

Process MeNtOR 3.0

Uni-SEP

Επέκταση Πρωτοκόλλου SIP

Design Document

Version:	1.0
Print Date:	
Release Date:	
Release State:	Initial
Approval State:	
Approved by:	
Prepared by:	Χαρδούβελης Παναγιώτης -Ιάσων Ζάκκας Παύλος Παππάς Σπυρίδων
Reviewed by:	
Path Name:	
File Name:	design-document.doc
Document No:	

Document Change Control

Version	Date	Authors	Summary of Changes

Document Sign-Off

Name (Position)	Signature	Date
Χαρδούβελης Παναγιώτης -Ιάσων		21/12/2016
Ζάκκας Παύλος		21/12/2016
Παππάς Σπυρίδων		21/12/2016

Περιεχόμενα

[1 INTRODUCTION](#)

- 1.1 Overview
- 1.2 Resources – References

[2 MAJOR DESIGN DECISIONS](#)

- 2.1 Αποθήκευση δεδομένων και Έλεγχος Πρόσβασης Χρήστη
- 2.2 Υποστήριξη για Μπλοκάρισμα Χρήστη (Call Blocking)
- 2.3 Υποστήριξη για Προώθηση Κλήσης (Call Forwarding)
- 2.4 Υποστήριξη για Χρέωση Κλήσεων (Call Billing)

[3 ARCHITECTURE](#)

- 3.1 Component Diagram
- 3.2 Deployment Diagram

[4 DETAILED CLASS DIAGRAMS](#)

- 4.1 UML Class Diagrams
- 4.2 Method Details

[5 STATE DIAGRAMS](#)

[6 OPEN ISSUES](#)

1 Introduction

1.1 Overview

- Ενότητα 2:
Περιλαμβάνονται οι λεπτομέρειες σχετικά με το σχεδιασμό της εφαρμογής. Οι λεπτομέρειες αυτές έκριναν σε μεγάλο βαθμό την τελική υλοποίηση.
- Ενότητα 3:
Περιγράφονται οι δομικές μονάδες του συστήματος (αρχιτεκτονική νοοτροπία). Για την αναπαράστασή τους χρησιμοποιούνται ένα component και ένα deployment diagram.
- Ενότητα 4:
Γίνεται περιγραφή των κλάσεων του συστήματος και των μεταξύ τους σχέσεων. Χρησιμοποιούνται διαγράμματα κλάσεων και σχόλια.
- Ενότητα 5:
Αναλύεται η κλάση Call (η βασικότερη ίσως κλάση της εφαρμογής) μέσω ενός state diagram.
- Ενότητα 6:
Γίνεται αναφορά στα ανοικτά ζητήματα της εφαρμογής για μελλοντική ανάπτυξη.
- Ενότητα 7
Στην τελευταία ενότητα παραθέτουμε τον κώδικα δημιουργίας της βάσης που χρησιμοποιήθηκε.

1.2 Resources – References

- RFC: <http://www.ietf.org/rfc/rfc3261.txt>
- Διαγράμματα και σημειώσεις στην ιστοσελίδα του μαθήματος
- SIP: http://en.wikipedia.org/wiki/Session_Initiation_Protocol
- Τα διαγράμματα δημιουργήθηκαν με χρήση των online εφαρμογών www.lucidchart.com και draw.io

2 Major Design Decisions

Παρακάτω θα αναλύσουμε τις πιο σημαντικές σχεδιαστικές αποφάσεις που πήραμε για κάθε νέο feature που υλοποιήσαμε.

2.1 Αποθήκευση δεδομένων και Έλεγχος Πρόσβασης Χρήστη

Το πρώτο feature που θελήσαμε να υλοποιήσουμε είναι η μόνιμη αποθήκευση των δεδομένων ενός χρήστη, από τη στιγμή που αυτός δηλωθεί στο σύστημα. Για να επιτευχθεί αυτό, δημιουργήσαμε στον proxy μία βάση δεδομένων στην οποία αποθηκεύονται τα δεδομένα των χρηστών (ονοματεπώνυμο, διεύθυνση, username, password, χρεωστικό πρόγραμμα, ρυθμίσεις για blocking και forwarding κλπ). Η σχεδιαστική αυτή επιλογή έγινε για να μπορεί ο proxy να εξυπηρετεί με άνεση μεγάλο αριθμό χρηστών. Σημειώνεται εδώ ότι ο κατάλογος των χρηστών που έχουν συνδεθεί (ο οποίος κρατείται από τον location server) δεν αποθηκεύεται στη βάση δεδομένων, αλλά δημιουργείται και ενημερώνεται στη RAM.

Με το παραπάνω επιτυγχάνουμε τα εξής:

1. Ένας χρήστης δεν μπορεί να κάνει log in (και άρα να πραγματοποιήσει κλήση) στην εφαρμογή αν προηγουμένως δεν έχει κάνει sign up.
2. Ο κάθε χρήστης δεν χρειάζεται να αποθηκεύει κάθε φορά τα στοιχεία του και τις διάφορες παραμετροποιήσεις που αφορούν τον λογαριασμό του.

2.2 Υποστήριξη για Μπλοκάρισμα Χρήστη (Call Blocking)

Οι διάφορες παραμετροποιήσεις του κάθε χρήστη που αφορούν στο blocking αποθηκεύονται μόνιμα στη βάση δεδομένων και οι χρήστες μπορούν να τις επεξεργαστούν μέσω της εφαρμογής. Η υλοποίησή μας δεν επιτρέπει σε έναν χρήστη που έχει μπλοκαριστεί από έναν άλλον να τον καλέσει.

2.3 Υποστήριξη για Προώθηση Κλήσης (Call Forwarding)

Όπως και στα προηγούμενα, οι ρυθμίσεις που αφορούν τις προωθήσεις κλήσεων για κάθε χρήστη αποθηκεύονται στη βάση δεδομένων. Οι χρήστες έχουν τη δυνατότητα να τις επεξεργαστούν μέσω της εφαρμογής. Σημειώνεται εδώ ότι επιτρέπονται πολλαπλές προωθήσεις μεταξύ χρηστών, αρκεί αυτές να μην δημιουργούν κύκλους. Ο έλεγχος για τη δημιουργία κύκλων αποφασίστηκε ότι είναι βέλτιστο να διεξάγεται στην εφαρμογή που τρέχει στον client, έτσι ώστε αμέσως πριν καταχωρηθεί ένα forwarding, να γνωρίζουμε αν αυτό είναι valid ή όχι, και αυτό να μην πρέπει να διαπιστωθεί κατά τη διάρκεια της κλήσης. Αν υπάρχει προώθηση, και ο χρήστης στον οποίο προωθείται η κλήση έχει μπλοκάρει τον καλών, η κλήση δεν πραγματοποιείται (όπως στην προηγούμενη ενότητα).

2.4 Υποστήριξη για Χρέωση Κλήσεων (Call Billing)

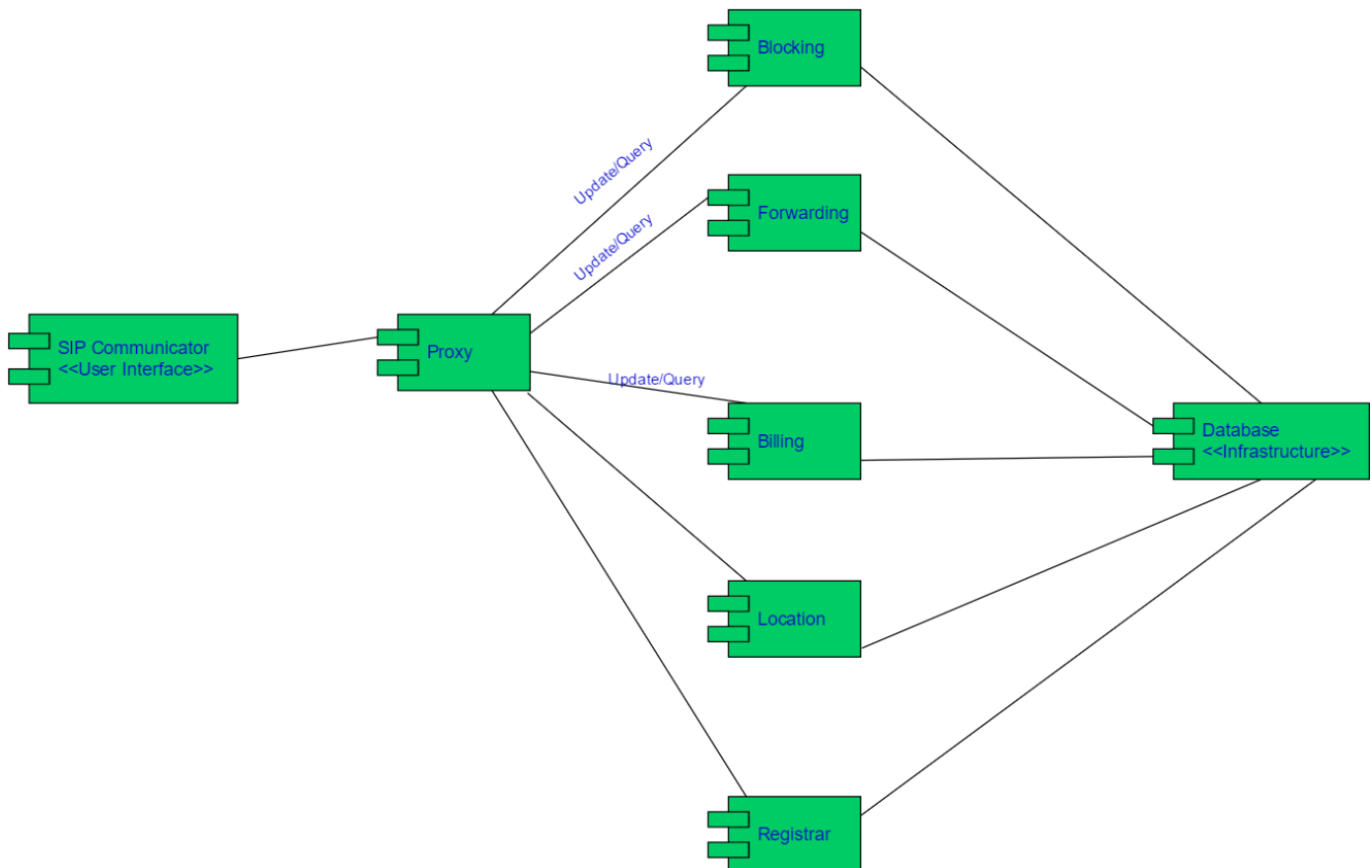
Κατά την εγγραφή ενός χρήστη στην εφαρμογή, αυτός επιλέγει το χρεωστικό πρόγραμμα που του ταιριάζει. Με βάση αυτό, κάθε φορά που πραγματοποιείται μια

έγκυρη κλήση δημιουργείται μία εγγραφή στον αντίστοιχο πίνακα με την ώρα έναρξης της κλήσης. Όταν η κλήση τελειώσει, σημειώνεται η ώρα λήξης και με βάση αυτά, όταν και αν ζητηθεί από τον χρήστη, υπολογίζεται η διάρκεια και η χρέωση της κλήσης, ανάλογα με το χρεωστικό πρόγραμμα που έχει επιλεγεί. Η χρέωση των κλήσεων γίνεται με ακρίβεια δευτερολέπτου.

3 Architecture

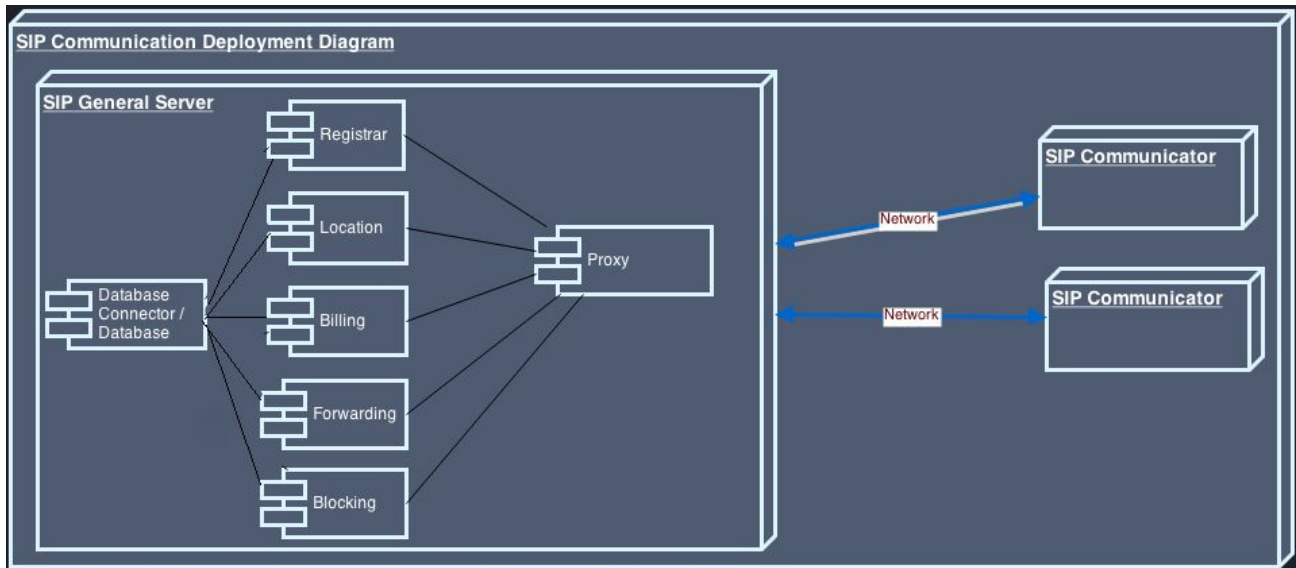
3.1 Component Diagram

Στην συνέχεια έχουμε το component diagram (ψηφιδωτό διάγραμμα) του συστήματος:



3.2 Deployment Diagram

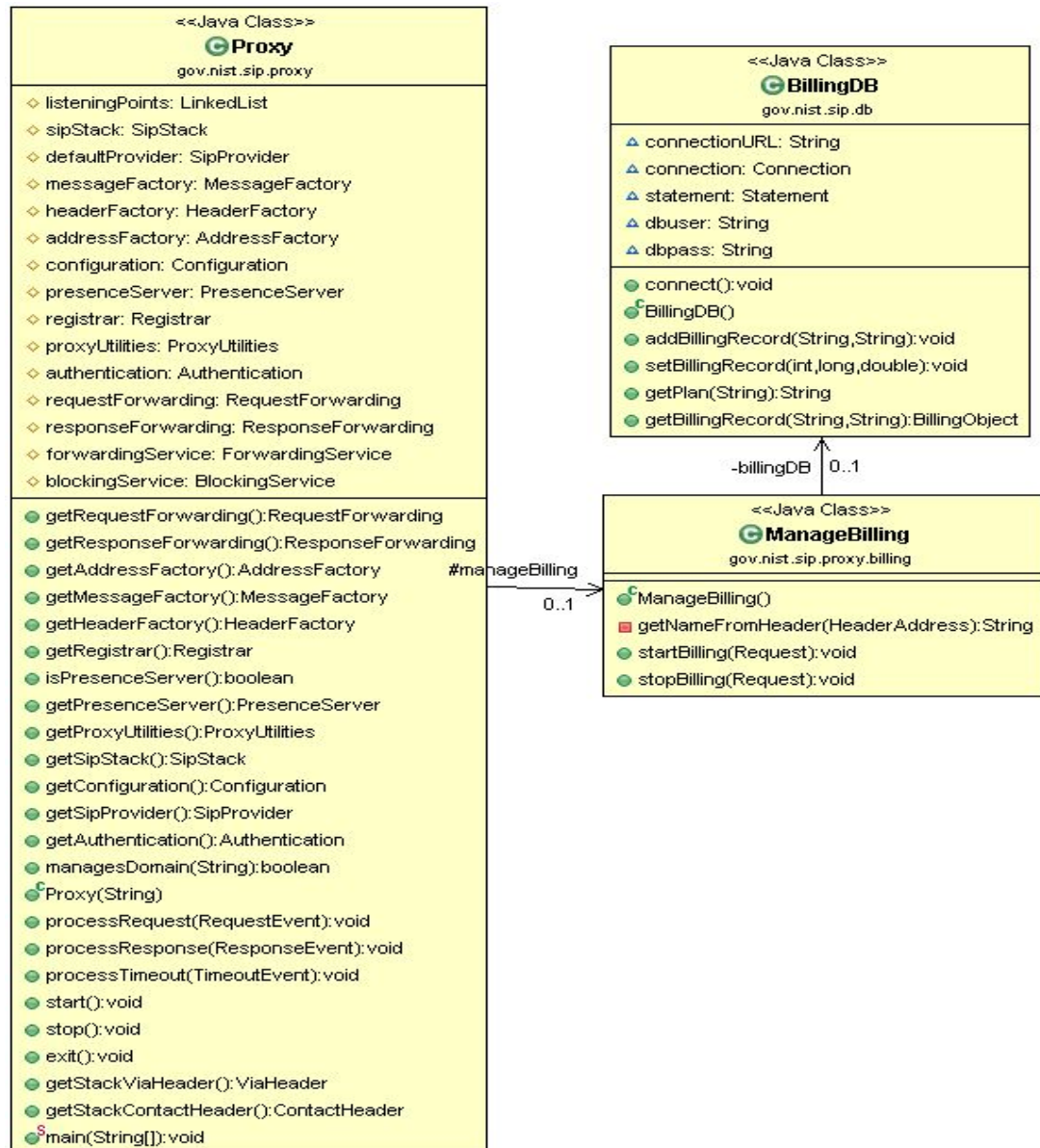
Ακολουθεί το deployment diagram (παραταξιακό διάγραμμα) του συστήματος:



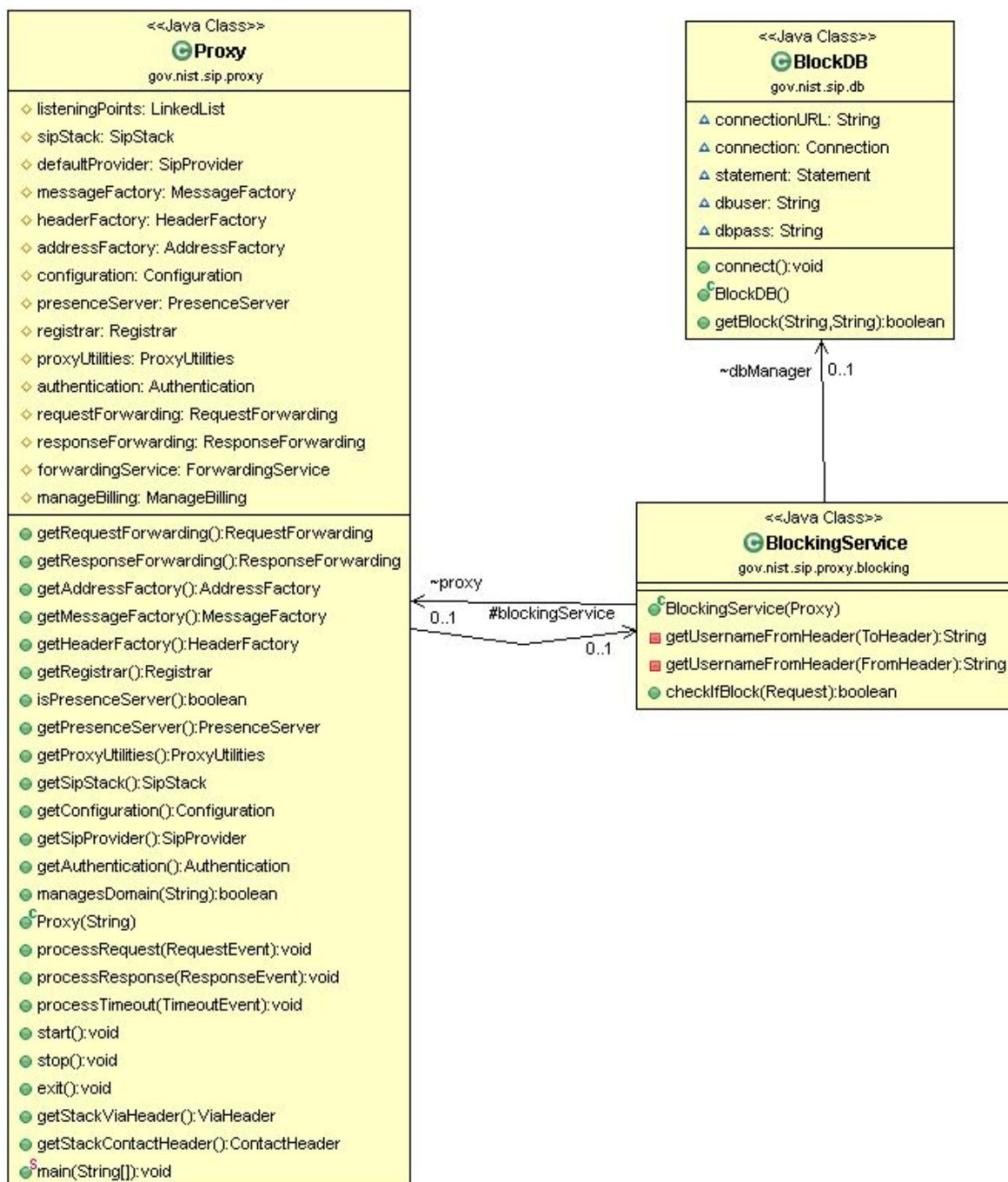
4 Detailed Class Diagrams

4.1 Class Diagrams

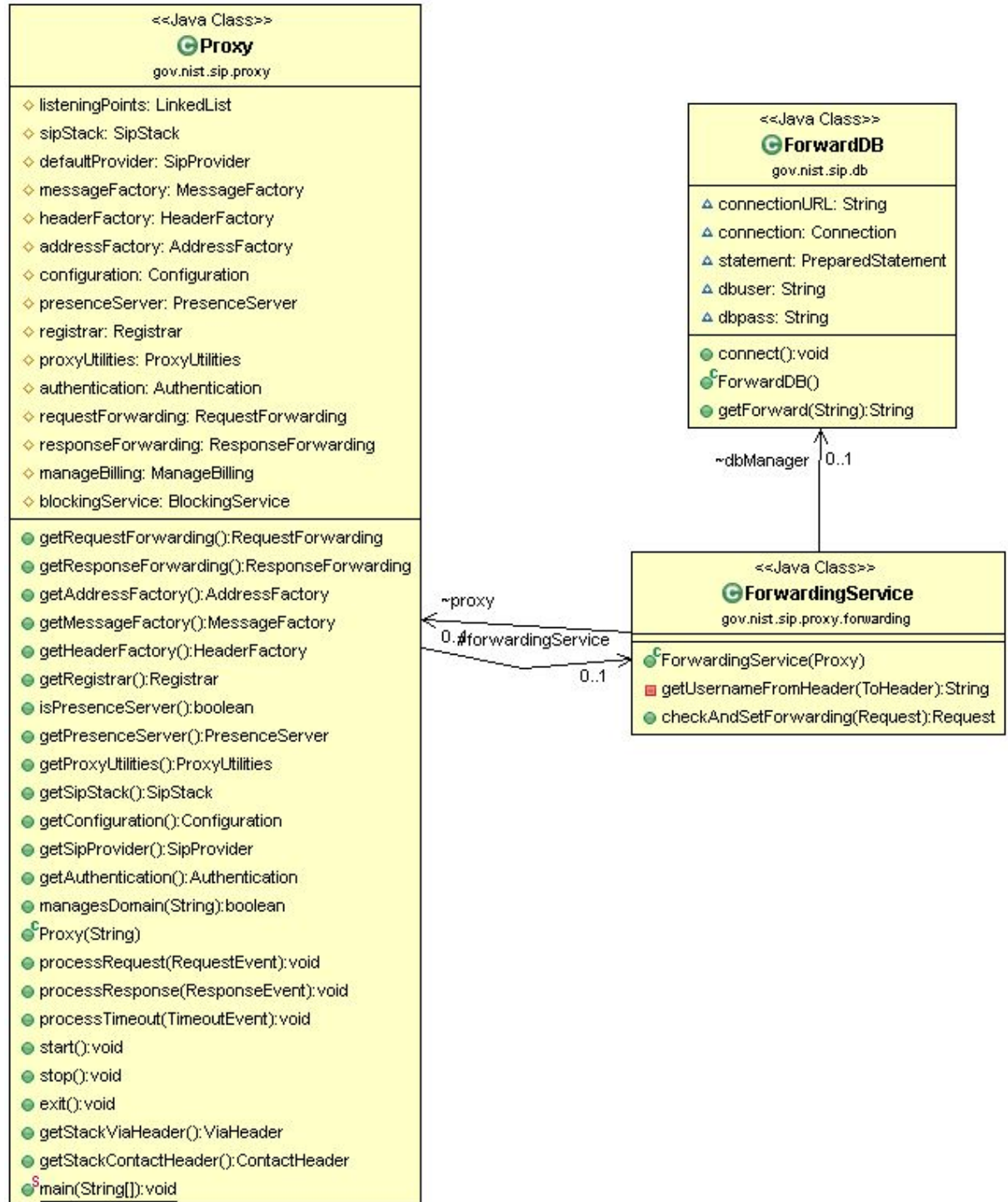
Η κλάση **ManageBilling** αναλαμβάνει τη χρέωση των κλήσεων. Για το σκοπό αυτό παρακολουθεί της διάρκεια της επικοινωνίας, τα στοιχεία των χρηστών, καθώς και το πρόγραμμα που αυτοί έχουν επιλέξει. Τα στοιχεία των χρηστών αντλούνται από τη βάση μέσω της κλάσης **BillingDB**. Στο τελικό project θα υλοποιήσουμε διαφορετικές πολιτικές χρέωσης.



Η κλάση **BlockingService** διαχειρίζεται τα μπλοκαρίσματα μεταξύ των χρηστών. Για να αντλήσει πληροφορίες για ήδη υπάρχοντα μπλοκαρίσματα, και να πραγματοποιήσει τους απαραίτητους ελέγχους χρησιμοποιεί την κλάση **BlockDB**, η οποία διαχειρίζεται τον αντίστοιχο πίνακα στη βάση.



Η κλάση **ForwardingService** διαχειρίζεται τις προωθήσεις των κλήσεων μεταξύ των χρηστών. Μέσω της κλάσης **ForwardDB** αλληλεπιδρά με την βάση δεδομένων.



4.2 Method Details

Ακολουθως γίνεται μια ανάλυση των μεθόδων των κλάσεων. Μερικές μέθοδοι δεν απεικονίζονται απο το export του eclipse:

BillingDB:

- BillingDB

- Δημιουργία σύνδεσης στη βάση
- addBillingRecord
 - Κάνει καταχώρηση της κλήσης και της ώρας έναρξης.
- setBillingRecord
 - Ενημερώνει μία καταχώρηση κλήσης για τον τερματισμό της με την διάρκειά της και το κόστος της.
- getBillingRecord
 - Επιστρέφει τα στοιχεία μιας καταχωρημένης κλήσης που δεν έχει ολοκληρωθεί ακόμα.
- getPlan
 - Επιστρέφει την πολιτική χρέωσης το χρήστη με το δοσμένο username

BlockDB:

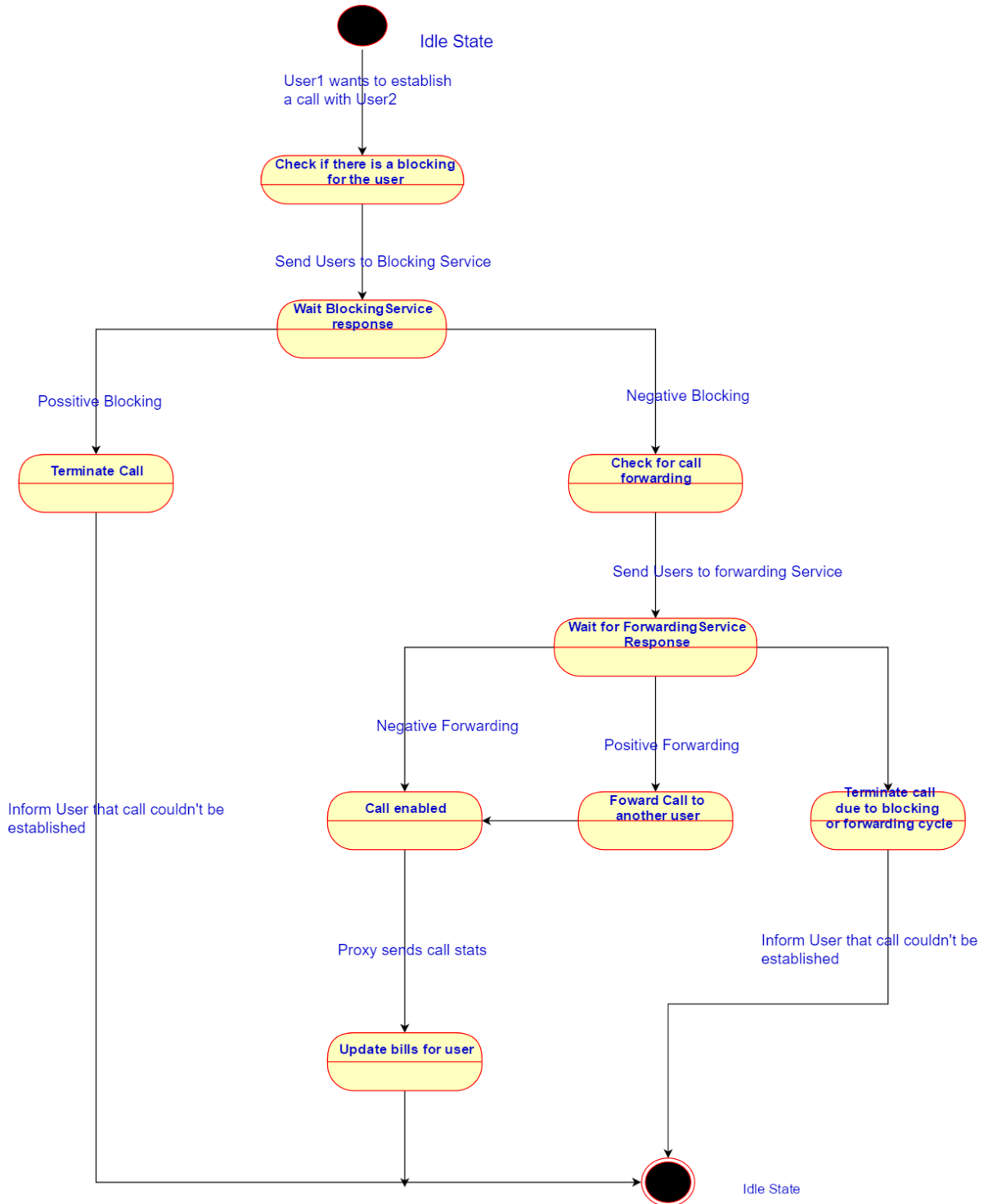
- BlockDB
 - Δημιουργία σύνδεσης στη βάση
- getBlock
 - Δέχεται τους Callee και Caller και επιστρέφει αν είναι ή όχι μπλοκαρισμένος ο Caller.

ForwardDB:

- ForwardDB
 - Δημιουργία σύνδεσης στη βάση
- getForward
 - Βρίσκει τον τελικό χρήστη στον οποίο πρέπει να προωθηθεί η κλήση

5 State Diagrams

Ακολουθως απεικονίζεται το διάγραμμα καταστάσεων (state diagram) κατά τη διάρκεια μια κλήσης:



6 Open Issues

User Interface

Η διεπαφή της εφαρμογής που χρησιμοποιούν οι χρήστες, αν και λειτουργική, θα μπορούσε να βελτιωθεί αρκετά.

Αναβάθμιση της διαδικασίας call billing για υποστήριξη παθολογικών σεναρίων

Σε περιπτώσεις crashing του proxy ή του client, δε στέλνεται το μήνυμα BYE για τερματισμό της κλήσης και κατ' επέκταση της χρέωσης. Είναι σημαντικό λοιπόν να προβλεφθούν τα παραπάνω σενάρια μέχρι η εφαρμογή να βγει στην αγορά.