RECIPE REPOSITORY PROJECT DESIGN

Version 1.0

Adam Howell

Alexander Macwilliams

Claire Breer

Eliot Pearson

Justin Helphenstine

Obinna Ojialor

09/18/2016

Table of Contents

Revision History

Date	Version	Description	Author
09/18/2016	1.0	Initial Release	A. MacWilliams

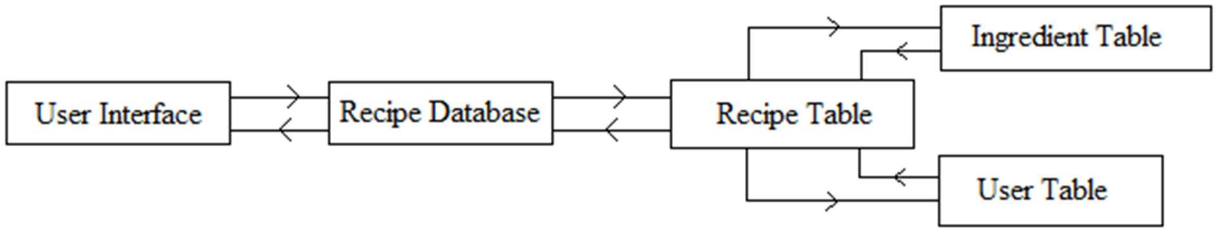
SYSTEM DESIGN DOCUMENT

1 INTRODUCTION

1.1 Purpose and Scope

The purpose of this design document is to describe the architecture requirements as well as the design architecture for the Shark Bread Recipe Repository Project. This includes the operating environment, file and database design, input and output design, and so on.

1.2 Project Executive Summary



The program that is being designed will be built out from three core components: the front-end graphical user interface (GUI), the back end database, and a middle management module that facilitates the queries from the user interface to the database. Three teams will work on each of the components.

1.2.1 System Overview

The user interface provides an environment for the user to enter, edit, and search for recipes. The recipes are stored in a database, with sub-tables that list user and ingredient information.

1.2.2 Design Constraints

The constraints of this project will be that it will be a stand-alone application on the user's computer, as opposed to a web-based application. This has the downside of making it difficult for users to access their database from a separate system.

1.2.3 Future Contingencies

Contingencies for this application come in the form of malformed or unexpected data types. Bounds-Checking and input validation are the system's guards against these contingencies.

1.3 Document Organization

The design document is organized into 7 sections:

1. Introduction
2. System Architecture
3. File and Database Design
4. Human Machine Interface
5. Detailed Design
6. External Interfaces
7. System Integrity Controls

1.4 Points of Contact

The points of contact are listed in Section 7, Staffing Management Plan, of the Project Plan listed in section 1.5.

1.5 Project Reference

Document Type	Document Name
Project Plan	Shark_Bread_CMSC495_Plan
Test Plan	Shark_Bread_CMSC495_Test_Plan
User Guide	Shark_Bread_CMSC495_User_Guide

1.6 Glossary

Graphical User Interface - GUI

Integrated Development Environment – IDE

CRUD – Create, Read, Update, and Delete

2 SYSTEM ARCHITECTURE

This system will use a database-centric architecture style, that will consist of a relational database storing the recipe by field types, and a front end user interface that will create queries to this database.

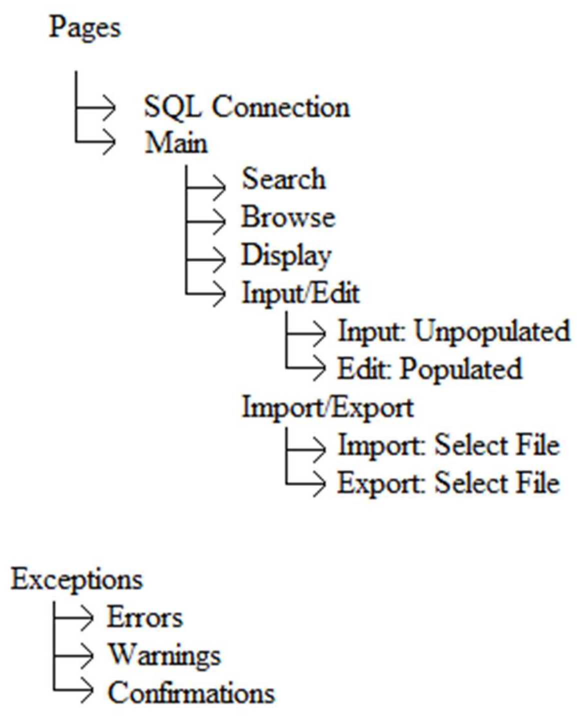
2.1 System Hardware Architecture

N/A

2.2 System Software Architecture

The software architecture will consist of two languages and integrated development environments: Java, through the IDE Netbeans, for the user interface, and SQL, through the IDE sqlite, for the database.

The user interface will consist of a display box supported by select, search, enter, and delete classes. A module will be included in order to access the database and return queries.



2.3 Internal Communications Architecture

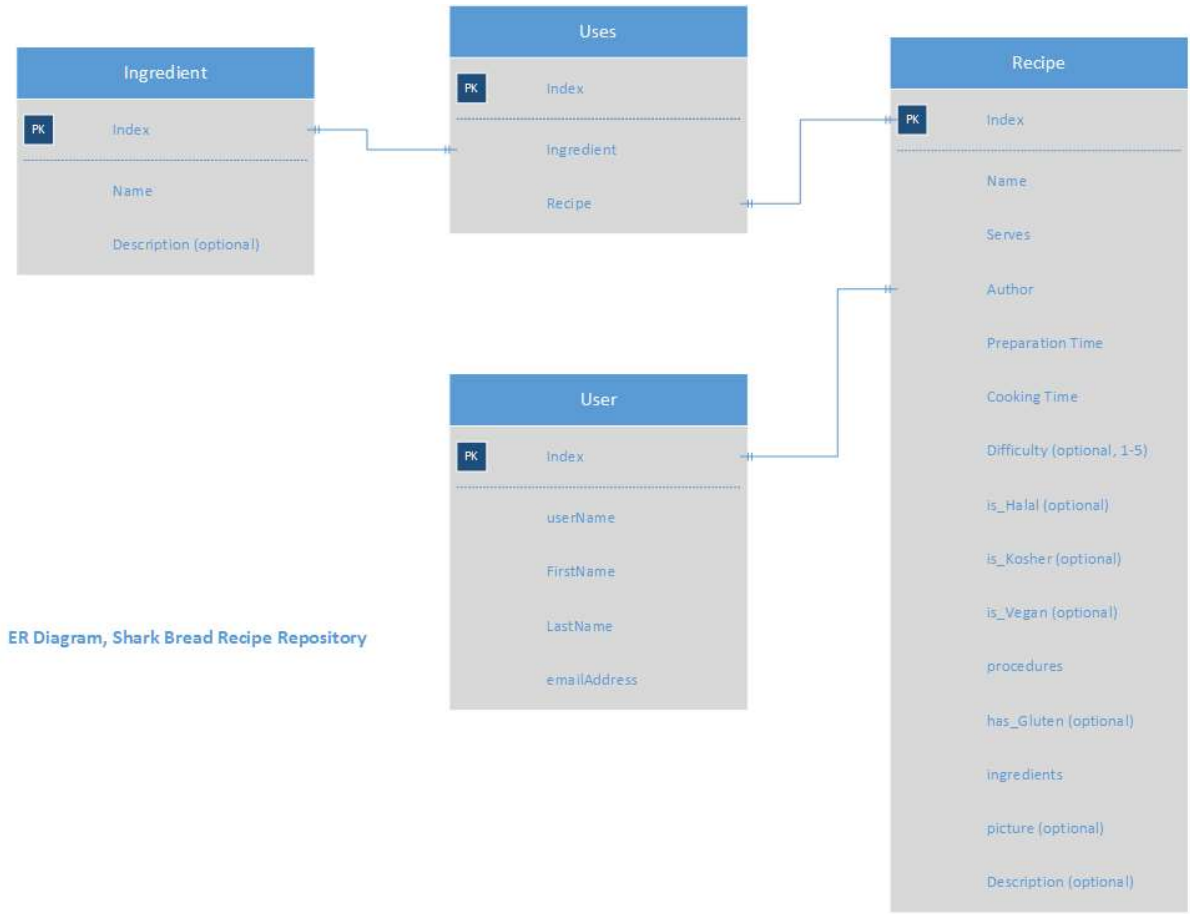
N/A

3 FILE AND DATABASE DESIGN

ID	Field Name	Description	Maintenance	Output	Validation Rules	Data Type	Length
DD001	recipeName	Name of the recipe.	CRUD	Recipe Name: [recipeName]	N/A	String	50
DD002	serveSize	How many people the recipe serves.	CRUD	Serves: [serveSize]	Is entry a number?	Integer	2
DD003	Author	Name of the author.	CRUD	Author: [author]	N/A	String	50
DD004	prepTime	Time it takes to prepare the recipe.	CRUD	Preparation Time: [prepTime]	Is entry a number.	Time	20
DD005	cookTime	Time it takes to cook the recipe.	CRUD	Cooking Time: [cookTime]	Is entry a number.	Time	20
DD006	Difficulty	Perceived difficulty of the recipe.	CRUD	Difficulty: [difficulty]	Is entry a number.	Integer	1
DD007	Procedures	Instructions for preparing the recipe.	CRUD	Procedures: [procedures]	N/A	String	Str.Max_Length
DD008	Description	Description of the recipe.	CRUD	Description: [description]	N/A	String	Str.Max_Length
DD009	Ingredients	List of ingredients.	CRUD	Ingredients: [ingredients]	N/A	String	Stri.Max_Length

3.1 Database Management System Files

Logical Model:



Physical Description: Three tables will exist: one for the main fields, with the data organized by the recipe name, and two other sub-tables will exist for storing data based on author name, and ingredient list.

Access Methods: The data will be stored via an indexed key, by recipe name.

DBMS File Size Estimate: 10mb

Update Frequency: N/A

3.2 Non-Database Management System Files

No non-database management system files will be used other than the program executable. Data files will exist to import and export recipes via text files. Format will be:

Recipe Name::

<NAME>

Recipe Ingredients::

<Ingredient>::Quantity

.....

<Ingredient>::Quantity

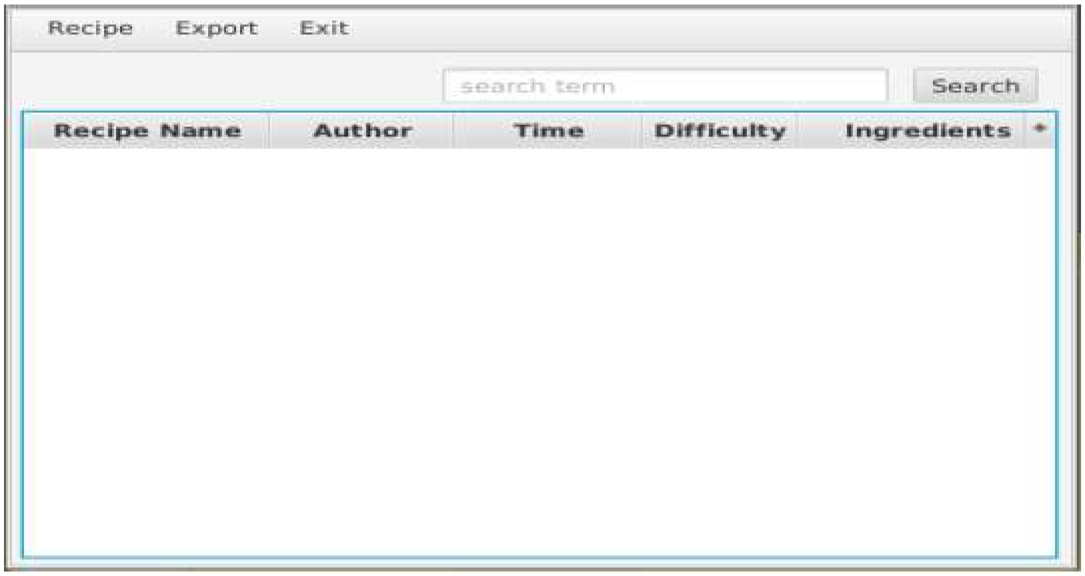
Recipe Steps::

<Steps>

(Note: Data format is a prototype and may be subject to change in final version)

4 HUMAN-MACHINE INTERFACE

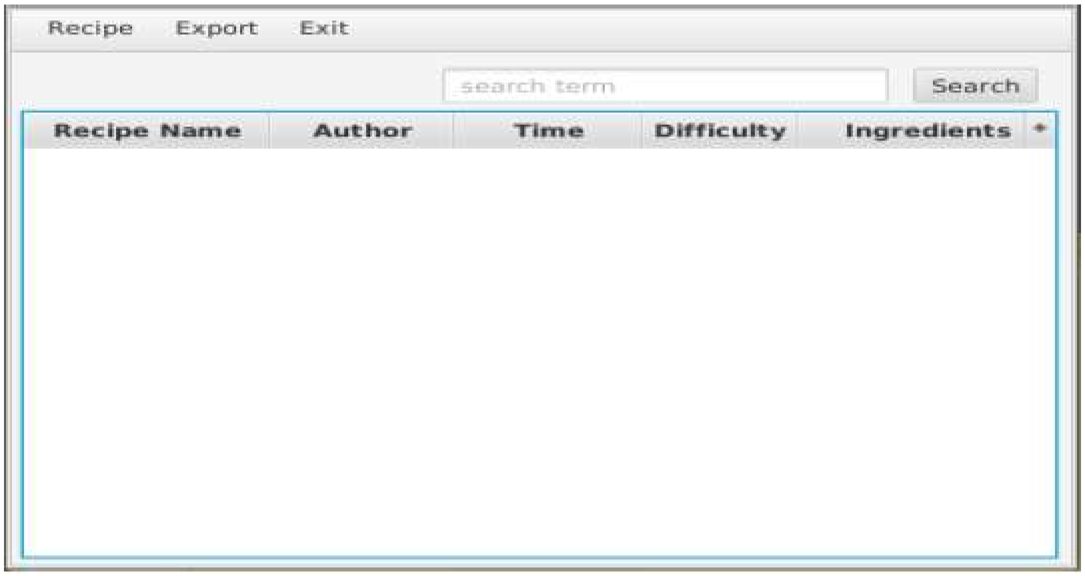
4.1 Inputs



(Note: GUI images represent prototype model and may be subject to change in final version)

The system for inputs will be done through a simple GUI. The user can select Recipe from the drop down menu at the top right, and select from Add, Edit, List, and other modules. A display box will be added for user input, allowing them to search through the database based on their query term.

4.2 Outputs



(Note: GUI images represent prototype model and may be subject to change in final version)

The output will be on the same page as the input, with the recipes being listed by their field terms, in accordance with their key field: recipeName. In addition, an export function will also allow the user to export recipes to text documents.

5 DETAILED DESIGN

5.1 Hardware Detailed Design

N/A

5.2 Software Detailed Design

Main – This module executes the program.

Search – This module allows the user to search table through select field types.

Browse – This module displays a list of all recipes for the user.

Display – This module provides the initial view that structures the interface for the user to interact with.

Input/Edit – These modules provide the method for creating, modifying, and removing recipes.

Input – This module provides a form for the user to fill in to allow them to add recipes to the database.

Edit – This module allows the user to select a recipe and change the field entries.

Import – This module allows the user to import recipes from a separate text file.

Export – This module allows the user to export recipes to a separate text file.

SQLite – This module connects the front end user interface with the database.

Warnings/Errors – This module provides warnings for the user when improper information is added.

5.3 Internal Communications Detailed Design

Each team will be able to build out their design with the knowledge that other teams will provide them with the available APIs to test their code. As data is passed from GUI to Interface, the interface level will automatically translate the data into the appropriate syntax as it is passed to the database.

6 EXTERNAL INTERFACES

The system is a standalone application running in the Java Virtual Machine, and has no external interface requirements.

6.1 Interface Architecture

N/A

6.2 Interface Detailed Design

N/A

7 SYSTEM INTEGRITY CONTROLS

Since the data used in the program is of the user's own design, there is no reason to restrict access of critical data items by user.

Additions, deletions, and updates will be controlled through standardization, in which duplicate entries cannot be made, and field entries will be given a post text that will prevent issues with certain query types (ex: recipeName = 'From' is changed to 'From_1' to prevent a syntax error).