



# Software Engineering

## Use Cases

### <Members>

김찬현 / 20172977

장동훈 / 20174353

안재형 / 20171248

이지호 / 20174266

김동욱 / 20173299

김상렬 / 20165020

# <Table of Contents>

## 1. Use Cases(1~11)

- 1) 회원가입 – UC1
- 2) 로그인 – UC2
- 3) 메뉴 보기 – UC3
- 4) 메뉴 상세 보기 – UC4
- 5) 장바구니 – UC5
- 6) 결제하기 – UC6
- 7) 유저 프로필 관리(프로필 수정) – UC7
- 8) 주문 관리 – UC8
- 9) 가게 정보 수정 – UC9
- 10) 음식수령 – UC10
- 11) 판매내역 확인 – UC11

## Use Case 1

<b>Use Case – UC1</b>	<b>회원가입</b>
<b>Related Requirements</b>	OFR-1, SFR-1
<b>Initiating Actor</b>	주문자
<b>Actor's Goal</b>	사이트에 회원가입을 한다.
<b>Participating Actors</b>	주문자, 점원, 유저DB
<b>Precondition</b>	<ul style="list-style-type: none"> <li>- 동일한 아이디를 사용하는 회원이 있어서는 안 된다.</li> <li>- 비밀번호는 4자리 이상 이어야 한다.</li> <li>- 전화번호가 있어야 한다.</li> </ul>
<b>Postcondition</b>	- 자동으로 로그인한 후 메인화면 으로 이동한다.
<b>Flow of Events for main Success Scenario</b>	
<p>→ 1. 주문자 또는 점원이 원하는 아이디를 입력한다.</p> <p>← 2. 서버는 아이디 중복을 확인한다.</p> <p>    ← 2.a 중복된 아이디가 존재하는 경우 "이미 존재하는 아이디입니다."라고 화면에 표시한다.</p> <p>→ 3. 주문자 또는 점원이 원하는 비밀번호를 입력한다.</p> <p>← 4. 서버는 비밀번호 양식을 확인한다.</p> <p>    ← 4.a 비밀번호 양식이 잘못된 경우 "비밀번호는 4자리 이상이어야 합니다."라고 화면에 표시한다.</p> <p>→ 5. 회원가입이 성공하면 자동로그인 후 메인 화면으로 이동한다.</p>	

## Use Case 2

<b>Use Case – UC2</b>	<b>로그인</b>
<b>Related Requirements</b>	OFR-2, SFR-2
<b>Initiating Actor</b>	주문자, 점원
<b>Actor's Goal</b>	사이트에 로그인 한다.
<b>Participating Actors</b>	주문자, 점원, 유저DB
<b>Precondition</b>	- 회원가입을 한 회원만 로그인 할 수 있다.
<b>Postcondition</b>	-로그인에 성공하면 메인 화면으로 이동한다. -로그인에 실패하면 회원가입을 추천한다.
<b>Flow of Events for main Success Scenario</b>	
<p>→ 1. 주문자 또는 점원이 자신의 아이디와 비밀번호를 입력한다.</p> <p>← 2. 서버는 등록된 회원인지 확인한다.</p> <p>    ← 2.a 등록된 회원인 경우 메인 화면으로 이동한다.</p> <p>    ← 2.b 등록되지 않은 회원인 경우 다시 로그인을 시도하도록 한다.</p> <p>→ 3. 로그인에 3회 이상 실패할 경우 회원가입을 권유하는 메시지를 표시한다.</p>	

## Use Case 3

<b>Use Case – UC3</b>	<b>메뉴보기</b>
<b>Related Requirements</b>	O-FR4, O-FR5
<b>Initiating Actor</b>	주문자
<b>Actor's Goal</b>	피자가게에 있는 메뉴를 확인한다
<b>Participating Actors</b>	주문자, 메뉴DB
<b>Precondition</b>	<ul style="list-style-type: none"> <li>- 주문자는 로그인 되어 있어야한다.(UC2)</li> <li>- 점원이 메뉴에 대한 정보를 넣어야 한다.</li> <li>- 각 메뉴에 대한 사진과 설명이 데이터베이스에 있어야 한다.</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>- 메뉴를 클릭하면 메뉴에 대한 상세보기 페이지로 넘어간다</li> </ul>
<b>Flow of Events for main Success Scenario</b>	
<p>← 1. 선택된 탭에 따라(피자 탭, 사이드 탭, 음료 탭)서버에서 메뉴 목록을 유저한테 띄워준다</p> <p>→ 2. 유저는 띄워진 메뉴 중에 원하는 메뉴를 누른다.</p> <p>← 3. 서버는 유저가 클릭한 메뉴에 대한 설명이 담긴 사진과 내용을 유저한테 보낸다</p> <p>→ 4. 유저가 선택한 메뉴에 대한 상세한 설명이 담긴 페이지를 볼 수 있다.</p>	

## Use Case 4

<b>Use Case – UC4</b>	<b>메뉴 상세 보기</b>
<b>Related Requirements</b>	O-FR5, O-FR6, O-FR7, O-FR8
<b>Initiating Actor</b>	주문자
<b>Actor's Goal</b>	피자가게에 있는 메뉴의 상세 정보를 확인하고 장바구니에 넣는다.
<b>Participating Actors</b>	주문자, 메뉴DB, 장바구니
<b>Precondition</b>	<ul style="list-style-type: none"> <li>- 주문자는 로그인 되어 있어야 한다.(UC2)</li> <li>- 점원이 메뉴에 대한 정보를 넣어야 한다.</li> <li>- 각 메뉴에 대한 사진과 설명이 데이터베이스에 있어야 한다.</li> <li>- 메뉴보기 페이지에서만 접근할 수 있다.</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>- 장바구니 담기 버튼을 누르면 장바구니에 담긴다.</li> <li>- 뒤로 가기 버튼을 누르면, 메뉴 보기 페이지로 이동한다.</li> </ul>
<b>Flow of Events for main Success Scenario</b>	
<p>← 1. 선택된 메뉴에 따라 메뉴 상세 정보를 유저한테 띄워준다.</p> <p>→ 2. 유저는 토핑을 추가할 수 있고 예상 조리 시간을 확인할 수 있다</p> <p>→ 3. 장바구니 버튼을 통해 선택된 메뉴를 장바구니에 담을 수 있다</p> <p>→ 4. 뒤로 가기 버튼을 통해 메뉴 페이지로 돌아갈 수 있다.</p>	

## Use Case 5

<b>Use Case – UC5</b>	<b>장바구니</b>
<b>Related Requirements</b>	O-FR9, O-FR10
<b>Initiating Actor</b>	주문자
<b>Actor's Goal</b>	장바구니에 담긴 목록과 총 금액 및 예상 소요시간을 확인할 수 있다. 장바구니에 담기 메뉴를 취소할 수 있다.
<b>Participating Actors</b>	주문자, 메뉴DB, 장바구니
<b>Precondition</b>	- 주문자는 로그인 되어 있어야한다.(UC2)
<b>Postcondition</b>	- 결제하기 버튼을 누르면 결제하기 페이지로 넘어간다. - 뒤로 가기 버튼을 누르면 메뉴보기 페이지로 넘어간다.
<b>Flow of Events for main Success Scenario</b>	
<p>← 1. 장바구니에 담긴 목록과 총 금액 및 예상 소요 시간을 확인할 수 있다</p> <p>→ 2. 유저가 장바구니에 담긴 메뉴를 취소하면 장바구니 목록에서 삭제할 수 있다.</p> <p>→ 3. 결제하기 버튼을 누르면 결제하기 페이지로 넘어간다</p>	

## Use Case 6

<b>Use Case – UC6</b>	<b>결제하기</b>
<b>Related Requirements</b>	O-FR11, O-FR12, O-FR13, O-FR14
<b>Initiating Actor</b>	주문자
<b>Actor's Goal</b>	주문내역에 대해서 금액을 지불한다
<b>Participating Actors</b>	주문자, 결제서버, 주문DB
<b>Precondition</b>	- 장바구니에 주문자의 주문내용이 저장되어 있다. (UC5)
<b>Postcondition</b>	- 주문자의 주문이 점원에게 나타난다
<b>Flow of Events for main Success Scenario</b>	
<p>→ 1. 주문자는 '결제 수단'을 선택한다</p> <p>← 2. 시스템은 결제 수단 별 결제 대행사 페이지를 보여준다</p> <p>→ 3. 사용자는 결제를 완료한다.</p> <p>← 4. 시스템은 주문내용을 처리한다</p> <p>    ← 4-a. 시스템은 사용자에게 주문 완료 메시지를 전달한다</p> <p>    ← 4-b. 시스템은 주문 내역을 점원에게 전달한다</p>	



## Use Case 7

<b>Use Case – UC7</b>	<b>유저 프로필 관리(프로필 수정)</b>
<b>Related Requirements</b>	O-FR3
<b>Initiating Actor</b>	주문자
<b>Actor's Goal</b>	유저는 유저 프로필 페이지에서 개인정보를 수정할 수 있다.
<b>Participating Actors</b>	주문자, 유저DB, 주문DB
<b>Precondition</b>	- 주문자는 로그인 되어 있어야한다.(UC2)
<b>Postcondition</b>	- 주문자의 프로필이 수정된 정보로 나타난다.
<b>Flow of Events for main Success Scenario</b>	
<p>→ 1. 주문자는 '개인 프로필' 메뉴를 선택한다.</p> <p>← 2. 주문자의 기본 프로필을 보여주고 사용자의 주문 기록이 나타난다.</p> <p>→ 3. 주문자는 '프로필 수정'을 선택한다.</p> <p>← 4. 화면에 '본인 확인을 위해 비밀번호를 입력해주세요.'를 띄운다.</p> <p>    ← 4a. 주문자가 잘못된 비밀번호가 입력되면 '잘못된 입력입니다.'를 띄우고 '개인 프로필' 메뉴로 돌아간다.</p> <p>    ← 4b. 주문자가 옳은 비밀번호를 입력하면 '프로필 수정' 으로 이동한다.</p> <p>→ 5. 주문자는 '프로필 수정'에서 원하는 정보를 수정하고 수정완료를 선택한다.</p> <p>← 6. 입력된 정보를 저장하고 '수정이 완료되었습니다.'를 띄우고 '개인 프로필' 메뉴로 이동한다.</p>	

## Use Case 8

<b>Use Case – UC8</b>	<b>주문 관리</b>
<b>Related Requirements</b>	S-FR6, S-FR7, S-FR8, S-FR9
<b>Initiating Actor</b>	점원
<b>Actor's Goal</b>	<p>점원은 주문이 들어오면 주문 내역을 확인하고 수락과 거절을 선택할 수 있다.</p> <p>점원은 수락된 주문 내역을 확인할 수 있다.</p> <p>점원은 주문 내역에서 '조리 완료'를 눌러 활성화 할 수 있다.</p> <p>수락을 선택하면, 메시지를 보내고 QR코드를 주문DB에 저장한다.</p>
<b>Participating Actors</b>	주문DB, 주문자, 결제서버
<b>Precondition</b>	- 주문자가 메뉴를 정해 주문한 뒤에 결제를 마친 정보가 주문DB에 저장되어 있다. (UC4&UC5)
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>- 점원은 주문자의 주문정보에 QR코드가 활성화된다.</li> <li>- 주문자의 주문 취소가 비활성화 된다.</li> <li>- 점원은 주문자에게 조리가 완료되었음을 알린다.</li> <li>- 점원이 주문을 거절하면 해당 주문은 환불된다.</li> </ul>
<b>Flow of Events for main Success Scenario</b>	
<p>→ 1. 점원은 '주문 관리' 메뉴를 선택한다.</p> <p>← 2. 시스템은 가게에 들어온 주문의 정보(메뉴 구성, 접수 시간, QR코드)를 띄운다.</p> <p>    ← 2a. 이미 주문자가 수령완료한 주문을 띄운다.</p> <p>    ← 2b. '조리 완료'되었지만 주문자가 아직 수령하지 않은 주문을 띄운다.</p> <p>    ← 2c. 점원이 아직 '조리 완료'를 선택하지 않은 주문을 띄운다.</p> <p>    ← 2d. 새로 들어왔지만 '주문 확인'하지 않은 주문을 띄운다.</p> <p>← 3. 시스템은 새로 들어온 주문을 '주문 수락/거절'과 함께 2d에 띄운다.</p> <p>    ← 3a. 점원이 '주문 수락'을 선택하면 해당 주문이 2c로 이동하고</p>	

시스템은 주문자에게 '주문 수락'과 QR코드를 활성화한다.

← 3b. 점원이 '주문 거절'을 선택하면 해당 주문이 2d에서 삭제된다.  
또한 해당 주문이 취소되었다고 주문자에게 알리고 주문자가 결제한 금액이  
환불된다.

← 3c. 점원이 선택하기 전에 주문자가 취소하면 해당 주문이 2d에서  
삭제된다.(UC-11)

→ 4. 점원은 해당 주문의 조리가 완료되면 해당 주문의 '조리 완료'를  
선택한다.

← 5. 시스템은 해당 주문을 2b로 옮기고 주문자에게 '조리 완료'를  
활성화한다.

## Use Case 9

<b>Use Case – UC9</b>	<b>가게 정보 수정</b>
<b>Related Requirements</b>	S-FR3, S-FR4, S-FR5
<b>Initiating Actor</b>	점원
<b>Actor's Goal</b>	자신의 가게에 대한 메뉴, 영업시간, 휴무일 등의 정보를 관리한다.
<b>Participating Actors</b>	매장DB, 메뉴DB
<b>Precondition</b>	- 점원의 계정(관리자 계정)으로 로그인되어있어야 한다.
<b>Postcondition</b>	- 수정한 정보가 화면에 보여진다.
<b>Flow of Events for main Success Scenario</b>	
<p>← 1. 매장 정보 수정에 관한 활동들을 화면에 보여준다. (ex) 메뉴 수정, 휴무일 수정, 영업시간 수정, 매장 기본 정보 수정)</p> <p>→ 2. 점원은 '메뉴 수정' 같은 활동을 선택하고 데이터를 삽입, 수정, 삭제 등을 한다.</p> <p>→ 2a. 점원은 '휴무일 수정'을 선택한다.</p> <p>→ 2b. 점원은 '영업시간 수정'을 선택한다.</p> <p>→ 2c. 점원은 '매장 기본 정보 수정'을 선택한다.</p> <p>← 3. 매장DB는 새로운 데이터를 저장하고, 수정이 완료되었다는 알림을 띄운다.</p>	

## Use Case 10

<b>Use Case – UC9</b>	<b>음식수령</b>
<b>Related Requirements</b>	S-FR9, S-FR10,O-FR17
<b>Initiating Actor</b>	주문자
<b>Actor's Goal</b>	<ul style="list-style-type: none"> <li>- 해당 주문자에게 올바르게 음식을 전달 할 수 있다.</li> <li>- 주문자가 음식을 찾으러 오면 직원이 주문자의 QR 코드를 스캔해서 인증을 해줄 수 있다.</li> <li>- QR인증이 성공적으로 되면 주문자에게 음식을 전달해주며, 동시에 자동으로 주문서가 완료 처리될 수 있다.</li> </ul>
<b>Participating Actors</b>	주문DB, 주문자, 점원
<b>Precondition</b>	- 주문자가 메뉴를 정해 주문한 뒤에 결제를 마친 정보가 주문DB에 저장되어 있다. (UC4&UC5)
<b>Postcondition</b>	- 주문목록의 대기 큐에서 해당 주문건이 사라지며, 남은 주문 목록에 대해서 업데이트가 되어야 한다..
<b>Flow of Events for main Success Scenario</b>	
<p>→ 1. 주문자가 음식을 찾으러 방문한다.</p> <p>→ 2. 주문이 음식에 대한 QR 코드를 제출하고 직원은 이를 스캔한다.</p> <p>← 3. 주문DB는 스캔한 정보가 유효한지 체크한 뒤 결과를 보여준다.</p> <p>    ← 3a. 스캔 정보가 유효하면 "성공되었음" 결과를 보여준다.</p> <p>    ← 3b. 스캔 정보가 유효하지 않으면 "다시 스캔해주세요" 결과를 보여준다.</p> <p>→ 4. 스캔정보가 유효할 경우 직원은 "성공되었음" 결과를 확인한 뒤 주문자에게 음식을 전달한다.</p> <p>← 5. 시스템은 해당 주문건에 대해서 큐에서 제거하고, 남은 주문목록에 대해서 업데이트 한다.</p>	

## Use Case 11

<b>Use Case – UC11</b>	<b>판매내역 확인</b>
<b>Related Requirements</b>	S-FR11
<b>Initiating Actor</b>	점원
<b>Actor's Goal</b>	점원은 각각 시간 범위에 대해서 매출 내역을 확인할 수 있어야 한다.
<b>Participating Actors</b>	점원, 주문DB
<b>Precondition</b>	- 점원의 계정으로 로그인되어있어야 한다.
<b>Postcondition</b>	- 주문DB는 점원이 요구한 시간 범위에 대한 매출 내역에 대한 화면을 보여주어야 한다.
<b>Flow of Events for main Success Scenario</b>	
<p>→ 1. 점원이 로그인한다.</p> <p>← 2. 주문DB는 이제까지 모든 매출을 담고 있는 화면을 최근 목록부터 보여준다.</p> <p>→ 3. 점원이 조회하고 싶은 시간대를 입력해서 주문DB에 정보를 제공한다.</p> <p>← 4. 주문DB는 조회된 시간 조건에 따른 결과를 반환한다.</p> <p>    ← 4a. 조회시간이 잘못된 입력이면, "다시 입력 하세요"를 보여준다</p> <p>    ← 4b. 조회시간이 올바른 입력인데, 해당시간에 판매내역이 없으면, "조회된 결과가 없습니다"를 보여준다</p> <p>    ← 4c. 조회시간이 올바른 입력이고, 해당시간에 정보가 있으면, 최신순부터 정렬해서 결과를 보여준다.</p> <p>→ 5. 점원은 해당내역을 볼 수 있다.</p> <p>    → 5a. "엑셀로 추출하기" 버튼을 누르면 csv파일 형태로 다운을 받을 수 있다.</p> <p>    → 5b. "프린트 하기" 버튼을 누르면 해당정보를 종이로 출력할 수 있다.</p>	