



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΥΣΤΗΜΑ ΑΝΑΘΕΣΗΣ ΠΤΥΧΙΑΚΩΝ ΕΡΓΑΣΙΩΝ



ΣΥΣΤΗΜΑ ΑΝΑΘΕΣΗΣ ΠΕ



TEST

RUN

RESET

LOGOUT

Του φοιτητή
Καγιά Γρηγόρη

Αρ. Μητρώου: 09/3520

Επιβλέπων Καθηγητής
κ. Αμανατιάδης Δημήτριος

Θεσσαλονίκη 2014

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία έχει σαν στόχο την αυτοματοποίηση της διαδικασίας ανάθεσης πτυχιακών εργασιών για το τμήμα Μηχανικών Πληροφορικής. Μέχρι και το προηγούμενο ακαδημαϊκό εξάμηνο η διαδικασία των αναθέσεων ΠΕ μέσω διαγωνισμού γίνονταν με έναν ιδιαίτερα χρονοβόρο και επίπονο για την επιτροπή ΠΕ τρόπο.

Η ολοκλήρωση της παρούσας πτυχιακής δίνει ένα σημαντικό εργαλείο στα χέρια της επιτροπής ΠΕ, για την πλήρη αυτοματοποίηση της διαδικασίας των αναθέσεων. Μέσω της διαδικτυακής εφαρμογής που δημιουργήθηκε και απλού interface ο διαχειριστής του συστήματος ελέγχει αν έχουν τοποθετηθεί τα αναγκαία δεδομένα για να τρέξει η εφαρμογή (δεδομένα που αποθηκεύτηκαν από τη φάση των δηλώσεων) και κάνει τις αναθέσεις με το πάτημα ενός κουμπιού. Με το πέρας εκτέλεσης του αλγορίθμου, η εφαρμογή δίνει ως output αρχεία αποτελεσμάτων, στατιστικά στοιχεία, καθώς και αρχεία πληροφοριών που συγκεντρώθηκαν κατά την εκτέλεση του αλγορίθμου.

ΠΕΡΙΛΗΨΗ

Τα κεφάλαια που θα ακολουθήσουν έχουν ως στόχο να προσφέρουν μια ολοκληρωμένη εικόνα ως προς το τι ασχολήθηκε η πτυχιακή αυτή, τι προβλήματα αντιμετωπίζει καθώς και πια λύνει, ανάλυση της μεθοδολογίας με την οποία συλλέγει – επεξεργάζεται – παρουσιάζει τις πληροφορίες, τεχνικά χαρακτηριστικά της εφαρμογής και επεξήγηση κομματιών κώδικα σε σημεία που κρίθηκε απαραίτητο. Γίνεται αναφορά τόσο σε θεωρητικές έννοιες που αφορούν το σύστημα, όσο και σε πρακτικά θέματα που απαιτήθηκαν για την πραγματοποίηση αυτής της πτυχιακής εργασίας.

ABSTRACT

The following chapters have a dual purpose: on one hand, they seek to present a complete analysis of this dissertation's objective; on the other hand, our aim was to cover the entire gamut of issues that the presented application resolves. In addition, we present the methodology of collecting, processing and analyzing the subsequent data, the technical characteristics of its application and, when deemed necessary, we have included portions of the code used. There are references for the theoretical concepts of the system and also for the practical issues required for this thesis.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT	4
ΠΕΡΙΕΧΟΜΕΝΑ	5
ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ.....	6
ΕΙΣΑΓΩΓΗ.....	8
ΚΕΦΑΛΑΙΟ 1 ^ο	10
ΥΠΑΡΧΟΥΣΑ ΔΙΑΔΙΚΑΣΙΑ & ΑΝΑΓΚΗ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗΣ	10
ΕΙΣΑΓΩΓΗ.....	10
ΥΠΑΡΧΟΥΣΑ ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΘΕΣΗΣ ΠΕ	10
ΑΝΑΓΚΗ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗΣ.....	13
ΕΠΙΛΟΓΟΣ.....	13
ΚΕΦΑΛΑΙΟ 2 ^ο	14
ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΟΥ	14
ΕΙΣΑΓΩΓΗ.....	14
ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΘΕΣΗΣ ΠΤΥΧΙΑΚΩΝ ΕΡΓΑΣΙΩΝ.....	14
ΒΗΜΑ – ΒΗΜΑ ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	16
ΜΕΡΟΣ Α: ΔΙΑΔΙΚΑΣΙΕΣ ΠΡΙΝ ΑΡΧΙΣΟΥΝ ΟΙ ΑΝΑΘΕΣΕΙΣ	16
ΜΕΡΟΣ Β: ΔΙΑΔΙΚΑΣΙΕΣ ΑΝΑΘΕΣΕΩΝ	18
ΕΠΙΛΟΓΟΣ.....	21
ΚΕΦΑΛΑΙΟ 3 ^ο	22
ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	22
ΕΙΣΑΓΩΓΗ.....	22
ΤΟ ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ & Η ΒΑΣΙΚΗ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ	22
ΕΙΚΟΝΙΔΙΟ «TEST».....	23
ΕΙΚΟΝΙΔΙΟ «RUN»	24
ΕΙΚΟΝΙΔΙΟ «RESET».....	25
ΕΠΙΛΟΓΟΣ.....	26
ΚΕΦΑΛΑΙΟ 4 ^ο	27
ΤΕΧΝΟΛΟΓΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	27
ΕΙΣΑΓΩΓΗ.....	27

ΠΛΑΤΦΟΡΜΑ ΥΛΟΠΟΙΗΣΗΣ – MICROSOFT WEBMATRIX	27
ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ WebMatrix	28
ΔΥΝΑΤΟΤΗΤΕΣ	28
ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ – PHP	29
ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ PHP	30
ΧΡΗΣΗ ΤΗΣ PHP	30
ΠΛΕΟΝΕΚΤΗΜΑΤΑ - ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ PHP	31
ΑΝΑΚΤΗΣΗ & ΕΠΕΞΕΡΓΑΣΙΑ ΠΛΗΡΟΦΟΡΙΩΝ – MySQL	32
ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ	32
ΔΥΝΑΤΟΤΗΤΕΣ & ΠΕΡΙΟΡΙΣΜΟΙ	33
ΕΠΙΛΟΓΟΣ.....	34
ΚΕΦΑΛΑΙΟ 5 ^ο	35
ΕΠΙΛΟΓΟΣ.....	35
ΕΙΣΑΓΩΓΗ.....	35
ΣΗΜΑΝΤΙΚΟΤΗΤΑ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗΣ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΑΝΑΘΕΣΕΩΝ	35
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	36
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	37
ΠΑΡΑΡΤΗΜΑΤΑ.....	38
ΠΑΡΑΡΤΗΜΑ 1.....	38
ΠΑΡΑΡΤΗΜΑ 2.....	42
ΠΑΡΑΡΤΗΜΑ 3.....	44
ΠΑΡΑΡΤΗΜΑ 4.....	48
ΠΑΡΑΡΤΗΜΑ 5.....	51
ΠΑΡΑΡΤΗΜΑ 6.....	53
ΠΑΡΑΡΤΗΜΑ 7.....	56
ΠΑΡΑΡΤΗΜΑ 8.....	57
ΠΑΡΑΡΤΗΜΑ 9.....	57

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1 Αρχική οθόνη του συστήματος.....	22
---	----

Εικόνα 2	Μήνυμα πετυχημένης εισαγωγής δεδομένων και παρουσίαση στατιστικών στοιχείων	23
Εικόνα 3	Μήνυμα αποτυχημένης εύρεσης δεδομένων	24
Εικόνα 4	Μήνυμα μετά την εκτέλεση του αλγορίθμου.....	25
Εικόνα 5	Ενημερωτικό μήνυμα μετά την επαναφορά του συστήματος στην αρχική του κατάσταση.....	25
Εικόνα 6	Microsoft WebMatrix.....	27
Εικόνα 7	Γλώσσα προγραμματισμού PHP	29
Εικόνα 8	Γλώσσα ερωτοαποκρίσεων MySQL	32

ΕΙΣΑΓΩΓΗ

Η πτυχιακή εργασία είναι μια από τις σημαντικότερες παραμέτρους στις σπουδές ενός φοιτητή στην τριτοβάθμια εκπαίδευση, καθώς ενσωματώνει όχι μόνο τις γνώσεις που έχει αποκτήσει κατά τη διάρκεια της φοίτησής του, αλλά και την ικανότητα στο να συνδυάζει και να συνεργάζεται, αποκομίζοντας μια σημαντική εμπειρία. Η παρούσα πτυχιακή εργασία δημιουργήθηκε με σκοπό να αυτοματοποιήσει την διαδικασία ανάθεσης πτυχιακών εργασιών για το τμήμα Μηχανικών Πληροφορικής του Αλεξάνδρειου Τ.Ε.Ι. Θεσσαλονίκης, δημιουργώντας ένα εργαλείο για την επιτροπή ΠΕ του τμήματος, το οποίο αναλαμβάνει τη διαδικασία ανάθεσης θεμάτων στους φοιτητές όσο γίνεται δικαιότερα, λαμβάνοντας υπόψη το θεσμό της πτυχιακής εργασία στο τμήμα, αλλά και τις ξεχωριστές ανάγκες του κάθε φοιτητή που αιτείται για ανάληψη θέματος.

Στα κεφάλαια που θα ακολουθήσουν γίνεται προσπάθεια να επεξηγηθούν όσο γίνεται καλύτερα έννοιες στενά συνδεδεμένες με το σύστημα που αναπτύχθηκε, τόσο σε θεωρητικό, όσο και σε πρακτικό επίπεδο. Σε θεωρητικό επίπεδο γίνεται σημαντική αναφορά στην διαδικασία που εφαρμόζει η επιτροπή ΠΕ για την πραγματοποίηση των αναθέσεων, αλλά και με λεπτομερή ανάλυση της μεθοδολογίας πάνω στην οποία βασίζονται τα αποτελέσματα που εξάγονται από το σύστημα που αναπτύχθηκε. Η μεθοδολογία παρουσιάζεται σε μορφή ψευδοκώδικα, με τις αντίστοιχες παραπομπές στο παράρτημα, σε σημεία που θα χρειαστεί μια πιο προσεκτική ματιά. Σε τεχνικό επίπεδο, παρουσιάζεται το γραφικό περιβάλλον που δημιουργήθηκε για τον έλεγχο του συστήματος από τον διαχειριστή του, αλλά και αναφορά στα προγραμματιστικά εργαλεία που βασίστηκε η πτυχιακή αυτή.

Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στην παρούσα διαδικασία ανάθεσης ΠΕ που χρησιμοποιεί η επιτροπή, πως προκύπτει η ανάγκη αυτοματοποίησης της διαδικασίας αυτής, καθώς και τα προβλήματα που λύνει. Επίσης, αναλύει τους λόγους για τους οποίους θεωρείται σημαντική η προσπάθεια για αυτοματοποίηση της διαδικασίας αναθέσεων.

Στο επόμενο κεφάλαιο, υπάρχει αναλυτική, βήμα προς βήμα, επεξήγηση της μεθοδολογίας – αλγόριθμου στον οποίο βασίζεται το σύστημα και κατ'

επέκταση πραγματοποιούνται οι αναθέσεις και παράγονται όλα τα αποτελέσματα. Στα σημεία που κρίθηκε απαραίτητο υπάρχουν οι αντίστοιχες παραπομπές στο παράρτημα με τα κομμάτια κώδικα του κάθε σημείου.

Το τρίτο κεφάλαιο, ασχολείται με το περιβάλλον της εφαρμογής, τις δυνατότητες που αυτό προσφέρει στον χρήστη/διαχειριστή του συστήματος καθώς και τα παραγόμενα από αυτό αποτελέσματα. Στο τέταρτο κεφάλαιο, παρουσιάζονται τα βασικά μέρη των προγραμματιστικών εργαλείων και του περιβάλλοντος που χρησιμοποιήθηκε για τη δόμηση της εφαρμογής.

Τέλος, στο πέμπτο και τελευταίο κεφάλαιο της πτυχιακής αυτής, αποτυπώνονται κάποια συμπεράσματα και σκέψεις από την χρήση του συστήματος, επισήμανση των πλεονεκτημάτων που προσφέρει και κάποιες τελευταίες λεπτομέρειες που κρίθηκαν σημαντικό να καταγραφούν.

ΚΕΦΑΛΑΙΟ 1^ο

ΥΠΑΡΧΟΥΣΑ ΔΙΑΔΙΚΑΣΙΑ & ΑΝΑΓΚΗ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗΣ

ΕΙΣΑΓΩΓΗ

Το πρώτο κεφάλαιο περιλαμβάνει μια περιγραφή της υπάρχουσας διαδικασίας ανάθεσης πτυχιακών εργασιών που χρησιμοποιεί η επιτροπή του τμήματος για τις δηλώσεις θεμάτων που γίνονται μέσω διαγωνισμού. Επίσης, υπογραμμίζει και την αναγκαιότητα της αυτοματοποίησης της υπάρχουσας διαδικασίας.

ΥΠΑΡΧΟΥΣΑ ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΘΕΣΗΣ ΠΕ

Η διαδικασία για την ανάθεση θεμάτων μέσω του διαγωνισμού που διεξάγει το τμήμα για την ανάθεση πτυχιακών εργασιών στους φοιτητές, περιελάμβανε αρκετό φόρτο, στον μεγαλύτερο βαθμό χειρονακτικό, δημιουργία αρκετών αρχείων και μια διαδικασία που μπορεί να κρατούσε έως και μερικές μέρες.

Στην αρχή κάθε εξαμήνου ανακοινώνεται από τους εισηγητές θεμάτων ΠΕ η έναρξη ανάρτησης θεμάτων με συγκεκριμένη ημερομηνία λήξης και αριθμό θεμάτων για διαγωνισμό. Αυτό που έχουν να κάνουν οι εισηγητές είναι να αναρτήσουν στην Ύδρα αρχείο με τις λεπτομέρειες της πτυχιακής εργασίας. Κατά τη διαδικασία ανάρτησης θεμάτων, η επιτροπή ΠΕ ελέγχει το κάθε θέμα για την ύπαρξη τουλάχιστον 2 προαπαιτούμενων μαθημάτων. Σε αυτό το σημείο, οι πληροφορίες που καταγράφονταν ήταν:

1. Αρχείο με το πόσα θέματα αναρτήθηκαν από τον κάθε καθηγητή και το είδος της ανάρτησης (απευθείας ανάθεση – διαγωνισμός)
2. Αρχείο με τα θέματα που δόθηκαν με απευθείας ανάθεση, καθώς και τα στοιχεία των φοιτητών που τις αναλαμβάνουν
3. Αρχείο με τα διαθέσιμα θέματα για τον διαγωνισμό, τους κωδικούς των προαπαιτούμενων που έχουν οριστεί και τον τύπο του θέματος (ομαδική – ατομική)

Μετά το πέρας της προθεσμίας για την ανάρτηση θεμάτων από τους καθηγητές, ανακοινώνεται στους φοιτητές η έναρξη της διαδικασίας δήλωσης των προτιμήσεών τους. Ο κάθε φοιτητής έχει το δικαίωμα να δηλώσει μέχρι και πέντε (5) θέματα, με σειρά προτίμησης και τα οποία επισυνάπτει σε αρχείο το οποίο περιέχει επίσης την αναλυτική βαθμολογία του από το σύστημα της ΠΥΘΙΑΣ, καθώς και τα απαραίτητα στοιχεία επικοινωνίας.

Μετά το τέλος της διορίας για δήλωση θεμάτων από τους φοιτητές, η οποία γινόταν με αποστολή e-mail, άρχισε η διαδικασία συλλογής ενός πολύ μεγάλου όγκου πληροφοριών. Ένας όγκος πληροφοριών ο οποίος δεν αφορούσε μόνο τις προτιμήσεις των φοιτητών, όπως αυτές αναγραφόταν στα e-mail που έπρεπε να στείλουν, αλλά και αρχεία με πληροφορίες που αφορούσε τους φοιτητές (αριθμός εξαμήνου, μέσος όρος μαθημάτων, συντελεστής προόδου) κτλ. Μόλις τελείωνε η συλλογή αυτών των πληροφοριών, ξεκινούσε η διαδικασία δημιουργίας των αρχείων βάση των οποίων θα γινόνταν οι αναθέσεις.

Στο επόμενο στάδιο και αφού πλέον η επιτροπή είχε τις απαραίτητες πληροφορίες για να πάρει τις αποφάσεις αναθέσεων, δημιουργούνταν οι κατάλληλοι δείκτες, στους οποίους θα βασιζόταν για να αναθέσει τα θέματα πτυχιακών εργασιών. Όλες αυτές τις πληροφορίες θα τις χρησιμοποιούσε τελικά βάση ενός αλγορίθμου – μεθοδολογίας, ο οποίος έδινε τα κατάλληλα βήματα για την χρησιμοποίηση αυτών των πληροφοριών. Όπως είναι λογικό, ο αλγόριθμος υπέστη τις απαραίτητες μεταβολές έτσι ώστε να μπορεί να χρησιμοποιηθεί για τις ανάγκες αυτοματοποίησης της διαδικασίας.

Σημαντικό βήμα της προηγούμενης διαδικασίας ήταν η δημιουργία των απαραίτητων αρχείων που θα αποτύπωναν όλες εκείνες τις πληροφορίες που είχαν συγκεντρωθεί από τις δηλώσεις των φοιτητών, καθώς και τη δημιουργία των δεικτών βάση των οποίων θα γινόνταν οι αναθέσεις. Αρχικά, δημιουργείται ένα αρχείο excel που οι δηλώσεις θεμάτων ταξινομούνται ανά φοιτητή. Για κάθε φοιτητή καταγράφονται οι εξής πληροφορίες:

- Γενικές πληροφορίες που αφορούν τον φοιτητή, όπως ο αριθμός μητρώου του, το τρέχων και το εξάμηνο εισαγωγής, καθώς και οι τιμές του στους δείκτες του τμήματος (γενικός μέσος όρος, συντελεστής προόδου, περασμένες διδακτικές μονάδες)

- Πληροφορίες για το κάθε θέμα ξεχωριστά, όπως αριθμός προτίμησης του θέματος, τίτλος και εισηγητής, καθώς και οι βαθμολογίες στα προαπαιτούμενα μαθήματα

Στη συνέχεια δημιουργείται αρχείο excel το οποίο περιλαμβάνει τις δηλώσεις θεμάτων των φοιτητών ταξινομημένες ανά καθηγητή. Δημιουργείται ένα φύλλο για κάθε καθηγητή, με στόχο να οργανωθεί ο μεγάλος όγκος πληροφοριών όσο γίνεται καλύτερα. Για κάθε δήλωση φοιτητή καταγράφονται οι εξής πληροφορίες:

- Πληροφορίες που αφορούσαν το θέμα της πτυχιακής εργασίας, όπως ο τίτλος και ο εισηγητής του θέματος
- Πληροφορίες που αφορούσαν τον φοιτητή, όπως όνομα και επώνυμο, οι βαθμολογίες του στα προαπαιτούμενα μαθήματα της πτυχιακής, δείκτες του τμήματος, όπως γενικός μέσος όρος, συντελεστής προόδου από το σύστημα της ΠΥΘΙΑΣ και περασμένες διδακτικές μονάδες, εξάμηνο εισαγωγής και τρέχων εξάμηνο

Αφού πλέον έχουν δημιουργηθεί όλα τα απαραίτητα πληροφοριακά αρχεία και οι κατάλληλοι δείκτες ξεκινάει η διαδικασία ανάθεσης πτυχιακών εργασιών, όπου γίνεται προσπάθεια να δοθούν όσες περισσότερες γίνεται. Αρχικά δίνονται οποίες πτυχιακές είναι δηλωμένες από έναν μόνο φοιτητή/ομάδα στην πρώτη επιλογή προτίμησης και στη συνέχεια αρχίζει η ανάθεση θεμάτων που τα διεκδικούν δύο ή και περισσότεροι φοιτητές/ομάδες. Για κάθε ανάθεση που γίνεται αφαιρείται το όνομα του φοιτητή αυτού από την διεκδίκηση των υπολοίπων θεμάτων που είχε δηλώσει, έχοντας ως αποτέλεσμα την αναπροσαρμογή του πόσοι διεκδικούν το κάθε θέμα. Η διαδικασία αυτή συνεχίζεται μέχρις όπου να μην υπάρχει άλλο θέμα που να ζητείται από φοιτητή.

Ανάλογα με τον αριθμό των φοιτητών που έμειναν χωρίς θέμα στην πρώτη φάση δηλώσεων, η επιτροπή ΠΕ συγκεντρώνει τα θέματα που δεν ζητήθηκαν κατά την πρώτη φάση και ανακοινώνει εκ νέου νέα διορία δήλωσης θεμάτων για τους φοιτητές. Με τη λήξη και της δεύτερης προθεσμίας επαναλαμβάνεται η διαδικασία ανάθεσης και δημιουργείται το τελικό αρχείο αναθέσεων για το τρέχων εξάμηνο.

ΑΝΑΓΚΗ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗΣ

Όπως προκύπτει από την συνοπτική ανάλυση της υπάρχουσας διαδικασίας ανάθεσης πτυχιακών εργασιών, γίνεται εύκολα αντιληπτό ό,τι ο φόρτος που προέκυπτε από την όλη διαδικασία ήταν μεγάλος. Με την εφαρμογή που δημιουργήθηκε στα πλαίσια της πτυχιακής αυτής, λύνει αρκετά από τα προβλήματα που προκύπτουν από την ανάλυση του υπάρχοντος συστήματος, όπως:

1. Με όλη την απαιτούμενη πληροφορία συγκεντρωμένη στη βάση δεδομένων της εφαρμογής, δεν υπάρχει πλέον η ανάγκη συλλογής των email – δηλώσεων από τους φοιτητές και κυρίως η εξαγωγή του συνόλου της πληροφορίας από αυτά
2. Όλοι οι απαραίτητοι υπολογισμοί για τη δημιουργία των δεικτών που χρειάζονται γίνονται από την εφαρμογή
3. Η χρονική διάρκεια που χρειάζεται για να γίνουν οι αναθέσεις, από κάποιες μέρες που απαιτούνταν με το χειροκίνητο σύστημα, έχει μειωθεί στο ελάχιστο δυνατό, μικρότερο του ενός λεπτού.
4. Δυναμική δημιουργία αρχείων. Η εφαρμογή μετά το πέρας της λειτουργίας της δημιουργεί τα εξής αρχεία:
 - Προτιμήσεις πτυχιακών εργασιών ανά τίτλο θέματος
 - Προτιμήσεις πτυχιακών εργασιών ανά φοιτητή
 - Αποτελέσματα αναθέσεων
 - Πτυχιακές που δεν ζητήθηκαν από κανέναν φοιτητή
 - Πτυχιακές οι οποίες δεν δόθηκαν

ΕΠΙΛΟΓΟΣ

Αρχικά, έγινε μια εισαγωγή στο μέχρι τώρα σύστημα ανάθεσης πτυχιακών εργασιών που χρησιμοποιούσε η επιτροπή ΠΕ και συνεχίζει με μια αναλυτική παρουσίαση του. Κλείνοντας, τονίζεται η σημασία που έχει η αυτοματοποίηση της διαδικασίας αναθέσεων, καθώς και τα πλεονεκτήματα και τα αποτελέσματα που προσφέρει.

ΚΕΦΑΛΑΙΟ 2^ο

ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΟΥ

ΕΙΣΑΓΩΓΗ

Σκοπός του κεφαλαίου που ακολουθεί είναι η αναλυτική παρουσίαση του αλγόριθμου, βάση του οποίου, το σύστημα που αναπτύχθηκε πραγματοποιεί τις αναθέσεις. Γίνεται βήμα προς βήμα επεξήγηση της μεθοδολογίας, σε μορφή ψευδοκώδικα.

ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΘΕΣΗΣ ΠΤΥΧΙΑΚΩΝ ΕΡΓΑΣΙΩΝ

Ο αλγόριθμος – μεθοδολογία που χρησιμοποιήθηκε ουσιαστικά αποτελεί την μετεξέλιξη του ήδη υπάρχοντος αλγορίθμου που χρησιμοποιούσε η επιτροπή ΠΕ για να πραγματοποιεί τις αναθέσεις. Όπως είναι λογικό, έχει υποστεί τις κατάλληλες μετατροπές έτσι ώστε να μπορεί να χρησιμοποιηθεί προγραμματιστικά. Ο αλγόριθμος χωρίζεται ουσιαστικά σε δύο μέρη:

- διαδικασίες οι οποίες είναι απαραίτητο να πραγματοποιηθούν πριν αρχίσουν οι αναθέσεις
- διαδικασίες ανάθεσης

Οι διαδικασίες πριν αρχίσουν οι αναθέσεις, ουσιαστικά περιλαμβάνουν τους απαραίτητους υπολογισμούς για τη δημιουργία των δεικτών πάνω στους οποίους θα βασιστούν οι αποφάσεις για την ανάθεση της κάθε πτυχιακής εργασίας. Συνοπτικά, οι διαδικασίες αυτές είναι:

- υπολογισμός μέσου όρου των βαθμολογιών προαπαιτούμενων μαθημάτων για την κάθε πτυχιακή
- υπολογισμός συνολικής βαθμολογίας για την κάθε δήλωση, σε συνάρτηση με τον ΜΟ των προαπαιτούμενων, δεικτών του τμήματος (γενικός ΜΟ,

συντελεστής προόδου, περάσμενες διδακτικές μονάδες) και άλλων χρήσιμων για την ανάθεση τιμών

- υπολογισμός ενός μέσου όρου βαθμολογιών για το κάθε θέμα, βάση των βαθμολογιών των φοιτητών που έχουν δηλώσει το κάθε θέμα
- με βάση τον ΜΟ του κάθε θέματος, αποκλείονται από την διεκδίκηση του φοιτητές που έχουν μέσο όρο βαθμολογίας μικρότερο από τον μέσο όρο του θέματος
- εντόπισμός για κάθε θέμα πόσοι φοιτητές το διεκδικούν

Όσον αφορά τις διαδικασίες που αφορούν τις αναθέσεις, ασχολούνται με την επεξεργασία των τιμών που έχουν υπολογιστεί στο πρώτο μέρος του αλγορίθμου. Επίσης, παράγουν τα αρχεία των αποτελεσμάτων, στην ολοκλήρωση εκτέλεσης του αλγορίθμου. Οι διαδικασίες είναι:

1. εντοπίζει και αναθέτει τις πτυχιακές εκείνες που έχουν δηλωθεί από έναν μόνο φοιτητή / ομάδα στην πρώτη προτίμηση
2. εντοπίζει και αναθέτει τις πτυχιακές εκείνες που έχουν δηλωθεί από δύο φοιτητές / ομάδες στις πρώτες τους προτιμήσεις
3. τρέχει τον κύριο αλγόριθμο αναθέσεων
 - εντοπίζει όλες τις διαθέσιμες ΠΕ
 - αναθέτει βάση συγκεκριμένων κριτηρίων
 - βάζει πτυχιακές σε αναμονή
4. μετά από κάθε ανάθεση επαναπροσδιορίζει την συνολική βαθμολογία για το κάθε θέμα
5. προσπαθεί να αναθέσει θέμα σε όσους περισσότερους φοιτητές γίνεται, γι' αυτό στο τέλος συγκεντρώνει θέματα που δεν ανατέθηκαν και φοιτητές που δεν πήραν θέμα και προσπαθεί να κάνει αναθέσεις βάση αυτών.

ΒΗΜΑ – ΒΗΜΑ ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

ΜΕΡΟΣ Α: ΔΙΑΔΙΚΑΣΙΕΣ ΠΡΙΝ ΑΡΧΙΣΟΥΝ ΟΙ ΑΝΑΘΕΣΕΙΣ

Το πρώτο μέρος του αλγορίθμου ασχολείται με διαδικασίες υπολογισμού κάποιων βασικών τιμών στις οποίες θα βασιστούν αργότερα οι αποφάσεις ανάθεσης της κάθε πτυχιακής. Πιο αναλυτικά, οι διαδικασίες για την εύρεση των τιμών αυτών είναι οι ακόλουθες:

A1. Υπολογισμός μέσου όρου προαπαιτούμενων μαθημάτων για το κάθε θέμα πτυχιακής που δήλωσαν οι φοιτητές

Ένας από τους πλέον σημαντικούς δείκτες καταλληλότητας του φοιτητή για να του ανατεθεί ή όχι ένα θέμα, είναι ο μέσος όρος που συγκεντρώνει στα προαπαιτούμενα μαθήματα των πτυχιακών που έχει δηλώσει. Με αυτή τη λογική, αυτή είναι η πρώτη διεργασία που αναλαμβάνει να εκτελέσει ο αλγόριθμος. Ο υπολογισμός του πεδίου αυτού αποτελείται από τρία βήματα:

Υπολογίζουμε τα null πεδία. Ο εισηγητής κάθε πτυχιακής εργασίας μπορεί να ορίσει μέχρι και τέσσερα προαπαιτούμενα σχετικά με το θέμα μαθήματα. Ο αλγόριθμος «βλέπει» στη βάση της εφαρμογής για το πόσα από τα πεδία που αφορούν τα προαπαιτούμενα έχουν μείνει κενά από τη φάση της ανάρτησης των θεμάτων, για να ξέρει με ποιο παρονομαστή θα πρέπει να υπολογιστεί ο μέσος όρος για το κάθε θέμα.

Υπολογισμός βαθμολογίας προαπαιτούμενων. Ξέροντας πλέον τον αριθμό μαθημάτων που αντιστοιχεί σε κάθε δήλωση, υπογίεται το άθροισμα των βαθμών των προαπαιτούμενων και με μια απλή διαίρεση υπολογίζεται ο μέσος όρος τους.

Ανάθεση τιμής σε κάθε δήλωση. Έχοντας υπολογίσει πλέον την ζητούμενη τιμή, με βάση το id της κάθε δήλωσης ανατίθενται στην κάθε μια ο μέσος όρος των προαπαιτούμενων της.

A2. Υπολογισμός βαθμολογίας για όλα τα θέματα

Στο επόμενο βήμα ο αλγόριθμος υπολογίζει τη βαθμολογία του κάθε θέματος. Η βαθμολογία αυτή είναι ένας σημαντικός δείκτης ως προς τις πιθανότητες που έχει ένας φοιτητής να του ανατεθεί θέμα. Η τιμή του δείκτη αυτού εξαρτάται από αρκετές τιμές, όπως μέσος όρος προαπαιτούμενων, σειρά προτίμησης της πτυχιακής, αλλά και δείκτες του τμήματος όπως συντελεστής προόδου, γενικός μέσος όρος και περασμένες διδακτικές μονάδες. Πιο αναλυτικά, ο τύπος υπολογισμού για την βαθμολογία κάθε δήλωσης:

$$\checkmark \text{ Μέσος_όρος_προαπαιτούμενων} + \text{ γενικός_ΜΟ} + (\text{διδακτικές_μονάδες}/21) + (\text{συντελεστής_προόδου}/100) + (11 - \text{αριθμός_προτίμησης}).$$

Όπως είναι λογικό για τιμές όπως διδακτικές μονάδες, συντελεστής προόδου και αριθμός προτίμησης γίνονται οι κατάλληλες αναγωγές, έτσι ώστε η τελική τιμή να κυμαίνεται σε όριο με άριστα το δέκα.

A3. Υπολογισμός συνολικής βαθμολογίας για το κάθε θέμα και αποκλεισμός δηλώσεων βάση αυτού

Σε αυτό το βήμα το σύστημα υπολογίζει τον μέσο όρο για το κάθε θέμα, προσμετρώντας την βαθμολογία που έχουν συγκεντρώσει όσοι έχουν δηλώσει την κάθε πτυχιακή εργασία. Στη συνέχεια έχοντας βρει τον μέσο όρο, αποκλείεται όποια δήλωση έχει βαθμολογία μικρότερη από το μέσο όρο του θέματος, βήμα το οποίο βοηθάει αρκετά στο να μειωθεί ο αριθμός φοιτητών που διεκδικούν το κάθε θέμα.

A4. Υπολογίζεται για κάθε θέμα πόσοι το διεκδικούν

Ξεροντας πλέον πόσοι ουσιαστικά διεκδικούν το κάθε θέμα, έχοντας κάνει τους αποκλεισμούς, υπολογίζεται πόσοι διεκδικούν το κάθε θέμα. Ένα βήμα που είναι αρκετά χρήσιμο και στο δεύτερο μέρος του αλγορίθμου, στις διαδικασίες αναθέσεων.

ΜΕΡΟΣ Β: ΔΙΑΔΙΚΑΣΙΕΣ ΑΝΑΘΕΣΕΩΝ

Στο δεύτερο μέρος της εκτέλεσης του αλγορίθμου είναι όπου δημιουργούνται τα αρχεία αποτελεσμάτων, πληροφοριών και στατιστικών στοιχείων. Από το πρώτο μέρος, έχουν δημιουργηθεί όλοι οι δείκτες που απαιτούνται για να παρθούν οι αποφάσεις αναθέσεων. Στις επόμενες γραμμές γίνεται προσπάθεια να γίνει ανάλυση των κομματιών που αφορούν τις αναθέσεις, όσο το δυνατόν αναλυτικότερα.

B1. Απομονώνονται οι ομαδικές πτυχιακές, με τις οποίες ο αλγόριθμος ασχολείται πρώτα. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα

Οι ομαδικές πτυχιακές είναι μια αρκετά ιδιάζουσα περίπτωση. Κατά τη διάρκεια δημιουργίας του αλγορίθμου και γενικότερα του συστήματος έγινε προσπάθεια να αντιμετωπιστούν με τον ίδιο τρόπο που αντιμετωπίζονται οι ατομικές πτυχιακές εργασίες, χωρίς αυτό όμως να μπορέσει να καταστεί δυνατό. Ο κυριότερος λόγος έχει να κάνει με το ότι μια ομαδική εξαρτάται από δύο φοιτητές, με αποτέλεσμα και όλοι οι περαιτέρω έλεγχοι να πρέπει να γίνουν με βάση τους δύο φοιτητές. Έλεγχοι όπως αν διεκδικεί η ομάδα άλλη πτυχιακή και σε ποια θέση προτίμησης.

Η ανάθεση των ομαδικών πτυχιακών χωρίζεται σε τρεις κατηγορίες:

1. Ανάθεση θεμάτων που τα διεκδικεί μία ομάδα στην πρώτη της επιλογή
2. Ανάθεση θεμάτων που τα διεκδικούν δύο ομάδες στις πρώτες τους επιλογές
3. Ανάθεση θεμάτων για όλες τις υπόλοιπες περιπτώσεις διεκδίκησης

- Ο αλγόριθμος τρέχει για όλες τις υπόλοιπες ομαδικές πτυχιακές που έχουν απομείνει
- Για την κάθε μία βρίσκει τον τίτλος της, τον εισηγητή της, καθώς και την ομάδα που την διεκδικεί με την μεγαλύτερη βαθμολογία
- Αν η ομάδα αυτή έχει το θέμα σαν πρώτη επιλογή, τότε παίρνει το θέμα
- Αν όχι, η πτυχιακή μπαίνει σε αναμονή¹, μέχρι να έχουμε τα κατάλληλα δεδομένα για να αποφασίσουμε

B2. Εντοπίζονται και ανατίθονται ατομικές πτυχιακές που είναι δηλωμένες από έναν φοιτητή στην πρώτη του επιλογή. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα 2.

B3. Εντοπίζονται και ανατίθονται πτυχιακές που είναι δηλωμένες από δύο φοιτητές στην πρώτη τους επιλογή. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα 3.

1. Εντοπίζονται και συγκρίνονται οι βαθμολογίες τους για το θέμα αυτό
2. Στην περίπτωση ισοβαθμίας, εντοπίζεται και συγκρίνεται ο μέσος όρος των φοιτητών για τα προαπαιτούμενα της πτυχιακής
3. Στην περίπτωση νέας ισοβαθμίας, Εντοπίζονται και συγκρίνονται οι περασμένες διδακτικές μονάδες
4. Τέλος στην (πολύ σπάνια) περίπτωση που όλοι οι παραπάνω δείκτες έχουν την ίδια τιμή, εντοπίζεται και συγκρίνεται ο γενικός μέσος όρος

B4. Ανάθεση θεμάτων για όλες τις υπόλοιπες περιπτώσεις διεκδίκησης. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα 4.

¹ Σε αναμονή μπαίνει μια πτυχιακή όταν από τις ομάδες που τη διεκδικούν αυτή με τη μεγαλύτερη βαθμολογία δεν την έχει σαν πρώτη επιλογή. Τότε, περιμένουμε να ξεκαθαρίσει η πρώτη επιλογή της ομάδας.

Είναι το βήμα του αλγορίθμου με τον μεγαλύτερο φόρτο. Αφού έχουν πραγματοποιηθεί οι αναθέσεις στις ομαδικές πτυχιακές και στις ατομικές που απαιτούσαν καμία ή λίγες αποφάσεις ανάθεσης (ομαδικές δηλώμενες από έναν ή δύο φοιτητές στις πρώτες τους προτιμήσης), πλέον ο αλγόριθμος ασχολείται με όλες τις υπόλοιπες, χωρίς να υπάρχουν άλλες ομαδοποιήσεις. Το βήμα αυτό λοιπόν αποτελείται από τις εξής διαδικασίες:

1. Τρέχουμε για όλες τις πτυχιακές που έχουν απομείνει και μέχρι τα διαθέσιμα θέματα να είναι περισσότερα από αυτά σε αναμονή
 2. Για κάθε θέμα βρίσκουμε τα απαραίτητα γι' αυτό στοιχεία (τίτλο ΠΕ – εισηγητής)
 3. Βρίσκουμε τον φοιτητή με την μεγαλύτερη βαθμολογία για το κάθε θέμα
 - Αν το θέμα είναι στην πρώτη του επιλογή, ο φοιτητής παίρνει το θέμα
 - Αν όχι, το θέμα μπαίνει σε αναμονή μέχρι να ανατεθεί η πρώτη επιλογή του φοιτητή
 4. Σε κάθε πέρασμα του αλγορίθμου (κάθε θέμα και ένα πέρασμα) υπολογίζεται το σύνολο των θεμάτων που είναι σε αναμονή και που είναι διαθέσιμα
 - Αν το σύνολο των διαθέσιμων θεμάτων είναι περισσότερα συνεχίζει κανονικά ο αλγόριθμος
 - Αν όχι, απομονώνουμε τις πτυχιακές που είναι σε αναμονή και αναθέτουμε αυτές μέχρι να ισχύει πάλι Διαθέσιμες_Πτυχιακές > Πτυχιακές_Αναμονή.
- B5. Όταν πλέον δεν υπάρχουν άλλες διαθέσιμες, ο αλγόριθμος βγαίνει από αυτό το loop και πλέον υπάρχει η πιθανότητα να υπάρχουν μόνο πτυχιακές σε αναμονή. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα 5.*
- B6. Επαναπροσδιορισμός μέσου όρου βαθμολογίας για κάθε θέμα μετά από κάθε ανάθεση*

Μετά από κάθε πτυχιακή που ανατίθεται, γίνονται καινούριοι υπολογισμοί όσον αφορά τον μέσο όρο βαθμολογιών των θεμάτων. Αυτό γίνεται έτσι ώστε να μην επηρεάζουν τους μέσους όρους φοιτητές οι οποίοι έχουν πάρει ήδη θέμα. Εφόσον γίνεται αναπροσαρμογή του μέσου όρου των θεμάτων, είναι λογικό πως αναπροσαρμόζονται κάθε φορά και οι αποκλεισμοί φοιτητών.

B7. Έλεγχος φοιτητών που δεν τους έχει ανατεθεί θέμα. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα 6.

Για να πετύχει το σύστημα έναν από τους πρωταρχικούς στόχους του (που είναι να πάρουν πτυχιακή όσο περισσότεροι φοιτητές γίνεται) εκτελείται ένα τελευταίο κομμάτι που αφορά τις αναθέσεις:

1. Εντοπίζονται φοιτητές οι οποίοι δεν πήραν πτυχιακή
2. Εντοπίζονται θέματα πτυχιακών τα οποία δεν ανατέθηκαν
3. Εντοπίζονται αν κάποιο από τα θέματα που έμειναν είναι σε δήλωση φοιτητή που δεν έχει πάρει θέμα
4. Σε περίπτωση που ένα από τα εναπομείναντα θέματα διεκδικείται από δύο φοιτητές, απλά συγκρίνονται οι βαθμολογίες τους για το θέμα

ΕΠΙΛΟΓΟΣ

Οι γραμμές που προηγήθηκαν έχουν ως στόχο την αναλυτική παρουσίαση της μεθοδολογίας που χρησιμοποιείται στην λειτουργία της εφαρμογής. Αφού γίνει μια εισαγωγή με γενικά σχόλια για τον αλγόριθμο, παρουσιάζονται τα μέρη που τον αποτελούν και καταλήγει με αναλύση του καθενός ξεχωριστά.

ΚΕΦΑΛΑΙΟ 3^ο

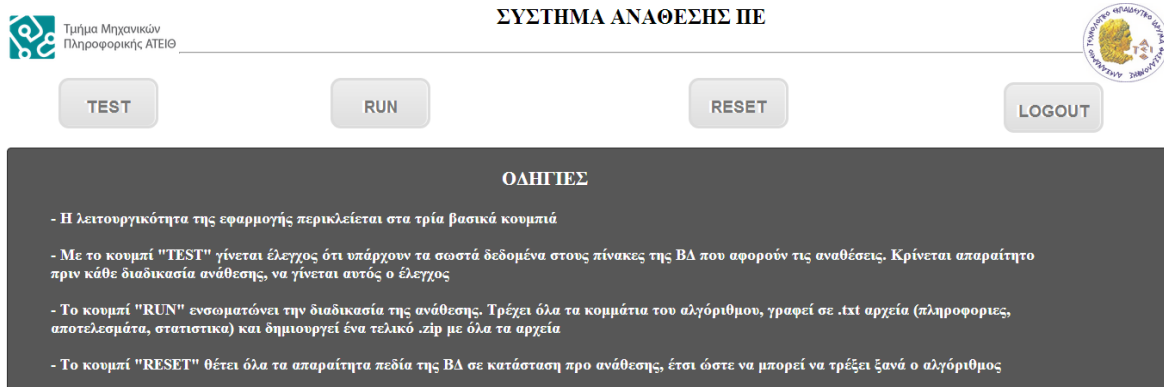
ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

ΕΙΣΑΓΩΓΗ

Το κεφάλαιο που ακολουθεί παρουσιάζει και αναλύει το περιβάλλον της εφαρμογής που αναπτύχθηκε στα πλαίσια της πτυχιακής αυτής. Παρουσιάζονται μέσω εικόνων οι δυνατότητες που έχει ο διαχειριστής του συστήματος, επεξηγείται η λειτουργικότητα στο κάθε σημείο και αναφέρονται τα αποτελέσματα που προκύπτουν.

ΤΟ ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ & Η ΒΑΣΙΚΗ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ

Η αρχική οθόνη της εφαρμογής περιλαμβάνει τη διαδικασία εισόδου του διαχειριστή του συστήματος. Μετά από την πετυχημένη είσοδο, γίνεται η μεταφορά του χρήστη στην αρχική σελίδα, όπου είναι ορατά τα τρία εικονίδια στα οποία περικλείεται η λειτουργικότητα της εφαρμογής, το εικονίδιο αποσύνδεσης, καθώς και οι οδηγίες χρήσης.



Εικόνα 1 Αρχική οθόνη του συστήματος

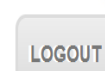
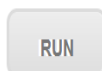
EΙΚΟΝΙΔΙΟ «TEST»

Πριν ακόμα ξεκινήσει ο αλγόριθμος να εκτελείται, μια πολύ σημαντική ενέργεια είναι να γίνει η επιβεβαίωση ότι όλα τα απαιτούμενα δεδομένα έχουν αποθηκευτεί στη βάση δεδομένων του συστήματος από τη φάση των δηλώσεων. Αυτή ακριβώς είναι η λειτουργικότητα του εικονιδίου «TEST». Θεωρείται απαραίτητο πριν από κάθε προσπάθεια εκτέλεσης του αλγόριθμου, να πραγματοποιείται αυτός ο έλεγχος.

Η εξακρίβωση της επιτυχίας της φάσης των δηλώσεων γίνεται με έλεγχο στη βάση για την ύπαρξη εγγραφών. Αν υπάρχουν εγγραφές, θεωρείται πως η βάση είναι έτοιμη να προσφέρει στην εφαρμογή όλες τις απαιτούμενες πληροφορίες, για να μπορέσει να εξάγει ακριβής αποτελέσματα. Επίσης, αν έχουν βρεθεί εγγραφές, συλλέγονται και κάποια στατιστικά στοιχεία για να παρουσιαστούν στο διαχειριστή του συστήματος, όπως ο συνολικός αριθμός φοιτητών που υπέβαλαν δήλωση, καθώς και ο αριθμός των διαθέσιμων θεμάτων προς ανάθεση. Αν δεν βρεθούν εγγραφές, θεωρείται ότι κάτι πήγε λάθος στη φάση αποθήκευσης των πληροφοριών. Και στις δύο περιπτώσεις το σύστημα εμφανίζει το κατάλληλο ενημέρωτικό μήνυμα προς τον διαχειριστή. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα 7.



ΣΥΣΤΗΜΑ ΑΝΑΘΕΣΗΣ ΠΕ



Η βάση έχει ενημερωθεί κανονικά. Πατήστε "RUN" για να τρέξει ο αλγόριθμος

- Αιτήθηκαν για θέμα 21 φοιτητές
- Τα θέματα προς διεκδίκηση είναι 61

Εικόνα 2 Μήνυμα πετυχημένης εισαγωγής δεδομένων και παρουσίαση στατιστικών στοιχείων

TEST

RUN

RESET

LOGOUT

Πρόεκυψε σφάλμα κατά τη φάση των δηλώσεων

Εικόνα 3 Μήνυμα αποτυχημένης εύρεσης δεδομένων

ΕΙΚΟΝΙΔΙΟ «RUN»

Το εικονίδιο με το οποίο αρχίζει η διαδικασία εκτέλεσης του αλγορίθμου. Τρέχει όλα τα κομμάτια του αλγόριθμου και εξάγει το τελικό συμπιεσμένο αρχείο (.zip) με όλα τα αρχεία που έχουν δημιουργηθεί σε όλες τις φάσεις εκτέλεσης. Τέλος, παράγει μήνυμα προς τον χρήστη, που ενημερώνει για την επιτυχή εκτέλεση και παρουσιάζει κάποια στατιστικά στοιχεία, όπως πόσοι φοιτητές πήραν θέμα, πόσες ομαδικές πτυχιακές εργασίες ανατέθηκαν, καθώς και πόσοι φοιτητές έμειναν χωρίς θέμα και πόσα θέματα δεν ανατέθηκαν.

Σημαντική παράμετρος σε αυτό το στάδιο, είναι το μήνυμα – πρόταση που εξάγει το σύστημα, το οποίο προτείνει τη διεξαγωγή ή μη δεύτερης φάσης ανάθεσης θεμάτων. Αν οι φοιτητές στους οποίους δεν ανατέθηκε θέμα είναι λιγότεροι από τα θέματα, το σύστημα προτείνει μια χειρωνακτική κατανομή των θεμάτων μετά από επικοινωνία των φοιτητών με την επιτροπή ΠΕ. Στην αντίθετη περίπτωση, αν οι φοιτητές είναι περισσότεροι από τα εναπομείναντα θέματα, το σύστημα προτείνει διεξαγωγή δεύτερης φάσης ανάθεσης θεμάτων, μέσω διαγωνισμού. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα 8.



Η ανάθεση πτυχιακών εργασιών ολοκληρώθηκε επιτυχώς

- Πήραν πτυχιακή εργασία 20 φοιτητές
- Δόθηκαν 2 ομαδικές
- Οι αναπομείναντες φοιτητές (1) είναι λιγότεροι από τα θέματα (43) οπότε προτείνεται μια χειρονακτική κατανομή των υπολοίπων θεμάτων

Εικόνα 4 Μήνυμα μετά την εκτέλεση του αλγορίθμου

ΕΙΚΟΝΙΔΙΟ «RESET»

Το τρίτο εικονίδιο ασχολείται με διαδικασίες μετά την εκτέλεση του συστήματος. Πιο συγκεκριμένα, η λειτουργικότητα που προσφέρει το εικονίδιο «RESET» είναι να επαναφέρει το σύστημα στην αρχική του κατάσταση, έτσι ώστε να είναι σε θέση να εκτελεστεί ξανά. Ουσιαστικά, επαναφέρει όλα τα χρησιμοποιούμενα πεδία της βάσης στις αρχικές τους τιμές. Ο αντίστοιχος κώδικας παρουσιάζεται στο παράρτημα 9.



Τα πεδία στην ΒΔ τέθηκαν στις αρχικές τους τιμές - η εφαρμογή είναι σε κατάσταση προ ανάθεσης.
Πατήστε "RUN" για να τρέξει ο αλγόριθμος

Εικόνα 5 Ενημερωτικό μήνυμα μετά την επαναφορά του συστήματος στην αρχική του κατάσταση

ΕΠΙΛΟΓΟΣ

Στο τρίτο κεφάλαιο της πτυχιακής αυτής παρουσιάστηκε το γραφικό περιβάλλον του συστήματος. Αρχικά έγιναν κάποια γενικά σχόλια για το γραφικό κομμάτι στο σύνολο του και στη συνέχεια παρουσιάστηκε η λειτουργικότητα της εφαρμογής και ειδικότερα του κάθε εικονιδίου ξεχωριστά.

ΚΕΦΑΛΑΙΟ 4^ο

ΤΕΧΝΟΛΟΓΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο γίνεται αναφορά στο τεχνολογικό περιβάλλον που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής. Θα γίνει ξεχωριστή αναφορά στη πλατφόρμα υλοποίησης που χρησιμοποιήθηκε (Microsoft WebMatrix), στη γλώσσα προγραμματισμού που χρησιμοποιήθηκε (PHP), καθώς και στη γλώσσα που χρησιμοποιήθηκε για την αποθηκεύση – ανάκτηση πληροφοριών (MySQL).

ΠΛΑΤΦΟΡΜΑ ΥΛΟΠΟΙΗΣΗΣ – MICROSOFT WEBMATRIX

Το Microsoft WebMatrix είναι μια δωρεάν πλατφόρμα της Microsoft για τη δημιουργία διαδικτυακών εφαρμογών. Δίνει τη δυνατότητα στους προγραμματιστές να δημιουργούν web εφαρμογές χρησιμοποιώντας κάποια από τα ήδη υπάρχοντα πρότυπα, είτε άλλες δημοφιλείς ανοιχτού κώδικα εφαρμογές. Το συγκεκριμένο πρόγραμμα προσφέρει πλήρη υποστήριξη για ASP.NET, PHP, javascript και HTML5 εφαρμογές. Ο στόχος της Microsoft με τη δημιουργία του, ήταν να δώσει τη δυνατότητα στους χρήστες του μέσω ενός προγράμματος να προγραμματίζουν, να παραμετροποιούν και να δημοσιεύουν τις εφαρμογές που δημιουργούν.



Εικόνα 6 Microsoft WebMatrix

ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ WebMatrix

Το WebMatrix είναι ουσιαστικά ο διάδοχος του ASP.NET WebMatrix, το οποίο κυκλοφόρησε πρώτη φορά το 2003, αλλά στη συνέχεια διακόπηκε η κυκλοφορία του. Το 2011, το WebMatrix κυκλοφόρησε με σκοπό να παρέχει πλήρη υποστήριξη σε μια σειρά από ανοιχτού τύπου εργαλεία διαχείρισης περιεχομένου (Content Management Systems – CMS) και να προσφέρει μια εφαρμογή για τη γλώσσα προγραμματισμού PHP και για την απλοποιημένη έκδοση της ASP.NET. Επικεντρώνεται κυρίως σ' ένα απλό user interface, το οποίο δίνει τη δυνατότητα στους προγραμματιστές να δημιουργήσουν μια web εφαρμογή από την αρχή είτε να παραμετροποιήσουν ένα CMS, όπως Joomla!, Drupal, WordPress, DotNetNuke και αρκετά άλλα.

Από το 2011 έως το 2012, κυκλοφόρησε το WebMatrix2 Beta το οποίο παρείχε πρόσθετη υποστήριξη για τη γλώσσα javascript, αλλά και mobile emulators, πρόσθετα template, τη δυνατότητα δημοσίευσης στο Windows Azure Web Sites και αρκετές ακόμα βελτιώσεις. Το 2012 υπήρξε η επίσημη έκδοση του WebMatrix2. Τον 2013, κυκλοφόρησε η έκδοση WebMatrix 3, το οποίο σε αντίθεση με την 2^η έκδοση, δεν τρέχει σε λειτουργικό προγενέστερο των Windows7.

ΔΥΝΑΤΟΤΗΤΕΣ

Μερικές από τις δυνατότητες που προσφέρει η τρίτη έκδοση του προγράμματος είναι:

- Απλοποιημένη δημιουργία, δημοσίευση και συγχρονισμός με κάποια ήδη υπάρχουσα ιστοσελίδα, μέσω της υπηρεσίας cloud του προγράμματος
- Smart coding και υπογράμμιση λαθών για HTML5, CSS3 και JavaScript
- Υποστήριξη των server-side γλωσσών ASP.NET, PHP
- Υποστήριξη των jQuery, jQuery Mobile, LESS, Sass
- Mobile simulators
- Database manager για MySQL, MS SQL Server, SQL CE
- Διαμοιρασμός αρχείων μέσω FTP & Web Deploy

- Δυνατότητα download για απομακρυσμένα site και τοπική επεξεργασία
- Ενσωματωμένες δυνατότητες SEO και αναφορές απόδοσης

Από την άλλη, σαν μειονέκτημα θα μπορούσε να χαρακτηριστεί η υποχρεωτική εγκατάσταση του Microsoft IIS Server, ο οποίος δεν είναι όπως οι περισσότεροι web server (π.χ. apache) unix based.

ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ – PHP

Η PHP είναι μια server – side γλώσσα προγραμματισμού σχεδιασμένη για διαδικτυακές, αλλά χρησιμοποιείται και για γενικής χρήσης εφαρμογές. Σε πρώτη φάση τα αρχικά PHP σήμαιναν «Personal Home Page», πλέον όμως σημαίνουν «PHP : Hypertext Preprocessor». Μέχρι το 2013, η PHP είχε εγκατασταθεί στο 39% επί του συνόλου των ιστοσελίδων και σε παραπάνω από 2 εκατομμύρια web servers.

Ο κώδικας της PHP εκτελείται διαμέσου ενός web server, από τον οποίο και παράγεται το αποτέλεσμα σε μορφή web page. Οι εντολές γραμμένες σε PHP μπορούν να εκτελεστούν είτε αυτόνομες, είτε να ενσωματωθούν σε HTML αρχεία. Περιέχει επίσης command line interface και μπορεί να χρησιμοποιηθεί αυτόνομα σε γραφικές εφαρμογές. Σημαντική επισήμανση είναι πως είναι ελεύθερο λογισμικό και μπορεί να σταθεί σε κάθε λειτουργικό σύστημα και πλατφόρμα δωρεάν.



Εικόνα 7 Γλώσσα προγραμματισμού PHP

ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ PHP

Η ιστορία της php ξεκινά το 1994 όταν ένας φοιτητής με το όνομα Rasmus Lerdorf δημιούργησε ένα απλό script με όνομα php.cgi χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το βιογραφικό του φοιτητή. Η γλώσσα τότε ονομαζόταν PHP/FI, από τα αρχικά Personal Home Page/Form Interpreter.

Το 1997 η γλώσσα έφτασε στην έκδοση 2.0 βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμούσε περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν. Την ίδια χρονιά οι προγραμματιστές Andi Gutmans και Zeev Suraski έγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρχικά στην έκδοση PHP 2.0, φτάνοντας τη γλώσσα στην έκδοση 3.0, η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Το 1998 ακολούθησε η έκδοση 4, το 2004 η έκδοση 5, εκδόσεις τις οποίες χρησιμοποιεί ο μεγαλύτερος αριθμός ιστότοπων, που χρησιμοποιούν την PHP.

ΧΡΗΣΗ ΤΗΣ PHP

Η php είναι μια γενικής χρήσης γλώσσα προγραμματισμού, που χρησιμοποιείται κυρίως για την ανάπτυξη server-side διαδικτυακών εφαρμογών, στις οποίες η php εκτελείται σ' έναν web server. Είναι μία γλώσσα που μπορεί να χρησιμοποιηθεί στους περισσότερους web servers, καθώς και στις περισσότερες πλατφόρμες και λειτουργικά συστήματα. Επίσης, μπορεί να συσχετιστεί με τα περισσότερα συστήματα διαχείρισης βάσεων δεδομένων.

Αρχικά σχεδιάστηκε με σκοπό τη δημιουργία ιστοσελίδων, αλλά στην πορεία εστίασε κυρίως στο «server-side scripting». Σε αυτό σημείο, φέρει αρκετές ομοιότητες με παρόμοιες γλώσσες, οι οποίες προσφέρουν κι αυτές δυναμικό περιεχόμενο από τον web server προς τον client, όπως Microsoft ASP.NET, Java Server Pages και Perl.

Τον 2007 πάνω από 20 εκατομύρια domain στο διαδίκτυο φιλοξενούνταν σε servers με εγκατεστημένη την php. Το 2014 η php χρησιμοποιείται στο 82% όλων των web εφαρμογών και είναι το δημοφιλέστερο λογισμικό ανοιχτού κώδικα. Πολλά συστήματα διαχείρισης περιεχόμενου βασίζονται στην php, όπως Joomla, WordPress, Drupal, Moodle και αρκετά άλλα.

ΠΛΕΟΝΕΚΤΗΜΑΤΑ - ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ PHP

Η php προσφέρει πολλά πλεονεκτήματα στους χρήστες της και τα κυριότερα, σε σύγκριση με γλώσσες που θεωρούνται βασικοί της ανταγωνιστές, όπως Perl, ASP.NET, JSP κτλ, είναι:

- Υψηλή απόδοση
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες web διαδικασίες
- Ευκολία μάθησης και χρήσης
- Δυνατότητα μεταφερισιμότητας
- Ύπαρξη υποστήριξης – τεκμηρίωσης

Πιο αναλυτικά, διάφορα χαρακτηριστικά της php ανά κατηγορία είναι:

- Php και βάσεις δεδομένων: προσφέρει συνδέσεις με πολλά συστήματα ΒΔ., όπως PostgreSQL, Oracle, dbm, DB2 κτλ.
- Ενσωματωμένες βιβλιοθήκες: σχεδιάστηκε για να χρησιμοποιείται στο web, οπότε έχει πολλές ενσωματωμένες βιβλιοθήκες, όπως σύνδεση με άλλες web υπηρεσίες, δημιουργία pdf εγγράφων, αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου και αρκετά άλλα.
- Μεταφερισιμότητα: η php είναι σχεδιασμένη έτσι ώστε να λειτουργεί σε πολλά συστήματα. Υπάρχει η δυνατότητα να γράψει κάποιος κώδικα σε php σε διάφορα λειτουργικά, όπως εκδόσεις των Microsoft Windows, εκδόσεις του Unix κτλ.

ΑΝΑΚΤΗΣΗ & ΕΠΕΞΕΡΓΑΣΙΑ ΠΛΗΡΟΦΟΡΙΩΝ – MySQL

Η mysql είναι το δεύτερο πιο χρησιμοποιούμενο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοιχτού κώδικα. Είναι αρκετά δημοφιλές κυρίως στις διαδικτυακές εφαρμογές και είναι ένα από τα κύρια συστατικά του LAMP (Linux / Apache / MySQL / Perl-PHP-Python). LAMP είναι ο συνδυασμός λογισμικών ανοιχτού κώδικα για τη δημιουργία web εφαρμογών, που χρησιμοποιούν ως σύστημα διαχείρισης της βάσης δεδομένων την MySQL. Για εμπορικούς σκοπούς, αρκετές επί πληρωμή εκδόσεις είναι διαθέσιμες, οι οποίες προσφέρουν επιπλέον λειτουργικότητα. Γνωστές εφαρμογές που τη χρησιμοποιούν είναι:

- Joomla
- WordPress
- Drupal
- MyBB
- MODx



Εικόνα 8 Γλώσσα ερωτοαποκρίσεων MySQL

ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Μέσω του MySQL Workbench δίνεται η δυνατότητα στους χρήστες να διαχειριστούν με γραφικό τρόπο MySQL βάσεις δεδομένων και την εικονική σχεδίαση της δομής μιας ΒΔ. Το MySQLWorkbench είναι διαθέσιμο σε δύο εκδόσεις, μία που διανέμεται δωρεάν (Community Edition) και την βασική έκδοση (Standard Edition), η οποία προσφέρει επιπλέον δυνατότητες και λειτουργίες.

Υπάρχουν αρκετά «τρίτα» προγράμματα, τα οποία δίνουν τη δυνατότητα να διαχειριστούμε MySQL βάσεις δεδομένων. Μερικά από τα πιο γνωστά είναι:

- Adminer
- DBEdit
- OpenOffice.org Base
- phpmyAdmin
- Webmin

ΔΥΝΑΤΟΤΗΤΕΣ & ΠΕΡΙΟΡΙΣΜΟΙ

Όσον αφορά τις δυνατότητες που προσφέρει η MySQL και οι δύο εκδόσεις (Community Server – Enterprise Server) βασίζονται στον ίδιο κώδικα υλοποίησης και έχουν αρκετά κοινά χαρακτηριστικά – δυνατότητες, όπως:

- Δυνατότητα χρήσης σε πολλές πλατφόρμες
- Χρήση αποθηκευμένων διαδικασιών (stored procedures)
- Χρήση triggers
- Πληροφορίες για το σχεσιακό σχήμα
- Υποστήριξη SSL
- Ανεξάρτητες μηχανές αποθήκευσης (MyISAM, InnoDB, MySQL Archive)
- Εμφωλευμένα SELECT
- Υποστήριξη UNICODE χαρακτήρων

Όσον αφορά τους περιορισμούς του προγράμματος, οι πιο σημαντικοί είναι:

- Όπως και άλλες βάσεις δεδομένων SQL, έτσι και η MySQL δεν συμμορφώνεται πλήρως με το πλήρες πρότυπο SQL, όπως οι αναφορές στα ξένα κλειδιά όταν χρησιμοποιούνται μηχανές αποθήκευσης διαφορετικές από την InnoDB.
- Μέχρι και την έκδοση 5.7, υπάρχει ο περιορισμός στα triggers, τα οποία περιορίζονται σ' ένα ανά εκτέλεση, γεγονός που σημαίνει πως το πολύ ένα trigger μπορεί να οριστεί μετά την εκτέλεση ενός INSERT και ένα πριν. Δεν επιτρέπονται triggers στις όψεις (views).
- Η MySQL, όπως και τα περισσότερα RDBMS, εξαρτώνται από την απόδοση του σκληρού δίσκου.

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο που προηγήθηκε γίνεται προσπάθεια να παρουσιαστούν τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν στην εφαρμογή. Για το κάθε ένα έγινε ξεχωριστή αναφορά στα σημαντικότερα ιστορικά του στοιχεία, στις δυνατότητες και τους περιορισμούς του και στα πεδία εφαρμογής του καθενός.

ΚΕΦΑΛΑΙΟ 5^ο

ΕΠΙΛΟΓΟΣ

ΕΙΣΑΓΩΓΗ

Το πέμπτο και τελευταίο κεφάλαιο της παρούσας πτυχιακής περιλαμβάνει τον επίλογο όσων παρουσιάστηκαν στα προηγούμενα κεφάλαια. Παρουσιάζει κάποια συμπεράσματα στα οποία κατέληξε ο γραφών από την διαδικασία συγγραφής της πτυχιακής εργασίας αυτής και κλείνει με τη σημαντικότητα από την αυτοματοποίηση της διαδικασίας ανάθεσης πτυχιακών.

ΣΗΜΑΝΤΙΚΟΤΗΤΑ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗΣ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΑΝΑΘΕΣΕΩΝ

Η ανάλυση που προηγήθηκε έκανε κατανοητό σε μεγάλο βαθμό πως προέκυπτε η ανάγκη αυτοματοποίησης της διαδικασίας ανάθεσης πτυχιακών εργασιών. Ακολουθεί μια λίστα με βασικότερα πλεονεκτήματα που προσφέρει η αυτοματοποίηση αυτή:

- Μείωση φόρτου στο ελάχιστο. Μέσω της εφαρμογής που δημιουργήθηκε ο φόρτος για την επιτροπή ΠΕ μειώνεται στο ελάχιστο δυνατό. Δεν υπάρχει η ανάγκη δημιουργίας αρχείων πλέον, καθώς δημιουργούνται αυτόματα, ούτε η ανάγκη υπολογισμών για τους απαραίτητους δείκτες, τα οποία επίσης πραγματοποιεί η εφαρμογή.
- Άμεση εξαγωγή αποτελεσμάτων. Πλέον, μετά το πέρας της διορίας δήλωσης προτιμήσεων από τους φοιτητές ο χρόνος που απαιτείται για την ανακοίνωση των αποτελεσμάτων είναι απλά μερικά λεπτά.
- Αποδοτικότητα. Η εφαρμογή δημιουργήθηκε με τέτοιο τρόπο ώστε να αναθέτει όσες περισσότερες πτυχιακές γίνεται. Αυτό επιτυγχάνεται με την κατάλληλη επεξεργασία των δεικτών και πρόληψη για ύπαρξη εναλλακτικών περιπτώσεων και μεθόδων ανάθεσης θεμάτων, σε συνάρτηση με τις σύνθηκες κάθε φορά.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η προσπάθεια για την δημιουργία του εργαλείου αυτού δεν ήταν εύκολη, καθώς οι αναθέσεις πτυχιακών εργασιών είναι μια σύνθετη διαδικασία, με πολλές διαφορετικές περιπτώσεις ανάθεσης, οι οποίες έπρεπε να απεικονιστούν όσο το δυνατόν με περισσότερη ακρίβεια με προγραμματιστικό τρόπο. Ένα από τα μεγαλύτερα θέματα που παρουσιάστηκαν είχε να κάνει με το πως θα αποτυπώνονταν οι ομαδικές πτυχιακές. Αν και έγιναν προσπάθειες ώστε να αντιμετωπιστούν με παρόμοιο τρόπο με τα ατομικά θέματα, δεν κατέστη δυνατό, δημιουργήθηκαν όμως οι κατάλληλες προϋποθέσεις ώστε να ενσωματωθεί η παράμετρος αυτή με το υπόλοιπο σύστημα, χωρίς να επηρεάζει την ομαλή λειτουργία του.

Καθώς η ψηφιακή εποχή βρίσκεται στην κορυφή της ανάπτυξής της, πρέπει να υπάρχει η δυνατότητα εκμετάλλευσης του κάθε πλεονεκτήματος που προσφέρεται. Στην περίπτωση της πτυχιακής αυτής, έγινε ο καλύτερος δυνατός συνδυασμός όλων των παραμέτρων ώστε να δημιουργηθεί ένα αποδοτικό εργαλείο, αλλά και απλό στη χρήση για την επιτροπή ΠΕ του τμήματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- <http://en.wikipedia.org/wiki/PHP>
- <http://en.wikipedia.org/wiki/MySQL>
- http://en.wikipedia.org/wiki/Microsoft_WebMatrix

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ 1

```

<?php
//ομαδικές ΠΕ

    $am1=""; $am2="";
    $teamThesis=0;
    $findBetter1=0; $findBetter2=0;
    $dis=0;
    //ψάχνουμε για ομαδική ΠΕ που να είναι δηλωμένη από μία μόνο ομάδα
    στην 1η προτίμηση της
    $sql="SELECT c.dissertation as dissertation, c.id AS aa, d.titlegr AS title,
    d.sup_lastname AS prof, c.student AS am1, c.student2 AS am2, d.id AS id FROM
    dissertations d, candidacy c WHERE d.id=c.dissertation AND d.sumChoices=1
    AND c.preference=1 AND c.status=0 AND d.typeassign='Ομαδική'";
    $result=mysql_query($sql);
    if($result==false){
        die(mysql_error());
    }
    while($row=mysql_fetch_array($result)){
        $am1=$row['am1']; $am2=$row['am2'];
        $firstam=$am1.".";
        $am=$firstam.$am2;
        $id=$row['dissertation'];

    $dis=$id;

        $givenID[]=$id;
        $teamThesis++;
        $aa++;

    $title=$row['title'];
    $prof=$row['prof'];
        fwrite($open, $aa); fwrite($open, "\t");
        fwrite($open, $row['title']); fwrite($open, "\t");
        fwrite($open, $row['prof']); fwrite($open, "\t");
        fwrite($open, $am); fwrite($open, ".PHP_EOL");
    require ('../algorithm/updateFields/insertResults.php');//εισάγουμε τα
    αποτελέσματα σε πίνακα στη βάση
        $givenAMArray[]=$am1;
        $givenAMArray[]=$am2;
        require ('../algorithm/updateFields/setDate.php');//θέτουμε
    ημερομηνία ανάθεσης
    require
    ('../algorithm/updateFields/updateGivenTeamThesisFields.php');//ανανεώνουμε τα
    πεδία των φοιτητών που πήραν ΠΕ
    require
    ('../algorithm/updateFields/updateSumChoices.php');//ανανεώνουμε το πεδίο
    sumChoices
    
```

```
require
('./algorithm/updateFields/updateCandidacyStatusTeam.php');//ανανεώνουμε τις
προτιμήσεις για όσες ΠΕ επηρεάζουν οι φοιτητές που πήραν θέμα
require ('../algorithm/findValues/findTeamPreference.php');//βρίσκουμε
προτίμηση ΠΕ για τα στατιστικά
}
```

?>

<?php

```
$dis=0;
//ψάχνουμε για ομαδικές ΠΕ που είναι δηλωμένες από δύο ομάδες στην πρώτη
$sumPref=0;//θα αθροίσουμε τον αριθμό επιλογής, ο οποίος πρέπει να βγεί δύο
(1+1)
$sql="SELECT id FROM dissertations WHERE sumChoices=2 AND status=0
AND typeassign='Ομαδική';
$result=mysql_query($sql);
if(!$result){
    echo(mysql_error());
    exit();
}
while($row=mysql_fetch_array($result)){
    $id=$row['id'];
    $sql2="SELECT dissertation, id, preference, dissertation FROM candidacy
WHERE dissertation='".$id."'";//παιρνουμε τα στοιχεία του θέματος
    $result2=mysql_query($sql2);
    if(!$result2){
        echo(mysql_error());
        exit();
    }
    while($row2=mysql_fetch_array($result2)){
        $sumPref=$row2['preference'];
    }

    if($sumPref==2){//σημαίνει ότι και οι δυο ομάδες έχουν την ΠΕ στην 1η επιλογή
        //από τις ομάδες που την διεκδικούν βρίσκουμε αυτή με την καλύτερη
        βαθμολογία
        $dis=$id;
        require ('../algorithm/findValues/findMaxTeam.php');//βρίσκουμε την ομάδα με
        την μεγαλύτερη βαθμολογία για το θέμα που είμαστε
        require ('../algorithm/findValues/findTitleProf.php');//βρίσκουμε τα στοιχεία του
        θέματος
        require ('../algorithm/updateFields/insertResults.php');//εισάγουμε
        αποτελέσματα
```

```

        require ('../algorithm/txtFiles/writeTxt.php');//γράφουμε στο αρχείο
αναθέσεων
        require
('../algorithm/updateFields/updateGivenTeamThesisFields.php');//ανανεώνουμε τα
πεδία των φοιτητών που πήραν ΠΕ
        require ('../algorithm/updateFields/updateSumChoices.php');//ανανεώνουμε το πεδίο sumChoices
        require
('../algorithm/updateFields/updateCandidacyStatusTeam.php');//ανανεώνουμε τις
προτίμησεις για όσες ΠΕ επηρεάζουν οι φοιτητές που πήραν θέμα
        require
('../algorithm/findValues/findTeamPreference.php');//βρίσκουμε προτίμηση ΠΕ για
τα στατιστικά
        $givenID[]=$id;
        $firstam=$am1.".";
        $am=$firstam.".$am2;
        $teamThesis++;
        $givenAMArray[]=$am1; $givenAMArray[]=$am2;
    }
}
?>

```

```

<?php
//τρέχουμε τον βασικό αλγόριθμο για τις ομαδικές

$dis=0;
$sql="SELECT id FROM dissertations WHERE typeassign='Ομαδική'
AND status=0";
$result=mysql_query($sql);
if(!$result){
    die('Invalid Query: ' .mysql_error());
    exit();
}
while($row=mysql_fetch_array($result)){
    $id=$row['id'];
//παίρνουμε student, student2, preference, sumChoices
    require ('../algorithm/findValues/findMaxTeamThesis.php');
    $idaa=$row['id'];
    require('../algorithm/findValues/findTeamBetterPref.php');
    if($preference==1 || $sumChoices==1){
        if($findBetter==1){
            require
('../algorithm/findValues/findMaxTeamThesis.php');
            $id=$prefID;//το ID της ΠΕ που βρήκαμε σε
καλύτερη προτίμηση

            $idaa2=$row['id'];
            if($idaa==$idaa2){

```



```

$dis=$id;
                                $givenID[]=$id;
                                $firstam=$am1.".";
                                $am=$firstam.".$am2;
                                $teamThesis++;
                                require
('..algorithm/findValues/findTitleProf.php');//βρίσκουμε στοιχεία της ΠΕ
                                require ('..algorithm/updateFields/insertResults.php');//εισάγουμε
αποτελέσματα
                                require
('..algorithm/txtFiles/writeTxt.php');//γράφουμε στο αρχείο αναθέσεων
                                require
('..algorithm/updateFields/updateGivenTeamThesisFields.php');//ανανεώνουμε
τα πεδία των φοιτητών που πήραν ΠΕ
                                require
('..algorithm/updateFields/updateSumChoices.php');//ανανεώνουμε το πεδίο
sumChoices
                                require
('..algorithm/findValues/findTeamPreference.php');//βρίσκουμε προτίμηση ΠΕ για
τα στατιστικά
                                require
('..algorithm/updateFields/updateCandidacyStatusTeam.php');//ανανεώνουμε τις
προτίμσεις για όσες ΠΕ επηρεάζουν οι φοιτητές
                                $givenAMArray[]=$am1;
$givenAMArray[]=$am2;
                                }
                                }
                                else{
                                $givenID[]=$id;
                                $firstam=$am1.".";
                                $am=$firstam.".$am2;
                                $teamThesis++;
                                require
('..algorithm/findValues/findTitleProf.php');
                                require ('..algorithm/updateFields/insertResults.php');
                                require('..algorithm/txtFiles/writeTxt.php');

                                require('..algorithm/updateFields/updateGivenTeamThesisFields.php');
                                require
('..algorithm/updateFields/updateSumChoices.php');
                                require
('..algorithm/findValues/findTeamPreference.php');
                                require ('..algorithm/updateFields/updateCandidacyStatus.php');
                                $givenAMArray[]=$am1;
$givenAMArray[]=$am2;
                                }
                                }
                                else{

```

```
        require
(' ../algorithm/findValues/findTeamBetterPref.php');
        if($findBetter1==1 || $findBetter2==1){//αν ένας απο τους
δουο εχει προτιμηση σε καλυτερη θεση, ΠΕ σε αναμονή
        require (' ../algorithm/standby/setStandby.php');
        }
    }
}
```

?>

```

<?php
//ΠΕ που τις διεκδικεί ένας στην 1η του επιλογή
    $preferenceValue=0;
    $dis=0;
    //ψάχνουμε για ΠΕ με sumChoices=1 που είναι στην 1η προτίμηση
    $sql="SELECT d.id AS id, d.titlegr AS title, d.sup_lastname AS prof,
c.student AS am FROM dissertations d, candidacy c WHERE d.id=c.dissertation
AND d.sumChoices=1 AND c.preference=1 AND c.status=0 AND
d.typeassign='Ατομική'";
    $result=mysql_query($sql);
    if(!$result){
        echo(mysql_error());
        exit();
    }
    while($row=mysql_fetch_array($result)){
        //αν υπάρχει, βρίσκουμε τίτλο, εισηγητή και την δίνουμε
        $id=$row['id'];
        $dis=$id;
        $givenID[]=$id;//αποθηκεύουμε το ID του θέματος στον
αντίστοιχο πίνακα
        $am=$row['am'];
        $title=$row['title'];
        $prof=$row['prof'];
        require ('../algorithm/txtFiles/writeTxt.php');//γράφουμε τα στοιχεία
της ανάθεσης στο αρχείο αποτελεσμάτων
        require ('../algorithm/updateFields/insertResults.php');//εισάγουμε τα
αποτελέσματα στον αντίστοιχο πίνακα στη βάση
        require
('../algorithm/updateFields/updateGivenThesisFields.php');
        require ('../algorithm/updateFields/setDate.php');//θέτουμε
ημερομηνία ανάθεσης
        require ('../algorithm/findValues/findPreference.php');//βρίσκουμε
σε τι προτίμηση ήταν το θέμα για να υπολογίσουμε τα αντίστοιχα στατιστικά
        $givenAMArray[]=$am;
    }

    require
('../algorithm/updateFields/updateSumChoices.php');//ανανεώνουμε το πόσοι
διεκδικούν το κάθε θέμα
        $lenght=sizeof($givenAMArray);
        foreach($givenAMArray as $am){
            require
('../algorithm/updateFields/updateCandidacyStatus.php');//μετά από κάθε ανάθεση
υπολογίζουμε αποκλείουμε φοιτητές από θέματα με βάση τα νέα δεδομένα
        }
?>

```

ΠΑΡΑΡΤΗΜΑ 3

```

<?php
//ΠΕ που τις διεκδικούν δύο στη πρώτη τους επιλογή
    $preferenceValue=0;
        //ψάχνουμε για ΠΕ με sumChoices=2 και τα preference=1
        $countPref=0;
    $dis=0;
    $idArray=array();
    $scoreArray=array();
    $amArray=array();
    $cavgArray=array();
    $dmo1=0; $dmo2=0; $gmo1=0; $gmo2=0;
    $give1=0; $give2=0; //για να ξέρουμε σε ποιον από τους δύο τη δώσαμε
    $sql="SELECT sumChoices, id FROM dissertations WHERE status=0
AND sumChoices=2";
    $result=mysql_query($sql);
    if(!$result){
        echo(mysql_error());
        exit();
    }
    while($row=mysql_fetch_array($result)){
        array_push($idArray, $row['id']);
    }
    for($i=0;$i<sizeof($idArray);$i++){//τρέχουμε για όλες περιπτώσεις
        πτυχιακών βρέθηκαν
            $countPref=0;
            $id=array_shift($idArray);
            //ψάχνουμε για το αν αυτήν την ΠΕ την έχουν στην πρώτη
            προτίμηση
                $sql="SELECT preference, dissertationScore, student,
coursesAVG FROM candidacy WHERE id='".$id."'";
                $result=mysql_query($sql);
                if(!$result){
                    echo(mysql_error());
                    exit();
                }
                while($row=mysql_fetch_array($result)){
                    $amArray[]=$row['student'];
                    $scoreArray[]=$row['dissertationScore'];
                    $cavgArray[]=$row['coursesAVG'];
                    if($row['preference']==1){
                        $countPref++;
                    }
                }
            }
    }

```

```

        if($countPref==2){//αν είναι στις πρώτες προτιμήσεις, βρίσκουμε
        ότι χρειάζεται για τον έλεγχο & την ανάθεση
            $am1=array_shift($amArray);
        $am2=array_shift($amArray);
            $score1=array_shift($scoreArray);
        $score2=array_shift($scoreArray);
            $cavg1=array_shift($cavgArray);
        $cavg2=array_shift($cavgArray);
            $sql="SELECT titlegr, sup_lastname FROM
        dissertations WHERE id=".$id."";//βρίσκουμε τίτλο & εισηγητή
            $result=mysql_query($sql);
            while($row=mysql_fetch_array($result)){
                $title=$row['titlegr'];
                $prof=$row['sup_lastname'];
            }
            $sql="SELECT DM, gmo FROM students WHERE
        am=".$am1."";//βρίσκουμε ΔΜ, ΓΜΟ
            $result=mysql_query($sql);
            while($row=mysql_fetch_array($result)){
                $dm1=$row['DM'];
                $gmo1=$row['gmo'];
            }
            $sql="SELECT DM, gmo FROM students WHERE
        am=".$am2."";//βρίσκουμε ΔΜ, ΓΜΟ
            $result=mysql_query($sql);
            while($row=mysql_fetch_array($result)){
                $dm2=$row['DM'];
                $gmo2=$row['gmo'];
            }
            //όποιος έχει μεγαλύτερη βαθμολογία στην ΠΕ παίρνει το θέμα
            if($score1>$score2){
                require ('../algorithm/txtFiles/writeTxt1.php');
                $givenAMArray[]=$am1;
                $dis=$id;
                $givenID[]=$id;
                $first++;
                $give1=1;
            }
            elseif($score2>$score1){
                require ('../algorithm/txtFiles/writeTxt2.php');
                $givenAMArray[]=$am2;
                $dis=$id;
                $givenID[]=$id;
                $first++;
                $give2=1;
            }
            else{//αν ίσα, κοιτάμε ΜΟ προαπαιτούμενων
                if($cavg1>$cavg2){
                    require
        ('../algorithm/txtFiles/writeTxt1.php');

```

```

        $givenAMArray[]=$am1;
        $totalIDArray[]=$id;
        $first++;
        $give1=1;
    }
    elseif($scavg2>$scavg1){
        require
(' ../algorithm/txtFiles/writeTxt2.php');
        $dis=$id;
        $givenAMArray[]=$am2;
        $givenID[]=$id;
        $first++;
        $give2=1;
    }
    else{//αν ίσα, κοιτάμε ΔM
        if($dm1>$dm2){
            require
(' ../algorithm/txtFiles/writeTxt1.php');
            $dis=$id;
            $givenAMArray[]=$am1;
            $givenID[]=$id;
            $first++;
            $give1=1;
        }
        elseif($dm2>$dm1){
            require
(' ../algorithm/txtFiles/writeTxt2.php');
            $dis=$id;
            $givenAMArray[]=$am2;
            $givenID[]=$id;
            $first++;
            $give2=1;
        }
        else{//αν ίσα, κοιτάμε ΓMO
            if($gmo1>$gmo2){
                require
(' ../algorithm/txtFiles/writeTxt1.php');
                $givenAMArray[]=$am1;
                $dis=$id;
                $givenID[]=$id;
                $first++;
                $give1=1;
            }
            elseif($gmo2>$gmo1){
                require
(' ../algorithm/txtFiles/writeTxt2.php');
                $givenAMArray[]=$am2;
                $dis=$id;
            }
        }
    }
}

```

```
        $givenID[]=$sid;
        $first++;
        $give2=1;
    }
}
}
}
    }
    if($give1==1){//αν την πήρε ο πρώτος
        $am=$am1;
        require ('../algorithm/updateFields/insertResults.php');
        require
        ('../algorithm/updateFields/updateCandidacyStatus.php');
        require
        ('../algorithm/updateFields/updateGivenThesisFields1.php');
        require
        ('../algorithm/updateFields/updateSumChoices.php');
        require ('../algorithm/findValues/findPreference.php');
    }
    elseif($give2==1){
        $am=$am2;
        require ('../algorithm/updateFields/insertResults.php');
        require
        ('../algorithm/updateFields/updateCandidacyStatus.php');
        require
        ('../algorithm/updateFields/updateGivenThesisFields2.php');
        require
        ('../algorithm/updateFields/updateSumChoices.php');
        require ('../algorithm/findValues/findPreference.php');
    }
    require ('../algorithm/updateFields/setDate.php');
}
else{
    unset($scoreArray);
    unset($amArray);
    unset($cavgArray);
}
}
```

?>

ΠΑΡΑΡΤΗΜΑ 4

```
<?php
```

```
//ο βασικός αλγόριθμος. τρέχει όσο υπάρχουν διαθέσιμες
```

```

    $find=0;//κρατάει αν βρέθηκε ΠΕ σε καλύτερη προτίμηση
    $countAvailable=0;
    $sumAvailable=0;
    $sumStandby=0;
    $c=0;
    $preferenceValue=0;
    $dis=0;
    //βρίσκουμε όλες τις διαθέσιμες ΠΕ
    $idArray=array();
    $sql="SELECT id FROM dissertations WHERE status=0 AND
typeassign='Ατομική'";
    $result=mysql_query($sql);
    if(!$result){
        echo(mysql_error());
        exit();
    }
    while($row=mysql_fetch_array($result)){
        $idArray[]=$row['id'];
        $countAvailable++;
    }
    require ('../algorithm/findValues/findStandbyAvailable.php');
    while($countAvailable>0 && $sumAvailable>$sumStandby){//τρέχουμε για
όλες τις διαθέσιμες ΠΕ
        $sumAvailable=0;
        $sumStandby=0;
        $countAvailable--;
        $id=array_shift($idArray);
        //βρίσκουμε τίτλο και εισηγητή
        /*παίρνουμε και το status γιατί καθώς δίνουμε ΠΕ
και ανανεώνεται το sumChoices μπορεί κάποια ΠΕ που πήραμε ως
διαθέσιμη να κλείσει λόγω μηδενικής προτίμησης*/
        $sql="SELECT titlegr, sup_lastname, status FROM dissertations
WHERE id='".$id."'";
        $result=mysql_query($sql);
        if(!$result){
            echo(mysql_error());
            exit();
        }
        while($row=mysql_fetch_array($result)){
            $title=$row['titlegr'];
            $prof=$row['sup_lastname'];
            $status=$row['status'];

```



```

    }
    if($status==0){//συνεχίζουμε μόνο αν η ΠΕ είναι διαθέσιμη
        //βρίσκουμε αυτόν με τη μέγιστη βαθμολογία
        require ('../algorithm/findValues/findMaxStudent.php');
        if($preference==1){//αν είναι η 1η προτίμηση τη δίνουμε
            require ('../algorithm/txtFiles/writeTxt.php');

$first++;

                $givenAMArray[]=$sam;

$dis=$id;

                $givenID[]=$id;
        require ('../algorithm/updateFields/insertResults.php');
            require
('..../algorithm/updateFields/updateGivenThesisFields.php');
            require ('../algorithm/standby/unStandby.php');
            require
('..../algorithm/updateFields/updateSumChoices.php');
            require
('..../algorithm/updateFields/updateCandidacyStatus.php');
                //require
('..../algorithm/findValues/findPreference.php');
            require ('../algorithm/updateFields/setDate.php');

        }
        else{//αν δεν την έχει 1η προτίμηση ψάχνουμε να
        βρούμε αν διεκδικεί κάποια από τις παραπάνω προτιμήσεις του
            $sql="SELECT preference FROM candidacy
WHERE student='".$sam.'" AND status=0";
            $result=mysql_query($sql);
            if(!$result){
                echo(mysql_error());
                exit();
            }
            while($row=mysql_fetch_array($result)){
                if($preference>$row['preference']){
                    $find=1;
                }
            }
            if($find==1){//έχει ΠΕ σε καλύτερη προτίμηση--
>ΠΕ που είμαστε σε αναμονή
                require
('..../algorithm/standby/setStandby.php');
            }
            else{//αν δεν βρέθηκε σε καλύτερη προτίμηση
ΠΕ παίρνει αυτήν που είμαστε
                require
('..../algorithm/txtFiles/writeTxt.php');

                    $givenAMArray[]=$sam;
                    $sql34="SELECT preference FROM candidacy WHERE
dissertation='".$id.'" AND student='".$sam.'";
                    $result34=mysql_query($sql34);

```

```

if(!$result34){
    echo(mysql_error());
    exit();
}
while($row34=mysql_fetch_array($result34)){
    $idPreference=$row34['preference'];
    $preferenceValue=1;
}
/*if($idPreference==1){
    $first++;
} */
if($idPreference==2){
    $second++;
}
elseif($idPreference==3){
    $third++;
}
elseif($idPreference==4){
    $fourth++;
}
elseif($idPreference==5){
    $fifth++;
}
}

$dis=$id;
    $givenID[]=$id;
require ('../algorithm/updateFields/insertResults.php');
require
('../algorithm/updateFields/updateGivenThesisFields.php');
require
('../algorithm/standby/unStandby.php');
require
('../algorithm/updateFields/updateSumChoices.php');
require
('../algorithm/updateFields/updateCandidacyStatus.php');
//require
('../algorithm/findValues/findPreference.php');
require
('../algorithm/updateFields/setDate.php');
}
}
} //τρέχουμε μόνο αν η ΠΕ παραμένει διαθέσιμη
if($countAvailable==0){
    $sql="SELECT id FROM dissertations WHERE status=0
AND standby=0 AND typeassign='Ατομική';
    $result=mysql_query($sql);
    if(!$result){
        echo(mysql_error());
        exit();
    }
    while($row=mysql_fetch_array($result)){

```

```
        $idArray[]=$row['id'];
        $countAvailable++;
    }
}
require ('../algorithm/findValues/findStandbyAvailable.php');
/*πριν τελειώσει το κάθε πέρασμα ελέγχουμε τον αριθμό διαθέσιμων - σε
αναμονή
αν είναι οι σε αναμονή περισσότερες, τις απομωνώνουμε και αναθέτουμε από
αυτές
μέχρι ο αριθμός τους να γίνει μικρότερος από τις διαθέσιμες*/
if($sumAvailable<=$sumStandby){
    require ('../algorithm/standby/giveStandby.php');
}
}//τρέχουμε για κάθε ΠΕ
/*τελειώνοντας αυτό το αρχείο δεν πρέπει να υπάρχουν ΠΕ διαθέσιμες,
ΜΟΝΟ ΠΕ διαθέσιμες που είναι σε αναμονή
όποτε πλέον δίνουμε αυτές*/
require ('../algorithm/giveThesis/stayStandbyThesis.php');
```

?>

```

<?php
    $preferenceValue=0;
    $idArray2=array();
    $dis=0;
    /*βλέπουμε αν έχει μέγεθος ο πίνακας idStayStandby
    στον πίνακα αυτόν αποθηκεύονται όλα τα θέματα που είναι σε αναμονή
    Αν έχει μέγεθος σημαίνει πως υπάρχουν ακόμα ΠΕ σε αναμονή*/
    $sql="SELECT id AS sum FROM dissertations WHERE status=0 AND
typeassign='Ατομική';//για ατομικές
    $result=mysql_query($sql);
    while($row=mysql_fetch_array($result)){
        $idArray2[]=$id;
    }
    //πρώτα τρέχουμε για τις ατομικές - μετά για τις ομαδικές
    //τρέχουμε για όσες ΠΕ βρήκαμε ότι είναι ακόμα σε αναμονή
    for($i=0;$i<sizeof($idArray2);$i++){
        $id=array_shift($idArray2);
        require ('../algorithm/findValues/findTitleProf.php');//βρίσκουμε
στοιχεία του θέματος
        require ('../algorithm/findValues/findMaxStudent.php');//βρίσκουμε
τον φοιτητή με την καλύτερη βαθμολογία
        require ('../algorithm/txtFiles/writeTxt.php');//γράφουμε τα
αποτελέσματα
        require ('../algorithm/updateFields/insertResults.php');//τα εισάγουμε στη ΒΔ
        require ('../algorithm/findValues/findPreference.php');//βρίσκουμε
σε ποια προτίμηση την είχε ο φοιτητής για στατιστικούς λόγους
        require ('../algorithm/updateFields/setDate.php');//θέτουμε
ημερομηνία ανάθεσης
        $givenAMArray[]=$$am;
        $dis=$id;
        $givenID[]=$id;
    }

    $sql="SELECT id FROM dissertations WHERE status=0 AND
typeassign='Ομαδική';//για ομαδικές
    $result=mysql_query($sql);
    while($row=mysql_fetch_array($result)){
        require ('../algorithm/findValues/findTitleProf.php');
        require ('../algorithm/findValues/findMaxTeamThesis.php');
        $am=$am1.".$am2;
        $dis=$id;
        require ('../algorithm/txtFiles/writeTxt.php');
        require ('../algorithm/updateFields/insertResults.php');
        require
('..../algorithm/updateFields/updateGivenTeamThesisFields.php');
        require ('../algorithm/updateFields/updateSumChoices.php');
        require ('../algorithm/updateFields/updateCandidacyStatus.php');
        require ('../algorithm/findValues/findPreference.php');
        require ('../algorithm/updateFields/setDate.php');
        $givenAMArray[]=$$am;

```

```
        $givenID[]=$id;  
    }  
}
```

?>

ΠΑΡΑΡΤΗΜΑ 6

<?php

```
    $preferenceValue=0;  
    $preference=0;  
    $dis=0;  
    $student=0;  
    //βρίσκουμε αμ που δεν πήραν ΠΕ  
    $notGivenAMArray=array();  
    foreach($candidacyArray as $am){  
        $find=0;  
        if(in_array($am, $givenAMArray)){  
            $find=1;  
        }  
        if($find==0){  
            $notGivenAMArray[]=$am;  
        }  
    }  
  
    //βρίσκουμε ΠΕ που δεν δόθηκαν  
    $notGivenIDArray=array();  
    foreach($totalIDArray as $id){  
        $find=0;  
        if(in_array($id, $givenID)){  
            $find=1;  
        }  
        if($find==0){  
            $notGivenIDArray[]=$id;  
        }  
    }  
  
    //θέτουμε σε κατάσταση επανέλεγχου τις δηλώσεις των αμ που έχουν ΠΕ  
    που δεν δόθηκε  
    foreach($notGivenAMArray as $am){  
        foreach($notGivenIDArray as $value){  
            $id=$value;  
            require  
            ('../algorithm/updateFields/updateNotGivenAMStatus.php');  
        }  
    }
```

```

}
//μετράμε πόσοι ζητάν κάθε ΠΕ
foreach($notGivenIDArray as $value){
    $id=$value;
    require ('../algorithm/findValues/findTitleProf.php');
    $sql="SELECT COUNT(*) AS sum FROM candidacy WHERE
dissertation='".$id.'" AND status=2";
    $result=mysql_query($sql);
    if(!$result){
        echo(mysql_error());
        exit();
    }
    while($row=mysql_fetch_array($result)){
        $sum=$row['sum'];
        if($sum==1){//αν τη ζητάει ένας, ελέγχουμε αν ζητάει και
κάποια άλλη από τις ΠΕ που μείναν σε καλύτερη προτίμηση
            $dis=$id;
            require ('../algorithm/findValues/findStudent.php');
            require ('../algorithm/txtFiles/writeTxt.php');
            require ('../algorithm/updateFields/insertResults.php');
            require ('../algorithm/findValues/findPreference.php');
            require
('../algorithm/updateFields/updateGivenThesisFields.php');
            require ('../algorithm/updateFields/setDate.php');

            $givenID[]=$id;
            $givenAMArray[]=$sum;
        }
        elseif($sum>1){
            //αν τη ζητάει παραπάνω από ένας, εφαρμόζουμε τη διαδικασία των
προηγούμενων αρχείων
            $find=0;
            //βρίσκει αυτόν με την καλύτερη βαθμολογία για
την ΠΕ που είμαστε
            $sql2="SELECT student, preference,
dissertationScore FROM candidacy WHERE dissertation='".$id.'" AND
dissertationScore=(SELECT max(dissertationScore) FROM candidacy WHERE
dissertation='".$id.'" AND status=2)";
            $result2=mysql_query($sql2);
            while($row2=mysql_fetch_array($result2)){

                $preference=$row2['preference'];
                $am=$row2['student'];
            }
            $sql3="SELECT dissertation, preference FROM
candidacy WHERE student='".$am.'" AND status=2";
            $result3=mysql_query($sql3);
            if(!$result3){
                echo(mysql_error());
                exit();
            }
        }
    }
}

```

```

    }
    while($row3=mysql_fetch_array($result3)){
        if($preference>$row3['preference']){
            $find=1;

            $betterID=$row3['dissertation'];
        }
    }
    if($find==0){//αν δεν ζητάει ΠΕ σε καλύτερη
προτίμηση, παίρνει την ΠΕ που είμαστε
        require
('..algorithm/txtFiles/writeTxt.php');
        require ('..algorithm/updateFields/insertResults.php');
        require
('..algorithm/updateFields/updateGivenThesisFields.php');
        $dis=$id;
        require
('..algorithm/findValues/findPreference.php');
        require
('..algorithm/updateFields/setDate.php');
        $givenID[]=$id;
        $givenAMArray[]=$am;
    }
}
}
}
unset($notGivenIDArray);
$notGivenIDArray=array();
foreach($totalIDArray as $id){
    $find=0;
    if(in_array($id, $givenID)){
        $find=1;
    }
    if($find==0){
        $notGivenIDArray[]=$id;
    }
}
$_SESSION['notGivenID']=sizeof($notGivenIDArray);

unset($notGivenAMArray);
$notGivenAMArray=array();
foreach($candidacyArray as $am){
    $find=0;
    if(in_array($am, $givenAMArray)){
        $find=1;
    }
    if($find==0){
        $notGivenAMArray[]=$am;
    }
}
}

```

```
$_SESSION['notGivenAM']=sizeof($notGivenAMArray);  
$_SESSION['totalAM']=sizeof($givenAMArray);
```

?>

ΠΑΡΑΡΤΗΜΑ 7

<?php

```
/*ψάχνουμε να βρούμε αν έχει εγγραφές ο πίνακας candidacy.  
Αν έχει θεωρούμε ότι η διαδικασία των δηλώσεων ολοκληρώθηκε με επιτυχία*/  
require ('../algorithm/before/dbconnect.php');  
session_start();  
$_SESSION['test']=1;  
$sql="SELECT * FROM candidacy";  
$result=mysql_query($sql);  
if(!$result){  
    echo(mysql_error());  
    exit();  
}  
$totalRows=mysql_num_rows($result);  
$_SESSION['totalRows']=$totalRows;  
if($totalRows==0){  
    $_SESSION['error']=-1;  
    header('location: ../index.php');  
}  
  
$sql="SELECT DISTINCT student FROM candidacy";  
$result=mysql_query($sql);  
if(!$result){  
    echo(mysql_error());  
    exit();  
}  
$totalAM=mysql_num_rows($result);  
$sql="select distinct student2  
from candidacy  
where student2<>"  
and student2 not in  
(  
    select distinct student  
    from candidacy  
)";  
$result=mysql_query($sql);  
while($row=mysql_fetch_array($result)){  
    $teamStudents++;  
}  
$_SESSION['totalAM2']=$totalAM+$teamStudents++;
```



```
$sql="SELECT id FROM dissertations";
$result=mysql_query($sql);
if(!$result){
    echo(mysql_error());
    exit();
}
$totalID=mysql_num_rows($result);
$_SESSION['totalID']=$totalID;
header('location: ../index.php');
```

?>

ΠΑΡΑΡΤΗΜΑ 8

<?php

```
session_start();
$_SESSION['run']=1;
require ('../algorithm/main.php');//το σημείο που καλείται ο αλγόριθμος με όλα τα
αρχεία του
//header('location: ../index.php');//μόλις εκτελεστούν όλα γυρνάμε στην αρχική
σελίδα
```

?>

ΠΑΡΑΡΤΗΜΑ 9

<?php

```
//θέτουμε όλα τα στοιχεία της βάσης που χρειαζόμαστε στις αρχικές τους τιμές
session_start();
require ('../algorithm/before/dbconnect.php');
$_SESSION['reset']=1;//θέτουμε την μεταβλητή σε κατάσταση reset box, έτσι
ώστε να εμφανιστεί το αντίστοιχο μήνυμα
$sql="UPDATE dissertations SET status=0, standby=0, sumChoices=0,
scoreAVG=0";
$result=mysql_query($sql);
$sql="UPDATE candidacy SET status=0, dissertationScore=0";
$result=mysql_query($sql);
header('location: ../index.php');
```

?>

