

소프트웨어 프로젝트2 보고서

20191561 김서현

20191563 김수연

목차

1. 목적
2. 동기
3. 역사와 현황
4. 시스템 구성도
5. 시나리오/기능/화면
6. 구조 및 구현
7. 결론

1. 목적

경매장에 직접 가지 않고도 누구나 손쉽게 물건을 판매하고 구매할 수 있는 온라인 경매 시스템을 개발하는 것이 목적이다.

2. 동기

대학생들이 주로 이용하는 에브리타임이라는 사이트에는 장터게시판이라는 이름의 게시판이 있다. 장터게시판에서는 판매, 구매, 나눔 등의 카테고리에 맞게 제목과 글, 사진, 사진설명 등을 넣어 게시물을 작성함으로써 거래를 할 수 있다. 각 게시글에 구매나 판매 혹은 나눔을 원하는 사용자가 게시글에 댓글을 달거나 해당 게시글 작성자에게 쪽지로 연락하여 개별적으로 거래를 진행하는 방식이다.

3. 역사와 현황



193년 로마 황제가 살해된 후 귀족들이 황제 자리를 경매에 붙이며 처음으로 경매를 도입했고 이후 노예경매 위주로 성장하였다. 국내 인터넷 경매 서비스는 1998년 4월 최초로 도입하기 시작했다.



세계적으로 인터넷 상에서 경매를 통해 거래되는 건수 및 금액이 폭발적으로 증가 중이다. 인터넷 경매 사이트 수가 97년 161개 대비 현재 1691개로 10배 이상 증가하였다. 인터넷 경매의 세계시장 규모는 56억 달러이며 이용자는 20003년 기준 1400만 명이다.

4. 시스템 구성도



위는 시스템 구성도이다. 판매자와 구매자는 경매 클라이언트를 사용하여 경매에 참여한다. 경매 클라이언트들은 경매 서버에 연결되어 경매 요청 정보를 경매 서버에 전달하고 응답을 받는다. 또한 경매 서버로부터 실시간으로 입찰 정보를 통지받아 화면에 표시한다.

5. 시나리오

5.1 서버기동

```
suyeon@suyeon-ThinkPad-T480s: ~/소프트웨어 프로젝트II/소프 AD
suyeon@suyeon-ThinkPad-T480s:~/소프트웨어 프로젝트II/소프 AD$
suyeon@suyeon-ThinkPad-T480s:~/소프트웨어 프로젝트II/소프 AD$
suyeon@suyeon-ThinkPad-T480s:~/소프트웨어 프로젝트II/소프 AD$
suyeon@suyeon-ThinkPad-T480s:~/소프트웨어 프로젝트II/소프 AD$
suyeon@suyeon-ThinkPad-T480s:~/소프트웨어 프로젝트II/소프 AD$
suyeon@suyeon-ThinkPad-T480s:~/소프트웨어 프로젝트II/소프 AD$
suyeon@suyeon-ThinkPad-T480s:~/소프트웨어 프로젝트II/소프 AD$ ~/anaconda3
/bin/python kookmin_auction_server.py 8084
```

5.2 회원가입

판매자, 구매자가 시스템에 회원가입한다.

5.3 로그인

판매자, 구매자가 시스템에 로그인한다.



5.4 경매 물품 등록

판매자가 경매 물품 및 시작가격, 경매 물품 이미지 및 설명, 경매시작시각, 경매종료시각을

등록한다.

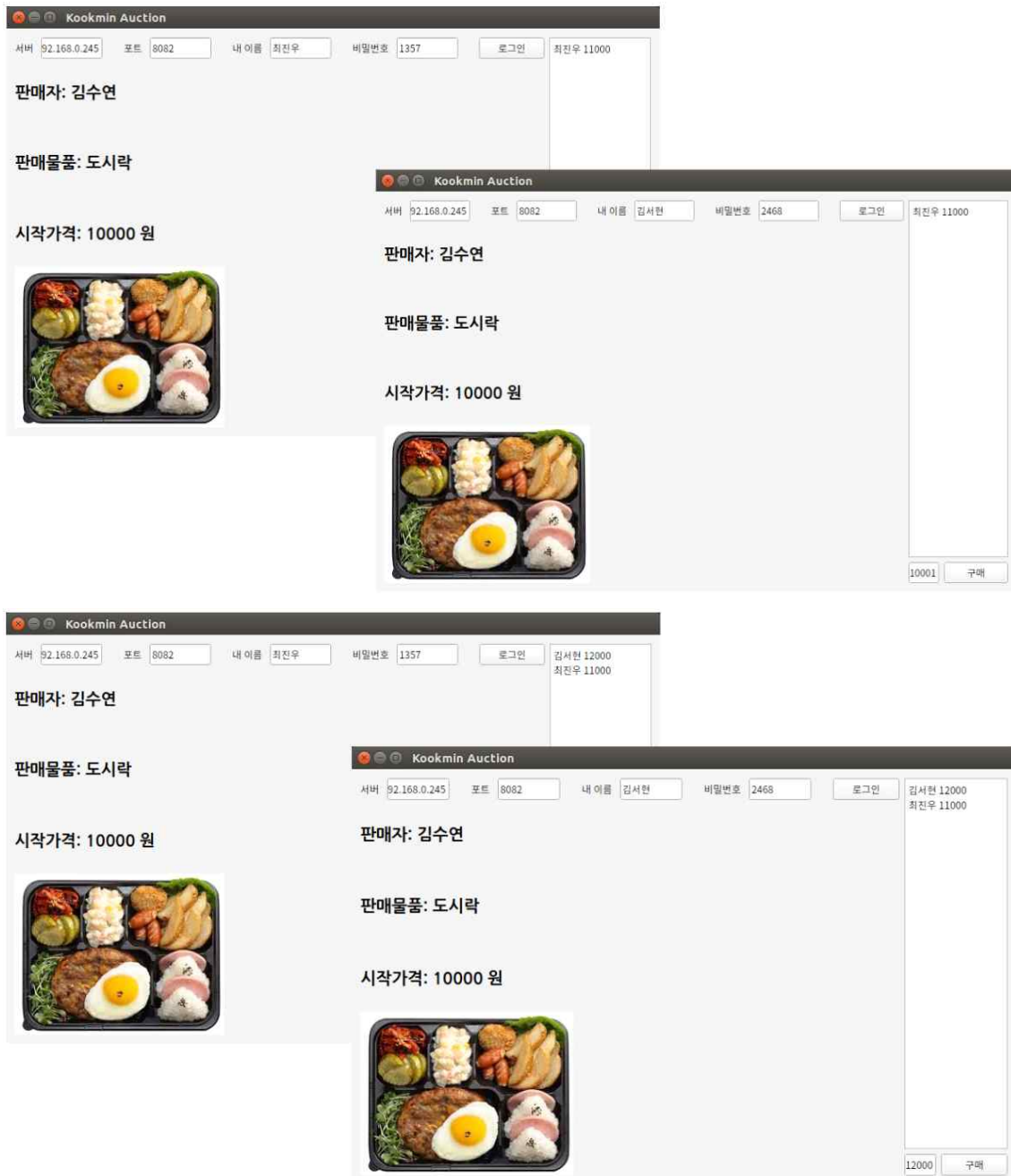
5.5 경매 개시

경매시작시간이 되면 자동으로 경매를 개시한다.

5.6 경매 진행

구매자가 경매시작시간 이후에 경매 화면에 들어와서 입찰가를 입력한다.

오른쪽 status 창에 현재 최고가가 표시된다.



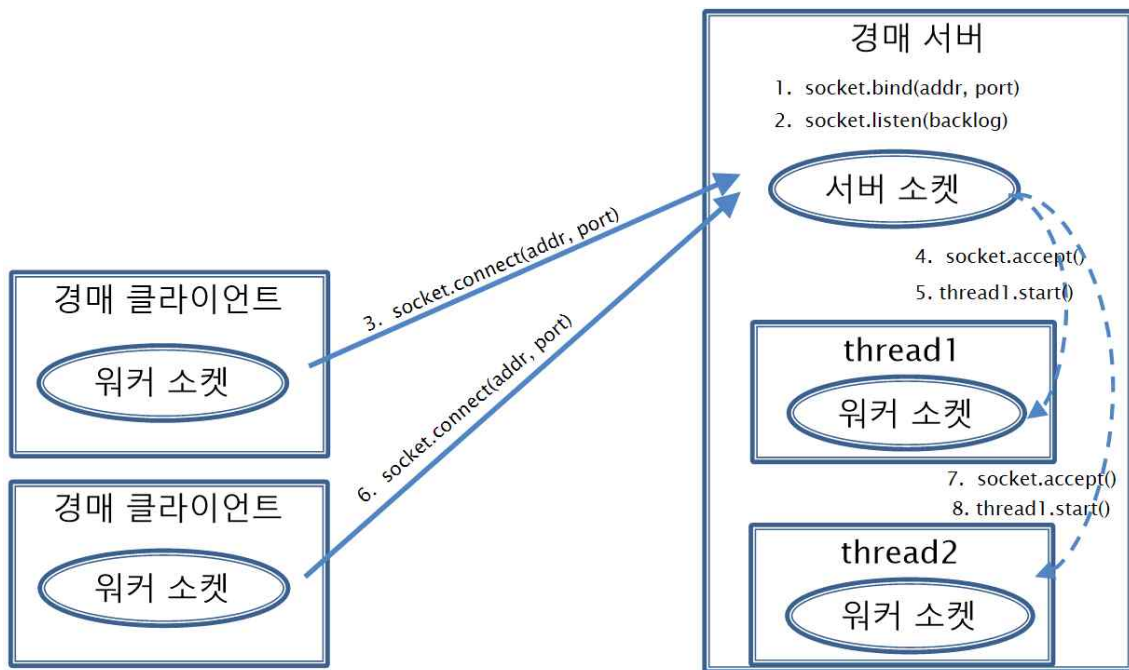
5.7 낙찰

경매종료시각이 되면 낙찰가 및 낙찰자를 표시한다.

6. 시스템 구조 및 구현

6.1 인터넷 소켓과 멀티스레드 프로그래밍 모델

전체 시스템은 다음과 같이 클라이언트-서버 구조로 구성된다. 네트워크 통신을 통해 원격으로 경매를 진행하기 위하여 인터넷 소켓을 사용하였다. 여러 클라이언트로부터의 요청을 동시에 처리하기 위해 멀티스레드 구조로 설계하였다.



(1) `socket.bind(addr, port)`

경매 서버에서 사용할 포트 `port`번을 점유한다.

(2) `socket.listen(backlog)`

경매 서버에서 경매 클라이언트로부터 전달되는 TCP 연결 요청을 받아들인다.

(3) `socket.connect(addr, port)`

경매 클라이언트는 주소 `addr`의 포트 `port`번에 binding된 경매 서버에 TCP 연결요청을 한다.

(4) `socket.accept()`

서버 소켓은 클라이언트로부터 수신한 TCP 연결요청을 승인하고 그 클라이언트와 1:1 통신하는 워커 소켓을 생성한다.

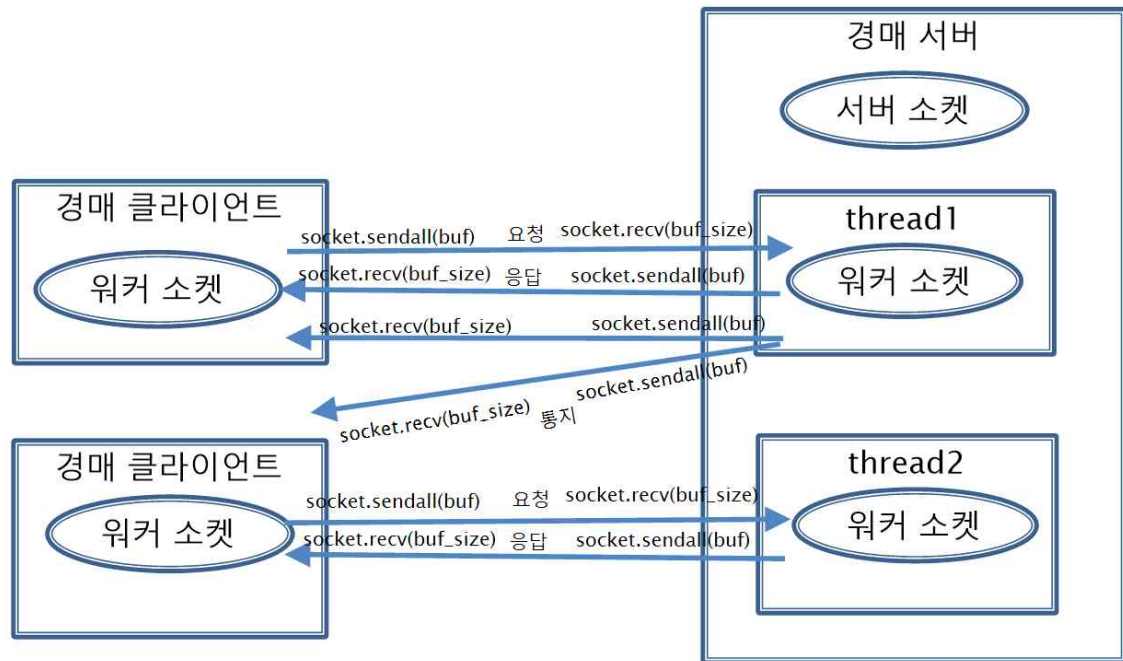
(5) `thread1.start()`

경매 서버는 (4)번에서 생성한 워커 소켓을 통해 경매 클라이언트와 통신하는 스레드를 시작한다.

(6~8) 이후 새로운 경매 클라이언트로부터 TCP 연결요청이 들어와도 위와 같은 과정을 반복하여 새로운 워커 소켓과 스레드를 생성하여 통신한다.

6.2 경매 통신 방식

경매 서버와 경매 클라이언트 간 통신은 요청 - 응답 메시지 방식과 전체 클라이언트에 방송하는 통지 메시지 방식을 제공한다.



6.2.1 요청 - 응답 메시지

대부분의 메시지는 경매 클라이언트에서 보내는 요청 메시지에 대하여 경매 서버가 응답 메시지를 전송하는 방식을 사용한다.

(1) 요청 메시지 전송

경매 클라이언트가 경매 서버에 요청 메시지를 전송할 때에는 경매 클라이언트가 `socket.sendall(buf)` 메서드를 호출하여 전송하고 경매 서버가 `socket.recv(buf_size)` 메서드를 호출하여 메시지를 수신한다.

(2) 응답 메시지 전송

경매 서버는 요청 메시지에 대한 응답 메시지를 준비하여 `socket.sendall(buf)` 메서드를 호출하여 전송하고 경매 클라이언트는 `socket.recv(buf_size)` 메서드를 호출하여 응답 메시지를 수신한다.

6.2.2 통지 메시지

실시간으로 최고가를 방송하는 경우 경매 서버가 경매 클라이언트에게 일방적으로 전송하는 통지 메시지 방식을 사용한다.

경매 서버가 경매 클라이언트에 통지 메시지를 전송할 때에는 경매 서버가 `socket.sendall(buf)` 메서드를 호출하여 통지 메시지를 전송하고 경매 클라이언트가 `socket.recv(buf_size)` 메서드를 호출하여 통지 메시지를 수신한다.

6.3 경매 통신 프로토콜

(1) 회원가입

- 회원가입 요청 메시지 포맷
'signup', '\n', ID(문자열), password(문자열), '\n'
- 회원가입 응답 메시지 포맷
'OK', '\n'
'ERROR', '\n', 오류 내용(문자열), '\n'

(2) 로그인

- 로그인 요청 메시지 포맷
'login', '\n', ID(문자열)
- 로그인 응답 메시지 포맷
'OK', '\n'
'ERROR', '\n', 오류 내용(문자열), '\n'

(3) 경매 물품 등록

- 물품등록 요청 메시지 포맷
'regi', '\n', 물품이름(문자열), '\n', 경매시작시각(문자열), '\n', 경매종료시각(문자열),
'\n', 이미지 길이(문자열), '\n', 물품 설명(문자열), '\n'
- 물품등록 응답 메시지 포맷
'OK', '\n'
'ERROR', '\n', 오류 내용(문자열), '\n'

(4) 경매진행

- 경매진행 요청 메시지 포맷
'purchase', '\n', 시작가격, '\n'
- 경매진행 응답 메시지 포맷
'OK', '\n'
'ERROR', '\n', 오류 내용(문자열), '\n'
- 최고가 방송 메시지 포맷
'current', '\n', 최고가 입찰자, '\n', 최고 가격, '\n'

(5) 낙찰

- 낙찰 방송 메시지 포맷
'finish', '\n', 낙찰자, '\n', 낙찰가, '\n'

7. 결론

경매가 성사되었을 때 구매자가 경매물품을 받는다는 보장이 있어야 하고, 판매자는 구매자로부터 낙찰가를 입금받는다는 보장을 하기 위해 결제나 기타 보장을 위한 방안이 필요하다.