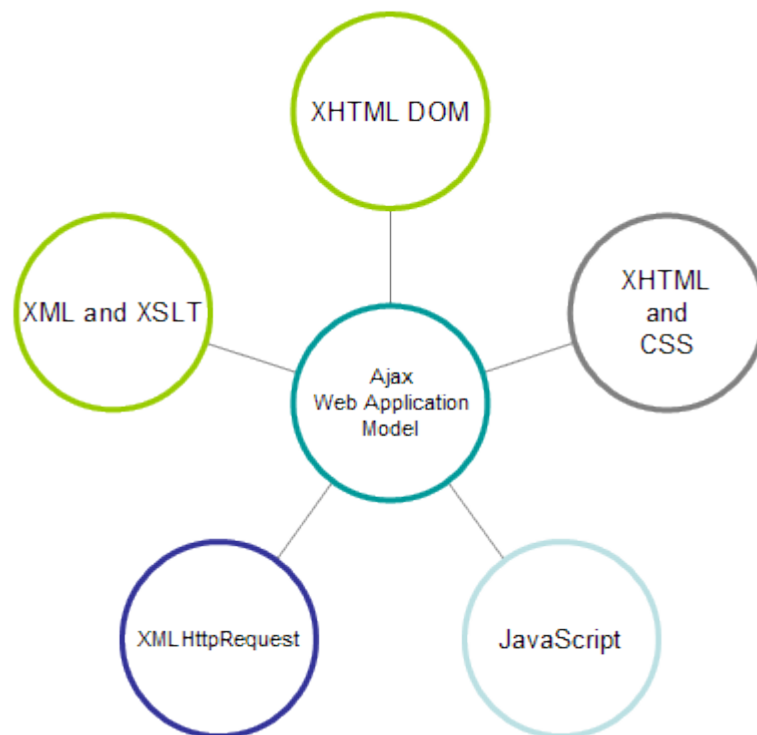


**ΑΝΩΤΑΤΟ
ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ
Θεσσαλονίκης**



***Οι τεχνολογίες της Web 2 εποχής
εφαρμοζόμενες σε σύστημα ηλεκτρονικών δημοπρασιών.***



Πτυχιακή εργασία των φοιτητών:
Μαρκόπουλος Γρηγόριος
Παναγιωτάκης Ευάγγελος

Επιβλέπων καθηγητής:
Μάργαρης Αθανάσιος

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΘΕΣΣΑΛΟΝΙΚΗ ΟΚΤΩΒΡΙΟΣ 2007**

Περιεχόμενα

Εισαγωγή.....	4
1. Διαδίκτυο.....	5
1.1 Ορισμός του Διαδικτύου.....	5
1.2 Η γέννηση του Διαδικτύου.....	5
1.3 Το Διαδίκτυο σήμερα.....	6
1.4 Νομικά και ηθικά ζητήματα.....	7
1.5 eCommerce.....	7
1.5.1 Web development.....	8
1.5.2 Ηλεκτρονικά καταστήματα.....	9
1.5.3 Online Δημοπρασίες.....	10
1.5.4 eBay.....	11
Προέλευση και πρόωρη ιστορία.....	11
Στοιχεία και υπηρεσίες.....	12
2. Web Development.....	13
2.1 Ασφάλεια.....	14
2.2 Server side coding.....	14
2.2.1 PHP.....	15
2.3 Client side coding.....	17
2.4 AJAX.....	17
2.4.1 Request – Response web μοντέλο.....	17
2.4.2 AJAX (Javascript, CSS, DOM και XMLHttpRequest object).....	19
3. Σύστημα Ηλεκτρονικών Δημοπρασιών.....	23
3.1 Η αρχιτεκτονική του συστήματος.....	23
3.2 Η αρχιτεκτονική MVC (model – view – controller).....	24
3.2.1 Model (μοντέλο).....	26
3.2.2 View (παρουσίαση).....	27
3.2.3 Controller (ελεγκτής ή μεσολαβητής).....	27
3.3 Υλοποίηση της MVC στο Σύστημα Δημοπρασιών.....	28
3.4 Registry Pattern ή απλώς 'basket'.....	29
3.5 Σενάρια Χρήσης (use cases).....	30
3.6 Ανάλυση της Βάσης Δεδομένων.....	60
3.6.1 Λεπτομερή ανάλυση της βάσης δεδομένων.....	62
Πίνακας : user.....	62
Πίνακας : sms.....	63
Πίνακας : email.....	64
Πίνακας : privileges.....	64
Πίνακας : announcement.....	65
Πίνακας : personal_message.....	65
Πίνακας : auction.....	66
Πίνακας : rating.....	68
Πίνακας : image.....	69
Πίνακας : category.....	69
Πίνακας : favorites.....	70
Πίνακας : bid.....	70
Πίνακας : offer.....	71
3.7 Οδηγίες Εγκατάστασης Συστήματος.....	71
3.7.1 Απαιτήσεις Υλικού.....	72
3.7.1.1 Εξυπηρετητής.....	72
3.7.1.2 Πελάτης.....	74
3.7.2 Απαιτήσεις Λογισμικού.....	74

3.7.2.1 Εξυπηρετητής.....	74
3.7.2.3 Πελάτης.....	75
3.7.3 Εγκατάσταση και Ρύθμιση.....	75
Βήμα 1.....	76
Βήμα 2.....	76
Βήμα 3.....	76
includePath.php.....	76
eauctions-conf.php.....	77
MySmarty.php.....	78
3.8 Εγχειρίδιο Τελικού Χρήστη.....	80
3.8.1 Δημιουργία νέου λογαριασμού (Γίνε μέλος).....	82
3.8.2 Κατηγορίες και Δημοπρασίες.....	85
3.8.3 Δημιουργία νέας δημοπρασίας (Νέα δημοπρασία).....	91
3.8.4 Αγαπημένα.....	94
Παράρτημα Α: Βιβλιογραφία.....	96
Ευχαριστήρια.....	97

Εισαγωγή

Η πτυχιακή αυτή είχε ως σκοπό την ανάπτυξη ενός διαδικτυακού συστήματος ηλεκτρονικών δημοπρασιών (electronic auctions, e-auctions) με την χρήση τεχνολογιών του παγκοσμίου ιστού (web technologies). Είναι λογικό στο σημείο αυτό στο μυαλό του αναγνώστη αυτής της πτυχιακής να γίνουν συνειρμοί του τύπου <<άλλη μια πτυχιακή εργασία που ασχολείται με τον ευρύτερο τομέα του ηλεκτρονικού εμπορίου, γραμμένη σε μια server side γλώσσα προγραμματισμού>> ή να αναρωτιέται <<τι μπορεί να προσφέρει αυτή η πτυχιακή ώστε να γίνει ελκυστική στον αναγνώστη της>>. Μπορεί η ιδέα αλλά και το είδος της εφαρμογής να φαίνεται κοινότυπο, παρόλα αυτά ο βαθύτερος σκοπός της πτυχιακής αυτής δεν είναι να αναπτυχθεί απλώς άλλη μια εφαρμογή ηλεκτρονικού εμπορίου. Σκοπός της είναι να χρησιμοποιηθούν όλες οι νέες τεχνολογίες και καινοτομίες των τελευταίων 2 ετών που εμφανίστηκαν στον web προγραμματισμό και μέσω μιας εφαρμογής ηλεκτρονικών δημοπρασιών να δείξει τις δυνατότητές τους. Τεχνολογίες που έχουν την δυνατότητα να προσφέρουν σε μια web εφαρμογή νέες και πλούσιες σε δυνατότητες λειτουργίες, πιο εύχρηστο και διαδραστικό περιβάλλον εργασίας και φυσικά ταχύτητα απόκρισης σε τέτοιο επίπεδο που μπορούν πλέον να συγκριθούν άμεσα με αυτά των desktop εφαρμογών. Η ιδέα των ηλεκτρονικών δημοπρασιών επιλέχτηκε αυθόρμητα και ως ένα σημείο τυχαία. Για τις τεχνολογίες αυτές θα αναφερθούμε αναλυτικά μια προς μια σε επόμενα κεφάλαια. Με δύο λόγια όμως θα μπορούσαμε να πούμε ότι θα συζητήσουμε για όλες εκείνες τις νέες τεχνολογίες / καινοτομίες που απαρτίζουν την βάση για την μετάβαση στην άτυπη εποχή του web 2.0. Ενδεικτικά θα αναφέρουμε τα ονόματα μερικών εξ αυτών:

- AJAX (Asynchronous Javascript And Xml)
- XML (eXtended Markup Language)
- XSLT (Extensible Stylesheet Language Transformations)
- JSON (JavaScript Object Notation)

1. Διαδίκτυο

1.1 Ορισμός του Διαδικτύου

Στην γενική του έννοια, **διαδίκτυο** είναι ένα δίκτυο ηλεκτρονικών υπολογιστών που (δια)συνδέει άλλα δίκτυα. Ο αντίστοιχος αγγλικός όρος *internet* προκύπτει από τη σύνθεση των λέξεων *inter-network*.

1.2 Η γέννηση του Διαδικτύου

Τα θεμέλια του διαδικτύου τα έθεσε ο Βάνεβαρ Μπους (Vannevar Bush) όταν στο κείμενό του "As We May Think" αναφέρθηκε σε ένα "γαλαξιακό δίκτυο" συνδεδεμένων υπολογιστών. Ο πυρήνας του Διαδικτύου ξεκίνησε το 1969 με την ονομασία ARPANET στις Ηνωμένες Πολιτείες στην Υπηρεσία Προηγμένων Αμυντικών Ερευνών (Defense Advanced Research Projects Agency, *DARPA*) του υπουργείου Άμυνας των ΗΠΑ. Η αρχική έρευνα που συνέβαλε στο ARPANET περιελάμβανε εργασίες στα αποκεντρωμένα δίκτυα, queuing theory και ανταλλαγή πακέτων packet switching. Στις 11 Ιανουαρίου 1983 το ARPANET άλλαξε το βασικό του δικτυακό πρωτόκολλο επικοινωνίας από το NCP στο TCP/IP ξεκινώντας έτσι το Διαδίκτυο όπως το γνωρίζουμε σήμερα. Ένα σημαντικό βήμα στην ανάπτυξη έκανε το Εθνικό Ίδρυμα Επιστημών (National Science Foundation, *NSF*) των ΗΠΑ το οποίο έχτισε την πανεπιστημιακή ραχοκοκαλιά, το NSFNet, το 1986. Σημαντικά διαφορετικά δίκτυα που έχουν επιτυχώς ενσωματωθεί στο Διαδίκτυο είναι μεταξύ άλλων το Usenet, το Fidonet και το Bitnet. Στη δεκαετία του 1990 το Διαδίκτυο προσάρμοσε επιτυχώς την πλειοψηφία των παλιότερων δικτύων υπολογιστών. Αυτή η ανάπτυξη συχνά αποδίδεται στην έλλειψη κεντρικού ελέγχου, η οποία επιτρέπει οργανική ανάπτυξη του δικτύου, όπως και στο μη ιδιοκτησιακό καθεστώς των πρωτοκόλλων του Διαδικτύου, τα οποία ενθάρρυναν την ευκολία στην εσωτερική λειτουργία των μικροπωλητών και απέτρεψαν μία και μόνο εταιρεία να ασκεί την επιρροή της σε όλο το δίκτυο.

1.3 Το Διαδίκτυο σήμερα

Το Διαδίκτυο συγκροτείται από αμφί- ή πολύπλευρα εμπορικά συμβόλαια (π.χ. ομότιμες συμφωνίες) και από τεχνικές προδιαγραφές ή πρωτόκολλα που περιγράφουν την ανταλλαγή δεδομένων στο δίκτυο. Τα πρωτόκολλα αυτά μορφοποιούνται με συζητήσεις μέσα στο Internet Engineering Task Force (IETF) και τις ομάδες εργασίας του, οι οποίες είναι ανοιχτές για δημόσια συμμετοχή και κριτική. Αυτές οι επιτροπές παράγουν κείμενα που είναι γνωστά ως Αιτήματα για Σχολιασμό. Ορισμένα ΑΓΣ εγείρονται από την κατάσταση του Προτύπου Διαδικτύου από το Συμβούλιο Αρχιτεκτονικής του Διαδικτύου (IAB). Μερικά από τα πιο γνωστά πρωτόκολλα Internet protocol suite είναι το IP, TCP, το UDP, το DNS, το PPP, το SLIP, το ICMP, το POP3, IMAP, το SMTP, το HTTP, το HTTPS, το SSH, το Telnet, το FTP, το LDAP και το SSL. Μερικές από τις πιο γνωστές ιντερνετικές υπηρεσίες που χρησιμοποιούν αυτά τα πρωτόκολλα είναι το ηλεκτρονικό ταχυδρομείο (e-mail), τα Usenet newsgroups, η διαμοίραση αρχείων (file sharing), ο Παγκόσμιος Ιστός (World Wide Web), το Gopher, το session access, το WAIS, το finger, το IRC, το MUD και το MUSH. Από αυτές, το ηλεκτρονικό ταχυδρομείο και ο Παγκόσμιος Ιστός είναι οι πιο ευρέως χρησιμοποιούμενες, ενώ πολλές άλλες υπηρεσίες έχουν βασιστεί πάνω σε αυτές, όπως ταχυδρομικές λίστες και αρχεία καταγραφής ιστού. Το Διαδίκτυο καθιστά δυνατή τη διάθεση υπηρεσιών σε πραγματικό χρόνο όπως ραδιόφωνο μέσω Ιστού και προβλέψεις μέσω Ιστού που μπορούν να προσπελαστούν από οπουδήποτε στον κόσμο. Μερικές ακόμα γνωστές υπηρεσίες του Διαδικτύου δεν δημιουργήθηκαν με αυτόν τον τρόπο, αλλά βασίστηκαν αρχικά σε ιδιόκτητα συστήματα. Αυτές περιλαμβάνουν το ICQ, το AIM, το CDDDB και το Gnutella. Υπάρχουν πολλές αναλύσεις για το Διαδίκτυο και τη δομή του. Για παράδειγμα, είναι καθορισμένο ότι η δομή δρομολόγησης του Διαδικτύου και οι υπερσύνδεσμοι του Παγκόσμιου Ιστού είναι παραδείγματα μη κλιμακούμενων δικτύων.

1.4 Νομικά και ηθικά ζητήματα

Υπάρχει μια γενική ανησυχία αναφορικά με το Διαδίκτυο που πηγάζει από μέρος του περιεχομένου του που είναι αρκετά αμφισβητήσιμο. Η παραβίαση πνευματικών δικαιωμάτων, η πορνογραφία, η ψευδοπροσωπία, η προσφορά παρανόμων προϊόντων είναι φαινόμενα παρόντα και

εύκολα προσβάσιμα στο Διαδίκτυο, και ο περιορισμός τους είναι ιδιαίτερα δύσκολος. Για παράδειγμα, η λέξη "sex" παραμένει μία από τις λέξεις που χρησιμοποιούνται περισσότερο για αναζήτηση στο Διαδίκτυο μέσω μηχανών αναζήτησης. Συχνά, η ανησυχία αυτή, που θεωρείται από κάποιους αβάσιμη, έχει ως αποτέλεσμα τον ηθικό πανικό, όπως έχει σημειωθεί σε περιπτώσεις όπου εγκλήματα ή αποτρόπαιες καταστάσεις έρχονται σε επαφή με την κοινή γνώμη (συνήθως περιπτώσεις παιδεραστίας, ιστοσελίδες σατανιστών κ.ά.).

1.5 eCommerce

Η έννοια του όρου «ηλεκτρονικό εμπόριο» έχει αλλάξει κατά τη διάρκεια των τελευταίων 30 ετών. Αρχικά, το «ηλεκτρονικό εμπόριο» σήμανε τη διευκόλυνση των εμπορικών συναλλαγών ηλεκτρονικά, συνήθως χρησιμοποιώντας την τεχνολογία όπως την ηλεκτρονική ανταλλαγή δεδομένων (EDI) και την ηλεκτρονική μεταφορά Ταμείων (EFT), όπου και τα δύο εισήχθησαν προς το τέλος της δεκαετίας του '70. Παραδείγματος χάριν, για να στείλουν τα εμπορικά έγγραφα, όπως τις εντολές αγοράς ή τα τιμολόγια ηλεκτρονικά. «Ο ηλεκτρονικός» ή «το ε» στο ηλεκτρονικό εμπόριο αναφέρεται στην τεχνολογία, το «εμπόριο» αναφέρεται στα παραδοσιακά επιχειρησιακά πρότυπα. Το ηλεκτρονικό εμπόριο είναι το πλήρες σύνολο διαδικασιών που υποστηρίζουν τις εμπορικές επιχειρησιακές δραστηριότητες σε ένα δίκτυο. Στη δεκαετία του '70 και τη δεκαετία του '80, αυτό επίσης περιελάμβανε την ανάλυση πληροφοριών. Η αύξηση και η αποδοχή των πιστωτικών καρτών, των μηχανών ATM και των τηλεφωνικών τραπεζικών εργασιών στη δεκαετία του '80 ήταν επίσης μορφές ηλεκτρονικού εμπορίου. Εντούτοις, από τη δεκαετία του '90 και μετά, αυτό περιελάμβανε τα συστήματα προγραμματισμού των επιχειρηματικών πόρων (cEpr) και την αποθήκευση στοιχείων. Ίσως το πιο πρόωρο παράδειγμα του πολλαπλού ηλεκτρονικού εμπορίου στα φυσικά αγαθά ήταν η ανταλλαγή υπολογιστών της Βοστώνης, μια αγορά για τους χρησιμοποιημένους υπολογιστές, που προωθήθηκαν το 1982. Στην εποχή COM, η αγορά των αγαθών και των υπηρεσιών πέρα από το World Wide Web, συνήθως με τις ασφαλείς συνδέσεις (HTTPS, ένα πρόσθετο πρωτόκολλο κεντρικών υπολογιστών που κρυπτογραφεί τα εμπιστευτικά στοιχεία για την προστασία των πελατών) των ηλεκτρονικών αγορών και με τις ηλεκτρονικές υπηρεσίες πληρωμής, όπως τις εγκρίσεις πληρωμής με πιστωτική κάρτα. Σήμερα, καλύπτει ένα πολύ ευρύ φάσμα των επιχειρησιακών δραστηριοτήτων και των διαδικασιών, από την ηλεκτρονική

κατάθεση. Η συνεχώς αναπτυσσόμενη εξάρτηση των σύγχρονων βιομηχανιών στις ηλεκτρονικά επιχειρησιακές διαδικασίες έδωσε την ώθηση στην αύξηση και την εξέλιξη της υποστήριξης των συστημάτων, συμπεριλαμβανομένων των εφαρμογών και του υλικολογισμικού. Παραδείγματα είναι ευρυζωνικά και οπτικών ινών δίκτυα, συστήματα ελέγχου καταλόγων και λογισμικό οικονομικής λογιστικής.

Τεχνικές και οργανωτικές πτυχές

1. Ικανοποιητική εργασία που γίνεται στην έρευνα και στην ανάλυση αγοράς.
2. Μια καλή διοικητική ομάδα.
3. Παροχή ενός εύκολου και εξασφαλισμένου τρόπου για τους πελάτες να εκτελεστούν οι συναλλαγές.
4. Παροχή της αξιοπιστίας και της ασφάλειας.
5. Κατασκευή ενός εμπορικά υγιούς επιχειρησιακού προτύπου.
6. Παροχή ενός ελκυστικού διαδικτυακού χώρου.
7. Επιχειρησιακές διαδικασίες βελτιώσεων.

Προβλήματα

1. Αποτυχία στο να γίνουν κατανοητοί οι όροι χρήσης από τους πελάτες.
2. Ανικανότητα να προβλεφθεί η αντίδραση των ανταγωνιστών.
3. Υπερεκτίμηση της ικανότητας των διαθέσιμων πόρων.
4. Αποτυχία στον συντονισμό της επιχείρησης.
5. Υποτίμηση των χρονικών απαιτήσεων.

1.5.1 Web development

Όταν το web έγινε αρχικά γνωστό μεταξύ του ευρέως κοινού το 1994, πολλοί δημοσιογράφοι και αυθεντίες προέβλεπαν ότι το ηλεκτρονικό εμπόριο θα γινόταν σύντομα ένας σημαντικός οικονομικός τομέας. Εντούτοις, διάρκεσε περίπου τέσσερα έτη για τα πρωτόκολλα ασφάλειας (όπως HTTPS) για να γίνει αρκετά αναπτυγμένο και ευρέως επεκτειμένο. Στη συνέχεια, μεταξύ 1998 και 2000, ένας ουσιαστικός αριθμός επιχειρήσεων στις Ηνωμένες Πολιτείες και τη δυτική Ευρώπη ανέπτυξε τους στοιχειώδεις ιστοχώρους. Αν και ένας μεγάλος αριθμός επιχειρήσεων

«καθαρού ηλεκτρονικού εμπορίου» εξαφανίστηκε το 2000 και το 2001, πολλοί πωλητές αναγνώρισαν ότι τέτοιες επιχειρήσεις είχαν προσδιορίσει τις πολύτιμες αγορές θέσεων και άρχισαν να προσθέτουν τις ικανότητες ηλεκτρονικού εμπορίου στους ιστοχώρους τους. Παραδείγματος χάριν, μετά από την κατάρρευση δύο παραδοσιακών αλυσίδων υπεραγορών, Albertsons και Safeway, και οι δύο άρχισαν τα υποκαταστήματα ηλεκτρονικού εμπορίου μέσω των οποίων οι καταναλωτές θα μπορούσαν να διατάξουν τα παντοπωλεία on-line. Η εμφάνιση του ηλεκτρονικού εμπορίου επίσης, μείωσε σημαντικά τα εμπόδια στην είσοδο στην πώληση πολλών τύπων αγαθών. Συχνά, οι μικροί πωλητές χρησιμοποιούν τις απευθείας περιοχές δημοπρασίας όπως το eBay, ή πωλούν μέσω των μεγάλων εταιρικών ιστοχώρων όπως το Amazon.com, προκειμένου να εκμεταλλευτούν την ευκολία έκθεσης και οργάνωσης.

1.5.2 Ηλεκτρονικά καταστήματα

Η Πληροφορική Επανάσταση άλλαξε ριζικά τον τρόπο ζωής των πολιτών, επέφερε μια σειρά αλλαγών που επηρεάζουν και τις επιχειρήσεις. Αυτές οι αλλαγές μπορούν να αποτελέσουν σημαντικό όπλο στα χέρια των επιχειρήσεων που θέλουν να ανταποκριθούν στις απαιτήσεις ενός νέου περιβάλλοντος που χαρακτηρίζεται από συνεχώς μεταβαλλόμενες συνθήκες, διεθνοποίηση και εντατικοποίηση του ανταγωνισμού, κ.α. Οι επιχειρήσεις που θα επιβιώσουν είναι αυτές που στον παρόντα χρόνο θα κάνουν τις στρατηγικές επιλογές για την ενσωμάτωση των νέων τεχνολογιών στις πρακτικές τους. Η ανάπτυξη της Ελληνικής αγοράς Internet χρονολογείται από τα μέσα της προηγούμενης δεκαετίας και έχει διαφορά φάσης περίπου 5 χρόνων από τις αναπτυγμένες διεθνείς αγορές. Σύμφωνα με την Κλαδική Μελέτη «Υπηρεσίες Internet», για την Ελλάδα, ICAP, Νοέμβριος 2004: το ηλεκτρονικό επιχειρείν και εμπόριο βρίσκεται ακόμα σε νηπιακό στάδιο και σε τούτο συμβάλλει και η έλλειψη παράδοσης στις συναλλαγές εξ αποστάσεως. 23% των επιχειρήσεων έχει παρουσία στο Internet το 2004. Οι επιχειρήσεις δεν έχουν συνειδητοποιήσει ακόμα τις δυνατότητες που απορρέουν από την εφαρμογή μεθόδων και τεχνικών ηλεκτρονικού επιχειρείν. Όπως, χαρακτηριστικά έχει πει ο Bill Gates: "Το Internet δεν είναι απλώς άλλο ένα κανάλι πωλήσεων. Θα μετασχηματίσει την επιχείρησή σας. Η μελλοντική επιχείρηση θα λειτουργεί με ένα ψηφιακό νευρικό σύστημα". Με λίγα λόγια : Πωλήσεις σε οποιονδήποτε, οπουδήποτε, οποτεδήποτε!!! Έτσι αναπτύχθηκαν τα ηλεκτρονικά καταστήματα που προσφέρουν ημερησίως

χιλιάδες προϊόντα σε τιμές πολύ λογικές και μερικές φορές πολύ χαμηλές. Στην Ελλάδα μόνο τα τελευταία 5 χρόνια δραστηριοποιούνται πάνω από 100 επιχειρήσεις που ανήκουν σε αυτό το είδος. Οι πιο γνωστές είναι το technixx.gr και το eshop.gr.

1.5.3 Online Δημοπρασίες

Οι online δημοπρασίες είναι ένα πρότυπο στο οποίο οι συμμετέχοντες προσφέρουν χρήματα για τα προϊόντα και τις υπηρεσίες για το Διαδίκτυο. Η λειτουργία της αγοράς και της πώλησης με ένα σχήμα δημοπρασίας γίνεται μέσω του λογισμικού δημοπρασίας που ρυθμίζει τις διάφορες σχετικές διαδικασίες.

Τα στρατηγικά πλεονεκτήματα αυτού του επιχειρησιακού προτύπου περιλαμβάνουν:

1. Κανένας χρονικός περιορισμός. Οι προσφορές μπορούν να τοποθετηθούν οποιαδήποτε στιγμή. Τα στοιχεία παρατίθενται για μερικές ημέρες κατά την κρίση του πωλητή, δίνοντας το χρόνο στους αγοραστές να ψάξουν και να αποφασίσουν για την αγορά. Αυτή η ευκολία αυξάνει τον αριθμό των πλειοδοτών.
2. Κανένας γεωγραφικός περιορισμός. Οι πωλητές και οι πλειοδότες μπορούν να συμμετέχουν από οπουδήποτε έχοντας πρόσβαση στο Διαδίκτυο. Αυτό τους καθιστά πιο προσιτούς και μειώνει το κόστος σε μια δημοπρασία.
3. Ένταση των κοινωνικών αλληλεπιδράσεων. Οι κοινωνικές αλληλεπιδράσεις που περιλαμβάνονται στη διαδικασία προσφοράς είναι πολύ παρόμοιες με τις πραγματικές. Οι πλειοδότες περιμένουν στην αναμονή ελπίζοντας ότι θα κερδίσουν.
4. Μεγάλος αριθμός πλειοδοτών. Λόγω της δυνατότητας για μια σχετικά χαμηλή τιμή, του εύρους των προϊόντων και των διαθέσιμων υπηρεσιών, της ευκολίας στη πρόσβαση και των κοινωνικών παροχών της διαδικασίας, υπάρχει μεγάλος αριθμός πλειοδοτών.

Το eBay, το μεγαλύτερο site online δημοπρασιών, είναι ένα από τα καλύτερα γνωστά παραδείγματα. Ενεργεί ως αγορά για τα άτομα και τις επιχειρήσεις που χρησιμοποιούν την περιοχή που δημοπρατεί τα αγαθά και τις υπηρεσίες. Σχεδόν όλες οι online δημοπρασίες χρησιμοποιούν την αγγλική μέθοδο δημοπρασίας. Δηλαδή, το προϊόν ξεκινάει από μια αρχική τιμή και οι πλειοδότες αυξάνουν τις προσφορές τους με κάποιο βήμα αύξησης. Νικητής είναι αυτός που θα προσφέρει τα

περισσότερα χρήματα.

1.5.4 eBay

Προέλευση και πρόωρη ιστορία

Ο ιστοχώρος online δημοπρασιών ιδρύθηκε στο San Jose, στην Καλιφόρνια στις 3 Σεπτεμβρίου 1995 από τον προγραμματιστή υπολογιστών Pierre Omidyar ως Auction Web. Το πρώτο στοιχείο που πωλήθηκε στο eBay ήταν ένας σπασμένος δείκτης λέιζερ για \$14.83. Έκπληκτος, ο Omidyar ήρθε σε επαφή με τον πλειοδότη που κέρδισε και τον ρώτησε εάν κατάλαβε ότι ο δείκτης λέιζερ ήταν σπασμένος. Στο email που του απάντησε, ο αγοραστής εξήγησε: «Είμαι συλλέκτης των σπασμένων δεικτών λέιζερ». Η ιστορία πως το eBay ιδρύθηκε για να βοηθήσει τους διανομείς εμπορικών PEZ καραμελών, κατασκευάστηκε από έναν δημόσιο διευθυντή σχέσεων το 1997 για να προσελκύσει τα μέσα μαζικής ενημέρωσης. Αυτό αποκαλύφθηκε στο βιβλίο του Adam Cohen's, που εκδόθηκε το 2002, και επιβεβαιώθηκε από το eBay. Ο Chris Agarpao μισθώθηκε ως ο πρώτος υπάλληλος του eBay και ο Jeff Skoll μισθώθηκε ως ο πρώτος Πρόεδρος της επιχείρησης το 1996. Το Νοέμβριο του 1996, το eBay εισήγαγε την πρώτη διαπραγμάτευση χορήγησης αδειών, για να πωλήσει τα εισιτήρια αεροπλάνων και άλλα προϊόντα ταξιδιού. Η επιχείρηση άλλαξε επίσημα το όνομα της υπηρεσίας της από Auction Web σε eBay το Σεπτέμβριο του 1997. Αρχικά, η περιοχή άνηκε στην ομάδα Echo Bay Technology Group, η οποία ήταν συμβουλευτική εταιρία του Omidyar. Ο Omidyar είχε προσπαθήσει να καταχωρήσει το όνομα echobay.com, αλλά μια επιχείρηση εξόρυξης χρυσού είχε ήδη καταλάβει αυτό το όνομα. Έτσι επέλεξε τη δεύτερη επιλογή του, eBay.com. Το eBay έγινε ευρέως γνωστό το 1998, και οι Omidyar και Skoll έγιναν δισεκατομμυριούχοι.

Στοιχεία και υπηρεσίες

Τα εκατομμύρια των συσκευών, των υπολογιστών, των επίπλων, του εξοπλισμού, των οχημάτων, και άλλων διάφορων στοιχείων είναι απαριθμημένα, αγορασμένα, και πωλούνται

καθημερινά. Το 2005, το eBay προώθησε την επιχείρηση & τη βιομηχανική κατηγορία του, που κερδίζουν πάρα πολύ στη βιομηχανική επιχείρηση. Μερικά στοιχεία είναι σπάνια και πολύτιμα, ενώ πολλά άλλα είναι σκονισμένα gizmos που θα είχαν απορριφθεί εάν δεν υπήρχαν χιλιάδες πρόθυμοι πλειοδότες παγκοσμίως. Οτιδήποτε μπορεί να πωληθεί εφ' όσον δεν είναι παράνομο ή δεν παραβιάζει την πολιτική του eBay. Υπηρεσίες και άυλα αγαθά μπορούν να πωληθούν επίσης. Οι μεγάλες διεθνείς επιχειρήσεις, όπως η IBM, πωλούν τα νεώτερα προϊόντα τους και προσφέρουν τις υπηρεσίες τους στο eBay χρησιμοποιώντας ανταγωνιστικές δημοπρασίες. Οι αναζητήσεις της βάσης δεδομένων καθιστούν τη πλοήγηση ελαφρώς γρηγορότερη και φτηνότερη. Οι χωριστές περιοχές του eBay, όπως το eBay ΗΠΑ και το eBay UK επιτρέπουν στους χρήστες να κάνουν συναλλαγές χρησιμοποιώντας το τοπικό νόμισμα ως πρόσθετη επιλογή σε PayPal. Τον Ιούνιο του 2005, υπήρξαν πάνω από 15.000 μέλη στο πρόγραμμα του eBay που είναι υπεύθυνο για την ανάπτυξη, που περιλαμβάνει μια ευρεία σειρά επιχειρήσεων που δημιουργούν τις εφαρμογές λογισμικού για να υποστηριχθούν οι αγοραστές και οι πωλητές καθώς επίσης και οι θυγατρικές εταιρείες του eBay. Γενικά, η επιχείρηση αφαιρεί τις δημοπρασίες που παραβιάζουν τους όρους της συμφωνίας υπηρεσιών μέσα σε έναν σύντομο χρόνο. Είχε προκύψει διαμάχη με την δημοπρασία ενός ανθρώπου, ο οποίος είχε βάλει ως προϊόν το νεφρό του.

2. Web Development

Το web development είναι ένας ευρύς όρος για οποιεσδήποτε δραστηριότητες σχετικές με την ανάπτυξη ενός ιστοχώρου για το World Wide Web ή ένα ενδοδίκτυο. Αυτό μπορεί να περιλαμβάνει την επιχειρησιακή ανάπτυξη του ηλεκτρονικού εμπορίου, την client – side / server – side κωδικοποίηση και την διαμόρφωση κεντρικών υπολογιστών δικτύου. Το web development μπορεί να κυμανθεί από την ανάπτυξη της απλούστερης στατικής ενιαίας σελίδας, στις πιο σύνθετες εφαρμογές Διαδικτύου, τις ηλεκτρονικές επιχειρήσεις ή τις κοινωνικές υπηρεσίες δικτύων. Από τα μέσα της δεκαετίας του 90, το web development είναι μια από τις ταχύτερες αναπτυσσόμενες βιομηχανίες στον κόσμο. Το 1995 υπήρξαν λιγότερες από 10000 επιχειρήσεις web development στις ΗΠΑ. Το 2005 υπήρχαν πάνω από 30000 τέτοιες επιχειρήσεις. Η βιομηχανία web development αναμένεται να αυξηθεί πάνω από 20% μέχρι το 2010. Η αύξηση αυτή της βιομηχανίας ωθείται από τις μεγάλες επιχειρήσεις, οι οποίες επιθυμούν να πωλούν τα προϊόντα τους και τις υπηρεσίες τους στους πελάτες τους και να αυτοματοποιήσουν την επιχειρησιακή ροή της δουλειάς, καθώς επίσης και την αύξηση πολλών μικρών επιχειρήσεων. Χρησιμοποιώντας τέτοιο λογισμικό, ουσιαστικά καθένας μπορεί να αναπτύξει ιστοσελίδα σε θέμα λεπτών. Η γνώση γλώσσας σήμανσης υπερκειμένων (HTML), ή άλλες γλώσσες προγραμματισμού δεν απαιτείται, αλλά συστήνεται για επαγγελματικά αποτελέσματα. Αντί του τρεξίματος του εκτελέσιμου κώδικα σε έναν τοπικό υπολογιστή, οι χρήστες αλληλεπιδρούν με εφαρμογές ανοιχτής γραμμής για να δημιουργηθεί το νέο περιεχόμενο. Αυτό έχει δημιουργήσει τις νέες μεθόδους στην επικοινωνία και έχει επιτρέψει πολλές ευκαιρίες να αποκεντρωθεί η πληροφόρηση και η διανομή των μέσων ενημέρωσης. Οι χρήστες είναι τώρα ικανοί να αλληλεπιδρούν με τις εφαρμογές από πολλές θέσεις, αντί της σύνδεσης σε έναν συγκεκριμένο τερματικό σταθμό για το περιβάλλον εφαρμογής τους. Το web development μπορεί να χωριστεί σε δύο κατηγορίες:

- Client side κωδικοποίηση
 - CSS
 - XHTML

- Javascript
- DOM
- Server Side κωδικοποίηση
 - PHP
 - ASP
 - ASP.net
 - JSP
 - CGI και PERL

Ο ανωτέρω κατάλογος είναι μια απλή ιεραρχία ανάπτυξης ιστοχώρου και μπορεί να επεκταθεί. Εξετάζοντας αυτά τα στοιχεία, η server – side κωδικοποίηση όπως η XHTML εκτελείται και αποθηκεύεται σε ένα τοπικό client ενώ ο server δεν είναι διαθέσιμος στον client και εκτελείται σε ένα κεντρικό υπολογιστή δικτύου, ο οποίος παράγει το κατάλληλο XHTML που επιστρέφεται έπειτα στον client.

2.1 Ασφάλεια

Το web development λαμβάνει υπόψη πολλά πράγματα, όπως ο έλεγχος λάθους εισαγωγών δεδομένων μέσω των μορφών, καθώς επίσης και την εγκυρότητα του στοιχείου που εισάγεται σε εκείνους τους τομείς. Επίσης, τα scripts που τρέχουν μπορεί να χρησιμοποιηθούν από τους χάκερ, ώστε να υποκλέψουν πληροφορίες κέρδους, όπως οι διευθύνσεις ηλεκτρονικού ταχυδρομείου, οι κωδικοί πρόσβασης και το προστατευόμενο περιεχόμενο.

2.2 Server side coding

Η server – side κωδικοποίηση είναι μια τεχνολογία κεντρικών υπολογιστών δικτύου στην οποία το αίτημα του χρήστη εκπληρώνεται με το να τρέξει ένα script άμεσα στον κεντρικό υπολογιστή δικτύου, για να παράγει τις δυναμικές σελίδες HTML. Χρησιμοποιείται συνήθως για να παρέχει τους διαλογικούς χώρους που διασυνδέουν τις βάσεις δεδομένων με άλλα στοιχεία. Το αρχικό πλεονέκτημα είναι η δυνατότητα να προσαρμοστεί η απάντηση του χρήστη με βάση τις απαιτήσεις του, τα δικαιώματα πρόσβασης, ή τις ερωτήσεις του στην «αποθήκη» των στοιχείων.

2.2.1 PHP

Η PHP (το όνομα PHP προέρχεται από το αναδρομικό ακρωνύμιο PHP Hypertext Preprocessor) είναι μια open source γενικού σκοπού scripting γλώσσα προγραμματισμού που μπορεί να βοηθήσει πάρα πολύ στην ανάπτυξη web εφαρμογών. Το πλεονέκτημα της PHP έναντι άλλων τεχνολογιών ανάπτυξης web εφαρμογών είναι η δυνατότητα του να εισαχθεί μέσα στην HTML. Αντί δηλαδή να γράφεις μια εφαρμογή που θα εκτυπώνει μια ολόκληρη σελίδα σε HTML, μπορείς να γράφεις την HTML και στο σημείο που θέλει να εμφανιστεί κάτι δυναμικά να εισάγεις την PHP. Παρακάτω βλέπουμε ένα παράδειγμα.

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      echo "Hi, I'm a PHP script!";
    ?>
  </body>
</html>
```

Στο παραπάνω κομμάτι κώδικα παρατηρούμε πέραν της HTML ένα σημείο το οποίο ξεκινά με την δήλωση:

```
<?php
```

Ακολουθεί η φράση:

```
echo "Hi, I'm a PHP script";
```

Και τέλος φαίνεται να τελειώνει αυτό το περίεργο κομμάτι με την δήλωση:

```
?>
```

Αυτή ακριβώς είναι και η PHP. Όπως αναφέραμε παραπάνω, κώδικας της PHP μπορεί να εισαχθεί οπουδήποτε μέσα σε μια σελίδα HTML. Οτιδήποτε υπάρχει ενδιάμεσα στις δηλώσεις <?php και ?> θεωρείται κώδικας της PHP και θα εκτελεστεί στην web εξυπηρετητή προτού επιστρέψει η σελίδα στον χρήστη που έκανε την αίτηση. Κάθε σελίδα HTML μπορεί να έχει όσα κομμάτια με κώδικα PHP θέλει και χρειάζεται. Όλα αυτά θα εκτελεστούν, τα αποτελέσματα τους θα μουν στο

σημείο εκεί ακριβώς που βρίσκονται και το τελικό HTML έγγραφο θα ταξιδέψει για τον χρήστη. Έτσι για παράδειγμα όταν κάποιος χρήστης αιτήσει το παραπάνω HTML έγγραφο του παραδείγματός μας, αυτό που θα πάρει πίσω θα είναι το έγγραφο έτσι όπως το βλέπει, πλην όμως το κομμάτι του κώδικα της PHP το οποίο θα εκτελεστεί και στο σημείο αυτό θα μείνει μόνο το “Hi, I'm a PHP script” γιατί αυτό του προστάζει να κάνει η εντολή echo της PHP.

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    Hi, I'm a PHP script
  </body>
</html>
```

Παρατηρούμε από το παραπάνω παράδειγμα πως με την χρήση της PHP μπορούμε να δώσουμε ζωή στην στατική HTML. Είναι γνωστό πως η HTML δεν προσφέρεται για την κατασκευή δυναμικού περιεχομένου μιας και στην ουσία η HTML δεν είναι τίποτα παραπάνω παρά μορφοποίηση περιεχομένου για την προβολή του μέσα από φυλλομετρητές. Η στατική φύση της HTML ήταν και είναι βασικό εμπόδιο για την χρήση του web ως πλατφόρμα για την προσφορά υπηρεσιών. Με το παράδειγμα όμως παραπάνω είδαμε πως με την χρήση της scripting γλώσσας PHP μπορούμε να αλλάζουμε τα περιεχόμενα της HTML και τελικά να επιστρέφουμε στον αιτών μια άλλη έκδοση της. Επεκτείνοντας την ιδέα αυτή μπορούμε να καταλήξουμε στο συμπέρασμα πως η PHP μπορεί να χρησιμοποιηθεί για την κατασκευή ολόκληρων εφαρμογών μιας και μπορούμε να κατασκευάζουμε την HTML δυναμικά και σε πραγματικό χρόνο με αυτήν και φυσικά κάθε φορά να επιστρέφεται διαφορετικό περιεχόμενα βάση της ταυτότητας του χρήστη. Σίγουρα η ιδέα αυτή περιέχει πολλές δυσκολίες που πρέπει να ξεπεραστούν, δυσκολίες που προέρχονται από την φύση του διαδικτύου μιας και αυτό κατασκευάστηκε για την προσφορά στατικού περιεχομένου, παρόλα αυτά με την χρήση διαφόρων τεχνικών αυτά ξεπερνιούνται. Θα ήταν χρήσιμο να αναφέρουμε πως η PHP μας δίνει τη δυνατότητα να συνδεθούμε με ένα μεγάλο πλήθος συστημάτων διαχείρισης βάσεων δεδομένων. Με αυτό και μόνο φανταστείτε τι προοπτικές ανοίγονται αν συνδυάσουμε την δύναμη της HTML να μορφοποιεί δεδομένα μαζί με την PHP που μπορεί να συνδεθεί σε βάσης δεδομένων και φυσικά να παράγει HTML σε πραγματικό χρόνο.

2.3 Client side coding

Η client side κωδικοποίηση αναφέρεται στις διαδικασίες που εκτελούνται από τον πελάτη σε μια σχέση κεντρικών υπολογιστών. Χαρακτηριστικά, ένας client είναι μια εφαρμογή, όπως μια μηχανή αναζήτησης, η οποία τρέχει στον τοπικό υπολογιστή ή τον τερματικό σταθμό ενός χρήστη και τον συνδέει με έναν κεντρικό υπολογιστή ανάλογα με τις ανάγκες του. Αν οι διαδικασίες μπορούν να εκτελεστούν από τον πελάτη χωρίς αποστολή των στοιχείων του πέρα από το δίκτυο, μπορούν να πάρουν λιγότερο χρόνο, να χρησιμοποιήσουν λιγότερο εύρος ζώνης και να υποστούν μικρότερο κίνδυνο ασφάλειας. Όταν ο κεντρικός υπολογιστής εξυπηρετεί τα στοιχεία με έναν συνήθως τρόπο, παραδείγματος χάριν, σύμφωνα με τα πρωτόκολλα HTTP ή FTP, οι χρήστες μπορούν να έχουν την επιλογή των διαφόρων client – programs. Στην περίπτωση των πιο εξειδικευμένων εφαρμογών, οι προγραμματιστές μπορούν να γράψουν τον κώδικα για τον κεντρικό υπολογιστή, τον client και το πρωτόκολλα επικοινωνίας τους.

2.4 AJAX

2.4.1 Request – Response web μοντέλο

Όπως είδαμε στην ενότητα Server side coding, έχουν αναπτυχθεί τεχνολογίες που μας επιτρέπουν να χρησιμοποιήσουμε το web και όλα τα παρελκόμενα πρωτόκολλά του ως την πλατφόρμα για την ανάπτυξη εφαρμογών. Εφαρμογών φυσικά που θα εκτελούνται μέσα στο περιβάλλον ενός φυλλομετρητή. Αυτή η δυνατότητα “αγαπήθηκε” πάρα πολύ από τους προγραμματιστές μιας και προσφερόταν για αυτό που ονομάζουν οι τελευταίοι “write once, run anywhere”. Οι προγραμματιστές μπορούσαν δηλαδή να γράψουν μια φορά τον κώδικα της εφαρμογής και από εκεί και πέρα δεν χρειαζόταν τίποτα άλλο παρά ένας σύγχρονος φυλλομετρητής από την πλευρά του χρήστη για να την χρησιμοποιήσει. Βέβαια η φύση του web παρουσιάζει πολλά προβλήματα όταν το τελευταίο χρησιμοποιείται για την κατασκευή εφαρμογών. Το web σχεδιάστηκε έτσι ώστε να παρέχει στους χρήστες του στατικό περιεχόμενο. Αυτό το εμπόδιο ξεπεράστηκε με διάφορες τεχνολογίες που αναπτύχθηκαν, όπως η PHP που αναφέραμε παραπάνω.

Οι τεχνολογίες αυτές δίνουν τη δυνατότητα σε έναν web εξυπηρετητή να παράγει το περιεχόμενο που θα επιστρέψει πίσω στον χρήστη εκείνη τη στιγμή (real time). Έτσι μπορεί να προσφερθεί διαφορετικό περιεχόμενο σε κάθε χρήστη βάση των προτιμήσεών του και φυσικά της ηλεκτρονικής του ταυτότητας. Ένα δεύτερο και πολύ σημαντικό πρόβλημα του web στην κατασκευή εφαρμογών είναι το request – response μοντέλο το οποίο με την σειρά του εισάγει πλειάδα από δυσκολίες που πρέπει να ξεπεραστούν. Βάση αυτού του μοντέλου ένας φυλλομετρητής κάνει μια νέα αίτηση σε έναν web εξυπηρετητή όταν ο χρήστης προκαλέσει κάποιο event όπως το να κάνει κλικ πάνω σε ένα κουμπί ή σε ένα υπερδεσμό. Ο φυλλομετρητής αδειάζει τα περιεχόμενά του, κάνει την αίτηση για το νέο περιεχόμενο από τον web εξυπηρετητή, περιμένει μέχρις ότου το νέο περιεχόμενο ταξιδέψει πίσω και τελικά το παρουσιάζει στον χρήστη. Η επαναλαμβανόμενη αυτή διαδικασία μπορεί να προκαλέσει πολύ μεγάλο εκνευρισμό στον χρήστη, ειδικότερα αν μιλάμε για εφαρμογή. Ο χρόνος που μεσολαβεί από την ώρα που ο χρήστης προσπαθεί να εκτελέσει μια λειτουργία μέχρι την στιγμή που επιστρέφονται τα αποτελέσματα πίσω σε αυτόν μπορεί να είναι σημαντικά μεγάλος αποσπώντας την προσοχή του χρήστη. Επίσης ο χρόνος αυτός του μεσοδιαστήματος είναι μη παραγωγικός μιας και ο χρήστης απλώς βλέπει έναν άδειο ή λευκό φυλλομετρητή που περιμένει το νέο περιεχόμενο χωρίς να μπορεί να κάνει τίποτα. Για παράδειγμα δεν υπάρχει η δυνατότητα ακύρωσης μιας λειτουργίας σε περίπτωση που αλλάξει ο χρήστης γνώμη. Πρέπει πρώτα να περιμένει για τα αποτελέσματα και μετά φυσικά να ακυρώσει την προηγούμενη ενέργεια. Η αδυναμία αυτή είναι φυσικά αποτέλεσμα του request – response μοντέλου με το οποίο δουλεύει το web. Το μοντέλο αυτό τέλος δεν επιτρέπει στους προγραμματιστές και φυσικά κατά επέκταση στην εφαρμογή να διατηρεί σε ποια κατάσταση είναι ο εκάστοτε χρήστης. Επειδή κάθε request του φυλλομετρητή δεν έχει καμία σχέση με τα προηγούμενα request που είχε κάνει στον εξυπηρετητή της εφαρμογής, ο τελευταίος δεν μπορεί να γνωρίζει την κατάσταση του χρήστη ανά πάσα στιγμή και για αυτό το λόγο έχουν επινοηθεί διάφορες τεχνικές για να ξεπεραστεί και η αδυναμία αυτή.

Στις τεχνικές αυτές περιλαμβάνονται τα:

- cookies
- server sessions

Παρόλα αυτά το μεσοδιάστημα χρόνου που υπάρχει ανάμεσα σε μια ενέργεια του χρήστη και στην απάντηση του εξυπηρετητή εξακολουθεί να είναι ένα από τα πιο σοβαρά προβλήματα που υπάρχουν στις web εφαρμογές. Η κατάσταση αυτή αφαιρεί από αυτού του είδους τις εφαρμογές τις

ταχύτητες, την αμεσότητα αλλά και την πλούσια παροχή λειτουργιών που απολαμβάνουν οι κλασικές desktop εφαρμογές. Την κατάσταση αυτή όμως ήρθε για να αλλάξει το AJAX.

2.4.2 AJAX (Javascript, CSS, DOM και XMLHttpRequest object)

Όπως αναφέραμε και πριν το AJAX ήρθε για να δώσει μια λύση στα προβλήματα που προκαλεί το request – response μοντέλο του web. Τι είναι όμως το AJAX και πως μπορεί κάποιος να το χρησιμοποιήσει. Ας αρχίσουμε λίγο ομαλά και ας αναφέρουμε πως το AJAX δεν είναι κάτι εντελώς καινούριο, πλην ενός στοιχείου του. Το AJAX είναι ένα σύνολο από τεχνολογίες που βρίσκονται εγκαταστημένες στους σύγχρονους φυλλομετρητές του web.

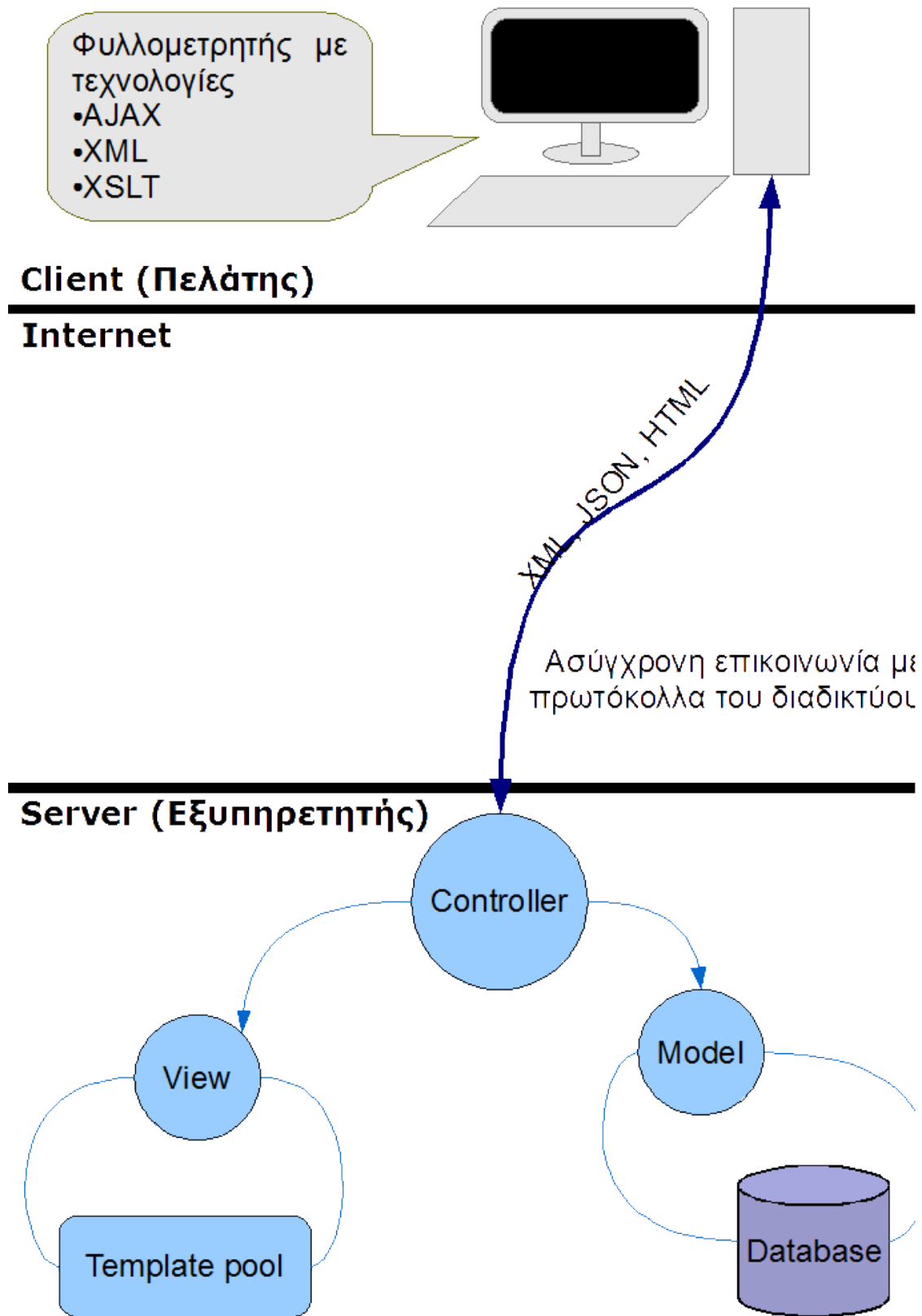
Οι τρεις από αυτές είναι οι:

- Javascript
- Cascade Style Sheets (CSS)
- Document Object Model (DOM)

Το τέταρτο θα το αναφέρουμε αργότερα μιας και είναι το μόνο που δεν προϋπήρχε και φυσικά είναι αυτό που δίνει πολύ μεγάλη δύναμη στο AJAX. Οι παραπάνω τρεις τεχνολογίες όλες μαζί είναι γνωστότερες και με τον όρο DHTML (Dynamic HTML). Είχαν εισαχθεί στους φυλλομετρητές έτσι ώστε οι παραγωγοί περιεχομένου να μπορούν να προσφέρουν web σελίδες που περιείχαν animation και αλλά πιο αλληλεπιδραστικά στοιχεία σε σχέση με την στατική HTML. Έτσι δηλαδή με την τεχνολογία CSS μπορούμε να αλλάξουμε την εμφάνιση των HTML στοιχείων που περιέχει μια σελίδα, με την Javascript, που είναι μια scripting γλώσσα γενικού σκοπού, να δημιουργήσουμε animations αλλάζοντας χαρακτηριστικά των HTML στοιχείων όπως την θέση τους. Τέλος, όλα τα παραπάνω και άλλα πολλά δεν θα ήταν δυνατά αν το DOM δεν έδινε πρόσβαση στα HTML στοιχεία μιας web σελίδας. Το DOM δημιουργεί μια δενδρική αναπαράσταση των στοιχείων μιας HTML σελίδας και φυσικά μέσω αυτής της αναπαράστασης μπορούμε να προσπελάσουμε οποιοδήποτε στοιχείο. Πρέπει να δώσουμε λίγη προσοχή σε αυτό το σημείο και να κατανοήσουμε ότι όλες αυτές τις τεχνολογίες που αναφέραμε ως τώρα ως συστατικά της AJAX όλα είναι εγκατεστημένα και λειτουργούν στον φυλλομετρητή του χρήστη. Δεν έχουν καμία σχέση με τον εξυπηρετητή του web. Το AJAX είναι ένα σύνολο τεχνολογιών που βρίσκεται στην πλευρά του πελάτη και εκεί φυσικά θα λειτουργήσουν για να προσφέρουν καλύτερες εμπειρίες στους χρήστες

των AJAX enabled web εφαρμογών. Καλά όλα αυτά, όμως πως τελικά το AJAX μπορεί να βοηθήσει τους προγραμματιστές να φτιάξουν web εφαρμογές που θα είναι πιο interactive, θα προσφέρουν πιο πλούσιες λειτουργίες και γενικότερα θα έχουν εφάμιλλη συμπεριφορά με τις κλασικές desktop εφαρμογές; Η απάντηση έρχεται με το τέταρτο χαρακτηριστικό της AJAX, το XMLHttpRequest object. Το αντικείμενο αυτό, που το διαθέτουν όλοι οι σύγχρονοι φυλλομετρητές, δίνει την δυνατότητα στους τελευταίους να κάνουν requests στον web εξυπηρετητή για νέο περιεχόμενο (η καλύτερα για νέα δεδομένα, προς γιατί αυτή η σημείωση θα το αναφέρουμε παρακάτω) χωρίς να χάσουν το περιεχόμενό τους και φυσικά μόλις αυτό ταξιδέψει πίσω στον πελάτη, να παρουσιαστεί σε αυτούς. Έτσι με την τεχνική αυτή πετυχαίνουμε αυτό ακριβώς που θέλουμε. Μόλις ο χρήστης προκαλέσει ένα event πάνω στα περιεχόμενα της web σελίδας ο φυλλομετρητής θα κάνει την αίτηση στο εξυπηρετητή στο παρασκήνιο, χωρίς δηλαδή να καταλάβει τίποτα ο χρήστης. Το περιεχόμενο του φυλλομετρητή εξακολουθεί να είναι διαθέσιμο και ορατό προς τον χρήστη και μόλις τα νέα δεδομένα φτάσουν πίσω στον πελάτη τότε ο φυλλομετρητής αναλαμβάνει να ενημερώσει το περιεχόμενο. Κάνοντας τις αιτήσεις για νέα δεδομένα στο παρασκήνιο, χωρίς δηλαδή η αίτηση να γίνεται αντιληπτή από τον χρήστη, έχει πάρα πολλά πλεονεκτήματα. Καταρχήν ο χρήστης δεν παρατηρεί ποτέ πλέον νεκρούς χρόνους. Τα περιεχόμενα του φυλλομετρητή δεν αδειάζουν ποτέ και παραμένουν πάντα ορατά και διαθέσιμα στον χρήστη, ακόμα και όταν στον παρασκήνιο υπάρχουν μια ή και παραπάνω αιτήσεις για νέα δεδομένα. Καθώς ολοκληρώνονται επιτυχώς οι αιτήσεις που υπάρχουν στο παρασκήνιο, τότε ο φυλλομετρητής ενημερώνει τα περιεχόμενα του με τα νέα δεδομένα. Έτσι το τελικό αποτέλεσμα που απολαμβάνει ο χρήστης της web εφαρμογής είναι όπως αυτό στις κλασικές desktop εφαρμογές. Έτσι βλέπουμε πως ενώ διατηρούμε το request – response μοντέλο του web, με τις ασύγχρονες αιτήσεις στο παρασκήνιο για νέα δεδομένα από τον εξυπηρετητή το αναβαθμίζουμε αποφεύγοντας τις εκ φύσεως αδυναμίες του. Στην προηγούμενη πρόταση αναφέραμε και τον όρο “ασύγχρονες αιτήσεις” ο οποίος χρειάζεται λίγο εξήγηση. Είπαμε ότι όταν ένας φυλλομετρητής κάνει μια αίτηση σε ένα εξυπηρετητή, τότε αυτός αδειάζει τα περιεχόμενά τους και φυσικά περιμένει μέχρις ότου το νέο περιεχόμενο ταξιδέψει πίσω. Στο χρονικό αυτό διάστημα ο χρήστης του φυλλομετρητή δεν μπορεί να κάνει τίποτα άλλο παρά μόνο να περιμένει και αυτός για το νέο υλικό. Σε αυτή την περίπτωση έχουμε σύγχρονη (προσοχή όχι ασύγχρονη) κλήση προς ένα εξυπηρετητή με αποτέλεσμα να μην μπορεί να πραγματοποιηθεί άλλη ενέργεια κατά την διάρκειά της. Στην περίπτωση όμως της AJAX,

οι αιτήσεις γίνονται ασύγχρονα. Μόλις ο χρήστης ενεργοποιήσει μια λειτουργία μέσω ενός event τότε ο φυλλομετρητής θα αποστείλει την αίτηση προς τον εξυπηρετητή αλλά αυτό δεν θα έχει σαν αποτέλεσμα το “πάγωμα” των περιεχομένων του. Ο χρήστης εξακολουθεί να μπορεί να πλοηγείται μέσα σε αυτά και μάλιστα να καλέσει και άλλες λειτουργίες, οι οποίες με την σειρά τους θα ενεργοποιήσουν κι άλλες ασύγχρονες αιτήσεις προς τον εξυπηρετητή. Ένας προγραμματιστής ελαφρώς πιο διορατικός θα διακρίνει ότι κάθε ασύγχρονη κλήση προς τον εξυπηρετητή μπορεί να γίνεται με τον ίδιο τρόπο αλλά ο χειρισμός όμως των απαντήσεων μπορεί να διαφέρει σε κάθε μια. Στο κλασικό request – response μοντέλο του web οι απαντήσεις ήταν πάντα περιεχόμενο κωδικοποιημένο σε HTML. Έτσι ο φυλλομετρητής δεν είχε και πολλά να κάνει παρά να παρουσιάζει το καινούριο περιεχόμενο. Στο AJAX όμως είπαμε ότι ο φυλλομετρητής θα πρέπει να ανανεώσει τα περιεχόμενα που προβάλλει με τα νέα δεδομένα χωρίς απαραίτητα να διαγράψει όλα τα προηγούμενα που ήδη περιέχει. Για παράδειγμα μπορεί ο φυλλομετρητής να παρουσιάζει μια λίστα με επαφές και ο χρήστης να απενεργοποιήσει ένα φίλτρο. Έτσι ο φυλλομετρητής θα κάνει μια αίτηση για να παραλάβει από τον εξυπηρετητή τις επαφές που πληρούν τα νέα κριτήρια. Όταν αυτά επιστρέψουν πίσω ο φυλλομετρητής πρέπει απλώς να ανανεώσει την προβαλλόμενη λίστα και όχι όλο του το περιεχόμενο. Με αυτό το παράδειγμα και μόνο βλέπουμε πως η ανησυχία του διορατικού προγραμματιστή είναι σωστή. Κάθε απάντηση του εξυπηρετητή στον φυλλομετρητή πρέπει να χειριστεί με διαφορετικό τρόπο από τον τελευταίο. Αυτό επιβάλλει την ύπαρξη κώδικα ο οποίος θα αναλαμβάνει κάθε φορά την σωστή αξιοποίηση των δεδομένων των απαντήσεων και μιας και το περιβάλλον του χρήστη είναι ένας φυλλομετρητής άρα δεν έχουμε και πολλές επιλογές από την Javascript. Έτσι καταλήγουμε στο συμπέρασμα ότι στον φυλλομετρητή του χρήστη πρέπει να τρέχει μια εφαρμογή που θα έχει κανονική συμπεριφορά ενός client και ο οποίος θα αναλαμβάνει να παρακολουθεί τις κινήσεις του χρήστη και να ανταποκρίνεται σε αυτές. Στην παρακάτω εικόνα βλέπουμε την δομή μιας web εφαρμογής που κάνει χρήση της τεχνολογίας AJAX. Στην πλευρά του χρήστη έχουμε έναν φυλλομετρητή που υποστηρίζει τις απαιτούμενες τεχνολογίες Javascript, CSS, DOM, XML και XSLT. Η Javascript μπορεί να θεωρηθεί η κόλλα που θα ενώσει και θα συντονίσει όλες τις υπόλοιπες τεχνολογίες. Ο κώδικας που θα γραφεί στην Javascript φυσικά θα έχει χρέη πελάτη (client). Θα συμπεριφέρεται σαν να ήταν μια εφαρμογή τύπου πελάτη γραμμένη σε άλλη γλώσσα προγραμματισμού, μόνο που αυτή θα εκτελείται μέσα σε ένα περιβάλλον φυλλομετρητή.



3. Σύστημα Ηλεκτρονικών Δημοπρασιών

3.1 Η αρχιτεκτονική του συστήματος

Για ένα οποιοδήποτε λογισμικό είναι πάρα πολύ σημαντικό να αναπτυχθεί από τους προγραμματιστές του με μια συγκεκριμένη αρχιτεκτονική. Τα πλεονεκτήματα που προσδίδει η τήρηση μια αρχιτεκτονικής στην δόμηση μια εφαρμογής για ηλεκτρονικούς υπολογιστές είναι πάρα πολλά, όπως πολλές είναι και οι αρχιτεκτονικές που έχουν στα χέρια τους και μπορούν να επιλέξουν οι προγραμματιστές. Μερικά από τα πιο βασικά πλεονεκτήματα είναι φυσικά η κατανόηση του κώδικα του λογισμικού από τρίτους προγραμματιστές που μπορεί να εισαχθούν στον δυναμικό ανάπτυξης του αργότερα, αλλά και η εύκολη αλλαγή και/ή προσθήκη λειτουργιών σε μεταγενέστερο στάδιο (μεταγενέστερο στάδιο μπορεί να χαρακτηριστεί είτε μετά την κυκλοφορία του λογισμικού στο εμπόριο, είτε κατά την αρχική φάση ανάπτυξης πλην του ξεκινήματός του). Επίσης βοηθάει πολύ και στην αποσφαλμάτωση του κώδικα λόγω της καθαρότητας και της ομοιογένειας που τον χαρακτηρίζει, είτε αυτό γίνεται στην φάση της ανάπτυξης του, είτε αργότερα, όταν δηλαδή η εφαρμογή έχει βγει στο εμπόριο. Τέλος δεν θα μπορούσαμε να μην αναφέρουμε πως η δόμηση εφαρμογών βάση γνωστών αρχιτεκτονικών παράγει πολύ επαναχρησιμοποιήσιμο κώδικα, βοηθώντας τους προγραμματιστές σε μελλοντικές τους δουλειές.

Έτσι και στο σύστημα ηλεκτρονικών δημοπρασιών έγινε χρήση της πολύ δυνατής και γνωστής, ειδικότερα στον χώρο του web development, αρχιτεκτονικής MVC (Model – View – Controller). Όπως αναφέραμε νωρίτερα, το σύστημα ηλεκτρονικών δημοπρασιών αποτελείται από δύο κύρια κομμάτια. Αυτό που εκτελείται σε έναν application server και αυτό που “κατεβαίνει” στον ηλεκτρονικό υπολογιστή του χρήστη, ο client. Το κομμάτι που υπόκειται στους κανόνες της δόμησης βάση της αρχιτεκτονικής MVC είναι φυσικά αυτό που τρέχει στον application server. Σε αυτό το κομμάτι θα επικεντρωθεί και η τεχνική ανάλυσή μας. Παρόλα αυτά η MVC αρχιτεκτονική ακολουθήθηκε και στην δόμηση του client χωρίς όμως να παρουσιάζει ενδιαφέρον η εκεί ανάλυση.

Άλλα πράγματα θα μας απασχολήσουν σε αυτόν. Οι τεχνικές λεπτομέρειες σίγουρα είναι πολύ πιο ενδιαφέρουσες για τον server μιας και εκεί δομούνται όλοι οι λεγόμενοι business κανόνες του συστήματος ηλεκτρονικών δημοπρασιών. Θα δούμε αναλυτικά πως η MVC αρχιτεκτονική ξεχωρίζει τον κώδικα που ασχολείται με τον προγραμματισμό των business κανόνων που πρέπει να ισχύουν, για παράδειγμα σε ένα σύστημα ηλεκτρονικών, με τον κώδικα που προγραμματίζει το πώς θα εμφανιστούν αυτά στον χρήστη.

3.2 Η αρχιτεκτονική MVC (model – view – controller)

Από τα πρώτα λάθη που κάνει ένας νέος , και προφανώς άπειρος, προγραμματιστής είναι φυσικά το να αναμιγνύει κώδικα ο οποίος για παράδειγμα συνδέεται με την βάση δεδομένων μαζί με κώδικα που μορφοποιεί τα δεδομένα της βάσης για να εμφανιστούν στον χρήστη. Αν το πάρουμε λίγο πιο αφαιρετικά, θα μπορούσαμε να πούμε ότι στο ίδιο σημείο γράφεται κώδικας ο οποίος αναλαμβάνει να αυτοματοποιήσει τους επιχειρηματικούς κανόνες που πρέπει να προγραμματιστούν, κανόνες φυσικά που τους ορίζει ο πελάτης της εφαρμογής αλλά και η φύση του προβλήματος για το οποίο γράφεται η εφαρμογή, μαζί με κώδικα ο οποίος αναλαμβάνει να μορφοποιήσει τα δεδομένα που προκύπτουν από τους παραπάνω κανόνες σε μορφή τέτοια ώστε να παρουσιαστούν στον χρήστη. Όταν λέμε μορφοποίηση δεδομένων για εμφάνιση στον χρήστη εννοούμε για παράδειγμα την real time σύνταξη html κώδικα με δεδομένα που παράγονται από τους επιχειρηματικούς κανόνες για την αποστολή τους και φυσικά την παρουσίασή τους μέσα από έναν web φυλλομετρητή στον χρήστη. Η διάσπαση αυτών των λογικών τμημάτων μιας εφαρμογής σε δύο όπου το καθένα αναλαμβάνει εξειδικευμένα πλέον κάτι ξεχωριστό θεωρείται πολύ σημαντικό μιας και προσδίδει στην εφαρμογή τεράστια ευελιξία και καθαρότητα στον κώδικα. Σε αυτή τη λογική στηρίζεται η αρχιτεκτονική MVC η οποία προσπαθεί να ξεχωρίσει όσο παραπάνω γίνεται τη συγγραφή κώδικα που αναλαμβάνει να υλοποιήσει τους business κανόνες από αυτόν που αναλαμβάνει να τα μορφοποιήσει σε κατανοητή από τον χρήστη μορφή. Έτσι το γράμμα M από την αρχιτεκτονική MVC προέρχεται από την λέξη model (μοντέλο) και το οποίο είναι υπεύθυνο να κωδικοποιήσει όλους του επιχειρηματικούς κανόνες που διέπουν το σύστημα, ενώ το γράμμα V προέρχεται από την λέξη view (εμφάνιση) από όπου και μπορούμε να καταλάβουμε πως κωδικοποιεί την μορφοποίηση των δεδομένων σε κατάλληλη μορφή για τους πελάτες – χρήστες.

Για το γράμμα C όμως; Το γράμμα αυτό προέρχεται από την λέξη controller (ελεκτήρας) και οι ευθύνες του είναι πολλές και σημαντικές. Εν συντομία θα λέγαμε ότι ο controller έχει τον πολύ σημαντικό ρόλο της εννοχρήστρωσης του model και του view. Στην πραγματικότητα η άριστη υλοποίηση την αρχιτεκτονικής MVC είναι αυτή κατά την οποία ο κώδικας που ανήκει στο model δεν γνωρίζει τίποτα για τον κώδικα του view και φυσικά και το αντίστροφο. Δηλαδή κώδικας του view να μην ξέρει τίποτα για τον κώδικα του model. Για να γίνουμε λίγο πιο κατανοητοί, όταν εννοούμε ότι ο ένας δεν πρέπει να ξέρει τίποτα για τον άλλο στην πραγματικότητα ζητάμε από το κάθε ένα κομμάτι, το model και το view δηλαδή, να μπορεί να εκτελεστεί το καθένα χωρίς να βασίζεται σε τεχνικές λεπτομέρειες υλοποίησης του άλλου. Να είναι δηλαδή δύο εντελώς ανεξάρτητα κομμάτια κώδικα. Κάποιος όμως πρέπει να γνωρίζει καλά και τους δύο, έτσι ώστε να μπορεί να τους συνδυάσει και να παράγει έργο. Αυτός είναι ο controller. Αυτός γνωρίζει πολύ καλά το πως θα συνδυάσει λειτουργίες του μοντέλου με λειτουργίες της προβολής έτσι ώστε να φέρει εις πέρας τις λειτουργίες που απαιτούν οι χρήστες του συστήματος.



Στο παραπάνω σχήμα εικονίζουμε και σχηματικά τις σχέσεις που πρέπει να υπάρχουν μεταξύ των κομματιών κώδικα που απαρτίζουν το μοντέλο (model), την παρουσίαση (view) και φυσικά τον μεσολαβητή (controller). Παρατηρούμε πως το κομμάτι του μοντέλου (model) σαν ένα κομμάτι ενός παζλ ταιριάζει μόνο με τον μεσολαβητή όπως κάνει και το κομμάτι της παρουσίασης (view). Από την εικόνα καταλαβαίνουμε πως το μοντέλο και η παρουσίαση είναι δύο εντελώς ξένα κομμάτια χωρίς κανένα κοινό σημείο. Βέβαια υπάρχει η ανάγκη το μοντέλο και η παρουσίαση να βρουν ένα κοινό τρόπο επικοινωνίας, παρόλα αυτά τα κοινά τους χαρακτηριστικά τελειώνουν εδώ. Αργότερα θα δούμε ποιος τρόπος υπάρχει έτσι ώστε ακόμα και ο διάυλος επικοινωνίας μεταξύ μοντέλου και εμφάνισης να μην επηρεάζει καθόλου την τάση τους ο ένας να αγνοεί την ύπαρξη του άλλου.

3.2.1 Model (μοντέλο)

Ας δούμε λίγο πιο αναλυτικά τι πρέπει να περιέχει και φυσικά τι πρέπει να κωδικοποιηθεί μέσα στο μοντέλο.

Το μοντέλο, όπως αναφέραμε και νωρίτερα, περιλαμβάνει, ή καλύτερα κωδικοποιεί, τους επιχειρηματικούς κανόνες που ισχύουν, χρειάζεται και έχει ζητήσει ο πελάτης του συστήματος. Αυτοί είναι κανόνες ισχύουν συνήθως πάνω στα δεδομένα που θα εισάγονται ή που θα παράγονται από το ίδιο το σύστημα. Για παράδειγμα σε ένα σύστημα ηλεκτρονικών δημοπρασιών ένας τέτοιος κανόνας θα έλεγε πως δεν επιτρέπεται σε ένα χρήστη να πλειοδοτήσει σε μια δημοπρασία αν του έχει αφαιρεθεί το δικαίωμα αυτό. Τέτοιου είδους κανόνες ονομάζονται επιχειρηματικοί και κατά κάποιο τρόπο ορίζουν την συμπεριφορά του συστήματος στους χρήστες του. Το μοντέλο είναι ο ιδανικός χώρος για να προγραμματιστούν αυτές οι συμπεριφορές. Μια εφαρμογή όμως δεν θα είχε καμία αξία αν δεν μπορούσε να αποθηκεύει την πληροφορία που παράγει και επεξεργάζεται. Έτσι το μοντέλο είναι επίσης υπεύθυνο στο να αποθηκεύει αλλά και να ανακτά από μονάδες αποθήκευσης όλα τα δεδομένα που παράγει - διαχειρίζεται. Η αποθήκευση και η ανάκτηση δεδομένων από μονάδες αποθήκευσης δεν περιορίζεται μόνο από και σε ένα τοπικό σκληρό δίσκο αλλά μπορεί να κωδικοποιηθεί έτσι ώστε η αποθήκευση και η ανάκτηση να γίνετε και από απομακρυσμένους υπολογιστές και βάσεις δεδομένων με χρήση δικτύων και άλλων σύνθετων τρόπων ανταλλαγής δεδομένων με πρωτόκολλα και σύνθετες διαδικασίες.

3.2.2 View (παρουσίαση)

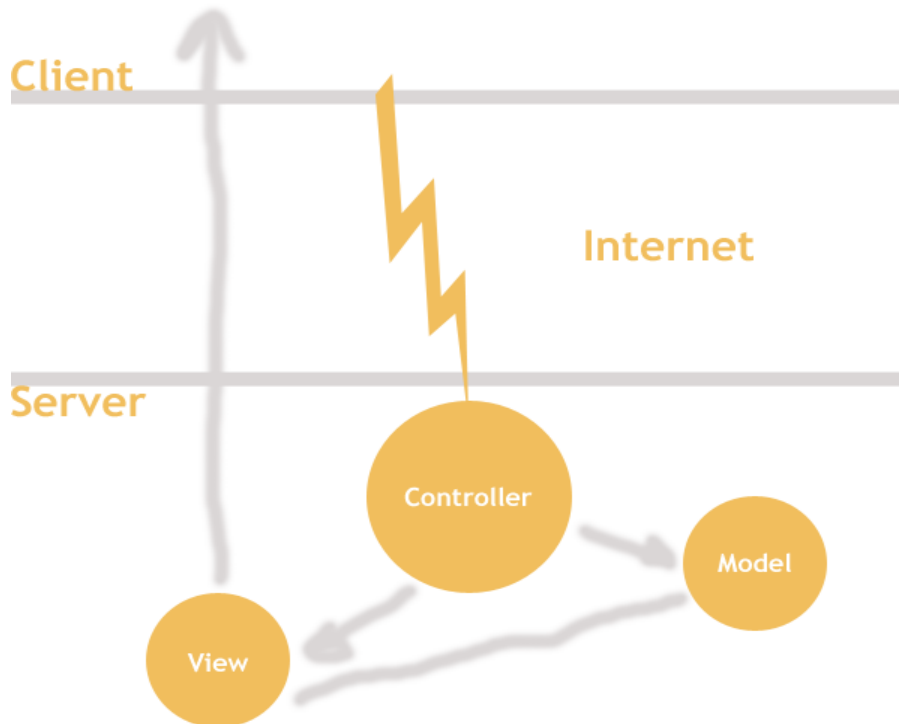
Στην αρχιτεκτονική MVC επίσης πολύ σημαντικό ρόλο παίζει και το κομμάτι του view. Ο κώδικας που αποτελεί αυτό το κομμάτι της αρχιτεκτονικής είναι υπεύθυνο μόνο στο να μορφοποιεί τα δεδομένα που επεξεργάζονται ή παράγονται από το model σε μορφή κατάλληλη για να παρουσιαστούν στον χρήστη. Η κατάλληλη μορφή από χρήστη σε χρήστη μπορεί να διαφέρει πάρα πολύ. Μπορεί κάποιος χρήστης να θέλει να προβάλει τα δεδομένα σε έναν web browser. Έτσι το κατάλληλο view θα κωδικοποιήσει τα δεδομένα σε HTML. Ένας άλλος χρήστης μπορεί να τα χρειάζεται σε XML μορφή, οπότε ένα άλλο view θα τα κωδικοποιήσει σε XML. Κατά επέκταση τα ίδια δεδομένα μπορούν να κωδικοποιηθούν σε οποιαδήποτε μορφή χρειάζονται οι πελάτες –

χρήστες του συστήματος προγραμματίζοντας απλώς το κατάλληλο view. Έτσι μπορούμε να εξυπηρετήσουμε οποιαδήποτε ανάγκη σε μορφοποίηση των δεδομένων, ακόμα και για μελλοντικούς χρήστες χωρίς να χρειάζεται να πειράξουμε ούτε μια γραμμή στον κώδικα του model.

3.2.3 Controller (ελεγκτής ή μεσολαβητής)

Ο ακρογωνιαίος λίθος στην αρχιτεκτονική MVC είναι φυσικά ο controller. Ο τελευταίος, όπως αναφέραμε και παραπάνω, έχει την ευθύνη να ενορχηστρώνει την λειτουργία του model και των view. Είναι αυτός που αναλαμβάνει δράση πρώτος όταν φτάνει στο σύστημα μια αίτηση ενός χρήστη για εξυπηρέτηση. Βάση της αίτησης του χρήστη ο controller καταλαβαίνει τι ζητά και καλεί προς εκτέλεση το κατάλληλο κομμάτι του model. Αφού αυτό ολοκληρώσει την εκτέλεσή του ο controller δεν έχει τίποτα παραπάνω να κάνει παρά να διαλέξει το ανάλογο view που θα μορφοποιήσει τα αποτελέσματα της εκτέλεσης του model σε μορφή κατανοητή από τον χρήστη. Έτσι αν για παράδειγμα ο χρήστης κάνει την αίτηση μέσω ενός φυλλομετρητή τότε όπως είναι λογικό τα αποτελέσματα πρέπει να επιστραφούν σε HTML μορφή για να παρουσιαστούν στον φυλλομετρητή του.

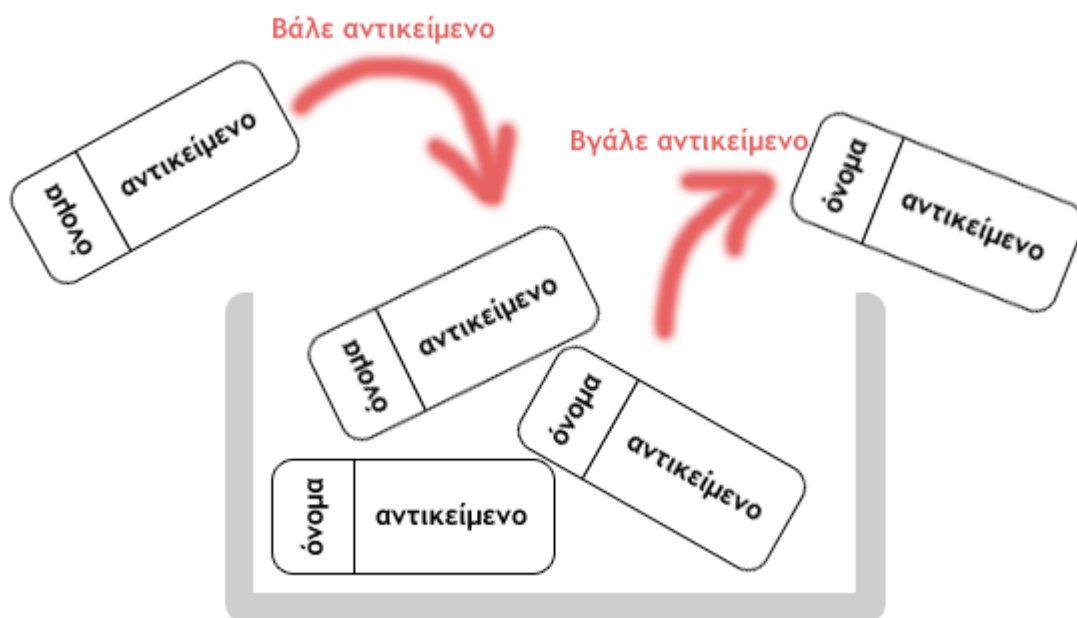
3.3 Υλοποίηση της MVC στο Σύστημα Δημοπρασιών



Στην πάνω εικόνα βλέπουμε ένα αφαιρετικό σχήμα στο πως υλοποιείται η αρχιτεκτονική MVC σε μια εφαρμογή web. Σχολιάζοντας την εικόνα βλέπουμε πως όλες οι αιτήσεις που προέρχονται από τους clients μέσω του internet, δρομολογούνται όλες στον controller. Ο controller αυτός και μόνον αυτός παραλαμβάνει αιτήσεις από χρήστες. Είναι αυτός που ξέρει πως θα χρησιμοποιήσει κατάλληλα το model και το view έτσι ώστε να εξυπηρετήσει του χρήστες και τις αιτήσεις τους. Στο σχήμα φαίνεται πως ο controller είναι συγκριτικά μεγαλύτερος από το model και το view. Αυτό όμως δεν ισχύει αν μιλάμε για πλήθος γραμμών κώδικα. Στην πραγματικότητα το model και το view περιέχουν πολλές εκατοντάδες ή χιλιάδες γραμμές κώδικα σε σχέση με τον controller, παρόλα αυτά η σπουδαιότητα που χαρακτηρίζει τον controller μας επιβάλλει να τον σχεδιάσουμε μεγαλύτερο.

3.4 Registry Pattern ή απλώς 'basket'

Όπως επαναλαμβάνουμε συχνά στην πτυχιακή αυτή, είναι πάρα πολύ σημαντικό κατά MVC ανάπτυξη λογισμικού το model κομμάτι του να μην γνωρίζει αλλά ούτε να χρειάζεται να γνωρίσει κάτι για το view έτσι ώστε να εκτελέσει κομμάτια κώδικά του. Το ίδιο φυσικά ισχύει και για το view. Κανένας από τους δύο δεν πρέπει να ξέρει τεχνικές λεπτομέρειες για την δομή και υλοποίηση του άλλου. Όμως μεταξύ τους υπάρχει η ανάγκη ενός διαύλου επικοινωνίας. Αυτοί οι δύο (το model και view) είναι και πρέπει να παραμείνουν ξένοι ο ένας προς τον άλλον. Όμως θα πρέπει να μπορούν να συνεννοηθούν μεταξύ τους με μία γλώσσα που δεν επηρεάζει τις τεχνικές τους λεπτομέρειες ούτε να φανερώνει λεπτομέρειες στην υλοποίηση του ενός στον άλλο. Η γλώσσα αυτή υπάρχει και ακούει στο όνομα registry pattern ή απλώς 'basket'. Στην πραγματικότητα μπορούμε να εικονοποιήσουμε την γλώσσα αυτή με ένα καλάθι το οποίο έχει τις εξής ιδιότητες. Μπορούμε μέσα του να αποθηκεύσουμε ό,τι θέλουμε, αλλά οτιδήποτε βάζουμε μέσα πάει παρέα (πακέτο) με ένα αναγνωριστικό όνομα. Αργότερα όταν θέλουμε να βγάλουμε κάτι από το καλάθι, δεν έχουμε παρά να διατάξουμε το καλάθι να μας δώσει πίσω αυτό που είχε αποθηκεύσει με το συγκεκριμένο όνομα. Στην παρακάτω εικόνα βλέπουμε πως λειτουργεί το καλάθι μας.



Έτσι για να υλοποιηθεί ο διάυλος επικοινωνίας μεταξύ model και view το μόνο που χρειάζονται

να ξέρουν και οι δύο είναι απλώς με ποιο κωδικό όνομα θα αποθηκεύσουν ή θα αντλήσουν την πληροφορία που θέλουν μέσα από το καλάθι. Έτσι οποιαδήποτε στιγμή το model ή το view χρειαστεί να πάρει ή να βάλει μια πληροφορία μέσα, δεν έχει παρά να ζητήσει από το καλάθι να του πει αν έχει μέσα του κάποιο αντικείμενο με αυτό το όνομα ή να το προστάξει να αποθηκεύσει κάτι με ένα συγκεκριμένο όνομα – κωδικό.

Ας δούμε ένα απλό παράδειγμα στο σύστημα ηλεκτρονικών δημοπρασιών. Ας πούμε πως ένας χρήστης του συστήματος θέλει σε XML μορφή όλες τις ανοιχτές δημοπρασίες που κλείνουν σε 2 ημέρες. Ο χρήστης κάνει την αίτησή του καλώντας τον ανάλογο ελεγκτή (controller). Ο ελεγκτής αυτός ξέρει επακριβώς πιο κομμάτι του model διαβάζει από τη βάση δεδομένων τις πληροφορίες που ζητά ο χρήστης και τον καλεί να το διεκπεραιώσει. Το κομμάτι αυτό του model εκτελείται και τελικά βάζει μέσα στο καλάθι μια λίστα με δημοπρασίες ας πούμε με το όνομα – κωδικό “Δημοπρασίες που κλείνουν σε δύο μέρες από σήμερα”.

Παρατήρηση: Το όνομα – κωδικός δεν είναι υποχρεωτικά μια λέξη, μπορεί να είναι και μια φράση όπως έχουμε στο παράδειγμα μας. Όμως καλό είναι να έχουμε μονολεκτικά ονόματα.

Μετά από την εκτέλεση του model ο ελεγκτής πρέπει να καλέσει το κατάλληλο view που μπορεί μετατρέψει μια λίστα από δημοπρασίες σε XML μορφή ώστε να ταξιδέψουν πια πίσω στον χρήστη. Οπότε το κομμάτι view που καλεί προς εκτέλεση ο ελεγκτής ζητάει από το καλάθι την λίστα με τις δημοπρασίες που θα κωδικοποιήσει σε XML με το όνομα – κωδικό “Δημοπρασίες που κλείνουν σε δύο μέρες”. Έτσι πλέον το view έχει στην διάθεσή του τα δεδομένα που διάβασε το model από τη βάση δεδομένων χωρίς να ξέρει τίποτα για αυτόν, τα κωδικοποιεί σε XML μορφή και τα δίνει πίσω στον ελεγκτή όπου αυτός με την σειρά του θα τα επιστρέψει στον χρήστη.

3.5 Σενάρια Χρήσης (use cases)

Σε αυτή την ενότητα θα δούμε αναλυτικά τα σενάρια χρήσης που έχουν κατασκευαστεί για το σύστημα ηλεκτρονικών δημοπρασιών. Κάθε αναλυτικό σενάριο αποτελείται από τρεις σελίδες.

Στην πρώτη σελίδα έχουμε την αναλυτική περιγραφή του σεναρίου όπως:

- Actor

Ποιος δηλαδή μπορεί να εκκινήσει αυτό το σενάριο χρήσης.

- Description

Περιγραφή του σεναρίου χρήσης.

- Precondition

Συνθήκες που πρέπει να ισχύουν για να εκκινήσει το σενάριο χρήσης,

- Postcondition

Συνθήκες που πρέπει να ισχύουν όταν ολοκληρωθεί το σενάριο

- Exception

Διάφορες εξαιρέσεις που είναι πιθανό να πετάξει το σενάριο χρήσης σε περίπτωση που κάτι δεν πάει καλά.

Στην δεύτερη σελίδα έχουμε την αναλυτική περιγραφή των βημάτων που γίνονται για την πραγματοποίηση του σεναρίου από το λογισμικό (το λεγόμενο και ως Normal flow ή Happy senario, τα βήματα δηλαδή κατά τα οποία δεν πάει κάτι στραβά). Τα βήματα αυτά περιγράφονται αναφέροντας την ενέργεια του χρήστη και μετά την απάντηση του συστήματος στην ενέργεια αυτή. Τα βήματα αυτά δεν μπορούν να θεωρηθούν πλήρη για να οδηγήσουν τον προγραμματιστή στην παραγωγή του κώδικά, παρά αρκετά αφηρημένα ώστε να παρουσιάσουν στον πελάτη τις λειτουργίες που θα προσφέρει το σύστημα. Τέλος η τελευταία σελίδα σε κάθε σενάριο χρήσης αναλαμβάνει να περιγράψει τα βήματα που πρέπει να ακολουθηθούν σε περίπτωση που κάποιο από τα βήματα στο Happy Senario δεν πάει καλά. Αυτά ονομάζονται και εναλλακτικά σενάρια χρήσης και φυσικά μπορεί να είναι πάρα πολλά.

Στον πίνακα κάτω έχουμε την λίστα με τα σενάρια χρήσης και την σελίδα από την οποία ξεκινούν.

Τίτλος σεναρίου	Σελίδα
Δημιουργία νέου λογαριασμού	
Εισαγωγή στο Σύστημα Ηλεκτρονικών Δημοπρασιών	
Εισαγωγή Νέας Δημοπρασίας	
Εισαγωγή Δημοπρασία στα Αγαπημένα	
Δημιουργία Προσφοράς σε Δημοπρασία	
Αποστολή Προσωπικού Μηνύματος	
Άνοιγμα Νέων Δημοπρασιών	

Τίτλος σεναρίου	Σελίδα
Κλείσιμο Δημοπρασιών	
Αποστολή SMS	

Use Case ID: 001	
Use Case Name: Δημιουργία νέου λογαριασμού χρήστη	
Created By: Μαρκόπουλος Γρηγόριος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

Actors: System User	
Description: Ένας νέος επισκέπτης εκδηλώνει το ενδιαφέρον να κάνει χρήση των υπηρεσιών ηλεκτρονικών δημοπρασιών που προσφέρονται από το σύστημα. Για το λόγο αυτό πρέπει να δημιουργήσει (ανοίξει) νέο λογαριασμό στο σύστημα ώστε να έχει πρόσβαση στις υπηρεσίες του.	
Precondition: Δεν υπάρχουν ορατές συνθήκες που πρέπει να πληρούνται για να ξεκινήσει ένας επισκέπτης την διαδικασία αυτή.	
Postcondition:	<ul style="list-style-type: none"> ● Πρέπει να έχει σταλεί ένα ηλεκτρονικό μήνυμα (e-mail) στην διεύθυνση που έδωσε ο υποψήφιος χρήστη με τον κωδικό του. <p style="text-align: center;">ή</p> <ul style="list-style-type: none"> ● Πρέπει να έχει σταλεί ένα ηλεκτρονικό μήνυμα (sms) στο κινητό τηλέφωνο του υποψήφιου χρήστη με τον κωδικό του.
Exceptions:	<ul style="list-style-type: none"> ● Database server down or not reachable ● Application server down
Priority:	Frequency Of Use:
Includes:	
Business Rule:	
Special Requirements:	Δεν υπάρχουν ορατές ειδικές απαιτήσεις που πρέπει να πληρούνται για να ξεκινήσει ο υποψήφιος χρήστης την διαδικασία δημιουργίας νέου λογαριασμού.
Notes:	

Normal Flow: (Happy Scenario)		User Action	System Response
1	Ο χρήστης κάνει κλικ πάνω στο button “Γίνε μέλος” στο κυρίως μενού του συστήματος.	2	Το σύστημα ανταποκρίνεται εμφανίζοντας την φόρμα εισαγωγής στοιχείων νέου χρήστη
3	Ο υποψήφιος χρήστης εισάγει τα στοιχεία του στα πεδία της φόρμας και πατάει το button “Εντάξει”.	4	Το σύστημα ελέγχει αν ο υποψήφιος χρήστης έχει εισάγει όλα τα απαραίτητα στοιχεία.
		5	Το σύστημα στέλνει πίσω στον server τα στοιχεία του χρήστη.
		6	Τα στοιχεία του χρήστη ελέγχονται από τον server για τυχών αδυναμίες εγκυρότητας.
		7	Δημιουργείτε ένας νέος χρήστης με τα δοθέντα στοιχεία και αποθηκεύεται στην βάση δεδομένων.
		8	Ο κωδικός του χρήστη αποστέλλεται στην ηλεκτρονική διεύθυνση του χρήστη.
		9	Εμφανίζεται μήνυμα επιτυχίας στον “πλέον κανονικό” χρήστη του συστήματος.

Alternative Flows:	Για το βήμα 4		
	User Action		System Response
			<p>4.1 Το σύστημα καταλαβαίνει πως ο χρήστης δεν έχει συμπληρώσει όλα τα υποχρεωτικά πεδία της φόρμας.</p> <p>4.2 Το σύστημα εμφανίζει διακριτικά σημάδια στις άκρες των πεδίων που δεν έχουν συμπληρωθεί αλλά που είναι απαραίτητα και περιμένει μέχρις ότου ο χρήστης τα συμπληρώσει και πατήσει πάλι το button “Εντάξει”.</p>
Για το βήμα 6			
	User Action		System Response
			<p>6.1 Το σύστημα κατά τον έλεγχο της εγκυρότητας των στοιχείων ανακαλύπτει πως το όνομα χρήστη που επέλεξε ο χρήστης δεν μπορεί να χρησιμοποιηθεί διότι έχει ήδη δοθεί σε άλλο χρήστη.</p> <p>6.2 Το σύστημα επιστρέφει στον πελάτη την φόρμα με τα στοιχεία του και με ένα μήνυμα που τον ενημερώνει να επιλέξει άλλο όνομα χρήστη.</p>
	6.3 Ο χρήστης προσπαθεί να διορθώσει τα προβλήματα και φυσικά πατάει το button “Εντάξει” μόλις γίνει αυτό.	6.4	Το σύστημα πάει πίσω στο βήμα 4 του happy σεναρίου.

Use Case ID: 002	
Use Case Name: Εισαγωγή στο Σύστημα Ηλεκτρονικών Δημοπρασιών	
Created By: Μαρκόπουλος Γρηγόριος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

Actors: System User	
Description: Ένας χρήστης του συστήματος επισκέπτεται την ιστοσελίδα, φορτώνεται το πρόγραμμα πελάτης στον χρήστη και πλέον ο τελευταίος θέλει να εισαχθεί στο σύστημα για να κάνει χρήση των υπηρεσιών του.	
Precondition:	<ul style="list-style-type: none"> ● Ο χρήστης πρέπει να έχει λογαριασμό στο σύστημα. ● Το πρόγραμμα πελάτης να έχει φορτωθεί και να εκτελείται σωστά.
Postcondition:	<ul style="list-style-type: none"> ● Το σύστημα να έχει αναγνωρίσει τον χρήστη βάση των κωδικών που έδωσε. ● Το σύστημα να παρουσιαστεί προσαρμοσμένο στις κινήσεις του χρήστη.
Exceptions:	<ul style="list-style-type: none"> ● Database server down or not reachable ● Application server down
Priority:	Frequency Of Use:
Includes:	
Business Rule:	
Special Requirements:	Δεν υπάρχουν ορατές ειδικές απαιτήσεις που πρέπει να πληρούνται για να ξεκινήσει ο χρήστης το αναλυόμενο σενάριο χρήσης.
Notes:	

Normal Flow: (Happy Scenario)		User Action	System Response
1	Ο χρήστης γράφει το όνομα χρήστη και τον κωδικό του στα κατάλληλα πεδία στην TODAY καρτέλα του πρόγραμμα πελάτη και πατάει το button “Εντάξει”.	2	Το πρόγραμμα πελάτη ελέγχει να έχουν γραφεί όνομα χρήστη και κωδικός.
		3	Το πρόγραμμα πελάτη στέλνει πίσω στον server το όνομα χρήστη και κωδικό.
		4	Ο server ελέγχει να υπάρχει χρήστης με αυτό το όνομα και αν ναι αν ο κωδικό που έδωσε τεριάζει
		5	Το σύστημα επιστρέφει πίσω στον πελάτη μήνυμα επιτυχίας με το πραγματικό του όνομα.
		6	Το πρόγραμμα πελάτη ανανεώνει την καρτέλα TODAY με τα προσωπικά μηνύματα και ανακοινώσεις του χρήστη.

Alternative Flows:	Για το βήμα 4		
	User Action		System Response
	4.3	Ο χρήστης προσπαθεί ξανά.	4.1 Το σύστημα βλέπει πως δεν υπάρχει χρήστης με το δοθέντα όνομα.
			4.2 Το σύστημα επιστρέφει και παρουσιάζει μήνυμα λάθους στον χρήστη.
4.4 Το σύστημα εκτελείται πάλι από το βήμα 2.			
Για το βήμα 4			
User Action		System Response	
4.3	Ο χρήστης προσπαθεί ξανά	4.1 Το σύστημα βλέπει ότι υπάρχει χρήστης με το δοθέντα όνομα αλλά ο κωδικός που έδωσε ο χρήστη δεν ταιριάζει με αυτόν που έχει το σύστημα.	
		4.2 Το σύστημα επιστρέφει και παρουσιάζει στον χρήστη μήνυμα λάθους.	
		4.4 Το σύστημα εκτελείται πάλι από το βήμα 2.	

Use Case ID: 003	
Use Case Name: Εισαγωγή Νέας Δημοπρασίας	
Created By: Παναγιωτάκης Ευάγγελος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

Actors: System User	
Description: Χρήστης του συστήματος θέλει να εκτελέσει μια δημοπρασία. Δηλαδή να δώσει την δυνατότητα σε άλλους χρήστες του συστήματος να βρουν και να πλειοδοτήσουν ώστε να αποκτήσουν τον αντικείμενο που τίθεται σε δημοπρασία.	
Precondition:	<ul style="list-style-type: none">● Ο χρήστης πρέπει να έχει λογαριασμό στο σύστημα.● Ο χρήστης να έχει δικαίωμα εισαγωγής νέας δημοπρασίας στο σύστημα
Postcondition:	
Exceptions:	<ul style="list-style-type: none">● Database server down or not reachable● Application server down
Priority:	Frequency Of Use:
Includes: 002	
Business Rule:	
Special Requirements:	Δεν υπάρχουν ορατές ειδικές απαιτήσεις που πρέπει να πληρούνται για να ξεκινήσει ο χρήστης το αναλυόμενο σενάριο χρήσης.
Notes:	

Normal Flow: (Happy Scenario)	User Action		System Response	
	1	Ο χρήστης κάνει κλικ στο button “Εισαγωγή νέας δημοπρασίας”.	2	Το σύστημα ανταποκρίνεται στην αίτηση του χρήστη εμφανίζοντάς του την φόρμα εισαγωγή νέας δημοπρασίας.
	3	Ο χρήστης συμπληρώνει τη φόρμα εισαγωγής νέας δημοπρασίας με τα στοιχεία του αντικειμένου του. Στο τέλος πατάει το button “Εντάξει”.	4	Το σύστημα ελέγχει αν έχουν συμπληρωθεί όλα τα πεδία που είναι απαραίτητα.
			5	Το σύστημα στέλνει πίσω στον server τα δεδομένα της νέας δημοπρασίας.
			6	Ο server ελέγχει για αδυναμία εγκυρότητας των δεδομένων.
			7	Το σύστημα καταχωρεί την νέα δημοπρασία.
			8	Το σύστημα επιστρέφει και εμφανίζει στον χρήστη μήνυμα επιτυχίας.

Alternative Flows:		Για το βήμα 4	
		User Action	System Response
			4.1 Το σύστημα βλέπει πως κάποιο/κάποια υποχρεωτικά πεδία δεν έχουν συμπληρωθεί.
			4.2 Το σύστημα εμφανίζει διακριτικά σημάδια στις άκρες των πεδίων που υπάρχει πρόβλημα
4.3	Ο χρήστης προσπαθεί να διορθώσει τα προβλήματα συμπληρώνοντας τα άδεια πεδία.		4.4 Το σύστημα εκτελείται πάλι από το βήμα 4.
		Για το βήμα 6	
		User Action	System Response
			6.1 Κατά τον έλεγχο εγκυρότητας των δεδομένων δημοπρασίας παρουσιάστηκε κάποιο πρόβλημα με αυτά. Για παράδειγμα η ημερομηνία λήξη να είναι πιο πριν από την ημερομηνία έναρξης.
			6.2 Το σύστημα επιστρέφει και εμφανίζει στον χρήστη την φόρμα εισαγωγή νέας δημοπρασίας με τα πεδία συμπληρωμένα και με διακριτικά σημάδια εκεί που υπάρχει πρόβλημα εγκυρότητας.
6.3	Ο χρήστης προσπαθεί να διορθώσει τα προβλήματα και πατάει το button “Εντάξει”.	6.4	Το σύστημα εκτελείται πάλι από το βήμα 4.

Use Case ID: 004	
Use Case Name: Εισαγωγή Δημοπρασία στα Αγαπημένα	
Created By: Παναγιωτάκης Ευάγγελος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

Actors: System User	
Description: Χρήστης του συστήματος εκδηλώνει ενδιαφέρον να παρακολουθεί την εξέλιξη μια δημοπρασίας στη διάρκεια του χρόνου με μια σειρά ανακοινώσεων με δυνατότητα αποστολής τους στο e-mail του χρήστη ή/και στο κινητό τηλέφωνο του χρήστη με sms ή/και στο σύστημα ανακοινώσεων του συστήματος ηλεκτρονικών δημοπρασιών.	
Precondition:	<ul style="list-style-type: none">● Ο χρήστης πρέπει να έχει λογαριασμό στο σύστημα.
Postcondition:	
Exceptions:	<ul style="list-style-type: none">● Database server down or not reachable● Application server down
Priority:	Frequency Of Use:
Includes: 002	
Business Rule:	
Special Requirements:	<ul style="list-style-type: none">● Η προς παρακολούθηση δημοπρασία πρέπει να είναι ανοιχτή.
Notes:	

Normal Flow: (Happy Scenario)	User Action	System Response
	1 Ο χρήστης κάνει κλικ πάνω στο button “Εισαγωγή Δημοπρασίας στα αγαπημένα”.	2 Το σύστημα ελέγχει αν η δημοπρασία αυτή είναι ήδη στα αγαπημένα του χρήστη.
	4 Ο χρήστης επιλέγει ή αποεπιλέγει ποιες θέλει ή ποιες δεν θέλει από τους διαθέσιμους τρόπους ενημέρωσης και πατάει το button “Εντάξει”.	3 Το σύστημα εμφανίζει στον χρήστη τον διάλογο εισαγωγή της δημοπρασίας στα αγαπημένα με τικαρισμένες τις επιλογές ενημέρωσης που είχε επιλέξει ο χρήστης την προηγούμενη φορά, αν υπήρχε ήδη στα αγαπημένα του.
5 Το σύστημα στέλνει πίσω στον server τις νέες επιλογές του χρήστη.		
6 Το σύστημα αποθηκεύει τις αλλαγές.		
		7 Το σύστημα επιστρέφει και εμφανίζει μήνυμα επιτυχίας στον χρήστη.

Alternative Flows:

Use Case ID: 005	
Use Case Name: Δημιουργία Προσφοράς σε Δημοπρασία	
Created By: Παναγιωτάκης Ευάγγελος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

Actors: System User	
Description: Χρήστης του συστήματος εκδηλώνει το ενδιαφέρον για την αγορά ενός αντικειμένου σε δημοπρασία οπότε και κάνει μια προσφορά σε αυτό.	
Precondition:	<ul style="list-style-type: none">● Ο χρήστης πρέπει να έχει λογαριασμό στο σύστημα.● Ο χρήστης να έχει δικαίωμα δημιουργίας προσφοράς σε δημοπρασία
Postcondition:	
Exceptions:	<ul style="list-style-type: none">● Database server down or not reachable● Application server down
Priority:	Frequency Of Use:
Includes: 002	
Business Rule:	
Special Requirements:	<ul style="list-style-type: none">● Η δημοπρασία στην οποία γίνεται η προσφορά πρέπει να είναι ανοιχτή.
Notes:	

Normal Flow: (Happy Scenario)		User Action	System Response
1	Ο χρήστης γράφει τον κωδικό του και το μέγιστο ποσό που θέλει να διαθέσει για την αγορά του προς πώληση αντικειμένου της δημοπρασίας και φυσικά πατάει το button “Εντάξει”.	2	Το σύστημα ελέγχει αν έχουν γραφεί και τα δύο πεδία.
		3	Το σύστημα στέλνει πίσω στον server τα στοιχεία της προσφοράς.
		4	Ο server ελέγχει αν ο χρήστης έχει ήδη μια προσφορά για την δημοπρασία αυτή.
		5	Ο server ελέγχει την εγκυρότητα των δεδομένων για την προσφορά.
		6	Το σύστημα καταχωρεί την προσφορά στην βάση δεδομένων για την δεδομένη δημοπρασία.
		7	Το σύστημα επιστρέφει και εμφανίζει μήνυμα επιτυχίας στον χρήστη.

Alternative Flows:	Για το βήμα 4	
	User Action	System Response
		4.1 Το σύστημα βλέπει πως ο χρήστης έχει ήδη κάνει μια προσφορά σε αυτή τη δημοπρασία.
		4.2 Το σύστημα επιστρέφει και εμφανίζει στον χρήστη μήνυμα στο οποίο του αναφέρει πως έχει κάνει παλαιότερα μια προσφορά, το ποσό που έχει δώσει και φυσικά τον ενημερώνει πως να κάνει νέα προσφορά διαγράφοντας την παλιά.
	4.3 Ο χρήστης παίρνει τις κατάλληλες αποφάσεις βάση των επιθυμιών του.	
Για το βήμα 6		
	User Action	System Response
		6.1 Στον έλεγχο εγκυρότητας των δεδομένων προκύπτει σφάλμα. Για παράδειγμα το ποσό της προσφοράς δεν είναι έγκυρο είτε γιατί δεν ακολουθεί το βήμα αύξησης είτε γιατί είναι μικρότερο από το επιτρεπτό σε εκείνη τη φάση της δημοπρασίας.
		6.2 Το σύστημα επιστρέφει και εμφανίζει ανάλογο μήνυμα σφάλματος στον χρήστη.

Use Case ID: 006	
Use Case Name: Αποστολή Προσωπικού Μηνύματος	
Created By: Μαρκόπουλος Γρηγόριος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

Actors: System User	
Description: Χρήστης θέλει να επικοινωνήσει με άλλο χρήστη του συστήματος, παράδειγμα για να τον ρωτήσει παραπάνω πληροφορίες για μια δημοπρασία του δεύτερου.	
Precondition:	<ul style="list-style-type: none">● Ο χρήστης πρέπει να έχει λογαριασμό στο σύστημα.● Ο χρήστης να έχει δικαίωμα αποστολής προσωπικού μηνύματος.
Postcondition:	
Exceptions:	<ul style="list-style-type: none">● Database server down or not reachable● Application server down
Priority:	Frequency Of Use:
Includes: 002	
Business Rule:	
Special Requirements:	Δεν υπάρχουν ορατές ειδικές απαιτήσεις που πρέπει να πληρούνται για να ξεκινήσει ο υποψήφιος χρήστης το αναλυόμενο σενάριο χρήσης.
Notes:	

Normal Flow: (Happy Scenario)		User Action	System Response
1	Ο χρήστης πατάει το button “Αποστολή Προσωπικού Μηνύματος”.	2	Το σύστημα εμφανίζει την φόρμα αποστολή προσωπικού μηνύματος.
3	Ο χρήστης γράφει το μήνυμά του και πατάει το button “Εντάξει”.	4	Το σύστημα στέλνει πίσω στον server τα δεδομένα του προσωπικού μηνύματος.
		5	Το σύστημα ελέγχει να ο χρήστης που στέλνει το μήνυμα έχει το δικαίωμα να το κάνει αυτό.
		6	Το σύστημα καταχωρεί το μήνυμα στα προσωπικά μηνύματα του παραλήπτη.
		7	Το σύστημα επιστρέφει και εμφανίζει μήνυμα επιτυχίας στον χρήστη.

Alternative Flows:	Για το βήμα 5		
	User Action		System Response
			5.1 Κατά τον έλεγχο του αν έχει το δικαίωμα να στέλνει προσωπικά μηνύματα, βλέπει πως του έχει αφαιρεθεί το δικαίωμα αυτό.
			5.2 Η αίτηση αποστολής προσωπικού μηνύματος αγνοείται.
		5.3 Το σύστημα επιστρέφει και εμφανίζει κατάλληλο μήνυμα στον χρήστη.	

Use Case ID: 007	
Use Case Name: Άνοιγμα Νέων Δημοπρασιών	
Created By: Μαρκόπουλος Γρηγόριος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

Actors: Scheduler	
Description: Το σύστημα αυτοματοποίησης διαδικασιών σαν χρήστης του συστήματος ζητά να ανοίξει όλες τις δημοπρασίες που έχει φτάσει ο χρόνος ανοίγματός τους. Μια δημοπρασία γίνεται διαθέσιμη στους χρήστες από την ώρα που ανοίγει.	
Precondition: Δεν υπάρχουν ορατές συνθήκες που πρέπει να πληρούνται για να ξεκινήσει η διαδικασία αυτή.	
Postcondition:	
Exceptions:	<ul style="list-style-type: none">● Database server down or not reachable● Application server down
Priority:	Frequency Of Use:
Includes:	
Business Rule:	
Special Requirements:	Δεν υπάρχουν ορατές ειδικές απαιτήσεις που πρέπει να πληρούνται για να ξεκινήσει ο χρήστης το αναλυόμενο σενάριο χρήσης.
Notes:	

Normal Flow: (Happy Scenario)		User Action	System Response
1	Μόλις εκπληρωθούν όλες οι χρονικές συνθήκες ενός schedule task (συγκεκριμένα του task που ζητά το άνοιγμα δημοπρασιών), ζητά από το σύστημα να εκτελέσει το ανάλογο task.	2	Το σύστημα καταγράφει την τρέχουσα ημερομηνία και ώρα και ψάχνει να βρει στην βάση δεδομένων για δημοπρασίες που πρέπει να ανοίξουν.
		3	Για κάθε μια δημοπρασία που έχει επιστρέψει το πιο πάνω βήμα, το σύστημα θέτει την κατάσταση της καθεμιά σε "ΑΝΟΙΧΤΗ".
		4	Το σύστημα καταγράφει σε log αναλυτικές λεπτομέρειες για τις κινήσεις του.
		5	Το σύστημα επιστρέφει με τον αριθμό των δημοπρασιών που άνοιξε.

Alternative Flows:

Use Case ID: 008	
Use Case Name: Κλείσιμο Δημοπρασιών	
Created By: Μαρκόπουλος Γρηγόριος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

Actors: Scheduler	
Description: Το σύστημα αυτοματοποίησης διαδικασιών σαν χρήστης του συστήματος ζητά να κλείσει όλες τις δημοπρασίες που έχει φτάσει ο χρόνος κλεισίματος τους. Μετά από το κλείσιμο μιας δημοπρασίας έχουμε και τον νικητή της.	
Precondition: Δεν υπάρχουν ορατές συνθήκες που πρέπει να πληρούνται για να ξεκινήσει η διαδικασία αυτή.	
Postcondition:	
Exceptions:	<ul style="list-style-type: none">● Database server down or not reachable● Application server down
Priority:	Frequency Of Use:
Includes:	
Business Rule:	
Special Requirements:	Δεν υπάρχουν ορατές ειδικές απαιτήσεις που πρέπει να πληρούνται για να ξεκινήσει ο χρήστης το αναλυόμενο σενάριο χρήσης.
Notes:	

Normal Flow: (Happy Scenario)		User Action	System Response
1	Μόλις εκπληρωθούν όλες οι χρονικές συνθήκες ενός schedule task (συγκεκριμένα του task που ζητά το κλείσιμο δημοπρασιών), ζητά από το σύστημα να εκτελέσει το ανάλογο task.	2	Το σύστημα καταγράφει την τρέχουσα ημερομηνία και ώρα και ψάχνει να βρει στην βάση δεδομένων για δημοπρασίες που πρέπει να κλείσουν.
		3	Για κάθε μια δημοπρασία που έχει επιστρέψει το πιο πάνω βήμα, το σύστημα θέτει την κατάσταση της καθεμιά σε "ΚΛΕΙΣΤΗ".
		4	Το σύστημα για την κάθε μια δημοπρασία που κλείνει υπολογίζει και τον νικητή της. Αυτό γίνεται παίρνοντας τον χρήστη με την πιο μεγάλη προσφορά.
		5	Το σύστημα στέλνει μήνυμα στον εκάστοτε νικητή για το αποτέλεσμα.
		6	Το σύστημα καταγράφει σε log αναλυτικές λεπτομέρειες για τις κινήσεις του.
		7	Το σύστημα επιστρέφει με τον αριθμό των δημοπρασιών που έκλεισε.

Alternative Flows:

Use Case ID: 009	
Use Case Name: Αποστολή SMS	
Created By: Παναγιωτάκης Ευάγγελος	Last Updated By:
Date Created: 22/03/2007	Date Last Updated:

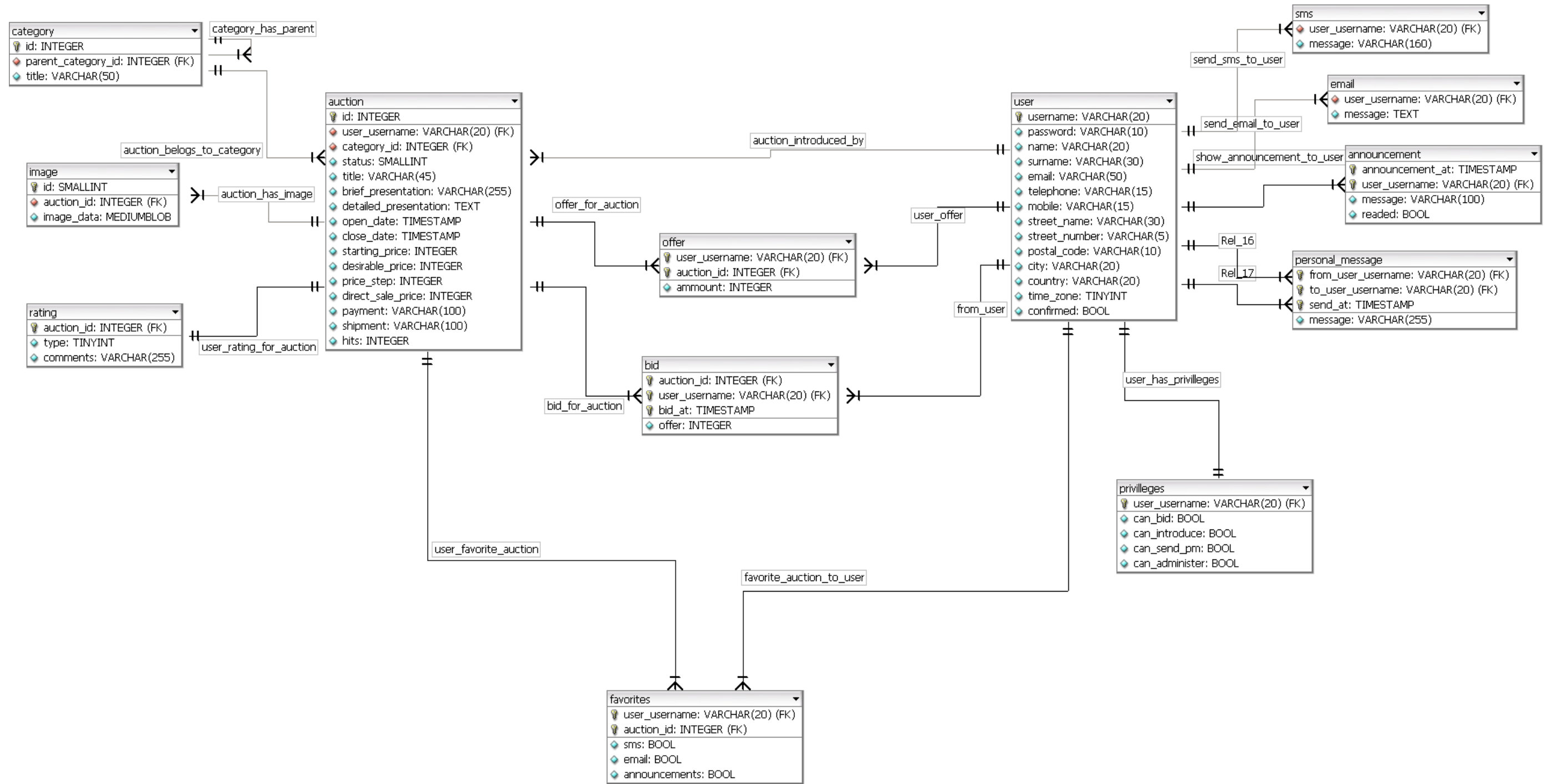
Actors: Scheduler	
Description: Το σύστημα αυτοματοποίησης διαδικασιών σαν χρήστης του συστήματος ζητά να αποσταλούν όλες οι ανακοινώσεις που έχουν βγει και έχουν δρομολογηθεί για αποστολή με γραπτό μήνυμα σε κινητό τηλέφωνο.	
Precondition: Δεν υπάρχουν ορατές συνθήκες που πρέπει να πληρούνται για να ξεκινήσει η διαδικασία αυτή.	
Postcondition:	
Exceptions:	<ul style="list-style-type: none">● Database server down or not reachable● Application server down
Priority:	Frequency Of Use:
Includes:	
Business Rule:	
Special Requirements:	Δεν υπάρχουν ορατές ειδικές απαιτήσεις που πρέπει να πληρούνται για να ξεκινήσει ο χρήστης το αναλυόμενο σενάριο χρήσης.
Notes:	

Normal Flow: (Happy Scenario)		User Action	System Response
1	Μόλις εκπληρωθούν όλες οι χρονικές συνθήκες ενός schedule task (συγκεκριμένα του task που ζητά την αποστολή sms ανακοινώσεων), ζητά από το σύστημα να εκτελέσει το ανάλογο task.	2	Το Σύστημα διαβάζει όλες τις ανακοινώσεις που έχουν μπει στην λίστα αναμονής για αποστολή.
		3	Το σύστημα αρχινά την αποστολή τους ένα προς ένα με σειρά που υπακούει στους κανόνες FIFO (First In First Out). Σε κάθε αποστολή το εκάστοτε μήνυμα διαγράφεται. Μετά από το βήμα 2, δηλαδή αφού έχουν διαβαστεί όλες οι ανακοινώσεις που περιμένουν στην λίστα αποστολής, όσα εισαχθούν μετά θα αποσταλούν την επόμενη φορά που το scheduler εκτελέσει αυτό το σενάριο πάλι.

Alternative Flows:

3.6 Ανάλυση της Βάσης Δεδομένων

Το σύστημα βάσης δεδομένων που επιλέχθηκε για να υποστηρίξει τις ανάγκες αποθήκευσης του συστήματος ηλεκτρονικών δημοπρασιών είναι η MySQL στην 5^η της έκδοση. Για την κατασκευή του απαραίτητου σχήματος της βάσης δεδομένων (ER διάγραμμα και εξαγωγή της SQL) έγινε χρήση της εφαρμογής DBDesigner. Στην παρακάτω εικόνα μπορούμε να δούμε συνοπτικά το σχήμα της βάσης δεδομένων.



3.6.1 Λεπτομερή ανάλυση της βάσης δεδομένων.

Πίνακας : user

user	
username:	VARCHAR(20)
password:	VARCHAR(10)
name:	VARCHAR(20)
surname:	VARCHAR(30)
email:	VARCHAR(50)
telephone:	VARCHAR(15)
mobile:	VARCHAR(15)
street_name:	VARCHAR(30)
street_number:	VARCHAR(5)
postal_code:	VARCHAR(10)
city:	VARCHAR(20)
country:	VARCHAR(20)
time_zone:	TINYINT
confirmed:	BOOL

Κρίναμε απαραίτητο να δημιουργήσαμε ένα πίνακα όπου θα αποθηκεύονται τα στοιχεία του χρήστη. Ο πίνακας αυτός θα μας βοηθήσει στο να διαχειριζόμαστε σωστά όλα τα στοιχεία των χρηστών. Το πεδίο username είναι το κύριο κλειδί του πίνακα αυτού. Επειδή το κάθε username θα αντιστοιχεί σε έναν μόνο χρήστη κρίναμε απαραίτητο να είναι αυτό το κύριο κλειδί του πίνακα. Παρακάτω υπάρχει ένας πίνακας όπου αναλύονται όλα τα στοιχεία του πίνακα λεπτομερώς.

Όνομα	Τύπος	Υποχρεωτικό	Περιγραφή
Username	Varchar : 20	NAI	Το πεδίο username είναι το κύριο κλειδί. Δεν μπορεί δύο χρήστες να έχουν το ίδιο username. Εδώ θα αποθηκεύεται το επιθυμητό username του χρήστη
Password	Varchar : 10	NAI	Στο πεδίο αυτό αποθηκεύεται ο κωδικός εισόδου του χρήστη.
Name	Varchar : 20	NAI	Στα δύο αυτά πεδία αποθηκεύεται το ονοματεπώνυμο του χρήστη.
Surname	Varchar : 30	NAI	
Email	Varchar : 50	NAI	Στο πεδίο αυτό αποθηκεύεται το email του χρήστη για να υπάρχει μια επικοινωνία με το σύστημα και για να του αποστέλλονται διάφορες χρήσιμες

Όνομα	Τύπος	Υποχρεωτικό	Περιγραφή
			πληροφορίες
Telephone	Varchar : 15	NAI	Στα πεδία αυτά αποθηκεύονται τα τηλέφωνα επικοινωνίας του χρήστη. Κάποιο σταθερό του τηλέφωνο και αν υπάρχει κάποιο κινητό.
Mobile	Varchar : 15	OXI	
Street_name	Varchar : 30	NAI	Για να γίνει η αποστολή ή η παραλαβή κάποιου προϊόντος από τον χρήστη χρειάζεται η πλήρης διεύθυνσή του. Έτσι με αυτά τα 5 πεδία αποθηκεύουμε την πλήρη διεύθυνση του χρήστη.
Street number	Varchar : 5		
Postal code	Varchar : 10		
City	Varchar : 20		
Country	Varchar : 20		
Timezone	Tiny Integer	NAI	Επειδή κάποιοι χρήστες μπορεί να βρίσκονται σε άλλη χώρα, κρίναμε απαραίτητο να βάλουμε την ζώνη ώρας. Έτσι όλες οι δημοπρασίες θα ξεκινούνε την ώρα που πρέπει.
Confirmed	Boolean	NAI	Σε αυτό το πεδίο αποθηκεύουμε αν ο χρήστης έχει ενεργοποιήσει τον λογαριασμό του. Εξ' ορισμού έχει την τιμή false. Θα γίνει true μόλις ο χρήστης απαντήσει στο email «εγγραφής».

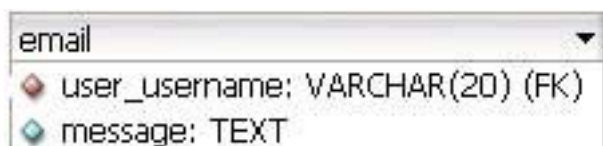
Πίνακας : sms

sms
user_username: VARCHAR(20) (FK)
message: VARCHAR(160)

Ο πίνακας user συσχετίζεται με μια συσχέτιση 1 προς πολλά με τον πίνακα sms. Αυτός ο πίνακας έχει μόλις δύο πεδία. Το πρώτο είναι το πεδίο user_username, το οποίο είναι το ξένο κλειδί που προέρχεται από τον πίνακα user. Είναι τύπου varchar και έχει μέγιστο πλήθος 20 χαρακτήρων. Όπως βλέπουμε το κύριο κλειδί αυτού του πίνακα είναι ξένο κλειδί. Προέρχεται από τον πίνακα user και είναι το κλειδί του πίνακα user. Αυτό έγινε γιατί το κάθε μήνυμα θα αποστέλεται στον χρήστη. Το δεύτερο πεδίο είναι το message. Εδώ θα αποθηκεύεται το μήνυμα που είναι να σταλεί στον χρήστη. Είναι και αυτό τύπου varchar με μέγιστο μέγεθος τους 160

χαρακτήρες.

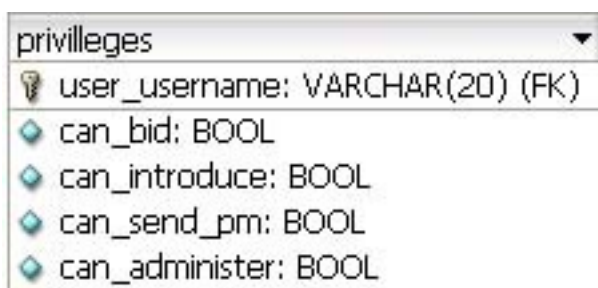
Πίνακας : *email*



Όπως και με τον πίνακα sms, κρίναμε απαραίτητο να αποθηκεύουμε και τα μηνύματα ηλεκτρονικού ταχυδρομείου που θα αποστέλλονται στους χρήστες. Έτσι

δημιουργήσαμε τον πίνακα email γι' αυτό τον σκοπό. Συνδέεται με τον πίνακα user με μια συσχέτιση τύπου ένα προς πολλά, δηλαδή ένας χρήστης μπορεί να δεχθεί πολλά email. Υπάρχουν δύο πεδία το user_username και το message. Το user_username είναι το ξένο κύριο κλειδί του πίνακα user. Το πεδίο message είναι τύπου text και θα αποθηκεύεται εκεί το μήνυμα που θα αποσταλεί στον χρήστη.

Πίνακας : *privileges*



Για να διευκολυνθούμε στην ανάπτυξη του συστήματος και στο χειρισμό των δικαιωμάτων του κάθε χρήστη, κρίναμε ότι θα ήταν πιο εύχρηστο να δημιουργήσουμε έναν πίνακα όπου θα έχουμε τα δικαιώματα και με μια συσχέτιση με τον χρήστη θα προκύπτουν τα δικαιώματά του. Η συσχέτιση αυτή είναι τύπου ένα προς ένα. Το κύριο κλειδί αυτού

του πίνακα όπως βλέπουμε προέρχεται από τον πίνακα user και είναι κύριο – ξένο κλειδί. Καταλήξαμε σε αυτό γιατί τα δικαιώματα θα είναι ατομικά για τον κάθε χρήστη. Έτσι, επειδή ο κάθε χρήστης «ξεχωρίζει» με το username του, κρίναμε ότι αυτό θα έπρεπε να είναι το κλειδί αυτού του πίνακα.

Όνομα	Τύπος	Περιγραφή
User_username	Varchar : 20	Το πεδίο αυτό είναι το κύριο – ξένο κλειδί του πίνακα και προέρχεται από τον πίνακα user
Can_bid	Boolean	Τα πεδία αυτά έχουν εξ' ορισμού την τιμή false.

Όνομα	Τύπος	Περιγραφή
Can_introduce	Boolean	Γίνονται true ανάλογα με τα δικαιώματα του κάθε χρήστη.
Can_send_pm	Boolean	
Can_administer	Boolean	

Πίνακας : *announcement*



Field Name	Field Type
announcement_at	TIMESTAMP
user_username	VARCHAR(20) (FK)
message	VARCHAR(100)
readed	BOOLEAN

Στην ανάλυση που κάναμε θεωρήσαμε ότι θα είναι πιο εύχρηστο να δημιουργήσουμε έναν πίνακα με τις ανακοινώσεις. Αυτό θα μας επιτρέπει να αποθηκεύουμε τις ανακοινώσεις και να τις στέλνουμε στους χρήστες. Η συσχέτιση μεταξύ των δύο πινάκων είναι ένα προς πολλά.

Έχει 4 πεδία και τα δύο εξ αυτών αποτελούν το κύριο κλειδί. Το πεδίο `user_username` το οποίο είναι ξένο – κύριο κλειδί και είναι του πίνακα `user`. Το άλλο πεδίο που αποτελεί το κύριο κλειδί είναι το πεδίο `announcement_at`. Είναι τύπου `timestamp` και αποθηκεύει την ακριβή ημερομηνία και ώρα που θα σταλεί η ανακοίνωση. Το πεδίο `message` είναι τύπου `varchar` με μέγεθος το πολύ 100 χαρακτήρες. Τέλος, υπάρχει και το πεδίο `readed`, το οποίο είναι τύπου `Boolean`. Έχει εξ' ορισμού την τιμή `false` και γίνεται `true` μόλις ο χρήστης διαβάσει την ανακοίνωση.

Πίνακας : *personal_message*

personal_message	
🔑	from_user_username: VARCHAR(20) (FK)
🔑	to_user_username: VARCHAR(20) (FK)
🔑	send_at: TIMESTAMP
💎	message: VARCHAR(255)

Όπως και με τον πίνακα announcement, έτσι και με τον πίνακα personal_message έχουμε τα ίδια αποτελέσματα. Ο πίνακας αυτός αποτελείται από 4 πεδία από τα οποία τα τρία δημιουργούν το κύριο κλειδί.

Τα πεδία from_user_username και

to_user_username τα οποία είναι τα ξένα κλειδιά από τον πίνακα user. Αυτά μας βοηθούν στο να αποθηκεύουμε από ποιον χρήστη στάλθηκε το μήνυμα σε ποιον χρήστη. Το πεδίο send_at είναι το τρίτο πεδίο που συντελεί στην δημιουργία του κύριου κλειδιού. Είναι τύπου timestamp και αποθηκεύει την ακριβή ημερομηνία και ώρα αποστολής του μηνύματος. Τέλος, υπάρχει ακόμη ένα πεδίο, το πεδίο message. Και σε αυτόν τον πίνακα αποθηκεύουμε εδώ το περιεχόμενο του μηνύματος. Είναι και αυτό τύπου varchar με μέγιστο πλήθος χαρακτήρων 255.

Πίνακας : auction

id: INTEGER
user_username: VARCHAR(20) (FK)
category_id: INTEGER (FK)
status: SMALLINT
title: VARCHAR(45)
brief_presentation: VARCHAR(255)
detailed_presentation: TEXT
open_date: TIMESTAMP
close_date: TIMESTAMP
starting_price: INTEGER
desirable_price: INTEGER
price_step: INTEGER
direct_sale_price: INTEGER
payment: VARCHAR(100)
shipment: VARCHAR(100)
hits: INTEGER

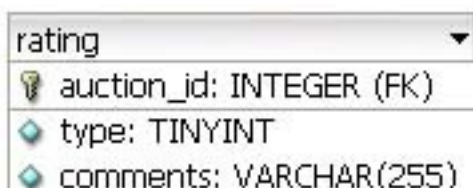
Ο άλλος βασικός πίνακας στη βάση δεδομένων μας είναι ο πίνακας auction. Όπως και με τον πίνακα user, τον δημιουργήσαμε έτσι, γιατί κρίναμε ότι έτσι θα είναι πιο εύκολο και εύχρηστο να διαχειριζόμαστε τις δημοπρασίες που θα υπάρχουν στο σύστημά μας. Το κύριο κλειδί σε αυτόν τον πίνακα κρίναμε ότι θα είναι το πεδίο id. Δηλαδή ο μοναδιαίος κωδικός της κάθε δημοπρασίας όταν αυτή θα δημιουργείται και θα αποθηκεύεται στο σύστημά μας. Τα πεδία user_username και category_id είναι ξένα κλειδιά που προέρχονται από τους πίνακες user και category αντίστοιχα. Τον πίνακα category θα τον αναλύσουμε παρακάτω.

Όνομα	Τύπος	Υποχρεωτικό	Περιγραφή
id	Integer	NAI	Το σύστημα αυτόματα δίνει έναν μοναδιαίο κωδικό σε κάθε καινούρια δημοπρασία.
User_username	Varchar : 20	NAI	Το σύστημα βλέπει ποιος χρήστης δημιουργεί την δημοπρασία και αποθηκεύει το username του.
Category_id	Integer	NAI	Το σύστημα βλέπει σε ποια κατηγορία θέλει να το βάλει ο χρήστης και αποθηκεύει τον κωδικό της κατηγορίας.

Όνομα	Τύπος	Υποχρεωτικό	Περιγραφή
status	Small Integer	NAI	Υποδηλώνει το αν η δημοπρασία είναι ανοιχτή ή κλειστή ή αν θα ανοίξει σε λίγο.
title	Varchar : 45	NAI	Ο τίτλος του προϊόντος που βάζει σε δημοπρασία ο χρήστης.
Brief_presentation	Varchar : 255	OXI	Ο χρήστης μπορεί εδώ να γράψει δυο λόγια για το προϊόν.
Detailed_presentation	Text	OXI	Εδώ ο χρήστης μπορεί να δώσει λεπτομερή ανάλυση των στοιχείων του προϊόντος.
Open_date	Timestamp	NAI	Σε αυτά τα δύο πεδία αποθηκεύονται οι ημερομηνίες και οι ακριβείς ώρες που θα ανοίξει – κλείσει η κάθε δημοπρασία.
Close_date	Timestamp	NAI	
Starting_price	Integer	NAI	Στο πεδίο αυτό ο χρήστης θέτει την τιμή εκκίνησης της δημοπρασίας.
Desirable_price	Integer	OXI	Εδώ ο χρήστης θέτει αν θέλει μια επιθυμητή τιμή που θα ήθελε να πιάσει το προϊόν του.
Price_step	Integer	NAI	Ο χρήστης εδώ μπορεί να ορίσει το με τη βήμα θα αυξάνεται η τιμή της δημοπρασίας
Direct_sale_price	Integer	NAI	Σε αυτό το πεδίο ο χρήστης ορίζει μια τιμή με την οποία κάποιος χρήστης μπορεί να αγοράσει το προϊόν άμεσα
Payment	Varchar : 100	NAI	Αποθηκεύεται το πώς επιθυμεί να πληρωθεί ο χρήστης από τον νικητή της δημοπρασίας.

Όνομα	Τύπος	Υποχρεωτικό	Περιγραφή
shipment	Varchar : 100	NAI	Αποθηκεύεται ο τρόπος με τον οποίο ο χρήστης θα αποστείλει το προϊόν στον νικητή της δημοπρασίας.
hits	Integer	NAI	Είναι ένας μετρητής ο οποίος μας δείχνει πόσοι χρήστες έχουν δει την συγκεκριμένη δημοπρασία.

Πίνακας : *rating*



rating
⚡ auction_id: INTEGER (FK)
💎 type: TINYINT
💎 comments: VARCHAR(255)

Κάθε δημοπρασία δέχεται κάποια κριτική σχετικά με το προϊόν, αλλά σχετικά και με τον χρήστη που την δημιούργησε. Γι' αυτό στην ανάλυσή μας σκεφτήκαμε να δημιουργήσουμε ένα πίνακα όπου θα σχετίζεται με τον πίνακα auction και θα αποθηκεύει την κριτική της κάθε

δημοπρασίας. Έτσι δημιουργήσαμε τον πίνακα rating. Ο πίνακας αυτός έχει τρία πεδία. Το πρώτο πεδίο που είναι το auction_id είναι το ξένο – κύριο κλειδί που προέρχεται από τον πίνακα auction. Το πεδίο type, το οποίο είναι τύπου tiny integer και αποθηκεύει το είδος της κριτικής (θετική +1, αρνητική -1, ουδέτερη 0). Τέλος, το πεδίο comments το οποίο είναι τύπου varchar με πλήθος το πολύ 255 χαρακτήρες. Εκεί πέρα αποθηκεύονται τα σχόλια του χρήστη προς τον χρήστη της δημοπρασίας.

Πίνακας : *image*

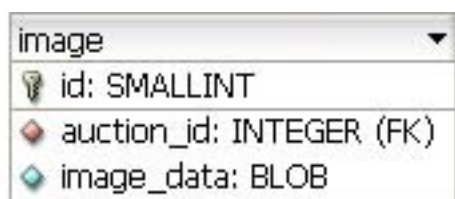


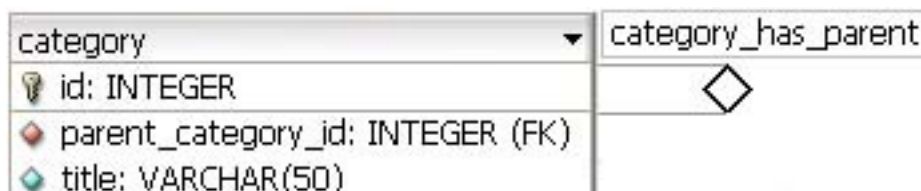
image
⚡ id: SMALLINT
💎 auction_id: INTEGER (FK)
💎 image_data: BLOB

Σε όλες τις εφαρμογές δημοπρασιών, το κάθε προϊόν έχει κάποιες φωτογραφίες για να προσελκύσει τους πελάτες. Έτσι κι εμείς δημιουργήσαμε έναν πίνακα image, ο οποίος συνδέεται με τον πίνακα auction και αποθηκεύει τις φωτογραφίες για την κάθε δημοπρασία. Το κύριο κλειδί του

πίνακα αυτού είναι το πεδίο id, το οποίο είναι τύπου small integer και παίρνει τιμή αυτόματα από το σύστημα κάθε φορά που κάποιος χρήστης προσθέτει μια φωτογραφία στην δημοπρασία του. Το

πεδίο `auction_id` είναι ξένο κλειδί και προέρχεται από τον πίνακα `auction`. Τέλος, υπάρχει το πεδίο `image_data` όπου αποθηκεύεται σε δεδομένα η εικόνα. Το πεδίο αυτό είναι τύπου `blob`.

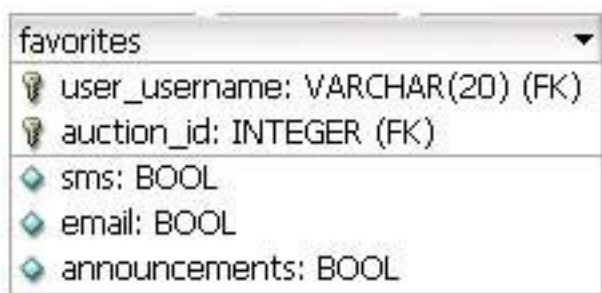
Πίνακας : *category*



Η κάθε δημοπρασία υπάγεται σε μία κατηγορία. Για να είναι πιο εύκολη η διαχείριση των δημοπρασιών και

των κατηγοριών, δημιουργήσαμε ένα πίνακα, τον `category`, στον οποίο αποθηκεύουμε την κάθε κατηγορία. Πρώτα από όλα το κύριο κλειδί. Το κύριο κλειδί είναι το πεδίο `id`. Είναι τύπου `integer` και συμπληρώνεται αυτόματα από το σύστημα κατά τη δήλωση της κάθε κατηγορίας. Το πεδίο `parent_category_id` είναι ξένο κλειδί και προέρχεται από τον πίνακα `category`. Το πεδίο `title` αποθηκεύει τον τίτλο της κατηγορίας. Είναι τύπου `varchar` και έχει μέγεθος το πολύ 50 χαρακτήρες.

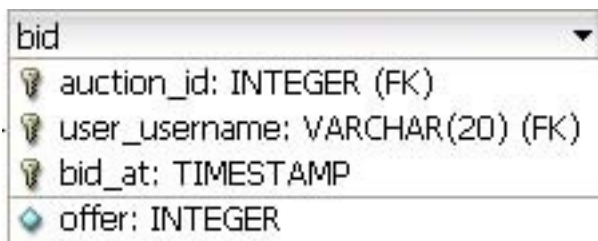
Πίνακας : *favorites*



Σε αυτόν τον πίνακα αποθηκεύονται οι δημοπρασίες τις οποίες επιθυμεί ο χρήστης να υπάρχουν στα αγαπημένα του. Το κύριο κλειδί αυτού του πίνακα είναι μέρος δύο κύριων – ξένων κλειδίων. Το πρώτο είναι από τον πίνακα `user` το πεδίο `user_username`. Το δεύτερο είναι από τον πίνακα `auction` και είναι το πεδίο

`auction_id`. Τα πεδία `sms`, `email` και `announcement` είναι τύπου `Boolean`. Έχουν εξ' ορισμού την τιμή `false`. Γίνονται `true` όταν ο χρήστης ζητήσει να λαμβάνει ειδοποίηση για κάποια δημοπρασία που έχει βάλει στα αγαπημένα με κάποιον από αυτούς τους τρόπους.

Πίνακας : bid

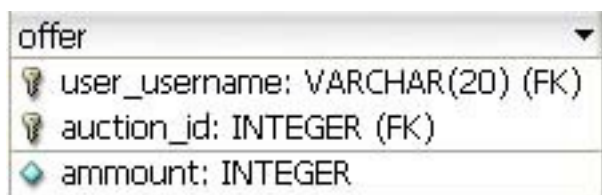


bid
🔑 auction_id: INTEGER (FK)
🔑 user_username: VARCHAR(20) (FK)
🔑 bid_at: TIMESTAMP
🔹 offer: INTEGER

Στον πίνακα bid αποθηκεύουμε τα πονταρίσματα του κάθε χρήστη για την συγκεκριμένη δημοπρασία. Το κύριο κλειδί αυτού του πίνακα σχηματίζεται από τρία πεδία. Το πρώτο, auction_id είναι το ξένο – κύριο κλειδί από τον πίνακα auction. Το user_username είναι

το ξένο – κύριο κλειδί από τον πίνακα user. Το τρίτο πεδίο είναι το bid_at. Είναι τύπου timestamp και αποθηκεύει την ακριβή ημερομηνία και ώρα που έγινε το ποντάρισμα από τον χρήστη. Τέλος, στο πεδίο offer αποθηκεύεται το ποσό του πονταρίσματος. Το πεδίο offer είναι τύπου integer.

Πίνακας : offer



offer
🔑 user_username: VARCHAR(20) (FK)
🔑 auction_id: INTEGER (FK)
🔹 ammount: INTEGER

Ο τελευταίος πίνακας στην βάση δεδομένων μας είναι ο πίνακας offer. Στον πίνακα αυτόν αποθηκεύουμε τη διαφορά δύο προσφορών που έγιναν από τους χρήστες σε μια δημοπρασία. Το κύριο κλειδί του ορίζεται από δύο πεδία. Το πεδίο

user_username το οποίο είναι ξένο – κύριο κλειδί και είναι από τον πίνακα user και το πεδίο auction_id το οποίο είναι ξένο – κύριο κλειδί και είναι από τον πίνακα auction. Τέλος, υπάρχει το πεδίο amount το οποίο είναι τύπου integer. Σε αυτό αποθηκεύεται η διαφορά των δύο πονταρισμάτων. Ο τρίτος και τελευταίος κοινός πίνακας είναι ο πίνακας offer.

3.7 Οδηγίες Εγκατάστασης Συστήματος

Το σύστημα ηλεκτρονικών δημοπρασιών για να μπορέσει να εξυπηρετήσει του χρήστες του, δηλαδή να τους προσφέρει όλες εκείνες τις υπηρεσίες που στο σύνολό τους βοηθούν στην αλληλεπίδραση των χρηστών για την εκτέλεση δημοπρασιών, πρέπει να γίνουν συγκεκριμένα βήματα εγκατάστασης και ενεργοποίησης του. Βέβαια για να εγκατασταθεί και να ενεργοποιηθεί

σωστά το σύστημα στο σύνολό του, υπάρχουν κάποιες απαιτήσεις, από πλευράς υλικού και λογισμικού, που πρέπει να πληρούνται προτού ξεκινήσει η διαδικασία αυτή. Στο σημείο αυτό θα ήταν άξιο να αναφερθεί πως σε αυτόν το οδηγό θα γίνει μια προσπάθεια να περιγραφούν αναλυτικά όλα τα βήματα, ένα προς ένα, αυτών που απαιτούνται προτού τοποθετηθεί το σύστημα ηλεκτρονικών δημοπρασιών στο υπολογιστικό σύστημα και φυσικά των επομένων για την σωστή ρύθμιση, εγκατάσταση και ενεργοποίηση του. Παρόλα αυτά ο οδηγός αυτός δεν μπορεί να θεωρηθεί πλήρης, ειδικά στις οδηγίες που προσφέρει για την εγκατάσταση και ρύθμιση των προαπαιτούμενων χαρακτηριστικών, αφού αποτελούν προϊόντα, είτε υλικού είτε λογισμικού, τρίτων κατασκευαστών που δεν είναι στην ευθύνη ανάπτυξης και συντήρησης των κατασκευαστών του συστήματος ηλεκτρονικών δημοπρασιών. Αρχικά θα ήταν λογικό να αναφερθούμε στις απαιτήσεις υλικού που έχει το σύστημα ώστε να λειτουργήσει απρόσκοπτα. Το σύστημα ηλεκτρονικών δημοπρασιών έχει κατασκευαστεί έτσι ώστε να λειτουργεί μέσω του διαδικτύου. Ως εκ τούτου η αρχιτεκτονική του είναι βασισμένη σε αυτή του πελάτη – εξυπηρετητή (client – server). Εδώ αμέσως διαφαίνεται πως υπάρχουν ξεχωριστές απαιτήσεις για την λειτουργία του συστήματος στην πλευρά του εξυπηρετητή και του πελάτη. Βασικός μας στόχος είναι όμως να αναφέρουμε και φυσικά να σχολιάσουμε τις απαιτήσεις που υπάρχουν στην μεριά του εξυπηρετητή μιας και εκεί θα στηθεί το σύστημα. Παρόλα αυτά θα αναφερθούμε εκτενώς και στις απαιτήσεις του πελάτη, μιας και αυτές είναι αυξημένες και σίγουρα πάρα πολύ σημαντικές για την τελική και ορθή χρήση των υπηρεσιών του συστήματος. Στη συνέχεια θα δούμε το απαραίτητο λογισμικό που χρειάζεται να έχει στη διάθεσή του ο εξυπηρετητής όπως επίσης και ο πελάτης αλλά και της απαιτούμενες ρυθμίσεις σε αυτό. Στο τελευταίο μας βήμα, αφού πλέον θα έχουμε το υπολογιστικό μας σύστημα έτοιμο να δεχθεί και να εκτελέσει το σύστημα, θα μιλήσουμε για την εγκατάστασή του και φυσικά την σωστή του ρύθμιση.

3.7.1 Απαιτήσεις Υλικού

3.7.1.1 Εξυπηρετητής

Όπως προαναφέραμε το σύστημα ηλεκτρονικών δημοπρασιών είναι κατασκευασμένο για να

λειτουργεί μέσω του διαδικτύου και φυσικά αξιοποιώντας όλα τα πρωτόκολλα, τεχνολογίες και στάνταρ του παγκόσμιου ιστού. Λόγο αυτής της φύσης του πρέπει να υπάρχει ένα υπολογιστικό σύστημα που θα εξυπηρετεί αιτήσεις χρηστών από το διαδίκτυο. Τον ρόλο αυτό παίζει ένας εξυπηρετητή web ο οποίος μέσω κατάλληλου λογισμικού (το τι θα το δούμε στο κεφάλαιο για τις απαιτήσεις λογισμικού) θα απαντάει στις αιτήσεις που θα κάνουν οι χρήστες του. Αμέσως διαφαίνεται και η ανάγκη ύπαρξης μια γρήγορης σύνδεση στο διαδίκτυο μέσω της οποίας θα καταφτάνουν οι αιτήσεις προς εξυπηρέτηση και φυσικά θα αποστέλλονται οι κατάλληλες απαντήσεις. Από πλευράς τεχνικών χαρακτηριστικών ο εξυπηρετητής θα πρέπει να διαθέτει έναν ή κατά προτίμηση παραπάνω δυνατούς επεξεργαστές σε συνδυασμό με τουλάχιστον δύο gigabytes κεντρικής μνήμης για να μπορεί να εξυπηρετήσει άνετα πολλές αιτήσεις ταυτόχρονα. Στην τεχνική ανάλυση του συστήματος θα δούμε πως οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος ηλεκτρονικών δημοπρασιών επιβαρύνουν με πολλές παραπάνω αιτήσεις τον εξυπηρετητή από ό,τι οι κλασικές web εφαρμογές έως τώρα. Σε αυτό το σημείο θα ήταν σκόπιμο να αναφέρουμε πώς το σύστημα ηλεκτρονικών δημοπρασιών κάνει εκτενή χρήση ενός συστήματος βάσεων δεδομένων (λεπτομέρειες στο κεφάλαιο απαιτήσεις λογισμικού). Είναι αρκετά συνηθισμένο το σύστημα βάσεων δεδομένων να λειτουργεί στο ίδιο υπολογιστικό σύστημα με χρέη εξυπηρετή αιτήσεων. Σε αυτή τη περίπτωση πρέπει να γνωρίζουμε πως τα συστήματα βάσεων δεδομένων είναι πάρα πολύ απαιτητικά σε υπολογιστικούς πόρους με αποτέλεσμα οι δυνατότητες εξυπηρέτησης του να μειώνονται σε μεγάλο βαθμό. Γιαυτό και το υπολογιστικό σύστημα πρέπει να είναι αρκετά ισχυρό για να ανταποκριθεί άνετα σε καταστάσεις αυξημένου φόρτου. Παρόλα αυτά υπάρχει η δυνατότητα το σύστημα βάσεων δεδομένων να λειτουργεί σε ξεχωριστό υπολογιστικό σύστημα και με τη χρήση γρήγορου τοπικού δικτύου τα δύο υπολογιστικά συστήματα να επικοινωνούν μεταξύ τους. Αυτό θα έχει σαν αποτέλεσμα την τρομερή αύξηση στην αποτελεσματικότητα και φυσικά στην ανταπόκριση του όλου συστήματος, ακόμα και σε καταστάσεις αυξημένου όγκου αιτήσεων, χωρίς φυσικά καμιά διαφορά στην τελική προσφερόμενη υπηρεσία προς τους χρήστες. Κατ' επέκταση της παραπάνω τεχνικής να αναφέρουμε πως θα μπορούσαμε να μοιράσουμε σε πολλά παραπάνω υπολογιστικά συστήματα διάφορα κομμάτια και υπηρεσίες του συστήματος των δημοπρασιών έτσι ώστε τελικά να έχουμε μια “φάρμα” υπολογιστών όπου ο καθένας τους θα έχει αναλάβει να φέρει εις πέρας μόνο μια λειτουργία. Μια τέτοια δομή θα μπορούσε άνετα να εξυπηρετήσει ταυτόχρονα πολύ μεγάλο αριθμό χρηστών από

όλο τον κόσμο χωρίς ιδιαίτερες ανάγκες. Η όλη θεωρία όμως για αυτές τις τεχνολογίες βρίσκονται πίσω από τη θεματική ενότητα των παράλληλων και κατανεμημένων συστημάτων (parallel and distributed computing) που από μόνο του είναι ένας πολύ μεγάλος τομέας της επιστήμης των υπολογιστών. Εδώ απλώς το αναφέρουμε μιας και μια υπηρεσία που θέλει να σέβεται τον εαυτό της, πρέπει υπό οποιεσδήποτε καταστάσεις να είναι πάντα διαθέσιμη εξήντα λεπτά την ώρα, εικοσιτέσσερις ώρες την ημέρα, επτά ημέρες την εβδομάδα. Συνεχίζοντας τις απαιτήσεις υλικού, βασική ανάγκη για να δουλέψει η υπηρεσία ενημέρωσης μέσω σύντομου γραπτού μηνύματος (sms) σε κινητό είναι ένα GPRS modem με μια ενεργή σύνδεση σε μια εταιρία κινητής τηλεφωνίας. Βέβαια σε περίπτωση αυξημένης χρήσης της υπηρεσίας αυτή ένα απλό GPRS modem δεν θα μπορέσει να ανταποκριθεί στις απαιτήσεις αποστολής γραπτών μηνυμάτων μιας και η ταχύτητά τους περιορίζεται σε το πολύ 20 μηνύματα το λεπτό. Αριθμός πολύ μικρός για να εξυπηρετήσει μεγάλο όγκο χρηστών. Σε περίπτωση αυξημένων αναγκών αποστολής γραπτών μηνυμάτων σίγουρα πρέπει να ενοικιαστεί και να χρησιμοποιηθεί ένας sms gateway που παρέχεται συνήθως από κάποια εταιρία κινητής τηλεφωνίας.

3.7.1.2 Πελάτης

Στην πλευρά του πελάτη τώρα οι απαιτήσεις σε υλικό είναι επίσης σημαντικές. Βέβαια το υλικό του πελάτη δεν είναι κάτι που βρίσκεται υπό τον έλεγχο, είτε φυσικό είτε οικονομικό, αυτού που προσφέρει την υπηρεσία. Παρόλα αυτά είναι χρήσιμο να αναφερθεί. Έτσι ένας πελάτης θα χρειαστεί ένα τυπικό προσωπικό ηλεκτρονικό υπολογιστή φυσικά με δική του επεξεργαστική δύναμη και κεντρική μνήμη έτσι ώστε να μπορεί να εκτελεστεί ένα σύγχρονο λειτουργικό σύστημα πάνω στο οποίο θα στηρίζει την λειτουργία του ένας μοντέρνος φυλλομετρητής. Τις λεπτομέρειες λογισμικού όμως θα της δούμε στο κεφάλαιο απαιτήσεις λογισμικού για τον πελάτη. Επίσης ο τελικός χρήστης χρειάζεται και μια αρκετά γρήγορη σύνδεση στο διαδίκτυο. Η ταχύτητα σύνδεση του πελάτη στο διαδίκτυο επηρεάζει άμεσα την απόδοση που θα απολαμβάνει ο χρήστης από το σύστημα ηλεκτρονικών δημοπρασιών, παρόλα αυτά και μια απλή dialup στα 56kbs (kilobits per second / χιλιάδες bit ανά δευτερόλεπτο) μπορεί να ανταποκριθεί στις απαιτήσεις διασύνδεσης του χρήστη.

3.7.2 Απαιτήσεις Λογισμικού

3.7.2.1 Εξυπηρετητής

Πέρα από τις απαιτήσεις υλικού υπάρχουν και αρκετές απαιτήσεις που αφορούν το λογισμικό που πρέπει να διαθέτει ο εξυπηρετητής αλλά και ο πελάτης για την σωστή λειτουργία του συστήματος ηλεκτρονικών δημοπρασιών. Είναι σίγουρα πιο δίκαιο να αναφερθούμε πρώτα στον εξυπηρετητή μιας και αυτός είναι ο συντονιστής όλων. Όπως αναφέραμε λίγο παραπάνω ο εξυπηρετητής θα δέχεται και θα απαντά σε αιτήσεις που κάνουν οι πελάτες στηριζόμενοι στο http πρωτόκολλο. Φυσικά και οι απαντήσεις του εξυπηρετητή θα είναι μέσω του ίδιου πρωτοκόλλου. Η δυνατότητα αυτή μπορεί να αποκτηθεί μέσω εξειδικευμένου λογισμικού που ονομάζεται web server (εξυπηρετητής για το web) ο οποίος φυσικά εκτελείται μέσα από ένα λειτουργικό σύστημα με υποστήριξη για TCP/IP δίκτυα. Πιο συγκεκριμένα ο web server αναφοράς μας θα είναι ο open source της Apache στην έκδοση 2.2.4. Αν και στην θεωρία οποιοσδήποτε web server θα μπορούσε να ανταποκριθεί στις απαιτήσεις που θέτει το σύστημα ηλεκτρονικών δημοπρασιών, παρόλα αυτά είναι συνετό να χρησιμοποιηθεί αυτός. Επίσης πρέπει να εγκατασταθεί και να ρυθμιστεί ένα περιβάλλον εκτέλεσης script που είναι γραμμένα στην server side γλώσσα προγραμματισμού PHP. Για την PHP αναλυτικά αναφερόμαστε στο κομμάτι της πτυχιακής που συζητάμε τις τεχνολογίες στις οποίες στηρίχτηκε για να αναπτυχθεί το σύστημα. Συνεχίζουμε αναφέροντας πως είναι αναγκαίο το περιβάλλον της PHP να είναι βασισμένο τουλάχιστον στην έκδοση 5.2.3 και πάνω. Επιπλέον αναγκαίες είναι και μια σειρά από βιβλιοθήκες γραμμένες σε PHP. Αυτές είναι οι εξής:

- Zend Framework 0.9
- Smarty 2.5

Όπως και με την PHP έτσι και για αυτές της βιβλιοθήκες αναφερόμαστε στο κεφάλαιο που θήγουμε θέματα τεχνολογιών.

3.7.2.3 Πελάτης

Στον πελάτη τώρα οι απαιτήσεις λογισμικού είναι αρκετά πιο περιορισμένες. Στην

πραγματικότητα ο πελάτης χρειάζεται μόνο ένα μοντέρνο φυλλομετρητή ο οποίος υποστηρίζει όλες τις τεχνολογίες που απαιτεί το σύστημα από τον πελάτη.

Οι τεχνολογίες αυτές είναι οι εξής:

- XHTML 1
- Javascript
- AJAX
- XML
- XSLT

3.7.3 Εγκατάσταση και Ρύθμιση

Όταν θα έχουμε στίσει το υπολογιστικό μας σύστημα με το απαραίτητο λογισμικό που δημιουργούν το περιβάλλον που χρειάζεται το σύστημα ηλεκτρονικών δημοπρασιών για να εκτελεστεί μέσα σε τρία βήματα μπορούμε να το κάνουμε να λειτουργεί

Βήμα 1

Αντιγράφουμε όλο το σύστημα σε ένα κατάλογο στο τοπικό σύστημα αρχείων του εξυπηρετητή μας και δίνουμε στον κατάλογο αυτό δικαιώματα ανάγνωσης, εγγραφής και εκτέλεσης στον χρήστη που έχουμε ορίσει να τρέχει τον apache εξυπηρετητή μας.

Βήμα 2

Δημιουργούμε το σχήμα της βάσης δεδομένων στην MySQL κάνοντας χρήση του αρχείου database_Schema.sql που βρίσκεται μέσα στον κατάλογο installation/database κάτω από τον κατάλογο στον οποίο αντιγράψαμε το σύστημα. Δημιουργούμε και ένα χρήστη με δικαιώματα insert, update και delete.

Βήμα 3

Να ρυθμίσουμε σωστά τρία αρχεία του συστήματος που περιέχουν τις διαδρομές προς βασικά συστατικά του συστήματος αλλά και τον λογαριασμό εισαγωγής στην βάση δεδομένων. Τα αρχεία αυτά είναι τα εξής:

- includePath.php
- eauctions-conf.php
- MySmarty.php

includePath.php

Το αρχείο αυτό περιέχει της πλήρης διαδρομές προς μερικά βασικά συστατικά του συστήματος. Παρακάτω βλέπουμε ένα παράδειγμα των περιεχομένων του αρχείου αυτού.

```
$paths = "C:/php-5.2.1/PEAR  
$paths .= ";C:/e-auctions/library/Zend";  
$paths .= ";C:/e-auctions/library/Smarty/libs";  
$paths .= ";C:/e-auctions/application/models";  
$paths .= ";C:/e-auctions/application/views";  
$paths .= ";C:/e-auctions/application/shared";
```

Βλέπουμε πως στο αρχείο αυτό πρέπει να ορίσουμε έξι διαδρομές συνολικά. Σε όλες τις διαδρομές πλην της πρώτης αν για παράδειγμα “x” είναι το κατάλογος στον οποίο έχουμε αντιγράψει το σύστημα τότε με την σειρά οι διαδρομές πρέπει να κοιτούν κάπως έτσι:

x/library/Zend

x/library/Smarty/Libs

x/application/models

x/view

x/shared

eauctions-conf.php

Σε αυτό το αρχείο πρέπει μόνο να αλλάξουμε το όνομα και τον κωδικό με αυτά του χρήστη που φτιάξαμε όταν δημιουργήσαμε το σχήματα της βάσης.

```
return array (
  'log' =>
  array (
    'ident' => 'eauctions',
    'level' => '7',
  ),
  'propel' =>
  array (
    'datasources' =>
    array (
      'eauctions' =>
      array (
        'adapter' => 'mysql',
        'connection' =>
        array (
          'phptype' => 'mysql',
          'hostspec' => 'localhost',
          'database' => 'eauctions',
          'username' => 'root',
          'password' => 'root',
        ),
      ),
    ),
    'default' => 'eauctions',
  ),
);
```

MySmarty.php

Και εδώ έχουμε να αλλάξουμε δύο σημεία τις κατάλληλες διαδρομές, έτσι όπως δουλέψαμε και στο πρώτο αρχείο. Παρακάτω βλέπουμε ένα παράδειγμα.

```
<?php
require_once("Smarty.Class.php");
class MySmarty extends Smarty
{
    public function __construct()
    {
        $this->template_dir = 'C:/e-
auctions/application/views/templates'
        $this->compile_dir = 'C:/e-
auctions/application/rubbish';
    }
}
?>
```

Έστω “x” είναι ο κατάλογος κάτω από τον οποίο υπάρχει το σύστημα. Οι διαδρομές πρέπει να είναι του τύπου:

x/application/view/templates

x/application/rubbish

3.8 Εγχειρίδιο Τελικού Χρήστη

Συγχαρητήρια!!!!

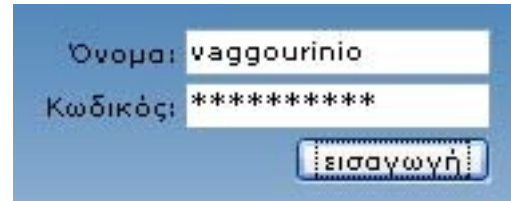
Καλώς ήρθατε στο σύστημα ηλεκτρονικών δημοπρασιών. Εδώ θα μπορέσετε να βρείτε μια πληθώρα προϊόντων σε πολύ καλές τιμές και θα μπορέσετε να τα «χτυπήσετε» για να τα αποκτήσετε. Ας δούμε όμως πως λειτουργεί το σύστημα αυτό. Μόλις επισκεφτείτε την ιστοσελίδα του συστήματος θα δείτε την παρακάτω εικόνα:



Όπως μπορείτε να δείτε η εικόνα χωρίζεται σε δύο μέρη. Στο δεξί μέρος υπάρχει η καρτέλα “Today” η οποία δείχνει τις τέσσερις πιο δημοφιλείς δημοπρασίες αυτήν την στιγμή. Στο αριστερό μέρος υπάρχουν πρώτα από όλα τα κουμπιά λειτουργιών. Αυτά είναι:

- Γίνε μέλος
- Κατηγορίες και δημοπρασίες
- Νέα δημοπρασία
- Οι δημοπρασίες μου
- Τα πονταρίσματά μου
- Αγαπημένα

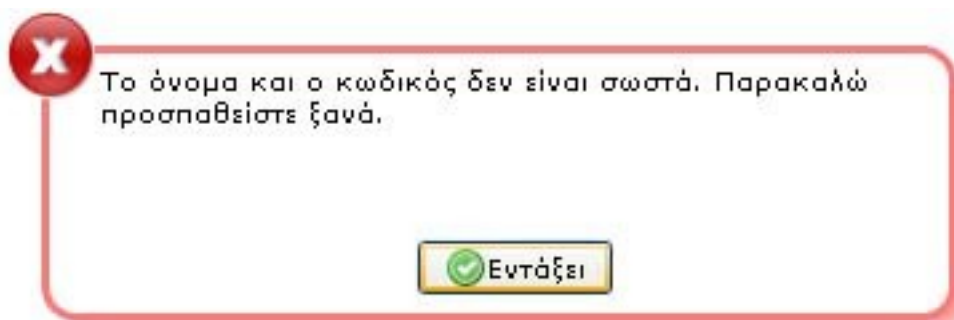
Τις λειτουργίες του κάθε κουμπιού θα τις δούμε αναλυτικά στη συνέχεια του εγχειριδίου χρήσης αφού πρώτα δούμε πως ένας χρήστης που έχει λογαριασμό στο σύστημα, μπορεί να συνδεθεί και να κάνει χρήση των υπηρεσιών. Στο κάτω μέρος της πιο πάνω εικόνας (κάτω από τα κουμπιά των υπηρεσιών) παρατηρούμε την φόρμα εισαγωγή του ονόματος και του κωδικού του χρήστη. Στην δίπλα εικόνα εστιάζουμε στην φόρμα αυτή μόνο. Αφού ο χρήστης γράψει το όνομα και τον κωδικό του, στη συνέχεια πρέπει να πατήσει το κουμπί εισαγωγή. Το σύστημα κάνει τον απαραίτητο έλεγχο και αν είναι σωστά τα στοιχεία του εμφανίζεται το παρακάτω μήνυμα :



Όνομα: vaggourinio
Κωδικός: *****
Εισαγωγή



Αν όμως κάποιο από τα στοιχεία του είναι λάθος θα εμφανιστεί το διπλανό μήνυμα σφάλματος που τον πληροφορεί ότι κάποιο από τα στοιχεία που έδωσε για την εισαγωγή του δεν είναι σωστά.



Τότε ο χρήστης θα πρέπει να διορθώσει τα στοιχεία του και να βάλει τα σωστά έτσι ώστε να μπορέσει να μπει στο σύστημά. Μόλις κάνει "login" με επιτυχία θα ενεργοποιηθούν όλες οι λειτουργίες του συστήματος.

3.8.1 Δημιουργία νέου λογαριασμού (Γίνε μέλος)

Όπως είναι λογικό, για να εκμεταλλευτεί ο χρήστης όλες τις δυνατότητες της εφαρμογής, θα πρέπει να γίνει μέλος. Για να δημιουργήσει λοιπόν κάποιος χρήστης ένα νέο λογαριασμό θα πρέπει να επιλέξει το κουμπί «Γίνε Μέλος» που βρίσκεται στην αριστερή περιοχή της εφαρμογής. Μόλις το επιλέξει, εμφανίζεται μία νέα καρτέλα με την κατάλληλη φόρμα στην οποία πρέπει να συμπληρώσει τα στοιχεία του. Η φόρμα φαίνεται στην παρακάτω εικόνα.

The screenshot shows a web form titled "Φόρμα για νέο λογαριασμό" (Form for new account). The form is divided into several sections. On the left, there is a sidebar with navigation options: "Γίνε μέλος", "Κατηγορίες και δημοπρασίες", "Νέα δημοπρασία", "Οι δημοπρασίες μου", "Τα παντοπωριάτιά μου", and "Αγαπημένα". Below these are input fields for "Όνομα:" and "Κωδικός:" with an "Εισαγωγή" button. The main form area contains the following fields: "Όνομα", "Όροι χρήσης", "Επίθετο", "Ηλεκτρονική διεύθυνση", "Σταθερό Τηλέφωνο", "Κινητό Τηλέφωνο", "Διεύθυνση Κατοικίας", "Αριθμός διεύθυνση Κατοικίας", "Ταχυδρομικός κώδικός", "Πόλη", "Χώρα", "Ζώνη Ώρας", and "Όνομα χρήστη για το σύστημα". A warning message is displayed in the center: "Παρακαλώ συμπληρώστε/διορθώστε όλα τα πεδία που έχουν σημάδι λάθους." (Please complete/correct all fields that have an error sign). At the bottom right, there are buttons for "Εντάξει" and "Άκυρο".

Αν δεν συμπληρώσει κάτι ο χρήστης και πατήσει το κουμπί «Εντάξει», θα εμφανιστούν στα υποχρεωτικά προς συμπλήρωση πεδία το σύμβολο X. Εμφανίζεται επίσης και το μήνυμα «Παρακαλώ συμπληρώστε / διορθώστε όλα τα πεδία που έχουν το σημάδι λάθους». Άρα ο χρήστης

θα πρέπει να συμπληρώσει τα στοιχεία του και να είναι αληθή για να διεξάγεται σωστά η διαδικασία της δημοπρασίας.

Τα στοιχεία είναι:

- Όνομα
- Επώνυμο
- Ηλεκτρονική διεύθυνση (e-mail)
- Σταθερό τηλέφωνο
- Διεύθυνση κατοικίας (όνομα οδού, ταχυδρομικός κώδικας, πόλη, χώρα)
- Όνομα χρήστη συστήματος

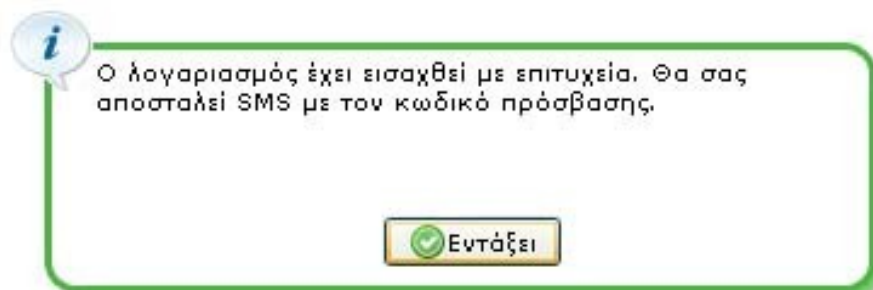
Αυτά τα πεδία ελέγχονται αν έχουν συμπληρωθεί. Επίσης, χρειάζεται το κινητό τηλέφωνο του χρήστη και το e-mail του χρήστη. Αυτά είναι απαραίτητα να συμπληρωθούν, γιατί εκεί θα έρθει ο κωδικός ενεργοποίησης του λογαριασμού του. Ακόμη καλό θα ήταν να επιλέξει ο χρήστης και την ζώνη ώρας όπου βρίσκεται η χώρα του. Όλοι γνωρίζουμε ότι στην δημοπρασία παίζουμε πολύ με τον χρόνο. Έτσι έχουμε βάλει μέσα στο σύστημα να μετατρέπονται οι ώρες στην ζώνη ώρας του κάθε χρήστη. Τέλος, θα πρέπει να επιλέξει ο χρήστης και το όνομα που θα χρησιμοποιεί μέσα στην εφαρμογή, το «παρατσούκλι» του δηλαδή. Σε αυτό το πεδίο, το σύστημα εκτός από το να ελέγξει αν υπάρχει τιμή, ελέγχει και αν υπάρχει το ίδιο όνομα από κάποιον άλλο χρήστη. Επίσης, θα πρέπει να διαβάσει και τους όρους χρήσης που βρίσκονται στα δεξιά, και να επιλέξει το κουτί «Αποδέχομαι τους όρους». Αν για κάποιο λόγο ο χρήστης επιλέξει όνομα χρήστη που υπάρχει ήδη βγαίνει μήνυμα λάθους, το οποίο φαίνεται στην παρακάτω εικόνα.

The screenshot shows a web browser window with a form titled 'Φόρμα για νέο λογαριασμό'. The form contains the following fields and values:

- Όνομα: [Empty]
- Όνομα χρήσης: [Empty]
- Επίθετο: Παναγιωτάκης
- Ηλεκτρονική Διεύθυνση: vaggos_p@yahoo.com
- Σταθερό Τηλέφωνο: 2310838552
- Κινητό Τηλέφωνο: 6947993904
- Διεύθυνση Κατοικίας: Λ. Επραστού
- Αριθμός Διεύθυνση Κατοικίας: 17
- Ταχυδρομικός Κώδικας: 54639
- Πόλη: [Empty]
- Θεσσαλονίκη
- Χώρα: Ελλάδα
- Ζώνη Ώρας: 2
- Όνομα χρήση για το σύστημα: vaggosupio

A yellow warning box with a triangle icon contains the text: "Παρακαλώ διαλέξτε άλλο όνομα χρήστη." Below the warning box is a green button labeled "Εντάξει". At the bottom right of the form, there are two buttons: "Εντάξει" and "Άκυρο".

Ο χρήστης πρέπει να αλλάξει το όνομα χρήστη και να επιλέξει άλλο όνομα χρήστη. Μόλις ολοκληρώσει την διαδικασία και είναι όλα εντάξει και σωστά, η διαδικασία έχει ολοκληρωθεί.




Μόλις εμφανιστεί αυτό το μήνυμα θα αποσταλεί ένα SMS στο κινητό του χρήστη με τον κωδικό πρόσβασης. Έτσι, πλέον ο χρήστης θα μπορεί να δει και τις υπόλοιπες λειτουργίες της εφαρμογής.

3.8.2 Κατηγορίες και Δημοπρασίες

Για αυτήν την λειτουργία δεν είναι απαραίτητο να είναι κανείς εγγεγραμμένος χρήστης. Φυσικά σε αυτήν την περίπτωση θα έχει μόνο την δυνατότητα να δει ανοιχτές ή που θα ανοίξουν προσεχώς δημοπρασίες χωρίς την δυνατότητα πονταρίσματος. Μόλις ενεργοποιήσει τη συγκεκριμένη λειτουργία, εμφανίζεται μια καινούργια καρτέλα στην οποία φαίνονται όλες οι κατηγορίες και οι δημοπρασίες. Στην εικόνα παρακάτω βλέπουμε ένα στιγμιότυπο από την λειτουργία αυτή.

The screenshot shows a web application interface. On the left, there is a sidebar with a search bar and a list of categories: Computer (0), Software (0), Games (0), Operating Systems (0), Hardware (0), Hard Drives (0), CPU (0), Cars (1), Electric Devices (0), T.V (0), and Books (0). The 'Cars' category is selected. On the right, there are two tabs: 'Ανοιχτές Δημοπρασίες' (Active Auctions) and 'Δημοπρασίες που θα ανοίξουν προσεχώς' (Auctions to be opened soon). The 'Ανοιχτές Δημοπρασίες' tab is active, showing a table with the following data:

Τίτλος	Κλείνει σε	Τελευταίο ποντάρισμα	Τιμή άμεσης πώλησης	Προβολές
 Ferrari	0 εβδομάδες, 3 ημέρες, 8 ώρες, 16 λεπτά	Δεν υπάρχουν προσφορές		1

Όπως παρατηρούμε στο αριστερό μέρος της καρτέλας που άνοιξε υπάρχει μια δενδρική μορφή η οποία αναπαριστά όλες τις κατηγορίες που υπάρχουν στο σύστημα δημοπρασιών. Ο αριθμός μέσα στην παρένθεση υποδεικνύει τον αριθμό ανοιχτών δημοπρασιών που υπάρχουν αυτή τη δεδομένη στιγμή στην κατηγορία αυτή. Πάνω από την δενδρική αναπαράσταση των κατηγοριών προσφέρεται η δυνατότητα γρήγορης αναζήτησης μια κατηγορίας. Δεν έχουμε παρά να γράψουμε μερικά γράμματα από τον τίτλο της κατηγορίας και να κάνουμε κλικ πάνω στο εικονίδιο με τον μεγεθυντικό φακό. Δίπλα από τις κατηγορίες υπάρχουν δύο καρτέλες. Η καρτέλα με τον τίτλο «Ανοιχτές Δημοπρασίες» παρουσιάζει όλες τις δημοπρασίες, φυσικά της επιλεγμένης κατηγορίας, που είναι στη διάθεση των χρηστών. Η δεύτερη καρτέλα με τίτλο «Δημοπρασίες που θα ανοίξουν

προσεχώς» εμφανίζει όλες τις δημοπρασίες, φυσικά πάλι της επιλεγμένης κατηγορίας, που προφανώς θα ανοίξουν προσεχώς.

Τίτλος	Θα ανοίξει	Τιμή εκκίνησης	Τιμή άμεσης πώλησης
hello2!!!	12/09/2007 στις 21:23:48	435	345

Έτσι για κάθε κατηγορία μπορούμε να δούμε δύο τύπους δημοπρασιών. Τις ανοιχτές, δηλαδή αυτές που περιμένουν από τους χρήστες του συστήματος να ανταποκριθούν σε πονταρίσματα και φυσικά αυτές που θα ανοίξουν σε λίγο (για παράδειγμα την ίδια ημέρα). Φυσικά στις μη ανοιχτές δημοπρασίες η δυνατότητα εισαγωγής στα αγαπημένα του χρήστη αλλά και του πονταρίσματος είναι απενεργοποιημένες. Αυτό όμως θα το δούμε αναλυτικότερα παρακάτω.

Ας δούμε τώρα τι γίνεται μόλις κάποιος χρήστης ο οποίος δεν είναι εγγεγραμμένος επιλέξει μια ανοιχτή δημοπρασία.

The screenshot shows a web browser window with a blue header containing 'Today', 'Κατηγορίες και δημοπρασίες', and 'Ferrari'. The main content area is titled 'Ferrari' and features a large image of a red Ferrari sports car. To the left of the main image is a vertical sidebar with navigation links: 'Γίνε μέλος', 'Κατηγορίες και δημοπρασίες', 'Νέα δημοπρασία', 'Οι δημοπρασίες μου', 'Τα πωτησιατά μου', and 'Αγαπημένα'. Below these links are input fields for 'Όνομα:' and 'Κωδικός:' with an 'Εισαγωγή' button. The main image area includes a vertical strip of four smaller images of the car from different angles. Below the main image, there are sections for 'Συνοπτική περιγραφή' (Super Car), 'Αναλυτική περιγραφή' (A fantastic Super Sports Car), and 'Πληροφορίες δημοπρασίας'. The information section contains the following details:

Κατάσταση δημοπρασίας:	1
Αρχική τιμή:	10000
Ρυθμός αύξησης προσφοράς:	1000
Άμεση πώληση:	30000
Άνοιξε:	13/09/2007 στις 12:02:31
Κλείνει:	25/09/2007 στις 20:00:00
Κλείνει σε:	0 εβδομάδες, 3 ημέρες, 8 ώρες, 14 λεπτά
Περιοχή:	b
Τρόπος αποστολής:	???? ????? ? ?????????
Τρόπος πληρωμής:	???? ????? ? ?????????

On the right side of the information section, there is a star icon and the text 'Εισαγωγή στα αγαπημένα'.

Όπως βλέπουμε στην παραπάνω εικόνα εμφανίζονται τα στοιχεία της δημοπρασίας. Στο επάνω μέρος φαίνονται όλες οι φωτογραφίες που έχει εισάγει ο χρήστης. Ακριβώς από κάτω υπάρχουν οι συνοπτική και αναλυτική περιγραφή της και μετά άλλες λεπτομέρειες όπως η αρχική τιμή, το βήμα αύξησης προσφοράς, το πόσο της άμεσης πώλησης, ημερομηνίες έναρξης και λήξης και τέλος τρόποι πληρωμής και αποστολής. Υπάρχει ένα κουμπί δεξιά που γράφει «Εισαγωγή στα αγαπημένα», αλλά είναι απενεργοποιημένο γιατί ο χρήστης δεν είναι εγγεγραμμένος ακόμα (ή δεν έχει κάνει εισαγωγή στο σύστημα). Θα αναλύσουμε παρακάτω τι κάνει το κουμπί αυτό, αν και είναι αρκετά προφανές.

Στοιχεία χρήστη μέλος **vaggouras**

αξιολόγηση: 0 (0) 0 (0) 0 (0)
 θετικές ουδέτες αρνητικές ψήφους

μπορεί να εισάγει δημοπρασίες

μπορεί να παίξει σε δημοπρασία

Μήνυμα στον vaggouras

Μέλη που έχουν κάνει προσφορά

Όνομα μέλους	Προσφορά	Ημερομηνία

Πως θα κάνετε προσφορές σε δημοπρασία

1. Εγγραφείτε στο e-auctions System AE, αν δεν το έχετε κάνει ήδη.
2. Διαβάστε την αξιολόγηση του πωλητή και τα σχόλια που έχουν αφήσει άλλα μέλη για αυτόν.
3. Διαβάστε καλά τα στοιχεία που αφορούν στον τρόπο πληρωμής και αποστολής του προϊόντος, καθώς και την περιγραφή του.
4. Αν έχετε οποιαδήποτε αμφισβήτηση, επικοινωνήστε με τον πωλητή πριν κάνετε την προσφορά σας.
5. Κάντε την προσφορά!

Αμέσως μετά τις λεπτομέρειες της δημοπρασίας παρουσιάζονται τα στοιχεία του χρήστη που έχει εισάγει την δημοπρασία στο σύστημα και τις αξιολογήσεις που έχει δεχθεί από άλλους χρήστες σε προηγούμενες δημοπρασίες του (παραπάνω εικόνα). Στα στοιχεία του χρήστη υπάρχει και ένα κουμπί με τίτλο «Μήνυμα στον XXXXXX» και το οποίο είναι απενεργοποιημένο σε αυτή τη φάση.

Πληροφορίες δημοπρασίας

Κατάσταση δημοπρασίας: 1

Αρχική τιμή: 10000

Ρυθμός αύξησης προσφοράς: 1000

Άμεση πώληση: 30000

Άνοιξε: 22/09/2007 στις 11:46:22

Κλείνει: 25/09/2007 στις 20:00:00

Κλείνει σε: 0 εβδομάδες, 3 ημέρες, 8 ώρες, 12 λεπτά

Περιοχή: b

Τρόπος αποστολής: ??? ???? ? ????????

Τρόπος πληρωμής: ??? ???? ? ????????

Εισαγωγή στα αγαπημένα

Στοιχεία χρήστη μέλος **vaggouras**

αξιολόγηση: 0 (0) 0 (0) 0 (0)
 θετικές ουδέτες αρνητικές ψήφους

μπορεί να εισάγει δημοπρασίες

μπορεί να παίξει σε δημοπρασία

Μήνυμα στον vaggouras

Μέλη που έχουν κάνει προσφορά

Όνομα μέλους	Προσφορά	Ημερομηνία	Η προσφορά μου
			Ο κωδικός σας <input type="text"/>
			Ποσό 10000
			<input type="button" value="Πόνταρε"/>

Όπως είναι προφανές, μέσω αυτής της λειτουργίας μπορεί ένας υποψήφιος αγοραστής να συνομιλήσει με τον πωλητή. Όπως αναφέραμε πιο πάνω, αυτή τη στιγμή βλέπει την δημοπρασία ένας μη εγγεγραμμένος χρήστης του συστήματος εξ ου και η απενεργοποίηση της υπηρεσίας. Παρακάτω θα δούμε όμως την λειτουργία αυτή αναλυτικότερα. Τέλος η τελευταία περιοχή που βλέπουμε στην παραπάνω εικόνα παρουσιάζει όλα τα πονταρίσματα που έχει γίνει για την δημοπρασία αυτή από άλλους χρήστες. Φυσικά πρώτος στην σειρά είναι αυτός με το μεγαλύτερο

ποσό.

Τώρα αν κάποιος είναι μέλος και μπει στο σύστημα με τον κωδικό του, θα ενεργοποιηθούν όλες οι προστιθέμενες λειτουργίες της δημοπρασίας. Η πιο πάνω εικόνα δείχνει αυτήν την κατάσταση. Έτσι αν ο χρήστης πατήσει στο κουμπί «Εισαγωγή στα αγαπημένα» θα εμφανιστεί ένα πλαίσιο ίδιο με αυτό της παρακάτω εικόνας.



Εισαγωγή στα αγαπημένα

Θέλω να παρακολουθώ την δημοπρασία:

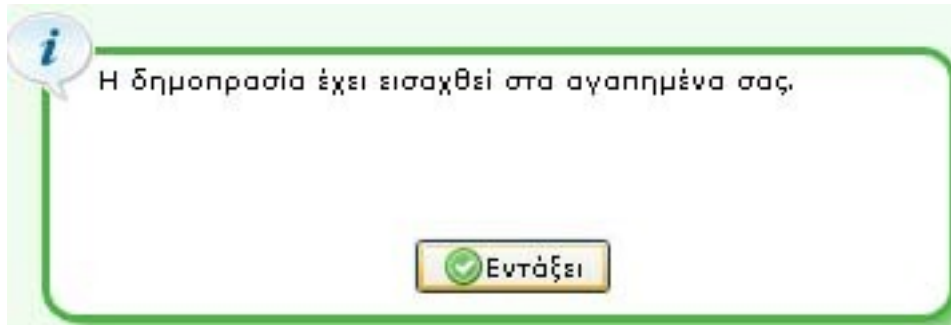
- στο e-mail μου
- από το σύστημα ανακοινώσεων
- στον αριθμό του κινητού με sms

Όταν κάποιος χρήστης θελήσει να προσθέσει μια δημοπρασία στα αγαπημένα του σημαίνει και ότι θέλει να παρακολουθεί την εξέλιξή της. Έτσι, μπορεί να ζητήσει από το σύστημα να τον ειδοποιεί με τους εξής τρόπους:

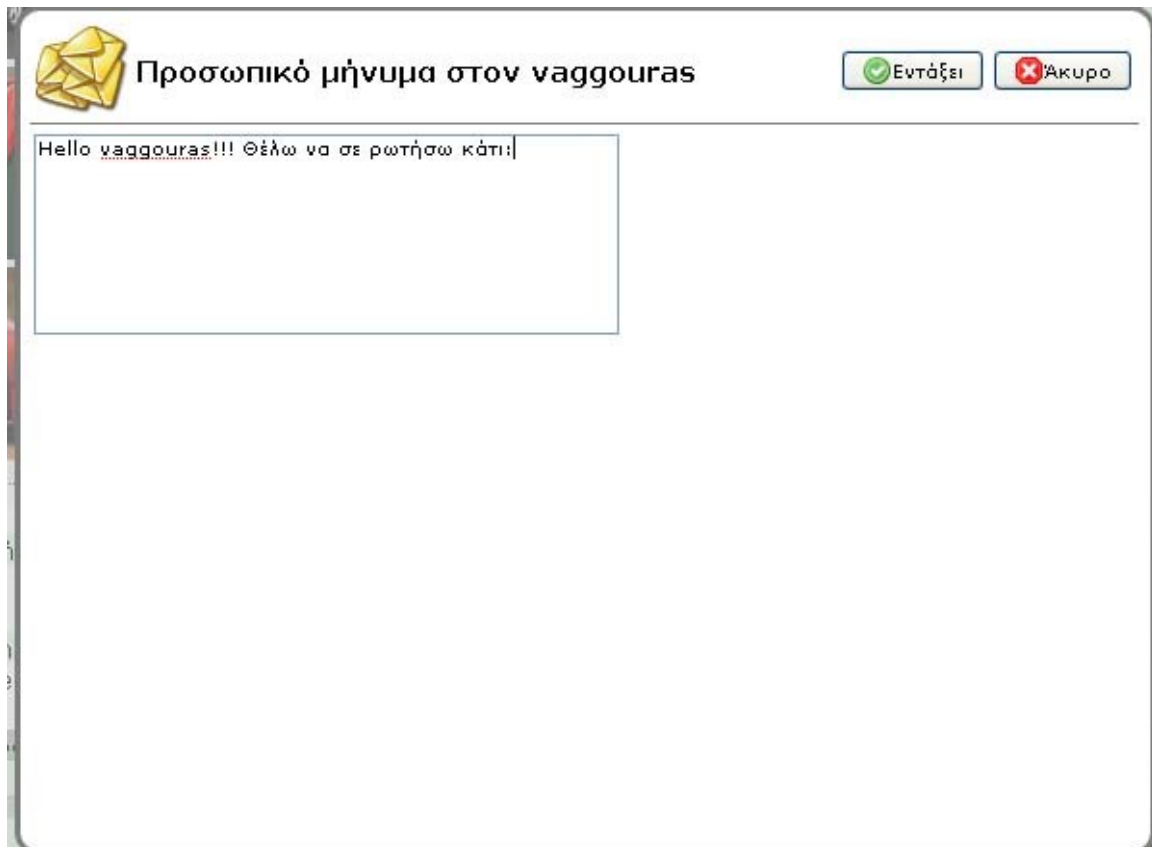
- στην ηλεκτρονική του διεύθυνση (e-mail)
- από τη λειτουργία ανακοινώσεων του συστήματος
- στο κινητό του τηλέφωνο με γραπτό μήνυμα (sms)

Αφού κάνει τις επιλογές του ο χρήστης πατάει στο κουμπί εντάξει και περιμένει να δει το παρακάτω πλαίσιο διαλόγου για να βεβαιωθεί πως έχει καταχωρηθεί στα επιτυχώς στα αγαπημένα

του.

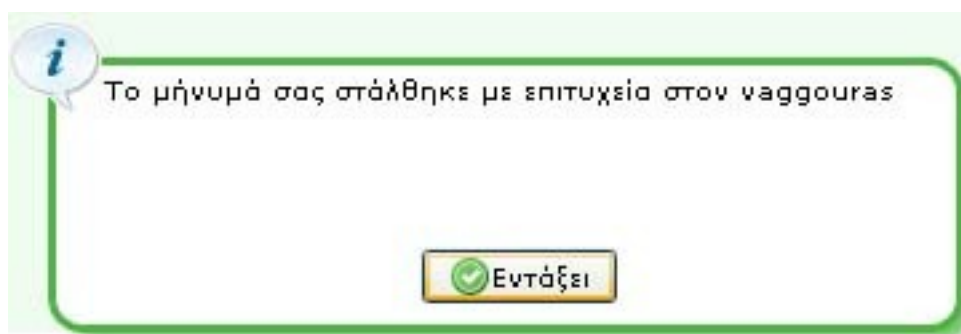


Ενεργοποιημένο είναι επίσης και το κουμπί «Μήνυμα στον XXXXXX». Μόλις ο χρήστης επιλέξει το κουμπί αυτό θα μπορέσει να στείλει κάποιο μήνυμα στον χρήστη που έχει εισάγει την δημοπρασία. Στην παρακάτω εικόνα φαίνεται ο διάλογος αποστολή προσωπικού μηνύματος.



Παρατηρούμε ότι υπάρχει ένα πεδίο στο οποίο ο χρήστης μπορεί να γράψει το μήνυμά που θέλει να στείλει στον χρήστη που δημοσίευσε την δημοπρασία. Αφού τελειώσει το μήνυμά του και πατήσει το κουμπί "Εντάξει" και όλα πάνε καλά, το παρακάτω μήνυμα της εικόνας θα του επιβεβαιώσει την επιτυχή του αποστολή.

Μαρκόπουλος Γρηγόριος (grimark@it.teithe.gr)
Παναγιωτάκης Ευάγγελος (evpanag@it.teithe.gr)
Μηχανικοί Πληροφορικής



3.8.3 Δημιουργία νέας δημοπρασίας (Νέα δημοπρασία)

Για να προσθέσει κάποιος μια νέα δημοπρασία στο σύστημα δημοπρασιών, θα πρέπει πρώτα να γίνει μέλος του (Βλ. Δημιουργία νέου λογαριασμού). Αφού δημιουργήσει ένα λογαριασμό και κάνει εισαγωγή στο σύστημα, πατώντας το κουμπί «Νέα Δημοπρασία» θα ανοίξει μια νέα καρτέλα με την φόρμα εισαγωγής νέας δημοπρασίας που βλέπουμε στην παρακάτω εικόνα.

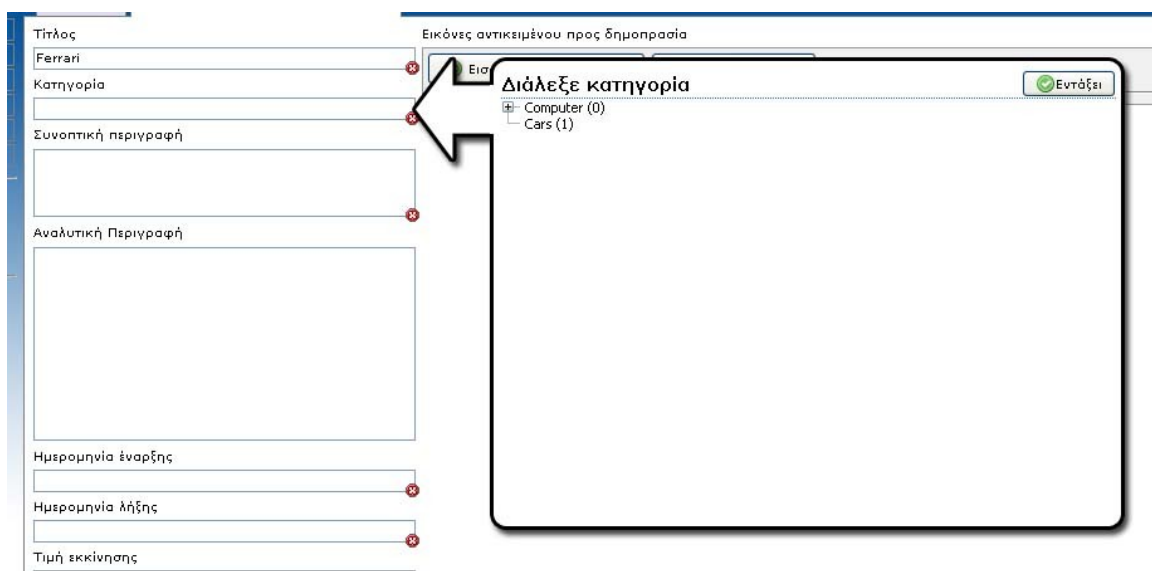
The screenshot shows a web browser window with the title "Φόρμα νέας δημοπρασίας". The page is divided into a left sidebar with navigation links and a main content area. The main area contains a form for creating a new auction. The form fields include: "Τίτλος" (Title), "Κατηγορία" (Category), "Συνοπτική περιγραφή" (Summary description), "Αναλυτική Περιγραφή" (Detailed description), "Ημερομηνία έναρξης" (Start date), "Ημερομηνία λήξης" (End date), "Τιμή εκκίνησης" (Starting price), "Θέμα αύξησης προσφορών" (Bid increment), "Επιθυμητή τιμή πώλησης" (Desired selling price), "Επιθυμητή τιμή άμεσης πώλησης" (Desired immediate selling price), "Τρόπος αποστολής προϊόντος" (Product shipping method), and "Τρόπος πληρωμής" (Payment method). There are also buttons for "Εισαγωγή νέου αρχείου" (Upload new file) and "Διαγραφή όλων" (Delete all). A warning message box is visible in the center-right, stating "Παρακαλώ συμπληρώστε/διορθώστε όλα τα πεδία που έχουν σημάδι λάθους." (Please complete/correct all fields with error signs). At the bottom right, there are buttons for "Εντάξει" (OK) and "Άκυρο" (Cancel).

Όπως παρατηρούμε στην εικόνα και σε αυτή την φόρμα υπάρχουν κάποια πεδία τα οποία είναι απαραίτητα να συμπληρωθούν. Τα πεδία αυτά είναι:

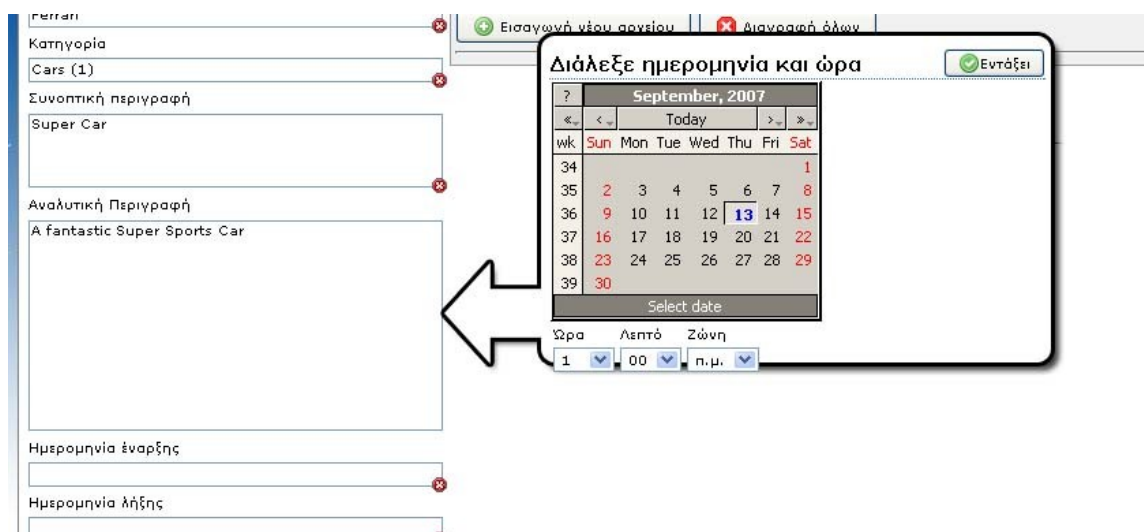
Μαρκόπουλος Γρηγόριος (grimark@it.teithe.gr)
Παναγιωτάκης Ευάγγελος (evranag@it.teithe.gr)
Μηχανικοί Πληροφορικής

- Τίτλος
- Οι ημερομηνίες έναρξης και λήξης
- Η τιμή εκκίνησης
- Το βήμα αύξησης προσφορών
- Η συνοπτική περιγραφή
- Η κατηγορία της δημοπρασίας
- Τρόπος πληρωμής
- Τρόπος αποστολής

Ας δούμε όμως τα στοιχεία ένα ένα. Πρώτα απ' όλα είναι ο τίτλος της δημοπρασίας τον οποίον τον δίνει ο χρήστης όπως επιθυμεί εκείνος. Στη συνέχεια είναι η επιλογή κατηγορίας. Μόλις ο χρήστης επιλέξει το πεδίο αυτό εμφανίζεται ένας διάλογος όπως αυτός της παρακάτω εικόνας.

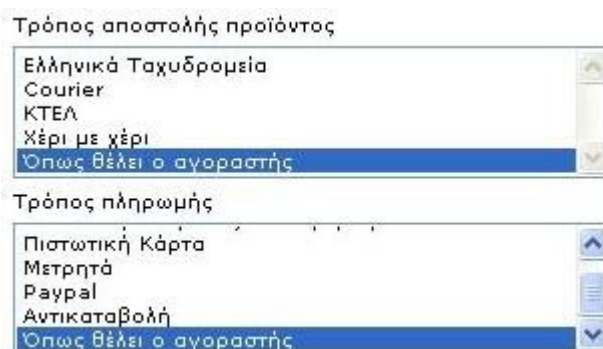


Στον διάλογο αυτό βλέπουμε πως εμφανίζονται όλες οι διαθέσιμες κατηγορίες που έχει καταχωρημένες το σύστημα τη δεδομένη χρονική στιγμή. Ο χρήστης μέσω αυτής της δενδρικής παρουσίασής τους μπορεί να επιλέξει αυτήν που ταιριάζει καλύτερα στο προϊόν του. Στη συνέχεια μπορεί να γράψει μια συνοπτική περιγραφή του προϊόντος όπως και μια πιο αναλυτική, χωρίς αυτό να είναι απαραίτητο (μιλάμε για την αναλυτική περιγραφή). Μετά έχουμε τις ημερομηνίες έναρξης και λήξης. Μόλις ο χρήστης εστιάσει σε κάποιο από αυτά τα δύο πεδία, εμφανίζεται ένας διάλογος επιλογής πλήρους ημερομηνίας και ώρας έτσι όπως παρουσιάζεται στην παρακάτω εικόνα.



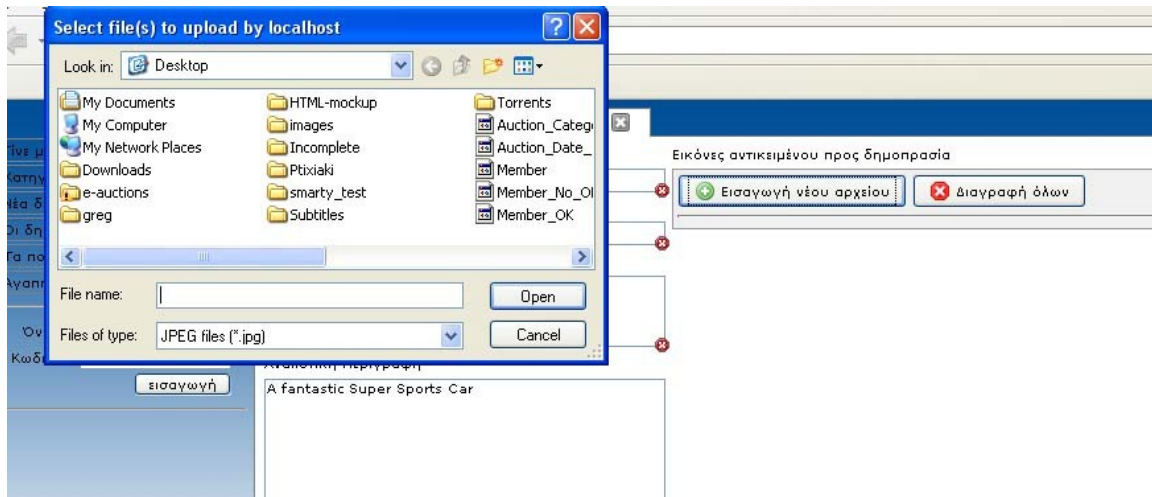
Όπως βλέπουμε εμφανίζεται ένα ημερολόγιο, στο οποίο μπορεί να επιλέξει την ημέρα, τον μήνα, την χρονιά και φυσικά την ώρα με ακρίβεια 15λέπτου κατά την οποία επιθυμεί να ανοίξει η δημοπρασία του. Στο πεδίο της ημερομηνίας λήξης εμφανίζεται ακριβώς ο ίδιος διάλογος του ημερολογίου. Στην επιλογή της ημερομηνίας και ώρας λήξης γίνεται έλεγχος με αυτόν της έναρξης σε περίπτωση που ο χρήστης εκ παραδρομής επέλεξε μια προγενέστερη.

Το αμέσως επόμενο πεδίο είναι η τιμή εκκίνησης. Εκεί ο χρήστης δίνει την τιμή από την οποία επιθυμεί να ξεκινήσουν οι προσφορές. Στη συνέχεια επιλέγει το βήμα αύξησης αυτών. Βέβαια μπορεί να δηλώσει και μια επιθυμητή τιμή με την οποία θα ήταν ικανοποιημένος όπως επίσης και μια τιμή άμεσης πώλησης. Αυτά τα δύο πεδία όμως δεν είναι υποχρεωτικά. Στο τέλος ο χρήστης πρέπει να επιλέξει από το μενού τον τρόπο πληρωμής και τον τρόπο αποστολής του προϊόντος. Το μενού αυτό φαίνεται παρακάτω εικόνας.



Τέλος ο χρήστης έχει την δυνατότητα να προσθέσει ως και τέσσερις εικόνες του προϊόντος του για την καλύτερη πώλησή του. Δίπλα από τα πεδία προς συμπλήρωση βρίσκεται το control για την

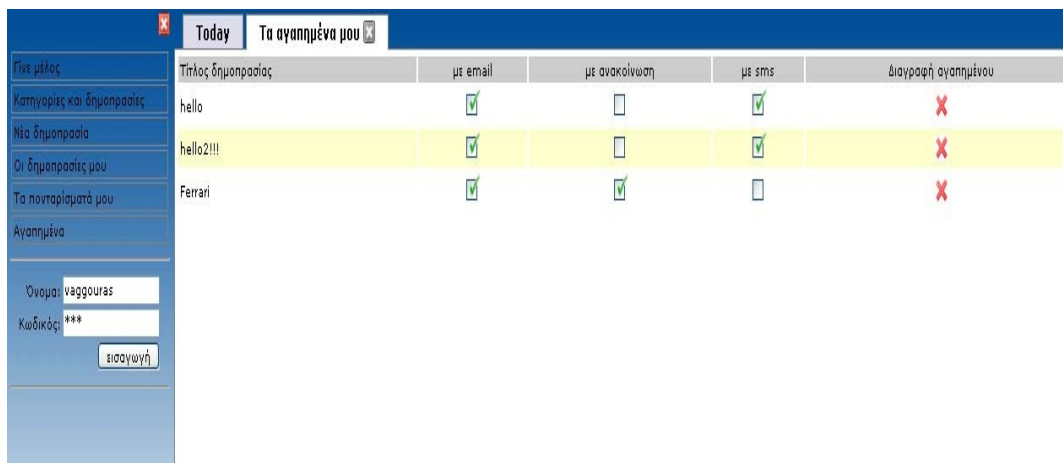
εισαγωγή των εικόνων. Μόλις ο χρήστης πατήσει το κουμπί εισαγωγή αρχείου εμφανίζεται ο διάλογος επιλογής αρχείων που βλέπουμε στην κάτω εικόνα.



Έχοντας πατημένο το ctrl ο χρήστης μπορεί να επιλέξει ταυτόχρονα και τις τέσσερις φωτογραφίες που επιθυμεί. Με αυτόν το τρόπο ο χρήστης αποφεύγει την βαρετή διαδικασία αναζήτησης εικόνων μέσα στο τοπικό σύστημα αρχείων του χρήστη.

3.8.4 Αγαπημένα

Σε αυτή την λειτουργία θα δούμε πως μπορεί ο χρήστης να διαχειρισθεί τις δημοπρασίες που έχει εισάγει στα αγαπημένα. Μόλις ο χρήστης πατήσει το κουμπί «Αγαπημένα» θα ανοίξει μια καινούργια καρτέλα η οποία θα του δείχνει τις δημοπρασίες που έχει εισάγει στα αγαπημένα του και πως επιθυμεί να ενημερώνεται για αυτές. Στην παρακάτω εικόνα βλέπουμε πως παρουσιάζονται στο χρήστη οι δημοπρασίες που έχει εισάγει στα αγαπημένα του.



The screenshot shows a web application interface with a navigation menu on the left and a main content area. The main content area has a tabbed interface with 'Today' and 'Τα αγαπημένα μου' (My Favorites) selected. Below the tabs is a table with columns: 'Τίτλος δημοπρασίας' (Auction Title), 'με email' (via email), 'με ανακοίνωση' (via announcement), 'με sms' (via sms), and 'Διαγραφή αγαπημένου' (Delete favorite). The table contains three rows: 'hello', 'hello2!!!', and 'Ferrari'. The 'hello' and 'hello2!!!' rows are highlighted in yellow. Below the table is a login form with fields for 'Όνομα: vaggouras' and 'Κωδικός: ***', and a 'Εισαγωγή' (Login) button.

Τίτλος δημοπρασίας	με email	με ανακοίνωση	με sms	Διαγραφή αγαπημένου
hello	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
hello2!!!	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ferrari	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Βλέπουμε ότι υπάρχει ο τίτλος της δημοπρασίας και δίπλα στα κουτάκια είναι επιλεγμένοι οι τρόποι ενημέρωσης. Μπορεί να αλλάξει τον τρόπο ενημέρωσης για κάποια δημοπρασία απλώς επιλέγοντας ή μη επιλέγοντας το αντίστοιχο κουτάκι. Φυσικά, αν πατήσει πάνω στον τίτλο της δημοπρασίας μπορεί να ξαναδεί τα αναλυτικά της στοιχεία σε καινούργια καρτέλα, χωρίς να χάνεται η καρτέλα των αγαπημένων. Στο δεξί μέρος υπάρχει το κουμπί «Διαγραφή αγαπημένου». Αν ο χρήστης επιλέξει το κουμπί αυτό μπορεί να σβήσει την δημοπρασία από τον κατάλογο των αγαπημένων του. Αξίζει να σημειωθεί ότι μόλις λήξει μια δημοπρασία και κλείσει, αυτόματα διαγράφεται από τα αγαπημένα, γιατί πλέον δεν είναι ενεργεί, οπότε ο χρήστης δεν χρειάζεται να ενημερώνεται πλέον.

Παράρτημα Α: Βιβλιογραφία

- Ελληνική wikipedia, el.wikipedia.org στις 10/04/2007
- Αγγλική wikipedia, en.wikipedia.org στις 10/04/2007
- Βιβλίο Ajax In Action των Dave Crane, Eric Pascarello και Darren James, εκδόσεις Manning Publications, ISBN 1-932394-61-3

Ευχαριστήρια

Την παρούσα πτυχιακή εργασία θα ήθελα να την αφιερώσω στον πατέρα μου, Γεράσιμο και στην μητέρα μου, Σοφία, όπου πέρα από την οικονομική τους ενίσχυση, η ψυχολογική τους υποστήριξη και η νοητή τους παρουσία ήταν πάντα πλούσια και παντού υπαρκτή σε όλα τα χρόνια της φοιτητικής μου ζωής.

Σας ευχαριστώ πολύ...

Μαρκόπουλος Γρηγόριος

Την παρούσα πτυχιακή θα ήθελα να την αφιερώσω στους γονείς μου Γιώργο και Ελένη για όλη τους τη βοήθεια που μου προσέφεραν όλα αυτά τα χρόνια, ψυχολογική και οικονομική και που είναι κοντά μου συνέχεια.

Σας ευχαριστώ από την καρδιά μου

Παναγιωτάκης Ευάγγελος