Smart Fridge

Final report

Project on Databases (2)

Tamara Savkova

August 30, 2018

1 About the document

This document represents Final Report submitted under subject Databases (2). Its purpose is to describe data model and functions of the system for Smart Fridge, which is designed for usage in restaurant, and it keeps account on groceries inventory and recognizes which items were inserted and ejected.

2 Data model

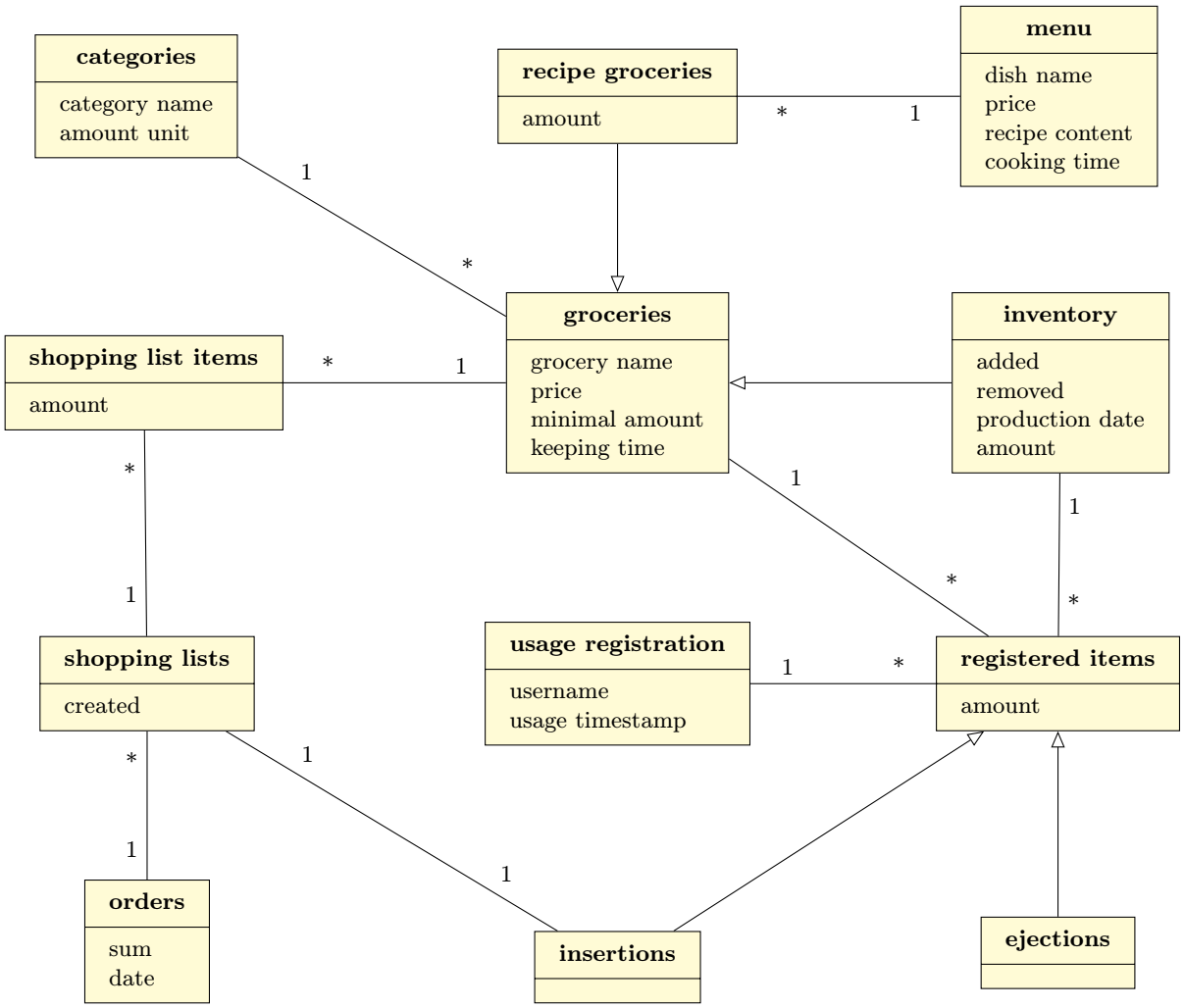

Figure 1: Entity relationship model of the Smart Fridge system

The system (obr. 1) keeps the inventory of the Fridge (`inventory`), which contains the groceries in use (`groceries`).

It has some information for every item: grocery name, categories (`categories`), to which it may belong, measurement unit, keeping time in days, price and minimal amount.

The Fridge keeps doors openings history, distinguishes such an actions as insertion (`insertions`) and ejection (`ejections`) of the groceries, recognizes which items were inserted and ejected, their amount, and saves this information as well as date and time of the usage. The restaurant workers are the Fridge users. Every grocery belongs to the category, which define the measurement unit.

The Fridge generates shopping lists (`shipping lists`) which contain chosed by user or missing items, i.e. amount of which is less than minimal or spoiled ones. According to the shopping lists, the Fridge can create the orders (`orders`). The groceries are ejected when are taken by the users for cooking the dish by recipe or after ejecting the spoiled items, and are inserted after when the orders are completed.

The Fridge functionality suggests receipts suggesting (`recipes`) or menu (`menu`) according to the contents.

3 Functions

System lets list, insert, eject, remove, edit and display details of:

- groceries,
- menu items with recipes,
- shopping lists

and only list the details of:

- the Smart Fridge content,
- categories,
- available items (by required amount),
- shopping lists,
- orders,
- doors openings,
- groceries with lacking amount,
- spoiled groceries,
- statistict.

Further the system provides:

- adding the removing the items;
- handling the shopping lists – goes through all the shopping lists and tries to pair each to the Fridge opening and insertion record. The list is paired if items from the one were added to the Fridge. An order is created containing every unpaired shopping list;
- creating the shopping list of minimal requirements – creates shopping list (i.e. list of missing items) with required minimal amounts of groceries;
- expiration control – lists all the items after expiration, creates shopping list with required minimal amounts of groceries;
- arranging the menu – for every dish chooses the recipe by its complexity based on the recipes in database and actual inventory. The chosen recipe contains only present in the inventory groceries. If there are several such of recipes, any of the most complex is chosen (i.e. which requires the most kinds of groceries).

The system creates statistics of:

- inventarization – checks its current state comparing to the Fridge usage history and lists all the differences (i.e. such an items, which were inserted to the Fridge but are actually missing and which were ejected, but are still in the inventory);
- expenses – counts the expenses on purchases;
- for every month within last 12 months calculates the coefficient for every category: how many groceries were ejected from the Fridge after their expiration.

4 Relation database

The image 2 displays the final relation database, which was gained after transformation of entity relation model at the image 1. Subsets `inventory`, `recipe groceries` were transformed using the strategy *every (sub)set to the separate table* and subsets `insertions`, `ejections` were transformed using the strategy *every set to single table*.

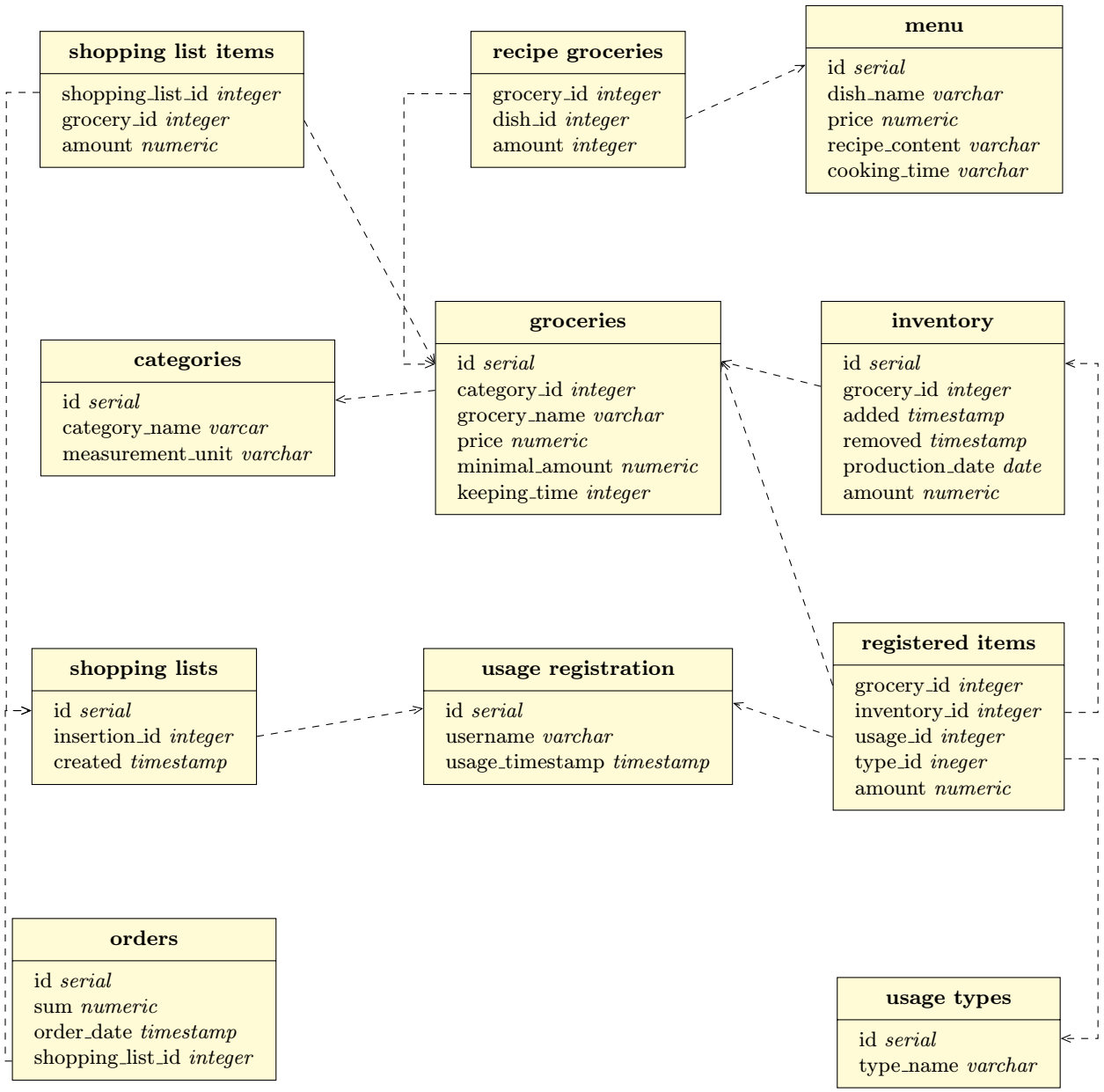

Figure 2: Relational model of Smart Fridge system data

5 Code organization

Application is programmed in *Java* programming language. Uses pattern *Row Data Gateway* and *Transaction Script*. Access to the database is realizes via *JDBC*. Source code is separated to the following packages:

`smart_fridge.application`

This package contains class `Main`, with which the application is run. During the run the application has a single connection to the database system created, which is maintained in class `DbContext`.

smart_fridge.application.rdg

This package contains *Data Gateway* for every table in database. *Row Data Gateway* is realizes with two classes – one is for *Gateway* itself and second is for *Finder* (package `smart_fridge.application.rdg.finders`). Duplicate code is separated to classes `BaseGateway` and `BaseFinder`, inherited by other classes.

Subset relation between sets `groceries` and `inventory` was realized using the inheritance, as soon as every inventory item belongs to `groceries`. Class `Items` inherits class `Groceries` and adds specific attributes.

smart_fridge.application.ts

This package contains code for complex domain operations. It is realizes with use of pattern *Transaction Script*. Related domain functions will be grouped to single classes. Functions providing groceries insertion to the Fridge are realized with class `InsertionService`. Further putting of groceries to the shopping lists is realized by class `PairingService`. Class `OrderingService` realizes orders creation for the shopping lists, which were left unpaired. Functions providing groceries ejection from the Fridge are realized with class `EjectionService`.

smart_fridge.application.ui

This package contains code for user interface using ASCII graphics and controlled by user input in the console. The core of the interface are menus, using which user can choose the menu item by typing its number. Every menu is realizes with single class. Duplicate code is separated to class `Menu` (package `smart_fridge.application.ui.menus`), from which the other menus inherit. Independency of *Row Data Gateway* from user interface is provided with printing the data using classes `Printer` (package `smart_fridge.application.ui.printers`).

6 Chosen solved problem

Creating the statistics on *Product waste ratio* was impossible within the original sketch for data model. The goal is for every month within last 12 months calculates the coefficient for every category: how many groceries were ejected from the Fridge after their expiration. It was needed to find out the amount of groceries which were out of expiry date at the moment of registered ejection from the Fridge within last 12 moths. These groceries were grouped by category. It was impossible to get this information as soon as items present in the Fridge didn't have own id and the date and time of ejection were not registered. The data of the table `Inventory` informed only about an actual state of inventory, and after insertion of the rows `id` and `removed` the data represented the complete history of the groceries in the fridge. After adaptation of other tables and data according to the changes, it was possible already to create such a statistics.

The Final report on Smart Fridge project was inspired by sample Final report by Alexander Šimko for sample project on Databases (2) "Telekomunikačný systém" ("Telecommunication network")