CS-306 DATA COMMUNICATION AND COMPUTER NETWORKS

LAB PROJECT

SECTION: "B"

SUBMITTED TO:

Mam Momina Behzad

SUBMITTED BY:

Shahzad Babar (22-SE-087)

DEPARTMENT OF COMPUTER SCIENCE

HITEC UNIVERSITY, TAXILA

Project: Auction System Using Socket Programming

This project implements a simple auction system where clients can connect to a server to participate in an auction. The server manages the auction process, including handling bids and determining the winner. The system simulates real-time bidding, including random competitor bids.

Key Features

1. **Server-Side (Auction Server):**
    - Starts the auction by asking the first client for the item name and initial bid. ○ Maintains the highest bid and the highest bidder throughout the auction.
    - Simulates competitor bids using a random number generator. ○ Ends the auction after a fixed number of rounds or if the client exits.
    - Displays a message if no bids are placed, indicating that the item is sold to the initial bidder.

2. **Client-Side (Auction Client):**
    - Connects to the auction server. ○ Allows the user to place bids or exit the auction at any time.
    - Receives real-time updates from the server about the current highest bid and competitor activity. ○ Ends gracefully when the auction concludes.

CODE:

Auction Client:

```
import java.io.*; import

java.net.*; import

java.util.Random; import

java.util.Scanner;
```

```
class AuctionClient {     public static void main(String[]
args) {        try (Socket socket = new Socket("localhost",
5000);
          BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
          PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
Scanner scanner = new Scanner(System.in)) {


         System.out.println("Connected to auction server.");
String message;           while ((message = in.readLine())
!= null) {
             System.out.println(message);


             if (message.startsWith("Enter the item name:") || message.startsWith("Enter the
initial bid:") || message.startsWith("Enter your bid")) {                String userInput =
scanner.nextLine();                out.println(userInput);


                if (userInput.equalsIgnoreCase("exit")) {
                   break;
                }
             } else if (message.startsWith("Auction ended.") || message.startsWith("No bids
received.")) {
                break;
             }
          }
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

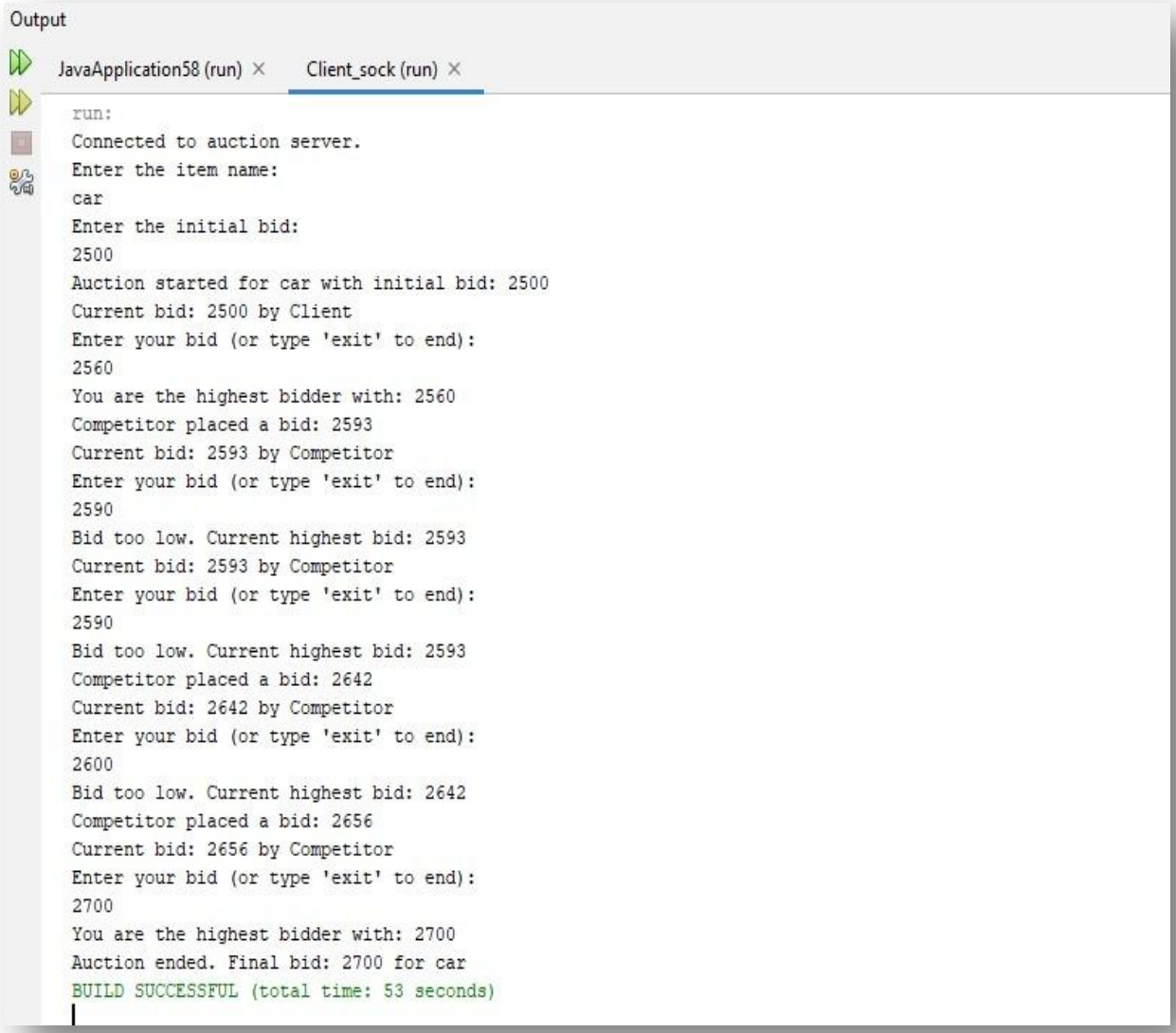
Auction Server:

```
import java.io.*; import
java.net.*; import
java.util.Random; import
java.util.Scanner;

class AuctionServer {     private static int highestBid =
0;     private static String highestBidder = "No Bids
Yet";     private static String auctionItem = "Unknown
Item";

    public static void main(String[] args) {        try (ServerSocket
serverSocket = new ServerSocket(5000)) {
            System.out.println("Auction server started. Waiting for a client...");

            while (true) {              try (Socket clientSocket =
serverSocket.accept();                  BufferedReader in =
new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
```

```
PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true)) {

System.out.println("Client connected.");

out.println("Enter the item name:");
auctionItem = in.readLine();
out.println("Enter the initial bid:"); highestBid
= Integer.parseInt(in.readLine());
highestBidder = "Client";

out.println("Auction started for " + auctionItem + " with initial bid: " +
highestBid);
Random random = new Random(); boolean bidReceived
= false; for (int i = 0; i < 5; i++) { out.println("Current
bid: " + highestBid + " by " + highestBidder); out.println("Enter
your bid (or type 'exit' to end):"); String input = in.readLine();

if (input == null || input.equalsIgnoreCase("exit")) {
break;
}
int clientBid = Integer.parseInt(input);
if (clientBid > highestBid) { highestBid =
clientBid; highestBidder = "Client";
bidReceived = true; out.println("You are the highest bidder
with: " + highestBid);
```

```
                } else {
                    out.println("Bid too low. Current highest bid: " + highestBid);
                }
                int competitorBid = highestBid + random.nextInt(50) + 1;
if (random.nextBoolean()) {                    highestBid =
competitorBid;                    highestBidder = "Competitor";
bidReceived = true;                    out.println("Competitor placed a
bid: " + highestBid);
                }
            }
            if (!bidReceived) {                    out.println("No bids received. The item
is sold to the initial bidder.");
            } else {
                out.println("Auction ended. Final bid: " + highestBid + " for " + auctionItem);
            }
break;
        } catch (Exception e) {
            e.printStackTrace();
        }  }
    } catch (IOException e) {
        e.printStackTrace();
    }
  }
}
```

OUTPUT:


Output
JavaApplication58 (run)
Client_sock (run)
run:
Connected to auction server.
Enter the item name:
car
Enter the initial bid:
2500
Auction started for car with initial bid: 2500
Current bid: 2500 by Client
Enter your bid (or type 'exit' to end):
2560
You are the highest bidder with: 2560
Competitor placed a bid: 2593
Current bid: 2593 by Competitor
Enter your bid (or type 'exit' to end):
2590
Bid too low. Current highest bid: 2593
Current bid: 2593 by Competitor
Enter your bid (or type 'exit' to end):
2590
Bid too low. Current highest bid: 2593
Competitor placed a bid: 2642
Current bid: 2642 by Competitor
Enter your bid (or type 'exit' to end):
2600
Bid too low. Current highest bid: 2642
Competitor placed a bid: 2656
Current bid: 2656 by Competitor
Enter your bid (or type 'exit' to end):
2700
You are the highest bidder with: 2700
Auction ended. Final bid: 2700 for car
BUILD SUCCESSFUL (total time: 53 seconds)