COMP 3095 – Web Application Development with Java

Assignment #2
Create a Recipe Project Using Spring/Spring Boot
Part B

Table of Contents

Assignment 2 – Recipe Spring Boot Application

Group Assignment: 2 - 4 members per group (mandatory)
Due Date: Sunday, December 5th, 11:59 pm

1. Objective

The goal of this assignment is to extend Assignment 1 and complete the building a Spring Recipe Assignment Spring Boot. We continue to with the concepts of service layers and Spring MVC.

2. Assignment Instructions/Steps

Please note, because this is an assignment, the instructions provided are not instructive as say in a class lab. The expectation is for you to research those unfamiliar elements within your group, to determine how to implement them in your solution. **The design of the final solution is left entirely up to you and your team.**

3. Background

The objective of this second COMP3095 assignment is to complete the development of the Spring Recipe application that can be used to help facilitate a client/user's kitchen skills and modernize grocery shopping, that is, instead of relying on index cards, a rolodex of a file folder.

Requirements

1. The objective of this assignment is to extends your first assignment (Assignment #1 Spring Boot Project), the **mandatory requirement** is fundamentally the same, that is for you to utilize the following dependencies at a minimum:
   - Java version 11
   - Spring Boot
   - Spring DevTools
   - Spring Web
   - Thymeleaf – must be used for your view layer
   - Spring Data JPA
   - H2 Database (embedded) – must be used for your database
   - MVC design – utilizing a combination of controllers, service and repository layers.

Use Case Diagram

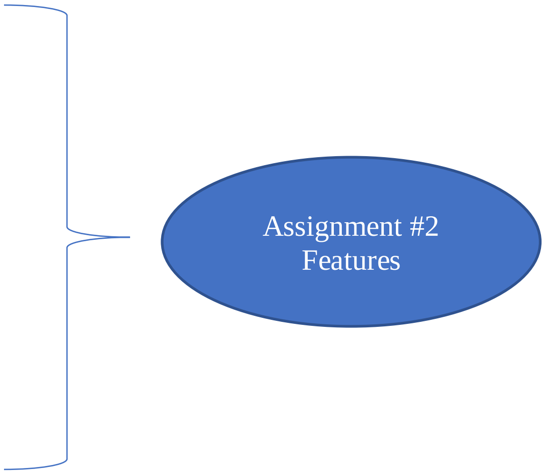
For this first part (Part B/ Assignment #2) you are to satisfy the following use cases.

Unregistered Users

- ~~Register into the application - Assignment 1~~

Registered Users

- ~~Login Into the application – Assignment 1~~
- ~~Logout of the application – Assignment 1~~
- ~~Create a recipe – Assignment 1~~
- ~~View a Recipe – Assignment 1~~
- ~~Search for a Recipe – Assignment 1~~
- ~~Plan a meal – Assignment 1~~
- ~~View Profile – Assignment 1~~
- View Ingredients
- View Steps
- Mark as Favourite
- View Shopping List
- View My Recipes
- View My Favourite Recipes
- Edit Profiles
- View My Event Plans
- Add to Shopping Cart
- Delete Ingredients
- Update Events
- Delete Events



Use Case Diagram – Assignment #2

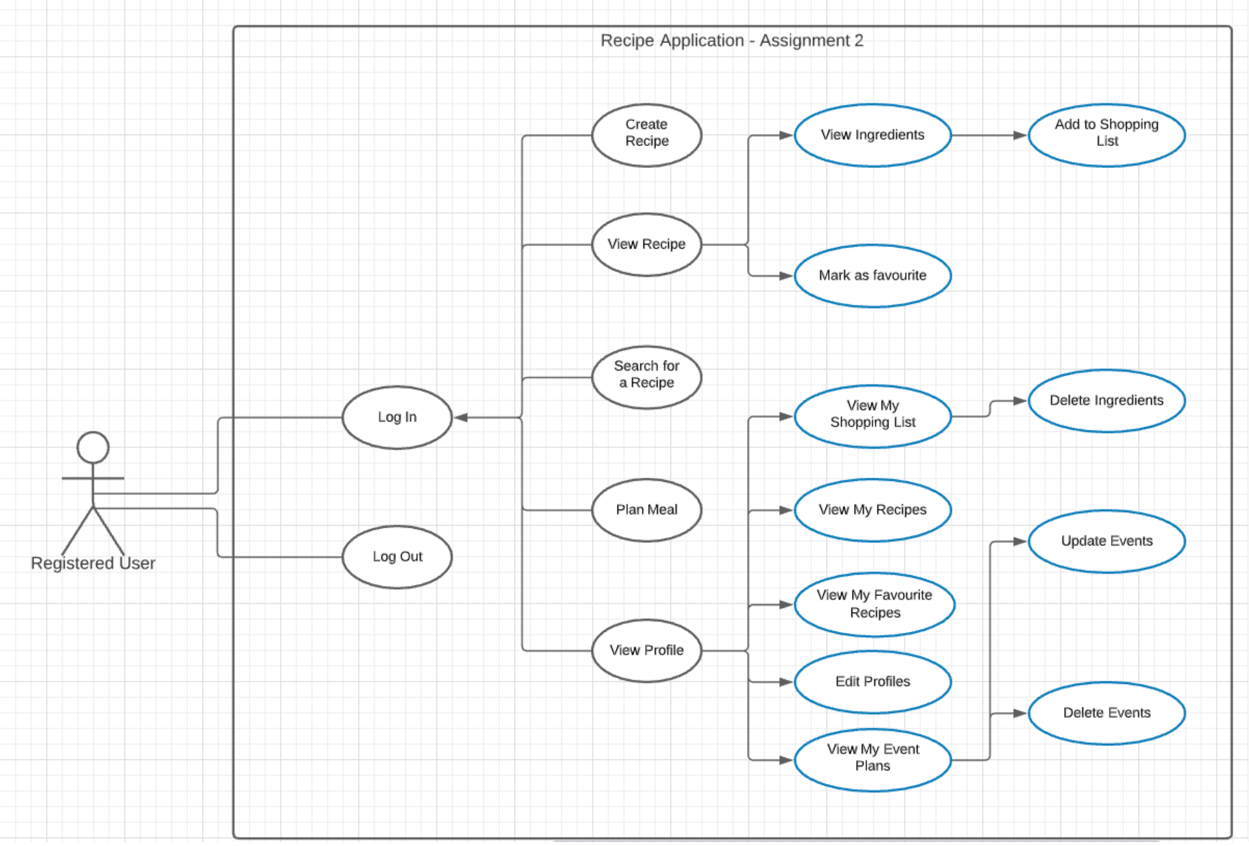


Figure 1: New features highlighted in Blue

Functional Requirements

For this second part (Part B/ Assignment #2) you are to satisfy the following use cases.

View Ingredients: A registered user should be able to view ingredients for a certain recipe. This option should allow the user the ability to mark those ingredients as needed, and add them to the shopping list.

Mark as Favourite: A registered user, after viewing a recipes information, will be presented with the added ability to mark the recipe as a favourite.

View My Shopping List: Registered users must have the ability to view their shopping list. The shopping list should exportable (i.e. be able to download and print).

View My Recipes: Registered users must have the ability to view all recipes they have created exclusively.

View My Favourite Recipes: Registered users must have the ability to view and search through their marked favourite recipes

Edit Profile: A Registered user must have the ability to edit their profile, editing such things as username, profile image/icon, password reset if desired.

Password Reset / Forgotten Password: The design and implementation of a password reset capability, is incumbent upon the developers to determine and ultimately design, but such a feature is a mandatory requirement in any web application.

View My Event Plans: Registered users must have the ability to view the list of events that they have created.

Add to Shopping List: Registered users must have the ability to maintain a shopping list. The list is generally utilized as a means to plan shopping for the user. Items can be added and removed from the list, and the system should allow of the typical CRUD operations.

Update/Delete Ingredients: Registered users must have the ability to update/delete ingredients for their recipes.

Update/Delete Events: Registered users must have the ability to update/delete an event.

Final Thoughts

This is the final assignment of the course. As described in the first assignment, there are many areas of group interpretation, that is, each groups creativity is welcomed.

The desire is to see a wide range of differing interpretations on how to solve the client/application needs. The scope of interpretation is left up to each group, **provided the minimum requirements as specified within this document, are met.**

Things you will need to consider and design in order for your application to be deemed a success:

- The application requires a form of data persistence. H2 database (embedded) is a mandatory requirement. Your database should be seeded with any necessary configuration/operational data, upon startup (if needed).
- Your professor should/must not have to run and SQL scripts to have your fully functioning database up and running, that is, a simple run/deployment is all that should be required from your professor in order to test and use your fully functioning solution.
- Presentation Layer, Thymeleaf is a compulsory requirement, however, the UI design and creativity are left to you and your group members.
- This is the final assignment (assignment 2), you will want to ensure you have a functioning assignment 1, to build on top of to ensure you have no issues implementing the necessary use cases.
- Your solution must meet each of the requirements documented within to gain full marks at a minimum.
- Design is key, as a groups submission will be evaluated against all submissions within the class.
- Assign work to all group members, the recommendation is to use some form of version control, create and distribute tasks as demonstrated within class. A group lead is recommended.
- This is a **mandatory group assignment**, please select your team members diligently, as the distribution and contribution of each group members effort will directly result in how well and quickly your assignments are constructed.
- Individual submission will be docked marked **-10%**

Assignment Submission Guidelines:

1. You must email your assignment to your Professor at **Sergio.Santilli@georgebrown.ca**

2. All members in the project team must be cc'd on the final assignment submission. Failure to do so will result in a mark of zero for those members not cc'd on the assignment submission email.

3. Within the body of the email, clarify course code, team name, team members and student numbers. Title the email accordingly **COMP 3095 – Assignment 2**.

   Example:

   Course: COMP 3095
   Team Name: The Hackers
   Team Members: John Smith - 1234567
   Sally Jones - 7654321
   Jane Wilson - 2342342

4. When submitting, cc' a copy to yourself for backup and time verification.

    1. The project submission, must contain the complete project source code compressed in **.zip** format (no source code, no marks)

5. Each java file (.java) should include a header.

```
//********************************************************************************
* Project:              < project name … >
* Assignment:           < assignment # >
* Author(s):            < author name …>
* Student Number: < student number … >
* Date:
* Description:          <describe the java file and its purpose briefly only – 1 or 2 lines>
********************************************************************************//
```

6. Your code should be modular and should show no signs of dry (don't repeat yourself) code.

7. Be cautious **DO NOT** share your application with others. Complete failures will be assigned if code is shared. All assignments will be reviewed and analyzed strictly within these regards.

8. Late assignments are assigned at a penalty of **25%** per day.