ΤΙΤΛΟΣ ΠΤΥΧΙΑΚΗΣ

Κατασκευή πλήρους Client – Server Εφαρμογής σε Java

Όνομα σπουδαστή: Κρασιάς Αλέξης – Ιάκωβος ΑΜ 177

Εισηγητής καθηγητής: Αϊβαλής Κωνσταντίνος

Ενότητες

- **Εισαγωγή στις έννοιες της Άμεσης Αποστολής και Λήψης Μηνυμάτων.**
- **Η ιστορία της ανταλλαγής μηνυμάτων**
- **Περιγραφή προγράμματος.**
- **Εργαλεία που χρησιμοποιήθηκαν**
- **UML Class Diagrams**
- **Εκτέλεση προγράμματος**
- **Class**
- **Βιβλιογραφία.**

- **Εισαγωγή στις έννοιες της Άμεσης Αποστολής και Λήψης Μηνυμάτων.**

Τα συστήματα της άμεσης αποστολής και λήψης μηνυμάτων, τα οποία χρησιμοποιούν τη γλώσσα προγραμματισμού Java, είναι έτοιμα να αποτελέσουν ένα μεγάλο μέρος του δικτύου των καταναλωτών και των επιχειρήσεων καθώς και να διαδραματίσουν ένα ουσιαστικό και πρωταρχικό ρόλο, παρόμοιο με αυτόν της ηλεκτρονικής αλληλογραφίας.

Βέβαια, η επικοινωνία μέσω μηνυμάτων υπήρξε ανέκαθεν ένα δομικό στοιχείο του διαδικτύου. Λόγου χάριν, η ηλεκτρονική αλληλογραφία αποτελεί μία από τις πρώτες και πιο εξαπλωμένες τεχνολογίες τιυ διαδικτύου. Παραμένει χαρακτηριστικά ως μία παντοδύναμη εφαρμογή του διαδικτύου. Ωστόσο, όλοι γνωρίζουμε ότι η επικοινωνία στο διαδίκτυο μπορεί σίγουρα να είναι πιο ενδιαφέρουσα και ισχυρή από την "απλή παλιά ηλεκτρονική αλληλογραφία". Θα έπρεπε να είμαστε σε θέση να την εκμεταλλευτούμε καλύτερα, ως ένα μη ακριβό μέσο για την μεταφορά δεδομένων σε άμεσο χρόνο. Αυτός άλλωστε είναι και ο σκοπός αυτής της πτυχιακής.

Τα περισσότερα από τα παραδείγματα και τα θέματα της ΙΜ (ταυτόχρονη αποστολή και λήψη μηνυμάτων) τα οποία ερευνώνται και αναπτύσσονται εδώ, θεωρούνται υπό το πρίσμα ενός επιχειρησιακού παράγοντα ο οποίος ενδιαφέρεται για τη δημιουργία συστημάτων για μεσαίες και μεγάλες επιχειρήσεις. Βεβαίως, επιχειρώ εδώ να σας προκαταβάλω, καθώς οι εργασίες (μελέτες) μου τείνουν να εμπίπτουν όλες σε αυτήν την κατηγορία. Εντούτοις τα ίδια θέματα, τα οποία ενδιαφέρουν τις εταιρίες, αποτελούν εξίσου σημαντικό πόλο ενδιαφέροντος και για οποιοδήποτε παράγοντα ο οποίος επιθυμεί να δημιουργήσει αξιόπιστα και ασφαλή συστήματα.

Μπαίνοντας στην ουσία της συζήτησης, θα θέσουμε σε εφαρμογή ένα σύστημα ΙΜ (άμεση αποστολή και λήψη μηνυμάτων) γραμμένο στην Java. Σε αυτό το κεφάλαιο θα εξετάσουμε γενικά τα συστήματα ΙΜ (άμεση αποστολή και λήψη μηνυμάτων). Η συζήτηση στο μεγαλύτερό της μέρος θα είναι μη- τεχνική έτσι ώστε να συγκεντρωθούμε στο γιατί και που χρειαζόμαστε την ΙΜ (άμεση ανταλλαγή μηνυμάτων). Αφορά στα πλεονεκτήματα και τα μειονεκτήματα τα οποία συνδέονται με την, βασισμένη πάνω στα πρωτόκολλα, ανταλλαγή μηνυμάτων (ΙΜ).

- **Η ιστορία της ανταλλαγής μηνυμάτων**

Η ιδέα της άμεσης ανταλλαγής μηνυμάτων υπάρχει εδώ και καιρό. Όλα τα ορατά χαρακτηριστικά της ΙΜ (Άμεσης Αποστολής και Λήψης Μηνυμάτων), όπως η συζήτηση μεταξύ ατόμων ένα προς ένα και οι ομάδες συζήτησης, υπήρχαν και σε άλλες εφαρμογές του διαδικτύου πολύ πριν κάνει την εμφάνισή της στο χώρο η Άμεση Ανταλλαγή Μηνυμάτων (ΙΜ). Παραδείγματος χάριν, η κλασσική εφαρμογή ομιλίας στα Unix επέτρεπε στους χρήστες να συνομιλούν στο διαδίκτυο πολλά χρόνια πριν την εμφάνιση της Άμεσης Ανταλλαγής Μηνυμάτων (ΙΜ). Επιπλέον, πραγματοποιούνται ομαδικές συζητήσεις βασισμένες πάνω στην τεχνολογία των συστημάτων IRC, δηλαδή της Αναμετάδοσης Ομιλίας στο Διαδίκτυο, από τη στιγμή που έχει συλληφθεί και υλοποιηθεί η ομιλία ως ιδέα στο διαδίκτυο.

Παλαιότερα συστήματα συνομιλίας, όπως η ομιλία στα Unix, λειτουργούν παρομοίως με το σύστημα τηλεφωνίας καθώς υπάρχει και εδώ η έλλειψη παρουσίας. Κατά την προσπάθεια συνομιλίας στο διαδίκτυο, κάποιος είναι υποχρεωμένος να κάνει τυφλές κλήσεις προς το άτομο εκείνο με το οποίο θέλει να επικοινωνήσει, ελπίζοντας ότι θα είναι διαθέσιμο για να απαντήσει στην κλήση. Σε αντίθεση ωστόσο με το τηλεφωνικό σύστημα, οι περισσότεροι άνθρωποι δεν είναι διαθέσιμοι για να μιλήσουν συνδεόμενοι με το διαδίκτυο τόσο συχνά όσο όταν μιλάνε μέσω τηλεφώνου. Για το λόγο αυτό οι πιθανότητες του να βρεις κάποιον "σπίτι" συνδεδεμένο στο διαδίκτυο μειώνονται σημαντικά.

Η απλότητα και η ενσωμάτωση των συστημάτων της Άμεσης Ανταλλαγής Μηνυμάτων (IM), διαδόθηκε πρώτα στην καταναλωτική αγορά. Το πιο επιτυχές καταναλωτικό σύστημα είναι το AIM (AOL Υπηρεσία Διεκπεραίωσης 'Άμεσης Αποστολής και Λήψης μηνυμάτων), το οποίο και εισήγαγε την τεχνολογία IM (Άμεση Ανταλλαγή Μηνυμάτων) στο κύριο ρεύμα των καταναλωτών και ενθάρρυνε περισσότερα μέλη να την αγκαλιάσουν. Η καταναλωτική αγορά για τις υπηρεσίες άμεσης ανταλλαγής μηνυμάτων συνεχώς αναπτύσσεται και υπάρχουν αρκετές ευκαιρίες για τους επιχειρησιακούς παράγοντες που θέλουν να δραστηριοποιηθούν στο χώρο.

Αν και η καταναλωτική αγορά για την IM (άμεση αποστολή και λήψη μηνυμάτων) είναι αρκετά μεγάλη, το μεγαλύτερο μέρος της αναπτυξιακής κοινότητας ενδιαφέρεται για τις ευκαιρίες εφαρμογής των τεχνολογιών IM στις επιχειρήσεις. Βασικά, οι επιχειρήσεις ευημερούν ή καταστρέφονται εξαιτίας της ικανότητάς τους να επικοινωνούν τόσο εντός της εταιρίας όσο και με τους εταίρους και με τους πελάτες. Η Άμεση Ανταλλαγή Μηνυμάτων (IM) παρέχει νέα κανάλια επικοινωνίας τα οποία σίγουρα θα εξυπηρετήσουν πολλές απαιτήσεις της επικοινωνίας μέσω μηνυμάτων.

Προβολή μίας έκθεσης πεπραγμένων για την χρήση της IM (Άμεση Ανταλλαγή Μηνυμάτων) στις επιχειρήσεις και στις αγορές καταναλωτών, από την εταιρία IDC.

Έτος	***Επιχείρηση IM (Μηνύματα/Ετος)***	***Καταναλωτές IM (Μηνύματα/'Ετος)***
2001	145	262
2003	626	409
2005	1.2B	800

Μία από τις πιο πολλά υποσχόμενες εφαρμογές της IM (Άμεση Ανταλλαγή Μηνυμάτων) στον τομέα των επιχειρήσεων εντοπίζεται στη διαχείρηση των σχέσεων με τους πελάτες (CRM) ή στην εξυπηρέτηση των πελατών. Καταρχήν, η Άμεση Ανταλλαγή Μηνυμάτων παρέχει ακόμη ένα τρόπο για να επικοινωνήσει μία εταιρία με τους πελάτες της. Επίσης, παρέχει τη δυνατότητα ένταξης στην εμπειρία του πελάτη με σκοπό την περαιτέρω υπστήριξή του.

Φανταστείτε ένα πελάτη υπολογιστών του οποίου η εφαρμογή του λειτουργικού συστήματος μόλις κατέρρευσε. Ξεκινάνε τη χρήση της Βοήθειας που συνόδευε την εφαρμογή. Αυτή η χρησιμότητα αποτελεί στην ουσία ένα τροποποιημένο πελάτη της IM (Άμεση Ανταλλαγή Μηνυμάτων). Ο πελάτης συνδέεται με μία ομάδα συνομιλίας αφιερωμένη στους χρήστες της εφαρμογής. Ο κάθε χρήστης μπορεί να αναζητήσει βοήθεια από οποιονδήποτε είναι συνδεδεμένος τη συγκεκριμένη στιγμή. Αν κανένας δεν είναι διαθέσιμος, τότε ο πελάτης επικοινωνεί με ένα αυτόματο κουτί συνομιλίας ρυθμισμένο βάσει της IM(Άμεση Αποστολή και Λήψη Μηνυμάτων). Το κουτί συνομιλίας δύναται να κάνει βασικές ερωτήσεις για το πρόβλημα στον πελάτη και να χρησιμοποιήσει τις πληροφορίες για να καθοδηγήσει το χρήστη στον ειδικό εκείνο που θα του παρέχει την καλύτερη τεχνική υποστήριξη.

Είναι αξιοσημείωτο το πόσο άμεσα συνδέεται ο πελάτης με τις πηγές υποστήριξης από τη στιγμή που ανακύπτει ένα πρόβλημα. Επιπλέον, η άμεση ανταλλαγή μηνυμάτων (IM) μας επιτρέπει να παράσχουμε καθοδηγούμενη βοήθεια ξεκινώντας με ομάδες χρηστών οι οποίοι είναι ελεύθερα συνδεδεμένοι στο διαδίκτυο. Υπάρχουν αυτοματοποιημένα κουτιά ομιλίας τα οποία μπορούν να ελέγξουν την πρόοδο του πελάτη, να κάνουν υποδείξεις και σταδιακά να τον κατευθύνουν σ'εναν υπάλληλο της εταιρίας ο οποίος θα προσφέρει την κατάλληλη βοήθεια. Ο χειρισμός και το φιλτράρισμα των απλών προβλημάτων από την ελεύθερη ομάδα των χρηστών μπορεί να μειώσει πολλά περιστατικά κακής εξυπηρέτησης των πελατών τα οποία ελαχιστοποιούν τα κέρδη και μειώνουν την ικανοποίηση του πελάτη. Η παγκοσμιότητα μάλιστα της άμεσης αποστολής και λήψης μηνυμάτων (IM) επιτρέπει στην κοινότητα των χρηστών να έχουν ένα σφαιρικό εύρος το οποίο είναι ιδιαίτερα χρηστικό όταν οι πελάτες χρειάζονται υποστήριξη εκτός των ωρών εργασίας.

Εκτός από τις προαναφερθείσες ικανότητες επικοινωνίας, η Άμεση Αποστολή και Λήψη Μηνυμάτων (IM) παρέχει σημαντικά οφέλη σε μία εταιρία, όπως εξοικονόμηση του κόστους.

Η ανταλλαγή μηνυμάτων μεταξύ ιδιωτών δεν είναι το μόνο πλεονέκτημα που η χρήση του συστήματος IM μπορεί να παρέχει σε μία εταιρία. Οι επιχειρήσεις θα έπρεπε να επιτρέπουν σε υπολογιστές να επικοινωνούν μεταξύ τους. Αυτό αληθεύει είτε σε περιπτώσεις όπου οι υπολογιστές είναι εσωτερικοί σε μία επιχείρηση όπως όταν οι λογιστικές εφαρμογές ελέγχουν τις βάσεις δεδομένων της εξυπηρέτησης πελατών, είτε όταν η επικοινωνία των υπολογιστών λαμβάνει χώρα μεταξύ εταίρων με συναλλαγές του τύπου "από επιχείρηση προς επιχείρηση".

Η χρήση πλαισίων ανταλλαγής μηνυμάτων για την επικοινωνία των υπολογιστών δεν αποτελεί καινούργια ιδέα. Προϊόντα όπως τα IBM MQSeries, Microsoft MSMQ, TIBCO Rendevous, Open Horizon Ambrosia και το Modulus InterAgent χρησιμοποιούνται εδώ και χρόνια στις επιχειρήσεις. Τα πλεονεκτήματα της ανταλλαγής μηνυμάτων στον επιχειρηματικό τομέα έχουν καλά αποδειχθεί.

Στην πραγματικότητα, η Επιχειρηματική 'Εκδοση της Java 2 (J2EE) συμπεριλαμβάνει τις πρότυπες βιβλιοθήκες της Υπηρεσίας Μηνυμάτων της Java (JMS), ώστε να δοθεί μία

πρότυπη επιφάνεια Java στο σύστημα αλληλογραφίας για τις ανάγκες μηχανοργάνωσης της εταιρίας. Η ισχύς και η ευκαμψία της Java και του ενδιάμεσου λογισμικού (MOM) το οποίο έχει σαφή προσανατολισμό στην επιχειρηματική αλληλογραφία, οδήγησαν πολύ γρήγορα στην υιοθέτηση του JMS. Οι περισσότεροι πωλητές επιχειρηματικών συστημάτων ανταλλαγής μηνυμάτων υποστηρίζουν το JMS.

Παραδείγματος χάριν, φανταστείτε ότι είστε μία τηλεπικοινωνιακή εταιρία η οποία παρέχει τηλεφωνικές υπηρεσίες σε κατοικίες. Θέλετε να δημιουργήσετε ένα σύστημα υπολογιστών το οποίο θα σας βοηθήσει να χειρίζεστε τα προβλήματα εξυπηρέτησης. Ας σκεφτούμε το σενάριο σύμφωνα με το οποίο έχει κοπεί μία τηλεφωνική γραμμή. Ένα σύστημα βασισμένο στην τεχνολογία Άμεσης Αποστολής και Λήψης Μηνυμάτων (ΙΜ), μπορεί να καταχωρίσει το πρόβλημα ως ένα μήνυμα ΙΜ και να το δρομολογήσει σε ένα γραφείο διαχείρισης προβλημάτων υπάλληλος του γραφείου λαμβάνει το μήνυμα και γνωρίζει ότι πρέπει να αποστείλει ένα συνεργείο στην τοποθεσία. Μπορεί να ελέγξει την παρουσία ΙΜ (Άμεσης Αποστολής) των συνεργείων, να βρει ένα διαθέσιμο και να αποστείλει μία ειδοποίηση προβλήματος για να επισκευάσουν τη γραμμή.

- **Περιγραφή προγράμματος.**

Το πρόγραμμα που θα υλοποιηθεί αποτελείται από δύο κομμάτια: το διακομιστή και τον πελάτη τα οποία συνδέονται μέσω δικτύου μεταξύ τους και ανταλλάσουν μηνύματα σε πραγματικό χρόνο. Η υλοποίηση της εργασίας θα γίνει εξολοκλήρου στην γλώσσα προγραμματισμού java όπως και όλες οι βιβλιοθήκες που θα χρησιμοποιηθούν είναι γραμμένες στην java.

Το πρόγραμμα έχει ως κύριο στόχο μικρές ομάδες χρηστών ή επιχειρήσεων που επιθυμούν μια ενδοεπικοινωνία σε κλειστό ή ανοιχτό δίκτυο το οποίο έχει πρόσβαση στο διαδίκτυο ή όχι.

Καταρχήν, υπάρχει ένας διακομιστής και πολλοί πελάτες. Ο server περιμένει τις κλήσεις των συμμετεχόντων στο chat και είναι υπεύθυνος για τη μεταφορά των μηνυμάτων μεταξύ τους. Λαμβάνει τα ιδιωτικά μηνύματα και τα προωθεί ανάλογα. Τα ιδιωτικά μηνύματα απευθύνονται από έναν χρήστη σε έναν άλλο μόνο. Στο σύστημα μας ο κάθε client έχει σύνδεση μόνο με τον server και με κανέναν από τους άλλους clients, συνεπώς ο server παίρνει όλα τα μηνύματα και πρέπει να τα προωθήσει μόνο στους κατάλληλους παραλήπτες.

Η λίστα των συμμετεχόντων καθώς αυτοί μπαίνουν και βγαίνουν από τη συζήτηση αλλάζει και υπεύθυνος για την παρακολούθηση της και την ενημέρωση των συμμετεχόντων είναι φυσικά ο διακομιστής . Οι συμμετέχοντες μπορούν να ξεκινήσουν μια ιδιωτική συζήτηση επιλέγοντας κάποιο άλλο συμμετέχοντα από τη λίστα.

Το πρόγραμμα, όπως προαναφέρθηκε παραπάνω, αποτελείται από δύο μέρη τα οποία αφορούν στο διακομιστή και τον πελάτη.Τα χαρακτηριστικά του καθενός αναγράφονται παρακάτω.

Τα χαρακτηριστικά του πελάτη είναι τα ακόλουθα:

1. Ευκολία χρήσης του προγράμματος.
2. Ανταλλαγή μηνυμάτων μεταξύ τον χρηστών.
3. Εμφάνιση κατάστασης χρήστη.
4. Αλλαγή εμφάνισης προγράμματος.
5. Αλλαγή γλώσσας απεικόνισης.
6. Μοναδιαίο όνομα χρήστη και κωδικός πρόσβασης.
7. Αποθήκευση ρυθμίσεων χρήστη.

Τα χαρακτηριστικά του διακομιστή είναι τα ακόλουθα:

1. Δημιουργία χρηστών.
2. Κατάργηση χρηστών.
3. Αποθήκευση των χρηστών σε βάση δεδομένων.
4. Δρομολόγηση και έλεγχος των μηνυμάτων των πελατών.

- **Εργαλεία που χρησιμοποιήθηκαν**

Προγραμματισμός με NetBeans 5.0

Το NetBeans IDE είναι ένα περιβαλλοντικό ανάπτυγμα IDE - ένα εργαλείο στη διάθεση των προγραμματιστών για να γράψουν, να κάνουν compile, debug και να αναπτύξουν προγράμματα. Είναι γραμμένο σε Java - αλλά μπορεί να υποστηρίξει όλες τις γλώσσες προγραμματισμού. Υπάρχει επίσης ένας μεγάλος αριθμός υπομονάδων (modules) που βοηθάνε στην επέκταση της λειτουργικότητας του NetBeans IDE.

Παράλληλα, η έκδοση 5 έχει σημαντικές καινοτομίες σε σχέση με την προηγούμενη και η πιο σημαντική βρίσκεται στο γραφικό περιβάλλον το οποίο είναι πλέον πολύ εύκολο στην χρήση. Ταυτόχρονα, βοηθάει με την αυτόματη δημιουργία κώδικα έτσι ώστε γλιτώνεις πολύ χρόνο στην υλοποίηση της εφαρμογής. Επίσης, μέσω του γραφικού περιβάλλοντος μπορείς και εντοπίζεις τα λάθη και τις ανωμαλίες στον κώδικα που γράφεις πολύ γρήγορα. Στην εφαρμογή έχει χρησιμοποιηθεί και η δοκιμαστική έκδοση 5.5 από όπου έχουν δημιουργηθεί και τα διαγράμματα UML.

Προγραμματισμός δικτύου σε Java

Οι κλάσεις για προγραμματισμό δικτύου σε Java παρέχονται από το πακέτο java.net. Το πακέτο αυτό δίνει την δυνατότητα στον προγραμματιστή να εγκαταστήσει επικοινωνία μεταξύ δυο υπολογιστών χρησιμοποιώντας sockets. Οι δύο κλάσεις που χρησιμοποιούνται για τον προγραμματισμό με TCP/IP sockets είναι η Socket και η Server Socket.

Η κλάση Socket έχει έναν αριθμό κατασκευαστών που επιτρέπουν στον προγραμματιστή να δημιουργήσει ένα socket και να συνδεθεί με έναν απομακρυσμένο υπολογιστή. Ο πιο απλός κατασκευαστής παίρνει δυο παραμέτρους. Η πρώτη είναι είτε η IP διεύθυνση είτε το συμβολικό όνομα του υπολογιστή με τον οποίο πρόκειται να εφαρμοστεί η επικοινωνία. Η δεύτερη είναι ο αριθμός της θύρας στην οποία ακούει η εφαρμογή με την οποία θέλουμε να επικοινωνήσουμε στον απομακρυσμένο υπολογιστή

Η κλάση Socket, όπως την περιγράψαμε εντοπίζεται στην πλευρά των πελατών (όπως θα δούμε, με διαφορετική αρχικοποίηση μπορεί να εμφανίζεται και στην πλευρά του διακομιστή). Αντίθετα, η κλάση ServerSocket εντοπίζεται αποκλειστικά στην πλευρά του διακομιστή.

Server Database

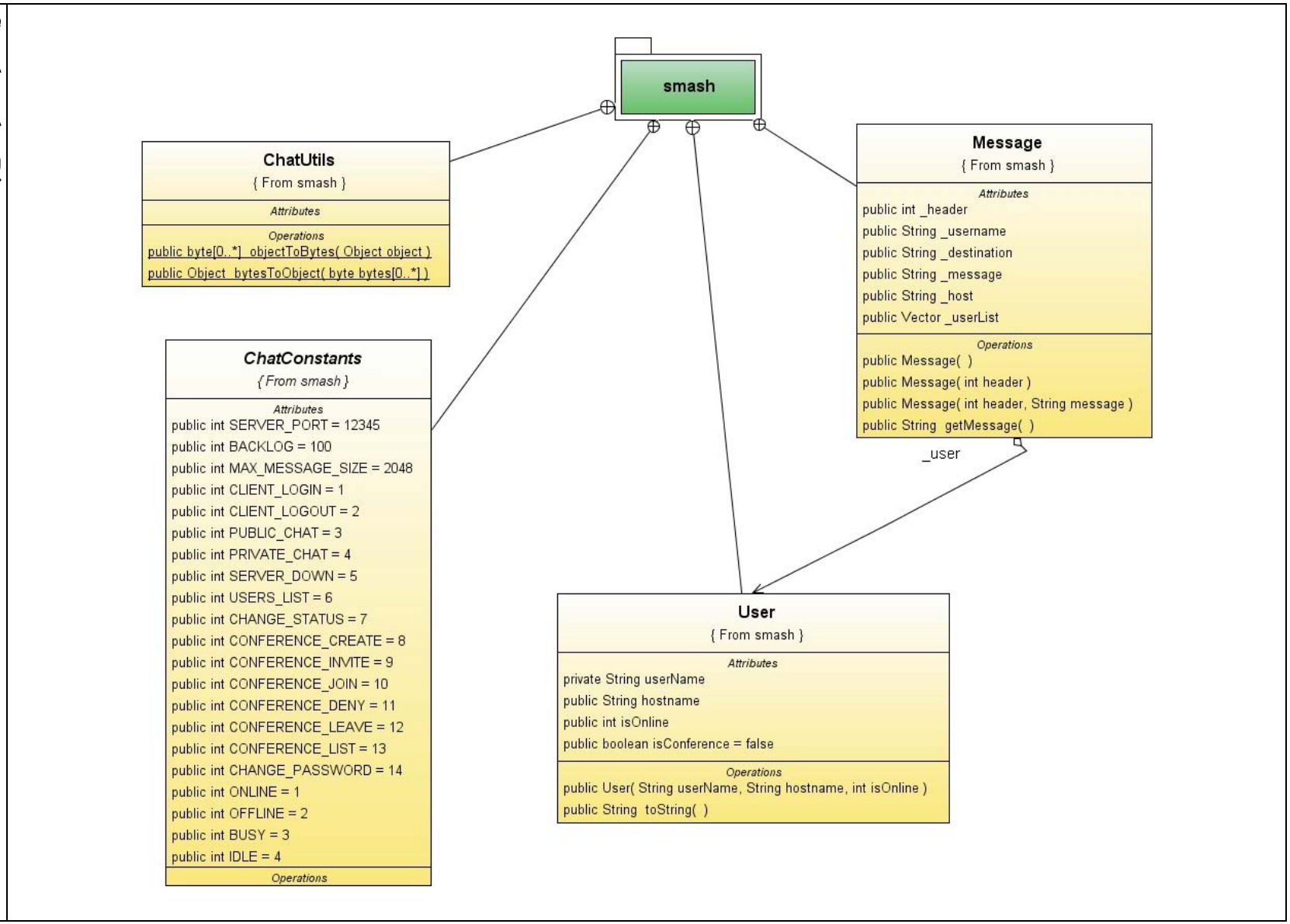
Η βάση δεδομένων που επιλέχτηκε για την εφαρμογή είναι η Java DB ή αλλιώς derby, η οποία είναι 100% γραμμένη στην java. Παράλληλα, είναι μια βάση δεδομένων η οποία είναι ανοιχτού κώδικα καθώς και πλήρως υποστηριζόμενη από την Sun Microsystems. Σημαντικό κριτήριο έπαιξε επίσης και η ευκολία χρήσης της όπως και η ασφάλεια και το μικρό μέγεθος της.

- **UML Class Diagrams**

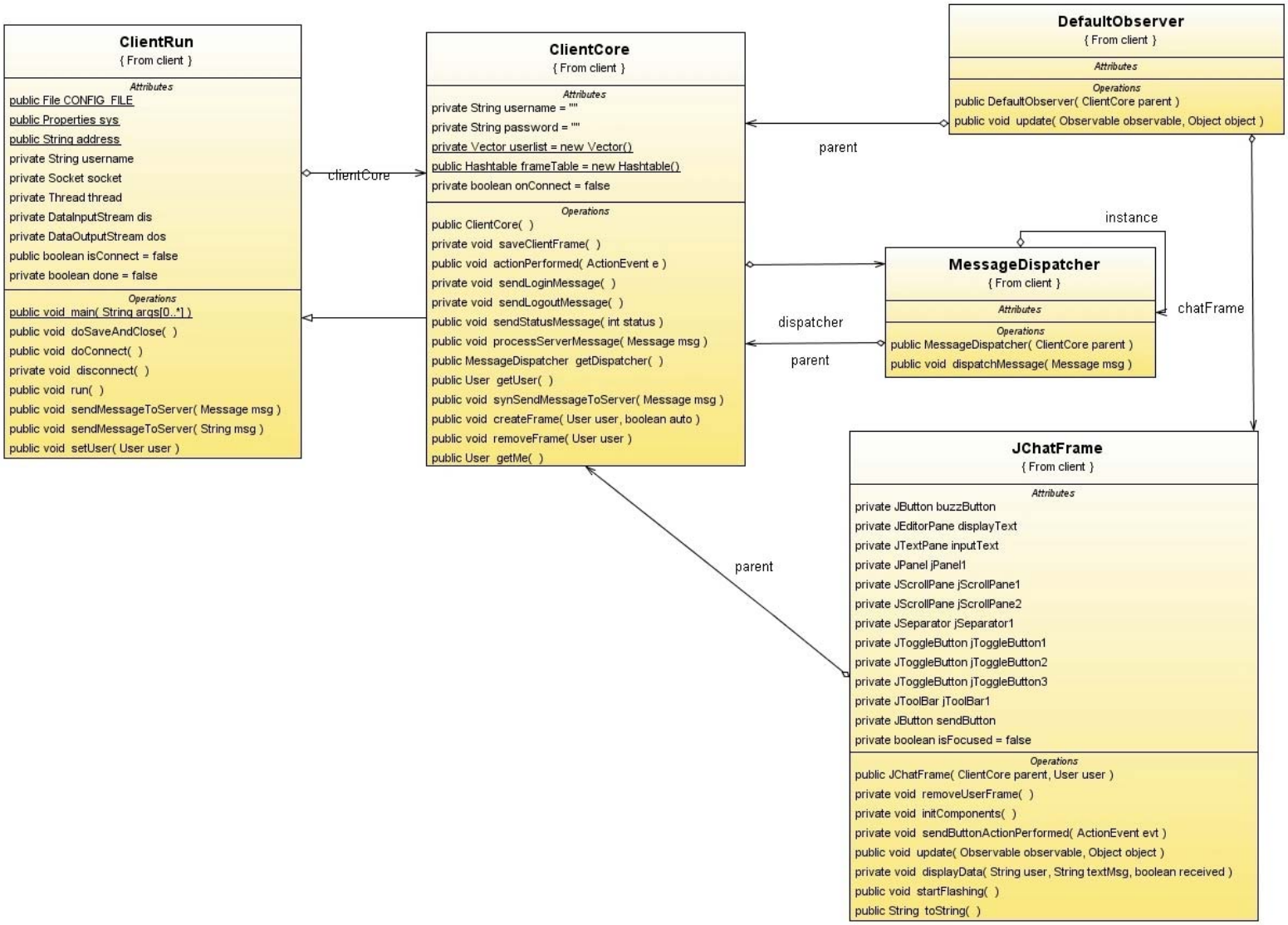
Τα παρακάτω διαγράμματα παριστάνουν τα βασικά αντικείμενα και την συσχέτιση μεταξύ τους. Παράλληλα, απεικονίζουν τις ιδιότητες και τις λειτουργίες της κάθε κλάσης και μας δείχνουν και τον τρόπο με τον οποίο ενώνονται τα αντικείμενα.

Τα τρία σχήματα που ακολουθούν περιγράφουν τον διακομιστή, το πακέτο smash και τον πελάτη και απεικονίζουν της κύριες κλάσεις που χρησιμοποιούν.

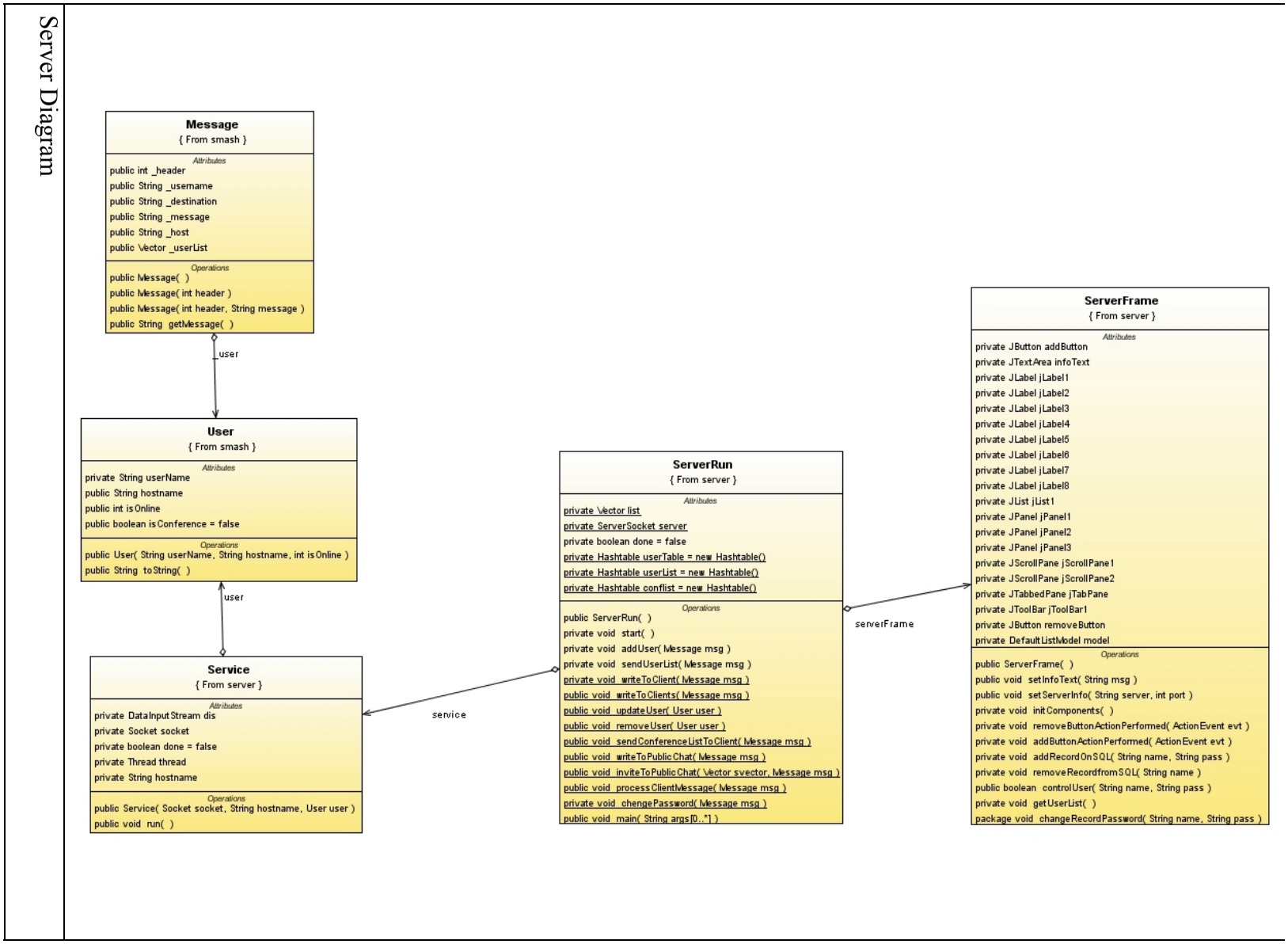
Smash packet Diagram



Client Diagram

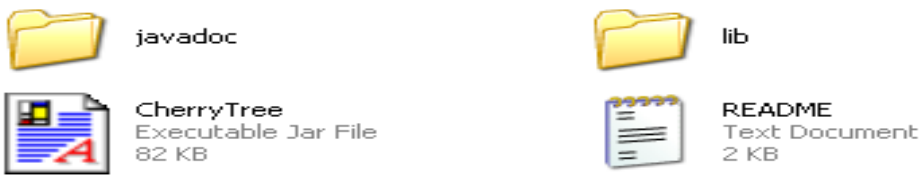


Server Diagram

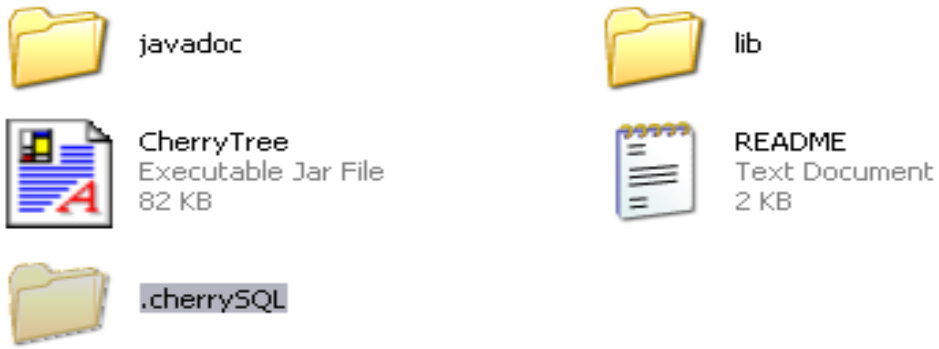


Εκτέλεση προγράμματος

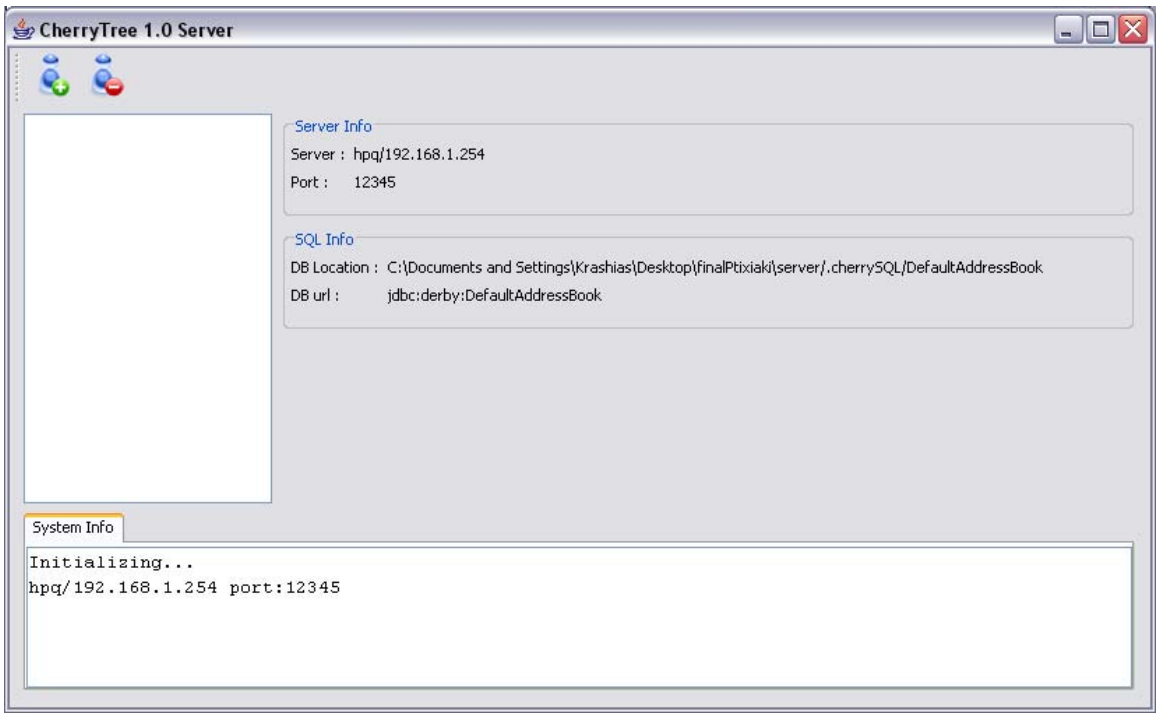
Για να τρέξει η εφαρμογή, ανοίγουμε πρώτα το αρχείο CherryTree.Jar το οποίο και ενεργοποιούμε πατώντας το για να ανοίξει η εφαρμογή του Server IM.



Αν είναι η πρώτη φορά που εκτελείται η εφαρμογή, παρατηρούμε ότι θα δημιουργήσει την βάση δεδομένων με το όνομα cherrySQL μέσα στον φάκελο, στον οποίο βρίσκεται η εφαρμογή.

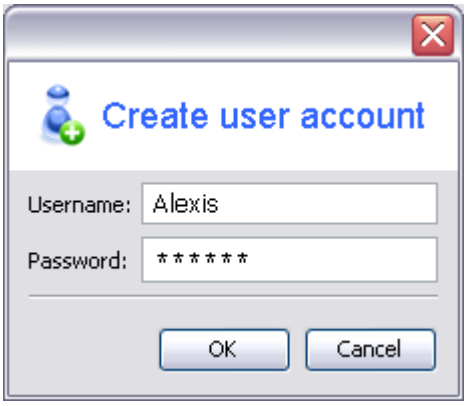


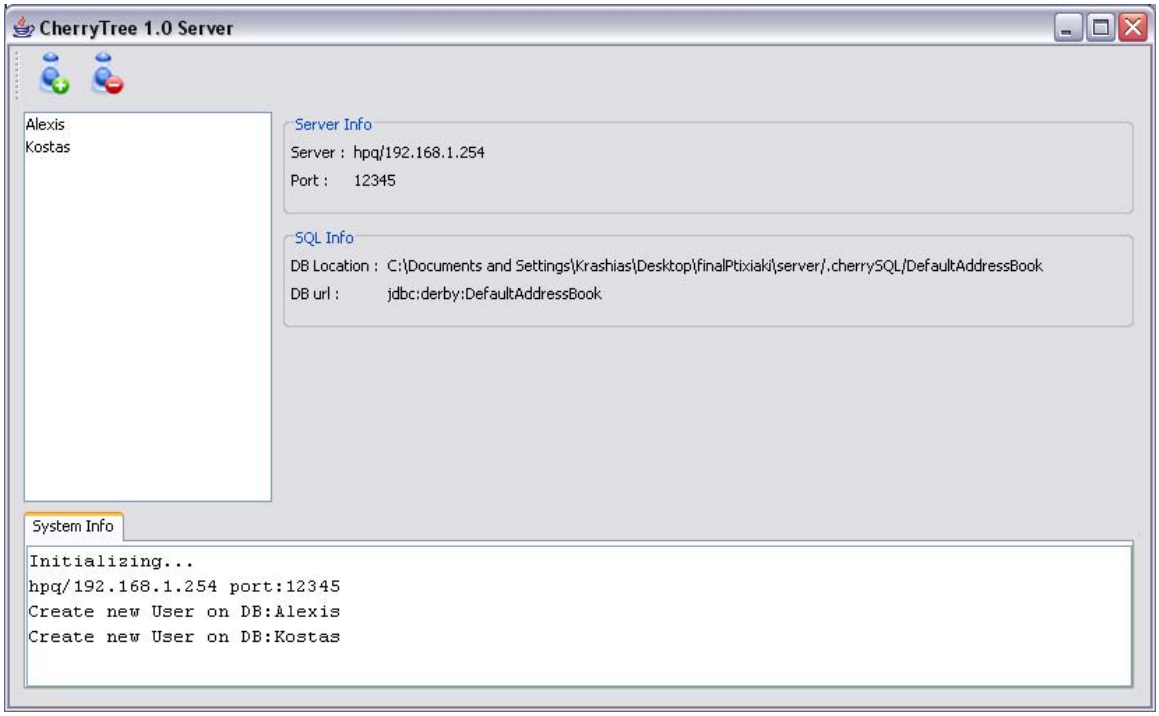
Παράλληλα, θα ανοίξει και ο κεντρικός πίνακας ελέγχου της εφαρμογής του Server.



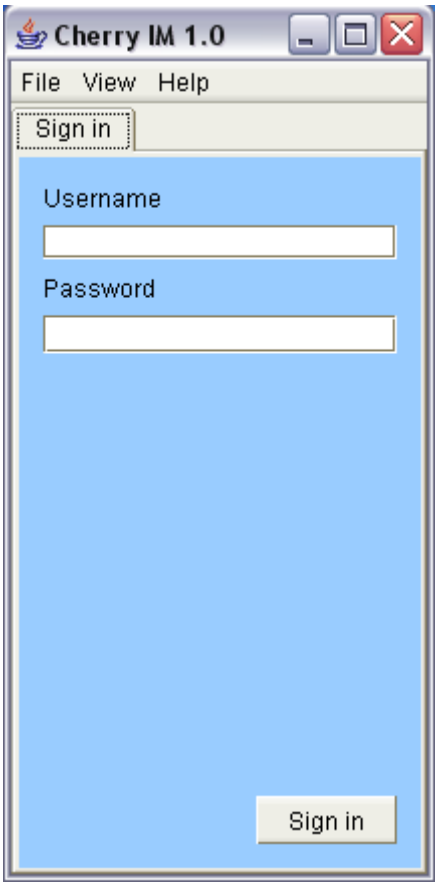
Το παράθυρο που έχει ανοίξει, αναφέρει γενικές πληροφορίες για την εφαρμογή, όπως την διεύθυνση στην οποία έχει ανέβει ο server και την πόρτα την οποία χρησιμοποιεί η εφαρμογή. Παράλληλα, αναφέρει πληροφορίες για την βάση δεδομένων όπως το αρχείο στο οποίο βρίσκεται, και το Url της. Επίσης, στο κάτω μέρος της εφαρμογής εγκαθιστούμε γενικές πληροφορίες της εφαρμογής και στην αριστερή πλευρά βρίσκεται η μπάρα με τους χρήστες. Αν εκτελείται για πρώτη φορά, δεν έχει χρήστες και θα πρέπει να δημιουργήσουμε χρήστες οι οποίοι θα έχουν πρόσβαση στην εφαρμογή, δηλαδή θα μπορούν να συνδεθούν με τον server.

Για να προσθέσουμε ένα χρήστη, πρέπει να πατήσουμε πάνω στο αριστερό εικονίδιο το οποίο θα εμφανίσει το παράθυρο το οποίο φαίνεται παρακάτω και το οποίο αποτελείται από δύο πεδία: το όνομα του χρήστη και τον κωδικό πρόσβασης του χρήστη. Συμπληρώνοντας τα πεδία και πατώντας ok, δημιουργούμε τον χρήστη στην βάση δεδομένων.

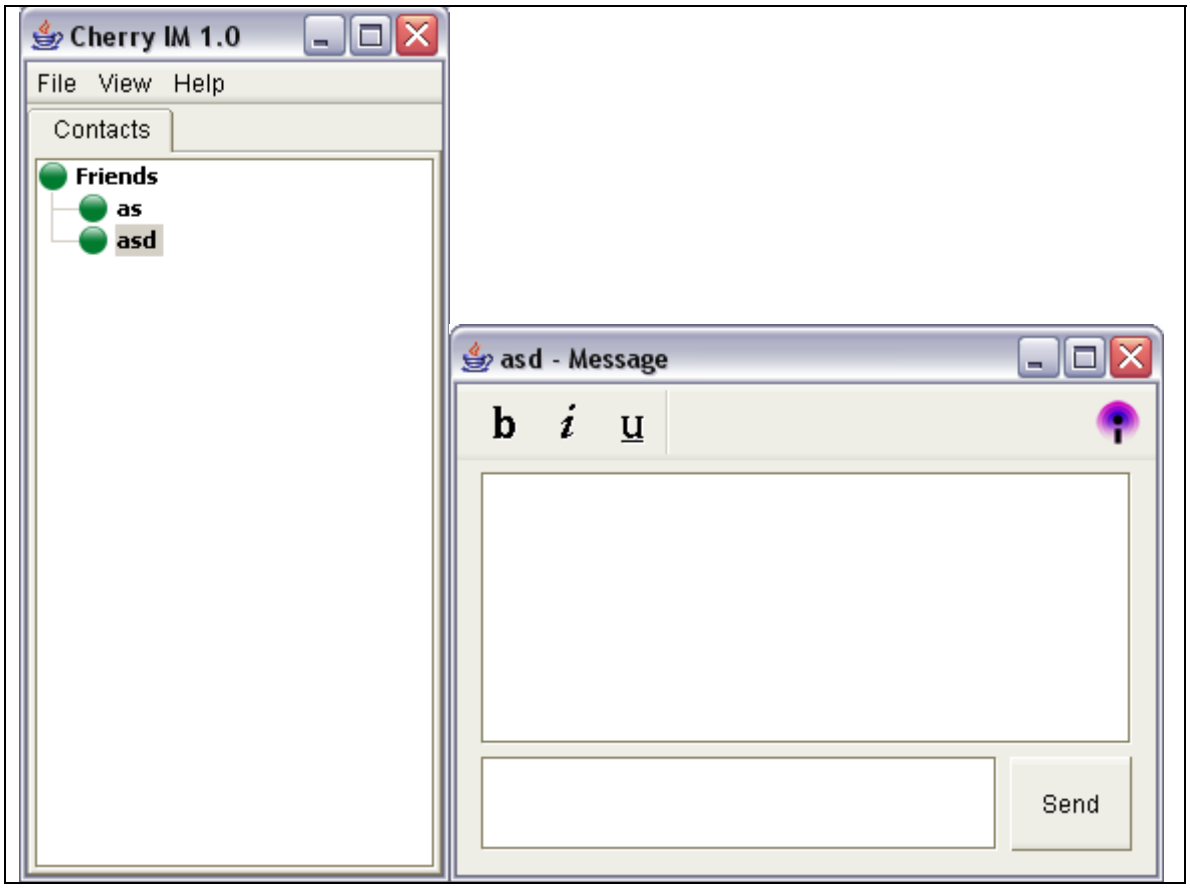




Επαναλαμβάνουμε την παρακάτω ενέργεια για όσους χρήστες θέλουμε να προσθέσουμε, ώστε να έχουν πρόσβαση στην εφαρμογή. Όταν τελειώσουμε την εισαγωγή χρηστών, θα ανοίξουμε τον client. Αυτό γίνεται πατώντας το αρχείο Cherry.Jar το οποίο και θα εμφανίσει ένα splash screen και παράλληλα θα ανοίξει και την εφαρμογή όπως διαπιστώνεται παρακάτω.



Η διαδικασία εδώ είναι απλή, καθώς ο κάθε χρήστης εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης που έχει. Αν τα στοιχειά είναι σωστά τότε ο χρηστής μπορεί και μπαίνει στην εφαρμογή. Αν δεν προχώρηση στο επόμενο παράθυρο θα ζητηθεί από την εφαρμογή να εισάγει ο χρηστής την διεύθυνση η το όνομα του διακομιστεί που είναι να συνδεθεί.



Class Message

```
java.lang.Object
  └ smash.Message
```

All Implemented Interfaces:

java.io.Serializable

```
public class Message
extends java.lang.Object
implements java.io.Serializable
```

See Also:

Serialized Form

Field Summary	
java.lang.String	_destination

int	_header
java.lang.String	_host
java.lang.String	_message
User	_user
java.util.Vector	_userList
java.lang.String	_username

Constructor Summary
Message() Creates a new instance of Message
Message(int header)
Message(int header, java.lang.String message)

Method Summary	
java.lang.String	getMessage()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

_header

public int **_header**

_username

public java.lang.String **_username**

_destination

```
public java.lang.String _destination
```

_message

```
public java.lang.String _message
```

_host

```
public java.lang.String _host
```

_user

```
public User _user
```

_userList

```
public java.util.Vector _userList
```

Constructor Detail

Message

```
public Message()
```

Creates a new instance of Message

Message

```
public Message(int header)
```

Message

```
public Message(int header,
               java.lang.String message)
```

Method Detail

getMessage

```
public java.lang.String getMessage()
```

Class User

```
java.lang.Object
  └ smash.User
```

All Implemented Interfaces:

java.io.Serializable

```
public class User
extends java.lang.Object
implements java.io.Serializable
```

See Also:

Serialized Form

Field Summary	
java.lang.String	hostname
boolean	isConference
int	isOnline

Constructor Summary
User(java.lang.String userName, java.lang.String hostname, int isOnline) Creates a new instance of User

Method Summary	
java.lang.String	toString()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

hostname

```
public java.lang.String hostname
```

isOnline

```
public int isOnline
```

isConference

```
public boolean isConference
```

Constructor Detail

User

```
public User(java.lang.String userName,
            java.lang.String hostname,
            int isOnline)
```

Creates a new instance of User

Method Detail

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

Class ChatUtils

```
java.lang.Object
  └ smash.ChatUtils
```

```
public class ChatUtils
extends java.lang.Object
```

Constructor Summary	
ChatUtils()	

Method Summary	
static java.lang.Object	bytesToObject(byte[] bytes)

static byte[]	objectToBytes(java.lang.Object object)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ChatUtils

```
public ChatUtils()
```

Method Detail

objectToBytes

```
public static byte[] objectToBytes(java.lang.Object object)
                            throws java.io.IOException
```

Throws:
`java.io.IOException`

---

bytesToObject

```
public static java.lang.Object bytesToObject(byte[] bytes)
                                       throws
java.io.IOException,

java.lang.ClassNotFoundException
```

Throws:
`java.io.IOException`
`java.lang.ClassNotFoundException`

ΒΙΒΛΙΟΓΡΑΦΙΑ

- UML distilled 3rd Edition, (ISBN 0-321-19368-7)
- On to Java 2nd Edition, (ISBN 0-201-38598-8)
- Java how to program 5th Edition (ISBN 0-13-120236-7)
- Core Java 2 Volume II Advanced Features (ISBN 0-13-111826-9)