**Project στο Εργαστήριο Βάσεις Δεδομένων**

Έτος 2020 – 2021

*Staff Evaluation*

Καβούλας Αλέξανδρος

AM 1067498

Μηλιώνης Βασίλης

AM 1067415

***Περιεχόμενα:***

[Εισαγωγή: 3](#_Toc64202576)

[Λόγοι επιλογής Qt: 3](#_Toc64202577)

[Μέρος 1ο – Βάση δεδομένων (backend): 4](#_Toc64202578)

[ΣΧΗΜΑΤΙΚΟ: 5](#_Toc64202579)

[ΠΑΡΑΔΟΧΕΣ: 6](#_Toc64202580)

[Επιπλέον σημεία άξια αναφοράς: 7](#_Toc64202581)

[Μέρος 2ο – GUI (frontend) 9](#_Toc64202582)

[Connect: 9](#_Toc64202583)

[Evaluator: 11](#_Toc64202584)

[Employee: 15](#_Toc64202585)

[Admin: 18](#_Toc64202586)

[Manager: 21](#_Toc64202587)

[ΤΙ ΘΑ ΚΑΝΑΜΕ ΔΙΑΦΟΡΕΤΙΚΑ 25](#_Toc64202588)

[Πηγές: 26](#_Toc64202589)

# Εισαγωγή:

Στο παρόν έγγραφο θα αναλύσουμε τις παραδοχές και τη πορεία υλοποίησης της εργασίας για το εργαστήριο «Βάσεις Δεδομένων». Η εργασία έχει ως στόχο την υλοποίηση μιας βάσης δεδομένων, που μπορεί να χρησιμοποιηθεί για την αξιολόγηση του προσωπικού εταιρειών. Αποτελείται από δύο επιμέρους κομμάτια, το backend, που θα είναι η ίδια η βάση, υλοποιημένη σε MySQL και το frontend που θα είναι η διεπαφή που θα χρησιμοποιούν οι χρήστες. Η διεπαφή υλοποιήθηκε σε c++ με τη βοήθεια του Qt, open source IDE και βιβλιοθήκη, με το οποίο έχουν σχεδιαστεί εφαρμογές όπως το Google Earth (για linux) και [άλλες](https://en.wikipedia.org/wiki/Qt_(software)#Applications_using_Qt).

# Λόγοι επιλογής Qt:

Όπως αναφέραμε και προηγουμένως το Qt είναι εργαλείο που φαίνεται να έχει χρησιμοποιηθεί σε αρκετές εφαρμογές. Αυτός όμως δεν ήταν ο μόνος λόγος για τον οποίο το επιλέξαμε, αντί το NetBeans το οποίο μας προτάθηκε. Την απόφασή μας επηρέασε πολύ το γεγονός ότι είχαμε ήδη μια καλή βάση με τη c++, αφού είναι η γλώσσα στην οποία έχουμε γράψει τις περισσότερες από τις εργασίες μας, ενώ παράλληλα εντοπίσαμε εύκολα το documentation, απαραίτητο για την σωστή ανάπτυξη του GUI. Πέρα από το αρκετά αναλυτικό documentation που χρησιμοποιήσαμε κατά βάση, υπήρχαν και διαθέσιμα video – tutorial, που ήταν αρκετά ώστε να μας δώσουν μια μεγαλύτερη άνεση στο χειρισμό, τόσο του software αυτού καθαυτού όσο και στις βασικές βιβλιοθήκες που χρησιμοποιήσαμε. Τα βίντεο αυτά θα τα παραθέσουμε στο τέλος της αναφοράς, μαζί με ό,τι πηγές τελικά χρησιμοποιήσαμε, ώστε να καταλήξουμε στην τελική μορφή της εργασίας.

# Μέρος 1ο – Βάση δεδομένων (backend):

Για την οργάνωση της βάσης μας, ακολουθήσαμε τη διαδικασία που μας προτάθηκε από τους καθηγητές του εργαστηρίου, χρησιμοποιώντας παράλληλα την εμπειρία που είχαμε αποκτήσει από τις εργαστηριακές ασκήσεις. Έτσι, ξεκινήσαμε διαβάζοντας προσεκτικά την εκφώνηση της εργασίας, ώστε να οργανώσουμε καλύτερα το πλάνο υλοποίησης του project αλλά και να κατανοήσουμε πλήρως τις απαιτήσεις του. Καταρχάς λοιπόν σχεδιάσαμε το ER διάγραμμα, το οποίο κατευθείαν χρησιμοποιήσαμε ώστε να εξάγουμε το σχεσιακό μοντέλο που θα ακολουθήσουμε. Το σχεσιακό είναι απαιτούμενο και της εκφώνησης και το παραθέτουμε:

## ΣΧΗΜΑΤΙΚΟ:

Έπειτα, προχωρήσαμε στην οργάνωση και υλοποίηση των συναρτήσεων create table, κάναμε insert ικανοποιητικό όγκο δεδομένων στη βάση, ώστε να μπορούμε έπειτα να ελέγχουμε πιο εύκολα και αποτελεσματικά τα stored procedure και τα triggers τα οποία υλοποιούσαμε. Κατά τη διαδικασία αυτή κάναμε παραδοχές που αναφέρουμε παρακάτω:

## ΠΑΡΑΔΟΧΕΣ:

1. Δεν θεωρούμε του manager κάθε εταιρείας ως employee, οπότε αυτοί δεν θα μπορούν να αξιολογηθούν από το σύστημα.
2. Θεωρήσαμε επίσης ότι δύο employees μπορεί να έχουν υλοποιήσει κάποιο κοινό project.
3. Υποθέσαμε επίσης ότι ο κάθε evaluator μπορεί να ανακοινώσει θέσεις εργασίας, για τις οποίες όμως θα είναι υπεύθυνος μόνο αυτός. Με άλλα λόγια την αξιολόγηση θα την διεξάγει εκείνος και όχι κάποιος άλλος evaluator.
4. Ο κάθε manager ανήκει σίγουρα σε κάποια εταιρεία.
5. Φτιάξαμε 3 διαφορετικούς πίνακες 1 για τη κάθε φάση της αξιολόγησης (eval\_phase1,2,3).
6. Δημιουργήσαμε έναν ακόμα πίνακα report ώστε να υπάρχει άμεση σύνδεση του employee με τον manager.
7. Σε περίπτωση που έχει παρέλθει η ημερομηνία υποβολής αίτησης (sub\_date), τότε δεν μπορούν να διαγραφούν οι αιτήσεις που έχει κάνει ο employee.
8. Άμα δεν έχει παρέλθει η ημερομηνία υποβολής αίτησης δεν μπορεί ο evaluator να συμπληρώσει κάποια από τα phases. Αυτά, τόσο και ο συνολικός βαθμός της αξιολόγησης θα πρέπει να συμπληρωθούν μετά το submission\_date
9. Ο συνολικός βαθμός του evaluation, μπαίνει αυτόματα, μόλις συμπληρωθούν και οι 3 φάσεις αξιολόγησης
10. Ο βαθμός του evalutator είναι -1 αντί για NULL (μας διευκολύνει στην υλοποίηση του GUI).

## Επιπλέον σημεία άξια αναφοράς:

Επιπρόσθετα των παραδοχών, καθώς υλοποιούσαμε τη βάση, πήραμε μερικές αποφάσεις, που επηρεάζουν το τελικό αποτέλεσμα. Κυριότερη, κατά τη γνώμη μου, ήταν το γεγονός ότι επιλέξαμε να έχουμε τελικά τρεις διαφορετικούς πίνακες, έναν για κάθε φάση της αξιολόγησης. Αυτή μας η απόφαση βασίστηκε κυρίως στο γεγονός ότι η κανονική μορφή 1NF επιβάλει σε έναν πίνακα, αυτός να έχει πεδία τα οποία να είναι διαφορετικά μεταξύ τους, γεγονός το οποίο δεν θα ικανοποιούταν αν διατηρούσαμε στον ίδιο πίνακα τους τρεις βαθμούς (έναν για κάθε φάση) που ζητείται από την εκφώνηση.

Η προηγούμενη απόφαση φάνηκε να διευκολύνει την υλοποίηση της βάσης, όμως πήραμε και αποφάσεις που τελικά δεν ήταν τόσο βοηθητικές. Μια από αυτές ήταν η επιλογή μας να διατηρήσουμε ένα διαφορετικό κλειδί (auto increment) για κάθε φάση της αξιολόγησης. Το κλειδί τελικά δεν χρησιμοποιήθηκε πουθενά. Αυτή μας η επιλογή στηρίχτηκε κυρίως στο γεγονός ότι ένα auto\_increment key, δημιουργεί [μικρότερο overhead](https://stackoverflow.com/questions/4103433/why-use-an-auto-incrementing-primary-key-when-other-unique-fields-exist) από ένα κλειδί τύπου string. Σε δεύτερη ανάγνωση όμως, παρατηρήσαμε ότι μπορούσαμε να χρησιμοποιήσουμε το foreign key από τον πίνακα evaluation result ως primary key, αφού οι οντότητες phase1,2 και 3 είναι ασθενείς οντότητες. Παρόμοια λογική μπορούσαμε να είχαμε ακολουθήσει και για τον πίνακα evaluation\_result, ο οποίος θα μπορούσε να θεωρηθεί ασθενής οντότητα, που εξαρτάται από το request\_evaluation που κάνει ο χρήστης (προφανώς δεν μπορεί να υπάρξει αξιολόγηση employee, αν εκείνος δεν κάνει αίτηση για μια θέση). Αυτές οι επιλογές νομίζουμε θα διευκόλυναν κυρίως την υλοποίηση μερικών stored procedure και trigger.

Βέβαια, μεγαλύτερη δυσκολία συναντήσαμε στην υλοποίηση stored procedure και trigger, στα οποία χρειαζόταν να χρησιμοποιήσουμε πληροφορία που αφορούσε κάποιον employee και manager της εταιρείας στην οποία εργάζεται, αφού δεν προνοήσαμε, ώστε να δημιουργήσουμε έναν πίνακα που να τους συνδέει. Κάτι τέτοιο όμως θα προσέθετε πληροφορία που επαναλαμβάνεται (έστω και έμμεσα, λόγω του σχήματος employee -> company -> manager), οπότε θα χρειαζόταν ιδιαίτερη προσοχή σε insert, update και delete των υπαλλήλων και manager.

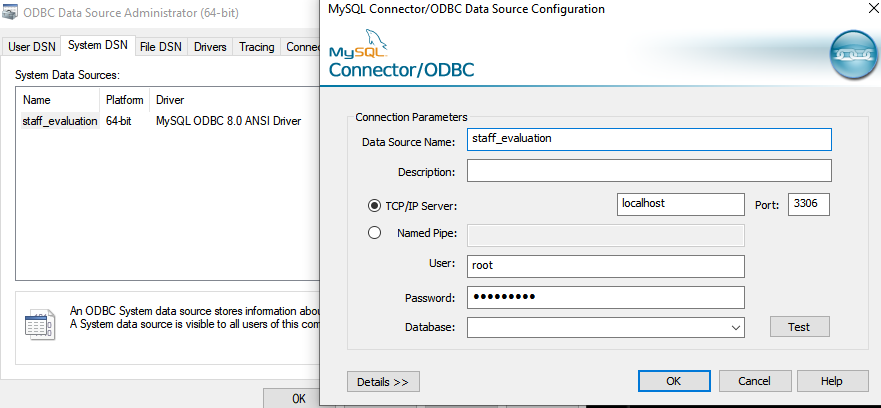
Τέλος, συζητώντας, συμπεράναμε ότι ο τρόπος με τον οποίο υλοποιήσαμε το log table που ζητείται από την εκφώνηση, δεν είναι ο ιδανικότερος. Χρησιμοποιήσαμε global μεταβλητή που να περιέχει το username του χρήστη, ο οποίος είναι συνδεδεμένος στη βάση, ώστε τα trigger που έχουμε σχεδιάσει για να ενημερώνουν αυτόν τον πίνακα, να παίρνουν το όνομα του χρήστη από αυτή τη μεταβλητή. Προφανώς αυτό δημιουργεί πρόβλημα αν δύο χρήστες συνδεθούν παράλληλα στη βάση. Πρώτη λύση που σκεφτήκαμε ήταν το όνομα του user που είναι συνδεδεμένος στη βάση να αποθηκεύεται σε κάθε παράθυρο που προγραμματίσαμε στο GUI, ώστε να ενημερώνουμε την global μεταβλητή, πριν από κάθε ενέργεια που πρόκειται να ενεργοποιήσει trigger. Γρήγορα όμως καταλάβαμε πως αυτή μας η ιδέα, αν και θα μας διευκόλυνε ούτως ή αλλιώς στην υλοποίηση του GUI, εν τέλει δεν θα έλυνε τελείως το πρόβλημά μας. Τι θα γινόταν αν δυο χρήστες πατούσαν κάποιο κουμπί, που θα ενημέρωνε το log table, ταυτόχρονα; Σε αυτό το σημείο κάναμε κάποια σκέψη που όμως δεν προσπαθήσαμε να υλοποιήσουμε, λόγω έλλειψης χρόνου, ενώ παράλληλα θεωρήσαμε ότι ξεφεύγει από τον σκοπό της εργασίας, αφού αυτή περιλάμβανε [critical section](https://dev.mysql.com/doc/refman/8.0/en/innodb-locking-reads.html) στην MySQL, το οποίο περισσότερο ταιριάζει στο μάθημα των λειτουργικών συστημάτων.

# Μέρος 2ο – GUI (frontend)

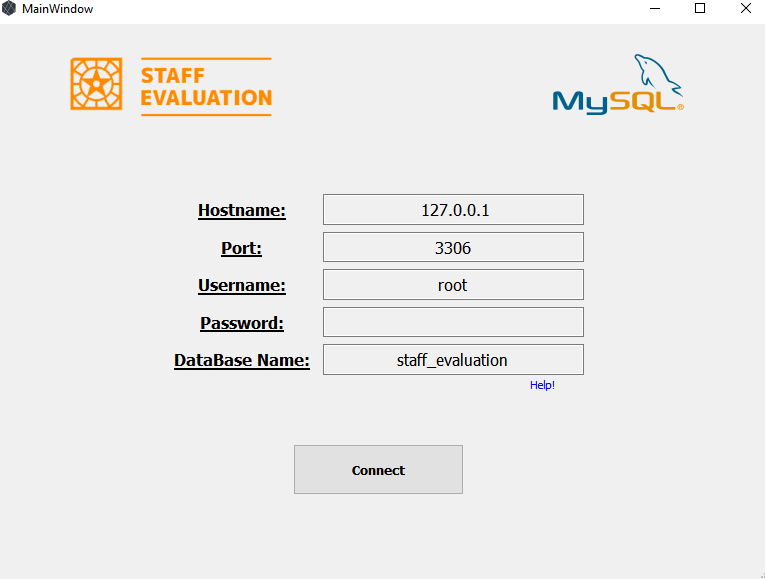
Οφείλουμε να ομολογήσουμε, το δεύτερο μέρος ήταν εκείνο το οποίο μας τράβηξε περισσότερο το ενδιαφέρον, αφού μπορούσαμε άμεσα κιόλας να δούμε όλα όσα είχαμε σχεδιάσει να παίρνουν μορφή. Δεν θα επεκταθούμε σε πολλές λεπτομέρειες που αφορούν τον κώδικα σε c++, αλλά θα παρουσιάσουμε κυρίως το αποτέλεσμα.

Πριν παραθέσουμε οποιαδήποτε screenshot από το GUI, θα αναφερθώ με συνοπτικά βήματα στο τρόπο με τον οποίο κάναμε install και configure το ODBC driver για σύστημα Windows 10 (64bit). Η ίδια διαδικασία περιγράφεται εδώ.

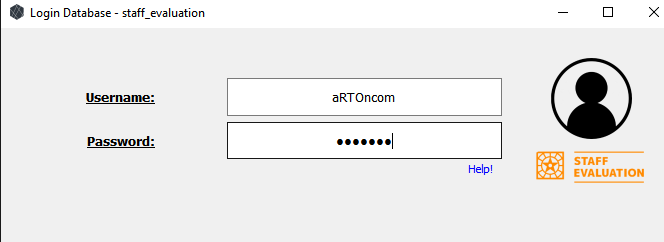
## Connect:



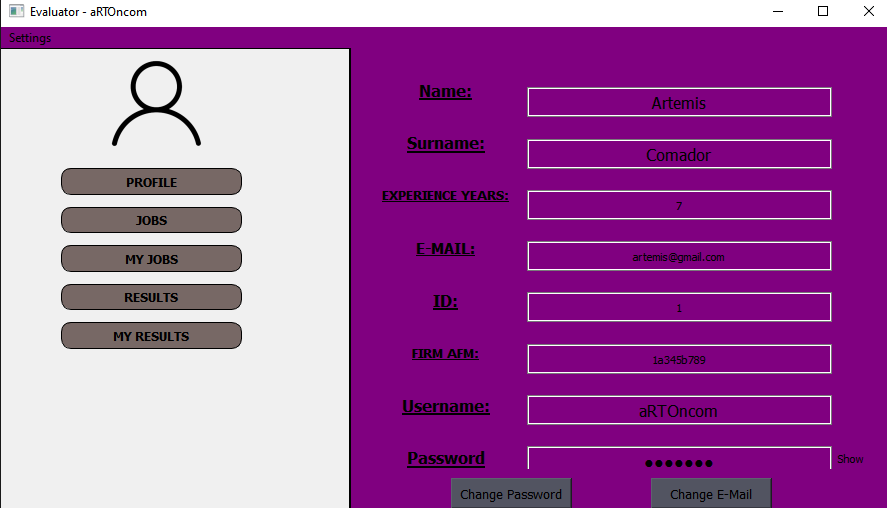
Στην παραπάνω εικόνα βλέπουμε το πρόγραμμα το οποίο χρησιμοποιήσαμε για να συνδέσουμε την βάση μας στην MySQL με το Qt Creator και τα στοιχεία που προτείνουμε να βάλετε, εκτός του κωδικού, ώστε να συνδέεστε με τα default στοιχεία που έχουμε βάλει στο παρακάτω παράθυρο, πιο εύκολα.



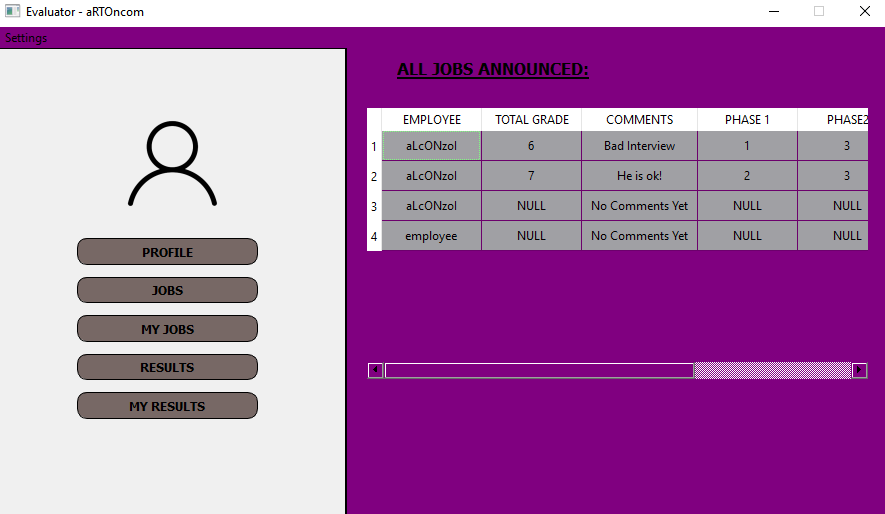
Παραπάνω βλέπουμε τον τρόπο με τον οποίο συνδεόμαστε στην βάση.

Το username που φαίνεται, είναι αξιολογητής και στην συνέχεια περιγράφονται οι διεργασίες που μπορεί να κάνει αυτός.

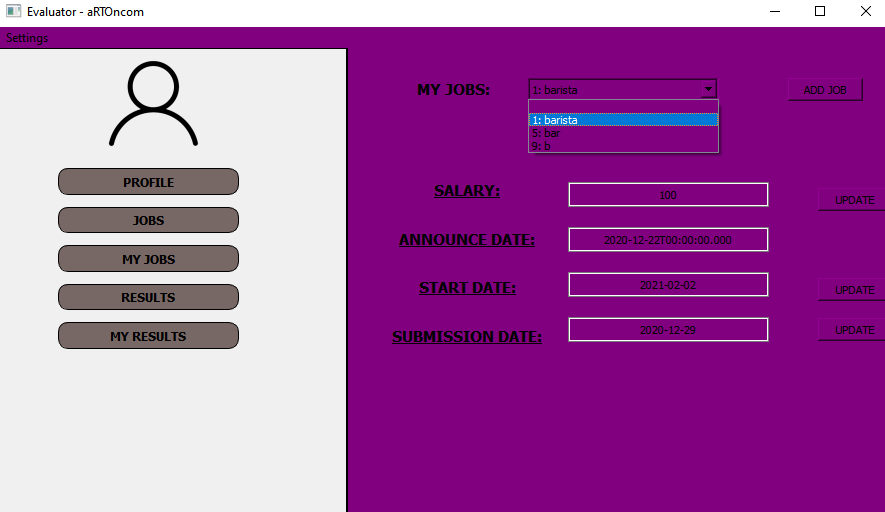
## Evaluator:



Πατώντας το κουμπί “PROFILE” βλέπουμε τα στοιχεία του evaluator που έχει συνδεθεί στην βάση. Από αυτά τα στοιχεία ο evaluator μπορεί να επεξεργαστεί μόνο τον κωδικό του και το email του.



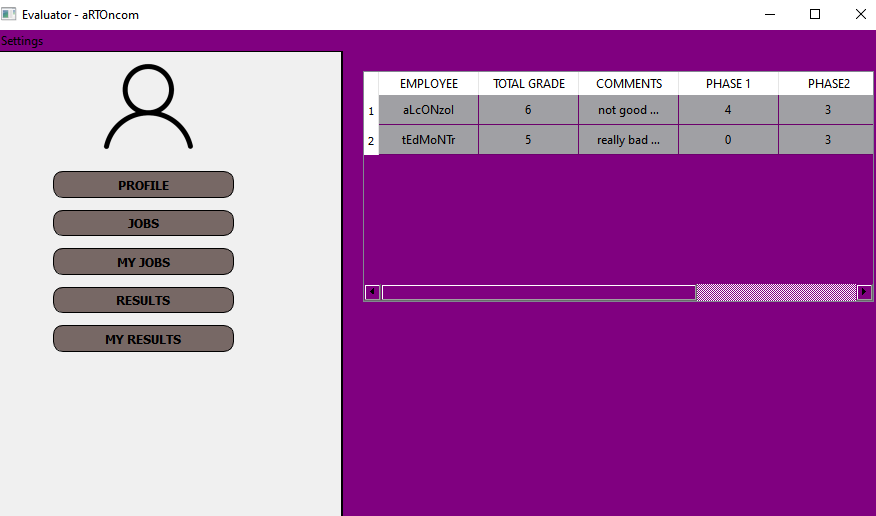
Εδώ εμφανίζονται όλες οι θέσεις εργασίες με όλα τα στοιχεία που τoυς. Ο αξιολογητής δεν μπορεί να επεξεργαστεί τίποτα στον πίνακα αυτό.

Ενώ παρακάτω βλέπουμε όλες τις θέσεις εργασίας που έχει αναρτήσει ο αξιολογητής που είναι συνδεδεμένος στη βάση. Εκεί μπορεί να επεξεργαστεί τον μισθό , την ημερομηνία έναρξης και την ημερομηνία υποβολής της αίτησης. Επίσης βλέπουμε ένα κουμπί με το οποίο μπορεί να υποβάλλει μία νέα θέση εργασίας(ADD JOB).

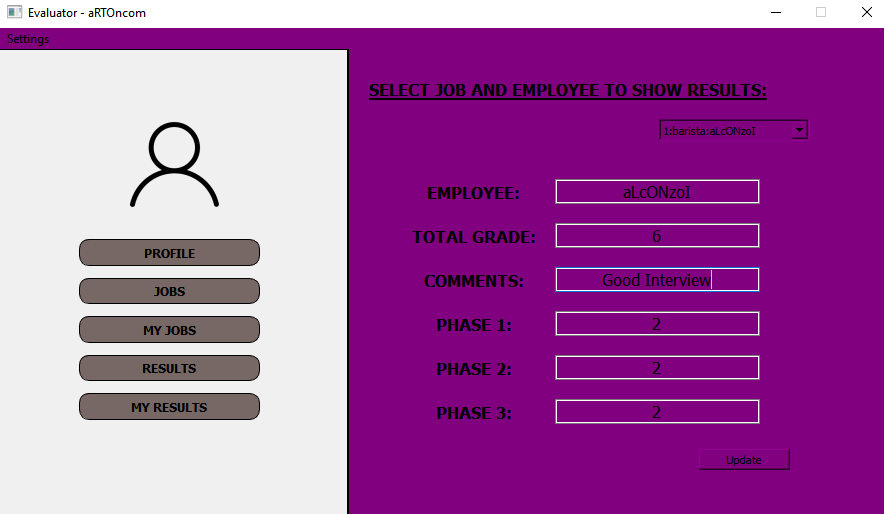
Ο αξιολογητής έχει την υποχρέωση να συμπληρώσει τα στοιχεία της θέσης εργασίας, ενώ η βάση αναλαμβάνει να εισάγει αυτόματα το username του και την ημερομηνία που θα ανακοινωθεί η νέα θέση.



Στην συνέχεια ο αξιολογητής μπορεί να δει στον πίνακα όλες τις ολοκληρωμένες αξιολογήσεις μαζί με τα σχόλια και τις επιμέρους βαθμολογίες των φάσεων αναλυτικά. Ενώ στην από κάτω εικόνα βλέπει τις δικές του αξιολογήσεις.

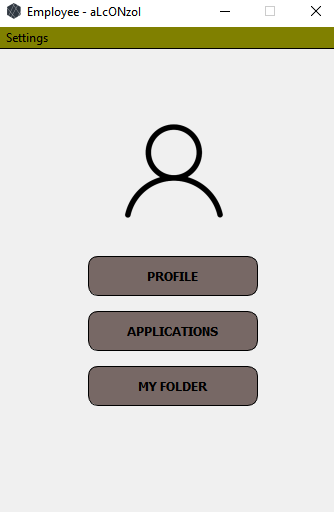


Εδώ ο αξιολογητής μπορεί να αλλάξει βαθμούς των φάσεων και τα σχόλια της καθε αξιολόγησης. Στην περίπτωση που δεν έχει περάσει η ημερομηνία υποβολής ο συνολικός βαθμός δεν αλλάζει και θα πρέπει ο ίδιος ο αξιολογητής μόλις έρθει η ώρα να πατήσει “Update” ώστε να καταχωρηθεί ο νέος συνολικός βαθμός.

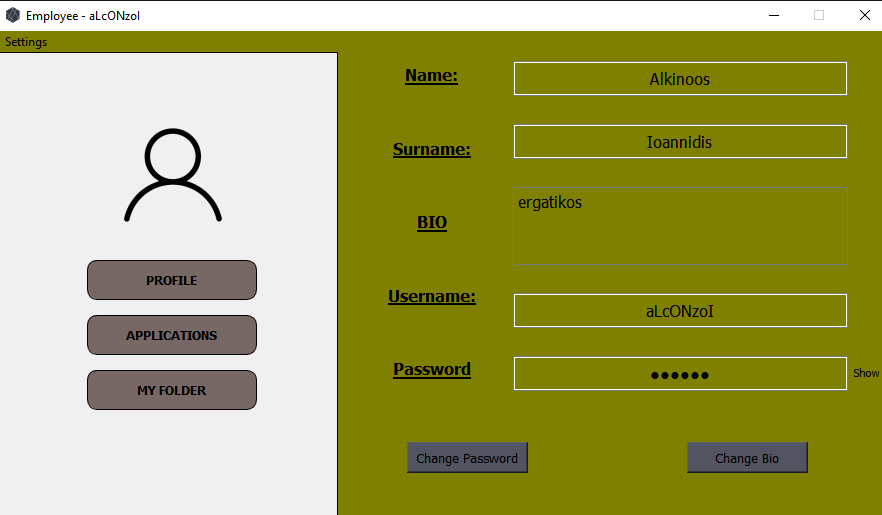


## Employee:

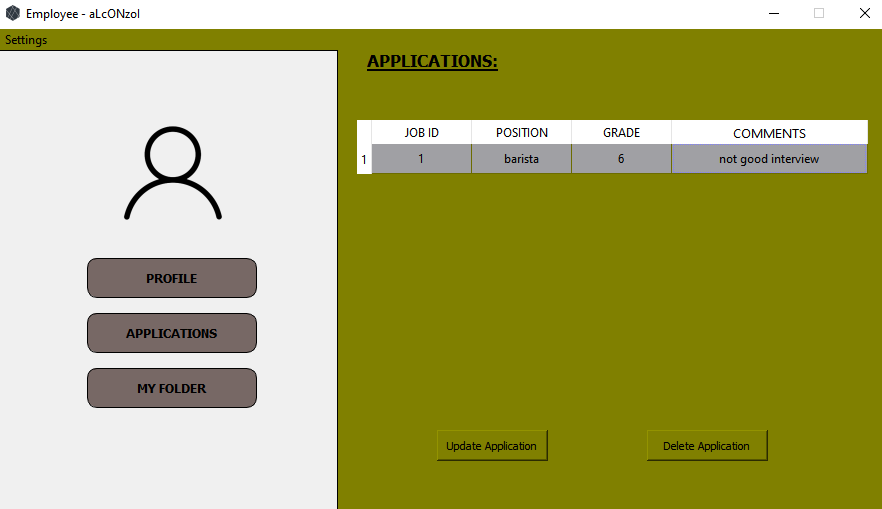
Στην περίπτωση που συνδεθεί στην βαση κάποιος employee η μορφη θα είναι η παρακάτω.



Ο employee πατώντας το κουμπί “PROFILE” μπορεί να δει τα στοιχεία του και να ενημερώσει τον κωδικό και το βιογραφικό του.



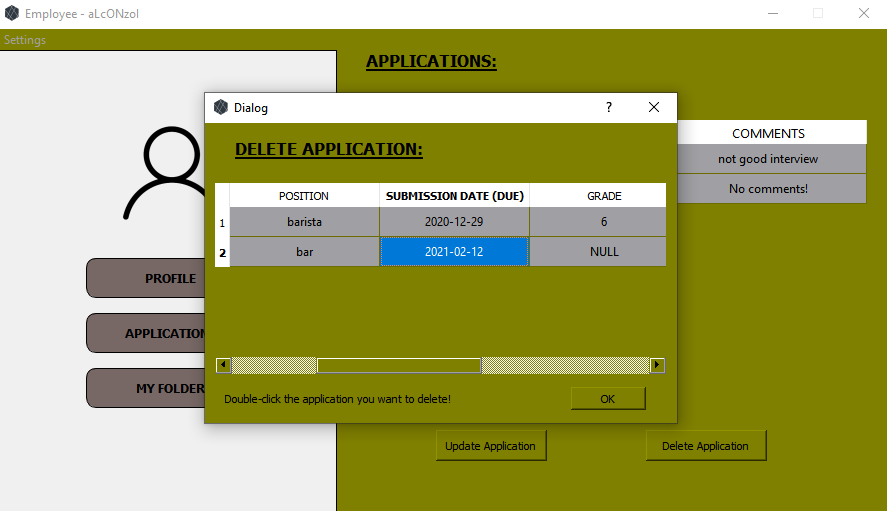
Επιλέγοντας να πατήσει το κουμπί “APPLICATIONS” τότε θα του εμφανιστούν οι αιτήσεις που έχει κάνει για τις θέσεις εργασίας , μαζί στοιχεία που τις χαρακτηρίζουν.



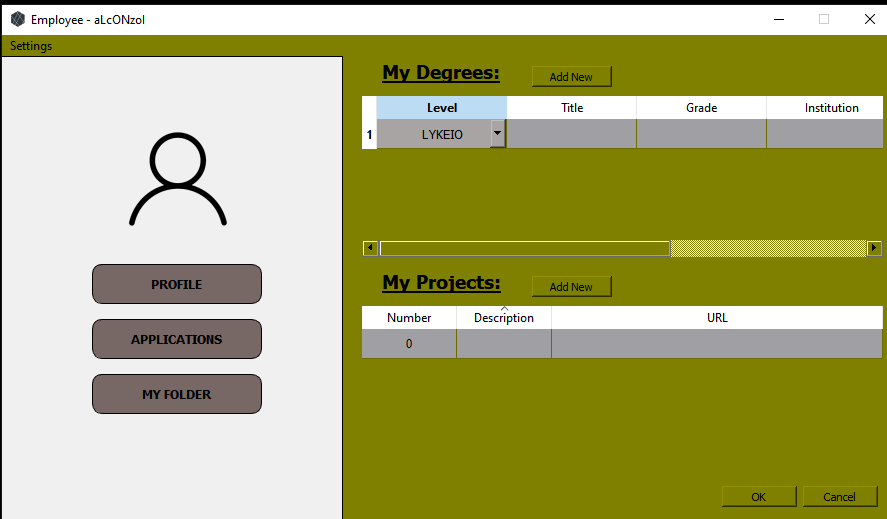
Επίσης, μέσω του κουμπιού “Update Application” μπορεί να δει τις διαθέσιμες θέσεις εργασίας και να επιλέξει άμα θέλει κάποια ώστε να κάνει μια νέα αίτηση. Επιλέγοντας την θέση εργασίας στο combobox του εμφανίζονται περισσότερα στοιχεία.



Παρόμοια μπορεί να διαγράψει μια αίτηση μέσω του “Delete Application”, όπως βλέπουμε στην παρακάτω εικόνα.

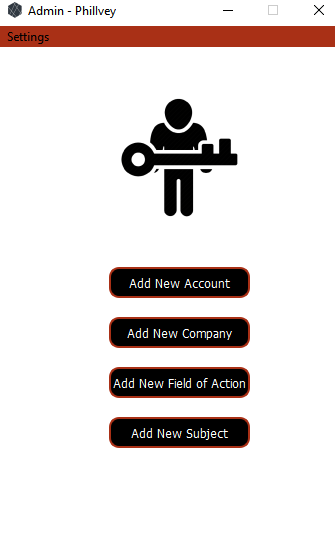


Τέλος, ο εργαζόμενος μπορεί να δει και να προσθέσει πτυχία που κατέχει και εργασίες που ενδεχομένως να έχει.

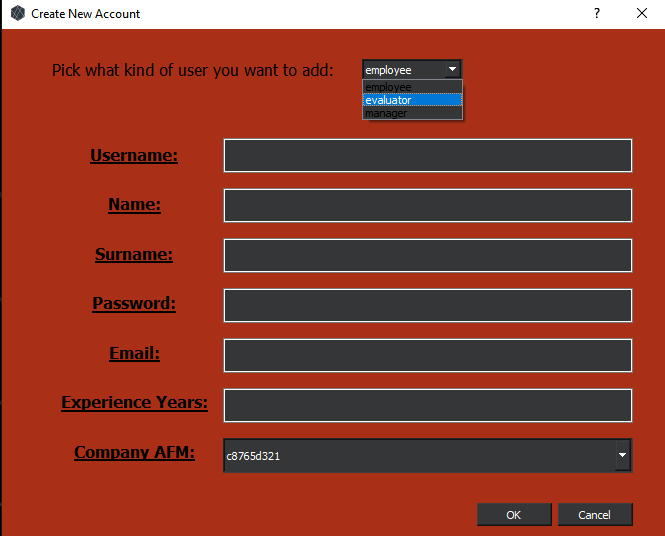


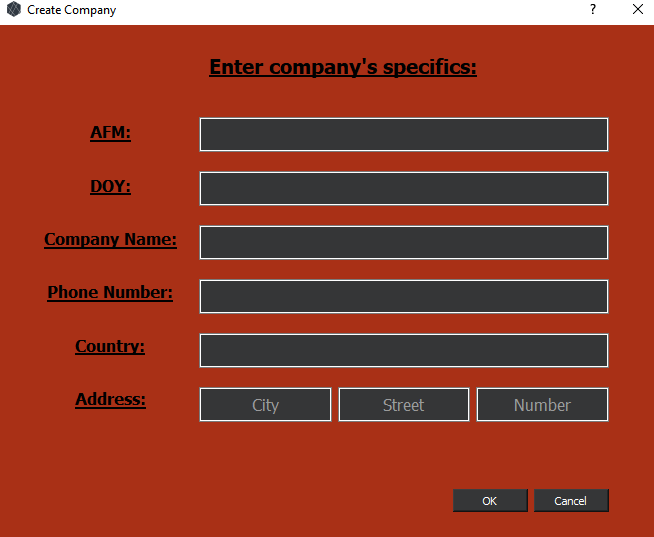
## Admin:

Στην περίπτωση που συνδεθεί κάποιος διαχειριστής το σύστημά μας θα έχει την παρακάτω μορφή.

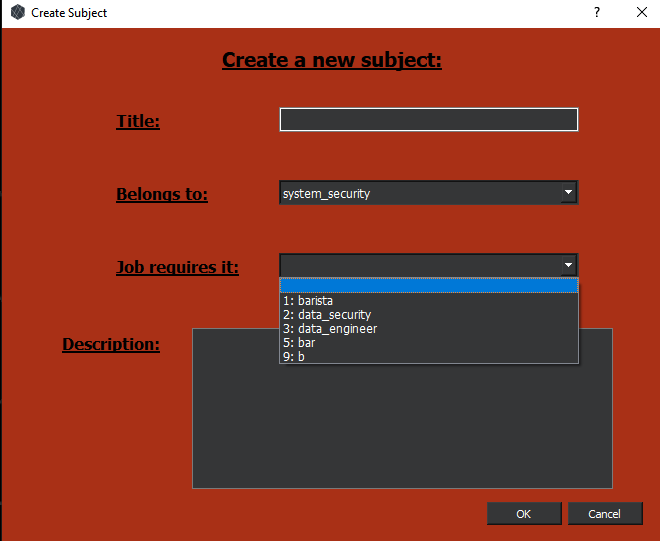


Πατώντας το κουμπί “Add New Account” του εμφανίζεται το παρακάτω παράθυρο. Εδώ ο διαχειριστής είναι υποχρεωμένος να επιλέξει την ιδιότητα του νέου χρήστη και μετά να συμπληρώσει στα κατάλληλα πεδία τα στοιχεία του.

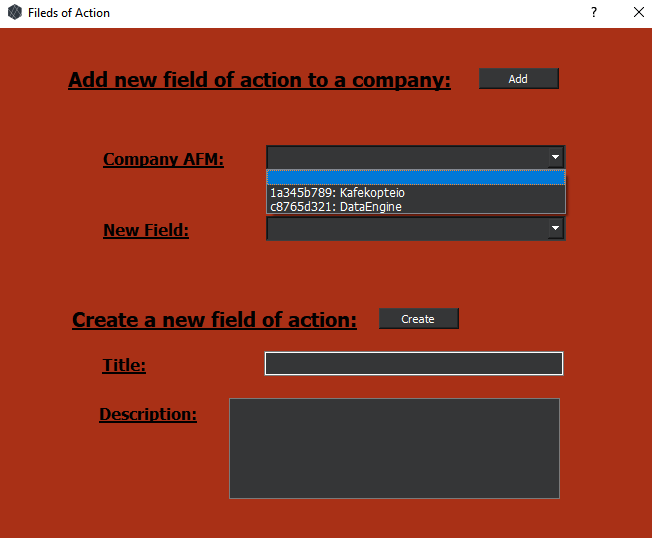


Ομοίως , πατώντας το κουμπί “Add New Company” θα πρέπει να συμπληρώσει τα στοιχεία της εταιρείας κατάλληλα.

Προχωράμε με το ίδιο μοτίβο, αυτή την φορά πατώντας το κουμπί “Add New Subject” ο διαχειριστής συμπληρώνει τα στοιχεία του αντικειμένου που θέλει να εισάγει στην βάση.

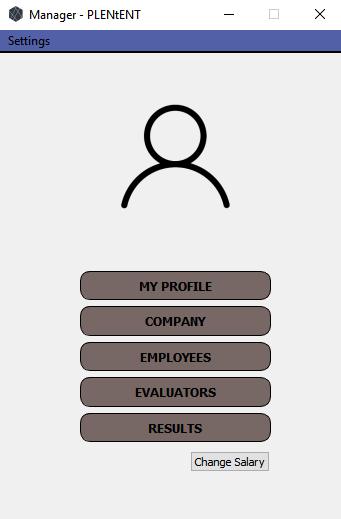


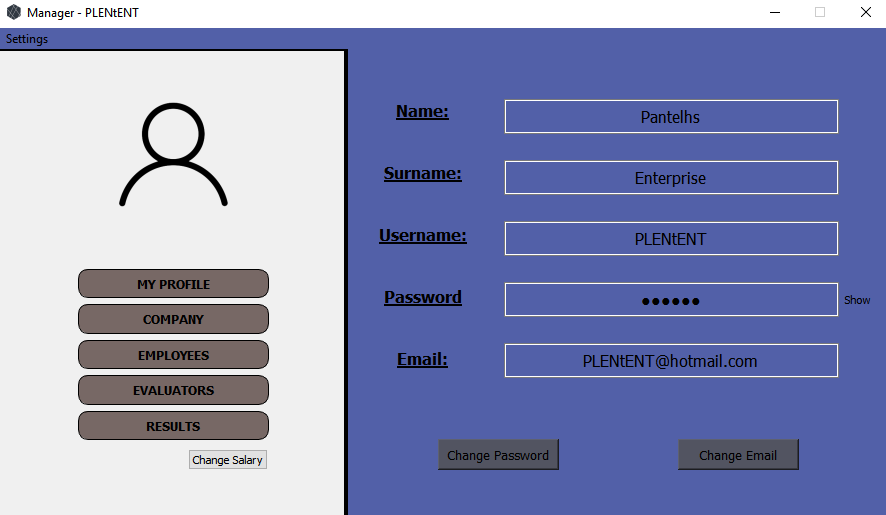
Τέλος, ο διαχειριστής μέσω του κουμπιού “Add New Field of Action” μπορεί να προσθέσει κάποια στοιχεία στην εταιρεία βάσει του ΑΦΜ της.



## Manager:

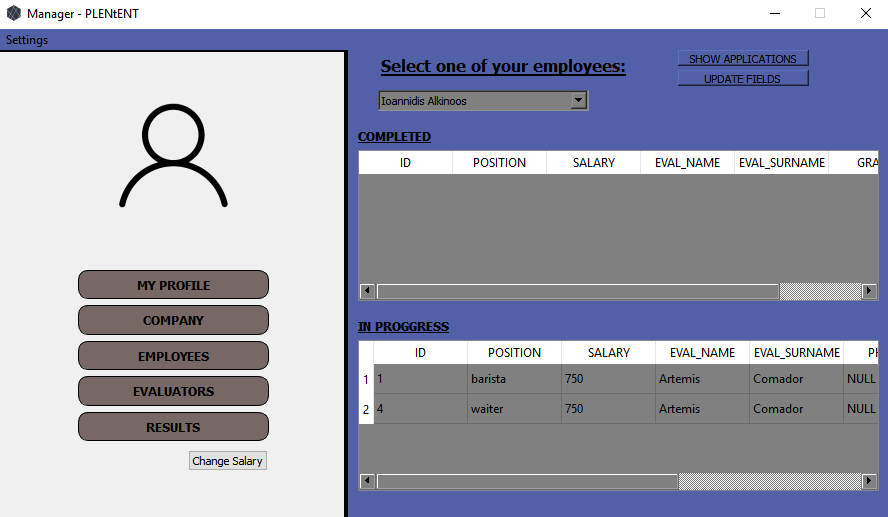
Στην περίπτωση που συνδεθεί στο σύστημα κάποιος διευθυντής η μορφή θα είναι παρακάτω.



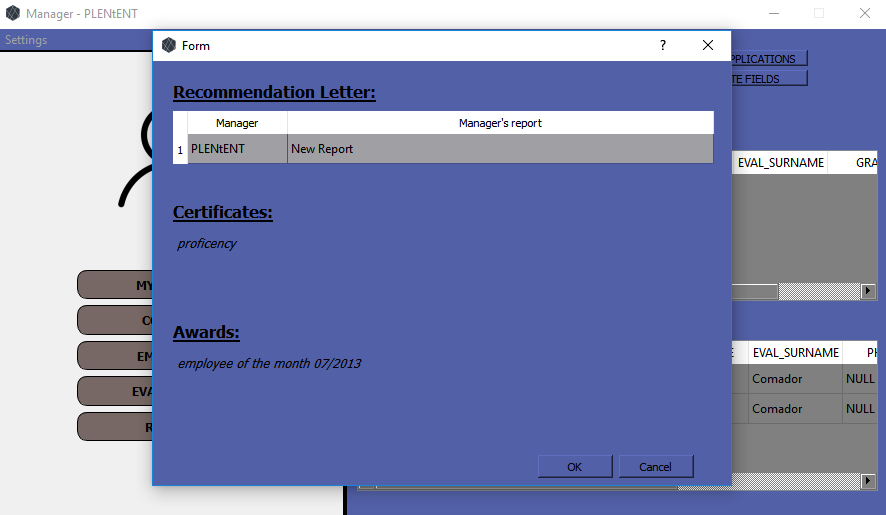
Ο διευθυντής μπορεί να δει τα στοιχεία του λογαριασμού του και να επεξεργαστεί τον κωδικό του και το email του.

Επίσης, μπορεί να δει αναλυτικά τα στοιχεία της εταιρείας που διευθύνει.

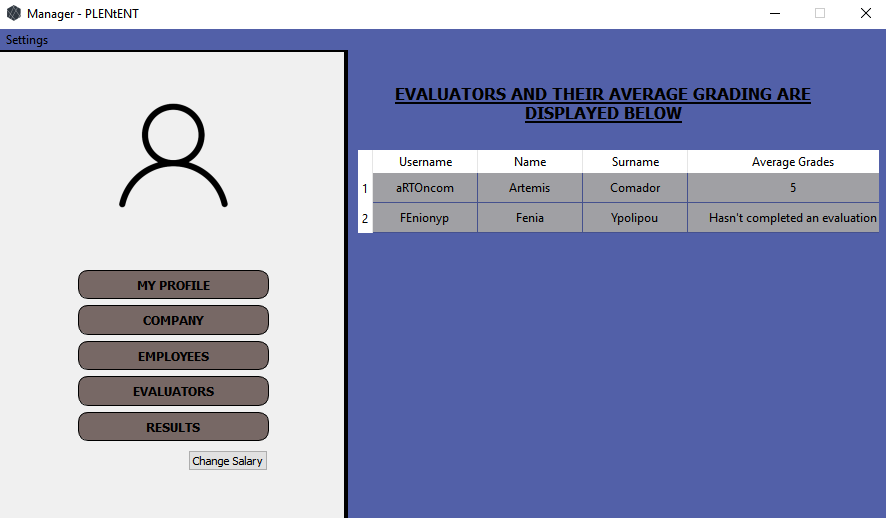
Στη συνέχεια βλέπουμε ότι μπορεί να δει τους εργαζόμενούς του και τα στοιχεία των αιτήσεων που έχουν αναρτήσει στο σύστημα. Πατώντας το κουμπί “SHOW APPLICATIONS” εμφανίζονται οι δύο πίνακες της παρακάτω εικόνας.



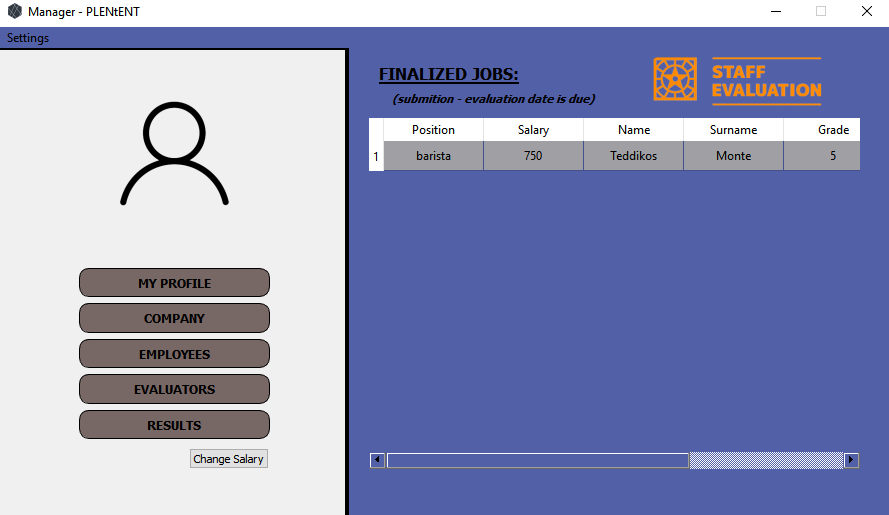
Αν πατήσει το κουμπί “UPDATE FIELDS” θα εμφανιστεί το παρακάτω παράθυρο, στο οποίο βλέπει μερικά στοιχεία του εργαζομένου που αφορούν τα βραβεία του και τα διπλώματα που ενδεχομένως να έχει. Ο διευθυντής μπορεί να επεξεργαστεί όλα τα παρακάτω πεδία.



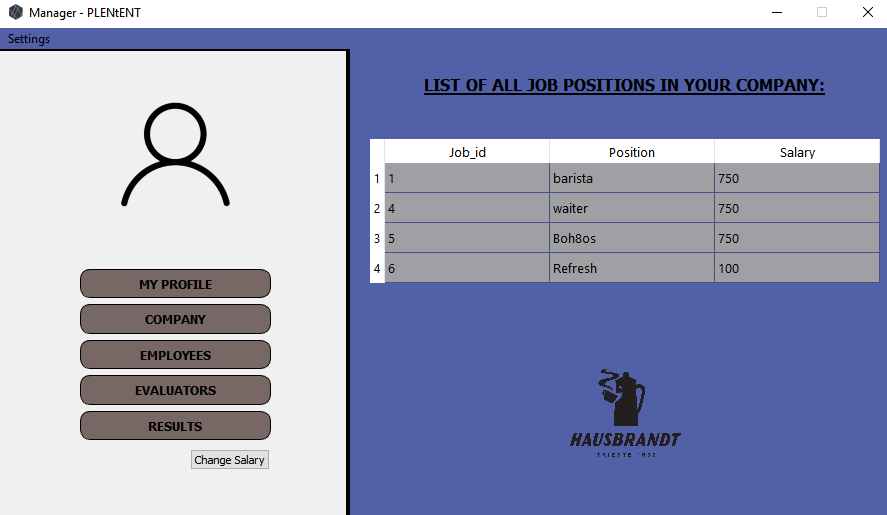
Επιλέγοντας το κουμπί “Evaluators” μπορεί να δει τα μερικά από τα στοιχεία των αξιολογητών καθώς και τον μέσο όρο που βαθμολογούν.



Στην συνέχεια , άμα πατήσει το κουμπί “RESULTS” θα εμφανιστεί ο παρακάτω πίνακας με χαρακτηριστικά όπως το position, salary, name-surname (employee) και grade.



Τέλος, πατώντας το κουμπί “Change Salary” εμφανίζεται ένας πίνακας όπου μέσω αυτού μπορεις να αλλαξεις τον μισθό της κάθε θέσης.



# ΤΙ ΘΑ ΚΑΝΑΜΕ ΔΙΑΦΟΡΕΤΙΚΑ

Καταρχάς να αναφέρουμε ότι κατά τη διάρκεια υλοποίησης της εργασίας, μαθαίναμε τα προγράμματα που χρησιμοποιήσαμε (το Qt κυρίως), αλλά παράλληλα αποκτούσαμε εμπειρία πάνω στο πως χτίζουμε μια βάση και πως τη διαχειριζόμαστε μέσα από το GUI. Συνεπώς υποκύψαμε σε λάθη, όπως το γεγονός ότι χρησιμοποιήσαμε username και κωδικούς από online generators, πράγμα που μας δυσκόλεψε αρκετά στο debugging και τον έλεγχο του κώδικα από το GUI. Πιο σημαντικό βέβαια είναι το γεγονός ότι αν και πολύς κώδικας επαναχρησιμοποιείται σε παράθυρα και σε widget (όπως τους πίνακες του GUI), δεν φτιάξαμε συναρτήσεις, ώστε να μειώσουμε τις γραμμές κώδικα. Αυτό μας εμπόδισε τελικά από το να κάνουμε refine τον κώδικα και να εντοπίσουμε πιο εύκολα τυχόν σφάλματα που μπορεί να υπάρχουν. Λάθη έγιναν προφανώς και στη βάση, με κυριότερα αυτά που αναφέρονται στην παράγραφο [εδώ](#_Επιπλέον_σημεία_άξια). Αυτά, καταλήξαμε ότι τελικά οφείλονται στο «επιπόλαιο» ER διάγραμμα από το οποίο ξεκινήσαμε και είναι σίγουρα ένα λάθος που δεν θα ξανακάνουμε, αφού τον σπαταλήσαμε πολύ χρόνο και σκέψη στην αντιμετώπιση των προβλημάτων που αυτό δημιούργησε (ίσως βοηθούσε και η κανονικοποίηση των πινάκων).

Έπειτα, θα αναφερθώ στο κομμάτι του GUI. Σε αυτό πιστεύουμε ότι σίγουρα μπορούσαμε να είχαμε αφιερώσει κάποιες μέρες, σχεδιάζοντας το interface σε ένα χαρτί ή σε κάποιο πρόγραμμα, ώστε να καταλήξουμε πιο ελκυστικό αποτέλεσμα. Τέλος, σίγουρα θα έπρεπε να διαβάσουμε καλύτερα τους διάφορους διαθέσιμους connector, τα θετικά και τα αρνητικά τους, πριν αποφασίσουμε ποιον τελικά θα χρησιμοποιήσουμε (θυμίζουμε για την c++ υπάρχει ο C++ connector και προφανώς το ODBC Driver, που χρησιμοποιήσαμε). Εμείς αν και η επιλογή μας να χρησιμοποιήσουμε τον ODBC έγινε λόγω του ότι ήταν ο πρώτος τρόπος διασύνδεσης του GUI με τη βάση που συναντήσαμε, δεν μας δημιούργησε κάποιο πρόβλημα, πέρα από το ότι έπρεπε να εγκαταστήσουμε και να κάνουμε configure τον Driver.

# Πηγές:

Link από βίντεο, τα οποία παρακολουθήσαμε πριν ξεκινήσουμε το GUI, ώστε να εξοικειωθούμε καλύτερα με το Qt:

* <https://www.youtube.com/watch?v=I96uPDifZ1w>
* <https://www.youtube.com/watch?v=txGRU7OrTZo&t=6s>
* <https://www.youtube.com/watch?v=3XE2bKUAxfw&t=299s>

Documentation από το Qt, που χρησιμοποιήσαμε κατά βάση ώστε να χτίσουμε το GUI:

* <https://doc.qt.io/qt-5/qstring.html>
* <https://doc.qt.io/qt-5/qtablewidget.html>
* <https://doc.qt.io/qt-5/qcombobox.html>
* <https://doc.qt.io/qt-5/qpixmap.html>
* <https://doc.qt.io/qt-5/signalsandslots.html>