**HIGH LEVEL DESCRIPTION**

Our implementation of Auction House system is very similar to the one we came up with for the previous coursework. AuctionHouseImp singleton class which implements AuctionHouse interface is central to the whole system. Every message from any of the actors goes to AuctionHouse first, which then calls methods of other classes or executes an operation itself.

Due to the classes and their methods already provided we needed to make a couple of changes to our system from coursework 2. Most of them was caused by modelling AuctionHouse as an interface. We decided to link all our extra classes which need an access to the AuctionHouse to the AuctionHouse interface. This makes further development easier, i.e. adding another implementation of AuctionHouse to the system, but on the other hand it caused tighter coupling, i.e. we had to connect Auction with Catalogue.

We made use of several types of Java collections. Since there is a requirement for lots to be ordered, TreeMap is used for storing CatalogueEntry objects in Catalogue and HashMap for storing their reserve prices. Buyers, sellers and auctions don’t need to be stored in order we chose a simple ArrayList for all 3 of these objects. In a case where storing a same element twice would be redundant we used HashSet (i.e. set of all the lot numbers where a buyer noted interest). Had we added equals method in buyer, seller or auction class, Set would have probably been a better choice for these classes as well. This way we have to manually loop through lists in order to find out whether some user is already registered under the same name.

Maybe we could’ve created few more extra classes instead of making AuctionHouseImp look like an all-powerful class. Back during coursework 2 we decided to stick with a smaller number of classes, because otherwise we would’ve ended up with a messy UML and more complicated system structure. But now, when we actually implemented all the AuctionHouse methods, adding for example UserService interface or keeping lists of registered users out of the AuctionHouseImpl in a separate class with all the methods needed to operate on them was probably a good idea.