FACH HOCHSCHULE LÜBECK

University of Applied Sciences

Software Engineering II Project

Digital Cookbook

Part task: Include picture in recipes

Group 4

Hu Zhengjiang

Zhang Yujia

Zhou Jiacheng

Hu Bocheng

Li Fangtian

Content

1. Specification

1.1. Description

1.1.1. Digital cookbook

Have you met a problem that you do not know what and how to cook with the ingredients you have? Our App, *Over Chef* , is a digital cooking software, providing functionalities such as quickly off-line searching, updating recipes based on users' preference and add use's own recipe. *OverChef* helps users to cook a dish in an efficient and less wasting way.

When user decides what dish he/she wants to cook, he/she can search recipes by the recipe's name and get a list of relevant recipe. Then by clicking on a certain recipe, user will get into a recipe detail window, which containing description of the recipe, all needed ingredients' information, cooking and preparing time and instructions of cooking steps. User can change the serving number and will get a response as ingredients' quantities' changing. User can also search recipes by the ingredient's name.

Pictures of recipes will be included in *OverChef*.

OverChef provides user with utility to add, edit and delete recipes. It can be a easy and efficient guide for a user to make delicious dishes.

1.1.2. Additional Task

User can add his/her own recipe and picture into *OverChef* with given file standard (.jpg/.png).

1.1.3. Product functions

User can:

1. Search for recipes, e.g. by recipe's name or ingredient's name.

Search function will allow user to search a recipe in recipe's name or ingredient's name. If there is no such recipe in this name, the searching result will return either "no recipe found" or find a most similar recipe (This depends on if there any relevant key words).

2. Add, modify and delete any recipe.

User can add a new recipe with standard form (Recipe name (required), description, ingredients, steps, picture, serving number, and time) in main window, and this one will add into database as well. User can change the quantity of ingredients as well as the steps according to their preference in recipe modify window. If user do not like one recipe, he or she can delete it in recipe modify window, and this one will delete from the database as well.

3. Get the quantity of each ingredient for serving certain number of people.

In recipe detail window, user can change serving number and new quantity of each ingredient.

4. Change, save and delete picture for each recipe.

In recipe modify window, user can change, save and delete picture for this recipe.

1.2. User characteristics

a.) Users are assumed to:

1. Be oversea students, housewife/husband, someone have the need to cook, or just like cooking.
2. Speak English.
3. Have basic knowledge in cooking. (E.g. know how to use a knife without hurting anyone.)
4. Have basic computer operation knowledge. (E.g. know how to open our App on computer.)
5. Have a device that can install our App.

b.) Users need the software to:

1. Find a name-known dish. (E.g. User knows what he/she wants to cook but do not know steps.
2. Find dishes with name-known ingredient. (E.g. user has an ingredient but do not know what dish to cook.)
3. Get the quantity of each ingredient for a certain serving number in a recipe.
4. Store user's own recipe and its picture.

1.3. Functional requirements

1. Allow user to add, modify and delete any recipe.

2. Allow user to search for a recipe.

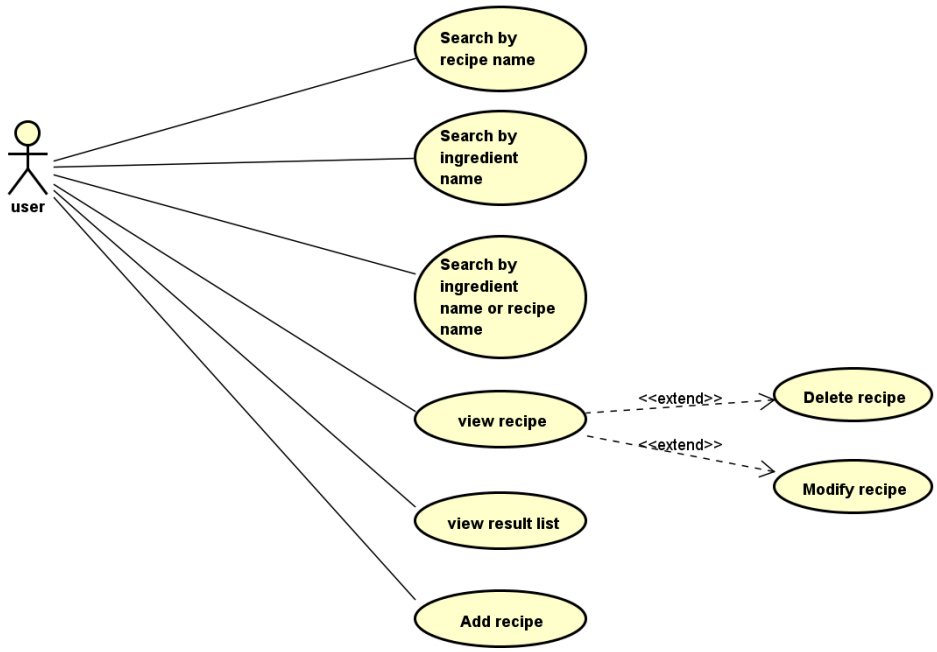
3. Allow user to change serving number of a recipe to get right quantity of each ingredient.

1.4. Non-functional requirements

1. Take less than 100M storage in device.
2. The software is programmed by Java
3. The software should have clear instruction.

2. UML Specification

2.1. Use Cases and ER - Diagram



2.1.1 Use Case Diagram

The *Overchef* app will load all the recipe information to the memory from the database when the app opened. Considering the number of the recipes will not occupy too much memory and the reading speed requirement from the search method, we made the decision of initializing the recipe data from database at the first beginning.

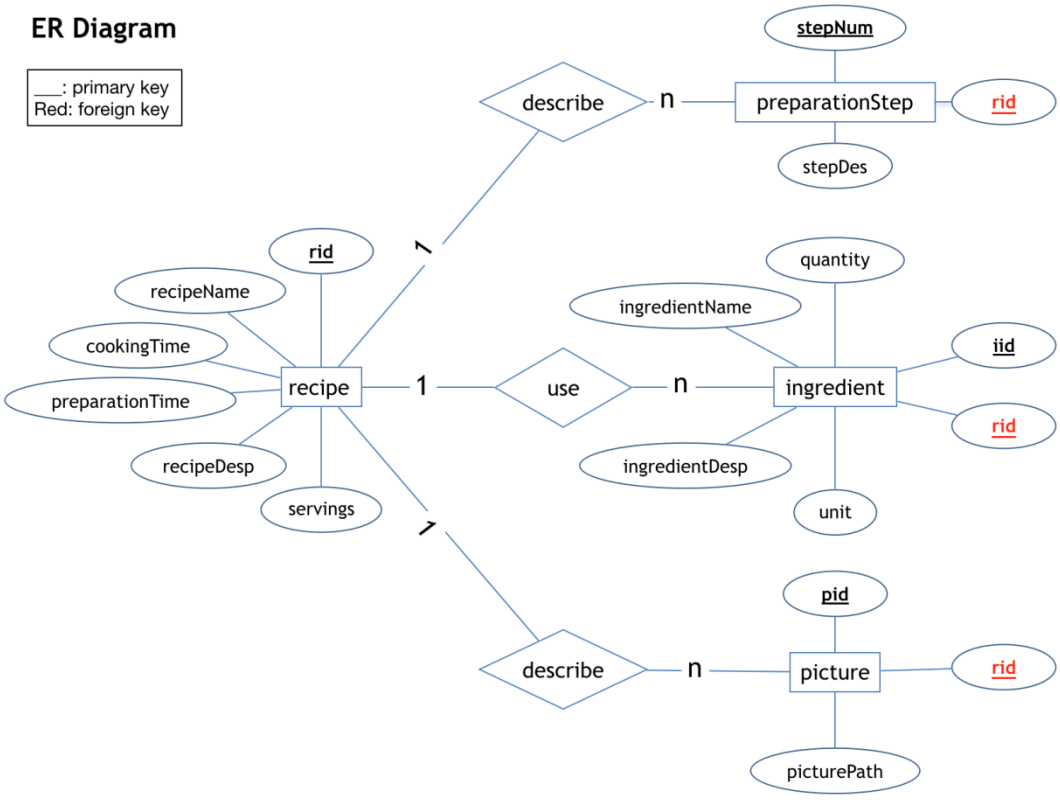
The software will provide user with an interface to enter the keywords. These keywords user entered in the view will be transmitted to the search method as parameter, then the search method in the controller will search for the relevant

recipes. The app provides three search options: search by ingredients, search by recipes and search by both of them. User can get result simultaneously when each single character is typed. Results recipes are showed in the table view as a list.

User can view recipe details by double clicking the recipe in the table view. A new window will be opened to display the details of clicked recipe. Three options are offered on the bottom right of the recipe view, which are 'delete', 'edit', and 'back'. User can delete the certain recipe by clicking the 'delete' button, can enter the recipe modify view to edit the recipe by clicking the 'edit' button, and can go back to the main view by clicking 'back' button.

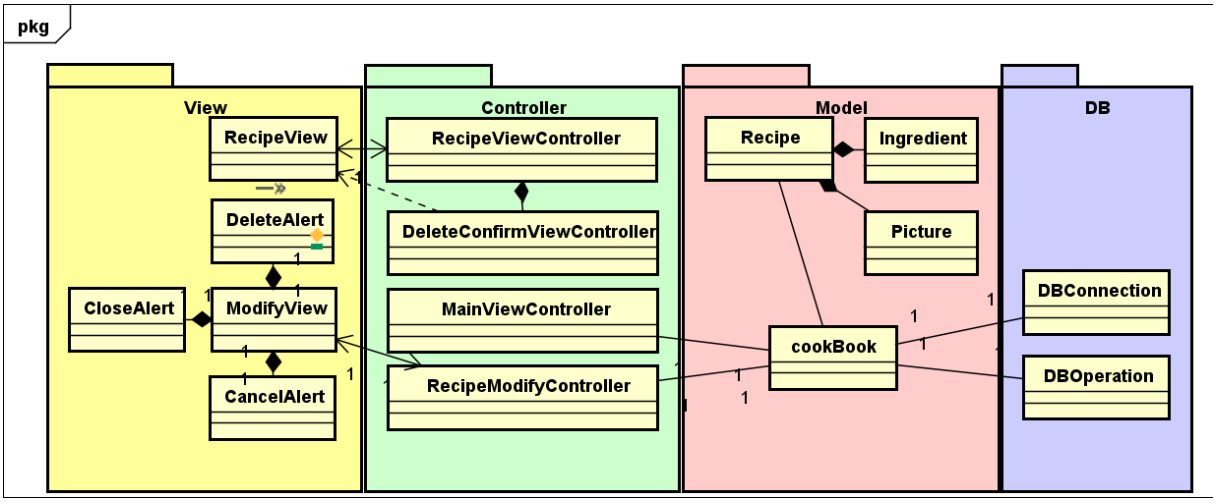
Back to the main view, User can add a new recipe by clicking the 'add' button and add the recipe details in the recipe modify view. And then newly-added recipe will be saved by 'save' button clicked or cancelled by 'cancel' button clicked. To avoid 'cancel' mistake, an alert box will be open to ensure the user want to cancel the edition.

'Delete a recipe' , 'add a recipe', and 'modify a recipe' will connect to the database and directly operate with the database at the same time doing the same operation like deleting or adding or modifying in the memory, but searching will only search in the memory.

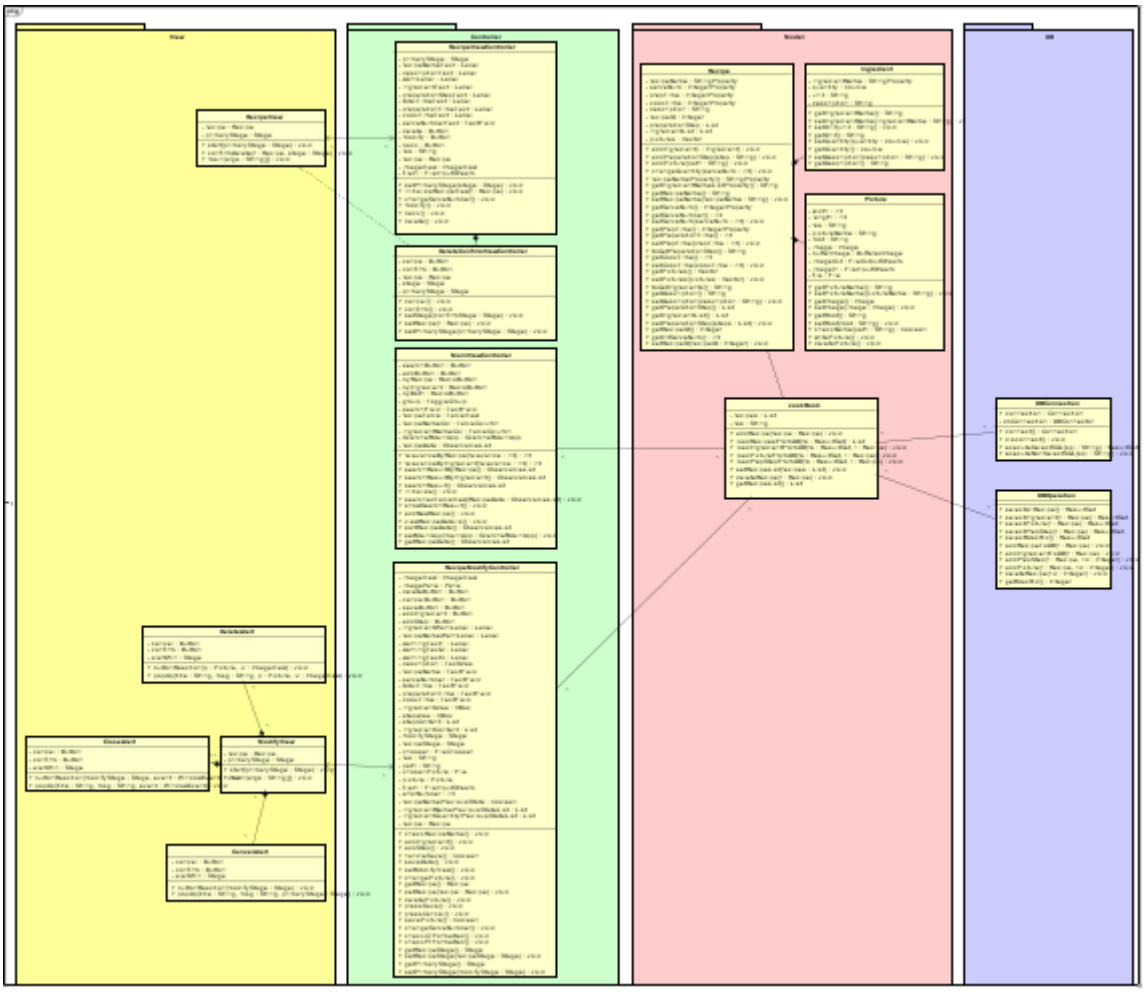


2.1.2 E-R Diagram

2.2. Class Diagrams



2.2.1 Class diagram overview



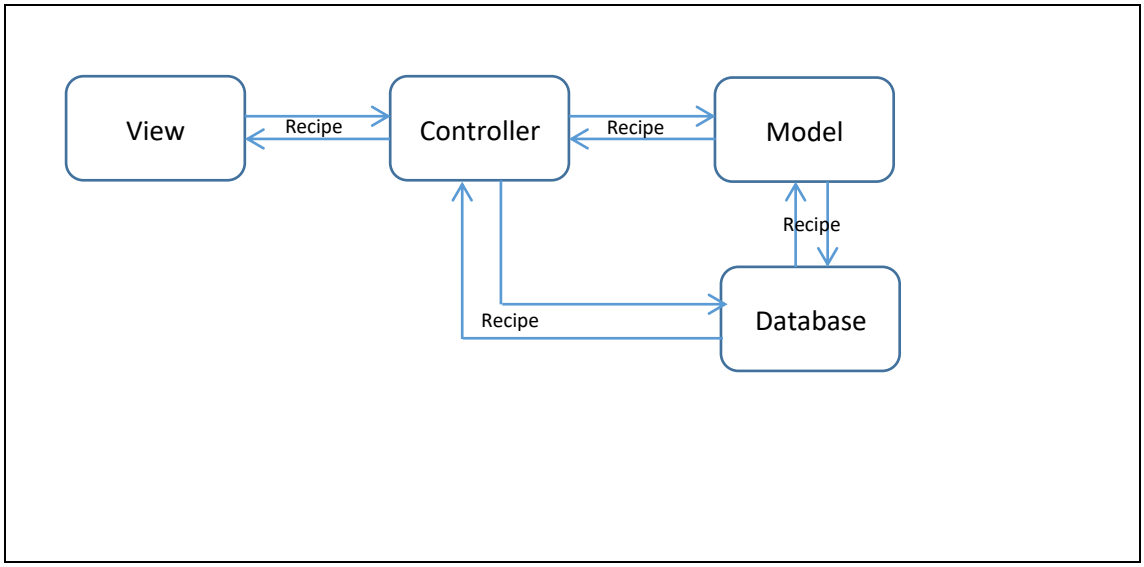
2.2.2 Detailed Class diagram

3. GUI Design

3.1. Structure

In the part of GUI designing, the design pattern Model View Controller (MVC) is applied to the structure of the software. The MVC design pattern separates the structure of the software into 3 layers which are Model layer (business logic layer), View layer (presentation layer) and Controller layer (link layer). The View is responsible for interacting with user, collecting data and passing data into controller which provides interfaces to accept inputs from the view and has access to the model and database. The controller will call functions of model which constitutes the business logic and interacts with database to process the data, get results and then pass results back to the view to present the results to the user.

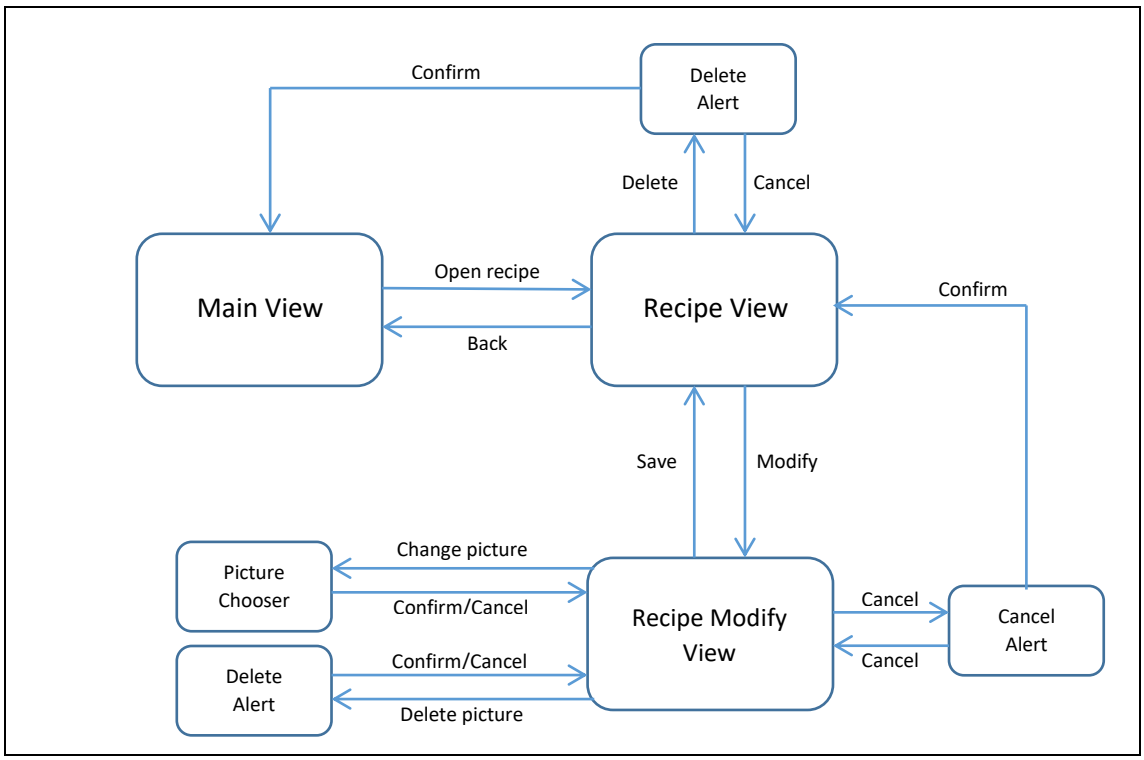
The benefit of the MVC design pattern is that these three layers are separated from each other, which makes each layer has less concern and disturbance from the other layers. In the project OverChef digital cookbook, each layer is transparent from other layers, communications between layers are implemented through passing the parameter recipe (a data type of Recipe created in model) containing all the information about a recipe.



3.1.1 MVC Design Pattern

Thus, the structure of GUI, which belongs to the presentation layer, is separated from the model layer, and connected with model layer through the controller. In the *OverChef* project, 3 primary views which are main view, recipe view and recipe

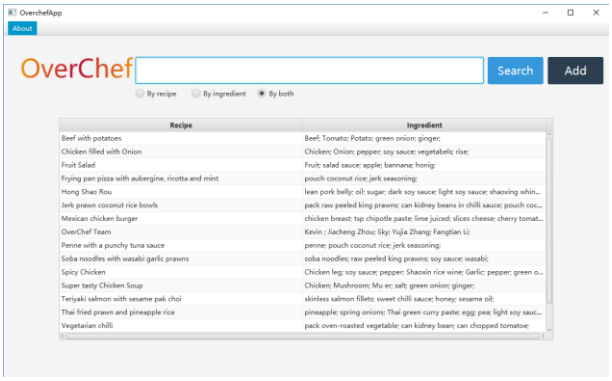
modify view and several alert windows based on JavaFX compose the GUI of the software.



3.1.2 GUI Structure (View Layer)

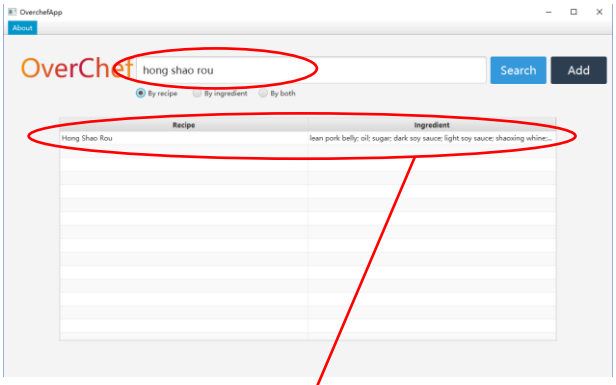
In general, according to the MVC design pattern, the GUI of this software is responsible for presentation and interaction with users, which is isolated from the model layer. All the operations and inputs happening on the GUI will be detected and collected by the controller and then passed into model and database. The result from model and database will then be passed to the GUI through controllers to display the result to the users. The GUI has no direct interaction with model layer and the main parameter used to be passed through GUI, controller, model and database is the recipe (A class created in the model layer), which to some extent makes the interaction between different layers simple and stable.

3.2. Screenshots



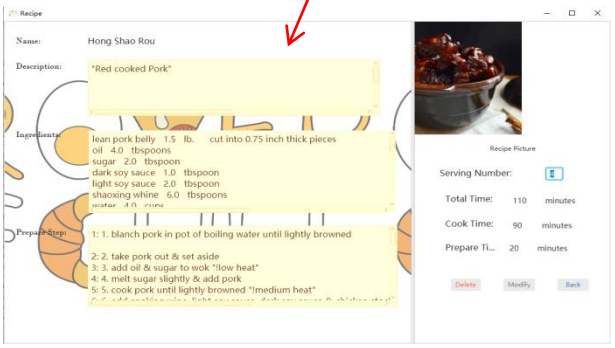
Main View

In the main view, users are allowed to input the keyword in the search field to search the recipe by recipe name, ingredient name or by both of them.



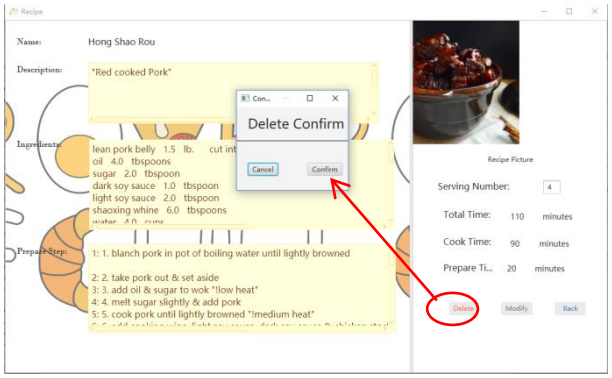
Searching for a recipe

Besides that the user can click “search” button to get the results displayed in the table view below the search field, the result will be shown in the table dynamically at the same time when user is typing keywords in the search field. A specific recipe will be opened in the recipe view when user double clicks the recipe in the table view. Also, user can click “Add” button to open a recipe modify view to edit a new recipe.



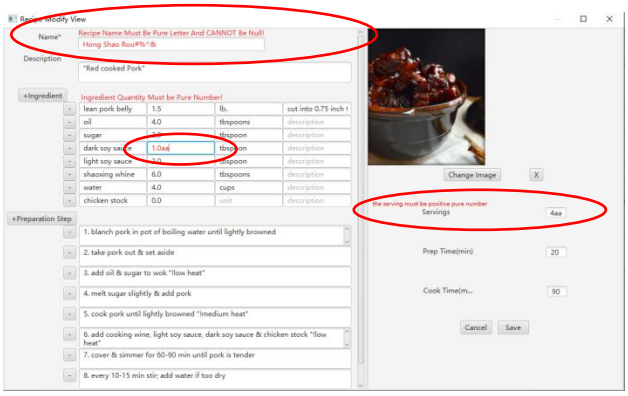
Recipe view(after opening a recipe)

In the recipe view, a recipe opened by users will be displayed in detail. This view presents all the content of the recipe including recipe name, recipe description, ingredients, and so on. Basically this view is a split pane, the left side displays recipe name, recipe description, ingredients and preparation steps while the right side shows a picture of the recipe, servings, cooking time and preparation time.



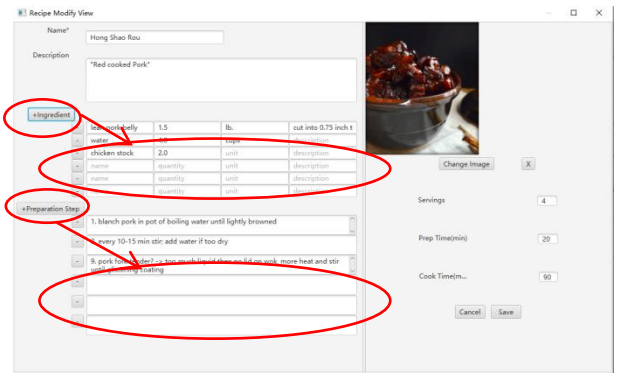
Alert window for deleting a recipe

User can delete a recipe here by clicking the “delete” button. An alert window will pop up to let the user confirm if he wants to delete the recipe.



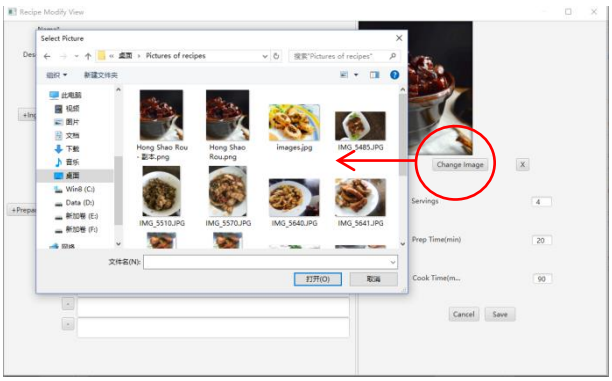
Recipe Modify View(checking format)

In the recipe modify view, no matter whether the user is adding a new recipe or modifying an existing recipe, this view will be responsible for editing the content of the recipe and save the recipe into database. The formats of some of the inputs from user will be checked in case of illegal data.



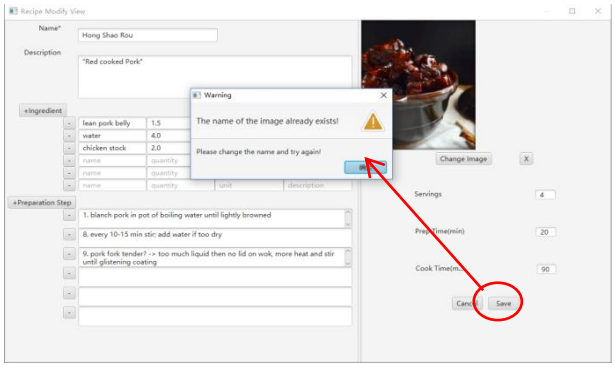
Deleting/adding ingredients and steps

The layout of recipe modify view is similar to the recipe view. However, on the left side, users can edit the content of the recipe by typing content into specific text field. “+ingredients” and “+preparation step” buttons are used to add a new ingredient and a preparation step. “-” buttons will delete an ingredient or a preparation step.



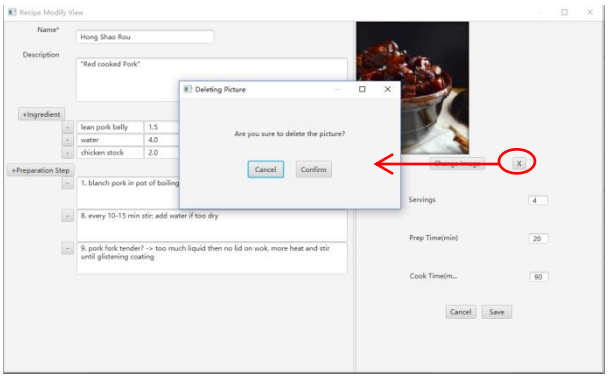
Dialog for selecting picture

On the right side, picture can be added or switched by clicking “change picture” button where a dialog will be generated to allow users to choose pictures.



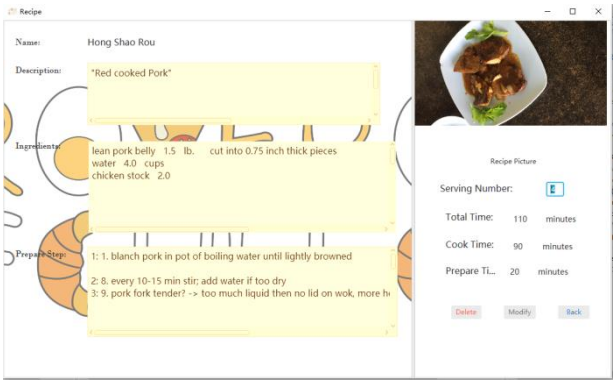
Reminder for same picture

If there is already a picture in the software with the same name as the picture chosen by the user, a reminder window will pop up after clicking “save” button to inform the user that a same picture has already existed.



Alert window for deleting picture

Also, the user can easily delete a picture by clicking the “x” button to delete a picture where a deleting alert window will pop up to warn the user.



Displaying modified picture

After editing the content of the recipe, users can click “save button” to save this recipe to database and a recipe view will be opened to show the recipe that has been edited just now.

4. Test

4.1. Description

1. Junit test is executed in model and database part.
2. Boundary test is executed in view and controller part.
3. Usability test is executed in whole project.

4.2. Results

4.2.1 Usability Test

4.2.1.1 Test Tasks

-Task 1: It is summer. You and your friends are on a diet, so you decide to make some salad for breakfast and the salad should be exactly for 3 people.
Aim: Test “Search field”, see if user select radio button “by name” to simple result, and test “Change quantity”.
Record: step of the user searching for “salad”; time of getting a right recipe; time of changing serving number. (User’s activity, user’s feedback)
-Task 2: A chocolate cake has too many calories. Since you are on a diet, you should delete this recipe.
Aim: Test “Search field” and delete function.
Record: time of deleting this recipe successfully. (User’s feedback, time consumption)
-Task 3: For lunch you want to have some low calorie meat, like chicken. Choose one recipe that contains chicken.
Aim: Test “Search field”, see if user select radio button “by ingredient” or “by both” to search faster.
Record: time of finding recipes contain “chicken”; ask which part do the user firstly want to look at. (User activity; user’s feedback, time consumption)
-Task 4: After reading it, you want to make some change. Maybe creating a more interesting name or adding/deleting some ingredient/step/picture would make this recipe better.
Aim: Test “Modify” and “Save” function.

Record: Degree of difficulty for the user to modify/save. (User's feedback, time consumption)

-Task 5: Tonight you are having a party in your apartment. During the party, one of your China buddy shares with you a secret recipe he learned from his grandma, and you want to record it. Find a way to do that. Also, remember you are still in the party. You may not have enough time to type in all details of this recipe. The details of this secret recipe are in the back of this paper.

Aim: Test add function and see which parts user thinks should be necessarily non-empty to save a new recipe in limited time.

Record: Parts that user have input; time of adding a new recipe; after that tell the user how long he/she spend on adding a new recipe, and ask how long does he/she think is appropriate for adding a new recipe. (User's activity; user's feedback, time consumption)

-Task 6: Congratulation! You have done all the tasks. Now you can do anything with this app and we will be grateful for your feedback! Thank you so much for your cooperation!

Aim: Get user's feedback/evaluation.

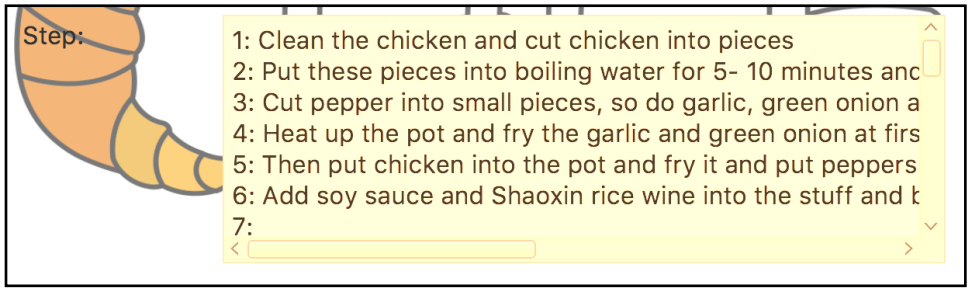
Record: user activity; window entry; user feedback or evaluation.

4.2.1.2 First round usability test

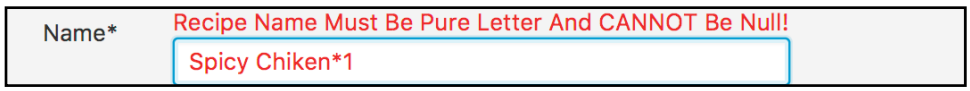
We test 5 people for first round usability test. The record and result are as follow.

Issues:

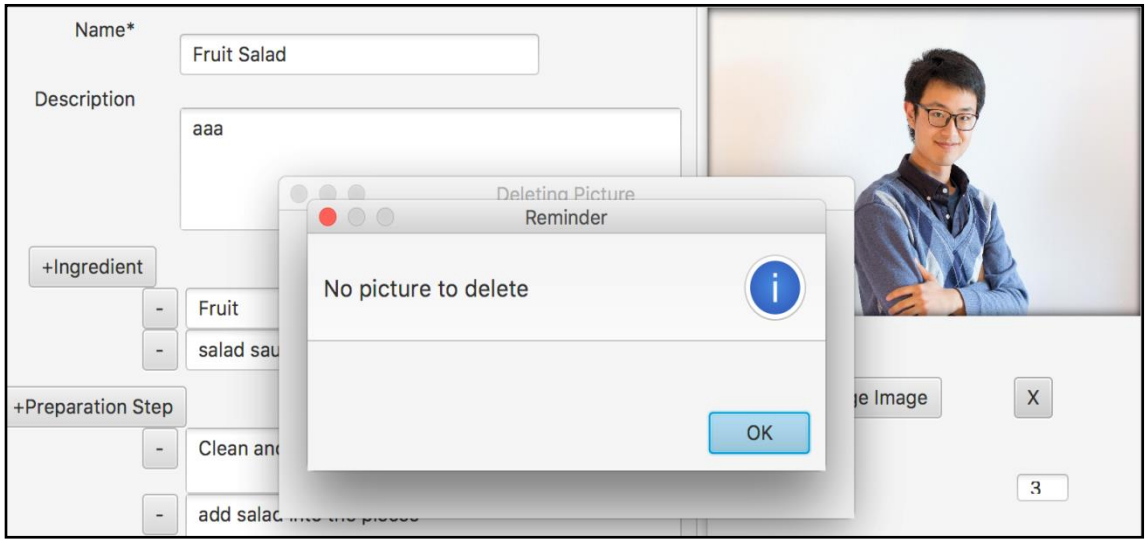
-Bug1: When the user adds a new step without enter any value and save it, this empty step will also be stored and displayed.



-Bug2: User can enter some symbols like '+' and '*' in recipe name's text field, but we assumed in recipe name's text field should only be able to enter pure letters.



-Bug3: User do not know this personal picture is default picture, so when user want to delete the picture there will be a reminder that says there is no picture to delete, which confuses the user.



-Bug4: If user deletes an ingredient, saves this recipe, back to recipe detail view and changes the serving number, then there is an outOfBound error.

-Bug5: One user adds a new recipe, enters an ingredient with a certain quantity but does not change serving number, whose default is 0. When user opens this recipe again and changes serving number, the quantity cannot change.

-One user spends significantly more time finding how to delete a picture, which is longer than we expected. The user is confused with ‘X’ button’s meaning and suggests change it to specific words ‘delete Image’.

-One user is confused with ‘modify’ button. The user suggests change ‘modify’ to ‘edit’ which is easier to understand.

-One user enters the quantity as ‘3/4’ which is not allowed in quantity’s text field. Then the user changes it to ‘0.75’. Two users enter ‘0.75’ directly without trying ‘3/4’.

-One user is wondering why the preparation time and cook time will not change after user changing serving number.

-One user suggests delete the space between quantity and unit in recipe detail window.

-One user suggests add delete button after each recipe in main view.

-User may miss click delete ingredient when trying to click add ingredient.

-In task6, users think recipe's name and its description are most important and other parts can be left to default value.

Questionnaire Result:

	Main window	**Recipe detail window**	**recipe modify window**
is the layout clear	★★★★★	★★★★★	★★★★★
are functions' names understandable	★★★★★	★★★★★	★★★★★

Improvement plan:

-Fix all the bugs found in usability test1.

-Change some buttons' texts: 'X' -> 'Delete Image', 'Modify' -> 'Edit'

-Delete the space between ingredient's quantity and unit.

-Change default picture to *OverChef* logo picture.

-Change the degree of transparency of background picture.

4.2.1.3 Second Round usability test

We test 5 people for second round usability test. The record and result are below.

-Two users suggest that it would be more convenient if the software can switch lines automatically if the content cannot be displayed within one line.

-One user write all steps in the text field of the first step and does not notice the add step button.

-One user suggests that the software should provide buttons allowing user to adjust the order of steps.

4.2.2 Equivalence Classes and Boundary Test

4.2.2.1 EC and BT for Main view

Task1: test for the text field of search recipe

Input: set of values (value must be element of recipe name or ingredient name)

EC:

Valid EC V1 = {value| value= hong shao rou} (recipe name)

Valid EC V2 = {value| value= salt} (ingredient name)

Valid EC V3 = {value| value=chicken} (both contain)

Invalid EC V4 = {value| value= 123}

Invalid EC V5 = {value| value=" " }

Test cases: hong shao rou, salt, chicken, 123," "

Testing Result:

Input V1: get "hong shao rou" recipe

Input V2: get "chicken soup" recipe

Input V3: get "Chicken soup", "Chicken with Onion" and "Spicy Chicken"

Input V4: nothing got

Input V5: nothing got

4.2.2.2 EC and BT for Recipe view

Task1: Test for the input text field of server number:

Input: range of values

EC:

Valid EC V1 = {value|0<=value}

Invalid EC V2 = {value| 0>=value}

Boundaries:

-10000 0	1 10000
Invalid	valid

Test cases: -10000, -6, 0, 125, 10000

Testing Result:

When input the cases of "1", "125" and "10000", there is not warning message. But when we input "-6", ''0" and "-10000", there is warning message.

Input: Formatted

EC: (value must be only positive Integer)

Invalid EC V1 = {value| value=123}

Invalid EC V2 = {value| value=a1b2c3}

Invalid EC V3 = {value| value=!+#}

Test cases: 123, a1b3c3, !+#

Testing Result:

When input the cases of “123”, there is not warning message. But when we input ‘’a1b2c3” and “!+#”, there is warning message.

4.2.2.3 EC and BT for Recipe modify view

Task1: Test for the text field of recipe name

Input: Formatted

EC: (value can letter, integer and some character (“&”, ” ’ ” , ” _”))

Valid EC V1 = {value| value=abc}

Valid EC V2 = {value| value=123}

Valid EC V3 = {value| value=& or ‘ or _}

Invalid EC V3 = {value| value=!#}

Test cases: abc, 123, &, ’ , _, !#

Testing Result:

V1: no warning message

V2: no warning message

V3: no warning message

V4: warning message

Task2: Test for the text field of recipe description

Input: Formatted

EC: (no limitation)

Valid EC V1 = {value| value=123didhidj}

Result: V1: no exception

Task3: Test for text field of ingredient name

Input: Formatted

EC: (value can letter, integer and some character (“&”, ” ’ ” , ” _”))

Valid EC V1 = {value| value=abc}

Valid EC V2 = {value| value=123}

Valid EC V3 = {value| value=& or ‘ or _}

Invalid EC V3 = {value| value=#}

Test cases: abc, 123,&, ’ ,_,#

Testing Result:

V1: no warning message

V2: no warning message

V3: no warning message

V4: warning message

Task4: Test for text field of ingredient quantity

Input: range of values

EC:

Valid EC V1 = {quantity|0<= quantity}

Invalid EC V2 = {quantity| quantity<0}

Boundaries:

-10000 -1	0 10000
invalid	valid

Test cases:-10000,-1, 0, 125, 10000

Result:

When input the cases of “1”, “122” and “10000”, there is not warning message. But when we input “-6”, ‘’0” and “-10000”, there is warning message.

Input: Formatted

EC: (value must be only positive Integer)

Valid EC V1 = {value| value=123}

Invalid EC V2 = {value| value=a1b2c3}

Invalid EC V3 = {value| value=*}

Test cases: 123, a1b3c3,*

Testing Result:

When input the cases of “123”, there is not warning message. But when we input “*” and ‘’a1b2c3”, there is warning message.

Task5: Test for the text field of ingredient unit

Input: Formatted

EC: (no limitation)

Valid EC V1 = {value| value=kg}

Testing Result: no warning message

Task6: Test for the text field of ingredient description

Input: Formatted

EC: (no limitation)

Valid EC V1 = {value| value = main flavor}

Testing Result: no warning message

Task7: Test for the text field of preparation step

Input: Formatted

EC: (no limitation)

Valid EC V1 = {value| value =add salt and rice}

Testing Result: no warning message

Task8: Test for inserting picture

Input: set of entries

EC: (the type of file should be element of {.jpg, .png})

Valid EC V1 = {value| value=cook1.jpg}

Valid EC V2 = {value| value=overchef.png}

Invalid EC V3 = {value| value= cool _sky.doc}

Testing Result:

V1: image can be shown

V2: image can be shown

V3: file cannot be shown

Task9: Test for the input text field of server number:

Input: range of values

EC:

Valid EC V1 = {value|0<=value}

Invalid EC V2 = {value| 0>=value}

Boundaries:

-10000 0	1 10000
invalid	valid

Test cases: -10000,-30,0,9,10000

Testing Result:

When input the cases of “1”, “9” and “10000”, there is not warning message. But when we input “-30”, “0” and “-10000”, there is warning message.

Input: Formatted

EC: (value must be only positive Integer)

Valid EC V1 = {value| value=167}

Invalid EC V2 = {value| value=1h2k}

Invalid EC V3 = {value| value=&}

Test cases: 167, 1h2k, &

Testing Result:

When input the cases of “123”, there is not warning message. But when we input”1h2k” and “&”, there is warning message.

Task10: Test for the input text field of prepare time:

Input: range of values

EC:

Valid EC V1 = {value|0<=value}

Invalid EC V2 = {value| 0>=value}

Boundaries:

-10000 0	1 10000
invalid	valid

Test cases:-10000,-30, 0, 1, 30, 10000

Testing Result:

When input the cases of “1”, “30” and “10000”, there is not warning message. But when we input “-30”, ‘’0” and “-10000”, there is warning message.

Input: Formatted

EC: (value must be only positive Integer)

Valid EC V1 = {value| value=50}

Invalid EC V2 = {value| value=a20}

Invalid EC V3 = {value| value=#}

Test cases: 123, a20

Testing Result:

When input the cases of “123”, there is not warning message. But when we input “a20” and “#”, there is warning message.

Task11: Test for the input text field of cook time:

Input: range of values

EC:

Valid EC V1 = {value|0<=value}

Invalid EC V2 = {value| 0>=value}

Boundaries:

-10000 0	1 10000
invalid	valid

Test cases:-10000,-30, 0, 1, 30, 10000

Testing Result:

When input the cases of “1”, “30” and “10000”, there is not warning message. But when we input “-30”, ‘’0” and “-10000”, there is warning message.

Input: Formatted

EC: (value must be only positive Integer)

Valid EC V1 = {value| value=50}

Invalid EC V2 = {value| value=a20}

Invalid EC V3 = {value| value=%}

Test cases: 123, a20, %

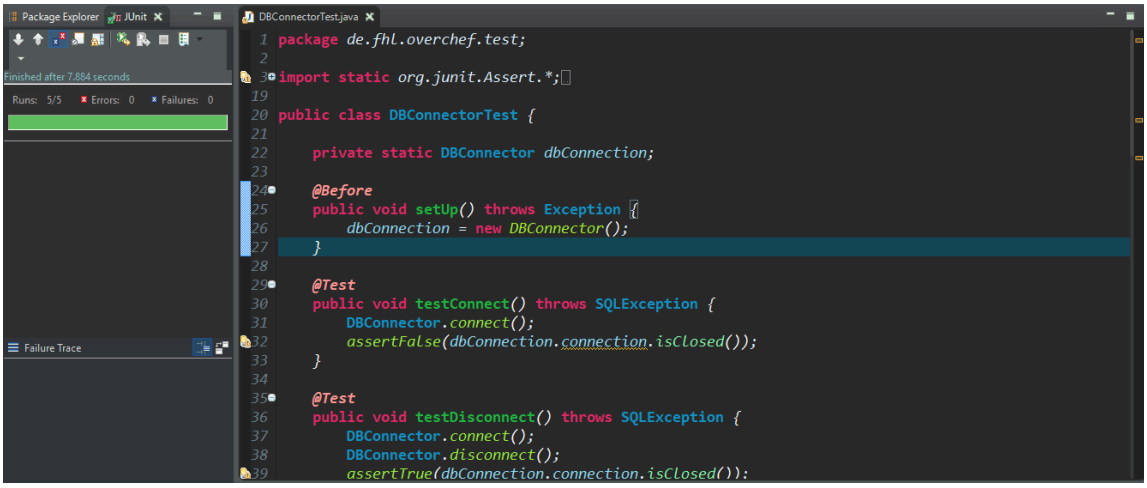
Testing Result:

When input the cases of “123”, there is not warning message. But when we input “a20” and “%”, there is warning message.

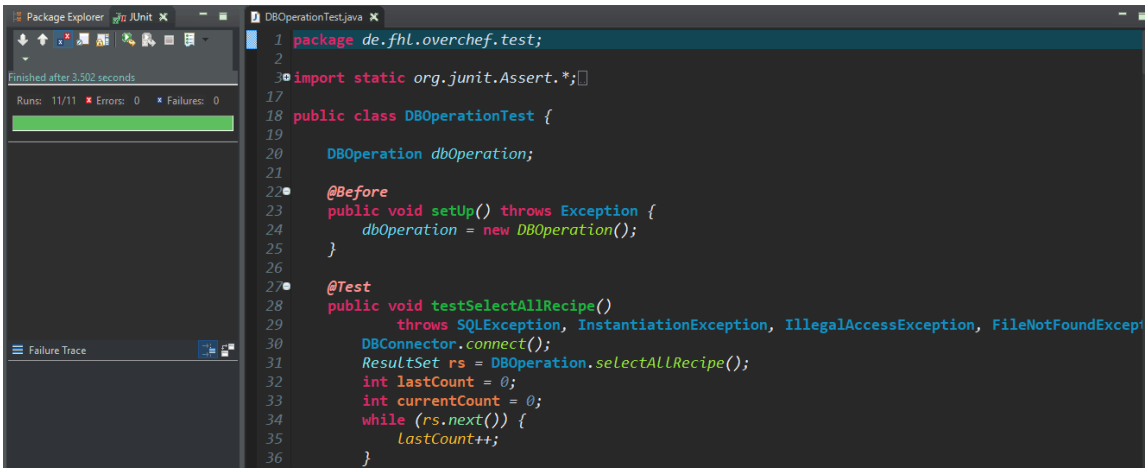
4.2.3 Junit Test

Description: The most difficult part is about testing of the database.

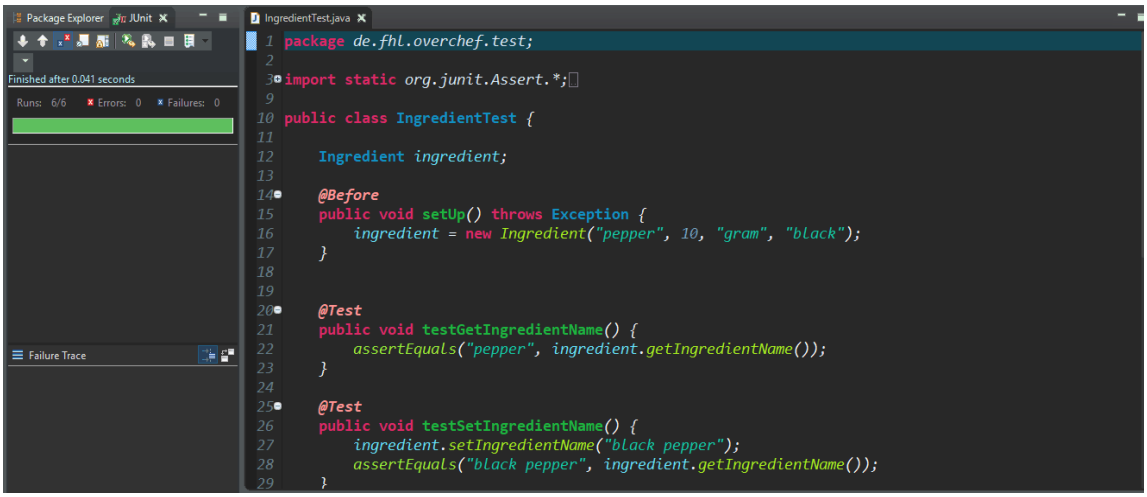
1. The JUnit tests of class DBConnector are all successful.



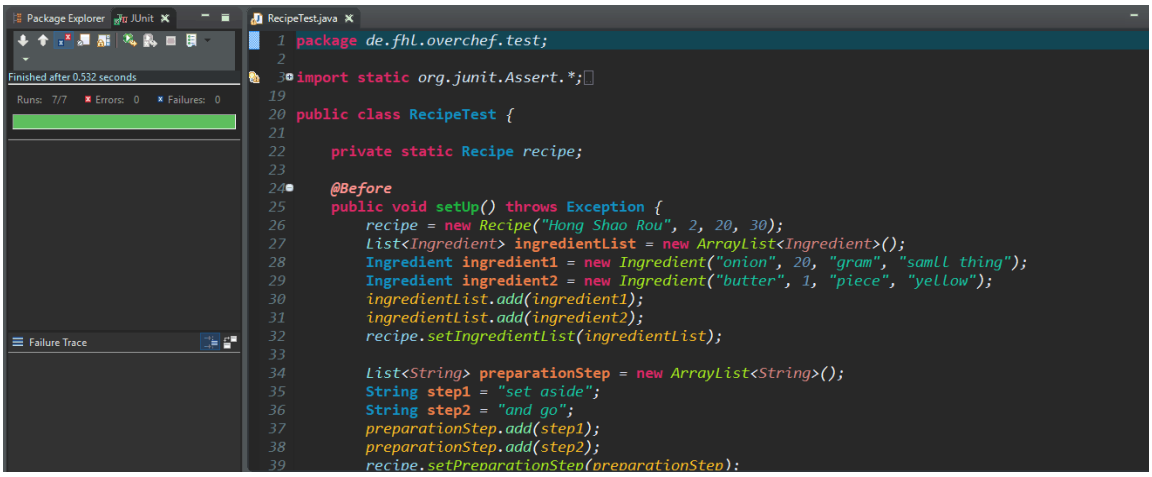
2. The JUnit tests of the class of DBOpration are all successful.



3. The JUnit tests of the class of Ingredient are all successful.



4. The JUnit tests of the class of Recipe are all successful.



5. The JUnit tests of the class of Cookbook are all successful.

6. The JUnit tests of the class of Cookbook are all successful.


Package Explorer
JUnit
Finished after 0.681 seconds
Runs: 7/7
Errors: 0
Failures: 0
Failure Trace
PictureTest.java
1 package de.fhl.overchef.test;
2
3 import static org.junit.Assert.*;
15
16 public class PictureTest {
17
18 private static Picture picture;
19
20 @Before
21 public void setUp() throws Exception {
22 picture = new Picture(System.getProperty("user.dir") + "\\src\\de\\fhl\\overchef\\view\\rec
23 }
24
25 @Test
26 public void testGetImage() {
27 assertTrue(picture.getImage() != null);
28 }
29
30 @Test
31 public void testSetImage() {
32 picture.setImage(picture.getImage());
33 assertTrue(picture.getImage() != null);
34 }

5. Evaluation

5.1. Group Work

Jiacheng Zhou:

In team part, I find it's difficult to code cooperatively during the project development. Our project tasks are distributed so separate that each of us take charge of some classes in the model part and then controller part and so on. But after I finish my on code without any bugs, then we combine the codes as a whole part. Many bugs appear. Usually, it takes long time to finish the combination.

In personal part, I have learned a lot from the project. I'm responsible for the recipe class, cookbook class, main view, main app and database part. Almost all my parts will be used by the other views or methods, especially the database. So I need to be familiar with others codes and give them solution to operate the database. And when they found bugs, I also need to help them to find whether there is something wrong with the database. Also, the main view part is the first view when users open the app, I need to initialize the data from database. It also means when user operates with the database, the initialized data should change synchronously. It's also a difficult task for me.

During the progress of coding, I find many problems. For example, when add a new recipe to the database, I need also add a recipe to the recipe list, so that the content of the list is the same as the data in the database. However, when I want to delete the recipe, the problem comes--how to delete the recipe when I don't know the recipe id in the database.(because the recipe id is automatically increased in the database) The final solution is to get the maximum recipe id in the database and set the recipe id equals (max rid +1), so the id is the same as the automatically generated id in the database. Another big problem is to show the user with the newest result recipe list when some delete modify or add operation implemented. The final solution is to define the recipeData list as static, so that when the other operation happens, it will be the same in all the other methods. There is also a problem, when I want to store picture directory into the database, because the "/" is a special character in the database, so when I store a normal directory as a string into the database, the "/" always missing in the database. The problem is solved by me and ZhengJiang Hu by just storing the name of the picture and then add the other directory when take it out of the database. Problems are so many ,like how to set relative position in view design, how to ensure there is only one database connection at the same time ,and so on. During the development of the project, I learned how to solve the programming problems more than just the solution of the specific problems itself.

Yujia Zhang:

In team part, I find it is hard to make sure everyone is on the same state, which means after we divided tasks to everyone and it is possible that someone finished earlier and someone missed the deadline. Luckily we managed that by discussing and reporting everyone’s work, and then we could help each other in a better and efficient way.

In personal part, I learned a lot through this project. First of all, my operation system is different than other team members, which caused some trouble at the beginning. For example, I cannot use mySQLworkbench to forward engineer. To solve that, I find another mysql server which can replace mySQLworkbench. With this application, MAMP, I can synchronise the database from other team member and connect it to eclipse. This is not work for Mac user but also someone like Zhengjiang Hu, whose operation system is windows but cannot use MyWorkBench to forward engineer. I will write

more problems I met in problem report part. All I want to say is that during this process, I did meet lots of problem and some of them even made me really frustrated and sad, but at the end of day, I managed all of them.

Some tips for next generation: The time is precious! It is better for you to design your data structure at the very beginning. Also, it is better to distribute task as model/view/controller and code according the data structure you would design so that you can start at the same time.

In the last, I want to say thank you to all the team members and our supervisor, for working with each other and managing all problems we met.

Bocheng Hu:

In the beginning of this project, we discuss the specification, UML specification and GUI together. We all express our thoughts together and combine them finally. In the implement of the software, I am responsible for the full part of recipe view except the additional task about the picture and some part of recipe-modify view. We use the MVC model, so when I realize the controller function of recipe view and recipe-modify view, I should write codes in the model. I make programming most in the class of recipe and ingredient. I spend a lot of time to beatify the GUI because I have to restart to learn the CSS. In the process of the testing, I am responsible for the testing part. I almost do full part of the EC and boundary test and do some part of usability test.

I have made a great progress and learn something in the Software Engineering lecture. First, I know how to develop a software and the full thinking about the specification and UML is necessary. What's more, the knowledge which what I have learnt make into practice and I have deeper comprehension about the java knowledge. In addition, now I know how to use some software, such as Astah, MySQL, JavaFX builder. And when I design the GUI, I have learnt something about the CSS style, which will be helpful for the network page designing. And I really come to realize that the testing is very important for developing the software and also master some basic testing skill for software. Finally, I think that it may be most important wealth that I know how to work together well with others. The cooperation is very important when we work in the society. I am really thankful for giving the chance to cooperate with us to make a software with all of you.

Fangtian Li:

I am glad to work with people who are brave to express their ideas and insist on them. This spirit ensures that the project was developed as everyone's thoughts rather than thoughts of one of us.

Zhengjiang Hu:

Firstly, this is my first time to develop a software project within a team, which has given me much unforgettable memory and precious experience on java programming and software developing. This project has taught me that software development is very different from programming. Certainly, programming is a necessary and unavoidable section in the development but it is not the most important. The most important part is to design the software such as making specification and UML. What's more, besides implementation, there are also many kinds of tests which are also crucial to the software. During the development of software, I have met so many problems such as code errors, logical errors and exceptions. I have learned to solve these problems through cooperating with my team members and doing research on my own. Now I have a much deeper understanding of software. Meanwhile, I have realized that software developing is a teamwork. It is impossible to image how a single person can finish the software. If a single person can do this, a team will definitely do the project much better more efficiently.

Secondly, as the leader of the 5-member team, I had to unite every member to cooperate which is kind of difficult at the beginning. I have the responsibility of organizing project meeting, team discussion and so on. When there are different opinions between team members, where even a heated debate happens, I try to respect, understand, combine and balance every opinion. And I also try to make a positive atmosphere and maintain a harmonic relationship in our team. From being a leader, I have learned so much that in some way it is even more than what I have learned in the implementation of the software. The beauty of the collisions of ideas impressed me deeply, which taught me to listen to others and understand them. I often worried my team about the progress, problems, cooperation and so on. However finally we made through all the difficulties which makes me so proud of my team.

I am really grateful for giving me such a chance to work within a team.

5.2. Task Responsibilities

Team Member	Implementation Responsibility	Testing Responsibility	Project Report Responsibility
Zhengjiang Hu	Model: ◆ Creation and methods' implementations of class "Picture"; View: ◆ Creation of basic GUI of Recipe Modify View (RecipeModifyView.fxml); ◆ Creation and methods' implementations of class "RecipeModifyView"; ◆ Creation and methods' implementations of classes----- "CancelAlert" ,"CloseAlert", "DeleteAlert"; Controller: ◆ Creation of class "RecipeModifyController"; ◆ implementations of methods of RecipeModifyController: changePicture(), deletePicture(), pressSave(), pressCancel(), savePicture(), handleSave(); ◆ Partly participating in implementation of methods: setModifyView(), initializeRecipeView(); Primarily responsible for additional task (picture including)	Junit Test: ◆ Picture (primary) ◆ The other classes of model (partial) Boundary Value Test: ◆ Recipe Modify View Usability Test: ◆ Executing usability test	GUI Design(Chapter3) Evaluation(Chapter5) Report Integration
Jiacheng Zhou	Model: ◆ Recipe.java: creation and design the data structure and methods of a recipe related to data storage in database(Group work with Yujia Zhang) ◆ Cookbook.java: creation and implementation of methods like loadRecipeFromDB, loadIngredientFromDB.... View: ◆ MainView.fxml: GUI design and implementation (Group work with Yujia Zhang) ◆ OverchefMainApp.java: Creation and Implementation of methods to open an app. ◆ MainViewRootLayout.fxml: Creation of this view. Controller: ◆ MainViewController.java: creation of methods of initializing the data from database, and set up the content of tableview, and methods of view shifting from recipe modify view and recipe view. (Group work with Yujia Zhang, she takes charge in search methods) Database:	JUnit test: ◆ Cookbook ◆ database Usability test: ◆ executing usability test Boundary Value test: ◆ MainView	Chapter 2: Use Case diagram E-R diagram Evaluation---Jiacheng Zhou part (Chapter 5) Readme.txt

	◆ EER model ◆ database connection(DBConnector.java) ◆ database operation(DBOperation, method of select data and insert data (main responsibility; group work with Yujia Zhang) ◆ participate in all methods involving the database(group work with Yujia Zhang) ◆ database export and import		
Yujia Zhang	Model: ◆ Create and implement all methods of class "Relationship" and class "Ingredient"; ◆ Work with JiachengZHOU on creating and implementing all methods of class "Cookbook" and class "Recipe"(pair coding); View: ◆ Create GUI of MainView (MainView.fxml and MainViewRootLayout.fxml); ◆ Create and implement all methods in MainViewRootLayout (display introduction in MainView menubar "about") ◆ Work with JiachengZHOU on creating and implementing all methods of class "OverChefMainApp"(pair coding); Controller: ◆ Work with JiachengZHOU on creating class "MainViewController"; ◆ Implement all methods of RootLayoutController: ◆ Implementl methods of MainViewController relevanceByIngredient(), relevanceByRecipe(), searchResultByRecipe(), searchResultByIngredient(), searchResult(), initialize(), showSearchResult(), addNewRecipe(), setMainApp() Database: ◆ Work with JiachengZHOU on creating and implementing all methods for database(pair coding); Responsible for Design all GUI in sketch;	Usability test: ◆ Design and write documentation for usability test; ◆ Execute usability test ◆ Integrate and analyse usability test	Specification (Chapter 1) Usability Test (Chapter 4) Evaluation---Yujia Zhang part (Chapter 5) Integrate and format report
Fangtian Li	Model: ◆ Create and implement part of the recipe class and ingredient class. View: ◆ Only partially adjustment in Recipe Modify View. ◆ Inherit the basic GUI from Mr.Hu Zhengjiang and then make a great adjustment to final version. Controller: ◆ Implement some methods in recipeModifyController, including checkRecipeName(), addIngredient(), addStep(), saveData(), setModifyView() and modifyPunctuation(); ◆ Partly implement method in recipeModifyController: handleSave(); ◆ Small adjustment in method initializeRecipeView() in recipeViewController; Other: ◆ Provide 1/3 of the recipes that database contains	Junit Test ◆ Mostly all the Model and DB test (except picture class test), get lots of help from Mr. Hu Zhengjiang and Mr. Zhou Jiacheng Usability Test ◆ Executing usability test	Class Diagram (Chapter 2) Evaluation---Fangtian Li part (Chapter 5)
Bocheng Hu	Model: ◆ Implement of recipe class(some basic set method, toGetIngredients(), toGetPreparationStep() , changeQuantity(int serveNum)), implement of the class ingredient (some basic method about setting and getting) View: ◆ recipe view except the picture, some part of recipe modify view, delete confirm view	Usability Test ◆ Executing usability test Boundary Value Test ◆ Recipe View ◆ Recipe Modify View	Test(Chapter4) ◆ boundary test ◆ Junit test

	controller: ◆ recipe view controller except about picture, recipe modify view controller(changeServeNumber(),checkCTFormatted(),checkPTFormatted()), delete confirm view controller		

5.3 Problem Report

Jiacheng Zhou:

Problem:

1. Store all the recipe data from database in cookbook, and then send the recipe date to view. So that Model, View and Database are isolated.

2. Set the recipeData list static so that all the methods can have the same data after some operations.

3. It is a must to set the recipe id before storing in the data base. Because if user want to modify a newly-added recipe, the certain recipe can't be find without recipe id (because the recipe name is not unique)

4. Use singleton pattern to ensure only 1 connection is existed.

5. Only store the picture name and add the directory to the picture name after reading from the database (because the "/" is escape character).

6. Ensure the RECIPE table without foreign-key by putting recipe id as foreign key into other tables, to avoid the "cannot add or update the child row " error.

7. Add a certain test recipe to the database and then delete them to make the JUnit test in database can pass in any computer.

8. Double click event is not an option in scene builder. I choose the mouse click option and add a controller to it and then write a double-click mouse event to detect when there is double click ,the recipe view will be opened.

Yujia Zhang:

Problem:

1. Cannot use mySQLworkbench to forward engineer. I solved this problem with using another mySQL server called MAMP to use the database.
2. Search recipe method. I firstly wrote a fully-matched searching method, which can only find a recipe when user enter the full and correct name of a recipe. This searching method is user unfriendly. Then I was thinking how about if I can split user's entering, then I can compare each of the word to the recipe and return all relevant recipes. The same is to ingredient. Based on this design, I write two method to compare recipe's and ingredient's name with every single words that user enters. The problem is about which data type to choose to store the returned recipes so that it can be displayed in table view in javaFX.
3. Design the data structure. At the very beginning I was thinking about using s map to store recipes and ingredients, so that when user enters an ingredient name, it can return all recipes that contains this ingredient. But later we found if we store data in this map<ingredient, relationship>(relationship stores quantity/unit/description of ingredient), it is hard to do data type conversion. In this case, we chose to ingredients' information apart from recipe, but used recipe id to link them.
4. Data store. We chose to store data in memory and in database at the same time. The advantage is that because of initialisation process can read all data from database into memory, the speed of searching and loading recipe is faster than read from database every time. Because of the quantity of recipes is not too large, storing recipes' data into memory can working.

5. Export the project as a runnable jar file. I exported the project as a .jar file, and thenI could only open the main view but cannot enter the recipe view. I ran this file in terminal and then found out the problem is that the program cannot find all picture files. This is caused by the path in picture class which is not relative path. If I put this .jar under the project file, then it can read all picture files. We did not find this problem earlier is that we always use eclipse to run the program. In this case, the path is though to be relative path because all pictures are stored under the source file. Unfortunately, it will take quite a lot time to change the structure. If we want to run a .jar file in somewhere except the project file, we need to build a resource folder and store the picture file under it. I will look into it when i have time, but at this moment our main app can only either run in eclipse or we open the .jar file under project file.

Fangtian Li:

1. I have thought about how to update all the recipe information. Here are two solutions. The first one is, I always connect to database and everything I do on the software, or do it on database. The disadvantage is that the speed is much slow. The second solution is, I get all the information from database and store it in a list when I run this software. The advantage is that this software cannot contain thousands of recipes. Eventually I choose second solution because I of the small amount of recipes(less than 10000)

2. I need to access the values in the list and I search for some informations on Wikipedia, and I found that the Arraylist is the most suitable for rapidly random access.

3. In three iterate methods, I choose the most efficient method: random access iterator.

4. I found that the software cannot distinguish the recipe name or ingredient name when users type something in a text field, so I determined to change the research method from searchByRecipeName() and searchByIngredientName() to searchByKeyword().

5. When I try some error hint, I firstly try to change the color of the text field frame to red, but it looks Iird. Therefore, I altered to change the font to red.

6. When I test the functionality of the error hint, I found the function reaction will have a delay. For example, if the recipe's name is only allowed to use letter, when user types "a1", the error hint will not activate. However, if user type "a11", the error hint will activate. After my multiple test, I located the error to the activate condition onKeyTyped(). This method acts one-step slower than user's type. Eventually, I choose to use addListener().

7. Because of the shortage of knowledge of scene builder, I don't know how to bind a button and a method with parameter. Therefore, I gave up the way that put buttons and steps into different list. Then I choose to bind a button and a step, and put them into an HBox, finally I add this HBox into a list.

8. We found that it is not reasonable to limit recipe name in pure letters, therefore I change it and make it accept number, letter, blanks, _,&,'

9. When I typed some double number, the interface report an error, remind me that the dot in double number didn't match the expression.

10. The ingredient quantity can be empty. However, when we save data, the empty string cannot transfer to a double number. The solution is that the quantity is always assigned to 0.

11. In expression, .represent a random character. This mistake takes me long time to discover.

12. I can simply set setWrapText(Boolean) to make sure that text area can automatically change its line.

13. Add functionality: delete an ingredient if its name is empty.

14. Feel so tired to write Junit Test after we complete our code.

15. I found that local variables in an inner class must be a final type.

16. I cannot insert ‘ into DB, because ‘ has special meaning in database. Solution: write a method called modifyPunctuation() to change every ‘ to ‘’.

Zhengjiang Hu:

NullPointerException on writing and reading image:

1. Images can not be read and written normally.
   The reason is leaving the FileInputStream and FileOutputStream not closed.

   Solution: close the streams.

Problem on the compatibility of swing and javafx.

1. JavaFX amd Java.Swing are not compatible. Buffered Image was used to read the image into the software but it’s compatible to java.swing and not compatible to javafx. Neither do JLabel which was used to display picture and ImageIO which was used to read and write image. Therefore, I rewrote the picture class to make it suitable for javafx, such as changing buffered image into image, using file input and output streams instead of ImageIO.

Problem on changing picture:

1. Including picture has already been implemented. Every time the “change Image” button is clicked, the picture chosen by user will be saved. And alert will be generated if user select a picture which has a same name as another picture that the software has already saved in the directory. But if the user firstly saves a picture and then change it into another on, then he change it back again, the alert will be generated since the picture has already been saved.

   Solution, separate the name checking function from the writing function.

2. Picture will now be displayed without writing into file at the same time. The picture will only be checked and saved when the save button is clicked, which means user can now switch picture without any limit. But if user set a new picture, the old picture still remain in the directory, which has no use and gives a risk of crashing other picture which perhaps has a same name. Therefore once user has changed the picture, the old picture should be deleted.

   Solution: check if the list of picture is empty. If it’s not, delete the picture first and then write the new picture. During the implementation, exception such as index out of bound had been met when using lastElement() method. This problem was solved by changing the method into get(0) (get the first element, since we only save one picture per recipe). But this is not a very good solution, and risk still remains.

Problem on relative directory:

1. “.\\” and “~\\” do not work. I use System.getProperty(“user.dir”) to get the current directory and combine it with \\src\\de\\fhl\\overchef\\model\\Pictures\\ to get the relative directory. However, a problem has occurred after my code was run on the computer of one of my group members whose computer’s operating system is mac. It encountered a nullpointerexception. I couldn’t find the reason but it works on others’ computers whose computers’ operating system are windows. I guess it’s because the difference of operating system. Finally the code worked on mac by changing ”\” into “/”.

Problem on saving

1. If the user selects a picture (press “change Image” button) and save it, there is no problem. But if a user never select a picture and he wants to directly save the recipe, then a nullPointerException shows up. Because if user doesn’t select a picture, the software cannot get a valid path to write a non-exist picture to the given directory. This problem makes user have to select a picture(otherwise program will have error), which definitely should not occur.

   Solution: give the program a default path linking to a default picture saved in the directory in advance. Further, changing the checkName() method to not only compare if the picture’s name is exist in the directory but also compare it with the default picture which also should not be the same.

Problem on displaying picture

1. Picture can be selected, saved and deleted. However it cannot be displayed after it is saved in the given directory. Finally the reason for this problem is that I used fileInputStream and FileOutputStream to read and write images. Reading is working but writing is making Image into binary files and no longer a pixel file anymore. That's why the program cannot display these files. Because they are not images any more.

   Solution: use SwingFXUtils.fromFXImage(image, null) to generate a image of BufferedImage which makes javafx images back into javax.swing images. Then the ImageIO can be applied to process images.

Problem on deleting picture

1. Saving pictures successfully as real image files solve the problems on displaying images. However it brings new problems to deleting. The delete() method worked in early version because the "picture files" were not images actually which means the program didn't use these files. Now these saved images are real image files and they are used in the program. And deleting a file will of course fail if the file is being used.
   Make an explicit declaration of FileInputStream and close it explicitly after it is used. By doing in this way, deleting method almost works. However program still cannot delete the picture that is first loaded into the recipe. The picture will be deleted without problems once the recipe is modified, or more specifically, the picture of the recipe is changed. The reason for this new problem hasn't been found yet.

   This problem has been solved by explicitly closing the FileInputStream used in the constructor of picture class.

NullPointerException on closing RecipeView

1. The stage of RecipeView has been passed into the RecipeModifyController and then into the SaveAlert. When the program attempts to close the stage in the SaveAlert class, sometimes a nullPointerException will occur but sometimes it will not occur and RecipeView will be closed successfully. Reason for this problem hasn't been found.

Exception on some problems

1. Exception on user selecting files

If a user chooses a picture and then confirms, the path will be got and can be then used to deal with all kinds of file handling issues. However, if the user clicks on the "change image" button and opens the file choosing dialog and then cancel this dialog, program cannot got a legal path and then IO problems and exceptions will occur.

Solution: To prevent this problem, give path(an attribute) which will be used later to receive the path selected by user a default value "default" and surround the code responsible for receiving path selected by users with try/catch blocks. Once the user doesn't select the file (still has opened the file choosing dialog), exception will be caught and the default will be held which makes program have no NullPointerException.

2. Exception on deleting picture
   2.1 If a recipe doesn't contain a picture. A user wants to modify this recipe and then click the button of deleting picture before any picture is included in this recipe, a NullPointerException will occur because the program will not know which picture it will delete. It cannot delete a picture that doesn't even exist. What's more, sometimes the default picture which is used to prevent exception will be deleted by mistake.

   Solution: Ask the program to check if the file to be deleted exists and is not the default picture. If the file can be deleted then the program will deleted it, if not, no execution.

   2.2 If a user deletes a picture and doesn't want to add any other picture to the recipe. After the user clicks the save button and confirms, an exception will occur because the value of the root of the picture deleted was still the path of the picture where it was before being

deleted. And after save button is clicked a recipe view will be generated to show the recipe which has been just modified and this view will use the root of the picture to load the picture. Then exception will occur because the picture deleted doesn't exist anymore

Solution: Let the program change the root of the picture deleted into the default path linking to a default picture stored in the directory. Once the picture is deleted, the root will no longer be linked to the picture but be linked to a default picture. Therefore, a default will show up on the recipe view after the user save the recipe. And the default picture can indicate that this recipe doesn't include a specific picture now.

Problem on existing file

1. If user chooses a picture and the name of the picture has already existed, after closing the alert window indicating same images, there will be no problem if the user change a picture. But if user doesn't change the picture and just want to give up adding a picture, he will still be informed of same image existing and cannot save the recipe. The reason is that user has already selected a picture which means a legal path is generated. Although the program will check if the image exists, the path is still legal. But since user doesn't select another picture, this path will hold there causing user fail to save the recipe until he cancel the view or select another picture.

   Solution: if user press delete picture button, execute delete and then set the path of chosen picture back to "default". Then recipe will be saved even if user doesn't want to choose another picture.

Bocheng Hu：

1. NullPointerException on the recipe view controller
   Reason: did not create object for the all javafx button and field, then I change the function which I just invoke the controller by using the javafx statement.

2. Having some problem about the layout:
   At the beginning, I just use the static layout and the window is so ugly, and the content of field become longer, all content is not shown in the window. So finally, I decide to use the dynamic layout. Luckily, the window is nicer and all of content can be shown in the recipe view.

3. When I delete the recipe in the recipe view, the information is not shown immediately in the main view. So I look up on the internet, I should use the static variables recipeData, the problem is solved.

4. When I change the serve number, there is a problem that the ingredient quantity do not change immediately. So I try to solve this problem through rewrite the controller about the field of serving number. And then the bit of ingredient quantity is too many, so I rewrite the method of changServeNumber, the ingredient quantity is limited by two bits.

5. I have made a server error in making programming which is about the using of the try-catch statement.

6. At first, I do not use prompting message when I input letter, character or negative integer in the field of serving number. So finally, I add a field for prompting message.

7. There is some problem about background, the font is not seen clearly because of the color of the background picture. So finally i change the transparency of the picture. It is a simple way to solve the problem.

◆ Appendix1 Usability Test for Users

Thank you very much for your participation. Your assistance will help us develop software that is friendlier to use. Notice: all the test results would be **anonymous**.

OverChef is a digital cooking software, providing functionalities such as quickly off-line searching, updating recipes based on users' preference. You may search for a certain recipe with recipe's name or ingredient's name. You can also modify or even delete any recipe. Now we have some simple tasks for you to try with *OverChef*. Please do the following instructions.

Task 1: It is summer. You and your friends are on a diet, so you decide to make some salad for breakfast and the salad should be exactly for 3 people.
Task 2: A chocolate cake has too many calories. Since you are on a diet, you should delete this recipe.
Task 3: For lunch you want to have some low calorie meat, like chicken. Choose one recipe that contains chicken.
Task 4: After reading it, you want to make some change. Maybe creating a more interesting name or adding/deleting some ingredient/step/picture would make this recipe better.
Task 5: Tonight you are having a party in your apartment. During the party, one of your China buddy shares with you a secret recipe he learned from his grandma, and you want to record it. Find a way to do that. Also, remember you are still in the party. You may not have enough time to type in all details of this recipe. The details of this secret recipe are in the back of this paper.
Task 6: Congratulation! You have done all the tasks. Now you can do anything you want with this app and we will be grateful for your feedback! Thank you so much for your cooperation!

Recipe name: Hong Shao Rou

Description:
Shanghai-Style Braised Pork Belly (Hong Shao Rou), or "red cooked pork," is a very famous dish in China. Everyone knows it, and there are many versions and twists based on the original. Some of the more well-known variations include the addition of squid (sounds odd, but boy, is it tasty), hard boiled eggs, and tofu knots.

Ingredient: 3 /4 lb. of lean pork belly (cut into 3/4-inch thick pieces)
2 tablespoons oil
1 tablespoon sugar (rock sugar is preferred if you have it)
1/2 tablespoon dark soy sauce
1 tablespoon light soy sauce

3 tablespoons shaoxing wine

2 cups water

Serving Number: 2

Step:

1. ”Bring a pot of water to a boil and blanch the pork for a couple minutes.
2. ”Take the pork out of the pot and set aside."
3. "Over low heat, add oil and sugar to your wok."
4. "Melt the sugar slightly and add the pork."
5. "Raise the heat to medium and cook until the pork is lightly browned."
6. "Turn the heat back down to low and add cooking wine, light soy sauce, dark soy sauce, and chicken stock."
7. "Cover and simmer for about 60 minutes to 90 minutes until pork is fork tender."
8. "Every 5-10 minutes, stir to prevent burning and add water if it gets too dry."
9. "Once the pork is fork tender, if there is still a lot of visible liquid, uncover the wok, turn up the heat, and stir continuously the sauce has reduced to a glistening coating."

Picture: (on desktop)

◆ Appendix2 Questionnaire

Thank you for your assistance. This questionnaire would be anonymous.

1. Name the task which you had difficulty to finish, if there was one.
2. Rate each window from following perspective: (full stars means really good)

	Main window	Recipe detail window	recipe modify window
is the layout clear	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆
is functions's name understandable	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆

3. Is there any problem or confusing design you run into while doing tasks ?

4. Any suggestion for the app?