IIKH Project - Team 1

A. Intro

### project title

TEAM1\_IIKH\_Project

### list of team members

Class 2 Team 1

고지흔 20196014, 음호준 20150736,

임동현 20183037, 이강민 20195517,

김유진 20203838, 권순욱 20212181

### presentation speaker name

이강민

### brief project description(summary)

The goal of this project is to create a program that allows users to efficiently manage recipes and plans. Also, the program can print the results in a document.

The program provides users with the function to add, delete, change recipes and output all recipes to help users manage recipes.

B. Compile method

-Recommended Operating System : Windows10(x64)

-Compiled by Microsoft® C/C++ Compiler(**Microsoft Visual C++** (**MSVC**) 14.34)

-Required Visual Studio version: 2022

-How to compile:

1. open IIKH.sln file in IIKH\_src directory with Visual Studio2022
2. before run build, make sure your following things
   1. build configuration is ISO C++17 Standard
   2. set build options to “Release” and for “x64”
   3. Windows SDK version 10.0.0(latest)
3. click “build” tab at left up side on your Visual Studio application window
4. click “solution build” option
5. check the successful build message on output window

-How to execute:

1. go to “x64” directory in the “IIKH\_src” directory
2. go to “Release” directory in “x64” directory
3. you will find execution file named “IIKH”
4. open that file.  
   (If you don’t have txt files, IIKH will make one for you  
   but if you want to use existing files, please locate the txt files in the same directory with IIKH.exe and name it as “DB\_Recipe.txt” and “DB\_Plans.txt”)

C. Functionality

[1] Recipe Manager

Go to the recipe manager window

[2] Plan Manager

Go to the plan manager window

[3] Exit

Exit program

[1] Add Recipe

Add a new recipe to our Recipe Database.

[2] Delete Recipe

First, the program shows a list of all the recipes in the Recipe database.

Delete existing recipe selected by the user from our Recipe Database.

[3] Revise Recipe

First, the program shows a list of all the recipes in the Recipe database.

Users can revise the selected recipe.

[4] Search Recipe by Ingredient

Users can search the recipe using the ingredient.

Program checks whether the ingredient selected by the user is in the database.

[5] Search Recipe by Recipe Name

Users can search the recipe using the recipe name.

Program checks whether the recipe name selected by the user is in the database.

[6] Show All Recipe

All recipes stored in the recipe database are displayed to the user one by one.

[0] Back To Menu

Return to the main window

(Our PlanManager automatically sort currently saved plans in Order of date(Ascending))

[1] Add Plan

Add a new plan to our Plan Database.

[2] Delete Plan

First, the program shows a list of all the plans in the Plan database.

Delete existing plan selected by the user from our Plan Database.

[3] Show All Plan

All plans stored in the Plan database are displayed to the user one by one.

[4] Show Plans By Period

Users can check the plan for a specific period(year,month,week,day).

Also, users can print the integrated grocery list for a specific period.

[5] Search Plans By Meal Type

Users can search for plans by meal type(Breakfast, Launch, Dinner).

And show the search results.

[0] Back To Menu

Return to the main window

D. Implementation method

Before we started the project, we had a meeting and made the big framework of the project based on the IIKH of the book. However, as the project was carried out in a bottom-up manner with the IIKH that the members thought was slightly different, some classes needed to be implemented additionally and some classes were not used, so they needed to be changed. We kept considering what is good software and how the classes are organized like relation between each class and data structure they should have. For example, while wording on the project the database and parsing classes of the plan were required, so they were added later in the project. Inputmanager, which receives file input, was deleted in the middle of the project, and the function of inputmanager was divided into planmanager and database classes. We continued to hold additional meetings to align the framework of the project and gradually integrated IIKH's appearance into one by continuously exchanging and modifying code using github. As a result, the IIKH implementation was successfully completed.

[1] Our team had to solve many problems to implement IIKH. There was also the problem of storing file types in the database. At first, we tried to set the delimiter as ',' or tab input, but it was hard to save it as we wanted, so we set the delimiter as '/' in the database.

[2] Our team aimed to make IIKH with the exe file of the window. However, it was difficult to build the project because each computer had a different operating system and code editor.

[3] In our team's IIKH, meal class needs to know the RecipeDatabase. But it was pointed out that meal class has to have multiple instances and depending on RecipeDatabase which is a heavy object is not a good program. To solve this problem, our team implemented it in a Singleton pattern and used the Recipe Database class as a global variable through using static variables and functions.

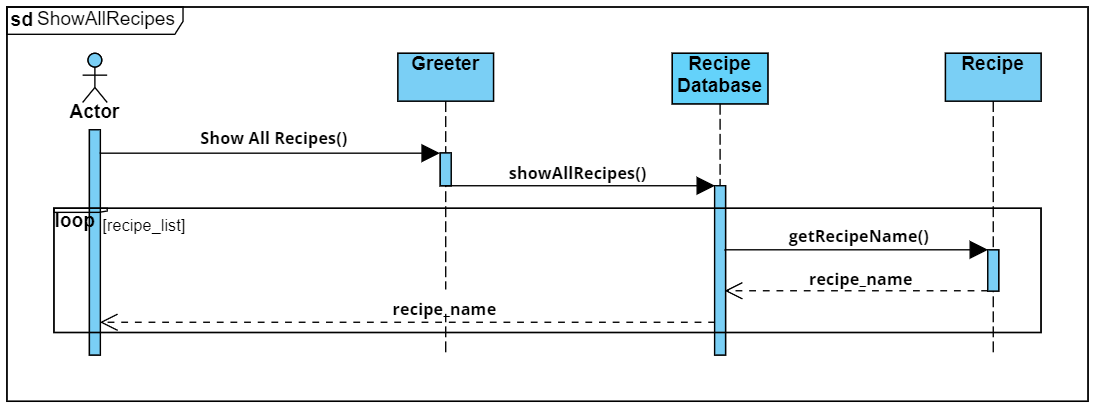
E. SW Design Result

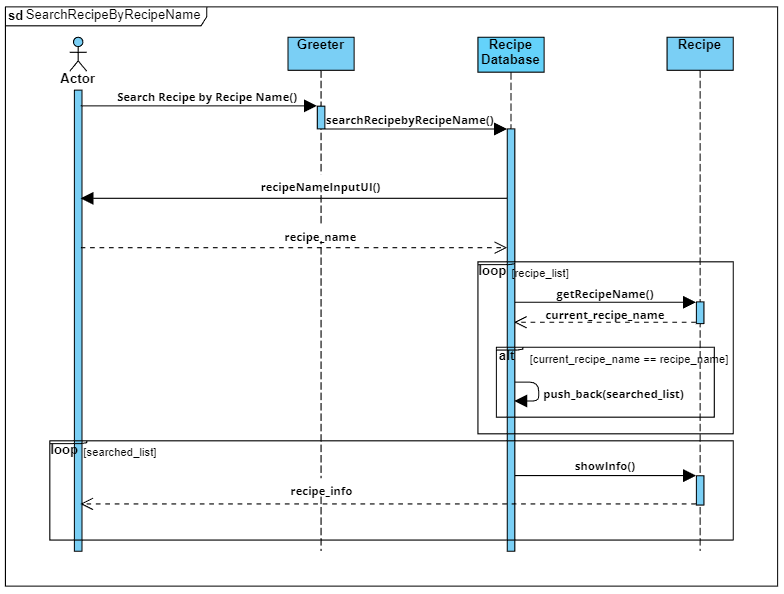
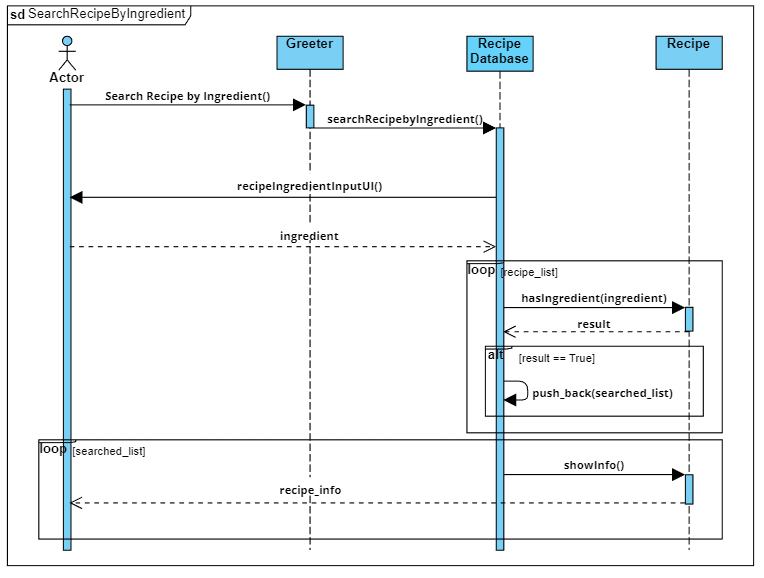
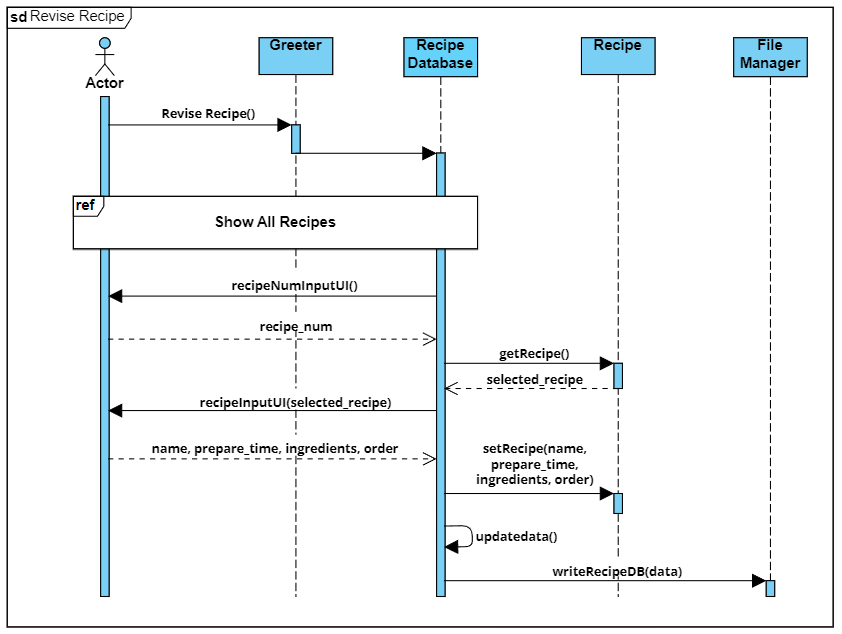
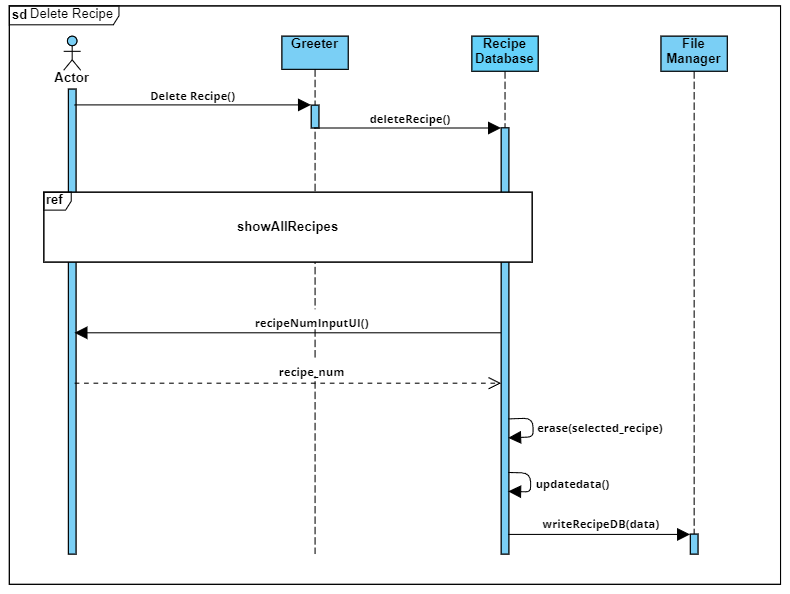
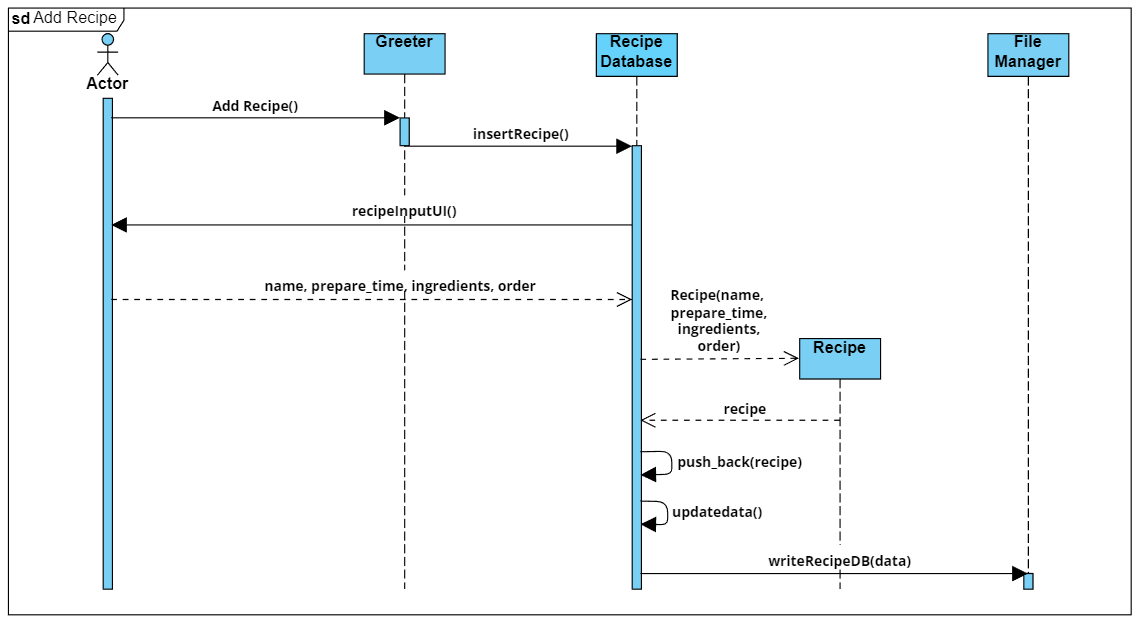
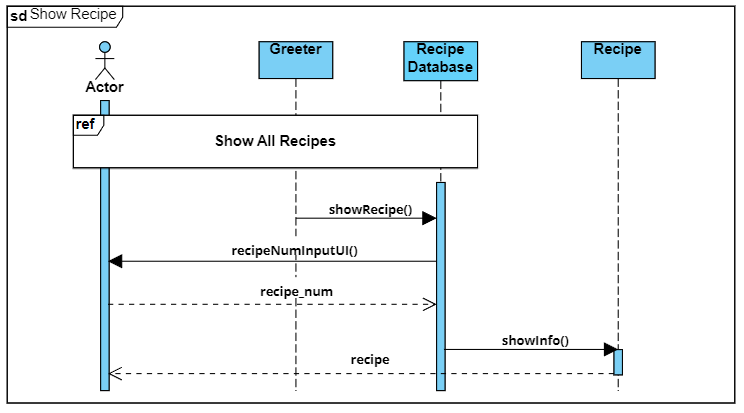
# <Class diagram>

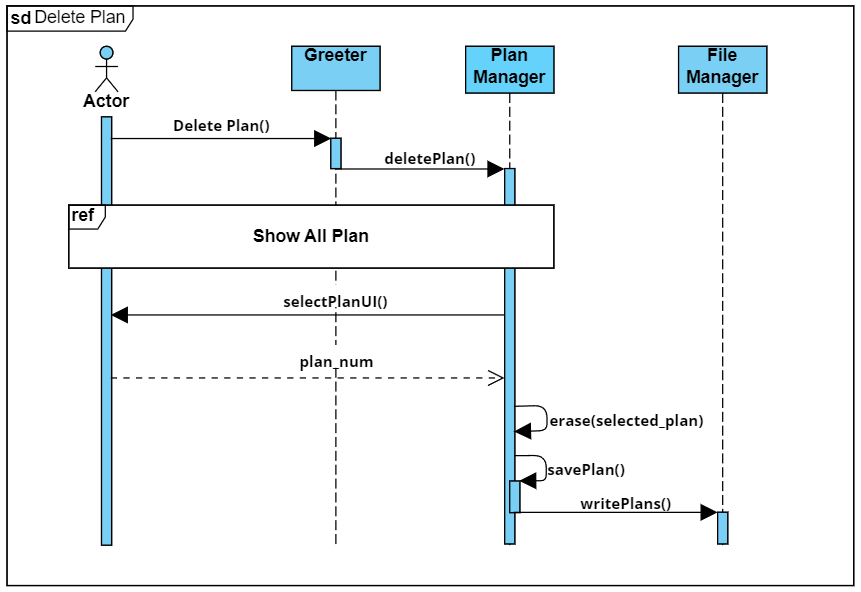
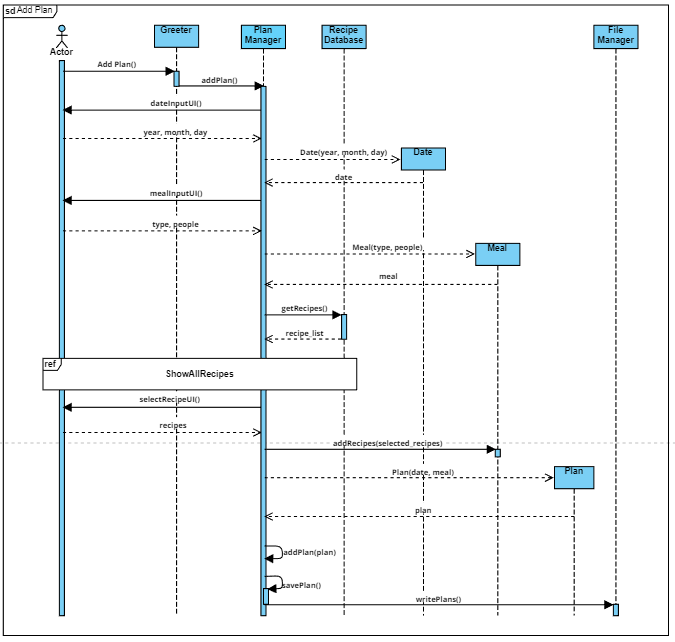
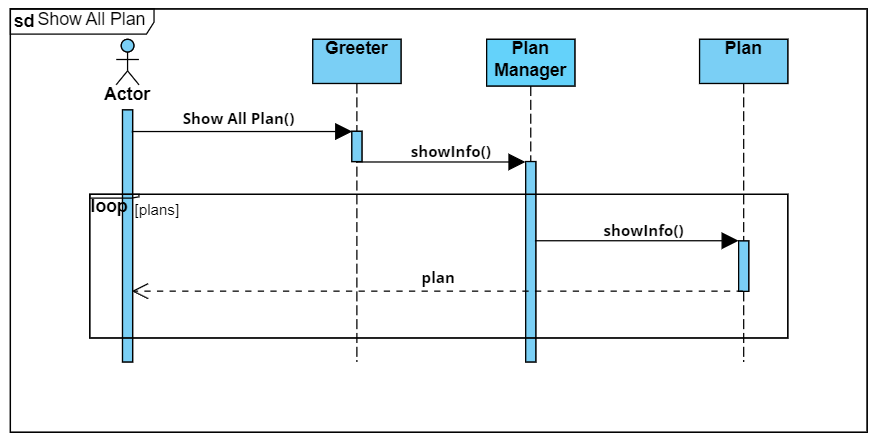
# 

You can check the image in detail here. : <https://ibb.co/jWV3Q7G>

# <Sequence diagram>







F. Test (execution result)

## <This is the menu window when execution. >

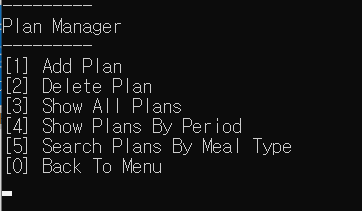
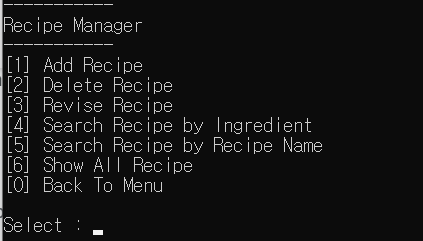


-[1] Recipe Manager

-[2] Plan Manager

-[3] Exit

## <Each screen when selected>



(When [0] Back to Menu is selected, return to the menu window.)

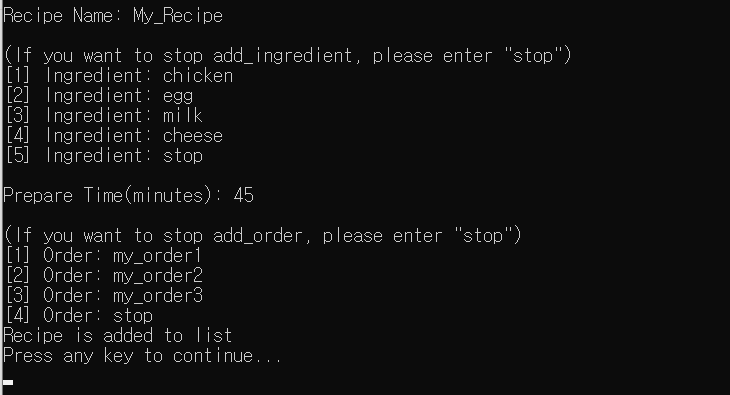
## [1] Recipe Manager

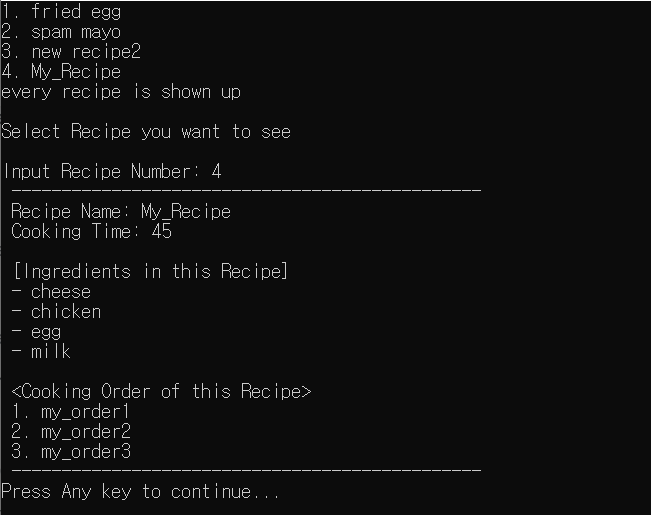
[1] Add Recipe

- When selecting [1] Add Recipe, you can enter the Recipe Name, Ingredients, Prepare time, and orders to save.

-Add “My\_Recipe” information.

- You can check the stored recipe through [6] Show All Recipe.



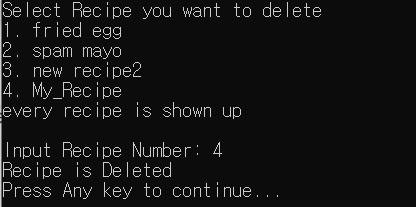


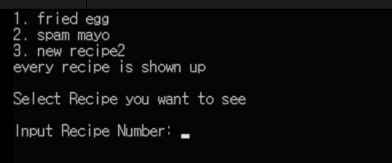
[2] Delete Recipe

- You can remove the recipe.

- Delete “My\_Recipe” information.

- You can see that it has been removed from [6] Show All Recipe.



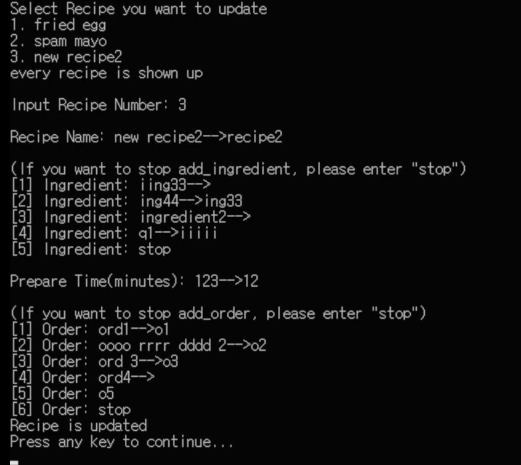


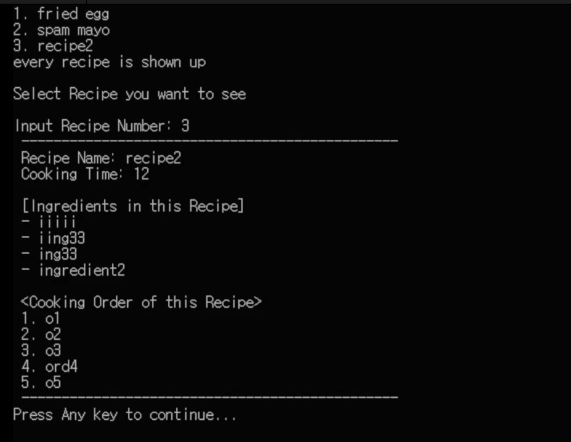
[3] Revise Recipe

- You can revise the name, element, time, and order of the recipe.

- Revise “3. new recipe” information

- You can check that the recipe has been revised through [6] Show All Recipe.

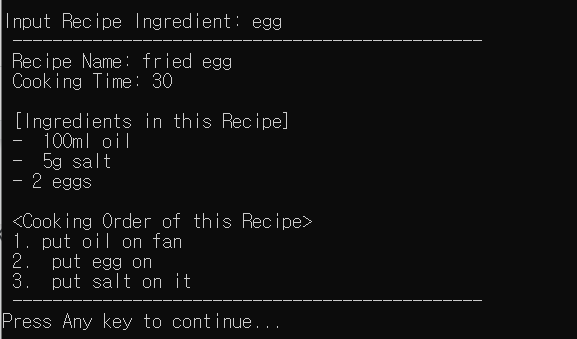


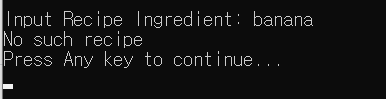


[4] Search Recipe by Ingredient

- Users can search for recipes by ingredient.

- We also handled cases in which we received a recipe that did not exist.

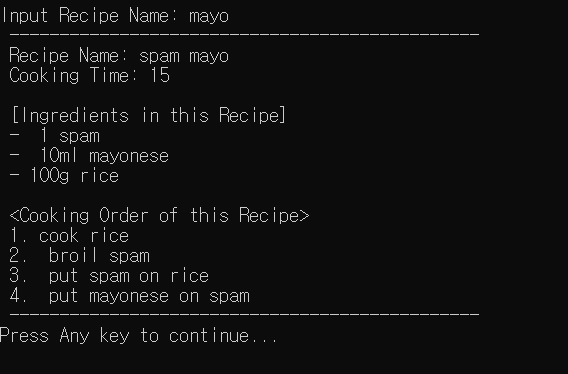


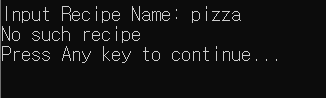


[5] Search Recipe by Recipe Name

- Users can search for recipes by recipe name.

- We also handled cases in which we received a recipe that did not exist.

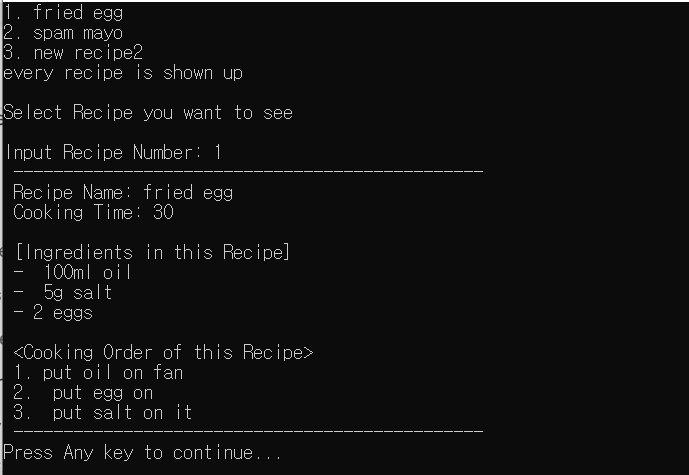




[6] Show All Recipe

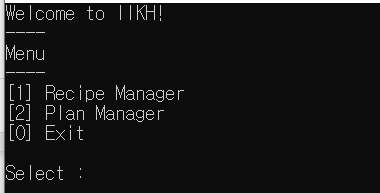
- Show all currently saved recipes.

- If you press the recipe number, it shows the details



[0] Back To Menu

- return to the first menu window

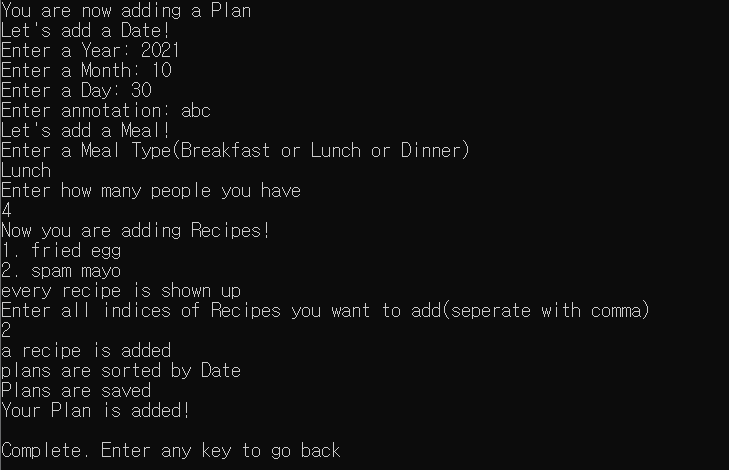


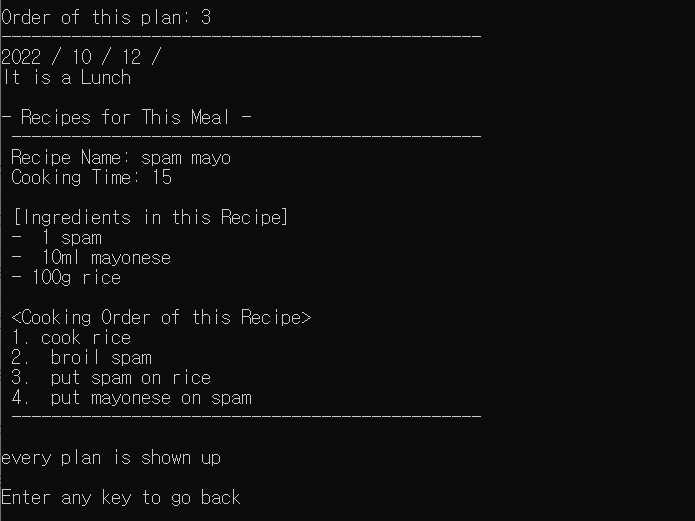
[2] Plan Manager

[1] Add Plan

-When selecting [1]Add Plan, enter Date, Meal type, and how many people you want to eat with, then import the recipe from RecipeDB and add it to the plan.

- You can check the stored plan through [3] Show All Plan.

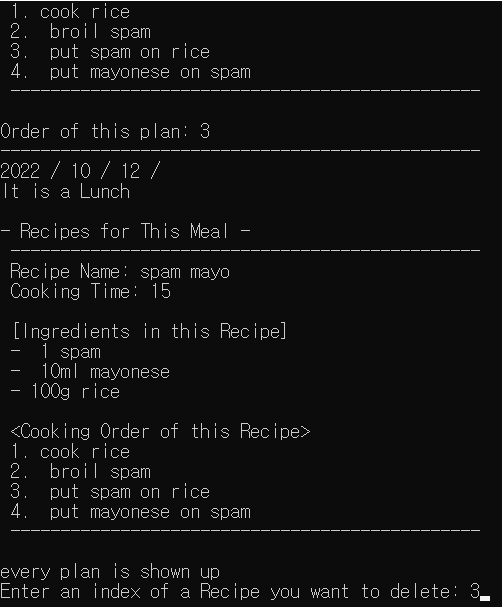




[2] Delete Plan

-When [2] Delete Plan is selected, the IIKH shows all the plans and can delete the plan.

-You can check the deleted plan through [3] Show All Plan.



[3] Show All Plan

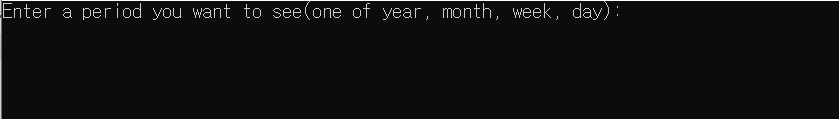
-[3]Show All Plan shows all currently saved plans

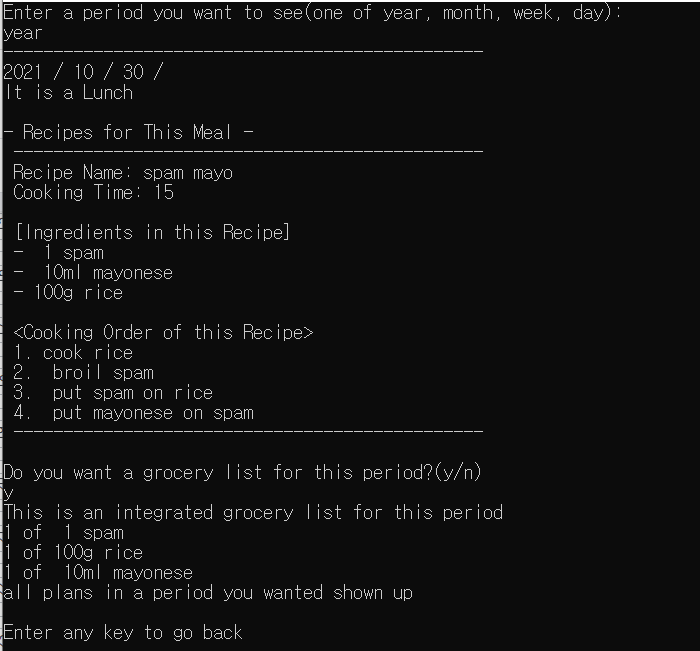


[4] Show Plans By Period

- Users can check the plan for a specific period.

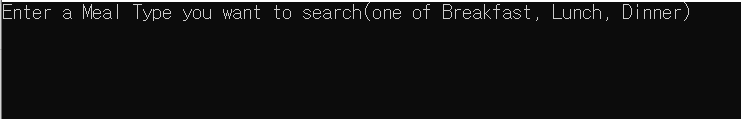
- Users can check the entire ingredient of the selected plan.

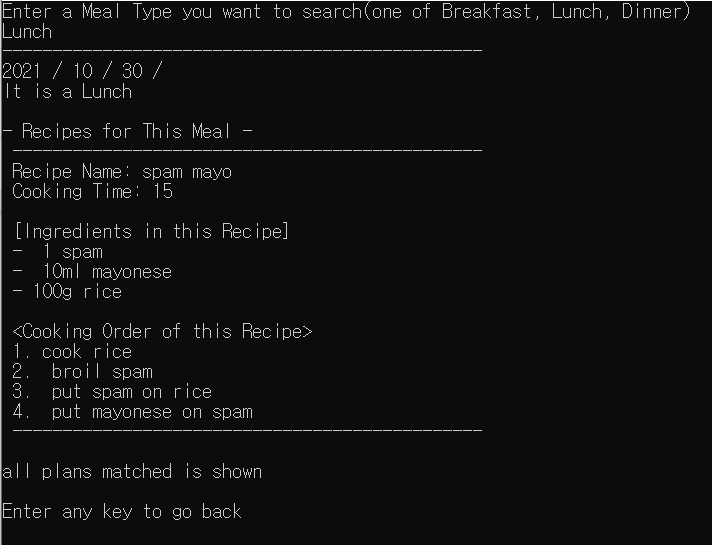




[5] Search Plans By Meal Type

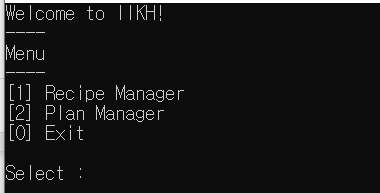
-Users can search for plans by meal type.





[0] Back To Menu

-return to the first menu window



G. OOP Concepts & Thoughts

고지흔

IIKH was my first project in my life. And I came across c++ for the first time. Because of this reason, I didn't know what to do at the start of a project. It was really hard to work on this IIKH project. But thanks to my team, I got the direction. I was able to follow the project well because the team members adjusted my role properly.

There were a lot of new concepts or a concept that has never been applied that I had to study, such as objects, classes, split implementations, STL, github usage, and so on. It was an opportunity for me to understand these concepts by looking at the source code of IIKH. IIKH was an opportunity for me to see how concepts like ADT, class & object, Encapsulation, Polymorphism, and dynamic binding that I learned in class were applied.

I also understood the importance of specific direction in team projects and the importance of endless meetings. In fact, our team's direction continued to change little by little as classes were created and disappeared through meetings. The part I was in charge of was the plan side of the Inputmanager class and Date class. At first, it was difficult to create a class without knowing what variables to receive, but later, through meetings, I understood exactly what variables to receive and created a class based on it.

Through this IIKH project and through the consideration of team members, I have gained a lot of new knowledge, and based on this, I think I can participate more actively in the next project.

음호준

I tried to keep the basics of OOP.

First of all, I tried to classify the role and responsibility of each component.

I examined the behavior and property, then clarified the responsibility of each component.

Next, I made each component detail class.

I was careful to not make each class too big or too small.

Also, I would like to keep the Single Responsibility Principle, but it was not easy.

I hide information which is needed only in a Class using private members.

And exposed the interface which is needed from outside of a Class using public members.

When a Class has a dependency to another Class, I considered that it should have the dependency and if it should, then there is anything I can do to lower the dependency.

So, what I objected to was loosely coupled Classes.

Further, I thought about the behavior and responsibility of a Class.

For the case of Class which does not need to be instantiated many times, and has the high cost for the operations, I applied the Singleton design pattern to make better program composition.

To make a Class dependent on interface, I made an abstract class that has virtual a function to be implemented. So the Classes inherit(extend) the abstract class should implement the function. The inheritance means relation of dependency between classes.

What I taught in this project is, it is more difficult than I thought to consider the requirements enough and devise a good program design from the consideration.

At first, as we didn’t consider enough about the details of the given requirements, we had to change our design of IIKH. Considerations like this took more time than implementing a real program. We are a team, so we deal with many different thoughts each teammate has.

It also took a long time to unify the thoughts.

임동현

With the bottom-up development process in mind from the first meeting, we started development from the most basic ADT, went up to the object that manages them, and ended the project by linking the Greeter object and the sub-objects that interact directly with the user.

The part I was responsible for was the RecipeDatabase part. When designing this object, I first divide the functions and data that only works inside the class and the interface that appears outside to hide information.

In the recipe input method, all functions are the same except for the presence of the existing recipe, so I used dynamic binding to maintain the consistency of the name of methods.

In addition, I tried to separate the UI part where it interacted with the user and the Model part where the data was processed by applying the MVC pattern , but I didn't have enough time to complete it.

The most difficult part of the project was communication with the team members. Because all team members had different levels, it was difficult to divide roles in a balanced manner, and because the background knowledge was also different, it was difficult to understand and persuade other team members in designing each function.

However, I asked relatively inexperienced team members to design basic ADTs, and high-skilled team members to develop important logics, and we overcame the second problem by continuing to agree on each other's opinion through continuous meetings.

Through these activities, I became a little more familiar with the project as a team.

이강민

While completing the plan file, I had trouble fully understanding encapsulation. Since object oriented programming doesn’t allow private data to be accessed from outside, it was troublesome for me to know how to access and show it. for example, when i needed to access the data, at first I had no idea how to get the data and show it to the user and save the data for further use. But the solution was quite simple i just had to save it in another variable and print it using the outstream function. This experience made me more familiar with the OOP but also made me realize that I have a long way ahead on becoming a developer.

김유진

It was a difficult project for me when I first came across the concept of object orientation. I learned that each class represents the role of the components that make up the program. In class, I learned that the interface is exposed to public, but the basic information that does not have to be exposed is hidden privately, and I tried to organize the code like that.

I was assigned each class through the meeting and made it, but I was assigned an ingredient class. At first, I received the name and weight and saved it, so I declared a string variable and an integer variable. However, through several meetings, declaring an ingredient class separately requires a more complex implementation, so we thought about how to receive the ingredient's name and weight as a string variable at once. And instead of creating a separate class, we put it in another class so that the code can be implemented more conveniently.

I worked on a project with many people for the first time in college. It was my first time, so I didn't know much about what to do with the team project, but it was a time to meet good team members and learn a lot. I'm just grateful to the team members who helped me a lot because I'm not good enough.

권순욱

The part I was in charge of was creating the recipe class. I was able to directly use the basic ADT concepts learned from class. To do this, I created a header file corresponding to the interface to demonstrate what recipe elements are in a recipe and what functions they can perform. And I defined the detailed operation method in the source file corresponding to the implementation. With the help of team members, I used the operator overloading in c++ to increase the utility of the recipe class.

This was my first team project. When I first heard about the project, the project was terrifying in that it had to combine more than 20 files and successfully execute instead of just running one file. However, I was able to finish the project successfully by continuously collaborating with the team members. In the process, I felt that dealing with a lot of opinions of different people and making large choices were an important part of the programming project. And above all, I want to thank the hard work of my programmer colleagues.

The experience of successfully implementing a huge team project was really impressive. Through this project, it was a time to learn the team collaboration method in the programming field and the knowledge of OOP concepts.

H. Conclusion

We made a well-run kitchen aid IIKH. We started the project with a bottom-up method that is characteristic of OOP. We created individual classes and then linked them to make one project successful. This process required a lot of collaboration and discussion skills, and we were able to develop them while working on the project. Above all, it was an opportunity to learn and implement the characteristics of OOP.