**Software Requirements Specification**

**Automated Electronic Cookbook**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pete Koehn**

**Theodore Lindsey**

**Brian Kennedy**

**EECS 448**

**1. Introduction**

1.1 Purpose

1.2 Intended Audience

1.3 Contact Information / SRS Team Members

1.4 References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 User Environment

2.6 Design / Implementation Constraints

2.7 Assumptions and Dependencies

**3. External Interface Requirements**

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

**4. System Features**

4.1 Description and Priority

4.2 Functional Requirements

**5. Other Nonfunctional Requirements**

5.1 Performance Requirements

5.2 Software Quality Attributes

5.3 Project Documentation

5.4 User Documentation

**1. Introduction**

**1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) document is to provide a thorough description of the constraints, expectations, and timeline of the project, as well as an explanation of the application’s operation and output. The deliverable in this project is an electronic cookbook application that allows a user to add, modify, and view a collection of recipes contained within the application.

**1.2 Intended Audience**

This SRS document intended for review and use by Dr. Chakrabarti of the University of Kansas.

**1.3 Contact Information / SRS Team Members**

Contributors to this project include:

Pete Koehn

E-mail: [pete.koehn@ku.edu](mailto:pete.koehn@ku.edu)

Phone: (913) 522-7021

Theodore Lindsey

E-mail: [theodore.lindsey@gmail.com](mailto:theodore.lindsey@gmail.com)

Phone: (503) 898-0184

Brian Kennedy

E-mail: b919k168@ku.edu

Phone: (XXX) XXX-XXXX

**1.4 References**

The author of this document has completed several software projects for other clients. References will be made available upon request.

**2. Overall Description**

**2.1 Product Perspective**

The product to be developed will consist of a stand-alone computer application. Once the user executes the application, they GUI will appear, and the user will be presented with several ways to interact with the program. These ways include: importing a recipe from a specifically-formatted text file, adding a recipe to the database via the GUI, modifying recipes already in the database, and generating a PDF of the contents of the database, which can be organized by various parameters (style, ethnicity, etc.).

The application will be implemented in Python.

**2.2 Product Functions**

After the desired image is selected for conversion and imported into the application, the application will perform its function without any further input from the user. All parameters for conversion will be set prior to project completion.

**2.3 User Classes and Characteristics**

The general class of user of the application will be the general public. The software will be distributed for free and is available to anyone who might be interested in using such an application. The GUI and its actions will be simple and straightforward, and will be delivered for use on Windows-based machines.

**2.4 Operating Environment**

The application will be executed on a Windows-based desktop computer. Appropriate operating systems for running the application include Windows XP, Windows 7, and Windows 8. If the target machine does not meet these operational requirements, please contact me.

**2.5 User Environment**

The application will initially be delivered to Dr. Chakrabarti, after which point it will be made available in the public domain.

**2.6 Design / Implementation Constraints**

The application is constrained by the formatting of the recipe text files it can import. All recipes imported by the application must use XML markup. Instructions and sample markup will be provided for the user to ensure their text files will import properly.

**2.7 Assumptions and Dependencies**

It will be assumed that the computer(s) chosen to run the application will meet minimum performance requirements. The application will execute within a reasonable amount of time given the host machine has proper/acceptable hardware.

**3. External Interface Requirements**

**3.1 User Interfaces**

The user interface will be simple and straightforward, and it will consist of the following parts: recipe import (via text file), recipe input (via GUI), recipe modification, and recipe book PDF generation. Each functional component will be represented in the GUI with a standard, labeled button. If further input is requested from the user, it will be requested via a window prompt.

**3.2 Hardware Interfaces**

This application does not provide any interfaces to hardware, external or otherwise.

**3.3 Software Interfaces**

This application does not require any interface with external software.

**4. System Features**

**4.1 Description and Priority**

The following is a list of each functional component of the application, and a corresponding description:

To execute the recipe import feature, the user will launch the application and select ‘Import a Recipe’. The application will then prompt the user to ensure that the text file is properly formatted and is located in the source directory of the application. The user can confirm both of these requirements have been met by selecting ‘OK’. Once the user selects ‘OK’, the application will request the filename of the recipe from the user. The user will input the correct filename, and the application will import the recipe.

To add a recipe to the database, the user will launch the application and select ‘Add a Recipe’. The application will then prompt the user to add information via a series of prompts. These prompts include the recipe’s name, serving size, ingredients, bake/cook time, instructions, etc.

To modify a recipe, the user will launch the application and select ‘Modify a Recipe’. The application will then prompt the user to enter the name of the recipe. Once the user inputs the name of a recipe within the database, the contents of that recipe will be presented to the user. The user will then be able to select a field (name, ingredients, bake/cook time, etc.) to modify. The user will be prompted to enter the updated information. Once the user inputs the new information, the application will update the recipe.

To generate a PDF of the cookbook, the user will launch the application and select ‘Generate Cookbook PDF’. The generation of the PDF is fully automated, and will not require further input from the user. The generated PDF will be saved in the application’s source directory.

**4.2 Functional Requirements**

Importing a recipe: The user must possess a text file in the specified XML format. Instructions and examples of this formatting requirement will be made available as a deliverable along with the application.

Generating a PDF: PDF generation will only be possible if there is at least one recipe in the user’s recipe database.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

The application may be used at any number of workstations concurrently, by any number of independent users.. There is no limit (beyond the limit implied by the available hard drive space on the user’s machine) to the number of recipes that may be imported or added to the application’s database. The application may be run at the same time as other applications on the workstation, as it will not place a high demand on the workstation’s resources.

**5.2 Software Quality Attributes**

The application provides an extremely simple interface for the user. All available actions (importing, adding, and modifying, and printing (to a PDF) recipes) will be plainly visible in a straightforward menu system.

**5.3 Project Documentation**

The application will be implemented in Python. Importing will be facilitated using an XML parser within in Python. Using a parser, the application will be able to populate the various fields of a recipe (name, servings, ingredients, etc.).

Adding and modifying recipes will take place within the application via a series of input prompts. Input will be taken directly from the user at this point, so no additional facilitation is needed.

PDF generation will be entirely hidden from the user. The user will select ‘Generate Cookbook PDF’, and the PDF may be generated using the Adobe Reader print option.

**5.4 User Documentation**

The user will run the application by opening the given executable file. Once the application has launched, the user has several options:

To import a recipe, the user will select ‘Import Recipe’. This action will launch a familiar file system browser window, from which the user will navigate to the recipe text file they wish to import. Once the text file is selected, the user will confirm that the recipe is in the proper format, and the application will import the recipe into the database.