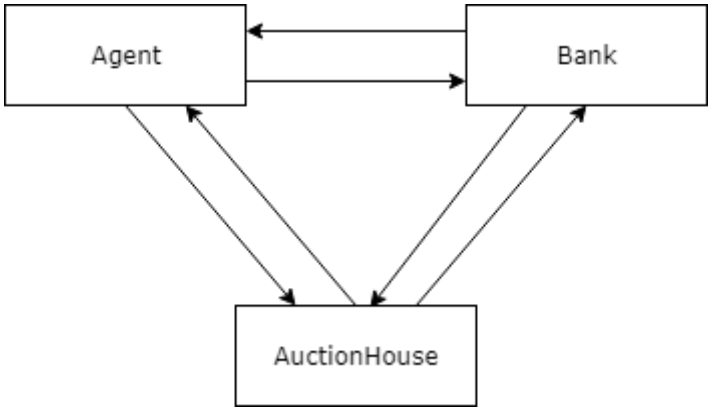
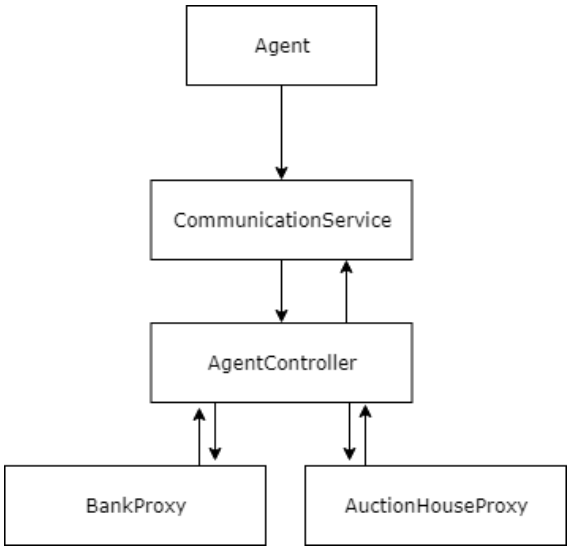
Design Descriptions

Overall Design



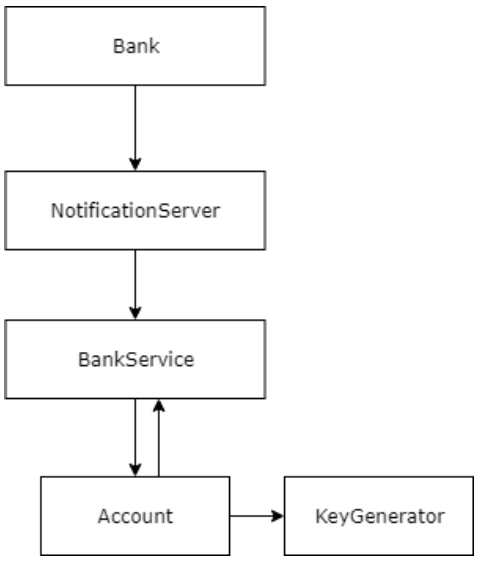
Agent Design



AgentController

The AgentController object is the main client controller used by the Agent. It is connected to a BankProxy and an AuctionHouseProxy. It is a thread that listens for input from the user via command line and provides a menu of options for the user.

Bank Design



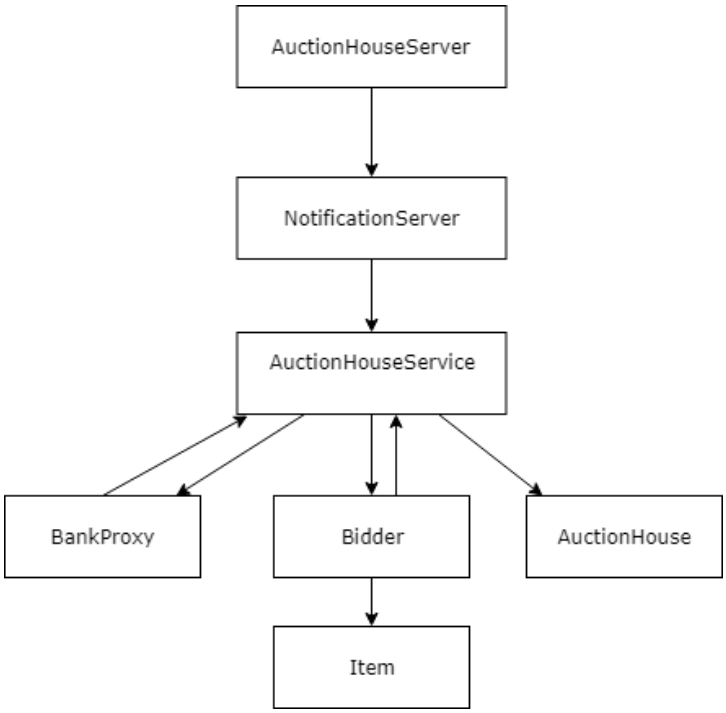
BankService

The BankService is the main service thread for the bank server. It handles incoming client/socket connections given to it by the NotificationServer. It stores a thread pool and creates a new Account thread for each new connection which handles that specific client's requests. The BankService also stores a list of all Agent and AuctionHouse accounts and handles functions between accounts (transfer funds, block funds, unblock funds).

Account

The Account object handles all incoming/outgoing requests of a given client. It stores all information that is specific to a client's account such as the balance, current holds amount, bidding key or account key, etc. The Account object listens for incoming requests and handles the request (sends the requested information or transfers funds, etc.).

Auction House Design



AuctionHouseService

The AuctionHouseService object is the main service thread for the auction house service. It is similar to BankService in that it handles incoming client/socket connections given to it by the NotificationServer. The AuctionHouseService is also a runnable thread that provides a small, built-in command-line gui to create the auction house. The AuctionHouseService handles new connections by spawning a new Bidder object thread that handles that given client's requests.

Bidder

The Bidder object is a communication thread between AuctionHouseService and the input/output of the Socket connected to the specified client.

Bid

The bid object is a runnable that extends thread. It is meant to be a countdown for the 30sec that must pass for a bid to win. It is to send notification to the bidder that they have won if it awakens to find the bid information on the item the same.

Item

The item class is a helper class that keeps track of items. Items are stored with an ID number, their name/description, the minimum bid required to bid on them, the current bid on them (which is one number less than the minimum bid), and the bidder who has the current bid. The bidder is null until someone actually bids on the item.

Auction House

The AuctionHouse object holds all the relevant inventory information required. It reads in from a set list of items, and randomly picks the specified number (specified over commandline). It also randomly sets the minimum prices. It has an internal method which will allow bids to be made, however, it assumes that AuctionHouseService has already vetted the bid as valid.

Shared Objects Design

BankProxy

BankProxy is used by the AgentController and the AuctionHouseService to communicate with the BankService. It handles the I/O of the socket connection.

AuctionHouseProxy

AuctionHouseProxy is used by the AgentController to communicate with AuctionHouseService. It handles the I/O of the socket connection.

Helper Objects Used

Agent

The Agent object contains the main method for the Agent jar file. The Agent object is used to start the AgentController thread and connect the AgentController to the bank. The thread dies once AgentController is connected to bank via BankProxy. Agent uses the CommunicationService object to connect and initialize AgentController with a BankProxy object.

AuctionHouseServer

The AuctionHouseServer object is used in the same way as the Agent and Bank objects. It contains the main method for the AuctionHouseServer jar file. It is used to connect the AuctionHouseService object to the BankService via a BankProxy, and to open a NotificationServer for the AuctionHouseService to listen on for incoming socket connections.

Bank

The Bank object contains the main method for the BankService jar file. It is used to start the BankService and to listen for incoming socket connections. The sockets are sent to BankService to handle.

KeyGenerator

The KeyGenerator object is used by Bank to generate random strings to be used as bidding keys and bank account numbers for Agents to use. They are used for agents to give to auction houses, and for agents to give to bank to transfer funds to an auction house.

CommunicationService

The CommunicationService object is used to connect an agent or an auction house to a server. It provides two methods to connect to either a bank or an auction house. It takes a host name and port number, and returns a BankProxy or AuctionHouseProxy object corresponding to the connection made.

NotificationServer

The NotificationServer object is used to start a server and to listen for incoming

connections. It is used by the BankService and by the AuctionHouseService. It takes a port number and uses it to start the corresponding service. Once the service is created, its thread then listens for incoming socket connections and sends them to the service to handle.

AuctionHouseMessages

AuctionHouseMessages is an enum object that is used by the Agent to communicate with the AuctionHouseService. It is used by the AuctionHouseProxy and the Bidder object. The Bidder object is a subclass of AuctionHouseService

BankMessages

BankMessages is an enum object that is used by the Agent and the AuctionHouseService to communicate with the BankService. It is used by BankProxy and the Account object. The Account object is a subclass of the BankService.

Unused Objects

DisplayController

The DisplayController was going to be used for an agent GUI. The GUI was converted to command line input in the AgentController run thread.