
ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΈΤΟΣ 2019-2020

ΟΜΑΔΑ 4336-4443-4444

ΜΥΛΩΝΑΣ ΔΙΟΝΥΣΗΣ, ΑΜ:4443

ΔΗΜΗΤΡΗΣ ΓΙΑΝΝΑΚΟΠΟΥΛΟΣ,

ΑΜ:4336

ΜΥΡΙΤΖΗΣ ΕΥΣΤΡΑΤΙΟΣ, ΑΜ:4444

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2019

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

| Ημερομηνία | Έκδοση | Περιγραφή | Συγγραφείς |
|------------|--------|---|------------------------|
| 2019/11/3 | v.01 | Οργάνωση απαιτήσεων σε use cases | 4336-4443-4444 |
| yyyy/mm/dd | v.02 | Αρχική σχεδίαση κλάσεων και ελέγχων | XX,YY,ZZ |
| yyyy/mm/dd | ... | Διορθώσεις στις uses cases, επεκτάσεις στη σχεδίαση κλάσεων και ελέγχων | YY,ZZ (ο XX αποχώρησε) |
| yyyy/mm/dd | ... | ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ | YY |

1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USE CASES

Στην παρούσα ενότητα, παρατίθενται οι περιγραφές των use cases με βάση τις καταγεγραμμένες απαιτήσεις.

UC1:ΔΗΜΙΟΥΡΓΙΑ ΙΣΤΟΡΙΚΟΥ

DESCRIPTION AND GOAL

Η UC1 έχει σκοπό τη δημιουργία ή πρόσβαση αρχείου Ιστορικού, όπου θα αποθηκεύονται τα πεδία που δίνονται από τον χρήστη.

ACTORS (ESP. PRIMARY ACTOR)

- 1)Χρήστης(Αναλυτής)
- 2)Tests

PRECONDITIONS

Πρέπει να έχει αρχίσει επιτυχώς(launch) το πρόγραμμά μας.

BASIC FLOW

- 1)Η UC ξεκινάει όταν ο χρήστης ξεκινήσει το πρόγραμμα.
- 2)Το σύστημα δημιουργεί ένα αρχείο, στο οποίο θα αποθηκεύονται τα στοιχεία που ζητήθηκαν από τον αναλυτή.

EXTENSIONS / VARIATIONS

1)Αν υπάρχει ήδη το αρχείο ιστορικού το βήμα 2 του basic flow αγνοείται και απλά αποκτά πρόσβαση σε αυτό το αρχείο.

POST CONDITIONS

Πρόσβαση σε αρχείο Ιστορικού.

SPECIAL REQUIREMENTS , ISSUES , RISKS AND OTHER COMMENTS

-

UC2: ΕΠΙΛΟΓΗ ΛΕΙΤΟΥΡΓΙΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

DESCRIPTION AND GOAL

Η UC2 έχει σκοπό την εμφάνιση των λειτουργιών του προγράμματος και την έγκυρη επιλογή του χρήστη.

ACTORS (ESP. PRIMARY ACTOR)

- 1)Χρήστης(Αναλυτής)
- 2)Tests

PRECONDITIONS

Πρέπει να έχει πρόσβαση σε αρχείο ιστορικού.

BASIC FLOW

- 1)Η UC ξεκινάει όταν έχει γίνει επιτυχής πρόσβαση σε αρχείο ιστορικού.
- 2)Το σύστημα παρουσιάζει στον χρήστη τις διάφορες πιθανές λειτουργίες του προγράμματος.
 - 2.1)Η επιλογή 1 επιτρέπει στο χρήστη να επεξεργαστεί τα δεδομένα κατανάλωσης
 - 2.2) Η επιλογή 2 επιτρέπει στο χρήστη να διαβάσει το αρχείο ιστορικού.
 - 2.3) Η επιλογή 3 επιτρέπει στο χρήστη να βγει από το πρόγραμμα.
 - 2.3.1)Αν ο χρήστης επιλέξει την επιλογή 3 το πρόγραμμα τερματίζει
- 3)Ο χρήστης επιλέγει 1 από τις 3 επιλογές .

EXTENSIONS / VARIATIONS

1) Στην περίπτωση που ο χρήστης επιλέξει μη υπαρκτή λειτουργία γίνεται επανάληψη της ερώτησης.

POST CONDITIONS

-

SPECIAL REQUIREMENTS , ISSUES , RISKS AND OTHER COMMENTS

-

UC3: ΦΟΡΤΩΣΗ ΑΡΧΕΙΟΥ ΔΕΔΟΜΕΝΩΝ

DESCRIPTION AND GOAL

Η UC3 έχει σκοπό τη σωστή παροχή του αρχείου δεδομένων στο σύστημα.

ACTORS (ESP . PRIMARY ACTOR)

- 1)Χρήστης(Αναλυτής)
- 2)Tests

PRECONDITIONS

Πρέπει να έχει υπάρξει "έγκυρο" αρχείο δεδομένων και στην UC2 ο χρήστης να επέλεξε την επιλογή 1(επεξεργασία δεδομένων κατανάλωσης).

BASIC FLOW

- 1)Η UC3 ξεκινάει όταν ο χρήστης επιλέξει την επιλογή 1(επεξεργασία δεδομένων κατανάλωσης) στην UC2.
- 2)Το σύστημα ζητάει από τον χρήστη ένα αρχείο δεδομένων.
- 3) Ο χρήστης παρέχει στο σύστημα το ανάλογο path.
 - 3.1) Ο χρήστης επιλέγει το επιθυμητό αρχείο.
- 4) Φόρτωση αρχείου δεδομένων.

EXTENSIONS / VARIATIONS

- 1) Στην περίπτωση που ο χρήστης επιλέξει μη υπαρκτό/ακατάλληλο path, γίνεται επανάληψη της ερώτησης παροχής path.
- 2) Στην περίπτωση που ο χρήστης επιλέξει αρχείο ακατάλληλης μορφής, γίνεται επανάληψη της ερώτησης και έπειτα, της παροχής path.
- 3) Στην περίπτωση που έχει ήδη φορτωθεί το αρχείο δεδομένων δεν επαναλαμβάνεται η UC3.

POST CONDITIONS

-

SPECIAL REQUIREMENTS , ISSUES , RISKS AND OTHER COMMENTS

-

UC4: ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΕΠΙΘΥΜΗΤΩΝ ΣΤΑΤΙΣΤΙΚΩΝ

DESCRIPTION AND GOAL

Η UC4 έχει ως σκοπό την σωστή εισαγωγή των στατιστικών που ο χρήστης επιθυμεί να εμφανιστούν στο Report .

ACTORS (ESP . PRIMARY ACTOR)

- 1) Χρήστης(Αναλυτής)
- 2) Tests

PRECONDITIONS

Η ύπαρξη και η σωστή εισαγωγή του αρχείου δεδομένων στο σύστημα.

BASIC FLOW

- 1) Η UC ξεκινάει όταν γίνει η εισαγωγή του αρχείου δεδομένων στο σύστημα και όταν αυτό φορτώσει.
- 2) Το σύστημα ζητάει από τον χρήστη τις πληροφορίες που αυτός επιθυμεί να μάθει.
- 3) Ο χρήστης πληκτρολογεί τις επιθυμητές πληροφορίες.
- 4) Το σύστημα ζητάει από τον χρήστη ένα μικρό κείμενο.
- 5) Ο χρήστης πληκτρολογεί το μικρό κείμενο.
- 6) Το σύστημα αποθηκεύει τα πεδία που του δόθηκαν από τον χρήστη (αν είναι έγκυρα) στο αρχείο ιστορικού.

EXTENSIONS / VARIATIONS

- 1) Στην περίπτωση που ο χρήστης πληκτρολογήσει ακατάλληλα πεδία ως επιθυμητές πληροφορίες, γίνεται επανάληψη της ερώτησης.

POST CONDITIONS

- 1) Το μικρό κείμενο του χρήστη εμφανίζεται στο Report.
- 2) Αποθήκευση των πεδίων που έδωσε ο χρήστης στο αρχείο ιστορικού.

SPECIAL R EQUIREMENTS , ISSUES , R ISKS AND OTHER C OMMENTS

-

UC5: ΠΑΡΑΓΩΓΗ ΠΛΗΡΟΦΟΡΙΩΝ

DESCRIPTION AND GOAL

Η UC5 έχει σκοπό την εμφάνιση των επιθυμητών, από τον χρήστη, πληροφοριών στην επιθυμητή από τον χρήστη μορφή (html κλπ.)

ACTORS (ESP. PRIMARY ACTOR)

- 1) Χρήστης (Αναλυτής)
- 2) Tests

PRECONDITIONS

Η ύπαρξη του αρχείου δεδομένων, η σωστή εισαγωγή του στο σύστημα και η σωστή εισαγωγή των επιθυμητών στατιστικών που θέλει να μάθει ο χρήστης.

BASIC FLOW

- 1) Η UC5 ξεκινάει όταν έχουν εισαχθεί επιτυχώς οι επιθυμητές πληροφορίες από τον χρήστη.
- 2) Το σύστημα ζητάει από τον χρήστη το path στο οποίο θέλει να αποθηκεύσει το Report.
- 3) Ο χρήστης επιλέγει το επιθυμητό path.
- 4) Το σύστημα ζητάει από τον χρήστη την μορφή που επιθυμεί να έχει το αρχείο του Report.
- 5) Ο χρήστης επιλέγει την επιθυμητή μορφή.
- 6) Το σύστημα δημιουργεί το Report στην συγκεκριμένη μορφή και στο συγκεκριμένο path.
- 7) Το σύστημα εμφανίζει τα επιθυμητά στατιστικά στοιχεία και το μικρό κείμενο (από την UC4) του χρήστη στο Report.
- 8) Επιστροφή του προγράμματος στην UC2.

EXTENSIONS / VARIATIONS

- 1) Στην περίπτωση που ο χρήστης δεν επιλέξει υπαρκτό, στον υπολογιστή, path, η ερώτηση επαναλαμβάνεται.
- 2) Αν ο χρήστης πληκτρολογήσει μια μορφή για το Report την οποία δεν υποστηρίζει το πρόγραμμα, η ερώτηση επαναλαμβάνεται.

POST CONDITIONS

- 1) Δημιουργία του Report, με όλα τα επιθυμητά στοιχεία.

SPECIAL REQUIREMENTS , ISSUES , RISKS AND OTHER COMMENTS

-

UC6: ΕΜΦΑΝΙΣΗ ΙΣΤΟΡΙΚΟΥ

DESCRIPTION AND GOAL

Η UC6 έχει σκοπό την εμφάνιση την εμφάνιση του αρχείου ιστορικού.

ACTORS (ESP . PRIMARY ACTOR)

- 1)Χρήστης(Αναλυτής)
- 2)Tests

PRECONDITIONS

Πρέπει να έχει υπάρξει "έγκυρο" αρχείο ιστορικού και στην UC2 ο χρήστης να επέλεξε την επιλογή 2(ανάγνωση αρχείου ιστορικού).

BASIC FLOW

- 1)Η UC5 ξεκινάει όταν στην UC2 ο χρήστης να επέλεξε την επιλογή 2(ανάγνωση αρχείου ιστορικού).
- 2)Η επιλογή του χρήστη καταγράφεται στο αρχείο ιστορικού.
- 3)Το πρόγραμμα εμφανίζει στον χρήστη τα περιεχόμενα του αρχείου ιστορικού
- 4)Επιστροφή του προγράμματος στην UC2.

EXTENSIONS / VARIATIONS

POST CONDITIONS

-

SPECIAL REQUIREMENTS , ISSUES , RISKS AND OTHER COMMENTS

-

Οι έλεγχοι που σχεδιάστηκαν και εντάχθηκαν στην υλοποίηση περιγράφονται παρακάτω. Εδώ, ως υπόδειγμα: το project με την διάσπαση χρονοσειράς σε φάσεις.

2.1 ΕΛΕΓΧΟΣ USE CASES VIA SYSTEM TESTS

UC1 ΔΗΜΙΟΥΡΓΙΑ ΙΣΤΟΡΙΚΟΥ
UC2 ΕΠΙΛΟΓΗ ΛΕΙΤΟΥΡΓΙΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ
UC3 ΦΟΡΤΩΣΗ ΑΡΧΕΙΟΥ ΔΕΔΟΜΕΝΩΝ
UC4 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΕΠΙΘΥΜΗΤΩΝ ΣΤΑΤΙΣΤΙΚΩΝ
UC5 ΠΑΡΑΓΩΓΗ ΠΛΗΡΟΦΟΡΙΩΝ
UC6 ΕΜΦΑΝΙΣΗ ΙΣΤΟΡΙΚΟΥ

2.1.1 TRACEABILITY MATRIX

| | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 |
|----------|-----|-----|-----|-----|-----|-----|
| T1_V0_01 | X | | | | | |
| T2_V0_01 | | X | | | | |
| T2_V0_02 | | X | | | | |
| T3_V0_01 | | | X | | | |
| T3_V0_02 | | | X | | | |
| T3_V0_03 | | | X | | | |
| T4_V0_01 | | | | X | | |
| T4_V0_02 | | | | X | | |
| T4_V0_03 | | | | X | | |
| T5_V0_01 | | | | | X | |
| T5_V0_02 | | | | | X | |
| T5_V0_03 | | | | | X | |
| T5_V0_04 | | | | | X | |
| T6_V0_01 | | | | | | X |
| T6_V0_02 | | | | | | X |

2.1.2 USE CASE UC1: LOAD DATA

Involved methods

```
MainEngine.setTimeLine(),  
IParser.parse --> SimpleTextParser.parse(filename)
```

Test cases

| | | |
|----------------|-----------|---|
| ID | T1_V0_01 | HappyDayScenario for |
| Description | ON | any context |
| | RECEIVING | Request to create a new file |
| | ENSURE | That the System |
| | OUTPUTS | A file that is used to track the history of the users choices and when they were made (date). |
| | SUCH THAT | It can be available for the user |
| Pre-cond. | | No specific precondition constructed |
| Input | | Small strings , all valid |
| Output | | The collection of strings and the dates |
| Post-cond. | | No state properties tested |
| Method To test | | |

| | | |
|----------------|----------|---|
| ID | T1_V0_02 | HappyDayScenario for MainEngine.setTimeLine() |
| | | ...identical setup with T1_V0_01 |
| Method To test | | MainEngine.setTimeLine(filename) |

Not designed yet: T1_V1: missing file, T1_V2: invalid values in input file

2.1.3 USE CASE UC2: ANALYZE TIMELINE (AND PRODUCE PHASES)

Involved methods

Test cases

| | | |
|----------------|-----------|---|
| ID | T2_V0_01 | HappyDayScenario |
| Description | ON | Choosing a program function |
| | RECEIVING | The wanted program function |
| | ENSURE | That the System |
| | OUTPUTS | The wanted results |
| | SUCH THAT | The choice's methods are followed |
| Pre-cond. | | The program accesses the history file |
| Input | | An int from 1 to 3 |
| Output | | None |
| | | If 3 is chosen the program ends, else the program |
| Post-cond. | | continues according to the users choice |
| Method To test | | |

2.1.4 USE CASE UC3: VISUALIZE TIMELINE

Involved methods

Test cases

ID T3_V0_01 HappyDayScenario for

| | | |
|----------------|-----------|---|
| Description | ON | Choosing an option from UC2 for the 1st time |
| | RECEIVING | A path |
| | | |
| | ENSURE | That the client properly provides a correct path |
| | OUTPUTS | None |
| | SUCH THAT | The system accesses a database |
| Pre-cond. | | This UC has not been accessed yet , the option 1 was Chosen (from the UC2) and the file is ok |
| Input | | None |
| | | |
| Output | | None |
| Post-cond. | | None |
| Method To test | | |

2.1.5 USE CASE UC4: VISUALIZE PHASES (TO CONSOLE)

Involved methods

Test cases

2.1.5 USE CASE UC4: VISUALIZE TIMELINE

Involved methods

Test cases

| | | |
|-------------|-----------|--|
| ID | T4_V0_01 | HappyDayScenario for |
| | | |
| Description | ON | Choosing the option 1 from UC2 |
| | RECEIVING | A path |
| | ENSURE | That the client properly provides a correct path |
| | OUTPUTS | None |
| | SUCH THAT | The system accesses a database |
| Pre-cond. | | This UC has not been accessed yet , the option 1 was |
| | | Chosen (from the UC2) and the file is ok |
| Input | | <ul style="list-style-type: none">• None |
| | | |
| Output | | None |
| Post-cond. | | None |

Method To test

.....

| | | |
|----------------|-----------|---|
| ID | T5_V0.1 | HappyDayScenario for |
| Description | ON | Creating the report file |
| | RECEIVING | Request to show the wanted info in the correct form |
| | ENSURE | That the System |
| | OUTPUTS | The necessary info correctly |
| | SUCH THAT | state is intact |
| Pre-cond. | | None |
| Input | | The info wanted and the form of the Report |
| Output | | The user's wanted info |
| Post-cond. | | A report file created in the path from the last Test Case |
| Method To test | | |

| | | |
|----------------|-----------|--|
| ID | T6_V0.1 | HappyDayScenario for |
| Description | ON | Showing the history of the program |
| | RECEIVING | The order to show the program's history |
| | ENSURE | That the System |
| | OUTPUTS | The program's history |
| | SUCH THAT | state is intact |
| Pre-cond. | | The system being able to access the date and time to Present it with the corresponding order of the user |
| Input | | Selecting the option 2 of the UC2 |
| Output | | The program's history |
| Post-cond. | | None |
| Method To test | | |

2.2 UNIT TESTS

Στην παρούσα υποενότητα, παραθέτουμε πολύ'συνοπτικά'τα πιο σημαντικά'unit test που σχεδιάστηκαν, οργανωμέ'να ανά'κλά'ση. Δί'νουμε μό'νο το OREOS text.
Class ParserFactory

ON any context, RECEIVING a request for a new parser, ENSURE THAT THE SYSTEM OUTPUTS a not null parser SUCH THAT state remains intact.
Class AnalyzerFactory

ON any context, RECEIVING a request for a new analyzer, ENSURE THAT THE SYSTEM OUTPUTS a not null analyzer of the correct kind SUCH THAT state remains intact.

2.3 ΕΚΚΡΕΜΟΤΗΤΕΣ (TODO)

Εκφρεμού'ν μη υλοποιημέ'νοι έ'λεγχοι ως ακολού'θως (αν υπάρχουν εκκρεμότητες, παραθέστε την TODO λίστα ελέγχων που πρέπει να ετοιμαστούν)

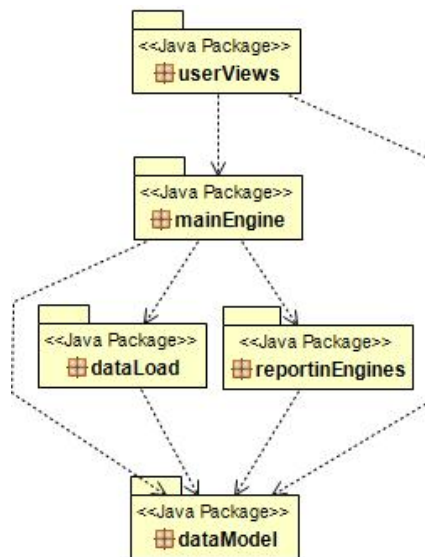
1. Unit tests are missing for several classes, both at the model and at the business logic level

3 ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

3.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ

και Η ανάλυση του πολυπλοκότητας του κώδικα του συστήματος και η επεξεργασία των πακέτων του συστήματος, με στόχο να γίνει ο κώδικας πιο εύκολο να διαχειριστείται, όταν το μέγεθος του συστήματος είναι μεγάλο.

Το διάγραμμα των πακέτων του συστήματος ακολουθεί στο Σχ. 1.



Σχήμα 1. Διάγραμμα πακέτων (εδώ: από την αξιολόγηση εστιατορίου)

Ακολουθεί μια συνοπτική περιγραφή των πακέτων του συστήματος.

ΠΑΚΕΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

| | |
|------------|--|
| userViews | Περιέχει τις boundary classes που είναι υπεύθυνες για την αλληλεπίδραση με το χρήστη |
| mainEngine | Κεντρική business logic engine, along with the necessary interface to export to the boundary classes |
| dataLoad | Υποσύστημα αλληλεπίδρασης με τα αρχεία δεδομένων, για την ανάκτησή τους από το σύστημα |

reportEngines Υποσύστημα παραγωγής αναφορών
dataModel Domain classes of the system

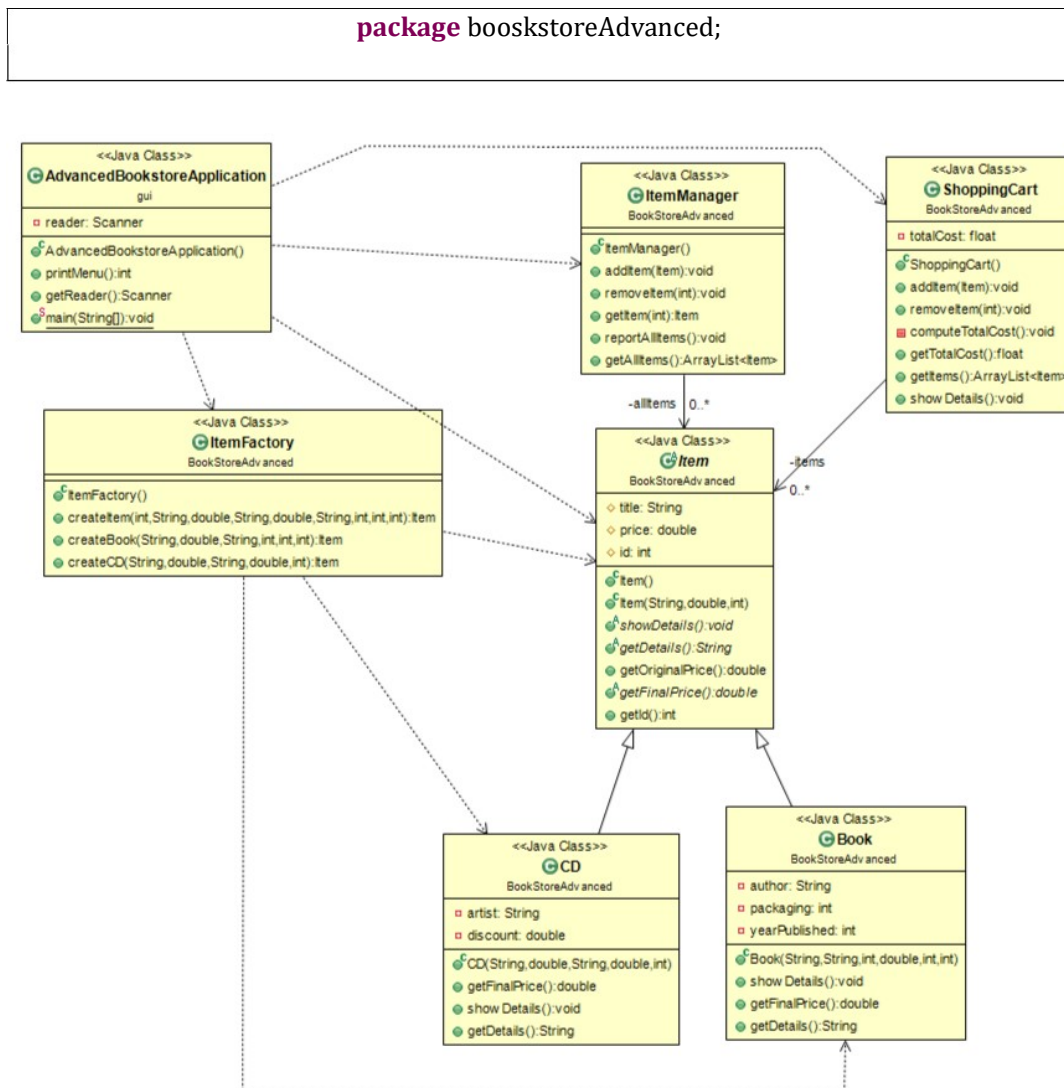
Πίνακας 3. Συνοπτική περιγραφή πακέτων συστήματος (εδώ: από την αξιολόγηση εστιατορίου)

3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

Στην παρούσα υποενότητα, παρατίθενται τα διαγράμματα κλάσεων και ακολουθιών.

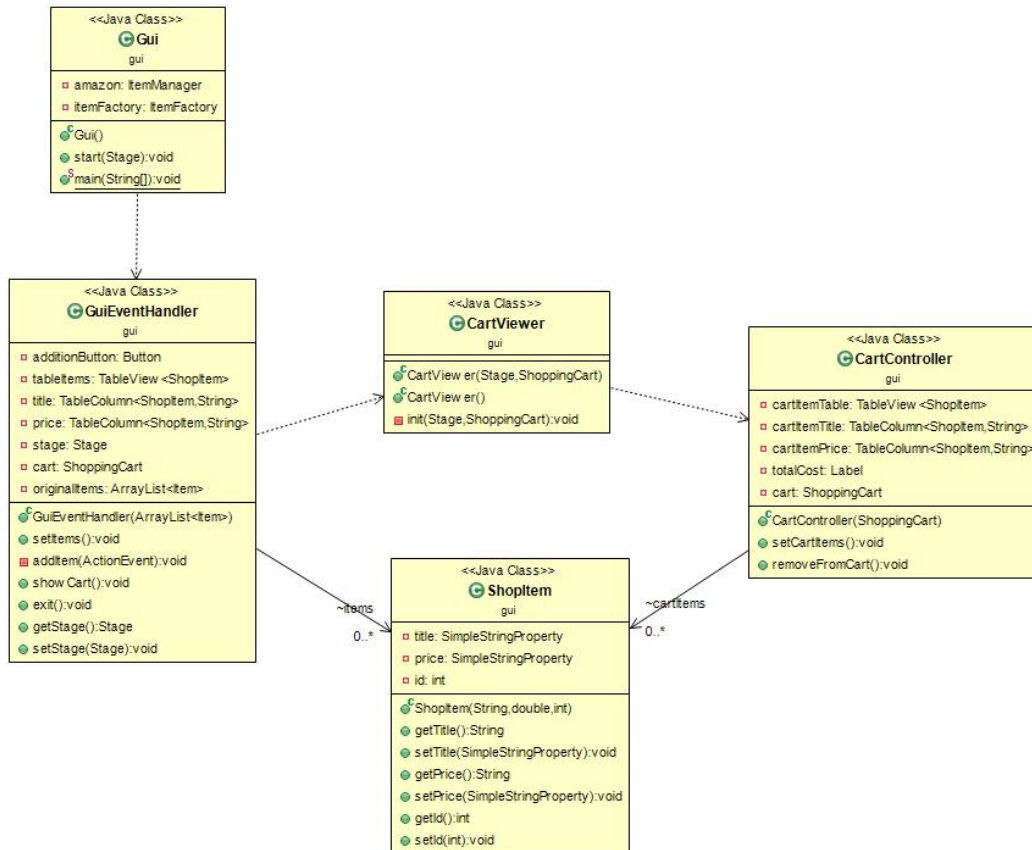
Π.χ., για το παράδειγμα με τον έλεγχο του βιβλιοπωλείου έχουμε περισσότερα του ενός διαγράμματα.

(ΠΡΟΣΟΧΗ: μόλις αλλάξαμε project απ' όπου δανειζόμαστε παραδείγματα!)



Σχήμα 2. Διάγραμμα κλάσεων για το πακέτο `bookstoreAdvanced`

package gui;



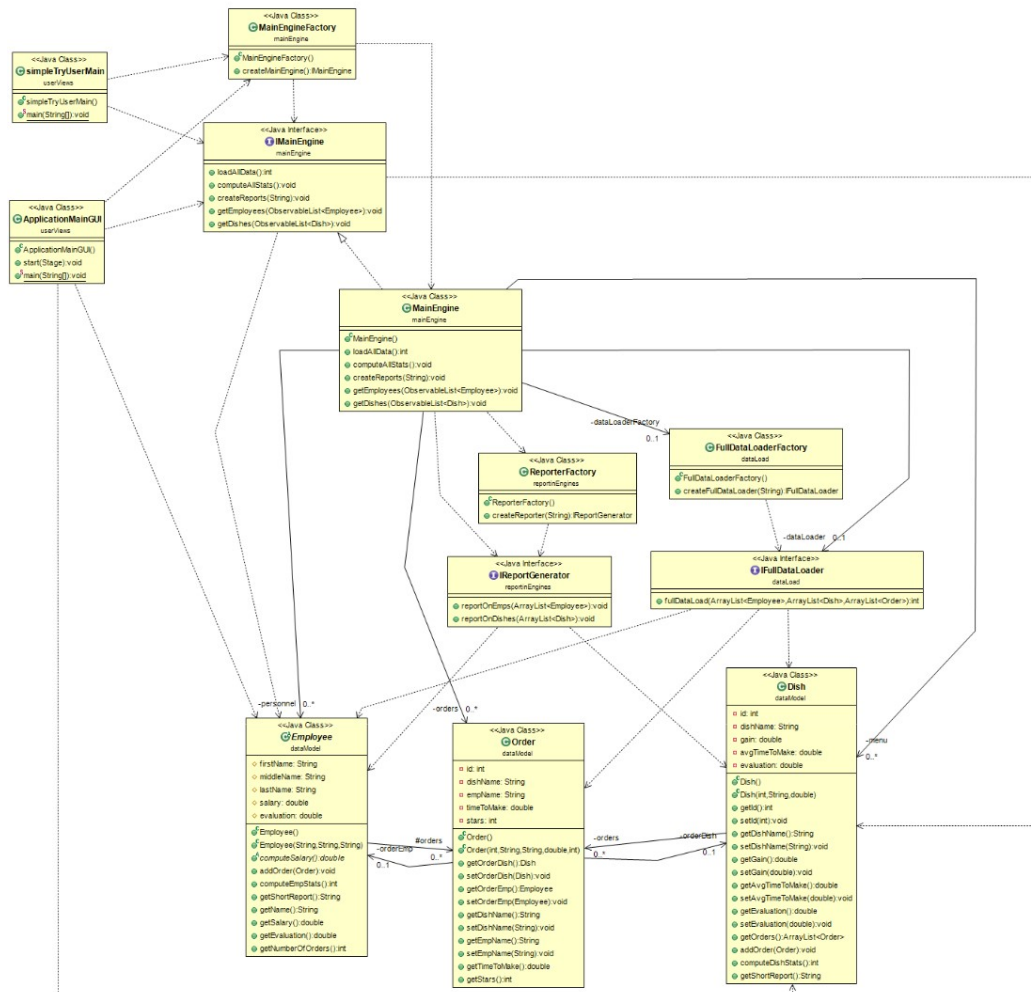
Σχήμα 3. Διάγραμμα κλάσεων για το πακέτο gui

Μπορείτε, επίσης, στα διαγράμματα που δίνετε, να βάζετε και συνεργαζόμενες κλάσεις από άλλα πακέτα. Το διάγραμμα κλάσεων του Σχ. 4 παρουσιάζει τις κεντρικές κλάσεις της εφαρμογής αξιολόγησης εστιατορίου, μαζί με τις συνεργαζόμενες κλάσεις.

- Πλεονεκτήματα: συνολική εσοπτεία του συστήματος σε μία (1) απεικόνιση.
- Μειονεκτήματα: η απεικόνιση είναι στα όρια του οπτικά διακριτού. Από ένα σημείο κι έπειτα, είναι αδύνατο να διακρίνει κανείς τι είναι στο διάγραμμα (ακόμα κι αν έχει ακολουθήσει μια προσεγμένη διάταξη στο χώρο), από το διάγραμμα γίνεται πρακτικά άχρηστο. (θυμηθείτε: το διάγραμμά σας, εγώ θα το δω σε χαρτί – θα έχουμε μπροστά μας μόνο ό,τι τυπώσετε)

Παρατηρήστε επίσης, ότι αν τυχόν χρειάζεται να τεκμηριώσουμε / εξηγήσουμε / ανακτήσουμε την συνεργασία ενός υποσυνόλου κλάσεων, είναι απολύτως νόμιμο να κατασκευάζουμε διαγράμματα με κλάσεις από διαφορετικά πακέτα -- δεν μας περιορίζουν τα όρια των πακέτων, δηλ. Τα διαγράμματα πρέπει να εξυπηρετούν μια ομάδα ανάπτυξης λογισμικού και όχι οι άνθρωποι τα διαγράμματα.

(Εννοείται πως αυτό δεν είναι δικαιολογία για να τα ισοπεδώσουμε όλα...)



Σχήμα 4. Διάγραμμα κλάσεων επεξήγησης ενός πακέτου με συνεργαζόμενες κλάσεις

3.3 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΩ'Ν Δv

ζητού'νται / υπά'ρχουν

3.4 ΑΝΑΛΥΣΗ ΚΛΑΣΕΩΝ ΚΑΙ ΣΥΝΕΠΕΙΑ ΠΡΟΣ ΤΙΣ ΑΠΑΙΤΗΣΕΙΣ Στην

παρου'σα ενό'τητα παραθέ'τουμε μια ανά'λυση των κλά'σεων και μια τεκμηρί'ωση της κά'λυψης των βασικώ'ν use cases του συστή'ματος .

Πρέπει ΥΠΟΧΡΕΩΤΙΚΑ να μου εξηγήσετε:

(α) Την ταξινόμηση των κλάσεων σε Domain/Business Logic/Boundary classes

(β) Tα interfaces between subsystems (emph., for Business Logic classes)

(γ) Την απεικόνιση των use cases σε μεθόδους (όχι σε κλάσεις, σε μεθόδους)

Αυτού'του εί'δους η τεκμηρί'ωση δεν θα υπή'ρχε σε μια επαγγελματική'αναφορά'—
ό'μως, επαληθεύ'ει την οργά'νωση και την πληρό'τητα της σχεδί'ασή'ς σας.

3.4.1 DOMAIN CLASSES

Package CD, Book, μια αφηρημένη κλάση Item (A) για αυτές τα δύο, bookstoreAdvanced και ένα factory, το ItemFactory, για την κατασκευή στιγμιούτων. hopping art.

3.4.2 BUSINESS LOGIC CLASSES

Package ItemManager, για την υλοποίηση όλων των use cases στο bookstoreAdvanced back-end. ItemManager :

- Interfaces with domain classes via Item + Factory.
- **Has no interface to boundary classes (!!!)**

3.4.3 BOUNDARY CLASSES

Package gui AdvancedBookstoreApplication, μια client class για την αλληλεπίδραση με το χρήστη μέσω κονσόλας.

Gui, GuiEventHandler, CartViewer, CartController, ShoppingItem, για το graphical user interface με το χρήστη (οι handler/controller classes χειρίζονται τα events από τη γραφική διαπροσωπεία).

3.4.4 ΑΠΕΙΚΟΝΙΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΕ ΜΕΘΟΔΟΥΣ

Υπάρχουν 4 use cases για το ηλ. βιβλιοπωλείο: εμφάνισε προϊόντα καταστήματος, πρόσθεσε προϊόν στο καλάθι, διέγραψε προϊόν από το καλάθι, εμφάνισε προϊόντα του καλαθιού.

ΑΠΕΙΚΟΝΙΣΗ USE CASES ΣΕ ΜΕΘΟΔΟΥΣ

| Use case | Back-end methods | Front-end methods |
|--------------------------------|------------------------------|---|
| εμφάνισε προϊόντα καταστήματος | ItemManager.reportAllItems() | Gui.start() GuiEventHandler.setItems() |
| πρόσθεσε προϊόν στο καλάθι | ShoppingCart.addItem() | GuiEventHandler.addItem() |
| διέγραψε προϊόν από το καλάθι | ShoppingCart.removeItem() | CartController.removeFromCart() |
| εμφάνισε προϊόντα του καλαθιού | ShoppingCart.showDetails() | GuiEventHandler.showCart() |

Πίνακας 4 Επαλήθευση απεικόνισης use cases σε μεθόδους

3.5 ΕΚΚΡΕΜΟΤΗΤΕΣ (TODO)

Αν υπάρχουν. Π.χ., εδώ:

- Missing interface(s) between ItemManager and boundary classes

4 ΛΟΙΠΑ ΣΧΟΛΙΑ

Εδώ προστίθενται ό'ποια σχό'λια μπορεί'να υπά'ρχουν (αν υπά'ρχουν) για σχεδιαστικές υποθέ'σεις, αποφά'σεις, ελλεί'ψεις και σημεί'α κινδύ'νου, ή'για οτιδή'ποτε ά'λλο κρί'νετε σημαντικό'να καταγραφεί'για τη μελλοντική'συντή'ρηση του κώ'δικα.