Recipe management system
MyCookbooks
- a Blazor WebAssembly app -

Dinco Andreea-Elena

Contents

I. Overview

MyCookbooks is a recipe management system that allows users to create, modify, and delete cookbooks and recipes. Users can add images, nutrition facts, instructions, ingredients, and cooking time details. Recipes can be viewed for each cookbook or all cookbooks.

II. Getting Started

First, clone the repository for this app or download the source code zip file.

To run this app, SQL Server and Entity Framework are needed.

To install the tools needed for Entity Framework, type in the following command line in the Package Manager Console : **dotnet tool install --global dotnet-ef**.

After configuring SQL Server on your machine, update the connection string in the appsettings.json file of the Server project to match your server name and user credentials.

To create the database, navigate to the Server directory using this command in the Package Manager Console: **cd Server**. Then update the database using this command: **dotnet ef database update**.

Run the app using this command: **dotnet watch run** [1].

III. Project Structure

This app is an ASP.NET Core Hosted Blazor WebAssembly project and contains the following ASP.NET Core projects:

- **Client**: The Blazor WebAssembly app.
- **Server**: The app that serves the Blazor WebAssembly app and all the data needed in the app.
- **Shared**: The project that contains the common classes [2].

The Client project contains the following folders and files:

- **Properties**: Holds the default development environment configuration in the launchSettings.json file.
- **wwwroot**: the Web Root folder containing the root page of the app, **index.html**, and the **css** folder.
- **Pages**: Contains the Blazor app's routable Razor components:
    - **Index.razor**: implements the Home page.
    - **RecipeForm.razor:** implements the edit form for adding or editing recipes.
    - **RecipeView.razor**: implements the page which displays details about recipes.
- **Services**: Contains a folder for each class having methods that request data from the API.
- **Shared**: Contains a razor component together with the stylesheet for the app's layout component.

- **_Imports.razor**: Includes @using directives for the services and the other ASP.NET Core projects, BlazorBootstrap and other namespaces.
- **App.razor**: The root component of the app that sets up client-side routing using the Router component. The Router component intercepts browser navigation and renders the page that matches the requested address.
- **Program.cs**: The app's entry point that sets up the WebAssembly host. Services are added and configured here. Services for BlazorBootstrap and services for requesting data from the API have been added here [2].

The Server project contains the following folders and files:

- **Controllers**: Includes controllers for receiving and returning data from/to the database to the pages in the Client project.
- **Data**: Contains the Data class which inherits DbContext and interacts with data from the database as objects.
- **Migrations**: Contains the files that encapsulate the changes made to the database schema.
- **Pages**: Contains a view that is rendered when an error occurs.
- **appsettings.json**: Contains the connection string for connecting to the database.
- **Program**: The entry point to the Blazor WebAssembly app. When the app executes, it creates and runs a web host instance using defaults specific to web apps. The web host manages the Blazor WebAssembly app's lifecycle and sets up host-level services [3].

The Shared project contains the classes that represent the recipes and cookbooks data for the Client and Server project.
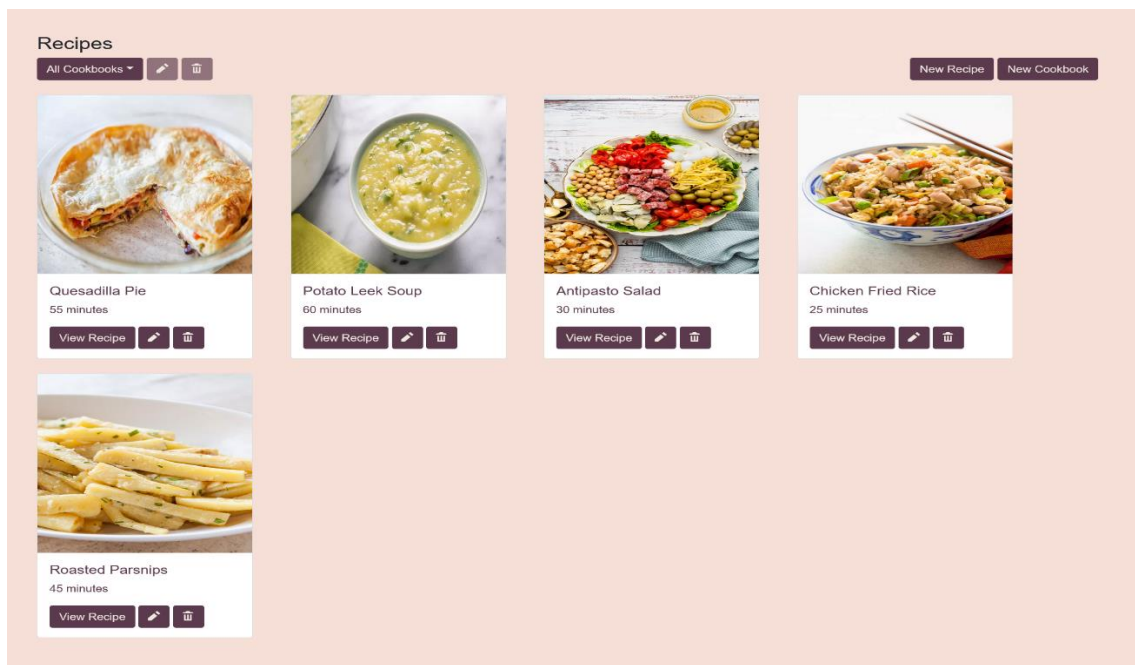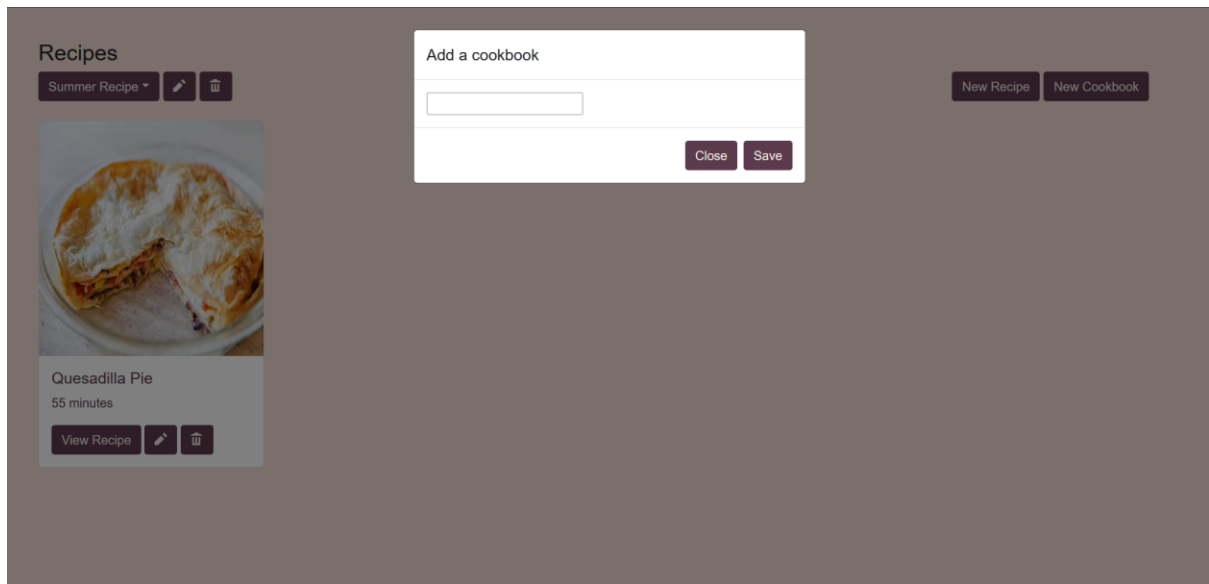

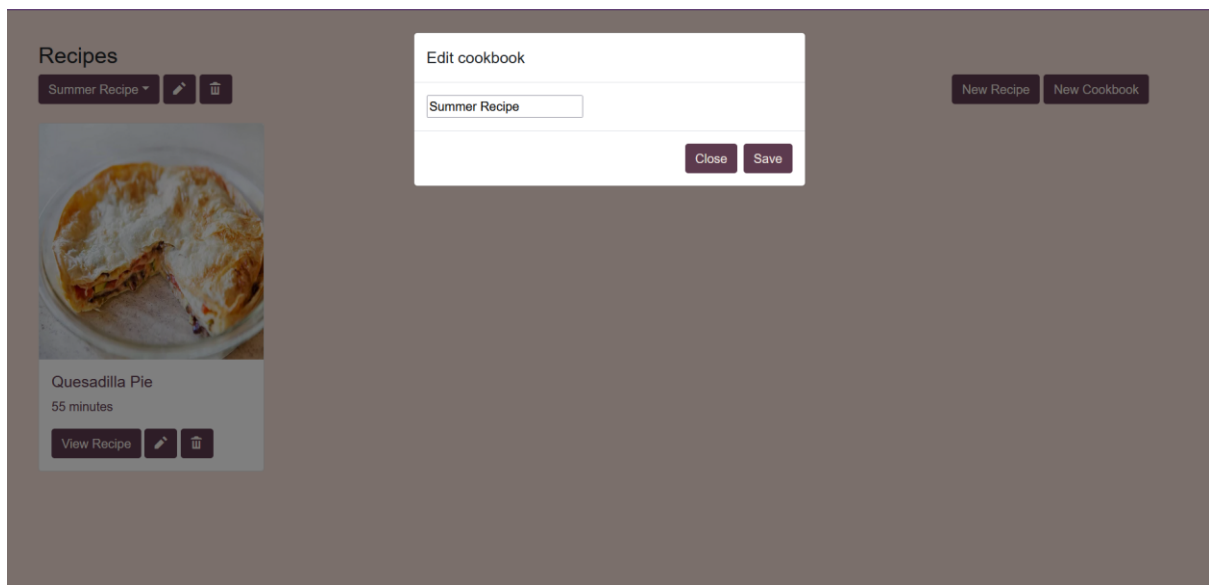Figure III.1. Home page UI

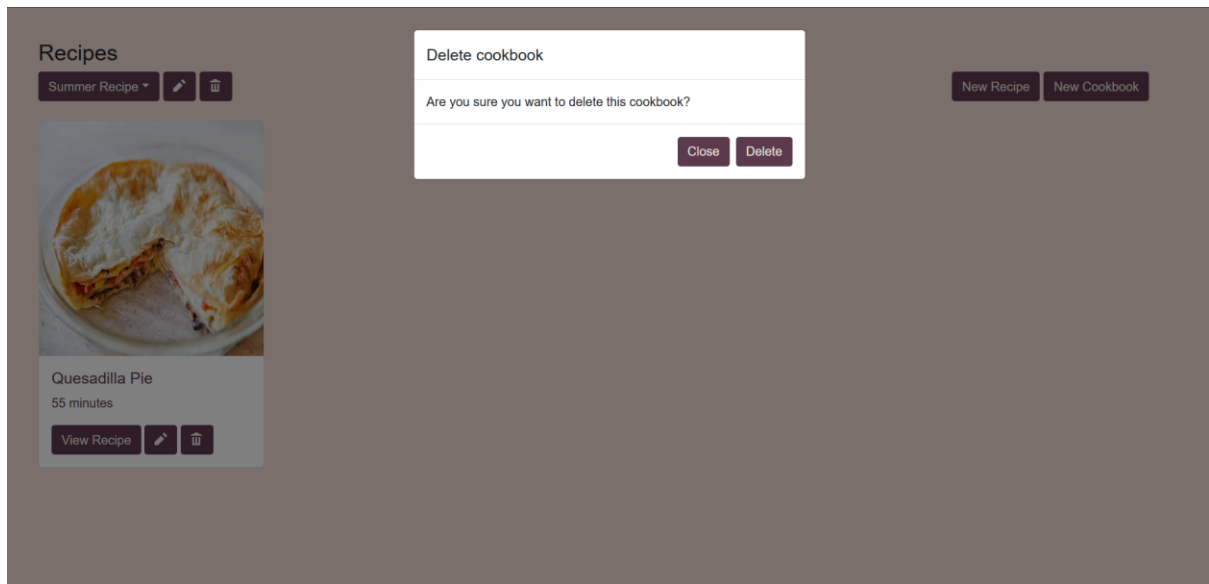Figure III.2. Add a cookbook

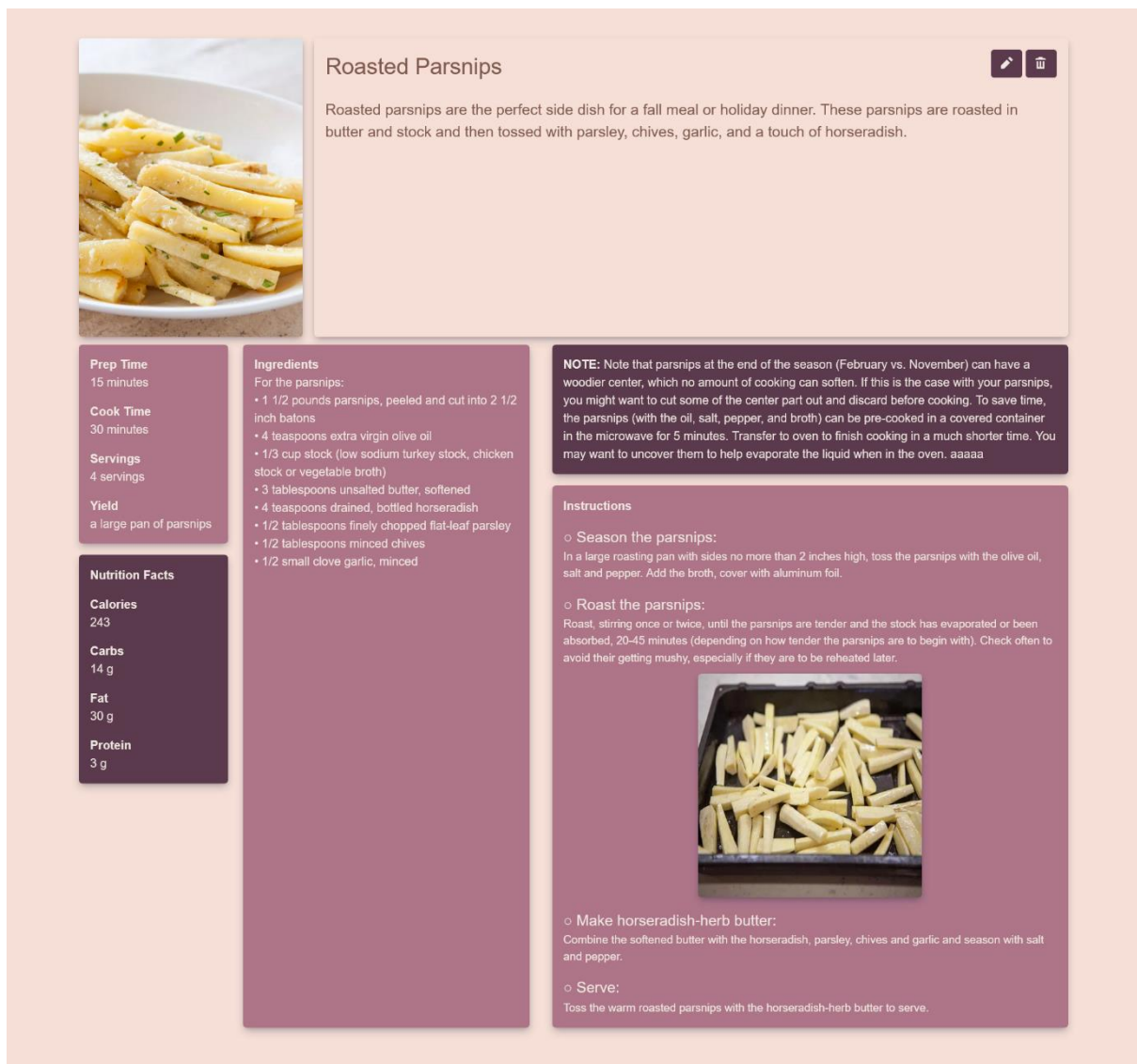Figure III.3. Edit a cookbook

Figure III.4. Delete cookbook

Figure III.5. View recipe page UI

Edit Recipe

Title

Roasted Parsnips

Description

Roasted parsnips are the perfect side dish for a fall meal or holiday dinner. These parsnips are roasted in butter and stock and then tossed with parsley, chives, garlic, and a touch of horseradish.

Default Cookbook ▾

Browse… No file selected.

Prep Time
15

Cook Time
30

Additional Time
0

Servings
4

Yield
a large pan of pa

Nutrition Facts
Calories
243

Carbs
14

Fat
30

Protein
3

Ingredients

For the parsnips:

parsnips, peeled and

1 1/2 pounds

extra virgin olive oil

4 teaspoons

stock (low sodium tu

1/3 cup

unsalted butter, softe

3 tablespoons

drained, bottled hors

4 teaspoons

finely chopped flat-le

1/2 tablespoons

minced chives

1/2 tablespoons

small clove garlic, mi

1/2

Add Ingredient

Add Ingredient Note

Note

Note that parsnips at the end of the season (February vs. November) can have a woodier center, which no amount of cooking can soften. If this is the case with your parsnips, you might want to cut some of the center part out and discard before cooking.
To save time, the parsnips (with the oil, salt, pepper, and broth) can be pre-

Instructions

Season the parsnips:

In a large roasting pan with sides no more than 2 inches high, toss the parsnips with the olive oil, salt and pepper.

Add the broth, cover with aluminum foil.

Browse… No file selected.

Roast the parsnips:

Roast, stirring once or twice, until the parsnips are tender and the stock has evaporated or been absorbed, 20-45 minutes (depending on how tender the parsnips are to begin with). Check often to avoid their getting mushy, especially if they are to be reheated later.

Browse… No file selected.

Make horseradish-herb but

Combine the softened butter with the horseradish, parsley, chives and garlic and season with salt and pepper.

Browse… No file selected.

Serve:

Toss the warm roasted parsnips with the horseradish-herb butter to serve.

Browse… No file selected.

Add instructions

Figure III.6. Edit recipe page UI

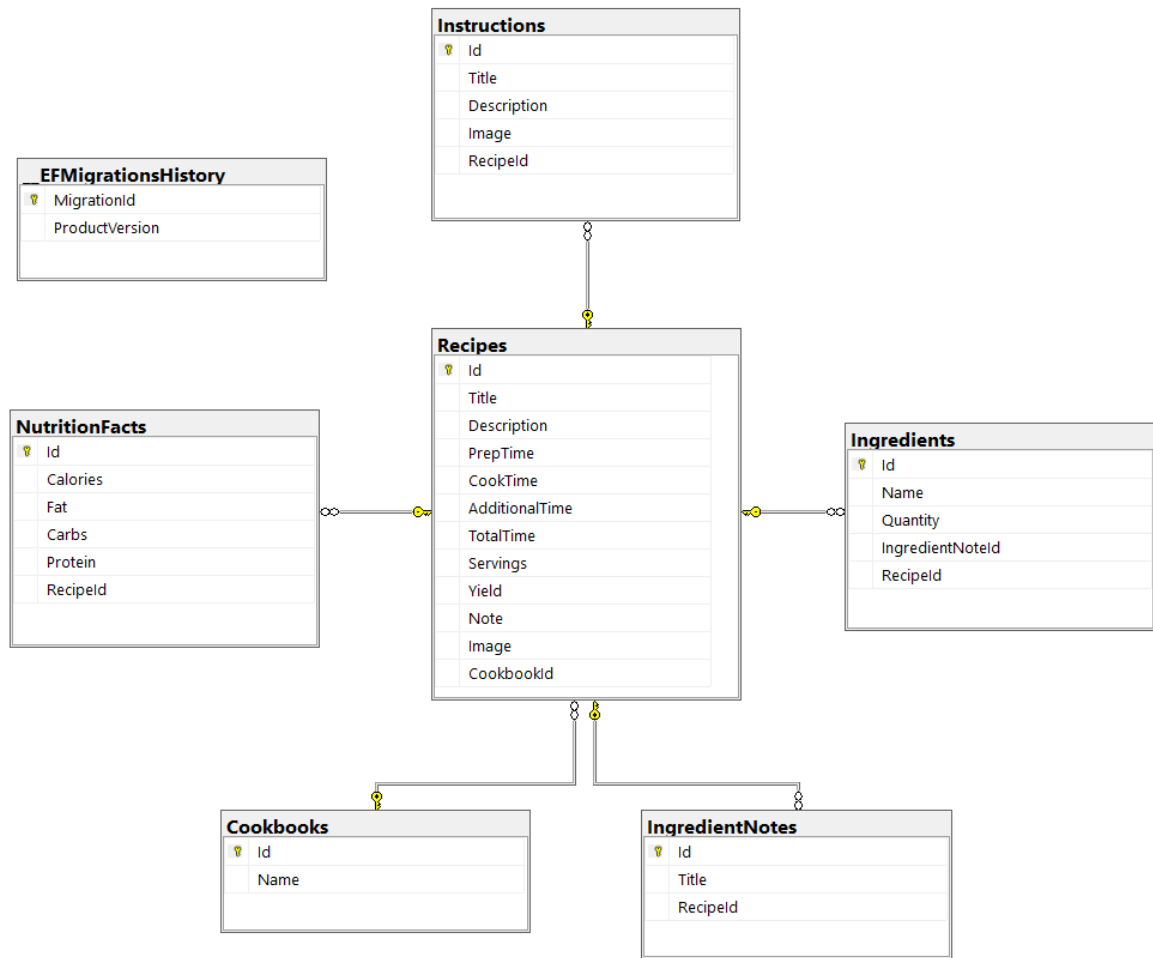Figure III.7. Add recipe page UI
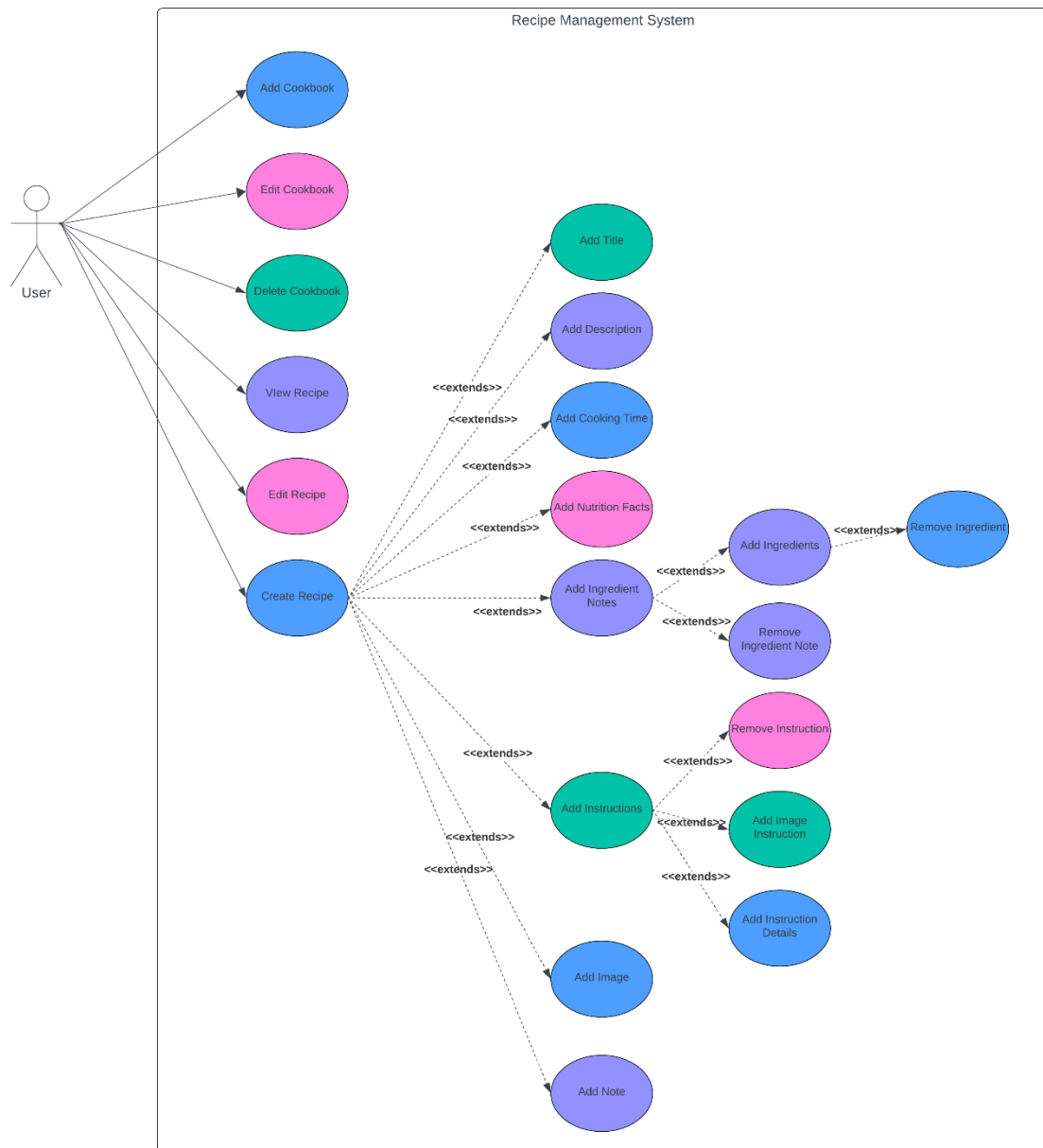
Figure III.8. Entity-Relationship Diagram

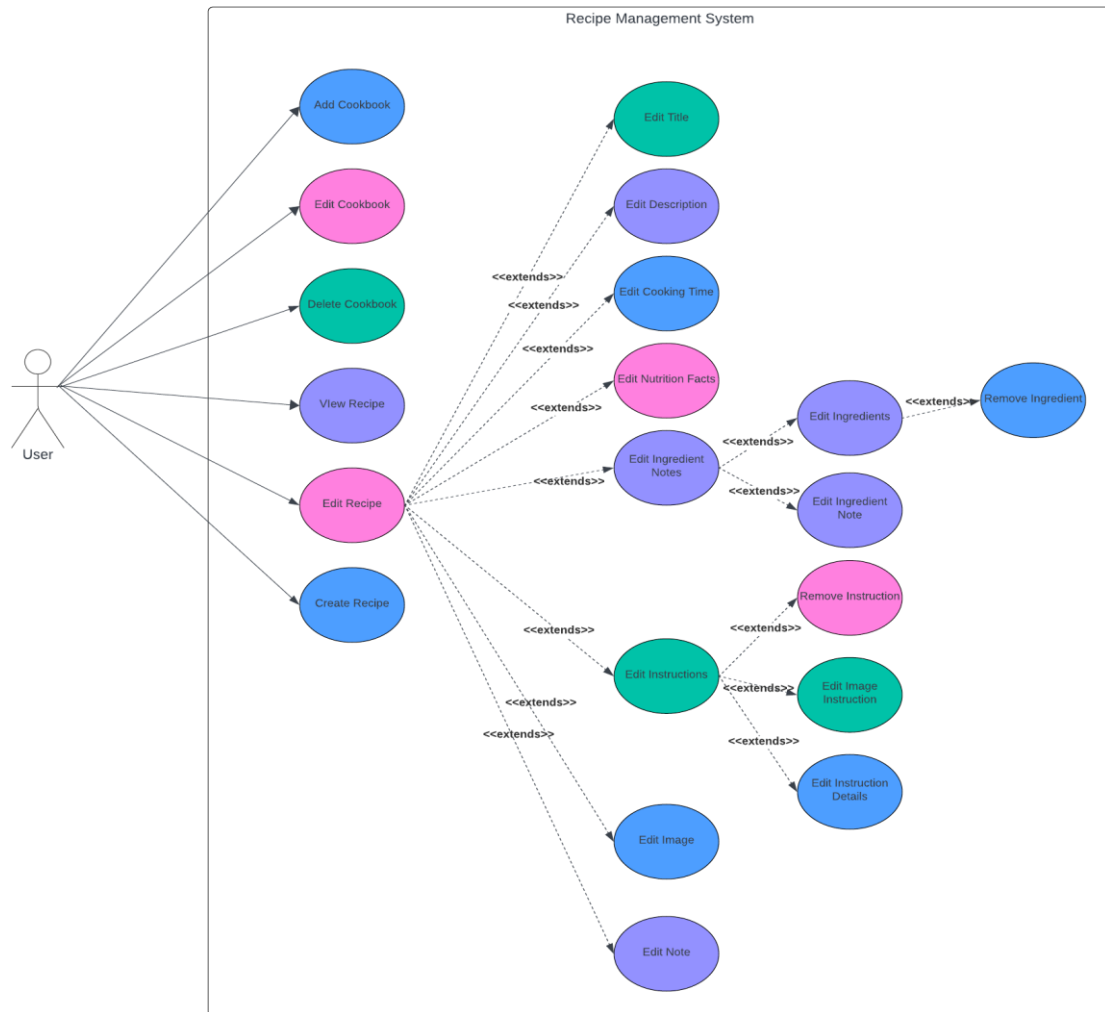Figure III.9. Use Case Diagram (Create Recipe)

Figure III.10. Use Case Diagram (Edit Recipe)

IV.   User Guide

When the home page is accessed, all the recipes existent in the database will be shown as cards that display an image, the recipe name, the total cooking time for the recipe and three buttons that let the user view details about the recipe, edit the recipe or delete the recipe. Above the recipes is a dropdown from where the user can change which cookbook they want to see and display all the recipes in that cookbook. Next to the dropdown are two buttons that lets the user edit or delete the currently selected cookbook. The *Default Cookbook* and *All Cookbooks* can't be deleted or edited.
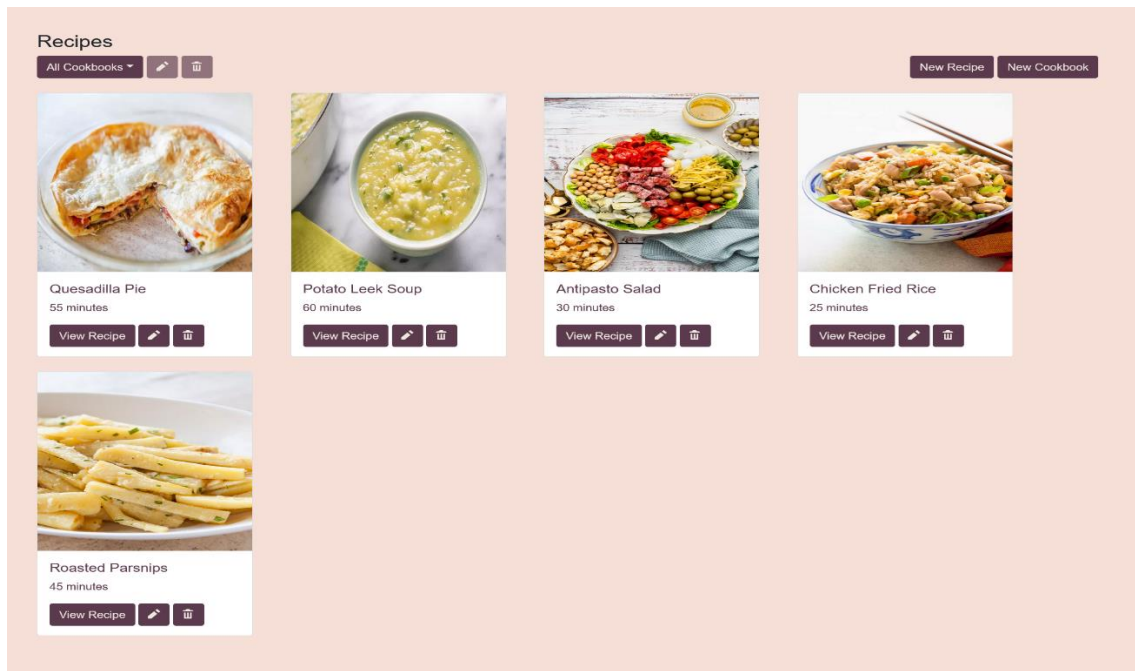
Figure IV.1. View cookbooks and recipes

To view a recipe and all the details about it click on the *View Recipe* button which is located on the card which displays the recipe image, the name of the recipe and the total time it takes to make it.
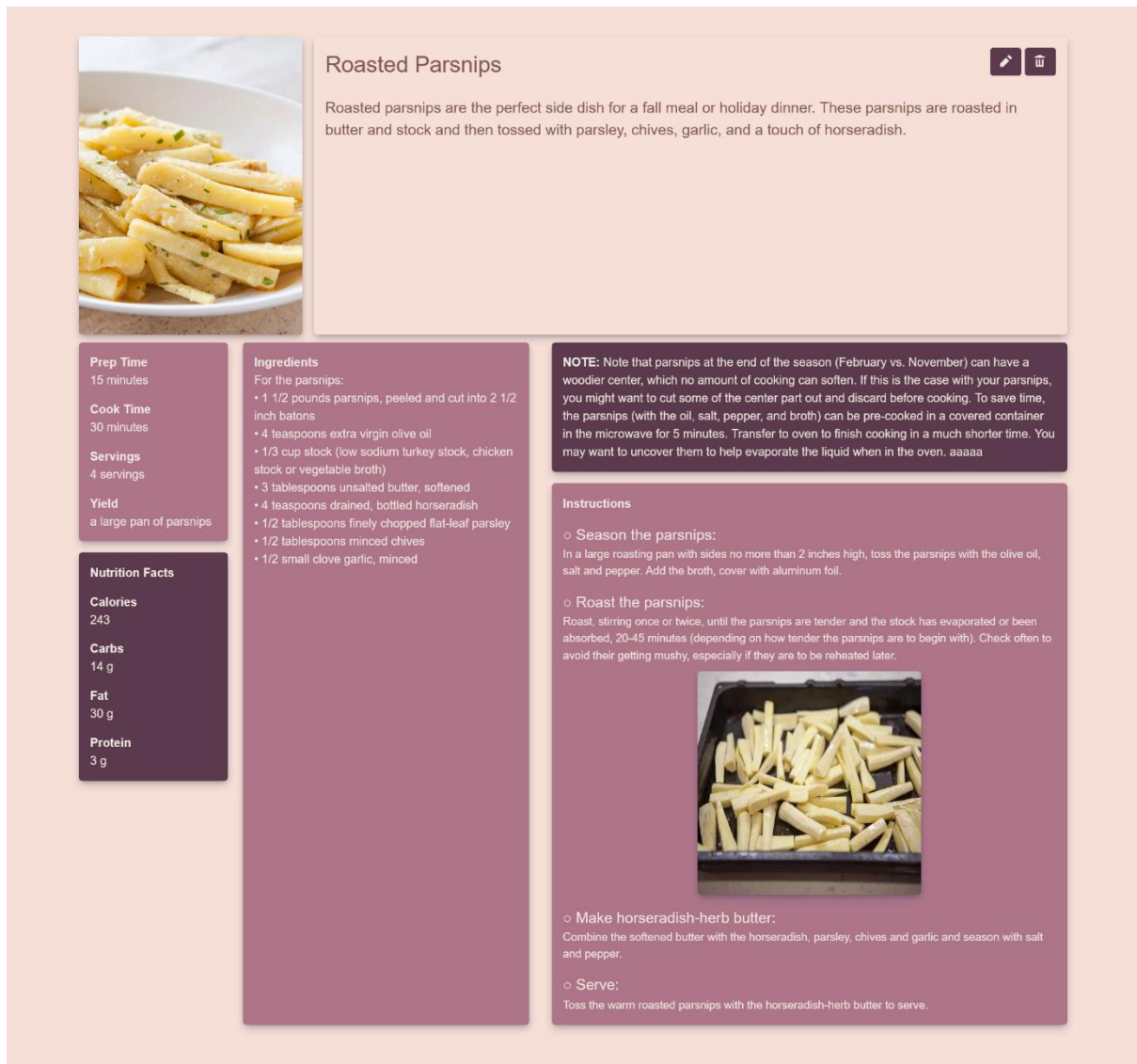
Figure IV.2. View recipe page

To edit a recipe, click the button with a pencil icon on it which is located on the card which displays the recipe image, the name of the recipe and the total time it takes to make it. The user will be redirected to the edit form which lets the user edit:

- The image for the recipe.
- The name of the recipe.
- The description of the recipe.
- The cookbook in which the recipe is.
- Prepping time, cooking time, additional time, servings and how much the recipe yields.
- Nutritional facts.
- Ingredient notes (ex: "For the crust:") and ingredients (name and quantity).
- Notes.
- Instructions and the image for each instruction.

To edit the recipe, click the button with a file icon on it which is located right under the dropdown menu which lets the user choose in which cookbook will the recipe be.

To cancel editing the recipe, click the button with a circled x located right under the dropdown menu which lets the user choose in which cookbook will the recipe be.

To delete the recipe, click the button with a trash can icon on it right under the dropdown menu which lets the user choose in which cookbook will the recipe be. A recipe can be deleted from the main page as well by clicking the button with a circled x on it which is located on the card which displays the recipe image, the name of the recipe and the total time it takes to make it.

Figure IV.3. Edit recipe page

To create a new recipe, click the button *New Recipe* on the main page which is located in the top-right corner. The user will be redirected to the edit form where they will be able to add:

- An image for the recipe.
- A name for the recipe.
- A description for the recipe.
- The cookbook in which the recipe will be added.
- Prepping time, cooking time, additional time, servings and how much the recipe yields.
- Nutritional facts.
- Ingredient notes (ex: "For the crust:") and ingredients (name and quantity).
- Notes.
- Instructions and an image for each instruction.

To save a recipe, click the button with a file icon on it which is located right under the dropdown menu which lets the user choose in which cookbook will the recipe be added.

To cancel creating the recipe, click the button with a circled x located right under the dropdown menu which lets the user choose in which cookbook will the recipe be added.

Figure IV.4. Add recipe page

To create a new cookbook, click the button *New Cookbook* on the main page which is located in the top-right corner. A modal dialog will pop-up where the user will be able to input the cookbook name. Click the *Save* button to add the cookbook. To cancel this action, click the *Close* button.

Figure IV.5. Add a new cookbook

To edit a cookbook, select it from the dropdown and click the button with a pencil icon on it next to the dropdown. A modal dialog will pop-up that will let the user edit the name of the cookbook. To save the changes, click the *Save* button. To cancel this action, click the *Close* button.
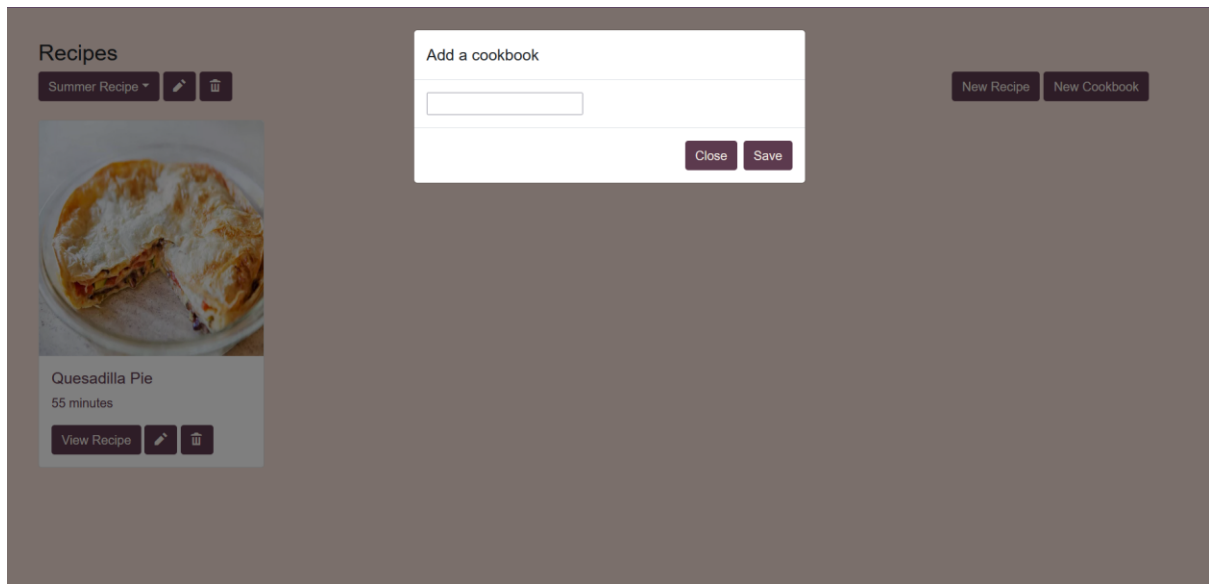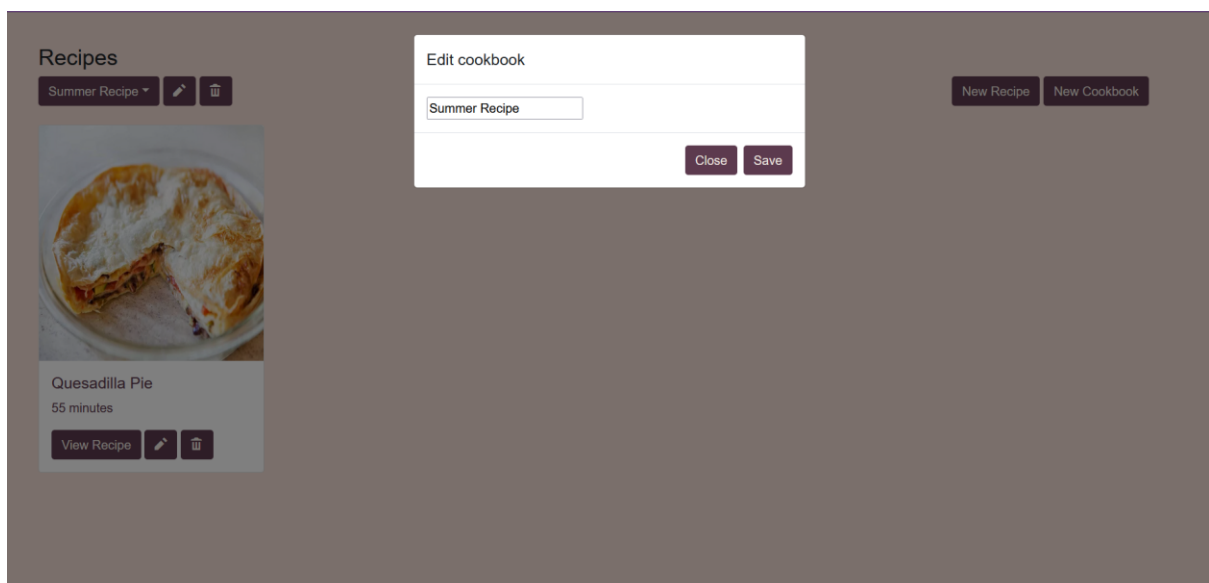

Figure IV.6. Edit cookbook

To delete a cookbook, select it from the dropdown and click the button with a trash can icon on it next to the dropdown. A modal dialog will pop-up that will ask the user if they want to delete the cookbook. To delete the cookbook, click the *Delete* button. To cancel this action, click the *Close* button.
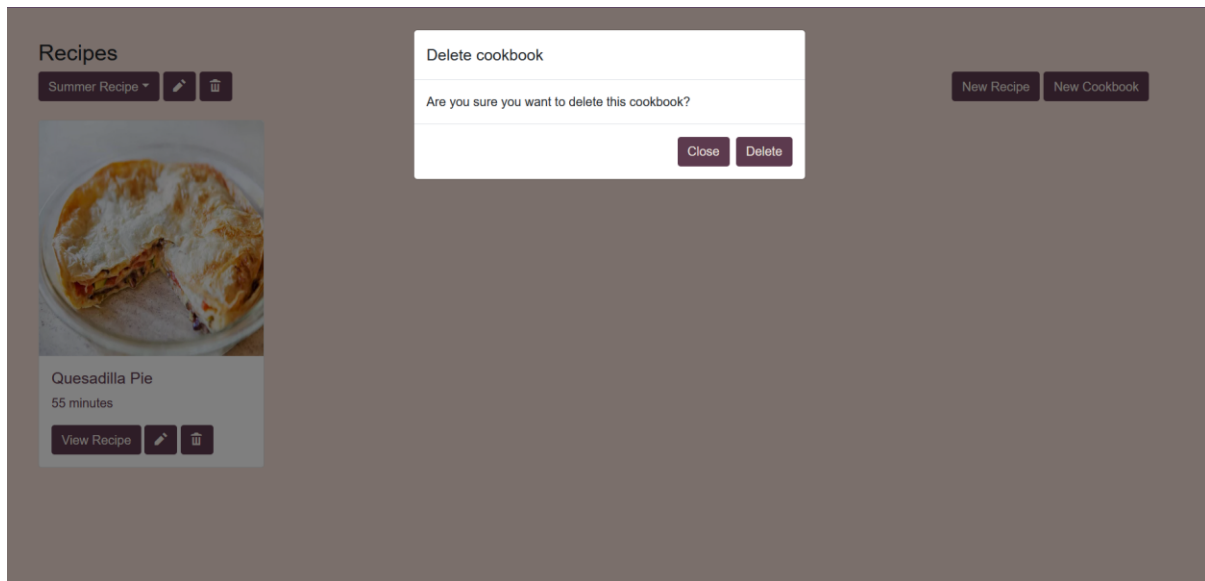
Figure IV.7. Delete cookbook

V.   Implementation

For this project, the *Blazor.Bootstrap (1.10.3)* (Client, Server and Shared projects), *Microsoft.EntityFrameworkCore   (7.0.13)*   (Server   project), *Microsoft.EntityFrameworkCore.Design   (7.0.13)*   (Server   project), *Microsoft.EntityFrameworkCore.SqlServer   (7.0.13)*   (Server   project), *Microsoft.EntityFrameworkCore.Tools (7.0.13)* (Server project) NuGet packages have been installed.

V.1   Shared project
The shared project contains classes for data about cookbooks (*Cookbook.cs*), recipes (*Recipe.cs*), ingredients (*Ingredient.cs*), ingredient notes (*IngredientNote.cs*), instructions (*Instruction.cs*) and nutrition facts (*NutritionFacts.cs*).

V.2   Server project
The database was created with a code first approach, specifying the primary key and the relationships in the Shared project classes. The DataContext.cs that inherits DbContext was created to map the entities and relationships that are defined in the model classes in the Shared project to the database. Services for connecting to the SQL Server database and adding the context for it have been added in the *Program.cs* file. The connection string for the SQL Server has been added in the appsettings.json (*SqlServerRecipesConnection*).

The *Migrations* folder was generated when updating the database through the Package Manager Console.

The *Controllers* folder contains API controllers that handle CRUD operations for the entities created in the Shared project.

V.3   Client project
A *Services* folder has been added in the Client project to communicate with the server-side API for CRUD operations related to the entities created in the Shared project. These services have been registered in the *Program.cs* class to be shared across all components.

The *Pages* folder contains three razor components that render pages which display the recipes and cookbooks in the database (*Index.razor*), add, edit and delete cookbooks (*Index.razor*), view a recipe (*RecipeView.razor*), edit or add a recipe (*RecipeForm.razor*).

To use BlazorBootstrap for the dropdowns and buttons, the instructions on the official web page have been followed, ignoring the *Remove default references* and the *Add CSS references*, the latter step resulted in an error when starting the application. More details can be found here: https://docs.blazorbootstrap.com/getting-started/blazor-webassembly [4].

VI.   Known Issues

There are some issues when choosing a cookbook from the dropdown menu; the recipes from that cookbook won't be shown until the dropdown is clicked again.

When editing a recipe, there is a problem when adding an image to an existing instruction, the image doesn't get saved in the database.

Stylesheets for each razor component in the *Pages* folder in the Client project didn't have any effect on the appearance of the HTML elements, therefore a style section was added directly in the razor component.

These issues and other bugs will be solved in a future update.

VII.   Future considerations

- User authentication together with a login and registration form.
- A search bar for finding recipes by name.
- A dropdown for sorting recipes by name,in alphabetical order, and cooking time.
- Fix known issues and other bugs that may appear.
- Add more page responsive structures.
- Add unit testing.

VIII.   Conclusion

The development of *MyCookbooks* encountered some challenges along the way but managed to provide an intuitive and efficient web app for organizing and creating cookbooks and recipes.

Future considerations include implementing user authentication, adding search and sorting functionalities, improving responsive design, resolving existing bugs, and introducing unit testing.

IX.   Bibliography

[1]   https://github.com/patrickgod/BlazingShop

[2]   https://learn.microsoft.com/en-us/aspnet/core/blazor/project-structure?view=aspnetcore-7.0

[3]   https://learn.microsoft.com/en-us/dotnet/architecture/blazor-for-web-forms-developers/project-structure

[4]   https://docs.blazorbootstrap.com/getting-started/blazor-webassembly