Fork & Function

Overview

This project is a **Recipe Manager** application developed in C++ using full Object-Oriented Programming principles. It allows users to:

- Add, edit, and remove recipes
- Categorize recipes by meal type or dietary restriction
- Filter recipes by ingredients
- Search for recipes by title or keyword
- Sort recipes alphabetically, by prep time, or by dietary category

Features

- Add, edit, delete recipes
- Ingredient and dietary filtering
- Categorization by:
    - Meal type (breakfast, lunch, dinner)
    - Dietary restrictions (vegetarian, vegan, pescatarian, omnivore)
- Search (by title or ingredient)
- Sort recipes (by name, prep time, or category)
- Blacklist ingredients to exclude unwanted items

How Project Meets Guidelines

- **OOP Concepts**:
    - Heavy use of inheritance, composition, abstract classes, and

polymorphism

- **Data Structures**:
  - Custom `LinkedList` for recipe management, including support for dynamic insertion, removal, sorting, and searching
- **Algorithms**:
  - Insertion sort to sort recipes alphabetically or by prep time
  - Linear search for finding recipes based on title or ingredients
- **File I/O**:
  - Reading/writing recipes to a `.txt` file to persist user data across sessions
- **Templates/Polymorphism**:
  - `LinkedList<T>` uses templates; recipe management uses polymorphism
- **Focus**:
  - Primarily an **OOP application** focused on user-driven management and filtering of recipes

Stages of Development

Stage 1: Basic Class Structure

- Build basic classes: `Recipe`, `Ingredient`
- Implement `LinkedList<T>` template

Stage 2: Core Features

- Implement recipe creation, editing, and deletion
- Basic search by title

Stage 3: Advanced Filtering

- Add filtering by meal type, dietary restriction, and blacklist ingredients

Stage 4: Sorting and File I/O

- Implement Insertion Sort for recipe lists
- Implement file loading/saving for persistence

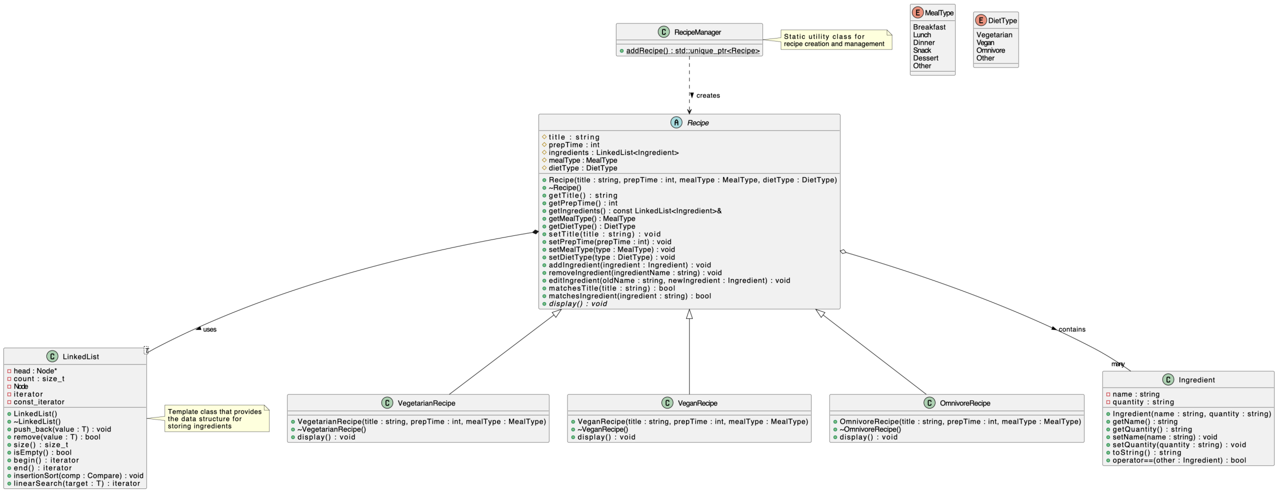
Stage 5: UI Polish & Testing

- Refine console interface
- Final testing and bug fixes

Stage 6: Stretch Features (Optional)

- Add ratings or tags
- Enhance blacklist system with suggestions for substitutions

UML Diagram


RecipeManager
addRecipe() : std::unique_ptr<Recipe>
Static utility class for recipe creation and management
MealType
Breakfast
Lunch
Dinner
Snack
Dessert
Other
DietType
Vegetarian
Vegan
Omnivore
Other
creates
Recipe
title : string
prepTime : int
ingredients : LinkedList<Ingredient>
mealType : MealType
dietType : DietType
Recipe(title : string, prepTime : int, mealType : MealType, dietType : DietType)
~Recipe()
getTitle() : string
getPrepTime() : int
getIngredients() : const LinkedList<Ingredient>&
getMealType() : MealType
getDietType() : DietType
setTitle(title : string) : void
setPrepTime(prepTime : int) : void
setMealType(type : MealType) : void
setDietType(type : DietType) : void
addIngredient(ingredient : Ingredient) : void
removeIngredient(ingredientName : string) : void
editIngredient(oldName : string, newIngredient : Ingredient) : void
matchesTitle(title : string) : bool
matchesIngredient(ingredient : string) : bool
display() : void
uses
T
LinkedList
head : Node*
count : size_t
Node
iterator
const_iterator
LinkedList()
~LinkedList()
push_back(value : T) : void
remove(value : T) : bool
size() : size_t
isEmpty() : bool
begin() : iterator
end() : iterator
insertionSort(comp : Compare) : void
linearSearch(target : T) : iterator
Template class that provides the data structure for storing ingredients
VegetarianRecipe
VegetarianRecipe(title : string, prepTime : int, mealType : MealType)
~VegetarianRecipe()
display() : void
VeganRecipe
VeganRecipe(title : string, prepTime : int, mealType : MealType)
~VeganRecipe()
display() : void
OmnivoreRecipe
OmnivoreRecipe(title : string, prepTime : int, mealType : MealType)
~OmnivoreRecipe()
display() : void
contains
many
Ingredient
name : string
quantity : string
Ingredient(name : string, quantity : string)
getName() : string
getQuantity() : string
setName(name : string) : void
setQuantity(quantity : string) : void
toString() : string
operator==(other : Ingredient) : bool