Health Copilot

Requirements Specification

# Authors: Stephen Durham, Eric Eldridge, Justin Essian, Brian Rossman

### Requested By: Prof. Nilesh Patel CSE337 – Software Engineering Feb, 13, 2013

# Vision Statement

The Health Copilot portal is a comprehensive healthy living assistant that can make the daily life of every person easier and healthier. It is comprised of several services which combine to maximize a family’s money, time and physical health.

Buying healthy groceries on any budget is becoming harder given skyrocketing fuel and grain prices. Food mega corporations are consistently conspiring to cheapen their ingredients while maximizing their bottom line. What is needed is a way to find, price and prepare healthy alternatives to normal supermarket fare. The Meal Planner, Grocery Assistant and Pantry Assistant sub systems of the Healthy Copilot application make these time consuming and impossible chores a reality.

The Meal Planner application provides a base of recipes contributed, rated and reviewed by users around the world. Each recipe will be analyzed to provide accurate nutritional information and portioning based on the household size when scheduled for the household. The meal planner application will query users about their favorite foods when signing up for an account and make recommendations from highly rated community recipes. It will also acquire the user’s favorite recipes from non-electronic sources, websites and text based inputs. The meal planner will tentatively schedule recipes based on the user’s favorite recipes and accept alterations to the plan through an easy to use drag and drop interface.

The Pantry Assistant works with the Meal Planner application to provide the user with a real-time view of their food stuffs. This can be segmented by location, such as pantry, refrigerator and freezer. The pantry assistant will add foods via UPC, photo based recognition or user selected item and weight entry which can be augmented by a Bluetooth scale for easy data entry. Additional home automation appliances can also be integrated through the apps public API via third-party vendors.

The Grocery Assistant application can use data obtained from the meal planning application and the pantry application to generate a grocery list. The grocery list will be forwarded to the user in a variety of mediums including e-mail, SMS, the Google task app, or directly to participating grocery stores which could then choose to deliver the items. The grocery list app, will work with a user supported database of current and averaged prices to help assemble the most time and budget friendly grocery list for the user. The user can select his favorite and alternate grocery stores, as well as favorite brands.

In addition to the food aspects of this application, the Healthy Copilot will help plan a daily exercise regimen based on the planned meal intake, and additionally modified by any additional food or activity on the user’s part. This application will help target problem areas for users such as a weak back or flabby abs. The workout regimen will then be tailored to the user and help track the users progress as his/her health improves through the use of a workout/activity log. This data can be output for use by the user’s doctor or personal trainer, as an additional tool to further fine tune the user’s fitness level and overall health.

Every one of these applications is essential to every member of a household’s day to day wellbeing and in today’s fast paced society an all-encompassing health app is just what people, government agencies and health insurers need to battle the growing obesity epidemic. Time is a commodity. Eating and exercising are important overlooked aspects of confused and busy people. Automating healthy recommendations and making them budget friendly is the best way to improve a societies health.

***\*(Below here specification scoped to limited Meal Planner application only)***

# User Stories

Kara, a newly wed, is having trouble managing the meals in her household. She doesn’t ever know what to cook. She decides to browse to a website a coworker told her about. The Meal Planner application had thousands of public recipes she could browse through. As an unregistered user she was able to browse through the planned meals by month, week and day. The application advertised that it could automatically schedule meals using public and user input recipes. She decided to click on create account link. On the create account screen she enters a user name ‘badcook’ and her e-mail and password but declines to enter her name. When she clicks the submit button, she is informed the user name is already in use and a serialized variant is suggested ‘badcook2013.’ She decides to change to a less generic name ‘ysosad,’ she appreciates that she doesn’t have to reenter her password and clicks the submit button. This time the username is accepted and she is now logged in.

Bob has just finished creating an account and is currently on the home screen. He has realized he doesn’t like the password he just choose so he clicks on the account management link which shows him his account information. He enters his old password and his new password twice and clicks submit to change his password. He clicks the log off link and then attempts to log back in with the new password, it doesn’t work. He tries one more time to no avail. Instead of trying again, he clicks on the forgot password link and enters his e-mail address to get a temporary password sent to his registered e-mail. When he receives the e-mail he logs in with the temporary password. Then he continues back to account management to change to a permanent password. This time he is able to log off and log back in with no difficulty.

Sherry searched for pot pie recipes using the keywords “pot” and “pie.” The results appear in the Main View section of the web page pushing the calendar and upcoming meals view off to the right. She clicks on a recipe in the list and its description appears in the Main View area pushing the recipe list off to the right. She doesn’t like the ingredients this recipe lists, so she chooses the next recipe in the list and it appears in place of the last recipe description. She likes this recipe so she clicks on the check box ‘favorite recipe.’ The recipe immediately appears in her favorites recipe list. She also decides to add the recipe to her next unplanned dinner.

Ingrid has been using the app for a while and has finally decided to add her own recipes to the application. She clicks the ‘create’ button in the recipe toolbox which opens a create recipe dialog in the Main View portion of the application pushing the calendar and upcoming meals off to the right. She copies her text from a file into the recipe description text box and submits the recipe. The new recipe appears in the recently added recipe list. The application also opens an empty create recipe dialog in the Main View in place of the submitted recipe box. This saves Ingrid time and she starts to enter the next recipe by hand using her paper cookbook as a reference. She clicks ‘create’, and again the recipe appears at the top of the recently added recipe list. She clicks on the recipe in the recently added recipe list and it replaces the empty create recipe with the recipe she just entered. She makes a change to the description and clicks ‘edit.’ She then clicks on the area of the calendar, and a dialog appears to confirm leaving the recipe edit screen. She clicks ok and the recipe edit screen disappears leaving only the calendar and upcoming meals.

Arnold is looking for a new recipe to add to his personal cookbook so he chooses the browse all recipes link and sees all the public recipes listed starting with the highest rated. He decides to change the view to show newest recipes instead because he had previously looked at the highest rated recipes. He chooses the first recipe on the list. The recipe appears in the Main View and shifts the list off to the right. He clicks on the ‘add to recipes’ link and adds it to his personal cookbook. He then changes his mind and clicks the ‘delete from recipes’ link which removes it from his personal cookbook only.

Barbara wants to see what meals she has planned. She clicks the expand icon in the lower left of the ‘upcoming meals’ view and it expands the view into the Main View area pushing the calendar to the right. Every meal that is planned is listed with the soonest at the top of the list.

# Glossary of terms

**Calendar** - A view of the account that displays a calendar of days and has the ability to have 0 to N recipes attached to a specified meal on a specified day as a means to plan meals in advance.

**Favorite Recipe List** - An auto-populated list comprised of the recipes most frequently added to the calendar.  It is automatically limited based on the screen space available.

**Potential User (visitor) -** A person who is evaluating the service, and has the ability to create an account and become a Meal Planner.

**Meal Planner (customer)** - The term to describe the individual or group of individuals using the application for the purpose of planning meals based on a collection of recipes.

**Planned Meal** - A collection of recipes that are on the calendar for the same day and meal.

**Recently Added List** - An auto-populated list comprised of the recipes most recently added to Meal Planner’s account.  It is automatically limited based on the screen space available.

**Recipe** - A single item comprised of a list of ingredients and steps to combine them into a item that is part of a meal.

**Search Filter** - A text input that dynamically adjusts what is displayed in a list by restricting it to recipes that contain the matching text in the name.

**Personal Cookbook** - A users view of selected recipes to be used in planning meals.

# Functional and Non-functional Requirements

1. **User Account Module**Functional:
   1. Users will be required to login to an existing account, or create a new account to view and update his/her data.
   2. Once logged in, users must be able to update their account information via account management.
   3. Once logged in, users must be able to log out of their account.

Non-functional:

* 1. The customer must be able to access the account screen 24 hours a day 7 days a week.
  2. Users must fill in all required fields to log in.
  3. System response time must be efficient to prevent unnecessary delays.
  4. System must verify login information and detect a failed login attempt.
     1. If users are not logged in, or a failed login attempt has been detected the application must redirect users to the login/create account page.
  5. System must maintain an account timeout to securely log out an inactive user.

1. **Account Management Module**

Functional:

* 1. Users must be able to create an account, update, or delete an existing account.

Non-functional:

* 1. Users must be logged in to access account management.
  2. Users must fill in all required fields to create an account, reset password, or change their username.
  3. All user changes must be confirmed by the user.

1. **Recipe Toolbox Module**

Functional:

* 1. Users must be able to add a new recipe to their account.
  2. Users must be able to update or delete their store recipes.
  3. Users must be able open/display all recipes stored on their account.
  4. Users must be able to “search” for desired receipts using a search feature based on user input.

Non-functional:

* 1. User must be logged in to search, view, or make any changes to their recipes.
  2. The database must be updated in real time, and maintain connection/accessibility.
  3. All required fields must be filled in, and the search field cannot be empty.
  4. The recipe toolbox must be able to be closed or moved based on the users demand.

1. **Calendar Module**

Functional:

* 1. User must be able to access and view the calendar, including scheduled events.
  2. User must be able to update, schedule, and delete events on their calendar.

Non-functional:

* 1. User must be logged in to make changes or view the calendar.
  2. The calendar must update systematically to maintain current day, month, and year.
  3. The calendar must be able to be closed or moved based on the users demand.

1. **Upcoming Meals Module (Extended by Recipe List Function)**

Functional:

* 1. User must be able to view current planned meals for that given day.
  2. Users must be able to view planned meals for future days.

Non-functional:

* 1. User must be logged in to view meals.
  2. There must be a meal plan for that given day.
  3. The current day must be maintained by the system.
  4. All meals must be displayed in order of occurrence i.e. (breakfast, lunch, dinner).

1. **Favorites and Most Used Recipes Module (Extended by Recipe List Function)**

Functional:

* 1. User must be able to view most eaten meals and favorite meals.
  2. Users must be able to update their favorite meal list.

Non-functional:

* 1. User must be logged in to view and update favorite and most eaten meals list.
  2. Favorite meals must be located at the top of the list.
  3. System must keep track of how many times each meal is eaten by counting each every past entry in their meal plan.

1. **Recently added Module (Extended by Recipe List Function)**

Functional:

* 1. Users must be able to view all recently added meal entries.

Non-functional:

* 1. Users must be logged in to add recipes.
  2. There must exist at least one recipe in the recipes list.
  3. System must only display a predefined number of new entries, where the oldest entry drops off from the list.
  4. Users must have the option to close or minimize list.

1. **Add Recipe Module (Always displayed in Main View Module)**

Functional:

* 1. Users may add recipes by text box inputs.
  2. Users must be able to confirm unrecognized words.

Non-functional:

* 1. Users must be logged in to add recipes.
  2. The text box must not be empty.
  3. The system must prompt for user input for unrecognized words.
  4. Once a word is confirmed, the system must add it to a predefined list/dictionary.

1. **Main View Module**

Functional:

* 1. Based on user input, each desired module must expand into a main view.
  2. Users may expand or minimize different modules.

Non-functional:

* 1. User must be logged in to view the main page.
  2. Only a predefined number of modules may be expanded at any given time.
  3. If a module is expanded, any previous expanded modules must minimize.
  4. View must be user friendly, and easily accessible.

1. **Recipe List Functionality (Module Extension)**

Functional:

* 1. User must be able to add a recipe to favorites.
  2. User must be able to filter recent or popular recipes.
  3. User must be able to search for specific recipes.
  4. User must be able to add recipes to planned meals.

Non-functional:

* 1. User must be logged in to access the recipe list.
  2. A recipe must exist or be added before any other functionality takes place.

# Use Case Model



# Use Case 1

Name: Add recipe from text input  
Actors: Meal Planner  
Description: Meal Planner creates a new recipe by entering the data in a text field.  
Precondition: Meal Planner is authenticated and at the main UI page.  
Entry Action:

* Meal Planner clicks on “New Recipe”

Steps:

* If the currently displayed has unsaved edits the Meal Planner is prompted to save or discard changes.
* The recipe fields are brought into focus and the calendar is deactivated and moved off to the right side of the user interface.
* The Meal Planner fills in the Recipe Title and Recipe Information
* The Meal Planner either checks or ignores the checkbox to share the recipe.
* When the data is input and valid the Meal Planner clicks on Save
* The information must meet the following validation rules
  + The title must be unique from any other use entered by that use
  + The title may contain any alphanumeric and symbol characters with the   
    exception of “ [ ]
  + The ingredients must be parsed to be non-ambiguous
    - ambiguous examples are a quantity without a recognized unit or a step that starts in a manner that may be confused with an ingredient.

For example “1 chopped onion” could be “1 cup of chopped onions” or “1 each onion chopped” The tooltip would reflect that a unit or to include ea. or each should be added.

Exceptions:

* The title is in use
  + This will highlight the title field in red with a tooltip explaining the error.
* The title includes invalid characters
  + This will highlight the title field  in red with a tooltip explaining the error.
* Some of the text of Recipe information is not properly identified by the parsing engine.
  + This will highlight the text in question and provide tooltips.

Post-condition:

* The recipe is stored as a record in the Recipe DB with a reference to the Meal Planner’s account
* The recipe name is displayed at the top of the ‘Recently Added’ list
* If the option to share the recipe was selected the recipe is listed publicly for any other Meal Planner accounts to view and add to their recipe book.

# Use Case 2

Name: Create Account  
Actors: Potential User  
Description: Potential User creates a new account using sign up functionality  
Precondition: No existing Meal Planner is authenticated  
Entry Action: Potential User clicks on “Create Account”  
Steps:

* Potential User fills in required account creation information in the displayed form
* The following information is required
  + email
  + email confirm email
  + password
  + confirm password
* The following information is optional
  + first name
  + last name
* The information must meet these validation rules
  + The email and confirm email must match
  + The email must not already be in the database
  + The password and confirm password must match.
  + The password must be at least 8 characters

Exceptions:

* Potential User enters incorrect information.
  + Password is too short
  + Password and confirm password do not match
  + email is not a properly formatted email address
  + email and confirm email do not match
* Email already in use
* A message indicating the nature of the exception will be displayed and a red asterisk will be placed next to the field that needs to be altered.

Post-condition:

* A message of success is displayed.
* A message of success is sent to the registered email.
* Potential User is now logged in (authenticated) as a registered Meal Planner
* Redirect to the main application user interface.

# Use Case 3

Name: Add meal from public recipe database  
Actors: Meal Planner  
Description: Browse or search the publicly shared meals and add one to the Meal Planner Account.  
Precondition: Meal Planner is authenticated and at the main UI page.  
Entry Action:

* Meal Planner Clicks on Browse All Recipes

Steps:

* Meal planner is shown the Recipe Browser Interface.
* The Option for My Recipes is changed to Public Recipes
* Search and Filter are used to find a desired recipe.
* When each recipe is selected the information is displayed.
* When the use finds a recipe that would like to add they click on “Add to Recipe Book”

Exceptions:

* Recipe is already in the Meal Planner’s account Recipe Book
  + Returns an error message and remains on the same screen

Post-condition:

* A reference to the public recipe is added the Meal Planner’s account
* The recipe name is displayed at the top of the ‘Recently Added’ list
* The Meal Planner changes back to the main UI

# User Screens





