1. **테스트 케이스(Test/Use Case)**

**[] 대괄호**: Entity Object, 실제 데이터 속성을 가지고 있는 객체, 주로 테이블과 맵핑된 엔티티/리포지터리 객체

**{} 중괄호**: Control Object(시스템), 요청 및 응답에 대한 흐름을 처리하는 객체, 주로 API 컨트롤러/서비스 객체

**() 소괄호**: Boundary Object: 사용자와 시스템 간에 경계를 처리하는 객체, 주로 UI객체

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **케이스 명** | **액터** | **이벤트 흐름** | **선행조건** | **예상결과** |
| 1. | 홈페이지 접속 | subscriber | 1. **[구독자]**가 **(Beanstore Home)**에 접속  2. 상단에 **(제품목록)**, **(마이페이지)**, **(Contact US) (장바구니)** 항목이 있는 **(네비게이션바)** 출력  3. **(네비게이션바)** 하단에 **(브랜드 소개)**, **(배경이미지)** 출력 | 홈화면을 **(Beanstore Home)**으로 지정해야함 | **(네비게이션바)**를 통해 **[구독자]**가 페이지 이동을 할 수 있음 |
| 2. | 제품목록 조회 | subscriber | 1. **[구독자]**가 네비게이션바의 **(제품목록)** 메뉴를 클릭  2. **(제품목록)** 화면으로 이동  3. **{제품목록시스템}**이 전체 **[제품정보]** 목록을 출력  4. 모든 **[제품정보]** 목록을 **(4X4 그리드 형태)**로 출력하고, 각 그리드에 **(가격)**, **(제품명)**, **(제품사진)**, **(업체명)** 출력 | **[제품정보]** DB테이블이 있어야함  **[Partner]**와 메세지큐로 통신할 서버가 있어야함 | 모든 **[제품정보]** 목록을 **(4X4 그리드 형태)**로 출력  **(가격)**, **(원두이름)**, **(제품사진)**, **(업체명)** 등 요약한 정보를 출력 |
| 3. | 제품상세 조회 | subscriber | 1. **[구독자]**가 **(제품목록)** 중 한 제품을 클릭  2. **(제품상세)** 화면으로 이동  3. 화면의 왼편에 **(제품사진)**, **(제품명)**, **(원두정보)**, **(파트너정보)**, **(하트아이콘)**을 출력  4. 화면 오른편에 **(구독옵션)** 선택 창을 출력함  4-1. **[구독자]**는 **(가격)**, **(원두용량)**, **(구독기간)**, **(분쇄도)**, **(수량)**을 선택할 수 있고, 선택 사항에 따라 **(총 결제금액)**을 보여줌  **[구독자]**는 **(구독옵션)** 항목을 모두 선택하고 **(구독하기) (장바구니)** 버튼 클릭 | **[제품정보]** DB테이블이 있어야함  **{제품상세시스템}**이 전체 **[제품정보]** 중 1개의 **[제품정보]**를 id로 조회해서 출력함  **[구독자]**는 **(구독옵션)** 항목을 모두 선택해야 다음 화면으로 넘어갈 수 있음 | 왼편에 한 건의 **[제품정보]**를 폼 형식으로 출력  오른편에 **(구독옵션)** 선택 폼 출력  **(구독옵션)**창은 오른편에 고정, **(제품정보)**는 왼편에서 스크롤 가능  **(총 결제금액)** 옆에  **(구독하기) (장바구니)**  버튼 출력  **[구독자]**가 옵션 선택 후 **(구독하기)** 클릭 시 **(구독신청)**으로 이동  **(장바구니)** 클릭 시 **{장바구니시스템}**이 해당 **[주문정보]**를 **(장바구니)**에 추가  **(장바구니)**를 클릭 시  **(장바구니로 이동하시겠습니까 y/n)** 메시지 출력 |
| 4. | 장바구니 | subscriber | 1. **[구독자]**가 **(네비게이션바)**의 **(장바구니)**클릭  1-1. 혹은 **(제품상세)**의 **(장바구니)** -> **(장바구니이동)** 클릭 시 **(장바구니)** 화면 출력  2. **{장바구니시스템}**이 **(제품상세)**에서 **(장바구니)**에담았던 모든 **[주문목록]**을 테이블 형식으로 조회  3. 테이블 한 **행에 (제품명) (수량) (가격) (업체명) (배송비)**와 **(주문하기)** 버튼 출력  4. **(전체주문) (선택주문)** 버튼이 있고,  **(체크박스)**를 클릭해서 원하는 목록만 선택 주문 가능 | **{장바구니시스템}**에 들어갈 **[주문정보]** 임시데이터 테이블 있어야함  최소 한 개 이상의 **[주문정보]**를 {장바구니시스템}에 추가해야함 | **(장바구니)**에 제품이 없으면 **(제품없음)** 메시지 표시  테이블 형식으로 **(장바구니)**에 담은 **[주문목록]** 표시  **(체크박스)**를 클릭해서 주문 별 선택 가능  **(전체주문) (선택주문)**  버튼으로 **{구독신청시스템}**에 주문 건수 전송 |
| 5. | 구독신청 | subscriber | 1. **(제품상세)**에서 **[구독자]**가 **(구독옵션)** 항목을 모두 선택하고 **(구독하기)** 버튼 클릭  1-1. 혹은 **(장바구니)** 하단 **(구독하기)** 버튼 클릭  2. **(구독옵션)**에서 선택한 옵션을 테이블 한 행에 출력  3. **(주문자정보입력폼)**을 **(구독신청)**에 출력  3-1. **[구독자]**는 **(주문자정보입력폼)**에 **(연락처) (이름) (카드정보) (배송지정보) (배송메모)**를 입력함  3-2. **[구독자]**가 **[주문자정보]**를 **(주문자정보입력폼)**에 입력 완료  4. **[구독자]**가 **(구독)** 버튼 클릭  5. **{구독신청시스템}**에서 위 정보를 **[주문정보]** DB에 추가.  6. 페이지 이동하여 **(주문완료)** 메시지를 보여줌 | **[구독자]**는 **(구독옵션)**과 **(주문자정보입력폼)** 모두 기입 해야함  **[주문정보]** DB테이블이 있어야함  **[Partner]**/**[Admin]**와 메세지큐로 통신할 서버가 있어야함  **{구독신청시스템}**이구독을 진행하는 **[구독자]**에게 임의의 **[구독id]** 발급 | **{구독신청시스템}**의 DB에 **[구독자]**가 등록한 **[주문정보]**의 데이터가 들어감  **[Partner]**/**[Admin]**에게 각각 **[주문정보]**를 보내줌  페이지 이동하여 **(주문완료)** 메시지를 보여줌 |
| 6. | 마이페이지 | subscriber | 1. **[구독자]**가 **(마이페이지)** 메뉴 클릭  2. **{마이페이지시스템}**이 모든 **(주문정보)** 항목을 테이블로 출력  3. **(주문정보)** 테이블 한 행에 **(제품사진) (제품명) (결제가격) (구독기간)**을 출력함 | **{구독신청시스템}**에서 저장한 **[주문정보]** DB테이블이 있어야함  **(주문정보)** 한 행을 클릭하면 **(주문상세)**로 넘어가도록 링크 연결 | **(마이페이지)**목록에 **{마이페이지시스템}**이 전체 **[주문정보]**를 테이블 형식으로 화면에 출력 |
| 7. | 주문상세 | subscriber | 1. **[구독자]**가 **(마이페이지)**에서 **(주문정보)**테이블 중 행 하나를 클릭  2. **{주문상세시스템}**이 1건의 **[주문정보]**를 **[주문id]**로 조회해서 **(주문상세)** 화면에 출력  3. **(주문상세)** 화면에 테이블 형식으로 **(제품사진) (제품명) (수량) (결제가격) (구독기간) (배송지정보)**를 출력  3-1. **(배송지정보)**는 **(수령인) (연락처) (배송지) (배송메모)** 포함 | **{구독신청시스템}**에서 저장한 **[주문정보]** DB테이블이 있어야함  **[주문정보]** DB테이블에 **[주문id]**가 포함 되어야함 | **[구독자]**가 입력한 **[주문정보]** 중 1건을 **(주문상세)**에서 보여줌 |
| 8. | 원두  좋아요  표시 | subscriber | 1. **[구독자]**가 **(제품상세)**에서 빈**(하트아이콘)** 클릭  2. **{제품상세시스템}**에서 **[원두like]** 숫자를 +1씩 증가함  3. **(하트아이콘)**이 채워짐 | **[제품정보]** DB에 **[원두like]**항목이 있어야함  **[Admin]**과 메세지큐로 통신할 서버가 있어야함 | **(하트아이콘)**이 채워지고 **[제품정보]**DB의 **[원두like]** 숫자가 1 증가함  **[Admin]**에 **[원두like]**를 보내면 **[Admin]**에서 like숫자를 합계해서 메시지큐로 보내줌 |
| 9. | 공지사항 | subscriber | 1. **[구독자]**가 **(Notice)** 메뉴를 클릭  2. **{Notice 시스템}**의 **[Notice]** 데이터를 테이블 형식으로 보여줌  3. **(Notice)**의 **(공지사항)**의 한 행을 클릭하면 해당 **(포스팅)**으로 이동함 | **{Notice시스템}**에 **[Notice]** DB테이블이 있어야함 **[Admin]**의 **[notice]**와 동일  **[Admin]**과 메세지큐로 통신할 서버가 있어야함 | **(네비게이션)**의 **(공지사항)** 버튼을 클릭하면 **(공지사항 테이블)**을 보여줌. |