**ΜΗΧΑΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ (Ε)**

*ΔΙΑΧΕΙΡΙΣΗ ΠΡΟΣΩΠΙΚΟΥ ΕΤΑΙΡΕΙΑΣ*

**

ΣΠΟΥΔΑΣΤΕΣ :

* ΝΑΣΤΟΣ ΝΙΚΟΣ, ΑΜ 39191
* ΖΑΝΕΤΗΣ ΓΕΩΡΓΙΟΣ, ΑΜ 38027
* ΚΑΡΑΓΙΑΝΝΗΣ ΒΙΚΕΝΤΙΟΣ, ΑΜ 39346

**ΠΕΡΙΕΧΟΜΕΝΑ**

-Εισαγωγή ............................................................................. 3

-Έντυπο προδιαγραφών απαιτήσεων λογισμικού .................. 4

-User acceptance test .............................................................. 8

-Συγκεντρωτικό σκέλος δημιουργίας λογισμικού .................. 10

-Πηγαίος κώδικας και GUI ..................................................... 11

-Debugging -Testing ............................................................... 18

-Διάγραμμα UML ................................................................... 22

**ΕΙΣΑΓΩΓΗ**

Σκοπός της εργασίας είναι η δημιουργία ενος λογισμικού ώστε να μπορούν να διαχειριστούν οι προιστάμενοι και το τμήμα προσωπικού τους εργαζόμενους μιας εταιρείας. Οπότε ουσιαστικά, οι χρήστες που θα έχουν την αρμοδιότητα να χρησιμοποιήσουν το συγκεκριμένο λογισμικό θα είναι δύο, το ΗR και οι προιστάμενοι.

Οι λειτουργίες που θα μπορούν να κάνουν οι προιστάμενοι είναι οι εξής :

* Δημιουργία / επεξεργασία / διαγραφή αξιολόγησης

Οι λειτουργίες που θα μπορεί να κάνει το τμήμα προσωπικού ( HR ) είναι:

* Δημιουργία / επεξεργασία / διαγραφή τμήματος
* Δημιουργία / επεξεργασία / διαγραφή / αναζήτηση εργαζομένου

**ΕΝΤΥΠΟ ΠΡΟΔΙΑΓΡΑΦΩΝ ΑΠΑΙΤΗΣΕΩΝ ΛΟΓΙΣΜΙΚΟΥ**

Το παρόν κομμάτι αποτελεί τον προσδιορισμό των απαιτήσων για τη δημιουργία του έργου της διαχείρισης του προσωπικού της εταιρείας. Η εφαρμογή έχει σκοπό μέσα απο μια ιδιαίτερα απλή και εύχρηστη διεπαφή, να παρέχει στον χρήστη ένα περιβάλλον στο οποίο θα μπορεί να διαχειρίζεται τους εργαζόμενους μιας εταιρείας.

Το λογισμικό αυτό, απευθύνεται στο τμήμα προσωπικού της εταιρείας και στους προιστάμενους της εταιρείας καθώς εκείνοι θα είναι οι μόνοι αρμόδιοι που θα μπορούν να λειτουργήσουν το λογισμικό.

Το κομματί αυτό ακολουθεί το πρότυπο του IEEE για Έγγραφα Προδιαγραφής Απαιτήσεων με κάποιες μικρές τροποποιήσεις που έγιναν για να εξυπηρετήσουν την πληρέστερη περιγραφή του συγκεκριμένου λογισμικού.

* ΕΚΤΑΣΗ ΤΟΥ ΚΟΜΜΑΤΙΟΥ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ :

Όλες οι λειτουργίες του προγράμματος έχουν ελεγθεί και κατανοηθεί πλήρως πριν οποιαδήποτε απόπειρα συγγραφής.

Το έγγραφο αυτό για να παραμείνει χρήσιμο, πρέπει να ενημερώνεται και να προσαρμόζεται στις απαιτήσεις και τις τυχόν πρόσθετες λειτουργίες που μπορεί να προκύψουν σε κάποια μελλοντική έκδοση του προγράμματος.

* Απευθυνόμενο κοινό και εισηγησείς για μελέτη του εγγράφου :

Το έγγραφο απευθύνεται κυρίως σε:

* *Τεχνολόγους Λογισμικού(software engineers)*: Το παρον έγγραφο αποτελεί απαραίτητο εργαλείο για την κατανοήση, σχεδίαση και υλοποίηση των απαιτήσεων του προγράμματος. Πρέπει να παρέχουν διορθώσεις / βελτιώσεις και να προσθαφαιρούν χαρακτηριστικά σε αυτό.
* *Προγραμματιστές (developers)*: Βάσει αυτού του εγγράφου πρέπει να πιστοποιείται η ποιότητα των λειτουργιών που προσφέρει το πρόγραμμα.
* *Δοκιμαστές (testers)*: Πρέπει να πιστοποιούν ότι οι δοκιμές που εκτελούν επικυρώνουν τις απαιτήσεις.
* *Απλους χρήστες*: Μπορεί να χρησιμοποιήθει σαν ένας Οδηγός Χρήσης (manual) όπου περιγράφονται αναλυτικά όλες οι λειτουργίες του προγράμματος.

# ΣΚΟΠΟΣ ΤΟΥ ΣΥΓΚΕΚΡΙΜΕΝΟΥ ΛΟΓΙΣΜΙΚΟΥ

Όπως ήδη έχουμε αναφερθεί, σκοπός του συγκεκριμένου έργου είναι η διαχείριση του προσωπικού εταιρείας. Το ακροατήριο θα είναι ο ceo της εταιρείας και η διαχείριση του προσωπικού. Μέσω του συγκεκριμένου λογισμικού θα είναι πολύ πιο εύκολη η διαχείριση των εργαζομένων της εταιρείας και αυτό θα συμβαίνει γιατί θα μειώνεται πολύ ο χρόνος που ίσως χρειαζόταν για να γίνει για παράδειγμα μια καταχώρηση ενος καινούργιο εργαζομενου σε ενα πρόγραμμα όπως το excel. Οι χρήστες που θα διαχειρίζονται το συγκεκριμένο λογισμικό θα είναι οι προιστάμενοι καθώς και το τμήμα προσωπικού.

**ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ**

# Προοπτική του Προϊόντος

Οι developers του προγράμματος είναι ιδιαίτερα ενεργοί ενημερώνοντας συνεχώς το κοινό για οποιαδήποτε αλλαγή ή νέα κυκλοφορία του λογισμικού και προτρύνουν ιδιαίτερα τους χρήστες να δοκιμάζουν τις BETA εκδόσεις ώστε να υπάρχει έντονο feedback που θα βοηθήσει στη βελτίωση του.

# Χαρακτηριστικά του προϊόντος

1. Διαχείριση προσωπικού εταιρείας
2. Καταχώρηση καινούργιων εργαζόμενων
3. Σχόλια/αξιολόγηση προσωπικού
4. Επεξεργασία προσωπικού
5. Διαγραφή προσωπικού απο το λογισμικό
6. Διαγραφή αξιολόγησης (μόνο απο προιστάμενους
7. Αναζήτηση προσωπικού της εταιρείας

# Κατηγορίες και χαρακτηριστικά χρηστών

Οι χρήστες που θα έχουν την αρμοδιότητα να χρησιμοποιήσουν το συγκεκριμένο λογισμικό θα είναι δύο, το ΗR και οι προιστάμενοι.

Οι λειτουργίες που θα μπορούν να κάνουν οι προιστάμενοι είναι οι εξής :

1. Δημιουργία / επεξεργασία / διαγραφή αξιολόγησης

Οι λειτουργίες που θα μπορεί να κάνει το τμήμα προσωπικού ( HR ) είναι:

1. Δημιουργία / επεξεργασία / διαγραφή τμήματος
2. Δημιουργία / επεξεργασία / διαγραφή / αναζήτηση εργαζομένου

# Λειτουργικό περιβάλλον

Το λειτουργικό περιβάλλον είναι windows 7.

# Εξωτερικές απαιτήσεις διεπαφών

Για όλες τις εξωτερικές απαιτήσεις διεπαφών θα αναφερθούμε σε επόμενο κεφάλαιο (πηγαίος κώδικας και GUI) καθώς θα είναι πολύ πιο εύκολη η εξήγηση των χαρακτηριστικών του λογισμικού έχοντας και τα screenshots απο το λογισμίκο και το γραφικό του περιβάλλον.

# Μη Λειτουργίκες Απαιτήσεις

Ταχύτητα συναλλαγών < 10 δευτερολέπτων

# Ασφάλεια πληροφορίας:

1. Κρυπτογράφηση SSL.
2. Data Protection Act 1992.

**USER ACCEPTANCE TEST**

|  |  |  |
| --- | --- | --- |
| **Role** | **Department** | **Name** |
| Developer |  | ΝΑΣΤΟΣ ΝΙΚΟΣ |
| Developer |  | ΖΑΝΕΤΗΣ ΓΕΩΡΓΙΟΣ |
| Developer |  | ΚΑΡΑΓΙΑΝΝΗΣ ΒΙΚΕΝΤΙΟΣ |
| Tester |  | ΝΑΣΤΟΣ ΝΙΚΟΣ |

# Version control:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Section** | **Amendment** |
| 28/12/2014 |  | ΝΑΣΤΟΣ ΝΙΚΟΣ |  |  |
|  |  |  |  |  |

## User Acceptance Definition

User Acceptance Testing should ensure that the application performs at an acceptable level for the Customer.

## UAT Responsibilities

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Responsibilities** |
| Tester | ΝΑΣΤΟΣ ΝΙΚΟΣ |  |
| Δοκιμή της τελικής έκδοσης του προγράμματος |

**ΣΥΓΚΕΝΤΡΩΤΙΚΟ ΣΚΕΛΟΣ ΔΗΜΙΟΥΡΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ**

Το ακροατήριο θα είναι :

* CEO εταιρίας
* Διαχείριση προσωπικου

Στόχος

Νέο λογισμικό

Κατηγορίες Χρηστών

* HR
* Προϊστάμενοι

Περιβάλλον

* Windows 7

Περιορισμοί

* LAN 1GBps

Διεπαφή χρήστη

* Καρτέλες

Επιδόσεις

* Συναλλαγή < 10 sec

Ασφάλεια πληροφορίας

- Κρυπτογράφηση SSL

**ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΚΑΙ GUI**

Ο κώδικας για το συγκεκριμένο λογισμικό ενσωματώνει το μοντέλο

Model-view-control μέσω του οποίου λειτουργούν τρείς βασικές κλάσης, η model class, η view class και η control class.

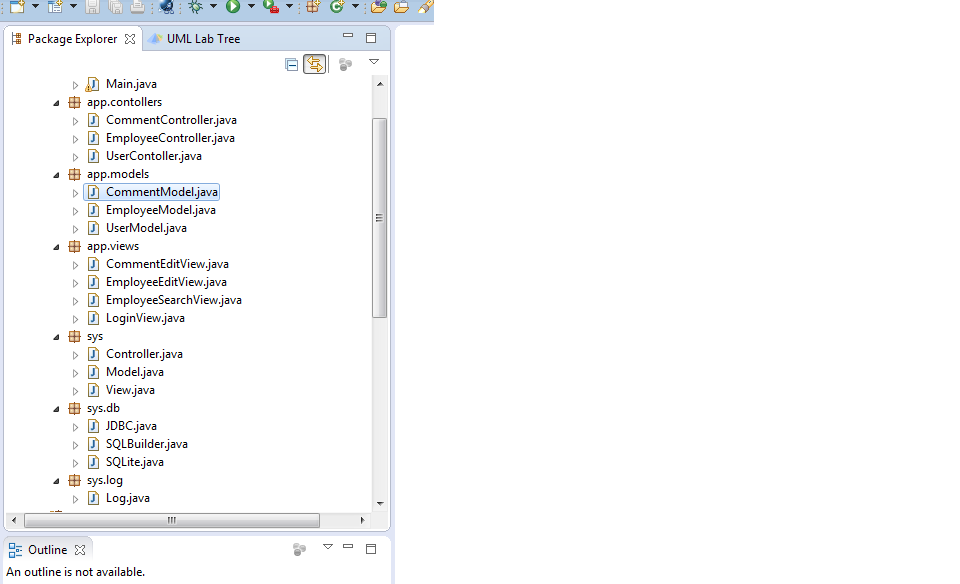
Μέσω της class controller μπορούμε να ενώσουμε και να και να κάνουμε testing στις άλλες δύο classes. Η view class είναι αυτή που ευθύνεται ουσιαστικά για το GUI του λογισμικού μας.

Για την δημιουργία της βάσης δεδομένων μας καθώς και την επεξεργασία των στοιχείων ευθύνονται οι classes :

* JDBC
* SQLBuilder
* SQLite

Μέσω της class UserModel μπορούμε να διαχωρίσουμε το login των προιστάμενων και του τμήματος προσωπικού.

Όλες οι classes φαίνονται απο την παρακάτω εικόνα :



Στις παρακάτω εικόνες φαίνεται ο τρόπος λειτουργίας του λογισμικού :

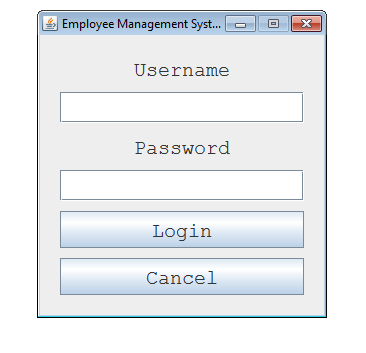
Αρχικά σαν πρώτη εικόνα έχουμε το login μας, το οποίο έχει διαφορετικό username και password το οποίο εξαρτάται απο το αν αυτός που θα συνδεθεί είναι προιστάμενος ή απο το τμήμα προσώπικου.

Αν είναι **προιστάμενος** τότε για username και password έχει :

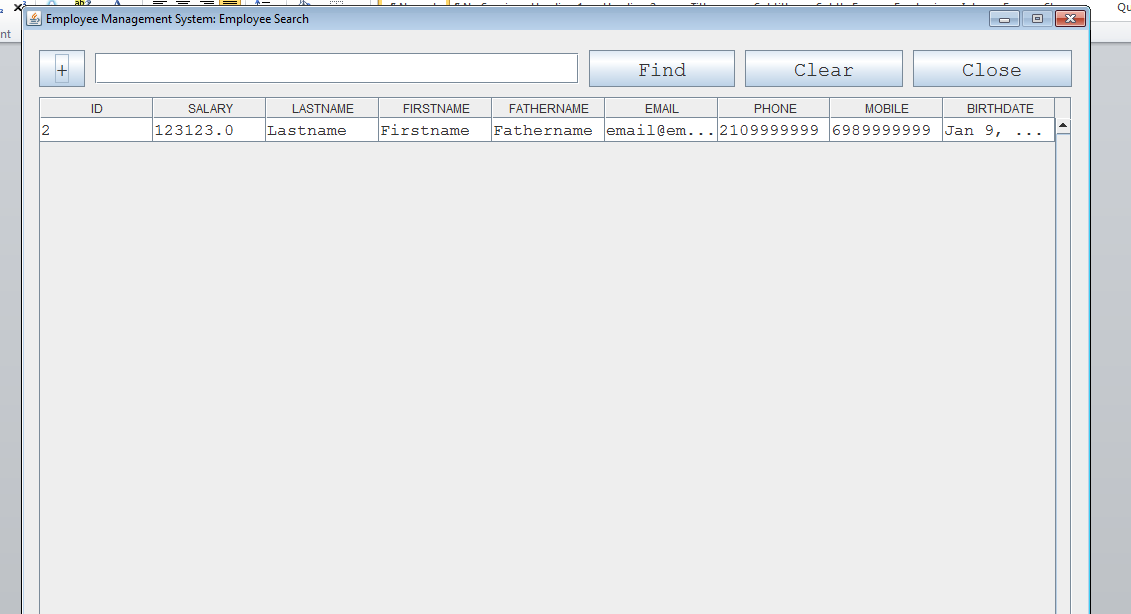
* Username : foreman
* Password : foreman

Αν είναι κάποιος αρμόδιος απο το **τμήμα προσωπικού** τότε :

* Username : hr
* Password : hr

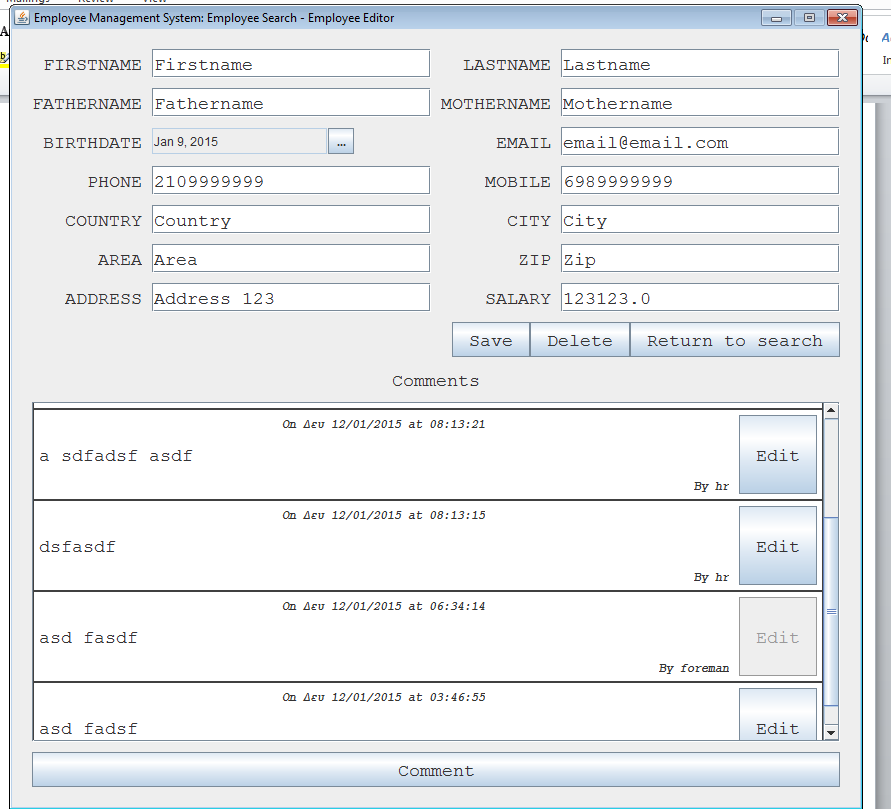


Αφού λοιπόν κάνουμε login είτε ως HR είτε ως κάποιος προιστάμενος τότε βλέπουμε την παρακάτω εικόνα :



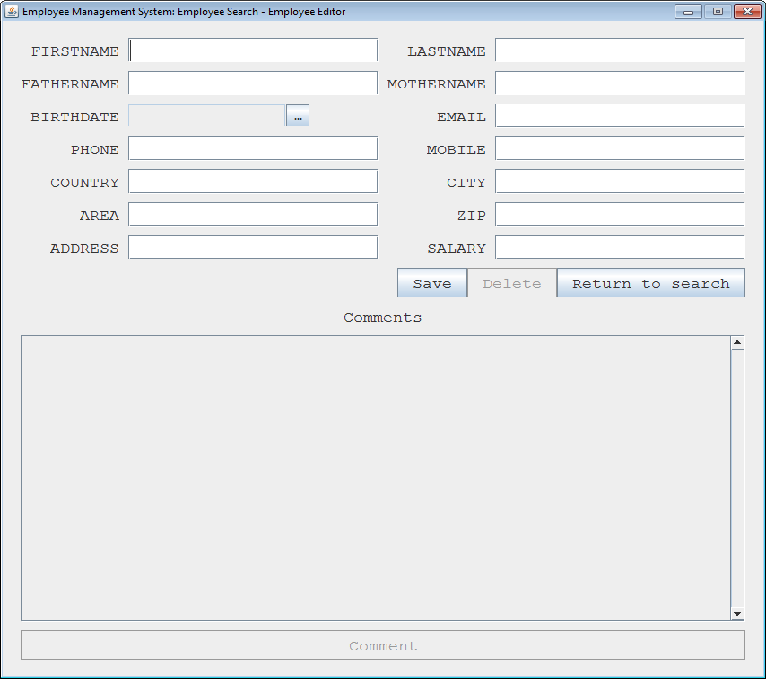
Με βάση αυτή την εικόνα βλέπουμε πως υπάρχει μια διεπάφη καρτέλας όπου στο κέντρο βλέπουμε διάφορες πληροφορίες σχετικά με το ποιός εργαζόμενος έχει καταχωρηθεί και αρκετές πληροφορίες γι’αυτόν.

Κάνοντας κλικ επάνω στον συγκεκριμένο εργαζόμενο, παίρνουμε την παρακάτω εικονά



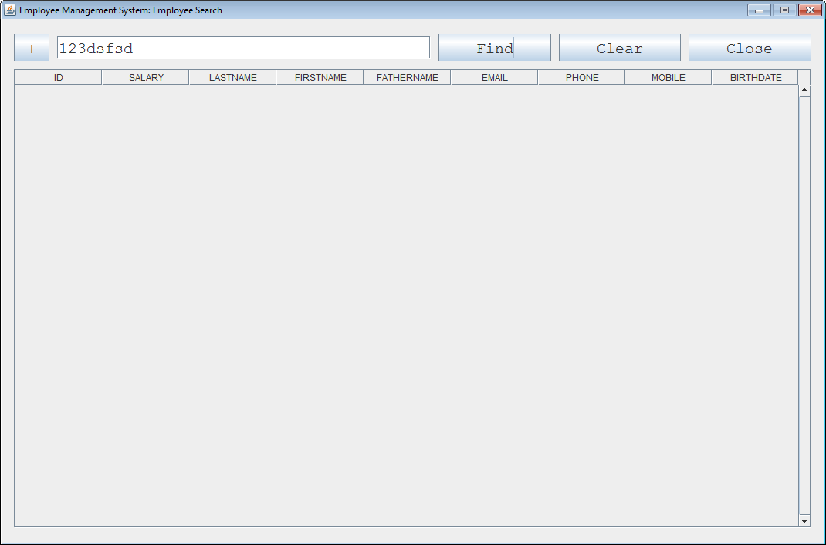
Οπότε βλέπουμε όχι μόνο τα στοιχεία του εργαζόμενου αλλά και την λειτουργία να μπορούμε να εισάγουμε σχόλια/αξιολογήσεις απο το hr ή τους προιστάμενους. Επίσης βλέπουμε πως υπάρχει το κουμπί save για αποθήκευση, το κουμπί delete για διαγραφή και το κουμπί return to search ωστε να μπορούμε να γυρίσουμε ξανά στην προηγούμενη καρτέλα.

Βασική λειτουργία του προγράμματος είναι επίσης και το κουμπί ‘+’. Αυτό το καταλαβαίνουμε διότι με το συγκεκριμένο κουμπί μπορεί να εισάγει είτε προιστάμενος είτε hr, εναν εργαζόμενο. Η εικόνα που μας δείχνει την καρτέλα αφού πατήσουμε το συγκεκριμένο κουμπί είναι η εξής :



Μια επίσης πάρα πολυ σημαντική λειτουργία είναι και το κουμπί ‘find’ μέσω του οποίου μπορούμε να κάνουμε αναζήτηση στα πεδία που έχουμε δώσει για τον κάθε εργαζόμενο και να μας επιστρέψει κάποιο αποτέλεσμα ή να μην μας επιστρέψει τίποτα.

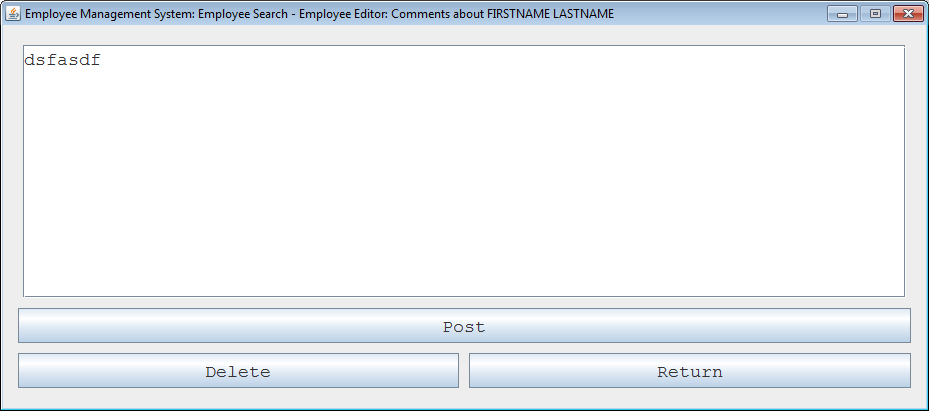
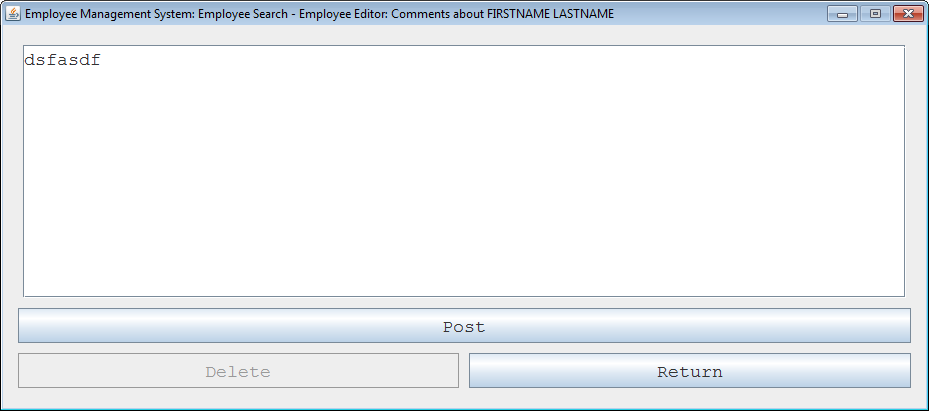
Παρακάτω φαίνεται μια αναζήτηση σε κάποιον εργαζόμενο που δεν υπάρχει :



Στην παραπάνω εικόνα δεν μας έχει εμφανίσει τίποτα γιαί δεν υπάρχει κανένα πεδίο σε κανέναν εργαζόμενο που να εχει το ‘123dsfsd’.

ΜΙΑ ΣΗΜΑΝΤΙΚΗ ΔΙΑΦΟΡΑ ΜΕΤΑΞΥ ΗR ΚΑΙ ΠΡΟΙΣΤΑΜΕΝΩΝ

Ένας προιστάμενος έχει την παραπάνω λειτουργία να μπορεί να διαγράφει και μια αξιολόγηση που μπορεί να έχει γίνει σε κάποιον εργαζόμενο σε αντίθεση με το τμήμα που δεν έχει αυτή τη δυνατότητα. Αυτό το βλέπουμε απο τις δυο εικόνες παρακάτω, η πρώτη είναι για τους προιστάμενους ενώ η δεύτερη είναι για το τμήμα προσωπικού :

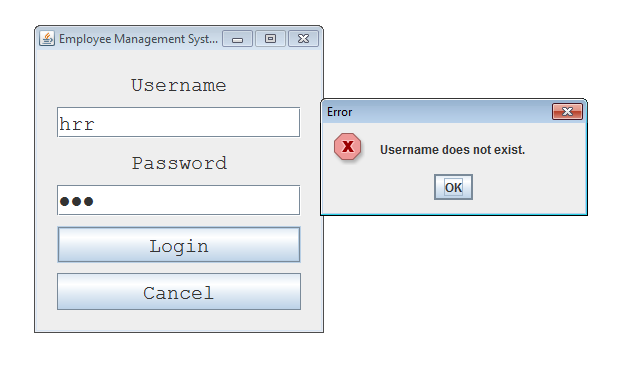
1. 
2. 

**DEBUGGING / TESTING**

Λίγα λόγια για το debugging / testing, αρχίκα μέσω του debugging μπορεί ο developer να ελένχει τον κώδικα που έχει γράψει πολύ πιο εύκολα, γρήγορα και αποδοτίκα. Στο συγκεκριμενο λογισμικό έχουν γίνει δύο κατηγορίες debugging.

Ο πρώτος τρόπος για να κάνουμε debugging και ο πιο απλός είναι να βάζουμε συνέχεια μεθόδους ώστε να μας εμφανίζουν ενα μύνημα είτε αν αυτό το μήνυμα είναι για επιτυχία της εκτέλεσης του κώδικα είτε για το αν έχει υπάρξει κάποιο συγκεκριμένο πρόβλημα και μέσω του μηνύματος να καταλάβουμε που μπορεί να φταίει ο κώδικας.

Για τον συγκεκριμένο τρόπο debugging / testing φτιάξαμε ξεχωριστή class, την log class, μέσω της οποίας εκτελούμε κάθε φορά τα μηνύματα μας. Για παράδειγμα αν εμείς δώσουμε ενα λάθος username ή password τοτε θα μας εκτυπωθεί το ακόλουθο μήνυμα:



Οπότε απευθείας μπορεί να καταλάβει ο developer πως έχει γίνει κάποιο λάθος στην διαδικασία εισαγωγής username και password.

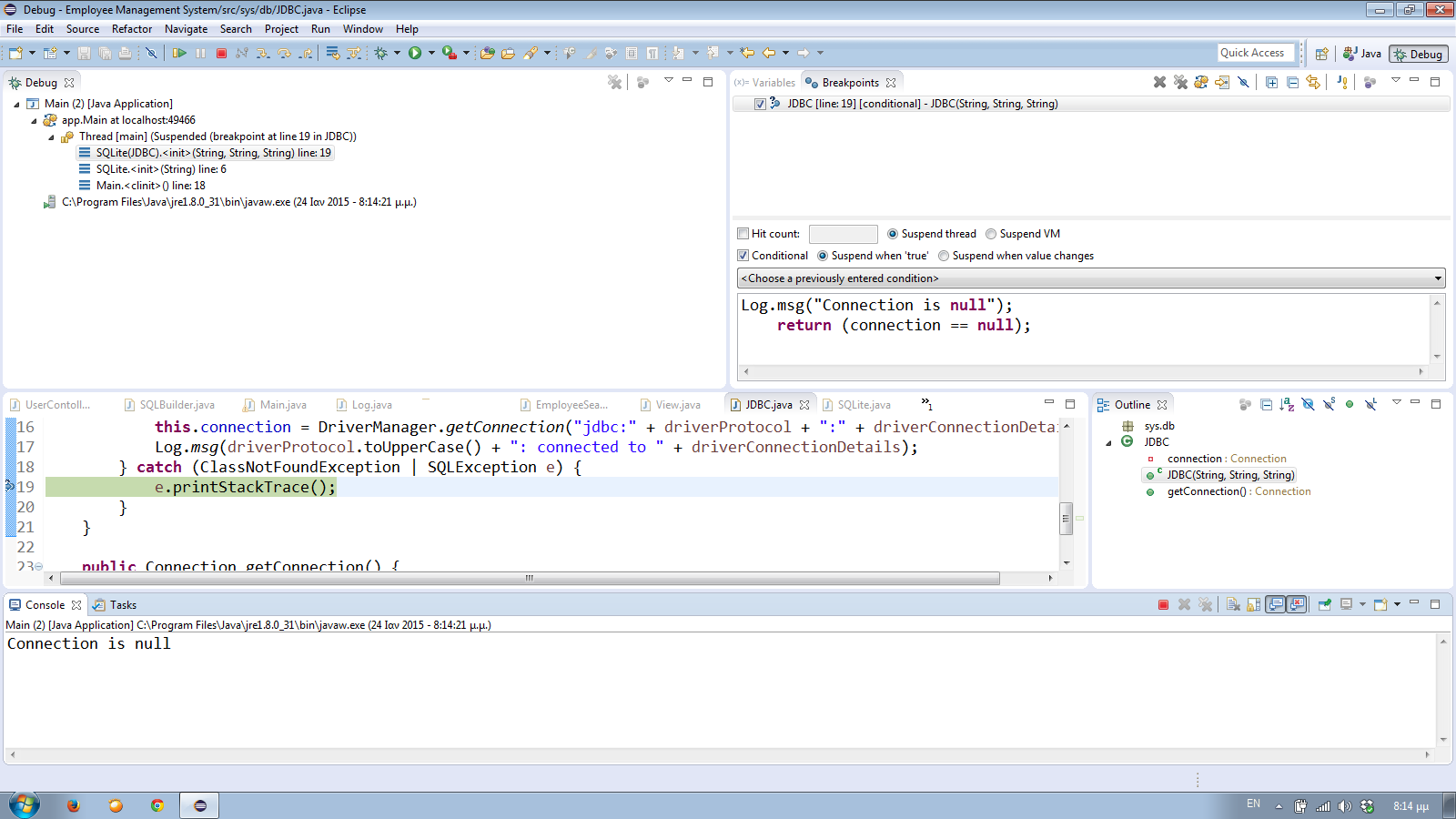
Μπορούμε επίσης να δούμε τη χρησιμότητα που προσφέρουν τα μηνύματα στο θέμα του debugging/testing και μέσα απο τον κώδικα. Βασικό στοιχείο είναι να υπάρχει μήνυμα ότι κάτι δεν πήγε καλά σε οτιδήποτε σημαντικό πάει να γίνει, ενα παράδειγμα είναι και το παρακάτω όπου :



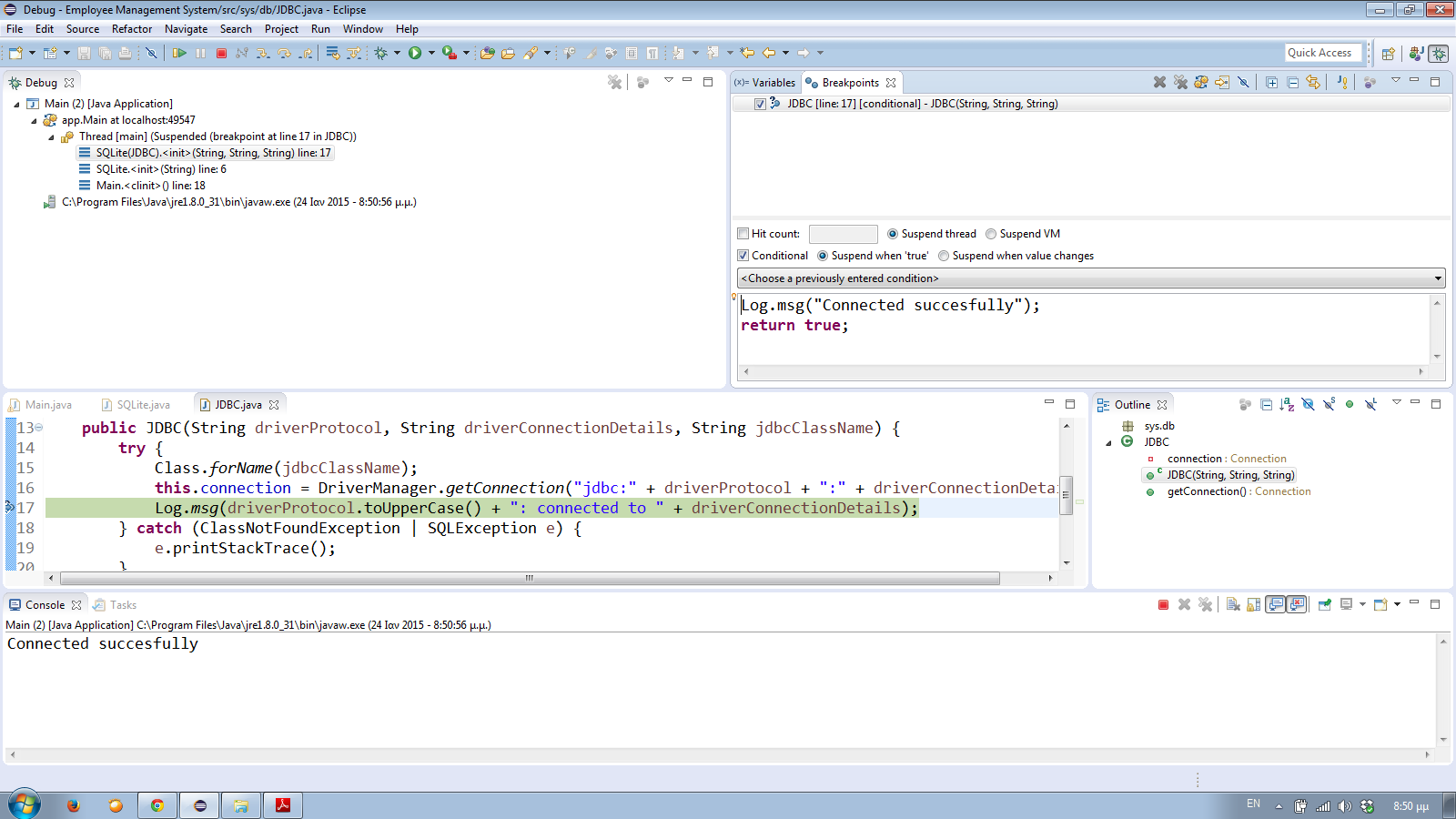
Με βάση την εικόνα βλέπουμε πως υπάρχουν αρκετές μέθοδοι όπου κάθε φορά που δεν έχει γίνει κάτι καλά να εμφανίζεται μήνυμα κάτι που βοηθάει πάρα πολύ τους developers γιατί μπορούν πολύ εύκολα και γρήγορα να καταλάβουν που έχει γίνει το λάθος και να ανατρέξουν στον κώδικα και να το διορθώσουν. Επομένως γενικά καταλαβαίνουμε οτι σε ότι έχει να κάνει με debugging / testing τα messages που μπορούμε να βάζουμε για το αν κάτι έχει πάει στραβα μας βοηθάνε πάρα πολύ.

Ένας δεύτερος τρόπος debugging/testing είναι να χρησιμοποιήσουμε breakpoints στον κώδικα μας, σε σημεία που εμείς θέλουμε και να κοιτάξουμε αν έχει πάει καλα το συγκεκριμένο τμήμα ή αλλιώς να εμφανίζεται ότι έχει γίνει κάποιο λάθος. Στο συγκεκριμένο λογισμικό ενδεικτικά παραθέτουμε δύο παραδείγματα στα οποία έχει γίνει debugging με breakpoints μέσα στα οποία έχει γραφτεί κώδικας :

* Στο πρώτο παράδειγμα debugging που θα δείξουμε έχει γίνει εσκεμένα ένα λάθος στο γράψιμο κώδικα για την σύνδεση μας με την βάση δεδομένων. Βάζοντας λοιπόν ένα breakpoint μέσα στο οποίο έχει γραφτεί κώδικας, βρίσκουμε ότι το connection με την βάση μας είναι null οπότε απευθείας μπορούμε να κοιτάξουμε το συγκεκριμένο κομμάτι κώδικα και να το διορθώσουμε πολύ γρήγορα χώρις να χρειαστεί να ψάχνουμε 10000 γραμμές κώδικα.



* Στο δεύτερο παράδειγμα που θα δείξουμε συμβαίνει το ακριβώς αντίθετο, χρησιμοποιούμε το breakpoint μας ώστε να μας δείξει ότι το connection με την βάση μας έγινε σωστά. Αυτό μπορεί είναι πολύ χρήσιμο σε σημεία κώδικα τα οποία είναι λεπτεπίλεπτα και θέλουμε να λάβουμε μήνυμα πως έγιναν όλα σωστά αλλιώς να σταματήσει η εκτέλεση του προγράμματος.

****

**ΔΙΑΓΡΑΜΜΑΤΑ UML**

Λίγα λόγια για τα διαγράμματα γενικά :

ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

Τα διαγράμματα κλάσεων μας επιτρέπουν:

* + Να ορίσουμε κλάσεις
  + Να ορίσουμε ιδιότητες, μεθόδους και καθήκοντα των κλάσεων
  + Να ορίσουμε περιορισμούς στις ιδιότητες στις μεθόδους και στα καθήκοντα των κλάσεων

ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ

Τα διαγράμματα αυτά αποτελούνται απο :

* Στιγμιότυπο
* Ενα σημείο ενεργοποίησης
* Αποστολή μηνύματος
* Λήψη απάντησης

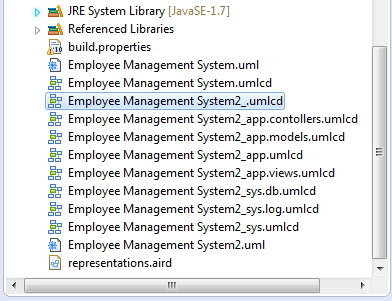
ΔΙΑΓΡΑΜΜΑΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ

Τα διαγράμματα δραστηριότητας αποτελούνται απο :

* Ενα σημείο έναρξης
* Ενα βελάκι το οποίο είναι η μετάβαση
* Αμέσως μετά έχουμε την δραστηριότητα μας
* Τέλος έχουμε το κυκλικό σύμβολο που δηλώνει την λήξη της δραστηριότητας.

Στο συγκεκριμένο λογισμικό, όλα τα UML διαγράμματα περιέχονται μέσα στον κώδικα μας και μπορούμε να τα ανοίξουμε και να τα δούμε πολύ εύκολα με ένα πρόγραμμα όπως για παράδειγμα το eclipse μέσω του οποίου «τρέχουμε» και τον κώδικα μας.

Στην παρακάτω εικόνα βλέπουμε που μπορούμε να πάμε ώστε να βρούμε όλα τα uml διαγράμματα που έχουμε δημιουργήσει απο τον πηγαίο κώδικα μας.



ΕΝΔΕΙΚΤΙΚΕΣ ΕΙΚΟΝΕΣ ΑΠΟ UML ΔΙΑΓΡΑΜΜΑΤΑ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ ΜΑΣ

