this.exRice * 23);
                this.computedPRO = 2 + (this.exMilk * 8) +
                                (this.exRice * 2) + (this.exMeatL * 8) + (this.exMeatM * 8);
                this.computedFAT = (this.exMilk * 10) + (this.exMeatL * 1) + (this.exMeatM * 6) + (this.exFat * 5);
        }

}


package operations;

import java.util.Random;

public class CodeGenerator {
        private static final char[] code;
        private final Random random = new Random();
    private final char[] buf;
    private String randomCode;

    static {
        StringBuilder tmp = new StringBuilder();
        for (char ch = '0'; ch <= '9'; ch++) {
            tmp.append(ch);
        }
        for (char ch = 'a'; ch <= 'z'; ch++) {
            tmp.append(ch);
        }
        code = tmp.toString().toCharArray();
    }
```

```
    public CodeGenerator(int length) {
        if (length < 1) {
            throw new IllegalArgumentException("length < 1: " + length);
        }
        buf = new char[length];
        for (int i = 0; i < buf.length; i++) {
            buf[i] = code[random.nextInt(code.length)];
        }
        this.randomCode =  new String(buf);
    }

    public String getRandomCode(){
        return this.randomCode;
    }
}
```

```
package operations;

import java.util.Properties;

import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class SendEmail{

        public static boolean emailPassword(String to, String password){
            Properties properties = System.getProperties();
            final String USERNAME = "weightmanagementportal@gmail.com";
            String host = "smtp.gmail.com";
            String from = "weightmanagementportal@gmail.com";
            String pass = "mealplan";
            properties.put("mail.smtp.ssl.enable", "true");
            properties.put("mail.smtp.host", host);
            properties.put("mail.smtp.user", from);
            properties.put("mail.smtp.password", pass);
            properties.put("mail.smtp.port", "465");
            properties.put("mail.smtp.auth", "true");
            //properties.put("mail.debug", "true");

            Session session = Session.getInstance(properties);
            session.setDebug(true);
            try {

                MimeMessage mime = new MimeMessage(session);
                mime.setFrom(new InternetAddress(USERNAME));

                mime.setRecipient(Message.RecipientType.TO, new InternetAddress(to));
                mime.setSubject("Reset Password");
                mime.setText("Your new password is: " + password);
                Transport transport = session.getTransport("smtp");
                transport.connect(host, from, pass);
                transport.sendMessage(mime, mime.getAllRecipients());
                transport.close();
                //System.out.println("Sent");
                return true;
            } catch (MessagingException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
                return false;
            }
        }
}
```

```
package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBInsert;
import operations.AlertMessage;

/**
 * Servlet implementation class AddStudent
 */
@WebServlet("/AddFoodItem")
public class AddFoodItem extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
```

```
 */
public AddFoodItem() {
    super();
    // TODO Auto-generated constructor stub
}

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
            // TODO Auto-generated method stub
            HttpSession session = request.getSession();
            if(null != session.getAttribute("role") && session.getAttribute("role").equals(1)){
                    RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Teach
                dispatcher.forward(request,response);
            } else if(session.getAttribute("role") == null){
                    response.sendRedirect("/MealPlanning/Log-in");
            } else{
                    RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                dispatcher.forward(request,response);
            }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
            // TODO Auto-generated method stub
            String group = request.getParameter("group");
            String name = request.getParameter("foodname");
            String sAP = request.getParameter("wtAP");
            String sEP = request.getParameter("wtEP");
            String measure = request.getParameter("measure");

            int wtAP = 0;
            int wtEP = 0;
            int cho = 0;
            int pro = 0;
            int fat = 0;
            int energy = 0;

            try{
                    wtAP = Integer.parseInt(sAP);
                    wtEP = Integer.parseInt(sEP);
            }catch(Exception e){

            }

            switch(group){
                    case "vegA":
                        break;
                    case "vegB":
                        cho = 3;
                        pro = 1;
                        energy = 16;
                        break;
                    case "fruit":
                        cho = 10;
                        energy = 40;
                        break;
                    case "milk_whole":
                        cho = 12;
                        pro = 8;
                        fat = 10;
                        energy = 170;
                        break;
                    case "milk_lowfat":
                        cho = 12;
                        pro = 8;
                        fat = 5;
                        energy = 125;
                        break;
                    case "milk_skimmed":
                        cho = 12;
                        pro = 8;
                        energy = 80;
                        break;
                    case "rice":
                        cho = 23;
                        pro = 2;
                        energy = 100;
                        break;
                    case "meatfish_lowfat":
                        pro = 8;
                        fat = 1;
                        energy = 41;
                        break;
                    case "meatfish_mediumfat":
                        pro = 8;
                        fat = 6;
                        energy = 86;
                        break;
```

```
                        case "meatfish_highfat":
                                pro = 8;
                                fat = 10;
                                energy = 122;
                                break;
                        case "sugar":
                                cho = 5;
                                energy = 20;
                                break;
                        case "fat":
                                fat = 5;
                                energy = 45;
                                break;
                }

                HttpSession session = request.getSession();
                boolean type = false;
                String head = "";
                String subhead = "";
                String title = "";
                String redirect = "";

                if(DBInsert.addFoodItem(name, group, wtAP, wtEP, measure, cho, pro, fat, energy)){
                        type = true;
                        head = "Success! Food Item has been added";
                        subhead = "Redirecting to Add Food Item page.";
                        title = "Food Item Added";
                        redirect = "3;url=/MealPlanning/Teacher/Add_Food_Item";
                } else{
                        type = false;
                        head = "Error encountered while adding new food item";
                        subhead = "Redirecting to Add Food Item page.";
                        title = "ERROR";
                        redirect = "3;url=/MealPlanning/Teacher/Add_Add_Food_Item";
                }
                session.setAttribute("alert", new AlertMessage(type, head, subhead, title, redirect));
                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainPage/Aler
            dispatcher.forward(request,response);
        }

}

package servlet;

import java.io.IOException;
import java.sql.Timestamp;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBInsert;
import dbutils.DBUserExtract;
import operations.AlertMessage;

/**
 * Servlet implementation class AddStudent
 */
@WebServlet("/AddPatient")
public class AddPatient extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AddPatient() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(2)){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Stude
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
```

```
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                String fname = request.getParameter("fname");
                String mname = request.getParameter("mname");
                String lname = request.getParameter("lname");
                String username = request.getParameter("username");
                String password = request.getParameter("password");
                String email = request.getParameter("email");
                String contact = request.getParameter("contact");
                String birthdate = request.getParameter("birthdate");
                String sex = request.getParameter("sex");
                String role = "3";
                Timestamp ts = new Timestamp(System.currentTimeMillis());

                try{
                    SimpleDateFormat dateFormat = new SimpleDateFormat("dd MMMMM, yyyy");
                    Date parsedDate = dateFormat.parse(birthdate);
                    ts = new java.sql.Timestamp(parsedDate.getTime());
                }catch(Exception e){//this generic but you can control another types of exception
                        e.printStackTrace();
                }

                HttpSession session = request.getSession();
                boolean type = false;
                String head = "";
                String subhead = "";
                String title = "";
                String redirect = "";

                if(DBUserExtract.isUserExisting(username, email)){
                        type = false;
                        head = "Error encountered while adding new patient";
                        subhead = "Redirecting to Add Patient page.";
                        title = "ERROR";
                        redirect = "3;url=/MealPlanning/Student/Add_Patient";
                } else{
                        if(DBInsert.addPatient(username, password, email, role, fname, mname, lname, contact,
                                        (Integer)session.getAttribute("id"), ts, sex)){
                                type = true;
                                head = "Success! Patient has been added";
                                subhead = "Redirecting to Add Patient page.";
                                title = "Patient Added";
                                redirect = "3;url=/MealPlanning/Student/Add_Patient";
                        } else{
                                type = false;
                                head = "Error encountered while adding new patient";
                                subhead = "Redirecting to Add Patient page.";
                                title = "ERROR";
                                redirect = "3;url=/MealPlanning/Teacher/Add_Patient";
                        }
                }
                session.setAttribute("alert", new AlertMessage(type, head, subhead, title, redirect));
                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainPage/Aler
            dispatcher.forward(request,response);
        }

}


package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBInsert;
import dbutils.DBUserExtract;
import operations.AlertMessage;

/**
 * Servlet implementation class AddStudent
 */
@WebServlet("/AddStudent")
public class AddStudent extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AddStudent() {
        super();
```

```
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(1)){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Teach
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                String fname = request.getParameter("fname");
                String mname = request.getParameter("mname");
                String lname = request.getParameter("lname");
                String username = request.getParameter("username");
                String password = request.getParameter("password");
                String email = request.getParameter("email");
                String contact = request.getParameter("contact");
                String role = "2";

                HttpSession session = request.getSession();
                boolean type = false;
                String head = "";
                String subhead = "";
                String title = "";
                String redirect = "";

                if(DBUserExtract.isUserExisting(username, email)){
                        type = false;
                        head = "Error encountered while adding new student";
                        subhead = "Redirecting to Add Student page.";
                        title = "ERROR";
                        redirect = "3;url=/MealPlanning/Teacher/Add_Student";
                } else{
                        if(DBInsert.addStudent(username, password, email, role, fname, mname, lname, contact,
                                        (Integer)session.getAttribute("id"))){
                                type = true;
                                head = "Success! Student has been added";
                                subhead = "Redirecting to Add Student page.";
                                title = "Student Added";
                                redirect = "3;url=/MealPlanning/Teacher/Add_Student";
                        } else{
                                type = false;
                                head = "Error encountered while adding new student";
                                subhead = "Redirecting to Add Student page.";
                                title = "ERROR";
                                redirect = "3;url=/MealPlanning/Teacher/Add_Student";
                        }
                }
                session.setAttribute("alert", new AlertMessage(type, head, subhead, title, redirect));
                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainPage/Aler
            dispatcher.forward(request,response);
        }

}


package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBInsert;
import dbutils.DBUserExtract;
import operations.AlertMessage;

/**
 * Servlet implementation class AddTeacher
 */
@WebServlet("/AddTeacher")
public class AddTeacher extends HttpServlet {
```

```
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AddTeacher() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(0)){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Admin
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                String fname = request.getParameter("fname");
                String mname = request.getParameter("mname");
                String lname = request.getParameter("lname");
                String username = request.getParameter("username");
                String password = request.getParameter("password");
                String email = request.getParameter("email");
                String contact = request.getParameter("contact");
                String role = "1";

                HttpSession session = request.getSession();
                boolean type = false;
                String head = "";
                String subhead = "";
                String title = "";
                String redirect = "";

                if(DBUserExtract.isUserExisting(username, email)){
                        type = false;
                        head = "Error encountered while adding new teacher";
                        subhead = "Redirecting to Add Teacher page.";
                        title = "ERROR";
                        redirect = "3;url=/MealPlanning/Admin/Add_Teacher";
                } else{
                        if(DBInsert.addTeacher(username, password, email, role, fname, mname, lname, contact)){
                                type = true;
                                head = "Success! Teacher has been added";
                                subhead = "Redirecting to Add Teacher page.";
                                title = "Teacher Added";
                                redirect = "3;url=/MealPlanning/Admin/Add_Teacher";
                        } else{
                                type = false;
                                head = "Error encountered while adding new teacher";
                                subhead = "Redirecting to Add Teacher page.";
                                title = "ERROR";
                                redirect = "3;url=/MealPlanning/Admin/Add_Teacher";
                        }
                }
                session.setAttribute("alert", new AlertMessage(type, head, subhead, title, redirect));
                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainPage/Aler
            dispatcher.forward(request,response);
        }

}


package servlet;

import java.io.IOException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
import dbutils.DBDataExtract;
import dbutils.DBInsert;
import dbutils.DBUserExtract;
import models.MealPlanModel;
import models.PatientModel;

/**
 * Servlet implementation class CreateMenu
 */
@WebServlet("/CreateMenu")
public class CreateMenu extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public CreateMenu() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(2)){
                        int patientId = 0;
                        try{
                                patientId = Integer.parseInt(request.getParameter("patient"));
                                //System.out.println("okay " + patientId);
                        } catch(Exception e){
                                e.printStackTrace();
                        }
                        session.setAttribute("patientId", patientId);
                        ArrayList<PatientModel> patients = DBUserExtract.getPatients();
                        for(PatientModel patient : patients){
                                if(patient.getId() == patientId){
                                        String patientName = patient.getLname() + ", " + patient.getFname() + " " +
                                        session.setAttribute("patientName", patientName);
                                }
                        }
                        //System.out.println("okay2 " + patientId);
                        MealPlanModel mpm = (MealPlanModel)session.getAttribute("mealPlan");
                        int week = 0;
                        if(request.getParameter("from").equals("addReport")){
                                week = DBDataExtract.getLastWeek(mpm.getPatientId());
                                mpm.setWeek(week);
                                DBInsert.addWeeklyReport(mpm);
                        } else{
                                //
                                week = DBDataExtract.getLastWeek(patientId) - 1;
                                mpm = DBDataExtract.getReport(patientId, week);
                                session.setAttribute("mealPlan", mpm);

                        }

                        int id = (Integer)session.getAttribute("patientId");
                        MealPlanModel mptemp = DBDataExtract.getReport(id, week);
                        mpm.setId(mptemp.getId());
                        session.setAttribute("mealPlan", mpm);
                        session.setAttribute("datestart", mpm.getStart().getTime());
                        session.setAttribute("dateend", mpm.getEnd().getTime());
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Stude
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }

        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                String bvegA = request.getParameter("bvegA");
                String bvegB = request.getParameter("bvegB");
                String bfruit = request.getParameter("bfruit");
                String bmilk = request.getParameter("bmilk");
                String bsugar = request.getParameter("bsugar");
                String brice = request.getParameter("brice");
                String bmeatL = request.getParameter("bmeatL");
                String bmeatM = request.getParameter("bmeatM");
                String bfat = request.getParameter("bfat");

                String lvegA = request.getParameter("lvegA");
```

```
String lvegB = request.getParameter("lvegB");
String lfruit = request.getParameter("lfruit");
String lmilk = request.getParameter("lmilk");
String lsugar = request.getParameter("lsugar");
String lrice = request.getParameter("lrice");
String lmeatL = request.getParameter("lmeatL");
String lmeatM = request.getParameter("lmeatM");
String lfat = request.getParameter("lfat");

String dvegA = request.getParameter("dvegA");
String dvegB = request.getParameter("dvegB");
String dfruit = request.getParameter("dfruit");
String dmilk = request.getParameter("dmilk");
String dsugar = request.getParameter("dsugar");
String drice = request.getParameter("drice");
String dmeatL = request.getParameter("dmeatL");
String dmeatM = request.getParameter("dmeatM");
String dfat = request.getParameter("dfat");

HttpSession session = request.getSession();

double ebvegA = 0;
double ebvegB = 0;
double ebfruit = 0;
double ebmilk = 0;
double ebsugar = 0;
double ebrice = 0;
double ebmeatL = 0;
double ebmeatM = 0;
double ebfat = 0;

double elvegA = 0;
double elvegB = 0;
double elfruit = 0;
double elmilk = 0;
double elsugar = 0;
double elrice = 0;
double elmeatL = 0;
double elmeatM = 0;
double elfat = 0;

double edvegA = 0;
double edvegB = 0;
double edfruit = 0;
double edmilk = 0;
double edsugar = 0;
double edrice = 0;
double edmeatL = 0;
double edmeatM = 0;
double edfat = 0;



try{
        ebvegA = Double.parseDouble(request.getParameter("ebvegA"));
        ebvegB = Double.parseDouble(request.getParameter("ebvegB"));
        ebfruit = Double.parseDouble(request.getParameter("ebfruit"));
        ebmilk = Double.parseDouble(request.getParameter("ebmilk"));
        ebsugar = Double.parseDouble(request.getParameter("ebsugar"));
        ebrice = Double.parseDouble(request.getParameter("ebrice"));
        ebmeatL = Double.parseDouble(request.getParameter("ebmeatL"));
        ebmeatM = Double.parseDouble(request.getParameter("ebmeatM"));
        ebfat = Double.parseDouble(request.getParameter("ebfat"));

        elvegA = Double.parseDouble(request.getParameter("elvegA"));
        elvegB = Double.parseDouble(request.getParameter("elvegB"));
        elfruit = Double.parseDouble(request.getParameter("elfruit"));
        elmilk = Double.parseDouble(request.getParameter("elmilk"));
        elsugar = Double.parseDouble(request.getParameter("elsugar"));
        elrice = Double.parseDouble(request.getParameter("elrice"));
        elmeatL = Double.parseDouble(request.getParameter("elmeatL"));
        elmeatM = Double.parseDouble(request.getParameter("elmeatM"));
        elfat = Double.parseDouble(request.getParameter("elfat"));
        edvegA = Double.parseDouble(request.getParameter("edvegA"));
        edvegB = Double.parseDouble(request.getParameter("edvegB"));
        edfruit = Double.parseDouble(request.getParameter("edfruit"));
        edmilk = Double.parseDouble(request.getParameter("edmilk"));
        edsugar = Double.parseDouble(request.getParameter("edsugar"));
        edrice = Double.parseDouble(request.getParameter("edrice"));
        edmeatL = Double.parseDouble(request.getParameter("edmeatL"));
        edmeatM = Double.parseDouble(request.getParameter("edmeatM"));
        edfat = Double.parseDouble(request.getParameter("edfat"));

}catch(Exception e){

}

String breakfast = "";
String lunch = "";
String dinner = "";

if(bvegA != ""){
```

```
        breakfast += bvegA + "||" + ebvegA + " vegA||";
}
if(bvegB != ""){
        breakfast += bvegB + "||" + ebvegB + " vegB||";
}
if(bfruit != ""){
        breakfast += bfruit + "||" + ebfruit + " fruit||";
}
if(bmilk != ""){
        breakfast += bmilk + "||" + ebmilk + " milk||";
}
if(bsugar != ""){
        breakfast += bsugar + "||" + ebsugar + " sugar||";
}
if(brice != ""){
        breakfast += brice + "||" + ebrice + " rice||";
}
if(bmeatL != ""){
        breakfast += bmeatL + "||" + ebmeatL + " meatL||";
}
if(bmeatM != ""){
        breakfast += bmeatM + "||" + ebmeatM + " meatM||";
}
if(bfat != ""){
        breakfast += bfat + "||" + ebfat + " fat||";
}

if(lvegA != ""){
        lunch += lvegA + "||" + elvegA + " vegA||";
}
if(lvegB != ""){
        lunch += lvegB + "||" + elvegB + " vegB||";
}
if(lfruit != ""){
        lunch += lfruit + "||" + elfruit + " fruit||";
}
if(lmilk != ""){
        lunch += lmilk + "||" + elmilk + " milk||";
}
if(lsugar != ""){
        lunch += lsugar + "||" + elsugar + " sugar||";
}
if(lrice != ""){
        lunch += lrice + "||" + elrice + " rice||";
}
if(lmeatL != ""){
        lunch += lmeatL + "||" + elmeatL + " meatL||";
}
if(lmeatM != ""){
        lunch += lmeatM + "||" + elmeatM + " meatM||";
}
if(lfat != ""){
        lunch += lfat + "||" + elfat + " fat||";
}

if(dvegA != ""){
        dinner += dvegA + "||" + edvegA + " vegA||";
}
if(dvegB != ""){
        dinner += dvegB + "||" + edvegB + " vegB||";
}
if(dfruit != ""){
        dinner += dfruit + "||" + edfruit + " fruit||";
}
if(dmilk != ""){
        dinner += dmilk + "||" + edmilk + " milk||";
}
if(dsugar != ""){
        dinner += dsugar + "||" + edsugar + " sugar||";
}
if(drice != ""){
        dinner += drice + "||" + edrice + " rice||";
}
if(dmeatL != ""){
        dinner += dmeatL + "||" + edmeatL + " meatL||";
}
if(dmeatM != ""){
        dinner += dmeatM + "||" + edmeatM + " meatM||";
}
if(dfat != ""){
        dinner += dfat + "||" + edfat + " fat||";
}

Date date = null;
DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
MealPlanModel mpm = (MealPlanModel)session.getAttribute("mealPlan");

date = (Date)session.getAttribute("fdate");
String day1 = "";
String day2 = "";

day1 = dateFormat.format(date);
```

```
                day2 = dateFormat.format(mpm.getEnd());
                DBInsert.addSampleMenu(mpm.getId(), date, breakfast, lunch, dinner);
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(2)){
                        //DBInsert.addWeeklyReport((MealPlanModel)session.getAttribute("mealPlan"));
                        if(day1.equals(day2)){
                                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/j
                            dispatcher.forward(request,response);
                        }else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Stude
                    dispatcher.forward(request,response);
                        }
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }
}

package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBUpdate;

/**
 * Servlet implementation class DeactivateUser
 */
@WebServlet("/DeactivateUser")
public class DeactivateUser extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public DeactivateUser() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOExcep
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && (session.getAttribute("role").equals(0) ||
                                session.getAttribute("role").equals(1) || session.getAttribute("role").equals(2))){
                        response.sendRedirect("/MealPlanning/Admin/View_Teachers");
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/EmailBlast/Login");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Messa
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOExce
                // TODO Auto-generated method stub
                try{
                        int userId = Integer.parseInt(request.getParameter("id"));
                        DBUpdate.deactivateUser(userId);
                        //request.getSession().setAttribute("alertOn", userId);
                } catch(Exception e){
                        e.printStackTrace();
                }
        }

}

package servlet;

import java.io.IOException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
```

```
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBUserExtract;
import models.MealPlanModel;
import models.PatientModel;
import operations.DietCalculation;

/**
 * Servlet implementation class DietCalculatorServlet
 */
@WebServlet("/DietCalculatorServlet")
public class DietCalculatorServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public DietCalculatorServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
		// TODO Auto-generated method stub
		HttpSession session = request.getSession();
		if(null != session.getAttribute("role") && session.getAttribute("role").equals(2)){
			int patientId = 0;
			try{
				patientId = Integer.parseInt(request.getParameter("patient"));
			} catch(Exception e){
				e.printStackTrace();
			}
			session.setAttribute("patientId", patientId);
			ArrayList<PatientModel> patients = DBUserExtract.getPatients();
			for(PatientModel patient : patients){
				if(patient.getId() == patientId){
					String patientName = patient.getLname() + ", " + patient.getFname() + " " +
					session.setAttribute("patientName", patientName);
				}
			}
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Stude
		    dispatcher.forward(request,response);
		} else if(session.getAttribute("role") == null){
			response.sendRedirect("/MealPlanning/Log-in");
		} else{
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
		    dispatcher.forward(request,response);
		}
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
		// TODO Auto-generated method stub
		String sHeight = request.getParameter("height");
		String sWeight = request.getParameter("weight");
		String sActivity = request.getParameter("activity");
		String sPdCHO = "65";
		String sPdPRO = "15";
		String sPdFAT = "20";
		double height = 0, weight = 0;
		int activity = 0, pdCHO = 0, pdPRO = 0, pdFAT  = 0;

		try{
			height = Double.parseDouble(sHeight);
			weight = Double.parseDouble(sWeight);
			activity = Integer.parseInt(sActivity);
			pdCHO = Integer.parseInt(sPdCHO);
			pdPRO = Integer.parseInt(sPdPRO);
			pdFAT = Integer.parseInt(sPdFAT);
		} catch(Exception e){
			e.printStackTrace();
		}
		Calendar cal = Calendar.getInstance();
		Date start;
		Date end;
		start = cal.getTime();
		cal.add(Calendar.DATE, 6);
		end = cal.getTime();

		DietCalculation dietcalc = new DietCalculation(height, weight, activity, pdCHO, pdPRO, pdFAT);
```

```
                HttpSession session = request.getSession();
                session.setAttribute("dietcalc", dietcalc);

                MealPlanModel mealPlan = new MealPlanModel();
                mealPlan.setPatientId((Integer)session.getAttribute("patientId"));
                mealPlan.setStart(start);
                mealPlan.setEnd(end);
                mealPlan.setHeight(height);
                mealPlan.setWeight(weight);
                mealPlan.setBmi(dietcalc.getBMI());
                mealPlan.setClassification(dietcalc.getClassification());
                mealPlan.setDbw(dietcalc.getDBW());
                mealPlan.setActivity(activity);
                mealPlan.setTea(dietcalc.getTEA());
                mealPlan.setgCHO(dietcalc.getWtCHO());
                mealPlan.setgPRO(dietcalc.getWtPRO());
                mealPlan.setgFAT(dietcalc.getWtFAT());
                mealPlan.setExVegA(dietcalc.getExVegA());
                mealPlan.setExVegB(dietcalc.getExVegB());
                mealPlan.setExFruit(dietcalc.getExFruit());
                mealPlan.setExMilk(dietcalc.getExMilk());
                mealPlan.setExSugar(dietcalc.getExSugar());
                mealPlan.setExRice(dietcalc.getExRice());
                mealPlan.setExMeatL(dietcalc.getExMeatL());
                mealPlan.setExMeatM(dietcalc.getExMeatM());
                mealPlan.setExFat(dietcalc.getExFat());

                session.setAttribute("mealPlan", mealPlan);

                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Student/dietP
            dispatcher.forward(request,response);

                }
}


package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBUpdate;
import dbutils.DBUserExtract;
import operations.AlertMessage;

/**
 * Servlet implementation class EditProfile
 */
@WebServlet("/EditProfile")
public class EditProfile extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public EditProfile() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role")){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                String username = request.getParameter("username");
                String password = request.getParameter("password");
                String email = request.getParameter("email");
```

```
                String contact = request.getParameter("contact");

                HttpSession session = request.getSession();
                boolean type = false;
                String head = "";
                String subhead = "";
                String title = "";
                String redirect = "";
                int id = (Integer) session.getAttribute("id");
                int role = (Integer)session.getAttribute("role");
                boolean change = true;

                if(DBUserExtract.isUserExisting(username, email)){
                        if(DBUserExtract.isEmailExisting(email)){
                                if(DBUserExtract.isEmailTheSame(id, email)){
                                        change = change && true;
                                } else{
                                        change = change && false;
                                }
                        }
                        if(change && DBUserExtract.isUsernameExisting(username)){
                                if(DBUserExtract.isUsernameTheSame(id, username)){
                                        change = change && true;
                                } else{
                                        change = change && false;
                                }
                        }
                } else{
                }


                if(change && DBUpdate.editProfile(id, username, contact, email, password)){
                        type = true;
                        switch(role){
                                case 1: subhead = "Redirecting to Add Student page.";
                                                redirect = "3;url=/MealPlanning/Teacher/Add_Student";
                                break;
                                case 2: subhead = "Redirecting to Add Patient page.";
                                                redirect = "3;url=/MealPlanning/Student/Add_Patient";
                                break;
                                case 3: subhead = "Redirecting to Input Intake page.";
                                                redirect = "3;url=/MealPlanning/Patient/Input_Intake";
                                break;
                        }
                        head = "Success! Profile has been updated";
                        title = "Profile Updated";
                        session.setAttribute("username", username);
                        session.setAttribute("email", email);
                        session.setAttribute("contact", contact);
                } else{
                        type = false;
                        head = "Error encountered while updating";
                        subhead = "Redirecting to Edit Profile page.";
                        redirect = "3;url=/MealPlanning/Edit_Profile";
                        title = "ERROR";
                }

                session.setAttribute("alert", new AlertMessage(type, head, subhead, title, redirect));
                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainPage/Aler
            dispatcher.forward(request,response);
        }

}

package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class FoodListServlet
 */
@WebServlet("/FoodListServlet")
public class FoodListServlet extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public FoodListServlet() {
        super();
        // TODO Auto-generated constructor stub
    }
```

```
        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(3)){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Patie
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                doGet(request, response);
        }

}


package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBUpdate;
import dbutils.DBUserExtract;
import operations.AlertMessage;
import operations.CodeGenerator;
import operations.SendEmail;



/**
 * Servlet implementation class Login
 */
@WebServlet("/ForgetPassword")
public class ForgetPassword extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ForgetPassword() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
            RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainPage/passwordl
            dispatcher.forward(request,response);
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                boolean type = false;
                String head = "";
                String subhead = "";
                String title = "";
                String redirect = "";

                String email = request.getParameter("email");
                if(DBUserExtract.isEmailExisting(email)){
                        int id = DBUserExtract.getIdUsingEmail(email);
                        CodeGenerator cg = new CodeGenerator(6);
                        String password = cg.getRandomCode();
                        if(DBUpdate.resetPassword(id, password) && SendEmail.emailPassword(email, password)){
                                type = true;
                                head = "Password has been reset <br> Your new password is sent to your email ";
```

```
                                subhead = "Redirecting to Log-in Page.";
                                title = "Success resetting password";
                                redirect = "3;url=/MealPlanning/Log-in";
                        } else{
                                type = false;
                                head = "Cannot reset password";
                                subhead = "Redirecting to Log-in Page.";
                                title = "ERROR";
                                redirect = "3;url=/MealPlanning/Log-in";
                        }
                }else{
                        type = false;
                        head = "Email does not match a user";
                        subhead = "Redirecting to Log-in Page.";
                        title = "ERROR";
                        redirect = "3;url=/MealPlanning/Log-in";
                }
                session.setAttribute("alert", new AlertMessage(type, head, subhead, title, redirect));
                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainPage/Aler
            dispatcher.forward(request,response);
        }

}


package servlet;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBDataExtract;
import dbutils.DBInsert;
import models.MenuModel;

/**
 * Servlet implementation class InputIntake
 */
@WebServlet("/InputIntake")
public class InputIntake extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public InputIntake() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(3)){
                        int id = (Integer)session.getAttribute("id");
                        Calendar cal= Calendar.getInstance();
                        Date date = cal.getTime();
                        if(session.getAttribute("foods") == null){
                                ArrayList<String> foods = new ArrayList<String>();
                                session.setAttribute("foods", foods);
                        }
                        MenuModel menu = DBDataExtract.getSampleMenu(id, date);
                        if(menu == null){
                                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/j
                                dispatcher.forward(request,response);
                        } else{
                                session.setAttribute("smenu", menu);
                                RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/j
                                dispatcher.forward(request,response);
                        }
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }

        }

        /**
```

```
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(3)){
                        ArrayList<String> foods = (ArrayList<String>)session.getAttribute("foods");
                        String intake = request.getParameter("intake");
                        String purpose = request.getParameter("purpose");
                        if(purpose.equals("add")){
                                foods.add(intake);
                        } else if(purpose.equals("remove")){
                                foods.remove(intake);
                        } else{
                                MenuModel menu = (MenuModel)session.getAttribute("smenu");
                                int menuId = menu.getId();
                                String foodItem = "";
                                for(String food : foods){
                                        foodItem += food + "||";
                                }
                                DBInsert.addIntake(menuId, foodItem);
                        }
                        session.setAttribute("foods", foods);
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Patie
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

}

package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;



/**
 * Servlet implementation class Login
 */
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession(false);
                if(session != null){
                        session.invalidate();
                }
            RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainPage/login.js
            dispatcher.forward(request,response);
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                doGet(request, response);
        }

}

package servlet;

import java.io.IOException;
```

```
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBUpdate;

/**
 * Servlet implementation class AddTeacher
 */
@WebServlet("/ReassignPatients")
public class ReassignPatients extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ReassignPatients() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
		// TODO Auto-generated method stub
		HttpSession session = request.getSession();
		if(null != session.getAttribute("role") && session.getAttribute("role").equals(1)){
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Teach
		    dispatcher.forward(request,response);
		} else if(session.getAttribute("role") == null){
			response.sendRedirect("/MealPlanning/Log-in");
		} else{
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
		    dispatcher.forward(request,response);
		}
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
		// TODO Auto-generated method stub
		HttpSession session = request.getSession();
		try{
			int teacherTo = Integer.parseInt(request.getParameter("data"));
			int id = Integer.parseInt(request.getParameter("id"));
			DBUpdate.reassignPatient(id, teacherTo);
			//request.getSession().setAttribute("alertOn", userId);
		} catch(Exception e){
			e.printStackTrace();
		}
	}

}


package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBUpdate;

/**
 * Servlet implementation class AddTeacher
 */
@WebServlet("/ReassignStudents")
public class ReassignStudents extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ReassignStudents() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
```

```
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(1)){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Teach
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                try{
                        int teacherTo = Integer.parseInt(request.getParameter("data"));
                        int id = Integer.parseInt(request.getParameter("id"));
                        DBUpdate.reassignStudent(id, teacherTo);
                        //request.getSession().setAttribute("alertOn", userId);
                } catch(Exception e){
                        e.printStackTrace();
                }
        }

}


package servlet;

import java.io.IOException;
import java.util.Calendar;
import java.util.Date;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBDataExtract;
import dbutils.DBInsert;
import dbutils.DBUserExtract;
import models.UserModel;
import operations.AlertMessage;
import operations.CodeGenerator;

/**
 * Servlet implementation class UserVerification
 */
@WebServlet("/UserVerification")
public class UserVerification extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public UserVerification() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Messa
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                String username = request.getParameter("username");
                String password = request.getParameter("password");
```

```
                int id = DBUserExtract.verifyUser(username, password);
                if(id != 0){
                        UserModel user = DBUserExtract.getUserDetails(id);
                        int role = user.getRole();
                        String email = user.getEmail();
                        String contact = user.getContact();
                        HttpSession session = request.getSession(true);
                        session.setAttribute("id", id);
                        session.setAttribute("role", role);
                        session.setAttribute("username", username);
                        session.setAttribute("email", email);
                        session.setAttribute("contact", contact);
                        String nextPage = "/MealPlanning";
                        switch(role){
                                case 0:
                                        nextPage += "/Admin/Add_Teacher";
                                        break;
                                case 1:
                                        Calendar cal = Calendar.getInstance();
                                        Date date = cal.getTime();
                                        if(DBDataExtract.getCode(date) == "ERROR"){
                                                CodeGenerator cg = new CodeGenerator(6);
                                                DBInsert.addDailyCode(date, cg.getRandomCode());
                                        }
                                        nextPage += "/Teacher/Add_Student";
                                        break;
                                case 2:
                                        nextPage += "/Student/Add_Patient";
                                        break;
                                case 3:
                                        nextPage += "/Patient/Input_Intake";
                                        break;
                        }
                        response.sendRedirect(nextPage);
                } else{
                        HttpSession session = request.getSession();
                        boolean type = false;
                        String head = "Account Not Found/Account Deactivated";
                        String subhead = "Redirecting to login page.";
                        String title = "ERROR";
                        String redirect = "3;url=/MealPlanning/Log-in";
                        session.setAttribute("alert", new AlertMessage(type, head, subhead, title, redirect));
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                        // return to login
                        // user does not exist
                }
        }

}


package servlet;

import java.io.IOException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import dbutils.DBDataExtract;
import dbutils.DBUpdate;
import dbutils.DBUserExtract;
import models.PatientModel;

/**
 * Servlet implementation class ViewMenu
 */
@WebServlet("/ViewIntake")
public class ViewIntake extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ViewIntake() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
```

```
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(2)){
                        int patientId = 0;
                        try{
                                patientId = Integer.parseInt(request.getParameter("patient"));
                        }catch(Exception e){

                        }
                        ArrayList<PatientModel> patients = DBUserExtract.getPatients();
                        for(PatientModel patient : patients){
                                if(patient.getId() == patientId){
                                        String patientName = patient.getLname() + ", " + patient.getFname() + " " +
                                        session.setAttribute("patientName", patientName);
                                }
                        }
                        ArrayList<Date> dates = DBDataExtract.getAllDates(patientId);
                        session.setAttribute("dates", dates);
                        session.setAttribute("curr", dates.get(0));
                        session.setAttribute("patient", patientId);
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Stude
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }

        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(request.getParameter("purpose").equals("ct")){
                        try{
                                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                                Date curr = dateFormat.parse(request.getParameter("week"));
                                session.setAttribute("curr", curr);
                        }catch(Exception e){

                        }
                }else if(request.getParameter("purpose").equals("gf")){
                        String feedback = request.getParameter("textarea1");
                        String stea = request.getParameter("tea");
                        String sCHO = request.getParameter("gCHO");
                        String sPRO = request.getParameter("gPRO");
                        String sFAT = request.getParameter("gFAT");
                        try{
                                int tea = Integer.parseInt(stea);
                                int gCHO = Integer.parseInt(sCHO);
                                int gPRO = Integer.parseInt(sPRO);
                                int gFAT = Integer.parseInt(sFAT);
                                int id = (Integer)session.getAttribute("intakeId");
                                DBUpdate.giveFeedback(id, feedback, gCHO, gPRO, gFAT, tea);
                        }catch(Exception e){

                        }
                }
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(2)){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Stude
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

}


package servlet;

import java.io.IOException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
import dbutils.DBDataExtract;
import dbutils.DBUpdate;
import dbutils.DBUserExtract;
import models.PatientModel;

/**
 * Servlet implementation class ViewMenu
 */
@WebServlet("/ViewMealPlan")
public class ViewMealPlan extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ViewMealPlan() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
		// TODO Auto-generated method stub
		HttpSession session = request.getSession();
		if(null != session.getAttribute("role") && session.getAttribute("role").equals(3)){
			int patientId = (Integer)session.getAttribute("id");
			ArrayList<PatientModel> patients = DBUserExtract.getPatients();
			for(PatientModel patient : patients){
				if(patient.getId() == patientId){
					String patientName = patient.getLname() + ", " + patient.getFname() + " " +
					session.setAttribute("patientName", patientName);
				}
			}
			ArrayList<Date> dates = DBDataExtract.getAllDates(patientId);
			if(dates.size() == 0){
				RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/j
			    dispatcher.forward(request,response);
			} else{
			session.setAttribute("dates", dates);
			session.setAttribute("curr", dates.get(0));
			session.setAttribute("patient", patientId);

			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Patie
		    dispatcher.forward(request,response);
			}
		} else if(session.getAttribute("role") == null){
			response.sendRedirect("/MealPlanning/Log-in");
		} else{
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
		    dispatcher.forward(request,response);
		}


	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
		// TODO Auto-generated method stub
		HttpSession session = request.getSession();
		if(request.getParameter("purpose").equals("ct")){
			try{
				DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
				Date curr = dateFormat.parse(request.getParameter("week"));
				session.setAttribute("curr", curr);
			}catch(Exception e){

			}
		}
		if(null != session.getAttribute("role") && session.getAttribute("role").equals(3)){
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Patie
		    dispatcher.forward(request,response);
		} else if(session.getAttribute("role") == null){
			response.sendRedirect("/MealPlanning/Log-in");
		} else{
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
		    dispatcher.forward(request,response);
		}
	}

}

package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class ViewMyPatients
 */
@WebServlet("/ViewMyPatients")
public class ViewMyPatients extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ViewMyPatients() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
		// TODO Auto-generated method stub
		HttpSession session = request.getSession();
		if(null != session.getAttribute("role") && session.getAttribute("role").equals(2)){
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Stude
		    dispatcher.forward(request,response);
		} else if(session.getAttribute("role") == null){
			response.sendRedirect("/MealPlanning/Log-in");
		} else{
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
		    dispatcher.forward(request,response);
		}

	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
		// TODO Auto-generated method stub
		doGet(request, response);
	}

}

package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class AddTeacher
 */
@WebServlet("/ViewPatients")
public class ViewPatients extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ViewPatients() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
		// TODO Auto-generated method stub
		HttpSession session = request.getSession();
		if(null != session.getAttribute("role") && session.getAttribute("role").equals(0)){
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Admin
		    dispatcher.forward(request,response);
		} else if(session.getAttribute("role") == null){
			response.sendRedirect("/MealPlanning/Log-in");
		} else{
			RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
		    dispatcher.forward(request,response);
		}
```

```
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                doGet(request, response);
        }

}

package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class AddTeacher
 */
@WebServlet("/ViewStudents")
public class ViewStudents extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ViewStudents() {
        super();
        // TODO Auto-generated constructor stub
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(0)){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Admin
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                doGet(request, response);
        }

}

package servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class AddTeacher
 */
@WebServlet("/ViewTeachers")
public class ViewTeachers extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ViewTeachers() {
        super();
        // TODO Auto-generated constructor stub
```

```
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOE
                // TODO Auto-generated method stub
                HttpSession session = request.getSession();
                if(null != session.getAttribute("role") && session.getAttribute("role").equals(0)){
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/Admin
                    dispatcher.forward(request,response);
                } else if(session.getAttribute("role") == null){
                        response.sendRedirect("/MealPlanning/Log-in");
                } else{
                        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/WEB-INF/jsp/MainP
                    dispatcher.forward(request,response);
                }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IO
                // TODO Auto-generated method stub
                doGet(request, response);
        }

}

jdbc.host = jdbc:mysql://localhost:3306/mpdb
jdbc.driver = com.mysql.jdbc.Driver
jdbc.username = root
jdbc.password = root

upload.directory = files/


$(document).ready(function() {
        $('#compute').click(function(){
                var height = $('input[name=height]').val();
                var activity = $('input[name=activity]:checked').val();
                var pdCHO = $('input[name=pdCHO]').val();
                var pdPRO = $('input[name=pdPRO]').val();
                var pdFAT = $('input[name=pdFAT]').val();
                if(height > 0 && pdCHO > 0 && pdPRO > 0 && pdFAT > 0)
                        $.ajax({
                                type : 'POST',
                            url : '/MealPlanning/DietCalculator',
                            data : {'height' : height, 'activity' : activity, 'pdCHO' : pdCHO, 'pdPRO' : pdPRO, 'pdl
                            success:function(html){
                                $('#initResult').html(html);
                            }
                        });
        });

        $('.removebtn').on('click', function() {
                var data = $(this).data();
            $('.deactivate').data('recordId', data.record);
            $('.sMessage').text(data.message);
        });

        $('#confirm-delete').on('click', function() {
            var id = $('.deactivate').data("recordId");
            $.ajax({
                        type:'POST',
                url:'/MealPlanning/Admin/Deactivate_User',
                data:'id='+id,
                success:function(data){
                    location.reload();
                }
            });
        });

        $('input.zx').change(function() {
                var daycode = $("#daycode").val();
                var codeinput = $("#codeinput").val();
                if(daycode == codeinput){
                        $('#cmenu').attr("disabled", false);
                } else{
                        $('#cmenu').attr("disabled", true);
                }
        });

        $('input.qw').change(function() {
                var ebvegA = parseFloat($("#ebvegA").val());
                var ebvegB = parseFloat($("#ebvegB").val());
                var ebfruit = parseFloat($("#ebfruit").val());
                var ebmilk = parseFloat($("#ebmilk").val());
                var ebsugar = parseFloat($("#ebsugar").val());
                var ebrice = parseFloat($("#ebrice").val());
                var ebmeatL = parseFloat($("#ebmeatL").val());
                var ebmeatM = parseFloat($("#ebmeatM").val());
```

```
        var ebfat = parseFloat($("#ebfat").val());

        var elvegA = parseFloat($("#elvegA").val());
        var elvegB = parseFloat($("#elvegB").val());
        var elfruit = parseFloat($("#elfruit").val());
        var elmilk = parseFloat($("#elmilk").val());
        var elsugar = parseFloat($("#elsugar").val());
        var elrice = parseFloat($("#elrice").val());
        var elmeatL = parseFloat($("#elmeatL").val());
        var elmeatM = parseFloat($("#elmeatM").val());
        var elfat = parseFloat($("#elfat").val());

        var edvegA = parseFloat($("#edvegA").val());
        var edvegB = parseFloat($("#edvegB").val());
        var edfruit = parseFloat($("#edfruit").val());
        var edmilk = parseFloat($("#edmilk").val());
        var edsugar = parseFloat($("#edsugar").val());
        var edrice = parseFloat($("#edrice").val());
        var edmeatL = parseFloat($("#edmeatL").val());
        var edmeatM = parseFloat($("#edmeatM").val());
        var edfat = parseFloat($("#edfat").val());

        var etvegA = parseFloat($("#etvegA").val());
        var etvegB = parseFloat($("#etvegB").val());
        var etfruit = parseFloat($("#etfruit").val());
        var etmilk = parseFloat($("#etmilk").val());
        var etsugar = parseFloat($("#etsugar").val());
        var etrice = parseFloat($("#etrice").val());
        var etmeatL = parseFloat($("#etmeatL").val());
        var etmeatM = parseFloat($("#etmeatM").val());
        var etfat = parseFloat($("#etfat").val());

        var vegA = ebvegA + elvegA + edvegA;
        var vegB = ebvegB + elvegB + edvegB;
        var fruit = ebfruit + elfruit + edfruit;
        var milk = ebmilk + elmilk + edmilk;
        var sugar = ebsugar + elsugar + edsugar;
        var rice = ebrice + elrice + edrice;
        var meatL = ebmeatL + elmeatL + edmeatL;
        var meatM = ebmeatM + elmeatM + edmeatM;
        var fat = ebfat + elfat + edfat;

        var daycode = $("#daycode").val();
        var codeinput = $("#codeinput").val();

        if(vegA == etvegA && vegB == etvegB &&
                        fruit == etfruit && milk == etmilk &&
                        sugar == etsugar && rice == etrice &&
                        meatL == etmeatL && meatM == etmeatM &&
                        fat == etfat && daycode == codeinput){
        //alert(meatL + " == " + etmeatL);
        //if(meatL == etmeatL){
                $('#cmenu').attr("disabled", false);
        }else{
                $('#cmenu').attr("disabled", true);
        }
});

$('#confirm-intake').on('click', function() {
    var id = $('.intakebtn').data("purpose");
    $.ajax({
                type:'POST',
        url:'/MealPlanning/Patient/Input_Intake',
        data:'purpose='+id,
        success:function(data){
            location.reload();
        }
    });
});

$('.sReassign').on('click', function() {
    var btn = $(this).data("btn");
    var sub = "sel" + btn.substring(3);
    var sel = $("#"+sub);
    var data = sel.val();
    $.ajax({
                type:'POST',
        url:'/MealPlanning/Teacher/Reassign_Students',
        data : {'data' : data, 'id' : btn.substring(3)},
        success:function(data){
            location.reload();
        }
    });
})

$('.pReassign').on('click', function() {
    var btn = $(this).data("btn");
    var sub = "sel" + btn.substring(3);
    var sel = $("#"+sub);
    var data = sel.val();
    $.ajax({
                type:'POST',
```

```
                url:'/MealPlanning/Teacher/Reassign_Patients',
                data : {'data' : data, 'id' : btn.substring(3)},
                success:function(data){
                    location.reload();
                }
            });
        })

        $('#teacherList').DataTable( {
        columnDefs: [
            {
                targets: [ 0, 1, 2 ],
                className: 'mdl-data-table__cell--non-numeric'
            }
        ]
    } );
});

$('.modal').modal();


$('#textarea1').trigger('autoresize');

$('.collapsible').collapsible();

$('select').material_select();

$("#teacher-form").validate({
    rules: {
        username: {
            minlength: 6,
            maxlength: 16
        },
        password: {
                        minlength: 6
                },
                passwordConfirm: {
                        minlength: 6,
                        equalTo: "#password"
                }
    },
    //For custom messages
    messages: {
    },
    errorElement : 'div',
    errorPlacement: function(error, element) {
      var placement = $(element).data('error');
      if (placement) {
        $(placement).append(error)
      } else {
        error.insertAfter(element);
      }
    }
 });

$('.datepicker').pickadate({
    selectMonths: true, // Creates a dropdown to control month
    selectYears: 50 // Creates a dropdown of 15 years to control year
  });


jdbc.host = jdbc:mysql://localhost:3306/mpdb
jdbc.driver = com.mysql.jdbc.Driver
jdbc.username = root
jdbc.password = root

upload.directory = files/


<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schema
  <display-name>MealPlanning</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <display-name>Login Servlet</display-name>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>servlet.LoginServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/Log-in</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Forget Password</display-name>
    <servlet-name>ForgetPassword</servlet-name>
    <servlet-class>servlet.ForgetPassword</servlet-class>
```

```
</servlet>
<servlet-mapping>
  <servlet-name>ForgetPassword</servlet-name>
  <url-pattern>/Forget_Password</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Edit Profile</display-name>
  <servlet-name>EditProfile</servlet-name>
  <servlet-class>servlet.EditProfile</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>EditProfile</servlet-name>
  <url-pattern>/Edit_Profile</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Add Student</display-name>
  <servlet-name>AddStudent</servlet-name>
  <servlet-class>servlet.AddStudent</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>AddStudent</servlet-name>
  <url-pattern>/Teacher/Add_Student</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Add Food Item</display-name>
  <servlet-name>AddFoodItem</servlet-name>
  <servlet-class>servlet.AddFoodItem</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>AddFoodItem</servlet-name>
  <url-pattern>/Teacher/Add_Food_Item</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Reassign Students</display-name>
  <servlet-name>ReassignStudents</servlet-name>
  <servlet-class>servlet.ReassignStudents</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ReassignStudents</servlet-name>
  <url-pattern>/Teacher/Reassign_Students</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Reassign Patients</display-name>
  <servlet-name>ReassignPatients</servlet-name>
  <servlet-class>servlet.ReassignPatients</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ReassignPatients</servlet-name>
  <url-pattern>/Teacher/Reassign_Patients</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Add Patient</display-name>
  <servlet-name>AddPatient</servlet-name>
  <servlet-class>servlet.AddPatient</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>AddPatient</servlet-name>
  <url-pattern>/Student/Add_Patient</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>View My Patients</display-name>
  <servlet-name>ViewMyPatients</servlet-name>
  <servlet-class>servlet.ViewMyPatients</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ViewMyPatients</servlet-name>
  <url-pattern>/Student/View_My_Patients</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Calculate Diet</display-name>
  <servlet-name>CalculateDiet</servlet-name>
  <servlet-class>servlet.DietCalculatorServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>CalculateDiet</servlet-name>
  <url-pattern>/Student/Calculate_Diet</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Create Menu</display-name>
  <servlet-name>CreateMenu</servlet-name>
  <servlet-class>servlet.CreateMenu</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>CreateMenu</servlet-name>
  <url-pattern>/Student/Create_Menu</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>Add Teacher</display-name>
  <servlet-name>AddTeacher</servlet-name>
  <servlet-class>servlet.AddTeacher</servlet-class>
</servlet>
<servlet-mapping>
```

```
    <servlet-name>AddTeacher</servlet-name>
    <url-pattern>/Admin/Add_Teacher</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View Teachers</display-name>
    <servlet-name>ViewTeachers</servlet-name>
    <servlet-class>servlet.ViewTeachers</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewTeachers</servlet-name>
    <url-pattern>/Admin/View_Teachers</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>DeactivateUser</servlet-name>
    <url-pattern>/Admin/Deactivate_User</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Deactivate User</display-name>
    <servlet-name>DeactivateUser</servlet-name>
    <servlet-class>servlet.DeactivateUser</servlet-class>
  </servlet>
  <servlet>
    <display-name>View Students</display-name>
    <servlet-name>ViewStudents</servlet-name>
    <servlet-class>servlet.ViewStudents</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewStudents</servlet-name>
    <url-pattern>/Admin/View_Students</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View Patients</display-name>
    <servlet-name>ViewPatients</servlet-name>
    <servlet-class>servlet.ViewPatients</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewPatients</servlet-name>
    <url-pattern>/Admin/View_Patients</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Input Intake</display-name>
    <servlet-name>InputIntake</servlet-name>
    <servlet-class>servlet.InputIntake</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>InputIntake</servlet-name>
    <url-pattern>/Patient/Input_Intake</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Food List</display-name>
    <servlet-name>FoodList</servlet-name>
    <servlet-class>servlet.FoodListServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>FoodList</servlet-name>
    <url-pattern>/Patient/Food_List</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View Meal Plan</display-name>
    <servlet-name>ViewMealPlan</servlet-name>
    <servlet-class>servlet.ViewMealPlan</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewMealPlan</servlet-name>
    <url-pattern>/Patient/View_Meal_Plan</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View Intake</display-name>
    <servlet-name>ViewIntake</servlet-name>
    <servlet-class>servlet.ViewIntake</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewIntake</servlet-name>
    <url-pattern>/Student/View_Intake</url-pattern>
  </servlet-mapping>
</web-app>


<%@ page import='models.*'%>
<%@ page import='dbutils.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Patient List/Admin</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed grey darken-2">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/black-gradient.jpg"  height="200px" width="300p
                </div>
                <a class="transparent white-text" style="font-size:25px" data-position="bottom" data-delay="50" dat
```

```
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <br>
                        <div class="col offset-s10 s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                 href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Admin/Add_Teacher"><i class="material-icons">add</i>Add Teacher</a></li>
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Admin/View_Teachers"><i class="material-icons">list</i>Teacher List</a></li>
        <li><a href="/MealPlanning/Admin/View_Students"><i class="material-icons">list</i>Student List</a></li>
        <li class="active"><a href="/MealPlanning/Admin/View_Patients"><i class="material-icons">list</i>Patient Li
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

        <!-- Modal Structure -->
        <div id="removeModal" class="modal">
        <div class="modal-content">
                <h4>Confirm Delete</h4>
                <p>Are you sure you want to remove <strong class="sMessage"></strong></p>
        </div>
        <div class="modal-footer">
                <a href="#!" class="modal-action modal-close waves-effect waves-green btn-flat">Cancel</a>
                <a id="confirm-delete" class="waves-effect waves-green btn-flat deactivate">Confirm</a>
        </div>
        </div>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="black darken-2 white-text">    Patient List</h4>
                                <br />
                                <div>
                                        <table class="mdl-data-table" id="teacherList">
                                                <thead>
                                                        <tr>
                                                                <th>Username</th>
                                                                <th>Name</th>
                                                                <th>Email</th>
                                                                <th>Contact Number</th>
                                                                <th>Option</th>
                                                        </tr>
                                                </thead>
                                                <tbody>
                                                        <%
                                                                ArrayList<PatientModel> patients = DBUserExtract.ge
                                                                for (PatientModel patient : patients) {
                                                                        if(!patient.isDeactivated()) {
                                                        %>
                                                        <tr>
                                                                <td><%=patient.getUsername()%></td>
                                                                <td><%=patient.getFname() + " " + patient.getMname(
                                                                <td><%=patient.getEmail()%></td>
                                                                <td><%=patient.getContact()%></td>

                                                                <td class="check">
                                                                        <button class="waves-effsect waves-light bt
                                                                        data-record="<%=patient.getId()%>" data-mes
                                                                        Deactivate</button>
                                                                </td>
                                                        </tr>
                                                        <%
                                                                        }
                                                                }
                                                        %>
                                                </tbody>

                                        </table>
                                </div>
                        </div>
                </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='models.*'%>
<%@ page import='dbutils.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Student List/Admin</title>
</head>
<body>
```

```
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed grey darken-2">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/black-gradient.jpg"  height="200px" width="300p
                </div>
                <a class="transparent white-text" style="font-size:25px" data-position="bottom" data-delay="50" dat
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <br>
                        <div class="col offset-s10 s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                 href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Admin/Add_Teacher"><i class="material-icons">add</i>Add Teacher</a></li>
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Admin/View_Teachers"><i class="material-icons">list</i>Teacher List</a></li>
        <li class="active"><a href="/MealPlanning/Admin/View_Students"><i class="material-icons">list</i>Student Li
        <li><a href="/MealPlanning/Admin/View_Patients"><i class="material-icons">list</i>Patient List</a></li>
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

        <!-- Modal Structure -->
        <div id="removeModal" class="modal">
        <div class="modal-content">
                <h4>Confirm Delete</h4>
                <p>Are you sure you want to remove <strong class="sMessage"></strong></p>
        </div>
        <div class="modal-footer">
                <a href="#!" class="modal-action modal-close waves-effect waves-green btn-flat">Cancel</a>
                <a id="confirm-delete" class="waves-effect waves-green btn-flat deactivate">Confirm</a>
        </div>
        </div>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="black darken-2 white-text">    Student List</h4>
                                <br />
                                <div>
                                        <table class="mdl-data-table" id="teacherList">
                                                <thead>
                                                        <tr>
                                                                <th>Username</th>
                                                                <th>Name</th>
                                                                <th>Email</th>
                                                                <th>Contact Number</th>
                                                                <th>Option</th>
                                                        </tr>
                                                </thead>
                                                <tbody>
                                                        <%
                                                                ArrayList<StudentModel> students = DBUserExtract.ge
                                                                for (StudentModel student : students) {
                                                                        if(!student.isDeactivated()) {
                                                        %>
                                                        <tr>
                                                                <td><%=student.getUsername()%></td>
                                                                <td><%=student.getFname() + " " + student.getMname(
                                                                <td><%=student.getEmail()%></td>
                                                                <td><%=student.getContact()%></td>

                                                                <td class="check">
                                                                        <button class="waves-effsect waves-light bt
                                                                        data-record="<%=student.getId()%>" data-mes
                                                                        Deactivate</button>
                                                                </td>
                                                        </tr>
                                                        <%
                                                                        }
                                                                }
                                                        %>
                                                </tbody>

                                        </table>
                                </div>
                    </div>
            </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>

<%@ page import='models.*'%>
```

```
<%@ page import='dbutils.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Teacher List/Admin</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed grey darken-2">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/black-gradient.jpg"  height="200px" width="300p
                </div>
                <a class="transparent white-text" style="font-size:25px" data-position="bottom" data-delay="50" dat
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <br>
                        <div class="col offset-s10 s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                 href="/MealPlanning/Log-in">
                                 <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Admin/Add_Teacher"><i class="material-icons">add</i>Add Teacher</a></li>
        <li><div class="divider"></div></li>
        <li class="active"><a href="/MealPlanning/Admin/View_Teachers"><i class="material-icons">list</i>Teacher Li
        <li><a href="/MealPlanning/Admin/View_Students"><i class="material-icons">list</i>Student List</a></li>
        <li><a href="/MealPlanning/Admin/View_Patients"><i class="material-icons">list</i>Patient List</a></li>
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

        <!-- Modal Trigger -->


  <!-- Modal Structure -->
        <div id="removeModal" class="modal">
        <div class="modal-content">
                <h4>Confirm Delete</h4>
                <p>Are you sure you want to remove <strong class="sMessage"></strong></p>
        </div>
        <div class="modal-footer">
                <a href="#!" class="modal-action modal-close waves-effect waves-green btn-flat">Cancel</a>
                <a id="confirm-delete" class="waves-effect waves-green btn-flat deactivate">Confirm</a>
        </div>
        </div>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="black darken-2 white-text">    Teacher List</h4>
                                <br />
                                <div>
                                        <table class="mdl-data-table" id="teacherList">
                                                <thead>
                                                        <tr>
                                                                <th>Username</th>
                                                                <th>Name</th>
                                                                <th>Email</th>
                                                                <th>Contact Number</th>
                                                                <th>Option</th>
                                                        </tr>
                                                </thead>
                                                <tbody>
                                                        <%
                                                                ArrayList<TeacherModel> teachers = DBUserExtract.ge
                                                                for (TeacherModel teacher : teachers) {
                                                                        if(!teacher.isDeactivated()){
                                                        %>
                                                        <tr>
                                                                <td><%=teacher.getUsername()%></td>
                                                                <td><%=teacher.getFname() + " " + teacher.getMname(
                                                                <td><%=teacher.getEmail()%></td>
                                                                <td><%=teacher.getContact()%></td>

                                                                <td class="check">
                                                                        <button class="waves-effsect waves-light bt
                                                                        data-record="<%=teacher.getId()%>" data-mes
                                                                        Deactivate</button>
                                                                </td>
                                                        </tr>
                                                        <%
                                                                        }
                                                                }
                                                        %>
```

```
                                                </tbody>

                                            </table>
                                        </div>
                                    </div>
                                </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Add Teacher/Admin</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed grey darken-2">
        <li><div class="userView">
                <div class="background">
                    <img src="/MealPlanning/assets/imgs/default/black-gradient.jpg"  height="200px" width="300p
                </div>
                <a class="transparent white-text" style="font-size:25px" data-position="bottom" data-delay="50" dat
                    <%= session.getAttribute("username") %></a>
                <div class="row">
                    <div class="col s10">
                        <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                    </div>
                    <br>
                    <div class="col offset-s10 s1">
                        <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                         href="/MealPlanning/Log-in">
                        <i class="material-icons">input</i></a>
                    </div>
                </div>
        </div></li>
        <li class="active"><a href=""/MealPlanning/Admin/Add_Teacher""><i class="material-icons">add</i>Add Teacher
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Admin/View_Teachers"><i class="material-icons">list</i>Teacher List</a></li>
        <li><a href="/MealPlanning/Admin/View_Students"><i class="material-icons">list</i>Student List</a></li>
        <li><a href="/MealPlanning/Admin/View_Patients"><i class="material-icons">list</i>Patient List</a></li>
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="black darken-2 white-text">    Add Teacher</h4>
                                <br />
                                <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Adm
                                            <div class="row">
                                                <div class="input-field col s4">
                                                    <input placeholder="First name" name="fname" id="fna
                                                    class="validate" required>
                                                    <label for="fname">First Name</label>
                                                </div>
                                                <div class="input-field col s4">
                                                    <input placeholder="Middle name" name="mname" id="m
                                                    class="validate" required>
                                                    <label for="mname">Middle Name</label>
                                                </div>
                                                <div class="input-field col s4">
                                                    <input placeholder="Last name" name="lname" id="lnam
                                                    class="validate" required>
                                                    <label for="lname">Last Name</label>
                                                </div>
                                            </div>
                                            <div class="row">
                                                <div class="input-field col s6">
                                                    <i class="material-icons prefix">email</i>
                                                    <input placeholder="Email" name="email" id="email"
                                                    class="validate" required>
                                                    <label for="email">Email</label>
                                                </div>
                                                <div class="input-field col s6">
                                                    <i class="material-icons prefix">phone</i>
                                                    <input placeholder="xxxxxxxxxx" name="contact" id=
                                                    class="validate" pattern="^\d{11}$" required>
                                                    <label for="contact">Contact Number</label>
                                                </div>
                                            </div>
                                            <div class="row">
                                                <div class="input-field col s6">
                                                    <i class="material-icons prefix">account_circle</i>
                                                    <input placeholder="Username" name="username" id="u
                                                    class="validate" required>
                                                    <label for="username">Username</label>
                                                </div>
                                            </div>
                                            <div class="row">
```

```
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">lock_open</i>
                                                                <input name="password" id="password" type="password
                                                                <label for="password">Password</label>
                                                        </div>
                                                        <div class="input-field col s6">
                                                                <input name="passwordConfirm" id="passwordConfirm"
                                                                 data-match-error="Passwords don't match" required>
                                                                <label for="passwordConfirm">Confirm Password</labe
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="col 6 ">
                                                                <button class="btn waves-effect waves-light" type="
                                                                <i class="material-icons right">send</i>
                                                                </button>
                                                        </div>
                                                        <div class="col 6 ">
                                                                <button class="red btn waves-effect waves-light" ty
                                                                <i class="material-icons right">replay</i>
                                                                </button>
                                                        </div>
                                                </div>
                                        </form>
                        </div>

                </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='operations.*'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<% AlertMessage message = (AlertMessage)session.getAttribute("alert"); %>
<meta http-equiv="Refresh" content="<%=message.getRedirect()%>">
<title><%= message.getTitle() %></title>
</head>
<body>
        <div class=container>
                <div class="row">
                <br /><br /><br />
                        <div class="col s6 offset-s3 z-depth-3">
                                <div class="row">
                                        <div class="col s12">
                                                <% if(message.isType()){ %>
                                                <h3 class="green white-text">Success</h3>
                                                <% } else{%>
                                                <h3 class="red white-text">Error</h3>
                                                <% } %>
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="col s9">
                                                <h5><%= message.getHead() %></h5>
                                                <%= message.getSubhead() %>
                                        </div>
                                        <div class="col s3 valign-wrapper">
                                                <% if(message.isType()){ %>
                                                <i class="medium material-icons green-text">done</i>
                                                <% } else{ %>
                                                <i class="medium material-icons red-text">error</i>
                                                <% } %>
                                        </div>
                                </div>
                        </div>
                </div>
        </div>
<% session.removeAttribute("alert"); %>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='java.util.Calendar'%>
<%@ page import='java.util.Date'%>
<%@ page import='dbutils.*'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Edit Profile</title>
</head>
<body>

        <%int role = (Integer)session.getAttribute("role");
                switch(role){
                        case 0:
        %>
        <%              break;
                        case 1: %>

        <!-- Teacher Side -->
        <ul id="slide-out" class="side-nav fixed green lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/green-gradient.jpg" height="200px" width="300px
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
```

```
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Teacher/Add_Students"><i class="material-icons">add</i>Add Student</a></li>
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Teacher/Reassign_Students"><i class="material-icons">list</i>Reassign Student </
        <li><a href="/MealPlanning/Teacher/Reassign_Patients"><i class="material-icons">assignment_ind</i>Reassign
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Teacher/Add_Food_Item"><i class="material-icons">add</i>Add New Food Item</a></l
        <li><div class="divider"></div></li>
        <%
                Calendar cal = Calendar.getInstance();
                Date date = cal.getTime();
        %>
        <li>Today's Code<%=DBDataExtract.getCode(date) %></li>
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

<!-- Teacher Page Layout -->
        <div class="row">
                <div class="col s8 offset-s3 z-depth-3">
                        <h4 class="green darken-2 white-text">    Edit Profile</h4>
                        <br />
                        <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Edi
                                <div class="row">
                                        <div class="input-field col s6">
                                                <i class="material-icons prefix">email</i>
                                                <input placeholder="Email" name="email" id="email"
                                                class="validate" required  value="<%=session.getAtt
                                                <label for="email">Email</label>
                                        </div>
                                        <div class="input-field col s6">
                                                <i class="material-icons prefix">phone</i>
                                                <input placeholder="xxxxxxxxxx" name="contact" id=
                                                class="validate" pattern="^\d{11}$" required value=
                                                <label for="contact">Contact Number</label>
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="input-field col s6">
                                                <i class="material-icons prefix">account_circle</i>
                                                <input placeholder="Username" name="username" id="u
                                                class="validate" required value="<%=session.getAttr
                                                <label for="username">Username</label>
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="input-field col s6">
                                                <i class="material-icons prefix">lock_open</i>
                                                <input name="password" id="password" type="password
                                                <label for="password">Change Password</label>
                                        </div>
                                        <div class="input-field col s6">
                                                <input name="passwordConfirm" id="passwordConfirm"
                                                 data-match-error="Passwords don't match">
                                                <label for="passwordConfirm">Confirm New Password</
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="col 6 ">
                                                <button class="btn waves-effect waves-light" type="
                                                <i class="material-icons right">send</i>
                                                </button>
                                        </div>
                                        <div class="col 6 ">
                                                <button class="red btn waves-effect waves-light" ty
                                                <i class="material-icons right">replay</i>
                                                </button>
                                        </div>
                                </div>
                        </form>
                </div>
        </div>

        <%              break;
                        case 2: %>

        <!-- Student Side -->
```

```
    <ul id="slide-out" class="side-nav fixed light-blue lighten-1">
    <li><div class="userView">
            <div class="background">
                    <img src="/MealPlanning/assets/imgs/default/blue-gradient.jpg" height="200px" width="300px">
            </div>
            <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
            href="/MealPlanning/Edit_Profile">
                    <%= session.getAttribute("username") %></a>
            <div class="row">
                    <div class="col s10">
                            <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                    </div>
                    <div class="col s1">
                            <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                            href="/MealPlanning/Log-in">
                            <i class="material-icons">input</i></a>
                    </div>
            </div>
    </div></li>
    <li><a href="/MealPlanning/Student/Add_Patient"><i class="material-icons">add</i>Add Patient</a></li>
    <li><div class="divider"></div></li>
    <li><a href="/MealPlanning/Student/View_My_Patients"><i class="material-icons">list</i>Patient List</a></li
    <li><div class="divider"></div></li>
    <!--
    <li><a class="subheader">Subheader</a></li>
    <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
     -->
    </ul>
    <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

<!-- Student Page Layout -->
        <div class="row">
                <div class="col s8 offset-s3 z-depth-3">
                        <h4 class="light-blue darken-2 white-text">    Edit Profile</h4>
                        <br />
                        <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Edi
                                <div class="row">
                                        <div class="input-field col s6">
                                                <i class="material-icons prefix">email</i>
                                                <input placeholder="Email" name="email" id="email"
                                                class="validate" required  value="<%=session.getAtt
                                                <label for="email">Email</label>
                                        </div>
                                        <div class="input-field col s6">
                                                <i class="material-icons prefix">phone</i>
                                                <input placeholder="xxxxxxxxxx" name="contact" id=
                                                class="validate" pattern="^\d{11}$" required value=
                                                <label for="contact">Contact Number</label>
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="input-field col s6">
                                                <i class="material-icons prefix">account_circle</i>
                                                <input placeholder="Username" name="username" id="u
                                                class="validate" required value="<%=session.getAttr
                                                <label for="username">Username</label>
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="input-field col s6">
                                                <i class="material-icons prefix">lock_open</i>
                                                <input name="password" id="password" type="password
                                                <label for="password">Change Password</label>
                                        </div>
                                        <div class="input-field col s6">
                                                <input name="passwordConfirm" id="passwordConfirm"
                                                  data-match-error="Passwords don't match" required>
                                                <label for="passwordConfirm">Confirm New Password</
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="col 6 ">
                                                <button class="btn waves-effect waves-light" type="
                                                <i class="material-icons right">send</i>
                                                </button>
                                        </div>
                                        <div class="col 6 ">
                                                <button class="red btn waves-effect waves-light" ty
                                                <i class="material-icons right">replay</i>
                                                </button>
                                        </div>
                                </div>
                        </form>
                </div>

        </div>

    <%              break;
                    case 3: %>
                    <ul id="slide-out" class="side-nav fixed red lighten-1">
    <li><div class="userView">
            <div class="background">
```

```
                        <img src="/MealPlanning/assets/imgs/default/red-gradient.svg" height="200px" width="300px">
                    </div>
                    <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                    href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                    <div class="row">
                        <div class="col s10">
                            <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                            <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                             href="/MealPlanning/Log-in">
                            <i class="material-icons">input</i></a>
                        </div>
                    </div>
                </div></li>
                <li><a href="/MealPlanning/Patient/Input_Intake"><i class="material-icons">add</i>Input Intake</a></li>
                <li><a href="/MealPlanning/Patient/View_Meal_Plan"><i class="material-icons">assignment</i>View Feedback</a
                <li><div class="divider"></div></li>
                <li><a href="/MealPlanning/Patient/Food_List"><i class="material-icons">list</i>Check_Food_Item</a></li>
                <li><div class="divider"></div></li>
                <!--
                <li><a class="subheader">Subheader</a></li>
                <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
                 -->
                </ul>
                <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>
                <div class="row">
                        <div class="col s8 offset-s3 z-depth-3">
                            <h4 class="red darken-2 white-text">    Edit Profile</h4>
                            <br />
                            <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Edi
                                <div class="row">
                                    <div class="input-field col s6">
                                        <i class="material-icons prefix">email</i>
                                        <input placeholder="Email" name="email" id="email"
                                        class="validate" required  value="<%=session.getAtt
                                        <label for="email">Email</label>
                                    </div>
                                    <div class="input-field col s6">
                                        <i class="material-icons prefix">phone</i>
                                        <input placeholder="xxxxxxxxxx" name="contact" id=
                                        class="validate" pattern="^\d{11}$" required value=
                                        <label for="contact">Contact Number</label>
                                    </div>
                                </div>
                                <div class="row">
                                    <div class="input-field col s6">
                                        <i class="material-icons prefix">account_circle</i>
                                        <input placeholder="Username" name="username" id="u
                                        class="validate" required value="<%=session.getAttr
                                        <label for="username">Username</label>
                                    </div>
                                </div>
                                <div class="row">
                                    <div class="input-field col s6">
                                        <i class="material-icons prefix">lock_open</i>
                                        <input name="password" id="password" type="password
                                        <label for="password">Change Password</label>
                                    </div>
                                    <div class="input-field col s6">
                                        <input name="passwordConfirm" id="passwordConfirm"
                                         data-match-error="Passwords don't match" required>
                                        <label for="passwordConfirm">Confirm New Password</
                                    </div>
                                </div>
                                <div class="row">
                                    <div class="col 6 ">
                                        <button class="btn waves-effect waves-light" type="
                                        <i class="material-icons right">send</i>
                                        </button>
                                    </div>
                                    <div class="col 6 ">
                                        <button class="red btn waves-effect waves-light" ty
                                        <i class="material-icons right">replay</i>
                                        </button>
                                    </div>
                                </div>
                            </form>
                        </div>

                    </div>
                <%              break;
                        } %>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<script type="text/javascript" src="/MealPlanning/assets/js/jquery-3.0.0.min.js"></script>
<script type="text/javascript" src="/MealPlanning/assets/js/jquery.validate.min.js"></script>
<script type="text/javascript" src="/MealPlanning/assets/js/materialize.min.js"></script>
<script type="text/javascript" src="/MealPlanning/assets/js/jquery.dataTables.min.js"></script>
<script type="text/javascript" src="/MealPlanning/assets/js/dataTables.material.min.js"></script>
```

```
<script type="text/javascript" src="/MealPlanning/assets/js/function.js"></script>
<script type="text/javascript" src="/MealPlanning/assets/js/validator.js"></script>

</body>
</html>


<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
        <link rel='stylesheet' href='/MealPlanning/assets/css/materialize.css'>
        <link rel='stylesheet' href='/MealPlanning/assets/css/layout.css'>
        <link rel="shortcut icon" href="data:image/x-icon;," type="image/x-icon">


<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Log-in Diet Portal</title>
</head>
<body class="teal">
        <div class="container">
                <div class="row">
                <br /><br /><br />
                        <div class="col s4 offset-s4 z-depth-3 white">
                                <div class="col s10 offset-s1">
                                        <br /><br />
                                        <div class="row">
                                                <h5 class="teal-text">Log-in to Weight Management Portal</h5>
                                        </div>
                                        <hr><br>
                                        <form class="form-signin" method="post" action="/MealPlanning/UserVerificat
                                                <div class="row">
                                                        <div class="input-field">
                                                                <i class="material-icons prefix">account_circle</i>
                                                                <input placeholder="Username" name="username" id="u
                                                                <label for="username">Username</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field">
                                                                <i class="material-icons prefix">lock_open</i>
                                                                <input name="password" id="password" type="password
                                                                <label for="password">Password</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <a href="/MealPlanning/Forget_Password">Forgot password?</a
                                                </div>
                                                <div class="row">
                                                        <div class="col s10 offset-s2">
                                                                <button class="btn waves-effect waves-light" type="
                                                                <i class="material-icons right">send</i>
                                                                </button>
                                                        </div>
                                                </div>
                                        </form>
                                </div>
                        </div>
                </div>
        </div>

<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ include file = '/WEB-INF/jsp/MainPage/header.jsp' %>
<%      String subhead = "";
        if(null == session.getAttribute("role")){
                subhead = "Login";
%>              <meta http-equiv="Refresh" content="3;url=/MealPlanning/Login">
<%      } else if(session.getAttribute("role").equals(0)){
                subhead = "/Login";
%>              <meta http-equiv="Refresh" content="3;url=/MealPlanning/Admin/Add_Teacher">
<%      } else if(session.getAttribute("role").equals(1)){
                subhead = "/Admin/Add_Teacher";
%>              <meta http-equiv="Refresh" content="3;url=/MealPlanning/Teacher/Add_Student">
<%      } else if(session.getAttribute("role").equals(2)){
                subhead = "/Teacher/Add_Student";
%>              <meta http-equiv="Refresh" content="3;url=/MealPlanning/Student/Add_Patient">
<%      }else if(session.getAttribute("role").equals(3)){
                subhead = "/Student/Add_Patient";
%>              <meta http-equiv="Refresh" content="3;url=/MealPlanning/Patient">
<%      }
%>
<title>Access Denied</title>
</head>
<body>
        <div class=container>
                <div class="row">
                <br /><br /><br />
                        <div class="col s6 offset-s3 z-depth-3">
                                <div class="row">
```

```
                                        <div class="col s12">
                                                <h3 class="orange white-text">Access Denied!</h3>
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="col s9">
                                                <h5>You don't have to access this page</h5>
                                                Redirecting to <%=subhead %>
                                        </div>
                                        <div class="col s3 valign-wrapper">
                                                <i class="medium material-icons red-text">report_problem</i>
                                        </div>
                                </div>
                        </div>
                </div>
        </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>

<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Forget Password</title>
</head>
<body class="teal">
        <div class="container">
                <div class="row">
                <br /><br /><br />
                        <div class="col s6 offset-s3 z-depth-3 white">
                                <div class="col s10 offset-s1">
                                        <br /><br />
                                        <div class="row">
                                                <h5 class="teal-text">Reset Password</h5>
                                        </div>
                                        <hr><br>
                                        <form class="form-signin" method="post" action="/MealPlanning/Forget_Passwo
                                                <div class="row">
                                                        <p>Please enter your email address</p>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field">
                                                                <i class="material-icons prefix">email</i>
                                                                <input placeholder="Email" name="email" id="email"
                                                                 class="validate" required>
                                                                <label for="email">Email</label>
                                                        </div>
                                                </div>

                                                <div class="row">
                                                        <div class="col s6">
                                                                <button class="btn waves-effect waves-light" type="
                                                                <i class="material-icons right">send</i>
                                                                </button>
                                                        </div>
                                                        <div class="col s6">
                                                                <a class="btn waves-effect waves-light" href="/Meal
                                                                <i class="material-icons right">arrow_back</i>
                                                                </a>
                                                        </div>
                                                </div>
                                        </form>
                                </div>
                        </div>
                </div>
        </div>

<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>

<%@ page import='models.*'%>
<%@ page import='dbutils.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Food List/Patient</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed red lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/red-gradient.svg" height="200px" width="300px">
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                 href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
```

```
        </div></li>
        <li><a href="/MealPlanning/Patient/Input_Intake"><i class="material-icons">add</i>Input Intake</a></li>
        <li><a href="/MealPlanning/Patient/View_Meal_Plan?week=<%=DBDataExtract.getLastWeek((Integer)session.getAtt
        <i class="material-icons">assignment</i>View Feedback</a></li>
        <li><div class="divider"></div></li>
        <li class="active"><a href="/MealPlanning/Patient/Food_List"><i class="material-icons">list</i>Check_Food_It
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="red darken-2 white-text">    Food List</h4>
                                <br />
                                <div>
                                        <table class="mdl-data-table" id="teacherList">
                                                <thead>
                                                        <tr>
                                                                <th>Group</th>
                                                                <th>Name</th>
                                                                <th>CHO(g)</th>
                                                                <th>PRO(g)</th>
                                                                <th>FAT(g)</th>
                                                                <th>Energy(kcal)</th>
                                                                <th>wt. A.P.(g)</th>
                                                                <th>wt. E.P.(g)</th>
                                                                <th>Measure</th>
                                                        </tr>
                                                </thead>
                                                <tbody>
                                                        <%
                                                                ArrayList<FoodListModel> foodList = DBDataExtract.g
                                                                for (FoodListModel food : foodList) {
                                                        %>
                                                        <tr>
                                                                <td><%=food.getGroup()%></td>
                                                                <td><%=food.getName()%></td>
                                                                <td><%=(food.getgCHO() == 0) ? "-" : food.getgCHO()%
                                                                <td><%=(food.getgPRO() == 0) ? "-" : food.getgPRO()%
                                                                <td><%=(food.getgFAT() == 0) ? "-" : food.getgFAT()%
                                                                <td><%=food.getEnergy()%></td>
                                                                <td><%=(food.getWtAP()== 0) ? "-" : food.getWtAP()%
                                                                <td><%=food.getWtEP()%></td>
                                                                <td><%=food.getMeasure()%></td>
                                                        </tr>
                                                        <%
                                                                }
                                                        %>
                                                </tbody>

                                        </table>
                                </div>
                    </div>

            </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='dbutils.*'%>
<%@ page import='models.*'%>
<%@ page import='operations.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Daily Intake/Patient</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed red lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/red-gradient.svg" height="200px" width="300px">
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li class="active"><a href="/MealPlanning/Patient/Input_Intake"><i class="material-icons">add</i>Input Inta
```

```
        <li><a href="/MealPlanning/Patient/View_Meal_Plan?week=<%=DBDataExtract.getLastWeek((Integer)session.getAtt
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Patient/Food_List"><i class="material-icons">list</i>Check_Food_Item</a></li>
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

        <div id="intakeModal" class="modal">
        <div class="modal-content">
                <h4>Submit Daily Food Intake</h4>
                <p>You can no longer edit your food intake for the day once you submit it.</p>
                <p>Are you sure you want to submit all the food items?</p>
        </div>
        <div class="modal-footer">
                <a href="#!" class="modal-action modal-close waves-effect waves-green btn-flat">Cancel</a>
                <a id="confirm-intake" class="waves-effect waves-green btn-flat intakeConfirm">Confirm</a>
        </div>
        </div>

<!-- Page Layout here -->
        <div class="row">
                <div class="col s8 offset-s3 z-depth-3">
                        <h4 class="red darken-2 white-text">    Daily Intake</h4>
                        <br />
                        <%MenuModel menu = (MenuModel)session.getAttribute("smenu");
                        int count = 0;
                        boolean submitted = DBDataExtract.hasSubmittedIntake(menu.getId());
                        if(!submitted){
                        %>
                        <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Pat
                                <div class="row">
                                        <div class="input-field col s8">
                                                <input name="intake" id="intake" type="text" placeh
                                                class="validate" required>
                                                <label for="intake">Food</label>
                                        </div>
                                </div>
                                <input type="hidden" id="purpose" name="purpose" value="add">
                                <div class="row">
                                        <div class="col 6 ">
                                                <button class="btn waves-effect waves-light" type="
                                                <i class="material-icons right">add</i>
                                                </button>
                                        </div>
                                </div>
                        </form>
                        <%
                }
                %>

                        <div class="row">
                                <div class="col s6">
                                <table class="highlight">
                                        <% if(!submitted){%>
                                        <tr>
                                                <th>Food</th>
                                                <th>Option</th>
                                        </tr>
                                        <%
                                                ArrayList<String> foods = (ArrayList<String>)sessio
                                                for(String food: foods){
                                                        count++;
                                        %>
                                                <tr>
                                                        <td><%=food %></td>
                                                        <td>
                                                        <form id="removeFoodItem" method="post" act
                                                                <input type="hidden" id="intake" na
                                                                <input type="hidden" id="purpose" n
                                                                <button type="submit"
                                                                class="waves-effsect waves-light bt
                                                        </form>
                                                        <%} %>
                                                        </td>
                                                </tr>
                                        <%
                                                }else{
                                                        int id = (Integer)session.getAttribute("id"
                                                        IntakeModel intakes = DBDataExtract.getInta
                                                        for(String food: intakes.getFoodItem()){
                                        %>
                                                <tr>
                                                        <td><%=food %></td>
                                                </tr>
                                        <%
                                                        }
                                                }
                                        %>
```

```
                                        </table>
                                        </div>
                                </div>

                            <div class="row">
                                <%if(count > 0 && !submitted) {%>
                                        <button class="waves-effsect waves-light btn intakebtn" dat
                                                data-purpose="final" >
                                        Submit Daily Intakes</button>
                                <%
                                        }
                                %>
                            </div>
                    </div>
                        <div class="col s8 offset-s3 z-depth-3">
                            <h4 class="red darken-2 white-text">    Sample Menu for Today</h4>
                            <br />
                                <div class="row">
                                <h6 class="red darken-2 white-text" style="font-size:20px">Breakfast</h6>
                                </div>
                                <div class="row">
                                        <table class="highlighted">
                                                <tr>
                                                        <th>Food Item</th>
                                                        <th>Exchanges</th>
                                                </tr>
                                                <%

                                                        ArrayList<String> breakfast = menu.getBreak
                                                        ArrayList<String> lunch = menu.getLunch();
                                                        ArrayList<String> dinner = menu.getDinner()
                                                        for(int i=0; i<breakfast.size(); i++){
                                                %>
                                                <tr>
                                                        <td><%=breakfast.get(i) %></td>
                                                        <td><%=breakfast.get(i+1) %></td>
                                                </tr>
                                                <%
                                                        i++;
                                                        }
                                                %>
                                        </table>
                                </div>
                                <div class="row">
                                <h6 class="red darken-2 white-text" style="font-size:20px">Lunch</h6>
                                </div>
                                <div class="row">
                                        <table class="highlighted">
                                                <tr>
                                                        <th>Food Item</th>
                                                        <th>Exchanges</th>
                                                </tr>
                                                <%
                                                        for(int i=0; i<lunch.size(); i++){
                                                %>
                                                <tr>
                                                        <td><%=lunch.get(i) %></td>
                                                        <td><%=lunch.get(i+1) %></td>
                                                </tr>
                                                <%
                                                        i++;
                                                        }
                                                %>
                                        </table>
                                </div>
                                <div class="row">
                                <h6 class="red darken-2 white-text" style="font-size:20px">Dinner</h6>
                                </div>
                                <div class="row">
                                        <table class="highlighted">
                                                <tr>
                                                        <th>Food Item</th>
                                                        <th>Exchanges</th>
                                                </tr>
                                                <%
                                                        for(int i=0; i<dinner.size(); i++){
                                                %>
                                                <tr>
                                                        <td><%=dinner.get(i) %></td>
                                                        <td><%=dinner.get(i+1) %></td>
                                                </tr>
                                                <%
                                                        i++;
                                                        }

                                                %>
                                        </table>
                                </div>

                    </div>
            </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>
```

```
<%@ page import='dbutils.*'%>
<%@ page import='models.*'%>
<%@ page import='operations.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Daily Intake/Patient</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed red lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/red-gradient.svg" height="200px" width="300px">
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li class="active"><a href="/MealPlanning/Patient/Input_Intake"><i class="material-icons">add</i>Input Intak
        <li><a href="/MealPlanning/Patient/View_Meal_Plan?week=<%=DBDataExtract.getLastWeek((Integer)session.getAtt
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Patient/Food_List"><i class="material-icons">list</i>Check_Food_Item</a></li>
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

        <div id="intakeModal" class="modal">
        <div class="modal-content">
                <h4>Submit Daily Food Intake</h4>
                <p>You can no longer edit your food intake for the day once you submit it.</p>
                <p>Are you sure you want to submit all the food items?</p>
        </div>
        <div class="modal-footer">
                <a href="#!" class="modal-action modal-close waves-effect waves-green btn-flat">Cancel</a>
                <a id="confirm-intake" class="waves-effect waves-green btn-flat intakeConfirm">Confirm</a>
        </div>
        </div>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="red darken-2 white-text">    View Meal Plan and Feedback</h4
                                <br />
                                <p>
                                        There's no data available.
                                </p>
                    </div>
            </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='dbutils.*'%>
<%@ page import='models.*'%>
<%@ page import='operations.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Daily Intake/Patient</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed red lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/red-gradient.svg" height="200px" width="300px">
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
```

```
        </div></li>
        <li class="active"><a href="/MealPlanning/Patient/Input_Intake"><i class="material-icons">add</i>Input Intak
        <li><a href="/MealPlanning/Patient/View_Meal_Plan?week=<%=DBDataExtract.getLastWeek((Integer)session.getAtt
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Patient/Food_List"><i class="material-icons">list</i>Check_Food_Item</a></li>
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

        <div id="intakeModal" class="modal">
        <div class="modal-content">
                <h4>Submit Daily Food Intake</h4>
                <p>You can no longer edit your food intake for the day once you submit it.</p>
                <p>Are you sure you want to submit all the food items?</p>
        </div>
        <div class="modal-footer">
                <a href="#!" class="modal-action modal-close waves-effect waves-green btn-flat">Cancel</a>
                <a id="confirm-intake" class="waves-effect waves-green btn-flat intakeConfirm">Confirm</a>
        </div>
        </div>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="red darken-2 white-text">    Daily Intake</h4>
                                <br />
                                <p>
                                        There's no sample menu for today.
                                </p>
                    </div>
            </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='dbutils.*'%>
<%@ page import='operations.*'%>
<%@ page import='models.*'%>
<%@ page import='java.text.DecimalFormat'%>
<%@ page import='java.text.DateFormat'%>
<%@ page import='java.text.SimpleDateFormat'%>
<%@ page import='java.util.ArrayList'%>
<%@ page import='java.util.Date'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Daily Intake/Patient</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed red lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/red-gradient.svg" height="200px" width="300px">
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Patient/Input_Intake"><i class="material-icons">add</i>Input Intake</a></li>
        <li class="active"><a href="/MealPlanning/Patient/View_Meal_Plan?week=<%=DBDataExtract.getLastWeek((Integer
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Patient/Food_List"><i class="material-icons">list</i>Check_Food_Item</a></li>
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3" id="mealPlan">
                                <h4 class="red darken-2 white-text">    View Meal Plan and Feedback</h4
                                <br />
                                <form class="s-12" method="POST" action="/MealPlanning/Patient/View_Meal_Plan">
                                <div class="row">
                                        <input type="hidden" name="purpose" id="purpose" value="ct">
                                        <div class="input-field col s4">
                                                <select id="week" name="week">
```

```
                                        <%      ArrayList<Date> dates = (ArrayList<Date>)session.getAttribute("date
                                        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                                        DecimalFormat df = new DecimalFormat("#.#");
                                        String sdate = "";
                                        Date curr = (Date)session.getAttribute("curr");
                                        int patient = (Integer)session.getAttribute("patient");
                                        String scurr = dateFormat.format(curr);
                                        MenuModel menu = DBDataExtract.getSampleMenu(patient, curr);
                                                MealPlanModel mpm = DBDataExtract.getMealPlan(menu.getWeekId());
                                                int id = (Integer)session.getAttribute("patient");
                                                IntakeModel intakes = DBDataExtract.getIntakes(id, menu.getDate());
                                                session.setAttribute("intakeId", intakes.getId());
                                        for(Date date: dates){
                                                sdate = dateFormat.format(date) + "";
                                        %>
                                                <option value="<%=sdate%>"
                                                <%=     (scurr.equals(sdate)) ? "selected" : ""%>
                                                ><%=sdate%></option>
                                        <%
                                        }

                                        %>
                                                </select>
                                                <label for="week">Date</label>
                                        </div>
                                                <div class="input-field col s3">
                                                        <button class="btn waves-effect waves-light" type="submit"
                                                        </button>
                                                </div>
                                </div>
                                </form>
                                        <div class="row">
                                                        <div class = "form-group">
                                                                <div class="input-field col s3">
                                                                        <label for = "tea"> Energy(kcal): </label>
                                                                         <input class="validate" type = "number" id
                                                                         min="0" value="<%= intakes.getEnergy()%>">
                                                                </div>
                                                                <div class="input-field col s3">
                                                                        <label for = "gCHO"> Carbohydrates(g): </la
                                                                        <input type = "number" id = "gCHO" name = "
step = "1" readonly
                                                                        min="0" value="<%= intakes.getgCHO()%>">
                                                                </div>
                                                                <div class="input-field col s3">
                                                                        <label for = "gPRO"> Protein(g): </label>
                                                                        <input type = "number" id = "gPRO" name = "
step = "1" readonly
                                                                        min="0" value="<%= intakes.getgPRO()%>">
                                                                </div>
                                                                <div class="input-field col s3">
                                                                        <label for = "gFAT"> Fat(g): </label>
                                                                        <input type = "number" id = "gFAT" name = "
step = "1" readonly
                                                                        min="0" value="<%= intakes.getgFAT()%>">
                                                                </div>
                                                        </div>
                                                        <div class="row">
                                                                <div class="input-field col s12">
                                                                <textarea id="textarea1" name="textarea1" class="ma
                                                                 <label for="textarea1">General Feedback</label>
                                                        </div>
                                                        </div>
                                                </div>


                                        <div class="row">
                                                <div class="col s6">
                                                <div class="row">
                                        <h6 class="red darken-2 white-text" style="font-size:20px">Daily Intake</h6
                                        </div>
                                                <table class="highlight">
                                                        <%

                                                                        for(String food: intakes.getFoodItem()){
                                                        %>
                                                                <tr>
                                                                        <td><%=food %></td>
                                                                </tr>
                                                        <%
                                                                        }
                                                        %>
                                                </table>
                                                </div>
                                        </div>

                        </div>
                            <div class="col s8 offset-s3 z-depth-3">
                                        <h4 class="red darken-2 white-text">    Meal Plan Sample Menu</h4>
                                        <br />
                                                <div class="row">
                                                <div class="input-field col s4">
```

```
                                <label for = "classification"> Body Mass Index: </l
                                <input class="validate" type = "number" id = "bmi"
                                value="<%=df.format( mpm.getBmi())%>" readonly>
                        </div>
                        <div class="input-field col s4">
                                <label for = "classification"> Classification: </la
                                <input class="validate" type = "text" id = "classif
                                value="<%=mpm.getClassification()%>" readonly>
                        </div>
                        <div class="input-field col s4">
                                <label for = "dbw"> Desired Body Weight(kg): </labe
                                <input class="validate" type = "number" id = "dbw"
                                value="<%=df.format( mpm.getDbw())%>" readonly>
                        </div>
                </div>
                <div class="row">
                        <div class = "form-group">
                                <div class="input-field col s3">
                                        <label for = "tea"> Total Energy Allowance(
                                        <input class="validate" type = "number" id
                                        value="<%=df.format(mpm.getTea()) %>" reado
                                </div>
                                <div class="input-field col s3">
                                        <label for = "gCHO"> Grams of Carbohydrate:
                                        <input type = "number" id = "gCHO" name = "
                                </div>
                                <div class="input-field col s3">
                                        <label for = "gPRO"> Grams of Protein: </la
                                        <input type = "number" id = "gPRO" name = "
                                </div>
                                <div class="input-field col s3">
                                        <label for = "gFAT"> Grams of Fat: </label>
                                        <input type = "number" id = "gFAT" name = "
                                </div>
                        </div>
        </div>
        <div class="row">
        <h6 class="red darken-2 white-text" style="font-size:20px">Breakfast</h6>
        </div>
        <div class="row">
                        <table class="highlighted">
                                <tr>
                                        <th>Food Item</th>
                                        <th>Exchanges</th>
                                </tr>
                                <%

                                        ArrayList<String> breakfast = menu.getBreak
                                        ArrayList<String> lunch = menu.getLunch();
                                        ArrayList<String> dinner = menu.getDinner()
                                        for(int i=0; i<breakfast.size(); i++){
                                %>
                                <tr>
                                        <td><%=breakfast.get(i) %></td>
                                        <td><%=breakfast.get(i+1) %></td>
                                </tr>
                                <%
                                        i++;
                                        }
                                %>
                        </table>
                </div>
                <div class="row">
        <h6 class="red darken-2 white-text" style="font-size:20px">Lunch</h6>
        </div>
        <div class="row">
                        <table class="highlighted">
                                <tr>
                                        <th>Food Item</th>
                                        <th>Exchanges</th>
                                </tr>
                                <%
                                        for(int i=0; i<lunch.size(); i++){
                                %>
                                <tr>
                                        <td><%=lunch.get(i) %></td>
                                        <td><%=lunch.get(i+1) %></td>
                                </tr>
                                <%
                                        i++;
                                        }
                                %>
                        </table>
                </div>
                <div class="row">
        <h6 class="red darken-2 white-text" style="font-size:20px">Dinner</h6>
        </div>
        <div class="row">
                        <table class="highlighted">
                                <tr>
                                        <th>Food Item</th>
                                        <th>Exchanges</th>
```

```
                                                                </tr>
                                                                <%
                                                                        for(int i=0; i<dinner.size(); i++){
                                                                %>
                                                                <tr>
                                                                        <td><%=dinner.get(i) %></td>
                                                                        <td><%=dinner.get(i+1) %></td>
                                                                </tr>
                                                                <%
                                                                        i++;
                                                                        }

                                                                %>
                                                        </table>
                                                </div>

                                        </div>
                                </div>

<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Add Patient/Student</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed light-blue lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/blue-gradient.jpg" height="200px" width="300px"
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Student/Add_Patient"><i class="material-icons">add</i>Add Patient</a></li>
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Student/View_My_Patients"><i class="material-icons">list</i>Patient List</a></li
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                            <h4 class="light-blue darken-2 white-text">    Calculate Diet for <%=se
                            <br />
                            <form class="form-signin" id="diet-form" method="post" action="/MealPlanning/Studen
                                    <div class="row">
                                            <div class="input-field col s4">
                                                    <label for = "height"> Height(cm): </label>
                                                    <input class="validate" type = "number" id = "heigh
                                                    min="125" max="244" required>
                                            </div>
                                            <div class="input-field col s4">
                                                    <label for = "weight"> Weight(kg): </label>
                                                    <input class="validate" type = "number" id = "weigh
                                                    min="35" max="120" required>
                                            </div>
                                    </div>
                                    <div class="row">
                                            <div class="input-field col s-6">
                                            Activity: <br>
                                            <p>
                                                    <input id="act1" type = "radio" name = "activity" v
                                                    <label for = "act1">  Bed rest but mobile (hospital
                                            </p><p>
                                                    <input id="act2" type = "radio" name = "activity" v
                                                    <label for = "act2">  Sedentary (mostly sitting) </
                                            </p><p>
                                                    <input id="act3" type = "radio" name = "activity" v
                                                    <label for = "act3">  Light (tailor, nurse, physici
                                            </p><p>
                                                    <input id="act4" type = "radio" name = "activity" v
                                                    <label for = "act4">  Moderate (carpenter, painter,
                                            </p><p>
                                                    <input id="act5" type = "radio" name = "activity" v
```

```
                                                        <label for = "act5">  Very Active (swimming, lumber
                                                </p>
                                                </div>
                                        </div>
                                        <br>

                                        <div class="row">
                                                <div class="col 6 ">
                                                        <button class="btn waves-effect waves-light" type="
                                                        <i class="material-icons right">send</i>
                                                        </button>
                                                </div>
                                                <div class="col 6 ">
                                                        <button class="red btn waves-effect waves-light" ty
                                                        <i class="material-icons right">replay</i>
                                                        </button>
                                                </div>
                                        </div>
                                </form>
                        </div>

                </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>s


<%@ page import='dbutils.*'%>
<%@ page import='operations.*'%>
<%@ page import='models.*'%>
<%@ page import='java.text.DecimalFormat'%>
<%@ page import='java.text.SimpleDateFormat'%>
<%@ page import='java.text.DateFormat'%>
<%@ page import='java.util.Calendar'%>
<%@ page import='java.util.Date'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Add Patient/Student</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed light-blue lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/blue-gradient.jpg" height="200px" width="300px"
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Student/Add_Patient"><i class="material-icons">add</i>Add Patient</a></li>
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Student/View_My_Patients"><i class="material-icons">list</i>Patient List</a></li
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="light-blue darken-2 white-text">    Diet Presciption for <%=
                                <br />
                                <%
                                MealPlanModel mpm = (MealPlanModel)session.getAttribute("mealPlan");
                                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                                String start = "", end = "";
                                start = dateFormat.format(mpm.getStart()) + "";
                                end = dateFormat.format(mpm.getEnd()) + "";
                                Calendar cal = Calendar.getInstance();
                                Date today = cal.getTime();
                                String daycode = DBDataExtract.getCode(today);
                                DietCalculation dietcalc = (DietCalculation)session.getAttribute("dietcalc");
                                        DecimalFormat df = new DecimalFormat("#.#");%>
                                <form class="form-signin" id="diet-form" method="get" action="/MealPlanning/Student
                                        <div class="row">
                                                <div class="input-field col s3">
                                                <input name="from" id="from" type="hidden"
                                                        value="addReport">
                                                <input name="patient" id="patient" type="hidden"
                                                        value="<%=mpm.getPatientId()%>">
                                                <input class="datepicker" name="start" id="start" type="dat
```

```
                class="validate" value="<%=start%>" required disable
        <label for="end">Start Date</label>
        </div>
        <div class="input-field col s3">
        <% %>
        <input class="datepicker" name="end" id="end" type="date"
                class="validate" value="<%=end%>" required disabled
        <label for="end">End Date</label>
        </div>
</div>
<div class="row">
        <div class="input-field col s4">
                <label for = "height"> Height(cm): </label>
                <input class="validate" type = "number" id = "heigh
                min="125" max="244" value="<%= dietcalc.getHeight()
        </div>
        <div class="input-field col s4">
                <label for = "weight"> Weight(kg): </label>
                <input class="validate" type = "number" id = "weigh
                min="35" max="120" value="<%=dietcalc.getWeight() %
        </div>
        <div class="input-field col s4">
                <label for = "activity"> Activity: </label>
                <input class="validate" type = "text" id = "activity
                value="<%=dietcalc.getActivity()%>" readonly>
        </div>
</div>
<div class="row">
        <div class="input-field col s4">
                <label for = "bmi"> Body Mass Index: </label>
                <input class="validate" type = "number" id = "bmi"
                value="<%=df.format( dietcalc.getBMI())%>" readonly
        </div>
        <div class="input-field col s4">
                <label for = "classification"> Classification: </la
                <input class="validate" type = "text" id = "classifi
                value="<%=dietcalc.getClassification()%>" readonly>
        </div>
        <div class="input-field col s4">
                <label for = "dbw"> Desired Body Weight(kg): </labe
                <input class="validate" type = "number" id = "dbw"
                value="<%=df.format( dietcalc.getDBW())%>" readonly
        </div>
</div>
<div class="row">
        <div class = "form-group">
                <div class="input-field col s3">
                        <label for = "tea"> Total Energy Allowance(
                        <input class="validate" type = "number" id
                        value="<%=df.format(dietcalc.getTEA()) %>"
                </div>
                <div class="input-field col s3">
                        <label for = gCHO"> Grams of Carbohydrate:
                        <input type = "number" id = "gCHO" name = "
                </div>
                <div class="input-field col s3">
                        <label for = "gPRO"> Grams of Protein: </la
                        <input type = "number" id = "gPRO" name = "
                </div>
                <div class="input-field col s3">
                        <label for = "gFAT"> Grams of Fat: </label>
                        <input type = "number" id = "gFAT" name = "
                </div>
        </div>
</div>
<table class="highlight">
<thead>
        <tr>
        <th>Food Group</th>
        <th>No. of Exchanges</th>
        <th>CHO(g)</th>
        <th>PRO(g)</th>
        <th>FAT(g)</th>
        <th>Energy(kcal)</th>
        </tr>
</thead>

<tbody>
        <tr>
        <td>Vegetable, List I-A</td>
        <td><%=df.format(dietcalc.getExVegA())%></td>
        <td>3</td>
        <td>1</td>
        <td>-</td>
        <td>16</td>
        </tr>
        <tr>
        <td>Vegetable, List I-B</td>
        <td><%=df.format(dietcalc.getExVegB()) %></td>
        <td>3</td>
        <td>1</td>
        <td>-</td>
```

```
                                                <td>16</td>
                                                </tr>
                                                <tr>
                                                <td>Fruit, List II</td>
                                                <td><%=df.format(dietcalc.getExFruit()) %></td>
                                                <td><%=df.format(dietcalc.getExFruit()*10) %></td>
                                                <td>-</td>
                                                <td>-</td>
                                                <td><%=df.format(dietcalc.getExFruit()*40) %></td>
                                                </tr>
                                                <tr>
                                                <td>Milk, List III</td>
                                                <td><%=df.format(dietcalc.getExMilk()) %></td>
                                                <td><%=df.format(dietcalc.getExMilk()*12) %></td>
                                                <td><%=df.format(dietcalc.getExMilk()*8) %></td>
                                                <td><%=df.format(dietcalc.getExMilk()*10) %></td>
                                                <td><%=df.format(dietcalc.getExMilk()*170) %></td>
                                                </tr>
                                                <tr>
                                                <td>Sugar, List VII</td>
                                                <td><%=df.format(dietcalc.getExSugar()) %></td>
                                                <td><%=df.format(dietcalc.getExSugar()*5) %></td>
                                                <td>-</td>
                                                <td>-</td>
                                                <td><%=df.format(dietcalc.getExSugar()*20) %></td>
                                                </tr>
                                                <tr>
                                                <td>Rice, List IV</td>
                                                <td><%=df.format(dietcalc.getExRice()) %></td>
                                                <td><%=df.format(dietcalc.getExRice()*23) %></td>
                                                <td><%=df.format(dietcalc.getExRice()*2) %></td>
                                                <td>-</td>
                                                <td><%=df.format(dietcalc.getExRice()*100) %></td>
                                                </tr>
                                                <tr>
                                                <td>Meat Low-Fat, List Va</td>
                                                <td><%=df.format(dietcalc.getExMeatL()) %></td>
                                                <td>-</td>
                                                <td><%=df.format(dietcalc.getExMeatL()*8) %></td>
                                                <td><%=df.format(dietcalc.getExMeatL()*1) %></td>
                                                <td><%=df.format(dietcalc.getExMeatL()*41) %></td>
                                                </tr>
                                                <tr>
                                                <td>Meat, Medium-Fat List Vb</td>
                                                <td><%=df.format(dietcalc.getExMeatM()) %></td>
                                                <td>-</td>
                                                <td><%=df.format(dietcalc.getExMeatM()*8) %></td>
                                                <td><%=df.format(dietcalc.getExMeatM()*6) %></td>
                                                <td><%=df.format(dietcalc.getExMeatM()*86) %></td>
                                                </tr>
                                                <tr>
                                                <td>Fat, List VI</td>
                                                <td><%=df.format(dietcalc.getExFat()) %></td>
                                                <td>-</td>
                                                <td>-</td>
                                                <td><%=df.format(dietcalc.getExFat()*5) %></td>
                                                <td><%=df.format(dietcalc.getExFat()*45) %></td>
                                                </tr>
                                                <tr>
                                                <th>TOTAL</th>
                                                <th></th>
                                                <th><%=df.format(dietcalc.getComputedCHO()) %></th>
                                                <th><%=df.format(dietcalc.getComputedPRO()) %></th>
                                                <th><%=df.format(dietcalc.getComputedFAT()) %></th>
                                                <th><%=df.format(dietcalc.getComputedEnergy()) %></th>
                                                </tr>
                                            </tbody>
                                        </table>

                                            <div class="row">
                                                <input type="hidden" id="daycode" name="daycode" value="<%=
                                                <div class="input-field col s3">
                                                        <label for = "codeinput"> Enter Approval Code </lab
                                                        <input class="validate zx" type = "password" id = "
                                                </div>
                                                <div class="input-field col s3">
                                                        <button class="btn waves-effect waves-light" type="
                                                        id="cmenu">Create Menu
                                                        <i class="material-icons right">send</i>
                                                        </button>
                                                </div>
                                            </div>
                                    </form>
                        </div>

                    </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>s


<%@ page import='models.*'%>
<%@ page import='dbutils.*'%>
<%@ page import='java.util.ArrayList'%>
```

```
<%@ page import='java.text.DecimalFormat'%>
<%@ page import='java.text.SimpleDateFormat'%>
<%@ page import='java.text.DateFormat'%>
<%@ page import='java.util.Calendar'%>
<%@ page import='java.util.Date'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>My Patient/Student</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed light-blue lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/blue-gradient.jpg" height="200px" width="300px"
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Student/Add_Patient"><i class="material-icons">add</i>Add Patient</a></li>
        <li><div class="divider"></div></li>
        <li class="active"><a href="/MealPlanning/Student/View_My_Patients"><i class="material-icons">list</i>Patie
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s9 offset-s3 z-depth-3">
                                <h4 class="light-blue darken-2 white-text">    My Patients</h4>
                                <br />
                                <div>
                                        <table class="mdl-data-table" id="teacherList">
                                                <thead>
                                                        <tr>
                                                                <th>Username</th>
                                                                <th>Name</th>
                                                                <th>Email</th>
                                                                <th>Birthdate</th>
                                                                <th>Sex</th>
                                                                <th>Option</th>
                                                        </tr>
                                                </thead>
                                                <tbody>
                                                        <%
                                                                Calendar cal = Calendar.getInstance();
                                                                DateFormat dateFormat = new SimpleDateFormat("yyyy-
                                                                Date curr = cal.getTime();
                                                                Date last = cal.getTime();
                                                                ArrayList<PatientModel> patients = DBUserExtract.ge
                                                                for (PatientModel patient : patients) {
                                                                        if(!patient.isDeactivated() &&
                                                                                        session.getAttribute("id").
                                                        %>
                                                        <tr>
                                                                <td><%=patient.getUsername()%></td>
                                                                <td><%=patient.getFname() + " " + patient.getMname(
                                                                <td><%=patient.getEmail()%></td>
                                                                <td><%=dateFormat.format(patient.getBirthdate())%><
                                                                <td><%=patient.getSex()%></td>

                                                                <td class="check">
                                                                <%
                                                                        int patientId = patient.getId();
                                                                        if(DBDataExtract.getLastWeek(patientId) - 1
                                                                %>
                                                                        <a class="waves-effsect waves-light btn crea
                                                                                        href="/MealPlanning/Student
                                                                                        Calculate Diet</a>
                                                                <%
                                                                        } else{
                                                                                last = DBDataExtract.getLastSampleD
                                                                                if(last == null){

                                                                %>
                                                                                        <a class="waves-effsect wav
                                                                                        href="/MealPlanning/Student
                                                                                        Create Menu</a>
```

```
                                                                <%
                                                                        }else{
                                                                                if(curr.after(last)){
                                                                %>
                                                                                <a class="waves-effsect wav
                                                                                href="/MealPlanning/Student
                                                                                Calculate Diet</a>
                                                                <%
                                                                                }else{
                                                                                        if(!DBDataExtract.is

                                                                %>
                                                                                        <a class="waves-eff
                                                                                        href="/MealPlanning
                                                                                        Create Menu</a>
                                                                <%
                                                                                        }
                                                                                }
                                                                        }

                                                                %>
                                                                <%if(DBDataExtract.getAllDates(patientId).size() >
                                                                        <a class="waves-effsect waves-light btn vie
                                                                        View Intake</a>
                                                                <% }%>
                                                                </td>
                                                        </tr>
                                                        <%              }
                                                        }
                                                        %>
                                                </tbody>

                                        </table>
                                </div>
                        </div>
                </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Add Patient/Student</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed light-blue lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/blue-gradient.jpg" height="200px" width="300px">
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li class="active"><a href="/MealPlanning/Student/Add_Patient"><i class="material-icons">add</i>Add Patient
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Student/View_My_Patients"><i class="material-icons">list</i>Patient List</a></li
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                            <h4 class="light-blue darken-2 white-text">    Add Patient</h4>
                            <br />
                            <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Stu
                                    <div class="row">
                                            <div class="input-field col s4">
                                                    <input placeholder="First name" name="fname" id="fna
                                                    class="validate" required>
                                                    <label for="fname">First Name</label>
                                            </div>
                                            <div class="input-field col s4">
                                                    <input placeholder="Middle name" name="mname" id="m
                                                    class="validate" required>
                                                    <label for="mname">Middle Name</label>
```

```
                                                        </div>
                                                        <div class="input-field col s4">
                                                                <input placeholder="Last name" name="lname" id="lnam
                                                                class="validate" required>
                                                                <label for="lname">Last Name</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">redeem</i>
                                                                <input class="datepicker" name="birthdate" id="birth
                                                                class="validate" required>
                                                                <label for="birthdate">Birthdate</label>
                                                        </div>
                                                        <div class="input-field col s6">
                                                                <p>
                                                                        <label>Sex</label><br />
                                                                <input name="sex" type="radio" id="male" value="mal
                                                                <label for="male">Male</label>
                                                                <input name="sex" type="radio" id="female" value="f
                                                                <label for="female">Female</label>
                                                            </p>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">email</i>
                                                                <input placeholder="Email" name="email" id="email"
                                                                class="validate" required>
                                                                <label for="email">Email</label>
                                                        </div>
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">phone</i>
                                                                <input placeholder="xxxxxxxxxx" name="contact" id=
                                                                class="validate" pattern="^\d{11}$" required>
                                                                <label for="contact">Contact Number</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">account_circle</i>
                                                                <input placeholder="Username" name="username" id="u
                                                                class="validate" required>
                                                                <label for="username">Username</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">lock_open</i>
                                                                <input name="password" id="password" type="password
                                                                <label for="password">Password</label>
                                                        </div>
                                                        <div class="input-field col s6">
                                                                <input name="passwordConfirm" id="passwordConfirm"
                                                                  data-match-error="Passwords don't match" required>
                                                                <label for="passwordConfirm">Confirm Password</labe
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="col 6 ">
                                                                <button class="btn waves-effect waves-light" type="
                                                                <i class="material-icons right">send</i>
                                                                </button>
                                                        </div>
                                                        <div class="col 6 ">
                                                                <button class="red btn waves-effect waves-light" ty
                                                                <i class="material-icons right">replay</i>
                                                                </button>
                                                        </div>
                                                </div>
                                        </form>
                        </div>

                </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='dbutils.*'%>
<%@ page import='operations.*'%>
<%@ page import='java.text.DecimalFormat'%>
<%@ page import='models.*'%>
<%@ page import='java.text.DecimalFormat'%>
<%@ page import='java.text.SimpleDateFormat'%>
<%@ page import='java.text.DateFormat'%>
<%@ page import='java.util.Calendar'%>
<%@ page import='java.util.Date'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Add Patient/Student</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed light-blue lighten-1">
```

```
    <li><div class="userView">
            <div class="background">
                    <img src="/MealPlanning/assets/imgs/default/blue-gradient.jpg" height="200px" width="300px">
            </div>
            <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
            href="/MealPlanning/Edit_Profile">
                    <%= session.getAttribute("username") %></a>
            <div class="row">
                    <div class="col s10">
                            <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                    </div>
                    <div class="col s1">
                            <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                            href="/MealPlanning/Log-in">
                            <i class="material-icons">input</i></a>
                    </div>
            </div>
    </div></li>
    <li><a href="/MealPlanning/Student/Add_Patient"><i class="material-icons">add</i>Add Patient</a></li>
    <li><div class="divider"></div></li>
    <li><a href="/MealPlanning/Student/View_My_Patients"><i class="material-icons">list</i>Patient List</a></li
    <li><div class="divider"></div></li>
    <!--
    <li><a class="subheader">Subheader</a></li>
    <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
     -->
    </ul>
    <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

<!-- Page Layout here -->
        <div class="row">
                <div class="col s8 offset-s3 z-depth-3">
                        <h4 class="light-blue darken-2 white-text">    Sample Menu for <%=sessio
                        <br />
                        <%
                        MealPlanModel mpm = (MealPlanModel)session.getAttribute("mealPlan");
                        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                        String last = "";
                        Calendar cal = Calendar.getInstance();
                        Date today = cal.getTime();
                        String daycode = DBDataExtract.getCode(today);
                        Date date = DBDataExtract.getLastDay(mpm.getPatientId());
                        if(date == null){
                                date = DBDataExtract.getStartDay(mpm.getPatientId());
                        }else{
                                cal.setTime(date);
                                cal.add(Calendar.DATE, 1);
                                date = cal.getTime();
                        }
                        last = dateFormat.format(date) + "";
                        session.setAttribute("fdate", date);
                        MealPlanModel dietcalc = (MealPlanModel)session.getAttribute("mealPlan");
                                DecimalFormat df = new DecimalFormat("#.#");%>
                        <form class="form-signin" id="diet-form" method="post" action="/MealPlanning/Studen
                                        <div class="row">
                                                <div class="input-field col s4">
                                                <input class="datepicker" name="fdate" id="fdate" type="dat
                                                        class="validate" required disabled value="<%=last%>
                                                <label for="fdate">Date</label>
                                                </div>
                                        </div>
                                        <div class="row">
                                        <h6 class="light-blue darken-2 white-text" style="font-size:20px">C
                                        </div>
                                        <div class="row">
                                                <div class="input-field col s1">
                                                        <input name="etvegA" id="etvegA" type="number" clas
                                                        step="0.5" value="<%=dietcalc.getExVegA()%>" readon
                                                </div>
                                                <div class="input-field col s1">
                                                        <input name="etvegB" id="etvegB" type="number" clas
                                                        step="0.5" value="<%=dietcalc.getExVegB() %>" reado
                                                </div>
                                                <div class="input-field col s1">
                                                        <input name="etfruit" id="etfruit" type="number" cl
                                                        step="0.5" value="<%=dietcalc.getExFruit() %>" read
                                                </div>
                                                <div class="input-field col s1">
                                                        <input name="etmilk" id="etmilk" type="number" clas
                                                        step="0.5" value="<%=dietcalc.getExMilk() %>" reado
                                                </div>
                                                <div class="input-field col s1">
                                                        <input name="etsugar" id="etsugar" type="number" cl
                                                        step="0.5" value="<%=dietcalc.getExSugar() %>" read
                                                </div>
                                                <div class="input-field col s1">
                                                        <input name="etrice" id="etrice" type="number" clas
                                                        step="0.5" value="<%=dietcalc.getExRice() %>" reado
                                                </div>
                                                <div class="input-field col s1">
                                                        <input name="etmeatL" id="etmeatL" type="number" cl
                                                        step="0.5" value="<%=dietcalc.getExMeatL() %>" read
```

```
        </div>
        <div class="input-field col s1">
                <input name="etmeatM" id="etmeatM" type="number" cl
                step="0.5" value="<%=dietcalc.getExMeatM() %>" read
        </div>
        <div class="input-field col s1">
                <input name="etfat" id="etfat" type="number" class=
                step="0.5" value="<%=dietcalc.getExFat() %>" readon
        </div>
</div>

<div class="row">
<h6 class="light-blue darken-2 white-text" style="font-size:20px">E
</div>
<div class="row">
        <div class="input-field col s3">
                <input name="bvegA" id="bvegA" type="text" class="v
                <label for="bvegA">Vegetable A</label>
        </div>
        <div class="input-field col s1">
                <input name="ebvegA" id="ebvegA" type="number" clas
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="bvegB" id="bvegB" type="text" class="v
                <label for="bvegB">Vegetable B</label>
        </div>
        <div class="input-field col s1">
                <input name="ebvegB" id="ebvegB" type="number" clas
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="bfruit" id="bfruit" type="text" class=
                <label for="bfruit">Fruit</label>
        </div>
        <div class="input-field col s1">
                <input name="ebfruit" id="ebfruit" type="number" cl
                step="0.5" value="0"><label>exchange</label>
        </div>
</div>
<div class="row">
        <div class="input-field col s3">
                <input name="bmilk" id="bmilk" type="text" class="v
                <label for="bmilk">Milk</label>
        </div>
        <div class="input-field col s1">
                <input name="ebmilk" id="ebmilk" type="number" clas
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="bsugar" id="bsugar" type="text" class=
                <label for="bsugar">Sugar</label>
        </div>
        <div class="input-field col s1">
                <input name="ebsugar" id="ebsugar" type="number" cl
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="brice" id="brice" type="text" class="v
                <label for="brice">Rice</label>
        </div>
        <div class="input-field col s1">
                <input name="ebrice" id="ebrice" type="number" clas
                step="0.5" value="0"><label>exchange</label>
        </div>
</div>
<div class="row">
        <div class="input-field col s3">
                <input name="bmeatL" id="bmeatL" type="text" class=
                <label for="bmeatL">Meat Low Fat</label>
        </div>
        <div class="input-field col s1">
                <input name="ebmeatL" id="ebmeatL" type="number" cl
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="bmeatM" id="bmeatM" type="text" class=
                <label for="bmeatM">Meat Medium Fat</label>
        </div>
        <div class="input-field col s1">
                <input name="ebmeatM" id="ebmeatM" type="number" cl
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="bfat" id="bfat" type="text" class="val
                <label for="bfat">Fat</label>
        </div>
        <div class="input-field col s1">
                <input name="ebfat" id="ebfat" type="number" class=
                step="0.5" value="0"><label>exchange</label>
        </div>
</div>
```

```
<div class="row">
<h6 class="light-blue darken-2 white-text" style="font-size:20px">L
</div>
<div class="row">
        <div class="input-field col s3">
                <input name="lvegA" id="lvegA" type="text" class="v
                <label for="lvegA">Vegetable A</label>
        </div>
        <div class="input-field col s1">
                <input name="elvegA" id="elvegA" type="number" clas
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="lvegB" id="lvegB" type="text" class="v
                <label for="lvegB">Vegetable B</label>
        </div>
        <div class="input-field col s1">
                <input name="elvegB" id="elvegB" type="number" clas
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="lfruit" id="lfruit" type="text" class=
                <label for="lfruit">Fruit</label>
        </div>
        <div class="input-field col s1">
                <input name="elfruit" id="elfruit" type="number" cl
                step="0.5" value="0"><label>exchange</label>
        </div>
</div>
<div class="row">
        <div class="input-field col s3">
                <input name="lmilk" id="lmilk" type="text" class="v
                <label for="lmilk">Milk</label>
        </div>
        <div class="input-field col s1">
                <input name="elmilk" id="elmilk" type="number" clas
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="lsugar" id="lsugar" type="text" class=
                <label for="lsugar">Sugar</label>
        </div>
        <div class="input-field col s1">
                <input name="elsugar" id="elsugar" type="number" cl
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="lrice" id="lrice" type="text" class="v
                <label for="lrice">Rice</label>
        </div>
        <div class="input-field col s1">
                <input name="elrice" id="elrice" type="number" clas
                step="0.5" value="0"><label>exchange</label>
        </div>
</div>
<div class="row">
        <div class="input-field col s3">
                <input name="lmeatL" id="lmeatL" type="text" class=
                <label for="lmeatL">Meat Low Fat</label>
        </div>
        <div class="input-field col s1">
                <input name="elmeatL" id="elmeatL" type="number" cl
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="lmeatM" id="lmeatM" type="text" class=
                <label for="lmeatM">Meat Medium Fat</label>
        </div>
        <div class="input-field col s1">
                <input name="elmeatM" id="elmeatM" type="number" cl
                step="0.5" value="0"><label>exchange</label>
        </div>
        <div class="input-field col s3">
                <input name="lfat" id="lfat" type="text" class="val
                <label for="lfat">Fat</label>
        </div>
        <div class="input-field col s1">
                <input name="elfat" id="elfat" type="number" class=
                step="0.5" value="0"><label>exchange</label>
        </div>
</div>
<div class="row">
<h6 class="light-blue darken-2 white-text" style="font-size:20px">D
</div>
<div class="row">
        <div class="input-field col s3">
                <input name="dvegA" id="dvegA" type="text" class="v
                <label for="dvegA">Vegetable A</label>
        </div>
        <div class="input-field col s1">
                <input name="edvegA" id="edvegA" type="number" clas
                step="0.5" value="0"><label>exchange</label>
```

```
                                        </div>
                                        <div class="input-field col s3">
                                                <input name="dvegB" id="dvegB" type="text" class="v
                                                <label for="dvegB">Vegetable B</label>
                                        </div>
                                        <div class="input-field col s1">
                                                <input name="edvegB" id="edvegB" type="number" clas
                                                step="0.5" value="0"><label>exchange</label>
                                        </div>
                                        <div class="input-field col s3">
                                                <input name="dfruit" id="dfruit" type="text" class=
                                                <label for="dfruit">Fruit</label>
                                        </div>
                                        <div class="input-field col s1">
                                                <input name="edfruit" id="edfruit" type="number" cl
                                                step="0.5" value="0"><label>exchange</label>
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="input-field col s3">
                                                <input name="dmilk" id="dmilk" type="text" class="v
                                                <label for="dmilk">Milk</label>
                                        </div>
                                        <div class="input-field col s1">
                                                <input name="edmilk" id="edmilk" type="number" clas
                                                step="0.5" value="0"><label>exchange</label>
                                        </div>
                                        <div class="input-field col s3">
                                                <input name="dsugar" id="dsugar" type="text" class=
                                                <label for="dsugar">Sugar</label>
                                        </div>
                                        <div class="input-field col s1">
                                                <input name="edsugar" id="edsugar" type="number" cl
                                                step="0.5" value="0"><label>exchange</label>
                                        </div>
                                        <div class="input-field col s3">
                                                <input name="drice" id="drice" type="text" class="v
                                                <label for="drice">Rice</label>
                                        </div>
                                        <div class="input-field col s1">
                                                <input name="edrice" id="edrice" type="number" clas
                                                step="0.5" value="0"><label>exchange</label>
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="input-field col s3">
                                                <input name="dmeatL" id="dmeatL" type="text" class=
                                                <label for="dmeatL">Meat Low Fat</label>
                                        </div>
                                        <div class="input-field col s1">
                                                <input name="edmeatL" id="edmeatL" type="number" cl
                                                step="0.5" value="0"><label>exchange</label>
                                        </div>
                                        <div class="input-field col s3">
                                                <input name="dmeatM" id="dmeatM" type="text" class=
                                                <label for="dmeatM">Meat Medium Fat</label>
                                        </div>
                                        <div class="input-field col s1">
                                                <input name="edmeatM" id="edmeatM" type="number" cl
                                                step="0.5" value="0"><label>exchange</label>
                                        </div>
                                        <div class="input-field col s3">
                                                <input name="dfat" id="dfat" type="text" class="val
                                                <label for="dfat">Fat</label>
                                        </div>
                                        <div class="input-field col s1">
                                                <input name="edfat" id="edfat" type="number" class=
                                                step="0.5" value="0"><label>exchange</label>
                                        </div>
                                </div>
                                <div class="row">
                                        <input type="hidden" id="daycode" name="daycode" value="<%=
                                        <div class="input-field col s3">
                                                <label for = "codeinput"> Enter Approval Code </lab
                                                <input class="validate qw" type = "password" id = "
                                        </div>
                                        <div class="input-field col s6">
                                                <button class="tooltipped btn waves-effect waves-li
                                                data-position="right" data-delay="50" data-tooltip=
                                                <i class="material-icons right">send</i>
                                                </button>
                                                <a class="tooltipped"
                                                data-position="right" data-delay="50" data-tooltip=

                                                </a>
                                        </div>
                                </div>
                        </form>
                </div>

        </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>
```

```
<%@ page import='dbutils.*'%>
<%@ page import='operations.*'%>
<%@ page import='models.*'%>
<%@ page import='java.text.DecimalFormat'%>
<%@ page import='java.text.DateFormat'%>
<%@ page import='java.text.SimpleDateFormat'%>
<%@ page import='java.util.ArrayList'%>
<%@ page import='java.util.Date'%>
<%@ page import='java.util.Calendar'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>View Intake/Student</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed light-blue lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/blue-gradient.jpg" height="200px" width="300px"
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Student/Add_Patient"><i class="material-icons">add</i>Add Patient</a></li>
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Student/View_My_Patients"><i class="material-icons">list</i>Patient List</a></li
        <li><div class="divider"></div></li>
        <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
<!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3" id="mealPlan">
                            <h4 class="light-blue darken-2 white-text">    View Intake of <%=sessio
                            <br />
                            <form class="s-12" method="POST" action="/MealPlanning/Patient/View_Intake">
                            <div class="row">
                                    <input type="hidden" name="purpose" id="purpose" value="ct">
                                    <div class="input-field col s4">
                                            <select id="week" name="week">
                                    <%      ArrayList<Date> dates = (ArrayList<Date>)session.getAttribute("date
                                    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                                    DecimalFormat df = new DecimalFormat("#.#");
                                    String sdate = "";
                                    Calendar cal = Calendar.getInstance();
                                    Date today = cal.getTime();
                                    String daycode = DBDataExtract.getCode(today);
                                    Date curr = (Date)session.getAttribute("curr");
                                    int patient = (Integer)session.getAttribute("patient");
                                    String scurr = dateFormat.format(curr);
                                    MenuModel menu = DBDataExtract.getSampleMenu(patient, curr);
                                            MealPlanModel mpm = DBDataExtract.getMealPlan(menu.getWeekId());
                                            int id = (Integer)session.getAttribute("patient");
                                            IntakeModel intakes = DBDataExtract.getIntakes(id, menu.getDate());
                                            session.setAttribute("intakeId", intakes.getId());
                                    for(Date date: dates){
                                            sdate = dateFormat.format(date) + "";
                                    %>
                                            <option value="<%=sdate%>"
                                            <%=     (scurr.equals(sdate)) ? "selected" : ""%>
                                            ><%=sdate%></option>
                                    <%
                                    }

                                    %>
                                            </select>
                                            <label for="week">Date</label>
                                    </div>
                                            <div class="input-field col s3">
                                                    <button class="btn waves-effect waves-light" type="submit"
                                                    </button>
                                            </div>
                            </div>
                            </form>
                            <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Stu
                                            <div class="row">
                                                    <div class = "form-group">
```

```
                                                        <div class="input-field col s3">
                                                                <label for = "tea"> Energy(kcal): </label>
                                                                 <input class="validate" type = "number" id
                                                                 min="0" value="<%= intakes.getEnergy()%>">
                                                        </div>
                                                        <div class="input-field col s3">
                                                                <label for = "gCHO"> Carbohydrates(g): </la
                                                                <input type = "number" id = "gCHO" name = "
step = "1" required
                                                                min="0" value="<%= intakes.getgCHO()%>">
                                                        </div>
                                                        <div class="input-field col s3">
                                                                <label for = "gPRO"> Protein(g): </label>
                                                                <input type = "number" id = "gPRO" name = "
step = "1" required
                                                                min="0" value="<%= intakes.getgPRO()%>">
                                                        </div>
                                                        <div class="input-field col s3">
                                                                <label for = "gFAT"> Fat(g): </label>
                                                                <input type = "number" id = "gFAT" name = "
step = "1" required
                                                                min="0" value="<%= intakes.getgFAT()%>">
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field col s12">
                                                        <textarea id="textarea1" name="textarea1" class="ma
                                                         <label for="textarea1">General Feedback</label>
                                                </div>
                                                </div>
                                        </div>
                                        <input type="hidden" id="purpose" name="purpose" value="gf">
                                        <div class="row">
                                                <input type="hidden" id="daycode" name="daycode" value="<%=
                                                <div class="input-field col s3">
                                                        <label for = "codeinput"> Enter Approval Code </lab
                                                        <input class="validate zx" type = "password" id = "
                                                </div>
                                                <div class="input-field col s3">
                                                        <button class="btn waves-effect waves-light" type="
                                                        id="cmenu" disabled>Give Feedback
                                                        <i class="material-icons right">add</i>
                                                        </button>
                                                </div>
                                        </div>
                                </form>


                                <div class="row">
                                        <div class="col s6">
                                        <div class="row">
                                <h6 class="light-blue darken-2 white-text" style="font-size:20px">Daily Int
                                </div>
                                        <table class="highlight">
                                                <%

                                                        for(String food: intakes.getFoodItem()){
                                                %>
                                                        <tr>
                                                                <td><%=food %></td>
                                                        </tr>
                                                <%
                                                                }
                                                %>
                                        </table>
                                        </div>
                                </div>

                        </div>
                            <div class="col s8 offset-s3 z-depth-3">
                                    <h4 class="light-blue darken-2 white-text">    Meal Plan Sample Menu</h
                                    <br />
                                            <div class="row">
                                            <div class="input-field col s4">
                                                        <label for = "classification"> Body Mass Index: </l
                                                        <input class="validate" type = "number" id = "bmi"
                                                        value="<%=df.format( mpm.getBmi())%>" readonly>
                                                </div>
                                                <div class="input-field col s4">
                                                        <label for = "classification"> Classification: </la
                                                        <input class="validate" type = "text" id = "classif
                                                        value="<%=mpm.getClassification()%>" readonly>
                                                </div>
                                                <div class="input-field col s4">
                                                        <label for = "dbw"> Desired Body Weight(kg): </labe
                                                        <input class="validate" type = "number" id = "dbw"
                                                        value="<%=df.format( mpm.getDbw())%>" readonly>
                                                </div>
                                        </div>
                                        <div class="row">
                                                <div class = "form-group">
                                                        <div class="input-field col s3">
```

```
                                        <label for = "tea"> Total Energy Allowance(
                                        <input class="validate" type = "number" id
                                        value="<%=df.format(mpm.getTea()) %>" reado
                                </div>
                                <div class="input-field col s3">
                                        <label for = "gCHO"> Grams of Carbohydrate:
                                        <input type = "number" id = "gCHO" name = "
                                </div>
                                <div class="input-field col s3">
                                        <label for = "gPRO"> Grams of Protein: </la
                                        <input type = "number" id = "gPRO" name = "
                                </div>
                                <div class="input-field col s3">
                                        <label for = "gFAT"> Grams of Fat: </label>
                                        <input type = "number" id = "gFAT" name = "
                                </div>
                        </div>
                </div>
                <div class="row">
                <h6 class="light-blue darken-2 white-text" style="font-size:20px">Breakfast
                </div>
                <div class="row">
                        <table class="highlighted">
                                <tr>
                                        <th>Food Item</th>
                                        <th>Exchanges</th>
                                </tr>
                                <%

                                        ArrayList<String> breakfast = menu.getBreak
                                        ArrayList<String> lunch = menu.getLunch();
                                        ArrayList<String> dinner = menu.getDinner()
                                        for(int i=0; i<breakfast.size(); i++){
                                %>
                                <tr>
                                        <td><%=breakfast.get(i) %></td>
                                        <td><%=breakfast.get(i+1) %></td>
                                </tr>
                                <%
                                        i++;
                                        }
                                %>
                        </table>
                </div>
                <div class="row">
                <h6 class="light-blue darken-2 white-text" style="font-size:20px">Lunch</h6
                </div>
                <div class="row">
                        <table class="highlighted">
                                <tr>
                                        <th>Food Item</th>
                                        <th>Exchanges</th>
                                </tr>
                                <%
                                        for(int i=0; i<lunch.size(); i++){
                                %>
                                <tr>
                                        <td><%=lunch.get(i) %></td>
                                        <td><%=lunch.get(i+1) %></td>
                                </tr>
                                <%
                                        i++;
                                        }
                                %>
                        </table>
                </div>
                <div class="row">
                <h6 class="light-blue darken-2 white-text" style="font-size:20px">Dinner</h
                </div>
                <div class="row">
                        <table class="highlighted">
                                <tr>
                                        <th>Food Item</th>
                                        <th>Exchanges</th>
                                </tr>
                                <%
                                        for(int i=0; i<dinner.size(); i++){
                                %>
                                <tr>
                                        <td><%=dinner.get(i) %></td>
                                        <td><%=dinner.get(i+1) %></td>
                                </tr>
                                <%
                                        i++;
                                        }

                                %>
                        </table>
                </div>

        </div>
</div>
```

```
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='java.util.Calendar'%>
<%@ page import='java.util.Date'%>
<%@ page import='dbutils.*'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Add Food/Teacher</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed green lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/green-gradient.jpg" height="200px" width="300px
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Teacher/Add_Student"><i class="material-icons">add</i>Add Student</a></li>
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Teacher/Reassign_Students"><i class="material-icons">list</i>Reassign Student </
        <li><a href="/MealPlanning/Teacher/Reassign_Patients"><i class="material-icons">assignment_ind</i>Reassign
        <li><div class="divider"></div></li>
        <li class="active"><a href="/MealPlanning/Teacher/Add_Food_Item"><i class="material-icons">add</i>Add New F
        <li><div class="divider"></div></li>
        <%
                Calendar cal = Calendar.getInstance();
                Date date = cal.getTime();
        %>
        <li>    Today's Code: <%=DBDataExtract.getCode(date) %></li>
        <li><div class="divider"></div></li>
                <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="green darken-2 white-text">    Add Food</h4>
                                <br />
                                <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Tea
                                        <div class="row">
                                                <div class="input-field col s4">
                                                        <input placeholder="Food name" name="foodname" id="
                                                        class="validate" required>
                                                        <label for="foodname">Food</label>
                                                </div>
                                                <div class="input-field col s4">
                                                        <select id="group" name="group">
                                                                <option value="vegA">vegetable A</option>
                                                                <option value="vegB">vegetable B</option>
                                                                <option value="fruit">fruit</option>
                                                                <option value="milk_whole">milk whole</opti
                                                                <option value="milk_lowfat">milk lowfat</op
                                                                <option value="milk_whole">milk skimmed</op
                                                                <option value="rice">rice</option>
                                                                <option value="meatfish_lowfat">meatfish_lo
                                                                <option value="meatfish_mediumfat">meatfish
                                                                <option value="meatfish_highfat">meatfish_h
                                                                <option value="fat">fat</option>

                                                        </select>
                                                        <label for="group">Food Group</label>
                                                </div>
                                        </div>
                                        <div class="row">
                                                <div class="input-field col s3">
                                                        <input id="wtAP" type="number" name ="wtAP" min="0"
                                                        class="validate" required>
                                                        <label for="wtAP">Weight as purchased(g)</label>
                                                </div>
                                                <div class="input-field col s3">
                                                        <input id="wtEP" type="number" name ="wtEP" min="1"
                                                        class="validate" required>
                                                        <label for="wtEP">Weight edible part(g)</label>
                                                </div>
```

```
                                                        <div class="input-field col s3">
                                                                <input placeholder="Measure" name="measure" id="mea
                                                                class="validate" required>
                                                                <label for="measure">Measure</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="col 6 ">
                                                                <button class="btn waves-effect waves-light" type="
                                                                <i class="material-icons right">send</i>
                                                                </button>
                                                        </div>
                                                        <div class="col 6 ">
                                                                <button class="red btn waves-effect waves-light" ty
                                                                <i class="material-icons right">replay</i>
                                                                </button>
                                                        </div>
                                                </div>
                                        </form>
                        </div>

                </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='models.*'%>
<%@ page import='dbutils.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ page import='java.util.Calendar'%>
<%@ page import='java.util.Date'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Student List/Teacher</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed green lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/green-gradient.jpg" height="200px" width="300px
                </div>
                <h5><a href="#!name"><span class="white-text name"><%= session.getAttribute("username") %></span></
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li><a href="/MealPlanning/Teacher/Add_Student"><i class="material-icons">add</i>Add Student</a></li>
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Teacher/Reassign_Students"><i class="material-icons">list</i>Reassign Student</a
        <li class="active"><a href="/MealPlanning/Teacher/Reassign_Patients"><i class="material-icons">assignment_in
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Teacher/Add_Food_Item"><i class="material-icons">add</i>Add New Food Item</a></l
        <li><div class="divider"></div></li>
        <%
                Calendar cal = Calendar.getInstance();
                Date date = cal.getTime();
        %>
        <li>    Today's Code: <%=DBDataExtract.getCode(date) %></li>
        <li><div class="divider"></div></li>
                <!--
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s9 offset-s3 z-depth-3">
                            <h4 class="green darken-2 white-text">    Reassign Patient</h4>
                            <br />
                            <div>
                                    <table class="mdl-data-table" id="teacherList">
                                            <thead>
                                                    <tr>
                                                            <th>Username</th>
                                                            <th>Name</th>
                                                            <th>Email</th>
                                                            <th>Contact Number</th>
                                                            <th>Reassign</th>
                                                    </tr>
                                            </thead>
                                            <tbody>
                                                    <%
                                                            ArrayList<PatientModel> patients = DBUserExtract.ge
                                                            for (PatientModel patient : patients) {
```

```
                                                        if(!patient.isDeactivated()) {
                                                %>
                                                <tr>
                                                        <td><%=patient.getUsername()%></td>
                                                        <td><%=patient.getFname() + " " + patient.getMname(
                                                        <td><%=patient.getEmail()%></td>
                                                        <td><%=patient.getContact()%></td>

                                                        <td class="check">
                                                                <div class="input-field col s7">
                                                                Student:
                                                                <select id="sel<%=patient.getId()%>" name="
                                                                        <%      ArrayList <StudentModel> st
                                                                        for(StudentModel student : students
                                                                                if(!student.isDeactivated()
                                                                        %>
                                                                                        <option value="<%=
                                                                                        <%if(patient.getStu
                                                                                                %>
                                                                                                selected
                                                                                        <%
                                                                                          }
                                                                                        %>
                                                                                                >
                                                                                        <%=student.getLname
                                                                                        </option>
                                                                        <%
                                                                                }
                                                                        }
                                                                        %>
                                                                </select>
                                                                </div>
                                                                <div class="col s2 ">
                                                                <button class="waves-effsect waves-light bt
                                                                Reassign</button>
                                                                </div>
                                                        </td>
                                                </tr>
                                                <%
                                                        }
                                                }
                                                %>
                                        </tbody>

                                </table>
                        </div>
                </div>
        </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='java.util.Calendar'%>
<%@ page import='java.util.Date'%>
<%@ page import='dbutils.*'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Add Student/Teacher</title>
</head>
<body>
        <!-- Sidenav -->
        <ul id="slide-out" class="side-nav fixed green lighten-1">
        <li><div class="userView">
                <div class="background">
                        <img src="/MealPlanning/assets/imgs/default/green-gradient.jpg" height="200px" width="300px
                </div>
                <a class="tooltipped transparent white-text" style="font-size:25px" data-position="bottom" data-del
                href="/MealPlanning/Edit_Profile">
                        <%= session.getAttribute("username") %></a>
                <div class="row">
                        <div class="col s10">
                                <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                        </div>
                        <div class="col s1">
                                <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                                href="/MealPlanning/Log-in">
                                <i class="material-icons">input</i></a>
                        </div>
                </div>
        </div></li>
        <li class="active"><a href="/MealPlanning/Teacher/Add_Student"><i class="material-icons">add</i>Add Student
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Teacher/Reassign_Students"><i class="material-icons">list</i>Reassign Student </
        <li><a href="/MealPlanning/Teacher/Reassign_Patients"><i class="material-icons">assignment_ind</i>Reassign
        <li><div class="divider"></div></li>
        <li><a href="/MealPlanning/Teacher/Add_Food_Item"><i class="material-icons">add</i>Add New Food Item</a></l
        <li><div class="divider"></div></li>
        <%
                Calendar cal = Calendar.getInstance();
                Date date = cal.getTime();
        %>
        <li>    Today's Code: <%=DBDataExtract.getCode(date) %></li>
        <li><div class="divider"></div></li>
        <!--
```

```
        <li><a class="subheader">Subheader</a></li>
        <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
         -->
        </ul>
        <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

    <!-- Page Layout here -->
            <div class="row">
                    <div class="col s8 offset-s3 z-depth-3">
                                <h4 class="green darken-2 white-text">    Add Student</h4>
                                <br />
                                <form class="form-signin" id="teacher-form" method="post" action="/MealPlanning/Tea
                                                <div class="row">
                                                        <div class="input-field col s4">
                                                                <input placeholder="First name" name="fname" id="fn
                                                                class="validate" required>
                                                                <label for="fname">First Name</label>
                                                        </div>
                                                        <div class="input-field col s4">
                                                                <input placeholder="Middle name" name="mname" id="m
                                                                class="validate" required>
                                                                <label for="mname">Middle Name</label>
                                                        </div>
                                                        <div class="input-field col s4">
                                                                <input placeholder="Last name" name="lname" id="lna
                                                                class="validate" required>
                                                                <label for="lname">Last Name</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">email</i>
                                                                <input placeholder="Email" name="email" id="email"
                                                                class="validate" required>
                                                                <label for="email">Email</label>
                                                        </div>
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">phone</i>
                                                                <input placeholder="xxxxxxxxxx" name="contact" id=
                                                                class="validate" pattern="^\d{11}$" required>
                                                                <label for="contact">Contact Number</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">account_circle</i>
                                                                <input placeholder="Username" name="username" id="u
                                                                class="validate" required>
                                                                <label for="username">Username</label>
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="input-field col s6">
                                                                <i class="material-icons prefix">lock_open</i>
                                                                <input name="password" id="password" type="password
                                                                <label for="password">Password</label>
                                                        </div>
                                                        <div class="input-field col s6">
                                                                <input name="passwordConfirm" id="passwordConfirm"
                                                                 data-match-error="Passwords don't match" required>
                                                                <label for="passwordConfirm">Confirm Password</labe
                                                        </div>
                                                </div>
                                                <div class="row">
                                                        <div class="col 6 ">
                                                                <button class="btn waves-effect waves-light" type="
                                                                <i class="material-icons right">send</i>
                                                                </button>
                                                        </div>
                                                        <div class="col 6 ">
                                                                <button class="red btn waves-effect waves-light" ty
                                                                <i class="material-icons right">replay</i>
                                                                </button>
                                                        </div>
                                                </div>
                                        </form>
                    </div>

            </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<%@ page import='models.*'%>
<%@ page import='dbutils.*'%>
<%@ page import='java.util.ArrayList'%>
<%@ page import='java.util.Calendar'%>
<%@ page import='java.util.Date'%>
<%@ include file='/WEB-INF/jsp/MainPage/header.jsp'%>
<title>Reassign Patient/Teacher</title>
</head>
<body>
        <!-- Sidenav -->
```

```
    <ul id="slide-out" class="side-nav fixed green lighten-1">
    <li><div class="userView">
            <div class="background">
                    <img src="/MealPlanning/assets/imgs/default/green-gradient.jpg" height="200px" width="300px
            </div>
            <h5><a href="#!name"><span class="white-text name"><%= session.getAttribute("username") %></span></
            <div class="row">
                    <div class="col s10">
                            <a href="#!email"><span class="white-text email"><%= session.getAttribute("email")
                    </div>
                    <div class="col s1">
                            <a class="tooltipped transparent white-text" data-position="bottom" data-delay="50"
                            href="/MealPlanning/Log-in">
                            <i class="material-icons">input</i></a>
                    </div>
            </div>
    </div></li>
    <li><a href="/MealPlanning/Teacher/Add_Student"><i class="material-icons">add</i>Add Student</a></li>
    <li><div class="divider"></div></li>
    <li class="active"><a href="/MealPlanning/Teacher/Reassign_Students"><i class="material-icons">list</i>Reas
    <li><a href="/MealPlanning/Teacher/Reassign_Patients"><i class="material-icons">assignment_ind</i>Reassign
    <li><div class="divider"></div></li>
    <li><a href="/MealPlanning/Teacher/Add_Food_Item"><i class="material-icons">add</i>Add New Food Item</a></l
    <li><div class="divider"></div></li>
    <%
            Calendar cal = Calendar.getInstance();
            Date date = cal.getTime();
    %>
    <li>    Today's Code: <%=DBDataExtract.getCode(date) %></li>
    <li><div class="divider"></div></li>
    <!--
    <li><a class="subheader">Subheader</a></li>
    <li><a class="waves-effect" href="#!">Third Link With Waves</a></li>
     -->
    </ul>
    <a href="#" data-activates="slide-out" class="button-collapse"><i class="material-icons">menu</i></a>

<!-- Page Layout here -->
        <div class="row">
                <div class="col s9 offset-s3 z-depth-3">
                        <h4 class="green darken-2 white-text">    Reassign Student</h4>
                        <br />
                        <div>
                                <table class="mdl-data-table" id="teacherList">
                                        <thead>
                                                <tr>
                                                        <th>Username</th>
                                                        <th>Name</th>
                                                        <th>Email</th>
                                                        <th>Contact Number</th>
                                                        <th>Reassign</th>
                                                </tr>
                                        </thead>
                                        <tbody>
                                                <%
                                                        ArrayList<StudentModel> students = DBUserExtract.ge
                                                        for (StudentModel student : students) {
                                                                if(!student.isDeactivated()) {
                                                %>
                                                <tr>
                                                        <td><%=student.getUsername()%></td>
                                                        <td><%=student.getFname() + " " + student.getMname(
                                                        <td><%=student.getEmail()%></td>
                                                        <td><%=student.getContact()%></td>

                                                        <td class="check">
                                                                <div class="input-field col s7">
                                                                Teacher:
                                                                <select id="sel<%=student.getId()%>" name="
                                                                        <%      ArrayList <TeacherModel> te
                                                                        for(TeacherModel teacher : teachers
                                                                                if(!teacher.isDeactivated()
                                                                        %>
                                                                                <option value="<%=
                                                                                <%if(student.getTea
                                                                                        %>
                                                                                        selected
                                                                                <%
                                                                                }
                                                                                %>
                                                                                >
                                                                                <%=teacher.getLname
                                                                                </option>
                                                                        <%
                                                                                }
                                                                        }
                                                                        %>
                                                                </select>
                                                                </div>
                                                                <div class="col s2 ">
                                                                <button class="waves-effsect waves-light bt
                                                                Reassign</button>
```

```
                                                                    </div>
                                                                </td>
                                                            </tr>
                                                            <%
                                                                    }
                                                                }
                                                            %>
                                                        </tbody>

                                                    </table>
                                                </div>
                                            </div>
                                        </div>
<%@ include file='/WEB-INF/jsp/MainPage/footer.jsp'%>


<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schema
  <display-name>MealPlanning</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <display-name>Login Servlet</display-name>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>servlet.LoginServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/Log-in</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Forget Password</display-name>
    <servlet-name>ForgetPassword</servlet-name>
    <servlet-class>servlet.ForgetPassword</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ForgetPassword</servlet-name>
    <url-pattern>/Forget_Password</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Edit Profile</display-name>
    <servlet-name>EditProfile</servlet-name>
    <servlet-class>servlet.EditProfile</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>EditProfile</servlet-name>
    <url-pattern>/Edit_Profile</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Add Student</display-name>
    <servlet-name>AddStudent</servlet-name>
    <servlet-class>servlet.AddStudent</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AddStudent</servlet-name>
    <url-pattern>/Teacher/Add_Student</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Add Food Item</display-name>
    <servlet-name>AddFoodItem</servlet-name>
    <servlet-class>servlet.AddFoodItem</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AddFoodItem</servlet-name>
    <url-pattern>/Teacher/Add_Food_Item</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Reassign Students</display-name>
    <servlet-name>ReassignStudents</servlet-name>
    <servlet-class>servlet.ReassignStudents</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ReassignStudents</servlet-name>
    <url-pattern>/Teacher/Reassign_Students</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Reassign Patients</display-name>
    <servlet-name>ReassignPatients</servlet-name>
    <servlet-class>servlet.ReassignPatients</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ReassignPatients</servlet-name>
    <url-pattern>/Teacher/Reassign_Patients</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Add Patient</display-name>
```

```
    <servlet-name>AddPatient</servlet-name>
    <servlet-class>servlet.AddPatient</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AddPatient</servlet-name>
    <url-pattern>/Student/Add_Patient</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View My Patients</display-name>
    <servlet-name>ViewMyPatients</servlet-name>
    <servlet-class>servlet.ViewMyPatients</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewMyPatients</servlet-name>
    <url-pattern>/Student/View_My_Patients</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Calculate Diet</display-name>
    <servlet-name>CalculateDiet</servlet-name>
    <servlet-class>servlet.DietCalculatorServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>CalculateDiet</servlet-name>
    <url-pattern>/Student/Calculate_Diet</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Create Menu</display-name>
    <servlet-name>CreateMenu</servlet-name>
    <servlet-class>servlet.CreateMenu</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>CreateMenu</servlet-name>
    <url-pattern>/Student/Create_Menu</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Add Teacher</display-name>
    <servlet-name>AddTeacher</servlet-name>
    <servlet-class>servlet.AddTeacher</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AddTeacher</servlet-name>
    <url-pattern>/Admin/Add_Teacher</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View Teachers</display-name>
    <servlet-name>ViewTeachers</servlet-name>
    <servlet-class>servlet.ViewTeachers</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewTeachers</servlet-name>
    <url-pattern>/Admin/View_Teachers</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>DeactivateUser</servlet-name>
    <url-pattern>/Admin/Deactivate_User</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Deactivate User</display-name>
    <servlet-name>DeactivateUser</servlet-name>
    <servlet-class>servlet.DeactivateUser</servlet-class>
  </servlet>
  <servlet>
    <display-name>View Students</display-name>
    <servlet-name>ViewStudents</servlet-name>
    <servlet-class>servlet.ViewStudents</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewStudents</servlet-name>
    <url-pattern>/Admin/View_Students</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View Patients</display-name>
    <servlet-name>ViewPatients</servlet-name>
    <servlet-class>servlet.ViewPatients</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewPatients</servlet-name>
    <url-pattern>/Admin/View_Patients</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Input Intake</display-name>
    <servlet-name>InputIntake</servlet-name>
    <servlet-class>servlet.InputIntake</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>InputIntake</servlet-name>
    <url-pattern>/Patient/Input_Intake</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>Food List</display-name>
    <servlet-name>FoodList</servlet-name>
    <servlet-class>servlet.FoodListServlet</servlet-class>
```

```
  </servlet>
  <servlet-mapping>
    <servlet-name>FoodList</servlet-name>
    <url-pattern>/Patient/Food_List</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View Meal Plan</display-name>
    <servlet-name>ViewMealPlan</servlet-name>
    <servlet-class>servlet.ViewMealPlan</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewMealPlan</servlet-name>
    <url-pattern>/Patient/View_Meal_Plan</url-pattern>
  </servlet-mapping>
  <servlet>
    <display-name>View Intake</display-name>
    <servlet-name>ViewIntake</servlet-name>
    <servlet-class>servlet.ViewIntake</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ViewIntake</servlet-name>
    <url-pattern>/Student/View_Intake</url-pattern>
  </servlet-mapping>
</web-app>
```

XII. Acknowledgement

Finally I've finished this, after a long time of procrastination, which I hate for knowing. Anyway, first, thanks family, especially to my sister who guided me through the entire procedure of doing this things, since it is not my field. Great thanks to my adviser who pushed me to finish this SP before I work in the industry, which might made me forgot that I didn't finish my degree yet. And also thanks to all of my professors.

The OJT really come in handy, for teaching this framework for my SP. The Java and web application development courses, especially the jsp, thanks to all of those.

Thanks to all of my friends and acquaintances in college who helped me thoughout these years. So much love to all, Block 12 2013, extended blockmates, especially to southern pips.

For the experiences and memories, all places, RH114, GAB 2nd floor and Luneta, I will never forget. Lakwatsa, gala, tambay laro, kalokohan, kain, kulit, ... di mauubusan, TMTM. Ayoko na, gusto ka na lang 'tong tapusin, BASTA salamat sa lahat lahat, di ko man mapasalamatan, salamat pa din,

Ms vale tarde que nunca