```
#Recipe management system
class Recipes:
    def __init__(self,name,ingredients,category,cooking_time,steps):
        self.name=name
        self.ingredients=ingredients
        self.category=category
        self.cooking_time=cooking_time
        self.steps=steps
        self.view_count=0
        self.cook_count=0
class Recipemanagement:
    def __init__(self):
        self.recipes = [
            Recipes("Pancakes", ["flour", "milk", "egg", "sugar"], "Breakfast", 15, ["Mix ingredients", "Cook on pan"]),
            Recipes("Omelette", ["egg", "onion", "salt"], "Breakfast", 10, ["Beat eggs", "Cook on pan"]),
            Recipes("Chicken Curry", ["chicken", "spices", "yogurt"], "Lunch", 40, ["Marinate chicken", "Cook with spices"]),
            Recipes("Pasta", ["pasta", "cheese", "tomato sauce"], "Lunch", 25, ["Boil pasta", "Mix with sauce"]),
            Recipes("Steak", ["beef", "salt", "pepper"], "Dinner", 30, ["Season beef", "Cook on grill"]),
            Recipes("Pizza", ["dough", "cheese", "tomato sauce"], "Dinner", 20, ["Prepare dough", "Bake with toppings"]),
            Recipes("Brownies", ["flour", "cocoa", "sugar", "butter"], "Dessert", 30, ["Mix ingredients", "Bake"]),
            Recipes("Ice Cream", ["milk", "sugar", "cream"], "Dessert", 120, ["Mix ingredients", "Freeze"])
        ]

    def add_recipe(self):
        print()
        name = input("Enter Recipe Name you want to add: ")
        ingredients = []
        while True:
            ingredient = input("Enter Ingredients (type 'done' to finish): ")
            if ingredient.lower() == "done":
                break
            ingredients.append(ingredient)
        category = input("Enter Recipe Category (Breakfast, Lunch, Dinner, Dessert): ")
        cooking_time = int(input("Enter Cooking Time (in minutes): "))

        steps = []
        while True:
            step = input("Enter Steps (type 'done' to finish: ")
            if step.lower() == "done":
                break
            steps.append(step)

        new_recipe = Recipes(name, ingredients, category, cooking_time, steps)
        self.recipes.append(new_recipe)
        print()
        print(f"Recipe '{name}' added successfully!")

    def edit_recipe(self):
        print()
        name = input("Enter the name of the recipe you want to edit: ")
        for recipe in self.recipes:
            if recipe.name.lower() == name.lower():
                print()
                print(f"Editing {recipe.name}")
                print("You can type 'done' to finish or leave blank if you do not want to change a specific attribute of the recipe.")
                print()
                new_name = input("Enter New Name : ")
                if new_name:
                    recipe.name = new_name

                print("Enter New Ingredients :")
                ingredients = []
                while True:
                    ingredient = input("Ingredient: ")
                    if ingredient.lower() == "done":
                        break
                    ingredients.append(ingredient)
                if ingredients:
                    recipe.ingredients = ingredients

                category = input("Enter New Category : ")
                if category:
                    recipe.category = category

                cooking_time = input("Enter New Cooking Time (in minutes): ")
                if cooking_time:
                    recipe.cooking_time = int(cooking_time)

                print("Enter New Steps:")
                steps = []
                while True:
                    step = input("Step: ")
                    if step.lower() == "done":
                        break
                    steps.append(step)
                if steps:
                    recipe.steps = steps

                print(f"Recipe '{recipe.name}' updated successfully!")
                return
        print("Recipe not found.")

    def delete_recipe(self):
        print()
        name = input("Enter the name of the recipe you want to delete: ")
        for recipe in self.recipes:
            if recipe.name.lower() == name.lower():
                self.recipes.remove(recipe)
                print(f"Recipe '{name}' deleted successfully!")
                return
        print("Recipe not found.")

    def search_recipes(self):
        print("\n****** SEARCH RECIPES *******")
        print("1. By Ingredients")
        print("2. By Category")
        print("3. By Cooking Time")
        choice = input("Enter your choice: ")
        if choice == "1":
            print()
            ingredient = input("Enter an ingredient: ").lower()
            found = [
                recipe
                for recipe in self.recipes
                if any(ingredient in i.lower() for i in recipe.ingredients)
            ]
            if found:
                for recipe in found:
                    print(f"{recipe.name} - {recipe.category}")
            else:
                print("No recipes found with that ingredient.")
        elif choice == "2":
            print()
            category = input("Enter category (Breakfast, Lunch, Dinner, Dessert): ").lower()
            found = [recipe for recipe in self.recipes if recipe.category.lower() == category]
            if found:
                for recipe in found:
                    print(f"{recipe.name} - {recipe.category}")
            else:
                print("No recipes found in that category.")
        elif choice == "3":
            print()
            time = int(input("Enter cooking time (in minutes): "))
            found = [recipe for recipe in self.recipes if recipe.cooking_time <= time]
            if found:
                for recipe in found:
                    print(f"{recipe.name} - {recipe.category} ({recipe.cooking_time} minutes)")
            else:
                print("No recipes found within that cooking time.")
        else:
            print("Invalid choice.")

    def view_recipe(self, recipe):
        current_step = 0
        while True:
            print(f"\nStep {current_step + 1}/{len(recipe.steps)}: {recipe.steps[current_step]}")
            print("\nOptions:")
            if current_step > 0:
                print("1. Previous Step")
            if current_step < len(recipe.steps) - 1:
                print("2. Next Step")
            print("3. Exit to Main Menu")

            choice = input("Enter your choice: ")
            choice = input("Enter your choice: ")
            if choice == "1" and current_step > 0:
                current_step -= 1
            elif choice == "2" and current_step < len(recipe.steps) - 1:
                current_step += 1
            elif choice == "3":
                print("Returning to main menu...")
                break
            else:
                print("Invalid choice. Please try again.")

    def select_recipe(self):
        print("\n****** Select a Recipe to View Steps *******")
        print("Available Recipes:")
        for i in range(len(self.recipes)):
            print(f"{i + 1}. {self.recipes[i].name} - {self.recipes[i].category}")
        try:
            recipe_choice = int(input("Enter the number of the recipe to view steps: ")) - 1
            if 0 <= recipe_choice < len(self.recipes):
                self.view_recipe(self.recipes[recipe_choice])
            else:
                print("Invalid recipe choice.")
        except ValueError:
            print("Please enter a valid number.")


    def save_recipe(self):
        print("\n****** Save Your Favorite Recipe *******")
        print("Available Recipes:")
        for i in range(len(self.recipes)):
            print(f"{i + 1}. {self.recipes[i].name} - {self.recipes[i].category}")

        try:
            recipe_choice = int(input("Enter the number of the recipe to save as favorite: ")) - 1
            if 0 <= recipe_choice < len(self.recipes):
                recipe = self.recipes[recipe_choice]
                file_name = "Myfavorite_recipes.txt"
                with open(file_name, "w") as file:
                    file.write(f"Recipe: {recipe.name}\n")
                    file.write(f"Category: {recipe.category}\n")
                    file.write(f"Cooking Time: {recipe.cooking_time} minutes\n")
                    file.write("Ingredients:\n")
                    for ingredient in recipe.ingredients:
                        file.write(f"  - {ingredient}\n")
                    file.write("Steps:\n")
                    for step in recipe.steps:
                        file.write(f"  - {step}\n")
                    file.write("\n")
                print(f"Recipe '{recipe.name}' has been saved to {file_name}!")
            else:
                print("Invalid recipe choice.")
        except ValueError:
            print("Please enter a valid number.")



    def recipe_stats(self):
        print("\n*** Recipe Statistics ****")
        total_recipes = len(self.recipes)
        print(f"Total Number of Recipes: {total_recipes}")

        if self.recipes:
            def get_most_viewed(recipe_list):
                most_viewed_recipe = recipe_list[0]
                for recipe in recipe_list:
                    if recipe.view_count > most_viewed_recipe.view_count:
                        most_viewed_recipe = recipe
                return most_viewed_recipe

            def get_most_cooked(recipe_list):
                most_cooked_recipe = recipe_list[0]
                for recipe in recipe_list:
                    if recipe.cook_count > most_cooked_recipe.cook_count:
                        most_cooked_recipe = recipe
                return most_cooked_recipe

            most_viewed = get_most_viewed(self.recipes)
            most_cooked = get_most_cooked(self.recipes)
            print(f"Most Viewed Recipe: {most_viewed.name} (Views: {most_viewed.view_count})")
            print(f"Most Cooked Recipe: {most_cooked.name} (Cooked Count: {most_cooked.cook_count})")
        else:
            print("No recipes available to calculate most viewed or cooked.")

        category_counts = {}
        for recipe in self.recipes:
            category_counts[recipe.category] = category_counts.get(recipe.category, 0) + 1

        if category_counts:
            most_common_category = max(category_counts, key=category_counts.get)
            print(f"Category with the Most Recipes: {most_common_category} ({category_counts[most_common_category]} recipes)")
        else:
            print("No categories available.")

class User:
    def __init__(self):
        self.username="admin"
        self.password="12345"

    def login(self):
        print("***** LOGIN TO ACCESS THE RECIPE MANAGEMENT SYSTEM *****")
        print()
        print("Please enter your credentials to continue")
        print()
        enter_username=input("Enter Username:")
        enter_password=input("Enter Password:")
        if enter_username==self.username and enter_password==self.password:
            recipe_manager=Recipemanagement()
            while True:
                print()
                print("======== MAIN  MENU ===========")
                print("1.Add New Recipe")
                print("2.Edit Recipe")
                print("3.Delete Recipe")
                print("4.Search Recipe")
                print("5.View Cooking Steps")
                print("6.Save Favourite Recipe")
                print("7.Check Recipe Statistics")
                print("8.Exit....")
                choice= input("Enter Your Choice:")
                if choice=="1":
                    recipe_manager.add_recipe()
                elif choice=="2":
                    recipe_manager.edit_recipe()
                elif choice=="3":
                    recipe_manager.delete_recipe()
                elif choice=="4":
                    recipe_manager.search_recipes()
                elif choice=="5":
                    recipe_manager.select_recipe()
                elif choice=="6":
                    recipe_manager.save_recipe()
                elif choice=="7":
                    recipe_manager.recipe_stats()
                elif choice=="8":
                    print("loging out...")
                    break
        else:
            print("invalid username or password")

user=User()
user.login()
```

```
***** LOGIN TO ACCESS THE RECIPE MANAGEMENT SYSTEM *****

Please enter your credentials to continue

Enter Username: admin
Enter Password: 12345

======== MAIN  MENU ===========
1.Add New Recipe
2.Edit Recipe
3.Delete Recipe
4.Search Recipe
5.View Cooking Steps
```

```
6.Save Favourite Recipe
7.Check Recipe Statistics
8.Exit....
Enter Your Choice: 4

****** SEARCH RECIPES *******
1. By Ingredients
2. By Category
3. By Cooking Time
```

Start coding or generate with AI.