**Week 6 Status Report**

Awesome Inc.

Work started on designing the database structure and UI functionality this week. The database has been normalized, and consists of 12 tables. It currently consists of 10 tables that work to develop a Recipe object, for use in the project.

The Recipe object itself consists of many individual pieces of information:

* Name
* Picture
* Prep time
* Total time
* Servings
* Summary of recipe
* Author / credits
* Prep Style
* All available nutritional information (serving size, calories, etc.)
* Multiple steps of the directions
* Multiple ingredients and their amounts
* Multiple tips (substitutions, etc.)

The Database also contains a table for user information, and a table for keeping track of users’ ratings of individual recipes.

The first complete draft of the Database UML has been attached as a pdf.

The UI functionality has finished the planning phase, and will be reiterated upon during development. It is expected that the user will be able to perform several functions while using the RecipEasy application (this information is subject to change):

**Use Case Name**: Case01

**Alias:** Login Selection

**Description**: User will be prompted with entrance screen, requesting whether the user is an existing user, and would like to login, or whether user is new, and needs to register an account. Buttons will be provided for each condition.

**Successful Completion**: User will be given access to whichever interface they have selected.

**Alternative**: Error message detailing the problem, and closure of application.

**Precondition**: Application start.

**Post condition**: The user will be directed to the appropriate interface.

**Use Case Name**: Case02

**Alias:** User Login

**Description**: User will be prompted with login screen, requesting username and password. Submission button will be provided to submit this information

**Successful Completion**: User will be given access to the main application interface. The application will be able to access and edit user specific information, such as ratings.

**Alternative**: If the information submitted does not match the information on record, the user will be asked to re-submit the information, and will not be allowed to continue. A link to register new information will also be provided.

**Precondition**: The user must have chosen to login as an existing user.

**Post condition**: The user will be directed to the main application interface.

**Use Case Name**: Case03

**Alias:** User Registration

**Description**: User will be prompted with registration screen, requesting username and password. There will be a second password entry box to verify the password matches, along with a submission button.

**Successful Completion**: User will be notified of successful account creation, and given access to the main application interface. The application will create the appropriate database records, and be able to access and edit user specific information, such as ratings.

**Alternative**: If the passwords do not match, or the username already exists, a message will appear informing the user of these issues. If the application encounters another error, the detailed information will be given to the user, followed by application closure.

**Precondition**: The user must have chosen to register as a new user.

**Post condition**: The user will be directed to the main application interface.

**Use Case Name**: Case04

**Alias:** Recommendations

**Description**: User will be given the option to search for recommended recipes, with specific categories to narrow down the selection of recipes (cooking style, total time, etc.), along with a submission button.

**Successful Completion**: User will choose categories for narrowing, and an algorithm will search the database using the narrowing categories, as well as taking into account user’s previous ratings. A selection of recipes will populate the window, allowing the user to select an individual one.

**Alternative**: If the search of the database returns no recipes, inform the user with a message.

**Precondition**: The user must have access to the main interface

**Post condition**: The user will be directed to the recipe selection interface.

**Use Case Name**: Case06

**Alias:** Favorites

**Description**: User will be shown a list of all of their rated recipes, in order of their highest and most recently rated.

**Successful Completion**: User will be able to select a previously rated recipe.

**Alternative**: If the user has not rated any recipes, display a message asking them to.

**Precondition**: The user must have access to the main interface

**Post condition**: The user will be directed to the recipe selection interface.

**Use Case Name**: Case07

**Alias:** Search

**Description**: User will enter in information to search box, and be provided a button for submission.

**Successful Completion**: The application will run the search term through an algorithm that queries the database for the term, and displays a list of recipes that most closely resemble the search term. The user will then be able to select an individual recipe.

**Alternative**: If the search returns no results, inform the user with a message.

**Precondition**: The user must have access to the main interface.

**Post condition**: The user will be directed to the recipe selection interface.

**Use Case Name**: Case08

**Alias:** Rate Recipe

**Description**: User will be able to rate the selected recipe based on cost, ease of production, and whether it was liked.

**Successful Completion**: The application will determine which information was rated, and enter that information into the database, including the user identification information. A message will be provided that reiterates the user’s recipe selection and rating(s).

**Alternative**: If an error occurs, inform the user.

**Precondition**: The user must have access to the recipe interface.

**Post condition**: The user will stay on the recipe interface.

The website information scraping tool has also begun development this week. The scraping application will run standalone to the RecipEasy application. It will receive the web address(es) of the recipe(s) as command line arguments, and use the web address to get the code of the address and implement it as a StringBuilder object. The StringBuilder object will be broken into sections using keyword searches that determine the type of information needed to create a Recipe entry in the database. These sections will be further broken down to extract just the information needed, in the necessary formats. That information will then be displayed to the scraper application user, who will verify it and determine whether to commit it to the database using another method of the application.