**Sub group1 domain-model**

**[ sequence diagram for uc-2 ]**

Sub group1 담당자 : 고은서, 오예원­­

기능 : 회원 등록 / UI 화면 상에서의 메뉴 선택 / 결제

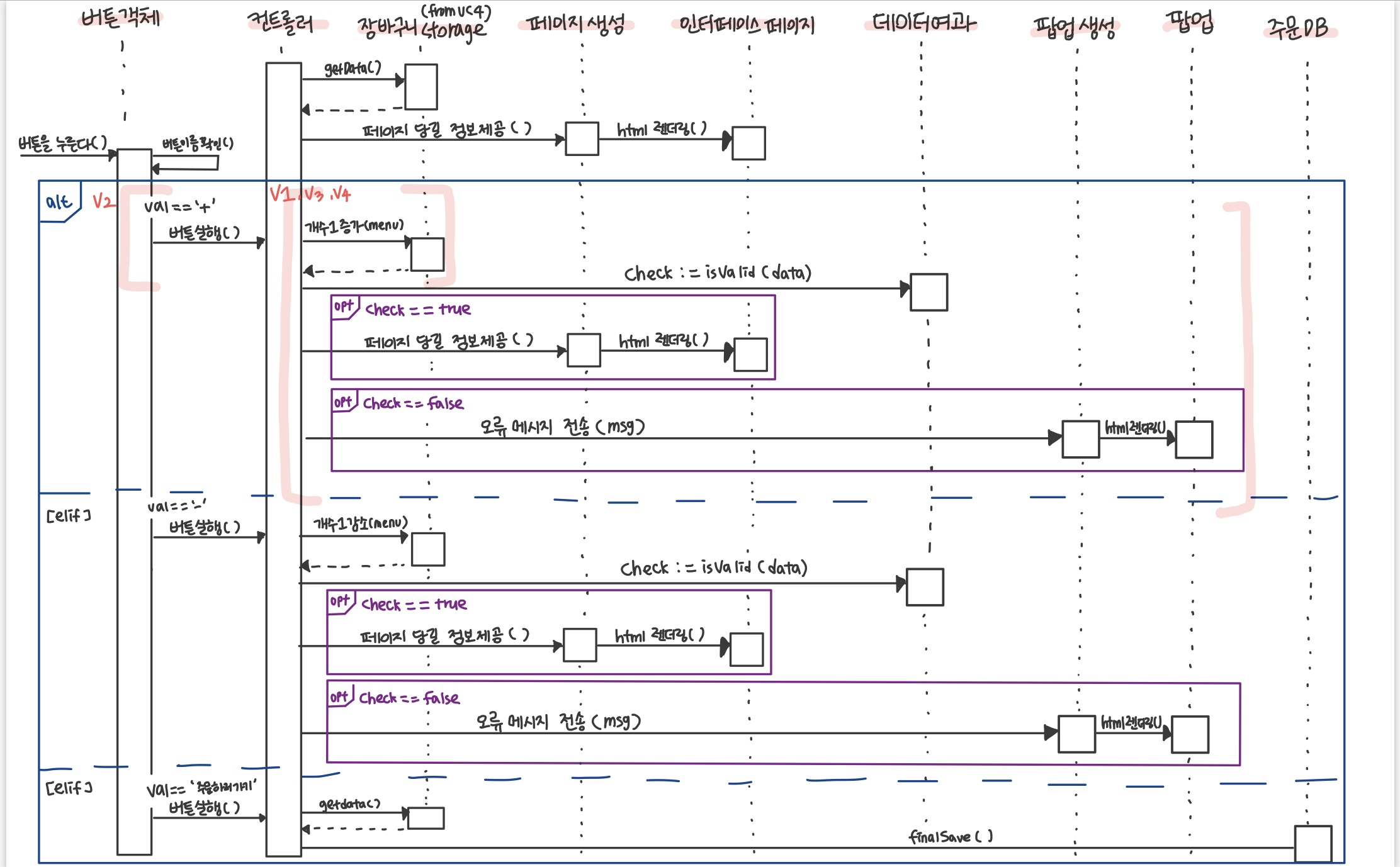
횟수 : 20

회의 시작 날짜 : 2021.05.10 (월)

📌 회의 내용 📌

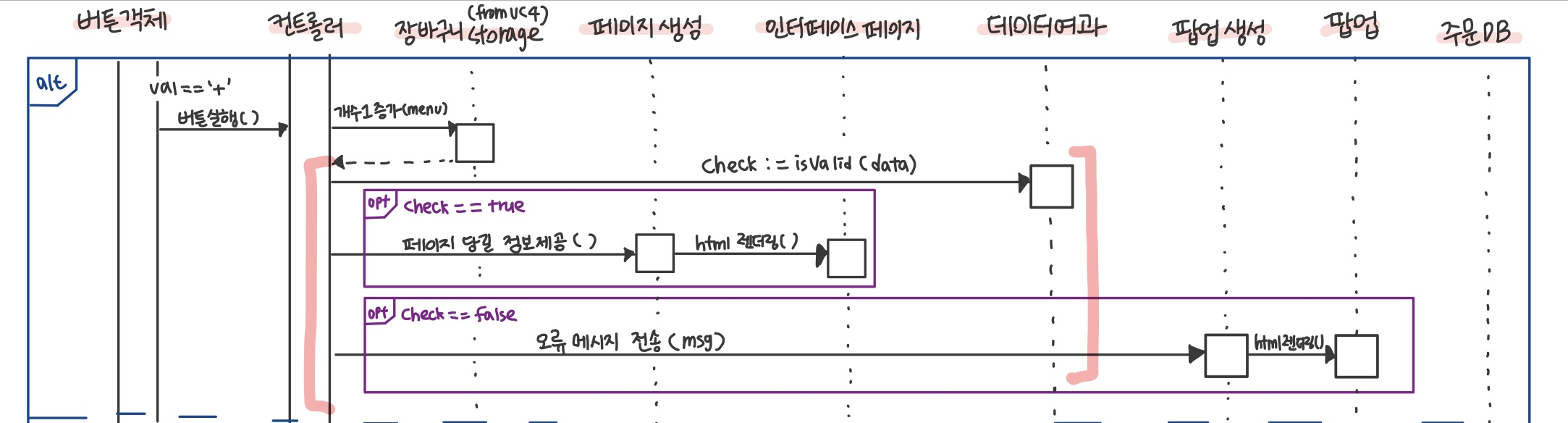
1. Domain model for uc-2에 대한 전체적인 sequence design 그리기
2. 각자 더 나은 설계를 위한 variation 구상해오기
3. Variation들 문서화하여 기록해두기
4. Class diagram 설계를 위해 가장 좋은 설계 선택하기

📘 Sequence Diagram (temporary)



📘 Variation 1

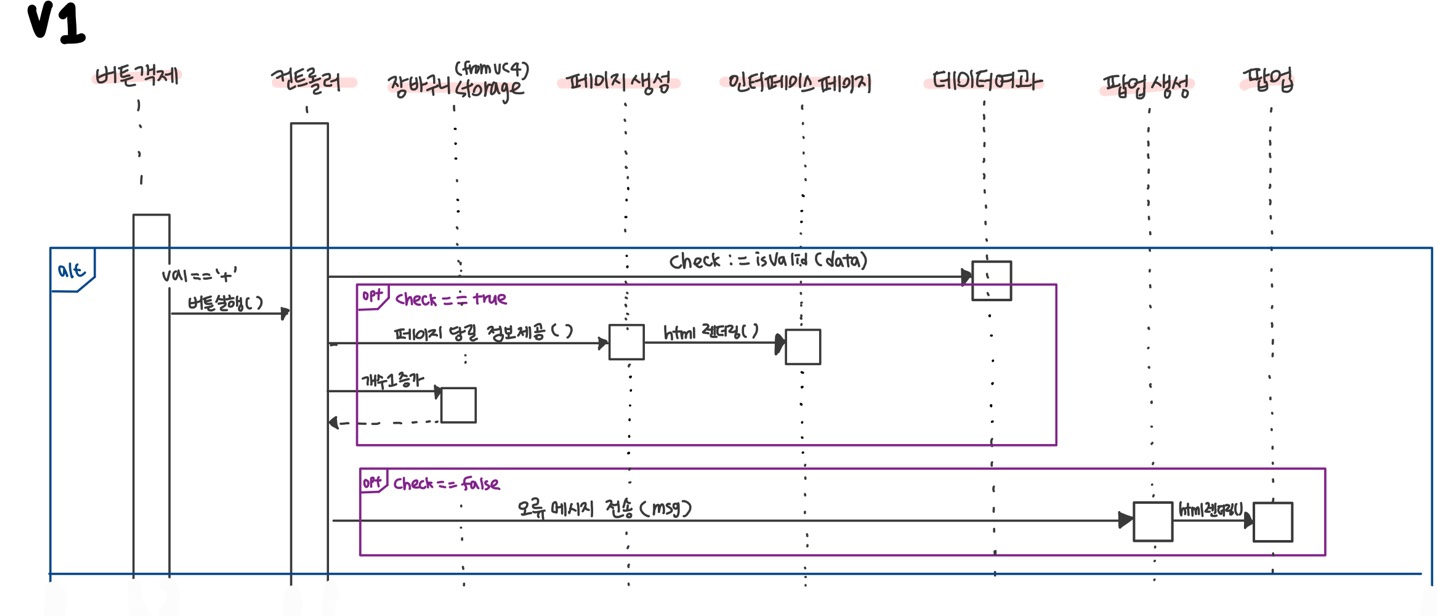
**(기존 sequence diagram)**



**고민)**

초기 설계 당시 +/-버튼객체가 컨트롤러에게 버튼실행() 메세지를 호출하면 컨트롤러는 즉시 장바구니 storage에게 클릭한 메뉴에 대한 개수 증감을 표시하도록 설계하였다. 이는 직관적으로 이해하려 하였을때는 정상적인 설계처럼 보이나, 위에서 아래로 설계과정이 진행된다고 하였을때 check 메세지에 대한 반환값 여부에 따라 값이 true일때에만 개수를 증감시키는 것이 옳다고 판단하였다.

**(variation1)**



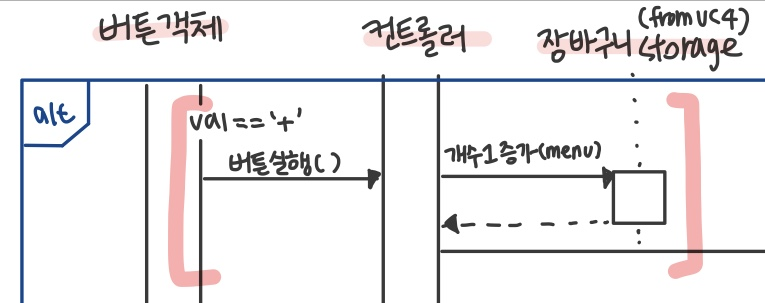
.\

**수정내용)** 먼저 컨트롤러가 장바구니 storage에게 바로 개수 증감()을 호출하는 것이 아니라 check로 정의되어 있는 isvaild() 메소드를 데이터여과 object에게 호출시켜 개수 증감의 행위가 유효한지 파악한다. check의 결과 반환값이 true인 경우만 장바구니 storage 내 특정메뉴의 개수를 증감하여 이를 저장하도록 해야하므로 alt-else로 초기설계와 동일하게 조건을 분기시켰다.

[ 해당 variation최종설계반영 여부 : O ]

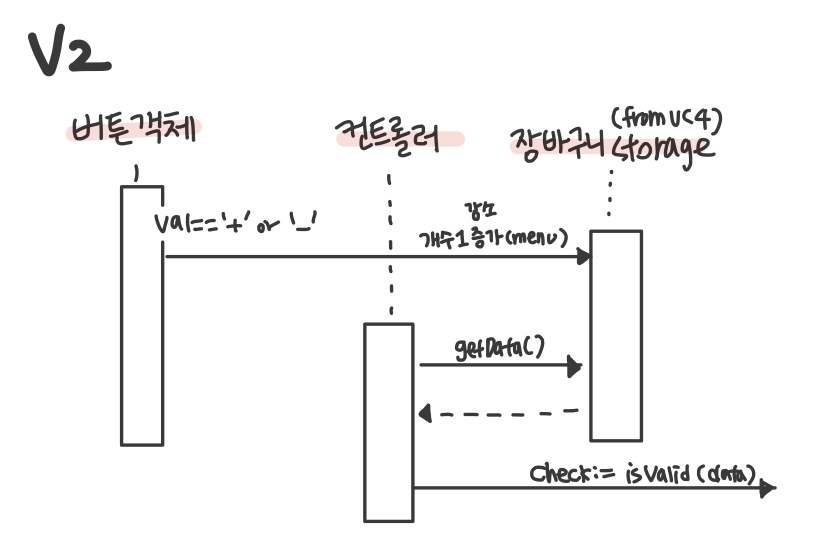
📘 Variation 2

**(기존 sequence diagram)**



**고민** ) 초기엔 +/-버튼을 누를 시에 해당 행위가 컨트롤러에 전해지고, 컨트롤러는 장바구니 storage에 개수를 증가시키는 함수를 호출했다. 하지만 버튼 객체 자체에 연결된 함수를 만들 수 있는데 컨트롤러를 통해 간접적으로 개수를 증가시키는 것은 컨트롤러의 부하를 늘리는 일이라고 판단하였다. 따라서 버튼을 클릭하면 장바구니 storage에 직접적으로 개수를 늘리는 메서드를 설정하였고, 이는 +/- 버튼 두 케이스 모두에 해당한다.

**(variation2)**



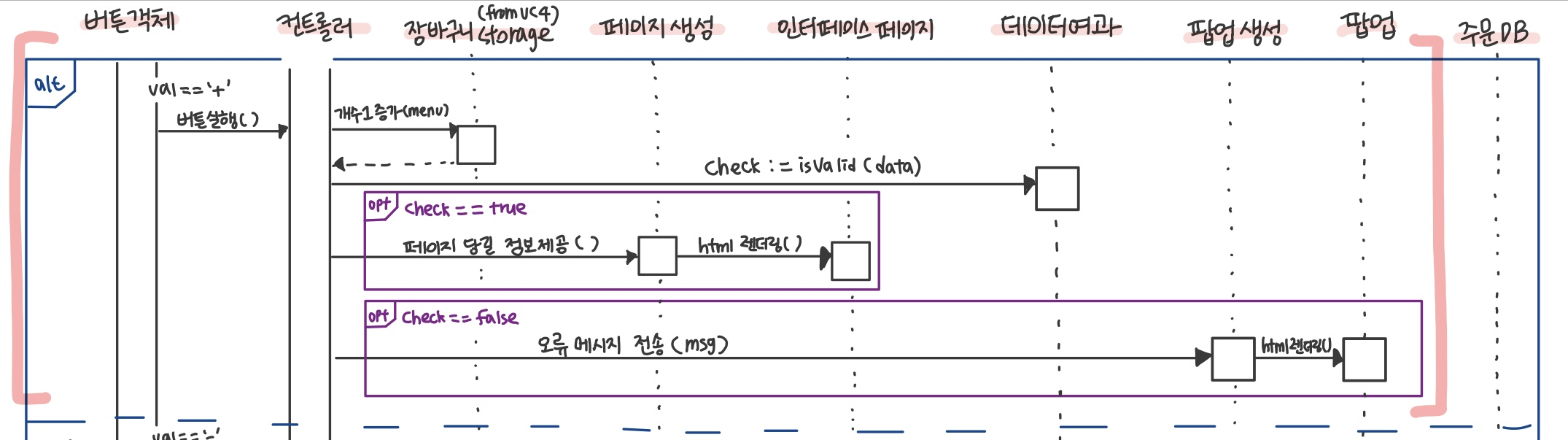
**수정 내용** ) +/- 버튼을 누를 시에 바로 장바구니 storage에 개수 1 증가 또는 감소 메서드를 호출할 수 있도록 수정하였다. 하지만 장바구니 storage속 데이터의 유효성은 따로 판단해주어야 하므로, 컨트롤러에서 해당 데이터를 읽어올 수 있게끔 수정했다.

[ 해당 variation최종설계반영 여부 : X ]

**반영되지 못한 이유)** variation1 이 적용되면서, 메뉴의 개수를 증가/감소시키는 행위가 isValid()이후에 수행되고, isValid의 리턴값을 담는 check 변수가 컨트롤러 산하에 정의되므로 개수를 증가/감소 요청을 컨트롤러가 수행하는게 맞다고 판단하였다.

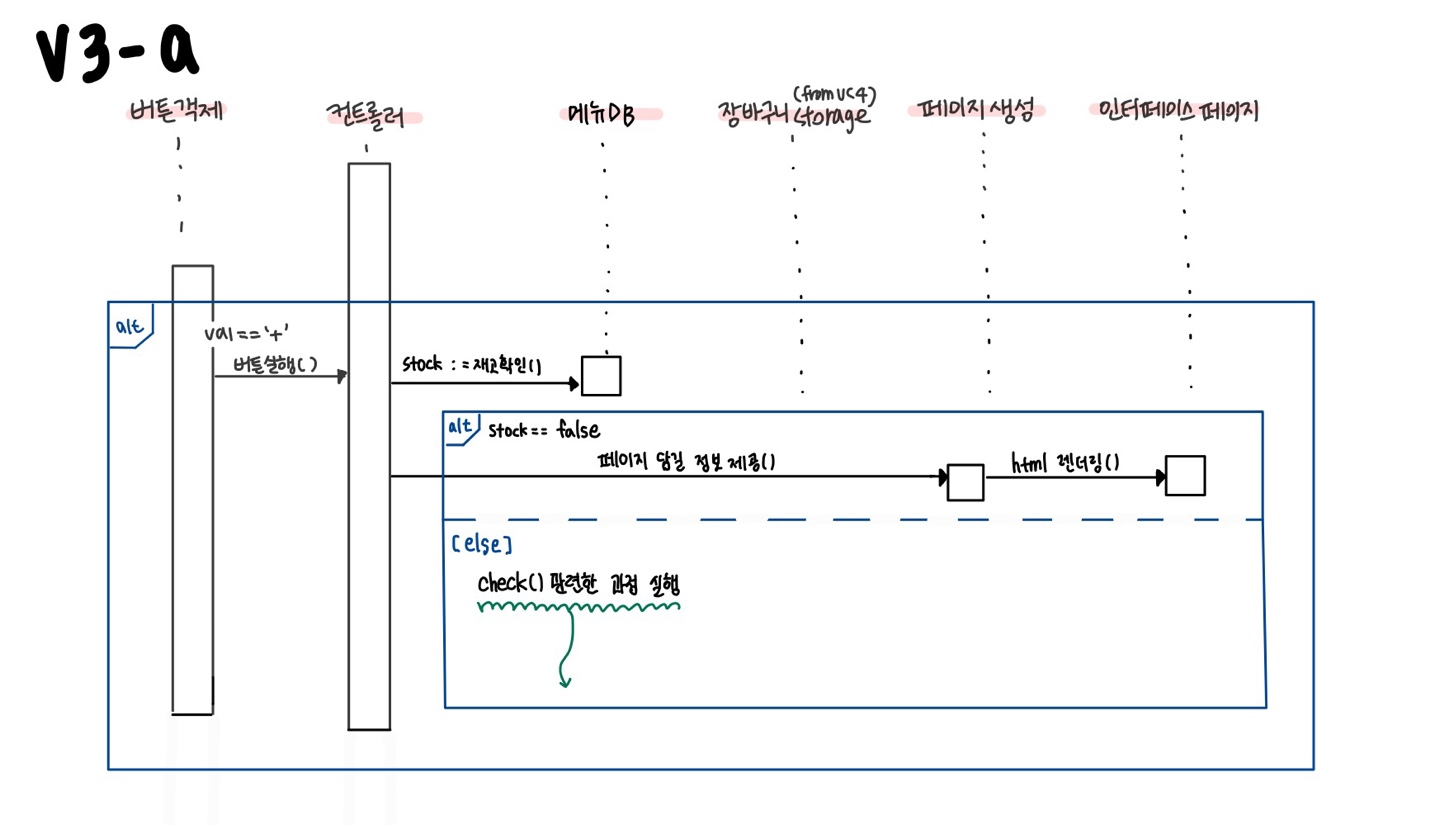
📘 Variation 3

**(기존 sequence diagram)**



**고민(3-a)** ) 초기 설계때 개수조절이 유효한지 파악하는 방법은 오직 컨트롤러가 데이터 여과에게 isvalid() 메세지를 호출하는 부분에서만 존재하였다. 그리고 isvalid에서 처리하는 것은 개수 범위가 정상범위 내에 있는지 파악해주는 용도로 쓰였다. 그런데 +버튼의 경우 개수를 1개 증가시키기 위해선 단순히 범위만을 확인하는 것이 아니라 여전히 해당 메뉴의 재고가 남아있는지까지 고려해야한다고 판단하였다. 따라서 이를 핸들링해주는 object가 필요할 것이라고 생각하였다.

**(variation3-a)**



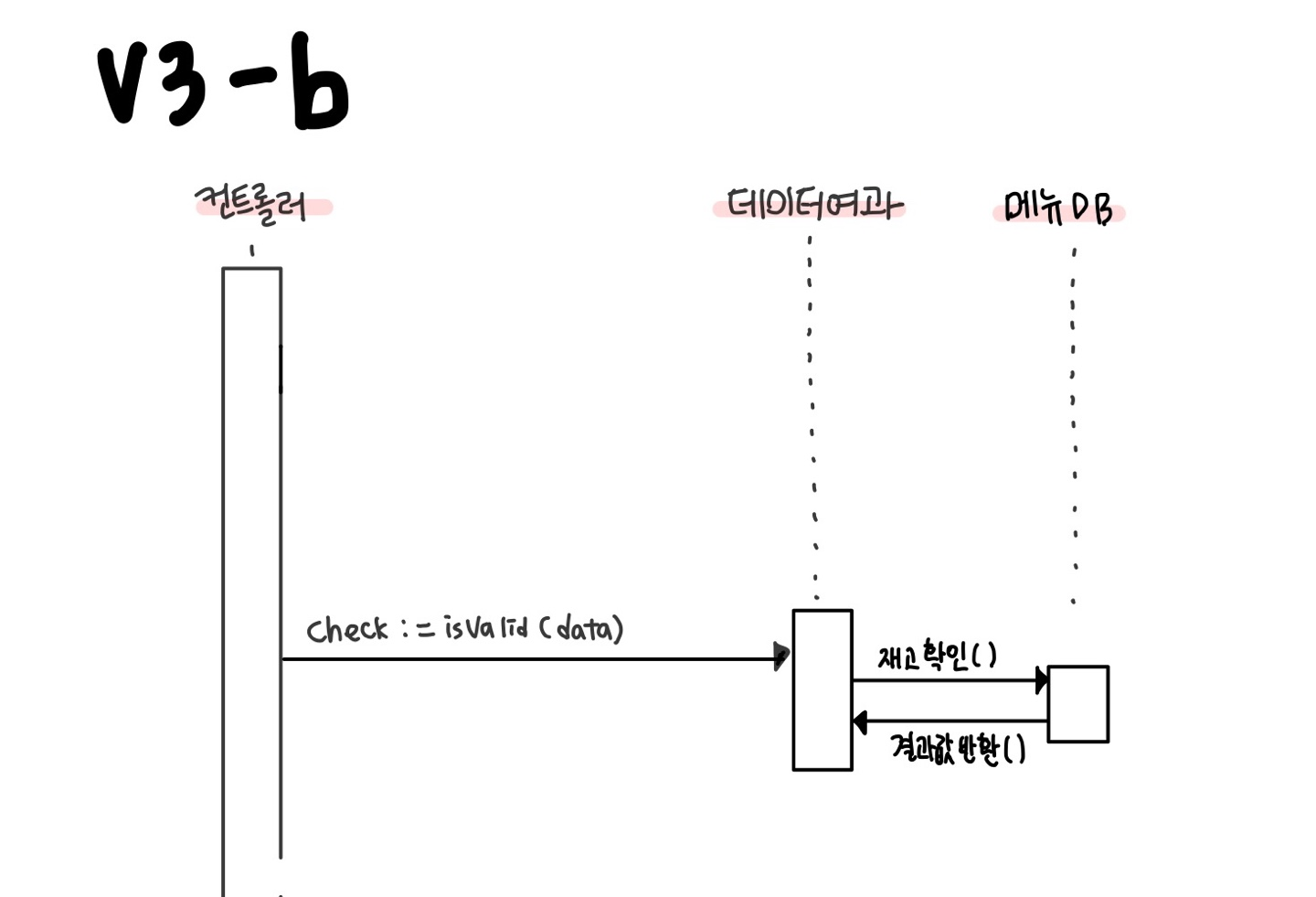
**수정 내용** ) 컨트롤러는 먼저 메뉴db라는 object를 통해 특정메뉴가 1개 이상인지 확인하는 stock() 메세지를 호출시킨다. 이후 stock의 결과값이 false인 경우엔 바로 예외처리를 해주어 팝업을 띄우고, true인 경우엔 최조 설계대로 진행한다.

[ 해당 variation최종설계반영 여부 : X ]

**반영되지 못한 이유)** 메뉴db object를 통해 stock()이란 메세지를 호출하는 것도 이미 아래에 정의된 isvaild() 메세지와 똑같은 목적을 가졌기때문에 이를 분리시키는 것은 오랜기간 뒤에 설계 다이어그램을 보았을 때 혼란이 생길 수도 있다고 생각하여 해당 고민의 부분을 isvalid() 메세지에 포함시켜 cohesion과 통일성을 높이기로 하였다

**고민(3-b)** ) v3-a가 최종 설계에 반영되지 못하면서 다시 최초 설계 다이어그램을 보았을 때, isvalid함수에서 개수조절에 관한 모든 유효성검사를 도맡아한다고 한다면, 재고 확인을 위한 메뉴db와의 연결이 필수적이라고 생각하였다.

**(variation3-b)**

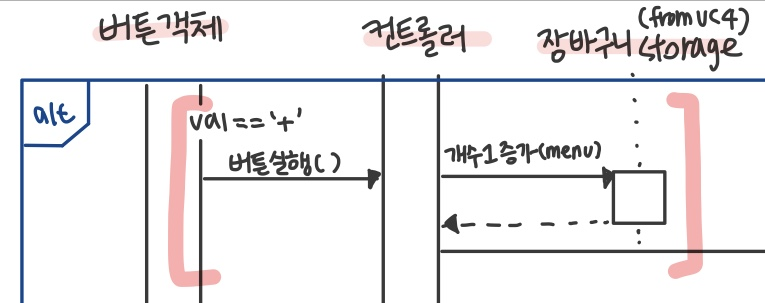


**수정 내용)** v3-a가 최종 설계에 반영되지 못하면서 다시 최초 설계 다이어그램을 보았을 때, isvalid함수에서 개수조절에 관한 모든 유효성검사를 도맡아한다고 한다면, 재고 확인을 위한 메뉴db와의 연결이 필수적이라고 생각하였다.

[ 해당 variation최종설계반영 여부 : O ]

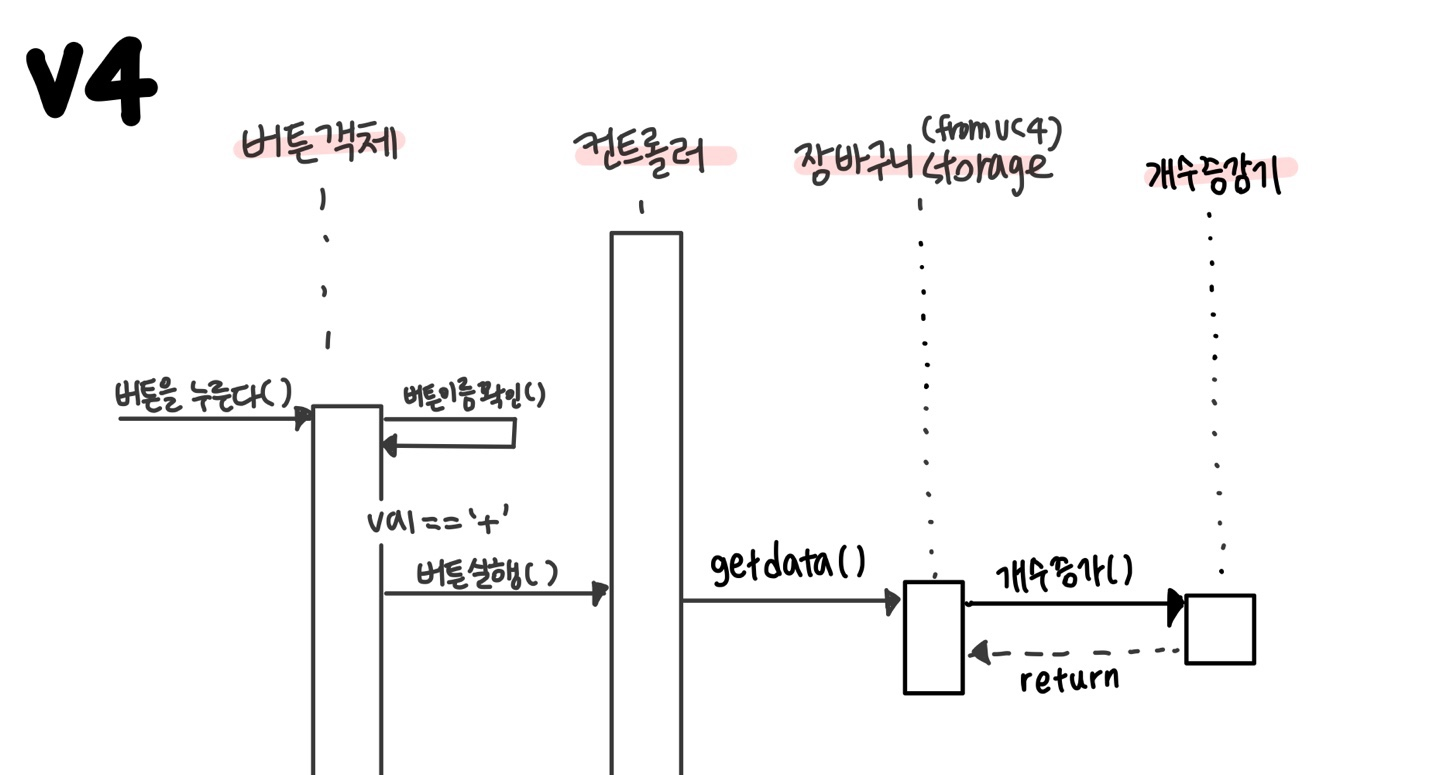
📘 Variation 4

**(기존 sequence diagram)**



**고민(3-b)** ) 초기 설계때엔 컨트롤러가 바로 장바구니 storage에게 개수 증가() 메세지를 호출하도록 설계하였다. 하지만 장바구니 storage는 장바구니에 담겨있는 목록과 그 개수를 담고 있는 object이므로 개수를 조정하는 행위를 하는 것은 cohesion을 낮추게 될 수도 있겠다고 생각하였다.

**(variation4)**



**수정 내용)** 고민의 내용을 바탕으로 variation4를 구상한 모습니다. 컨트롤러는 먼저 장바구니 storage에게 +버튼이 눌린 특정 메뉴를 얻도록 getdata() 메세지를 호출한다. 이후 얻은 메뉴에 대하여 개수 증감기가 연산을 통해 개수를 조절하고 이를 장바구니 storage에 반환하여 저장한다.

[ 해당 variation최종설계반영 여부 : X ]

**반영되지 못한 이유)** 장바구니 storage에서 수행되는 메세지가 많지도 않을 뿐더러 cohesion이 낮아질 수 있다는 단일적인 이유만으로 새로운 object를 생성하는 것은 비효율적이라고 판단하였다.

📘 What is the best design? [최종 sequence diagram]

