Reverse Buying System

Peeyush Tugnawat
Haitham Hamza
Dr. Mohamed Fayad

Table of Contents

Abstract

Today's web-based auction systems are designed to sell available products to potential buyers, and little more. This paper explores the means by which these systems could be improved in terms of value provided to web auction customers. Imagine a system where a buyer can specify his/her requirement and price. For example, buyer of a car can actually specify the needs, estimated price and other details that he/she is looking in a car. And then the users (potential sellers and/or manufacturers) sharing and using the system can respond to the buyer's desired request posted on the system and really can bid for the same. Talking about the scope of the system, the system and basically idea can be applied to a variety of applications including the *Business to Customer (B to C)*, *Customer-to-Customer (C to C)* and *Business-to-Business (B to B)*.

Desired System

Online auction sites abound on the web. These online auctions allow buyers to bid for the thousands of products described on your personal computer. Largely, these systems are narrowly focused on selling items and have very little awareness of the buyer, and his or her requirements and needs.

Recently, a few web sites have appeared with the notion of a reverse auction. Reverse auctions are focused and dedicated to the buyer. A buyer describes his or her product requirements and the price he or she expects to pay. The need of this system arises in the areas of complex requirements where a buyer probably is looking for custom product that does not exist and there is a need to build it and then supply it. The example of this can be a particular new mechanical part of an automobile that is required to be manufactured according to the buyer's requirement and specification. The product may not have been developed or produced in past.

Reverse auction systems are not only specific to a particular industry, area or product. It is a concept that can be applied universally to any business need or an individual requirement. However the best utilization can be achieved in the *B-to-B* applications where the need is to really build, supply and transport the product that is customized according to the buyer's requirement

We desire to implement the instance of this idea in car selling and buying business. The implemented system should address the requirement of the buyers of the car (used and new). Our System should be able to provide solution for the problems that a car buyer faces when he/she wants to buy a new/used car. Also there exists a need for facilitating the buyer by letting him demand existing/customized car according to his needs and desire and not impose the traditional and existing system of buying the cars by just selecting from the available existing options only.

Thus the buyer should be able to provide the specifications of the car he/she desires and provide a time limit for receiving the selling proposals from sellers. The seller's bid (propose their selling price) and thus the prices lower down as auction goes on. At a particular point it can be known what is the current/final lowest bid or quote offered by a seller.

All the bid offers are eligible for selling and further communication as the last decision remains with the buyer. But the system facilitates the buyer by selecting the lowest bid received and also other bids arranged with respect to offer price. The buyer in this case may or may not choose the lowest bid.

The interface should provide the facility to all the users involved in a particular biding process for getting detailed information about buyer/seller. Also they may have additional features such as enquire by email. The bids should be maintained in a tabular interface both at buyers and suppliers end for sake of readability and clarity. This summary table could be drill down as the user follows the hyperlinks provided.

In this system, we focus on facilitating the buyer more, but in the process the other users sellers and manufacturers also benefit by the business they do.

Detailed Requirements

The user or buyer should be provided with the following features and capabilities when he/she communicates with the system to help buy a car:

1. The user should be able to post a requirement posting on the system for its other users that are the suppliers, dealers or the manufacturers. The requirement includes details and specifications about the desired car. Also it includes the bid period for the buying request.

2. The system should provide an interface for entering users' preferences, which are:

   The Manufacturer's Company

   This field should be populated from the existing manufacturers associated with the system. Option to specify other if doesn't exist.

   The Model of the Car

   The buyer should be able to select and/or specify his own requirement from a list showing the different models of the car available.

   The year of Make (if opting for a used car)

   Choose and/or select

   Transmission, Automatic/Manual

   Choose and/or select

Preference of Buying (from a manufacturer, dealer or a owner)

The preference of buying the car from a dealer, company or an individual lies with buyer and should be able to make selection for the same.

Price Range

Select the provided ranges and/or select his own.

Target Price

This price indicates the approximation of the price that the buyer is looking forward to pay for.

3. The system should be able to classify and categorize the various needs of buyers and post them to the corresponding suppliers, helping the suppliers to view the details of the initiated posting by a particular buyer.
4. After specifying the details and submitting it to the system, the user should be able to see a drill down feature list for the bids and functionalities that he/she has initiated. The bids should be maintained in a tabular format and the initial details should include time left and minimum price quoted. For other details such as the dealers name and/or company specification the user should be able to click on hyperlink for the bid submitted by the dealer/company to get more details about the same.
5. User should have the option of making his/her posting public, private or say for a particular dealer or manufacturer.
6. The user should be able to provide the time frame of the reverse auction. It means that user should limit the time for the dealers and manufacturers to bid for the particular posting.
7. The user should have the facility to view the final result categorized and customized according to his requirement. Options for the custom output could be the table sorted according to some preferences as seller's capacity or delivery time etc.
8. The final verdict remains with the buyer whether to choose a particular offer or not.

Use Cases

Use case #1: Posting request

Actors: buyer

In this case, a user can post his request for a car by specifying his requirement including the manufacturer's company, the model of the car, the year of Make (if opting for a used car), transmission type, preference of buying, target price and so on. A buyer also needs to specify a deadline for the reversed auction (starting date and ending date).

Use case #2: Sending buyer's request to corresponding User(s) (Sellers)

Actors: supplier, dealer, owner and buyer

The system classifies and categorizes the requests posted by buyers and posts them to the corresponding suppliers, dealers, or owners.

Use case #3: Collecting bids from suppliers, dealers or owners.

Actors: supplier and/or dealer and/or owner

The system sets a time frame (provided by buyer) that limits the time for bidding for a specific posting. Any bidding within that time frame will be collected and posted to allow bidders to view their biddings. After the deadline, no bidding will be allowed and collected. All biddings will be categorized and sorted.

Use case #4: Selecting the Current Lowest Bid

Actors: buyer and/or supplier and/or dealer and/or owner

The buyer views all the biddings posted to his account. The system searches for all the bid prices offered for the particular request and finds the lowest amongst them. This lowest bid price is shown to the buyer/seller on request.

Use case #5: Showing the User (buyer/Seller) Details

Actors: buyer and/or supplier and/or dealer and/or owner

The user (buyer/seller) follows the hyperlink to view the details of the corresponding seller/buyer. The system should provide the details as the name, company, manufacturing capacity etc depending upon the user.

Use case #6: Selecting the Final Lowest Bid.

Actors: buyer and/or supplier and/or dealer and/or owner

The buyer views all the biddings after the deadline. The final lowest bid is selected in the same manner as selecting the current lowest bid (Use Case #4).

The buying decision could be made if he is satisfied. Also he can contact and negotiate with the particular supplier. If he is not satisfied, he can call for another reverse auction, going back to *Usecase* #1. If he wants to obtain more information regarding certain bids, he can contact several suppliers for more detailed description.

Detailed Use Cases:

Use Case Id: 1.0
Title: *Posting the Request*

Actors	Roles
IndividualUser	Buyer
CarDealer	Buyer

Class	Attributes	Interface
IndividualUser	UserId, listOfPreferences	Set the Preferences for the Desired Car(s). State the approximate price. State the bid duration.
CarDealer	UserId, listOfPreferences	Set the Preferences for the Desired Car(s). State the Bid duration. State the approximate Price.
Car	model,color,manufacturer, yearofProduction, transmission,engineCapacity, mileage, category	Set the Corresponding attributes for the desired Car.
UserBuyingReq	reqId, priceQuoted, reqStartDate, reqEndDate	Set the specifications and details of the request initiated by the user(buyer). Post the specification of the desired car to the RequestHandler.

Use Case description:

1- The User specifies the preferences of the car that is desired like the model, year and other parameter desired.
2- The user sets the request starting date and request ending date for biding.
3- The user specifies the approximate price desired.
4- The user selects the type of sellers that should be shown the request (Optional).
5- The user confirms the specifications to be posted.

Enduring Business Themes: (EBT): Buying
Business Objects: Individual User
Industrial Objects: Car, CarDealer

Use Case Id: 2.0
Title: *Show User Request to Corresponding Sellers*

Actors	Roles
RequestHandler	Conveyer
UserBuyingRequest	Information Provider

Class	Attributes	Interface
RequestHandler	reqId, sellerPreferences	Receive the request made by the UserBuyingRequest. Search for corresponding users (sellers) and compare the available and potential cars by request parameters.
UserBuyingRequest	reqId, reqDuration, reqQuantity,	Receive the request made by the Buyer. Validate the request.
IndividualUser(Seller)	listOfSellingCars	To cater the information about the available car(s) for being able to compare to the request.
CarDealer(Seller)	listOfBrandsAvailable, listOfCars	To cater the information about all available car(s) for being able to compare to the request.
ManufacturerCompany (Seller)	carTypeList, modelList	To cater the information about all the cars for being able to compare to the request.

Use Case description:

1. The UserBuyingRequest Class provides the specifications of the buying request to the RequestHandler.
2. Search for the corresponding users (sellers) according to preferences.
3. Specifications and preferences provided by the user (buyer) are compared with the posted available car(s) and/or future/potential availability.

4. If the posting preferences match the details of seller, the request is posted to the corresponding seller.
5. RequestHandler class sends information to AccountUpdater Class that updates the account of both the prospective seller and prospective buyer.

Enduring Business Themes: (EBT): Management, Data Storage, Trading
Business Objects: Individual User, ManufacturerCompany, RequestHandler
Industrial Objects: Car, CarDealer

Use Case Id: 3.0
Title: *Show the Requests Initiated by Buyer*

Actors	Roles
UserBuyingRequest	Information Provider
User	Provide userId

Class	Attributes	Interface
User	UserId	To let be identified as a buyer. Provide userId for extracting the corresponding Requests.
UserBuyingRequest	reqId, reqDuration, reqQuantity,	Provide the details for the buying request initiated by the user (buyer).

Use Case description:

1. The user is identified as the buyer.
2. The user Id is searched in the buying requests for corresponding match in the buyer field of the User Buying Request.
3. Present the output to the buyer in a tabularized format that has all the buying requests that he has initiated.

Enduring Business Themes: (EBT): Buying, Management
Business Objects: IndividualUser
Industrial Objects: Car, CarDealer

Use Case Id: 4.0
Title: *Show Buying Request Details to User (seller/buyer)*

Actors	Roles
User	Account Holder
UserBuyingRequest	To cater the information about buying request.

Class	Attributes	Interface
User	userId, password	User is identified as a seller/buyer. User request the details of the buying request.
UserBuyingRequest	ReqId, reqDuration, reqQuantity, expDeliveryTime, upperPriceLimit	Provides the details supplied by the buyer at the time of making the buying request.
RequestStatus	reqId	To provide latest status as alive/not alive on the request posted in the seller's account.

Use Case description:

1. The user (seller/buyer) requests for details of the buying request.
2. All the requests corresponding to this user are shown in a tabularized form with hyperlink to get the details of the same.
3. The user clicks on the hyperlink to get the detailed information about the request.
4. All the information (data and current status) is shown to the user.

Enduring Business Themes: (EBT): Management, Trading, Data Storage
Business Objects: IndividualUser, ManufacturerCompany
Industrial Objects: CarDealer

Use Case Id: 5.0
Title: *Sellers Post their Bid Response*

Actors	Roles
IndividualUser	Bidder
Dealer	Bidder
ManufacturerCompany	Bidder

Class	Attributes	Interface
IndividualUser	userId, listOfSellingCars	IndividualUser bid for the received car request. Specify the offered bid details.
CarDealer	UserId, listOfBrandsAvailable, listOfCars	CarDealer bid for the received car request. Send bid to the bid response. Specify the offered bid details.
ManufacturerCompany	carTypeList, modelList	ManufacturerCompany bid for the received car request. Specify the offered bid details.

Bid	bidId;
	timePosted;
	priceQuoted;
	agreedQuantity;
	dateOfAvailability;
	deliveryExpenses;
		Gather the bid details that the seller bided for.

BidResponseHandler	respId	Updates the bid response to both the accounts i.e the seller's and the bid initiator.

Use Case description:

1. The user selects to bid for a particular request posted in his account.
2. User (seller in this case) quotes for all the required fields as price, quantity, availability time period etc.
3. The BidresponseHandler checks for price to be lower than any previous posted price (Refer Use Case 5.0) and provides Error Checking.
4. BidResponseHandler updates the account of the bid sender and the initiator.

Enduring Business Themes: (EBT): Negotiation, Selling, Customization, Management
Business Objects: IndividualUser, ManufacturerCompany, BidResponseHandler, Bid
Industrial Objects: CarDealer, Car

Use Case Id: 5.1
Title: *Selecting the Current Lowest Bid*

Actors	Roles
RequestStatus	Selector
Bid	Information Provider

Class	Attributes	Interface
Bid	priceQuoted	Bid class here provides information about the price quoted by seller.
UserBuyingRequest	reqId	Provides the Id for associating the bid with corresponding request.
RequestStatus	lowestBid, timeRemaining	Extract the lowest current bid from all the bids.

Use Case description:

1. The RequestStatus class reads all the bids posted for the particular buying request.
2. RequestStatus class gathers the priceQuoted from the bid.
3. It compares the price quoted with a large number and recursively compare it with other prices for the lowest price. The prices are arranged in an ascending order.
4. Provide the Value of the current lowest bid.

Enduring Business Themes: (EBT): Negotiation, Buying
Business Objects: Bid
Industrial Objects: None

Use Case Id: 5.2
Title: *Summary of All the Bids Posted by Seller*

Actors	Roles
UserBuyingRequest	Fetching the request responded to
Bid	Bid details

Class	Attributes	Interface
UserBuyingRequest	reqId	To provide the buying request details for the request that has been bided for.
Bid	BidId, userId(buyer)	Select the bids initiated by the seller. For each bid done, get the details.
RequestStatus	StatusOfBid	Supply the status of bid alive/not alive
User (Seller)	UserId	Provides identification as seller.

Use Case description:

1. The user Id is identified as the Potential Seller.
2. All the bids in which the user is identified as potential seller are selected and those alive are finally chosen.
3. Present the output (bids posted) to the user in a tabularized and ascending order format with regard to time.

Enduring Business Themes: (EBT): Selling, Management
Business Objects: Bid, IndividualUser, ManufacturerCompany
Industrial Objects: CarDealer

Use Case Id: 5.3
Title: *Get the Remaining Time*

Actors	Roles
UserBuyingRequest	Time Provider
RequestStatus	Status Checker

Class	Attributes	Interface
RequestStatus	isAlive	Determine whether the bid is live/not alive. To calculate the remaining time.
UserBuyingRequest	ReqDuration, initializingTime	To provide the time and duration for the particular request.

Use Case description:

1. A comparison is made for the buying Request about its status for checking whether it is still active or not.
3. If alive, for a particular request the current time is subtracted from the sum of the initiating time and the duration given by the potential buyer at the time of initiating the buying request.
4. The time remaining once calculated is used to provide screen alerts for the user Account.

Enduring Business Themes: (EBT): Management, Negotiation
Business Objects: None
Industrial Objects: None

Use Case Id: 5.4
Title: *Get Detailed Information of Seller/Buyer*

Actors	Roles
User	Information Requester Provide userId for extracting the information
RequestHandler	Search and get the user information and details.

Class	Attributes	Interface
User	userId	Provides userId for extracting the detailed information.
RequestHandler	User_id, password, list_of_brands	Search the particular userId and get the corresponding information.

Use Case description:

1. The user (buyer/seller) clicks on the hyper link of the seller/buyer name/id.
2. The user Id is picked up and searched for.
3. The details of the user and selected and presented in the corresponding HTML Template.
4. Show the corresponding template to the user.

Enduring Business Themes: (EBT): Data Storage, Management
Business Objects: IndividualUser, ManufacturerCompany, RequestHandler
Industrial Objects: CarDealer

Use Case Id: 6.0
Title: *Enquire by E-Mail*

Classes not yet fully implemented

Use Case description:

1. The potential buyer clicks on the enquire by e mail button.
2. A HTML screen template is presented with the field of outgoing mail populated with the user Id of the seller.
3. The potential buyer fills in the Text area for the inquiry and also particular comments if any and press send button for sending the message.
4. An e mail inquiry is posted to the corresponding seller's account.

Enduring Business Themes: (EBT): Negotiation, Buying, Management
Business Objects: IndividualUser, ManufacturerCompany
Industrial Objects: CarDealer

Use Case Id: 7.0
Title: *Get Time Remaining for User Request*

Refer Use Case 5.3

Use Case Id: 8.0
Title: *Freeze the UserRequest Bids (Close Bids)*

Actors	Roles
RequestStatus	To Mark a request as Closed
UserBuyingRequest	Provide Time information

Class	Attributes	Interface
RequestStatus	IsAlive	To calculate the time remaining and check the status. If time is over mark as Closed.
UserBuyingRequest	ReqId, timeInitiated, startingTime	Provide time information for the particular Request.

Use Case description:

1. Time Remaining is calculated for the corresponding user request by use case 5.3.
2. If the time elapsed is greater than or equal to the bid duration time the bid is marked for closing (free zed).
3. If a user tries to access the same or bid for it the system gives an error message saying that the bid period is already over for this request.

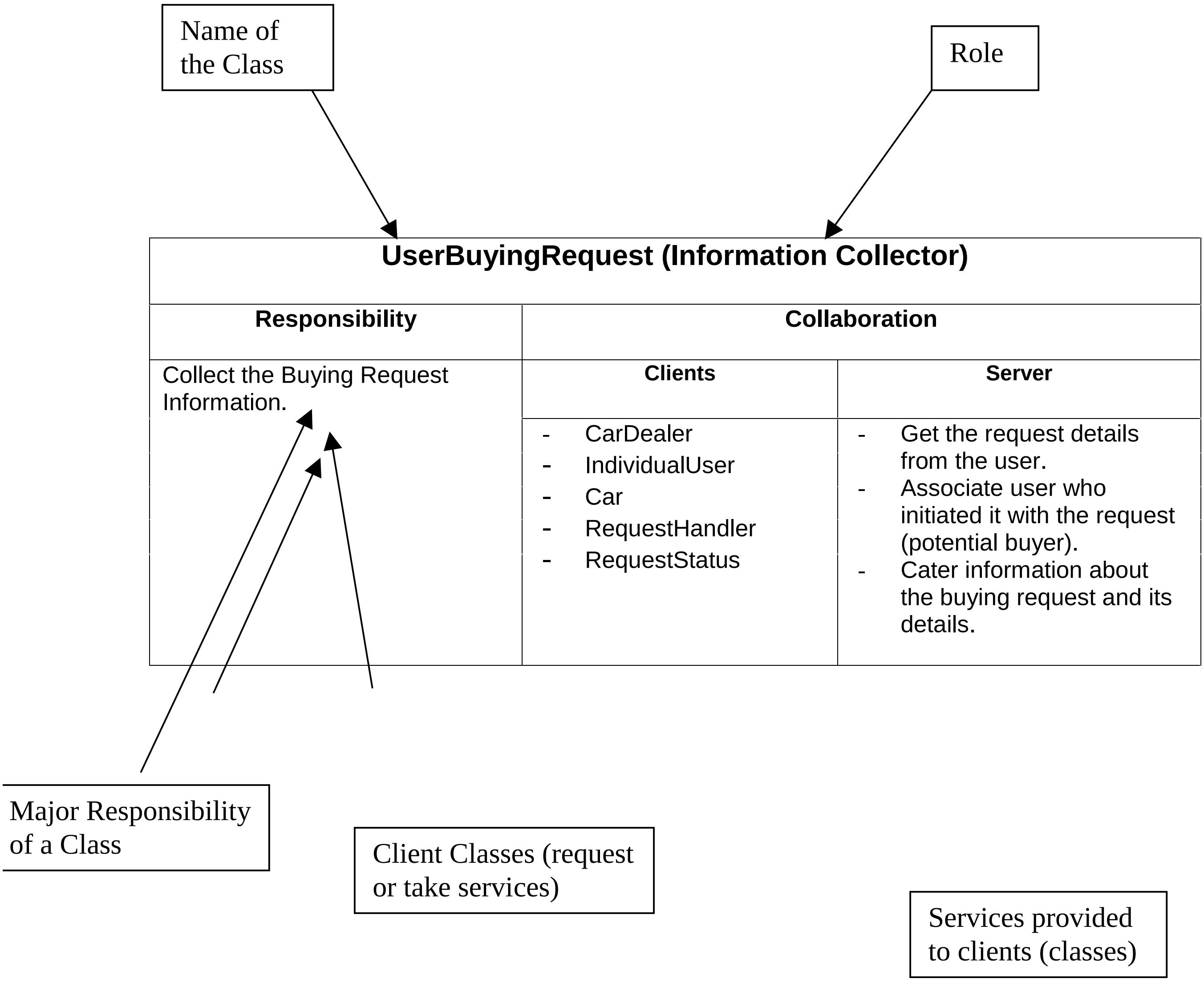
Enduring Business Themes: (EBT): Negotiation, Management, and Trading
Business Objects: Bid
Industrial Objects: None

Use Case Id: 9.0
Title: *Get the Lowest Final Bid*

This is the same as extracting the current lowest bid but it is done only after the bid is closed.

CRC Cards:

Description of the Template Used


Name of the Class
Role
Major Responsibility of a Class
Client Classes (request or take services)
Services provided to clients (classes)

UserBuyingRequest (Information Collector)

Responsibility	Collaboration	
Collect the Buying Request Information.	**Clients**	**Server**
	- CarDealer - IndividualUser - Car - RequestHandler - RequestStatus	- Get the request details from the user. - Associate user who initiated it with the request (potential buyer). - Cater information about the buying request and its details.

Bid (Bid Details Collector)		
Responsibility	Collaboration	
Maintain Details of Bids made by User(s)(sellers).	**Clients**	**Server**
	- User - Car - BidResponseHandler - UserBuyingRequest - RequestStatus	- Facilitate the User(s)(seller)to Bid for a particular buying request. - Maintain bid records by Bid Id. - Cater information as biding price, buyer, seller, quantity and time.

RequestStatus (Status Information Provider)		
Responsibility	Collaboration	
Provide Status parameters of the Request.	**Clients**	**Server**
	- User - UserBuyingRequest - Bid	- Confirm Alive/Not Alive Status of the Request - Provide the time remaining for the Request (Bid Period). - Calculate current lowest bid for a request.

User (Seller)		
Responsibility	Collaboration	
Provide Car(s) details for selling.	Clients	Server
	- Car - UserRegistration	- Provide identification as a seller. - Provide the car(s) details intended to be sold.

User (Seller)		
Responsibility	Collaboration	
Responding to requests	Clients	Server
	- Bid - BidResponseHandler	- Bid for the corresponding posted request.

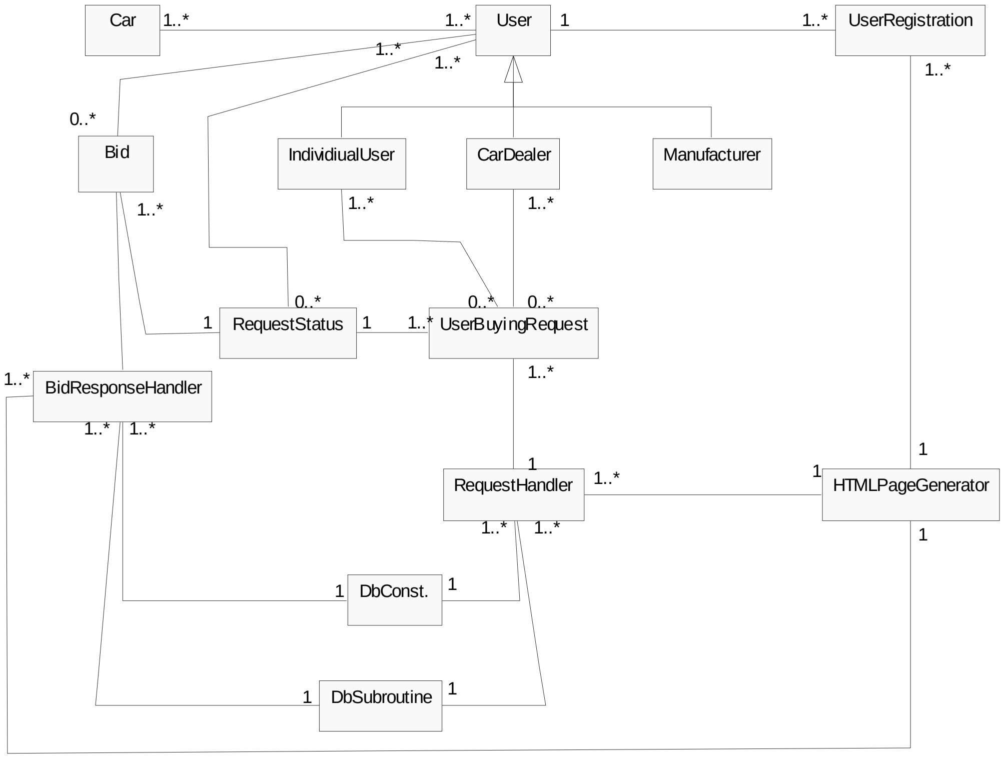
User (Buyer)		
Responsibility	Collaboration	
Provide Details of the Desired Car(s).	Clients	Server
	- UserRegistration - UserBuyingRequest - Car	- Provide identification as a buyer. - Provide the car(s) details intended to buy.

Car (Car detail Storer)		
Responsibility	Collaboration	
Storing the details of a car(s)	**Clients**	**Server**
	- IndividualUser - CarDealer - ManufacturerCompany - UserBuyingRequest	- Store car details. - Provide Car Details to the clients (classes). - Update car information.

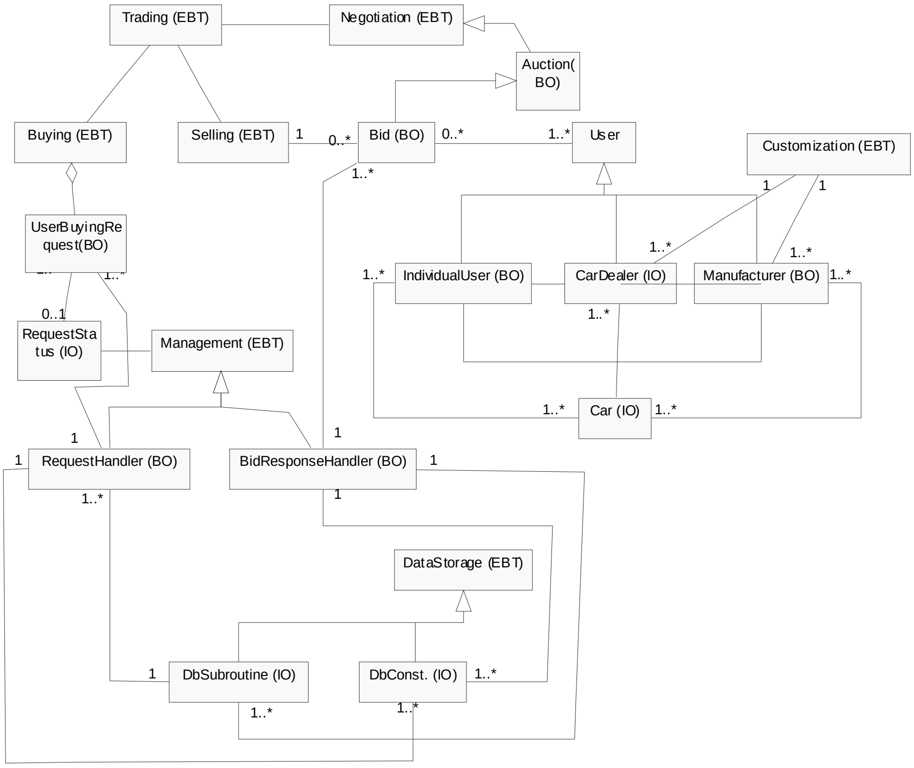
BidResponseHandler (Bid Management)		
Responsibility	**Collaboration**	
Manage the bid(s) for buying request(s).	**Clients**	**Server**
	- User (seller) - Bid	- Gather bid details from Bid class. - Identify the corresponding buyer. - Map the bid with buyer and seller. - Store the bid.

RequestHandler (Request Management)		
Responsibility	**Collaboration**	
Manage the user(buyer) request.	**Clients**	**Server**
	- User (buyer) - UserBuyingRequest	- Gather request details from UserBuyingRequest. - Identify the userId as a buyer. - Store the request.

Traditional Class Diagram:



System Stability Model:

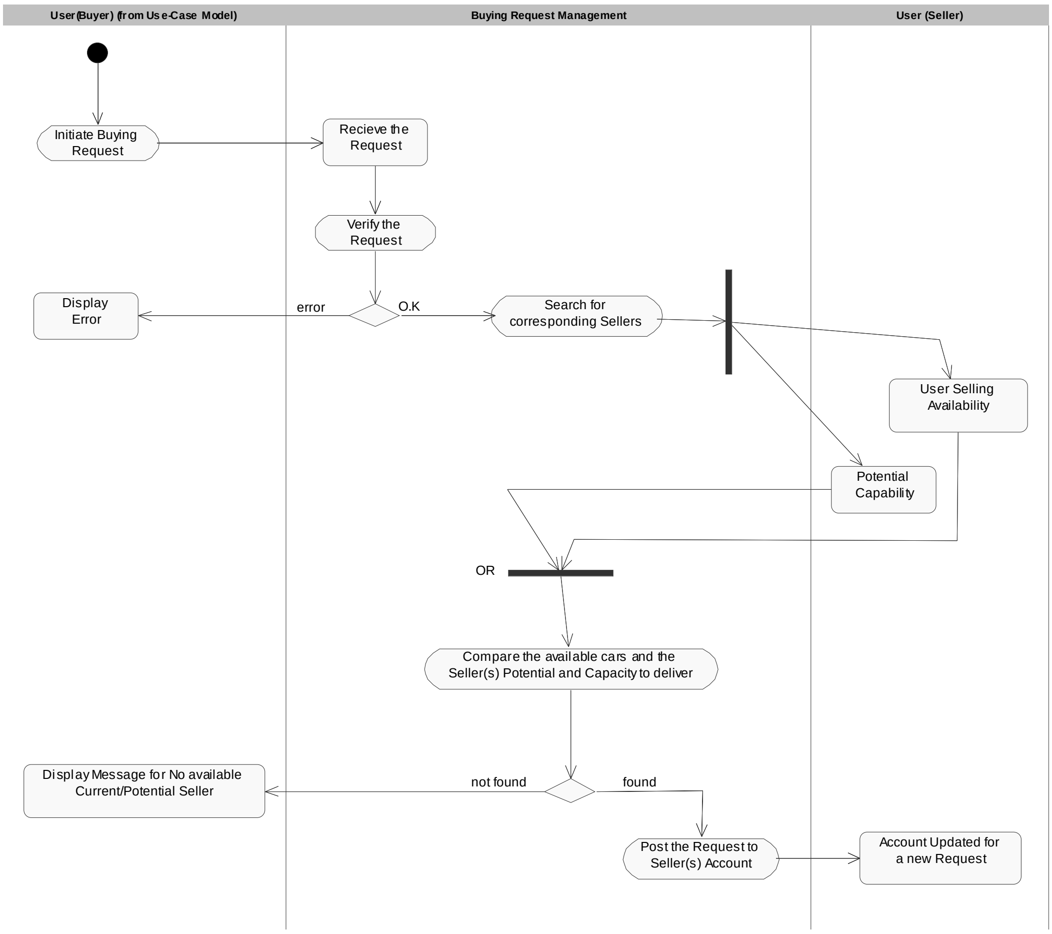


Behavior Diagrams:

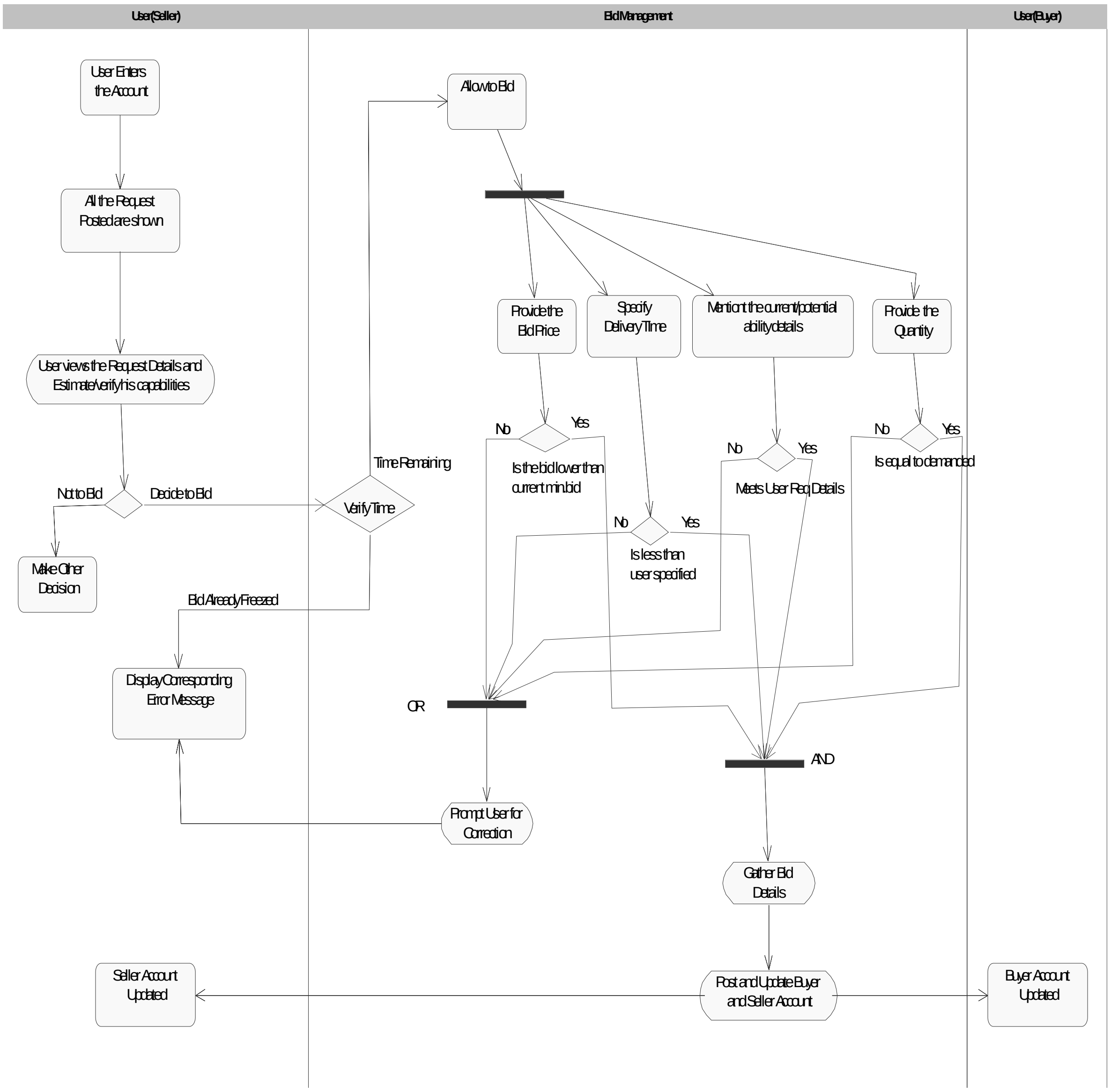
1. Activity Diagram:

Note: because of the similarity between all our activity diagrams, we represent only two of them.

Posting the Buying Request

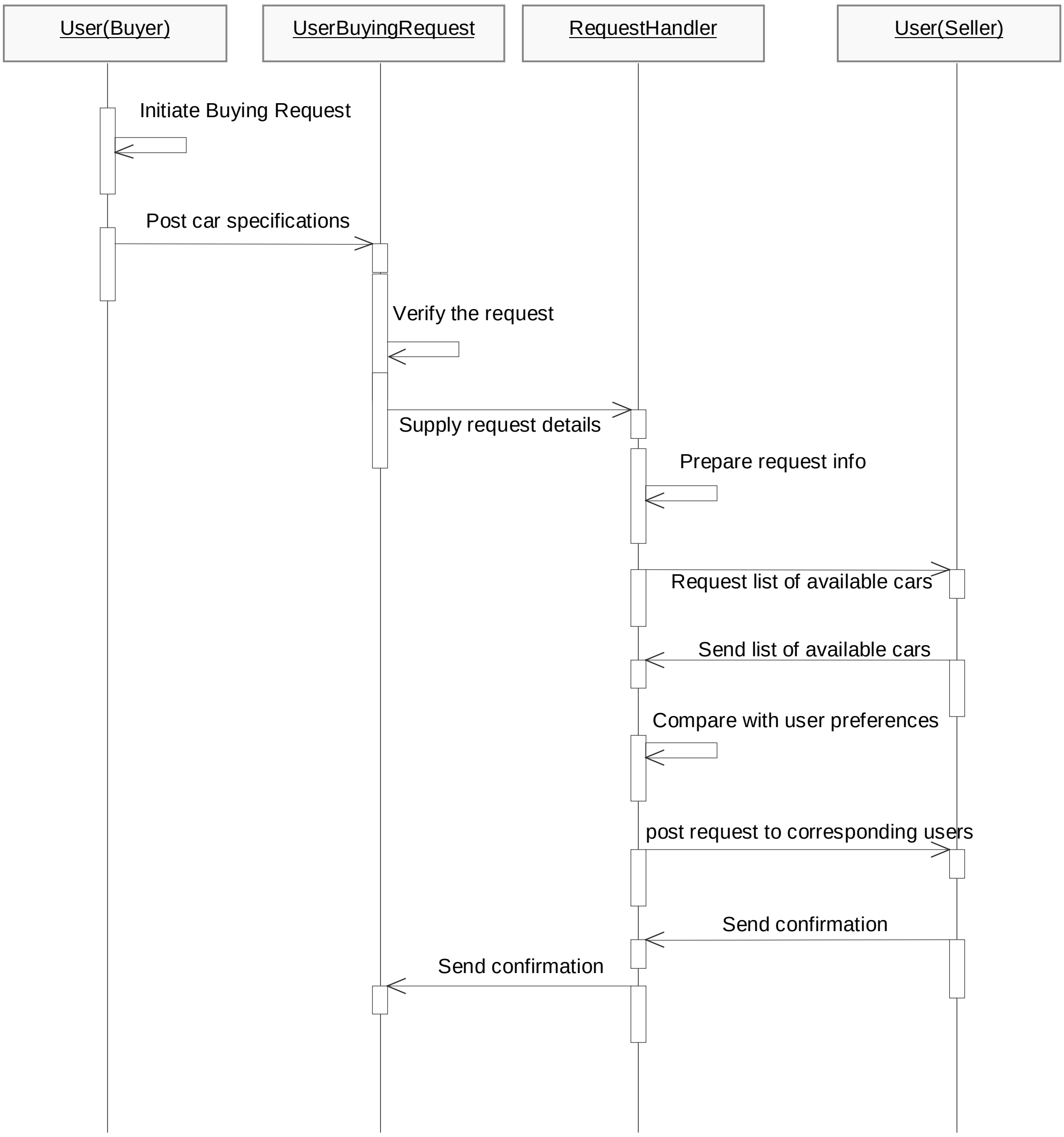

User(Buyer) (from Use-Case Model)
Buying Request Management
User (Seller)
Initiate Buying Request
Recieve the Request
Verify the Request
Display Error
error
O.K
Search for corresponding Sellers
User Selling Availability
Potential Capability
OR
Compare the available cars and the Seller(s) Potential and Capacity to deliver
Display Message for No available Current/Potential Seller
not found
found
Post the Request to Seller(s) Account
Account Updated for a new Request

Biding Process

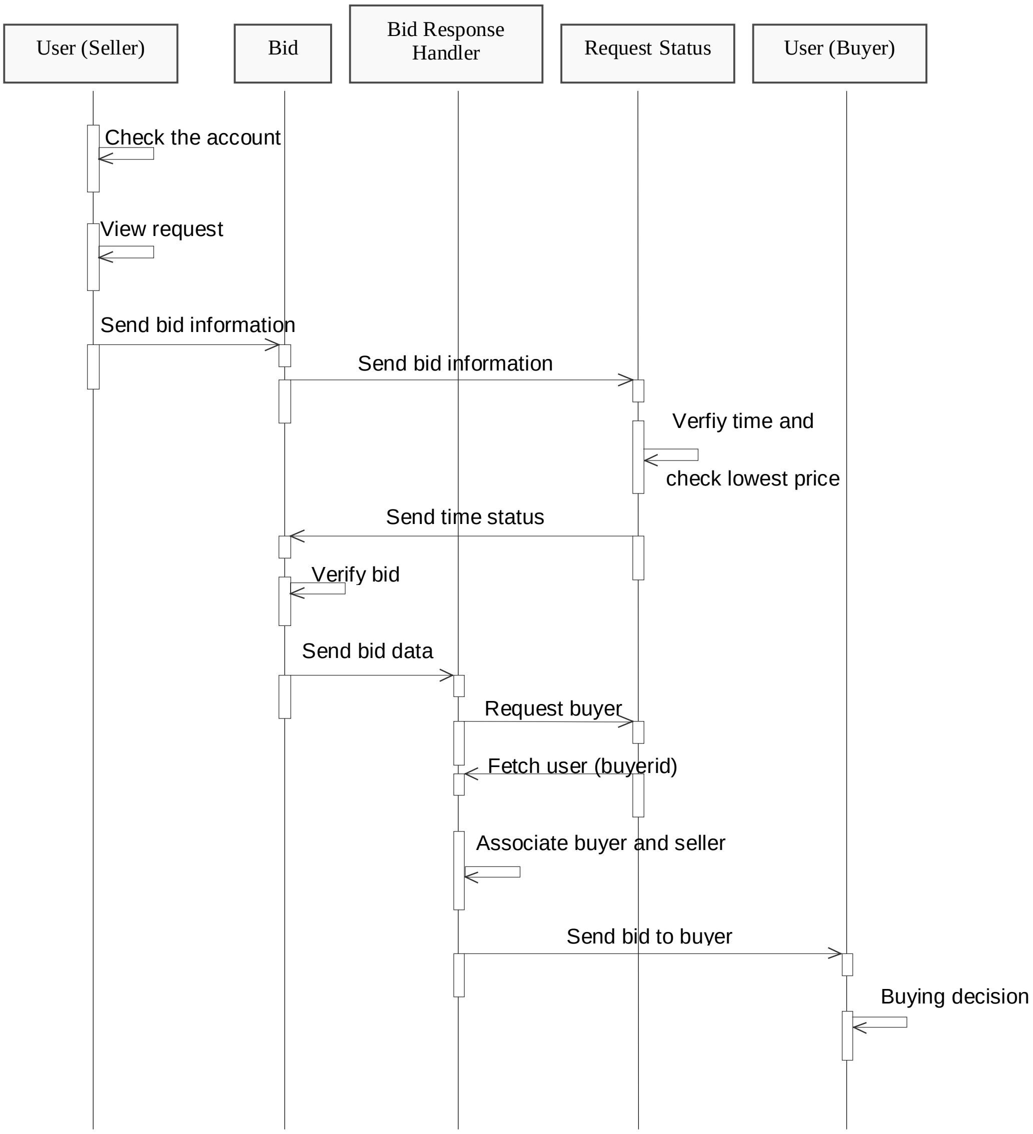


2. Sequence Diagrams:

To Initiate and Post the Buying Request



To View the Posted Request and the Bid Process



Conclusion:

The reverse auction concept has broad potential for a variety of applications, especially in the support of Business-to-Business transactions. The stability parameters for such a system are not obvious. But the strength it provides to the system is invaluable and thus compensates for all the hard work and thoughts put in. The traditional model in our system is not very traditional (well thought of) and thus not many industrial objects change in the stable model. Off course the identification and arrangement of EBT’s and I.O’s provides the stable backbone of the system.

There are some issues that are particular to the implementation preferences as the database models, constants and subroutines. Also there is a major thought process involved in our model and that is the idea of having a module(class) called the HTMLPageGenerator as we are dealing with a web based system. At present these classes are not actively involved in the system but they are present in the class diagrams.

Our modeling is just an instantiation of the approach and thus it looks pretty small and complete but by working on the stability of the system we already have made it very easy to scale and plug and play for other applications also. Thus the system has all the features required to have the scalability and extendibility, also with some effort it could be thought of providing a whole new perspective by not limiting the application to only a particular product.

We would like to thank Dr. Mohamed Fayad, OOPSLA reviewers and all the team members for providing us with this invaluable insight for corrections, suggestions and providing stability model concepts.

References

Books

1. Developing Software with UML, Bernd Oestereich, Bernd Cestereich. Addison Wesley, March 1999
2. Object-Oriented Design Heuristics, A.J. Reil. Addison Wesley Longman, 1996.
3. Objects, Components, and Frameworks with UML: The Catalysis Approach,D.F. D'Souza & A.C. Will Addison Wesley, 1999.

Web References

1. http://www.rational.com/uml/resources/documentation/index.jsp
2. http://www.db.informatik.uni-bremen.de/umlbib/home.html
3. http://www.uml-zone.com/