

ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΩΝ ΛΟΓΙΣΜΙΚΟΥ

ΑΚΑΔΗΜΑΙΚΟ ΕΤΟΣ 2018-2019

Περιγραφή Συστήματος e-broker

ΜΕΛΗ:

ΓΙΑΝΝΗΣ ΓΙΑΝΝΑΚΙΔΗΣ 1115 2015 00025

ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΣΧΟΠΟΥΛΟΣ 1115 2015 00127

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

| | |
|--|----|
| ΕΙΣΑΓΩΓΗ..... | 4 |
| Σκοπός της εργασίας..... | 4 |
| Διαμοιρασμός εργασίας..... | 4 |
| Συνεργασία..... | 4 |
| Τελικό κείμενο-Υποβολή εργασίας..... | 5 |
| Επόμενα κεφάλαια..... | 5 |
| ΖΗΤΟΥΜΕΝΑ..... | 6 |
| ΔΟΜΗΜΕΝΗ ΑΝΑΛΥΣΗ..... | 6 |
| 1.Γενικό Διάγραμμα Ροής Δεδομένων..... | 7 |
| 2.Διαγράμματα Ροής Δεδομένων..... | 8 |
| 3.Προδιαγραφή διαδικασίας διαβίβασης εντολής σε Δομημένα Αγγλικά..... | 10 |
| 4.Προδιαγραφή διαδικασίας υπολογισμού της προμήθειας σε Πίνακα Απόφασης..... | 11 |
| 5.Προδιαγραφή διαδικασίας σύνδεσης σε Δένδρο Απόφασης..... | 12 |
| 6.Προδιαγραφή Λεξικού δεδομένων..... | 13 |
| 7.Δένδρο Αποσύνθεσης Διεργασιών..... | 14 |
| UML..... | 15 |
| 8.Διάγραμμα Περιπτώσεων Χρήσης..... | 15 |
| 9.Διάγραμμα Κλάσεων..... | 16 |
| 10.Διάγραμμα Καταστάσεων..... | 17 |
| 11.Κύριο Σενάριο Επιτυχίας και Εναλλακτικά Σενάρια..... | 18 |
| 12.Διάγραμμα Δραστηριοτήτων..... | 19 |
| 13.Λεπτομερές Διάγραμμα Κλάσεων..... | 20 |
| 14. Διάγραμμα Ακολουθίας..... | 21 |
| 15.Διάγραμμα Επικοινωνίας..... | 22 |

| | |
|--------------------------------------|----|
| ΔΟΜΗΜΕΝΟΣ ΣΧΕΔΙΑΣΜΟΣ..... | 23 |
| 16.Διάγραμμα Δομής Προγράμματος..... | 23 |
| 17.Ψευδοκώδικας..... | 24 |
| ΕΠΙΛΟΓΟΣ..... | 26 |
| Δομημένη ανάλυση..... | 26 |
| UML..... | 26 |
| Συνδυασμός μεθόδων..... | 27 |

ΕΙΣΑΓΩΓΗ

Σκοπός της εργασίας

Ο σκοπός της εργασίας είναι μέσω της γλώσσας σχεδιασμού UML και του μοντέλου δομημένης ανάλυσης να κατανοήσουμε τις προδιαγραφές απαιτήσεων καθώς και να κάνουμε την οπτική αναπαράσταση ενός συστήματος μέσω αυτών των 2 αντικειμενοστραφών συστημάτων.

Διαμοιρασμός εργασίας

Χωρίσαμε την εργασία σε 2 βασικά κομμάτια. Ο Κωνσταντίνος Πασχόπουλος πήρε το τμήμα των ζητούμενων της δομημένης ανάλυσης και ο Γιάννης Γιαννακίδης πήρε το τμήμα των ζητούμενων της UML. Πραγματοποιήσαμε μαζί τα ζητούμενα του δομημένου σχεδιασμού καθώς και το τελικό κείμενο.

Συνεργασία

Πραγματοποιήσαμε εβδομαδιαίες συζητήσεις στην σχολή κάθε Τρίτη όπου αναλύαμε την πορεία μας στην εργασία και χρησιμοποιήσαμε το Discord για την ανταλλαγή μηνυμάτων. Προκειμένου να έχουμε κάπου συγκεντρωτικά όλα τα αρχεία μας για την εργασία δημιουργήσαμε ένα private repository στο GitHub όπου ο καθένας έκανε τις υποβολές του εκεί.

Τελικό κείμενο-Υποβολή εργασίας

Το τελικό κείμενο το δημιουργήσαμε μαζί. Την τελική υποβολή της εργασίας την πραγματοποίησε ο Γιάννης Γιαννακίδης.

Επόμενα κεφάλαια

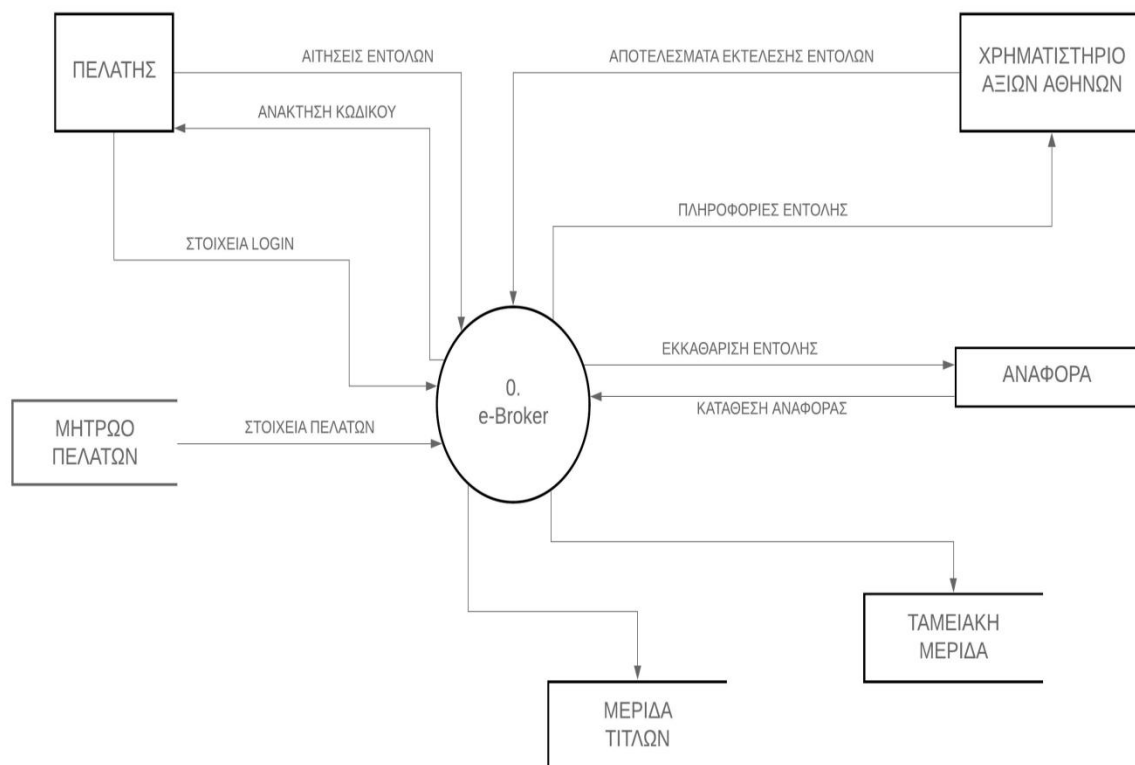
Τα επόμενα κεφάλαια περιλαμβάνουν ένα κεφάλαιο για κάθε ζητούμενο με μία σύντομη περιγραφή όπου χρειάζεται καθώς και τον επίλογο.

ΖΗΤΟΥΜΕΝΑ

ΔΟΜΗΜΕΝΗ ΑΝΑΛΥΣΗ

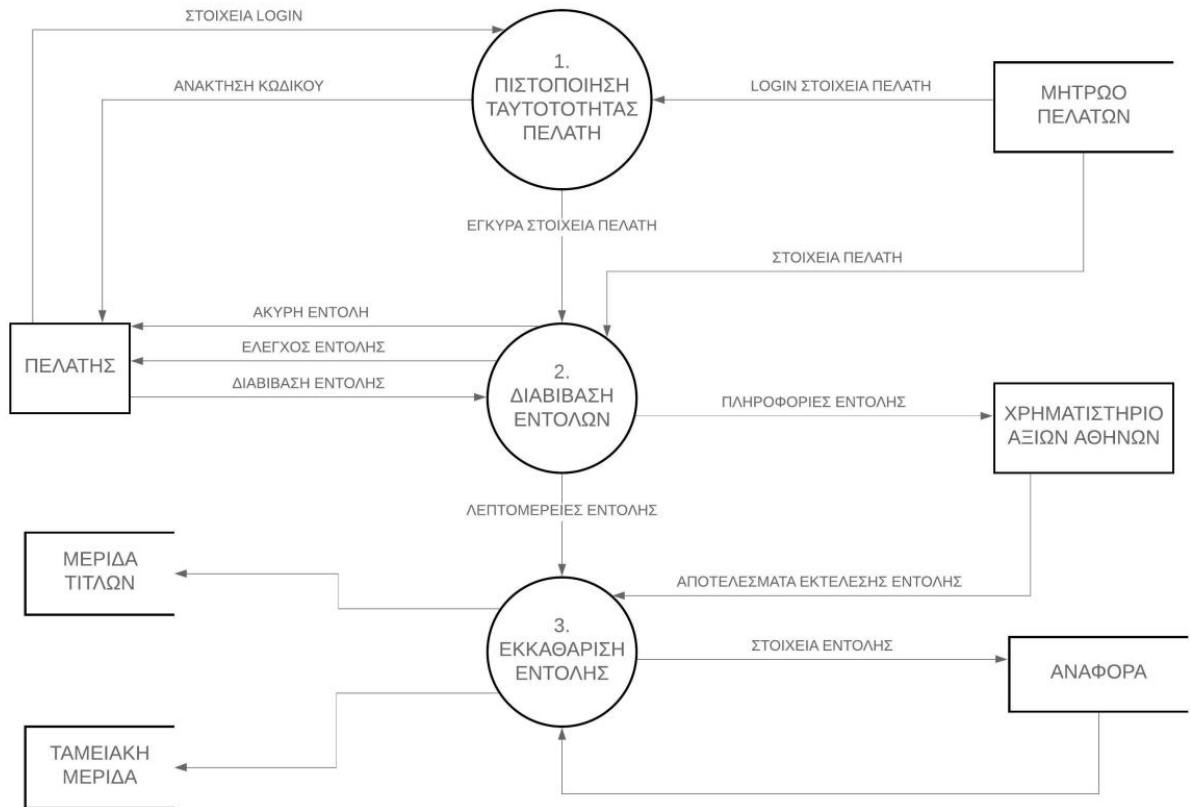
1.Γενικό Διάγραμμα Ροής Δεδομένων

Γενικό Διάγραμμα Ροής Δεδομένων
(επίπεδο 0)

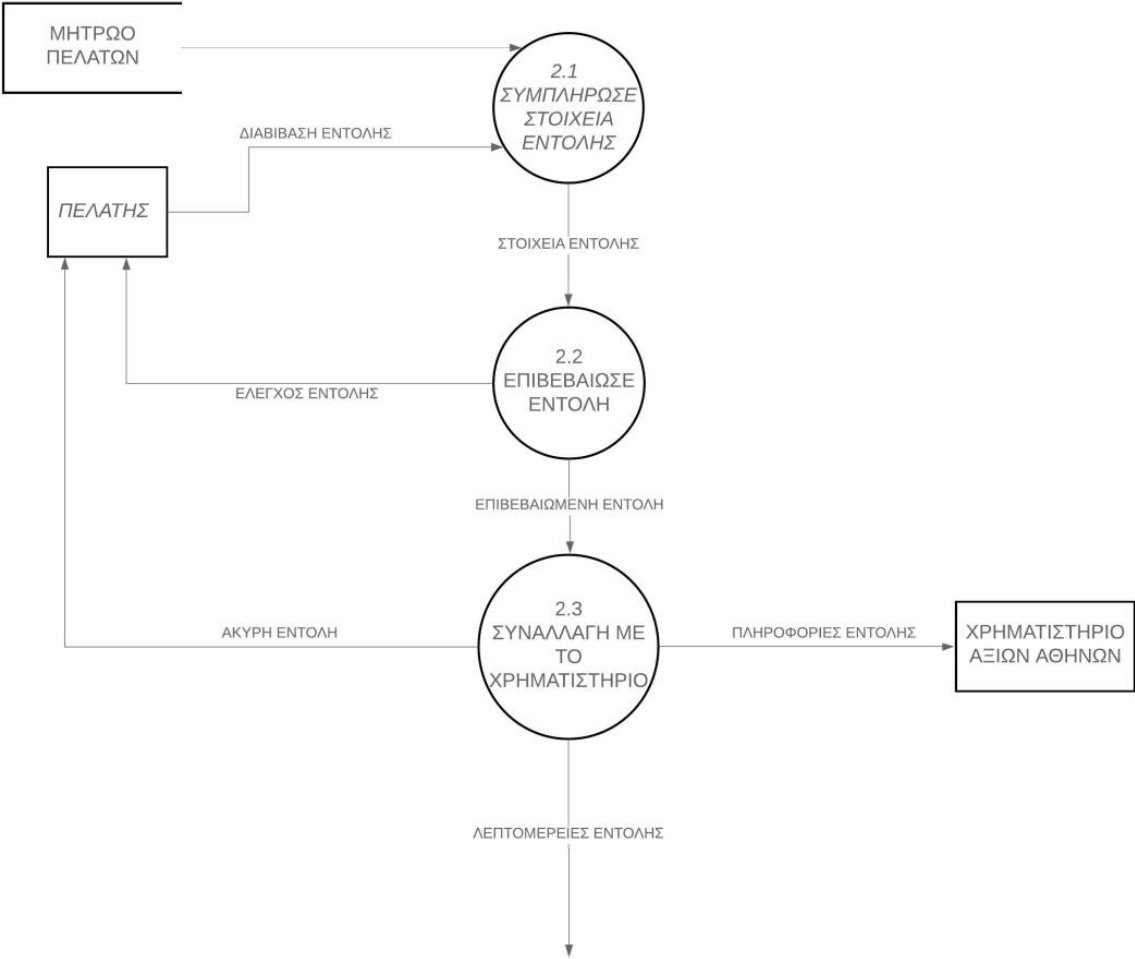


2. Διαγράμματα Ροής Δεδομένων

Διάγραμμα Ροής Δεδομένων
(επίπεδο 1)



Διάγραμμα Ροής Δεδομένων
(επίπεδο 2)



3.Προδιαγραφή διαδικασίας διαβίβασης εντολής σε Δομημένα Αγγλικά

Η Προδιαγραφή της διαδικασίας διαβίβασης εντολής σε Δομημένα Αγγλικά

```
RECEIVE order-details from interface
REVIEW order-details
IF client confirms order-details are correct
    SEND order to XAA
ELSE
    EDIT order-details
    SEND order to XAA
IF (XAA acknowledges order) AND (XAA accepts order)
    CONTINUE
ELSE
    SHOW error message to client
```

4.Προδιαγραφή διαδικασίας υπολογισμού της προμήθειας σε Πίνακα Απόφασης

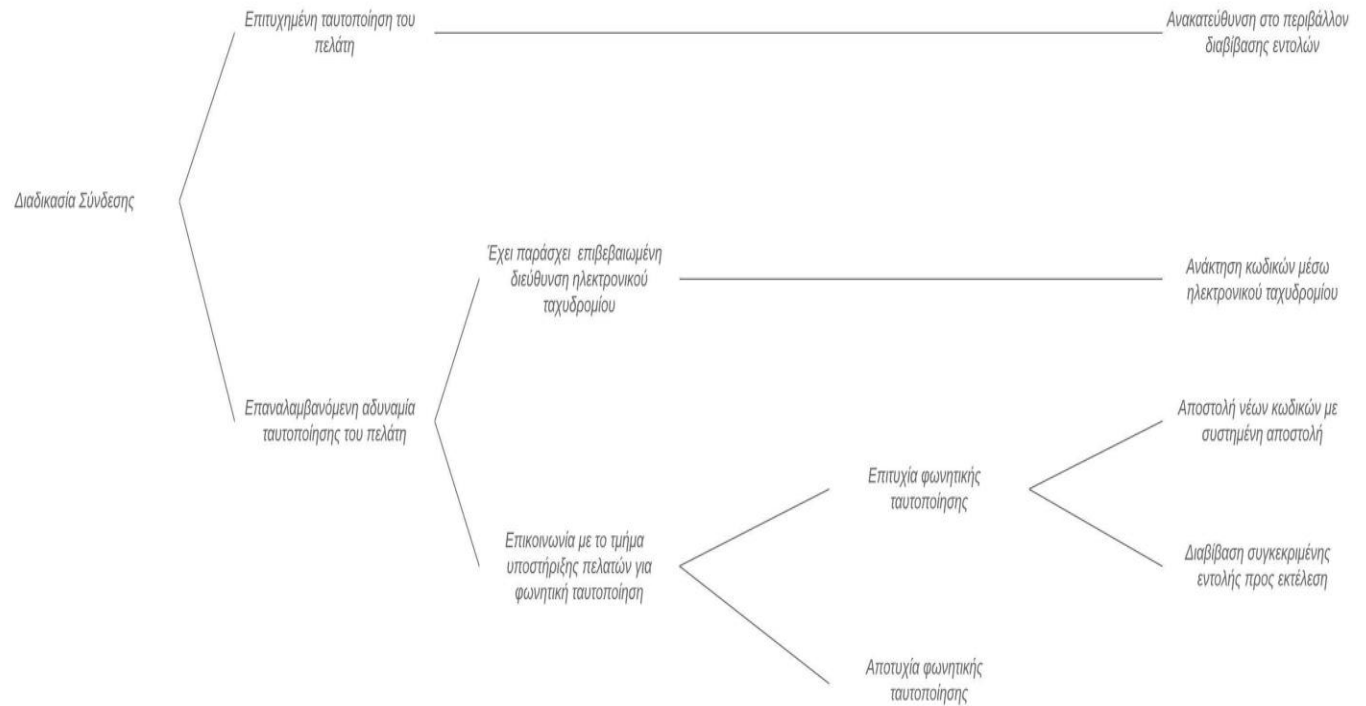
Η Προδιαγραφή της διαδικασίας υπολογισμού της προμήθειας σε Πίνακα Απόφασης Εκτεταμένων Καταχωρήσεων

| | 1 | 2 | 3 | 4 |
|----------------------------------|---|------|-------|-----|
| Εγκεκριμένη εντολή | N | Y | Y | Y |
| Συνολικός πραγματοποιηθείς όγκος | - | 0-Ο1 | Ο1-Ο2 | >Ο2 |
| Προμήθεια (%) | - | Π1 | Π2 | Π3 |

Όπου για την προμήθεια ισχύει: $\Pi 1 > \Pi 2 > \Pi 3$ και για τα παραμετρικά κατώφλια: $Ο 1 < Ο 2$.

5. Προδιαγραφή διαδικασίας σύνδεσης σε Δένδρο Απόφασης

Η Προδιαγραφή της διαδικασίας
σύνδεσης του πελάτη στο σύστημα
e-Broker σε Δένδρο Απόφασης

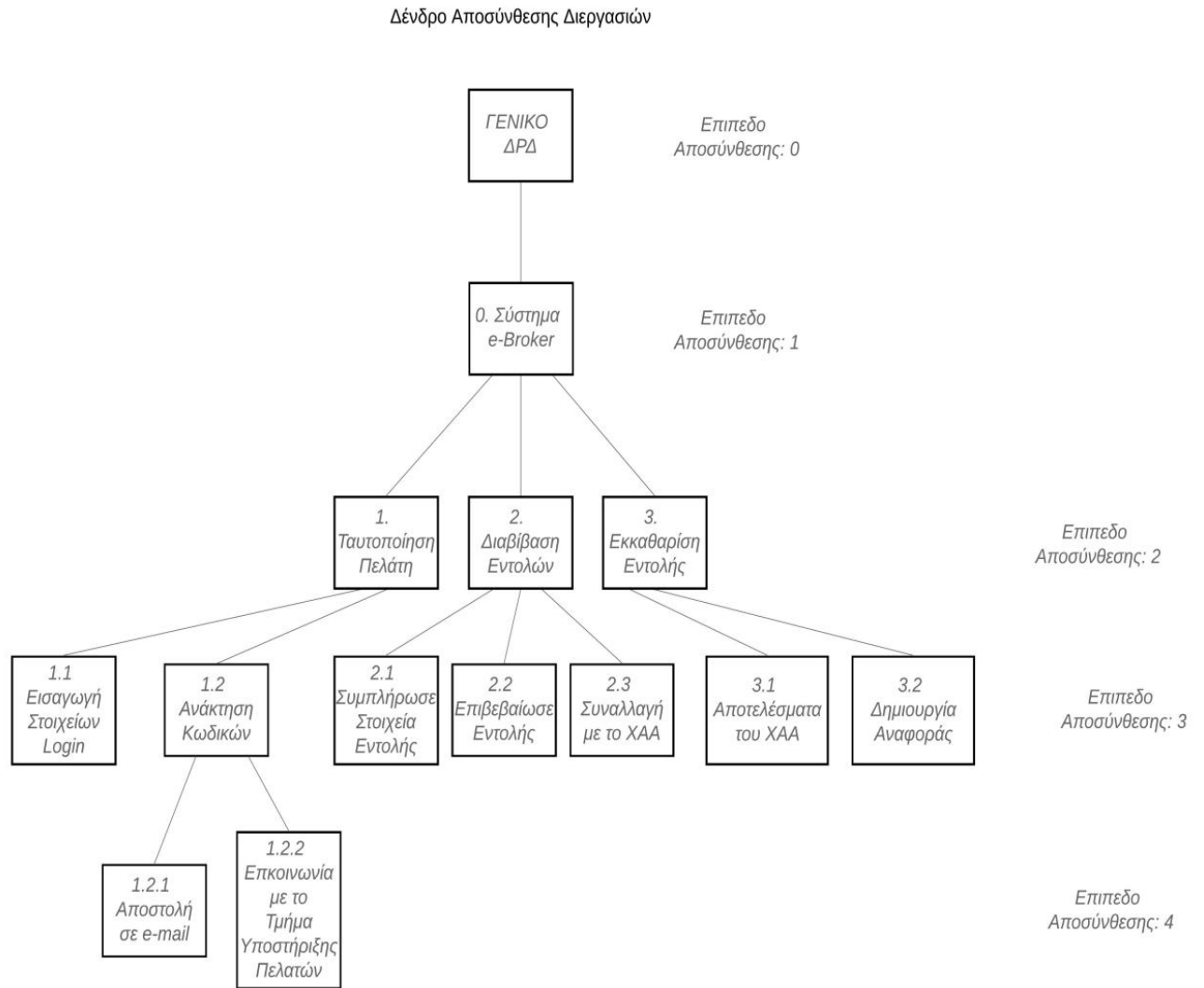


6.Προδιαγραφή Λεξικού δεδομένων

Η Προδιαγραφή Λεξικού Δεδομένων για την αναφορά εκκαθάρισης εντολής

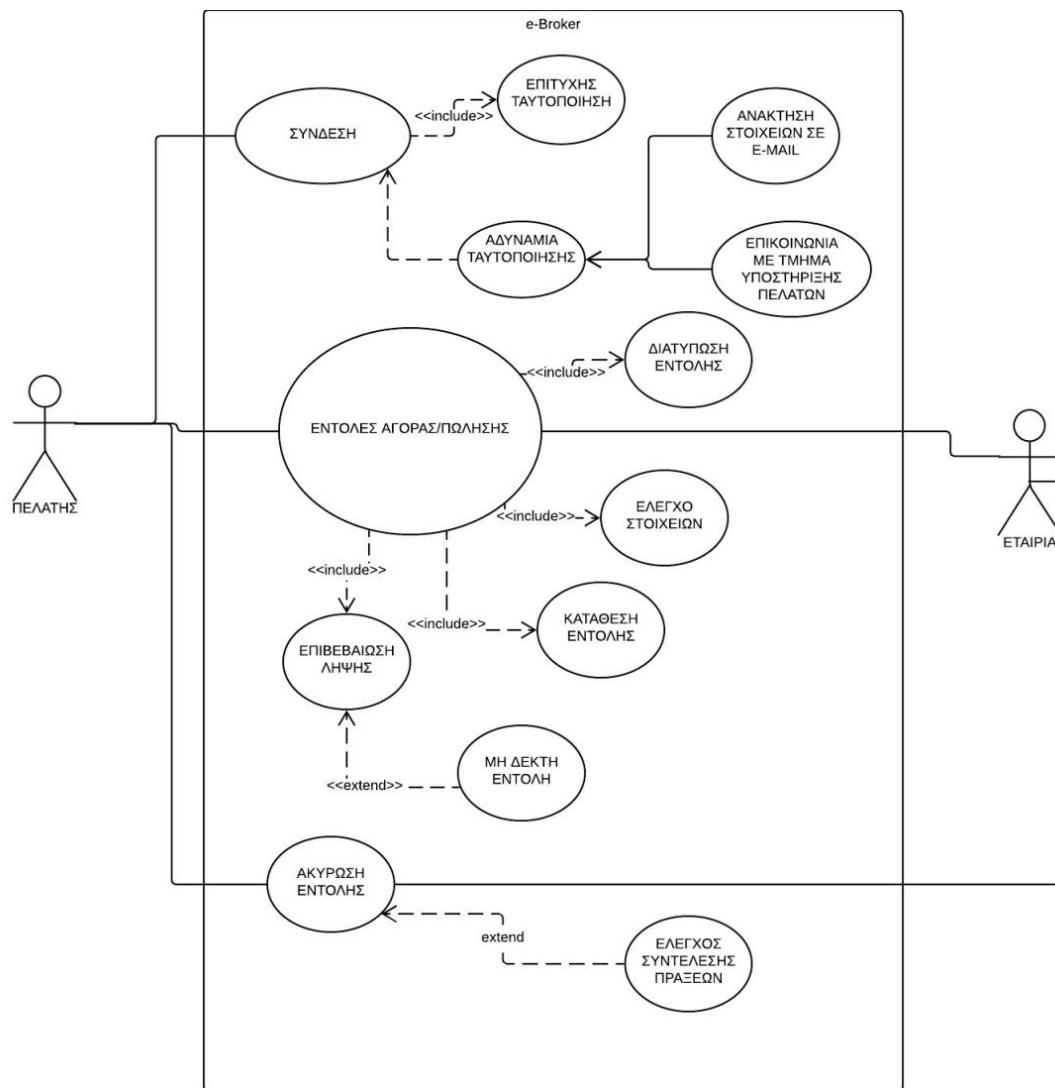
| Όνομα μεταβλητής | Τύπος μεταβλητής | Περιγραφή τύπου | Περιγραφή μεταβλητής | Παράδειγμα |
|------------------|------------------|--------------------------|--|---------------------|
| NAME | String | Μέχρι 50 χαρακτήρες | Επωνυμία της οικείας εταιρίας | Apple |
| TYPE | Boolean | 0 = αγορά, 1 = πώληση | Είδος πράξης (αγορά, πώληση) | 1 |
| PRICE | Float | | Τιμή μετοχής | 0.65 |
| TIME | String | YYYY-MM-DD:HH:MM:SS | Χρονική στιγμή εκτέλεσης της πράξης | 2019-03-23:23:56:01 |
| CODE | Integer | Μέχρι 6 ψηφία | Κωδικός του συγκεκριμένου τίτλου μετοχής | 123456 |
| QUANTITY | Integer | | Πλήθος μετοχών | 34 |
| VOLUME | Float | | Συνολικό τίμημα (όγκος συναλλαγής) | 66.5 |
| TOTAL_VOLUME | Float | | Γενικό σύνολο όγκου αγοράς ή πώλησης | 302 |
| TOTAL_QUANTITY | Integer | | Συνολικό πλήθος μετοχών | 2 |
| FEE | Float | | Προμήθεια της χρηματιστηριακής εταιρίας | 23.5 |

7. Δένδρο Αποσύνθεσης Διεργασιών

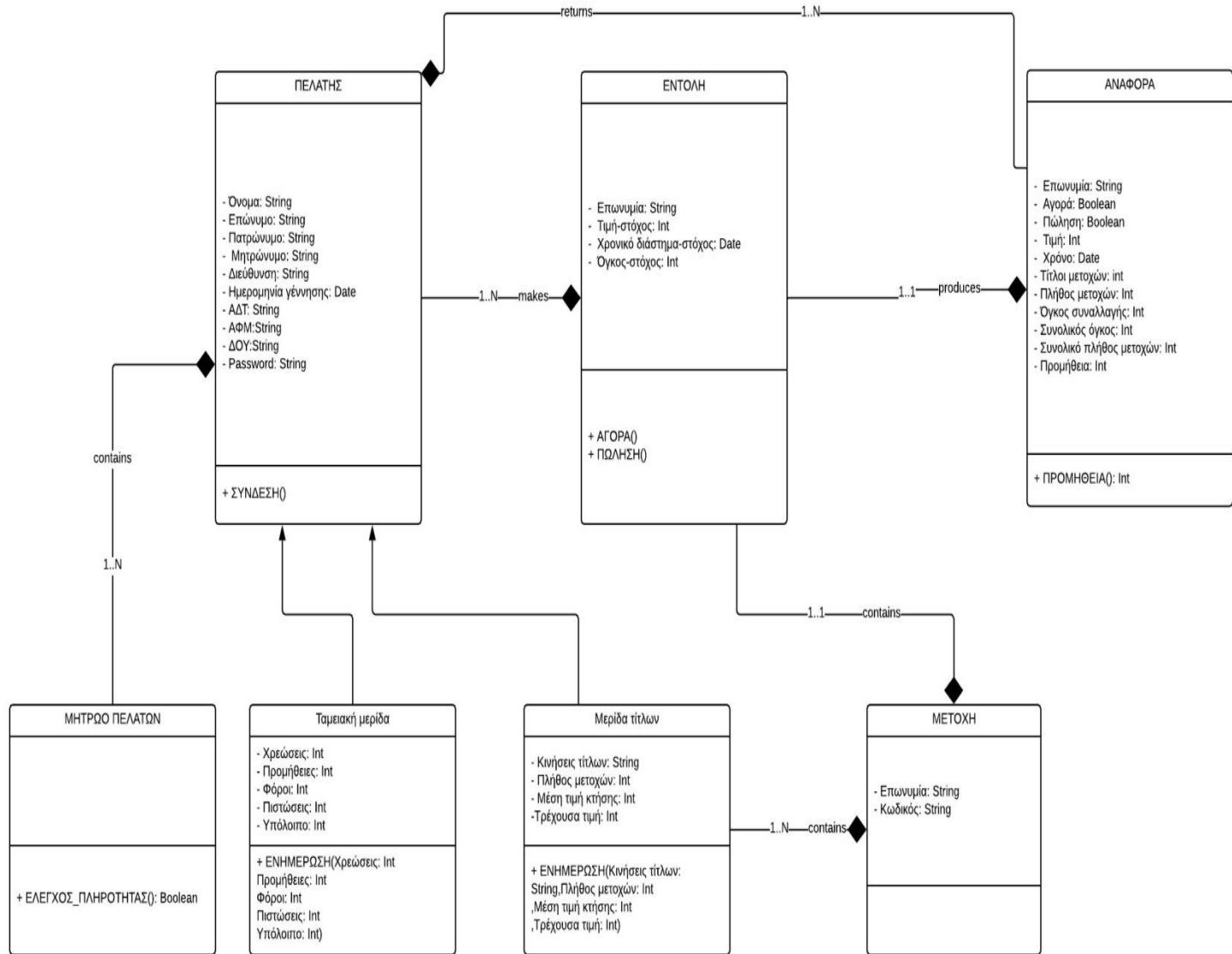


UML

8. Διάγραμμα Περιπτώσεων Χρήσης



9. Διάγραμμα Κλάσεων

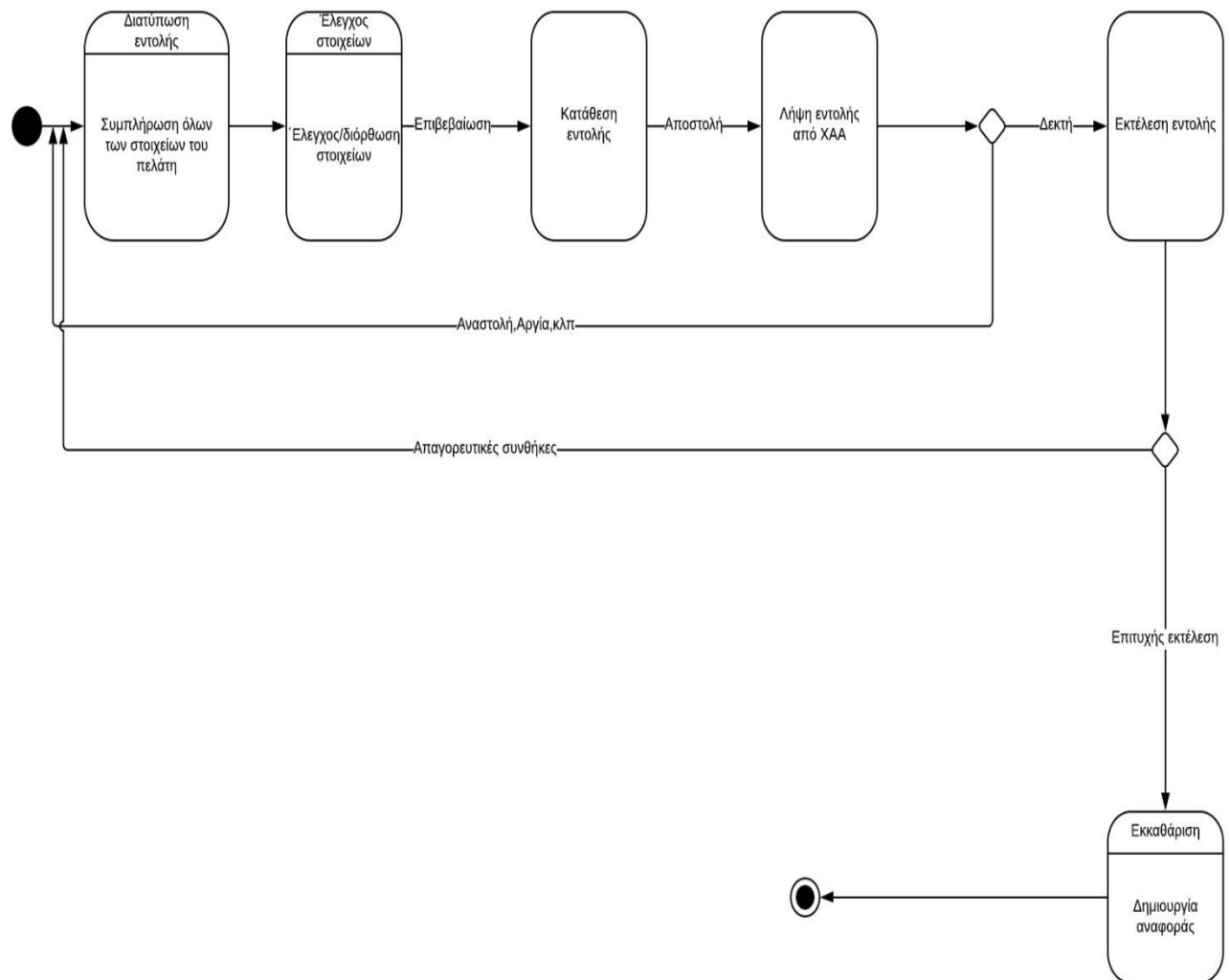


10.Διάγραμμα Καταστάσεων

bull_3

John Johnakidis | May 22, 2019

ΕΝΤΟΛΗ

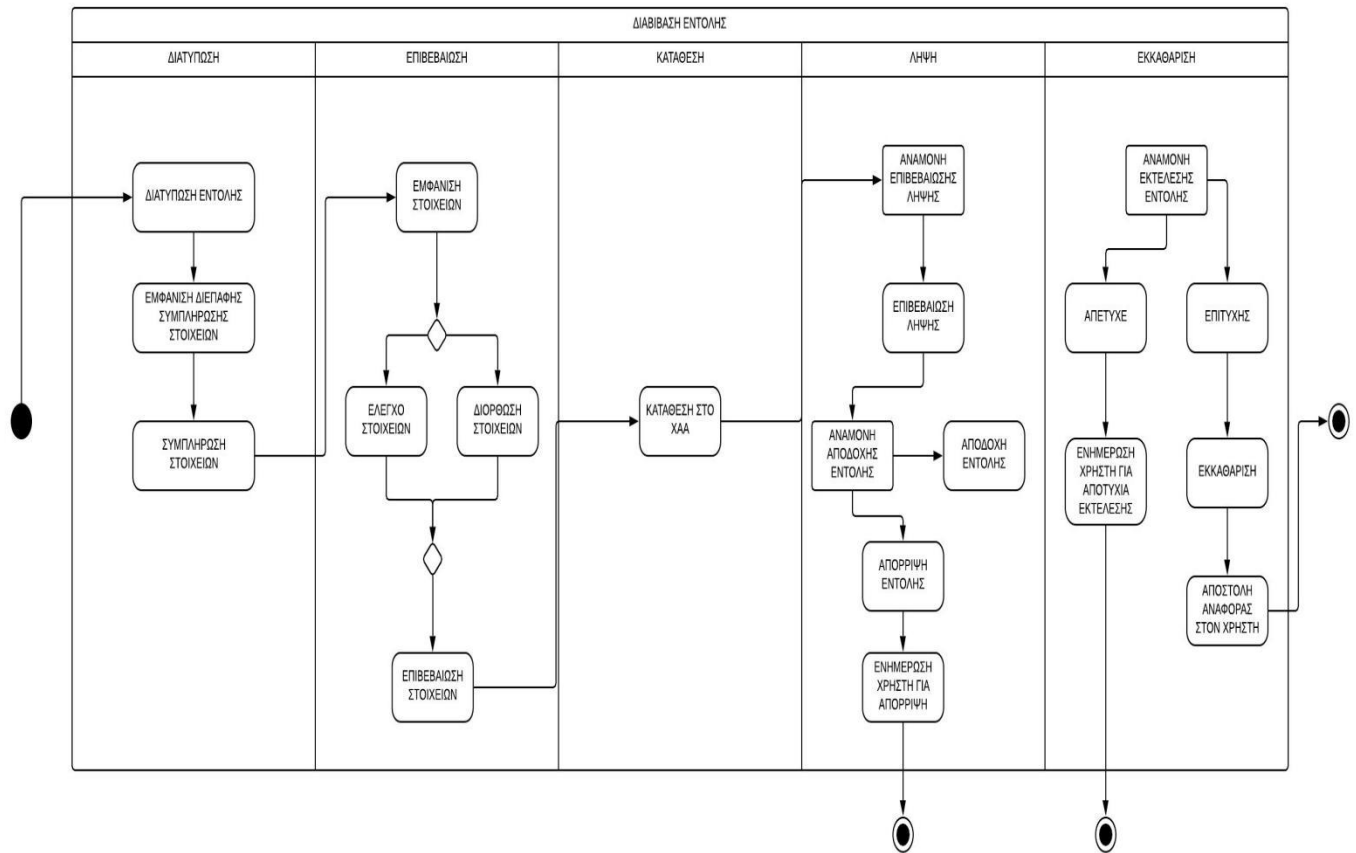


11.Κύριο Σενάριο Επιτυχίας και Εναλλακτικά Σενάρια

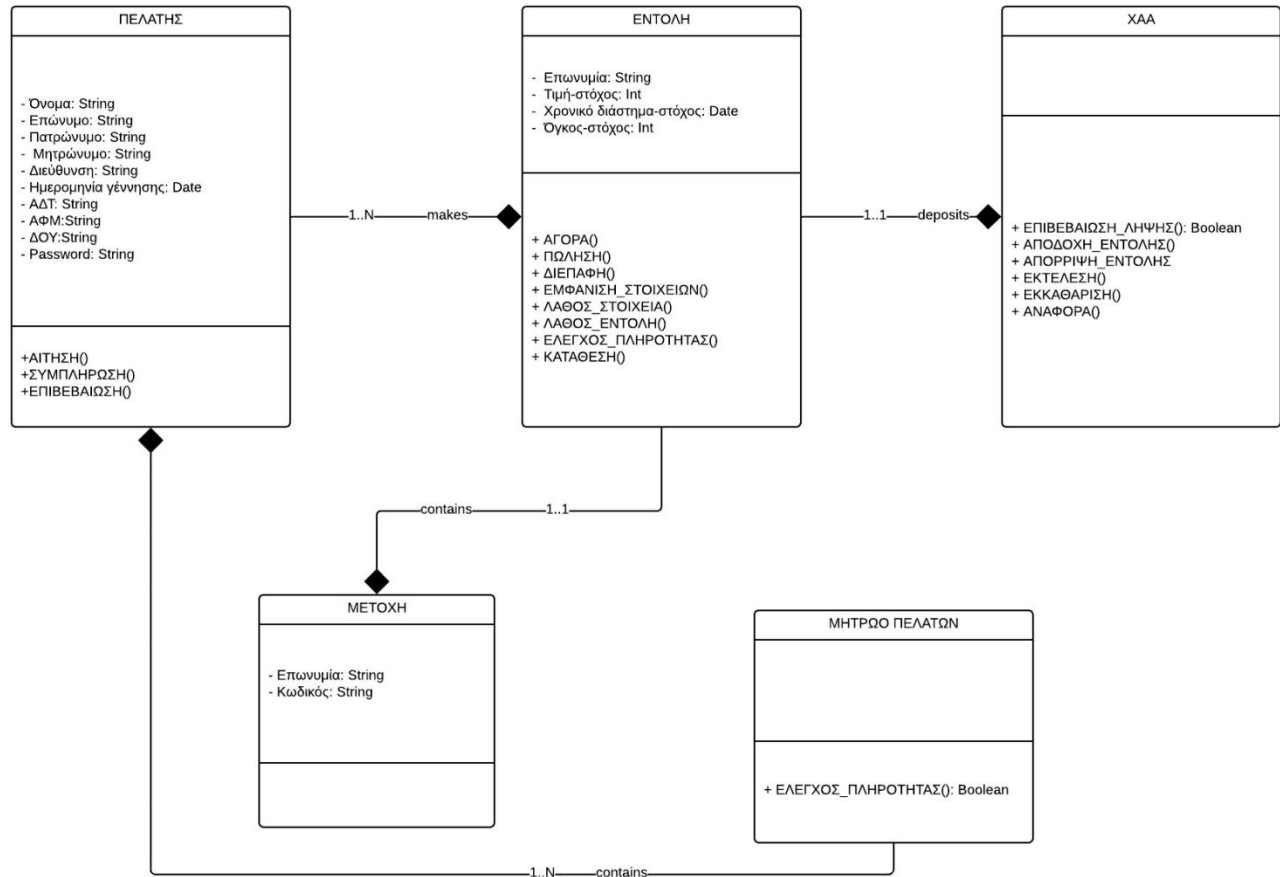
Κύριο σενάριο επιτυχίας και εναλλακτικά σενάρια:

- 1) Ο πελάτης θέλει να διατυπώσει μια εντολή αγοράς/πώλησης μετοχών.
- 2) Το σύστημα παρουσιάζει διεπαφή για συμπλήρωση των στοιχείων.
- 3) Ο πελάτης συμπληρώνει τα στοιχεία της εντολής.
- 4) Το σύστημα παρουσιάζει τα στοιχεία που συμπλήρωσε ο πελάτης.
- 5) Ο πελάτης ελέγχει τα στοιχεία του.
 1. Ο πελάτης κάνει κάποια αλλαγή επειδή εντόπισε λάθος στα στοιχεία.
- 6) Ο πελάτης επιβεβαιώνει τα στοιχεία του.
- 7) Το σύστημα καταθέτει την εντολή στο ΧΑΑ.
- 8) Το ΧΑΑ επιβεβαιώνει την λήψη της εντολής.
- 9) Το ΧΑΑ αποδέχεται την εντολή.
 1. Το ΧΑΑ απορρίπτει την εντολή για διάφορους λόγους
 2. Ο πελάτης ενημερώνεται με σχετικό μήνυμα και επιστρέφουμε στο βήμα 1
- 9) Το ΧΑΑ εκτελεί την εντολή.
 1. Η εντολή αποτυγχάνει λόγω απαγορευτικών συνθηκών.
 2. Ο πελάτης ενημερώνεται με σχετικό μήνυμα και επιστρέφουμε στο βήμα 1
- 10) Εκκαθάριση της εντολής-δημιουργία αναφοράς από ΧΑΑ.
- 11) Μεταβίβαση αναφοράς στον πελάτη και ενημέρωση.

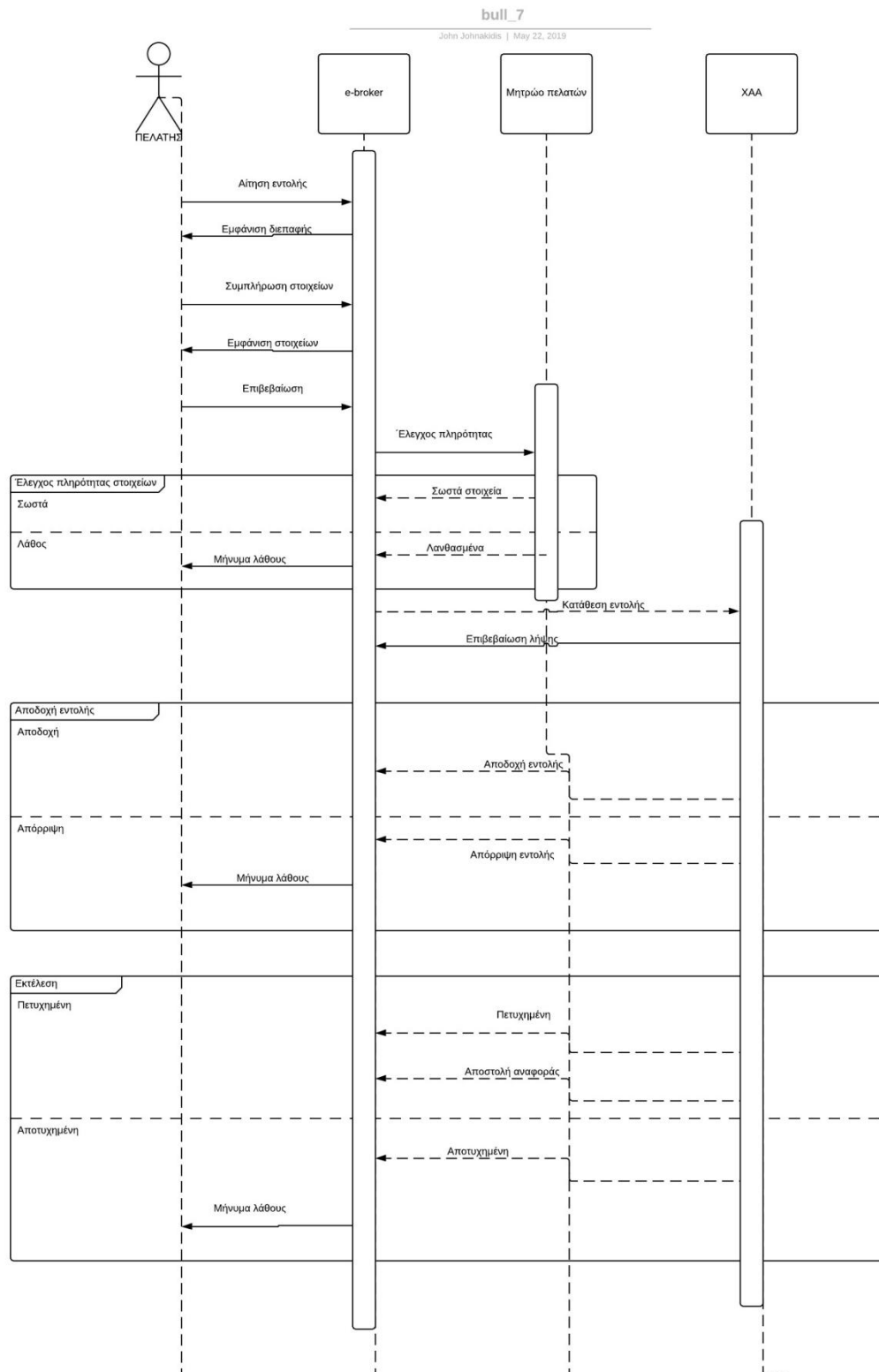
12.Διάγραμμα Δραστηριοτήτων



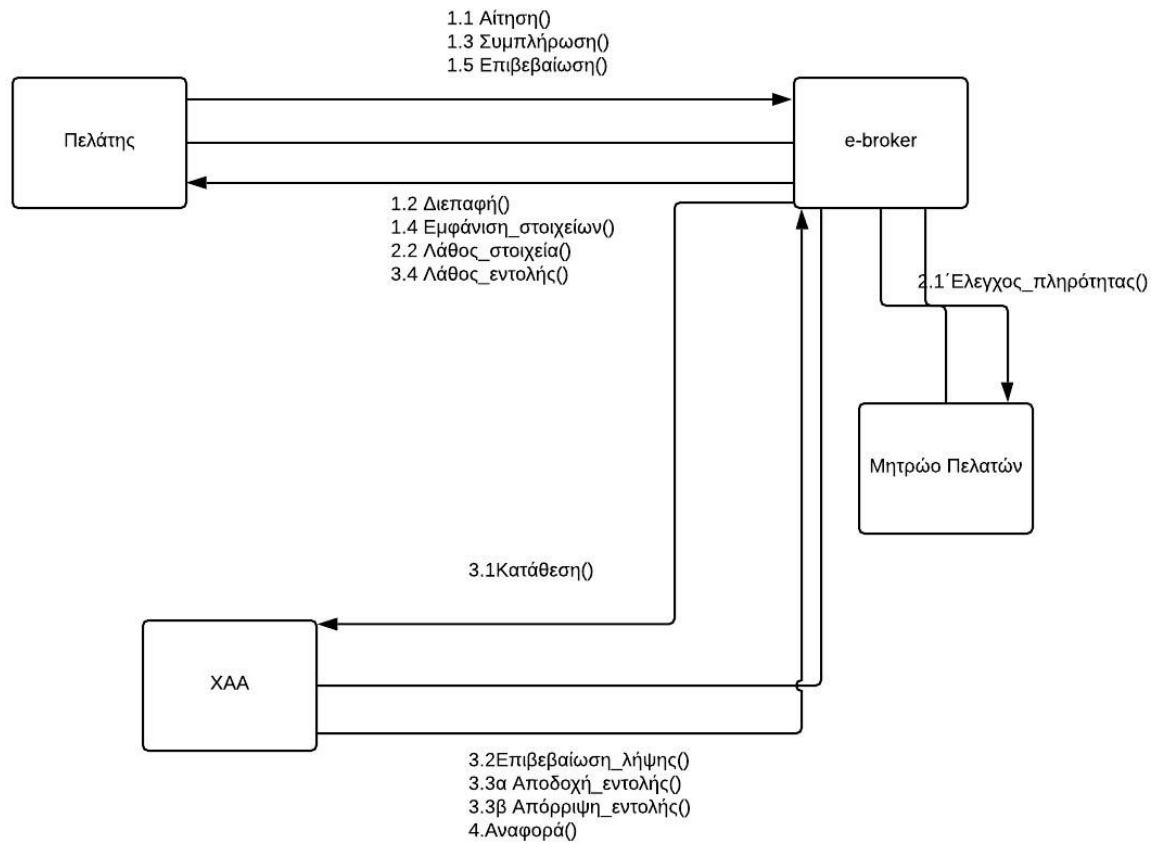
13.Λεπτομερές Διάγραμμα Κλάσεων



14. Διάγραμμα Ακολουθίας

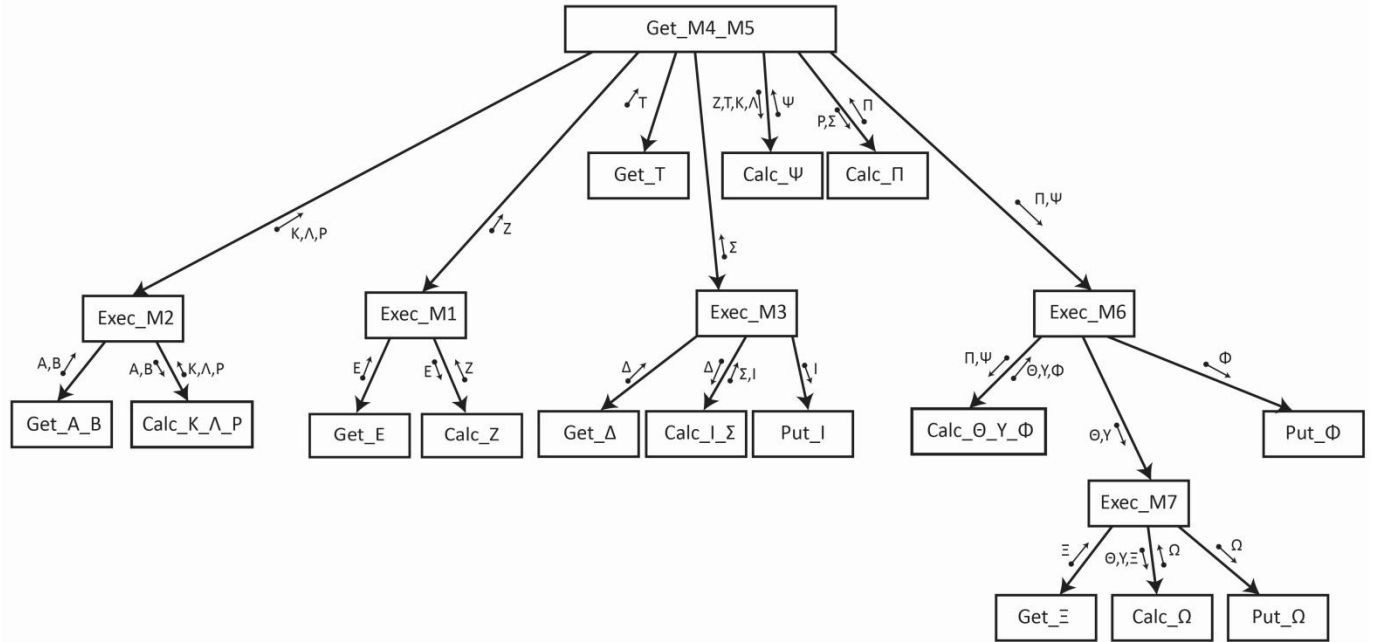


15.Διάγραμμα Επικοινωνίας



ΔΟΜΗΜΕΝΟΣ ΣΧΕΔΙΑΣΜΟΣ

16. Διάγραμμα Δομής Προγράμματος



17.Ψευδοκώδικας

(α)

Procedure ExecM4M5

Local Var T, Z, Σ, Κ, Λ, Ρ, Ψ, Π

Αρχικοποίηση T, Z, Σ, Κ, Λ, Ρ, Ψ, Π

Call GetT(T)

Call ExecM1(Z)

Call ExecM3(Σ)

Call ExecM2(Κ, Λ, Ρ)

Call CalcM4(Z, T, Κ, Λ, Ψ)

Call CalcM5(Ρ, Σ, Π)

Call ExecM6(Ψ, Π)

End_Procedure

(β)

Procedure ExecM3

Local Var Δ, Σ, Ι

Αρχικοποίηση Δ, Σ, Ι

Call GetΔ(Δ)

Call CalcM3(Δ, Σ, Ι)

Call ExecM5(Σ)

Call PutI(I)

End_Procedure

(γ)

Procedure CalcM4 (Z, T, K, Λ: IN, Ψ: IN/OUT)

Υπολόγισε το δεδομένο Ψ από τα δεδομένα T, Z, K και Λ

End_Procedure

(δ)

Procedure GetAB (A, B: IN/OUT)

Διάβασε το A και το B από την ΠΗΓΗ ΔΕΔΟΜΕΝΩΝ 2

End_Procedure

ΕΠΙΛΟΓΟΣ

Δομημένη ανάλυση

Αναλύοντας την εργασία με αυτόν τον τρόπο καταφέραμε να έχουμε μια πλήρη εικόνα των απαιτήσεων της εργασίας πριν αρχίσουμε να γράφουμε τον κώδικα. Πολλές φορές όταν ξεκινάμε να γράφουμε κώδικα δεν είμαστε σίγουροι για το πως θα εξελιχθεί με αποτέλεσμα να χρειάζεται να πηγαίνουμε πίσω σε αυτά που έχουμε γράψει και να κάνουμε αλλαγές. Αυτό έχει ως αποτέλεσμα να χάνουμε χρόνο και πολλές φορές να γράφουμε ασύνδετα κομμάτια τα οποία δημιουργούν προβλήματα στην συνέχεια. Το δένδρο αποσύνθεσης διεργασιών και τα Διαγράμματα ροής δεδομένων βοηθάνε στο να χωρίζουμε τα απαιτούμενα σε κομμάτια με αποτέλεσμα την καλύτερη επίλυση τους. Η μεγαλύτερη δυσκολία είναι ότι πρέπει να έχεις κατανοήσει πολύ καλά όλα τα επιμέρους κομμάτια, το οποίο μπορεί να είναι αρκετά χρονοβόρο.

UML

Βλέπουμε πως με την χρήση της UML υπάρχει μία καλύτερη οργάνωση στον καθορισμό των απαιτήσεων και προδιαγραφών ενός συστήματος και πως λαμβάνεται υπόψιν η αλληλεπίδραση μεταξύ των διαφορετικών παραγόντων που χρησιμοποιούν το σύστημα. Επίσης αποτελεί έναν πιο επίσημο σχεδιασμό πλάνων και είναι πιο χρηστικός τρόπος από το να γίνει ένα τέτοιο σχέδιο “στο χέρι”.

Κάποιος που δεν έχει εκπονήσει άκομα αυτό το κομμάτι θα του πρότεινα να διαβάσει αρκετά τις σημειώσεις του μαθήματος καθώς και να ερευνήσει το διαδίκτυο διότι υπάρχει αρκετό βοηθητικό υλικό. Επίσης, όλα τα προγράμματα για UML (StarUML, lucidchart, κλπ) είναι αρκετά απλά και εύκολα στην χρήση. Τέλος, θα πρότεινα να δώσει ιδιαίτερη προσοχή στην εκφώνηση και να κατανοήσει πλήρως την λειτουργία του συστήματος καθώς και να κάνει κάποια αρχικά πλάνα σε ένα χαρτί για να έχει μία ιδέα του τελικού διαγράμματος που θα υλοποιήσει.

Συνδυασμός μεθόδων

Στην δομημένη ανάλυση το πιο σημαντικό εργαλείο είναι το διάγραμμα ροής δεδομένων και είναι μια γλώσσα που χρησιμοποιείται για την μοντελοποίηση διεργασιών σε δομημένα περιβάλλοντα. Από την άλλη πλευρά η UML είναι μια συλλογή διαγραμμάτων που χρησιμοποιείται για την μοντελοποίηση διαφορετικών όψεων ενός προγράμματος.

Συνδυάζοντας τα αποτελέσματα της δομημένης ανάλυσης και της UML μπορούμε να κατασκευάσουμε ένα πλήρες μοντέλο για τις ανάγκες ενός συστήματος. Και οι δύο μέθοδοι είναι πολύ φιλικές προς τον χρήστη και για αυτόν τον λόγο μπορούμε να πάρουμε τα πλεονεκτήματα και των 2 και να εκφράσουμε τα σημαντικά χαρακτηριστικά ενός συστήματος. Ακόμη, μετά από τον συνδυασμό αυτών των δύο μεθόδων έχουμε διασπάσει το σύστημα σε αρκετά επιμέρους κομμάτια και έτσι μπορούμε να δώσουμε την απαραίτητη προσοχή στο καθένα και να είναι ταυτόχρονα ανεξάρτητα έτσι ώστε να μην αλληλοκαλύπτονται. Τέλος, υπάρχει η δυνατότητα της παραλληλοποίησης των διαφορετικών συνιστωσών του συστήματος και η καλύτερη τμηματοποίηση των εργασιών ειδικά για τους προγραμματιστές.

Δεν ξεχωρίσαμε κάποια από τις 2 μεθόδους ως πιο σημαντική, καθώς θεωρούμε πως και οι 2 έχουν σημαντική θέση στην ανάλυση των απαιτήσεων του συστήματος και η κάθε μια με διαφορετικό τρόπο.