Software Domain Analysis & Design

-Final Phase Document-

「Cafe inventory management system」

Professor : Kristin Chung

Team Name : 공조

Team Members

201220978, Jang Jin Seop, jjs0110zz@naver.com
201420961, Kim So Hee , rlathgml0891@naver.com
201421017, Kim Dong Hoon, kdh94@ajou.ac.kr
201423047, Nam Hwa Su, imyhs2@naver.com
201423055, Hong Min Seok, hong2102@naver.com

Contents

Tables

<Table 4> Glossary.

Figures

1. Vision

1.1. Revision History

Version	Date	Description	Author
Inception draft	Mar 22, 2017	First draft. To be refined elaboration step.	Team 공조
Elaboration 1	Apr 29, 2017	Elaboration 1. Refined version to the previous version.	Team 공조
Elaboration 2/Final	June 05, 2017	Final draft	Team 공조

1.2. Introduction

We plan to design the ‘Cafe Inventory Management System’, which manages the inventory quantities and ingredients’ expiration dates, and this app should support the various customer's business rules and interworking with external systems.

1.3. Business Opportunity

Store Managers at many of the cafes have to write the inventory quantities and ingredients’ expiration dates by hand. It takes a lot of time for them to write themselves out manually and to check out the ingredients’ expiration dates and discard them when it’s over. Also, the variety of ingredients that need to be managed is increasing as the menu gradually grows. For this reason, there is a need for a system to manage ingredients automatically.

1.4. Problem Statement

Existing cafés, every time when ingredients are arrived, Store Manager writes down the ingredients’ expiration dates and holding time directly or use management book to check and manage the whole ingredients. However, if the Store Manager miss it , he may not be able to use the ingredients that has passed the expiration date, so he may throw away the worst ingredients. In addition, the Store Manager needs to check inventory quantities every day and put the order to the supplier as much as necessary, which is not only cumbersome, but also can cause the situation to be missed and the order can not be placed in time.

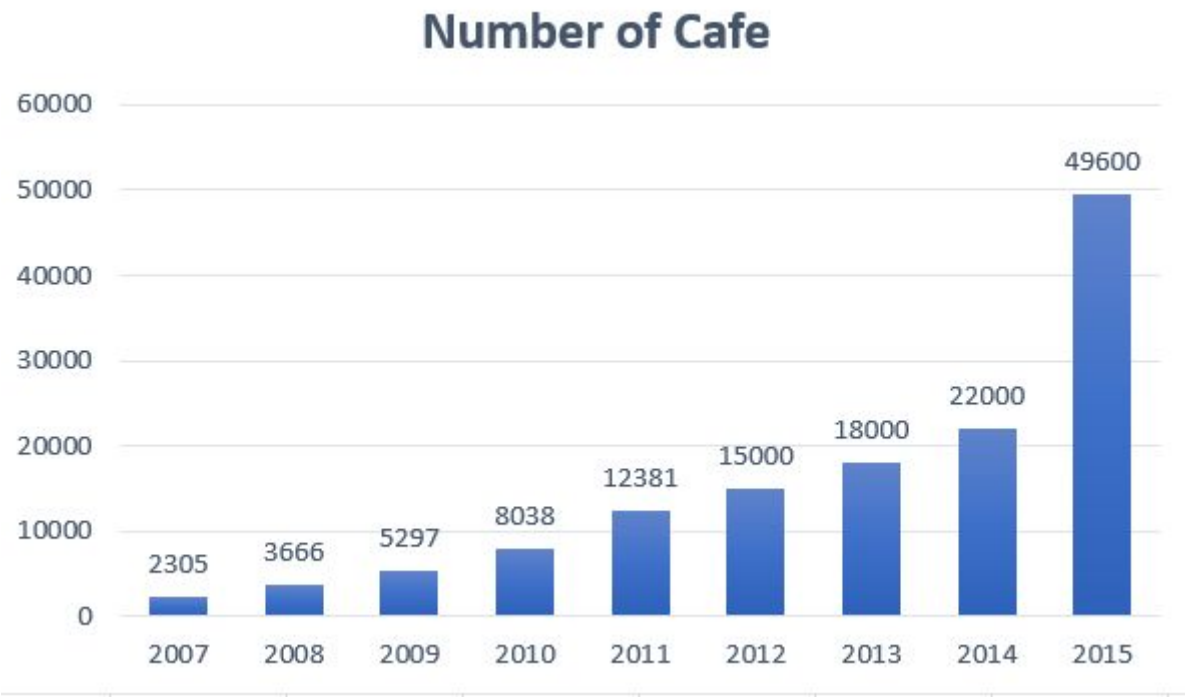
<Figure 1.1> Current cafe Inventory management System

1.5. Product Position Statement

This service will be very useful for Store Managers who want to manage inventory quantities and the ingredients' expiration dates efficiently. The user can check the expiration dates of ingredients and reduce mistakes and errors caused by inventory management.

1.6. Stakeholder Descriptions

1.6.1. Market Demographics



<Figure 1.5> Number of coffee shops according to the year

As shown above, cafes are growing exponentially every year. Additionally , many franchise cafes offer new menus every year or every season.

1.6.2. Stakeholders(non user) Summary

- **customer** : customers who visit the cafe using this system will get better services and drinks made of fresh groceries.
- **cafe brand** : cafe brands such as Starbucks can be certified as a brand of fresh groceries.

1.6.3. User Summary

- **Store Manager** : Manage the inventory quantities and expiration dates of ingredients
- **P.O.S System** : Transfer the information about the menu to be sold to the system.
- **System Administrator** : Register the store that requested the service and manage service maintenance
- **Supplier System** : Get the ordering information sent from the store.
- **Payment System** : Proceed with automatic payment for orders sent from the store.

1.6.4. Key High-Level Goals and Problems of the Stakeholders

High-Level Goal	Priority	Problems and Concerns	Current Solutions
Systematically manage the expiration dates and manage the inventory quantities easily	High	Check the inventory quantity passively and it's inconvenient	No automatic management
		The ingredients' expiration dates are checked passively and it's difficult to find which one is over.	No automatic management
		Inventory quantities and ingredients' expiration dates can be missed or mistaken.	An effort to check accurately and repeatedly
		Consumers distrust the condition of ingredient management.	Customers assume the ingredients' condition by only complete food.
Suppliers of ingredients know the amount of deliveries conveniently and accurately	Middle	If the order is not submitted within the specified time, it will be difficult to prepare the delivery.	Direct contact with the store manager
		The ordering format is different for each store, so management is difficult	No solution

<Table 1.1> Key high-level goals and problems of stakeholders

1.6.5. User-Level Goals

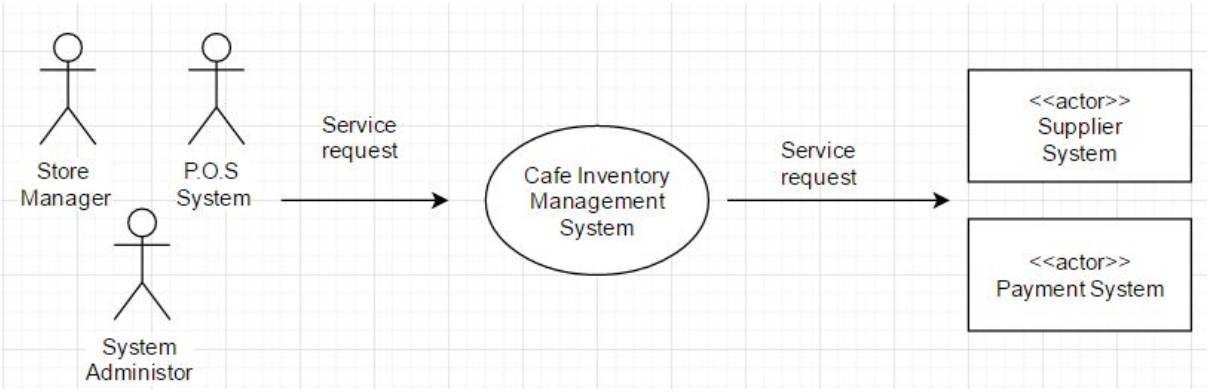
Actor	Goal
Store Manager	Inventory identification, automatic logging, and improved accuracy
	Automatically checks each ingredient expiration date and improves accuracy.
	If no modifications are made, it is automatically ordered
	It is possible to improve the sales to the consumers by forming the trust of ingredients.
P.O.S System	Share newly created menu information
System Administrator	Register or manage the store and maintain the services.

<Table 1.2> User-level goals

1.7. Product Overview

1.7.1. Product Perspective

This cafe inventory management system provides application services that allow Store Managers and ingredient suppliers to easily manage their ingredients. Store Managers can easily and accurately manage the expiration dates of ingredients. Ingredient suppliers can easily manage ingredient status by store. This system can be used not only in cafes but also in restaurants that use food ingredient.



<Figure 1.6> Context Diagram

1.7.2. Summary of Benefits

Supporting Feature	Stakeholder Benefit
Functionally, the system provides services to manage inventory, inventory supply and expiration dates of ingredients.	Automated and rapid inventory management services
Business rules that can connect to branches in various scenarios during inventory management	Building valuable and flexible business logic

Make sure that the ingredients closest to the expiration date is used first, and ingredients that passed the expiration date is not used	It can give consumers a strong sense of trust that they are using fresh food.

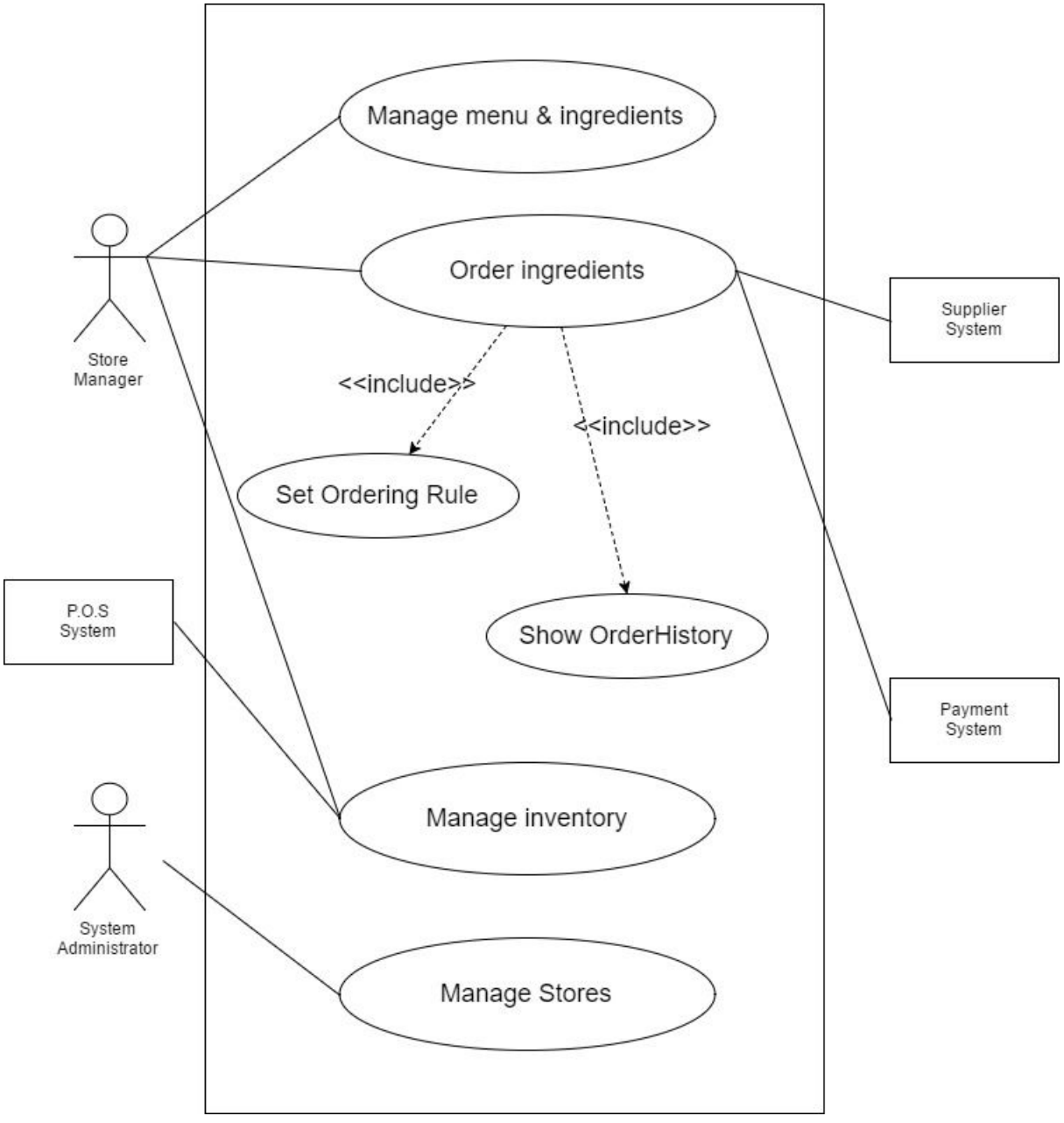
<Table 1.3> Summary of benefits

1.7.3. Summary of System Features

1. Register and modify the information of menu and ingredients
2. Manage the inventory quantity and expiration dates of ingredients
3. Manage the store where request the service.
4. Automatically order to supplier system.
5. Real-time communicates with P.O.S system
6. Alarm when ingredients' expiration date is over.

2. Requirements

2.1. Use Case Diagram



<Figure 2.1> Use Case Diagram

2.2. Use Case Text

2.2.1. Functional Requirements

1) Manage menu & ingredients

The manager selects the Menu Management and the system shows a list of the menu which is currently registered. Manager enters the name of the menu items after registering new menu. The system shows the category of ingredients and manager selects the category which the ingredient is belongs to. The system shows the ingredients belonging to the selected category..The manager selects the required ingredients for the menu item and enters the required amount for the ingredients.After information is entered, the information on the new menu will be registered in the system and can be modified or deleted later.

2) Order Ingredients

매니저는 수동으로 발주를 할 수 있다. 매니저는 재료 발주를 선택하게되면 시스템은 재료 카테고리를 보여준다. 매니저는 발주가 필요한 재료의 카테고리를 선택한다. 선택하게되면 시스템은 선택된 카테고리에 속한 재료리스트를 보여준다. 매니저는 발주가 필요한 재료와 수량을 지정하고 매니저는 발주를 요청한다. 시스템은 발주 정보를 Supplier System에 발주한 후 오나료된 발주 내역을 매니저에게 보여준다.

Managers can place orders manually. When the manager selects material order, the system displays the material category. The manager selects the category of material that requires ordering. When selected, the system displays a list of materials belonging to the selected category. The manager specifies the materials and quantity required for the order, and the manager requests the order. The system orders the supplier system and displays the completed order to the manager.

2-1) Set Ordering Rule

전반적인 자동 발주의 규칙을 설정할 수있다. 매니저가 발주관리를 선택한다. 매니저가 발주 요일과 발주 시간 등록을 시작하면 시스템은 선택가능한 요일과 시간을 보여준다. 매니저가 발주되기를 원하는 요일 및 시간을 선택하면 시스템은 등록된 발주 요일 및 시간을 보여준다. 지정된 발주 요일의 시간이 되면 시스템은 발주가 필요한 재료를 supplier system에 발주한다. 발주할 필요량은 발주량 계산 규칙에 따라 계산된다. 또한 매니저는 언제든지 자동발주와 수동발주를 선택할 수 있다.

2-2) Show Order History

매니저가 언제든지 지난 발주 내역을 확인할 수 있다. 시스템은 발주내역의 목록을 보여준다. 매니저는 확인하고자 하는 발주내역을 선택하면 시스템은 매니저가 선택한 발주내역에 대한 정보를 보여준다.

3) Manage inventory

매니저가 ingredient에 대한 재고관리를 선택하면 시스템은 재료 카테고리를 보여준다. 매니저가 관리할 재료가 있는 카테고리를 선택한다. 시스템은 해당 카테고리에 속한 재료 목록과 각 재료의 수량을 보여준다. 매니저가 관리할 재료를 선택하게 되면 시스템은 선택된 재료에 해당하는 재료아이템들의 정보 (입고일, 수량, 유통기한)를 보여준다. 매니저는 재료아이템을 추가한다. 매니저가 재료아이템의 정보(입고일, 수량, 유통기한)를 입력하면 시스템은 추가된 재료아이템을 포함한 재료의 정보를 보여준다. 정보가 입력된 후 시스템에 새로운 재료아이템의 정보가 등록되고 추후 수정이나 삭제가 가능하다.

4) **Manage store**
시스템은 시스템상의 등록된 매장 현황을 보여준다. 시스템 관리자는 매장 관리를 시작한다. 시스템 관리자는 시스템상의 매장 중 관리 할 매장을 선택한다. 시스템 관리자는 선택된 매장의 정보를 수정하면 시스템은 관자리의 수정된 매장 상태를 저장한다. 시스템은 수정이 완료된 매장 상태를 보여준다.

2.2.2. Use Case Model

Use Case 1 : Manage menu & ingredients

Scope
- Cafe inventory management System

Level
- User-goal

Primary Actor
- Store Manager

Stakeholders and Interests
- Store Manager : 올바른 메뉴와 재료품목에 대한 정보가 잘 관리가 되기를 원한다.
- **Store Manager** : Store manager wants information on the correct menus and ingredient items to be well managed.
- POS System : 시스템으로부터 올바른 메뉴에 대한 정보를 전달받길 원한다.
- **POS System** : POS System wants to get information about the correct menu from the system.

Preconditions
- Store Manager is identified and authenticated.

Postconditions
- 새로운 메뉴가 등록된다.
- New menu is registered.
- 새로운 재료품목이 등록된다.
- New ingredient is registered.
- 각 메뉴를 만드는데 필요한 재료정보가 등록된다.
- Ingredient information required to create each menu is registered.
- 각 재료품목의 입고일로부터의 유통기한, holding time,최소 보유량이 등록된다.
- expiration date, holding time, and minimum holding amount from the receiving date of each ingredient item are registered.

Main Success Scenario
1. 매니저는 메뉴 관리를 선택한다. The manager selects menu management.
2. 시스템은 현재 등록되어 있는 메뉴의 카테고리 목록을 보여준다.
3. 매니저는 관리하고자 하는 메뉴의 카테고리를 선택한다.
4. 시스템은 선택한 카테고리에 속하는 메뉴를 보여준다.
5. 매니저는 메뉴 등록을 시작한다. The manager starts to register the meu.

6. 매니저는 새로 등록할 메뉴아이템의 이름을 입력한다. The manager enters the name of the menu item to be newly registered.
7. 시스템은 재료의 카테고리를 보여준다. The system shows the category of the ingredient.
8. 매니저는 메뉴에 필요한 재료가 속한 카테고리를 선택한다. The manager selects the category which the ingredient needed for the menu is included.
9. 시스템은 해당 카테고리에 속한 재료를 보여준다. The system shows ingredients belonging to that category.
10. 매니저는 메뉴아이템에 추가할 재료를 선택한다. The manager selects the ingredient to add to the menu item.
11. 매니저는 선택한 재료에 대한 필요량을 입력한다. The Manager enters the required amount for the selected ingredient.

매니저는 해당 메뉴 아이템에 필요한 모든 재료에 대해 7~9을 반복한다.
The manager repeats steps 5 through 9 for all the ingredients needed for the menu item.

12. 매니저는 메뉴에 필요한 재료정보입력을 종료한다.
13. 시스템은 추가된 메뉴를 포함한 메뉴 목록을 보여준다. The system displays a list of menus with additional menus.

매니저는 필요한 모든 메뉴에 대해 단계 3~12를 반복한다.
The manager repeats steps 3 through 10 for all necessary menus.

14. 매니저는 메뉴 등록을 종료한다. The manager ends menu registration.

Extensions (Alternative Flows)

***a.** 언제든지 , 시스템이 실패한 경우 At any time, if the system fails

1. 매니저는 시스템을 재시작하고 이전 상태로의 복구를 요청한다. The manager restarts the system and requests recovery to the previous state.
2. 시스템은 이전 상태로 복구한다. The system returns to the previous state.
   2a. 시스템이 복구되는 데 방해가 되는 예외적인 것들이 발생하는 경우 There are exceptions that interfere with the recovery of the system.
   1. 시스템은 매니저에게 오류를 표시하고 , 이를 기록하여 초기의 깨끗한 상태로 돌아간다. The system displays an error to the manager, records it, and returns to the initial state.
   2. 매니저는 시스템을 사용한다. The manager uses the system.

***b**. 언제든지, 매니저가 새로운 재료품목 추가를 원할 경우 At any time, if the manager wants to add a new ingredient item.

1. 매니저는 재료품목에 대한 추가를 선택한다. The manager selects to add to the ingredient item.
2. 시스템은 재료 카테고리를 보여준다. The system shows the ingredient category.
3. 매니저는 생성하고자하는 재료가 속한 카테고리를 선택한다. The manager selects the category where ingredient to be created is included.
4. 시스템은 카테고리에 속한 재료목록을 보여준다. The system displays a list of ingredients in the category.
5. 매니저는 추가할 재료의 이름을 입력한다. The manager enters the name of the ingredient to be added.
   5a. 매니저가 추가할 재료품목의 이름을 잘못 입력한 경우If the manager incorrectly entered the name of the ingredient item to be added.
   1. 매니저는 올바른 재료품목의 이름을 입력한다. The manager enters the name of the correct ingredient item.
6. 매니저는 추가할 재료품목의 정보(유통기한, 홀딩타임, 최소보유량)를 입력한다. The manager enters the information of the ingredient items to be added (expiration date, holding time, minimum holding amount).

6a. 매니저가 추가할 재료품목의 정보를 잘못 입력한 경우 If the manager incorrectly entered the name of the ingredient item to be added.

1. 매니저는 올바른 재료품목의 정보를 입력한다. The manager enters the correct ingredient item information.

7. 시스템은 추가된 재료품목을 포함한 재료품목의 목록을 보여준다. The system displays a list of ingredient items, including added ingredient items.

***c**. 언제든지, 매니저가 재료의 정보 수정을 원할 경우 At any time, if the manager wants to modify the ingredient's information

1. 매니저는 재료품목에 대한 수정을 선택한다. The manager selects to modify the ingredient item.
2. 시스템은 재료 카테고리를 보여준다. The system shows the ingredient category.
3. 매니저는 수정하고자하는 재료가 속한 카테고리를 선택한다. The manager selects the category to which the ingredient to be modified belongs.
4. 시스템은 카테고리에 속한 재료 목록을 보여준다. The system displays a list of ingredients in the category.
5. 매니저는 수정할 재료품목을 선택한다. The manager selects the ingredient item to modify.
6. 시스템은 선택한 재료의 정보(유통기한,홀딩타임,최소보유량)를 보여준다. The system displays information on the selected ingredients (expiration date, holding time, minimum holding amount).
7. 매니저는 수정할 재료의 정보를 입력한다. The manager enters the information of the ingredient to be modified.
8. 시스템은 해당 카테고리의 수정된 재료품목 목록을 보여준다. The system displays a list of modified ingredient items for that category.

***d.** 언제든지, 매니저가 재료의 삭제를 원할 경우 At any time, if the manager wants to delete the ingredient.

1. 매니저는 재료에 대한 삭제를 선택한다. The manager selects to delete the ingredient.
2. 시스템은 재료 카테고리를 보여준다. The system shows the ingredient category.
3. 매니저는 삭제하고자하는 재료가 속한 카테고리를 선택한다. The manager selects the category to which the ingredient to be deleted belongs.

3a. 선택한 재료의 재료item이 존재하는 경우 If the ingredient item of the selected ingredient is present.

1. 시스템은 선택한 재료품목에 대한 재료item이 존재함을 알린다. The system announces that there is a ingredient item for the selected ingredient item.

2. 매니저는 재료 품목 삭제를 취소한다. The manager cancels the deletion of the ingredient item.

2a. 매니저가 재료품목 삭제를 원하는 경우 If the manager wants to delete ingredient items.

1. 매니저는 재료품목 삭제를 요청한다. The manager requests to delete the ingredient item.

2. 시스템은 해당 재료품목을 삭제한다. The system deletes the selected ingredient.

3. 시스템은 재료 카테고리를 보여준다. The system shows the ingredient category.

4. 시스템은 카테고리에 속한 메뉴 목록을 보여준다. The system displays a list of menus belonging to the category.
5. 매니저는 삭제할 재료품목을 선택한다. The manager selects the ingredient item to be deleted.
6. 시스템은 해당 카테고리에 남아있는 재료품목의 목록을 보여준다. The system displays a list of ingredient items remaining in the category.

3a. 매니저가 카테고리를 잘못 선택한 경우,

1. 매니저는 메뉴 카테고리를 다시 보여주기를 요청한다.
2. 시스템은 단계2로 돌아가 메뉴 카테고리를 보여준다.

5a. 매니저가 메뉴정보 수정을 원할 경우 If the manager wants to modify menu information.

1. 매니저는 메뉴 정보 수정을 시작한다. The manager starts modifying the menu information.
2. 매니저는 수정할 메뉴를 선택한다. The manager selects the menu to be modified.
3. 시스템은 선택된 메뉴를 구성하는 재료와 각 재료의 필요량을 보여준다. The system shows the ingredients that make up the selected menu and the amount of each ingredient needed.
4. 매니저는 수정할 재료를 선택한다. The manager selects the ingredient to be modified.

4a. 수정하고자 하는 재료가 메뉴정보에 존재하지 않는 경우 If the ingredient manager wants to modify does not exist in the menu information.

1. 확장 시나리오 *b 재료품목 추가 Process를 통해 재료품목을 추가한다. Add ingredient items through the extension scenario * b

5. 매니저는 해당 재료의 필요량을 수정한다. The manager modifies the required amount of the ingredient.

5a. 매니저가 해당 재료를 메뉴구성 재료에서 삭제하고싶은 경우 If the manager wants to remove the ingredient from the menu component.

1. 매니저는 해당 메뉴재료 삭제를 요청한다. The manager requests to delete the selected menu ingredient.

6. 시스템은 수정된 메뉴를 포함한 메뉴 목록을 보여준다. The system displays a list of menus with modified menus.

5b. 매니저가 메뉴정보 삭제를 원할 경우 If the manager wants to delete menu information.

1. 매니저는 메뉴 정보 삭제를 시작한다. The manager starts deleting menu information.
2. 시스템은 메뉴 목록을 보여준다. The system displays a menu list.
3. 매니저는 삭제할 메뉴를 선택한다. The manager selects the menu to be deleted.
4. 시스템은 남아있는 메뉴 목록을 보여준다. The system displays a list of the remaining menus.

6a. 매니저가 새로 추가할 메뉴의 이름을 잘못 입력한 경우 If the manager incorrectly entered the name of the new menu to be added.

1. 매니저는 메뉴 이름을 올바르게 다시 입력한다. The manager re-enters the menu name correctly.

6b. 매니저가 입력한 메뉴 이름의 재료가 이미 존재 할 경우 If the ingredient of the menu name entered by the manager already exists.

1. 시스템은 매니저에게 동일한 이름의 메뉴가 존재한다고 알린다. The system informs the manager that there is a menu of the same name.
2. 시스템은 기존에 존재하는 동일한 이름의 메뉴의 정보를 보여준다. The system displays the information of the existing menu of the same name.
3. 매니저는 진행중인 메뉴아이템 추가를 취소한다. The manager cancels the addition of ongoing menu items.

   3a. 매니저가 동일한 이름의 메뉴 아이템을 생성하고 싶은 경우 If the manager wants to create a menu item with the same name.

   1. 매니저는 동일한 이름의 메뉴 아이템 생성을 요청한다. The manager requests to create a menu item of the same name.
   2. 시스템은 기존 메뉴이름에 번호를 추가하여 새로운 메뉴를 생성한다. The system adds a number to the existing menu name to create a new menu.

3. Main Success scenario의 5단계부터 진행한다. Proceed from step 5 of the Main Success scenario.

4. 시스템은 메뉴 목록을 보여준다. The system displays a menu list.

10a. 매니저가 추가할 메뉴에 들어가는 재료품목을 잘못 선택한 경우 If the manager incorrectly selects the ingredient item to be added to the menu.

1. 매니저는 재료품목을 올바르게 선택한다. The manager selects the ingredient item correctly.

10b. 추가할 메뉴에 들어가는 재료품목이 없을 경우 If there is no ingredient item in the menu to be added.

1. 확장시나리오 *b 재료품목 추가 Process를 통해 재료 품목을 추가한다. Add the ingredient item through the extension scenario * b.

11a. 재료품목의 개수를 잘못 입력한 경우. If the manager enters the number of ingredient items incorrectly.

1. 재료품목의 개수를 올바르게 다시 입력한다. Re-enter the number of ingredient items correctly.

Special Requirements

Technology and Data Variations List

Frequency of Occurrence: Often used by users

Open Issues

원격 서비스 복구 주제에 대한 조사 필요 Need to investigate remote service recovery topics

Use Case 2 : Order Ingredients

Scope
- Cafe inventory management System

Level
- User-goal

Primary Actor
- Store Manager

Stakeholders and Interests
- Store Manager : 매니저는 언제든지 자신이 원하는 수량만큼 재료를 수동으로 발주 할 수 있다.
- Store Manager :

Preconditions
- Store Manager is identified and authenticated.
- 재료들의 목록을 가지고 있다.

Postconditions
- 발주가 완료된다.
- 발주가 승인된다.
- 발주 내역이 기록된다.
- The order day and the order time are registered.
- Ordering is completed.
- The order is approved.
- Order details are recorded.

Main Success Scenario
1. 매니저는 재료 발주를 선택한다. Manager selects Order Management.
2. 시스템은 재료 카테고리를 보여준다.
3. 매니저는 발주가 필요한 재료의 카테고리를 선택한다..
4. 시스템은 선택된 카테고리에 속한 재료리스트를 보여준다.
5. 매니저는 발주가 필요한 재료와 수량을 입력한다.-

매니저는 발주가 필요한 모든 재료에 대해 2~5를 반복한다.

6. 매니저는 발주를 요청한다.
7. 시스템은 매니저가 지정한 정보대로 supplier system에 발주한다. the system makes an order to the supplier system as the manager decided.
8. 시스템은 완료된 발주 내역을 매니저에게 보여준다. The system shows the ordering histories to the manager.

Extensions (Alternative Flows)

***a.** 언제든지 , 시스템이 실패한 경우 At any time, if the system fails
1. 매니저는 시스템을 재시작하고 이전 상태로의 복구를 요청한다. The manager restarts the system and requests recovery to the previous state.
2. 시스템은 이전 상태로 복구한다. The system returns to the previous state.
   2a. 시스템이 복구되는 데 방해가 되는 예외적인 것들이 발생하는 경우 There are

exceptions that interfere with the recovery of the system.

1. 시스템은 매니저에게 오류를 표시하고 , 이를 기록하여 초기의 깨끗한 상태로 돌아간다. The system displays an error to the manager, records it, and returns to the initial state.
2. 매니저는 시스템을 사용한다. The manager uses the system.

7a. 시스템이 Supplier System과 통신하는 데 실패한 경우 If the system fails to communicate with the Supplier System.

1. 시스템은 Supplier 단말기와 통신 서비스를 재시작하고 계속한다. The system restarts and continues the Supplier terminal and communication service.

   1a. 시스템은 서비스 재시작이 안 되는 것을 발견한다.if the system detects that the service can not be restarted.

   1. 시스템은 오류를 보여준다. The system displays an error.
   2. 매니저는 판매된 메뉴를 파악하고 메뉴를 구성하는 재료를 직접 계산해서 해당재료의 재고수량을 수정한다. The manager grasps the menus sold and directly calculates the ingredients that make up the menu to modify the inventory quantity of that ingredient.

Special Requirements

- Supplier system의 서비스 실패와 같은 원격서비스에 접근하는 경우, 제대로 복구되기를 원한다.
- When accessing remote services such as service failures in the Supplier system, they want to be restored correctly.

Technology and Data Variations List

Frequency of Occurrence: Often used by users

Open Issues

- **원격 서비스 복구 주제에 대한 조사 필요** Need to investigate remote service recovery topics

Use Case 2-1 : Set OrderingRule

Scope
- Cafe inventory management System

Level
- Subfunction

Primary Actor
- Store Manager

Stakeholders and Interests
- Store Manager : 매니저가 직접 발주를 하지 않아도 , 시스템이 필요한 재료의 양을 계산하여 발주를 진행하도록 자동 발주가 진행될 요일과 , 시간을 설정 할 수 있다.

Preconditions
- Store Manager is identified and authenticated.

Postconditions
- 발주 요일과 발주 시간이 등록된다.
- 자동 발주 여부가 결정된다.
- The order day and the order time are registered.
- Ordering is completed.
- The order is approved.
- Order details are recorded.
- The order status is specified.

Main Success Scenario
1. 매니저는 발주관리를 선택한다. Manager selects Order Management.
2. 매니저는 발주 요일과 발주시간 등록을 시작한다. The manager begins registering the date of the order and ordering time on the order day.
3. 시스템은 선택 가능한 요일과 시간을 보여준다. The system shows the selectable days and hours of the week.
4. 매니저는 발주되기를 원하는 요일 및 시간을 선택한다. The manager selects the day and time of day for which the manager wants to be placed.
5. 시스템은 등록된 발주 요일 및 시간을 보여 준다. The system shows the date and time which are registered.

Extensions (Alternative Flows)

***a.** 언제든지, 시스템이 실패한 경우 At any time, if the system fails
1. 매니저는 시스템을 재시작하고 이전 상태로의 복구를 요청한다. The manager restarts the system and requests recovery to the previous state.
2. 시스템은 이전 상태로 복구한다. The system returns to the previous state.
   2a. 시스템이 복구되는 데 방해가 되는 예외적인 것들이 발생하는 경우 There are exceptions that interfere with the recovery of the system.
      1. 시스템은 매니저에게 오류를 표시하고 , 이를 기록하여 초기의 깨끗한 상태로 돌아간다. The system displays an error to the manager, records it, and returns to the initial state.
      2. 매니저는 시스템을 사용한다. The manager uses the system.

***b.** 언제든지, 매니저가 자동 발주 기능을 선택 또는 해제하고 싶을 경우

1. 매니저는 발주관리를 선택한다. Manager selects Order Management.
2. 매니저는 자동 발주 상태 변경을 선택한다.
3. 시스템은 기존의 자동 발주 상태를 보여준다.
4. 매니저는 자동 발주 상태를 원하는 대로 변경한다.
5. 시스템은 변경된 자동 발주 상태를 보여준다.

2a. 발주 요일과 발주 시간이 이미 등록되어 있는 경우 if order date and order time are already registered.

1. 시스템은 현재 등록되어 있는 발주 요일과 발주 시간을 보여준다.The system shows the date and time of the order on which the order is currently registered.
2. 매니저가 발주 요일 또는 발주 시간의 변경을 원하는 경우. If the manager wants to modify the order day or order time.
   2a. main 시나리오 1번부터 그대로 진행 Proceed from the main scenario 1.
3. 매니저가 발주 요일 또는 발주 시간의 변경을 원하지 않는 경우 If the manager does not want to modify the order of the order day or order time
   3a. main 시나리오 7번부터 시작한다. Proceed from the main scenario 7.

4a. 매니저가 발주 요일과 발주 시간을 잘못 선택한 경우 If the manager incorrectly selects the ordering date and ordering time.

1. 매니저는 선택가능한 요일과 시간을 보여주기를 요청한다.
2. 시스템은 선택 가능한 요일과 시간을 보여준다.
3. 매니저는 원하는 발주 요일과 시간을 다시 선택한다. The manager will re-select the desired ordering date and time.

Special Requirements

Technology and Data Variations List

Frequency of Occurrence: Often used by users

Open Issues

Use Case 2-2 : Show Order History

Scope
- Cafe inventory management System

Level
- Subfuction

Primary Actor
- Store Manager

Stakeholders and Interests
- Store Manager : 이전에 자동 또는 수동으로 발주된 모든 발주에 대한 발주내역 (발주날짜와 시간 , 각 재료별 발주량)을 확인 할 수 있다.

Preconditions
- 이전에 자동 또는 수동으로 발주된 모든 발주에 대한 발주내역이 저장되어 있다.

Postconditions

Main Success Scenario
1. 매니저는 발주관리를 선택한다.
2. 매니저는 발주내역 확인을 선택한다.
3. 시스템은 발주내역의 목록을 보여준다.
4. 매니저는 발주내역 목록 중 확인하고자 하는 발주내역을 선택한다.
5. 시스템은 매니저가 선택한 발주내역에 대한 정보(발주 날짜, 발주 시간 , 발주된 재료목록과 각 재료의 발주 수량)을 보여준다.

Extensions (Alternative Flows)
3a. 매니저가 발주내역을 잘 못 선택한 경우,
1. 매니저는 시스템에 발주내역 목록을 다시 보여주기를 요청한다.
2. 시스템은 단계2로 돌아간다.

Special Requirements

Technology and Data Variations List

Frequency of Occurrence: Often used by users

Open Issues

Use Case 3 : Manage inventory

Scope
- Cafe inventory management System

Level
- User-goal

Primary Actor
- Store Manager

Stakeholders and Interests
- Store Manager : 재고의 수량과 유통기한을 확인하기 위해 전체 제품을 일일이 찾아봐야 하는 번거로움을 줄이는 것을 원한다.
- Store Manager : Store Manager wants to reduce the hassle of finding the whole product to check inventory quantity and expiration date.
- POS system

Preconditions
- Store Manager is identified and authenticated.

Postconditions
- 시스템상의 재료의 수량과 유통기한이 갱신된다.
- The quantity and expiration date of ingredients of the system are updated.
- 시스템의 재고량과 현 재고량이 일치한다.
- The amount of inventory in the system matches the current inventory.

Main Success Scenario
1. 매니저는 ingredient에 대한 재고관리를 선택한다. The manager selects inventory management for the ingredient.
2. 시스템은 재료 카테고리를 보여준다. The system shows the ingredient category.
3. 매니저는 관리할 재료가 있는 카테고리를 선택한다. The manager selects the category with the ingredient to manage.
4. 시스템은 해당 카테고리에 속한 재료 목록과 각 재료의 수량을 보여준다. The system displays a list of ingredients in that category and the quantity of each ingredient.
5. 매니저는 관리할 재료를 선택한다. The manager selects the ingredient to be managed.
6. 시스템은 선택된 재료에 해당하는 재료아이템들의 정보 (입고일 , 수량 , 유통기한)를 보여준다. The system shows the information (receiving date, quantity, expiration date) of the ingredient items corresponding to the selected ingredient.
7. 매니저는 재료아이템을 추가한다. The manager adds ingredient items.
8. 매니저는 새로 입고된 재료아이템의 정보(입고일 , 수량 , 유통기한)를 입력한다. The manager enters the information of the new ingredient item (receiving date, quantity, expiration date).
9. 시스템은 추가된 재료아이템을 포함한 재료의 정보를 보여준다. The system displays ingredient information, including added ingredient items.

매니저는 모든 입고된 재료아이템에 대하여 단계 2~9를 반복한다.
The manager repeats steps 2 through 9 for all ingredients received.

10. 매니저는 재고 관리를 종료한다. The manager ends inventory management.

Extensions (Alternative Flows)

***a.** 언제든지 , 시스템이 실패한 경우 At any time, if the system fails

1. 매니저는 시스템을 재시작하고 이전 상태로의 복구를 요청한다. The manager restarts the system and requests recovery to the previous state.
2. 시스템은 이전 상태로 복구한다. The system returns to the previous state.
   2a. 시스템이 복구되는 데 방해가 되는 예외적인 것들이 발생하는 경우 There are exceptions that interfere with the recovery of the system.
   1. 시스템은 매니저에게 오류를 표시하고 , 이를 기록하여 초기의 깨끗한 상태로 돌아간다. The system displays an error to the manager, records it, and returns to the initial state.
   2. 매니저는 시스템을 사용한다. The manager uses the system.

***b.** At any time, if the expiration date of ingredients remains within 3 days

1. The system informs to the Store Manager that there are x days left until the expiration date of ingredients.

***c.** 언제든지, 유통기한이 지난경우 At anytime, if the expiration date is over

1. 시스템은 카페매니저에게 재료의 유통기한이 현 시간부로 지났음을 알린다.The system informs the Store Manager that the expiration date of the ingredient has passed the current date.

1a. 단계 1이 진행된 후에도 같은 재료가 제거되지 않은 경우 If the same ingredient is not removed after Step 1 has been carried out.

1. 1시간마다 매니저에게 어떤 재료의 유통기한이 얼마나 지났는지 알린다. System Informs the manager how long the expiration date of the ingredient material has passed.

1c. P.O.S system에서 메뉴가 판매된 경우 If the menu is sold in POS system

1. 연동된 시스템(P.O.S System)은 판매된 메뉴와 수량을 시스템에 전송한다. The linked system (P.O.S. System) sends the sold menu to the system.
2. 시스템은 전송받은 메뉴를 구성하는 재료를 파악한다. The system identifies the ingredients that make up the menu that is transmitted.
3. 시스템은 해당재료의 재고수량을 수정한다. The system modifies the stock quantity of the ingredient.

3a. 수정하고자 하는 재료에 대해 삭감해야하는 재료량보다 존재하는 재료량이 적은 경우If there is less amount of ingredient that needs to be cut down than amount of ingredient that you want to modify..

1. 시스템은 수정하고자 하는 재료가 부족함을 알린다. The system informs that there is insufficient ingredient to modify.

1c-1a. 시스템이 P.O.S System과 통신하는 데 실패한 경우 If the system fails to communicate with P.O.S System

1. 시스템은 POS register 와 통신 서비스를 재시작하고 계속한다. The system restarts the POS register and communication services and continues.
   1a. 시스템은 서비스 재시작이 안 되는 것을 발견한다.The system detects that the service can not be restarted.
   1. 시스템은 오류를 보여준다. The system displays an error.
   2. 매니저는 판매된 메뉴를 파악하고 메뉴를 구성하는 재료를 직접 계산해서 해당재료의 재고수량을 수정한다. The manager grasps the menus sold and directly calculates the ingredients that make up the menu to modify the inventory quantity of that ingredient.

5a. 매니저가 재료 선택을 잘못한 선택한 경우 If the manager incorrectly selects the ingredient

1. 매니저는 이전에 선택한 카테고리에 속한 재료 목록과 각 재료의 수량을 보여주기를 시스템에 요청한다.
2. 메인 시나리오 4번부터 다시 시작한다.

5b. 선택하고자하는 재료항목이 카테고리에 없는 경우 If the ingredient item you want to select is not in the category

1. 새로운 재료품목을 추가하는 Use Case 1번의 확장 ***b** 시나리오를 수행한다. Perform Use Case 1 extension * b scenario to add new ingredient item.

7a. 매니저가 재료아이템 정보 수정을 원할 경우 If the manager wants to modify ingredient item information

1. 매니저는 수정하고자 하는 재료아이템을 선택한다. The manager selects the ingredient item to modify.
2. 매니저는 실 재고량에 맞춰 재료아이템의 정보를 수정한다. The manager modifies the ingredient item information to match the actual inventory.
3. 시스템은 수정한 재료를 포함한 재료들의 정보를 보여준다. The system displays information about the ingredients including the modified ingredient.

7b. 매니저가 재료아이템 삭제를 원할 경우 If the manager wants to delete a ingredient item.

1. 매니저는 재료아이템의 삭제를 요청한다.. The manager asks for a deletion of the ingredient items.
2. 시스템은 남아있는 재료아이템들을 보여준다. The system shows the remaining ingredient items.

8a. 매니저가 재료수량 정보를 입력하지 않은 경우 If the manager has not entered ingredient quantity information

1. 시스템은 매니저에게 재료수량을 입력하지 않았음을 알린다. The system informs the manager that the ingredient quantity has not been entered.
2. 매니저는 메인 시나리오의 단계8부터 다시 진행한다. The manager goes back to Step 8 of the main scenario.

8b. 매니저가 입고날짜를 입력하지 않은 경우 The manager has not entered a receiving date.

1. 시스템은 매니저에게 입고날짜를 입력하지 않았음을 알린다. The system informs the manager that it has not entered a receiving date.
2. 시스템은 현재 날짜를 입고날짜로 입력한다. The system enters the current date as the receiving date.
3. 매니저는 메인 시나리오의 단계9부터 다시 진행한다. The manager goes back to Step 9 of the main scenario.

Special Requirements

- P.O.S System의 서비스 실패와 같은 원격서비스에 접근하는 경우, 제대로 복구되기를 원한다. If you access a remote service such as P.O.S System service failure, you want to be recovered properly.

Technology and Data Variations List

Frequency of Occurrence: Often used by users

Open Issues

- **원격 서비스 복구 주제에 대한 조사 필요** Need to investigate remote service recovery topics

Use Case 4 : Manage store

Scope
- Cafe inventory management System

Level
- User-goal

Primary Actor
- Store Manager

Stakeholders and Interests
-

Preconditions
- Store Manager is identified and authenticated.

Postconditions
- 매장에 대한 정보가 변경된다.

Main Success Scenario
1. 시스템 관리자는 매장 관리를 시작한다.
2. 시스템은 시스템상의 등록된 매장 현황을 보여준다.
3. 시스템 관리자는 시스템상의 매장 중 관리 할 매장을 선택한다.
4. 시스템 관리자는 선택된 매장의 정보를 수정한다.

시스템 관리자는 모든 수정할 매장에 대하여 단계 2~4를 반복한다.

5. 시스템 관리자는 시스템에 수정된 매장 상태를 반영하기를 요청한다.
6. 시스템은 수정이 완료된 매장 상태를 보여준다.

Extensions (Alternative Flows)

***a.** 언제든지 , 시스템이 실패한 경우 At any time, if the system fails
1. 매니저는 시스템을 재시작하고 이전 상태로의 복구를 요청한다. The manager restarts the system and requests recovery to the previous state.
2. 시스템은 이전 상태로 복구한다. The system returns to the previous state.
   2a. 시스템이 복구되는 데 방해가 되는 예외적인 것들이 발생하는 경우 There are exceptions that interfere with the recovery of the system.
3. 시스템은 매니저에게 오류를 표시하고 , 이를 기록하여 초기의 깨끗한 상태로 돌아간다. The system displays an error to the manager, records it, and returns to the initial state.
4. 매니저는 시스템을 사용한다. The manager uses the system.

3a. 관리자가 관리할 매장을 잘못 선택한 경우
1. 관리자는 등록된 매장 현황을 다시 보여주기를 요청한다.
2. 시스템은 단계2로 돌아가 매장 현황을 보여준다.

3a. 관리자가 원하는 매장이 없는경우
1. 관리자는 새로운 매장에 대한 추가를 선택한다.
2. 관리자는 추가할 매장에 대한 정보를 입력한다.

2a. 관리자가 추가할 매장에 대한 정보를 잘못 입력한 경우
1. 관리자는 올바른 매장의 정보를 입력한다.

3. 시스템은 추가된 매장을 포함한 매장 현황을 보여준다.

4a. 관리자가 선택된 매장의 정보 삭제를 원할경우
1. 관리자는 해당 매장 삭제를 요청한다.
2. 시스템은 남아있는 매장의 현황을 보여준다.

2.2.3. Non-functional Requirements

1) Revision History

Version	Date	Description	Author
Inception draft	Mar 22, 2017	First draft. To be refined elaboration step.	Team 공조
Elaboration 1	Apr 29, 2017	Elaboration 1. Refined version to the previous version.	Team 공조
Elaboration 2 / Final	June 05, 2017	Final draft	Team 공조

2) Introduction

이 문서에서 Cafe inventory management system의 요구사항 중 유스케이스에서 추출하지 못한 모든 요구사항이 작성된다. Requirements of this system which are not expressed in the use cases are treated in this document.

3) Usability Requirement

@ Help

- If user has difficulty in using this service, user can get support from the system administrator. System administrator provides users with manuals, system developer's e-mail addresses, and FAQs.
- 사용자 편의를 위해 초기 사용시 메뉴와 재료에 대한 기본적인 샘플 데이터를 제공합니다.
- 시스템이 실행중이지 않은 상태에서 P.O.S 사용이 있을 경우 자동적으로 시스템을 실행합니다.

4) Reliability Requirement

@ Recoverability

- 시스템은 한 달에 1번, 1시간씩 점검됩니다 (오전 2시 부터 새벽3시).
System will be checked 2 times a month for 30 minutes each. (02:00AM ~ 02:30AM).
- 외부 서비스를 사용할때 문제가 발생하면 해결방법을 강구해야 한다.
- 시스템 검사 중에 시스템이 백업됩니다. System will be backed up during system check.

@ Security

- user가 authentication을 시도 하거나 데이터 베이스의 값을 가져오고자 할 때 input을 validation 한다.
- Security is the capability of a system to prevent malicious or accidental actions outside of the designed usage, and to prevent disclosure or loss of information. A secure system aims to protect assets and prevent unauthorized modification of information

5) Performance Requirement

@ Response Time (Latency)

재고나 재료 정보를 업데이트하는 데 걸리는 시간, 기존의 재고나 재료정보, 발주내역을 조회하는데 걸리는 시간

- Updating inventory or ingredient information is done within 5 seconds 90% of time.
- View existing inventory or ingredient information and order history is done within 3 seconds 90% of

time.
@Throughput(일정 시간동안 처리할 수 있는 이벤트 수?)
- 5초 이내에 주요 이벤트 N개 처리

6) Supportability Requirement
@ Adaptability
- 사용자는 많은 다른 OS를 사용합니다. Users use many different operating systems.
- 따라서 Cafe inventory management system 서비스는 모든 Operating system에서 잘 작동합니다.
- 어떠한 종류의 P.O.S system과도 연동이 가능합니다.
Therefore, the cafe inventory management system service works well on all operating systems.

7) Privacy Requirement
-각 매장과 납품업체에 대한 매출, 메뉴, 발주 정보는 해당 매장의 매니저 , 납품업체의 관리자 이외에는 열람할 수 없습니다.

8) Portability Requirement
-매장 매니저는 SIS 시스템을 매장 안에서 뿐만 아니라 어디서든 사용하여 재고와 발주 상황을 확인할 수 있다.

9) Profit Structure
- 이 서비스를 이용하고자 하는 카페 회사와 식품 납품업체로부터 매달 정해진 금액의 사용료를 받는 형태로 수익 창출
- Make a profit by receiving the monthly fee from the cafe company and the ingredient supplier who want to use this service.

3. Domain Model

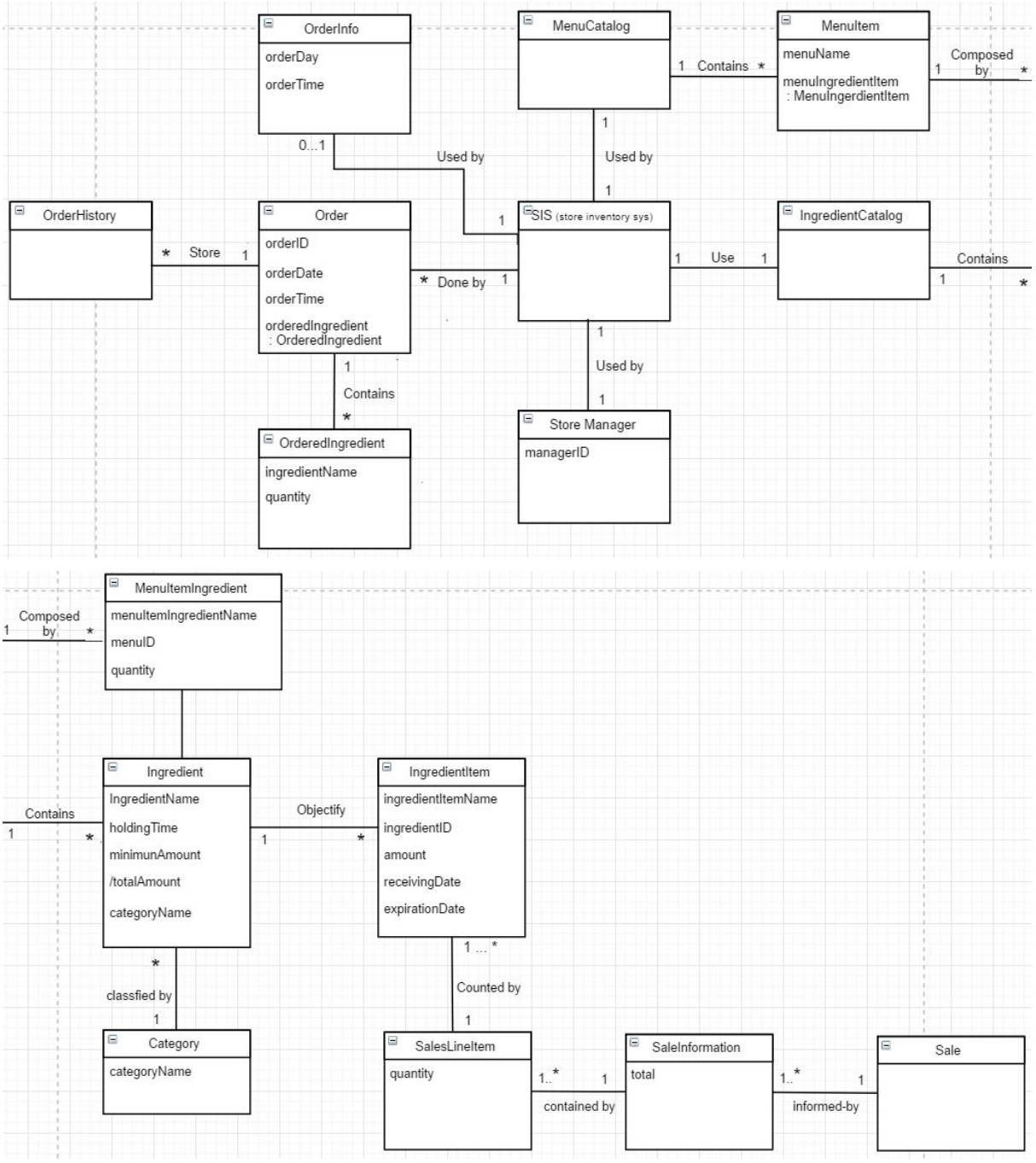
3.1. Revision History

Version	Date	Description	Author
Inception draft	Mar 22, 2017	First draft. To be refined elaboration step.	Team 공조
Elaboration 1	Apr 29, 2017	Elaboration 1. Refined version to the previous version.	Team 공조
Elaboration 2 / Final	June 05, 2017	Final draft	Team 공조

3.2. Domain Model Diagram

<Figure 3.1> Domain Model Diagram

(This Picture is difficult to see. So we distribute 2 parts below.)



3.2.1. Domain Classes and Attributes

1) StoreManager

class Description : Manager who uses a system for managing stores in stores

Attribute	Description
managerID	Manager's ID
Association	Description
Store Manager(1) - SIS(1)	Store Manager uses SIS

<Table 3.1>Description of Domain class 'StoreManager'

2) SIS (Store inventory system)

class Description : Application that Store Manager uses on its device

Attribute	Description
-	-
Association	Description
SIS(1) - IngredientCatalog(1)	SIS uses IngredientCatalog
SIS(1) - MenuCatalog(1)	SIS uses MenuCatalog
SIS(1) - Store Manager(1)	SIS is used by StoreManager
SIS(1) - AutoOrderRule(0...1)	AutoOrderRule is defined in SIS
SIS(1) - Order(*)	SIS proceeds Order
SIS(1) - Order(1)	SIS records OrderHistory
SIS(1) - Store(0….*)	SIS manages Store
SIS(1) - Administrator(1)	SIS is used by Administrator

<Table 3.2>Description of Domain class 'SIS''

3) NewIngredientItem

class Description : 새롭게 추가하고자 하는 재료아이템

Attribute	Description
ingredientitemID	재료아이템의 고유 ID
ingredientID	재료의 고유 ID Unique ID of ingredient

ingredientItemName	재료의 이름 name of ingredient
amount	재료의 양 quantity of ingredient
receivingDate	매장에 입고된 날짜 Date which the ingredient entered in the store
expirationDate	재료가 사용 가능한 날짜
Association	Description
NewIngredientItem(1) - IngredientItem(1)	NewIngredientItem becomes IngredientItem

<Table 3.3>Description of Domain class ‘NewIngredientItem’

4) IngredientItem

class Description : Ingredient items per each received date

Attribute	Description
ingredientitemID	추가 하고자하는 재료아이템의 고유 ID
ingredientID	추가하고자하는 재료의 고유 ID Unique ID of ingredient
ingredientItemName	추가하고자하는 재료의 이름 name of ingredient
amount	추가하고자하는 재료의 양 quantity of ingredient
receivingDate	매장에 입고된 날짜 Date which the ingredient entered in the store
expirationDate	재료가 사용 가능한 날짜
Association	Description
IngredientItem(1) - NewIngredientItem(1)	NewIngredientItem becomes IngredientItem
IngredientItem(*) - Ingredient(1)	Ingredient Objectifies IngredientItem
IngredientItem(1…*) - SalesLineItem	IngredientItem is counted by SalsLineItem

<Table 3.4> Description of Domain class ‘IngredientItem’

5) Ingredient

class Description : Ingredient to use in stores

Attribute	Description
ingredientID	재료의 고유 ID Unique ID of ingredient
ingredientName	재료의 이름

holdingTime	재료를 유지 할 수 있는 기간 Period for maintaining ingredients
minimumAmount	영업시 최소로 보유 해야 하는 양 Minimum quantity of the ingredient during the operating hour
totalAmount	그 재료의 모든 양 sum of the all the ingredient items which belongs to same ingredient
category	그 재료가 속해있는 카테고리 category where the ingredients belongs.
Association	Description
Ingredient(1) - MenuItemIngredient(1)	MenuItemIngredient represents Ingredient
Ingredient(1) - IngredientItem(1...*)	Ingredient Objectifies IngredientItem
Ingredient(*) - IngredientCatalog(1)	IngredientCatalog contains Ingredient

<Table 3.5> Description of Domain class ‘Ingredient’

6) ingredientCatalog

class Description : list of ingredients

Attribute	Description
-	-
Association	Description
IngredientCatalog(1) - Ingredient(*)	IngredientCatalog contains Ingredient
IngredientCatalog(1) - SIS(1)	SIS uses IngredientCatalog.

<Table 3.6>Description of Domain class ‘ingredientCatalog’

7) MenuItemIngredient

class Description : Ingredients needed to create the corresponding menus

Attribute	Description
menuID	메뉴의 고유 ID
ingredientID	재료의 고유 ID
IngredientName	재료의 이름
quantity	메뉴를 구성하는데 해당 재료가 필요한 양
Association	Description

MenuItemIngredient(*) - MenuItem(1)	MenuItem is composed of MenuItem
MenuItemIngredient(1) - Ingredient(1)	MenuItemIngredient represents Ingredient

<Table 3.7> Description of Domain class ‘MenuItemIngredient’

8) NewMenuItem

class Description : 새로 추가하고자 하는 메뉴

Attribute	Description
menuID	menu’s ID
menuName	메뉴의 이름
menuCategory	해당 메뉴가 속하는 카테고리
menuIngredientItem	해당 메뉴를 만들기 위해 필요한 재료 ingredients needed to create the menu
Association	**Description**
NewMenuItem(1) - MenuItem(1)	NewMenuItem becomes MenuItem

<Table 3.8> Description of Domain class ‘NewMenuItem’’

9) MenuItem

class Description : the menu which is sold in the store.

Attribute	Description
menuID	menu’s ID
menuName	메뉴의 이름
menuCategory	해당 메뉴가 속하는 카테고리
menuIngredientItem	해당 메뉴를 만들기 위해 필요한 재료 ingredients needed to create the menu
Association	**Description**
MenuItem(1) - NewMenuItem(1)	NewMenuItem becomes MenuItem
MenuItem (*) - MenuCatalog(1)	MenuCatalog contains MenuItem.
MenuItem(1) - MenuItemIngredient(*)	Menuitem is composed of MenuItemIngredient

<Table 3.9> Description of Domain class ‘MenuItem’’

10) MenuCatalog

class Description : list of menu

Attribute	Description
-	-
Association	Description
MenuCatalog(1) - SIS(1)	SIS uses MenuCatalog
MenuCatalog(1) - MenuItem(*)	MenuCatalog contains MenuItem

<Table 3.10> Description of Domain class ‘MenuCatalog’

11) AutoOrderRule

class Description : information needed to order automatically which includes ordering time and ordering day.

Attribute	Description
autoOrderingDay	자동발주가 진행되는 요일
autoOrderingTime	자동발주가 진행되는 시간
autoOrderFlag	자동발주여부를 결정하는 on/off flag
Association	Description
AutoOrderRule(0...1) - SIS(1)	AutoOrderRule is defined in SIS

<Table 3.11> Description of Domain class ‘AutoOrderRule’

12) OrderHistory

class Description : A class containing list of the order history which has been already ordered.

Attribute	Description
-	-
Association	Description
OrderHistory(1) - SIS(1)	OrderHistory is recored by SIS.
OrderHistory(1) - Order(0…*)	OrderHistory includes Orders.

<Table 3.12>Description of Domain class ‘OderHistory’

13) Order

class Description : information of an order which has been already ordered.

Attribute	Description
orderID	ID of an order.
orderDate	date the order is sent.
orderTime	time the order is sent.
orderedIngredient	ingredients which has been ordered.
supplierSystemAdapter	adapter used to interact with supplier system.
Association	Description
Order(1) - OrderedIngredient(*)	Order is composed of OrderedIngredient
Order(0…*) - OrderHistory(1)	OrderHistory includes Orders.

<Table 3.13> Description of Domain class ‘Order’

14) OrderedIngredient

class Description : ordered ingredients Information

Attribute	Description
ingredientName	발주한 재료 이름 name of ordered ingredient
quantity	발주한 재료의 양 Amount of ordered ingredient
Association	Description
OrderedIngredient(*) - Order(1)	Order is composed of OrderedIngredient

<Table 3.14> Description of Domain class ‘OrderIngredient’

15) SalesLineItem

class Description : Sales amount of each menu

Attribute	Description
quantity	메뉴의 판매량 quantity of menu which is sold.
Association	Description
SalesLineItem(1) - IngredientItem	IngredientItem is counted by SalesLineItem

SalesLineItem(1…*) - SaleInformation(1)	SaleInformation contains SalesLineitem

<Table 3.15>Description of Domain class ‘SalesLineItem’

16) SaleInformation

class Description : P.O.S 에서 판매한 전체 수량 P.O.S 에서 판매한 전체 수량

Attribute	Description
total	전체적인 소모량
Association	Description
SaleInformation(1) - SalesLineItem(1…*)	SaleInformation contains SalesLineitem
SaleInformation(1…*) - Sale(1)	SaleInformations are informed by Sale.

<Table 3.16>Description of Domain class ‘SaleInformation’

17) Sale

class Description : sales detail in on sale.

Attribute	Description
-	-
Association	Description
Sale(1) - SaleInformation(1…*)	SaleInformations are informed by Sale.

<Table 3.17> Description of Domain class ‘Sale’

18) Store

class Description : Cafe Inventory Management System을 사용하는 매장

Attribute	Description
storeName	매장 이름
storeID	매장 ID
address	매장 주소
phoneNumber	매장과 연락이 가능한 전화번호
Association	Description

Store(0…*) - SIS (1)	Store is managed by SIS

<Table 3.16>Description of Domain class ‘SaleInformation’

19) Administrator

class Description : sales detail in on sale.

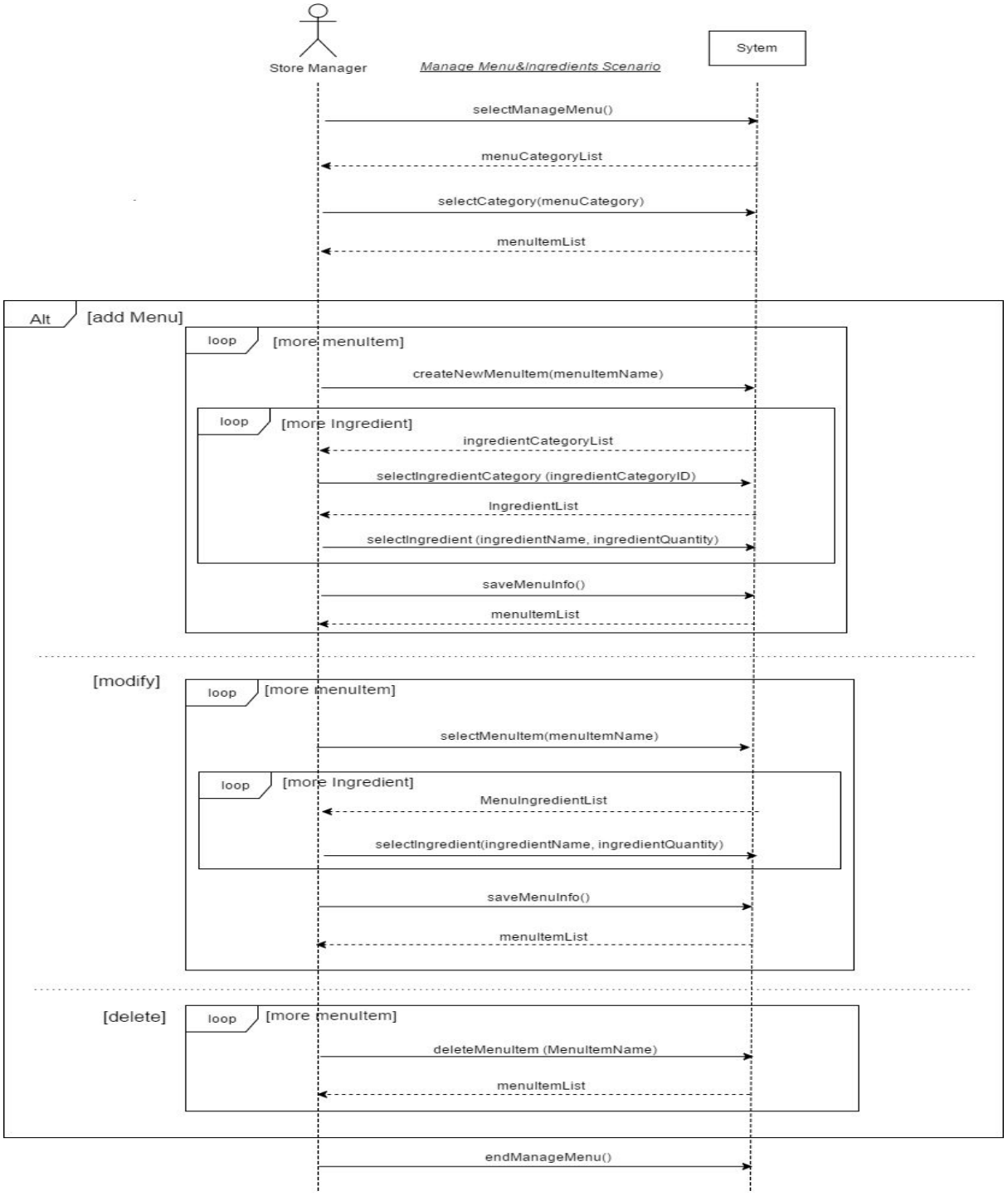
Attribute	Description
administratorID	관리자ID
Association	Description
Administrator(1) - SIS(1)	Administrator uses SIS

<Table 3.17> Description of Domain class ‘Sale’

4. System Sequence Diagram

4.1. Manage menu & ingredients

4.1.1. System Sequence Diagram



<Figure 4.1> Manage Menu & Inventory System Sequence Diagram

System Operation 1 : selectManageMenu()

- The manager selects the menu management.

System Operation 2 : createNewMenuItem(menuItemName)

- The manager adds a new menu item.
- The manager enters the name of the menu item.

System Operation 3 : selectIngredientCategoey(category)

- The manager selects the category of ingredient to manage when creating the menu item.

System Operation 4 : selectIngredient(ingredientName , ingredientQuantity)

- The manager selects the ingredients to manage within the category.
- The manager selects the quantity of the material.

System Operation 5 : saveMenuInfo()

- The manager selects the quantity of the material.
-

System Operation 6 : endNewMenuItem()

- The manager terminates the menu management.

System Operation 7 : selectMenuItem(menuItemName)

- The manager selects the menu item.

System Operation 8 : deleteMenuItem(menuItemName)

- The manager deletes the menu.

4.1.2. Operation Contracts

Operation	**selectManageMenu()**
Cross references	Manage Menu & Ingredients
Precondition	- Store Manager is identified and authenticated - menuCatalog 인스턴스 mc가 생성되어있다. - mc는 모든 메뉴아이템에 대한 정보를 가지고있다. - instance of menuCatalog, mc was created. - mc had all the information about all menuitems.
Postcondition	-

<Table 4.1> selectManageMenu Operation Contracts

Operation	**createNewMenuItem(menuItemName)**
Cross references	Manage Menu & Ingredients
Precondition	- 메뉴등록이 진행중이다. - menu registration is in progress. - 모든 menuItem에 대해 menuItem 인스턴스 mi가 생성되어있다. - 모든 mi는 menuCatalog 인스턴스와 연관되어 있다.
Postcondition	- newMenuItem 인스턴스 nmi가 생성되었다. - nmi의 모든 속성이 초기화되었다. - 새로운 menuItemID 값이 생성되고 nmi.menuItemID 값이 새로 생성된 menuItemID 값으로 변경되었다. - nmi.menuName이 menuItemName으로 변경되었다.

<Table 4.2> createNewMenuItem Operation Contracts

Operation	**selectIngredientCategory(menuItemID)**
Cross references	Manage Menu & Ingredients
Precondition	- 메뉴등록이 진행중이다. - 모든 ingredient의 인스턴스 i가 생성되어있다. - 모든 ingredient의 인스턴스 i는 각 ingredient의 정보를 가지고 있다. - menu registration is in progress.
Postcondition	- ???(i.categoryID가 categoryId와 일치하는 것들을 보여줌?)

<Table 4.4> selectIngredientCategory Operation Contracts

Operation	**selectIngredient(ingredientName, ingredientQuantity)**
Cross references	Manage Menu & Ingredients
Precondition	- 메뉴등록이 진행중이다. - menu registration is in progress.

Postcondition	- menuIngredientItem 인스턴스 mii가 생성되었다. - mii의 속성값이 초기화 되었다. - nmi와 mii가 연관되었다. - mii.name이 ingredientName으로 변경되었다. - mii.quantity 가 ingredientQuantity로 변경되었다. - instance of menuIngredientItem, mi is created. - attribute value of mii was initialized. - nmi was associated to mii. - value of mii.name was changed to ingredientName. - value of mii.quantity was changed to ingredientQuantity.

<Table 4.5> selectIngredient Operation Contracts

Operation	**saveMenuInfo()**
Cross references	Manage Menu & Ingredients
Precondition	- 메뉴등록이 진행중이다. - menu registration is in progress.
Postcondition	- nmi became mi.

<Table 4.7> saveMenuInfo Operation Contracts

Operation	**selectMenuItem(menuItemID)**
Cross references	Manage Menu & Ingredients
Precondition	- 모든 menuItem의 인스턴스 mi가 생성되어있다. - instances of all menuItem had been created.
Postcondition	

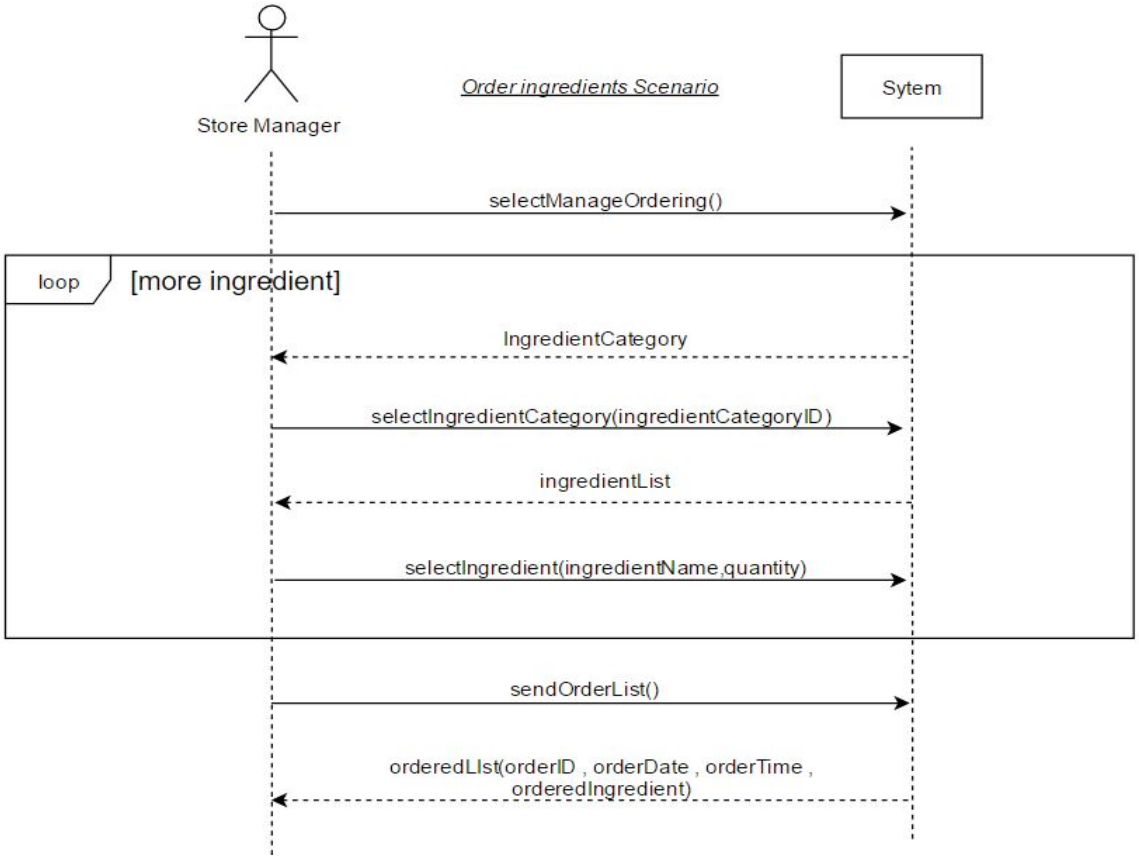
<Table 4.8> selectMenuItem Operation Contracts

Operation	**deleteMenuItem(menuItemID)**
Cross references	Manage Menu & Ingredients
Precondition	- 메뉴삭제가 진행중이다.
Postcondition	- menuItemID값을 가지는 인스턴스 mi가 삭제되었다.

<Table 4.9> deleteMenuItem Operation Contracts

4.2. Order Ingredient

4.2.1. System Sequence Diagram



<Figure 4.3> Order Ingredient System Sequence Diagram

System Operation 1 : SelectManageOrdering()

- The manager selects the order management.

System Operation 2 : selectIngredientCategory(ingredientCategoryID)

- 매니저는 시스템에게 발주하고자 하는 재료가 속한 카테고리를 선택한다.

System Operation 3 :selectIngredient(ingredientName, quantity)

- 매니저는 발주하고자 하는 재료의 이름과 수량을 선택한다.

System Operation 4 :sendOrderList()

- 매니저가 시스템에게 발주를 요청한다.

4.2.2. Operation Contracts

Operation	**selectManageOrdering()**
Cross references	Order Ingredients
Precondition	- Store Manager is identified and authenticated - order 인스턴스 o가 생성되어있다.
Postcondition	- 재료발주가 진행된다.

<Table 4.18> selctMangeOrdering Operation Contracts

Operation	**selectIngredientCategory(ingredientCategoryID)**
Cross references	Order Ingredients
Precondition	- 재료발주가 진행중이다.
Postcondition	- orderIngredient 인스턴스 oi가 생성되었다. - An orderIngredient instance oi was created. - oi의 속성들이 초기화 되었다. - The attributes of oi have been initialized.

<Table 4.19> registerOrderingTime Operation Contracts

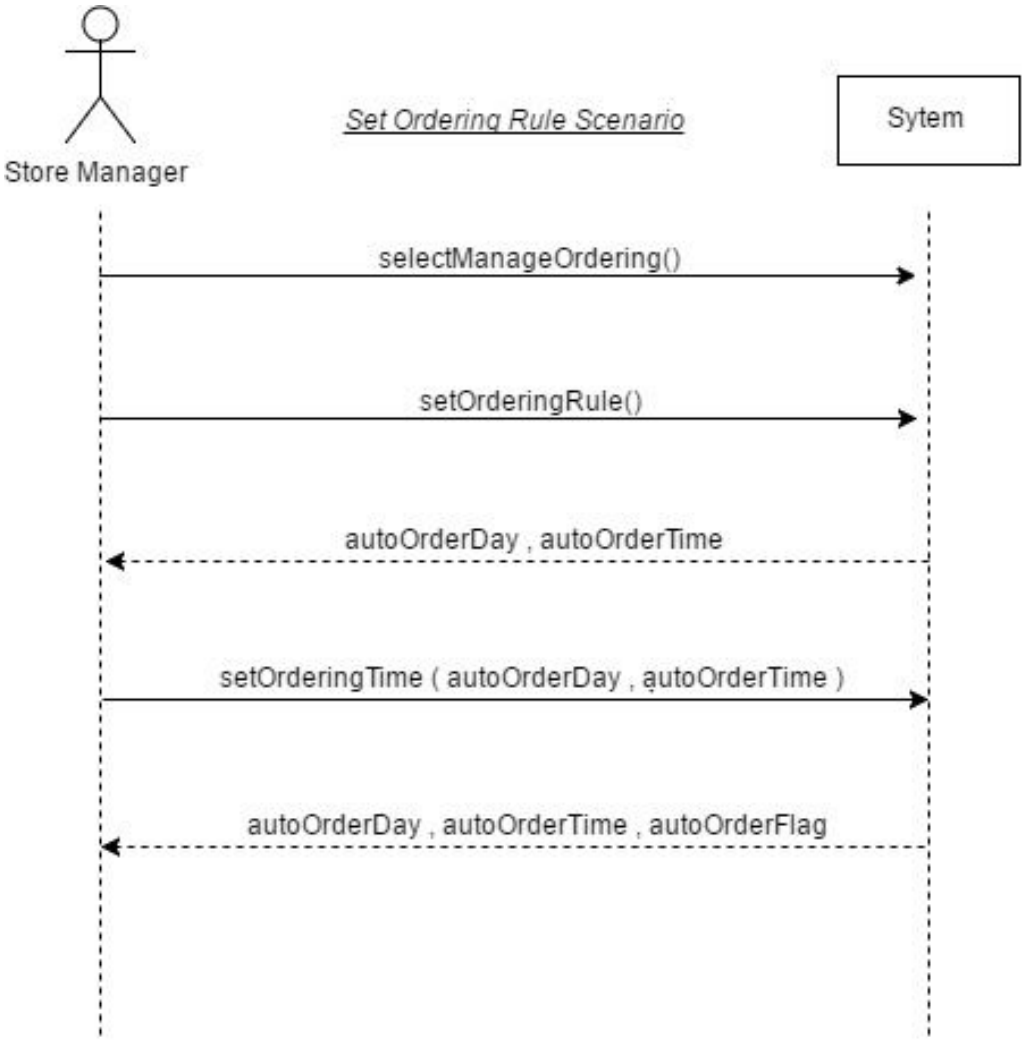
Operation	**selectIngredient(ingredientName, quantity)**
Cross references	Order Ingredients
Precondition	- 재료발주가 진행중이다. - Ordering time registration is in progress.
Postcondition	- oi.ingredientName이 ingredientName가 되었다. - oi.ingredientName has become ingredientName.. - oi.quantity가 quantity이 되었다. - oi.quantity has become quantity.

<Table 4.20> selectOrderingTime Operation Contracts

Operation	**sendOrderList()**
Cross references	Order Ingredients
Precondition	- 재료발주가 진행중이다. - Ordering time registration is in progress.
Postcondition	

4.3. Set Order Info

4.3.1. System Sequence Diagram



<Figure 4.3> Set Order Info System Sequence Diagram

System Operation 1 : SelectManageOrdering()

- The manager selects the order management.

System Operation 2 : setOrderingRule()

- The manager requests the system to register the ordering day and time.

System Operation 3 : setOrderingTime(orderingDay , orderingTime)

- The manager selects the day and time of the order.

4.3.2. Operation Contracts

Operation	**selectSetOrderInfo()**
Cross references	Set OrderRule
Precondition	
Postcondition	- orderInfo 설정이 시작되었다.

Operation	**setOrderingRule()**
Cross references	Set OrderRule
Precondition	- orderInfo 설정이 진행중이다.
Postcondition	- orderInfo 인스턴스 oi가 생성되었다. - oi의 속성값이 초기화 되었다.

Operation	**setOrderingTime(autoOrderDay, autoOrderTime)**
Cross references	Set OrderRule
Precondition	- orderInfo 설정이 진행중이다.
Postcondition	- oi.autoOrderDay 값이 autoOrderDay로 변경되었다. - oi.autoOrderTime 값이 autoOrderTime으로 변경되었다.

4.4. Show Order History

4.4.1. System Sequence Diagram

<Figure 4.5> Show Order History System Sequence Diagram

System Operation 1 : selectManageOrdering()
- 매니저가 발주관리를 선택한다.

System Operation 2 : selectOrderHistory()
- 매니저가 발주내역관리를 선택한다.

System Operation 3 : selectOrder(orderID)
- 매니저가 확인하고자하는 발주내역을 선택한다.

4.4.2. Operation Contracts

Operation	**selectManageOrdering()**
Cross references	Show Order History
Precondition	- The store manager is identified and authenticated
Postcondition	-

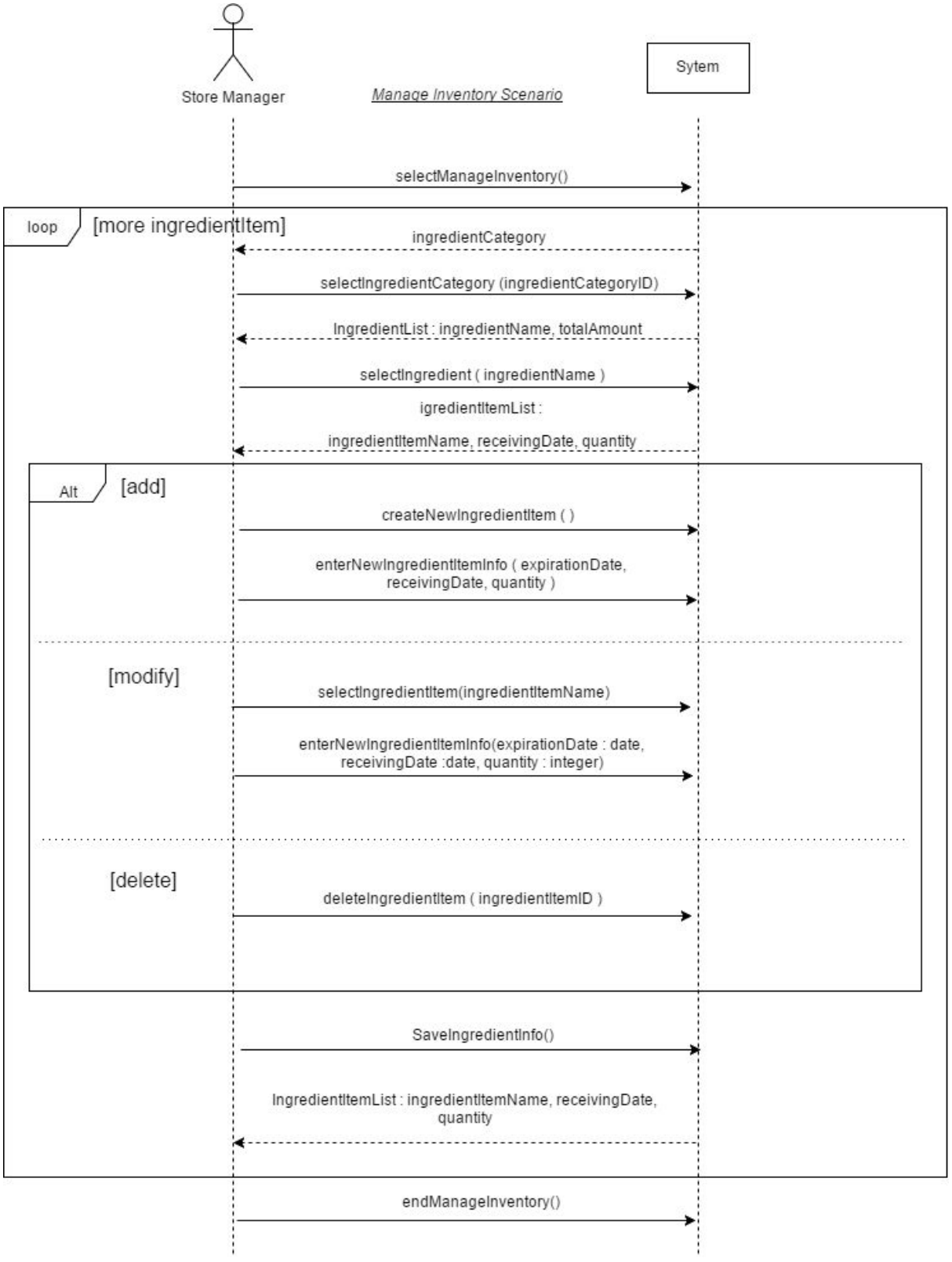
Operation	**selectOrdeHistory()**
Cross references	Show Order History
Precondition	- order 인스턴스 o가 수동 및 자동발주를 포함한 모든 발주내역에 대한 정보를 가지고 있다.
Postcondition	-

Operation	**selectOrder(orderID)**
Cross references	Show Order History
Precondition	- order 인스턴스 o가 수동 및 자동발주를 포함한 모든 발주내역에 대한 정보를 가지고 있다.
Postcondition	-

<Table 4.18> Show Order History Operation Contracts

4.5. Manage inventory

4.5.1. System Sequence Diagram



<Figure 4.2> Manage Inventory System Sequence Diagram

System Operation 1 : selectManageInventory()

- Manager selects inventory management.

System Operation 2 : selectIngredientCategory (categoryName)

- The manager selects the category to which the ingredient is to be managed.

System Operation 3 : selectIngredient(ingredientName)

- The manager selects the ingredient to manage.

System Operation 4 : createNewIngredientItem()

- The manager requests adding ingredient items to the selected ingredient.

System Operation 5 : enterNewIngredientItemInfo(expirationDate, receivingDate, quantity)

- The manager enters the expiration date, warehousing date, and quantity of the selected ingredient.

System Operation 6 : endManageInventory()

- The manager terminates the ingredient management.

System Operation 7 : selectIngredientItem(ingredientItemName)

- The manager selects the ingredient item to modify.

System Operation 8 : deleteIngredientItem(ingredientItemName)

- The manager requests deletion of the selected materials.

4.5.2. Operation Contracts

Operation	**selectManageInventory()**
Cross references	Manage Inventory
Precondition	- Store Manager is identified and authenticated - ingredientCatalog 인스턴스 ic가 생성되어있다. - ic는 모든 ingredeint에 대한 정보를 가지고 있다.
Postcondition	-

<Table 4.10> selectManageInventory Operation Contracts

Operation	**selectIngredientCategory (ingredientCategoryID)**
Cross references	Manage Ingredient
Precondition	- 재고 관리가 진행중이다.
Postcondition	-

<Table 4.11> selectIngredientcategory Operation Contracts

Operation	**selectIngredient (ingredientName)**
Cross references	Manage Ingredient
Precondition	- 재고 관리가 진행중이다. - 모든 ingredient에 대해 ingredient 인스턴스 i가 생성되어있다. - ingredient 인스턴스는 해당재료에 대한 모든 재고 정보를 갖고있다. - 모든 ingredient 인스턴스 i는 ingredientCatalog 인스턴스와 연관되어있다.
Postondition	-

<Table 4.12> selectIngredient Operation Contracts

Operation	**createNewIngredientItem()**
Cross references	Manage Ingredient
Precondition	- 재고 관리가 진행중이다. - 모든 ingredientItem에 대해 ingredientItem 인스턴스 ii가 생성되어있다. - 모든 ii는 해당 ingredient 인스턴스와 연관되어있다.
Postondition	- NewIngredientItem 인스턴스 nii가 생성되었다. - nii의 속성값이 초기화되었다. - 새로운 ingredientItemID 값이 생성되고 nii.ingredientItemID 값이 새로 생성된 ingredientItemID 값으로 변경되었다.

<Table 4.13> creatNewIngredientItem Operation Contracts

Operation	**enterNewIngredientItemInfo (expirationDate , receivingDate , quantity)**
Cross references	Manage Ingredient
Precondition	- NewIngredientItem 인스턴스 nii가 생성되었다. - nii의 속성값이 초기화되었다.
Postcondition	- nii.expirationDate 와 nii.receivingDate와 nii.quantity가 expirationData, receiveingDate, quantity로 되었다. - value of nii.expirationDate changed to expirationDate - value of nii.receivingDate changed to receivingDate - value of nii.quantity changed to quantity

<Table 4.14> enterNewIngredientItemInfo Operation Contracts

Operation	**saveNewIngredientItem()**
Cross references	Manage Ingredient
Precondition	- 재고 관리가 진행중이다. - IngrdientItem인스턴스 ii 가 각각의 기존 ingredientItem에 대한 정보를 가지고 있다.
Posondition	- NewIngredientItem 인스턴스 nii가 ii가 되었다.

<Table 4.15> saveNewIngredientItem Operation Contracts

Operation	**selectIngredientItem(ingredientItemID)**
Cross references	Manage Ingredient
Precondition	- 재고 관리가 진행중이다. - 모든 ingredientItem의 인스턴스 ii가 생성되어있다.
Postcondition	-

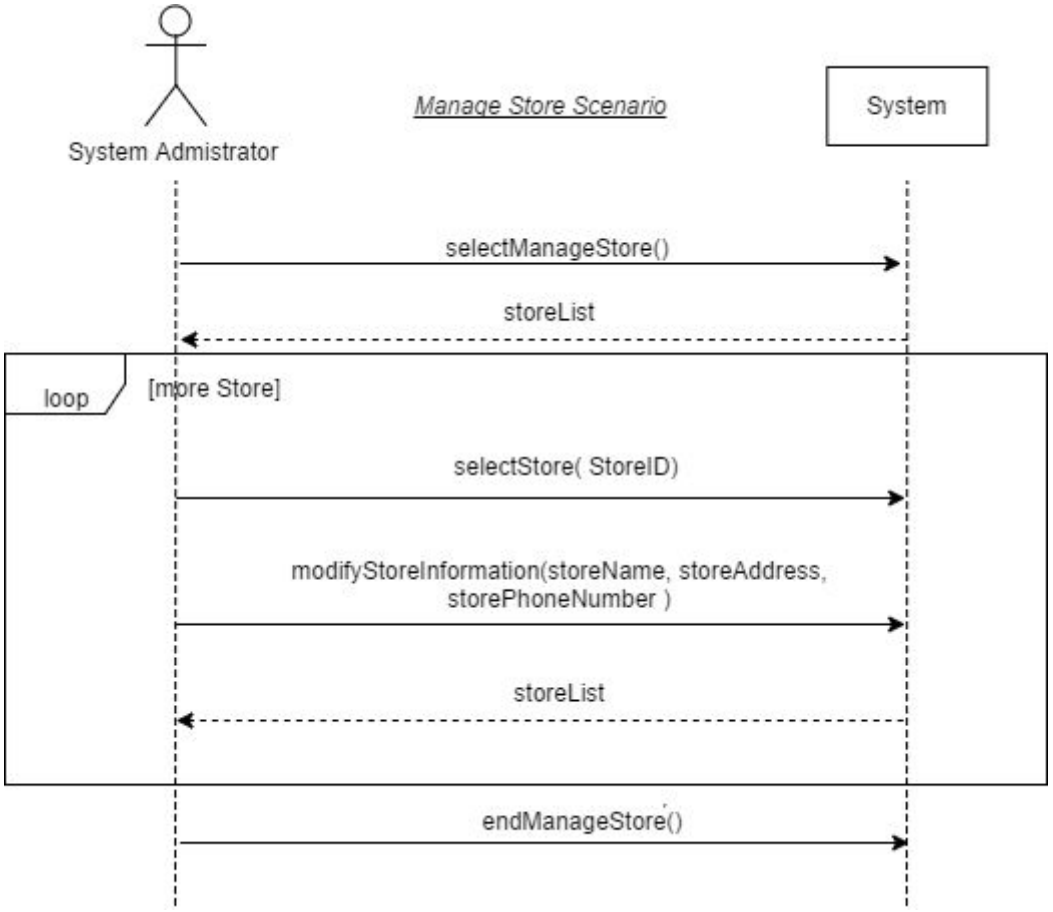
<Table 4.16> selectIngredientItem Operation Contracts

Operation	**deleteIngredientItem(ingredientItemID)**
Cross references	Manage ngredient
Precondition	- 재고 관리가 진행중이다. - IngrdientItem인스턴스 ii 가 각각의 기존 ingredientItem에 대한 정보를 가지고 있다.
Postcondition	- 인스턴스 ii중 ingredientItemID를 속성값으로 갖는 인스턴스 ii가 삭제된다.

<Table 4.17> deletIngredientItem Operation Contracts

4.6. Manage Stores

4.6.1. System Sequence Diagram



<Figure 4.6> Order Ingredient System Sequence Diagram

System Operation 1 : selectManageStore()
- 관리자가 매장관리를 선택한다.

System Operation 2 : selectStore(storeID)
- 관리자가 시스템상의 매장 중 관리할 매장을 선택한다.

System Operation 3 :
modifyStoreInformation(storeName,storeAddress,storePhoneNumber)
- 관리자가 선택된 매장의 정보를 수정한다.

System Operation 4 : endManageStore()
- 관리자가 매장관리를 종료한다.

4.6.2. Operation Contracts

Operation	**selectManageStore()**
Cross references	Manage Stores
Precondition	- Administrator is identified and authenticated
Postcondition	- 매장관리가 진행된다.

<Table 4.18> selectMangeStore Operation Contracts

Operation	**selectStore(storeID)**
Cross references	Manage Stores
Precondition	- 매장관리가 진행중이다.
Postcondition	- 모든 store에 대해 store 인스턴스 s가 생성되어있다. - s인스턴스는 해당 매장에 대한 모든 정보를 갖고있다.

<Table 4.19> selectStore Operation Contracts

Operation	**modifyStoreInformation(storeName,storeAddress,storePhoneNumber)**
Cross references	Manage Stores
Precondition	- 매장관리가 진행중이다.
Postcondition	- s.storeName이 storeName이 되었다. - s.storeAddress가 storeAddress가 되었다. - s.storePhoneNumber가 storePhoneNumber가 되었다.

<Table 4.20> modifyStoreInformation Operation Contracts

Operation	**endManageStore()**
Cross references	Manage Stores
Precondition	- 매장관리가 진행중이다
Postcondition	-

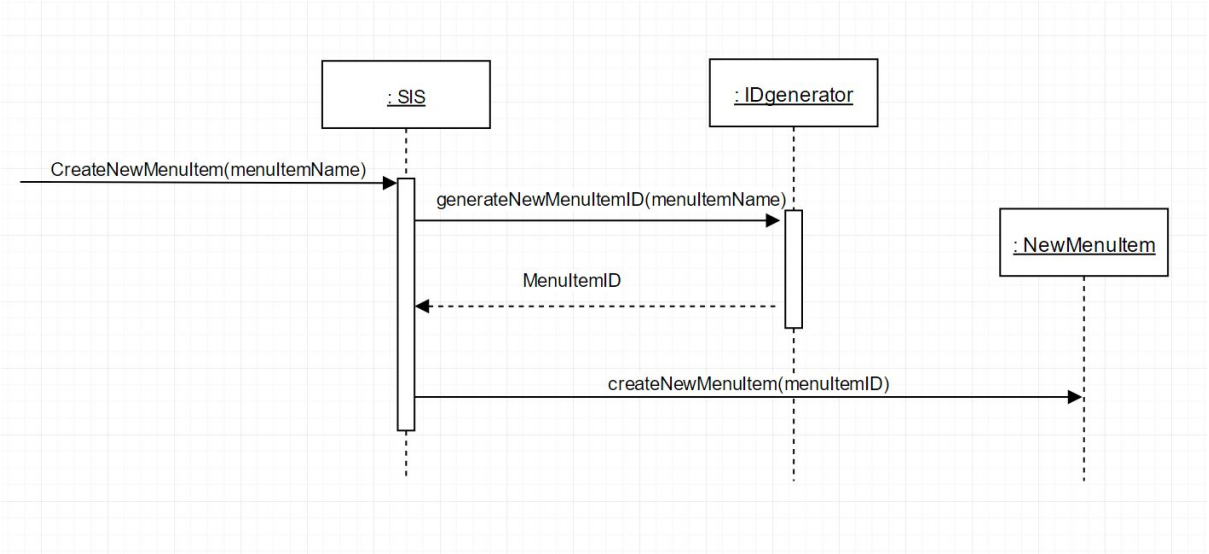
<Table 4.21> endManageStore Operation Contracts

5. Design Model

5.1. Manage menu & ingredients Realization

5.1.1. Design Sequence Diagram

1) createNewMenuItem(menuItemName)

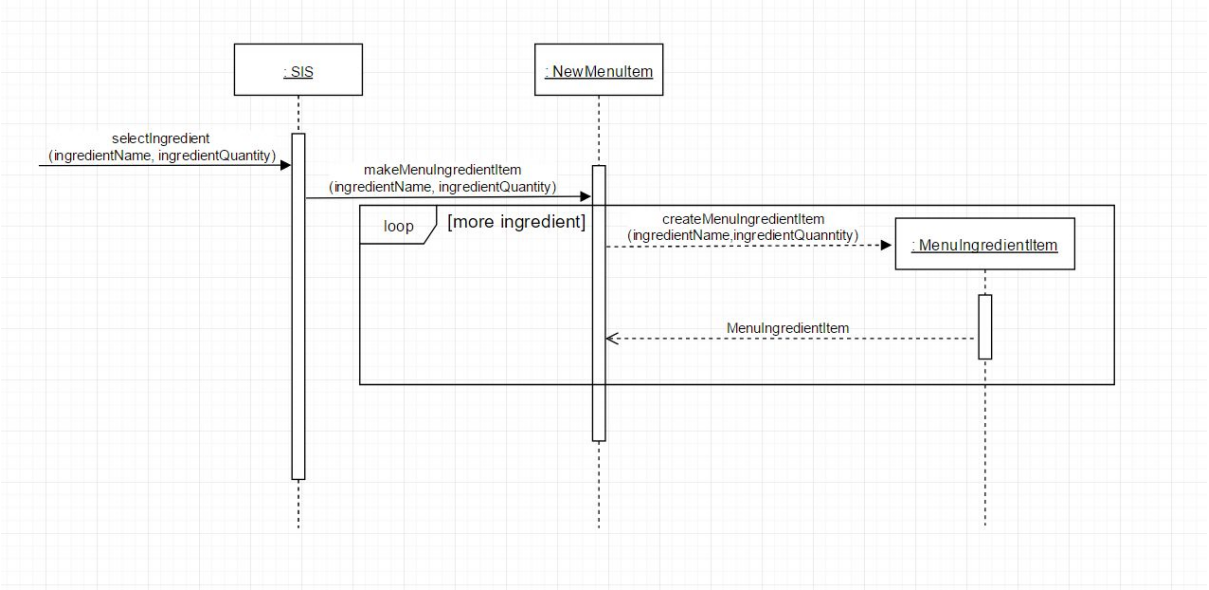


<Figure 5.1> createManageMenu Design Sequence Diagram

GRASP Pattern	Description
Low Coupling	- The menu catalog class has information about all menu items, but SIS directly creates a newMenuItem object to reduce coupling.
Pure Fabrication	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

<Table 5.1> createManageMenu DSD GRASP Pattern

2) selectIngredient(ingredientName, ingredientQuantity)

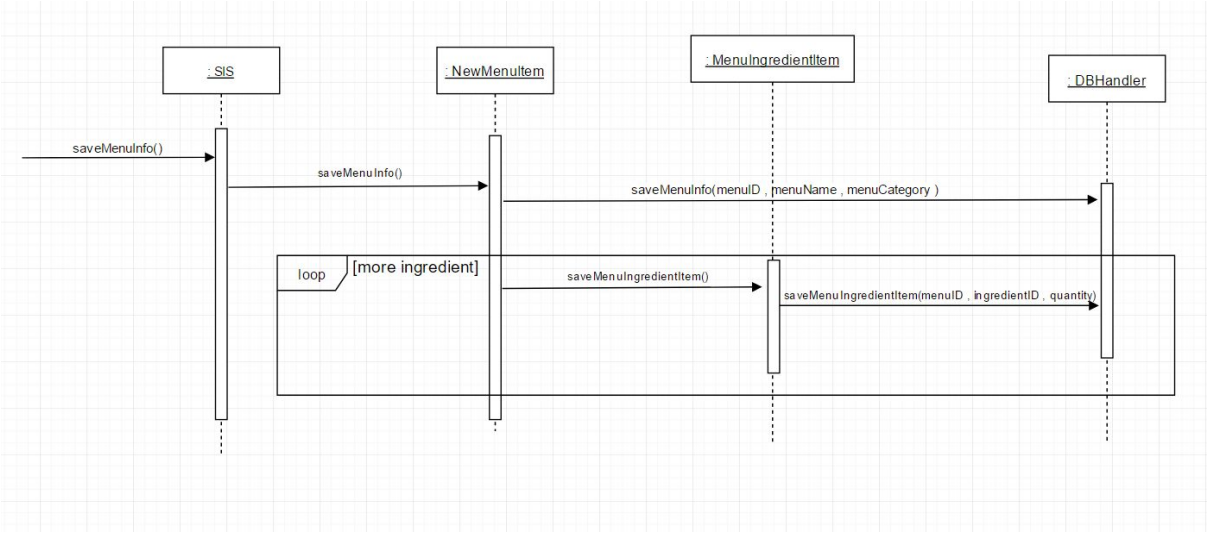


<Figure 5.2> selectIngredient Design Sequence Diagram

GRASP Pattern	Description
Information Expert	-Since the MenuIngredientItem is the information contained in the newly created MenuItem, the NewMenuItem object creates a MenuIngredientItem object.
Controller	-SIS is a facade Controller. SIS is ‘root object’ for overall system.

<Table 5.2> selectIngredient DSD GRASP Pattern

3) saveMenuInfo()

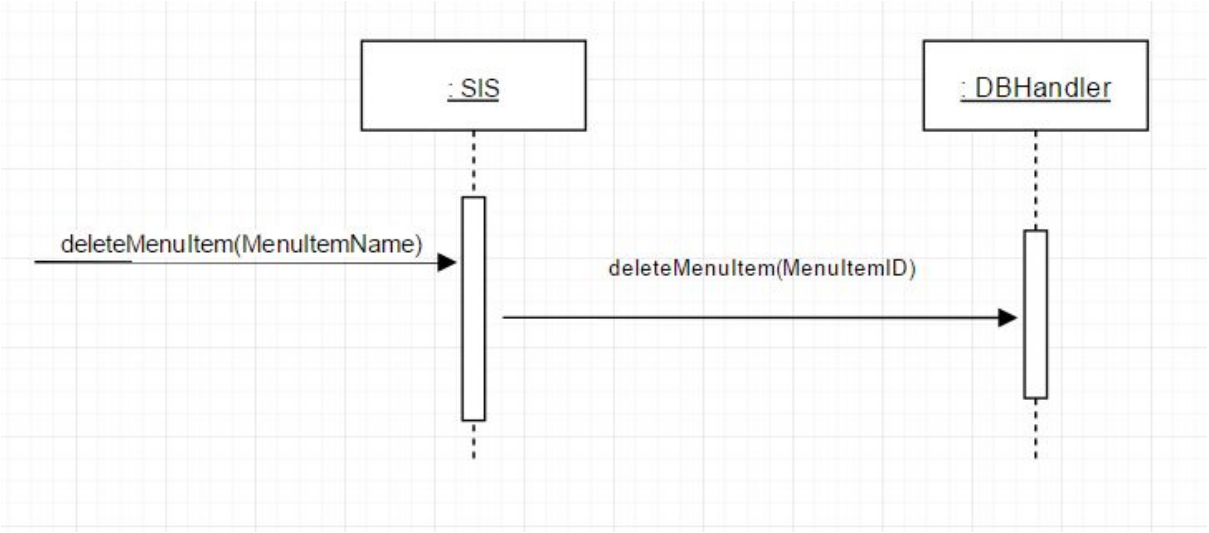


<Figure 5.3> saveMenuInfo Design Sequence Diagram

GRASP Pattern	Description
Low Coupling	-Reduce coupling by showing MenuInfo which means ‘saved’, directly through SIS rather than through a newMenuItem object.
Controller	-SIS is a facade Controller. SIS is ‘root object’ for overall system.
Pure Fabrication	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

<Table 5.3> saveMenuInfo DSD GRASP Pattern

4) deleteMenuItem(MenuItemName)



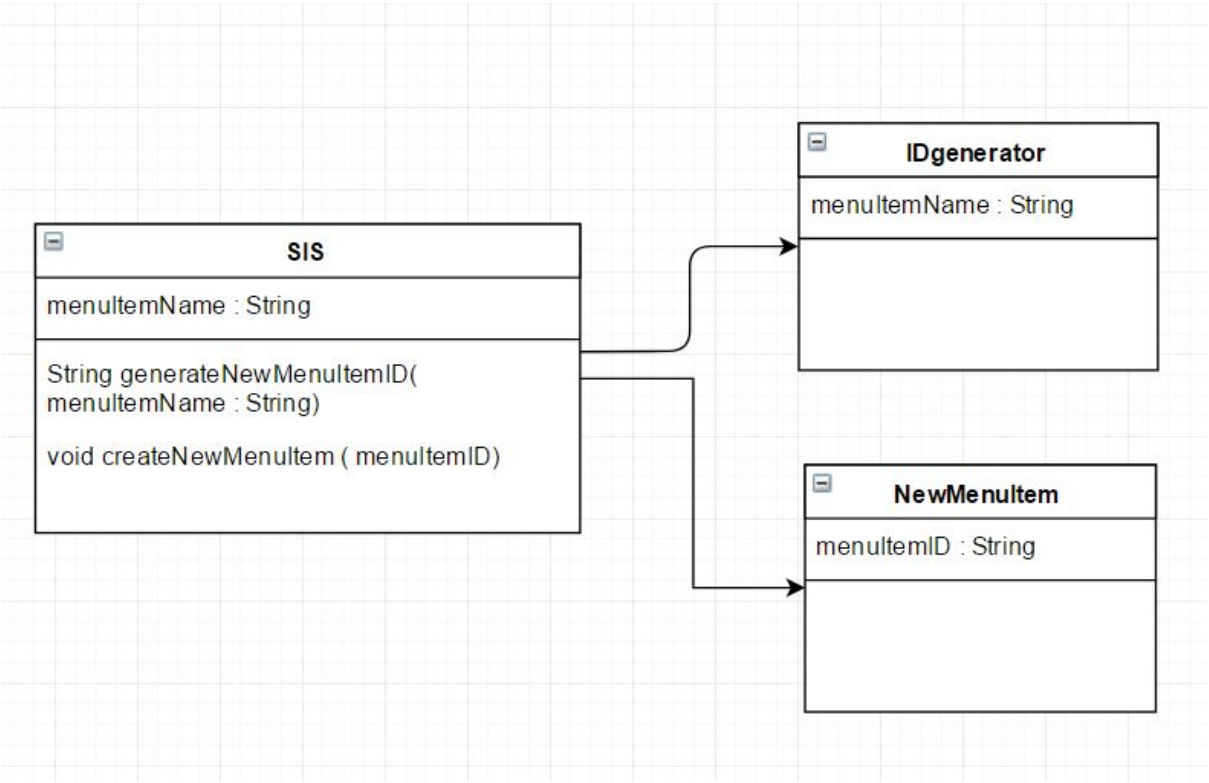
<Figure 5.4> deleteMenuItem Design Sequence Diagram

GRASP Pattern	Description
Low Coupling	-When deleting a MenuItem, rather than associating it with a MenuItem object that matches the MenuItemId you want to delete, you simply pass the MenuItemId you want to delete to the DBHandler and delete it from the DB, so reduce coupling.
Controller	-SIS is a facade Controller. SIS is 'root object' for overall system

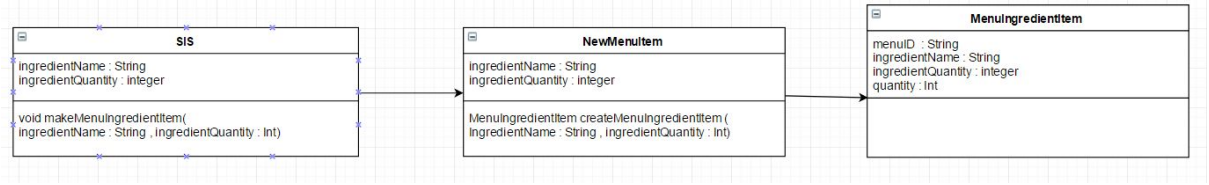
<Table 5.4> deleteMenuItem DSD GRASP Pattern

5.1.2. Design Class Diagram

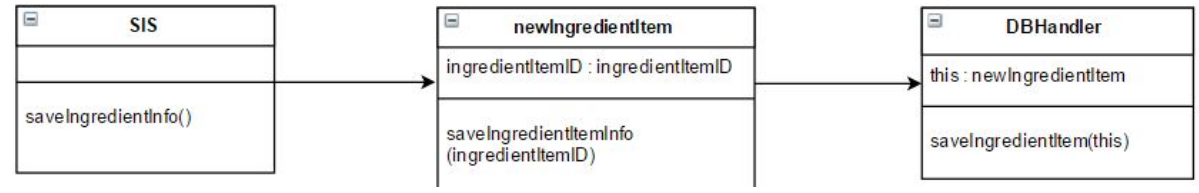
1) createNewMenuItem(menuItemName)



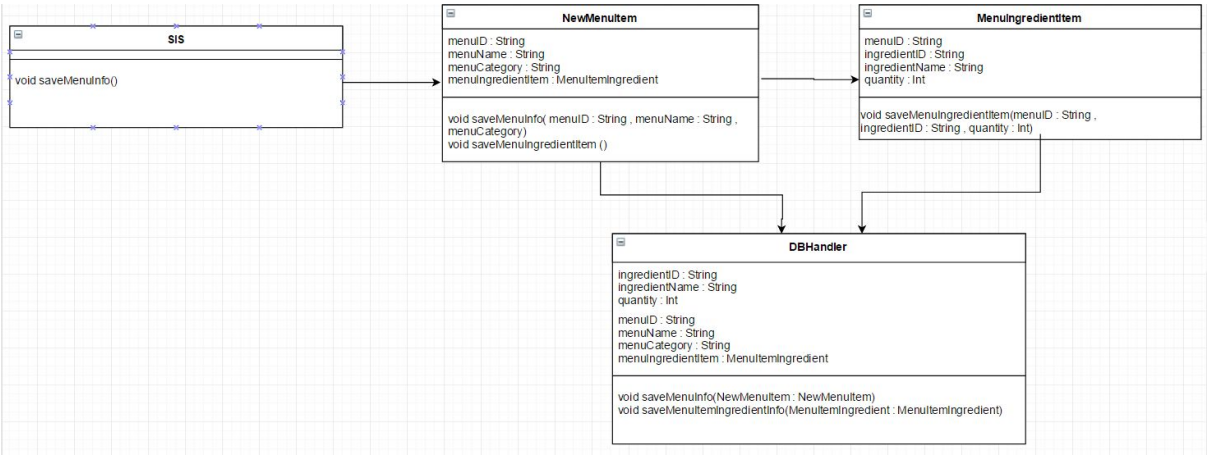
2) selectIngredient(ingredientName, ingredientQuantity)



3) saveMenuInfo()

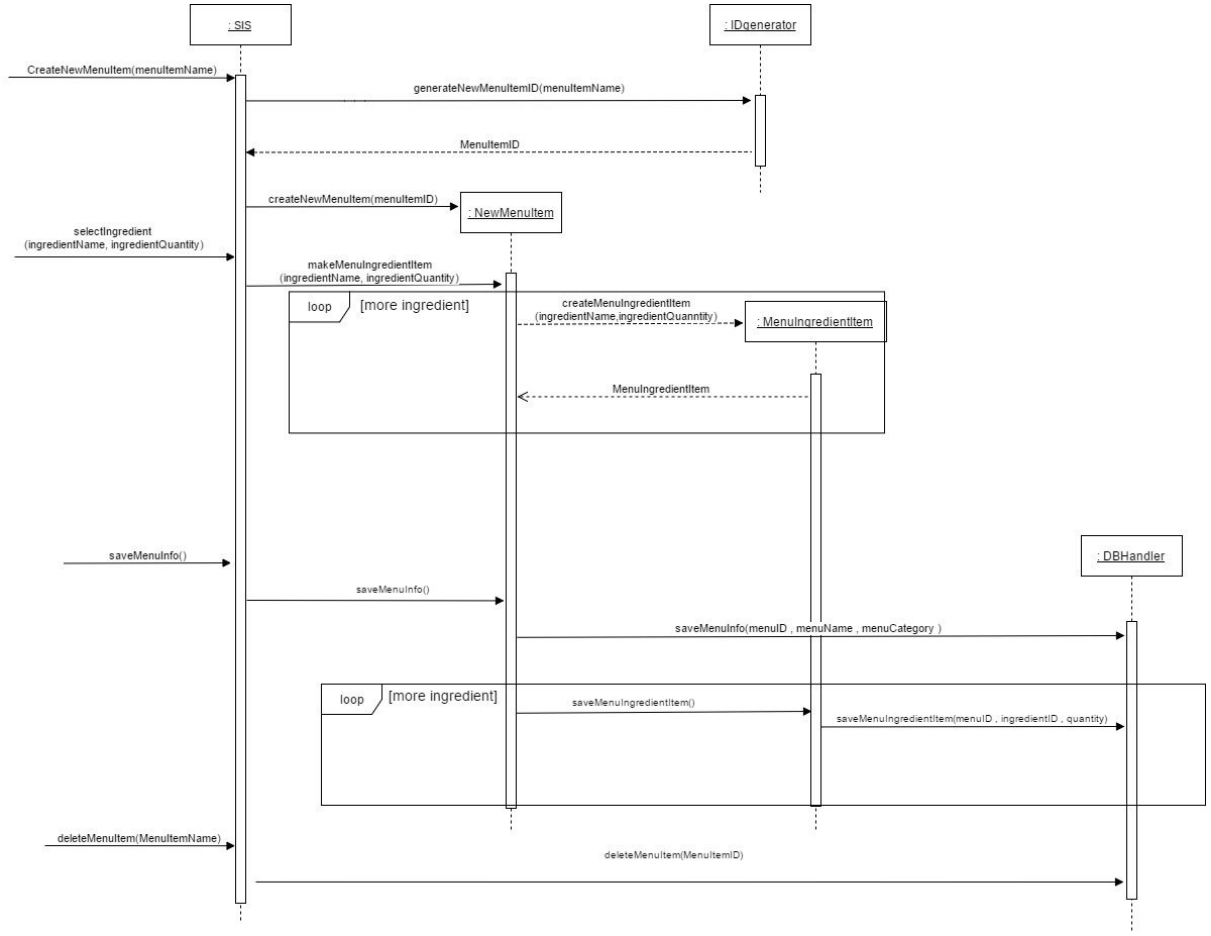


4) deleteMenuItem(MenuItemName)

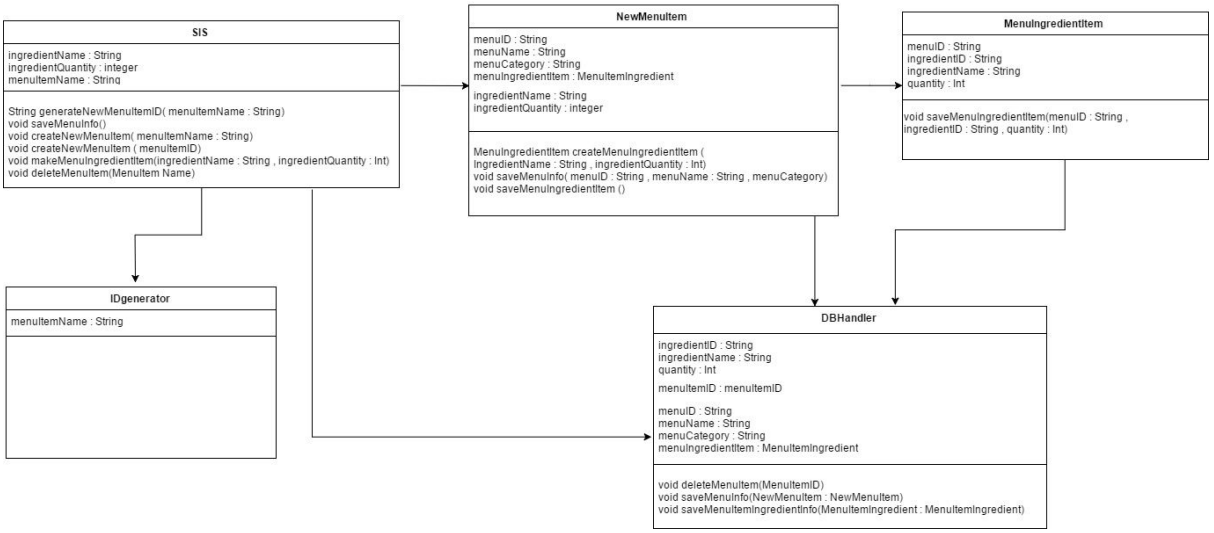


5.1.3. Combined DSD and DCD

1) combined DSD



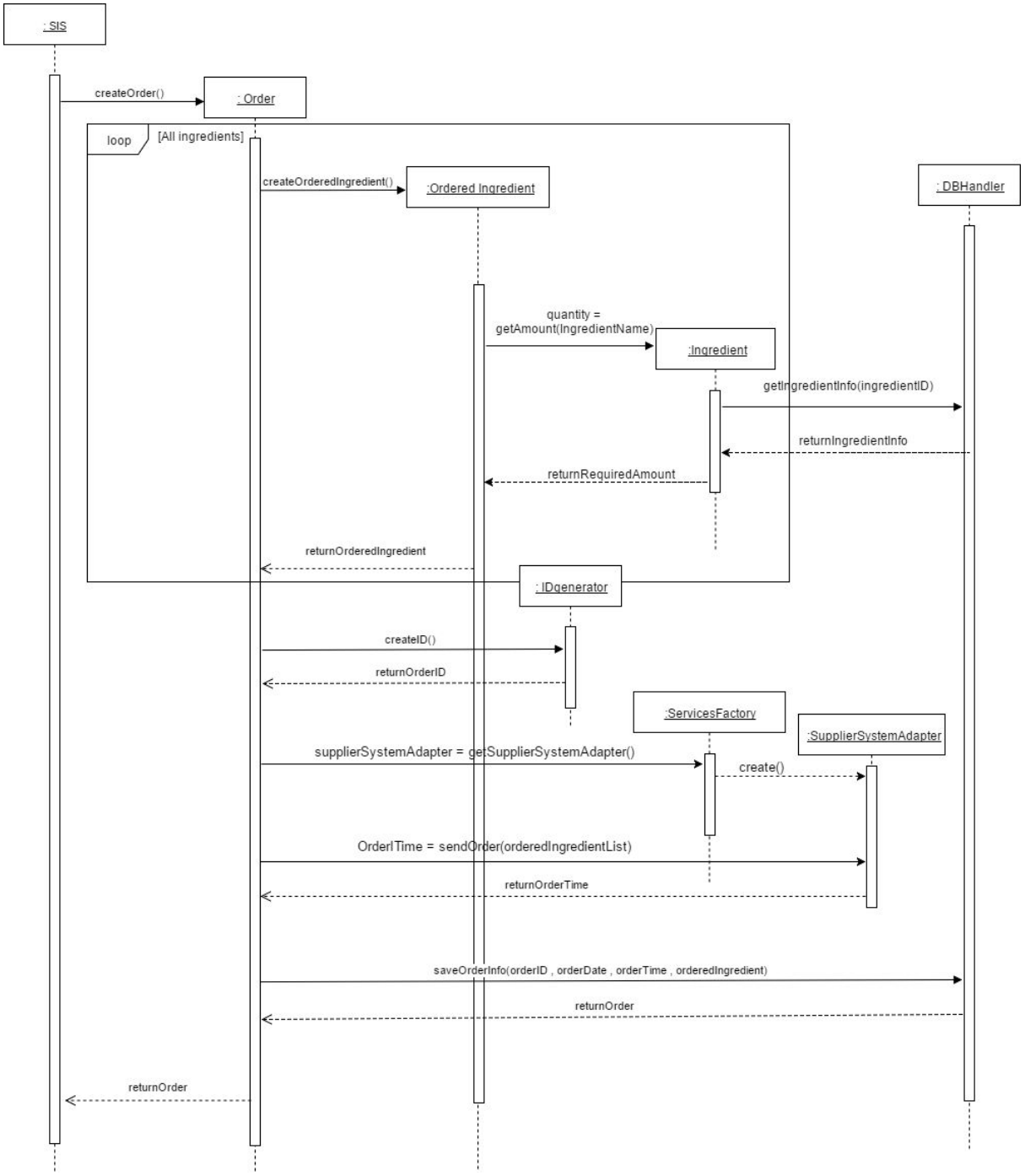
2) combined DCD


SIS
ingredientName : String
ingredientQuantity : integer
menuItemName : String
String generateNewMenuItemID(menuItemName : String)
void saveMenuInfo()
void createNewMenuItem(menuItemName : String)
void createNewMenuItem (menuItemID)
void makeMenuIngredientItem(ingredientName : String , ingredientQuantity : Int)
void deleteMenuItem(MenuItem Name)
NewMenuItem
menuID : String
menuName : String
menuCategory : String
menuIngredientItem : MenuItemIngredient
ingredientName : String
ingredientQuantity : integer
MenuIngredientItem createMenuIngredientItem (
IngredientName : String , ingredientQuantity : Int)
void saveMenuInfo(menuID : String , menuName : String , menuCategory)
void saveMenuIngredientItem ()
MenuIngredientItem
menuID : String
ingredientID : String
ingredientName : String
quantity : Int
void saveMenuIngredientItem(menuID : String ,
ingredientID : String , quantity : Int)
IDgenerator
menuItemName : String
DBHandler
ingredientID : String
ingredientName : String
quantity : Int
menuItemID : menuItemID
menuID : String
menuName : String
menuCategory : String
menuIngredientItem : MenuItemIngredient
void deleteMenuItem(MenuItemID)
void saveMenuInfo(NewMenuItem : NewMenuItem)
void saveMenuItemIngredientInfo(MenuItemIngredient : MenuItemIngredient)

5.2. Order Ingredients Realization

5.2.1. Design Sequence Diagram

1) OrderIngredients



<Figure 5.> OrderIngredients Sequence Diagram

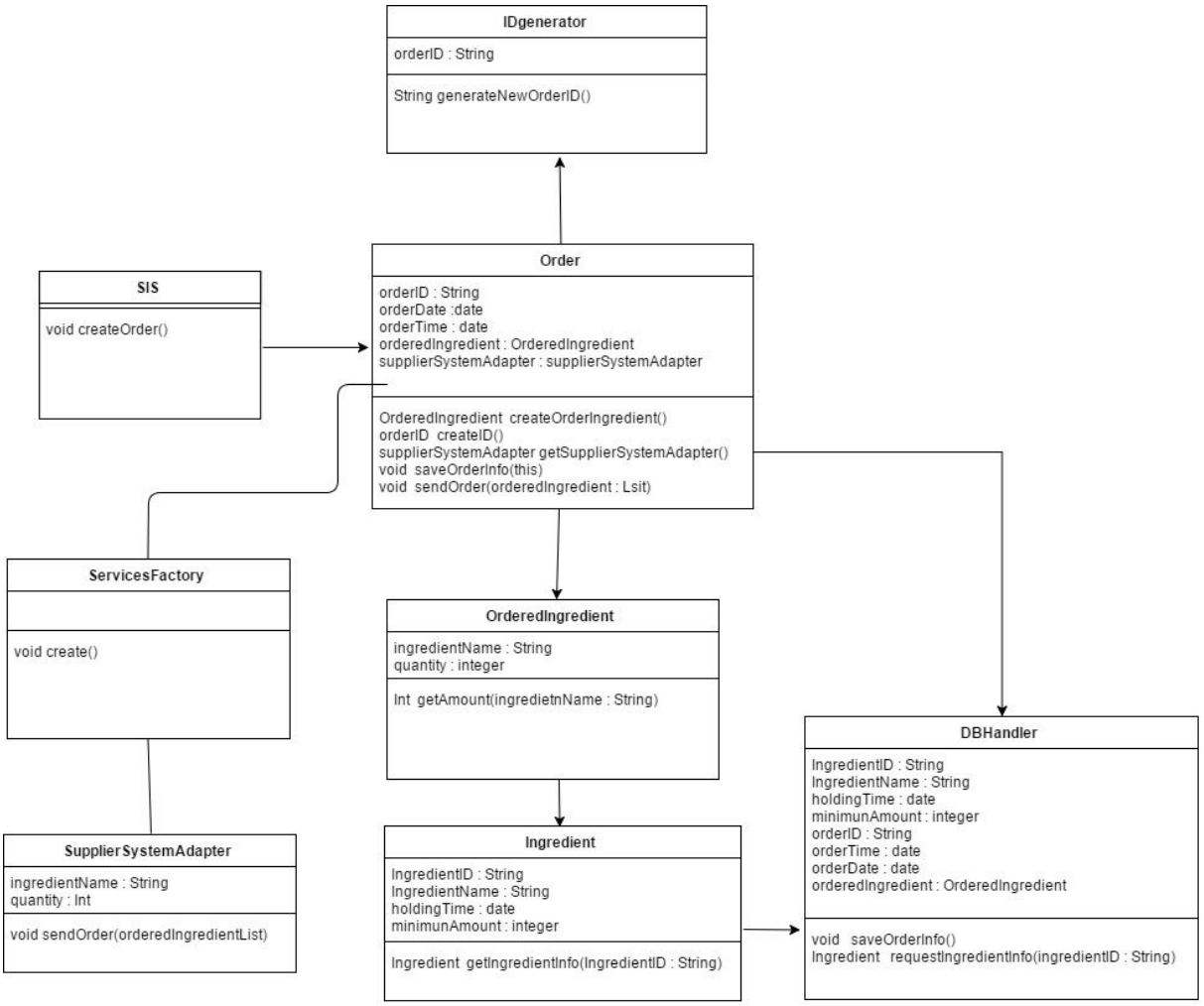
GRASP Pattern	Description
Controller	-SIS is a facade Controller. SIS is ‘root object’ for overall system
Pure Fabrication	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

<Table 5.6> OrderIngredients DSD GRASP Pattern

GOF Pattern	Description
Adapter	- ‘SupplierSystemAdapter’ needs to support several kinds of external supplier system.
Factory	- ‘ServicesFactory’ create a Pure Fabrication object called a Factory that handles the creation.

5.2.2. Design Class Diagram

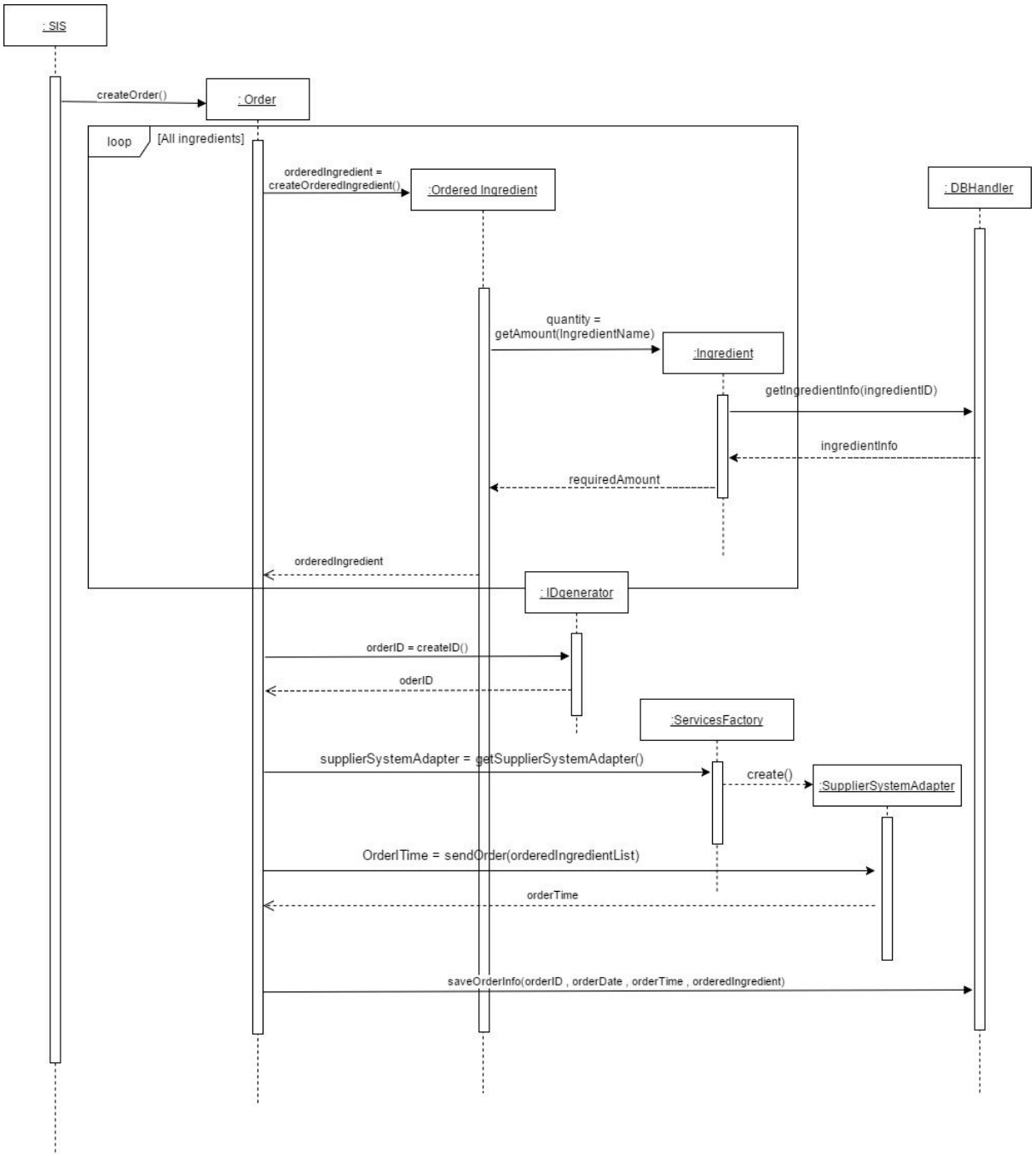
1) OrderIngredients



<Figure 5.> OrderIngredients Design Class Diagram

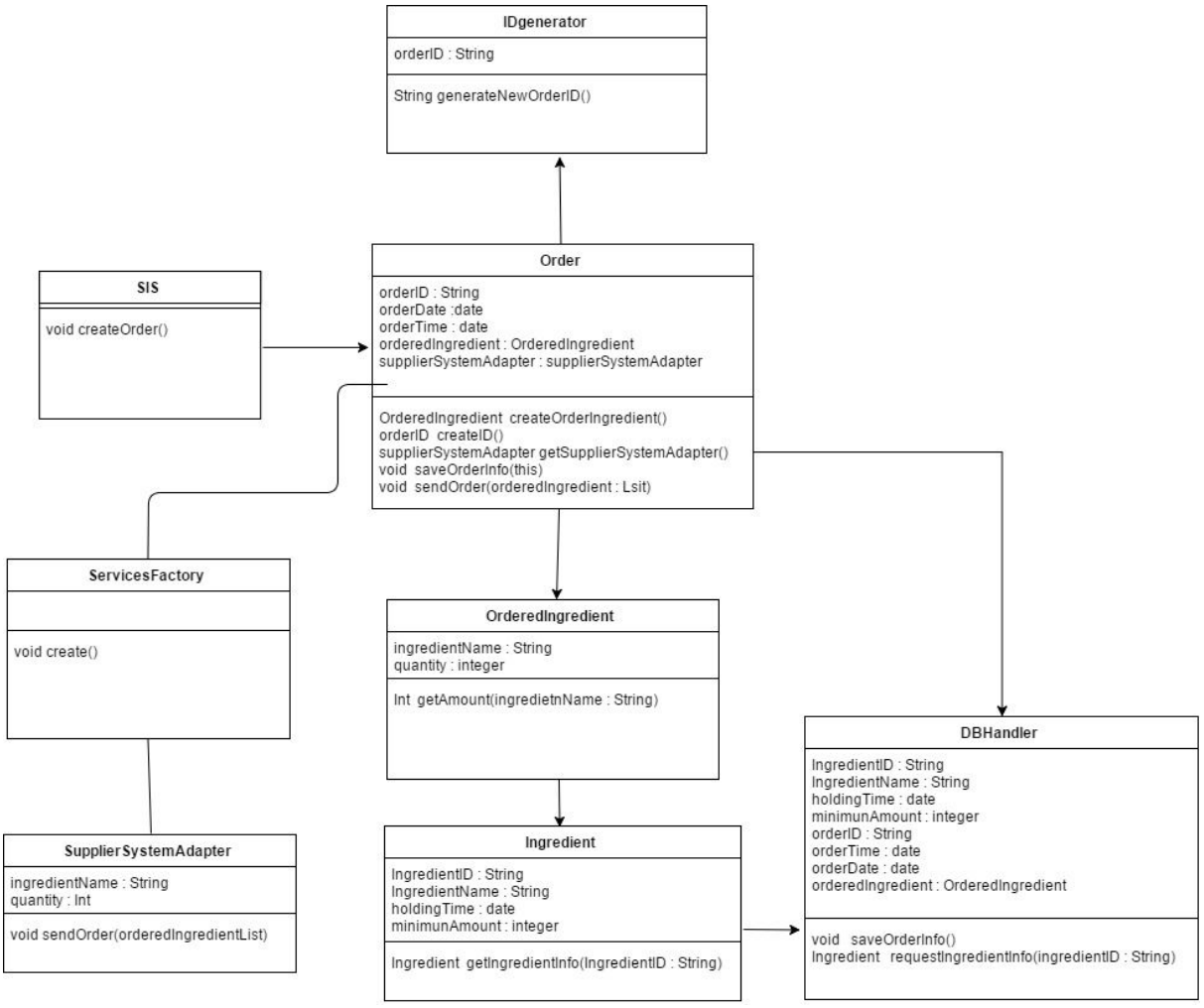
5.2.3. Combined DSD and DCD

1) combined DSD



<Figure 5.> OrderIngredients Combined DSD

2) combined DCD

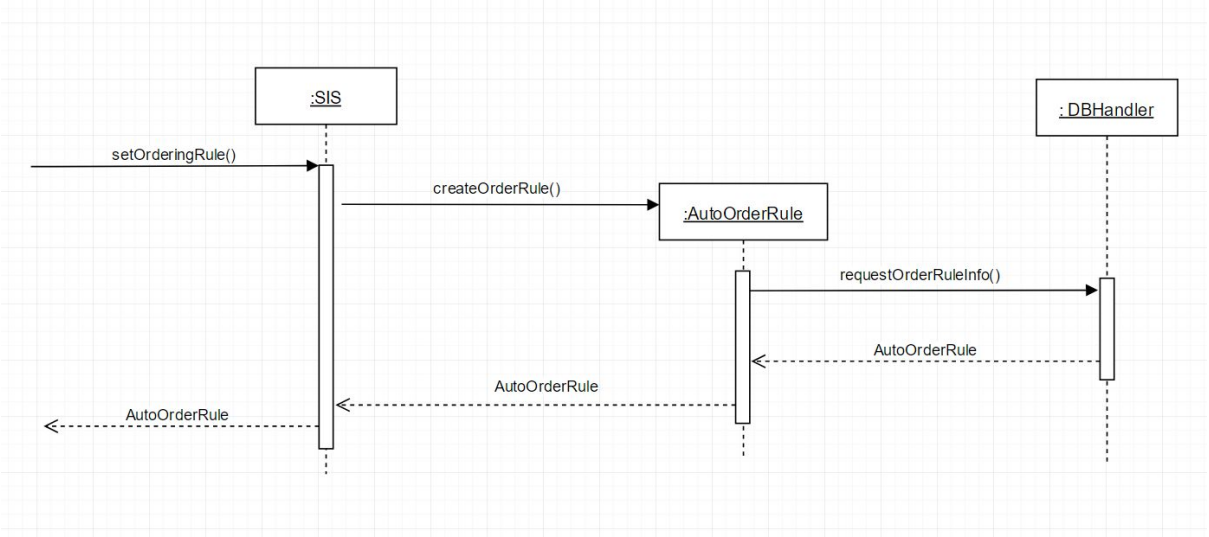


<Figure 5.> OrderIngredients Combined DCD

5.2-1. Set OrderingRule

5.2-1.1. Design Sequence Diagram

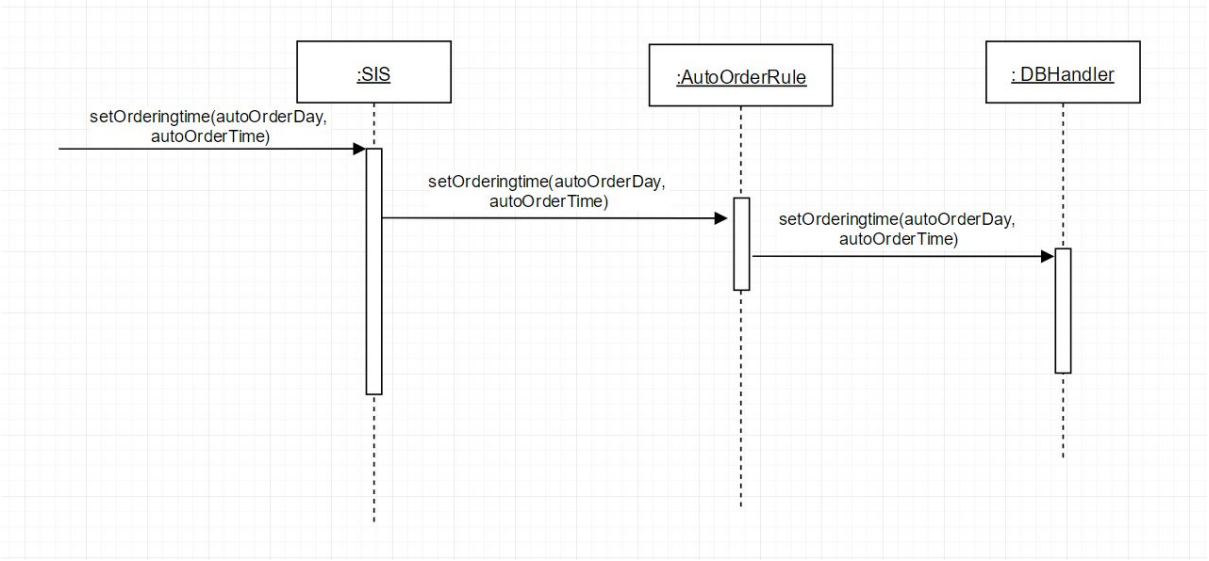
1) setOrderingRule()



GRASP Pattern	Description
Controller	-SIS is a facade Controller. SIS is ‘root object’ for overall system.
Pure Fabrication	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

<Table 5.> setOrderingRule DSD GRASP Pattern

2) setOrderingTime(autoOrderDay, autoOrderTime)

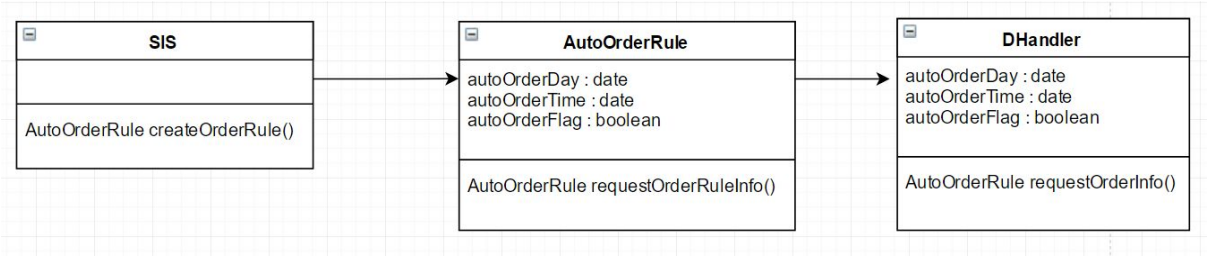


GRASP Pattern	Description
Controller	-SIS is a facade Controller. SIS is 'root object' for overall system
Pure Fabrication	- Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

<Table 5.> setOrderingTime DSD GRASP Pattern

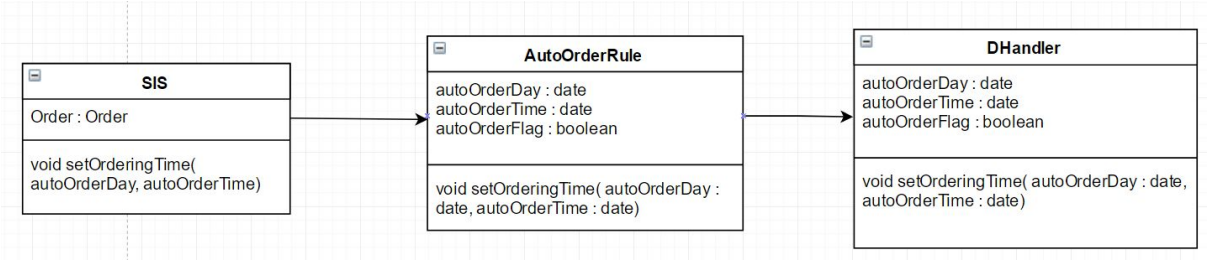
5.2-1.2. Design Class Diagram

1) setOrderingRule()



<Figure 5. > selectOrderRule() Design Sequence Diagram

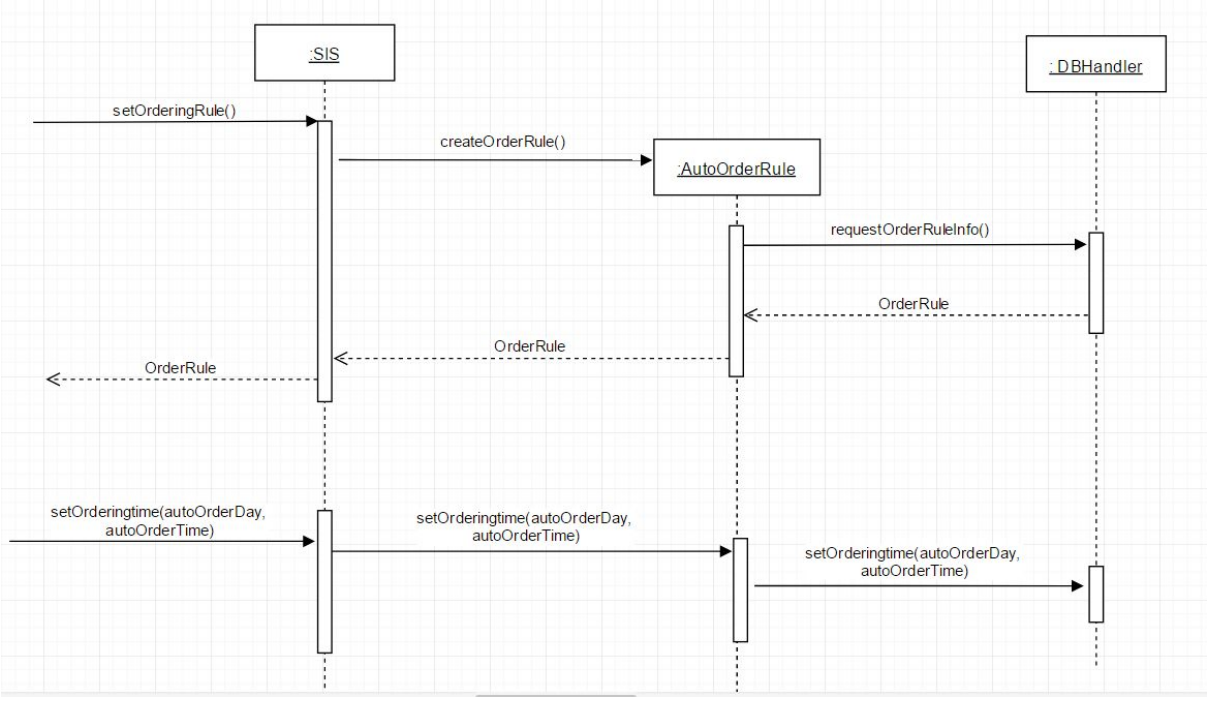
2) setOrderingTime(autoOrderDay, autoOrderTime)



<Figure 5. > selectOrderingTime Design Sequence Diagram

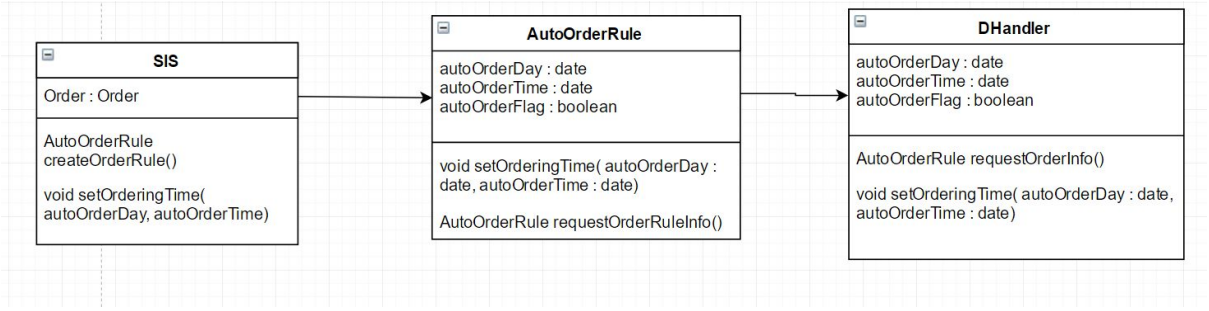
5.2-1.3. Combined DSD and DCD

1) combined DSD



<Figure 5. > selectOrderRule Combined DSD

2) combined DCD

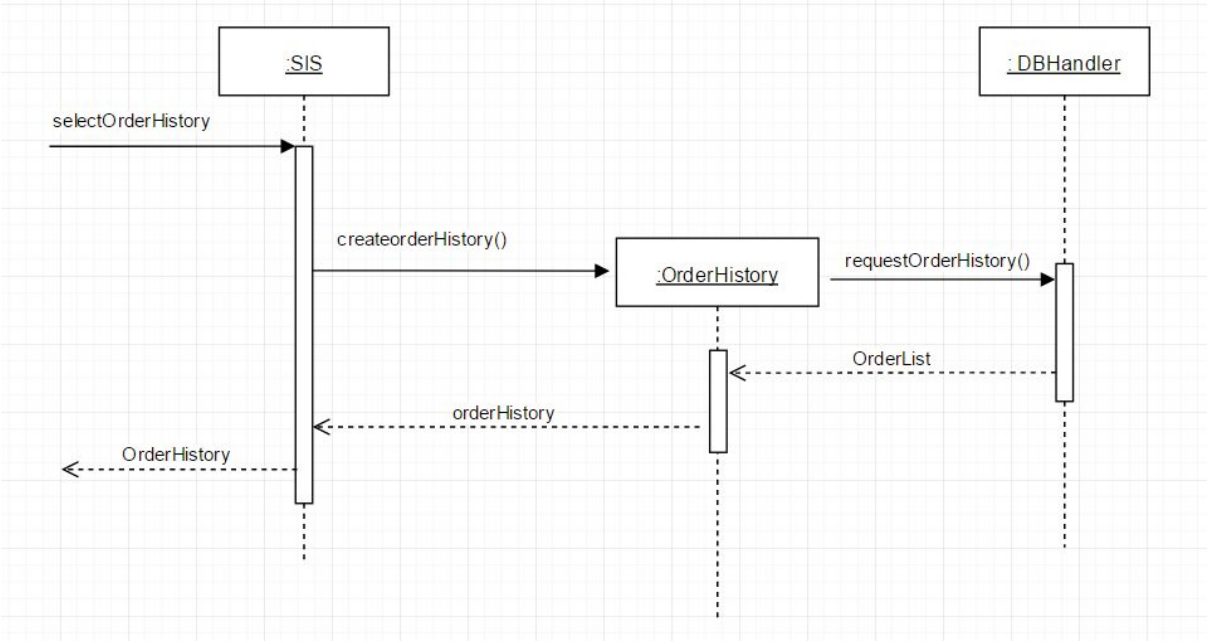


<Figure 5. > selectOrderingTime Combined DCD

5.2-2. Order History

5.2-2.1. Design Sequence Diagram

1) selectOrderHistory()



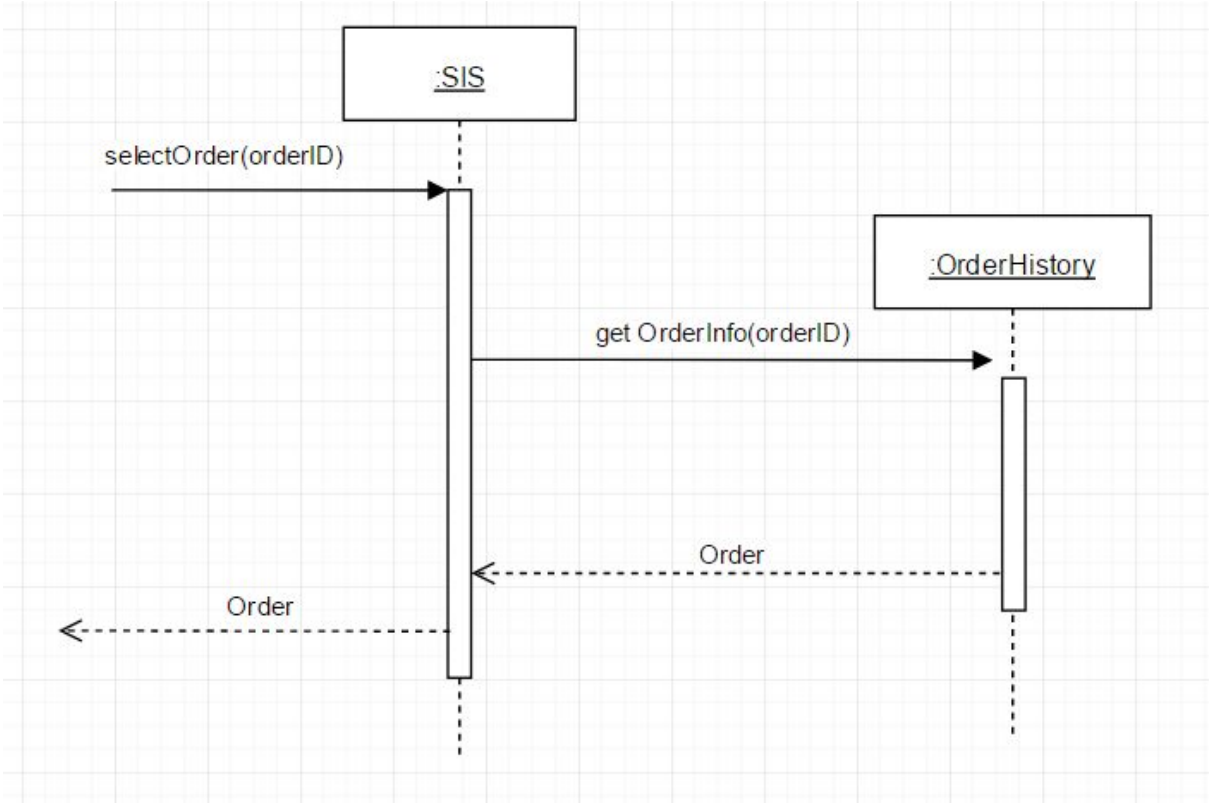
<Figure 5. > selectOrderHistory Design Sequence Diagram

GRASP Pattern	Description
Creator	
Information Expert	
High Cohesion	
Low Coupling	
Controller	-SIS is a facade Controller. SIS is ‘root object’ for overall system
Polymorphism	
Pure Fabrication	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.
Indirection	
Protected Variations	

<Table 5.> selectOrderHistory DSD GRASP Pattern

GOF Pattern	Description
Adapter	- None
Factory	- None
singleton	- None
Strategy	- None
Composite	- None
Facade	- None
Observer	- None

2) selectOrder(orderID)



<Figure 5.> selectOrder Design Sequence Diagram

GRASP Pattern	Description
Creator	
Information Expert	

High Cohesion	
Low Coupling	
Controller	-SIS is a facade Controller. SIS is 'root object' for overall system.
Polymorphism	
Pure Fabrication	
Indirection	
Protected Variations	

<Table 5.> selectOrder DSD GRASP Pattern

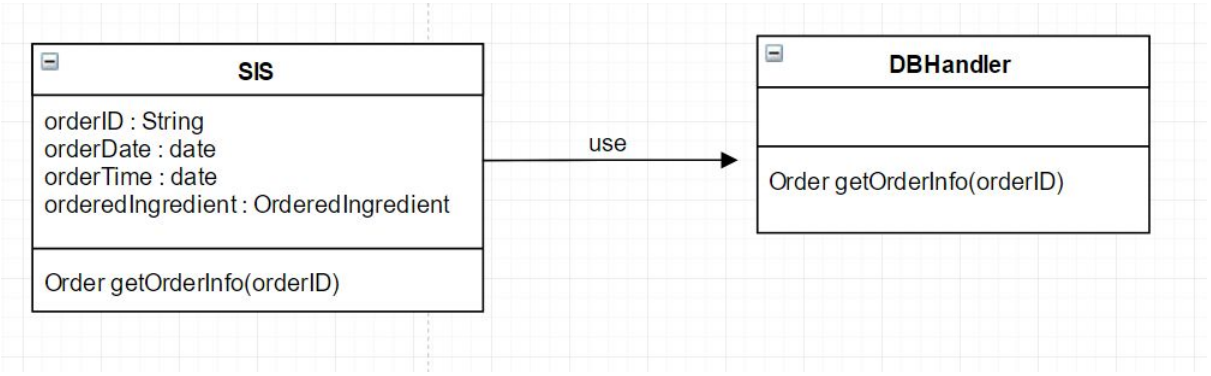
GOF Pattern	Description
Adapter	- None
Factory	- None
singleton	- None
Strategy	- None
Composite	- None
Facade	- None
Observer	- None

5.2-2.2. Design Class Diagram

1) selectOrderHistory()

<Table 5.> selectOrderHistory Design Class Diagram

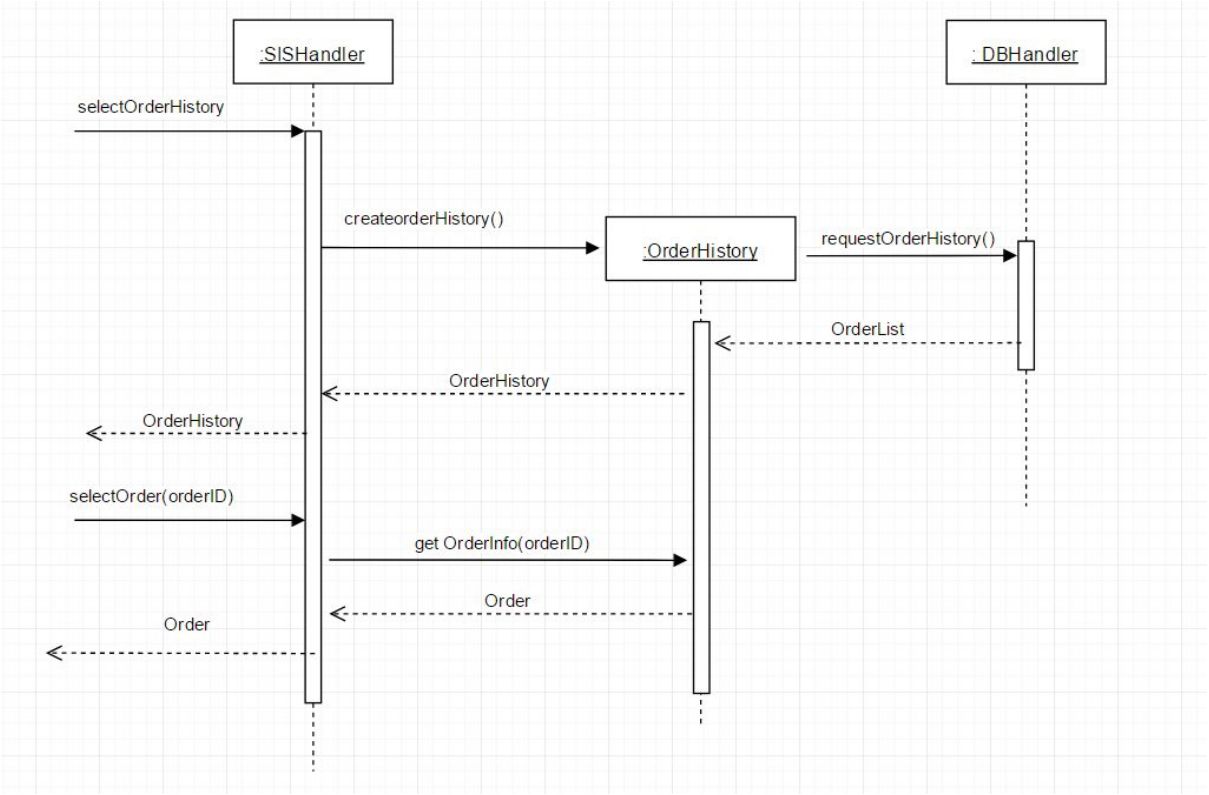
2) selectOrder(orderID)



<Table 5.> select Order Design Class Diagram

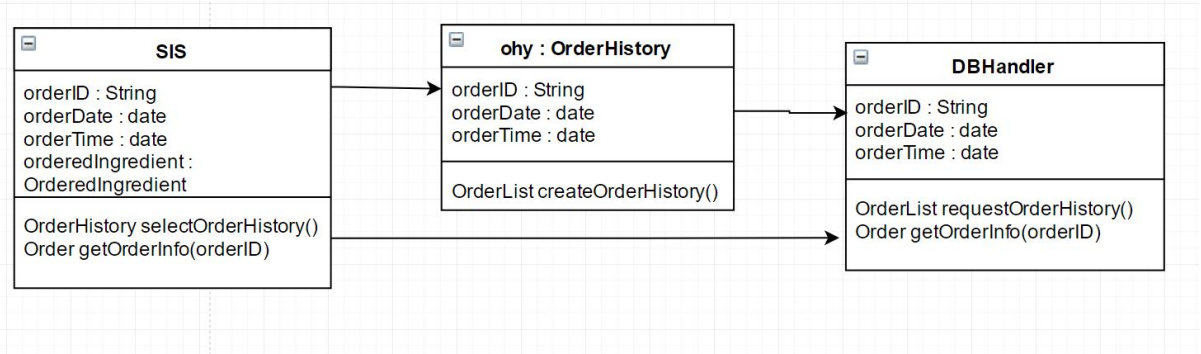
5.2-2.3. Combined DSD and DCD

1) combined DSD



<Figure 5.> Order History Combined DSD

2) combined DCD

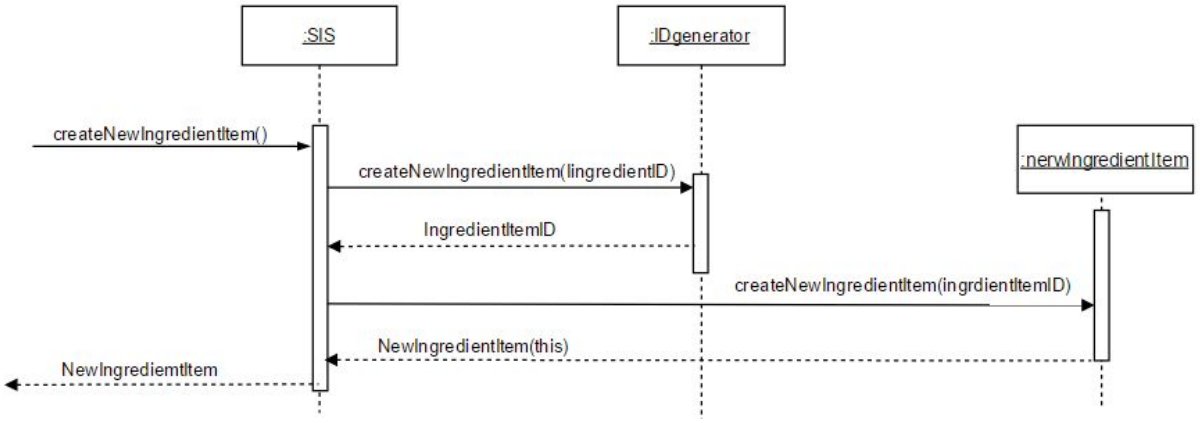


<Figure 5.> Order History Combined DCD

5.3. Manage inventory Realization

5.3.1. Design Sequence Diagram

1) createNewIngredientItem()

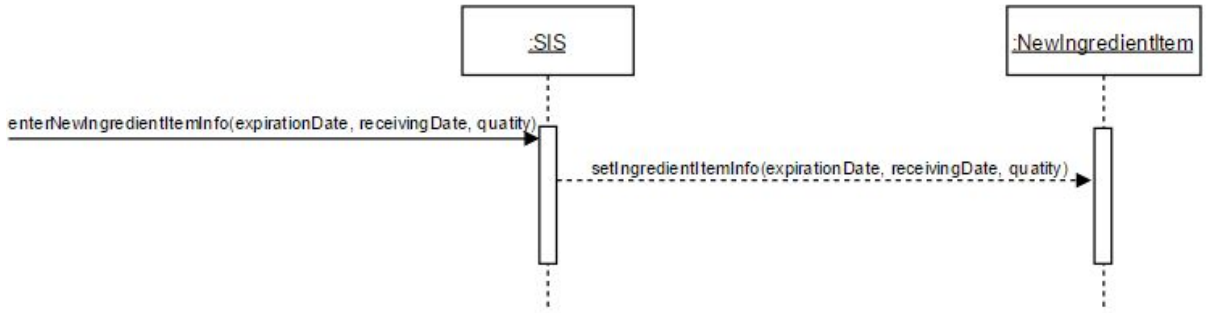


<Figure 5.> createNewIngredientItem Design Sequence Diagram

GRASP Pattern	Description
Low Coupling	- The ingredient catalog class has information about all materials, but SIS directly creates a nerwIngredientItem object to reduce coupling.
Controller	-SIS is a facade Controller. SIS is 'root object' for overall system.
Protected Variations	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

<Table 5.>createNewIngredientItem DSD GRASP Pattern

2) enterNewIngredientItemInfo(expirationDate, receivingDate, quatity)

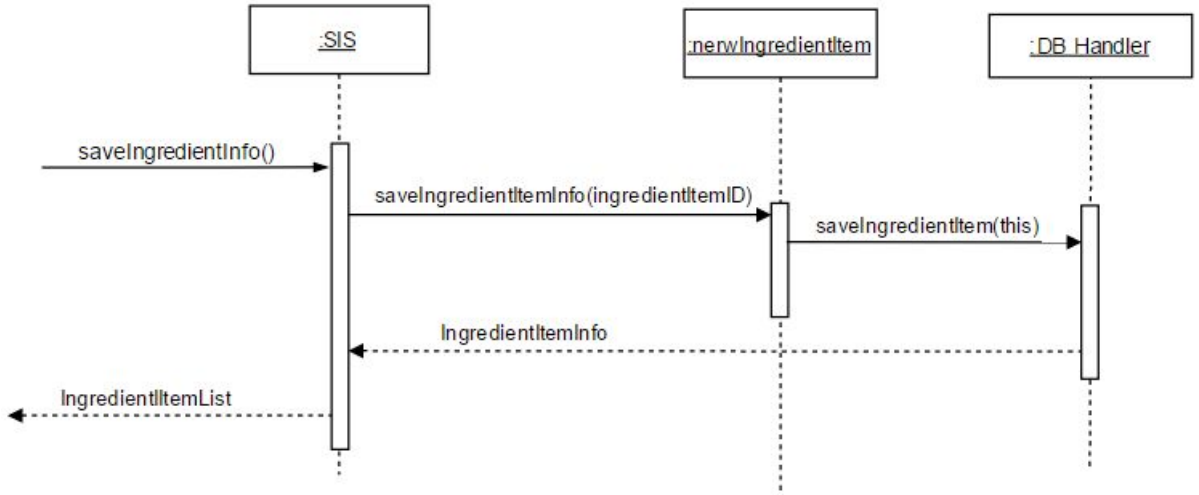


<Figure 5.>enterNewIngredientItemInfo Design Sequence Diagram

GRASP Pattern	Description
Controller	-SIS is a facade Controller. SIS is 'root object' for overall system
Pure Fabrication	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

<Table 5.>enterNewIngredientItemInfo DSD GRASP Pattern

3) saveIngredientInfo()

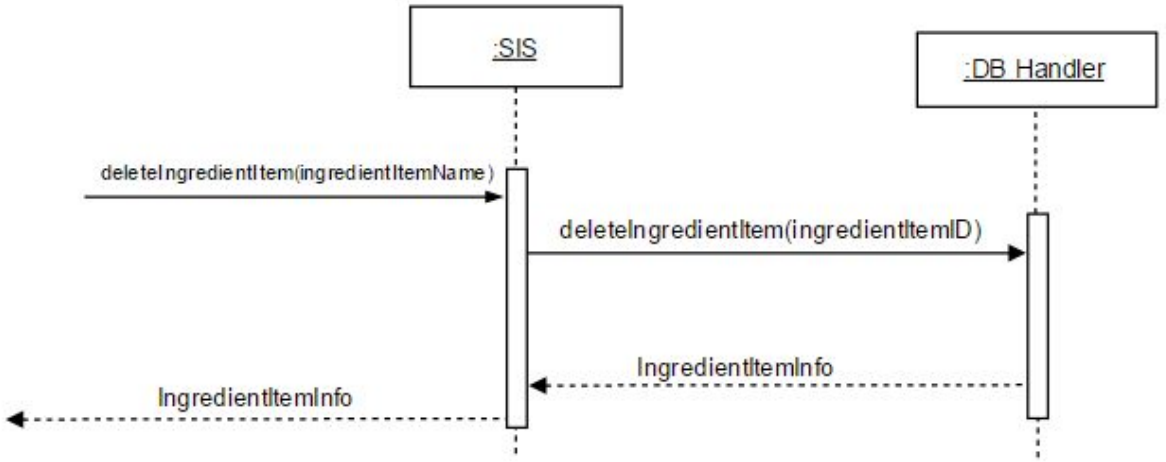


<Figure 5.>saveIngredientInfo Design Sequence Diagram

GRASP Pattern	Description
Low Coupling	-Reduce coupling by showing IngredientItemInfo which means ‘saved’, directly through SIS rather than through a newIngredientItem object.
Controller	-SIS is a facade Controller. SIS is ‘root object’ for overall system
Pure Fabrication	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

<Table 5.>saveIngredientInfo DSD GRASP Pattern

4) deleteIngredientItem(ingredientitemName)

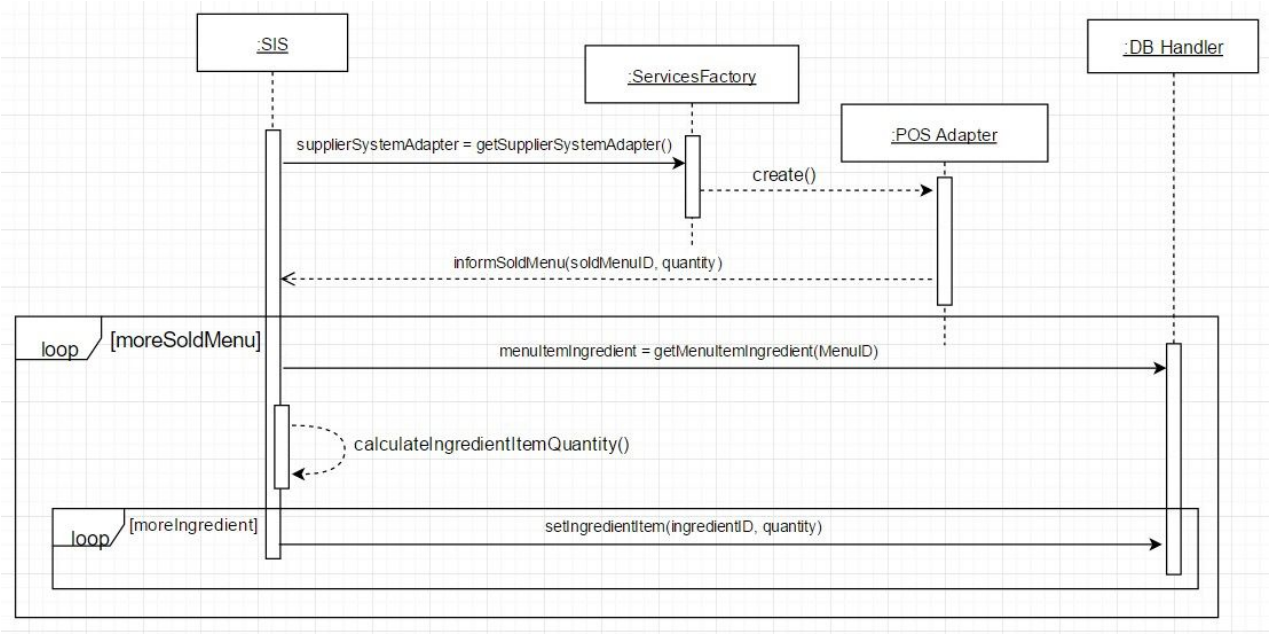


<Figure 5.>deleteIngredientItem Design Sequence Diagram

GRASP Pattern	Description
Low Coupling	-When deleting a IngredientItem, rather than associating it with a IngredientItem object that matches the IngredientItemID you want to delete, you simply pass the IngredientItemID you want to delete to the DBHandler and delete it from the DB, so reduce coupling.
Controller	-SIS is a facade Controller. SIS is 'root object' for overall system.

<Table 5.>deleteIngredientItem DSD GRASP Pattern

5) modify ingredientItemInfo by POS



<Figure 5.>modify ingredientItemInfo by POS Design Sequence Diagram

GRASP Pattern	Description
Controller	-SIS is a facade Controller. SIS is 'root object' for overall system.
Pure Fabrication	-Having a pure fabrication object named DB Handler for high cohesion and low coupling. Handler acts as an intermediator to DB, which is external service.

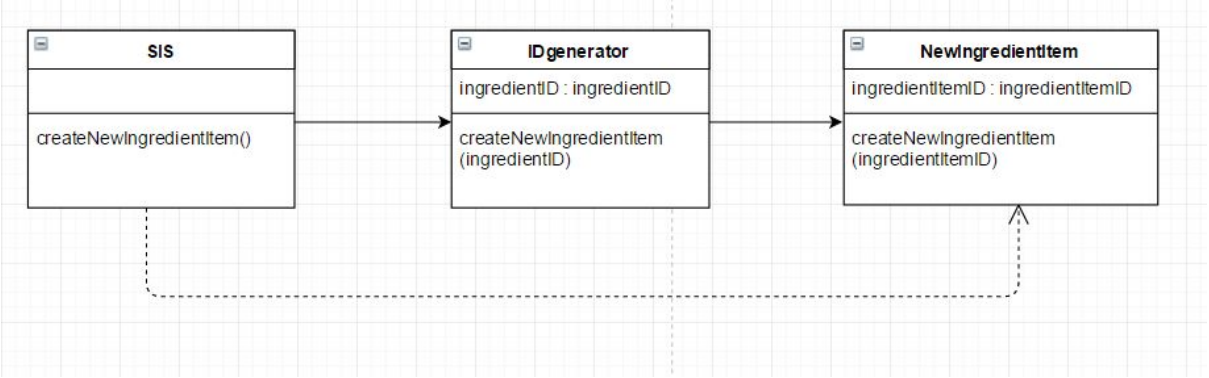
<Table 5.>modify ingredientItemInfo by POS DSD GRASP Pattern

GOF Pattern	Description
Adapter	- 'POSAdapter' needs to support several kinds of external POS system.
Factory	- 'ServicesFactory' create a Pure Fabrication object called a Factory that handles the creation.

<Table 5.>modify ingredientItemInfo by POS DSD GOF Pattern

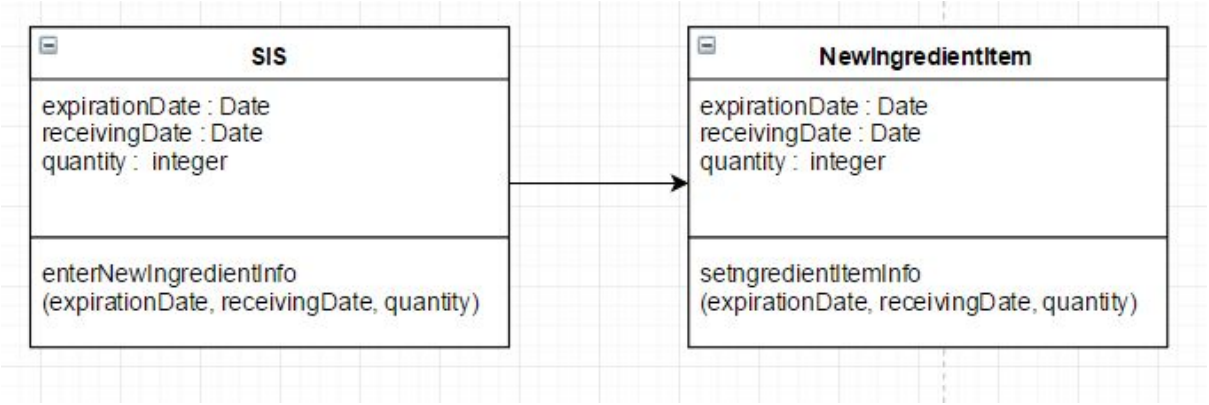
5.3.2. Design Class Diagram

1) createNewIngredientItem()



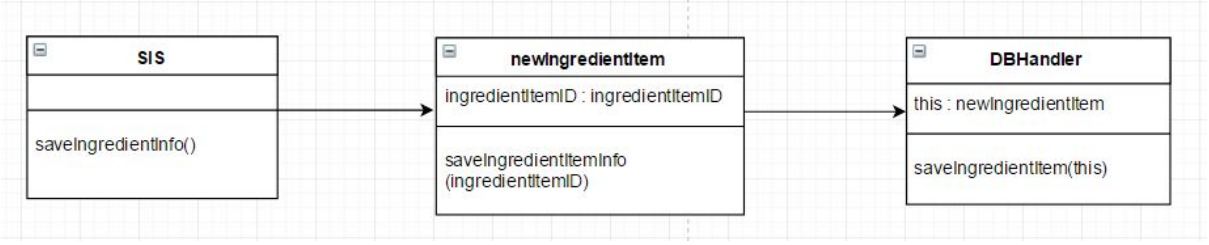
<Figure> createNewIngredientItem Design Class Diagram

2) enterNewIngredientInfo(expirationDate , receivingDate , quantity)



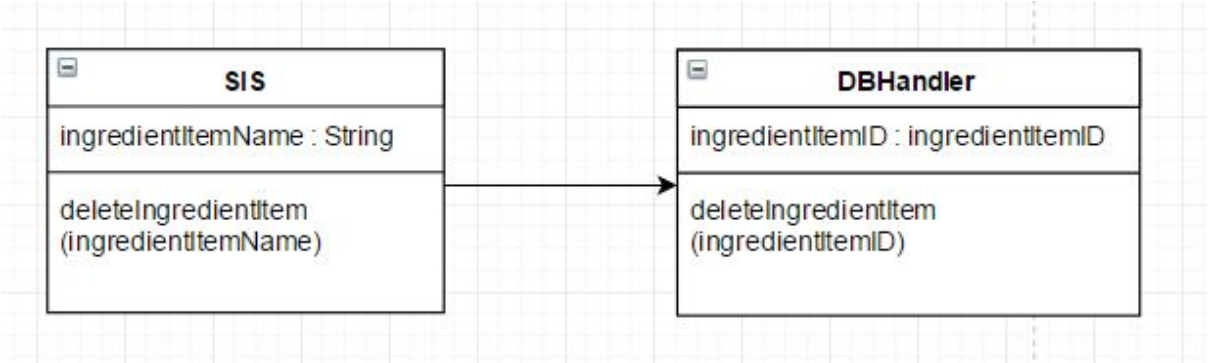
<Figure> enterNewIngredientInfo Design Class Diagram

3) saveIngredientInfo()



<Figure> saveIngredientInfo Design Class Diagram

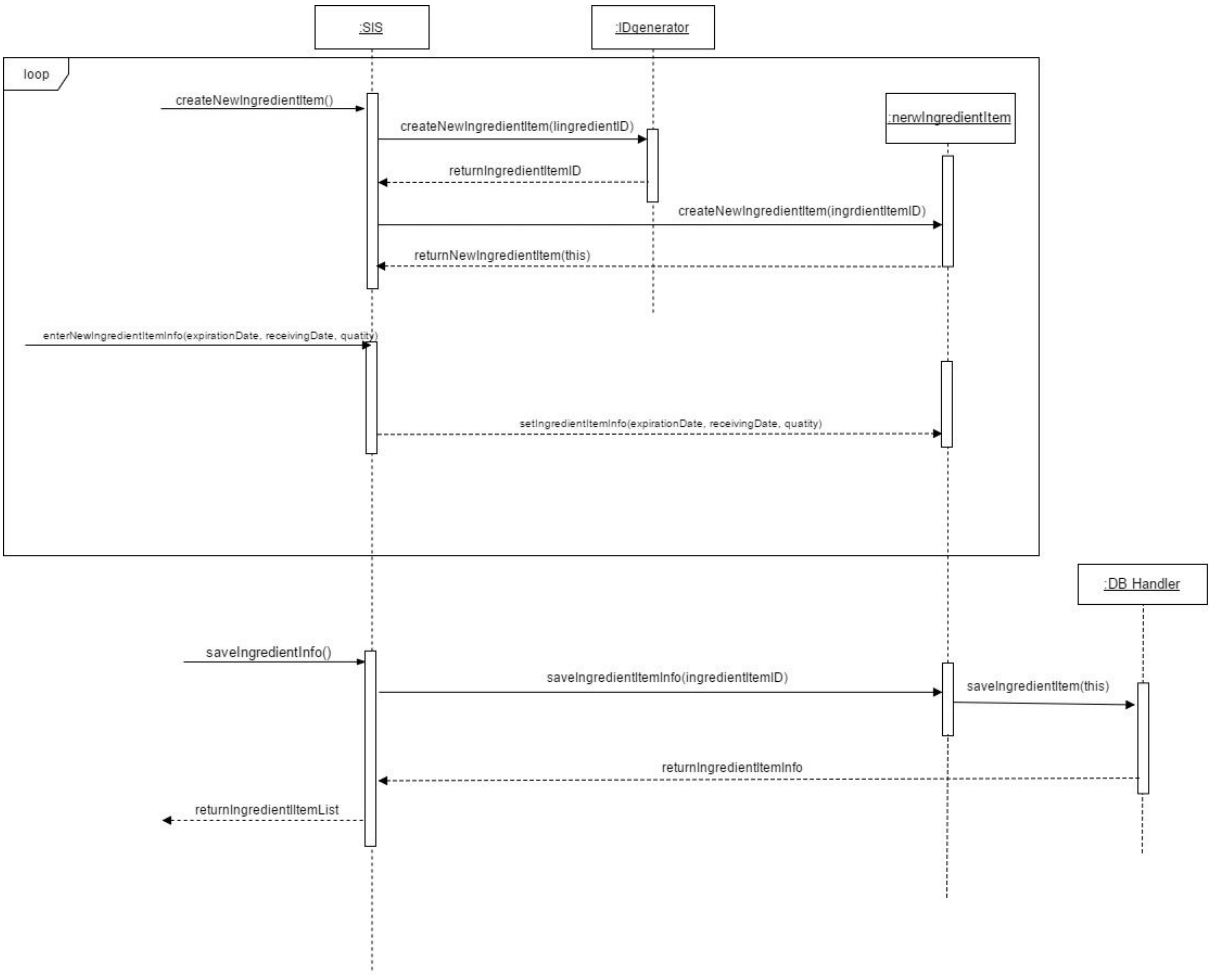
4) deleteIngredientItem(ingredientName)



<Figure> deleteIngredientItem Design Class Diagram

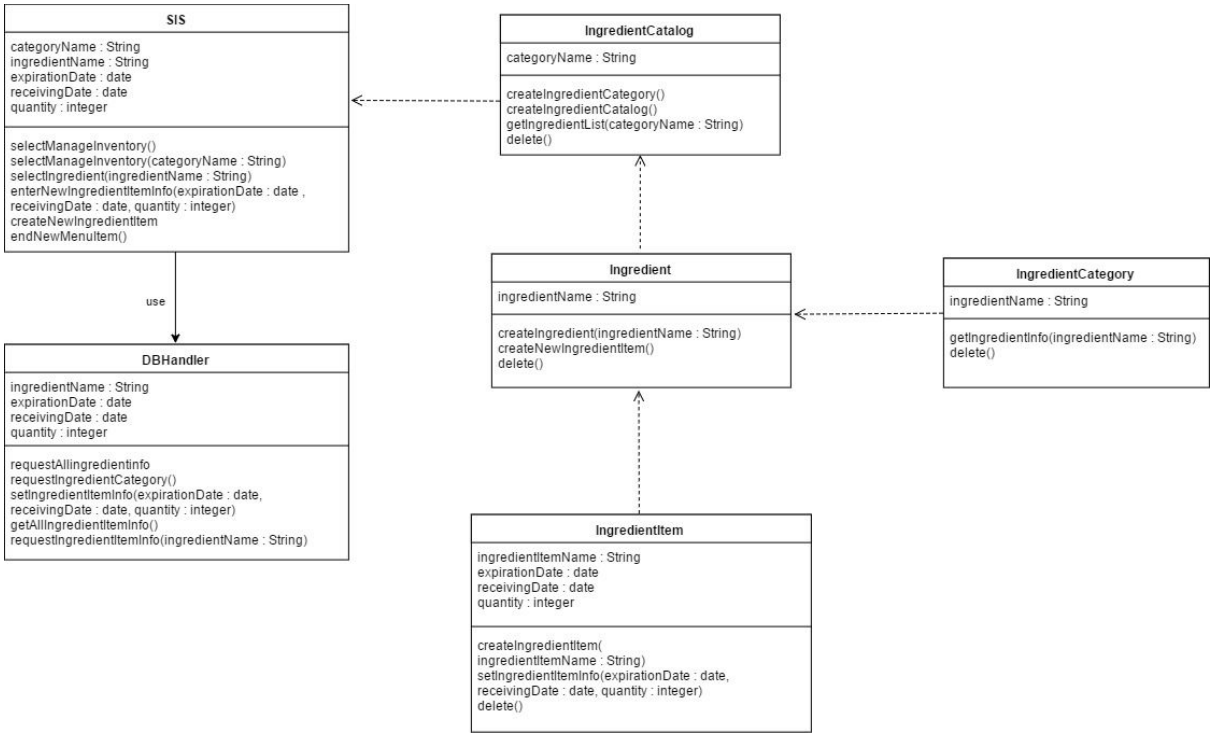
5.3.3. Combined DSD and DCD

1) Combined DSD



<Figure> Manage Inventory Combined DSD

2) Combined DCD



<Figure> Manage Inventory Combined DCD

6. Design Class Diagram of the system

7. Architecture

7.1. Introduction.

7.1.1. Logical View

Describe ‘Cafe Inventory Management System’ in view point of layer (UI, Domain, Technical services), package, framework, subsystem, interface. Show UC1, UC3’s implementation scenario looks like interaction diagram.

7.1.2. Process View

It shows main ‘Cafe Inventory Management System’ execute process. Grouping thread to process and each threads have their executive factor.

7.1.3. Deployment View

It shows relationships about server and database.

7.1.4. Use Case View

It shows use-case view about UC1,UC2-1,UC2-2,UC3 which was implemented based on each scenario.

7.1.5. Data View

It shows data flows about UC1,UC2-1,UC2-2,UC3. These are show relation between data flow and database.

7.1.6. Implementation View

It shows application (implementation scene) and source code about ‘Cafe Inventory Management System’.

7.2. Architectural Factors

Factor	Measures and quality scenarios	Variability	Impact of factor	Priority for success	Difficulty or Risk
Reliability - Recoverability					
정기적인 점검을 한다	시스템은 한 달에 1번, 1시간씩 점검됩니다 (오전 2시 부터 새벽3시).	Current flexibility: - 요인에 의해 기술된 것과 동일하다. Evolution : - none	설계에 미치는 영향력은 적다.	H	M
외부 서비스와 연결 실패로부터 복구한다	외부 서비스와 연결을 실패 했을때 3분내에 재연결을 하 이용가능하게 만든다.	Current flexibility: - SME는 현재 시스템이 작동하지 않는다고 말하고 나중에 접속할 것을 추천한다. Evolution : - 3년내에 서비스를 이용하는 사용자의 수가 늘어난다. 그 사용자에 따른 외부서비스의 과부하가 에상된다.	대규모 설계에 큰 영향을 미친다.	H	M
Reliability - Security					
내부 데이터베이스를 보호한다.	user가 authentication을 시도 하거나 데이터 베이스의 값을 가져오고자 할 때 input을 validation 한다.	Current flexibility: - 요인에 의해 기술된 것과 동일하다. Evolution : - none	설계에 미치는 영향력은 적다.	H	H
Performance - Response Time					
응답시간을 줄인다.	Updating inventory or ingredient information is done within 5 seconds 90% of time. View existing	Current flexibility: - 요인에 의해 기술된 것과 동일하다. Evolution : - 3년 이내에 업데이트 정보를 3초안에 95% 이상의 시간안에 업데이트 한다.	대규모 설계에 큰 영향을 미친다.	M	L

	inventory or ingredient information and order history is done within 3 seconds 90% of time.	3년 이내에 존재하는 정보를 1초안에 95% 이상 보여준다.			
Performance - Throughput					
일정시간동안 처리할 수 있는 양을 늘린다.	사용자가 발생시키는 이벤트를 신속하게 처리한다.	Current flexibility: - 5초 이내에 사용자의 주요 이벤트를 95%이상 처리한다. Evolution : - 3년내에 3초이내에 사용자의 주요 이벤트를 95%이상 처리한다.	대규모 설계에 큰 영향을 미친다.	M	L
Supportability - Adaptability					
다양한 환경을 지원한다.	다른 환경을 사용하는 유저가 Cafe Inventory System을 사용하기 원하는 경우 15일 내에 지원한다.	Current flexibility: - as described by factor Evolution : - none	많은 요소로부터 보호된 변형 관점에서 볼 때 설계에 중대한 영향을 미친다. 예를 들어, 운영체제와 UI 는 각각 다르다.	M	H

7.3. Architectural Decisions (Technical memos)

7.3.1. Technical Memo 1

Issue : Reliability - 외부 서비스와 연결 실패로부터 복구한다

Solution Summary : To recover from errors with external services and connection failures

Factor :

- Recovery from external services connection failures

Solution :

- Provide a local service server. Prevent unexpected system failures. Provide a local implementation of the database. And the database of information about user information, history information, and system state before failure will be a small cache recently reserved. Therefore, if a failure occurs with an external system, use cache storage to recover.

Motivation :

- Users don’t want to have problem using Cafe Inventory Management System. Therefore, if we offer this level of reliability, it will be a very attractive service. And, we want to provide our system eternally without server failure.

Unresolved Issues :

- None.

Alternatives Considered :

- None.

7.3.2. Technical Memo 2

Issue : Supportability - 다양한 환경을 지원한다

Solution Summary : Protected Variation using interfaces and Devices

Factor :

- Support other environment(mobile phone, tablet, …)

Solution :

- Provide common interface for different Operation System.(Android, IOS, …). For example, you can use swift and android. With various environments, users can use our Cafe Inventory Management System service comfortably in all device.
-

Motivation :

- These days, there are a lot of devices with different Operation System. Users want to get consistent services for various device they use.

Unresolved Issues :

- None.

Alternatives Considered

- None.

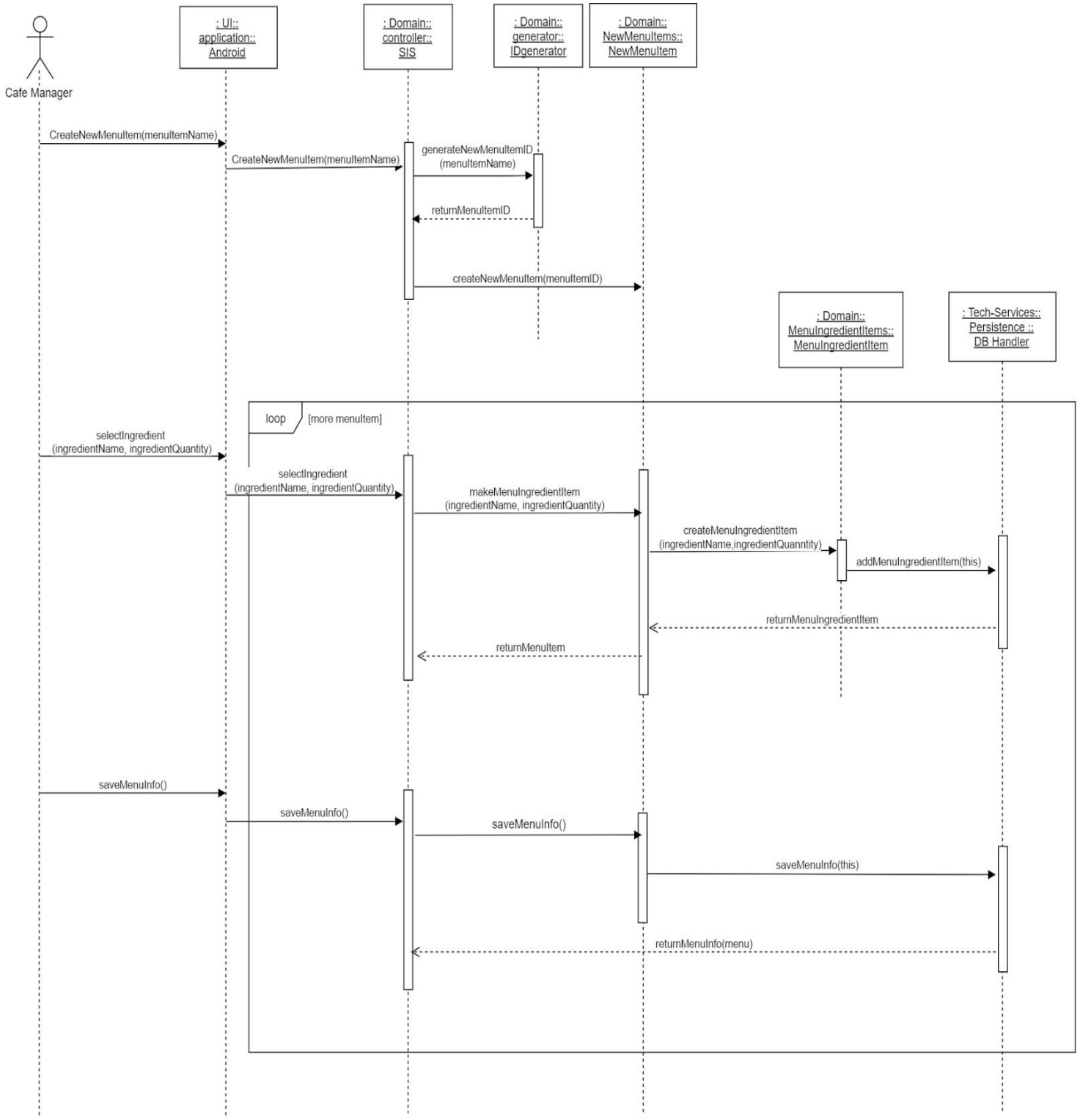
7.4. Logical View

7.5. Process View

7.6. Deployment View

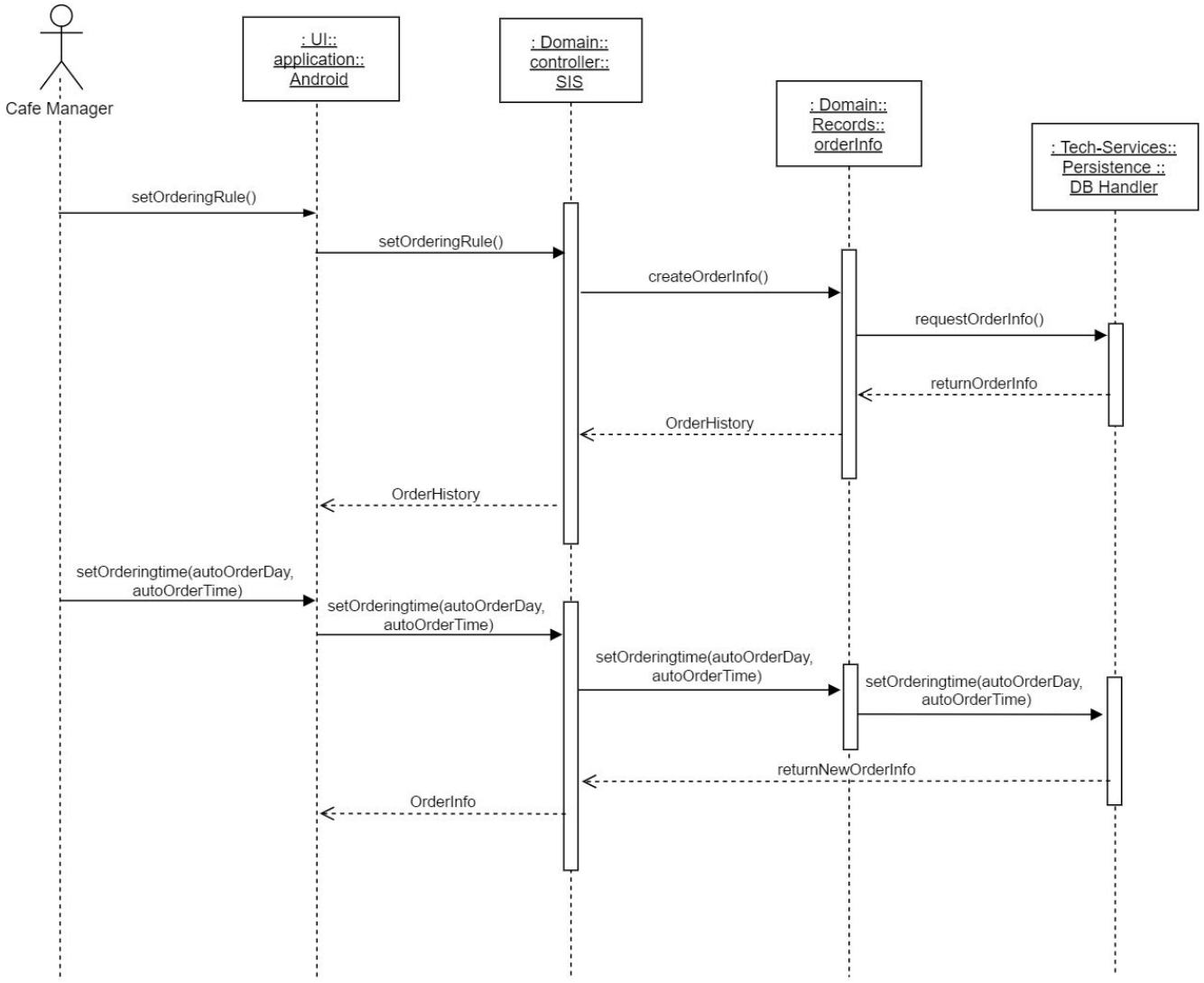
7.7. Use Case View

7.7.1. Manage menu & ingredients



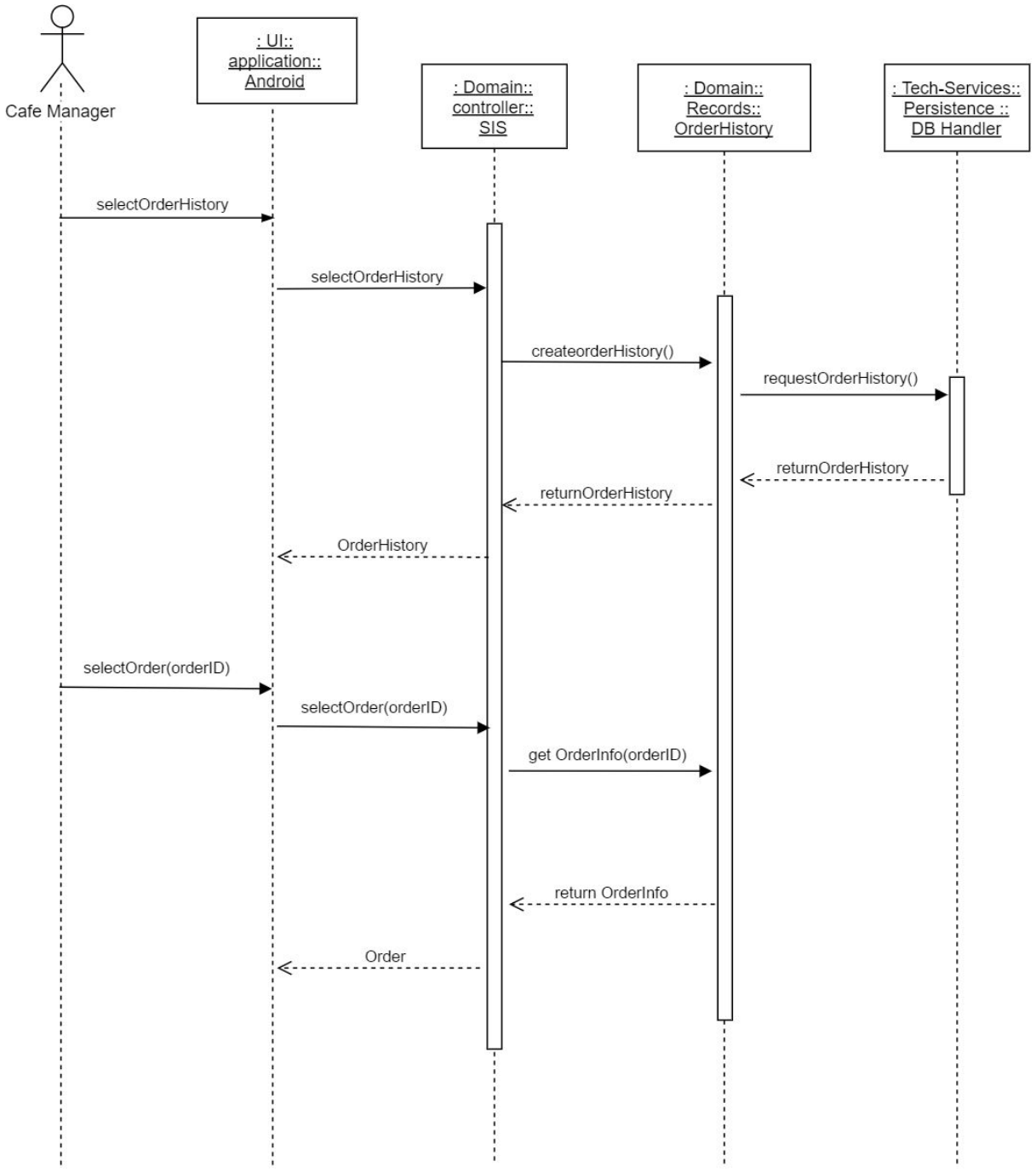
<Figure> Manage menu & ingredients Use Case View

7.7.2. Set ordering rule



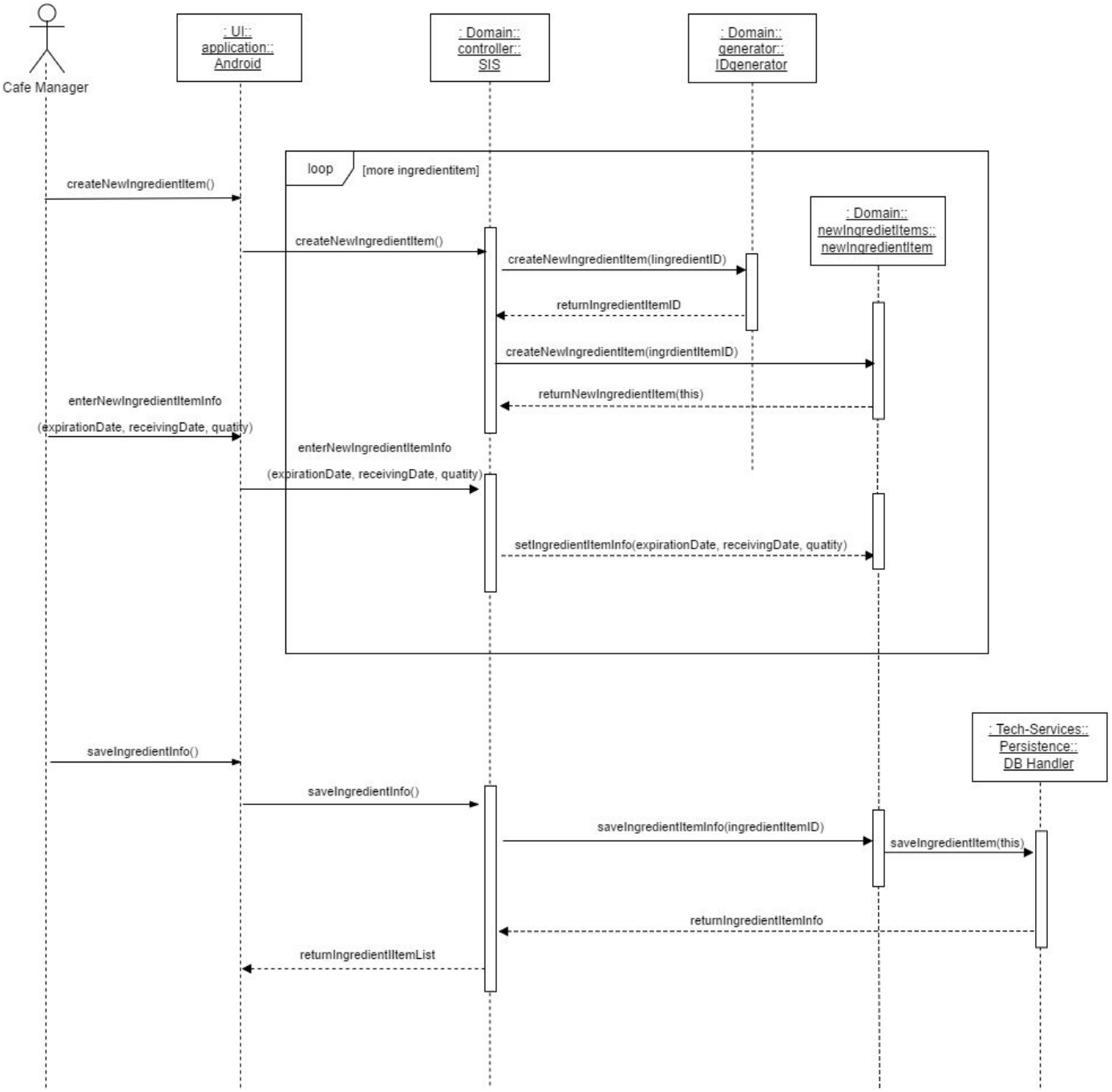
<Figure> Set ordering rule Use Case View

7.7.3. Show order history



<Figure> Show order history Use Case View

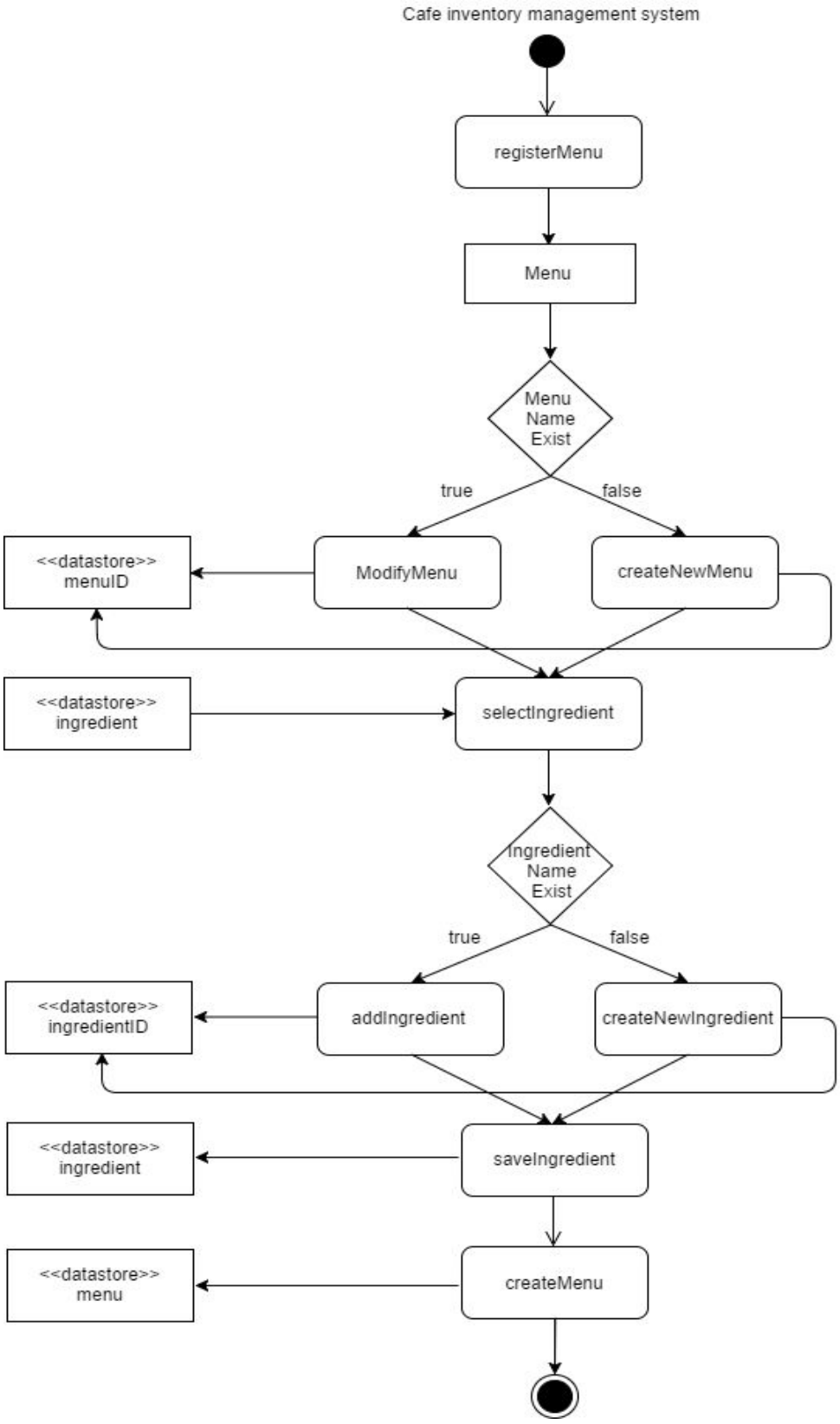
7.7.4. Manage inventory



<Figure> Manage inventory Use Case View

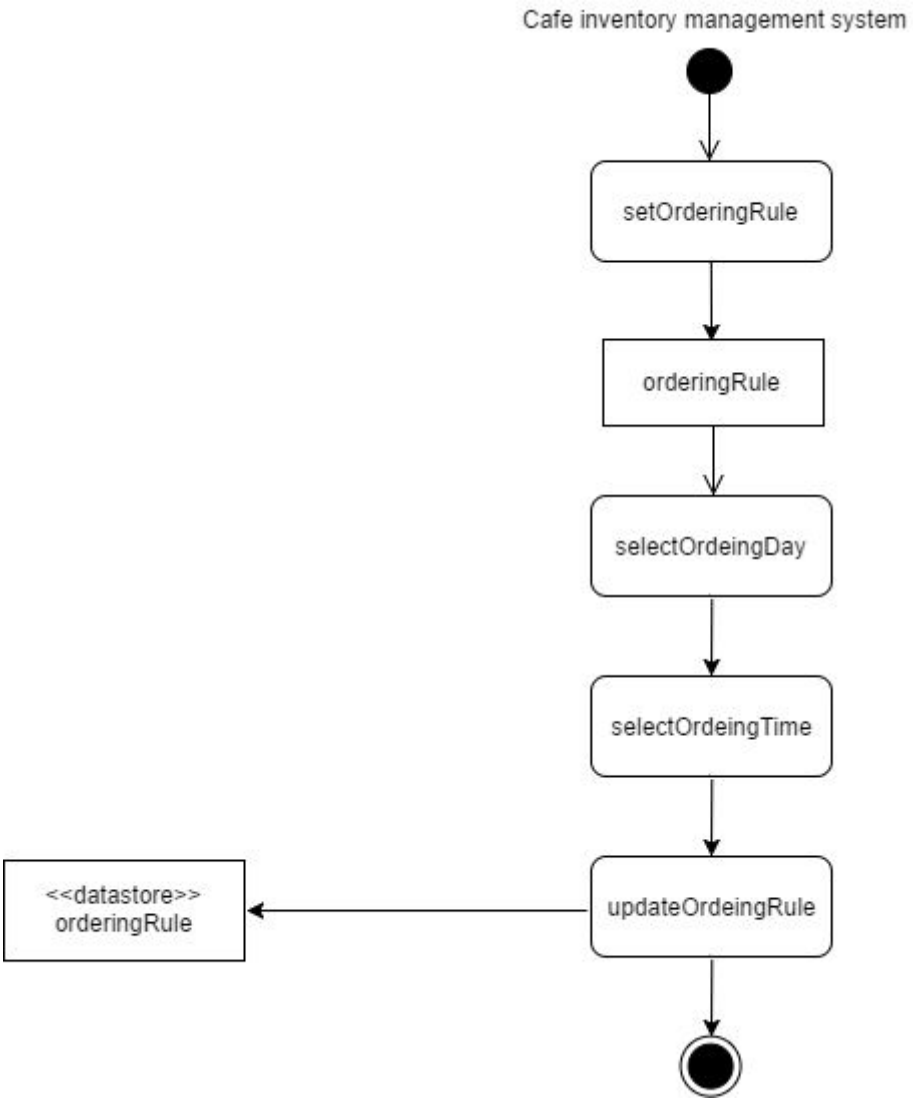
7.8. Data View

7.8.1. Manage menu & ingredients



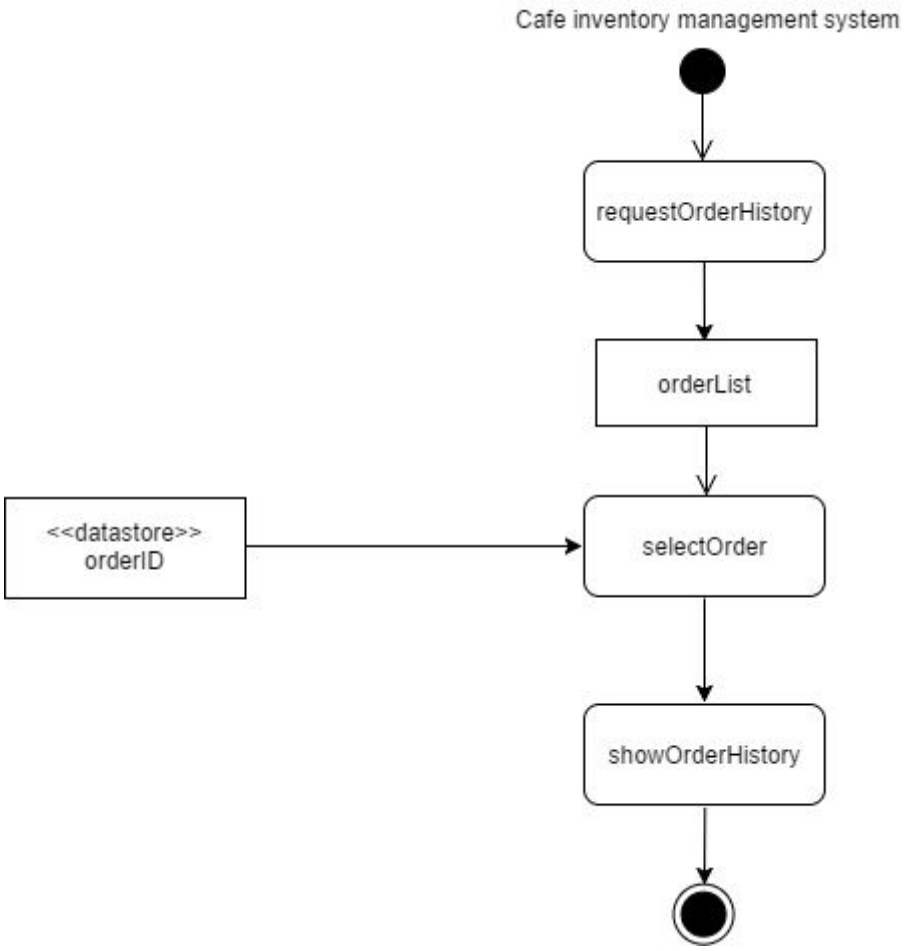
<Figure> Manage menu & ingredients Data View

7.8.2. Set ordering rule



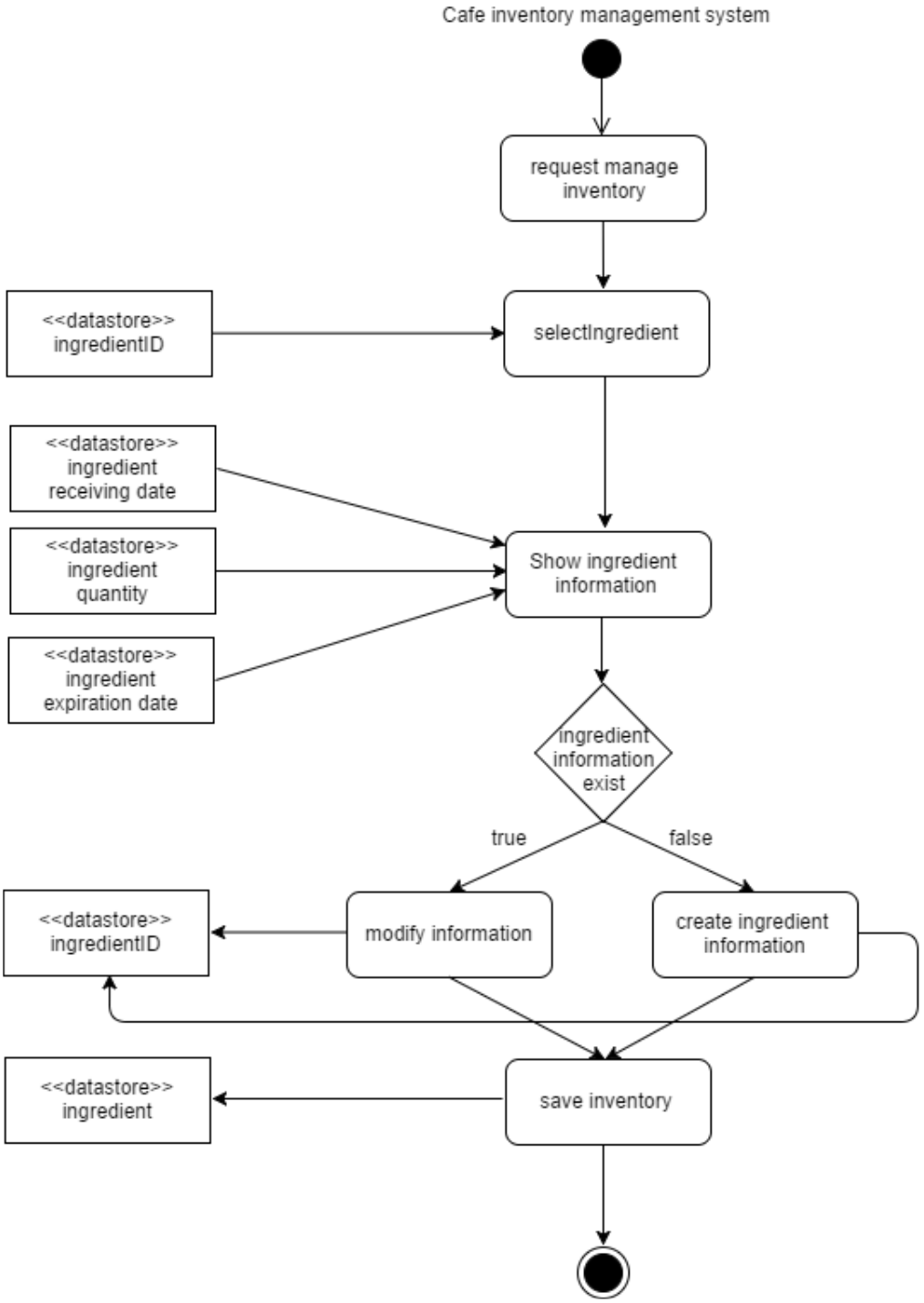
<Figure> Set ordering rule Data View

7.8.3. Show order history



<Figure> Show order history Data View

7.8.4. Manage inventory



<Figure> Manage inventory Data View

7.9. Implementation View

8. Conclusion

8.1. Objectives

- Developed a cafe inventory management system application service that helps cafe manager manage food inventory and expiration date easily and accurately, and enables food suppliers to easily manage deliveries by store. This service can be extended to such restaurants as well as a cafe using a food material.

8.2. Current State of Design & Implementation

- Design is 80% done.
  Because our design doesn't include Security/Implementation/Development view.
- Implementation is 70% done. We implemented following use case instances.
  - UC1. Manage menu & ingredients
  - UC2. Order Ingredients
  - UC2-1. Set ordering Rule
  - UC2-2. Show orderHistory
  - UC3. Manage Inventory
- Implementation in detail :
  @ Manage menu & ingredients
  1)
  2)
  3)
  @ Order Ingredients
  1)
  2)
  3)
  @ Set ordering Rule
  1)
  2)
  3)
  @ Show orderHistory
  1)
  2)
  3)
  @ Manage Inventory
  1)
  2)
  3)

8.3. Addition Work(Refine the design)

- None.

8.4. Lesson learned through this project

Through this class, we learned objecanalysis t oriented analysis and design. First, we discussed about the idea. And then, we started to analyze and design step by step. Because it was our first experience of object oriented and design, we made a lot of mistakes from the beginning. So we had meeting whenever we have time. We continuously modified what we were doing. Thanks to all members effort, we could complete our project. Of course, the result is not perfect, but we learned a lot from it. We understood the procedure of object oriented analysis & design. We learned how to work in team. We learned a lot by trial and error. So, we believe that we can do better next time.

9. References

- 『Applying UML And Patterns, 3rd Edition』, Craig Larman, Addison Wesley Professional
- 통계청 *Statistics Korea*

10. Appendix

A. Glossary

<Revision History>

Version	Date	Description	Author
Inception draft	Mar 22, 2017	First draft. To be refined elaboration step.	Team 공조
Elaboration 1	Apr 29, 2017	Elaboration 1. Refined version to the previous version.	Team 공조
Elaboration 2/Final	June 05, 2017	Final draft	Team 공조

<Definitions>

Term	Definition and Information	Format	Validation Rules	Aliases
Store Manager	person who manages inventory in the store.	-	-	-
Supplier	person who supplies inventory to the Store Manager. 하나의 서플라이어가 모든 재고를 공급한다	-	-	-
Administrator	person who manages this system.	-	-	-
SIS	Cafe Management Invertory System	-	-	-
Holding Time	The holding time is the period that can be kept from the date of receipt of the ingredient	-	-	-