

Worksheet: Understanding MVC and Layered Architecture

Where Do I Make This Change?

Purpose:

You will be given several development tasks that might appear in a real software engineering job. Your goal is to decide which parts of a Spring Boot MVC application would need to be updated to complete each task. This activity will help you practice identifying responsibilities across the View, Controller, Service, Repository, Entity, and Database layers.

Instructions:

For each task below, read the description carefully and determine:

1. Which layer or layers of the application would be involved.
2. Which specific parts of the codebase you would need to modify.

Do not write code. Focus only on identifying the correct architectural location(s) where changes should occur.

Part 1: Developer Tasks

For each task, list the layer(s) that would be affected and briefly explain what changes would be required there.

Task 1: Add a “Difficulty Level” to Each Recipe

A new feature requires displaying a difficulty level (Easy, Medium, Hard) for each recipe throughout the application.

Layers Involved:

Your Notes:

Task 2: Prevent Saving Recipes With No Ingredients

A bug report indicates that users are able to save recipes that do not include any ingredients. This should not be allowed.

Layers Involved:

Your Notes:

Task 3: Add a Page to Show All Recipes Under 20 Minutes

The application should include a new page listing all recipes that can be made in under 20 minutes.

Layers Involved:

Your Notes:

Task 4: Display Recipes Sorted by Rating

A stakeholder has requested that recipes appear sorted by rating before being shown to users.

Layers Involved:

Your Notes:

Task 5: Add an Optional “Notes” Field to Recipes

Users would like to store optional notes for each recipe. This field should be saved and displayed when viewing the recipe.

Layers Involved:

Your Notes:

Task 6: Add Search by Ingredient Keyword

Users want to type in a keyword such as “chicken” and see recipes that match that ingredient.

Layers Involved:

Your Notes:

Task 7: Show an Error When Deleting a Recipe That Is Linked to a Meal Plan

If a recipe is associated with a meal plan, the user should not be able to delete it. The application should display an appropriate error message.

Layers Involved:

Your Notes:

Task 8: Fix Incorrect Cooking Time Display

A bug report says that recipes are showing an incorrect cooking time. You must investigate and update the correct location(s) in the application.

Layers Involved:

Your Notes:

Task 9: Add Pagination to the Recipe List

The main list of recipes should show a limited number per page with buttons for navigating between pages.

Layers Involved:

Your Notes:

Task 10: Create a “Recently Added” Section

The homepage should display a list of newly added recipes.

Layers Involved:

Your Notes:

Part 2: Reflection Questions

Answer each question in 2–4 sentences.

1. Why is it important to separate responsibilities among multiple layers in an application?

Your Response:

2. Describe a situation where placing logic in the wrong layer could cause problems.

Your Response:

3. Give one example of work that belongs in each of the following areas:

- View
- Controller
- Service
- Repository
- Entity
- Database

Your Response:

Part 3: Summary

In your own words, summarize how MVC and layered architecture help make large software applications easier to build, test, and maintain.

Your Summary: