

A.

- Project title: CAU-IKH
- List of team members:  
20182705 고주형, 20185784 김호성, 20182610 손희승, 20162874 이준협, 20142611 이하람
- Presentation speaker name: 이하람, 고주형
- Brief project description (summary)  
Efficient kitchen helper that is used for managing a lot of recipes, and planning daily meals like breakfast, lunch or dinner.  
By using our kitchen helper, you can SAVE/EDIT/DELETE/SEARCH for recipes and manage your meal plans.  
We also put ascii arts inside our program. Pretty UI makes program livelier and fresher. ☺

B.

- How to compile and execute
  - ① Change directory to Source Code Package Folder
  - ② Open the Solution File with Visual Studio 2019
  - ③ Compile (Ctrl-F5) and Execute
- And also an executable file is in Release Folder so you can directly execute our project file. (How to use executable file is in README.txt)
- System requirement for compilation and execution  
Target OS: Windows 7 / 10  
System Requirement: Same as Visual Studio 2019 system requirement

C. Description on functionality that was implemented in your SW system:

- Add recipe: Add new recipe to our Database.
- Edit recipe: Edit existing recipe from our Database.
- Delete recipe: Delete existing recipe from our Database.
- Search recipe (Print all recipe):  
Search existing recipe from our Database by recipe name.  
(구현 실패 시 삭제) Containing Search is supported(If you search "pie", all kinds of recipes whose name include substring "pie" will be searched. For Example, "apple pie", "raspberry pie" ...)
- Add meal plan: Add new meal plan.
- Print meal plan: Print meal plan's title and breakfast, lunch, dinner's menu.
- Pretty Ascii art: Maybe it will make user happy and make program fancier.

#### D. How you implemented (important implementation issues):

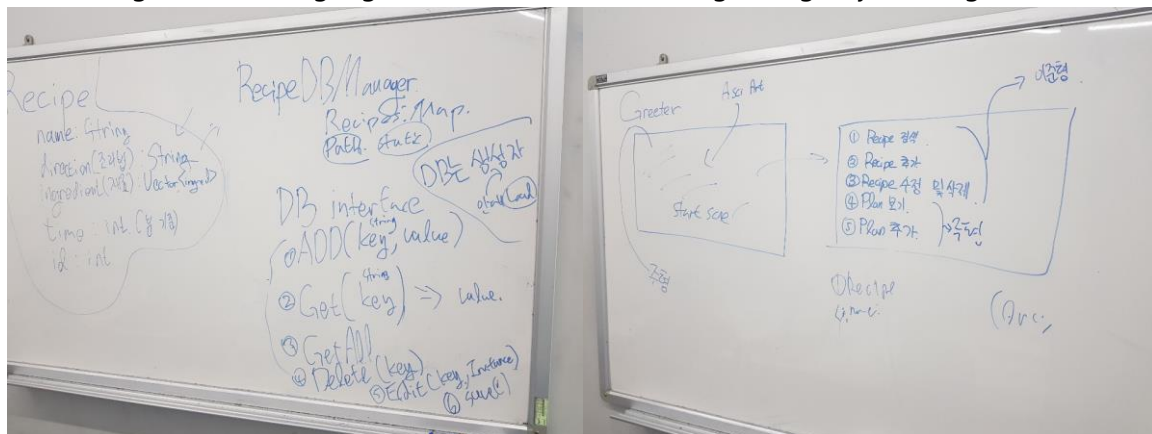
##### ● Issue01: Problem was ambiguous.

Due to ambiguous specification, we had to talk a lot about what this program, what is our target, what should we implement and how we'll implement this system. Though we had clarified our objective (what we'll implement). The problem was that the overall program design which is about how we'll implement this program, was chosen by just talking. There was no document or logs about what we talked. Also, after some coding we found out everyone was thinking slightly different design.

By that problem when we are talking about our program, we had many issues such as, Team Member\_A thought it is better to make Database Manager for all objects (recipeDBManager, planDBManager, mealDBManager). But other Team Member\_B thought it is better to make integrated Database Manager that can handle all of the objects. By this mismatched when someone coded a Database header, few team members could understand it.

Also, the Plan class was confusing. It was not described correctly and specification was made by just talking. For instance, our one team member thought date should be addressed in plan but other team member thought date will be addressed in meal class. Such kinds of situation led our design more complicated and finally became not understandable. So, we had an emergency meeting and unified all of our design. After that we could start programming our given part for each peacefully and confidently.

Below images are meeting logs taken when we were doing Emergency Meeting.

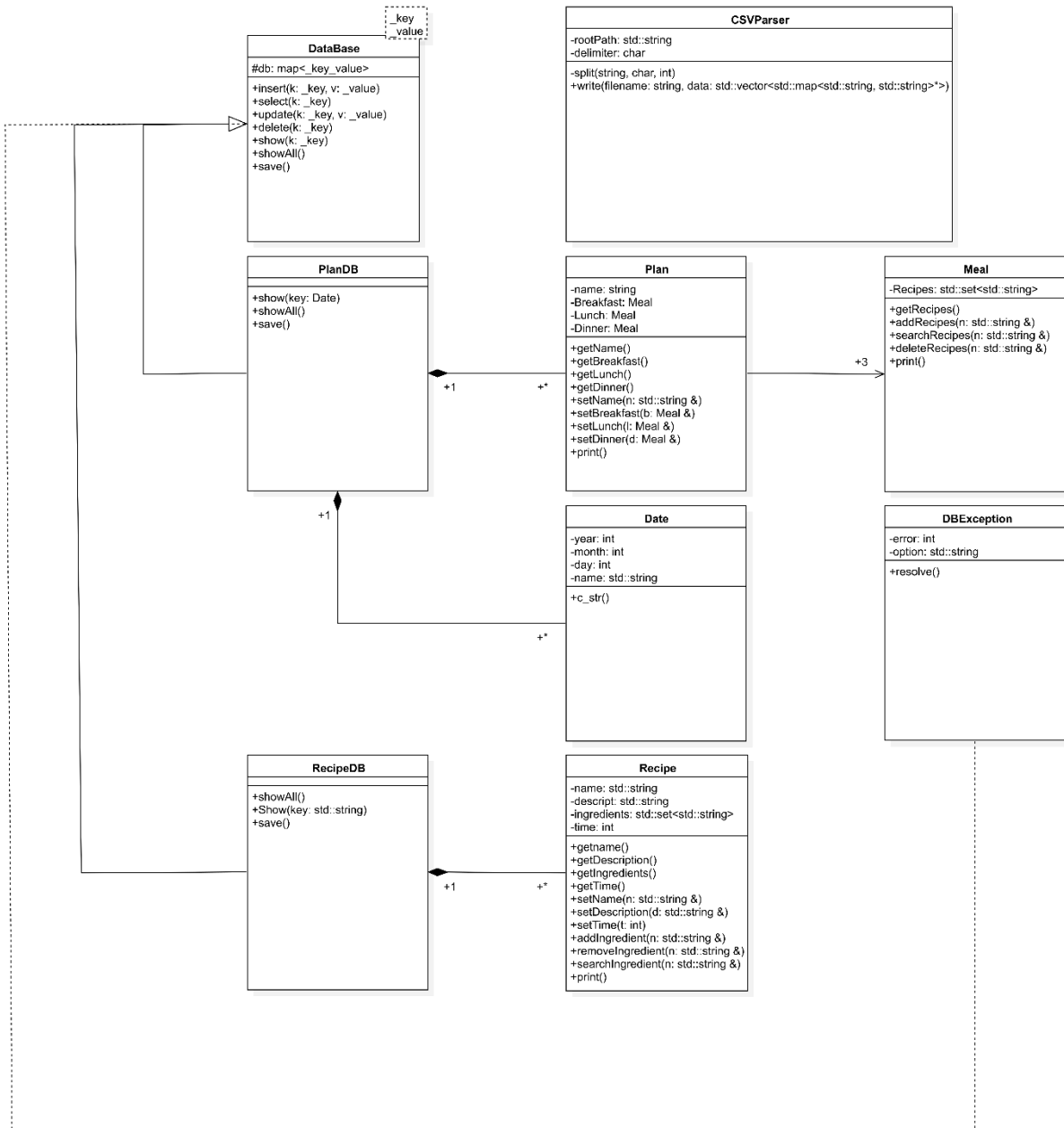


##### ● Issue02: Designing component and giving responsibilities to specific class.

We argued a lot about responsibilities. If a specific class controls every class, the dependency becomes high. This was bad. Bad to debug or understand or extend... as we learned.

For example, meal information was needed in everywhere. A Planner had to know meals for daily meal plan, Database Manager should contain every meal for saving, and meal should contain date for searching meal plan by date. After some coding meal class became too huge, and dependent. So we lowered the dependency of class by giving responsibility to each class and requesting to other class when other class' responsibility is need.

E. The result of SW system design [UML]:



(use screen capture)

Show that each function of the SW system is working correctly.

## 1. Start Scene

[illegible]

## 2. Adding Thigh Burger Recipe (Menu 2)

```
You will enter recipe information in [[ NAME => DESCRIPTION => TIME => INGREDIENTS ]] order
Name : 짜이번거
Description : 1. 계란 2개를 쪄후 허벅지 살을 담근다. 2. 허벅지 살에 튀김을 입힌다. 3. 닭고기를 튀긴다. 4. 빵 사이에 마요네즈와 튀긴 고기를 넣는다.
Cooking Time(min) : 20
Ingredients (If you want to stop enter "stop" ) : 뼈 없는 닭 허벅지 살
닭
계란
튀김
빵
마요네즈
가루
stop
```

### 3. Searching Thigh Burger Recipe that I just added (Meue 1)

```
You've selected [1] to search recipe
Search : 싸이버거
Recipe Name : 싸이버거
Ingredients :
- 달걀
- 마늘 가루
- 마요네즈
- 밀가루
- 뼈 없는 닭 허벅지 살
- 소시지
- 양파
- 양파 가루
- 계란
- 피클
- 햄버거 빵
- 치즈

Expected Time : 20
Direction :
- 1. 계란 2개를 푼후 허벅지 살을 담근다. 2. 허벅지 살에 튀김을 입힌다. 3. 닭고기를 튀긴다. 4. 빵 사이에 마요네즈와 튀긴 고기를 넣는다.
=====
Press anykey to continue
```

## 4. See All Recipes including Thigh Burger Recipes that I just added(Menu 3)

```

- 1 teaspoon salt
- 3 cups water
Expected Time : 720
Direction :
- 1. Combine water with salt ground black pepper oregano basil onion salt parsley garlic powder bay leaf and salad dressing mix in a saucepan. Stir well and bring to a boil.
- 2. Place roast in slow cooker and pour salad dressing mixture over the meat.
- 3. Cover and cook on Low for 10 to 12 hours or on High for 4 to 5 hours. When done remove bay leaf and shred meat with a fork.

Recipe Name : Spinach-Ricotta Quiche
Ingredients :
- 1 (8 ounce) package fresh spinach
- 1 tablespoon butter
- 1/2 (14 ounce) package pastry for 9-inch double-crust pie
- 1/2 teaspoon salt
- 1/3 cup finely chopped red onion
- 1/2 cup grated Parmigiano-Reggiano cheese
- 1/4 teaspoon ground black pepper
- 3/4 cup heavy cream
- 3/4 cup whole milk ricotta cheese
- 4 large eggs
Expected Time : 55
Direction :
- 1. Melt butter in a skillet over medium heat; add onion and cook for 1 minute. Add spinach and cook stirring occasionally until starting to wilt about 1 minute. Cover skillet and cook for 1 minute more allowing condensation to build inside the skillet. Remove from heat uncover and stir.
- 2. Preheat the oven to 375 degrees F (190 degrees C). Press pie pastry into a 9 12-inch deep-dish pie pan. Prick bottom and sides with a fork.
- 3. Bake crust in the preheated oven for 10 minutes. Remove from oven and set aside.
- 4. Combine eggs ricotta cheese cream Parmigiano Reggiano cheese basil salt and pepper in a blender or food processor. Blend until smooth.
- 5. Spread spinach and onion mixture evenly over the crust. Pour egg mixture on top.
- 6. Bake in the preheated oven until center of quiche is set and the top is lightly browned about 40 minutes. Allow to stand for 10 minutes before serving.

Recipe Name : Strawberry-Avocado-Mint Smoothie
Ingredients :
- 1 cup vanilla-flavored almond milk
- 1 small avocado/ cut into chunks
- 10 fresh mint leaves/ torn in half
- 2 (5.3 ounce) containers nonfat strawberry-flavored Greek yogurt (such as Dannon(R) Light & Fit)
- 2 cups frozen strawberries
Expected Time : 10
Direction :
- 1. Place mint strawberries yogurt almond milk and avocado into an electric blender and blend until smooth.

Recipe Name : 싸이버거
Ingredients :
- 토마토
- 마늘 가루
- 마늘 슬라이스
- 분쇄마늘
- 통깨
- 얇은 허벅지 살
- 식용유
- 양파
- 양파가루
- 피클
- 피클액
- 빵
- 후추
Expected Time : 20
Direction :
- 1. 제단 2개를 쪼른 허벅지 살을 얇게 다. 2. 허벅지 살에 튀김을 얇게 다. 3. 양고기를 튀긴다. 4. 빵 사이에 마요네즈와 튀긴 고기를 넣는다.

Recipe Name : 피자

```

- For convenience, we used CSV format so you can see and edit Recipes in the Excel.

Shin Byeon

IIKHRecipeDB.csv - Excel

파일 홈 삽입 그리기 페이지 레이아웃 수식 데이터 검토 보기 도움말 팀

🔍 어떤 작업을 원하시나요?

🔗 공유 🗨 메모

📁 불러오기 📄 새 문서 📄 서식 복사 📄 클립보드 📄 글꼴 📄 맞춤 📄 표시 방식 📄 조건부 서식 📄 표 📄 셀 📄 삽입 📄 삭제 📄 서식 📄 📊 자동 합계 📊 채우기 📊 지우기 📊 정렬 및 필터 📊 선택

🔔 업데이트 가능 Office용 업데이트를 설치할 준비가 되었으나 먼저 일부 앱을 닫아야 합니다. [지금 업데이트](#)

🔔 데이터가 손실될 수 있음 이 통합 문서를 실패로 구분된 형식(csv)으로 저장하면 일부 기능이 손실될 수 있습니다. 기능을 유지하려면 Excel 파일 형식으로 저장하세요. [다시 표시 안 함](#) [다른 이름으로 저장...](#)

	A	B	C	D	E	F	G
		index	ingredients	name	time		
1	description						
2	1. Pour can of beef broth into slow cooker/ and add boneless	0	1 (14 ounce) can beef broth\$1 (18 ounce) bottleBBQ Pork for Sandwiches		270		
3	1. In a medium glass bowl/ prepare marinade by mixing garlic	1	1 tablespoon fresh parsley/ chopped\$1 tablespoBaked Salmon II		45		
4	Step 1: In a small saucepan over low heat/ combine the corns	2	1 clove garlic/ minced\$1 tablespoon cold water\$Baked Teriyaki Chicken		60		
5	1. Combine flour/ white sugar/ baking powder and salt. In a s	3	1 cup all-purpose flour\$1 cup milk\$1 egg/ beateBanana Pancakes I		10		
6	1. Preheat an oven to 375 degrees F (190 degrees C). Prepare	4	1 tablespoon baking powder\$1 teaspoon bakingBrown Sugar Bacon Waffles		30		
7	1. Preheat oven to 350 degrees F (175 degrees C).\$2. In a large	5	1 (10 ounce) can diced tomatoes with green chiBurrito Pie		30		
8	1. In a large bowl/ mix together flour/ salt/ baking powder an	6	1 1/2 cups warm milk\$1 teaspoon salt\$1 teaspocClassic Waffles		15		
9	Step 1: Bring a large pot of lightly salted water to a boil. Add	7	1 green bell pepper/ chopped\$1 green onion/ nCreamy Cajun Chicken Pasta		15		
10	1. Bake biscuits according to package directions.\$2. Meanwhil	8	1 (16 ounce) can refrigerated jumbo buttermilk Easy Sausage Gravy and Biscuits		10		
11	1. Combine milk with vinegar in a medium bowl and set aside	9	1 cup all-purpose flour\$1 egg\$1 teaspoon bakinFluffy Pancakes		10		
12	Step 1: Place crackers in a large resealable plastic bag; seal ba	10	1 egg\$1 teaspoon seasoned salt\$2 cups vegetabFried Chicken		20		
13	1. In a large bowl/ sift together the flour/ baking powder/ salt	11	1 1/2 cups all-purpose flour\$1 1/4 cups milk\$1 €Good Old Fashioned Pancakes		15		
14	Step 1: Preheat oven to 425 degrees F (220 degrees C). Lightly	12	1 pinch cayenne pepper\$1 tablespoon dried oreGreek Lemon Chicken and Potatoes		50		
15	1. Season salmon fillets with lemon pepper/ garlic powder/ an	13	1 1/2 pounds salmon fillets\$1/3 cup brown sugaGrilled Salmon I		16		
16	1. In a large bowl/ mix together the ground beef/ egg/ bread	14	1 cup beef broth\$1 cup thinly sliced onion\$1 egHamburger Steak with Onions and Gravy		25		
17	1. In a small bowl/ mix the maple syrup/ soy sauce/ garlic/ gai	15	1 clove garlic/ minced\$1 pound salmon \$1/4 cupMaple Salmon		20		
18	1. Combine water with salt/ ground black pepper/ oregano/ b	16	1 (7 ounce) package dry Italian-style salad dressSlow Cooker Italian Beef for Sandwiches		720		
19	1. Melt butter in a skillet over medium heat; add onion and co	17	1 (8 ounce) package fresh spinach\$1 tablespoorSpinach-Ricotta Quiche		55		

IIKHRecipeDB

100%

## 5. Setting Meal Plan for 2019/10/03 (Menu 4)

```
You've selected [4] to add plan
You will enter plan information in [[ YEAR -> MONTH -> DAY -> DAYNAME -> BREAKFAST -> LUNCH -> DINNER ]] order
Year      : 2019
Month     : 10
Day       : 3
Plan Name  : 내 생일
Breakfast (If you want to stop enter "stop" ) : 미역국
밥
stop
Lunch      (If you want to stop enter "stop" ) : 싸이버거
콜라
감자튀김
stop
Dinner     (If you want to stop enter "stop" ) : 치즈 케이크
커피 우유
stop
Press anykey to continue
```

## 6. You can see all Meal Plans including Plan you just added (Menu 5)

```
You've selected [5] to view plan
***** Meal plan for <20190930> *****
Plan Name      : 오늘 뭐 먹지
*****
>> Breakfast
- 바나나 우유
- 참치마요 삼각 김밥

>> Lunch
- 감자튀김
- 카우버거-치즈버거
- 콜라

>> Dinner
- 감자튀김
- 사이다
- 싸이버거

***** Meal plan for <20191003> *****
Plan Name      : 내 생일
*****
>> Breakfast
- 미역국
- 밥

>> Lunch
- 감자튀김
- 싸이버거
- 콜라

>> Dinner
- 치즈 케이크
- 커피 우유

Press anykey to continue
```

- For convenience, we used CSV format so you can see and edit Plans in the Excel.

breakfast	date	dinner	index	lunch	name
바나나 우유	20190930	감자튀김\$	0	감자튀김\$	오늘 뭐 먹지
미역국\$밥	20191003	치즈 케이크\$	1	감자튀김\$	내 생일

G. Explain how you applied object-oriented concepts to the development for your project.

1. [Encapsulation] In our OOP class we learned how to hide unnecessary details.

We always started from private and when it has to be exposed, we made it public.

Plus, we designed the class with their appropriate responsibility. So, our class has only required variables or functions that are related with its responsibility.



```
Recipe.h  Plan.h  Greeter.h  main.cpp
Team1-ILKH  Recipe
10 // stores recipe's name, description, ingredients, time
11 // and performs action related to recipe and it's member.
12 class Recipe {
13 private:
14     // Recipe's Name
15     std::string name;
16     // Recipe's Description
17     std::string description;
18     // Recipe's Ingredients: it is set of ingredient(we use "string" for ingredient)
19     std::set<std::string> ingredients;
20     // Recipe's Expected Time it takes to cook.
21     int time;
22     // ingredientString for saving(Buffer)
23     std::string ingredientString;
24 public:
25     // CTOR: name, description, time, ingredients will be initialized
26     Recipe(std::string && n, std::string && d, int t, std::vector<std::string> i) :
27         name(n), description(d), time(t), ingredients(i) {
28         // insert all ingredient to ingredients set
29         for (std::string s : i)
30             ingredients.insert(s);
31     };
32
33     /* GET */
34     // get Recipe name
35     const std::string & getName() const { return name; }
36     // get Recipe description
37     const std::string & getDescription() const { return description; }
38     // get Recipe ingredients
39     const std::set<std::string> & getIngredients() const { return ingredients; }
40     // get Recipe time
41     int getTime() const { return time; }
42
43     /* SET */
44     // sets Recipe name
45     void setName(std::string & n) { name = n; }
46     // sets Recipe description
47     void setDescription(std::string & d) { description = d; }
48     // sets Recipe time
49     void setTime(int t) { time = t; }
50
51     /* Functions */
52     // insert Recipe ingredient to ingredients set. and returns second stored value
53     bool addIngredient(std::string & n) { return ingredients.insert(n).second; }
54     // remove Recipe ingredient to ingredients set. and returns second stored value
55     bool removeIngredient(std::string & n) { return ingredients.erase(n) - 1; }
56     // find if there is ingredient
57     bool searchingIngredient(std::string & n) { return ingredients.find(n) != ingredients.end(); }
58     // make ingredients into single String
59     std::string getIngredientsIntoString() {
60         for (auto ingredient : ingredients) {
61             ingredientString.append(ingredient);
62             ingredientString.append("$");
63         }
64
65         return ingredientString;
66     }
67     // print Recipe data
68     void print() {
69         std::cout << "Recipe Name : " << name << std::endl;
70         std::cout << "Ingredients : " << endl;
71         for (std::string s : ingredients) {
72             std::cout << " - " << s << std::endl;
```



2. [Inheritance] We looked at the objects relation carefully and made programming more efficient.

For Instance, in case of database manager, many kinds of database managers were needed. We needed Plan Database manager, Meal Database Manager, Recipe Database Manager. Each Database managers was similar to each other and also the code was almost the same but saving format or few functions details were little bit different. So, we made DataBase [parent class] which had all overlapping details of all kinds of Database Managers. Inheriting Database [parent class] traits, making other Database Managers were easier (High Reusability). Also, we could eliminate the same codes.

In addition, with virtual keyword we implemented different details in the Database Managers' functions. That keyword acted like interface and all Database Managers [child class] had implemented mandatory details.

- DataBase [Parent]

```
23 // stores db map
24 // and controls entire DB system
25 class DataBase {
26 protected:
27     std::map<_key, _value> db;
28 public:
29     // static std::string recipeKeys[] = { "index", "name", "description", "time", "ingredient0", "ingredient1", "ingredient2", "ingredient3", "ingredient4" };
30
31     // insert pair of key and value
32     void _insert(_key k, _value v) { db.insert(std::pair<_key, _value>(k, v)); }
33
34     _value & _select(_key k) {
35         // if you do not find key (throws exception)
36         if (db.find(k) == db.end()) {
37             if (isEqual<_key, std::string>::value) throw DBException(0, k.c_str());
38             else if (isEqual<_key, Date>::value) throw DBException(1, k.c_str());
39             else throw DBException(2, k.c_str());
40         }
41
42         // if you find key (return value)
43         return (*db.find(k)).second;
44     }
45
46     // delete pair of key and value, insert new pair of key and value
47     void _update(_key k, _value v) {
48         db.erase(k);
49         db.insert(std::pair<_key, _value>(k, v));
50     }
51
52     // delete pair of key and value
53     void _delete(_key k) { db.erase(k); }
54
55     // show items with given key
56     virtual void _show(_key k) = 0;
57
58     // show all items in DB
59     virtual void _showAll() = 0;
60
61     // save DB(items) to local
62     virtual void _save() = 0;
63 };
64
65
```

## ● RecipeDB [Child]

```
95 // database manager for Recipe
96 // performs file I/O operations, db items printing
97 class RecipeDB : public DataBase<std::string, Recipe> {
98 private:
99
100 public:
101     // CTOR
102     RecipeDB() {
103         // msg
104         // std::cout << "[디버그] recipe DB 생성" << std::endl;
105         // Load File
106         std::vector<std::map<std::string, std::string>>> parsedData = CSVParser::instance().read("11KHRecipeDB.csv");
107
108         for (auto myMap : parsedData) {
109             // Construct Recipe Name from csv
110             std::string recipeName = myMap->find("name")->second;
111
112             // Construct Plan from csv
113             Recipe* recipePtr = new Recipe(
```

## ● PlanDB[Child]

```
152 // database manager for Plan(name + 3 meal)
153 // performs file I/O operations, db items printing
154 class PlanDB : public DataBase<Date, Plan> {
155 private:
156
157 public:
158     // CTOR
159     PlanDB() {
160         // Load File
161         std::vector<std::map<std::string, std::string>>> parsedData = CSVParser::instance().read("11KHPlanDB.csv");
162
163         for (auto myMap : parsedData) {
164             // Construct Date from csv
165             Date* datePtr = new Date(strdup(myMap->find("date")->second.c_str()));
166
167             // Construct Plan from csv
168             Plan* planPtr = new Plan(
169                 // plan name
170                 strdup(myMap->find("name")->second),
171                 *datePtr,
172                 {
173                     //아침
174                     Meal( CSVParser::instance().split( myMap->find("breakfast")->second , '$', -1 ) ),
175                     //점심
176                     Meal( CSVParser::instance().split( myMap->find("lunch")->second , '$', -1 ) ),
```

- Explain what you felt and learned from the project.

[고주형] It was nice to experience object-oriented programming as a team project. It was good chance to think deeply about object-oriented programming and design.

I was team leader so I kept track of project progress. I had to give everyone's work equally to until the program is fully implemented. Also, I had to think in Bottom Up manner. This gave me enlightenment about what is good design and why we use object-oriented programming. I had to rapidly gathered all of our team member's code and I integrated it into our master branch (we used git). When convention was different, I had hard time understanding it. But, unified convention and nicely encapsulated code was easy to use and I didn't confuse much what to use because unneeded functions or variables were forbidden.

[김호성] I've never had a chance to do a project on a team-by-team basis before, and it's been great to be able to do it through this team project, and the process of working together was fun.

[손희승] Coding with others was hard at first but after few days later, it became comfortable. I learned few tips. I know what Github is but didn't use it much. In this project I got chance to know about Git like, how to cooperate using Git, how to invite collaborator. With kind team members, I think I learnt a lot about cooperating.

[이준협] It was interesting to think in various ways because no implementation or details were restricted. And I felt again that teamwork is important.

[이하람] Through this project, I learned and knew about what an Object-oriented programming in the abstract. Also, it was very interesting to discuss together for making the programming because it was the first time for me to learn about c++.

## H. Conclusion

Our team project was successful. We argued a lot about what is more good design or what data structure we should use for many models. We learnt from each other by talking. We learnt how to cooperate in bottom up style code. This kind of experience will be helpful in the future.

Thank You😊