FACH
HOCHSCHULE
LÜBECK

University of Applied Sciences

Software Engineering II Project

Digital Cookbook

Part task: Recommendation

Group 8

Dong Tianyuan

Sun Xiaoan

Yang Yiming

Shi FangBO

Content

1. Specification

1.1. Description

1.1.1. Digital cookbook

Nowadays, some people are annoyed with what for meals. Even if they have ideas about it, they may not know how to cook it. At the same time, some people who love cooking may want to have a platform to share their styled dishes. So Digital Cookbook is offered so that they can learn others‘ recipes or share their own recipes. It provides people with new ideas for delicious meals, presents people easy recipes how to create dishes for their desired amount of people step by step.

1.1.2. Additional Task

The Digital Cookbook should have the function of register and login. Also there will be recommendations for the users based on their last three searching histories.

1.2. Product functions

1.2.1. Register and Login

This function realizes register and login so that users can have their own accounts and store their personal information (such as user's ID, password and search history). This information can provide recommendation function with basic data (The search history).

1.2.2. Recipe Management

This function entitles users to add, edit and delete their own recipes (ingredients, preparation steps and so on).

1.2.3. Search and Display

This function entitles users to search for the recipe according to several key words. The recipe display includes the name of the recipe, prep time, serve number, cook time, ingredients and instructions. User can also choose the number of diner and the App will display different ingredients amount information depended on this number.

1.2.4. Recommendation

This function recommends users with recipes based on their searching history.

1.3. User characteristics

1.3.1. People who are not good at cooking. (Most of them can be the young people who have just been independent from their parents or can be the International students who are far away from their families)

1.3.2. People who love cooking and are also good at cooking. They are willing to share their recipe with others.

1.3.3. People who are tired of the daily dishes and want to find something new, to get inspiration of cooking.

1.4. Functional requirements

1.4.1. Login interface

There's a login interface which includes 2 blankets which are used to fill in password and username. If the password and the username is matched, the interface will jump to main interface. If the password or username is wrong, there will be a notification writing "Wrong user name or password". And there will be a "login" button to check whether the information is matched.

1.4.2. Register interface

There's a Register button on the login interface linking to an independent interface, which will ask the user to fill in a form when the button is clicked. The form includes necessary information such as username and password. The password will be filled in twice in order to ensure correctness. When the users have finished this step, they can choose to cancel or submit it by clicking the "cancel" or "submit" button. If they submit it, their information will be sent and saved in the database.

1.4.3. Main interface

The main interface will display the basic information about users (such as avatars). And it will include three links with three buttons. One links to Recipe Management Interface, one links to Personal Information Management interface, and the rest one links to Searching interface.

1.4.4. Recipe Management interface

This management has its own interface. Users can add, edit and delete their own recipes (ingredients, preparation steps and so on) here. The interface will display all of the user's existed recipes. And every recipe is followed by an "edit" and a "delete" button. The edit button links to corresponding Recipe Edition interface. The delete button will delete this recipe record from database. A notification will come out to confirm when users click the delete button. At the bottom of the interface, there is an "add" button which links to a new blank Recipe Edition interface. And there will be a "back" button to jump back to the main interface.

1.4.5. Recipe Edition interface

This interface will display the detail of a existed recipe such as recipe name, taste type, prepare time, cook time, serve number, ingredient list and preparation step. All of these details are editable (except ingredients, which have its own edition interface). There are five buttons inside the interface. The "OK" button is used to update the recipe and close this interface. The "Cancel" button is used to close the recipe without update this recipe(except ingredient, which will be update when you click the "Ok" button in the ingredient edition interface). "Add new ingredient" button is used to add new ingredient into the recipe. "Edit this ingredient" button is used to edit the details of an existed ingredient. "Delete this recipe" button is used to delete selected recipe. A notification will come out to confirm when users click the delete button. If this interface is called by "add" button in the Recipe Management Interface, it will display a new blank form.

1.4.5. Ingredient Edition interface

This interface will display the details of an existed ingredient including the name, quantity, unit and description of the ingredient. All of these details is editable. There are two button inside the interface. “OK” button will update this ingredient and close this interface. “Cancel” button will close this interface without update this ingredient.

1.4.7. Searching interface

The Searching interface has an input box where users can fill in the key words of the recipe they want to search. There is a “Search” button on the right side. User can start their search by clicking the button and the correspondent results will come out. Then Users can link to the correspondent recipe display interface by clicking on the pictures or the recipe name beside the pictures.

Under the input box, there is a column for recommended recipes with pictures. Users can link to the correspondent recipe display interface directly by clicking on the recipe name.

1.4.8. Recipe Display interface

In the Recipe Display interface, users can find the specific information about the recipe. The information includes the title (the name of the recipe), prep time, serve number, cook time, ingredients, instructions and the picture of this recipe. All of this information is unchangeable except the server number. If the user changes the server number. The amount of each ingredients will change according to the server number. There’s a “back” button inside the interface, which is used to close this interface.

1.4.7. Personal Information Management interface

In the Personal Information Management interface, users can change the password by correctly filling in the former password and the new password. User can also logout by click the “logout” button.

1.5. Non-functional requirements

1.5.1. Practicality

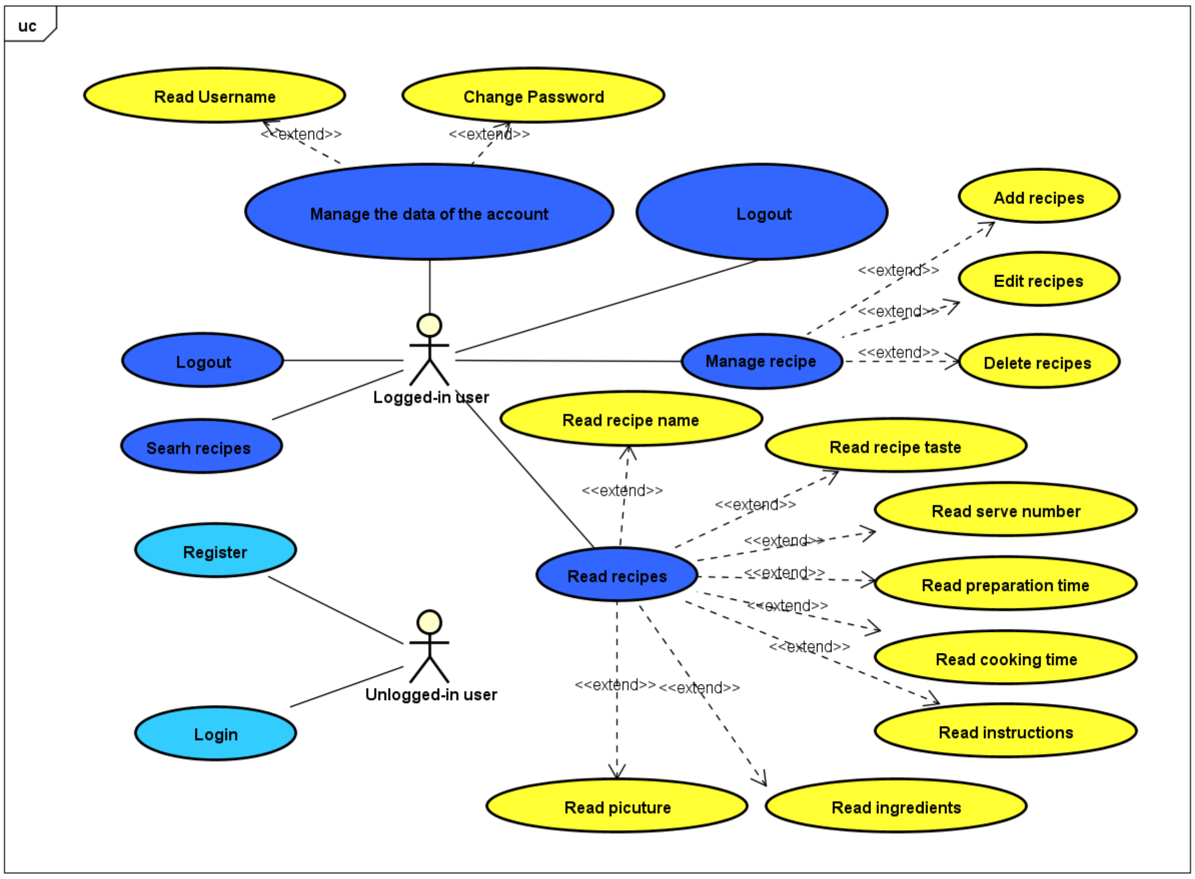
Ensure that the users will receive expected responses based on their requirement.

1.5.2. Convenience

Ensure that the user can use this application comfortably and without complicated learning. Every button’s function is obvious for users.

2. UML Specification

2.1. Use Cases



As shown in the Use Case Diagram above, there are two types of actor.

For Unlogged-in user, they can do two kinds of operations.

1. The user can login his/her account in the login interface.
2. The user can register a new account in the registry interface.

For Logged-in user, they can do following operations.

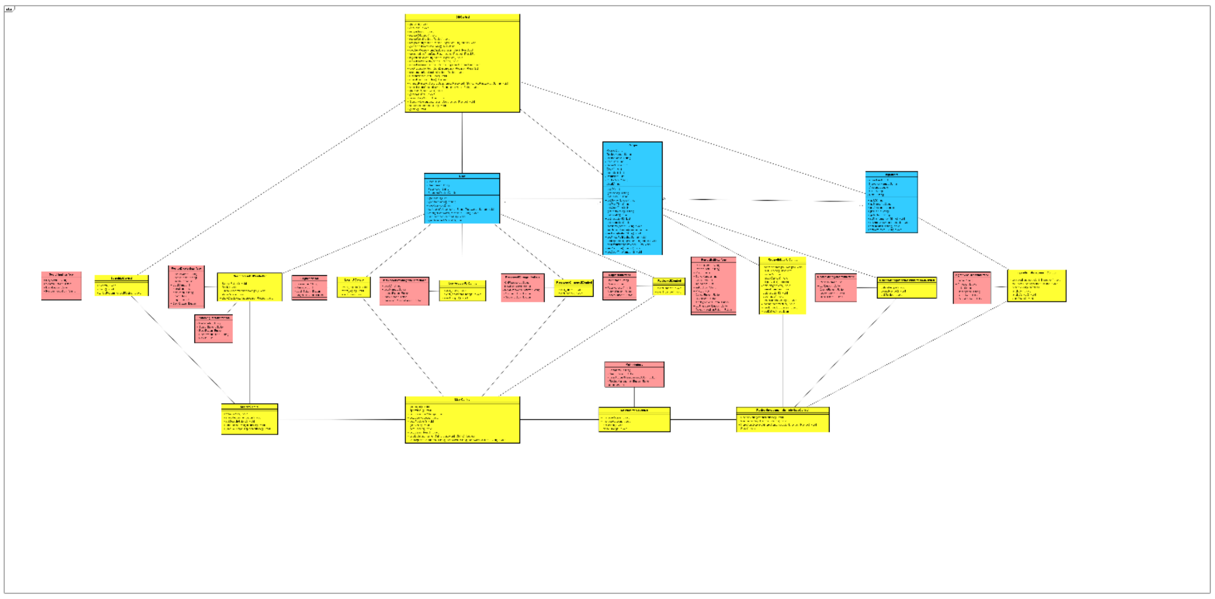
1. Logout at account management interface.
2. Search recipes by key words in recipe name in searching interface.
3. Manage the data of the account in account management interface. In details, they can
   a) read their user name
   b) change their password
4. Manage their recipe in recipe management interface, which allows them to
   a) Add a new recipe.
   b) Edit an existed recipe belongs to them.
   c) delete an existed recipe belongs to them.

5. Read a recipe in the search result interface. Which inform them

   a) Recipe name
   b) Recipe taste

c) Default server number
d) Time for preparation
e) Time for cooking
f) Detailed steps
g) Details of each ingredients used
h) Image of the recipe

2.2. Class Diagrams



(Bigger one is involved in the archive)

As shown in the Use Case diagram above, the software uses the MVC model and it is separated into three main parts: The Controller, the Model and the View.

There are 3 kinds of controller. MainControl, MainInterfaceControl and SearchControl are responsible for jumping between each interface. DBControl is responsible for operating the database. And the rest controllers are respectively responsible for realizing functions of related one or two interfaces.

As can be seen in the diagram. The DBControl communicate with other controllers by transfer a complete Object such as User, Recipe and Ingredient. This method makes the whole structure clear and concise, and is easy for DBControl to operate the database and get and send message from database to each controller.

3. GUI Design

3.1. Structure

In our digital cookbook, MVC model is used. This way we can change our view without changing the structure of our database. And vice versa, changing the structure of database would not influence the view.

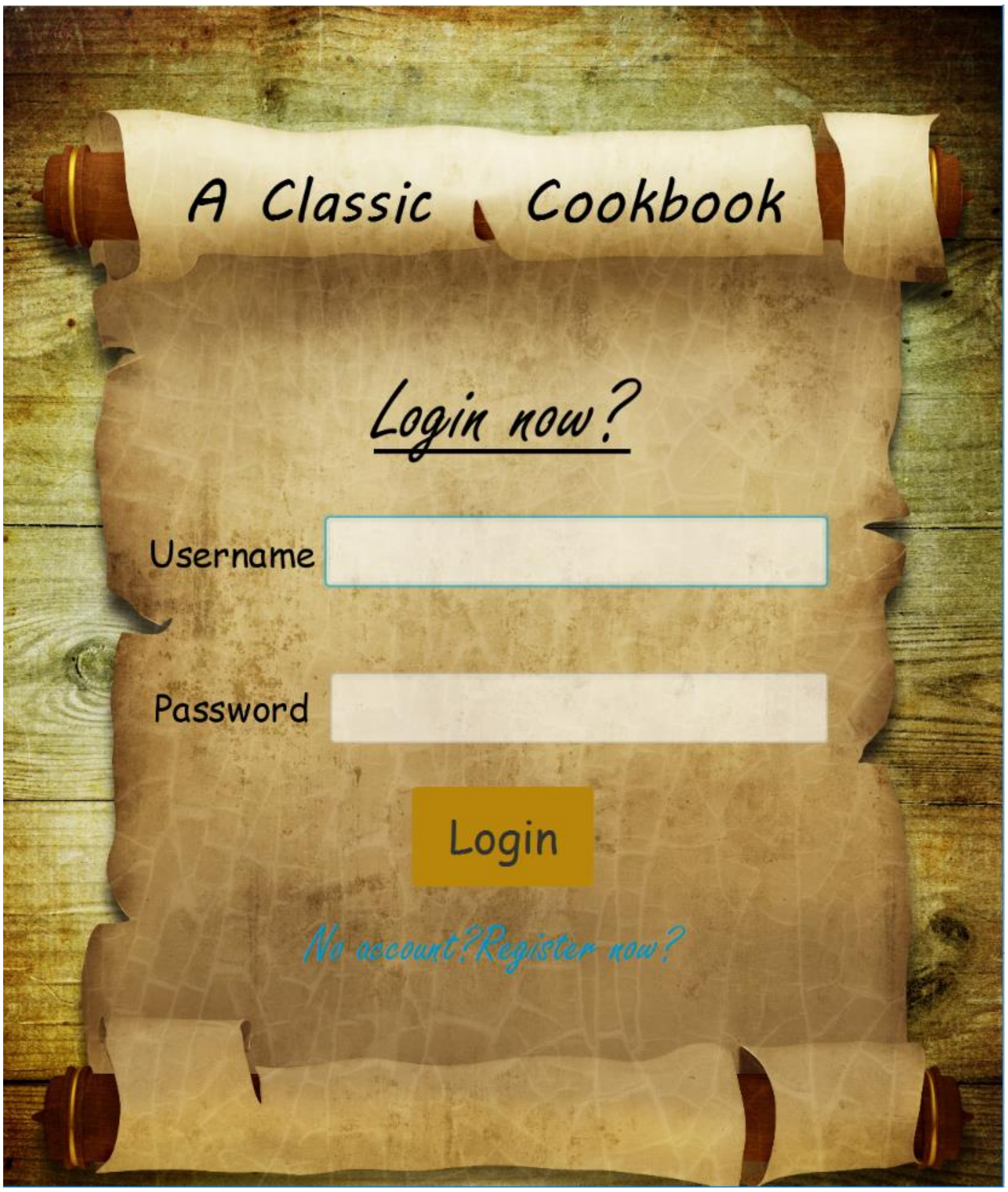
As for view, we have four 13 fxml files which are IngredientManagementUI.fxml, IngredientManagementUI_forNext.fxml, LoginUI.fxml, MainUI.fxml, PasswordChangeUI.fxml, RecipeDisplayUI.fxml, RecipeDisplayUI(for search) .fxml, RecipeManagementUI.fxml, RegisterUI.fxml, RootLayout.fxml, SearchResultUI.fxml, SearchUI.fxml, UserAccountUI.fxml. Each fxml file except for RootLayout, has a correspondent java file to put in all the components in the view. The java files can be easily changed, when the fxml files change regardless of changing the structure of database.

As for controller, normally for each view we bind a controller with it. These controllers are used to implements interactive accidents, and do something related to database. Among them, RecipeDisplayUIController.java and IngredientManagementUI.java are responsible for two fxml files because both fxml files share the same components with the only difference on the appearance. Besides these controllers. We have Main.java and RecipeManagement-MainInterface.java. java to mainly take responsible for the changing of the interfaces.

As for model, we have five classes: CookBook.java, CookBookApp.java, Ingredient.java, User.java and Recipe.java. Among them, Ingredient.java, User.java and Recipe.java are for the project. Almost all the data transmission are based on these three objects.

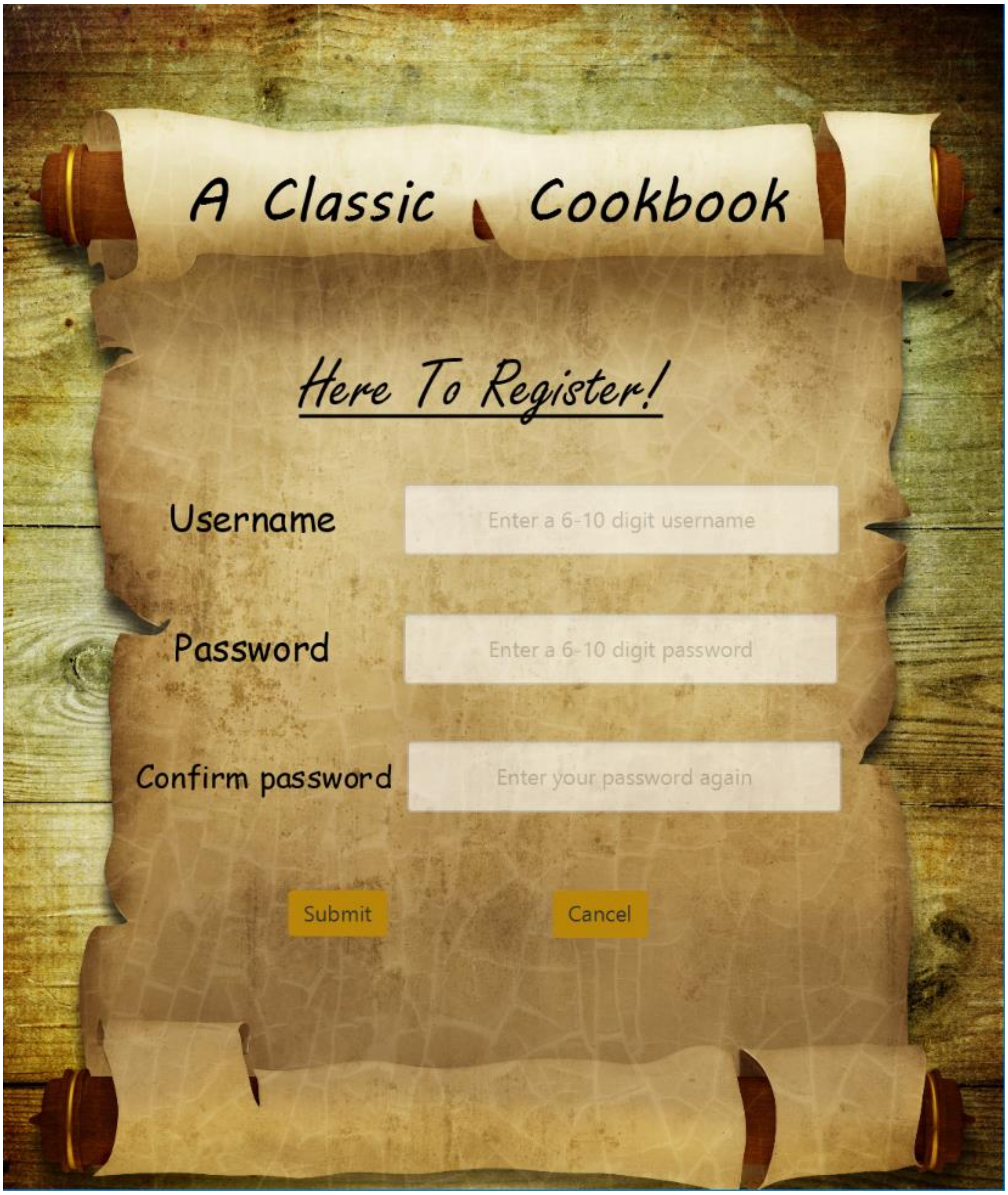
3.2. Screenshots

3.1.1.Login page



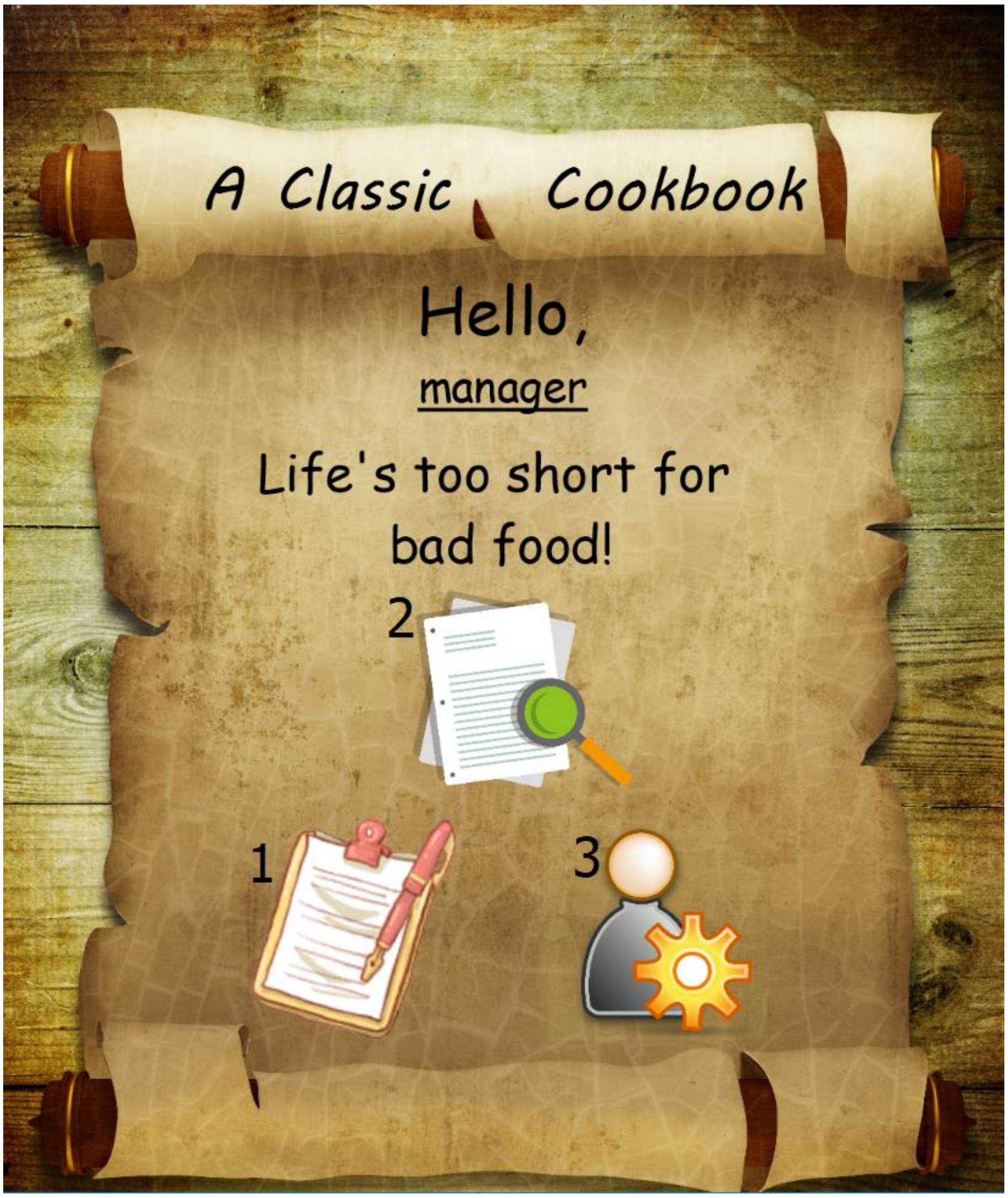
The login page consists of two text fields, one register button, one login button and a tip used to remind user to login. The login button and two fields are used for user to login. Two text fields are used to fill in existed username and matched password. When user clicks on login button, program will check whether the username and password match. If they match, it will lead user to main interface. Otherwise there will be some prompts to remind the user that what is wrong with their input. When user clicks on register button, the program will lead user to the registry interface.

3.1.2.Registry interface


A Classic Cookbook
Here To Register!
Username
Enter a 6-10 digit username
Password
Enter a 6-10 digit password
Confirm password
Enter your password again
Submit
Cancel

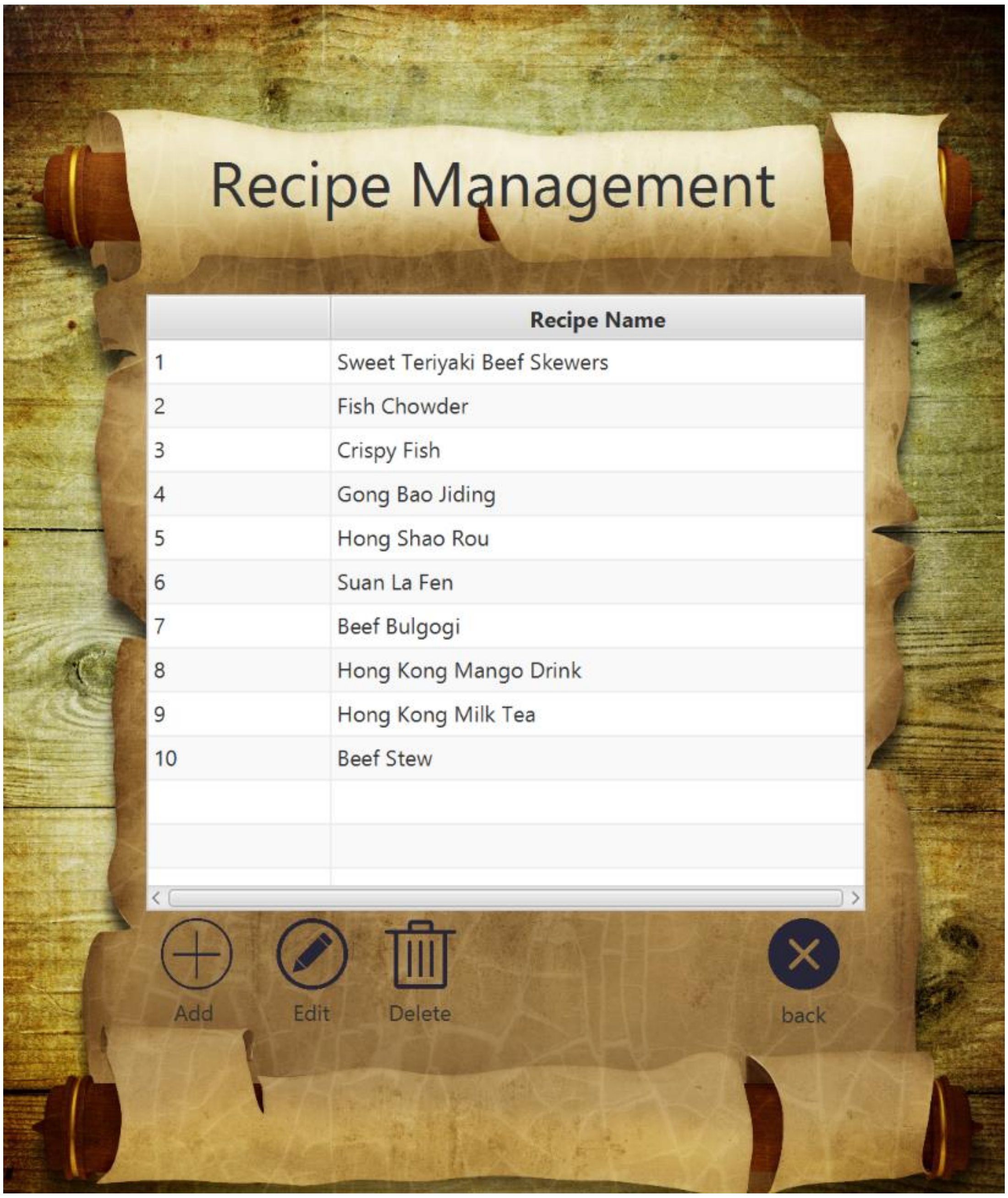
The registry interface is used for registry. This interface consists of three text fields, one submits button, one cancel button and some warm-hearted tips to remind the user that this interface is used for registry. The Username field is used for filling in username. The password field is used for filling in password and the confirm password is used to check whether the password is correct as the user want to set. When user click on submit button. The program will check 1). Whether the password is the same as the confirm password. 2). Whether is value of username and password are legal. After checking everything is correct, it will lead user to login interface again and ask them to input their account. Otherwise there will be some prompt box to remind the user that what is wrong with their input. If the user clicks cancel button, the program will lead user to login interface.

3.1.3.Main interface



The main interface consists of three image buttons. ①button is used to open the recipe management interface. ②button is used to open search interface. ③button is used to open account management interface.

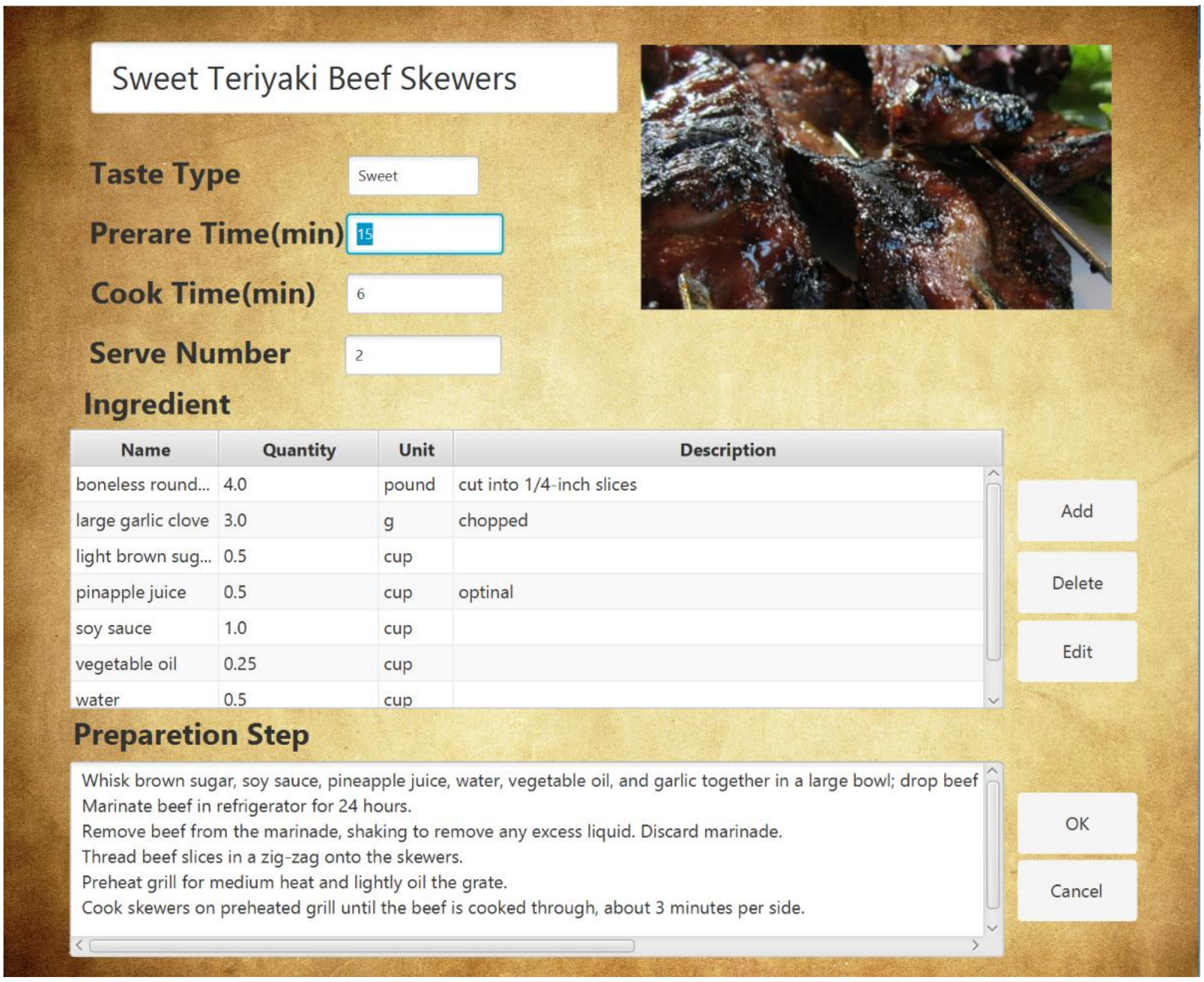
3.1.4.Recipe management interface



	Recipe Name
1	Sweet Teriyaki Beef Skewers
2	Fish Chowder
3	Crispy Fish
4	Gong Bao Jiding
5	Hong Shao Rou
6	Suan La Fen
7	Beef Bulgogi
8	Hong Kong Mango Drink
9	Hong Kong Milk Tea
10	Beef Stew

The recipe management interface is used to manage user’s own recipes. This interface consists of a list of recipe that created by the used and four buttons. User can use add new recipe button to open a new editable blank form to create a new recipe. When user selects a recipe and clicks edit this recipe button, user will open a editable form filled with existed information about the recipes. When user selects a recipe and clicks delete this recipe button, a warning box will jump out and ask about whether user want to deletes this recipe. If you clicks Yes, this recipe will be deleted. Else if you clicks cancel, you will not delete this recipe. If user has not selected any recipe and clicks delete or edit button, there will jump out a confirm box to remind the user. The back button is used to return to the main interface.

3.1.5.Recipe design interface



Sweet Teriyaki Beef Skewers

Taste Type	Sweet
Prerare Time(min)	15
Cook Time(min)	6
Serve Number	2

Ingredient

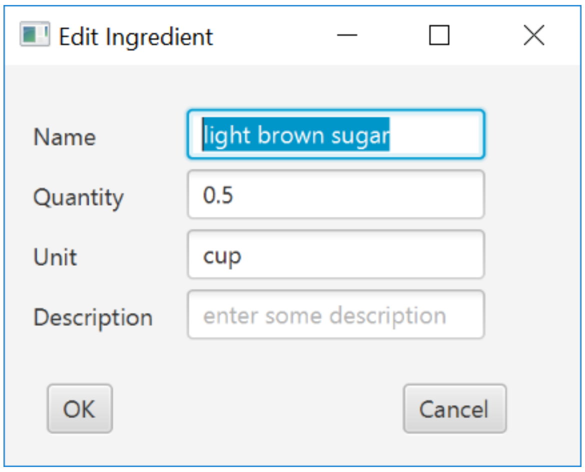
Name	Quantity	Unit	Description
boneless round...	4.0	pound	cut into 1/4-inch slices
large garlic clove	3.0	g	chopped
light brown sug...	0.5	cup	
pinapple juice	0.5	cup	optinal
soy sauce	1.0	cup	
vegetable oil	0.25	cup	
water	0.5	cup	

Preparetion Step

Whisk brown sugar, soy sauce, pineapple juice, water, vegetable oil, and garlic together in a large bowl; drop beef
Marinate beef in refrigerator for 24 hours.
Remove beef from the marinade, shaking to remove any excess liquid. Discard marinade.
Thread beef slices in a zig-zag onto the skewers.
Preheat grill for medium heat and lightly oil the grate.
Cook skewers on preheated grill until the beef is cooked through, about 3 minutes per side.

This interface consists with a form and 5 buttons. The form is used to fill in details about the recipe except ingredient, which will be edited separately. The add new ingredient button will allow user to inserts a new ingredient into the recipe. The edit this ingredient will allow user to edit selected ingredient. The delete this recipe button will delete a recipe or not after confirm whether user want to delete this recipe. If no recipe is selected, a prompt box will jump out. The OK button will hand in the form after checking whether all input is legal. Otherwise a prompt box will jump out to remind the user what is wrong with their inputs. And the cancel button will close this interface without update this recipe.

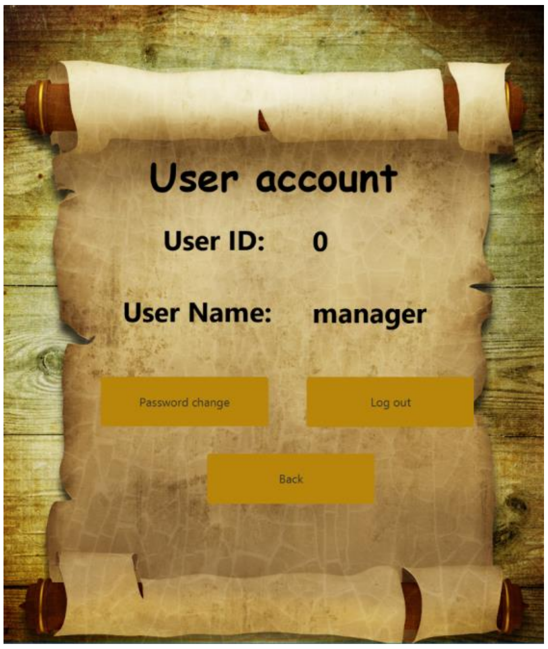
3.1.6.Ingredient design interface


Edit Ingredient

Name	light brown sugar
Quantity	0.5
Unit	cup
Description	enter some description

This interface allow user to fill in the details of an ingredient, if this interface is open by create button, there will be an additional next button. Otherwise this button will not exist. The OK button will update this ingredient after checking legality. Cancel button will close this interface without changing anything. Next button allows user to create ingredient continuously without repeatedly open and close this interface. If there’s and error exists, a prompt box will jump out.

3.1.7.Account management interface

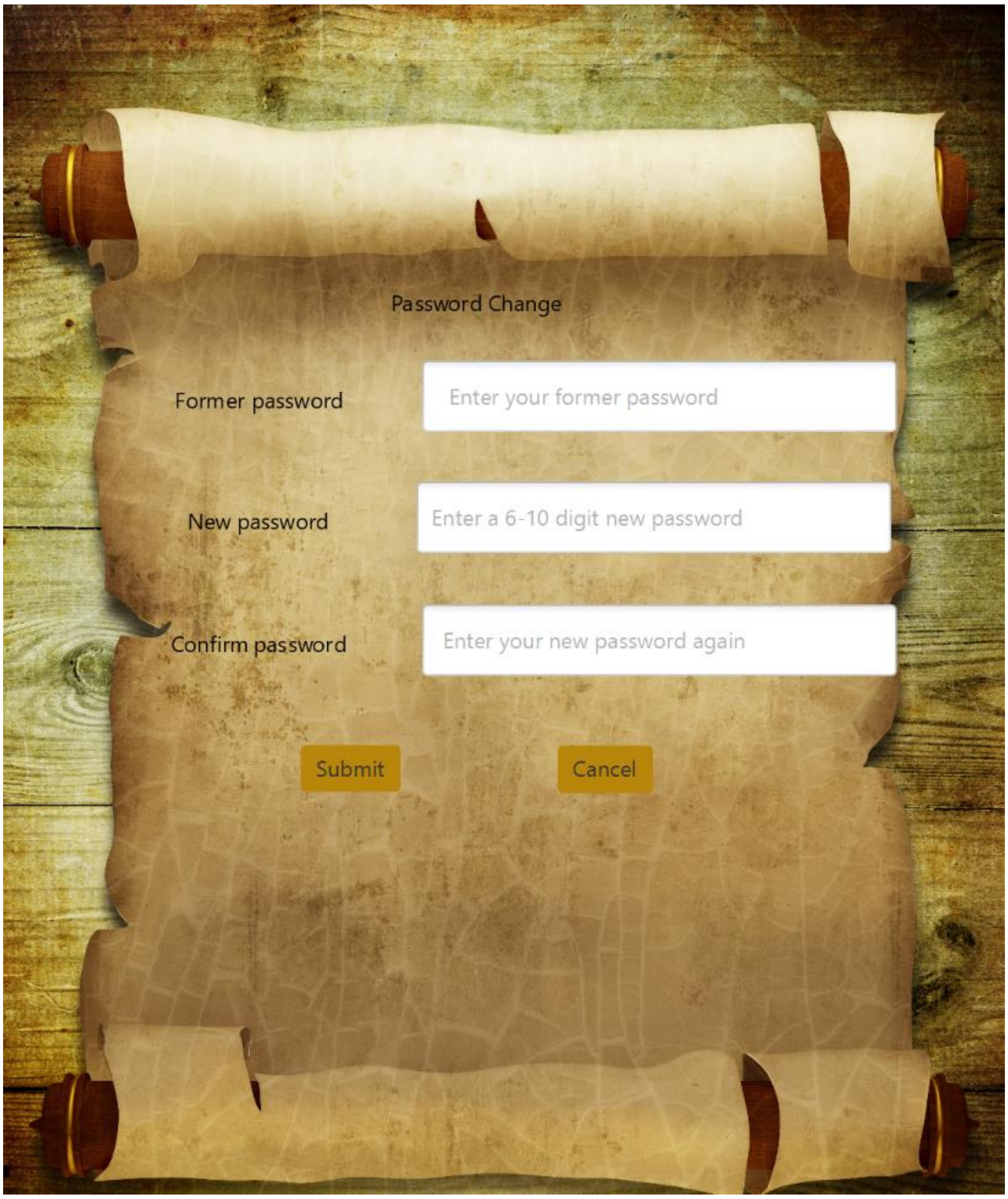

User account

User ID: 0

User Name: manager

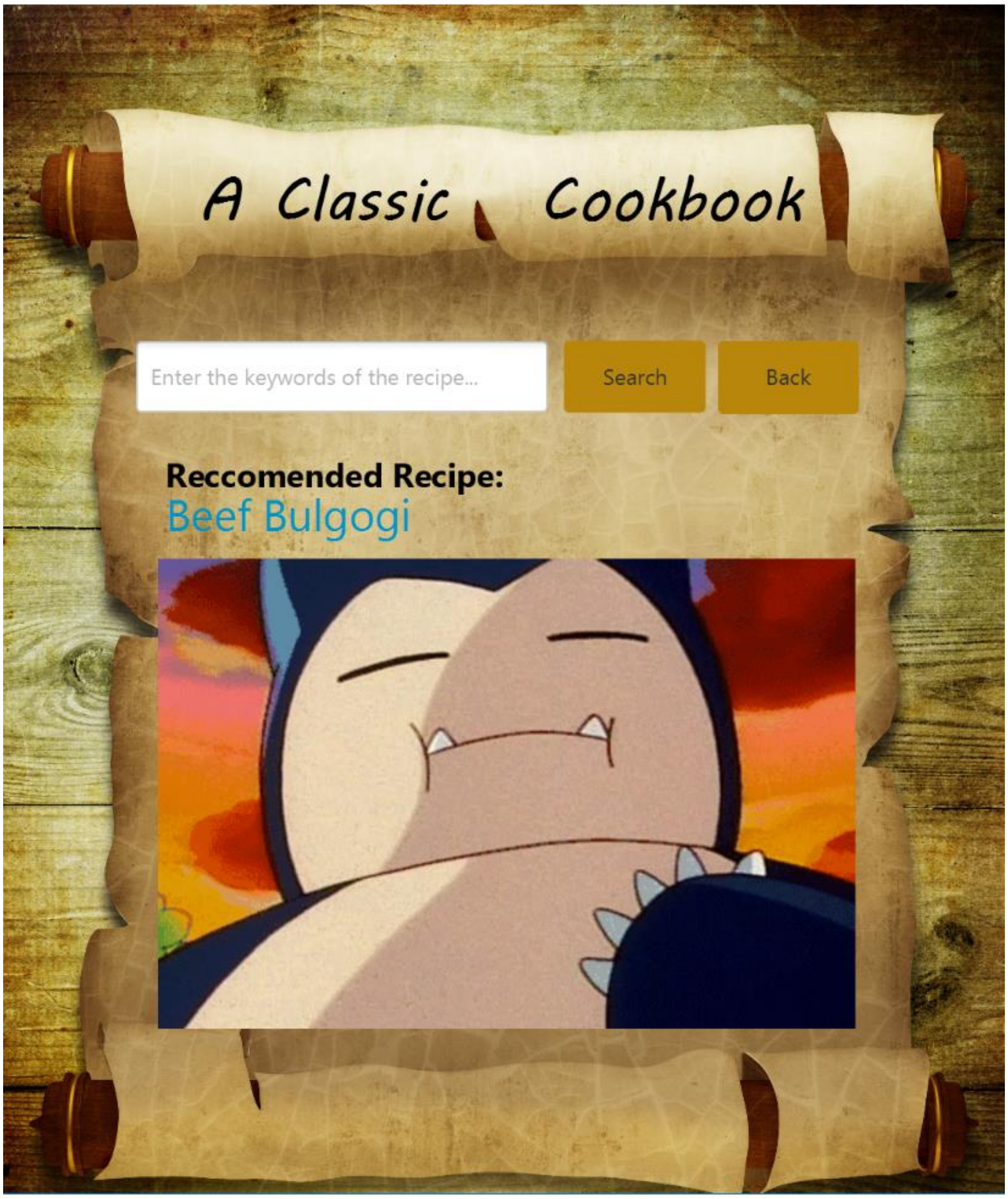
This interface will display details of user’s account. And there’s a password change button that leads user to password change interface. Log out button will allow user to log out or not after a confirm box. Back button will jump to main interface.

3.1.8.Password change interface


Password Change
Former password
Enter your former password
New password
Enter a 6-10 digit new password
Confirm password
Enter your new password again
Submit
Cancel

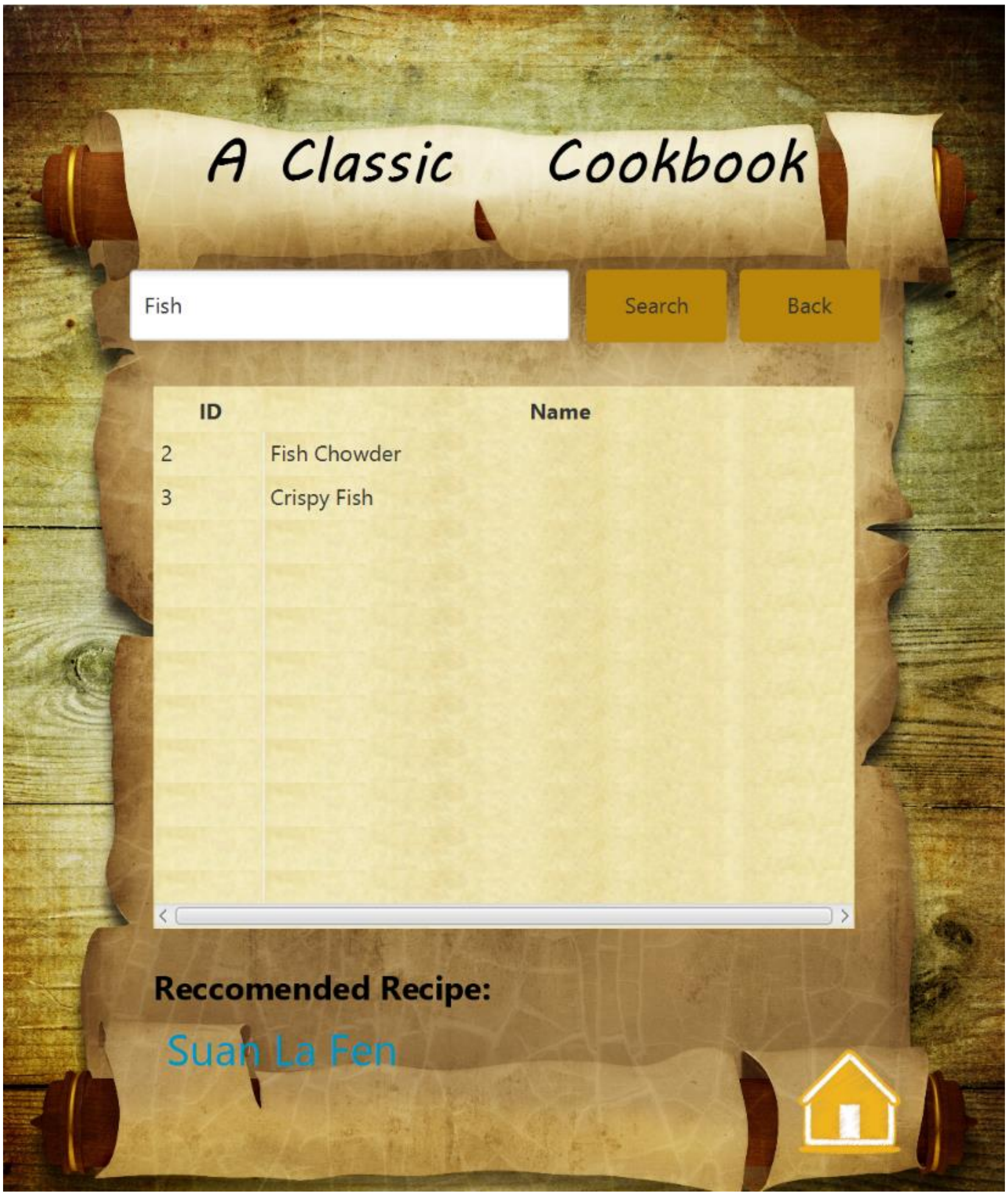
This interface allows user to change his or her password, which includes three text field and two button. Former password field is used to fill in former password. New password field is used to fill in new password and confirm password will be used to check whether this new password is correct. When user click submit button, program will update this new password after checking legality of new password and correctness of former password. If there are some error, a prompt box will jump out. The cancel button will go back to account management interface.

3.1.9.Search interface



This interface allows user to search recipes by key words. The text field is used to input keyword. When user clicks on Search button, program will search name-related recipes. If there are some recipes matching the key word, program will jump to search result interface. Otherwise it will remind user that there is no matching result. Under the search field locates a recommendation, which will recommend a recipe to user based on their latest 3 search histories. If user use this such function for the first time, recommendation will recommend a predetermined recipe. Back button will go back to main interface.

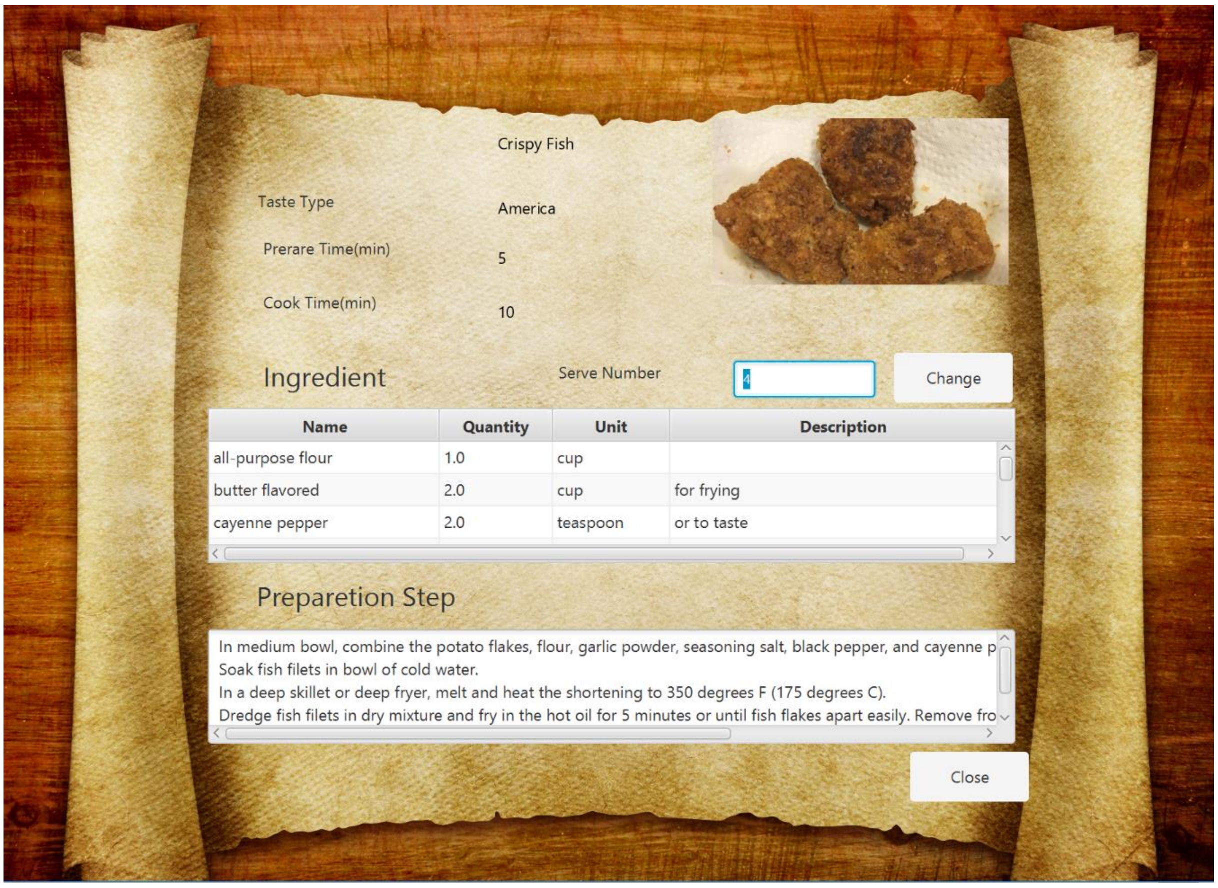
3.1.10.Search result interface



ID	Name
2	Fish Chowder
3	Crispy Fish

This interface is similar to search interface, but it will display a list of matching recipes depending on key words. When user clicks on a recipe, program will jump to recipe display interface which have all the details of this recipe.

3.1.11.Recipe display interface


Crispy Fish

Taste Type	America
Prerare Time(min)	5
Cook Time(min)	10

Ingredient

Serve Number 4

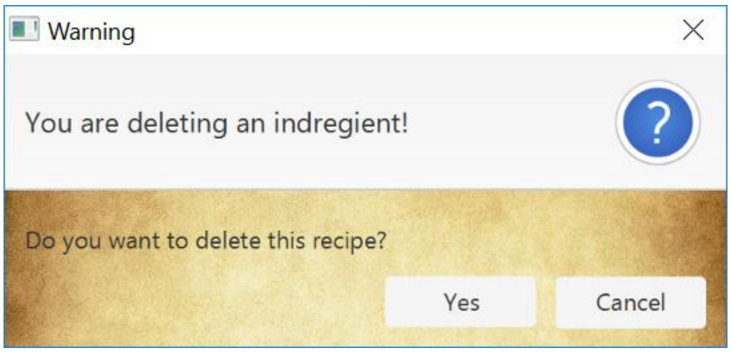
Name	Quantity	Unit	Description
all-purpose flour	1.0	cup	
butter flavored	2.0	cup	for frying
cayenne pepper	2.0	teaspoon	or to taste

Preparetion Step

In medium bowl, combine the potato flakes, flour, garlic powder, seasoning salt, black pepper, and cayenne p
Soak fish filets in bowl of cold water.
In a deep skillet or deep fryer, melt and heat the shortening to 350 degrees F (175 degrees C).
Dredge fish filets in dry mixture and fry in the hot oil for 5 minutes or until fish flakes apart easily. Remove fro

This interface will show all the information of a recipe including ingredients and preparation steps. While all of this information is uneditable except the server number. When user change the number of server number and clicks on change button, the quantity of each ingredients will change based on input server number. Back button will go back to search result interface

3.1.12.Confirm box

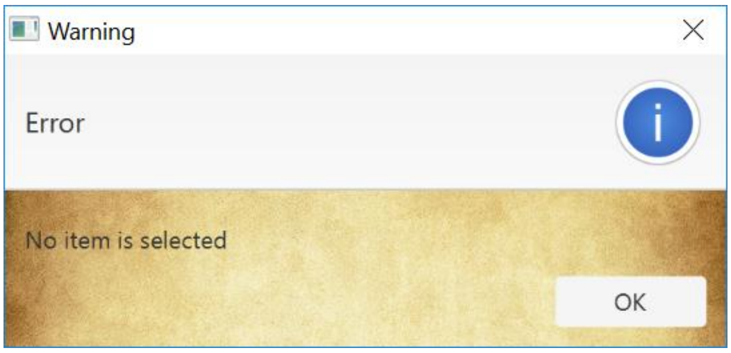

Warning

You are deleting an indregient!

Do you want to delete this recipe?

This box is used to confirm whether user want to execute this function or not.

3.1.13.Inform box


Warning

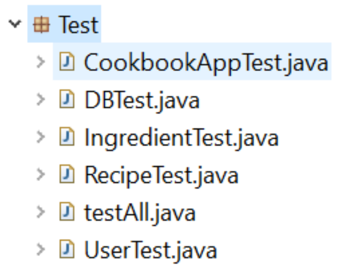
Error

No item is selected

This box is used to remind user what is wrong with their operations.

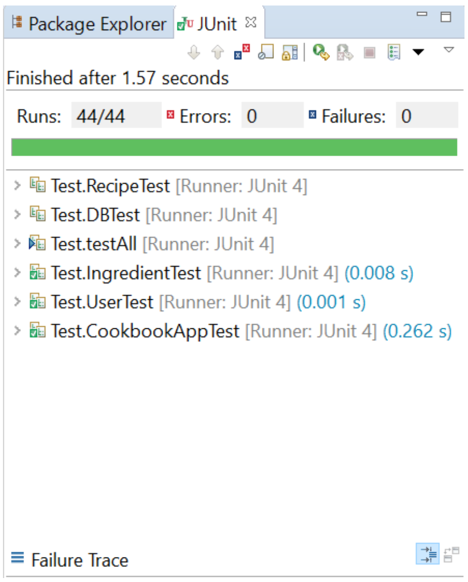
4. Test

4.1 Junit Test



```
Test
  CookbookAppTest.java
  DBTest.java
  IngredientTest.java
  RecipeTest.java
  testAll.java
  UserTest.java
```

We have five test class for Junit class, and another one for initialize all the Junit test

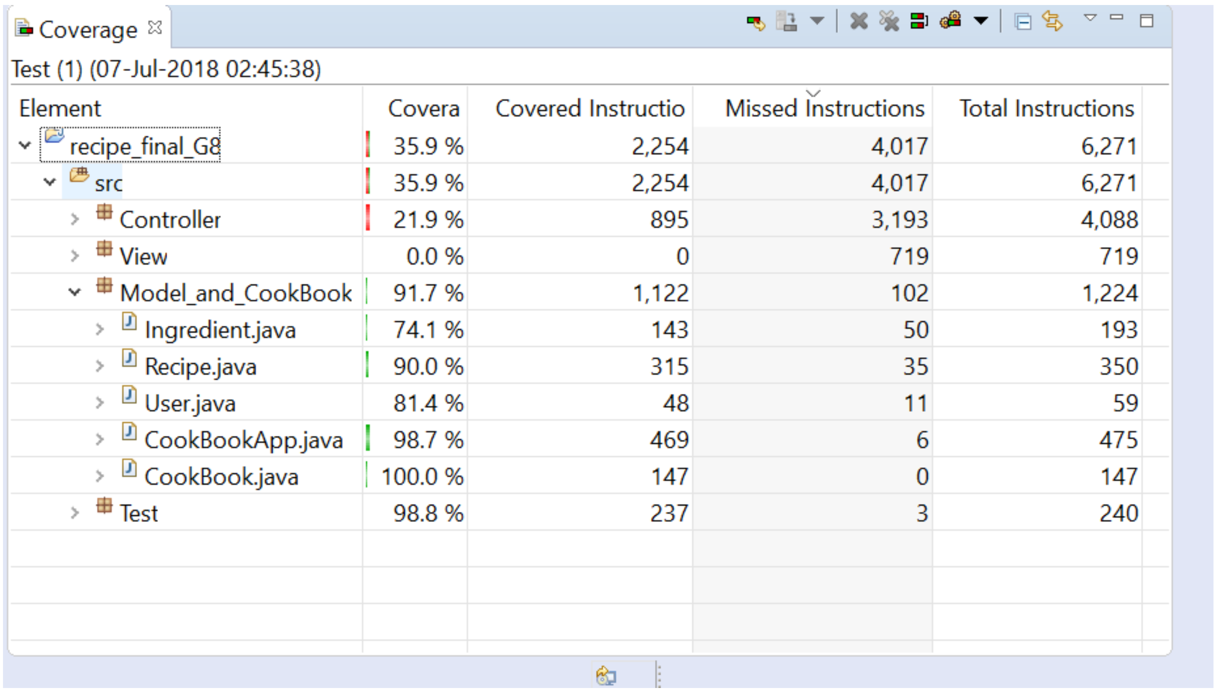


```
Package Explorer  JUnit
Finished after 1.57 seconds
Runs: 44/44   Errors: 0   Failures: 0

Test.RecipeTest [Runner: JUnit 4]
Test.DBTest [Runner: JUnit 4]
Test.testAll [Runner: JUnit 4]
Test.IngredientTest [Runner: JUnit 4] (0.008 s)
Test.UserTest [Runner: JUnit 4] (0.001 s)
Test.CookbookAppTest [Runner: JUnit 4] (0.262 s)

Failure Trace
```

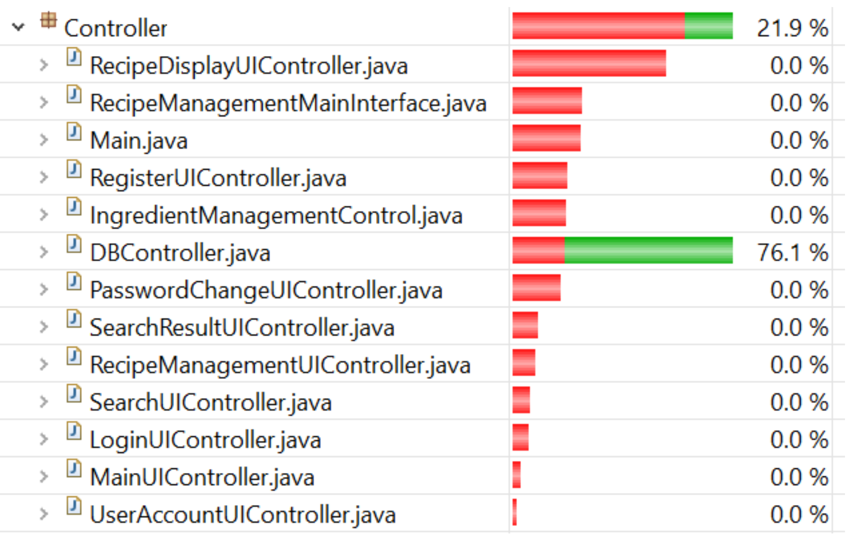
We used EclEmma to analysis Junit test coverage. The coverage result is following:



Coverage

Test (1) (07-Jul-2018 02:45:38)

Element	Covera	Covered Instructio	Missed Instructions	Total Instructions
recipe_final_G8	35.9 %	2,254	4,017	6,271
src	35.9 %	2,254	4,017	6,271
Controller	21.9 %	895	3,193	4,088
View	0.0 %	0	719	719
Model_and_CookBook	91.7 %	1,122	102	1,224
Ingredient.java	74.1 %	143	50	193
Recipe.java	90.0 %	315	35	350
User.java	81.4 %	48	11	59
CookBookApp.java	98.7 %	469	6	475
CookBook.java	100.0 %	147	0	147
Test	98.8 %	237	3	240



Controller	🟥🟩	21.9 %
RecipeDisplayUIController.java	🟥	0.0 %
RecipeManagementMainInterface.java	🟥	0.0 %
Main.java	🟥	0.0 %
RegisterUIController.java	🟥	0.0 %
IngredientManagementControl.java	🟥	0.0 %
DBController.java	🟥🟩	76.1 %
PasswordChangeUIController.java	🟥	0.0 %
SearchResultUIController.java	🟥	0.0 %
RecipeManagementUIController.java	🟥	0.0 %
SearchUIController.java	🟥	0.0 %
LoginUIController.java	🟥	0.0 %
MainUIController.java	🟥	0.0 %
UserAccountUIController.java	🟥	0.0 %

As we can see, the test has a high coverage of the Model part. It also cover 76.1% of DBController for almost all the function, but not every branch.

It's hard to test the code about view interactions.

4.2 Equivalence test

According to our specification, there are five situations where we need to get input from user. So, we implemented Equivalence tests for all the five situations.

4.2.1 Equivalence test for register

The EC are defined as below:

Test field	Valid EC classes	Invalid EC classes

Username(register)	Name1 = {text | text must be String type, 6 =< length <= 10}	Name2 = {text | empty text}
		Name3 = {text | length < 6}
		Name4 = {text | length > 10}

Password(register)	Password1 = {text | text must be String type, 6 =< length <= 10}	Password2 = {text | empty text}
		Password3 = {text | length < 6}
		Password4 = {text | length > 10}

Confirm(register)	Confirm1 = {text | text must be String type}	Confirm 2 = {text | empty text}

The test results:

Test field	input	output
Username(register)	1. Name1 = 12345678 2. Name2 = null 3. Name3 = 123	1. The application can run normally. 2. The application will give you a prompt text. 3. The application will give you a prompt text.

	4. Name4 = 1234567890	4. The application will give you a prompt text.
Password(register)	1. Password1 = 12345678 2. Password2 = null 3. Password3 = 123 4. Password4 = 1234567890	1. The application can run normally. 2. The application will give you a prompt text. 3. The application will give you a prompt text. 4. The application will give you a prompt text.
Confirm(register)	1. Confrim1 = 12345678 2. Confirm2 = null	1. The application can run normally. 2. The application will give you a prompt text.

4.2.2 Equivalence test for login

The EC are defined as below:

Test field	Valid EC classes	Invalid EC classes
Username(login)	Name1 = {text | text must be String type}	Name2 = {text | empty text}
Password(login)	Password1 = {text | text must be String type}	Password2 = {text | empty text}

The test results:

Test field	input	output
Username(login)	1. Name1 = 12345678 2. Name2 = null	1. The application can run normally. 2. The application will give you a prompt text.
Password(login)	1. Password1 = 12345678 2. Password2 = null	1. The application can run normally. 2. The application will give you a prompt text.

4.2.3 Equivalence test for recipe search

The EC are defined as below:

Test field	Valid EC classes	Invalid EC classes

Keyword(search)	Keyword1 = {text | text must be String type}	Keyword 2 = {text | empty text}

The test results:

Test field	input	output
Keyword(search)	1. Keyword1 = Fish 2. Keyword2 = null	1. Giving all the recipes which name includes the word 'Fish' 2. The application don't display any result.

4.2.4 Equivalence test for password change

The EC are defined as below:

Test field	Valid EC classes	Invalid EC classes

Former Password (password change)	Former1 = {text | text must be String type, 6 =< length <= 10}	Former2 = {text | empty text}
		Former3 = {text | length < 6}
		Former4 = {text | length > 10}

New Password (password change)	New1 = {text | text must be String type, 6 =< length <= 10}	New2 = {text | empty text}
		New3 = {text | length < 6}
		New4 = {text | length > 10}

Confirm (password change)	Confirm1 = {text | text must be String type}	Confirm 2 = {text | empty text}

The test results:

Test field	input	output
Former Password (password change)	1. Former 1 = 12345678 2. Former 2 = null 3. Former 3 = 123 4. Former 4 = 1234567890	1. The application can run normally. 2. The application will give you a prompt text. 3. The application will give you a prompt text. 4. The application will give you a prompt text.
New Password (password change)	1. New 1 = 12345678 2. New 2 = null 3. New 3 = 123 4. New 4 = 1234567890	1. The application can run normally. 2. The application will give you a prompt text. 3. The application will give you a prompt text. 4. The application will give you a prompt text.

Confirm (password change)	1. Confrim1 = 12345678 2. Confirm2 = null	1. The application can run normally. 2. The application will give you a prompt text.

4.2.5 Equivalence test for add recipe and update recipe

The EC are defined as below:

Test field	Valid EC classes	Invalid EC classes
Recipe name	Recipe name1 = {text | text must be String type}	Recipe name2 = {text | empty text}
Taste type	Taste type1 = {text | text must be String type}	

Preparer Time	Preparer Time1 = {text | text must be numeric type, text != 0}	Preparer Time2 = {text | empty text}
		Preparer Time3 = {text | text = 0}

Cook time	Cook time1 = {text | text must be numeric type, text != 0}	Cook time2 = {text | empty text}
		Cook time3 = {text | text = 0}

Serve number	Server number1 = {text | text must be numeric type, text != 0}	Server number2 = {text | empty text}
		Server number3 = {text | text = 0}

Ingredient name	Ingredient name1 = {text | text must be String type}	Ingredient name2 = {text | empty text}

Quantity	Quantity1 = {text | text must be Double type, text != 0 }	Quantity2 = {text | empty text}
		Quantity3 = {text | text = 0}

Unit	Unit1 = {text | text must be String type}	Unit2 = {text | empty text}
Preparation step	EachStep1 = {text | 0 <= length < =255}	EachStep1 = {text | length > 255}

The test results:

Test field	input	output
Recipe name	1. Recipe 1 = Fish 2. Recipe 2 = null	1. The application can run normally. 2. The application will pop up a warning dialog.
Taste type	1. Taste type1 = Hu Nan	1. The application can run normally.

Preparer Time	1. Preparer Time1 = 10 2. Preparer Time2 = null 3. Preparer Time3 = 0	1. The application can run normally. 2. The application will pop up a warning dialog. 3. The application will pop up a warning dialog.
Cook time	1. Cook time1 = 10 2. Cook time2 = null 3. Cook time3 = 0	1. The application can run normally. 2. The application will pop up a warning dialog. 3. The application will pop up a warning dialog.
Serve number	1. Server number1 = 10 2. Server number2 = null 3. Server number3 = 0	1. The application can run normally. 2. The application will pop up a warning dialog. 3. The application will pop up a warning dialog.
Ingredient name	1. Ingredient name1 = Beef 2. Ingredient name2 = null	1. The application can run normally. 2. The application will pop up a warning dialog.
Quantity	1. Quantity1 = 10.0 2. Quantity2 = null 3. Quantity3 = 0.0	1. The application can run normally. 2. The application will pop up a warning dialog. 3. The application will pop up a warning dialog.
Unit	1. Unit1 = kg 2. Unit2 = null	1. The application can run normally. 2. The application will pop up a warning dialog.
Preparation step	1. EachStep1 = “wash the tomato” 2. EachStep2 (more than 255 words)	1. The application can run normally. 2. The application will pop up a warning dialog.

Boundary value analysis

Input: Username (register)

The length of register username must between 6 to 10

0 5	6 10	11
invalid	valid	invalid

Test case for

Username.length =0, 2, 5, 6, 8, 10, 11, 15

0: e.g. null The program clears the input text and gives you a prompt text said “Enter a 6-10 digit username” .

2: e.g. “11” The program clears the input text and gives you a prompt text said, “Enter a 6-10-digit username” .

5: e.g. “12345” The program clears the input text and gives you a prompt text said “Enter a 6-10-digit username”

6: e.g. “666666” The program can run normally.

8: e.g. “88888888” The program can run normally.

10: e.g. “1234567890” The program can run normally.

11: e.g. “11111111111” The program clears the input text and gives you a prompt text. said “Enter a 6-10-digit username”

15: e.g. “151515151515151” The program clears the input text and gives you a prompt text said “Enter a 6-10-digit username”

Input: Password (register)

The length of register password must between 6 to 10

0 5	6 10	11
invalid	valid	invalid

Test case for

Password.length =0,2, 5, 6, 8, 10, 11, 15

0: e.g. null. The program clears the input text and gives you a prompt text said “Enter a 6-10 digit password” .

2: e.g. “12” The program clears the input text and gives you a prompt text said “Enter a 6-10 digit password” .

5: e.g. “12345” The program clears the input text and gives you a prompt text said “Enter a 6-10-digit password”

6: e.g. “666666” The program can run normally.

8: e.g. “88888888” The program can run normally.

10: e.g. “1234567890” The program can run normally.

11: e.g. “11111111111” The program clears the input text and gives you a prompt text. said “Enter a 6-10-digit password”

15: e.g. “151515151515151” The program clears the input text and gives you a prompt text said “Enter a 6-10-digit password”

Input: New password (password change)

The length of new password must between 6 to 10

0 5	6 10	11
invalid	valid	invalid

Test case for

Password.length =0, 2, 5, 6, 8, 10, 11, 15

0: e.g. null The program clears the input text and gives you a prompt text said "Enter a 6-10 digit new password".

2: e.g. "22" The program clears the input text and gives you a prompt text said "Enter a 6-10 digit new password".

5: e.g. "12345" The program clears the input text and gives you a prompt text said "Enter a 6-10-digit new password"

6: e.g. "666666" The program can run normally.

8: e.g. "88888888" The program can run normally.

10: e.g. "1234567890" The program can run normally.

11: e.g. "11111111111" The program clears the input text and gives you a prompt text. said "Enter a 6-10-digit new password"

15: e.g. "151515151515151" The program clears the input text and gives you a prompt text said "Enter a 6-10-digit new password"

4.3 Usability Test

4.3.1. Description

A good software not only need excellent developers, but also need users to make some constructive suggestions for improving the software based on real life scenarios. And when users use the software, because of the operating system or various environments, so it is easier for user to find out bugs and errors. As a software developer, we need to collect some problems or suggestion from user for improving our software, thus, we need do usability test in our application test.

Target users and respondents

Before doing usability test, we need identify our target users. After discussing in our group, we decide our target users are young adults (age between 20 – 30) who can use computer software and have interests in cooking. On the other hand, our software is an English version. Thus, our target users need have basic knowledge of English. After determining the target user feature, we find 5 respondents who have nearly no knowledge of computer science to help us finish the usability test.

Goals and methods

After finding 5 respondents, 2e would like to specifically know usable our is our application from usability test. We would like to get some feedback from users. For example, is prompt information and layout clear to users, how long does it take for user to have a command of the application. Therefore, we design some task and scenarios from real life to get the information of usability test, and we made a questionnaire for user about our test.

Tasks for usability test:

1. One day, you want to invite your friends to eat dinner together at home. And you want to cook two traditional Chinese dishes. One is Hong Shao Rou and you have no idea about the second dish. Firstly, you register the software and then you use our software to search the Hong Shao Rou recipe. Through the software, you know the cook step and the amount of ingredients. And after searching you get a recommendation.

2. You want to add a new recipe. And you need to fill the blanks and edit ingredient/step in recipe interface. (Try to make mistakes, such as inputting letters to time field.)

3. You want to edit your own recipe. Try to edit a saved recipe and save it. Then check if the recipe changes in search interface.

4. You want to delete your own recipe. Delete it and then check if the recipe exists.

5. Check your user ID and try to change your password by prompt text.

6. If you want, you can log out and log in another account.

Questionnaire

1. To what extent do you think it is easy to accomplish these tasks? Very hard/ hard/ common/ easy/ very easy

2. Is prompt information clear to you?

3. Does our search engine provide the right result and related recommendation？

4. How usable is our application?

5. How clear and well organized the layout is? Do you like the layout?

6. What suggestions you have for our software?

4.3.2. Results

Tester	Test time	Bugs	Suggestions
Ye	10 min	Password change method	1. The prompt text should be clearer. 2. Add a home button. 3. Embellish the login interface.
Li	13 min		1. Add a home button. 2. Optimize the recommendation.
Mr. Ye	12 min	Ingredient might be deleted randomly.	1. The prompt text should be red font. 2. Before logout, software need pop up a dialog. 3. There should be a next button in ingredientAdd interface.
Zhou	15 min	Ingredient might be deleted randomly.	
Guo	30min		1. Embellish the login interface.

Totally 5 persons are invited to do our usability test. And we collected the questionnaire and got some feedbacks and suggestions from users.

From our feedback questionnaire, we know that most of test users think it is easy for them to use our software and the layout is clear to them. But some improvements need to be made, for example, embellish the login button, setting the prompt text clearer. Each of them is satisfied with our software. They think our software is usable and our software can provide the right result and recommendation for users.

There are some bugs when tester use cook book. For example, Mr. Ye cannot change his password and the ingredient might be deleted randomly. Thanks to their discovery we are able to fix bugs we have not noticed before. We fix the bugs in our project.

We thought over the suggestions and adopted some of them. For example, Mr. Ye told us it is inconveniently for users to add ingredients. So, we added a “next” button for improving the operational feeling. As for feedbacks towards screen design, some can be changed with slight effort and quite helpful to improve usability, like changing the position of some buttons, change the font of prompt text and so on.

After finishing the usability test, we improve our software and make it more usable. Thanks for our tester.

5. Evaluation

5.1. Group Work

We discussed the issues at least every week and we consulted our supervisor to solve our questions every week. Our supervisor plays the key role in solving our questions. And we also expressed our opinions in group meeting.

Here is our meeting record:

4.27 evening, meeting: specification absent：Shi

4.30 evening, meeting: specification absent：Sun

5.4 morning, meeting: usecase diagram

5.10 evening, meeting: framework of class diagram

5.12 evening, meeting: attributes and function of class diagram absent：Shi

5.21 evening, meeting: E-R diagram absent：Shi

5.23 evening, meeting: class diagram, use-case diagram and specification refactor

5.24 morning, meeting: code discuss and writing

6.4 evening, meeting: code discuss and writing

6.10 evening, meeting: code discuss and writing

6.13 evening, meeting: code discuss and writing

6.18 evening, meeting: code discuss and writing absent：Shi

6.24 evening, meeting: code discuss and writing absent：Shi

6.27 evening, meeting: code improvement and report writing absent：Shi

6.30 evening, meeting: code improvement and report writing absent：Shi

7.1 evening, meeting: code improvement and report writing absent：Shi

7.2 evening, meeting: code improvement and report writing absent：Shi

7.3 evening, meeting: code improvement and report writing absent：Shi

7.4 evening, meeting: code improvement and report writing absent：Shi

7.5 evening, meeting: code improvement and report writing absent：Shi

After finishing the specification, we decided to divide our project into three parts approximately. The first part is user, the second part is recipe and ingredient, the last part is search. Yang Yiming is responsible for the user part. Sun Xiaoan is responsible for the recipe and ingredient part. And Dong Tianyuan is responsible for the search part. Everyone wrote their own part code. After all of us completed our own parts. Dong Tianyuan and Sun Xiaoan then focused on the combination of codes from each part. Sun Xiaoan mainly took responsible for debug, Dong Tianyuan took responsible for beautifying.

When it went to report part, Sun Xiaoan rewrote the specification and modified the class diagram and use case diagram. He also wrote the screenshots part and implemented some comments. Yang Yiming wrote the boundary test, usability test and a part of evaluation. Yang Yiming and Sun Xiaoan fixed the bugs found in usability test together. Dong Tianyuan wrote Junit test and accomplish the evaluation.

We work together and work hard. Every one of us has learnt a lot and appreciates a lot.

5.2. Task Responsibilities

	Dong Tianyuan	**Sun Xiaoan**	**Yang Yiming**	**Shi Fangbo**
CookBook.java	X			
CookBookApp.java	X	X		
DBConnector.java	X	X	X	
Ingredient.java		X		
Recipe.java		X		
User.java	X		X	
IngredientManagementControl.java		X		
LoginUIController.java			X	
Main.java	X	X		
MainUIController.java	X	X		
PasswordChangeUIController.java			X	
RecipeDisplayUIController.java		X		

RecipeManagementMainInterface.java		X		
RecipeManagementUIController.java		X		
RegisterUIController.java			X	
SearchResultUIController.java	X			
SearchUIController.java	X			
UserAccountUIController.java			X	
ForMain.css	X			
ForRecipe.css	X			
ForUser.css	X			
IngredientManagmentUI_forNext.fxml		X		
IngredientManagementUI.fxml		X		
LoginUI.fxml			X	
MainUI.fxml	X			
PasswordChangeUI.fxml			X	
RecipeDisplayUI.fxml		X		
RecipeDisplayUI(for search).fxml		X		

RecipeManagementUI.fxml		X		
registerUI.fxml			X	
RootLayout.fxml		X		
SearchResultUi.fxml	X			
SearchUI.fxml	X			
UserAccounUI.fxml			X	
DBTest.java	X			
IngredientTest.java	X			
RecipeTest.java	X			
testAll.java	X			
UserTest.java	X			
Readme.txt		X		
Database work		X	X	
Report	X	X	X	