

[Schema for use-cases]

Use Case UC-1 : 회원인증	
Related Requirements :	ST-1
Initiating Actor :	회원, 비회원 (= 식당이용자)
Actor's Goal :	음식을 주문할 때 할인을 받기 위해
Participating Actors :	X
Preconditions :	<ul style="list-style-type: none"> 할인을 받기 위해서는 학생증 또는 교직원 카드가 있어야 한다.
Postconditions :	<ul style="list-style-type: none"> 인증이 완료되면 총 금액의 5 퍼센트가 할인된다.
Flow of Events for Main Success Scenario :	
→	1. 식당이용자는 사용자 인증을 하기 위해서 “회원인증”버튼을 클릭한다.
←	2. 시스템은 회원 인증을 받으면 결과를 알림창에 띄워준다.
→	3. 식당 이용자는 해당 알림창을 끌 수 있다.
←	4. 알림창이 꺼지면 메뉴 조회 화면으로 넘어간다.
Flow of Events for Extensions(Alternate Scenarios) :	
1a→	1. 회원 인증에 실패했다.
1a←	2. 실패했다는 알림창이 나온다.
4a→	1. 알림창을 끄지 않는다.
4a←	2. 5 초 뒤 자동으로 화면을 전환한다. 이때, 회원 인증에 성공하면 메뉴 선택 화면이, 실패 시 처음화면으로 전환한다.

Use Case UC-2 : 개수 조절 (sub-use case)	
Related Requirements :	ST-4
Initiating Actor :	회원, 비회원 (= 식당이용자)
Actor's Goal :	원하는 만큼의 주문을 하기 위함
Participating Actors :	개수 조절 버튼
Preconditions :	<ul style="list-style-type: none"> 메뉴를 선택해야한다.(UC-4)
Postconditions :	<ul style="list-style-type: none"> 장바구니 속 해당 음식의 개수가 1 씩 늘어나거나, 줄어든다.
Flow of Events for Main Success Scenario :	
→	1. 메뉴 조회(UC-3)을 포함한다.
→	2. 식당 이용자는 장바구니 속 음식의 목록 옆에 +/- 아이콘을 클릭한다.
←	3. + 버튼을 클릭하면 해당 음식의 개수가 하나 늘어나고, - 버튼을 클릭하면 하나 줄어든다.
Flow of Events for Extensions(Alternate Scenarios) :	
2a→	1. 음식의 개수가 1 인 음식의 -버튼을 클릭한다.
2a←	2. 해당 음식을 장바구니 속에서 삭제할 것인지를 물어보는 알림창을 띄운다.
2a→	1. 음식의 개수를 10 개 이상으로 만든다.
2a←	2. 시스템은 “더 이상 담을 수 없습니다”라는 경고 창을 띄우고, 해당 음식의 개수를 더 이상 늘려지지 않게 한다.

Use Case UC-3 : 메뉴 조회 (sub-use case)	
Related Requirements :	ST-2, ST-3, ST-5, ST-6
Initiating Actor :	회원, 비회원 (= 식당이용자)
Actor's Goal :	식당 별 메뉴 정보를 제공 받기 위해
Participating Actors :	메뉴 조회 버튼
Preconditions :	<ul style="list-style-type: none"> • 학식당 별로 날짜별 메뉴 데이터를 조회할 수 있어야한다.
Postconditions :	<ul style="list-style-type: none"> • 식당 이용자는 학식당 별로 메뉴 정보를 제공받을 수 있다.
Flow of Events for Main Success Scenario :	
→	1-1. 회원 인증을 포함한다.(UT-1)
→	1-2. 사용자는 곧바로 메뉴 선택 버튼을 누른다.
←	2. 시스템은 해당 날짜에 학식당 별로 메뉴 리스트를 띄워준다.
→	3. 식당 이용자는 식사를 원하는 학식당 데이터의 우측 하단에 “자세히”버튼을 누른다.
←	4. 시스템은 해당 학식당에서 현재 주문 가능한 음식 리스트를 보여준다.
Flow of Events for Extensions(Alternate Scenarios) :	
3a→	1. 원하지 않던 학식당을 클릭한다.
3a←	2. 왼쪽 상단에 뒤로가기 버튼을 클릭한다.
4a←	1. 데이터 조회 시 특정 메뉴의 잔여 개수가 0 이 될 경우 해당 메뉴의 정보란을 불러 처리한다.

Use Case UC-4 : 메뉴선택 (sub-use case)	
Related Requirements :	ST-4, ST-5, ST-6
Initiating Actor :	회원, 비회원 (= 식당이용자)
Actor's Goal :	현 시간에 주문가능한 메뉴를 선택하여 주문하기 위해
Participating Actors :	메뉴 선택 버튼
Preconditions :	<ul style="list-style-type: none"> • 현 시간에 주문가능한 메뉴가 있어야 한다.
Postconditions :	<ul style="list-style-type: none"> • 식당 이용자는 특정 메뉴를 선택할 때마다 장바구니에 메뉴가 하나씩 담기게 된다.
Flow of Events for Main Success Scenario :	
→	1. 메뉴 조회(UC-3)를 포함한다.
→	2. 식당이용자는 원하는 학식당에 접속하여 '주문하러가기'버튼을 클릭한다
←	3. 시스템은 데이터베이스로부터 선택한 학식당에 관한 데이터들을 추출하고 현재 주문 가능한 메뉴 리스트를 뽑기 위한 쿼리문을 실행시켜 식당 이용자에게 보여준다.
→	4. 식당이용자는 주문가능한 메뉴 리스트에서 각 메뉴 우측의 '담기' 버튼을 통해 특정 메뉴를 클릭한다.
←	5. 시스템은 식당이용자가 클릭한 메뉴를 장바구니에 담겨졌음을 알리기위해 장바구니 이모티콘 위에 +1 씩 더하여 담은 메뉴의 개수를 보여준다
→	6. 식당이용자는 장바구니 버튼을 클릭한다.
←	7. 시스템은 지금 담은 메뉴의 리스트가 담긴 장바구니 페이지를 보여준다.
→	8. 식당이용자는 담은 메뉴의 개수를 +,- 버튼을 누르며 조절할 수 있다 (UC-2)
←	9. 시스템은 주문 목록의 각 메뉴와 개수에 대한 데이터를 임시로 저장하여 메뉴구입-주문완료 기능에서 해당 데이터를 활용한다.

Flow of Events for Extensions(Alternate Scenarios) :	
2a→	1. 접속한 학식당에서 현재 주문가능한 메뉴가 없다
2a←	2. ‘주문하러가기’버튼은 비활성화 되어있으며, 그럼에도 사용자가 버튼을 클릭했을 시, “주문가능한 시간이 아니에요”라는 메시지를 띄운다.
4a→	1. 클릭한 메뉴의 재고가 없다(품절이다)
4a←	2. 품절인 메뉴의 ‘담기’버튼이 비활성화 되며 재고수량을 보여주는 칸에서도 ‘품절’표시를 강조하여 보여준다. 그럼에도 사용자가 버튼을 클릭했을 시, “해당 메뉴는 지금 품절입니다”라는 메시지를 띄운다.

Use Case UC-5 : 메뉴 구입 (sub-use case)	
Related Requirements :	ST-6, ST-7, ST-8, ST-9
Initiating Actor :	회원, 비회원
Actor's Goal :	선택한 모든 메뉴의 금액을 고객의 유형별로 계산하여 결제하기 위함
Participating Actors :	QR 코드
Preconditions :	<ul style="list-style-type: none"> 장바구니에 1 개 이상의 메뉴가 담겨져 있어야한다.
Postconditions :	<ul style="list-style-type: none"> 장바구니에 담긴 메뉴들과 각 메뉴의 개수를 고려하여 합산된 총금액, 식당이용자가 회원인지 비회원인지 구분하여 할인금액, 그리고 총금액과 할인금액을 뺀 총 결제금액을 보여준 뒤 카드결제를 진행한다.
Flow of Events for Main Success Scenario :	
→	1. 식당이용자는 장바구니페이지에서 메뉴 선택과 개수 조절을 모두 마친 후 (UC-4) ‘결제하러가기’을 클릭한다.
←	2. 시스템은 식당이용자가 어떤 유형인지 판단 후 (UC-1) 총 금액, 할인금액, 결제금액 등 결제정보를 상세하게 보여준다
→	3. 식당이용자는 ‘페이북 결제’을 클릭하고 실제 결제를 진행한다.
←	4. 시스템은 ISP 카드 결제를 진행시킨다. (페이북 등의 어플 이용)
→	5. 식당이용자는 ISP 결제가 가능한 어플을 통해 성공적으로 카드 결제를 완료 한 후 다시 어플로 돌아온다
←	6. 시스템은 완료된 주문의 주문 확인 페이지를 보여주고 이와 함께 음식수령을 위한 QR 코드를 생성한다.
→	7. 식당이용자는 주문한 학식당의 코너에 가서 식당담당자에게 QR 코드를 제시하면 음식을 받을 수 있다.
Flow of Events for Extensions(Alternate Scenarios) :	
1a→	1. 장바구니에 담겨진 메뉴가 0 개이다

1a←	2. 시스템은 “담겨진 메뉴가 없어요”라는 메시지와 함께 장바구니 페이지로 되돌아간다.
5a→	1. ISP 카드 결제 진행과정 중 오류가 생겼다.
5a←	2. 시스템은 “카드 결제가 성공적으로 완료되지 않았습니다”라는 메시지와 함께 장바구니 페이지로 리로딩된다.
7a→	1. 10 분 이내에 식당담당자에게 QR 코드를 제시하지 못했다.
7a←	2. 시스템은 접수되었던 주문을 취소시키고, 주문 결제 금액을 3 일 이내로 해당 카드의 계좌로 환불 처리한다.

Use Case UC-6 : 유저 추가 (sub-use case)	
Related Requirements :	ST-1
Initiating Actor :	회원 (식당 이용자)
Actor's Goal :	<ul style="list-style-type: none"> - 새로운 유저(회원)를 생성하고 회원 혜택을 제공하기 위해 - 단순 회원 인증에서 그치지 않고 중복되는 학생증 인증을 불허함으로써 효율적인 유저관리를 하기 위해
Participating Actors :	앱관리자
Preconditions :	<ul style="list-style-type: none"> • 회원 인증이(UC-1) 성공적으로 완료되어야 한다.
Postconditions :	<ul style="list-style-type: none"> • 성공적으로 유저가 생성되면 회원은 회원혜택을 제공받을 수 있다
Flow of Events for Main Success Scenario :	
→	1. 회원은 회원인증이 완료된 다음(UC-1), 본인임을 확인하는 행위로써 휴대폰 본인인증을 하기 위해 전화번호를 입력하고 ‘인증번호 받기’를 클릭한다..
←	2. 본인 확인 서버 시스템에서 일치여부를 확인한 후 SMS 로 4-6 자리 숫자로 된 인증번호를 보낸다.
→	3. 회원은 유효한 인증번호를 입력한 후 ‘유저 생성하기’버튼을 클릭한다.
←	4. 인증번호가 일치하면 시스템은 회원 데이터 베이스에 생성시킨 유저의 정보를 업데이트한다.
←	5. 시스템은 “ 성공적으로 가입되었습니다 “ 라는 메시지와 함께 학식당 리스트를 볼 수 있는 첫 화면을 보여준다.
Flow of Events for Extensions(Alternate Scenarios) :	
3a→	회원이 잘못된 인증번호를 입력한다.
3b←	시스템은 “유효하지 않은 인증번호입니다”라는 메시지와 함께 ‘인증문자 전송’버튼을 재활성화시킨다
4a→	이미 존재하는 회원이다.

4b←	시스템은 “이미 존재하는 회원입니다”라는 메시지와 함께 로그인 화면을 보여준다.
-----	--

Use Case UC-7 : 유저 삭제 (sub-use case)	
Related Requirements :	ST-1
Initiating Actor :	회원 (식당 이용자)
Actor's Goal :	존재하는 회원(유저) 계정을 삭제하기 위해
Participating Actors :	앱관리자
Preconditions :	<ul style="list-style-type: none"> 삭제할 유저의 계정이 있어야 한다. 유저의 계정으로 현재 로그인이 되어있어야 한다.
Postconditions :	<ul style="list-style-type: none"> 회원이었던 식당이용자의 모든 정보가 삭제되고, 앱의 최초화면으로 화면이 전환된다.
Flow of Events for Main Success Scenario :	
→	회원은 앱 내 ‘톱니바퀴 버튼’ - ‘계정삭제하기’ 버튼을 클릭한다.
←	시스템은 “계정 삭제 시 앱 내 모든 정보가 삭제됩니다”라는 메시지와 함께 팝업창(재확인창)을 띄운다.
→	회원은 ‘확인’버튼을 클릭한다.
←	시스템은 삭제하려는 회원과 관련된 모든 데이터베이스 내 정보들을 일괄 삭제한다.
←	시스템은 “성공적으로 유저가 삭제되었습니다”라는 메시지와 함께 앱 최초실행화면으로 돌아간다