

```

from recipe import Recipe
from ingredient import Ingredient
from ingredient_manager import IngredientManager
from ingredient_validator import IngredientValidator
from recipe_database import RecipeDatabase
from web_recipes import scrape_recipe_info

def display_menu():
    """
    Display the main menu of the Recipe Finder application.
    """
    print("Welcome to Recipe Finder!")
    print("1. Search for Recipes")
    print("2. Add a New Recipe")
    print("3. View All Recipes")
    print("4. Add Your Own Recipe")
    print("5. Exit")

def search_recipes(recipe_db, ingredient_manager):
    """
    Search for recipes in the Recipe Database based on user input.

    Args:
        recipe_db (RecipeDatabase): The recipe database instance.
        ingredient_manager (IngredientManager): The ingredient manager instance.
    """
    print("Search Recipes")
    search_term = input("Enter a recipe name to search for: ")

    # Get the list of matching recipes from the recipe_db
    matching_recipes = recipe_db.get_recipes_by_name(search_term)

    if not matching_recipes:
        print("No recipes found.")
        return

    print("Matching Recipes:")
    for recipe in matching_recipes:
        print_recipe(recipe)

def add_new_recipe(recipe_db, recipe_info, ingredient_manager,
ingredient_validator):
    """
    Add a new recipe to the Recipe Database based on user input.

    Args:
        recipe_db (RecipeDatabase): The recipe database instance.
        recipe_info (dict): Dictionary containing recipe information and URL.
        ingredient_manager (IngredientManager): The ingredient manager instance.
        ingredient_validator (IngredientValidator): The ingredient validator
instance.
    """
    new_recipe = recipe_info["recipe"]
    new_recipe_url = recipe_info["url"]

    try:
        # Pass the URL when adding the recipe
        recipe_db.add_recipe(new_recipe, new_recipe_url)

```

```

        print(f"Recipe '{new_recipe.name}' from {new_recipe_url} added
successfully!")

        # Code to add ingredients and update ingredient_manager goes here
        # ...

        # Code to update recipe views goes here
        # ...
    except Exception as e:
        print("Error:", e)
        print("Failed to add the recipe. Please make sure you've entered a valid
recipe page URL.")

def view_all_recipes(recipe_db):
    """
    View all recipes in the Recipe Database.

    Args:
        recipe_db (RecipeDatabase): The recipe database instance.
    """
    with recipe_db.conn:
        recipes = recipe_db.view_all_recipes()

    print("All Recipes:")
    for recipe in recipes:
        print_recipe(recipe)
        print("=" * 40)

def create_recipe(ingredient_validator):
    """
    Create a new recipe by taking user input for recipe name, instructions, and
    ingredients.

    Parameters:
        - ingredient_validator (IngredientValidator): An instance of
IngredientValidator to validate ingredient inputs.

    Returns:
        - Recipe: The created Recipe instance.
    """
    print("Creating a New Recipe")
    recipe_name = input("Enter the recipe name: ")
    instructions = input("Enter the recipe instructions: ")

    ingredients = []
    while True:
        ingredient_name = input("Enter an ingredient name (or 'done' to finish): ")
        if ingredient_name.lower() == "done":
            break

        unit = input("Enter the unit for the ingredient (e.g., cup, teaspoon,
etc.): ")
        amount = input("Enter the amount for the ingredient: ")

        if ingredient_validator.validate(ingredient_name, unit, amount):
            ingredients.append(Ingredient(ingredient_name, unit, float(amount)))
            print("Ingredient added successfully!")
        else:
            print("Invalid ingredient data. Please try again.")

```

```

    return Recipe(recipe_name, instructions, ingredients, "User Created")

def print_recipe(recipe):
    """
    Print detailed information about a recipe.

    Args:
        recipe (Recipe): The recipe instance.
    """
    print(f"Recipe Name: {recipe.name}")
    print(f"Instructions: {recipe.instructions}")
    print(f"URL: {recipe.url}")
    for ingredient in recipe.ingredients:
        print(f"Ingredient: {ingredient.name}")
        print(f"Unit: {ingredient.unit}")
        print(f"Amount: {ingredient.amount}")

def main():
    recipe_db = RecipeDatabase()
    ingredient_manager = IngredientManager(recipe_db)
    ingredient_validator = IngredientValidator()

    while True:
        display_menu()
        choice = input("Enter your choice: ")

        if choice == "1":
            search_recipes(recipe_db, ingredient_manager)
        elif choice == "2":
            user_input = input("Enter the recipe page URL: ")
            try:
                recipe_info = scrape_recipe_info(user_input)
                add_new_recipe(recipe_db, recipe_info, ingredient_manager,
ingredient_validator)
            except Exception as e:
                print("Error:", e)
                print("Failed to fetch recipe information from the provided URL.")
        elif choice == "3":
            view_all_recipes(recipe_db)
        elif choice == "4":
            new_recipe = create_recipe(ingredient_validator)
            add_new_recipe(recipe_db, {"recipe": new_recipe, "url": "User
Created"}, ingredient_manager, ingredient_validator)
        elif choice == "5":
            print("Thank you for using Recipe Finder!")
            break
        else:
            print("Invalid choice. Please select a valid option.")

if __name__ == "__main__":
    main()

```