ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής



Εργασία Μαθήματος «Προγραμματισμός στο διαδίκτυο και στον παγκόσμιο ιστό»

|  |  |
| --- | --- |
| 2 | ***Εργασία 2*** |
| Όνομα φοιτητή – Αρ. Μητρώου  (όλων σε περίπτωση ομαδικής εργασίας) | Αλεξανδρής Λάμπρος – P22007 |
| Λαζαρίδης Χρήστος-Λάζαρος – P22083 |
| Όσμα Φετί – P22126 |
| Καλογερόπουλος Αθανάσιος – P22223 |
| Ημερομηνία παράδοσης | 23/06/2024 |

**Εκφώνηση της άσκησης**

1 Εγκατάσταση και παραμετροποίηση application server και database server. 1.1 Εγκαταστήστε και παραμετροποιήστε τον application server (π.χ. Tomcat, Glassfish) και το Σύστημα Διαχείρισης Βάσης Δεδομένων (π.χ. mysql, postgress, sqlite). Η εγκατάσταση του Application Server να συνδεθεί με το προγραμματιστικό περιβάλλον IDE που χρησιμοποιείτε (π.χ. Eclipse). 1.2 Δημιουργήσετε την σύνδεση του application server με τον database server, χρησιμοποιώντας τον αντίστοιχο jdbc database connector για το σύστημα βάσης της επιλογής σας. Χρησιμοποιήστε τη σύνδεση του μοντέλου 3-tier. (μπορείτε να βρείτε αντίστοιχο παράδειγμα στα παραδείγματα κώδικα που περιλαμβάνονται στη σελίδα του μαθήματος). 2 Δημιουργία Βάσης Δεδομένων

2.1 Δημιουργήστε το Μοντέλο Οντοτήτων-Σχέσεων, το οποίο περιγράφει τη Βάση Δεδομένων που θα χρησιμοποιήσετε για την εφαρμογή σας. Ενδεικτικά (και όχι περιοριστικά) θα περιλαμβάνει πίνακες όπως, Πελάτες, Πωλητές, Προγράμματα Τηλεφωνίας, Αριθμούς Τηλεφώνων, Κλήσεις κτλ. Να περιλάβετε στο μοντέλο σας τις σχέσεις μεταξύ των πινάκων. 2.2 Με τη βοήθεια του Μοντέλου Οντοτήτων-Σχέσεων, να δημιουργήσετε και να εκκινήσετε τη βάση στον database server. Μπορείτε να χρησιμοποιήσετε οποιοδήποτε βοηθητικό εργαλείο για την εξαγωγή της βάσης από το μοντέλο (π.χ. mysql Workbench για mysql). 2.3 Εισάγετε εικονικά δεδομένα σε όλους τους πίνακες, λαμβάνοντας υπόψη τα εξωτερικά κλειδιά που πιθανώς έχουν οι πίνακες.

3 Δημιουργία web project 3.1 Δημιουργήστε ένα Dynamic Web Project. 3.2 Δημιουργήστε ένα ή περισσότερα πακέτα κλάσεων, τα οποία θα περιλαμβάνουν τις βασικές κλάσεις που έχετε υλοποιήσει στην προηγούμενη άσκηση. [1] 3.3 Δημιουργείστε ένα ή περισσότερα νέα πακέτα κλάσεων τα οποίο θα περιλαμβάνουν όλα τα servlet που θα χρησιμοποιήσετε στην εργασία (ενδεικτικά ClientServlet, SellerServlet, AdminServlet). Στην συγκεκριμένη άσκηση θα υλοποιήσετε μόνο ένα μέρος από αυτά, όπως αναφέρεται στο βήμα 5.

4 Δημιουργία διαδικτυακής διεπαφής 4.1 Σε αυτό το βήμα, θα υλοποιήσετε τη διαδικτυακή διεπαφή (html σελίδες) που θα χρησιμοποιούν οι χρήστες όλων των κατηγοριών (Πελάτες, Πωλητές, Διαχειριστές Εφαρμογής) για να αλληλεπιδρούν με την εφαρμογή και να χρησιμοποιούν τις αντίστοιχες μεθόδους που απαιτούνται. 4.2 Θα υπάρχει ένα κεντρικό μενού (π.χ.. μία σελίδα index.html), η οποία θα είναι η αρχική σελίδα για όλους τους χρήστες. Από την αρχική σελίδα οι διάφοροι χρήστες θα μπορούν να συνδεθούν (login) και να έχουν πρόσβαση στις λειτουργίες τους. (Σε αυτή την άσκηση θα υλοποιήσετε μόνο όσες λειτουργίες αναφέρονται στο βήμα 5)

5 Υλοποίηση επιλεγμένων μεθόδων (λειτουργιών) 5.1 Για τους Πωλητές (Sellers) της εφαρμογής, να υλοποιήστε τις εξής μεθόδους: 5.1.1 Λειτουργία σύνδεσης (login) και αποσύνδεσης (logout). 5.1.2 Προβολή όλων των διαθέσιμων προγραμμάτων/πακέτων τηλεφωνίας. 5.1.3 Εισαγωγή νέου πελάτη. 5.1.4 Αντιστοίχηση πελάτη σε πρόγραμμα τηλεφωνίας. 5.2 Για την προβολή του αποτελέσματος κάθε μίας από τις παραπάνω ενέργειες, θα δημιουργείται μία δυναμική html σελίδα μέσω του servlet (ή συνδυασμός servlet και JSP). Δημιουργήστε επίσης τις απαραίτητες στατικές html σελίδες που απαιτούνται.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

[1 Σύνοψη 4](#_Toc169113387)

[1.1 Βάση 4](#_Toc169113388)

[1.2 Sellers 4](#_Toc169113389)

[2 Βάση Δεδομένων 5](#_Toc169113390)

[2.1 Σχεσιακό Μοντέλο 5](#_Toc169113391)

[2.2 Συναρτήσεις Βάσης 7](#_Toc169113392)

[2.3 Εισαγωγή dummy δεδομένων στη βάση 10](#_Toc169113393)

[3 Επίπεδο Εφαρμογής 14](#_Toc169113394)

[3.1 Επικοιννία με τη Βάση 16](#_Toc169113395)

[3.2 Servlets 16](#_Toc169113396)

[3.3 Front End (jsp) 20](#_Toc169113397)

[4 Βιβλιογραφικές Πηγές 22](#_Toc169113398)

1. Σύνοψη
   1. Βάση

Η βάση δεδομένων φτιάχτηκε με τις παραδοχές:

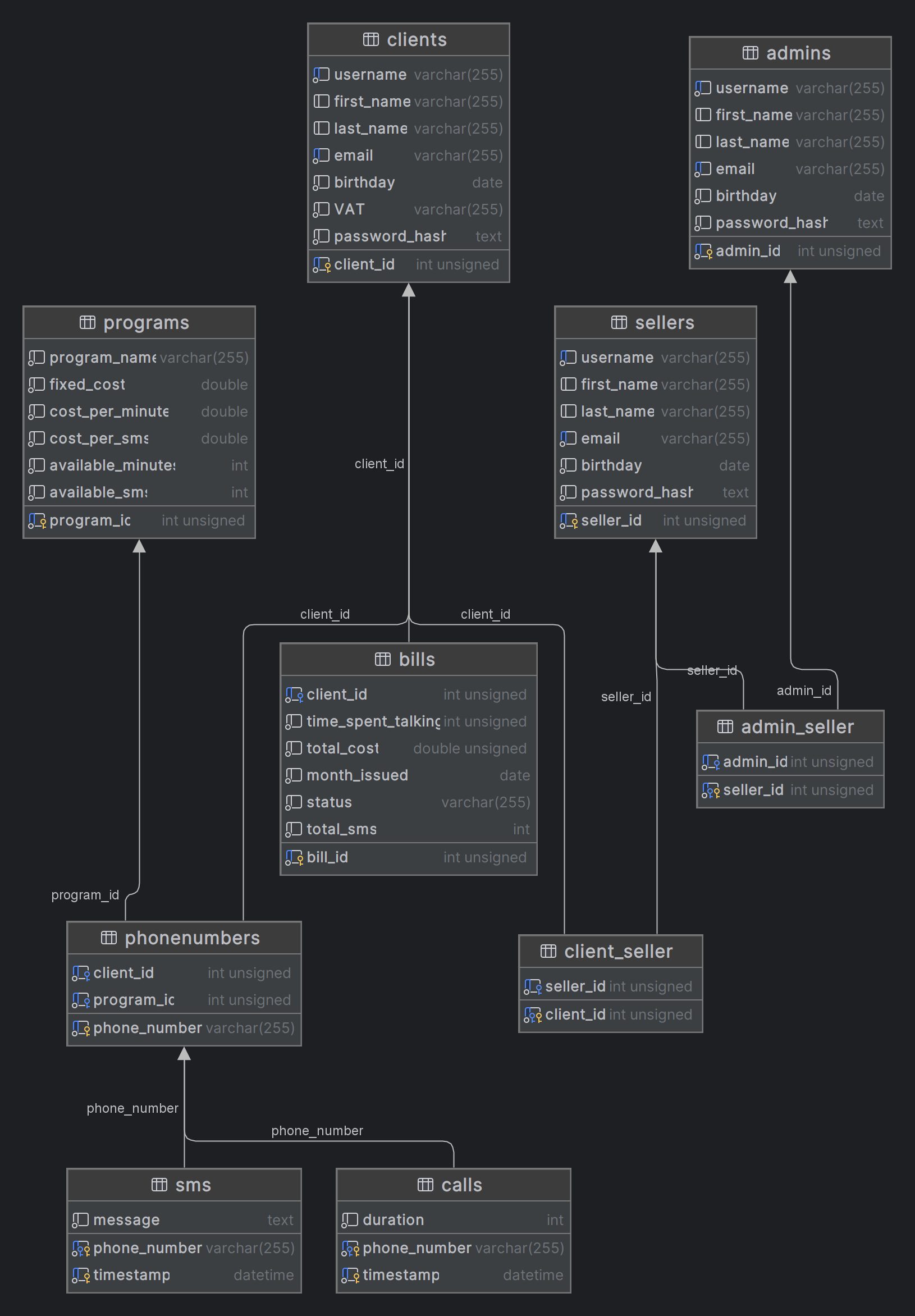
* Η κάθε οντότητα (Client, Seller, Admin) έχουν τους δικούς τους πίνακες με τα αντίστοιχα δεδομένα για τον καθένα (όνομα, επίθετο, username (μοναδικό), ΑΦΜ (μοναδικό και μόνο για τους clients), email (μοναδικό) και ημερομηνία γέννησης καθώς και ένα ID)
* Τα προγράμματα χρέωσης αφορούν τα διαθέσιμα λεπτά ομιλίας και μυνήματα. Μετά την υπερβολή αυτών, η χρέωση γίνεται ανά λεπτό ομιλίας ή ανα μύνημα σύμφωνα πάλι με τα στοιχεία του προγράμματος. Τα προγράμματα αυτά βρίσκονται σε ξεχωριστό πίνακα.
* Ο πελάτης (Client) έχει αριθμούς τηλεφώνου, 1 ή περισσότερους (ή και κανένα), καθένας από τους οποίους τιμολογείτε με βάση το πρόγραμμα το οποίο του έχει ανατεθεί, δηλαδή το πρόγραμμα χρέωσης σχετίζεται με αριθμούς και όχι με χρήστες.
* Για να γίνει ρεαλιστική τιμολόγηση έχουν φορτωθεί στη βάση περίπου 80.000 μυνήματα και περίπου 90.000 κλήσεις για όλη τη διάρκεια του έτους, για τους 5.000 dummy πελάτες. Προφανώς οι δημιουργημένοι πελάτες δεν θα έχουν κάποια κλήση στο όνομα τους, καθώς στη πραγματική ζωή η καταγραφή των κλήσεων θα γινόταν με άλλα μέσα. Το κάθε μύνημα και η κάθε κλήση βρίσκονται σε δικούς τους πίνακες και χαρακτιρίζονται από ένα μοναδικό συνδυασμό αριθμού που τα πραγματοποίησε και timestamp (έχουμε λάβει μέριμνα κατά τη διαδικασία του data population), καθώς δεν θα ήταν ρεαλιστικό να υπάρχουν πολλές κλήσεις από τον ίδιο αριθμό την ίδια χρονική στιγμή.
* Υπάρχουν μοναδικές σχέσεις 1 προς πολλά πελάτη-πωλητή και διαχειριστή-πωλητή (client-seller και admin-seller) για τη διαχείριση των οποίων έχουν δημιουργηθεί ξεχωριστοί πίνακες.
  1. Sellers

Οι διαδικασίες που μπορούν να εκτελούν οι πωλητές είναι οι εξής:

* Οριστική και μη αναστρέψιμη διαγραφή πελάτη από το σύστημα
* Επεξεργασία στοιχείων του πελάτη, η οποία γίνεται σε ξεχωριστή φόρμα, για την οποία έχουν γίνει οι εξής παραδοχές:
  + Ο πωλητής μπορεί να κάνει αλλαγές στο προγράμματα των τηλεφώνων των πελατών με τους οποίους σχετίζεται αλλά μόνο ένα τηλέφωνο ανα submit.
  + Μπορεί να διαγράψει τηλέφωνο οριστικά.
  + Μπορεί να προσθέσει αριθμό τηλεφώνου, ο οποίος όμως δημιουργείτε τυχαία.
  + Μπορεί να αλλάξει το συνθηματικό του πελάτη
* Η προσθήκη πελατών θα μπορεί να γίνει από πωλητές και με τη δημιουργία τους θα συσχετίζονται αυτόματα με το πελάτη. Αξίζει να σημειωθεί πως η δημιουργία του πελάτη γίνεται με μοναδικό και τυχαία δημιουργημένο αριθμό τηλεφώνου. Το συνθηματικό του πελάτη το ελέγχει ο πωλητής
* Ο πωλητής έχει δυνατότητα να επιρρεάσει τους πελάτες με τους οποίους διατηρεί σχέση στη βάση, η οποία σχέση δεν γίνεται να διαγραφτεί από το πωλητή ή το πελάτη. Η βασική σελίδα του πωλητή του εμφανίζει βασικές πληροφορίες για τους πελάτες του.
* Ο πωλητής δεν έχει δικαιοδοσία να αλλάξει τα δεδομένα στο πίνακα των προγραμμάτων.

1. Βάση Δεδομένων
   1. Σχεσιακό Μοντέλο

Η βάση που αναπτύχθηκε για την εφαρμογή, είναι μια σχεσιακή βάση SQL τριών οντοτήτων, των Client, Admin και Seller, οι οποίοι σχετίζονται μεταξύ τους με τους πίνακες client\_seller, admin\_seller. Οι ιδιότητες του client, όπως οι αριθμοί τηλεφώνου, οι λογαριασμοί αλλά και τα πρόγράμματα χρέωσης υπάρχουν στους αντίστοιχους πίνακες, όπως και η καταμέτρηση κλήσεων και μυνημάτων. Το διάγραμμα της βάσης είναι το παρακάτω:



Εικόνα 1: Διάγραμμα Σχεσιακής Βάσης

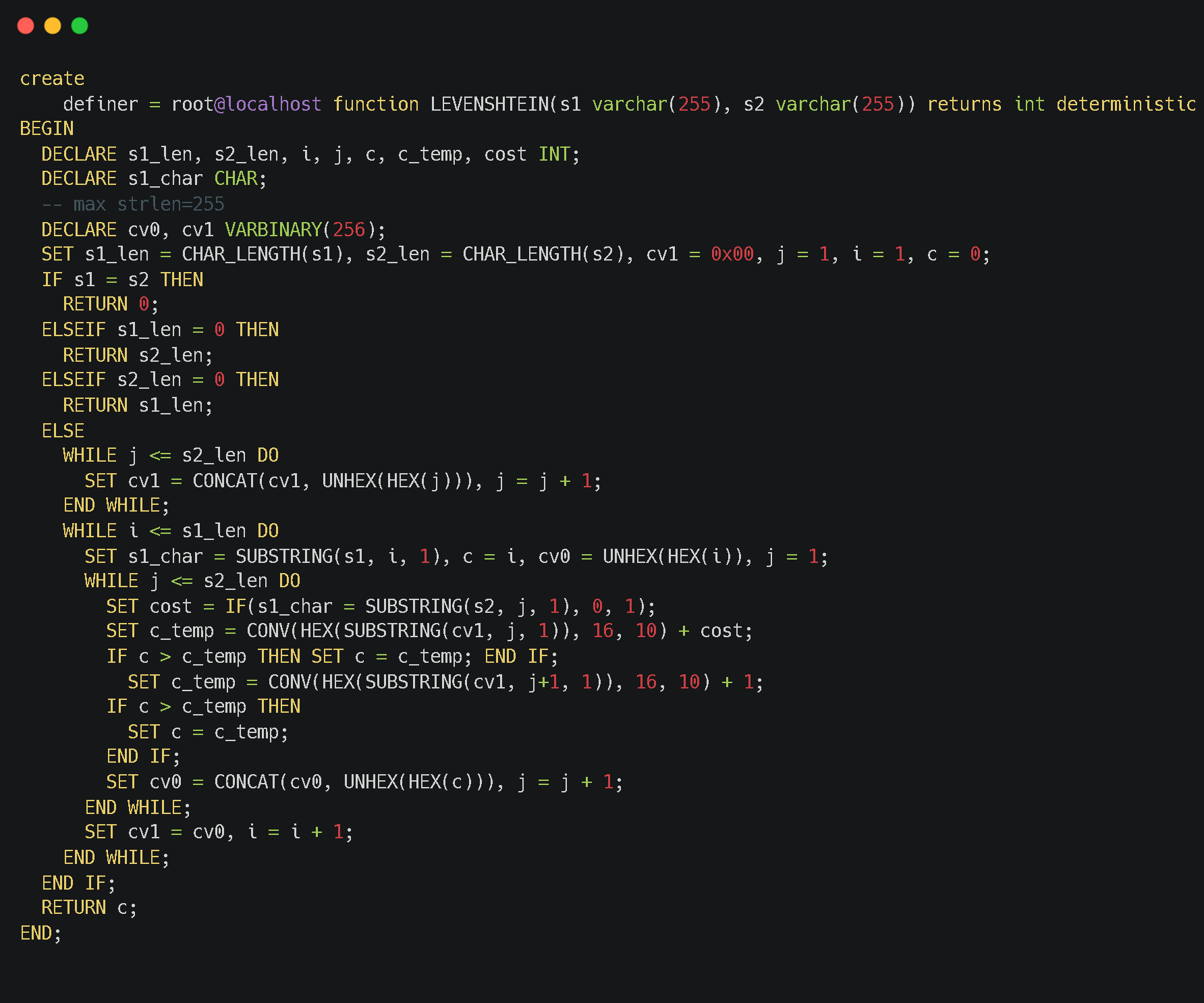
* 1. Συναρτήσεις Βάσης

Εκτός από τους πίνακες στη βάση έχουν υλοποιηθεί και οι εξής μέθοδοι:Μέθοδος CalculateBill η οποία υπολογίζει αυτόματα τον μηνιαίο λογαριασμό του εκάστοτε πελάτη. Παίρνει ως ορίσματα την ID του πελάτη και την ημερομηνία (μόνο το month έχει σημασία), και αφού ελέγξει αν υπάρχει είδη λογαριασμός για τον αντίστοιχο και μήνα, αθρίζει τα calls και sms για κάθε αριθμό τηλεφώνου του αντίστοιχου πελάτη.



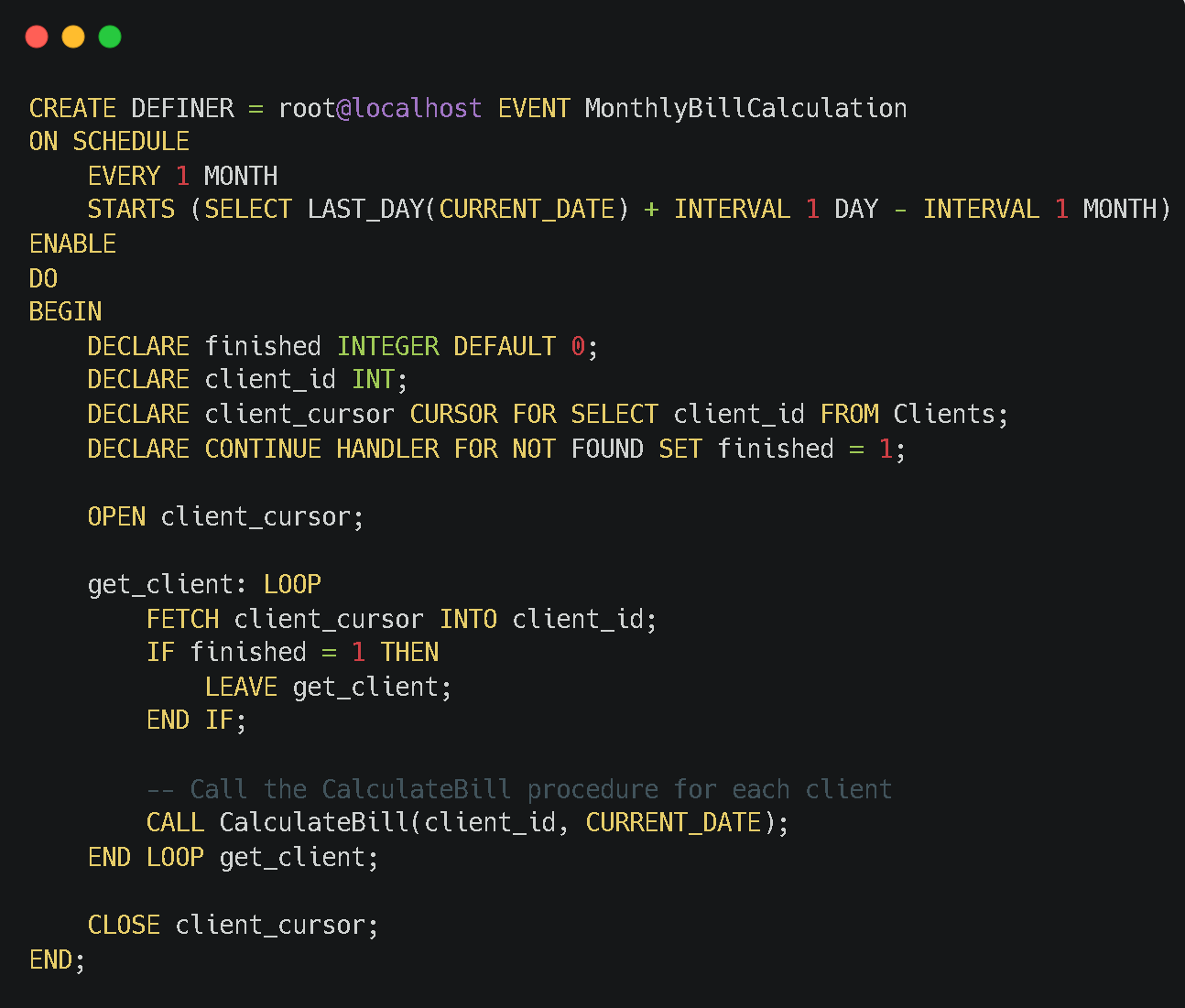
Εικόνα 2: CalculateBill

Ακόμα έχει υλοποιηθεί μια συνάρτηση String Distance Check με βάση τον αλγόριθμο του Levenshtein με σκοπό τη δυναμική αναζήτηση οντοτήτων στη βάση



Εικόνα 3: Levenshtein Distance

Τέλος έχει υλοποιηθεί ένα event trigger, το οποίο καλεί αυτόματα τη CalculateBill μηνιαία

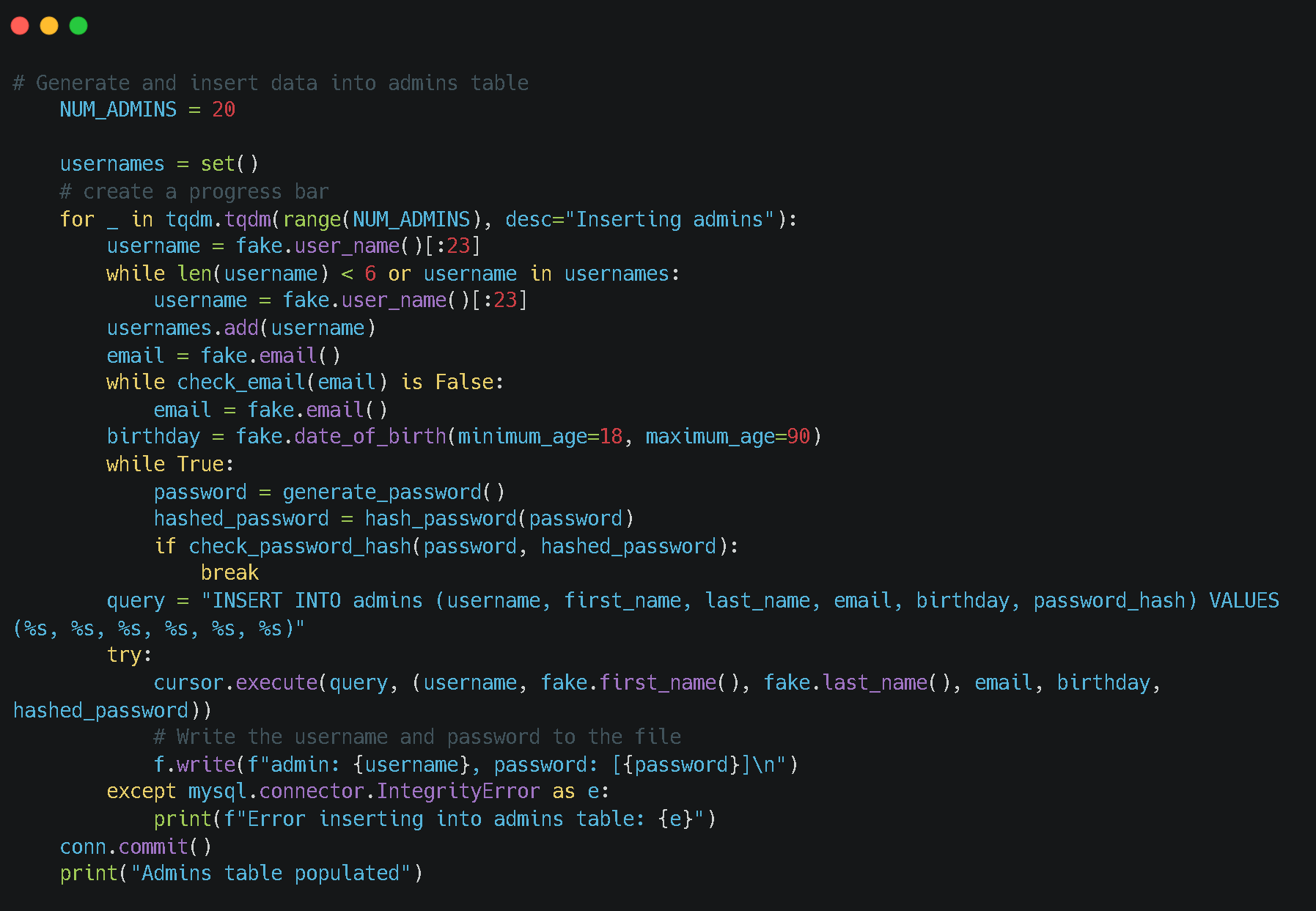


Εικόνα 4: MonthlyBillCalculation

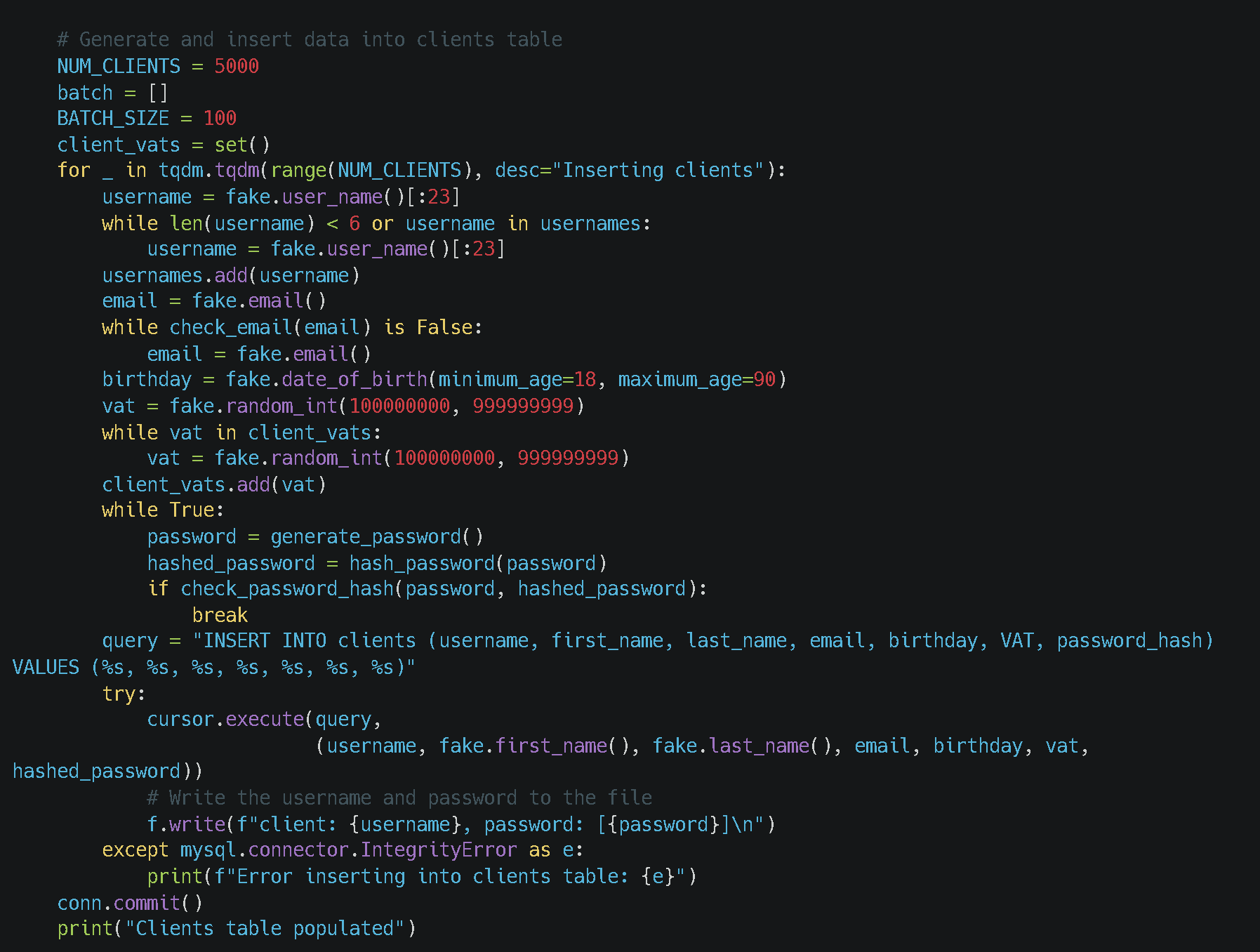
* 1. Εισαγωγή dummy δεδομένων στη βάση

Εισαγωγή δεδομένων στη βάση γίνεται με το αρχείο Main.py, το οποίο αποτελεί ένα python script που με τη χρήση της βιβλιοθήκης fake, μαζί με τη random και τη datetime, εισάγουν στη βάση 20 Admins, 100 Sellers, 5000 Clients, 11 προγράμματα (10 κανονικά και μια προεπιλογή), ένα τυχαίο αριθμό αριθμών τηλεφώνου ανάμεσα στα 6000 και 10000 και ένα τυχαίο αριθμό κλήσεων και sms, ανάμεσα σε 50000 και 100000. Τέλος δημιουργεί σχέσεις Client-Seller και Admin-Seller και δημιουργεί λογαριασμούς από τον Γενάρη μέχρι και το Μάιο του 2024. Τα στοιχεία σύνδεσης (username/password) για κάθε χρήστη καταγράφονται στο user\_credentials.txt

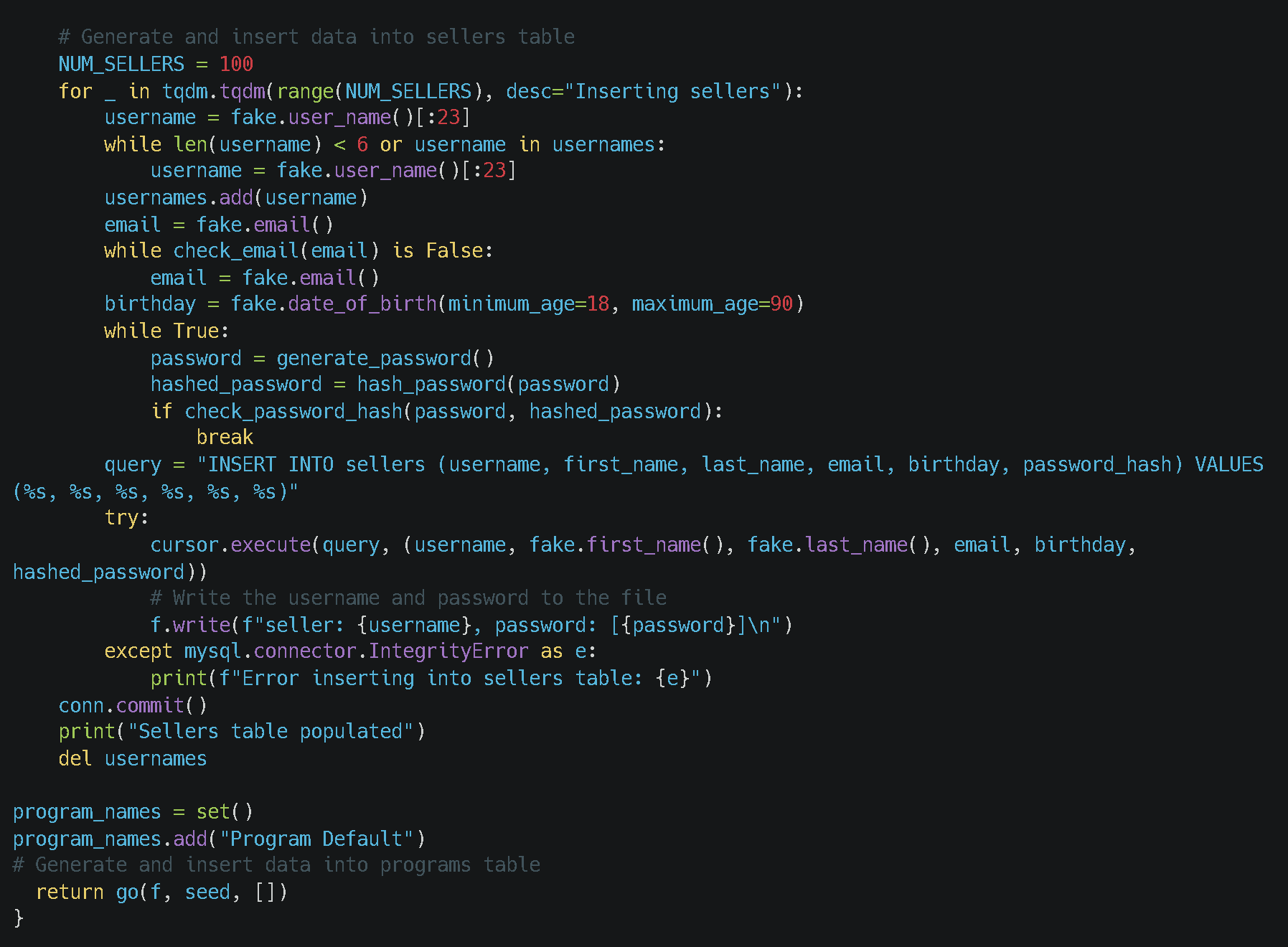
Παράδειγμα χρήσης Main.py από τους πίνακες Admins, Clients και Sellers



Εικόνα 5: Admins



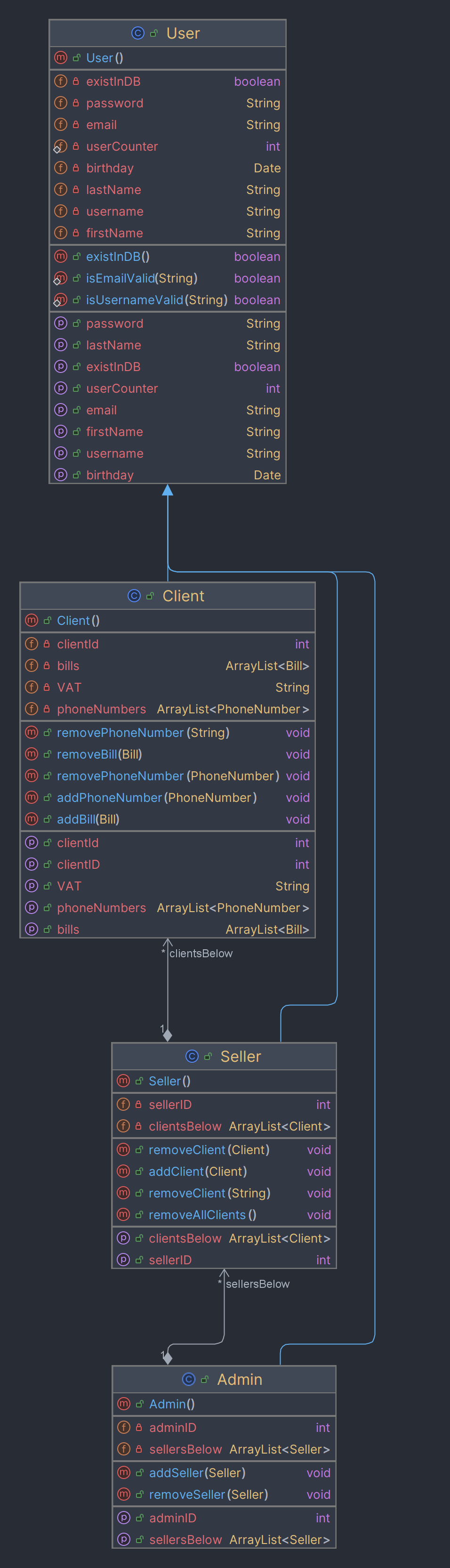
Εικόνα 6: Clients



Εικόνα 7: Sellers

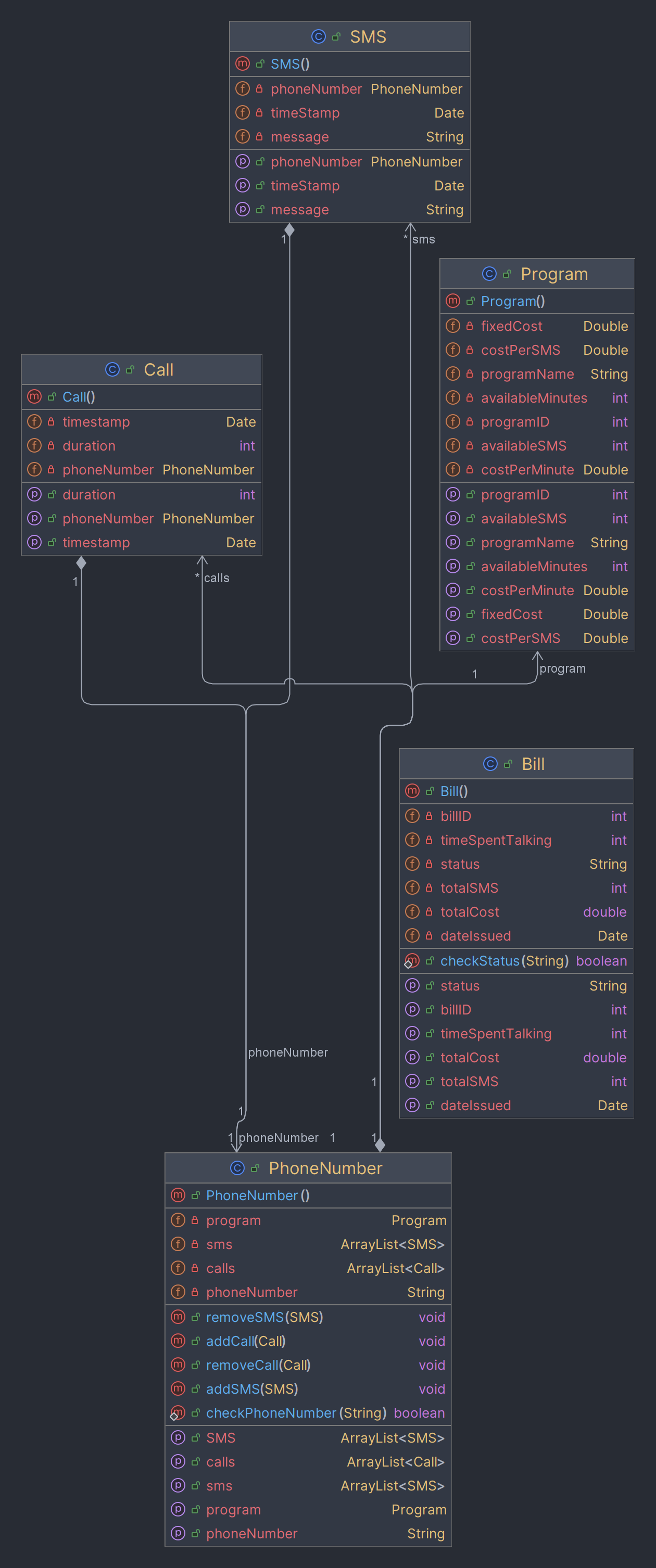
1. Επίπεδο Εφαρμογής

Ο κώδικας σε java της εργασίας έχει οργανωθεί σε 3 πακέτα. Το πακέτο Beans, το οποίο έχει τις συναρτήσεις JavaBeans για τους Χρήστες (com.Beans.Users)



Εικόνα 8: Διάγραμμα Users

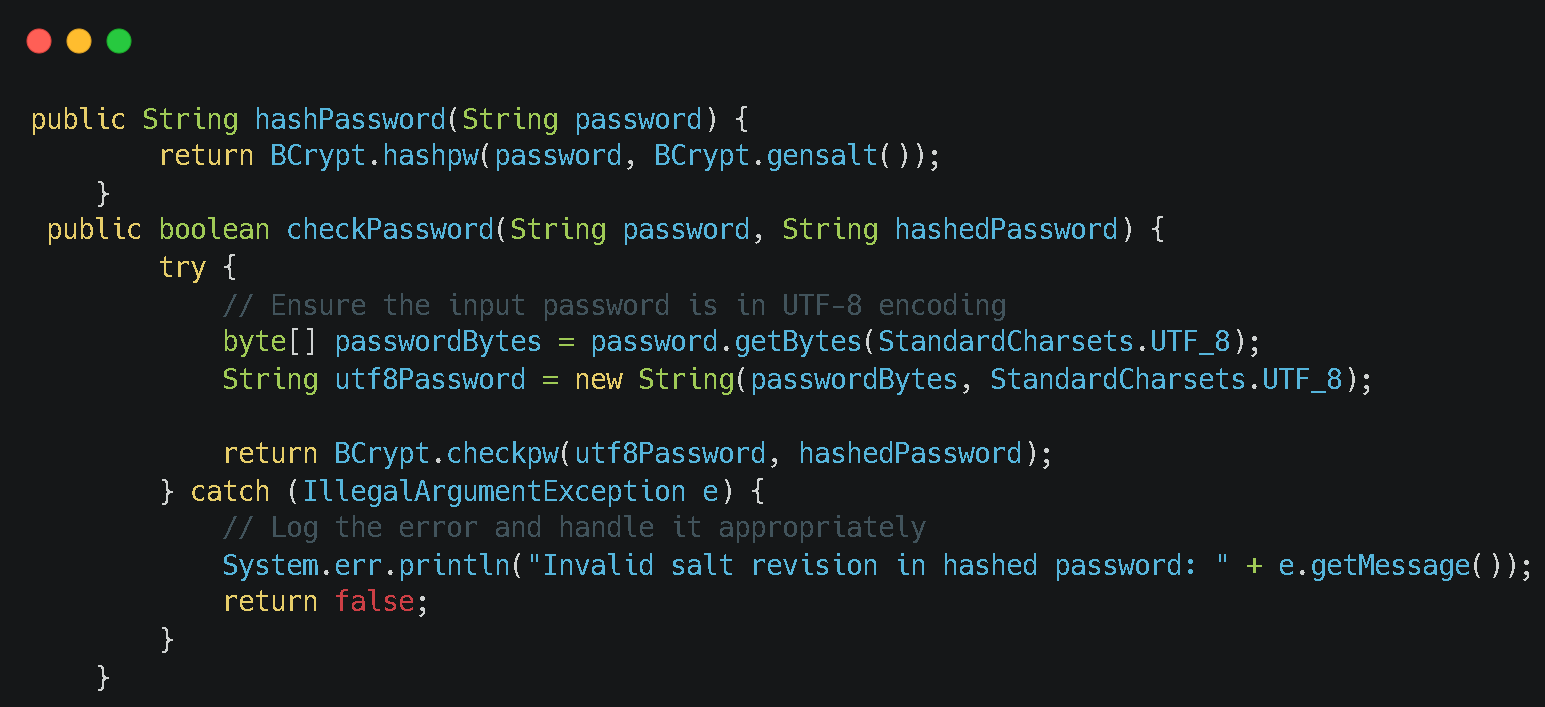
και τις βοηθητικές κλάσεις (com.Beans.Util)



Εικόνα 9: Διάγραμμα Utilities

* 1. Επικοιννία με τη Βάση

Η επικοινωνία της εφαρμογής με τη βάση επιτυγχάνεται με τις 4 κλάσεις του πακέτου db, Database, AdminDB, ClientD και SellerDB. Η κλάση Database πραγματοποιεί τη σύνδεση με τη βάση SQL και την αντικειμενοποιεί. Αντικείμενα της κλάσης Database χρησιμοποιούνται από τις υπόλοιπες συναρτήσεις του πακέτου. Αξίζει να σημειωθεί πως η συνάρτηση Database υλοποίει και το hashing των συνθηματικών καθώς και τον αντίστοιχο έλεγχο, μέσο της βιβλιοθήκης jbcrypt



Εικόνα 10: Συναρτήσεις Password Hashing στη Database.java

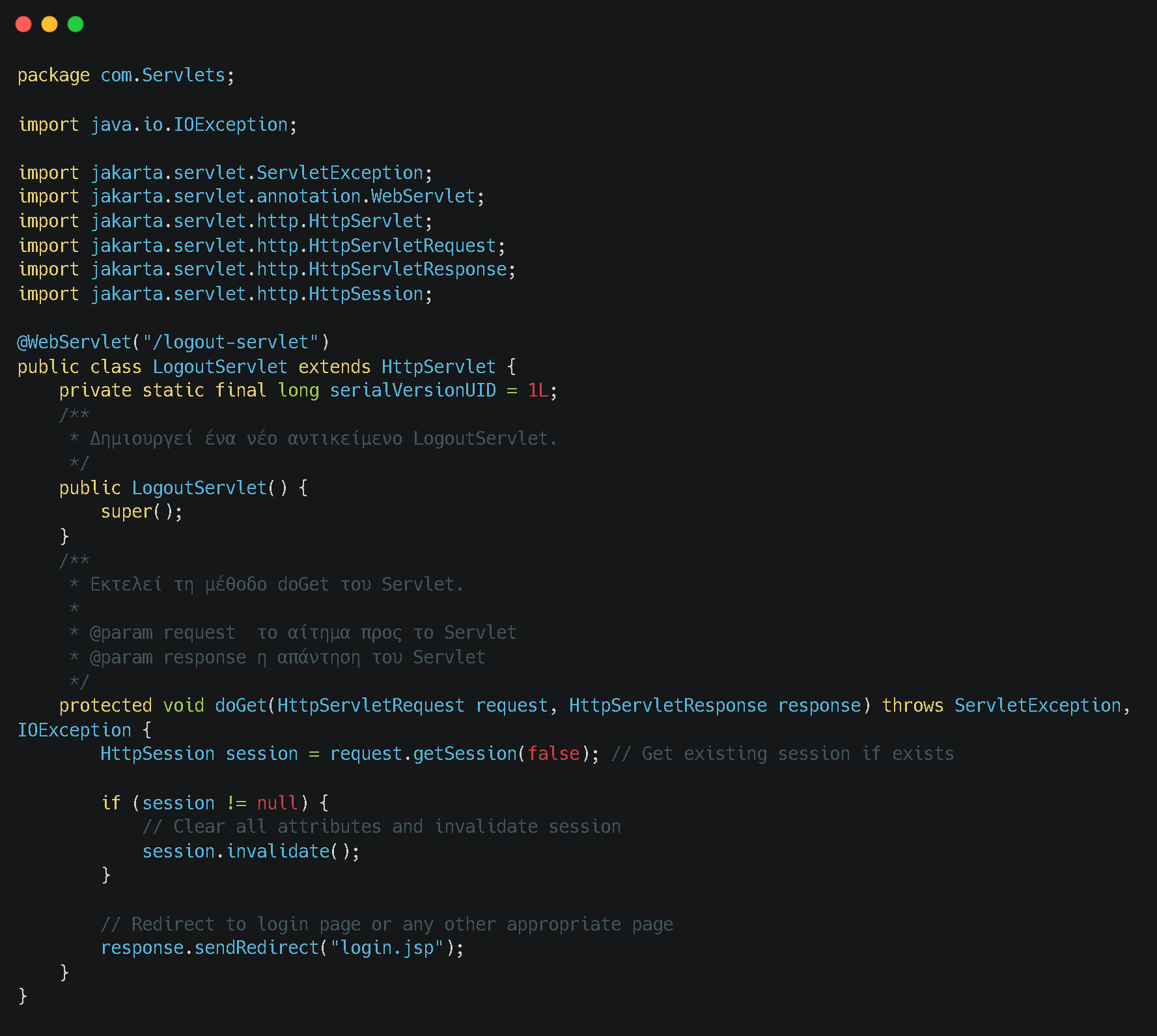
Οι συναρτήσεις ClientDB, AdminDB και SellerDB υλοποιούν την αλληλεπίδραση με τη βάση για τις λειτουργίες της εκάστοτε οντότητας (Client, Admin, Seller). Η εκτενής περιγραφή τους δεν είναι δυνατή εδώ καθώς οι κλάσεις αυτές είναι πολύ μεγάλες σε έκταση. Παρόλα αυτά στα δημιουργημένα javadocs περιγράφεται αναλυτικά η κάθε κλάση.

* 1. Servlets

Οι κλάσεις Servlets υλοποιούν τις λειτουργίες του Backend. Μέχρι στιγμής έχουν δημιουργηθεί τα LoginServlet, το οποίο ελέγχει το όνομα χρήστη και το συνθηματικό που δόθηκε, και εφόσον επαληθευθούν, εντάσει τα αντίστοιχα δεδομένα στο session και επαναδιευθύνει το χρήστη στην αντίστοιχη σελίδα jsp, μέσο ενός dispatcher ή της εντολής redirect. Τo LogoutServlet ουσιαστικά καταστρέφει το session. Το SellerServlet πραγματοποιεί τις λειτουργίες του Seller. Η doGet πραγματοποιεί κυρίως την περιήγηση του Seller ανάμεσα στα αντίστοιχα jsp καθώς και τις λειτουργίες ανανέωσης του αριθμού τηλεφώνου ενός πελάτη. Ακόμη υλοποίει την ολική και οριστική διαγραφή του πελάτη από το σύστημα. Η doPost χρησιμοποιείτε από τις φόρμες για την εισαγωγή και ανανέωση των στοιχείων του πελάτη, στα editClient.jsp και addClient.jsp



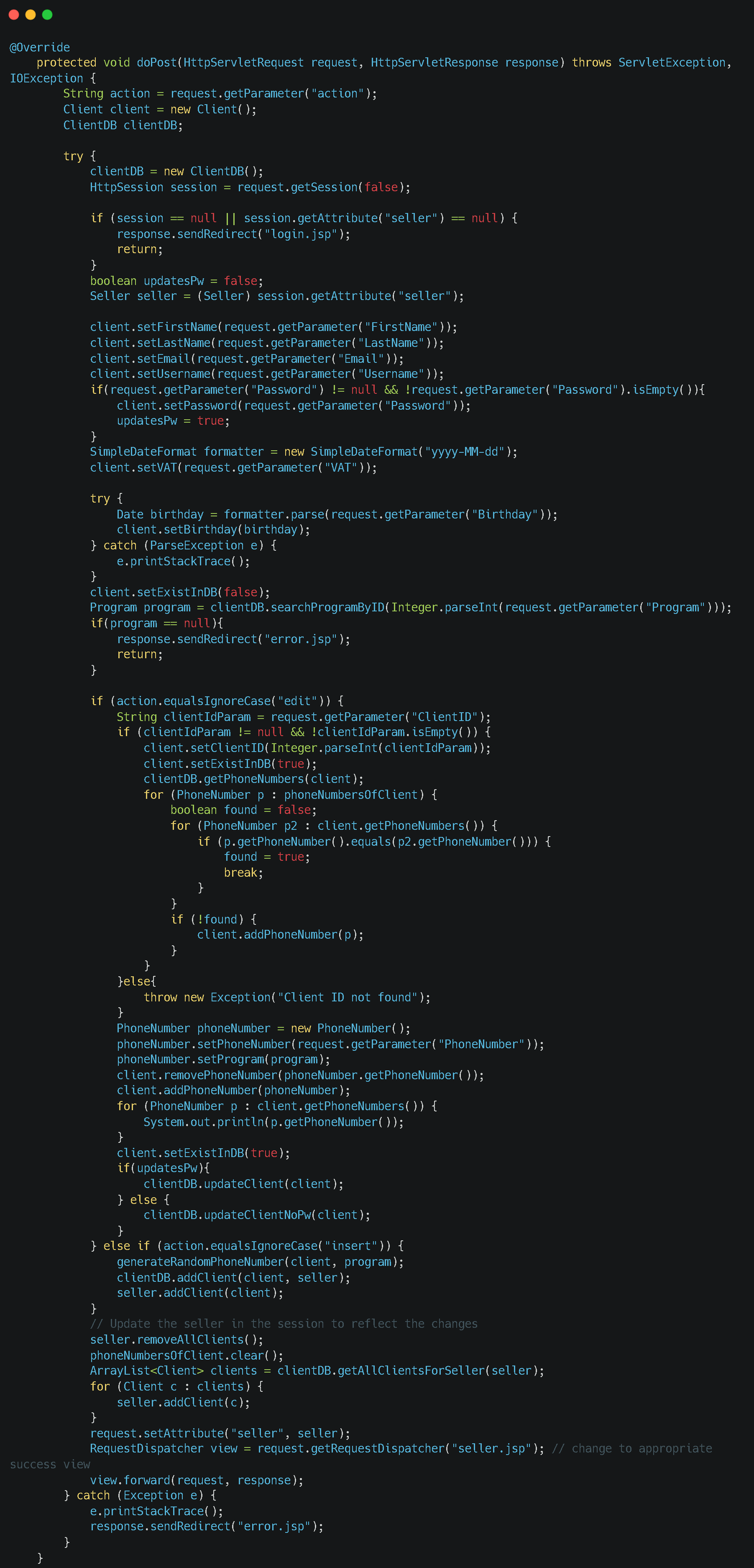
Εικόνα 11: Seller Login μέσα στο LoginServlet



Εικόνα 12: LogoutServlet

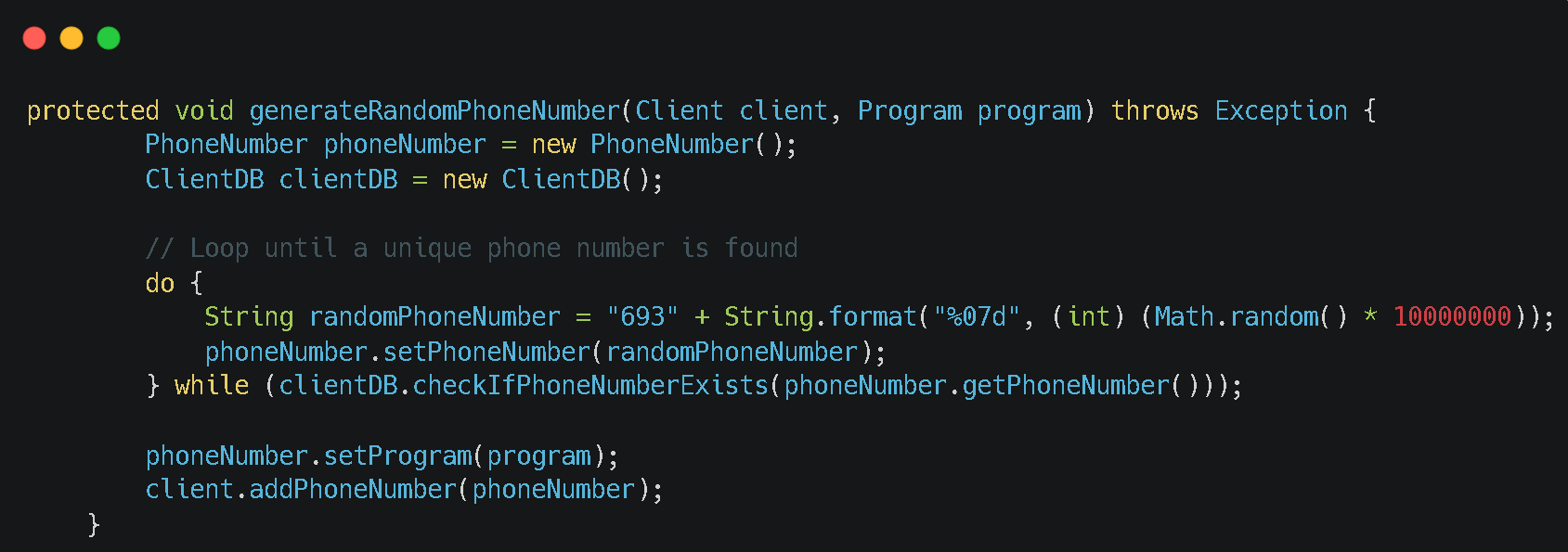


Εικόνα 13: doGet του SellerServlet



Εικόνα 14: doPost του SellerServlet

Τέλος για τα Servlets. Η ένταξη νέου Αριθμού τηλεφώνου γίνεται μόνο με δημιουργία τυχαίου αριθμού τηλεφώνου μέσο της συνάρτησης generateRandomPhoneNumber μέσα στο SellerServlet. Η χρήση τυχαίας γεννήτριας αριθμών κρίνεται αναγκαία, καθώς το PhoneNumber πρέπει να είναι Unique και είδη υπάρχει ένας μεγάλος αριθμός, αριθμών τηλεφώνου στη βάση.

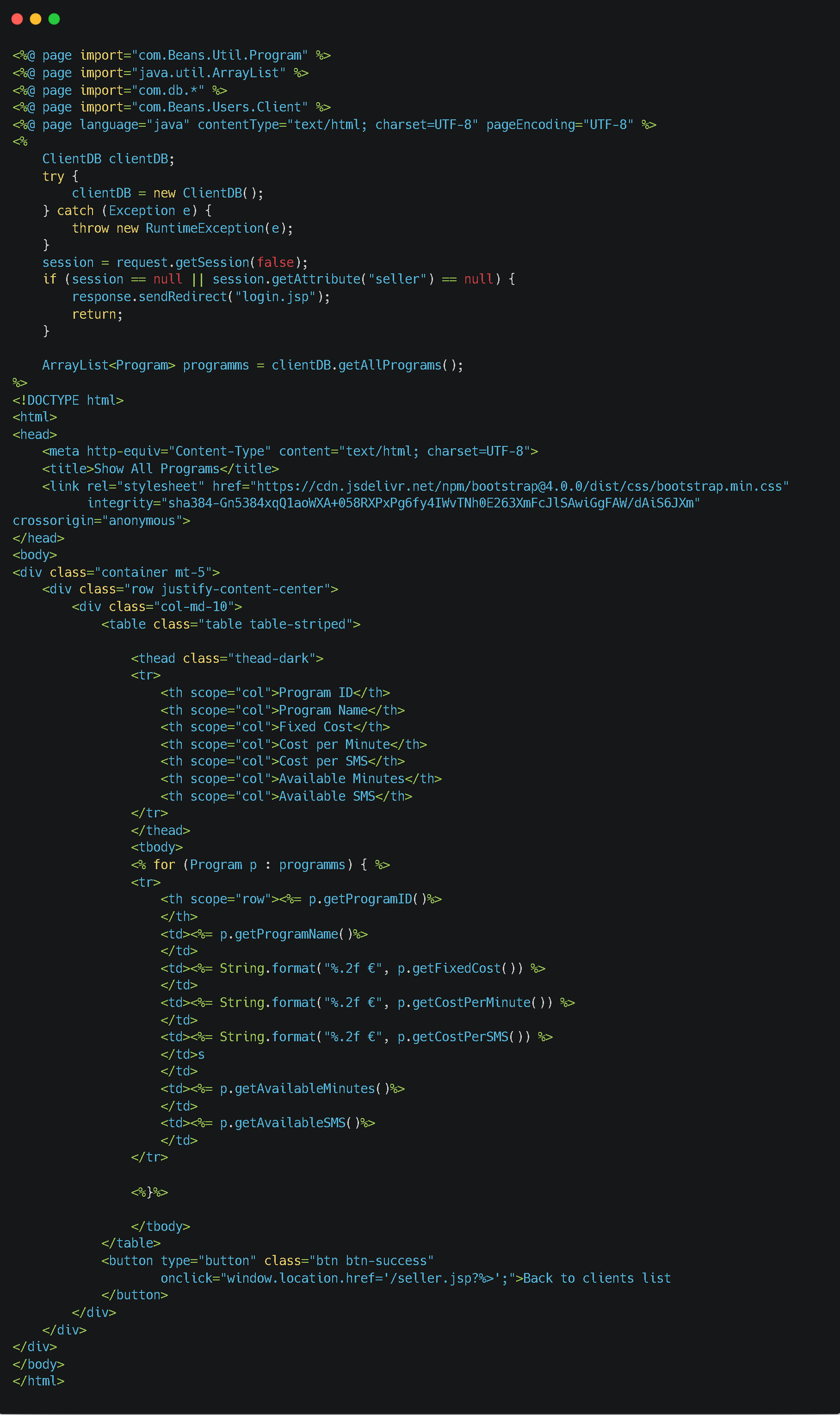


Εικόνα 15: GenerateRandomPhoneNumber

* 1. Front End (jsp)

Τα jsp είναι οι σελίδες που υλοποιούν το user interface. Μέχρι στιγμής έχουν υλοποιηθεί τα index.jsp, που είναι το landing page, το login.jsp που είναι η φόρμα login, το error.jsp, που είναι η σελίδα ανακατεύθυνσης σε περίπτωση λάθους, το seller.jsp, που αποτελέι τη προεπιλεγμένη σελίδα για τους sellers, μέσα στην οποία ο κάθε Seller βλέπει πληροφορίες για όλους τους πελάτες του. Τις addClient.jsp και editClient.jsp για την εισαγωγή και ανανέωση στοιχείων του εκάστοτε πελάτη, και οι οποίες καλούνται από κουμπιά μέσα στη seller.jsp. Τέλος υπάρχει το showAllPrograms.jsp και που αποτελεί ένα πίνακα όλων των προγραμμάτων τηλεφωνίας τα οποία υπάρχουν στο σύστημα.

Παράδειγμα χρήσης jsp με το showAllPrograms.jsp



Εικόνα 16: showAllPrograms.jsp

1. Βιβλιογραφικές Πηγές

* Η βάση δημιουργήθηκε με τη χρήση διαγράμματος μέσο του:  
  [DrawSQL](https://drawsql.app/)
* Για το data population script χρησιμοποιήθηκε η βιβλιοθήκη fake:

[fake repository](https://github.com/joke2k/faker?fbclid=IwAR1sDpaB2Y74bAPTY5AvQGdkt4ir8yy0nI6H-6P8rxS3MsfuPcnmQaGoutY)