ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ για το ακαδημαϊκο έτος 2022-2023

ΟΜΑΔΑ 4993-4970

ιωαννησ τσοχλασ, ΑΜ:4993

ιωαννησ γιαννακοσ, ΑΜ:4970

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2022

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

|  |  |  |  |
| --- | --- | --- | --- |
| Ημερομηνία | Έκδοση | Περιγραφή | Συγγραφείς |
| 2022/11/10 | v.01 | Οργάνωση απαιτήσεων σε use cases,complete oreos,uml diagramms | 4993,4970 |
| yyyy/mm/dd | v.02 | Αρχική σχεδίαση κλάσεων και ελέγχων | ΧΧ,ΥΥ,ΖΖ |
| yyyy/mm/dd | … | Διορθώσεις στις uses cases, επεκτάσεις στη σχεδίαση κλάσεων και ελέγχων | ΥΥ,ΖΖ (ο ΧΧ αποχώρησε) |
| yyyy/mm/dd | … | ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ | ΥΥ |

**Στην τρέχουσα σελίδα**, με **γκρι, αχνά γράμματα** παραθέτω οδηγίες, επεξηγήσεις και σχόλια. **Στη δική σας αναφορά, αυτό το κομμάτι προφανώς θα πρέπει να το σβήσετε.**

**Στο υπόλοιπο κείμενο,** για να μη περιπλέξω το στυλ περαιτέρω**, οι οδηγίες παρατίθενται σε απλό Βασικό/Normal στυλ. Εσείς εκεί πρέπει να βάλετε το δικό σας κείμενο στη θέση του δικού μου.**

ΟΔΗΓΙΕΣ

Αναμένεται να τηρήσετε το παρόν πρότυπο ΠΙΣΤΑ. Όχι επειδή είναι το απαύγασμα της καλαισθησίας ή της λειτουργικότητας, αλλά επειδή πρέπει να μάθετε να τηρείτε πρότυπα με πειθαρχία και συνέπεια. Για το λόγο αυτό, καλείσθε να ΜΕΙΝΕΤΕ ΠΙΣΤΑ ΣΤΟ ΣΤΥΛ ΤΟΥ ΚΕΙΜΕΝΟΥ! Είναι ευκαιρία επίσης, να μάθετε να χρησιμοποιείτε με ΕΥΧΕΡΕΙΑ επεξ. κειμένου.

Κατά τα λοιπά:

1. Συμπληρώστε με τα σωστά στοιχεία το εξώφυλλο και το header, όπου υπάρχουν κόκκινα γράμματα. Αν δεν σας έχει δοθεί α/α ομάδας, είναι το concatenation των ΑΜ σας, με παύλες ανάμεσα, ταξινομημένα με αύξουσα σειρά.
2. Ο παραπάνω πίνακας «Ιστορικό Εκδόσεων» συμπληρώνεται κάθε φορά που αλλάζετε κάτι στην αναφορά σας. Οι καταχωρήσεις που υπάρχουν ήδη είναι απλά ενδεικτικές.
3. Στις Use Cases συμπληρώνετε ΟΛΕΣ τις Use Cases που έχετε εξάγει (εδώ παρατίθεται μόνο μία, ενδεικτικά). Το στυλ είναι από το υπόδειγμα για use cases που χρησιμοποιείται στο μάθημα.
4. Δώστε ιδιαίτερη σημασία στους ελέγχους, καθώς είναι από τις λίγες φορές στην εκπαίδευσή σας που εξετάζεστε για την απάντηση που δώσατε στο ερώτημα «Πώς θα επιβεβαιώσω ότι ο κώδικάς μου κάνει αυτό που πρέπει?» (θυμηθείτε ότι στις συνεντεύξεις για δουλειά, ερωτήσεις επί του testing είναι από τις πιο κλασικές επιλογές των interviewers).
5. Συμπληρώστε τα διαγράμματα με τις σχετικές εικόνες. Προσοχή: τα σχήματα πρέπει να είναι ΕΥΔΙΑΚΡΙΤΑ. Ο σκοπός των διαγραμμάτων είναι να μπορούν να μεταδώσουν ΣΤΟΥΣ ΑΛΛΟΥΣ τις σχεδιαστικές σας αποφάσεις.

# Ανάλυση Απαιτήσεων – Use Cases

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

Φόρτωσε

ID: UC 1

Description and Goal

Η use case «Φόρτωσε» εμφανίζει τα διαθέσιμα προϊόντα του καταστήματος στην οθόνη.

Actors

Ο Χρήστης

Basic Flow

1. To use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού File την επιλογή «Load TSV».
2. Το σύστημα εμφανίζει ταξινομημένο το αρχείο στην οθόνη.

Extensions / Variations

Στην περίπτωση κατά την οποία δεν υπάρχει το αρχείο εμφανίζεται ένα μήνυμα που ενημερώνει ότι δεν υπάρχει ή δεν μπορεί να ανοιχτεί .

Post conditions

Ο χρήστης παραλαμβάνει φορτωμένο και ταξινομημένο το αρχείο

Ας

Να έχει φορτωθεί ταξινομημένο το αρχείο

1. το

2.

3.

Επεστρεψε Εργασια Βαση Id

ID: UC 3

Description and Goal

Η use case «ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΑ ΒΑΣΗ ID» εμφανίζει την εργασία που έχει το επιθυμητό Id

Actors

Ο Χρήστης

PRECONDITIONS

Να έχει φορτωθεί ταξινομημένο το αρχείο

Basic Flow

1. To use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού Filters την επιλογή «Filter by ID».
2. Στο παράθυρο αναζήτησης o χρήστης πληκτρολογεί το επιθυμητό taskID .
3. Το σύστημα εμφανίζει τo αποτελέσμα στην οθόνη.

Post conditions

Ανάκτηση εργασίας με συγκεκριμένο taskId

Επεστρεψε Εργασιεσ Κορυφαιου Επιπέδου

ID: UC 4

Description and Goal

Η use case «ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΚΟΡΥΦΑΙΟΥ ΕΠΙΠΕΔΟΥ» εμφανίζει τις εργασίες κορυφαίου επιπέδου(MamaId=0).

Actors

Ο Χρήστης

PRECONDITIONS

Να έχει φορτωθεί ταξινομημένο το αρχείο

Basic Flow

1. To use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού Filters την επιλογή «Top Level Tasks».
2. Το σύστημα εμφανίζει τo αποτέλεσμα στην οθόνη.

Post conditions

Ανάκτηση των εργασιών κορυφαίου επιπέδου

Δημιούργησε Αναφορά

ID: UC 5

Description and Goal

Η use case «ΔΗΜΙΟΥΡΓΗΣΕ ΑΝΑΦΟΡΑ» εξάγει τον τύπο αρχείου που έχουμε επιλέξει

Actors

Ο Χρήστης

PRECONDITIONS

Να έχει φορτωθεί ταξινομημένο το αρχείο

Basic Flow

1. To use case ξεκινάει όταν ο χρήστης επιλέξει το μενού Reports
2. Το σύστημα αποθηκεύει τo αρχείο στον υπολογιστή.

Extensions/Variations

1α. Όταν ο χρήστης επιλέξει την μορφή “Report Txt” η αναφορά αποθηκεύεται στον υπολογιστή σε μορφή κειμένου

1α. Όταν ο χρήστης επιλέξει την μορφή “Report Μd” η αναφορά αποθηκεύεται στον υπολογιστή σε μορφή markdown

1α. Όταν ο χρήστης επιλέξει την μορφή “Report Html” η αναφορά αποθηκεύεται στον υπολογιστή σε μορφή Html (Hypertext Markup Language)

Post conditions

Αποθήκευση σε απλό κείμενο, html και markdown

# Σχεδίαση Ελέγχων

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

## Ελεγχος use cases via system tests

Στην αρχική σχεδιαστική φάση, αρκεί να συμπληρώσετε την λεκτική περιγραφή με τις OREOS προδιαγραφές. Στην τελική φάση, συμπληρώστε και τις λεπτομέρειες σε σχέση με τις εμπλεκόμενες μεθόδους και το setup input, output, pre-post conditions, ...

### **Use case UC1: Φορτωσε**

**Test cases**

|  |  |  |
| --- | --- | --- |
| *Description* | *ON* | *any context* |
|  | *RECEIVING* | *Request to load a specific txt file* |
|  | *ENSURE* | *That the System* |
|  | *OUTPUTS* | *Stored data from the file as sorted (Tasks) objects in arraylist* |
|  | *SUCH THAT* | *state is intact* |

|  |  |  |
| --- | --- | --- |
| ID | T1\_V0\_01 | HappyDayScenario for SimpleTextParser.parse() |
| Pre-cond. |  | No specific precond constructed |
| Input |  | input\_test.txt, a small file with less than 10 entries, all valid |
| Output |  | a timeline with the same #entries as the contexts of input\_test.txt and no offending values |
| Post-cond. |  | No state properties tested |
| Method To test |  | SimpleTextParser.parse(filename) |

|  |  |  |
| --- | --- | --- |
| ID | T1\_V0\_02 | HappyDayScenario for MainEngine.setTimeLine() |
|  |  | …identical setup with T1\_V0\_01 |
| Method To test |  | MainEngine.setTimeLine(filename) |

**Involved methods**

MainEngine.setTimeLine(),

IParser.parse --> SimpleTextParser.parse(filename)

**Not designed yet**: T1\_V1: missing file, T1\_V2: invalid values in input file

### **Use case UC2: ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΠΟΥ ΠΛΗΡΟΥΝ ΤΟ ΚΡΙΤΗΡΙΟ ΑΝΑΖΗΤΗΣΗΣ**

**Test cases**

|  |  |  |
| --- | --- | --- |
| *Description* | *ON* | *A loaded file* |
|  | *RECEIVING* | *Request to return tasks by prefix* |
|  | *ENSURE* | *That the System* |
|  | *OUTPUTS* | *a set of tasks that match the prefix* |
|  | *SUCH THAT* | *state is intact* |

|  |  |  |
| --- | --- | --- |
| ID | T2\_V0\_01 | HappyDayScenario for NaiveAnalyser.producePhases() |
| Pre-cond. |  | Load input\_test.txt, a small file with less than 10 entries, all valid, for o(5) phases, and produce timeline |
| Input |  | the abovementioned timeline |
| Output |  | A correct #phases, with the correct points inside |
| Post-cond. |  | No state properties tested |
| Method To test |  | NaiveAnalyser.producePhasesFromTimeLine (TimeLine) |

|  |  |  |
| --- | --- | --- |
| ID | T2\_V0\_02 | HappyDayScenario for MainEngine.producePhases() |
|  |  | …identical setup with T2\_V0\_01 |
| Method To test |  | MainEngine.producePhases() |

**Involved methods**

MainEngine.producePhases()

AnalyserFactory.createAnalyzer()

IAnalyzer --> NaiveAnalyser.producePhasesFromTimeLine (TimeLine)

**Not designed yet**: T2\_V1: null timeline, T2\_V2: timeline with only one phase

### **Use case UC3: ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΑ ΒΑΣΗ ID**

**Test cases**

|  |  |  |
| --- | --- | --- |
| *Description* | *ON* | *A loaded file* |
|  | *RECEIVING* | *Request to return tasks by id* |
|  | *ENSURE* | *That the System* |
|  | *OUTPUTS* | *The task that match the id* |
|  | *SUCH THAT* | *state is intact* |

|  |  |  |
| --- | --- | --- |
| ID | T3\_V0\_01 | HappyDayScenario for MainEngine.visualize() |
| Pre-cond. |  | Load input\_test.txt, a small file with less than 10 entries, all valid, for o(5) phases, and produce timeline |
| Input |  | the abovementioned timeline, “HtmlVisualizer” as the tested visualizer |
| Output |  | A correct visualization, expressed as a 2D raster of chars |
| Post-cond. |  | No state properties tested |
| Method To test |  | MainEngine.setVisualizer(String)  MainEngine.visualize() |

|  |  |  |
| --- | --- | --- |
| ID | T3\_V0\_02 | HappyDayScenario for MainEngine.visualize() |
|  |  | …identical setup with T2\_V0\_01 |
| Input |  | “ConsoleVisualizer” as the tested visualizer |

**Involved methods**

MainEngine.visualize()

### **Use case UC4: ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΚΟΡΥΦΑΙΟΥ ΕΠΙΠΕΔΟΥ**

**Test cases**

|  |  |  |
| --- | --- | --- |
| *Description* | *ON* | *A loaded file* |
|  | *RECEIVING* | *Request to return top level tasks* |
|  | *ENSURE* | *That the System* |
|  | *OUTPUTS* | *a set of top level tasks* |
|  | *SUCH THAT* | *state is intact* |

|  |  |  |
| --- | --- | --- |
| ID | T4\_V0 | HappyDayScenario for MainEngine.reportPhases() |
| Pre-cond. |  | Load input\_test.txt, a small file with less than 10 entries, all valid, for o(5) phases, produce timeline and analyze it to phases |
| Input |  | the abovementioned timeline and its phases |
| Output |  | A correct visualization (approximation: the size of the descriptions produced is the same with the number of phases) |
| Post-cond. |  | No state properties tested |
| Method To test |  | MainEngine. reportPhases() |

**Involved methods**

MainEngine.reportPhases()

NaiveAnalyser.reportToConsole()

Phase.consoleVerticalReport()

### **Use case UC5: ΔΗΜΙΟΥΡΓΗΣΕ ΑΝΑΦΟΡΑ**

**Test cases**

|  |  |  |
| --- | --- | --- |
| *Description* | *ON* | *A loaded file* |
|  | *RECEIVING* | *Request to extract a report* |
|  | *ENSURE* | *That the System* |
|  | *OUTPUTS* | *A report in selected format* |
|  | *SUCH THAT* | *state is intact* |

## Traceability Matrix

Η αντιστοίχιση use cases σε id’s φαίνεται στον Πίνακα 1:

|  |  |
| --- | --- |
| UC1 | Load Data |
| UC2 | Analyze TimeLine |
| UC3 | Present TimeLine |
| UC4 | Present Phases |

Πίνακας 1 Σύνοψη use cases και των id’s τους

Ο Πίνακας 2 είναι ο traceability matrix για τους ελέγχους μας. Στη συνέχεια, οι έλεγχοι επεξηγούνται πιο αναλυτικά.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | UC1 | UC2 | UC3 | UC4 |
| T1\_V0\_01 | X |  |  |  |
| T1\_V0\_02 | X |  |  |  |
| T2\_V0\_01 |  | X |  |  |
| T2\_V0\_02 |  | X |  |  |
| T3\_V0\_01 |  |  | X |  |
| T3\_V0\_02 |  |  | X |  |
| T4\_V0 |  |  |  | X |

Πίνακας 2 Traceability matrix between use cases and tests

## ΕΚΚΡΕΜΟΤΗΤΕΣ (ToDo)

Εκκρεμούν μη υλοποιημένοι έλεγχοι ως ακολούθως *(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, specifically, class XXX, YYY, ZZZ

# Σχεδίαση Λογισμικού

## Διαγράμματα ΠΑΚΕΤΩΝ / υποσυστημάτων

To διάγραμμα των πακέτων του συστήματος ακολουθεί στο Σχ. 1. Diagram

Description automatically generated

Σχήμα 1. Διάγραμμα πακέτων

Ακολουθεί μια συνοπτική περιγραφή των λειτουργικών πακέτων του συστήματος.

(τα πακέτα που είναι υπεύθυνα για τα γραφικά δεν παρουσιάζονται εδώ)

ΠΑΚΕΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

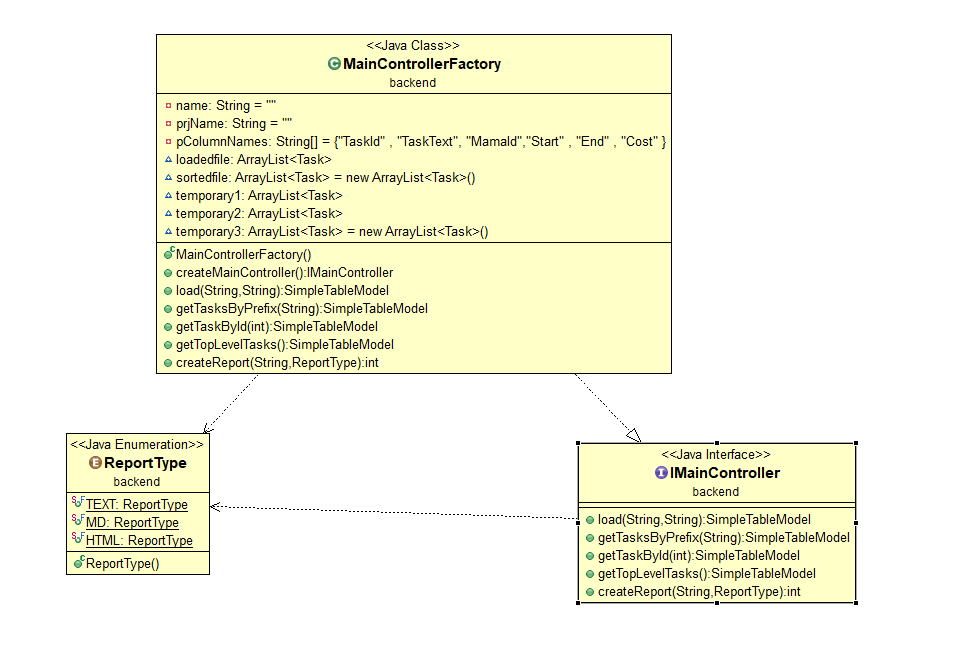
|  |  |
| --- | --- |
| task | Περιέχει data base που είναι υπεύθυνη για την αλληλεπίδραση με το σύστημα |
| backend | Κεντρική business logic engine, along with the necessary interface to export to the boundary classes |
| Searchmethods | Υποσύστημα αλληλεπίδρασης με τα αρχεία δεδομένων, για την ανάκτησή τους από το σύστημα |
| reports | Υποσύστημα παραγωγής αναφορών |
| filemanagment | Υποσύστημα αλληλεπίδρασης με τα αρχεία εισαγωγής, για την αποθήκευση τους από το σύστημα |

Πίνακας 3. Συνοπτική περιγραφή πακέτων συστήματος

## Διαγράμματα Κλάσεων

Στην παρούσα υποενότητα, παρατίθενται τα διαγράμματα κλάσεων και ακολουθιών.

**Package** backend;



**Package** filemanagment;

Graphical user interface, text

Description automatically generated

**package** searchmethods;

Table

Description automatically generated with low confidence

**Package** tasks;

A picture containing table

Description automatically generated

**Package** reports;

Diagram

Description automatically generated

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 εποπτεία 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), οπότε 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, δλδ. 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…)

Diagram, engineering drawing

Description automatically generated

Σχήμα 4. Διάγραμμα κλάσεων επεξήγησης ενός πακέτου με συνεργαζόμενες κλάσεις

## αναλυση κλασεων και συνεπεια προς τις απαιτησεισ

Στην παρούσα ενότητα παραθέτουμε μια ανάλυση των κλάσεων και μια τεκμηρίωση της κάλυψης των βασικών use cases του συστήματος .

**Πρέπει ΥΠΟΧΡΕΩΤΙΚΑ να μου εξηγήσετε:**

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

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

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

Αυτού του είδους η τεκμηρίωση δεν θα υπήρχε σε μια επαγγελματική αναφορά – όμως, επαληθεύει την οργάνωση και την πληρότητα της σχεδίασής σας.

### Domain Classes

|  |  |
| --- | --- |
| Package booskstoreAdvanced | CD, Book, μια αφηρημένη κλάση Item (A) για αυτές τα δύο, και ένα factory, το ItemFactory, για την κατασκευή στιγμιοτύπων. ShoppingCart. |

### Business Logic Classes

|  |  |
| --- | --- |
| Package booskstoreAdvanced | ItemManager, για την υλοποίηση όλων των use cases στο back-end. ItemManager:   * Interfaces with domain classes via Item + Factory. * Has no interface to boundary classes (!!!) |

### 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). |

### Απεικόνιση απαιτήσεων σε μεθόδους

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

## Διαγράμματα ακολουθιών

Αν ζητούνται / υπάρχουν

# λοιπά σχόλια

Εδώ προστίθενται όποια σχόλια μπορεί να υπάρχουν (αν υπάρχουν) για σχεδιαστικές υποθέσεις, αποφάσεις, ελλείψεις και σημεία κινδύνου, ή για οτιδήποτε άλλο κρίνετε σημαντικό να καταγραφεί για τη μελλοντική συντήρηση του κώδικα.

## ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΠΟΦΑΣΕΙΣ

Αν υπάρχει λόγος να καταγραφούν εναλλακτικές σχεδιάσεις και γιατί αποφασίσθηκε να προκριθεί κάποια από αυτές.

## ΣΗΜΕΙΑ ΚΙΝΔΥΝΟΥ

Αν υπάρχουν.

## ΕΚΚΡΕΜΟΤΗΤΕΣ (ToDo)

Αν υπάρχουν. Π.χ., εδώ:

* Missing interface(s) between ItemManager and boundary classes