

THE FINAL PROJECT REPORT

컴퓨터공학

20153180 박정은

1. POST System에 대한 Vision과 Scope

Vision:

- 소비자가 물건을 구매하고자 할 때 관리자는 위 시스템을 이용해서 결제를 진행할 수 있고 결제 정보는 기록된다.
- 관리자는 해당 시스템을 이용해서 물품을 관리할 수 있다.
- 모든 operation은 5초를 넘기지 않도록 한다.
- 결제가 완료되면 재고 수정을 자동적으로 실행한다.
- 사용자가 시스템을 손쉽게 사용할 수 있도록 화면이 구성되어 있다.

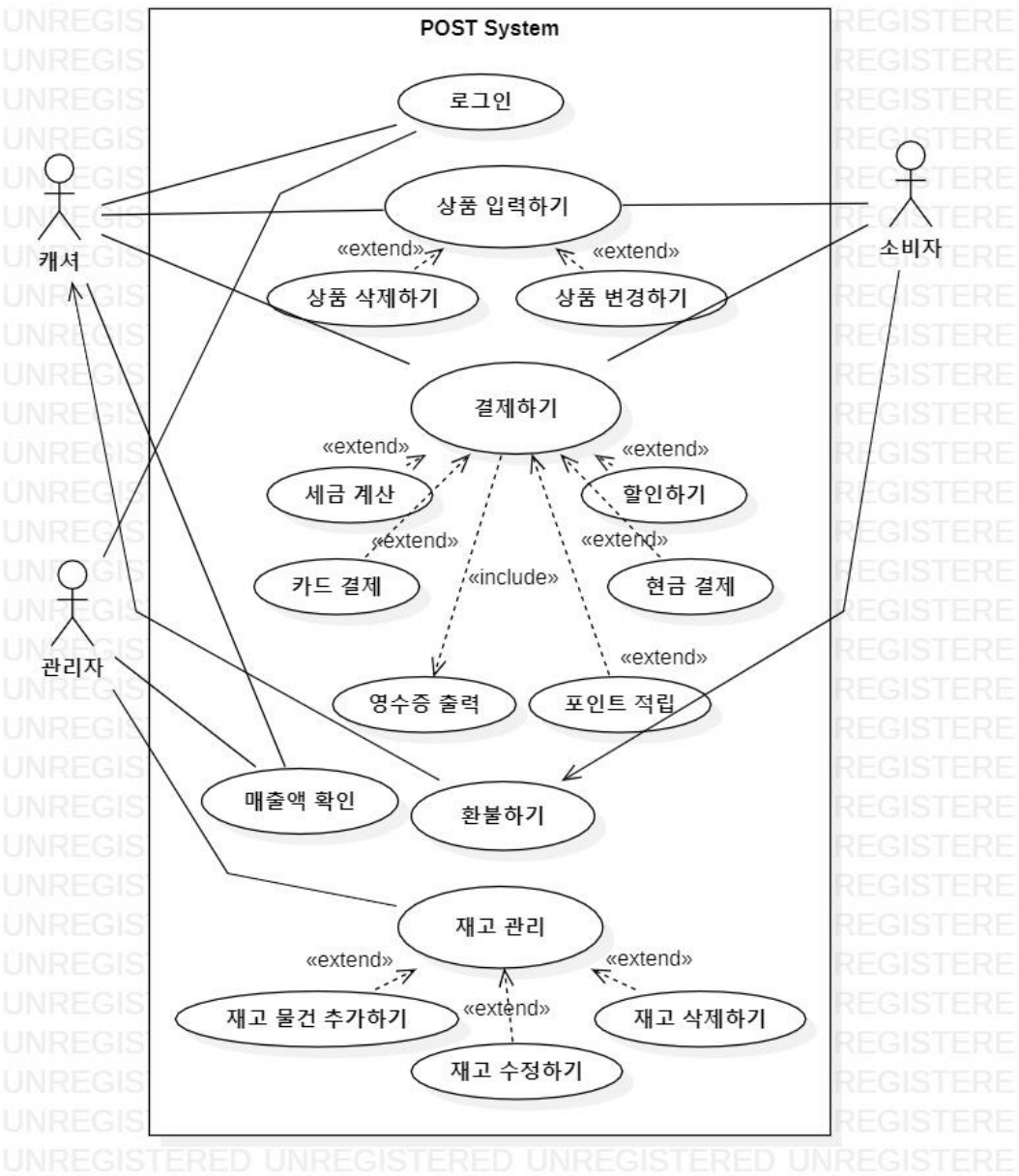
Scope:

- Basic POS application

위 시스템은 전반적인 POS 기능들을 포함한다. (재고관리, 물건 계산, 계산 기록)

2. Use Case

2.1) Use Case Diagram



2.2) Use Case Description

Use case Name	상품 입력하기	
Scenario	캐시가 소비자가 구매하려는 상품들을 포스기에 입력한다.	
Triggering event	소비자는 구매하려는 물건들을 올려놓는다.	
Brief description	소비자는 가져온 물건들을 올려놓는다. 캐시는 소비자가 올려놓은 물건들을 입력한다.	
Actors	캐시	
Related use cases		
Stakeholders	편의점 사장님, 소비자	
Preconditions	소비자가 한 개 이상의 물건을 가져와야 한다.	
Post conditions	소비자가 가져온 물건이 모두 입력된다.	
Flow of activities	Actor	System
	1.소비자는 가져온 물건들을 올려놓는다 2.캐시는 소비자가 가져온 물건들을 하나씩 포스기에 입력한다.	2.상품 입력창이 띄워져있다 3.캐시가 상품 정보를 입력할 때마다 상품 입력창에 입력된 상품에 대한 정보가 출력되고 총액이 계산된다.
Exception conditions	유통기한이 지난 음식이 있다면 이를 알린다. 구매 취소 시 상품 입력창을 초기화 시킨다.	

Use case Name	상품 삭제하기	
Scenario	캐시가 소비자가 원하지 않는 물품을 삭제한다.	
Triggering event	소비자는 원하지 않은 물건에 대해서 취소를 요청한다.	
Brief description	소비자는 변심으로 어떤 물건에 대한 취소를 요청한다. 캐시는 해당 물건을 포스기에 입력된 내용에서 삭제한다.	
Actors	캐시	
Related use cases	상품 입력하기	
Stakeholders	편의점 사장님, 소비자	
Preconditions	소비자가 한 개 이상의 물건을 취소하려 한다.	
Post conditions	소비자가 요청한 물건이 삭제된다.	
Flow of activities	Actor	System
	1.소비자는 변심으로 원치 않는 물건에 대한 취소를 요청한다.	

	2.캐서는 해당 물품을 삭제 처리한다.	2.상품 입력창에 있는 해당 물품을 클릭한다. 3.삭제 버튼을 눌러 해당 물품을 제거한다.
Exception conditions	아직 입력되지 않은 물품일 경우 아무런 일이 발생하지 않는다.	

Use case Name	상품 변경하기	
Scenario	캐서가 입력된 물품에 대한 개수를 변경한다.	
Triggering event	소비자는 물품 개수를 변경하길 원한다.	
Brief description	소비자는 개수를 변경하고 싶은 물품에 대한 요청을 한다. 캐서는 입력된 해당 물품에 대한 개수 변경을 진행한다.	
Actors	캐셔	
Related use cases	상품 입력하기	
Stakeholders	편의점 사장님, 소비자	
Preconditions	소비자가 한 개 이상의 물건에 대한 변경을 요청한다.	
Post conditions	소비자가 요청했던 개수대로 해당 물건의 개수가 변경된다.	
Flow of activities	Actor	System
	1.소비자는 어떤 물건에 대한 개수 변경을 요청한다. 2.캐서는 소비자가 요청한 물건에 대한 개수 변경을 진행한다.	2.상품 입력창에 있는 해당 물품을 클릭한다. 3.변경 버튼을 눌러 해당 물품의 개수를 변경한다
Exception conditions	아직 입력되지 않은 물품일 경우 아무런 일이 발생하지 않는다.	

Use case Name	결제하기
Scenario	캐서가 입력한 물건에 대한 결제를 진행한다.
Triggering event	물건이 다 입력되면 결제를 진행한다.
Brief description	캐서가 물건을 다 입력한 후 소비자에게 결제 방법에 대해 묻는다. 소비자는 결제 방법을 결정한다. 소비자가 결정한 결제 방법으로 결제를 진행한다.
Actors	캐셔
Related use cases	
Stakeholders	편의점 사장님, 소비자

Preconditions	캐셔가 소비자가 구매하려는 물건을 모두 입력한다.	
Post conditions	소비자는 구매하려는 물품들을 결제 완료했다.	
Flow of activities	Actor	System
	1.캐셔는 소비자에게 결제 방법에 대해 물어본다. 2.소비자는 결제 방법을 선택한다. 3.캐셔는 소비자가 선택한 결제 방법으로 결제를 진행한다.	3.결제 버튼을 누르고 결제를 진행한다.
Exception conditions	구매 취소 시 상품 입력창을 초기화 시킨다.	

Use case Name	카드 결제하기	
Scenario	카드로 결제를 진행한다.	
Triggering event	소비자가 결제 방법을 카드로 정했다.	
Brief description	캐셔는 소비자에게 카드를 건네받는다. 포스기에 있는 카드 결제기에 카드를 입력시킨다. 정상적으로 결제가 될 때까지 일정시간을 기다린다. 결제 완료!	
Actors	캐셔	
Related use cases	결제하기	
Stakeholders	편의점 사장님, 소비자	
Preconditions	캐셔가 소비자에게 결제 방법에 대해 묻는다.	
Post conditions	카드 결제가 정상적으로 완료되었다.	
Flow of activities	Actor	System
	1.캐셔는 소비자에게 카드를 받는다. 2.캐셔는 받은 카드를 카드 결제기에 입력시킨다. 6.결제가 완료되면 소비자에게 다시 카드를 건넨다.	3.카드에 대한 정보를 입력받는다 4.결제를 진행할 수 있는 카드인지 확인한다. 5.정상적으로 결제를 완료한다.
Exception conditions	카드 결제가 불가능한 경우 카드 결제가 불가능하다고 알린다.	

Use case Name	현금 결제하기	
Scenario	현금으로 결제를 진행한다.	
Triggering event	소비자가 결제 방법을 현금으로 정했다.	
Brief description	캐셔는 소비자에게 현금을 건네받는다. 포스기에서 현금 결제를 누르고 현금 계산을 한다. 거스름돈이 발생하면 거스름돈을 소비자에게 건네고 정상적으로 결제를 완료한다.	
Actors	캐셔	
Related use cases	결제하기	
Stakeholders	편의점 사장님, 소비자	
Preconditions	캐셔가 소비자에게 결제 방법에 대해 묻는다.	
Post conditions	현금 결제가 정상적으로 완료되었다.	
Flow of activities	Actor	System
	1.캐셔는 소비자에게 현금을 받는다. 2.캐셔는 현금 결제를 누른다 3.받은 금액을 입력한다. 5.캐셔는 시스템에 거스름돈이 계산이 되어있으면 소비자에게 거스름돈을 건넨다.	4.받은 금액과 총액에 대한 차액을 계산한다.
Exception conditions	총액에 비해 받은 현금이 부족하면 남은 금액을 소비자에게 알린다.	

Use case Name	영수증 출력하기	
Scenario	결제가 완료되면 영수증이 출력된다.	
Triggering event	총액에 대한 결제가 정상적으로 완료되었다.	
Brief description	모든 물품에 대한 결제가 되었다. 결제가 완료된 상품들에 대한 영수증이 자동으로 출력된다.	
Actors	시스템	
Related use cases	결제하기	
Stakeholders	편의점 사장님, 소비자, 캐셔	
Preconditions	모든 결제가 완료되었다.	
Post conditions	영수증이 출력되었다.	
Flow of activities	Actor	System
	1.캐셔는 결제를 완료한다.	

		2.캐서에 대한 정보, 현재 날짜와 시간, 구매한 품목들에 대한 정보, 등등 결제 정보들을 시스템에 기록한다 3.기록된 정보들을 영수증으로 출력한다
Exception conditions	용지가 없어서 출력이 불가능할 경우 용지 부족이라고 메시지를 표시한다.	

Use case Name	포인트 적립하기	
Scenario	결제시에 포인트를 적립한다.	
Triggering event	소비자가 포인트를 적립하길 원한다.	
Brief description	소비자는 포인트를 적립하려 한다. 캐서는 소비자에게 번호를 물어본다. 번호를 입력해 결제 가격에 맞는 포인트를 적립한다.	
Actors	캐서	
Related use cases	결제하기	
Stakeholders	편의점 사장님, 소비자	
Preconditions	캐서가 소비자에게 포인트 적립 여부에 대해 묻는다.	
Post conditions	포인트 적립이 정상적으로 완료되었다.	
Flow of activities	Actor	System
	1.캐서는 소비자에게 포인트 적립을 할 건지 묻는다 2.소비자는 포인트 적립을 하겠다고 하고 번호를 불러준다. 3.캐서는 포인트 적립 버튼을 누르고 번호를 입력한다. 5.캐서는 적립하기를 누른다.	4.포인트 적립 버튼을 누르면 번호 입력창이 뜬다. 5.해당 사용자의 계정에 포인트를 적립시킨다.
Exception conditions	존재하지 않는 사용자면 소비자에게 알린다. 소비자가 취소를 원하면 포인트 적립창을 닫는다.	

Use case Name	세금 계산하기	
Scenario	총액에서 세금을 추가로 계산한다.	
Triggering event	세금 계산이 필요한 경우 세금 계산을 진행한다.	
Brief description	캐서는 총액에서 세금 계산이 필요한 경우 세금 계산 버튼을 누른다. 시스템에서는 세금을 계산하고 총액에 합산한다.	
Actors	캐셔	
Related use cases	결제하기	
Stakeholders	편의점 사장님, 소비자	
Preconditions	캐셔가 세금 계산을 필요로 한다.	
Post conditions	총액에 세금 계산이 포함되었다.	
Flow of activities	Actor	System
	1.캐서는 세금 계산을 하려한다. 2.캐서는 세금 계산 버튼을 누른다.	3.총액에 세금을 계산해 포함시킨다.
Exception conditions	이미 세금 계산이 되었다면 이미 계산되었다고 알린다.	

Use case Name	할인하기	
Scenario	총액에서 할인 금액을 계산한다.	
Triggering event	소비자가 할인 수단을 제시한다.	
Brief description	캐서는 소비자에게 할인 수단을 받는다. 포스기에서 할인하기를 누르고 할인에 대한 정보를 입력한다. 그리고 할인 버튼을 누르면 총액에서 할인 금액만큼 제해진다.	
Actors	캐셔	
Related use cases	결제하기	
Stakeholders	편의점 사장님, 소비자	
Preconditions	소비자에게 한 개 이상의 할인 수단이 있다.	
Post conditions	총액에서 할인 금액만큼 차감되었다.	
Flow of activities	Actor	System
	1.소비자는 할인 수단을 캐셔에게 건넨다 2.캐서는 할인하기 버튼을 누르고 할인 정보를 입력한다. 3.캐서는 할인 버튼을 누른다	2.할인하기 버튼을 누르면 정보 입력창이 뜬다 4.총액에서 할인 금액만큼 제한다.

Exception conditions	이미 사용된 할인 수단이면 이미 사용된 할인이라고 알린다. 존재하지 않는 할인 수단이면 사용할 수 없는 할인이라고 알린다.
----------------------	---

Use case Name	환불하기	
Scenario	소비자가 결제했던 영수증을 가져와 환불을 하려한다.	
Triggering event	소비자가 환불을 요청한다.	
Brief description	캐서는 소비자의 영수증을 받는다. 환불에서 영수증 정보를 확인하고 환불을 진행한다.	
Actors	캐서, 소비자	
Related use cases		
Stakeholders	편의점 사장님	
Preconditions	소비자는 환불하려는 영수증을 가지고 있다.	
Post conditions	정상적으로 구매 취소가 되었다.	
Flow of activities	Actor	System
	1. 소비자는 캐서에게 영수증을 건네며 환불을 요청한다. 2. 캐서는 환불하기 버튼을 누르고 영수증 정보를 입력한다. 3. 캐서는 정보를 확인하고 환불을 진행한다. 5. 캐서는 환불 영수증을 소비자에게 건넨다.	2. 환불하기 버튼을 누르면 정보 입력창이 뜬다. 4. 환불을 누르면 결제가 취소되고 환불 영수증이 나온다.
Exception conditions	환불 기한이 지난 영수증이면 소비자에게 알린다. 환불이 불가능한 제품이면 소비자에게 알린다.	

Use case Name	매출액 확인
Scenario	하루 총 매출액을 확인한다.
Triggering event	관리자가 총매출액을 확인하려 한다.
Brief description	관리자는 총매출액 버튼을 누른다. 오늘 하루 얼마나 판매했는지 체크한다.
Actors	캐서, 관리자
Related use cases	

Stakeholders	편의점 사장님	
Preconditions	포스기에 관리자로 로그인 되어있어야 한다.	
Post conditions	총매출액이 표시된다.	
Flow of activities	Actor	System
	1.관리자는 총매출액 버튼을 누른다.	2.현재까지 계산했던 총매출액을 표시한다.
Exception conditions		

Use case Name	재고 관리	
Scenario	재고가 틀리지 않게 기록 되어있는지 확인한다.	
Triggering event	관리자가 재고 관리를 위해 접속한다.	
Brief description	관리자로 로그인한다. 관리자는 현재 재고가 어떻게 기록되어 있는지 체크한다.	
Actors	관리자	
Related use cases		
Stakeholders	편의점 사장님, 택배기사님	
Preconditions	관리자로 로그인한다.	
Post conditions	재고 관리가 잘 되고 있다.	
Flow of activities	Actor	System
	1.관리자는 재고 관리창을 보며 재고를 체크한다.	1.현재 재고에 대한 정보를 띄워준다.
Exception conditions		

Use case Name	재고 물건 추가하기	
Scenario	새로 입고된 물건을 재고에 추가한다.	
Triggering event	새로운 물건이 들어왔다.	
Brief description	관리자는 새로 들어온 물건에 대한 재고를 추가한다.	
Actors	관리자	
Related use cases	재고 관리	
Stakeholders	편의점 사장님, 택배기사님	
Preconditions	새로 들어온 물건이 하나 이상 존재한다.	

Post conditions	정상적으로 재고가 추가되었다.	
Flow of activities	Actor	System
	1.관리자는 재고 추가 버튼을 누른다. 3.관리자는 물품에 대한 정보를 입력하고 저장한다.	2.추가하려는 물품에 대한 정보 입력창이 뜬다. 4. 성공적으로 재고가 저장된다.
Exception conditions	이미 존재하는 물품이면 해당 정보에 대한 알림을 띄운다.	

Use case Name	재고 수정하기	
Scenario	재고 물품에 대한 정보를 수정한다.	
Triggering event	재고 물품에 대해 수정할 게 생겼다.	
Brief description	관리자는 수정하려는 물품에 대해 검색한다. 나온 제품에 대해서 수정하기를 진행한다. 정보를 수정하고 저장한다.	
Actors	관리자	
Related use cases	재고 관리	
Stakeholders	편의점 사장님, 택배기사님	
Preconditions	수정하려는 물건이 하나 이상 존재한다.	
Post conditions	정상적으로 재고 물품이 수정되었다.	
Flow of activities	Actor	System
	1.관리자는 재고 물품을 검색한다 2.관리자는 물품에 대해 수정하기를 눌러 제품 정보를 수정한다	3.입력한 정보대로 재고 정보를 수정한다.
Exception conditions	존재하지 않는 물품이면 존재하지 않는다고 메시지를 띄운다.	

Use case Name	재고 삭제하기
Scenario	재고 물품에 대한 정보를 삭제한다.
Triggering event	재고 물품을 삭제하려 한다.
Brief description	관리자는 수정하려는 물품에 대해 검색한다. 나온 제품에 대해 삭제를 한다.

Actors	관리자	
Related use cases	재고 관리	
Stakeholders	편의점 사장님, 택배기사님	
Preconditions	삭제하려는 물건이 하나 이상 존재한다.	
Post conditions	정상적으로 재고 물품이 삭제되었다.	
Flow of activities	Actor	System
	1.관리자는 재고 물품을 검색한다 2.관리자는 물품에 대해 삭제하기를 눌러 제품을 삭제한다.	2.재고 물품을 삭제한다.
Exception conditions	존재하지 않는 물품이면 존재하지 않는다고 메시지를 띄운다.	

Use case Name	로그인	
Scenario	포스기를 사용하기 위해서 로그인을 한다.	
Triggering event	프로그램 실행을 한다.	
Brief description	프로그램을 실행한 후 ID와 PW를 입력한다. 로그인이 정상적으로 됐다.	
Actors	캐셔, 관리자	
Related use cases		
Stakeholders	편의점 사장님	
Preconditions	프로그램을 정상적으로 실행시켰다.	
Post conditions	입력한 정보로 로그인을 성공했다.	
Flow of activities	Actor	System
	1.사용자는 프로그램을 실행시킨다 3.로그인 정보를 입력하고 로그인 한다.	1.프로그램이 실행된다 2.로그인 창을 출력한다. 4.입력 정보를 확인하고 로그인을 시도한다. 5.성공적으로 로그인 되었다.
Exception conditions	존재하지 않는 정보면 존재하지 않는다고 메시지를 띄운다.	

2.3) Requirement 도출

- 분석 설계 대상 시스템 명: POST System

기능적인 요구 사항(FR)	내용	중요도
FR1: 입력한 상품 표시	캐셔가 소비자가 구매하려는 물건들을 입력하면 입력된 물건들이 시스템에 표시된다.	★★★★★
FR2: 입력한 상품 취소	캐셔가 잘못 입력된 상품을 지운다.	★★★★☆
FR3: 입력한 상품 개수 선택	캐셔가 입력된 상품의 개수를 수정할 수 있다.	★★★☆☆
FR4: 총액 표시	시스템에서 입력된 물건들의 총액을 표시한다.	★★★★★
FR5: TAX 계산 (옵션)	시스템에서 Tax를 추가적으로 계산할지 물어보고 yes면 총액에 tax를 계산해 추가한다.	★★★☆☆
FR6: 할인 (옵션)	할인 받을 수 있는 수단이 있다면 해당 할인을 기록하고 총액에서 할인 금액을 계산한다.	★★★☆☆
FR7: 포인트 적립 (옵션)	소비자의 ID가 존재한다면 총액에서 적립할 수 있는 포인트를 계산해 포인트를 적립한다.	★★★☆☆
FR8: 결제 방법 선택	캐셔가 결제를 카드로 진행할지 현금으로 진행할 지 선택한다.	★★★★★
FR9: 카드 결제	시스템에서 카드로 결제를 진행한다	★★★★★
FR10: 현금 결제	시스템에서 현금으로 결제를 진행할 때는 거스름돈을 계산해준다.	★★★★★
FR11: 결제 완료	결제가 완료되면 시스템에서 자동적으로 재고 수정을 하고 결제 정보에 대해 기록하고 영수증을 출력한다.	★★★★★
FR12: 재고 물건 추가	관리자는 새로 들어온 물건에 대해서 재고를 추가할 수 있다.	★★★★★
FR13: 재고 수정	관리자는 물건 재고를 수정할 수 있다.	★★★★★
FR14: 재고 삭제	관리자는 물건 재고를 삭제할 수 있다.	★★★★★
FR15: 환불	영수증에 적힌 uuid나 사용자의 ID로 로그를 조회해서 환불을 할 수 있다. 환불은 결제 정보를 취소하고 재고 수정을 진행한다. 환불이 된 결과를 저장하고 영수증을 출력한다.	★★★★☆
FR16: 초기화	입력된 정보가 있거나 결제를 진행하기 전 초기화 버튼을 눌러서 입력정보를 삭제한다.	★★★★☆
FR17: 시스템 로그인	ID와 Password를 통해 로그인을 한다.	★★★★☆
FR18: 오류 로그 출력	오류가 발생하면 오류 발생 로그를 창으로 띄워준다.	★★★☆☆
FR19: 매출액 확인	관리자는 총매출액을 확인할 수 있다.	★★★☆☆

비기능적인 요구 사항 (NFR)	내용	품질 속성(Quality Attributes)과 내용	중요도
NFR1	관리자가 아닌 사용자는 security를 고려하여 기록된 로그를 열람할 수 없다.	Functional 관리자 정보를 확인한 후 기록을 열람할 수 있다.	★★★★★
NFR2	시스템의 operations은 차례대로 적절하게 진행하고 기능을 제대로 수행한다.	Functional 순서에 맞게 일을 진행하고 진행되는 일을 정확하게 수행한다.	★★★★★
NFR3	사용자가 문제없이 시스템을 사용할 수 있도록 시스템의 사용성을 고려하여 깔끔하게 화면에 출력한다.	Usability 사용자가 한눈에 이해하고 사용할 수 있도록 사용자에게 필요한 내용만 화면에 표시한다.	★★★★☆
NFR4	시스템에서 오류가 발생했을 때는 즉각적으로 handling 할 수 있도록 사용자에게 알리고 대처 방법을 선택할 수 있게 한다.	Reliability 오류가 발생하자마자 5초 안에 사용자에게 발생한 오류를 표시하고 만약 대처할 수 있는 방법이 존재하면 함께 표시한다.	★★★★★
NFR5	시스템에서의 모든 operation은 구동될 때 성능을 고려하여 빠르게 작동한다.	Performance 5초 안에 결과를 사용자에게 제공한다.	★★★★☆
NFR6	시스템에서 계산 완료 후에 사용된 리소스를 체크해보고 정상 작동될 수 있도록 한다.	Performance 사용가능 용량이 10%만 남아있다면 사용자에게 notice를 한다.	★★★★☆
NFR7	해당 시스템을 문제없이 setting 할 수 있도록 설치 매뉴얼을 지원한다.	Supportability 시스템 설치 셋팅 매뉴얼을 제공한다.	★★★★☆
NFR8	사용자가 입력하는 정보를 화면에서 바로바로 확인할 수 있도록 한다.	Performance 입력하는 텍스트가 딜레이 없이 보여지도록 한다.	★★★☆☆
NFR9	추가적으로 필요한 기능이 있다면 메일을 통해 제작자에게 문의할 수 있다.	Supportability Github나 메일 주소를 통해서 추가 기능에 대한 요구를 응할 수 있도록 한다.	★★★☆☆

(FURPS Model)

Scenario

Main Success Scenario:

시나리오 (Scenario)		자세한 시나리오 내용	
		Actor Action (or Intention)	System Responsibility
FR17, NFR8	NFR2, NFR3, NFR4, NFR5	0.캐셔로 시스템에 로그인한다.	0.정보를 확인하고 로그인한다.
		1.소비자가 구매하려는 물건들을 올려놓는다.	
FR16		2.캐셔는 결제 준비를 한다.	2. 화면을 초기화한다.
FR1, NFR8		3.캐셔는 소비자가 올려놓은 물건을 하나씩 입력한다.	
FR1, FR4		캐셔는 모든 물건을 입력 완료할 때까지 위의 행동을 반복한다.	4.화면에 입력되는 물건들에 대한 정보를 표시한다.
FR4			5.총액을 표시한다.
FR8		6.캐셔는 소비자에게 총액과 할인이나 포인트를 적립할 건지, 그리고 결제 방법에 대해 물어본다.	6. 결제 방법 선택 창을 띄운다. (옵션 사항들인 tax 추가, 포인트 적립, 할인 버튼을 함께 띄운다)
		7.소비자는 옵션을 선택하지 않고 결제수단을 건넨다.	
FR9 FR10		8.결제를 진행한다.	8.결제를 한다.
FR11			9.결제에 대한 정보를 기록하고 판매된 물품에 한에 재고 수정을 한다. 영수증을 출력한다.
FR16		10.소비자는 물품과 영수증을 챙겨간다.	10. 화면을 초기화 한다.

Extensions (Alternative Flows):

만약 메인 시나리오 6. 다음으로 소비자가 포인트 적립 후 결제를 하려한다면,

시나리오 (Scenario)		자세한 시나리오 내용	
		Actor Action (or Intention)	System Responsibility
	NFR2, NFR3,	7. 포인트 적립을 하기 위한 ID를 알려 준다.	
FR7, NFR8	NFR4, NFR5	8. 포인트 적립 버튼을 누르고 정보를 입력한 후 저장한다.	8. 입력된 ID에 포인트를 계산 해서 적립한다.
FR8		9. 소비자는 결제수단을 건넨다.	9. 결제방법 선택창으로 돌아 온다.
FR9 FR10		10. 결제를 진행한다.	10. 결제를 한다.
FR11			11. 결제에 대한 정보를 기록하고 판매된 물품에 한에 재고 수정을 한다. 영수증을 출력 한다.
FR16		12. 소비자는 물품과 영수증을 챙겨 간 다.	12. 화면을 초기화 한다.

만약 메인 시나리오 6. 다음으로 Tax를 포함하고 소비자의 할인 수단 계산 후 결제를 하려한다면,

시나리오 (Scenario)		자세한 시나리오 내용	
		Actor Action (or Intention)	System Responsibility
FR4, FR5	NFR2, NFR3,	7. 캐시는 tax 버튼을 누른다.	7. Tax 계산을 한 후 총액에 포함한다.
	NFR4,	8. 소비자는 할인 수단을 건넨다.	
FR4, FR6, NFR8	NFR5	9. 캐시는 할인 버튼을 누르고 정보를 입력한다.	9. 할인된 가격을 계산하고 표시한다.
FR4, FR8		10. 소비자는 할인된 금액을 보고 결제 수단을 건넨다.	10. 결제방법 선택창으로 돌아 온다.
FR9 FR10		11. 결제를 진행한다.	11. 결제를 한다.
FR11			12. 결제에 대한 정보를 기록하고 판매된 물품에 한에 재고 수정을 한다. 영수증을 출력 한다.
FR16		13. 소비자는 물품과 영수증을 챙겨 간 다.	13. 화면을 초기화 한다.

관리자 계정으로 들어가 재고 관리를 하려한다면,

시나리오 (Scenario)		자세한 시나리오 내용	
		Actor Action (or Intention)	System Responsibility
NFR1, FR17	NFR2, NFR3, NFR4, NFR5	1.관리자 ID로 로그인한다.	
		2. 재고를 확인한다.	2. 재고 화면을 띄운다.
FR1, NFR8		3.새로 들어온 물건을 추가하기 위해 추가 버튼을 누르고 정보를 입력한다.	
FR1, FR12		관리자는 모든 물건을 입력 완료할 때 까지 위의 행동을 반복한다.	4.입력된 물건을 추가한다.
FR13, NFR8		5. 재고 수량을 수정할 물건을 누르고 수량을 수정한다.	5.수정한 물건에 대한 정보를 저장한다.
		6. 필요없는 물건을 선택하고 삭제 버 튼을 누른다.	
FR14			7. 시스템에서 해당 물건 정보 를 삭제한다.
		8. 로그아웃한다.	

시스템 사용 중에 갑작스럽게 오류가 발생한다면,

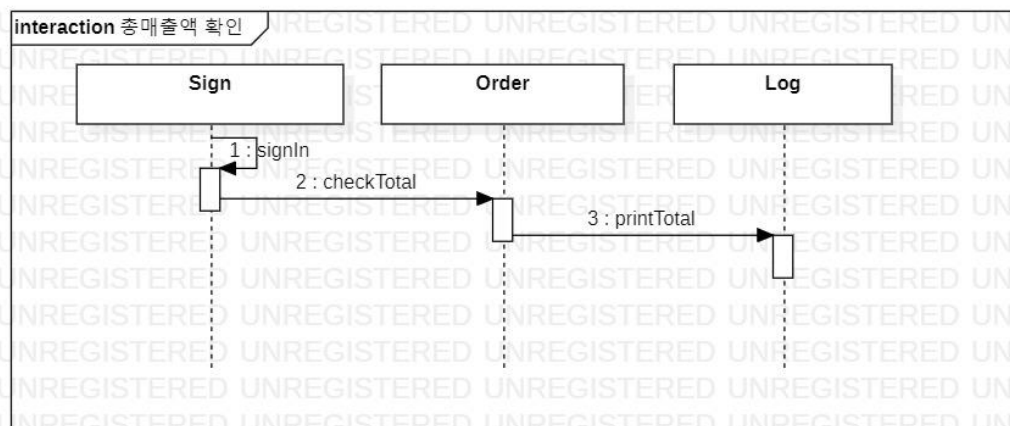
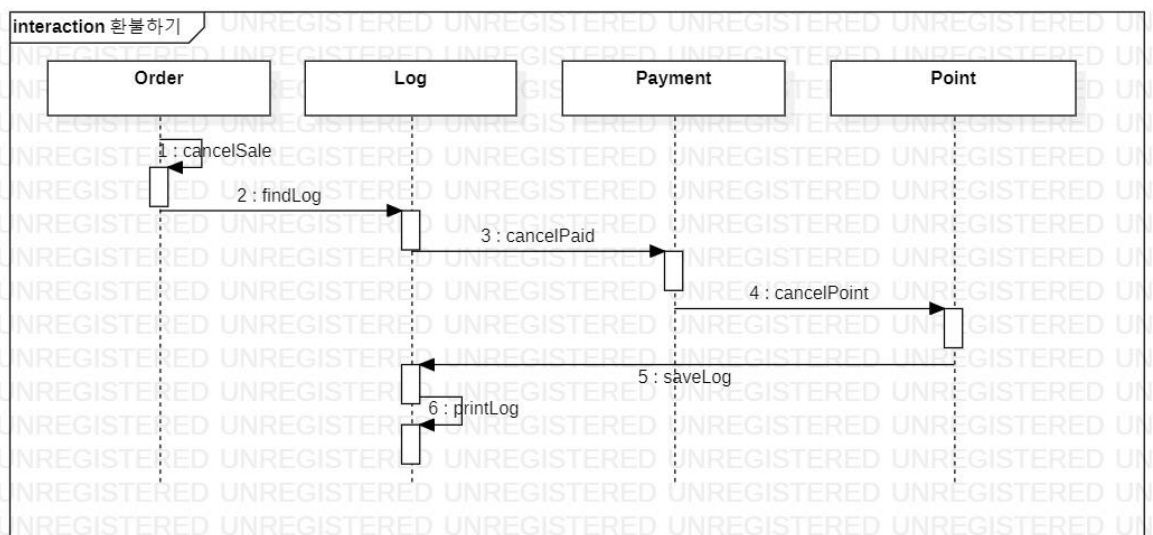
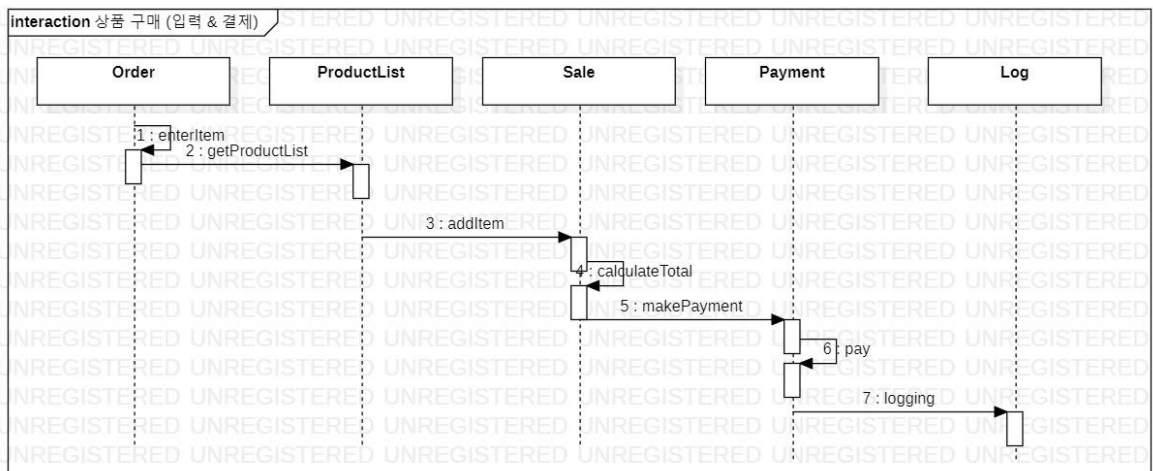
시나리오 (Scenario)		자세한 시나리오 내용	
		Actor Action (or Intention)	System Responsibility
FR18	NFR2,		0.발생한 오류 로그를 띄운다.
	NFR3,	1.시스템을 재부팅한다.	
FR16	NFR4,		2.초기화된 화면을 출력한다.
	NFR5	3.결제를 처음부터 다시 진행한다.	

3. Design Refinement by GRASP principle

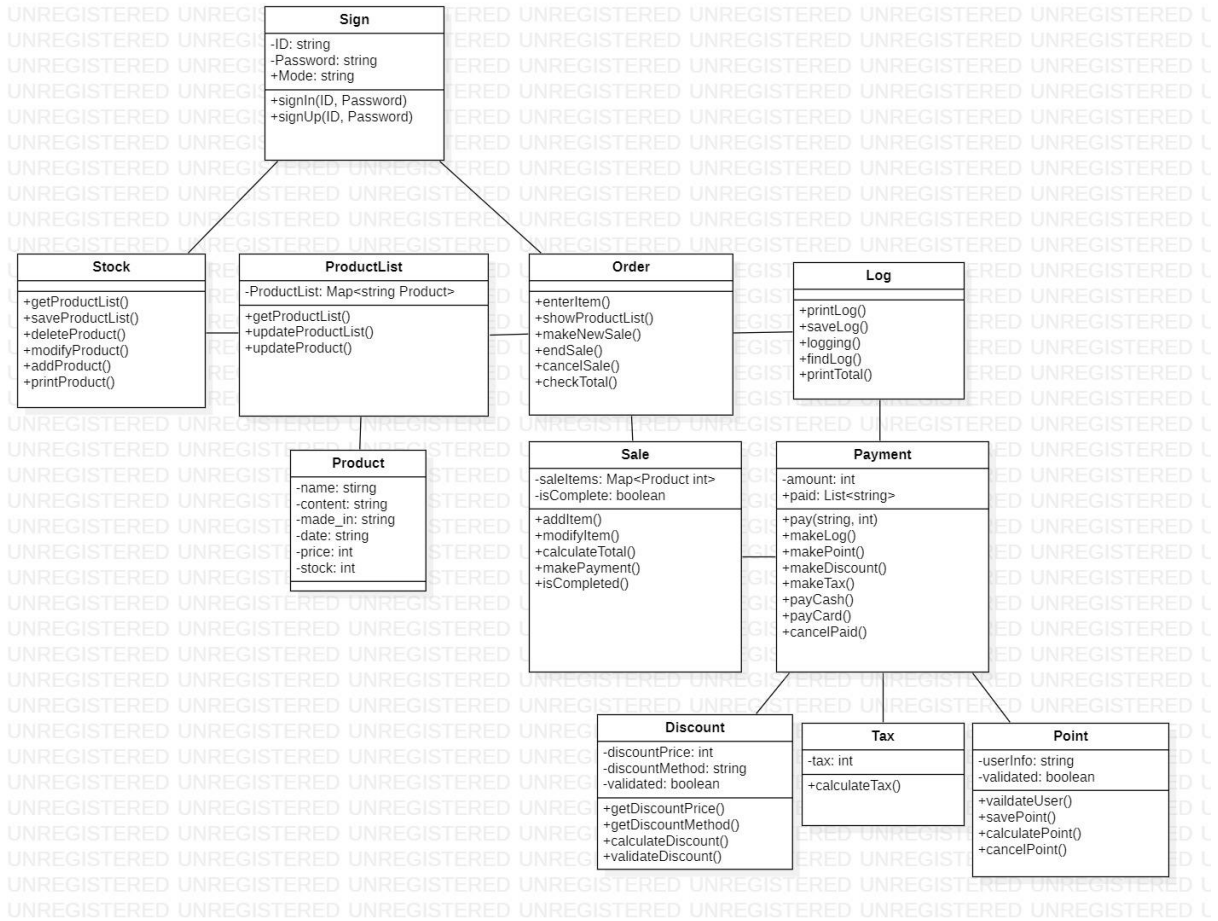
Design Refinement Specification

Before 단계의 Class 등 설계 Classifier	After 단계의 Class 등 설계 Classifier(s)	적용한 설계 개념 (GRASP에서 적용함)	Architecture Design Rationale (본인이 생각하는 합리성)	기타
SelectFunc	Order	Controller	POST System에서 캐셔가 사용할 수 있는 기능들을 모아놓은 클래스. 어떤 기능을 요청하면 그 기능을 수행할 수 있는 다른 오브젝트에게 포워드 해주는 역할을 수행함.	순서/주문하다/부탁하다는 의미의 order를 클래스 이름으로 정함.
Ordering	Sale	High Cohesion	상품 입력에 중점을 맞춤.	그 외의 상관없는 기능들은 다른 클래스로
Payment CashPayment CardPayment	Payment	Information Expert	Payment 클래스에서는 결제에 관련된 정보가 있고 굳이 sub class가 필요하지 않다는 판단이 들어 Payment 클래스로만 둠.	결제 방법만 다르다
	Log	High Cohesion Low Coupling	결제 완료 후 결제 클래스에서 기록을 자동적으로 하도록 구성했었음. 하지만 기록을 따로 핸들링할 수 있도록 기록을 따로 분리함.	
	Sign	Low Coupling	UI 클래스를 생각해서 UI에 합쳐있었지만 sign-in/up 기능을 하는 클래스를 만들어 분리함.	
StockUI	Stock	Low Coupling	UI를 제외한 기능만을 따로 분리해서 재고 관리를 담당하는 클래스를 만듦.	
ItemList	ProductList	Information Expert	가격 검색, 재고 검색, 계산 후 자동 재고 업데이트 등 현재고 상태에 대해 쓰일 일이 많고 동기적으로 관리되어야 하기 때문에 싱글톤으로 구성함.	싱글톤

4. Interaction Diagram



5. Class Diagram 그리기



6. The Results

```
----- 로그인 -----  
id: test  
password: asdf  
test: not exits  
  
----- 로그인 -----  
id: jeunna  
password: qwer  
jeunna: wrong password  
  
----- 로그인 -----  
id: jeunna  
password: q1w2e3  
  
-***- cashier -***-  
  
★      편의점      ★  
*****  
0. 메뉴 다시 보기  
1. 새결제하기  
2. 재고 체크하기  
3. 종료  
  
[OPTIONAL]  
4. 총매출 확인  
*****  
  
>>
```

- 로그인하기 (Sign)

파일 안에 저장되어 있는 정보로 로그인을 한다.

정보가 일치하지 않으면 로그인이 되지 않으며 캐셔와 관리자로 로그인을 할 수 있다.

정상적으로 로그인이 되면 로그인한 정보에 맞는 화면이 출력된다.

```
*****
0. 메뉴 다시 보기
1. 상품 입력 (물건이름, ±수량)
2. 상품 취소 (물건이름)
3. 입력된 상품 확인
4. 결제하기
5. 결제 종료

[OPTIONAL]
6. (결제전) 총액에 tax 계산 (x%)
*****

>> 1
추가하기: 상품 이름, 갯수
배추 1

-----
ITEM    NUM    PRICE  TOTAL
배추    1      5000   5000

total: 5000
-----

>>
```

- 상품 입력하기

손님이 가져온 가져온 상품을 입력할 수 있다.

```
-----
ITEM    NUM    PRICE  TOTAL
배추    1      5000   5000

total: 5000
-----

>> 1
추가하기: 상품 이름, 갯수
배추 3

-----
ITEM    NUM    PRICE  TOTAL
배추    4      5000  20000

total: 20000
-----

>> 1
추가하기: 상품 이름, 갯수
배추 -1

-----
ITEM    NUM    PRICE  TOTAL
배추    3      5000  15000

total: 15000
-----
```

```
추가하기: 상품 이름, 갯수
사탕 1

-----
ITEM    NUM    PRICE  TOTAL
배추    3      5000  15000
사탕    1      300    300

total: 15300
-----

>> 2
삭제하기: 상품 이름
사탕

-----
ITEM    NUM    PRICE  TOTAL
배추    3      5000  15000

total: 15000
-----
```

입력한 상품을 삭제하거나 개수를 수정할 수 있다.

```

total: 15000
-----
>> 0
*-*-*-*-*
0. 메뉴 다시 보기
1. 상품 입력 (물건이름 ±수량)
2. 상품 취소 (물건이름)
3. 입력된 상품 확인
4. 결제하기
5. 결제 종료

[OPTIONAL]
6. (결제전) 총액에 tax 계산 (x%)
*-*-*-*-*

>> 6
추가할 tax(%) : 3.3
15000 -> 15495
>>

```

- 세금 (tax) 계산

총액에 세금을 포함하기 위해 세금 계산을 할 수 있다.

```

결제해야할 금액: 15495
*-*-*-*-*
0. 메뉴 다시 보기
1. 카드결제 (금액)
2. 현금결제 (금액)
3. 종료

*-*-*-*-*

>> 1
카드결제할 금액 : 10000
남은 금액: 5495
>> 2
현금결제할 금액 : 5495
계산 완료!
updated: 배추's stock, 5 -> 2
=====

```

- 결제하기

결제는 카드/현금으로 분할 계산이 가능하다.

receipt.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Mon Jun 29 21:20:23 2020

ITEM	NUM	PRICE
배추	3	5000

total: 15495
(Tax 3.3% 포함)

결제방식: 카드(10000) 현금(5495)

- 계산 결과 기록 / Log

결제가 끝나면 자동적으로 기록을 저장한다.

	product_list.txt	Log.h	POS.cpp	Order.
1	배추	싱싱 한국산	2020/05 5000	2
2	별난바	커피맛 한국산	2020/08 500	2
3	빠빼코	초코맛 한국산	2020/05 500	5
4	사탕	딸기맛 한국산	2020/12 300	9

- 재고 갱신

결제가 끝나면 재고를 자동 갱신한다.


```

----- 로그인 -----
id: root
password: 1234

***- admin -***

★   편   의   점   ★ - 관리자 모드
*****
0. 메뉴 다시 보기
1. 전체 재고 조회
2. 상품 정보 조회 (물건이름)
3. 상품 추가 (물건이름 ±수량)
4. 상품 삭제 (물건이름)
5. 상품 정보 수정 (물건이름, 바꿀 항목, 바꿀 내용)
6. 재고 정보 저장
7. 종료

*****
>>

```

- 재고 관리

관리자 모드로 로그인시 재고관리를 할 수 있다.

```

>> 1
#####
ITEM    CONTENT  MADE IN DATE    PRICE  STOCK
배추    싱싱      한국산  2020/05  5000   2
별난바  커피맛     한국산  2020/08  500    2
빠빠코  초코맛     한국산  2020/05  500    5
사탕    딸기맛     한국산  2020/12  300    9

total: 4
#####

```

- 재고 확인하기

전체 재고를 확인할 수 있다.

```

>> 2
물건이름: 배추
#####
ITEM    CONTENT  MADE IN DATE    PRICE  STOCK
배추    싱싱      한국산  2020/05  5000   2
#####

```

- 물품 검색하기

원하는 상품에 대한 정보를 검색할 수 있다.

```

>> 3
입력하기: 상품 이름, 상품 설명, 제조한 곳, 유통기한, 가격, 갯수
상추 신선 한국산 2020/12 2500 10
added: 상추
>> 1
#####
ITEM    CONTENT  MADE IN DATE    PRICE  STOCK
배추    싱싱      한국산  2020/05 5000   2
별난바  커피맛    한국산  2020/08 500    2
빠빠코  초코맛    한국산  2020/05 500    5
사탕    딸기맛    한국산  2020/12 300    9
상추    신선      한국산  2020/12 2500   10

total: 5
#####

```

- 상품 추가

새로운 상품을 추가할 수 있다.

```

>> 4
삭제하기: 상품 이름
상추
정말 삭제하시겠습니까? (y/n)
y
deleted: 상추
>> 1
#####
ITEM    CONTENT  MADE IN DATE    PRICE  STOCK
배추    싱싱      한국산  2020/05 5000   2
별난바  커피맛    한국산  2020/08 500    2
빠빠코  초코맛    한국산  2020/05 500    5
사탕    딸기맛    한국산  2020/12 300    9

total: 4
#####

```

- 재고 삭제

재고 상품을 삭제할 수 있다.

```

>> 5
수정하기
상품 이름 : 배추
수정할 항목 : 이름:0, 내용:1, 제조한 곳:2, 유통기한:3, 가격:4, 갯수:5
5
수정할 내용 : 10
updated: 배추's stock, 2 -> 10
>> 1
#####
ITEM    CONTENT  MADE IN DATE    PRICE  STOCK
배추    싱싱      한국산   2020/05  5000   10
별난바  커피맛     한국산   2020/08  500    2
빠빠코  초코맛     한국산   2020/05  500    5
사탕    딸기맛     한국산   2020/12  300    9

total: 4
#####

```

- 재고 상품 내용 수정

상품 내용을 수정할 수 있다

```

product_list.txt  Log.h  POS.cpp  Order.h
1  배추 싱싱 한국산 2020/05 5000 10
2  별난바 커피맛 한국산 2020/08 500 2
3  빠빠코 초코맛 한국산 2020/05 500 5
4  사탕 딸기맛 한국산 2020/12 300 9
5

D:\workspace\Code\20153180_박정은_HW02\Debug\PO...
7. 종료

*-*-*-*-*
>> 6
저장되었습니다.
>> 1
#####
ITEM    CONTENT  MADE IN DATE    PRICE  STOCK
배추    싱싱      한국산   2020/05  5000   10
별난바  커피맛     한국산   2020/08  500    2
빠빠코  초코맛     한국산   2020/05  500    5
사탕    딸기맛     한국산   2020/12  300    9

total: 4
#####

```

- 변경된 사항 저장하기

변경된 재고 내용은 저장된다.