Group 14

Don’t Eat That!

15th October 2016

Introduction:

We plan to build Don’t Eat That! as a web application that both registered and general users can use to compare two food products and determine which choice is the healthiest for the user. The web application will use a basic algorithm that determines the healthiest choice for users without health concerns, or a specialized algorithm that determines the healthiest choice for users with dietary restrictions.

This document will attempt to explain the architecture and design of the system using the 4+1 architecture model.

System Architecture and Rationale:

This is a web application, so various languages and software platforms will be used. The application will follow a “Client-Server” architecture. We chose this architectural pattern because it resembles the type of interactions seen in our use cases. The front end is written in HTML and CSS, and also uses JavaScript and the Bootstrap framework. Server side code is used to link the front end to the databases (back end). The server side language is PHP.

The web application consists of the following major components.

1) User database used to create, manage, and access user accounts
2) Food database used to create, manage, and access food products and their corresponding nutritional information
3) Home page on which a user submits two food products to be compared
4) Account page on which users will create and manage their accounts
5) Recommendation page of which users are informed of the web application’s recommended choice

<u>Functional Requirements:</u>

As a web application, Don’t Eat That! should work on multiple operating systems and browsers to increase the accessibility of the application. As food recommendations can affect a user’s health, it is important that the food database returns the correct nutritional facts and that the users database stores user information (such as health conditions) correctly. Therefore, a reliable database system is needed.

The user's experience is also of utmost concern. The website should not take longer than 5 seconds to display an answer to a user request, as that would be an uncomfortably long wait. We will make sure our PHP code is optimized, and we will not make any unnecessary/redundant database queries. The databases will also be optimized so queries are performed as fast as possible.

As our databases store user information, the web application has to be secure. We will avoid directly using passwords in our PHP scripts, and instead reference an environment variable which contains the password. User passwords will not be directly stored in the database, but instead will be encrypted using a secure hash algorithm. Finally, we will have to protect against SQL injection. If a hacker types a malicious SQL query into the search box, we will have to detect this and discard the query.

HTML (HyperText Markup Language) & CSS (Cascading Style Sheets)

The user interfaces are written in HTML, the standard markup language for creating web pages and applications. HTML provides the framework of the user interfaces as and HTML elements are used as the building blocks. HTML describes the structure and layout of a web page by using tags and attributes. CSS describes how HTML elements are displayed on screen. CSS can be used to separate document content from presentation. This separation can improve content accessibility and provide flexibility and control in presentation characteristics. We chose to use HTML and CSS because these languages are commonly to make web pages, making it easy to find online reference/support. In addition, CSS and HTML are powerful languages and can be used to create dynamic, visually appealing, and fast webpages.

MySQL

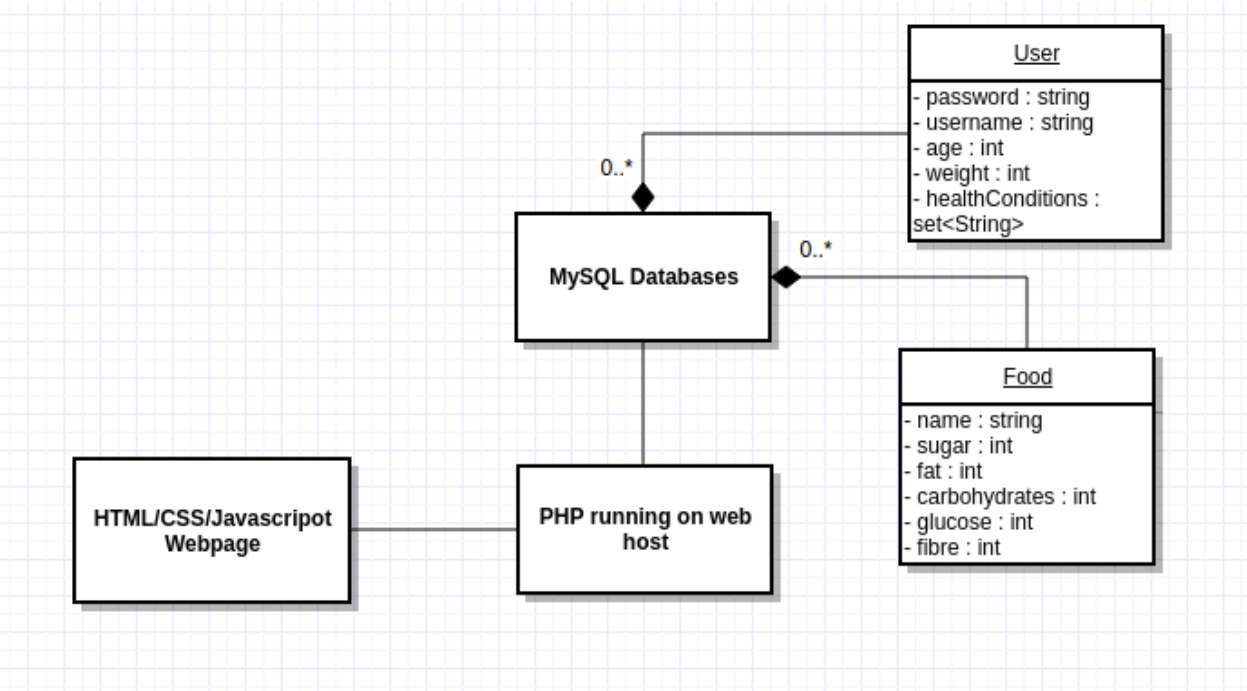
The web application is hosted by GoDaddy.com Inc., an Internet domain registrar and web hosting company. GoDaddy provides web domains with database services using the MySQL database system. MySQL is a database system that runs on a server and uses standard SQL. We chose to use a domain host with MySQL because MySQL is free to use, compiles on multiple platforms, works for applications of all sizes, and has earned a reputation for reliability. These attributes allow our web application to work on multiple operating systems and browsers,

PHP (Hypertext Preprocessor)

PHP is an open source, general purpose scripting language that can be embedded into HTML. PHP code is executed by the server (back end), which generates HTML and sends it to the client (front end). Within the HTML code of the user interfaces, PHP code can be embedded, allowing the user interface pages to interact with the databases. For example, a user can enter two food products and PHP code can be used to retrieve information about the food products from the food database. We chose to use PHP because PHP supports a wide range of databases, can be used on multiple operating systems, and supports most browsers. PHP is also a commonly used server side language, so online help and support is easily found.

Data

The user interface will consist of webpages built using HTML, CSS and JavaScript. Upon user input, the webpages will query the Users and/or Foods databases. The results of the query will be returned to a webpage that displays the recommendations made by the web application.

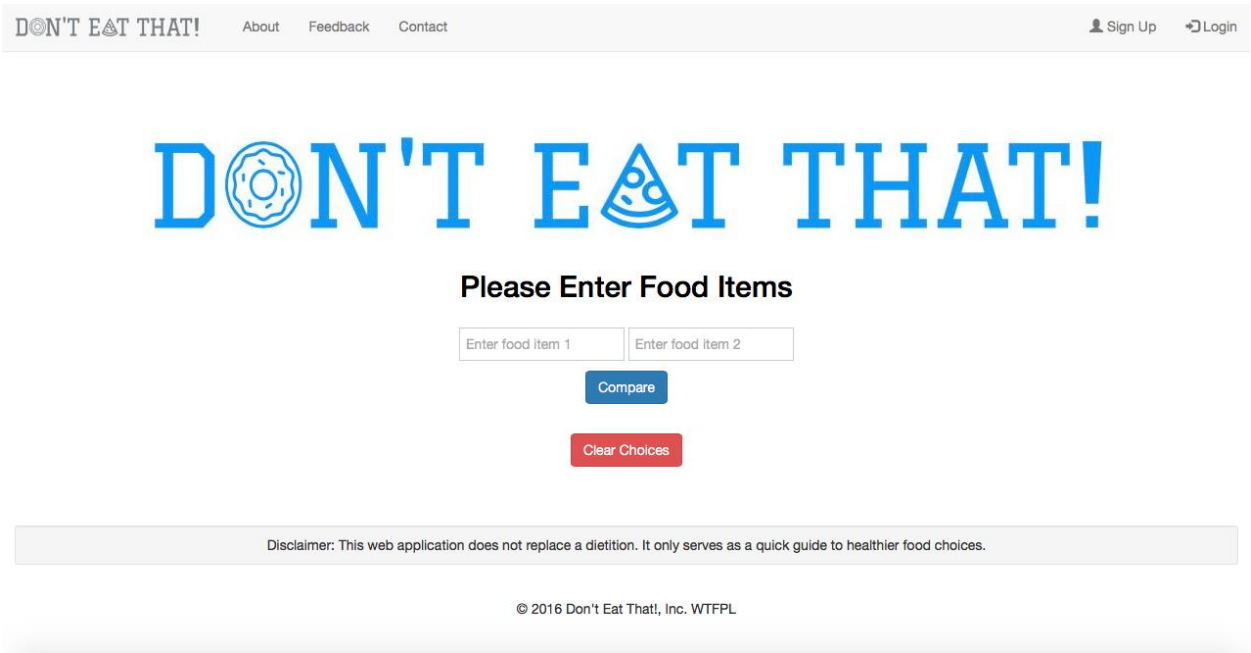


GUI

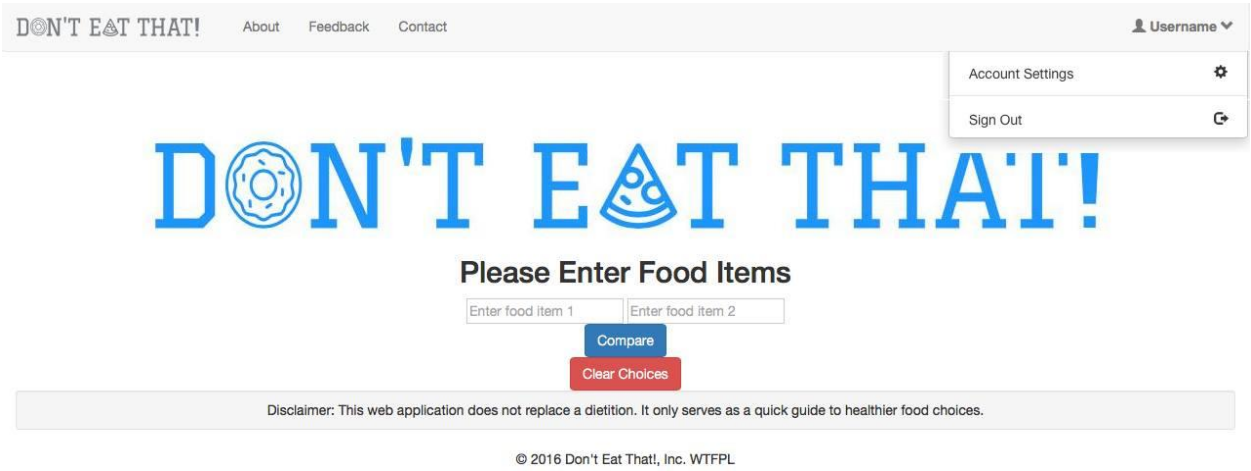
The user interface will be built using HTML, CSS, and the Bootstrap framework.

Home Page

The home page will be used by both registered and general users. In the centre of the homepage, the user will be able to enter in two food products for comparison. The user can clear the text boxes by using the “Clear Choices” button. At the top of the home page, the user can provide feedback about the web application, log into their account, or sign up for an account.

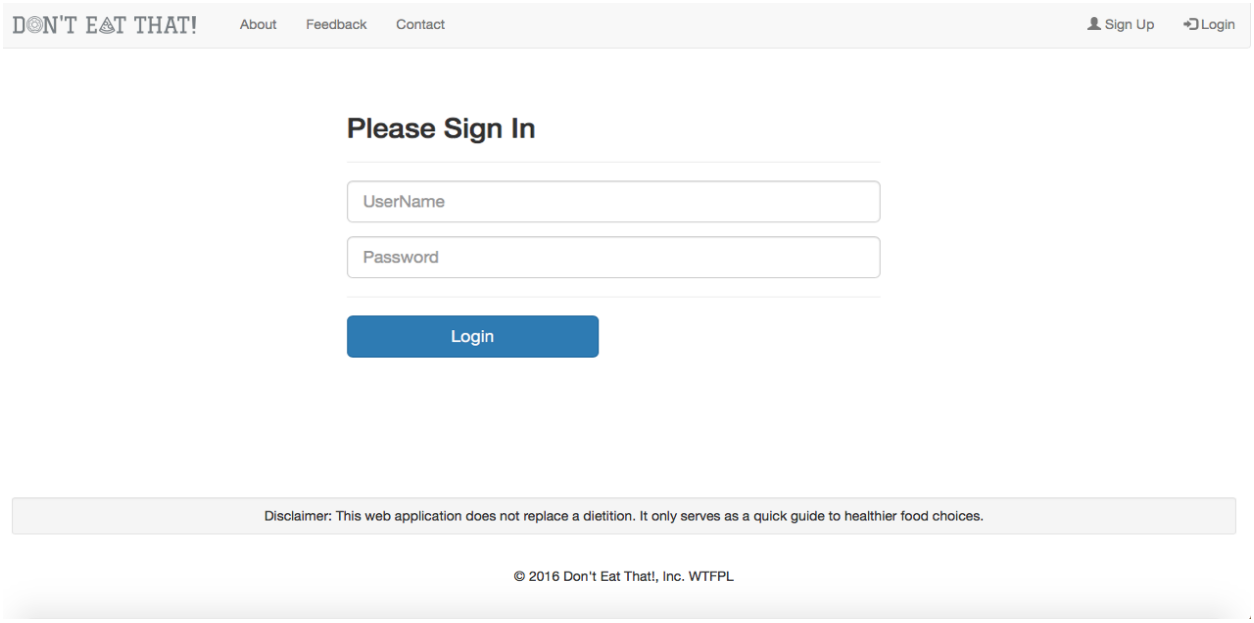


If the user has signed in, then the upper right corner of the home page will show the user’s username and allow the user to access account settings or sign out.



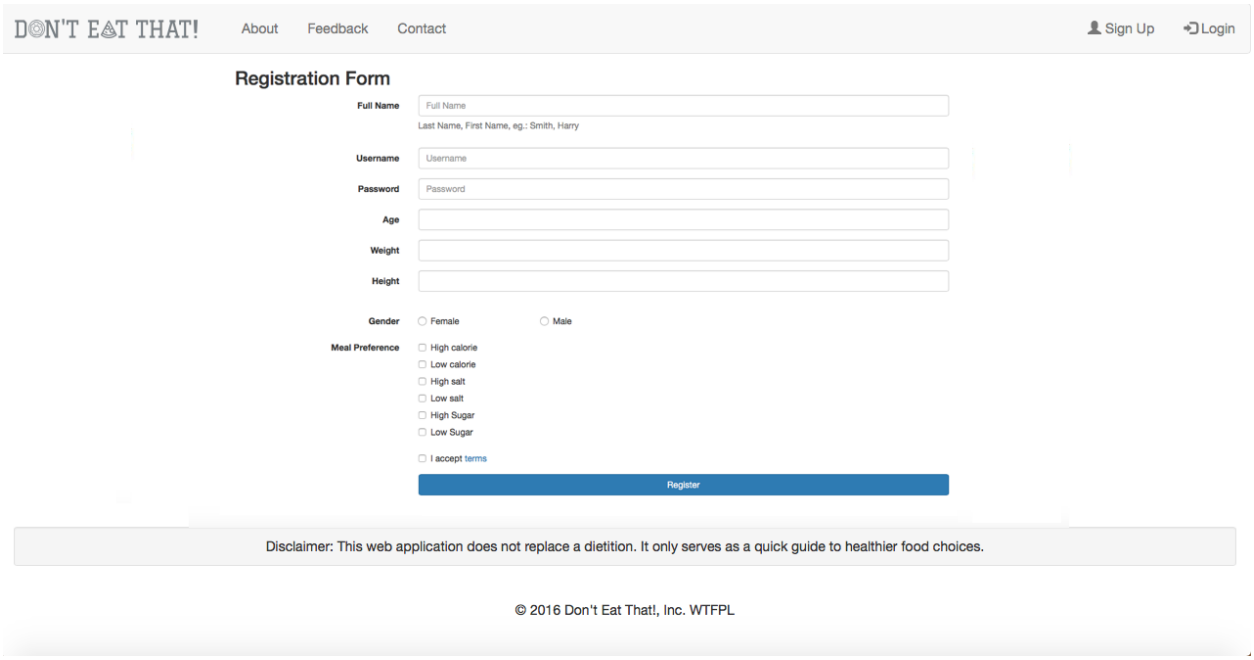
Login Page

The login page will be used by users to log into their account. The login page will have two textboxes: one for the user name and one for the password. After entering the information, the user will log in by pressing the “Login” button.



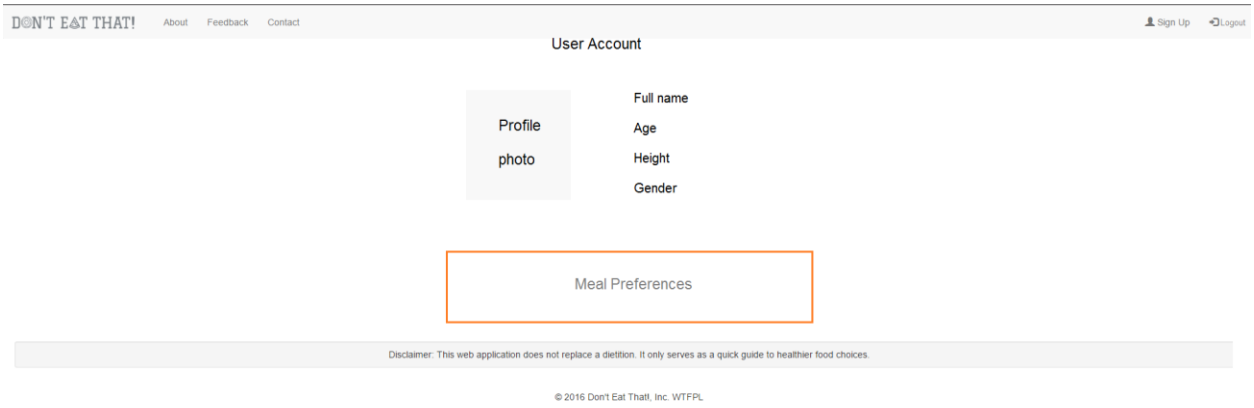
Account Sign-up Page

The account sign up page (registration form) will be used by general users to create new accounts. The registration form will consist of six textboxes: one for the full name, one for the username, one for the password, one for the age, and one for the weight, and one for the height. The form will also have two checkboxes: one for the gender and one for the meal preferences, which will provide the user with the at least six checkboxes for the different nutrition options. After entering the information and having done the registration, the user will be addressed to the main page where he/she will be able to log in to their newly created accounts by pressing the “Login” button.



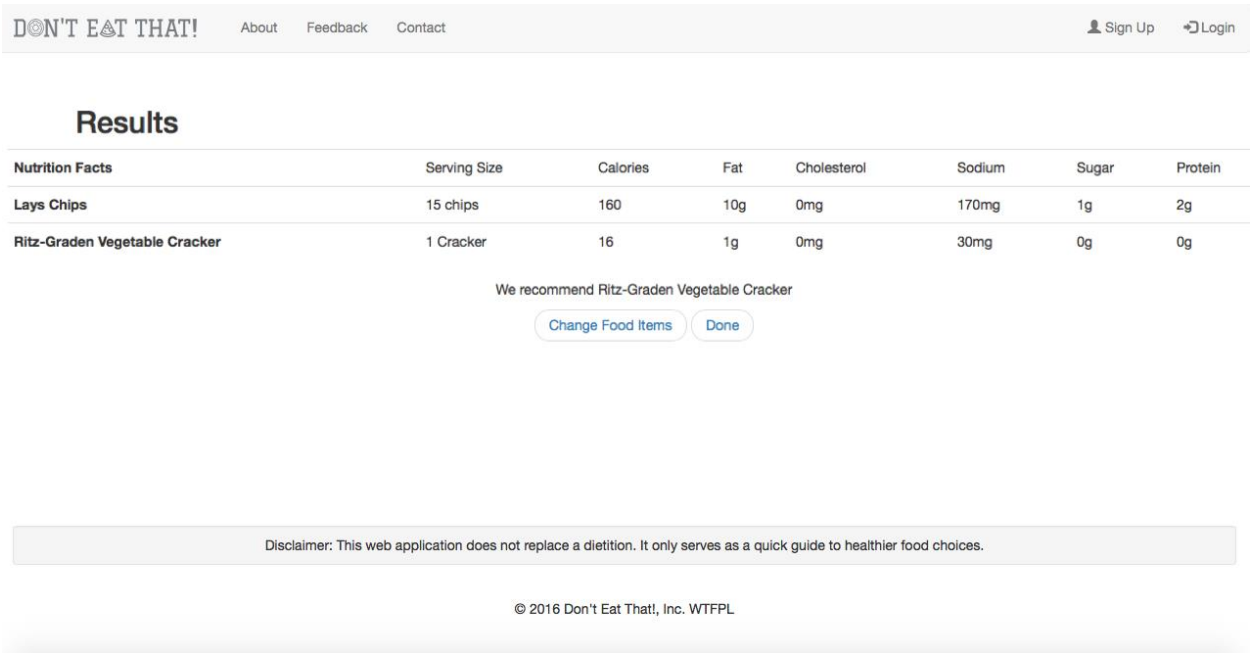
User Account View

User account page will have all the data from the registration form stored in the textboxes respectively. The user will be able to modify them and, based on preference, may or may not upload the profile picture. Once the user is done with all modifications, he/she may exit from the account by pressing “Logout” button, which will lead to home page.



Recommendation Page

The recommendation page is where users will see the proposed food product. The recommendation page will return the nutritional data of both foods and tell the user which one is the healthier of the two.



Validation

The user interface is reviewed with the client on a weekly basis. These reviews yielded some changes to our initial proposal of the user interface. This resulted in a more user-friendly interface. Likewise, the functional requirements are reviewed with the client on a weekly basis. The changes from these reviews are scheduled to be done in the upcoming iterations.