



카카오톡 – 선물하기 다이어그램

과목 : 소프트웨어 설계

조 : D조

조원 : 전유승, 권철현, 김종민, 최영은

목차

01

시스템 상황 분석

02

액터 분석

03

유스케이스 분석

04

유스케이스

다이어그램

05

유스케이스

명세서

06

클래스

다이어그램

07

순차

다이어그램

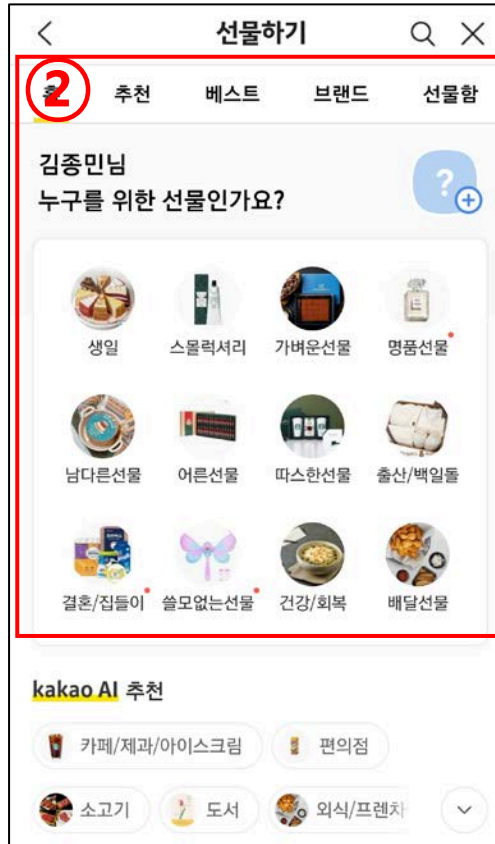
01

시스템 상황 분석

01. 시스템 상황 분석

- ① 회원이 선물하기 목록에 처음으로 접속하면 데이터베이스를 통해 등록된 상품을 보여준다.
- ② 상황(생일, 스몰럭셔리, 가벼운선물), 추천, 베스트, 브랜드, 검색을 선택하면 데이터베이스를 통해 등록된 상품을 보여준다.
- ③ 가격대와 정렬기준을 선택하면 리스트를 재정렬하여 보여준다.
- ④ 회원이 상품을 선택하면 해당 상품의 상세 정보를 표시한다.

①

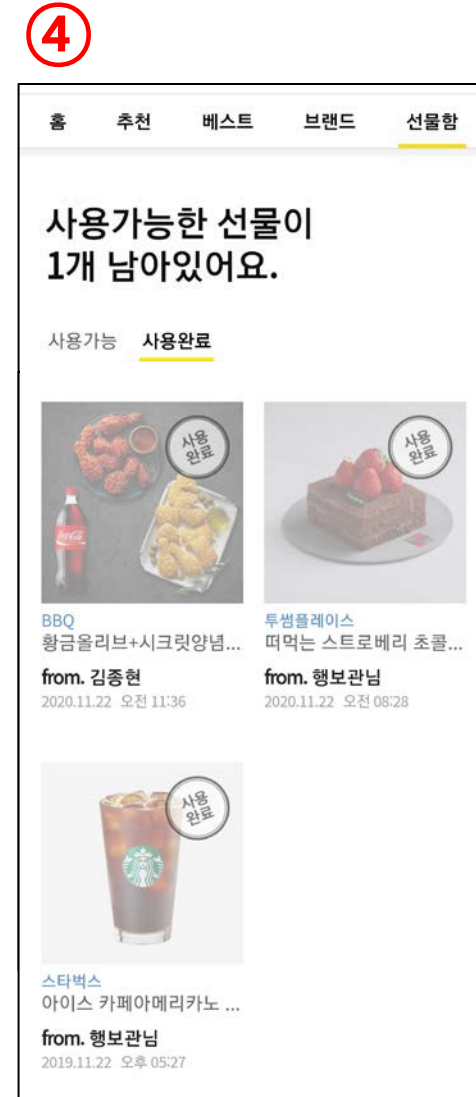


④



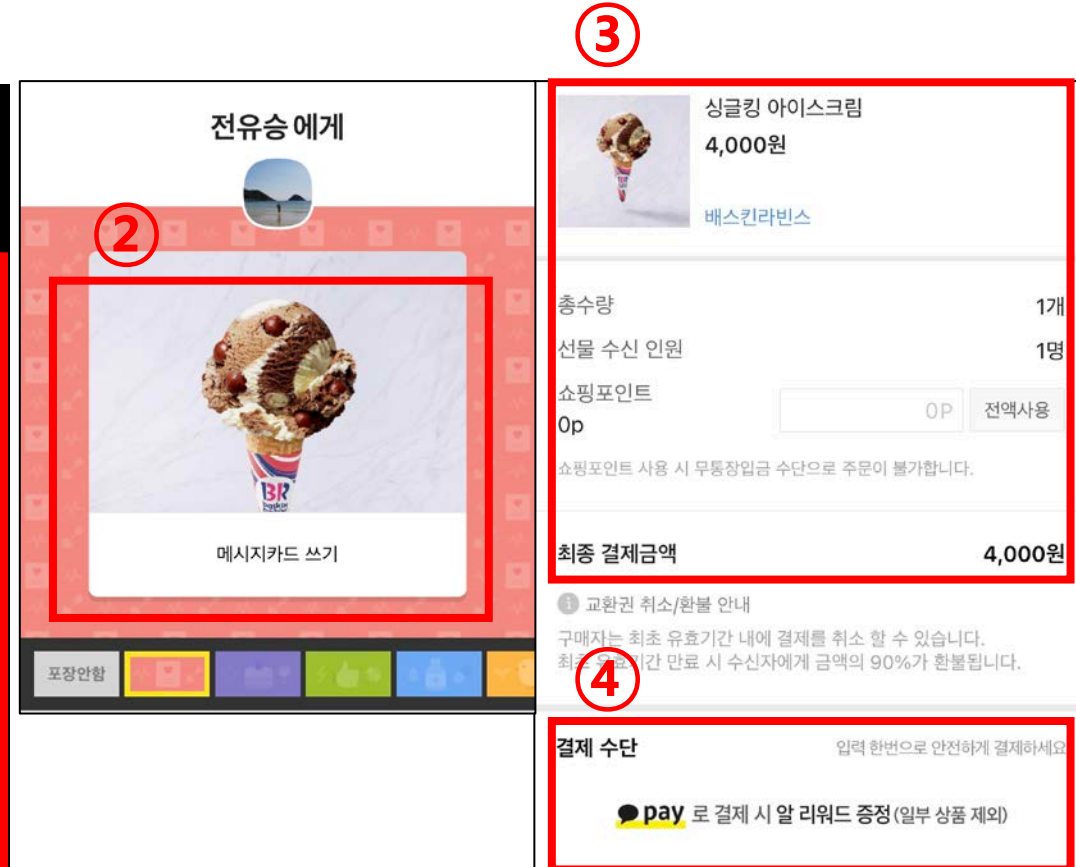
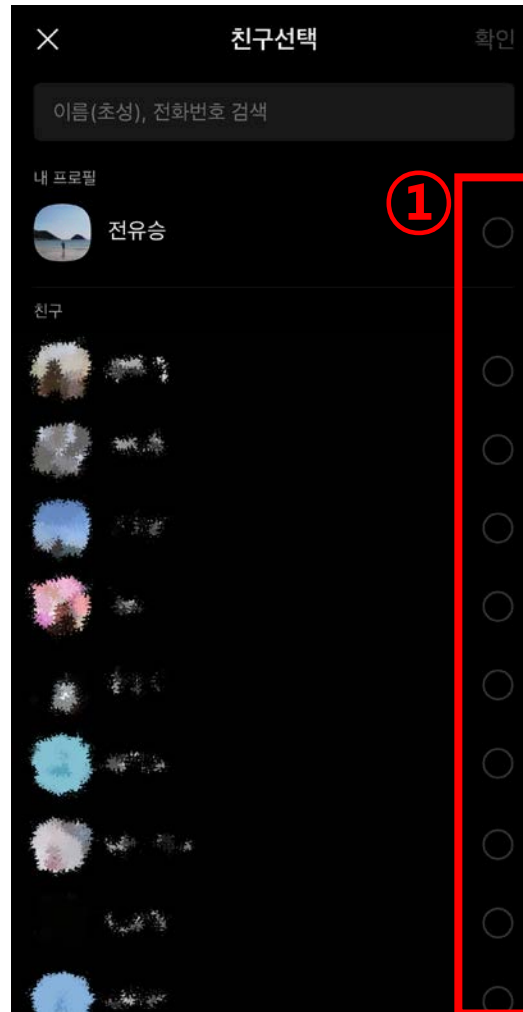
01. 시스템 상황 분석

- ① 회원은 **선물함**을 조회하면 **데이터베이스**를 통해 미사용한 선물의 정보를 볼 수 있다.
- ② 미사용한 선물 목록은 기간이 얼마 남지 않은 순으로 보여준다.
- ③ 해당 선물을 클릭하면 선물의 **상세 정보**를 보여준다.
- ④ 사용 완료 탭을 클릭하면 이전에 사용한 선물 정보의 기록을 보여준다.
- ⑤ 내가 보유한 선물을 **교환**, **환불** 할 수 있다.



01. 시스템 상황 분석

- ① 결제를 하기 위해서는 받는 사람을 선택하여야 한다.
- ② 상품을 보내면서 메시지카드를 쓸 수 있다.
- ③ 결제 창에서는 상품 정보, 가격, 수량, 쇼핑포인트 사용 여부를 확인할 수 있다.
- ④ 결제 수단을 선택할 수 있다.
- ⑤ 결제를 진행하고 나면 자신에게 구매 기록을 저장하고 메시지 카드와 선물을 받는 사람에게 저장한다.
- ⑥ 결제를 실패할 경우 다시 결제 수단을 선택할 수 있다.



01. 시스템 상황 분석

- ① 선물함에서 보유한 선물을 선택하여 환불을 요청할 수 있다.
- ② 환불여부를 한번더 확인하는 메시지를 출력하고, 회원이 확인 버튼을 클릭하면 환불요청을 처리한다.
- ③ 환불을 요청할 경우, 데이터베이스에 조회하여 해당 상품이 환불 가능한지 확인한 후,
- ④ 결제 시스템에서 상품에 대한 금액을 반환하고,
- ⑤ 고객에게서 환불 할 상품을 회수 한 후에
- ⑥ 고객에게 환불 완료 메시지를 전송한다.

홈	추천	베스트	브랜드	선물함
주문 2021.03.30 16:06:15				전체취소
		GS25 퍼페티)츄파춥스250		
		1개 250원		
to. 나				
주문번호 1190719626		거래확인증	①	1개 250원
결제 완료(미사용)		<div>취소</div> <div>교환권 확인</div>		
고객센터	상담하기	문의하기	도움말	

②



01. 시스템 상황 분석

- ① 기프티콘이 아닌 선물에 한하여 교환을 요청할 수 있다.
- ② 교환을 요청할 경우 경우, 데이터베이스에 조회하여 해당 상품이 교환 가능한지 확인한 후,
- ③ 1:1 문의를 통하여 해당 상품의 상태 등을 카카오에서 확인하고 교환이 가능한지 판단한다.
- ④ 교환이 가능할 경우, 새로운 상품을 고객에게 배송하고
- ⑤ 기존의 상품을 회수 한 후에
- ⑥ 고객에게 교환 완료 메시지를 전송한다.

홈추천베스트브랜드선물함

주문 2021.03.30 16:06:15전체취소



GS25
퍼페티)츄파춥스250

1개 250원

to. 나

주문번호 1190719626거래확인증

①

1개 250원

결제 완료(미사용)

취소

교환권 확인

고객센터

상담하기

문의하기

도움말

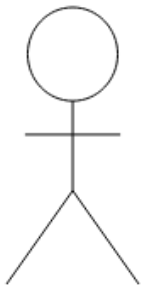
②



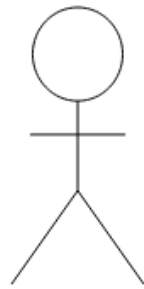
02

엑터 분석

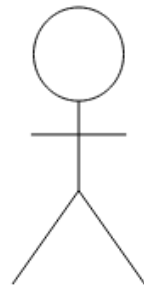
02. 액터 분석



회원



DB시스템



외부결제
시스템

03

유스케이스 분석

03. 유스케이스 분석

U1
상품 목록

U2
DB연결

U3
선물함

U4
상세보기

U5
결제

U6
결제 방식

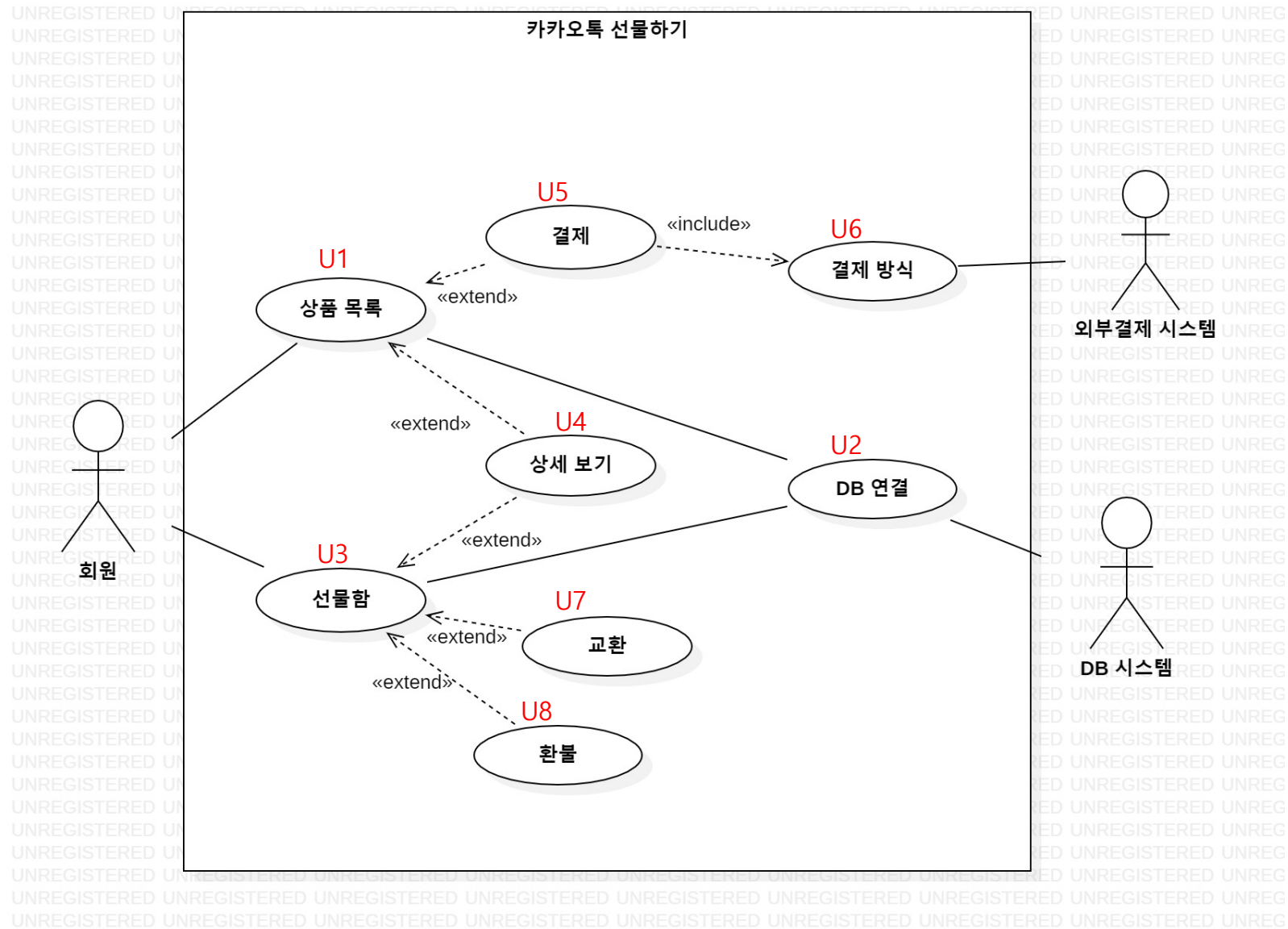
U7
교환

U8
환불

04

유스케이스 다이어그램

04. 유스케이스 다이어그램



05

유스케이스 명세서

05. 유스케이스 명세서 - U1

□ 유스케이스명: 상품목록

□ 액터명: 회원, DB시스템

□ 유스케이스 개요: 회원들이 선물하기 기능에 등록된 상품들을 둘러보고 상세정보를 볼 수 있는 유스케이스 이다.

□ 사전 조건: 회원이어야 한다. 상품이 데이터베이스에 등록되어 있는 상태여야 한다.

□ 이벤트 흐름

○ 정상 흐름

1. 첫 화면에 접속한다.
2. DB연결 유스케이스로 원하는 정보(첫화면, 베스트, 종류, 검색어 등)를 쿼리문 형태로 보내고, 결과를 리스트 형태로 저장한다.
3. 결과를 화면에 보여준다.

○ 선택 흐름

- 상황(생일, 스몰럭셔리, 가벼운선물), 추천, 베스트, 브랜드, 검색 중 하나를 선택하면 정상흐름 2,3을 실행한다.
- 원하는 가격대를 지정하면 포함되지 않는 상품을 리스트에서 제외한 뒤 정상흐름 3을 실행한다.
- 원하는 정렬 기준을 선택하면(추천순, 가격낮은순, 가격높은순, 할인율높은순, 할인율낮은순, 최근등록순) 그에 따라 리스트를 재정렬 후 정상흐름 3을 실행한다.
- 조회할 한가지의 상품을 선택하면 상세정보 유스케이스로 이동한다.

05. 유스케이스 명세서 – U2

□ 유스케이스명: DB연결

□ 액터명: DB시스템

□ 유스케이스 개요: DB시스템에 연결하여 정보를 가져오거나, 정보를 업데이트할 때 사용되는 유스케이스이다.

□ 사전 조건: 상품 목록과 선물함에서 데이터베이스에 검색을 할 때 실행된다.

□ 이벤트 흐름

○ 정상 흐름

1. 데이터베이스에 연결한다.
2. 상품 목록과 선물함에서 보낸 쿼리문을 실행한다.
3. 쿼리의 결과를 리스트로 저장한다.
4. 데이터베이스 연결을 닫는다.
5. 호출한 유스케이스로 복귀하며 결과를 반환한다.

○ 선택 흐름

➤ 데이터베이스 연결에 실패할 경우 에러를 발생한다.

05. 유스케이스 명세서 – U3

□ 유스케이스명: 선물함

□ 액터명: 회원, DB시스템

□ 유스케이스 개요: 회원이 받은 선물과 사용 완료한 기록을 보여주는 유스케이스이다.

□ 사전 조건: 선물하기 회원이어야 한다.

□ 이벤트 흐름

○ 정상 흐름

1. 회원이 선물함을 클릭한다.
2. 회원으로부터 받은 선물을 DB 시스템에 조회한다.
3. 미사용 선물을 기간이 짧은 순으로 회원에게 출력한다.

○ 선택 흐름

- 사용 완료 탭 혹은 사용가능 탭을 클릭하면 DB에서 사용 완료한 정보를 조회한 뒤 출력한다.
- 회원이 선물을 클릭하면 선물에 대한 정보 페이지로 이동한다.
- 환불을 신청할 경우, 환불 페이지로 이동한다.
- 교환을 신청할 경우, 교환 페이지로 이동한다.
- DB로부터 에러 결과를 받으면, 사용자에게 에러 페이지를 보여준다.

05. 유스케이스 명세서 – U4

□ 유스케이스명: 상세 보기

□ 액터명: 회원, DB시스템

□ 유스케이스 개요: 회원이 선택한 선물의 상세 정보를 보여주는 유스케이스 이다.

□ 사전 조건: 선물하기 회원이어야 한다.

□ 이벤트 흐름

○ 정상 흐름

1. 회원이 상품을 클릭한다.
2. 회원이 선택한 상품의 정보를 DB에 조회한다.
3. 받아온 데이터를 이용하여 사용자에게 정보를 보여준다.

○ 선택 흐름

- 선물함에서 상세보기를 했다면, 상품 정보와 결제 정보를 보여준다.
- 선물 목록에서 상세보기를 했다면, 상품 정보만 보여준다. 여기서, 결제 버튼을 클릭하면 결제 페이지로 이동한다.
- DB로부터 에러 결과를 받으면, 사용자에게 에러 페이지를 보여준다.

05. 유스케이스 명세서 – U5

□ 유스케이스명: 결제

□ 액터명: 외부 결제시스템

□ 유스케이스 개요: 회원이 상품을 결제하는 유스케이스이다.

□ 사전 조건: 회원이 상품을 먼저 선택하여야한다.

□ 이벤트 흐름

○ 정상 흐름

1. 상품 코드를 상세보기에서 받아온다.
2. DB로부터 받아온 정보를 표시한다.
3. 받는 사람을 선택한다.
4. 수량을 입력한다.
5. 메시지카드를 입력한다.
6. 쇼핑 포인트를 입력한다.
7. 최종 결제 금액을 표시한다.
8. 결제 방식을 요청한다.
9. 최종 결제가 승인 되었는지 확인한다.
10. 받는 사람에게 메세지를 전송한다.
11. "상품이 정상적으로 결제 되었습니다." 메시지를 보여준다.

○ 선택 흐름

- DB로부터 정보를 받아오지 못 했다면 결제를 취소한다.
- 메시지카드를 입력하지 않은 경우 생략한다.
- 쇼핑 포인트를 입력하지 않은 경우 생략한다.
- 결제를 실패한 경우 결제 방식을 다시 요청한다.
- DB로 메시지 전송을 실패할 경우 외부 결제 시스템에 결제를 취소한다.

05. 유스케이스 명세서 – U6

□ 유스케이스명: 결제 방식

□ 액터명: 외부 결제시스템

□ 유스케이스 개요: 외부 결제시스템에 맞는 결제 방식을 통하여 실질적인 결제가 이루어지는 유스케이스이다.

□ 사전 조건: 요청된 결제 방식과 결제 금액에 대한 정보를 알고 있어야한다.

□ 이벤트 흐름

○ 정상 흐름

1. 요청한 상품의 이름과 총 가격을 확인한다.
2. 선택한 결제 방식에 해당되는 외부 결제 시스템으로 결제를 요청한다.
3. 결제 승인을 확인하고 결제 시스템에서 결제 코드를 받는다.

○ 선택 흐름

- 결제를 실패하면 결제 코드 대신 실패를 반환한다.
- 카카오페이가 결제 방식으로 선택된 경우 카카오페이 결제시스템에 접속한다.
- 신용카드가 결제 방식으로 선택된 경우 신용카드 결제 시스템에 접속한다.
- 핸드폰이 결제 방식으로 선택된 경우 핸드폰 결제 시스템에 접속한다.

05. 유스케이스 명세서 – U7

□ 유스케이스명: 교환

□ 액터명: 회원, DB 시스템

□ 유스케이스 개요: 선물을 받은 회원이 상품에 하자가 있는 등의 이유로 교환을 요청하는 유스케이스이다.

□ 사전 조건: 선물을 수신한 회원이어야 한다.

□ 이벤트 흐름

○ 정상 흐름

1. 회원으로부터 교환 요청을 받는다.
2. 회원의 결제 정보를 확인, 유효성을 검사한다.
3. 1:1 문의를 통해 교환에 적합한지 확인한다.
4. 회원에게 새 상품을 발송하고, 기존의 상품을 돌려받는다.
5. 카카오톡으로 교환 완료 메시지를 보낸다.

○ 선택 흐름

- 1:1 문의 결과 교환이 적합하지 않은 경우에는 교환 절차를 진행하지 않는다는 메시지를 고객에게 발송한다.
- 유효성이 없을 경우 교환을 진행하지 않는다는 메시지를 발송한다.

05. 유스케이스 명세서 – U8

□ 유스케이스명: 환불

□ 액터명: 회원, 외부결제 시스템, DB 시스템

□ 유스케이스 개요: 선물을 보낸 회원이 상품을 취소하는 유스케이스 이다.

□ 사전 조건: 선물을 구매, 혹은 받은 회원이어야 한다.

□ 이벤트 흐름

○ 정상 흐름

1. 회원으로부터 환불 요청을 받는다.
2. 회원의 결제 정보를 확인, 유효성을 검사한다.
3. 결제 정보를 토대로 해당 통장으로 금액을 환불한다.
4. 환불완료 메시지를 출력한다.
5. 카카오톡으로 환불완료 메시지를 보낸다.

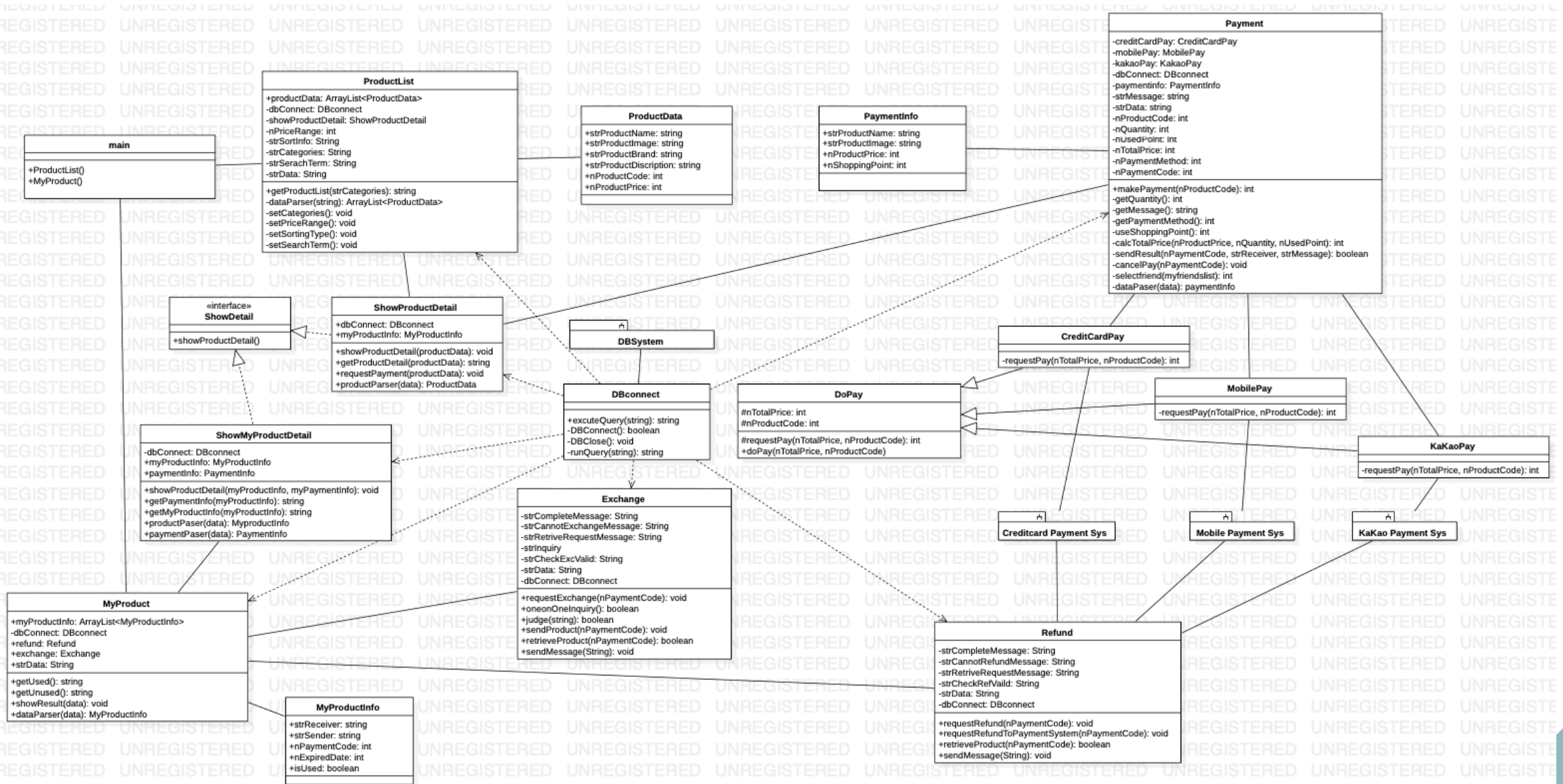
○ 선택 흐름

- 환불하려는 상품의 유효성이 없을 경우, 환불을 진행하지 않고, 고객에게 환불 불가 메시지를 발송한다.

06

클래스 다이어그램

06. 클래스 다이어그램



3. 내선물



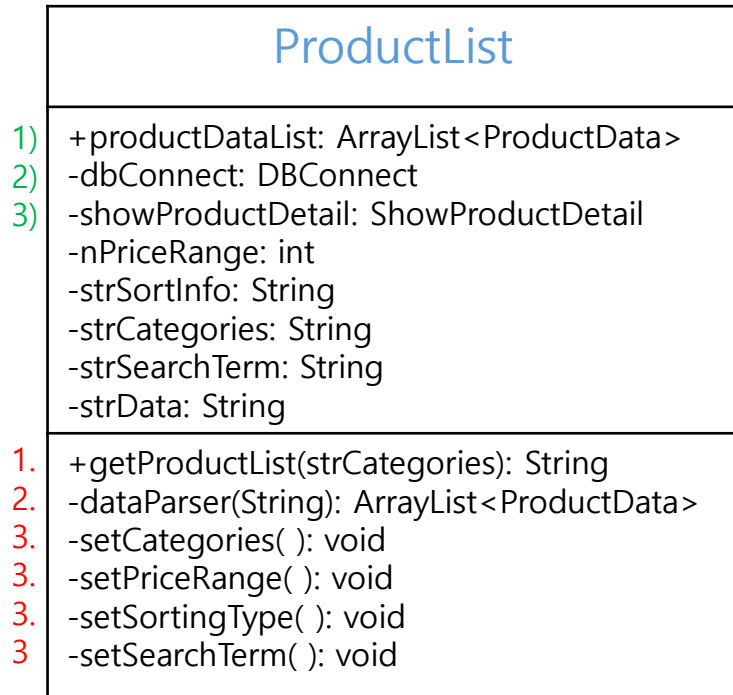
06. 클래스 다이어그램 – main

main	
	productList: ProductList myproduct: MyProduct
1.	MyProduct(): void
2.	ProductList(): void

main: 카카오톡 선물하기를 실행하는 main 클래스

1. MyProduct()
 - 내 선물함을 보여준다.
2. ProductList()
 - 상품목록을 보여준다.

06. 클래스 다이어그램 - 1



ProductList: 상품목록 클래스

- 1) ArrayList<ProductData> productDataList : 후술할 ProductData 클래스를 인자로 갖는 ArrayList, 상품 정보를 저장하는 변수
- 2) DBConnect dbConnect : 데이터베이스 연결 클래스 DBConnect를 실행하기 위한 변수
- 2) ShowProductDetail showProductDetail : 후술할 showProductDetail를 실행하기 위한 변수

1. getProductList(strCategories)

- o DB를 통하여 해당 카테고리의 상품들을 String형태로 받아온다.

2. dataParser(String)

- o 디비에서 가져온 String형태의 데이터를 ProductData List형태로 변환한다.

3. setCategories(), setPriceRange(), setSortingType(), setSearchTerm()

- o 카테고리, 가격범위, 정렬기준, 검색어를 사용자로부터 nPriceRange, strSortInfo, strCategories, strSearchTerm에 입력한다.

06. 클래스 다이어그램 - 2

	<<interface>> <i>DBconnect</i>
1.	+excuteQuery(String): String
2.	-DBconnect(): boolean
3.	-DBclose(): void
4.	-runQuery(String): String

DBconnect: 데이터베이스와 연결하기 위한 클래스, 데이터베이스와의 연결은 해당 클래스를 통하여 이루어진다.

1. excuteQuery(String)

- 쿼리를 실행할 때 호출하는 함수로, DBConnnect, runQuery, DBClose 순서대로 일어날 수 있도록 만든 함수이다.
- runQuery()의 결과를 반환한다.

2. DBconnect()

- 데이터베이스와의 통신을 시작한다. 연결 결과에 따라 성공하면 true, 에러가 나면 false를 반환한다.

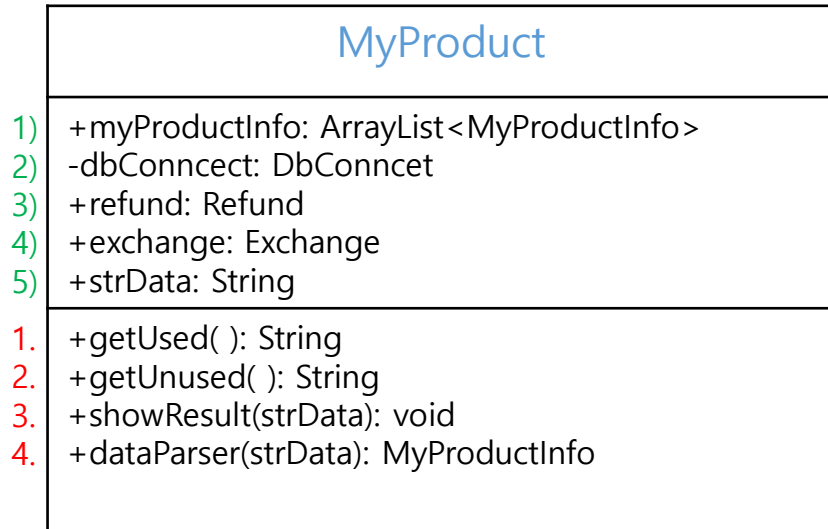
3. DBclose()

- 데이터베이스와의 연결을 해제한다.

3. runQuery(String)

- 데이터베이스에 쿼리문을 전송하고 결과값을 반환한다.
- 쿼리문의 내용에 따라 데이터를 선택하여 가져오거나, 입력할 수 있다.

06. 클래스 다이어그램 - 3

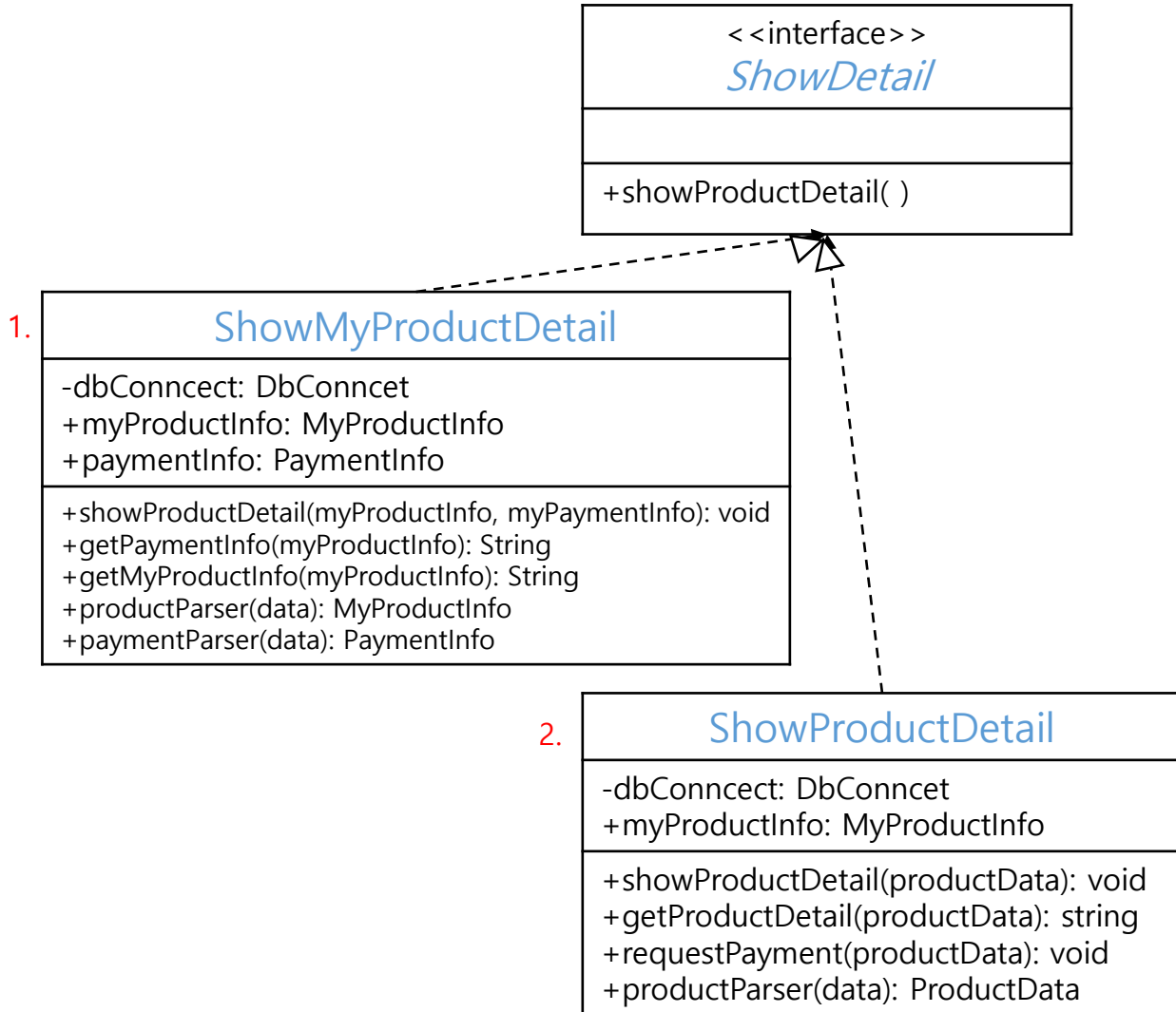


MyProduct: 선물함 클래스

- 1) ArrayList<MyproductInfo> myProductInfo : 후술할 **MyProductInfo** 클래스를 인자로 갖는 ArrayList, 내 선물에 대한 정보를 저장하는 변수
- 2) DBConnect dbConnect : 데이터베이스 연결 클래스 DBConnect를 실행하기 위한 변수
- 3) Refund refund: Refund 클래스 객체, Refund 객체를 사용하기 위한 변수
- 4) Exchange exchange: Exchange 클래스 객체, Exchange 객체를 사용하기 위한 변수
- 5) String strData: DB의 리턴값을 저장하기 위한 변수

1. getUsed()
 - 받은선물 중 사용자가 사용한 선물 목록을 가져온다.
2. getUnused()
 - 받은선물 중 사용자가 사용가능한 선물 목록을 가져온다.
3. showResult(strData)
 - 사용자에게 결과를 보여준다.
4. dataParser(strData)
 - DB로 부터 받은 데이터를 MyProductInfo 타입으로 리턴한다.

06. 클래스 다이어그램 - 4



ShowDetail: 상품의 상세정보를 보여주기 위한 인터페이스
showProductDetail() 메소드를 구현해야 한다.

1. *ShowMyProductDetail*: 선물 받은 상품의 상세정보를 보여주기 위한 클래스
MyProduct 클래스와 관계를 가진다.

- 1) DbConnect dbConnect: 데이터베이스 연결 클래스 DbConnect를 실행하기 위한 변수
- 2) MyProductInfo myProductInfo: MyProductInfo 클래스 객체, MyProductInfo 객체를 사용하기 위한 변수
- 3) PaymentInfo paymentInfo: PaymentInfo 클래스 객체, PaymentInfo 객체를 사용하기 위한 변수

showProductDetail(myProductInfo, myPaymentInfo)

- o 상품 상세정보를 사용자에게 보여준다.

getPaymentInfo(myProductInfo)

- o 선물의 결제정보를 가져온다.

getMyProductInfo(myProductInfo)

- o 선물의 결제정보를 가져온다.

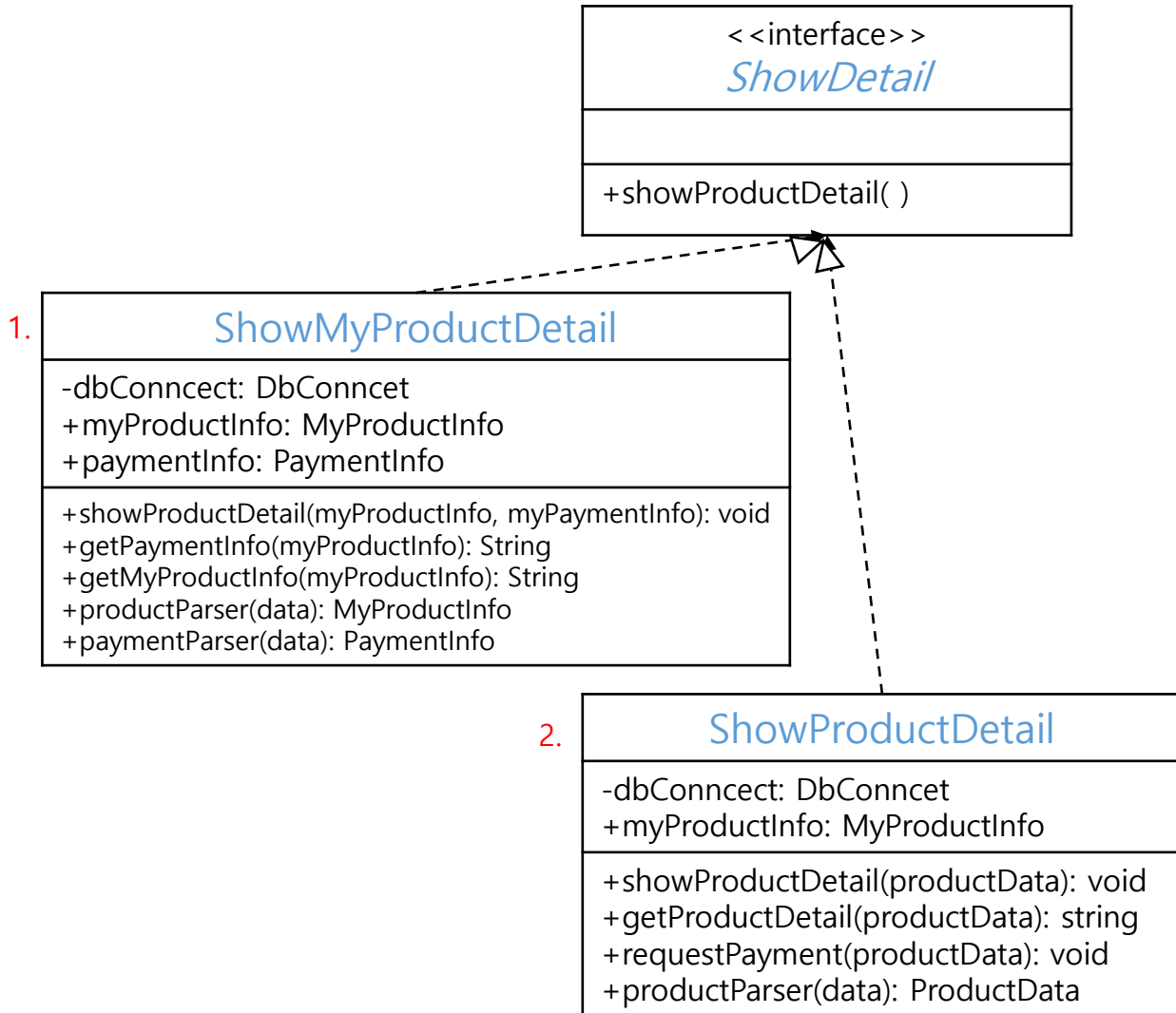
productParser(data)

- o 데이터 값을 파싱하여 MyProductInfo 타입으로 리턴한다.

paymentParser(data)

- o 데이터 값을 파싱하여 PaymentInfo 타입으로 리턴한다.

06. 클래스 다이어그램 - 4



2. **ShowProductDetail**: 상품의 상세정보를 보여주기 위한 클래스
ProductList 클래스와 관계를 가진다.

1) DbConnect dbConnect: 데이터베이스 연결 클래스 DBConnect를
실행하기 위한 변수

2) MyProductInfo myProductInfo: MyProductInfo 클래스 객체,
MyProductInfo 객체를 사용하기 위한 변수

showProductDetail(productData)

- o 상품 상세정보를 사용자에게 보여준다.

getProductDetail(productData)

- o 상품 상세정보를 가져온다.

requestPayment(productData)

- o 결제 페이지로 이동한다.

productParser(data)

- o 데이터 값을 파싱하여 ProductData 타입으로 리턴한다.

06. 클래스 다이어그램 - 5

Payment

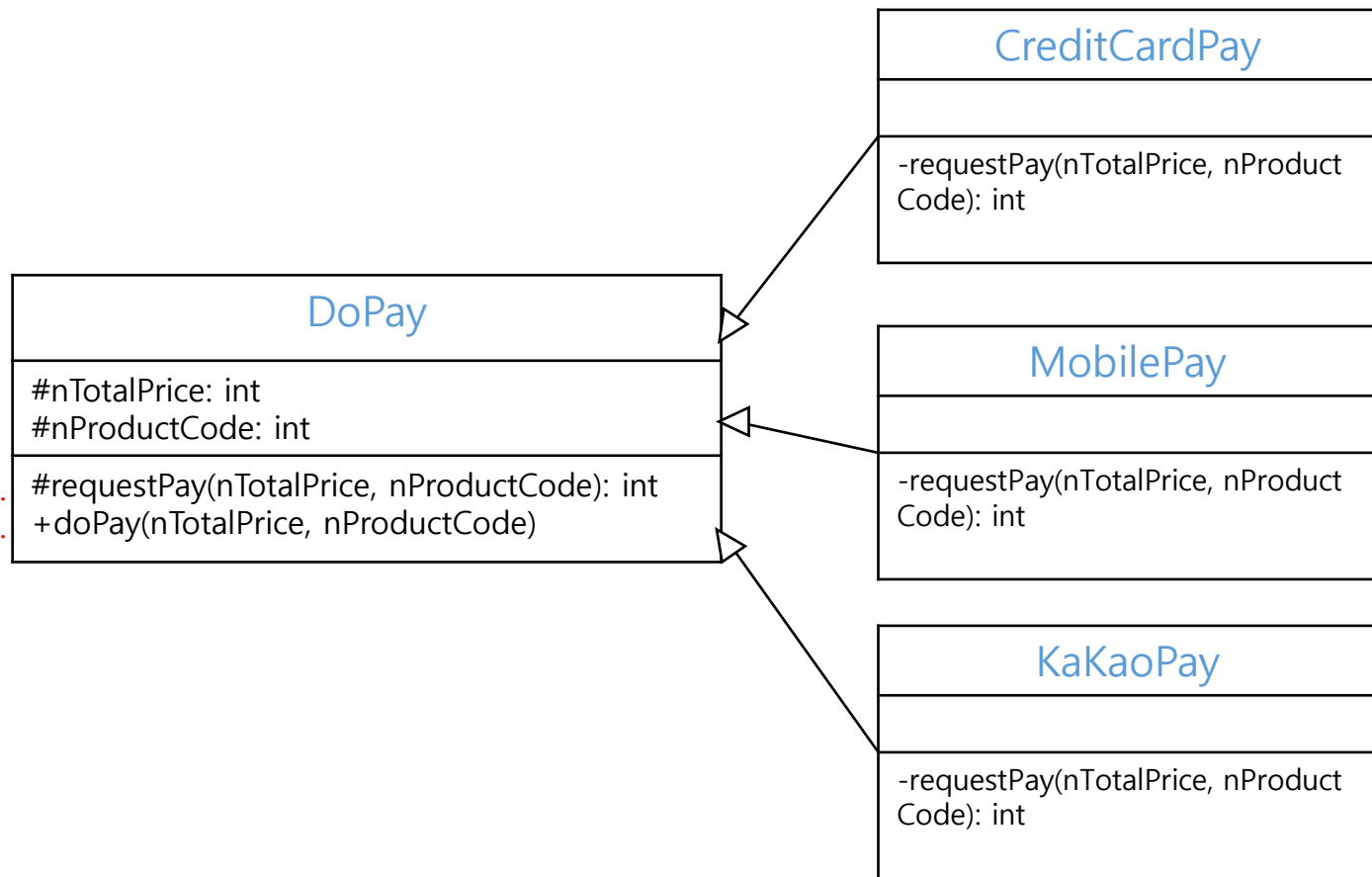
- | | |
|--|---|
| <ul style="list-style-type: none"> 1) -creditCardPay: CreditCardPay 1) -mobilePay: MobilePay 1) -kakaoPay: KakaoPay 2) -dbConnect: dbconnect 3) -paymentinfo: PaymentInfo 4) -myfriendslist: MyfriendsList -strMessage: string -strData: string -nProductCode: int -nQuantity: int -nUsedPoint: int -nTotalPrice: int -nPaymentMethod: int -nPaymentCode: int -nReceiver: int | <ul style="list-style-type: none"> 1. +makePayment(nProductCode): int 2. -getQuantity(): int 3. -getMessage(): string 4. -getPaymentMethod(): int 5. -useShoppingPoint(): int 6. -calcTotalPrice(nProductPrice, nQuantity, nUsedPoint): int 7. -sendResult(nPaymentCode, strReceiver, strMessage): boolean 8. -cancelPay(nPaymentCode) 9. -selectfriend(myfriendslist): int 10. -dataPaser(data): paymentInfo |
|--|---|

Payment: 특정상품을 결제할 때 실행되는 클래스

- 1) 결제 종류에 따라 각각 후술할 클래스와 연결하여 결제를 진행할 변수
- 2) DBconnect dbConnect : 데이터베이스 연결 클래스 DBconnect를 실행하기 위한 변수
- 3) PaymentInfo paymentInfo 후술할 PaymentInfo 클래스 변수, 결제에 필요한 정보를 저장하는 변수
- 4) 친구 목록은 선물하기 시스템에서 구현 요소가 아니므로 임의의 채팅 및 친구 목록 시스템 중의 클래스를 넘겨 받는다고 가정함. 해당 클래스는 구조체 형식이며 내부에는 친구의 이름, 프로필 사진, 고유 회원 번호 등이 있다고 가정.

1. makePayment(nProductCode): int
 - showProductDetail에서 유저가 결제를 누르면 해당 상품의 상품 코드를 넘겨 받는다.
2. getQuantity(): int
 - 수량을 입력 받고 nQuantity에 저장한다.
3. getMessage(): string
 - 메시지(편지)를 입력 받고 strMessage에 저장한다.
4. getPaymentMethod(): int
 - 결제 방식을 입력 받고 nPaymentMethod에 저장한다.
5. useShoppingPoint(): int
 - 사용하고자 하는 쇼핑포인트 금액을 입력 받고 nUsedPoint에 저장한다.
6. calcTotalPrice(nProductPrice, nQuantity, nUsedPoint): int
 - 상품 가격과 개수를 곱하여 나온 값에 포인트 금액을 차감한 최종 금액을 nTotalPrice에 저장한다.
7. sendResult(nPaymentCode, strReceiver, strMessage): boolean
 - DB를 통해 받는 사람에게 메시지와 상품을 전송한다.
8. cancelPay(nPaymentCode)
 - DB전송에 실패하였다면 외부 결제 시스템에 결제 코드를 바탕으로 취소를 요청한다.
9. selectfriend(myfriendslist): int
 - 친구 목록 중에서 선택된 친구의 고유 ID를 반환하여 nReceiver에 저장한다.
10. dataPaser(data): paymentInfo
 - 쿼리 실행을 통해 받은 string타입의 Data 변수를 paymentInfo에 맞게 가공한다.

06. 클래스 다이어그램 - 6



DoPay: 결제와 결제 후 결제코드를 생성하기 위한 인터페이스

1. `requestPay(nTotalPrice, nProductCode): int`
 - 결제를 외부시스템으로 진행한다.
 - 각 결제 시스템에 맞게 재정의 된다.
 - 외부시스템에서 결제를 성공하면 결제코드를 반환하고 실패했다면 0을 반환한다.
2. `doPay(nTotalPrice, nProductCode)`
 - 생성자로서 클래스가 생성될 때 변수를 초기화 한다.

카드(**CreditCardPay**), 휴대폰 결제(**MobilePay**), 카카오페이(**KaKaoPay**) 세 가지의 결제 방식에 따라 **DoPay**를 상속한 클래스들로 나누어진다.

06. 클래스 다이어그램 - 7

Exchange	
1) -strCompleteMessage: String	
2) -strCannotExchangeMessage: String	
3) -strRetriveRequestMessage: String	
4) -strInquiry: String	
5) -strCheckExcValid: String	
6) -strData: String	
7) -dbConnect: DBconnect	
1. +requestExchange(nPaymentCode): void	
2. +OneonOneInquiry(): boolean	
3. +judge(string): Boolean	
4. +sendProduct(nPaymentCode): void	
5. +retrieveProduct(nPaymentCode): boolean	
6. +sendMessage(String): void	

Exchange : Myproduct에서 호출되는 클래스로, 교환 기능을 담당하는 클래스

1. requestExchange(nPaymentCode)
 - o Myproduct에서 호출, 결제 코드에 대한 int 값을 가진다.
2. OneonOneInquiry()
 - o 고객과 1:1 문의를 진행하는 메소드, 고객에게 질문을 보내고, 답변을 받고, judge()의 결과를 리턴한다.
3. judge(string)
 - o 고객의 답변을 토대로 교환 가능 여부를 판별한다.
4. sendProduct(nPaymentCode)
 - o 유저에게 새 상품을 발송하는 메소드이다.
5. retrieveProduct(nPaymentCode)
 - o 유저에게 상품을 반환받는 메소드, 택배나 등기 등으로 상품을 받으면 true를 리턴하는 메소드이다.
6. sendMessage(String)
 - o 유저에게 메시지를 보내주는 메소드. 1), 2), 3), 4) 각각 교환 완료, 교환 불가능, 교환 상품 반환 요청 메시지, 1:1 문의내용이다. String 이고, 이 중 하나를 입력받아 유저에게 해당 메시지를 발송한다.
- 5) strCheckExcValid : 교환의 유효성을 확인할 쿼리, 문자열로 저장되어있다.
- 6) strData : 데이터베이스에서 가져온 정보를 임시로 저장할 String 변수이다.
- 7) dbConnect : DBconnect 클래스 객체, 데이터베이스 연결을 위해 사용한다.

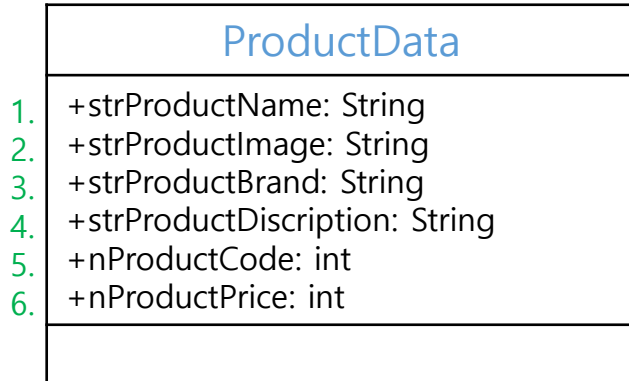
06. 클래스 다이어그램 - 8

Refund	
1)	-strCompleteMessage: String
2)	-strCannotRefundMessage: String
3)	-strRetriveRequestMessage: String
4)	-strCheckRefValid: String
5)	-strData: String
6)	-dbConnect: DBconnect
1.	+requestRefund(nPaymentCode): void
2.	+requestRefundtoPaymentSystem(nPaymentCode): void
3.	+retrieveProduct(nPaymentCode): boolean
4.	+sendMessage(String): void

Refund : Myproduct에서 호출되는 클래스로, 환불 기능을 담당하는 클래스

1. requestRefund(nPaymentCode)
 - o Myproduct에서 호출, 결제 코드에 대한 int 값을 가진다.
2. requestRefundtoPaymentSystem(nPaymentCode)
 - o 결제 시스템에 환불을 요청하는 메소드로, 호출할 때 받은 결제 코드를 사용하여 환불을 요청, 외부 시스템으로 항상 환불은 이루어진다고 가정, 리턴은 없다.
3. retrieveProduct(nPaymentCode)
 - o 유저에게 상품을 반환받는 메소드, 택배나 등기 등으로 상품을 받으면 true를 리턴하는 메소드이다.
4. sendMessage(String)
 - o 유저에게 메시지를 보내주는 메소드. 1), 2), 3), 각각 교환 완료, 교환 불가능, 교환 상품 반환 요청 메시지를 가진 String이고, 이 중 하나를 입력받아 유저에게 해당 메시지를 발송한다.
- 4) strCheckRefValid : 교환의 유효성을 확인할 쿼리, 문자열로 저장되어있다
- 5) strData: 데이터베이스에서 가져온 정보를 임시로 저장할 String 변수이다.
- 6) dbConnect : DBconnect 클래스 객체, 데이터베이스 연결을 위해 사용한다.

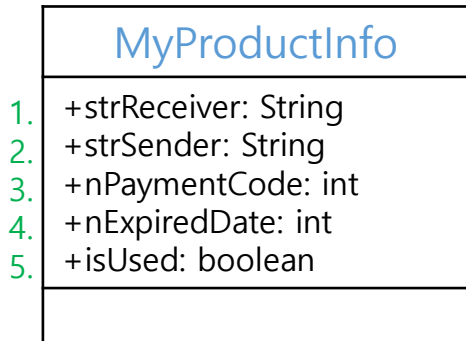
06. 클래스 다이어그램 - 9



ProductData : ProductList에서 struct구조체와 같은 방식으로 사용하는 클래스이며, 아래 값들을 포함한다.

1. strProductName
 - o 상품의 이름값이다.
2. strProductImage
 - o 상품의 이미지이다. URL 형식으로 사용하기 위해 String 값을 가진다
3. strProductBrand
 - o 상품의 브랜드이다.
4. strProductDiscription
 - o 상품의 상세설명이다.
5. nProductPrice
 - o 상품의 가격이다.
6. nProductCode
 - o 상품코드이다. 내부적으로 검색을 할 때 사용할 키 값이다.

06. 클래스 다이어그램 - 9



MyProductInfo : Myproduct에서 struct구조체와 같은 방식으로 사용하는 클래스이며, 아래 값들을 포함한다.

1. strReceiver

- 선물을 받는 사람에 대한 정보를 가진다. 내가 보낸 선물을 확인할 때 사용된다.

2. strSender

- 선물을 보낸 사람에 대한 정보를 가진다. 내가 받은 선물을 확인할 때 사용된다.

3. nPaymentCode

- 결제코드이다. 환불이나 교환을 요청하는데에 사용된다.

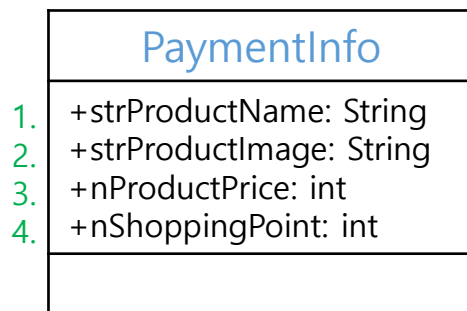
4. nExpiredDate

- 사용기간을 나타낸다. 이를 통하여 환불이나 교환 가능 여부를 확인한다.

5. isUsed

- 사용을 했는지 하지 않았는지를 구분하여 저장하는 boolean 값이다.

06. 클래스 다이어그램 - 9



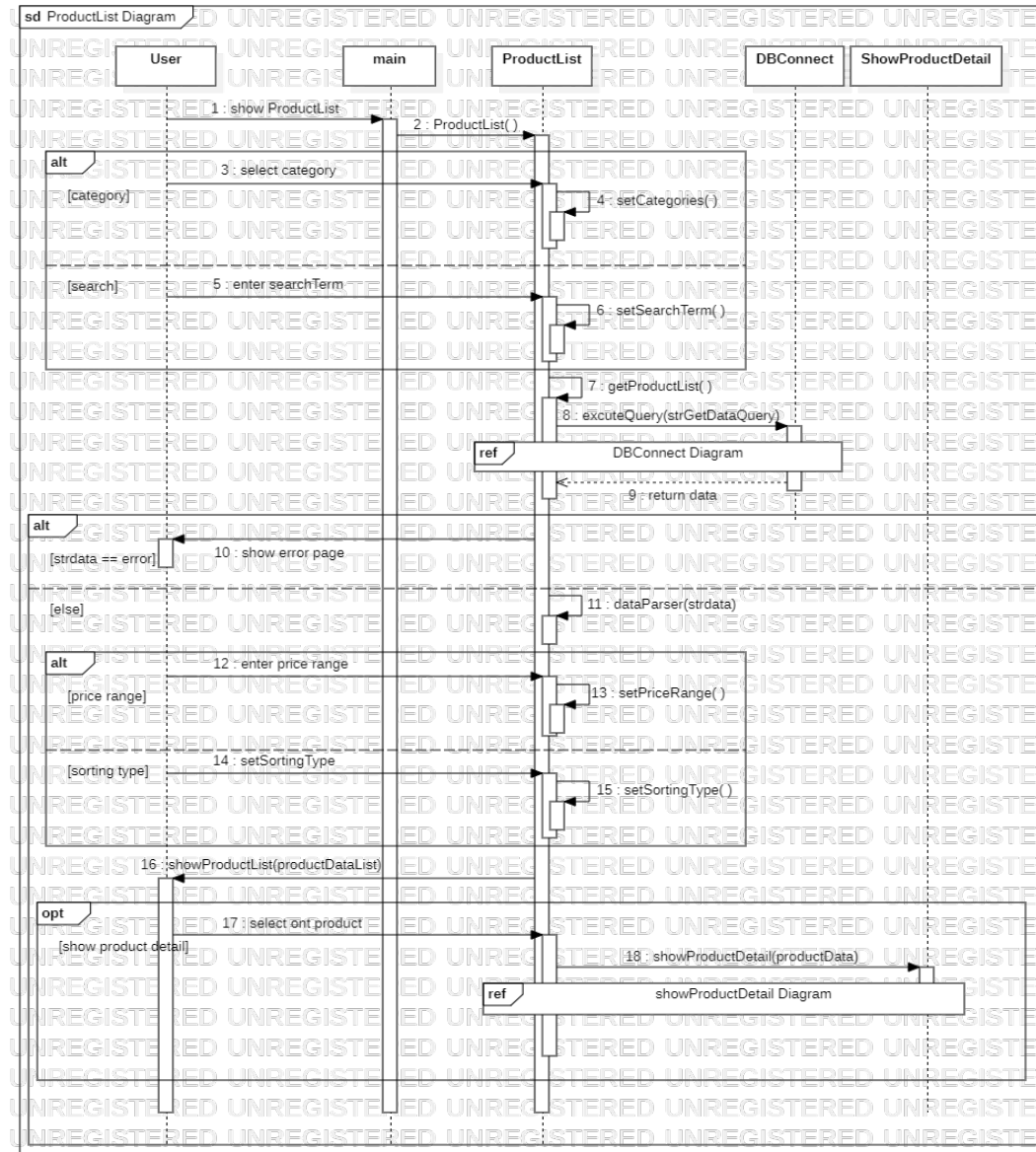
PaymentInfo : Payment에서 struct구조체와 같은 방식으로 사용하는 클래스이며, 아래 값들을 포함한다.

1. strProductName
 - 선물하기를 할 상품의 이름값이다.
2. strProductImage
 - 선물하기를 할 상품의 이미지이다. URL 형식으로 사용하기 위해 String 값을 가진다
4. nProductPrice
 - 선물하기를 할 상품의 가격이다.
5. nShoppingPoint
 - 유저가 가진 쇼핑 포인트의 값이다.

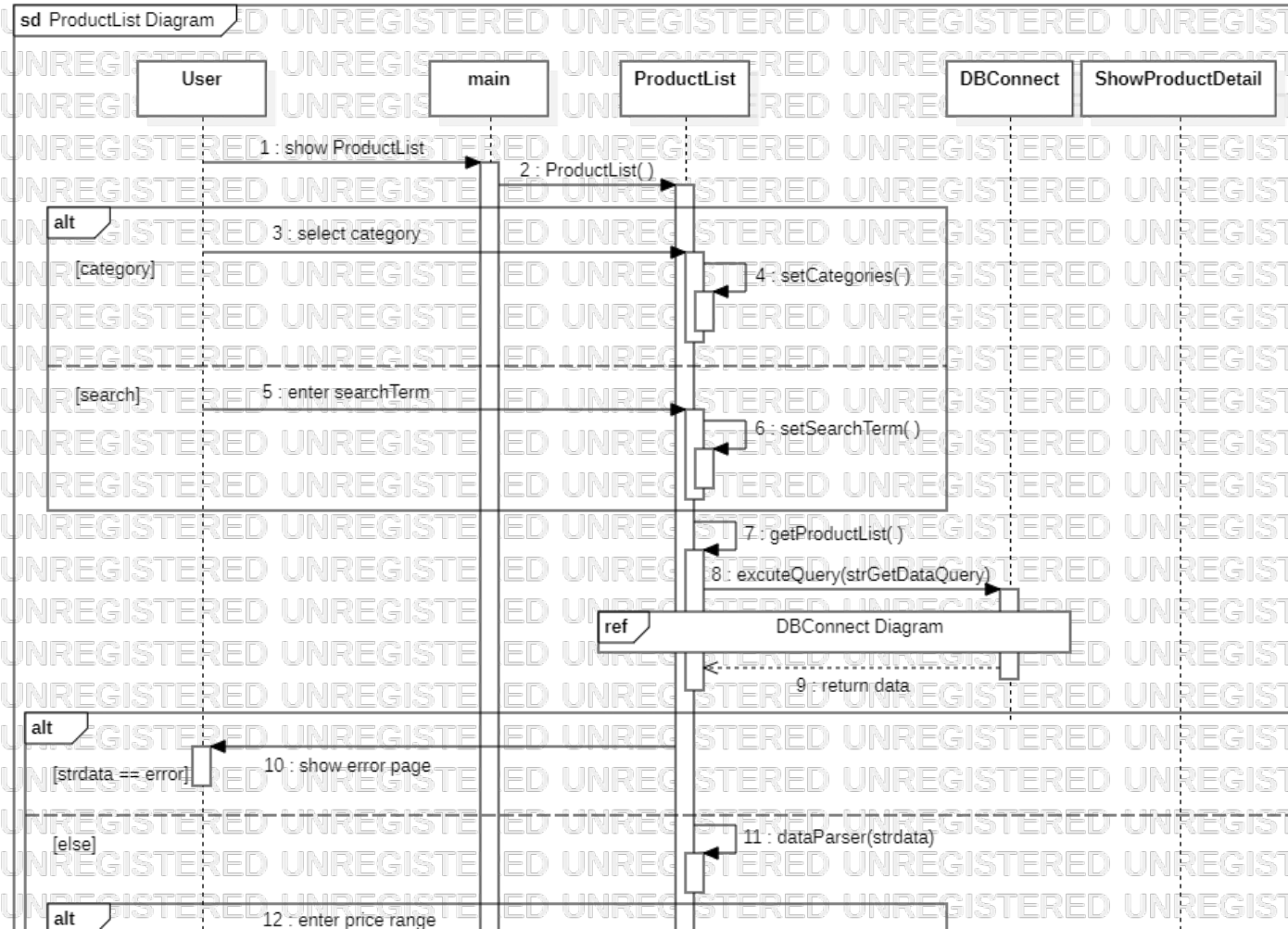
07

순차 다이어그램

07. 순차 다이어그램 - U1



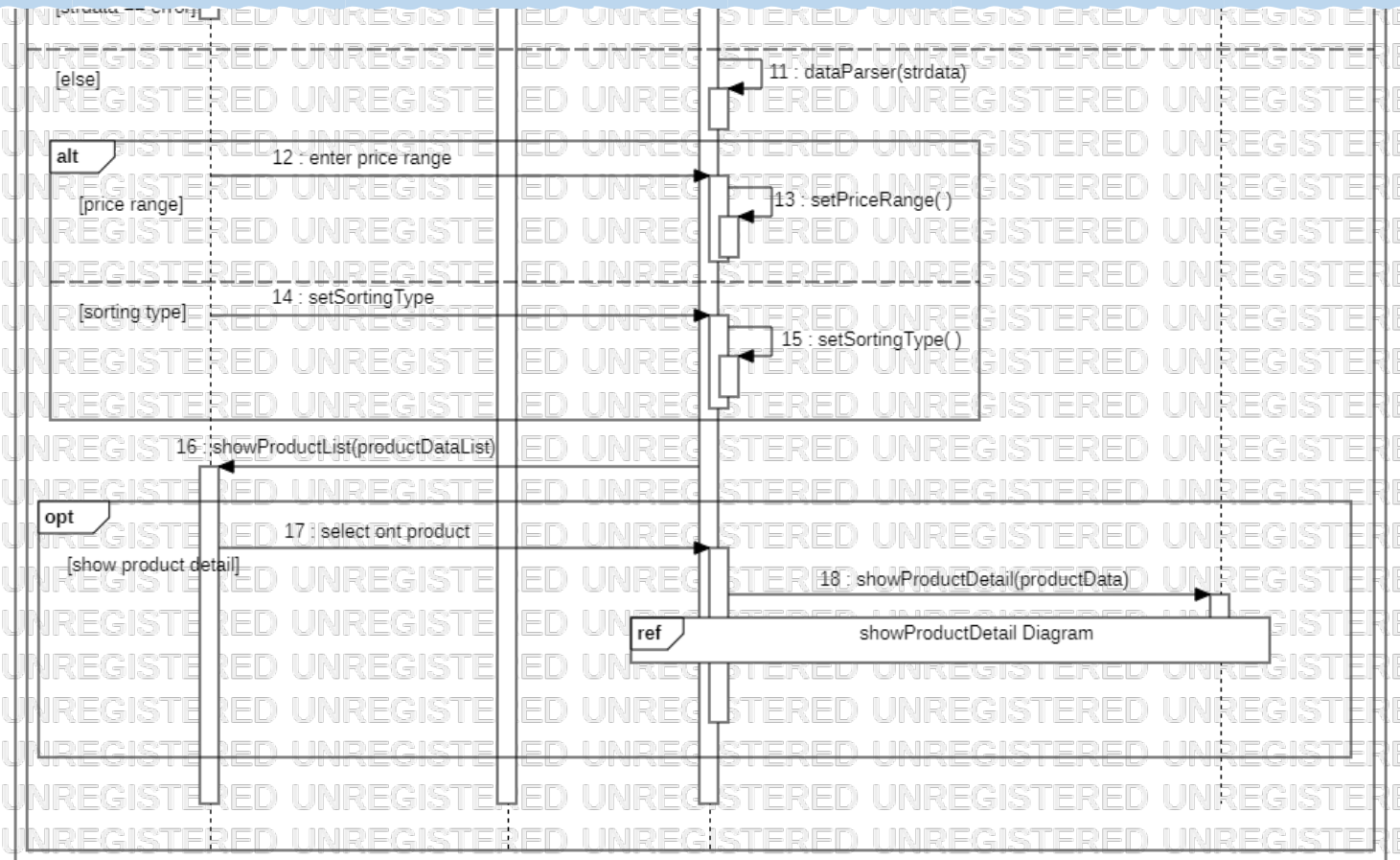
07. 순차 다이어그램 - U1



1. 사용자가 상품목록을 보고자 할 때 다음과 같은 순서로 실행된다.
2. main객체에서 ProductList객체를 생성한다.
3. 만약 사용자가 카테고리를 선택하면
4. setCategories() 함수를 호출하여 변수를 초기화한다.
5. 만약 사용자가 검색어를 선택하면
6. setSearchTerm() 함수를 호출하여 변수를 초기화한다.
7. 데이터베이스로부터 상품 리스트를 가져오는 getProductList() 함수를 호출한다.
8. DBConnect 클래스에 excuteQuery()함수로 디비에 쿼리를 보낸다.
9. 쿼리의 결과를 strdata에 저장한다.
10. 만약 strdata가 error라면 에러페이지를 출력한다.
11. 그렇지 않다면 data를 productData List형태로 변환한다.

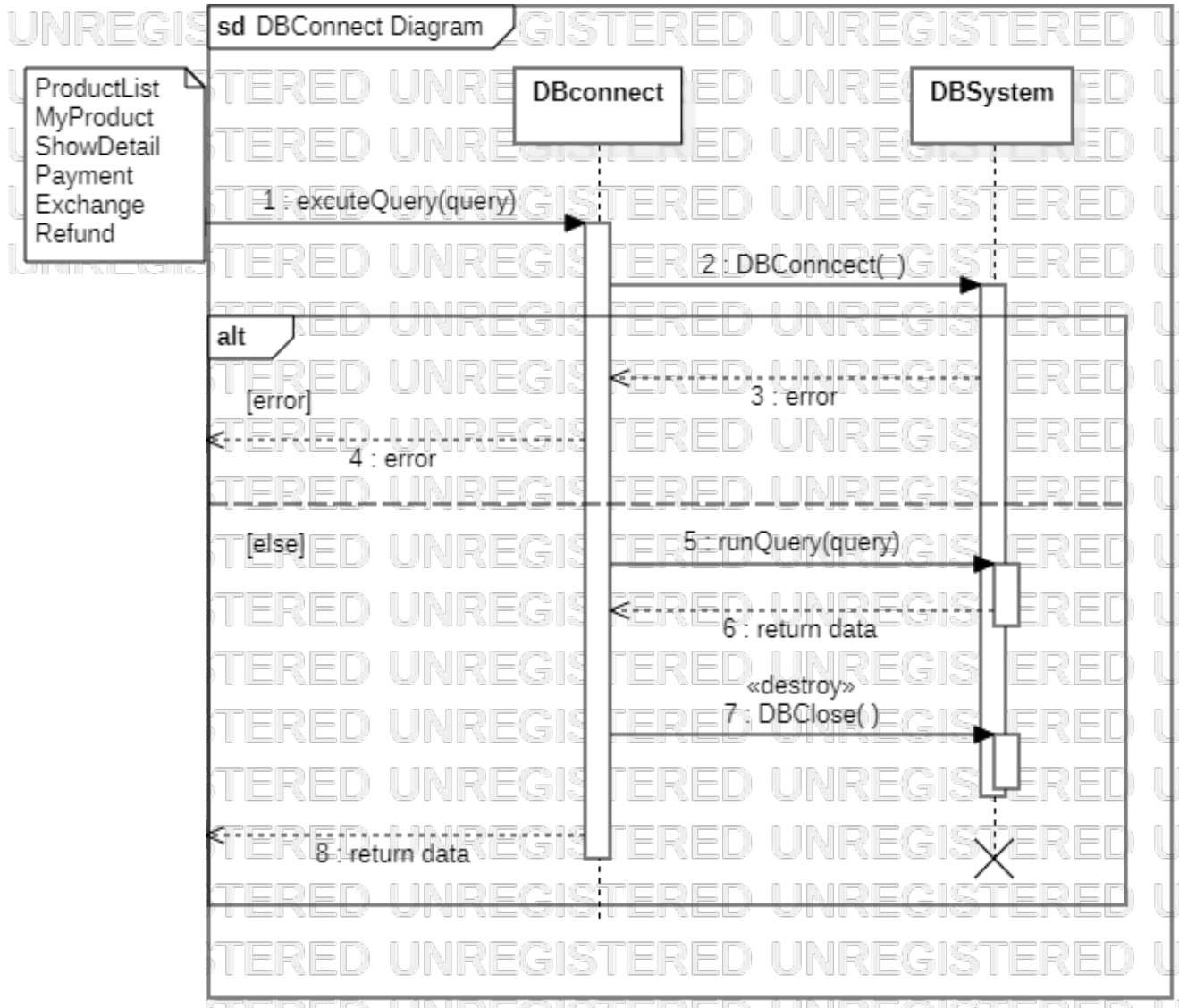
다음페이지...

07. 순차 다이어그램 - U1



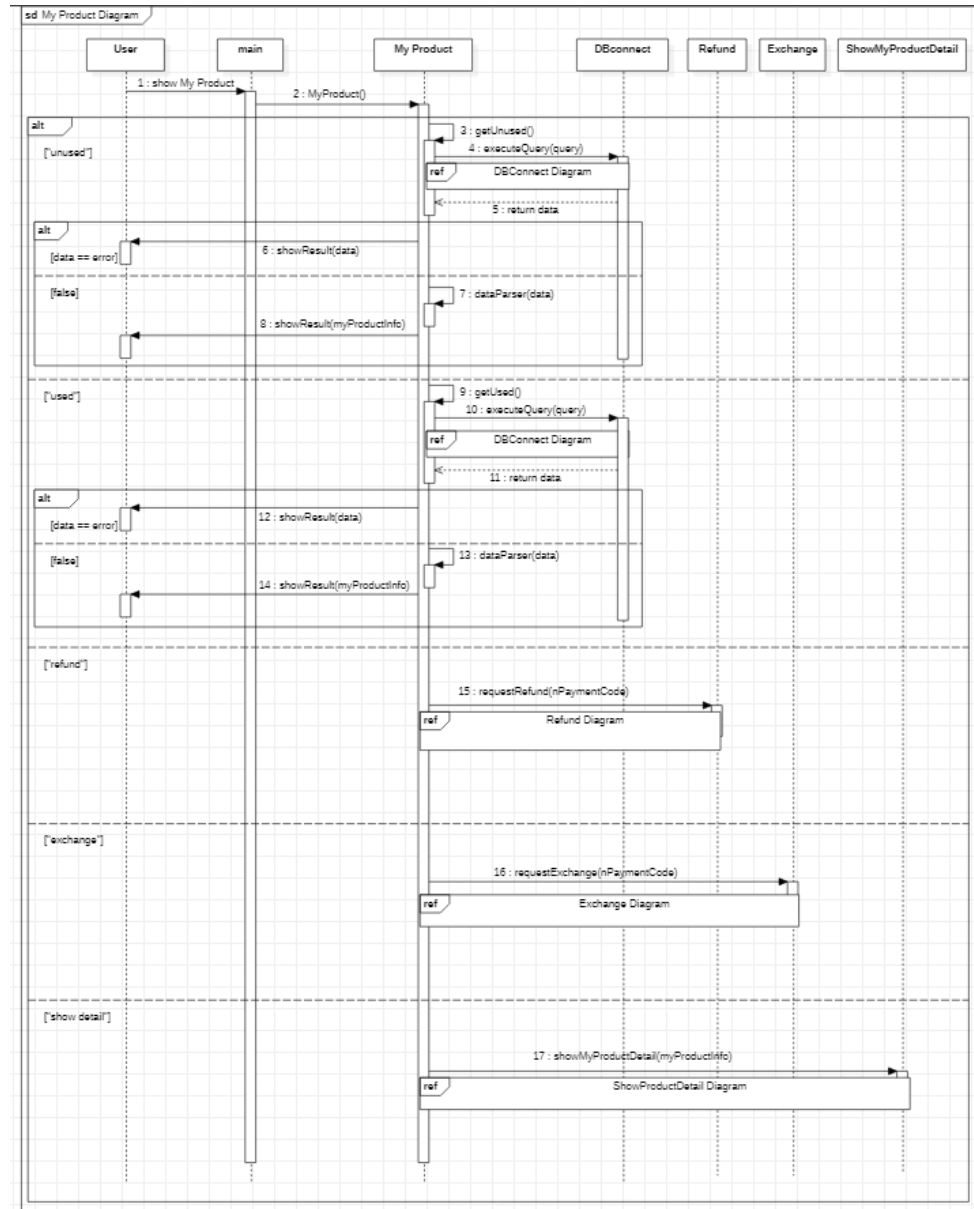
12. 만약 사용자가 가격대를 입력하면
13. setPriceRange()함수를 호출하여 변수를 초기화하고, 리스트를 초기화한다.
14. 만약 사용자가 정렬기준을 선택하면
15. setSortingType()함수를 호출하여 변수를 초기화하고, 리스트를 초기화한다.
16. showProductList()함수를 호출하여 사용자에게 리스트를 출력한다.
17. 만약 사용자가 한 개의 상품을 선택한다면
18. showProductDetail 함수를 실행한다.

07. 순차 다이어그램 - U2

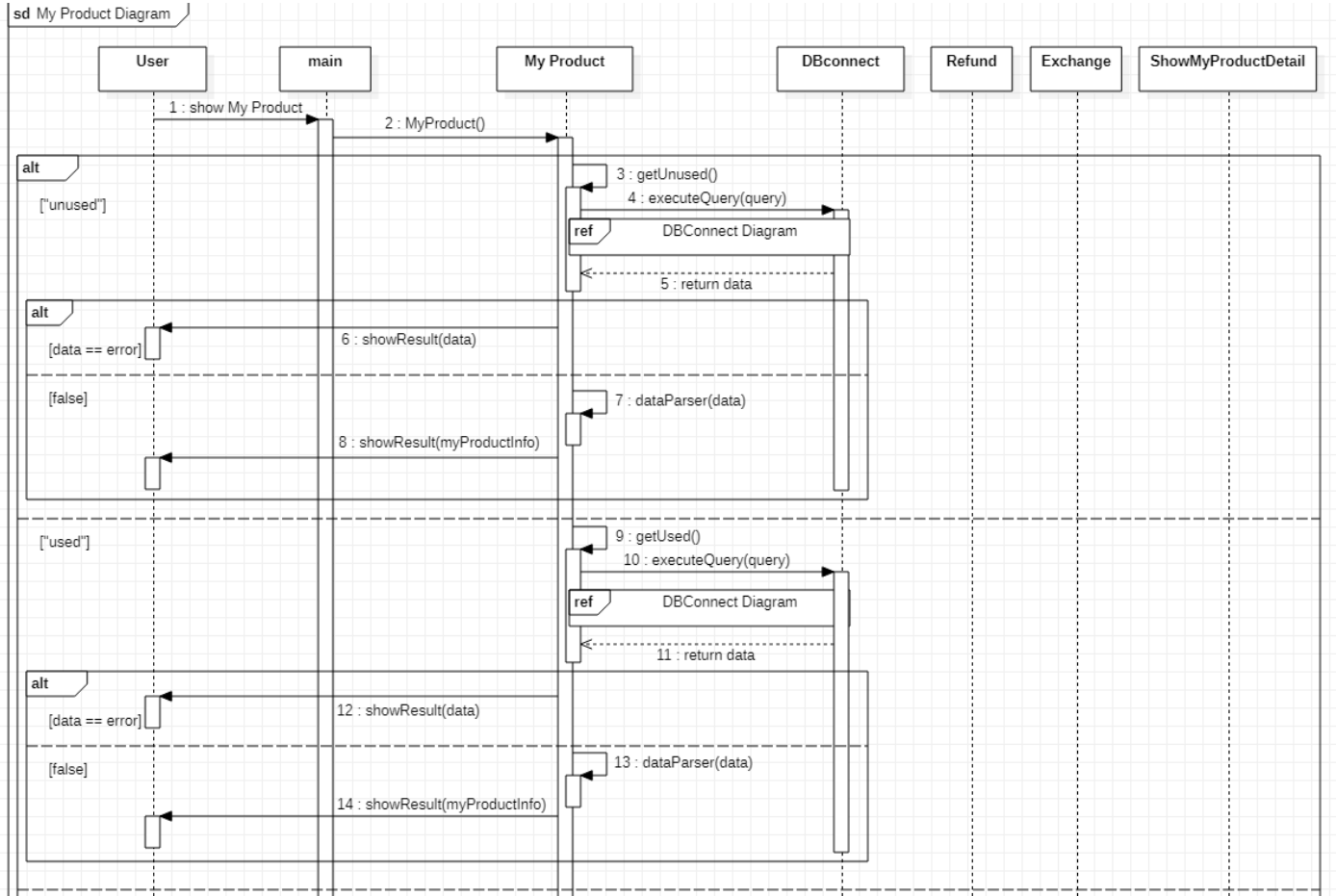


1. DBconnect 클래스에 접근하여 excuteQuery()메소드를 실행하면 다음과 같은 순서로 실행된다.
2. DBConnnect() 함수를 호출하여 DB시스템에 연결한다.
3. 만약 DB시스템 연결에 실패할 경우
4. 에러를 반환한다.
5. 성공한다면, runQuery() 함수를 호출하여 DB에 쿼리문을 전송한다.
6. 쿼리문의 결과를 받아온다.
7. DBClose() 함수를 호출하여 DB시스템의 연결을 닫는다.
8. 결과를 string형태로 반환한다.

07. 순차 다이어그램 – U3



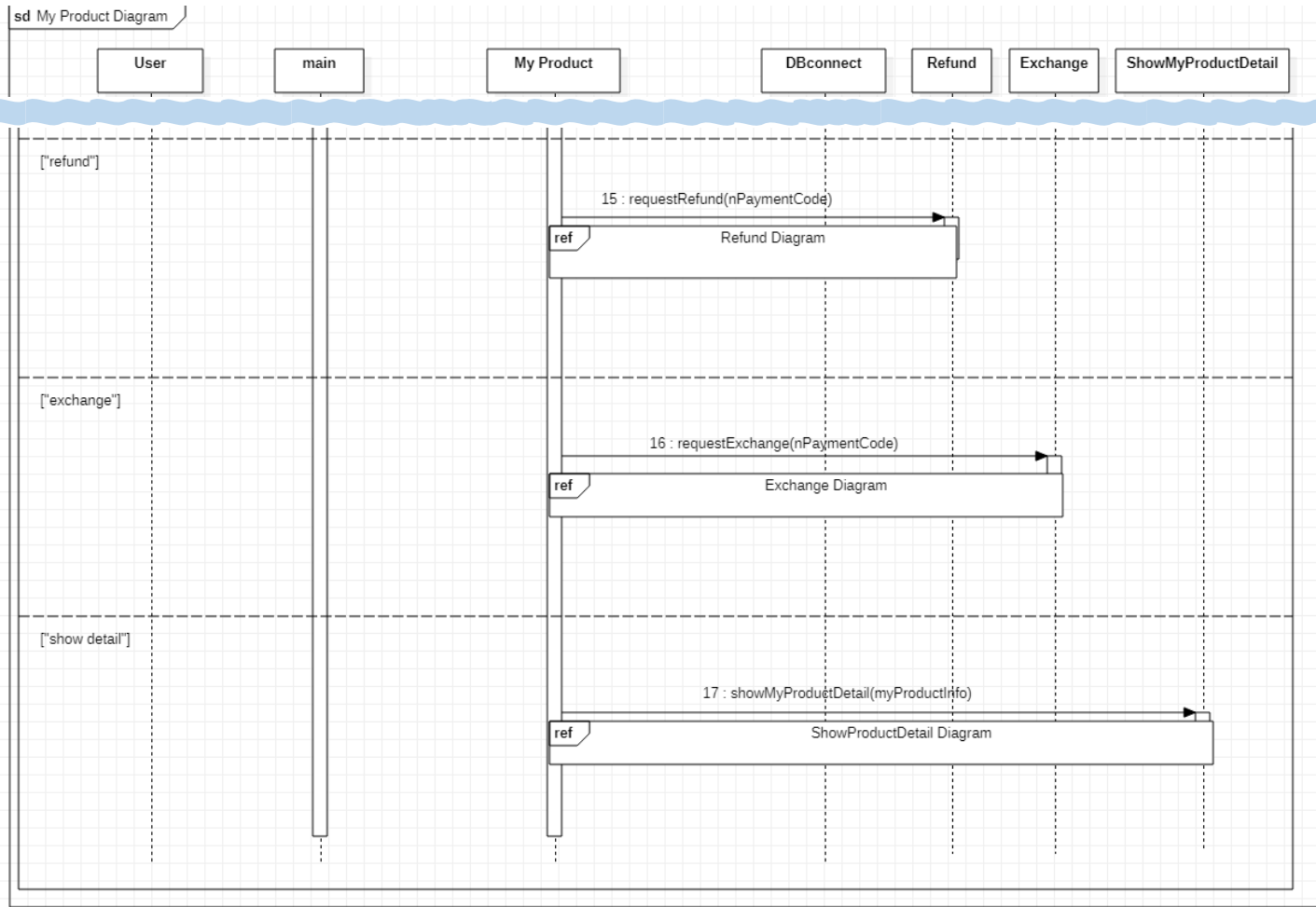
07. 순차 다이어그램 - U3



1. 사용자가 선물함을 클릭한다.
2. main 객체에서 선물함을 클릭하면 선물함 페이지로 이동한다.
3. 만약 사용가능한 선물 메뉴를 선택 한다면,
4. 사용가능한 선물 정보를 위한 쿼리를 실행한다.
5. DB로 부터 리턴값을 받아 온다.
6. 만약 DB의 리턴값이 error 일 경우, showResult 메서드로 사용자에게 결과를 보여준다.
7. 아닐경우, DB의 리턴값을 dataParser 메서드로 분석한다.
8. showResult 메서드로 사용자에게 결과를 보여준다.
9. 만약 사용했던 선물 메뉴를 선택 한다면,
10. 사용했던 선물 정보를 위한 쿼리를 실행한다.
11. DB로 부터 리턴값을 받아 온다.
12. 만약 DB의 리턴값이 error 일 경우, showResult 메서드로 사용자에게 결과를 보여준다.
13. 아닐경우, DB의 리턴값을 dataParser 메서드로 분석한다.
14. showResult 메서드로 사용자에게 결과를 보여준다.

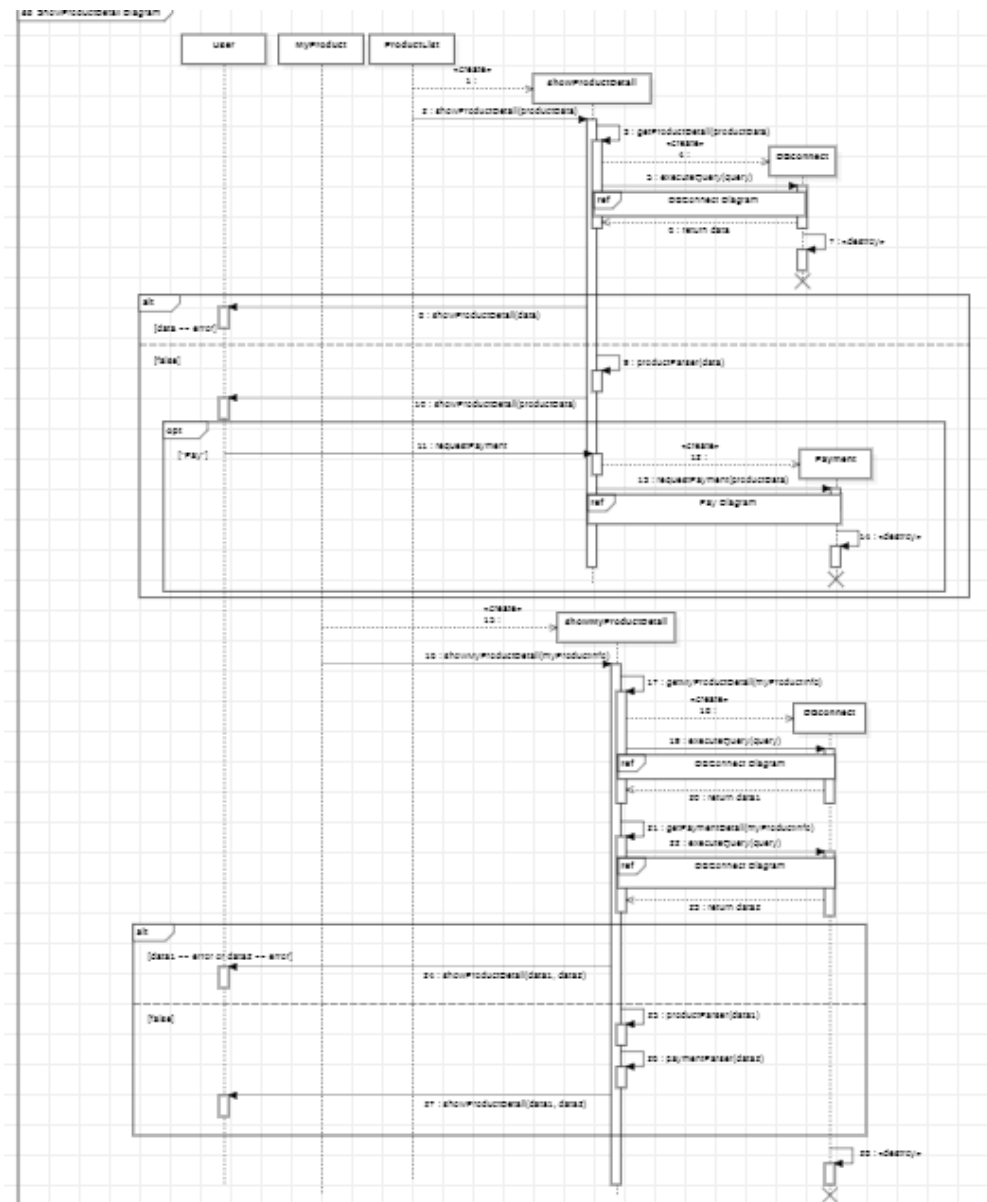
다음 페이지...

07. 순차 다이어그램 – U3

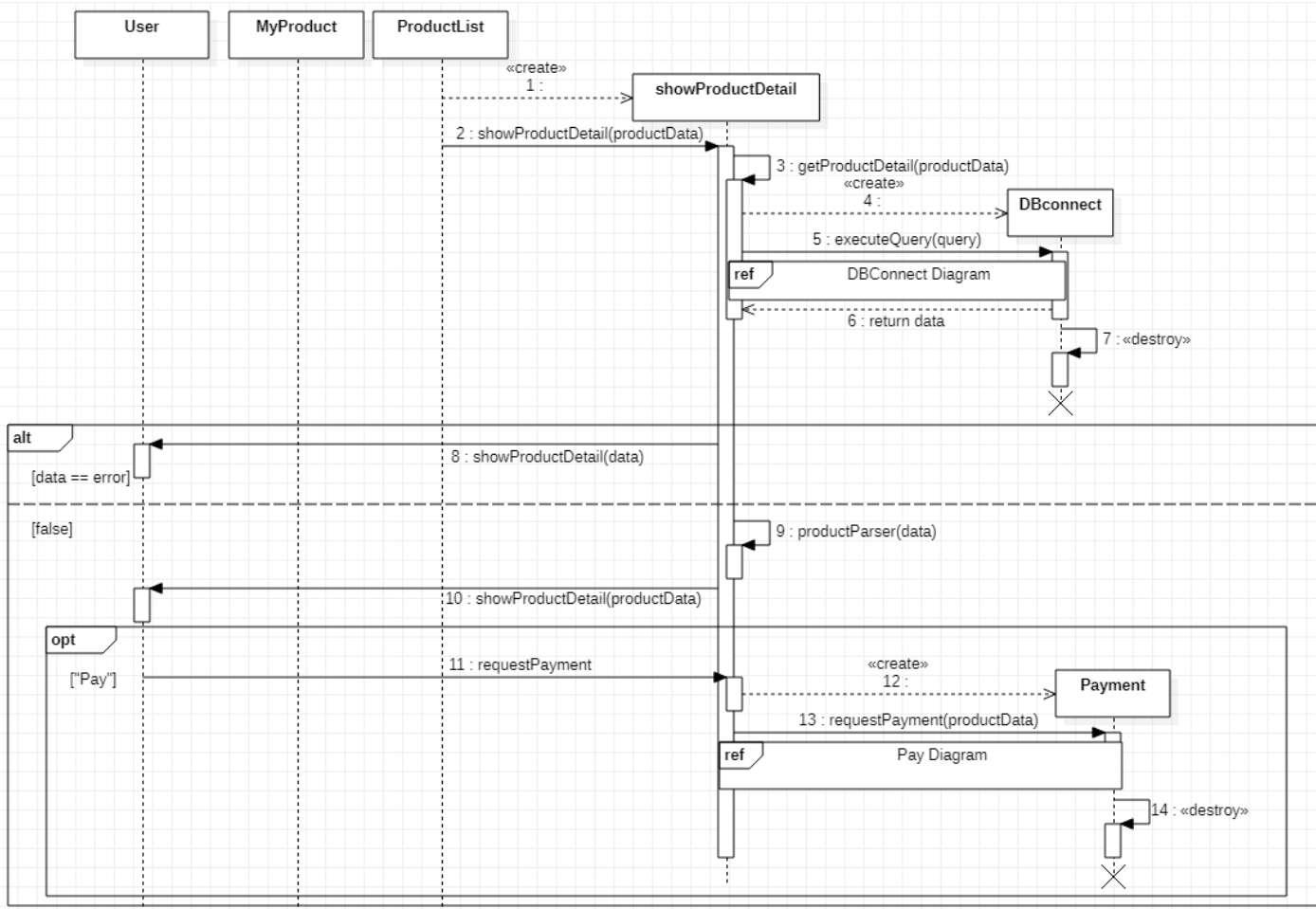


15. 만약 환불 메뉴를 선택 한다면, requestRefund 메서드를 호출하여 환불 페이지로 이동한다.
16. 만약 교환 메뉴를 선택 한다면, requestExchange 메서드를 호출하여 교환 페이지로 이동한다.
17. 만약 상세보기 선택 한다면, ShowMyProductDetail 메서드를 호출하여 상세보기 페이지로 이동한다.

07. 순차 다이어그램 - U4



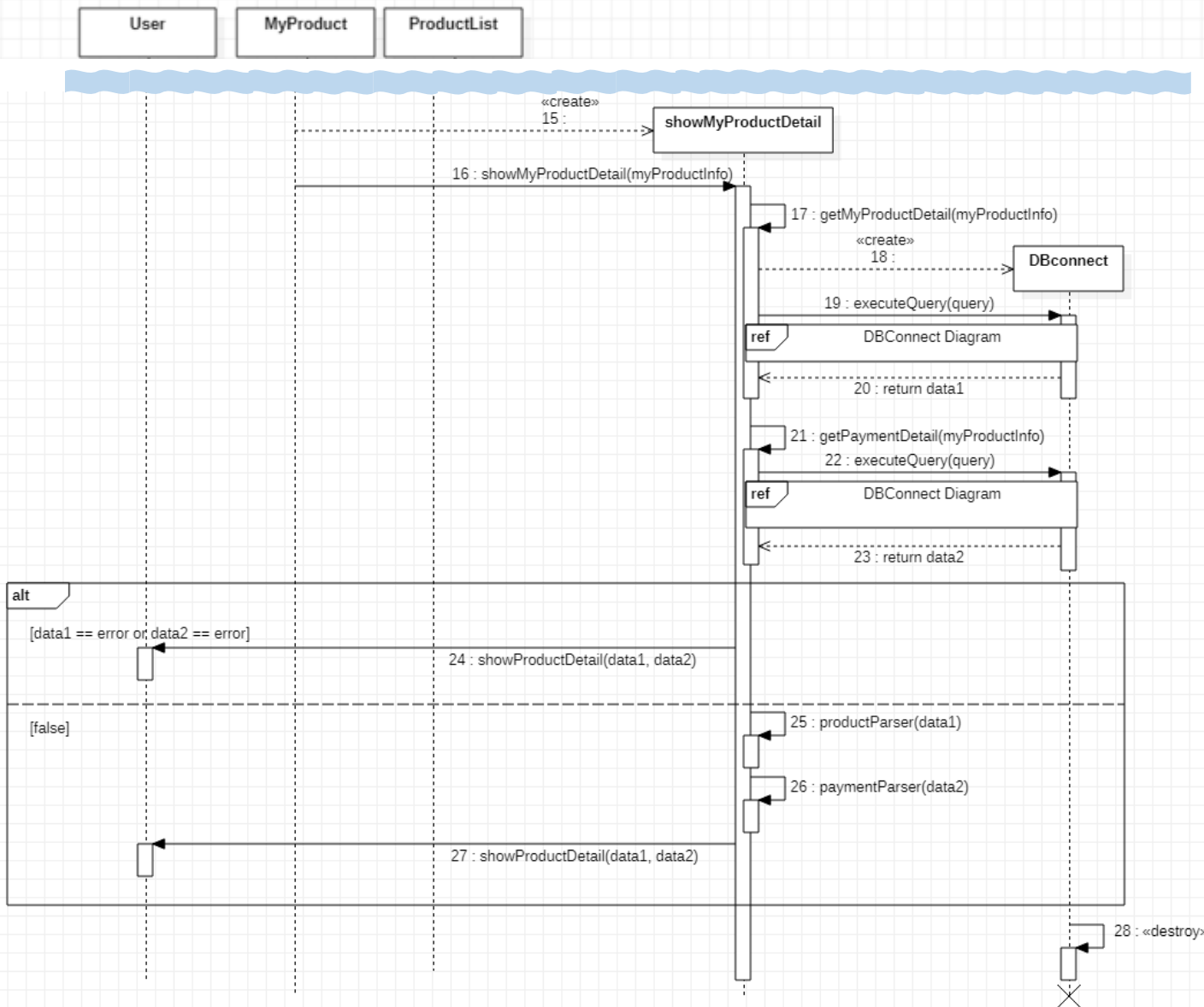
07. 순차 다이어그램 - U4



1. ProductList 객체에서 showProductDetail 객체를 생성한다.
2. showProductDetail 메서드를 호출하여 페이지를 이동한다.
3. 상품 상세 정보를 가져오기 위해
4. DBconnect 객체를 생성하여
5. DB에 쿼리를 실행한다.
6. 리턴값을 받아 온다.
7. 객체가 끝나면 소멸시킨다.
8. 만약, DB의 리턴값이 error 이면, 사용자에게 결과를 보여준다.
9. 아닐 경우, DB의 리턴값을 분석한다.
10. 사용자에게 결과를 보여준다.
11. 만약, 사용자가 결제를 요청했다면
12. Payment 객체를 생성하여
13. 결제 요청 페이지로 이동한다.
14. 객체가 끝나면 소멸시킨다.

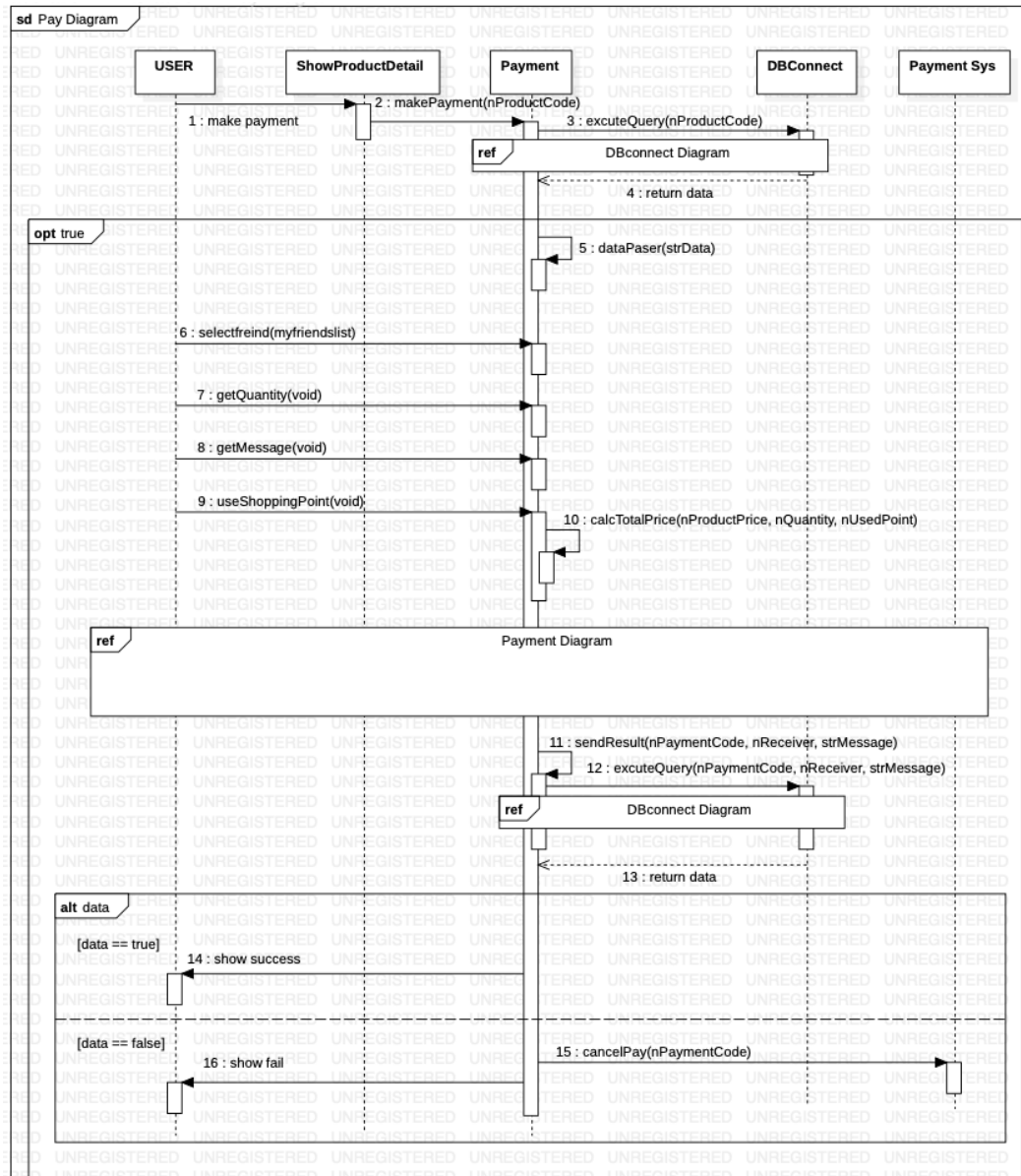
다음 페이지...

07. 순차 다이어그램 - U4



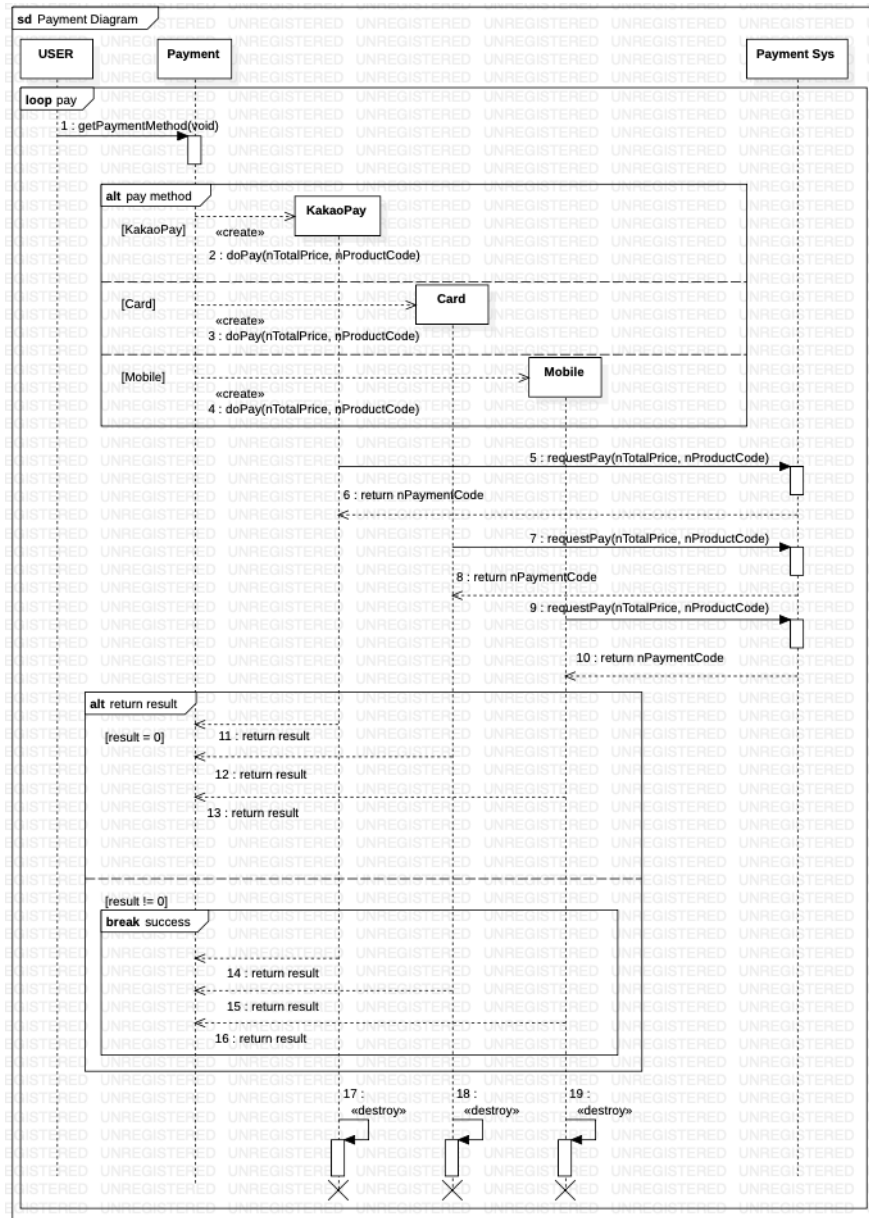
15. MyProduct 객체에서 showMyProductDetail 객체를 생성한다.
16. showMyProductDetail 메서드를 호출하여 페이지를 이동한다.
17. 상품의 정보를 가져오기 위해
18. DBconnect 객체를 생성하고
19. DB에 쿼리를 실행한다.
20. DB로부터 리턴값을 받아온다.
21. 결제 정보를 가져오기 위해
22. DB에 쿼리를 실행한다.
23. DB로부터 리턴값을 받아온다.
24. 만약, 두개의 리턴값 중 한개라도 error가 발생한다면, 사용자에게 결과를 보여준다.
25. 아닐 경우, 리턴 값을 MyProductInfo 타입으로 파싱한다.
26. 리턴 값을 PaymentInfo 타입으로 파싱한다.
27. 사용자에게 결과를 보여준다.
28. 객체가 끝나면 소멸시킨다.

07. 순차 다이어그램 – U5



1. 유저가 결제를 요청한다.
2. 해당 상품의 상품 코드를 넘겨 받는다.
3. 쿼리를 실행한다.
4. 리턴 값을 받아온다.
5. 정상적인 데이터를 받았다면 데이터를 가공한다.
6. 친구목록에서 친구를 선택한다.
7. 수량을 입력한다.
8. 메시지카드를 사용자가 입력한다.
9. 쇼핑 포인트를 사용자가 입력한다.
10. 최종 결제 금액을 표시한다.
11. 받는 사람에게 메시지를 전송한다.
12. 쿼리를 실행한다.
13. 리턴 값을 받아온다.
14. 리턴 받은 데이터가 true이면 결제를 성공했음을 메시지를 출력한다.
15. 리턴 받은 데이터가 false이면 결제를 취소한다.
16. 결제를 실패했음을 메시지로 출력한다.

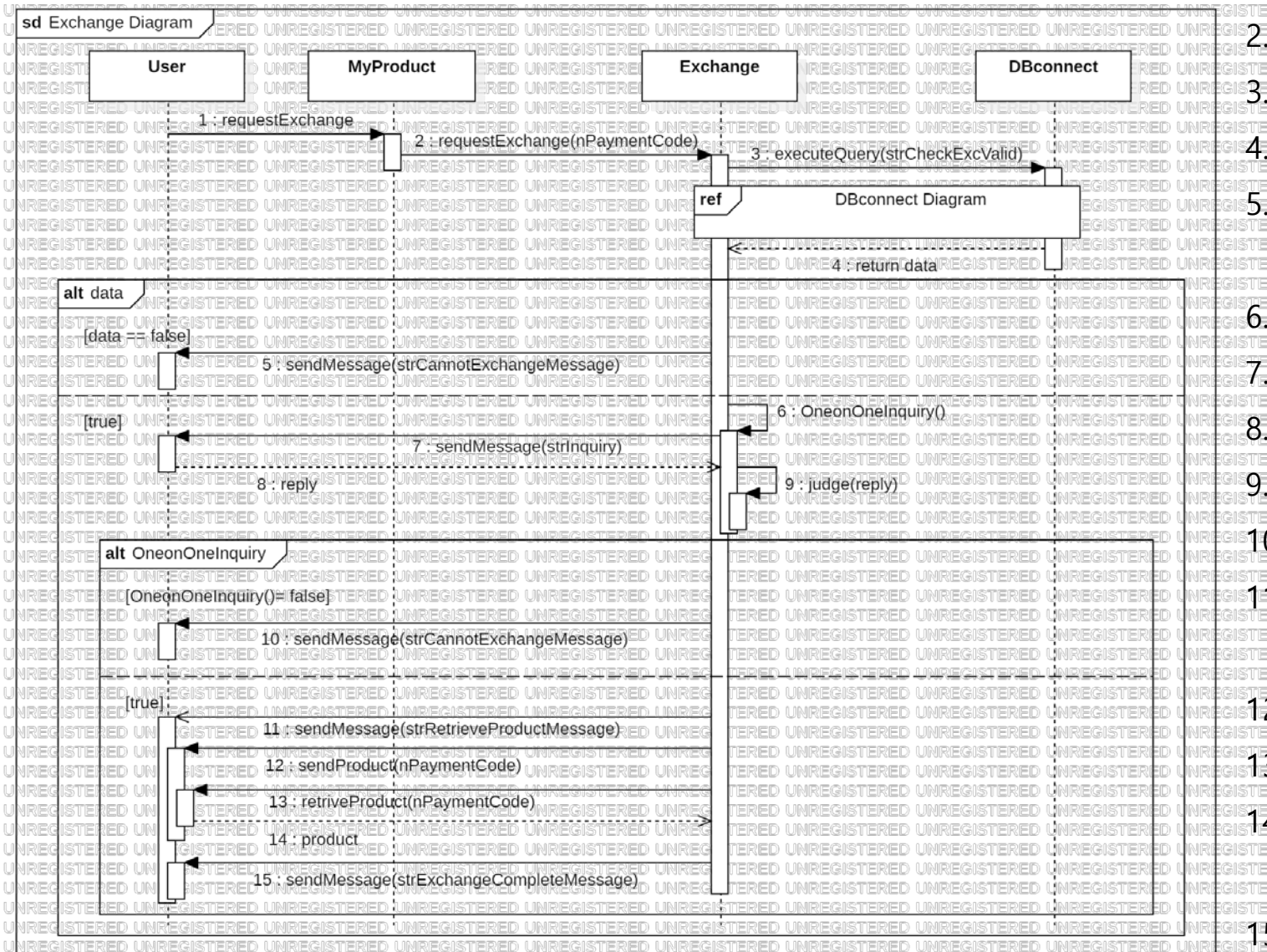
07. 순차 다이어그램 – U6



1. 결제 수단을 입력한다.
2. ~ 4. 생성자를 통해 결제 수단에 맞는 클래스를 생성한다.
5. 7. 9. 외부 결제 시스템에 결제를 요청한다.
6. 8. 10. 결제 결과를 받는다.
11. ~ 13. 결제를 실패했다면 계속 반복문을 진행한다.
14. ~ 16. 결제를 성공했다면 반복문을 나오고 결제 코드를 반환한다.
17. ~ 19. 다음 반복문으로 진행되거나 종료될 때 마다 생성된 클래스는 제거된다.

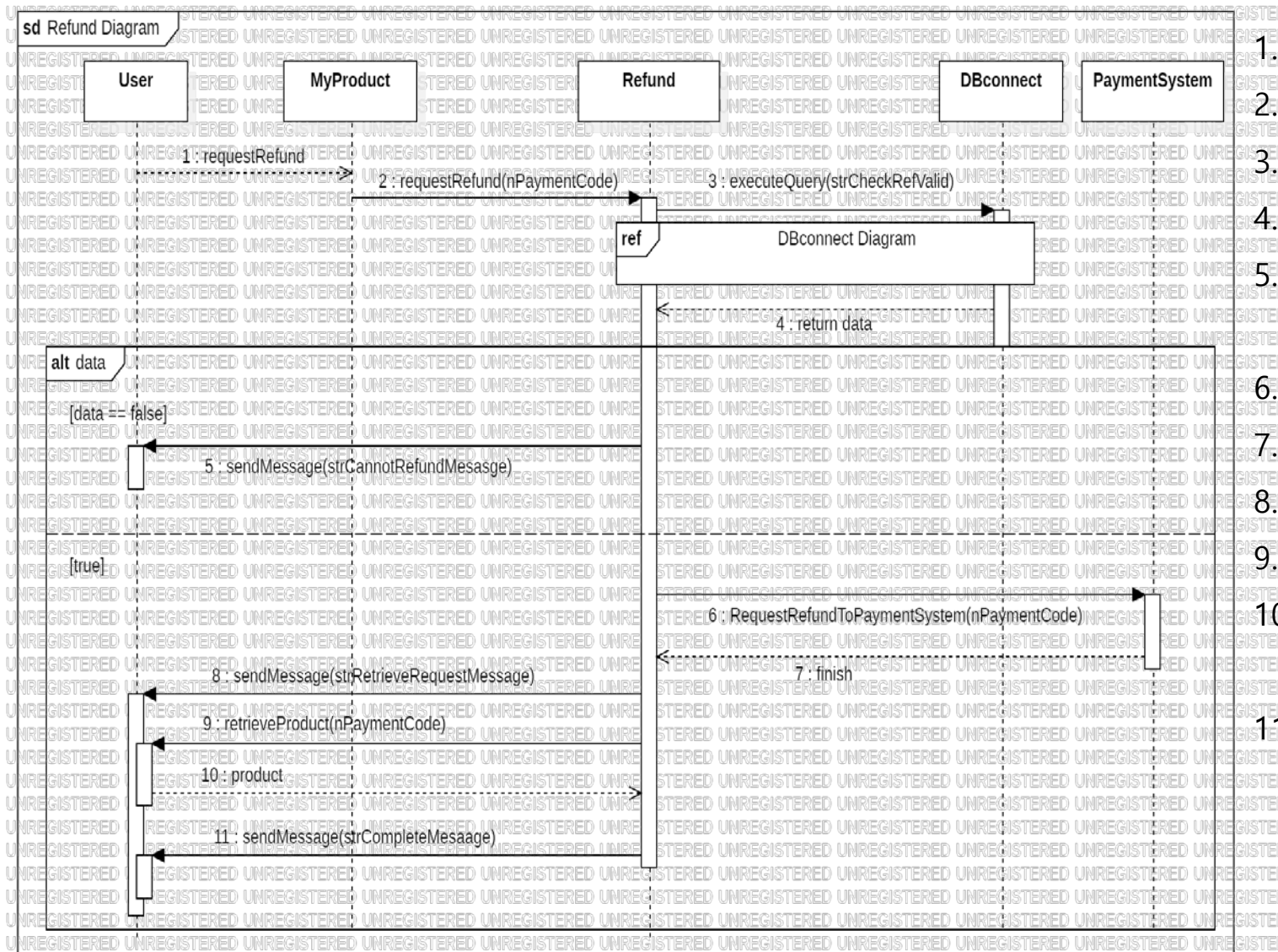
*Payment Sys는 외부 시스템으로 KakaoPay, Card, Mobile 모두 별도의 외부 시스템이 각각 존재해야하나 가독성을 위해 하나로 통일하였다.
(클래스 다이어그램 참조)

07. 순차 다이어그램 - U7



1. 사용자가 교환을 요청한다.
2. 선물함에서 **교환**을 실행한다.
3. DBconnect의 메소드, 유효성 검사를 진행한다.
4. string 데이터를 리턴받고, 이는 "true" 또는 "false"이다.
5. data == false 일 경우 잘못된 요청으로 교환 불가 메시지를 발송한다.
6. true일 경우 1:1 문의를 진행한다.
7. 문의를 위해 유저에게 메시지 발송한다.
8. 유저의 답변을 입력받아서, (scanf 등)
9. 응답으로 교환 여부를 판별한다.
10. 교환 진행이 불가능할 경우 교환 불가 메시지를 발송한다.
11. 교환이 가능한 경우, 고객에게 상품 반환 메시지를 전송한다.
12. 새 제품을 발송한다.
13. 고객에게 제품을 회수한다.
14. 고객이 제품을 보내주면(택배 등으로 물건을 받을 경우) Boolean 변수를 true로 바꿔 주는 방식이다.
15. 교환 절차가 끝났다는 메시지를 발송한다.

07. 순차 다이어그램 - U8



1. 사용자가 환불을 요청한다.
2. 선물함에서 환불을 실행한다.
3. DBconnect의 메소드, 유효성 검사한다.
4. string 데이터를 리턴받고, 이는 "true" 또는 "false"이다.
5. data == false 일 경우 유저에게 잘못된 요청으로 환불이 불가능하다고 메세지를 보낸다.
6. true 일 경우 결제 시스템에 환불을 요청한다.
7. 외부 시스템에서 환불이 끝나면
8. 고객에게 상품 반환 요청 메세지를 보내고
9. 상품 반환 절차를 진행한다.
10. 고객이 상품을 돌려 보내면(택배 등으로 물건을 받을 경우) Boolean 변수를 true로 바꿔 주는 방식
11. 환불 절차가 끝났다는 메세지를 유저에게 보낸다.

Q&A