

Καρέλης Μάριος AM 1067399

Λιούμη Κυριακή AM 1067410

Αναφορά Project

Εργαστήριο Βάσεων Δεδομένων

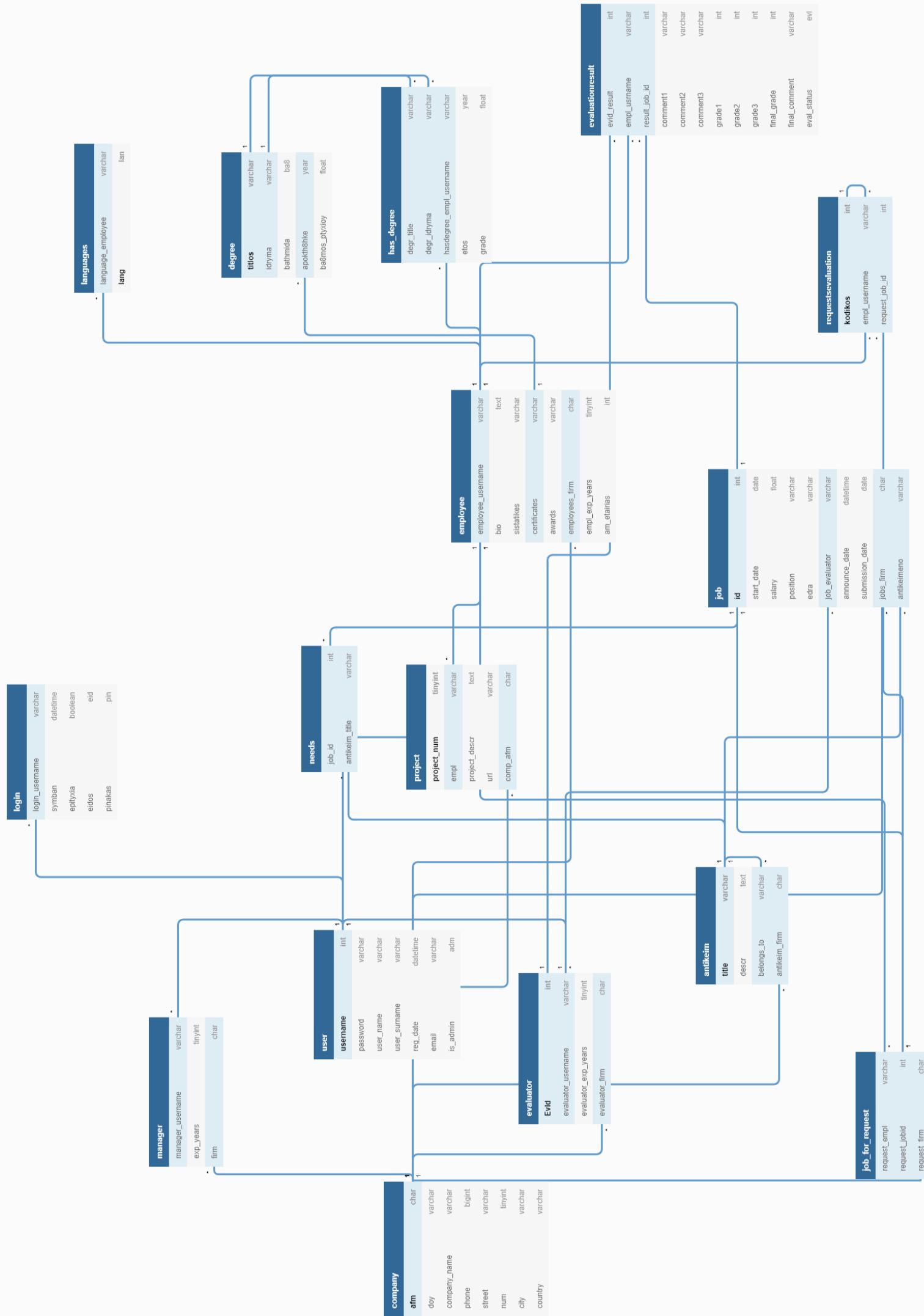
Περιεχόμενα :

- **A Μέρος**

- Σχεσιακό Διάγραμμα Σελ 3-4
- Create-Insert Σελ 5-15
- Procedures Σελ 16-18
- Triggers Σελ 19-25
- Έξτρα Procedure Σελ 26
- Έξτρα Triggers Σελ 27

- **B Μέρος**

- Manager..... Σελ 28-35
- Evaluator..... Σελ 36-40
- Employee..... Σελ 41-45
- Admin..... Σελ 46-53
- Β μέρος GUI (login table)..... Σελ 54-55
- Γενικά Σχόλια..... Σελ 56



A ΜΕΡΟΣ

Σχεσιακό Διάγραμμα :

Το βλέπουμε παραπάνω. Για την δημιουργία του χρησιμοποιήσαμε την ιστοσελίδα :

<https://dbdiagram.io/home>

Παραθέτουμε το σχεσιακό διάγραμμα και με τον συγκεκριμένο σύνδεσμο για ευκολία στην ανάγνωση:

<https://dbdiagram.io/d/60ef16124ed9be1c05ccdfda>

Οι εξαρτήσεις θα φανούν καλύτερα στον παραπάνω σύνδεσμο.

Όπου :

Enum adm {yes no}

Enum lan {EN FR SP GR}

Enum ba8 {LYKEIO UNIV MASTER PHD}

Enum evl {Pending Finished}

Enum eid {insert update delete}

Enum pin {job employee requestsevaluation}

Οι εξηγήσεις για τα πεδία και τους πίνακες που προστέθηκαν, βρίσκονται παρακάτω μαζί με το create και insert.

Create – Insert

(Ο σχολιασμός είναι με μπλε χρώμα)

Προσθέσαμε το πεδίο is_admin, το οποίο μας δείχνει αν ο χρήστης είναι administrator ή όχι.

```
CREATE TABLE user(
    username VARCHAR(12) NOT NULL,
    password VARCHAR(10) NOT NULL,
    user_name VARCHAR(25),
    user_surname VARCHAR(35),
    reg_date DATETIME,
    email VARCHAR(30),
    is_admin ENUM('yes','no'),
    PRIMARY KEY (username)
);

INSERT INTO user(username,password,user_name,user_surname,reg_date,email,is_admin) VALUES
("marmar","12345","marios","karelhs","2019-02-15 13:05:09","marioskar@gmail.com",'yes'),
("kyrlm","1234","kyriaki","lioumi","2020-12-19 12:40:00","kiriaki123@gmail.com",'yes'),
("kostasco","123456","kostas","papanikolaoy","2009-05-14 03:45:32","kostasco@gmail.com",'no'),
("marlp","12345678","maria","lamproy","2019-07-02 05:00:23","marialp@gmail.com",'no'),
("nikoskos","1234567890","nikos","giannopoulos","1996-12-24 14:32:54","nikoskos@gmail.com",'no'),
("petros123","123","petros","xristopoylos","2015-03-30 17:21:01","petros123@gmail.com",'no'),
("panosko","434343","panayoths","kostopoylos","1996-06-05 09:43:32","panosko@gmail.com",'no'),
("marios123","123456","marios","papanikos","2009-05-14 03:45:32","marios123@gmail.com",'no'),
("nionios","12345678","nionios","christodoulos","2019-07-02 05:00:23","nionios@gmail.com",'no'),
("johnny","1234567890","giannhs","giannopoulos","1996-12-24 14:32:54","johnny@gmail.com",'no'),
("jim12","123","dimitris","karamanos","2015-03-30 17:21:01","jim12@gmail.com",'no'),
("tommy","434343","thomas","petropoulos","1996-06-05 09:43:32","tommy@gmail.com",'no'),
("nefelaki","54321","nefelh","iatroy","2020-10-17 14:00:00","nefeliat@gmail.com",'no'),
("fotisre","2222","fotis","oikonomoy","2021-01-06 13:13:13","fotisre@gmail.com",'no'),
("euridik","434343","euridikh","petropoulou","1996-06-05 09:45:32","euridik@gmail.com",'no'),
("papad","54321","nikos","papadakis","2020-10-17 14:00:00","papad@gmail.com",'no'),
("takis12","2222","takis","loretos","2021-01-06 13:13:13","takis12@gmail.com",'no');
```

```
CREATE TABLE company(
    afm CHAR(9) NOT NULL,
    doy VARCHAR(15),
    company_name VARCHAR(35),
    phone BIGINT(16),
    street VARCHAR(15),
    num TINYINT(4),
    city VARCHAR(15),
    country VARCHAR(15),
    PRIMARY KEY (afm)
);
```

```
INSERT INTO company(afm,doy,company_name,phone,street,num,city,country) VALUES
("123456789","A Geraka","Kyrcomp",2610547698,"orfeos",10,"athena","ellada"),
("234567890","C Patras","Ucomp",2310482412,"hfaistoy",15,"patra","ellada"),
("345678909","B Patras", "UnitedFG",2299076542,"kanakari",127,"patra","ellada"),
("121234345","A Patras", "Patrascomp",2106054673,"norman",45,"patra","ellada"),
("343445456","D Patras", "Pixel",231056896,"kolokotroni",6,"patra","ellada");
```

Προστέθηκαν τα πεδία: employees_firm, empl_exp_years, am_etairias

Το πεδίο employees_firm μας δείχνει σε ποια εταιρία ανήκει ο εργαζόμενος. Είναι ξένο κλειδί, το οποίο παίρνουμε από το table company.

Το πεδίο empl_exp_years μας δείχνει τα χρόνια εμπειρίας του εργαζομένου.

Το πεδίο am_etairias μας δείχνει τον AM του εργαζομένου στην εταιρία του, τον οποίο κάναμε να αυξάνεται αυτόματα με το auto_increment. Ακόμα, το ορίσαμε unique, καθώς δεν πρέπει να υπάρχουν 2 εργαζόμενοι με το ίδιο AM.

```
CREATE TABLE employee(
    employee_username VARCHAR(12) NOT NULL,
    bio TEXT,
    sistatikes VARCHAR(35),
    certificates VARCHAR(35),
    awards VARCHAR(35),
    employees_firm CHAR(9),
    empl_exp_years TINYINT(2),
    am_etairias INT(4) AUTO_INCREMENT NOT NULL,
    UNIQUE(am_etairias),
    CONSTRAINT employees_name FOREIGN KEY (employee_username) REFERENCES user(username)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT emplofirm FOREIGN KEY (employees_firm) REFERENCES company(afm)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO
employee(employee_username,bio,sistatikes,certificates,awards,employees_firm,empl_exp_years)VALUES
("marios123","apofoitos mhxanologwn",NULL,NULL,NULL,"123456789",6),
("nionios","apofoitos mhxanikwn h/y kai plhroforikhs",NULL,"proficiensy, C2 in France",NULL,"234567890",3),
("johnny","apofoitos iatrikhs",NULL,NULL,NULL,"345678909",6),
("jim12","parakoloy8hsh seminarion, foititria hlekstrologon",NULL,"proficiensy, C2 in
France",NULL,"121234345",3),
("tommy","apofoitos hlekstrologwn, metaptuxiako sta diktua",NULL,"apofoitos iatrikhs",NULL,"343445456",4);
```

```
CREATE TABLE manager(
manager_username VARCHAR(12) NOT NULL,
exp_years TINYINT(4),
firm CHAR(9),
CONSTRAINT managers_name FOREIGN KEY (manager_username) REFERENCES user(username)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT manager_firm FOREIGN KEY (firm) REFERENCES company(afm)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO manager(manager_username,exp_years,firm)VALUES
("nefelaki",5,"123456789"),
("fotisre",2,"234567890"),
("euridik",1,"345678909"),
("papad",2,"121234345"),
("takis12",3,"343445456");
```

Προσθέσαμε το πεδίο antikeim_firm, έτσι ώστε να ξέρουμε το αντικείμενο σε ποια εταιρία ανήκει. Είναι ξένο κλειδί, το οποίο παίρνουμε από το table company.

```
CREATE TABLE antikeim(
title VARCHAR(36) NOT NULL,
descr TEXT,
belongs_to VARCHAR(36),
antikeim_firm CHAR(9),
PRIMARY KEY (title),
CONSTRAINT antike FOREIGN KEY (belongs_to) REFERENCES antikeim(title)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT firmaaa FOREIGN KEY (antikeim_firm) REFERENCES company(afm)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO antikeim(title,descr,belongs_to,antikeim_firm)VALUES  
("marketing","gia proo8hsh proionton",NULL,"123456789"),  
("iatrikh","gia thn ygeia ton ergazomenon",NULL,"234567890"),  
("oikonomika",NULL,NULL,"345678909"),  
("dhmosies sxeseis",NULL,NULL,"121234345"),  
("ypologistes",NULL,NULL,"343445456"),  
("hlektriko reyma","sosth leitoyrgia etairias",NULL,"121234345"),  
("gramateia","e3yphrethsh pelaton",NULL,"123456789");
```

Προσθέσαμε το πεδίο EvId, το οποίο είναι πρωτεύων κλειδί και αντιπροσωπεύει τον μοναδικό κωδικό που έχει ο κάθε αξιολογητής.

```
CREATE TABLE evaluator(  
evaluator_username VARCHAR(12) NOT NULL,  
evaluator_exp_years TINYINT(4),  
evaluator_firm CHAR(9),  
EvId INT(4),  
PRIMARY KEY (EvId),  
CONSTRAINT evaluator_name FOREIGN KEY (evaluator_username) REFERENCES user(username)  
ON DELETE CASCADE ON UPDATE CASCADE,  
CONSTRAINT firm_of_evaluator FOREIGN KEY (evaluator_firm) REFERENCES company(afm)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
INSERT INTO evaluator(evaluator_username,evaluator_exp_years,evaluator_firm,EvId)VALUES  
("kostasco",3,"123456789",1),  
("marlp",2,"234567890",2),  
("nikoskos",1,"345678909",3),  
("petros123",1,"121234345",4),  
("panosko",4,"343445456",5);
```

Προσθέσαμε τα πεδία jobs_firm και antikeimeno.

Το πεδίο jobs_firm μας δηλώνει σε ποια εταιρία ανήκει η συγκεκριμένη δουλειά. Είναι ξένο κλειδί και το παίρνουμε από το table company.

Το πεδίο antikeimeno μας δείχνει ποιο αντικείμενο χρειάζεται να γνωρίζει κάποιος εργαζόμενος για τη συγκεκριμένη δουλειά. Είναι ξένο κλειδί και το παίρνουμε από το table antikeim.

```
CREATE TABLE job(
    id INT(4) NOT NULL,
    start_date DATE,
    salary FLOAT(6,1),
    position VARCHAR(40),
    edra VARCHAR(45),
    job_evaluator VARCHAR(12),
    announce_date DATETIME,
    submission_date DATE,
    jobs_firm CHAR(9),
    antikeimeno VARCHAR(36),
    PRIMARY KEY (id),
    CONSTRAINT evaluatorname FOREIGN KEY (job_evaluator) REFERENCES evaluator(evaluator_username)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT jobsfirm FOREIGN KEY (jobs_firm) REFERENCES company(afm)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT antik FOREIGN KEY (antikeimeno) REFERENCES antikeim(title)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO
    job(id,start_date,salary,position,edra,job_evaluator,announce_date,submission_date,jobs_firm,antikeimeno
)VALUES
    (1234,"03-07-21",1350.8,"tamias","patra","kostasco","03-05-21","30-06-21","123456789","oikonomika"),
    (2345,"20-01-21",2456.7,"hlekstrologos","athena","marlp","29-12-20","15-01-21","234567890","hlektriko
    reyma"),
    (4567,"24-01-21",1765.4,"giatros","patra","nikoskos","01-12-20","13-01-21","345678909","iatrkh"),
    (6789,"23-4-21",3433.0,"dhmosies sxeseis","athena","petros123","15-1-21","22-4-21","121234345","dhmosies
    sxeseis"),
    (7890,"06-01-21",4234.7,"marketing","patra","panosko","12-12-20","05-01-21","343445456","marketing"),
    (7654,"20-01-21",3333.2,"mhxnikos","athena","marlp","01-01-21","19-01-21","234567890","ypolistes"),
    (5432,"30-01-21",7890.5,"boh8os","patra","kostasco","01-01-21","29-01-21","123456789","gramateia");
```

```

CREATE TABLE needs(
job_id INT(4) NOT NULL,
antikeim_title VARCHAR(36),
CONSTRAINT id_of_job FOREIGN KEY (job_id) REFERENCES job(id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT titlosAntikeimenou FOREIGN KEY (antikeim_title) REFERENCES antikeim(title)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

INSERT INTO needs(job_id,antikeim_title)VALUES
(1234,"oikonomika"),
(2345,"hlektriko reyma"),
(4567,"iatrikh"),
(6789,"dhmosies sxeseis"),
(7890,"marketing"),
(7654,"ypologistes"),
(5432,"gramateia");

```

Προσθέσαμε το πεδίο comp_afm, το οποίο μας δείχνει σε ποια εταιρία ανήκει το συγκεκριμένο project. Είναι ξένο κλειδί, παίρνουμε το afm από το table company.

```

CREATE TABLE project(
project_num TINYINT(4) NOT NULL AUTO_INCREMENT,
empl VARCHAR(12) NOT NULL,
project_descr TEXT,
url VARCHAR(60),
comp_afm CHAR(9),
PRIMARY KEY (project_num),
CONSTRAINT project_name FOREIGN KEY (empl) REFERENCES employee(employee_username)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT compafmm FOREIGN KEY (comp_afm) REFERENCES company(afm)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

INSERT INTO project(empl,project_descr,url,comp_afm)VALUES
("marios123","dhmioyrgia paixnidioy gia thn apasxolhsh sta dialleimata","https://paixnidi.com","121234345"),
("nionios","kataskeyh neoy programatos gia ton programmatismo ton meetings ths
etairias","https://programm.gr","123456789");

```

```

CREATE TABLE languages(
language_employee VARCHAR(12) NOT NULL,
lang SET('EN','FR','SP','GR') NOT NULL,
PRIMARY KEY (lang),
CONSTRAINT lang_employee_name FOREIGN KEY (language_employee) REFERENCES
employee(employee_username)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

INSERT INTO languages(language_employee,lang)VALUES
("johnny","FR,EN,GR"),
("jim12","EN,FR");

```

Προσθέσαμε τα πεδία apokth8hke και ba8mos_ptxiyoy, τα οποία μας δείχνουν ποιο έτος πήρε ο χρήστης το πτυχίο και με τι βαθμό αντίστοιχα.

```

CREATE TABLE degree(
titlos VARCHAR(50) NOT NULL,
idryma VARCHAR(40) NOT NULL,
bathmida ENUM('LYKEIO','UNIV','MASTER','PHD'),
apokth8hke YEAR,
ba8mos_ptxiyoy FLOAT(3,1),
PRIMARY KEY (titlos),
UNIQUE(idryma)
);

```

```

INSERT INTO degree(titlos,idryma,bathmida,apokth8hke,ba8mos_ptxiyoy)VALUES
("ptyxio mhx ypologiston","UoP","UNIV","1999",9.9),
("ptyxio nomikhs","UoPa","UNIV","1978",8.7),
("ptyxio iatrikhs","UoPe","UNIV","2000",9.3),
("ptyxio agglikh filologias","UoA","UNIV","2001",8.0),
("ptyxio hlektrologon mhhanikon","UoAth","UNIV","1980",7.1);

```

```

CREATE TABLE has_degree(
degr_title VARCHAR(50) NOT NULL,
degr_idryma VARCHAR(40) NOT NULL,
hasdegree_empl_username VARCHAR(12) NOT NULL,
etos YEAR(4),
grade FLOAT(3,1),
CONSTRAINT degreetitlos FOREIGN KEY (degr_title) REFERENCES degree(titlos)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT degreeidryma FOREIGN KEY (degr_idryma) REFERENCES degree(idryma)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT hasdegreeemployeeusername FOREIGN KEY (hasdegree_empl_username) REFERENCES
employee(employee_username)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

INSERT INTO has_degree(degr_title,degr_idryma,hasdegree_empl_username,etos,grade)VALUES
("ptyxio iatrikhs","UoPe","tommy","2000",9.3),
("ptyxio hlektrologon mhxanikon","UoAth","marios123","1980",7.1);

```

Προσθέσαμε τα πεδία comment1, comment2, comment3, grade1, grade2, grade3, final_grade, final_comm, eval_status.

Τα 3 πεδία με τα comments μας δείχνουν τα σχόλια που έχουν γίνει για τον εργαζόμενο κατά τη διάρκεια της αξιολόγησης του για μια δουλειά.

Τα 3 πεδία με τα grades μας δείχνουν τι βαθμούς έχει πάρει ο εξεταζόμενος στις 3 φάσεις αξιολόγησης.

Το πεδίο final_grade είναι η πρόσθεση των προηγούμενων βαθμών, η οποία γίνεται αυτόματα.

Το πεδίο final_comm περιέχει τα σχόλια από το comment1 σύμφωνα με την εκφώνηση.

Το πεδίο eval_status παίρνει 2 τιμές ανάλογα με το αν έχει ολοκληρωθεί η αξιολόγηση ή όχι.

```

CREATE TABLE evaluationresult(
evid_result INT(4) NOT NULL,
empl_usrname VARCHAR(12) NOT NULL,
result_job_id INT(4) NOT NULL,
comment1 VARCHAR(85),
comment2 VARCHAR(85),
comment3 VARCHAR(85),
grade1 INT(1),
grade2 INT(1),
grade3 INT(1),
final_grade INT(2) AS (grade1 + grade2 + grade3) STORED,
final_comm VARCHAR(85),

```

```

eval_status ENUM('Pending', 'Finished'),
CONSTRAINT evidresult FOREIGN KEY (evid_result) REFERENCES evaluator(EvId)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT resultempoename FOREIGN KEY (empl_usrname) REFERENCES
employee(employee_username)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT resultjobid FOREIGN KEY (result_job_id) REFERENCES job(id)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

INSERT INTO
evaluationresult(evid_result,empl_usrname,result_job_id,comment1,comment2,comment3,final_comm,grade
1,grade2,grade3,eval_status)VALUES
(1,"marios123",1234,NULL,NULL,NULL,3,2,3,'Finished'),
(1,"marios123",5432,NULL,NULL,NULL,NULL,4,3,NULL,'Pending'),
(2,"nionios",2345,NULL,NULL,NULL,NULL,4,3,3,'Pending'),
(3,"johnny",4567,NULL,NULL,NULL,NULL,4,3,3,'Pending'),
(2,"nionios",7654,NULL,NULL,NULL,NULL,3,2,3,'Finished'),
(4,"jim12",6789,NULL,NULL,NULL,NULL,4,3,3,'Pending'),
(5,"tommy",7890,NULL,NULL,NULL,NULL,3,2,3,'Finished');

```

Τον συγκεκριμένο πίνακα τον προσθέσαμε εμείς, έτσι ώστε να αποθηκεύει τις δουλειές που επιλέγει κάθε εργαζόμενος να δηλώσει για να διεκδικήσει. Το πεδίο request_empl περιέχει τα ονόματα των εργαζομένων που κάνουν τις αιτήσεις. Είναι ξένο κλειδί, το οποίο παίρνουμε από τον πίνακα employee. Το πεδίο request_jobid είναι για την δουλειά για την οποία θέλει να κάνει αίτηση ο εργαζόμενος. Είναι ξένο κλειδί και το παίρνουμε από τον πίνακα job. Το πεδίο request_firm μας δείχνει σε ποια εταιρία ανήκει η συγκεκριμένη δουλειά. Είναι ξένο κλειδί και το παίρνουμε από τον πίνακα company.

```

CREATE TABLE job_for_request(
request_empl VARCHAR(12),
request_jobid INT(4),
request_firm CHAR(9),
CONSTRAINT requestemplname FOREIGN KEY (request_empl) REFERENCES employee(employee_username)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT requestjobid FOREIGN KEY (request_jobid) REFERENCES job(id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT requestfirm FOREIGN KEY (request_firm) REFERENCES company(afm)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```
INSERT INTO job_for_request(request_empl,request_jobid,request_firm)VALUES  
("marios123",1234,"123456789"),  
("marios123",5432,"123456789"),  
("nionios",2345,"234567890"),  
("johnny",4567,"345678909"),  
("nionios",7654,"234567890"),  
("jim12",6789,"121234345"),  
("tommy",7890,"343445456");
```

Προσθέσαμε το πεδίο kodikos, το οποίο είναι ο κωδικός της κάθε αξιολόγησης. Ο κωδικός αυτός είναι μοναδικός και για αυτό είναι πρωτεύων κλειδί. Επίσης αυξάνεται μόνο του με το auto_increment.

```
CREATE TABLE requestsevaluation(  
kodikos INT(3) NOT NULL AUTO_INCREMENT,  
empl_username VARCHAR(12) NOT NULL,  
request_job_id INT(4) NOT NULL,  
PRIMARY KEY(kodikos),  
CONSTRAINT requestEmployeeName FOREIGN KEY (empl_username) REFERENCES  
employee(employee_username)  
ON DELETE CASCADE ON UPDATE CASCADE,  
CONSTRAINT requestJobId FOREIGN KEY (request_job_id) REFERENCES job_for_request(request_jobid)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
INSERT INTO requestsevaluation(kodikos,empl_username,request_job_id)VALUES  
(NULL,"marios123",1234),  
(NULL,"marios123",5432),  
(NULL,"nionios",2345),  
(NULL,"johnny",4567),  
(NULL,"nionios",7654),  
(NULL,"jim12",6789),  
(NULL,"tommy",7890);
```

Τον πίνακα αυτόν τον δημιουργήσαμε εμείς σύμφωνα με τα ζητούμενα της εκφώνησης, όπου καταγράφεται το συμβάν όταν κάποιος χρήστης πάει να κάνει insert, update ή delete σε κάποιον από τους πίνακες job, employee ή requestsevaluation. Έτσι στο πεδίο login_username αποθηκεύουμε το όνομα του χρήστη, στο πεδίο symban αποθηκεύουμε την ημερομηνία και ώρα που γίνεται το συμβάν, στο πεδίο epityxia αποθηκεύουμε αν το συμβάν είχε επιτυχία ή όχι, στο πεδίο eidos αποθηκεύουμε το είδος της ενέργειας που γίνεται και τέλος στο πεδίο pinakas αποθηκεύουμε τον πίνακα στον οποίο γίνεται η ενέργεια.

Insert δεν υπάρχει στην συγκεκριμένη περίπτωση, καθώς το insert θα γίνεται με triggers, τα οποία τα παραθέτουμε παρακάτω.

```
CREATE TABLE login(
    login_username VARCHAR(12),
    symban DATETIME,
    epityxia BOOLEAN,
    eidos ENUM("insert","update","delete"),
    pinakas ENUM("job","employee","requestsevaluation"),
    CONSTRAINT loginusername FOREIGN KEY (login_username) REFERENCES user(username)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

Procedures

1o procedure:

```
DROP PROCEDURE IF EXISTS FirstProcedure;

DELIMITER $

CREATE PROCEDURE FirstProcedure(IN u_name VARCHAR(25), IN u_sur VARCHAR(35))

BEGIN

DECLARE job INT(4);

DECLARE eval_id INT(4);

DECLARE e_status VARCHAR(10);

DECLARE finishedFlag INT;

DECLARE a_cursor CURSOR FOR SELECT e.result_job_id, e.evid_result,e.eval_status FROM evaluationresult
AS e INNER JOIN employee AS m ON m.employee_username=e.empl_username INNER JOIN user AS u ON
u.username=m.employee_username WHERE u.user_name=u_name AND u.user_surname=u_sur;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finishedFlag=1;

OPEN a_cursor;

SET finishedFlag=0;

FETCH a_cursor INTO job,eval_id,e_status;

WHILE (finishedFlag=0) DO

    SELECT job AS "Job Requested", eval_id AS "Evaluator", e_status AS "Status";

    FETCH a_cursor INTO job,eval_id,e_status;

END WHILE;

CLOSE a_cursor;

END $

DELIMITER ;
```

2ο procedure:

```
DROP PROCEDURE IF EXISTS SecondProcedure;

DELIMITER $

CREATE PROCEDURE SecondProcedure(IN id_erg INT(4), IN id_eval INT(4))

BEGIN

DECLARE checker INT;
DECLARE finishedFlag INT;

UPDATE evaluationresult SET eval_status='Finished' WHERE evid_result=id_eval AND result_job_id=id_erg
AND grade1 IS NOT NULL AND grade2 IS NOT NULL AND grade3 IS NOT NULL AND eval_status='Pending';

END $

DELIMITER ;
```

3ο procedure:

Μετράμε πόσες φορές έχουμε pending και πόσες finished, έτσι ώστε να μπορέσουμε να συγκρίνουμε μετά στα if και να πάρουμε τις περιπτώσεις που ζητούνται στην εκφώνηση.

```
DROP PROCEDURE IF EXISTS oloklhrosi_a3iologhseon;

DELIMITER $

CREATE PROCEDURE oloklhrosi_a3iologhseon( IN job INT(4) )

BEGIN

DECLARE flag INT;
DECLARE grade INT(1);
DECLARE pending INT(1);
DECLARE finished INT(1);
DECLARE empl_name VARCHAR(25);
DECLARE jobb INT(4);

DECLARE acursor CURSOR FOR

    SELECT empl_username,result_job_id,final_grade
    FROM evaluationresult
    WHERE job=result_job_id
    GROUP BY final_grade DESC;
```

```

DECLARE CONTINUE HANDLER FOR NOT FOUND SET flag=1;

OPEN acursor;
SET flag=0;
REPEAT
  FETCH acursor INTO empl_name,jobb,grade;
  IF (flag =0 ) THEN
    SET pending= (SELECT count(eval_status) FROM evaluationresult WHERE job=result_job_id AND eval_status= "Pending");
    SET finished= (SELECT count(eval_status) FROM evaluationresult WHERE job=result_job_id AND eval_status= "Finished");
    IF(pending> 0 ) THEN
      SELECT empl_name AS "employee's name",grade AS "grade";
      SELECT "Αξιολόγηση σε εξέλιση...εκκρεμούν," ,pending, " ,αιτήσεις";
    ELSEIF (pending= 0 AND finished =0) THEN
      SELECT "Δεν υπάρχουν υποψήφιοι για αυτή την δουλειά";
    ELSEIF (pending=0 AND finished >0) THEN
      SELECT "Οριστικοποιημένοι Πίνακες";
      SELECT empl_name AS "employee's name",grade AS "grade";
    END IF;
  END IF;
  UNTIL (flag=1)
END REPEAT;
CLOSE acursor;
END$
DELIMITER ;

```

Triggers

Triggers για τον πίνακα καταγραφής ενεργειών (login).

Τα 3 triggers για τα insert, update και delete στον πίνακα job

1)

```
DROP TRIGGER IF EXISTS trigger1;  
DELIMITER $  
CREATE TRIGGER trigger1 BEFORE INSERT ON job  
FOR EACH ROW  
BEGIN  
DECLARE i INT;  
DECLARE currDate DATETIME;  
DECLARE name VARCHAR(12);  
SET currDate=CURRENT_TIMESTAMP();  
SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;  
IF (i>0) THEN  
INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)  
VALUES(name,currDate,TRUE,"insert","job");  
END IF;  
END$  
DELIMITER ;
```

2)

```
DROP TRIGGER IF EXISTS trigger2;  
DELIMITER $  
CREATE TRIGGER trigger2 BEFORE UPDATE ON job  
FOR EACH ROW  
BEGIN  
DECLARE i INT;  
DECLARE currDate DATETIME;
```

```

DECLARE name VARCHAR(12);

SET currDate=CURRENT_TIMESTAMP();

SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;

IF (i>0) THEN

INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)
VALUES(name,currDate,TRUE,"update","job");

END IF;

END$


DELIMITER ;

```

3)

```

DROP TRIGGER IF EXISTS trigger3;

DELIMITER $

CREATE TRIGGER trigger3 BEFORE DELETE ON job
FOR EACH ROW
BEGIN
DECLARE i INT;
DECLARE currDate DATE;
DECLARE name VARCHAR(12);
SET currDate=CURRENT_TIMESTAMP();
SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;
IF (i>0) THEN
INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)
VALUES(name,currDate,TRUE,"delete","job");
END IF;
END$


DELIMITER ;

```

Tα 3 triggers για τα insert, update και delete στον πίνακα employee

1)

```
DROP TRIGGER IF EXISTS trigger4;

DELIMITER $

CREATE TRIGGER trigger4 BEFORE INSERT ON employee
FOR EACH ROW
BEGIN
DECLARE i INT;
DECLARE currDate DATE;
DECLARE name VARCHAR(12);
SET currDate=CURRENT_TIMESTAMP();
SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;
IF (i>0) THEN
INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)
VALUES(name,currDate,TRUE,"insert","employee");
END IF;
END$
```

DELIMITER ;

2)

```
DROP TRIGGER IF EXISTS trigger5;

DELIMITER $

CREATE TRIGGER trigger5 BEFORE UPDATE ON employee
FOR EACH ROW
BEGIN
DECLARE i INT;
DECLARE currDate DATETIME;
DECLARE name VARCHAR(12);
SET currDate=CURRENT_TIMESTAMP();
SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;
IF (i>0) THEN
INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)
VALUES(name,currDate,TRUE,"update","employee");
```

```
END IF;
```

```
END$
```

```
DELIMITER ;
```

3)

```
DROP TRIGGER IF EXISTS trigger6;
```

```
DELIMITER $
```

```
CREATE TRIGGER trigger6 BEFORE DELETE ON employee
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE i INT;
```

```
DECLARE currDate DATETIME;
```

```
DECLARE name VARCHAR(12);
```

```
SET currDate=CURRENT_TIMESTAMP();
```

```
SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;
```

```
IF (i>0) THEN
```

```
INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)
```

```
VALUES(name,currDate,TRUE,"delete","employee");
```

```
END IF;
```

```
END$
```

```
DELIMITER ;
```

Τα 3 triggers για τα insert, update και delete στον πίνακα requestsevaluation

1)

```
DROP TRIGGER IF EXISTS trigger7;
```

```
DELIMITER $
```

```
CREATE TRIGGER trigger7 BEFORE INSERT ON requestsevaluation
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE i INT;
```

```
DECLARE currDate DATE;
```

```
DECLARE name VARCHAR(12);
```

```
SET currDate=CURRENT_TIMESTAMP();
```

```
SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;
```

```
IF (i>0) THEN
INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)
VALUES(name,currDate,TRUE,"insert","requestsevaluation");
END IF;
END$
```

DELIMITER ;

2)

```
DROP TRIGGER IF EXISTS trigger8;
DELIMITER $
CREATE TRIGGER trigger8 BEFORE UPDATE ON requestsevaluation
FOR EACH ROW
BEGIN
DECLARE i INT;
DECLARE currDate DATE;
DECLARE name VARCHAR(12);
SET currDate=CURRENT_TIMESTAMP();
SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;
IF (i>0) THEN
INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)
VALUES(name,currDate,TRUE,"update","requestsevaluation");
END IF;
END$
```

DELIMITER ;

3)

```
DROP TRIGGER IF EXISTS trigger9;

DELIMITER $

CREATE TRIGGER trigger9 BEFORE DELETE ON requestsevaluation
FOR EACH ROW
BEGIN
DECLARE i INT;
DECLARE currDate DATE;
DECLARE name VARCHAR(12);
SET currDate=CURRENT_TIMESTAMP();
SELECT COUNT(kodikos) INTO i FROM requestsevaluation ;
IF (i>0) THEN
INSERT INTO login(login_username,symban,epityxia,eidos,pinakas)
VALUES(name,currDate,TRUE,"delete","requestsevaluation");
END IF;
END$
```

DELIMITER ;

Το δεύτερο trigger, στο οποίο απαγορεύεται να αλλάξουμε τα afm, doy και company name σε μια εταιρία. Και έτσι σε προσπάθεια αλλαγής, αντικαθιστούμε την νέα τιμή με την παλιά.

```
DROP TRIGGER IF EXISTS enhmerosh_etairias;

DELIMITER $

CREATE TRIGGER enhmerosh_etairias BEFORE UPDATE ON company
FOR EACH ROW
BEGIN
SET NEW.afm = OLD.afm;
SET NEW.doy = OLD.doy;
SET NEW.company_name=OLD.company_name;
END$
```

DELIMITER ;

Το τρίτο trigger, στο οποίο απαγορεύεται να αλλάξουμε το username μας και έτσι σε περίπτωση αλλαγής, αντικαθιστούμε το νέο username με το παλιό και βάζουμε στο πεδίο password την τιμή 0, έτσι ώστε να προκύψει error.

```
DROP TRIGGER IF EXISTS c_trigger;
```

```
DELIMITER $
```

```
CREATE TRIGGER c_trigger
```

```
BEFORE UPDATE ON user
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    SET NEW.username=OLD.username;
```

```
    SET NEW.password=NULL;
```

```
END $
```

```
DELIMITER ;
```

Έξτρα procedure:

Procedure για να μπορεί ο manager να βρίσκει το average των βαθμολογιών ενός evaluator. Ως όρισμα βάζουμε το Id του συγκεκριμένου evaluator που θέλουμε.

```
DROP PROCEDURE IF EXISTS average;

DELIMITER $

CREATE PROCEDURE average(IN id INT(4))

BEGIN

DECLARE avg FLOAT(2);

DECLARE finishedFlag INT;

DECLARE a_cursor CURSOR FOR SELECT AVG(final_grade) FROM evaluationresult WHERE evid_result = id;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finishedFlag=1;

OPEN a_cursor;

SET finishedFlag=0;

FETCH a_cursor INTO avg;

WHILE (finishedFlag=0) DO

    SELECT avg AS "Average";

    FETCH a_cursor INTO avg;

END WHILE;

CLOSE a_cursor;

END $

DELIMITER ;
```

Έξτρα triggers:

Φτιάξαμε ένα trigger για να βάζουν οι χρήστες την σωστή μορφή ενός email. Αν κάνουν λάθος θέτουμε την τιμή 0 στο πεδίο password για να βγει error.

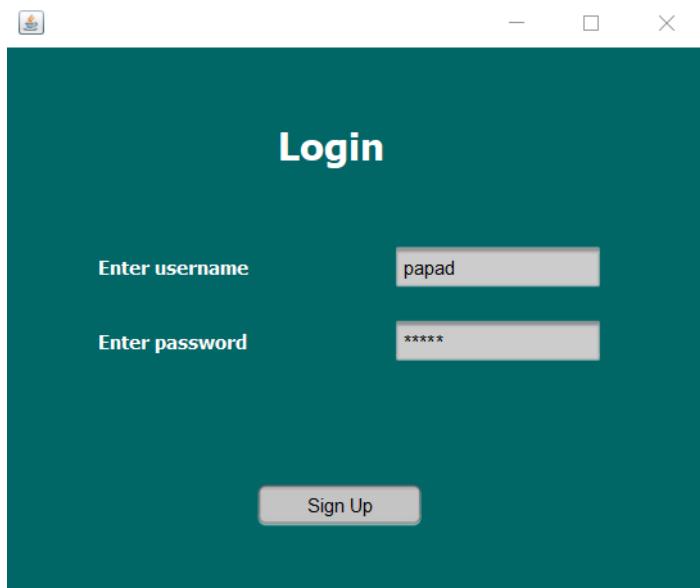
```
DELIMITER $  
CREATE TRIGGER validateEmail  
BEFORE INSERT ON user  
FOR EACH ROW  
BEGIN  
IF NEW.email NOT LIKE '%@%.%' THEN  
SET NEW.password=NULL;  
END IF;  
END$  
DELIMITER ;
```

Φτιάξαμε ένα trigger για να βάζουν οι χρήστες την σωστή μορφή ενός url. Αν κάνουν λάθος θέτουμε την τιμή 0 στο πεδίο empl για να βγει error.

```
DELIMITER $  
CREATE TRIGGER validateURL  
BEFORE INSERT ON project  
FOR EACH ROW  
BEGIN  
IF NEW.url NOT LIKE 'https://www.%.%' THEN  
SET NEW.empl=NULL;  
END IF;  
END$  
DELIMITER ;
```

Β ΜΕΡΟΣ (GUI)

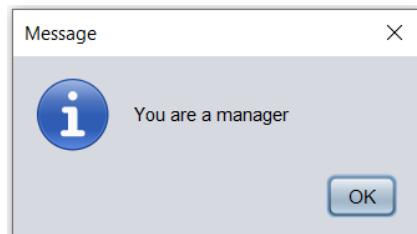
Manager:



Αν συνδεθώ με σωστά δεδομένα, τότε θα πάρω το κατάλληλο μήνυμα ότι ο το username και ο κωδικός ταίριαξαν με τα δεδομένα της βάσης, οπότε ο χρήστης αυτός υπάρχει :



Επίσης μου εμφανίζεται και το μήνυμα:

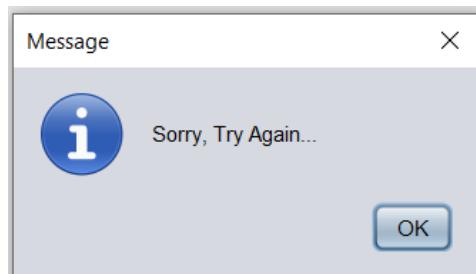


Το οποίο μου δείχνει τι χρήστης είμαι.

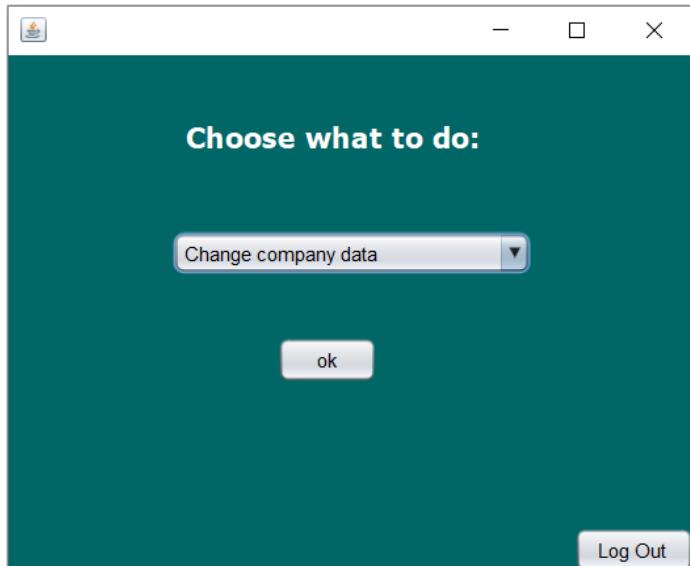
Κοιτάζοντας στην βάση, βλέπω ότι όντως ο papad είναι manager.

MariaDB [baseis]> SELECT * FROM manager;			
manager_username	exp_years	firm	
nefelaki	5	123456789	
fotisre	2	234567890	
euridik	1	345678909	
papad	2	121234345	
takis12	3	343445456	
5 rows in set (0.002 sec)			

Σε περίπτωση που βάλω λάθος username ή κωδικό, τότε θα πάρω το μήνυμα:



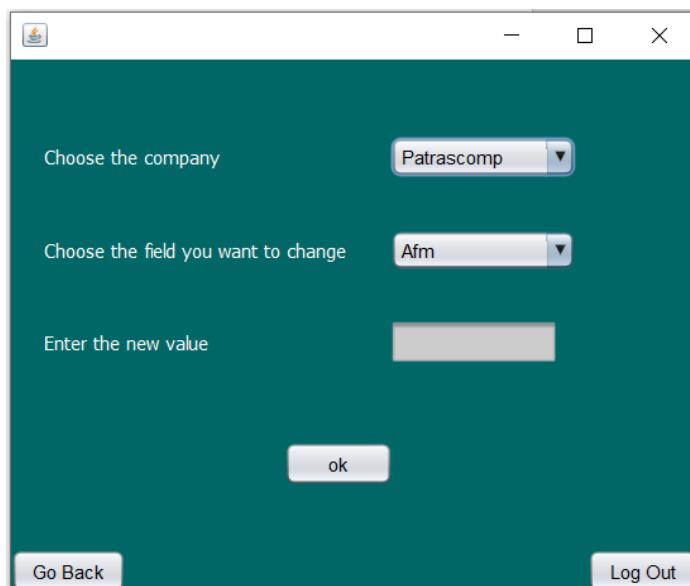
Αφού συνδεθώ, θα βρεθώ στην αρχική σελίδα του Manager:



Όπου οι ενέργειες που μπορεί να εκτελέσει φαίνονται πατώντας το κουτί:



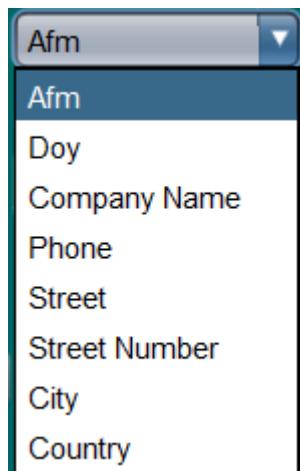
Επιλέγοντας την πρώτη και πατώντας το κουμπί "ok" έχω:



Όπου στο πρώτο κουτί έχουμε τις εταιρίες που υπάρχουν:



Ενώ στο δεύτερο κουτί έχουμε τα πεδία της εταιρίας που μπορούμε να αλλάξουμε:



Σε περίπτωση που επιλέξουμε κάποιο πεδίο που δεν επιτρέπεται να αλλάξουμε, τότε δεν μπορούμε να γράψουμε στο πεδίο κειμένου που βρίσκεται παρακάτω.

Όταν πατήσουμε το κουμπί "ok" θα αλλάξει στη βάση μας το πεδίο που επιλέξαμε και θα μπει εκεί η νέα τιμή.

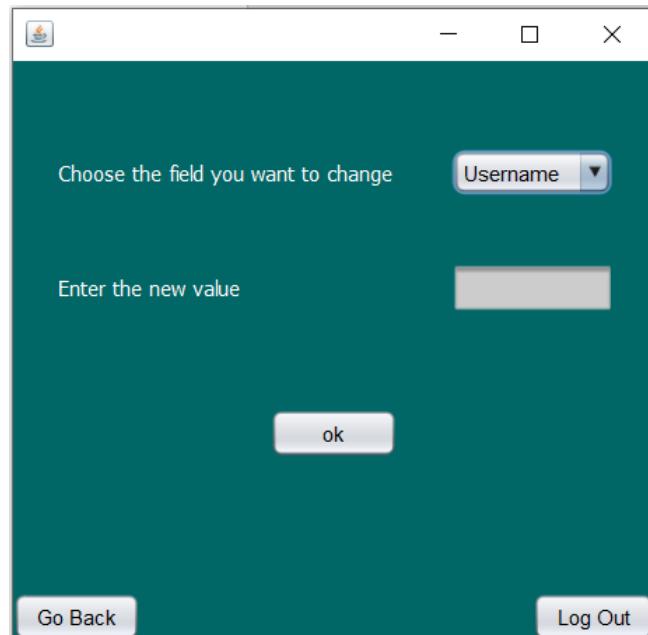
Go Back

και

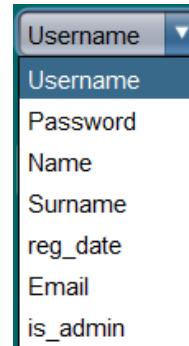
Log Out

Μπορούμε να πάμε ένα βήμα πίσω ή να κάνουμε log out αντίστοιχα.

Επιλέγοντας την δεύτερη επιλογή:



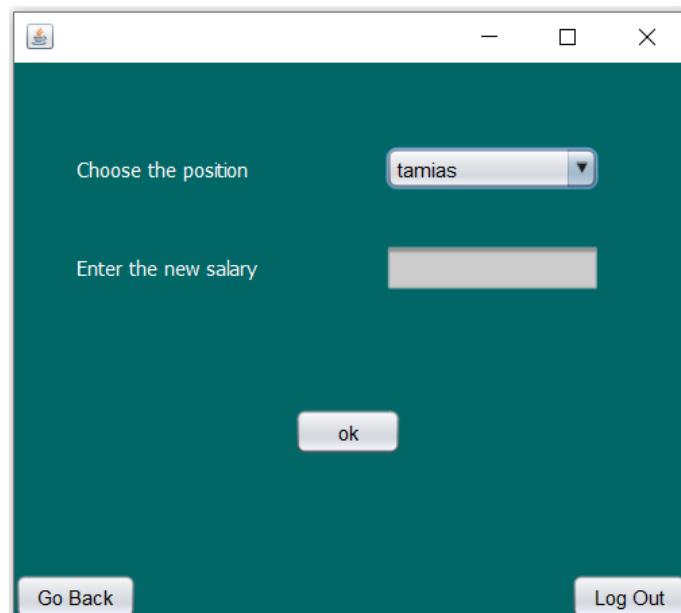
Όπου στο κουτί έχουμε τα πεδία του προφίλ του χρήστη και καλούμαστε να επιλέξουμε ένα για να αλλάξουμε κάποια τιμή.



Σε περίπτωση που επιλέξουμε κάποιο πεδίο που δεν επιτρέπεται να αλλάξουμε, τότε δεν μπορούμε να γράψουμε στο πεδίο κειμένου που βρίσκεται παρακάτω.

Όταν πατήσουμε το κουμπί "ok" θα αλλάξει στη βάση μας το πεδίο που επιλέξαμε και θα μπει εκεί η νέα τιμή.

Επιλέγοντας την τρίτη επιλογή:



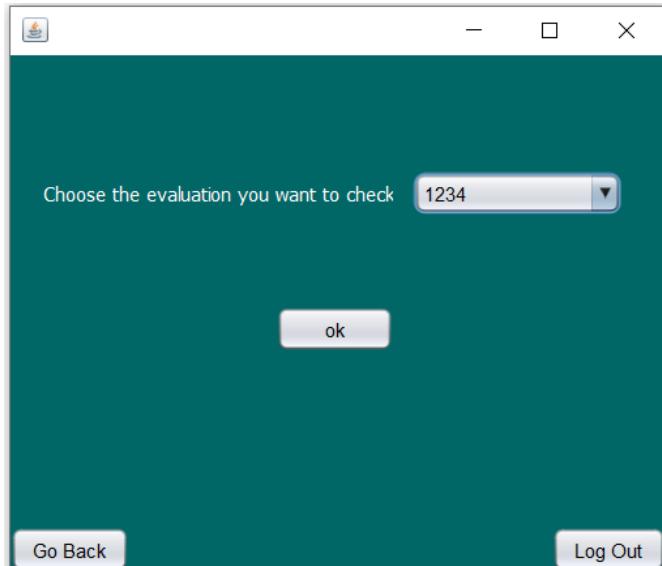
Όπου στο κουτί έχουμε τις θέσεις εργασίας που έχουν ανακοινωθεί για προαγωγή:



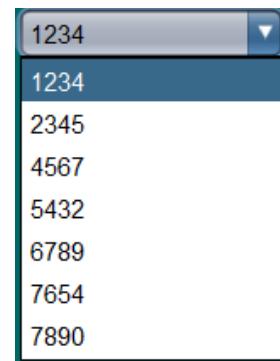
Μπορούμε να επιλέξουμε μια από τις θέσεις αυτές και να αλλάξουμε τον μισθό τους.

Όταν πατήσουμε το κουμπί “ok” θα αλλάξει στη βάση μας ο μισθός της συγκεκριμένης θέσης και θα μπει εκεί η νέα τιμή.

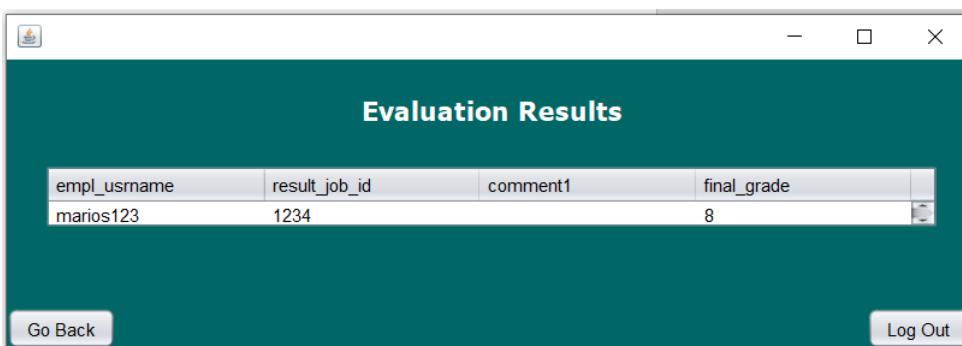
Επιλέγοντας την τέταρτη επιλογή:



Όπου στο κουτί έχουμε τα id των jobs που έχουν δηλωθεί.

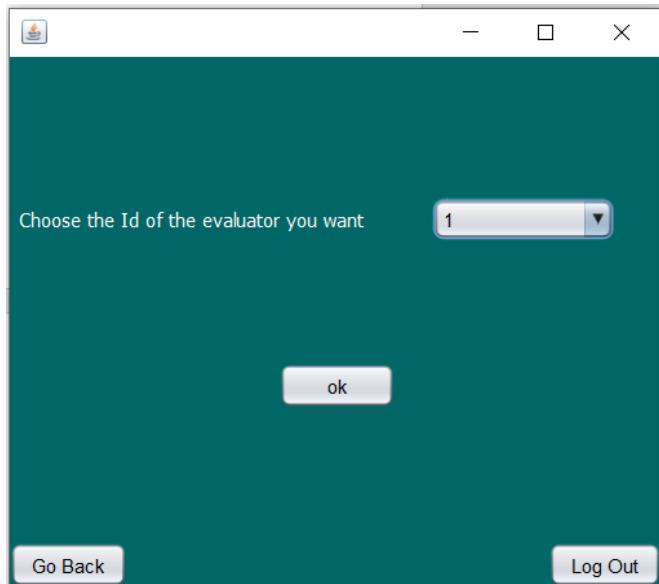


Επιλέγοντας ένα από τα id και πατώντας το κουμπί “ok” :

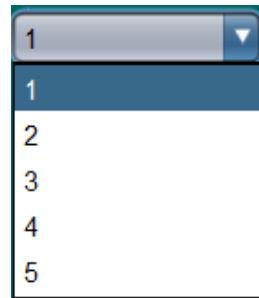


Βλέπουμε ένα πίνακα με την αξιολόγηση των ατόμων που έχουν κάνει αίτηση για αυτή την δουλειά και έχει ολοκληρωθεί η αξιολόγηση τους.

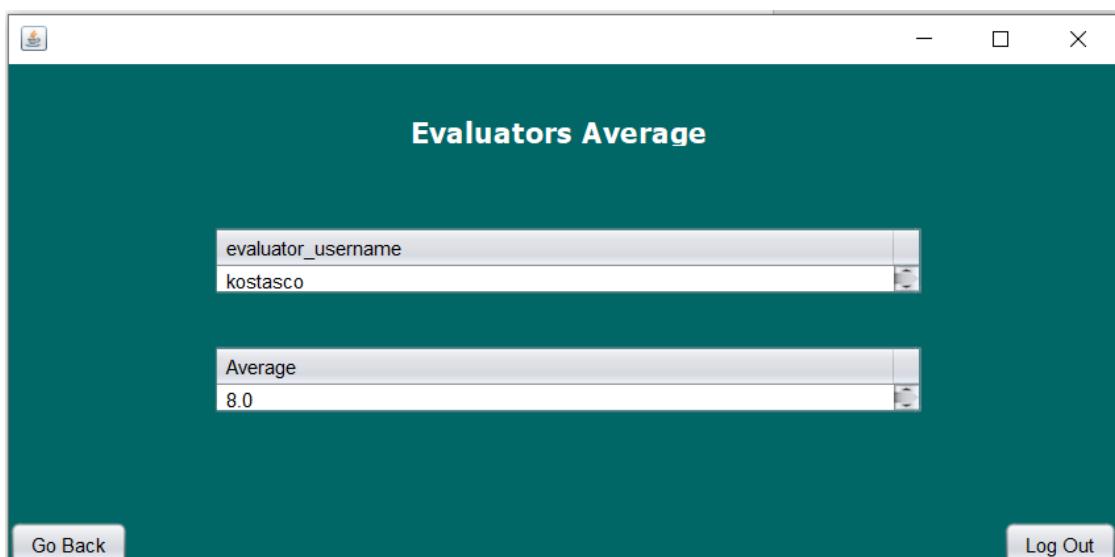
Επιλέγοντας την πέμπτη επιλογή:



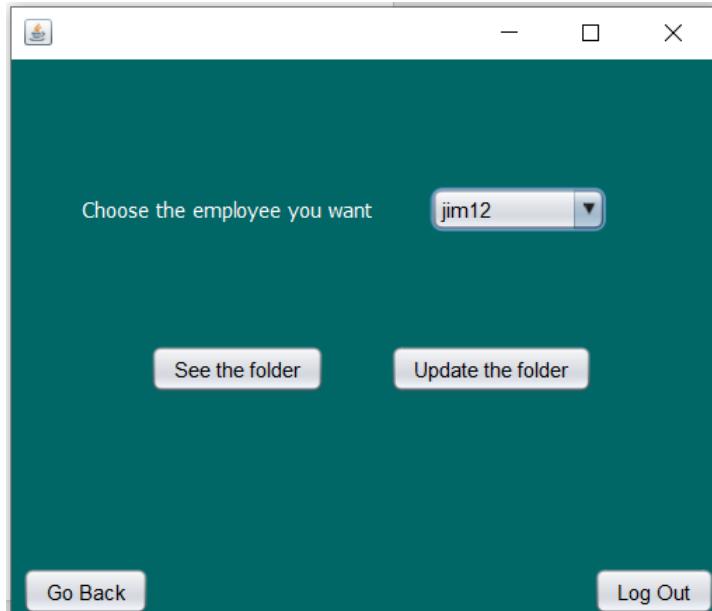
Όπου το κουτί περιέχει τα id των evaluators:



Μπορούμε να επιλέξουμε έναν evaluator (από το id του) για να δούμε τον μέσο όρο των αξιολογήσεων του.



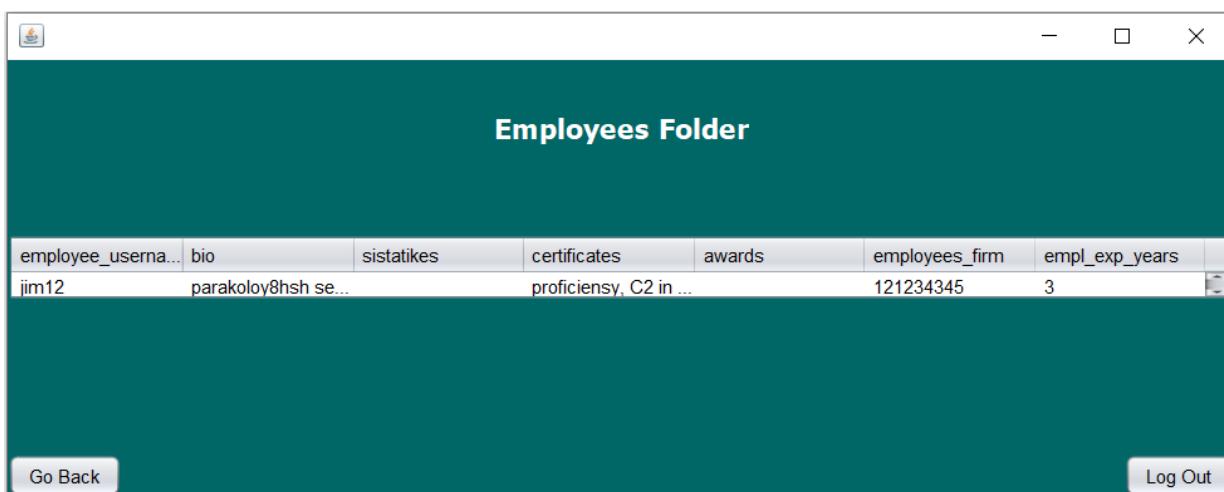
Επιλέγοντας την τελευταία επιλογή:



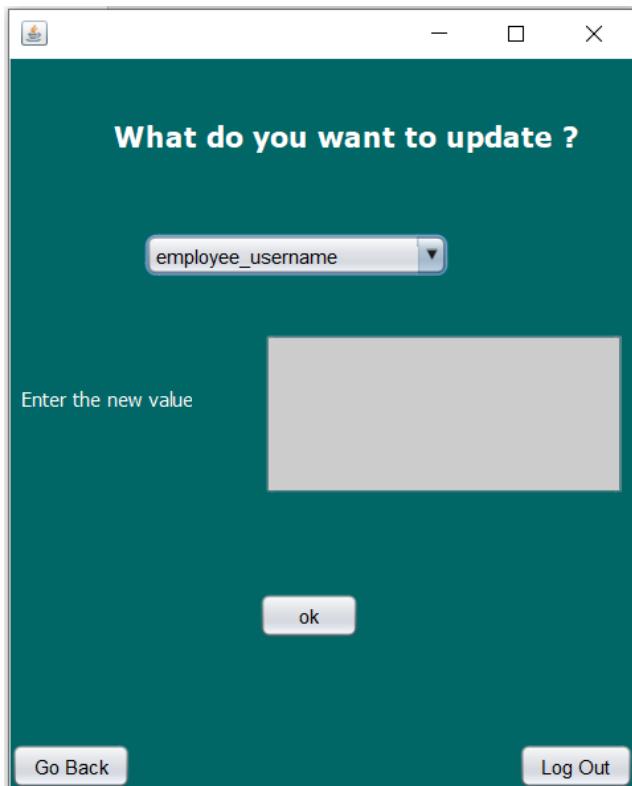
Όπου στο κουτί έχουμε τα username των employees:



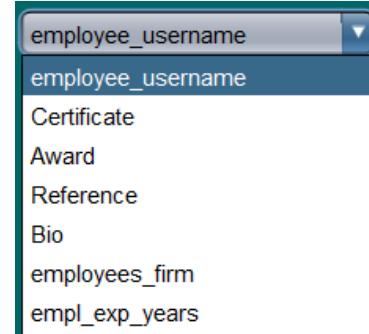
Μπορούμε να επιλέξουμε το κουμπί και να δούμε τα στοιχεία του συγκεκριμένου employee που επιλέξαμε:



Ή να επιλέξουμε το κουμπί **Update the folder** και να κάνουμε update στα στοιχεία ενός employee.



Όπου στο κουτί είναι τα πεδία του employee:



Σε περίπτωση που επιλέξουμε κάποιο πεδίο που δεν επιτρέπετε να αλλάξουμε, τότε δεν μπορούμε να γράψουμε στο πεδίο κειμένου που βρίσκεται παρακάτω.

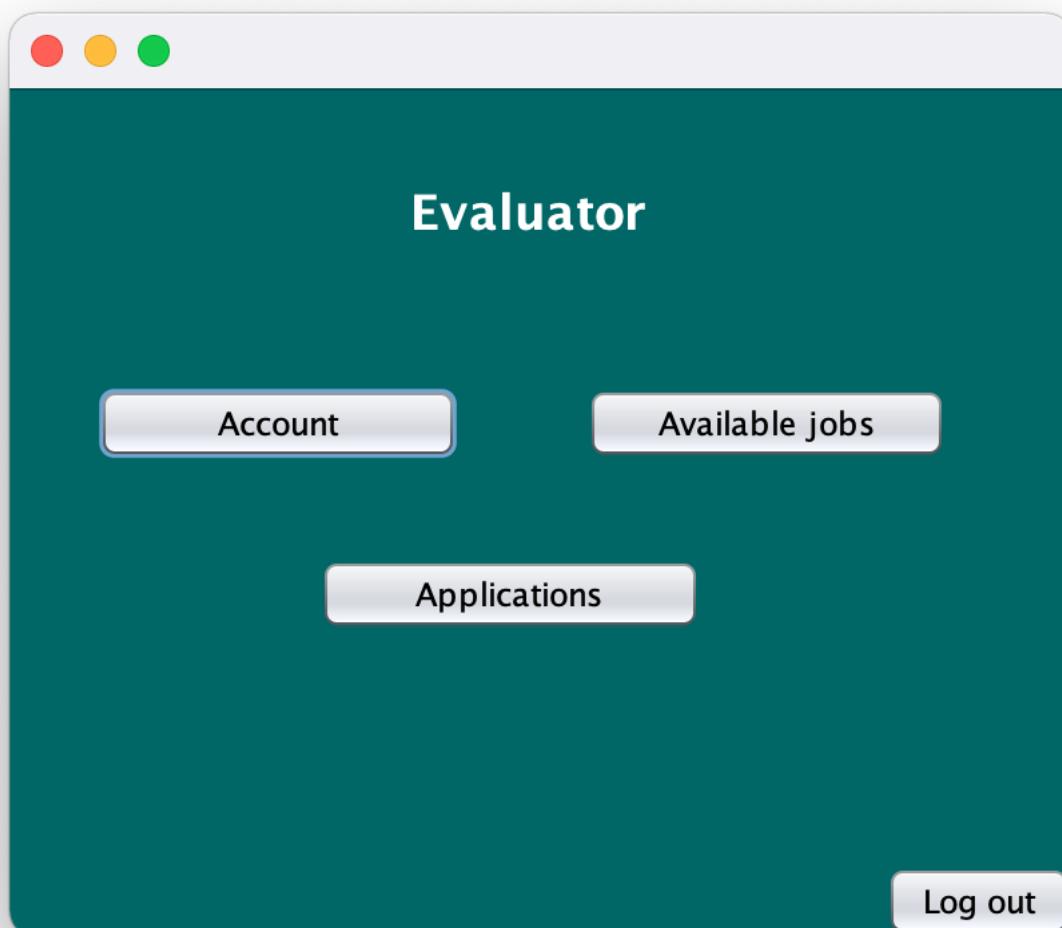
Όταν πατήσουμε το κουμπί “ok” θα αλλάξει στη βάση μας το πεδίο που επιλέξαμε και θα μπει εκεί η νέα τιμή.

Evaluator:

Μόλις κάνουμε Log in εμφανίζεται αντίστοιχο μήνυμα ότι ο χρήστης είναι employee:

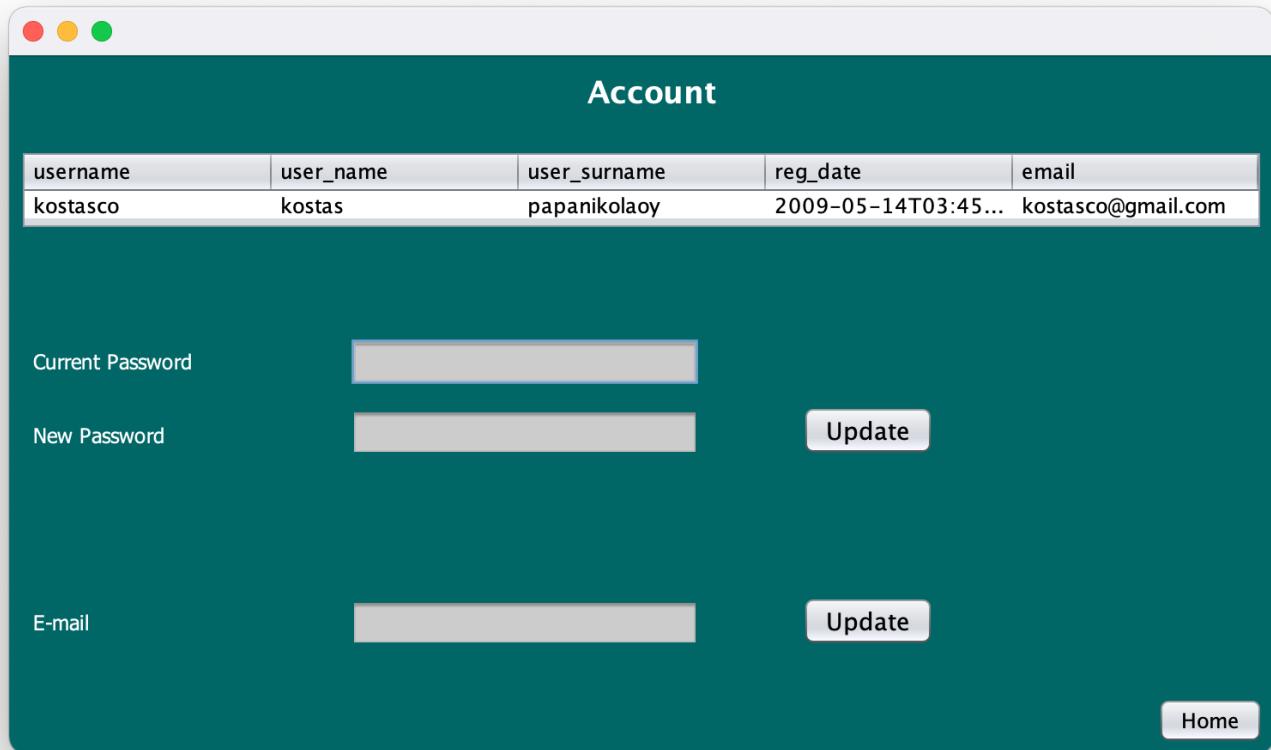


Εδώ βλέπουμε την αρχική σελίδα όπου υπάρχουν οι διαθέσιμες ενέργειες που μπορεί να κάνει ο evaluator:



Account:

Το γραφικό περιβάλλον πριν κάνει τις αλλαγές στο προφίλ του χρήστη:



Τα δεδομένα στον πίνακα πριν ο συγκεκριμένος evaluator κάνει αλλαγές στο προφίλ του:

```
[mysql> select username, password, email from user where username='kostasco';
+-----+-----+-----+
| username | password | email           |
+-----+-----+-----+
| kostasco | 123456   | kostasco@gmail.com |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Το γραφικό περιβάλλον όταν κάνει τις αλλαγές στο προφίλ του ο χρήστης:

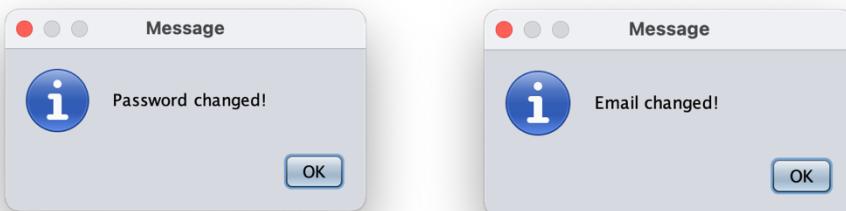
The screenshot shows a window titled "Account". At the top, there is a table with columns: username, user_name, user_surname, reg_date, and email. The data is as follows:

username	user_name	user_surname	reg_date	email
kostasco	kostas	papanikolaoy	2009-05-14T03:45...	kostasco@gmail.com

Below the table, there are two sets of input fields for password and email, each with an "Update" button. The first set is for "Current Password" (*****) and "New Password" (***). The second set is for "E-mail" (kostasco1@gmail.com).

At the bottom right of the window is a "Home" button.

Τα αντίστοιχα μηνύματα επιτυχίας:



Τα δεδομένα στον πίνακα αφού ο συγκεκριμένος evaluator κάνει αλλαγές στο προφίλ του:

```
[mysql] > select username, password, email from user where username='kostasco';
+-----+-----+-----+
| username | password | email          |
+-----+-----+-----+
| kostasco | 123      | kostasco1@gmail.com |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Available Jobs:

Ο πίνακας με τις δουλειές που είναι αναρτημένες στην εταιρεία:

The screenshot shows a software application window titled "Available Jobs". On the left, there is a vertical list of fields with input boxes: ID, Start Date, Salary, Position, Edra, Evaluator, Announce Date, Submission Date, Firm, and Antikeimeno (with a dropdown menu). Below these fields is a large table containing job records. The table has columns: id, start_date, salary, position, edra, job_eval..., announc..., submissi..., jobs_firm, and antikeim... . The data in the table is as follows:

	id	start_date	salary	position	edra	job_eval...	announc...	submissi...	jobs_firm	antikeim...
1234	1234	2003-0...	1350.8	tamias	patra	kostasco	2003-0...	2030-0...	123456...	oikonom...
3018	3018	2010-0...	800.0	tamias	athena	leonid	2003-0...	2030-0...	123456...	
5432	5432	2030-0...	890.5	boh8os	patra	kostasco	2001-0...	2029-0...	123456...	gramateia

At the bottom of the window are buttons for Add New, Update, Delete, Reset, and Home.

Τα πεδία Evaluator και Firm είναι μη επεξεργάσιμα.

Κάνοντας εισαγωγή μιας νέας θέσης:

The screenshot shows the same software application window after entering new job details. The fields on the left now contain: ID (9090), Start Date (2021-05-03), Salary (1000.0), Position (upalihos), Edra (athena), Evaluator (kostasco), Announce Date (2021-04-03), Submission Date (2021-05-01), Firm (123456789), and Antikeimeno (gramateia). The table on the right remains the same as in the first screenshot.

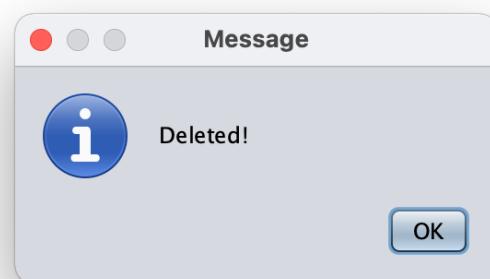
Μήνυμα επιτυχίας:



Πατώντας επάνω σε μια θέση εισάγονται αυτόματα τα στοιχεία στα πεδία αριστερά:

	id	start_date	salary	position	edra	job_evaluator	announce_date	submission_date	jobs_firm	antikeimeno
ID	1234	2003-01-01	1350.8	tamias	patra	kostasco	2003-01-01	2030-01-01	123456789	oikonomia
Start Date		2003-07-21								
Salary			1350.8							
Position				tamias						
Edra					patra					
Evaluator						kostasco				
Announce Date							2003-05-21T00:00			
Submission Date								2030-06-21		
Firm									123456789	
Antikeimeno										gramateia

Έτσι μπορούμε να κάνουμε αντίστοιχα Update αλλάζοντας κάποιο από τα επεξεργάσιμα πεδία και Delete της θέσης και παίρνουμε τα αντίστοιχα μηνύματα:



Applications:

Επιλέγοντας κάποιο ID θέσης εργασίας εμφανίζονται όλες οι αιτήσεις για αξιολόγηση που έχουν γίνει από employees:

A screenshot of a Mac OS X application window titled "Applications". The window contains a table with three columns: "kodikos", "empl_username", and "request_job_id". There is one row of data: "9", "marios123", and "5432". Below the table is a search bar containing "5432" and a dropdown arrow, and a "Home" button.

kodikos	empl_username	request_job_id
9	marios123	5432

Επιλέγοντας την αίτηση εμφανίζεται νέο παράθυρο με τα αποτελέσματα της αξιολόγησης:

A screenshot of a Mac OS X application window titled "Applications". The window contains a form for entering grades. It has fields for "Employee" (marios123) and "Job ID" (5432). Below these are four input fields for "Grade 1" (3), "Grade 2" (3), "Grade 3" (2), and "Final Grade" (8). There are three text areas for comments: "Comment 1" (oshfbgbosdhfg), "Comment 2" (gdsfgsg), and "Comment 3" (sdfgsdf). At the bottom are "Update" and "Back" buttons.

Employee: marios123
Job ID: 5432

Grade 1: 3 Grade 2: 3 Grade 3: 2 Final Grade: 8

Comment 1: oshfbgbosdhfg
Comment 2: gdsfgsg
Comment 3: sdfgsdf

Update Back

Κάνοντας Update ανανεώνονται οι πίνακες στη βάση και εμφανίζεται αυτόματα ο καινούργιος τελικός βαθμός:

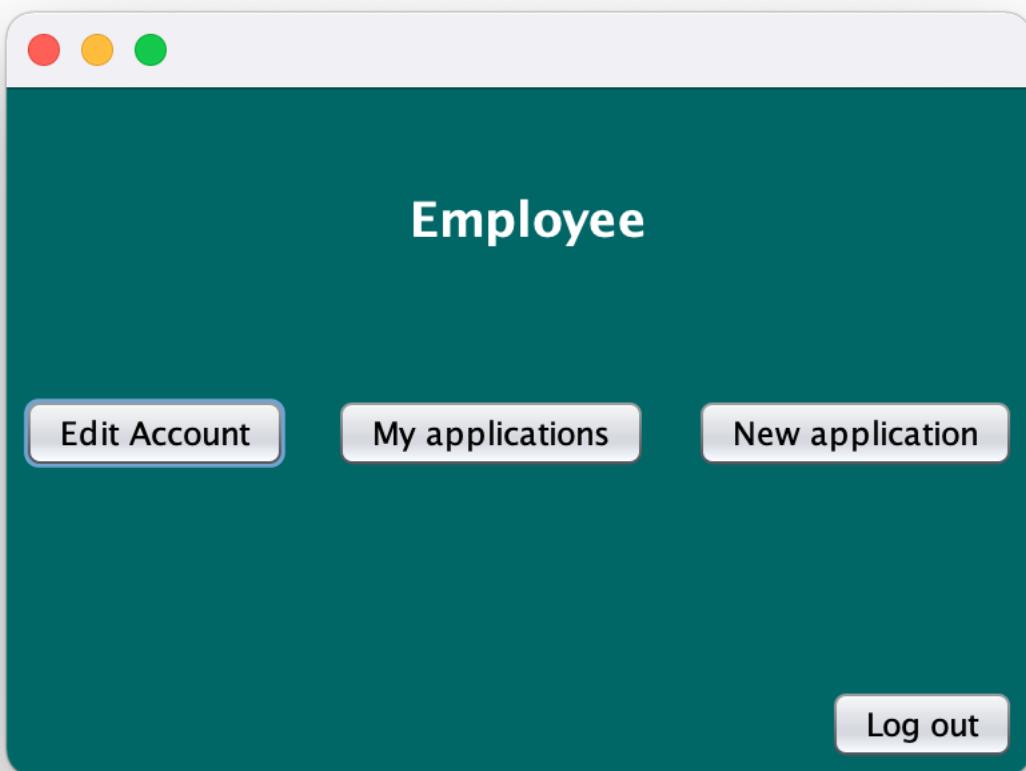


Employee:

Μόλις κάνουμε Log in εμφανίζεται αντίστοιχο μήνυμα ότι ο χρήστης είναι employee:



Εδώ βλέπουμε την αρχική σελίδα όπου υπάρχουν οι διαθέσιμες ενέργειες που μπορεί να κάνει ο employee:



Edit Account:

To γραφικό περιβάλλον πριν κάνει τις αλλαγές στο προφίλ του ο χρήστης:

Username: marios123

Update Bio: apofoitos mhxanologwn

Update Password:

Current Password:

New Password:

Update

Update

Home Page

Tα δεδομένα στον πίνακα πριν ο συγκεκριμένος employee κάνει αλλαγές στο προφίλ του:

```
[mysql] > select employee_username, bio from employee where employee_username='marios123';
+-----+-----+
| employee_username | bio           |
+-----+-----+
| marios123        | apofoitos mhxanologwn |
+-----+-----+
1 row in set (0.00 sec)

[mysql] > select username, password from user where username='marios123';
+-----+-----+
| username | password |
+-----+-----+
| marios123 | 123456   |
+-----+-----+
1 row in set (0.00 sec)
```

To γραφικό περιβάλλον όταν κάνει τις αλλαγές ο χρήστης:

Username: marios123

Update Bio: apofoitos CEID

Update Password:

Current Password:

New Password:

Update

Update

Home Page

Μηνύματα για την ολοκλήρωση των αλλαγών:



Τα δεδομένα στον πίνακα αφού ο συγκεκριμένος employee κάνει αλλαγές στο προφίλ του:

```
[mysql] select employee_username, bio from employee where employee_username='marios123';
+-----+
| employee_username | bio          |
+-----+
| marios123        | apofoitos CEID |
+-----+
1 row in set (0.00 sec)

[mysql] select username, password from user where username='marios123';
+-----+
| username | password |
+-----+
| marios123 | 123      |
+-----+
1 row in set (0.00 sec)
```

My Applications:

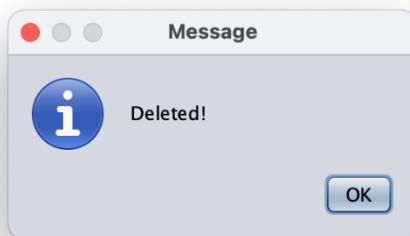
Οι αιτήσεις του συγκεκριμένου employee:

The screenshot shows a web application window titled "My Applications". The main content is a table with the following columns: id, start_d..., salary, position, edra, job_ev..., announ..., submis..., jobs_firm, and antikei... . There are two rows of data:

id	start_d...	salary	position	edra	job_ev...	announ...	submis...	jobs_firm	antikei...
1234	2003-...	1350.8	tamias	patra	kostasco	2003-...	2030-...	12345...	oikono...
5432	2030-...	7890.5	boh8os	patra	kostasco	2001-...	2029-...	12345...	gramat...

Below the table, there is a search input field containing "1234" and a dropdown arrow, a "Delete" button, and a "Home Page" button.

Απόσυρση μιας αίτησης:



Ο πίνακας με τις αιτήσεις ανανεώνεται αυτόματα μόλις αφαιρεθεί μια αίτηση (η θέση που αφαιρείται μεταφέρεται αυτόματα στον πίνακα με τις διαθέσιμες θέσεις):

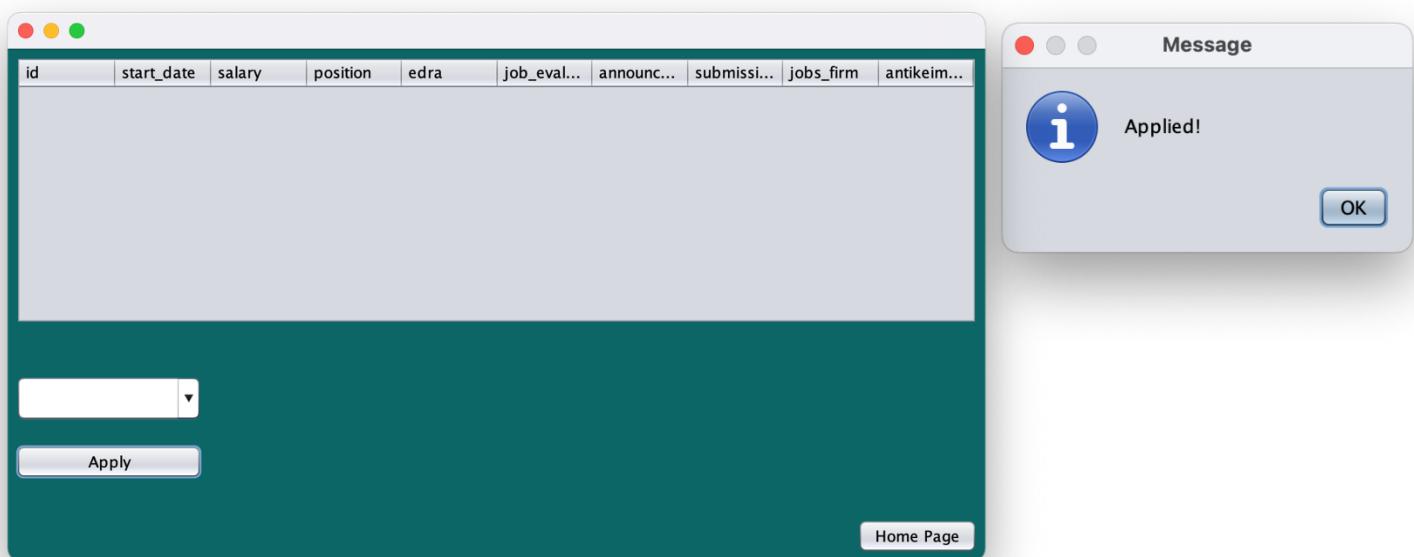
The screenshot shows a web application window titled "My Applications". At the top, there is a table with columns: id, start_d..., salary, position, edra, job_ev..., announ..., submis..., jobs_firm, and antikei... . A single row is visible: id 5432, start_d... 2030-..., salary 7890.5, position boh8os, edra patra, job_ev... kostasco, announ... 2001-..., submis... 2029-..., jobs_firm 12345..., and antikei... gramat... . Below the table is a search bar containing "5432" with a dropdown arrow icon. To the right of the search bar are two buttons: "Delete" and "Home Page".

New Application:

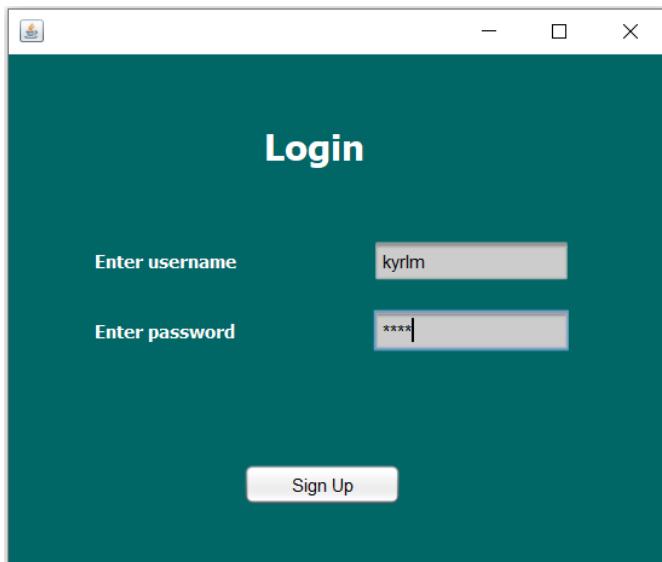
Οι διαθέσιμες θέσεις της εταιρείας που μπορεί να κάνει αίτηση:

The screenshot shows a web application window titled "New Application". At the top, there is a table with columns: id, start_date, salary, position, edra, job_eval..., announc..., submissi..., jobs_firm, and antikeim... . A single row is visible: id 1234, start_date 2003-0..., salary 1350.8, position tamias, edra patra, job_eval... kostasco, announc... 2003-0..., submissi... 2030-0..., jobs_firm 123456..., and antikeim... oikonomi... . Below the table is a search bar containing "1234" with a dropdown arrow icon. To the right of the search bar are two buttons: "Apply" and "Home Page".

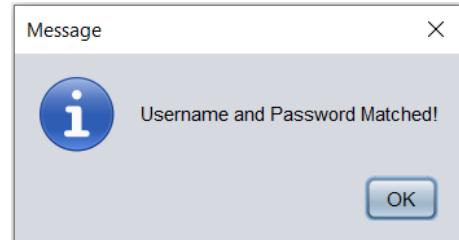
Όταν κάνει την αίτηση εμφανίζεται κατάλληλο μήνυμα και ενημερώνεται αυτόματα ο πίνακας (αποσύρεται η θέση εργασίας και εμφανίζεται πλέον στις αιτήσεις του employee):



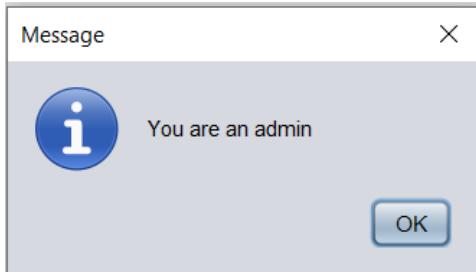
Admin:



Αν συνδεθώ με σωστά δεδομένα, τότε θα πάρω το κατάλληλο μήνυμα ότι ο το username και ο κωδικός ταίριαζαν με τα δεδομένα της βάσης, οπότε ο χρήστης αυτός υπάρχει :

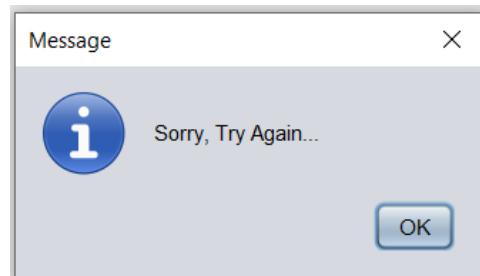


Επίσης μου εμφανίζεται και το μήνυμα:



Το οποίο μου δείχνει τι χρήστης είμαι.

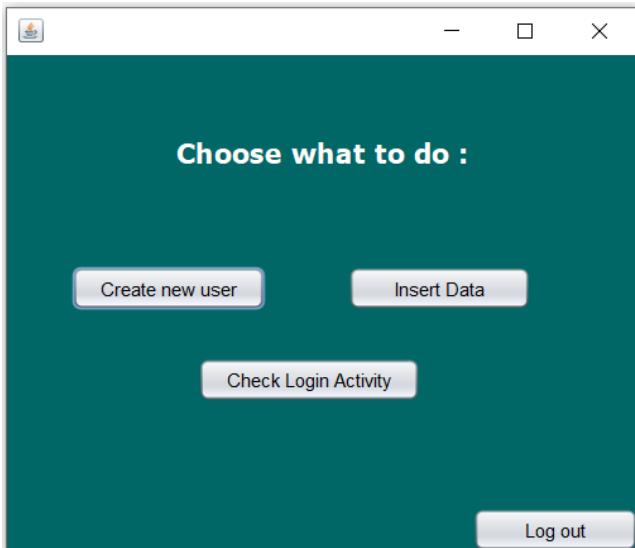
Σε περίπτωση που βάλω λάθος username ή κωδικό, τότε θα πάρω το μήνυμα:



Κοιτάζοντας στην βάση βλέπω ότι όντως ο χρήστης kyrlm είναι admin αφού στο πεδίο is_admin έχει την τιμή yes.

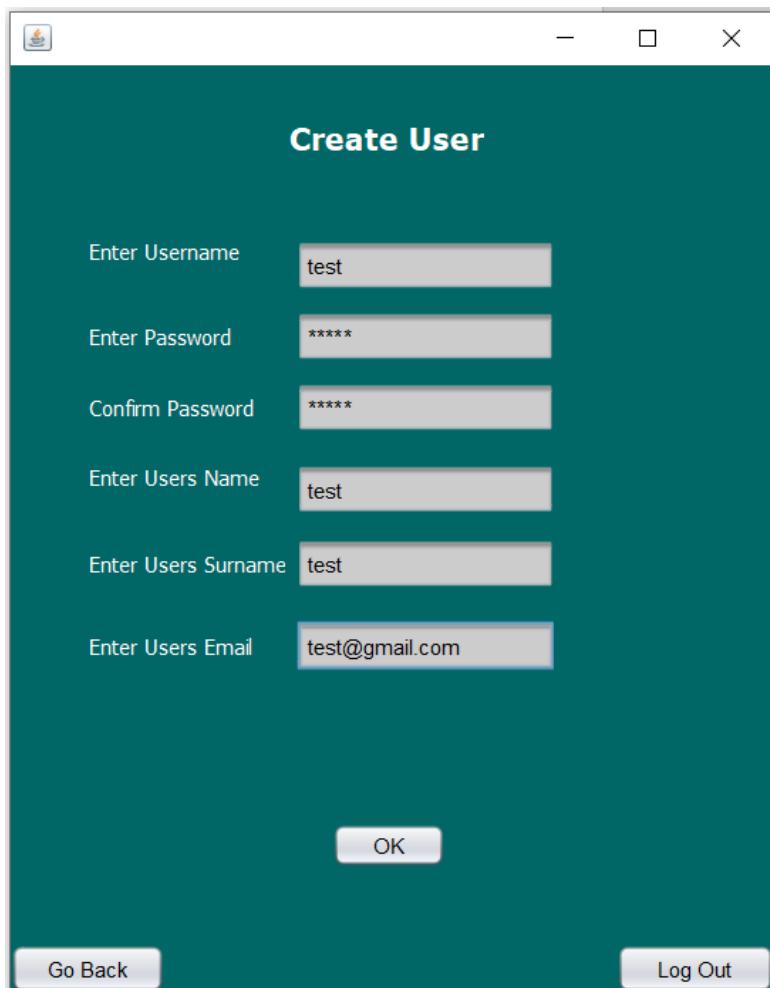
username	password	user_name	user_surname	reg_date	email	is_admin
euridik	434343	euridikh	petropoulou	1996-06-05 09:45:32	euridik@gmail.com	no
fotisre	2222	fotis	oikonomoy	2021-01-06 13:13:13	fotisre@gmail.com	no
jim12	123	dimitris	karamanos	2015-03-30 17:21:01	jim12@gmail.com	no
johnny	1234567890	giannhs	giannopoulos	1996-12-24 14:32:54	johnny@gmail.com	no
kostasco	123456	kostas	papanikolaoy	2009-05-14 03:45:32	kostasco@gmail.com	no
kyrlm	1234	kyriaki	lioumi	2020-12-19 12:40:00	kiriaki123@gmail.com	yes

Εδώ έχουμε την αρχική σελίδα του admin:



Όπου έχει 3 επιλογές.(Εδώ θα αναλυθούν οι δύο και η τελευταία θα αναλυθεί στο 2^o μέρος του gui)

Επιλέγοντας το κουμπί **Create new user** έχω :

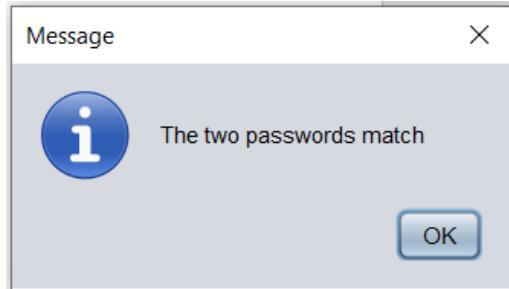


Οπού εισάγοντας τα στοιχεία που θέλω και πατώντας οκ δημιουργείται νέος χρήστης στη βάση.

Αν το username υπάρχει ήδη τότε θα βγει error και ο χρήστης δεν θα δημιουργηθεί.

Αν το email που βάλω έχει την λάθος μορφή ενεργοποιείται trigger, το οποίο πετάει error.

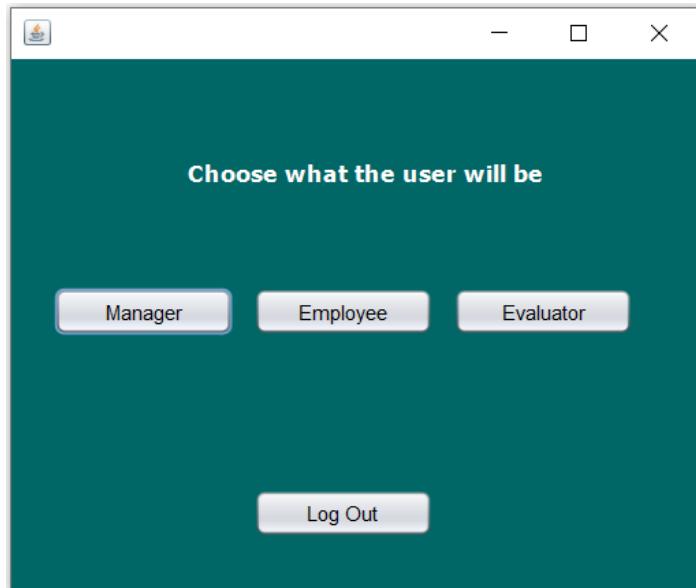
Πατώντας το κουμπί “ok” δημιουργείται ο χρήστης. Μένει όμως να επιλέξουμε τι είδους χρήστης θα είναι.
Πρώτα μας εμφανίζονται τα μηνύματα:



και

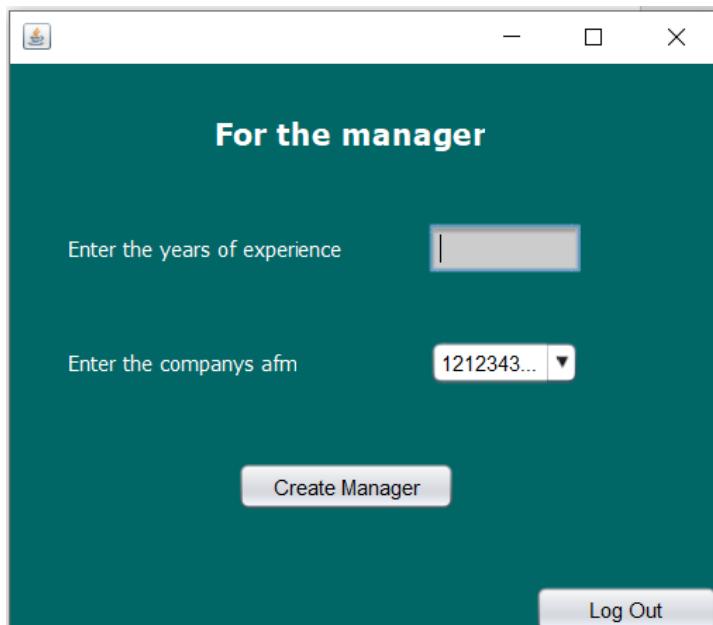
Σε περίπτωση που δεν ταιριάζουν οι δύο κωδικοί μεταξύ τους θα έχουμε error.

Έπειτα εμφανίζεται το παρακάτω παράθυρο:

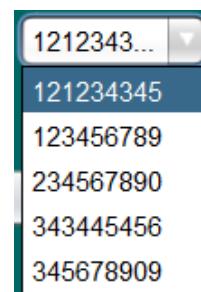


Όπου επιλέγουμε τι θα είναι ο νέος χρήστης.

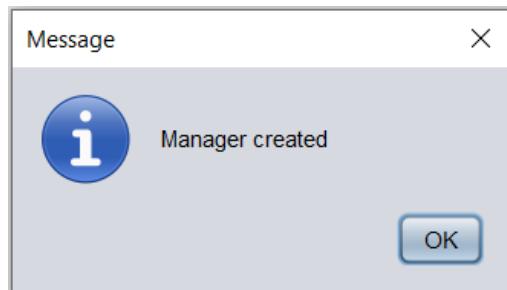
Αν επιλέξουμε manager :



Όπου το κουτί έχει τα afm των εταιριών, ώστε να επιλέξουμε σε ποια εταιρία θα είναι manager.



Εισάγοντας στο πεδίο κειμένου τα χρόνια εμπειρίας του manager, επιλέγοντας την εταιρία που θέλουμε και πατώντας “Create Manager ” δημιουργείται ο manager και παίρνουμε το συγκεκριμένο μήνυμα:



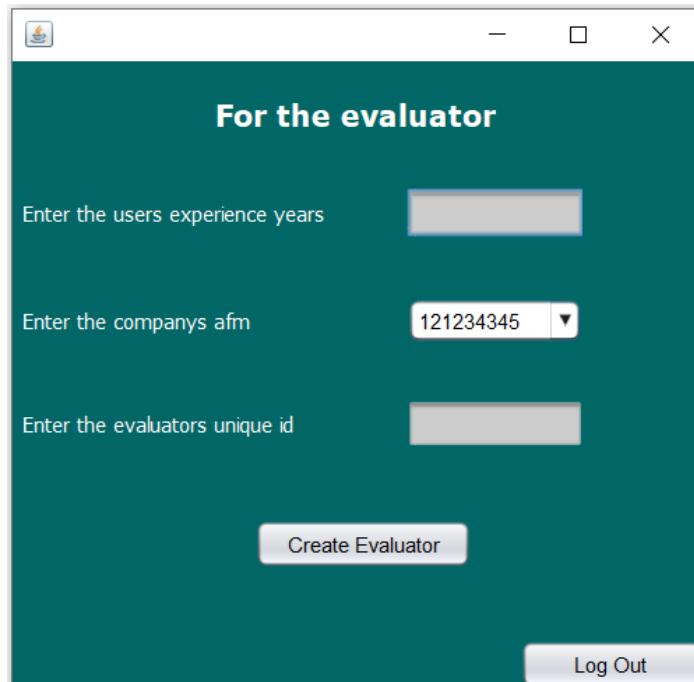
Και παρατηρούμε ότι δημιουργήθηκε ο manager στην βάση.

```
MariaDB [baseis]> SELECT * FROM manager;
+-----+-----+-----+
| manager_username | exp_years | firm      |
+-----+-----+-----+
| nefelaki          |      5 | 123456789 |
| fotisre           |      2 | 234567890 |
| euridik           |      1 | 345678909 |
| papad             |      2 | 121234345 |
| takis12           |      3 | 343445456 |
| test               |      7 | 121234345 |
+-----+-----+-----+
6 rows in set (0.000 sec)
```

Αν επιλέξουμε το κουμπί employee:

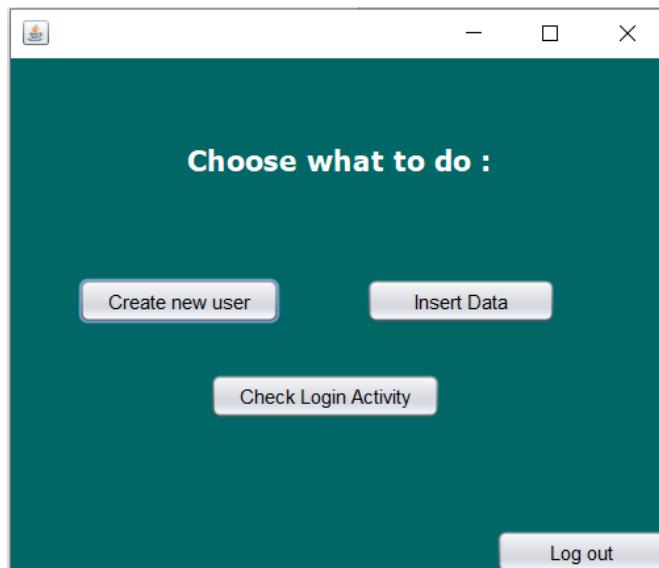
Ακολουθούμε την ίδια διαδικασία συμπληρώνοντας τις τιμές που μας ζητούνται και επιλέγοντας σε ποια εταιρία θα είναι ο employee και πατώντας το κουμπί Create Employee δημιουργείται ο employee στην βάση μας.

Αν επιλέξουμε το κουμπί evaluator:



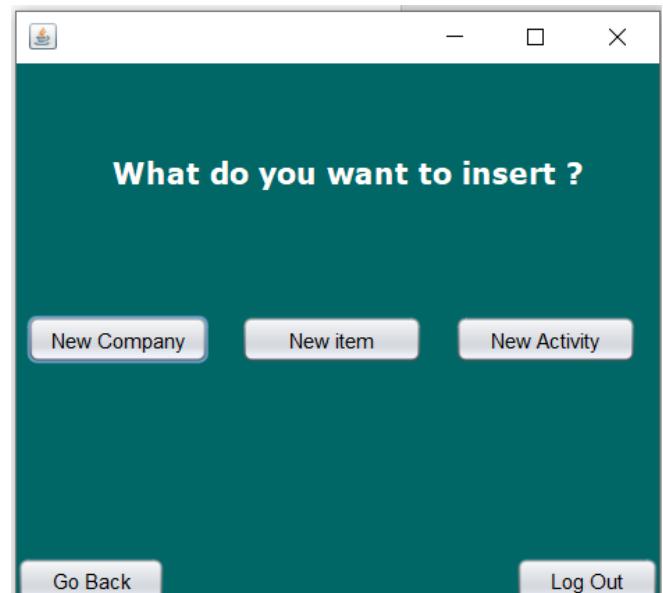
Ακολουθούμε την ίδια διαδικασία συμπληρώνοντας τις τιμές που μας ζητούνται και επιλέγοντας σε ποια εταιρία θα είναι ο evaluator και πατώντας το κουμπί Create Evaluator δημιουργείται ο evaluator στην βάση μας.

Αν επιλέγαμε στην αρχική σελίδα του admin το δεύτερο κουμπί (insert data)



Όπου μπορούμε να φτιάξουμε νέα εταιρία, νέο αντικείμενο και νέο project.

Θα μας εμφανιζόταν το εξής παράθυρο:



Επιλέγοντας το New Company:

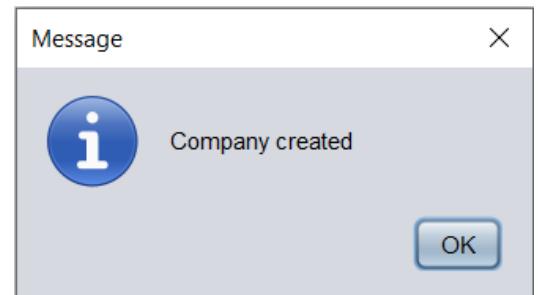
The window title is "Insert company data". It contains the following form fields:

AFM	001010011
DOY	C Patras
Company Name	TestCompany
Phone	2610989834
Street	korinthoy
Street's Number	2
City	patra
Country	ellada

Insert Company button at the bottom.

Buttons at the bottom left: Go Back. Buttons at the bottom right: Log Out.

Και εισάγοντας τα δεδομένα που θέλουμε και πατώντας το κουμπί Insert Company, η εταιρία δημιουργείται στην βάση και παίρνουμε το μήνυμα :



Βλέπουμε ότι η εταιρία μας δημιουργήθηκε :

```
MariaDB [baseis]> SELECT * FROM company;
```

afm	doy	company_name	phone	street	num	city	country
001010011	C Patras	TestCompany	2610989834	korinthoy	2	patra	ellada
121224245	A Patras	Patras	2106054672	Patras	45	Patras	Ellada

Επιλέγοντας το New Item στο προηγούμενο menu :

The dialog box is titled "Insert New Item". It contains the following fields:

- Title: test
- Description: to antikemeno einai ena test
- Belongs to: gramateia
- Company's afm: 001010011

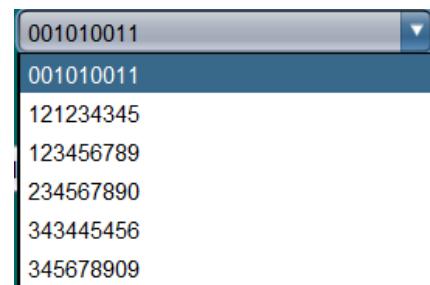
Buttons at the bottom include "Insert Item", "Go Back", and "Log Out".

Όπου στο πρώτο κουτί έχω την κατηγορία στην οποία μπορεί να ανήκει το νέο αντικείμενο:

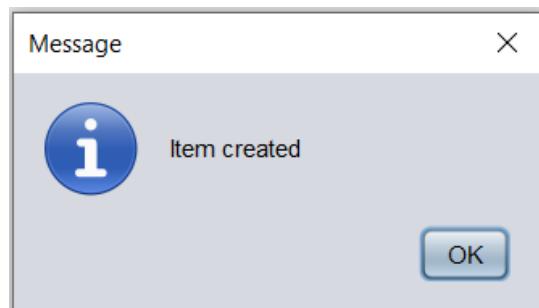


Και στο δεύτερο κουτί έχω τα afm των εταιριών στην οποία μπορεί να ανήκει το νέο αντικείμενο.

(παρατηρούμε ότι πλέον υπάρχει και η εταιρία με afm: 001010011 που φτιάξαμε παραπάνω)



Πατώντας το κουμπί Insert Item παίρνουμε το μήνυμα:



Και στην βάση δημιουργήθηκε το αντικείμενο:

MariaDB [baseis]> SELECT * FROM antikeim;			
title	descr	belongs_to	antikeim_firm
dhmosies sxeseis	NULL	NULL	121234345
gramateia	e3yphrethsh pelaton	NULL	123456789
hlektriko reyma	sosth leitoiorgia etairias	NULL	121234345
iatrikh	gia thn ygeia ton ergazomenon	NULL	234567890
marketing	gia proo8hsh proionton	NULL	123456789
oikonomika	NULL	NULL	345678909
test	to antikemeno einai ena test	gramateia	001010011
ypologistes	NULL	NULL	343445456

Επιλέγοντας το New Activity στο προηγούμενο menu :

Insert New Project

Employee for the project: jim12

Project description: to project ayto einai ena test

URL: https://test.com

Company's afm: 001010011

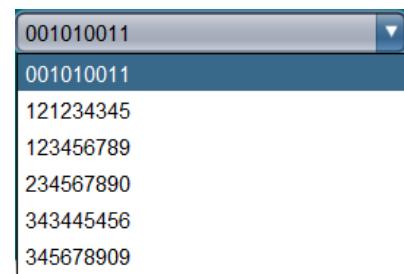
Insert Project

Go Back **Log Out**

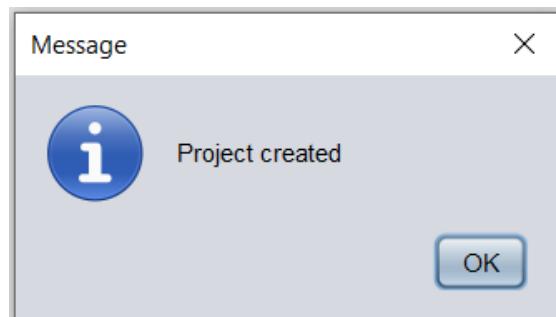
Όπου στο πρώτο κουτί έχω τα username των employees που μπορούν να αναλάβουν το project:



Και στο δεύτερο κουτί έχω τα afm των εταιριών, στις οποίες μπορεί να ανήκει το project:



Πατώντας το κουμπί Insert Project παίρνουμε το μήνυμα:



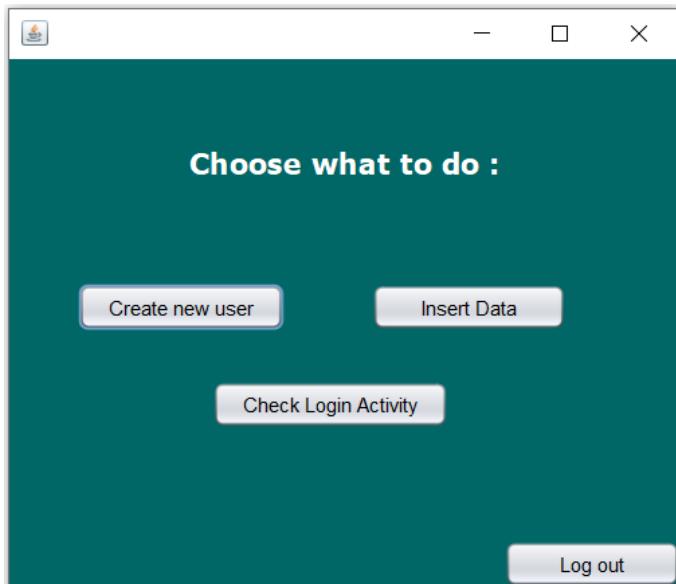
Σε περίπτωση που η τιμή του url δεν έχει την σωστή μορφή, ενεργοποιείται trigger και εμφανίζεται error.

Και στην βάση δημιουργήθηκε το project:

```
MariaDB [baseis]> SELECT * FROM project;
+-----+-----+-----+-----+-----+
| project_num | empl | project_descr | url | comp_afm |
+-----+-----+-----+-----+-----+
| 1 | marios123 | dhmioyrgia paixnidioy gia thn apasxolhsh sta diallelimata | https://paixnidi.com | 121234345 |
| 2 | nionios | kataskeyh neoy programmatos gia ton programmatismo ton meetings ths etairias | https://programm.gr | 123456789 |
| 3 | jim12 | to project ayto einai ena test | https://test.com | 001010011 |
+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)
```

Σε κάθε περίπτωση όταν ολοκληρώνεται μια ενέργεια, γυρνάμε αυτόματα στην αρχική σελίδα του χρήστη που έχει συνδεθεί.

GUI 2^ο μέρος

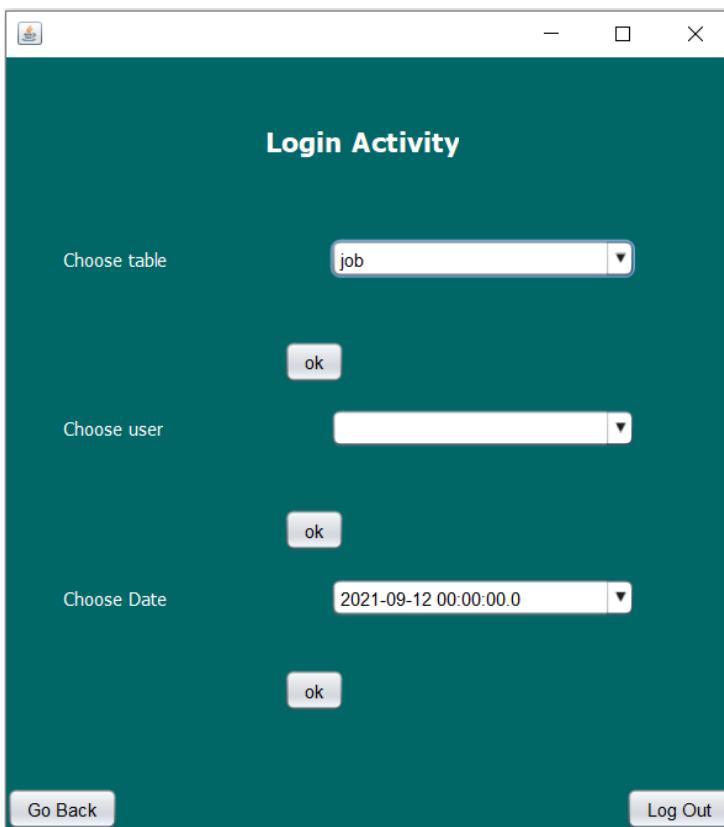


Έχοντας συνδεθεί ως admin έχουμε την επιλογή να πατήσουμε και το τρίτο κουμπί :

Check Login Activity

Όπου μπορούμε να δούμε πίνακες με το ιστορικό των login που έχουν κάνει άλλοι χρήστες, οι οποίοι έχουν κάνει insert, update ή delete στα tables: job,employee ή requestsevaluation.

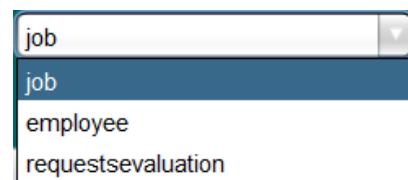
Οδηγούμαστε στο παράθυρο:



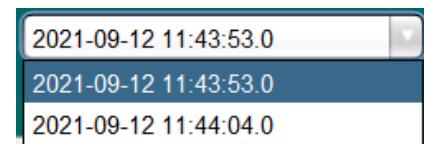
Όπου έχουμε την δυνατότητα να επιλέξουμε για ποιο table θέλουμε να δούμε τα login, για ποιόν user ή για ποια ημερομηνία.

Ανάλογα με το ποιο κουμπί θα πατήσουμε θα εμφανιστεί ένα πινακάκι με τα αντίστοιχα δεδομένα.

Στο πρώτο κουτί έχουμε τους πίνακες:



Στο τρίτο κουτί έχουμε τις ημερομηνίες:



Στο δεύτερο κουτί δεν έχουμε τίποτα:



Δεν μπορέσαμε να βρούμε πως θα βάλουμε στο trigger το όνομα του χρήστη που κάνει κάθε φορά την ενέργεια.

Οπότε δεν έχουμε και καθόλου usernames στα πινακάκια.

Παράδειγμα 1 : Αν επιλέξουμε το job προκύπτει το πινακάκι:

login_username	symban	epityxia	eidos	pinakas
2021-09-12 1...	true	update	job	

Παράδειγμα 2: Αν επιλέξουμε μια ημερομηνία προκύπτει το πινακάκι:

login_username	symban	epityxia	eidos	pinakas
2021-09-12...	true	update	employee	

Γενικά Σχόλια:

Τα προγράμματα που χρησιμοποιήσαμε είναι :

- xampp για την βάση
- netbeans για το gui

Επιπλέον για το gui χρειάστηκε και κατεβάσαμε το rs2xml.jar



Το οποίο μας βοήθησε με την εισαγωγή δεδομένων σε πίνακες.

Πως εργαστήκαμε :

Καθώς είμαστε ομάδα 2 ατόμων, χωρίσαμε τα πάντα στην μέση και στο πρώτο μέρος με την δημιουργία της βάσης και στο δεύτερο μέρος με το gui. Στο δεύτερο μέρος, ο ένας ανέλαβε τους χρήστες employee και evaluator και ο άλλος τους χρήστες manager και admin. Ανά χρονικά διαστήματα βρισκόμασταν και συζητούσαμε τι έχουμε κάνει, ανταλλάσσαμε απορίες και ενώναμε τα κομμάτια.

Τι δεν λειτουργεί:

Δεν έχουμε τα username των χρηστών στον πίνακα login, όπως περιγράφεται και παραπάνω.