



**Πανεπιστήμιο Θεσσαλίας**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

## **Τίτλος**

**«Πολυπρακτορικά συστήματα σε web 2.0 εφαρμογές»**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

της

**ΝΟΥΤΣΗ ΕΥΓΕΝΙΑΣ**

**Επιβλέποντες : Δασκαλοπούλου Ασπασία**  
Επίκουρη Καθηγήτρια Π.Θ.

**Ακρίτας Αλκιβιάδης**  
Αναπληρωτής Καθηγητής Π.Θ.

Βόλος, Φεβρουάριος 2012

## Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω την κα Δασκαλοπούλου για τη στήριξη και την καθοδήγησή της καθώς και για τις πολύτιμες συμβουλές της.  
Επιπλέον, θα ήθελα να ευχαριστήσω το Ν.Φραγκογιάννη για την τεχνική υποστήριξη που μου παρείχε.  
Οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου για την αγάπη τους, τη στήριξή τους και την πίστη τους σε μένα όλα αυτά τα χρόνια.  
Ασφαλώς, χωρίς τους φίλους μου δε θα έκανα τίποτα από όλα αυτά και ακόμη και αν τα έκανα δεν θα είχαν κανένα απολύτως νόημα. Είναι εκεί πάντα για να μου θυμίζουν τι έχει πραγματική αξία και τους ευχαριστώ γι' αυτό.

Βόλος, Φεβρουάριος 2012

## Περίληψη

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η υλοποίηση και ανάπτυξη μιας εφαρμογής για παροχή/εύρεση θέσεων εργασίας. Με την εφαρμογή αυτή καθίσταται εφικτή η αλληλεπίδραση εταιρειών που προσφέρουν θέσεις εργασίας και χρηστών που αναζητούν μια θέση εργασίας.

Η εφαρμογή **Earn Your Living** υλοποιήθηκε κυρίως σε δύο στάδια. Στο πρώτο στάδιο, αναπτύσσεται μία διαδικτυακή (web-based) διεπαφή προορισμένη για τους χρήστες. Το δεύτερο στάδιο αποτελείται από έναν αλγόριθμο λήψης απόφασης, εφαρμοσμένο σε ένα κατανεμημένο (distributed), πολυπρακτορικό (multiagent) περιβάλλον με τη βοήθεια της πλατφόρμας εκτέλεσης JADE.

Πιο συγκεκριμένα, στην εφαρμογή δίνεται έμφαση στη διαδικασία ανάπτυξης διαδικτυακών εφαρμογών WEB 2.0, με χρήση διαδικτυακών γλωσσών προγραμματισμού. Επιπλέον, ιδιαίτερη έμφαση δίνεται στην αποκεντρωμένη αρχιτεκτονική και στις δυνατότητες διασύνδεσης που παρέχει η Jade για ανάπτυξη εφαρμογών με πράκτορες, όπως επίσης και στη δυνατότητα των πρακτόρων να αλληλεπιδρούν και να επικοινωνούν τόσο μεταξύ τους, όσο και με βάσεις δεδομένων.

Αυτή η εφαρμογή μπορεί να χρησιμεύει ως οδηγός για ανάπτυξη παρόμοιων εφαρμογών σε διαφορετικά πεδία γνώσης, καθώς καθιστά σαφή την χρησιμότητα της αλληλεπίδρασης πρακτόρων σε εφαρμογές διαδικτυακού τύπου.

**Λέξεις Κλειδιά:** <<web-based, multiagent environment, JADE, WEB 2.0>>

## Abstract

The scope of this thesis is the development and implementation of a “smart” Job-portal application. Through this application companies offering a job are able to interact with customers that are currently looking for a job.

Our application, **Earn your Living** , was mainly implemented in two parts. During the first part, a web-based application is developed, in order to provide a user interface. The second part consists of a decision making algorithm applied through a distributed, multiagent environment, using the JADE communication platform.

Specifically, great emphasis was given in the procedure of creating WEB 2.0 applications. In addition to this, the decentralized architecture so as the capability of interconnection that JADE provides is of significant importance. Moreover, of equivalent importance is the capability of the agents to communicate not only with each other but also with databases.

This application could be considered as a guide to the implementation of similar applications, adjusted to different fields of knowledge. The mentioned is possible since the usefulness of agents’ interoperability into web applications becomes clear.

**Keywords:** <<web-based application, JADE, multiagent, interoperability>>



## Πίνακας περιεχομένων

<b>1</b>	<b>Εισαγωγή.....</b>	<b>1</b>
1.1	Εργασία και διαδίκτυο .....	1
1.2	Αντικείμενο διπλωματικής.....	2
1.2.1	Συνεισφορά .....	2
1.3	Οργάνωση κειμένου.....	3
<b>2</b>	<b>Σχετικές εργασίες.....</b>	<b>4</b>
2.1	<a href="http://www.job-portal.gr">www.job-portal.gr</a> .....	5
2.2	<a href="http://jobs.goabroad.com">http://jobs.goabroad.com</a> .....	5
<b>3</b>	<b>Θεωρητικό υπόβαθρο .....</b>	<b>7</b>
3.1	Η έννοια του πράκτορα λογισμικού.....	8
3.2	Η γλώσσα ACL.....	8
3.3	Η συμπεριφορά ενός πράκτορα και η κλάση behavior .....	9
3.4	Η κλάση agent.....	11
<b>4</b>	<b>Ανάλυση απαιτήσεων συστήματος.....</b>	<b>14</b>
4.1	Αρχιτεκτονική.....	14
4.1.1	Μοντέλο Πελάτη-Εξυπηρετητή( <i>Client-Server Model</i> ).....	14
4.1.2	Αρχιτεκτονική <i>JADE agents</i> .....	16
4.2	Περιγραφή Λειτουργιών .....	17
4.2.1	Διεπαφή χρήστη( <i>user interface</i> ).....	18
4.2.2	Υποσύστημα Λήψης Απόφασης.....	27
4.3	Μοντέλο Οντοτήτων Συσχετίσεων .....	29
4.4	Τεχνολογίες που χρησιμοποιήθηκαν.....	30
4.4.1	<i>HTML</i> .....	30
4.4.2	<i>CSS</i> .....	30
4.4.3	<i>Java Server Pages (JSP)</i> .....	30
4.4.4	<i>JAVASCRIPT</i> .....	31
4.4.5	<i>MySQL</i> .....	31
4.4.6	<i>AJAX</i> .....	31

4.4.7	<i>JQuery</i> .....	31
<b>5</b>	<b>Σχεδίαση Συστήματος</b> .....	<b>32</b>
5.1	Αρχιτεκτονική.....	32
5.2	Περιγραφή αρχείων διαδικτυακής εφαρμογής.....	34
5.2.1	<i>Styling.css</i> .....	34
5.2.2	<i>Code.html</i> .....	34
5.2.3	<i>Contact.jsp</i> .....	34
5.2.4	<i>form_company.jsp / form_employee.jsp</i> .....	34
5.2.5	<i>entryc.jsp / entry.jsp</i> .....	35
5.2.6	<i>ok.html/ok1.html</i> .....	35
5.2.7	<i>checkResults.jsp</i> .....	35
5.2.8	<i>check.jsp</i> .....	35
5.3	Περιγραφή κλάσεων συστήματος απόφασης.....	35
5.3.1	<i>Η κλάση employeeAgent</i> .....	36
5.3.2	<i>Η κλάση companyAgent</i> .....	38
5.4	Βάση Δεδομένων .....	40
<b>6</b>	<b>Υλοποίηση</b> .....	<b>42</b>
6.1	Λεπτομέρειες υλοποίησης.....	42
6.1.1	<i>Αλγόριθμος ταιριάσματος</i> .....	42
6.1.2	<i>Μηνύματα μεταξύ userAgent-companyAgent</i> .....	44
6.2	Πλατφόρμες και προγραμματιστικά εργαλεία .....	46
6.2.1	<i>Ενσωματωμένο περιβάλλον ανάπτυξης(IDE) Netbeans</i> .....	46
6.2.2	<i>Πλατφόρμα JADE</i> .....	46
6.2.3	<i>Wamp Server</i> .....	50
6.2.4	<i>Απαιτήσεις για εγκατάσταση διπλωματικής</i> .....	51
<b>7</b>	<b>Επίλογος</b> .....	<b>53</b>
7.1	Σύνοψη και συμπεράσματα.....	53
7.2	Μελλοντικές επεκτάσεις .....	53
<b>8</b>	<b>Βιβλιογραφία</b> .....	<b>55</b>

# 1

## *Εισαγωγή*

### *1.1 Εργασία και διαδίκτυο*

Τα τελευταία χρόνια γίνεται πολύ συχνά λόγος για συστήματα λήψης απόφασης(**Decision Support Systems,DSS**).Τα συστήματα αυτά απασχόλησαν στο παρελθόν και απασχολούν ακόμη ερευνητές, λόγω της πληθώρας των εφαρμογών τους σε χρηστικούς τομείς της ανθρώπινης δραστηριότητας.Σήμερα με την ολοένα και μεγαλύτερη εισροή δεδομένων στον παγκόσμιο ιστό και με την εκθετική αύξηση του όγκου πληροφορίας τα συστήματα αυτά αποκτούν ακόμη μεγαλύτερη σημασία.

Ο σκοπός της ανάπτυξης τέτοιων συστημάτων είναι η διαχείριση πληροφορίας από βάσεις δεδομένων με σκοπό τη λήψη της καλύτερης απόφασης βάσει ενός συνόλου κριτηρίων.Μέσα από τις επιλογές του χρήστη το σύστημα αποκτά γνώση για τα χαρακτηριστικά/απαιτούμενά του και ακολούθως οδηγείται στη λήψη απόφασης μέσω κάποιας αλγοριθμικής διαδικασίας.

Η εργασία αυτή έχει ως σκοπό τη δημιουργία ενός τέτοιου συστήματος για την επίλυση του προβλήματος της εύρεσης εργασίας.Πιο συγκεκριμένα,λόγω των αυξανόμενων ποσοστών των αιτούντων εργασίας, αλλά και των αυξανόμενων απαιτήσεων της παγκόσμιας παραγωγής κεφαλαίου,δηλαδή των εταιρειών σε παγκόσμιο επίπεδο, οδηγούμαστε σε μια πραγματιστική ανάγκη. Η ανάγκη αυτή αφορά στη μοντελοποίηση της δοθείσας πληροφορίας αλλά και σε μια αυτοματοποιημένη λήψη απόφασης έτσι ώστε να εξυπηρετηθούν άμεσα οι



απαιτήσεις και των δύο πλευρών(εταιρείες-άνεργοι). Έγινε προσπάθεια η απόφαση αυτή να λαμβάνει υπόψη τη γενική επαγγελματική εικόνα του αιτούντος εργασία,να συνυπολογίζει δηλαδή όλα τα προσόντα που αυτός απέκτησε στη μέχρι τώρα πορεία του.

## **1.2 Αντικείμενο διπλωματικής**

Στα πλαίσια της εργασίας αυτής αναπτύχθηκε ένα σύστημα λήψης απόφασης για θέσεις εργασίας.Ο τομέας αυτός πέρα από ότι αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας κάθε ανθρώπου, ιδιαίτερα αυτή τη χρονική στιγμή κρίνεται απαραίτητη η όποια διευκόλυνση μπορεί να παρέχεται.

Έτσι λοιπόν, αναπτύχθηκε ένας δικτυακός τόπος μέσα από τον οποίο αιτούντες εργασία στον τομέα των υπολογιστών και εταιρείες παρέχουν τα στοιχεία τους και αναμένουν τα αποτελέσματα της αίτησής τους.. Τα αποτελέσματα αυτά καθορίζονται από ένα back-end τμήμα, στο οποίο πράκτορες αλληλεπιδρούν μεταξύ τους μέσω της πλατφόρμας εκτέλεσης JADE, ανταλλάσσουν δεδομένα και παίρνουν μια "δίκαιη" απόφαση.

Ένας από τους βασικούς διαχωρισμούς τους οποίους κάνει το σύστημα είναι μεταξύ εταιρειών και εργαζομένων.Ο χρήστης μπορεί να συμπληρώσει τα στοιχεία του όντας εργαζόμενος ή εταιρεία.

Η εργασία αυτή επιδιώκει να έχει διπλό ρόλο.Πρώτον να αποτελέσει πεδίο ερευνητικής δουλειάς για τον τρόπο δημιουργίας λειτουργικών δικτυακών τόπων ,βασισμένο σε πολυπρακτορικό περιβάλλον. Κατά δεύτερον,επιδιώκει να προσφέρει μια αποτελεσματική, γρήγορη και αξιοκρατική αναζήτηση εργασίας.

### **1.2.1 Συνεισφορά**

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Μελετήσαμε τις γλώσσες προγραμματισμού για εφαρμογές διαδικτύου.(JAVA,JSP,HTML,CSS,JAVASCRIPT,MySQL,JQuery,AJAX)
2. Υλοποιήσαμε 1 αλγόριθμο ταιριάσματος κριτηρίων.
3. Αξιολογήσαμε την επίδοση των αλγορίθμου και καταλήξαμε στο ότι ο δυναμικός τρόπος επικοινωνίας ήταν αποτελεσματικός παρόλη την αυξανόμενη πολυπλοκότητα.
4. Μελετήσαμε την πλατφόρμα εκτέλεσης JADE(JADE middleware).
5. Μελετήσαμε την γλώσσα ACL,γλώσσα επικοινωνίας των πρακτόρων.

### ***1.3 Οργάνωση κειμένου***

Το κείμενο της διπλωματικής εργασίας παρουσιάζει την ακόλουθη διάρθρωση. Στο 1<sup>ο</sup> κεφάλαιο έχουμε μία εισαγωγή στο θέμα της διπλωματικής εργασίας. Στο κεφάλαιο 2 παρουσιάζονται σχετικές εργασίες και ακολουθεί σύγκριση αυτών με τη δική μας υλοποίηση. Θέματα θεωρητικού υποβάθρου συζητώνται στο κεφάλαιο 3. Στο κεφάλαιο 4 προχωρούμε σε περιγραφή της αρχιτεκτονικής του συστήματος καθώς και των γενικότερων απαιτήσεων και λειτουργιών του συστήματος. Συνεχίζοντας, στο κεφάλαιο 5 συζητώνται θέματα σχεδιασμού συστήματος ενώ στο κεφάλαιο 6 προχωρούμε σε παρουσίαση του τρόπου υλοποίησης της εφαρμογής. Τέλος, ο επίλογος στο κεφάλαιο 7 ολοκληρώνει τη διάρθρωση του κειμένου και τέλος στο κεφάλαιο 8 παρουσιάζεται η βιβλιογραφία που απαιτήθηκε.

# 2

## *Σχετικές εργασίες*

Είναι λογικό το γεγονός ότι τα τελευταία χρόνια υπάρχει πληθώρα αντίστοιχων εγχειρημάτων στο διαδίκτυο. Οι λόγοι ύπαρξης αυτών είναι προφανείς και σχετίζονται με την ανάγκη του κάθε ανθρώπου για εύρεση εργασίας. Επιπλέον, με την ύπαρξη τέτοιων εγχειρημάτων γίνεται πιο ευέλικτη η κίνηση στον επαγγελματικό χώρο και στην επικοινωνία προσφερόμενων και αιτούντων θέσεων εργασίας.

Τα κύρια στοιχεία των ήδη υπάρχοντων δικτυακών τόπων είναι τα εξής:

1. Κριτήρια Αναζήτησης:
  - Αναζήτηση εργασίας βάσει περιοχής.
  - Αναζήτηση εργασίας βάσει αντικειμένου(πεδίο ειδίκευσης).
2. Τρόπος κοινοποίησης θέσεων εργασίας:
  - Κατάλογος με προσφερόμενες θέσεις σε όλους τους τομείς.
  - Κατάλογος με προσφερόμενες θέσεις σε συγκεκριμένους τομείς.
  - Κατάλογος με σελίδες που παραπέμπουν σε γραφεία διασύνδεσης πανεπιστημίων ,διεθνείς οργανισμούς κ.α.
3. Τρόπος επικοινωνίας μεταξύ προσφερόμενων και αιτούντων:
  - Με αποστολή βιογραφικού σημειώματος(ψηφιακού και μη).
  - Με αποστολή e-mail για εκδήλωση ενδιαφέροντος.

## 2.1 [www.job-portal.gr](http://www.job-portal.gr)

Στον αναφερθέντα δικτυακό τόπο παρέχονται υπερσύνδεσμοι σε αγγελίες , εταιρείες ευρέσεως εργασίας ,γραφείων διασύνδεσης,κ.α ,όπως μπορεί να διαπιστώσει ο χρήστης με μια απλή πλοήγηση.

Παρατηρούμε ότι αν και ο απώτερος σκοπός είναι η εύρεση εργασίας,όπως και στη δική μας εφαρμογή,διαφέρουν εξολοκλήρου τα μέσα με τα οποία αυτό γίνεται εφικτό.Πιο συγκεκριμένα ,σε αυτό το παράδειγμα, ο δικτυακός τόπος λειτουργεί ως ένας συγκεντρωτικός κατάλογος υπηρεσιών που παρέχονται κατά μήκος του παγκόσμιου ιστού για εύρεση εργασίας.Συνεπώς ο τρόπος λειτουργίας διαφέρει από αυτό του Earn Your Living ,καθώς στο δεύτερο παρέχεται άμεσα η ευκαιρία για ανάθεση εργασίας βάσει πλήρωσης απαιτούμενων κριτηρίων,τα οποία κάθε εταιρεία καλείται να δώσει.Έτσι,απέχει από τη φιλοσοφία του συγκεκριμένου δικτυακού τόπου.

Σε αυτό το σημείο αξίζει να προστεθεί ότι κατά τη διεξαγωγή της διπλωματικής εργασίας παρατηρήσαμε ότι η ανάθεση εργασίας χωρίζεται σε επιμέρους διαδικασίες όπως :

- Αίτηση,
- Αξιολόγηση,
- Επιλογή Συνεντευξιαζόμενων,
- Συνέντευξη,
- Τελική Επιλογή.

Η εφαρμογή Earn Your Living πέρα από τη διευκόλυνση που παρέχει στην κατάθεση αίτησης ενδιαφέροντος,πράγμα που παρέχουν και όλοι οι άλλοι δικτυακοί τόποι, προχωρά και στο επόμενο στάδιο ,αυτό της αξιολόγησης κριτηρίων και επιλογής. Έτσι, η εφαρμογή αυτή μπορεί να υιοθετηθεί από εταιρείες που αναζητούν εργαζομένους και να προσαρμοστεί στη διαδικασία αξιολόγησης και επιλογής των.

## 2.2 <http://jobs.goabroad.com>

Στο ίδιο πνεύμα με την προηγούμενη εφαρμογή κινείται και το παραπάνω .Εδώ παρατηρούμε ότι προσφέρεται μια πιο εξειδικευμένη αναζήτηση εργασίας, καθώς ο χρήστης

μπορεί να περιορίσει το σύνολο αποτελεσμάτων, επιλέγοντας γνωστικό αντικείμενο και χώρα στην οποία επιθυμεί να δουλέψει.

Επίσης παρατηρούμε την ακόλουθη ομοιότητα με τη δική μας εφαρμογή: η αναζήτηση γίνεται βάσει γνωστικού αντικειμένου όπως και στη δική μας εφαρμογή, που προορίζεται για τον τομέα των Η/Υ. Πρέπει να αναφέρουμε ότι ο λόγος για τον οποίον περιοριστήκαμε σε αυτό τον τομέα είναι το γεγονός ότι η εφαρμογή μπορεί εύκολα να επεκταθεί για άλλους τομείς, εφόσον έχει υλοποιηθεί ήδη για ένα τομέα.

Παρόλα αυτά, όπως και στο προηγούμενο παράδειγμα έτσι και εδώ η εφαρμογή περιορίζεται στο στάδιο της αίτησης και δεν προχωρά στο στάδιο της αξιολόγησης κριτηρίων. Αξίζει επίσης να αναφέρουμε ότι στον αλγόριθμο αξιολόγησης που υλοποιήσαμε πραγματοποιούμε μια γενική αποτίμηση των χαρακτηριστικών του αιτούντος εργασία, λαμβάνοντας υπόψη και στοιχεία που μέχρι τώρα θεωρούνταν δευτερεύοντα στον τομέα εύρεσης εργασίας. Το παραπάνω διαφοροποιείται αρκετά σε σχέση με άλλους τρόπους αξιολόγησης.

# 3

## *Θεωρητικό υπόβαθρο*

Για την παρούσα εργασία, απαιτήθηκαν γνώσεις οι οποίες προετοίμασαν το έδαφος για την υλοποίηση της εφαρμογής. Αρχικά, ήρθαμε σε επαφή με βασικές έννοιες δικτύωσης, το πρωτόκολλο HTTP και αρχές σχεδιασμού εύχρηστων διεπαφών. Επίσης, κρίθηκε σημαντική η μελέτη των επιπέδων μιας διαδικτυακής εφαρμογής καθώς και των τεχνολογιών που σχετίζονται με τα επίπεδα αυτά.

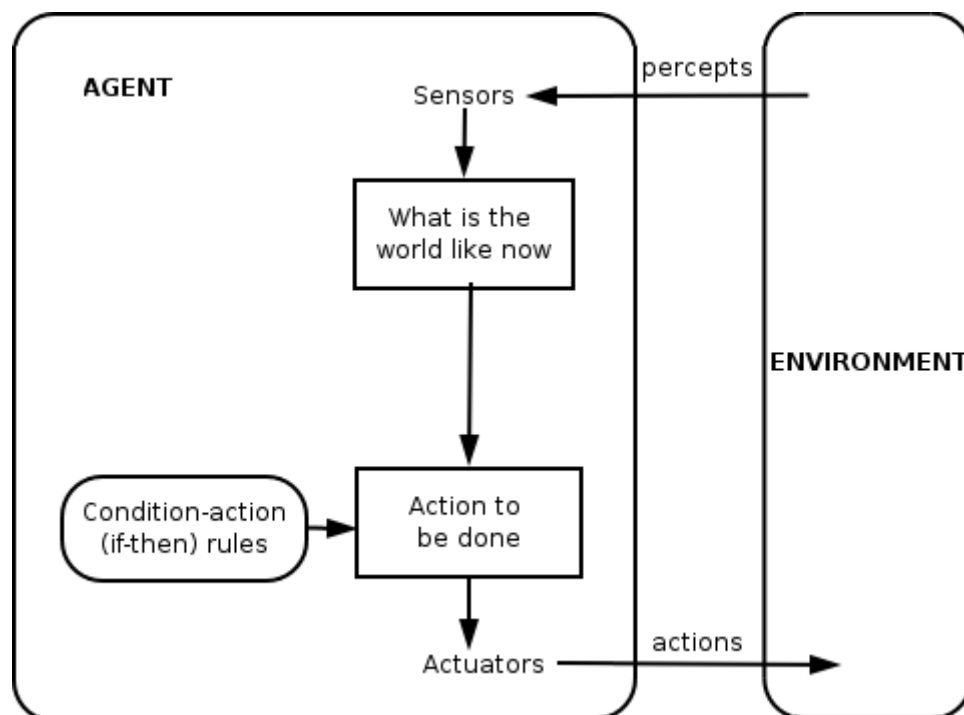
Είναι προφανές, δεδομένης της συγγραφής κώδικα σε JAVA,HTML,CSS,JAVASCRIPT,JQUERY και AJAX ,απαραίτητη ήταν η μελέτη αυτών των γλωσσών.

Επίσης,όσον αφορά στο δεύτερο τμήμα της εργασίας, ανακτήσαμε επαφή με την έννοια του πράκτορα και των πολυπρακτορικών συστημάτων.Επιπλέον,μελετήσαμε τους πράκτορες της πλατφόρμας εκτέλεσης και επικοινωνίας JADE και γνωρίσαμε τη γλώσσα ACL.

Παρακάτω ακολουθεί ανάλυση βασικών εννοιών,απαραίτητων για την κατανόηση της συνολικής εργασίας.

### 3.1 Η έννοια του πράκτορα λογισμικού

Όπως ήδη γνωρίζουμε ο πράκτορας(agent) είναι ένα κομμάτι λογισμικού που αντιλαμβάνεται το περιβάλλον του και δρα για κάποιον χρήστη ή κάποιον άλλο πράκτορα εκ μέρους εκείνου. Αυτή η δράση υποδεικνύει την εξουσιοδότηση για λήψη απόφασης για κάποια επόμενη δράση,σε περίπτωση που προβλέπεται μία.



Εικ.1:Πράκτορας Λογισμικού

Η εφαρμογή μας όπως προαναφέρθηκε χρησιμοποιεί πράκτορες που δημιουργούνται και επικοινωνούν στα πλαίσια της πλατφόρμας εκτέλεσης και επικοινωνίας JADE. Με τη χρήση των JADE πρακτόρων έχουμε στη διάθεσή μας επιπλέον λειτουργίες και εργαλεία που διευκολύνουν τη χρήση τους.Τα βασικότερα αυτών παρουσιάζονται ακολούθως.

### 3.2 Η γλώσσα ACL

Μηνύματα που ανταλλάσσονται μεταξύ JADE πρακτόρων ακολουθούν την τυποποίηση της γλώσσας ACL,που έχει καθοριστεί από τη FIPA(**F**oundation for **I**ntelligent

**Physical Agents**), που καθορίζει διεθνώς τα standard της διαλειτουργικότητας των πρακτόρων. Η τυποποίηση αυτή καθορίζει τα ακόλουθα πεδία:

- Τον αποστολέα του μηνύματος.
- Τη λίστα με τους παραλήπτες.
- Την πρόθεση της επικοινωνίας, που αναφέρεται στο λόγο για τον οποίο επιδιώκεται η επικοινωνία. Οι λόγοι μπορεί να είναι αίτηση (**Request**), αν ο αποστολέας επιθυμεί ο παραλήπτης να εκτελέσει μια λειτουργία, ενημέρωση (**Inform**), αν ο αποστολέας επιθυμεί ο παραλήπτης να είναι ενήμερος από κάποιο γεγονός, ερώτηση (**Query If**) για να επιβεβαιώσει ο αποστολέας αν μια συνθήκη είναι αληθής, κλήση για προτάσεις (**Call For Proposals**), πρόταση (**Proposal**), αποδοχή (**Accept Proposal**) και Απόρριψη (**Reject**).
- Το περιεχόμενο, το οποίο καθορίζεται από την πρόθεση επικοινωνίας.
- Τη γλώσσα περιεχομένου.
- Την οντολογία, που αναφέρεται στην έννοια των συμβόλων που χρησιμοποιούν αποστολείς και παραλήπτες. Εδώ πρέπει παραλήπτες και αποστολείς να αποδίδουν το ίδιο νόημα στα χρησιμοποιούμενα σύμβολα.
- Πεδία που ρυθμίζουν επιμέρους συζητήσεις που πραγματοποιούνται ταυτόχρονα (**Conversation\_ID, Reply-with**).

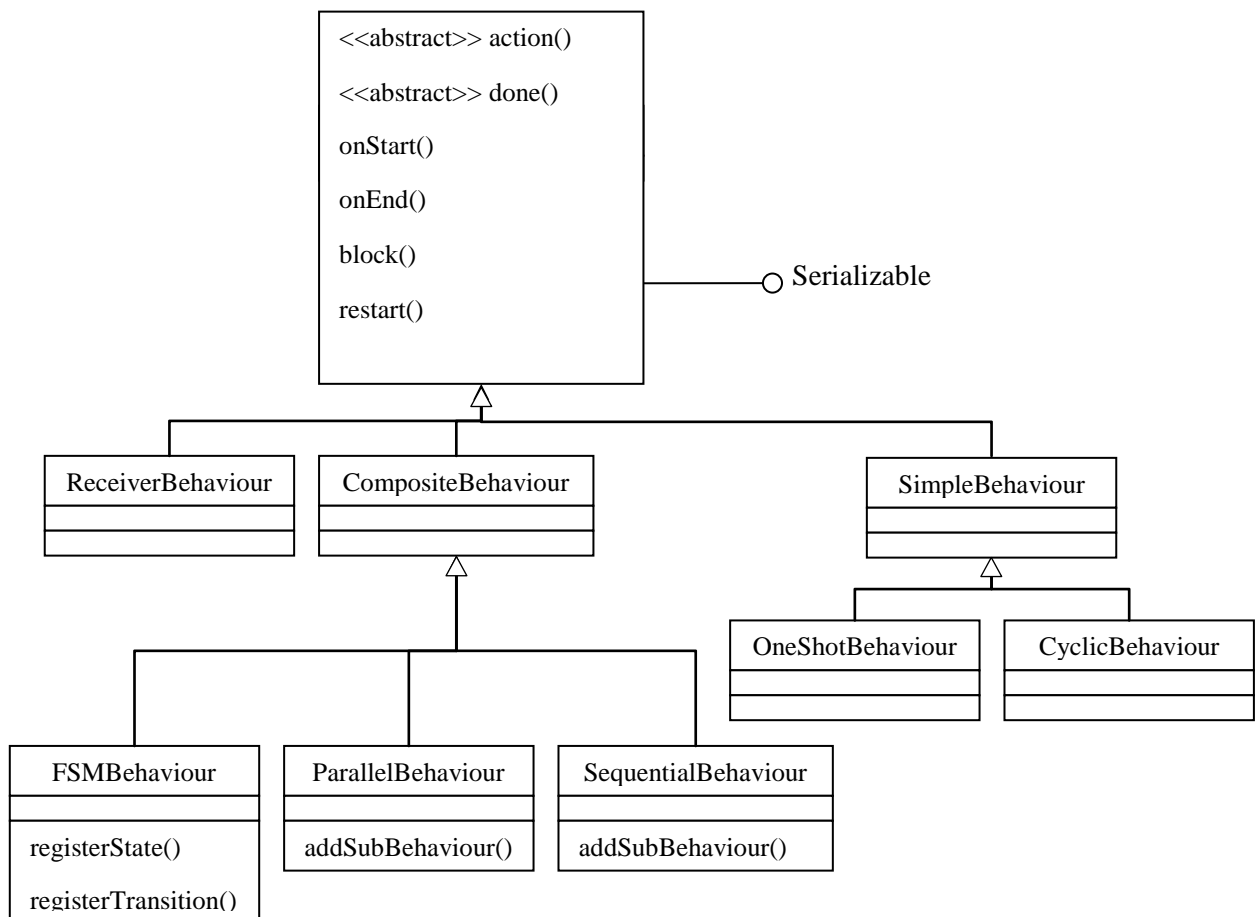
### ***3.3 Η συμπεριφορά ενός πράκτορα και η κλάση behavior***

Η συμπεριφορά είναι μία αφηρημένη κλάση που χρησιμοποιείται για τη μοντελοποίηση των εργασιών που ο πράκτορας εκτελεί. Έτσι, ένα σύνολο συμπεριφορών προγραμματίζονται και εκτελούνται για να διεκπεραιώσουν τις υποχρεώσεις του κάθε πράκτορα. Οι πράκτορες της κλάσης agent, που θα παρουσιαστεί εκτενώς παρακάτω, μπορούν να προσθέσουν ή να αφαιρέσουν συμπεριφορές με τις ακόλουθες μεθόδους:

- addBehaviour(Behavior)
- removeBehaviour(Behavior)



Ο λόγος για τον οποίο η προσθήκη/αφαίρεση συμπεριφορών είναι ουσιώδεις είναι το γεγονός ότι ένας πράκτορας πρέπει να εκτελεί πολλές εργασίες και να ανταποκρίνεται σε πολλά εξωτερικά ερεθίσματα παράλληλα. Το τελευταίο επιτυγχάνεται με την εισαγωγή της κλάσης behaviour, η οποία δίνει τη δυνατότητα στον πράκτορα να ανταποκρίνεται κατάλληλα στα ερεθίσματα. Η εκτέλεση τους δεν γίνεται παράλληλα, αλλά γίνεται διαδοχικά. Κάθε φορά που ολοκληρώνεται ένας κύκλος λειτουργίας μιας behaviour συνεχίζει η επόμενη. Το τέλος ενός κύκλου λειτουργίας μιας behaviour καθορίζεται από το χρήστη. Με τον τρόπο αυτό μπορούμε να πραγματοποιήσουμε πολύπλοκες διεργασίες για τους agents. Έτσι, η κλάση behaviour προσφέρει μεθόδους για την ενεργοποίηση της συμπεριφοράς(action()), την ολοκλήρωσή της(done()) και την επανεκκίνησή της(restart()).



Σχήμα 1: δομή της κλάσης Behaviour

Η βιβλιοθήκη της JADE έχει στη διάθεση της διάφορες κατηγορίες behaviours για την καλύτερη εξυπηρέτηση του προγραμματιστή. Αυτές είναι οι:

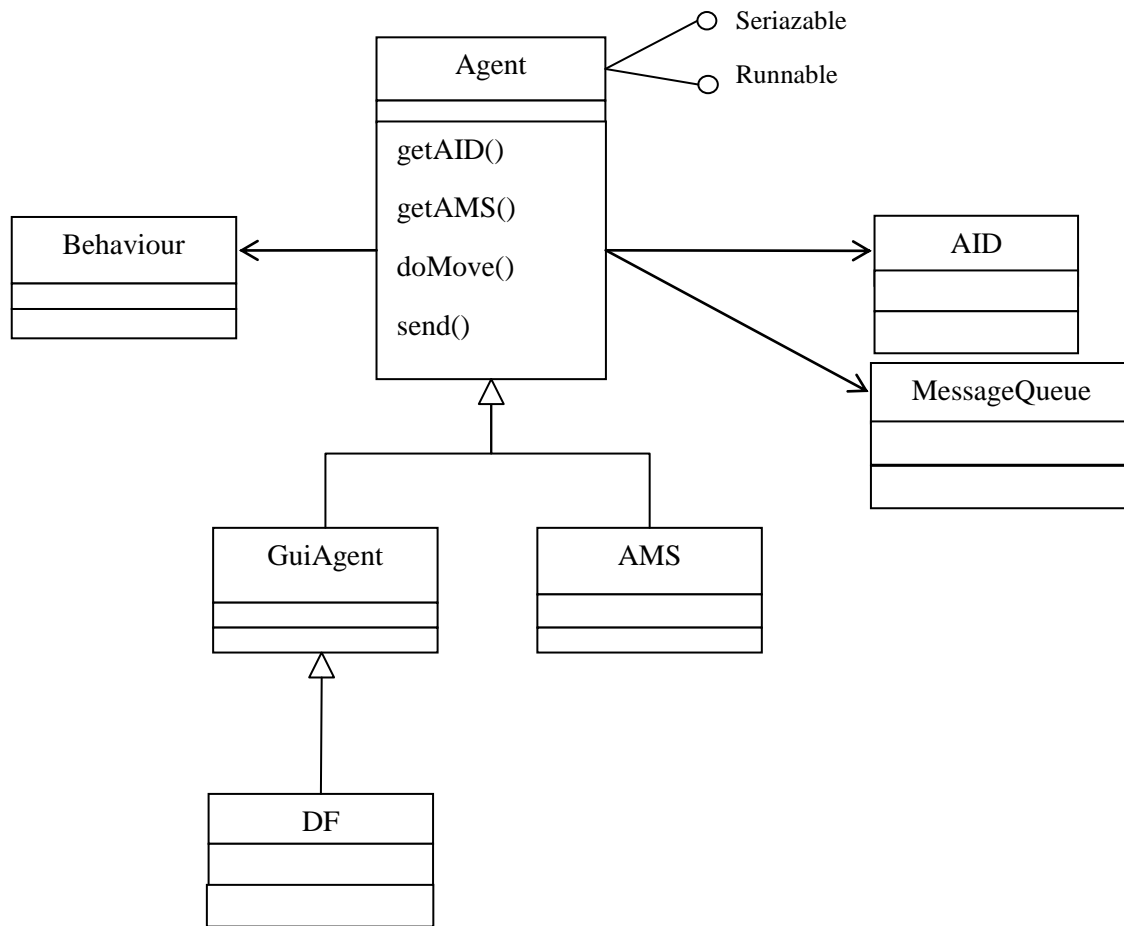
- **ReceiverBehaviour:** είναι η συμπεριφορά που εκτελείται κάθε φορά που έρχεται το μήνυμα που ανταποκρίνεται στο Message Template, που έχουμε ορίσει κατά την αρχικοποίηση της συμπεριφοράς.
- **SimpleBehaviour:** είναι η απλούστερη συμπεριφορά ενός διαμεσολαβητή, που είτε εκτελείται μια φορά(oneShotBehaviour), είτε εκτελείται περιοδικά(CyclicBehaviour).
- **CompositeBehaviour:** αποτελούν τις πιο περίπλοκες συμπεριφορές, όπου αποτελούνται από πολλές υποσυμπεριφορές και οι οποίες εκτελούνται ανάλογα με τις ανάγκες της εφαρμογής. Είναι είτε μια σειρά από καθορισμένες διαδοχικές συμπεριφορές

### ***3.4 Η κλάση agent***

Η κλάση agent που χρησιμοποιείται κατά κόρον στην εφαρμογή μας περιγράφει τη λειτουργικότητα ενός πράκτορα. Όταν λοιπόν δημιουργείται ένα αντικείμενο της κλάσης αυτής μας δίνεται η δυνατότητα να:

- Περιγράψουμε έναν πράκτορα(`Agent()`),
- Ορίσουμε/Βρούμε τη διεύθυνσή του(`setAdress()/getAdress()`),
- Ορίσουμε/Βρούμε το όνομά του(`setName()/getName()`),
- Ορίσουμε/Βρούμε το πρωτόκολλο επικοινωνίας με άλλους agents(`setProtocol()/getProtocol()`).

Πιο συγκεκριμένα οι πρακτορες JADE έχουν την ακόλουθη δομή:



Σχήμα 2: Agent Class diagram

Όπως περιγράψαμε και προηγουμένως, κάθε υπηρεσία και λειτουργία του πράκτορα υλοποιείται ως συμπεριφορά (class Behaviour). Ένας scheduler, ο οποίος δεν είναι φανερός στον προγραμματιστή, διαχειρίζεται και εκτελεί τις behaviors διαδοχικά.

Ένας πράκτορας μόλις εισέρχεται στην πλατφόρμα εκτέλεσης λαμβάνει μία μοναδική ταυτότητα(AID) και με χρήση αυτής της ταυτότητας δέχεται και στέλνει μηνύματα κατά μήκος αυτής.

Ο **AMS (Agent Management System)** παρέχει την «υπηρεσία ονόματος», naming service ,που διαβεβαιώνει ότι κάθε πράκτορας στην πλατφόρμα έχει μοναδικό όνομα και ασκεί εξουσία στο σύστημα. Η εξουσία αυτή συνοψίζεται στο γεγονός ότι μπορεί να μπορεί να δημιουργεί και να τερματίζει τη λειτουργία πρακτόρων.

Ο **DF (Directory Facilitator)** παρέχει την υπηρεσία «κίτρινων σελίδων», καθώς επιτρέπει σε κάθε πράκτορα να βρει κάποιον άλλο κατα μήκος της πλατφόρμας και να αιτηθεί άλλων υπηρεσιών.

Επιπλέον,παρέχονται μέθοδοι για την εναλλαγή μεταξύ καταστάσεων ενός πράκτορα, την αποστολή και παραλαβή μηνυμάτων, την εγγραφή και την τροποποίηση των εγγραφών στο DF και στο AMS.

# 4

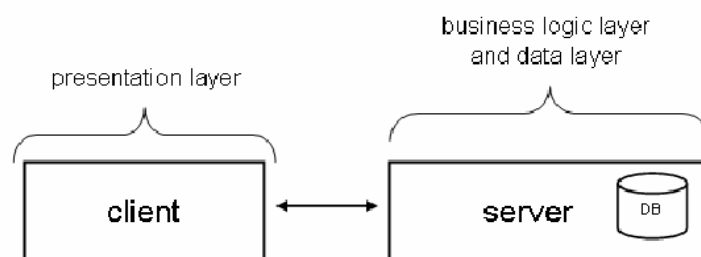
## Ανάλυση απαιτήσεων συστήματος

Στο συγκεκριμένο κεφάλαιο θα γίνει περιγραφή της αρχιτεκτονικής του συστήματος καθώς και ανάλυση απαιτήσεων για τις λειτουργίες του.

### 4.1 Αρχιτεκτονική

#### 4.1.1 Μοντέλο Πελάτη-Εξυπηρετητή(Client-Server Model)

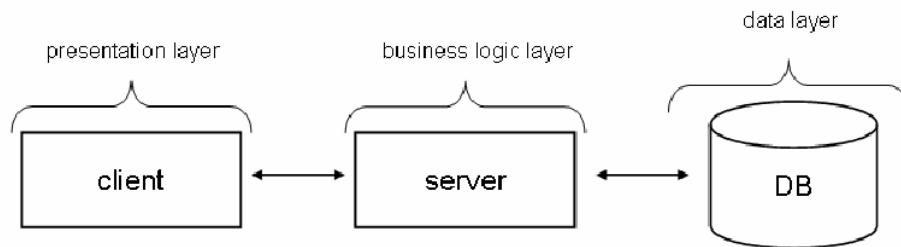
Όπως γνωρίζουμε, το μοντέλο πελάτη εξυπηρετητή μπορεί να είναι είτε 2-επιπέδων(2-tier), είτε 3-επιπέδων(3-tier), είτε γενικώς n-επιπέδων(n-tier).



Εικ.2: Αρχιτεκτονική 2-επιπέδων

Η αρχιτεκτονική που επιλέχθηκε για τη δική μας υλοποίηση είναι αυτή των τριών επιπέδων. Η αρχιτεκτονική αυτή αποτελεί την πλέον διαδομένη, όταν πρόκειται για διαδικτυακές εφαρμογές με βάση δεδομένων. Συγκεκριμένα, όπως φαίνεται και στο παρακάτω σχήμα, αποτελείται από τα εξής επίπεδα:

- Επίπεδο παρουσίασης(Presentation Tier),
- Επίπεδο επιχειρηματικής λογικής(Business Tier),
- Επίπεδο Δεδομένων(Data Tier).



Εικ.3:Αρχιτεκτονική 3-επιπέδων

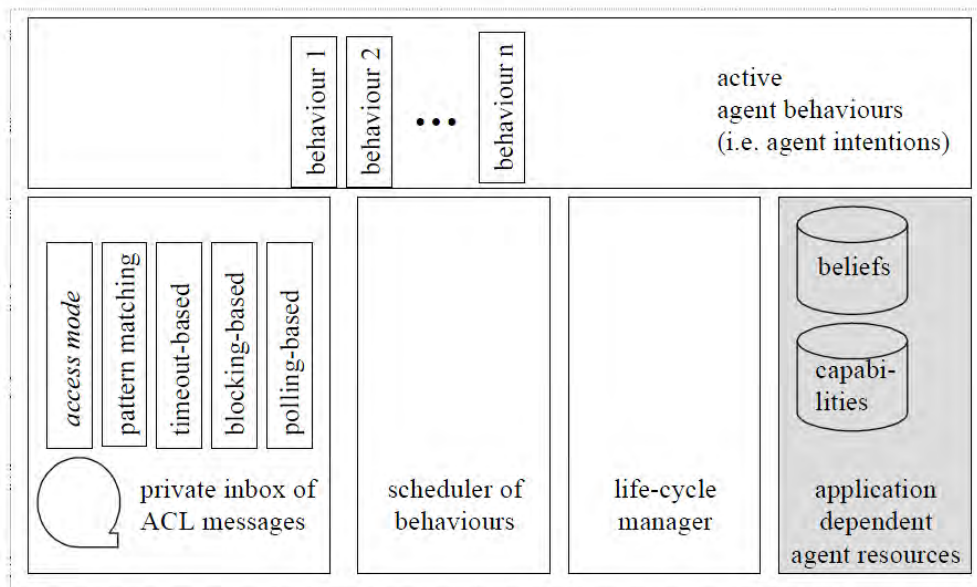
Ο λόγος που επιλέχθηκε αυτή η αρχιτεκτονική έναντι του απλουστευμένου μοντέλου 2-επιπέδων είναι γιατί στο 3-tier μοντέλο το επίπεδο δεδομένων είναι διαχωρισμένο από το επίπεδο επιχειρηματικής λογικής, γεγονός που δεν συμβαίνει στο 2-tier μοντέλο. Ο διαχωρισμός παρουσιάζει τα εξής πλεονεκτήματα έναντι του μη-διαχωρισμού:

- Μεγαλύτερη ευελιξία στο πρόγραμμά μας,
- Μεγαλύτερη αξιοπιστία και ασφάλεια, καθώς πρόσβαση στη βάση έχει μόνο ο server .
- Μεγαλύτερη ταχύτητα στις επερωτήσεις στη Βάση Δεδομένων.
- Δυνατότητα επεκτασιμότητας της εφαρμογής

Επίσης, αξίζει να αναφέρουμε ότι η αρχιτεκτονική περισσότερων επιπέδων δεν κρίθηκε απαραίτητη, παρόλη την αυξημένη ευελιξία που αυτή θα παρείχε, λόγω των απλουστευμένων λειτουργιών που απαιτούσε η εφαρμογή μας.

#### 4.1.2 Αρχιτεκτονική JADE agents

Όπως προαναφέρθηκε, χρησιμοποιούμε την πλατφόρμα εκτέλεσης JADE για την αλληλεπίδραση των πρακτόρων. Κάθε JADE πράκτορας αποτελείται από κάποια επιμέρους τμήματα. Όπως βλέπουμε και στο ακόλουθο σχήμα, κάθε πράκτορας διαθέτει ένα σύνολο συμπεριφορών ,μια ουρά μηνυμάτων ACL για την ανταλλαγή μηνυμάτων με άλλους πράκτορες και ένα δρομολογητή συμπεριφορών που ρυθμίζει τη εφαρμογή των συμπεριφορών. Επίσης, απαραίτητος είναι ο ρυθμιστής κύκλου ζωής, που δίνει εντολές για τον τερματισμό ή την έναρξη ζωής ενός agent.

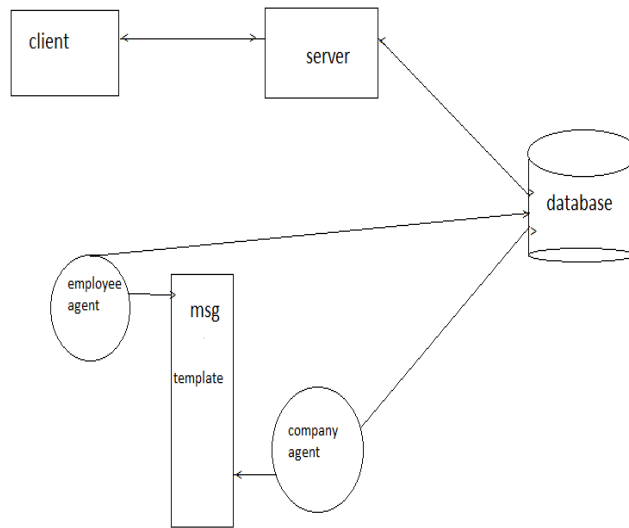


Εικ.4: Αρχιτεκτονική JADE agent

Τέλος, χρησιμοποιείται και μια βάση δεδομένων, στην οποία έχουν πρόσβαση οι πράκτορες και η οποία περιέχει τα στοιχεία των χρηστών(εργαζομένων/εταιρειών).

Στην εφαρμογή μας απαραίτητη ήταν η δημιουργία δύο πρακτόρων,έναν που θα εκπροσωπούσε τους χρήστες προς αναζήτηση εργασίας(employeeAgent) και έναν που θα εκπροσωπούσε τις εταιρείες προς αναζήτηση εργαζομένων(companyAgent).Αυτοί οι δύο πράκτορες έρχονται σε επαφή και πραγματοποιούν μία συμφωνία, με τον τρόπο που θα περιγραφεί αναλυτικά σε επόμενες ενότητες.

Συνολικά, προκύπτει το κάτωθι σχήμα:



Σχήμα 3:Συνολική εικόνα εφαρμογής

Εδώ παρουσιάζουμε τα επιμέρους κομμάτια (υποσυστήματα) από τα οποία θεωρούμε ότι αποτελείται το σύστημά μας. Για κάθε υποσύστημα δίνεται μια σύντομη περιγραφή (η αναλυτική περιγραφή των λειτουργιών του ακολουθεί στην επόμενη ενότητα). Τέλος, δίνουμε ένα γενικό σχήμα που δείχνει τα υποσυστήματα και πώς αυτά επικοινωνούν μεταξύ τους. Το σχήμα αυτό αρκεί να είναι απλό block diagram και όχι Διάγραμμα Ροής Δεδομένων (DFD).

## 4.2 Περιγραφή Λειτουργιών

Σκοπός της εφαρμογής μας ήταν η δημιουργία κάποιων λειτουργιών που θα πλαισίωναν τη διεπαφή του χρήστη αλλά και το περιβάλλον λήψης απόφασης βάσει κριτηρίων. Παρακάτω, για κάθε υποσύστημα γίνεται λεπτομερής περιγραφή των λειτουργιών του.

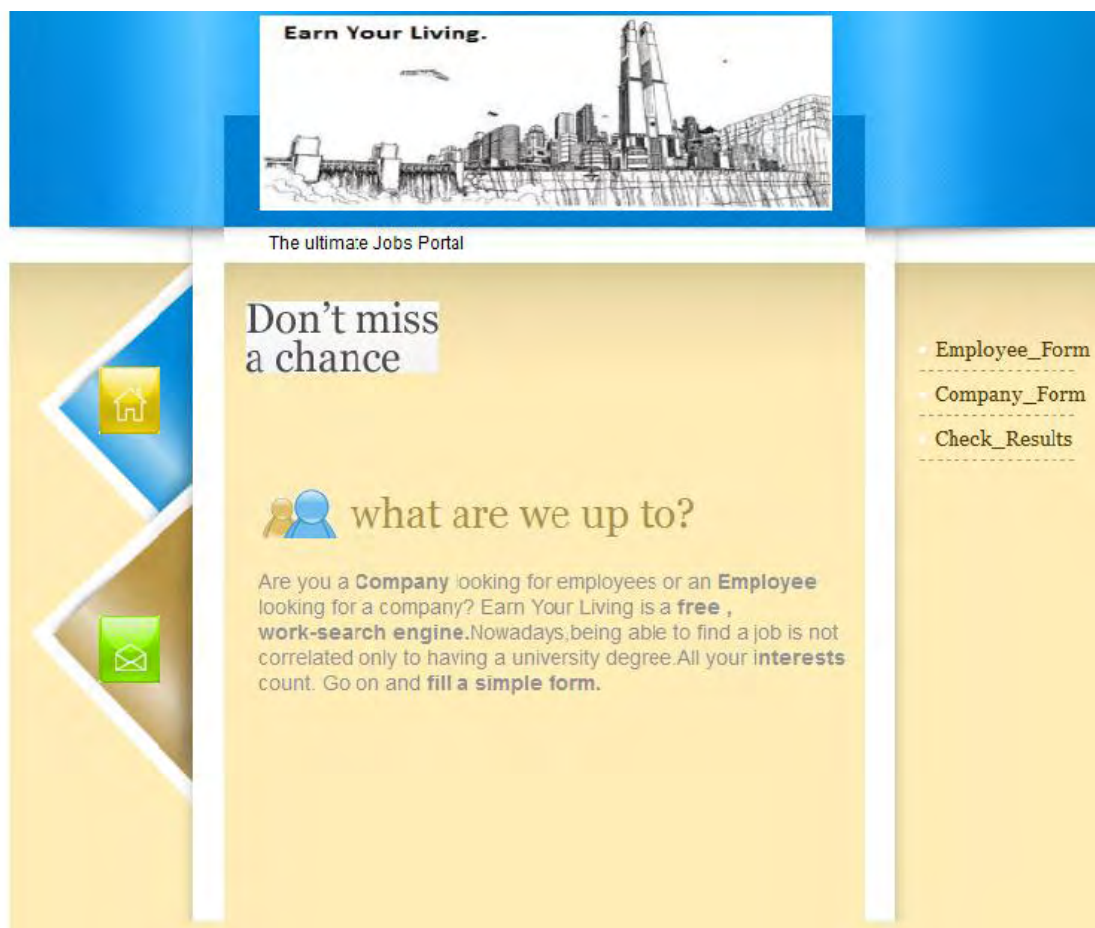


#### 4.2.1 Διεπαφή χρήστη(user interface)

Η εφαρμογή μας διαθέτει ένα περιβάλλον διεπαφής,με το οποίο έρχονται σε επαφή οι χρήστες. Όπως έχει γίνει ήδη σαφές έχουμε δύο τύπους χρηστών:

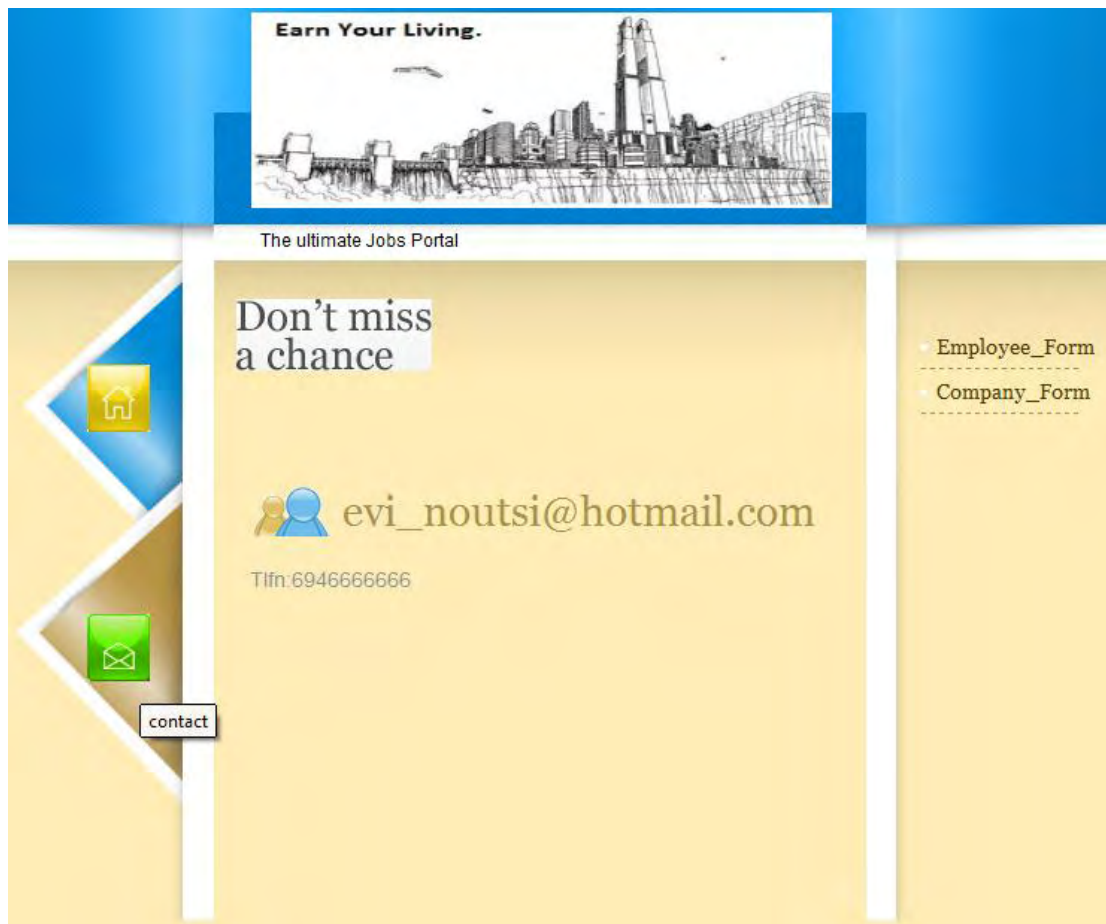
- Αιτούντες εργασία(employees),
- Προσφέροντες εργασία(companies)

Ένας χρήστης μπορεί μέσω ενός web browser να συνδεθεί με τη σελίδα Earn Your living πληκτρολογώντας το αντίστοιχο url. Στην κεντρική σελίδα μπορεί να δει πληροφορίες για τις δυνατότητες που του παρέχονται για εύρεση εργασίας/υπαλλήλων:



Εικ 5: Κεντρική σελίδα

Επιπλέον, μπορεί να έρθει σε επικοινωνία με τους εμπνευστές του εγχειρήματος αυτού, μέσω του υπερσυνδέσμου επικοινωνίας (contact).



Εικ.6: Σελίδα επικοινωνίας

Στη συνέχεια, δίνεται η δυνατότητα στους χρήστες να χρησιμοποιήσουν τις ουσιαστικές λειτουργίες που παρέχει η διαδικτυακή εφαρμογή. Οι τελευταίες προκαλούν καταχώρηση στη Βάση Δεδομένων μέσω σχετικού αιτήματος στον εξυπηρετητή, αφού ο εξυπηρετητής συνδέεται με τη Βάση Δεδομένων. Το τελευταίο βασίζεται στο μοντέλο πελάτη-εξυπηρετητή που δείξαμε παραπάνω. Έτσι λοιπόν, διακρίνουμε τις εξής λειτουργίες:

- Συμπλήρωση φόρμας εργαζομένου:

Στη συγκεκριμένη φόρμα ο πελάτης εισάγει τα στοιχεία του, όνομα, επώνυμο, προσωπικό όνομα χρήστη και κωδικό χρήστη, καθώς και την εκπαιδευτική και επαγγελματική κατάρτισή του. Επίσης συμπληρώνει επιπλέον προσόντα που έχει αποκτήσει κατά τη διάρκεια της ζωής του, όπως γνώσεις ξένων γλωσσών.

The image shows a web page for a job portal. At the top, there is a blue banner with the text "Earn Your Living." and a black and white illustration of a city skyline. Below the banner, the text "The ultimate Jobs Portal" is displayed. The main content area has a yellow background and features a registration form titled "Don't miss a chance". The form includes the following fields and options:

- Please fill the following form :
- Firstname:
- Surname:
- username:
- Password:
- Mail:
- EducationLevel:
- Grade:
- Postgraduate Diploma:
- Working Experience(in months):
- English speaker:
- French speaker:
- 

Εικ.7: Φόρμα συμπλήρωσης στοιχείων εργαζομένου


- Συμπλήρωση φόρμας εταιρείας

Ομοίως και οι εταιρείες συμπληρώνουν στη φόρμα που τους αντιστοιχεί τα στοιχεία που τους περιγράφουν(όνομα,όνομα χρήστη και κωδικός χρήστη) και συνεχίζουν με τις απαιτήσεις που θέτουν για τους μελλοντικούς υπαλλήλους τους.Στο σημείο αυτό αξίζει να αναφερθεί ότι για κάθε απαίτηση που θέτουν(ακαδημαϊκή κατάρτιση κλπ.) οι εταιρείες καλούνται να συμπληρώσουν και το βαθμό βαρύτητας σε κλίμακα από 1 έως 5 του συγκεκριμένου στοιχείου.

**Earn Your Living.**



The ultimate Jobs Portal




## Don't miss a chance

Please fill the following form :

Name

username

Password

Please fill the following by also providing weight of importance(5=higher):

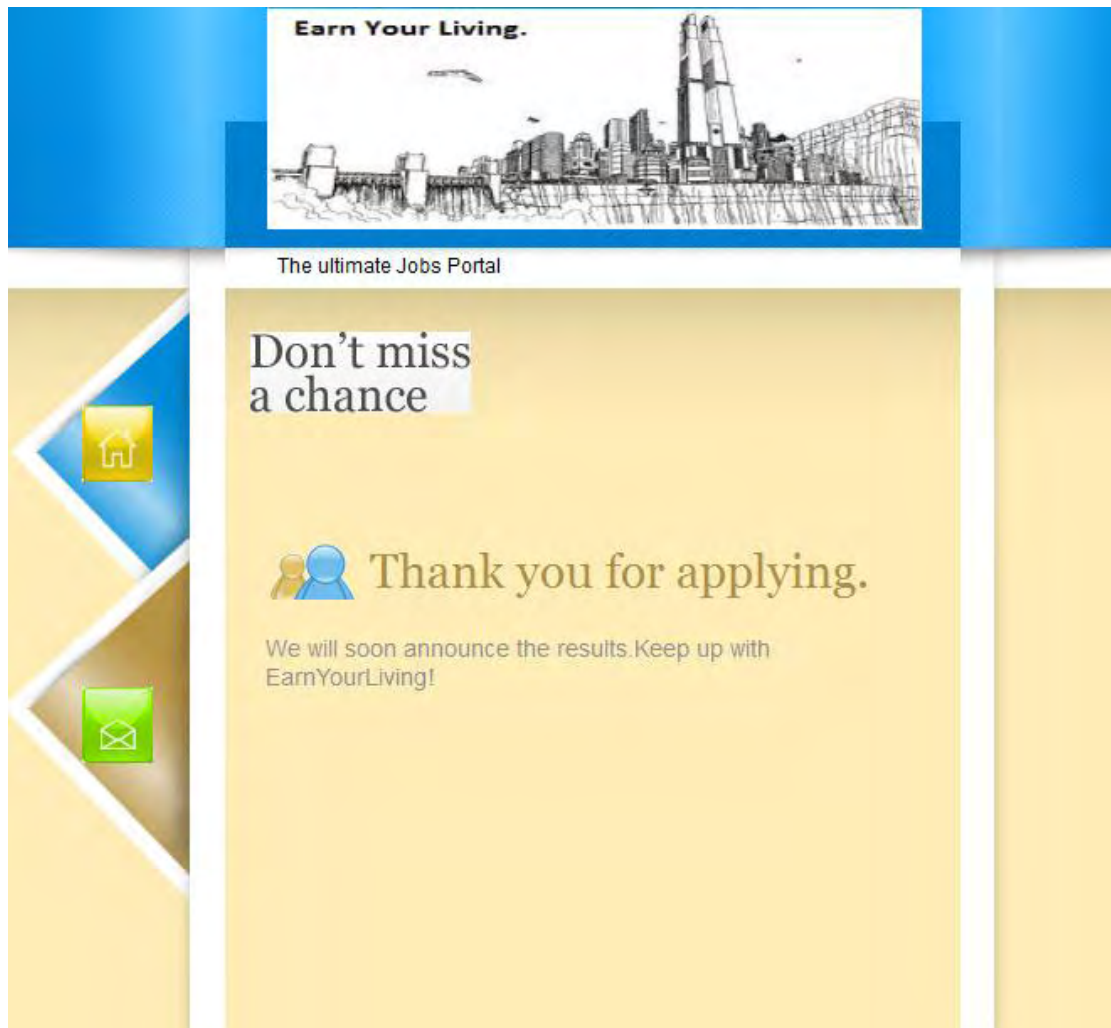
Wanted academic level	University Degree ▾	1 ▾
Degree(Lower wanted)	<input style="width: 100%;" type="text"/>	1 ▾
Postgraduate Diploma	Yes ▾	1 ▾
Working Experience(min months).	<input style="width: 100%;" type="text"/>	1 ▾
English speaker.	Yes ▾	1 ▾
French speaker.	Yes ▾	1 ▾

Εικ.8: Φόρμα συμπλήρωσης στοιχείων για εταιρείες

Με το βαθμό σημαντικότητας που προστέθηκε παραπάνω , μία εταιρεία μπορεί να δηλώσει το βαθμό στον οποίο θεωρεί απαραίτητο ένα συγκεκριμένο προσόν. Για παράδειγμα, ένας χρήστης-εταιρεία που θεωρεί πολύ σημαντικό το προσόν του μεταπτυχιακού διπλώματος μπορεί να το βαθμολογήσει με «5», ενώ ταυτόχρονα όταν η γνώση της γαλλικής δεν είναι ιδιαίτερως σημαντική μπορεί να τη βαθμολογήσει με «3». Η συγκεκριμένη βαθμολόγηση έχει ιδιαίτερη σημασία στη διαδικασία της αξιολόγησης του εργαζομένου, όπως θα εξηγήσουμε στη συνέχεια.

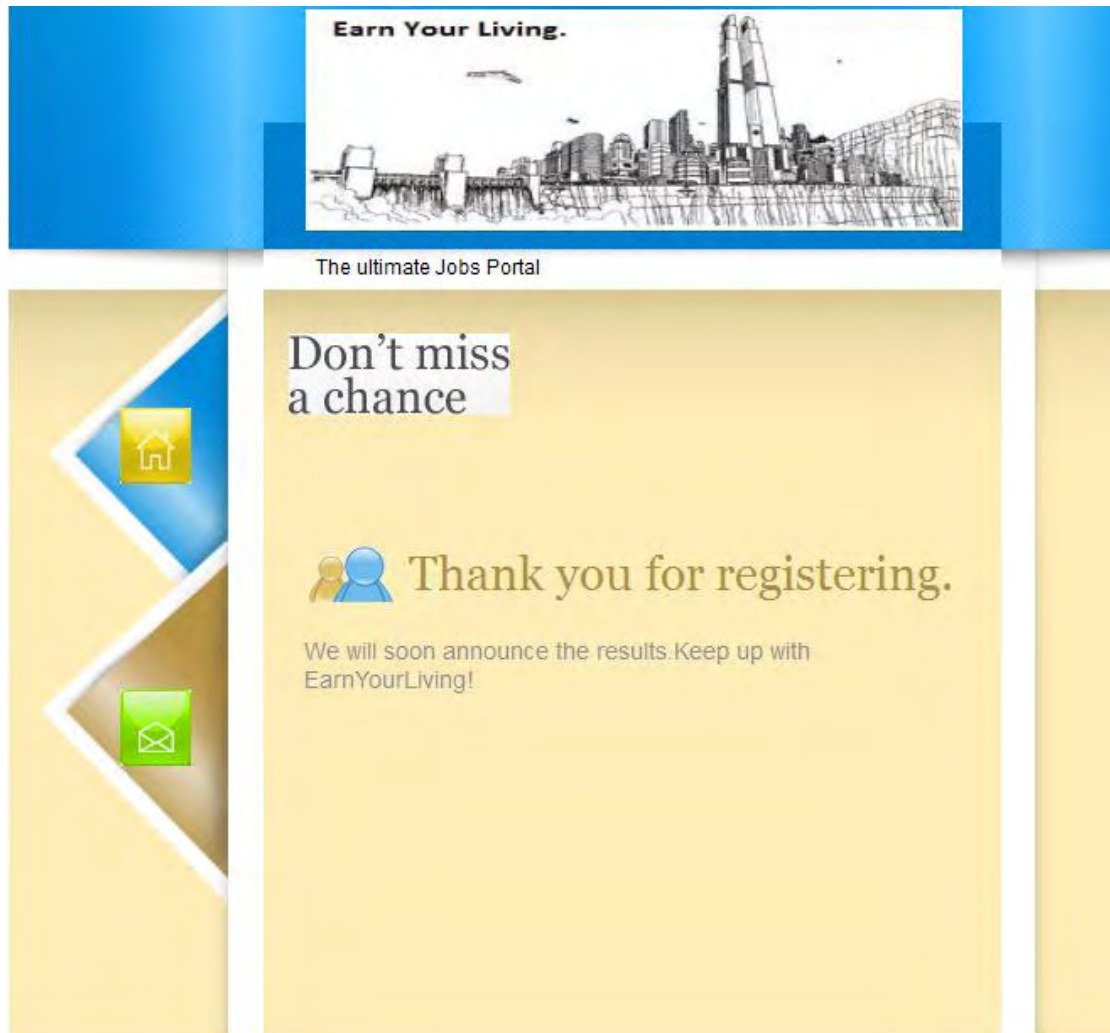
- Επιτυχής αποθήκευση δεδομένων στη βάση:

Μετά από επιτυχημένη αποθήκευση των δεδομένων στη βάση εμφανίζονται οι εξής σελίδες για τους εργαζομένους και τις εταιρείες αντιστοίχως:



Εικ.9:Επιτυχής αποθήκευση για εργαζομένους





Εικ.10:Επιτυχής αποθήκευση για εταιρείες

- Έλεγχος αποτελεσμάτων(check results),

Αφού ο χρήστης δηλώσει την ιδιότητά του ως εργαζόμενος ή εταιρεία, μέσω σχετικής επιλογής στη φόρμα ελέγχου, δίνεται η δυνατότητα ελέγχου των αποτελεσμάτων της αίτησης.

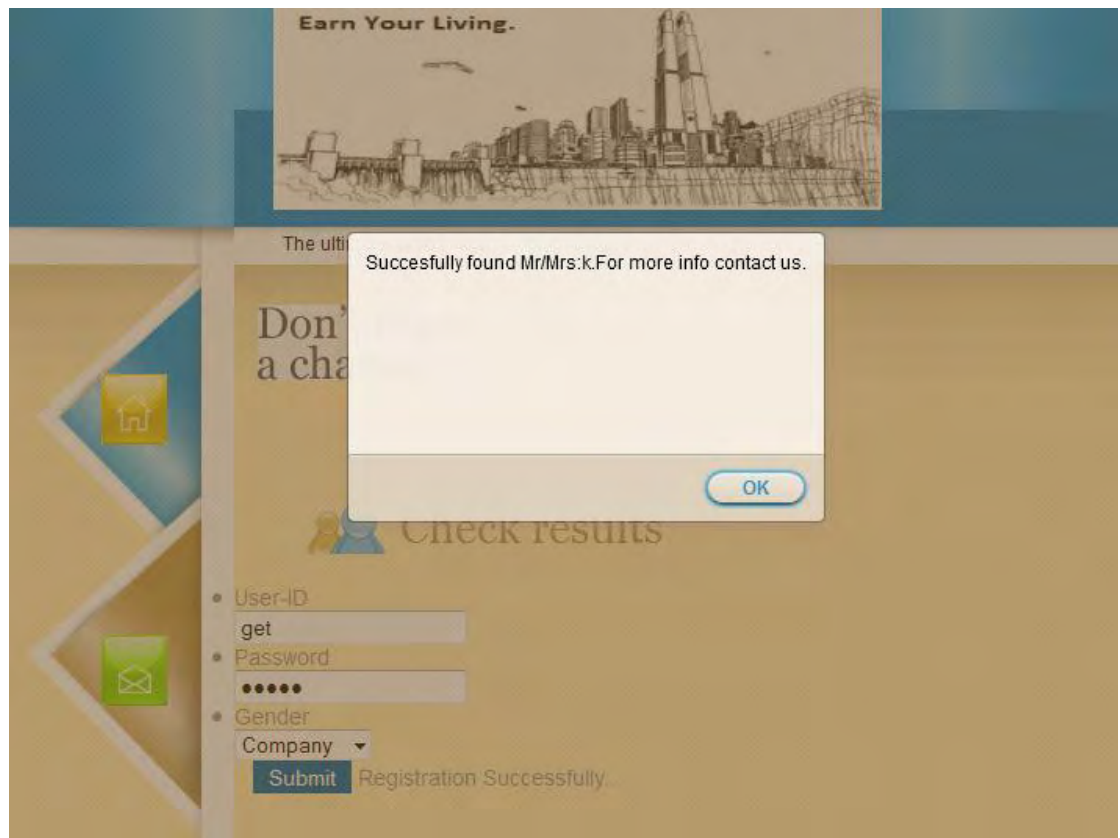
Ο χρήστης σε αυτό το σημείο δίνει ως στοιχεία εισόδου το όνομα χρήστη και τον κωδικό πρόσβασής του, που είχε συμπληρώσει προηγουμένως στη δοθείσα φόρμα συμπλήρωσης.

Με τον τρόπο αυτό ο χρήστης γνωρίζει αν υπάρχει κάποιο αποτέλεσμα στην αναζήτηση εργασίας/εργαζομένου για αυτόν.



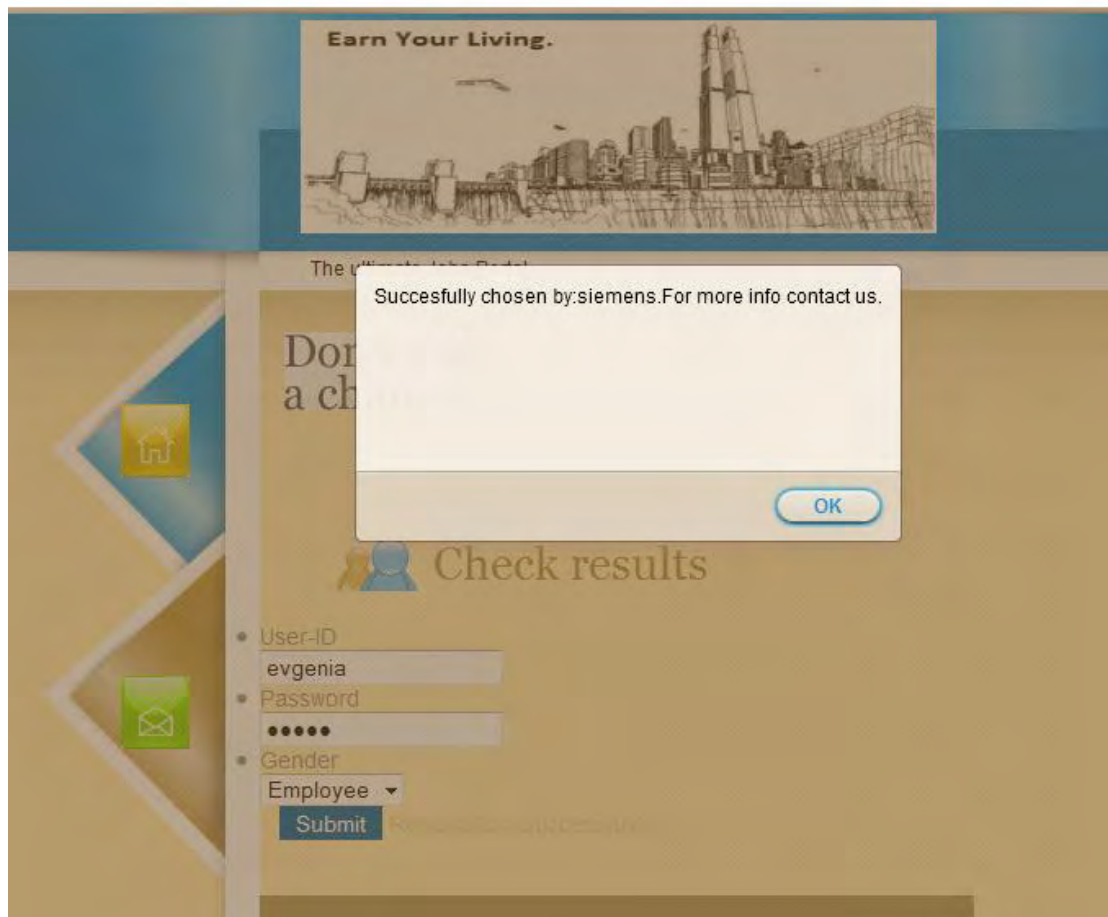
Εικ.11 : Έλεγχος Αποτελεσμάτων

- Εμφάνιση Αποτελεσμάτων

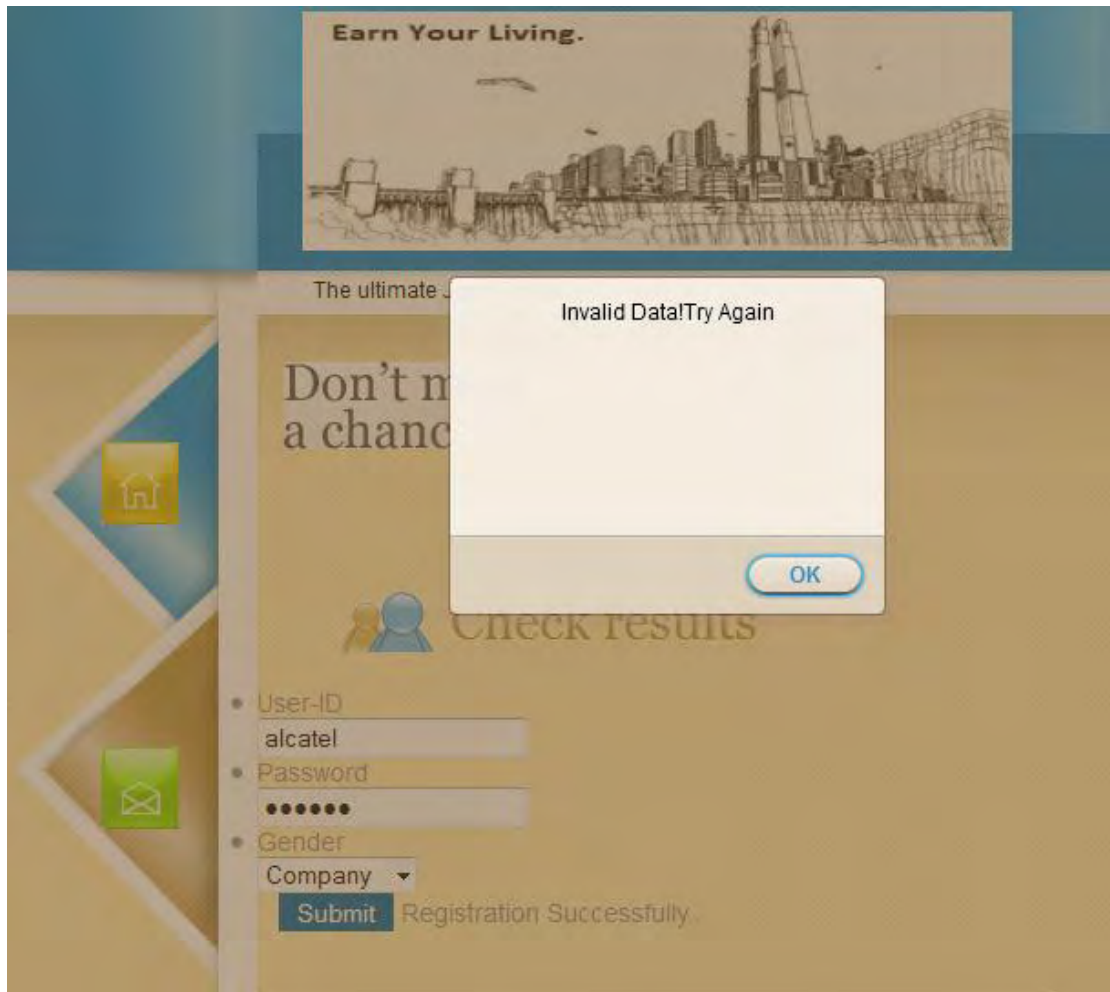


Εικ.12:Εμφάνιση επιτυχούς αποτελέσματος για εταιρείες





Εικ.13:Εμφάνιση επιτυχούς αποτελέσματος για εργαζομένους



Εικ.14:Εμφάνιση αποτυχημένης αναζήτησης

#### 4.2.2 Υποσύστημα Λήψης Απόφασης

Όπως προαναφέρθηκε, η εφαρμογή μας διαθέτει ένα υποσύστημα στο οποίο πράκτορες έρχονται σε επαφή με τη βάση δεδομένων, επικοινωνούν μεταξύ τους και γίνεται ανάθεση εργασίας ή εύρεση εργαζομένου. Σε αυτό το σημείο, οι λειτουργίες που πραγματοποιούν οι πράκτορες και τα σχετικά μηνύματα που εμφανίζονται στο προγραμματιστικό μας περιβάλλον είναι:

- Εισαγωγή στην πλατφόρμα και ορισμός AID (Agent Identification)
  - Οι companies:

```
Hallo! company a@192.168.1.65:1099/JADE is ready.  
Feb 16, 2012 7:30:52 MM jade.core.AgentContainerImpl joinPlatform  
INFO: -----  
Agent container Main-Container@192.168.1.65 is ready.  
-----
```

- Οι employees:

```
INFO: -----  
Agent container Container-1@192.168.1.65 is ready.  
-----  
Hallo! user b@192.168.1.65:1099/JADE is ready.
```

- Εντοπισμός users στην πλατφόρμα επικοινωνίας

*Found the following users:*

b@192.168.1.65:1099/JADE

c@192.168.1.65:1099/JADE

- Επικοινωνία και ανταλλαγή στοιχείων μεταξύ τους

*User:*

56    1        8        1        12        1        1

(η πλειάδα του συγκεκριμένου χρήστη σε πίνακα της Β.Δ.)

- Έλεγχος στη βάση δεδομένων,

*Checking User with the ID:56*

- Λήψη απόφασης βάσει κριτηρίων,

- *We have a match!*

- *Attempt failed : job still available.*

- Ενημέρωση της Βάσης Δεδομένων.

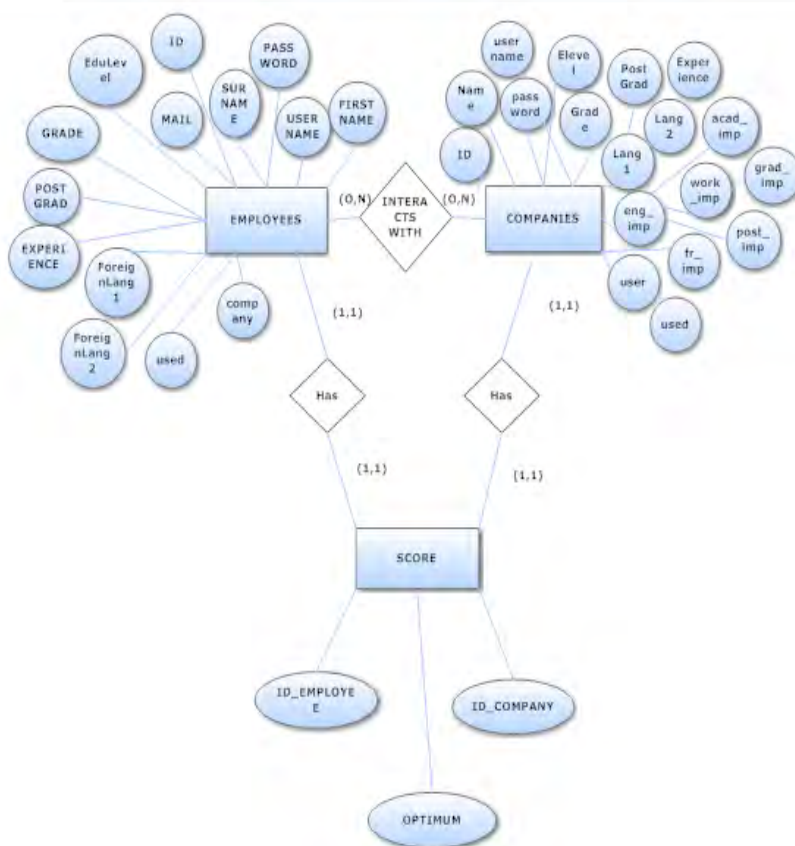
*Successfully purchased from agent a@192.168.1.65:1099/JADE*

*for the user with the ID = 58*

### 4.3 Μοντέλο Οντοτήτων Συσχετίσεων

Η πληροφορία την οποία χρησιμοποιεί η εφαρμογή Earn Your Living βρίσκεται σε μια βάση δεδομένων. Δόθηκε μεγάλη προσοχή στο να εξυπηρετεί η βάση με τον καλύτερο τρόπο τις λειτουργίες που αναπτύχθηκαν, αλλά και να είναι απλή και εύκολα επεκτάσιμη.

Τα στοιχεία που κρατούνται στη βάση είναι οι εργαζόμενοι που αιτούνται εργασίας, οι εταιρείες που προσφέρουν εργασία και η βαθμολογία των εργαζομένων, μετά την εξατομικευμένη αξιολόγηση των κριτηρίων τους. Όλα τα παραπάνω συνδέονται μεταξύ τους με τον τρόπο που φαίνεται στο ακόλουθο διάγραμμα οντοτήτων-συσχετίσεων (ER) με τις αντίστοιχες πληθικότητες:



Σχήμα 4: Μοντέλο Οντοτήτων Συσχετίσεων

Παρατηρούμε παραπάνω ότι υπάρχουν δύο τρόποι αλληλεπίδρασης μεταξύ εργαζομένων και εταιρειών: ένας **άμεσος** και ένας **έμμεσος**. Ο άμεσος τρόπος καθορίζεται από τον κόμβο `interacts with`, όπου και βλέπουμε ότι μετά από τη διαδικασία εισαγωγής των στοιχείων από την κάθε μεριά, γίνεται ένα απλό ταίριασμα κριτηρίων με τρόπο θα δείξουμε παρακάτω.

Όσον αφορά στο δεύτερο τρόπο αλληλεπίδρασης, αυτός γίνεται δημιουργώντας για έναν εργαζόμενο ένα score ,που αντιστοιχεί σε μία εταιρεία. Γι' αυτό τον λόγο βλέπουμε ότι στην οντότητα `score` υπάρχουν ως γνωρίσματα το `id` του εργαζομένου και της εταιρείας αλλά και το `optimum`, η μεγαλύτερη τιμή του score που έλαβε ο εργαζόμενος. Περισσότερα για τον τρόπο ταίριασματος κριτηρίων που επιλέχθηκε ακολουθεί σε επόμενα κεφάλαια.

## ***4.4 Τεχνολογίες που χρησιμοποιήθηκαν***

### ***4.4.1 HTML***

Η HTML(Hyper Text Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML, τα οποία αποτελούνται από ετικέτες(tags). Ένας web browser διαβάζει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει. Ο browser χρησιμοποιεί τις ετικέτες για να ερμηνεύσει το περιεχόμενο της ιστοσελίδας.

### ***4.4.2 CSS***

Η CSS(Cascading Style Sheets) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ, που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφεί με μια γλώσσα σήμανσης. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα.

### ***4.4.3 Java Server Pages (JSP)***

Η JSP είναι μία τεχνολογία της JAVA που διευκολύνει τους προγραμματιστές να εξυπηρετούν δυναμικές ιστοσελίδες βασισμένες σε HTML ή XML. Από αρχιτεκτονικής απόψεως η JSP μπορεί να θεωρηθεί μία αφαίρεση υψηλού επιπέδου των Java Servlets . Οι JSPs “φορτώνονται” στον εξυπηρετητή και εκτελούνται από ένα ειδικό δομημένο JAVA πακέτο ,το Java EE web application.

#### **4.4.4 JAVASCRIPT**

Η JAVASCRIPT είναι γλώσσα συγγραφής σεναρίων (scripting language), που σχεδιάστηκε για να προσθέσει διαδραστικότητα στις σελίδες HTML. Είναι interpreted language, δηλαδή τα σενάρια εκτελούνται χωρίς να έχει γίνει νωρίτερα compilation και επίσης είναι case-sensitive. Τέλος, είναι δωρεάν και αποτελεί ένα ιδιαίτερος χρήσιμο υπολογιστικό εργαλείο.

#### **4.4.5 MySQL**

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, που παρέχει σε πολλούς χρήστες δυνατότητα πρόσβασης σε ένα σύνολο πινάκων δεδομένων.

#### **4.4.6 AJAX**

Η AJAX (Asynchronous JavaScript and XML) είναι ένα σύνολο από συσχετιζόμενες τεχνικές διαδικτυακής ανάπτυξης, οι οποίες εφαρμόζονται στη μεριά του πελάτη (client-side) για να δημιουργήσουν ασύγχρονες διαδικτυακές εφαρμογές. Με τη χρήση της AJAX, οι διαδικτυακές εφαρμογές μπορούν να στέλνουν καθώς και να λαμβάνουν δεδομένα από τον εξυπηρετητή με ασύγχρονο τρόπο, χωρίς να εμπλέκονται με την εμφάνιση και τη συμπεριφορά της σελίδας. Τα δεδομένα ανακτώνται χρησιμοποιώντας το XMLHttpRequest αντικείμενο. Αξίζει να αναφερθεί ότι η ουσιαστική συμβολή είναι ότι η JavaScript και το XMLHttpRequest αντικείμενο παρέχουν μια μέθοδο ανταλλαγής δεδομένων μεταξύ εξυπηρετητή και browser με τρόπο ασύγχρονο. Το τελευταίο σημαίνει ότι αποφεύγονται reloads ολόκληρων σελίδων και γίνεται αντιληπτό ότι ενισχύεται η χρησιμότητα και η ταχύτητα της διαδικτυακής μας εφαρμογής.

#### **4.4.7 JQuery**

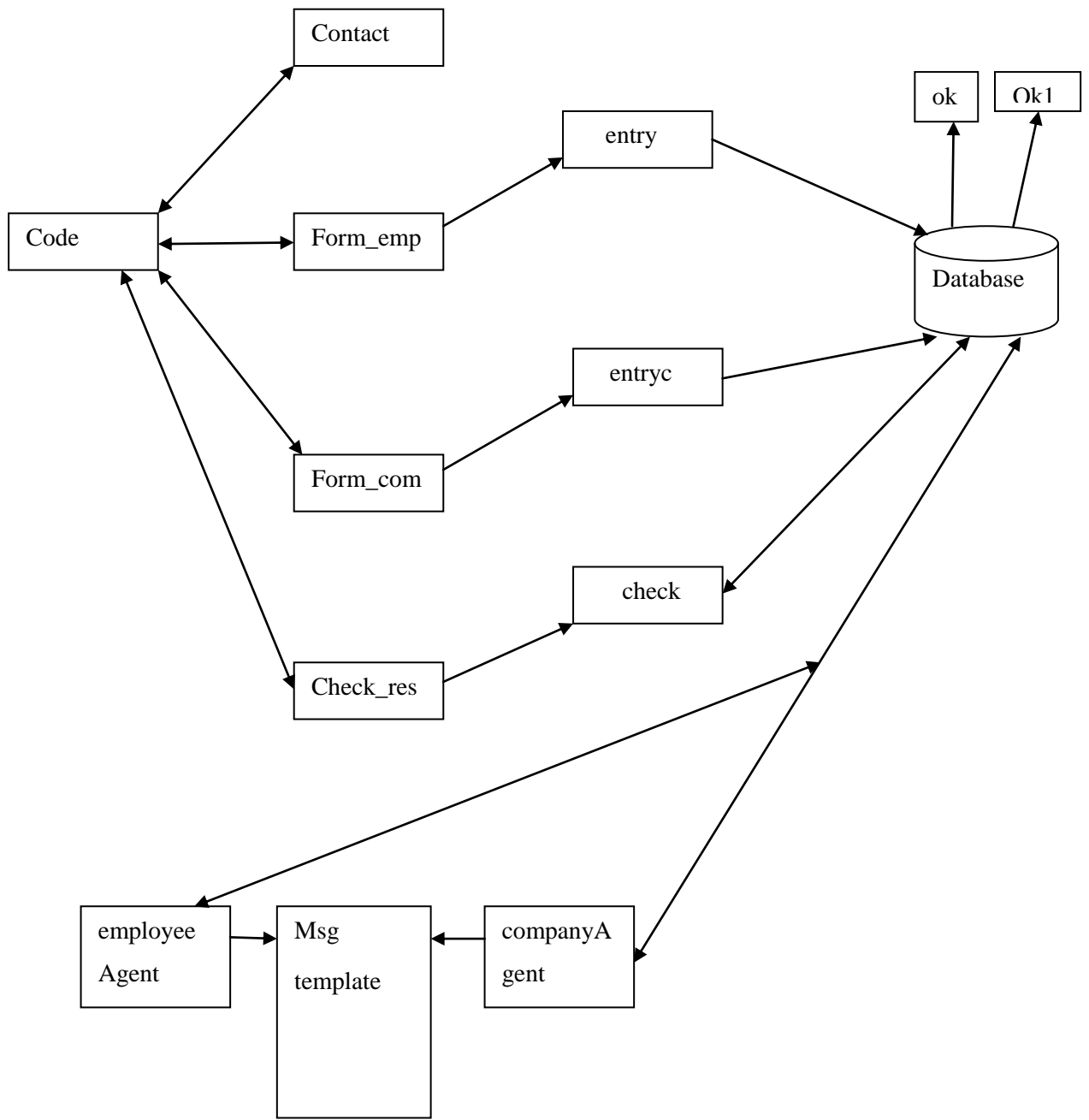
Η JQuery είναι μια βιβλιοθήκη της JavaScript σχεδιασμένη για να απλοποιήσει τη χρήση της τελευταίας. Πιο συγκεκριμένα, με χρήση της JQuery είναι πιο εύκολη η πλοήγηση σε αρχεία, διαχείριση γεγονότων (handle events), επιλογή DOM στοιχείων (Document Object Model) αλλά και ανάπτυξη AJAX εφαρμογών.

# 5

## *Σχεδίαση Συστήματος*

### *5.1 Αρχιτεκτονική*

Στη συγκεκριμένη ενότητα θα παρουσιαστούν τα επιμέρους κομμάτια από τα οποία θεωρούμε ότι χτίστηκε ο κώδικάς μας. Ακολουθεί ένα γενικό σχήμα που δείχνει τον τρόπο που επικοινωνούν μεταξύ τους τα αρχεία της διαδικτυακής εφαρμογής και οι κλάσεις των πρακτόρων.



Σχήμα 5: Διάγραμμα αρχείων web application



Η υλοποίηση της εφαρμογής μας, όσον αφορά στο διαδικτυακό μέρος, πραγματοποιήθηκε δημιουργώντας αρχεία .jsp τα οποία θα υπηρετούσαν διαδοχικά τις προαναφερθείσες λειτουργικότητες. Επιπλέον,όσον αφορά στο πολυπρακτορικό σύστημα ,οι πράκτορες μας υλοποιήθηκαν με χρήση της JAVA

## **5.2 Περιγραφή αρχείων διαδικτυακής εφαρμογής**

### **5.2.1 Styling.css**

Το αρχείο αυτό καθορίζει την εμφάνιση των δομικών στοιχείων.Έτσι, για κάθε δομικό στοιχείο καθορίζεται το μέρος και ο τρόπος με τον οποίο θα εμφανιστεί στο δικτυακό τόπο.Ο λόγος ύπαρξης του συγκεκριμένου αρχείου είναι η αποφυγή μεγάλων τμημάτων κώδικα που σχετίζονται με τη μορφοποίηση.Έτσι,κάθε φορά που ακολουθείται το συγκεκριμένο στυλ μορφοποίησης αρκεί να εισάγεται το αρχείο αυτό στην αρχή του πηγαίου κώδικα.

### **5.2.2 Code.html**

Το αρχείο code προσφέρει την εμφάνιση της κεντρικής σελίδας της εφαρμογής,η οποία ενημερώνει το χρήστη για τον τρόπο με τον οποίο μπορεί να χρησιμοποιήσει τον ιστότοπο και του παρέχει τους αντίστοιχούς συνδέσμους. Συνδυάζει την HTML ,η οποία καθορίζει τα δομικά στοιχεία της σελίδας ,με το CSS αρχείο ,το οποίο ορίζει την εμφάνιση των δομικών στοιχείων.

### **5.2.3 Contact.jsp**

Η σελίδα αυτή εμφανίζει τις μεθόδους επικοινωνίας που παρέχονται για το χρήστη. Ακολουθεί τον τρόπο εμφάνισης που καθορίζεται από το css αρχείο και εμφανίζει καταλλήλως τα αντίστοιχα δομικά στοιχεία .

### **5.2.4 form\_company.jsp / form\_employee.jsp**

Η σελίδα αυτή εμφανίζει τη φόρμα όπου οι εταιρείες/εργαζόμενοι συμπληρώνουν τα στοιχεία τους και τις απαιτήσεις τους(οι εταιρείες).Τα δεδομένα που συγκεντρώνονται σε αυτό το σημείο στέλνονται με τρόπο που θα περιγραφεί στη συνέχεια σε άλλη ενότητα.Το όνομα της σελίδας αυτής είναι entryc.jsp / entry.jsp.

### **5.2.5 *entryc.jsp / entry.jsp***

Στη σελίδα αυτή πραγματοποιείται σύνδεση με τη βάση δεδομένων και τα δοθέντα στοιχεία αποθηκεύονται στον πίνακα *companies* της Β.Δ..Αντίστοιχα αποθηκεύονται τα στοιχεία των εργαζομένων στον πίνακα *employees* της Β.Δ..

### **5.2.6 *ok.html/ok1.html***

Η σελίδα αυτή εμφανίζεται έφόσον τα στοιχεία των εταιρειών/εργαζομένων έχουν προστεθεί επιτυχώς στη βάση.

### **5.2.7 *checkResults.jsp***

Η σελίδα αυτή υλοποιεί τη λειτουργία ελέγχου αποτελεσμάτων,όπως αυτή παρουσιάστηκε σε προηγούμενη ενότητα.Εμφανίζει μία φόρμα εισαγωγής κωδικού ονόματος και κωδικού εισαγωγής σύμφωνα πάλι με τη μορφοποίηση του αρχείου *css*.Αφού λάβει τα δεδομένα από το χρήστη,τα στέλνει σε άλλη σελίδα με τρόπο που θα εξηγήσουμε σε επόμενη ενότητα.Η σελίδα αυτή είναι η *check.jsp*.

### **5.2.8 *check.jsp***

Η σελίδα αυτή λαμβάνει τα δεδομένα και πραγματοποιεί σύνδεση με τη βάση.Μετά από κατάλληλους ελέγχους,εμφανίζει στο χρήστη την απόφαση που έλαβε το back-end πολυπρακτορικό σύστημα.

## **5.3 *Περιγραφή κλάσεων συστήματος απόφασης***

Όπως παρουσιάστηκε ήδη ,για τη λήψη απόφασης χρειαστήκαμε δύο τύπους πρακτόρων:

- *employeeAgent*
- *companyAgent*

Ο κάθε ένας θα αντιπροσώπευε τους εργαζομένους και τις εταιρείες αντίστοιχα στην πλατφόρμα απόφασης.

### **5.3.1 Η κλάση *employeeAgent***

Η κλάση αυτή επεκτείνει την κλάση *agent* που περιγράψαμε στο κεφάλαιο 3. Στα πλαίσια αυτής δημιουργήσαμε μεθόδους κατάλληλες για την υλοποίηση απαραίτητων λειτουργιών, που απαιτούνταν να εκτελεί ο πράκτορας που εκπροσωπεί τους εργαζομένους.

Στην ακόλουθη υποενότητα ακολουθεί η περιγραφή τους.

#### **5.3.1.1 Η μέθοδος *setup***

Με τη μέθοδο αυτή ο πράκτορας :

- Εισέρχεται στην πλατφόρμα εκτέλεσης και λαμβάνει ένα μοναδικό AID με το οποίο μπορεί να εντοπιστεί από άλλους πράκτορες στην πλατφόρμα
- Εγγράφεται στην πλατφόρμα μέσω του ειδικού πράκτορα DF.
- Προσθέτει 2 συμπεριφορές (*OfferRequestsServer()*, *PurchaseOrdersServer()*) που υλοποιούν την κύρια λειτουργία του

#### **5.3.1.2 Η μέθοδος *takedown***

Με τη μέθοδο αυτή ο πράκτορας απεγγράφεται από την υπηρεσία *yellow pages* που προσφέρει ο ειδικός πράκτορας DF.

#### **5.3.1.3 Η μέθοδος *user\_data***

Στη μέθοδο αυτή πραγματοποιείται σύνδεση με τη βάση δεδομένων και μετά από σχετικό έλεγχο εξάγεται το ID του εργαζομένου που αντιπροσωπεί ο *employeeAgent*, εφόσον αυτός είναι διαθέσιμος. Το τελευταίο σχετίζεται με το κατά πόσο ο συγκεκριμένος χρήστης έχει βρει ήδη εργασία ή με την πιθανότητα να διενεργείται ταυτόχρονα όμοιος έλεγχος. Σε αυτές τις δύο περιπτώσεις αναζητείται ο επόμενος εργαζόμενος που να είναι διαθέσιμος.

#### **5.3.1.4 Η μέθοδος *discard***

Με τη συγκεκριμένη μέθοδο ενημερώνεται η Β.Δ. σε περίπτωση που ο εργαζόμενος επιλέχθηκε από κάποιο πάροχο εργασίας.

#### **5.3.1.5** *Η μέθοδος local\_opt*

Στα πλαίσια του αλγορίθμου ταιριάσματος ,ο εργαζόμενος υπολογίζει το προσωπικό μέγιστο score που διαθέτει και προχωρά σε αποθήκευση αυτού στον πίνακα score στη Βάση Δεδομένων.Ο αλγόριθμος ταιριάσματος θα περιγραφεί λεπτομερώς στο επόμενο κεφάλαιο.

#### **5.3.1.6** *Η μέθοδος check\_ID*

Εδώ πραγματοποιείται έλεγχος για το αν έχει υπολογιστεί ήδη το προσωπικό score.Στην περίπτωση αυτή δεν υπολογίζεται ξανά.

#### **5.3.1.7** *Η μέθοδος busy*

Με τη μέθοδο αυτή πραγματοποιείται ενημέρωση συναφούς πεδίου στη βάση δεδομένων με σκοπό να υποδεικνύεται ότι ο συγκεκριμένος εργαζόμενος είναι απασχολημένος τη δεδομένη στιγμή λόγω ελέγχου των στοιχείων του.

#### **5.3.1.8** *Η συμπεριφορά OfferRequestsServer*

Η συμπεριφορά αυτή είναι κυκλική.Αυτό σημαίνει ότι εκτελείται συνεχώς,μέχρι τον τερματισμό του πράκτορα.Ο πράκτορας επικοινωνεί με χρήση μηνυμάτων acl ,μέσω της πλατφόρμας μηνυμάτων που έχει δημιουργηθεί με σκοπό την εύρεση εργασίας, με τον πράκτορα που εκπροσωπεί μια εταιρεία. Στο σημείο αυτό ο userAgent αποστέλλει στην εταιρεία το ID του εργαζομένου και η εταιρεία προχωρά στον αλγόριθμο ελέγχου ταιριάσματος κριτηρίων.

#### **5.3.1.9** *Η συμπεριφορά PurchaseOrdersServer*

Με τη χρήση αυτής της συμπεριφοράς ο πράκτορας του εργαζόμενου λαμβάνει το πιθανό θετικό μήνυμα από τον πράκτορα της εταιρείας και στέλνει μήνυμα επικύρωσης της συμφωνίας.

#### **5.3.1.10 Η μέθοδος `free_all_comp`**

Λόγω της κυκλικής συμπεριφοράς που εκτελεί ο `employeeAgent` απαιτείται κατά το τέλος κάθε κύκλου της να καθίστανται διαθέσιμες όλες οι εταιρείες.

Ο λόγος που συμβαίνει αυτό είναι γιατί ο `userAgent` υπόκειται έλεγχο από όλες τις εταιρείες που βρίσκονται στη Β.Δ.εφόσον αυτές δεν έχουν συμπληρώσει την θέση εργασίας που προσφέρουν(`used=1`). Κατά τον έλεγχο αυτό, για να μη γίνει ταυτόχρονη προσπέλαση στο κρίσιμο τμήμα (πλειάδες Β.Δ.), κάθε εταιρεία μέσω πεδίου(`used=-1`) εμφανίζεται ως μη διαθέσιμη.

Με το πέρας της διαδικασίας καθίστανται διαθέσιμες όλες οι εταιρείες έτσι ώστε να είναι δυνατή η εφαρμογή αλγορίθμου ταιριάσματος και με άλλους εργαζομένους που θα επισκεφτούν την πλατφόρμα επικοινωνίας.

### **5.3.2 Η κλάση `companyAgent`**

Ομοίως, η κλάση αυτή επεκτείνει την κλάση `agent` και μέσω μεθόδων και κυκλικών συμπεριφορών υλοποιεί τη λειτουργικότητα του πράκτορα που εκπροσωπεί τις εταιρείες στην πλατφόρμα εκτέλεσης.

#### **5.3.2.1 Η μέθοδος `setup`**

Ο πράκτορας της εταιρείας εισάγεται στο σύστημα και προσθέτει μια κυκλική συμπεριφορά(`TickerBehaviour`). Στα πλαίσια αυτής της συμπεριφοράς, η εταιρεία εντοπίζει περιοδικά τους εργαζομένους που βρίσκονται στην πλατφόρμα με θέμα την εύρεση εργασίας, κάνοντας χρήση της υπηρεσίας κίτρινων σελίδων που παρέχει ο `DF agent`.

#### **5.3.2.2 Η μέθοδος `takedown`**

Όπως περιγράφηκε και νωρίτερα, με τη μέθοδο αυτή ο πράκτορας τερματίζει τη λειτουργία του και εγκαταλείπει την πλατφόρμα εκτέλεσης.

### **5.3.2.3 Η μέθοδος *match***

Η παραπάνω υλοποιεί μερικώς τον αλγόριθμο ταιριάσματος των στοιχείων των εργαζομένων και των απαιτήσεων των εταιρειών. Ο λόγος που αναφερόμαστε σε αυτό ως μερική υλοποίηση είναι το γεγονός ότι αποτελεί το ένα από τα δύο σκέλη του αλγορίθμου ταιριάσματος που χρησιμοποιεί το πολυπρακτορικό μας σύστημα.

### **5.3.2.4 Η μέθοδος *max\_optimum***

Η παραπάνω μέθοδος προσπελαύνει τον πίνακα optimum της β.δ. και εντοπίζει για ποια εταιρεία έχει συγκεντρώσει ο εργαζόμενος το μεγαλύτερο score. Αυτό, σε συνδυασμό με τη μέθοδο local\_opt του εργαζομένου ολοκληρώνει τον αλγόριθμο ταιριάσματος.

### **5.3.2.5 Η μέθοδος *company\_data***

Στη μέθοδο αυτή πραγματοποιείται σύνδεση με τη βάση δεδομένων και μετά από σχετικό έλεγχο εξάγεται το ID της εταιρείας που αντιπροσωπεύει ο companyAgent ,εφόσον αυτός είναι διαθέσιμος. Το τελευταίο σχετίζεται με το κατά πόσο ο συγκεκριμένος χρήστης έχει βρει ήδη υπάλληλο ή με την πιθανότητα να διενεργείται ταυτόχρονα όμοιος έλεγχος. Σε αυτές τις δύο περιπτώσεις αναζητείται η επόμενη εταιρεία που είναι διαθέσιμη.

### **5.3.2.6 Η μέθοδος *count\_row***

Η παραπάνω σχετίζεται με τον έλεγχο που πραγματοποιείται στη βάση δεδομένων. Δεδομένης της κυκλικής συμπεριφοράς, κρίθηκε αναγκαίο κατά την ανάπτυξη του κώδικα η μέτρηση των πλειάδων της β.δ. έτσι ώστε να μην πραγματοποιούνται άσκοποι έλεγχοι σε αυτή.

### **5.3.2.7 Η μέθοδος *covered***

Μετά τη διαδικασία επικοινωνίας μεταξύ πρακτόρων, εφόσον βρεθεί κάποιο ταίριασμα ενημερώνονται οι πίνακες των εργαζομένων και των εταιρειών για το προκύπτον ταίριασμα. Το τελευταίο λειτουργεί βοηθητικά στη λειτουργία ελέγχου αποτελεσμάτων της διαδικτυακής εφαρμογής.

### 5.3.2.8 Η συμπεριφορά *RequestPerformer*

Η συμπεριφορά αυτή υλοποιεί ουσιαστικά τη λειτουργικότητα του companyAgent. Σε αυτό το σημείο ο companyAgent λαμβάνει μέσω της πλατφόρμας επικοινωνίας το ID του εργαζομένου, εκτελεί τον αλγόριθμο ταιριάσματος, στέλνει μήνυμα αποδοχής στον αντίστοιχο χρήστη και αναμένει αντίστοιχο μήνυμα αποδοχής. Όταν το λάβει αυτό κατοχυρώνει τη συμφωνία τους με σχετικό μήνυμα.

## 5.4 Βάση Δεδομένων

Το σχήμα Οντοτήτων-Συσχετίσεων (ER) που παρουσιάστηκε στο Κεφάλαιο 4 αποτελεί έναν εύχρηστο τρόπο περιγραφής μιας βάσης δεδομένων. Το ER διευκολύνει τη διαδικασία δημιουργίας του σχεσιακού μοντέλου της Β.Δ. και προσφέρει στον προγραμματιστή ευκολία στην οργάνωση του τύπου. Παρακάτω φαίνεται το σχεσιακό μοντέλο το οποίο προκύπτει από το διάγραμμα οντοτήτων-συσχετίσεων της βάσης του Earn your Living.

Employees:

ID	password	username	Firstname	Surname	Mail	EducationLevel	Grade	PostGrad	Experience	ForeignLang1	ForeignLang2	used	comp
----	----------	----------	-----------	---------	------	----------------	-------	----------	------------	--------------	--------------	------	------

Companies:

ID	username	password	Name	ELevel	Grado	Post_grado	Experiencia	Lang1	Lang2	acad_imp	grad_imp	post_imp	work_imp	eng_imp	fr_imp	used	user
----	----------	----------	------	--------	-------	------------	-------------	-------	-------	----------	----------	----------	----------	---------	--------	------	------

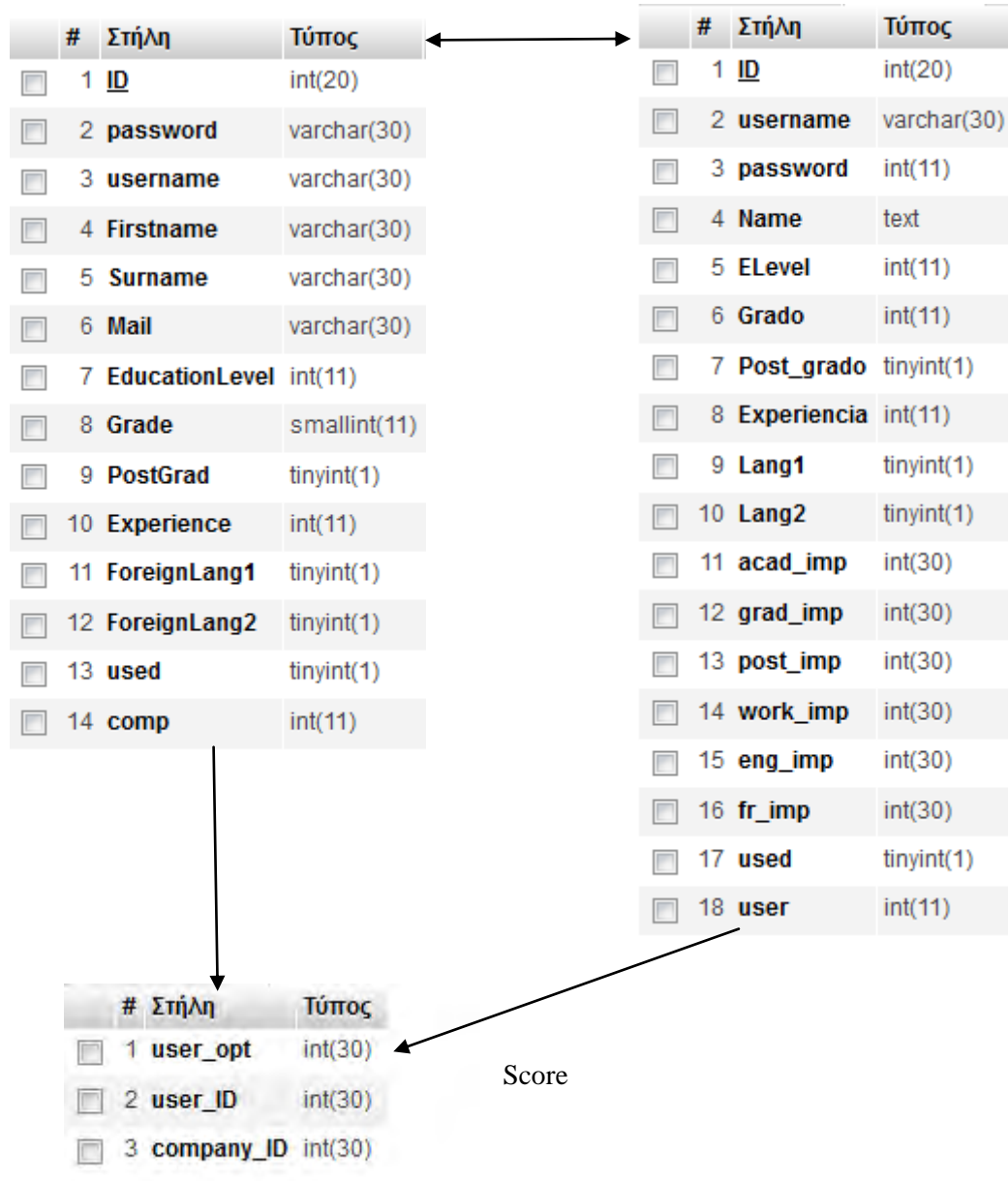
Score:

user_opt	user_ID	company_ID
----------	---------	------------

Και τέλος, το σχεσιακό μοντέλο της βάσης, συνδυασμένο με τους τύπους των γνωρισμάτων:

Employees:

Companies:



Σχήμα 6: Σχεσιακό Μοντέλο Βάσης



# 6

## *Υλοποίηση*

### *6.1 Λεπτομέρειες υλοποίησης*

Ιδιαίτερο ενδιαφέρον παρουσιάζει το αλγοριθμικό κομμάτι της εφαρμογής μας. Το τελευταίο σχετίζεται με τη λήψη απόφασης για το σύστημα. Έτσι, στη συγκεκριμένη ενότητα θα δώσουμε βάρος στον αλγόριθμο ταιριάσματος κριτηρίων που προαναφέρθηκε.

#### *6.1.1 Αλγόριθμος ταιριάσματος*

Σκοπός του συστήματός μας είναι η επιλογή του αιτούντος εργασία που συγκεντρώνει τα περισσότερα στοιχεία. Ο αλγόριθμος εκμεταλλεύεται την πληροφορία που έχουν ήδη δώσει οι χρήστες στο σύστημα και διαχωρίζεται σε δύο σκέλη.

Σε πρώτη φάση το σύστημα ελέγχει εάν ο εργαζόμενος πληροί κατά το ελάχιστο τα κριτήρια που έχουν δοθεί. Αν αυτός τα πληροί και εφόσον πραγματοποιηθεί έλεγχος μπορεί να διεκδικήσει τη δοθείσα θέση εργασίας.

Σε περίπτωση όμως που αυτά δεν πληρώνται, η εταιρεία αναγκάζεται να μειώσει τις απαιτήσεις τις και ενεργοποιείται η διαδικασία επιλογής βάσει score.

Πιο συγκεκριμένα,

1. Ο employeeAgent εκδηλώνει το ενδιαφέρον του και στέλνει το ID του.
2. Παράλληλα υπολογίζει το score του,σε περίπτωση που αυτό χρειασθεί.

Σε αυτό το σημείο λαμβάνονται υπόψη τα βάρη σημαντικότητας που έχουν δώσει οι χρήστες-εταιρείες. Υπολογίζεται το score ως ένας βεβαρυμένος μέσος όρος, βάσει του τύπου:

$$\text{Score} = \frac{\sum w * \text{crit}}{\sum w},$$

Όπου w το βάρος του συγκεκριμένου κριτηρίου και crit το συγκεκριμένο κριτήριο.Επιπλέον,υπολογίζεται το μέγιστο score και για ποια εταιρεία αυτό σημειώνεται.

3. Ο companyAgent λαμβάνει το ID και καλεί τη μέθοδο match με ορίσματα το ID του εργαζομένου και της εταιρείας που εκπροσωπεί.
4. Πραγματοποιείται σύνδεση με τη βάση δεδομένων και με απλό έλεγχο βλέπουμε αν πληροί τα κριτήρια.Το τελευταίο γίνεται με σύγκριση τιμών πεδίων εργαζομένου/εταιρείας. Σε περίπτωση που οι τιμές των πεδίων του εργαζομένου είναι ίσες ή υπερβαίνουν τις τιμές των κριτηρίων της εταιρείας,τα κριτήρια πληρούνται.Πηγαίνουμε στο βήμα 7.
5. Αν δεν τα πληροί συνεχίζεται ο έλεγχος από την επόμενη διαθέσιμη εταιρεία.Επιστροφή στο βήμα 1 .
6. Αν έχει υπολογιστεί το max score για όλους τους εργαζομένους ,γίνεται κλήση της max\_optimum.
7. Τέλος,ορίζεται ποιος εργαζόμενος έχει επιλεχθεί για τη θέση εργασίας

### 6.1.2 Μηνύματα μεταξύ *userAgent-companyAgent*

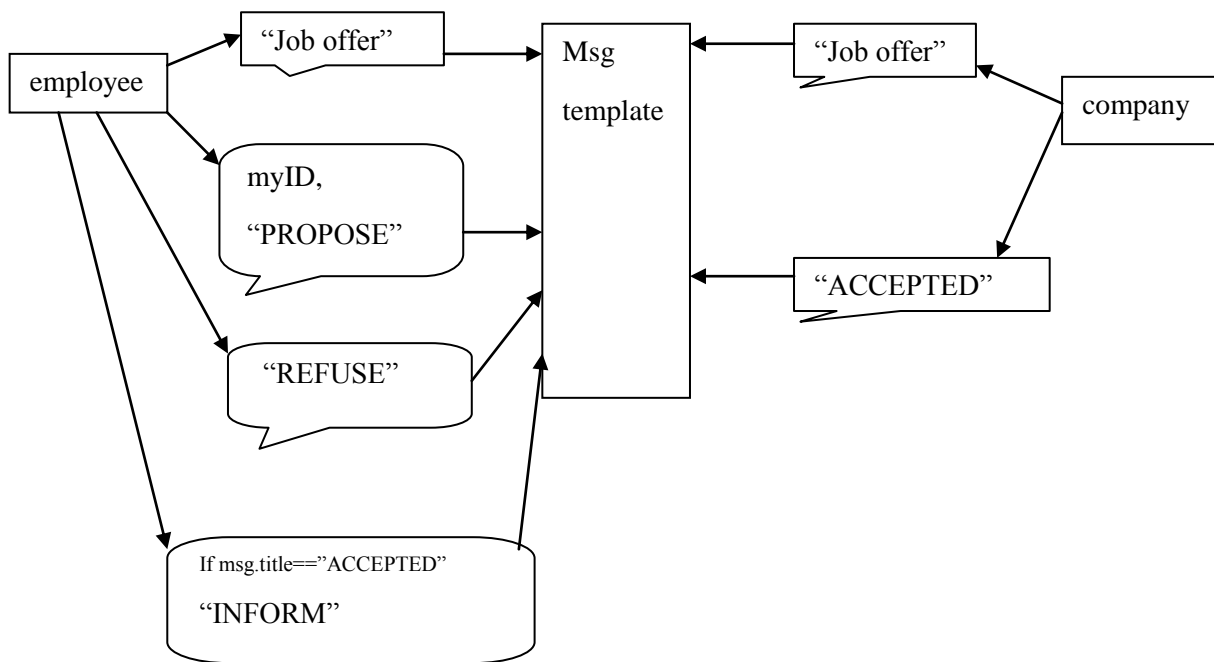
Παρακάτω βλέπουμε τα μηνύματα που ανταλλάσσουν οι εταιρείες και οι εργαζόμενοι.

Εργαζόμενοι:

- **“Job offer”**: αποτελεί το conversation ID. Με τον τρόπο αυτό οι εταιρείες και οι εργαζόμενοι γνωρίζουν ότι μιλούν για το ίδιο πράγμα
- **myID, “PROPOSE”**: Ο εργαζόμενος εκδηλώνει το ενδιαφέρον του και μαζί με αυτό αποστέλλει το ID του για να διευκολύνει το ταίριασμα κριτηρίων
- **“REFUSE”**: σε περίπτωση που δεν υπάρχει διαθέσιμος εργαζόμενος
- **“INFORM”**: για αποδοχή πρότασης από την εταιρεία

Εταιρείες:

- **“Job offer”**: ως conversation Id
- **“ACCEPTED”**: σε περίπτωση που μετά από έλεγχο επιλεγεί ο συγκεκριμένος εργαζόμενος



Σχήμα 7 :Ανταλλαγή μηνυμάτων employee-company

## **6.2 Πλατφόρμες και προγραμματιστικά εργαλεία**

### **6.2.1 Ενσωματωμένο περιβάλλον ανάπτυξης(IDE) Netbeans**

Για την υλοποίηση της εφαρμογής μας χρησιμοποιήθηκε το περιβάλλον ανάπτυξης Netbeans 7.0.1. Ο λόγος που επιλέχθηκε το συγκεκριμένο περιβάλλον είναι η ευχρηστία του, καθώς και η προηγούμενη εμπειρία χρήσης του.

### **6.2.2 Πλατφόρμα JADE**

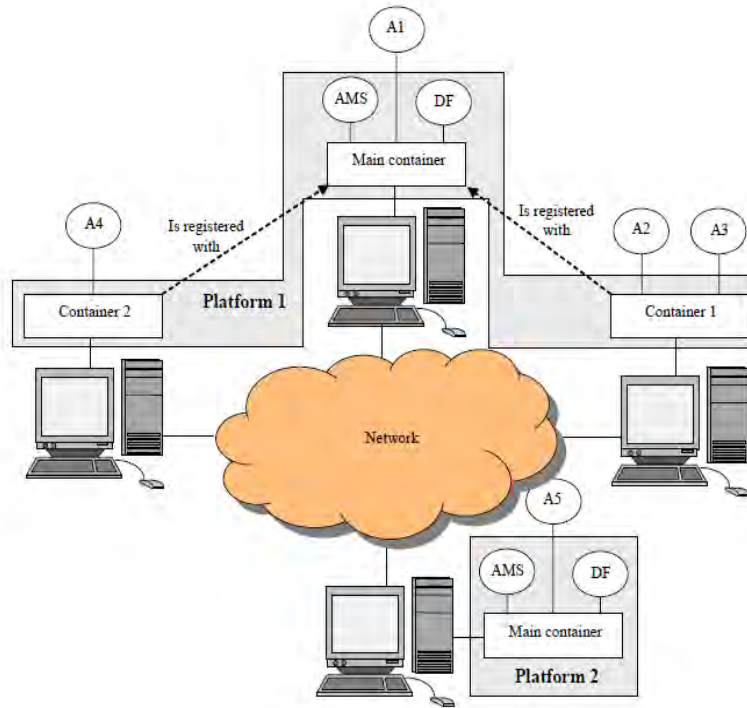
#### **6.2.2.1 Κύρια μέρη**

Η JADE είναι μια πλατφόρμα εκτέλεσης (middleware) ελεύθερου λογισμικού, που διευκολύνει την ανάπτυξη πολυπρακτορικών συστημάτων. Περιλαμβάνει:

- Περιβάλλον εκτέλεσης(runtime environment), όπου οι πράκτορες ζουν.
- Βιβλιοθήκη Κλάσεων που απαιτούνται για την ανάπτυξη των πρακτόρων.
- Γραφικό περιβάλλον(gui) που επιτρέπει τη διαχείριση και επίβλεψη της λειτουργίας των πρακτόρων.

#### **6.2.2.2 Containers και platforms**

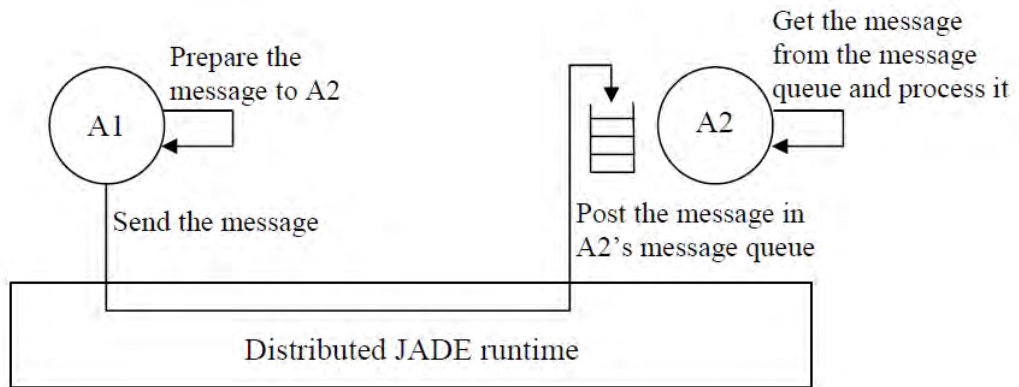
Κάθε στιγμιότυπο "τρεξίματος" του περιβάλλοντος εκτέλεσης JADE λέγεται *container*, καθώς μπορεί να περιέχει πολλούς πράκτορες. Το σύνολο των ενεργών πρακτόρων λέγεται *platform*. Απαιτείται πάντα να υπάρχει ένας *main Container*, όπου όλοι οι άλλοι *containers* πρέπει να εγγραφούν σε αυτόν για να χρησιμοποιήσουν την πλατφόρμα. Αν δημιουργηθεί κάποιος άλλος *main Container*, αμέσως συνιστά ένα άλλο *platform*. Όπως βλέπουμε στο παρακάτω σχήμα, απαιτούνται και δύο ειδικοί πράκτορες, ο AMS και ο DF με τους ρόλους που περιγράφηκαν νωρίτερα.



Εικ.15: Containers and Platforms

### 6.2.2.3 Επικοινωνία μεταξύ πρακτόρων στην πλατφόρμα JADE

Ένα από τα σημαντικότερα χαρακτηριστικά που παρέχει η πλατφόρμα εκτέλεσης JADE είναι η επικοινωνία μεταξύ πρακτόρων. Ο τρόπος επικοινωνίας είναι η ασύγχρονη ανταλλαγή μηνυμάτων (asynchronous message passing). Κάθε πράκτορας έχει μια ουρά από μηνύματα (agent message queue), όπου το περιβάλλον εκτέλεσης της JADE τοποθετεί τα μηνύματα που στέλνουν σε αυτόν άλλοι πράκτορες. Κάθε φορά που ένα μήνυμα τοποθετείται στην ουρά μηνυμάτων, ο πράκτορας ειδοποιείται. Όταν ο πράκτορας λάβει το μήνυμα, ο προγραμματιστής καθορίζει το χειρισμό του.



Εικ. 16: Ασύγχρονη επικοινωνία στην πλατφόρμα JADE.

#### 6.2.2.4 Το gui της JADE και τρόποι ενεργοποίησης πρακτόρων

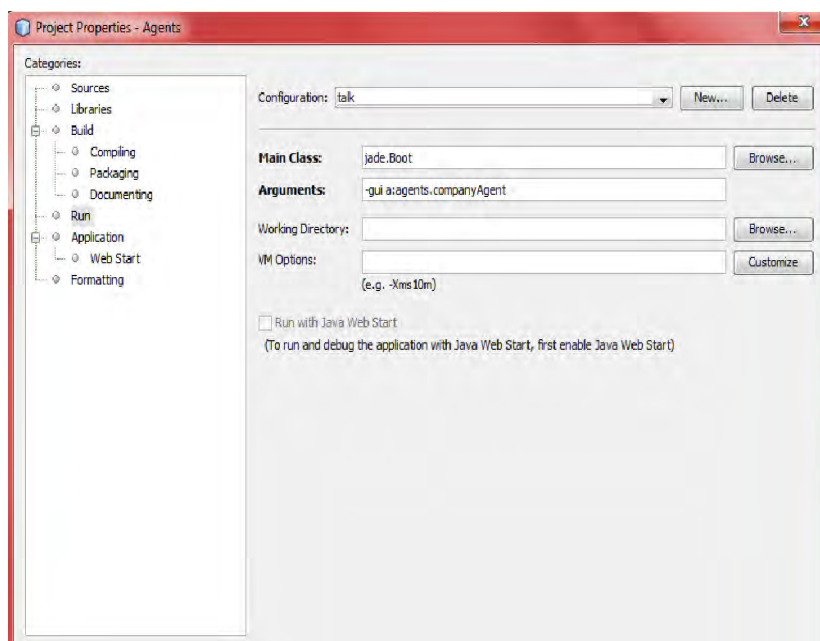
Αφού ολοκληρωθεί η διαδικασία ορισμού της λειτουργικότητας κάθε πράκτορα, αυτοί πρέπει να ενεργοποιηθούν με σκοπό να επιτελέσουν το έργο τους.

Με χρήση του Netbeans, και συγκεκριμένα με επιλογή Run -> Set Project Configuration ορίζουμε σαν main Class jade.Boot από τη βιβλιοθήκη της JADE (όπως αναφέρουμε παρακάτω). Το τελευταίο θα επιτρέψει την εκκίνηση της πλατφόρμας JADE.

Συνεχίζοντας, στο πεδίο Arguments πληκτρολογούμε την εντολή:

*-gui a: agents.companyAgent*

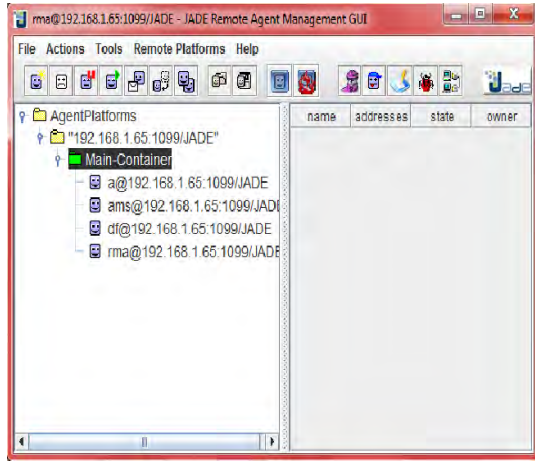
όπως φαίνεται και στο παράθυρο:



Εικ. 17 Ενεργοποίηση Πράκτορα

Με τον τρόπο αυτό εκκινούμε τον πράκτορα companyAgent του πακέτου agents και τον ονομάζουμε a. Με το όνομα αυτό θα εγγραφεί στην πλατφόρμα επικοινωνίας.

Παράλληλα εμφανίζεται και το γραφικό περιβάλλον διεπαφής χρήστη(gui), που δείχνει τις υπάρχουσες containers και platforms.



Εικ.18:Γραφικό περιβάλλον JADE(1)

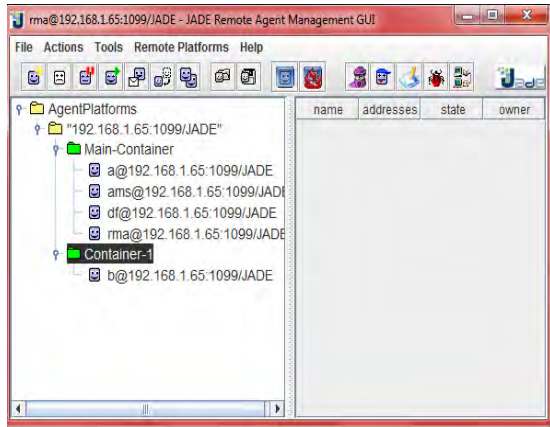
Παρατηρούμε στο σημείο αυτό ότι ο πράκτορας a που δημιουργήσαμε είναι μέλος του Main-Container.

Συνεχίζοντας,για να ενεργοποιήσουμε και τον πράκτορα των εργαζομένων ομοίως πληκτρολογούμε στο πεδίο Arguments το ακόλουθο:

*-container b: agents.employeeAgent*

Το παραπάνω έχει επιφέρει τις ακόλουθες αλλαγές στο gui της JADE:





Εικ.19: Γραφικό Περιβάλλον JADE(2)

Βλέπουμε λοιπόν ότι πλέον ο πράκτορας b έχει γίνει μέλος ενός container και η επικοινωνία μέσω της πλατφόρμας μπορεί να ξεκινήσει.

### 6.2.3 Wamp Server

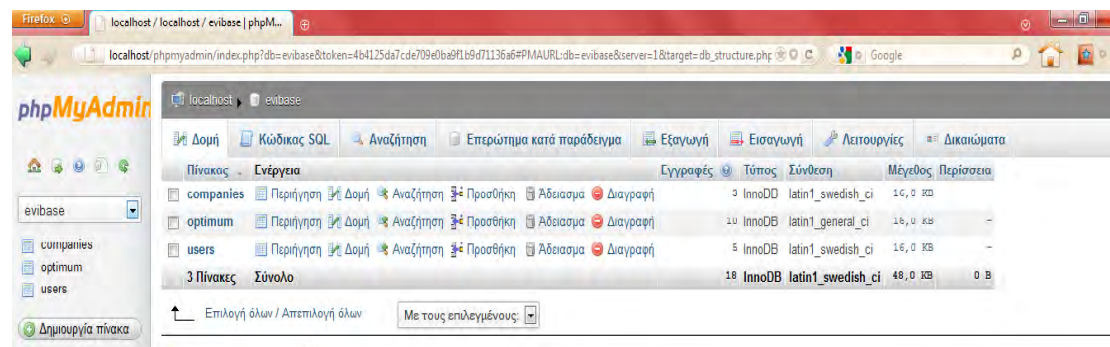
Απαραίτητο για την ανάπτυξη του εφαρμογής μας ήταν η χρησιμοποίηση των ακόλουθων server:

- Apache Tomcat 7.0.14.0
- Glassfish server 3.1
- MySQL Server

Τα παραπάνω προσφέρονται από το περιβάλλον ανάπτυξης διαδικτυακών εφαρμογών WampServer. Ο wampserver εγκαταστάθηκε σε περιβάλλον windows7, όπου και αναπτύχθηκε όλη η εφαρμογή. Ο λόγος που επιλέχθηκε το συγκεκριμένο περιβάλλον ανάπτυξης είναι το γεγονός ότι παρείχε συγκεντρωτικά τους απαιτούμενους servers αλλά και το γεγονός ότι προσφέρει και μία πλατφόρμα διαχείρισης της βάσης δεδομένων, το phpmyadmin.

Το phrmyadmin επιτρέπει τη συνεχή επίβλεψη των πινάκων της βάσης δεδομένων και επιτρέπει προσθήκη,αλλαγή και διαγραφή πλειάδων.Επιπλέον,παρέχει εργαλεία που επιτρέπουν το αρχικό στήσιμο της βάσης,με καθορισμό τύπου μεταβλητών και εύρους τιμών.

Ένα στιγμιότυπο του εργαλείου διαχείρισης φαίνεται ακολούθως:



Εικ.20:phrmyadmin

#### 6.2.4 Απαιτήσεις για την εγκατάσταση διπλωματικής

Αρχικά εγκαταστήσαμε το περιβάλλον ανάπτυξης wampserver , καθώς αυτό πρόσφερε πολλές διευκολύνσεις για τη δημιουργία και κατ'επέκταση τη διαχείριση της βάσης δεδομένων. Στη συνέχεια ,δεδομένης της συγγραφής κώδικα με χρήση του Netbeans ,κρίθηκε αναγκαία η σύνδεση της βάσης δεδομένων με το εργαλείο αυτό. Για το σκοπό αυτό εγκαταστήσαμε τα ακόλουθα:

- Jdk 1.7
  - Περιλαμβάνει εργαλεία για ανάπτυξη και έλεγχο προγραμμάτων γραμμένα σε JAVA.
- MySQL(Connector/J driver)
- JDBC (**J**ava **D**atabase **C**onnectivity) driver
  - Το JDBC API(**a**pplication **I**nterface)είναι ένα σύνολο κλάσεων για την ανάπτυξη εφαρμογών βάσεων δεδομένων client-server με χρήση της JAVA.Το λογισμικό client/server συνδέει το χρήστη με ένα παροχέα

πληροφοριών(DBMS) και για τη σύνδεση αυτή χρησιμοποιείται ο JDBC driver.

- JDBC-ODBC bridge
  - Αυτό το πρόγραμμα οδήγησης συνδέει το JDBC με ένα πρότυπο συνδεσιμότητας ΒΔ,το ODBC
- Το δομημένο πακέτο web εφαρμογών Java EE 5
- Τέλος, για τη χρήση πρακτόρων JADE απαιτήθηκε η βιβλιοθήκη της JADE.

# 7

## *Επίλογος*

### *7.1 Σύνοψη και συμπεράσματα*

Το **Earn Your Living** αποτέλεσε μια προσπάθεια για τη συγκέντρωση πληροφορίας που σχετίζεται με αιτούντες και προσφέροντες εργασία. Κατ' επέκταση ιδιαίτερη σημασία δόθηκε στην αλληλεπίδραση αυτών, με απώτερο σκοπό τη διευκόλυνση της διαδικασίας επιλογής μελλοντικών εργαζομένων-μελών μιας εταιρείας. Η αλληλεπίδραση αυτή στηρίχθηκε σε μία διαδικτυακή διεπαφή και ένα πολυπρακτορικό περιβάλλον, όπως έχει περιγραφεί.

### *7.2 Μελλοντικές επεκτάσεις*

Η εφαρμογή μας δημιουργήθηκε κυρίως για τον κλάδο των μηχανικών Η/Υ, δεδομένου ότι εφόσον αναπτυχθεί για ένα τομέα εύκολα μπορούν να καλυφθούν και διαφορετικοί τομείς. Έτσι, λόγω του ενδιαφέροντος που παρουσιάζει η άμεση εφαρμογή του στην καθημερινότητα του επαγγελματικού κόσμου, είναι προφανές ότι επιδέχεται επέκτασης και αλλαγών.

Πιο συγκεκριμένα, ακολουθούν κάποιες ιδέες.

- Επέκταση της εφαρμογής του σε άλλους επαγγελματικούς κλάδους.
- Βελτιστοποίηση του υπάρχοντος αλγορίθμου με σκοπό τη γρηγορότερη απόκριση του συστήματος.

- Βελτίωση του τρόπου εμφάνισης των σελίδων έτσι ώστε να γίνεται πιο ελκυστικό στο χρήστη.
- Εμπλουτισμός της βάσης δεδομένων με περισσότερα στοιχεία χρηστών έτσι ώστε να λαμβάνονται περισσότερα στοιχεία υπόψη στον αλγόριθμο λήψης απόφασης.

# 8

## *Βιβλιογραφία*

- [1] [http://en.wikipedia.org/wiki/Decision\\_support\\_system](http://en.wikipedia.org/wiki/Decision_support_system)
  
- [2] <http://www.w3schools.com>
- [3] Fabio Bellifemine<sup>1</sup>, Agostino Poggi<sup>2</sup>, and Giovanni Rimassa<sup>2</sup>  
*Developing Multi-agent Systems with JADE*
- [4] Giovanni Caire (TILAB, formerly CSELT)  
*JADE TUTORIAL*  
*JADE PROGRAMMING FOR BEGINNERS*
- [5] <http://jade.tilab.com/doc/api/jade/core/Agent.html>
  
- [6] <http://www.iro.umontreal.ca/~vaucher/Agents/Jade/primer4.html>
  
- [7] Fabio Bellifemine, Giovanni Caire, Tiziana Trucco (TILAB S.p.A., formerly CSELT)  
Giovanni Rimassa (FRAMETech s.r.l.), Roland Mungenast (PROFACTOR GmbH).  
*JADE ADMINISTRATOR'S GUIDE*
- [8] <http://www.wampserver.com/en/>