IIKH

Interactive Intelligent Kitchen Helper

TEAM 1

20224680 김경민

20223908 김주영

20200956 배정환

20214782 조주원

20225779 김제신

20202203 박호근

20225679 서규민

Contents

01. Introduce IIKH

02. DB

03. UML

04. Concepts for OOP

05. Demo

06. Efforts for efficient collaboration

07. Q&A

What IIKH do?

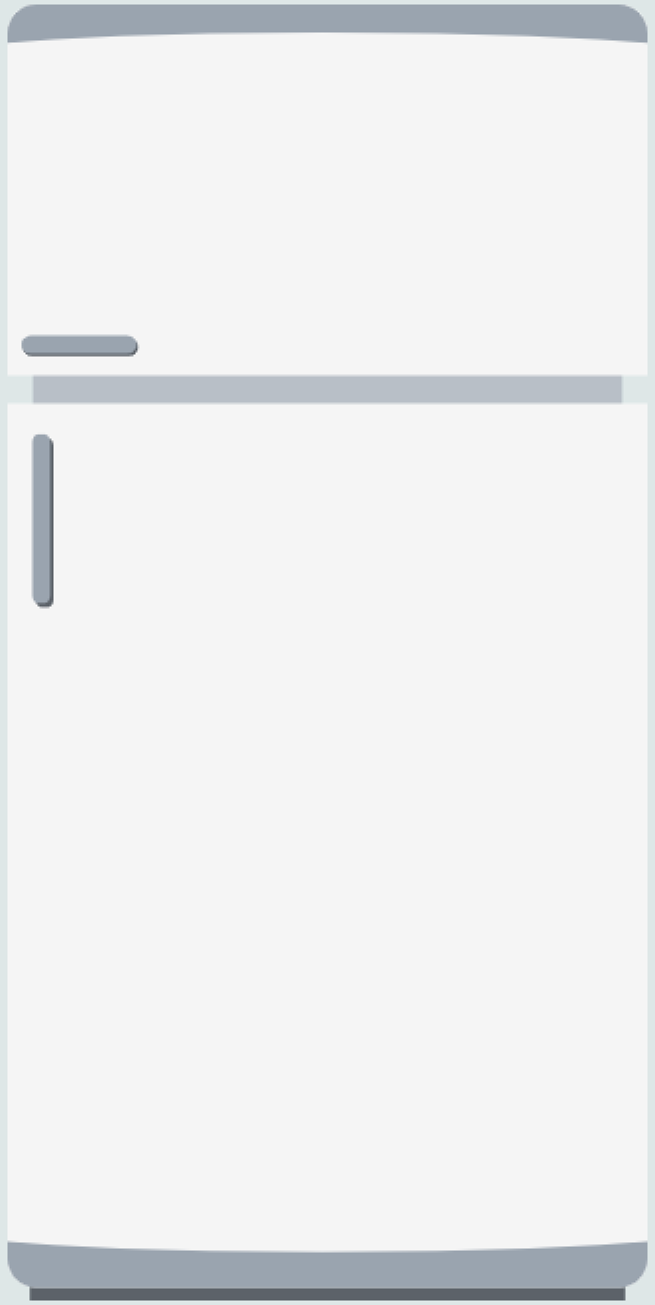
- Search Recipe DB
- Add new Recipe
- Plan a meal

.

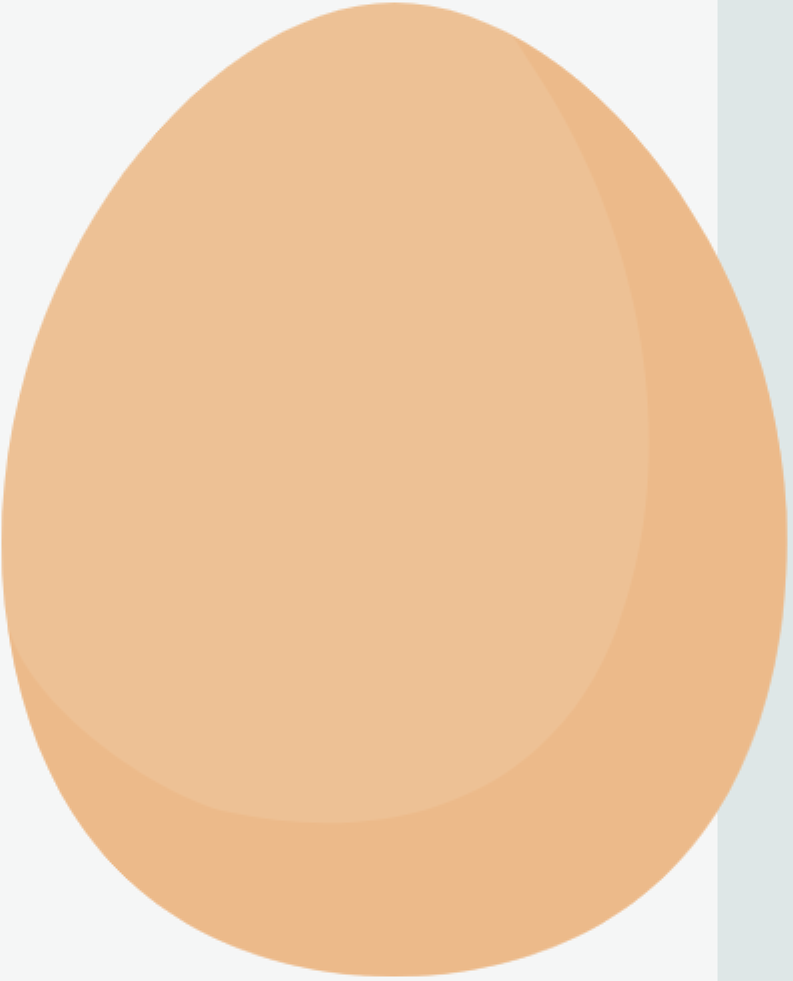
.

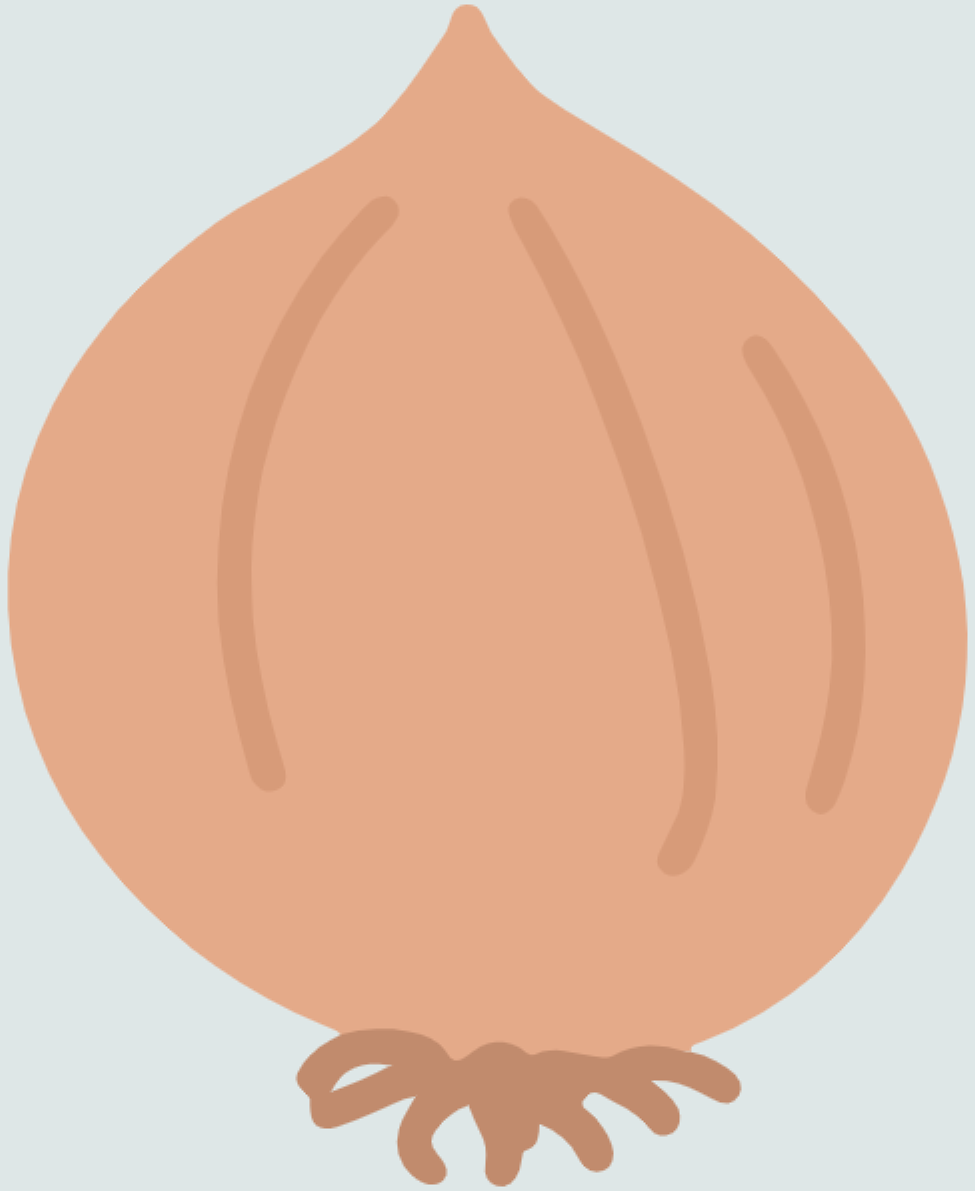
.

What IIKH do? _ Scenario



What IIKH do? _ Scenario

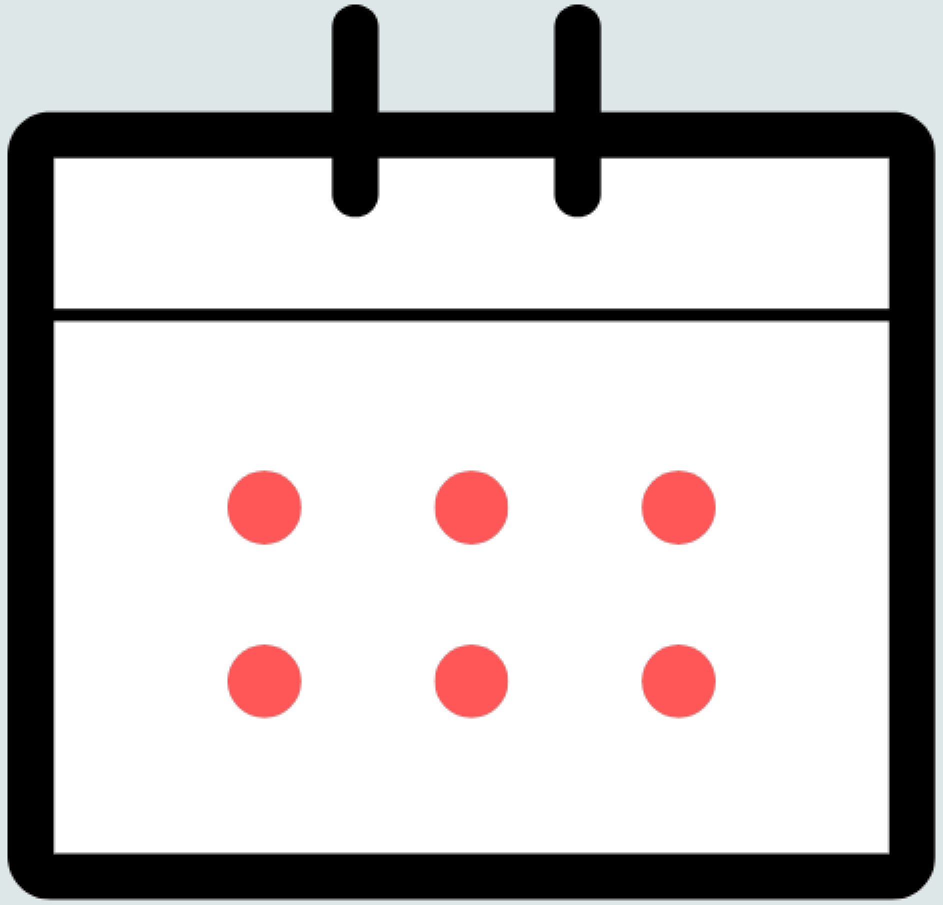




What IIKH do? _ Scenario

What IIKH do? _ Scenario

What IIKH do? _ Scenario



What IIKH do? _ Scenario

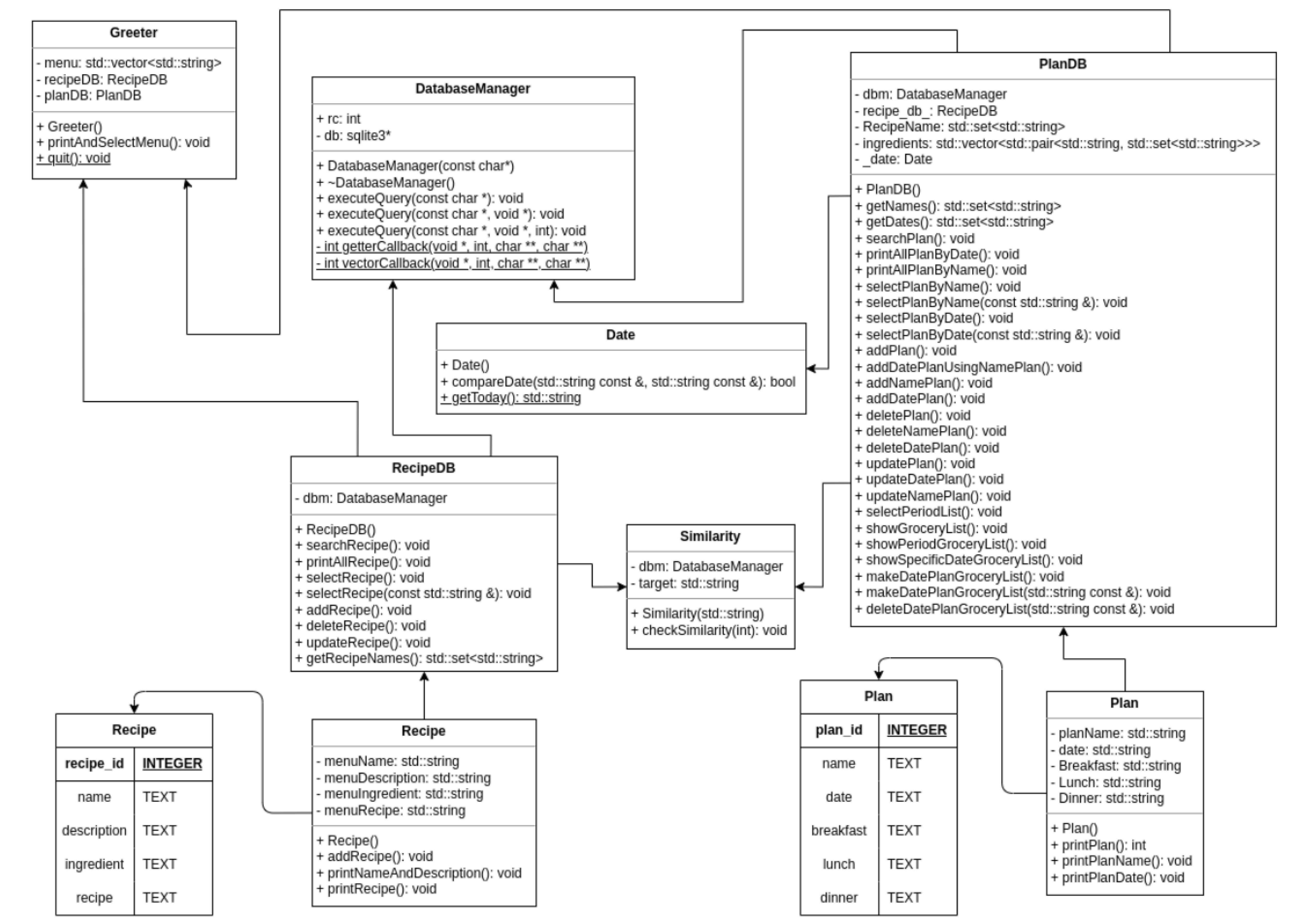
What IIKH do?

- Search Recipe/Plan from DB
- Add new Recipe/Plan
- Delete Recipe/Plan
- Update Recipe/Plan
- Print Grocery Lists

DB - SQLite

- Stability and Speed
- Light db for simple table structure in IIKH program
- File-based
- C++ library support

UML



Greeter

- menu: std::vector<std::string>
- recipeDB: RecipeDB
- planDB: PlanDB

+ Greeter()
+ printAndSelectMenu(): void
+ quit(): void

DatabaseManager

+ rc: int
- db: sqlite3*

+ DatabaseManager(const char*)
+ ~DatabaseManager()
+ executeQuery(const char *): void
+ executeQuery(const char *, void *): void
+ executeQuery(const char *, void *, int): void
- int getterCallback(void *, int, char **, char **)
- int vectorCallback(void *, int, char **, char **)

PlanDB

- dbm: DatabaseManager
- recipe_db_: RecipeDB
- RecipeName: std::set<std::string>
- ingredients: std::vector<std::pair<std::string, std::set<std::string>>>
- _date: Date

+ PlanDB()
+ getNames(): std::set<std::string>
+ getDates(): std::set<std::string>
+ searchPlan(): void
+ printAllPlanByDate(): void
+ printAllPlanByName(): void
+ selectPlanByName(): void
+ selectPlanByName(const std::string &): void
+ selectPlanByDate(): void
+ selectPlanByDate(const std::string &): void
+ addPlan(): void
+ addDatePlanUsingNamePlan(): void
+ addNamePlan(): void
+ addDatePlan(): void
+ deletePlan(): void
+ deleteNamePlan(): void
+ deleteDatePlan(): void
+ updatePlan(): void
+ updateDatePlan(): void
+ updateNamePlan(): void
+ selectPeriodList(): void
+ showGroceryList(): void
+ showPeriodGroceryList(): void
+ showSpecificDateGroceryList(): void
+ makeDatePlanGroceryList(): void
+ makeDatePlanGroceryList(std::string const &): void
+ deleteDatePlanGroceryList(std::string const &): void

Date

+ Date()
+ compareDate(std::string const &, std::string const &): bool
+ getToday(): std::string

RecipeDB

- dbm: DatabaseManager

+ RecipeDB()
+ searchRecipe(): void
+ printAllRecipe(): void
+ selectRecipe(): void
+ selectRecipe(const std::string &): void
+ addRecipe(): void
+ deleteRecipe(): void
+ updateRecipe(): void
+ getRecipeNames(): std::set<std::string>

Similarity

- dbm: DatabaseManager
- target: std::string

+ Similarity(std::string)
+ checkSimilarity(int): void

Recipe

recipe_id	INTEGER
name	TEXT
description	TEXT
ingredient	TEXT
recipe	TEXT

Recipe

- menuName: std::string
- menuDescription: std::string
- menuIngredient: std::string
- menuRecipe: std::string

+ Recipe()
+ addRecipe(): void
+ printNameAndDescription(): void
+ printRecipe(): void

Plan

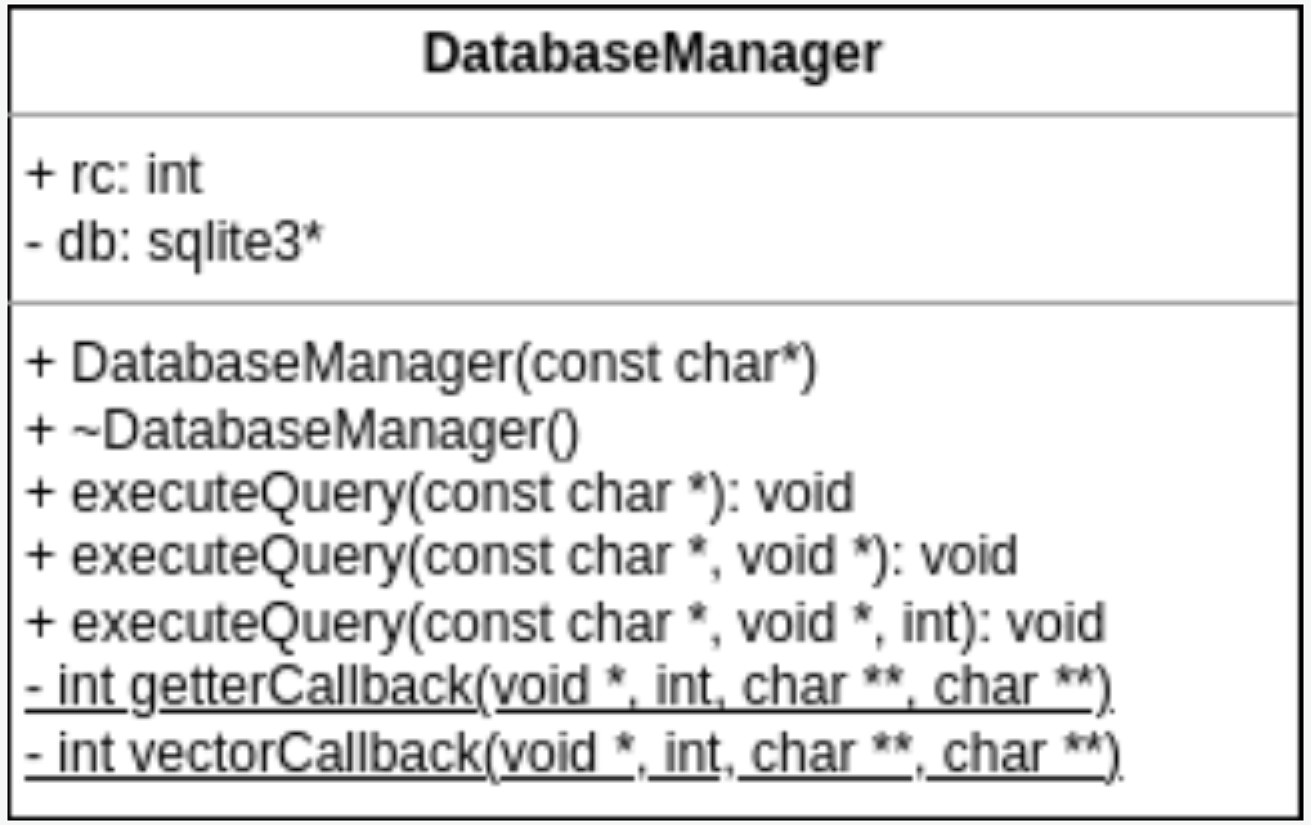
plan_id	INTEGER
name	TEXT
date	TEXT
breakfast	TEXT
lunch	TEXT
dinner	TEXT

Plan

- planName: std::string
- date: std::string
- Breakfast: std::string
- Lunch: std::string
- Dinner: std::string

+ Plan()
+ printPlan(): int
+ printPlanName(): void
+ printPlanDate(): void

DatabaseManager

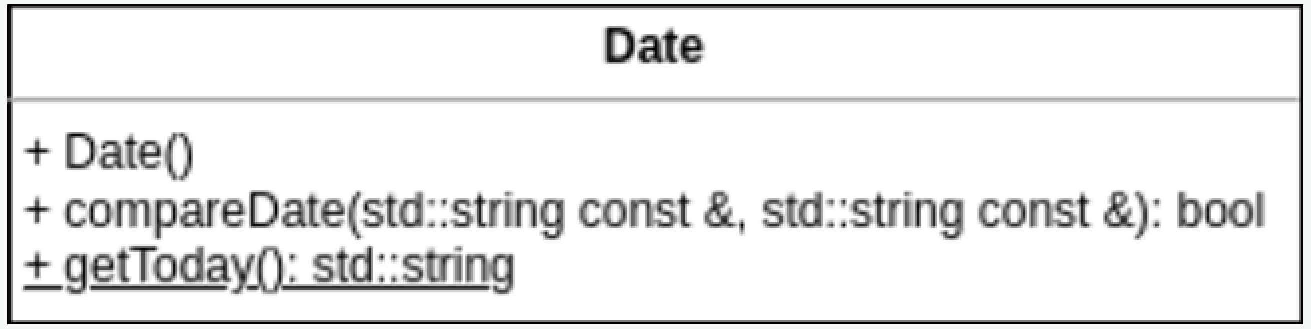


DatabaseManager

+ rc: int
- db: sqlite3*

+ DatabaseManager(const char*)
+ ~DatabaseManager()
+ executeQuery(const char *): void
+ executeQuery(const char *, void *): void
+ executeQuery(const char *, void *, int): void
- int getterCallback(void *, int, char **, char **)
- int vectorCallback(void *, int, char **, char **)

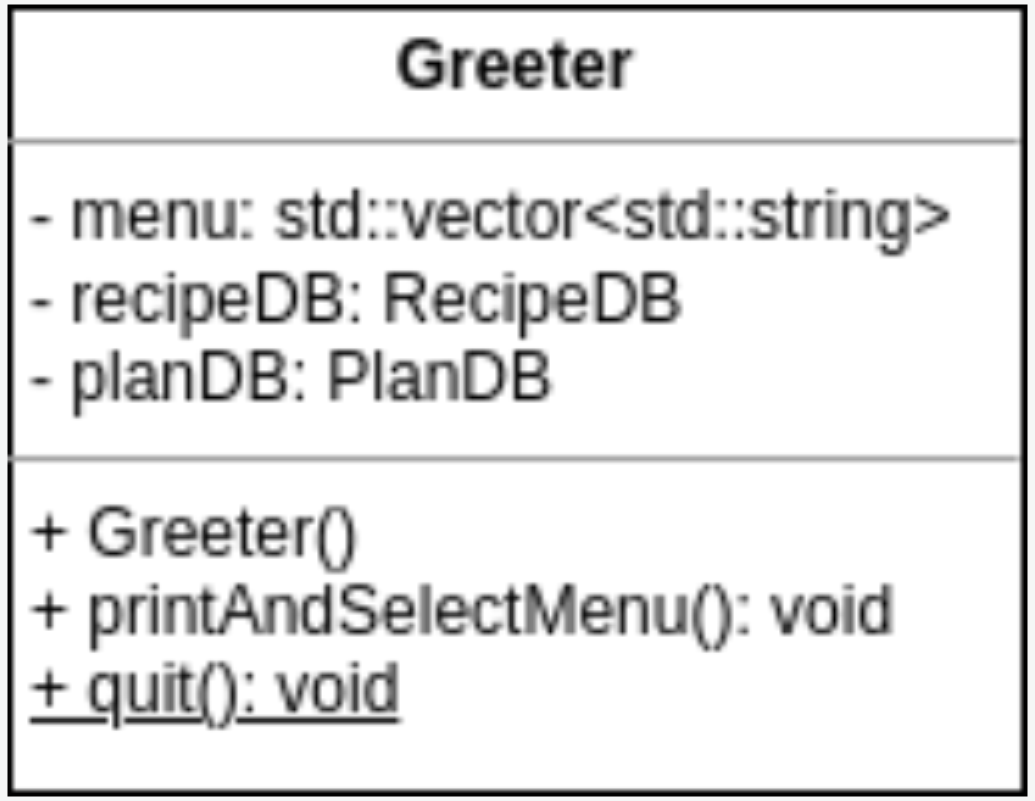
Date



Date

+ Date()
+ compareDate(std::string const &, std::string const &): bool
+ getToday(): std::string

Greeter

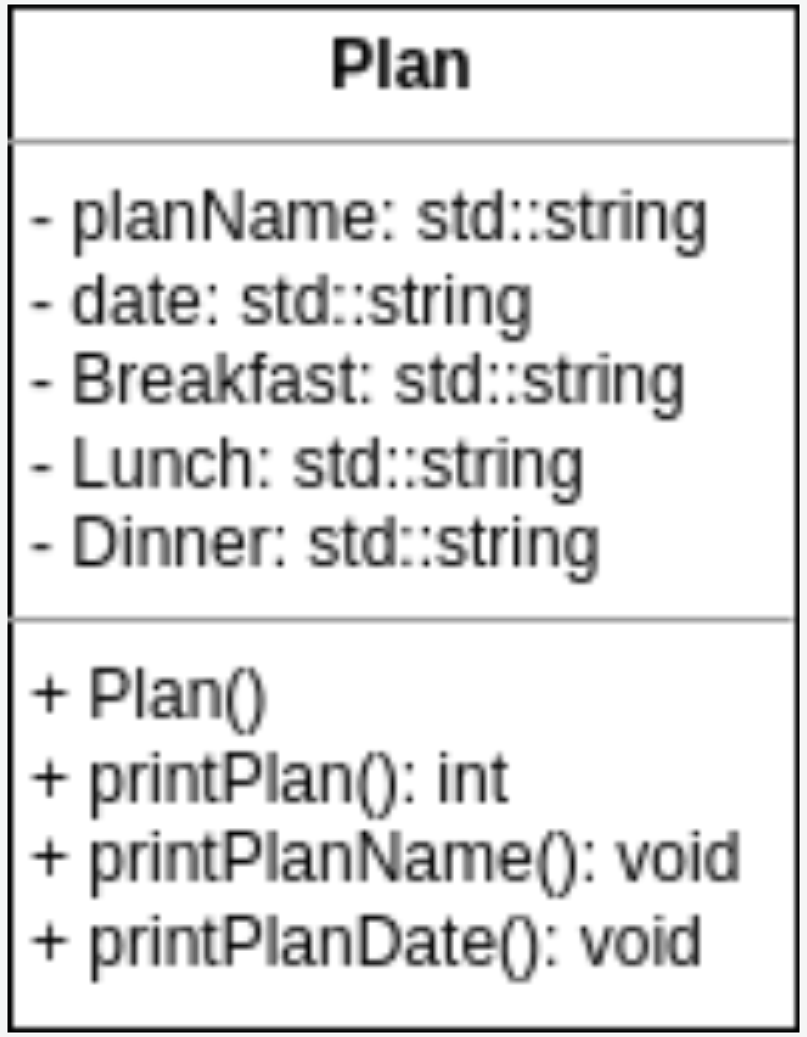


Greeter

- menu: std::vector<std::string>
- recipeDB: RecipeDB
- planDB: PlanDB

+ Greeter()
+ printAndSelectMenu(): void
+ quit(): void

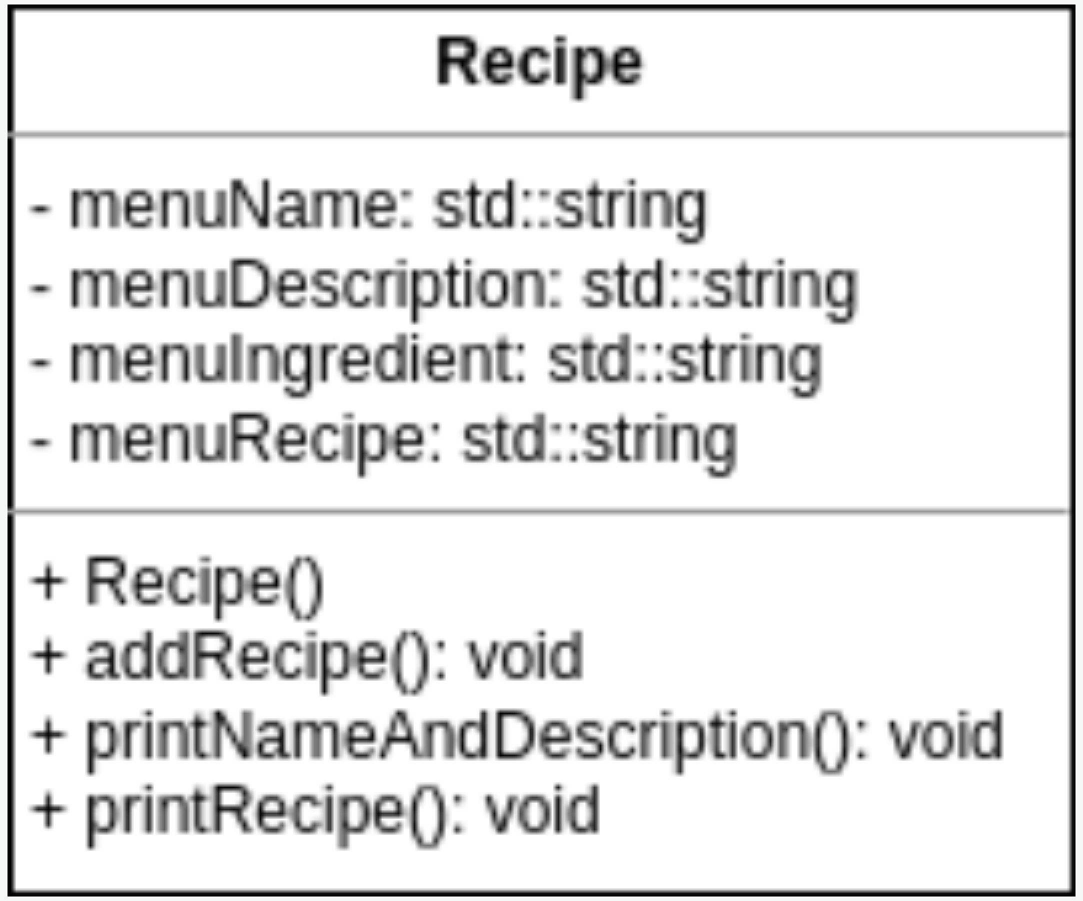
Plan, Recipe



Plan

- planName: std::string
- date: std::string
- Breakfast: std::string
- Lunch: std::string
- Dinner: std::string

+ Plan()
+ printPlan(): int
+ printPlanName(): void
+ printPlanDate(): void

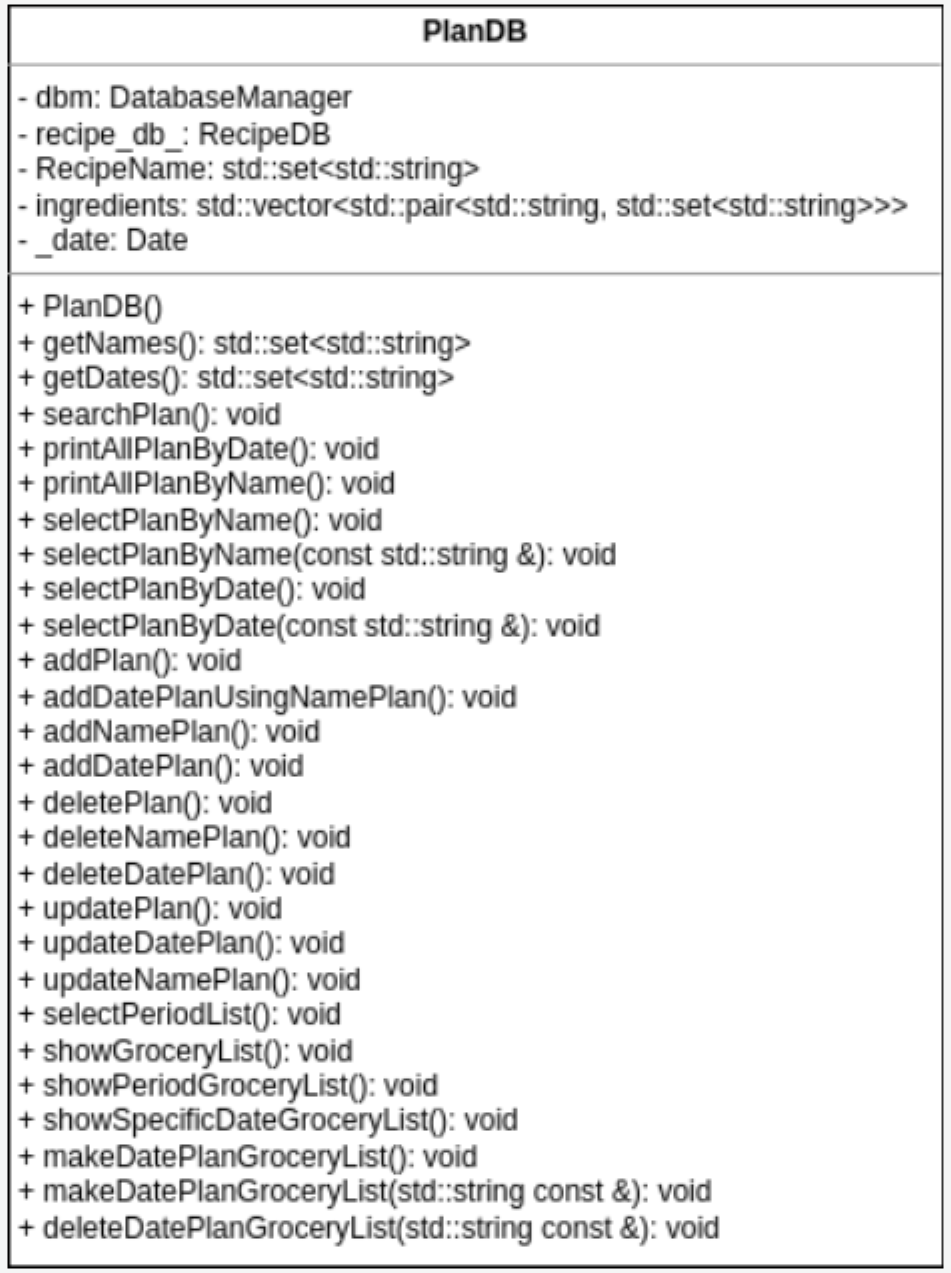


Recipe

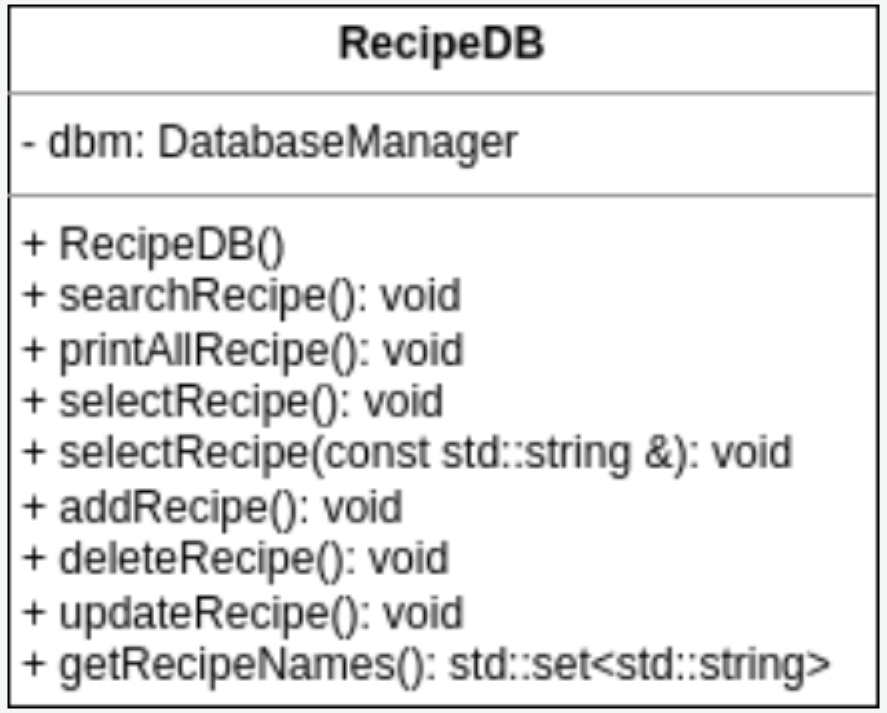
- menuName: std::string
- menuDescription: std::string
- menuIngredient: std::string
- menuRecipe: std::string

+ Recipe()
+ addRecipe(): void
+ printNameAndDescription(): void
+ printRecipe(): void

PlanDB, RecipeDB

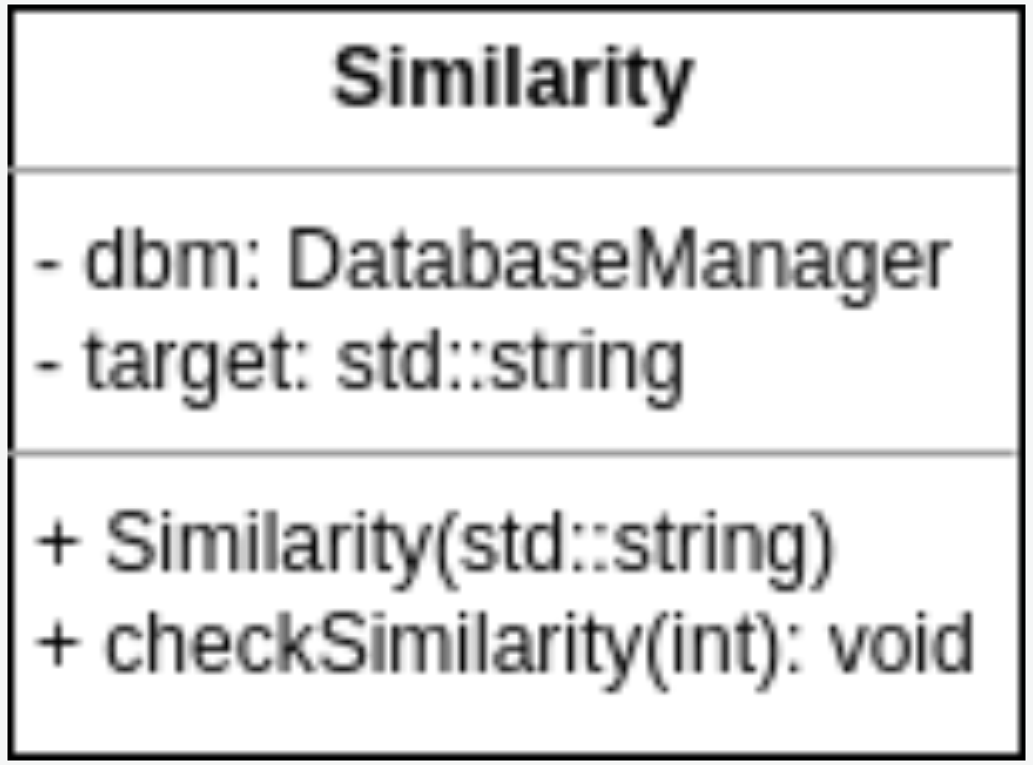


```
PlanDB
- dbm: DatabaseManager
- recipe_db_: RecipeDB
- RecipeName: std::set<std::string>
- ingredients: std::vector<std::pair<std::string, std::set<std::string>>>
- _date: Date
+ PlanDB()
+ getNames(): std::set<std::string>
+ getDates(): std::set<std::string>
+ searchPlan(): void
+ printAllPlanByDate(): void
+ printAllPlanByName(): void
+ selectPlanByName(): void
+ selectPlanByName(const std::string &): void
+ selectPlanByDate(): void
+ selectPlanByDate(const std::string &): void
+ addPlan(): void
+ addDatePlanUsingNamePlan(): void
+ addNamePlan(): void
+ addDatePlan(): void
+ deletePlan(): void
+ deleteNamePlan(): void
+ deleteDatePlan(): void
+ updatePlan(): void
+ updateDatePlan(): void
+ updateNamePlan(): void
+ selectPeriodList(): void
+ showGroceryList(): void
+ showPeriodGroceryList(): void
+ showSpecificDateGroceryList(): void
+ makeDatePlanGroceryList(): void
+ makeDatePlanGroceryList(std::string const &): void
+ deleteDatePlanGroceryList(std::string const &): void
```



```
RecipeDB
- dbm: DatabaseManager
+ RecipeDB()
+ searchRecipe(): void
+ printAllRecipe(): void
+ selectRecipe(): void
+ selectRecipe(const std::string &): void
+ addRecipe(): void
+ deleteRecipe(): void
+ updateRecipe(): void
+ getRecipeNames(): std::set<std::string>
```

Similarity



Similarity

- dbm: DatabaseManager
- target: std::string

+ Similarity(std::string)
+ checkSimilarity(int): void

Concepts for OOP

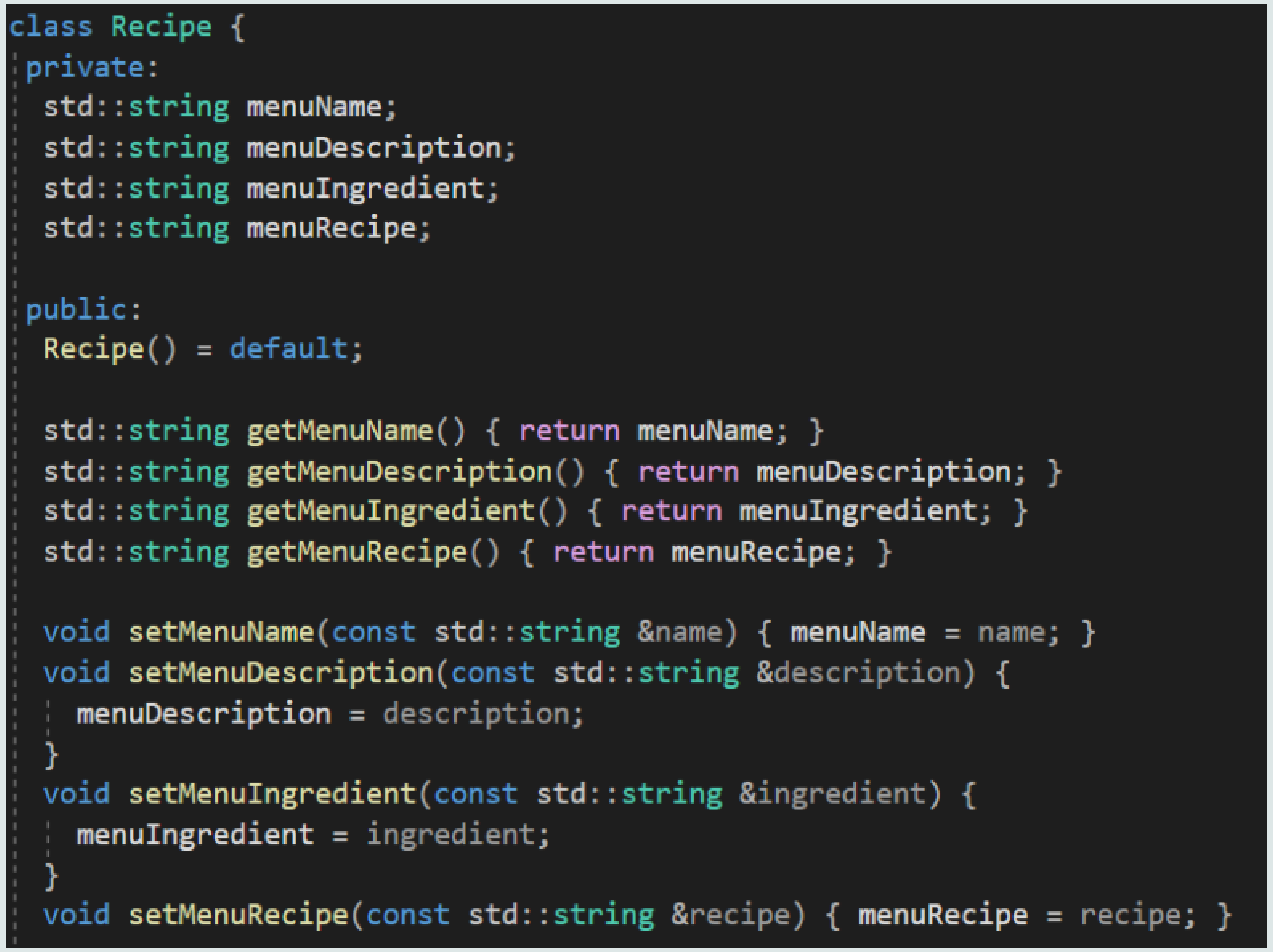
Encapsulation

Function Overloading

Encapsulation

Implementation independence

Information hiding

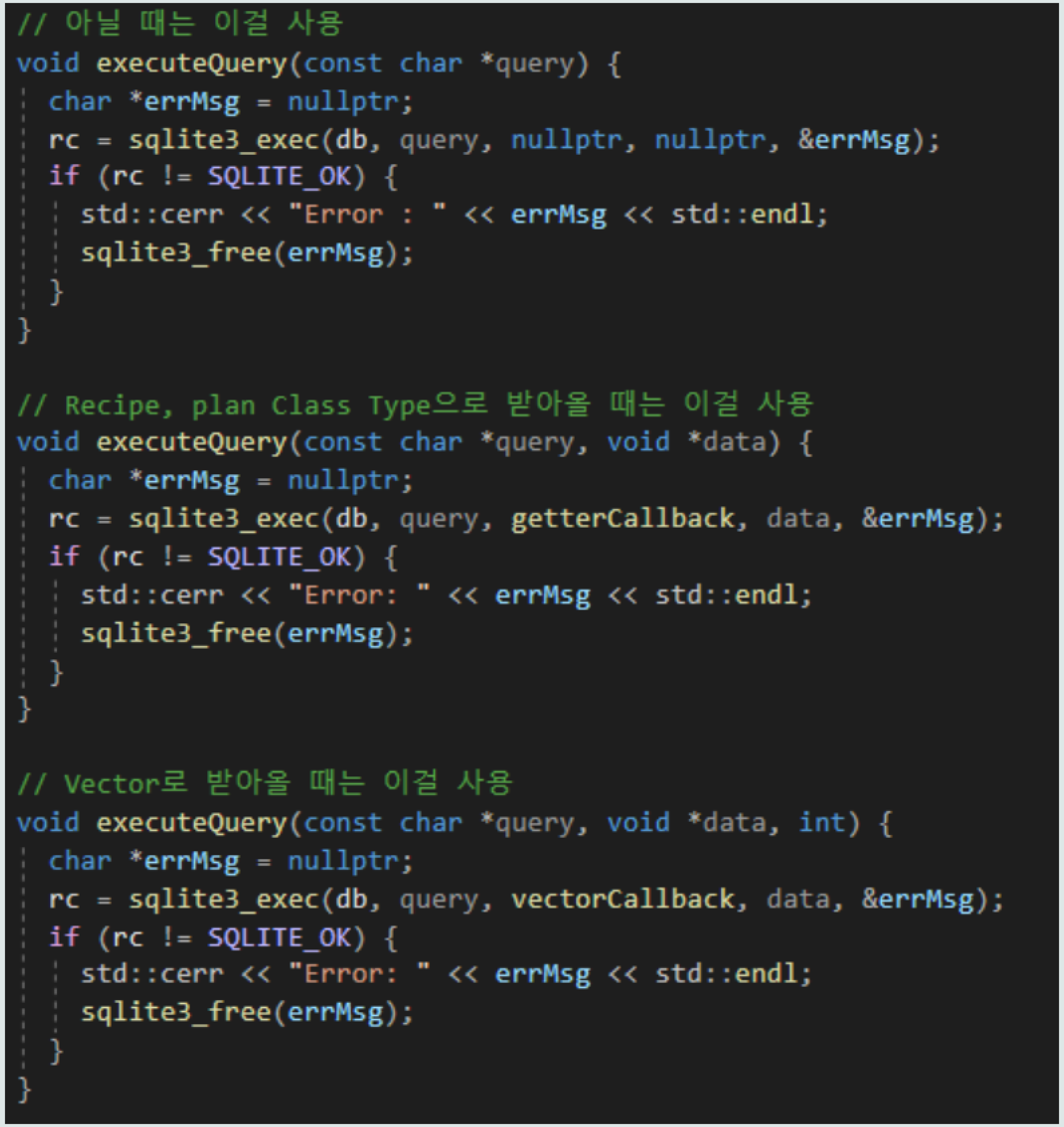


```
class Recipe {
 private:
  std::string menuName;
  std::string menuDescription;
  std::string menuIngredient;
  std::string menuRecipe;

 public:
  Recipe() = default;

  std::string getMenuName() { return menuName; }
  std::string getMenuDescription() { return menuDescription; }
  std::string getMenuIngredient() { return menuIngredient; }
  std::string getMenuRecipe() { return menuRecipe; }

  void setMenuName(const std::string &name) { menuName = name; }
  void setMenuDescription(const std::string &description) {
    menuDescription = description;
  }
  void setMenuIngredient(const std::string &ingredient) {
    menuIngredient = ingredient;
  }
  void setMenuRecipe(const std::string &recipe) { menuRecipe = recipe; }
```

Function Overloading

Diversify acceptable parameter

Reuse funtions



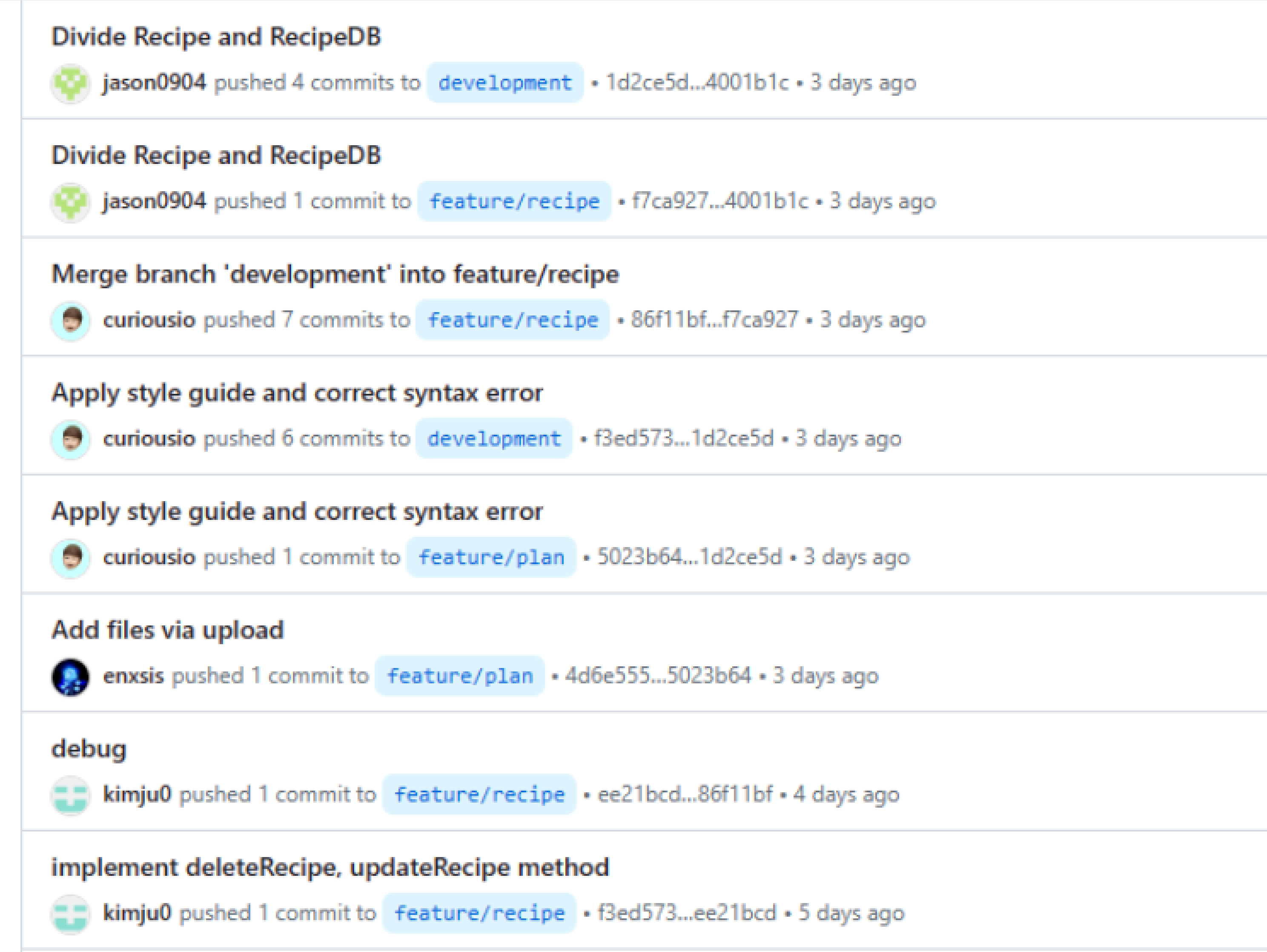
```
// 아닐 때는 이걸 사용
void executeQuery(const char *query) {
  char *errMsg = nullptr;
  rc = sqlite3_exec(db, query, nullptr, nullptr, &errMsg);
  if (rc != SQLITE_OK) {
    std::cerr << "Error : " << errMsg << std::endl;
    sqlite3_free(errMsg);
  }
}

// Recipe, plan Class Type으로 받아올 때는 이걸 사용
void executeQuery(const char *query, void *data) {
  char *errMsg = nullptr;
  rc = sqlite3_exec(db, query, getterCallback, data, &errMsg);
  if (rc != SQLITE_OK) {
    std::cerr << "Error: " << errMsg << std::endl;
    sqlite3_free(errMsg);
  }
}

// Vector로 받아올 때는 이걸 사용
void executeQuery(const char *query, void *data, int) {
  char *errMsg = nullptr;
  rc = sqlite3_exec(db, query, vectorCallback, data, &errMsg);
  if (rc != SQLITE_OK) {
    std::cerr << "Error: " << errMsg << std::endl;
    sqlite3_free(errMsg);
  }
}
```

Demo

Efforts for efficient collaboration

- Working separately in a branch by function
- Provide excutable files
- Update version frequently

Focus Efficiency and Sustainability beyond development



Divide Recipe and RecipeDB
jason0904 pushed 4 commits to `development` • 1d2ce5d...4001b1c • 3 days ago

Divide Recipe and RecipeDB
jason0904 pushed 1 commit to `feature/recipe` • f7ca927...4001b1c • 3 days ago

Merge branch 'development' into feature/recipe
curiousio pushed 7 commits to `feature/recipe` • 86f11bf...f7ca927 • 3 days ago

Apply style guide and correct syntax error
curiousio pushed 6 commits to `development` • f3ed573...1d2ce5d • 3 days ago

Apply style guide and correct syntax error
curiousio pushed 1 commit to `feature/plan` • 5023b64...1d2ce5d • 3 days ago

Add files via upload
enxsis pushed 1 commit to `feature/plan` • 4d6e555...5023b64 • 3 days ago

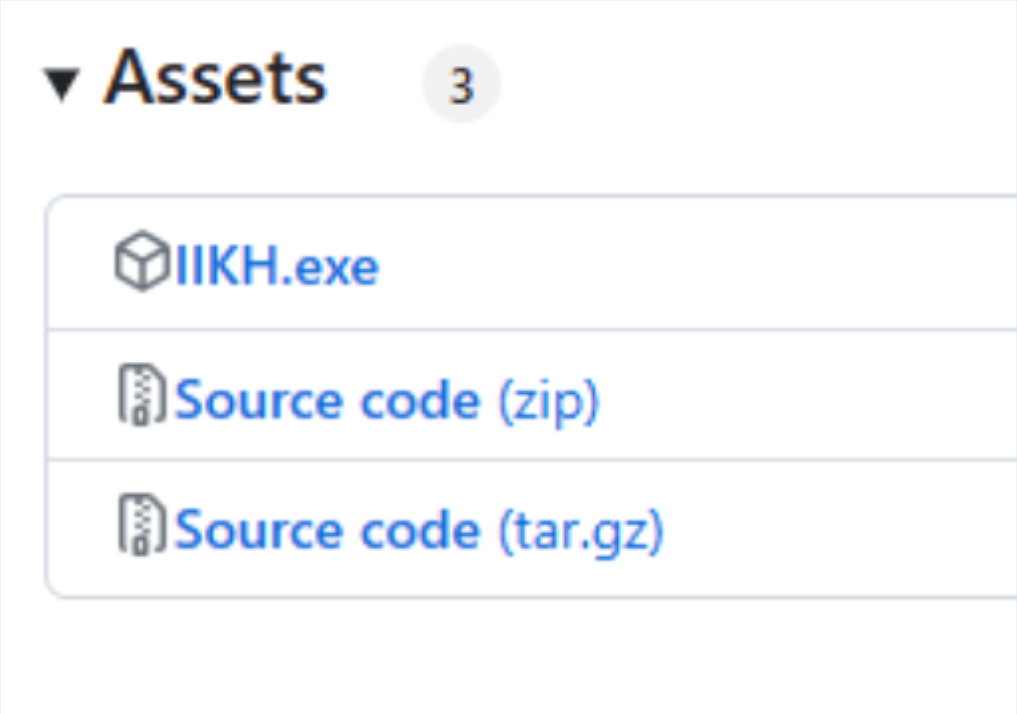
debug
kimju0 pushed 1 commit to `feature/recipe` • ee21bcd...86f11bf • 4 days ago

implement deleteRecipe, updateRecipe method
kimju0 pushed 1 commit to `feature/recipe` • f3ed573...ee21bcd • 5 days ago

Efforts for efficient collaboration

- Working separately in a branch by function
- Provide excutable files
- Update version frequently

Focus Efficiency and Sustainability beyond development



▾ Assets 3

IIKH.exe

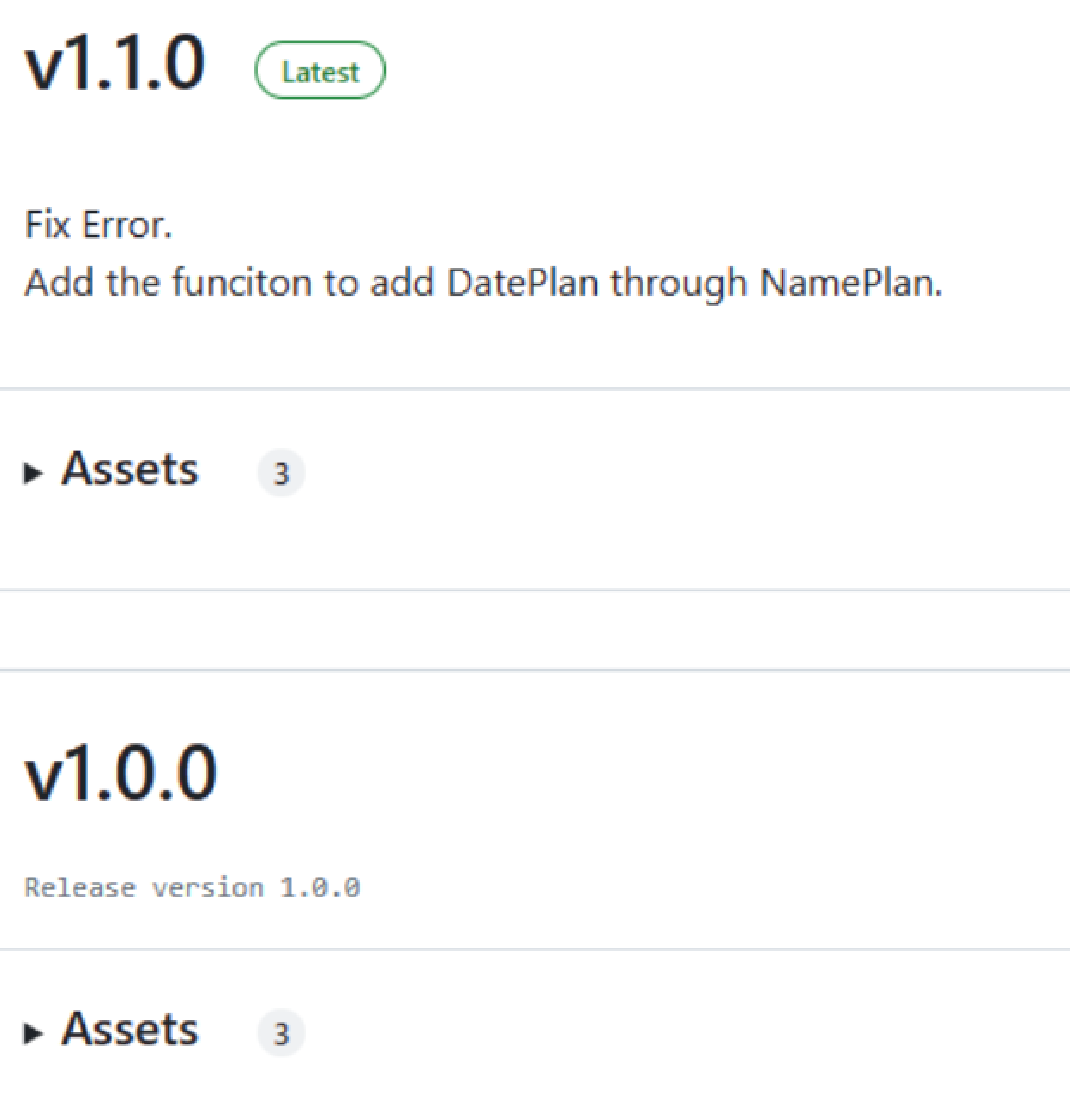
Source code (zip)

Source code (tar.gz)

Efforts for efficient collaboration

- Working separately in a branch by function
- Provide excutable files
- Update version frequently

Focus Efficiency and Sustainability beyond development



v1.1.0 Latest

Fix Error.
Add the funciton to add DatePlan through NamePlan.

▸ Assets 3

v1.0.0

```
Release version 1.0.0
```

▸ Assets 3

Thank you