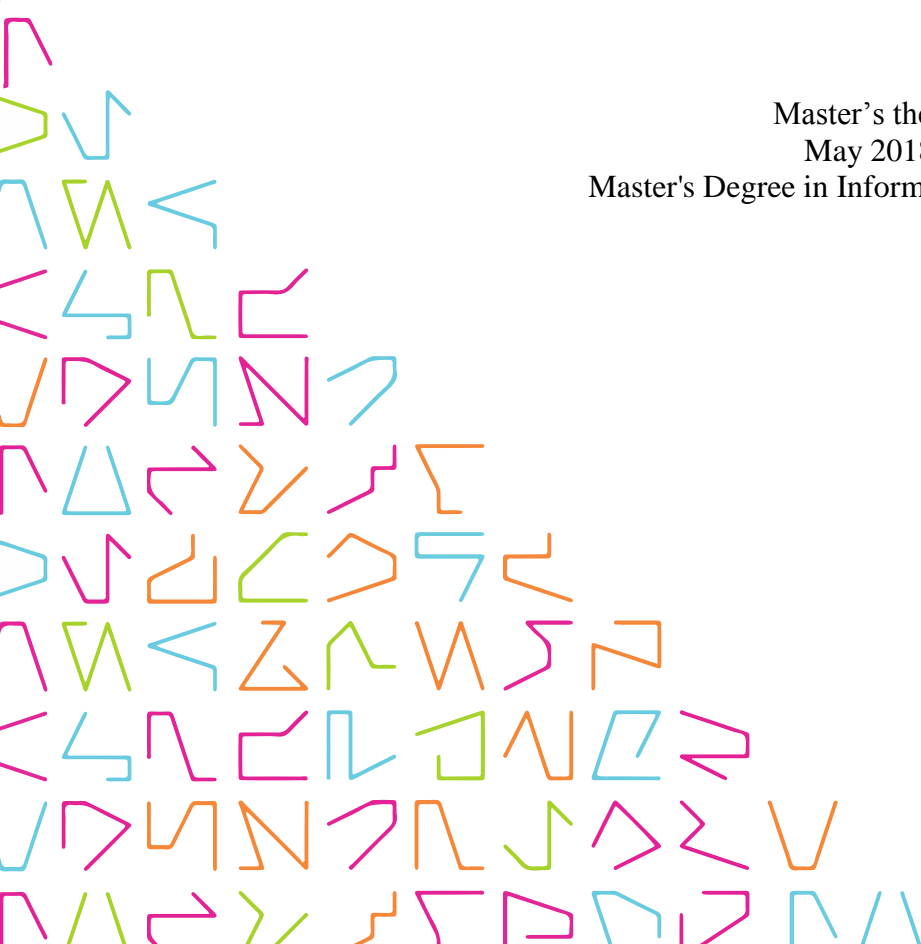


Health Fitness System

Healthfitness.com

Muhammad Jawad Ur Rehman Saif

Master's thesis
May 2018
Master's Degree in Information Technology



ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Master's Degree in Information Technology

Muhammad Jawad Ur Rehman Saif

Health Fitness System (Healthfitness.com)

Master thesis pages 45, appendices 17 pages
May 2018

Now a day's health fitness is an evolving issue for the people. People have no idea about their health maintenance. People's health, wellbeing and fitness are dependent on taking nutritious food and supplements. But they don't have an idea about their food intake and exercise. They also have no idea about the calories calculation and supplement intake. They don't know how much they must take water in a day, sleep in a day and how much their body burns calories when they do exercise, work, walk and even sleep.

The objective of this thesis is to implement the nutrition health fitness system for the health fitness club where the end user can track information about his nutrition related to food and exercise and his coach even can evaluate his progress. The significance of implementing this thesis is to give the awareness of food nutrition value, so the user can easily create an equilibrium between in his food nutrition chart and exercises.

The application is implemented by using most renowned programming tools and technologies. It is very substantial to build the application users friendly, so the user can understand and use it easily without any problem. This application has the client and server-side architecture. Normally client side has CSS, HTML and JavaScript. So, I am using Bootstrap (CSS HTML framework), Angular JS (JavaScript framework), JQuery (JavaScript function library). The server-side program has the need for a database with some programming language. I am using MySQL database with CakePHP (PHP framework). I am making RESTFUL API calls through which we can easily retrieve and store the data in the database. This thesis is demanding the extensive programming skills for design and implement Web applications.

Target Audience: technology serves every individual in several ways, crafting specialized products for small and large daily usage. This application is for the fitness clubs and sports organizations (for sports players) because they especially need to follow their food nutrition chart and exercise's plan to maintain their fitness.

Key words: fitness health, nutritious food, calories, protein and carbohydrate

CONTENTS

1	INTRODUCTION.....	7
1.1.	Meal Plan	8
1.1.1	Nutrition Information	9
1.1.2	Food Database	10
1.2.	Fitness Exercise	11
1.3.	Coach	11
2	APPLICABLE TOOLS AND TECHNOLOGIES	13
2.1.	Client-Side Technology	13
2.1.1	Bootstrap framework.....	13
2.1.2	Angular JS framework	14
2.1.3	JQuery library.....	14
2.1.4	Lodash JavaScript library.....	14
2.1.5	Highcharts library.....	15
2.2.	Server-Side Technology	15
2.2.1	CakePHP framework.....	15
2.2.2	MySQL Database	16
2.2.3	Xampp.....	16
2.3.	IDE.....	16
2.3.1	Sublime 3	17
2.3.2	Atom.....	17
2.4.	Utilities.....	17
2.4.1	Postman.....	17
2.4.2	SQLyog	18
2.4.3	Github.....	18
3	APPLICATION ARCHITECTURE	19
3.1.	Stateless	20
3.2.	Cacheable.....	20
3.3.	Layered System.....	20
4	DESCRIPTION OF SYSTEM AND DATABASE	21
4.1.	Meal Plan and Recipe	23
4.2.	Fitness Plan and Communication	25
4.3.	Use Case Diagram	26
4.4.	Class Diagram.....	26

4.5. Object Diagram	29
5 USER INTERFACE (GUI) DESIGN	32
5.1. Dashboard Page	32
5.2. Nutrition Page	33
5.3. Fitness Page	36
5.4. Create Recipe Page	38
5.5. View Recipes List Page	39
5.6. Communication Page	41
5.7. Coach Page View	41
6 Discussion	44
6.1. Future Recommendation	45
References	46
APPENDICES	47
Appendix 1. GitHub front end repository code	47
Appendix 2. GitHub backend repository code	47
Appendix 3. Nutrition definition	47

List of Figures

FIGURE 1: SYSTEM OVERVIEW	8
FIGURE 2. HOW MANY CALORIES	8
FIGURE 3 : NUTRITION INFORMATION	9
FIGURE 4 : USDA DATABASE VIEW	10
FIGURE 5 : CALORIES BURNED CHART	11
FIGURE 6: SYSTEM ARCHITECTURE DIAGRAM	19
FIGURE 7 : ERP DIAGRAM	22
FIGURE 8 : ESTIMATED CALORIES REQUIREMENT	24
FIGURE 9 : USE CASE DIAGRAM FOR END USER	26
FIGURE 100: CLASS DIAGRAM FOR HEALTH APPLICATION	28
FIGURE 11: DASHBOARD PAGE	33
FIGURE 12 : NUTRITION PAGE	34
FIGURE 13 : NUTRITION POPUP INFO	35
FIGURE 14 : NUTRITION PIE CHART	35
FIGURE 15 : MEAL CALENDAR	36
FIGURE 16 : FITNESS PAGE	37
FIGURE 17 : ADD STEP	37
FIGURE 18 : ADD EXERCISE	38
FIGURE 19 : CREATE RECIPE PAGE	39

FIGURE 20 : RECIPE PAGE	40
FIGURE 21 : RECIPE POPUP WINDOW	40
FIGURE 22 : COMMUNICATION PAGE.....	41
FIGURE 23 : COACH MEMBER LIST PAGE.....	42
FIGURE 24: COACH USER DETAIL PAGE	43

ABBREVIATIONS AND TERMS

USDA	U.S. DEPARTMENT OF AGRICULTURE
RDBMS	Relational database management system
SQL	Structured Query Language
LAMP	Linux, Apache, MySQL and PHP runtime environment for Web programs
IDE	Integrated Development Environment
REST	Representational State Transfer
MVC	Model, View , Controller software architecture style
CSS	Cascading Style Sheets
HTML	Hypertext Mark-up Language
JS	JavaScript commonly used programming language for Web programming
RIA	Rich Internet Application has many of the characteristics of desktop application software
C#	C Sharp is a multi-paradigm programming language
VB.NET	Visual Basic .NET is a multi-paradigm, object-oriented programming language
PHP	Hypertext Preprocessor is a server-side scripting language
GUI	Graphical User Interface allows users to interact with electronic devices
API	Application Programming Interface is a set subroutine operation
AI	Artificial Intelligence is intelligence demonstrated by machines
SCM	Source Code Management is a version control system for tracking changes

1 INTRODUCTION

Nowadays health fitness is an evolving issue for people. People have no idea about their health maintenance. People's health, wellbeing and fitness are dependent on taking nutritious food and supplements according to their weights and ages. But they don't have the idea about their food intake and exercise. They also have no idea about the calories calculation and supplement intake. They don't know how much they should take water in a day, sleep in a day and how much their body burns calories when they do exercise, work, walk and even sleep.

Now we are developing a health fitness application for the health club through which admin can make a different recipe from food database (USDA, n.d.) and different food plans for the application user. The food plan is based on age, weight and daily personal activities. Admin can also define the different exercises in which describes the time and quantities of calories burned according to activity (Exercise, n.d.).

The nutrition health fitness system contains following items:

- **Meal Plan:** people can make their daily food chart and view these foods according to their nutritious value and log it.
- **Fitness Exercise:** people can put their exercise and daily walk steps in the system and view the result in the form of the graph (Tubbier, Pie).
- **Coach:** the coach can see the user list assigned to him, so he can view the individual progress, evaluate the user progress and communicate with him.

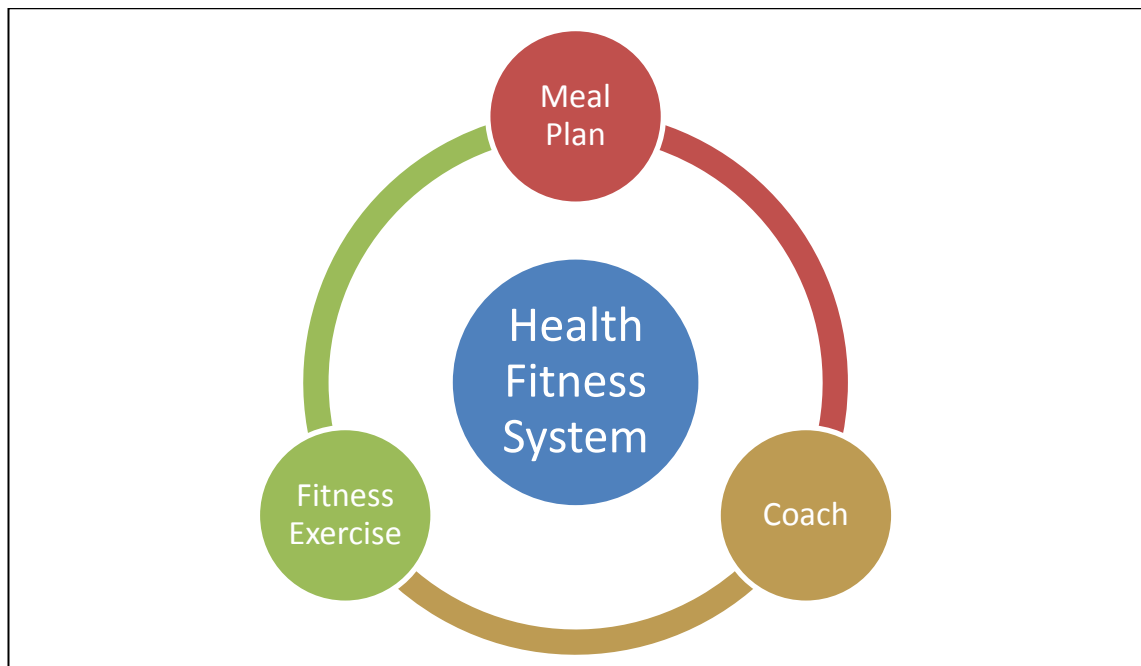


FIGURE 1: System overview

1.1. Meal Plan

Food is a basic element of our life and it's creating a great impact on our health and fitness. Most people want a healthy life and look smart but how they contain it on the constant basis, they have no idea. Many People have no idea about the food nutrition information as shown (figure 2), like how much they consume the food nutrition through their meal and how much they required etc.

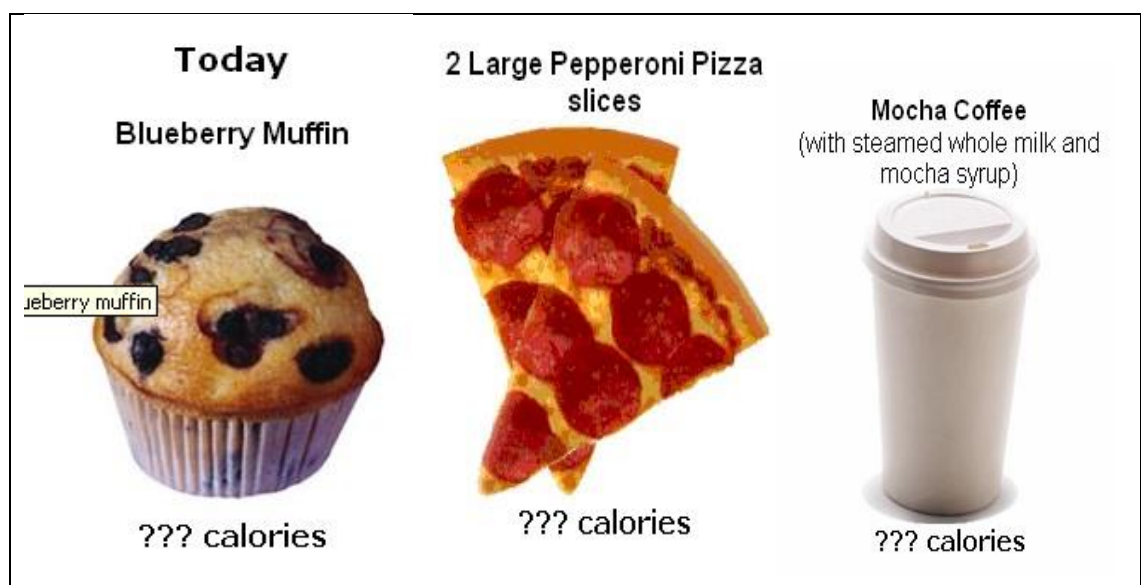


FIGURE 2. How many calories

1.1.1 Nutrition Information

First question here arises about what is the nutrition information? So as per nutrition expert the following below is the definition of nutrition information:

Nutrition information panels provide information on the average amount of energy (in kilojoules or both in kilojoules and kilocalories), protein, fat, saturated fat, carbohydrate, sugars and sodium (a component of salt) in the food, as well as any other claim that requires nutrition information. For example, if food has a 'good source of fiber' claim that the amount of fiber in the food must be shown in the nutrition information panel as shown (figure 3).

1

2

3

4

5

Nutrition Facts				
Serving Size 1 cup (228g)				
Servings Per Container 2				
Amount Per Serving				
Calories 250		Calories from Fat 110		
				% Daily Value*
Total Fat 12g		18%		
Saturated Fat 3g		15%		
Trans Fat 3g				
Cholesterol 30mg		10%		
Sodium 470mg		20%		
Total Carbohydrate 31g		10%		
Dietary Fiber 0g		0%		
Sugars 5g				
Protein 5g				
Vitamin A		4%		
Vitamin C		2%		
Calcium		20%		
Iron		4%		
* Percent Daily Values are based on a 2,000 calorie diet. Your Daily Values may be higher or lower depending on your calorie needs.				
	Calories:	2,000	2,500	
Total Fat	Less than	65g	80g	
Sat Fat	Less than	20g	25g	
Cholesterol	Less than	300mg	300mg	
Sodium	Less than	2,400mg	2,400mg	
Total Carbohydrate		300g	375g	
Dietary Fiber		25g	30g	

FIGURE 3 : Nutrition information

1.1.2 Food Database

The U.S. DEPARTMENT OF AGRICULTURE (USDA, n.d.) has a mission to support the country's nutrition protection by providing the food and nutrition help to college children and families; and help American agriculture by allotting high quality, 100% American-grown foods. So, I am getting the open source database as shown (figure 4) from United States Department of Agriculture and make my own database as per Web application requirements.

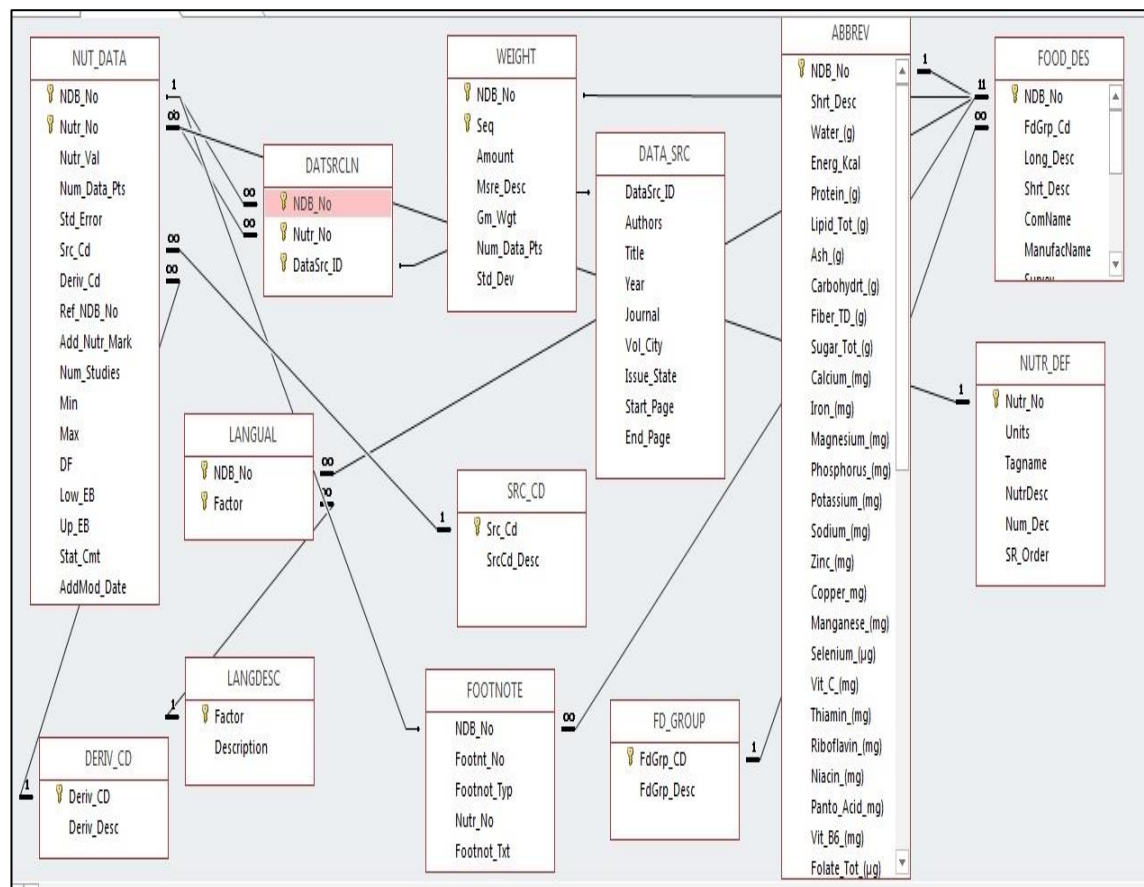


FIGURE 4 : USDA database view

The following below table is the brief overview of USDA database (USDA, n.d.) tables.

Table 1 : USDA Database Tables

Table Name	Detail
NUT_DATA	This table contains the nutrient values and information about the values, including expanded statistical information
ABBREV	This table contains the food items abbreviation

WEIGHT	This table has the food weight and measure data units i.e. tbsp, cup, slice etc.
FD_GROUP	This table contains a list of food groups and their descriptions.
NUTR_DEF	This table has the nutrient data. It offers the 3-digit nutrient code, unit of measure, INFOODS tag name, and description.

1.2. Fitness Exercise

Normally People have no idea how many calories burned when they do some exercise, so our application saves the exercises and sports activities based on calories burned per hour (energy expended) for a 130, 155, 180 or 205-kilogram persons as shown (figure 5). The number of calories consumed is influenced by many aspects, including body weight, the intensity of activity, physical conditioning level and metabolism. So, our application will also calculate the user burn calories (Exercise, n.d.).

Activity, Exercise or Sport (1 hour)	130 lb	155 lb	180 lb	205 lb
Cycling, mountain bike, bmx	502	598	695	791
Cycling, <10 mph, leisure bicycling	236	281	327	372
Cycling, >20 mph, racing	944	1126	1308	1489
Cycling, 10-11.9 mph, light	354	422	490	558
Cycling, 12-13.9 mph, moderate	472	563	654	745
Cycling, 14-15.9 mph, vigorous	590	704	817	931
Cycling, 16-19 mph, very fast, racing	708	844	981	1117
Unicycling	295	352	409	465
Stationary cycling, very light	177	211	245	279
Stationary cycling, light	325	387	449	512
Stationary cycling, moderate	413	493	572	651
Stationary cycling, vigorous	620	739	858	977
Stationary cycling, very vigorous	738	880	1022	1163
Calisthenics, vigorous, pushups, situps...	472	563	654	745
Calisthenics, light	207	246	286	326
Circuit training, minimal rest	472	563	654	745
Weight lifting, body building, vigorous	354	422	490	558
Weight lifting, light workout	177	211	245	279
Health club exercise	325	387	449	512
Stair machine	531	633	735	838
Rowing machine, light	207	246	286	326
Rowing machine, moderate	413	493	572	651
Rowing machine, vigorous	502	598	695	791
Rowing machine, very vigorous	708	844	981	1117
Ski machine	413	493	572	651
Aerobics, low impact	295	352	409	465
Aerobics, high impact	413	493	572	651

FIGURE 5 : Calories burned chart

1.3. Coach

In our application, Coach role is a very critical, a coach can see the list of users that are assigned to him, he can view the user meal plan and exercise data and also communicate on it, either user is following the plan as per instruction or not.

2 APPLICABLE TOOLS AND TECHNOLOGIES

The aim of this (thesis) project is to develop an online fitness health system. This project has been developed by the knowing technologies that are used to implement this project. So, in this section, the relevant tools and technologies that are used in the project will be explained.

2.1. Client-Side Technology

Client-side script is working in the user browser. It offers us functionality without demanding access to a Web server. JavaScript is an example of client-side programming. JavaScript is the core client-side language for the Web application. The client-side script has its own limitation on the browser and controls the data.

2.1.1 Bootstrap framework

Bootstrap is an open-source JavaScript framework established by the team at Twitter. It is an assortment of HTML, CSS, and JavaScript code. It assembles to help build user interface elements. Initially, Bootstrap was planned to manage both HTML5 and CSS3. That's why it is also called frontend framework. Bootstrap is an open-source set of tools for developing websites. It holds design templates for typography, buttons, forms, navigation and other interface components based on HTML, CSS and JavaScript.

Some reasons for programmers to prefer Bootstrap framework are listed in the following:

- Simple to get started: a person having basic knowledge of HTML and CSS may easily start using Bootstrap.
- Responsive Websites: Bootstrap can be used to build websites with responsive features which can adjust to phone, tablets and desktops screens accordingly.
- Support styling for most HTML elements (Typography, Code, Tables, Forms, Buttons, Images, Icons).
- Browser support: Bootstrap is compliant with all contemporary web browsers such as Chrome, Firefox, Internet Explorer, Safari, and Opera.

2.1.2 Angular JS framework

Angular JS is an open source frontend Web application framework. Angular JS is a structural framework described by a Google for dynamic Web applications development. It allows you to use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and briefly. Angular JS data binding and dependency injection minimize your code size and effort which you did before. And these all things make it an ideal partner with any server technology.

Below are some critical points to be consider while using AngularJS:

- Angular JS is a powerful JavaScript based development framework to create rich internet application (RIA).
- Angular JS provides developers options to write the client-side application (using JavaScript) in a clean MVC way.
- Applications written in Angular JS are cross-browser compliant. Angular JS automatically handles JavaScript code suitable for each browser.

2.1.3 JQuery library

JQuery library is a small, robust JavaScript library which contains enriched multiple features. It contains easy-to-use API which is compatible with multiple browsers. HTML document manipulation, traversal, animation, event handling and ajax call (way of communicating between frontend and backend server) is quite simple with that API. The combination of extensibility and versatility has changed the way people implement the JavaScript functionality.

2.1.4 Lodash JavaScript library

Lodash is a JavaScript library that supports programmers write more concise and easier way to maintain JavaScript code.

It can be broken down into several main areas:

- Utilities - for making simpler common programming job such as determining type as well as simplifying math operations.
- Function - simplifying constraining, binding, controlling, decorating, currying, and changing the pointer.
- String - alteration functions for performing basic string operations, such as converting to uppercase, camel case, trimming, etc.
- Array – for array related operations (creating, finding, splitting, merging, modifying, and compressing)
- Collection – for collection related operations (iterating, filtering, sorting, splitting, and building)
- Object – for object related operations (accessing, merging, defaults, extending, and transforming)

2.1.5 Highcharts library

Highcharts is a chart/graph library and this library is written in JavaScript, offers ease in adding interactive charts to your website or Web application. High stock JS and High maps JS is derived from High chart as it is mature, modular codebase and fast.

2.2. Server-Side Technology

Server-side scripts are run on the server so it needs a server. This reduces the number of bugs or compatibility issues since the code is run on one server using one language and hosting software. Server-side programming can also be coded when users send form variables, protecting users against any hack attempts. Java, C#, VB.NET, Node.js, and PHP are the examples of server-side language.

2.2.1 CakePHP framework

CakePHP is a framework, for development in PHP. It's free and opensource. For web application development it provides initial structure. It has MVC (model-view-controller) concept. Development with CakePHP requires less code and development is faster with it, which is very helpful for a non-technical person.

The following below some CakePHP benefits:

- Model view controller
- Flexible and quick start
- High security
- Easy to upgrade

2.2.2 MySql Database

MySQL is an open-source and free to use database. It has effective management for multiple databases by providing the functionality to connect with software.

It's reliable and stable with following benefits:

- On demand scalability
- Data security
- Very well memory management system
- Control of complete workflow
- Provides multiple development interfaces (JDBC, ODBC)

2.2.3 Xampp

XAMMP is open source web server developed by Apache. It's a multiplatform server, also contains MariaDB database, Apache HTTP server, MySQL, Perl and PHP.

It has following benefits:

- Easy to install
- Ability to run on multiple platforms
- Also contains other modules like, Joomla, WordPress, OpenSSL
- Start and stop server with single command.
- Strict security settings by default

2.3. IDE

An IDE is stand-alone application to facilitate software developers to develop applications quickly and easily through multiple functionalities such as code completion, code editor, indentation, debugging, testing-facilities, build tools (Maven, Gradle) etc.

2.3.1 Sublime 3

Sublime Text 3 is one of the popular code-editor which is used widely by developers.

Some of the main features offered by Sublime Text 3 are:

- Files, Text or lines can be navigated quickly
- Cross-platform compatibility
- Auto-Completion
- Syntax-correction/highlighting

2.3.2 Atom

Atom is an open-source editor which is compatible with all famous operating systems such as Mac, Windows and Linux. It provides support for integrating with GitHub via plug-ins. It also provides common features offered by any other editor in the market. It has been developed using CoffeeScript and Less.

2.4. Utilities

Sometimes developers use software's for checking the quick result or mock-up i.e. view the REST API response, check the database queries before writing in the actual code etc.

2.4.1 Postman

Postman is a strongly recommended tool for testing server-side Web services. Postman's friendly GUI makes it easy to test, develop and document APIs by allowing users to quickly put together both simple and complex HTTP requests.

2.4.2 SQLyog

SQLyog is a GUI tool for the RDBMS MySQL database server. SQLyog has a visual interface where tables and table structures can be defined, displayed and manipulated. SQLyog also provides the graphical way to perform common operations i.e. check the queries result, query errors, relational graphical diagram, import and export the database tables.

2.4.3 Github

GitHub is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all the distributed version control and SCM functionalities of Git as well as adding its own features. GitHub provides access control and several notable features including but not limited to bug-tracking, feature requests, task management etc.

3 APPLICATION ARCHITECTURE

In this project, we are using the client-server approach which means that client application and server application must be implemented separately without any dependencies on each other, client-side developer just needs to know the REST API URL's with parameters; this is the normal practice in Web development, so nothing fancy is required from client side.

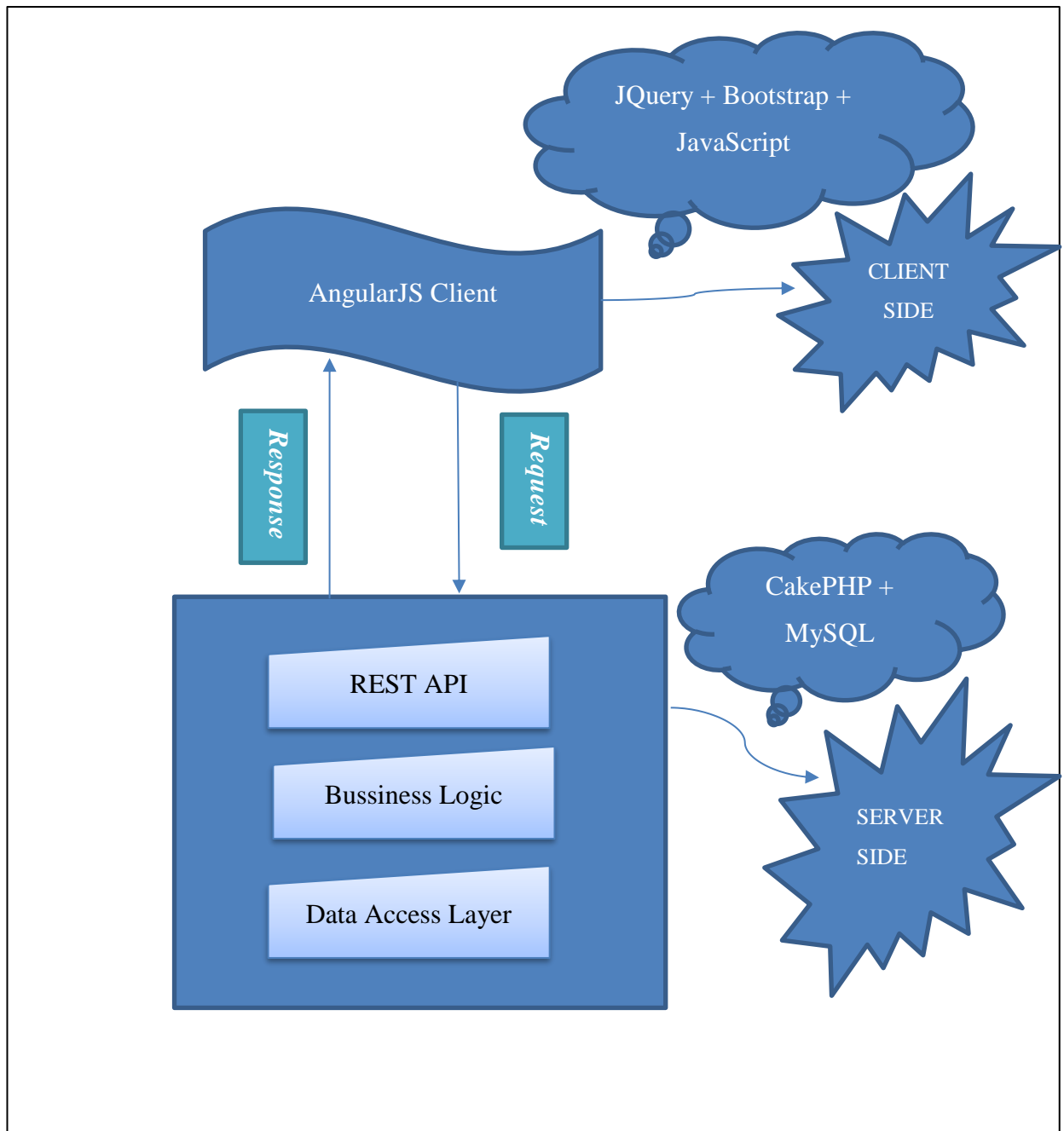


FIGURE 6: System architecture diagram

3.1. Stateless

In the client-server application, the interaction is stateless. The server will not store anything about the HTTP request which client makes. It will acknowledge each request as a new request, not use session and history. If client application needs to be a stateful application (where the user makes a login and do other authorized operations) for the end user, then each request from the client side should have all the necessary information to service the request. Normally developer uses browser local storage or session storage for saving the info.

3.2. Cacheable

In today's world, Web performance optimization is a key factor. Web application development team can improve the site performance and its loading, testing scores by selecting from a variety of client-side caching techniques. The effective client-side caching strategy can reduce load times by several factors. HTML5 has an application caching mechanism that allows to run applications in offline mode. Application Cache interface of the browser can be used to save resources in its local storage which then later can be made available to offline users.

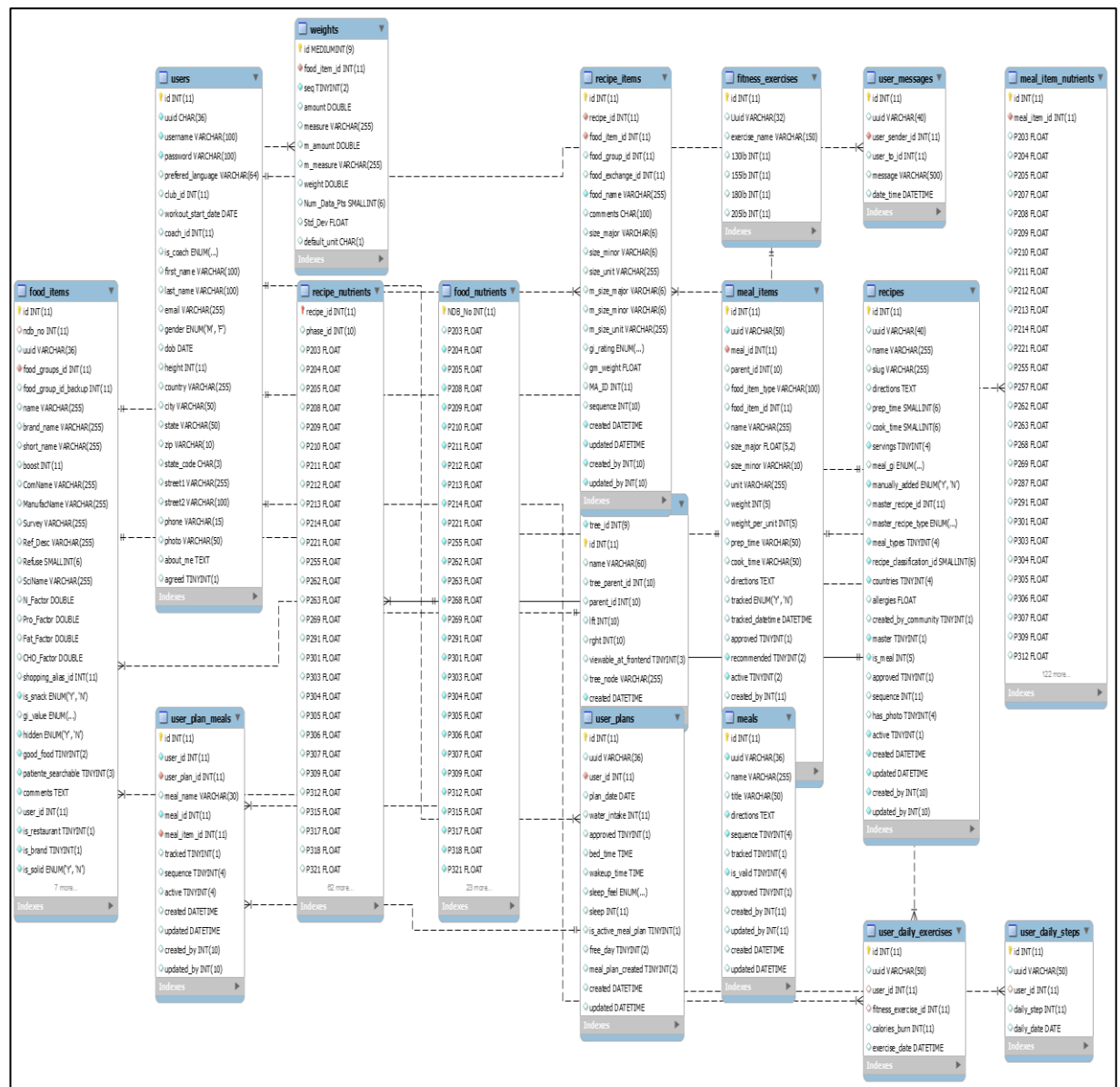
3.3. Layered System

REST API allows you to use a layered system architecture where you deploy the APIs on server A, and store data on server B and authenticate requests in Server C, for example. There is no way for a client to tell whether it has established connection with the end server or it is connected to the intermediate server.

4 DESCRIPTION OF SYSTEM AND DATABASE

Health Fitness System database is made of three components: nutrition, fitness and coach communication. In meal module we handle food related activities; in fitness module, we handle the physical activities, exercise and daily steps. In communication module, the user can send messages to the coach about his suggestion.

In the context of this database, term component refers to logical isolation of tables and does not include any physical isolation schemes. All tables are in same database and instance. In this project relational MySQL database has been used. Tables have been defined using multiple MySQL constraints (primary key, foreign key). Using those constrain CRUD (create, read, update, delete) operations perform easily. The following below is ERP diagram (figure 8) which is explaining the logical view of our application database.



	food group id, ndb_no (nutrition database number) etc.
Food_nutrients	This table contains the nutrition's information as per food item protein, fat, carbohydrate, vitamin etc.
Meal_item_nutrients	This table defines nutrition's information as per meal item (meal is also being the combination of food items) protein, fat, carbohydrate, vitamin etc.
Meal_items	This table defines the food items as per meal, meal-id is using as a foreign key, also defined the parent-key if some food item has multiple foods or combination of food's item.
Meals	This table defines the meal category like breakfast, am-snack, lunch etc.
Recipe_items	This table defines the food items of the recipe with quantity and unit as per food item, food id and recipe id is using as a foreign key.
Recipe_nutrients	This table contains the recipe nutrition information, it is the sum of foods nutrition information that is included in a recipe
Recipes	This table defines the recipe attributes like recipe name, preparation name, slug, direction/instruction preparation time, cooking time, serving people etc.
User_daily_exercises	This table contains the user calories burn information as per exercise.
User_daily_steps	This table contains the number of daily steps and also the user calories burn information in the result of daily steps.
User_messages	This table contains the messages between coach and user for communication.
User_plan_meals	This table contains the meal (breakfast, lunch, dinner) as per date and track if the user ate the meal or skipped the meal.
User_plans	This table defines the food plan of the day.
Users	This table contains the user related information like name, email, address, date of birth, height etc.

4.1. Meal Plan and Recipe

When the user registers/enrolls in the system the meal plan will be generated on daily basis. The following below is the estimated amounts of calories (figure 8) chart which a person required to maintain energy balance (WebMD LLC., n.d.). The estimated calories depend on gender, age group and physical activity (nature of work/professional).

Estimated Calorie Requirements (in Kilocalories) for Each Gender and Age Group at Three Levels of Physical Activity^a				
Gender	Age (years)	Activity Level^{b,c,d}		
		Sedentary^a	Moderately Active^c	Active^d
Child	2–3	1,000	1,000–1,400 ^e	1,000–1,400 ^e
	4–8	1,200	1,400–1,600	1,400–1,800
	9–13	1,600	1,600–2,000	1,800–2,200
	14–18	1,800	2,000	2,400
	19–30	2,000	2,000–2,200	2,400
	31–50	1,800	2,000	2,200
	51+	1,600	1,800	2,000–2,200
Male	4–8	1,400	1,400–1,600	1,600–2,000
	9–13	1,800	1,800–2,200	2,000–2,600
	14–18	2,200	2,400–2,800	2,800–3,200
	19–30	2,400	2,600–2,800	3,000
	31–50	2,200	2,400–2,600	2,800–3,000
	51+	2,000	2,200–2,400	2,400–2,800

Source: HHS/USDA Dietary Guidelines for Americans, 2005

FIGURE 8 : Estimated calories requirement

Normally we can break down the meal plan with the following numbers of calories as per day (Maya E Nahra, n.d.). So, after calculating the total number of calories of the day, we divide the whole day calories in different meals and according to each meal calories, the system will get the food item from the database.

TABLE 3: Breakdown of Calories of a Day

Calories	1200	1500	1800	2100	2400	2700	3000
Breakfast	300	400	500	500	600	600	700
AM Snack	150	150	150	300	300	450	450
Lunch	300	400	500	500	600	600	700
PM Snack	150	150	150	300	300	450	450
Dinner	300	400	500	500	600	600	700

After defining the meal calories system will create a meal plan for a current day or that particular day and each meal contains the required food or recipe. The food item has the nutrition information which system get from the USDA database, each item also has

food units and food group id. So, in the meal, the system will handle the number of drink water glasses, the number of sleep hours, take a supplement status and food-related operations. In the system, user can make the recipe, in the recipe user define recipe name, its preparation time, number of serving people, food ingredients, cooking instructions and the sum of recipe foods nutrition information based on USDA food database. Later user can use the recipe as a food meal item.

4.2. Fitness Plan and Communication

In Fitness Plan we are calculating burn calories as the result of exercise which user will do, for this purpose we are using www.nutristrategy.com website page (Exercise, n.d.) as a reference. We get all exercise information from this page and save in our database. Normally people have no idea how many calories burned when they do some exercise, so we calculate the burn calories, based on time (per hour) and weight (for a 130, 155, 180 or 205-pound person figure 5). The number of calories consumed is influenced by many aspects, including body weight, the intensity of activity, conditioning level and metabolism.

User can also calculate the calories burned by the daily number of steps. There's a common walking challenge called "10000 steps" – simple idea, take 10000 steps a day. As you now know 2000 steps are roughly a mile, which means the 10000-step challenge means you to walk 5 miles. (Bumgardner, 2018)

TABLE 4: Daily Step Calories Burn

Weight (lbs)	100	120	140	160	180	200	220	250	275	300
Mile 1	57	68	80	91	102	114	125	142	156	171
Mile 2	114	136	159	182	205	227	250	284	313	341
Mile 3	170	205	239	273	307	341	375	426	469	512
Mile 4	227	273	318	364	409	454	500	568	625	682
Mile 5	284	341	398	455	512	568	625	710	782	853

In communication user can send the messages to the coach and coach will also be able to response her/his messages.

4.3. Use Case Diagram

The following diagram (figure 9) is the use case diagram of the system. It's the over-view of user actions, which user can perform in the application. User can add/update his profile information like contact information, date of birth, weight and height etc. The user can also add, update and delete the food meal items. User can see the food nutrition information in the popup window and view the nutrition information in the pie chart for a better understanding of data. A user can add/update his daily fitness exercises and view the number of calories burned in these exercises. The user can add the number of daily steps as an activity in the system. The user can view the recipes list and its detail information like recipe name, instructions, preparation time, number of people, ingredient and nutrition information etc. User can also make communication with the coach through message chat.

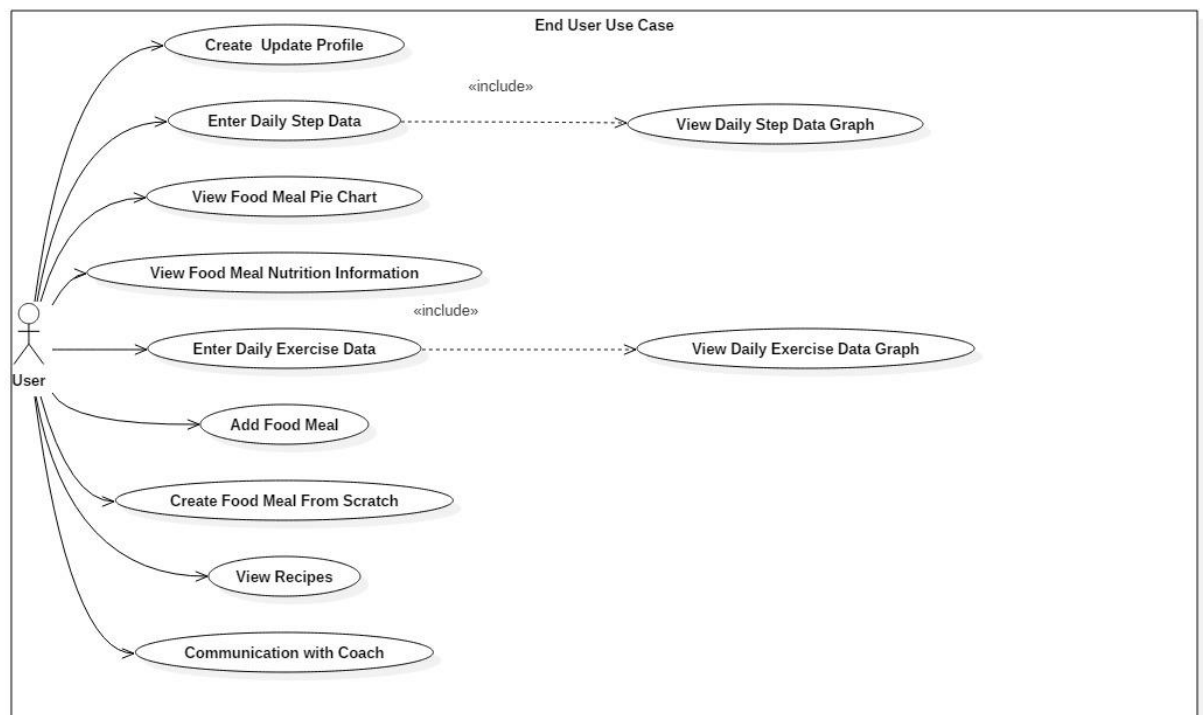


FIGURE 9 : Use case diagram for end user

4.4. Class Diagram

The class diagram represents the static view of an application. The class diagram is the only diagram which can be directly mapped by object-oriented languages and that's why it is widely used at the time of creation. It shows the attributes name, operations and relationships of the classes. Figure 10 is the class diagram of the health fitness system.

The user class inherits from the human class. User class contains the multiple parameters (id, club id, username, user type, email, password etc.) that are related to the registered user on the system. The user class also contains the different methods which perform user-related operations. The club class is associated with the user class, which is showing the user relationship with the club. Class userplan has a user id, the system generates a plan for each day to the user in which user saves the intake glasses of water and number of sleep hours. Later user plan id is used in user meal plan class as a reference user plan id.

The user daily exercise class has the user id and exercise id with the burn calories attributes. The user daily exercise class has several methods which calculate the calories burned based on exercise and time. Exercise class has the circle icon which is representing some physical instrument for exercise if need. User daily step class has the attributes and methods which handle the user daily steps operation.

User message class has the attributes and methods which handle message archive between user and coach. The user meal plan class has several methods and attributes for the user meal plan i.e. user id, meal name (breakfast, lunch, dinner), skip (is the user skip meal or not) and food item id. In the user meal plan class, the food item id shows the food consumed by the user in meal section.

The recipe class is showing recipe related attributes and methods, it has the recipe name, preparation time, number of serving people. The recipe food items class has a food items attributes those are using a recipe. Food item class is associated with the food group to describe its group i.e. dairy, baby foods, vegetables. The ndb_no is a food item id in food items class, FoodNutrients class has the ndb_no as a reference id, the food nutrients class has the parameters to describe the nutrients' information (protein, vitamin, fat, water, carbohydrate, energy, iron etc.) of the food item to describe the food.

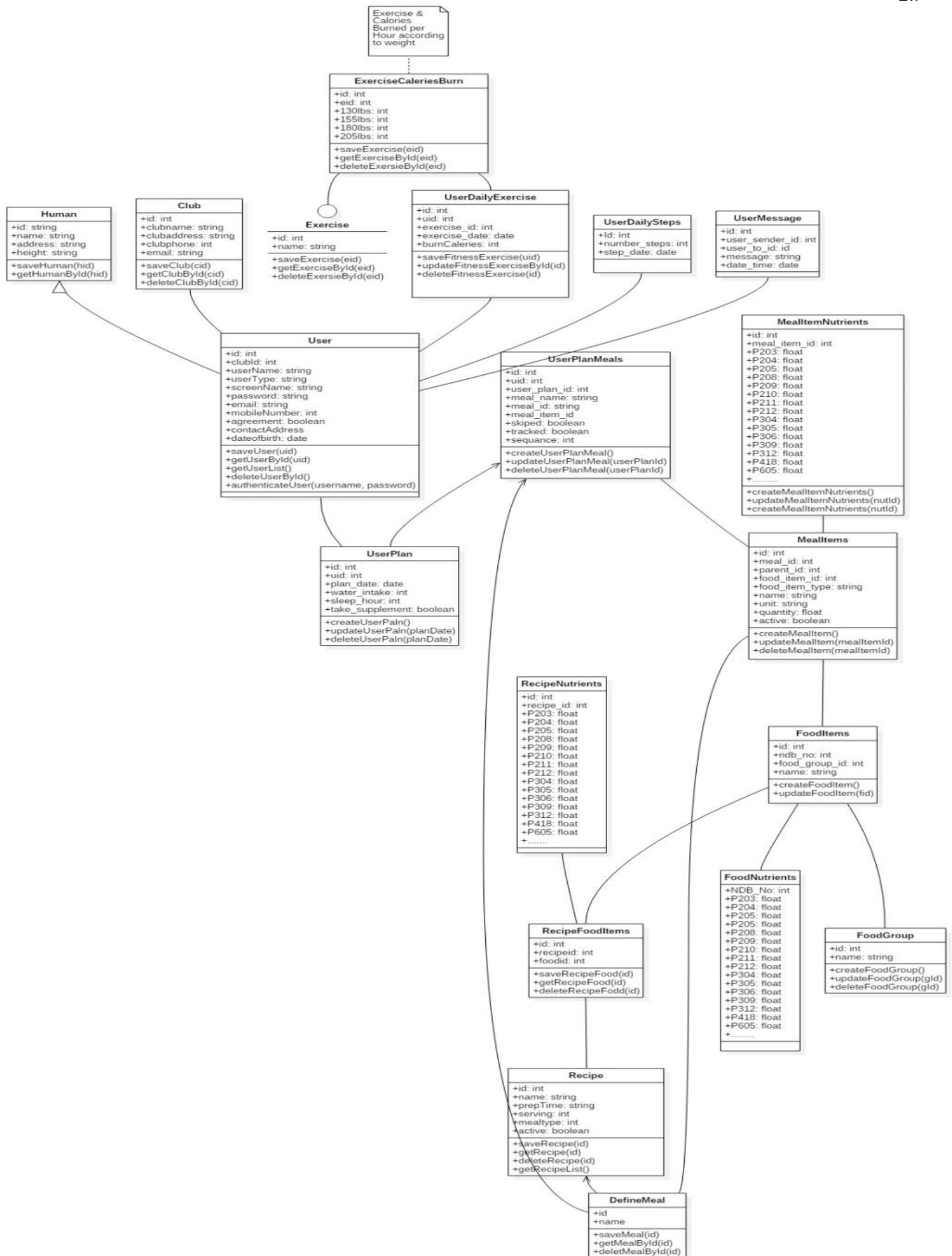


FIGURE 100: Class diagram for health application

4.5. Object Diagram

Object diagrams can be imagined as the snapshot of a running system at a particular moment. Figure 11 describes the state of the user on the server-side (CakePHP framework models instance) when user has registered in the system and doing his normal operations of fitness exercise, food meal and coach communication.

In Figure 11, user instance has the user related information, the username is 'Tom' user id is '55', the user date of birth is '17/09/1982'. User fitness club id is 1 which is a club reference id. Club instance has club information like club id, club name and others contact info. User daily step instance is showing user daily walked steps on that particular day. User daily exercise instance is showing how much user spent time on the exercise hours and in the result of this how many calories burned. The exercise calories burn instance has the burn calories information on the basis of weight and exercise as per hour.

In object diagram, we can refer the physical objects for exercise as showing the following 'boxing bag' is necessary for 'boxing' exercise. User instance plan show that everyday system generates the plan in which system save the user id and plan date, it also sketches quantity of glass waters and number of sleep hours of the user. The instance user plan meal is showing what user has eaten in the meal and meal item is showing nutrient information of food for example the following diagram user had eaten 'Turkey Cheese Pizza' in breakfast and the meal item nutrients instance has its nutrient information (protein 0.79, fat 0.27, carbohydrate 54.74, energy 208.74).

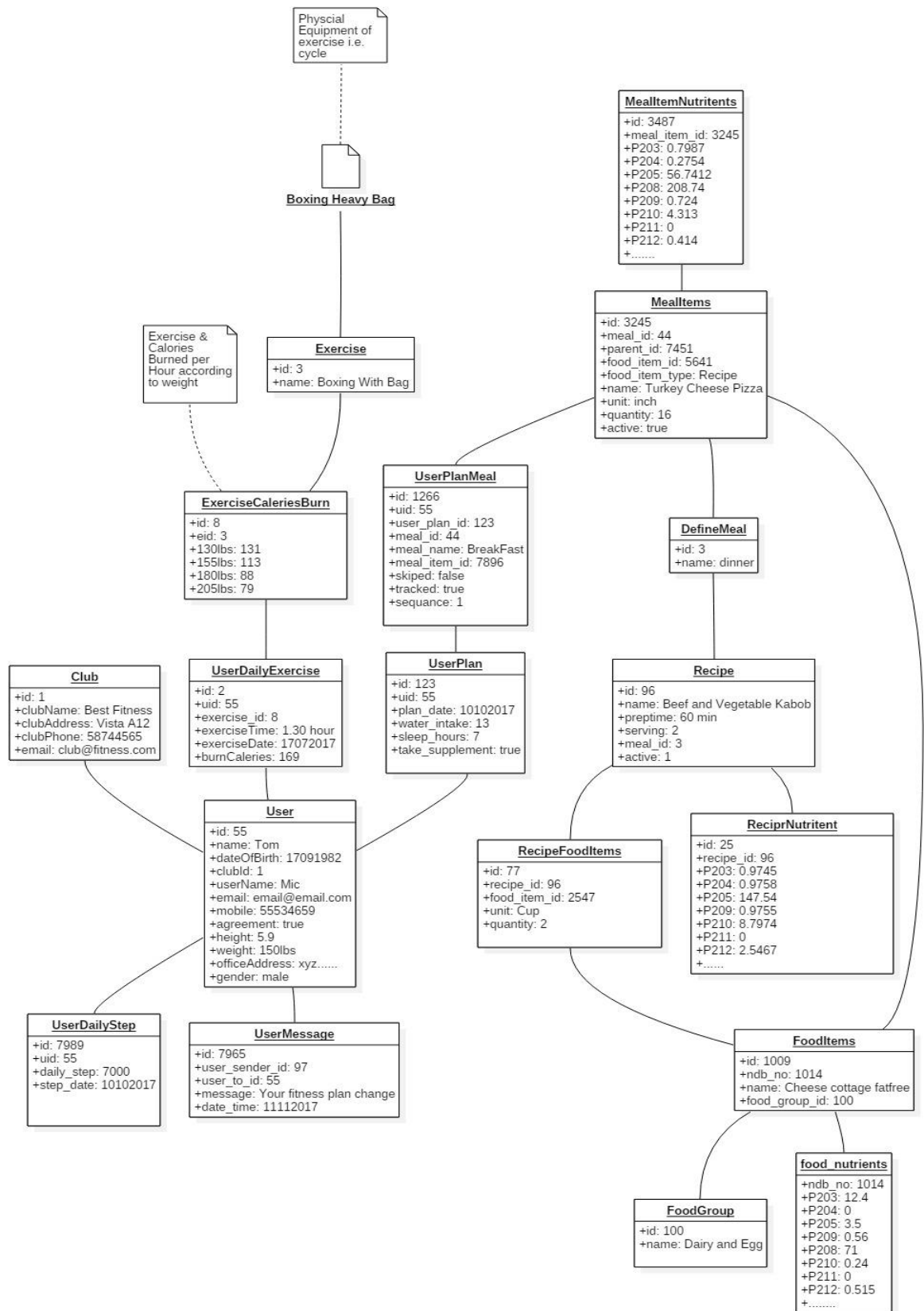


FIGURE 11: Instance/Object diagram for health application

The following below is the nutrition code definition of USDA data which we are using in our application (Appendix 1: Nutrition definition) NUT_NO column defines the nutrition number basically it's the id for nutrition name, units column is describing the measurement of nutrient, tag name column is showing the scientific name of nutrient and nutrdesc column is showing the brief description of the nutrient.

TABLE 5 : Nutrition definition

Nutr_No	Units	Tagname	NutrDesc
203	G	PROCNT	Protein
204	G	FAT	Total lipid (fat)
205	G	CHOCDF	Carbohydrate, by difference
207	G	ASH	Ash
208	Kcal	ENERC_KCAL	Energy

Note: Rest of the table append in appendix.

5 USER INTERFACE (GUI) DESIGN

The system has been divided into three main interfaces: but here we are discussing user interface. The user will be able to generate the meal plan, add and update food items, add the exercise data, communication with the coach and view the recipe information and show some fitness related external articles.

5.1. Dashboard Page

The dashboard page (figure 11) is the main page that will be accessed when a user successfully logins. It shows the user basic profile information as name, email, address, gender, date of birth, mobile, Facebook and Twitter links. After this, there are three sub-sections. In fitness section, a user can see the next 7 days plan for exercise and daily walk steps. In meal plan, the system is showing a recipe as a featured recipe, then in recommended meals and snacks section, the system is showing user current day foods of meals i.e breakfast, am snack, lunch, pm snack and dinner. At the end of meal section, there is water tracker in which user can update the status of the number of water glasses. In coaching section, the user can see the latest message from the coach about his performance or suggestions. After this user can see the fitness related external articles which help the user in her/his fitness and meal plan.

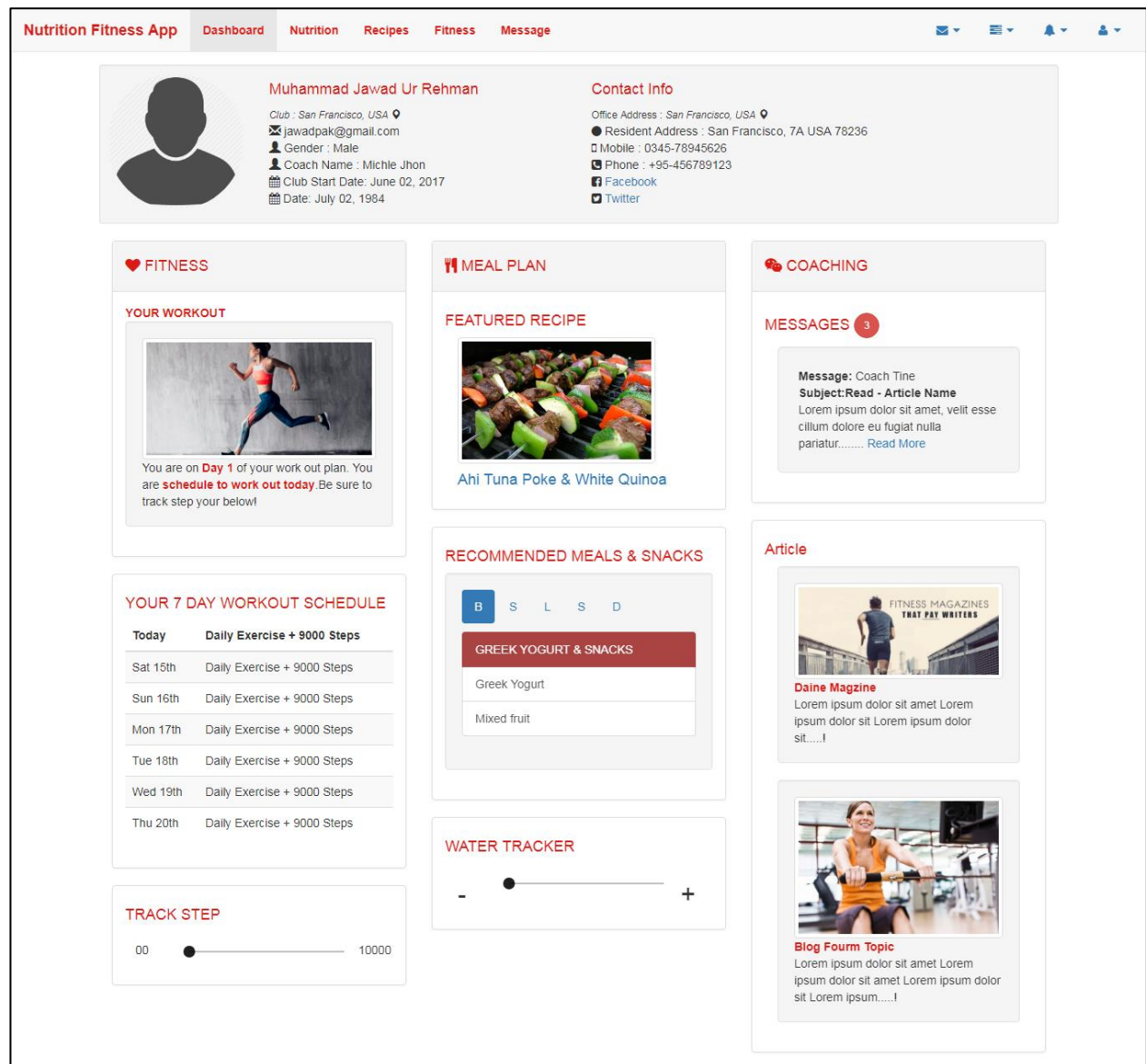


FIGURE 11: Dashboard page

5.2. Nutrition Page

The following nutrition page (figure 12) is showing the food meal information of the user, the user can save a daily number of glasses water, the number of sleep hours and supplement status whether he/she has taken the supplement or not. The column chart shows the number of calories consumed by the user in a day. The user can also view the food items in the meals i.e. in breakfast user can eat apple dried, the quantity is 1 and unit of the item is cup. The user can click on heart icon, to make a food a favourite. When user clicks on nutrition icon a popup is open and show nutrition information as shown (figure 13). The user can also see the nutrition information in the pie chart (figure 14). In addition, a food item can also be dealt from the meal. Eat status can be

saved by clicking on “I ate” button. From skip button, the user can skip the meal, If user need to add the extra food click on “Add Food” button. The user can also create the meal food from scratch through “Create From Scratch” button. The user can view the particular day food meal plan through date calendar input (figure 15). When the user selects a date in calendar system will call the rest API and load particular date related data. The user can see the recipe in reddish highlights background because the recipe has multiple food items and the user can also see each food nutrition information as well as recipe nutrition information.

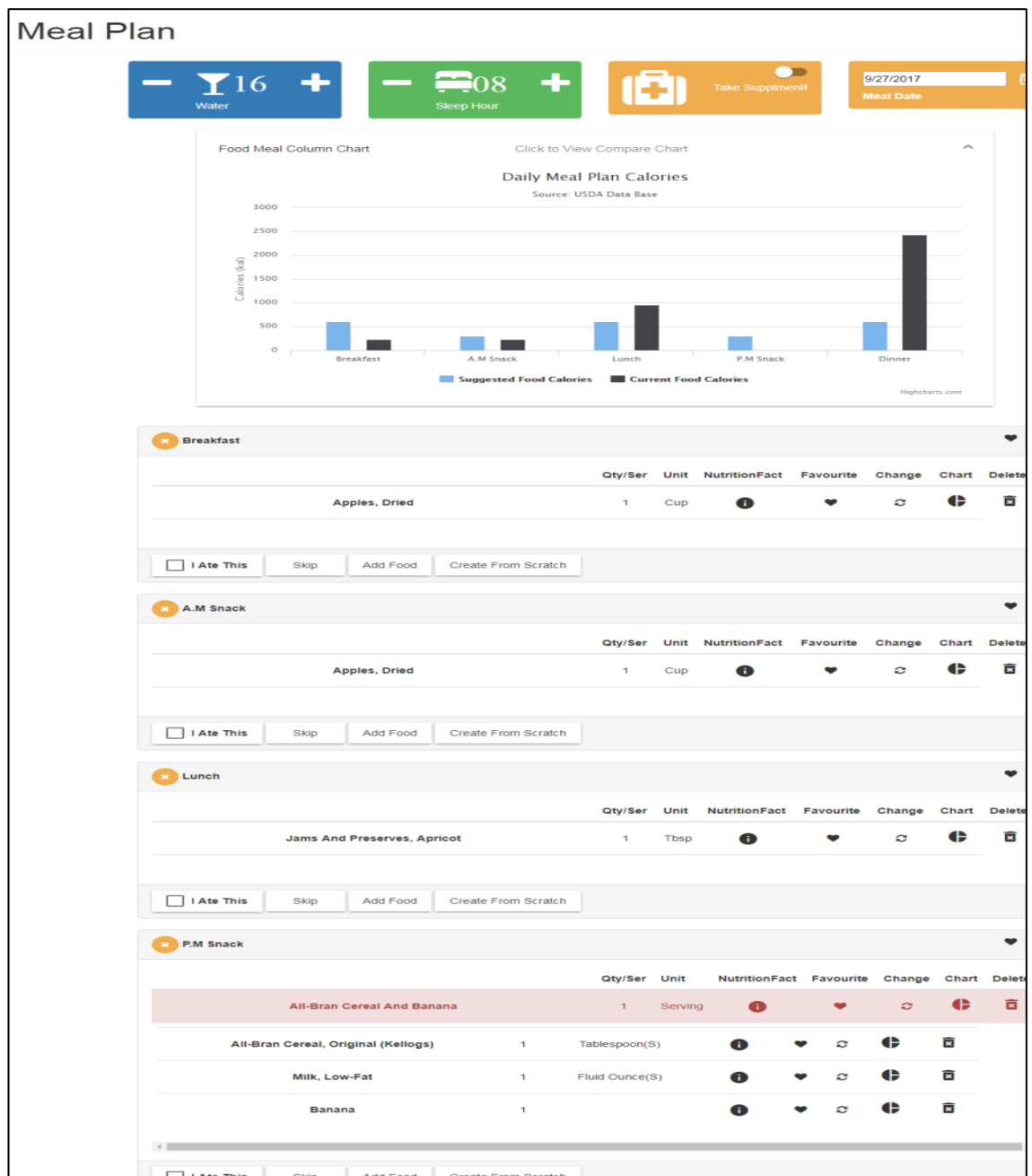


FIGURE 12 : Nutrition page

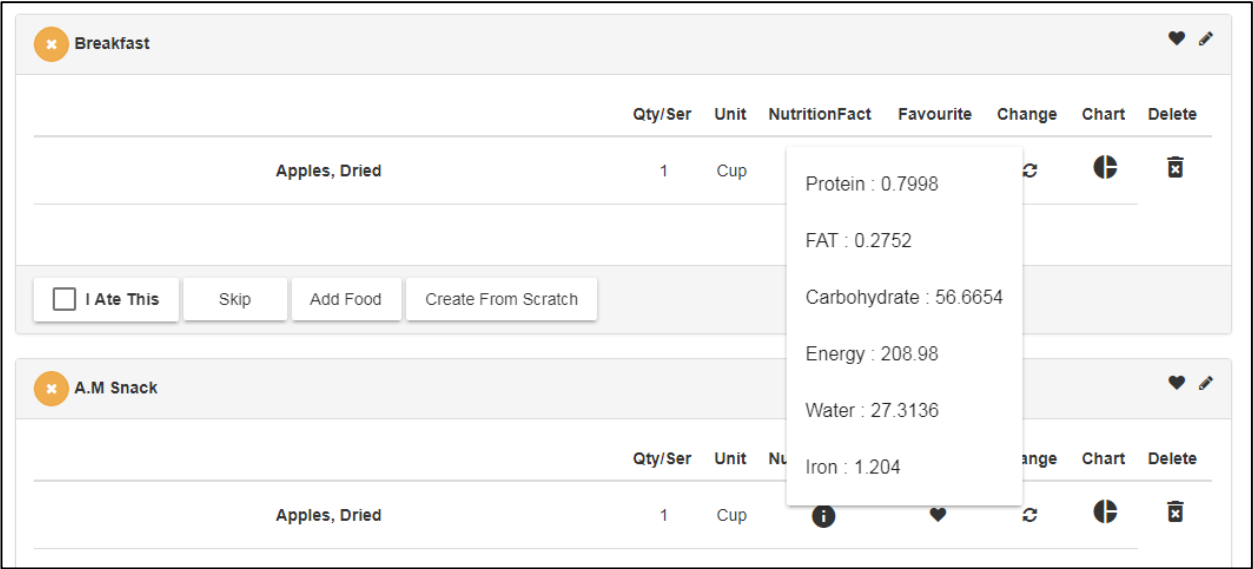


FIGURE 13 : Nutrition popup info

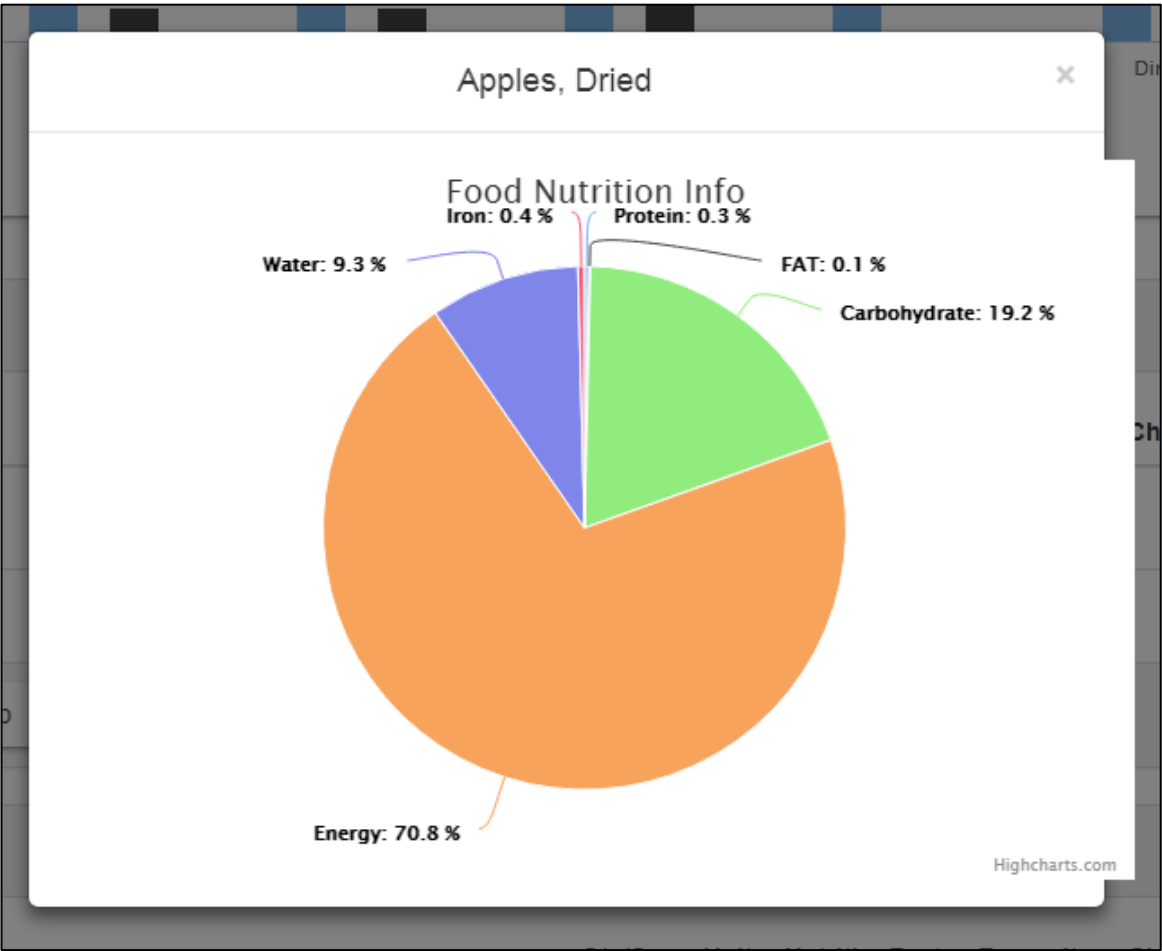


Figure 14 : Nutrition pie chart

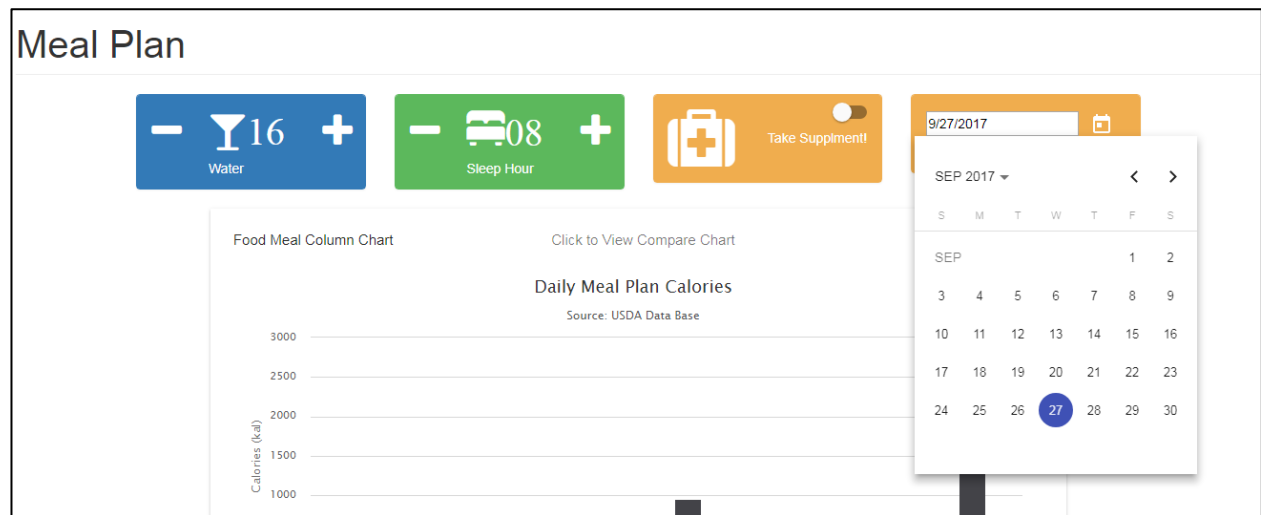


FIGURE 15 : Meal calendar

5.3. Fitness Page

The fitness page (figure 16) is showing fitness related information. There are two graphs, one is the daily step graph which shows the calories burn information of last 7 days walk. In daily step graph, the x-axis and y-axis indicate the date and number of steps values respectively. Another one is the daily exercise graph which shows the exercises data, meaning how many calories burned in the result of exercises on that particular day. In daily exercise graph, the x-axis and y-axis indicate the exercise name and number of calories burn values respectively. The user can enter the daily number of steps (figure 17) as well as the exercise data (figure 18) through popup windows.

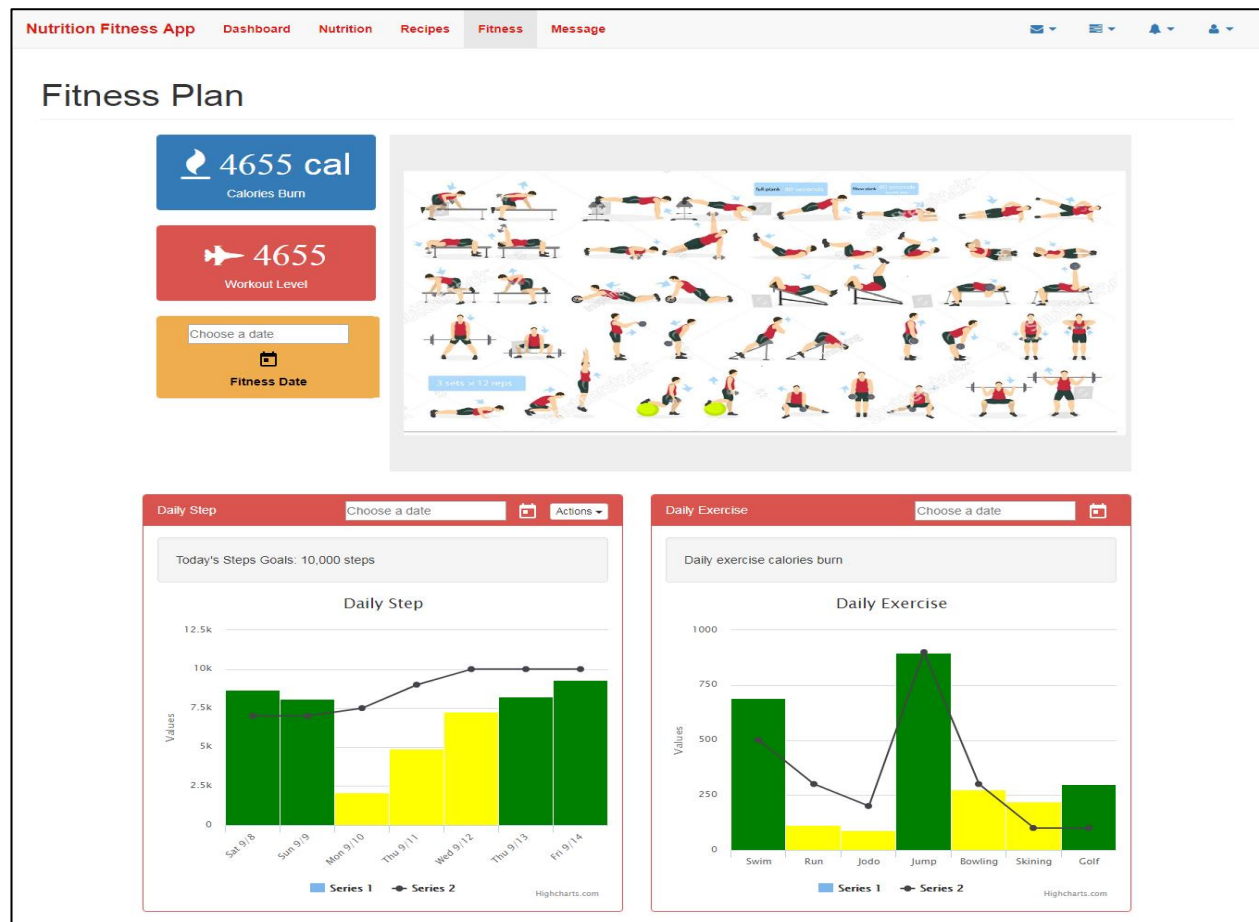


FIGURE 16 : Fitness page

Add Step
×

Daily Step's Step Date :

Close

Save changes

FIGURE 17 : Add step

The image shows a web application dialog titled "Add Exercise". It features a date input field set to "10/10/2017", an exercise name dropdown menu currently showing "Cycling, mountain bike, bmx", and an exercise time input field set to "25". The dialog includes "Close" and "Save changes" buttons at the bottom right.

FIGURE 18 : Add exercise

5.4. Create Recipe Page

The user can make a recipe from the create recipe page as shown figure 19. In the title text field, the user can write the recipe name. User can select the different values from the dropdowns: either it's a vegetarian or not, select the meal type (breakfast, lunch, dinner) for recipe, select the type of cuisine (Indian, Italian, Mexican, Turkish, Spanish) for recipe, define the food category or food group (soups and sauces, spices and herbs, seafood, beverages, baked products, sweet etc.).

After these dropdown selections, the user will type the food name under the item text field, an autocomplete popup is open, the user selects the value from the popup. On the value selection, the appropriate units of the selected food item are filled in measurement dropdown. The user enters the quantity and selects measurement value from the dropdown the calories, protein, carbs and fat value calculated implicitly. When the user clicks on 'plus' icon the food item row appears for adding a new food item, in the same way when the user clicks 'minus' icon the food item. The serve text field is showing the number of people for the recipe. In prep time user defines the preparation time for the recipe. In cook time user define the number of minutes for cooking recipe.

The user can also define the recipe instruction in the text area. The user can also see the ratio of protein, carbs and fat in progress bars, the recommendation of protein ratio should be 35-45 %, the carb ratio should be 30-40% and the fat ratio should be 25-35%.

Nutrition Fitness App Dashboard Nutrition **Recipes** Fitness Message

View Recipes
Create Recipe

Add Recipe

Title: Grandma's Recipe - Pork steak with Apple

Vegetarian?: No Meal: Dinner Cuisine: Asian Category: Meats and Poultry

Quantity	Measurement	Item	Preparation (Optional)	Cal	Protein	Carbs	Fat	Add/Delete
2	cup(s)	Pork tenderloin	Seasoned	275	15	25	10	+ -
1	cup(s)	Vegetable Oil		50	0	25	20	+ -
1/4	cup(s)	Dijon Mustard		25	0	10	1	+ -
1	cup(s)	On						+ -
		Shiitake Mushroom (Fresh)		350	15	60	31	
		Shiitake Mushroom (Dried)						
		Worcestershire Sauce						
		Siraz						

Serve: 350 Prep Time: 15 Cook Time: 60

Recipe Instruction (Press enter for each step)

Whisk flour, flaxseed meal, baking powder, baking soda and salt in a bowl. In another bowl, beat butter and stevia with a hand mixer until light and fluffy; add egg whites and vanilla, and beat until just combined. Gradually stir flour mixture into butter mixture until combined. Form dough into a log. Wrap tightly in a waxed paper and refrigerate for at least 3 hours. Cut out cookies using a 2-inch cutter and bake until firm, for about 6-8 minutes.

Recipe Nutrition Info

- Protein: 40% Complete
- Carbs: 30% Complete
- Fat: 30% Complete

*Note : Recommended Nutrition Ratio (Protein 35-45%, Carbs 30-40%, Fat 25-35%)

Success

FIGURE 19 : Create recipe page

5.5. View Recipes List Page

The user can see the recipes list (figure 20) on recipe page which are approved and recommended from the admin side, the user can see the recipes based on different filters and also provide the feedback on the recipe and make it a favourite recipe. When the user clicks on a recipe, a popup window will open (figure 21) and show recipe related information. Total cooking time is showing recipe cook time, serving people is showing how many numbers of people eat this recipe, the ingredient list is showing the name of ingredients which are used in the recipe. In the instruction area, the user can see the step by step info for making recipe. In nutrient fact, the user can see the sum of food nutrition information that the recipe contains.

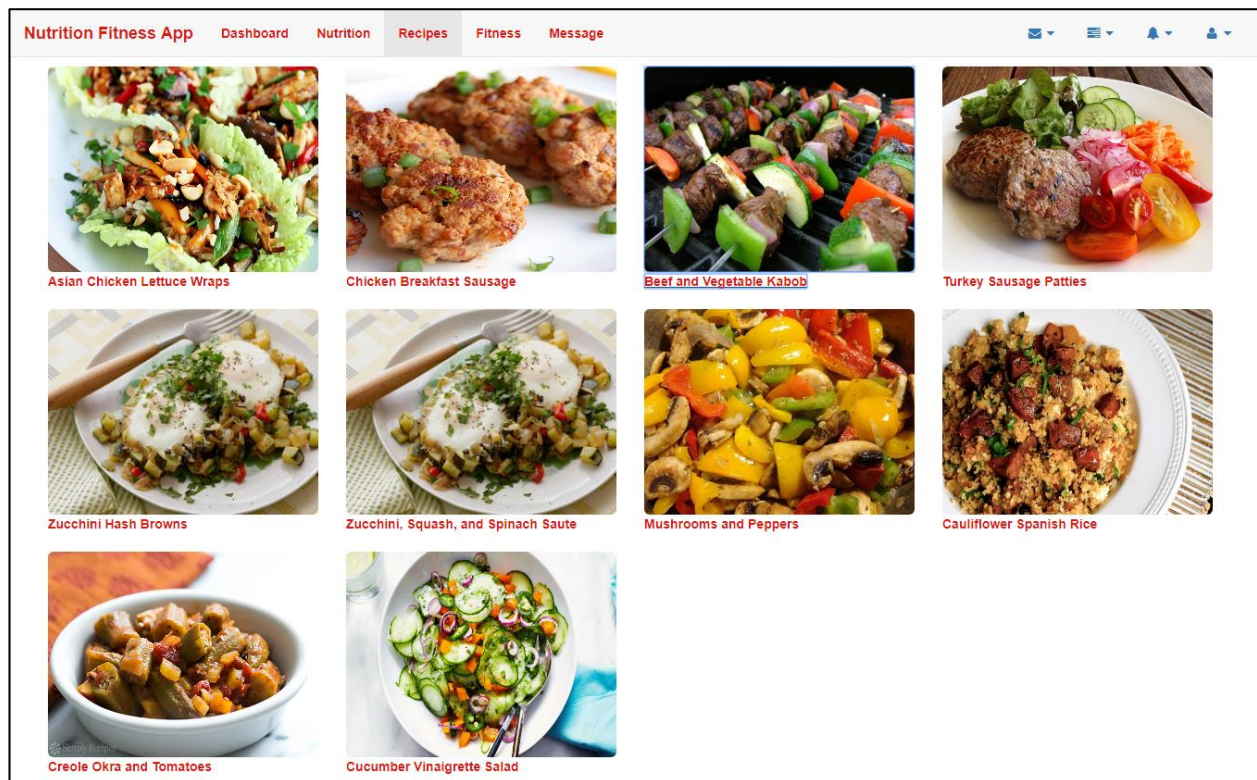


FIGURE 20 : Recipe page

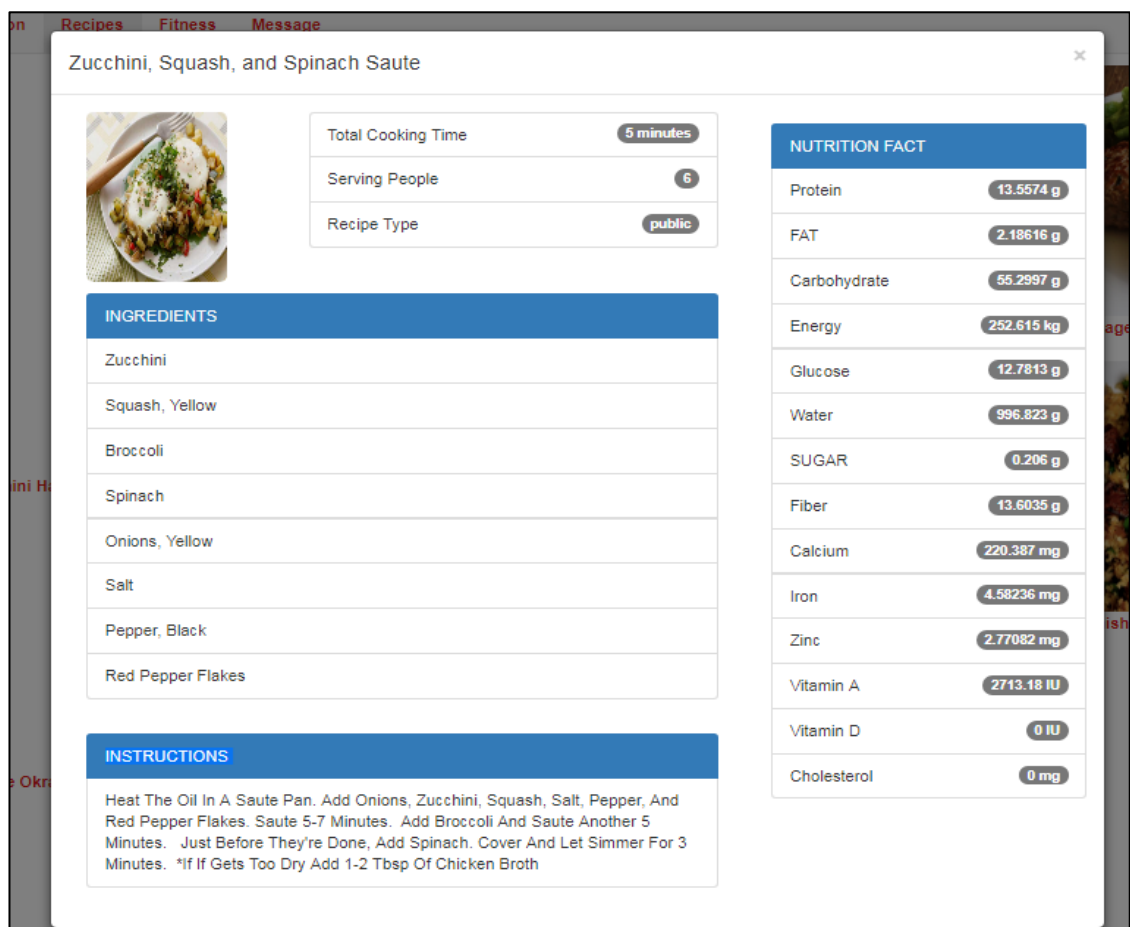


FIGURE 21 : Recipe popup window

5.6. Communication Page

User communication with the coach and his feedback is a very important functionality in the application. Through communication page, the user is able to send the messages to his coach about his meal plan and exercise. Vice versa coach is also able to give his feedback on user request as showing the (figure 22)

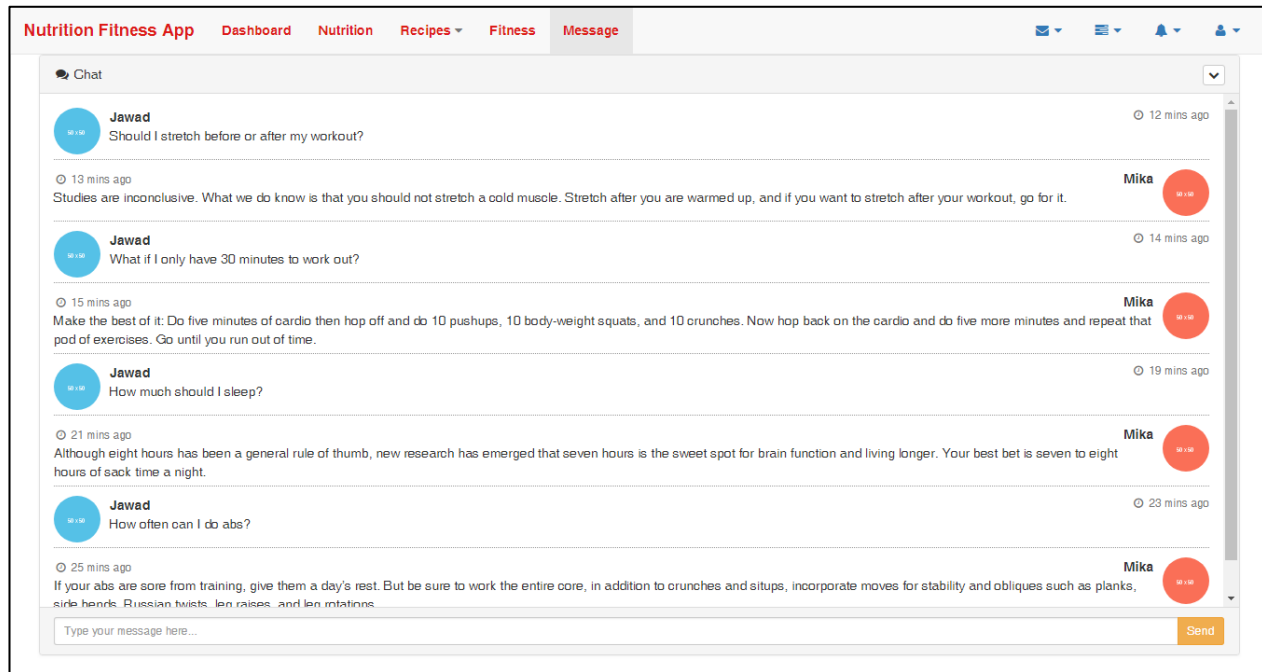


FIGURE 22 : Communication page


5.7. Coach Page View

After the successful login of coach user, he can see the list of registered users those assigned to him (figure 23). The coach can filter the member list by first name or last name from the member list. The coach can click on message link of any user and view his meal plan, exercise-related data and send the messages.

Club: San Francisco, USA Member Coach Name: ANAHEIM					Date: FRIDAY, MARCH 30, 2018
Member Name <input type="text" value="Enter a First Name and/or Last Name"/> <input type="button" value="Q"/>					
#	First Name	Last Name	Email	Weight	Action
1	Dawson	Debbie	dawson@gmail.com	200 lbs	Message
2	Jawad Ur Rehman	Saif	jawadpak@gmail.com	175 lbs	Message
3	Edward	Elana	edward.elana@yahoo.com	160 lbs	Message
4	Fisburne	Farrah	fisburne.farrah@hotmail.com	147 lbs	Message
5	Garcia	Glenda	garcia.glenda@hotmail.com	179 lbs	Message
6	Harris	Hatti	harris.hatti@hotmail.com	180 lbs	Message
7	Ingles	Ira	ingles.ira@hotmail.com	195 lbs	Message
7	Jarret	Joan	jarret.joan@yahoo.com	119 lbs	Message
8	Jackson	Jnot	jackson@hotmail.com	172 lbs	Message
9	Keller	Kate	kate@yahoo.com	197 lbs	Message
10	Larson	Lana	lana@hotmail.com	174 lbs	Message
Showing 1 to 10 of 57 entries					
Previous 1 2 3 4 5 6 Next					

FIGURE 23 : Coach member list page

When coach clicks on message link of the user, the user detail page opens (figure 24) where a coach can see the user profile information like name, email, date of birth etc. On the user detail page, a coach can see the menu links of nutrition, fitness and message. When coach clicks on nutrition link, the user nutrition-related information will appear, in the same way when the coach will click on fitness link the fitness related data appears. When coach clicks on message link, the message view appears as shown figure 24. The coach will write the title in the input box and the message in the text area. So, the coach can pass the comments to a user about his progress or any suggestion for his fitness; the later user can see this message on communication page and respond it.



Muhammad Jawad Ur Rehman

Club : San Francisco, USA

✉ jawadpak@gmail.com

♂ Gender : Male

👤 Coach Name : Michle Jhon

📅 Club Start Date: June 02, 2017

📅 Date: July 02, 1984

Contact Info

Office Address : San Francisco, USA

📍 Resident Address : San Francisco, 7A USA 78236

📞 Mobile : 0345-78945626

☎ Phone : +95-456789123


📘 Facebook

🐦 Twitter

[Nutrition](#) | [Fitness](#) | [Message](#)

Coach Member List

Title



Eating nothing but protein for three days will help drop a lot of water weight. It's not healthy, nor is it sustainable, but it works very well. Many commercial diets recommend this as a first step so that they can show that they have produced results and their customers don't ask for a refund. Just remember that it is only temporary, you lose very little fat, and you will put the water weight back on once you start to eat normally again. Probably best to not get into a state where you need to lose weight quickly...

[Cancel](#) [Send Message](#)

FIGURE 24: Coach user detail page

6 Discussion

Online health fitness application can be a very good technology to the consumers for maintaining her/his health fitness. The target audiences are fitness clubs and sports organizations (for sports players) because they especially need to follow their food nutrition chart and exercise's plan. But we can also use this application with minor changes for a common man. Now a day's health fitness is an evolving issue for the people. People have no idea about their health maintenance. People's health, well-being, and fitness are dependent on taking nutritious food and supplements. But they don't have an idea about their food intake and exercise. They also have no idea about the calories calculation and supplement intake. They don't know how much they must take water in a day, sleep in a day and how much their body burns calories when they do exercise, work, walk and even sleep.

The objective of this thesis is to implement the nutrition health fitness system for the health fitness clubs where end users can track information about her/his nutrition related to food and exercise and her/his coach even can evaluate her/his progress. The significance of implementing this thesis is to give the awareness of food nutrition value, so the user can easily create an equilibrium between in his food nutrition chart and exercises.

This thesis demands the extensive programming skills for designing and implementing Web application. This thesis work has been a continuous learning process, the food database creation from USDA has been a challenging task. Data inside USDA database was quite different from the requirements of the project. So, the transformation of data was required and for that process, whole USDA data was explored. To fetch the meaningful data, a separate script was written. To realize the envisioned application many new technologies were studied such as AngularJS 2 (version 5.0) which was a completely new thing for me, I did not have any prior experience with it. However, with continuous effort such issues were resolved which boosted my confidence and will be helpful in dealing with upcoming challenges.

This application is using client and server-based architecture which is the most renowned and modern architecture technique. On the server-side programming, I am using MySQL database with CakePHP (PHP framework). For data communication, I am mak-

ing RESTFUL API calls through which we can easily retrieve and store the data in the database.

On the client side, I am using the AngularJS for templating and mapping the data with the help of some useful JavaScript libraries: Lodash, JQuery. Highchart library is being used by the interactive graph. It's the responsive Web application compatible with all kinds of devices (mobile, tablet and desktop); for this purpose, I am using the bootstrap framework. To maintain the history of development life cycle and version controlling Git tool has been used.

6.1. Future Recommendation

For future enhancement point of view, I am sharing some ideas through which you can make this application more attractive and useful for the people.

- **Grocery Module:** the user can make his food meal plan in future dates, based on this plan user can find his grocery items for a month in advance. So, you can make a module where you extract the data from the food meal plan and show grocery items.
- **AI in Food Meal Plan:** during the user registration, the system can ask about user allergies, favorites cuisine for a food recommendation, if a woman is pregnant, she will need specific foods. If a user is allergic to some food items for example nuts or peanuts then the system will consider those items in meal/recipe planning. Similarly, if a woman is pregnant then the system will add only those items in the meal which are feasible for a pregnant woman.
- **Integrate Activity Tracker:** activity tracker also known as the fitness tracker is defined as “a device or application for monitoring and tracking fitness-related metrics such as distance walked or run, calorie consumption.” So, we can also integrate the application with the fitness tracker devices by the help of device API'S.

References

- Angular JS 2. (n.d.). *official website*. Retrieved from Angular JS 2: <https://angular.io/>
- BootStrap. (n.d.). *official website*. Retrieved from BootStrap: <https://getbootstrap.com/>
- Bumgardner, W. (2018). *Calories Does Walking Burn per Mile*. Retrieved from <https://www.verywell.com/walking-calories-burned-by-miles-3887154>
- Exercise, C. B. (n.d.). *Calories Burned During Exercise, Activities, Sports and Work*. Retrieved from Nutri Strategy: <http://www.nutristrategy.com/caloriesburned.htm>
- Lodash. (n.d.). *official website*. Retrieved from <https://lodash.com/>
- Maya E Nahra, R. (n.d.). *Breakdown of Daily Calories by Meal for Weight Loss*. Retrieved from <https://www.uccs.edu/Documents/healthcircle/pnc/health-topics/Meal%20Planning.pdf>
- NutriStrategy. (2015). *Calories Burned During Exercise, Activities, Sports and Work*. Retrieved from NutriStrategy Nutrition and Fitness: <http://www.nutristrategy.com/caloriesburned.htm>
- USDA. (n.d.). *Food Composition Databases*. Retrieved from United States Department of Agriculture: <https://ndb.nal.usda.gov/ndb/>
- WebMD LLC. (n.d.). *Estimated Calorie Requirements*. Retrieved from WebMd: <https://www.webmd.com/diet/features/estimated-calorie-requirement>

APPENDICES

Appendix 1. GitHub front end repository code

https://github.com/jawadpak/thesis_front_end

Appendix 2. GitHub backend repository code

https://github.com/jawadpak/thesis_rest_api

Appendix 3. Nutrition definition

NUTRITION DEFINATION			
Nutr_No	Units	Tagname	NutrDesc
203	g	PROCNT	Protein
204	g	FAT	Total lipid (fat)
205	g	CHOCDF	Carbohydrate, by difference
207	g	ASH	Ash
208	kcal	ENERC_KCAL	Energy
209	g	STARCH	Starch
210	g	SUCS	Sucrose
211	g	GLUS	Glucose (dextrose)
212	g	FRUS	Fructose
213	g	LACS	Lactose
214	g	MALS	Maltose
221	g	ALC	Alcohol, ethyl
255	g	WATER	Water
257	g		Adjusted Protein
262	mg	CAFFN	Caffeine
263	mg	THEBRN	Theobromine
268	kJ	ENERC_KJ	Energy
269	g	SUGAR	Sugars, total
287	g	GALS	Galactose
291	g	FIBTG	Fiber, total dietary
301	mg	CA	Calcium, Ca
303	mg	FE	Iron, Fe
304	mg	MG	Magnesium, Mg
305	mg	P	Phosphorus, P
306	mg	K	Potassium, K
307	mg	NA	Sodium, Na
309	mg	ZN	Zinc, Zn
312	mg	CU	Copper, Cu
313	µg	FLD	Fluoride, F
315	mg	MN	Manganese, Mn
317	µg	SE	Selenium, Se
318	IU	VITA_IU	Vitamin A, IU

319	µg	RETOL	Retinol
320	µg	VITA_RAE	Vitamin A, RAE
321	µg	CARTB	Carotene, beta
322	µg	CARTA	Carotene, alpha
323	mg	TOCPHA	Vitamin E (alpha-tocopherol)
324	IU	VITD	Vitamin D
325	µg	ERGCAL	Vitamin D2 (ergocalciferol)
326	µg	CHOCAL	Vitamin D3 (cholecalciferol)
328	µg	VITD	Vitamin D (D2 + D3)
334	µg	CRYPX	Cryptoxanthin, beta
337	µg	LYCPN	Lycopene
338	µg	LUT+ZEA	Lutein + zeaxanthin
341	mg	TOCPHB	Tocopherol, beta
342	mg	TOCPHG	Tocopherol, gamma
343	mg	TOCPHD	Tocopherol, delta
344	mg	TOCTRA	Tocotrienol, alpha
345	mg	TOCTRB	Tocotrienol, beta
346	mg	TOCTRG	Tocotrienol, gamma
347	mg	TOCTRD	Tocotrienol, delta
401	mg	VITC	Vitamin C, total ascorbic acid
404	mg	THIA	Thiamin
405	mg	RIBF	Riboflavin
406	mg	NIA	Niacin
410	mg	PANTAC	Pantothenic acid
415	mg	VITB6A	Vitamin B-6
417	µg	FOL	Folate, total
418	µg	VITB12	Vitamin B-12
421	mg	CHOLN	Choline, total
428	µg	MK4	Menaquinone-4
429	µg	VITK1D	Dihydrophyloquinone
430	µg	VITK1	Vitamin K (phyloquinone)
431	µg	FOLAC	Folic acid
432	µg	FOLFD	Folate, food
435	µg	FOLDFE	Folate, DFE
454	mg	BETN	Betaine
501	g	TRP_G	Tryptophan
502	g	THR_G	Threonine
503	g	ILE_G	Isoleucine
504	g	LEU_G	Leucine
505	g	LYS_G	Lysine
506	g	MET_G	Methionine
507	g	CYS_G	Cystine
508	g	PHE_G	Phenylalanine
509	g	TYR_G	Tyrosine
510	g	VAL_G	Valine
511	g	ARG_G	Arginine
512	g	HISTN_G	Histidine
513	g	ALA_G	Alanine

514	g	ASP_G	Aspartic acid
515	g	GLU_G	Glutamic acid
516	g	GLY_G	Glycine
517	g	PRO_G	Proline
518	g	SER_G	Serine
521	g	HYP	Hydroxyproline
573	mg		Vitamin E, added
578	µg		Vitamin B-12, added
601	mg	CHOLE	Cholesterol
605	g	FATR_N	Fatty acids, total trans
606	g	FASAT	Fatty acids, total saturated
607	g	F4D0	4:0
608	g	F6D0	6:0
609	g	F8D0	8:0
610	g	F10D0	10:0
611	g	F12D0	12:0
612	g	F14D0	14:0
613	g	F16D0	16:0
614	g	F18D0	18:0
615	g	F20D0	20:0
617	g	F18D1	18:1 undifferentiated
618	g	F18D2	18:2 undifferentiated
619	g	F18D3	18:3 undifferentiated
620	g	F20D4	20:4 undifferentiated
621	g	F22D6	22:6 n-3 (DHA)
624	g	F22D0	22:0
625	g	F14D1	14:1
626	g	F16D1	16:1 undifferentiated
627	g	F18D4	18:4
628	g	F20D1	20:1
629	g	F20D5	20:5 n-3 (EPA)
630	g	F22D1	22:1 undifferentiated
631	g	F22D5	22:5 n-3 (DPA)
636	mg	PHYSTR	Phytosterols
638	mg	STID7	Stigmasterol
639	mg	CAMD5	Campesterol
641	mg	SITSTR	Beta-sitosterol
645	g	FAMS	Fatty acids, total monounsaturated
646	g	FAPU	Fatty acids, total polyunsaturated
652	g	F15D0	15:0
653	g	F17D0	17:0
654	g	F24D0	24:0
662	g	F16D1T	16:1 t
663	g	F18D1T	18:1 t
664	g	F22D1T	22:1 t
665	g		18:2 t not further defined
666	g		18:2 i
669	g	F18D2TT	18:2 t,t

670	g	F18D2CLA	18:2 CLAs
671	g	F24D1C	24:1 c
672	g	F20D2CN6	20:2 n-6 c,c
673	g	F16D1C	16:1 c
674	g	F18D1C	18:1 c
675	g	F18D2CN6	18:2 n-6 c,c
676	g	F22D1C	22:1 c
685	g	F18D3CN6	18:3 n-6 c,c,c
687	g	F17D1	17:1
689	g	F20D3	20:3 undifferentiated
693	g	FATRNM	Fatty acids, total trans-monoenoic
695	g	FATRNP	Fatty acids, total trans-polyenoic
696	g	F13D0	13:0
697	g	F15D1	15:1
851	g	F18D3CN3	18:3 n-3 c,c,c (ALA)
852	g	F20D3N3	20:3 n-3
853	g	F20D3N6	20:3 n-6
855	g	F20D4N6	20:4 n-6
856	g		18:3i
857	g	F21D5	21:5
858	g	F22D4	22:4
859	g	F18D1TN7	18:1-11 t (18:1t n-7)

Appendix

/*

SQLyog Enterprise - MySQL GUI v8.1

MySQL - 5.7.14 : Database - tamk_nutrition_db

*/

/*!40101 SET NAMES utf8 */;

/*!40101 SET SQL_MODE="*/;

/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!Table structure for table `clubs` */

```
DROP TABLE IF EXISTS `clubs`;
```

```
CREATE TABLE `clubs` (
  `id` int(11) NOT NULL,
  `name` varchar(75) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `fitness_exercises` */
```

```
DROP TABLE IF EXISTS `fitness_exercises`;
```

```
CREATE TABLE `fitness_exercises` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `Uuid` varchar(32) DEFAULT NULL,
  `exercise_name` varchar(150) DEFAULT NULL,
  `130lb` int(11) DEFAULT NULL,
  `155lb` int(11) DEFAULT NULL,
  `180lb` int(11) DEFAULT NULL,
  `205lb` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=250 DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `food_groups` */
```

```
DROP TABLE IF EXISTS `food_groups`;
```

```
CREATE TABLE `food_groups` (
  `tree_id` int(9) NOT NULL,
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(60) COLLATE latin1_general_ci DEFAULT NULL,
  `tree_parent_id` int(10) unsigned DEFAULT '0' COMMENT 'self referece to FdGrp_CD',
  `parent_id` int(10) unsigned DEFAULT NULL,
  `lft` int(10) unsigned DEFAULT NULL,
```

```

`right` int(10) unsigned DEFAULT NULL,
`viewable_at_frontend` tinyint(3) unsigned DEFAULT '1',
`tree_node` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
`created` datetime NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5421055 DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;

```

```

/*Table structure for table `food_items` */

```

```

DROP TABLE IF EXISTS `food_items`;

```

```

CREATE TABLE `food_items` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `ndb_no` int(11) DEFAULT NULL,
  `uuid` varchar(36) COLLATE latin1_general_ci DEFAULT NULL,
  `food_groups_id` int(6) DEFAULT '0',
  `food_group_id_backup` int(11) DEFAULT NULL,
  `name` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `brand_name` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `short_name` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `boost` int(11) DEFAULT NULL,
  `ComName` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `ManufacName` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `Survey` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `Ref_Desc` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `Refuse` smallint(6) DEFAULT NULL,
  `SciName` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `N_Factor` double DEFAULT NULL,
  `Pro_Factor` double DEFAULT NULL,
  `Fat_Factor` double DEFAULT NULL,
  `CHO_Factor` double DEFAULT NULL,
  `shopping_alias_id` int(11) DEFAULT NULL,
  `is_snack` enum('Y','N') COLLATE latin1_general_ci NOT NULL DEFAULT 'N',

```

```

`gi_value` enum('high','medium','low') COLLATE latin1_general_ci DEFAULT
NULL,
`hidden` enum('Y','N') COLLATE latin1_general_ci NOT NULL DEFAULT 'N',
`good_food` tinyint(2) NOT NULL DEFAULT '0' COMMENT '0=normal, 1=good,
2=bad',
`patiente_searchable` tinyint(3) unsigned NOT NULL DEFAULT '1',
`comments` text COLLATE latin1_general_ci NOT NULL,
`user_id` int(11) DEFAULT NULL,
`is_restaurant` tinyint(1) NOT NULL DEFAULT '0',
`is_brand` tinyint(1) NOT NULL DEFAULT '0',
`is_solid` enum('Y','N') COLLATE latin1_general_ci NOT NULL DEFAULT 'Y',
`active` tinyint(4) NOT NULL DEFAULT '1',
`tls_food` tinyint(4) DEFAULT '0',
`tls_id` int(11) DEFAULT NULL,
`processed` tinyint(4) DEFAULT NULL,
`created` datetime DEFAULT NULL,
`updated` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
`serving` float(10,2) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `hide` (`hidden`),
KEY `hidden` (`hidden`,`active`),
KEY `name` (`name`),
KEY `active` (`active`,`name`,`hidden`)
) ENGINE=InnoDB AUTO_INCREMENT=770067 DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;

```

```

/*Table structure for table `meal_item_schedules` */

```

```

DROP TABLE IF EXISTS `meal_item_schedules`;

```

```

CREATE TABLE `meal_item_schedules` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `name` varchar(150) DEFAULT NULL,
  `item_id` int(11) DEFAULT NULL,

```

```

`item_type` enum('food','recipe','meal','simple_meal') DEFAULT NULL,
`meal_data` text,
`major` varchar(15) DEFAULT NULL,
`minor` varchar(15) DEFAULT NULL,
`unit` varchar(50) DEFAULT NULL,
`sat` tinyint(2) DEFAULT NULL,
`sun` tinyint(2) DEFAULT NULL,
`mon` tinyint(2) DEFAULT NULL,
`tue` tinyint(2) DEFAULT NULL,
`wed` tinyint(2) DEFAULT NULL,
`thu` tinyint(2) DEFAULT NULL,
`fri` tinyint(2) DEFAULT NULL,
`from_date` date DEFAULT NULL,
`end_date` date DEFAULT NULL,
`week` smallint(5) DEFAULT NULL,
`meal_sequences` varchar(300) DEFAULT NULL,
`created` datetime DEFAULT NULL,
`updated` datetime DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=180 DEFAULT CHARSET=latin1;

```

```

/*Table structure for table `meal_items` */

```

```

DROP TABLE IF EXISTS `meal_items`;

```

```

CREATE TABLE `meal_items` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `uuid` varchar(50) NOT NULL,
  `meal_id` int(11) NOT NULL,
  `parent_id` int(10) unsigned DEFAULT NULL,
  `food_item_type` varchar(100) DEFAULT NULL,
  `food_item_id` int(11) DEFAULT NULL,
  `name` varchar(255) DEFAULT NULL,
  `size_major` float(5,2) DEFAULT NULL,
  `size_minor` varchar(10) DEFAULT NULL,

```

```

`unit` varchar(255) DEFAULT NULL,
`weight` int(5) DEFAULT NULL,
`weight_per_unit` int(5) DEFAULT NULL,
`prep_time` varchar(50) DEFAULT NULL,
`cook_time` varchar(50) DEFAULT NULL,
`directions` text,
`tracked` enum('Y','N') DEFAULT 'N',
`tracked_datetime` datetime DEFAULT NULL,
`approved` tinyint(1) DEFAULT '1',
`recommended` tinyint(2) NOT NULL DEFAULT '0',
`active` tinyint(2) NOT NULL DEFAULT '1',
`created_by` int(11) DEFAULT NULL,
`updated_by` int(11) DEFAULT NULL,
`created` datetime DEFAULT NULL,
`updated` datetime DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `active` (`active`,`parent_id`,`meal_id`),
KEY `meal_id` (`meal_id`),
KEY `uuid` (`uuid`)
) ENGINE=InnoDB AUTO_INCREMENT=279420 DEFAULT CHARSET=latin1;

```

```

/*Table structure for table `meals` */

```

```

DROP TABLE IF EXISTS `meals`;

```

```

CREATE TABLE `meals` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `uuid` varchar(36) NOT NULL,
  `name` varchar(255) DEFAULT NULL,
  `title` varchar(50) DEFAULT NULL,
  `directions` text NOT NULL,
  `sequence` tinyint(4) NOT NULL,
  `tracked` tinyint(1) unsigned DEFAULT '0',
  `is_valid` tinyint(4) NOT NULL DEFAULT '1',
  `approved` tinyint(1) DEFAULT '1',

```

```

`created_by` int(11) DEFAULT NULL,
`updated_by` int(11) DEFAULT NULL,
`created` datetime DEFAULT NULL,
`updated` datetime DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `uuid` (`uuid`)
) ENGINE=InnoDB AUTO_INCREMENT=49775 DEFAULT CHARSET=latin1;

```

```

/*Table structure for table `recipe_items` */

```

```

DROP TABLE IF EXISTS `recipe_items`;

```

```

CREATE TABLE `recipe_items` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `recipe_id` int(11) NOT NULL,
  `food_item_id` int(10) unsigned NOT NULL,
  `food_group_id` int(11) DEFAULT NULL,
  `food_exchange_id` int(11) DEFAULT NULL,
  `food_name` varchar(255) NOT NULL,
  `comments` char(100) DEFAULT NULL,
  `size_major` varchar(6) DEFAULT NULL,
  `size_minor` varchar(6) DEFAULT NULL,
  `size_unit` varchar(255) DEFAULT NULL,
  `m_size_major` varchar(6) DEFAULT NULL,
  `m_size_minor` varchar(6) DEFAULT NULL,
  `m_size_unit` varchar(255) DEFAULT NULL,
  `gi_rating` enum('nogi','high','medium','low') DEFAULT 'nogi',
  `gm_weight` float unsigned DEFAULT NULL,
  `MA_ID` int(11) DEFAULT '0',
  `sequence` int(10) DEFAULT NULL,
  `created` datetime NOT NULL,
  `updated` datetime NOT NULL,
  `created_by` int(10) unsigned NOT NULL,
  `updated_by` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id`),

```



```

KEY `recipe_id` (`recipe_id`)
) ENGINE=InnoDB AUTO_INCREMENT=9200727 DEFAULT CHARSET=latin1;

```

```

/*Table structure for table `recipe_nutrients` */

```

```

/*Table structure for table `recipes` */

```

```

DROP TABLE IF EXISTS `recipes`;

```

```

CREATE TABLE `recipes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `uuid` varchar(40) DEFAULT NULL,
  `name` varchar(255) DEFAULT NULL,
  `slug` varchar(255) DEFAULT NULL,
  `directions` text,
  `prep_time` smallint(6) unsigned DEFAULT NULL,
  `image_name` varchar(50) DEFAULT NULL,
  `cook_time` smallint(6) unsigned DEFAULT NULL,
  `servings` tinyint(4) unsigned NOT NULL DEFAULT '1',
  `meal_gi` enum('low','medium','high','nogi') DEFAULT 'nogi',
  `manually_added` enum('Y','N') NOT NULL DEFAULT 'N',
  `master_recipe_id` int(11) DEFAULT NULL,
  `master_recipe_type` enum('public','self','favorite','none') DEFAULT 'none',
  `meal_types` tinyint(4) NOT NULL,
  `recipe_classification_id` smallint(6) NOT NULL,
  `countries` tinyint(4) NOT NULL,
  `allergies` float DEFAULT NULL,
  `created_by_community` tinyint(1) DEFAULT '0',
  `master` tinyint(1) unsigned NOT NULL DEFAULT '1',
  `is_meal` int(5) NOT NULL DEFAULT '0',
  `approved` tinyint(1) DEFAULT '0',
  `sequence` int(11) DEFAULT NULL,
  `has_photo` tinyint(4) DEFAULT '0',
  `active` tinyint(1) unsigned NOT NULL DEFAULT '1',
  `created` datetime NOT NULL,

```

```

`updated` datetime NOT NULL,
`created_by` int(10) unsigned NOT NULL,
`updated_by` int(10) unsigned NOT NULL,
PRIMARY KEY (`id`),
KEY `id` (`id`,`active`,`master`,`meal_types`),
KEY `master` (`master`,`active`,`meal_types`)
) ENGINE=InnoDB AUTO_INCREMENT=1324769 DEFAULT CHARSET=latin1;

```

```

/*Table structure for table `user_daily_exercises` */

```

```

DROP TABLE IF EXISTS `user_daily_exercises`;

```

```

CREATE TABLE `user_daily_exercises` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `uuid` varchar(50) DEFAULT NULL,
  `fitness_exercise_id` int(11) DEFAULT NULL,
  `calories_burn` int(11) DEFAULT NULL,
  `exercise_date` datetime DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

/*Table structure for table `user_daily_steps` */

```

```

DROP TABLE IF EXISTS `user_daily_steps`;

```

```

CREATE TABLE `user_daily_steps` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `uuid` varchar(50) DEFAULT NULL,
  `user_id` int(11) DEFAULT NULL,
  `daily_step` int(11) DEFAULT NULL,
  `daily_date` date DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

/*Table structure for table `user_messages` */

```

```
DROP TABLE IF EXISTS `user_messages`;
```

```
CREATE TABLE `user_messages` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `uuid` varchar(40) DEFAULT NULL,
  `user_sender_id` int(11) DEFAULT NULL,
  `user_to_id` int(11) DEFAULT NULL,
  `message` varchar(500) DEFAULT NULL,
  `date_time` datetime DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `user_plan_meals` */
```

```
DROP TABLE IF EXISTS `user_plan_meals`;
```

```
CREATE TABLE `user_plan_meals` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `user_id` int(10) unsigned NOT NULL,
  `user_plan_id` int(11) NOT NULL,
  `meal_name` varchar(30) DEFAULT NULL,
  `meal_id` int(10) NOT NULL,
  `meal_item_id` int(11) NOT NULL,
  `tracked` tinyint(1) DEFAULT '0',
  `sequence` tinyint(4) DEFAULT NULL,
  `active` tinyint(4) DEFAULT '1',
  `created` datetime DEFAULT NULL,
  `updated` datetime DEFAULT NULL,
  `created_by` int(10) unsigned DEFAULT NULL,
  `updated_by` int(10) unsigned DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `user_plans` */
```

```
DROP TABLE IF EXISTS `user_plans`;
```

```
CREATE TABLE `user_plans` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `uuid` varchar(36) NOT NULL,
  `user_id` int(11) NOT NULL,
  `plan_date` date NOT NULL,
  `water_intake` int(11) DEFAULT '0',
  `approved` tinyint(1) DEFAULT '1',
  `bed_time` time DEFAULT NULL,
  `wakeup_time` time DEFAULT NULL,
  `sleep_feel` enum('Not rested','Neutral','Rested') DEFAULT NULL,
  `sleep` int(11) DEFAULT NULL,
  `is_active_meal_plan` tinyint(1) NOT NULL DEFAULT '1',
  `free_day` tinyint(2) DEFAULT '0',
  `meal_plan_created` tinyint(2) DEFAULT '1',
  `created` datetime DEFAULT NULL,
  `updated` datetime DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`,`plan_date`)
) ENGINE=InnoDB AUTO_INCREMENT=14584 DEFAULT CHARSET=latin1;
```

```
/*Table structure for table `users` */
```

```
DROP TABLE IF EXISTS `users`;
```

```
CREATE TABLE `users` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `uuid` char(36) DEFAULT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `preferred_language` varchar(64) DEFAULT NULL,
  `club_id` int(11) DEFAULT '1',
  `workout_start_date` date DEFAULT NULL,
```

```

`coach_id` int(11) DEFAULT NULL,
`is_coach` enum('true','false') DEFAULT 'false' COMMENT 'if the user can collabo-
rate with other (selective) users as a coach',
`first_name` varchar(100) DEFAULT NULL,
`last_name` varchar(100) DEFAULT 'M',
`email` varchar(255) DEFAULT NULL,
`gender` enum('M','F') DEFAULT NULL,
`dob` date DEFAULT NULL,
`height` int(11) DEFAULT NULL,
`country` varchar(255) DEFAULT NULL,
`city` varchar(50) DEFAULT NULL,
`state` varchar(50) DEFAULT NULL,
`zip` varchar(10) DEFAULT NULL,
`state_code` char(3) DEFAULT NULL,
`street1` varchar(255) DEFAULT NULL,
`street2` varchar(100) DEFAULT NULL,
`phone` varchar(15) DEFAULT NULL,
`photo` varchar(50) DEFAULT NULL,
`about_me` text,
`agreed` tinyint(1) DEFAULT '0',
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

/*Table structure for table `weights` */

```

```

DROP TABLE IF EXISTS `weights`;

```

```

CREATE TABLE `weights` (
  `id` mediumint(9) NOT NULL AUTO_INCREMENT,
  `food_item_id` int(9) NOT NULL,
  `seq` tinyint(2) NOT NULL,
  `amount` double DEFAULT NULL,
  `measure` varchar(255) CHARACTER SET latin1 DEFAULT NULL,
  `m_amount` double DEFAULT NULL,
  `m_measure` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,

```

```

`weight` double DEFAULT NULL,
`Num_Data_Pts` smallint(6) DEFAULT NULL,
`Std_Dev` float DEFAULT '0',
`default_unit` char(1) COLLATE latin1_general_ci DEFAULT 'N',
PRIMARY KEY (`id`),
KEY `Num_Data_Pts` (`Num_Data_Pts`),
KEY `NDB_No` (`food_item_id`,`seq`),
KEY `NDB_No_2` (`food_item_id`),
KEY `food_item_id` (`food_item_id`,`measure`)
) ENGINE=InnoDB AUTO_INCREMENT=113096 DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

```