CSCI 3901 Assignment 1: External Documentation

Overview:

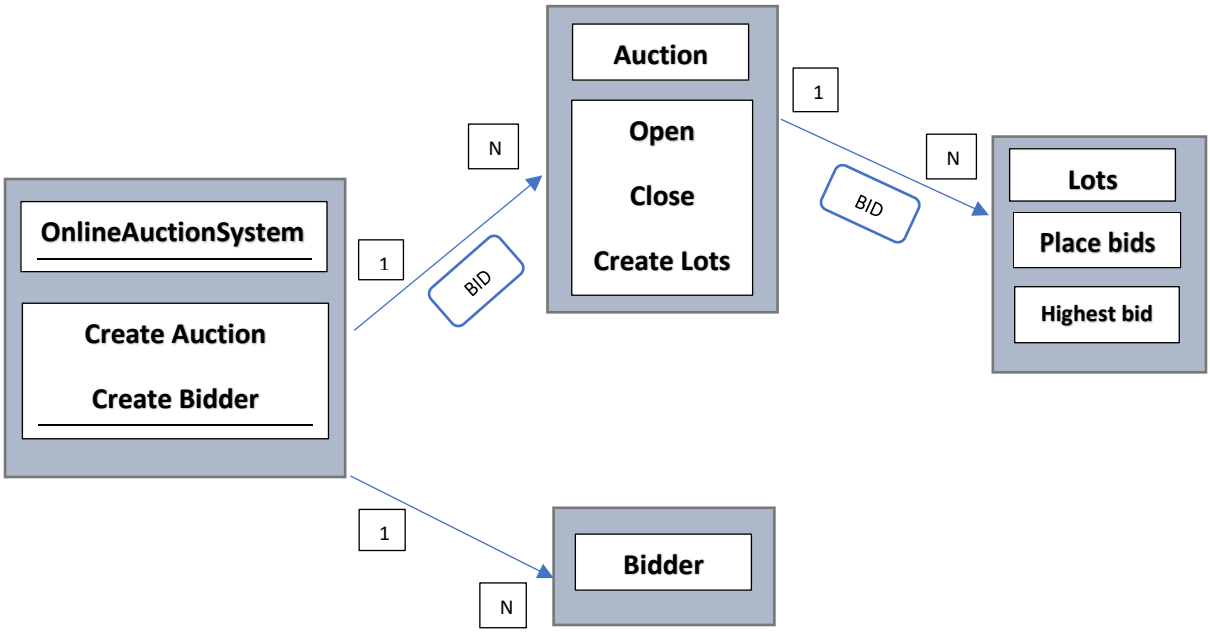
Auction system where the bidder can place bids on lots of items, one feature of the auction is an auto-bid system, where bids are automatically placed on behalf of the highest bidder trying to outbid the competition.

[Please refer to CSCI 3901 Assignment1 for more specific rules on how the auction should function.]

Approach:

Decomposed the functionalities and stakeholders into classes namely OnlineAuctionSystem, Auction, Bid, Lot, Bidder, which stored its own specific information and performed specific actions of its instances based on the set of rules, all held together inside OnlineAuctionSystem.

Class Relation Overview



Data structures and their relations to each other:

OnlineAuctionSystem holds a list of Array list of <Auctions> objects, it also holds all the <bidder> data in an Array list.

Each Auction object has a number of <lots> stored in Arraylist, each instance of lots holds all the essential data like highest bid, current bid to process a bid accordingly.

When a bid is placed and after validating the bid data we send the bid for a particular auction it belongs to and the further to the lot it belongs to and processed for that particular lot instance.

Assumptions:

1. Doesn't allow Redundant auction name: Assuming the auction object identifier to be auction name the system doesn't allow you have the same auction name for multiple auctions.
2. Assuming that the bids can be made other than the multiples of minimum increment, the system considers this bid and tries to outbid the same by the minimum increment, if an existing highest remembered bid exists.
3. If a bid matches the highest remembered bid this bid is outbid by the highest remembered bid automatically by the system and status is returned 2 for the current bid, because it is not the current leading bid.
4. If the existing highest bidder rebids on the lot with the same amount as that of the highest remembered bid amount then we return status 2.

If the highest bid cannot outbid the other bid by minimum increment then the following assumptions are made.

5. if an incoming bid is less than the highest remembered bid, and the highest remembered bid is less than the sum of [incoming bid + minimum increment] then the incoming bid is the current bid.
6. if an incoming bid is greater than the highest remembered bid, and the incoming bid is less than the sum of [previous highest remembered + minimum increment] then the highest remembered bid is the previous highest bid.

Key algorithms and design elements:

The design was focused on reducing the key components into different classes and objects and processing bids at different stages.

Helper functions were written to check bad data and handle checks at various stages, [OnlineAuctionSystem > Auction > lot].

Algorithm: Auto-bid system.

STEP 1: Create auction and Bidders in the system

STEP 2: Open auction for bids to be placed

STEP 3: place bids on lots

STEP 4: check if the bid is valid [bidder exists, lot exists, the bid amount is greater than > next legal bid].

STEP 5: Process the bid on the lot it was place and auto bid based on different conditions explained below in the key details.

STEP 6: Return 1 if a bid is not accepted, 2 if the bid is accepted, 3 if it's the highest bid, 4 if it's 3 and there is scope for further auto-bid.

STEP 7: Close auction through the command

STEP 8: Reporting function that retrieves [status, winning & owed] the current state of the auctions, lots and bidders.

Key Details:

The bid request is sent out to the associated auction it belongs to and further redirected to the particular lot instance and runs and updates the state of that lot on which the bid was placed.

Auto bid system automatically bids on behalf of the incoming bids with minimum increments of the auction specified, the auto bid system tries to outbid the incoming bid by minimum increment by automatically bidding on behalf of the highest bidder.

Any bid below the next legal bid Is not accepted and if the highest remembered bidder rebids on the same lot only the greater is accepted.

Any bid below or equal to the remembered highest bid is automatically outbid by the system on behalf of the highest bidder.

Limitations:

- Doesn't allow Redundant auction name:
- Cannot Re-open a closed auction.
- Cannot allocate lots of the one auction to another after it's closed.
- Lot ranges for auction is fixed and cannot be altered at a later stage
- Minimum increment if for the whole auction and not of individual lots
- No relevant error information is given to the user when bad data is passed
- No exception is thrown in other words try and catch block is not used to handle all exceptions.