

# Use Case Description

학식 POS 시스템 / 나자바바라 (강길웅, 권민수, 맹산하, 정준권, 홍진욱)

<b>Use Case Name</b>	메뉴 갱신 / UC_01	
<b>Scenario</b>	식단 데이터베이스에서 해당 날짜에 맞는 식단을 읽어 갱신한다.	
<b>Triggering Event</b>	관리자가 메뉴 갱신을 요청했을 때.	
<b>Brief Description</b>	관리자가 업데이트 버튼을 누르면, 시스템이 관리자의 ID/Password를 요청한다. 관리자가 ID/Password를 입력하면 시스템이 이를 확인하고, 식단 DB에서 날짜에 맞는 식단 정보를 받아와 메뉴 리스트를 갱신한다.	
<b>Actors</b>	관리자	
<b>Related Use Cases</b>	None	
<b>Stakeholders</b>	관리자, 생활협동조합	
<b>Pre-Conditions</b>	식단 데이터베이스에 식단정보가 저장되어 있다.	
<b>Post-Conditions</b>	현재 날짜에 맞게 메뉴 갱신이 완료된다.	
<b>Flow Of Activities</b>	Actor	System
	1. 관리자가 업데이트 버튼을 누른다. 2. 관리자가 ID와 Password를 입력한다 .	1.1. 시스템이 관리자에게 ID와 Password를 요구한다. 2.1. 시스템이 관리자가 입력한 ID와 Password가 맞는지 확인한다. 2.2. 시스템이 식단 DB에서 식단 정보를 받아온다. 2.3. 시스템이 기존 메뉴 리스트에 있던 메뉴들을 삭제하고, 새로 받아온 식단의 메뉴들을 추가한다.
<b>Exception Condition</b>	2.1 관리자가 입력한 ID/Password가 틀린 경우, 오류 메시지를 출력하고 2.1로 돌아간다. 2.2 식단 DB로부터 식단 정보를 정상적으로 받아오지 못했을 경우, 오류 메시지를 출력한다.	

<b>Use Case Name</b>	장바구니 작성 / UC_02		
<b>Scenario</b>	사용자가 원하는 메뉴(들)로 장바구니를 작성한다.		
<b>Triggering Event</b>	사용자가 자신이 원하는 메뉴들을 선택해 식권을 받고자 할 때.		
<b>Brief Description</b>	사용자가 장바구니에 메뉴를 추가/삭제하고, 시스템이 장바구니의 변동사항에 따라 가격총합을 갱신하여 표시한다. 사용자가 '+','-' 버튼을 이용해 메뉴의 수량 조절을 할 수 있다.		
<b>Actors</b>	사용자 (소비자)		
<b>Related Use Cases</b>	장바구니에 메뉴 추가 include, 장바구니에서 메뉴 삭제 include		
<b>Stakeholders</b>	사용자 (소비자)		
<b>Pre-Conditions</b>	현재 메뉴 리스트가 관리자에 의해 갱신되어 정상적으로 상호작용 할 수 있는 상태.		
<b>Post-Conditions</b>	장바구니에 원하는 만큼 메뉴를 추가/제거 한 상태.		
<b>Flow Of Activities</b>	Actor	System	
	1. 사용자가 원하는 메뉴를 장바구니에 추가(UC_03)한다.	1.1	시스템이 장바구니에 추가된 메뉴의 가격만큼 현재 가격총합에 더하고 표시한다.
	2. 사용자가 제거할 메뉴를 장바구니에서 삭제(UC_04) 한다.	2.1	시스템이 장바구니에서 제거된 메뉴의 가격*수량만큼 현재 가격총합에서 빼고 표시한다.
	3. 사용자가 장바구니에 있는 메뉴의 수량을 '+','-' 버튼으로 조절한다.	3.1	시스템이 해당 메뉴의 수량을 '+'가 눌렸을 때는 +1, '-'가 눌렸을 때는 -1만큼 변동한다.
<b>Exception Condition</b>	3.1 해당 메뉴의 수량이 0이 됐을 경우, 장바구니 메뉴 리스트에서 해당 메뉴를 삭제한다.		

<b>Use Case Name</b>	장바구니에 메뉴 추가 / UC_03					
<b>Scenario</b>	사용자가 원하는 메뉴를 선택하면 장바구니에 추가된다.					
<b>Triggering Event</b>	사용자가 메뉴를 선택한다.					
<b>Brief Description</b>	사용자가 오늘의 식단 메뉴 중 추가할 메뉴를 선택하면, 시스템이 선택한 메뉴를 장바구니에 추가하고 기존 장바구니 가격총합에 추가한 메뉴의 가격을 더해 가격총합을 갱신한다.					
<b>Actors</b>	사용자 (소비자)					
<b>Related Use Cases</b>	장바구니 작성에 include					
<b>Stakeholders</b>	사용자 (소비자)					
<b>Pre-Conditions</b>	현재 메뉴가 관리자에 의해 갱신되어 정상적으로 상호작용 할 수 있는 상태.					
<b>Post-Conditions</b>	장바구니에 선택한 메뉴가 정상적으로 담긴다.					
<b>Flow Of Activities</b>	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. 현재 식단에서 사용자가 원하는 메뉴를 선택한다.</td> <td>           1.1. 시스템이 사용자가 선택한 메뉴의 ID를 받아온다            1.2. 시스템이 해당 Product가 있는지 확인하고, 같은 내용의 새로운 Product 객체를 생성한다            1.3. 시스템이 메뉴 객체를 장바구니에 추가한다.         </td> </tr> </tbody> </table>	Actor	System	1. 현재 식단에서 사용자가 원하는 메뉴를 선택한다.	1.1. 시스템이 사용자가 선택한 메뉴의 ID를 받아온다 1.2. 시스템이 해당 Product가 있는지 확인하고, 같은 내용의 새로운 Product 객체를 생성한다 1.3. 시스템이 메뉴 객체를 장바구니에 추가한다.	
Actor	System					
1. 현재 식단에서 사용자가 원하는 메뉴를 선택한다.	1.1. 시스템이 사용자가 선택한 메뉴의 ID를 받아온다 1.2. 시스템이 해당 Product가 있는지 확인하고, 같은 내용의 새로운 Product 객체를 생성한다 1.3. 시스템이 메뉴 객체를 장바구니에 추가한다.					
<b>Exception Condition</b>	1.3. 장바구니에 해당 메뉴가 이미 있는 경우, 해당 메뉴의 주문 수량을 1만큼 늘리고, 새로 만든 객체는 제거한다.					

<b>Use Case Name</b>	장바구니에서 메뉴 삭제 / UC_04	
<b>Scenario</b>	장바구니에 담긴 메뉴 중 삭제를 원하는 메뉴를 삭제한다.	
<b>Triggering Event</b>	장바구니에서 삭제할 메뉴를 선택한다.	
<b>Brief Description</b>	사용자가 장바구니에서 삭제하고자 하는 메뉴의 삭제 버튼을 누른다. 시스템이 이 메뉴를 장바구니에서 삭제하고, 삭제한 메뉴의 가격을 수량에 곱해 가격총합을 갱신한다.	
<b>Actors</b>	사용자	
<b>Related Use Cases</b>	장바구니 작성에 include	
<b>Stakeholders</b>	사용자	
<b>Pre-Conditions</b>	장바구니에 선택한 메뉴가 담겨있다.	
<b>Post-Conditions</b>	삭제를 원하는 메뉴가 장바구니에서 삭제된다.	
<b>Flow Of Activities</b>	Actor	System
	1. 사용자가 장바구니에서 삭제하고자 하는 메뉴의 삭제 버튼을 누른다.	1.1. 시스템이 사용자가 선택한 메뉴의 ID를 가져온다. 1.2. 장바구니에서 해당 ID를 가진 객체를 삭제한다.
<b>Exception Condition</b>		

<b>Use Case Name</b>	결제 / UC_05							
<b>Scenario</b>	결제가 진행된다.							
<b>Triggering Event</b>	사용자가 결제를 요청했다.							
<b>Brief Description</b>	사용자가 결제 요청을 하면, 시스템이 결제 금액을 표시한다. 사용자가 결제를 진행하면, 시스템이 이를 확인한 뒤, 장바구니의 메뉴 리스트를 바탕으로 식권과 영수증을 출력하고, 결제정보를 매출정보 DB에 업데이트한다.							
<b>Actors</b>	사용자							
<b>Related Use Cases</b>	현금결제(UC_06), 카드결제(UC_07)를 일반화							
<b>Stakeholders</b>	사용자, 생활협동조합							
<b>Pre-Conditions</b>	사용자가 식권을 받고자 하는 메뉴가 장바구니에 담겨있다.							
<b>Post-Conditions</b>	결제가 정상적으로 완료되고, 식권과 영수증이 발급된다.							
<b>Flow Of Activities</b>	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. 사용자가 결제를 요청한다.</td> <td>           1.1 시스템이 현재 장바구니의 메뉴 리스트로부터 총합 금액을 받아온다.            1.2 시스템이 사용자에게 총합금액만큼의 결제를 요구한다.         </td> </tr> <tr> <td>2. 사용자가 결제를 진행한다.</td> <td>           2.1 시스템이 결제 유효성을 확인한다.            2.2 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.         </td> </tr> </tbody> </table>	Actor	System	1. 사용자가 결제를 요청한다.	1.1 시스템이 현재 장바구니의 메뉴 리스트로부터 총합 금액을 받아온다. 1.2 시스템이 사용자에게 총합금액만큼의 결제를 요구한다.	2. 사용자가 결제를 진행한다.	2.1 시스템이 결제 유효성을 확인한다. 2.2 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.	
Actor	System							
1. 사용자가 결제를 요청한다.	1.1 시스템이 현재 장바구니의 메뉴 리스트로부터 총합 금액을 받아온다. 1.2 시스템이 사용자에게 총합금액만큼의 결제를 요구한다.							
2. 사용자가 결제를 진행한다.	2.1 시스템이 결제 유효성을 확인한다. 2.2 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.							
<b>Exception Condition</b>	2.1 유효성 확인에 실패할 경우, 사용자에게 다시 결제를 진행 할 것을 요구하는 메시지를 출력한다.							

<b>Use Case Name</b>	현금 결제 / UC_06							
<b>Scenario</b>	현금을 통한 결제가 진행된다.							
<b>Triggering Event</b>	사용자가 결제를 현금으로 요청했다.							
<b>Brief Description</b>	사용자가 현금결제 버튼을 누르면, 시스템이 현재 장바구니의 내용을 바탕으로 입금요청을 한다. 사용자는 이에 맞는 금액을 넣고, 시스템은 이를 확인해 식권과 영수증을 출력하고, 매출정보 DB에 결제정보를 업데이트한다.							
<b>Actors</b>	사용자							
<b>Related Use Cases</b>	결제(UC_05)를 구체화							
<b>Stakeholders</b>	사용자, 생활협동조합							
<b>Pre-Conditions</b>	사용자가 식권을 받고자 하는 메뉴가 장바구니에 담겨있다. 현금결제를 선택했다.							
<b>Post-Conditions</b>	결제가 정상적으로 완료되고, 식권과 영수증이 발급된다.							
<b>Flow Of Activities</b>	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. 사용자가 현금결제 버튼을 누른다.</td> <td>1.1. 시스템이 장바구니의 메뉴들을 읽어온다. 1.2. 시스템이 장바구니에 있는 메뉴들의 가격총합을 계산한다. 1.3. 시스템이 가격총합만큼의 입금요청 메시지를 띄우고, 타이머를 시작한다.</td> </tr> <tr> <td>2. 사용자가 현금을 투입한다.</td> <td>2.1. 시스템이 사용자가 투입한 현금을 확인한다. 2.2. 시스템이 투입된 액수만큼 현재 총입금액 수에 더하고, 총입금액이 가격총합보다 큰지 확인한다. 2.3. 시스템이 총입금액과 가격총합의 차액만큼의 거스름돈을 반환한다. 2.4. 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.</td> </tr> </tbody> </table>	Actor	System	1. 사용자가 현금결제 버튼을 누른다.	1.1. 시스템이 장바구니의 메뉴들을 읽어온다. 1.2. 시스템이 장바구니에 있는 메뉴들의 가격총합을 계산한다. 1.3. 시스템이 가격총합만큼의 입금요청 메시지를 띄우고, 타이머를 시작한다.	2. 사용자가 현금을 투입한다.	2.1. 시스템이 사용자가 투입한 현금을 확인한다. 2.2. 시스템이 투입된 액수만큼 현재 총입금액 수에 더하고, 총입금액이 가격총합보다 큰지 확인한다. 2.3. 시스템이 총입금액과 가격총합의 차액만큼의 거스름돈을 반환한다. 2.4. 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.	
Actor	System							
1. 사용자가 현금결제 버튼을 누른다.	1.1. 시스템이 장바구니의 메뉴들을 읽어온다. 1.2. 시스템이 장바구니에 있는 메뉴들의 가격총합을 계산한다. 1.3. 시스템이 가격총합만큼의 입금요청 메시지를 띄우고, 타이머를 시작한다.							
2. 사용자가 현금을 투입한다.	2.1. 시스템이 사용자가 투입한 현금을 확인한다. 2.2. 시스템이 투입된 액수만큼 현재 총입금액 수에 더하고, 총입금액이 가격총합보다 큰지 확인한다. 2.3. 시스템이 총입금액과 가격총합의 차액만큼의 거스름돈을 반환한다. 2.4. 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.							
<b>Exception Condition</b>	1.3. 타이머가 끝날 때까지 사용자가 가격총합만큼의 현금을 투입하지 않을 경우, 사용자가 투입한 돈을 다시 반환하여 결제를 취소하고 메시지를 띄운 후 장바구니 작성단계로 돌아간다. 2.2. 총입금액이 가격총합보다 작을 경우, 계속해서 2.1로 돌아가 사용자가 현금을 투입하기를 기다린다. 2.3. 지급 할 거스름돈이 부족한 경우, 거스름돈이 부족하다는 에러 메시지를 띄운 후 직원을 호출한다. 2.4. 식권과 영수증을 출력할 용지가 부족한 경우, 용지가 부족하다는 에러 메시지를 띄운 후 직원을 호출한다.							

<b>Use Case Name</b>	카드 결제 / UC_07									
<b>Scenario</b>	카드를 통해 결제가 진행된다.									
<b>Triggering Event</b>	사용자가 카드를 통한 결제를 요청했다.									
<b>Brief Description</b>	사용자가 카드 결제 버튼을 누르면 시스템이 장바구니의 메뉴리스트를 읽어와 결제금액을 표시한다. 사용자가 카드리더기에 카드를 삽입하면, 시스템이 이를 인식해 카드 결제 승인 시스템에 결제금액 만큼의 결제 승인을 요청한다. 사용자가 카드를 회수하면, 시스템이 장바구니의 내용을 바탕으로 식권과 영수증을 출력하고, 결제정보를 매출정보 DB에 업데이트 한다.									
<b>Actors</b>	사용자									
<b>Related Use Cases</b>	결제(UC_05)를 구체화									
<b>Stakeholders</b>	사용자									
<b>Pre-Conditions</b>	사용자가 식권을 받고자 하는 메뉴가 장바구니에 담겨있다. 카드 결제를 선택했다.									
<b>Post-Conditions</b>	정상적으로 결제가 이루어지고 식권과 영수증이 발급된다.									
<b>Flow Of Activities</b>	<table border="1"> <thead> <tr> <th>Actor</th><th>System</th></tr> </thead> <tbody> <tr> <td>1. 사용자가 카드 결제 버튼을 누른다.</td><td>1.1. 시스템이 장바구니의 메뉴들을 읽어온다. 1.2. 시스템이 장바구니에 있는 메뉴들의 가격총합을 계산한다. 1.3. 시스템이 가격총합만큼의 카드 읽기 요청 메시지를 띄우고, 타이머를 시작한다.</td></tr> <tr> <td>2. 사용자가 카드를 카드리더기에 삽입한다.</td><td>2.1. 시스템이 카드 정보를 읽어온다. 2.2. 시스템이 카드 결제 승인 시스템에 결제금액 만큼의 결제승인을 요청한다.</td></tr> <tr> <td>3. 사용자가 카드를 카드리더기로부터 회수한다.</td><td>2.3. 시스템이 결제승인 결과를 받아오고, 사용자에게 카드 회수 요청 메시지를 띄운다. 3.1. 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.</td></tr> </tbody> </table>	Actor	System	1. 사용자가 카드 결제 버튼을 누른다.	1.1. 시스템이 장바구니의 메뉴들을 읽어온다. 1.2. 시스템이 장바구니에 있는 메뉴들의 가격총합을 계산한다. 1.3. 시스템이 가격총합만큼의 카드 읽기 요청 메시지를 띄우고, 타이머를 시작한다.	2. 사용자가 카드를 카드리더기에 삽입한다.	2.1. 시스템이 카드 정보를 읽어온다. 2.2. 시스템이 카드 결제 승인 시스템에 결제금액 만큼의 결제승인을 요청한다.	3. 사용자가 카드를 카드리더기로부터 회수한다.	2.3. 시스템이 결제승인 결과를 받아오고, 사용자에게 카드 회수 요청 메시지를 띄운다. 3.1. 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.	
Actor	System									
1. 사용자가 카드 결제 버튼을 누른다.	1.1. 시스템이 장바구니의 메뉴들을 읽어온다. 1.2. 시스템이 장바구니에 있는 메뉴들의 가격총합을 계산한다. 1.3. 시스템이 가격총합만큼의 카드 읽기 요청 메시지를 띄우고, 타이머를 시작한다.									
2. 사용자가 카드를 카드리더기에 삽입한다.	2.1. 시스템이 카드 정보를 읽어온다. 2.2. 시스템이 카드 결제 승인 시스템에 결제금액 만큼의 결제승인을 요청한다.									
3. 사용자가 카드를 카드리더기로부터 회수한다.	2.3. 시스템이 결제승인 결과를 받아오고, 사용자에게 카드 회수 요청 메시지를 띄운다. 3.1. 시스템이 장바구니 메뉴들의 식권과 결제 영수증을 출력한다.									
<b>Exception Condition</b>	1.3. 타이머가 끝날 때까지 사용자가 카드를 삽입하지 않을 경우, 에러메세지를 띄운 뒤 장바구니 작성단계로 돌아간다. 2.1. 읽어온 카드 정보가 비정상적일 경우, 에러메시지를 출력하고 다시 카드 읽기 요청 메시지를 띄운다. 2.3. 카드의 잔액 부족등의 문제로 인해 결제가 거절 된 경우 에러메시지를 출력하고, 장바구니 작성단계로 돌아간다. 3.1. 식권과 영수증을 출력할 용지가 부족한 경우, 용지가 부족하다는 에러 메시지를 띄운 후 직원을 호출한다.									

<b>Use Case Name</b>	매출 정산 / UC_08					
<b>Scenario</b>	관리자가 매출내역 확인을 통해 정산을 진행한다.					
<b>Triggering Event</b>	관리자가 정산을 원할 때					
<b>Brief Description</b>	관리자가 정산 버튼을 누르면 시스템이 ID/Password를 요청하고, 이를 입력하면 시스템이 확인 후 당일의 매출결과를 카드/현금으로 나누어 출력한다					
<b>Actors</b>	관리자					
<b>Related Use Cases</b>	None					
<b>Stakeholders</b>	관리자, 생활협동조합					
<b>Pre-Conditions</b>	매출정보 데이터베이스와 정상적으로 연결 된 상태. 결제 내역이 존재함.					
<b>Post-Conditions</b>	정산 시점까지의 당일 정산결과를 보여준다.					
<b>Flow Of Activities</b>	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. 관리자가 정산 버튼을 누른다. 2. 관리자가 ID/Password를 입력한다</td> <td>1.1. 시스템이 관리자에게 ID와 Password를 요구한다. 2.1. 시스템이 관리자가 입력한 ID와 Password가 맞는지 확인한다. 2.2. 현재까지 저장하고 있던 매출정보를 DB에 업데이트 한다. 2.3. 저장하고 있던 매출정보들을 제거한다.</td> </tr> </tbody> </table>	Actor	System	1. 관리자가 정산 버튼을 누른다. 2. 관리자가 ID/Password를 입력한다	1.1. 시스템이 관리자에게 ID와 Password를 요구한다. 2.1. 시스템이 관리자가 입력한 ID와 Password가 맞는지 확인한다. 2.2. 현재까지 저장하고 있던 매출정보를 DB에 업데이트 한다. 2.3. 저장하고 있던 매출정보들을 제거한다.	
Actor	System					
1. 관리자가 정산 버튼을 누른다. 2. 관리자가 ID/Password를 입력한다	1.1. 시스템이 관리자에게 ID와 Password를 요구한다. 2.1. 시스템이 관리자가 입력한 ID와 Password가 맞는지 확인한다. 2.2. 현재까지 저장하고 있던 매출정보를 DB에 업데이트 한다. 2.3. 저장하고 있던 매출정보들을 제거한다.					
<b>Exception Condition</b>	2.1 관리자가 입력한 ID/Password가 틀린 경우, 오류 메시지를 출력하고 1.1로 돌아간다.					