# Nutritional wizard

by

Nikitha Vootla

B.Tech., Jawaharlal Technological University, 2015

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2017

Approved by:

Major Professor
Dr. Mitchell Neilsen

# Copyright

# Abstract

It is well recognized that staying fit and healthy is a significant public problem now days and engaging in moderate levels of physical activity and having healthy food is associated with positive outcomes. Many dietary applications are being used at present times, but they don't include physical activity in their assessment and even if they do so they require user to manually enter the data which gives room for unintentional mistakes. Nutritional wizard is a dietary and nutritional coach that works synchronously with Fitbit data to achieve this. This application analyses the food intake of an individual and automatically inspects this with his physical activity recorded through the Fitbit to give a cover on his health and fitness.

The application has four main parts, input food, analysis, profile, and recipes. In input food part, user gives all the details of the food he had on that day and nutritional wizard gives the person's calorie intake, steps he made on that day, calories spent, excess calories stored, steps he has to still make on that day. Analysis part of the application gives graphical representations of daily calorie intake in a specific time frame given by user.

Profile has user's height and weight based on which BMI of user is calculated and is represented in a Gantt chart showing if he is in low, good or high BMI. Last Recipes module has few recipes categorized into different selections low carb, low fat, and low protein which user can use to decrease his calorie intake.

The completely developed app can be used by any user maintaining a fit bit account to provide good nutritional benefits.

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgements

I would like to express my sincere gratitude to my Major Professor, Dr. Mitchell Neilsen for his encouragement and for trusting my abilities to complete this project.

I take immense pleasure in extending my heartfelt thanks to my committee members Dr. Daniel Andresen and Dr. Torben Amtoft for their encouragement and for taking the time to serve on my committee.

I take this opportunity to acknowledge the support and help received from the academic staff of the Department of Computing and Information Sciences.

I would like to thank my parents and friends for their immense love and belief.

# Chapter 1 - Introduction

## 1.1 Project Description

Every day we make decisions of what food we are having and what nutrition it consists of and these choices determine how healthy we will be in long run. Right amount of food is required for us to be healthy because too much food makes us obese while too less can make us malnourishment. The proposed application 'Nutritional Wizard' provides solution for the above decisions. It provides you a clear picture of what your body is and how healthy you are. This decision of Nutritional Wizard is not only based on food that we have but also considers your physical activity of a person from data recorded through Fitbit.

The system consists of three major parts, input food, profile analysis and recipes. Input food part takes all the meal details that we had on the day and gives a report on how many calories we had, how many have been spent by walking and how many of them are stored in the body. Profile analysis takes up the present height and weight of a person and analyses if the person is correct weight range or overweight or underweight. Recipes section has several recipes that user can use based on his requirement and interests. This is achieved using asp.Net framework, SQL server, HTML and JavaScript.

## 1.2 Motivation

In this busy world, it has become almost impossible for a person to take care of his health. So, as the world is automating, user needs a system to keep track of his health as well. There are many existing nutritional websites but they rely on user manual input of physical activity which may give scope for errors. So, the idea of developing an application that would automatically sync his activity with his/her food plan got me excited to work on this application.

# Chapter 2 - Background and Technologies Used

Most of the system built is achieved using web application features. For the front-end development, HTML5, CSS is used. For server side scripting ASP.Net and for client side JQuery has been used. As part of database management, I choose SQL Server to achieve it. For Fitbit integration, IS Server should be used as Fitbit API communicates only through IS Server. All the programming is done in C# using Visual Studio IDE. Below is a detailed explanation of all the technologies that are used in this project development.

## 2.1 Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) developed by Microsoft. This IDE is used to develop computer programs for Microsoft Windows, web apps, web sites, web services and mobile apps.

Like any other IDE, it comprises of a code editor that supports syntax highlighting and code completion with IntelliSense for variables, functions, methods, loops and LINQ queries. It can produce both native code and managed code. Also, the integrated debugger works as a source-level debugger as well as a machine-level debugger. Additional built-in tools include web designer, forms designer for building GUI applications, class designer, a code profiler, and database schema designer.

Visual Studio supports diverse programming languages and allows the code editor and debugger to support almost any programming language, if a language-specific service exists. Built-in languages include C, C++ and C++/CLI, VB.NET, C#, F# and Typescript. Support for additional languages such as Python, Ruby, Node.js, and M among others is offered via language services installed separately. It similarly supports XML/XSLT, HTML/XHTML, JavaScript, and CSS. Java (and J#) were supported in the past.

Visual Studio uses Microsoft software development platforms such as Windows Store, Windows Presentation Foundation, Windows API, Windows Forms and Microsoft Silverlight. It accepts plug-ins that advance the functionality at nearly every level.

Microsoft provides a free version of Visual Studio called the Community edition that supports plugins and is available at no cost. Other versions are Professional, Enterprise, Test Professional, Express editions. The code editor is used for all supported languages. Visual Studio contains a debugger that works both as a source-level debugger and as a machine-level debugger which works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes and monitor and debug those processes. Visual Studio lets developers to write extensions for Visual Studio to spread its capabilities. These extensions plug into Visual Studio and extend its functionality. Extensions are available in the form of macros, add-ins, and packages. Macros signify repeatable tasks and actions that developers can record programmatically for saving, replaying, and distributing.

## 2.2 ASP.Net Framework

ASP.NET is an open-source server-side web application framework developed by Microsoft and is designed for web development to produce dynamic web pages. It allows programmers to build web applications, dynamic web sites and web services.

ASP.NET's successor is ASP.NET Core. ASP.NET core is a re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The new framework uses the new open-source .NET Compiler Platform and is cross platform.
It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language

Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages have merged into a unified MVC 6.

Architecture:

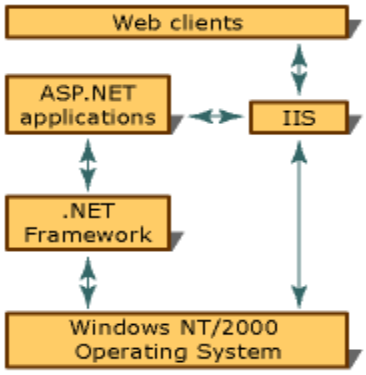


**Figure 2-1 ASP.Net Architecture**

Microsoft.*ASP.Net architecture.* [Animation].Retrieved from: https://msdn.microsoft.com/en-us/library/yedba920(v=vs.71).aspx

## 2.3 SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. This is a software product with the primary function of storing and retrieving data as requested by other software applications that may run either on the same computer or on another computer across a network. Microsoft makes SQL Server available in multiple editions, with different feature sets and targeting different users.

There are around a dozen different editions of Microsoft SQL Server which are aimed at different audiences and for different workloads ranging from small single-machine applications to large Internet-facing applications.

Microsoft Visual Studio includes native support for data programming with Microsoft SQL Server. It can be used to write and debug code to be executed by SQL CLR. Also, it includes a data designer that can be used to graphically view, create or edit database schemas. Queries can be created either visually or using code.

SQL Server Management Studio is a GUI tool included with SQL Server 2005 and later versions for configuring, managing, and administering all components within Microsoft SQL Server.

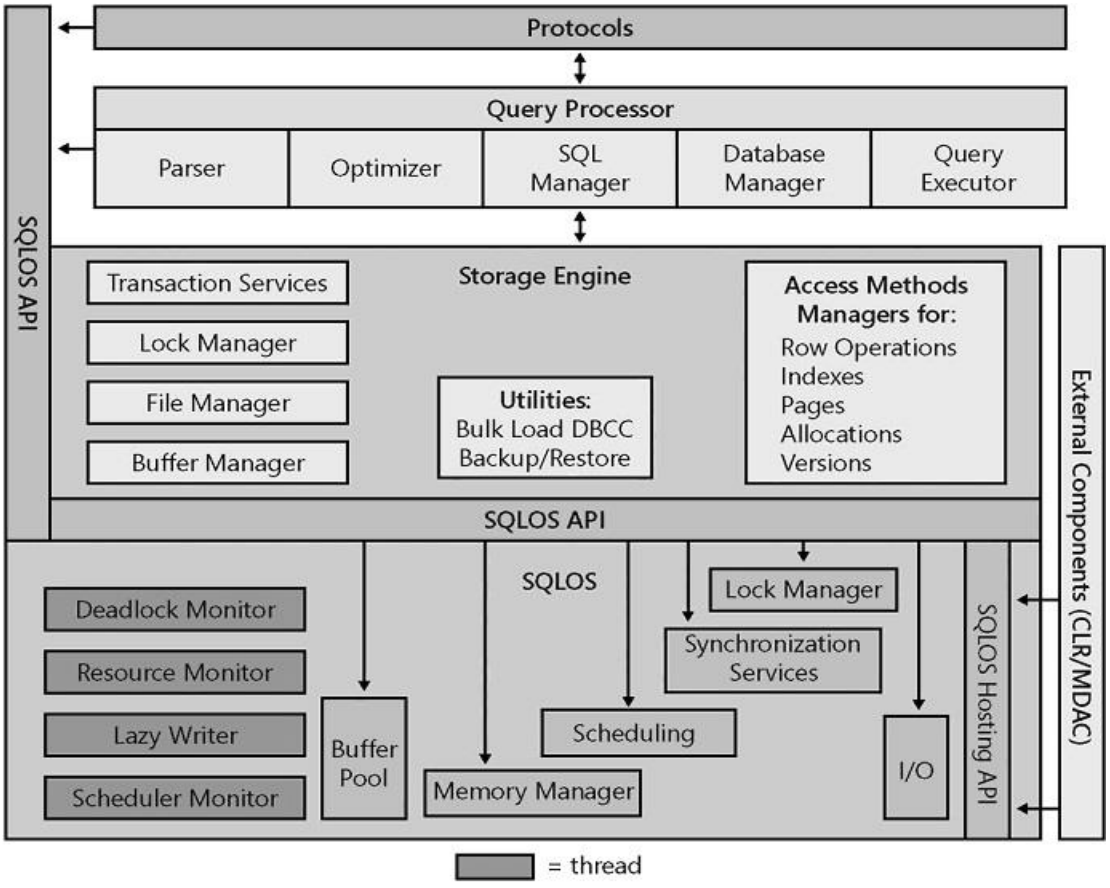


**Figure 2-2 SQL Server Architecture**

Saurabh Sinhain.*SQL server architecture.* [Animation].Retrieved from: http://saurabhsinhainblogs.blogspot.com/2010/02/sql-server-architechture-interview.html

## 2.4 JQuery

JQuery is a fast, small, and feature-rich JavaScript library. It makes things like event handling, animation, HTML document traversal and manipulation and Ajax much simpler. With a mixture of versatility and extensibility, it has altered the way that millions of people write JavaScript.

It is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT license. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

JQuery's syntax is intended to make it easier to create animations, handle events, navigate a document, select DOM elements and develop Ajax applications. JQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

The set of jQuery core features—DOM element selections, traversal and manipulation—enabled by its selector engine, created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API.

Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX and ASP.NET MVC frameworks while Nokia has integrated it into the Web Run-Time widget development platform.

JQuery, at its core, is a Document Object Model (DOM) manipulation library. The DOM is a tree-structure representation of all the elements of a Web page. JQuery simplifies the syntax

for finding, selecting, and manipulating these DOM elements. For example, jQuery can be used for finding an element in the document with a certain property (e.g. all elements with an h1 tag), changing one or more of its attributes (e.g. color, visibility), or making it respond to an event (e.g. a mouse click).

JQuery also provides a paradigm for event handling that goes beyond basic DOM element selection and manipulation. The event assignment and the event callback function definition are done in a single step in a single location in the code. JQuery also aims to incorporate other highly used JavaScript functionality (e.g. fade INS and fade outs when hiding elements, animations by manipulating CSS properties).

# Chapter 3 - Requirements and System Design

## 3.1 Requirement Analysis

Requirement analysis is a key part of software development and I can define this stage as more important than the actual build because a mistake in the requirements phase can cost a huge loss in the future. Unless the developer has a clear idea of what his/her requirements are he will not be able to build the system correctly. In the requirements analysis of this project, I initially thought about how a new user would face problems regarding their weight. With this thought in mind, I defined main parts of the system and their expectations. The requirements gathered with help of Dr. Mitch Neilsen for this project are defined below:

- User should be able to give details of all the food he had on that day and he should get total no. of calories he consumed.
- User should be able to see how many steps he had on that day and how many calories were consumed for this activity.
- User should get calories that he had stored on that day based on food he took and activity he performed.
- User should be able to calculate his BMI value based on present weight.
- User should be able to access many recipes categorized into low carbohydrate, low protein and low fat

### 3.1.1 Hardware Requirements

- Processor: Intel Core i5
- Processor speed: 2.3 GHz
- RAM: 8GB

### 3.1.2 Software Requirements

- Visual Studio
- .NET framework
- SQL Server
- HTML5, CSS
- IS Server
- C#

## 3.2 System Design

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. System design is a pictorial representation of the functioning of the system. System design is dependent on the specified requirements. In a nut shell, system design is a graphical representation of requirements document which can be built using UML diagrams.

### 3.2.1 UML Diagrams

UML stands for Unified Modeling Language. UML is a system that allows user to model the software application by graphical layout. UML gives both static and dynamic view of the system. The diagrams that show the static or structural view are class, object, storage, deployment and package diagrams whereas activity, interaction, state and use case diagrams show the behavioral or dynamic view of the system. Following UML diagrams give a brief idea about the present system.

#### 3.2.1.1 Class Diagram

Class diagram shows the static view of the application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing

executable code of the software application. The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. It provides analysis and design of static view of the system, describes responsibilities of the system and used for forward and reverse engineering. Class diagrams are the most popular UML diagrams used for construction of software applications. Below diagram shows the class diagram for this application.

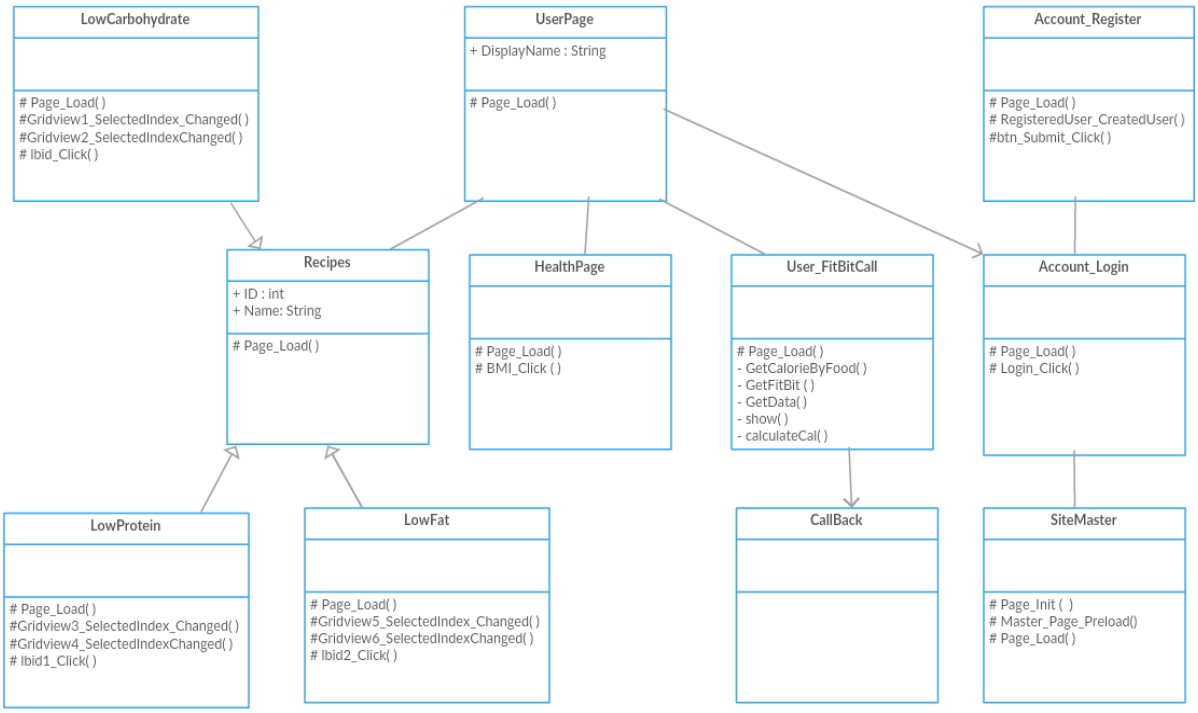


**Figure 3-1 Class Diagram**

### 3.2.1.2 Use Case Diagram

To model a system the most important aspect is to capture the dynamic behavior. To clarify a bit in details, dynamic behavior means the behavior of the system when it is running /operating. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. The use case diagram is used to gather

requirements of system, get an outside view, identify internal and external factors and show interaction among them with actors. The following diagram represents the use case diagram of this system where User is defined as actors and the operations they can perform are defined as use cases

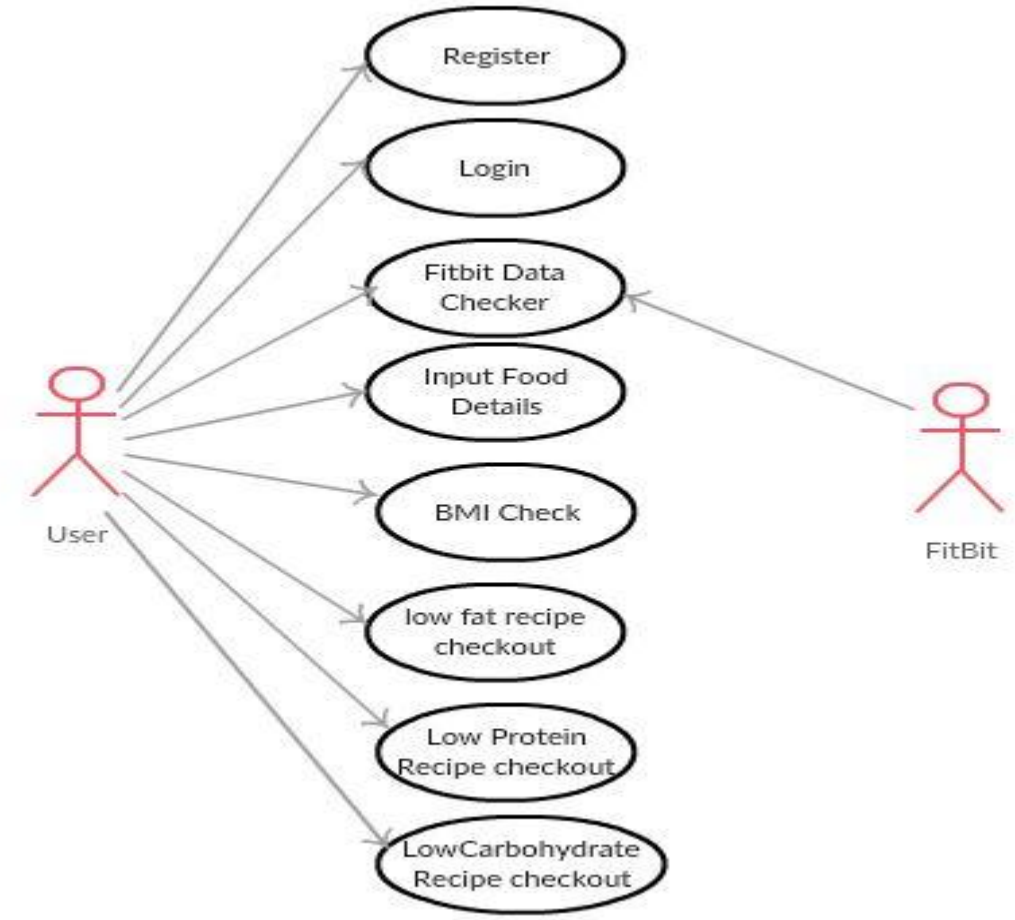


**Figure 3-2 Use Case Diagram**

### 3.2.1.3 Activity Diagram

Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagram is used to draw activity flow of

system, sequence from one activity to another and to describe parallel, branched and concurrent flow of the system. The activity diagram in this system is represented as shown in Figure 3-2

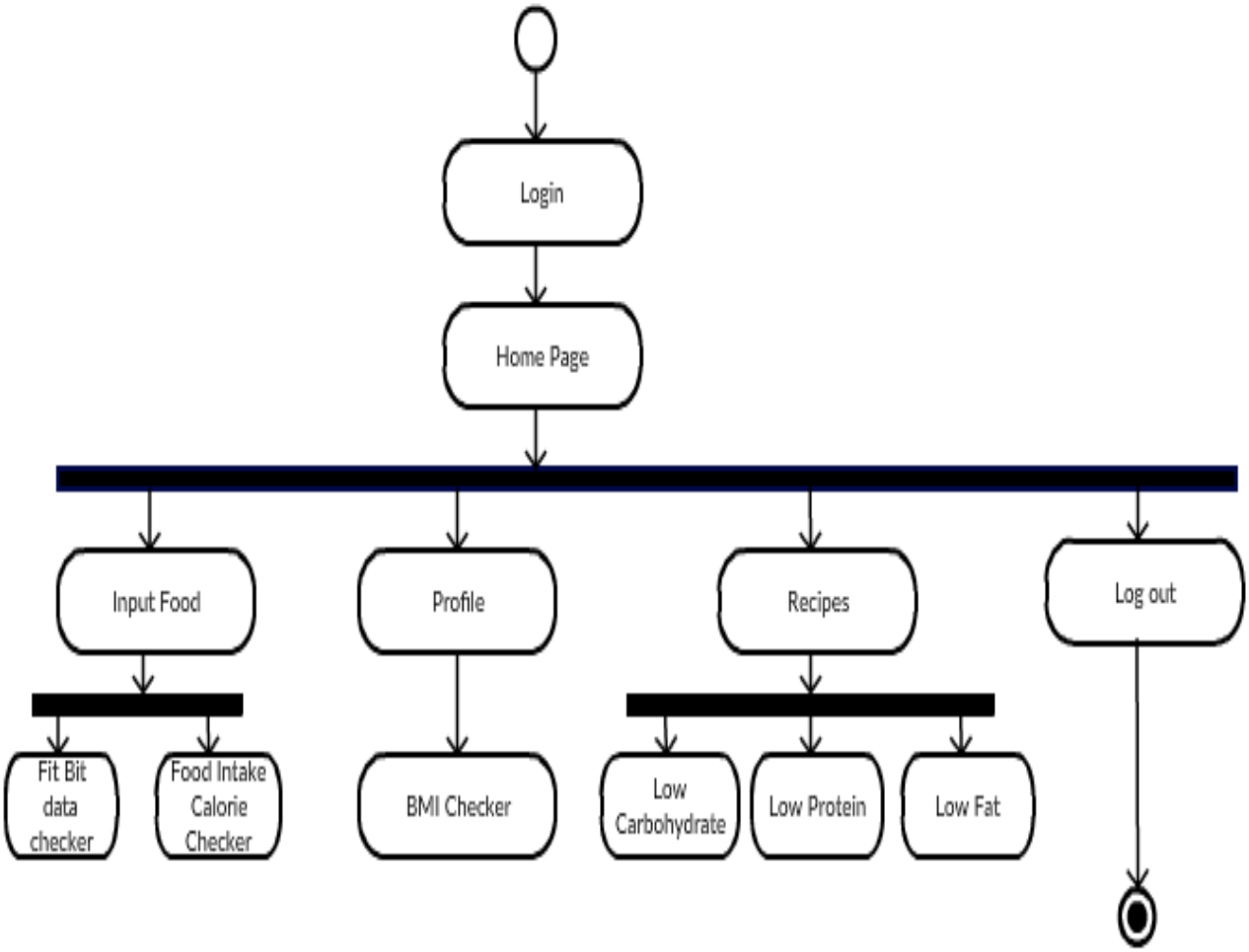


**Figure 3-3 Activity Diagram**

# Chapter 4 - Building the System

This system is built in two different parts. One part involved developing the nutritional part of this application and second part involved Fitbit integration into the web application. As First part of the project involved a web app, it required extensive coding in web scripting languages like HTML5, CSS and ASP.Net. It also involved communication with the database for which SQL server was used. For the second part of Fitbit integration, it involved creating an API and calling web services to the Fitbit API. Most of the scripting for it is done in c# programming language. Below is detailed explanation about each of these parts

## 4.1 Website Building

This part involved all the front end development and database connectivity. To achieve the front end development, HTML5, CSS, asp.Net framework and c# programming language was used. In any project to store the user data there should be a database. Likewise in this project too I have chosen SQL server for data management, HTML for content and structure, CSS for style and presentation, JQuery for client side scripting and ASP.Net for server side scripting.

### 4.1.1 Database

Based on the requirement, this module should have three entities namely Users, Health, Nutrition, low carbohydrates, low protein and low fat. Users will have data about the health. Nutrition has all food details and also low carbohydrates, low protein and low fat recipes. An E/R diagram shows the entities, attributes and their relation in a schema and can be later converted to a relational schema to maintain the project database. Below diagram shows the entity relationship between them.

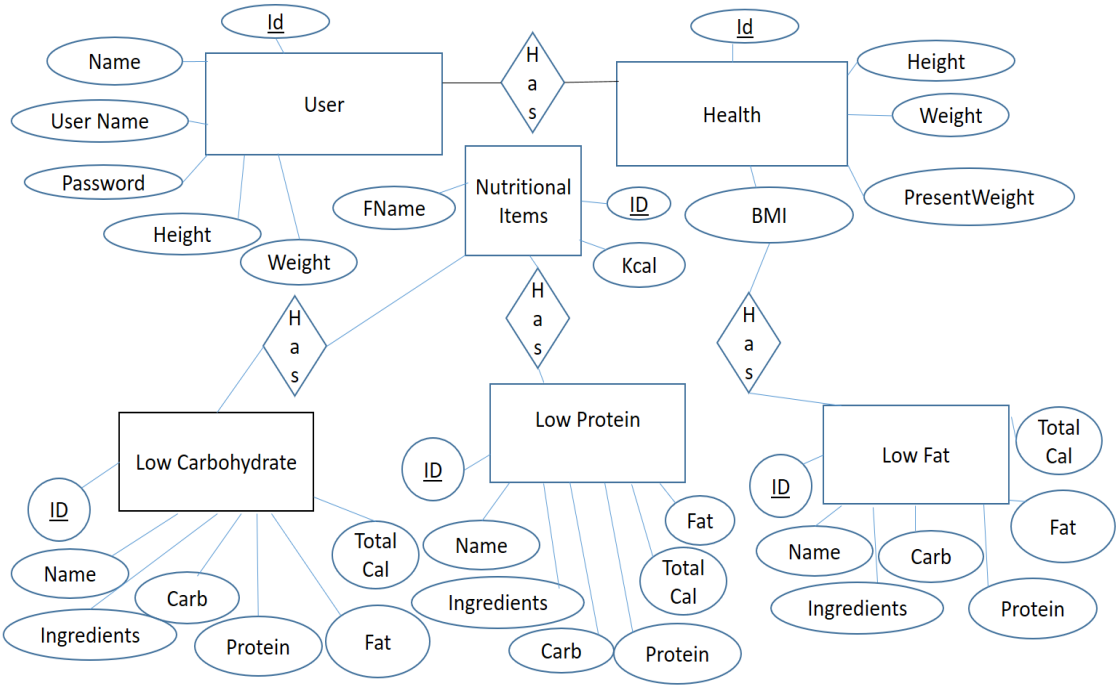


**Figure 4-1 E/R diagram**


**Figure 4-2 Relational Schema**

## 4.2 Fitbit Integration

To get data recorded through Fitbit, we need to create an API and call web services in it to Fitbit API.

- Register In Fitbit API
- Call web services in our API to Fitbit API

All these is achieved using the OAuth 2.0 authorization flow

### 4.2.1 Fitbit API

To use the Fitbit API we first need to register our application to the API. After registration we get our Client ID and Client Secret through which we can access the API. Next we need to implement an OAuth 2.0 authorization flow to allow people to give your app permission to access data on their behalf. Then we need to make HTTP requests to access the data. To register my app in Fitbit API I used following details

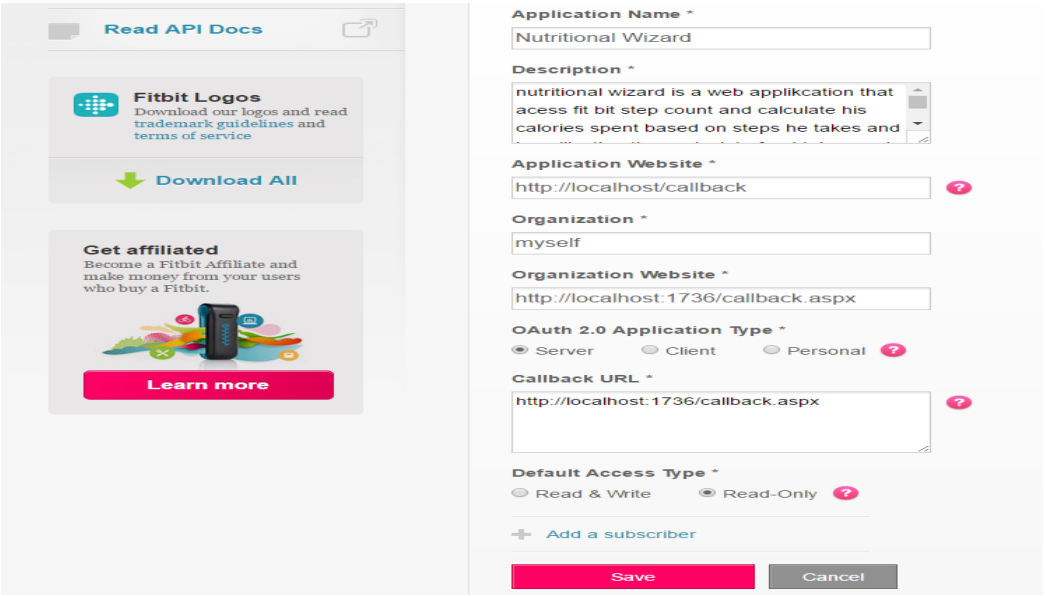


**Figure 4-3 Application Registration**

The main section of the app is Fitbit integration and which is achieved mainly by OAuth 2.0 authorization flow. The OAuth 2.0 framework requires your application to obtain an Access Token when the Fitbit user authorizes your app to access their data. The Access Token is used

for making HTTP request to the Fitbit API. Fitbit supports the **Authorization Code Grant** and **Implicit Grant** flows. The Authorization Code Grant flow is recommended for applications that have a web service. This flow requires server-to-server communication using an application's client secret. Authorization Code Grant flow functions as following:

- Application is redirected to Fitbit authorization page.
- After user's authorization, user is redirected back to the callback URL provided.
- Exchange of authorization code for access token and refresh tokens.
- Application stores these access tokens for making requests to Fitbit API

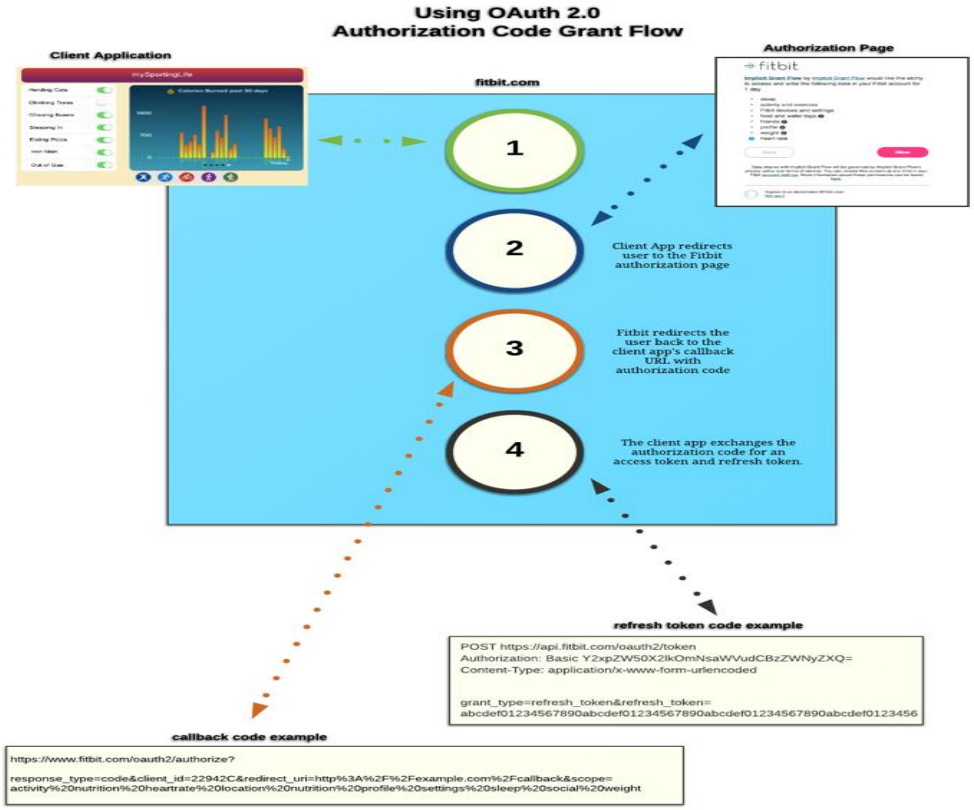


**Figure 4-4 OAuth 2.0 Authorization Code Grant Flow**

FitBit.*OAuth 2.0 Authorization Code Grant Flow*. [Animation].Retrieved from https://dev.fitbit.com/docs/oauth2/

### 4.2.2 Web Service Call

After we register our app in the Fitbit all of the authorization should be done in our project as mentioned above OAuth 2.0 Authorization Code Grant flow. This is implemented in following way. For security reasons client ID and Client Secret were not included in the documentation but used the words Client-ID and Client Secret for these two things.

- First, we need to authorize the user.

  Redirect("https://www.fitbit.com/oauth2/authorize?response_type=code&client_id=XXXXX&display=touch&redirect_uri=http://localhost:1736/callback.aspx&scope=activity%20location%20profile%20settings%20social%20weight");

- Then we need to exchange tokens.

```
var Atoken = Base64Encode("CLIENT-ID:CLIENT-SECRET");
 using (var wclient = new WebClient())
 {
var data = new NameValueCollection();
data["grant_type"] = "authorization-code";
data["client_id"] = "CLIENT-ID";
data["redirect_uri"] = " http://localhost:1736/callback.aspx ";
data["code"] = code;
wclient.Headers["Authorization"] = "Basic " + Atoken;
var response = wb.UploadValues("https://api.fitbit.com/oauth2/token", "POST", data);
var responseString = Encoding.ASCII.GetString(response);
});
```

- Then we need to make a web service to get data from Fitbit API

```
string userurl = "https://api.fitbit.com/1/user/" + user.encoded_userid + "/activities/date
selectedDate.json";
HttpWebRequest requesturl = (HttpWebRequest)WebRequest.Create(userurl);
requesturl.Method = "GET";
requesturl.Headers["Authorization"] = "Bearer " + user.access_token;
requesturl.Accept = "application/json";
WebResponse myResult= requesturl.GetResponse();
StreamReader httpwebStreamReader = new
StreamReader(myResult.GetResponseStream());
  String results = httpwebStreamReader.ReadToEnd();
```

In this way Fitbit is integrated into our web application. By using all the above components together in ASP.Net framework the complete system is built in modular way.

# Chapter 5 - Using the System

As mentioned in the requirement analysis, the system has three major parts. Below you see a brief description about how the three parts function.

## 5.1 Login Page

The home page of the system has the login form and the description about the project. You see register and login buttons on the top of the page. Login page is the first and foremost page through which user can enter into the system.

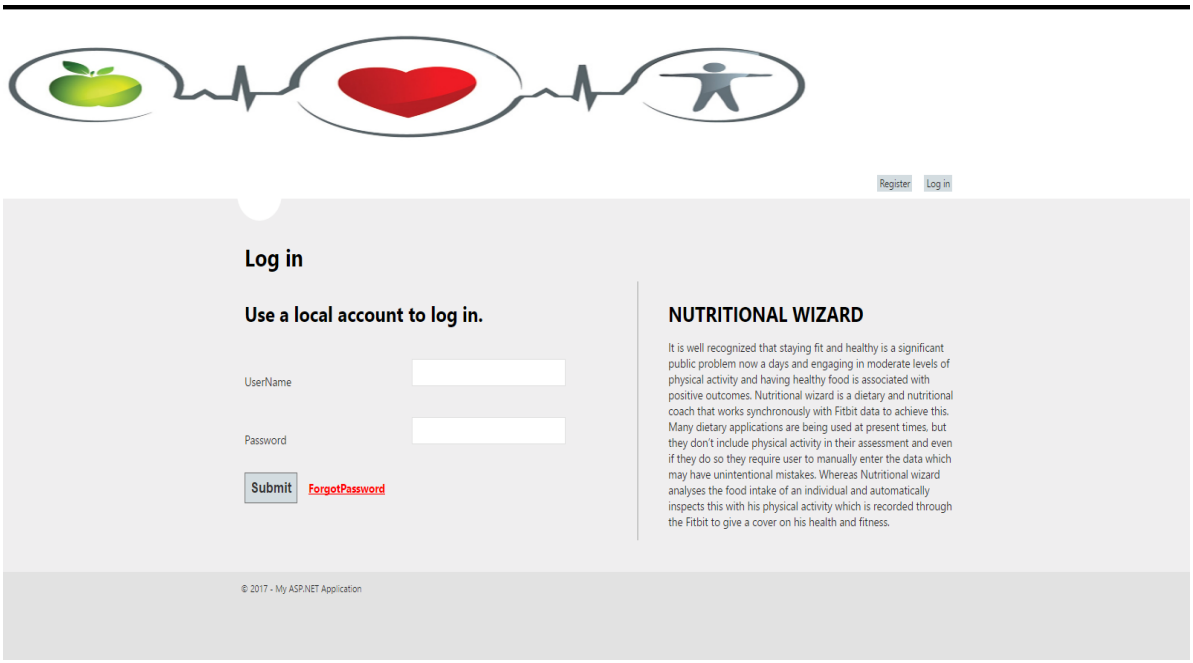


**Figure 5-1 Login page**

### 5.1.1 Login page validation:

When users enter wrong credentials they should not be able to log in and should pop up a message saying invalid user.

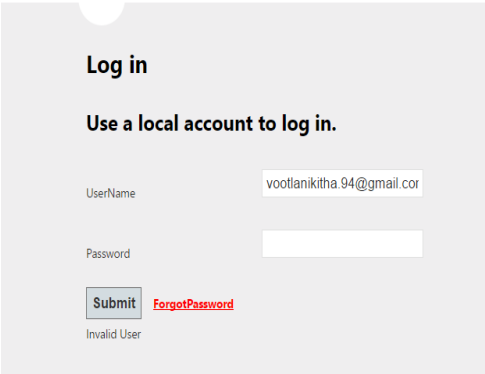


**Figure 5-2 Invalid User Details**

## 5.2 Registration Page

User can register into the application with filling in his details of Name, Username, Password, Height and Weight. When user gives up wrong format of data for the any of the fields in the registration, an error message pops up giving them the guidelines that should be followed for a proper registration. The registration page and its validations are shown as follow

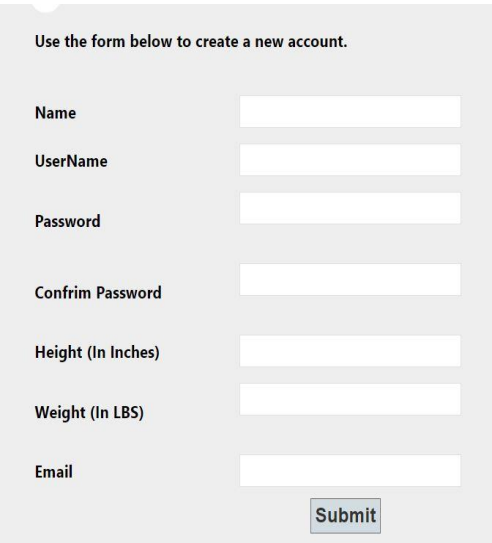


**Figure 5-3 Registration Page**

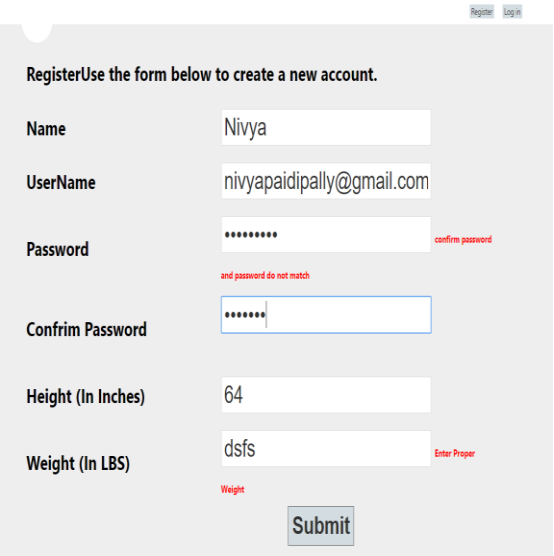


**Figure 5-4 Registration Validation Page**

## 5.3 Forgot Password Page

When user forgets his password, he can reset it from this page. User gives his email that he has set during registration and an email is sent with his present password and username is sent. After that he can go to forgot password link to give his new password after entering old password.


**Figure 5-5 Forgot Password Page**

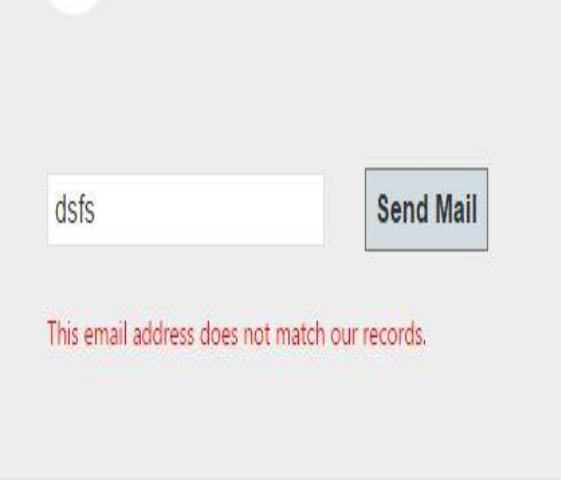


**Figure 5-6 Invalid Email**

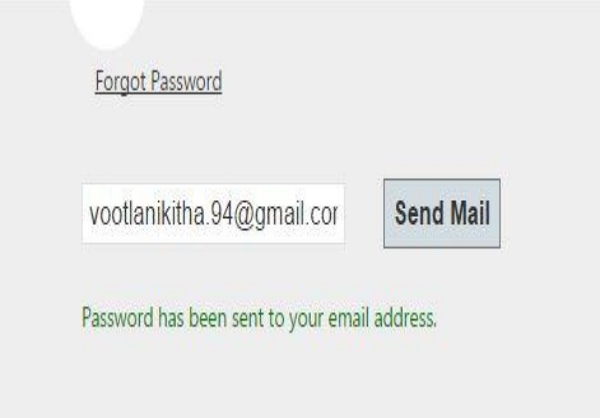


**Figure 5-7 Valid Email to send mail**

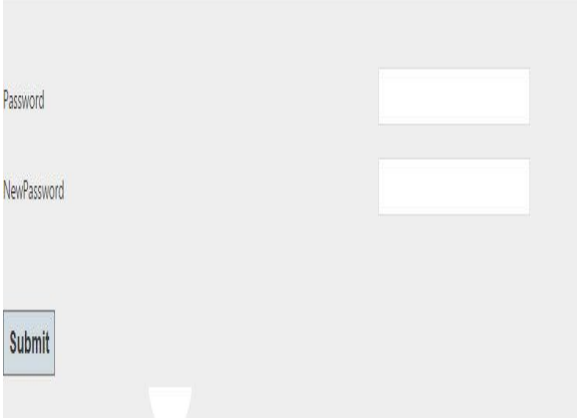


**Figure 5-8 new password update**

## 5.4 User Page

After successful login, user is navigated to the user home page where he can have the options to go to input food part or profile part or recipes part. He can also logout from this page if he wants to. User homepage looks as follows.

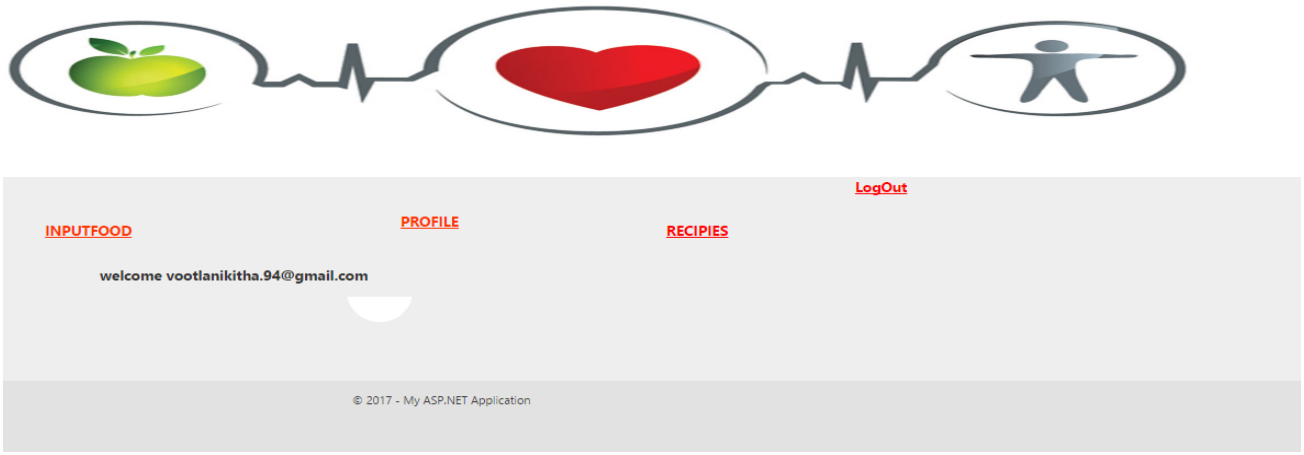


**Figure 5-9 User Home Page**

## 5.5 Input Food Page

Input food page looks like

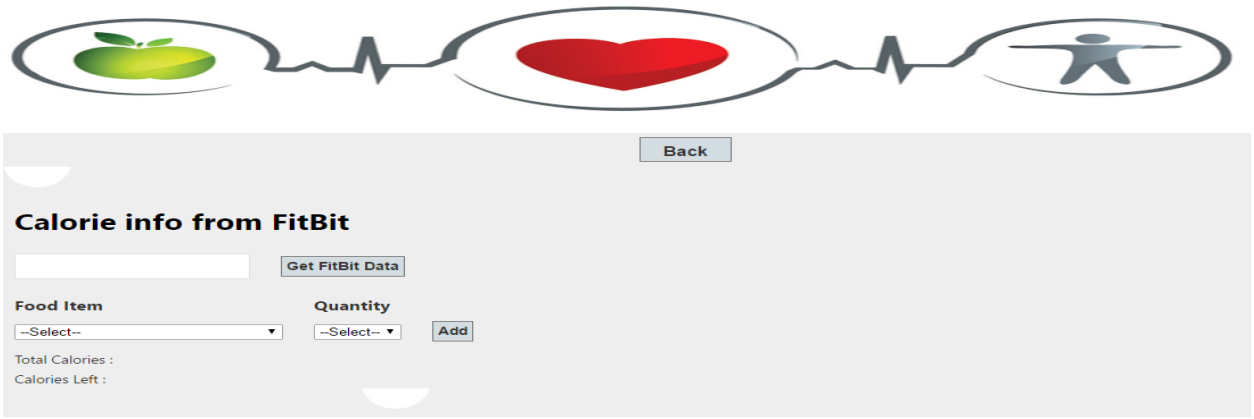


**Figure 5-10 Input Food Page**

### 5.5.1. Fit bit data receiver

In input food part you first need to enter which date's food details you want to check and then check Fitbit steps on that day. On checking Fitbit steps, it shows how many steps user took on that day and for that activity how many calories are consumed. If date is not selected, then page shows a pop up message saying that date should be selected.

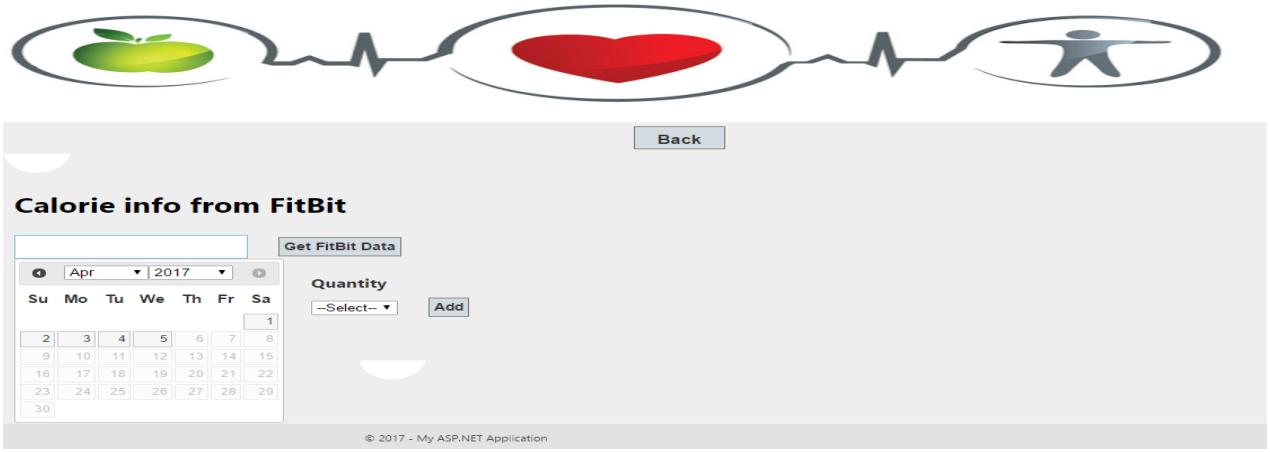


**Figure 5-11 Input Food Page Fitbit Date Selection**

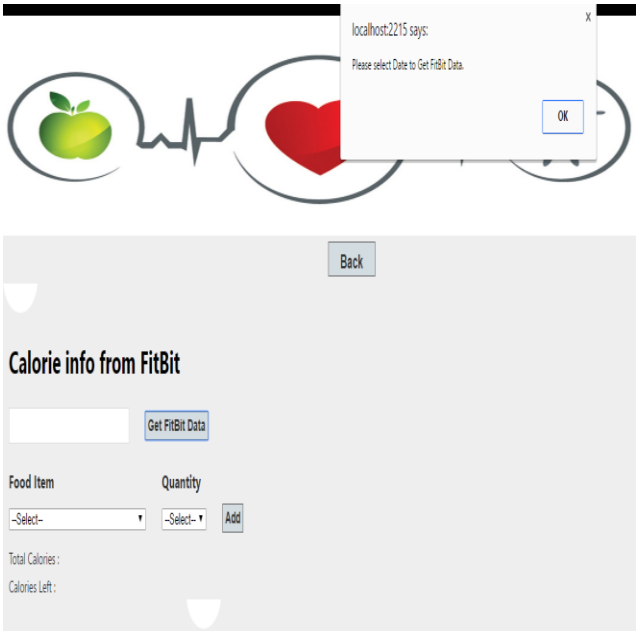


**Figure 5-12 Invalid Fitbit Date Selection**


**Figure 5-13 Valid Fitbit Date Selection**

### 5.5.2 Food Item Selector.

After getting step count from Fitbit, user needs to add the details of food that he had on that particular selected date. From list of food items saved, user selects all the food items he/she had and then the serving quantity. After the selection user can see the total no.of calories he/she had and total calories that are saved in the body

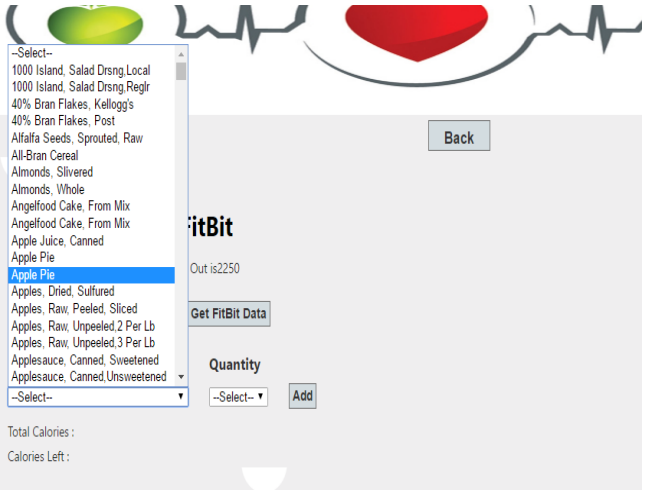


**Figure 5-14 Food Item Selection**

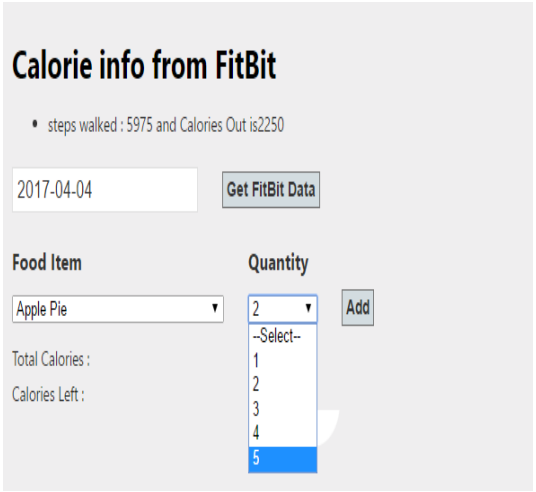


**Figure 5-15 Food Item Quantity Selection**

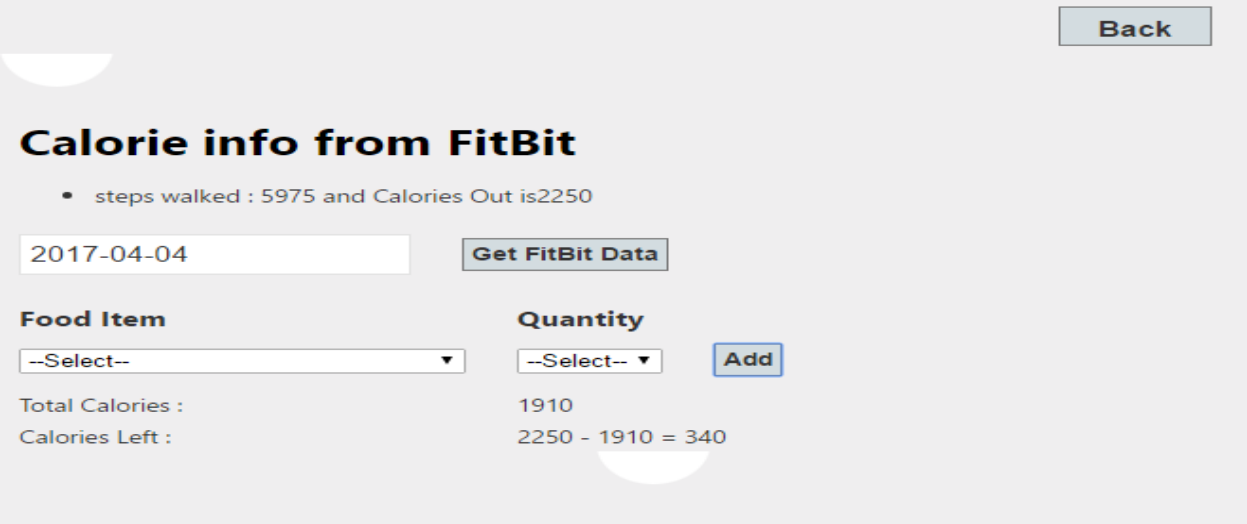


**Figure 5-16 Valid Total Input Food Page**

## 5.6 Profile Page

When user gets into profile page, he sees his height and weight given during registration and he can now enter his present weight and know his BMI and can also check his obese level from the chart displayed.

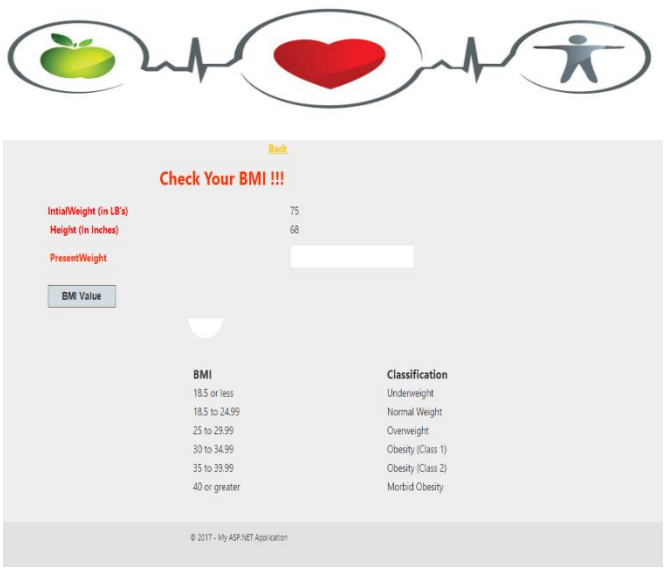


**Figure 5-17 Profile Page**

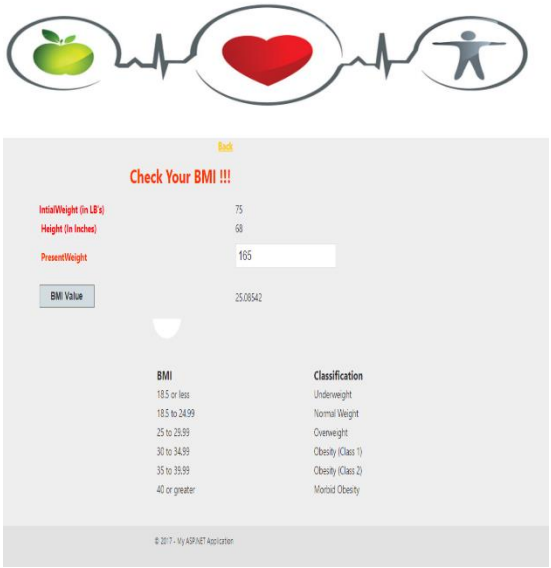


**Figure 5-18 Profile page displaying BMI**

## 5.7 Recipes page

As user gets into recipes page, users finds three categories that he can choose low fat, low carbohydrates and low protein.

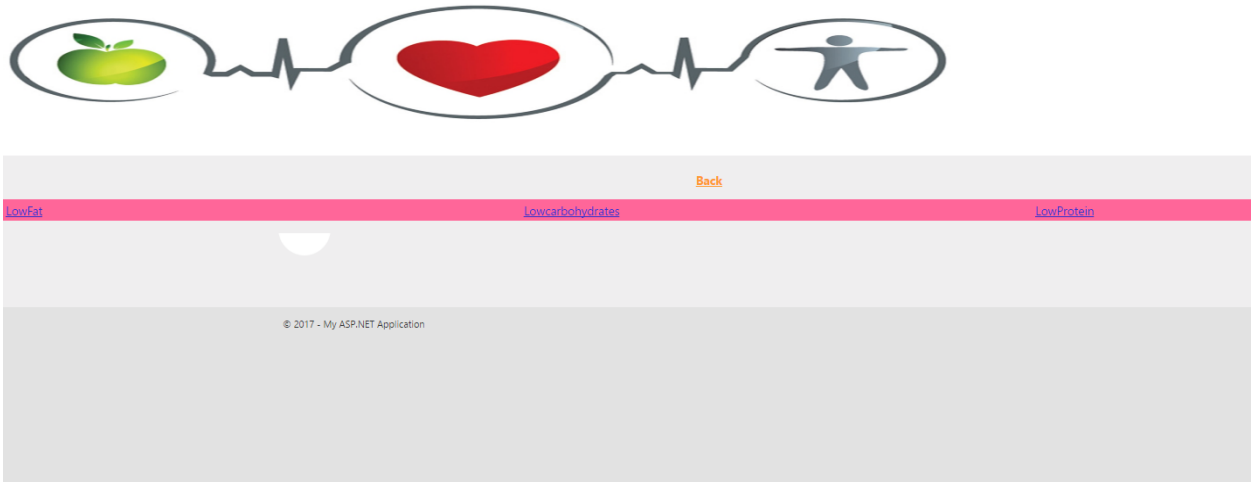


**Figure 5-19 Recipes Page**

### 5.7.1 Low Carbohydrates

Here the user finds all the recipes of dishes that have a very low carb content. On clicking any recipe, user gets, name, recipe, carb, protein, fat content and total calories of the food item.

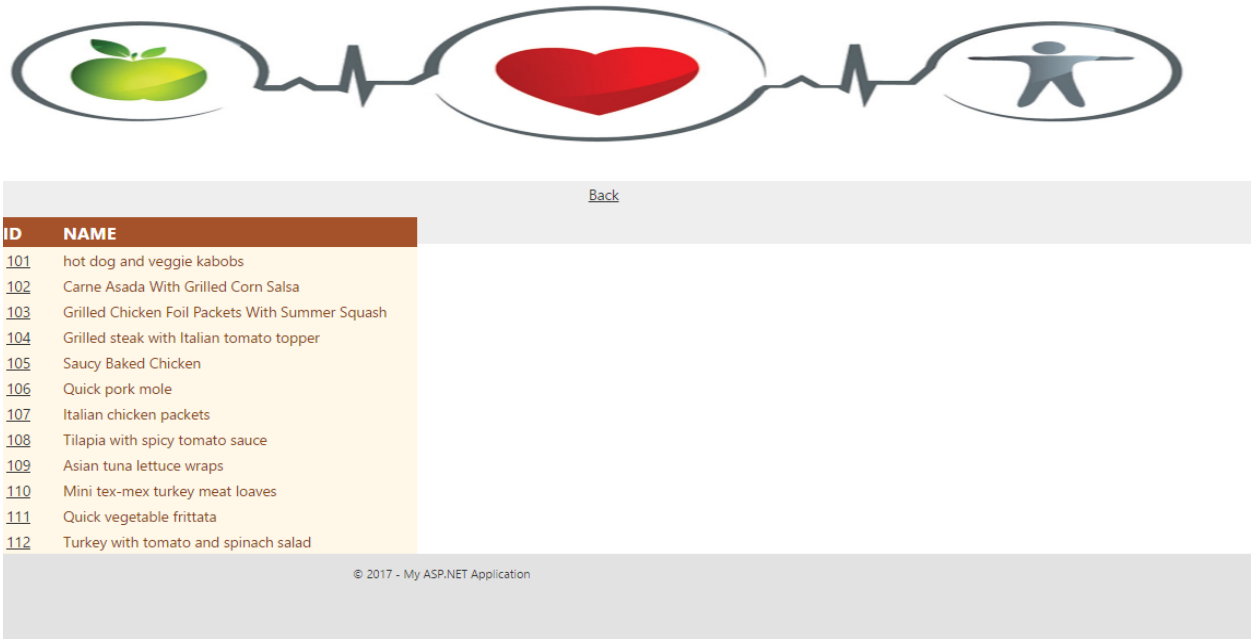


**Figure 5-20 Low Carbohydrates Recipe List Page**

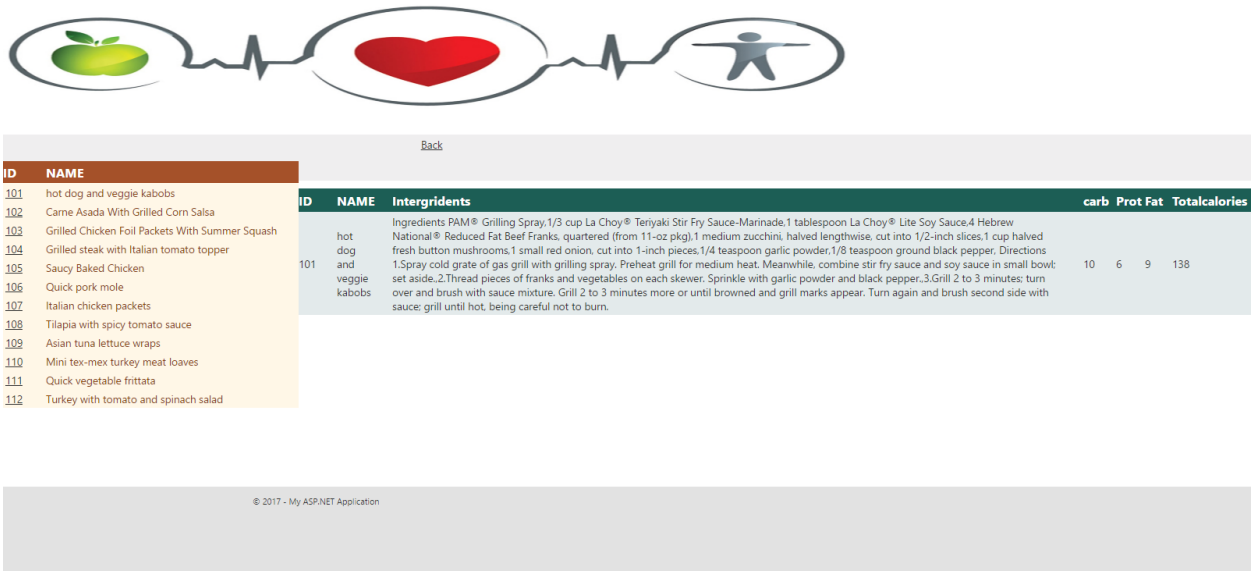


**Figure 5-21 Low Carbohydrates Recipe Page**

### 5.7.2 Low Fat

Here the user finds all the recipes of dishes that have a very low fat content. On clicking any recipe, user gets, name, recipe, carb, protein, fat content and total calories of the food item.

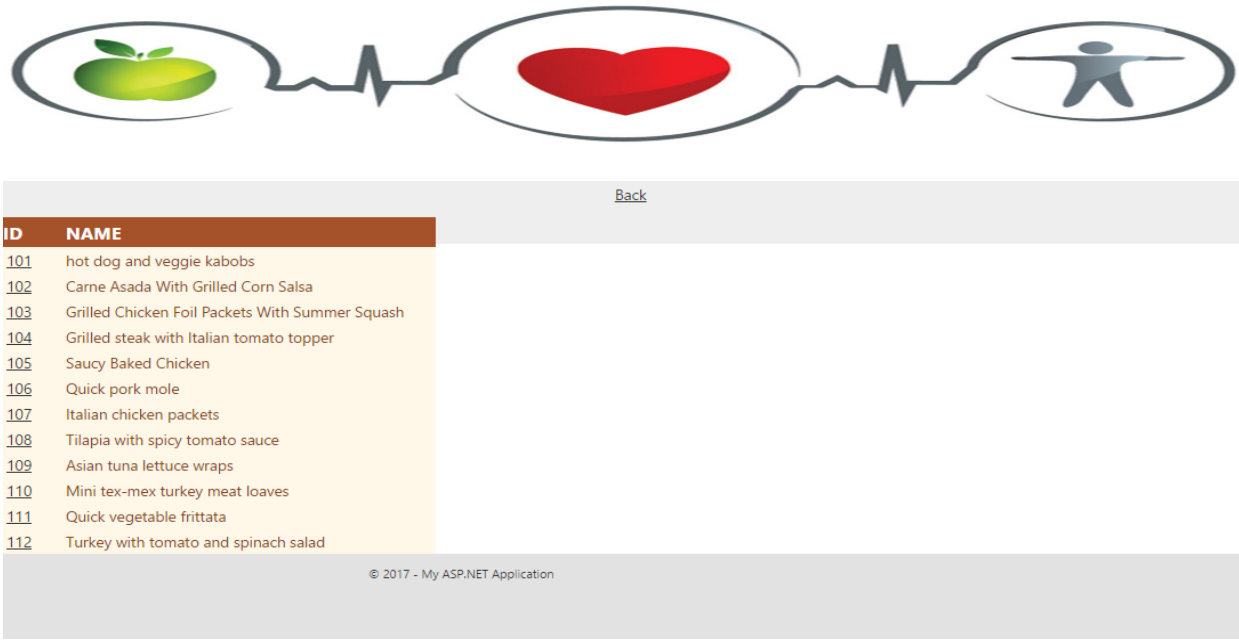


**Figure 5-22 Low Fat Recipe List Page**

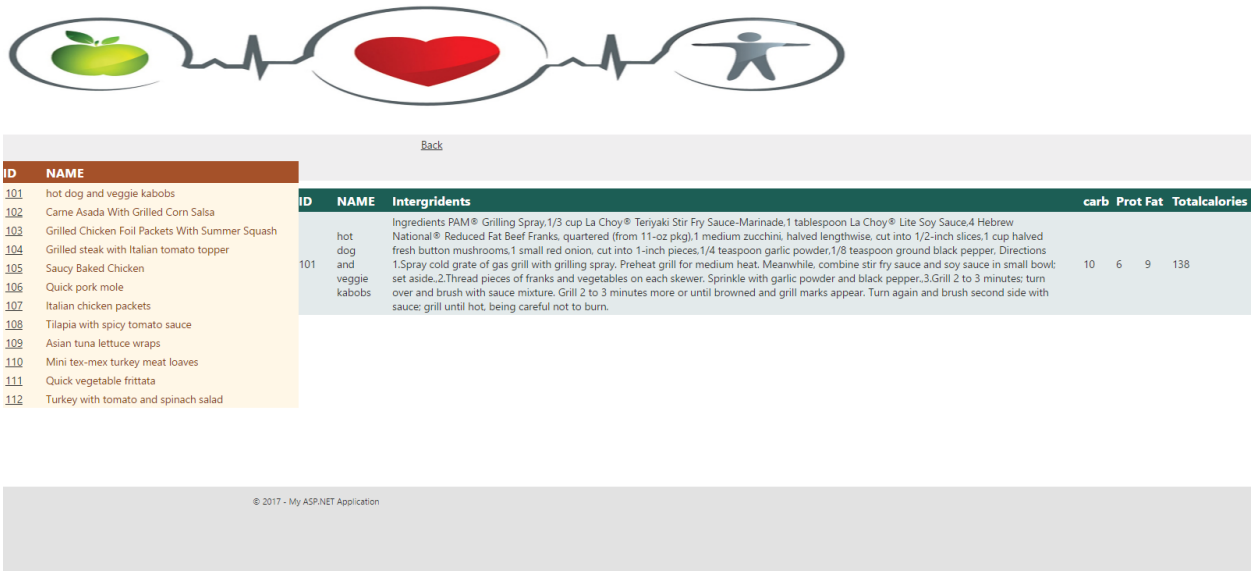


**Figure 5-23 Low Fat Recipe Page**

### 5.7.3 Low Protein

Here the user finds all the recipes of dishes that have a very low protein content. On clicking any recipe, user gets, name, recipe, carb, protein, fat content and total calories of the food item.

Back

| ID | NAME |
|---|---|
| 101 | hot dog and veggie kabobs |
| 102 | Carne Asada With Grilled Corn Salsa |
| 103 | Grilled Chicken Foil Packets With Summer Squash |
| 104 | Grilled steak with Italian tomato topper |
| 105 | Saucy Baked Chicken |
| 106 | Quick pork mole |
| 107 | Italian chicken packets |
| 108 | Tilapia with spicy tomato sauce |
| 109 | Asian tuna lettuce wraps |
| 110 | Mini tex-mex turkey meat loaves |
| 111 | Quick vegetable frittata |
| 112 | Turkey with tomato and spinach salad |

© 2017 - My ASP.NET Application

**Figure 5-24 Low Protein Recipe Page**

Back

| ID | NAME |
|---|---|
| 101 | hot dog and veggie kabobs |
| 102 | Carne Asada With Grilled Corn Salsa |
| 103 | Grilled Chicken Foil Packets With Summer Squash |
| 104 | Grilled steak with Italian tomato topper |
| 105 | Saucy Baked Chicken |
| 106 | Quick pork mole |
| 107 | Italian chicken packets |
| 108 | Tilapia with spicy tomato sauce |
| 109 | Asian tuna lettuce wraps |
| 110 | Mini tex-mex turkey meat loaves |
| 111 | Quick vegetable frittata |
| 112 | Turkey with tomato and spinach salad |

| ID | NAME | Intergridents | carb | Prot | Fat | Totalcalories |
|---|---|---|---|---|---|---|
| 101 | hot dog and veggie kabobs | Ingredients PAM® Grilling Spray,1/3 cup La Choy® Teriyaki Stir Fry Sauce-Marinade,1 tablespoon La Choy® Lite Soy Sauce,4 Hebrew National® Reduced Fat Beef Franks, quartered (from 11-oz pkg),1 medium zucchini, halved lengthwise, cut into 1/2-inch slices,1 cup halved fresh button mushrooms,1 small red onion, cut into 1-inch pieces,1/4 teaspoon garlic powder,1/8 teaspoon ground black pepper, Directions 1.Spray cold grate of gas grill with grilling spray. Preheat grill for medium heat. Meanwhile, combine stir fry sauce and soy sauce in small bowl; set aside.,2.Thread pieces of franks and vegetables on each skewer. Sprinkle with garlic powder and black pepper.,3.Grill 2 to 3 minutes; turn over and brush with sauce mixture. Grill 2 to 3 minutes more or until browned and grill marks appear. Turn again and brush second side with sauce; grill until hot, being careful not to burn. | 10 | 6 | 9 | 138 |

© 2017 - My ASP.NET Application

**Figure 5-25 Low Protein Recipe Page**

# Chapter 6 - Security

Security is the most important attribute of any system. In the process of gaining customer confidence security plays a very vital role. Now a days we see almost all the websites asking for user information for better understanding and to serve better. On getting this user information and saving them into servers, it is the organization's responsibility to confirm that customer's data is safe and secure. The vast majority of developers now a days focus on building new and flashy features that can impress their end users and managers. Security can take a backstage in initial stages but it definitely comes back to bite when you least expect it. So, in this project, I made sure that I would reduce maximum possible security threats possible. Implementing security in a web page mainly has following four aspects:

- **Authentication**: It is the process of ensuring the user's identity and authenticity. ASP.NET allows four types of authentications:
    - Windows Authentication
    - Forms Authentication
    - Passport Authentication
    - Custom Authentication
- **Authorization**: It is the process of defining and allotting specific roles to specific users.
- **Confidentiality**: It involves encrypting the channel between the client browser and the web server.
- **Integrity**: It involves maintaining the integrity of data. For example, implementing digital signature.

Apart from these four aspects that ASP.Net provides this application also has sustainability for other threats listed below.

- **SQL Injection**:

  SQL Injection is an attack used to inject unintended SQL commands (statements) in a database by accepting malicious, unsecured, un-validated user input. Injected SQL commands can alter SQL statement and compromise the security of a web application. Methods of SQL exploits are either by input boxes or query strings. Exploiting input boxes is giving a malicious SQL command in the input boxes to get an unfair access. Exploiting through query strings is getting the string value from URL and adding a SQL command to it to again get an unfair access to our data. In both the above approaches, the query is concatenated with user input and the user input is not validating properly. So the attacker takes advantage of it and concatenates the malicious query with it and attacker can get the passwords. In order to prevent this SQL injection, we can follow two approaches,

  1. *Validate User Input*: Almost all of the user input fields are validated so that they don't accept any SQL commands as input.
  2. *Parameterized SQL query and stored procedure*: Parameterized queries do proper substitution of arguments prior to running the SQL query. It completely removes the possibility of "dirty" input changing the meaning of your query, with parameterized queries, in addition to general injection, you get all the data types handled.

- Password is encrypted in the database because of which even if the hacker gets access to data, he/she will not be able to understand and use the data.
- There are firewalls present in every page we load to block unwanted HTTP traffic to and from web application.

# Chapter 7 - Testing

Software testing is the process of validating and verifying that a software program or application is working as expected and whether it meets the business and technical requirements that guided its design and development. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Testing should be introduce in the early stage of the SDLC, The cost of fixing the bug is larger if testing is not done in early stage & bugs found in later stages. It is difficult to determine when to stop testing, as testing is a never-ending process and no one can claim that a software is 100% tested. So it is suggestible to stop testing when bug rate becomes almost nullified. For this project, unit, integration and compatibility testing are performed.

## 7.1 Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. In this case, unit testing is performed on the web site deployed in local host in google chrome browser. Following table shows the operations conducted to ensure proper functionality

| S.No | Test Case | Expected Result | Result |
|---|---|---|---|
| 1 | Open Home Page | Home page should open | Pass |
| 2 | User should be able to log in with right credentials | User should successfully login | Pass |

| | | | |
|---|---|---|---|
| 3 | User should not be able to log in with wrong credentials | User should not Login | Pass |
| 4 | User should be able to register with valid details | User should be successfully registered | Pass |
| 5 | User should not be able to register with invalid details | User should not be registered | Pass |
| 6 | Open Input food | Input food page should open | Pass |
| 7 | On selecting food items and quantity user should see calories consumed | User should see calories consumed | Pass |
| 8 | On not selecting quantity food item should not be allowed to add | Dialog box should pop up asking for quantity field | Pass |
| 9 | On not selecting food item, quantity should not be added | Dialog box should pop up asking for food item field | Pass |
| 10 | Open profile page | Profile page should open | Pass |
| 11 | On opening profile page, user height and weight should be displayed | User height and weight are displayed in profile page | Pass |
| 12 | On clicking BMI, bmi should be displayed | BMI should be displayed | Pass |
| 13 | Open recipes page | Recipes page should be loaded | Pass |
| 14 | Open low carbohydrate page | Low carbohydrate page should be loaded | Pass |
| 15 | Open low fat page | Low fat recipes page should open | Pass |

| | | | |
|---|---|---|---|
| 16 | Open low protein page | Low protein recipes page should open | Pass |
| 17 | On clicking on ID of low carbohydrate recipe, details of the recipe should open | Details of the selected recipe should load | Pass |
| 18 | On clicking on ID of low protein recipe, details of the recipe should open | Details of the selected recipe should load | Pass |
| 19 | On clicking ID of low fat recipe, details of the recipe should open | Details of the selected recipe should load | Pass |
| 20 | On clicking on Logout user should log out | User should log out | Pass |
| 21 | Open forgot password page | Forgot password page should open | Pass |
| 22 | User should not be able to update password for invalid username | User should not update password | Pass |
| 23 | User should be able to update password for valid username | User should be able to update password | Pass |

**Table 7-1 Unit Testing for Application**

## 7.2 Integration Testing

Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing. The purpose of integration testing is to verify the functional, performance,

and reliability between the modules that are integrated. Following table shows operations and tests conducted on integrating Fitbit module to our web application

| S.No | Test Case | Expected Result | Result |
|---|---|---|---|
| 1 | On clicking register button on home page, register page should load | Register page should load | Pass |
| 2 | On successful login user should go to user page | User page should load | Pass |
| 3 | On clicking on input food button, input food page should load | Input food page should load | Pass |
| 4 | On clicking on profile button, profile page should load | Profile page should load | Pass |
| 5 | On clicking on Recipes button, recipes page should load | Recipes page should load | Pass |
| 6 | On input food page, after selecting a valid date, no. of fit bit steps should appear | On valid date input, steps taken on that date should load | Pass |
| 7 | On loading step count and calories consumed, calories stored should be visible | Calories stored are calculated and loaded. | Pass |
| 8 | On selecting new date in input food, new step count data of that date should be loaded | Updated step count should load | Pass |

**Table 7-2 Integration Testing**

## 7.3 Performance Testing

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability, and resource usage.

In this application, testing on CPU time and utilization is done and below are the results of the complete system being run at different conditions.

### 7.3.1 CPU time for the complete system:

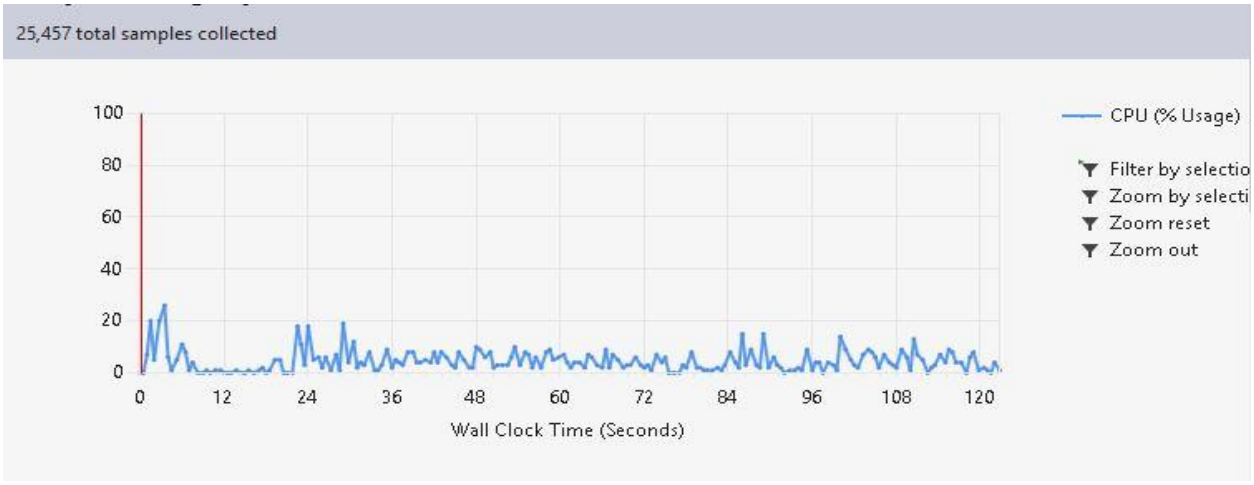


**Figure 7-1 CPU usage for whole system**

The above figure shows the CPU usage for the complete system and we can see that the second highest usage is at 24 seconds and 30 seconds i.e. when the Fitbit call is done.

### 7.3.2 CPU time for input food module:

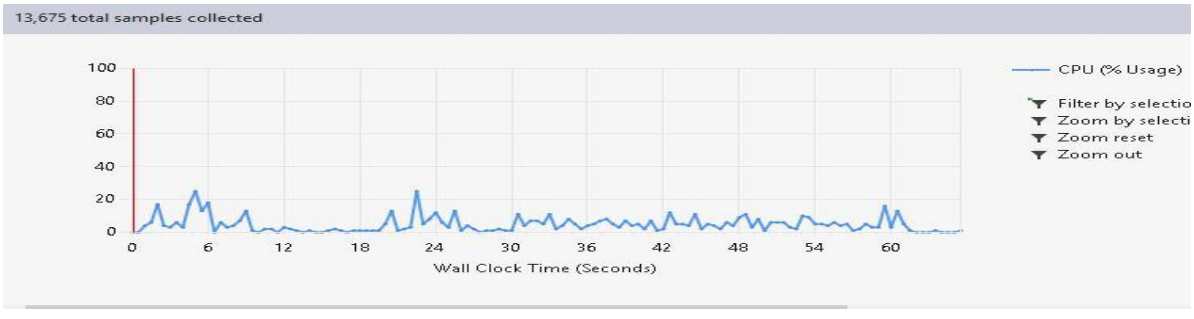


**Figure 7-2 CPU usage for Input food module**

The above figure shows the CPU usage for the complete system and we can see that the second highest usage is at 24 seconds and 30 seconds i.e. when the Fitbit call is done.

### 7.3.3 CPU time for Profile Part

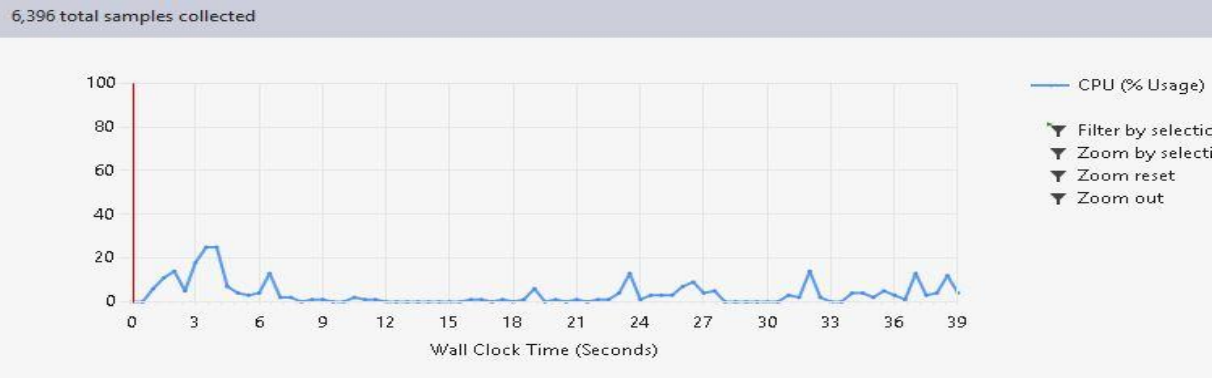


**Figure 7-3 CPU Usage for Profile part**

The above figure shows the CPU usage for the profile module run independently and most of the CPU utilization is done only in first 3 to 6 seconds when data is being retrieved from database.

### 7.3.4 CPU time for Recipes module

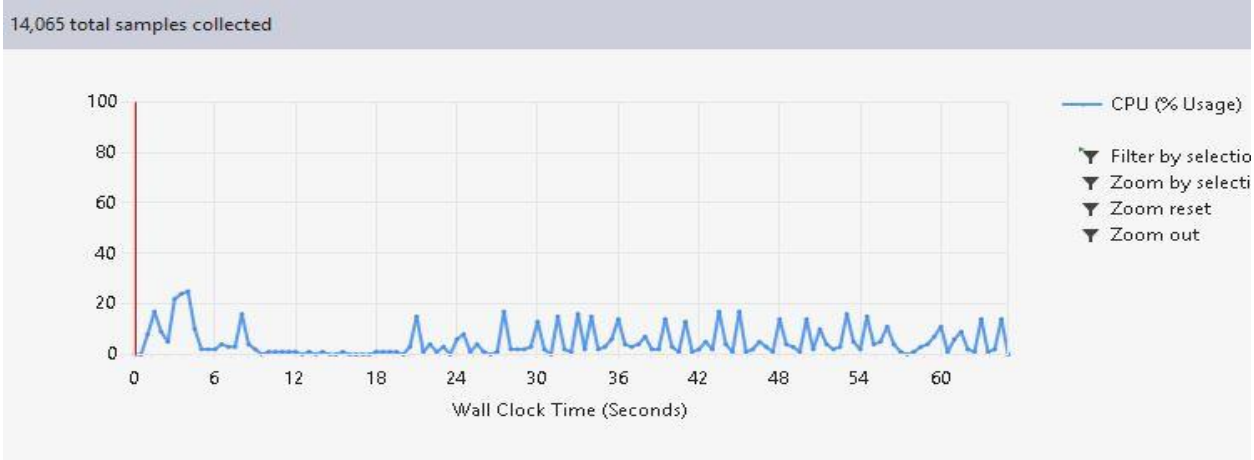


**Figure 7-4 CPU usage for recipes part**

Here as there is much retrieving of data from database in this module, we see all of the parts are equally utilizing CPU.

# Chapter 8 - Conclusion and Future Work

## 8.1 Conclusion

In conclusion, this system would be of great help for users wants to concentrate on their either weight increase or reduction plan. Any person who is unhappy on their body relative mass will find this project interesting as it developed including features that are needed from a user perspective. This application is apt for today's busy world where everything has to be automated.

Personally while developing this application, I mainly go to learn how to integrate an already existing API services into an application. I learnt technologies like ASP.Net and JQuery. I also learnt how important is getting requirement analysis done perfectly beforehand.

## 8.2 Future Work

The Application can be improved further by including an Admin who is responsible for creating new recipes and adding food items into the list rather than adding them directly to database. It can be also improved by adding the graphical analysis part where user gets data from a selected time frame. A mobile application developed would also be handier.

# Chapter 9 - References


[1]. (n.d.). Retrieved from Tutorials Point: https://www.tutorialspoint.com/uml/uml_class_diagram.htm

[2]. Certification, I. E. (2017, 03 25). *What is software testing*. Retrieved from istqbeamcertification.com: http://istqbexamcertification.com/what-is-a-software-testing/

[3]. Fitbit. (2017, 02 12). *Fitbit devoloper API*. Retrieved from FitBit : https://dev.fitbit.com/docs/oauth2/

[4]. security, T. P. (2017). *ASP.Net security*. Retrieved from Tutorials Point: https://www.tutorialspoint.com/asp.net/asp.net_security.htm

[5]. Testing, T. P. (2017). *Integration Testing*. Retrieved from Tutorials Point: https://www.tutorialspoint.com/software_testing_dictionary/integration_testing.htm

[6]. Testing, T. P. (2017). *Performance Testing*. Retrieved from Tutorials Point: https://www.tutorialspoint.com/software_testing_dictionary/performance_testing.htm

[7]. Testing, T. P. (2017). *Software Testing Overview*. Retrieved from Tutorials Point: https://www.tutorialspoint.com/software_testing/software_testing_overview.htm

[8]. Testing, T. P. (2017). *Unit Testing*. Retrieved from Tutorials point: https://www.tutorialspoint.com/software_testing_dictionary/unit_testing.htm

[9]. TutorialsPoint. (2017, 04 04). Retrieved from www.tutorialspoint.com: https://www.tutorialspoint.com/uml/uml_class_diagram.htm

[10]. TutorialsPoint. (2017, 04 04). *Class Diagram*. Retrieved from Tutorials Point: https://www.tutorialspoint.com/uml/uml_class_diagram.htm

[11]. UseCase, T. P. (2017, 04 04). *Tutorials Point*. Retrieved from https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm

[12]. WordPress, & Atahulpa. (2012, 06 01). *Software Teting Class*. Retrieved from http://www.softwaretestingclass.com/importance-of-testing/