

Kelly May

Pantry to Plate - Project Documentation



Pantry to Plate



Executive Summary

Food waste is an enormous problem in the United States, and other first world countries. Studies have shown that up to 40% of the food in the United States is never actually eaten (NRDC, 2020). This wastage comes from massive portions at restaurants that will throw out leftovers, grocery stores that overstock their shelves to provide variety, and farmers who cannot sell produce that is not perfect-looking. Most of all, food is wasted at home, with many items forgotten on shelves or in the fridge and only remembered when they are no longer edible.

Food wastage is costly for a variety of reasons. Firstly, wasted, rotting food emits greenhouse gases. This greenhouse effect is considered a major factor for global warming (Climate Central, 2020). Secondly, food wastage wastes the consumer's money because consumers are essentially throwing out their money every time, they throw out a rotten piece of purchased produce. Why do we waste food if it is so costly? There are many reasons why consumers end up wasting food, including being unaware of the amount of wastage they are creating, over shopping in order to maintain the ideal "image" of a full pantry, and a common desire to eat only "perfect" foods that are at the peak of ripeness and without flaw (Jackson, 2019). One of the ways consumers can help reduce this food wastage is by integrating awareness of what foods we are throwing away, when can help consumers to start making smarter decisions on what to buy, and what needs to be eaten before it is too late. That is where Pantry to Plate comes in.

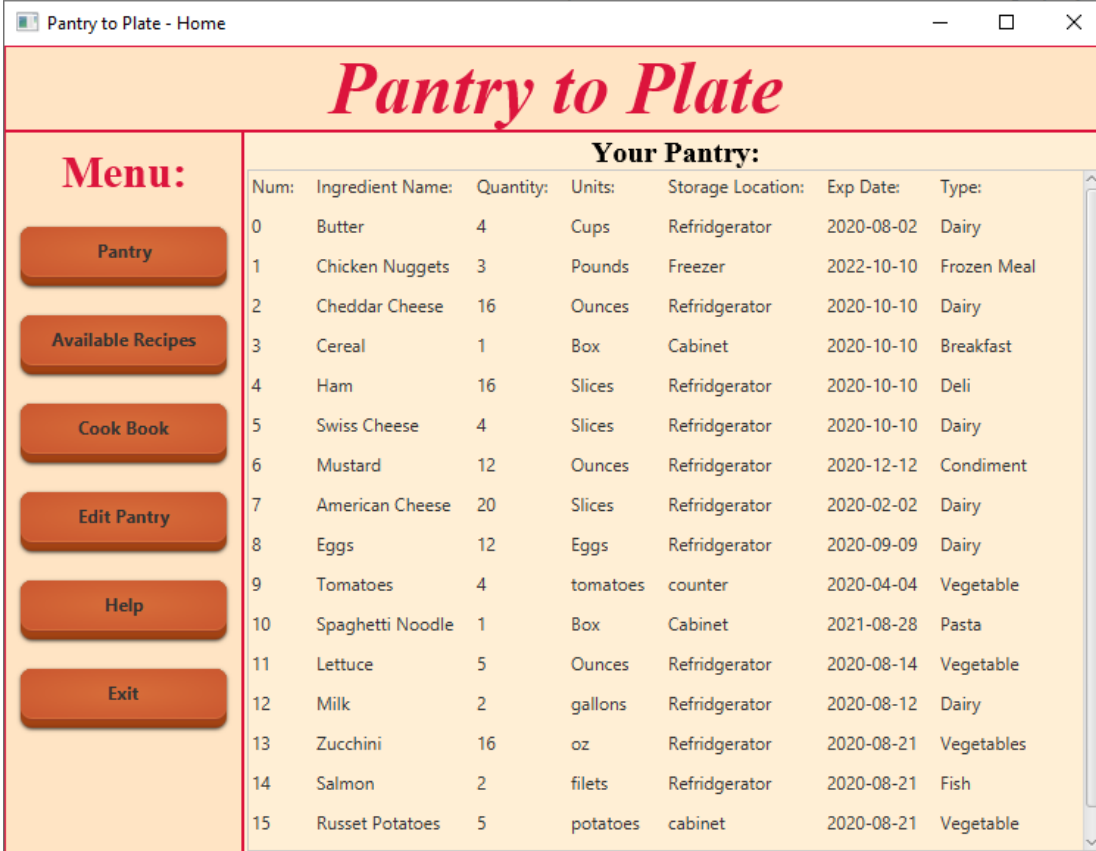
Pantry to Plate is an application that will be designed to help lessen the amount of food waste in the home by providing consumers knowledge about what they already have in their kitchens, and what they can do with what they have. Pantry to Plate will store information about the food items entered by the customer, including amount, where it is stored (refrigerator, freezer, or pantry shelves), and expiration dates. With this information, Pantry to Plate will match the food items logged in the application with recipes that can be made without taking a trip to the grocery store.

The basic model of Pantry to Plate is a desktop platform that has a database, which users can enter both ingredient items such as produce, spices, grains, etc., and enter recipes with ingredients, their amount, and cooking instructions. The application will search for recipes that utilize only food items stored in the ingredients database and display those recipe options to the user. This will give the user options for what to cook with the ingredients that are readily available to them.

The goal of Pantry to Plate is to lessen the number of trips to the grocery store that a consumer will make by providing the consumer with knowledge of delicious recipes that they can make with what is currently in their kitchen. This will lessen the number of food items that will be wasted by the user, which will in turn save the user money, unnecessary trips to the store, and reduce food wastage that will negatively impact the environment. Consumers looking to either lower their food costs or lower their impact on the global food wastage problem would greatly benefit from Pantry to Plate, because Pantry to Plate will provide them useful information on what foods they have, and what delicious things they can do with that food.

Scenario 1 | View Pantry

To view the current inventory, the user must click the “Pantry” button in the menu pane. The central pane will show a table containing all inventory items.



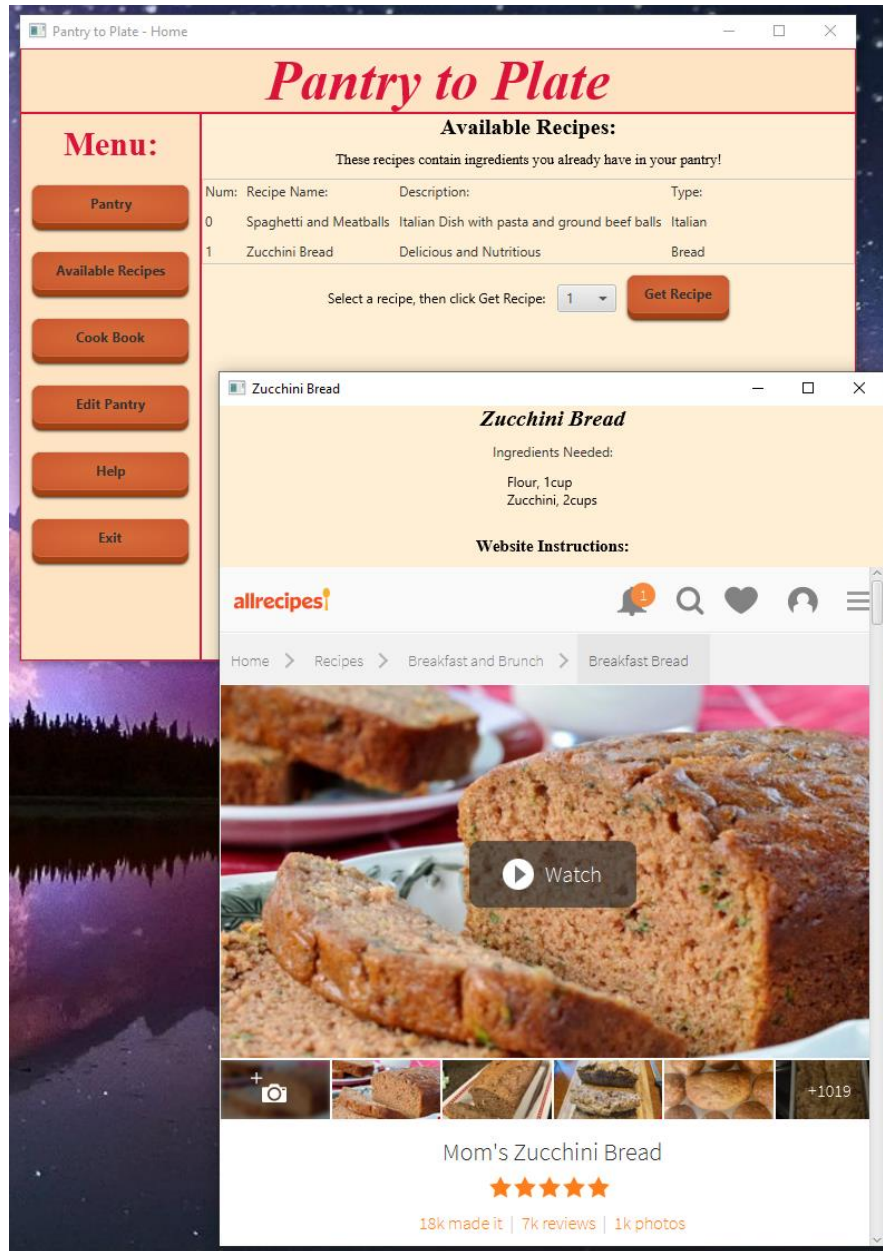
<i>Pantry to Plate</i>						
Menu:		Your Pantry:				
Num:	Ingredient Name:	Quantity:	Units:	Storage Location:	Exp Date:	Type:
0	Butter	4	Cups	Refridgerator	2020-08-02	Dairy
1	Chicken Nuggets	3	Pounds	Freezer	2022-10-10	Frozen Meal
2	Cheddar Cheese	16	Ounces	Refridgerator	2020-10-10	Dairy
3	Cereal	1	Box	Cabinet	2020-10-10	Breakfast
4	Ham	16	Slices	Refridgerator	2020-10-10	Deli
5	Swiss Cheese	4	Slices	Refridgerator	2020-10-10	Dairy
6	Mustard	12	Ounces	Refridgerator	2020-12-12	Condiment
7	American Cheese	20	Slices	Refridgerator	2020-02-02	Dairy
8	Eggs	12	Eggs	Refridgerator	2020-09-09	Dairy
9	Tomatoes	4	tomatoes	counter	2020-04-04	Vegetable
10	Spaghetti Noodle	1	Box	Cabinet	2021-08-28	Pasta
11	Lettuce	5	Ounces	Refridgerator	2020-08-14	Vegetable
12	Milk	2	gallons	Refridgerator	2020-08-12	Dairy
13	Zucchini	16	oz	Refridgerator	2020-08-21	Vegetables
14	Salmon	2	filets	Refridgerator	2020-08-21	Fish
15	Russet Potatoes	5	potatoes	cabinet	2020-08-21	Vegetable

Scenario 2 | View Available Recipes

To view the recipes that pair with items in the inventory, the user must click “Available Recipes” in the menu pane. The central pane will show a table containing all available recipes.

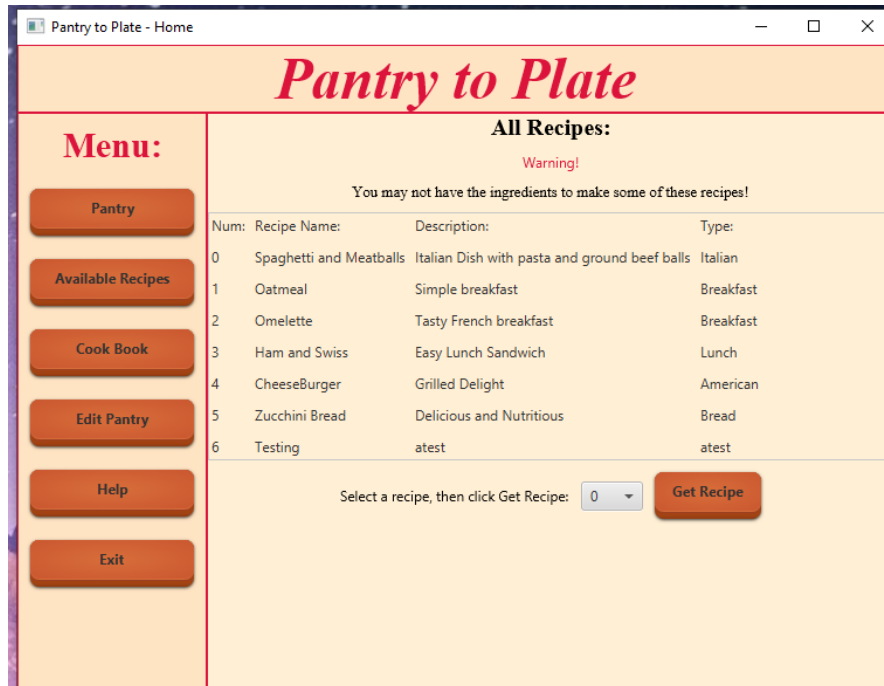


The user can use the drop-down menu to select a number representing a recipe, then click "Get Recipe". This will pop up a new page, describing the recipe, and showing the webpage for the recipe.

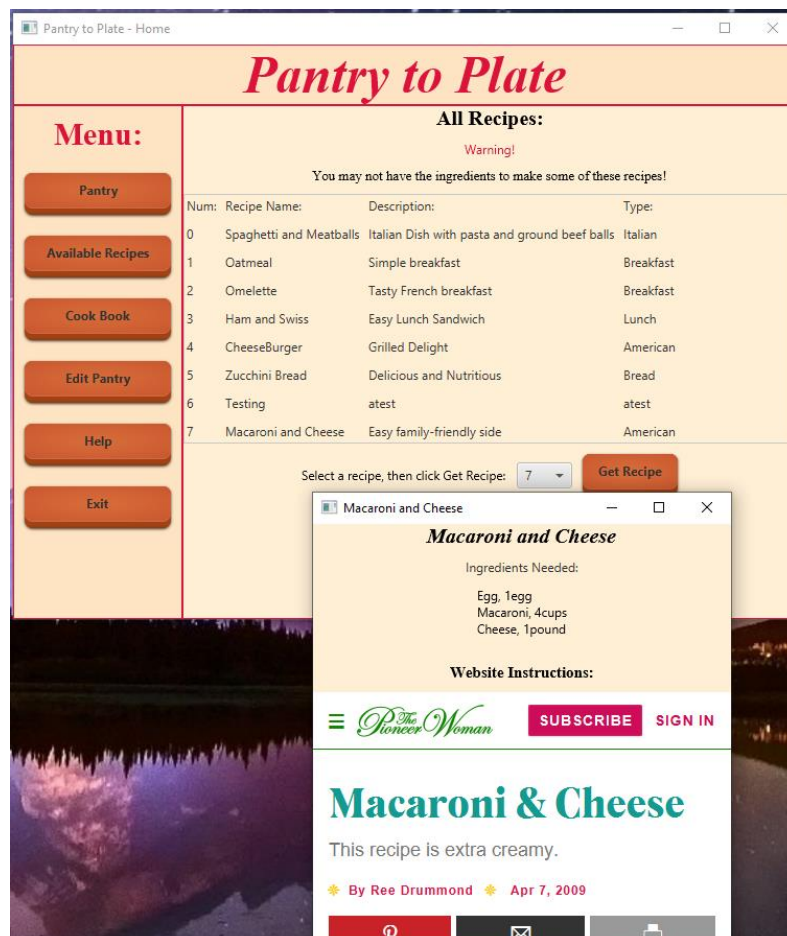


Scenario 3 | View All Recipes

To view all recipes, regardless of available ingredients, the user must click “Cook Book” in the menu pane. The central pane will show a table with all the recipes, similar to the available recipe page.

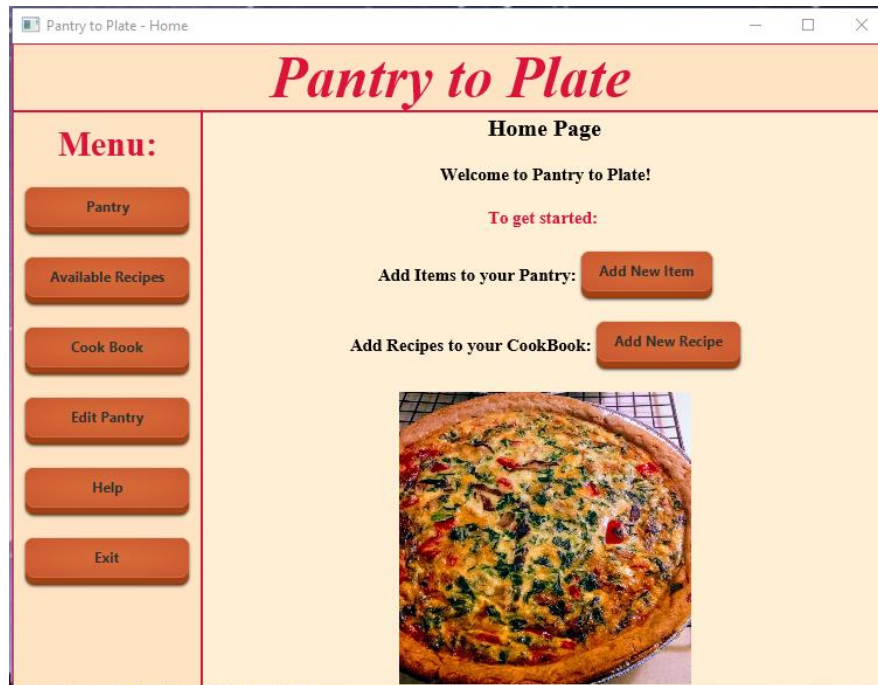


The user can select a recipe, using the drop-down menu, and clicking “Get Recipe”, and a page just like the one found in Available Recipes, will appear.



Scenario 4 | Quick- insert Inventory or Recipe

In the home page, that appears when the user opens Pantry To Plate, the user can click “Add New Item” to quickly add items into the Pantry. The user can also click “Add New Recipe” to quickly add a new recipe into the cookbook. Both of these functionalities are also available in the Edit Pantry page, but are also available on the home page, for quick access. Please see Scenario 5 and Scenario 6 to see how items are added into the pantry and the cookbook.



Scenario 5 | Add New Item into Pantry

The user can either click “Add New Item” on the home page, or navigate to “Edit Pantry”, then click “Add New Item”. The user will see the following screen:

The screenshot shows a web application window titled "Add Item to Inventory". The main heading is "Insert an Item into the Inventory" in a bold, italicized serif font. Below the heading is the instruction "Enter the values of the item below:". The form contains several input fields: "Item Name:" with a text box, "Item Quantity:" with a text box containing the number "1" and a spinner control, "Quantity Units:" with a text box, "Item Storage Loca..." with a text box, "Expiration Date:" with a text box and a calendar icon, and "Item Type:" with a text box. At the bottom of the form are two orange buttons: "Enter" and "Exit".

The user can enter in the information for an item to be added into the inventory. Then the user clicks “Enter”. A second screen appears, asking the user to confirm the information they entered is correct. If “Confirm” is clicked, the item is added to the database. If “Edit” is clicked, the user can go back to the previous page, and edit their answers prior to adding to the database. The User can click “Exit” on the editor page to close it.

The image displays two side-by-side screenshots of a web application interface for inventory management.

The left screenshot, titled "Add Item to Inventory", shows a form titled "Insert an Item into the Inventory". Below the title, it says "Enter the values of the item below:". The form contains the following fields and values:

- Item Name: Baking Soda
- Item Quantity: 6
- Quantity Units: oz
- Item Storage Loca...: Cabinet
- Expiration Date: 8/21/2020
- Item Type: Baking

At the bottom of the form are two buttons: "Enter" and "Exit".

The right screenshot, titled "Verify Data", shows a confirmation page titled "Add to Inventory Confirmation Page". It displays the same information entered in the previous form:

- Item Name: Baking Soda
- Quantity: 6 oz
- Storage Location: Cabinet
- Expiration Date: 2020-08-21
- Type of Item: Baking

At the bottom of the confirmation page are two buttons: "Confirm" and "Edit".

Scenario 6 | Add New Recipe in Cook Book

The User can either click “Add New Recipe” from the home page, or navigate to “Edit Pantry”, then click “Add New Recipe”. The User will see the following screen:

The image shows a screenshot of a web application window titled "Add Recipe to Cookbook". The form is titled "Add a new Recipe to the Cook Book" and prompts the user to "Enter recipe information below:". The form contains the following fields and values:

- Recipe Name: (empty text box)
- Description: (empty text box)
- Recipe Type: (empty text box)
- Link to Recipe: (empty text box)
- Number of Ingredients: 1

Below the "Number of Ingredients" field is a button labeled "Enter Ingredient Number". At the bottom of the form are two buttons: "Enter Recipe Into CookBook" and "Close".

The user must first enter the general information about the recipe, such as recipe name and description. The user then, using the spinner, selects the number of ingredients in the recipe, then clicks “enter ingredient number”. The screen will populate additional text fields and spinners to enter information about each ingredient. The user will fill all these text fields, then click “Enter Recipe into CookBook”. A confirmation screen will appear:

The screenshot displays a web application window titled "Add Recipe to Cookbook". The main form is titled "Add a new Recipe to the Cook Book" and prompts the user to "Enter recipe information below:". The form includes a "Number of Ingredients" spinner set to 3, an "Enter Ingredient" button, and three rows of input fields for ingredients. The first row is for "Egg" (Quantity: 1, Units: egg), the second for "Macaroni" (Quantity: 4, Units: cups), and the third for "Cheese" (Quantity: 1, Units: pound). At the bottom are "Enter Recipe Into CookBook" and "Close" buttons. A modal dialog box titled "Add New Recipe Confirmation Page" is overlaid, showing the recipe details: "Recipe Name: Macaroni and Cheese", "Description: Easy family-friendly side", "Type of Recipe: American", and "Website URL: https://www.thepioneerwoman.com/food-cooking/recipe...". It also lists the ingredients: "1- Ingredient: Egg", "1- Quantity: 1 egg", "2- Ingredient: Macaroni", "2- Quantity: 4 cups", "3- Ingredient: Cheese", and "3- Quantity: 1 pound". At the bottom of the dialog are "Confirm" and "Edit" buttons.

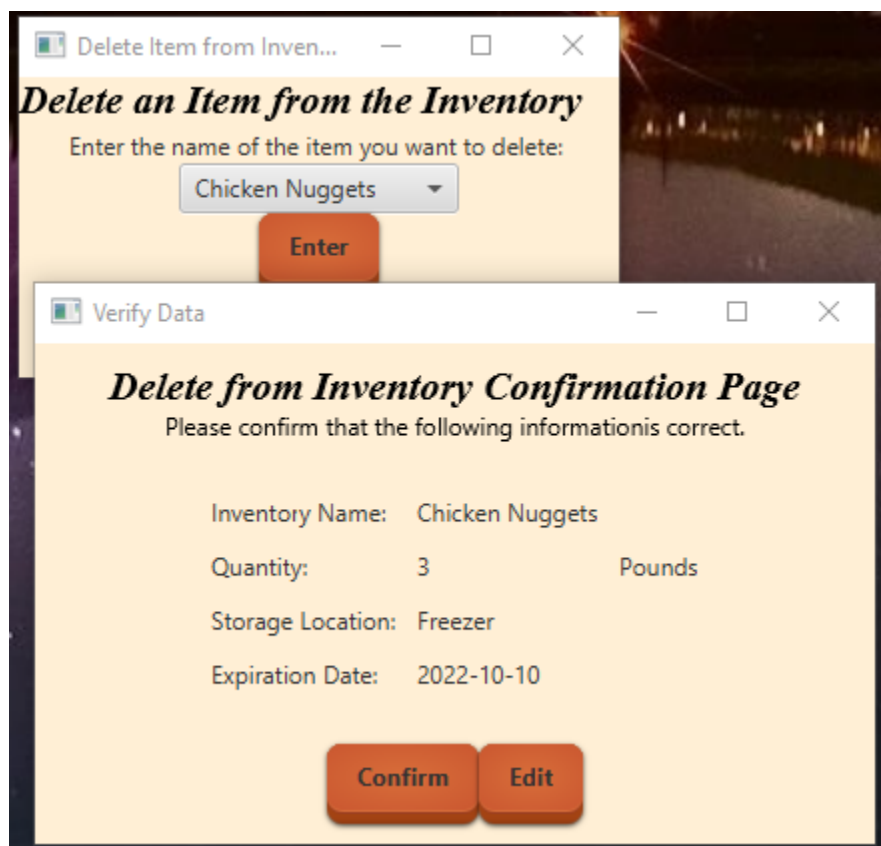
If the information is correct, the user clicks “Confirm” and the recipe, and ingredients are added to the database. Otherwise, the user can click “Edit”, to edit the entry prior to adding to the database. The user can click “Close” to leave the add recipe editor completely.

Scenario 7| Delete Item from Inventory

The user clicks “Edit Pantry” from the menu pane, then clicks “Delete Item” in the central pane. The following screen will appear:



Using the drop-down menu, the user can select the item in the inventory they wish to delete. The user clicks "Enter" and a confirmation page will appear, showing details about the ingredient. If the item is correct, the user clicks "Confirm", and the item is deleted from the database. The user can click "Edit" to close the confirmation page without deleting an item, and select a different item from the drop down menu.

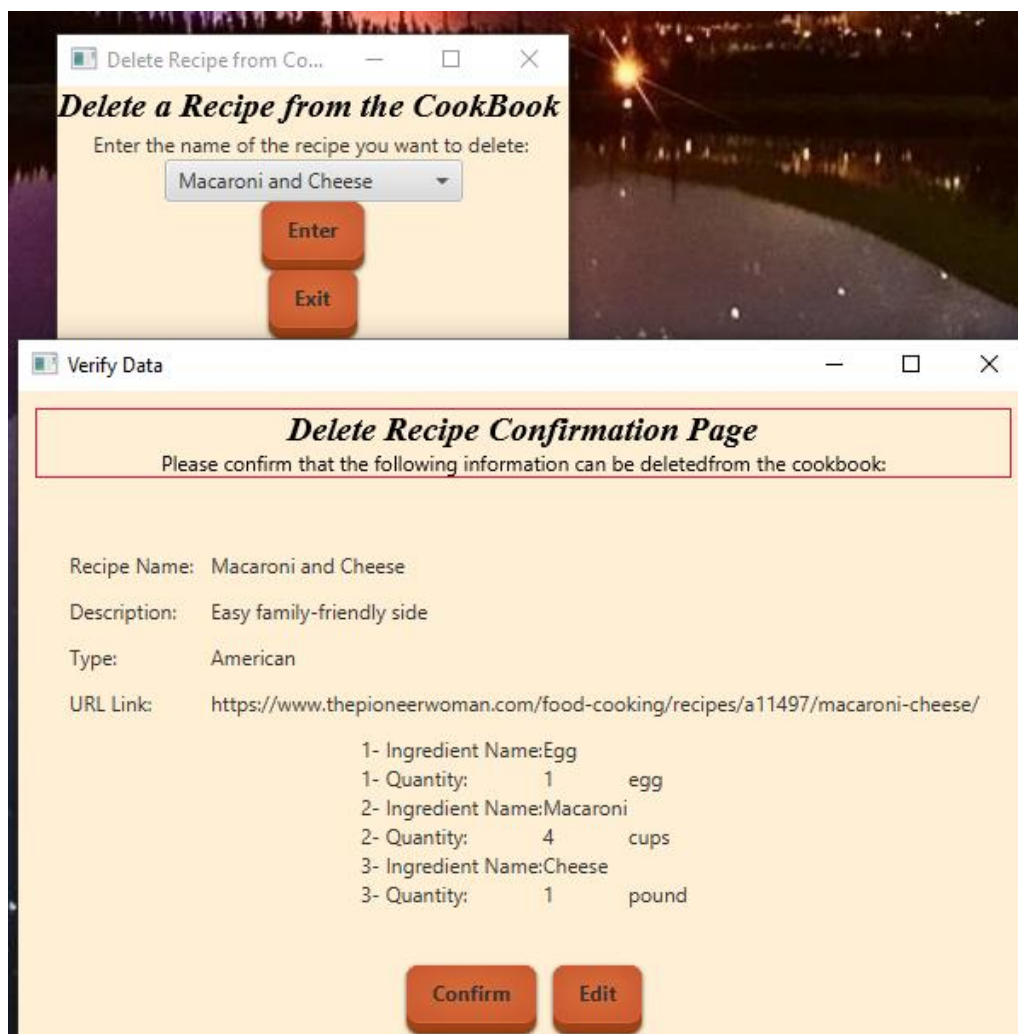


Scenario 8| Delete a Recipe from the CookBook

The User clicks "Edit Pantry" in the menu pane, then clicks "Delete Recipe" in the central pane. The following screen will appear:



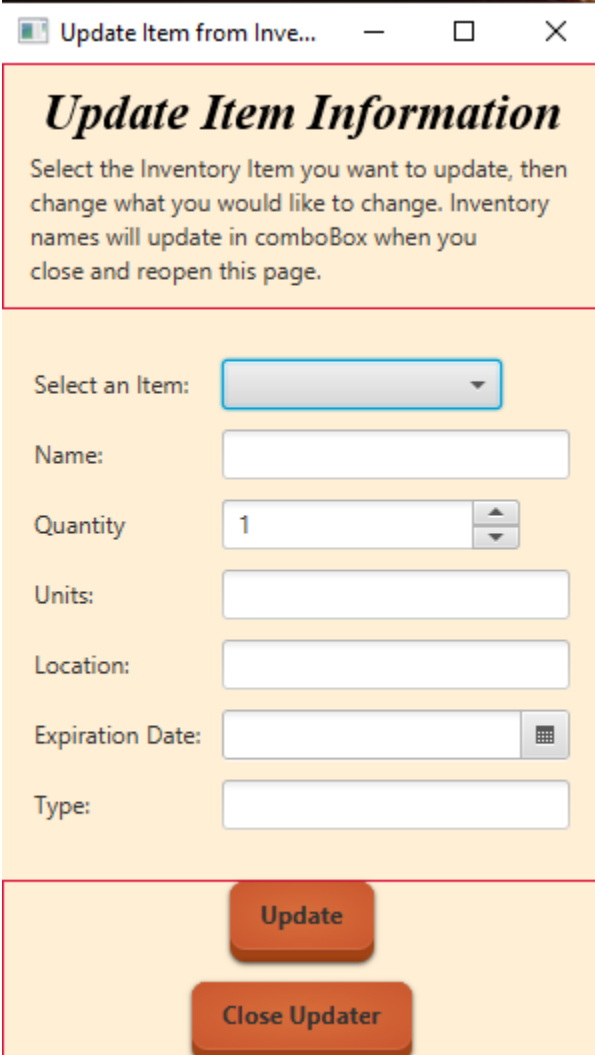
The user uses the drop-down menu to select the recipe they wish to delete, then clicks "Enter". A confirmation page will appear, detailing information about the recipe. If the recipe is correct, the user clicks "Confirm", and the recipe, and the associated recipe ingredients will be deleted from the database. The user can click "Edit" to exit the confirmation page without deleting the recipe from the database.



The user can click "Exit" in the editor page to leave the close the Delete Recipe page.

Scenario 9| Modify an Existing Inventory Item

The user clicks “Edit Pantry” in the menu pane, then clicks “Modify Item” in the central pane.



The screenshot shows a window titled "Update Item from Inve...". Inside, there is a section titled "Update Item Information" with a descriptive paragraph: "Select the Inventory Item you want to update, then change what you would like to change. Inventory names will update in comboBox when you close and reopen this page." Below this, there are several input fields: "Select an Item:" with a dropdown menu, "Name:" with a text box, "Quantity" with a spinner box showing "1", "Units:" with a text box, "Location:" with a text box, "Expiration Date:" with a text box and a calendar icon, and "Type:" with a text box. At the bottom, there are two buttons: "Update" and "Close Updater".

In the editor page that appears, the user selects an item in the drop-down menu. The other fields will be filled with the existing information about that item. The user can edit any field with new information, then click “Update”. A confirmation page will appear, showing the user the new information to be added for that item. If the information is correct, the user can click “Confirm”, and the changes will be made in the database. Otherwise, the user can click “edit” to close the confirmation page and continue editing their item. The user can click “Close Updater” to close the editor page.

The screenshot displays two overlapping windows from a software application. The background is a dark, atmospheric image of a night scene with lights reflecting on water.

Update Item from Inve... (Title bar)

Update Item Information

Select the Inventory Item you want to update, then change what you would like to change. Inventory names will update in comboBox when you close and reopen this page.

Select an Item: Baking Soda

Name: Baking Soda

Quantity: 4

Units: ounces

Location: Cabinet

Expiration Date: 8/27/2020

Type: Baking

Update

Close Updater

Verify Data (Title bar)

Delete from Inventory Confirmation Page

Item Name: Baking Soda

Quantity: 4 ounces

Location: Cabinet

Expiration Date: 2020-08-27

Type: Baking

Confirm Edit

Scenario 10 | Modify an Existing Recipe

The user clicks “Edit Pantry” in the menu pane, then “Modify Recipe” in the central pane. The following editor page appears:

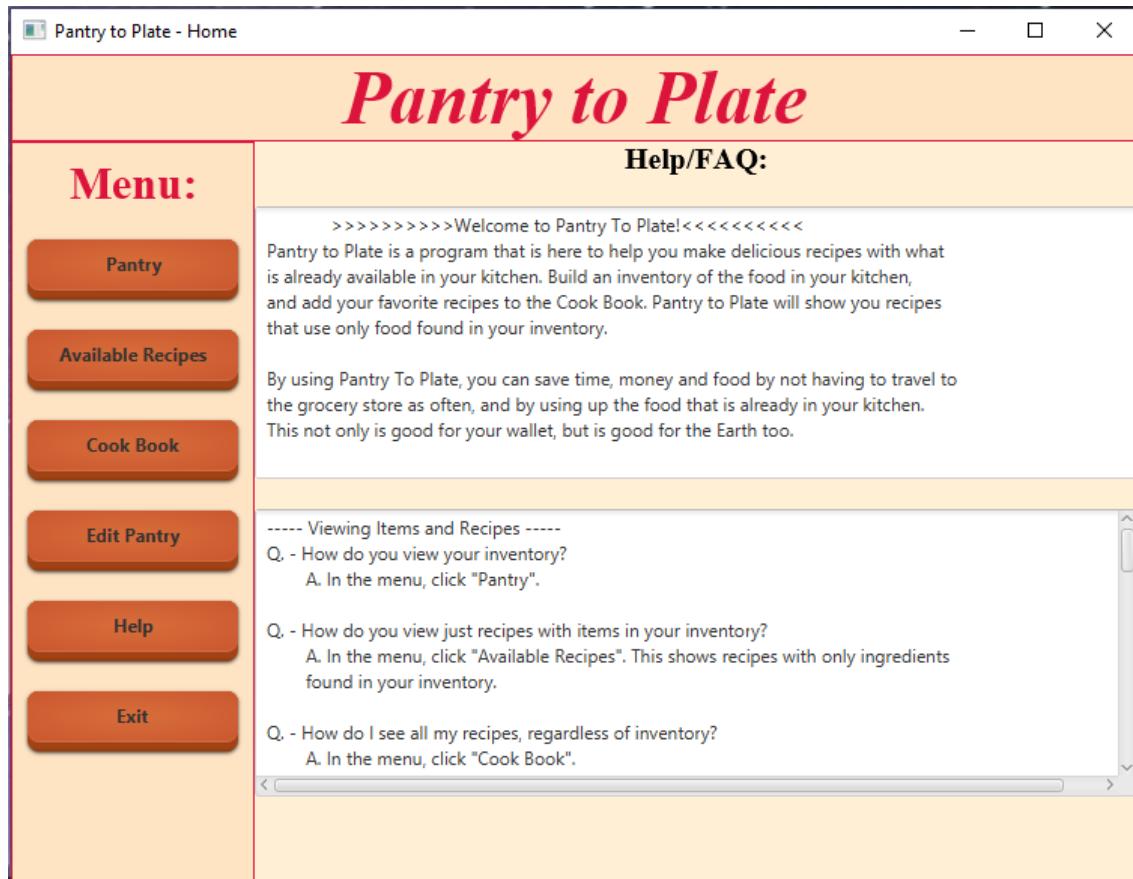
The screenshot shows a web application window titled "Update Recipe". Inside, there is a section titled "Update Recipe Information" with the instruction: "Select a Recipe you wish to update, then update any field(s). Press Update Recipe when you are done." Below this instruction, there is a dropdown menu labeled "Select a Recipe:". Underneath the dropdown, there are four text input fields labeled "Recipe Name:", "Description:", "Type:", and "Link:". At the bottom of the window, there are two buttons: "Update Recipe" and "Close".

The user selects a recipe from the drop-down menu, and the recipe information, as well as any ingredient information will appear in editable text fields. The user can edit any of the fields, then click "Update Recipe". A confirmation page will appear, detailing the changes. If the user clicks "Confirm", the changes will be made in the database. Otherwise, the user can click "Edit" to return to the editor page. The user can click "Close" to leave the editor page completely.

This screenshot shows the "Update Recipe" window with the "Update Recipe Confirmation Page" overlay. The "Update Recipe" window has the "Select a Recipe:" dropdown set to "Macaroni and Cheese". The "Recipe Name:" field contains "Macaroni and Cheese", "Description:" contains "Easy family-friendly side", "Type:" contains "American", and "Link:" contains "https://www.thepioneerwoman.com/food-cooking/recipes/a11497/macaroni-cheese/". Below these are three ingredient entries: "1- Ingredient Name: Egg", "Quantity Required: 1", "Units: egg"; "2- Ingredient Name: Macaroni", "Quantity Required: 4", "Units: cups"; and "3- Ingredient Name: Cheese", "Quantity Required: 1", "Units: pound". At the bottom are "Update Recipe" and "Close" buttons. The "Update Recipe Confirmation Page" overlay has the title "Update Recipe Confirmation Page" and the instruction "Please confirm that the following information is correct." It displays the same recipe information as the "Update Recipe" window. At the bottom of the overlay are "Confirm" and "Edit" buttons.

Scenario 11 | View Frequently Asked Questions and About

The User clicks “Help” to view information about Pantry To Plate, and FAQs. There is also contact information, should the user need more help with the system.

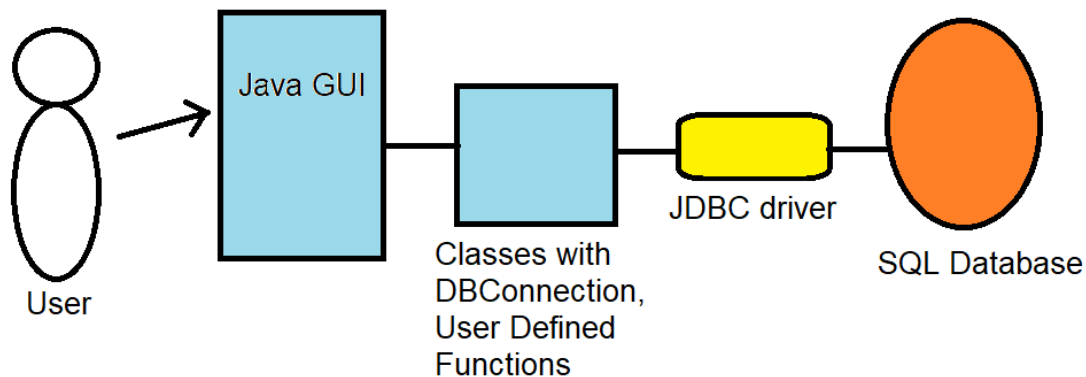


Scenario 12 | Exiting the Program

The User clicks the “Exit” button in the Menu pane, and the program will close.

System Architecture

The User interacts directly with the GUI, which is built in Java. There are backend Java classes which access a JDBC Driver, which connects the user interface with a SQL database. The SQL database is where Inventory and Recipe data is stored.



Source Code Structure

Source code structure introduction. The following is a summary of the source code directories and their contents:

Code Directory	
Directory	Usage
Java	Contains all the Java source code, folders/files.
Java>src	Contains the Java source code, and image used in user interface
Java>build	Java build files
Java> dist	Contains Java executable JAR file and .EXE executable file
Java>nbproject	Miscellaneous java config files
Java>test	Old, tested, and unused source files- saved here for potential future use
Project_Managment	Folder that contains the project design, documentation, and demo videos
SQL_Database	Contains scripts for the SQL database, that needs to be built for the program to work

Highlighted rows indicate directories containing source code.

Executables

Describe the executables that are a part of this project. If there are multiple executables break out each one into its own heading and give its name and a description of what function it plays in the system.

PantryToPlate (PantryToPlate.exe)

Executable file created using Launch4j.

PantryToPlate(PantryToPlate.jar)

JAR file created by NetBeans, can also be used to execute the program.

SQL JDBC Driver(sqljdbc4-2.0.jar)

JAR file used to connect SQL server to Java program.

Code Architecture

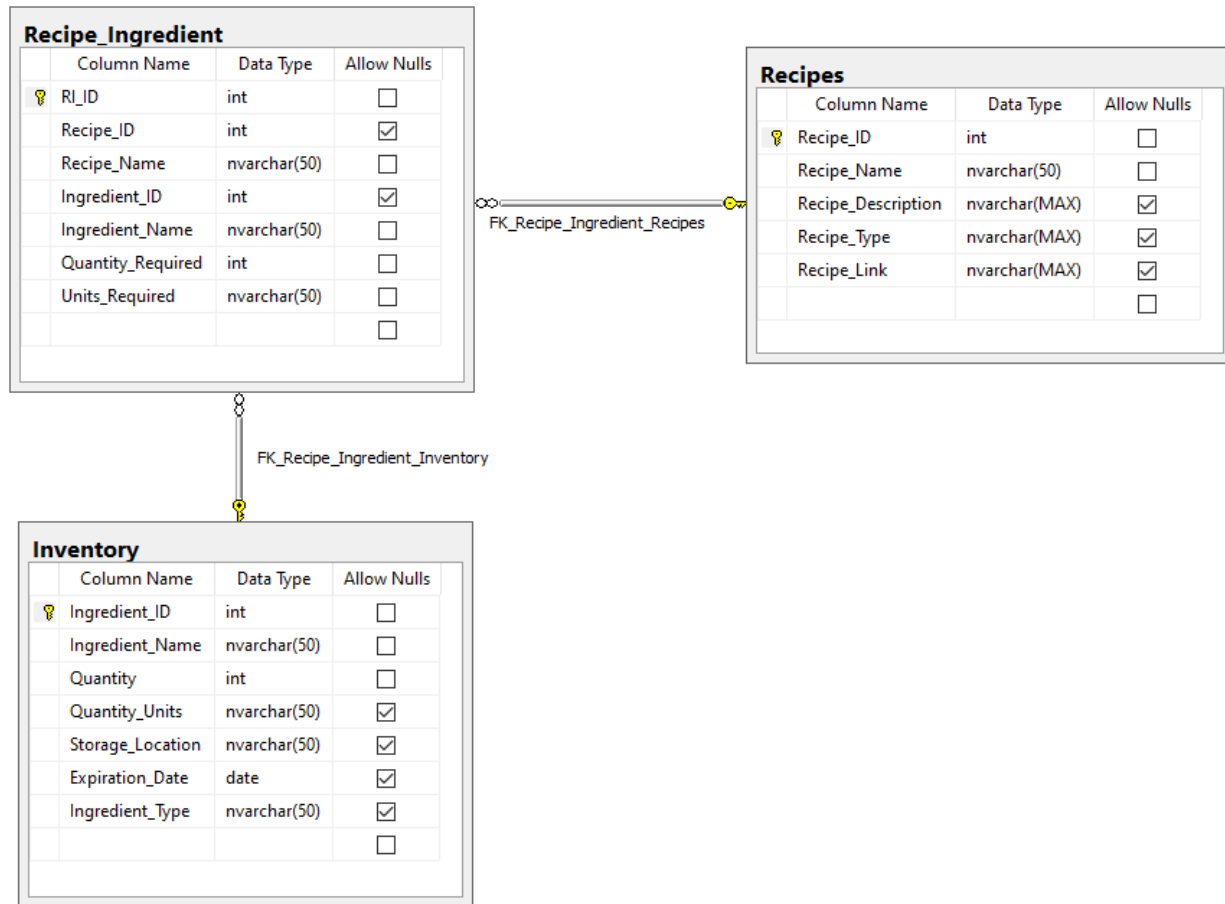
Explain how the system is put together. Provide an introduction and then dig into the database or data store design and then explore the internals of the code.

The inventory and recipe data is stored in a SQL database, and is connected to the Java program using a JDBC driver. The JDBC driver is accessed using the Java class “DBConnection”. This class connects the Java program with the database. Subsequent classes access stored procedures or have their own defined query functions. Classes that access the DBConnection class are:

- | | |
|--------------------------|---------------------|
| 1. ViewRecipes | 7. UpdateInventory |
| 2. ViewRecipeIngredients | 8. InsertRecipe |
| 3. ViewPantry | 9. InsertInventory |
| 4. ViewAvailableRecipes | 10. DeleteRecipe |
| 5. Verify | 11. DeleteInventory |
| 6. UpdateRecipe | |

These classes all contain methods that either call a stored procedure or contain a prepared statement which utilizes a user defined function/query directly.

Database or Data Store



The database was built in SQL using Microsoft SQL Server 2014 Management Studio Express. There are three tables within the database, Inventory, Recipes and Recipe_Ingredient. Inventory stores ingredient items found in the user's kitchen. Recipes holds general information about a recipe. Recipe_Ingredient stores the ingredients needed for each individual recipe. If that ingredient exists in the inventory, an Ingredient_ID from the Inventory will be paired to a Recipe_Ingredient.

Views, Stored Procedures and User Defined Functions

There are a few stored procedures, which are accessed by the Java portion of the program to maintain the data accessed by the user:

1. `dbo.Delete_Inventory` – This stored procedure deletes an item in the inventory based on a given Ingredient_Name
2. `dbo.Delete_Recipe` – This stored procedure deletes any Recipe_Ingredient associated with a Recipe, and then deletes the Recipe based on a given Recipe_Name
3. `dbo.Insert_Inventory` – inserts a new inventory item into `dbo.Inventory`, including Ingredient_Name, Quantity, Quantity_Units, Storage_Location, Expiration_Date and Ingredient_Type.
4. `dbo.Insert_Recipe_Info` – inserts general information about a recipe into `dbo.Recipes`. This includes Recipe_Name, Recipe_Description, Recipe_Type, Recipe_Link.
5. `dbo.Insert_Recipe_Ingredient` – inserts information about each ingredient associated with a given recipe. It will insert Ingredient_Name, Quantity_Required and Units_Required. This is accessed after `dbo.Insert_Recipe_Info` is used.

6. `dbo.Update_Recipe_Ingredient` – this should be used after each time `dbo.Insert_Recipe_Ingredient` is accessed. This stored procedure fills in `Recipe_ID` and `Ingredient_ID` information into `Recipe_Ingredient`, based on the `Recipe_Name` and `Ingredient_Name` which were inserted previously with `dbo.Insert_Recipe_Ingredient`.

There are also multiple user-defined functions within the Java program, which are found within the 11 classes mentioned previously in the Code Architecture summary.

1. `ViewRecipes`:
 - a. `setRecipes` – print all the information in the Recipes table of the database.
 - b. `showARecipe` – displays an individual recipe, and its `recipe_ingredient` information.
2. `ViewPantry`:
 - a. `setInventory` – print all the information from the Inventory table of the database
3. `ViewAvailableRecipes`:
 - a. `setAvailables` – print only the recipe information where all ingredient items are also found in the inventory.
 - b. `showARecipe`- displays an individual recipe, and its `recipe_ingredient` information
4. `ViewRecipeIngredients`:
 - a. `setIngredients` – Shows all the ingredients associated with a given recipe
5. `Verify`:
 - a. `recipeExists` – returns a `Recipe_ID` number if a recipe with the `recipe_name` given already exists.
6. `UpdateRecipe`:
 - a. `setRecipeNames` – gets all the recipe names from the recipes table, and stores them in an `ArrayList`
 - b. `setRecipeFromSQL` – selects all the information for a given recipe so that it can be viewed, and later updated
 - c. `setIngredientsFromSQL` – selects all the ingredient information from `Recipe_Ingredient` for a given recipe, and stores that information into `ArrayLists`
 - d. `updateRecipe` – takes new information, and updates an existing recipe in `Recipes` with that information.
 - e. `updateRI` – takes new information, and updates an existing ingredient associated with a recipe, with that information.
 - f. `updateIngredients` – calls the `Update_Recipe_Ingredient` stored procedure
7. `UpdateInventory`:
 - a. `setInventoryNames` – gets all the inventory names from the Inventory table and stores them in an `ArrayList`.
 - b. `setItemsFromSQL` – gets all the inventory information, and stores it into multiple `ArrayLists`
 - c. `update` – takes new information and updates an existing inventory item in the Inventory table with that information
8. `DeleteInventory`:
 - a. `Delete` – calls the `Delete_Inventory` stored procedure
9. `DeleteRecipe`:
 - a. `Delete` – calls the `Delete_Recipe` stored procedure
10. `InsertInventory`:
 - a. `Insert` – calls the `Insert_Inventory` stored procedure
11. `InsertRecipe`
 - a. `addRecipe` – calls the `Insert_Recipe_Info` stored procedure
 - b. `updateIngredients` – calls the `Update_Recipe_Ingredient` stored procedure

Programming Language | Java

The program is written primarily in Java, using JRE version 1.8.0 on NetBeans. The database is written in Microsoft SQL Server 2014 Management Studio Express. To link the Java program to the database, sqljdbc_auth.dll was required.

Project Classes

Classes within the project are used to abstract re-usable pieces of code. Classes are also used to group related values, known as properties. The project utilizes these classes:

Delete Inventory Actionable | ActionsDeleteInventory.java

Houses Editor GUI and confirmation pages, as well as controls actions needed to access SQL classes for deleting inventory information.

Delete Recipe Actionable | ActionsDeleteRecipe.java

Holds Editor GUI and confirmation page as well as controls access to SQL classes for deleting recipe information.

Insert Inventory Actionable | ActionsInsertInventory.java

Holds Editor GUI and confirmation page, as well as controls access to SQL classes for inserting an ingredient into the database

Insert Recipe Actionable | ActionsInsertRecipe.java

Holds Editor GUI and confirmation page, as well as controls access to SQL classes for inserting a recipe into the database

Update Inventory Actions | ActionsUpdateInventory.java

Holds Editor GUI and confirmation page, as well as controls access to SQL classes for updating an item in the inventory.

Update Recipe Actions | ActionsUpdateRecipe.java

Holds Editor GUI and confirmation page, as well as controls access to SQL classes for updating recipe information and recipe_ingredient information.

Alert Messages | Alerts.java

Holds Alert pages that will appear if user input is incorrect.

Database connection | DBConnection.java

Connects the Java program to Microsoft SQL server database PantryToPlate.

Delete Inventory SQL | DeleteInventory.java

Calls stored procedures to delete an inventory item in the database. Extends DBConnection.

Delete Recipe SQL | DeleteRecipe.java

Calls stored procedure to delete a recipe in the database. Extends DBConnection.

Insert into Inventory SQL | InsertInventory.java

Calls stored procedure to insert an item into the inventory. Extends DBConnection

Insert new Recipe into CookBook SQL | InsertRecipe.java

Calls stored procedures to insert recipe information and recipe_ingredient information into the database. Extends DBConnection.

Main GUI | PantryToPlate.java

Central control and user interface for the program. Contains GUI panes for the main stage of the program, and Action Events to control other actionable classes.

Styles | Styles.java

Class that holds CSS style strings that can be accessed by all other classes with GUI components, so that the style of the interface is congruent throughout the program.

Update Inventory SQL | UpdateInventory.java

Methods call user defined function to take in new information for a given inventory item, and update the information in the database. Extends DBConnection.

Update Recipe SQL | UpdateRecipe.java

Methods call user defined functions to take in new information for a given recipe and update the recipe information as well as the recipe_ingredient information in the database. Extends DBConnection.

Verification Class | Verify.java

Class that verifies user inputs, and also checks inputs with what is already in the database. Extends DBConnection.

View Available Recipes | ViewAvailableRecipes.java

Class that uses a User Defined function to display recipes that have ingredients available in the pantry. This is used to fill in a page in the GUI. Extends DBConnection.

View Pantry Ingredients | ViewPantry.java

Class that uses a user defined function to display all the inventory items and their information. This is used to fill a page in the GUI. Extends DBConnection.

View Recipe Ingredients for a given Recipe | ViewRecipeIngredients.java

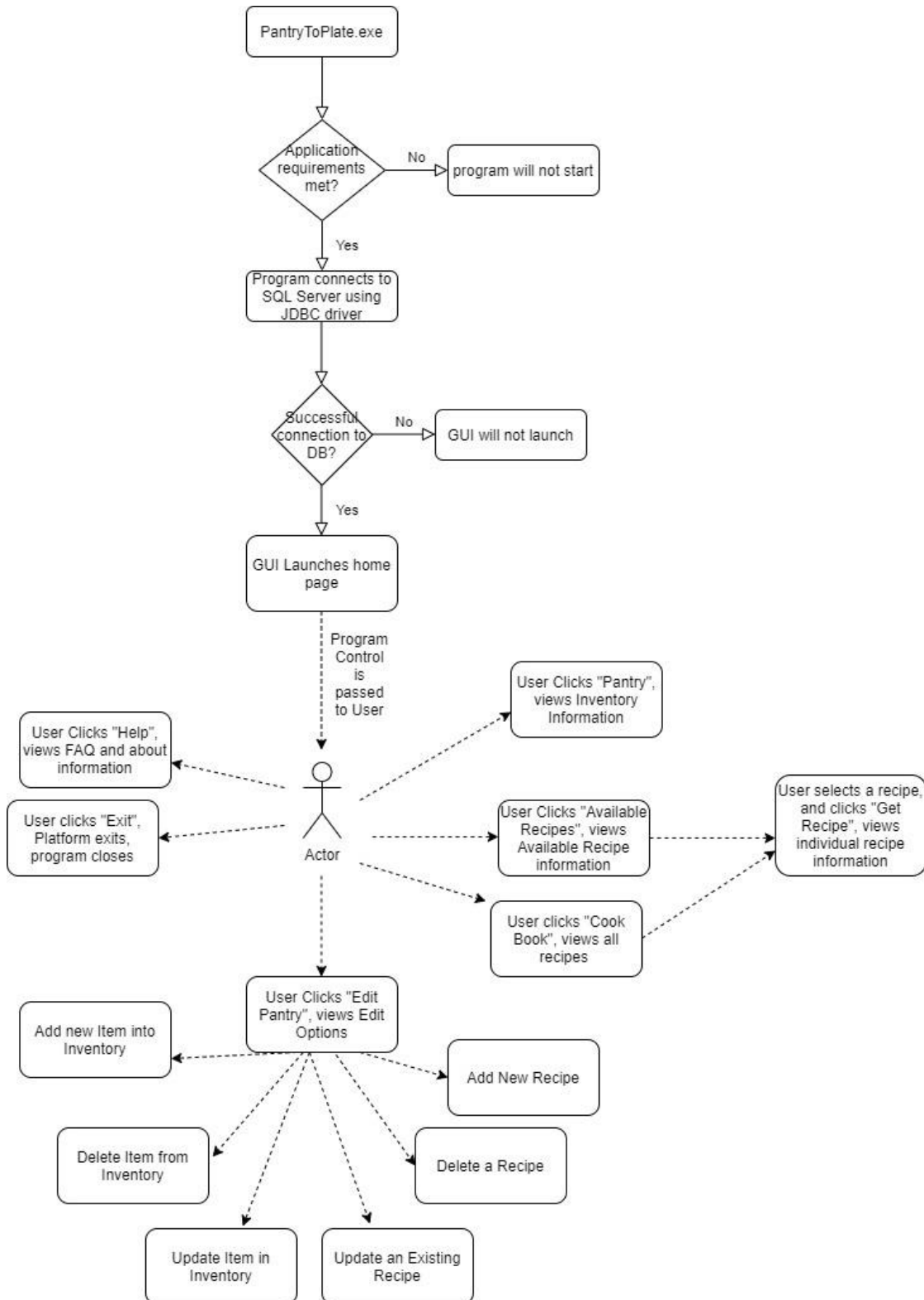
Used in both the AvailableRecipes and All Recipes pages, this class uses a user defined function to show all the ingredients needed for a given recipe. Extends DBConnection.

View Recipes | ViewRecipes.java

Class that uses a user defined function to display all the recipes in the Recipes table of the database. This is used to fill a page in the GUI. Extends DBConnection.

Program Start and End Flow

The User accesses the program by clicking on the PantryToPlate.exe executable file. The program then connects to the SQL database, and will launch the home page GUI. From there, the user can interact with the program as they wish.



Summary

Pantry to Plate is a Java-based application that allows users to build an inventory of their kitchen food items, and pair the available food items with recipes. Users can view the recipe information and be given a link to detailed recipe instructions from the website in which the recipe is from. Users can enter and edit both pantry items and recipe items. These changes are be saved within the back-end database for future use.

APPENDIX B (BUILD AND RELEASE PROCESS)

1. Information about the weaknesses and flaws in the program will be accumulated through continual investigation, and from messages sent to me from users experiencing problems. These flaws will be logged and prioritized based on security and functionality.
2. Changes will be made in a testing environment with a test version of the program. These changes will be tested for functionality in a typical user environment.
3. Each new version will be saved in case an issue with an update occurs and a rollback is necessary.
4. The updated files will be sent to the user, who will be given a prompt to begin the download for the new build. The user can decide when they should update their program at their earliest convenience, but can continue to use the old version.
5. After a release has been sent out, messages will be monitored for catastrophic failures in the update. If so, a rollback will be sent out to all users. Otherwise, the process can begin again by gathering information about weaknesses and flaws in the program.

APPENDIX C (CLIENT INSTALLATION INSTRUCTIONS)

Detail how a client machine or device is prepared to utilize the project.

1. User requirements: Java JRE version 1.7.0 or newer, Microsoft SQL Server 2014. sqljdbc_auth.dll is also required.
2. Clone/Download all files from <https://github.com/kelly-may/PantryToPlate>
3. Create a Shortcut of the PantryToPlate.exe file wherever the user wants quick access to the program.

APPENDIX D (DEVELOPER SETUP INSTRUCTIONS)

Detail how a developer must setup their environment in order to work on the code.

1. User has NetBeans IDE 8.2 installed
2. User has Microsoft SQL Server 2014 Management Studio Express installed
4. Clone/Download all files from <https://github.com/kelly-may/PantryToPlate>
3. Run PantryToPlate_DB.sql in Microsoft SQL Management Studio to create the database locally.
4. Open Java>src files in NetBeans to begin modifications of the Java code