

---

# **RECIPE REPOSITORY**

---

Version 1.0

Adam Howell  
Alexander Macwilliams  
Claire Breer  
Eliot Pearson  
Justin Helphenstine

09/06/2016

# VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Claire Breer	9/7/16			Initial Release

## Table of Contents

1	Introduction .....	5
1.1	Purpose .....	5
2	Scope .....	5
3	Test Items .....	5
3.1	Test Items Included .....	5
3.2	Items Excluded .....	5
4	Functional Testing .....	5
4.1	Definition .....	5
4.2	Test Risks / Issues .....	5
4.3	Items to be Tested .....	5
4.4	Test Approach(s) .....	6
4.4.1	Create Recipes .....	6
4.4.2	Search Recipes .....	6
4.4.3	Browse Recipes .....	6
4.4.4	Edit Recipes .....	6
4.4.5	Export Recipes .....	6
4.4.6	Delete Recipes .....	6
4.5	Test Regulatory / Mandate Criteria .....	6
4.6	Test Pass / Fail Criteria .....	7
4.7	Test Entry / Exit Criteria .....	7
4.8	Test Deliverables .....	7
4.9	Test Suspension / Resumption Criteria .....	7
4.10	Test Environmental / Staffing / Training Needs .....	7
5	User Interface Testing .....	7
5.1	Definition .....	7
5.2	Test Risks / Issues .....	7
5.3	Items to be Tested / Not Tested .....	8
5.3.1	Items to be Tested .....	8
5.4	Test Approach(s) .....	8
5.5	Test Regulatory / Mandate Criteria .....	8
5.6	Test Pass / Fail Criteria .....	8

## RECIPE REPOSITORY

5.7	Test Entry / Exit Criteria .....	8
5.8	Test Deliverables .....	8
5.9	Test Suspension / Resumption Criteria .....	8
5.10	Test Environmental / Staffing / Training Needs.....	8
6	Test Schedule .....	9
7	Recipe Repository Approval.....	10

# **1 INTRODUCTION**

## **1.1 PURPOSE**

This test plan describes the testing framework for the Recipe Repository application. This plan will address elements of the application to be tested. The test plan identifies the types of tests to be conducted and the scope.

## **2 SCOPE**

This test plan describes the items to be tested. The test items include modules created by the team for the Recipe Repository program. Third party modules will be excluded from the test.

# **3 TEST ITEMS**

## **3.1 TEST ITEMS INCLUDED**

Items included in the test plan are listed below:

- The GUI and associated Java code.
- The implementation of the sqlite database component.
- The code developed by the team.

## **3.2 ITEMS EXCLUDED**

The items listed below are excluded from the test plan because they are externally provided and assumed to be standard within the industry:

- Java library code.
- Database library code.

# **4 FUNCTIONAL TESTING**

## **4.1 DEFINITION**

Function testing will ensure that the program exhibits the behaviors specified by the project documentation. The functions of the program will be tested by using test data and manually verifying the results.

## **4.2 TEST RISKS / ISSUES**

None known

## **4.3 ITEMS TO BE TESTED**

The main functions of the program will be tested.

- Create Recipes
- Search Recipes
- Browse Recipes
- Edit Recipes
- Export Recipes
- Delete Recipes

#### **4.4 TEST APPROACH(S)**

The tester will use the provided test data to perform the tests in the order listed below.

##### **4.4.1 Create Recipes**

1. The tester will manually enter a recipe from the test data using the GUI.
2. The tester will submit the completed recipe.
3. The tester will verify that the recipe was created.
4. The tester will repeat the process with another set of test data, but leaving the recipe name blank.
5. The tester will verify that the recipe is not created.
6. The tester will re-enter the second recipe and enter the recipe name.
7. The tester will verify that the recipe is created.

##### **4.4.2 Search recipes**

1. The tester will search for the first recipe created by the recipe name.
2. The tester will verify that the search finds the first recipe.
3. The tester will search for the first recipe by the author.
4. The tester will verify that the search finds the first recipe.
5. The tester will search for the first recipe by an ingredient.
6. The tester will verify that the search finds the first recipe.
7. The tester will search for a recipe that doesn't exist by the recipe name.
8. The tester will verify that the non-existent recipe is not found.

##### **4.4.3 Browse Recipes**

1. The tester will browse through the recipes in the application.
2. The tester will verify that there are two recipes.

##### **4.4.4 Edit recipes**

1. The tester will edit the first recipe and change the name.
2. The tester will verify that the first recipe's name has changed.
3. The tester will change one ingredient of the first recipe.
4. The tester will verify that the recipe has changed.

##### **4.4.5 Export recipes**

1. The tester will export the first recipe.
2. The tester will verify that the first recipe was successfully exported.

##### **4.4.6 Delete Recipes**

1. The tester will delete the first recipe.
2. The tester will verify that the recipe was deleted.

#### **4.5 TEST REGULATORY / MANDATE CRITERIA**

None

#### 4.6 TEST PASS / FAIL CRITERIA

An item fails if upon action it does not meet intended results. Errors and program improvements will be posted to the team Github repository.

#### 4.7 TEST ENTRY / EXIT CRITERIA

Testing will begin at the release of any patch. Testing will be concluded after verification by each team member, and before the release of subsequent patches.

#### 4.8 TEST DELIVERABLES

The deliverable is an Excel spreadsheet of every action and the results of each test. A sample format is shown below.

Test	Step	Action	Expected Response	Pass/Fail	Notes
Create Recipe	1	Create Recipe	Recipe Created		

#### 4.9 TEST SUSPENSION / RESUMPTION CRITERIA

Testing is to be suspended if a feature failure is determined to be “fatal”, where the application is unable to be operated to an appreciable result.

#### 4.10 TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS

The functional test has the following system requirements:

- Computer Capable of running Java Runtime Engine version 1.8.0\_77 or higher.
- At least 200MB of main memory available for the application’s execution, with more being preferred. Larger recipe catalogs will benefit from additional memory.
- Minimum disk space of 100MB to store the recipe database.
- The system shall be usable with standard human input and output devices.

### 5 USER INTERFACE TESTING

#### 5.1 DEFINITION

User interface testing ensures that the graphical user interface is easy to use and that it presents data in a clear and coherent manner.

#### 5.2 TEST RISKS / ISSUES

Risks and issues are deemed to be negligible due to the limited scope of anticipated use of software. Operating System integration is minimal and limits software conflict, and hardware risk is minimal due to the limit of anticipated use (personal use for small scale simulation)

### 5.3 ITEMS TO BE TESTED / NOT TESTED

#### 5.3.1 ITEMS TO BE TESTED

The tester will use the provided test data to verify that the user interface renders correctly and is easy to use.

#### 5.4 TEST APPROACH(S)

1. The tester will launch the application and verify that the GUI renders correctly.
2. The tester will re-size the window five times and verify that user interface re-sizes without distortion.
3. The tester will provide feedback on the user interface related to ease of use and clarity.

#### 5.5 TEST REGULATORY / MANDATE CRITERIA

None

#### 5.6 TEST PASS / FAIL CRITERIA

The test will be considered a FAIL if the GUI does not render correctly, it is not easy to use, or it does not present data in a user-friendly format. Errors and program improvements will be posted to the team Github repository.

#### 5.7 TEST ENTRY / EXIT CRITERIA

Testing begins when the application is considered to have entered phase 1 where an applicable GUI is interacting with functional data objects.

#### 5.8 TEST DELIVERABLES

The test will be solely a go/no report amongst testers, with results suggesting feature or further design improvement. The deliverable is an Excel spreadsheet of every action and the results of each test. A sample format is shown below.

Test	Step	Action	Expected Response	Pass/Fail	Notes
GUI	1	Launch Application	GUI renders correctly		

#### 5.9 TEST SUSPENSION / RESUMPTION CRITERIA

Testing will be suspended if the test results in fatal error where the application is unable to operate at set speed/complexity or fails to render.

#### 5.10 TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS

The functional test has the following system requirements:

- Computer Capable of running Java Runtime Engine version 1.8.0\_77 or higher.



- At least 200MB of main memory available for the application's execution, with more being preferred. Larger recipe catalogs will benefit from additional memory.
- Minimum disk space of 100MB to store the recipe database.
- The system shall be usable with standard human input and output devices.

## 6 TEST SCHEDULE

Task	Date Due	Owner
Test Plan	9/11/2016	Test Lead
Function Testing	TBD	Team Members
GUI Testing	TBD	Team Members

## 7 RECIPE REPOSITORY APPROVAL

The undersigned acknowledge they have reviewed the Recipe Repository document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
 Print Name: \_\_\_\_\_  
 Title: \_\_\_\_\_  
 Role: \_\_\_\_\_ Design \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
 Print Name: \_\_\_\_\_  
 Title: \_\_\_\_\_  
 Role: \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
 Print Name: \_\_\_\_\_  
 Title: \_\_\_\_\_ Programmer \_\_\_\_\_  
 Role: \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
 Print Name: \_\_\_\_\_  
 Title: \_\_\_\_\_ Test Lead \_\_\_\_\_  
 Role: \_\_\_\_\_