Online auction system

[redacted]

Submitted as part of the degree of BSc Computer Science to the

Board of Examiners in the School of Engineering and Computing Sciences, Durham University

Abstract —

Context/Background
With the internet being used by an increasing number of people, various different real life concepts have been converted into a digital format. One of these which has been converted and is highly popular is auctions. Although online auctions are highly popular sites, they still suffer from issues of trust, usability and issues which occurred in the original physical equivalent.

Aims
The aims of the project is to investigate and determine whether it is possible to create a fair, secure, trustworthy and user-friendly online auction system which allows multiple concurrent auctions to occur at once, which also allows users to bid on these auctions. The auction system will aim to increase the fairness of the system by attempting to detect unfair behaviours such as shill bidding and trying to prevent reputation manipulation. The usability of the auction system will also be investigated and suitable methods to increase the usability of the system will be implemented. As a secondary aim, an auction software agent will be developed to aid the usability of the site which is capable of monitoring auctions and bidding on a user's behalf.

Method
An online auction system will be developed which provides a user interface allowing users to interact with the auction system, appropriate auction protocols will be implemented to determine how the auctions are carried out. Appropriate algorithms will also be implemented to detect shill bidding, calculate reputation scores and carry out the operations of the auction software agent. After the functionality of the auction system is complete, further changes will be investigated to determine if there is further ways to increase the usability of the system.

Proposed solution
The proposed solution is to create an online auction system as a website, which will provide the interface which users will use to interact with the auction system. This interface will provide registering, logging in, searching, auctions and auction creation screens. This website will incorporate the reputation calculation algorithm as well as the shill detection algorithm. Also a separate program will also be developed to implement the software agent which is capable of interacting with the auction system and placing bids on the user's behalf.

Keywords — Online auction, security, auction protocol, software agent, website, shill bidding, trust, reputation

I INTRODUCTION

A Auctions

Auctions have been used around the world in order to sell items to buyers from groups of bidders. The method in which auctions are carried out, also known as auction protocols, has varied across different countries. In most cases, auctions would last a short while (a number of minutes) and would take place at a fixed location and time. In order for potential bidders to participate in these auctions, they would be required to be at the auction during the time specified and if bidders wanted to evaluate the items for auction, they would usually have access to them a number of days before.

B Auction protocols

Out of all the best known auction protocols, such as English, Dutch, Japanese and first and second sealed auction protocol, the English auction protocol is the most popular across the bigger online auction systems such as eBay. Due to it's popularity, this project will focus primarily on the English auction protocol, although the second-price sealed auction protocol will also be implemented to increase the functionality and choice of the auction system;

The *English auction protocol* is an auction protocol in which bidders are able to bid as many times as they wish as long as their bid is higher than the current highest bid. This continues until there is no bidders willing to out bid the highest current bid, at which point the auction is completed and the highest bidder is the winner of the auction. (Atallah and Blanton, 2009)

The *second-price sealed bid auction protocol* is a variation of the first-price sealed bid auction protocol. Each bidder within the auction places a bid which is concealed from all other bidders and is handed to the auctioneer. After a fixed period of time is complete or all bidders who wish to bid have placed a bid, the auctioneer reviews all the bids and the bidder with the highest bid wins the auction but pays the second highest bid for the item. (Atallah and Blanton, 2009)

C Online auctions

With the rise in popularity of the internet, e-commerce has become a major part of life. This has led to aspects of life being converted into a digital format, a key example of this is auctions. With the use of online auctions there is no longer a need for bidders to gather at a certain location and users can bid from around the world. The auctions also tend to run longer than traditional auctions (a lot of the time for a number of days) which allows bidders to place bids in advance. A popular example of an online auction would be eBay, which handles over $57 billion every year (Beyene et al., 2008) and was the 20th most visited website in January 2015 (Alexa.com, 2015). Sites such as eBay run similar to traditional auctions in the sense they run according to an auction protocol, allow users to place bids on items and allow users to put items on the site for auction. The crucial difference with online auctions is that they require a lot more on the trust between the person auctioning the item and the buyer, as bidders are unable to view the item physically beforehand and has to make judgements based on information and images provided by the person selling the item. The seller and the successful bidder also have to trust in each to complete the transaction of the funds as well as the item.

Although online auctions have thrived in recent years they still to suffer from the issues, with auction fraud making up 35.7% of total complaints to the Internet Crime and Complaint Centre

in 2007 and consisting of actions such as ”non-delivery of goods”, ”item misrepresentation”, ”selling black-market goods” and ”shill bidding” (Mamun and Sadaoui, 2013). As well as the issue of auction fraud, a study (Lin, Tu and Fang, 2008) on people who do not use online auctions and why they do not use online auctions, identified several factors which ultimately led to a sense of ”risk” and therefore deterred them from using online auctions. The factors identified included, ease of use, product uncertainty and wasting of time. These issues highlight the challenge faced by online auctions and the issues which online auctions need to overcome to ensure they continue to be popular and successful.

Shill bidding is a key method of auction fraud and a crucial issue which effects popular auction protocols used by online auctions. ”Shilling is used for a particular auction item to artificially inflate the price in case of forward auctions, and deflate the price in case of reverse auctions” (Mamun and Sadaoui, 2013), which in the cases of the English auction protocol would mean users would purposely placing false bids, which they expect to be beaten, in order to raise the value of the auction. This artificial method of increasing the price of the auction makes the auction unfair on others genuinely trying to participate within the auction and makes users pay more for the auction.

”Ease of use” is an common issue faced by many websites across the world and doesn't have a universal solution. When it comes to specifically online auctions there are several areas which makes the using the system tedious and difficult such as searching, bidding quickly and monitoring multiple auctions. For example, when it comes to monitoring multiple auctions, a user may identify several auctions they could be interested in and therefore has to monitor them all. These auctions could all end at different times of the day, and due to the tactic of bid sniping, where ”a bidder bids the minimal amount required to win in the closing seconds of an auction thereby denying other bidders time to react and keeping the price as low as possible” (Balingit, Trevathan and Read, 2009) it often means that the user is required to be active on the auction system at the end of the auction to ensure they have a fair chance of counteracting this tactic. Therefore there are multiple areas where the usability of an auction system can be investigated and the ”ease of use” which ultimately comes back to the usability of the system can be addressed. Addressing the usability of the auction system is crucial to enabling this project and auction system to be a success.

The issues of ”product uncertainty” and ”risk” (Lin, Tu and Fang, 2008) again leads to the important issue of trust within online auctions due to not physically being able to evaluate the item before placing a bid and not being able to obtain the item straight after the auction ends. Online sites such as eBay have attempted to deal with these issues by providing online ratings of users, which are used to help potential bidders determine whether or not a user can be trusted. These trust or reputation systems often use a simple averaging model which can ”bring about many problems such as reputation collusion and unfair ratings” (Zhang, 2007) meaning that such simple reputation systems often used by online auction sites can be easily manipulated into giving inaccurate ratings which could potentially trick users of the site into thinking a user is more trustworthy than they truly are. If a reputation system can be easily manipulated it can damage the reputation of the site which then feeds back into the issue of user's not trusting online auctions. Therefore the issue of trust within online auctions is a critical issue which requires to be looked into.

D Project purpose

The purpose of this project is to investigate the issues challenging online auctions. These issues include: lack of trust in online auctions, fairness within the auction system and usability of the auction system. To ensure that an auction system is fair, the system must give no unfair advantage to any user of the system should try to prevent any non-genuine actions taken within auctions which could alter the outcome or price of the auction. Therefore this project will seek the answer to "Is it possible to build an online auction system which is usable by a wide range of users as well as being trustworthy and fair?".

In order to evaluate the usability of the auction system as a whole, a user study will be conducted on a wide range of histories of online auction usage in order to determine the success of the implementation. To test the reputation calculations and shilling detection algorithm, fixed sets of data with predetermined outcomes will be used to verify the out comes of the algorithms.

E Deliverables

Deliverable	**Description**
Basic	Implement web-based auction system with English auction protocol which allows users to bid on auctions as well as create auctions with item names, descriptions, start price and optional image upload.
Basic	Investigate issues affecting usability and implement suitable improvements to the auction system to assist users using the web-based auction site.
Basic	Integrate a program within the auction system which on completion of an auction, sends an email to all bidders and exchanges details with the seller and winning bidder.
Intermediate	Implement a shilling detection algorithm, which is capable of reliably and accurately detecting shill bidding and blocking the bids.
Intermediate	Implement the second-price sealed auction protocol in the auction system.
Intermediate	Implement a reputation algorithm for all users which cannot be easily manipulated to give inaccurate reputation scores.
Advanced	Design and develop an auction software agent which is capable of monitoring multiple auctions on the user's behalf and also bid on the user's behalf.
Advanced	Evaluate the trust algorithm and shilling prevention algorithm against fixed data sets with predetermined outcomes.
Advanced	Evaluate the entirety of the auction system with a test group of users.

II DESIGN

A Tools

For this project there will be a heavy focus on web development along with desktop programming for the separate programs that run along side the main website. The website and the API will be developed using Dreamweaver CS5, which is an all-in-one IDE which has support for PHP, HTML, CSS and Javascript which are all the languages which will be used in order to develop the web proportion of the project.

To create the database, PHPMyAdmin will be used in order to visually create the database structure without having to manually write the DDL (Data definition language) queries. This software also allows for quick modification to any part of the database, allowing for quick database development.

In order to create the desktop programs which will be running alongside the website, Eclipse will be used to develop the Java applications. Eclipse is an IDE which has support for many different languages as well as a number of plugins. One of the most important features of Eclipse for this project, is that it supports the plugin "WindowBuilder". WindowBuilder is visual GUI editor, allowing for visual creation and editing of graphical user interfaces in Java. This plugin will significantly decrease the time taken to create the GUI for the auction software agent and allow more focus upon the actual functionality of the agent rather than the GUI.

B Life cycle and maintenance

Due to the project focusing upon the usability of the system and extensive testing to ensure each of the components of the system are functioning correctly, an agile development approach will be taken. This will allow for the algorithms to be tested adequately throughout the project as well as allowing users to give feedback on the usability of the system before the development is finished. This feedback during development will be fed back into the development to ensuring that the usability part of the project goal is achieved.

To ensure that the system will be easily maintainable within the short to medium term, the languages chosen to develop the auction system were selected in part due to their wide support, well documentation and no obvious signs that they are or will become redundant in the near future. In order to ensure that any part of the auction system can be updated, maintained and extended, the programs will be built up of smaller files which contain part of the implementation. In the case of the Java programs, this is known as object orientated programming, which enables the program to be broken down into smaller files called classes which allows the program to be updated without requiring to change multiple files. A similar approach will be used with the core PHP files which make up the website and API, the website will be broken down into separate PHP files each providing different functionality. These separate PHP files then can be edited separate to update sections of the website without necessarily modifying the rest of the website or API.

C System architecture

Figure 1 shows an outline of the system architecture that will be developed to implement the online auction system. The system itself is broken up into many smaller components which all work together in order to build up the entire system. The website and the API will both be developed using PHP due to past experience using this server-side scripting language as well as being a well documented and well supported language. The programs, "auction monitoring software" and "software auction agent" will both be developed in Java. Java has been chosen due to previous experience with the language as well as it's cross-compatibility, well documented, well supported, number of user created libraries and also has a large install base internationally.

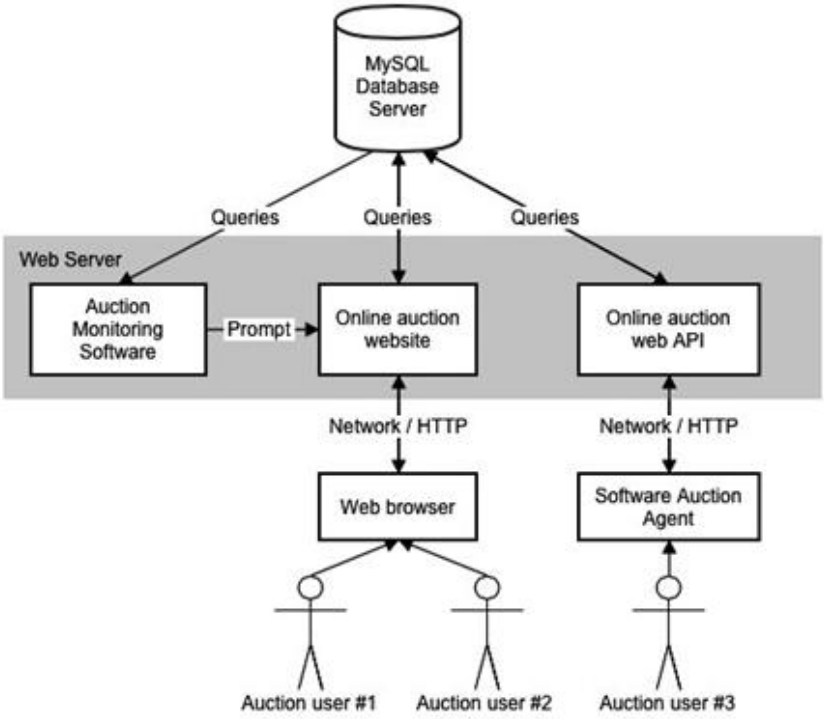

Figure 1: Architecture of the auction system

D Requirements

NFR / FR ID	Description	Priority
FR1	Implement web-based online auction system with English auction protocol	High
FR2	Implement a reputation algorithm	High
FR3	Implement a shilling detection algorithm	High
FR4	Create a program to sends a prompt to send emails out when an auction ends to all involved	High
FR5	Create a software agent capable of monitoring auctions & bidding on auctions	Medium
FR6	Auction creation should allow an image to be uploaded	Medium
FR7	Searching should update part of the page and not have to reload the entire page	Low
FR8	Placing a bid should not require a page refresh	Low
FR8	Users not logged in should not be able to bid or create auctions	Medium
FR9	Users should be able to place bids depending on the auction protocol used	High
NFR1	Auction website should support the following browsers: Chrome 39.0+, Internet Explorer 8+, Firefox 32+ and Safari 7+	Medium
NFR2	Improve aspects of website to aid usability	Medium
NFR3	User interface on the auction website should be consistent	Low
NFR4	Passwords in the database should be hashed and the actual password not stored	High

E MySQL Database

A database-driven approach will be taken in order to store the data for the auction system. The use of MySQL allows support across a number of different programming language via built in functionality or libraries, quick development of a data storage solution and efficient data access. The database designed, see **figure 3**, has been normalised to minimise data duplications in order to save space and help with performance over the auction system's lifetime.

The *users table* will contain the information on the user. To increase the security of the passwords stored in the database are hashed. Upon registering a users password will have a salt randomly generated by the server and appended onto their password, it will then be hashed using the hashing algorithm, SHA256.

The *auctions table* stores the detail about all the auctions. UserID is a foreign key which belongs to a user in the users table who initiated the auction. Protocol stores the name of the protocol being used for the auction. ItemImageURL stores the path to the image of the item being auctioned, if the user chose not to upload an image, this value is NULL. StartPrice is the starting price of the auction and finally, EndTimeStamp is the Unix timestamp for when the auction will end.

The *bids table* stores all the bids made on the auction system. The UserID is a foreign key which links the bid to the user who placed the bid. AuctionID is another foreign key which links the bid to the auction it was placed on. Amount stores the value of the users bid and TimeStamp stores the Unix timestamp when this bid occurred.

Within the *feedback table*, AuctionID stores the ID of the auction of which the feedback belongs to. ToID stores the UserID of the user who the feedback is about and FromID stores the UserID who placed the feedback. The Rating column stores the numerical rating which indicates how the user felt about the auction, with an optional Message column to store a message about the auction for other users to see. The TimeStamp column stores the Unix timestamp of when the feedback was left.

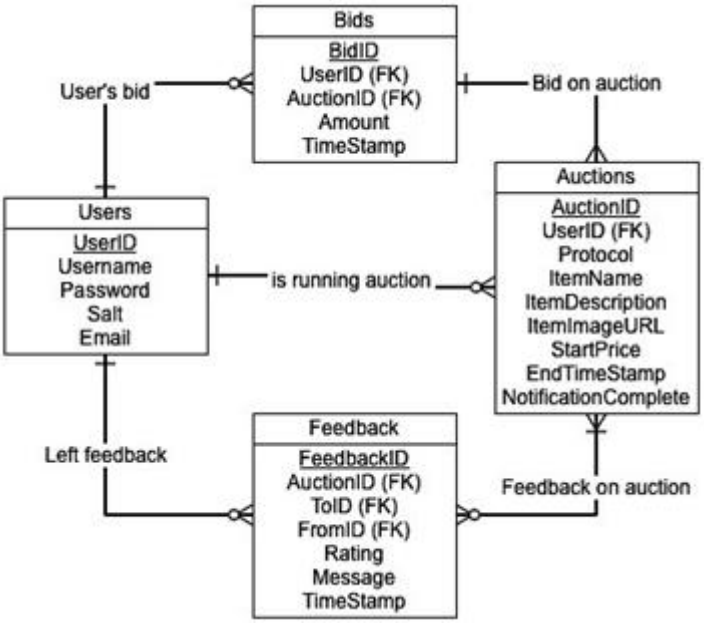

Figure 2: Entity-relationship diagram of auction system database

F Online auction website and API

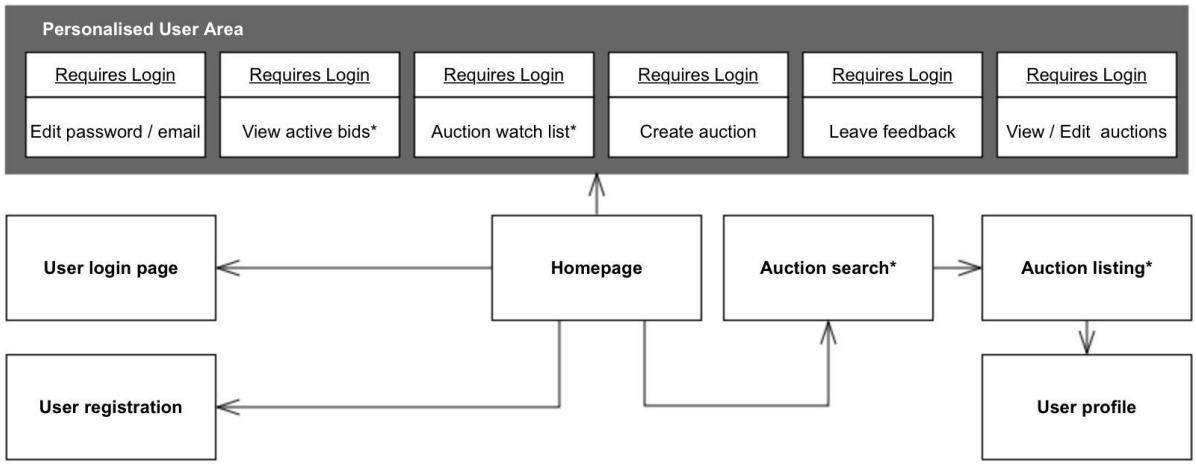

Figure 3: Site map outline, * indicates bids can be made from this page if user is logged in

Both the website and API will be constructed using PHP which will generate dynamic HTML (with CSS) pages or in the case of the API, JSON formatted data. To aid in the construction of the website two open-source pieces of software will be used, jQuery and Bootstrap. jQuery is a Javascript library which allows for complicated Javascript features to built quickly and easily. Bootstrap is a HTML, CSS, Javascript framework allowing for a standardised user interface across the website as well as helping the development of the website by providing pre-designed user interface elements ready to use. Using this framework and library will also allow for cross-browser compatible to be achieved due to both supporting a vast number of internet browsers. Ajax will also be used on a number of pages of the website which will allow asynchronous HTTP requests which when used in conjunction with Javascript (or jQuery) can be used to dynamically update the web pages of the site without reloading the entire website. This method of asynchronously loading new content will be used across the site on areas where reloading the entire page isn't necessary or timing is vital (such as searching auctions and bidding), this will be used primarily to increase the usability of the site. See **figure 3** for the site map of the website.

The "Auction listing" page is the page generated for an auction on the website, depending on the auction the user wishes to view the page will be generated. For example if a user clicks the title of an auction, it will bring them to a page where they can get more information about the auction, such as the description, a bigger picture of the item, seller reputation score and various other pieces of information.

The "Auction search" will dynamically generate auction listings from all the auctions currently occurring on the auction system depending on the search criteria inputted by the user. This page will have a central element within the page which is dynamically updated when the parameters of the search has been changed. This will be achieved by using Ajax, which will send an asynchronous HTTP request to a separate PHP file which will then generate the correct HTML for the search, this will be returned by the Ajax and will update an element on the search page.

In the auction site, there will be multiple pages listing auctions. Within these listings, to help with the usability of the site, the user will be able to place a bid without requiring to load the listing page. This will be accomplished by using Ajax from the listing element on the page to

send a request to a separate PHP file and if successful it will display a visible notification on the website. Pages which contain this feature are denoted by a * symbol in the site map.

”Auction watch list” and ”View auction bids” are central pages which display the listings the user has placed in their watch list or have actively placed a bid in. These pages are central to helping users monitor all the auctions they’re interested in and will as stated above, allow the user to bid directly on the listings on these pages without having to load the entire listing page.

The API is a small add-on to the website, it provides an interface for which the auction software agent can interact with the auction system. It will response to all HTTP requests using the JSON (Javascript object notation) data format, which is a data format which is easily readable by both machines and humans (Json.org, 2015). The API will provide the following functionality: logging in, searching auctions, placing bids and returning the status of auctions.

G Software auction agent

The auction agent is software which is capable of placing bids on behalf of a user. It’s primary goal is to increase the usability of the auction system by allowing users to set a criteria of item they’re looking for and then the agent will monitor the auction system non-stop until it has managed to purchase the item. With all agents, an agent is attempting to maximise a utility function, which is a method of numerically measuring it’s success, in the case of the auction agent the difference in the item’s value to the bidder and the amount paid for it in the auction is the utility function it will attempt maximise.

The way in which the auction agent function is by providing an graphical user interface which will allow the user to input an item criteria. The agent will then send a HTTP request to the auction API periodically, which will provide a JSON formatted list of relevant auctions on-going on the auction system. This list will be stored locally and used with an appropriate heuristic function which will be experimented to select, if any, an auction to place a bid on. It will then keep monitoring this auction to check if it is out bided or if the auction is complete. If the agent is out bided it will go back to start and look at all auctions again. If the auction ends with the agent being the highest bid the program will stop monitoring the auction API and will leave a prompt alerting the user of it’s success.

The heuristic function used to determine if there is an auction which the agent should place a bid on will be experimented with during the development of the agent and several incarnations will be tested on a fixed set of data to see which maximises the utility function for a number of concurrent agents.

A suitable intelligent agent described in (Sow, Anthony and Mun Ho, 2010), will be used as base for this software agent and modified in order to produce the auction agent capable of the requirements in this project. See **algorithm 1** for pseudo code of the modified agent.

H Auction monitoring software

The auction monitoring software is a small, lightweight program written in Java and will run in the command line (in order to reduce the workload which a GUI would add). The program regularly checks the database for new auctions and stores the end time of the auctions locally, when an auction ends it sends a HTTP request to a separate PHP file on the web server. This PHP file will then check all auctions recently finished and send notifications, in the form of an email, to all those involved within the auctions which have ended.

Algorithm 1 Auction agent pseudo code

```
1: function AUCTIONAGENT
2:     item ← User defined criteria, including name, worth, minimum seller reputation
3:     item_bought, userStop ← False
4:     lastCheck, activeAuctionId ← -1
5:     auctionList ← Null
6:     while item_bought ≠ True OR userStop≠ True do
7:         currentTimestamp ← Current timestamp
8:         if lastCheck == −1 OR (currentTimestamp − lastCheck) ≥ 5 minutes then
9:             auctionList ← Query API with item name and update
10:            lastCheck ← currentTimestamp
11:        end if
12:        if activeAuctionId == −1 then
13:            selectedAuction ← HEURISTICAUCTIONSELECT()
14:            newBid ← selectedAuction_Highestbid+ 0.1 × item_worth
15:            if newBid ≤ item_worth then PLACEBID()
16:                activeAuctionId ← selectedAuction_ID
17:            end if
18:        end if
19:        if activeAuctionId ≠ -1 then
20:            if ISHIGHESTBID(activeAuctionId) == False then
21:                activeAuctionId ← -1
22:            else
23:                if ISAUCTIONOVER(activeAuctionId) == True then
24:                    item_bought ← True
25:                end if
26:            end if
27:        end if
28:    end while
29: end function
```

I Reputation system

The reputation system is used to determine a value for each user based on their past participation in auctions, as both a buyer and a seller. The primary goal of using a reputation system is to increase the level of trust within the auction system and aid users when determining whether or not to trust an auction. Sites such as eBay use a basic averaging algorithm which takes into account all auctions a user has participated in and adds up all the feedback then divides by the number of auctions they've been in. As described in the introduction, this has been found to be vulnerable to certain techniques to allow users to seem more trust worthy than they truly are. Therefore for this auction system a reputation system will be implemented which uses methods to weigh the feedback from users based on multiple factors. Using elements as outlined in (Wei Zhang, 2007), a reputation algorithm has been constructed, these elements are:

#	Weighting Description
1	Using a time window to discount older feedback.
2	Give weightings to feedback based on it's age, with older feedback weighted less.
3	Size of transaction weighting, giving more weight for auctions which are won with large bids.
4	Feedback left by others will be weighted according to their own feedback, with highly reputable member's feedback rated better than those with a low or poor reputation.

$$\sum_{i}^{Feedback} Score_i \times ReputationWeighting_i \times TimeWeighting_i \times ValueWeighting_i \quad (1)$$

Above is an outline of the reputation calculation, it loops through all the feedback within the time window and adds up the value of the feedback. The "Score" is the value left by the user, either a position (+1), neutral (0) or negative (-1) score. This is then multiplied by the "ReputationWeighting" which is influenced by the reputation score of the user who left the feedback. The "TimeWeighting" is a simple linear weighting, with those closer to the start of the feedback window weighted more than those at the end. "ValueWeighting" is determined by the value of the transaction of the auction, this means an auction ending with a bid of 1000 will be weighted significantly more than one for 1. During the agile development, the method in which these weightings are distributed will be experimented with against fixed sets of data to try and optimise and ensure that they are accurate of the actions of the users.

J Shilling detection

Shill bidding as explained above is a method used to inflate the price of the auction which makes the seller more profit and the bidders pay more to win the auction. As this is an unfair method, this auction system will evaluate the bidder when a new bid is placed by running the bid and the user through the shill detection algorithm. As outlined in (Mamun and Sadaoui, 2013) there are several behaviours that a shill bidder may display, the behaviours the shilling algorithm will look for will include:

1	Placing bids to purposely outbid themselves to inflate the price.
2	Placing large bids significantly larger than a bid required to outbid the current bid.
3	Using multiple accounts to place bids and may or may not change their IP address as well.
4	Stopping bidding after a set amount of time & bidding more in the first half of the auction.
5	Placing a large number of bids in short amount of time

Using these behaviours and using an algorithm based on the algorithm which is outlined in (Mamun and Sadaoui, 2013), a shill detection algorithm will be implemented into the auction system. This algorithm uses thresholds to determine whether or not a user has displayed the behaviours sufficiently to suspect them of a "shilling behaviour" and the number of behaviours is added up numerical. If the user displays enough of "shilling behaviours" the user is flagged as a shill bidder and appropriate action (banning or preventing the bids) will be taken. During the testing phases, the shill detection will be ran against this fixed set of data and the thresholds will be adjusted until a adequate level of accuracy is achieved. For the pseudo code of the outline of the shill detection algorithm see **algorithm 2**.

Algorithm 2 Shilling detection algorithm

function SHILLDETECTION($auction$, $ipAddress$, $user$, bid, $shillingScore$ = 0)
    $maxShillingScore \leftarrow$ Threshold of how many behaviours before detected as a shill bid
    $outbidSelfThreshold \leftarrow$ Threshold for number of times allowed to outbid self
    $outbiddingThreshold \leftarrow$ Threshold of time between bidding after being out bided
    $biddingFreqThreshold \leftarrow$ Threshold of bidding frequency relative to auction time
    $outbiddingChangeThreshold \leftarrow$ Threshold % change in auction price with new bid
    if $ipAddress$ == $auction_{sellerIP}$ OR $ipAddress$ in $auction_{otherBidderIPs}$ **then**
        $shillingScore \leftarrow shillingScore + 1$
    end if
    if USEROUTBIDDEDSELF($auction$, $user$) $\geq outbidSelfThreshold$ **then**
        $shillingScore \leftarrow shillingScore + 1$
    end if
    if $auction_{timeSinceLastBid} \geq outbiddingThreshold$ **then**
        $shillingScore \leftarrow shillingScore + 1$
    end if
    if bid - $auction_{currentBid} \geq (outbiddingChangeThreshold \times auction_{currentBid})$ **then**
        $shillingScore \leftarrow shillingScore + 1$
    end if
     return $shillingScore \geq maxShillingScore$
end function

References

Alexa.com, (2015). ebay.com Site Overview. [online] Available at: http://www.alexa.com/siteinfo/ebay.com [Accessed 7 Jan. 2015].

Atallah, M. and Blanton, M. (2009). Algorithms and theory of computation handbook. London: Chapman & Hall.

Balingit, Rodel, Jarrod Trevathan, and Wayne Read. 'Analysing Bidding Trends In Online Auctions'. 2009 Sixth International Conference on Information Technology: New Generations (2009): 928-933.

Beyene, Yordanos et al. 'The Ebay Graph: How Do Online Auction Users Interact?'. IEEE INFOCOM 2008 - IEEE Conference on Computer Communications Workshops (2008): 1-6.

Json.org, (2015). JSON. [online] Available at: http://www.json.org/ [Accessed 9 Jan. 2015].

Lin, Chwen-Yea, Chien-Chung Tu, and Kwoting Fang. 'An Explore Study Of Non-Using Online-Auction People'. 2008 IEEE International Conference on Industrial Engineering and Engineering Management (2008): 1456-1459.

Mamun, Kazi, and Samira Sadaoui. 'Combating Shill Bidding In Online Auctions'. Information Society (i-Society), 2013 International Conference on (2013): 170-176.

Sow, Jacob, Patricia Anthony, and Chong Mun Ho. 'Competition Among Intelligent Agents And Standard Bidders With Different Risk Behaviors In Simulated English Auction Marketplace'. 2010 IEEE International Conference on Systems, Man and Cybernetics (2010): 1022-1028.

Zhang, Wei. 'Computational Trust Model In Online Auctions'. 2007 International Conference on Wireless Communications, Networking and Mobile Computing (2007): 3767-3770.