System Design Document

RecipEasy

By: David Palmieri, Brandon Adams, Yunkai Deng, Chris Costa and Nathaniel Poli

3/29/2016

Table of Contents

[1 INTRODUCTION 3](#_Toc447057622)

[1.1 Purpose and Scope 3](#_Toc447057623)

[1.2 Project Executive Summary 3](#_Toc447057624)

[1.2.1 System Overview 3](#_Toc447057625)

[1.2.2 Design Constraints 4](#_Toc447057626)

[2 SYSTEM ARCHITECTURE 5](#_Toc447057627)

[2.1 System Software Architecture 5](#_Toc447057628)

[3 FILE AND DATABASE DESIGN 5](#_Toc447057629)

[3.1 Database Management System Files 5](#_Toc447057630)

[4 HUMAN-MACHINE INTERFACE 8](#_Toc447057631)

[4.1 Inputs 8](#_Toc447057632)

[4.1.1 Login Interface 8](#_Toc447057633)

[4.1.2 Search Box 9](#_Toc447057634)

[4.1.3 Recipe Display 10](#_Toc447057635)

[4.2 Outputs 11](#_Toc447057636)

[4.2.1 Login Interface 11](#_Toc447057637)

[4.2.2 Search Box 12](#_Toc447057638)

[4.2.3 Recipe Display 13](#_Toc447057639)

[5 DETAILED DESIGN 15](#_Toc447057640)

[5.1 Hardware Detailed Design 15](#_Toc447057641)

[5.2 Software Detailed Design 15](#_Toc447057642)

[5.2.1 Data Package 15](#_Toc447057643)

[5.2.2 ModifyDB Package 15](#_Toc447057644)

[5.2.3 UserInterface Package 15](#_Toc447057645)

[5.2.4 RecipeDB Package 16](#_Toc447057646)

# INTRODUCTION

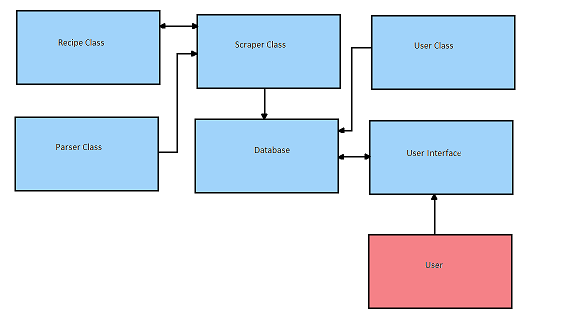
## Purpose and Scope

The Systems Design Document (SDD) keeps track of the various subsystems within the system and how they fit together. It is meant to be useful to people such as the project manager, project team, and development team. Some parts such as the user interface (UI) will be shared with the user. The reader of this document should be able to understand the various architectures of the system (RecipEasy).

## Project Executive Summary

The project management is being handled by two teams. The UI team is responsible for creating and maintaining the user interface, and the backend team is responsible for the database, the various classes, and the logic to make RecipEasy function. All testing is being handled internally.

### **System Overview**

At the center of RecipEasy is a JDBC derby database. This will be queried by the user interface to return data to the user. It will also contain user information and passwords, as well as user favorites and preferences. The scraper class will only be used once, to build the initial recipe database. The scraper class gathers data via HTML forms from bettycrocker.com and the parser class gathers all the pertinent information from those forms. Then that information is passed to the recipe class to build recipe objects. These recipe objects are then put into the database. 

### **Design Constraints**

Project scope is limited by our lack of knowledge creating anything outside of java console applications, so the application will only be able to be run on a PC. Android/IPhone development would be good but we are limited in our knowledge of developing for other operating systems. This also will be a standalone application, as we don't have the knowledge to create a web application.

# SYSTEM ARCHITECTURE

## System Software Architecture

# FILE AND DATABASE DESIGN

## Database Management System Files

The database consists of 12 tables that help to define users, recipes, and the details of each recipe, including user created ratings. The database file size is around 75 megabytes, with another 750 megabytes of space needed for picture storage. It is an offline in-application database that can be updated when required by an administrative member.

User: contains login information for each user

* UserID: the primary key used to relate users to specific recipe ratings
* Username: the login name for the specified user
* Password: the login password for the specified user

Ratings: contains a list of each individual rating made by each user

* RatingID: the primary key used to denote specific ratings
* UserID: foreign key used to relate a user to the rating
* RecipeID: foreign key used to relate a recipe to the rating
* Cost: numerical rating of how expensive the user found making the recipe to be
* Ease: numerical rating of how easy the user found making the recipe to be
* Liked: numerical rating of how well the user appreciated the taste of the recipe

The recipe data object used in the application is built using the following 10 tables:

Recipe: contains all single value details of the recipe.

* RecipeID: the primary key used to relate all multiple value details from other tables
* Title: The title of the recipe
* URL: the url of the recipe where the information was drawn from
* PrepTime: the preparation time of the recipe
* TotalTime: the full cooking time for the recipe
* Servings: the total amount of servings that the recipe provides
* Summary: an explanation of the recipe
* ServingSize: nutritional value- serving size
* Calories: nutritional value- calories
* CalFat: nutritional value- calories from fat
* TotFat: nutritional value- total fat
* SatFat: nutritional value- saturated fat
* TransFat: nutritional value- trans fat
* Cholesterol: nutritional value- cholesterol
* Sodium: nutritional value- sodium
* Carbs: nutritional value- carbohydrates
* Fiber: nutritional value- fiber
* Sugar: nutritional value- sugars
* Protein: nutritional value- protein
* VitA: nutritional value- vitamin A
* VitC: nutritional value- vitamin C
* Calcium: nutritional value- calcium
* Iron: nutritional value- iron

Category: contains all of the possible categories that each recipe can be a part of

* CategoryID: the primary key used to relate categories to recipes
* Category: the actual category name

RecipeCategory: Linking table between Recipe table and Category table

* RecCatID: the primary key used to denote specific relationships
* RecipeID: foreign key used to relate recipes
* CategoryID: foreign key used to relate categories

Tip: contains special tips written for certain recipes, not always present

* TipID: the primary key used to relate tips to recipes
* Tip: the actual tip

RecipeTip: Linking table between Recipe table and Tip table

* RecTipID: the primary key used to denote specific relationships
* RecipeID: foreign key used to relate recipes
* TipID: foreign key used to relate tips

Ingredient: contains all ingredients used in all of the recipes

* IngredientID: the primary key used to relate ingredients to recipes
* Ingredient: the actual ingredient

RecipeIngredient: Linking table between Recipe table and Ingredient table

* RecIngID: the primary key used to denote specific relationships
* RecipeID: foreign key used to relate recipes
* IngredientID: foreign key used to relate ingredients

Instruction: contains all instructions used in all of the recipes

* InstructionID: the primary key used to relate instructions to instruction steps
* Instruction: the actual instruction

RecipeInstructionStep: contains the instruction and its step number in a recipe

* RecInstStepID: the primary key used to relate instruction steps to recipes
* StepID: the numbered step that this instruction is for a given recipe (instructions must be in order for each recipe)
* InstructionID: foreign key used to relate instructions

RecipeInstruction: Linking table between Recipe table and RecipeInstructionStep table

* RecInstID: the primary key used to denote specific relationships
* RecipeID: foreign key used to relate recipes
* RecInstStepID: foreign key used to relate instruction steps



# HUMAN-MACHINE INTERFACE

## Inputs

The users interact with a system through the interfaces. To increase the satisfaction of the users, a set of user-friendly GUI was designed by our team. It helps users perform their tasks easily.

### **Login Interface**

User name text field: input username

Password field: input password associated with the username

Sign In button: click to launch main page

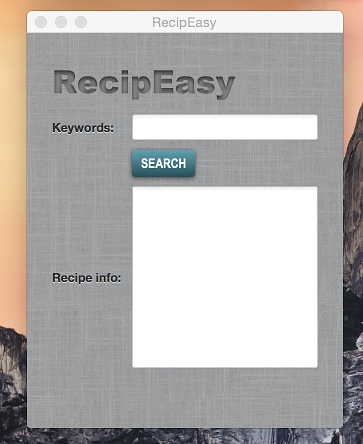


### **Search Box**

Keywords text field: input keywords related to interested recipes

Recipe info field: return a list of search results

Search button: click to perform search recipes action



### **Recipe Display**

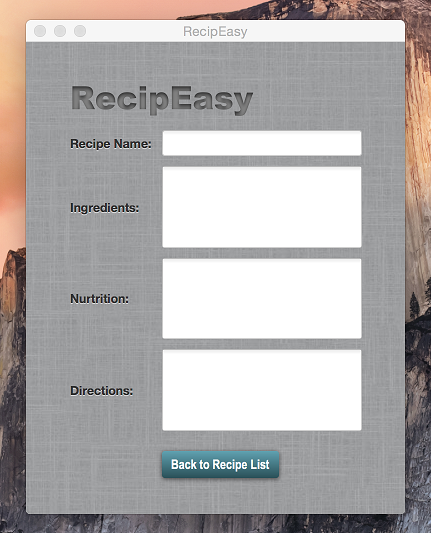
Recipe name text field: is used to display recipe name

Ingredients field: is used to display detailed ingredients

Nutrition field: is used to display nutrition information

Direction field: is used to display steps on how to cook

Back button: click to perform go back to recipe list action



## Outputs

### **Login Interface**

User name text field: is expected to show username

Password field: is expected to hide password

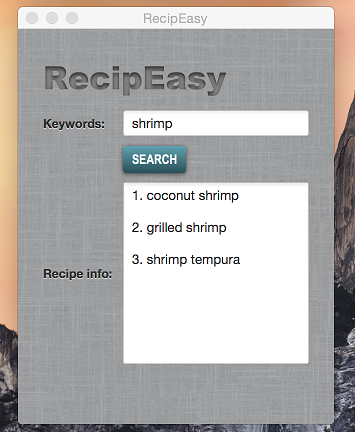
Sign In button: click to launch main page



### **Search Box**

Recipe info field: is expected to return a list of search results

Search button: click to perform search recipes action



### **Recipe Display**

Recipe name text field: is expected to display recipe name

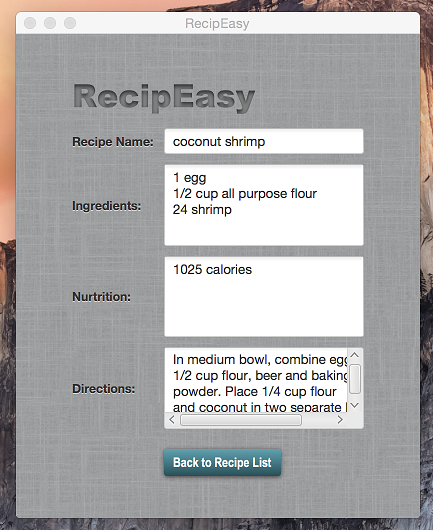
Ingredients field: is expected to display detailed ingredients

Nutrition field: is expected to display nutrition information

Direction field: is expected to display steps on how to cook

Back button: click to perform go back to recipe list action

Image filed: a field will be added further to display images



# DETAILED DESIGN

## Hardware Detailed Design

The project is designed to work on a PC with the following specifications:

* + Operating System: Windows 7 / 8.1 / 10
  + Memory: 2 GB
  + Storage: 1.5 GB
  + Processor: X86, X64
  + Monitor: 800 x 600
  + Network: Optional
  + Software: JRE 1.8

## Software Detailed Design

### **Data Package**

-Recipe Class: Contains a summary of the recipe, number of servings,

 nutrition, etc.

-RecipeList: Maintains a list of recipes

-User Class: user object

### **ModifyDB Package**

-CreateDB: Create Database

-DBInsertCategory: Insert a category

-DBInsertRecipe: Insert a recipe

-HTMLParser: Reads html and splits by tags

-QueryDB: query database

-Scraper: builds the recipe objects from the html

-URLGrabber: grabs the URL of the html

-CreateDB: Create Database

-DBInsertCategory: Insert a category

-DBInsertRecipe: Insert a recipe

-HTMLParser: Reads html and splits by tags

-QueryDB: query database

-Scraper: builds the recipe objects from the html

-URLGrabber: grabs the URL of the html

### **UserInterface Package**

UserInterface Class

+main

FrontPage

-fxml

-css

### RecipeDB Package

+Derby database of recipes