

ΣΥΓΧΡΟΝΑ ΘΕΜΑΤΑ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ



Τελική Εργασία

Ημερομηνία Παράδοσης: 02/03/2017

Ακαδημαϊκό Έτος 2016 – 2017

Περιεχόμενα

1. Περιγραφή Εφαρμογής
2. Ανάπτυξη μέρους του μοντέλου ανάπτυξης λογισμικού χρησιμοποιώντας 2 διαγράμματα της UML:
 - 2.1. Διάγραμμα Τάξεων (Class Diagram)
 - 2.2. Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram)
3. Ανάπτυξη του project που ορίζεται από την εργασία σας χρησιμοποιώντας τη μέθοδο της δομημένης ανάλυσης και σχεδιασμού:
 - 3.1. Λειτουργικές Απαιτήσεις
 - 3.2. Διάγραμμα Ροής Δεδομένων Επιπέδου 0
 - 3.3. Διάγραμμα Ροής Δεδομένων Επιπέδου 1
 - 3.4. Διάγραμμα Ροής Δεδομένων Επιπέδου 2
 - 3.5. Διάγραμμα Δομής Προγράμματος και Ψευδοκώδικας

1. Περιγραφή Εφαρμογής

Η εργασία που αναπτύξαμε είναι μια web εφαρμογή που ονομάζεται **VITA**, μια συντομογραφική ονομασία του Curriculum Vitae.

Η εφαρμογή είναι υλοποιημένη με PHP, CSS, HTML και MySQL. Πρόκειται για μια πλατφόρμα καταχώρησης βιογραφικών, στην οποία οι φοιτητές μπορούν να υποβάλλουν το βιογραφικό τους σημείωμα σε μορφή Pdf και οι διαχειριστές της σελίδας μπορούν να πραγματοποιούν αναζητήσεις στους καταχωρημένους φοιτητές, βάσει κριτηρίων που αυτοί ορίζουν.

Αρχικά, απαραίτητη προϋπόθεση για να υποβάλλει ένας φοιτητής το βιογραφικό του σημείωμα είναι να εγγραφεί στο σύστημα. Η εγγραφή στο σύστημα γίνεται με την εξής διαδικασία: ο χρήστης συμπληρώνει τα στοιχεία FullName, Subject και Email στην ειδική φόρμα της "Εγγραφής" και πατώντας Send, στέλνει το σύστημα αυτόματα Email στο καταχωρημένο Email του διαχειριστή. Με τη σειρά του ο διαχειριστής στέλνει στο Email που έχει υποβάλλει ο φοιτητής, τα απαραίτητα στοιχεία (username, password) και ο φοιτητής πλέον μπορεί να κάνει Log In συμπληρώνοντας τα πεδία με τα στοιχεία αυτά.

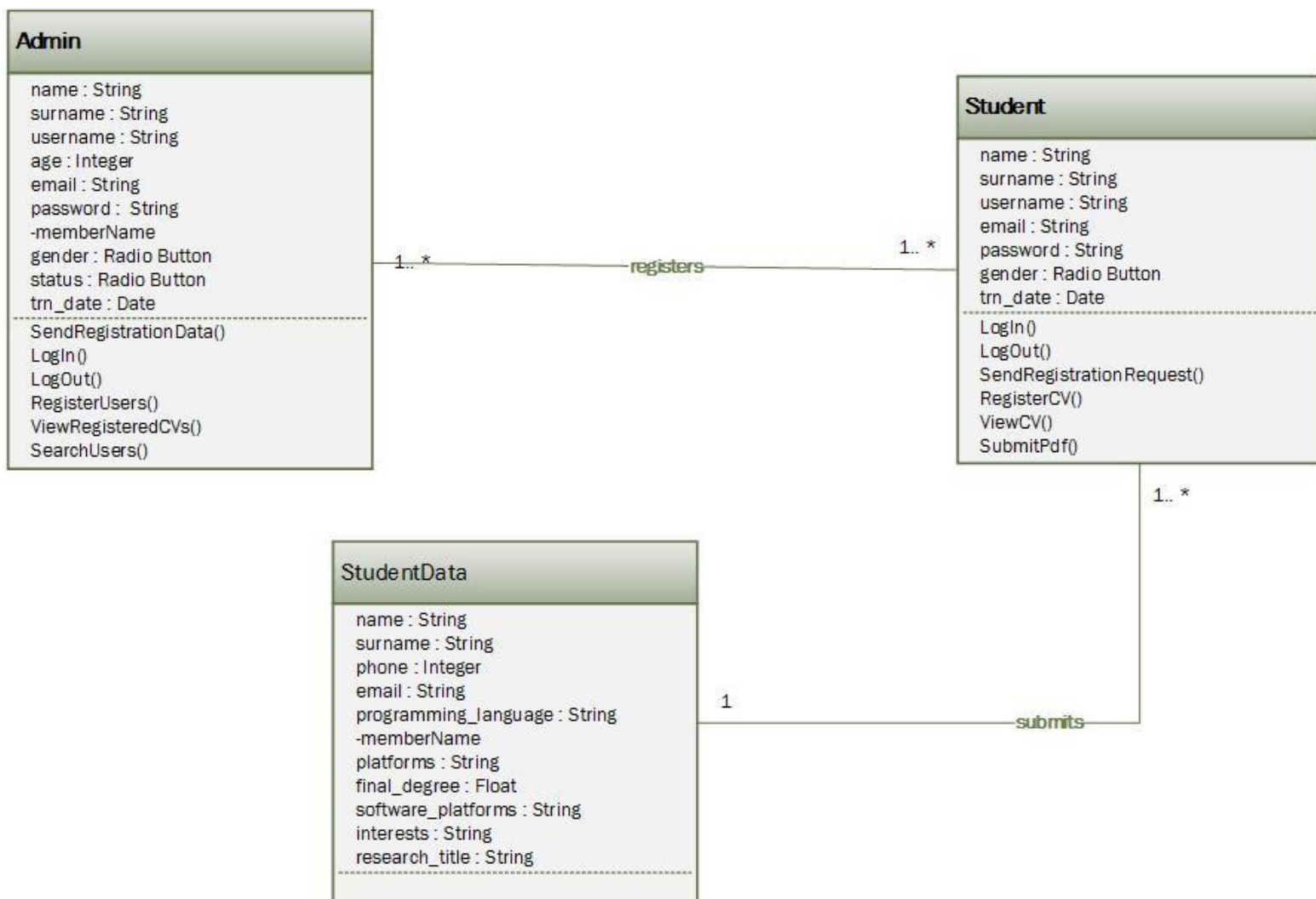
2. Ανάπτυξη μέρους του μοντέλου ανάπτυξης λογισμικού χρησιμοποιώντας 2 διαγράμματα της UML.

Για τον σχεδιασμό των παρακάτω διαγραμμάτων της UML χρησιμοποιήθηκε το πρόγραμμα Microsoft Visio.

2.1. Διάγραμμα Τάξεων (Class Diagram)

Το διάγραμμα τάξεων αναπαριστά μια στατική δομή της εφαρμογής. Χρησιμοποιείται για να περιγράψει τα χαρακτηριστικά και τις λειτουργίες μιας κλάσης, καθώς και τις σχέσεις μεταξύ των κλάσεων.

Οι τάξεις αναπαρίστανται ως ορθογώνια τα οποία περιέχουν διαχωρισμούς. Το πρώτο τμήμα του ορθογώνιου περιέχει το όνομα της τάξης, το δεύτερο τα χαρακτηριστικά της και το τρίτο τις λειτουργίες της.

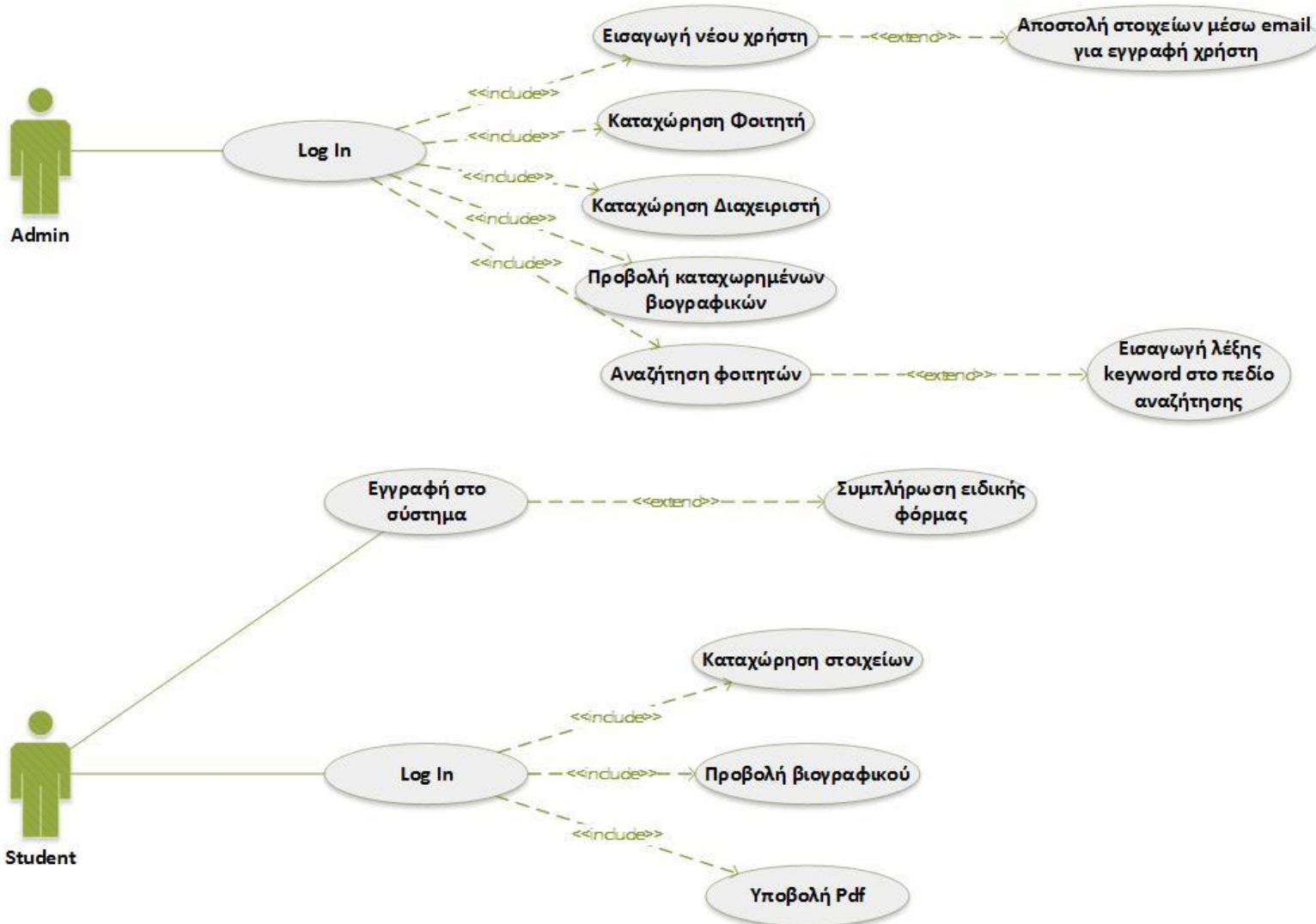


2.2. Διάγραμμα Περιπτώσεων Χρήσης (Use case Diagram)

Το διάγραμμα περιπτώσεων χρήσης περιγράφουν τη συμπεριφορά ενός συστήματος από την οπτική γωνία ενός χρήστη. Το μοντέλο περιπτώσεων χρήσης περιλαμβάνει:

- τις ίδιες τις περιπτώσεις χρήσης
- τους ενεργοποιούς (actors)
- και την σχέση μεταξύ των δυο παραπάνω

Οι ενεργοποιοί συμβολίζονται με ανθρωπάκια που ενεργοποιούν τις περιπτώσεις χρήσεις. Οι περιπτώσεις χρήσης συμβολίζονται με ελλείψεις.



3. Ανάπτυξη του project χρησιμοποιώντας τη μέθοδο της δομημένης ανάλυσης και σχεδιασμού.

3.1 Λειτουργικές απαιτήσεις:

1. ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ ΧΡΗΣΤΗ

ΠΕΡΙΓΡΑΦΗ: Κάθε χρήστης της εφαρμογής την πρώτη φορά που θα χρησιμοποιήσει την εφαρμογή θα πρέπει να συμπληρώσει την ειδική φόρμα αποστολής e-mail.

ΕΙΣΟΔΟΣ: Ο χρήστης εισάγει τα στοιχεία του (όνομα, επώνυμο, θέμα και διεύθυνση e-mail)

.

ΕΠΕΞΕΡΓΑΣΙΑ: Αν πρόκειται για νέο χρήστη η εφαρμογή αποστέλλει μήνυμα στο διαχειριστή, ο οποίος εισάγει στο σύστημα τον νέο χρήστη και ορίζει την ιδιότητα του (αν θα είναι διαχειριστής ή φοιτητής).

ΕΞΟΔΟΣ: Ο διαχειριστής, ως υπεύθυνος για τη διαχείριση χρηστών πρέπει να αποστέλλει e-mail με τα στοιχεία που έχει παραχωρήσει στα νέα μέλη (username, password). Μετά την επιτυχή είσοδο ενός χρήστη, το σύστημα εμφανίζει σε αυτόν ένα αντίστοιχο μενού λειτουργιών, ανάλογα με το ρόλο του εκάστοτε χρήστη (μενού λειτουργιών φοιτητή, μενού λειτουργιών διαχειριστή). Σε περίπτωση ανεπιτυχούς εισόδου, ο χρήστης λαμβάνει μήνυμα αποτυχημένης προσπάθειας εισόδου.

2. ΥΠΟΒΟΛΗ PDF ΚΑΙ ΣΥΜΠΛΗΡΩΣΗ ΒΙΟΓΡΑΦΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΑΠΟ ΤΟ ΦΟΙΤΗΤΗ

ΠΕΡΙΓΡΑΦΗ: Κάθε χρήστης με ιδιότητα student που χρησιμοποιεί την εφαρμογή θα πρέπει να συμπληρώσει την ειδική φόρμα με τα στοιχεία του βιογραφικού του και να υποβάλλει το resume του σε μορφή pdf.

ΕΙΣΟΔΟΣ: Ο χρήστης εισάγει τα στοιχεία του (όνομα, επώνυμο, κινητό, e-mail, γλώσσες προγραμματισμού, πλατφόρμες, τελικός βαθμός πτυχίου, τίτλος πτυχιακής, πλατφόρμες λογισμικού και ενδιαφέροντα καθώς και το pdf) .

ΕΠΕΞΕΡΓΑΣΙΑ: Αν πρόκειται για νέο χρήστη, το σύστημα ελέγχει βάσει username αν ο φοιτητής έχει ήδη καταχωρήσει το cv του, αλλιώς εισάγει τα στοιχεία.

ΕΞΟΔΟΣ: Το σύστημα εμφανίζει μήνυμα επιτυχούς ή ανεπιτυχούς εισαγωγής βιογραφικών στοιχείων.

3. ΠΡΟΒΟΛΗ ΤΩΝ ΚΑΤΑΧΩΡΗΜΕΝΩΝ ΒΙΟΓΡΑΦΙΚΩΝ ΑΠΟ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ ΚΑΙ ΠΡΑΓΜΑΤΟΠΟΙΗΣΗ ΑΝΑΖΗΤΗΣΕΩΝ ΣΤΟΥΣ ΚΑΤΑΧΩΡΗΜΕΝΟΥΣ ΦΟΙΤΗΤΕΣ

ΠΕΡΙΓΡΑΦΗ: Αν ο χρήστης έχει την ιδιότητα administrator μπορεί πέρα από το να εισάγει νέους χρήστες, να δει όλους τους καταχωρημένους φοιτητές και τα βιογραφικά τους και να πραγματοποιήσει αναζητήσεις βάσει κριτηρίων που ορίζει αυτός.

ΕΙΣΟΔΟΣ: Ο διαχειριστής βλέπει σε ένα πίνακα όλους τους καταχωρημένους χρήστες και εισάγει σε ένα πεδίο την λέξη keyword με βάση την οποία θα γίνει η αναζήτηση.

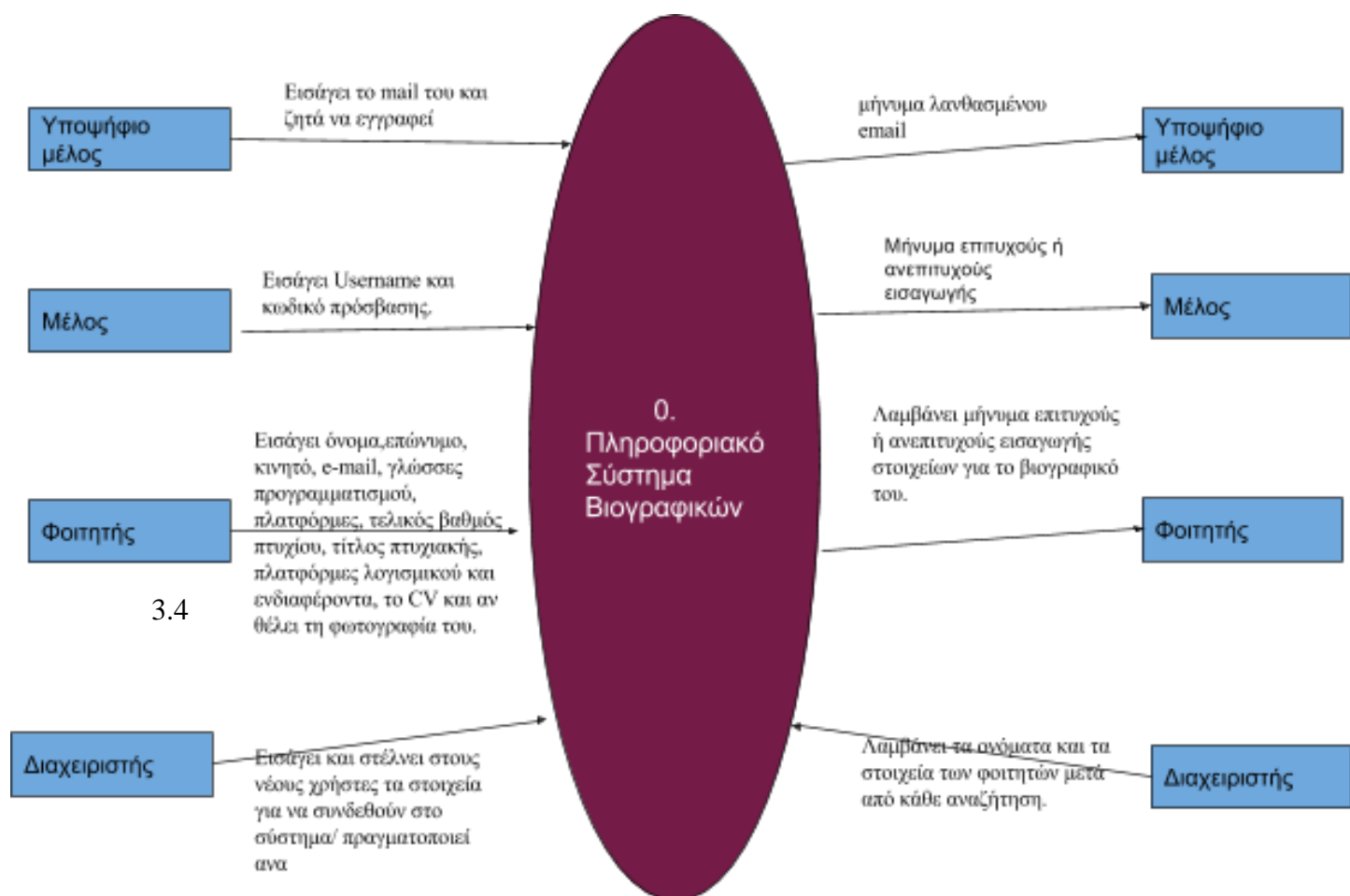
ΕΠΕΞΕΡΓΑΣΙΑ: Το σύστημα αναζητά φοιτητές που έχουν δηλώσει στις ικανότητές τους και στα ενδιαφέροντα τους τη συγκεκριμένη λέξη. (πχ. Ο διαχειριστής αναζητά όλους τους φοιτητές που ξέρουν java ή που έχουν ΜΟ. 8).

ΕΞΟΔΟΣ: Το σύστημα εμφανίζει τα ονόματα και τα στοιχεία των φοιτητών που πληρούν την παραπάνω προϋπόθεση.

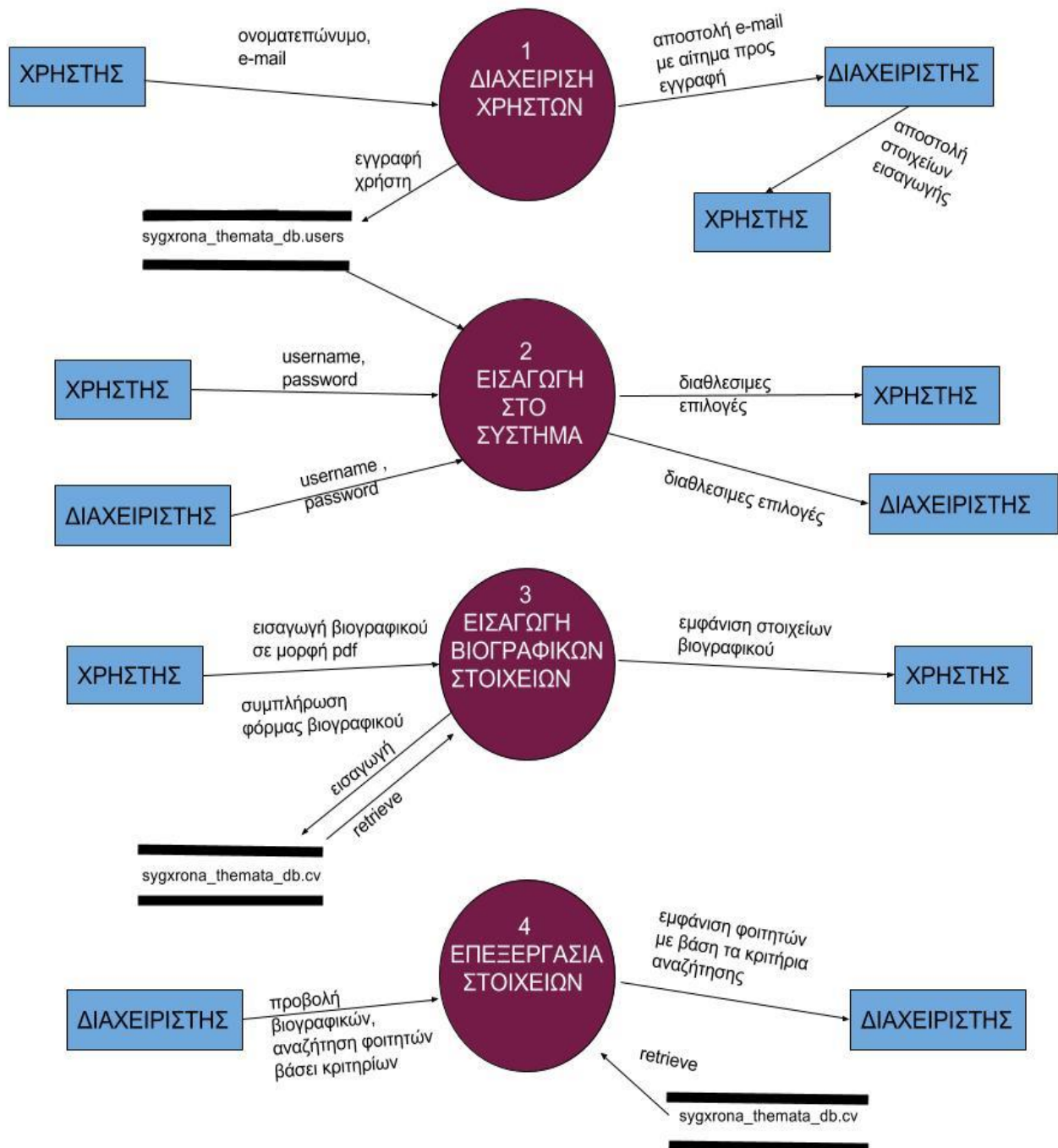
3.2. Διάγραμμα Ροής Δεδομένων Επιπέδου 0

Πηγές και αποδέκτες:

Πηγές:	
Υποψήφιο μέλος	Εισάγει τα στοιχεία του.
Μέλος	Εισάγει Username και κωδικό πρόσβασης.
Φοιτητής	Εισάγει όνομα, επώνυμο, κινητό, e-mail, γλώσσες προγραμματισμού, πλατφόρμες, τελικός βαθμός πτυχίου, τίτλος πτυχιακής, πλατφόρμες λογισμικού, ενδιαφέροντα και το CV
Διαχειριστής	Εισάγει και στέλνει στους νέους χρήστες τα στοιχεία για να συνδεθούν στο σύστημα.
Αποδέκτες:	
Υποψήφιο μέλος	Λαμβάνει τον αρχικό κωδικό πρόσβασης ή μήνυμα για ανεπιτυχή εγγραφή.
Φοιτητής	Λαμβάνει μήνυμα επιτυχούς ή ανεπιτυχούς εισαγωγής στοιχείων για το βιογραφικό του.
Διαχειριστής	Λαμβάνει τα ονόματα και τα στοιχεία των φοιτητών μετά από κάθε αναζήτηση.

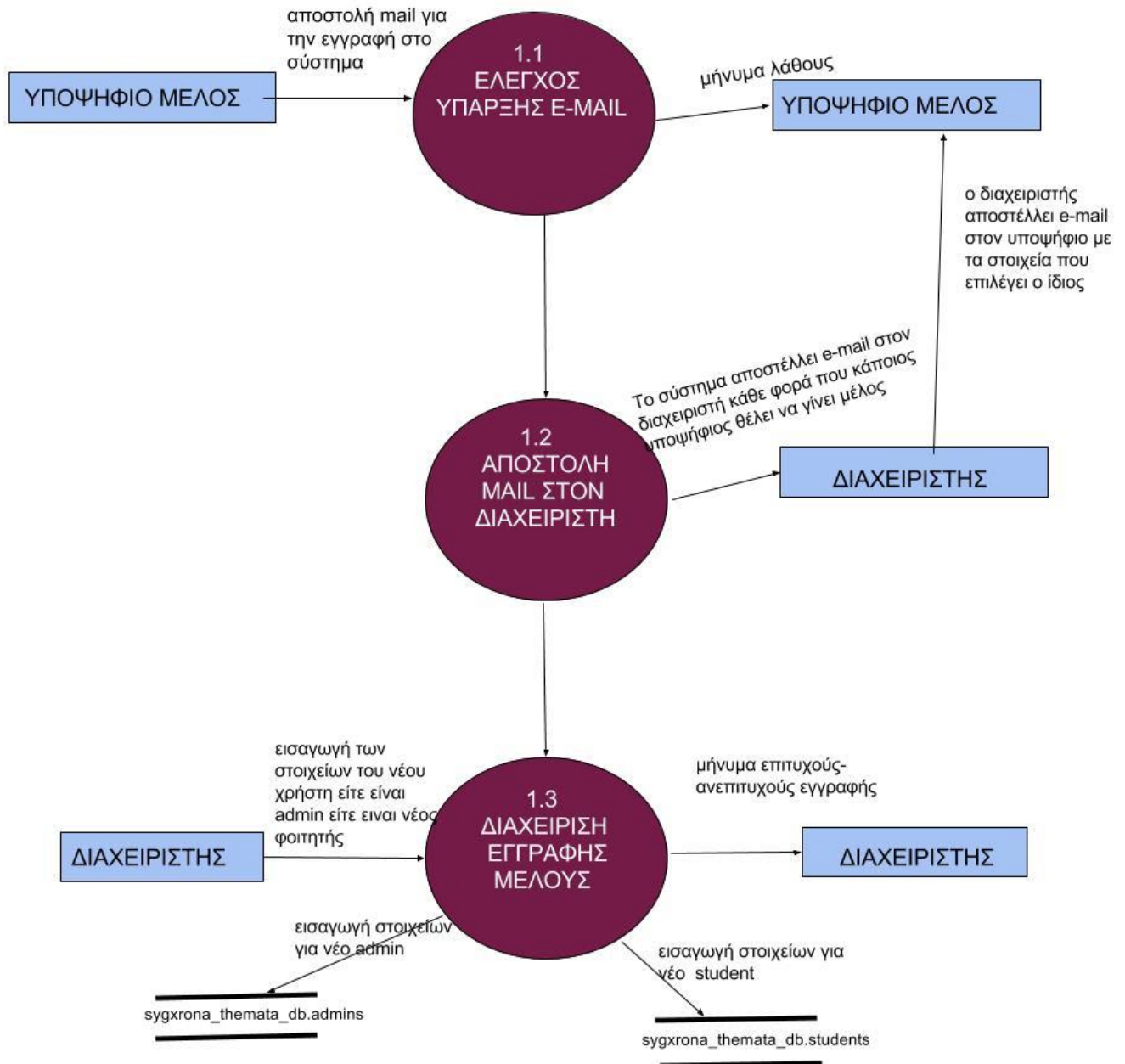


3.3. Διάγραμμα Ροής Δεδομένων Επιπέδου 1

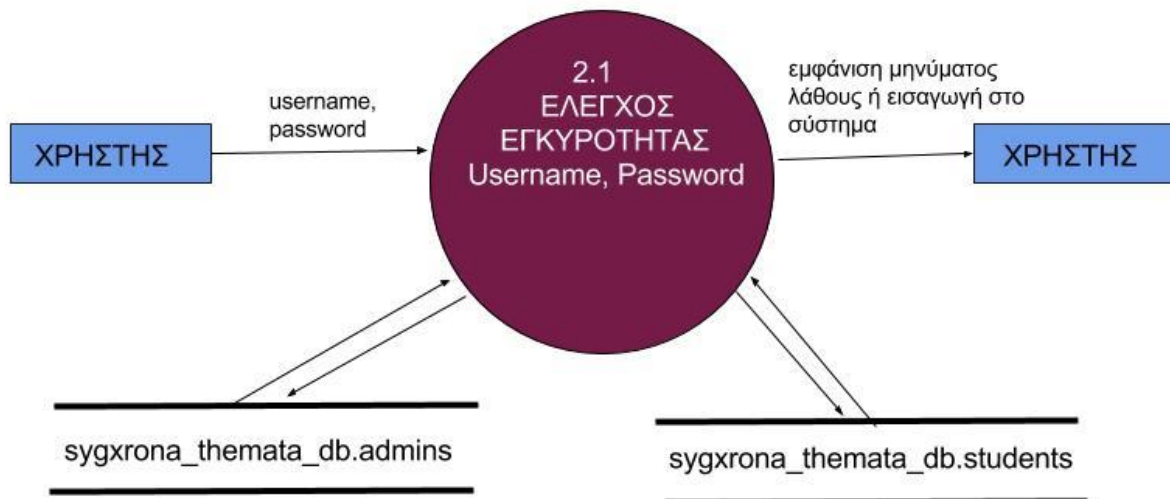


3.4. Διάγραμμα Ροής Δεδομένων Επιπέδου 2

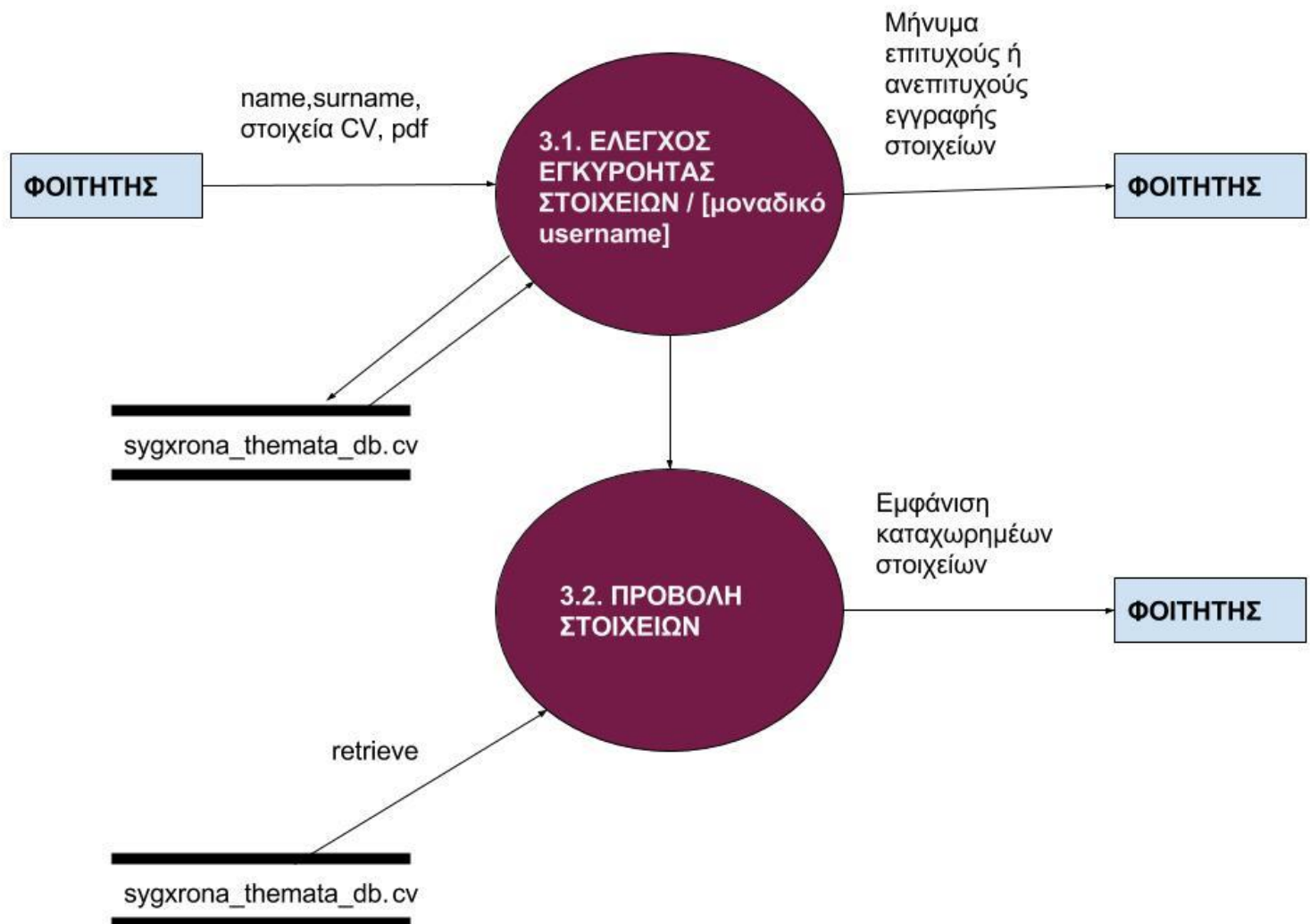
α) Για τη διαχείριση των χρηστών



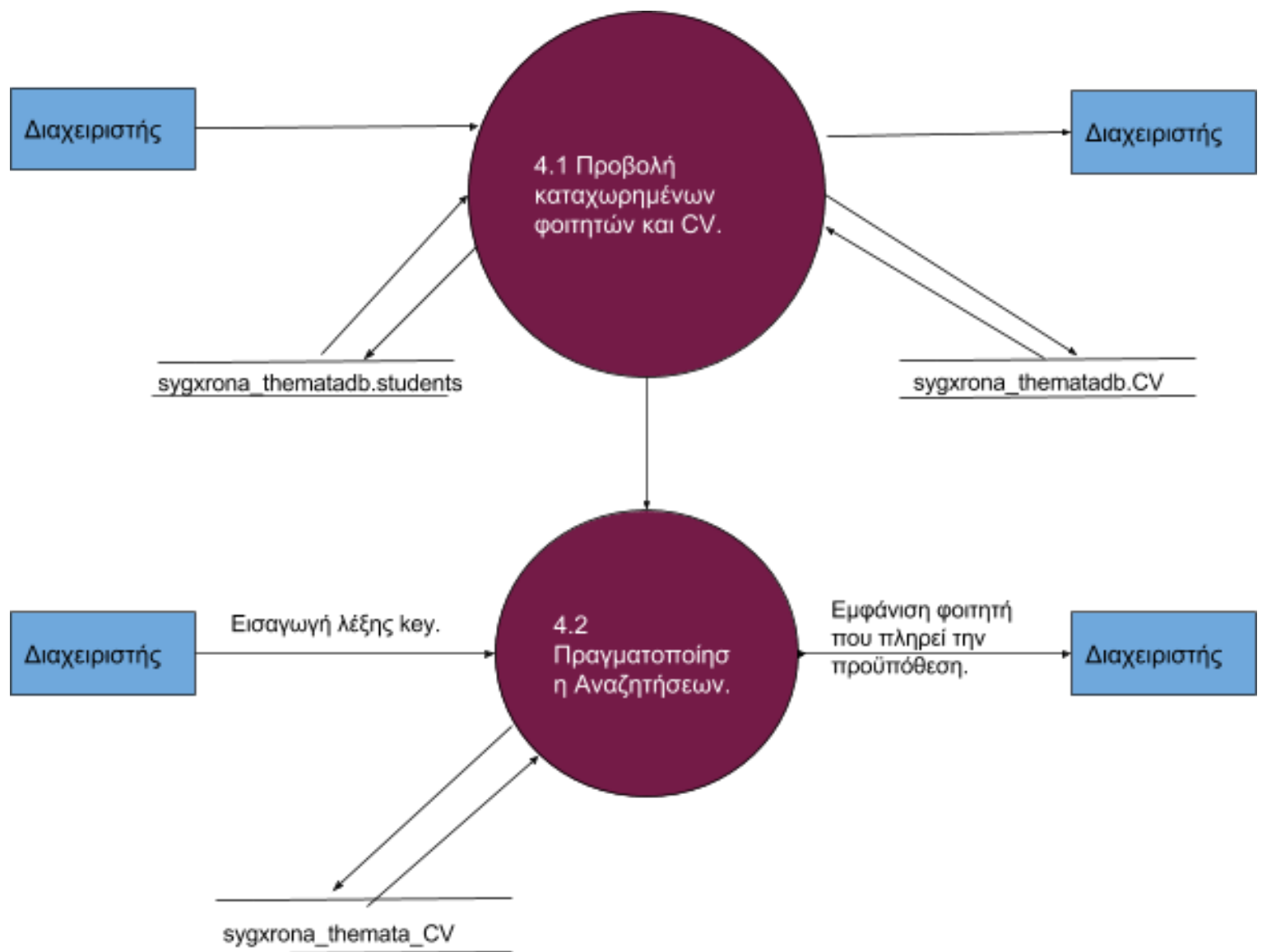
β) Για την εισαγωγή στο σύστημα



γ) Για την εισαγωγή βιογραφικών στοιχείων



δ) Για την επεξεργασία στοιχείων

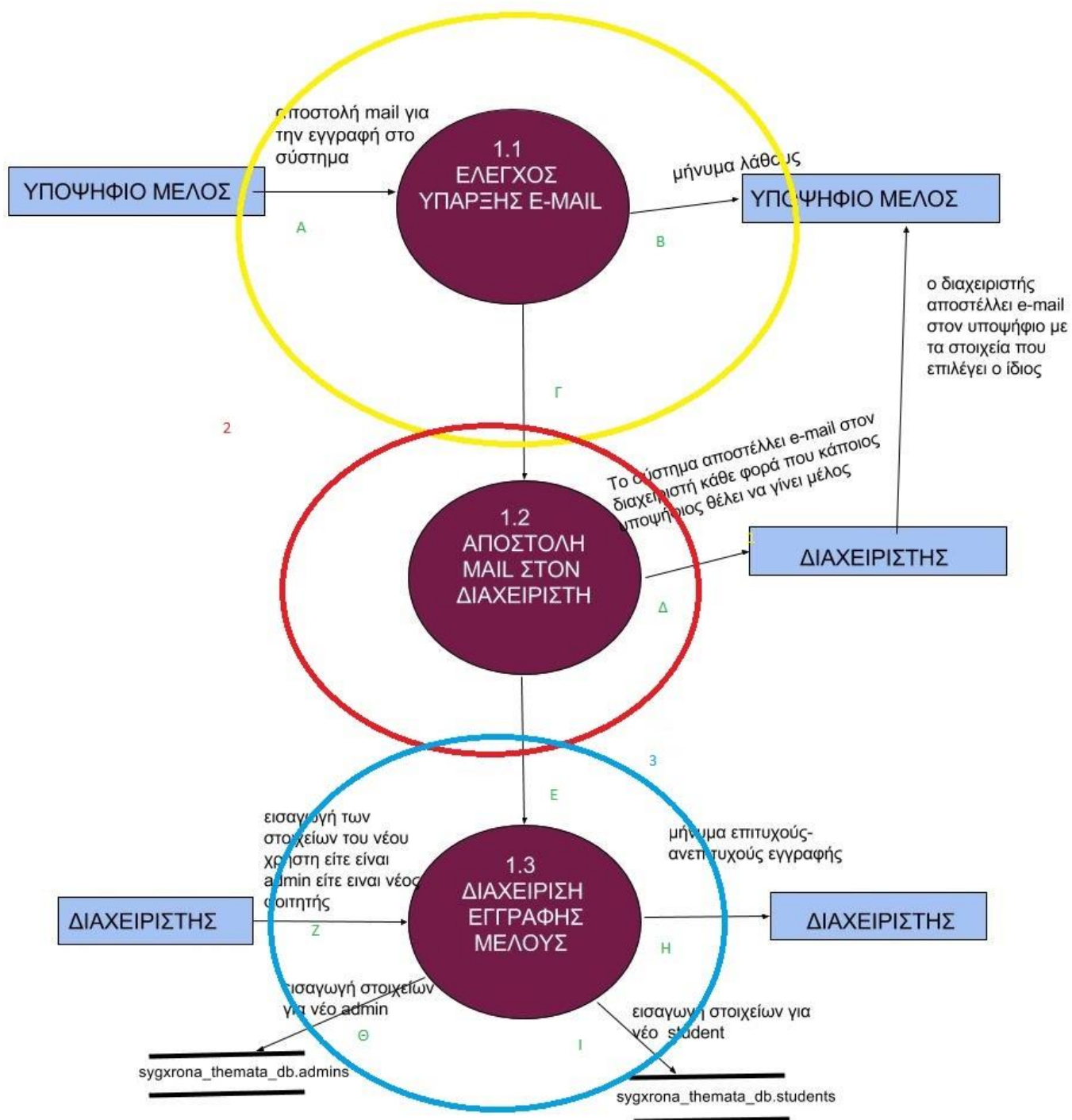


3.5 Διάγραμμα Δομής Προγράμματος και Ψευδοκώδικας

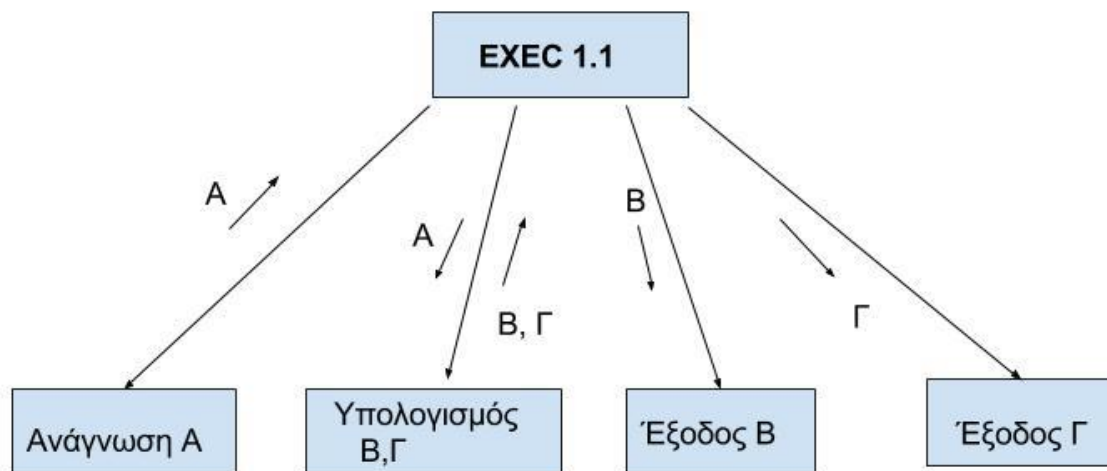
3.5.1.

Για τις λειτουργίες του λογισμικού που ασχολούνται με τη διαχείριση χρηστών παραθέσαμε το παρακάτω διάγραμμα επιπέδου 2.

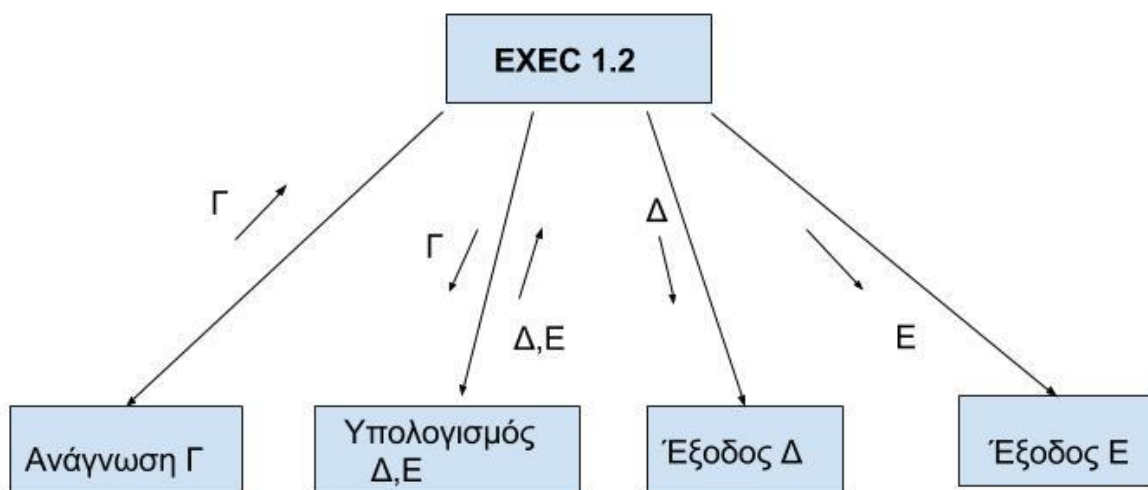
Για να απλοποιήσω την κατάσταση αντικαθιστώ ως εξής τις καταστάσεις με γράμματα του αλφαβήτου.

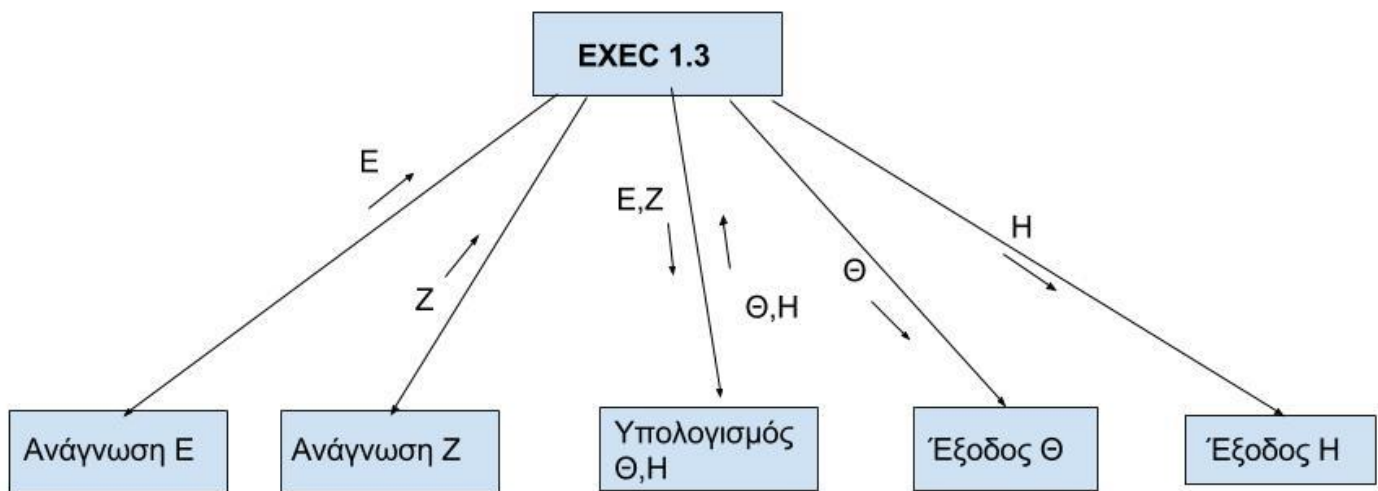


Κατόπιν εντοπίζουμε ως Κεντρικό Μετασχηματισμό τη διεργασία 1.1 και για αυτήν έχουμε:



Αντίστοιχα για τις διεργασίες 1.2 και 1.3 έχουμε τα εξής ΔΔΠ:





Σημείωση προηγουμένως αντικαταστήσαμε με γράμματα της αλφαβήτου τις μεταβάσεις ώστε να είναι πιο απλοϊκή η σχεδίαση του ΔΔΠ.

Έτσι Α=αποστολή e-mail για εγγραφή στο σύστημα

Β= μήνυμα επιτυχούς ή ανεπιτυχούς mail

Δ=e-mail στον διαχειριστή

Ζ=Εισαγωγή στοιχείων νέου χρήστη

Η=Μήνυμα επιτυχούς ή ανεπιτυχούς εγγραφής

Θ= Εισαγωγή στοιχείων νέου admin

Ι= Εισαγωγή στοιχείων νέου φοιτητή

Σε αυτό το σημείο μπορούμε, αφού έχουμε ολοκληρώσει με τις μονάδες του ΔΔΠ, να διατυπώσουμε ψευδοκώδικα:

Για το ΔΔΠ 1.1

*/

PROCEDURE Exec 1.1

/*

LOCAL VAR Α,Β,Γ

Αρχικοποίησε Α

WHILE Α <> κενό DO

CALL ΑΝΑΓΝΩΣΗ (Α)

```
IF A <> KENO THEN  
CALL ΥΠΟΛΟΓΙΣΜΟΣ(Β,Γ)  
CALL ΕΞΟΔΟΣ(Β,Γ)
```

```
END_IF
```

```
END_WHILE
```

```
END_PROCEDURE
```

Για το ΔΔΠ 1.2

```
*/  
PROCEDURE Exec 1.2  
/*  
LOCAL VAR Γ,Δ,Ε  
Αρχικοποίησε Γ
```

```
WHILE Γ <> κενό DO
```

```
CALL ΑΝΑΓΝΩΣΗ (Δ,Ε)  
IF Γ <> KENO THEN  
CALL ΥΠΟΛΟΓΙΣΜΟΣ(Δ,Ε)  
CALL ΕΞΟΔΟΣ(Δ,Ε)
```

```
END_IF
```

```
END_WHILE
```

```
END_PROCEDURE
```

Για το ΔΔΠ 1.3

```
*/  
PROCEDURE Exec 1.3  
/*  
LOCAL VAR Ε,Ζ,Θ,Η  
Αρχικοποίησε Ε,Ζ
```

```
WHILE Ε,Ζ<> κενό DO
```

```
CALL ΑΝΑΓΝΩΣΗ (Ε,Ζ)  
IF Ε,Ζ<> KENO THEN
```

```
CALL ΥΠΟΛΟΓΙΣΜΟΣ(Η,Θ)  
CALL ΕΞΟΔΟΣ(Η,Θ)
```

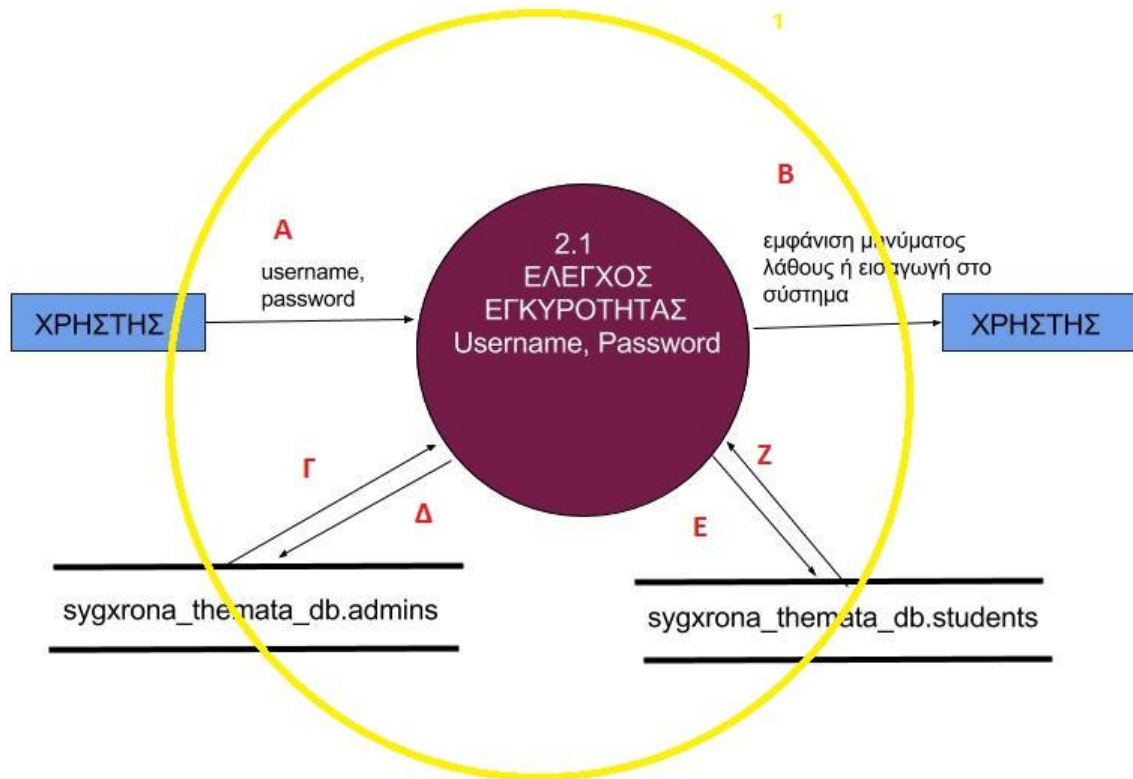
```
END_IF
```

```
END_WHILE
```

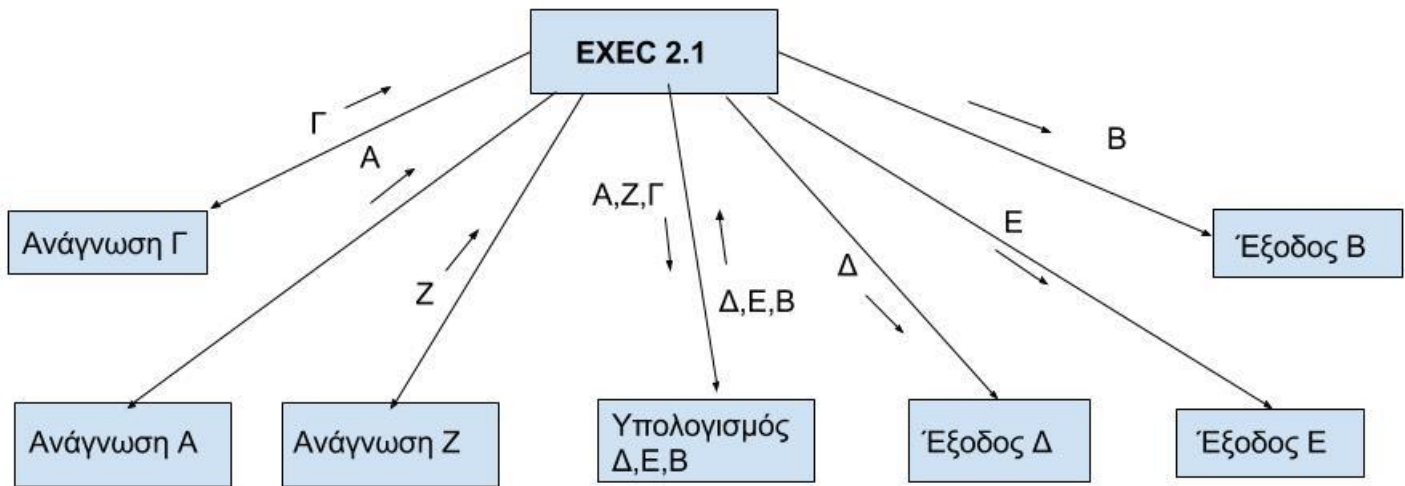
```
END_PROCEDURE
```

3.5.2.

Για τις λειτουργίες του λογισμικού που ασχολούνται με την εισαγωγή στο σύστημα παραθέσαμε το παρακάτω διάγραμμα επιπέδου 2.



Κατόπιν εντοπίζουμε ως Κεντρικό Μετασχηματισμό τη διεργασία 2.1 και για αυτήν έχουμε:



Σημείωση προηγουμένως αντικαταστήσαμε με γράμματα της αλφαβήτου τις μεταβάσεις ώστε να είναι πιο απλοϊκή η σχεδίαση του ΔΔΠ.

Έτσι A=username, password

B= μήνυμα επιτυχούς ή ανεπιτυχούς εισαγωγής στο σύστημα

Γ,Δ,Ε,Z = Ανάκτηση ή αναζήτηση από τη βάση

Για το ΔΔΠ 2.1.:

```

*/
PROCEDURE Exec 2.1
/*
LOCAL VAR A,Γ,Z,Δ,Ε,Β
Αρχικοποίησε Α,Γ,Z

WHILE A,Γ,Z <> κενό DO

CALL ΑΝΑΓΝΩΣΗ (Α,Γ,Z)
IF A,Γ,Z <> ΚΕΝΟ THEN
CALL ΥΠΟΛΟΓΙΣΜΟΣ(Δ,Ε,Β)
CALL ΕΞΟΔΟΣ(Δ,Ε,Β)

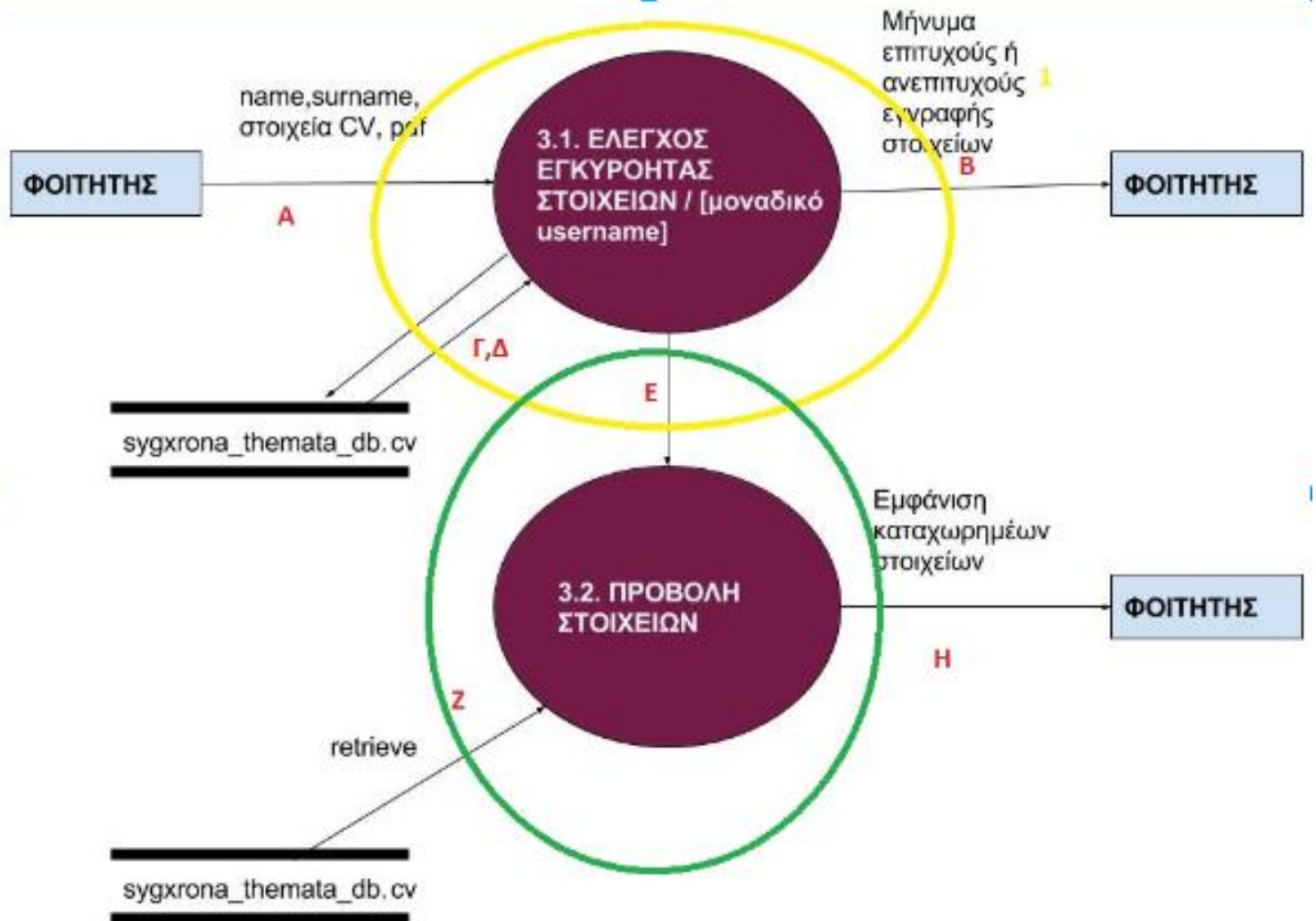
END_IF

END_WHILE

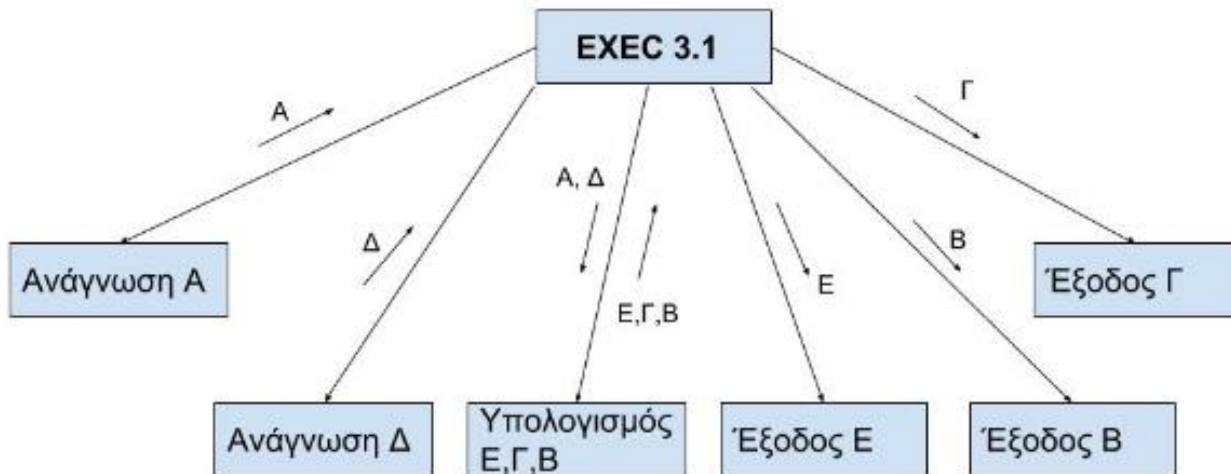
END_PROCEDURE
  
```

3.5.3.

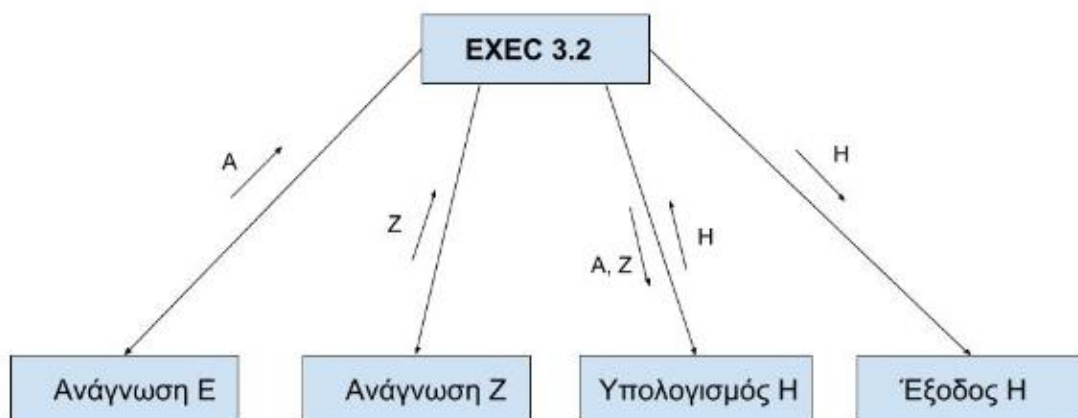
Για τις λειτουργίες του λογισμικού που ασχολούνται με την εισαγωγή βιογραφικών στοιχείων παραθέσαμε το παρακάτω διάγραμμα επιπέδου 2.



Κατόπιν εντοπίζουμε ως Κεντρικό Μετασχηματισμό τη διεργασία 3.1 και για αυτήν έχουμε:



και για την διεργασία 3.2 έχουμε:



Σημείωση προηγουμένως αντικαταστήσαμε με γράμματα της αλφαβήτου τις μεταβάσεις ώστε να είναι πιο απλοϊκή η σχεδίαση του ΔΔΠ.

Έτσι A=name, surname, στοιχεία cv , pdf

B= μήνυμα επιτυχούς ή ανεπιτυχούς εγγραφής στοιχείων

Γ,Δ=αλληλεπίδραση με τη βάση

E= retrieve

Z=Εμφάνιση καταχωρημένων στοιχείων

Για το ΔΔΠ 3.1

```
*/  
PROCEDURE Exec 3.1  
/*  
LOCAL VAR A,Δ,Γ,B,E  
Αρχικοποίησε A,Δ  
  
WHILE A,Δ <> κενό DO  
  
CALL ΑΝΑΓΝΩΣΗ (A,Δ)  
IF A,Δ<> KENO THEN  
CALL ΥΠΟΛΟΓΙΣΜΟΣ(B,Γ,E)  
CALL ΕΞΟΔΟΣ(B,Γ,E)  
  
END_IF  
  
END_WHILE  
  
END_PROCEDURE
```

Για το ΔΔΠ 3.2

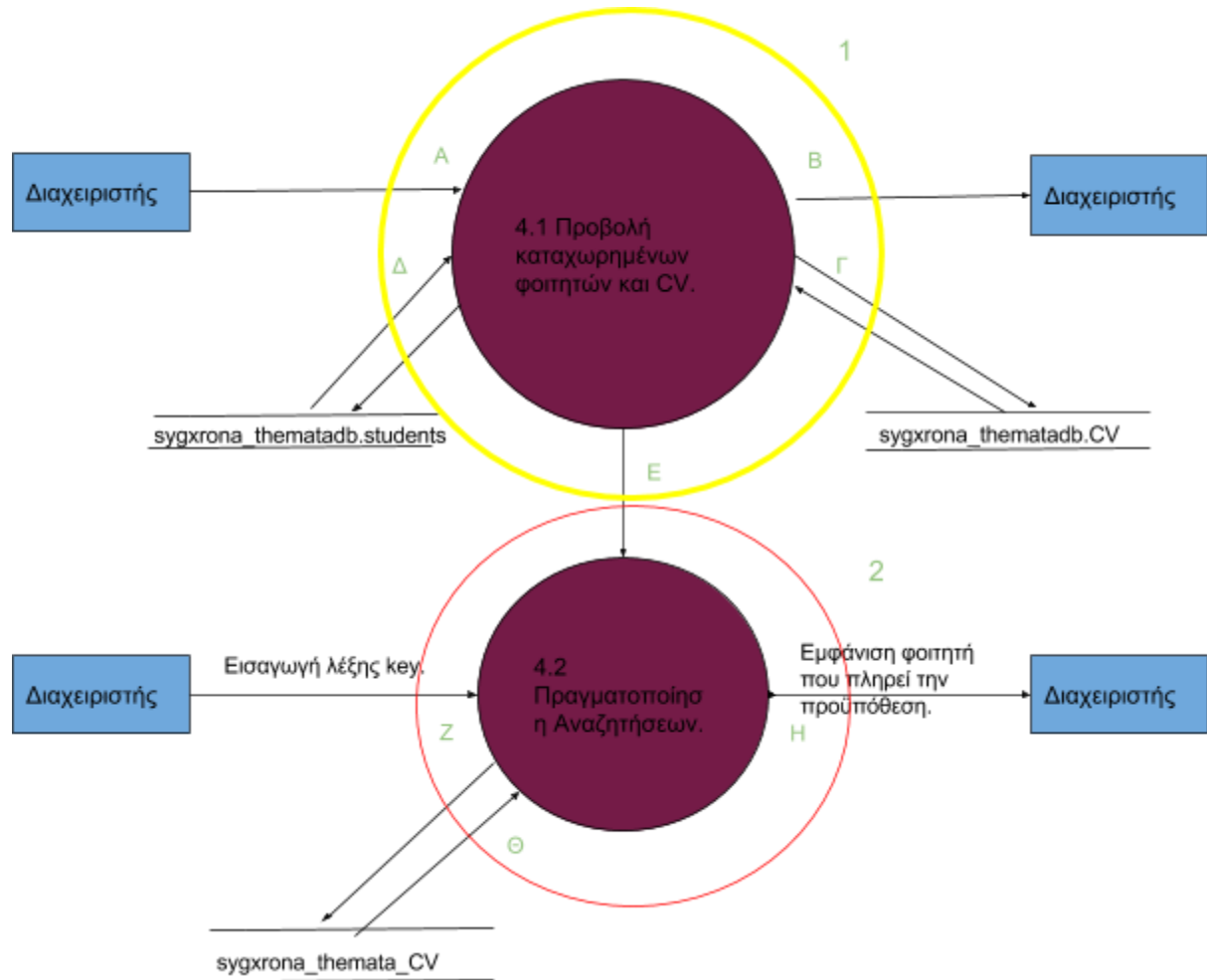
```
*/  
PROCEDURE Exec 3.2  
/*  
LOCAL VAR E,Z,H  
Αρχικοποίησε E,Z  
  
WHILE E,Z <> κενό DO  
  
CALL ΑΝΑΓΝΩΣΗ (E,Z)  
IF E,Z <> KENO THEN  
CALL ΥΠΟΛΟΓΙΣΜΟΣ(H)  
CALL ΕΞΟΔΟΣ(H)  
  
END_IF  
  
END_WHILE
```


END_PROCEDURE

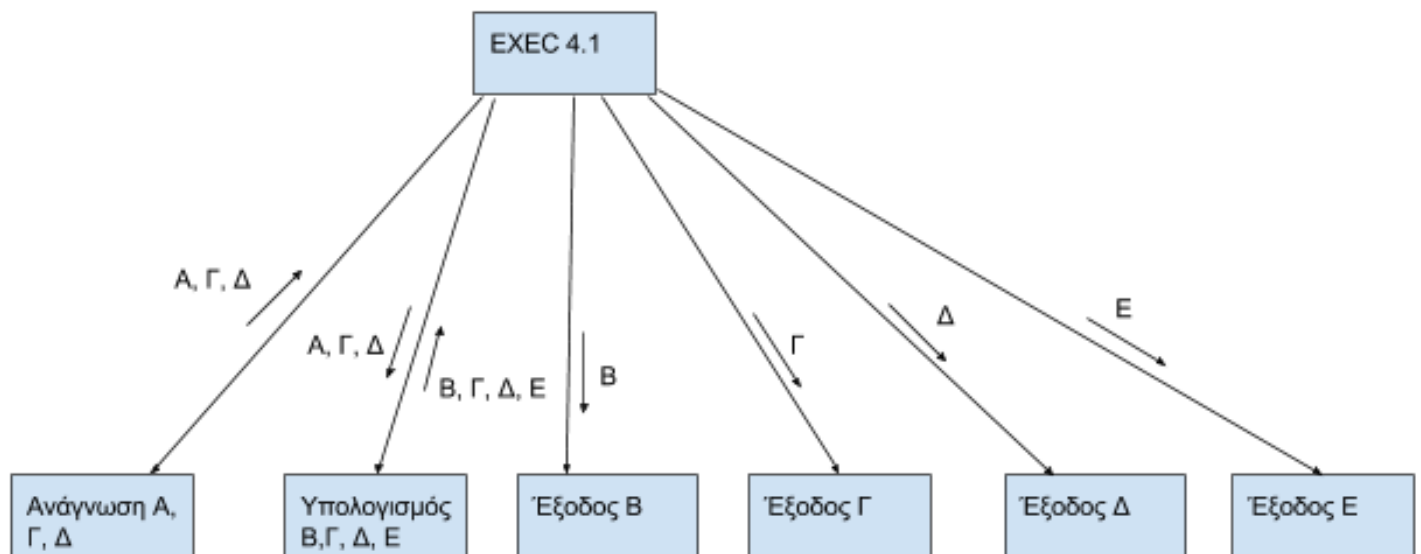
3.5.4

Για τις λειτουργίες του λογισμικού που ασχολούνται με την επεξεργασία των στοιχείων παραθέσαμε το παρακάτω διάγραμμα επιπέδου 2.

Για να απλοποιήσω την κατάσταση αντικαθιστώ τις παρακάτω καταστάσεις με γράμματα.



Κατόπιν εντοπίζουμε ως Κεντρικό Μετασχηματισμό τη διεργασία 4.1 και γι' αυτήν έχουμε ότι:



Για το ΔΔΠ 4.1

```

/*
PROCEDURE Exec 4.1
/*
LOCAL VAR Α,Γ,Δ, Β, Ε
Αρχικοποίησε Α,Γ,Δ
WHILE Α,Γ,Ζ <> κενό DO

CALL ΑΝΑΓΝΩΣΗ (Α,Γ,Δ)
IF Α,Γ,Δ <> ΚΕΝΟ THEN
CALL ΥΠΟΛΟΓΙΣΜΟΣ(Β, Γ, Δ, Ε)
CALL ΕΞΟΔΟΣ(Β, Γ, Δ, Ε)

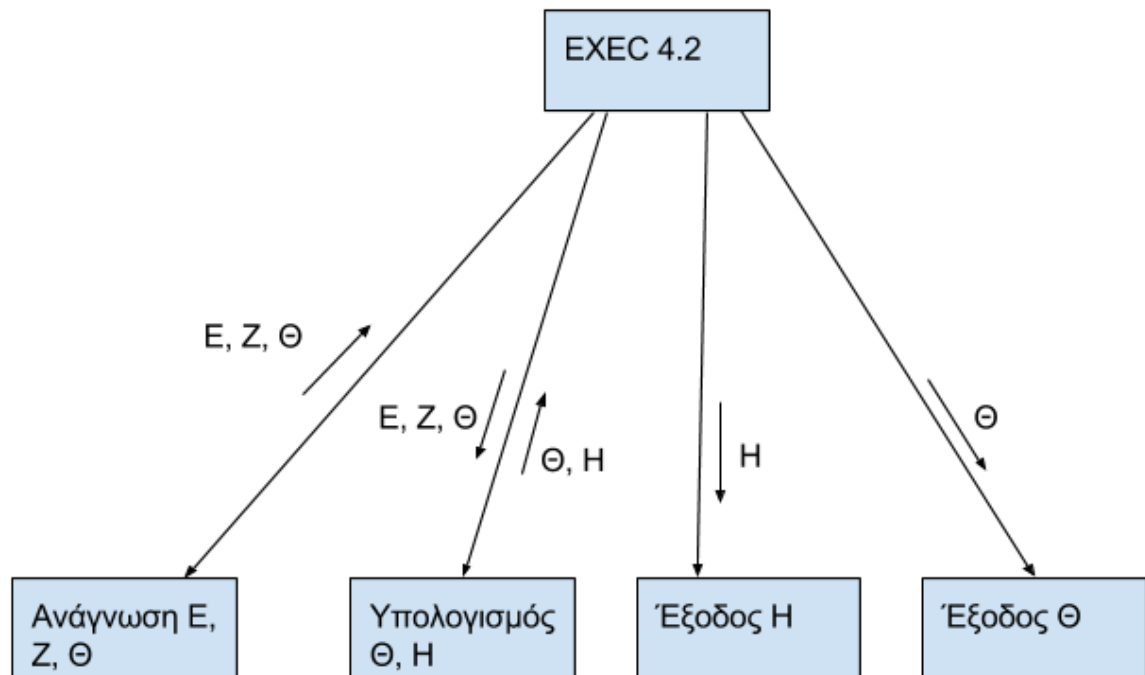
END_IF

END_WHILE

END_PROCEDURE

```

Για τη διεργασία 4.2 προκύπτει το παρακάτω ΔΔΠ:



Για το ΔΔΠ 4.2

```

*/
PROCEDURE Exec 4.2
/*
LOCAL VAR Ε, Ζ, Θ, Η
Αρχικοποίησε Ε, Ζ, Θ
WHILE Ε, Ζ, Θ <> κενό DO

CALL ΑΝΑΓΝΩΣΗ (Ε, Ζ, Θ)
IF Ε, Ζ, Θ <> ΚΕΝΟ THEN
CALL ΥΠΟΛΟΓΙΣΜΟΣ(Θ, Η)
CALL ΕΞΟΔΟΣ(Η, Θ)

END_IF

END_WHILE

END_PROCEDURE
  
```