CSE3OAD/CSE4OAD - Assignment 1

Due Date: 10 am Monday 18 September 2017

Assessment: This assignment 1 is worth 15% of the final mark for CSE3OAD and CSE4OAD.

This is an individual assignment.

Copying, Plagiarism: Plagiarism is the submission of somebody else's work in a manner that gives the impression that the work is your own. The Department of Computer Science and Computer Engineering treats plagiarism very seriously. When it is detected, penalties are strictly imposed. Students are referred to the Department of Computer Science and Computer Engineering's Handbook and policy documents with regard to plagiarism.

No extensions will be given: Penalties are applied to late assignments (5% of your total assignment mark given is deducted per day, accepted up to 5 days after the due date only). If there are circumstances that prevent the assignment being submitted on time, an application for special consideration may be made. See the departmental Student Handbook for details. Note that delays caused by computer downtime cannot be accepted as a valid reason for a late submission without penalty. Students must plan their work to allow for both scheduled and unscheduled downtime. Assignments submitted more than 5 days late (i.e. after 10 AM, Monday 19th September 2016) will receive the mark of 0.

Return of Assignments: Students are referred to the departmental Student Handbook for details.

Objectives: To design and implement an application in JavaFX and to use SQL and JDBC for data persistence.

Introduction

The aim of the assignment is to create an application to maintain a collection of recipes.

The recipes are stored in a MySQL database. The system consists of the following main classes:

- Recipe
- RecipeDSC
- RecipeMain
- RecipeStageMaker
- Ingredient

The Recipe class represents the recipe. This class is provided. As can be seen from the provided code, the class has the following attributes:

- id: the id of the recipe. It uniquely identifies a recipe.
- name: the name of the recipe.
- serves: the number of servings the recipe's description is for, an integer.
- ingredients. This attribute, of type Arraylist<Ingredient>, shows the ingredients used in the recipe.
- steps. This attribute, of type String, contains the instructions to cook the food.
- remarks: to store any further remarks the user want to make about the recipe.

The id's of the recipes are actually automatically generated by the database, as can be seen in the SQL script. You can also see this from the add method in the RecipeDSC class.

(So you may ask: why do we have the constructor in class Recipe? This will be left for you to figure out.)

The second class, RecipeDSC, is the data source controller. The Data Source Controller (DSC) is the class that handles all of the add/remove/update operations between your running program and the database. A skeleton of this class is provided.

The third and the fourth classes, RecipeSystem and RecipeStageMaker, provide the graphical user interface for the users to interact with the system. These classes are to be implemented in JavaFX. A skeleton for each of these classes is provided.

An SQL script is provided to create the tables, and seeds the database with some sample data

Task 1

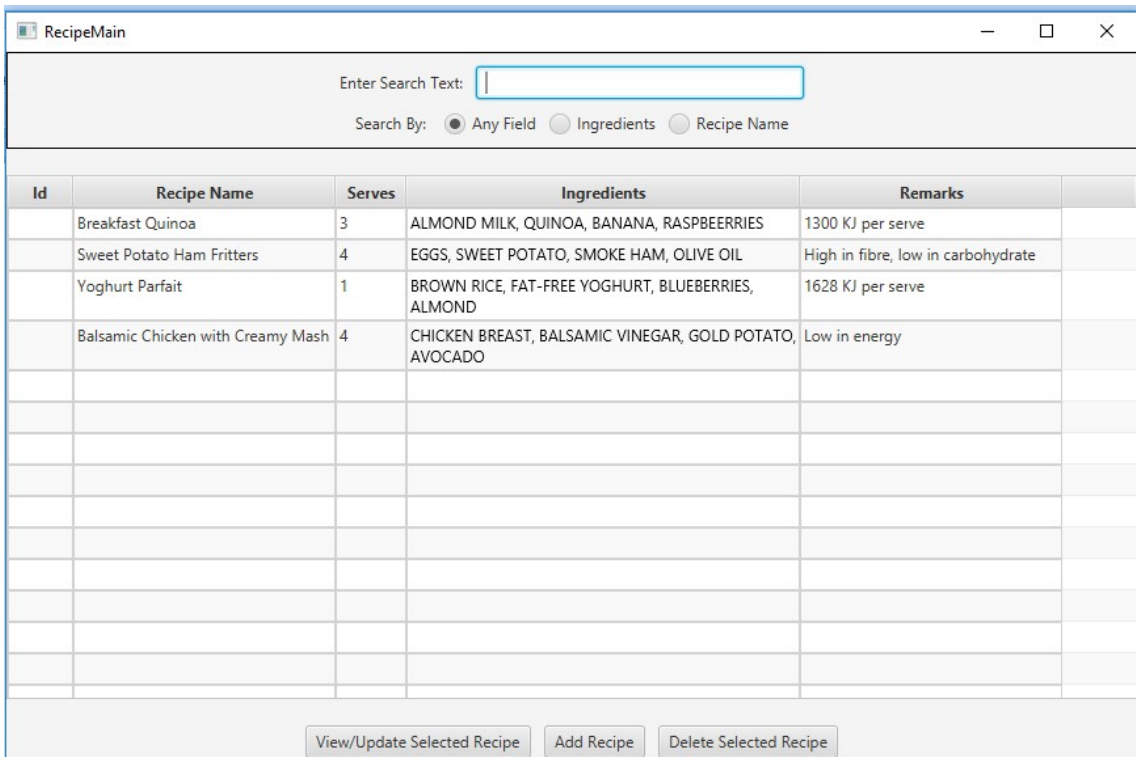
Implement the RecipeDSC class, with

- The ability to get the username and password from the user (i.e., not hard coded).
- Be able to connect to the database and retrieve and display the sample data.

A skeleton of the class is provided, which indicates the methods that you are required to implement.

Task 2

Implement the RecipeMain class. When the system is started, a screen similar to the one shown in Figure 1 is displayed.


RecipeMain
Enter Search Text:
Search By: Any Field Ingredients Recipe Name
View/Update Selected Recipe
Add Recipe
Delete Selected Recipe

Search By: [✓ Any Field] Ingredients Recipe Name

Id	Recipe Name	Serves	Ingredients	Remarks
	Breakfast Quinoa	3	ALMOND MILK, QUINOA, BANANA, RASPBEERRIES	1300 KJ per serve
	Sweet Potato Ham Fritters	4	EGGS, SWEET POTATO, SMOKE HAM, OLIVE OIL	High in fibre, low in carbohydrate
	Yoghurt Parfait	1	BROWN RICE, FAT-FREE YOGHURT, BLUEBERRIES, ALMOND	1628 KJ per serve
	Balsamic Chicken with Creamy Mash	4	CHICKEN BREAST, BALSAMIC VINEGAR, GOLD POTATO, AVOCADO	Low in energy

Figure 1. Screen displayed by RecipeMain

Searching. At the top of the screen, we can enter text that is used for searching. If we check "Any Field", the table view displays recipes that have any field containing the search string (this constitutes a "match"). If we check "Ingredients" button, the table view displays recipes that have the ingredients containing the search string. And if we check “Recipe Name” button, the table view displays items that have the name containing the search string.

Table View. The table view displays the recipes in the collection, one recipe per row. The table view does not display the steps of the recipes, or the quantities of each ingredient. The steps will be displayed when the user select a recipe and clicks the “View/Update Selected Recipe” button. The ingredients field is displayed in the text-wrap mode. An example of the ingredients is shown below:

```
ALMOND MILK, QUINOA, BANANA, RASPBEERRIES
```

The string shows four ingredients. The names of the ingredients are in capital. The ingredients are separated by commas.

Buttons for all key functionality. This is the “View/Update Selected Recipe” button, “Add New Recipe” button, and “Delete Selected Recipe” Button.

Task 3

Implement the RecipeStageMaker class and link it to the three buttons on the primary screen.

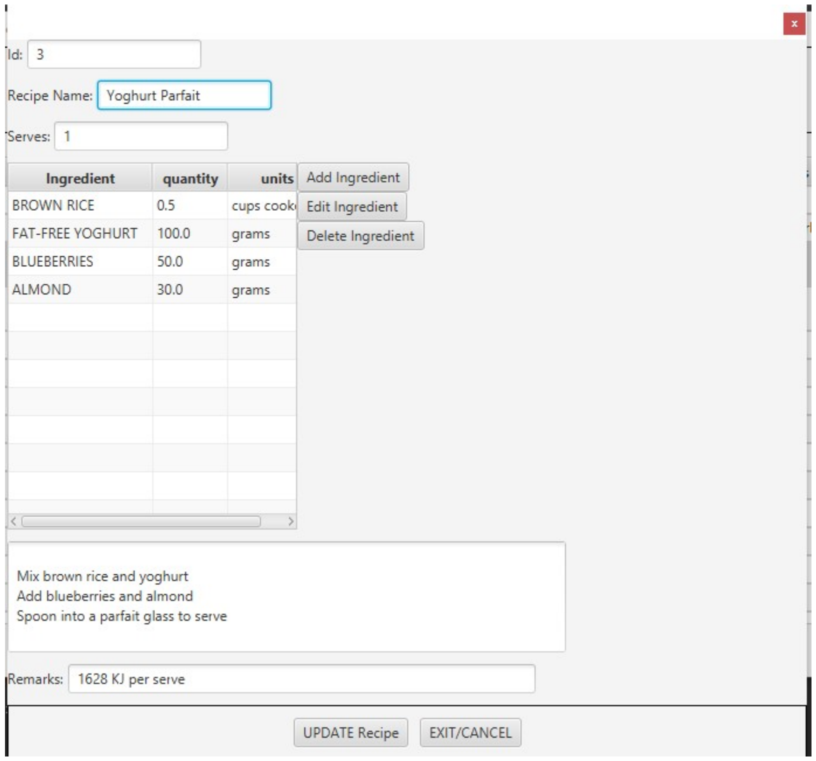
“View/Update Selected Recipe” button.

When we click on the “View/Update Selected Recipe” button, the program calls method showViewRecipeStage of the RecipeStageMaker class, which should display a screen similar to the one shown in Figure 2.

The program should display this screen some distance to the right and below that of the main screen.

The stage displays all the details of the selected recipe for the user to view. The user can also update the details of the displayed recipe. To do so, the user changes the details displayed in the text fields and text area, except the id. The user can choose to update the recipe (by clicking on the “UPDATE Recipe” button, or cancel the operation (by clicking on the “EXIT/CANCEL” button).

Note: if cancel is pressed then no changes should occur to the recipe, this includes the ingredient list. Also, closing the window should follow the same behaviour as cancel.



Id: 3

Recipe Name: Yoghurt Parfait

Serves: 1

Ingredient	quantity	units
BROWN RICE	0.5	cups cook
FAT-FREE YOGHURT	100.0	grams
BLUEBERRIES	50.0	grams
ALMOND	30.0	grams

Mix brown rice and yoghurt
Add blueberries and almond
Spoon into a parfait glass to serve

Remarks: 1628 KJ per serve

Figure 2.

“Add New Recipe” button.

When we click on the “Add New Recipe” button, the program calls method showAddRecipeStage of the RecipeStageMaker class, which should display a screen similar to the one shown in Figure 3.

The user can enter the details, and then choose to add the new recipe or to cancel the operation.

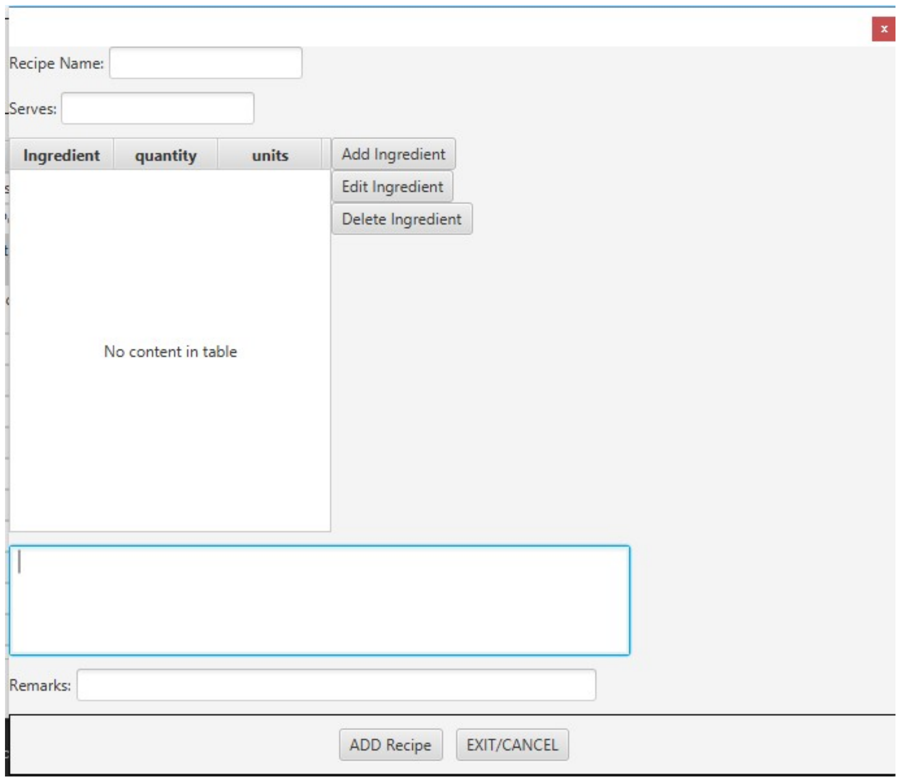


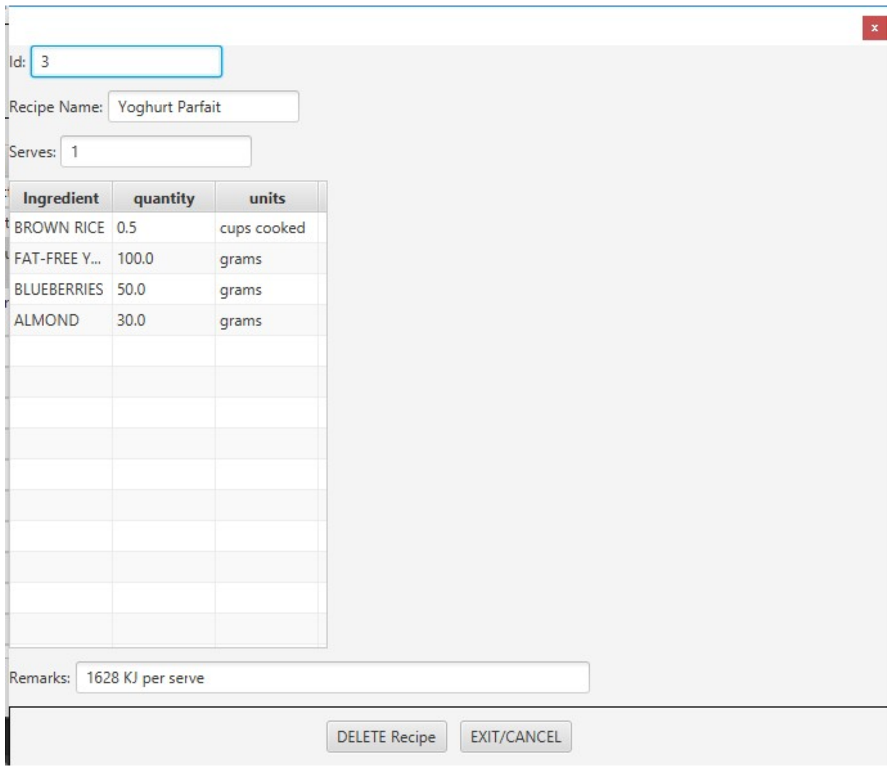
Figure 3.

“Delete Selected Recipe” Button.

And when we click on the “Delete Selected Recipe” button, the program calls method showDeleteRecipeStage of the RecipeStageMaker class, which should display a screen similar to the one shown in Figure 4.

The screen displays details of the recipe selected in the table view. The text fields and the text area should be un-editable.

The user can choose to delete the recipe or to cancel the operation.


Id: 3

Recipe Name: Yoghurt Parfait

Serves: 1

Ingredient	quantity	units
BROWN RICE	0.5	cups cooked
FAT-FREE Y...	100.0	grams
BLUEBERRIES	50.0	grams
ALMOND	30.0	grams

Remarks: 1628 KJ per serve

Figure 4.

Adding updating and deleting Ingredients

During both the view/update and the adding of recipes the user will need to be able to also add and update the ingredients of that recipe.

This will require you to make a 3rd stage dedicated to the creation and editing of ingredients. It is up to you to design a user friendly interface for this.

Note that at any stage if the user selects to cancel on the adding or view/update stage then the data in the database should remain the way it was when the user first opened the dialog.

Exception Handling

Make sure you add the exception handling code so that whenever an exception occurs, the program should not crash and furthermore it should display an alert which shows a brief message about the exception.

Files Provided

- Recipe.java
- RecipeDSC.java
- RecipeMain.java
- RecipeStageMaker.java
- Ingredient.java
- CreateDatabase.sql

What to submit

- Electronic copy of all of the classes required to run the application, including those that are provided, are to be submitted to latcs8 using the

  ```
  submit OAD <directory or filename>
  ```

  command. Note that the submission is not through LMS.
- All of your classes must be able to be compiled from their current directory. This means, they must not be contained in any package.
- For classes that you completed or created, you must include, as part of the comments,
  - your name (with the surname in capital, e.g. John SMITH),
  - your student ID,
  - your user name, and
  - the subject code (CSE3OAD or CSE4OAD)

Assignments without any of these will have 5% of the mark deducted.