Software Design Specification for The Healthy Table

Version: 1.4 **Date: 5/2/2015**

Prepared by Shuaibo Gao

Table of Contents

Change Log

- ***Version 1.0 (03/24/2015): Document created. - Shuaibo Gao***
- ***Version 1.1 (03/26/2015): Added Chapter 1 & 2 – Introduction & Consideration. - Shuaibo Gao***
- ***Version 1.2 (03/28/2015): Added Chapter 3 & 4– Business Models. - Shuaibo Gao***
- ***Version 1.3 (03/31/2015): Updated Chapter 5– User Interfaces. - Shuaibo Gao***
- ***Version 1.4 (05/2/2015): Removed some feature. - Shuaibo Gao***

1. Introduction

1.1 System Overview

The Healthy Table serves as a platform to bridge restaurants with the public. It would also allow consumers to be able to monitor their everyday food combustion as well as are able to track how they have progressed over time. Restaurants can use the platform to promote their business and attract customers. On both ends of the system, a new dynamic is established to form a completely new eco-system. The system provide dinning recommendations for individual customer base on their health conditions and or special requirements which are customizable base on their preference. Orders can be made online via the system then updated among restaurants and customers. The payment can be paid via the system online or offline.

1.2 Definitions, Acronyms and Abbreviations

- **THT**: The Healthy Table, the system that is being discussed in this design specification.
- **CS**: Customer Service, a role that is in charge of processing customer related business
- **SM**: Sales Manager, a role that is in charge of processing restaurant related business
- **DR**: Diet Recommendation, THT make diet recommendation to customers
- **HDF**: Healthy Diet Formula, the formula that is used to calculate healthy diet
- **UML**: Unified Modeling Language

1.3 References

1.3.1. 3rd Party APIs

- Google Charts: https://developers.google.com/chart/
- Google Map API: https://developers.google.com/maps

1.3.2. 3rd Party Frameworks

- Spring: https://spring.io/
- MVC: http://spring.io/guides/gs/serving-web-content/
- Maven: https://maven.apache.org/

2. Design Considerations

2.1 Assumptions

2.1.1. System

The initial version of THT for this course will only support web service. Later, we will build smart phone application, so the THT will support the request from the app client. THT will also provide web services API for other websites in the future.

2.1.2. Registration

- Customer can register to the system directory without any approval. After customer registered the system, he or she will be able to sign in with the account directly.
- After restaurant registered to the system, it have to be approved by sales. During the pending time before sales approve new restaurants. The restaurant's user account is inactive and can't sign in to the system. After sales confirm the restaurant's certification (by some other ways, like go to the restaurant), sales should approve the restaurant so that the restaurant can sign in to the system.
- The other user account like sales, customer, finance and admin will be created by admin role directly.

2.1.3. Other assumptions

- Each restaurant have one menu that contains zero or more dishes.
- Each dish include one or more ingredient.
- Each ingredient was mapped to a calorie table, so that total calorie of the dish can be calculated.

2.2 System Environment

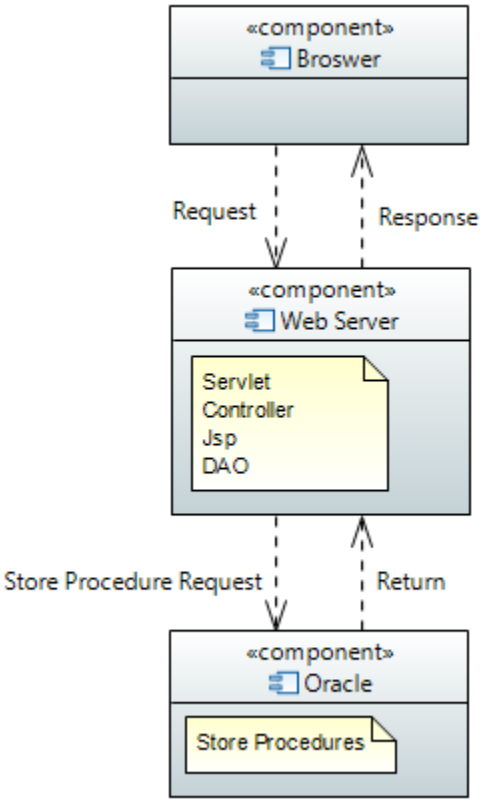
THT will run on Tomcat 8.0 server. The server will be hold by windows operating system. The development environment is spring tool suite. Oracle database is used. Hibernate will be used to communicate between server and database.

2.3 Design Methodology

The design methodology being used is the UML. Component, use case and class diagrams are attached in the following chapters.

3. Architectural Design

3.1 Component Diagram



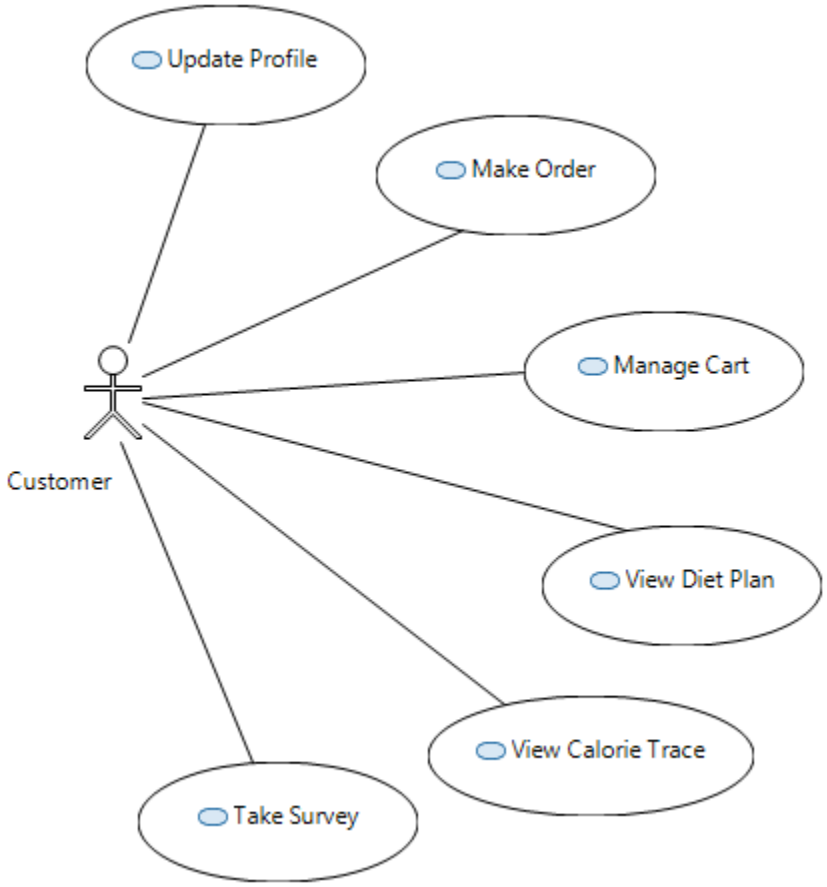
Note:

- Browser send request or get response to the web server.
- Web server process the request from browser and execute store procedures needed by using DAO.
- Oracle database holds the store procedures. Execute store procedures as needed by web server and return the data set.

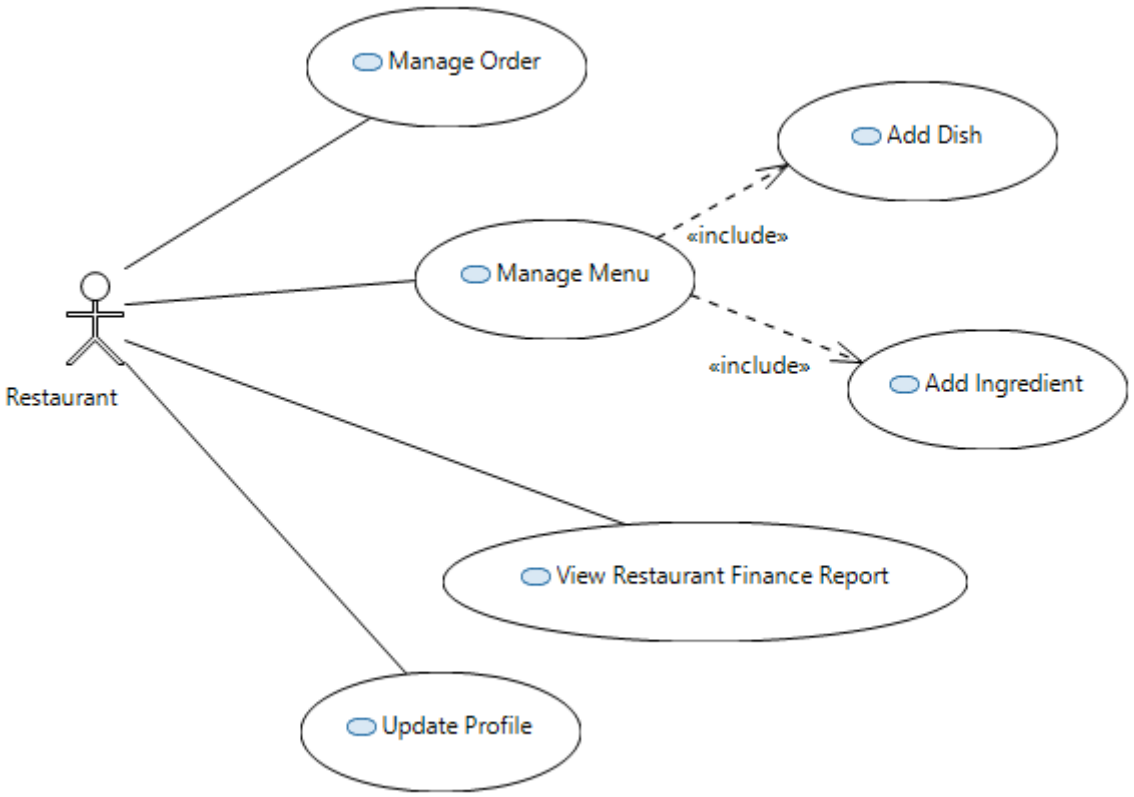
4. Low Level Design

4.1 Use Case Diagram

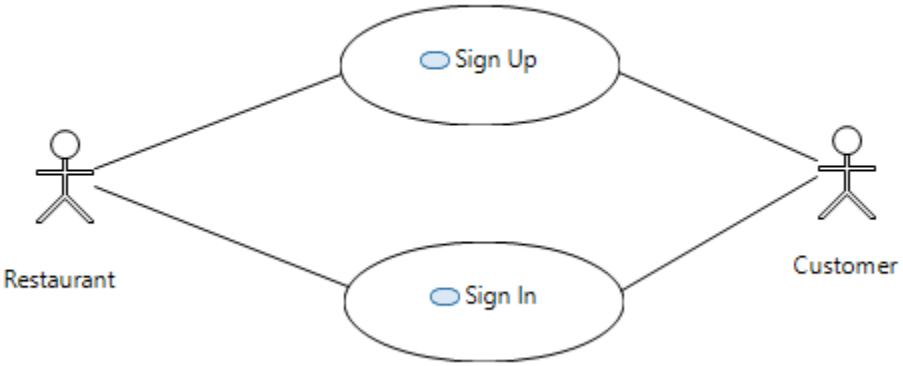
4.1.1. Customer Role



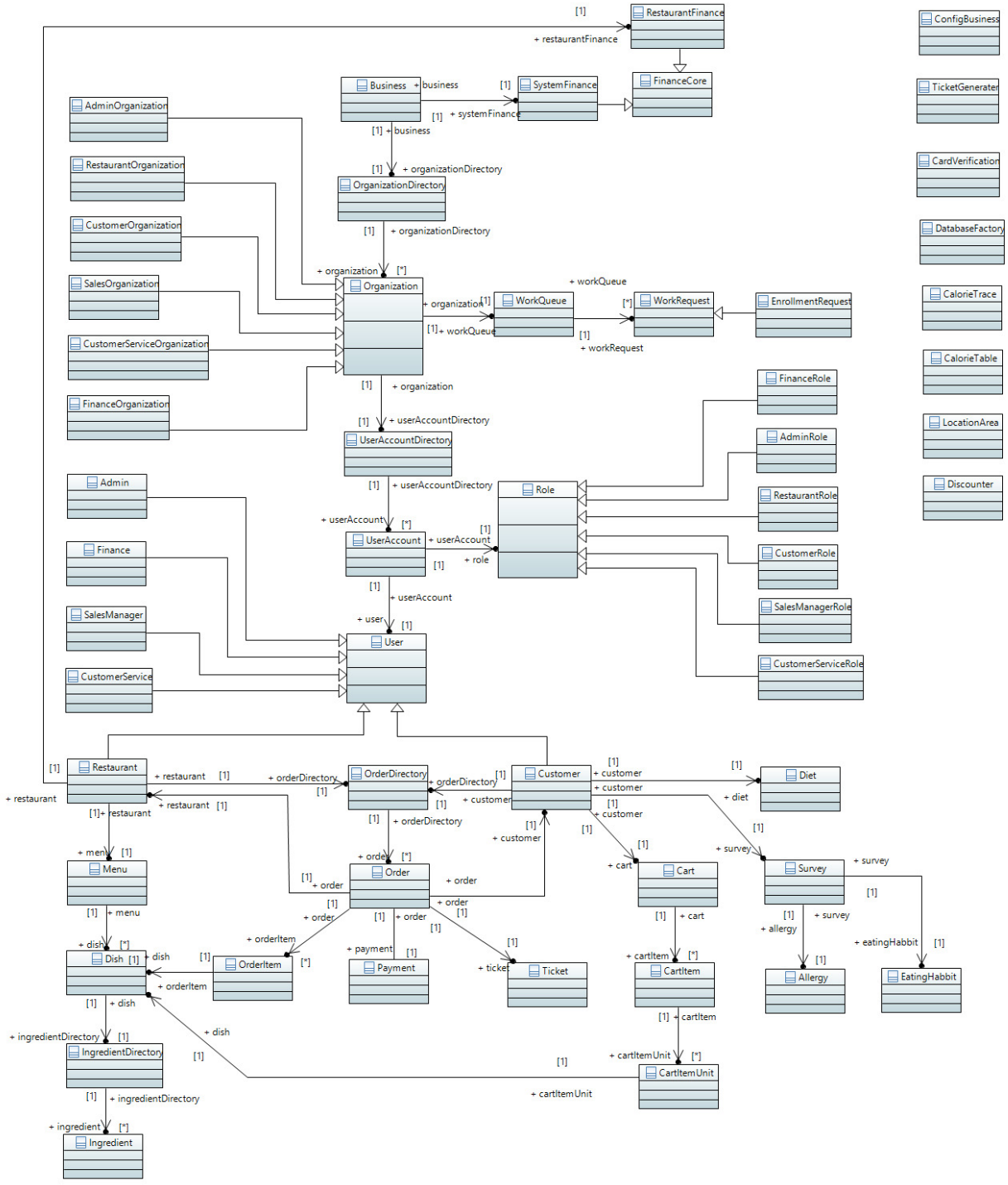
4.1.2. Restaurant Role



4.1.3. Register



4.2 Class Diagram



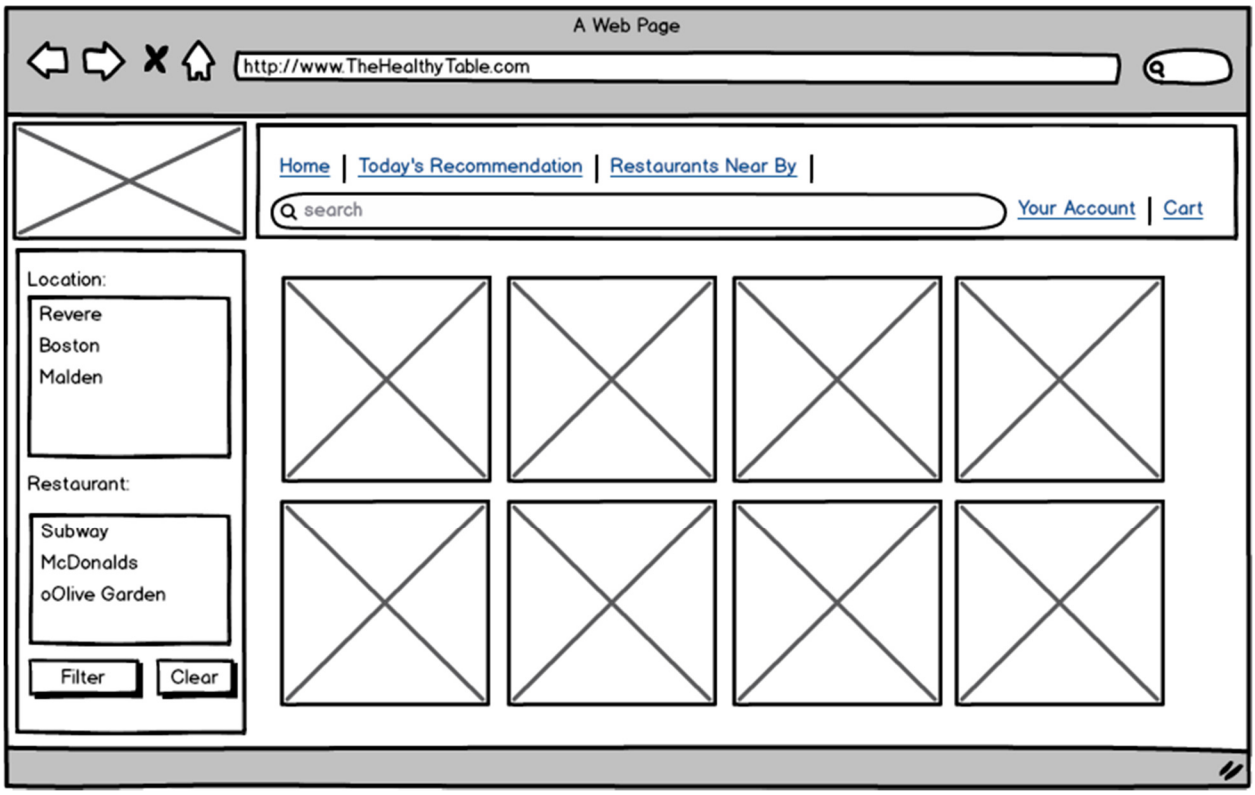
5. User Interface Design

Note: I just listed some important page here. In fact, there will be much more pages.

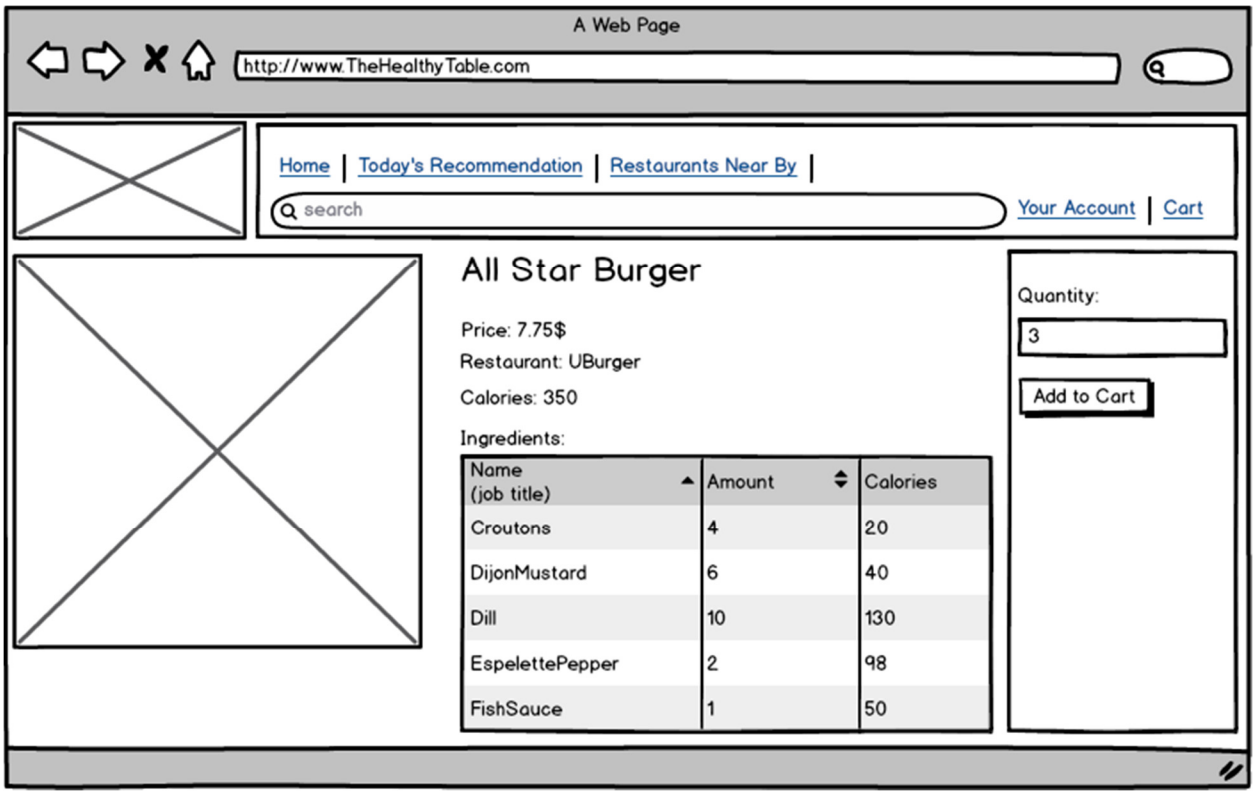
Every role have their own work space

There is also Registration page and sign in page.

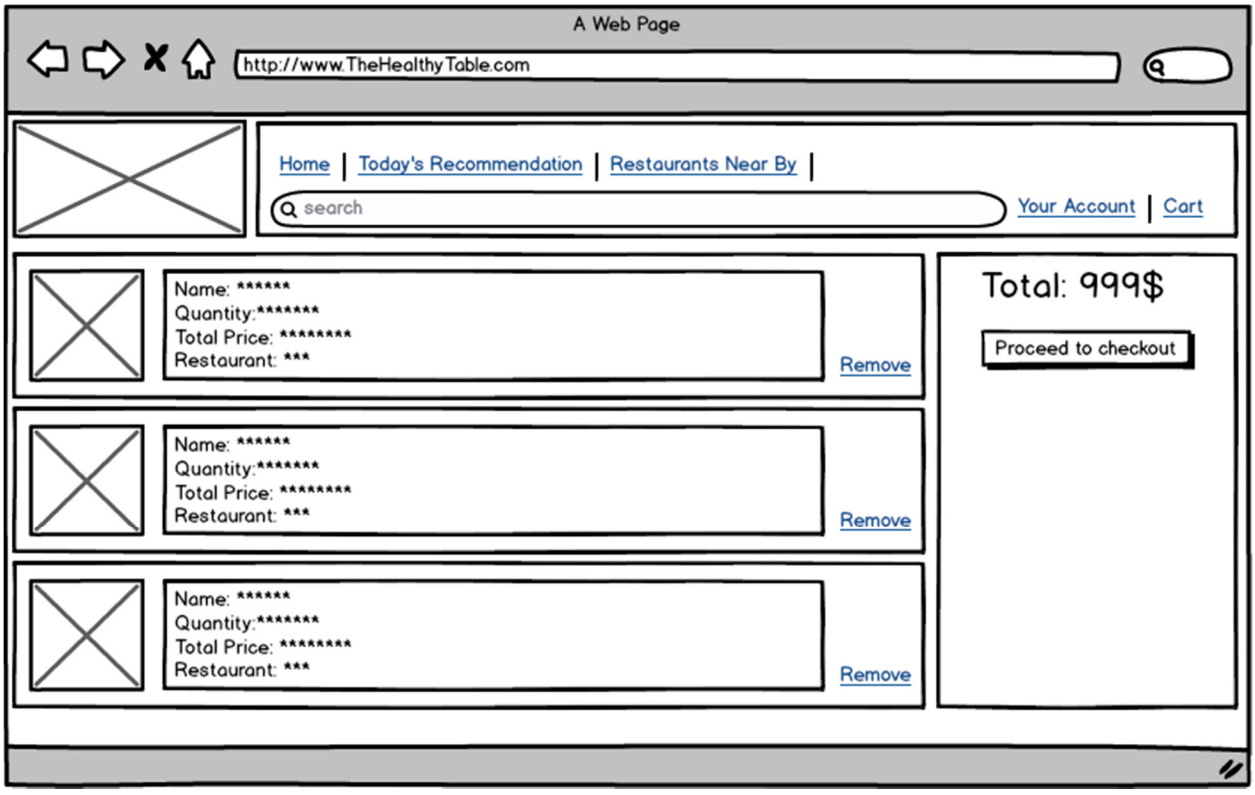
5.1 Home Page



5.2 Dish Details Page



5.3 Cart Page



5.4 Calorie Trace Page

5.5 Recommendation Page