



Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων
Department of Computer & Communication Engineering



***ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ
ΑΓΓΛΙΚΟΥ ΠΡΟΤΥΠΟΥ ΔΗΜΟΠΡΑΣΙΑΣ ΚΑΙ
ΠΡΟΣΟΜΟΙΩΣΗ ΤΗΣ ΜΕΣΩ ΠΡΑΚΤΟΡΩΝ***



Πανεπιστήμιο Θεσσαλίας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΖΗΣΟ ΕΛΤΟΝ

Υπεύθυνη Καθηγήτρια :
Δασκαλοπούλου Ασπασία

ΒΟΛΟΣ 2010

ΠΕΡΙΕΧΟΜΕΝΑ

<u>1. Εισαγωγή</u>	<u>4</u>
<u>2. Θεωρητικό Υπόβαθρο</u>	<u>6</u>
<u>2.1 Δημοπρασίες</u>	<u>7</u>
<u>2.1.1 Ορισμός της Δημοπρασίας – Ιστορική Αναδρομή</u>	<u>7</u>
<u>2.1.2 Κατηγορίες και Παράμετροι Δημοπρασιών</u>	<u>8</u>
<u>2.1.3 Ηλεκτρονικές Δημοπρασίες</u>	<u>10</u>
<u>2.2 FIPA English Auction</u>	<u>12</u>
<u>2.2.1 The Foundation of Intelligent Physical Agents (FIPA)</u>	<u>12</u>
<u>2.2.2 Το Πρωτόκολλο της Αγγλικής Δημοπρασίας (English Auction)</u>	<u>13</u>
<u>2.2.2.1 Αναπαράσταση του Πρωτοκόλλου μέσω UML – AUML</u>	<u>13</u>
<u>2.2.2.2 Περιγραφή του Πρωτοκόλλου</u>	<u>17</u>
<u>2.3 Το Μοντέλο Client – Server</u>	<u>18</u>
<u>3. Χρήσιμα Εργαλεία</u>	<u>19</u>
<u>3.1 Uniform Server</u>	<u>20</u>
<u>3.2 Html, Php, MySql και JavaScript</u>	<u>21</u>
<u>4. Ανάπτυξη Λογισμικού</u>	<u>24</u>
<u>4.1 Σχεδίαση και Δημιουργία κατάλληλης Βάσης Δεδομένων μέσω phpMyAdmin</u>	<u>25</u>
<u>4.2 Υλοποίηση Διαδικτυακής Αγγλικής Δημοπρασίας</u>	<u>27</u>
<u>4.2.1 Αρχικοποίηση Συστήματος</u>	<u>27</u>
<u>4.2.2 Εγγραφή Νέου Αγοραστή στη Δημοπρασία</u>	<u>28</u>
<u>4.2.3 Ο Διοργανωτής της Δημοπρασίας (auctioneer)</u>	<u>29</u>
<u>4.2.3.1 Εκκίνηση της Δημοπρασίας</u>	<u>29</u>
<u>4.2.3.2 Διαχείριση Νέας Προσφοράς από Αγοραστή</u>	<u>30</u>
<u>4.2.3.3 Λήξη Δημοπρασίας - Ανακήρυξη Νικητή</u>	<u>31</u>
<u>4.2.4 Οι Αγοραστές (bidders)</u>	<u>33</u>
<u>4.3 Προσομοίωση Αγγλικής Δημοπρασίας</u>	<u>37</u>
<u>4.3.1 Αρχικοποίηση Πρακτόρων – Συστήματος</u>	<u>38</u>
<u>4.3.2 Λειτουργία Πρακτόρων - Λήψη αποφάσεων</u>	<u>41</u>
<u>5. Συμπεράσματα και Προτάσεις για Μελλοντική Έρευνα</u>	<u>44</u>
<u>Παράρτημα Α</u>	<u>47</u>
<u>Παράρτημα Β</u>	<u>56</u>
<u>Αναφορές – Βιβλιογραφία</u>	<u>63</u>

Ευχαριστίες,

Ολοκληρώνοντας την παρούσα διπλωματική εργασία που σηματοδοτεί και το τέλος των προπτυχιακών σπουδών μου, θα ήθελα να ευχαριστήσω την οικογένειά μου, για την υλική και ψυχολογική ενίσχυση που μου προσέφερε όλα τα χρόνια των σπουδών μου καθώς και την υπεύθυνη καθηγήτρια της διπλωματικής μου εργασίας κ. Δασκαλοπούλου Ασπασία, που με βοήθησε με κάθε τρόπο να ανταπεξέλθω στις ανάγκες της παρούσας εργασίας.

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Αντικείμενο της παρούσας Διπλωματικής Εργασίας αποτελεί η ηλεκτρονική δημοπρασία και πιο συγκεκριμένα το FIPA English Auction Protocol. Σκοπός της εργασίας είναι η υλοποίηση του παραπάνω πρωτοκόλλου με απώτερο σκοπό την προσομοίωση του με αγοραστές – πράκτορες. Για την υλοποίηση του επιλέχθηκε η ανάπτυξη μίας διαδικτυακής εφαρμογής Αγγλικής Δημοπρασίας που στη συνέχεια χρησιμοποιήθηκε και στη διαδικασία της προσομοίωσης.

Αρχικά αποκτήθηκε το απαραίτητο θεωρητικό υπόβαθρο που επιβάλλουν η ανάγκες της συγκεκριμένης εργασίας. Τι είναι δημοπρασία, ποια είναι τα είδη της και τα χαρακτηριστικά της, σε τι διαφέρουν οι ηλεκτρονικές σε σχέση με τις κοινές δημοπρασίες είναι ερωτήματα που απαντώνται στην αρχή του 2^{ου} Κεφαλαίου. Στη συνέχεια γίνεται αναφορά στο Foundation of Intelligent Physical Agents (FIPA) και υπογραμμίζεται ο ρόλος του στην ανάπτυξη προτύπων για συστήματα πρακτόρων. Έπειτα παρουσιάζεται το Αγγλικό Πρότυπο Δημοπρασίας, τόσο

διαγραμματικά όσο και περιγραφικά. Τέλος, παρουσιάζεται το μοντέλο Client – Server που αποτέλεσε το μοντέλο επικοινωνίας στην υλοποίηση της Αγγλικής Δημοπρασίας.

Στο 3^ο Κεφάλαιο καταγράφονται τα εργαλεία που χρησιμοποιήθηκαν στην ανάπτυξη λογισμικού στα πλαίσια της παρούσας διπλωματική εργασίας. Αρχικά παρουσιάζεται το πακέτο εργαλείων Uniform Server. Αναφέρονται τα χαρακτηριστικά του και καταγράφονται τα εργαλεία που ενσωματώνει. Ιδιαίτερη αναφορά γίνεται στο εργαλείο phpMyAdmin. Στη συνέχεια αναλύονται οι γλώσσες Html, Php, MySql και JavaScript. Δίνετε ο ορισμός για κάθε μία, καταγράφονται τα χαρακτηριστικά τους και οι λειτουργίες που μπορούν να επιτελέσουν. Επίσης αναφέρονται ορισμένα ιστορικά στοιχεία για τον τρόπο και το σκοπό της ανάπτυξής τους.

Στο 4^ο Κεφάλαιο παρουσιάζεται η ανάπτυξη λογισμικού της εφαρμογής Αγγλικής Δημοπρασίας. Αρχικά παρουσιάζεται η σχεδίαση της Βάσης Δεδομένων και η υλοποίηση της μέσω του εργαλείου phpMyAdmin. Παρουσιάζονται, η δημιουργία των κατάλληλων πινάκων και ο ρόλος τους στην εφαρμογή. Το απαραίτητο λογισμικό για την παρούσα διπλωματική εργασία αναπτύχθηκε σε δύο στάδιο. Στο πρώτο στάδιο (παράγραφος 4.2) αναπτύχθηκε η εφαρμογή της διαδικτυακής Αγγλικής Δημοπρασίας. Στην παράγραφο αυτή παρουσιάζονται αναλυτικά οι υλοποιήσεις των συντελεστών μιας δημοπρασίας (διοργανωτής δημοπρασίας, αγοραστής) με τη χρήση html, php και MySql, ώστε να ανταποκρίνονται στις απαιτήσεις του πρωτοκόλλου Αγγλικής Δημοπρασίας. Η υλοποίηση των κανόνων της δημοπρασίας όπως αυτές αναφέρονται στο πρωτόκολλο. Στο δεύτερο στάδιο (παράγραφος 4.3) αναπτύχθηκε ο μηχανισμός προσομοίωσης της Αγγλικής Δημοπρασίας. Η προσομοίωση επιτυγχάνεται με τη χρήση πρακτόρων ως αγοραστών και των λειτουργιών του 1^{ου} σταδίου. Καταγράφονται οι απαραίτητες αλλαγές σε επίπεδο πηγαίου κώδικα, ώστε να υποστηριχθεί η προσομοίωση και αναλύεται η υλοποίηση των πρακτόρων μαζί με τον απλό αλγόριθμο απόφασης.

Στο 5^ο Κεφάλαιο παρουσιάζονται τα αποτελέσματα της εργασίας. Καταγράφονται τυχόν προβλήματα που συναντήθηκαν κατά τη διάρκειά της ή τυχόν παραδοχές που χρειάστηκαν να γίνουν στην ανάπτυξη λογισμικού, οι οποίες και αναλύονται. Γενικότερα πραγματοποιείται μία κριτική επί του τελικού αποτελέσματος και παρουσιάζονται οι όποιες παραλείψεις εντοπίστηκαν. Τέλος, γίνονται προτάσεις και δίνονται ιδέες για μελλοντική έρευνα πάνω στο αντικείμενο των δημοπρασιών. Τόσο στο κομμάτι που αφορά τις online δημοπρασίες ως αξιόπιστες εφαρμογές, όσο και στο κομμάτι της εκπαίδευσης κατάλληλων πρακτόρων για συμμετοχή σε δημοπρασίες.

ΚΕΦΑΛΑΙΟ 2

Θεωρητικό Υπόβαθρο

Στο παρόν κεφάλαιο καταγράφεται το θεωρητικό υπόβαθρο που αποκτήθηκε για την εκπόνηση της παρούσας διπλωματικής εργασίας. Αρχικά, δίνεται ο ορισμός του όρου δημοπρασία, παρουσιάζονται οι κατηγορίες δημοπρασιών, αναλύονται οι παράμετροι μιας δημοπρασίας και γίνεται μια πιο εκτενής αναφορά στις ηλεκτρονικές δημοπρασίες (σύγκριση με τις κοινές δημοπρασίες). Στη συνέχεια, παρουσιάζεται το FIPA English Auction Protocol που αποτελεί και το κύριο αντικείμενο της παρούσας εργασίας. Γίνεται μια σύντομη αναφορά στον οργανισμό Foundation of Intelligent Physical Agents (FIPA) και στη συνέχεια παρουσιάζεται το Αγγλικό Πρότυπο Δημοπρασίας τόσο σχηματικά όσο και περιγραφικά. Τέλος, παρουσιάζεται το μοντέλο Client – Server που αποτέλεσε το μοντέλο επικοινωνίας στην υλοποίηση της Αγγλικής Δημοπρασίας.

2.1 Δημοπρασίες

2.1.1 Ορισμός της Δημοπρασίας – Ιστορική Αναδρομή

Με τον όρο δημοπρασία εννοούμε τη διαδικασία πώλησης ενός προϊόντος από έναν πωλητή (διοργανωτής δημοπρασίας), μέσω της υποβολής βελτιωμένων προτάσεων από τους επίδοξους αγοραστές. Μέσω αυτής της διαπραγματεύσης διοργανωτή – αγοραστών προκύπτει η τιμή πώλησης του προϊόντος.

Η ιστορία των δημοπρασιών ξεκινά ήδη από το 500 π.Χ. στη Βαβυλώνα. Εκεί, βάσει των ιστορικών κειμένων του Ηρόδοτου, συναντάται για πρώτη φορά μια μορφή δημοπρασίας που ωστόσο δεν αφορούσε την αγοραπωλησία προϊόντων, αλλά γυναικών. Κάθε χρόνο διατηρούσαν το έθιμο να δημοπρατούν τις υποψήφιες για γάμο γυναίκες, ξεκινώντας από τις πιο όμορφες (σύμφωνα πάντα με την άποψη του διοργανωτή) και καταλήγοντας στις λιγότερο όμορφες. ^[1]

Στη Ρωμαϊκή Αυτοκρατορία συναντώνται επίσης αναφορές για μορφές δημοπρασίας. Οι Ρωμαίοι στρατιώτες, μετά από κάθε νικηφόρα πολεμική μάχη, δημοπρατούσαν τα λάφυρα που είχαν συγκεντρωθεί. Σαν λάθυρο θεωρούνταν επίσης και οι αιχμάλωτοι, οι οποίοι και πωλούνταν ως δούλοι. Στη Ρωμαϊκή Αυτοκρατορία συνηθίζονταν επίσης η δημοπράτηση περιουσιακών στοιχείων ανθρώπων που χρωστούσαν και τους είχε κατασχεθεί η περιουσία (ώστε να αποπληρωθούν τα χρέη τους). Μια σημερινή μορφή δημοπρασίας τέτοιου είδους, αποτελεί η δημοπρασία κατασχεμένων περιουσιακών στοιχείων από τράπεζες, όταν οι οφειλέτες αδυνατούν να ανταπεξέλθουν στις υποχρεώσεις τους. ^[1]

Οι πρώτες δημοπρασίες στην μετά-Ρωμαϊκή εποχή εμφανίζονται στην Αγγλία του 17^{ου} – 18^{ου} αιώνα. Η εκκίνηση της δημοπρασίας δινόταν με το άναμμα ενός κεριού και διαρκούσε μέχρι να σβήσει (λιώσει) το κερί. Κατά τη διάρκειά της οι υποψήφιοι αγοραστές πραγματοποιούσαν προσφορές, με την υποχρέωση να βελτιώνουν κάθε φορά την τελευταία καλύτερη προσφορά. Όταν το κερί είχε λιώσει, ο αγοραστής με την καλύτερη προσφορά αγόραζε και το προϊόν. Η δημοπρασία αυτή είναι γνωστή ως “*auction by candle*” (“δημοπρασία κεριού”) και αποτελεί μία μορφή Αγγλικής Δημοπρασίας. ^[1]

Μετά το τέλος της Γαλλικής επανάστασης οι δημοπρασίες άρχισαν να λαμβάνουν χώρα σε εστιατόρια και σε καφετέριες της εποχής. Πλέον, τα προϊόντα προς πώληση ήταν κυρίως έργα τέχνης και σπάνια αντικείμενα. Καθημερινά, κατασκευάζονταν οι ανάλογοι κατάλογοι όπου καταγράφονταν τα αντικείμενα προς δημοπράτηση. ^[1]

2.1.2 Κατηγορίες και Παράμετροι Δημοπρασιών

Σήμερα, ο απλός ορισμός της δημοπρασίας που δόθηκε στην παράγραφο 2.1.1, δεν επαρκεί για να περιγράψει όλες τις μεθόδους δημοπρασίας που υπάρχουν. Έτσι παρακάτω ακολουθούν οι βασικές μέθοδοι δημοπρασίας που χρησιμοποιούνται στις μέρες μας, με την αντίστοιχη περιγραφή τους.

- **Αγγλική Δημοπρασία :** Η δημοπρασία ξεκινάει με την ανακοίνωση της τιμής εκκίνησης της δημοπρασίας από τον διοργανωτή. Η τιμή αυτή είναι πάντα χαμηλότερη από την πραγματική αξία του προϊόντος και ορίζεται από τον διοργανωτή. Ακολουθούν οι προσφορές των αγοραστών. Κάθε προσφορά, για να γίνει αποδεκτή, θα πρέπει να βελτιώνει την προηγούμενη καλύτερη προσφορά. Όταν κανένας αγοραστής δεν είναι διατεθειμένος να κάνει νέα προσφορά η δημοπρασία λήγει και το προϊόν πωλείται στον αγοραστή που πραγματοποίησε την τελευταία προσφορά (αν η προσφορά αυτή καλύπτει την επιδιωκόμενη τιμή πώλησης την οποία γνωρίζει μόνο ο διοργανωτής).^[2]
- **Ολλανδική Δημοπρασία :** Η δημοπρασία ξεκινάει με την ανακοίνωση της τιμής εκκίνησης από τον διοργανωτή. Η τιμή αυτή είναι σημαντικά μεγαλύτερη από την αναμενόμενη τιμή πώλησης του προϊόντος. Σταδιακά η τιμή εκκίνησης μειώνεται, έως ότου κάποιος αγοραστής εκδηλώσει ενδιαφέρον – πραγματοποιήσει προσφορά. Τότε η δημοπρασία λήγει και το προϊόν πωλείται στον αγοραστή στην τιμή που ανακοινώθηκε τελευταία. Συνήθως, ο διοργανωτής έχει και μία επιδιωκόμενη τιμή πώλησης (την οποία γνωρίζει μόνο ο ίδιος), κάτω από την οποία δεν είναι διατεθειμένος να πουλήσει το προϊόν.^[3]
- **Δημοπρασία ενός χτυπήματος :** Η δημοπρασία ξεκινάει με την ανακοίνωση από τον διοργανωτή της τιμής εκκίνησης, που καθορίζεται από τον ίδιο και είναι χαμηλότερη από την αναμενόμενη τιμή πώλησης. Οι αγοραστές έχουν δικαίωμα να πραγματοποιήσουν μία μοναδική προσφορά κατά τη διάρκεια της δημοπρασίας, η οποία ουσιαστικά εκφράζει την τιμή που είναι διατεθειμένοι να πληρώσουν για το προϊόν. Οι διάφορες προσφορές δεν γίνονται γνωστές σε κανέναν αγοραστή παρά μόνο στον διοργανωτή. Στο τέλος της δημοπρασίας ο διοργανωτής ανακοινώνει τον νικητή ο οποίος έχει πραγματοποιήσει την καλύτερη προσφορά. Αξίζει να σημειωθεί πως αυτού του είδους η δημοπρασία χρησιμοποιείται από το Ελληνικό Δημόσια για την δημοπράτηση δημοσίων έργων.^[4]
- **Δημοπρασία Vickrey :** Και σε αυτή την περίπτωση η δημοπρασία ξεκινάει με την ανακοίνωση της τιμής εκκίνησης. Στη συνέχεια η μέθοδος διαπραγμάτευσης που ακολουθείτε είναι όμοια με αυτή της δημοπρασίας ενός χτυπήματος. Δηλαδή, οι αγοραστές έχουν δικαίωμα να πραγματοποιήσουν μία μοναδική προσφορά (συνήθίζεται

οι προσφορές να γίνονται γραπτώς ώστε να μένουν κρυφές) και οι προσφορές είναι γνωστές μόνο στον διοργανωτή. Η διαφορά έγκειται στο γεγονός πως ο νικητής της δημοπρασίας (αυτός που έχει κάνει την καλύτερη προσφορά) καλείται να πληρώσει την δεύτερη καλύτερη προσφορά ώστε να αποκτήσει το προϊόν. Αυτός είναι ο λόγος που ονομάζεται και δημοπρασία σφραγισμένης προσφοράς, δεύτερης τιμής.^[5]

- **Ανοιχτές δημοπρασίες:** Οι ανοιχτές δημοπρασίες ακολουθούν τη μέθοδο διαπραγμάτευσης Αγγλικής Δημοπρασίας. Δηλαδή υπάρχει η τιμή εκκίνησης και οι αγοραστές πραγματοποιούν τις προσφορές τους βελτιώνοντας την τελευταία προσφορά. Η διαφορά έγκειται στο γεγονός πως ο διοργανωτής δεν δεσμεύεται από την επιδιωκόμενη τιμή. Στην περίπτωση των ανοιχτών δημοπρασιών δεν υπάρχει η έννοια της επιδιωκόμενης τιμής και η πώληση ή μη του προϊόντος μετά το τέλος της δημοπρασίας βρίσκεται στην ευχέρεια του εκάστοτε διοργανωτή – κατόχου του προϊόντος.^[6]

Παραπάνω, παρουσιάστηκαν οι βασικότερες και πιο διαδεδομένες κατηγορίες δημοπρασιών. Από τις περιγραφές τους, διαπιστώνεται πως υπάρχουν κάποιοι παράγοντες (θα μπορούσαν να ονομαστούν και μεταβλητές δημοπρασίας) που συναντώνται σχεδόν σε όλα τα είδη των δημοπρασιών. Ουσιαστικά, υπάρχει ένα είδος ορολογίας που χρησιμοποιείται στον χώρο των δημοπρασιών. Οι βασικότερες παράμετροι, με τους αντίστοιχους ορισμούς είναι :

- **Τιμή Εκκίνησης (Opening Price) :** Η τιμή με τη οποία ξεκινάει η δημοπρασία. Αποφασίζεται και ανακοινώνεται από το διοργανωτή της δημοπρασίας και σηματοδοτεί την έναρξη της διαδικασίας διαπραγμάτευσης. Ανάλογα με την κατηγορία της δημοπρασίας η τιμή εκκίνησης είναι μικρότερη (π.χ. Αγγλική Δημοπρασία) ή μεγαλύτερη (π.χ. Ολλανδική Δημοπρασία) της αναμενόμενης τιμής πώλησης του προϊόντος.
- **Επιδιωκόμενη Τιμή (Reservation Price) :** Η τιμή κάτω από την οποία δεν πωλείται το προϊόν. Αποφασίζεται από τον διοργανωτή και δεν γίνεται γνωστή στους αγοραστές. Χρησιμοποιείται μετά τη λήξη της δημοπρασίας για να ελεγχτεί αν η καλύτερη προσφορά καλύπτει την συγκεκριμένη τιμή και να αποφασιστεί αν θα υπάρξει νικητής ή όχι. Η επιδιωκόμενη τιμή είναι προαιρετική και δεν υπάρχει πάντα στις δημοπρασίες.
- **Βήμα Δημοπρασίας (Bid Increment) :** Το ελάχιστο ποσό βελτίωσης της τελευταίας καλύτερης προσφοράς, ώστε η νέα προσφορά να γίνει αποδεκτή. Καθορίζεται από τον διοργανωτή και γνωστοποιείται στους αγοραστές πριν την έναρξη της δημοπρασίας μαζί με τους λοιπούς κανόνες που τυχόν υπάρχουν στη διαδικασία.
- **Διάρκεια δημοπρασίας-παράταση :** Η διάρκεια της δημοπρασίας καθορίζεται από το διοργανωτή. Κοινή πρακτική αποτελεί ο ορισμός της χρονικής στιγμής έναρξης και

λήξης της δημοπρασίας και η ανακοίνωσή τους στους αγοραστές πριν την έναρξη της δημοπρασίας. Σε πολλές δημοπρασίες, υπάρχει η δυνατότητα παράταση της διαδικασίας. Η παράταση έχει ως σκοπό να αποσοβήσει πρακτικές αγοραστών κατά τις πραγματοποιούνται προσφορές λίγο πριν τη λήξη της κανονικής διάρκειας της δημοπρασίας, ώστε να μην αφήσουν περιθώρια αντίδρασης στους ανταγωνιστές. Με την παράταση οι αγοραστές έχουν το χρόνο να αντιδράσουν στις προσφορές της τελευταίας στιγμής.

- **Τιμή κλεισίματος (Closing Price)** : Αποτελεί την τιμή πώλησης του προϊόντος. Στην ουσία είναι η τιμή της τελευταίας καλύτερης προσφοράς, που καλύπτει την επιδιωκόμενη τιμή του διοργανωτή.

2.1.3 Ηλεκτρονικές Δημοπρασίες

Ο τρόπος με τον οποίο γίνονται αγοραπωλησίες στις μέρες μας έχει αλλάξει σε σχέση με το παρελθόν. Με την ανάπτυξη του διαδικτύου έχουν αναπτυχθεί νέοι τρόποι για να πραγματοποιούν οι άνθρωποι της αγοράς τους, με online καταστήματα στα οποία μπορεί κάποιος να βρει όλα τα προϊόντα που κυκλοφορούν στα κοινά καταστήματα. Παράλληλα με την ανάπτυξη των διαδικτυακών καταστημάτων, αναπτύχθηκαν και οι διαδικτυακές ή ηλεκτρονικές δημοπρασίες.^[71]

Κατά κανόνα οι ηλεκτρονικές δημοπρασίες ακολουθούν κάποιο πρότυπο των κλασικών δημοπρασιών. Οι κανόνες που ορίζει κάθε μέθοδος ισχύουν με τον ίδιο τρόπο. Τα στοιχεία που μεταβάλλονται αφορούν κυρίως τον τρόπο που πραγματοποιούνται οι προσφορές από τους αγοραστές (μέσω email ή κάποιας φόρμας στοιχείων), τον τρόπο επικοινωνίας διοργανωτή – αγοραστών (πραγματοποιείται συνήθως μέσω της ιστοσελίδας δημοπρασίας) και τον τρόπο απόκτησης του προϊόντος (ταχυδρομική αποστολή).^[71]

Οι ηλεκτρονικές δημοπρασίες σε ορισμένα σημεία πλεονεκτούν και σε άλλα μειονεκτούν, σε σχέση με τις κλασικές δημοπρασίες. Όσον αφορά τα πλεονεκτήματα, στις ηλεκτρονικές δημοπρασίες:

- ◆ Οι αγοραστές δεν είναι υποχρεωμένοι να βρίσκονται σε κάποιο συγκεκριμένο χώρο για να συμμετέχουν στη δημοπρασία. Μπορούν από οποιοδήποτε μέσο σύνδεσης στο διαδίκτυο διαθέτουν να πραγματοποιήσουν την προσφορά τους. Με αυτό τον τρόπο οι αγοραστές είναι σε θέση να συμμετέχουν σε παραπάνω από μία δημοπρασίες ταυτόχρονα. Για συγκεκριμένο προϊόν μάλιστα μπορούν να αναζητήσουν τη δημοπρασία που τους συμφέρει περισσότερο. Επίσης, μπορούν εύκολα να βρίσκουν δημοπρασίες με προϊόντα που τους ενδιαφέρουν, μέσω αναζήτησης στο διαδίκτυο.^[71]

- ◆ Οι πωλητές – διοργανωτές των δημοπρασιών μπορούν εύκολα να δημοπρατούν ταυτόχρονα παραπάνω από ένα προϊόν καθώς η δημοπράτηση ενός προϊόντος δεν τους δεσμεύει. Ένα επιπλέον όφελος για τον πωλητή είναι ο αυξημένος αριθμός υποψηφίων αγοραστών που μπορούν να συμμετέχουν στη δημοπρασία. Μπορούν να συμμετέχουν άνθρωποι από όλο τον κόσμο χωρίς να χρειάζεται να ταξιδέψουν σε κάποιο συγκεκριμένο τόπο στον οποίο μπορεί να λαμβάνει χώρα μια δημοπρασία. Αυτό συνεπάγεται και μια πιθανή καλύτερη τιμή πώλησης για τον προϊόν, καθώς αύξηση του αριθμού των αγοραστών σημαίνει ταυτόχρονα και αύξηση των προσφορών.^[7]

Όσον αφορά τα μειονεκτήματα, αυτά αφορούν κυρίως τους αγοραστές και είναι :

- ◆ Οι αγοραστές δεν έχουν τη δυνατότητα να εξετάσουν το αντικείμενο προς δημοπράτηση από κοντά. Η περιγραφή του σε μια ηλεκτρονική δημοπρασία μπορεί να είναι ελλιπής και να μην καλύπτει την πλευρά του αγοραστή. Εκτός από αυτό, η περιγραφή μπορεί να μην ανταποκρίνεται στην πραγματικότητα. Εύκολα κάποιος πωλητής μπορεί να παραπλανήσει τους αγοραστές μέσω μιας ψεύτικης και ωραιοποιημένης περιγραφής του προϊόντος του.^[7]
- ◆ Τα έξοδα αποστολής του προϊόντος στον τελικό αγοραστή του, που επιβαρύνουν είτε τον διοργανωτή είτε τον αγοραστή (ανάλογα με τη συμφωνία), αποτελούν επιπλέον κόστος που στην περίπτωση των κοινών δημοπρασιών δεν υπάρχει.

2.2 FIPA English Auction

Μια βασική απαίτηση για τα συστήματα πολλαπλών πρακτόρων είναι η δυνατότητα διαλειτουργικότητας μεταξύ διαφορετικών και ετερογενών συστημάτων. Για το σκοπό αυτό έχουν προταθεί κατά καιρούς κάποια πρότυπα, τα σημαντικότερα από τα οποία έχουν συσταθεί από την οργάνωση FIPA (Foundation for Intelligent Physical Agents).

2.2.1 The Foundation of Intelligent Physical Agents (FIPA)



Η FIPA είναι μια διεθνής μη - κερδοσκοπικού χαρακτήρα ένωση εταιριών και οργανισμών, που έχει ως στόχο την παραγωγή προδιαγραφών για γενικευμένες και ενοποιημένες agent τεχνολογίες.^{[8] [9]}

Η FIPA διαμορφώθηκε αρχικά ως μια οργάνωση με έδρα την Ελβετία το 1996, με σκοπό να παράγει πρότυπα προδιαγραφών λογισμικού για ετερογενείς και πράκτορες που αλληλεπιδρούν, καθώς και συστήματα πρακτόρων. Από την ίδρυση της έχει διαδραματίσει καίριο ρόλο στην ανάπτυξη προτύπων πρακτόρων και έχει προωθήσει μια σειρά πρωτοβουλιών και γεγονότων που συνέβαλαν στην ανάπτυξη και την αφομοίωση της τεχνολογίας του πράκτορα. Τον Μάρτιο του 2005, το Διοικητικό Συμβούλιο της FIPA πρότεινε στα μέλη της οργάνωσης την ένταξη του οργανισμού στην κοινότητα του IEEE. Η πρόταση έγινε δεκτή ομόφωνα. Το επόμενο βήμα ήταν να αναπτυχθούν πρότυπα για τον ευρύτερο χώρο της ανάπτυξης λογισμικού και όχι μόνο για πράκτορες και συστήματα πρακτόρων. Δηλαδή, η τεχνολογία πρακτόρων να ενσωματωθεί και να δουλεύει και σε μη-πρακτορικά συστήματα.^{[8] [9]}

Οι προδιαγραφές της FIPA περιγράφουν πρωτόκολλα, τα οποία στοχεύουν στην ομαλή συνδεσιμότητα μεταξύ πρακτόρων οποιασδήποτε προέλευσης και υλοποίησης, καθώς και μεταξύ πρακτόρων και οντοτήτων που ανήκουν σε άλλα συστήματα. Οι προδιαγραφές αυτές μπορούν να διαχωριστούν σε τέσσερις γενικές κατηγορίες :

1. Επικοινωνία Πρακτόρων
2. Μεταφορά Πρακτόρων
3. Διαχείριση Πρακτόρων
4. Αρχιτεκτονική Συστήματος

Από τις παραπάνω κατηγορίες σημαντικότερα είναι τα πρωτόκολλα που σχετίζονται με την επικοινωνία των πρακτόρων στο μοντέλο FIPA.

Στην εργασία αυτή θα στηριχτούμε στο πρωτόκολλο που έχει αναπτύξει η FIPA για την υλοποίηση της Αγγλικής Δημοπρασίας, το οποίο και θα εξηγήσουμε αναλυτικά παρακάτω.

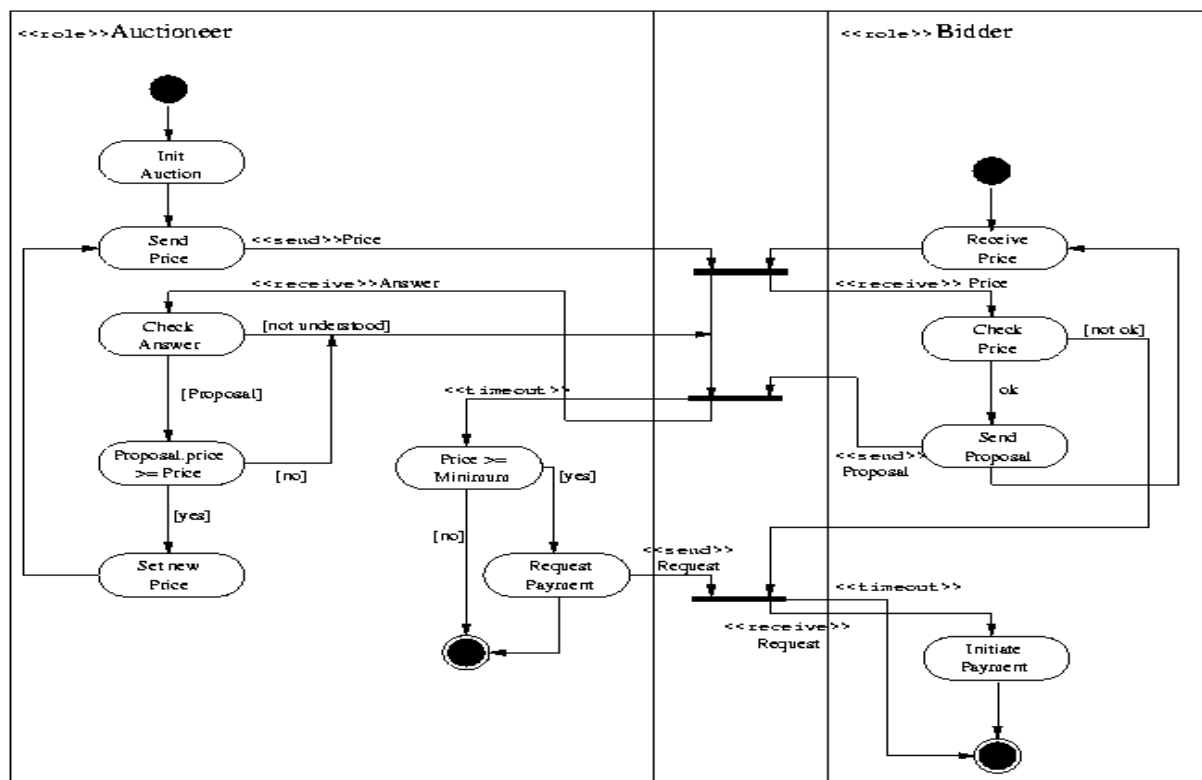
2.2.2 Το Πρωτόκολλο της Αγγλικής Δημοπρασίας (English Auction)

2.2.2.1 Αναπαράσταση του Πρωτοκόλλου μέσω UML – AUML

Η UML (Unified Modeling Language) δημιουργήθηκε ως μία γλώσσα για τη μοντελοποίηση εφαρμογών λογισμικού, η οποία προήλθε από τον συνδυασμό διαφόρων τέτοιων γλωσσών που χρησιμοποιούνταν για τον ίδιο σκοπό (πάνω από πέντε τέτοιες μέθοδοι στις αρχές του 1990). Χρησιμοποιείται κυρίως στην ανάπτυξη και σχεδίαση λογισμικού με σκοπό την οπτικοποίηση (visualization) τμημάτων λογισμικού. Αποτελεί ουσιαστικά μια γλώσσα γραφικής αναπαράστασης με κύριο πλεονέκτημα της την μη εξάρτηση της από συγκεκριμένες πλατφόρμες και γλώσσες προγραμματισμού. [\[12\]](#)

Παρόλο που η UML δημιουργήθηκε για μοντελοποίηση εφαρμογών λογισμικού, χρησιμοποιείται γενικότερα στην αναπαράσταση συστημάτων όπως για παράδειγμα σε εταιρικά και επιχειρησιακά συστήματα. [\[12\]](#)

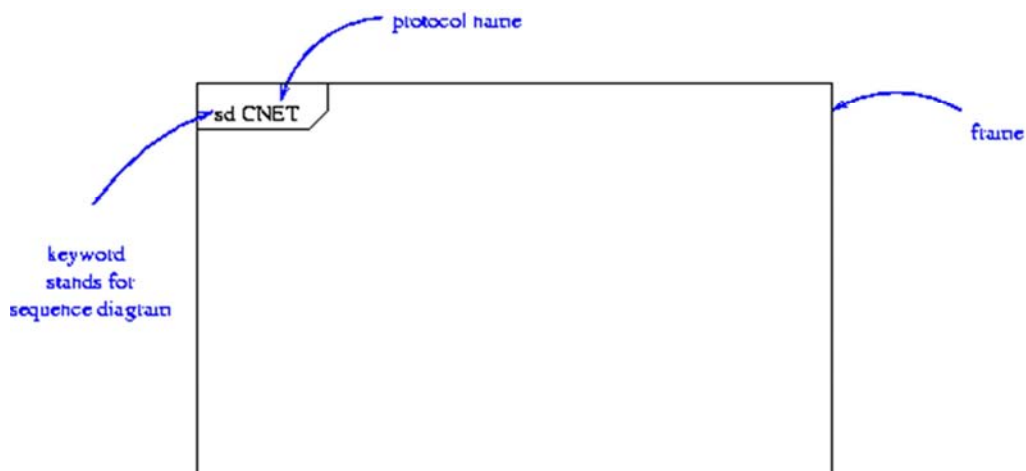
Η αναπαράσταση της Αγγλικής Δημοπρασίας με διάγραμμα UML φαίνεται παρακάτω:



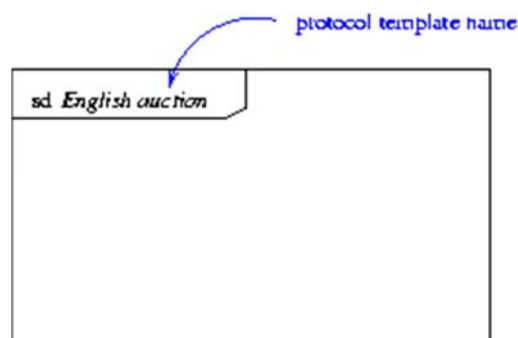
Εικόνα 2.1 (Πηγή [\[10\]](#)) : Αναπαράσταση της Αγγλικής Δημοπρασίας με Διάγραμμα UML

Παρόλο που η UML χρησιμοποιείται στη σχεδίαση διάφορων συστημάτων (όχι μόνο λογισμικού), η FIPA έχει ανάπτυξη την δική της γλώσσα για την μοντελοποίηση των πρακτόρων η οποία ονομάζεται AUML. Έτσι λοιπόν, η FIPA AUML αποτελεί μία ενοποιημένη πρότυπη γλώσσα μοντελοποίησης για συστήματα πρακτόρων.

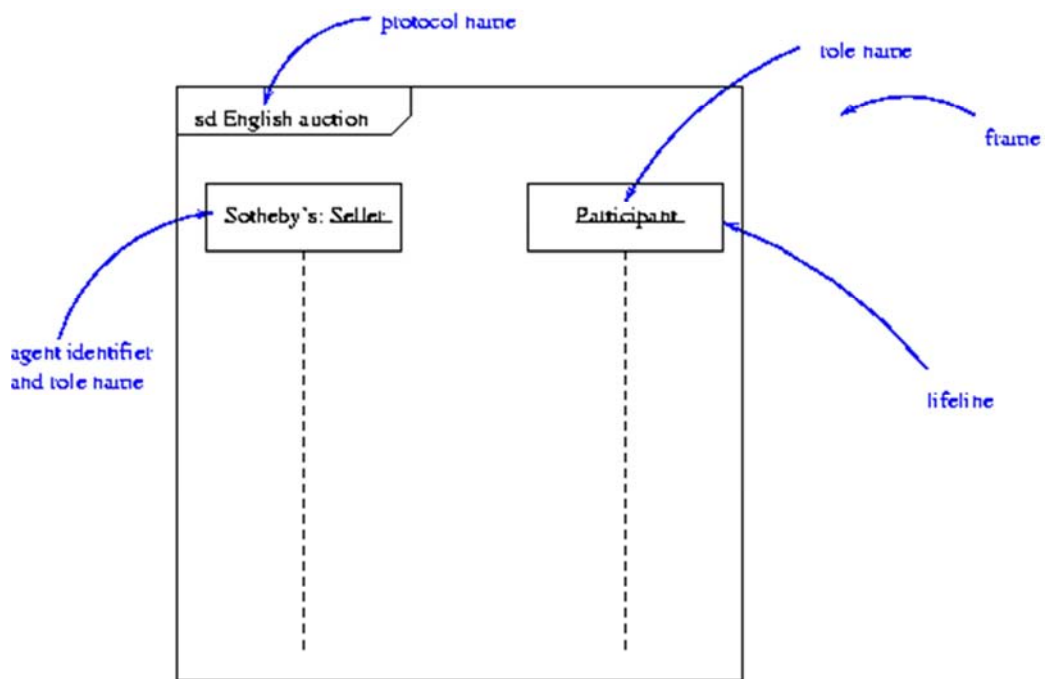
Η δομή του πρωτοκόλλου της FIPA Αγγλικής Δημοπρασίας με τη χρήση AUML φαίνεται στα παρακάτω σχήματα :



Εικόνα 2.2 (Πηγή [11]) : Agent AUML Frame Notation

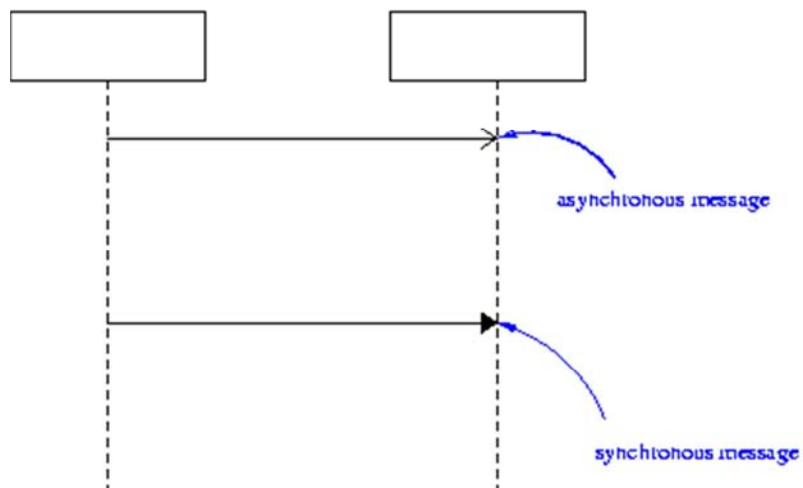


Εικόνα 2.3 (Πηγή [11]) : Agent AUML Frame Notation with Parameters

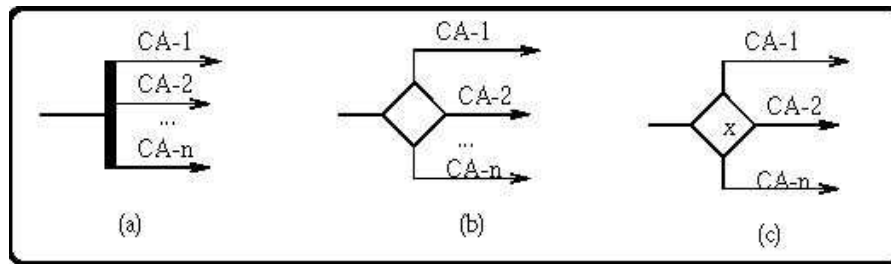


Εικόνα 2.4 (Πηγή [11]): Agent AUML Lifeline Notation

Η FIPA έχει υιοθετήσει κανόνες και τρόπους ανταλλαγής μηνυμάτων μεταξύ των πρακτόρων όπως φαίνεται στα επόμενα σχήματα :

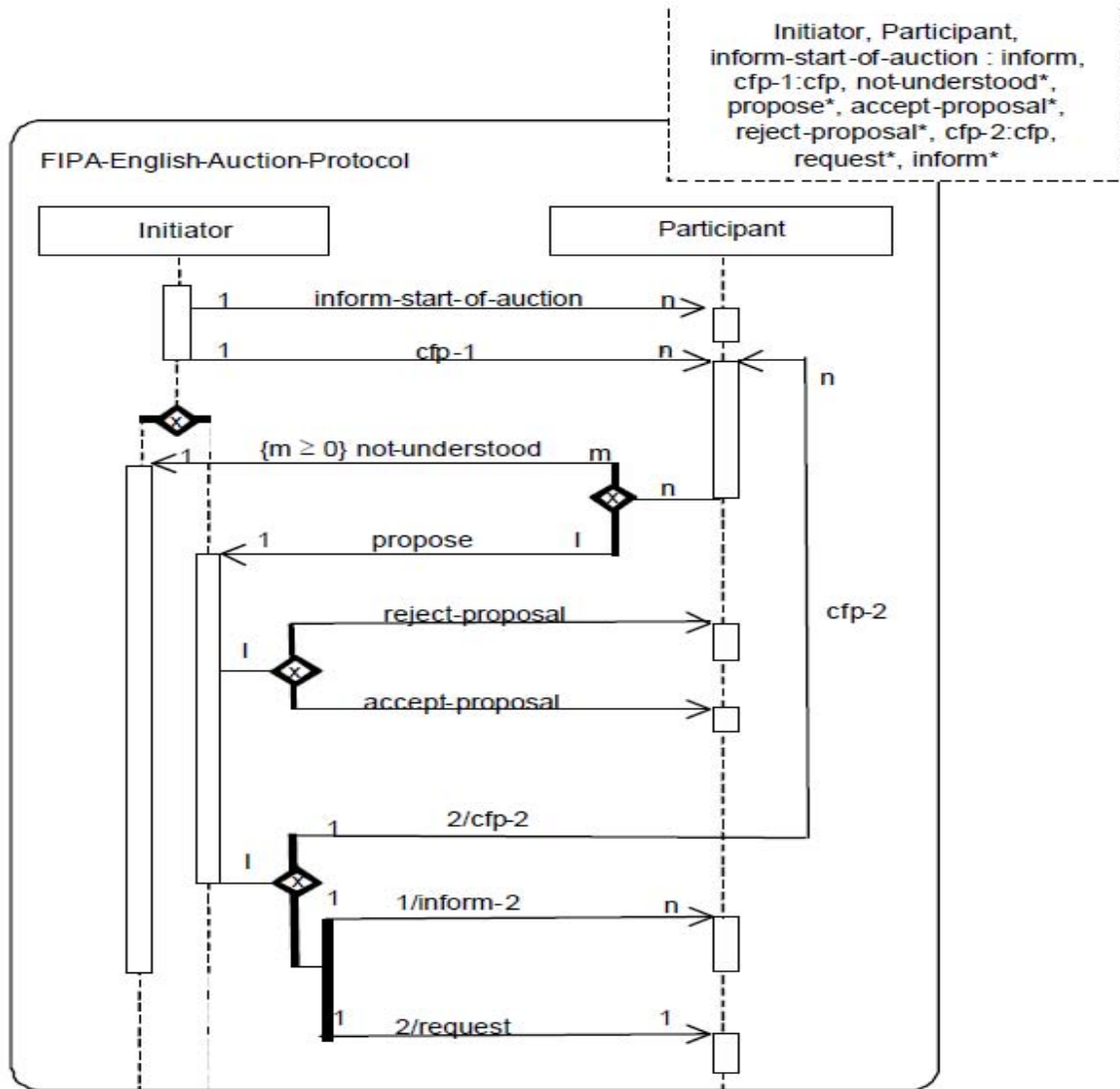


Εικόνα 2.5 (Πηγή [11]): Agent AUML Message Notation



Εικόνα 2.6 (Πηγή [11]): Agent UML Connectors: (a) AND (b) OR (c) XOR

Το πρωτόκολλο της αγγλικής δημοπρασίας που προκύπτει σύμφωνα με την FIPA AUML φαίνεται στην παρακάτω εικόνα :



Εικόνα 2.6 (Πηγή [2]): FIPA English Auction Interaction Protocol

2.2.2.2 Περιγραφή του Πρωτοκόλλου

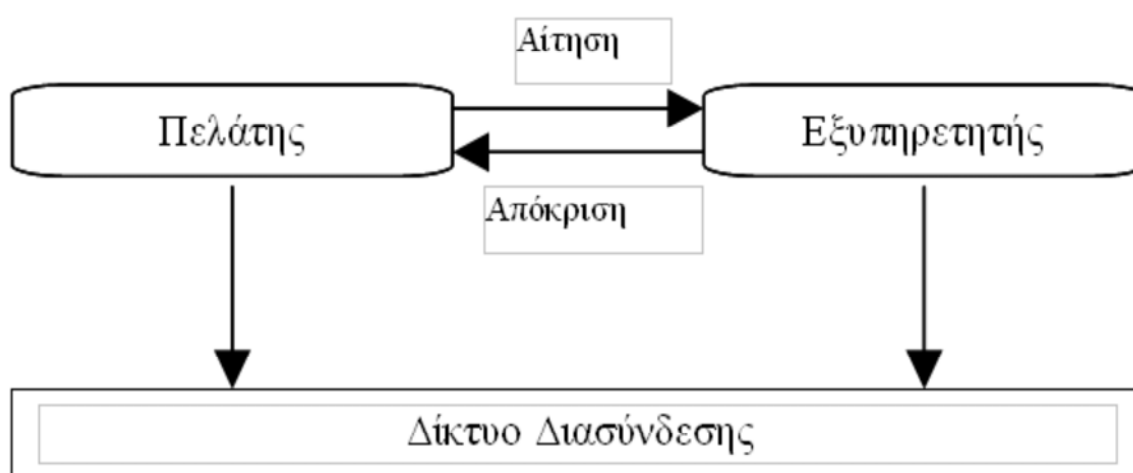
Στην Αγγλική Δημοπρασία ο διοργανωτής της δημοπρασίας ψάχνει να βρει την τιμή πώλησης ενός αγαθού, προτείνοντας αρχικά μια τιμή χαμηλότερη από την υπολογιζόμενη τιμή πώλησης, η οποία βαθμιαία αυξάνεται. Κάθε φορά που η τιμή έχει ανακοινωθεί, ο διοργανωτής της δημοπρασίας περιμένει να δει αν υπάρχουν αγοραστές που είναι διατεθειμένοι να προτείνουν βελτιωμένη τιμή. Μόλις ένας αγοραστής δηλώσει τη βελτιωμένη τιμή και γίνει αποδεκτή, ο διοργανωτής της δημοπρασίας εκδίδει νέα πρόσκληση για την υποβολή προσφορών ως προς τη νέα τιμή. Η δημοπρασία συνεχίζεται έως ότου κανένας αγοραστής δεν είναι διατεθειμένος να καταβάλει βελτιωμένη πρόταση, στο οποίο σημείο τελειώνει η δημοπρασία. Εάν η τελευταία τιμή που προτάθηκε από αγοραστή υπερβαίνει την επιδιωκόμενη τιμή (reservation price) του διοργανωτή της δημοπρασίας (γνωστή μόνο στον διοργανωτή), το αγαθό πωλείται στον αγοραστή στην συμφωνημένη τιμή. Εάν η τελευταία τιμή που προτάθηκε από αγοραστή είναι μικρότερη από την reservation price του διοργανωτή, το αγαθό δεν πωλείται.^[2]

Από την παραπάνω περιγραφή του Αγγλικού προτύπου Δημοπρασίας και από τα υπόλοιπα χαρακτηριστικά του πρωτοκόλλου προκύπτουν τα εξής στοιχεία που χαρακτηρίζουν το πρότυπο :

- Ο διοργανωτής καθορίζει την τιμή εκκίνησης, το βήμα της δημοπρασίας και τον χρόνο λήξης.
- Η τιμή προοδευτικά βελτιώνεται από τους αγοραστές αυξάνεται μέχρι τη λήξη της δημοπρασίας. Οι προσφορές ανταγωνίζονται μεταξύ τους και γίνονται δεκτές μόνο αν βελτιώνουν την καλύτερη προσφορά κατά τουλάχιστον το βήμα της δημοπρασίας.
- Οι αγοραστές βλέπουν κάθε στιγμή την υπερισχύουσα τιμή και μπορούν να τη βελτιώσουν.
- Η δημοπρασία λήγει σε προκαθορισμένο χρονικό διάστημα. Υπάρχει η δυνατότητα αυτόματης παράτασης, η οποία ενεργοποιείται αν κατά τη διάρκεια των τελευταίων λεπτών υποβληθεί νέα προσφορά. Η διαθεσιμότητα ή μη αυτής της λειτουργίας γνωστοποιείται στους αγοραστές πριν την έναρξη της δημοπρασίας.
- Μετά τη λήξη της δημοπρασίας, οι προσφορές κατατάσσονται με βάση την τιμή για να προκύψει ο νικητής.

2.3 Το Μοντέλο Client - Server

Το μοντέλο client – server αποτελεί ένα από τα κυρίαρχα μοντέλα επικοινωνίας μεταξύ τμημάτων λογισμικού. Για να είναι αποτελεσματική η επικοινωνία, ο αποστολέας και ο παραλήπτης πρέπει να συμφωνούν στη σημασία των πληροφοριών που ανταλλάσσονται μεταξύ τους και αυτό γίνεται σε διάφορα επίπεδα.^[13]



Εικόνα 2.7: Μοντέλο πελάτη-εξυπηρετητή

Στο μοντέλο αυτό, το σύστημα δομείται ως ένα σύνολο διεργασιών (server) οι οποίες προσφέρουν υπηρεσίες στις διεργασίες του χρήστη (client). Οι διεργασίες και οι υπηρεσίες εκτελούνται σε διαφορετικές μηχανές πάνω από το ίδιο ή διαφορετικά λειτουργικά συστήματα, αρκεί να υπάρχει συμφωνία μεταξύ τους στην αναπαράσταση και την ερμηνεία των μηνυμάτων που ανταλλάσσονται. Οι υπηρεσίες που προσφέρονται στις διεργασίες του χρήστη χρησιμοποιούν ένα απλό πρωτόκολλο αίτησης/απάντησης (request/reply protocol) και σε αυτές περιλαμβάνονται και υπηρεσίες βάσεως δεδομένων.^[13]

Μία τέτοια αρχιτεκτονική είναι ιδιαίτερος αξιόπιστη όταν εφαρμόζεται σε ένα ενσύρματο τοπικό δίκτυο καθώς η επικοινωνία πελάτη-εξυπηρετητή υλοποιείται χωρίς εγκατάσταση συνδέσεων. Το πλεονέκτημα είναι η απλότητα της επικοινωνίας : ο πελάτης στέλνει μία αίτηση, και παίρνει πίσω μία απάντηση, χωρίς να χρειάζεται να εγκατασταθεί ούτε να τερματιστεί μία σύνδεση. Το μήνυμα απάντησης μπορεί να χρησιμοποιηθεί ως υπονοούμενο μήνυμα παραδοχής (acknowledgement message) της αίτησης.

Η απλότητα του μοντέλου αυτού σημαίνει μικρή επιβάρυνση στο σύστημα, αφού απαιτούνται λιγότερα επίπεδα πρωτοκόλλων.

ΚΕΦΑΛΑΙΟ 3

Χρήσιμα Εργαλεία

Σε αυτό το κεφάλαιο καταγράφονται τα εργαλεία που χρησιμοποιήθηκαν στην ανάπτυξη λογισμικού στα πλαίσια της παρούσας διπλωματική εργασίας. Αρχικά παρουσιάζεται (παράγραφος 3.1) το πακέτο εργαλείων Uniform Server. Αναφέρονται τα χαρακτηριστικά του και καταγράφονται τα εργαλεία που ενσωματώνει. Ιδιαίτερη αναφορά γίνεται στο εργαλείο phpMyAdmin. Στη συνέχεια αναλύονται οι γλώσσες Html, Php, MySql και JavaScript. Δίνετε ο ορισμός για κάθε μία, καταγράφονται τα χαρακτηριστικά τους και οι λειτουργίες που μπορούν να επιτελέσουν. Επίσης αναφέρονται ορισμένα ιστορικά στοιχεία για τον τρόπο και το σκοπό της ανάπτυξής τους.

3.1 Uniform Server

Ο Uniform Server αποτελεί “ένα ανοιχτού κώδικα πακέτο εργαλείων WAMP (ακρωνύμιο των *Windows Apache MySQL Php ή Perl ή Python*), αποτελώντας μία προ-διαμορφωμένη εγκατάσταση ανοιχτού κώδικα εργαλείων για *web server*, τον *Apache HTTP server*, την *MySQL database* και γλώσσες όπως *Php* και *Perl*. Κυκλοφορεί κάτω από την προστασία της *BSD* (πακέτο αδειών χρήσης για ανοιχτού κώδικα λογισμικό) άδειας χρήσης”.^[14]

Μπορεί να ενεργοποιηθεί χωρίς καμία εγκατάσταση, είναι ανεξάρτητο πρόγραμμα (περιέχει όλα όσα χρειάζεται) και έχει μέγεθος κάτω από 10 MB. Για αυτούς τους λόγους, συνηθίζεται να χρησιμοποιείται για την δοκιμή web εφαρμογών σε περιβάλλον Microsoft Windows (είναι συμβατό με όλες τις εκδόσεις Windows), αλλά μπορεί να χρησιμοποιηθεί και σαν φορητός web server αν στηθεί σε ένα φορητό δίσκο. Επίσης, επειδή είναι σχεδιασμένος και για ασφάλεια, ο Uniform Server μπορεί να χρησιμοποιηθεί και για πραγματικά websites.^[14]

Στην παρούσα εργασία χρησιμοποιήθηκε ο Uniform Server 5.6.5-Nano που περιέχει τις εξής εκδόσεις εργαλείων :

- Apache 2.2.15
- MySQL 5.1.46-community
- PHP 5.3.2
- phpMyAdmin 3.3.3
- ActivePerl 5,10,0,1006
- eAccelerator 1.0-dev
- SSL pre-installed (ssl_0981)
- UTF-8 Compatible

Ένα πολύ χρήσιμο εργαλείο που περιλαμβάνεται στον Uniform Server είναι το phpMyAdmin. Το phpMyAdmin είναι “ένα ανοιχτού κώδικα εργαλείο, υλοποιημένο σε *php*, με σκοπό τη διαχείριση της *MySQL* στο *World Wide Web*”. Μπορεί να πραγματοποιήσει διάφορες λειτουργίες όπως είναι η δημιουργία, η μορφοποίηση ή η διαγραφή μιας Βάσης Δεδομένων, ενός πίνακα, μιας σειράς ή μιας στήλης, η εκτέλεση SQL ερωτήσεων ή η διαχείριση των χρηστών – ανάθεση δικαιωμάτων χρήσης.^[15]

Το interface του εργαλείου phpMyAdmin φαίνεται στην επόμενη εικόνα:



Εικόνα 3.1 : Το Γραφικό περιβάλλον του phpMyAdmin στον Uniform Server

3.2 Html, Php, MySql και JavaScript

Για την ανάπτυξη της διαδικτυακής εφαρμογής του Αγγλικού πρωτοκόλλου Δημοπρασίας χρησιμοποιήθηκαν και συνδυάστηκαν στοιχεία από html, php, MySQL και JavaScript (όπως περιγράφεται αναλυτικά στο κεφάλαιο 4). Τι ακριβώς είναι καθένα από αυτά και τι λειτουργίες επιτελούν αναφέρονται παρακάτω.

Τα αρχικά **HTML** προέρχονται από τις λέξεις **HyperText Markup Language**. Η html δεν είναι μια γλώσσα προγραμματισμού αλλά μια γλώσσα σήμανσης. Ουσιαστικά είναι ένας τρόπος για να γράφεται και να μορφοποιείται (μέσω των tags) ένα κείμενο. Αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM.^[16]

Ο Browser είναι αποτελεί μία εφαρμογή που μπορεί να “διαβάζει” αυτή τη γλώσσα σήμανσης και το αντίστοιχο περιεχόμενο να το εμφανίζει στην οθόνη του χρήστη. Το κύριο

συστατικό της html είναι τα tags. Κάθε λειτουργία που καλείται να εκτελέσει ο browser περικλείεται σε ένα tag. ^[16]

Η εκτέλεση αυτών των html εντολών μπορεί να διαφέρει σε διαφορετικούς browsers. Για αυτό το λόγο έχουν θεσπιστεί κάποιοι κανόνες – προδιαγραφές που πρέπει να πληρούνται από τους browsers ώστε να αποφεύγονται πολλαπλές ερμηνείες της html. Ωστόσο ακόμα και σήμερα πολλά από τα html tags αναγνωρίζονται διαφορετικά από διαφορετικούς browser. Πολλοί web developers αποφεύγουν να χρησιμοποιούν deprecated html tags και προτιμούν να πετύχουν την επιθυμητή λειτουργία με άλλες τεχνικές, ώστε το αποτέλεσμα να είναι ανεξάρτητο του browser. ^[16]

Η **PHP** αποτελεί μία γλώσσα προγραμματισμού που χρησιμοποιείται στην ανάπτυξη δυναμικών ιστοσελίδων. Είναι μία server side γλώσσα. Δηλαδή εκτελείται στη μεριά του server και το αποτέλεσμα αποστέλλεται στον client σε html μορφή. Για να τρέξει, απαιτείται κατάλληλος server (π.χ. Apache) ο οποίος να είναι συμβατός με τη γλώσσα PHP και να έχει της κατάλληλες ρυθμίσεις. ^[17]

Βασικό πλεονέκτημα της PHP είναι πως μπορεί να επικοινωνεί, να συνδέεται και να διαχειρίζεται τη Βάση Δεδομένων στη μεριά του server. Αυτή η συνδεσιμότητα επιτυγχάνεται μέσω κοινών SQL ερωτήσεων, μέσα στο πηγαίο κώδικα PHP. ^[17]

Η πρώτη μορφή της PHP (1995 – φοιτητής Rasmus Lerdorf) ήταν γραμμένη σε γλώσσα Perl και ξεκίνησε από ένα απλό script. Ονομάστηκε αρχικά PHP/FI, από τα αρχικά Personal Home Page/Form Interpreter. Οι επόμενες εκδόσεις της PHP βασίζονται στη γλώσσα προγραμματισμού C. Σήμερα, έχει κυκλοφορήσει μέχρι και η PHP 6 (σε δοκιμαστικό στάδιο), με τις εκδόσεις PHP 4 και 5 ωστόσο να χρησιμοποιούνται ακόμη στα περισσότερα sites. ^[17]

Η **MySQL** είναι ένα πολύ γρήγορο και δυνατό, σύστημα *διαχείρισης βάσεων δεδομένων*. Μια βάση δεδομένων επιτρέπει την αποθήκευση, την αναζήτηση, την ταξινόμηση και την ανάκληση δεδομένων αποτελεσματικά . Ο MySQL server ελέγχει την πρόσβαση στα δεδομένα, για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς, η MySQL είναι ένας πολυνηματικός server πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια . Η MySQL είναι διαθέσιμη από το 1996 αλλά η ιστορία της ξεκινά από το 1979. ^[18]

Η **JavaScript** είναι γλώσσα προγραμματισμού που χρησιμοποιείται στην ανάπτυξη δυναμικών ιστοσελίδων, όπως και η PHP. Η διαφορά τους είναι πως η JavaScript είναι client side. Δηλαδή εκτελείται στη μεριά του πελάτη – browser. Για να είναι δυνατή η εκτέλεση της θα

πρέπει ο browser να έχει ρυθμιστεί κατάλληλα (να έχει ενεργοποιηθεί η επιλογή για JavaScript).^[19]

Το σημαντικό πλεονέκτημα της JavaScript είναι πως δεν επιβαρύνει καθόλου τον server. Δεν έχει καμία απαίτηση από αυτόν και δεν χρησιμοποιεί καθόλου τους πόρους του ή υπολογιστική ισχύ. Αντιθέτως, επιβαρύνει υπολογιστικά μόνο τον client παρέχοντάς του ωστόσο πιο γρήγορα αποτελέσματα.^[19]

Το σημαντικό μειονέκτημα της JavaScript είναι πως δεν είναι σε θέση να συνδεθεί απευθείας (όπως η PHP) με τη Βάση Δεδομένων του server. Είναι ωστόσο εφικτό, να έχει στη διάθεσή της δεδομένα από τη Βάση Δεδομένων του server, τα οποία θα βρίσκονται αποθηκευμένα στη μεριά του client. Αυτή η τεχνική, της ασύγχρονης ανταλλαγής δεδομένων μεταξύ client και server είναι γνωστή ως AJAX.^[19]

ΚΕΦΑΛΑΙΟ 4

Ανάπτυξη Λογισμικού

Σε αυτό το κεφάλαιο παρουσιάζεται η ανάπτυξη λογισμικού της εφαρμογής Αγγλικής Δημοπρασίας. Αρχικά παρουσιάζεται η σχεδίαση της Βάσης Δεδομένων και η υλοποίηση της μέσω του εργαλείου phpMyAdmin. Παρουσιάζονται, η δημιουργία των κατάλληλων πινάκων και ο ρόλος τους στην εφαρμογή. Το απαραίτητο λογισμικό για την παρούσα διπλωματική εργασία αναπτύχθηκε σε δύο στάδια. Στο πρώτο στάδιο (παράγραφος 4.2) αναπτύχθηκε η εφαρμογή της διαδικτυακής Αγγλικής Δημοπρασίας. Στην παράγραφο αυτή παρουσιάζονται αναλυτικά οι υλοποιήσεις των συντελεστών μιας δημοπρασίας (διοργανωτής δημοπρασίας, αγοραστής) με τη χρήση html, php και MySQL, ώστε να ανταποκρίνονται στις απαιτήσεις του πρωτοκόλλου Αγγλικής Δημοπρασίας. Η υλοποίηση των κανόνων της δημοπρασίας όπως αυτές αναφέρονται στο πρωτόκολλο. Στο δεύτερο στάδιο (παράγραφος 4.3) αναπτύχθηκε ο μηχανισμός προσομοίωσης της Αγγλικής Δημοπρασίας. Η προσομοίωση επιτυγχάνεται με τη χρήση πρακτόρων ως αγοραστών και των λειτουργιών του 1^{ου} σταδίου. Καταγράφονται οι απαραίτητες αλλαγές σε επίπεδο πηγαίου κώδικα, ώστε να υποστηριχθεί η προσομοίωση και αναλύεται η υλοποίηση των πρακτόρων μαζί με τον απλό αλγόριθμο απόφασης.

4.1 Σχεδίαση και δημιουργία κατάλληλης Βάσης Δεδομένων μέσω phpMyAdmin

Η πρώτη φάση ανάπτυξης της εφαρμογής αποτέλεσε η δημιουργία της κατάλληλης βάσης δεδομένων για της ανάγκες της εφαρμογής. Η βάση που δημιουργείται ονομάζεται **database**. Για να υποστηριχθούν οι κανόνες του πρωτοκόλλου Αγγλικής Δημοπρασίας όσο για να επιτευχθούν οι απαραίτητες λειτουργίες που απαιτεί το πρωτόκολλο (σωστή και έγκυρη επικοινωνία client-server, συγχρονισμός μεταξύ αγοραστών και διοργανωτή δημοπρασίας) δημιουργούνται οι παρακάτω tables :

bidders table : Περιέχει ένα μοναδικό πεδίο (username). Το πεδίο αυτό ορίζεται σαν varchar (20) και ως primary key του table ώστε να μην υπάρχει δυνατότητα εγγραφής παραπάνω του ενός αγοραστή με το ίδιο username. Για κάθε νέο αγοραστή που συμμετέχει στη δημοπρασία δημιουργείται μια νέα εγγραφή από τον διοργανωτή – server στον bidders με το unique username που επέλεξε ο αγοραστής. Όπως θα παρουσιαστεί και στις επόμενες παραγράφους ο bidders table αρχικοποιείται με την τιμή εκκίνησης της δημοπρασίας η οποία ορίζεται από τον διοργανωτή της δημοπρασίας.

bids table : Περιέχει δυο πεδία. Τα πεδία αυτά είναι το username (varchar(20)) και το bid (int(11)). Σκοπός του είναι να αποθηκεύει το ιστορικό των προσφορών των εκάστοτε αγοραστών. Κάθε εγγραφή περιέχει το unique username του αγοραστή μαζί με την προσφορά που έχει πραγματοποιήσει.

time table : Περιέχει ένα μοναδικό πεδίο (time), με τύπο int(11). Στον time αποθηκεύεται πάντα η χρονική στιγμή μέχρι την οποία μπορεί να πραγματοποιηθεί μια νέα βελτιωμένη προσφορά από αγοραστή. Στον time table υπάρχει πάντα μία μοναδική εγγραφή. Για κάθε νέα, αποδεκτή από το διοργανωτή της δημοπρασίας, προσφορά αγοραστή, η τιμή της εγγραφής ανανεώνεται με νέα χρονική στιγμή. Είναι προφανές πως η τιμή αυτή είναι πάντα μελλοντική και προκύπτει από άθροισμα της χρονικής στιγμής που γίνεται η τελευταία προσφορά, με το χρονικό διάστημα που ορίζεται από το διοργανωτή ως χρόνος υποβολής νέας προσφοράς (στα πλαίσια της παρούσας εργασίας ο χρόνος αυτός ορίζεται στα 20 δευτερόλεπτα όπως θα αναλυθεί και στις επόμενες παραγράφους).

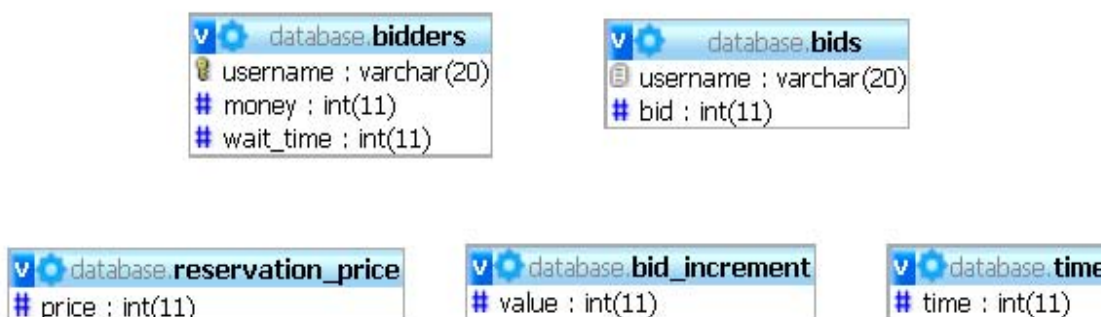
reservation_price : Περιέχει ένα μοναδικό πεδίο (price), με τύπο int(11). Στον reservation_price table αποθηκεύεται η τιμή πάνω από την οποία ο διοργανωτής πουλάει το προϊόν. Η τιμή αυτή είναι γνωστή μόνο στον διοργανωτή. Ο πίνακας αυτός περιέχει πάντα μόνο μία εγγραφή (μία εγγραφή για κάθε προϊόν δημοπράτησης, ωστόσο στην περίπτωση μας δημοπρατείτε πάντα ένα πιάνο) η οποία αρχικοποιείται με την reservation price του διοργανωτή πριν την έναρξη της δημοπρασίας.

Σχηματικά η παραπάνω βάση δεδομένων αναπαριστάται σύμφωνα με την παρακάτω εικόνα :



Εικόνα 4.1: Γραφική Απεικόνιση της Βάσης Δεδομένων του 1^{ου} Σταδίου Υλοποίησης.

Στο δεύτερο στάδιο της υλοποίησης (στάδιο προσομοίωσης) ο bidders table μεταβάλλεται. Πλέον τη θέση των αγοραστών – ανθρώπων, αναλαμβάνουν πράκτορες. Στον bidders table προστίθενται δύο ακόμη πεδία (το money με τύπο int(11) και το wait_time με τύπο int(11)), ενώ οι εγγραφές γίνονται μόνο κατά την αρχικοποίηση της προσομοίωσης, πριν την εκκίνηση της. Δημιουργείτε επίσης ο table bid_increment με ένα πεδίο (value), τύπου int(11). Στον πίνακα αυτό αποθηκεύεται το βήμα δημοπρασίας (bid increment) που δίνεται σαν παράμετρος κατά την αρχικοποίηση της προσομοίωσης. Το νέο σχήμα της Βάσης Δεδομένων που προκύπτει είναι το παρακάτω:



Εικόνα 4.2: Γραφική Απεικόνιση της Βάσης Δεδομένων του 2^{ου} Σταδίου Υλοποίησης (Προσομοίωση).

Η δημιουργία των παραπάνω Βάσεων Δεδομένων πραγματοποιήθηκε με τη χρήση του εργαλείου phpMyAdmin του προγράμματος Uniform Server version 5.6.13-Nano.

4.2 Υλοποίηση Διαδικτυακής Αγγλικής Δημοπρασίας

4.2.1 Αρχικοποίηση Συστήματος

Το πρωτόκολλο Αγγλικής Δημοπρασίας απαιτεί πως ο διοργανωτής της δημοπρασίας γνωρίζει εκ των προτέρων ορισμένες παραμέτρους που αφορούν τη δημοπρασία. Επίσης, για την υλοποίηση του πρωτοκόλλου σαν μια Real-time εφαρμογή απαιτούνται ορισμένες παραδοχές ως προς το χρόνο αναμονής του διοργανωτή για νέα προσφορά από αγοραστή. Οι παράμετροι, μαζί με τις όποιες παραδοχές του συστήματος οφείλουν να αρχικοποιηθούν πριν την έναρξη της δημοπρασίας.

Το σύστημα, για το πρώτο στάδιο της υλοποίησης (Online English Auction) αρχικοποιείται με τους εξής τρόπους:

1. Γίνεται εισαγωγή στον bids table της πλειάδας (“auctioneer” , “1000”). Η εγγραφή αυτή αποτελεί ουσιαστικά την τιμή εκκίνησης της δημοπρασίας. Θυμίζουμε πως για τις ανάγκες δοκιμής της εφαρμογής χρησιμοποιούμε τη δημοπρασία ενός πιάνου, που αποτελεί το μοναδικό προς δημοπράτηση προϊόν. Η εισαγωγή αυτή στη Βάση Δεδομένων γίνεται μέσω του phpMyAdmin με την sql query `“INSERT INTO bids VALUES ('auctioneer', '1000')”`.
2. Γίνεται εισαγωγή στον πίνακα reservation_price της πλειάδας (“2000”). Η εγγραφή αυτή αποτελεί την τιμή όριο της δημοπρασίας. Κάτω από αυτήν την τιμή το προϊόν δεν πωλείται από τον διοργανωτή. Η τιμή αυτή χρησιμοποιείται κατά το τελευταίο στάδιο της δημοπρασίας, όταν πλέον η χρονική περίοδος προσφορών από τους αγοραστές έχει λήξει, για να αποφασιστεί αν θα υπάρξει νικητής της δημοπρασίας ή όχι. Η εισαγωγή της παραπάνω εγγραφής στη Βάση Δεδομένων γίνεται μέσω του phpMyAdmin με την sql query `“INSERT INTO reservation_price VALUES ('2000')”`.
3. Αποφασίζεται το βήμα δημοπρασίας (bid increment). Στην περίπτωση μας (δημοπρασία ενός πιάνου με τιμή εκκίνησης τα 1000 ευρώ) ορίζουμε ως βήμα δημοπρασίας την τιμή των 100 ευρώ. Κάθε βελτιωμένη προσφορά από αγοραστή θα πρέπει να είναι τουλάχιστον κατά 100 ευρώ υψηλότερη από την τελευταία βελτιωμένη τιμή. Αν δεν συμβαίνει αυτό η προσφορά θεωρείται άκυρη. Το βήμα δημοπρασίας ορίζεται μέσα στον πηγαίο κώδικα με την σταθερά 100.
4. Αποφασίζεται ο χρονική διάρκεια αναμονής του διοργανωτή για την υποβολή νέων προσφορών από τους αγοραστές. Στην περίπτωση της παρούσας εργασίας για τις ανάγκες της δοκιμής και της παρουσίασης της εργασίας, ο χρόνος αυτός περιορίζεται στα 20 δευτερόλεπτα. Ο χρόνος αναμονής ορίζεται με την τιμή 20 (δευτερόλεπτα) στα σημεία του πηγαίου κώδικα που χρησιμοποιείται.

4.2.2 Εγγραφή Νέου Αγοραστή στη Δημοπρασία

Για να την εγγραφή νέου αγοραστή στην δημοπρασία ο ενδιαφερόμενος πρέπει να επιλέξει ένα unique username το οποίο και θα τον προσδιορίζει. Έτσι στην αρχική σελίδα του ιστοτόπου (index.php) υπάρχει η κατάλληλη φόρμα την οποία ο χρήστης – υποψήφιος αγοραστής συμπληρώνει με το username που επιθυμεί να λάβει. Αξίζει να σημειωθεί πως στα πλαίσια της παρούσας εργασίας δεν ζητούνται περεταίρω στοιχεία του χρήστη (π.χ. ονοματεπώνυμο, email για ταυτοποίηση, τρόποι πληρωμής), καθώς το ζητούμενο είναι η υλοποίηση του πρωτοκόλλου με τις λειτουργίες και τους κανόνες που αυτό επιβάλλει.

Μόλις ο χρήστης συμπληρώσει τη φόρμα και επιλέξει Register (μέσω του αντίστοιχου button της φόρμας), η επεξεργασία μεταβαίνει στο αρχείο register.php. Εκεί γίνεται ο έλεγχος από το server για τη διαθεσιμότητα του ή όχι του επιθυμητού username. Ο παραπάνω έλεγχος γίνεται μέσω του πίνακα bidders της Βάσης Δεδομένων, όπου αποθηκεύονται τα usernames όλων των αγοραστών. Στον bidders σκόπιμα έχει οριστεί το μοναδικό του πεδίο (username) ως primary key και notNull. Με αυτόν τον τρόπο γίνεται πιο εύκολος ο έλεγχος για τη διαθεσιμότητα του username ο οποίος επιτυγχάνεται ως εξής :

- Αρχικά λαμβάνεται ο συνολικός αριθμός των εγγραφών στον bidders table με τον παρακάτω κώδικα php:

```
$query = "SELECT * from bidders";
```

```
$num1 = mysql_query($query);
```

```
$num1 = mysql_num_rows($num1);
```

- Στη συνέχεια με τον παρακάτω κώδικα php ζητείται να γίνει εισαγωγή της πλειάδας με το επιθυμητό username στον bidders:

```
$query = "INSERT INTO bidders VALUES ('$username')";
```

```
mysql_query($query);
```

Θυμίζουμε πως το username στον πίνακα bidders έχει οριστεί ως primary key και notNull. Έτσι αν υπάρχει εγγραφή με το συγκεκριμένο username δεν θα γίνει νέα και ακόμη, αν ο χρήστης δεν έχει δώσει username (κενή φόρμα) πάλι δεν θα πραγματοποιηθεί εγγραφή.

- Τέλος, λαμβάνεται ο νέος συνολικός αριθμός των εγγραφών στον bidders με τον php κώδικα:

```
$query = "SELECT * from bidders";
```

```
$num2 = mysql_query($query);
```

```
$num2 = mysql_num_rows($num2);
```

Με τη σύγκριση των τιμών \$num1 και \$num2 προκύπτει το συμπέρασμα για το αν έγινε η εγγραφή ή όχι. Για (\$num1 == \$num2) τότε δεν έγινε η εγγραφή, συνεπώς το

username δεν είναι διαθέσιμο, ενώ για ($\$num2 > \$num1$) η εγγραφή πραγματοποιήθηκε επιτυχώς.

Στην περίπτωση που το username δεν είναι διαθέσιμο, εκτυπώνεται ανάλογα μήνυμα στην πλευρά του χρήστη “*Username X already in use. Please choose another one.*” και ο χρήστης καλείται να επιλέξει διαφορετικό username. Στην περίπτωση που η εγγραφή γίνει επιτυχώς ο χρήστης μεταβαίνει στην ιστοσελίδα της δημοπρασίας (auction.php).

4.2.3 Ο Διοργανωτής της Δημοπρασίας (auctioneer)

Ο διοργανωτής της δημοπρασίας υλοποιείται στη μεριά του server. Έχει ως αρμοδιότητες την εκτέλεση συγκεκριμένων λειτουργιών και την τήρηση των κανόνων της δημοπρασίας. Ο πηγαίος κώδικας που αφορά τον διοργανωτή είναι γραμμένος αποκλειστικά σε γλώσσα php. Πρέπει να σημειωθεί πως ενώ ο διοργανωτής υλοποιείτε αποκλειστικά στην μεριά του server, ο server δεν υλοποιεί μόνο τον διοργανωτή. Οι διάφορες λειτουργίες επικοινωνίας μεταξύ αγοραστών και διοργανωτή, που επιτυγχάνονται με client requests – server responses υλοποιούνται επίσης από τον server.

4.2.3.1 Εκκίνηση της Δημοπρασίας

Για να ξεκινήσει η δημοπρασία απαιτείται να υπάρχει τουλάχιστον ένας αγοραστής εγγεγραμμένος. Το αν η δημοπρασία βρίσκεται σε εξέλιξη προσδιορίζεται από την εγγραφή στον time table της Βάσης Δεδομένων. Αν υπάρχει εγγραφή εκεί (θυμίζουμε πως στον συγκεκριμένο πίνακα πάντα υπάρχει το πολύ μία εγγραφή) τότε η δημοπρασία βρίσκεται σε εξέλιξη. Η εκκίνηση της δημοπρασίας επιτυγχάνεται με τον εξής τρόπο :

Κάθε φορά που γίνεται μια πετυχημένη εγγραφή ενός νέου αγοραστή γίνεται έλεγχος για το αν υπάρχει εγγραφή στον time table. Ουσιαστικά δηλαδή γίνεται έλεγχος για το αν ο συγκεκριμένος αγοραστής είναι ο πρώτος που συμμετέχει στην δημοπρασία. Αν αυτό συμβαίνει τότε η χρονική στιγμή εκείνη αποτελεί και τη χρονική στιγμή εκκίνησης της δημοπρασίας. Γίνεται εγγραφή στον time table με τιμή $time() + 20$ (η συνάρτηση $time()$ αποτελεί μία php συνάρτηση που επιστρέφει τη χρονική στιγμή μετρημένη σε δευτερόλεπτα). Ο αγοραστής πλέον μεταφέρεται στο χώρο της δημοπρασίας. Εκεί σαν πρώτη προσφορά εμφανίζεται η τιμή εκκίνησης της δημοπρασίας. Αν υπάρχει εγγραφή στον time table τότε απλά ο αγοραστής μεταφέρεται στο χώρο των της δημοπρασίας όπου εμφανίζονται οι τελευταίες καλύτερες προσφορές

Η παραπάνω λειτουργικότητα υλοποιείτε με τον παρακάτω κομμάτι php κώδικα που βρίσκεται στην αρχή του αρχείου auction.php :

```
$query = "SELECT time FROM time";
$result = mysql_query($query);
$num = mysql_numrows($result);
if($num == 0){
    $time = time() + 20;
    $query = "INSERT INTO time VALUES ('$time')";
    mysql_query($query);
}
```

4.2.3.2 Διαχείριση Νέας Προσφοράς από Αγοραστή

Μια από τις λειτουργίες του διοργανωτή είναι να διαχειρίζεται τις προσφορές των αγοραστών. Να τις επεξεργάζεται και να αποφασίζει αν θα γίνουν δεκτές ή όχι σύμφωνα με τους κανόνες που θέτει το πρωτόκολλο. Η λειτουργικότητα αυτή υλοποιείται στο αρχείο bid.php. Τα βήματα που ακολουθούνται για την αποδοχή ή μη μίας προσφοράς είναι τα παρακάτω :

1. Γίνεται έλεγχος για το αν η προσφορά είναι χρονικά έγκυρη. Δηλαδή αν είναι μέσα στο χρόνο που ο διοργανωτής έχει ορίσει ως το χρόνο αναμονής για νέες προσφορές (ουσιαστικά αν έχουν περάσει τα 20 δευτερόλεπτα από την τελευταία αποδεκτή προσφορά από αγοραστή). Αυτό επιτυγχάνεται μέσω της μεταβλητής valid που ορίζεται σε κάθε νέο session αγοραστή. Αν η τιμή της μεταβλητής είναι true τότε ο διοργανωτής συνεχίζει τους ελέγχους για την αποδοχή της προσφοράς. Αν η τιμή είναι false τότε η προσφορά θεωρείται άκυρη.
2. Στη συνέχεια (και αν η προσφορά δεν έχει ήδη απορριφθεί), γίνεται ανάκτηση από τον bids table της Βάσης Δεδομένων της μεγαλύτερης έως τότε προσφοράς με το εξής κομμάτι php κώδικα :

```
$query = "SELECT * FROM bids ORDER BY bid DESC";
$result = mysql_query($query);
$num = mysql_numrows($result);
$higherbid = mysql_result($result,0,"bid");
```

Όπου η sql query επιστρέφει τις πλειάδες του bids ταξινομημένες κατά φθίνουσα σειρά, με βάση το πεδίο bid (η τιμή της προσφοράς). Η ζητούμενη καλύτερη προσφορά λοιπόν θα βρίσκεται στην πρώτη γραμμή των αποτελεσμάτων και αποθηκεύεται στη μεταβλητή \$higherbid.

3. Γίνεται έλεγχος για το αν η προσφορά, 1^{ov} είναι μεγαλύτερη από την τελευταία καλύτερη προσφορά αγοραστή και 2^{ov} είναι υψηλότερη τουλάχιστον κατά το βήμα δημοπρασίας (που στην παρούσα εργασία είναι 100 ευρώ) από την τελευταία καλύτερη προσφορά. Οι δύο αυτοί έλεγχοι διενεργούνται από το παρακάτω if statement :

```
if( ($bid <= $higherbid) || ( ($bid - $higherbid) < 100) ){  
    /*Ignore bid*/  
}
```

4. Η προσφορά γίνεται αποδεκτή. Αν η προσφορά περάσει επιτυχώς όλους τους ελέγχους, τότε γίνεται αποδεκτή και αποθηκεύεται στον bids table της Βάσης Δεδομένων με το εξής κομμάτι κώδικα :

```
$query = "INSERT INTO bids VALUES ('$username','$bid')";  
mysql_query($query);
```

Ο διοργανωτής κηρύσσει νέο κύκλο προσφορών για τους αγοραστές με διάρκεια 20 δευτερολέπτων. Για το λόγο αυτό η τιμή της time στον πίνακα time (η μοναδική εγγραφή σε αυτόν τον πίνακα) ανανεώνεται ώστε να περιέχει τη νέα χρονική στιγμή λήξης της δημοπρασίας. Το κομμάτι κώδικα για την παραπάνω λειτουργικότητα είναι:

```
$time = time() + 20;  
$query = "UPDATE time SET time='$time'";  
mysql_query($query);
```

4.2.3.3 Λήξη Δημοπρασίας - Ανακήρυξη Νικητή

Η τελευταία κατά σειρά λειτουργία που καλείται ο διοργανωτής να εκτελέσει σύμφωνα με το πρωτόκολλο είναι να κηρύξει τη λήξη της δημοπρασίας και να ανακηρύξει το νικητή. Δεν είναι απαραίτητο πως θα υπάρξει νικητής. Αν η τιμή της τελευταίας μεγαλύτερης προσφοράς από αγοραστή είναι μικρότερη από την επιδιωκόμενη τιμή (reservation price) του διοργανωτή τότε το προϊόν δεν πωλείται. Η λειτουργίες της λήξης της δημοπρασίας και της ανακήρυξης του νικητή υλοποιούνται στο αρχείο timer.php.

Η λήξη της δημοπρασίας κηρύσσεται όταν ο διαθέσιμος χρόνος για νέα βελτιωμένη προσφορά από αγοραστή έχει παρέλθει. Θυμίζουμε πως ο χρόνος λήξης της δημοπρασίας είναι αποθηκευμένος στον time table της Βάσης Δεδομένων. Ανακτώντας το χρόνο από τη βάση δεδομένων και συγκρίνοντας τον με την παρούσα χρονική στιγμή, διαπιστώνεται αν ο διαθέσιμος χρόνος έχει παρέλθει. Αν υπάρχει ακόμη διαθέσιμος χρόνος τότε αυτός γίνεται γνωστός στους αγοραστές. Αν δεν υπάρχει επιπλέον χρόνος τότε καλείται η συνάρτηση λήξης της δημοπρασίας (end_auction()). Ο πηγαίος κώδικας php που υλοποιεί την παραπάνω λειτουργία είναι:

```
$query = "SELECT time FROM time"
$result = mysql_query($query)'
$time = mysql_result($result,0,"time") - time();
if($time > 0){
    echo "<h1>$time seconds</h1><h3>left for a new bid...</h3>";
}
else{
    end_auction();
}
```

Η συνάρτηση `end_auction()` αποτελεί τη συνάρτηση που αποφασίζει αν υπάρχει νικητής στη δημοπρασία και ανακοινώνει το όνομα του (`username`). Για την απόφαση αυτή ανακτάται η τελευταία καλύτερη προσφορά αγοραστή από τον `bids` table της Βάσης Δεδομένων. Επίσης από τη Βάση ανακτάται και η επιδιωκόμενη τιμή του διοργανωτή (`reservation price`) από τον `reservation_price` table. Ο διοργανωτής συγκρίνει τις δύο τιμές και αν η τιμή της προσφοράς είναι μεγαλύτερη από την `reservation price`, ο αγοραστής που έκανε τη προσφορά ανακηρύσσεται και νικητής, ειδάλλως ανακοινώνεται πως το προϊόν δεν πουλήθηκε. Η υλοποίηση της συνάρτησης `end_auction()` με `php` φαίνεται παρακάτω :

```
function end_auction(){

    mysql_connect(localhost,root,root);
    @mysql_select_db(database) or die( "Unable to select database");

    $query = "SELECT * FROM bids ORDER BY bid DESC";
    $result = mysql_query($query);
    $num = mysql_numrows($result);
    $username = mysql_result($result,0,"username");
    $bid = mysql_result($result,0,"bid");

    $query = "SELECT * FROM reservation_price";
    $result = mysql_query($query);
    $num = mysql_numrows($result);
    $reservation_price = mysql_result($result,0,"price");

    $_SESSION['valid'] = 'false';

    if( $bid < $reservation_price){

        echo "<h1>The product is not sold.</h1><br>";
    }
    else{

        echo "<h1>And is sold to $username for $bid Euros.</h1>";
    }
}
```



```
}  
  
    echo "<a href=\"new_auction.php\">Start New Auction<a>";  
    mysql_close();  
}
```

Παρατηρώντας τον κώδικα της συνάρτησης end_auction() παρατηρούμε πως μεταβάλλει τη μεταβλητή “valid” του client session σε false (εντολή `$_SESSION['valid'] = 'false';`). Αυτό, όπως αναφέρεται και στην παράγραφο 4.2.3.2, γίνεται ώστε ο αγοραστής να μην έχει πλέον τη δυνατότητα να κάνει νέα προσφορά. Η μεταβολή αυτή στην τιμή της valid δεν θα γίνει μόνο στον αγοραστή με την καλύτερη προσφορά αλλά σε όλους τους αγοραστές από τη στιγμή που θα επιχειρήσουν να μάθουν τον διαθέσιμο χρόνο μέσω auto refresh όπως θα δουμε στις επόμενες παραγράφους.

4.2.4 Οι Αγοραστές (bidders)

Οι αγοραστές στο 1^ο στάδιο της υλοποίησης είναι clients στο πρωτόκολλο client – server πίσω από τους οποίους βρίσκονται άνθρωποι, οι οποίοι αποφασίζουν και κάνουν προσφορές μέσω των επιλογών που τους παρέχονται από την ιστοσελίδα.

Το user interface του χρήστη (η ιστοσελίδα της δημοπρασίας – auction.php) χωρίζεται σε τέσσερα τμήματα :

1. Το τμήμα παρουσίασης του προϊόντος.
2. Το τμήμα παρουσίασης των τελευταίων καλύτερων προσφορών από αγοραστές (Bid History).
3. Το τμήμα όπου εμφανίζεται ο διαθέσιμος χρόνος μέχρι την επόμενη προσφορά από αγοραστή. Σε αυτό το τμήμα ανακοινώνεται και ο νικητής της δημοπρασίας μετά τη λήξη της.
4. Το τμήμα όπου ο αγοραστής μπορεί να πραγματοποιήσει την προσφορά που επιθυμεί μέσω html form.

Το **τμήμα παρουσίασης του προϊόντος** εμφανίζει το προϊόν προς δημοπράτηση. Στην περίπτωσή μας αυτό είναι ένα πιάνο, συνεπώς εμφανίζεται η εικόνα του πιάνου. Ο εντολή - html tag που το επιτυγχάνει είναι :

```
<div style = "width:700px">  
      
</div>
```

Το τμήμα παρουσίασης των τελευταίων καλύτερων προσφορών από αγοραστής (**Bid History**) εμφανίζει τις πέντε τελευταίες προσφορές των αγοραστών. Πρώτη εμφανίζεται πάντα (και με πιο έντονο χρώμα) η πιο πρόσφατη προσφορά. Κάθε νέα προσφορά θα πρέπει να είναι μεγαλύτερη από αυτή. Προφανώς στην εκκίνηση της δημοπρασίας δεν εμφανίζονται άλλες προσφορές παρά μόνο η τιμή εκκίνησης της δημοπρασίας. Για να είναι έγκυρα τα δεδομένα – προσφορές που εμφανίζονται στον κάθε αγοραστή θα πρέπει το συγκεκριμένο τμήμα να ανανεώνεται συνεχώς. Δηλαδή να βρίσκεται σε συνεχή επικοινωνία με τον server ώστε οποιαδήποτε αλλαγή (νέα βελτιωμένη προσφορά από κάποιο αγοραστή) πραγματοποιηθεί ο αγοραστής να είναι ενήμερος.

Για αυτό το λόγο δημιουργήθηκε το αρχείο display_bids.php. Το αρχείο αυτό υλοποιεί τη λειτουργία του Bid History. Δηλαδή, ανακτά από τη Βάση Δεδομένων τα περιεχόμενα του bids table (πίνακας που περιέχει όλες τις προσφορές) και εμφανίζει στην μεριά του χρήστη τις πέντε καλύτερες προσφορές κατά φθίνουσα σειρά, μαζί με το username του αντίστοιχου αγοραστή. Η ανάκτηση όλων των δεδομένων του bids table κατά φθίνουσα σειρά πραγματοποιείτε με την εξής sql query :

```
$query = "SELECT * FROM bids ORDER BY bid DESC";
```

Ενώ το display των πέντε καλύτερων προσφορών πραγματοποιείται από τον εξής κώδικα php:

```
echo "<h1 style=\\"text-decoration:underline;\">BID HISTORY</h1>";
```

```
$i=0;
```

```
$username = mysql_result($result,$i,"username");
```

```
$bid = mysql_result($result,$i,"bid");
```

```
echo "<h3 style=\\"color :#FA3B06\">$username : $bid Euros</h3>";
```

```
$i++;
```

```
while( $i < $num){
```

```
    if($i > 4){
```

```
        break;
```

```
    }
```

```
    $username = mysql_result($result,$i,"username");
```

```
    $bid = mysql_result($result,$i,"bid");
```

```
    echo "<h3>$username : $bid Euros</h3>";
```

```
    $i++;
```

```
}
```

Το αρχείο display_bids.php αποτελεί όπως αναφέραμε ένα τμήμα του user interface. Δηλαδή ουσιαστικά αποτελεί ένα κομμάτι του αρχείου auction.php. Πιο συγκεκριμένα αποτελεί ένα div του αρχείου. Συνεπώς, ανανεώνοντας συνεχώς αυτό το div, τα δεδομένα θα είναι πάντα έγκυρα στη μεριά του κάθε χρήστη. Η λειτουργία αυτή (auto refresh) επιτυγχάνεται με τη χρήση της συνάρτησης setInterval σε γλώσσα JavaScript. Για τη χρήση της συγκεκριμένης συνάρτησης

χρειάζεται να ενσωματώσουμε στον ιστότοπο το JavaScript αρχείο jquery-latest.js. Το script που επιτυγχάνει το auto refresh είναι :

```
<script>
    var refreshId = setInterval(function() { $('#refr').load('display_bids.php'); }, 500);
</script>
```

Όπου refr είναι το id του div που εμφανίζει το αρχείο display_bids.php, και 500 είναι η περίοδος ανανέωσης της σελίδας (μετρημένη σε millisecond). Δηλαδή το Bid History ανανεώνεται κάθε 0.5 δευτερόλεπτα.

Το τμήμα όπου εμφανίζεται ο διαθέσιμος χρόνος μέχρι την επόμενη προσφορά από αγοραστή λειτουργεί με τρόπο παρόμοιο με αυτό του Bid History τμήματος. Όπως έχει ήδη αναφερθεί στον time table υπάρχει αποθηκευμένη η χρονική στιγμή λήξης της περιόδου υποβολής νέων προσφορών από τους αγοραστές (που σηματοδοτεί τη λήξη της δημοπρασίας). Για να γίνεται γνωστός ο εναπομείναντας χρόνος στους αγοραστές καθώς θα κυλούν τα δευτερόλεπτα, θα πρέπει το τμήμα αυτό της ιστοσελίδας να ανανεώνεται συνεχώς. Για αυτό το λόγο δημιουργούμε το αρχείο timer.php. Στο αρχείο αυτό μεταξύ άλλων λειτουργιών που υποστηρίζονται, υλοποιείται και ο υπολογισμός του διαθέσιμου χρόνου. Από τον timer table ανακτάτε η χρονική στιγμή λήξης, και πραγματοποιώντας την πράξη final_time – present_time προκύπτει το ζητούμενο. Ο php κώδικας που τροποποιεί την παραπάνω λειτουργία είναι:

```
mysql_connect(localhost,root,root);
@mysql_select_db(database) or die( "Unable to select database");
$query = "SELECT time FROM time";
$result = mysql_query($query);
$time = mysql_result($result,0,"time") - time();
mysql_close();
```

Προφανώς ο διαθέσιμος χρόνος εμφανίζεται και στη μεριά του αγοραστή. Αυτό γίνεται με'σω της εντολής :

```
echo "<h1>$time seconds</h1><h3>left for a new bid...</h3>";
```

Το αρχείο timer.php αποτελεί όπως και το display_bids.php προηγουμένως, ένα κομμάτι του user interface ή καλύτερα του αρχείου auction.php. Δηλαδή αποτελεί και αυτό ένα div. Με τον ίδιο τρόπο και την ίδια λογική όπως και στην περίπτωση του display_bids.php το div αυτό πρέπει να ανανεώνεται συνεχώς. Θέτοντας και πάλι σαν περίοδο ανανέωσης τα 0.5 δευτερόλεπτα, το script που επιτυγχάνει το auto refresh είναι:

```
<script>
    var refreshId = setInterval(function() { $('#timer').load('timer.php');}, 500);
</script>
```

Το αρχείο timer.php θυμίζουμε πως εκτός από το διαθέσιμο χρόνο που εμφανίζει στον αγοραστή, πραγματοποιεί και τον έλεγχο για τη λήξη της δημοπρασίας. Δηλαδή ελέγχει αν έχει παρέλθει ο διαθέσιμος χρόνος των 20 δευτερολέπτων. Σε αυτή την περίπτωση δεν εμφανίζεται ο διαθέσιμος χρόνος στο αντίστοιχο τμήμα στη σελίδα του αγοραστή, αλλά το αποτέλεσμα της δημοπρασίας μαζί με την επιλογή “Start New Auction”. Η επιλογή αυτή συμπεριλήφθηκε για να είναι γρηγορότερη και ευκολότερη η έναρξη νέας δημοπρασίας στις δοκιμές που έγιναν αλλά και στην παρουσίαση της παρούσας εργασίας. Η επιλογή “Start New Auction” αυτό που κάνει είναι να διαγράφει από τον bids table όλες τις εγγραφές εκτός από την τιμή εκκίνησης και να ανανεώνει το διαθέσιμο χρόνο στον time table. Η επιλογή αυτή υλοποιείται στο αρχείο new_auction.php από το οποίο ο αγοραστής αυτόματα μεταβαίνει στο χώρο της δημοπρασίας. Η υλοποίηση σε php κώδικα είναι :

```
mysql_connect(localhost,root,root);
@mysql_select_db(database) or die( "Unable to select database");
$query = "DELETE FROM bids";
mysql_query($query);
$query = "INSERT INTO bids VALUES ('auctioneer','1000)";
mysql_query($query);
$query = "DELETE FROM time";
mysql_query($query);
header("Location: auction.php");
```

Το τμήμα όπου ο αγοραστής μπορεί να πραγματοποιήσει την προσφορά που επιθυμεί αποτελείτε ουσιαστικά από μία html form. Εκεί ο αγοραστής μπορεί να τοποθετήσει την προσφορά που επιθυμεί και πατώντας το button bid η προσφορά του να σταλεί στον διοργανωτή. Ο διοργανωτής αποφασίζει για το αν η προσφορά πληροί τα κριτήρια – κανόνες για να γίνει αποδεκτή (παράγραφος 4.2.3.2). Ο html κώδικας για τη φόρμα είναι :

```
<h3><?php echo $_SESSION['username'];?>, place your bid...</h3>
<form action="bid.php" method="post"> <br>
Your Bid: <input type="text" name="bid" />
<button type="submit" >Bid</button>
</form>
```

Εκτός από τη φόρμα πονταρίσματος στο συγκεκριμένο τμήμα υπάρχει και μία επιλογή για έξοδο από τη δημοπρασία. Η “EXIT AUCTION” επιλογή, αυτό που κάνει είναι να διαγράφει τον αγοραστή από τη δημοπρασία, διαγράφοντας την πλειάδα με το αντίστοιχο username από τον bidders table της Βάσης Δεδομένων. Η διαδικασία της διαγραφής υλοποιείται στο αρχείο exit_auction.php και στη συνέχεια ο χρήστης μεταφέρεται στην αρχική σελίδα.

Σε αυτό το τμήμα του user interface εμφανίζονται και οι κανόνες διεξαγωγής της δημοπρασίας. Στην προκειμένη περίπτωση εμφανίζεται η τιμή του βήματος δημοπρασίας (bid increment), καθώς οι υπόλοιπες παράμετροι έχουν ήδη οριστεί. Αυτό επιτυγχάνεται με το html tag : `<h3>Bid Increment : 100 Euros</h3>`

Για τη ορθή λειτουργία των αγοραστών, εκτός από τα παραπάνω τμήματα που αφορούν κυρίως το user interface, ορίζονται και τα αντίστοιχα sessions για κάθε ένα από τους αγοραστές που συμμετέχουν στη δημοπρασία. Κάθε bidder session περιέχει δύο μεταβλητές, τις username και valid. Στη μεταβλητή username προφανώς αποθηκεύεται το unique username του αγοραστή. Το username αυτό χρησιμοποιείται σε διάφορα σημεία της εφαρμογής. Κυρίως το χρησιμοποιεί ο server για την εγγραφή νέας προσφοράς στον bids table, για το τμήμα Bid History του client. Η μεταβλητή valid παίρνει τις τιμές true ή false. Η valid χρησιμοποιείται από το σύστημα για να διαπιστωθεί αν ο αγοραστής είναι ελεύθερος για να κάνει προσφορά (τιμή true) ή όχι (τιμή false), που σημαίνει πως η δημοπρασία έχει λήξει. Οι τιμές της valid μεταβάλλονται κατά την εκκίνηση και κατά τη λήξη της δημοπρασίας. Στην εκκίνηση της ανατίθεται η τιμή "true". Πιο συγκεκριμένα κατά την εγγραφή νέου χρήστη, καθώς η δημοπρασία μπορεί ήδη να βρίσκεται σε εξέλιξη. Μετά τη λήξη της δημοπρασίας, η μεταβλητή valid παίρνει την τιμή false. Την ανάθεση αυτή αναλαμβάνει το αρχείο timer.php (που καθορίζει και τη λήξη της δημοπρασίας), το οποίο συνεχώς ανανεώνεται ώστε να είναι έγκυρο στην μεριά του αγοραστή. Οι δύο php εντολές που εκτελούν τις παραπάνω αναθέσεις είναι:

```
$_SESSION['valid'] = "true";
```

Και

```
$_SESSION['valid'] = "false";
```

4.3 Προσομοίωση Αγγλικής Δημοπρασίας

Στην παρούσα παράγραφο παρουσιάζεται η υλοποίηση της προσομοίωσης της Αγγλικής Δημοπρασίας. Όπως αναλύεται παρακάτω, η προσομοίωση βασίζεται στην υλοποίηση του 1^{ου} σταδίου (Online English Auction). Οι λειτουργίες του πρωτοκόλλου που υλοποιήθηκαν εκεί και αφορούν τους κανόνες και τη λειτουργικότητα του διοργανωτή της δημοπρασίας παραμένουν αμετάβλητες.

4.3.1 Αρχικοποίηση Πρακτόρων – Συστήματος

Για τις ανάγκες της παρούσας προσομοίωσης (όπως και κάθε άλλης προσομοίωσης) απαιτείται η αρχικοποίηση του συστήματος. Στην παρούσα εργασία, η αρχικοποίηση του συστήματος αφορά τον καθορισμό των τιμών των παραμέτρων της δημοπρασίας (που αφορούν κυρίως το διοργανωτή) και των χαρακτηριστικών που θα έχουν οι αγοραστές – πράκτορες (username, διαθέσιμα χρήματα, χρόνος αναμονής τους μέχρι την επόμενη προσφορά τους).

Η αρχική σελίδα του ιστότοπου της δημοπρασίας μεταβάλλεται ώστε πλέον να μην υπάρχει η φόρμα εγγραφής χρήστη – αγοραστή, αλλά μα ζητείτε ο αριθμός των αγοραστών – πρακτόρων της δημοπρασίας. Ο αριθμός αυτός έχει εύρος από 1 έως 5 (δημιουργία από έναν μέχρι και πέντε πράκτορες). Προφανώς, ο αριθμός αυτός μπορεί να είναι πολύ μεγαλύτερος από 5, ωστόσο για τις ανάγκες δοκιμής και παρουσίασης της προσομοίωσης ο αριθμός αυτός περιορίστηκε ώστε να είναι εύκολη η παρακολούθηση των πρακτόρων που δημιουργούνται.

Μετά την επιλογή του αριθμού των πρακτόρων η διαδικασία της αρχικοποίησης ζητά από το χρήστη την εισαγωγή των παραμέτρων της προσομοίωσης που αφορούν τον διοργανωτή και είναι :

- ◆ Τιμή εκκίνησης (initial price)
- ◆ Επιδιωκόμενη τιμή (reservation price)
- ◆ Βήμα δημοπρασίας (bid increment)

Για κάθε ένα από τους πράκτορες ο χρήστης – διαχειριστής της προσομοίωσης υποχρεούται να δώσει τα στοιχεία και τα χαρακτηριστικά του κάθε πράκτορα ξεχωριστά που περιλαμβάνουν :

- ◆ Username πράκτορα
- ◆ Διαθέσιμα χρήματα (Available money)
- ◆ Χρόνος αναμονής μέχρι τη νέα προσφορά (wait time)

Παράμετροι δημοπρασίας και χαρακτηριστικά πρακτόρων, περιλαμβάνονται σε μία html form. Πατώντας ο χρήστης το button “Save” της φόρμας, η επεξεργασία μεταβαίνει στο αρχείο save.php. Εκεί τα δεδομένα σώζονται στους κατάλληλους πίνακες της Βάσης Δεδομένων. Για τις ανάγκες της προσομοίωσης απαιτούνται ορισμένες αλλαγές στους πίνακες της Βάσης Δεδομένων και η δημιουργία κάποιων νέων (όπως περιγράφεται και στην παράγραφο 4.1). Στον bidders table εισάγουμε δύο νέα πεδία, τα money και wait_time με τύπο int(11) και τα δύο, ενώ δημιουργείται και ο πίνακας bid_increment με ένα πεδίο (value), για την αποθήκευση του βήματος δημοπρασίας (bid increment). Με δεδομένες τις αλλαγές στη Βάση Δεδομένων, οι καταχωρήσεις που πραγματοποιούνται από το αρχείο save.php είναι :

- Η τιμή εκκίνησης αποθηκεύεται στον bids table με πλειάδα καταχώρησης την ('auctioneer','τιμή εκκίνησης') με την sql query "INSERT INTO bids VALUES ('auctioneer','\$bid)" όπου price η τιμή που ανακτήθηκε από την φόρμα.
- Η επιδιωκόμενη τιμή αποθηκεύεται στον reservation_price table. Αυτό επιτυγχάνεται μέσω της sql "INSERT INTO reservation_price VALUES ('\$reservation_price)".
- Το βήμα δημοπρασίας αποθηκεύεται στον bid_increment table με την εξής sql query "INSERT INTO bid_increment VALUES ('\$value)".
- Για κάθε πράκτορα - αγοραστή γίνεται εισαγωγή της αντίστοιχης πλειάδας, στον bidders table. Ο php κώδικας που εκτελεί την παραπάνω λειτουργία είναι:

```
for($i=1; $i<$NumOfBidders+1; $i++){
```

```
    $field = "username".$i;
```

```
    $username = $_POST[$field];
```

```
    $field = "money".$i;
```

```
    $money = $_POST[$field];
```

```
    $field = "wait_time".$i;
```

```
    $wait_time = $_POST[$field];
```

```
    $query = "INSERT INTO bidders VALUES ('$username','$money', '$wait_time)";
```

```
    mysql_query($query);
```

```
}
```

Αξίζει να σημειωθεί πως οι αντίστοιχες τιμές των πεδίων της πλειάδας προς εγγραφή προκύπτουν συναρτήσει του δείκτη i. Έτσι για τον 1^ο, 2^ο ... πράκτορα τα ονόματα των μεταβλητών ορίζονται με την ένωση του ονόματος της μεταβλητής με την τιμή του δείκτη i δημιουργώντας έτσι ένα αλφαριθμητικό. Αυτή η διαδικασία πραγματοποιείται μέσω της εντολής " \$field = "variable name" . \$i " .

Μετά την αποθήκευση των παραπάνω παραμέτρων, ο χρήστης – δημιουργός της προσομοίωσης ενημερώνεται για την αποθήκευση όλων των δεδομένων και του δίνεται η επιλογή έναρξης της προσομοίωσης. Μόλις ο χρήστης επιλέξει την έναρξη της προσομοίωσης, αυτόματα ανοίγουν τόσα νέα tabs όσος και ο αριθμός των πρακτόρων που έχουν οριστεί. Κάθε νέο tab οδηγείται στο αρχείο auction.php και περιέχει τα χαρακτηριστικά του αντίστοιχου πράκτορα (username, available money, wait time). Η λειτουργία αυτή επιτυγχάνεται μέσω του JavaScript script :

```
<script>
```

```
window.open("auction.php?username=<?php echo $username;?>&money=<?php echo $money;?>&wait_time=<?php echo $wait_time;?>");
```

```
</script>
```

Το κάθε tab – πράκτορας περιέχει τα τμήματα του αγοραστή όπως παρουσιάστηκαν στην παράγραφο 4.2.4. Η διαφορά τους, όπως θα αναλυθεί και στην επόμενη παράγραφο, έγκειται στο γεγονός πως ο πράκτορας λειτουργεί αυτόνομα, χωρίς ανθρώπινο έλεγχο.

Το αρχικό tab του χρήστη – δημιουργού μετατρέπεται σε ένα μενού με τις εξής δύο επιλογές (αρχείο start.php):

1. **RESTART AUCTION:** Με αυτή την επιλογή γίνεται επανεκκίνηση της προσομοίωσης με τα ίδια χαρακτηριστικά πρακτόρων και δημοπρασίας. Η επανεκκίνηση επιτυγχάνεται μέσω του αρχείου new_auction.php, στο οποίο διαγράφονται όλες οι εγγραφές από τον bids table (εκτός αυτής της τιμής εκκίνησης (“auctioneer”, “initial price”)) και αρχικοποιείται ο χρόνος στον time table (current time + 20 seconds). Ο χρήστης οδηγείται πάλι στο ίδιο μενού, ενώ η προσομοίωση έχει παράλληλα ξανά ξεκινήσει.
2. **CREATE NEW AUCTION:** Με αυτή την επιλογή διαγράφονται όλα τα δεδομένα από τη Βάση Δεδομένων (όχι οι πίνακες, αλλά μόνον οι εγγραφές που υπήρχαν σε κάθε έναν). Ο χρήστης οδηγείται στην αρχική σελίδα ξεκινώντας τη διαδικασία της αρχικοποίησης από την αρχή. Η αρχική σελίδα (index.php) είναι αυτή που “αδειάζει” τη Βάση Δεδομένων που μέσω των αντίστοιχων sql delete queries. Ο αντίστοιχος php κώδικας βρίσκεται στην αρχή του αρχείου, ώστε πριν εκτελεστεί οποιαδήποτε άλλη λειτουργία να έχει “καθαριστεί” το σύστημα. Η θέση του php κώδικα στην αρχή του αρχείου index.php, εγγυάται πως κατά την έναρξη της αρχικοποίησης του συστήματος προσομοίωσης, Η Βάση Δεδομένων δεν θα περιέχει δεδομένα από προηγούμενες προσομοιώσεις.

Πριν γίνει ανάλυση της συμπεριφοράς των πρακτόρων – αγοραστών στην επόμενη παράγραφο, θα πρέπει να υπογραμμιστεί πως οι λειτουργίες του διοργανωτή της δημοπρασίας (δεν πρέπει να συγχέετε με το χρήστη - δημιουργό της προσομοίωσης) παραμένουν υλοποιημένες με τον ίδιο τρόπο που περιγράφηκε στο 1^ο στάδιο της υλοποίησης (παράγραφος 4.2.3). Δηλαδή Ο διοργανωτής λειτουργεί ακριβώς με το ίδιο τρόπο όπως και στο 1^ο στάδιο της υλοποίησης. Δηλαδή, ο διοργανωτής διαχειρίζεται τις προσφορές, ανανεώνει το χρόνο, κηρύσσει το τέλος της δημοπρασίας και ανακηρύσσει το νικητή (εάν υπάρχει). Ωστόσο συναντούνται και κάποιες διαφορές.

Το bid increment πλέον δεν ορίζεται μέσα στον κώδικα αλλά αντλείται από τη Βάση Δεδομένων, όπου αποθηκεύεται κατά την αρχικοποίηση όπως είδαμε παραπάνω. Το ίδιο συμβαίνει και με την Επιδιωκόμενη τιμή (reservation price). Επίσης καταργείται το αρχείο bid.php. Οι προσφορές πλέον δεν γίνονται από ανθρώπους – αγοραστές μέσω html φόρμας αλλά από λογισμικό (πράκτορας). Οι προσφορές πραγματοποιούνται πλέον, όπως θα παρουσιαστεί και στην επόμενη παράγραφο, μέσω του αρχείου bidder.php που υλοποιεί σε μεγάλο βαθμό το λογισμικό του πράκτορα.

4.3.2 Λειτουργία Πρακτόρων - Λήψη αποφάσεων

Στην προσομοίωση της Αγγλικής Δημοπρασίας στην παρούσα εργασία, κάθε tab που δημιουργείτε από το χρήστη - δημιουργό της προσομοίωσης αποτελεί και ένα πράκτορα. Κάθε τέτοιο tab, είναι ίδιο με αυτό ενός αγοραστή στο 1^ο στάδιο της υλοποίησης. Δηλαδή, περιέχει τα τμήματα, “παρουσίασης του προϊόντος”, “παρουσίασης των τελευταίων καλύτερων προσφορών από αγοραστές (Bid History)”, “τμήμα όπου εμφανίζεται ο διαθέσιμος χρόνος μέχρι την επόμενη προσφορά από αγοραστή”. Αυτά τα τμήματα είναι αυτούσια τα τμήματα του 1^{ου} σταδίου. Η διαφοροποίηση υπάρχει στο τμήμα όπου ο αγοραστής μπορεί να πραγματοποιήσει προσφορά. Σε αυτό το τμήμα δεν υπάρχει πλέον η html φόρμα για πραγματοποίηση προσφοράς. Αντικαθίσταται από τα χαρακτηριστικά του πράκτορα (εμφάνιση username και available money) και από ένα auto refresh html div, το οποίο φορτώνει το αρχείο bidder.php.

Στην υλοποίηση του 1^{ου} σταδίου ο αγοραστής – άνθρωπος αντλούσε τις απαραίτητες πληροφορίες οπτικά (από την ιστοσελίδα δημοπρασίας). Στην προσομοίωση με πράκτορες ωστόσο, η ενημέρωση των αγοραστών - πρακτόρων με τα απαραίτητα στοιχεία γίνεται μέσω client request – server response. Το ρόλο αυτό, μεταξύ άλλων, παίζει το αρχείο bidder.php. Το αρχείο αυτό περιέχει κομμάτι κώδικα που αφορά τόσο τον αγοραστή όσο και το διοργανωτή. Μέσω του παρακάτω JavaScript script,

`<script>`

```
var refreshId = setInterval(function() { $('#bidder').load('bidder.php?username=?php echo $username;?>&money=?php echo $money;?>');}, <?php $wait_time = $wait_time*1000; echo $wait_time;?>);
```

`</script>`

ο πράκτορας ανά τακτά χρονικά διαστήματα (το χρονικό διάστημα ορίζεται από την τιμή της μεταβλητής wait_time, λαμβάνει τιμή κατά την αρχικοποίηση των πρακτόρων και διαφέρει από πράκτορα σε πράκτορα), κάνει client-request για να λάβει τα πιο πρόσφατα δεδομένα της δημοπρασίας (ουσιαστικά την τελευταία καλύτερη προσφορά και το bid increment) από τον server. Γνωρίζοντας την τελευταία καλύτερη προσφορά αποφασίζει, αν πραγματοποιήσει νέα βελτιωμένη προσφορά. Δηλαδή ο πράκτορας κάθε συγκεκριμένο, ορισμένο στην αρχικοποίηση, χρονικό διάστημα ενημερώνετε για την τρέχουσα καλύτερη προσφορά και πράττει ανάλογα.

Ως στόχος του κάθε πράκτορα ορίζεται η αγορά του προϊόντος στη χαμηλότερη δυνατή τιμή. Για αυτό το λόγο κάθε νέα προσφορά του πράκτορα προκύπτει ως άθροισμα της καλύτερης τρέχουσας προσφοράς και του βήματος δημοπρασίας. Ο πράκτορας κάνει βελτιωμένες προσφορές έως ότου εξαντληθούν τα διαθέσιμα χρήματα που διαθέτει (available money). Έτσι λοιπόν, ο πράκτορας μέσω του αρχείου bidder.php μαθαίνει την τελευταία καλύτερη προσφορά και το username του αγοραστή με τον εξής php κώδικα:

```
$query = "SELECT * FROM bids ORDER BY bid DESC";
```

```
$result = mysql_query($query);
```

```
$num = mysql_numrows($result);
```

```
$Winusername = mysql_result($result,$i,"username");
```

```
$Winbid = mysql_result($result,$i,"bid");
```

Αν είναι αυτός που έχει πραγματοποιήσει την καλύτερη προσφορά δεν πραγματοποιεί νέα. Επίσης εάν η καλύτερη προσφορά μαζί με το βήμα δημοπρασίας υπερβαίνουν τα διαθέσιμα χρήματα του δεν πραγματοποιεί προσφορά. Σε αντίθετη περίπτωση (δηλαδή δεν είναι δική του η τελευταία καλύτερη προσφορά και τα διαθέσιμα χρήματά του είναι αρκετά) πραγματοποιεί νέα προσφορά. Ο πηγαίος κώδικας που περιγράφει την παραπάνω λειτουργία είναι :

```
if( ($money > ($Winbid + $value )) && ( $Winusername != $username)){
```

```
    $Winbid += $value;
```

```
    $query = "INSERT INTO bids VALUES ('$username','$Winbid)";
```

```
    mysql_query($query);
```

```
    /*auctioneer code, updating time*/
```

```
}
```

Μετά τη νέα πρόταση του ο πράκτορας αναμένει για wait_time μέχρι να επιχειρήσει νέα προσφορά.

Πρέπει να σημειωθεί πως ο παραπάνω κώδικας προστατεύεται από κώδικα διοργανωτή και αφορά το χρονικό περιθώριο που έχουν οι αγοραστές για να υποβάλλουν νέα βελτιωμένη πρόταση (20 δευτερόλεπτα). Ο έλεγχος αυτός επιτυγχάνεται με τον εξής κώδικα php:

```
$query = "SELECT time FROM time";
```

```
$result = mysql_query($query);
```

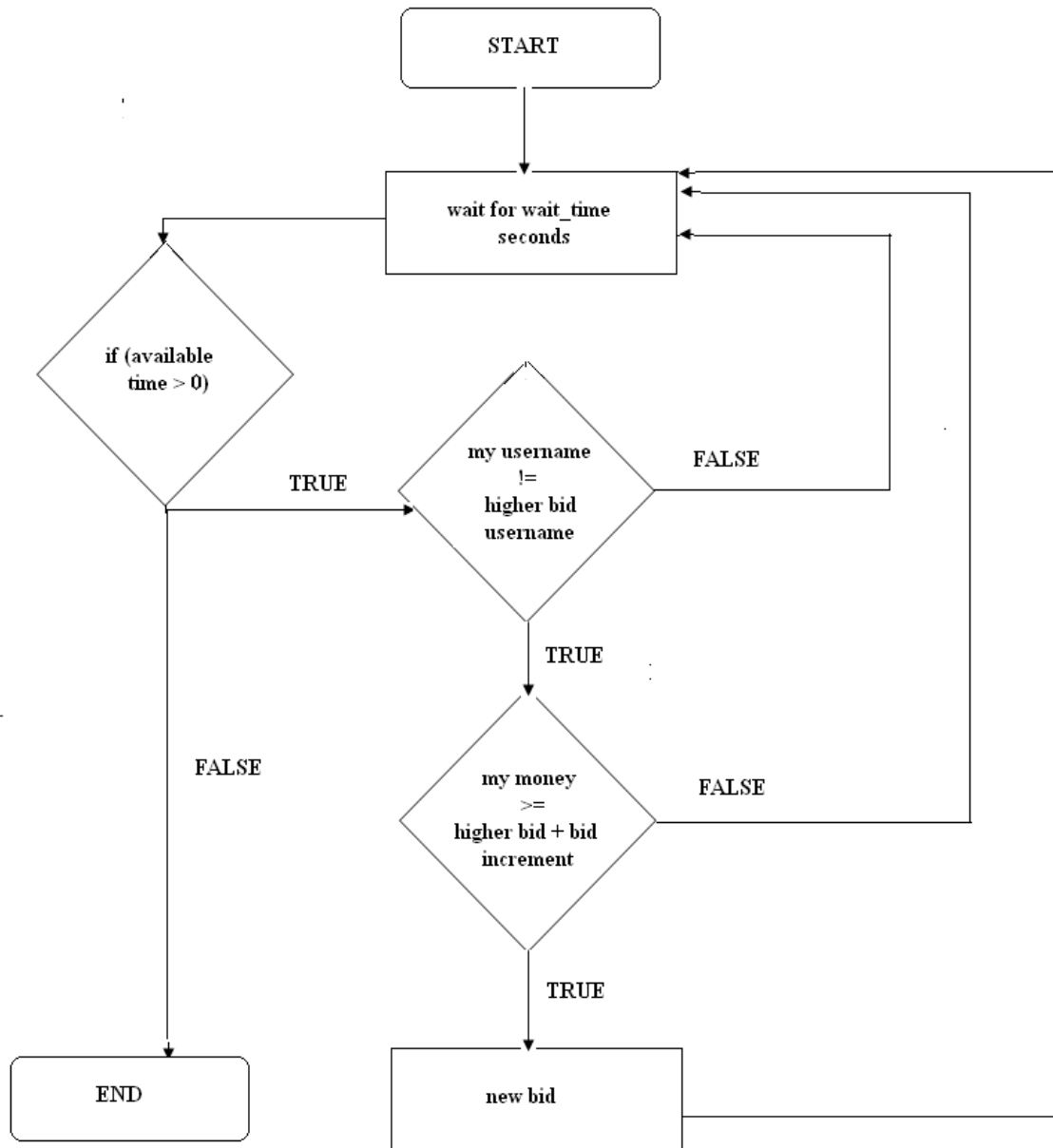
```
$time = mysql_result($result,0,"time");
```

```
if(time() < $time){
```

```
    /*Go on the bidding process*/
```

```
}
```

Ακολουθεί το διάγραμμα ροής για τον αγοραστή πράκτορα, που περιγράφει τον τρόπο λήψης της απόφασης για ποντάρισμα από τον πράκτορα :



Εικόνα 4.2 : Διάγραμμα Ροής – Λήψης Απόφασης Πράκτορα

ΚΕΦΑΛΑΙΟ 5

Συμπεράσματα και Προτάσεις για Μελλοντική Έρευνα

Στο τελευταίο αυτό κεφάλαιο παρουσιάζονται τα αποτελέσματα της εργασίας. Καταγράφονται τυχόν προβλήματα που συναντήθηκαν κατά τη διάρκειά της και τυχόν παραδοχές που χρειάστηκαν να γίνουν στην ανάπτυξη λογισμικού, οι οποίες και αναλύονται. Γενικότερα πραγματοποιείται μία κριτική επί του τελικού αποτελέσματος και παρουσιάζονται οι όποιες παραλείψεις εντοπίστηκαν. Τέλος, γίνονται προτάσεις και δίνονται ιδέες για μελλοντική έρευνα πάνω στο αντικείμενο των δημοπρασιών. Τόσο στο κομμάτι που αφορά τις online δημοπρασίες ως αξιόπιστες εφαρμογές, όσο και στο κομμάτι της εκπαίδευσης κατάλληλων πρακτόρων για συμμετοχή σε δημοπρασίες.

Η υλοποίηση που αφορά την online English Auction (1^ο στάδιο υλοποίησης) αποτελεί, όπως προκύπτει και από την ανάλυση και περιγραφή του, μία πιστή εφαρμογή του πρωτοκόλλου. Η μοναδική ίσως παραδοχή, να εντοπίζεται στην αναμονή του διοργανωτή της δημοπρασίας ώστε να αποφασίσει με ασφάλεια ότι κανένας αγοραστής δεν είναι διατεθειμένος να προσφέρει μια νέα βελτιωμένη τιμή (στην υλοποίηση μας ο διοργανωτής αναμένει 20 δευτερόλεπτα μέχρι να κηρύξει το τέλος της δημοπρασίας).

Η εφαρμογή της υλοποίησης Online English Auction σε ένα πραγματικό περιβάλλον (online server) μπορεί να συναντήσει προβλήματα. Αυτό οφείλεται στο γεγονός πως η παρούσα υλοποίηση αποτελεί ουσιαστικά μια εφαρμογή πραγματικού χρόνου. Η επιτυχία μιας τέτοιας προσπάθειας εξαρτάται άμεσα από τα χαρακτηριστικά του server. Την γρήγορη ανταπόκριση του στα client requests, την ταχύτητα ανάκτησης δεδομένων από τη βάση δεδομένων (ταχύτητα απάντησης ερωτήσεων σε επίπεδο sql). Εξαρτάται ακόμη και από τον αριθμό των αγοραστών που θα συμμετέχουν στη δημοπρασία (μεγάλος αριθμός αγοραστών = μεγάλος φόρτος για τον server). Θέματα ασφάλειας και αξιοπιστίας επίσης αποτελούν ζήτημα. Στα πλαίσια της παρούσας εργασίας δεν αναπτύχθηκαν σοβαροί μηχανισμοί ασφάλειας συστήματος. Σε μια πραγματική εφαρμογή ωστόσο απαιτούνται μηχανισμοί άμυνας απέναντι σε οποιαδήποτε κακόβουλη προσπάθεια προς το σύστημα.

Όσον αφορά το 2^ο στάδιο της υλοποίησης, που αποτελεί και την προσομοίωση του συστήματος μέσω πρακτόρων:

1. Χρησιμοποιούνται με επιτυχία οι λειτουργίες του πρωτοκόλλου από το 1^ο σταδίου της υλοποίησης, που αποτελεί και ένα από τα ζητούμενα σε μια προσομοίωση.
2. Δημιουργούνται πράκτορες – αγοραστές οι οποίοι λειτουργούν αυτόνομα και με συγκεκριμένη ρεαλιστική συμπεριφορά.

Σε αυτό το στάδιο το μειονέκτημα εντοπίζεται στον τρόπο λήψης αποφάσεων των πρακτόρων. Δημιουργούνται πράκτορες που αποφασίζουν με γνώμονα μόνο τα διαθέσιμα χρήματα τους (μοναδικό κριτήριο, ωστόσο πάντα η απόφαση τους είναι προς όφελος τους). Αλγόριθμοι που περιγράφουν τρόπους λήψης αποφάσεων πρακτόρων σε δημοπρασία, απαιτούν είτε τη συμμετοχή των πρακτόρων σε παραπάνω από μία δημοπρασίες για το ίδιο προϊόν (εκεί οι πράκτορες αποφασίζουν σε ποια δημοπρασία συμφέρει να κάνουν προσφορά), είτε τη γνώση δεδομένων από δημοπρασίες του παρελθόντος.

Τα σημεία που εντοπίζονται ως πιθανά σημεία για μελλοντική έρευνα προκύπτουν κυρίως από την προηγούμενη παράγραφο και είναι:

1. Ανάπτυξη νέων ή χρήση ενδεδειγμένων μηχανισμών ασφάλειας για Real-time Online English Auction συστήματα.

2. Εκπαίδευση πρακτόρων για συμμετοχή τους σε δημοπρασίες. Υλοποίηση αλγορίθμων για την πιο σύνθετη λήψη αποφάσεων από πράκτορες που συμμετέχουν σε English Auction. Δημιουργία βάσης δεδομένων όπου ο πράκτορας θα μπορεί να αντλεί πληροφορίες για παλαιότερες παρόμοιες δημοπρασίες.

ΠΑΡΑΡΤΗΜΑ Α

Βάση Δεδομένων 1^ο Σταδίου Υλοποίησης (Online English Auction)

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
CREATE TABLE `bidders` (
  `username` varchar(20) NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
CREATE TABLE `bids` (
  `username` varchar(20) NOT NULL,
  `bid` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
CREATE TABLE `reservation_price` (
  `price` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
CREATE TABLE `time` (
  `time` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Πηγαίος Κώδικας 1^ο Σταδίου Υλοποίησης

Αρχείο index.php

```
<?php
    session_start();
?>

<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>English Auction Simulation</title>
<meta name="keywords" content="" />
<link href="default.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>

<div id="header">
    <div id="logo">
        <h1>English Auction</h1>
```

```
</div>

</div>
<div id="page" style="height:400px;">
  <br><h2 style="color:#AF9371;">Welcome to the Auction</h2><br>
<?php
  if(isset($_SESSION['valid'])){
    if($_SESSION['valid'] == "false"){
      echo "<p>Username {$_SESSION['username']} already in use. Please
      choose another one.</p>";
      echo " <form style=\"padding: 0px 0px 20px 0px\"
      action=\"register.php\" method=\"post\"> <br>
      Username: <input type=\"text\" name=\"username\" /> <br> <br>
      <button type=\"submit\" >Register</button>
      </form> ";
    }
    else{
      echo "<p>Hello {$_SESSION['username']}!! It is time you bid...</p>";
      echo "<a href=\"auction.php\"><h2>Auction</h2></a>";
    }
  }
  else{
    echo "<p>Please choose a username to participate:</p>";
    echo " <form style=\"padding: 0px 0px 20px 0px\"
    action=\"register.php\" method=\"post\"> <br>
    Username: <input type=\"text\" name=\"username\" /> <br> <br>
    <button type=\"submit\" >Register</button>
    </form> ";
  }
?>
</div>
</body>
</html>
```

Αρχείο auction.php

```
<?php
  session_start();
  mysql_connect(localhost,root,root);
  @mysql_select_db(database) or die( "Unable to select database");
  $query = "SELECT time FROM time";
  $result = mysql_query($query);
```



```
$num = mysql_numrows($result);
if($num == 0){
    $time = time() + 20;
    $query = "INSERT INTO time VALUES ('$time')";
    mysql_query($query);
}
mysql_close();
?>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>English Auction Simulation</title>
<meta name="keywords" content="" />
<link href="default.css" rel="stylesheet" type="text/css" media="screen" />

<script src="jquery-latest.js"></script>
<script>
    var refreshId = setInterval(function() { $('#refr').load('display_bids.php'); }, 500);
</script>
<script>
    var refreshId = setInterval(function() { $('#timer').load('timer.php');}, 500);
</script>
</head>
<body>
<div id="header">
    <div id="logo">
        <h1>English Auction</h1>
    </div>
    
</div>
<div id="page">
    <h3 style="padding:5px">Today's auctioned product is this piano:</h3>
    <div style = "width:700px">
        
    </div>
    <div id="timer" style="width: 700px; height: 120px; padding:10px 0px 0px 0px"></div>
    
    <div style="width:350px; float:left; padding:20px 0px 0px 0px;">
        <h3><?php echo $_SESSION['username'];?>, place your bid...</h3>
        <form action="bid.php" method="post"> <br>
        Your Bid: <input type="text" name="bid" />
        <button type="submit" >Bid</button>
        </form>
        <br><h3>Bid Increment : 100 Euros</h3>
    </div>
</div>
</body>
</html>
```

```
        <a href="exit_auction.php">EXIT AUCTION</a>
    </div>
    
    <div id="refr" style="width:340px; height:280px; float:right; padding:20px 0px 0px 0px;" >
    </div>
    
</div>
</body>
</html>
```

Αρχείο bid.php

```
<?php
    session_start();
    if($_SESSION['valid'] == "true"){
        $bid = $_POST['bid'];
        $username = $_SESSION['username'];
        mysql_connect(localhost,root,root);
        @mysql_select_db(database) or die( "Unable to select database");
        $query = "SELECT * FROM bids ORDER BY bid DESC";
        $result = mysql_query($query);
        $num = mysql_numrows($result);
        $higherbid = mysql_result($result,0,"bid");
        if( ($bid <= $higherbid) || ( ($bid - $higherbid) < 100) ){
        }
        else{
            $query = "INSERT INTO bids VALUES ('$username','$bid)";
            mysql_query($query);
            $time = time() + 20;
            $query = "UPDATE time SET time='$time'";
            mysql_query($query);
        }
        mysql_close();
    }
    else{
        mysql_connect(localhost,root,root);
        @mysql_select_db(database) or die( "Unable to select database");
        $bid = $_POST['bid'];
        $username = $_SESSION['username'];
        $query = "SELECT time FROM time";
        $result = mysql_query($query);
        $time = mysql_result($result,0,"time");
```

```
        if($time > time()){
            $_SESSION['valid'] = "true";
            if($bid <= mysql_result($result,0,"bid") ){
                }
            else{
                $query = "INSERT INTO bids VALUES ('$username','$bid')";
                mysql_query($query);
                $time = time() + 20;
                $query = "UPDATE time SET time='$time'";
                mysql_query($query);
            }
        }
        mysql_close();
    }
    header("Location: auction.php");
?>
```

Αρχείο display_bids.php

```
<?php
    session_start();
    mysql_connect(localhost,root,root);
    @mysql_select_db(database) or die( "Unable to select database");
    $query = "SELECT * FROM bids ORDER BY bid DESC";
    $result = mysql_query($query);
    $num = mysql_numrows($result);
    echo "<h1 style=\"text-decoration:underline;\">BID HISTORY</h1>";
    $i=0;
    $username = mysql_result($result,$i,"username");
    $bid = mysql_result($result,$i,"bid");
    echo "<h3>WINNING BID $username : $bid Euros</h3>";
    $i++;
    while( $i < $num){
        if($i > 4){
            break;
        }
        $username = mysql_result($result,$i,"username");
        $bid = mysql_result($result,$i,"bid");
        echo "<h3>$username : $bid Euros</h3>";
        $i++;
    }
}
```

```
mysql_close();  
?>
```

Αρχείο exit auction.php

```
<?php  
session_start();  
$username = $_SESSION['username'];  
session_destroy();  
mysql_connect(localhost,root,root);  
@mysql_select_db(database) or die( "Unable to select database");  
$query = "DELETE FROM bidders WHERE username = '$username'";  
mysql_query($query);  
mysql_close();  
header("Location: index.php");  
?>
```

Αρχείο new auction.php

```
<?php  
session_start();  
$_SESSION['valid'] = "true";  
mysql_connect(localhost,root,root);  
@mysql_select_db(database) or die( "Unable to select database");  
$query = "DELETE FROM bids";  
mysql_query($query);  
$query = "INSERT INTO bids VALUES ('auctioneer','1000)";  
mysql_query($query);  
$query = "DELETE FROM time";  
mysql_query($query);  
mysql_close();  
header("Location: auction.php");  
?>
```

Αρχείο register.php

```
<?php
    session_start();
    $username = $_POST['username'];
    $_SESSION['username'] = $username;
    mysql_connect(localhost,root,root);
    @mysql_select_db(database) or die( "Unable to select database");
    $query = "SELECT * from bidders";
    $num1 = mysql_query($query);
    $num1 = mysql_num_rows($num1);
    $query = "INSERT INTO bidders VALUES ('$username', '0')";
    mysql_query($query);
    $query = "SELECT * from bidders";
    $num2 = mysql_query($query);
    $num2 = mysql_num_rows($num2);
    mysql_close();
    if($num1 == $num2 ){
        $_SESSION['valid'] = "false";
        header("Location: index.php");
    }
    else{
        $_SESSION['valid'] = "true";
        header("Location: auction.php");
    }
?>
```

Αρχείο timer.php

```
<?php
    session_start();
    function end_auction(){
        mysql_connect(localhost,root,root);
        @mysql_select_db(database) or die( "Unable to select database");
        $query = "SELECT * FROM bids ORDER BY bid DESC";
        $result = mysql_query($query);
        $num = mysql_numrows($result);
        $username = mysql_result($result,0,"username");
        $bid = mysql_result($result,0,"bid");
```

```
$query = "SELECT * FROM reservation_price";
$result = mysql_query($query);
$num = mysql_numrows($result);
$reservation_price = mysql_result($result,0,"price");
$_SESSION['valid'] = 'false';
if( $bid < $reservation_price){
    echo "<h1>The product is not sold.</h1><br>";
}
else{
    echo "<h1>And is sold to $username for $bid Euros.</h1>";
}
echo "<a href=\"new_auction.php\">Start New Auction<a>";
mysql_close();
}
mysql_connect(localhost,root,root);
@mysql_select_db(database) or die( "Unable to select database");
$query = "SELECT time FROM time";
$result = mysql_query($query);
$time = mysql_result($result,0,"time") - time();
mysql_close();
if($time > 0){
    echo "<h1>$time seconds</h1><h3>left for a new bid...</h3>";
}
else{
    end_auction();
}
?>
```

Αρχείο default.css

```
body {
    margin: 0;
    padding: 0;
    text-align : center;
    background: #544C40 url(images/img01.jpg) repeat left top;
    font-family: "Trebuchet MS", Verdana, Arial, Helvetica, sans-serif;
    font-size: 13px;
    color: #FFFFFF;
}
h1, h2, h3 {
    margin-top: 0;
}
```

```
h1 {
    font-size: 1.6em;
    font-weight: normal;
}
h2 {
    font-size: 1.6em;
}
h3 {
    font-size: 1.2em;
}
a {
    text-decoration: underline;
    font-size: 1.2em;
    color: #AF9371;
}
a:hover {
    border-bottom: none;
}
#header {
    width: 700px;
    margin: auto;
    height: 148px;
    background: url(images/img02.jpg) repeat-x left top;
}
#logo {
    width: 700px;
    height: 148px;
    margin: 0 auto;
}
#logo h1 {
    font-weight: normal;
    font-size: 3em;
    padding: 50px 0 0 0px;
}
#page {
    width: 700px;
    height: 710px;
    margin: 0 auto;
    padding: 20px 0 0 0;
    background: #392C23;
}
```

ΠΑΡΑΡΤΗΜΑ Β

Βάση Δεδομένων 2^ο Σταδίου Υλοποίησης (Προσομοίωση English Auction)

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
CREATE TABLE `bidders` (
  `username` varchar(20) NOT NULL,
  `money` int(11) NOT NULL,
  `wait_time` int(11) NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
CREATE TABLE `bids` (
  `username` varchar(20) NOT NULL,
  `bid` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
CREATE TABLE `bid_increment` (
  `value` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
CREATE TABLE `reservation_price` (
  `price` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
CREATE TABLE `time` (
  `time` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Πηγαίος Κώδικας 2^ο Σταδίου Υλοποίησης

Αρχείο index.php

```
<?php
$NumOfBidders = $_GET['NumOfBidders'];
if($NumOfBidders == NULL){
    mysql_connect(localhost,root,root);
    @mysql_select_db(database) or die( "Unable to select database");
    $query = "DELETE FROM bids";
    mysql_query($query);
    $query = "DELETE FROM bidders";
    mysql_query($query);
    $query = "DELETE FROM reservation_price";
    mysql_query($query);
}
```



```
        $query = "DELETE FROM bid_increment";
        mysql_query($query);
        $query = "DELETE FROM time";
        mysql_query($query);
        mysql_close();
    }
?>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>English Auction Simulation</title>
<meta name="keywords" content="" />
<link href="default.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>
<div id="header">
    <div id="logo">
        <h1>English Auction Simulation</h1>
    </div>
    
</div>
<div id="page" style="height:900px;">
    <br><h2 style="color:#AF9371;">Initiliaze Auction</h2><br>
    <?php
    if($NumOfBidders == NULL){
    ?>
        <form action="index.php" method="get">
            <p>Select the number of bidders : </p>
            <select name="NumOfBidders">
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
                <option value="4">4</option>
                <option value="5">5</option>
            </select>
            <br><br>
            <button type="submit" >Create</button>
        </form>
    <?php
    }
    else{
        echo "<form style=\"padding: 0px 0px 20px 0px\"
        action=\"save.php?NumOfBidders=$NumOfBidders\" method=\"post\">";
        echo "<h3 style=\"text-decoration:underline\">AUCTION PARAMETERS</h3><br>";
        echo "Give the initial price of the product: <input type=\"int\" name=\"bid\"
        /><br><br>";
        echo "Give the reservation price of the product: <input type=\"int\" name=\"price\"
        /><br><br>";
        echo "Give the bid increment : <input type=\"int\" name=\"value\" /> ";
```

```
echo "<br><br><br><h3 style=\"text-decoration:underline\">AGENT
PARAMETERS</h3>";
for($i=1; $i<$NumOfBidders+1; $i++){
    echo "<h3>Agent Number $i</h3>";
    echo "Username: <input type=\"text\" name=\"username$i\" /><br><br>";
    echo "Money Available: <input type=\"int\" name=\"money$i\" /><br><br>";
    echo "Wait Time (in seconds): <input type=\"int\" name=\"wait_time$i\"
    /><br><br>";
?>
<?php
}
?>
        <br><br><button type="submit" >SAVE</button>
</form>
<?php
}
?>
</div>
</body>
</html>
```

Αρχείο auction.php

```
<?php
mysql_connect(localhost,root,root);
@mysql_select_db(database) or die( "Unable to select database");
$query = "SELECT time FROM time";
$result = mysql_query($query);
$num = mysql_numrows($result);
if($num == 0){
    $time = time() + 20;
    $query = "INSERT INTO time VALUES ('$time)";
    mysql_query($query);
}
$query = "SELECT * FROM bid_increment";
$result = mysql_query($query);
$num = mysql_numrows($result);
$value = mysql_result($result,0,"value");
mysql_close();
$username = $_GET['username'];
$money = $_GET['money'];
$wait_time = $_GET['wait_time'];
?>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>English Auction Simulation</title>
<meta name="keywords" content="" />
```

```
<link href="default.css" rel="stylesheet" type="text/css" media="screen" />
<script src="jquery-latest.js"></script>
<script>
    var refreshId = setInterval(function() { $('#refr').load('display_bids.php'); }, 100);
</script>
<script>
    var refreshId = setInterval(function() { $('#timer').load('timer.php');}, 500);
</script>
<script>
    var refreshId = setInterval(function() { $('#bidder').load('bidder.php?username=?php echo
    $username;?&money=?php echo $money;?>');}, <?php $wait_time = $wait_time*1000; echo
    $wait_time;?>);
</script>
</head>
<body>
<div id="header">
    <div id="logo">
        <h1>English Auction</h1>
    </div>

</div>
<div id="page">
    <h3 style="padding:5px">Today's auctioned product is this piano:</h3>
    <div style = "width:700px">
        
    </div>
    <div id="timer" style="width: 700px; height: 120px; padding:10px 0px 0px 0px"></div>
    
    <div style="width:350px; float:left; padding:20px 0px 0px 0px;">
        <h3 style="text-decoration:underline;">AGENT INFO</h3>
        <p style="font-size:1.2em;">Username : <?php echo $username;?></p>
        <p style="font-size:1.2em;">Available Money : <?php echo $money;?>
        Euros</p><br><br>
        <br><h3>Bid Increment : <?php echo $value;?> Euros</h3>
        <div id="bidder" style="width:350px; float:left; padding:20px 0px 0px 0px;"></div>
    </div>
    
    <div id="refr" style="width:340px; height:280px; float:right; padding:20px 0px 0px 0px;"
    ></div>
    
</div>
</body>
</html>
```

Αρχείο bidder.php

```
<?php
mysql_connect(localhost,root,root);
@mysql_select_db(database) or die( "Unable to select database");
$query = "SELECT time FROM time";
$result = mysql_query($query);
$time = mysql_result($result,0,"time");
if(time() < $time){
    $query = "SELECT * FROM bids ORDER BY bid DESC";
    $result = mysql_query($query);
    $num = mysql_numrows($result);
    $Winusername = mysql_result($result,$i,"username");
    $Winbid = mysql_result($result,$i,"bid");
    $query = "SELECT * FROM bid_increment";
    $result = mysql_query($query);
    $num = mysql_numrows($result);
    $value = mysql_result($result,0,"value");
    $money = $_GET['money'];
    $username = $_GET['username'];
    if( ($money > ($Winbid + $value )) && ( $Winusername != $username)){
        $Winbid += $value;
        $query = "INSERT INTO bids VALUES ('$username','$Winbid)";
        mysql_query($query);
        $time = time() + 20;
        $query = "UPDATE time SET time='$time'";
        mysql_query($query);
    }
}
mysql_close();
?>
```

Αρχείο timer.php

Βλέπε αντίστοιχο αρχείο Παραρτήματος Α.

Αρχείο display_bids.php

Βλέπε αντίστοιχο αρχείο Παραρτήματος Α.

Αρχείο new_auction.php

Βλέπε αντίστοιχο αρχείο Παραρτήματος Α.

Αρχείο default.css

Βλέπε αντίστοιχο αρχείο Παραρτήματος Α.

Αρχείο save.php

```
<?php
    $NumOfBidders = $_GET['NumOfBidders'];
?>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>English Auction Simulation</title>
<meta name="keywords" content="" />
<link href="default.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>
<div id="header">
    <div id="logo">
        <h1>English Auction Simulation</h1>
    </div>
    
</div>
<div id="page">
    <?php
        mysql_connect(localhost,root,root);
        @mysql_select_db(database) or die( "Unable to select database");
        $bid = $_POST['bid'];
        $query = "INSERT INTO bids VALUES ('auctioneer','$bid)";
        mysql_query($query);
        $price = $_POST['price'];
        $query = "INSERT INTO reservation_price VALUES ('$price)";
        mysql_query($query);
        $value = $_POST['value'];
        $query = "INSERT INTO bid_increment VALUES ('$value)";
        mysql_query($query);
        for($i=1; $i<$NumOfBidders+1; $i++){
            $field = "username".$i;
            $username = $_POST[$field];
            $field = "money".$i;
            $money = $_POST[$field];
            $field = "wait_time".$i;
            $wait_time = $_POST[$field];
            $query = "INSERT INTO bidders VALUES ('$username','$money',
            '$wait_time)";
            mysql_query($query);
        }
        mysql_close();
    ?>

    <br><br><h2>All DATA HAVE BEEN SAVED</h2><br>
    <h3>To start the simulation click <a href="start.php?flag=0">HERE</a></h3>
</div>
</body>
</html>
```

Αρχείο start.php

```
<?php
    $flag = $_GET['flag'];
    if($flag == 0){
        mysql_connect(localhost,root,root);
        @mysql_select_db(database) or die( "Unable to select database");
        $query = "SELECT * FROM bidders";
        $result = mysql_query($query);
        $num = mysql_numrows($result);
        $i=$num-1;
        while( $i > -1 ){
            $username = mysql_result($result,$i,"username");
            $money = mysql_result($result,$i,"money");
            $wait_time = mysql_result($result,$i,"wait_time");
        }
    }
    ?>
    <script>
        window.open("auction.php?username=<?php echo $username;?>&money=<?php
        echo $money;?>&wait_time=<?php echo $wait_time;?>");
    </script>
    <?php
        $i--;
    }
    mysql_close(); }
?>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>English Auction Simulation</title>
<meta name="keywords" content="" />
<link href="default.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>
<div id="header">
    <div id="logo">
        <h1>English Auction</h1>
    </div>
    
</div>
<div id="page">
    <h2 style="padding:5px; text-decoration:underline;">MENU</h2>
    <a href="new_auction.php">RESTART AUCTION</a><br><br>
    <a href="index.php">CREATE NEW AUCTION</a>
</div>
</body>
</html>
```

Αναφορές – Βιβλιογραφία

- [1] <http://en.wikipedia.org/wiki/Auction>
- [2] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, *FIPA English Auction Interaction Protocol Specification (2001)*.
- [3] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, *FIPA Dutch Auction Interaction Protocol Specification (2000)*.
- [4] <http://www.ibid.gr/help/onebid.asp>
- [5] http://en.wikipedia.org/wiki/Vickrey_auction
- [6] <http://www.ibid.gr/help/OpenAuction.asp>
- [7] http://www.epiqtech.com/auction_software-Internet-Auction.htm
- [8] <http://en.wikipedia.org/wiki/FIPA>
- [9] <http://www.fipa.org>
- [10] http://www.agentlab.de/protocol_specification.html
- [11] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, *FIPA Modeling: Interaction Diagrams*.
- [12] http://en.wikipedia.org/wiki/Unified_Modeling_Language
- [13] http://en.wikipedia.org/wiki/Client%E2%80%93server_model
- [14] http://en.wikipedia.org/wiki/The_Uniform_Server
- [15] http://www.phpmyadmin.net/home_page/index.php
- [16] <http://el.wikipedia.org/wiki/HTML>
- [17] <http://el.wikipedia.org/wiki/PHP>
- [18] <http://en.wikipedia.org/wiki/MySQL>
- [19] <http://el.wikipedia.org/wiki/JavaScript>