

## 2班最終レポート

cy17248\_新倉旭

### 目次

1.	仕様書	2
1.1.	アプリケーションの目的	2
1.2.	アプリケーションの仕様・機能	2
1.3.	データベースの概要説明	4
2.	DB設計書	6
2.1.	実体関連図	6
2.2.	データベーススキーマ	6
3.	アプリケーションの設計書	7
3.1.	SQL一覧	7
3.2.	クラス図	10
4.	考察	11



図2はユースケース、図3~8はそれぞれのユースケース記述である。

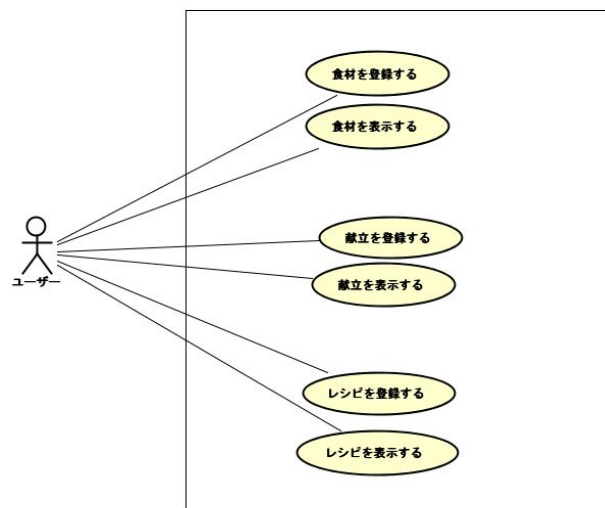


図2 ユースケース

項目	内容
ユースケース	レシピを登録する
概要	レシピデータベースに新しいレシピを登録する
アクター	ユーザー
事前条件	
事後条件	
基本系列	1)ユーザーはレシピ登録ボタンを押す 2)ユーザーはレシピの情報を登録する。そのときに必要な食材を登録する。 3)システムはレシピデータベースにレシピを追加する 4)システムは登録完了の旨を表示する
代替系列	
例外系列	
サブユースケース	
備考	

図3 レシピ登録

ユースケース	レシピを表示する
概要	レシピデータを表示する。
アクター	ユーザー
事前条件	
事後条件	
基本系列	1)ユーザーはレシピボタンを押す 2)システムはレシピデータベースに登録してあるレシピを表示する。 3)ユーザーはレシピの一覧を閲覧する
代替系列	
例外系列	
サブユースケース	
備考	

図4 レシピ表示

項目	内容
ユースケース	献立を登録する
概要	献立データベースに献立を登録する
アクター	ユーザー
事前条件	
事後条件	
基本系列	1) ユーザーは献立登録ボタンを押す 2) ユーザーは決めた献立を入力する 3) システムは献立データベースに決めた献立を追加する
代替系列	
例外系列	もし決めた献立が直近の献立と同じだった場合、かぶっている旨を警告する
サブユースケース	
備考	

図5 献立登録

項目	内容
ユースケース	献立を表示する
概要	献立データベース上の献立を表示する
アクター	ユーザー
事前条件	
事後条件	
基本系列	1) ユーザーは献立ボタンを押す 2) 日時・献立の絞り込み検索する 3) システムは献立データベースから献立の履歴を表示する 4) ユーザー献立の履歴を閲覧する
代替系列	データベースに何もなければその説明を表示する
例外系列	
サブユースケース	
備考	

図6 献立表示

項目	内容
ユースケース	食材を登録する
概要	冷蔵庫に食材を登録する。またその食材の消費期限も登録する。
アクター	ユーザー
事前条件	
事後条件	
基本系列	1) ユーザーは食材登録ボタンを押す 2) ユーザーは食材の情報を登録する。このとき消費期限についても入力する 3) システムは食材データベースに食材の情報を追加する 4) システムは登録完了の旨を表示する 5) ユーザーは食材を冷蔵庫に入れる
代替系列	
例外系列	
サブユースケース	
備考	

図7 食材の登録

項目	内容
ユースケース	食材を表示する
概要	冷蔵庫の食材を把握するために表示する。また、食材の情報として消費期限が表示される
アクター	ユーザー
事前条件	
事後条件	
基本系列	1) ユーザーが食材を表示するボタンを押す 2) システムは冷蔵庫データベースにある現在の冷蔵庫の中身の食材を表示する 3) ユーザーは食材の一覧と各食材の情報を閲覧する
代替系列	冷蔵庫に食材が何もなければその説明を表示する
例外系列	
サブユースケース	
備考	

図8 食材の表示

### 1.3. データベースの概要説明

データベースは全部で献立、レシピ、食料品、冷蔵庫、登録、食材の6個である。以下では、それぞれのデータベースについて目的と概要を述べる。

**献立**：作ったレシピを記録し参照するためDB化した。従ってこのデータベースから直近に作った料理名を参照し登録する料理名と比較することで短期間に同じ献立が被ることを防ぐことができる。

レシピ：その料理を作るために必要な食材とその個数を参照することでその日に選択したレシピの食材を使った分だけ冷蔵庫の中に保存されている個数から引くことができる。

食料品：買った食材をその都度登録していくことで今何が冷蔵庫のなかに幾つ在庫があるのかということを一覧から管理することができる。また、食材ごとに表示され賞味期限でソートされているので、期限の早いモノから消費していくことが可能である。

冷蔵庫：食料品を定義するにあたり識別には”その冷蔵庫に入っている”というキーが必要であり、冷蔵庫には主キーがなかったためDB化した。しかしこれによって、複数台の冷蔵庫を所有している場合にも使える仕様となった。

登録：どのレシピに何をどれだけ使うかということ、レシピを参照する際に必要であったためDB化した。

食材：冷蔵庫の中に入っている食料品とは別で、レシピを登録する際に使う食材を一般的に存在するモノから参照して選べるようにDB化した。

## 2. DB設計書

### 2.1. 実体関連図

本アプリケーションで使用する実体関連図の一覧は以下である。

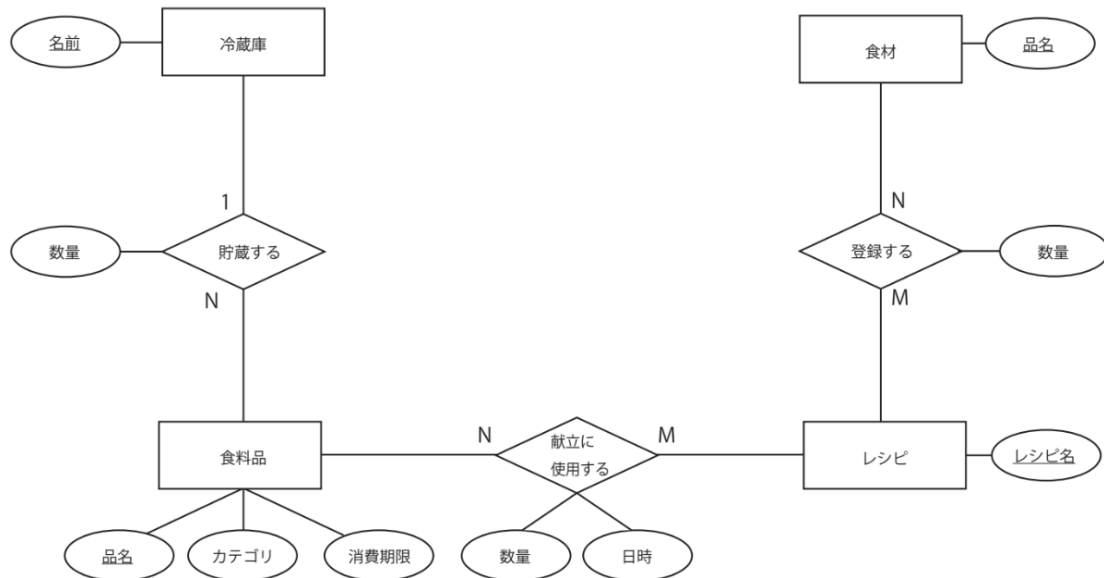


図9 実体関連図

### 2.2. データベーススキーマ

本アプリケーションで使用するデータベーススキーマの一覧は以下である。

冷蔵庫 ( ref ) :

(名前[VARCHAR])

食料品 ( reffood ) :

(名前[VARCHAR], 数量[FLOAT], 品名[VARCHAR], カテゴリ[VARCHAR],  
消費期限[DATE])

( reffood\_temp ) :

(名前[VARCHAR], 数量[FLOAT], 品名[VARCHAR], カテゴリ[VARCHAR],  
消費期限[DATE])

献立に使用する ( menu ) :

(品名[VARCHAR],数量[FLOAT],日時[DATETIME],レシピ名[VARCHAR])

レシピ ( recipe ) :

(レシピ名[VARCHAR])

登録する ( register ) :

(レシピ名[VARCHAR],数量[FLOAT],品名[VARCHAR])

食材 ( food ) :

(品名[VARCHAR])

### 3. アプリケーションの設計書

#### 3.1. SQL一覧

本アプリケーションで使用するSQLの一覧とその目的は以下である。

<表示>

SQL : SELECT \*  
FROM reffood ORDER BY 消費期限

使用クラス : ShowGrocery

目的 : 食料品を消費期限の古い順で表示。冷蔵庫の中身に何が入っているかを確認するため。

SQL : SELECT 品名 ,消費期限  
FROM reffood ORDER BY 消費期限

使用クラス : ShowLimit

目的 : 消費期限の表示。消費期限の近いものから使用するためにほしい情報だけを表示した。

SQL : SELECT レシピ名  
FROM recipe

**使用クラス** : ShowRecipe

**目的** : レシピを表示する

SQL : SELECT \*

FROM menu ORDER BY 日時

**使用クラス** : ShowRecipe

**目的** : 献立を表示

#### < 登録 >

SQL : INSERT INTO reffood

VALUES('冷蔵庫名',数量,'品名','カテゴリ',消費期限(INT))

**使用クラス** : AddGrocery

**目的** : 冷蔵庫に追加した食料品の情報を登録するため。

SQL : INSERT INTO menu

VALUES ('品名',数量,',',日時,'レシピ名')

**使用クラス** : AddMenu

**目的** : 献立で使用した食料品とその数量とレシピを登録する。日時は自動取得。

SQL : INSERT INTO recipe

VALUE('レシピ名')

**使用クラス** : AddRecipe

**目的** : レシピを追加する。

#### < 更新 >

SQL : UPDATE reffood

SET 数量 = 数量 - 1

WHERE 消費期限 IN (SELECT MIN(消費期限)

FROM reffood\_temp

WHERE 品名 = 'name' )

AND 品名 = 'name'

**使用クラス** : UseGrocery

**目的** : 食料品の更新。献立で使用された食料品の数を自動で更新するため



SQL : UPDATE reffood\_temp, reffood  
SET reffood\_temp.数量 = reffood.数量  
WHERE reffood\_temp.名前=reffood.名前  
AND reffood\_temp.品名=reffood.品名  
AND reffood\_temp.消費期限=reffood.消費期限

使用クラス : UpdateTemp

目的 : mySQLのエラー回避のために使うテーブルのアップデートするため

SQL : SELECT 品名  
FROM menu  
WHERE 日時 IN (SELECT MAX(日時)  
FROM menu)

使用クラス : GetName

目的 : 献立で使用された品名を取得し、UseGroceryに渡すため。

SQL : SELECT 数量  
FROM menu  
WHERE 日時 IN (SELECT MAX(日時)  
FROM menu)

使用クラス : GetQuantity

目的 : 献立で使用された数量を取得し、UseGroceryに渡すため。

SQL : DELETE FROM reffood  
WHERE 数量 <= 0

使用クラス : DeleteGrocery, DeleteGrocery

目的 : 無くなった食料品の項目の削除。

### 3.2. クラス図

以下の図10はアプリケーションのクラス図である。

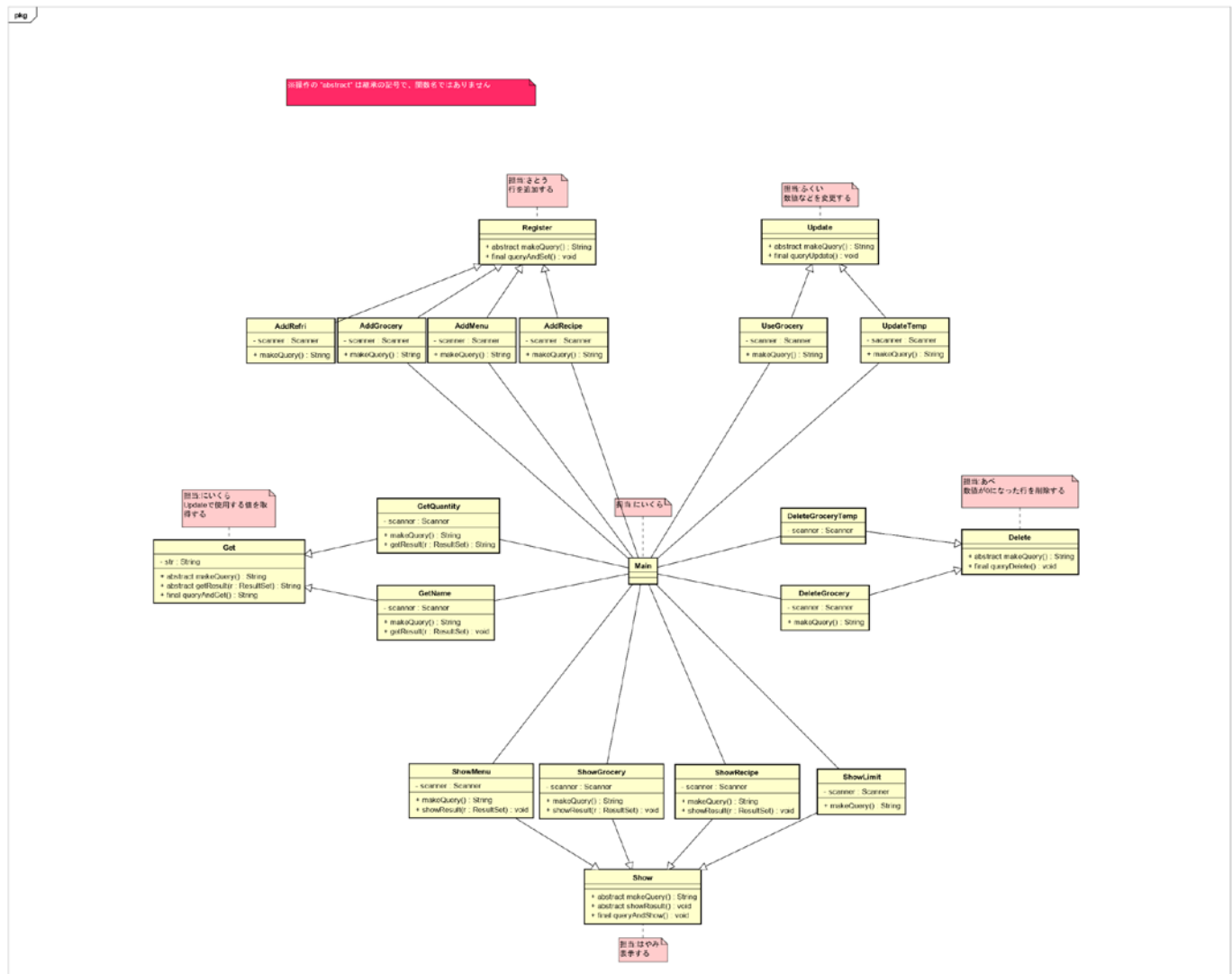


図10

## 4. 考察

システムの当初の目標はほとんどすべて達成できた。特に冷蔵庫側に関わる部分は完全に作り終えることができた。しかし、レシピから直接献立に追加する機能が作れなかったのと、献立に追加するときに材料を一種類ずつしか追加できないようになってしまっているため、まだまだ改善できる部分は残ってしまった。

グループで考えている中で、同じビジョンを共有し切れていなかったため、実際にプログラムを作るときに何度も確認作業が必要になってしまい、無駄に時間がかかってしまった。

当初実装しようとしていた機能だけでは物足りない部分があると感じたので、同じコンセプトで一から作り直したらまたもっとよい物ができそう。