

Επέκταση Πρωτοκόλλου SIP

Προδιαγραφές Σχεδίασης

Έκδοση:	1.2
Ημ/νία Εκτύπωσης:	
Ημ/νία Έκδοσης:	08/03/2014
Κατάσταση Έκδοσης:	Βασικό
Κατάσταση Έγκρισης:	Προσχέδιο
Εγκρίθηκε από:	
Ετοιμάστηκε από:	Σαρήλης Δημήτριος (03109078) Σταθακοπούλου Χρυσούλα (03109065) Τζαννέτος Δημήτριος (03109010)
File Name:	SDD_SIP_Team20.pdf

Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ.....	3
1.1	Περίληψη.....	3
1.2	Αναφορές.....	3
2	ΘΕΜΕΛΙΩΔΕΙΣ ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΠΟΦΑΣΕΙΣ.....	4
2.1	Δημιουργία Βάσης Δεδομένων.....	4
2.1.1	Στοιχεία Χρήστη.....	4
2.1.2	Στοιχεία Κλήσεων.....	4
2.1.3	Λίστα Φραγής.....	5
2.1.4	Λίστα Προώθησης.....	5
2.1.5	Αγαπημένοι Χρήστες.....	5
2.1.6	Στοιχεία Πολιτικής Χρέωσης.....	5
2.2	Συμβατότητα με την υπάρχουσα έκδοση και με το πρωτόκολλο.....	6
2.2.1	Ο Proxy server μεσολαβεί στη φραγή και την προώθηση.....	6
2.2.2	Επικοινωνία client εφαρμογής με τη βάση δεδομένων για εισαγωγή και ενημέρωση στοιχείων χρήστη.....	6
2.3	Πολιτική χρέωσης.....	6
3	ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....	7
3.1	Ψηφιδικά Διαγράμματα.....	7
3.2	Παραταξιακά Διαγράμματα.....	8
4	ΛΕΠΤΟΜΕΡΗ ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ.....	9
4.1	Διαγράμματα κλάσεων σε UML.....	9
4.2	Λεπτομέρειες μεθόδων.....	11
4.2.1	SipProxy.....	11
4.2.2	SIP Communicator.....	14
5	ΔΙΑΓΡΑΜΜΑ ΚΑΤΑΣΤΑΣΗΣ.....	18
6	ΑΝΟΙΧΤΑ ΖΗΤΗΜΑΤΑ.....	19
6.1	Ανεξέλεγκτη χρέωση καλούντος.....	19
6.2	Ανίχνευση κύκλων στην προώθηση κλήσης.....	19
6.3	Σχεσιακό Μοντέλο Βάσης.....	19
6.4	Πολιτικές Χρέωσης.....	19

1 Εισαγωγή

1.1 Περίληψη

Σκοπός της παρούσας εργασίας είναι η επέκταση και η υλοποίηση συγκεκριμένων εφαρμογών των Jain-Sip-Presence-Proxy και Sip Communicator, οι οποίες θα υλοποιούν την επικοινωνία μεταξύ των χρηστών που είναι εγγεγραμμένοι στο πρόγραμμα και συνδεδεμένοι στον server, σύμφωνα με το πρότυπο RFC 3261 που ορίζει το πρωτόκολλο SIP. Συγκεκριμένα, έχουμε να υλοποιήσουμε 3 επιπλέον λειτουργίες:

- Φραγή κλήσεων(call blocking), μέσω της οποίας ένα χρήστης μπορεί να επιλέξει να μη δέχεται κλήσεις από άλλους χρήστες.
- Προώθηση κλήσεων(call forwarding), λειτουργία που επιτρέπει στους χρήστες να ορίσουν ότι οι κλήσεις που δέχονται θα προωθούνται σε κάποιον τρίτο χρήστη.
- Χρέωση κλήσεων(call billing) σύμφωνα με μια τιμολογιακή πολιτική που έχει επιλέξει ο καλών χρήστης.

Θα πρέπει επίσης να αντιμετωπιστούν κάποιες παθολογικές καταστάσεις, όπως αποτυχίες σύνδεσης ή βίαιης διακοπής σύνδεσης.

Το παρόν έγγραφο προδιαγραφών σχεδίασης οργανώνεται ως εξής:

- Στο Κεφάλαιο 2 σχολιάζουμε τις κύριες σχεδιαστικές αποφάσεις που λάβαμε κατά τη σχεδίαση του έργου.
- Στο Κεφάλαιο 3 συμπεριλαμβάνουμε ψηφιδικά διαγράμματα (component diagrams) τα οποία μοντελοποιούν τη δομή του συστήματος σαν ένα σύνολο από υπομονάδες και τις σχέσεις που τις συνδέουν. Παρουσιάζουμε επίσης τα παραταξιακά διαγράμματα (deployment diagrams), τα οποία μοντελοποιούν τη σχετική διάταξη των υπομονάδων κατά την εκτέλεση και λειτουργία του συστήματος. Τέλος, αναφέρουμε τα κύρια αρχιτεκτονικά στυλ που χρησιμοποιήσαμε στη σχεδίαση του συστήματος.
- Στο Κεφάλαιο 4 συμπεριλαμβάνουμε διαγράμματα κλάσεων (class diagrams) για κάθε κλάση που τροποποιήσαμε ή αλλάξαμε κατά τη σχεδίαση. Ακόμη, αναφέρουμε τα σχεδιαστικά μορφώματα που χρησιμοποιήσαμε.
- Στο Κεφάλαιο 5 παραθέτουμε ένα διάγραμμα κατάστασης (state diagrams) για την κύρια κλάση (Call).
- Στο Κεφάλαιο 6 σχολιάζουμε κάποια ανοιχτά ζητήματα σχεδίασης που ακόμα δεν έχουν οριστικοποιηθεί.

1.2 Αναφορές

- [1] RFC 3261 - SIP: Session Initiation Protocol: <http://www.ietf.org/rfc/rfc3261.txt>
- [2] Eclipse: <http://www.eclipse.org/downloads/index.php>
- [3] SIP Proxy: Από τον ιστότοπο του μαθήματος στο mycourses.ntua.gr
- [4] SIP Communicator: Από τον ιστότοπο του μαθήματος στο mycourses.ntua.gr
- [5] Project Description: <http://courses.softlab.ntua.gr/softeng/project-sip.html>

2 Θεμελιώδεις Σχεδιαστικές Αποφάσεις

Οι παρακάτω σχεδιαστικές αποφάσεις λήφθηκαν με κριτήριο τη λειτουργικότητα και αποδοτικότητα της εφαρμογής. Μέριμνα λήφθηκε επιπλέον για την κατά το δυνατό συμβατότητα με την ήδη υπάρχουσα εφαρμογή, και για την εξασφάλιση της εύκολης συντήρησης και της δυνατότητας για περαιτέρω επέκταση της εφαρμογής.

2.1 Δημιουργία Βάσης Δεδομένων

Για την υλοποίηση των επιπλέον λειτουργικοτήτων (δυνατότητα εγγραφής/εισόδου στο σύστημα, φραγής κλήσεων, προώθησης κλήσεων και χρέωσης βάσει πολιτικής) κρίθηκε σκόπιμη η υλοποίηση μιας βάσης δεδομένων. Επιλέξαμε να αποθηκεύουμε πληροφορίες σε βάση δεδομένων για να είναι το σύστημα πιο εύκολα παραμετροποιήσιμο. Επιπλέον, προτιμήσαμε τη χρήση μιας βάσης δεδομένων από ένα xml document για λόγους ευχρηστίας και αποδοτικότητας.

Στη βάση θα υπάρχουν οι εξής πίνακες:

2.1.1 Στοιχεία Χρήστη

Στον πίνακα αυτόν θα κρατούνται στοιχεία για την εισαγωγή του χρήστη στο σύστημα καθώς και για την πολιτική χρέωσης που χρησιμοποιεί και τη συνολική χρέωση του για τον τρέχοντα μήνα:

users
id
username
password
policy_id
total_cost

2.1.2 Στοιχεία Κλήσεων

Στον πίνακα αυτόν θα περιέχονται στοιχεία για την τιμολόγηση των κλήσεων. Απαραίτητη είναι η γνώση τόσο του καλούντος όσο και του καλούμενου καθώς η χρέωση της κλήσης στον πρώτο μπορεί να υπόκειται ειδική χρέωση ανάλογα με την ταυτότητα του δεύτερου. Επιπλέον απαραίτητη είναι η γνώση της ακριβούς ημερομηνίας έναρξης και λήξης της κλήσης καθώς μπορεί να εμπίπτει σε ειδική κατηγορία, ανάλογα με την πολιτική χρέωσης του καλούντος :

calls
id
caller
callee
time_start
time_end
cost

2.1.3 Λίστα Φραγής

Ο πίνακας αυτός αντιστοιχίζει ένα χρήστη με τους χρήστες που μπλοκάρει. Κάθε χρήστης θα μπορεί να εμφανίζεται σε μία ή περισσότερες εγγραφές δεδομένου ότι μπορεί να μπλοκάρει περισσότερου από ένα χρήστες. Κάθε συνδυασμός (blocker_id, blockee_id) θα είναι όμως μοναδικός.

blocking_list
blocker
blockee

2.1.4 Λίστα Προώθησης

Ο πίνακας αυτός αντιστοιχίζει ένα χρήστη με το χρήστη στον οποίο έχει επιλέξει να προωθεί τις κλήσεις του. Αν δεν προωθεί κάπου τις κλήσεις του το αντίστοιχο πεδίο θα είναι null :

forwarding_list
forwarder
forwardee

2.1.5 Αγαπημένοι Χρήστες

Ο πίνακας αυτός θα αντιστοιχίζει ένα χρήστη με τους χρήστες που έχει επιλέξει ως «αγαπημένους» του. Θα εξυπηρετεί σε υλοποίηση πολιτικής χρέωσης για την οποία θα υπάρχει ειδική χρέωση για ομάδα «αγαπημένων» χρηστών.

favorites
username
faved

2.1.6 Στοιχεία Πολιτικής Χρέωσης

Ο πίνακας αυτός θα κρατάει στοιχεία για τις διάφορες πολιτικές χρέωσης. Επιλέγουμε να αποθηκεύουμε αυτές τις πληροφορίες στη βάση δεδομένων ώστε το σύστημα να είναι εύκολα παραμετροποιήσιμο. Ενδεικτικά:

policy
id
name
cost_per_min
weekends_discount
late_night_discount
favorite_group_discount

2.2 Συμβατότητα με την υπάρχουσα έκδοση και με το πρωτόκολλο

2.2.1 Ο Proxy server μεσολαβεί στη φραγή και την προώθηση

Θέλοντας να διατηρήσουμε τον υπάρχοντα τρόπο εγκατάστασης της κλήσης, οι επιπλέον λειτουργικότητες για την φραγή και προώθηση κλήσεων κρύβονται από τους clients. Οι clients εξακολουθούν απλά να στέλνουν πρόσκληση στον Proxy Server σύμφωνα με το SIP πρωτόκολλο. Ο Proxy Server εμπλουτίζεται ώστε να ελέγχει αν μια κλήση μπλοκάρεται ή προωθείται.

2.2.2 Επικοινωνία client εφαρμογής με τη βάση δεδομένων για εισαγωγή και ενημέρωση στοιχείων χρήστη

Στα πλαίσια της υλοποίησης των επιπλέον λειτουργικοτήτων θα πρέπει να μπορούν να εισαχθούν και ενημερωθούν στη βάση δεδομένων πληροφορίες που αφορούν το χρήστη (username/password, χρήστες που μπλοκάρει, χρήστης στον οποίο γίνεται προώθηση κλήσεων, πολιτική χρέωσης που επιθυμεί, αγαπημένοι χρήστες σε περίπτωση που αυτό υπάγεται στην πολιτική χρέωσης που επιλέγει).

Αποσυνπλέκουμε τον Proxy server από το κομμάτι αυτό της επικοινωνίας με τον client και έχουμε την client εφαρμογή να επιτελεί απευθείας τέτοιου είδους συναλλαγές με τη βάση δεδομένων. Η σχεδιαστική αυτή επιλογή αφενός καθιστά τη σχεδίαση και υλοποίηση πιο εύκολη, αφού δεν απαιτεί την ανταλλαγή ειδικά δομημένων μηνυμάτων μεταξύ client και Proxy Server, αφετέρου διατηρεί την επικοινωνία client – Proxy Server ως έχει και ως είναι καθορισμένη στο SIP πρωτόκολλο.

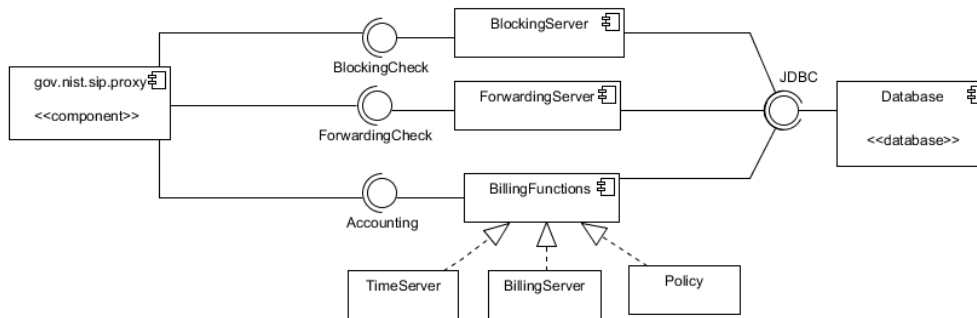
2.3 Πολιτική χρέωσης

Για την υλοποίηση της Χρέωσης Κλήσεων αποφασίστηκε η χρέωση του καλούντος ανά κλήση και όχι ανά κάποιο χρονικό διάστημα (π.χ. ανά μήνα) ανάλογα με την πολιτική χρέωσης – πακέτο στην οποία είναι συνδρομητής (ενδεικτικά: free weekends , friends & family , late at night). Παράλληλα, προσφέρεται στους χρήστες τη δυνατότητα να αλλάζουν πακέτα συνδρομής ανά πάσα στιγμή. Η σχεδιαστική αυτή επιλογή απλοποιεί τη διαδικασία της χρέωσης καθώς δεν χρειάζεται να κρατούνται πληροφορίες στη Βάση Δεδομένων για την κατάσταση χρέωσης του κάθε τηλεφωνήματος καθώς και την πολιτική χρέωσης του καλούντος τη χρονική στιγμή που πραγματοποιήθηκε το τηλεφώνημα.

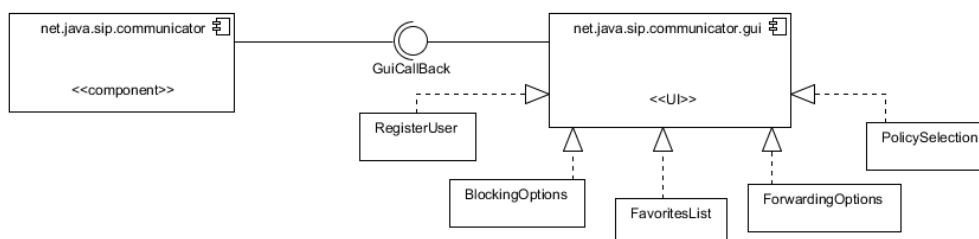
3 Αρχιτεκτονική

3.1 Ψηφιακά Διαγράμματα

Στη σχεδίασή μας συμπεριλάβαμε δύο ψηφιακά διαγράμματα, ένα για την πλευρά του Proxy(server side) και ένα για την πλευρά του Sip Communicator (client side). Παρουσιάζουμε πρώτα το διάγραμμα για τον Proxy:



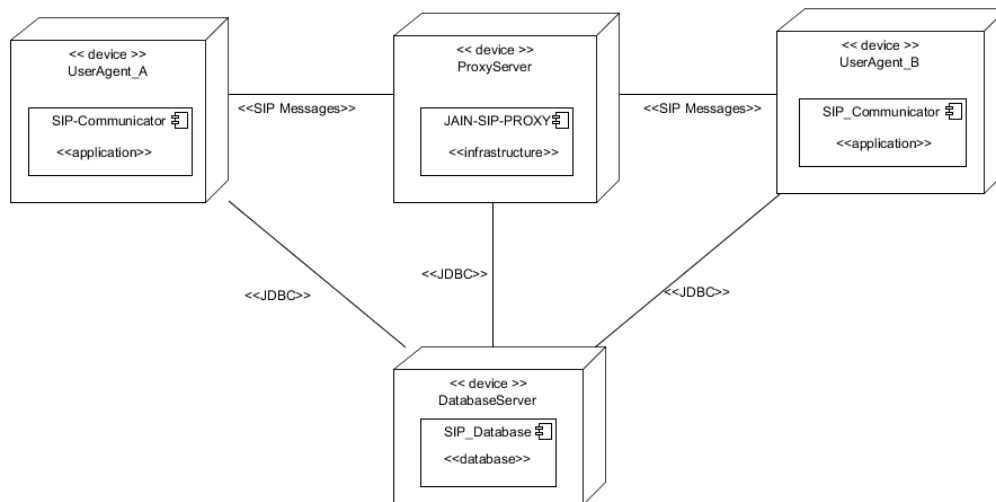
Σε αυτό το διάγραμμα επισημαίνουμε ότι χρησιμοποιούμε την αρχιτεκτονική pipe and filter. Συγκεκριμένα, ο Proxy προωθεί δεδομένα (push) στους BlockingServer, ForwardingServer και BillingFunctions αυτοί κάνουν κάποιου είδους επεξεργασία και επιστρέφουν κατάλληλα μηνύματα στον Proxy. Στη συνέχεια παρουσιάζουμε το διάγραμμα από την πλευρά του Sip Communicator.



Ένα ακόμη αρχιτεκτονικό στυλ που χρησιμοποιούμε είναι αυτό του Information Hiding. Ο Sip Communicator εκτελεί απλώς ένα invite και στη συνέχεια ο Proxy εκτελεί αρκετές λειτουργίες (έλεγχο για φραγή κλήσης, προώθηση κλήσης) και κατόπιν συνεχίζει με την εξυπηρέτηση της αίτησης του χρήστη. Αυτό γίνεται για να είναι απλός ο κώδικας πελάτη και όλη η λειτουργία μεταφέρεται στην πλευρά του server.

3.2 Παραταξιακά Διαγράμματα

Τα παραταξιακά διαγράμματα αναφέρονται στη γενική σχεδιαστική δομή του συστήματος. Παρουσιάζουν και μοντελοποιούν τη σχετική διάταξη και παράταξη των υπομονάδων κατά την εκτέλεση και λειτουργία του συστήματος. Στη δική μας περίπτωση κατασκευάσαμε το παρακάτω διάγραμμα:

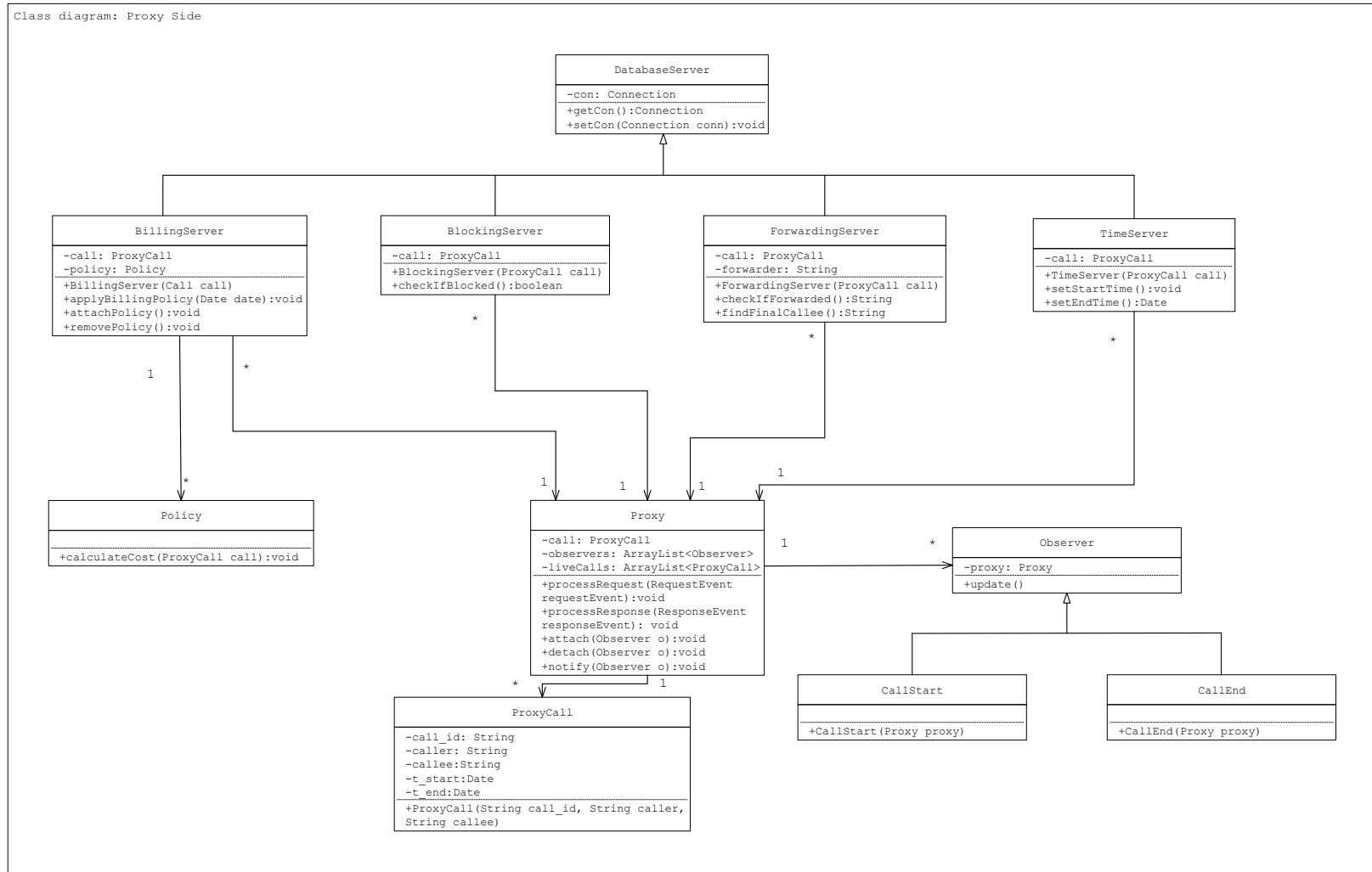


Από το παραπάνω διάγραμμα είναι φανερό ότι χρησιμοποιούμε το μοντέλο client-server. Οι δύο χρήστες A και B επικοινωνούν μέσω του Proxy αρχικά μέχρι να γίνει η εγκατάσταση της σύνδεσης μεταξύ τους, ενώ από τη στιγμή που γίνει η σύνδεση το μοντέλο μετατρέπεται ουσιαστικά σε peer-to-peer.

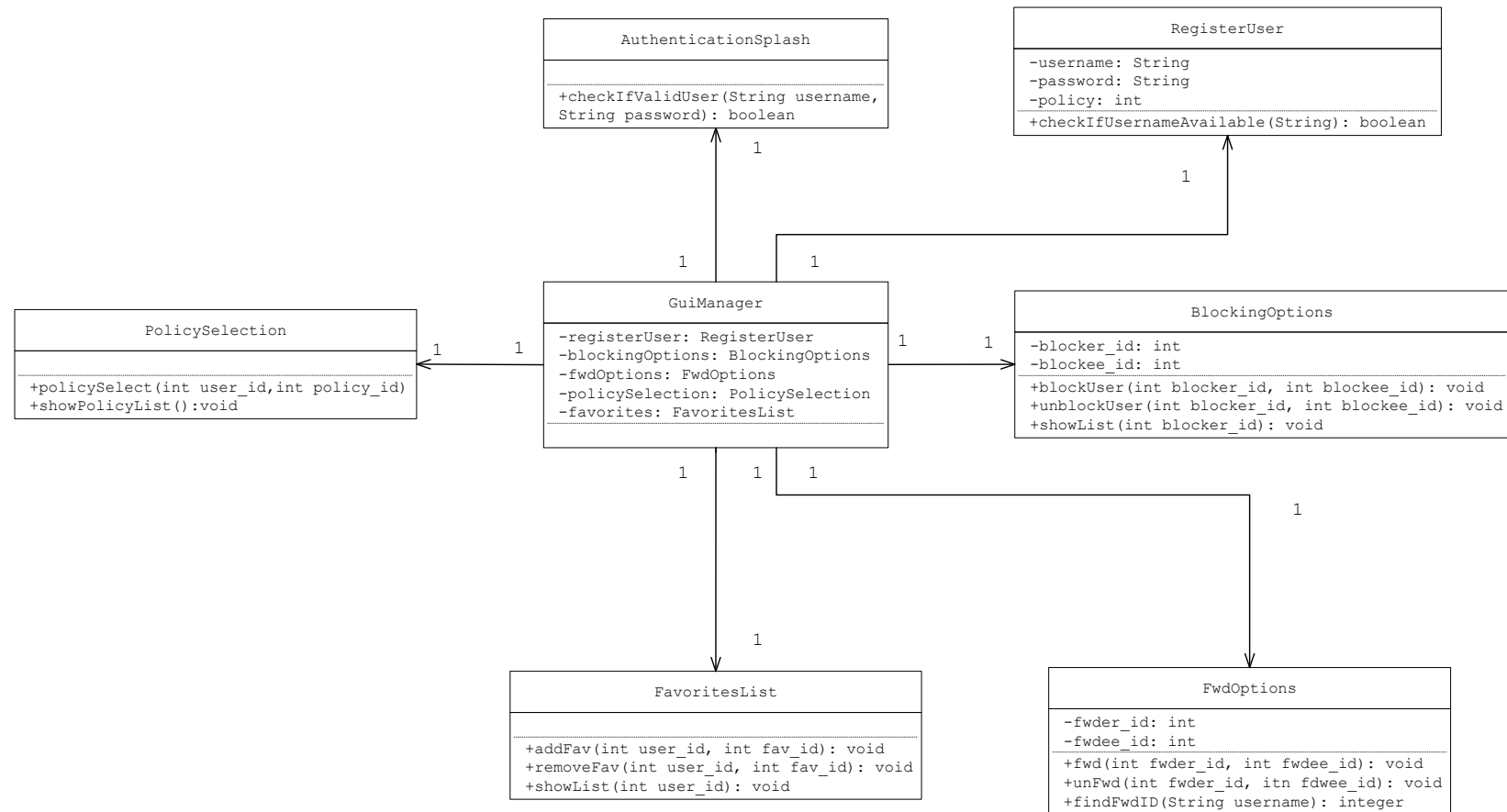
4 Λεπτομερή διαγράμματα κλάσεων

4.1 Διαγράμματα κλάσεων σε UML

Παρουσιάζουμε τα λεπτομερή διαγράμματα των κλάσεων που κατασκευάσαμε/ τροποποιήσαμε για να υλοποιήσουμε τις ζητούμενες λειτουργίες.



Class diagram: Communicator Side



4.2 Λεπτομέρειες μεθόδων

4.2.1 SipProxy

4.2.1.1 ProxyCall

Τα αντικείμενα αυτής της κλήσης θα έχουν τις απαραίτητες πληροφορίες για την κοστολόγηση της κλήσης.

```
public ProxyCall(int caller_id, int callee_id, Date  
t_start, Date t_end)
```

```
Αρχικοποίησε το call_id      //το id της κλήσης  
Αρχικοποίησε το caller        //ο καλών  
Αρχικοποίησε το callee        //ο καλούμενος  
Αρχικοποίησε το t_start       //χρονική στιγμή έναρξης κλήσης  
Αρχικοποίησε το t_end        //χρονική στιγμή λήξης κλήσης
```

4.2.1.2 Database Server

Υπερκλάση που πραγματοποιεί σύνδεση με τη βάση δεδομένων.

Δημιουργεί ένα καινούριο αντικείμενο τύπου Connection ώστε να είναι δυνατή η σύνδεση με τη βάση.

4.2.1.3 BlockingServer

Επεκτείνει την DatabaseServer.

```
public BlockingServer(ProxyCall call)
```

```
Αρχικοποίησε το call
```

```
public boolean checkIfBlocked()
```

Έλεγχξε στη βάση δεδομένων αν υπάρχει εγγραφή για την οποία ο callee μπλοκάρει τον caller.

```
Αν ναι
```

```
    επίστρεψε true
```

```
Αλλιώς
```

```
    επίστρεψε false
```

4.2.1.4 ForwardingServer

Επεκτείνει την DatabaseServer.

```
public ForwardingServer(ProxyCall call)
```

```
Αρχικοποίησε το call
```

```
Αρχικοποίησε το forwarder με την τιμή του callee
```

```
public int checkIfForwarded()
```

Βρες στη βάση δεδομένων σε ποιον προωθεί ο callee

```
Αν επιστραφεί null
```

```
    Επίστρεψε ""
```

```
Αλλιώς
```

```
    Επίστρεψε το username αυτού στον οποίο προωθείται η κλήση
```

```
public void findFinalCallee()
```

```

Φτιάξε ένα καινούριο HashSet fList
toUser = checkIfForwarded()
fList.add(forwarder)
forwarder = toUser
while(toUser != "")
    Θέσε το πεδίο callee του αντικειμένου call ίσο με την τιμή
    επιστροφής της checkIfForwarded
    Δημιούργησε νέο BlockingServer με παράμετρο call
    Αν checkIfBlocked()
        επίστρεψε BUSY
    Αν περιέχεται στην fList ο toUser τότε ανιχνεύτηκε κύκλος και
    επομένως επίστρεψε BUSY

```

4.2.1.5 TimeServer

Επεκτείνει την DatabaseServer.

```
public TimeServer(ProxyCall call)
```

Αρχικοποίησε το call

```
public void setStartTime()
```

Εισήγαγε στη βάση δεδομένων νέα εγγραφή στον πίνακα κλήσεων με την τρέχουσα ώρα και ημερομηνία στα αντίστοιχα πεδία

```
public Date setEndTime()
```

Επίστρεψε την τρέχουσα ημερομηνία και ώρα ως τέλος της τρέχουσας κλήσης

4.2.1.6 Policy

Υλοποιεί το σχεδιαστικό μοτίβο συμπεριφοράς **Strategy** ώστε να υπολογίζεται το κόστος της κλήσης ανάλογα με την εκάστοτε πολιτική. Εξειδικεύεται σε υποκλάσεις, μία για κάθε πολιτική χρέωσης.

```
public void calculateCost(ProxyCall call)
```

4.2.1.7 Billing Server

Επεκτείνει την DatabaseServer.

```
public BillingServer(ProxyCall call)
```

Αρχικοποίησε το call

Βρες από τη βάση την πολιτική χρέωσης που έχει επιλέξει ο καλών, δημιούργησε ένα νέο αντικείμενο της αντίστοιχης κλάσης και κάνε attach αυτό το αντικείμενο στο πεδίο policy.

```
public void applyBillingPolicy(Date date)
```

Βρες το τρέχον ποσό για τον caller στη βάση δεδομένων

Κάλυψε την calculateCost για το policy

Ενημέρωσε τη βάση με τα στοιχεία ώρας τερματισμού και κόστους της κλήσης

```
public void attachPolicy(Policy policy)
```

Έλεγχε αν το πεδίο BillingServer.policy έχει αρχικοποιηθεί

```

Αν BillingServer .policy είναι null
    Θέσε BillingServer.policy = policy
αλλιώς
    Εκτέλεσε removePolicy()
    Θέσε BillingServer.policy = policy

public void removePolicy()
    Θέσε BillingServer.policy = null

```

4.2.1.8 Observer

Η κλάση αυτή υλοποιεί το σχεδιαστικό μοτίβο συμπεριφοράς **Observer**. Προσάπτεται στην κλάση Proxy και χρησιμοποιείται ώστε να ενημερώνεται κατάλληλα η βάση δεδομένων όταν ληφθεί μήνυμα επιβεβαίωσης/αποχώρησης.

4.2.1.9 CallStart

Επεκτείνει τον Observer.

```

public void update()
    Φτιάξε νέο αντικείμενο TimeServer
    Κάλεσε την setStartTime

```

4.2.1.10 CallEnd

Επεκτείνει τον Observer.

```

public void update()
    Φτιάξε νέο αντικείμενο TimeServer
    Κάλεσε την setEndTime()
    Φτιάξε νέο αντικείμενο BillingServer
    Κάλεσε την applyBillingPolicy(date)

```

4.2.1.11 Proxy

Η κλάση αυτή και οι δύο μέθοδοι τροποποιήθηκαν ώστε να καλούνται οι νέες κλάσεις που κατασκευάστηκαν για την πραγματοποίηση των πρόσθετων λειτουργικοτήτων. Διατηρεί επιπλέον λίστα με δύο observers ώστε ανάλογα με το μήνυμα επιβεβαίωσης ή αποχώρησης που θα ληφθεί να καλείται η αντίστοιχη update του observer. Επίσης, διατηρεί λίστα liveCalls με τις κλήσεις σε εξέλιξη ώστε να μπορεί να βρει τα στοιχεία της κάθε κλήσης

```

public void processRequest(RequestEvent requestEvent)
    Δημιούργησε νέο αντικείμενο call τύπου ProxyCall
    Δημιούργησε νέο BlockingServer με παράμετρο call
    Αν checkIfBlocked
        επίστρεψε BUSY
    αλλιώς
        Δημιούργησε νέο αντικείμενο Forwarding με παράμετρο call
        Κάλεσε την findFinalCallee

```

Τροποποίησε το ToHeader του request ώστε το αίτημα να προωθηθεί στον κατάλληλο callee

```
public void processResponse(ResponseEvent responseEvent)
```

Βρες από τα headers του response τον caller και τον callee

Αν το response είναι τύπου INVITE και ο κωδικός απάντησης είναι 200 OK τότε

Φτιάξε ένα αντικείμενο τύπου ProxyCall

Βάλε στη λίστα liveCalls την κλήση

Κάλεσε τον observer που περιμένει έναρξη κλήσης

Αν το response είναι τύπου BYE τότε

Βρες την κλήση στη λίστα liveCalls

Κάλεσε τον observer που περιμένει τέλος κλήσης

Αφαίρεσε την κλήση από τη λίστα liveCalls

```
public void attach(Observer observer)
```

Βάλε το αντικείμενο observer στο τέλος της λίστας observers

```
public void detach(Observer observer)
```

Βγάλε το αντικείμενο observer από τη λίστα

```
public void notify(Observer observer)
```

Κάλεσε την update για το αντικείμενο observer

4.2.2 SIP Communicator

Όπως έχει ήδη αναφερθεί η επιπλέον λειτουργικότητα που απαιτείται για την επέκταση του Πρωτοκόλλου SIP στην πλευρά του User Agent περιορίζεται μόνο στη σχεδίαση κατάλληλων διεπαφών έτσι ώστε προσδιοριστούν οι Λίστες Φραγής, Προώθησης και Αγαπημένων Χρηστών και να αποθηκευτούν στη Βάση Δεδομένων. Επομένως, επεκτείναμε κατάλληλα το net.java.sip.communicator.gui , τροποποιώντας την κλάση GuiManager.java και προσθέτοντας τις απαραίτητες κλάσεις.

4.2.2.1 GuiManager

Προστέθηκαν τα κατάλληλα buttons/tabs για την Είσοδο , Εγγραφή , Προσδιορισμό χρηστών προς φραγή εισερχόμενων κλήσεων, προώθηση και Επιλογή Πακέτου Χρέωσης.

4.2.2.2 AuthenticationSplash

Εμπλουτίζουμε τη κλάση με τη μέθοδο CheckIfValidUser και αντίστοιχο «Login» button.

```
public boolean CheckIfValidUser(String username,String password)
```

Εκτελείται όταν ο χρήστης εισάγει τα στοιχεία (username,password) και πατήσει το «Login» button.

Έλεγχξε αν το username και password που εισήγαγε ο χρήστης βρίσκονται στη Βάση Δεδομένων

Αν username και password είναι έγκυρα
επίστρεψε true

αλλιώς
επίστρεψε false

4.2.2.3 RegisterUser

Η κλάση RegisterUser εμφανίζει ένα παράθυρο JFrame για την πρώτη εγγραφή του χρήστη στην υπηρεσία . Μόλις ο χρήστης συμπληρώσει τα στοιχεία του και πατήσει το κουμπί «Submit» εκτελείται το actionPerformed όπου συμβαίνουν τα εξής :

```
Αν checkIfUsernameAvailable(username)
    καταχώρησε τα στοιχεία χρήστη στον πίνακα Users της ΒΔ
αλλιώς
    εμφάνισε το μήνυμα «Το username χρησιμοποιείται ήδη από άλλο
    χρήστη»
```

```
public boolean checkIfUsernameAvailable (String username)
```

Έλεγχξε αν το username που επέλεξε ο χρήστης υπάρχει στη ΒΔ

Αν το username υπάρχει

επίστρεψε false

αλλιώς

επίστρεψε true

4.2.2.4 BlockingActions

Η κλάση BlockingActions εμφανίζει ένα παράθυρο JFrame για τον προσδιορισμό των χρηστών για φραγή εισερχόμενων κλήσεων (Block/Unblock).

```
public void showList(int blocker_id)
```

Εμφανίζει την λίστα με τους υπάρχοντες χρήστες της υπηρεσίας και το αντίστοιχο button για τη διαχείριση της φραγής εισερχομένων κλήσεων από τον εκάστοτε χρήστη. Δηλαδή :

Για κάθε χρήστη

Αν βρίσκεται στην blockedList του συγκεκριμένου χρήστη

εμφάνισε «unblock» button

αλλιώς

εμφάνισε «block» button

Με το πάτημα του αντίστοιχου button εκτελείται το κατάλληλο actionPerformed και πραγματοποιείται η αντίστοιχη μέθοδος :

```
public void blockUser(int blocker_id, int blockee_id)
```

Εκτελείται με το πάτημα του «block» button και περιλαμβάνει τα εξής βήματα :

Αποθήκευσε την επιλογή του χρήστη εκτελώντας το κατάλληλο sql insert statement στον πίνακα BlockingList

Εμφάνισε κατάλληλο μήνυμα

```
public void unblockUser(int blocker_id, int blockee_id)
```

Εκτελείται με το πάτημα του «unblock» button και περιλαμβάνει τα εξής βήματα :

Αποθήκευσε την επιλογή του χρήστη εκτελώντας το κατάλληλο sql delete statement στον πίνακα blockingList

Εμφάνισε κατάλληλο μήνυμα

4.2.2.5 ForwardingActions

Η κλάση ForwardingActions εμφανίζει ένα παράθυρο JFrame για τον προσδιορισμό της προώθησης κλήσεων προς ένα άλλο χρήστη της υπηρεσία (Forward/UnForward)

Αν ο χρήστης έχει ενεργοποιημένη την προώθηση κλήσεων (υπάρχει εγγραφή στη Βάση Δεδομένων)

εμφάνισε «unforward» button

αλλιώς

εμφάνισε μήνυμα «Εισάγετε username χρήστη για προώθηση κλήσεων» text field για συμπλήρωση του username και button «forward»

Με το πάτημα του αντίστοιχου button εκτελείται το κατάλληλο actionperformed και πραγματοποιείται η αντίστοιχη μέθοδος :

public void forwardUser(int fwder_id, int fwdee_id)

Εκτελείται με το πάτημα του «block» button και περιλαμβάνει τα εξής βήματα :

Μάθε τα ids των εμπλεκόμενων χρηστών (forwarder, forwarder) από αντίστοιχο session χρησιμοποιώντας την findForwarderID

Αποθήκευσε την επιλογή του χρήστη εκτελώντας το κατάλληλο sql insert statement στο πίνακα ForwardingList

Εμφάνισε κατάλληλο μήνυμα

public void unforwardUser(int fwder_id, int fwdee_id)

Εκτελείται με το πάτημα του «unblock» button και περιλαμβάνει τα εξής βήματα :

Αποθήκευσε την επιλογή του χρήστη εκτελώντας το κατάλληλο sql delete statement

Εμφάνισε κατάλληλο μήνυμα

public int findForwarderID(String username)

Επιστρέφει το id του forwarder με βάση το username.

Αν το username δεν υπάρχει στη Βάση Δεδομένων

επίστρεψε -1

αλλιώς

επίστρεψε user_id

4.2.2.6 PolicySelection

Η κλάση PolicySelection εμφανίζει ένα παράθυρο JFrame για τον προσδιορισμό της Πολιτικής Χρέωσης.

public void showPolicyList()

Εμφανίζει την λίστα με τις διαθέσιμες πολιτικές χρέωσης (πακέτα) και κατάλληλα buttons «Subscribe».

Με το πάτημα του button της αντίστοιχης πολιτικής χρέωσης εκτελείται το actionperformed και πραγματοποιείται η μέθοδος :

public void PolicySelect(int user_id,int policy_id)

Αποθήκευσε την επιλογή του χρήστη εκτελώντας το κατάλληλο sql insert/update statement στο πίνακα Users

Εμφάνισε κατάλληλο μήνυμα

4.2.2.7 Favorites

Η κλάση Favorites εμφανίζει ένα παράθυρο JFrame για τον προσδιορισμό των Αγαπημένων Χρηστών του κάθε χρήστη(Add/Remove)

```
public void showList()
```

Εμφανίζει την λίστα με τους υπάρχοντες χρήστες της υπηρεσίας και το αντίστοιχο button για τη διαχείριση των Αγαπημένων Χρηστών. Δηλαδή :

Για κάθε χρήστη

Αν βρίσκεται στην FavoritesList του συγκεκριμένου χρήστη

εμφάνισε «Remove» button

αλλιώς

εμφάνισε «Add» button

Με το πάτημα του αντίστοιχου button εκτελείται το κατάλληλο actionPerformed και πραγματοποιείται η αντίστοιχη μέθοδος :

```
public void AddFavorites(int user_id, int fav_id)
```

Εκτελείται με το πάτημα του «add» button και περιλαμβάνει τα εξής βήματα :

Αποθήκευσε την επιλογή του χρήστη εκτελώντας το κατάλληλο sql insert statement στον πίνακα FavoritesList

Εμφάνισε κατάλληλο μήνυμα

```
public void RemoveFavorites(int user_id, int fav_id)
```

Εκτελείται με το πάτημα του «remove» button και περιλαμβάνει τα εξής βήματα :

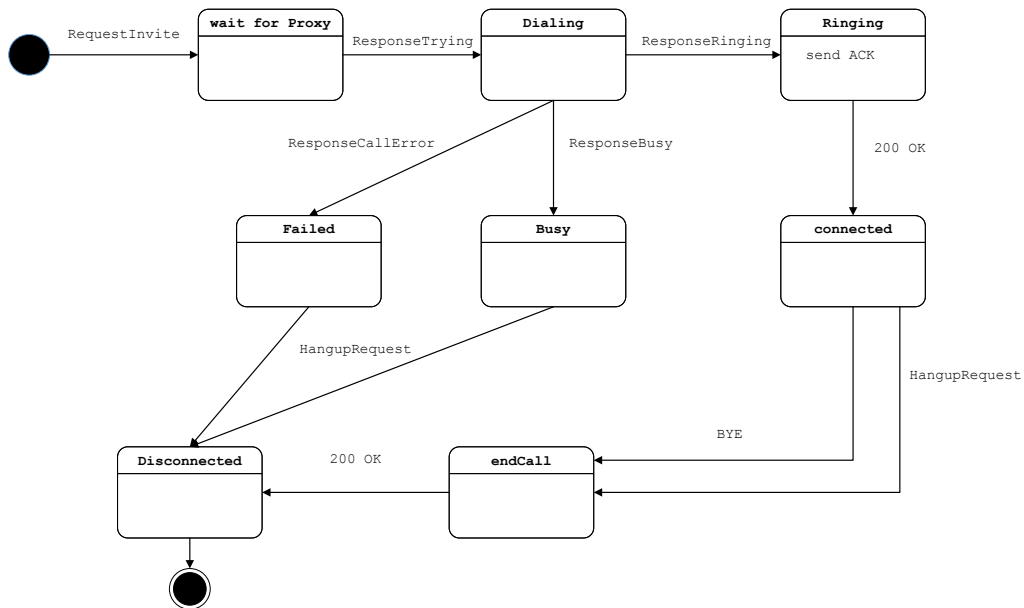
Αποθήκευσε την επιλογή του χρήστη εκτελώντας το κατάλληλο sql delete statement στον πίνακα FavoritesList

Εμφάνισε κατάλληλο μήνυμα

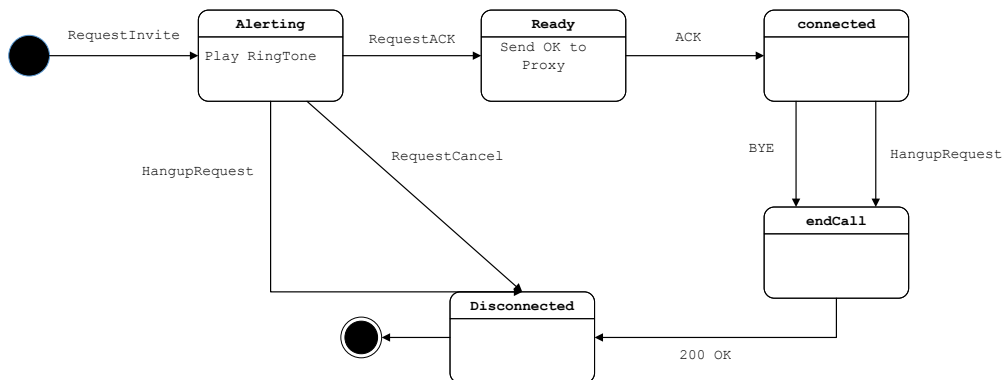
5 Διάγραμμα Κατάστασης

Παρακάτω παρατίθεται το διάγραμμα κατάστασης της κλάσης `net.java.sip.communicator.sip.Call`. Στο διάγραμμα αυτό παρουσιάζονται οι καταστάσεις στις οποίες μπορεί να βρεθεί η κλάση αυτή καθώς και τα μηνύματα που σχετίζονται με τη μετάβαση από τη μια κατάσταση στην άλλη. Επειδή παρατηρείται μια σχετική ασυμμετρία στην περίπτωση του caller και του callee δείχνουμε δύο διαγράμματα ένα για κάθε περίπτωση.

State machine for class `call.java` (caller side)



State machine for class `call.java` (callee side)



6 Ανοιχτά Ζητήματα

6.1 Ανεξέλεγκτη χρέωση καλούντος

Στην περίπτωση που ο Proxy δεν αντιλαμβάνεται σωστά το τέλος της συνδιάλεξης (πχ λόγω συντριβής του Client) θα πρέπει να γίνεται κάποιος έλεγχος, ώστε ο καλών να μην χρεώνεται παραπάνω από όσο πρέπει.

6.2 Ανίχνευση κύκλων στην προώθηση κλήσης

Δεν έχει αποφασιστεί ακόμα ο τρόπος που θα ελέγχεται η ύπαρξη κύκλων στη λειτουργία προώθησης κλήσης και ο χειρισμός προβληματικών καταστάσεων όπως να επανερχόμαστε στον αρχικό χρήστη προς τον οποίο έγινε η κλήση με διαδοχικές προωθήσεις.

6.3 Σχεσιακό Μοντέλο Βάσης

Το ακριβές σχεσιακό μοντέλο βάσης δεν έχει καθοριστεί πλήρως (πεδία πινάκων, κλειδιά, indexes κλπ).

6.4 Πολιτικές Χρέωσης

Τελευταίο ζήτημα είναι τα προγράμματα χρέωσης που θα συμπεριλάβουμε και θα διαθέτουμε στους χρήστες για να επιλέξουν.