

Επέκταση Πρωτοκόλλου SIP

Software Design Document

| | |
|----------------------------|--|
| Έκδοση: | 1.0 |
| Ημερομηνία Εκτύπωσης: | 21/12/2016 |
| Ημερομηνία Κυκλοφορίας: | 21/12/2016 |
| Κατάσταση Έκδοσης: | Αρχική |
| Κατάσταση Έγκρισης: | Πρόχειρο |
| Εγκρίθηκε από: | - |
| Προετοιμάστηκε από: | Αργυρίου Στέφανος Παναγιώτης Βάρη Χρυσούλα Κάτσιος Αθανάσιος Κονιδάρης Φίλιππος |
| Επιθεωρήθηκε από: | - |
| Όνομα μονοπατιού: | - |
| Όνομα αρχείου | SDD_Cover_Contents.odtSDD_Docu ment |

ΠΕΡΙΕΧΟΜΕΝΑ

- 1. Εισαγωγή**
- 2. Θεμελιώδεις Σχεδιαστικές Αποφάσεις**
 - 2.1 Δημιουργία Βάσης Δεδομένων
 - 2.2 Εγγραφή και σύνδεση χρήστη
 - 2.3 Προώθηση εισερχόμενων κλήσεων
 - 2.4 Περιορισμός εισερχόμενων κλήσεων
 - 2.5 Χρέωση εξερχόμενων κλήσεων
 - 2.6 Επιπρόσθετες τροποποιήσεις
- 3. Αρχιτεκτονική**
- 4. Λεπτομερή διαγράμματα κλάσεων**
 - 4.1 Διαγράμματα Κλάσης UML
 - 4.1.1 SIP Communicator
 - 4.1.2 SIP Proxy
 - 4.2 Λεπτομέρειες Μεθόδων
 - 4.2.1 Proxy Server
 - 4.2.2 Blocking Server
 - 4.2.3 Billing Server
 - 4.2.4 Forwarding Server
 - 4.2.5 Timer Server
 - 4.2.6 Registrar Server
 - 4.2.7 Database Server
 - 4.2.8 Proxy Extension
 - 4.2.9 Socket Server
 - 4.2.10 Worker
 - 4.2.11 BlockUI
 - 4.2.12 ForwardingUI
 - 4.2.13 BillingUI
 - 4.2.14 NoEditable Table Model
 - 4.2.15 GUIManager
 - 4.2.16 Phone Frame
 - 4.2.17 Authentication Splash
 - 4.2.18 SIP Communicator
- 5. Διαγράμματα Κατάστασης**
- 6. Ανοιχτά Ζητήματα**
 - 6.1 Ασφάλεια Συστήματος
 - 6.2 Εύρεση Κύκλων στην Προώθηση Κλήσεων
 - 6.3 Γραφικό Περιβάλλον
 - 6.4 Billing Σύστημα

1 Εισαγωγή

Επισκόπηση:

Σκοπός της παρούσας εργασίας είναι η επέκταση των λειτουργιών των εφαρμογών JAIN-SIP-Presence-Proxy και SIP Communicator, οι οποίες επιτρέπουν την επικοινωνία δύο εγγεγραμμένων χρηστών του συστήματος μέσω του πρωτοκόλλου SIP.

Συγκεκριμένα οι λειτουργίες, οι οποίες υλοποιούνται είναι οι εξής:

- Εγγραφή χρήστη** στο σύστημα με την εισαγωγή κατάλληλων προσωπικών δεδομένων (username, password).

- Φραγή κλήσεων (call blocking)**: Με αυτή τη λειτουργία ένας χρήστης έχει τη δυνατότητα να απορρίπτει αυτόματα κλήσεις, οι οποίες προέρχονται από άλλους χρήστες του συστήματος (περιορισμένοι χρήστες).

- Προώθηση κλήσεων (call forwarding)**: Ο χρήστης έχει τη δυνατότητα να προωθεί τις εισερχόμενες κλήσεις σε κάποιον τρίτο χρήστη του συστήματος (τον οποίο έχει ορίσει ρητά ως υπεύθυνο αποδοχής των κλήσεων).

- Χρέωση κλήσεων (call billing)**: Η χρέωση των κλήσεων γίνεται μέσω κάποιας τιμολογιακής πολιτικής.

Εκτός από τις παραπάνω λειτουργίες, εξετάζονται και επιλύονται ορισμένα παθολογικά σενάρια όπως η αποτυχία ή βίαιη διακοπή σύνδεσης.

Στο παρών έγγραφο παρουσιάζονται τα εξής θέματα:

- Κεφάλαιο 2**: Περιέχει τις σχεδιαστικές αποφάσεις της υλοποίησης της συγκεκριμένης εφαρμογής.

- Κεφάλαιο 3**: Παρουσιάζει τα ψηφιδωτά και παραταξιακά διαγράμματα της δομής του συστήματος και των υπομονάδων του.

- Κεφάλαιο 4**: Περιέχει τις κλάσεις, οι οποίες υλοποιούνται στη συγκεκριμένη εργασία και τα αντίστοιχα UML διαγράμματα κλάσεων.

- Κεφάλαιο 5**: Παραθέτει τα διαγράμματα κατάστασης του συστήματος.

- Κεφάλαιο 6**: Παρουσιάζει ανοιχτά σχεδιαστικά ζητήματα, τα οποία δεν έχουν οριστικοποιηθεί.

Αναφορές

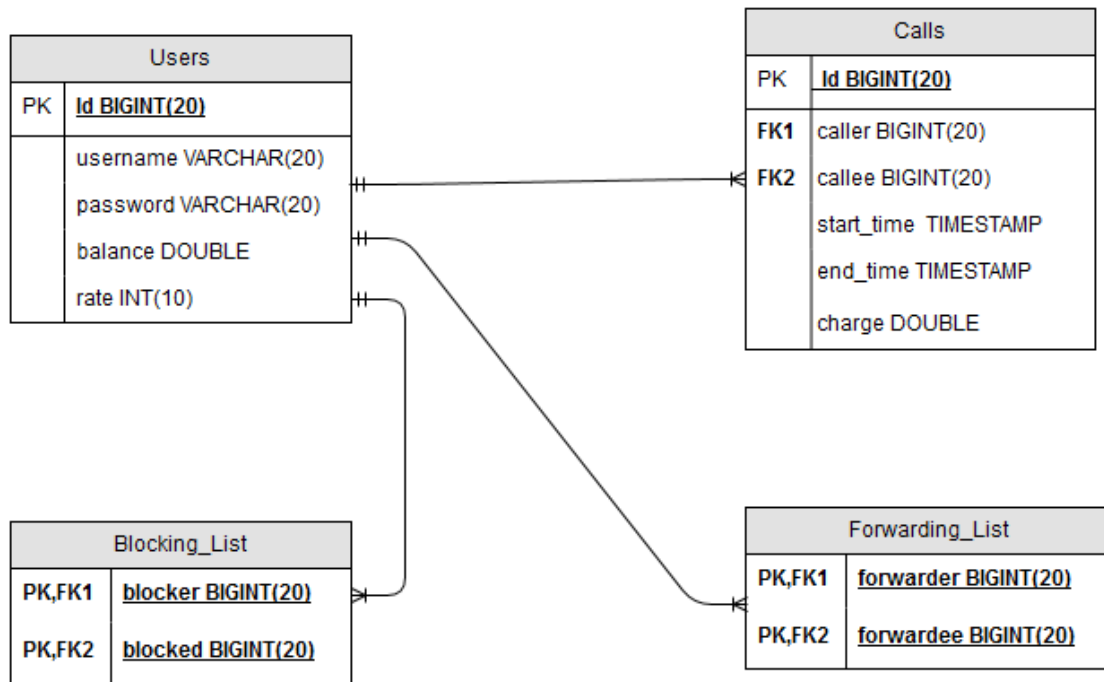
Προδιαγραφές άσκησης στην τοποθεσία:

<http://courses.softlab.ntua.gr/softeng/Projects/Project-SIP/Instructions/>

2 Θεμελιώδεις Σχεδιαστικές Αποφάσεις

2.1 Δημιουργία Βάσης Δεδομένων:

Για την αποθήκευση των δεδομένων, τα οποία απαιτούνται από τις επιπρόσθετες λειτουργικότητες του συστήματος, χρησιμοποιείται μια βάση δεδομένων. Το αντίστοιχο σχεσιακό μοντέλο της παρουσιάζεται παρακάτω:



Πίνακας Users:

Χρησιμοποιείται για την αποθήκευση των προσωπικών δεδομένων ενός χρήστη του συστήματος. Ενημερώνεται κατά την πρώτη εγγραφή του χρήστη. Τα στοιχεία balance και rate αναφέρονται στη συνολική χρέωση του χρήστη και στην τιμολογιακή πολιτική αντίστοιχα, ενώ τα στοιχεία username και password είναι απαραίτητα για την εγγραφή και την είσοδο του στο σύστημα της υπηρεσίας.

Πίνακας Calls:

Σε αυτόν τον πίνακα αποθηκεύονται οι απαραίτητες πληροφορίες κάθε τηλεφωνικής σύνδεσης / κλήσης που πραγματοποιείται. Τα πεδία caller και callee δηλώνουν τα id των χρηστών, οι οποίοι συμμετέχουν στην κλήση. Τα πεδία start_time και end_time αφορούν την έναρξη και λήξη της τηλεφωνικής κλήσης, ενώ το πεδίο charge αποθηκεύει τη συνολική της χρέωση.

Πίνακας Blocking_List:

Αντιπροσωπεύει τη λίστα των περιορισμένων χρηστών. Σε αυτή τη δομή αποθηκεύονται ζεύγη id (blocker, blocked), τα οποία δηλώνουν ότι ο χρήστης blocker έχει στη λίστα περιορισμένων χρηστών τον blocked.

Πίνακας Forwarding_List:

Στο συγκεκριμένο πίνακα αποθηκεύονται ζεύγη id (forwarder, forwardee), τα οποία δηλώνουν ότι ο χρήστης forwarder έχει επιλέξει να προωθεί τις εισερχόμενες κλήσεις του στον χρήστη forwardee.

2.2 Εγγραφή και Σύνδεση Χρήστη

Η εγγραφή ενός νέου χρήστη γίνεται μέσω του Registrar Server, ο οποίος ελέγχει τις υπάρχουσες εγγραφές της βάσης και ενεργεί αναλόγως. Αν τα στοιχεία username και

password δεν είναι έγκυρα ή υπάρχει ήδη εγγραφή στο σύστημα, το αίτημα του νέου χρήστη απορρίπτεται, διαφορετικά ενημερώνεται κατάλληλα η βάση δεδομένων.

Αντίστοιχα η είσοδος του χρήστη στο σύστημα γίνεται με τη μεσολάβηση του Registrar Server, ο οποίος αναλαμβάνει τον έλεγχο της εγκυρότητας των στοιχείων, τα οποία εισάγει ο χρήστης.

2.3 Προώθηση εισερχόμενων κλήσεων

Η συγκεκριμένη λειτουργικότητα υλοποιείται με την παρέμβαση του Forwarding Server, ο οποίος είναι υπεύθυνος για τον έλεγχο της λίστας προώθησης. Ανάλογα με τις πληροφορίες που υπάρχουν στη βάση δεδομένων ο Forwarding Server προωθεί ή όχι το αίτημα της κλήσης. Σε κάθε περίπτωση επιστρέφεται ο τελικός αποδέκτης της εισερχόμενης κλήσης. Ο συντονισμός της λειτουργίας του γίνεται μέσω του Proxy Server.

2.4 Περιορισμός εισερχόμενων κλήσεων

Ο αποκλεισμός χρηστών και η απόρριψη των αντίστοιχων κλήσεων επιτυγχάνεται με χρήση του Blocking Server. Αυτός είναι υπεύθυνος για τον έλεγχο του Blocking_List πίνακα της βάσης δεδομένων και την ενημέρωση του Proxy Server αν το αίτημα κλήσης απορρίπτεται ή όχι. Ο συντονισμός της λειτουργίας του Blocking Server γίνεται μέσω του Proxy Server.

2.5 Χρέωση εξερχόμενων κλήσεων

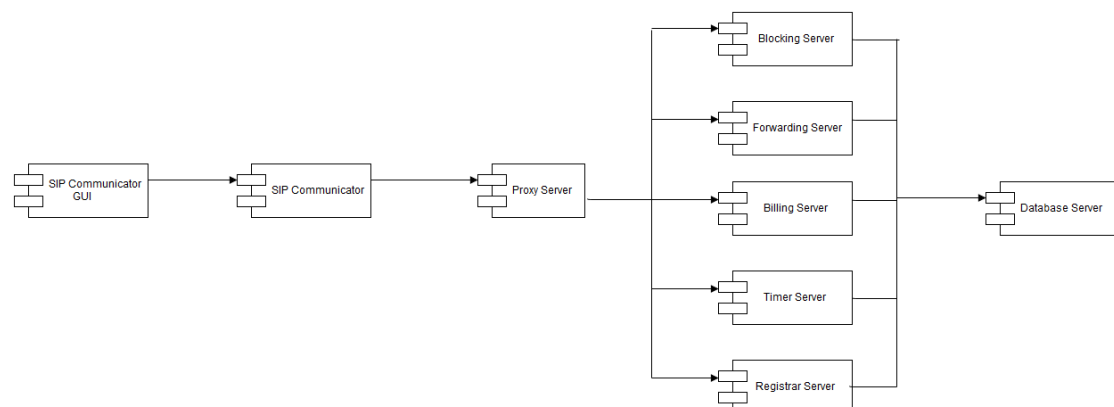
Για τη χρέωση των κλήσεων είναι υπεύθυνος ο Billing Server. Ο συγκεκριμένος server υπολογίζει το κόστος μιας κλήσης με τη βοήθεια του Timer Server, ο οποίος βρίσκει την αντίστοιχη χρονική διάρκεια. Ο Billing Server αναλαμβάνει να ενημερώσει τα στοιχεία της βάσης με το νέο υπόλοιπο του χρήστη. Ο συντονισμός της λειτουργίας των δύο servers γίνεται με την παρέμβαση του Proxy Server.

2.6 Επιπρόσθετες τροποποιήσεις

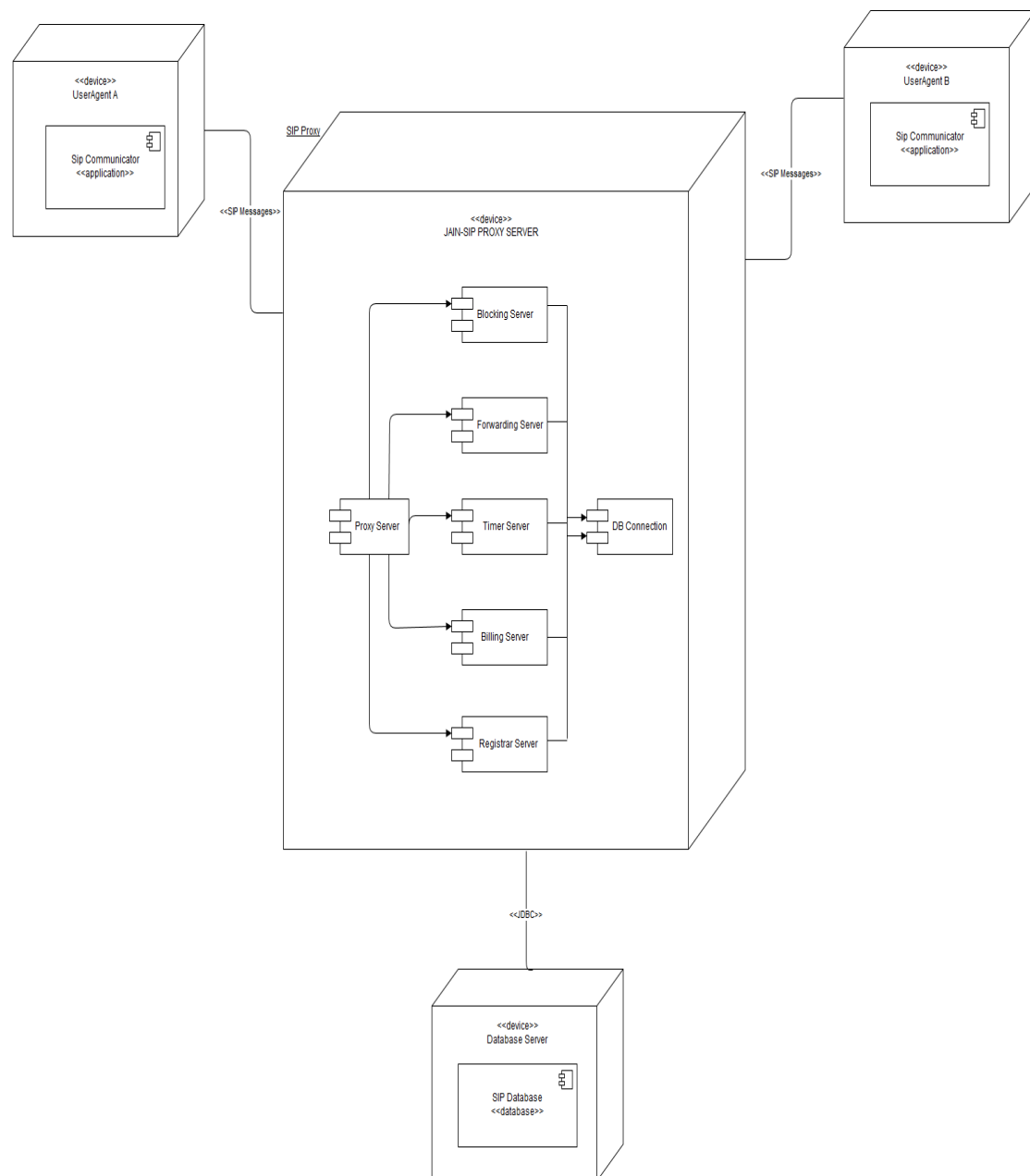
Για την υλοποίηση ορισμένων λειτουργιών και την παρουσίαση των αποτελεσμάτων στο χρήστη χρειάζονται ορισμένες τροποποιήσεις στο γραφικό περιβάλλον της εφαρμογής. Αναλυτικότερα παρουσιάζονται στην ενότητα 4 των διαγραμμάτων κλάσης.

3 Αρχιτεκτονική

Ψηφιακό Διάγραμμα (Component Diagram):



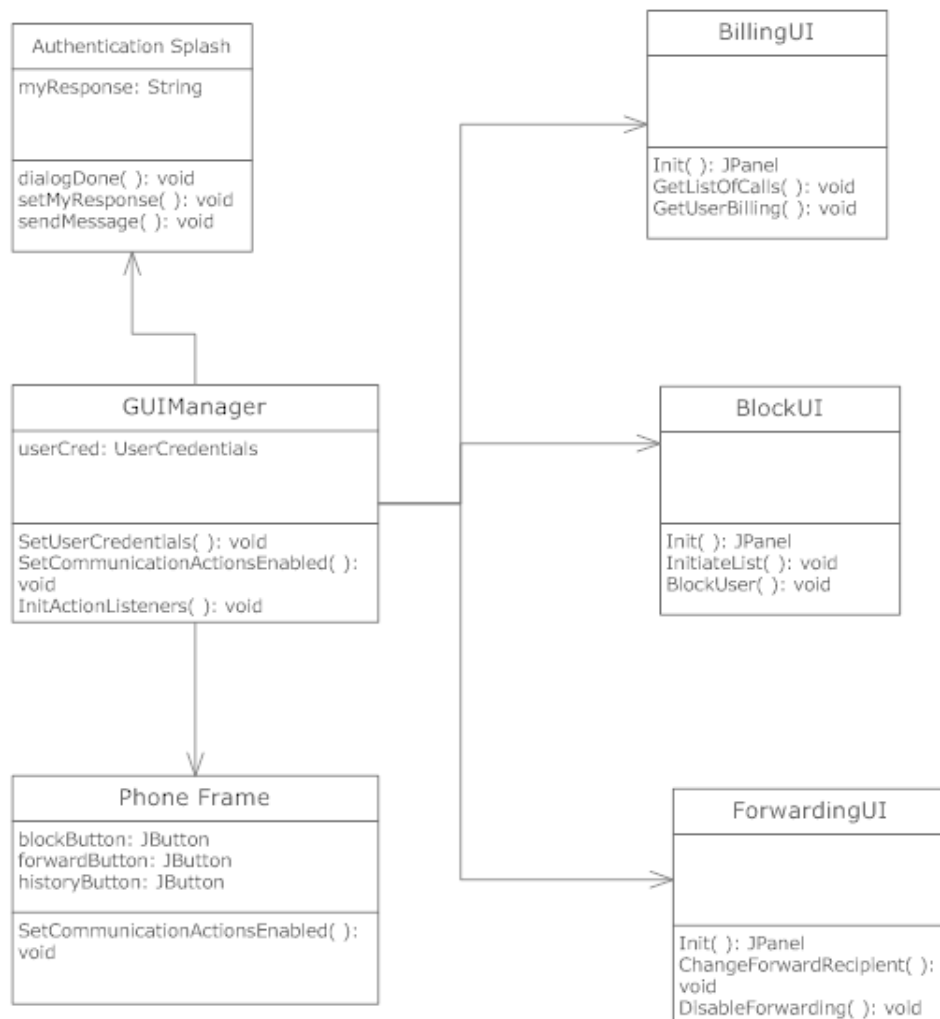
Παραταξιακό διάγραμμα (Deployment Diagram):

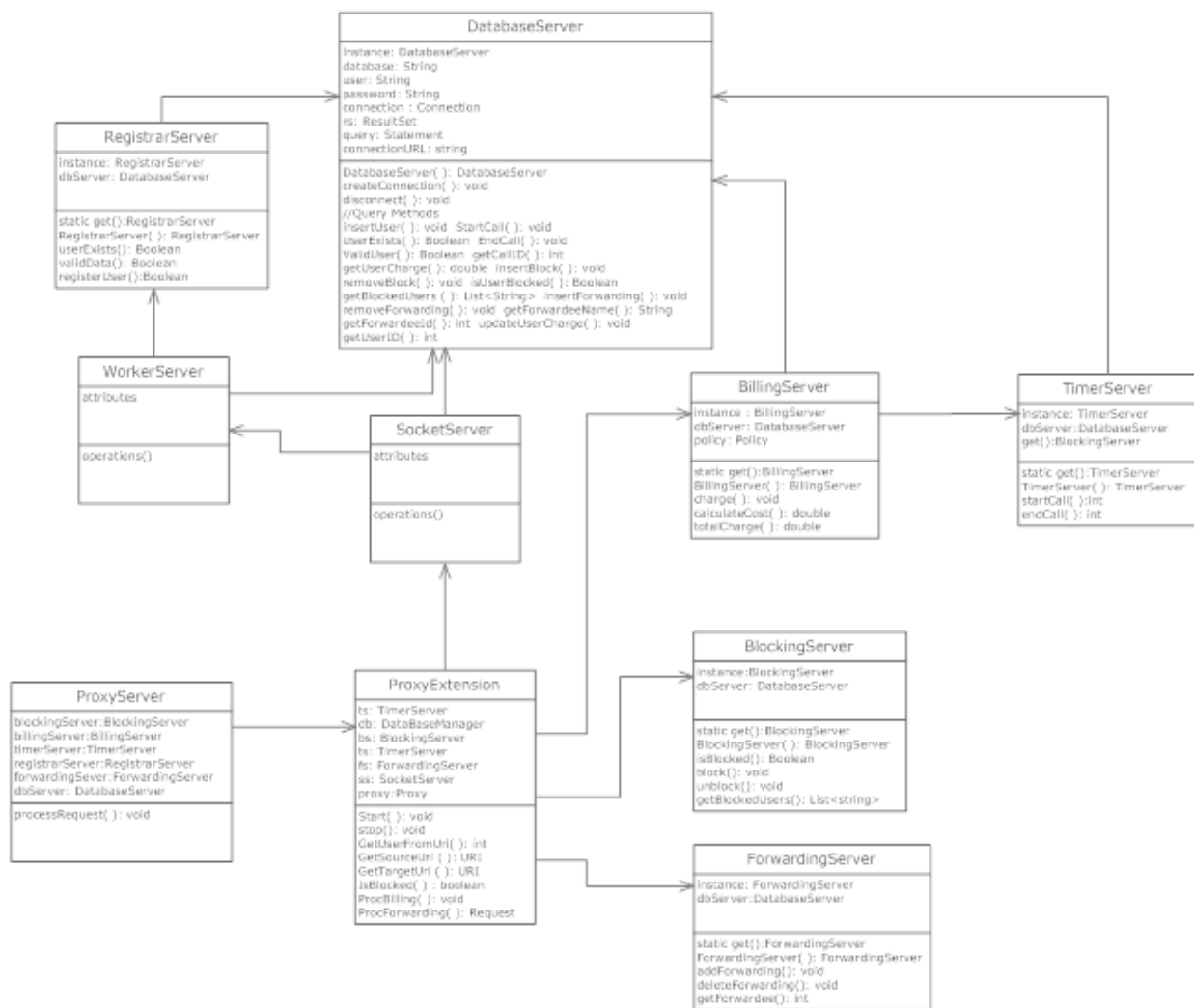


4 Λεπτομερή Διαγράμματα Κλάσεων

4.1 Διαγράμματα Κλάσης UML

4.1.1 SIP Communicator





4.1.2 SIP Proxy

4.2 Λεπτομέρειες Μεθόδων:

4.2.1 Proxy Server :

Ο proxy server είναι υπεύθυνος για τον συντονισμό των Blocking, Billing και Forwarding server όταν απαιτείται η εξυπηρέτηση κάποιου αιτήματος από τη μεριά του χρήστη. Είναι επέκταση της έτοιμης κλάσης Proxy του κώδικα του προγράμματος. Ακολουθούν πεδία και μέθοδοι, οι οποίες προστέθηκαν.

| Πεδία: | |
|--------------------------------|--|
| blockingServer: BlockingServer | Αναφέρεται στο στιγμιότυπο της κλάσης BlockingServer |
| billingServer: BillingServer | Αναφέρεται στο στιγμιότυπο της |

| | |
|-----------------------------------|--|
| | κλάσης BillingServer |
| timerServer: TimerServer | Αναφέρεται στο στιγμιότυπο της κλάσης TimerServer |
| registrarServer: RegistrarServer | Αναφέρεται στο στιγμιότυπο της κλάσης RegistrarServer |
| forwardingSever: ForwardingServer | Αναφέρεται στο στιγμιότυπο της κλάσης ForwardingServer |
| dbServer: DatabaseServer | Αναφέρεται στο στιγμιότυπο της κλάσης DatabaseServer |
| Μέθοδοι: | |
| processRequest(): void | Ανάλογα με τον κωδικό αιτήματος, το οποίο λαμβάνει μέσω του SIP Communicator, ο Proxy Server επικοινωνεί με τους υπόλοιπους Servers για την εξυπηρέτηση του αιτήματος. |

4.2.2 Blocking Server:

Ο Blocking Server είναι υπεύθυνος για τον χειρισμό των αιτημάτων, τα οποία σχετίζονται με τον έλεγχο, την εμφάνιση και unblock των περιορισμένων χρηστών μιας επαφής.

| | |
|-----------------------------------|---|
| Πεδία: | |
| instance:BlockingServer | Αναφέρεται στο στιγμιότυπο της κλάσης BlockingServer |
| dbServer: DatabaseServer | Αναφέρεται στο στιγμιότυπο της κλάσης DatabaseServer |
| Μέθοδοι: | |
| static get():BlockingServer | Επιστρέφει το υπάρχον instance της κλάσης, και σε περίπτωση που αυτό δεν υπάρχει, το δημιουργεί (Singleton) |
| BlockingServer(): BlockingServer | Αποτελεί τον constructor της κλάσης BlockingServer. |
| isBlocked(): Boolean | Δέχεται ένα ζεύγος από id (caller, callee) και ελέγχει αν ο callee έχει προσθέσει τον caller στη λίστα των περιορισμένων. Αν ο caller είναι περιορισμένος επιστρέφει true, αλλιώς επιστρέφει false. |

| | |
|----------------------------------|---|
| block(): void | Δέχεται ένα ζεύγος id (user, blocked) και ενημερώνει τη βάση δεδομένων εισάγοντας τον χρήστη blocked στη λίστα των περιορισμένων επαφών του user. |
| unblock():void | Δέχεται ένα ζεύγος id (user, unblocked) και ενημερώνει τη βάση δεδομένων αφαιρώντας τον χρήστη unblocked από τη λίστα των περιορισμένων επαφών του χρήστη user. |
| getBlockedUsers(): List<string> | Δέχεται το id ενός user και επιστρέφει τη λίστα με τους περιορισμένους χρήστες του. |

4.2.3 Billing Server:

Ο Billing Server είναι υπεύθυνος για τον υπολογισμό των χρεώσεων ενός χρήστη και την ενημέρωση της βάσης δεδομένων μετά τον τερματισμό κάθε κλήσης.

| | |
|---------------------------------|--|
| Πεδία: | |
| instance : BillingServer | Αναφέρεται στο στιγμιότυπο της κλάσης BillingServer |
| dbServer: DatabaseServer | Αναφέρεται στο στιγμιότυπο της κλάσης DatabaseServer |
| policy: Policy | Αναφέρεται στην τιμολογιακή πολιτική, την οποία επιλέγει κάθε χρήστης. Αν ο χρήστης δεν έχει επιλέξει συγκεκριμένη, τότε το πεδίο αρχικοποιείται με την default τιμολογιακή πολιτική του συστήματος κατά την αρχικοποίηση του BillingServer. |
| Μέθοδοι: | |
| static get():BillingServer | Επιστρέφει το υπάρχον instance της κλάσης, και σε περίπτωση που αυτό δεν υπάρχει, το δημιουργεί (Singleton) |
| BillingServer(): BillingServer | Αποτελεί τον constructor της κλάσης BillingServer. |
| charge(): void | Δέχεται το id του user και το επιπλέον ποσό χρέωσης και ενημερώνει τη βάση δεδομένων με το νέο υπόλοιπο του χρήστη |

| | |
|--------------------------|---|
| | user. |
| calculateCost(): double | Δέχεται τον συνολικό χρόνο χρέωσης και το id του χρήστη user και επιστρέφει το συνολικό ποσό χρέωσης. |
| totalCharge(): double | Δέχεται το id ενός χρήστη user και επιστρέφει τη συνολική του χρέωση, ελέγχοντας τον αντίστοιχο πίνακα της βάσης δεδομένων. |

4.2.4 Forwarding Server:

Ο Forwarding Server είναι υπεύθυνος για τον έλεγχο και χειρισμό των προωθήσεων, οι οποίες είναι ενεργές από κάθε χρήστη.

| | |
|---------------------------------------|--|
| Πεδία: | |
| instance: ForwardingServer | Αναφέρεται στο στιγμιότυπο της κλάσης ForwardingServer |
| dbServer: DatabaseServer | Αναφέρεται στο στιγμιότυπο της κλάσης DatabaseServer |
| Μέθοδοι: | |
| static get():ForwardingServer | Επιστρέφει το υπάρχον instance της κλάσης, και σε περίπτωση που αυτό δεν υπάρχει, το δημιουργεί (Singleton) |
| ForwardingServer(): ForwardingServer | Αποτελεί τον constructor της κλάσης ForwardingServer. |
| addForwarding():void | Δέχεται ένα ζεύγος id (user, forwarder) και ενημερώνει τη βάση δεδομένων εισάγοντας μια νέα εγγραφή προώθησης. Η εγγραφή δηλώνει ότι ο χρήστης forwarder είναι το άτομο, στο οποίο επέλεξε ο χρήστης user να προωθούνται οι κλήσεις του. |
| deleteForwarding(): void | Δέχεται το id ενός χρήστη user και διαγράφει την εγγραφή προώθησης του από τη βάση δεδομένων (εφόσον αυτή υπάρχει). |
| getForwarder(): int | Δέχεται το id ενός χρήστη user και επιστρέφει το id του forwarder (αν υπάρχει αλλιώς |

| | |
|--|---|
| | -1), στον οποίο προωθούνται οι κλήσεις του user. Η συγκεκριμένη μέθοδος είναι υπεύθυνη για τον έλεγχο κύκλων. Στην περίπτωση που υπάρχει κύκλος επιστρέφεται κάποια αρνητική τιμή ως ένδειξη. |
|--|---|

4.2.5 Timer Server:

Ο Timer Server είναι υπεύθυνος για τον υπολογισμό της διάρκειας μιας κλήσης, προκειμένου να γίνει έπειτα υπολογισμός της χρέωσης κλήσης από τον Billing Server. Ειδοποιείται από τον Proxy Server για την έναρξη και τον τερματισμό μιας τηλεφωνικής σύνδεσης.

| | |
|-----------------------------|--|
| Πεδία: | |
| instance: TimerServer | Αναφέρεται στο στιγμιότυπο της κλάσης TimerServer |
| dbServer:DatabaseServer | Αναφέρεται στο στιγμιότυπο της κλάσης DatabaseServer |
| get():BlockingServer | Επιστρέφει το υπάρχον instance της κλάσης, και σε περίπτωση που αυτό δεν υπάρχει, το δημιουργεί (Singleton) |
| Μέθοδοι: | |
| static get():TimerServer | Επιστρέφει το υπάρχον instance της κλάσης, και σε περίπτωση που αυτό δεν υπάρχει, το δημιουργεί (Singleton) |
| TimerServer(): TimerServer | Αποτελεί τον constructor της κλάσης TimerServer. |
| startCall():int | Δέχεται ένα ζεύγος id (caller, callee) και προσθέτει μια νέα εγγραφή στον πίνακα κλήσεων της βάσης. Στη νέα εγγραφή καταγράφεται επίσης η χρονική στιγμή της έναρξης της κλήσης των χρηστών caller και callee. Επιστρέφει το id της νέας κλήσης. |
| endCall(): int | Δέχεται το id μιας κλήσης και βρίσκει τη διάρκεια της κλήσης, την οποία και επιστρέφει. Ενημερώνει κατάλληλα το πεδίο |

| | |
|--|--|
| | τερματισμού στον πίνακα κλήσεων της βάσης δεδομένων. |
|--|--|

4.2.6 Registrar Server:

Ο Registrar Server είναι υπεύθυνος για την εγγραφή νέων χρηστών στο σύστημα της υπηρεσίας και τον έλεγχο πιστοποίησης δεδομένων ήδη εγγεγραμμένων χρηστών. Για να μην επεκταθεί σε μεγάλο βαθμό η υλοποίηση του προγράμματος, αποφασίστηκε ότι ο Registrar Server θα είναι υπεύθυνος για το γενικό έλεγχο πιστοποίησης ενός χρήστη στη βάση δεδομένων. Στο έγγραφο SRS αυτή η λειτουργία αντιστοιχούσε στον Location Server.

| | |
|-------------------------------------|---|
| Πεδία: | |
| instance: RegistrarServer | Αναφέρεται στο στιγμιότυπο της κλάσης RegistrarServer. |
| dbServer: DatabaseServer | Αναφέρεται στο στιγμιότυπο της κλάσης DatabaseServer. |
| Μέθοδοι: | |
| static get():RegistrarServer | Επιστρέφει το υπάρχον instance της κλάσης, και σε περίπτωση που αυτό δεν υπάρχει, το δημιουργεί (Singleton) |
| RegistrarServer(): RegistrarServer | Αποτελεί τον constructor της κλάσης RegistrarServer. |
| userExists(): Boolean | Δέχεται το id ενός χρήστη user και ελέγχει αν υπάρχει στη βάση δεδομένων. Αν υπάρχει αντίστοιχη εγγραφή επιστρέφει true, αλλιώς επιστρέφει false. |
| validData(): Boolean | Δέχεται το username και password ενός χρήστη user και ελέγχει αν είναι έγκυρα. Επιστρέφει false σε περίπτωση που τα στοιχεία δεν είναι σύμφωνα με τις εγγραφές της βάσης και true σε διαφορετική περίπτωση. |
| registerUser():Boolean | Δέχεται το username και password ενός χρήστη, ο οποίος θέλει να εγγραφεί στο σύστημα για πρώτη φορά. Ελέγχει αν δεν |

| | |
|--|---|
| | υπάρχει ήδη εγγραφή στη βάση με τα ίδια στοιχεία και ενημερώνει αναλόγως τους αντίστοιχους πίνακες. Σε περίπτωση επιτυχίας της εγγραφής επιστρέφει true, διαφορετικά false. |
|--|---|

4.2.7 Database Server:

Η συγκεκριμένη κλάση αναλαμβάνει τη σύνδεση, επικοινωνία και διαχείριση της βάσης δεδομένων.

| Πεδία: | |
|--------------------------|--|
| instance: DatabaseServer | Αναφέρεται στο στιγμιότυπο της κλάσης DatabaseServer. |
| database: String | Το όνομα της βάσης δεδομένων. |
| user: String | Το όνομα του χρήστη για τη σύνδεση στη βάση δεδομένων. |
| password: String | Ο κωδικός του χρήστη για τη σύνδεση στη βάση δεδομένων. |
| connection : Connection | Η σύνδεση στη βάση δεδομένων. |
| rs: ResultSet | Τα αποτελέσματα εκτέλεσης ενός query. (Νομίζω δεν χρειάζεται αφού η αντιστοιχη μεθοδος θα κανει return οτι αποτελεσμα προκυπτει) |
| query: Statement | Πεδίο απαραίτητο για την εκτέλεση των queries. (Ομοίως με παραπάνω) |
| connectionURL: string | Η διεύθυνση του server της βάσης δεδομένων. |

| Μέθοδοι: | |
|-----------------------------------|---|
| DatabaseServer(): DatabaseServer | Αποτελεί τον constructor της κλάσης DatabaseServer. |
| createConnection(): void | Δημιουργεί μια σύνδεση με τη βάση δεδομένων. |
| disconnect(): void | Τερματισμός σύνδεσης με τη βάση δεδομένων. |

| Μέθοδοι για queries: | |
|-----------------------------------|-----------------------------|
| insertUser(): void | StartCall(): void |
| UserExists(): Boolean | EndCall(): void |
| ValidUser(): Boolean | getCallID(): int |
| getUserCharge(): double | insertBlock(): void |
| removeBlock(): void | isUserBlocked(): Boolean |
| getBlockedUsers (): List<String> | insertForwarding(): void |
| removeForwarding(): void | getForwardeeName(): String |
| getForwardeeId(): int | updateUserCharge(): void |
| getUserID(): int | |

4.2.8 Proxy Extension:

Η κλάση ελέγχει τις νέες λειτουργίες που προστέθηκαν στο σύστημα μέσω του proxy (και συγκεκριμένα της Proxy κλάσης).

| Πεδία: | |
|----------------------|---|
| ts: TimerServer | Αναφορά στο instance του TimerServer. |
| db: DataBaseManager | Αναφορά στο instance του DataBaseManager |
| bs: BlockingServer | Αναφορά στο instance του BlockingServer |
| ts: TimerServer | Αναφορά στο instance του TimerServer |
| fs: ForwardingServer | Αναφορά στο instance του ForwardingServer |
| ss: SocketServer | Αναφορά στο instance του SocketServer |
| proxy:Proxy | Αναφορά στο instance του proxy |
| Μέθοδοι: | |
| Start(): void | Εκκίνηση νέου νήματος για SocketServer |
| stop(): void | Τερματισμός νήματος SocketServer και επικοινωνίας |

| | |
|----------------------------|--|
| | με τη βάση |
| GetUserFromUri(): int | Επιστρέφει το ID του χρήστη από το Uri |
| GetSourceUri (): URI | Επιστρέφει το URI που περιέχει τα στοιχεία του χρήστη που κάλεσε |
| GetTargetUri (): URI | Επιστρέφει το URI που περιέχει τα στοιχεία του χρήστη που καλέστηκε |
| IsBlocked() : boolean | Επιστρέφει αν ο καλούμενος έχει μπλοκάρει τον καλών |
| ProcBilling(): void | Ανάλογα με το μήνυμα που έχει λάβει προσθέτει μία νέα κλήση(ACK) ή τερματίζει μία κλήση(BYE) |
| ProcForwarding(): Request | Αν έχει γίνει προώθηση επιστρέφει το νέο Request με τον τελικό αποδέκτη της κλήσης |

4.2.9 Socket Server:

Ρόλος του είναι να ακούει την πόρτα(6000), να λαμβάνει τα μηνύματα, τα οποία στέλνονται στον Proxy από τον Sip Communicator και να καλεί τον WorkerServer για να τα εξυπηρετήσει.

4.2.10 Worker:

Καλείται από τον SocketServer και διαβάζει από το DataInputStream του Socket το ID του χρήστη και το μήνυμα που έστειλε (REGISTER, CKECK, LOG, CHARGES, FORWARDING_TO, FORWARD, STOP_FORWARD, BLOCKING_LIST, UNBLOCK, BLOCK) και καλεί την ανάλογη κλάση/μέθοδο για να την εξυπηρετήσει και επιστρέφει μέσω του DataOutputStream κατάλληλο μήνυμα (επιβεβαίωσης, αποτυχίας).

4.2.11 BlockUI:

Ρόλος της κλάσης είναι η υλοποίηση και επικοινωνία του γραφικού περιβάλλοντος του Blocking.

| | |
|-----------------------|-----------------------------------|
| Μέθοδοι: | |
| Init(): JPanel | Αρχικοποιεί |
| InitiateList(): void | Εμφανίζει τη λίστα με τις φραγές. |

| | |
|----------------------|--|
| BlockUser(): void | Στέλνει το κατάλληλο μήνυμα στο Proxy μέσω sockets για να προστεθεί ένας νέος χρήστης ως περιορισμένος. |
| UnblockUser(): void | Στέλνει το κατάλληλο μήνυμα στο Proxy μέσω sockets για να αφαιρεθεί ένας χρήστης από τη λίστα των περιορισμένων. |

4.2.12 ForwardingUI:

Ρόλος της κλάσης είναι η υλοποίηση και επικοινωνία του γραφικού περιβάλλοντος του Forwarding.

| Μέθοδοι: | |
|---------------------------------|--|
| Init(): JPanel | Αρχικοποιεί |
| ChangeForwardRecipient(): void | Στέλνει το κατάλληλο μήνυμα στο Proxy μέσω sockets για να προστεθεί ένας νέος χρήστης ως ο παραλήπτης των κλήσεων. |
| DisableForwarding(): void | Στέλνει το κατάλληλο μήνυμα στο Proxy μέσω sockets για να απενεργοποιήσει το forwarding. |

4.2.13 BillingUI:

Ρόλος της κλάσης είναι η υλοποίηση και επικοινωνία του γραφικού περιβάλλοντος του Billing.

| Μέθοδοι: | |
|-------------------------|----------------------------------|
| Init(): JPanel | Αρχικοποιεί |
| GetListOfCalls(): void | Γεμίζει τη λίστα με τις κλήσεις. |
| GetUserBilling(): void | Λαμβάνει τη χρέωση της κλήσης. |

4.2.14 NoEditable Table Model:

Κλάση για την υλοποίηση μη επεξεργάσιμου πίνακα.

4.2.15 GUIManager (τροποποίηση της αρχικής κλάσης):

| | |
|---|---|
| Πεδία: | |
| userCred: UserCredentials | Κρατάει τα διαπιστευτήρια για τον παρόν χρήστη |
| Μέθοδοι: | |
| SetUserCredentials(): void | Set μέθοδος για το πεδίο userCred |
| SetCommunicationActionsEnabled(): void | Τα κουμπιά blockButton, forwardButton, historyButton γίνονται enabled βάσει του ορίσματος |
| InitActionListeners(): void | Προσθέτει στα κουμπιά blockButton, forwardButton, historyButton τους αντίστοιχους ActionListeners |

4.2.16 Phone Frame (τροποποίηση της αρχικής κλάσης):

| | |
|---|---|
| Πεδία: | |
| blockButton: JButton | Το κουμπί που πατάει ο χρήστης για να ανοίξει το παράθυρο με τις επιλογές Blocking |
| forwardButton: JButton | Το κουμπί που πατάει ο χρήστης για να ανοίξει το παράθυρο με τις επιλογές Forwarding |
| historyButton: JButton | Το κουμπί που πατάει ο χρήστης για να ανοίξει το παράθυρο με το ιστορικό των κλήσεων |
| Μέθοδοι: | |
| SetCommunicationActionsEnabled(): void | Τα κουμπιά blockButton, forwardButton, historyButton γίνονται enabled βάσει του ορίσματος |

4.2.17 Authentication Splash (τροποποίησης αρχικής κλάσης):

| | |
|---------------|--|
| Πεδία: | |
|---------------|--|

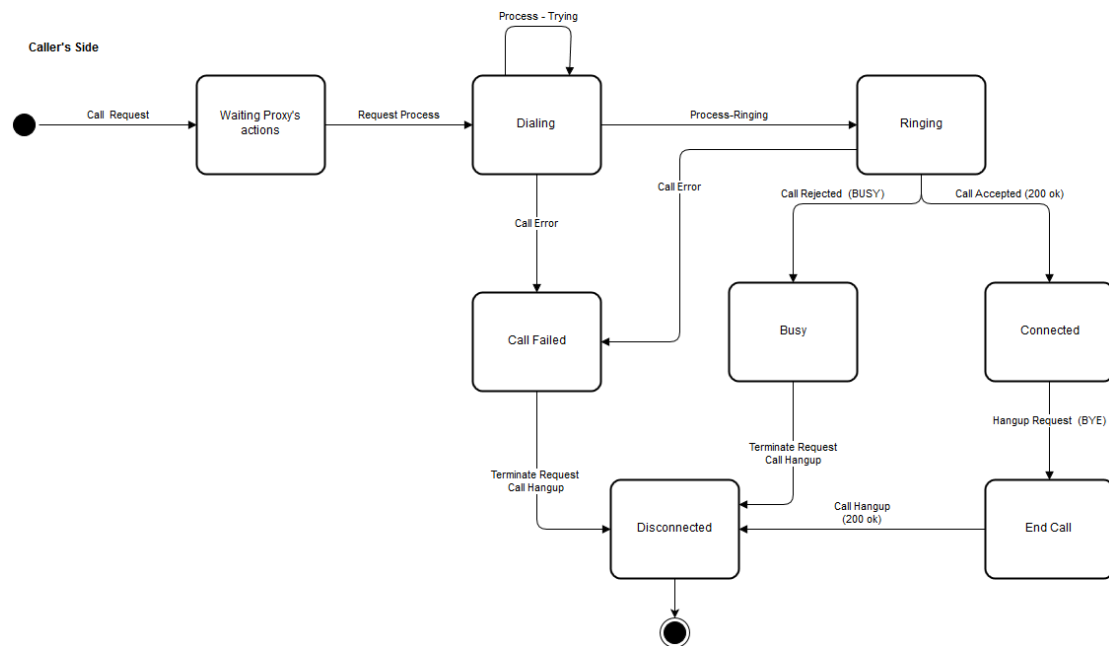
| | |
|-----------------------|---|
| myResponse: String | Το μήνυμα που αποστέλλεται στον Proxy |
| Μέθοδοι | |
| dialogDone(): void | Στέλνει τα κατάλληλα μηνύματα για register, login, cancel |
| setMyResponse(): void | Η set μέθοδος για το myResponse |
| sendMessage(): void | Στέλνει το κατάλληλο μήνυμα στον Proxy με χρήση sockets |

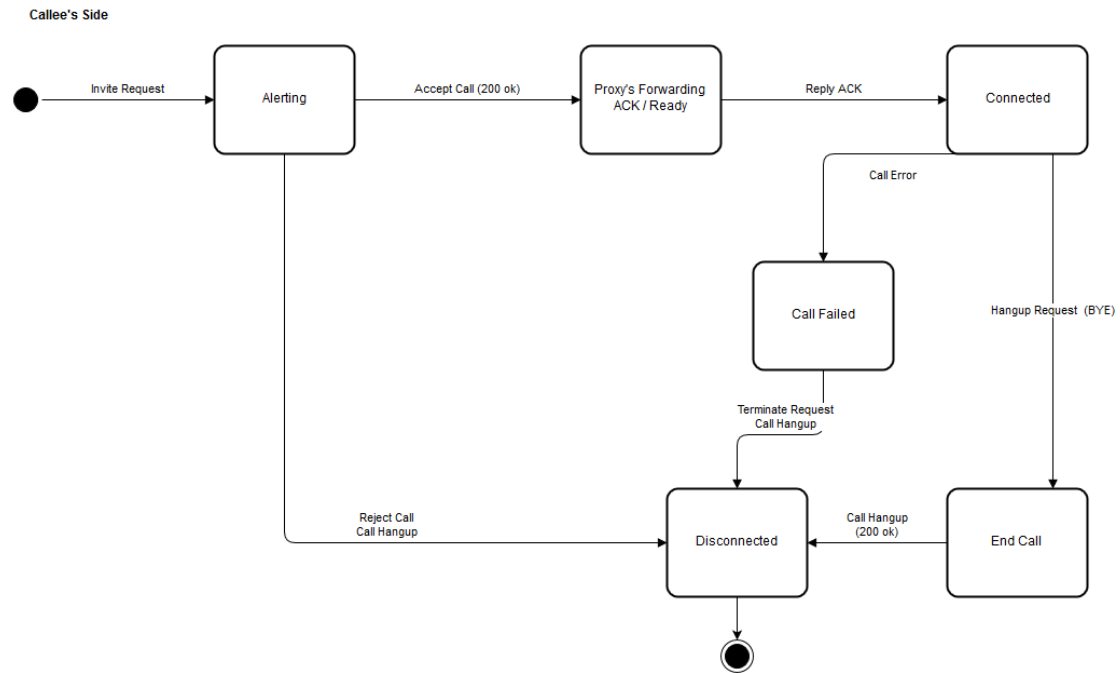
4.2.18 SIP Communicator (τροποποίησης υπάρχουσας κλάσης):

Προσθήκη για αρχικοποίηση setUserCredentials του GuiManager.

5 Διαγράμματα Κατάστασης

Τα διαγράμματα κατάστασης από τη μεριά των caller και callee παρουσιάζονται παρακάτω:





6 Ανοιχτά Ζητήματα

6.1 Ασφάλεια Συστήματος

Είναι απαραίτητο να δοθεί ιδιαίτερη έμφαση στην προστασία απέναντι σε μη εξουσιοδοτημένους χρήστες που προσπαθούν να αποκτήσουν πρόσβαση στη βάση δεδομένων ή στους servers.

6.2 Εύρεση Κύκλων στην Προώθηση Κλήσεων

Δεν έχει καθοριστεί ο τρόπος με τον οποίο θα ανιχνεύονται οι κύκλοι, ωστόσο κατά την υλοποίηση του προγράμματος θα γίνει αναλυτικότερη εξέταση των απαραίτητων παραμέτρων.

6.3 Γραφικό Περιβάλλον

Για την καλύτερη χρήση της εφαρμογής είναι αναγκαία η βελτίωση του GUI, μέσω τεχνικών αλληλεπίδρασης ανθρώπου – μηχανής.

6.4 Billing Σύστημα

Κρίνεται απαραίτητο να καθοριστούν τα προγράμματα χρεώσεων που θα είναι διαθέσιμα στο κοινό, καθώς και ο χειρισμός προβληματικών καταστάσεων, όπως κάποια πιθανή κατάρρευση του Billing Server.