ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ για το ακαδημαϊκο έτος 2022-2023

ΟΜΑΔΑ 4970-4993

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 |
| 2022/11/20 | v.01.5 | Διορθώσεις στις uses cases,προσθήκη ΟtherComments ώστε να συσχετιστεί με τον κώδικα | 4993,4970 |
| 2022/11/26 | v.02 | Αρχική σχεδίαση κλάσεων και ελέγχων, | 4993,4970 |
| 2022/12/05 | v.03 | Διορθώσεις στις uses cases, επεκτάσεις στη σχεδίαση κλάσεων και ελέγχων | 4993,4970 |
| 2022/12/08 | v.04 | Υλοποίηση Test Case | 4993,4970 |
| 2022/12/13 | v.05 | ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ | 4993,4970 |

# 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 ταξινομημένο το αρχείο κειμένου που έχουμε επιλέξει .

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

Other comments

Σε αντιστοιχία με το project αντιπροσωπεύει το πακέτο filemanagment και κατά συνέπια τις κλάσεις FileOpen.java η οποία είναι υπεύθυνη για να ανοίξει ένα αρχείο tsv από το σύστημα ώστε να μπορεί να το χειριστεί η εφαρμογή μας και την κλάση Classifification.java η οποία επιστρέφει ταξινομημένο το αρχείο (διάγραμμα gant )

Επέστρεψε Εργασίες που Πληρούν το Κριτήριο Αναζήτησης

ID: UC 2

Description and Goal

Η use case «ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΠΟΥ ΠΛΗΡΟΥΝ ΤΟ ΚΡΙΤΗΡΙΟ ΑΝΑΖΗΤΗΣΗΣ» εμφανίζει ταξινομημένες, μόνο τις εργασίες που η περιγραφή τους ξεκινά (prefix) με ένα δοθέν string .

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 Prefix».

2. Στο παράθυρο αναζήτησης o χρήστης πληκτρολογεί το επιθυμητό string .

3. Το σύστημα εμφανίζει τα αποτελέσματα στην οθόνη.

Post conditions

Ανάκτηση όλων των εργασιών που πληρούν το ίδιο κριτήριο περιγραφής

Other comments

Σε αντιστοιχία με το project αντιπροσωπεύει την κλάση TasksByPrefix.java του πακέτου searchmethods

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 αποτέλεσμα στην οθόνη.

Post conditions

Ανάκτηση εργασίας με συγκεκριμένο taskID

Other comments

Σε αντιστοιχία με το project αντιπροσωπεύει την κλάση TaskById.java του πακέτου searchmethods

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

Other comments

Σε αντιστοιχία με το project αντιπροσωπεύει την κλάση TopLevelTasks.java του πακέτου searchmethods

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

Other comments

Σε αντιστοιχία με το project αντιπροσωπεύει τo πακέτο report και κατά αντιστοιχία τις ακόλουθες κλάσεις TextReport.java , MarkdownReport.java , HtmlReport.java επιπρόσθετα να σημειωθεί ότι υπάρχει μια βοηθητική κλάση FilterReport.java στο πακέτο η οποία είναι υπεύθυνη για να ενεργοποιήσει την κατάλληλη κλάση (βάση το αρχείο που θέλουμε να εξάγουμε (.txt,.md,.html))

# Σχεδίαση Ελέγχων

Οι έλεγχοι που σχεδιάσθηκαν και εντάχθηκαν στην υλοποίηση περιγράφονται παρακάτω.

## Ελεγχος use cases via system tests

### **Use case UC1: Φορτωσε (filemanagmenttest)**

**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 loadfile() //load a file WITHOUT BEEING SORTED |
| Pre-cond. |  | No specific precond constructed |
| Input |  | Input a EggsScrambled.tsv, a small valid file |
| Output |  | The total lines form the file that we have just open |
| Post-cond. |  | No state properties tested |
| Method To test |  | fileOpenToTest.loadfile(); |

|  |  |  |
| --- | --- | --- |
| ID | T1\_V0\_02 | HappyDayScenario for load() //load and sort the file |
|  |  | …identical setup with T1\_V0\_01 |
| Method To test |  | testClassification.load(“path”,”delimiter”) |

|  |  |  |
| --- | --- | --- |
| ID | T1\_V0\_03 | Test if the file is not null testfileOpenNull() |
|  |  | …identical setup with T1\_V0\_01 |
| Method To test |  | testfileOpenNull() |

|  |  |  |
| --- | --- | --- |
| ID | T1\_V0\_04 | test if the constructor prevents creation with null delimiter testfileOpenNoDelimiter() |
|  |  | …identical setup with T1\_V0\_01 |
| Method To test |  | **testfileOpenNoDelimiter()** |

**Involved methods**

FileOpen()

**Not designed yet**: T1\_V0\_04 That test fails so we must improve FileOpen to prevent creation with null delimiter

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

**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 getByPrefixTest() |
| Pre-cond. |  | Load the sorted file |
| Input |  | The prefix |
| Output |  | A SimpleTableModel with the matching results |
| Method To test |  | appController.getTasksByPrefix(Prefix) |

**Involved methods**

MainControllerFactory()

### **Use case UC3: ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΑ ΒΑΣΗ ID (TASKBYIDTEST)**

**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 getByIdTest() |
| Pre-cond. |  | Load the sorted file |
| Input |  | The id |
| Output |  | A SimpleTableModel with the matching results |
| Method To test |  | appController.getTaskById(Id); |

**Involved methods**

MainControllerFactory()

### **Use case UC4: ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΚΟΡΥΦΑΙΟΥ ΕΠΙΠΕΔΟΥ (topleveltaskstest)**

**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\_01 | HappyDayScenario for getTopLevelTaskTest() |
| Pre-cond. |  | Load the sorted file |
| Output |  | A SimpleTableModel with the matching results |
| Method To test |  | appController.getTopLevelTasks() |

**Involved methods**

MainControllerFactory()

### **Use case UC5: ΔΗΜΙΟΥΡΓΗΣΕ ΑΝΑΦΟΡΑ (Reportstest)**

**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* |

|  |  |  |
| --- | --- | --- |
| ID | T4\_V0\_01 | HappyDayScenario for createReportTextTest() |
| Pre-cond. |  | Load the sorted file |
| Output |  | A text file |
| Method To test |  | appController.createReportText() |

|  |  |  |
| --- | --- | --- |
| ID | T4\_V0\_02 | HappyDayScenario for createReportHtmlTest() |
| Pre-cond. |  | Load the sorted file |
| Output |  | A html file |
| Method To test |  | appController.createReportHtml() |

|  |  |  |
| --- | --- | --- |
| ID | T4\_V0\_03 | HappyDayScenario for createReportMdTest() |
| Pre-cond. |  | Load the sorted file |
| Output |  | A markdown file |
| Method To test |  | appController.createReportMd() |

## Traceability Matrix

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

|  |  |
| --- | --- |
| UC1 | ΦΟΡΤΩΣΕ |
| UC2 | ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΠΟΥ ΠΛΗΡΟΥΝ ΤΟ ΚΡΙΤΗΡΙΟ ΑΝΑΖΗΤΗΣΗΣ |
| UC3 | ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΑ ΒΑΣΗ ID |
| UC4 | ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΚΟΡΥΦΑΙΟΥ ΕΠΙΠΕΔΟΥ |
| UC5 | ΔΗΜΙΟΥΡΓΗΣΕ ΑΝΑΦΟΡΑ |

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 | UC5 |
| T1\_V0\_01 | X |  |  |  |  |
| T1\_V0\_02 | X |  |  |  |  |
| T1\_V0\_03 | X |  |  |  |  |
| T1\_V0\_04 | X |  |  |  |  |
| T2\_V0 |  | X |  |  |  |
| T3\_V0 |  |  | X |  |  |
| T4\_V0 |  |  |  | X |  |
| T5\_V0\_01 |  |  |  |  | X |
| T5\_V0\_02 |  |  |  |  | X |
| T5\_V0\_03 |  |  |  |  | X |

Πίνακας 2 Traceability matrix between use cases and tests

# 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

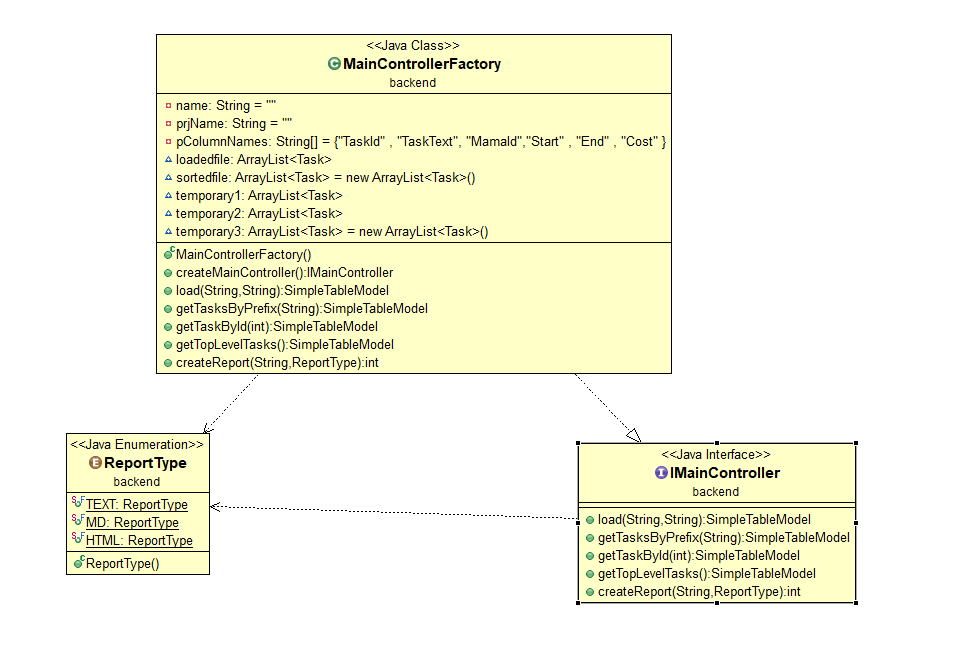
|  |  |
| --- | --- |
| task | Περιέχει data base που είναι υπεύθυνη για την αλληλεπίδραση με το σύστημα |
| backend | Κεντρική business logic engine, along with the necessary interface to export to the boundary classes |
| searchmethods | Υποσύστημα αλληλεπίδρασης με τα αρχεία δεδομένων, για την ανάκτησή 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 2. 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)**

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  Package  task | CD, Book, μια αφηρημένη κλάση Item (A) για αυτές τα δύο, και ένα factory, το ItemFactory, για την κατασκευή στιγμιοτύπων. ShoppingCart.  Task με τα πεδία του |

### Business Logic Classes

|  |  |
| --- | --- |
| Package  backend | 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 5 use cases για την απεικόνιση των διαγραμμάτων Gantt.

* ΦΟΡΤΩΣΕ
* ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΠΟΥ ΠΛΗΡΟΥΝ ΤΟ ΚΡΙΤΗΡΙΟ ΑΝΑΖΗΤΗΣΗΣ
* ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΑ ΒΑΣΗ ID
* ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΚΟΡΥΦΑΙΟΥ ΕΠΙΠΕΔΟΥ
* ΔΗΜΙΟΥΡΓΗΣΕ ΑΝΑΦΟΡΑ

ΑΠΕΙΚΟΝΙΣΗ USE CASES ΣΕ ΜΕΘΟΔΟΥΣ

|  |  |
| --- | --- |
| Use case | Back-end methods |
| * ΦΟΡΤΩΣΕ | appController.load() |
| * ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΠΟΥ ΠΛΗΡΟΥΝ ΤΟ ΚΡΙΤΗΡΙΟ ΑΝΑΖΗΤΗΣΗΣ | appController.  getByPrefix(Prefix) |
| * ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΑ ΒΑΣΗ ID | appController.  getById(Id); |
| * ΕΠΕΣΤΡΕΨΕ ΕΡΓΑΣΙΕΣ ΚΟΡΥΦΑΙΟΥ ΕΠΙΠΕΔΟΥ * ΔΗΜΙΟΥΡΓΗΣΕ ΑΝΑΦΟΡΑ | appController.  getTopLevel();  appController.createReportText  appController.createReportMd  appController.createReportHtml |

Πίνακας 4 Επαλήθευση απεικόνισης use cases σε μεθόδους

# λοιπά σχόλια

## ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΠΟΦΑΣΕΙΣ

Αποφασίσαμε να βάλουμε σε σχόλια ένα test (το οποίο υπάρχει μέσα στη FileManagement) σε σχόλια, διότι καλεί το system.exit(0) και αυτό έχει ως αποτέλεσμα να βγει από το JVM και κατά συνέπεια δε μας αφήνει να τρέξουμε τα υπόλοιπα test.

Ενδεικτική λύση θα ήτανε να χρησιμοποιούσαμε τη βιβλιοθήκη system lamda ή system rules, αλλά χρειάζονται τουλάχιστον JUnit 5 και 4.9 αντίστοιχα. Αλλά αυτό που έχουμε είναι 4.14.

## ΣΗΜΕΙΑ ΚΙΝΔΥΝΟΥ

1.Αν στο searchbyid εισάγεις αριθμό κρασαρει το πρόγραμμα