

Τεχνολογίες Εφαρμογών Διαδικτύου(ΤΕΔ)

Πλεύρης Κωσταντίνος 1115201400162

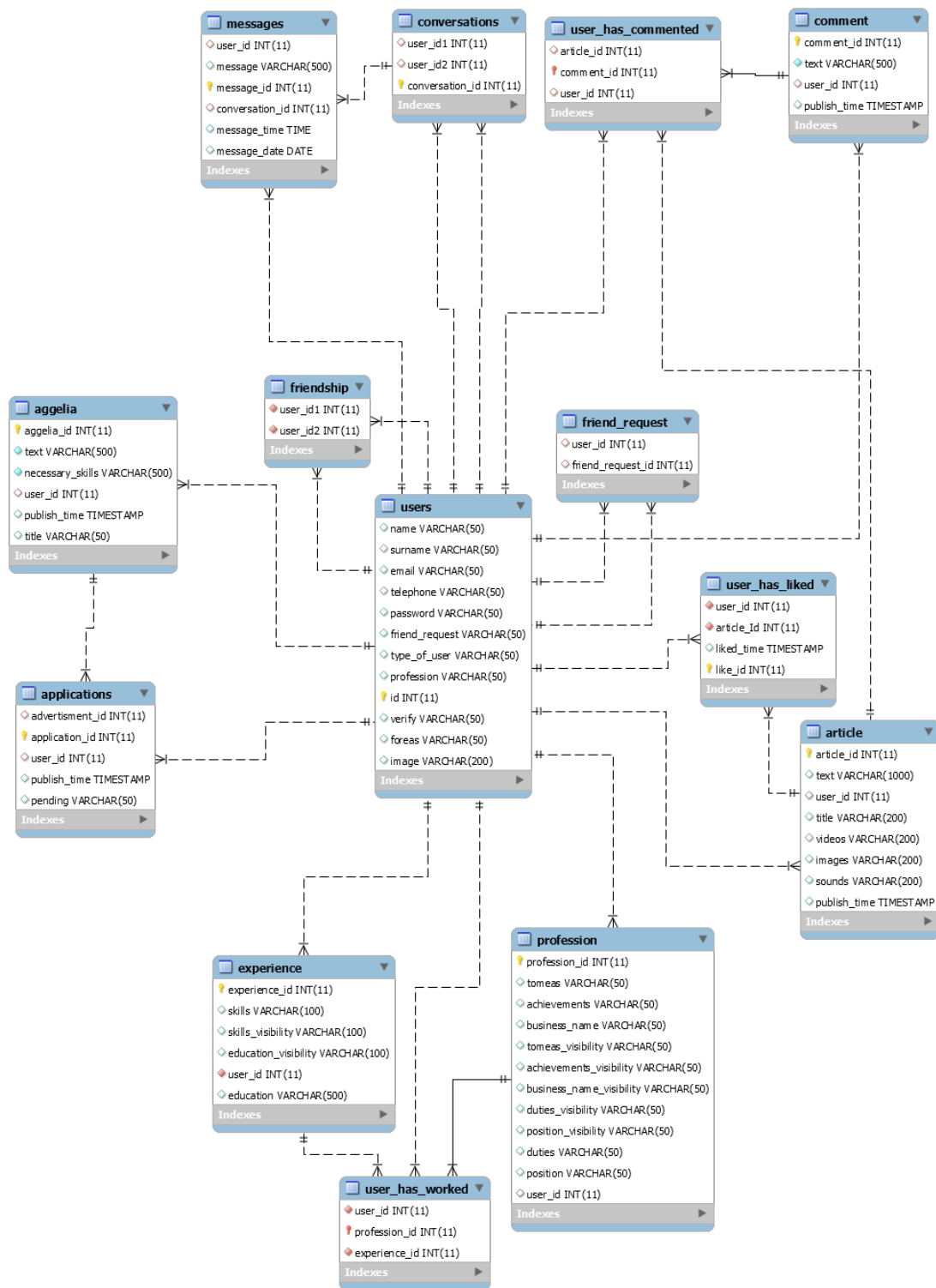
Βέργος Ηλίας 1115201400266

Δομή άσκησης από το Front στο Back End

Το front κομμάτι αποτελείται από τα αρχεία .jsp που κρατούν το html κομμάτι, τα javascript αρχεία καθώς επίσης και τα .css αρχεία που έχουν τις κλάσεις που χρησιμοποιούνται στο html. Όσο αναφορά το front κομμάτι χρησιμοποιήθηκε html, css, AngularJs, JSON, AJAX και κανονική javascript. Το css φτιάχτηκε εξ'ολοκλήρου από εμάς (δεν πήραμε τιποτα έτοιμο από κάποιο framework όπως bootstrap , κ.α.).

Το back κομμάτι αποτελείται από τα Servlets, την Βάση, το JPA και τα Dao.

Το σχημα της βάσης είναι:



Η βάση αποτελείται από 9 οντοτητες

(users,profession,experience,messages,aggelia,applications,article ,conversation,comment).

- 1)**users**: Είναι το table με τα βασικά στοιχεία των χρηστών.
- 2)**profession** :Είναι το table που περιεχει τις πληροφορίες του χρηστη αναφορικά με τις παλιες δουλειες που εχει κανει στο παρελθον.Επισης υπαρχουν και τα πεδια για public και private για κάθε πεδιο.
- 3)**experience**: Είναι το table με την εκπαιδευση και τις ικανοτητες του χρηστη καθως και τα visibility πεδια.
- 4)**messages**: Είναι το table με τα μηνυματα που εχει κανει ο κάθε χρηστης.
- 5)**aggelia**:Είναι το table με τις αγγελιες που εχει δημοσιευσει ο κάθε χρηστης.
- 6)**applications**:Είναι το table με τις αιτησεις που εχει κανει κάθε χρηστης.Τα applications αυτά ,εχουν από default ,pending =True.Αυτο σημαινει ότι όταν ενας χρηστης κανει μια αιτηση για μια αγγελια και περιμενει για απαντηση τοτε εχουμε pending =true.Μολις ο δημιουργος της αγγελιας απαντησει θετικα ή αρνητικα τοτε το pending γινεται false και χρησιμοποιειται μονο για τα στατιστικα που χρειαζομαστε για τον αλγοριθμο εμφανισης αγγελιων.Ετσι τα applications δεν σβηνονται,απλα αλλαζουν από pending =true σε pending =false εκτος και αν ο χρηστης που εκανε την αιτηση πατησει cancel apply ή αν ο διαχειριστης τον διαγραφει.
- 7)**article**:Είναι το table με τις δημοσιευσεις αρθρων.
- 8)**conversation**:Είναι το table που περιλαμβανει τις συνομιλιες των χρηστων.
- 9)**comment**:Είναι το table με τα σχολια των χρηστων σε αρθρα .
- 10)**user_has_liked**: Περιλαμβανει του χρηστες που εχουν like σε καποιο αρθρο καθως και την ωρα που εγινε το like.

Επισης η βαση για την σωστη χρηση της περιλαμβανει και 5 συσχετισεις(friendship,friend_request,user_has worked,user_has commented).

- 1)**friendship**: Περιλαμβανει τους χρηστες και τους φιλους τους και το αντοθετο.Δηλαδη για κάθε χρηστη , η φιλια είναι αμφιδρομη πχ.user1 ->user2,user2->user1 →user1 is friend with user2.
- 2)**friend_request**:Περιλαμβανει τους χρηστες που εκαναν αιτηση φιλιας και αυτους στους οποιους εκαναν την αιτηση φιλιας.
- 3)**user_has_worked**:Η συσχετιση αυτή περιλαμβανει τον χρηστη και ότι εχει να κανει με την επαγγελματικη του καριερα .Δηλαδη τον συνδεει με το table της εκπαιδευσης και των ικανοτητων και με το table με τις προηγουμενες του δουλειες.
- 4)**user_has_commented**:Περιλαμβανει τα σχολια του χρηστη σε αρθρα.Συνδεει τα σχολια με τα αρθρα και τους χρηστες.

Παρατηρήσεις για τις τεχνολογίες που χρησιμοποιήθηκαν:

- 1) Για την μεταφορά δεδομένων από τα Servlet στα jsr χρησιμοποιήθηκε η τεχνολογία JSON. Για την υλοποίηση του json χρησιμοποιήθηκαν κάποιες ετοιμες βιβλιοθηκες που κατεβάσαμε από το διαδίκτυο και βρίσκονται στον καταλογο WebContent/WEB-INF/lib/
- 2) Ο τρόπος με τον οποίο αλληλεπιδρούν servlet και jsr είναι ο εξής: Αρχικά καλούμε στο url το .jsp, ύστερα στο body του jsr υπάρχει μια συνάρτηση που καλείται μέσω angular ng-init=loadPage() , η οποία στέλνει αίτημα get ασύγχρονο με AJAX στο servlet για να πάρει τα απαραίτητα δεδομένα που χρειάζεται η σελίδα για να φορτωθεί σωστά. Στην συνέχεια ανάλογα με την σελίδα μπορεί να σταλθούν και άλλα ασύγχρονα αιτήματα get ή post με AJAX ή με την κλασσική μορφή της html φόρμας. Για την σωστή αποστολή και λήψη στοιχείων από το jsr στο servlet χρησιμοποιούμε κατάλληλες συνθήκες if που μας επιτρέπουν να βλέπουμε ποιες από τα δεδομένα είναι κενά και ποια μεταφέρουν υλικό. (Μπορεί στο servlet να έχουμε συνθήκες από 10 διαφορετικά αιτήματα, για αυτό χρησιμοποιούμε if για να δούμε ποια είναι η σωστή αίτηση που έγινε στον server.)
- 3) Για τα παραγώγη του Xml αρχείου έπρεπε με κάποιον τρόπο στα entities να βάλουμε ποιες από τις μεταβλητές δεν είναι απαραίτητες . Αυτό έγινε χρησιμοποιώντας το annotation @XmlTransient. Για τον ίδιο λόγο , όμως τώρα για το JSON , για να μην επιτρέπουμε την απειρή επανάληψη μεταβλητών χρησιμοποιήσαμε τον annotation @JsonIgnore.
- 4) Για τις σχέσεις μεταξύ των στοιχείων των entities χρησιμοποιήθηκαν τα annotations (@OneToOne, @ManyToOne, @OneToMany, @ManyToMany)
- 5) Για την επικοινωνία servlet και βάσης χρησιμοποιούμε τα Dao και τον Entity manager που μας επιτρέπει μέσω συναρτήσεων να κάνουμε τα κατάλληλα sql ερωτήματα στην βάση και να παίρνουμε τα αποτελέσματα που θέλουμε.

Γενικές Παρατηρήσεις:

- 1) Για την σύνδεση ενός χρήστη στον λογαριασμό του χρησιμοποιούνται cookies .
- 2) Το footer υπάρχει στις βασικές καρτέλες που αναφέρονται στην άσκηση. Περιλαμβάνει το όνομα του site (Business World), τους σχεδιαστές της άσκησης (εμάς τους 2) και 2 σχετικά πεδία που δεν κάνουν τίποτα κλικαροντας τα , απλά υπάρχουν για να γεμίσει το footer.
- 3) Για την εισαγωγή του κωδικού στην βάση και την ανακτήση του χρησιμοποιούμε μια hash function που βρήκαμε έτοιμη στο internet (Hash Function MD5). Αυτό γίνεται γιατί θεωρούμε πως ο κωδικός είναι ένα απροσχητό στοιχείο που ούτε όσοι έχουν πρόσβαση στην βάση πρέπει να ξέρουν.
- 4) Έχουμε φτιάξει κάποιες βοηθητικές συναρτήσεις όπως το ανέβασμα αρχείου στον υπολογιστή, την αποστολή email επιβεβαίωσης, την δημιουργία αντικειμένου json και

καποιες άλλες βοηθητικές κλάσεις για το xml ,και τους αλγορίθμους για τα recommended άρθρα και αγγελίες.

5)Τα στοιχεία των χρηστών από τα άρθρα αλλά και η εικόνα προφίλ τους αποθηκεύονται όλα σε φακέλους μέσα στο workspace της άσκησης.Πιο συγκεκριμένα μέσα στο WebContent/resources/Database/.Εκεί υπάρχει ένα φάκελος με το id κάθε χρήστη που μπορεί να περιλαμβάνει 2 φακέλους max.Ο 1^{ος} φάκελος είναι το images που περιέχει την εικόνα προφίλ του και ο 2^{ος} φάκελος περιέχει τα id από τα άρθρα που πιθανόν να έχει δημοσιεύσει.Μέσα στον φάκελο με τα id των άρθρων υπάρχουν 3 υποφάκελοι max.Αυτοί μπορεί να είναι τα videos ,photos,audios που μπορεί να περιέχει κάθε άρθρο που δημοσιεύει ο χρήστης.Τέλος για να παρούμε τα στοιχεία αυτά από την βάση και να τα χρησιμοποιήσουμε στο site υπάρχουν σχετικά πεδία στην βάση με τα paths των φακέλων αυτών.

Ερώτημα 1 (Υλοποιήθηκε):

Για το 1^ο ερώτημα φτιάξαμε μία σελίδα καλωσορίσματος όπου χρήστης μπορεί να πάρει κάποιες πληροφορίες για το site ,μπορεί επίσης να κάνει εγγραφή πατώντας το αντίστοιχο κουμπί και τέλος μπορεί να κάνει σύνδεση εφόσον πρώτα έχει φτιάξει ένα λογαριασμό και τον έχει επιβεβαιώσει από το email του.

Ερώτημα 2 (Υλοποιήθηκε):

Εχουμε υλοποιήσει την σελίδα εγγραφής όπως μας ζητείται στην άσκηση και έχουμε προσθέσει 2 επιπλέον πεδία(Επαγγελμα και φορέας απασχόλησης) τα οποία όμως σε αντίθεση με τα άλλα που ζητά η άσκηση δεν είναι υποχρεωτικά για τον χρήστη.

Ερώτημα 3 (Υλοποιήθηκε) :

Στην βάση υπάρχει ήδη ένας χρήστης με δικαιώματα διαχειριστή.Τα στοιχεία του για τυχόν δοκιμές είναι τα παρακάτω

username/email:admin@gmail.com

password: 1234

Ερώτημα 4 (Υλοποιήθηκε):

Στην σελίδα διαχείρισης ,υπάρχει μια λίστα με όλους τους χρήστες που έχουν κάνει εγγραφή στο site.Για κάθε χρήστη ο διαχειριστής έχει την δυνατότητα να βλέπει το profile του και τους φίλους αυτού του χρήστη.Επίσης έχει την δυνατότητα να διαγράψει έναν οποιονδήποτε χρήστη ή να εξαγει σε αρχείο XML τα δεδομένα ενός ή και παραπάνω χρηστών.Ο διαχειριστής δεν μπορεί να στείλει μηνύματα στους χρήστες.Για την δημιουργία του XML αρχείου χρησιμοποιούμε μια κλάση(XmlClass.java) που περιέχει τα εξής στοιχεία του χρήστη (ονομα,επίθετο,id,email,τηλέφωνο,δουλειά,φορέας απασχόλησης,τα άρθρα που έχει κάνει like,την εμπειρία του ,τα σχόλια που έχει κάνει σε άλλα άρθρα,τις αγγελίες και τα άρθρα που έχει δημοσιεύσει),και αυτά είναι που εμφανίζονται στο xml αρχείο.Για την δημιουργία του XML χρησιμοποιούμε τις εξής βιβλιοθήκες της

java(javax.xml.bind.JAXBContext, javax.xml.bind.JAXBException, και javax.xml.bind.Marshaller) και ως κλάση για τον marshaller έχουμε φτιάξει μια κλάση (Users.java) που περιέχει μια λίστα από τις παραπάνω κλάσεις (XmlClass.java). Το αρχείο αποθηκεύεται μέσα σε κάποιο φάκελο στον υπολογιστή μας που του έχουμε ορίσει εμείς.

Ερώτημα 5 (Υλοποιήθηκε):

Στην αρχική σελίδα, όπως και σε οποιαδήποτε σελίδα και να πλοηγηθεί ο χρήστης μέσα στον λογαριασμό του πάνω δεξιά θα βρει το κουμπί και έξοδο από τον λογαριασμό του. Στο αριστερό μέρος της αρχικής σελίδας έχουμε τους φίλους μας από πού μπορούμε με ένα κουμπί να στείλουμε μήνυμα σε αυτούς και με ένα άλλο να δούμε το προφίλ τους. Επίσης υπάρχει πεδίο που ο χρήστης μπορεί να συμπληρώσει κάποια από τα στοιχεία του (Δεξιότητες και Εκπαίδευση) ή αν θέλει να υπάρχει κουμπί που τον στέλνει στην σελίδα με τα προσωπικά του στοιχεία για να τα συμπληρώσει εκεί. Στο κεντρικό μέρος υπάρχει κουμπί όπου ο χρήστης πατά και μπορεί να δημιουργήσει καινούργιο άρθρο όπως η άσκηση ζητά (με εικόνα, βίντεο ή ήχο). Επίσης στο κεντρικό μέρος έχει την δυνατότητα να βλέπει τα άρθρα των φίλων του ή ότι αλγόριθμος του ερωτήματος 12 του προτείνει. Σε κάθε άρθρο που βλέπει ο χρήστης μπορεί να κάνει like, comment να δει ποιος το δημοσίευσε και τι ώρα έγινε η δημοσίευση. Για τα βίντεο και ήχο χρησιμοποιήθηκαν τα html5 στοιχεία video και sound.

Ερώτημα 6 (Υλοποιήθηκε):

Στην σελίδα του Δικτύου του χρήστη ο χρήστης μπορεί να δει σε μορφή πλεγματος τους φίλους του. Πατώντας να δει το προφίλ τους μπορεί να τους στείλει μήνυμα και να ξεκινήσουν νέα συνομιλία και ακόμα είναι σε θέση να δει και τους φίλους του φίλου του. Επίσης μπορεί να κάνει αναζήτηση στην μπάρα αναζήτησης για την εύρεση νέων μελών. Η αναζήτηση γίνεται με βάση το όνομα, το επίθετο ή και τα 2 μαζί. Για να γίνει σωστή αναζήτηση θα πρέπει τα στοιχεία που πληκτρολογούνται στην μπάρα αναζήτησης να είναι ίδια με αυτά της βάσης. Από την λίστα με τα αποτελέσματα ο χρήστης μπορεί να δει τα προφίλ των χρηστών, χωρίς όμως να μπορεί να δει τους φίλους του χρήστη αυτού. Αν ο χρήστης αυτός που δεν είναι φίλος μας έχει επιλέξει να έχει όλα τα στοιχεία του ιδιωτικά τότε δεν μπορούμε να δούμε τίποτα άλλο εκτός από το όνομα, το επίθετο του και την εικόνα του.

Ερώτημα 7 (Υλοποιήθηκε):

Στην καρτέλα με τις αγγελίες, έχουμε το κεντρικό τμήμα που περιέχει τις αγγελίες που ο χρήστης μπορεί να εξετάσει και να κάνει αίτηση για την θέση και μπορεί ακόμα να φτιάξει μια δικιά του αγγελία. Επίσης υπάρχει και το αριστερό τμήμα της σελίδας όπου ο χρήστης βλέπει τις αιτήσεις άλλων χρηστών για τις αγγελίες που ο ίδιος έχει αναρτήσει. Εκεί μπορεί να δει το προφίλ του κάθε χρήστη πατώντας πάνω στο όνομα του και ύστερα μπορεί να επιλέξει αν θα δεχτεί ή θα αρνηθεί την αίτηση για την αγγελία που δημοσίευσε. Πατώντας αποδοχή ή απορριψη δεν γίνεται τίποτα στο πρόγραμμα από άποψη λειτουργικότητας, το μόνο που γίνεται είναι μια αλλαγή στην βάση ότι ο χρήστης απάντησε στην αίτηση. Δεν

βγαίνει κάποια ανακοίνωση στον χρήστη ότι η αίτηση που έκανε απορρίφθηκε ή ότι την αποδέχτηκαν. Όσο αναφορά την εμφάνιση των αγγελιών, γίνεται με τον εξής τρόπο: Στο πρώτο κομμάτι εμφανίζουμε όλες τις αγγελίες των φίλων μας που μπορούμε να κάνουμε αίτηση ή έχουμε κάνει αίτηση και περιμένουμε απάντηση. Αν έχει δοθεί απάντηση για μια αίτηση η αγγελία παύει να υπάρχει στο κεντρικό τμήμα των αγγελιών. Το δεύτερο κομμάτι βασίζεται κυρίως στον αλγόριθμο που μας ζητείται. Για την υλοποίηση του αλγορίθμου παίρνουμε όλους τους χρήστες που δεν είναι φίλοι μας και όλες τις αγγελίες που έχουν δημοσιευτεί. Αν ο συνδεδεμένος χρήστης έχει κάνει αίτηση σε κάποια από αυτές τις αγγελίες τότε δίνουμε σκορ σε αυτήν την αγγελία ίσο με 1, όλες οι υπολοιπές παίρνουν 0. Ύστερα παίρνουμε όλες τις αγγελίες και συγκρίνουμε τα skills κάθε αγγελίας με τα δικά μας skills. Αν υπάρχει κοινό skill τότε στο σκορ της αγγελίας προσθέτουμε το 2. Έτσι στο τέλος, κάθε αγγελία που υπάρχει στην βάση έχει ένα σκορ. Εμείς παίρνουμε τις K=5 (τυχαίο νούμερο) αγγελίες με το μεγαλύτερο σκορ. Αυτές οι αγγελίες είναι για εμάς οι similar αγγελίες και με βάση αυτές θα παρούμε τις αγγελίες των χρηστών που δεν είναι φίλοι μας και οποια αγγελία από τις αγγελίες των μη φίλων μας έχει κάποιο κοινό required skill τότε την προτείνουμε στον συνδεδεμένο χρήστη. Επομένως οι αγγελίες που εμφανίζονται στον χρήστη είναι οι προτεινόμενες (που προέρχονται από τους χρήστες που δεν είναι φίλοι μας και λειτουργούν βάση αλγορίθμου) και οι αγγελίες των φίλων μας. Τα skills που βάζει ο χρήστης πρέπει να χωρίζονται με κόμμα(,).

Ερώτημα 8 (Υλοποιήθηκε):

Για την καρτέλα με τις συζητήσεις δεν υπάρχει κάτι το διαφορετικό από αυτό που ζητάει η άσκηση.

Ερώτημα 9 (Υλοποιήθηκε):

Στις καρτέλα με τις ειδοποιήσεις όπως ζητά η άσκηση, στο πάνω μέρος έχουμε τις αιτήσεις φιλίας αν υπάρχουν όπου ο χρήστης μπορεί να δει το προφίλ του χρήστη που του έκανε αίτηση πατώντας στο όνομα του και ύστερα να δεχτεί ή να αρνηθεί την αίτηση φιλίας. Όσο αναφορά το κάτω μέρος, έχουμε τις ειδοποιήσεις για τα άρθρα που έχουμε δημοσιεύσει (πρώτα οι ειδοποιήσεις για τα likes και μετά οι ειδοποιήσεις για τα comments). Πατώντας στην ειδοποίηση για το like που σου έγινε σε πετάει σε μια σελίδα με το άρθρο σου και όλα τα στοιχεία του (τίτλος, κείμενο, συννημένα, likes, comments). Το ίδιο γίνεται και στην περίπτωση των ειδοποιήσεων των σχολίων μόνο που εκεί μπορείς πατώντας στην ειδοποίηση να δεις και ποιο από τα σχόλια που φαίνονται στο άρθρο είναι το σχόλιο που κλικάρεις. Τέλος και για σχόλια και για τα likes φαίνεται με πιο έντονο χρώμα η τελευταία ειδοποίηση.

Ερώτημα 10 (Υλοποιήθηκε):

Στην καρτέλα με τα προσωπικά στοιχεία από default (χωρίς δηλαδή να συμπληρώσει κάτι ο χρήστης) υπάρχει το όνομα και το επίθετο του χρήστη, η εικόνα που έβαλε στην εγγραφή και τα κουμπιά για να συμπληρώσει ότι πληροφορία χρειαστεί. Υπάρχουν 3 κουμπιά, ένα για να συμπληρώσει τα skills του, ένα για να συμπληρώσει την εκπαίδευσή του και ένα για να προσθέτει τυχόν παλιές του επαγγελματικές θέσεις (δουλειά, τομέας

απασχόλησης,ονομα επιχειρησης ,καθηκοντα,επιτευγματα).Το προηγουμενο κουμπι το ονομασαμε 'προσθεσε επαγγελματικη εμπειρια' και θεωρειται οτι προσθετωντας προηγουμενες δουλειες προσθετεις εμπειρια.Αν εχουν συμπληρωθει εκπαιδευση και ικανοτητες μπορεις να τα κανεις edit ανα πασα στιγμη από αυτην την σελιδα όπως επισης και τις προηγουμενες σου δουλειες.Τελος για όλα αυτά τα πεδια που μπορεις να συμπληρωσεις εχεις την δυνατοτητα να επιλεξεις αν θα είναι δημοσια (τα βλεπουν ολοι) ή ιδιωτικα(τα βλεπουν μονο οι φιλοι).Σημειωση:Οσο αναφορα τα προηγουμενα επαγγελματα ,αν ενας χρηστης εχει δηλωσει ως ιδιωτικο το ονομα της δουλειας και όλα τα υπολλειπα δημοσια τοτε δεν εμφανιζεται καν σαν στοιχειο στο dropdown menu με τις δουλειες.

Ερωτημα 11 (Υλοποιηθηκε):

Στις ρυθμισεις ο χρηστης μπορει να αλλαξει κωδικο και email οποτε θελει.

Ερωτημα 12 (Bonus)(Υλοποιηθηκε):

Ο αλγοριθμος του ερωτηματος αυτου αρχικα παιρνει ολους τους χρηστες και όλα τα αρθρα που υπαρχουν στην βαση.Στην συνεχεια για κάθε χρηστη βλεπουμε που εχει κανει like ,comment και αναλογως προσθετουμε στο αντιστοιχο αρθρο έναν αριθμο.Οι αριθμοι αυτοι είναι οι εξης : 0 αν δεν εχει κανει ουτε like ουτε comment , 1 αν εχει κανει comment και όχι like ,2 αν εχει κανει like και όχι comment και 3 αν τα εχει κανει και τα δυο.Αν ο χρηστης δεν μπορει να δει καποια από τα αρθρα επειδη ο χρηστης που το δημοσιευσε δεν είναι φιλος του ειτε γιατι δεν του εχει προταθει ακομα ,τοτε το αντιστοιχο αρθρο παιρνει τιμε 10.Οσο μεγαλυτερη τιμη τοσο πιο μεγαλη θα βγει η ευκλειδια αποσταση και ετσι δεν θα επιλεγει,γιαυτο βαλαμε 10 και όχι 4.Υστερα αφου εχουμε βαθμολογησει κάθε αρθρο με βαση τον κάθε χρηστη,τοτε βρισκουμε την ευκλειδια αποσταση κάθε χρηστη από το συνδεδεμενο χρηστη.Αφου συγκεντρωσουμε ολες τις ευκλειδειες απόστασεις επιλεγουμε τις $k=2$ πιο κοντικες σε εμας ,αυτές δηλαδη που εχουν την μικροτερη τιμη.Ετσι βρισκουμε τους 2 πιο κοντινους μας χρηστες από αποψη like και comment(Τους πιο similar σε εμας δηλαδη).Οταν βρουμε τους similar user παιρνουμε παλι όλα τα αρθρα και με βαση αυτους φτιαχνουμε ένα rating για κάθε αρθρο που θα μας χρησιμευσει στο να βρουμε τα πιο καταλληλα αρθρα για recommendation.Το τυπο αυτό για το rating τον πηραμε από τις διαφανειες στο eclass.Εν συνεχεια αφου εχουμε φτιαξει τα ratings βρισκουμε τα $k=3$ καλυτερα και τα εμφανιζουμε στην αρχικη σελιδα του χρηστη μαζί με τα αρθρα των φιλων μας.Σε περιπτωση που καποιος δεν εχει φιλους εμφανιζονται στην αρχικη μας σελιδα μονο τα δικα μας αρθρα.Αλλιως αν εχουμε φιλους δεν εμφανιζονται τα δικα μας αρθρα ,μονο των φιλων μας και τα recommended.