



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

"Διδακτική της Τεχνολογίας και Ψηφιακά Συστήματα"

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΚΟΙΝΩΝΙΚΗΣ
ΔΙΚΤΥΩΣΗΣ ΓΙΑ ΤΗΝ ΠΑΡΑΚΟΛΟΥΘΗΣΗ
ΤΗΣ ΧΡΗΜΑΤΙΣΤΗΡΙΑΚΗΣ ΑΓΟΡΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΛΕΥΡΑ ΒΑΣΙΛΙΚΗ

ΑΜ :ΜΕ 09045

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΝΙΚΗΤΑΣ-ΜΑΡΙΝΟΣ ΣΓΟΥΡΟΣ

ΜΑΙΟΣ 2013, Αθήνα

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον καθηγητή του τμήματος Ψηφιακών Συστημάτων κ. Νικήτα-Μαρίνο Σγούρο για την πολύτιμη βοήθεια του σε όλη την διάρκεια της εκπόνησης της συγκεκριμένης Διπλωματικής εργασίας.

Η εργασία μου είναι αφιερωμένη στη μνήμη του πατέρα μου ο οποίος συνεχίζει και με ωθεί σε κάθε μου σημαντικό βήμα.

Περιεχόμενα

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ ΚΑΙ ΠΙΝΑΚΩΝ	7
Εικόνες	7
Πίνακες.....	7
1. ΕΙΣΑΓΩΓΗ	8
1.2 Δομή εργασίας	8
1.3 Σκοπός	9
2. ΚΟΙΝΩΝΙΚΑ ΜΕΣΑ ΔΙΚΤΥΩΣΗΣ	10
2.1 Ιστορία κοινωνικών μέσων δικτύωσης.....	10
2.2 Τύποι κοινωνικών μέσων δικτύωσης.....	17
3. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ	20
3.1 Αρχιτεκτονική δύο επιπέδων	20
3.1.1 Πλεονεκτήματα τεχνολογίας	22
3.1.2 Μειονεκτήματα Τεχνολογίας	23
4. ΔΟΜΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	24
4.1 Customer.....	24
4.2 Customer_Stock	25
4.3 Stock.....	26
4.4 Voice Message	27
4.5 Message	28
5 ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	31
5.1 Εργαλεία ανάπτυξης	31
Eclipse.....	31
Netbeans	32
Microsoft Sql Management Studio	32
Jogl	33
Φυλλομετρητής δικτύου.....	33
5.2 Τεχνολογίες Λογισμικού	34
5.2.1 Java.....	34
5.2.2 Sql.....	36
5.2.3 OpenGL.....	37

5.3 Υλοποίηση-Κώδικας	42
5.3.1 Πακέτα κλάσεις πηγαίου κωδικα.....	42
5.4 Τρόπος υλοποίησης	54
5.4.1 Είσοδος Χρήστη.....	54
5.4.2 Εγγραφή Χρήστη	55
5.4.3 Τελευταίο Διάγραμμα	55
5.4.4 Δημιουργία νέου διαγράμματος.....	56
5.4.5 Εμφάνιση διαγράμματος άλλου χρήστη	57
5.4.6 Ηχογράφηση νέου μηνύματος.....	58
5.4.7 Δημιουργια γραπτού μηνύματος.....	60
5.4.8 Ακρόαση ηχητικών μηνυμάτων	61
6. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ	63
<i>Εισαγωγή Χρήστη</i>	65
<i>Εμφάνιση Προφίλ</i>	66
<i>Δημιουργία νέου διαγράμματος</i>	68
<i>Μήνυμα στη περιοχή σχολίων</i>	69
<i>Ακρόαση ηχητικών μηνυμάτων</i>	70
<i>Αποστολή ηχητικών μηνυμάτων</i>	71
7. ΜΕΛΛΟΝΤΙΚΕΣ ΕΞΕΛΙΞΕΙΣ	73
Βιβλιογραφία	75

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ ΚΑΙ ΠΙΝΑΚΩΝ

Εικόνες

Εικόνα 1-Εξέλιξη των social media από το 1977 και μετά.....	12
Εικόνα 2-Πλήθος social media	15
Εικόνα 3-Δημοφιλή κοινωνικά δίκτυα ανα χώρα.....	16
Εικόνα 4-Αρχιτεκτονική δυο επιπέδων.....	21
Εικόνα 5 –Διάγραμμα E-R Βάσης Δεδομένων.	30
Εικόνα 6-Opengl pipeline.....	39
Εικόνα 7 –Πακέτα κώδικα.....	42
Εικόνα 8 – Αντικείμενα (κλάσεις) client-server	43
Εικόνα 9- Ηχογράφηση ηχητικού μηνύματος.....	58
Εικόνα 10- Παράθυρο ακρόασης ηχητικών μηνυμάτων	62
Εικόνα 11-Φόρμα Εγγραφής Χρήστη.....	64
Εικόνα 12- Φόρμα Εισαγωγής Χρήστη	65
Εικόνα 13- Αρχική Εικόνα Εφαρμογής.....	66
Εικόνα 14- Ενδεικτική αρχική εικόνα προφίλ.....	67
Εικόνα 15 -Φόρμα Επιλογής Χρήστη	67
Εικόνα 16- Φόρμα Δημιουργίας Νέου Διαγράμματος	69
Εικόνα 17 -Επιλογή χρονικού περιορισμού εμφανισης μηνυματος	70
Εικόνα 18- Ακρόαση ηχητικών μηνυμάτων	71
Εικόνα 19 -Ηχογράφηση ηχητικού μηνύματος.....	72

Πίνακες

Πίνακας 1-Πίνακας Customer Βάσης Δεδομένων	25
Πίνακας 2-Πίνακας Customer_Stock Βάσης Δεδομένων	26
Πίνακας 3-Πίνακας Stock Βάσης Δεδομένων	27
Πίνακας 4-Πίνακας VoiceMessage Βάσης Δεδομένων.....	28
Πίνακας 5-Πίνακας Message Βάσης Δεδομένων.....	29

1. ΕΙΣΑΓΩΓΗ

Η κοινωνική δικτύωση έχει μπει για τα καλά στη ζωή μας από την περίοδο εμφάνισης της δεύτερης γενιάς διαδικτύου. Η γενιά αυτή δίνει στον χρήστη την δυνατότητα πρόσβασης στη δημοσίευση περιεχομένων του, καθώς και στην διαδραστικότητα με τις δημοσιεύσεις άλλων χρηστών. Έτσι δημιουργείτε μια εικονική διαδικτυακή κοινότητα με τα μέλη της να είναι μεταξύ τους φίλοι, συνεργάτες ή απλά να είναι άτομα με κοινά ενδιαφέροντα. Στη παρούσα πτυχιακή εργασία γίνεται προσπάθεια δημιουργίας μίας εφαρμογής κοινωνικής δικτύωσης η οποία να έχει ως κύριο λόγο ύπαρξης την παρακολούθηση των χρηματιστηριακών μετοχών. Το έγγραφο περιλαμβάνει τη περιγραφή μίας εφαρμογής δύο επιπέδων και γίνεται λεπτομερή περιγραφή των 3d γραφικών της , καθώς και όλων των δυνατοτήτων που παρέχονται στο χρήστη μετά την εγγραφή του.

1.2 Δομή εργασίας

Η παρούσα εργασία αποτελείται από επτά κεφάλαια.

Το πρώτο κεφάλαιο αποτελεί την εισαγωγή της εργασίας παρουσιάζεται ο σκοπός της συγγραφής της και επίσης δίνονται συνοπτικά τα περιεχόμενα των κεφαλαίων.

Σο δεύτερο κεφάλαιο γίνεται εισαγωγή στο θεωρητικό υπόβαθρο των κοινωνικών μέσων δικτύωσης , παρουσιάζεται η ιστορική εξέλιξη τους καθώς και οι τύποι που έχουν δημιουργηθεί μέχρι και σήμερα.

Στο τρίτο κεφάλαιο παρουσιάζεται η αρχιτεκτονική του συστήματος ποια έχει επιλεγεί για το συγκεκριμένο σύστημα και για ποιο λόγο. Αναφέρονται τα πλεονεκτήματα από την επιλογή αυτή αλλά και ποια μειονεκτήματα έχει τα οποία μπορεί να δημιουργήσουν προβλήματα στο μέλλον.

Στο τέταρτο κεφάλαιο γίνεται η παρουσίαση της βάσης δεδομένων του συστήματος. Γίνεται αναλυτική περιγραφή των πινάκων και των στοιχείων της καθώς και το e-r διάγραμμα της.

Στο επόμενο κεφάλαιο το πέμπτο, περιγράφεται αναλυτικά το πώς έγινε η υλοποίηση του συστήματος. Αναφέρονται τα εργαλεία ανάπτυξης , οι τεχνολογίες λογισμικού, τα κυριότερα σημεία του κώδικα καθώς και ο τρόπος υλοποίησης των λειτουργιών του προγράμματος.

Στο έκτο κεφάλαιο παρουσιάζεται ένα σύντομο εγχειρίδιο χρήσης με εικόνες και βασικά βήματα λειτουργίας του συστήματος.

Τέλος στο έβδομο και τελευταίο κεφάλαιο γίνεται η αξιολόγηση της εργασίας και παραθέτονται μελλοντικές βελτιώσεις που θα μπορούσαν να υπάρξουν.

1.3 Σκοπός

Στην παρούσα πτυχιακή εργασία μελετούνται τα κοινωνικά μέσα δικτύωσης και αναλύονται οι ανάγκες τη σημερινής αγοράς. Πρωταρχικός σκοπός είναι η δημιουργία μιας εφαρμογής η οποία να παρέχει στο χρήστη μία ασφαλή και γρήγορη πρόσβαση σε οικονομικές μετοχές με σκοπό τη σωστή διαχείριση τους. Επίσης προσφέρει την άμεση επικοινωνία ανθρώπων με κοινό ενδιαφέρον το χρηματιστήριο και την ανταλλαγή απόψεων. Έτσι οι στόχοι είναι να μελετηθεί η βιβλιογραφία να κατανοηθούν οι βασικές έννοιες ,να σχεδιαστεί ένα σωστά δομημένο κοινωνικό οικονομικό μέσο δικτύωσης και τέλος να γίνει η υλοποίηση και η τεκμηρίωση του.

2. ΚΟΙΝΩΝΙΚΑ ΜΕΣΑ ΔΙΚΤΥΩΣΗΣ

Με τον όρο «κοινωνικό δίκτυο» (social network) εννοούμε μια κοινωνική δομή αποτελούμενη από ανθρώπους ή οργανισμούς που λέγονται “κόμβοι” (nodes) και οι οποίοι μοιράζονται μεταξύ τους κοινά ενδιαφέροντα, επικοινωνώντας με τη βοήθεια του διαδικτύου. Τα κοινωνικά δίκτυα δεν έχουν γεωγραφικούς περιορισμούς και συνήθως παρουσιάζουν μεγάλη αύξηση των μελών τους. Με τον ορισμό αυτό είναι παραπάνω από σαφές ότι ένα κοινωνικό δίκτυο, όπως υπαινίσσεται άλλωστε και το όνομά του, έχει κοινωνικές προεκτάσεις και θέτει σε αμφισβήτηση την άποψη ότι βασική προϋπόθεση της ύπαρξης κοινωνίας είναι η φυσική παρουσία των μελών της.

2.1 Ιστορία κοινωνικών μέσων δικτύωσης

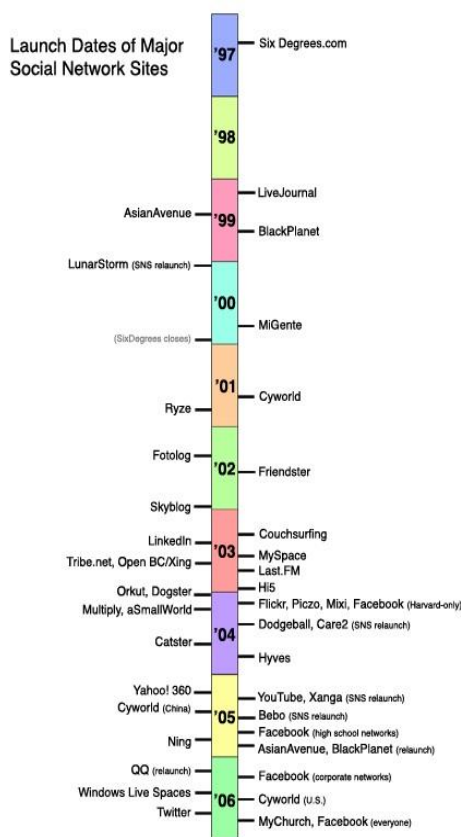
Η επικοινωνία των ανθρώπων μεταξύ τους δεν ήταν ποτέ τόσο εύκολη, οικονομική και προσιτή όσο είναι σήμερα και αυτό λόγω των κοινωνικών δικτύων. Πλέον οι κοινωνικές υπηρεσίες έχουν εξελιχτεί σε μια ηλεκτρονική βιομηχανία που πέρα από την επικοινωνία των ανθρώπων, χρησιμεύει για τη διαφήμιση εταιρειών αφού με το πάτημα ενός κουμπιού υπάρχει πρόσβαση σε εκατομμύρια χρήστες του διαδικτύου.

Οι πρώτες προσπάθειες για κοινωνικές υπηρεσίες έγιναν όταν η εξέλιξη της τεχνολογίας το επέτρεψε και αυτό συνέβη το 1971 όταν ανταλλάχτηκε το πρώτο ηλεκτρονικό μήνυμα μεταξύ δυο υπολογιστών που είχαν τοποθετηθεί ακριβώς δίπλα ο ένας με τον άλλο. Όμως οι πρώτες ηλεκτρονικές κοινωνίες εμφανίστηκαν αργότερα και πιο συγκεκριμένα το 1980. Ο Tom Truscott και Jim Ellis μεταπτυχιακοί φοιτητές του πανεπιστημίου Duke University ίδρυσαν το USENET ένα διαδικτυακό σύστημα συζήτησης το οποίο χρησιμοποιούσαν μεταξύ τους πανεπιστήμια και κυβερνητικές οργανώσεις για να γίνεται άμεση ανταλλαγή απόψεων μέσω μηνυμάτων. Επέτρεπε στους εγγεγραμμένους χρήστες να λαμβάνουν και να στέλνουν μηνύματα μέσα σε δημιουργημένες ενότητες γνωστές και ως ενότητες πληροφόρησης.

Για μια δεκαετία περίπου η εξέλιξη των κοινωνικών δικτύων έπεσε λόγω της έρευνας των εταιρειών όπου ετοίμαζαν την μετέπειτα πρόοδο των υπολογιστών και έτσι λίγα ήταν τα Site που έκαναν προσπάθεια για την επικοινωνία των ανθρώπων. Τα πιο σημαντικά από αυτά ήταν το Geocities(1994) και αμέσως μετά το Theglobe.com(1994) και το Tribod (1995). Μέσω αυτών οι χρήστες είχαν την ελευθερία να δημοσιοποιήσουν τις προσωπικές τους εμπειρίες και να αλληλεπιδράσουν με άλλους χρήστες που είχαν κοινά ενδιαφέροντα. Πάρα πολύ σημαντική είναι επίσης η δημιουργία του AOL Messenger το Μάιο του 1997 όπου πρόσφερε για πρώτη φορά σαν υπηρεσία την άμεση ανταλλαγή μηνυμάτων(chat) που αργότερα ενσωματώθηκε σε όλες σχεδόν τις σύγχρονες υπηρεσίες δικτύωσης.

Το 1997 είναι η πιο σημαντική χρονιά αφού δημιουργήθηκε το SixDegrees.com, το οποίο αποτελεί την αρχή των κοινωνικών δικτύων με τη μορφή που εμφανίζονται σήμερα και είναι το πρώτο αναγνωρισμένο Site κοινωνικής δικτύωσης. Επέτρεπε στους χρήστες τη δημιουργία δικού τους προσωπικού προφίλ το οποίο ενημέρωναν με προσωπικά χαρακτηριστικά και λίστα φίλων την οποία οποιαδήποτε στιγμή μπορούσαν να διατρέξουν. Μπορεί μέχρι τότε όλες αυτές τις υπηρεσίες να τις πρόσφεραν άλλες ηλεκτρονικές σελίδες, αλλά ήταν η πρώτη φορά που μαζεύτηκαν όλες μαζί σε μια. Εκείνη την εποχή προβαλλόταν σαν ένα εργαλείο που θα έδινε τη δυνατότητα στους χρήστες να επικοινωνούν και να στέλνουν μηνύματα σε άλλους και για αυτό το λόγο είχε προσελκύσει εκατομμύρια χρήστες. Παρόλη όμως την επιτυχία που είχε δεν κατάφερε να έχει κέρδη και έτσι το 2000 έκλεισε. Από το 1997 έως το 2001 ακολούθησαν εφαρμογές που δεν έκαναν σε καμία περίπτωση την επιτυχία του SixDegrees.com με πιο αξιόλογες προσπάθειες να είναι οι Black Planet, Mi Gente, CyWorld, Lunar Storm, Asian Avenue, LiveJournal κ.ά.

Στην εικόνα 1 φαίνεται η εξέλιξη των κοινωνικών δικτύων μέχρι σήμερα.



Εικόνα 1-Εξέλιξη των social media από το 1977 και μετά.

Το 2002 εμφανίστηκε το FriendSter το οποίο είχε ως κύριο σκοπό τον ανταγωνισμό του με ένα δημοφιλές site μέχρι τότε το Math.com, άλλα περιείχε βασικές καινοτομίες σχετικά με την οπτική γωνία που έβλεπε τις σχέσεις των χρηστών του. Ενώ ως τότε όλα τα μέσα κοινωνικής δικτύωσης επικεντρώνονταν στη σχέσεις άγνωστων ανθρώπων μεταξύ τους το FriendSter έδωσε βάρος στις σχέσεις των χρηστών με τους φίλους των φίλων του. Αυτό συνέβη με τη λογική ότι ήταν πολύ πιο εύκολο να αναπτυχθούν ρομαντικές σχέσεις μεταξύ όχι και τόσο άγνωστων ανθρώπων μεταξύ τους, αφού είχαν ένα κοινό φίλο τουλάχιστον, παρά μεταξύ εντελώς αγνώστων. Η εφαρμογή αρχικά είχε πολύ μεγάλη επιτυχία ανάμεσα στους bloggers και στην gay κοινότητα αλλά πολύ γρήγορα είχε τέτοια απήχηση που μέσα σε μόνο 6 μήνες έφτασε τα 3 εκατομμύρια χρήστες. Παρόλα αυτά ούτε το FriendSter κατάφερε να εξαπλωθεί

αλλά γνώρισε σύντομα τη παρακμή, αφού το υλικό τους επίπεδο δεν είχε την αντοχή να φιλοξενήσει τόσο μεγάλο πλήθος χρηστών.

Από το FriendSter και έπειτα παρά πολλοί έκαναν προσπάθειες δημιουργίας μιας επιτυχημένης εφαρμογής κοινωνικής δικτύωσης σπαταλώντας πάρα πολλά χρήματα και προσπαθώντας να προσφέρουν όσο περισσότερες ευκολίες στους χρήστες. Παρόλα αυτά σχεδόν πάντα οι εφαρμογές ξεκινούσαν δυναμικά και αποκτούσαν μεγάλο αριθμό χρηστών αλλά μετά από λίγο χάνονταν από τον διαδικτυακό χάρτη. Ο λόγος ήταν οι συνεχώς αυξανόμενες απαιτήσεις των χρηστών που τους ωθούσε μετά από κάποιο χρονικό διάστημα χρήσης κάποιας εφαρμογής να την εγκαταλείπουν αναζητώντας μία καλύτερη. Τέτοιες εφαρμογές ήταν οι tribe.net, linkedin, Clasmmates.com, Jaiku, Netlog κ.α.

Το 2003 στην αγορά εμφανίστηκε μία εφαρμογή η οποία μεσουρανού μέχρι και σήμερα με το όνομα MySpace.com και ουσιαστικά αποτελεί ένα βελτιωμένο κλώνο του FriendSter. Δημιουργήθηκε όταν πολλοί υπάλληλοι της eUniverse συμμετείχαν στο δικτυακό τόπο FriendSter και παρατήρησαν τις απίστευτες δυνατότητες που έδινε. Έτσι το 2003 και μέσα σε μόλις 10 μέρες δημοσιοποίησαν την πρώτη έκδοση του MySpace, την οποία αρχικά χρησιμοποιούσαν μόνο υπάλληλοι της εταιρείας. Όμως με τη βοήθεια των ηλεκτρονικών μηνυμάτων η δημοτικότητα του μεγάλωσε τόσο που έγινε εργαλείο για 20 εκατομμύρια ανθρώπους. Ο λόγος της δημοτικότητας του έναντι των άλλων δικτυακών τόπων ήταν ότι ναι μεν προσέφερε ότι και τα υπόλοιπα, δηλαδή δυνατότητα δημιουργίας προφίλ, αποστολής μηνυμάτων και διατήρησης λίστας φίλων αλλά είχε μια επιπλέον υπηρεσία. Αυτή ήταν ότι επέτρεπε στους χρήστες να διαμορφώσουν το προφίλ τους με όποιο τρόπο ήθελαν εμφανισιακά και επίσης να «ανεβάζουν» τραγούδια μετατρέποντας το έτσι σε εμπορικό εργαλείο για μουσικούς καλλιτέχνες. Έτσι η σελίδα είχε πλέον ως σκοπό όχι μόνο τις προσωπικές σχέσεις μεταξύ ανθρώπων αλλά και την δημιουργία μουσικών προφίλ προωθώντας την δουλειά τους.

Ενώ τα μέσα κοινωνικής δικτύωσης και γενικότερα το διαδίκτυο είχε γίνει απαραίτητο αντικείμενο στη καθημερινότητα των ανθρώπων το 2004 δημιουργήθηκε το πιο δημοφιλές δίκτυο κοινωνικής δικτύωσης μέχρι και

σήμερα, το Facebook. Η ελκυστικότητα του user interface και η ευκολία του προσέλκυσε πάρα πολλούς χρήστες. Είναι το μοναδικό μέχρι τότε που δεν το ενδιέφερε ιδιαίτερα η δημιουργία σχέσεων μεταξύ των ανθρώπων αλλά επικεντρωνόταν στο να βρει ένας χρήστης άτομα που ήδη γνώριζε έκανε παρέα ή είχε χάσει με το πέρασμα των χρόνων. Επίσης πολύ σημαντικό ήταν η λειτουργία δημιουργίας ομάδων (groups) που προσέλκυε άτομα με κοινά ενδιαφέροντα όπως η μουσική, το σχολείο, τα αθλητικά και ότι άλλο μπορεί να φανταστεί κανείς.

Το 2005 ένας ιδιαίτερος και εξίσου πετυχημένος ιστιότοπος με το Myspace έκανε την εμφάνιση του ο δημοφιλέστατος μέχρι σήμερα Youtube. Αυτό το site επιτρέπει στο χρήστη να φορτώσει στον ιστιότοπο τα προσωπικά του βίντεο καθώς και να παρακολουθήσει βίντεο άλλων χρηστών. Μάλιστα η πολύ εύκολη χρήση του και ο μεγάλος αριθμός χρηστών που προσελκύει, έκανε πολλές εταιρείες να το χρησιμοποιούν σήμερα για τη προώθηση των προϊόντων τους, καθώς και καθηγητές πανεπιστημίων να το χρειάζονται για την οπτικοποίηση της διδασκαλίας τους.

Από το 2006 το twitter έχει προσελκύσει αρκετούς ανθρώπους λόγω κυρίως της ευκολίας που έχει να χρησιμοποιείται και μέσω κινητών τηλεφώνων. Ο χρήστης αφήνει μικρού μήκους μηνύματα σχετικά με το τι κάνει, που είναι η απλά τις σκέψεις που έχει κάθε λεπτό της ημέρας. Όσο περνάει ο καιρός όλο και πιο πολλά μέλη αποκτά εξαιτίας της αλματώδους εξέλιξης των κινητών τηλεφώνων. Μάλιστα κατά το παγκόσμιο κύπελλο του 2010 στην Ιαπωνία μετρήθηκαν ότι οι χρήστες έστελναν μηνύματα μέσω της υπηρεσίας (tweets) με συχνότητα 3283 μηνύματα το δευτερόλεπτο.

Τέλος το 2011 ένα μεγάλο μερίδιο της πίτας των κοινωνικών μέσων δικτύωσης διεκδίκησε η google η οποία δημιούργησε το google+. Η ανακοίνωση του έγινε στις 28 Ιουνίου του 2011 και αρχικά ήταν ανοικτό μόνο για περιορισμένους χρήστες. Μετά από δύο εβδομάδες είχε 25 εκατομμύρια χρήστες και η ανοδική του τάση όλο και συνέχιζε. Σήμερα αποτελεί το δεύτερο μεγαλύτερο κοινωνικό δίκτυο αφού τον Ιανουάριο του 2013 ανακοινώθηκε ότι έχει συνολικά 500 εκατομμύρια χρήστες και έχει ξεπεράσει τους χρήστες του twitter.

Εκτός από αυτά που ήδη έχουν αναφερθεί μέσα στο πέρασμα των χρόνων υπήρχαν μία πληθώρα προσπαθειών για δημιουργία κοινωνικών μέσων. Κάποια από αυτά είναι hi5, Qzone, last.fm, fotolog, livejournal, orkut, skyblog, studiverzeichnis, bebo, κ.α. Μια εικόνα για τη πληθώρα των κοινωνικών μέσων που έχουν δημιουργηθεί στο φαίνεται στη παρακάτω εικόνα.



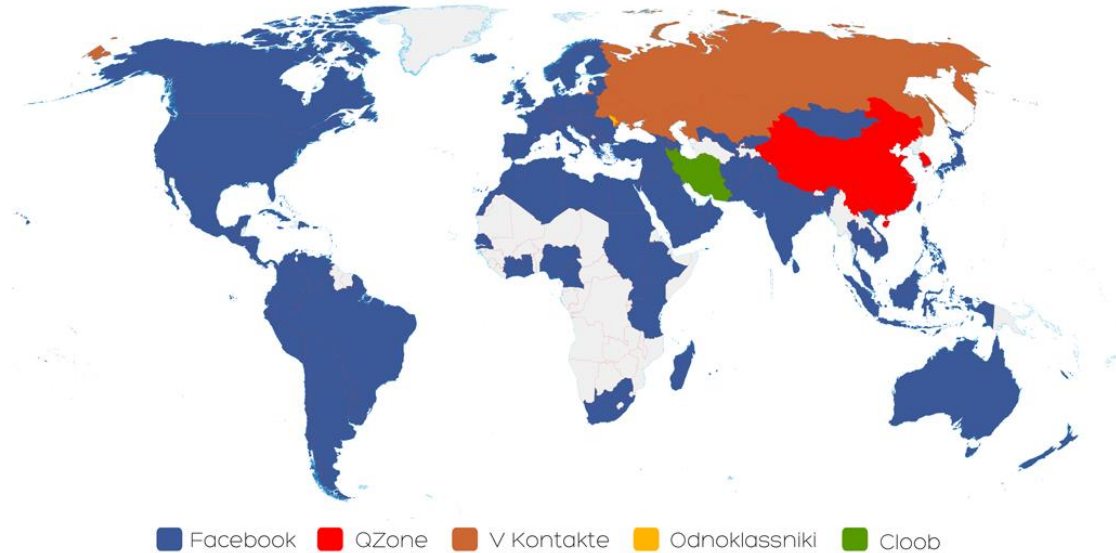
Εικόνα 2-Πλήθος social media

Παρόλα το πλήθος των κοινωνικών μέσων που έχουν δημιουργηθεί βέβαια η παγκόσμια κυριαρχία ανήκει στο facebook το οποίο έχει σήμερα πάνω από ένα δισεκατομμύρια ενεργούς εγγεγραμμένους χρήστες . Αναλυτικά μάλιστα στην Ασία, το Facebook μετρά 278 εκατομμύρια χρήστες, στην Ευρώπη 251 εκατομμύρια, στη Β. Αμερική 243, στη Ν. Αμερική 142, στην Αφρική 52 και στην Ωκεανία 15 εκατομμύρια χρήστες.

Έτσι η εικόνα που έχουμε για τα δημοφιλέστερα κοινωνικά δίκτυα ανά χώρα και η οποία παρουσιάστηκε έπειτα από στατιστικά στοιχεία τον Δεκέμβριο του 2012 είναι η παρακάτω

WORLD MAP OF SOCIAL NETWORKS

December 2012



credits: Vincenzo Cosenza vincos.it

license: CC-BY-NC

sources: Alexa

Εικόνα 3-Δημοφιλή κοινωνικά δίκτυα ανα χώρα

Συμφώνα με το χάρτη από τις 137 χώρες που παρουσιάζονται στις 127 έρχεται πρώτο το Facebookk ενώ στις υπόλοιπες δέκα οι προτιμήσεις των χρηστών περιορίζονται κυρίως λόγω λογοκρισίας σε τοπικά κοινωνικά δίκτυα. Συγκεκριμένα με 190 εκατομμύρια χρήστες το «V Kontakte» έρχεται πρώτο στη Ρωσία , στην Κίνα και τη Νότια Κορέα το «QZone» ενώ τέλος στο Ιράν κυρίως λόγω της απαγόρευσης της ιρανικής κυβέρνησης της πρόσβασης στο Facebook πρώτο έρχεται το «Cloobe»

2.2 Τύποι κοινωνικών μέσων δικτύωσης

Τα κοινωνικά μέσα δικτύωσης χαρακτηρίζονται από τις ιδιαίτερες δυνατότητες που δίνουν στο χρήστη και κατατάσσονται σε ορισμένα είδη ανάλογα με το χαρακτήρα και τη χρησιμότητα που έχουν. Αν τα χωρίζαμε σε δυο πολύ γενικές κατηγορίες αυτές θα ήταν στα μέσα αυτά που στηρίζονται στη διαμόρφωση των προφίλ των χρηστών τους, και σε αυτά που κύριο στόχο έχουν την παροχή πληροφοριών μέσω του περιεχομένου που δημιουργούν τα μέλη τους. Όμως η συνεχής εξέλιξη τους έχουν δημιουργήσει νέες κατηγορίες κοινωνικών μέσων πιο εξειδικευμένα σε συγκεκριμένες λειτουργίες. Έτσι τα μέσα κοινωνικής δικτύωσης μπορούν να χωριστούν στις παρακάτω κατηγορίες.

- *ΚΟΙΝΩΝΙΚΑ ΔΙΚΤΥΑ ΒΑΣΙΣΜΕΝΑ ΣΤΗ ΠΡΟΦΙΛ ΧΡΗΣΤΗ*

Αφορά τις υπηρεσίες αυτές που βασίζονται στο προφίλ των χρηστών και λειτουργούν έχοντας ως βάση τις σελίδες των μελών τους. Οι σελίδες αυτές περιέχουν πληροφορίες σχετικά με προσωπικά στοιχεία κάθε μέλους όπως είναι φωτογραφίες, ενδιαφέροντα που μπορεί να έχει και γενικά τις σκέψεις που οι χρήστες δημοσιεύουν όποτε το θελήσουν. Επίσης μια επιπρόσθετη δυνατότητα των χρηστών είναι πέρα από το δικό τους προφίλ να προθέτουν κείμενα, βίντεο, σχόλια στα προφίλ χρηστών που είναι φίλοι στο δίκτυο ή να χρησιμοποιούν τα εργαλεία αξιολόγησης για να επικροτήσουν ή όχι τις πληροφορίες που είναι ήδη αναρτημένες. Μάλιστα τελευταίες σημαντικές προσθήκες στα δίκτυα αυτά είναι η χρήση ζωντανού σύνδεσης με εικόνα και ήχο μεταξύ των χρηστών καθώς και τα σύντομα μηνύματα(chat). Σημαντικά παραδείγματα τέτοιων δικτύων είναι το Facebook με 750 εκατομμύρια χρήστες σήμερα, το MySpace και το Bebo.

- ΚΟΙΝΩΝΙΚΕΣ ΔΙΚΤΥΑΚΕΣ ΥΠΗΡΕΣΙΕΣ ΒΑΣΙΣΜΕΝΕΣ ΣΤΟ ΠΕΡΙΕΧΟΜΕΝΟ**

Είναι οι υπηρεσίες που βασίζονται όχι τόσο στο προφίλ του κάθε χρήστη αλλά στο περιεχόμενο που δημοσιεύεται. Υπάρχουν πάρα πολλά παραδείγματα τέτοιων σελίδων όπως είναι το Flickr, Yfrog τα οποία προσφέρουν στους χρήστες τη δυνατότητα κατασκευής προφίλ και προσθήκης φωτογραφιών. Τα μόνα σχόλια που γίνονται αφορούν τις φωτογραφίες και χρήστες που δεν είναι εγγεγραμμένα μέλη αυτών των υπηρεσιών δεν μπορούν να έχουν πρόσβαση σε αυτές. Άλλα παραδείγματα παρόμοιων υπηρεσιών είναι το πασίγνωστο YouTube με το οποίο γίνεται διαμοιρασμός βίντεο, του Shelfari που δίνει τη δυνατότητα στους χρήστες να δημιουργήσουν ράφια με βιβλία που έχουν διαβάσει ή που απλά κατέχουν και να γίνει η ανάλογη αξιολόγηση από τα υπόλοιπα μέλη. Τέλος το last.fm με το οποίο ακούει ο χρήστης μουσική μέσω του διαδικτύου, ή από το σκληρό δίσκο του υπολογιστή του και ταυτόχρονα χτίζεται ένα αναλυτικό προφίλ με τις προτιμήσεις του.
- ΚΟΙΝΩΝΙΚΕΣ ΥΠΗΡΕΣΙΕΣ ΑΣΠΡΗΣ-ΜΑΥΡΗΣ ΕΤΙΚΕΤΑΣ**

Μιλάμε για πλατφόρμες που δίνουν τη δυνατότητα στους χρήστες να δημιουργήσουν ομάδες ή να εισχωρήσουν σε κάποιες από τις ήδη υπάρχουσες. Σε αυτές έχουν κοινά ενδιαφέροντα με τα υπόλοιπα μέλη, η συμμετέχουν σε κάποιες δραστηριότητες μαζί και έχουν αυξημένες διαδικτυακές υποχρεώσεις ως μέλη αυτών των ομάδων. Τέτοιες υπηρεσίες προσφέρει το People Aggegator αλλά και εδώ και κάποιους μήνες το έχει δώσει σαν πρόσθετη λειτουργία το Facebook.
- ΕΙΚΟΝΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ ΠΟΛΛΩΝ ΧΡΗΣΤΩΝ**

Αφορά χρήστες που αλληλεπιδρούν μεταξύ τους χρησιμοποιώντας μια ψεύδη εικονική ταυτότητα μέσω avatar. Μπορούν μέσω αυτών να κοινωνικοποιούνται , να εξερευνούν τον εικονικό τους κόσμο ,να κάνουν εμπόριο ακόμα και να συμμετάσχουν σε ομαδικές

δραστηριότητες. Τέτοια παραδείγματα είναι οι ιστιότοποι Second life και World Of Warcraft.

- *ΚΙΝΗΤΕΣ ΚΟΙΝΩΝΙΚΕΣ ΥΠΗΡΕΣΙΕΣ ΔΙΚΤΥΩΣΗΣ*

Στη κατηγορία αυτή ανήκουν οι ιστοσελίδες που έχουν μεταφέρει τις υπηρεσίες τους σε κινητές τηλεφωνικές εκδόσεις και μπορούν να αλληλεπιδρούν τα μέλη μέσω κινητών τηλεφώνων. Τέτοια παραδείγματα είναι το Twitter ή το MySpace τα οποία όμως δεν είναι και τα μόνα πλέον αφού σχεδόν όλα τα κοινωνικά δίκτυα με την ανάπτυξη της τεχνολογίας των κινητών τηλεφώνων τείνουν να γίνουν και αυτά κινητές υπηρεσίες δικτύωσης

- *ΥΠΗΡΕΣΙΕΣ ΠΡΟΣΦΑΤΩΝ ΕΝΗΜΕΡΩΣΕΩΝ*

Αφορούν υπηρεσίες που επιτρέπουν στους χρήστες την δημοσίευση μηνυμάτων με περιορισμό χαρακτήρων που φτάνει τους 160 και στηρίζονται στο πρότυπο των κινητών τηλεφώνων. Μάλιστα από τους 160 μόνο τους 140 μπορούν να χρησιμοποιηθούν αφού οι υπόλοιποι είναι για να τη διεύθυνση που έχει ο χρήσης στην υπηρεσία αυτή. Ουσιαστικά όταν δημιουργήθηκαν αφορούσαν κατά κόρον χρήστες κινητών τηλεφώνων αλλά έγιναν τόσο δημοφιλή που πλέον τις χρησιμοποιούν και μέσω του ηλεκτρονικού υπολογιστή. Χρησιμεύουν ώστε να ενημερώνονται οι φίλοι του μέλους για τη διάθεση του, το τι κάνει ή ακόμα και το που βρίσκεται κάθε στιγμή. Είναι μια πολύ γρήγορη υπηρεσία αφού αρκούν δύο γραμμές για να την ενημερώσεις. Ο ποιο γνωστός και ποιο διαδεδομένος ιστιότοπος σήμερα είναι το twitter που έχει 200 εκατομμύρια χρήστες και δέχεται καθημερινά 200 εκατομμύρια περίπου μηνύματα από τους χρήστες του.

3. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

Ο όρος αρχιτεκτονική συνήθως χρησιμοποιείται για να περιγράψει συστήματα διαχείρισης βάσεων δεδομένων, λειτουργικά συστήματα και άλλους υπολογιστικούς μηχανισμούς λογισμικού και υλικού. Οι αρχιτεκτονικές περιγράφουν πως οι συσκευές και τα λογισμικά πακέτα ταιριάζουν για να φτιάξουν είναι εύκολο στην χρήση και στην διαχείριση σύνολο.

Συνήθως την έννοια αρχιτεκτονική τη χρησιμοποιούμε για να περιγράψουμε με ποιες τεχνικές και με ποια λογιστικά πακέτα λειτουργούν συστήματα βάσεων δεδομένων, λειτουργικά συστήματα και υπολογιστικοί πόροι, έτσι ώστε να έχουν καλύτερα αποτελέσματα. Ταυτόχρονα λοιπόν, με την ανάπτυξη των διαδικτυακών εφαρμογών άρχισαν να διατυπώνονται διάφορες προτάσεις σχετικά με τις κατάλληλες αρχιτεκτονικές για τη καλύτερη υποστήριξη τους. Άλλωστε αν αναλογιστεί κανείς ότι η αρχιτεκτονική ενός συστήματος το επηρεάζει στην απόδοση, στη λειτουργικότητα και στην αποτελεσματικότητα του σε ολόκληρο το κύκλο ζωής του μπορεί να καταλάβει πόσο σημαντικό είναι ένα σύστημα να είναι σωστά δομημένο. Τα διαδικτυακά χαρακτηριστικά των γλωσσών προγραμματισμού όπως της Java βοήθησαν τις νέες αρχιτεκτονικές στην ασφάλεια και τη λειτουργικότητα των εφαρμογών.

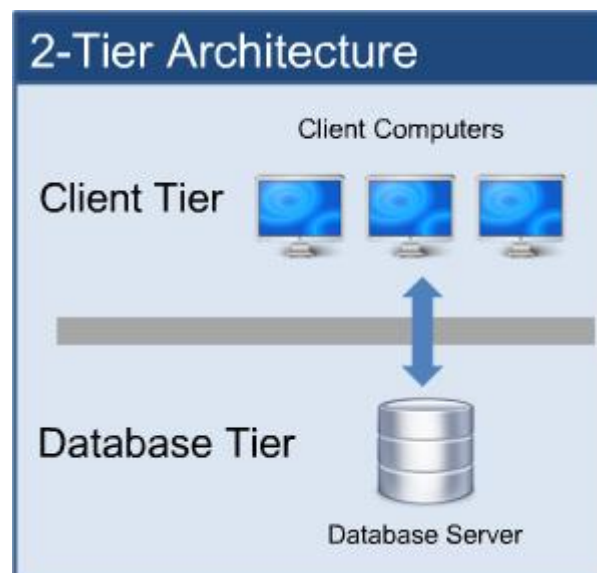
Το λογισμικό που αναπτύχθηκε στην συγκεκριμένη εργασία ,ακολούθησε την αρχιτεκτονική δύο επιπέδων(two tier architecture) ή αλλιώς client-server αρχιτεκτονική. Ο τρόπος κατασκευής της περιλαμβάνει δυο λογικά επίπεδα που συνεργάζονται μεταξύ τους μέσω διάφορων προγραμμάτων και πρωτοκόλλων. Αναλυτικά η αρχιτεκτονική δυο επιπέδων περιγράφεται στις παρακάτω σελίδες:

3.1 Αρχιτεκτονική δύο επιπέδων

Η αρχιτεκτονική διακομιστή -πελάτη αποτελεί μια κατανεμημένη εφαρμογή όπου ο φόρτος εργασίας μοιράζεται ανάμεσα σε δύο επίπεδα που αποτελούνται το ένα από τους χρήστες της εφαρμογής(client) και το άλλο από τον πάροχο υπηρεσιών(server). Η αρχιτεκτονική αυτή απεικονίζει τη σχέση

μεταξύ δυο υπολογιστικών προγραμμάτων που ο client πρόγραμμα ζητάει από τον server πρόγραμμα υπηρεσίες και που χωρίς αυτές θα ήταν αδύνατο να λειτουργήσει. Στον πελάτη βρίσκεται η εφαρμογή και εκεί «τρέχει» ενώ στον διακομιστή βρίσκεται η βάση δεδομένων. Όταν ο πελάτης ξεκινάει την εφαρμογή ταυτόχρονα γίνεται μία σύνδεση με τον διακομιστή για να μπορέσει να έχει πρόσβαση στα δεδομένα τη βάσης που δεν μπορεί να χειριστή απευθείας αλλά μόνο μέσω της σύνδεσης αυτής. Βασικές δικτυακές εφαρμογές όπως είναι η πρόσβαση στο διαδίκτυο , η πρόσβαση σε πληροφορίες βάσεων ή ακόμα και η ανταλλαγή δεδομένων μεταξύ χρηστών στηρίζονται στη συγκεκριμένη αρχιτεκτονική. Το ίδιο ισχύει και για τα κυριότερα πρωτόκολλα και αυτό κυρίως ισχύει στο ότι η αρχιτεκτονική των δύο επιπέδων θεωρείται η πρώτη σωστά δομημένη αρχιτεκτονική. Με την εξέλιξη των εφαρμογών δημιουργήθηκαν και άλλες αρχιτεκτονικές πιο πολύπλοκες αλλά πάντα είχαν ως βάση αυτήν.

Μια απεικόνιση της client-server αρχιτεκτονικής φαίνεται και στο παρακάτω σχήμα:



Εικόνα 4-Αρχιτεκτονική δυο επιπέδων

3.1.1 Πλεονεκτήματα τεχνολογίας

Τα πλεονεκτήματα που έχουν τα προγράμματα που χρησιμοποιούν αρχιτεκτονική δύο επιπέδων είναι τα εξής:

- Αρχικά στην αρχιτεκτονική client-server ο κάθε μέρος του συστήματος έχει το δικό του ξεχωριστό ρόλο χωρίς να μπερδεύονται οι αρμοδιότητες μεταξύ τους. Αυτό έχει ως συνέπεια η συντήρηση του συστήματος, οι αναβαθμίσεις, οι προσθήκες δυνατοτήτων γίνονται εύκολα αφού πραγματοποιούνται μόνο στον server και μάλιστα οι clients δεν καταλαβαίνουν τίποτα κατά την αλλαγή.
- Παρέχει μεγαλύτερη ασφάλεια από άλλες αρχιτεκτονικές αφού τα δεδομένα στη συγκεκριμένη περίπτωση είναι μαζεμένα όλα στην πλευρά τους server. Αυτός είναι που ελέγχει ποιος θα έχει πρόσβαση σε αυτά, τι ακριβώς από τα δεδομένα θα μπορεί να δει και να αλλάξει και φυσικά ποιες αιτήσεις σύνδεσης από τους πελάτες θα γίνουν δεκτές.
- Γίνεται εύκολα ενημέρωση των δεδομένων του συστήματος λόγω της κεντροποίησης της αποθήκευσης των δεδομένων μόνο στον server. Όταν λέμε ενημέρωση δεδομένων εννοούμε και τα δεδομένα που εισάγει ο server και που έπειτα μπορούν να δουν οι πελάτες, αλλά και τα δεδομένα που προσθέτουν οι υπόλοιποι χρήστες του συστήματος. Μάλιστα με κάθε αλλαγή των δεδομένων οι πελάτες ενημερώνονται αυτόματα.
- Δυνατότητα παροχής υπηρεσιών σε πάνω από ένα χρήστη και μάλιστα με καθορισμό διαφορετικών ρόλων για τον καθένα και για το τι θα μπορεί να κάνει στο σύστημα.
- Πάρα πολύ σημαντικό πλέον είναι ότι πολλά συστήματα με αρχιτεκτονική client-server έχουν αναπτυχθεί και παρέχουν στον χρήστη το πολύ απλό χειρισμό του.

3.1.2 Μειονεκτήματα Τεχνολογίας

Τα βασικά μειονεκτήματα της συγκεκριμένης τεχνολογίας είναι τα εξής

- Η επεκτασιμότητα του συστήματος εξαρτάται άμεσα από την ισχύ του εξυπηρετητή αφού όσο χαμηλότερη είναι τόσο λιγότερες είναι οι δυνατότητες που υπάρχουν.
- Επίσης υπάρχει ένα κεντρικό σημείο όπου μπορεί να προκαλέσει την αποτυχία του συστήματος και αυτό είναι όταν χρησιμοποιείται μόνο ένας εξυπηρετητής.

4. ΔΟΜΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Η Βάση δεδομένων έχει κατασκευαστεί με τον SQL SERVER 2005 και ονομάζεται Stockdata. Αποτελεί την κεντρική βάση της εφαρμογής και αποτελείται από τους εξής πέντε πίνακες.

- Customer
- Customer_Stock
- Stock
- VoiceMessage
- Message

4.1 Customer

Στον πίνακα customer αποθηκεύονται όλοι οι χρήστες που έχουν κάνει εγγραφή στο σύστημα και που έχουν τη δυνατότητα login. Τα στοιχεία που περιέχει είναι τα εξής:

ΠΕΔΙΟ	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
<u>CustomerID</u>	Int	Αποτελεί το κλειδί αναγνώρισης του χρήστη και αντιστοιχεί ένα σε κάθε πελάτη του συστήματος
CustomerFName	nvarchar(50)	Είναι το πεδίο όπου αποθηκεύεται το όνομα του χρήστη
CustomerLName	nvarchar(50)	Είναι το πεδίο όπου αποθηκεύεται το επώνυμο του χρήστη

Username	varchar(50)	Το χαρακτηριστικό αυτό είναι το όνομα που έχει ο χρήστης στο συγκεκριμένο σύστημα και χρησιμοποιεί κατά την είσοδο
Password	nvarchar(50)	Μαζί με το Username το password είναι τα χαρακτηριστικά που εισάγει στο σύστημα για να ταυτοποιηθεί ο χρήστης .
Email	nvarchar(50)	Εδώ αποθηκεύεται το email του χρήστη όπου βοηθάει στο να μην υπάρχουν διπλές εγγραφές στο σύστημα.

Πίνακας 1-Πίνακας Customer Βάσης Δεδομένων

4.2 Customer_Stock

Ο πίνακας αυτός αποτελεί τη συσχέτιση μεταξύ των διαφορετικών πελατών και των διαφορετικών μετοχών που επιλέγουν. Τα στοιχεία που περιέχει είναι τα ακόλουθα:

ΠΕΔΙΟ	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
<u>Customer Stock ID</u>	int	Είναι ο κωδικός που χαρακτηρίζει κάθε εγγραφή στο πίνακα customer_Stock
StockID	int	Αποτελεί το πεδίο αυτό που συνδέει το πίνακα customer_stock με το πίνακα

		stock
CustomerID	int	Είναι το κοινό πεδίο το πίνακα customer_stock με το πίνακα Customer. Με τη βοήθεια αυτού η κάθε εγγραφή αντιστοιχεί στον κάθε χρήστη
isActivate	int	Με τη βοήθεια αυτού του πεδίου ενεργοποιείται η απενεργοποιείται η ένωση μιας μετοχής με ένα πελάτη. Όταν η εγγραφή υφίσταται παίρνει τη τιμή 1 ενώ όταν απενεργοποιείται τη τιμή 0.

Πίνακας 2-Πίνακας Customer_Stock Βάσης Δεδομένων

4.3 Stock

Στο συγκεκριμένο πίνακα αποθηκεύονται όλες οι μετοχές που έχουν επιλέξει να βλέπουν οι χρήστες .Αν κάποια δεν υπάρχει μέσα στη βάση δημιουργείται σαν καινούργια εγγραφή. Τα στοιχεία που υπάρχουν είναι

ΠΕΔΙΟ	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
<u>StockID</u>	int	Είναι ο κωδικός που χαρακτηρίζει κάθε εγγραφή στο πίνακα Stock

StockName	nvarchar(50)	Η κάθε μετοχή διακρίνεται από το όνομα της το οποίο εισάγεται σαν αλφαριθμητικό σε αυτό το πεδίο
------------------	--------------	--

Πίνακας 3-Πίνακας Stock Βασης Δεδομένων

4.4 Voice Message

Είναι ο πίνακας στον οποίο αποθηκεύονται τα προσωπικά μηνύματα φωνής που στέλνουν οι χρήστες μεταξύ τους. Τα δεδομένα που αποθηκεύονται στο πίνακα αυτόν είναι τα παρακάτω:

ΠΕΔΙΟ	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
VoiceMessageID	int	Είναι ο κωδικός που χαρακτηρίζει κάθε εγγραφή στο πίνακα VoiceMessage
VoicePath	nvarchar(MAX)	Το κάθε φωνητικό μήνυμα αποθηκεύεται μέσα στον υπολογιστή και με τη βοήθεια της συγκεκριμένης εγγραφής βρίσκουμε το που είναι. Είναι το path του αρχείου.
Date	nvarchar(MAX)	Εδώ αποθηκεύεται η ημερομηνία που στάλθηκε το μήνυμα

Time	nvarchar(MAX)	Το πεδίο που αποθηκεύεται η ώρα που στάλθηκε το μήνυμα.
CustomerID	int	Είναι το κοινό πεδίο το πίνακα VoiceMessage με το πίνακα Customer. Με τη βοήθεια του συγκεκριμένου πεδίου το κάθε μήνυμα βρίσκει τον παραλήπτη του.
Sender	nvarchar(50)	Εδώ αποθηκεύεται το όνομα του χρήστη που έχει στείλει το φωνητικό μήνυμα.
isActivate	int	Με τη βοήθεια αυτού του πεδίου ενεργοποιείται η απενεργοποιείται το αν ένα μήνυμα είναι καινούργιο η παλιό. Όταν το μήνυμα το ακούσει ο χρήστης, τότε η τιμή εδώ γίνεται 0 διαφορετικά είναι 1

Πίνακας 4-Πίνακας VoiceMessage Βάσης Δεδομένων

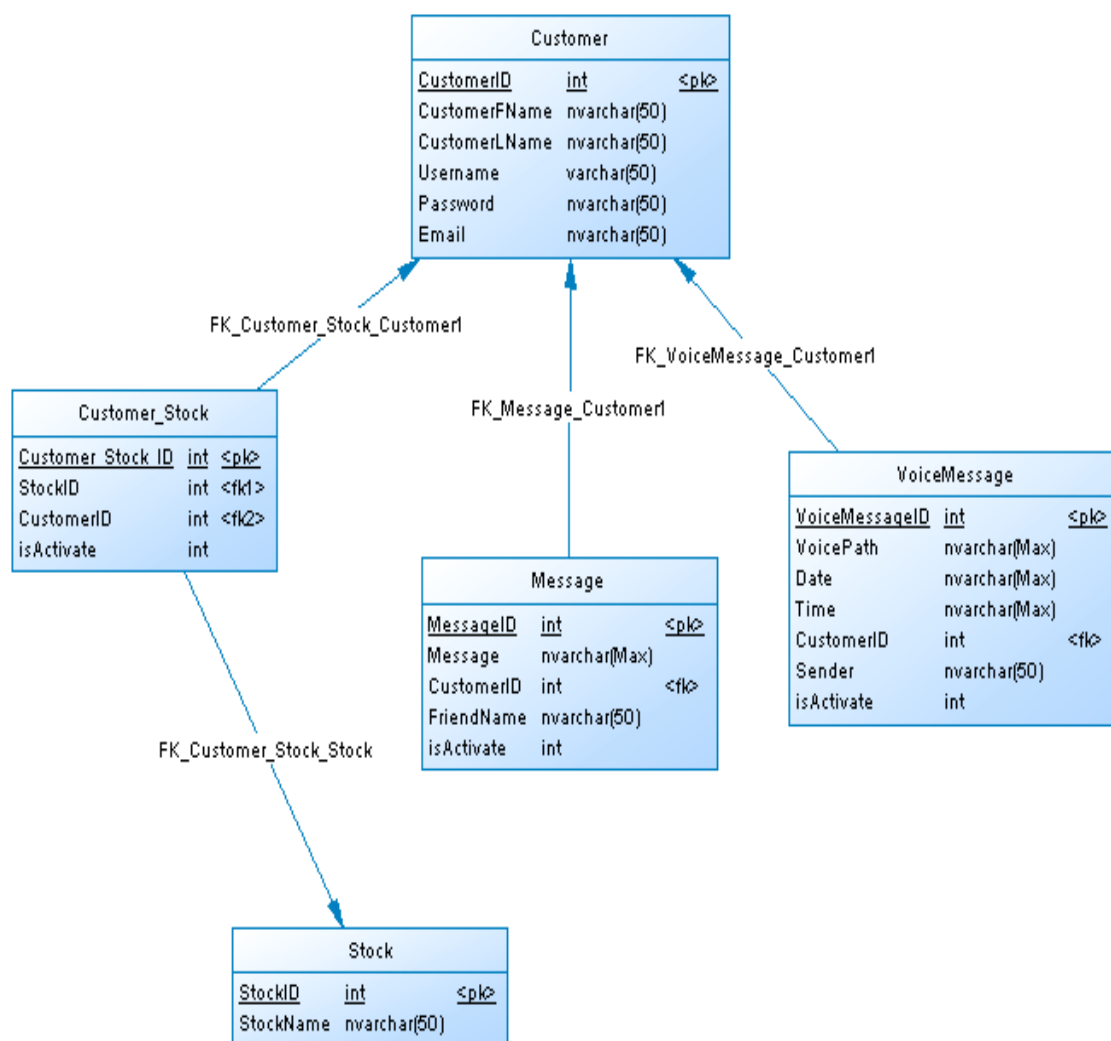
4.5 Message

Αποτελεί τον πίνακα στον οποίο αποθηκεύονται τα μηνύματα κειμένου που αφήνουν οι χρήστες σε κάθε διάγραμμα. Τα μηνύματα είναι άμεσα συνδεδεμένα με το διάγραμμα του πελάτη. Μέσα υπάρχουν τα στοιχεία.

ΠΕΔΙΟ	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
MessageID	int	Είναι ο κωδικός που χαρακτηρίζει κάθε εγγραφή στο πίνακα Message
Message	nvarchar(MAX)	Είναι το μήνυμα κειμένου που αφήνει ο χρήστης στο comment box.
CustomerID	int	Είναι το κοινό πεδίο του πίνακα Message με τον πίνακα Customer. Με τη βοήθεια αυτού η εγγραφή του πίνακα εμφανίζεται όταν εμφανιστεί και το διάγραμμα.
FriendName	nvarchar(50)	Εδώ αποθηκεύεται το όνομα του χρήστη που έχει στείλει το μήνυμα κειμένου.
afterDate	int	Είναι η ημερομηνία που έχει ορίσει ο χρήστης μετά από την οποία εμφανίζεται το μήνυμα στο comment box.
isActivate	int	Με τη βοήθεια αυτού του πεδίου ενεργοποιείται η απενεργοποιείται το αν ένα μήνυμα είναι καινούργιο η παλιό. Αυτό καθορίζεται αν ο χρήστης αλλάξει τις μετοχές που παρακολουθεί

Πίνακας 5-Πίνακας Message Βάσης Δεδομένων

Παρακάτω εμφανίζεται και το e-r διάγραμμα της βάσης δεδομένων.



Εικόνα 5 –Διάγραμμα E-R Βάσης Δεδομένων.

5 ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

5.1 Εργαλεία ανάπτυξης

Τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι τα εξής:

- Eclipse που είναι εργαλείο συγγραφής κώδικα.
- Netbeans που αποτελεί άλλο ένα εργαλείο συγγραφής κώδικα αλλά και δημιουργίας του gui για τον χρήστη
- Microsoft Sql management studio express που είναι το εργαλείο διαχείρισης της βάσης.
- Jogl αποτελεί βιβλιοθήκη για το opengl
- Φυλλομετρητές διαδικτύου που χρησιμοποιήθηκαν για την εύρεση πληροφοριών

Eclipse

Ο Eclipse IDE αποτελεί μία ευρέως διαδεδομένη πλατφόρμα ανάπτυξης εφαρμογών της κοινότητας του ανοικτού κώδικα που δημιουργήθηκε το Νοέμβριο του 2001 από την εταιρεία IBM. Είχε ως σκοπό τη δημιουργία μίας ισχυρής πλατφόρμας ανάπτυξης λογισμικού με βάση τη java που θα έδινε τη δυνατότητα στο προγραμματιστή να ελέγχει το πρόγραμμα σε ολόκληρο το κύκλο ζωής του. Με το πέρασμα τον χρόνων αναπτύχθηκε όλο και περισσότερο λόγω των λειτουργιών που προστέθηκαν (plug-in) κάνοντας τη δουλειά της ανάπτυξης εφαρμογής όλο και πιο εύκολη καθώς και έτοιμες βιβλιοθήκες κώδικα. Σήμερα υποστηρίζει τις πιο γνωστές γλώσσες προγραμματισμού όπως είναι οι Java, C, C++, Python, Ruby, Fortran, Cobol, PHP κ.α. Εξίσου σημαντικό είναι ότι ο eclipse είναι μια εντελώς δωρεάν πλατφόρμα την οποία μπορεί να χρησιμοποιηθεί για την δημιουργία προγραμμάτων προσωπικής χρήσης, εμπορικής ή ανοικτού λογισμικού.

Για τη συγκεκριμένη εφαρμογή ο χρησιμοποιήθηκε στη συγγραφή του κώδικα ο eclipse 4.2 και οι λόγοι επιλογής του ήταν ότι είναι ένα δωρεάν λογισμικό το οποίο δέχεται και έτοιμες βιβλιοθήκες γραφικών όπως η Jogl που χρειάστηκε για τα 3d γραφικά.

Netbeans

Το εργαλείο netbeans αποτελεί μια ανοικτή πλατφόρμα για την ανάπτυξη εφαρμογών που θα χρησιμοποιηθούν μέσα σε κάποιο δίκτυο ή ανεξάρτητες σε ένα μόνο υπολογιστή. Ο netbeans δουλεύει σε όλα τα λειτουργικά συστήματα και υποστηρίζει διάφορες γλώσσες προγραμματισμού όπως είναι η java,c,c++,php κ.α. Παρέχει στον προγραμματιστή βοηθήματα που κάνουν ευκολότερη και πιο αποδοτική την ανάπτυξη εφαρμογών όπως είναι η επισήμανση συντακτικού στον κώδικα, τα παράθυρα βοήθειας, τρόπους διαχείρισης βάσεων δεδομένου και τέλος ειδικούς τρόπους κατασκευής Gui.

Στην εφαρμογή χρησιμοποιήθηκε η έκδοση 7.0.1 και με τη βοήθεια της δημιουργήθηκαν όλα τα παράθυρα που φαίνονται στην οθόνη του χρήστη. Επίσης ο netbeans παρέχει και Plug-in για τη δημιουργία ενός μέρους από τα γραφικά που χρειάστηκαν.

Microsoft Sql Management Studio

Είναι ένα δωρεάν εργαλείο που περιλαμβάνεται στο Microsoft SQL Server 2005 και στις νεότερες εκδόσεις για τη ρύθμιση, διαχείριση όλων των στοιχείων της βάσης που κατασκευάζεται. Το εργαλείο περιλαμβάνει εργαλείο δημιουργίας της βάσης που συνεργάζονται με τα αντικείμενα και τα χαρακτηριστικά του server. Επίσης έχει και ένα εργαλείο περιήγησης το οποίο δίνει τη δυνατότητα να περιηγηθεί στα δεδομένα και να τα τροποποιήσει.

Jogl

Αποτελεί μία βιβλιοθήκη η οποία δημιουργήθηκε με σκοπό τη χρησιμοποίηση της OpenGL και από την γλώσσα προγραμματισμού Java. Το 2003 ήταν η χρονιά που δημιουργήθηκε και συνεργάστηκαν για αυτό το σκοπό η εταιρεία Sun και η Silicon Graphics. Έτσι δόθηκε η δυνατότητα στους προγραμματιστές να χρησιμοποιούν τα τρισδιάστατα γραφικά στα προγράμματα τους με περισσότερες δυνατότητες από αυτές που τους έδινε ως τότε τα Java3D. Υπήρχαν και παλαιότερα προσπάθειες να ενωθεί η Java με την OpenGL αλλά με ανεπιτυχή αποτελέσματα λόγω του ότι η γλώσσα αποτελεί μία πλήρως αντικειμενοστρεφής γλώσσα. Η λογική της διαδικασίας ήταν οι βιβλιοθήκες της OpenGL να μετατραπούν σε κλάσεις όπως ήθελε η Java έχοντας τη κληρονομικότητα και τα άλλα χαρακτηριστικά των αντικειμενικών γλωσσών. Έτσι σήμερα και προγραμματιστές της Java χρησιμοποιούν την OpenGL για τις εφαρμογές τους και μάλιστα τα πιο διαδεδομένα προγράμματα γραφής Java να δέχονται την JOGL σαν plugin.

Φυλλομετρητής δικτύου

Ο Φυλλομετρητής διαδικτύου(web browser) είναι το λογισμικό που επιτρέπει την επικοινωνία του χρήστη με τα δεδομένα που υπάρχουν στο διαδίκτυο όπως είναι εικόνες βίντεο, κείμενα και άλλες πληροφορίες που είναι ανηρτημένες στον παγκόσμιο ιστό. Οι φυλλομετρητές χρησιμοποιούν τη γλώσσα HTML για την προβολή των σελίδων και για αυτό υπάρχουν περιπτώσεις που διαφέρει η εμφάνιση μιας σελίδας από έναν Browser σε έναν άλλο. Οι φυλλομετρητές που χρησιμοποιήθηκαν στη συγκεκριμένη εργασία είναι οι Google Chrome, Internet Explorer και Mozilla Firefox.

5.2 Τεχνολογίες Λογισμικού

Με σκοπό της ανάπτυξη της συγκεκριμένης εφαρμογής χρησιμοποιήθηκαν κάποιες τεχνολογίες οι οποίες είναι

- Η Java ως γλώσσα προγραμματισμού
- Η Sql σαν βάση δεδομένων
- Η OpenGL που αποτελεί το εργαλείο ανάπτυξης των γραφικών της εφαρμογής

5.2.1 Java

Η Java είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού που δημιουργήθηκε από την εταιρεία *Sun Microsystems* στις αρχές του 1991. Ο κύριος σκοπός δημιουργίας της ήταν η κάλυψη αναγκών που είχαν προκύψει και που οι ήδη υπάρχουσες γλώσσες προγραμματισμού δεν μπορούσαν να καλύψουν. Πλέον η Java αποτελεί μια από τις πιο δημοφιλείς γλώσσες που χρησιμοποιείται για τη δημιουργία εφαρμογών αυτόνομων αλλά και εφαρμογές που εκτελούνται μέσω διαδικτύου(apple).

Υπάρχουν πολλά χαρακτηριστικά που την κάνουν να ξεχωρίζει με τα κυριότερα να είναι τα εξής:

- Είναι πολύ απλή στη χρήση της και αυτό οφείλεται στο ότι οι δημιουργοί της προσπάθησαν να αφήσουν έξω χαρακτηριστικά άλλων γλωσσών προγραμματισμού που δυσκόλευαν το χρήστη.
- Είναι αντικειμενοστρεφής δηλαδή κατά τον προγραμματισμό οργανώνει τον κώδικα σε αντικείμενα. Κάθε αντικείμενο έχει τις δικές του λειτουργίες και τεχνικές και έχει την ιδιότητα να κάνει απόκρυψη των συνδυασμών του. Κάθε αντικείμενο αλληλεπιδρά με τα υπόλοιπα αντικείμενα της εφαρμογής.
- Είναι μεταγλωττιζόμενη δηλαδή παράγει ένα ειδικό κώδικά byte ο οποίος μπορεί να εκτελεστεί από οποιοδήποτε λειτουργικό σύστημα

που απλά διαθέτει ένα διερμηνέα java.Επίσης επειδή ο μεταγλωττιστής της java δεν παράγει γηγενή κώδικα προσανατολισμένο σε ένα τύπο υπολογιστή οι εφαρμογές έχουν ανεξαρτησία και στην πλατφόρμα εκτέλεσης.

- Από την αρχή σχεδιάστηκε ώστε να προσφέρει *ασφάλεια* στην εκτέλεση του κώδικα σε δίκτυο επιτρέποντας την κατασκευή προγραμμάτων ελεύθερων από ιούς και με αδύνατη τροποποίηση τους. Εξίσου σημαντικό είναι και ο μηχανισμός που χρησιμοποιείται από τη java για τον έλεγχο αναμενόμενων και μη αναμενόμενων σφαλμάτων.
- Τα προγράμματα Java υποστηρίζουν *πολυνημάτωση* δηλαδή δίνει τη δυνατότητα σε ένα πρόγραμμα να έχει πολλές ξεχωριστές διαδικασίες ,οι οποίες να εκτελούνται ταυτόχρονα, συνεχώς και ανεξάρτητα η μία από την άλλη.
- Είναι μια *δυναμική* γλώσσα προγραμματισμού αφού έχει δημιουργηθεί για να μπορεί να προσαρμόζεται σε ένα συνεχώς εξελισσόμενο περιβάλλον. Παρέχει βιβλιοθήκες οι οποίες αναπτύσσονται και προσθέτονται στις ήδη υπάρχουσες δίνοντας τη δυνατότητα για δημιουργία νέων εφαρμογών.
- Τέλος υποστηρίζει τις *εφαρμογές πολυμέσων* με την προσθήκη βιβλιοθηκών για γραφικά ή για ήχο ή ακόμα και για βίντεο.
- Τέλος αποτελεί μια γλώσσα προγραμματισμού *υψηλής απόδοσης* αφού η παραγωγή εντολών μηχανής είναι απλή και γρήγορη, έχοντας ως συνέπεια η τελική μορφή του κώδικα να είναι μικρή σε μέγεθος και ταχύτατη στην εκτέλεση του.

Για την ανάπτυξη της συγκεκριμένης εφαρμογής χρησιμοποιήθηκε η Java 2 sdk standard edition version 6 η οποία είναι συμβατή με όλα τα υπόλοιπα προγράμματα που χρησιμοποιούνται και αλληλεπιδρούν μεταξύ τους.

5.2.2 Sql

Η SQL (Structured Query Language), αποτελεί σήμερα την πιο διαδεδομένη και δημοφιλής γλώσσα που χρησιμοποιείται για την ανάπτυξη και τη διαχείριση σχεσιακών βάσεων δεδομένων. Η αρχική έκδοση της γλώσσας ονομάστηκε SEQUEL και σχεδιάστηκε από την ερευνητική ομάδα της IBM. Η SQL αποτελείται από δυο τμήματα τη γλώσσα ορισμού δεδομένων (DDL, Data Definition Language) και τη γλώσσα χειρισμού δεδομένων (DML, Data Manipulation Language). Η γλώσσα ορισμού δεδομένων περιέχει τις εντολές αυτές που επιτρέπουν στο χρήστη να ορίζει και να τροποποιήσει το σχεσιακό του σχήμα αλλά και να δημιουργήσει, να αλλάξει ή να διαγράψει σχέσεις. Τέλος συμπεριλαμβάνονται και οι κατάλληλες εντολές για τη δημιουργία και την επεξεργασία όψεων. Από την άλλη η γλώσσα χειρισμού δεδομένων περιέχει όλες τις εντολές αυτές που χρειάζεται ο χρήστης για να επεξεργαστεί τα στοιχεία της βάσης του. Σε αυτές περιέχονται εντολές για την εμφάνιση καταχώρηση, τροποποίηση και διαγραφής δεδομένων και σχέσεων μέσα από τη βάση που δημιουργήθηκε μέσω της γλώσσας ορισμού δεδομένων.

Τα κυριότερα πλεονεκτήματα της είναι τα εξής:

- Υποστηρίζεται από οποιοδήποτε σχεσιακό σύστημα με συνέπεια η διατύπωση των ερωτημάτων προς τη βάση να είναι ανεξάρτητη από το σύστημα διαχείρισης βάσεων δεδομένων.
- Πλέον πολλές γλώσσες προγραμματισμού έχουν αναπτυχθεί ώστε να μπορούν να υποστηρίξουν τη διατύπωση ερωτημάτων σε SQL.
- Χρησιμοποιείται για όλες τις λειτουργίες του συστήματος διαχείρισης βάσεων δεδομένων.
- Είναι πολύ απλή στη σύνταξη της διευκολύνοντας το χρήστη από τη χρήση της και αποδεσμεύοντας τον από τις τεχνικές λεπτομέρειες υλοποίησης.

Στην εφαρμογή έχει χρησιμοποιηθεί ο sql server 2005 λόγω του απλού περιβάλλοντος εργασίας και λόγω της δυνατότητας να διαχειριστεί μεγάλο όγκο δεδομένων. Επίσης όλα τα δεδομένα είναι καταχωρημένα σαν UTF-8

Unicode μορφή για να μπορούν πλήρως να υποστηρίζονται οι γλώσσες και οι χαρακτήρες τους. Για τη πρόσβαση από την εφαρμογή στη βάση χρησιμοποιήθηκε η βιβλιοθήκη mySqlJConnector και για κάθε περίοδο λειτουργίας υπήρχε μόνο μια σύνδεση προς τη βάση για αποφυγή υπερφόρτωσης. Τέλος οι ερωτήσεις προς τη βάση έχουν γραφτεί με τέτοιο τρόπο έτσι ώστε να μπορούν να υποστηριχτούν και από άλλες βάσεις δεδομένων αν χρειαστεί να γίνει μεταφορά της βάσης σε κάποια από αυτές.

5.2.3 Opengl

Η Opengl αναπτύχθηκε το 1992 από την εταιρεία Silicon Graphics Inc και αποτελεί ένα ανοικτό πρότυπο προγραμματιστικού περιβάλλοντος εφαρμογής(API) που περιέχει συναρτήσεις δημιουργίας 2D και 3D γραφικών. Σήμερα χρησιμοποιείται ευρέως λόγω της κατασκευής εξαιρετικής οπτικής ποιότητας σκηνών αλλά και λόγω της πολύ γρήγορης απόδοσης τους.

Έχει χρησιμοποιηθεί σε όλα σχεδόν τα λειτουργικά συστήματα όπως είναι τα Mac, Pc ,Unix-based και έχει υποστηριχτεί από αρχιτεκτονικές υλικού, τόσο από εκείνες που υποστηρίζουν ελάχιστα το υλικό όσο και από αυτές που το υποστηρίζουν απόλυτα. Ενδεικτικά κατασκευαστές υλικού που υποστηρίζουν απόλυτα την Opengl σήμερα είναι η NVidia και η Ati. Επίσης δεν υπάρχει περιορισμός ως προς την γλώσσα προγραμματισμού στην οποία θα υλοποιηθεί το πρότυπο αφού υπάρχουν βιβλιοθήκες για τις περισσότερες γλώσσες προγραμματισμού όπως είναι η java,c++,Fortran.

Όλες οι εκδόσεις που είχαν δημιουργηθεί μέχρι την 1.5 έδιναν στο χρήστη τη δυνατότητα χειρισμού πολλών παραμέτρων, αλλά οι βασικές σχεδιαστικές λειτουργίες ήταν προκαθορισμένες από το πρότυπο. Μετά την εμφάνιση της έκδοσης 2,0 όμως δόθηκε η δυνατότητα στους προγραμματιστές να χρησιμοποιήσουν δικούς τους αλγόριθμους rendering,χρησιμοποιώντας μια υψηλού επιπέδου γλώσσα σκίασης. Με τον καιρό η Opengl αναπτύσσεται όλο και πιο γρήγορα αφού επιπλέον προγραμματιστές συνεισφέρουν τις γνώσεις τους ώστε να αναπτύξουν υψηλής απόδοσης εφαρμογές.

Σε κάθε υλοποίηση με την OpenGL συναντάς τρεις κατηγορίες βιβλιοθηκών

- Βασική Βιβλιοθήκη(OpenGL core library)

Στη βασική βιβλιοθήκη βρίσκονται όλες οι κύριες εντολές σχεδίασης οι οποίες διακρίνονται από το πρόθεμα gl. Πολλές συναρτήσεις έχουν προκαθορισμένα ορίσματα τα οποία έχουν οριστεί από την αρχή στη βιβλιοθήκη και αντιστοιχούν σε καταστάσεις λειτουργίας ή σε παραμέτρους. Αυτές οι σταθερές ξεκινούν με το πρόθεμα gl.

- OpenGL Utility Library(glu)

Στη βιβλιοθήκη αυτή βρίσκονται πιο εξειδικευμένες εντολές που καθορίζουν για παράδειγμα τα μητρώα προβολής και το σχηματισμό πιο σύνθετων καμπυλών και επιφανειών. Οι εντολές αυτής της βιβλιοθήκης ξεκινούν με το πρόθεμα glu και περιέχονται σε κάθε υλοποίηση της OpenGL.

- OpenGL Utility Toolkit(GLUT)

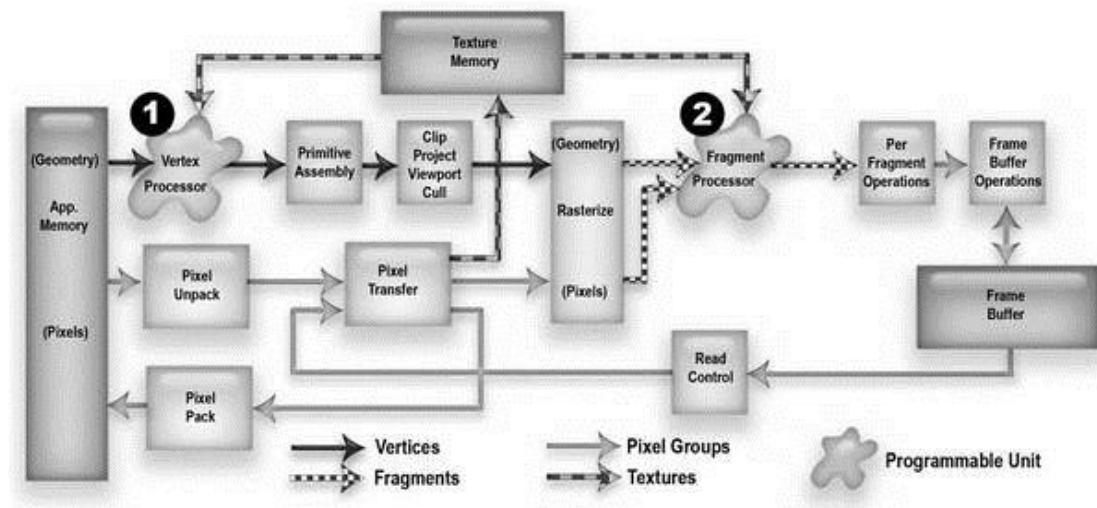
Περιέχει τις ρουτίνες εξόδου και εισόδου όπως είναι ο σχηματισμό παραθύρων κατά την εκτέλεση των εφαρμογών, η δημιουργία μενού και η διαχείριση γεγονότων. Τη βιβλιοθήκη αυτή τη χρειάζεται ένα πρόγραμμα γιατί το πρότυπο της OpenGL είναι ανεξάρτητο της πλατφόρμας αλλά οι εντολές αλληλεπίδρασης δεν είναι. Όλες οι εντολές της ξεκινούν με το πρόθεμα glu.

Τρόπος λειτουργίας

Τα προγράμματα που κάνουν χρήση της OpenGL λέμε ότι χρησιμοποιούν ένα προγραμματιστικό περιβάλλον εφαρμογής το οποίο είναι διαδικαστικό και όχι περιγραφικό. Αυτό σημαίνει ότι για να σχηματιστεί η επιθυμητή γεωμετρία στην οθόνη, δεν περιγράφεται απευθείας η εικόνα που επιθυμούμαι αλλά τα βήματα που χρειάζονται για να δημιουργηθεί. Αρχικά λοιπόν, κατασκευάζονται

τα πρωταρχικά σχήματα όπως είναι οι γραμμές, τα σημεία και τα πολύγωνα και έπειτα αυτά ενώνονται για να παρουσιαστεί η τελική εικόνα.

Η OpenGL είναι σαν ένα Pipeline επεξεργασίας, από την οποία περνάνε όλα τα δεδομένα, υποβάλλονται στις απαραίτητες επεξεργασίες και έπειτα προβάλλονται στην οθόνη. Η ακριβής διαδικασία λειτουργίας του φαίνεται στο παρακάτω σχήμα.



Εικόνα 6-OpenGL pipeline

Τα κυριότερα στάδια από την εικόνα είναι τα εξής:

- **ΟΡΙΣΜΟΣ ΓΕΩΜΕΤΡΙΑΣ**

Τα γεωμετρικά σχήματα που υποστηρίζει η OpenGL είναι τα σημεία, οι γραμμές και τα πολύγωνα. Οι τρόποι αποστολής τους στην OpenGL είναι τρεις. Ο πρώτος είναι να στέλνονται σταδιακά μια κορυφή στο πρότυπο ανάλογα με το τί χρειάζεται να σχεδιαστεί. Έπειτα στέλνονται τα υπόλοιπα στοιχεία της κατασκευής που είναι το χρώμα της κάθε γεωμετρίας, οι συντεταγμένες υψής ή τα διανύσματα του φωτισμού. Όλα στέλνονται ένα προς ένα προς το πρότυπο και το αρνητικό είναι ότι σε μεγάλες κατασκευές με πολλές κορυφές η διαδικασία είναι χρονοβόρα. Ο δεύτερος τρόπος είναι με τη βοήθεια των πινάκων όπου ο προγραμματιστής περνάει τα δεδομένα στους πίνακες με τέτοιο τρόπο ώστε οι κορυφές να περνιούνται όλες μαζί με την κλήση μίας

συνάρτησης. Το πλεονέκτημα του είναι ότι χρειάζονται πολύ λίγες συναρτήσεις για την κατασκευή της τελικής εικόνας. Η τελευταία μέθοδος παρέχεται από την OpenGL και δίνει τη δυνατότητα στο χρήστη να χρησιμοποιήσει μια δομή δεδομένων για να ενώσει τα γεωμετρικά σχήματα που λέγεται λίστες.

- *ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ ΚΟΡΥΦΩΝ*

Είναι το στάδιο όπου γίνονται οι μετατροπές στα γεωμετρικά σχήματα που στάλθηκαν στο πρότυπο κατά το προηγούμενο στάδιο. Σε αυτά συμπεριλαμβάνονται οι περιστροφές των οι μετατοπίσεις των κορυφών, υπολογίζονται οι νόρμες των αντικειμένων και υπολογίζεται ο φωτισμός.

- *ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΙΚΩΝ ΣΧΕΔΙΩΝ*

Γίνεται η ομαδοποίηση των κορυφών που έχουν σχηματιστεί μετά και τους μετασχηματισμούς. Είναι το στάδιο στο οποίο ουσιαστικά ενώνονται όλα τα προηγούμενα στοιχεία για να δημιουργηθούν τα βασικά σχήματα που είναι τα τετράγωνα, τα τρίγωνα οι γραμμές.

- *ΕΠΕΞΕΡΓΑΣΙΑ ΒΑΣΙΚΩΝ ΣΧΕΔΙΩΝ*

Είναι ο στάδιο που γίνεται η επεξεργασία των σχημάτων. Ελέγχονται οι συντεταγμένες που έχει ορίσει ο προγραμματιστής για τις κορυφές των σχημάτων και ποιες από αυτές είναι εντός σχεδίου. Όποιες είναι εκτός σχεδίου απορρίπτονται και δεν σχεδιάζονται ενώ όποιες είναι εντός διατηρούνται κανονικά. Για να αποφασιστεί το ποιες κορυφές θα σχεδιαστούν ορίζεται ένα επίπεδο στο χώρο και δοκιμάζονται όλες μία προς μία και για αυτές που θα γίνει απόρριψη θα πρέπει να γίνει εκ νέου ορισμός. Η διαδικασία αυτή γίνεται με τη βοήθεια αλγορίθμων όπως είναι ο Sutherland-Hodgman.

- *ΔΙΑΔΙΚΑΣΙΑ ΑΠΕΙΚΟΝΙΣΗΣ(RASTERIZATION)*

Η οθόνη του υπολογιστή αποτελείται από μικρά σημεία που ονομάζονται ιχνοστοιχεία(Pixel) και που χρωματίζονται ανάλογα με τα χρώματα που έχουν οριστεί. Μάλιστα κάθε ένα από αυτά τα Pixel ονομάζονται fragments. Το στάδιο λοιπόν που γίνεται αυτή η αντιστοίχιση σημείων με ιχνοστοιχεία προκειμένου να εμφανιστούν τα επιθυμητά σχήματα χρωματισμένα κατάλληλα ονομάζεται ραστεροποίηση.

- *ΕΠΕΞΕΡΓΑΣΙΑ FRAGMENT*

Είναι το στάδιο όπου γίνεται η επεξεργασία των fragment μετά τον καθορισμό τους. Σε αυτά περιλαμβάνονται ο καθορισμός του τελικού χρώματος που μπορεί να ρυθμιστεί απευθείας από τον προγραμματιστή, η εφαρμογή της υφής των σχημάτων καθώς επίσης και η προσπέλαση των τελικών συντεταγμένων από ένα χώρο μνήμης που διαθέτει η OpenGL.

- *ΕΝΕΡΓΕΙΕΣ ΑΝΑ FRAGMENT*

Είναι οι τελευταίες ενέργειες που γίνονται στα Fragment πριν εμφανιστούν στην οθόνη. Αρχικά γίνεται ο έλεγχος του βάθους και αν υπάρχει πρόβλημα γίνεται η αντικατάσταση του από άλλο μικρότερου βάθους. Επίσης γίνεται και η ανάμιξη δηλαδή όταν πρέπει να δημιουργηθεί ένα καινούργιο Pixel ελέγχεται αν υπάρχει κάποιο άλλο στη θέση του. Αν υπάρχει εξετάζεται το βάθος του και μετά από σύγκριση με αυτό που πρέπει να σχεδιαστεί επιλέγεται αυτό που έχει μικρότερο βάθος. Αυτή η διαδικασία ονομάζεται Blending.

5.3 Υλοποίηση-Κώδικας

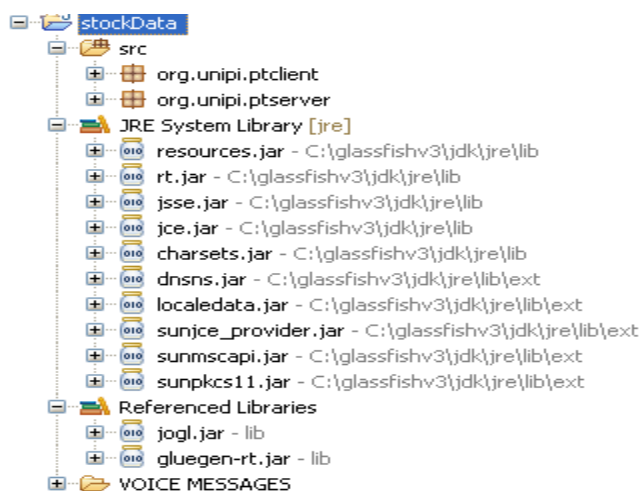
Τα βασικά στάδια για την υλοποίηση του συστήματος ήταν τα εξής:

- Σχεδιασμός της βάσης δεδομένων του συστήματος.
- Εγκατάσταση της Microsoft Sql Server 2005 και η δημιουργία της βάσης δεδομένων σύμφωνα με το σχεδιασμό όπως φαίνεται και στο σχήμα (1. E-R DIAGRAM)
- Εγκατάσταση του NetBeans IDE για την δημιουργία των γραφικών που βλέπει ο χρήστης.
- Εγκατάσταση του Eclipse IDE και συγγραφή του πηγαίου κώδικα της εφαρμογής.

Από τις παραπάνω ενέργειες θα περιγραφούν οι διαδικασίες και τα κυριότερα χαρακτηριστικά του πηγαίου κώδικα

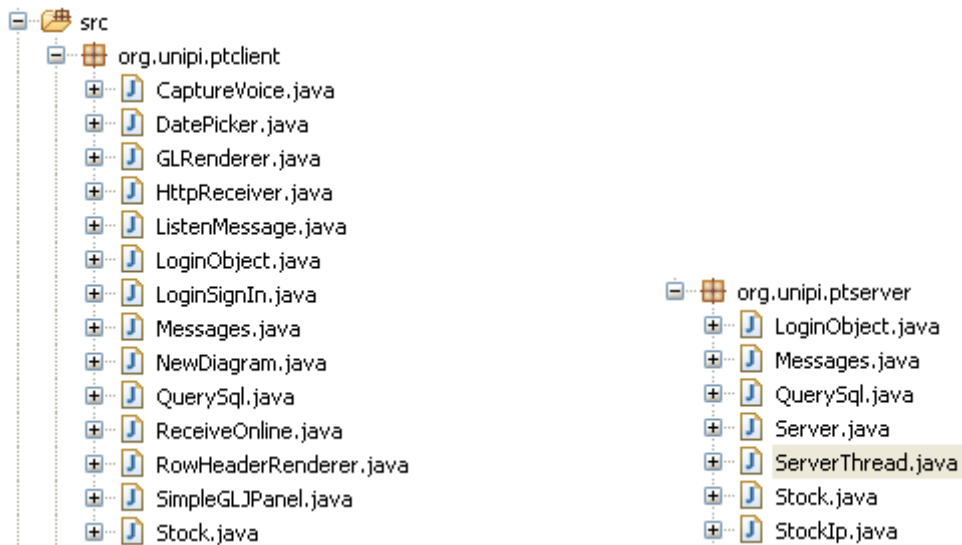
5.3.1 Πακέτα κλάσεις πηγαίου κωδικα

Όπως αναφέρθηκε η συγγραφή του πηγαίου κώδικα έγινε με τη βοήθεια του eclipse IDE. Ο χωρισμός των κλάσεων σε πακέτα έχει γίνει όπως φαίνεται στο παρακάτω σχήμα.



Εικόνα 7 –Πακέτα κώδικα

Στο πακέτο `org.unipi.ptclient` βρίσκονται όλες οι κλάσεις που χρησιμοποιούνται στην πλευρά του χρήστη ενώ αντίστοιχα στο `org.unipi.ptserver` βρίσκονται αυτές που χρειάζονται για να λειτουργήσει ο server. Υπάρχουν αντικείμενα που είναι κοινά και στις δύο πλευρές και αυτό συμβαίνει για να μπορούν να τα αναγνωρίζουν και οι δυο χωρίς πρόβλημα. Μέσα στα πακέτα λοιπόν υπάρχουν τα παρακάτω:



Εικόνα 8 – Αντικείμενα (κλάσεις) client-server

CaptureVoice

Αποτελεί την κλάση που βοηθάει τον χρήστη στην ηχογράφηση του φωνητικού μηνύματος που θέλει να στείλει σε κάποιον άλλο χρήστη. Παίρνει σαν παράμετρο το `ObjectOutputStream` που επικοινωνεί με τον server καθώς και το όνομα χρήστη το δικό του και του παραλήπτη του μηνύματος. Δημιουργεί το γραφικό μέσω του οποίου γίνεται η ηχογράφηση και η αποστολή του μηνύματος.

```

import java.io.ByteArrayInputStream;

public class CaptureVoice extends javax.swing.JFrame {

    private static final long serialVersionUID = 1L;

    public CaptureVoice(ObjectOutputStream Soutput, String usernameOtherUser, String myUserName) {
        initComponents();
        capture.setEnabled(true);
        stop.setEnabled(false);
        play.setEnabled(false);
        this.setVisible(true);
        this.Soutput = Soutput;
        this.usernameOtherUser = usernameOtherUser;
        this.myUserName = myUserName;
    }
}

```

DatePicker

Το datePicker καλείται κάθε φορά που ένας χρήστης θέλει να βάλει χρονικό περιορισμό στην εμφάνιση ενός μηνύματος στο comment box. Με τη βοήθεια αυτής της κλάσης δημιουργείται το γραφικό όπου διαλέγει τη μέρα μετά από την οποία θα εμφανιστεί το μήνυμα.

```

class DatePicker {
    int month = java.util.Calendar.getInstance().get(java.util.Calendar.MONTH);
    int year = java.util.Calendar.getInstance().get(java.util.Calendar.YEAR);;
    JLabel l = new JLabel("", JLabel.CENTER);
    String day = "";
    JDialog d;
    JButton[] button = new JButton[49];

    public DatePicker(){

        final JFrame f = new JFrame();
        d = new JDialog();
        d.setModal(true);
        String[] header = { "Sun", "Mon", "Tue", "Wed", "Thur", "Fri", "Sat" };
        JPanel p1 = new JPanel(new GridLayout(7, 7));
        p1.setPreferredSize(new Dimension(430, 120));
    }
}

```

GIRenderer

Είναι η κλάση αυτή όπου χρησιμοποιείται για τη δημιουργία του γραφικού στην οθόνη του χρήστη. Παίρνει σαν παραμέτρους ένα vector όπου περιέχονται

αντικείμενα stock όπου είναι οι μετοχές με τις τιμές τους. Επίσης παίρνει και το μέγεθος του αρχικού vector .

```
import com.sun.opengl.util.GLUT;
import java.util.Vector;
import javax.media.opengl.GL;
import javax.media.opengl.GLAutoDrawable;
import javax.media.opengl.GLEventListener;
import javax.media.opengl.glu.GLU;
import javax.media.opengl.glu.GLUQuadric;

public class GLRenderer implements GLEventListener {

    GLRenderer(Vector<Stock> allMetoxes, int sizeVector) {
        Vector<Stock> allMetoxes2 = allMetoxes;
        this.allMetoxes = allMetoxes2;
        sizeVector1 = sizeVector;
    }
}
```

HttpReceiver

Η HttpReceiver είναι η μέθοδος που χρησιμοποιείται για να παίρνει μέσω διαδικτύου της τιμές των μετοχών σε πραγματικό χρόνο. Παίρνει σαν παράμετρο το vector των μετοχών , το όνομα της μετοχής , το Textfield που θα εμφανιστεί η τιμή της μετοχής και το bufferereader.

```
import java.io.*;

public class HttpReceiver extends Thread {

    HttpReceiver(JTextField rt, BufferedReader is, Vector<Stock> allMetoxes, String name) {
        responseArea = rt;
        myReader = is;
        this.allMetoxes=allMetoxes;
        this.name=name;
    }
}
```

LoginObject

Το αντικείμενο loginObject είναι βοηθητικό για το πρόγραμμα και χρησιμοποιείται όταν γίνεται η είσοδος του χρήστη στο σύστημα. Μεταφέρει όλα εκείνα τα στοιχεία όπως είναι τα μηνύματα οι μετοχές που μπορεί αν χρειαστεί κατά τη διάρκεια της παραμονής του στην εφαρμογή.

```
public LoginObject(Vector<String> stock, Vector<Messages> message,  
    boolean pass,String customerID,String errorMessage) {  
    this.customerID=customerID;  
    this.stock = stock;  
    this.message = message;  
    this.pass = pass;  
    this.errorMessage=errorMessage;  
}
```

LoginSignIn

Είναι η αρχική εικόνα που βλέπει ο χρήστης κατά το άνοιγμα της εφαρμογής. Είναι η κλάση που ελέγχει την είσοδο του χρήστη στο σύστημα ,την εγγραφή του και κάνει και όλες τις αρχικές απαραίτητες ενέργειες σύνδεσης με τον server.

```
public class LoginSignIn extends javax.swing.JFrame {  
  
    private static final long serialVersionUID = 1L;  
    Socket socketIn;//insert data  
    Socket socketInMessage;//insert message voice  
    Socket socketOut;//out data  
    ObjectInputStream Sinput; // to read the socket  
    ObjectOutputStream Soutput;  
    ObjectInputStream SinputMessage;// to read the socket for voice message  
    LoginObject code;  
  
    /** Creates new form loginformNew */  
    public LoginSignIn() {  
        initComponents();  
    }  
}
```

Messages

Η κλάση Messages χρησιμοποιείται για τα γραπτά μηνύματα που αφήνουν οι χρήστες μεταξύ τους. Παίρνει σαν παράμετρο εισόδου το όνομα και το μήνυμα του αποστολέα.

```
public class Messages implements Serializable{
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    String message;
    String username;

    public Messages( String message,String username) {
        super();
        this.message = message;
        this.username = username;
    }
}
```

ListenMessage

Η ListenMessage είναι η κλάση που χρησιμοποιείται για τα φωνητικά μηνύματα που στέλνονται μεταξύ των χρηστών. Είναι η οθόνη που εμφανίζεται όταν ο χρήστης επιλέξει να ακούσει τα μηνύματα αυτά. Παίρνει σαν είσοδο ένα string που δείχνει αν τα μηνύματα είναι καινούργια η παλιά, ένα vector που μέσα περιέχονται τα αντίστοιχα μηνύματα της κατηγορίας και τέλος το OutputStream μέσω του οποίου γίνεται η ανανέωση των παλαιών και νέων μηνυμάτων στη βάση.

```

public ListenMessage(Vector<Messages> voiceMessage, String message,
    ObjectOutputStream Soutput) {
    this.voiceMessage = voiceMessage;
    this.Soutput=Soutput;
    initComponents();

    for (int i = 0; i < voiceMessage.size(); i++) {

        UserNames.add(voiceMessage.elementAt(i).username);

    }
    this.setVisible(true);
}

```

NewDiagram

Αποτελεί τη κλάση όπου καλείται όταν ο χρήστης επιλέξει να αλλάξει το ποιος μετοχές θέλει να παρακολουθεί. Παίρνει σαν παραμέτρους έναν Vector όπου εκεί αποθηκεύονται οι μετοχές που ενδιαφέρουν το χρήστη, το JFrame όπου δημιουργείται το γραφικό, μία βοηθητική μεταβλητή όπου ορίζεται πότε ο χρήστης έχει τελειώσει με τη δημιουργία του καινούργιου γραφήματος και το commentArea του παλιού γραφήματος.

```

public class NewDiagram extends javax.swing.JFrame implements Runnable{

    private static final long serialVersionUID = 1L;

    /** Creates new form NewDiagram */
    public NewDiagram(JTextArea commentArea,Vector<Stock> allMetoxes,JFrame myframe,ObjectOutputStream Soutput) {
        this.allMetoxes=allMetoxes;
        this.Soutput=Soutput;
        this.myframe=myframe;
        this.commentArea=commentArea;
    }
}

```

QuerySql

Σε κάθε επικοινωνία μεταξύ του server και του πελάτη της εφαρμογής χρησιμοποιείται αυτή η κλάση. Έχουν δημιουργηθεί 4 κατασκευαστές που χρησιμοποιούνται ανάλογα με τη περίπτωση. Για παράδειγμα όταν θέλει να γίνει εγγραφή χρήστη χρησιμοποιείται ο τελευταίος κατασκευαστής που

χρειάζεται σαν είσοδο το Username, first name,password,last name και το email του χρήστη. Αντίστοιχα με την εικόνα παρακάτω ο πρώτος κατασκευαστής είναι για την δημιουργία νέου διαγράμματος, ο δεύτερος για τη μεταφορά ηχητικού μηνύματος, ο τρίτος για τη μεταφορά των μηνυμάτων και ο τελευταίος για την εγγραφή νέου χρήστη.

```
public QuerySql(String query, int numquery, Vector<String> metoxes) {
    super();
    this.query = query;
    this.numquery = numquery;
    this.metoxes = metoxes;
}

public QuerySql(int numquery, String otherUserName, byte audio[],
    String username) {
    this.username = username;
    this.numquery = numquery;
    this.audio = audio;
    this.otherUserName = otherUserName;
}

public QuerySql(String query, int numquery, Vector<String> metoxes,
    String otherUserName, String comment) {
    super();
    this.query = query;
    this.numquery = numquery;
    this.comment = comment;
    this.otherUserName = otherUserName;
    System.out.println("deytero epipedo elenxou" + otherUserName);
}

public QuerySql(String firstName, String password, String lastName,
    String username, String email,int numquery) {

    this.firstName=firstName;
    this.lastName=lastName;
    this.email=email;
    this.password=password;
    this.username=username;
    this.numquery=numquery;
}
```

ReceiveOnline

Αποτελεί την κλάση αυτή που λαμβάνει σε πραγματικό χρόνο τα γραπτά μηνύματα που στέλνει κάποιος χρήστης καθώς και τα μηνύματα φωνής. Αυτή η κλάση καλείται από την αρχή της λειτουργίας της εφαρμογής και παίρνει σαν

παράμετρο το `ObjectInputStream` που ακούει τον server καθώς και το `TextArea` που το αλλάζει κάθε φορά που έρχεται ένα καινούργιο μήνυμα.

```
import java.io.*;

public class ReceiveOnline implements Runnable{

    ReceiveOnline(ObjectInputStream SinputMessage, JTextArea commentArea){
        this.commentArea=commentArea;
        this.SinputMessage=SinputMessage;
    }
}
```

RowHeaderRenderer

Είναι ένα βοηθητικό αντικείμενο το οποίο χρησιμοποιείται στην εμφάνιση των ηχητικών μηνυμάτων. Ουσιαστικά αποτυπώνει έναν δυναμικό πίνακα και καλείται κάθε φορά από τον `ReceiveNewMessage`.

Παίρνει σαν παράμετρο εισόδου τον πίνακα και τον διαμορφώνει.

```
class RowHeaderRenderer extends JLabel implements ListCellRenderer {

    /**
     private static final long serialVersionUID = 1L;

    RowHeaderRenderer(JTable table) {
        JTableHeader header = table.getTableHeader();
        setOpaque(true);
        setBorder(UIManager.getBorder("TableHeader.cellBorder"));
        setHorizontalAlignment(CENTER);
        setForeground(header.getForeground());
        setBackground(header.getBackground());
        setFont(header.getFont());
    }
}
```

Server

Αποτελεί την κλάση με την οποία ξεκινάει ο server του συστήματος. Ανοίγει δυο port όπου περιμένει αιτήματα από πελάτες τα αποθηκεύει σε ένα vector για να ελέγχει και ποιοι είναι online. Από εδώ προωθεί το αίτημα τους.

```
public class Server {  
  
    final int portIn = 3822;  
    final int portOut = 3823;  
    final int portMessageOut=3824;  
    Vector<StockIp> clients = new Vector<StockIp>();  
  
    Server() {
```

ServerThread

Ο serverThread αποτελεί ουσιαστικά τη σύνδεση κάθε χρήστη ξεχωριστά με τον server. Από εδώ γίνεται η επικοινωνία και ο διαχωρισμός των διάφορων αιτημάτων που γίνονται από τους χρήστες της εφαρμογής. Επίσης ο serverthread κάνει την επικοινωνία και με τη βάση δεδομένων ενημερώνοντας την και αντλώντας αποτελέσματα από αυτήν. Δέχεται σαν παραμέτρους τα κανάλια επικοινωνίας με τον κάθε πελάτη, το vector με όλους τους συνδεδεμένους πελάτες και το statement για τη βάση.

```
ServerThread(Socket socketIn, Socket socketOut,Socket MessageIn,Vector<StockIp> clients,  
    java.sql.Statement stn) {  
    try {  
        this.stn = stn;  
        this.socketIn = socketIn;  
        this.socketOut = socketOut;  
        myInputStream = new ObjectInputStream(socketIn.getInputStream());  
        myOutputStream = new ObjectOutputStream(socketOut.getOutputStream());  
        myMessageInputStream= new ObjectOutputStream(MessageIn.getOutputStream());  
        this.clients=clients;  
  
        // We pass the thread itself in the bid object...  
  
        this.start();  
    } catch (IOException e) {  
        e.printStackTrace();
```

SimpleGLPanel

Η *SimpleGLPanel* αποτελεί την βασική κλάση της εφαρμογής. Καλείται αμέσως μετά το login και δημιουργεί την οθόνη που βλέπει ο χρήστης. Μέσα από αυτήν μπορεί να χειριστεί όλες τις λειτουργίες που έχουν περιγραφεί και πιο πάνω. Μέσα περιέχονται όλες αυτές οι μέθοδοι που καλούνται κάθε φορά που ο χρήστης πατάει κάποιο κουμπί και μέσα από εδώ χρησιμοποιούνται όπου πρέπει και οι υπόλοιπες κλάσεις. Κατά την εκκίνηση του παίρνει όλα τα δεδομένα από το Login που είναι τα username, μετοχές τελευταίου διαγράμματος, καθώς και τα κανάλια επικοινωνίας με τον Server. Πλέον από εδώ κινείται από τη πλευρά του client ολόκληρη η εφαρμογή.

```
public SimpleGLJPanel(LoginObject mylogin, ObjectInputStream Sinput,
    ObjectOutputStream Soutput, String myUsername,
    ObjectInputStream SinputMessage) {
    thisFrame = this;
    this.SinputMessage = SinputMessage;
    this.Sinput = Sinput;
    this.Soutput = Soutput;
    this.myUsername = myUsername;
    this.mylogin = mylogin;
    this.otherUsers = (LoginObject) mylogin.clone();
    initComponents();
    setTitle("Simple JOGL Application");
    design = new GLRenderer(allMetoxes, 0);
    System.out.println("typwnw");
    panel.addGLEventListener(design);
    animator = new Animator(panel);
    VoiceMessageButton.setVisible(false);
    this.addWindowListener(new WindowAdapter() {
```

Stock

Αποτελεί βοηθητικό στιγμιότυπο που χρησιμοποιείται κάθε φορά που καλείται μία μετοχή να δημιουργηθεί στο διάγραμμα. Αποτελείται από το όνομα της και την τιμή της εκείνη τη στιγμή.

```

public class Stock {

    String name;
    float count;

    public Stock(String name, float count) {
        this.name = name;
        this.count = count;
    }
}

```

StockIp

Η Stockip είναι μια κλάση που βρίσκεται και χρησιμοποιείται μόνο από τη πλευρά του Server. Κάθε φορά που στον server συνδέεται κάποιος από τους users γεμίζει ένα vector με τα συγκεκριμένα αντικείμενα όπου περιέχουν το username του χρήστη και το outputStream. Έτσι όταν στέλνεται ένα μήνυμα είτε ηχητικό είτε γραπτό αν ο χρήστης είναι συνδεδεμένος εκείνη τη στιγμή τότε αυτό στέλνεται άμεσα.

```

class StockIp {

    String username;
    ObjectOutputStream myOutMessageStream;

    public StockIp(String username, ObjectOutputStream myOutMessageStream) {
        super();
        this.username = username;
        this.myOutMessageStream = myOutMessageStream;
    }
}

```

5.4 Τρόπος υλοποίησης

Πριν ξεκινήσουν όλες οι διαδικασίες που περιγράφονται παρακάτω ο Server με τη πλευρά του πελάτη επικοινωνούν με δύο ServerSocket. Το ένα από αυτά χρησιμοποιείται για να στέλνονται αιτήματα στη βάση ενώ το άλλο για να δέχεται τις απαντήσεις. Επίσης δημιουργείται και ένα ακόμα serverSocket το οποίο βρίσκεται πάντα σε κατάσταση αναμονής δεδομένων από τη πλευρά του πελάτη και χρησιμοποιείται για την Online επικοινωνία.

5.4.1 Είσοδος Χρήστη

Κατά τη διαδικασία της εισόδου του χρήστη εμφανίζεται το γραφικό περιβάλλον μέσω της κλάσης loginSignIn. Αφού συμπληρωθούν τα στοιχεία από το χρήστη δημιουργείται ένα αντικείμενο QuerySql με περιεχόμενα ένα String με το ερώτημα που θα τεθεί στη βάση και το νούμερο 1 που βοηθάει το server να «καταλάβει» τι ζητάει ο χρήστης. Η πλευρά του πελάτη μπαίνει σε κατάσταση αναμονής μέχρι να στείλει θετική ή αρνητική απάντηση ο Server.

- Αν σταλεί θετική , δηλαδή ότι υπάρχει πελάτης με τα στοιχεία που δηλώθηκαν, δέχεται και ένα αντικείμενο LoginObject που περιέχει τις μετοχές του, τα μηνύματα του και το κωδικό πελάτη. Τα στοιχεία αυτά λαμβάνονται με τα κατάλληλα queries προς τη βάση και με αυτά αρχικοποιείται όλη η εφαρμογή μέσω του SimpleGLJPanel
- Αν σταλεί αρνητική αρχικοποιείται το username και το password και ξαναζητείται από το χρήστη για να δοθεί το σωστό.

5.4.2 Εγγραφή Χρήστη

Για να γίνει η εγγραφή του χρήστη στο σύστημα χρησιμοποιείται ένα αντικείμενο QuerySql στο οποίο αποθηκεύονται σε μορφή αλφαριθμητικών (String) το όνομα, το επώνυμο, το όνομα χρήστη, ο κωδικός και τέλος το email του ατόμου που θέλει να εγγραφεί. Επίσης τα στοιχεία αυτά τα συνοδεύει ο αριθμός 7 που χαρακτηρίζει τη συγκεκριμένη διαδικασία στη πλευρά του server.

Ο server όταν δέχεται το αίτημα νέας εγγραφής ελέγχει αν υπάρχει κάποιος ήδη εγγεγραμμένος με το ηλεκτρονικό ταχυδρομείο ή το όνομα χρήστη που δηλώθηκε. Αν υπάρχει σταματάει τη διαδικασία και στέλνει μήνυμα στο πελάτη να αλλάξει τα στοιχεία. Αν δεν υπάρχει δημιουργείται το ερώτημα και γίνεται η εισαγωγή του στη βάση δεδομένων. Ένα νέο Login object αντικείμενο στέλνεται για να ξεκινήσει η δημιουργία της κεντρικής εικόνας γραφικού και επιλογών στο χρήστη. Πλέον σε κάθε επίσκεψη του δεν χρειάζεται να επαναλάβει αυτή τη διαδικασία.

5.4.3 Τελευταίο Διάγραμμα

Η διαδικασία για να δει το τελευταίο διάγραμμα μετοχών ξεκινάει όταν ο χρήστης πατήσει μέσα από το menu την επιλογή «Last Graf» . Μέσα από το LoginObject που έχει λάβει ο χρήστης όταν έκανε την εισαγωγή στο σύστημα παίρνει το πίνακα που περιέχονται τα ονόματα των μετοχών που είχε δει την τελευταία φορά. Αν ο πίνακας είναι μηδενικός τότε ζητάτε από το χρήστη να δημιουργήσει το πρώτο του διάγραμμα. Διαφορετικά καλείται η μέθοδος Yahoo() όπου για κάθε μετοχή παίρνει την τιμή της εκείνη τη στιγμή μέσω

```
URL u = new URL("http://finance.yahoo.com/q?s=" + metoxi);
URLConnection uc = u.openConnection();
String contentType = uc.getContentType();
```

Κάθε φορά που μια τιμή λαμβάνεται δημιουργείται ένα αντικείμενο Stock που αρχικοποιείται με το όνομα της μετοχής και τη τιμή της. Με αυτά τα αντικείμενα γεμίζει ένα Vector που στέλνεται όταν έχουμε πάρει τις τιμές όλων των μετοχών στο GLRenderer για να δημιουργηθεί το γραφικό. Αναλογικά με τις

τιμές είναι και το μέγεθος της ράβδου στο διάγραμμα. Επίσης σε αυτή τη διαδικασία καλείται και η StartLogin μέθοδος που περνάει τα μηνύματα κειμένου στο κουτί σχολίων.

```
jTextArea1.setFont(langFont);  
jTextArea1.append(show.message.elementAt(j).username + ":"  
+ show.message.elementAt(j).message + "\n");
```

5.4.4 Δημιουργία νέου διαγράμματος

Για τη δημιουργία νέου διαγράμματος χρησιμοποιείται η μέθοδος NewDiagram. Σαν παραμέτρους παίρνει το κουτί σχολίων για να το καθαρίσει αν διαλέξει νέες μετοχές, το vector με τις μετοχές για να το γεμίσει με τις νέες, το ήδη υπάρχον παράθυρο για να μπορέσει να γίνει η αλλαγή γραφικού και τέλος το ObjectOutputStream που θα επικοινωνήσει με τον server. Η NewDiagram δημιουργεί ένα παράθυρο μέσω του οποίου επιλέγονται οι νέες μετοχές. Με κάθε όνομα που πληκτρολογείται και πατώντας το κουμπί add συνδέεται με την κλάση HttpReceiver για να λάβει τη τιμή της μετοχής εκείνη τη στιγμή, να δημιουργήσει το αντικείμενο Stock και τέλος να γεμίσει τον κεντρικό vector με τις μετοχές.

```
myInputStream = uc.getInputStream();  
myReader = new BufferedReader(new InputStreamReader(myInputStream, encoding));  
recv = new HttpReceiver(responseArea, myReader, allMetoxes, myArea.getText());  
recv.start();
```

Όταν τελειώσει στέλνεται μέσω του QuerySql η νέα λίστα μετοχών του χρήστη στον server με το χαρακτηριστικό αριθμό «2» για να αναγνωριστεί η ενέργεια που γίνεται και να γίνουν τα σωστά query στη βάση δεδομένων. Τέλος ανανεώνεται το διάγραμμα στη κεντρική εικόνα της εφαρμογής με τις τιμές των νέων μετοχών που έχουν επιλεγεί και ένα καινούργιο comment box χωρίς μηνύματα.

Ταυτόχρονα γίνονται και τα βήματα από τη πλευρά του server που είναι να απενεργοποιήσει τις μετοχές που υπήρχαν στο παλιό του διάγραμμα βάζοντας

στο isActivate πεδίο τη τιμή 0, καθώς και στα μηνύματα που υπήρχαν στο comment area. Έπειτα συνδέει τις καινούργιες μετοχές με τον χρήστη και δημιουργώντας νέες σχέσεις στον πίνακα Customer_Stock.

```
stn.executeUpdate("UPDATE Customer_Stock SET isActivate=0 Where isActivate=1 and customerID="+
+ customerID);
stn.executeUpdate("UPDATE Message SET isActivate=0 Where isActivate=1 and customerID="+
+ customerID);
for (int i = 0; i < myquery.metoxes.size(); i++) {
String name = "SELECT StockID, StockName FROM Stock WHERE StockName='"
+ myquery.metoxes.elementAt(i) + "'";
ResultSet result2 = stn.executeQuery(name);
if (result2.next() == true) {
String idmetoxis = result2.getString(1);
stn.executeUpdate("INSERT INTO Customer_Stock VALUES ("
+ idmetoxis + "," + customerID + ",1)");
} else {

stn.executeUpdate("INSERT INTO Stock VALUES ('"
+ myquery.metoxes.elementAt(i) + "')");
System.out.println("executeupdate2");
ResultSet result = stn
.executeQuery("SELECT StockID, StockName FROM Stock WHERE StockName='"
+ myquery.metoxes.elementAt(i) + "'");

result.next();
String idmetoxis=result.getString(1);
stn.executeUpdate("INSERT INTO Customer_Stock VALUES ("
+ idmetoxis + "," + customerID + ",1)");
```

5.4.5 Εμφάνιση διαγράμματος άλλου χρήστη

Για να παρουσιαστεί το διάγραμμα κάποιου άλλου χρήστη πρέπει αρχικά να πατηθεί η επιλογή Other user από το κεντρικό μενού. Τότε ενεργοποιείται η μέθοδος otherUserDiagramListener όπου δημιουργεί ένα JOptionPane για να γραφτεί το username του χρήστη του οποίου το διάγραμμα θέλουμε να εμφανιστεί. Το username αποθηκεύεται σε ένα αλφαριθμητικό(String) όπου χρησιμοποιείται για να δημιουργηθεί το ερώτημα προς το server και κατ επέκταση στη βάση δεδομένων. Έπειτα στέλνεται ένα QuerySql αντικείμενο με το χαρακτηριστικό «1» για να αναγνωριστεί η ενέργεια και το ερώτημα.

Η πλευρά του Server όταν το παραλάβει ελέγχει αν υπάρχει ο χρήστης με το συγκεκριμένο username και παίρνει με νέα ερωτήματα προς τη βάση τις μετοχές και τα μηνύματα του τελευταίου διαγράμματος του. Με τα δεδομένα αυτά δημιουργεί ένα LoginObject αντικείμενο και στέλνεται πίσω στον πελάτη.

Αν δεν υπάρχει ο χρήστης επιστρέφει μήνυμα λάθους στον πελάτη για να επιλέξει άλλο όνομα χρήστη.

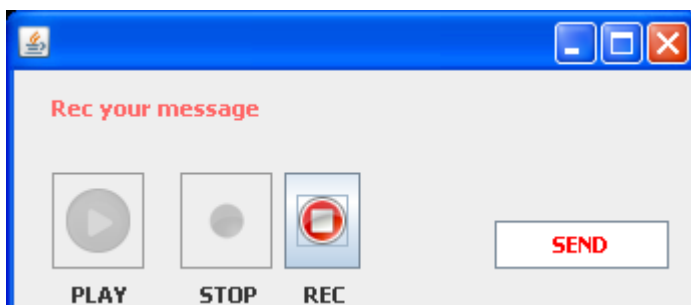
Μέσο του `ObjectInputStream` η πλευρά του client δέχεται τις μετοχές και τα μηνύματα του χρήστη του οποίου το διάγραμμα ενδιαφέρεται να παρακολουθήσει και καλεί τις μεθόδους `Yahoo` και `StartLogin` με τη βοήθεια των οποίων δημιουργείται το διάγραμμα. Σημαντικό είναι ότι σε αυτή τη περίπτωση εμφανίζεται και το κουμπί ηχογράφησης μηνύματος με τη χρήση του `SetVisiblue(true)` στο κουμπί.

5.4.6 Ηχογράφηση νέου μηνύματος

Η ηχογράφηση των μηνυμάτων γίνεται με τη χρήση του κουμπιού `CALL` που εμφανίζεται μόνο αν έχει επιλεγεί να παρακολουθήσει ο χρήστης κάποιου άλλου πελάτη το διάγραμμα. Όταν πατηθεί το κουμπί καλείται η μέθοδος `CaptureVoice` με παραμέτρους το `ObjectOutputStream` για να επικοινωνήσει με τον server το `username` το δικό του και το `username` του χρήστη στον οποίο στέλνει το ηχητικό μήνυμα.

```
new CaptureVoice(Soutput, usernameOtherUser, myUsername);
```

Έπειτα δημιουργείται το παρακάτω παράθυρο δίνοντας τις επιλογές ηχογράφησης, ακρόασης και αποστολής του ηχητικού μηνύματος.



Εικόνα 9- Ηχογράφηση ηχητικού μηνύματος

Αρχικά η μονή ενεργή επιλογή είναι το `rec` όπου επιτρέπει στο χρήστη να ηχογράφηση το μήνυμά του. Όταν πατηθεί το κουμπί καλείται η μέθοδος `captureAudio()` όπου μετατρέπει τον ήχο σε `byte` και γεμίζει ένα πίνακα (`buffer[]`). Ο πίνακας αυτός αποθηκεύεται σε ένα `ByteArrayOutputStream` και είναι έτοιμο να σταλεί μέσω του `Soutput` στον `Server`.

Το κουμπί `Stop` χρειάζεται για να σταματήσει η ηχογράφηση του ηχητικού μηνύματος και όταν ο χρήστης το επιλέξει η διαδικασία ενεργοποιείται κάνοντας τη βοηθητική μεταβλητή `running` να ισούται με `false`.

Η επιλογή `Play` βοηθάει το χρήστη στην ακρόαση του μηνύματος πριν σταλεί και στη διόρθωση πιθανών λαθών. Η διαδικασία ξεκινάει με τη βοήθεια της `playAudio()` μεθόδου όπου κάνει μετατροπή των `bytes` σε κατάλληλη μορφή ώστε να γίνει ήχος.

Τέλος με την επιλογή `send` καλείται η μέθοδος `sendButtonActionPerformed` όπου δημιουργεί ένα αντικείμενο `QuerySql` με δεδομένα το `username` του χρήστη που στέλνει το μήνυμα, το `username` του χρήστη στον οποίο απευθύνεται το ηχητικό μήνυμα, το ηχητικό μήνυμα σε κατάλληλη μορφή και τέλος το χαρακτηριστικό νούμερο 4 όπου βοηθάει το χρήστη να αναγνωρίσει την ενέργεια. Έπειτα μέσω `ObjectOutputStream Soutput` στέλνεται στον `server`.


```
QuerySql minima = new QuerySql(4, usernameOtherUser,  
    out.toByteArray(), myUserName);  
Soutput.writeObject(minima);
```

Η πλευρά του `Server` μόλις λάβει τα ηχητικά δεδομένα δημιουργεί ένα αρχείο κειμένου με βάση την ημερομηνία και την ώρα αποστολής του μηνύματος και το αποθηκεύει στο φάκελο με το όνομα του χρήστη. Ο λόγος που δημιουργείται το αρχείο κειμένου είναι για να γραφτούν μέσα τα `byte` ήχου. Έτσι πλέον το ηχητικό μήνυμα έχει μετατραπεί σε ένα αρχείο κειμένου του οποίου το `Path` αποθηκεύεται μέσα στη βάση δεδομένων. Ταυτόχρονα μέσα από το `Vector` που φαίνεται ποιοι χρήστες είναι σε συνδεση ελέγχεται αν ο χρήστης στον οποίο απευθύνεται το μήνυμα βρίσκεται συνδεδεμένος. Αν βρίσκεται τότε αποστέλλεται αμέσως με τη χρήση του `ObjectOutputStream` που έχει δημιουργηθεί μόνο για την `Online` επικοινωνία.

5.4.7 Δημιουργία γραπτού μηνύματος

Η λειτουργία του γραπτού μηνύματος είναι από τις πιο απλές τις εφαρμογής. Κάθε γραπτό μήνυμα συνδέεται με το comment area και το comment area με το αντίστοιχο διάγραμμα. Όταν θέλει ο χρήστης να στείλει το σχόλιο του το γράφει στο textArea και πατάει το κουμπί comment. Έτσι ενεργοποιείται η μέθοδος jButton1ActionPerformed όπου παίρνει σε ένα αλφαριθμητικό το κείμενο από το TextField το όνομα χρήστη του παραλήπτη και το όνομα χρήστη του αποστολέα. Με τα στοιχεία αυτά φτιάχνεται το ερώτημα προς τη βάση το οποίο μπαίνει μέσα σε ένα αντικείμενο QuerySql και στέλνεται μέσω του ObjectOutputStream στο server.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String commentText = jTextField1.getText();  
    System.out.println(otherUsers.customerID + "dsadacda");  
    String newquery = "INSERT INTO Message VALUES('" + commentText + "','"  
    + otherUsers.customerID + "','" + myUsername + "','"  
    + date + ",1)";  
    QuerySql mysql = new QuerySql(newquery, 3, null, usernameOtherUser,  
    commentText);  
    try {  
        Soutput.writeObject(mysql);  
        Soutput.flush();  
    }  
}
```

Αξίζει να σημειωθεί ότι στο ερώτημα εισάγει και μια ημερομηνία. Η ημερομηνία αυτή αποτελεί έναν περιορισμό ο οποίος μπαίνει από τον αποστολέα και ορίζει από πιο χρονικό διάστημα και μετά θα εμφανίζεται στο comment area το σχόλιο. Ο περιορισμός αυτός ορίζεται από το κουμπί που βρίσκεται δίπλα στο comment  όπου δημιουργεί και εμφανίζει το παρακάτω παράθυρο. Όποια ημερομηνία διαλέξει με το ποντίκι ο χρήστης αυτή αποθηκεύεται μέσα στο ερώτημα και μετά από αυτή θα εμφανιστεί το μήνυμα στο comment area.



5.4.8 Ακρόαση ηχητικών μηνυμάτων

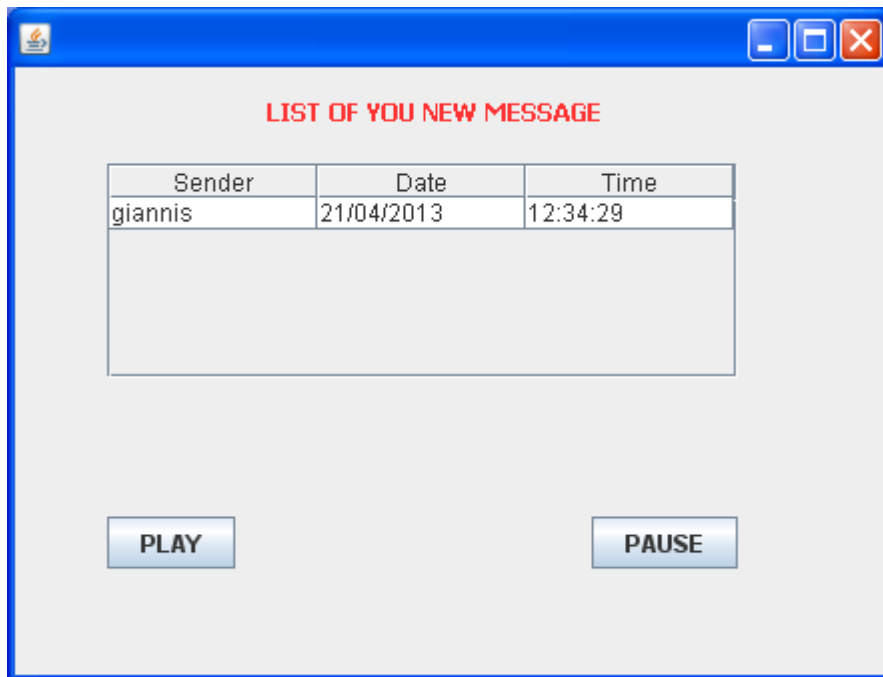
Η ακρόαση των ηχητικών μηνυμάτων μπορεί να χωριστεί σε δύο κατηγορίες. Η πρώτη αφορά τα νέα μηνύματα δηλαδή αυτά που δεν έχει ακούσει καμία φορά, ενώ η δεύτερη τα παλαιά δηλαδή αυτά που έχει γίνει η ακρόαση τους έστω μία φορά.

➤ Νέα Μηνύματα

Για να δει ο χρήστης αν έχει νέα μηνύματα από το μενού επιλέγει το κουμπί Messages→New Messages. Τότε καλείται η μέθοδος newVoiceMessage όπου δημιουργεί το query που απευθύνεται στο server και κατ επέκταση στη βάση δεδομένων. Ζητούνται όλα τα μηνύματα τα οποία στη βάση χαρακτηρίζονται με το isActivate=1 δηλαδή είναι καινούργια, καθώς και το path από όπου θα το διαβάσει. Αυτό στέλνεται με τη βοήθεια ενός αντικειμένου QuerySql και με το χαρακτηριστικό νούμερο 5 όπου βοηθάει το server να αναγνωρίσει την ενέργεια που πρέπει να γίνει. Από τη πλευρά του server αφού γίνει το ερώτημα στη βάση παίρνει τα νέα μηνύματα και τα βάζει μέσα σε ένα vector το οποίο και στέλνει πίσω στον πελάτη. Μόλις τα λάβει η μεριά του χρήστη καλεί τη μέθοδο ListenMessage με παραμέτρους το vector με τα μηνύματα το ObjectOutputStream με το οποίο θα επικοινωνήσει με τον server για να ενημερωθεί η βάση όταν ακούσει τα μηνύματα.

```
private void newVoiceMessage(ActionEvent evt) {
    try {
        System.out.println("new voice messages");
        QuerySql myquery = new QuerySql(
            "SELECT VoiceMessageID, VoicePath, Date, Time, Sender FROM VoiceMessage WHERE customerID="
            + mylogin.getCustomerId() + " AND isActivate=1", 5,
            null);
        Soutput.writeObject(myquery);
        Soutput.flush();
        voiceMessage = (Vector<Messages>) Sinput.readObject();
        new ListenMessage(voiceMessage, "New", Soutput);
    } catch (IOException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
    }
}
```

Η κλάση ListenMessage δημιουργεί το γραφικό περιβάλλον το οποίο θα χρησιμοποιήσει ο χρήστης για να ακούσει τα νέα του μηνύματα.



Εικόνα 10- Παράθυρο ακρόασης ηχητικών μηνυμάτων

Κάθε φορά που πατάει να ακούσει ένα μήνυμα γίνεται μετατροπή του file σε byte και γίνεται η ακρόαση. Μόλις τελειώσει στέλνεται ένα ερώτημα στο server για να αλλάξει το status του μηνύματος και πλέον να είναι παλιό.

➤ Παλαιά Μηνύματα

Η διαδικασία για την ακρόαση των παλαιών μηνυμάτων είναι παρόμοια με την ακρόαση των με μικρές διαφορές στο κώδικα. Αρχικά ο χρήστης πρέπει να πατήσει από το μενού την επιλογή Messages→Search. Τότε εμφανίζεται ένα νέο παράθυρο στο οποίο εισάγεται το όνομα χρήστη του οποίου τα παλαιά μηνύματα θα εμφανιστούν. Δημιουργείται το νέο query όπου στέλνεται στον server με τη βοήθεια ενός αντικειμένου QuerySql και με το χαρακτηριστικό 5 που αναγνωρίζεται η ενέργεια από τον server. Εκεί ζητούνται τα μηνύματα από το όνομα χρήστη που έχει επιλεγεί λίγο πριν και τα οποία έχουν στη βάση τη τιμή 0 στο πεδίο isActivate. Από εκεί γεμίζει ένα vector με τα μηνύματα και

στέλνεται πίσω στο client. Καλείται και πάλι η κλάση ListenMessage για να εμφανιστεί το γραφικό. Η διαδικασία από εδώ και κάτω είναι η ίδια όπως και στην ακρόαση νέων μηνυμάτων.

6. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ

Το stock market δίνει αρκετές δυνατότητες στο χρήστη οι κυριότερες από τις οποίες είναι οι εξής:

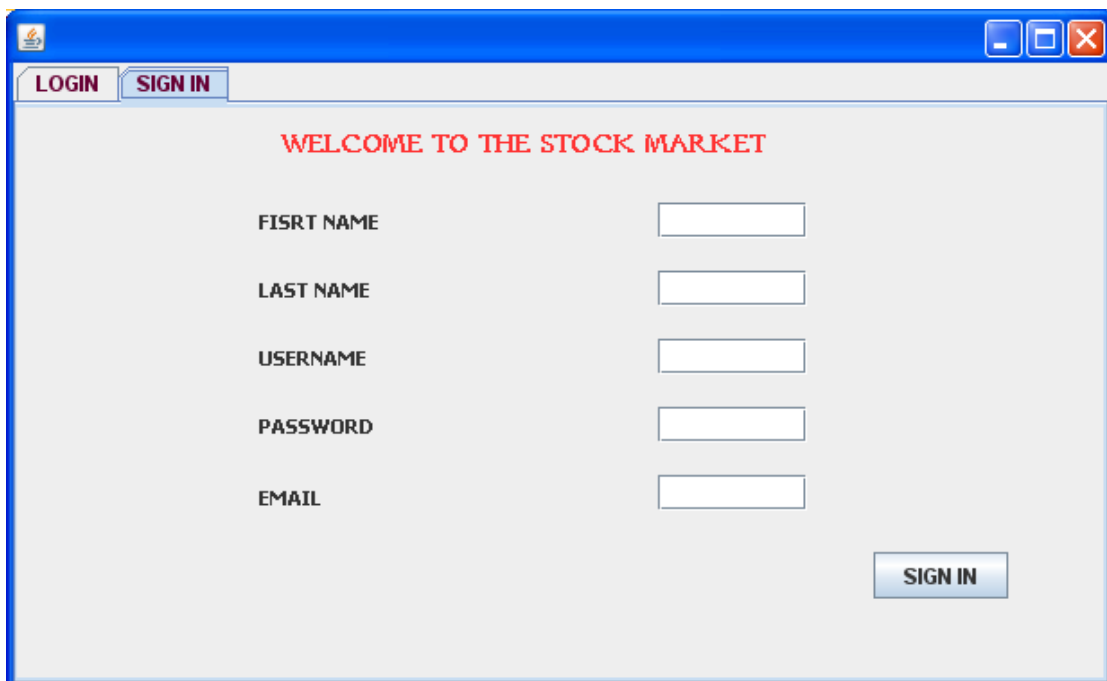
- Εγγραφή του χρήστη στο σύστημα και δημιουργία του δικού του προσωπικού προφίλ.
- Εισαγωγής στο σύστημα με χρήση των προσωπικών του κωδικών.
- Να δει τη τελευταία λίστα μετοχών που έχει παρακολουθήσει κατά το τελευταίο login.
- Να δει τη λίστα μετοχών που παρακολουθήσει κάποιος άλλος χρήστης.
- Να αφήσει κάποιο μήνυμα στο comment area του χρήστη κάποιου άλλου χρήστη ή στο δικό του.
- Να δημιουργήσει μία καινούργια λίστα μετοχών για παρακολούθηση.
- Να αφήσει κάποιο ηχητικό μήνυμα σε κάποιο άλλο χρήστη
- Να ακούσει τα καινούργια ηχητικά μηνύματα που του έχουν αφήσει άλλοι χρήστες.
- Να ακούσει παλιά ηχητικά μηνύματα.

Δημιουργία λογαριασμού

Κάθε χρήστης πριν ξεκινήσει να χρησιμοποιεί την εφαρμογή θα πρέπει να κάνει εγγραφή στο σύστημα ακολουθώντας τα παρακάτω βήματα.

- 1) Επιλέξτε από την αρχική εικόνα την καρτέλα «SIGN IN» όπου θα σας οδηγήσει στην καρτέλα εγγραφής.
- 2) Στη συνέχεια συμπληρώνεται όλα τα στοιχεία σας που είναι όνομα (First name), επώνυμο (Last Name), όνομα χρήστη (Username), κωδικός (Password), διεύθυνση ηλεκτρονικού ταχυδρομείου (Email).

Στη παρακάτω εικόνα φαίνεται η φόρμα εγγραφής του χρήστη



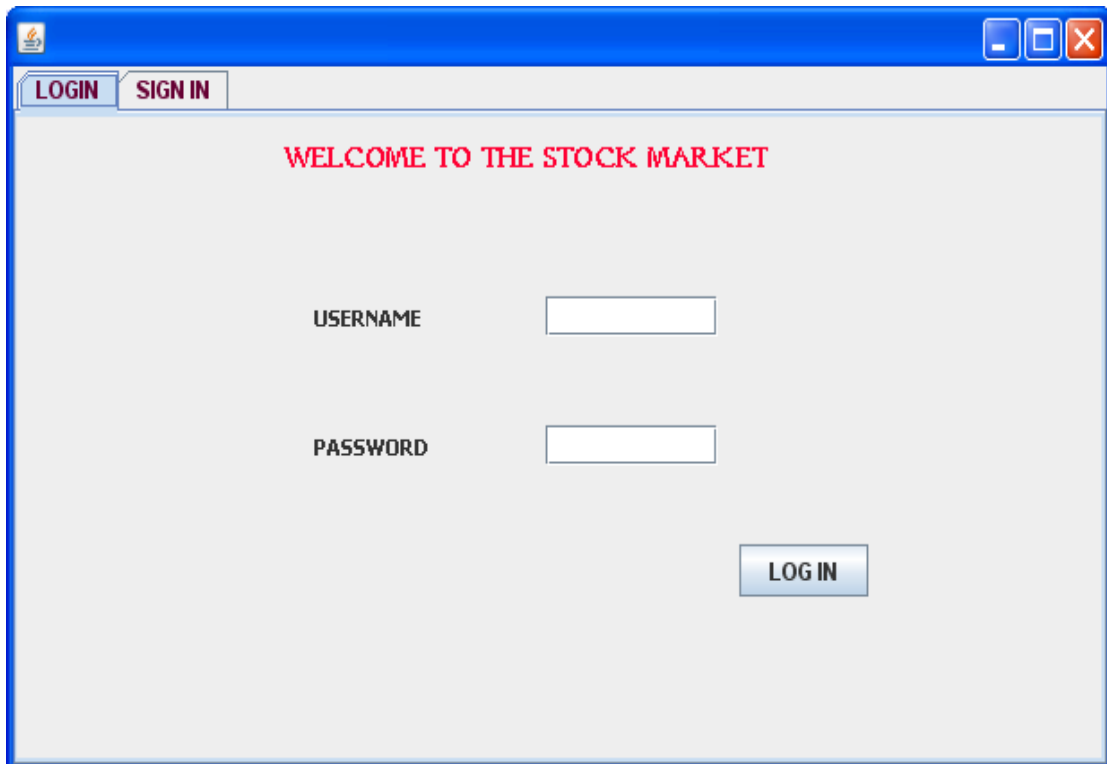
The screenshot shows a web application window with a blue title bar. At the top left, there are two tabs: 'LOGIN' and 'SIGN IN', with 'SIGN IN' being the active tab. Below the tabs, the text 'WELCOME TO THE STOCK MARKET' is displayed in red. The main content area contains five input fields, each with a label to its left: 'FIRST NAME', 'LAST NAME', 'USERNAME', 'PASSWORD', and 'EMAIL'. A 'SIGN IN' button is located at the bottom right of the form area.

Εικόνα 11-Φορμα Εγγραφής Χρήστη

Σημείωση: Για να πραγματοποιηθεί η εγγραφή θα πρέπει να συμπληρωθούν όλα τα πεδία. Επίσης επειδή το username και το email θα πρέπει να είναι μοναδικό για κάθε χρήστη αφού τον ταυτοποιεί στο σύστημα. Αν κάτι από τα δύο δεν έχει γίνει σωστά θα εμφανιστεί μήνυμα λάθους και η διαδικασία θα ξεκινήσει από την αρχή.

Εισαγωγή Χρήστη

Αφού αποκτήσετε το δικό σας λογαριασμό θα πρέπει κάθε φορά που θέλετε να εισέλθετε στο σύστημα να πληκτρολογήσετε στην αρχική σελίδα το Όνομα χρήστη(User name) και τον Κωδικό(Password). Η εισαγωγή στο σύστημα ολοκληρώνεται με το πάτημα του κουμπιού LOG IN που δίνει εξουσιοδοτημένη πρόσβαση σε όλες τις λειτουργίες του κυρίως μενού του stock market.

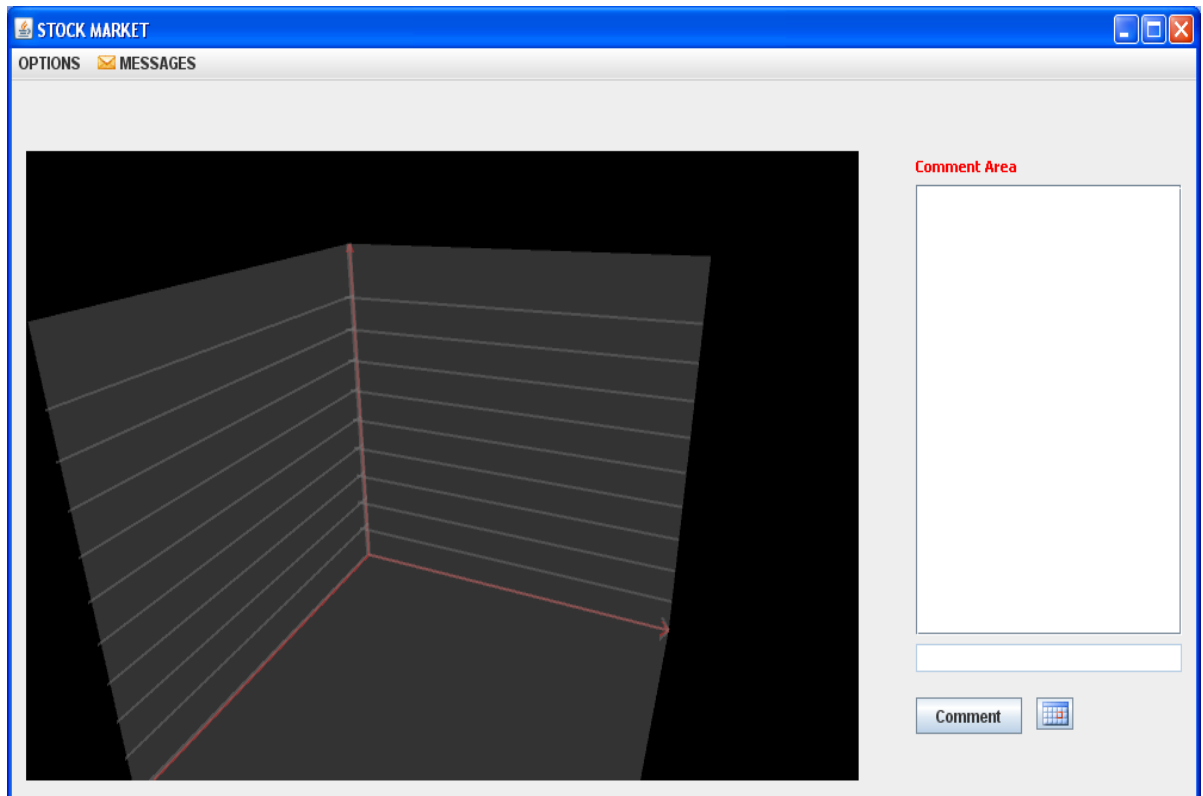


The image shows a web browser window with a blue title bar. Inside the window, there are two tabs: 'LOGIN' and 'SIGN IN'. The 'LOGIN' tab is active. The main content area has a light gray background and displays the text 'WELCOME TO THE STOCK MARKET' in red. Below this text, there are two input fields: 'USERNAME' and 'PASSWORD', each with a corresponding text label to its left. A 'LOGIN' button is positioned below the password field.

Εικόνα 12- Φόρμα Εισαγωγής Χρήστη

Αφού λοιπόν κάνετε την εισαγωγή στο σύστημα, εμφανίζεται η αρχική σελίδα της εφαρμογής και αναμένει για τις επόμενες επιλογές. Στο κέντρο του

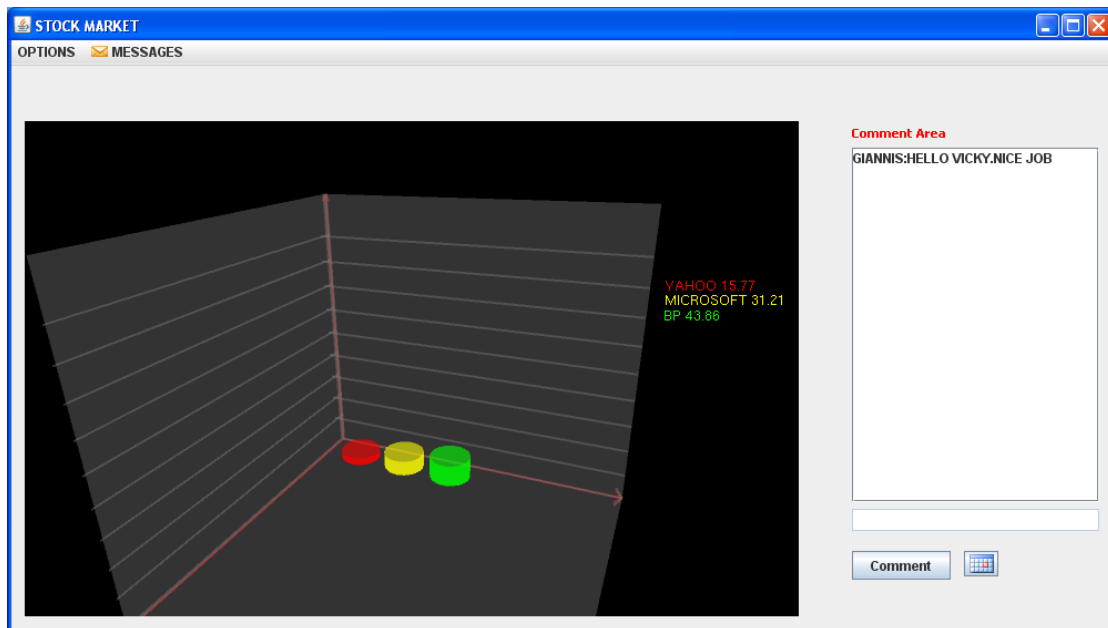
παραθύρου είναι ο χώρος όπου εμφανίζεται το γραφικό από τις μετοχές και πιο δεξιά το σημείο όπου αφήνουν τα μηνύματα τους οι χρήστες. Επίσης από εδώ επιλέγετε τα επόμενα βήματα του όπως είναι η δημιουργία καινούργιου γραφικού, να δείτε γραφικά άλλων χρηστών και διάφορα άλλα που θα τα δούμε αναλυτικά παρακάτω.



Εικόνα 13- Αρχική Εικόνα Εφαρμογής

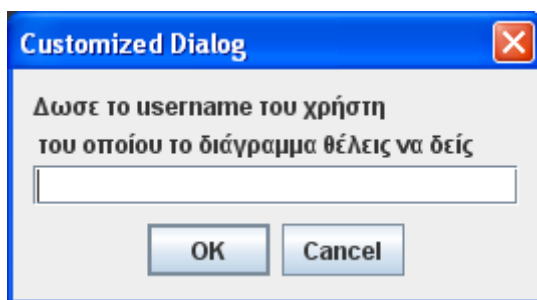
Εμφάνιση Προφίλ

Μέσα από την αρχική σελίδα και το κυρίως μενού πατάτε την επιλογή last graf για να εμφανιστεί το προσωπικό σας προφίλ μετοχών. Σε αυτό συμπεριλαμβάνεται το διάγραμμα των μετοχών και οι τιμές τους εκείνη τη στιγμή, καθώς και τα μηνύματα που έχουν αφήσει άλλοι χρήστες στο Comment Area. Η εικόνα φαίνεται και παρακάτω.




Εικόνα 14- Ενδεικτική αρχική εικόνα προφίλ


Κάτι αντίστοιχο γίνεται και όταν θέλετε να δείτε το διάγραμμα κάποιου άλλου χρήστη. Επιλέγεται από το menu αυτή τη φορά το otherUser όπου εμφανίζεται ένα άλλο παράθυρο. Εκεί συμπληρώνετε το όνομα χρήστη αυτού που θέλετε να δείτε και τέλος πατάει Ok.

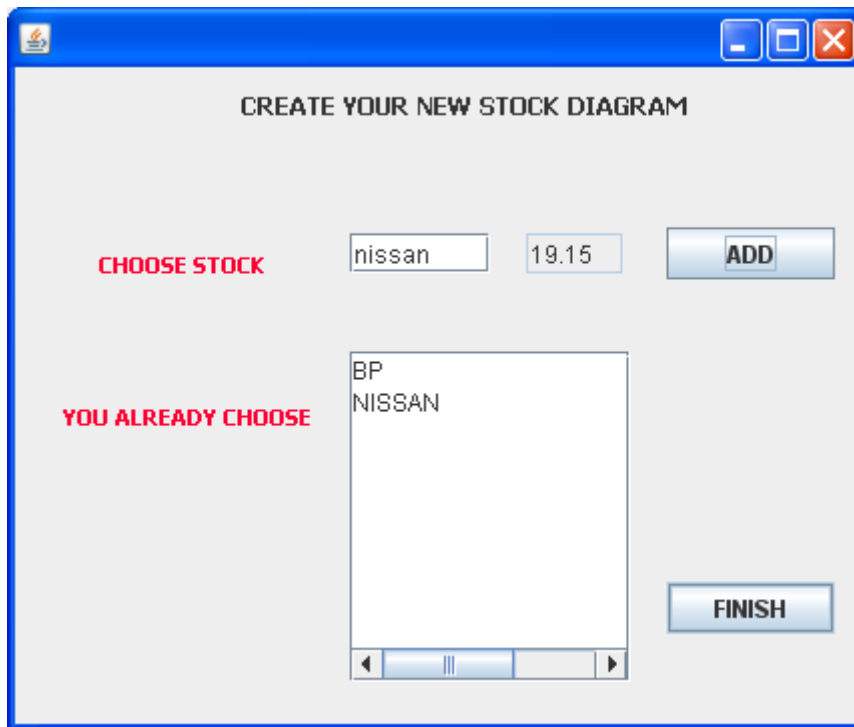


Εικόνα 15 -Φόρμα Επιλογής Χρήστη

Εκεί εμφανίζεται το ίδιο παράθυρο όπως πριν με την μόνη διαφορά ότι στο παράθυρο θα έχει προστεθεί το κουμπί  για να μπορείτε να αφήνετε ηχητικά μηνύματα. Αυτό όμως θα το δούμε παρακάτω.

Δημιουργία νέου διαγράμματος

Για τη δημιουργία νέου διαγράμματος αφού έχει γίνει login από τις επιλογές κάνετε κλικ στο create new. Εμφανίζεται έτσι ένα νέο παράθυρο που θα βοηθήσει στη δημιουργία του διαγράμματος. Στο πρώτο κουτί πληκτρολογείται το όνομα της μετοχής που θέλετε να προστεθεί στο διάγραμμα σας και πατάτε το κουμπί  . Τότε θα εμφανιστεί η τιμή της μετοχής και ταυτόχρονα θα τοποθετηθεί το όνομα της στη λίστα στο κέντρο του παραθύρου. Τη διαδικασία αυτή συνεχίζετε μέχρι να έχετε προσθέσει όλες σας τις μετοχές στο διάγραμμα. Μόλις ολοκληρωθεί η δημιουργία πατάμε το κουμπί τέλος (finish) .




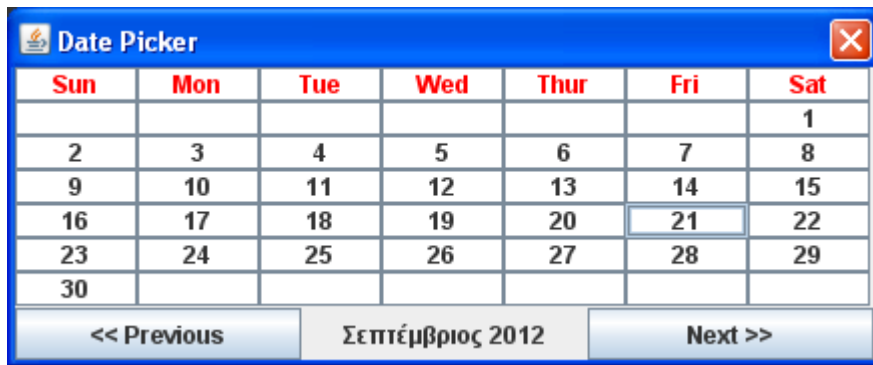
Εικόνα 16- Φόρμα Δημιουργίας Νέου Διαγράμματος

Μήνυμα στη περιοχή σχολίων

Η διαδικασία για να μπορέσετε να αφήσετε μήνυμα στο comment area το δικό σας ή κάποιου άλλου χρήστη είναι πολύ απλή. Αφού έχετε εμφανίσει το διάγραμμα που θέλετε κάτω από το σημείο που εμφανίζονται τα υπόλοιπα μηνύματα πληκτρολογείτε το δικό σας και πατάτε το κουμπί comment. Αυτό αυτόματα εμφανίζεται στο κουτί των σχολίων έχοντας μπροστά το username σας για να φαίνεται ποιος το έχει στείλει.

Σημείωση

Αν θέλετε μπορείτε να θέσετε χρονικό περιορισμό στην εμφάνιση του μηνύματος σας. Αυτό γίνεται με το κουμπί που υπάρχει δίπλα ακριβώς στο κουμπί comment  το οποίο εμφανίζει ένα ημερολόγιο. Μέσα από αυτό μπορείτε να αποφασίσετε από ποια ημερομηνία και μετά θα εμφανιστεί το μήνυμα σας. Η εικόνα του ημερολογίου φαίνεται και παρακάτω.



Εικόνα 17 -Επιλογή χρονικού περιορισμού εμφάνισης μηνυματος

Ακρόαση ηχητικών μηνυμάτων

Μια ακόμα δυνατότητα που σας παρέχει το πρόγραμμα είναι να ακούσετε τα ηχητικά μηνύματα που σας έχουν αφήσει άλλοι χρήστες. Μπορείτε να επιλέξετε ανάμεσα από δύο τύπους μηνυμάτων. Ο ένας είναι να ακούσετε τα νέα σας μηνύματα και ο άλλος να ψάξετε για παλιά. Και για τα δύο η διαδικασία είναι η ίδια. Από το μενού επιλέγεται το Message και μετά New messages η search αντίστοιχα για νέα μηνύματα και για παλιά. Αν πατήσετε νέα μηνύματα θα σας εμφανιστεί ένα παράθυρο με μία λίστα από τα νέα μηνύματα που έχετε χαρακτηρίζοντας τα με το όνομα χρήστη . Έτσι για παράδειγμα στην παρακάτω εικόνα υπάρχει ένα νέο μήνυμα από τον χρήστη vicky5 που στάλθηκε την 1/4/2013 και ώρα 22:36:54.



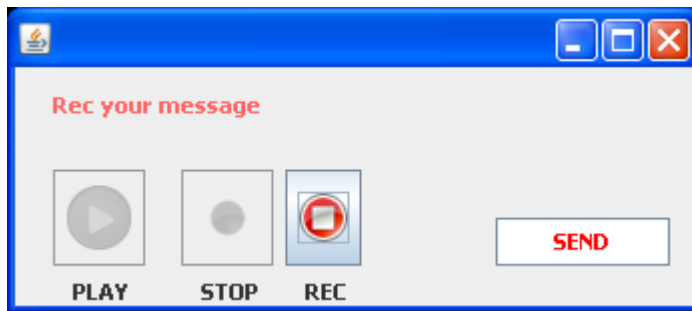
Εικόνα 18- Ακρόαση ηχητικών μηνυμάτων

Για να ακούσετε παλιά μηνύματα ακολουθείς την ίδια ακριβώς διαδικασία με την μόνη διαφορά ότι πατάς Message→Search. Έπειτα πληκτρολογείς το **username** του χρήστη του οποίου τα παλιά μηνύματα θέλεις να ακούσεις και το παραπάνω παράθυρο θα εμφανιστεί

Αποστολή ηχητικών μηνυμάτων

Για να κάνετε αποστολή νέου ηχητικού μηνύματος η διαδικασία είναι η εξής . Αφού έχετε κάνει login στο πρόγραμμα θα πρέπει από το Menu να επιλέξετε να δείτε το διάγραμμα του χρήστη στον οποίο θέλετε να στείλετε το μήνυμα. Έπειτα θα εμφανιστεί το κουμπί call. Αν το πατήσετε θα σας εμφανιστεί ένα νέο παράθυρο με 4 κουμπιά. PLAY-STOP-REC-SEND. Με το κουμπί rec κάνετε εγγραφή του μηνύματος. Με το stop σταματάτε την εγγραφή όταν έχετε

τελειώσει. Με το play ξαναπαίζει το μήνυμα για να το ακούσετε πριν αποσταλεί και τέλος με το send το γίνεται η αποστολή



Εικόνα 19 -Ηχογράφηση ηχητικού μηνύματος

7. ΜΕΛΛΟΝΤΙΚΕΣ ΕΞΕΛΙΞΕΙΣ

Η παρούσα εργασία καλύπτει τις πιο βασικές λειτουργίες μιας Social media εφαρμογής. Φυσικά και θα μπορούσαν να υπάρξουν βελτιώσεις που θα αύξαναν την αποδοτικότητα και χρησιμότητα της.

1) Αλλαγή του γραφικού περιβάλλον

Για το γραφικό περιβάλλον την εφαρμογής όπως αναφέρθηκε και πιο πάνω έχει χρησιμοποιηθεί το πακέτο swing που είναι τμήμα της jfc (java foundation classes) . Οι αρνητικές συνέπειες που έχει όπως η χρησιμοποίηση πόρων από τον υπολογιστή του χρήστη και η δύσκολη επεξεργασία τους μπορεί να αντιμετωπιστεί με την αντικατάστασή τους με Java Servlet.

Ο Java Servlet είναι η αρχιτεκτονική, η οποία χρησιμοποιεί τις δυνατότητες του εξυπηρετητή Apache Tomcat, όπου είναι εγκατεστημένη στον εξυπηρετητή δέχεται δεδομένα μέσω του πρωτοκόλλου HTTP. Την απάντηση στα ερωτήματα τα στέλνει στο πρόγραμμα περιήγησης με χρήση αρχείων τύπου .html. Τα servlet είναι ανεξάρτητα πλατφόρμας , είναι ασφαλή ,φορτώνονται μια φορά και μπορούν να αντλούν εύκολα στοιχεία από βάσεις δεδομένων

2) Δημιουργία Mobbile application

Μια σημαντική εξέλιξη της εφαρμογής θα ήταν η δυνατότητα πρόσβασης και μέσω φορητών συσκευών όπως είναι τα κινητά και τα tablet. Αυτό θα διευκόλυνε τους χρήστες στην ενημέρωσή τους Online για τις τιμές των μετοχών τους καθώς και για την άμεση απάντηση σε τυχόν μηνύματα. Στην ανάπτυξη μια τέτοιας εφαρμογής θα χρησιμοποιηθεί η τεχνολογία J2ME.

Αξίζει να αναφερθεί ότι οι χρήστες τέτοιων συσκευών αυξάνονται με τέτοιους ρυθμούς που πλέον έχουν ξεπεράσει το 1 δις.

3) *Σύνδεση με Online Αγορές*

Η εφαρμογή θα μπορούσε να γίνει ακόμα πιο χρήσιμη αν μελλοντικά συνδεόταν με κάποια εξωτερική εφαρμογή ή ακόμα καλύτερα να δημιουργηθεί μια καινούργια η οποία να επιτρέπει την αγορά και πώληση μετοχών απευθείας μέσα από το stock market. Με τον τρόπο αυτό ο χρήστης δεν θα χρειάζεται να έχει ανοικτά πολλαπλά παράθυρα αλλά όλη η δουλειά του θα γίνεται μέσα από μια και μόνο.

4) *Αλλαγή ηχητικών μηνυμάτων.*

Αυτή τη στιγμή δίνεται η δυνατότητα στο χρήστη να ηχογραφεί μηνύματα και να τα αποστέλλει σε κάποιον άλλο. Θα μπορούσε να αλλάξει και αντί για ηχογραφημένα μηνύματα να υπάρχει απευθείας τηλεφωνήματα μέσω διαδικτύου για πιο άμεση επικοινωνία

Βιβλιογραφία

Βιβλία

- 1) Γιώργος Λιακέας, (2003), *Εισαγωγή στην java2*, Κλειδάριθμος
- 2) Τσαραβάς Χαράλαμπος, (2009), *Μεταπτυχιακή Εργασία: Δημιουργία διαδραστικής e-Υπηρεσίας υποστήριξης συναλλαγών με χρήση XML, JAVA, WEB SERVICES*, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς
- 3) *Εισαγωγή στη γλώσσα προγραμματισμού Java*,
- 4) Ι.Θεοχαρόπουλος, (2010) *Αξιοποίηση της Ενοποιημένης Πλατφόρμας Eclipse για την Ανάπτυξη και Διαχείριση Ανοιχτού Εκπαιδευτικού Λογισμικού σε Java*, Τμήμα Πληροφορικής Πανεπιστημίου Πειραιώς
- 5) Αλευρά Βασιλική, Μπουγιάκας Γιώργος, Πανάρετου Αλεξάνδρα, Καλαιτσιδής Θεοδωρής, Νουβάκη Αγγελική,(2009), *ΕΡΓΑΣΙΑ: Ανάλυση και Σχεδιασμός του Συστήματος Ηλεκτρονικής Συνταγογράφησης*, Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων
- 6) Ρούσκας Αθανάσιος (2009), *Μεταπτυχιακή Εργασία: Υποστήριξη ατόμων με ειδικές ανάγκες (ΑμεΑ) με τη χρήση εκπαιδευτικών εφαρμογών μέσω διαδικτύου*, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς
- 7) Πουρνάρας Ευάγγελος(2006) *,Μεταπτυχιακή Εργασία : Ανάπτυξη τρισδιάστατου διαδικτυακού παιχνιδιού αγώνων αυτοκινήτων πολλαπλών παικτών*, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς
- 8) Δαμιανός Χατζηαντωνίου, *Σημειώσεις: Java και Sql*, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς
- 9) Ρήγας Μάρκος(2010), *Μεταπτυχιακή Εργασία: Ανάπτυξη εφαρμογής κοινωνικής δικτύωσης βασισμένης στην τοποθεσία με τη βοήθεια Web Services*, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς
- 10) Τζανέτος Πομόνης (2010), *Διδακτορική Διατριβή : Προς το Web 3.0: Διαδικασία ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0*, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής, Πανεπιστήμιο Πατρών

- 11) Χρήστος Μπούρας, Θρασύβουλος Τσιάτσος, Σημειώσεις: Τεχνολογίες και Πρότυπα για την Υλοποίηση Συνεργατικών Συστημάτων

Ιστοσελίδες.

- 1) <http://www.jogl.info/tutorial/lektion1.htm#>
- 2) http://en.wikipedia.org/wiki/Java_OpenGL
- 3) <http://www.jogl.info/>
- 4) <http://el.wikipedia.org/wiki/Java>
- 5) http://www.islab.demokritos.gr/gr/html/ptixiakos/kostas-aris_ptyxiakh/Phtml/java.htm
- 6) <http://www.filaderlis.com/ebooks/%C5%E9%F3%E1%E3%F9%E3%DE%20%F3%F4%E7%ED%20SQL.PDF>
- 7) http://www.medialab.ntua.gr/education/ComputerGraphics/OpenGL_Lectures/01-Introduction.pdf
- 8) http://el.wikipedia.org/wiki/Microsoft_SQL_Server
- 9) http://imu.ntua.gr/projects/dern/files/Papailiou%20Niki_dern_final.pdf
- 10) http://www.infosoc.gr/infosoc/el-GR/services/elibrary/reports_list/systimata_apomakrism_dedom/architecture.htm
- 11) http://en.wikipedia.org/wiki/Client%E2%80%93server_model
- 12) http://anamorfosi.teiser.gr/ekp_yliko/e-notes/Data/database/main.htm
- 13) http://www.it.uom.gr/project/client_server/theoria1.htm
- 14) <http://blog.simcrest.com/what-is-3-tier-architecture-and-why-do-you-need-it/>
- 15) <http://www.eclipse.org/org/>
- 16) [http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software))
- 17) <http://ginva.com/2011/01/the-history-of-social-networking-infographic/>
- 18) <http://en.wikipedia.org/wiki/Usenet>
- 19) http://en.wikipedia.org/wiki/AOL_Instant_Messenger
- 20) <http://en.wikipedia.org/wiki/SixDegrees.com>

- 21) http://www.google.gr/imgres?q=social+media+history&um=1&hl=el&biw=1280&bih=685&tbm=isch&tbnid=3_c0G2FQaGLj3M:&imgrefurl=http://www.technoskillonline.com/2011/03/16-popular-social-media-sites-cloned-by-china-infographics/&docid=GYe-z2JUptZiyM&w=1500&h=1500&ei=QCtRTvmKClqRswb4z8S1Aw&zoom=1&iact=rc&dur=348&page=2&tbnh=136&tbnw=136&start=18&ndsp=22&ved=1t:429,r:8,s:18&tx=36&ty=75
- 22) <http://www.java-tips.org/java-se-tips/javafx.sound/capturing-audio-with-java-sound-api.html>
- 23) <http://www.cs.vu.nl/~graphics/material.php>
- 24) <http://javasolution.blogspot.gr/2007/04/record-and-save-audio-using-java.html>
- 25) <http://finance.yahoo.com/actives?e=o>
- 26) http://nuclear.mutantstargoat.com/articles/lxf_ogl/lxf_gl01.pdf
- 27) [http://learn20.wikispaces.com/Κοινωνική+Δικτύωση+\(κειμενο\)](http://learn20.wikispaces.com/Κοινωνική+Δικτύωση+(κειμενο))