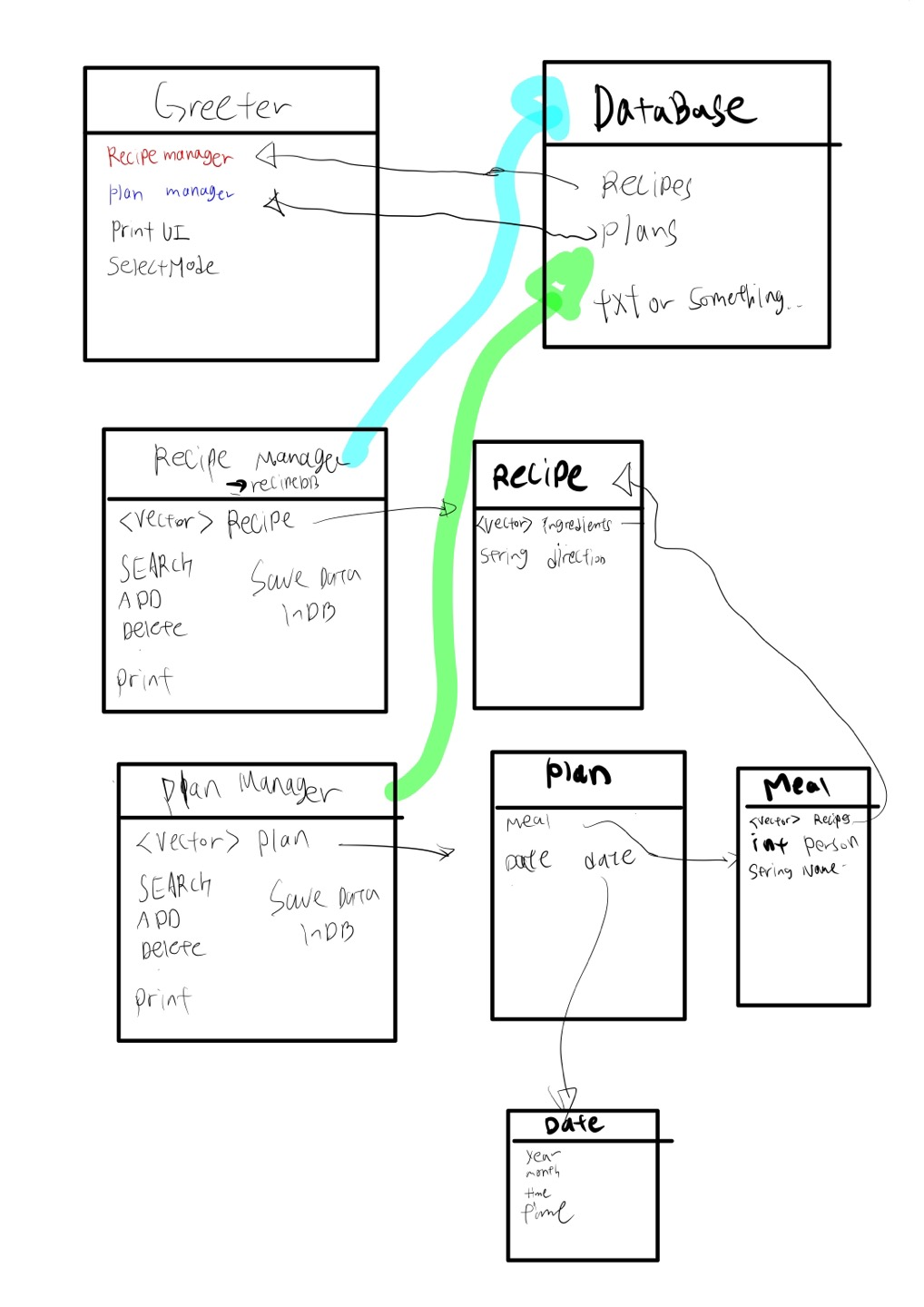
2022.2 Objected oriented programming prOJECT #1 PROB #1

### 20193802 이름

# 2022.2 Objected oriented programming prOJECT #1 PROB #1

before I make the structure of the IIKH(Interactive Interlligent Kitchen Helper), it is very ambiguous to make the structure. So I make classes by choosing two behaviors. First one is about Recipe, and second one is about meal plans. After choosing these two behaviors, I can make objects. And also I can figure out needed member data and functions. So I draw the flow of program by objects.



After drawing this, I make brief headefile of IIKH. Greeter is function that manage all part of IIKH, such as search,edit,find,view operations. And recipe manager and plan manager manages the whole data in IIKH by it’s own member variable and functions.

gretter.h

#include "recipemanager.h"

#include "planmanager.h"

class gretter

{

private:

recipemanager recipemanage();

planmanager planmanager();

public:

// load datas from db -> control datas by recipemanager, planmanager class.

gretter() {};

// basic IIKH intoduction print

void printUI();

// choose mode

int inputMode() {};

// edit, add, search ,delete recipes information in this func

// edit ,add, search, delete palns information in this func

void runprogramfunc(int inputmodenum);

};

Recipemanager.h

class recipemanager

{

private:

<vector> recipe recipes;

public:

// bring data from DB

recipemanager();

// manage recipes from db and update db

// search by name

void searchrecipe(string recipename);

// add recipe in recipe vector

void addrecipe();

// delete recipe that user want

void deleterecipe(string recipename);

// find recipe by name and edit

void editrecipe(sring recipename);

// print recipes by this func

void printrecipes();

// save recipes inDB

void saverecipse();

};

Recipe.h

class recipe

{

private:

<vector> string ingredients;

string directions;

public:

// contsructor

recipe();

recipe(<vector> string ingredients, string directions);

void editrecipe();

// setter

void setingredients(<vector> ingredient ingredients);

void setdirections(string direction);

// getter

<vector> ingredient getingredients();

string getdirections();

// print

void printrecipe();

};

planmanager.h

#include "plan.h"

class planmanager

{

private:

<vector> plan plans;

public:

// make plan vector by loading data in Database

planmanager();

void printplans();

// manage plans

void editplan(string planname);

void addplan();

void deleteplan(string planname);

void printplan();

void searchplan(string planname);

// save data in db

void saveplansindb();

// getter

<vector> plan getplans();

// setter

void setplans(<vector> plan);

};

Plan.h

#include "meal.h"

#include "date.h"

class plan

{

private:

meal mealinfo;

date dateinfo;

public:

plan();

plan(meal mealinfo, date dateinfo);

// setter

void setmealinfo(meal mealinfo);

void setdateinfo(date dateinfo);

// getter

meal getmealnfo(); s

date getdateinfo();

// print mealinfo, dateinfo

void printplan();

}

Meal.h

#include "recipe.h"

class meal

{

private:

<vector> recipe recipes;

int personamount;

string mealname;

public:

meal();

meal(<vector> recipe recipes, int personamount, string meal\_name);

// setter

void setrecipes(<vector> recipe recipes);

void setpersonamount(int personamount);

void setmealname(string mealname);

// edit meal

void editmeal();

// meal info print -> ingredients, recipe, direction by recipe class

void printmeal();

};

Date.h

#include <string.h>

class date

{

private:

int year;

int month;

int day;

string time;

public:

date();

date(int year, int month, int day, int time) {};

// Setter

void setYear(int year);

void setMonth(int month);

void setDay(int day);

void setTime(int time);

// Getter

int getYear();

int getMonth();

int getDay();

string time;

}