Process MeNtOR 3.o

Uni-SEP

<Talkie>

Έγγραφο Σχεδίασης

Ανδρουλιδάκης Ιωάννης (Α.Μ.: 03111030)
Βασιλάκης Γεώργιος (Α.Μ.: 03111061)
Θεοδωράκης Γιώργος Ραφαήλ (Α.Μ.: 03111071)

Version:	1.0
Print Date:	3/30/2016 14:11:00 PM
Release Date:	
Release State:	Initial/Core/Final
Approval State:	Draft/Approved
Approved by:	
Prepared by:	
Reviewed by:	
Path Name:	
File Name:	Talkie-SDD.doc
Document No:	

object oriented pty. ltd.

Document Change Control

Version	Date	Authors	Summary of Changes

Document Sign-Off

Name (Position)	Signature	Date

© Copyright 1998 Object Oriented Pty Ltd

Contents

© Copyright 1998 Object Oriented Pty Ltd

1 Introduction

1.1 Overview

Σκοπός αυτού του εγγράφου είναι να περιγράψει τις προδιαγραφές σχεδίασης και υλοποίησης του πρωτοκόλλου SIP, μαζί με τις επιπλέον λειτουργίες (επεκτάσεις) που ορίζει η εκφώνηση της άσκησης. Συγκεκριμένα, κατά τη σύνταξη του εγγράφου σχεδίασης λάβαμε υπ'όψη τις απαραίτητες τεχνολογίες, προγραμματιστικές τεχνικές και γλώσσες προγραμματισμού. Στο παρόν έγγραφο περιλαμβάνονται :

- Η ανάλυση της στατικής δομής του συστήματος σε υποσυστήματα, υπο-μονάδες (ψηφίδες) και κλάσεις.
- Η αρχιτεκτονική σχεδίαση που είναι η ανάλυση του συστήματος σε υποσυστήματα και υπομονάδες (αρχιτεκτονική σχεδίαση).
- Η λεπτομερής σχεδίαση , δηλαδή η ανάλυση υπομονάδων σε κλάσεις.
- Η σύνδεση συγκεκριμένων υποσυστημάτων, ψηφίδων και κλάσεων με συγκεκριμένες λειτουργικές απαιτήσεις.

Στη συνέχεια, θα παρουσιαστούν οι κυριότερες σχεδιαστικές μας αποφάσεις, τα Component Diagrams, τα State Diagrams για το Call, τα Deployment Diagrams, καθώς και τα ανλυτικά Class Diagrams.

1.2 Resources - References

- RFC 3261 – SIP: Session Initiation Protocol http://tools.ietf.org/html/rfc326
- jain-SIP-tutorial– Phelim O'Doherty, Mudumbai Ranganathan
- SIP: Protocol Overview– Radvision ® Ltd.

© Copyright 1998 Object Oriented Pty Ltd

2 Major Design Decisions

Κατά τη σχεδίαση του συστήματός μας, λάβαμε αρκετές σημαντικές αποφάσεις σχετικά με την υλοποίησή του και τον τρόπο εσωτερικής επικοινωνίας των υποσυστημάτων του :

1) Αρχικά έπρεπε να διαλέξουμε τον τρόπο με τον οποίο θα αποθηκεύσουμε τις πληροφορίες του συστήματος (που αφορούν τους χρήστες, τις υπηρεσίες κλήσεις, το blocking, το forwarding και το billing) ανάμεσα σε XML/JSON αρχεία και μια DBMS. Προτιμήσαμε να χρησιμοποιήσουμε την DBMS, λαμβάνοντας υπ'όψην τη δυνατότητά της να κλιμακώνει, να χειρίζεται αποδοτικά μεγάλες ποσότητες δεδομένων, την ανοχή σε λάθη (fault tolerance), την εύκολη διαχείρισή της (administration) και τις ACID (Atomicity, Consistency, Isolation, Durability) ιδιότητές της. Έτσι, ουσιαστικά εκμεταλλευτήκαμε έτοιμες built-in λειτουργίες της βάσης, ώστε να επικεντρωθούμε στο ανώτερο επίπεδο υλοποίησης της εφαρμογής. Συγκεκριμένα, επιλέξαμε μια MySQL βάση, καθώς διανέμεται δωρεάν, χρησιμοποιείται ευρέως, είναι open-source και εύκολη στη χρήση.

2) Στη συνέχεια έπρεπε να αποφασίσουμε σχετικά με τον τρόπο επικοινωνίας ανάμεσα στους servers και τους clients, έτσι ώστε να ανταλλαχθούν οι απαραίτητες πληροφορίες. Αυτό μπορεί να επιτευχθεί είτε με τη χρήση του SIP Protocol, είτε με κάποιο άλλο πρωτόκολλο/package. Επιλέξαμε να χρησιμοποιήσουμε το Socket package επειδή παρέχει ασφάλεια και ακεραιότητα δεδομένων, αλλά και ready-to-use υλοποίηση της επικοινωνίας.

3) **Registration (Εγγραφή στο Σύστημα) :**

   Αρχικά στο χρήστη εμφανίζεται ένα παραθύρο διαλόγου (GUI) στο οποίο ο χρήστης μπορεί επιλέξει "Register" ή «Login", ανάλογα αν έχει γραφτεί ήδη στο σύστημα ή όχι. Στην περίπτωση του Register, εμφανίζεται κατάλληλη φόρμα στην οποία συμπληρώνει τα στοιχεία του. Το σύστημά μας πρέπει να κάνει τον απαραίτητο έλεγχο ώστε το username κάθε χρήστη να είναι μοναδικό, και να ενημερώνει κατάλληλα τη βάση. Στην περίπτωση του Login, ο χρήστης μεταφέρεται στο αντίστοιχο παράθυρο στο οποίο συμπληρώνει το username και το password του. Όλα τα δεδομένα χρηστών, συμπεριλαμβανομένων blocking lists, forwarding info και τιμολογιακή πολιτική κάθε πελάτη, αποθηκεύονται στη βάση δεδομένων του συστήματός μας.

4) **Forwarding (Προώθηση Κλήσεων) :**

   Οι πληροφορίες σχετικά με την προώθηση κλήσεων αποθηκεύονται στον αντίστοιχο server, για ευκολότερη και γρηγορότερη πρόσβαση. Δεχόμαστε ότι κάθε χρήστης μπορεί να προωθεί τις εισερχόμενες κλήσεις του μόνο σε ένα χρήστη. Ο Forwarding Server είναι υπεύθυνος για τον εντοπισμό κύκλων που προκύπτουν δυναμικά από την προώθηση κλήσεων μεταξύ των χρηστών. Προκειμένου να βελτιστοποιηθεί η λειτουργία αυτή αποφασίσαμε να αποθηκεύουμε τον τελικό παραλήπτη της κλήσης στη βάση δεδομένων μας, αντί για κάθε ενδιάμεσο χρήστη (hop) της αλυσίδας προώθησης, ώστε η πρόσβασή μας σε αυτόν να είναι αποδοτικότερη. Ο έλεγχος αυτός γίνεται κάθε φορά που ένας χρήστης επιλέγει να κάνει forward τις εισερχόμενες κλήσεις του. Εάν δημιουργηθεί κύκλος, η κλήση τερματίζει με μήνυμα σφάλματος. Ακόμη, εάν κάποιος ενδιάμεσος χρήστης εντός του forwarding chain έχει μπλοκάρει τον αρχικό caller χρήστη, τότε η κλήση δεν προωθείται και η διαδικασία διακόπτεται με κατάλληλο μήνυμα . Τέλος, όταν η

© Copyright 1998 Object Oriented Pty Ltd

προώθηση της κλησης γίνει επιτυχώς, ειδοποιούμε τον caller ότι γίνεται forwarding της κλήσης του με μήνυμα «181 Call is Being Forwarded».

5) **Blocking (Αποκλεισμός Χρήστη)**

Ο χρήστης μέσω της επέκτασης του αρχικού GUI που παρέχει η εφαρμογή, θα έχει την επιλογή να μπλοκάρει έναν ήδη εγγεγραμμένο χρήστη του συστήματος. Αυτό πραγματοποιείται με τη διαμεσολάβηση του proxy και του blocking server, οι οποίοι έχουν πρόσβαση και ενημέρωνουν κατάλληλα τη βάση σχετικά με το ποιος χρήστης μπλόκαρε ποιόν. Όταν ο caller είναι blocked από τον callee και τον καλέσει, του επιστρέφεται μήνυμα “480 Temporary Unavailabe”. Ο έλεγχος για blocking έχει μεγαλύτερη προτεραιότητα από τον έλεγχο για forwarding. Δηλαδή μόνο εάν ένας χρήστης στην αλυσίδα του forwarding δεν μπλοκάρει τον αρχικό καλούντα, ελέγχουμε εάν ο ίδιος προωθεί τις κλήσεις του αλλού.

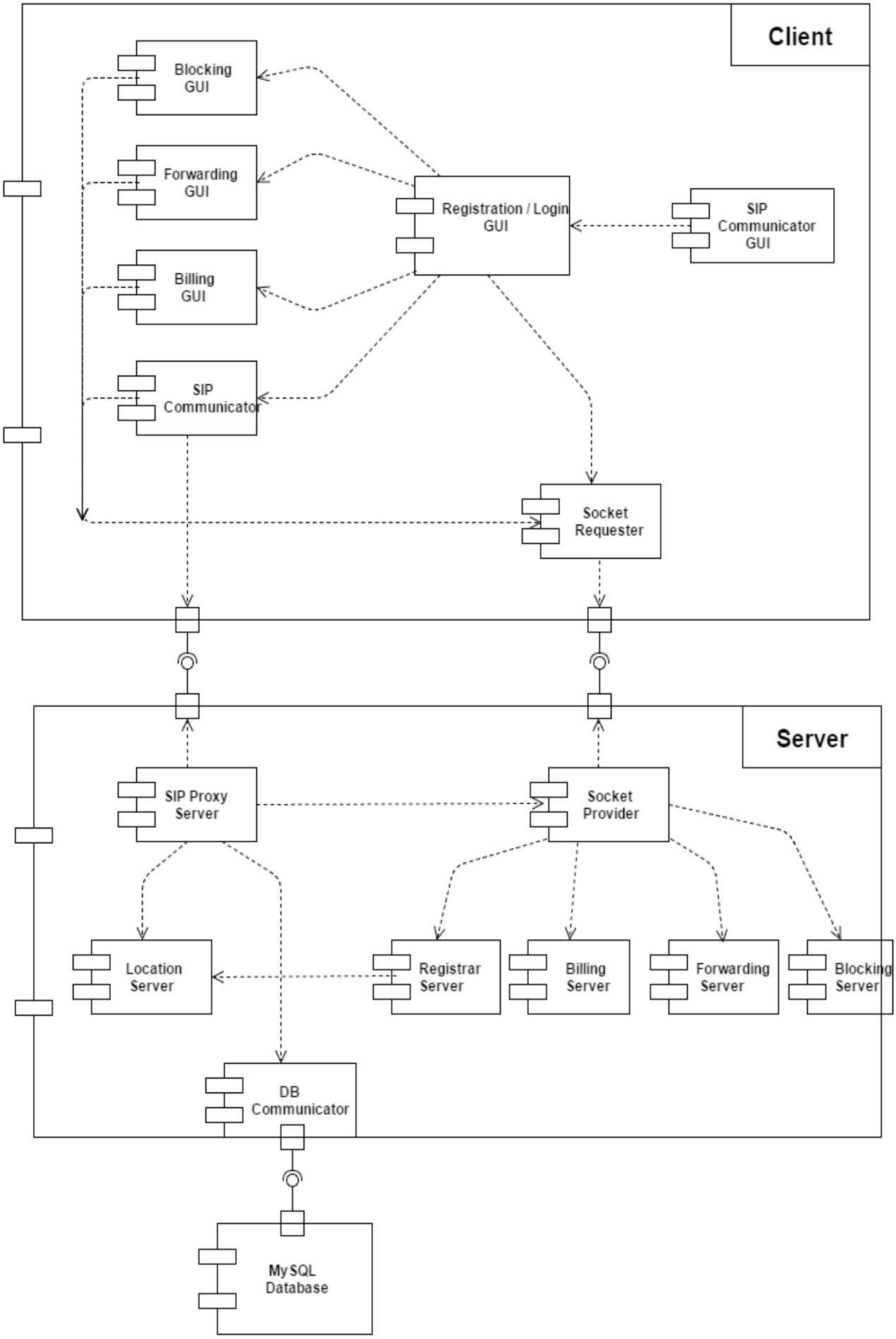
6) **Billing (Τιμολόγηση)**

Οι πληροφορίες που σχετίζονται με τη χρέωση κάθε κλήσης αποθηκεύονται στη βάση δεδομένων. Θεωρούμε ότι ο χρόνος για την κλήση ξεκινά να μετράει μόλις ο Proxy Server λάβει το σήμα «ACK» και σταματά όταν λάβει το σήμα «BYE». Στη συνέχεια ο Proxy ενημερώνει τον Billing Server για τη διάρκεια της κλήσης, ώστε αυτός να υπολογίσει το κόστος της και να το αποθηκεύσει στη βάση. Η χρέωση της κλήσης γίνεται ανά δευτερόλεπτο και λαμβάνεται υπ’όψη η πολιτική χρεώσης που έχει επιλέξει ο χρήστης από τις 2 διαθέσιμες : standard user και premium user.

3 Architecture

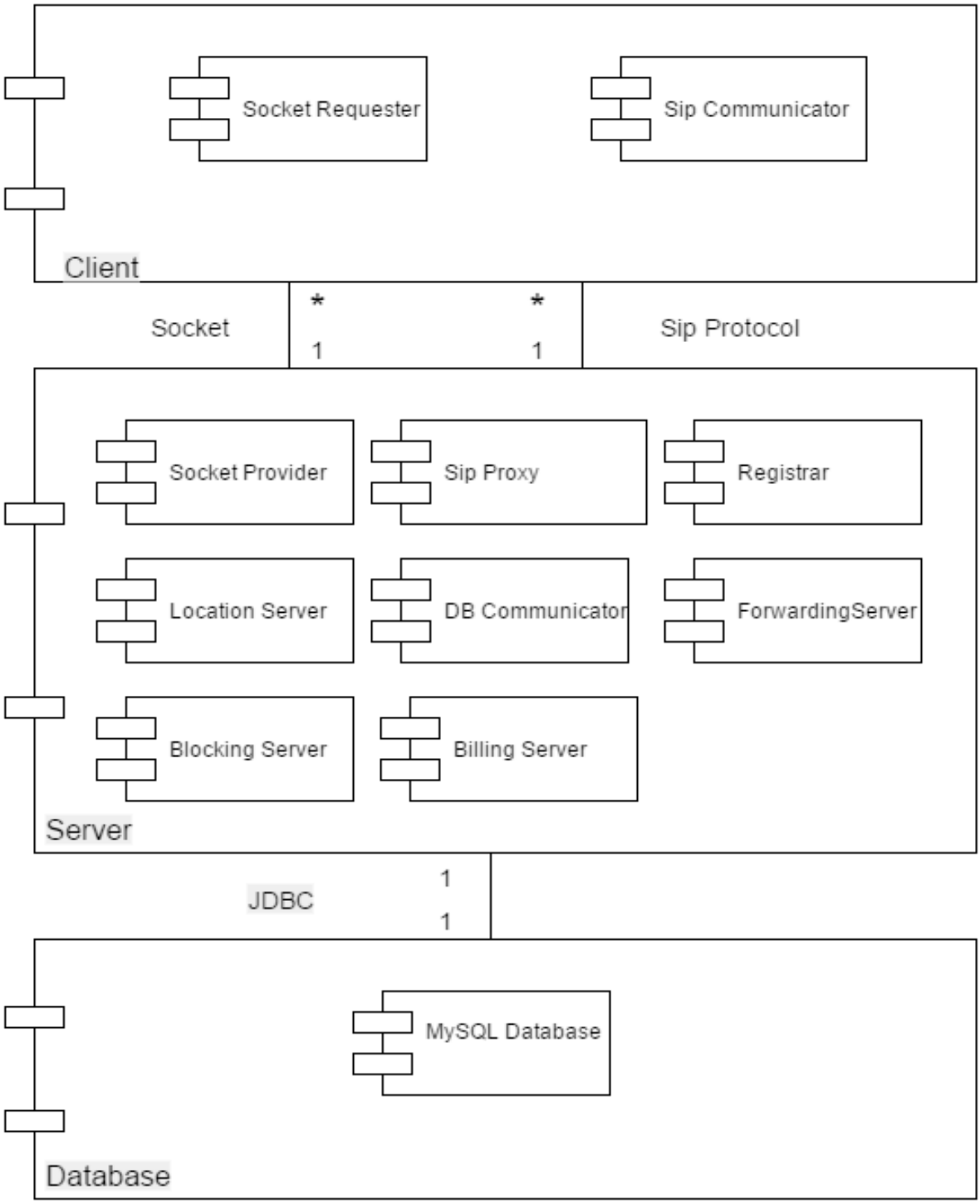
3.1 *Component Diagram*

© Copyright 1998 Object Oriented Pty Ltd



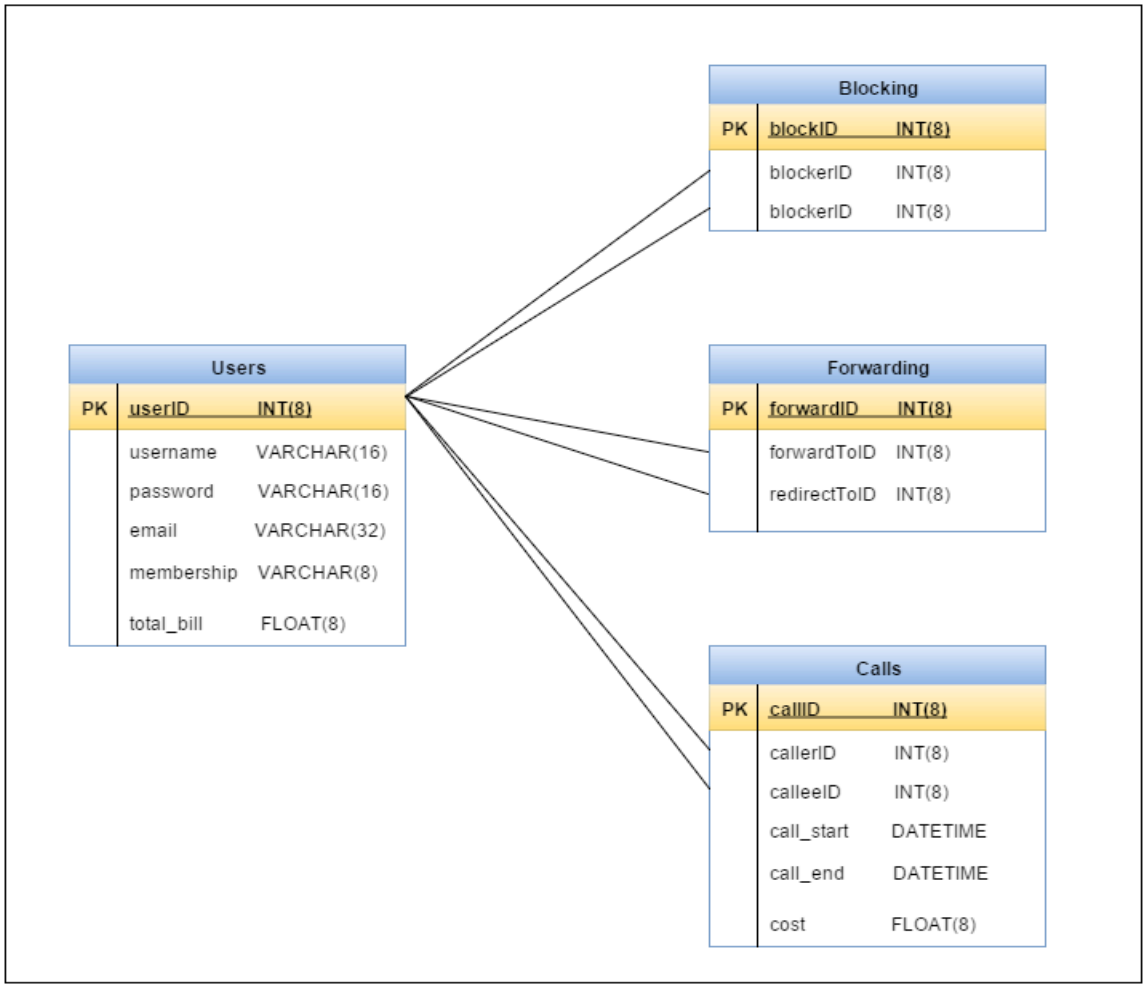
© Copyright 1998 Object Oriented Pty Ltd

3.2 *Deployment Diagram*



© Copyright 1998 Object Oriented Pty Ltd

3.3 *Database Schema*



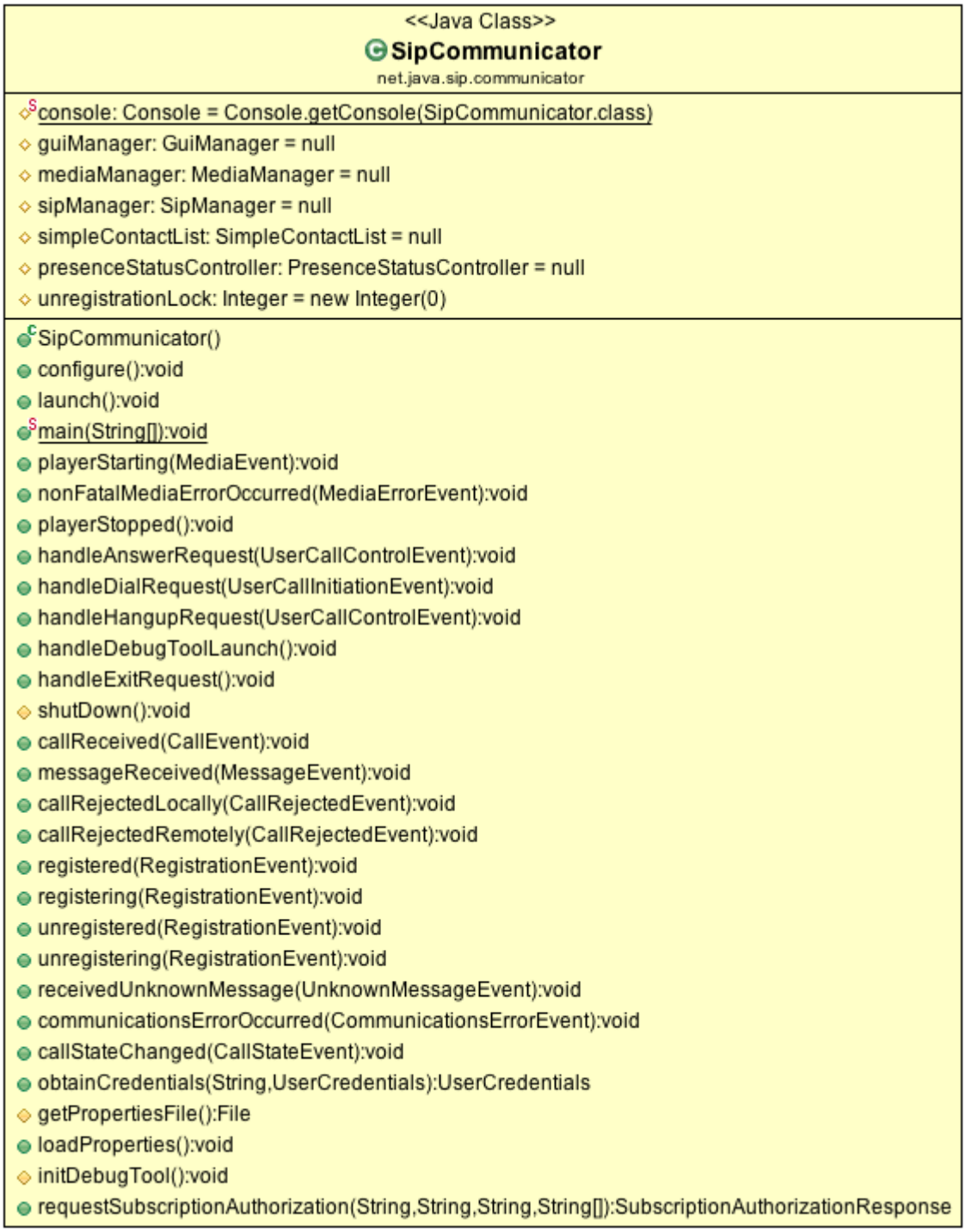
© Copyright 1998 Object Oriented Pty Ltd

4 Detailed Class Diagrams

4.1 Client (Sip Communicator)

4.1.1 net.java.sip.communicator

Αλλάξαμε τη μέθοδο communicationsErrorOccured() ώστε να εμφανίζει τα κατάλληλα διαγωνστικά μηνύματα όταν προκύπτει κάποιο error.



© Copyright 1998 Object Oriented Pty Ltd

4.1.2 net.java.sip.communicator.gui

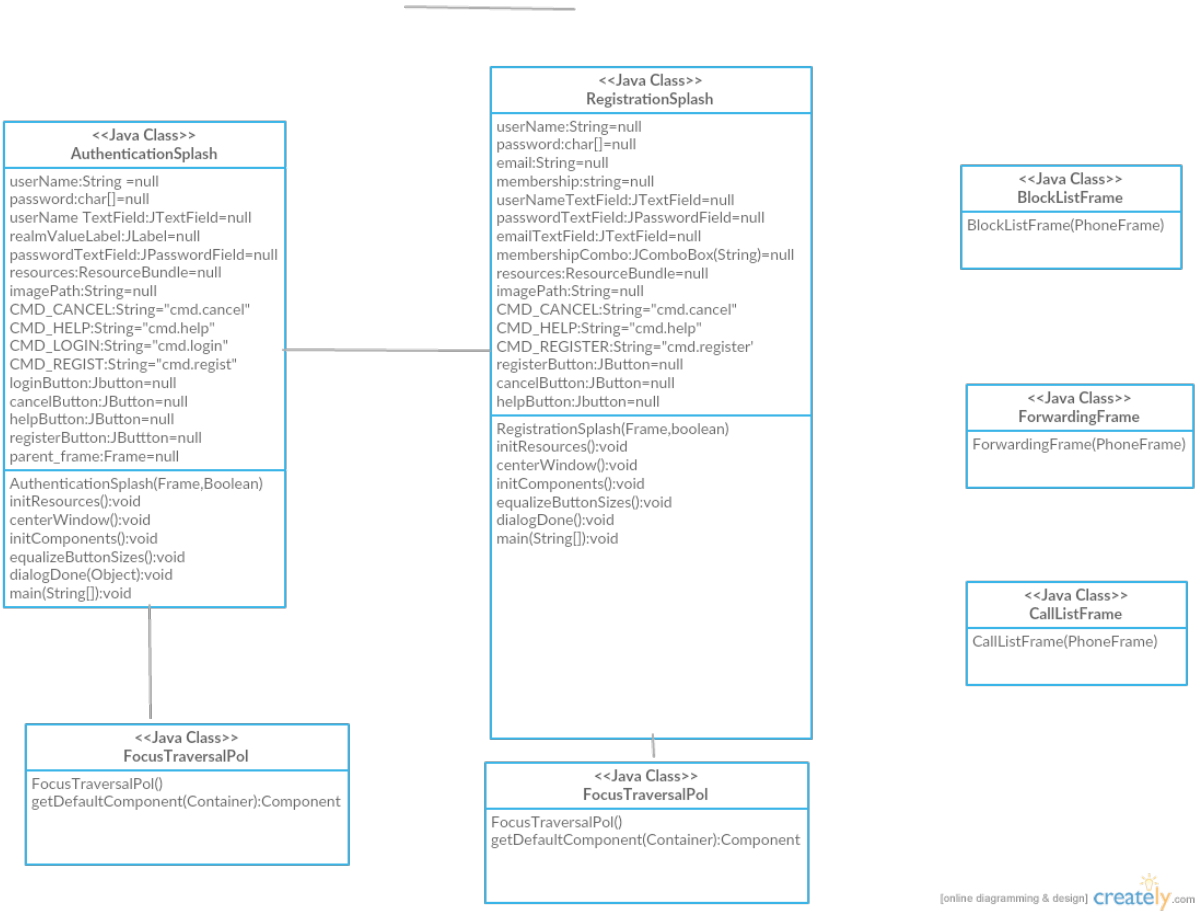
AuthenticationSplash: Προσθέσαμε ορισμένες σταθερές και μεταβλητές, και αλλάξαμε τη μέθοδο dialogDone() ώστε να υλοποιούνται σωστά οι ενέργειες registration και log in.

RegistrationSplash: Προσθέσαμε ορισμένες σταθερές και μεταβλητές, και αλλάξαμε τη μέθοδο dialogDone() ώστε να υλοποιούνται σωστά οι ενέργειες registration

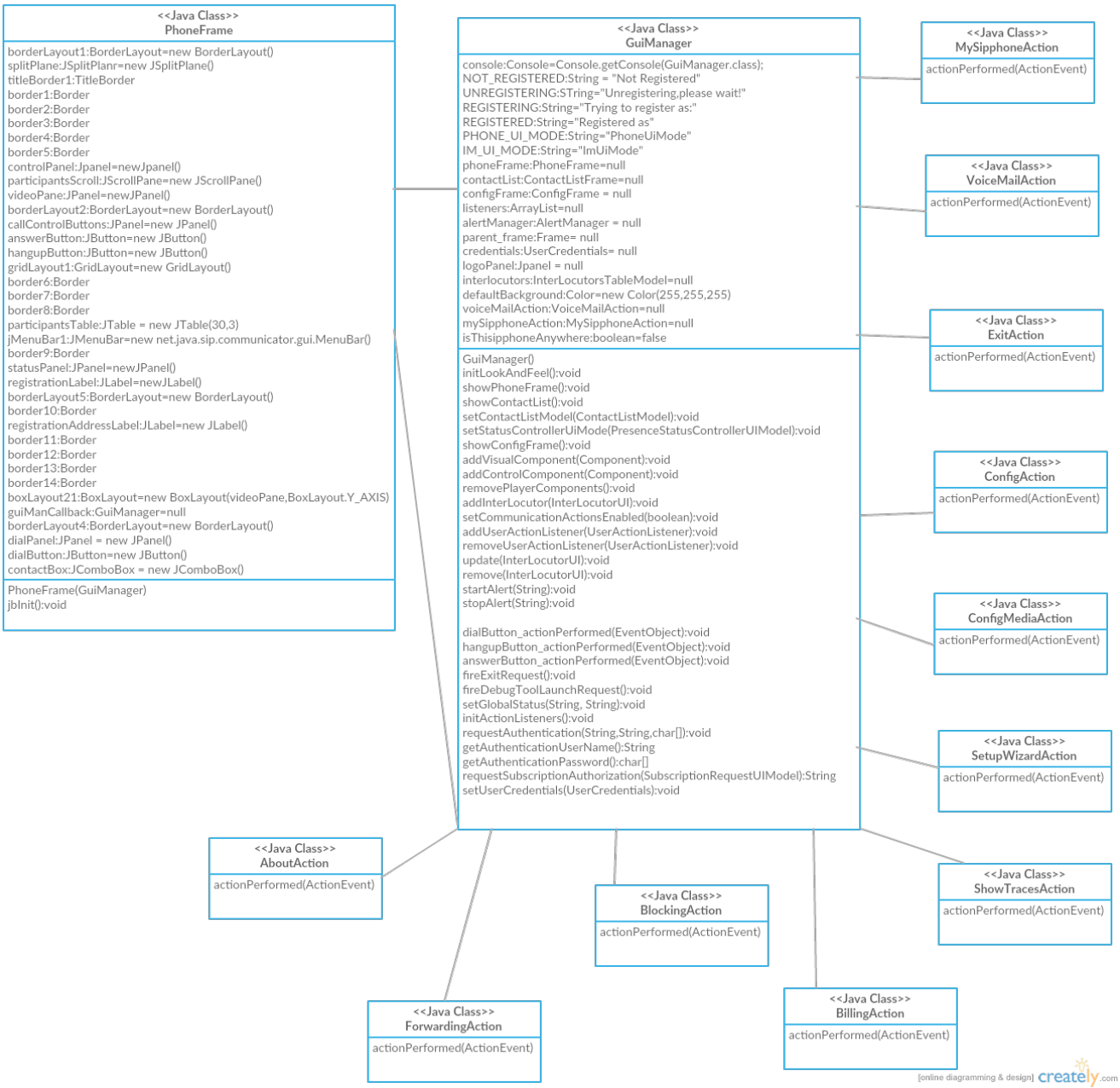
MenuBar: Προσθέσαμε την επιλογή “Services” στο menu bar και προσθέσαμε τις 3 νέες υπηρεσίες στο υπομενού

GuiManager: Προσθέσαμε μεθόδους ώστε να φαίνονται στο GUI τα νέα μας features

PhoneFrame: Προσθέσαμε τη μέθοδο “getRegistrationAddressLabel()” , η οποία μας επιτρέπει να γνωρίζουμε το username ενός συνδεδμένου user.



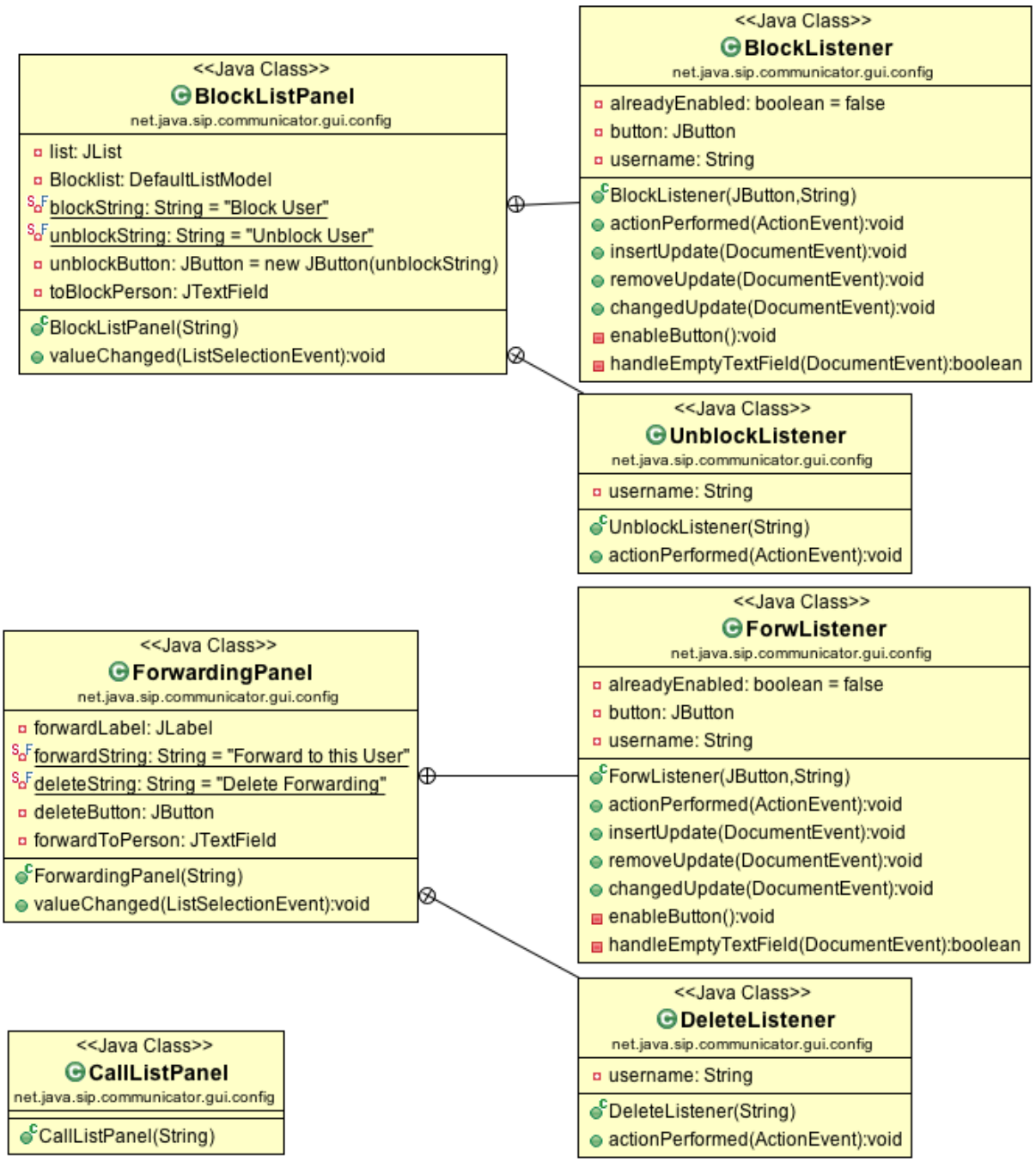
© Copyright 1998 Object Oriented Pty Ltd



4.1.3 net.java.sip.communicator.gui.config

Προσθέσαμε GUI panels και listeners για τα νέα μας features

© Copyright 1998 Object Oriented Pty Ltd



4.1.4 net.java.sip.communicator.sip

CallTimer: Νέα κλάση για να αποθηκεύει και να στέλνει προσωρινές πληροφορίες σχετικές με κάποια κλήση

Call: Προσθέσαμε μεθόδους για να γνωρίζουμε τους caller και callee. Προσθέσαμε μεταβλητή για να γνωρίζουμε εάν η κλήση προωθήθηκε. Προσθέσαμε αρχικοποίηση του CallTimer.

© Copyright 1998 Object Oriented Pty Ltd

Αλλάξαμε τη setState() μέθοδο για να χρησιμοποιεί τον CallTimer για να ορίσει το τέλος της κλήσης. Αλλάξαμε τη μέθοδο setInitialRequest() για αποθηκεύσουμε τους caller και callee στις νέες μεταβλητές – για το forwarding. Προσθέσαμε τη μέθοδο getInitialRequest() – για το forwarding.

CallProcessing: Αλλάξαμε την μέθοδο proccessTrying() για να υλοποιήσουμε το forwarding. Αλλάξαμε τη μέθοδο InviteOK() για να αρχικοποιήσουμε τον CallTimer με πληροφορίες από τον caller. Αλλάξαμε τη μέθοδο processAck() για να αρχικοποιήσουμε τον CallTimer με πληροφορίες από τον callee.

4.1.5 net.java.sip.communicator.Socket

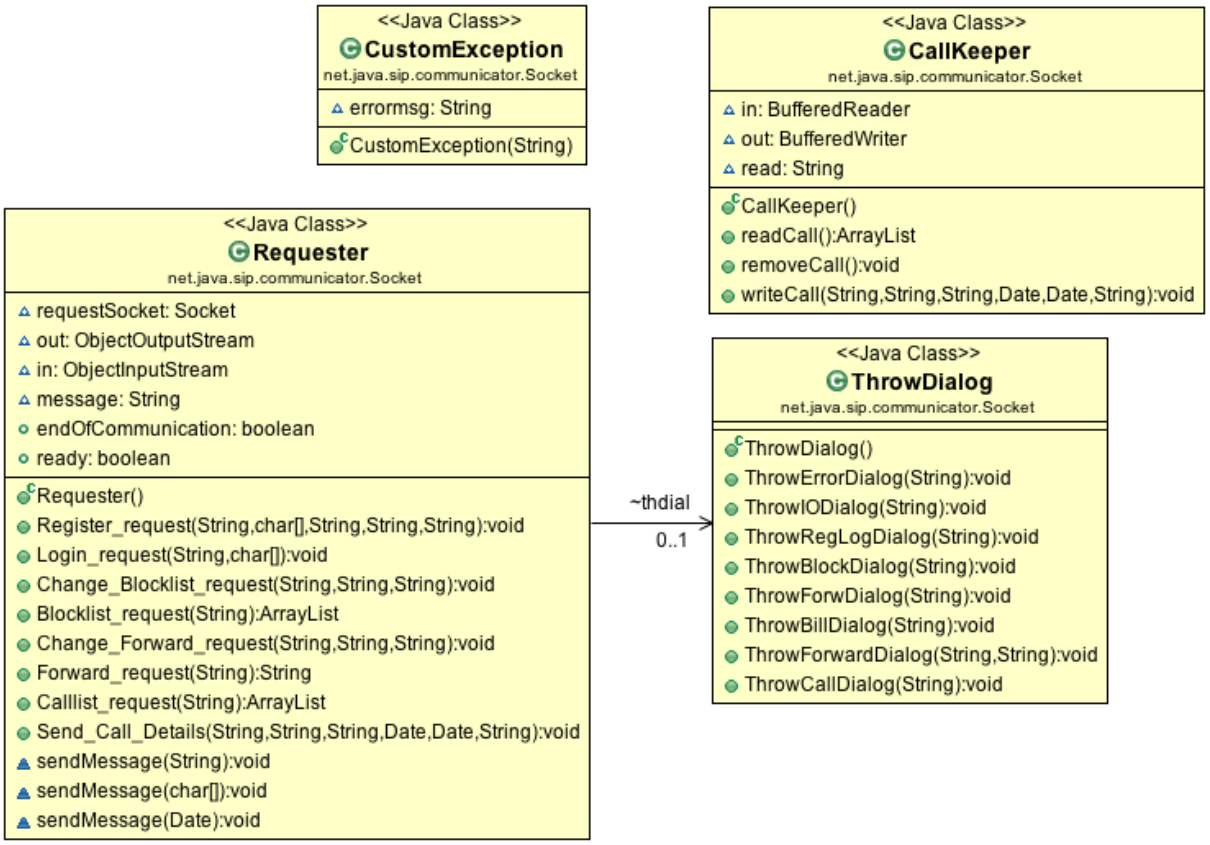
CallKeeper: Μια Class για να αποθηκευσουμε τις πληροφορίες της κλήσης σε ένα αρχείο, στην περίπτωση που προκύυει πρόβλημα στη μετάδοση τους, για να μπορούμε να τις στείλουμε στο server αργότερα.

© Copyright 1998 Object Oriented Pty Ltd

CustomException: Class for a new exception used to show messages to the user.

Requester: Class responsible for client-server communication through Socket for each feature that requires communication with the server.

ThrowDialog: Class that shows success or fail messages to user, depending on the result of his requests.



© Copyright 1998 Object Oriented Pty Ltd

4.2 Server (Sip Proxy)

Κλάση υπεύθυνη για τις πολιτικές χρέωσης των κλήσεων

4.2.1

govnist.sip.proxy

Configuration: Added variables to store settings like cost per category and Server Socket addressing.

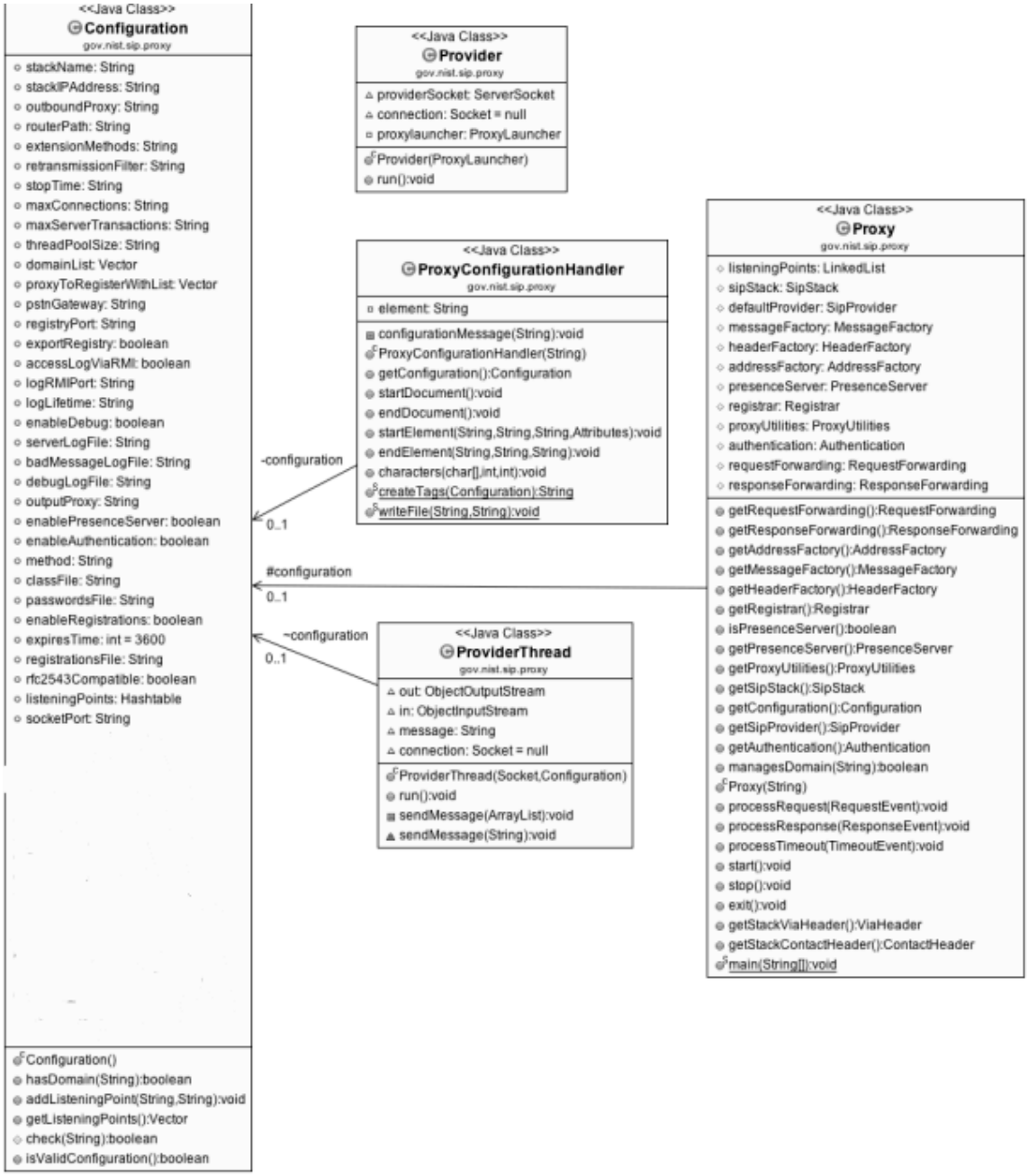
Provider: Class responsible for Server Socket setup that uses ProviderThread to start a new individual thread to serve each incoming request.

ProviderThread: Class which runs a new individual thread to serve a client's requests. Using threads allows for simultaneous user serving.

Proxy: Changed processRequest() method to add the blocking and forwarding features.

ProxyConfigurationHandler: Changed startElement() method to read the configuration file and set the billing plans and Socket address accordingly.

© Copyright 1998 Object Oriented Pty Ltd



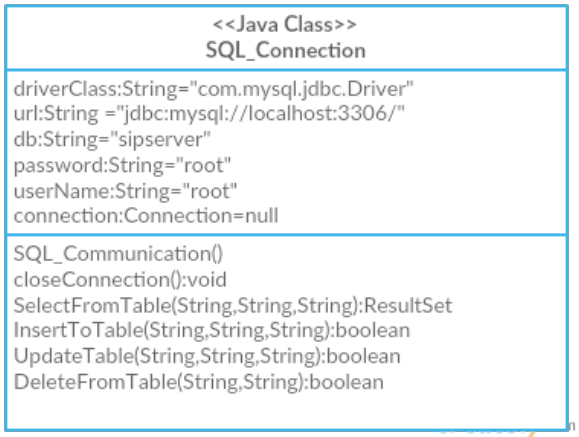
4.2.3 govnist.sip.proxy.db_com

DatabaseRequester: Class that uses jdbc to achieve communication with the SQL db.

SimpleMD5: Class that allows conversion of a string to MD5 hash so passwords are not stored as plain text in the database.

SQL_Communication: Class that accepts requests (select, insert delete and update) from the DatabaseRequester and executes them in the database.

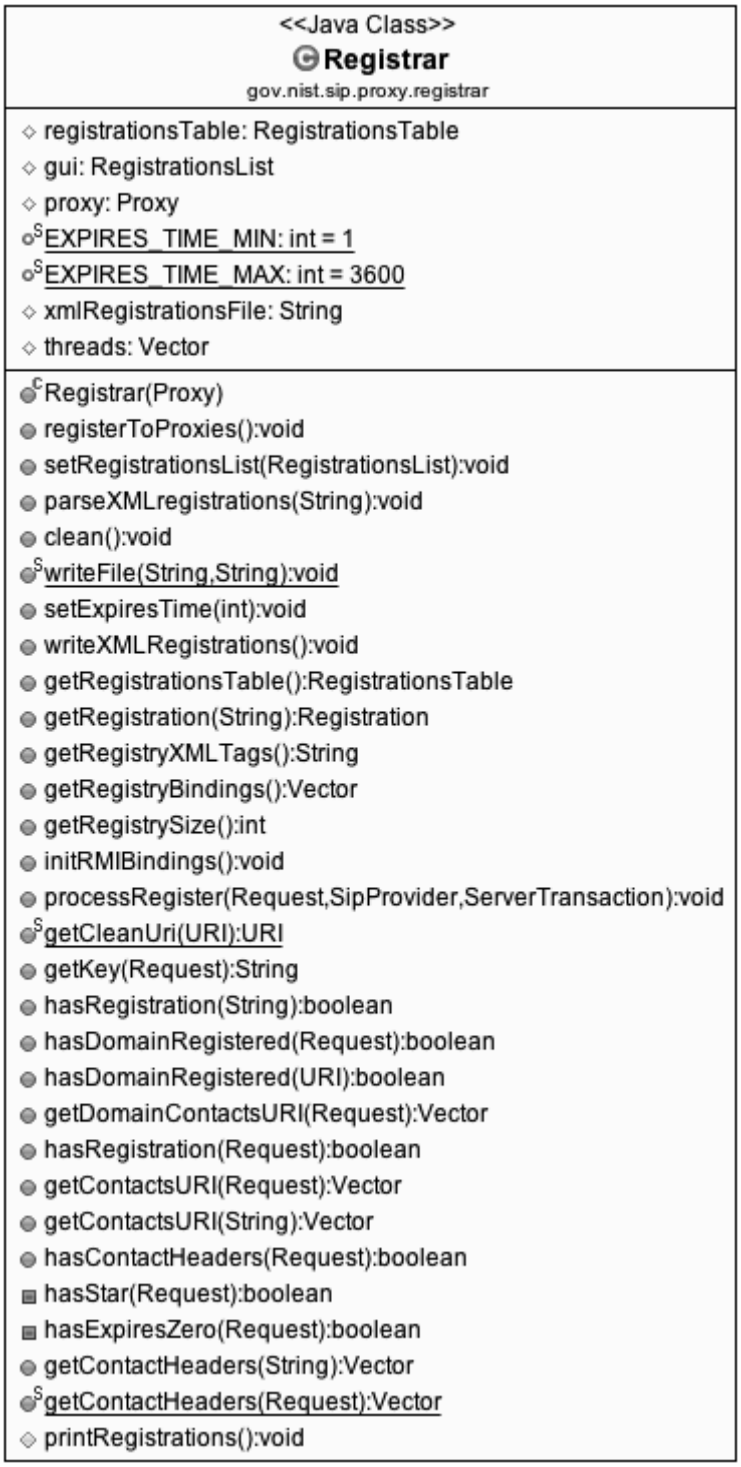
© Copyright 1998 Object Oriented Pty Ltd



4.2.5 govnist.sip.proxy.registrar

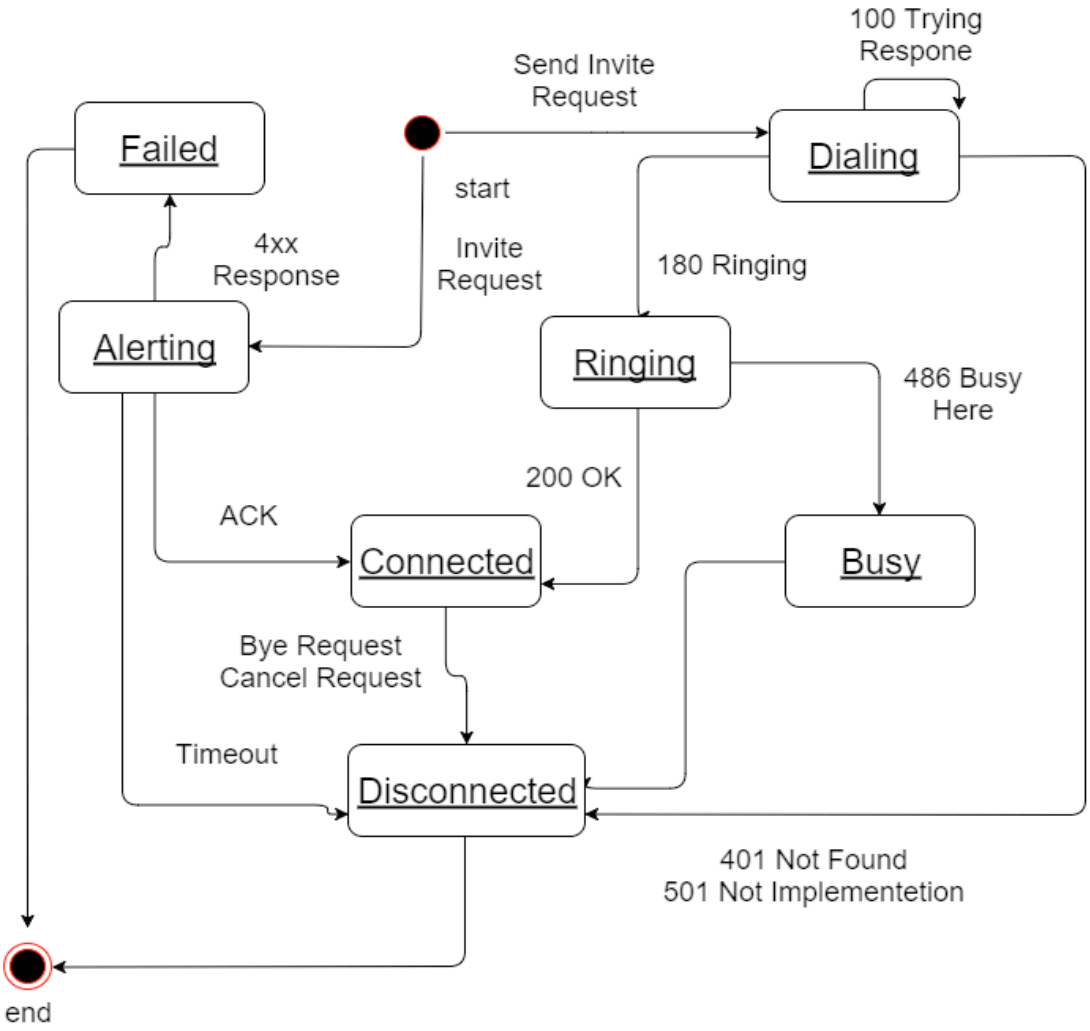
Changed parseXMLregistrations() and processRegister() methods to write connected users in an xml file. This file will be used in case of failure by the server, so it can restore the list of connected users before the failure.

© Copyright 1998 Object Oriented Pty Ltd



© Copyright 1998 Object Oriented Pty Ltd

5 State Diagrams



6 Open Issues

6.1 Κλήση όπου ο caller και ο callee είναι ο ίδιος χρήστης

Στην υλοποίησή μας, ένας χρήστης μπορεί να καλέσει τον εαυτό του και να λάβει αίτημα πρόσκλησης. Αυτό μπορεί να προκαλέσει προβλήματα στο μέλλον και για το λόγο αυτό σκοπεύουμε να το διορθώσουμε με το να προσθέσουμε ένα server ή έναν έλεγχο στη μεριά του πελάτη πριν ξεκινήσουμε την κλήση.

6.2 Επαλήθευση δεδομένων

Θα πρέπει να είμαστε σε θέση να ανιχνεύσουμε και να απορρίψουμε τροποποιημένα δεδομένα τα οποία στέλνονται στο server από καποιον πελάτη. Ειδικά,κατα τη διάρκεια της κλήσης ,η οποία μετράται από την πλευρά του πελάτη, θα μπορούσε να

© Copyright 1998 Object Oriented Pty Ltd

προκαλέσει αυξημένο κίνδυνο ασφάλειας.Οι χρήστες δεν πρέπει να έχουν πρόσβαση σε τέτοιες πληροφορίες.

6.3 Κρυπτογραφημένη server-client επικοινωνία

Στη συγκεκριμένη υλοποίηση , ο server και ο client επικοινωνούν μέσω μηνυμάτων απλού -μη κρυπτογραφημένου- κειμένου. Πράγμα το οποίο, προφανώς, δεν είναι μια ασφαλής πρακτική , ειδικότερα όταν μεταφέρονται "ευαίσθητα" δεδομένα (κωδικοί ή/και αριθμοί πιστωτικών καρτών). Ένα ασφαλές σχήμα κρυπτογράφησης πρέπει να προστεθεί , πιθανότερα χρησιμοποιώντας ασύμμετρη κρυπτογραφία - TLS.

6.4 Υλοποίηση πιστωτικού συστήματος

Όπως έχει υλοποιηθεί η εφαρμογή μας, ο Server , απλώς, χρεώνει έναν Serial Number ο οποίος δίνεται κατα την εγγραφή του χρήστη. Χρειάζεται να υλοποιήσουμε μια υπηρεσία που συνδέει το Serial Number σε μια πιστωτική/χρεωστική κάρτα ή κάποια άλλη μέθοδο πληρωμών(PayPal/Amazon Credit/PaySafe).

6.5 Επέκταση του Sip Proxy GUI

Σχεδιασμός ενός GUI για ένα Administrator Control Panel, το οποίο παρουσιάζει (μεταξύ άλλων) τις πρόσφατες αλλαγές που έγιναν στα νέα χαρακτηριστικά που προστέθηκαν.

6.6 Πολλαπλή προώθηση

Στην υλοποίηση αυτή, επιτρέπεται στο χρήστη να προσδιορίσει μόνο ένα άτομο στο οποίο θα προωθηθεί τις κλήσεις του. Θα έπρεπε να υπάρχει μια λίστα προώθησης με προτεραιότητα, έτσι ώστε σε περίπτωση που η πρώτη επιλεγμένη προώθηση αποτύχει, η κλήση να μην τελειώσει απ'ευθείας, αλλά να προσπαθήσει να την προωθήσει στους επόμενους χρήστες στη λίστα.

7 Domain Dictionary

a. Terms and Abbreviations

Term	Definition
Place term here	*Place a definition of the term here. Make the definition short and concise and consistent with other terms. Only used terms defined elsewhere in the domain dictionary.*

© Copyright 1998 Object Oriented Pty Ltd

Place synonym here	*This is a synonym of <another term>*
References	*Reference (URL, paper, book)*

© Copyright 1998 Object Oriented Pty Ltd