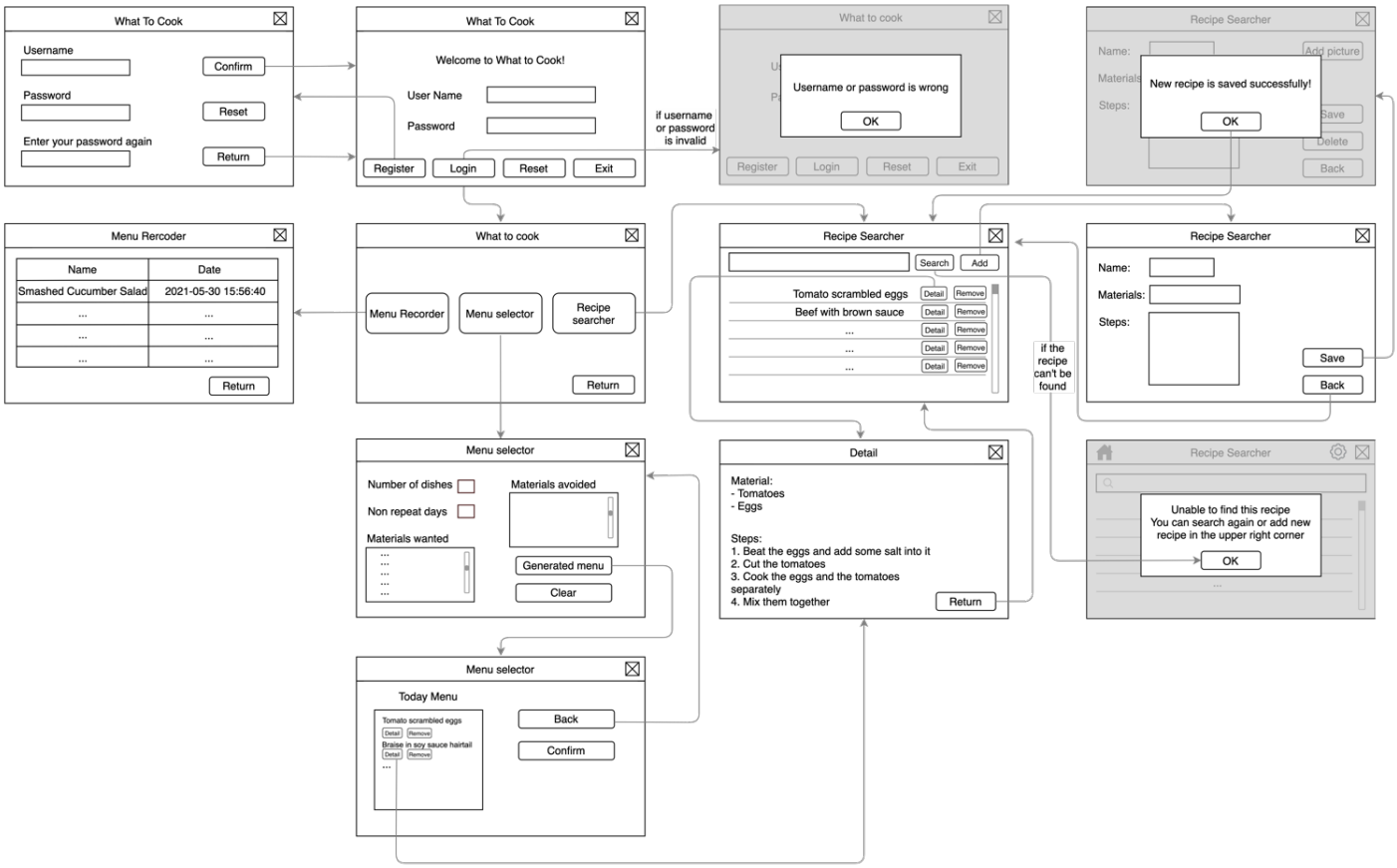
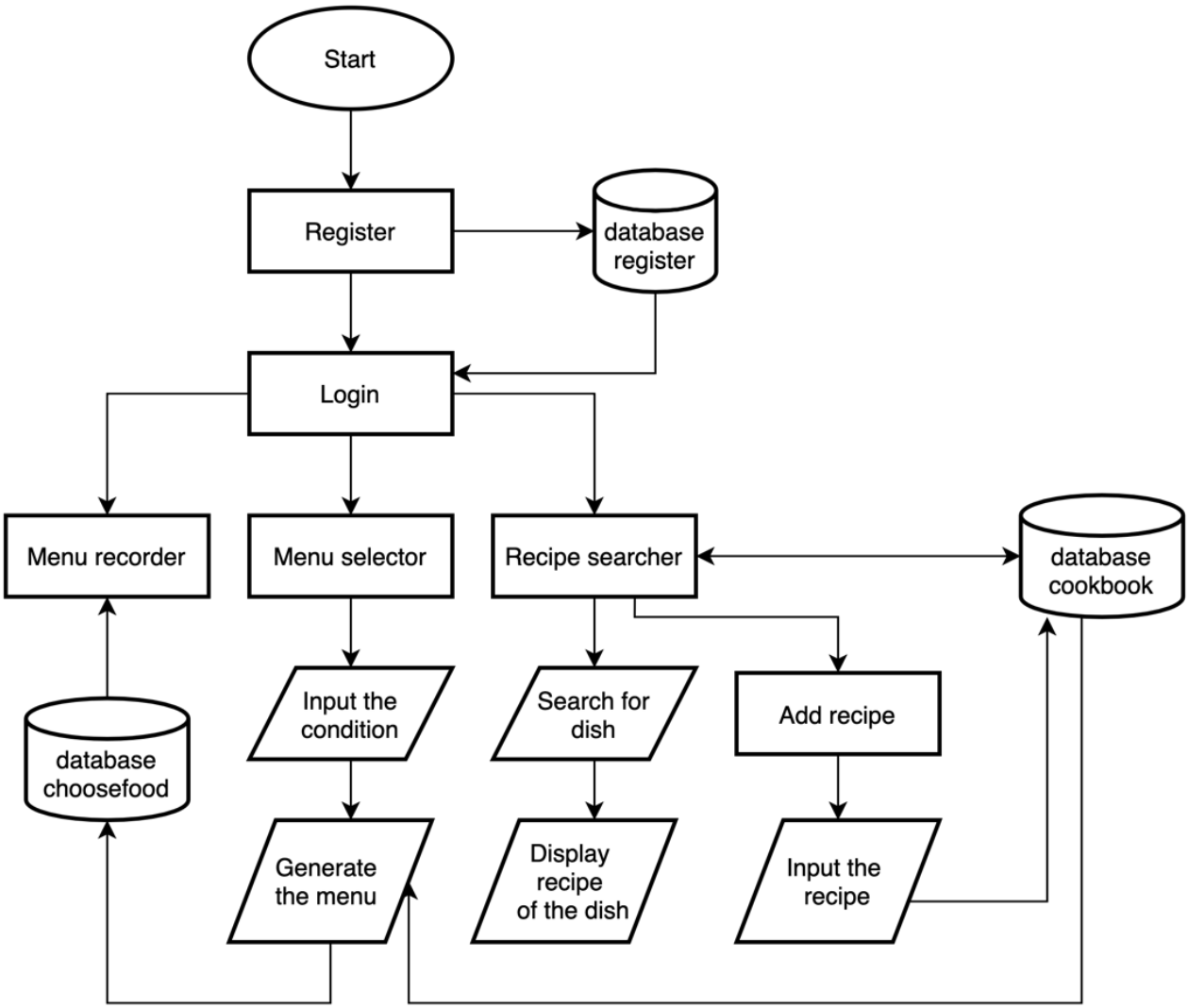
Criterion B: Design

Big picture

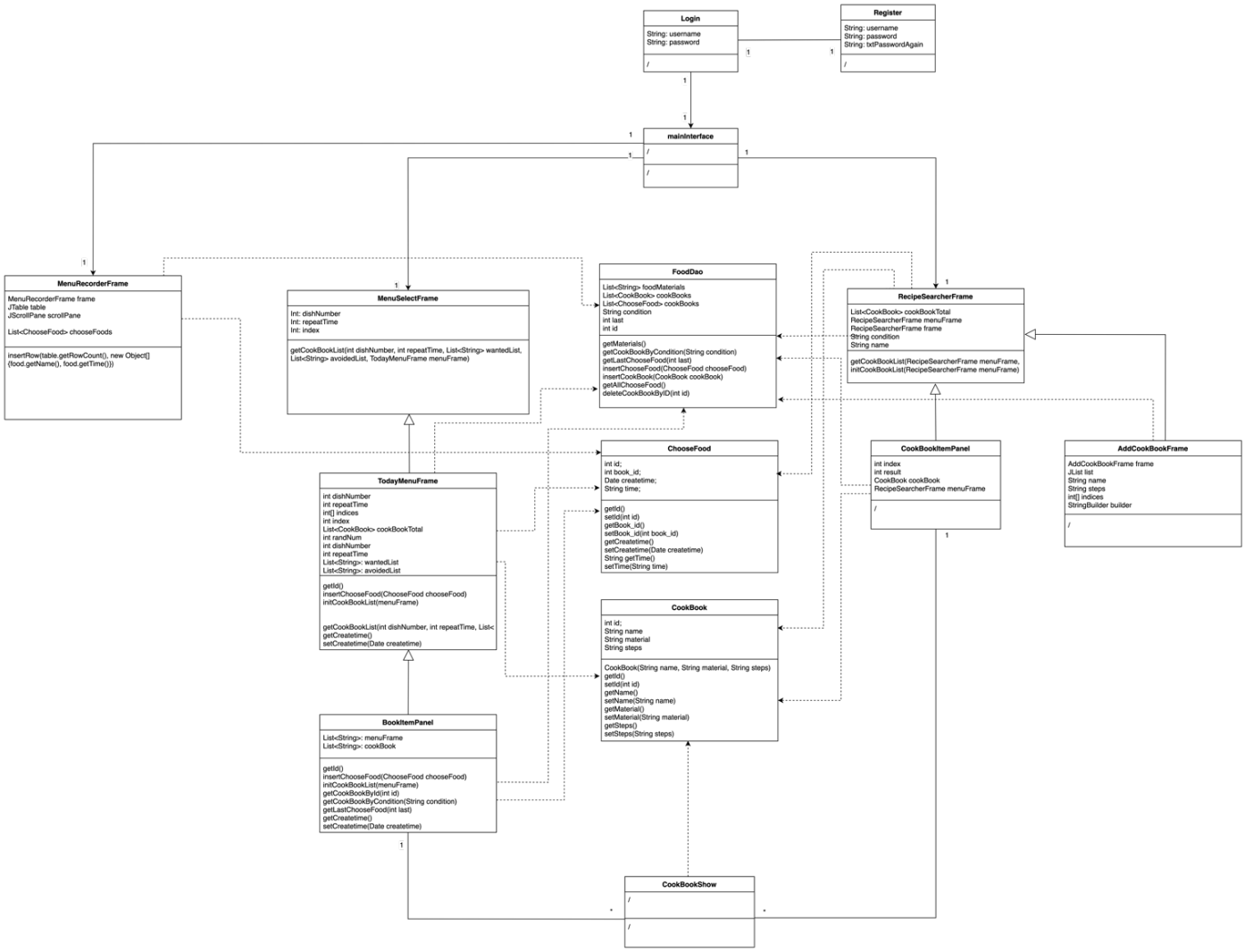


System flowchart

This flowchart shows the complete process of how the program What To Cook operates. It has three main functions which are Menu recorder, Menu selector, and Recipe searcher. Three databases are used in those functions, including register, cookbook, and choosefood.

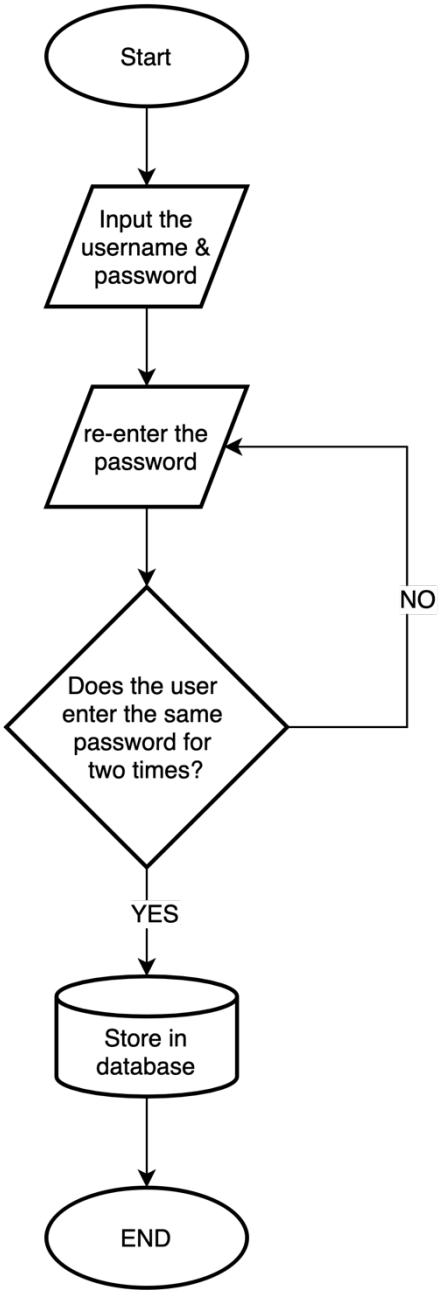


UML diagram


Login
String: username
String: password
/
Register
String: username
String: password
String: txtPasswordAgain
/
mainInterface
/
/
MenuRecorderFrame
MenuRecorderFrame frame
JTable table
JScrollPane scrollPane
List<ChooseFood> chooseFoods
insertRow(table.getRowCount(), new Object[]{food.getName(), food.getTime()})
MenuSelectFrame
Int: dishNumber
Int: repeatTime
Int: index
getCookBookList(int dishNumber, int repeatTime, List<String> wantedList, List<String> avoidedList, TodayMenuFrame menuFrame)
FoodDao
List<String> foodMaterials
List<CookBook> cookBooks
List<ChooseFood> cookBooks
String condition
int last
int id
getMaterials()
getCookBookByCondition(String condition)
getLastChooseFood(int last)
insertChooseFood(ChooseFood chooseFood)
insertCookBook(CookBook cookBook)
getAllChooseFood()
deleteCookBookByID(int id)
RecipeSearcherFrame
List<CookBook> cookBookTotal
RecipeSearcherFrame menuFrame
RecipeSearcherFrame frame
String condition
String name
getCookBookList(RecipeSearcherFrame menuFrame, initCookBookList(RecipeSearcherFrame menuFrame)
ChooseFood
int id;
int book_id;
Date createtime;
String time;
getId()
setId(int id)
getBook_id()
setBook_id(int book_id)
getCreatetime()
setCreatetime(Date createtime)
String getTime()
setTime(String time)
CookBookItemPanel
int index
int result
CookBook cookBook
RecipeSearcherFrame menuFrame
/
AddCookBookFrame
AddCookBookFrame frame
JList list
String name
String steps
int[] indices
StringBuilder builder
/
TodayMenuFrame
int dishNumber
int repeatTime
int[] indices
int index
List<CookBook> cookBookTotal
int randNum
int dishNumber
int repeatTime
List<String>: wantedList
List<String>: avoidedList
getId()
insertChooseFood(ChooseFood chooseFood)
initCookBookList(menuFrame)
getCookBookList(int dishNumber, int repeatTime, List<
getCreatetime()
setCreatetime(Date createtime)
CookBook
int id;
String name
String material
String steps
CookBook(String name, String material, String steps)
getId()
setId(int id)
getName()
setName(String name)
getMaterial()
setMaterial(String material)
getSteps()
setSteps(String steps)
BookItemPanel
List<String>: menuFrame
List<String>: cookBook
getId()
insertChooseFood(ChooseFood chooseFood)
initCookBookList(menuFrame)
getCookBookById(int id)
getCookBookByCondition(String condition)
getLastChooseFood(int last)
getCreatetime()
setCreatetime(Date createtime)
CookBookShow
/
/

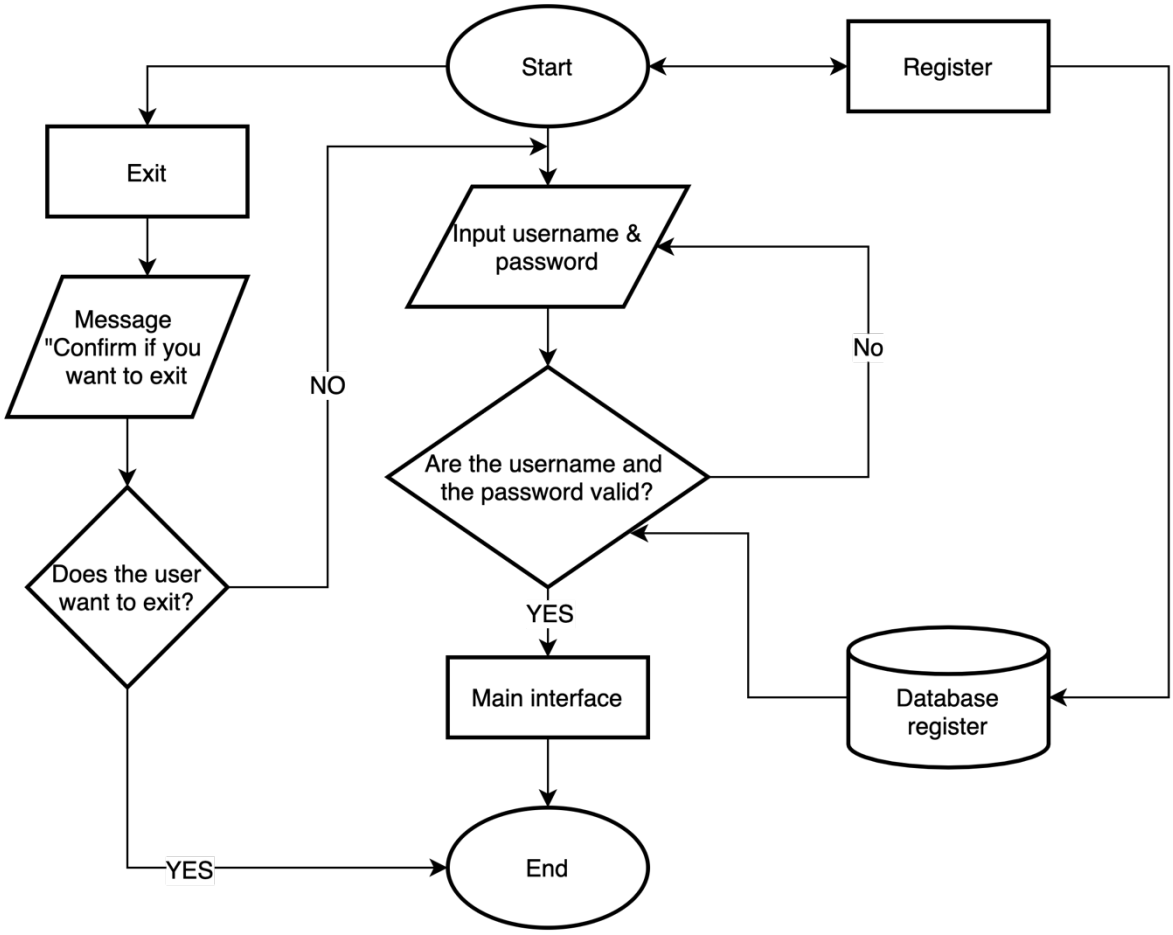
Flowchart for register window

This flowchart shows how the program adds newly registered users to the database by storing the username and the password. If the re-entered password is wrong, the program will ask the user to re-enter the password again.



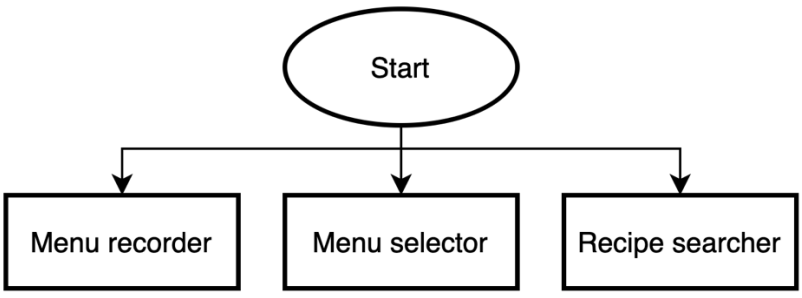
Flowchart for login window

This flowchart shows how the program verify the login information. It asks the user to reenter the username and password if they can't be found in the database.



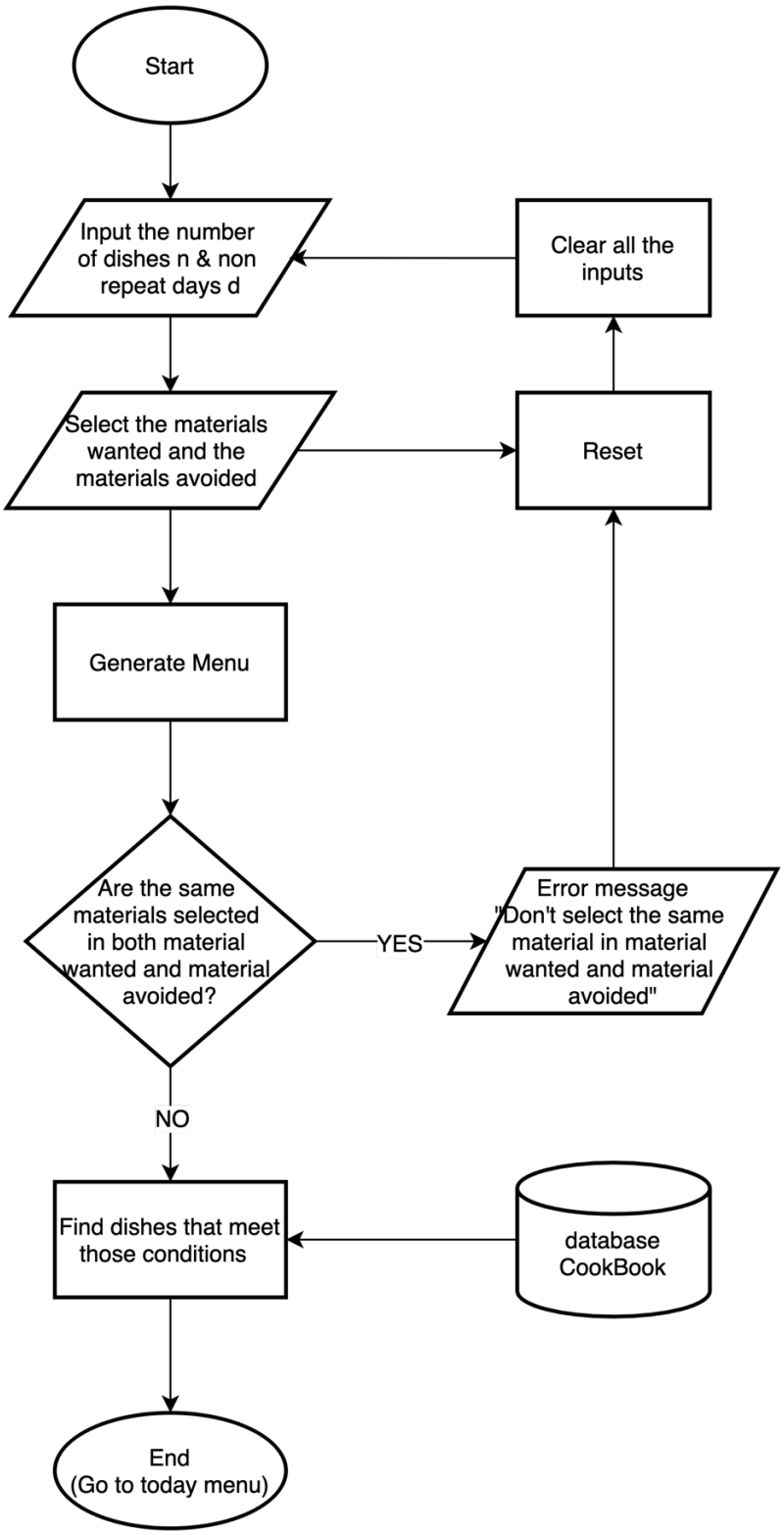
Flowchart for main window

This flowchart shows the options in the main window. There are three buttons in this window, the program jumps to the corresponding function by clicking each button.



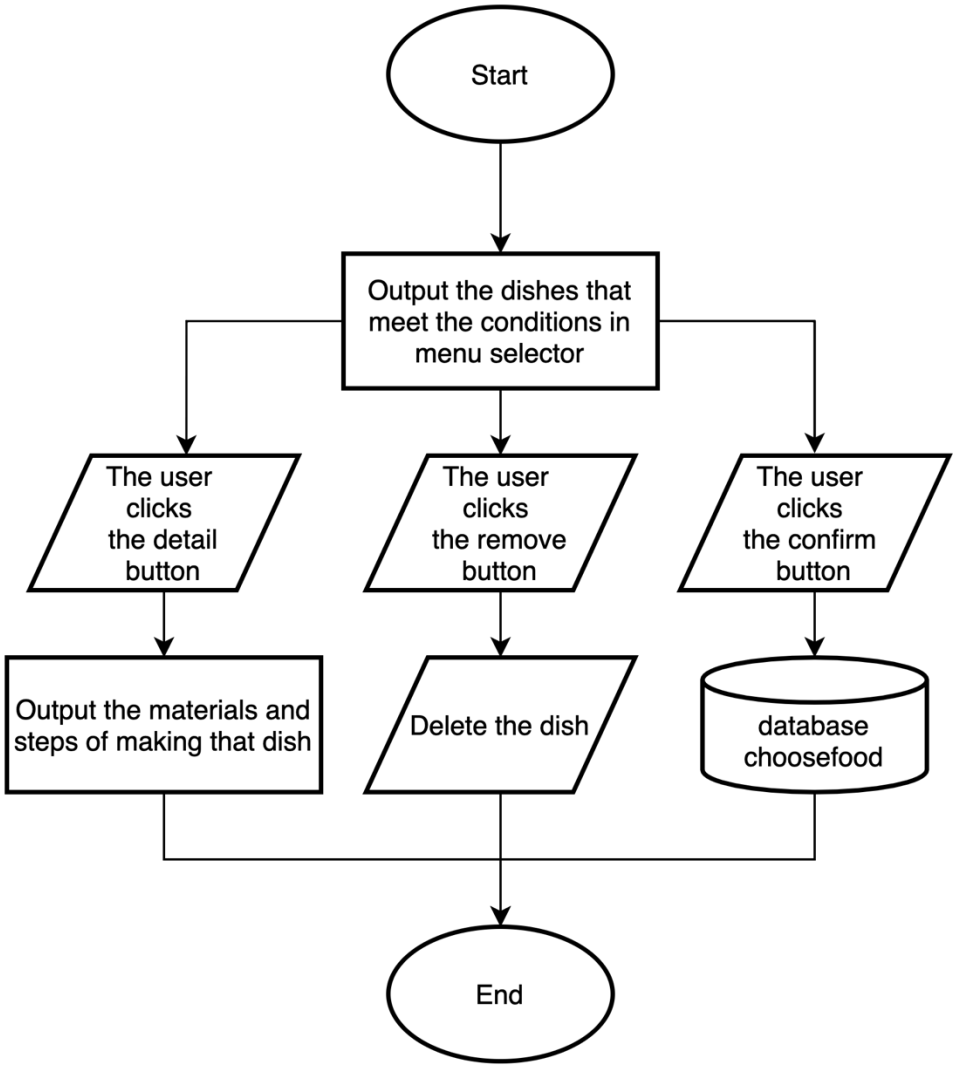
Flowchart for menu selector window

This flowchart shows the process of the menu selector to generate the menu based on the conditions that are input by the user. If the same materials are selected in materials wanted and materials avoided, an error message will display.



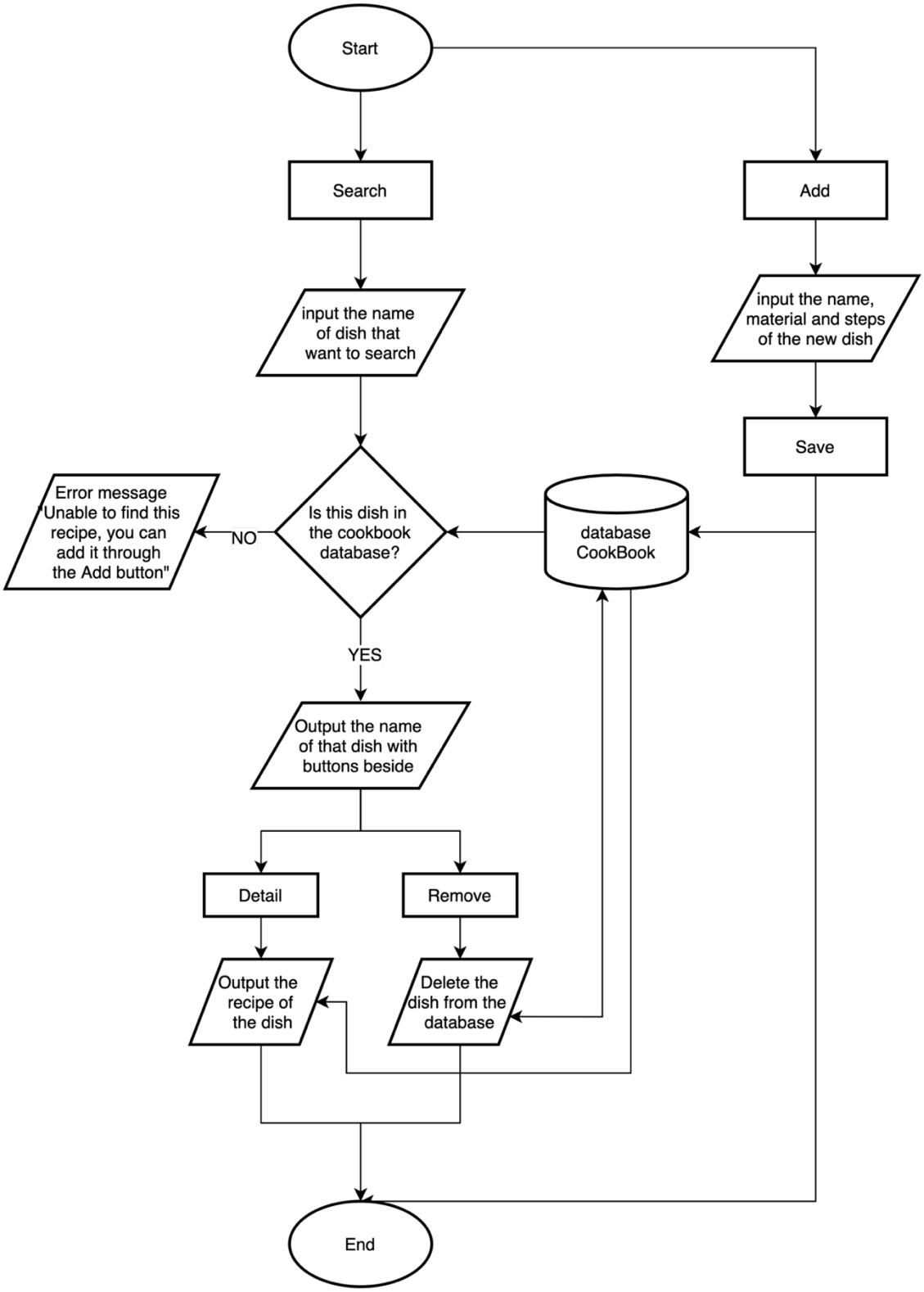
Flowchart for today menu window

This flowchart shows the flow of different operations on generated menus, including displaying details, deleting names and importing menus into the database.



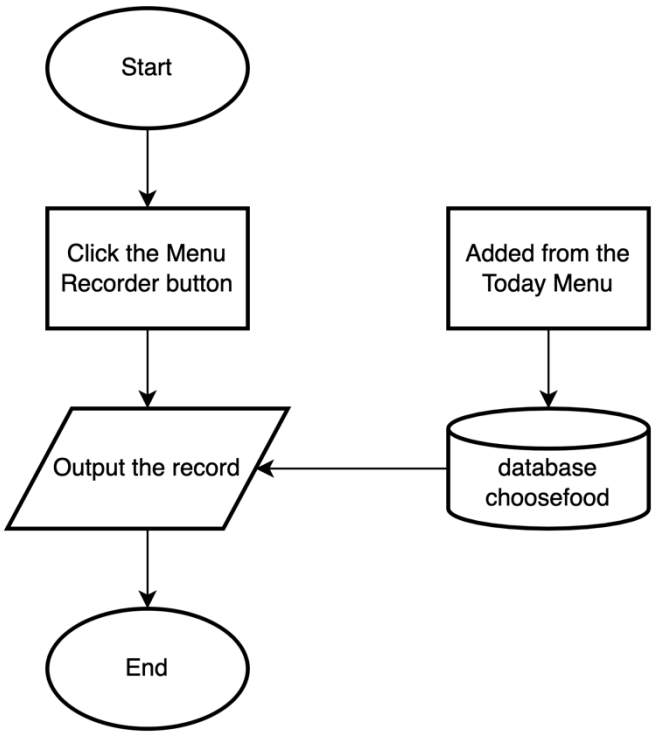
Flowchart for Recipe Searcher window & Add Recipe window

This flowchart shows how the program gets the corresponding recipe from the database based on the user's input and presents it to the user.



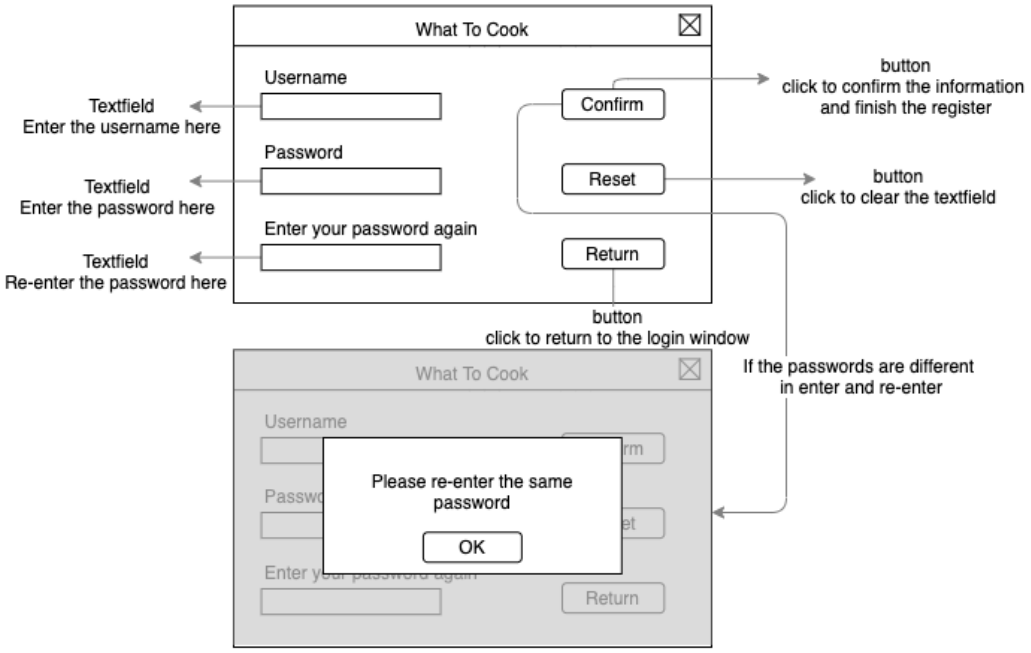
Flowchart for Menu Recorder

This flowchart shows how the program gets the record of generated menu from the database and displays it in the menu recorder. While the records are added into the database in the today menu window.



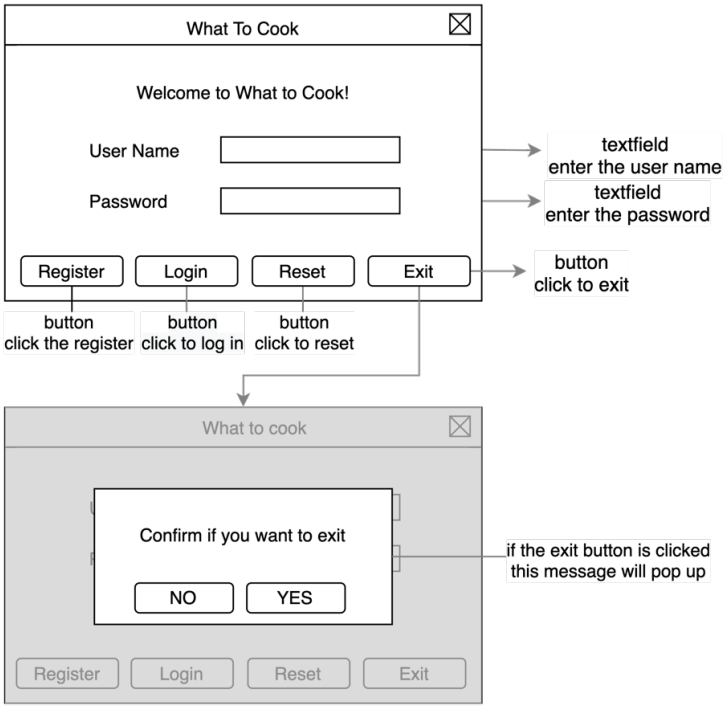
GUI for register window

The register window is used to sign up for an account and add the username & password into the database.



GUI for login window

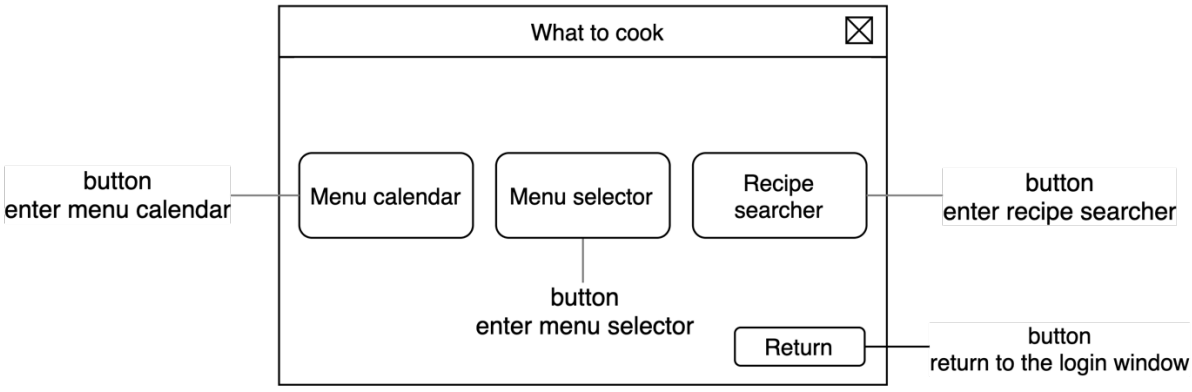
The login window is used to login in the program. If the username or password is wrong, it will ask the user to re-enter. If the user clicks the exit, a message will pop up which ask the user to confirm.



GUI of main window

The main window shows three main functions of this program, which are menu calendar, menu selector and recipe searcher.

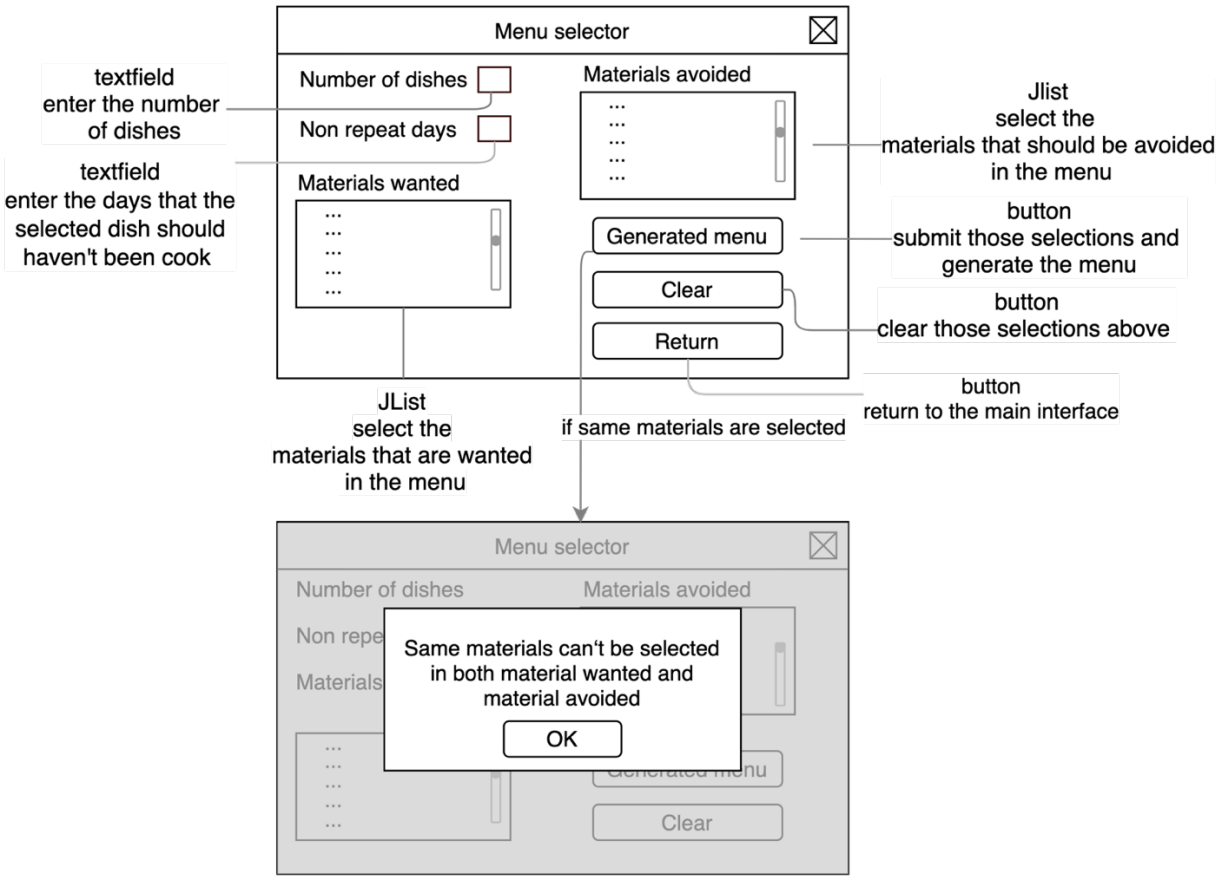
The user can enter each of these functions by clicking the buttons.



GUI of Menu Selector

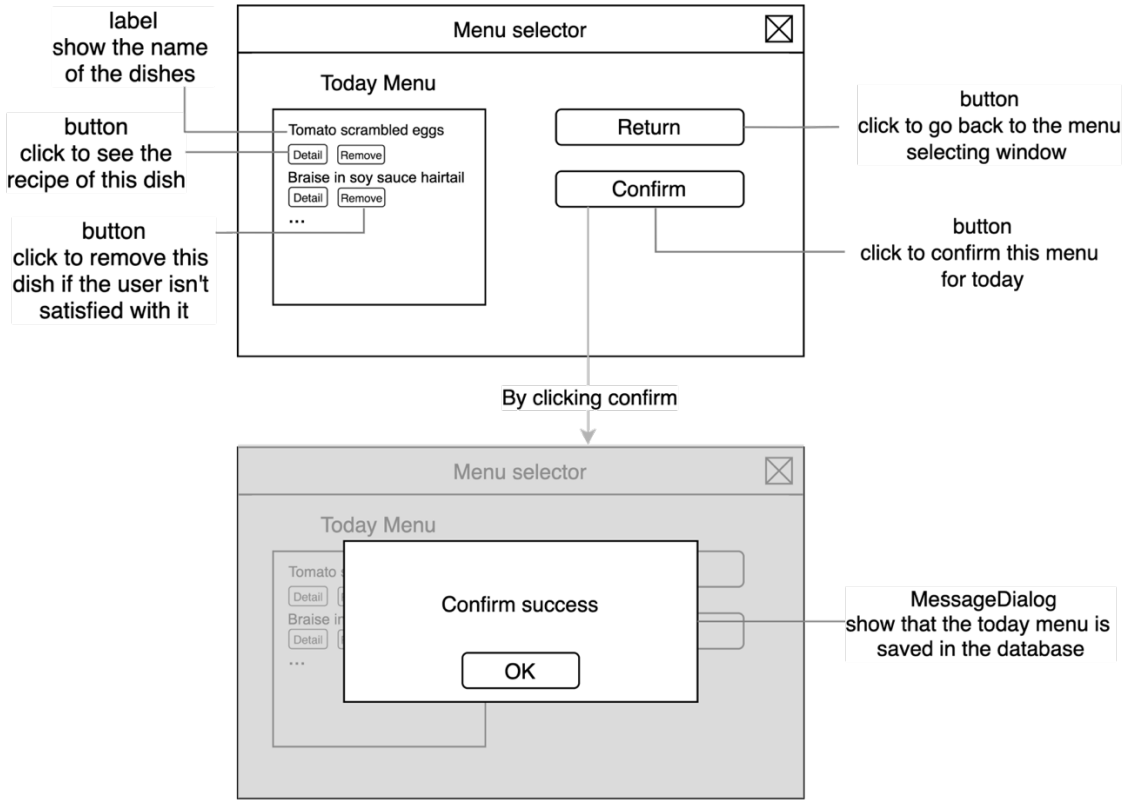
The menu selector interface allows the user to enter the requirements to generate the menu.

If same materials are selected in materials wanted and materials avoided, an error message will pop up.



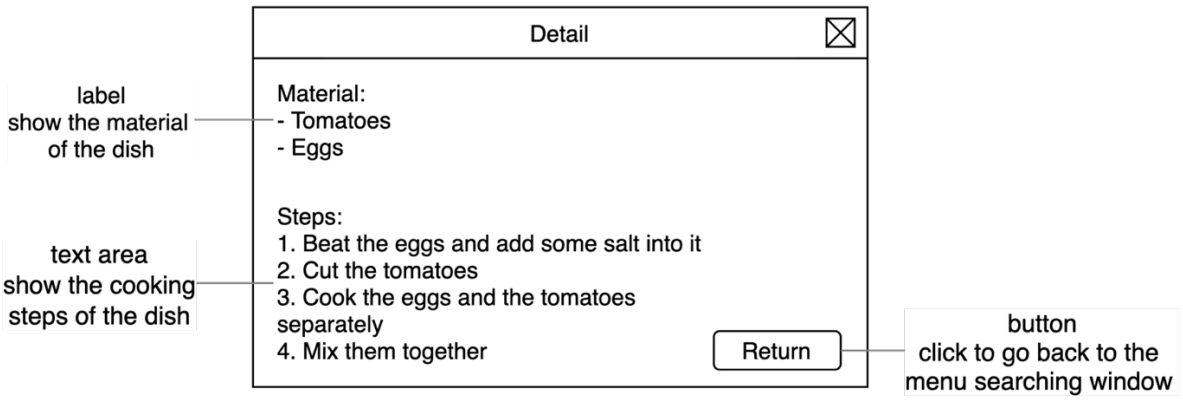
GUI of Today Menu

Today menu shows the menu that is generated based on the choices that selected in menu selector before. The user can click recipe to see the methods of making each dish. The remove button allows the user to delete this dish. If the user click confirm, the final today menu will be stored into the database as well as the menu recorder.



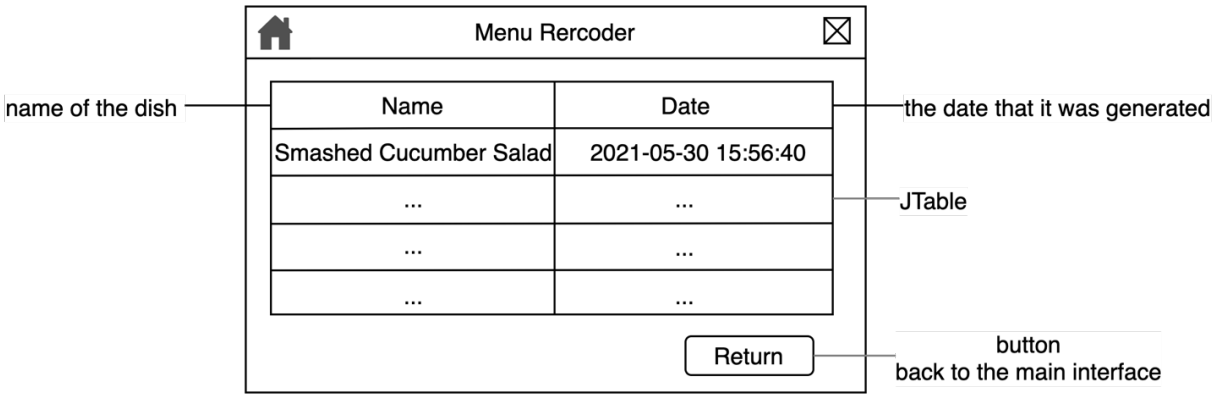
GUI of Recipe Detail

This window will be shown if the user click recipe in the today menu interface. It shows the material and steps of making that dish.



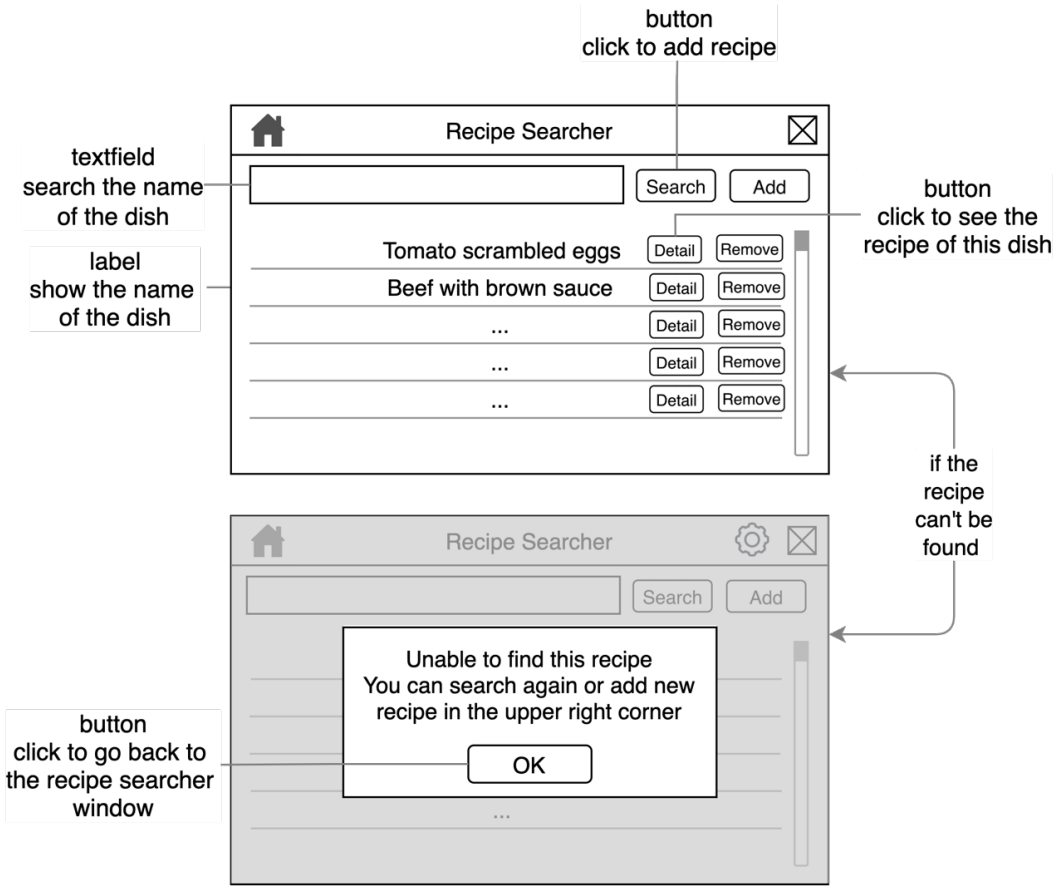
GUI of Menu Recorder

Menu Recorder records previously generated menus. It will be refreshed if the user generate new menu in the Menu Selector.



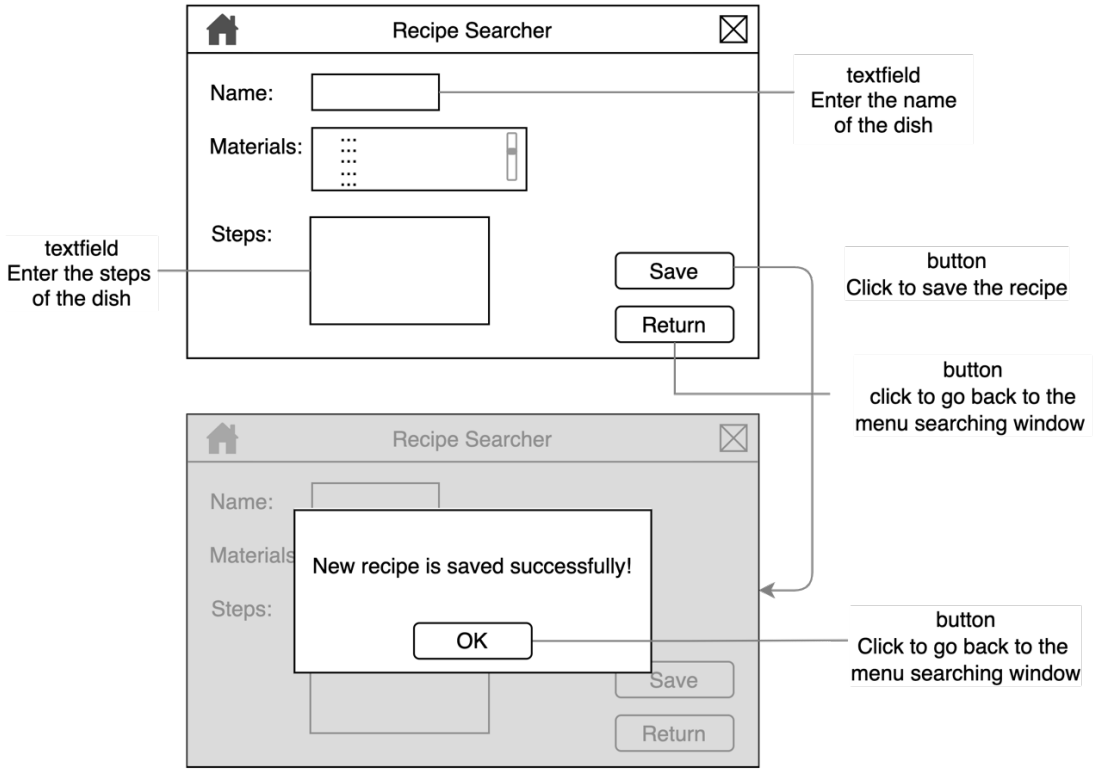
GUI of Recipe Searcher

User can search different dishes, click “detail” to read the recipe (refer to the GUI of Recipe Detail) and use “remove” to delete the dish from the database. If the dish can’t be found, a prompt window will pop up, showing that the user can use the “add” button to add new dish by himself.



GUI of Add Recipe into Recipe Searcher

User can add new recipe through this interface by entering the name and steps of the dish as well as the materials needed.



Database table

Register

Store the account of the users

Field name	Data type	Description
Username	String	/
Password	String	/

Cook Book

Store the information of each dish

Field name	Data type	Description
ID	Integer	The id of the dish
Name	String	The name of the dish
Material	String	The materials needed to cook this dish
Steps	String	The steps to make this dish

Food material

Store the list of the materials that will be shown in the JList of material wanted and material avoided

Field name	Data type	Description
ID	Integer	The id of the material
Name	String	The name of the material

Choose food

Store the record of the dishes that were generated

Field name	Data type	Description
ID	Integer	The id of the dish
Book id	Integer	The id of the dish in the database cook book
Create time	Datetime	The time that the dishes were generated

Test Plan

Test Type	Nature of test	Expected Result
Success criteria-1 Program requires user to login or register before entering the program, er-ror message will display if there's wrong input.	Unit test Validation	After register, the account will be stored into the database. The user can use the account in the database to login to the program. If the input is wrong, error message displays.
Success criteria-2 Program allows the users to input the materials that they want and the ones they don't want. The menu will be generated based on those materials. Error message will display if same materials are selected in both "material wanted" and "material avoided".	Unit test	The user can select the material wanted and material avoided. The final menu will include the dishes with the materials wanted but exclude the dishes with the materials avoided.
Success criteria-3 Program allows the users to input the number of dishes they want and out-put a menu with the corresponding number of dishes.	Unit test	The user can input the number of dishes and the program will output a menu with the number of dishes fit the one that the user input.
Success criteria-4 The program should make sure that the dish has not been generated during the previous n times while using the program, while the number of n is decided by the user.	Unit test	The user can input the times she doesn't want the dishes to be repeated. The program will check if the final menu includes the dishes that have been cooked in the past n days. if there are, these dishes will be deleted.
Success creteria-5 Error message will display if no menu can be generated.	Functional test Unit test	Error message displays if the program can't generate a menu based on the condition

Success creteria-6 The program should show the cooking method for each dish after the menu is generated.	Unit test	The cooking method of each dish should be stored in the database. The user can click a detail button to get the cooking method of each dish which is the same with the one in the database.
Success creteria-7 The unwanted dish can be removed from the menu after the menu is gen-erated.	Unit test	After one dish is removed, the final menu that stored in the database shouldn't include this dish.
Success creteria-8 The final menu will be stored into a database.	Unit test	The final menu can be stored into the database choosefood, so that the record in the database should fits the menu generated in the program.
Success creteria-9 The program should allow the user to search for the recipe of a dish, error message will display if no dish can be found	Unit test Functional test	The user can search dishes by using the name of the dish, while the program will look it up in the database cookbook. If it can't be found, there will be an error message.
Success creteria-10 The program should allow the user to delete unwanted recipe in the database.	Unit test	The user can delete the dish. After delete, the dish should be removed from the database.
Success creteria-11 The program should include an input which allows the user to add recipe of dishes by herself.	Unit test	The user can add recipes by using a button add. After the recipe is added, it should be store into the database and can be found in the menusearcher.
Success creteria-12 The program will have a recorder that shows what had been cooked before.	Unit test	The recorder shows the record of the previous menu while the record is the same to the one in the database cookbook
The whole program can run successfully	Integration test	The whole program can run successfully
The user can use the program smoothly	User acceptance test	The user is satisfied with the program.