

## SDD Subgroup2 : Auction & Chat

조원 : 최승원, 고재원

Subgroup2의 최상위 30%의 요구 조건은 논의 결과 핵심기능인 **UC-11 (경매하기)**, **UC-12(채팅하기)**로 결정하였으며, 이에 대한 sequence diagram과 class diagram을 작성하기로 하였습니다.

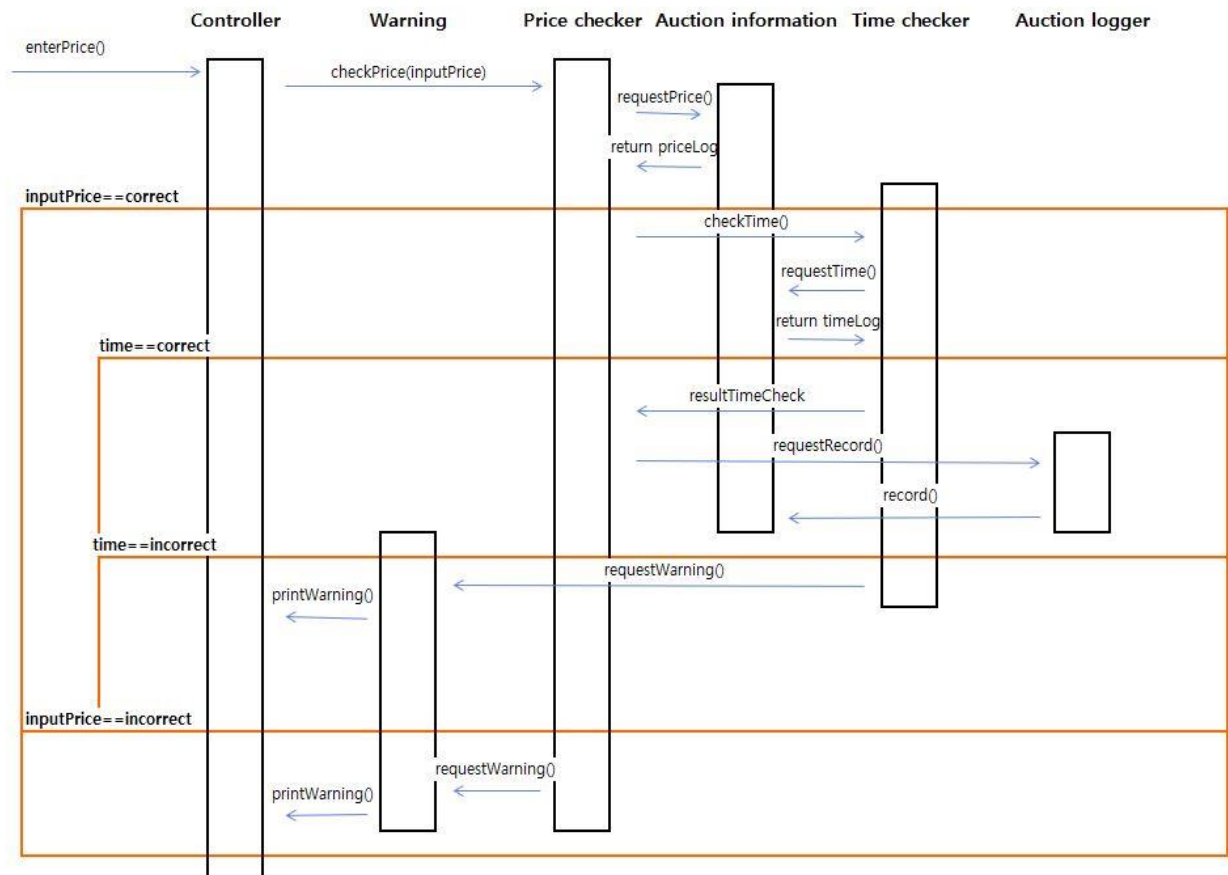
### 목차

1. 초기 Sequence Diagrams
2. Variation 및 이에 대한 논의 정리
3. Sequence diagram 최종 결과
4. Class diagram

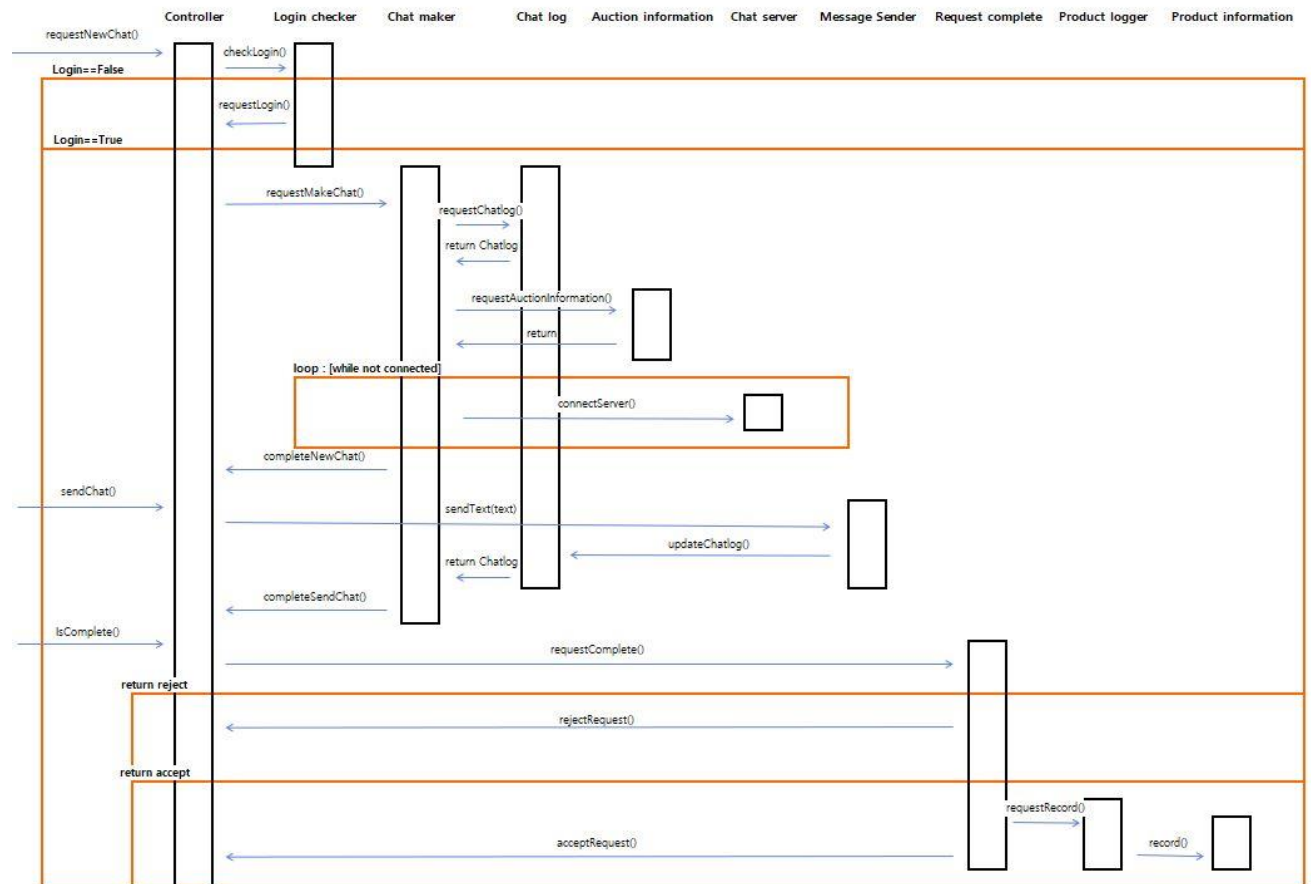
## 1. 초기 Sequence Diagram

### 1-1. Sequence Diagram 1

[UC- 11]

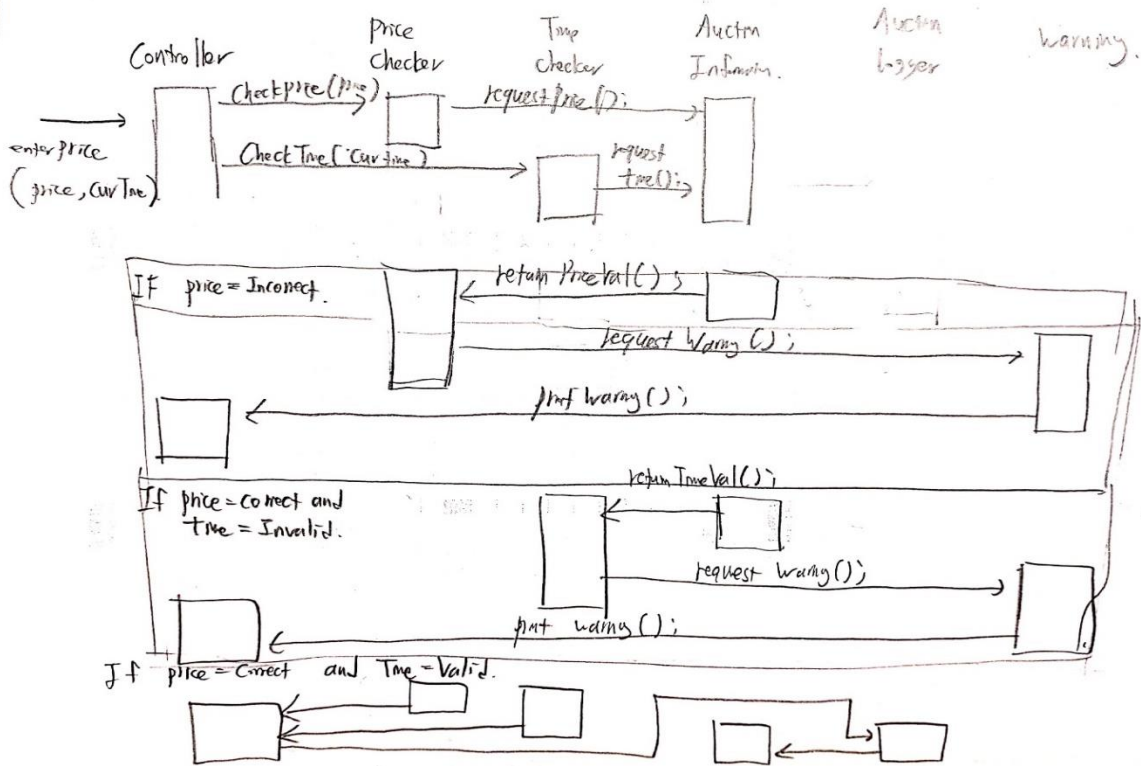


# [UC – 12]

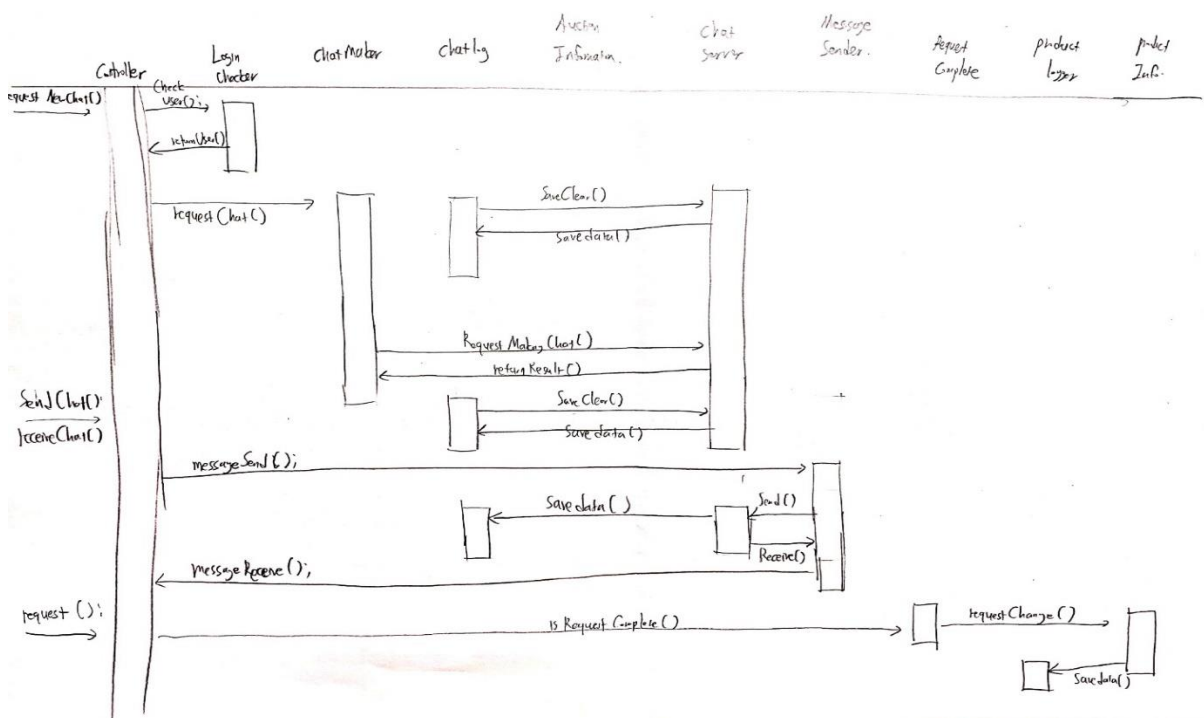


## 1-2. 초기 Sequence Diagram 2

[UC-11]



[UC-12]



## 2. Variation 및 이에 대한 논의

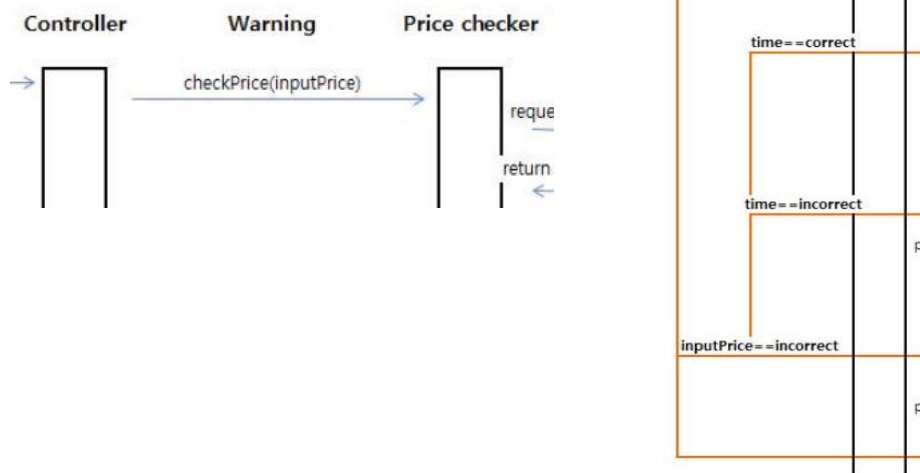
### 2-1. [UC-11]에 대한 논의

Q1) Controller가 모든 sequence를 컨트롤 하는 것과 요청을 하고 나머지 sub doer가 요청을 수행해야 하는가?

A) 논의 결과, Controller가 모든 sequence를 수행하기보다는, 각자의 역할을 수행할 수 있는 Checker를 만들었으므로 Checker들에게 이러한 역할을 맡기는 것이 좋다고 생각하여 Controller는 그저 요청을 보낼 뿐이고, Checker들이 알아서 명령을 실행하고 결과를 도출하도록 하였다.

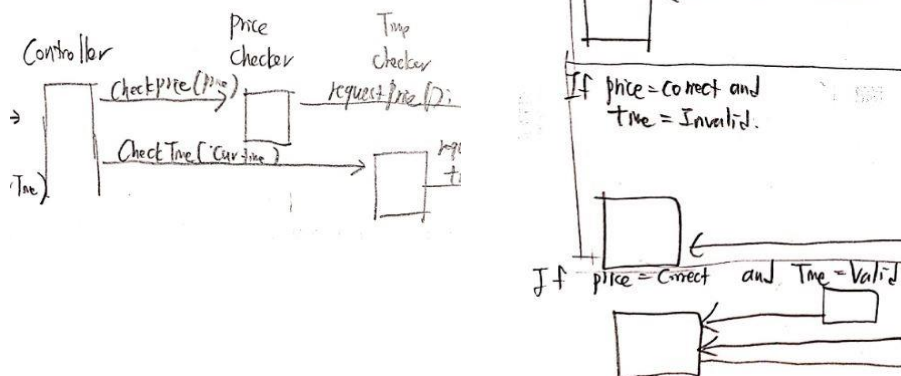
Q2) 경매를 할 때 요구조건에 따라 1)금액이 현재 경매 가격보다 높아야 하며, 2) 다른 사용자가 금액을 입찰한 후 몇 분 동안은 입찰을 할 수 없어야 한다. 따라서 경매 가격, 입찰 시간을 변수로 받아서 연산을 한다. 이때, 최승원 조원의 방법(**Variation 1**)은 먼저 가격이 적당한지 검사 후, 입찰할 수 있는 시간인지 검사를 하는 반면, 고재원 조원의 경우(**Variation 2**) 두 변수의 값이 적당한지 둘 다 검사 후 경우를 나누어 입찰 여부를 판단하였다. 이 두 방법 중 어떤 방법을 선택해야 하는가?

### < Variation 1 >



설명 : 가격을 먼저 검사하고, 옳은 값이 맞는지 아닌 지에 따라 시간을 검사하여 입찰 여부를 결정한다.

### < Variation 2 >



설명 : 가격과 시간을 모두 검사 후, 두 값의 옳고 그름에 따라 입찰여부를 결정한다.

A) 논의 결과, 가격이 올바르지 않으면 입찰 시간에 관계없이 입찰이 불가하기 때문에 입찰 시간이 올바른 지 검사할 필요가 없으므로 가격을 먼저 검사하는 방법을 선택할 시, 시간 복잡도를 고려하였을 때, 시간/가격 두개를 모두 검사하는 **Variation 2** 의 방법 **Variation 1**의 방법을 채택하기로 하였다.

## 2-2. [UC-12]에 대한 논의

Q) Chat Maker, Chat Sender 등 Sequence 적으로 기능이 모호한 것 들이 있다. 또한 이름이 그것의 내용을 충분히 담지 못하는 것 같다. (ex: Auction Information)

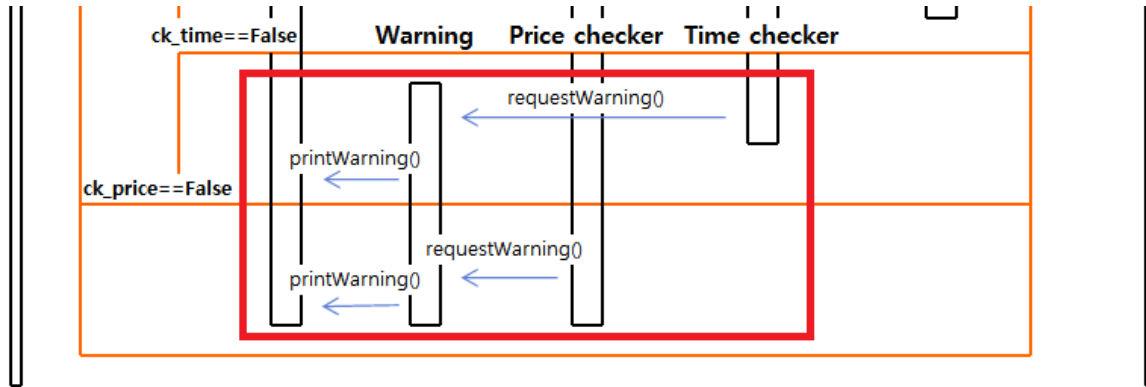
A) 논의 결과,

1. Chat Maker는 단지 채팅방을 개설하는 역할만을 한다.
2. Chat Sender -> Chat Manager로 명칭을 바꾸고, 이것이 채팅을 보내거나 (send), 받거나(receive) 하는 것을 책임지도록 한다.
3. Auction Information -> Auction DB로 명칭을 바꾸어 경매에 대한 데이터를 담고 있음을 표현할 수 있도록 한다.
4. 추가로 기타 이름 및 함수를 변경하여 Sequence를 이해하기 쉽도록 한다.

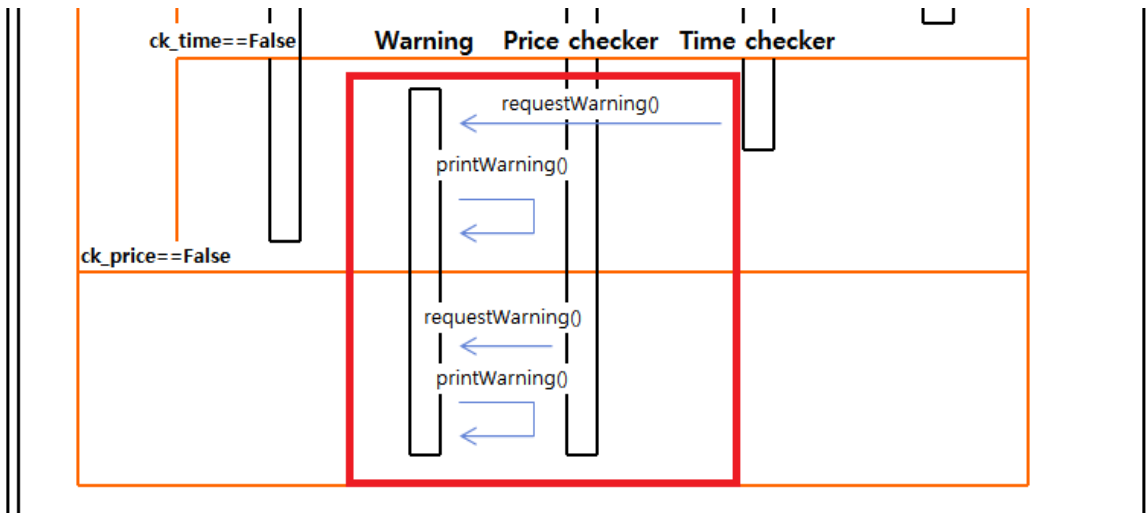
### 2-3. [UC-11], [UC-12] 모두 해당되는 것에 대한 논의

Q1) 메시지 출력의 주체 : Controller vs Warning Concept

1) Controller가 Warning에게 요청을 받아 직접 출력한다.



2) Warning 내부에서 출력한다. (선택)

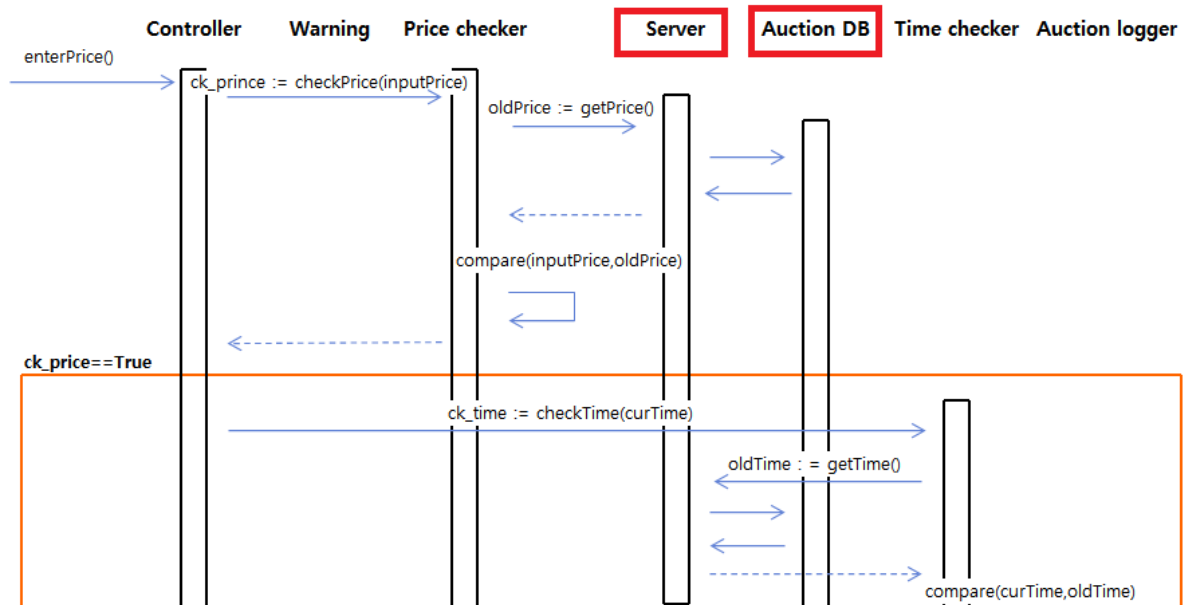


A) 논의 결과, 경고 메시지는 Controller보다는 Warning에서 맡아서 하는 것이 Concept적으로도 맞다고 생각하여 Warning 내부적으로 출력을 담당하기로 하였다. Controller는 다른 Concept들에게 요청하고 결과를 받는 역할정도로 한정 지었다. (UC-12에서 완료요청에 대한 결과 출력도 마찬가지로 Request completion에서 담당하여 출력한다.)

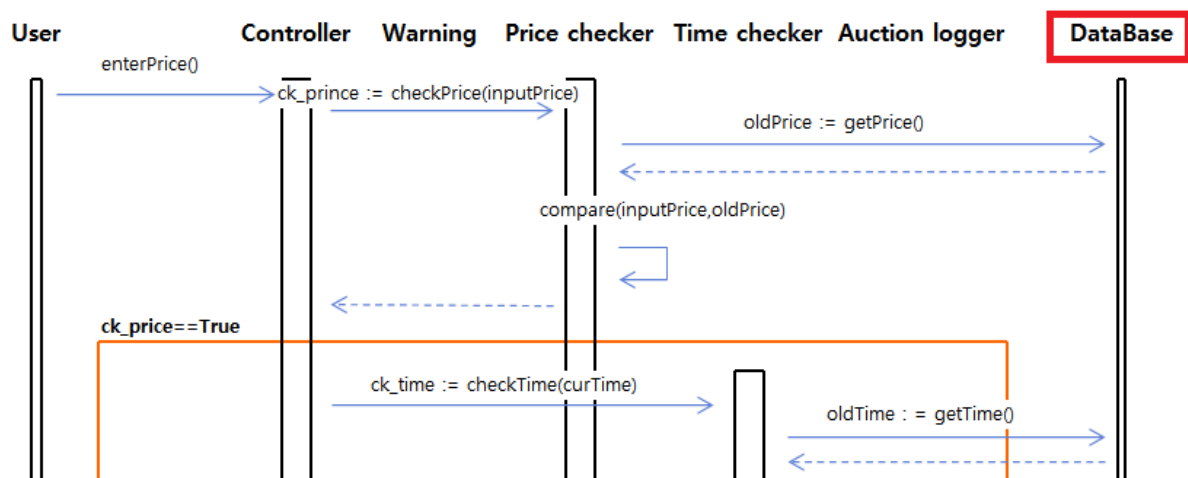


## Q2) Server, log(DB) concept vs DB (Actor)

### 1) 서버와 데이터베이스를 클래스로 나타낸 경우



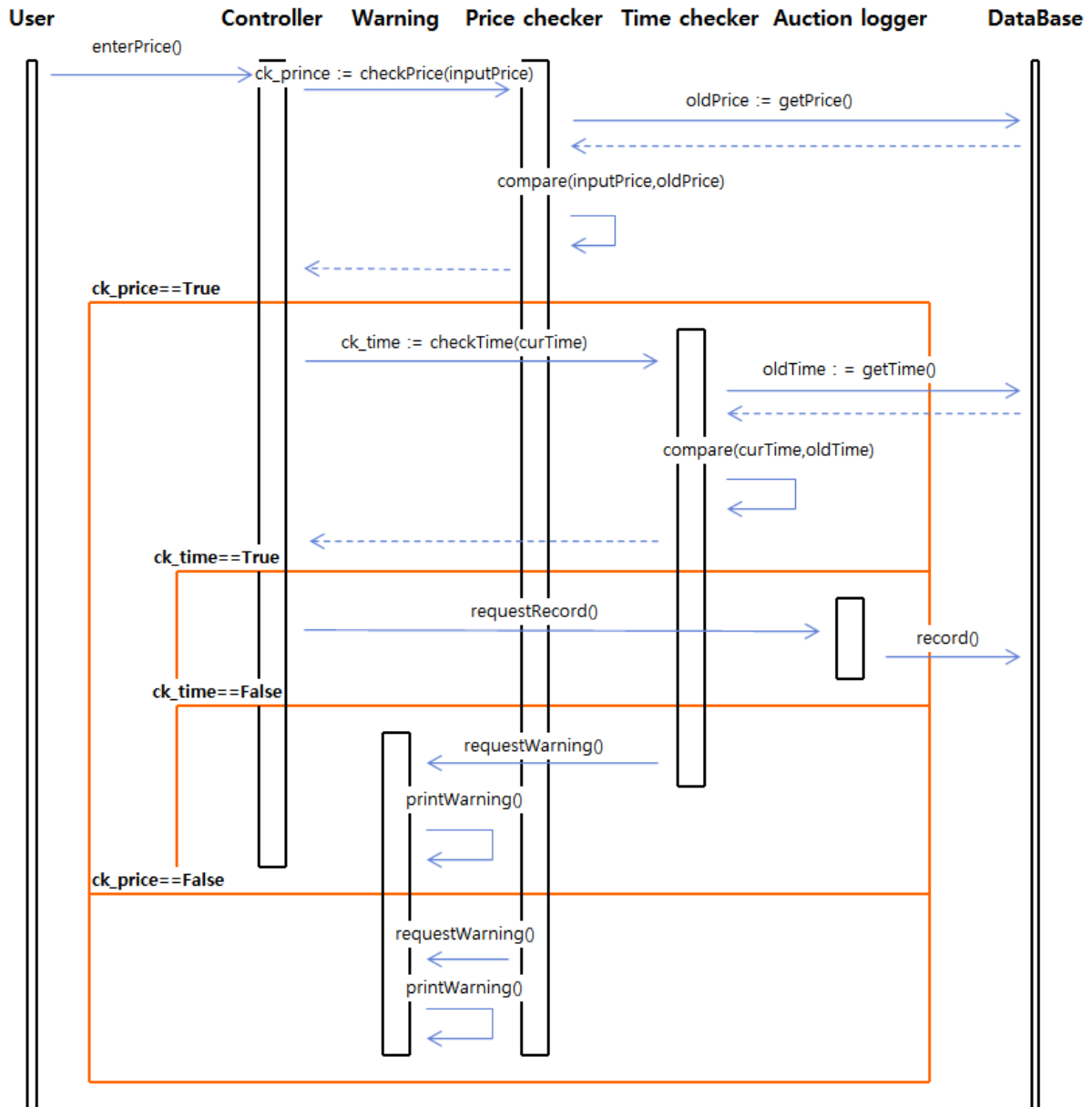
### 2) 서버를 제외시키고 데이터베이스를 Actor로 나타낸 경우 (선택)



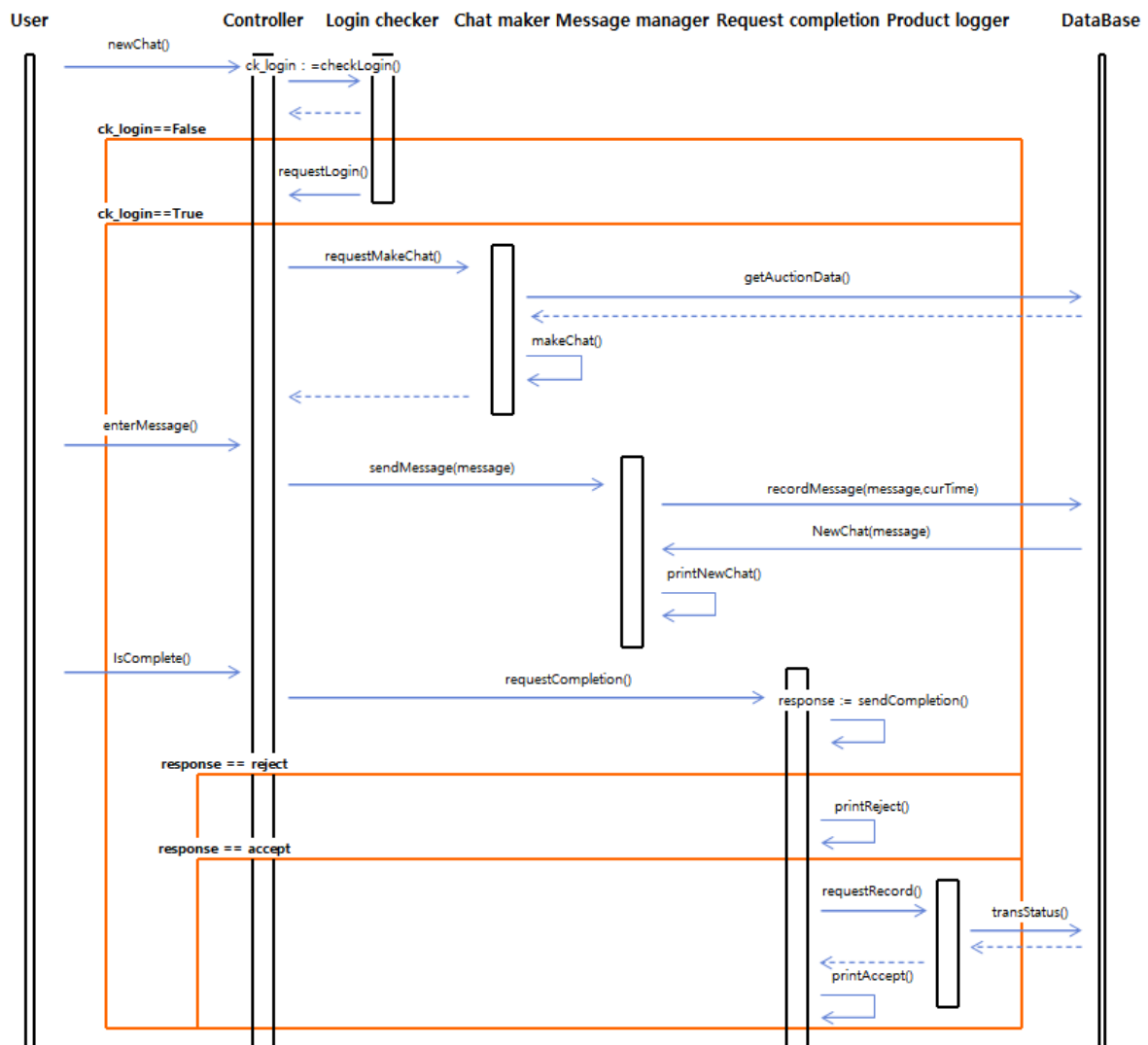
A) 논의 결과 : Class diagram을 만들면서 server, log(DB)는 하나의 클래스라기보다는 Actor에 더 가까운 역할을 수행하고 있었다. 또한 Server는 생각하였는데 DB로 접근하기 위해서는 당연히 Server를 통해야 하고 이 기능에서 Server만의 독자적인 기능이 있는 것은 아니었으며 SDD의 가독성을 떨어뜨리기 때문에 생각하였다. 따라서 설계 및 Sequence를 표현하는데 더 적합하다고 생각하여 이 Server와 log concepts를 제거하고 DB라는 Actor를 추가하였다.

### 3. Sequence Diagram 최종 결과

[ UC- 11 ]

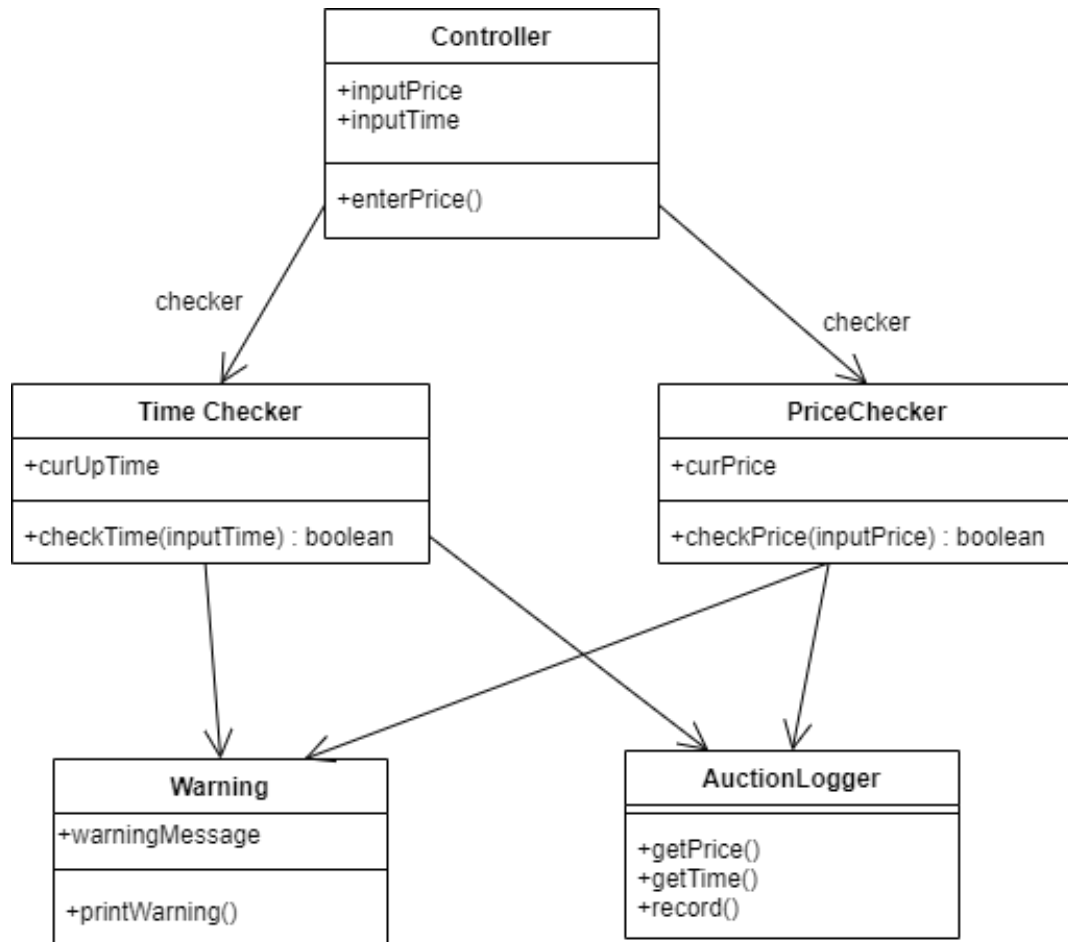


[ UC- 12 ]



#### 4. Class Diagram 결과

[ UC- 11 ]



[ UC- 12 ]

