Έντυπο Προδιαγραφών Απαιτήσεων Λογισμικού

για το Έργο

Διαχείριση Εταιρικού Προσωπικού

Από

Εμμανουήλ Βαρέσης 40270

Αναστασία Παντέλογλου 40197

Γιώργος Λιαπούρης 39284

Χειμερινό Εξάμηνο 2013 - 2014

Πίνακας Περιεχομένων

1. Εισαγωγή 1

1.1 Σκοπός του κειμένου 1

1.2 Συμβάσεις του κειμένου 1

1.3 Στόχος και εύρος του προϊόντος 1

1.4 Αναφορές 1

2. Γενική Περιγραφή 2

2.1 Προέλευση και προοπτική του προϊόντος 2

2.2 Γενική περιγραφή των λειτουργιών του προϊόντος 2

2.3 Κατηγορίες χρηστών 2

2.4 Περιβάλλον λειτουργίας 2

2.5 Περιορισμοί στη σχεδίαση και την υλοποίηση 2

2.6 Τεκμηρίωση για το χρήστη 3

3. Απαιτήσεις για τις εξωτερικές διεπαφές 4

3.1 Διεπαφή χρήστη 4

4. Λειτουργίες συστήματος 5

4.1 Λειτουργίες εξουσιοδοτημένου χρήστη 5

4.2 Λειτουργίες διαχειριστή 6

5. Μη λειτουργικές απαιτήσεις 7

5.1 Επιδόσεις 7

5.2 Ασφάλεια πληροφορίας 7

5.3 Ποιότητα λογισμικού 7

6. Διαγράμματα UML 8

6.1 Διάγραμμα Περιπτώσεων Χρήσης (Διαχειριστής – Υπεύθυνος Προσωπικού) 8

6.2 Διάγραμμα Περιπτώσεων Λειτουργίας 9

6.3 Διάγραμμα Κλάσεων 10

6.4 Διάγραμμα Ακολουθίας 11

6.5 Διάγραμμα Δραστηριοτήτων Διαχειριστή 12

6.6 Διάγραμμα Δραστηριοτήτων Υπεύθυνου Προσωπικού 13

7. User Manual 14

7.1 Σύνδεση και Αρχική Οθόνη 14

7.2 Προφίλ Υπαλλήλου 25

8. Παράρτημα Α: Κώδικας Προγράμματος 30

8.1 Κλάση Main.class 30

8.2 Κλάση Login.class 30

8.3 Κλάση MainFrame.class 33

8.4 Κλάση Profile.class 41

8.5 Κλάση Profile.class 59

8.6 Κλάση Income\_Outcome.class 62

8.7 Κλάση Income\_Outcome.class 65

9. Παράρτημα Β: Επιπλέον Στοιχεία 71

# Εισαγωγή

## Σκοπός του κειμένου

Οι προδιαγραφές απαιτήσεων λογισμικού περιγράφουν τις λειτουργικές και μη λειτουργικές απαιτήσεις για το έργο **διαχείριση προσωπικού εταιρείας**. Το παρών έγγραφο έχει σκοπό να ενημερώσει για τη λειτουργία του προγράμματος και να και να καταστήσει εφικτή την σωστή χρήση του.

## Συμβάσεις του κειμένου

Το κείμενο είναι γραμμένο με γραμματοσειρά Arial.

Οι κύριες επικεφαλίδες έχουν μέγεθος γραμματοσειράς 18 και έντονη γραφή. Οι δευτερεύουσες έχουν μέγεθος γραμματοσειράς 14 και έντονη γραφή και το υπόλοιπο κείμενο έχει μέγεθος γραμματοσειράς 12 και απλή γραφή.

## Στόχος και εύρος του προϊόντος

Με την ανάπτυξη και λειτουργία του συγκεκριμένου λογισμικού προσπαθούμε να δημιουργήσουμε έναν εύκολο και χρηστικό τρόπο για τη διαχείριση του προσωπικού μιας εταιρίας∙ τη δημιουργία νέων υπαλλήλων και τμημάτων, την τροποποίηση αυτών καθώς και την διαγραφή τους.

## Αναφορές

Το πρόγραμμα, πέρα από τη διαχείριση του εταιρικού προσωπικού κάνει αναφορές στο καταστατικό της εταιρείας και στο λογιστικό βιβλίο της.

# Γενική Περιγραφή

## Προέλευση και προοπτική του προϊόντος

Το συγκεκριμένο λογισμικό είναι καινούργιο και ανεξάρτητο και δεν αποτελεί συνέχεια ή αντικατάσταση ενός ήδη υπάρχον. Αναπτύσσεται για αποκλειστική χρήση για την διαχείριση του προσωπικού και των τμημάτων μιας εταιρίας.

## Γενική περιγραφή των λειτουργιών του προϊόντος

Το λογισμικό θα χρησιμοποιείται από τους Υπεύθυνους Προσωπικού (HR) και τον Διαχειριστή (admin). Η πρόσβαση για τους υπαλλήλους και τον διαχειριστή θα ελέγχεται μέσω ενός username και ενός password που θα χρησιμοποιούν.

## Κατηγορίες χρηστών

|  |  |
| --- | --- |
| **Διαχειριστής** | Ο διαχειριστής έχει δικαίωμα να εκτελέσει όλες τις λειτουργίες του λογισμικού. Αναλυτικότερα:   * Δημιουργία / Επεξεργασία / Διαγραφή ενός τμήματος. * Δημιουργία / Επεξεργασία / Διαγραφή / Αναζήτηση / Αξιολόγηση ενός υπαλλήλου. |
| **Υπεύθυνος Προσωπικού** | Οι υπεύθυνοι προσωπικού έχουν λιγότερα δικαιώματα πάνω στις λειτουργίες του λογισμικού. Αναλυτικότερα:   * Δημιουργία / Επεξεργασία / Διαγραφή ενός τμήματος. * Δημιουργία / Επεξεργασία / Διαγραφή / Αναζήτηση ενός υπαλλήλου. |

## Περιβάλλον λειτουργίας

Το περιβάλλον του λογισμικού θα πρέπει να είναι λιτό αφού οι Η/Υ είναι χαμηλής επεξεργαστικής ισχύος.

* Η εφαρμογή θα τρέχει σε Microsoft Windows XP.
* O κώδικας του λογισμικού είναι υλοποιημένος σε γλώσσα προγραμματισμού Java.

## Περιορισμοί στη σχεδίαση και την υλοποίηση

Η ταχύτητα του δικτύου περιορίζεται στα 1GBps.

## Τεκμηρίωση για το χρήστη

Οι τεκμηριώσεις που θα δοθούν προκειμένου να διασφαλισθεί η σωστή χρήση του προγράμματος είναι:

* Ένα user manual για τους Διαχειριστές το οποίο βρίσκεται στις επόμενες σελίδες του εγγράφου αυτού.
* Ένα σεμινάριο για τους Υπεύθυνους Προσωπικού με αναλυτικές διαφάνειες.

# Απαιτήσεις για τις εξωτερικές διεπαφές

## Διεπαφή χρήστη

Η διεπαφή του προγράμματος με το χρήστη έχει ως βασική βάση τις καρτέλες. Στα πλαίσια ανάπτυξης του προγράμματος προσθέσαμε και ορισμένα κουμπιά προκειμένου το πρόγραμμα να γίνει πιο φιλικό στο χρήστη.

# Λειτουργίες συστήματος

## Λειτουργίες εξουσιοδοτημένου χρήστη

**4.1.1 Περιγραφή και προτεραιότητα**

Η αρχική οθόνη του προγράμματος με το που ανοίγει η εφαρμογή.

Προτεραιότητα = Υψηλή

**4.1.2 Ακολουθίες ερεθίσματος / απόκρισης**

Ερέθισμα: Εισαγωγή ατομικών στοιχείων πρόσβασης

Απόκριση: Εμφάνιση αρχικής οθόνης εφόσον είναι σωστά.

Ερέθισμα: Επιλογή του ονόματός του

Απόκριση: Εμφάνιση του προφίλ του με τις καρτέλες περιήγησης και με τα κατάλληλα κουμπιά επεξεργασίας προφίλ.

Ερέθισμα: Πάτημα κουμπιού «Δημιουργία Προφίλ»

Απόκριση: Εμφάνιση κενού προφίλ με δυνατότητα εισαγωγής νέων στοιχείων και ακύρωσης αυτών.

Ερέθισμα: Πληκτρολόγηση ονόματος υπαλλήλου στην Αναζήτηση και πάτημα του κατάλληλου κουμπιού.

Απόκριση: Εμφάνιση λίστας των εργαζομένων που πληρούν τα στοιχεία αναζήτησης.

Ερέθισμα: Διπλό κλικ πάνω σε ένα όνομα στη λίστα.

Απόκριση: Εμφάνιση του προφίλ του με τις καρτέλες περιήγησης και με τα κατάλληλα κουμπιά επεξεργασίας προφίλ.

Ερέθισμα: Ερέθισμα: Πάτημα κουμπιού «Τμήματα Εταιρείας».

Απόκριση: Εμφάνιση λίστας των τμημάτων της εταιρείας καθώς και τα κατάλληλα κουμπιά επεξεργασίας αυτών.

Ερέθισμα: Μονό κλικ πάνω σε ένα τμήμα στη λίστα.

Απόκριση: Δυνατότητα Διαγραφής του.

Ερέθισμα: Διπλό κλίκ πάνω σε ένα τμήμα στη λίστα.

Ερέθισμα: Εμφάνιση παραθύρου με τους εργαζομένους που υπάρχουν στο τμήμα αυτό και δυνατότητα επεξεργασίας τους.

Ερέθισμα: Πάτημα του κουμπιού «Δημιουργία Τμήματος»

Απόκριση: Ερέθισμα: Εμφάνιση παραθύρου με τους εργαζομένους που δεν ανήκουν σε κάποιο τμήμα και δυνατότητα επεξεργασίας τους.

Ερέθισμα: Πάτημα του κουμπιού «Καταστατικό Εταιρείας»

Απόκριση: Άνοιγμα αρχείου .txt με το καταστατικό της εταιρείας.

Ερέθισμα: Πάτημα του κουμπιού «Έσοδα - Έξοδα»

Απόκριση: Ερέθισμα: Εμφάνιση παραθύρου με τη δυνατότητα εκχώρησης εσόδων – εξόδων ανάλογα το μήνα και τη χρονολογία και η αποθήκευση αυτών σε ένα αρχείο .txt.

Ερέθισμα: Πάτημα του κουμπιού «Έξοδος».

Απόκριση: Κλείσιμο προγράμματος.

## Λειτουργίες διαχειριστή

**4.1.1 Περιγραφή και προτεραιότητα**

Η αρχική οθόνη του προγράμματος με το που ανοίγει η εφαρμογή.

Προτεραιότητα = Υψηλή

**4.1.2 Ακολουθίες ερεθίσματος / απόκρισης**

Ερέθισμα: Εισαγωγή ατομικών στοιχείων πρόσβασης

Απόκριση: Εμφάνιση αρχικής οθόνης εφόσον είναι σωστά.

Ερέθισμα: Επιλογή του ονόματός του

Απόκριση: Εμφάνιση του προφίλ του με τις καρτέλες περιήγησης και με τα κατάλληλα κουμπιά επεξεργασίας προφίλ.

Ερέθισμα: Κλικ στην καρτέλα Αξιολόγηση σε κατάσταση επεξεργασίας προφίλ.

Απόκριση: Δυνατότητα εισαγωγής, επεξεργασίας και διαγραφής αξιολόγησης.

Ερέθισμα: Πάτημα κουμπιού «Δημιουργία Προφίλ»

Απόκριση: Εμφάνιση κενού προφίλ με δυνατότητα εισαγωγής νέων στοιχείων και ακύρωσης αυτών.

Ερέθισμα: Πληκτρολόγηση ονόματος υπαλλήλου στην Αναζήτηση και πάτημα του κατάλληλου κουμπιού.

Απόκριση: Εμφάνιση λίστας των εργαζομένων που πληρούν τα στοιχεία αναζήτησης.

Ερέθισμα: Διπλό κλικ πάνω σε ένα όνομα στη λίστα.

Απόκριση: Εμφάνιση του προφίλ του με τις καρτέλες περιήγησης και με τα κατάλληλα κουμπιά επεξεργασίας προφίλ.

Ερέθισμα: Ερέθισμα: Πάτημα κουμπιού «Τμήματα Εταιρείας».

Απόκριση: Εμφάνιση λίστας των τμημάτων της εταιρείας καθώς και τα κατάλληλα κουμπιά επεξεργασίας αυτών.

Ερέθισμα: Μονό κλικ πάνω σε ένα τμήμα στη λίστα.

Απόκριση: Δυνατότητα Διαγραφής του.

Ερέθισμα: Διπλό κλίκ πάνω σε ένα τμήμα στη λίστα.

Ερέθισμα: Εμφάνιση παραθύρου με τους εργαζομένους που υπάρχουν στο τμήμα αυτό και δυνατότητα επεξεργασίας τους.

Ερέθισμα: Πάτημα του κουμπιού «Δημιουργία Τμήματος»

Απόκριση: Ερέθισμα: Εμφάνιση παραθύρου με τους εργαζομένους που δεν ανήκουν σε κάποιο τμήμα και δυνατότητα επεξεργασίας τους.

Ερέθισμα: Πάτημα του κουμπιού «Καταστατικό Εταιρείας»

Απόκριση: Άνοιγμα αρχείου .txt με το καταστατικό της εταιρείας.

Ερέθισμα: Πάτημα του κουμπιού «Έσοδα - Έξοδα»

Απόκριση: Ερέθισμα: Εμφάνιση παραθύρου με τη δυνατότητα εκχώρησης εσόδων – εξόδων ανάλογα το μήνα και τη χρονολογία και η αποθήκευση αυτών σε ένα αρχείο .txt.

Ερέθισμα: Πάτημα του κουμπιού «Έξοδος».

Απόκριση: Κλείσιμο προγράμματος.



# Μη λειτουργικές απαιτήσεις

## Επιδόσεις

Κάθε λειτουργία του συστήματος ελέγχεται και διαπιστώθηκε ότι εκτελείται σε χρόνο λιγότερο από 10 sec.

## Ασφάλεια πληροφορίας

Για την μετάδοση ευαίσθητων δεδομένων στο διαδίκτυο θα χρησιμοποιηθεί το πρωτόκολλο SSL(Secure Sockets Layer).

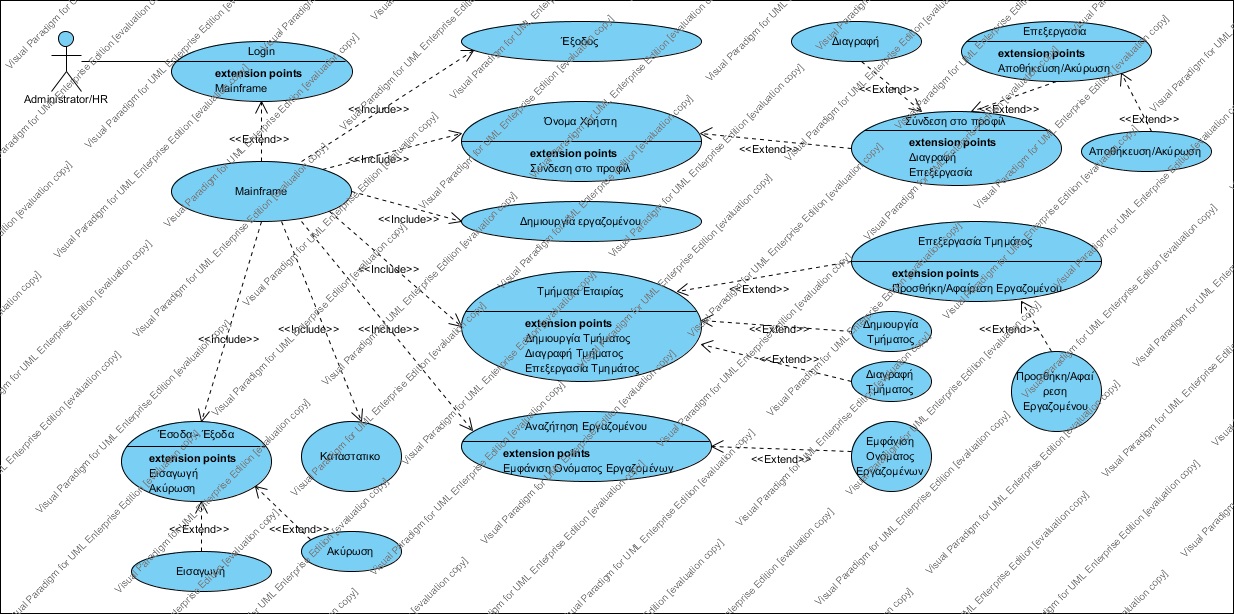
Καθώς επίσης θα υπάρχει και Data protection act 1992.

## Ποιότητα λογισμικού

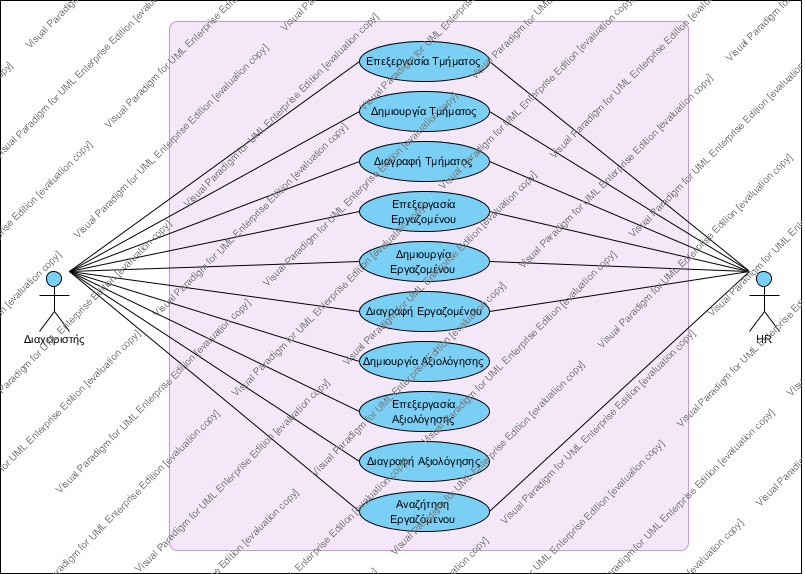
Η ποιότητα του λογισμικού ως προς το χρόνο και την εκτέλεση ελέγχθηκαν από unit testings.

# Διαγράμματα UML

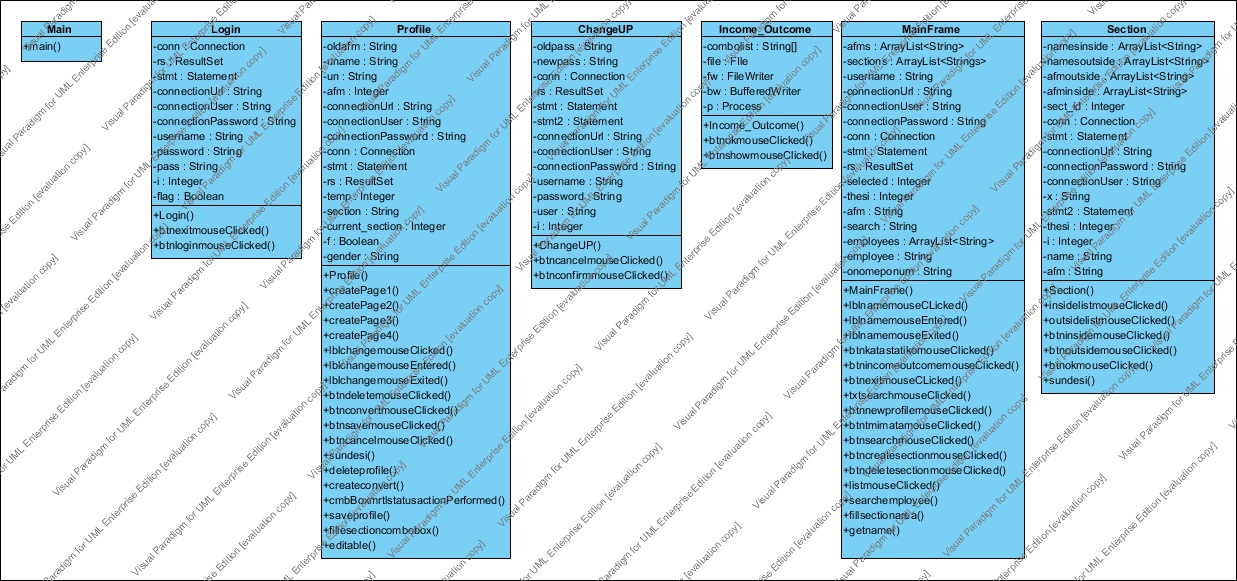
## Διάγραμμα Περιπτώσεων Χρήσης (Διαχειριστής – Υπεύθυνος Προσωπικού)



## Διάγραμμα Περιπτώσεων Λειτουργίας



## Διάγραμμα Κλάσεων



## connection diagram.jpgΔιάγραμμα Ακολουθίας

## activity diagram admin.jpgΔιάγραμμα Δραστηριοτήτων Διαχειριστή

## activity diagram hr.jpgΔιάγραμμα Δραστηριοτήτων Υπεύθυνου Προσωπικού

# User Manual

## Σύνδεση και Αρχική Οθόνη

Στις επόμενες σελίδες ακολουθεί μια περιγραφή της εφαρμογής με τη χρήση εικόνων.

Έχουν δημιουργηθεί ορισμένοι εικονικοί χρήστες, με σκοπό την κατανόηση της λειτουργίας του προγράμματος.

Τα usernames και τα passwords που έχουν χρησιμοποιηθεί είναι τα ακόλουθα:

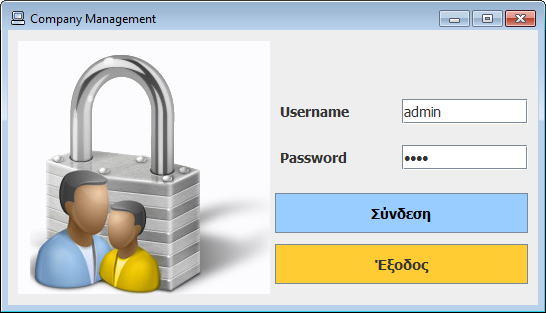
Administrator: username: admin password: 1234

HR1: username: user1 password: pass1

HR2: username: user2 password: pass2

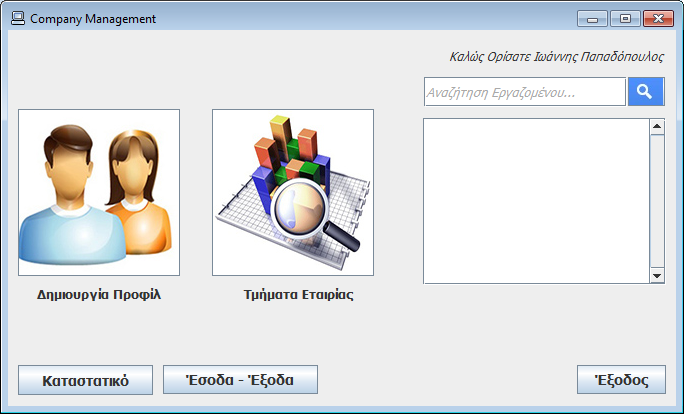
HR3: username: user3 password: pass3

Με τα παραπάνω στοιχεία ο χρήστης μπορεί να κάνει είσοδο στο πρόγραμμα πιέζοντας το κουμπί **Σύνδεση**



Διαφορετικά, έχει τη δυνατότητα να πατήσει το κουμπί **Έξοδος** και να εξέλθει από το πρόγραμμα.

Η αρχική οθόνη μετά την είσοδο στο πρόγραμμα:



Σε αυτό το σημείο του δίνεται η δυνατότητα να επιλέξει το τι θέλει να κάνει.

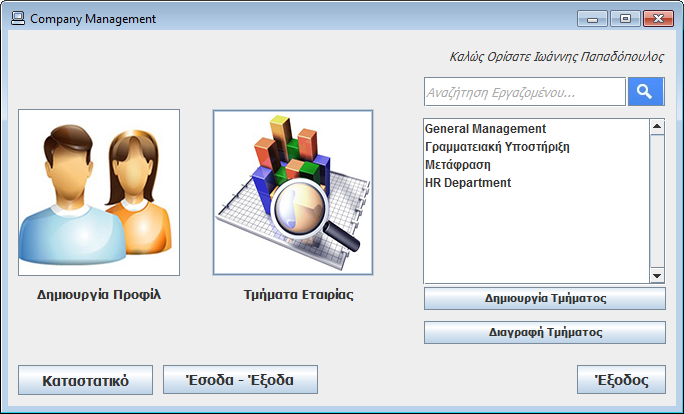
**Έχει επτά επιλογές**:

1. Να δημιουργήσει το προφίλ ενός εργαζομένου, πατώντας το κουμπί «Δημιουργία προφίλ».

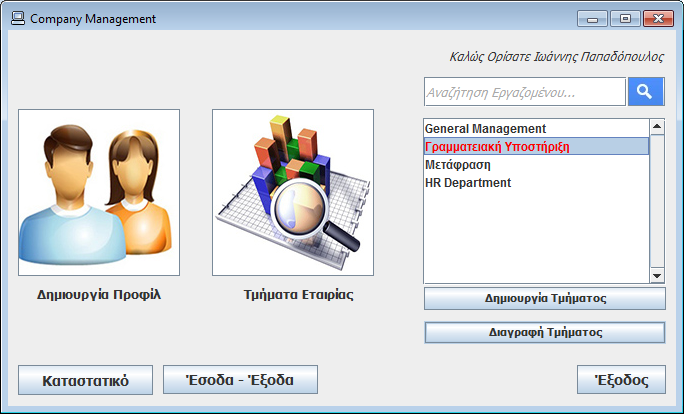
|  |
| --- |
|  |

*Σημείωση:* Η επιβεβαίωση της δημιουργίας του προφίλ ενός νέου υπαλλήλου πραγματοποιείται με το πάτημα του κουμπιού «Αποθήκευση Προφίλ». Επιπλέον, αν δεν είναι επιθυμητή η αποθήκευση του νέου προφίλ, δίνεται η δυνατότητα ακύρωσης, με το πάτημα του κουμπιού «Ακύρωση».

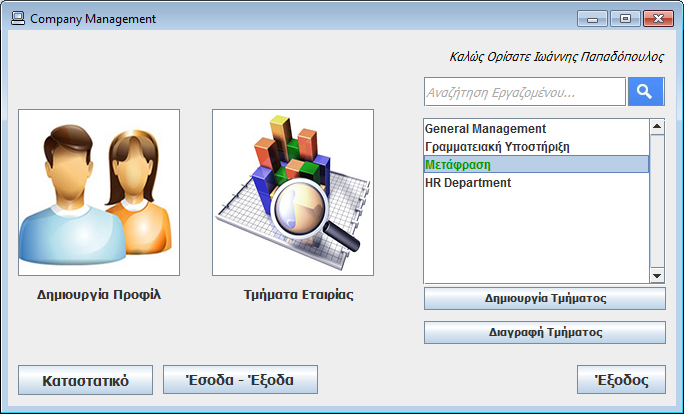
1. Να επιλέξει το κουμπί «Τμήματα Εταιρίας», όπου θα του εμφανιστεί μια λίστα τμημάτων και η δυνατότητα επεξεργασίας αυτών.



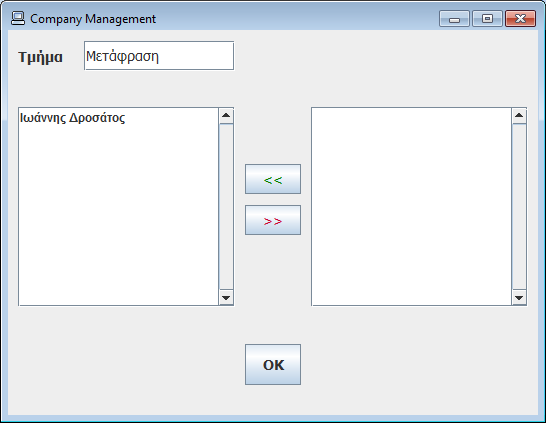
Αν επιλέξει κάποιο από τα τμήματα (με μονό click), τότε εκείνο θα γίνει κόκκινο και θα έχει τη δυνατότητα ο χρήστης να το διαγράψει.



Αν επιλέξει κάποιο τμήμα (με διπλό click) τότε η επιλογή θα γίνει πράσινη.



και θα του δίνεται η δυνατότητα αλλαγής ονόματος και χειρισμού υπαλλήλων εντός του τμήματος αυτού.



Στην αριστερή λίστα εμφανίζονται οι υπάλληλοι του τμήματος.

Αν ο χρήστης επιλέξει το όνομα ενός υπαλλήλου και πιέσει το κουμπί με το κόκκινο διπλό βέλος, το όνομα του υπαλλήλου μεταφέρεται αυτόματα στη δεξιά λίστα και ο υπάλληλος μένει εκτός τμήματος.

|  |
| --- |
|  |

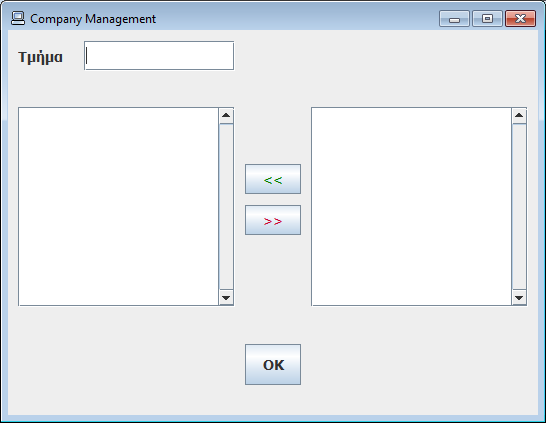
Στη δεξιά λίστα εμφανίζονται οι υπάλληλοι που βρίσκονται εκτός τμήματος.

Αν ο χρήστης επιλέξει το όνομα ενός υπαλλήλου και πιέσει το κουμπί με το πράσινο διπλό βέλος, το όνομα του υπαλλήλου θα μεταφερθεί αυτόματα στην αριστερή λίστα. Αυτό σημαίνει πως ο υπάλληλος θα βρίσκεται στο τμήμα, που θα αναγράφεται στο αριστερό μέρος.

|  |
| --- |
|  |

Στη συνέχεια πιέζοντας το κουμπί OK απλά κατοχυρώνει την επιλογή του.

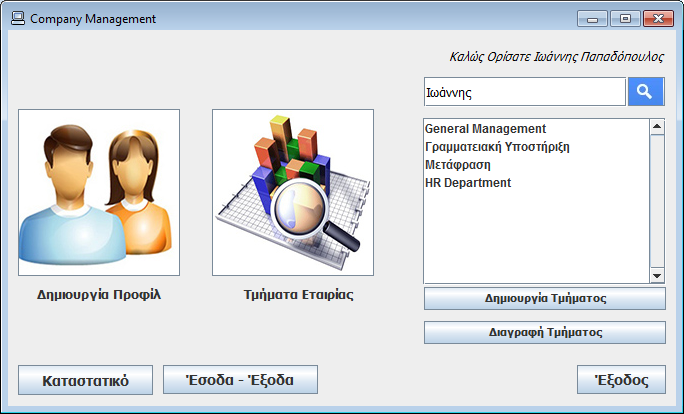
Αν επιλέξει να πατήσει το κουμπί «Δημιουργία τμήματος», τότε θα εμφανιστεί η ακόλουθη οθόνη

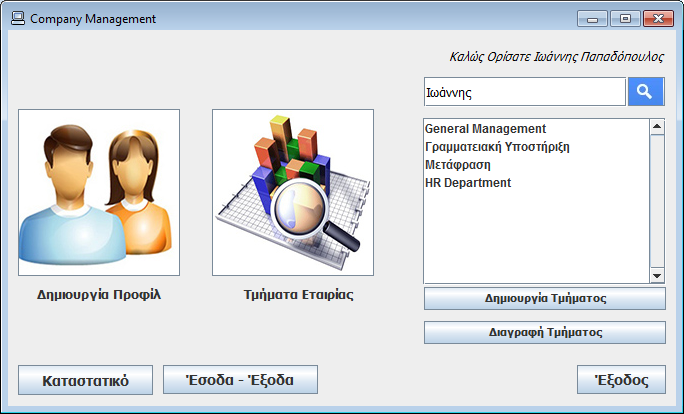


Στο σημείο αυτό, ο χρήστης θα πρέπει να τοποθετήσει τους υπαλλήλους που επιθυμεί στο τμήμα και στη συνέχεια να δώσει ένα όνομα στο τμήμα που δημιούργησε. Έπειτα, πρέπει να πιέσει το κουμπί OK, προκειμένου να κατοχυρώσει το νέο τμήμα.

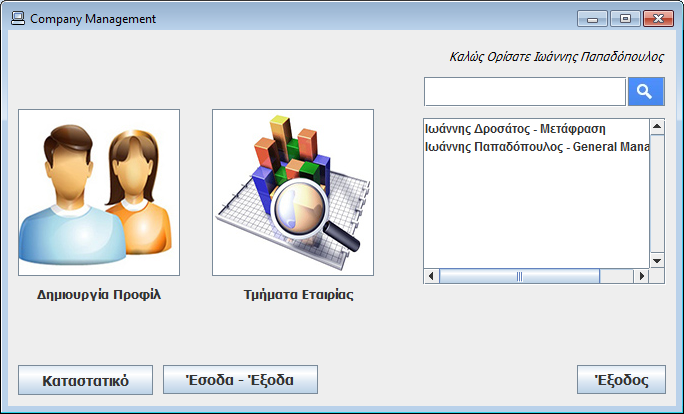
**Προσοχή!**

Δεν πρέπει να χρησιμοποιηθεί άσκοπα το κουμπί δημιουργίας τμήματος.

1. Να επιλέξει να κάνει αναζήτηση κάποιου υπαλλήλου, γράφοντας το όνομα, το επώνυμο, το ονοματεπώνυμο ή το ΑΦΜ του και πιέζοντας το κουμπί αναζήτησης 



Έπειτα από αυτή την ενέργεια θα έχουμε το παρακάτω αποτέλεσμα:

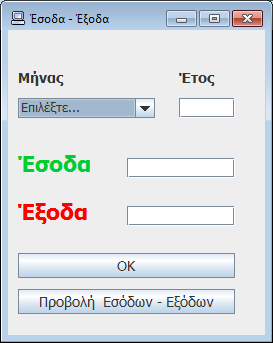


Όπου επιλέγοντας ένα όνομα και κάνοντας διπλό click εμφανίζεται το προφίλ του υπαλλήλου που έχει επιλεγεί.

|  |
| --- |
|  |

1. Να πατήσει το κουμπί «Καταστατικό», προκριμένου να μπορέσει να διαβάσει το καταστατικό της εταιρείας.
2. Να πατήσει το κουμπί «Έσοδα - Έξοδα» προκειμένου να μπορέσει να γράψει και να αποθηκεύσει τα έσοδα και τα έξοδα της εταιρίας.

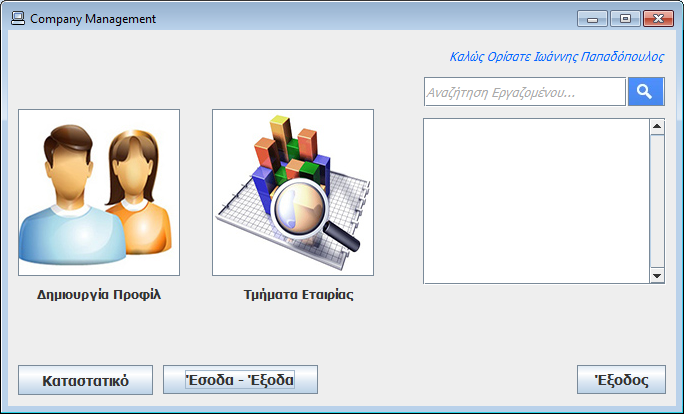
Πιέζοντας το κουμπί «Έσοδα - Έξοδα», ο χρήστης θα πρέπει να έχει την παρακάτω εικόνα:



Όπου θα μπορεί να επιλέξει μήνα, να πληκτρολογήσει το επιθυμητό έτος, καθώς και τα έσοδα και τα έξοδα του μήνα.

Στη συνέχεια με την επιλογή OK αποθηκεύει όσα έχει γράψει, σε ένα αρχείο, το οποίο μπορεί να εμφανίσει πιέζοντας το κουμπί «Προβολή Εσόδων - Εξόδων».

1. Να επιλέξει να κάνει προβολή του προφίλ του πατώντας επάνω στο όνομά του, το οποίο αλλάζει χρώμα με την είσοδο του κέρσορα και επανέρχεται στο αρχικό του χρώμα με την έξοδο αυτού.



Το αποτέλεσμα είναι το ακόλουθο:

|  |
| --- |
|  |

1. Να επιλέξει το κουμπί «Έξοδος», απενεργοποιώντας έτσι το πρόγραμμα.

## Προφίλ Υπαλλήλου

Επιλέγοντας ο χρήστης να κάνει είσοδο στο δικό του προφίλ ή στο προφίλ κάποιου άλλου υπαλλήλου της εταιρίας έχει τις ακόλουθες επιλογές:

1. Να δει το προφίλ του χρήστη

|  |
| --- |
|  |

1. Να κάνει επεξεργασία του προφίλ του υπαλλήλου που βλέπει

|  |
| --- |
|  |

Αφού επεξεργαστεί το προφίλ, ο χρήστης έχει τη δυνατότητα είτε να αποθηκεύσει τις μετατροπές που έχει πραγματοποιήσει, πιέζοντας το κουμπί «Αποθήκευση Προφίλ», είτε να πατήσει το κουμπί «Ακύρωση» και να επαναφέρει το προφίλ στην προηγούμενη κατάστασή του.

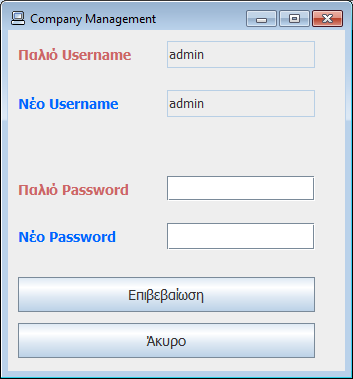
1. Να διαγράψει το προφίλ του υπαλλήλου που βλέπει

|  |
| --- |
|  |

1. Να αλλάξει τα προσωπικά του στοιχεία σύνδεσης (username & password)

|  |
| --- |
|  |

Πατώντας στο σημείο «Αλλαγή username και password», θα πρέπει να εμφανίζεται η ακόλουθη εικόνα στο χρήστη:



Στο σημείο αυτό δίνεται η δυνατότητα στο χρήστη να αλλάξει το προσωπικό του username και password.

**Προσοχή!**

Στον administrator δε δίνεται η δυνατότητα αλλαγής username, παρά μόνο αλλαγής password.

Για να πραγματοποιηθεί αποθήκευση των αλλαγών, πρέπει ο χρήστης να συμπληρώσει όλα τα πεδία τιμών και να πατήσει το κουμπί «Επιβεβαίωση». Διαφορετικά, έχει τη δυνατότητα ακύρωσης πιέζοντας το κουμπί «Άκυρο» και διατηρώντας κατ’ αυτόν τον τρόπο το παλιό username και password.

# Παράρτημα Α: Κώδικας Προγράμματος

## Κλάση Main.class

**package** omada15.softeng.teipir;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Login lg = **new** Login();

lg.setVisible(**true**);

}

}

## Κλάση Login.class

**package** omada15.softeng.teipir;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.border.EmptyBorder;

**import** java.awt.Toolkit;

**import** javax.swing.JLabel;

**import** java.awt.Font;

**import** javax.swing.JOptionPane;

**import** javax.swing.SwingConstants;

**import** javax.swing.JTextField;

**import** javax.swing.JPasswordField;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** java.awt.Color;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.sql.\*;

**public** **class** Login **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField txtusername;

**private** JPasswordField txtpassword;

**private** JLabel lblusername;

**private** JLabel lblpassword;

**private** JLabel lblimage;

**private** JButton btnexit;

**private** JButton btnlogin;

**public** Login() {

setTitle("Company Management");

setIconImage(Toolkit.*getDefaultToolkit*().getImage(Login.**class**.getResource("/com/sun/java/swing/plaf/windows/icons/Computer.gif")));

setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

setBounds(100, 100, 546, 313);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

lblusername = **new** JLabel("Username");

lblusername.setHorizontalAlignment(SwingConstants.*LEFT*);

lblusername.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblusername.setBounds(272, 69, 112, 25);

contentPane.add(lblusername);

lblpassword = **new** JLabel("Password");

lblpassword.setHorizontalAlignment(SwingConstants.*LEFT*);

lblpassword.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblpassword.setBounds(272, 115, 112, 25);

contentPane.add(lblpassword);

txtusername = **new** JTextField();

txtusername.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

txtusername.setBounds(394, 69, 126, 25);

contentPane.add(txtusername);

txtusername.setColumns(10);

lblimage = **new** JLabel();

lblimage.setIcon(**new** ImageIcon(Login.**class**.getResource("/omada15/softeng/teipir/login.jpg")));

lblimage.setBounds(10, 11, 252, 253);

contentPane.add(lblimage);

btnexit = **new** JButton("Έξοδος");

btnexit.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

System.*exit*(0);

}

});

btnexit.setBackground(**new** Color(255, 204, 51));

btnexit.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

btnexit.setBounds(267, 214, 253, 40);

contentPane.add(btnexit);

txtpassword = **new** JPasswordField();

txtpassword.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

txtpassword.setBounds(394, 115, 126, 25);

contentPane.add(txtpassword);

btnlogin = **new** JButton("Σύνδεση");

btnlogin.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

Connection conn = **null**;

ResultSet rs = **null**;

Statement stmt = **null**;

**try**

{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT username, password FROM login");

Boolean flag = **false**;

**while**(rs.next()){

String username = rs.getString("username");

String password = rs.getString("password");

String pass = "";

**for**(**int** i=0; i < (txtpassword.getPassword()).length; i++){

pass = pass + (txtpassword.getPassword())[i];

}

**if** ((username.equals(txtusername.getText()))&&(password.equals(pass))){

flag = **true**;

**break**;

}

}

**if**(flag){

MainFrame next = **new** MainFrame(txtusername.getText());

next.setVisible(**true**);

dispose();

}

**else**{

txtusername.setText("");

txtpassword.setText("");

JOptionPane.*showMessageDialog*(**null**, "Invalid username or password!", "Login Error",JOptionPane.*ERROR\_MESSAGE*);

}

conn.close();

stmt.close();

}

**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Invalid username or password!", "Login Error",JOptionPane.*ERROR\_MESSAGE*);

}

}

});

btnlogin.setBackground(**new** Color(153, 204, 255));

btnlogin.setForeground(Color.*BLACK*);

btnlogin.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

btnlogin.setBounds(267, 163, 253, 40);

contentPane.add(btnlogin);

}

}

## Κλάση MainFrame.class

**package** omada15.softeng.teipir;

**import** javax.swing.JFrame;

**public** **class** MainFrame **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField txtsearch;

**private** JList list;

**private** JLabel lblname;

**private** JLabel lblnewprofile;

**private** JLabel lblconvertsection;

**private** JButton btnkatastatiko;

**private** JButton btnincomeoutcome;

**private** JButton btnexit;

**private** JButton btnnewprofile;

**private** JButton btntmimata;

**private** JButton btnsearch;

**private** JButton btncreatesection;

**private** JButton btndeletesection;

**private** JScrollPane scrollPane;

**private** ArrayList<String> afms;

**private** ArrayList<String> sections;

**public** MainFrame(**final** String username) {

setTitle("Company Management");

setIconImage(Toolkit.*getDefaultToolkit*().getImage(MainFrame.**class**.getResource("/com/sun/java/swing/plaf/windows/icons/Computer.gif")));

setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

setBounds(100, 100, 684, 414);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

lblname = **new** JLabel("Καλώς Ορίσατε " + getname(username) + " ");

lblname.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

Integer afm=0;

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT afm FROM login WHERE login.username = '" + username + "'");

**while**(rs.next()){

afm=Integer.*parseInt*(rs.getString("afm"));

}

stmt.close();

conn.close();

Profile profil = **new** Profile(username);

profil.setVisible(**true**);

profil.sundesi(afm);

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

}

@Override

**public** **void** mouseEntered(MouseEvent arg0) {

lblname.setForeground(**new** Color(0, 102, 255));

}

@Override

**public** **void** mouseExited(MouseEvent e) {

lblname.setForeground(**new** Color(0, 0, 0));

}

});

lblname.setHorizontalAlignment(SwingConstants.*RIGHT*);

lblname.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblname.setBounds(257, 11, 401, 30);

contentPane.add(lblname);

btnkatastatiko = **new** JButton("Καταστατικό");

btnkatastatiko.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try** {

Process p=Runtime.*getRuntime*().exec("notepad omada15/softeng/teipir/katastatiko.txt" );

} **catch** (Exception e) {

JOptionPane.*showMessageDialog*(**null**, "Cannot find file!", "File Error",JOptionPane.*ERROR\_MESSAGE*);

}

}

});

btnkatastatiko.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

btnkatastatiko.setBounds(10, 335, 135, 30);

contentPane.add(btnkatastatiko);

btnincomeoutcome = **new** JButton("Έσοδα - Έξοδα");

btnincomeoutcome.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

Income\_Outcome esodae3oda = **new** Income\_Outcome();

esodae3oda.setVisible(**true**);

}

});

btnincomeoutcome.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

btnincomeoutcome.setBounds(155, 335, 155, 29);

contentPane.add(btnincomeoutcome);

btnexit = **new** JButton("Έξοδος");

btnexit.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

System.*exit*(0);

}

});

btnexit.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

btnexit.setBounds(569, 335, 89, 29);

contentPane.add(btnexit);

txtsearch = **new** JTextField();

txtsearch.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

txtsearch.setText("");

txtsearch.setForeground(**new** Color(0, 0, 0));

txtsearch.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

}

});

txtsearch.setText("Αναζήτηση Εργαζομένου...");

txtsearch.setForeground(**new** Color(169, 169, 169));

txtsearch.setFont(**new** Font("Tahoma", Font.*ITALIC*, 13));

txtsearch.setBounds(416, 47, 203, 30);

txtsearch.setColumns(20);

contentPane.add(txtsearch);

btnnewprofile = **new** JButton("");

btnnewprofile.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

Profile profil = **new** Profile(username);

profil.setVisible(**true**);

profil.createconvert();

}

});

btnnewprofile.setIcon(**new** ImageIcon(MainFrame.**class**.getResource("/omada15/softeng/teipir/newprofile.jpg")));

btnnewprofile.setBounds(10, 79, 162, 167);

contentPane.add(btnnewprofile);

btntmimata = **new** JButton("");

btntmimata.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

btncreatesection.setVisible(**true**);

btndeletesection.setVisible(**true**);

fillsectionarea();

}

});

btntmimata.setIcon(**new** ImageIcon(MainFrame.**class**.getResource("/omada15/softeng/teipir/tmimata.jpg")));

btntmimata.setBounds(204, 79, 162, 167);

contentPane.add(btntmimata);

btnsearch = **new** JButton("");

btnsearch.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

btncreatesection.setVisible(**false**);

btndeletesection.setVisible(**false**);

searchemployee(txtsearch.getText());

txtsearch.setText("");

}

});

btnsearch.setIcon(**new** ImageIcon(MainFrame.**class**.getResource("/omada15/softeng/teipir/search.jpg")));

btnsearch.setBounds(620, 47, 37, 29);

contentPane.add(btnsearch);

lblnewprofile = **new** JLabel("Δημιουργία Προφίλ");

lblnewprofile.setHorizontalAlignment(SwingConstants.*CENTER*);

lblnewprofile.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblnewprofile.setBounds(10, 257, 162, 14);

contentPane.add(lblnewprofile);

lblconvertsection = **new** JLabel("Τμήματα Εταιρίας");

lblconvertsection.setHorizontalAlignment(SwingConstants.*CENTER*);

lblconvertsection.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblconvertsection.setBounds(204, 257, 172, 14);

contentPane.add(lblconvertsection);

btncreatesection = **new** JButton("Δημιουργία Τμήματος");

btncreatesection.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

Integer thesi = 0;

Section neotmima = **new** Section();

neotmima.setVisible(**true**);

neotmima.sundesi(thesi);

}

});

btncreatesection.setFont(**new** Font("Tahoma", Font.*BOLD*, 11));

btncreatesection.setBounds(416, 257, 242, 23);

btncreatesection.setVisible(**false**);

contentPane.add(btncreatesection);

btndeletesection = **new** JButton("Διαγραφή Τμήματος");

btndeletesection.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(list.getSelectedIndex() == -1){

JOptionPane.*showMessageDialog*(**null**, "Επιλέξτε πρώτα ένα τμήμα", "Delete Section Error",JOptionPane.*ERROR\_MESSAGE*);

}

**else**{

Integer selected = list.getSelectedIndex() + 2;

Connection conn = **null**;

Statement stmt = **null**;

**if**(selected != 1){

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

stmt.executeUpdate("DELETE FROM sections WHERE section\_id = " + selected);

stmt.executeUpdate("UPDATE company SET section\_id = 1 WHERE section\_id = " + selected);

conn.close();

stmt.close();

fillsectionarea();

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

}

**else**{

JOptionPane.*showMessageDialog*(**null**, "Επιλέξτε Ένα Τμήμα!", "General Error",JOptionPane.*ERROR\_MESSAGE*);

}

}

}

});

btndeletesection.setFont(**new** Font("Tahoma", Font.*BOLD*, 11));

btndeletesection.setBounds(416, 291, 242, 23);

btndeletesection.setVisible(**false**);

contentPane.add(btndeletesection);

list = **new** JList();

list.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if** (e.getClickCount() == 2 && !e.isConsumed()){

e.consume();

list.setSelectionForeground(**new** Color(0, 150, 0));

**if**((btndeletesection.isVisible() == **false**) && (btncreatesection.isVisible() == **false**)){

Integer thesi = list.getSelectedIndex();

String afm = (String) (afms.toArray())[thesi];

Profile profil = **new** Profile(username);

profil.setVisible(**true**);

profil.sundesi(Integer.*parseInt*(afm));

}

**else**{

Integer thesi = list.getSelectedIndex() + 2;

Section tmima = **new** Section();

tmima.setVisible(**true**);

tmima.sundesi(thesi);

}

}

**else** {

list.setSelectionForeground(**new** Color(255, 0, 0));

}

}

});

list.setSelectionMode(ListSelectionModel.*SINGLE\_SELECTION* );

list.setSelectedIndex(-1);

scrollPane = **new** JScrollPane(list);

scrollPane.setVerticalScrollBarPolicy(ScrollPaneConstants.*VERTICAL\_SCROLLBAR\_ALWAYS*);

scrollPane.setHorizontalScrollBarPolicy(ScrollPaneConstants.*HORIZONTAL\_SCROLLBAR\_AS\_NEEDED*);

scrollPane.setBounds(415, 88, 243, 167);

contentPane.add(scrollPane);

}

**public** **void** searchemployee(String search) {

**if** ((txtsearch.getText().equals("")) || (txtsearch.getText().equals("Αναζήτηση Εργαζομένου..."))){

JOptionPane.*showMessageDialog*(**null**, "Το στοιχείο αναζήτησης είναι κενό!", "Search Error",JOptionPane.*ERROR\_MESSAGE*);

txtsearch.setText("");

}

**else**{

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

**final** ArrayList<String> employees = **new** ArrayList<String>();

afms = **new** ArrayList<String>();

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT general.name, surname, general.afm, sections.name FROM general, company, sections WHERE general.afm = company.afm AND company.section\_id = sections.section\_id");

**while**(rs.next()){

**if**((rs.getString("general.name").equals(search)) || ((rs.getString("general.name") + " " + rs.getString("surname")).equals(search)) || ((rs.getString("surname") + " " + rs.getString("general.name")).equals(search)) || (rs.getString("general.afm").equals(search)) || (rs.getString("surname").equals(search))){

String employee = rs.getString("general.name") + " " + rs.getString("surname") + " - " + rs.getString("sections.name");

String afm = rs.getString("general.afm");

employees.add(employee);

afms.add(afm);

}

}

conn.close();

stmt.close();

list.setModel(**new** AbstractListModel() {

**public** **int** getSize() {

**return** (employees.toArray()).length;

}

**public** Object getElementAt(**int** index) {

**return** (employees.toArray())[index];

}

});

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "No Employee Found!", "General Error",JOptionPane.*ERROR\_MESSAGE*);

}

}

}

**public** **void** fillsectionarea(){

sections = **new** ArrayList<String>();

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT name FROM sections");

rs.next();

**while**(rs.next()){

String section = rs.getString("name");

sections.add(section);

}

list.setModel(**new** AbstractListModel() {

**public** **int** getSize() {

**return** (sections.toArray()).length;

}

**public** Object getElementAt(**int** index) {

**return** (sections.toArray())[index];

}

});

conn.close();

stmt.close();

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Δεν υπάρχουν τμήματα στην εταιρεία!", "General Error",JOptionPane.*ERROR\_MESSAGE*);

}

}

**public** String getname(String username) {

String onomeponum = "";

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT name, surname FROM general, login WHERE login.username = '" + username + "' AND login.afm = general.afm");

**while**(rs.next()){

onomeponum = rs.getString("name") + " " + rs.getString("surname");

}

conn.close();

stmt.close();

}**catch**(Exception e1){

}

**return** onomeponum;

}

}

## Κλάση Profile.class

**package** omada15.softeng.teipir;

**import** java.awt.BorderLayout;

**public** **class** Profile **extends** JFrame {

**private** JTabbedPane tabbedPane;

**private** JPanel panel1;

**private** JPanel panel2;

**private** JPanel panel3;

**private** JPanel panel4;

**private** JTextField txtbirthday;

**private** JTextField txtsurname;

**private** JTextField txtname;

**private** JPanel contentPane;

**private** JRadioButton rdbtnfemale;

**private** JRadioButton rdbtnmale;

**private** JLabel lblreview;

**private** JLabel lblafm;

**private** JTextField txtafm;

**private** JTextField txtamka;

**private** JLabel lblamka;

**private** JLabel lblmrtlstatus;

**private** JComboBox cmbBoxmrtlstatus;

**private** JLabel lblinformation;

**private** JLabel lbladdress;

**private** JTextField txtaddress;

**private** JLabel lblarea;

**private** JTextField txtarea;

**private** JLabel lbltk;

**private** JTextField txttk;

**private** JLabel lblphone;

**private** JTextField txtphone;

**private** JLabel lblmobile;

**private** JTextField txtmobile;

**private** JLabel lblEmail;

**private** JTextField txtemail;

**private** JLabel lblFax;

**private** JTextField txtfax;

**private** JLabel lblhireday;

**private** JLabel lblmeikta;

**private** JLabel lblasfaleia;

**private** JLabel lblgamos;

**private** JLabel lbltekna;

**private** JLabel lblkathara;

**private** JTextField txthireday;

**private** JLabel lblsection;

**private** JLabel lbltypeofwork;

**private** JComboBox cmbboxtypeofwork;

**private** JComboBox comboboxsection;

**private** JLabel lblmoney;

**private** JLabel lblall;

**private** JLabel lblminus;

**private** JLabel lblsum;

**private** JLabel lbl1tekno;

**private** JTextField txttekna;

**private** TextArea txtareainformation;

**private** TextArea txtareareviews;

**private** JCheckBox checkBoxmarriage;

**private** JLabel lblimage;

**private** JButton btndelete;

**private** JButton btnconvert;

**private** JButton btnsave;

**private** JButton btncancel;

**private** JTextField txtmeikta;

**private** JTextField txtasfaleia;

**private** JTextField txtepgamou;

**private** JTextField txteptekna;

**private** JLabel lblYyyymmdd;

**private** JLabel lblyyyymmdd2;

**private** JLabel lblchange;

**private** JLabel lblnumberofchild;

**private** JLabel lblname;

**private** JLabel lblsurname;

**private** JLabel lblbirthday;

**private** ButtonGroup group;

**private** String oldafm = "";

**public** String uname;

**public** Profile(String un) {

uname = un;

setIconImage(Toolkit.*getDefaultToolkit*().getImage(Profile.**class**.getResource("/com/sun/java/swing/plaf/windows/icons/Computer.gif")));

setTitle("Company Management");

setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);

setBounds(100, 100, 595, 498);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

contentPane.setLayout(**new** BorderLayout(0, 0));

setContentPane(contentPane);

createPage1();

createPage2();

createPage3();

createPage4();

tabbedPane = **new** JTabbedPane();

contentPane.add( tabbedPane, BorderLayout.*CENTER* );

tabbedPane.addTab("Γενικά", panel1);

tabbedPane.addTab("Επικοινωνία", panel2);

tabbedPane.addTab("Εταιρεία", panel3);

tabbedPane.addTab("Αξιολόγηση", panel4);

}

**private** **void** createPage1()

{

panel1 = **new** JPanel();

panel1.setLayout( **null** );

lblimage = **new** JLabel();

lblimage.setBounds(10, 11, 87, 132);

panel1.add(lblimage);

lblYyyymmdd = **new** JLabel("YYYY-MM-DD"); //Label που δείχνει στο χρήστη πως διατυπώνεται η ημερομηνια.

lblYyyymmdd.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

lblYyyymmdd.setBounds(429, 89, 97, 28);

panel1.add(lblYyyymmdd);

lblchange = **new** JLabel("Αλλαγή username και password");

lblchange.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) { //Δημιουργία νέου παραθύρου Αλλαγής username & password

ChangeUP allagi = **new** ChangeUP(uname);

allagi.setVisible(**true**);

}

@Override

**public** **void** mouseEntered(MouseEvent e) { //Αλλαγή χρώματος στην είσοδο του κέρσορα.

lblchange.setForeground(**new** Color(0, 102, 255));

}

@Override

**public** **void** mouseExited(MouseEvent e) { //Αλλαγή χρώματος στην έξοδο του κέρσορα.

lblchange.setForeground(**new** Color(0, 0, 0));

}

});

lblchange.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblchange.setBounds(10, 397, 207, 14);

panel1.add(lblchange);

lblbirthday = **new** JLabel("Ημ/νία γέννησης");

lblbirthday.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblbirthday.setBounds(142, 89, 122, 28);

panel1.add(lblbirthday);

lblsurname = **new** JLabel("Επώνυμο");

lblsurname.setHorizontalAlignment(SwingConstants.*LEFT*);

lblsurname.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblsurname.setBounds(142, 50, 75, 28);

panel1.add(lblsurname);

lblname = **new** JLabel("Όνομα");

lblname.setHorizontalAlignment(SwingConstants.*LEFT*);

lblname.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblname.setBounds(142, 11, 70, 28);

panel1.add(lblname);

txtname = **new** JTextField();

txtname.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtname.setBounds(227, 11, 192, 28);

txtname.setColumns(20);

panel1.add(txtname);

txtsurname = **new** JTextField();

txtsurname.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtsurname.setBounds(227, 50, 192, 28);

txtsurname.setColumns(20);

panel1.add(txtsurname);

txtbirthday = **new** JTextField();

txtbirthday.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtbirthday.setBounds(297, 89, 122, 28);

txtbirthday.setColumns(10);

panel1.add(txtbirthday);

rdbtnfemale = **new** JRadioButton("Γυναίκα");

rdbtnfemale.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

rdbtnfemale.setBounds(227, 134, 106, 23);

panel1.add(rdbtnfemale);

rdbtnmale = **new** JRadioButton("Άνδρας");

rdbtnmale.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

rdbtnmale.setBounds(142, 134, 87, 23);

panel1.add(rdbtnmale);

group = **new** ButtonGroup();

group.add(rdbtnmale);

group.add(rdbtnfemale);

lblafm = **new** JLabel("Α.Φ.Μ.");

lblafm.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblafm.setBounds(25, 248, 70, 28);

panel1.add(lblafm);

txtafm = **new** JTextField();

txtafm.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtafm.setBounds(105, 248, 112, 28);

txtafm.setColumns(9);

panel1.add(txtafm);

txtamka = **new** JTextField();

txtamka.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtamka.setBounds(105, 287, 112, 28);

txtamka.setColumns(10);

panel1.add(txtamka);

lblamka = **new** JLabel("A.M.K.A.");

lblamka.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblamka.setBounds(25, 287, 70, 28);

panel1.add(lblamka);

lblmrtlstatus = **new** JLabel("Οικογενειακή Κατάσταση");

lblmrtlstatus.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblmrtlstatus.setBounds(25, 177, 176, 28);

panel1.add(lblmrtlstatus);

cmbBoxmrtlstatus = **new** JComboBox();

cmbBoxmrtlstatus.setModel(**new** DefaultComboBoxModel(**new** String[] {"Επιλέξτε...", "Έγγαμος/-η", "Άγαμος/-η", "Διαζευγμένος/-η", "Χήρος/-α"}));

cmbBoxmrtlstatus.setSelectedIndex(0);

cmbBoxmrtlstatus.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

cmbBoxmrtlstatus.setBounds(211, 177, 122, 28);

panel1.add(cmbBoxmrtlstatus);

lblinformation = **new** JLabel("Λοιπές Πληροφορίες ");

lblinformation.setHorizontalAlignment(SwingConstants.*RIGHT*);

lblinformation.setFont(**new** Font("Tahoma", Font.*ITALIC*, 13));

lblinformation.setBounds(350, 214, 192, 28);

panel1.add(lblinformation);

txtareainformation = **new** TextArea();

txtareainformation.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtareainformation.setBounds(307, 248, 235, 67);

panel1.add(txtareainformation);

//BUTTONS

btndelete = **new** JButton("Διαγραφή Προφίλ");

btndelete.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

deleteprofile();

}

});

btndelete.setForeground(**new** Color(204, 0, 0));

btndelete.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

btndelete.setBounds(307, 339, 235, 35);

panel1.add(btndelete);

btnconvert = **new** JButton("Επεξεργασία Προφίλ");

btnconvert.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

createconvert();

}

});

btnconvert.setForeground(**new** Color(0, 102, 255));

btnconvert.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

btnconvert.setBounds(25, 339, 192, 35);

panel1.add(btnconvert);

btnsave = **new** JButton("Αποθήκευση Προφίλ");

btnsave.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

saveprofile();

}

});

btnsave.setForeground(**new** Color(0, 100, 0));

btnsave.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

btnsave.setBounds(25, 339, 192, 35);

panel1.add(btnsave);

btncancel = **new** JButton("Ακύρωση");

btncancel.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

lblreview.setVisible(**true**);

lblmeikta.setVisible(**true**);

lblasfaleia.setVisible(**true**);

lblgamos.setVisible(**true**);

lbltekna.setVisible(**true**);

lblkathara.setVisible(**true**);

lblchange.setVisible(**true**);

lbl1tekno.setVisible(**false**);

sundesi(Integer.*parseInt*(oldafm));

}

});

btncancel.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

btncancel.setBounds(307, 339, 235, 35);

panel1.add(btncancel);

}

**private** **void** createPage2()

{

panel2 = **new** JPanel();

panel2.setLayout(**null**);

lbladdress = **new** JLabel("Διεύθυνση");

lbladdress.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lbladdress.setBounds(10, 40, 75, 28);

panel2.add(lbladdress);

txtaddress = **new** JTextField();

txtaddress.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtaddress.setBounds(95, 40, 259, 28);

panel2.add(txtaddress);

txtaddress.setColumns(10);

lblarea = **new** JLabel("Περιοχή");

lblarea.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblarea.setBounds(10, 79, 75, 28);

panel2.add(lblarea);

txtarea = **new** JTextField();

txtarea.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtarea.setBounds(95, 79, 189, 28);

panel2.add(txtarea);

txtarea.setColumns(10);

lbltk = **new** JLabel("Τ.Κ.");

lbltk.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lbltk.setBounds(10, 118, 75, 28);

panel2.add(lbltk);

txttk = **new** JTextField();

txttk.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txttk.setBounds(95, 118, 112, 28);

panel2.add(txttk);

txttk.setColumns(10);

lblphone = **new** JLabel("Τηλέφωνο");

lblphone.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblphone.setBounds(10, 157, 75, 28);

panel2.add(lblphone);

txtphone = **new** JTextField();

txtphone.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtphone.setBounds(95, 157, 189, 28);

panel2.add(txtphone);

txtphone.setColumns(10);

lblmobile = **new** JLabel("Κινητό");

lblmobile.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblmobile.setBounds(10, 196, 75, 28);

panel2.add(lblmobile);

txtmobile = **new** JTextField();

txtmobile.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtmobile.setBounds(95, 196, 189, 28);

panel2.add(txtmobile);

txtmobile.setColumns(10);

lblEmail = **new** JLabel("E-mail");

lblEmail.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblEmail.setBounds(10, 235, 75, 28);

panel2.add(lblEmail);

txtemail = **new** JTextField();

txtemail.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtemail.setBounds(95, 235, 259, 28);

panel2.add(txtemail);

txtemail.setColumns(10);

lblFax = **new** JLabel("FAX");

lblFax.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblFax.setBounds(10, 274, 75, 28);

panel2.add(lblFax);

txtfax = **new** JTextField();

txtfax.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtfax.setBounds(95, 274, 189, 28);

panel2.add(txtfax);

txtfax.setColumns(10);

}

**private** **void** createPage3(){

panel3 = **new** JPanel();

panel3.setLayout(**null**);

lblhireday = **new** JLabel("Ημ/νία Πρόσληψης");

lblhireday.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblhireday.setBounds(10, 11, 134, 25);

panel3.add(lblhireday);

lblyyyymmdd2 = **new** JLabel("YYYY-MM-DD");

lblyyyymmdd2.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

lblyyyymmdd2.setBounds(339, 11, 97, 28);

panel3.add(lblyyyymmdd2);

txthireday = **new** JTextField();

txthireday.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txthireday.setBounds(154, 11, 175, 25);

panel3.add(txthireday);

txthireday.setColumns(10);

lblsection = **new** JLabel("Τμήμα");

lblsection.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblsection.setBounds(10, 47, 134, 25);

panel3.add(lblsection);

comboboxsection = **new** JComboBox();

comboboxsection.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

comboboxsection.setBounds(154, 47, 175, 25);

panel3.add(comboboxsection);

lbltypeofwork = **new** JLabel("Απασχόληση");

lbltypeofwork.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lbltypeofwork.setBounds(10, 83, 134, 25);

panel3.add(lbltypeofwork);

cmbboxtypeofwork = **new** JComboBox();

cmbboxtypeofwork.setModel(**new** DefaultComboBoxModel(**new** String[] {"Επιλέξτε...", "Μερική", "Πλήρη", "Εποχιακή"}));

cmbboxtypeofwork.setSelectedIndex(0);

cmbboxtypeofwork.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

cmbboxtypeofwork.setBounds(154, 83, 175, 25);

panel3.add(cmbboxtypeofwork);

lblmoney = **new** JLabel("Μισθοδοσία");

lblmoney.setFont(**new** Font("Tahoma", Font.*BOLD*, 13));

lblmoney.setBounds(10, 161, 134, 25);

panel3.add(lblmoney);

lbl1tekno = **new** JLabel("/ \u03C4\u03AD\u03BA\u03BD\u03BF");

lbl1tekno.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lbl1tekno.setBounds(301, 275, 66, 19);

panel3.add(lbl1tekno);

lblall = **new** JLabel("Μεικτά");

lblall.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblall.setBounds(10, 197, 46, 25);

panel3.add(lblall);

lblminus = **new** JLabel("Ασφάλεια");

lblminus.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblminus.setBounds(10, 222, 72, 25);

panel3.add(lblminus);

lblsum = **new** JLabel("Καθαρά");

lblsum.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblsum.setBounds(10, 293, 59, 20);

panel3.add(lblsum);

checkBoxmarriage = **new** JCheckBox("Επίδομα Γάμου");

checkBoxmarriage.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

checkBoxmarriage.setBounds(10, 246, 134, 25);

panel3.add(checkBoxmarriage);

lblnumberofchild = **new** JLabel("Αριθμός τέκνων");

lblnumberofchild.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblnumberofchild.setBounds(10, 269, 93, 25);

panel3.add(lblnumberofchild);

txttekna = **new** JTextField();

txttekna.setFont(**new** Font("Tahoma", Font.*PLAIN*, 12));

txttekna.setBounds(114, 271, 18, 20);

panel3.add(txttekna);

txttekna.setColumns(10);

lblmeikta = **new** JLabel();

lblmeikta.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblmeikta.setBounds(175, 197, 93, 25);

panel3.add(lblmeikta);

lblasfaleia = **new** JLabel();

lblasfaleia.setForeground(**new** Color(255, 0, 0));

lblasfaleia.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblasfaleia.setBounds(175, 222, 93, 25);

panel3.add(lblasfaleia);

lblgamos = **new** JLabel();

lblgamos.setForeground(**new** Color(0, 100, 0));

lblgamos.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lblgamos.setBounds(175, 246, 93, 25);

panel3.add(lblgamos);

lbltekna = **new** JLabel();

lbltekna.setForeground(**new** Color(0, 100, 0));

lbltekna.setFont(**new** Font("Tahoma", Font.*ITALIC*, 12));

lbltekna.setBounds(175, 269, 93, 25);

panel3.add(lbltekna);

lblkathara = **new** JLabel();

lblkathara.setFont(**new** Font("Tahoma", Font.*BOLD*, 12));

lblkathara.setBounds(175, 293, 93, 20);

panel3.add(lblkathara);

txtmeikta = **new** JTextField();

txtmeikta.setFont(**new** Font("Tahoma", Font.*PLAIN*, 12));

txtmeikta.setBounds(185, 198, 106, 25);

txtmeikta.setColumns(10);

panel3.add(txtmeikta);

txtasfaleia = **new** JTextField();

txtasfaleia.setFont(**new** Font("Tahoma", Font.*PLAIN*, 12));

txtasfaleia.setColumns(10);

txtasfaleia.setBounds(185, 223, 106, 25);

panel3.add(txtasfaleia);

txtepgamou = **new** JTextField();

txtepgamou.setFont(**new** Font("Tahoma", Font.*PLAIN*, 12));

txtepgamou.setColumns(10);

txtepgamou.setBounds(185, 247, 106, 25);

panel3.add(txtepgamou);

txteptekna = **new** JTextField();

txteptekna.setFont(**new** Font("Tahoma", Font.*PLAIN*, 12));

txteptekna.setColumns(10);

txteptekna.setBounds(185, 272, 106, 25);

panel3.add(txteptekna);

}

**private** **void** createPage4()

{

panel4 = **new** JPanel();

panel4.setLayout(**null**);

lblreview = **new** JLabel("Αξιολόγηση Υπαλλήλου");

lblreview.setFont(**new** Font("Tahoma", Font.*ITALIC*, 13));

lblreview.setBounds(0, 44, 147, 38);

panel4.add(lblreview);

txtareareviews = **new** TextArea();

txtareareviews.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtareareviews.setBounds(10, 88, 532, 172);

panel4.add( txtareareviews );

}

**public** **void** sundesi(Integer afm){

lblchange.setVisible(**true**);

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT \* FROM general WHERE general.afm = " + afm);

**while**(rs.next()){

txtname.setText(rs.getString("name"));

txtsurname.setText(rs.getString("surname"));

txtbirthday.setText(rs.getString("birthday"));

txtafm.setText(rs.getString("afm"));

oldafm = txtafm.getText();

txtamka.setText(rs.getString("amka"));

**if** (rs.getString("sex").equals("Γυναίκα")){

rdbtnfemale.setSelected(**true**);

lblimage.setIcon(**new** ImageIcon(Profile.**class**.getResource("/omada15/softeng/teipir/female.jpg")));

}

**else** **if**(rs.getString("sex").equals("Άντρας")){

rdbtnmale.setSelected(**true**);

lblimage.setIcon(**new** ImageIcon(Profile.**class**.getResource("/omada15/softeng/teipir/male.jpg")));

}

**else**{

lblimage.setText("");

}

**if**(rs.getString("marital\_id").equals("1")){

cmbBoxmrtlstatus.setSelectedIndex(0);

}

**else**{

cmbBoxmrtlstatus.setSelectedIndex((Integer.*parseInt*(rs.getString("marital\_id"))) - 1);

**if**(cmbBoxmrtlstatus.getSelectedIndex() == 1){

checkBoxmarriage.setSelected(**true**);

}

**else**{

checkBoxmarriage.setSelected(**false**);

}

}

txtareainformation.setText(rs.getString("information"));

}

rs = stmt.executeQuery("SELECT \* FROM communication WHERE communication.afm = " + afm);

**while**(rs.next()){

txtaddress.setText(rs.getString("address"));

txtarea.setText(rs.getString("region"));

txttk.setText(rs.getString("tk"));

txtphone.setText(rs.getString("tel"));

txtmobile.setText(rs.getString("mobile"));

txtemail.setText(rs.getString("email"));

txtfax.setText(rs.getString("fax"));

}

rs = stmt.executeQuery("SELECT name FROM sections");

fillsectioncombobox(rs);

rs = stmt.executeQuery("SELECT \* FROM company, sections WHERE company.section\_id = sections.section\_id AND company.afm = " + afm);

**while**(rs.next()){

txthireday.setText(rs.getString("hire\_day"));

**if**(rs.getString("working\_id").equals("1")){

cmbboxtypeofwork.setSelectedIndex(0);

}

**else** {

cmbboxtypeofwork.setSelectedIndex(Integer.*parseInt*(rs.getString("working\_id"))-1);

}

txttekna.setText(rs.getString("children"));

lblmeikta.setText(rs.getString("meikta"));

lblasfaleia.setText(rs.getString("asfaleia"));

**if**(rs.getString("section\_id").equals(**null**)){

comboboxsection.setSelectedIndex(-1);

}

**else**{

comboboxsection.setSelectedIndex((Integer.*parseInt*(rs.getString("section\_id"))) - 1);

}

**if**((comboboxsection.getSelectedIndex() != 4) && (comboboxsection.getSelectedIndex() != 1)){

lblchange.setVisible(**false**);

}

}

rs = stmt.executeQuery("SELECT review FROM reviews WHERE reviews.afm = " + afm);

**while**(rs.next()){

txtareareviews.setText(rs.getString("review"));

}

rs = stmt.executeQuery("SELECT \* FROM statheres");

**while**(rs.next()){

txtepgamou.setText(rs.getString("epidoma\_gamou"));

txteptekna.setText(rs.getString("epidoma\_teknou"));

}

Integer temp;

temp= Integer.*parseInt*(txttekna.getText()) \* Integer.*parseInt*(txteptekna.getText());

lbltekna.setText(temp.toString());

**if**(checkBoxmarriage.isSelected()){

lblgamos.setText(txtepgamou.getText());

}

**else**{

lblgamos.setText("0");

}

temp = Integer.*parseInt*(lblmeikta.getText()) - Integer.*parseInt*(lblasfaleia.getText()) + Integer.*parseInt*(lblgamos.getText()) + Integer.*parseInt*(lbltekna.getText());

lblkathara.setText(temp.toString());

stmt.close();

conn.close();

editable(**false**);

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

btnconvert.setVisible(**true**);

btncancel.setVisible(**false**);

btnsave.setVisible(**false**);

btndelete.setVisible(**true**);

}

**public** **void** deleteprofile(){

**if**(!(txtafm.getText().equals(""))){

**try**{

Integer afm = Integer.*parseInt*(txtafm.getText());

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

Connection conn = **null**;

Statement stmt = **null**;

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

stmt.executeUpdate("DELETE FROM general WHERE afm = " + afm);

stmt.executeUpdate("DELETE FROM communication WHERE afm = " + afm);

stmt.executeUpdate("DELETE FROM company WHERE afm = " + afm);

stmt.executeUpdate("DELETE FROM reviews WHERE afm = " + afm);

stmt.executeUpdate("DELETE FROM login WHERE afm = " + afm);

stmt.close();

conn.close();

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

oldafm="";

dispose();

}

**else**{

JOptionPane.*showMessageDialog*(**null**, "Η Συμπλήρωση του ΑΦΜ είναι υποχρεωτική!", "Invalid Input",JOptionPane.*ERROR\_MESSAGE*);

}

}

**public** **void** createconvert(){

editable(**true**);

lblimage.setText("");

lblmeikta.setVisible(**false**);

lblasfaleia.setVisible(**false**);

lblgamos.setVisible(**false**);

lbltekna.setVisible(**false**);

lblkathara.setVisible(**false**);

lbl1tekno.setVisible(**true**);

lblchange.setVisible(**false**);

txtmeikta.setText(lblmeikta.getText());

txtasfaleia.setText(lblasfaleia.getText());

cmbBoxmrtlstatus.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

**if**(cmbBoxmrtlstatus.getSelectedIndex() == 1){

checkBoxmarriage.setSelected(**true**);

}

**else**{

checkBoxmarriage.setSelected(**false**);

}

}

});

Integer current\_section = comboboxsection.getSelectedIndex();

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT name FROM sections");

fillsectioncombobox(rs);

comboboxsection.setSelectedIndex(current\_section);

rs = stmt.executeQuery("SELECT \* FROM statheres");

**while**(rs.next()){

txtepgamou.setText(rs.getString("epidoma\_gamou"));

txteptekna.setText(rs.getString("epidoma\_teknou"));

}

rs = stmt.executeQuery("SELECT username FROM login WHERE username = '" + uname + "'");

Boolean f = **true**;

**while**(rs.next()){

f = **false**;

**if**(!(rs.getString("username").equals("admin"))){

txtareareviews.setEditable(**false**);

}

}

**if**(f){

txtareareviews.setEditable(**false**);

}

stmt.close();

conn.close();

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

btnconvert.setVisible(**false**);

btncancel.setVisible(**true**);

btnsave.setVisible(**true**);

btndelete.setVisible(**false**);

}

**public** **void** saveprofile(){

lblreview.setVisible(**true**);

String gender = "";

**if** (rdbtnmale.isSelected()){

gender = "Άντρας";

lblimage.setIcon(**new** ImageIcon(Profile.**class**.getResource("/omada15/softeng/teipir/male.jpg")));

}

**else**{

gender = "Γυναίκα";

lblimage.setIcon(**new** ImageIcon(Profile.**class**.getResource("/omada15/softeng/teipir/female.jpg")));

}

**if**(txttekna.getText().equals("")){

txttekna.setText("0");

}

**if**((comboboxsection.getSelectedIndex() != 4) && (comboboxsection.getSelectedIndex() != 1)){

lblchange.setVisible(**false**);

}

**else**{

lblchange.setVisible(**true**);

}

**if**(txtafm.getText().equals("")){

JOptionPane.*showMessageDialog*(**null**, "Η Συμπλήρωση του ΑΦΜ είναι υποχρεωτική!", "Invalid Input",JOptionPane.*ERROR\_MESSAGE*);

}

**else**{

**try**

{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

Connection conn = **null**;

Statement stmt = **null**;

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

**if**(oldafm.equals("")){

stmt.executeUpdate("INSERT INTO general values('" + txtname.getText() + "', '" + txtsurname.getText() + "', '" + txtbirthday.getText() + "', '" + gender + "', " + (cmbBoxmrtlstatus.getSelectedIndex() + 1) + ", '" + txtamka.getText() + "', '" + txtareainformation.getText() + "', " + Integer.*parseInt*(txtafm.getText()) + ")");

stmt.executeUpdate("INSERT INTO communication values('" + txtaddress.getText() + "', '" + txtarea.getText() + "', '" + txttk.getText() + "', '" + txtphone.getText() + "', '" + txtmobile.getText() + "', '" + txtemail.getText() + "', '" + txtfax.getText() + "', " + Integer.*parseInt*(txtafm.getText()) + ")");

stmt.executeUpdate("INSERT INTO reviews values(" + Integer.*parseInt*(txtafm.getText()) + ", '" + txtareareviews.getText() + "')");

stmt.executeUpdate("INSERT INTO company values('" + txthireday.getText() + "', " + (cmbboxtypeofwork.getSelectedIndex() + 1) + ", " + Integer.*parseInt*(txtmeikta.getText()) + ", " + Integer.*parseInt*(txtasfaleia.getText()) + ", " + Integer.*parseInt*(txttekna.getText()) + ", " + Integer.*parseInt*(txtafm.getText()) + ", " + (comboboxsection.getSelectedIndex() + 1) + ")");

**if**(comboboxsection.getSelectedIndex()==4){

stmt.executeUpdate("INSERT INTO login values('user', 'user', " + Integer.*parseInt*(txtafm.getText()) + ")");

}

}

**else**{

stmt.executeUpdate("UPDATE general SET name = '" + txtname.getText() + "', surname = '" + txtsurname.getText() + "', birthday = '" + txtbirthday.getText() + "', sex = '" + gender + "', marital\_id = " + (cmbBoxmrtlstatus.getSelectedIndex() + 1) + ", amka = '" + txtamka.getText() + "', information = '" + txtareainformation.getText() + "', afm = " + Integer.*parseInt*(txtafm.getText()) + " WHERE afm = " + oldafm);

stmt.executeUpdate("UPDATE communication SET address = '" + txtaddress.getText() + "', region = '" + txtarea.getText() + "', tk = '" + txttk.getText() + "', tel = '" + txtphone.getText() + "', mobile = '" + txtmobile.getText() + "', email = '" + txtemail.getText() + "', fax = '" + txtfax.getText() + "', afm = " + Integer.*parseInt*(txtafm.getText()) + " WHERE afm = " + oldafm);

stmt.executeUpdate("UPDATE reviews SET afm = " + Integer.*parseInt*(txtafm.getText()) + ", review = '" + txtareareviews.getText() + "' WHERE afm = " + oldafm);

stmt.executeUpdate("UPDATE company SET hire\_day = '" + txthireday.getText() + "', working\_id = " + (cmbboxtypeofwork.getSelectedIndex() + 1) + ", children = " + Integer.*parseInt*(txttekna.getText()) + ", meikta = '" + txtmeikta.getText() + "', asfaleia = '" + txtasfaleia.getText() + "', section\_id = " + (comboboxsection.getSelectedIndex() + 1) + ", afm = " + Integer.*parseInt*(txtafm.getText()) + " WHERE afm = " + oldafm);

stmt.executeUpdate("UPDATE login SET afm = " + Integer.*parseInt*(txtafm.getText()) + " WHERE afm = " + oldafm);

}

stmt.executeUpdate("UPDATE statheres SET epidoma\_gamou = " + Integer.*parseInt*(txtepgamou.getText()) + ", epidoma\_teknou = " + Integer.*parseInt*(txteptekna.getText()));

stmt.close();

conn.close();

}

**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

lblmeikta.setText(txtmeikta.getText());

lblasfaleia.setText(txtasfaleia.getText());

**if**(checkBoxmarriage.isSelected()){

lblgamos.setText(txtepgamou.getText());

}

**else**{

lblgamos.setText("0");

}

lblmeikta.setVisible(**true**);

lblasfaleia.setVisible(**true**);

lblgamos.setVisible(**true**);

lbltekna.setVisible(**true**);

lblkathara.setVisible(**true**);

Integer temp;

temp = Integer.*parseInt*(txttekna.getText()) \* Integer.*parseInt*(txteptekna.getText());

lbltekna.setText(temp.toString());

temp = Integer.*parseInt*(lblmeikta.getText()) - Integer.*parseInt*(lblasfaleia.getText()) + Integer.*parseInt*(lblgamos.getText()) + Integer.*parseInt*(lbltekna.getText());

lblkathara.setText(temp.toString());

btnconvert.setVisible(**true**);

btndelete.setVisible(**true**);

btnsave.setVisible(**false**);

btncancel.setVisible(**false**);

editable(**false**);

}

}

**private** **void** fillsectioncombobox(ResultSet rs){

ArrayList<String> sections = **new** ArrayList<String>();

**try** {

**while**(rs.next()){

String section = rs.getString("name");

sections.add(section);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

comboboxsection.setModel(**new** DefaultComboBoxModel(sections.toArray()));

}

**private** **void** editable(Boolean flag){

txtname.setEditable(flag);

txtsurname.setEditable(flag);

txtbirthday.setEditable(flag);

rdbtnmale.setEnabled(flag);

rdbtnfemale.setEnabled(flag);

txtafm.setEditable(flag);

txtamka.setEditable(flag);

cmbBoxmrtlstatus.setEnabled(flag);

txtareainformation.setEditable(flag);

txtaddress.setEditable(flag);

txtarea.setEditable(flag);

txttk.setEditable(flag);

txtphone.setEditable(flag);

txtmobile.setEditable(flag);

txtemail.setEditable(flag);

txtfax.setEditable(flag);

txthireday.setEditable(flag);

cmbboxtypeofwork.setEnabled(flag);

checkBoxmarriage.setEnabled(flag);

txttekna.setEditable(flag);

txtareareviews.setEditable(flag);

comboboxsection.setEnabled(flag);

txtmeikta.setVisible(flag);

txtasfaleia.setVisible(flag);

txtepgamou.setVisible(flag);

txteptekna.setVisible(flag);

lbl1tekno.setVisible(flag);

}

}

## Κλάση Profile.class

**package** omada15.softeng.teipir;

**import** java.awt.Color;

**public** **class** ChangeUP **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField txtousername;

**private** JTextField txtnusername;

**private** JPasswordField passopassword;

**private** JPasswordField passnpassword;

**private** JLabel lblousername;

**private** JLabel lblnusername;

**private** JLabel lblopassword;

**private** JLabel lblnpassword;

**private** JButton btncancel;

**private** JButton btnconfirm;

**public** ChangeUP(**final** String user) {

setIconImage(Toolkit.*getDefaultToolkit*().getImage(ChangeUP.**class**.getResource("/com/sun/java/swing/plaf/windows/icons/Computer.gif")));

setTitle("Company Management");

setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);

setBounds(100, 100, 353, 379);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

lblousername = **new** JLabel("Παλιό Username");

lblousername.setForeground(**new** Color(204, 102, 102));

lblousername.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblousername.setBounds(10, 11, 139, 27);

contentPane.add(lblousername);

lblnusername = **new** JLabel("Νέο Username");

lblnusername.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblnusername.setBounds(10, 60, 139, 27);

lblnusername.setForeground(**new** Color(0, 102, 255));

contentPane.add(lblnusername);

lblopassword = **new** JLabel("Παλιό Password");

lblopassword.setForeground(**new** Color(204, 102, 102));

lblopassword.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblopassword.setBounds(10, 146, 139, 27);

contentPane.add(lblopassword);

lblnpassword = **new** JLabel("Νέο Password");

lblnpassword.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblnpassword.setBounds(10, 193, 139, 27);

lblnpassword.setForeground(**new** Color(0, 102, 255));

contentPane.add(lblnpassword);

//Text gia palio username

txtousername = **new** JTextField(user);

txtousername.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtousername.setBounds(159, 11, 148, 27);

contentPane.add(txtousername);

txtousername.setColumns(10);

//Text gia kainourgio username

txtnusername = **new** JTextField();

txtnusername.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtnusername.setBounds(159, 60, 148, 27);

contentPane.add(txtnusername);

txtnusername.setColumns(10);

//PasswordField gia palio password

passopassword = **new** JPasswordField();

passopassword.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

passopassword.setBounds(159, 146, 148, 25);

contentPane.add(passopassword);

//PasswordField gia neo password

passnpassword = **new** JPasswordField();

passnpassword.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

passnpassword.setBounds(159, 193, 148, 27);

contentPane.add(passnpassword);

**if**(user.equals("admin")){

txtousername.setText(user);

txtnusername.setText(user);

txtousername.setEditable(**false**);

txtnusername.setEditable(**false**);

}

btncancel = **new** JButton("Άκυρο");

btncancel.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

dispose();

}

});

btncancel.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

btncancel.setBounds(10, 293, 297, 35);

contentPane.add(btncancel);

btnconfirm = **new** JButton("Επιβεβαίωση");

btnconfirm.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

String oldpass = "";

String newpass = "";

**for**(**int** i=0; i < (passopassword.getPassword()).length; i++){

oldpass = oldpass + (passopassword.getPassword())[i];

}

**for**(**int** i=0; i < (passnpassword.getPassword()).length; i++){

newpass = newpass + (passnpassword.getPassword())[i];

}

**if**((txtousername.getText().equals("")) || (txtnusername.getText().equals("")) || (oldpass.equals("")) || (newpass.equals(""))){

JOptionPane.*showMessageDialog*(**null**, "Συμπληρώστε όλα τα πεδία!", "Confirmation Error",JOptionPane.*ERROR\_MESSAGE*);

}

**else**{

Connection conn = **null**;

ResultSet rs = **null**;

Statement stmt = **null**;

Statement stmt2 = **null**;

**try**

{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT username, password FROM login WHERE username = '" + user + "'");

stmt2 = conn.createStatement();

String username = **null**;

String password = **null**;

**while**(rs.next()){

username = rs.getString("username");

password = rs.getString("password");

}

**if**((txtousername.getText().equals(username)) && (oldpass.equals(password))){

stmt2.executeUpdate("UPDATE login SET username = '" + txtnusername.getText() + "', password = '" + newpass + "' WHERE username = '" + txtousername.getText() + "'");

dispose();

}

**else**{

JOptionPane.*showMessageDialog*(**null**, "Λάθος παλιό username ή password!", "Confirmation Error",JOptionPane.*ERROR\_MESSAGE*);

}

stmt.close();

stmt2.close();

conn.close();

}

**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

}

}

});

btnconfirm.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

btnconfirm.setBounds(10, 247, 297, 35);

contentPane.add(btnconfirm);

}

}

## Κλάση Income\_Outcome.class

**package** omada15.softeng.teipir;

**import** javax.swing.JFrame;

**public** **class** Income\_Outcome **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField txtyear;

**private** JTextField txtincome;

**private** JTextField txtoutcome;

**private** JComboBox cmbmonth;

**private** JLabel lblmonth;

**private** JLabel lblyear;

**private** JLabel lblincome;

**private** JLabel lbloutcome;

**private** JButton btnok;

**private** JButton btnshow;

**public** Income\_Outcome() {

setTitle("Έσοδα - Έξοδα");

setIconImage(Toolkit.*getDefaultToolkit*().getImage(Income\_Outcome.**class**.getResource("/com/sun/java/swing/plaf/windows/icons/Computer.gif")));

setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);

setBounds(100, 100, 273, 343);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

String[] combolist = {"Επιλέξτε...", "Ιανουάριος", "Φεβρουάριος", "Μάρτιος", "Απρίλιος", "Μάιος", "Ιούνιος",

"Ιούλιος", "Αύγουστος", "Σεπτέμβριος", "Οκτώβριος", "Νοέμβριος", "Δεκέμβριος"};

cmbmonth = **new** JComboBox(combolist);

cmbmonth.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

cmbmonth.setSelectedIndex(0);

cmbmonth.setBounds(10, 68, 137, 20);

contentPane.add(cmbmonth);

lblmonth = **new** JLabel("Μήνας");

lblmonth.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblmonth.setBounds(10, 37, 64, 20);

contentPane.add(lblmonth);

txtyear = **new** JTextField();

txtyear.setFont(**new** Font("Tahoma", Font.*PLAIN*, 13));

txtyear.setBounds(171, 68, 56, 20);

contentPane.add(txtyear);

txtyear.setColumns(10);

lblyear = **new** JLabel("Έτος");

lblyear.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblyear.setBounds(171, 37, 56, 20);

contentPane.add(lblyear);

lblincome = **new** JLabel("Έσοδα");

lblincome.setHorizontalAlignment(SwingConstants.*LEFT*);

lblincome.setForeground(**new** Color(0, 204, 51));

lblincome.setFont(**new** Font("Tahoma", Font.*BOLD*, 22));

lblincome.setBounds(10, 115, 99, 37);

contentPane.add(lblincome);

txtincome = **new** JTextField();

txtincome.setBounds(119, 128, 108, 20);

contentPane.add(txtincome);

txtincome.setColumns(10);

lbloutcome = **new** JLabel("Έξοδα");

lbloutcome.setForeground(**new** Color(255, 0, 0));

lbloutcome.setFont(**new** Font("Tahoma", Font.*BOLD*, 22));

lbloutcome.setBounds(10, 163, 99, 37);

contentPane.add(lbloutcome);

txtoutcome = **new** JTextField();

txtoutcome.setBounds(119, 176, 108, 20);

contentPane.add(txtoutcome);

txtoutcome.setColumns(10);

btnok = **new** JButton("OK");

btnok.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

btnok.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**try** {

File file = **new** File("omada15/softeng/teipir/esoda-eksoda.txt");

FileWriter fw = **new** FileWriter(file.getAbsoluteFile(), **true**);

BufferedWriter bw = **new** BufferedWriter(fw);

bw.write("================================\r\n");

bw.write(cmbmonth.getSelectedItem() + "\t" + txtyear.getText() + "\r\n");

bw.write("================================\r\n");

bw.write("'Εσοδα: " + txtincome.getText() + "\r\n" + "'Εξοδα: " + txtoutcome.getText() + "\r\n\r\n\r\n");

bw.close();

} **catch** (IOException e1) {

JOptionPane.*showMessageDialog*(**null**, "File not Found!", "General Error",JOptionPane.*ERROR\_MESSAGE*);

}

}

});

btnok.setBounds(10, 223, 217, 25);

contentPane.add(btnok);

btnshow = **new** JButton("Προβολή Εσόδων - Εξόδων");

btnshow.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**try** {

Process p=Runtime.*getRuntime*().exec("notepad omada15/softeng/teipir/esoda-eksoda.txt" );

} **catch** (Exception e1) {

JOptionPane.*showMessageDialog*(**null**, "File not Found!", "General Error",JOptionPane.*ERROR\_MESSAGE*);

}

}

});

btnshow.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

btnshow.setBounds(10, 259, 217, 25);

contentPane.add(btnshow);

}

}

## Κλάση Income\_Outcome.class

**package** omada15.softeng.teipir;

**import** java.awt.Color;

**public** **class** Section **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField txtsection;

**private** JScrollPane scrollPaneinside;

**private** JList<String> insidelist;

**private** JScrollPane scrollPaneoutside;

**private** JList<String> outsidelist;

**private** JLabel lblsection;

**private** ArrayList<String> namesinside;

**private** ArrayList<String> namesoutside;

**private** JButton btninside;

**private** JButton btnoutside;

**private** JButton btnok;

**private** ArrayList<String> afmoutside;

**private** ArrayList<String> afminside;

**private** Integer sect\_id;

**public** Section() {

setTitle("Company Management");

setIconImage(Toolkit.*getDefaultToolkit*().getImage(Section.**class**.getResource("/com/sun/java/swing/plaf/windows/icons/Computer.gif")));

setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);

setBounds(100, 100, 546, 423);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

lblsection = **new** JLabel("Τμήμα");

lblsection.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

lblsection.setBounds(10, 11, 56, 30);

contentPane.add(lblsection);

txtsection = **new** JTextField();

txtsection.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

txtsection.setBounds(76, 11, 151, 30);

contentPane.add(txtsection);

txtsection.setColumns(10);

insidelist = **new** JList();

insidelist.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

insidelist.setSelectionForeground(**new** Color(204, 0, 51));

}

});

insidelist.setSelectionMode(ListSelectionModel.*SINGLE\_SELECTION* );

insidelist.setSelectedIndex(-1);

outsidelist = **new** JList();

outsidelist.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

outsidelist.setSelectionForeground(**new** Color(0, 128, 0));

}

});

outsidelist.setSelectionMode(ListSelectionModel.*SINGLE\_SELECTION* );

outsidelist.setSelectedIndex(-1);

scrollPaneinside = **new** JScrollPane(insidelist);

scrollPaneinside.setVerticalScrollBarPolicy(ScrollPaneConstants.*VERTICAL\_SCROLLBAR\_ALWAYS*);

scrollPaneinside.setHorizontalScrollBarPolicy(ScrollPaneConstants.*HORIZONTAL\_SCROLLBAR\_AS\_NEEDED*);

scrollPaneinside.setBounds(10, 77, 217, 200);

contentPane.add(scrollPaneinside);

scrollPaneoutside = **new** JScrollPane(outsidelist);

scrollPaneoutside.setVerticalScrollBarPolicy(ScrollPaneConstants.*VERTICAL\_SCROLLBAR\_ALWAYS*);

scrollPaneoutside.setHorizontalScrollBarPolicy(ScrollPaneConstants.*HORIZONTAL\_SCROLLBAR\_AS\_NEEDED*);

scrollPaneoutside.setBounds(303, 77, 217, 200);

contentPane.add(scrollPaneoutside);

btninside = **new** JButton("<<");

btninside.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(outsidelist.getSelectedIndex()<0){

JOptionPane.*showMessageDialog*(**null**, "Επιλέξτε πρώτα έναν υπάλληλο", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

**else**{

Connection conn = **null**;

Statement stmt = **null**;

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

String x = (String) (afmoutside.toArray())[outsidelist.getSelectedIndex()];

stmt.executeUpdate("UPDATE company SET section\_id = " + sect\_id + " WHERE section\_id = 1 AND afm = " + Integer.*parseInt*(x));

stmt.close();

conn.close();

sundesi(sect\_id);

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

}

}

});

btninside.setForeground(**new** Color(0, 128, 0));

btninside.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

btninside.setBounds(237, 134, 56, 30);

contentPane.add(btninside);

btnoutside = **new** JButton(">>");

btnoutside.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**if**(insidelist.getSelectedIndex()<0){

JOptionPane.*showMessageDialog*(**null**, "Επιλέξτε πρώτα έναν υπάλληλο", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

**else**{

Connection conn = **null**;

Statement stmt = **null**;

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

String x = (String) (afminside.toArray())[insidelist.getSelectedIndex()];

stmt.executeUpdate("UPDATE company SET section\_id = 1 WHERE section\_id = " + sect\_id + " AND afm = " + Integer.*parseInt*(x));

stmt.close();

conn.close();

sundesi(sect\_id);

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

}

}

});

btnoutside.setForeground(**new** Color(204, 0, 51));

btnoutside.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

btnoutside.setBounds(237, 175, 56, 30);

contentPane.add(btnoutside);

btnok = **new** JButton("OK");

btnok.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(txtsection.getText().equals("")){

JOptionPane.*showMessageDialog*(**null**, "Εισάγετε το Όνομα του Τμήματος!", "Error", JOptionPane.*ERROR\_MESSAGE*);

}

**else**{

Connection conn = **null**;

Statement stmt = **null**;

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

String x = txtsection.getText();

stmt.executeUpdate("UPDATE sections SET name = '" + x + "' WHERE section\_id = " + sect\_id);

stmt.close();

conn.close();

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

dispose();

}

}

});

btnok.setFont(**new** Font("Tahoma", Font.*BOLD*, 14));

btnok.setBounds(237, 314, 56, 41);

contentPane.add(btnok);

}

**public** **void** sundesi(Integer thesi){

namesinside = **new** ArrayList<String>();

namesoutside = **new** ArrayList<String>();

afminside = **new** ArrayList<String>();

afmoutside = **new** ArrayList<String>();

Connection conn = **null**;

Statement stmt = **null**;

Statement stmt2 = **null**;

ResultSet rs = **null**;

**try**{

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost:3306/companyapp";

String connectionUser = "root";

String connectionPassword = "6DhYFs";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

stmt = conn.createStatement();

stmt2 = conn.createStatement();

**if** (thesi==0){

Integer i = 1;

rs = stmt.executeQuery("SELECT section\_id FROM sections");

**while**(rs.next()){

**if**(rs.getString("section\_id").equals(i.toString())){

i = i + 1;

}

**else**{

**break**;

}

}

thesi=i;

stmt2.executeUpdate("INSERT INTO sections values(" + thesi + ", '')");

}

rs = stmt.executeQuery("SELECT name, surname, general.afm FROM general, company WHERE general.afm = company.afm AND section\_id = " + thesi);

**while**(rs.next()){

String name = rs.getString("name") + " " + rs.getString("surname");

namesinside.add(name);

String afm = rs.getString("afm");

afminside.add(afm);

}

insidelist.setModel(**new** AbstractListModel() {

**public** **int** getSize() {

**return** (namesinside.toArray()).length;

}

**public** Object getElementAt(**int** index) {

**return** (namesinside.toArray())[index];

}

});

rs = stmt.executeQuery("SELECT name, surname, general.afm FROM general, company WHERE general.afm = company.afm AND section\_id = 1");

**while**(rs.next()){

String name = rs.getString("name") + " " + rs.getString("surname");

namesoutside.add(name);

String afm = rs.getString("afm");

afmoutside.add(afm);

}

outsidelist.setModel(**new** AbstractListModel() {

**public** **int** getSize() {

**return** (namesoutside.toArray()).length;

}

**public** Object getElementAt(**int** index) {

**return** (namesoutside.toArray())[index];

}

});

rs = stmt.executeQuery("SELECT name FROM sections WHERE section\_id = " + thesi);

**while**(rs.next()){

txtsection.setText(rs.getString("name"));

}

stmt.close();

stmt2.close();

conn.close();

sect\_id = thesi;

}**catch**(Exception e1){

JOptionPane.*showMessageDialog*(**null**, "Unable to Connect to Database!", "Error!",JOptionPane.*ERROR\_MESSAGE*);

}

}

}

# Παράρτημα Β: Επιπλέον Στοιχεία

Το πρόγραμμα εκτελέσθηκε με τη μέθοδο Debugging και δεν παρουσίασε ιδιαίτερο πρόβλημα καθώς επίσης και με κώδικα unit testing χρόνου και ορθή επιστροφή δεδομένων.

package omada15.softeng.teipir;

import static org.junit.Assert.\*;

import org.junit.Test;

public class MainFrameTest {

@Test(timeout=9999)

public void testSearchemployee() {

MainFrame mf = new MainFrame("admin");

mf.searchemployee("Ιωάννης");

}

@Test(timeout=9999)

public void testFillsectionarea() {

MainFrame mf = new MainFrame("admin");

mf.fillsectionarea();

}

@Test(timeout=9999)

public void testGetname() {

MainFrame mf = new MainFrame("admin");

mf.getname("admin");

}

}

package omada15.softeng.teipir;

import static org.junit.Assert.\*;

import org.junit.Test;

public class ProfileTest {

@Test(timeout=9999)

public void testDeleteprofile() {

Profile p = new Profile("");

p.deleteprofile();

}

@Test(timeout=9999)

public void testCreateconvert() {

Profile p = new Profile("");

p.createconvert();

}

@Test(timeout=9999)

public void testSaveprofile() {

Profile p = new Profile("");

p.saveprofile();

}

}

package omada15.softeng.teipir;

import org.junit.Test;

public class SectionTest {

@Test(timeout=9999)

public void testSundesi() {

Section s = new Section();

s.sundesi(0);

s.sundesi(3);

}

}