*Draft*

Requirements specification

User types, roles, authentication

User types:

* Administrator, with the purpose of managing the application, which implies:
* 1. Establish user roles, which implies promoting a regular user to the state of admin
* 2. Modify the product category list
* 3. Export statistics based on products
* Basic user, with the purpose of managing its own activity in the application: pantry, grocery lists, meals, menus, recipes

Requirements

1. Administrative module
   1. Scenario 1. Establishing user roles
      1. The administrator can access the list of all users
      2. This list can be ordered by username
      3. This list can be filtered based on specified keywords during search
      4. Administrator can select an user and modify their state in the application (admin/regular user)
   2. Scenario 2. Modify the product category list
      1. The administrator can access the section corresponding to managing the product categories
      2. The list can be order alphabetically
      3. The list can be filtered based on specific keywords during search
      4. Administrator can add a new category to the list
2. Food module
   1. Scenario 1. Managing pantry
      1. User accesses their pantry
      2. They see current items in the pantry
         1. in a table, with columns that specify the name of the ingredient, quantity, in stock, date bought, expiration date, category
         2. The columns can be ordered alphabetically
         3. The columns can be filtered
         4. The category is specified automatically based on the produce
      3. An item can be deleted from the list
      4. An item can be added to the list
      5. An item can be edited (bought date, expiration date, quantity, in stock)
      6. Data conflict solving in case a user wants to add an item already in the pantry, permission denied
   2. Scenario 2. Managing grocery lists
      1. User accesses their grocery lists – a list with titles
         1. They can be ordered alphabetically
         2. user can delete a grocery list
         3. Add a new grocery list
      2. User accesses a grocery list
      3. Sees a list with categories
         1. Under each category the items corresponding to that category that are in the grocery list are included - they are in a table, with name, quantity, and notes
         2. They can add a new item
         3. They can delete an item
   3. Scenario 3. Managing meals
      1. User accesses their meals
      2. They see a list
         1. The meals can be displayed based on day, week or month
         2. The meals are in a list with 4 categories: breakfast, lunch, dinner and snacks
      3. They press the add button
      4. A popup is opened
         1. with a search button for recipes
         2. a list with categories (from breakfast, lunch, dinner, snacks)
         3. an add button
   4. scenario 4. Managing menus
      1. user accesses their menus
      2. they see a list
         1. the menus can be ordered alphabetically
         2. user can delete a menu
         3. add a new menu
      3. user accesses a menu
         1. sees a description
         2. sees a list with each day of the menu
         3. under each day there are the 4 categories of meals: breakfast, lunch, dinner, snacks
      4. user can add new recipe to the menu
      5. a popup is opened
         1. with a search button for recipes
         2. a list with categories
         3. an add button
      6. user can edit a menu item – the category and the day corresponding to it
      7. user can delete a menu item
   5. scenario 5. Managing recipes
      1. user accesses recipes list – the name of the recipe
         1. list can be sorted alphabetically
         2. list can be filtered based on search keywords
      2. they can delete a recipe
      3. they can add a new recipe specifying
         1. preparation instructions
         2. cooking time
         3. ingredients
         4. servings
         5. calories
      4. user accesses a recipe and they see details corresponding to said recipe
      5. user can modify any of the fields

**Requirements Specification**

**User Types and Roles**

* **Administrator**: Manages user roles and product categories.
* **Basic User**: Manages personal activity within the application, including pantry, grocery lists, meals, menus, and recipes.

**General functionalities (applicable to all modules)**

* **List management**:
  + All lists in the application can be sorted alphabetically and filtered based on specified keywords.
  + Users can add, delete, and edit items in any list where these functionalities are relevant.

**Requirements**

**1. Administrative module**

* **Establishing user roles**:
  + Access and view a list of all users.
  + Modify user roles (promote regular user to admin and vice versa).
* **Modify product category list**:
  + Manage (add, edit, delete) product categories.

**2. Food management module**

* **Pantry management**:
  + View and manage pantry items (including adding, editing, deleting).
  + Display items in a table with columns for name, quantity, in stock, date bought, expiration date, and category.
  + Handle data conflicts (e.g., adding an item already in the pantry).
* **Grocery list management**:
  + View and manage multiple grocery lists.
  + For each list, manage items categorized by type (add, delete items).
* **Meal management**:
  + Manage meals displayed by day, week, or month.
  + Add new meals using a searchable recipe database.
  + Organize meals into categories: breakfast, lunch, dinner, snacks.
* **Menu management**:
  + Manage menus, including viewing, adding, and deleting menus.
  + For each menu, manage meal plans for different days and meal types.
  + Edit specific menu items, including meal type and day.
* **Recipe management**:
  + View and manage a list of recipes.
  + Add new recipes with details like preparation instructions, cooking time, ingredients, servings, and calories.
  + Edit and delete recipes.

**Non-functional requirements**

* **Performance**:
  + The application should respond to user inputs within 2 seconds.
* **Usability**:
  + The interface should be user-friendly, adhering to accessibility standards.
* **Security**:
  + Implement secure authentication mechanisms for different user roles.
  + Use data encryption to protect user data and interactions within the application.