

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

---

## ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

---

---

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ  
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2022-2023

---

---

ΟΜΑΔΑ 5226-5380

ΚΩΝΣΤΑΝΤΙΝΟΣ ΖΩΗΣ, ΑΜ:5226

ΓΙΑΝΝΗΣ ΦΙΛΛΗΣ, ΑΜ:5380

---

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2022

## ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφείς
15/11/2022	v.01	Οργάνωση απαιτήσεων σε use cases Σχεδίαση των OREOS προδιαγραφών	Γιάννης Φίλλης Κωνσταντίνος Ζώης
17/12/2022	v.02	Διορθώσεις στις use cases και OREOS, Επεκτάσεις στη σχεδίαση ελέγχων και διαγραμμάτων	Γιάννης Φίλλης Κωνσταντίνος Ζώης
18/12/2022	v.03	ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ	Γιάννης Φίλλης Κωνσταντίνος Ζώης

## 1. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USE CASES

---

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

### LOADFILE

---

#### ID: UC 1

#### DESCRIPTION AND GOAL

Η use case «Load File» φορτώνει από ένα απλό αρχείο κειμένου το Gantt diagram.

#### ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης της εφαρμογής, Αρχείο εισόδου.

#### PRECONDITIONS

Πρέπει να υπάρχει το αρχείο σε μορφή κειμένου

#### BASIC FLOW

1. Το use case ξεκινά όταν ο χρήστης επιλέξει «Φόρτωση» από το μενού.
2. Το σύστημα εμφανίζει ένα παραθυράκι με τα αρχεία του υπολογιστή
3. Ο χρήστης επιλέγει το αρχείο που θέλει να φορτώσει.
4. Το σύστημα φορτώνει το αρχείο.
5. Το σύστημα εμφανίζει σε μια λίστα όλες τις κορυφαίου επιπέδου εργασίες καθώς και όλες τις υποεργασίες τους, ταξινομημένες και σε ένα διάγραμμα όλες τις εργασίες κορυφαίου επιπέδου και όλες τις υποεργασίες τους.

#### EXTENSIONS / VARIATIONS

3. Στην περίπτωση που ο χρήστης δεν επιλέξει τον σωστό τύπο αρχείου, το αρχείο δεν φορτώνεται.

#### POST CONDITIONS

# **FILTERBYTOPLEVEL**

---

## ID: UC 2

### DESCRIPTION AND GOAL

Η use case «FilterByTopLevel» εμφανίζει μόνο τις εργασίες κορυφαίου επιπέδου.

### ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης της εφαρμογής.

### PRECONDITIONS

Θα πρέπει το σύστημα να έχει φορτώσει ένα αρχείο κειμένου

### BASIC FLOW

1. Το use case ξεκινά όταν ο χρήστης επιλέξει “filter by top level” από το μενού «Filter».
2. Το σύστημα εμφανίζει ταξινομημένες τις εργασίες κορυφαίου επιπέδου σε μια λίστα και σε ένα διάγραμμα.

### EXTENSIONS / VARIATIONS

-

### POST CONDITIONS

-

# FILTERBYPREFIX

---

## ID: UC 3

### DESCRIPTION AND GOAL

Η use case «FilterByPrefix» εμφανίζει εργασίες που το όνομά τους ξεκινά με συγκεκριμένο πρόθεμα.

### ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης της εφαρμογής.

### PRECONDITIONS

Θα πρέπει το σύστημα να έχει φορτώσει ένα αρχείο κειμένου

### BASIC FLOW

1. Το use case ξεκινά όταν ο χρήστης επιλέξει “filter by prefix” από το μενού «Filter».
2. Το σύστημα εμφανίζει ένα παράθυρο για εισαγωγή κάποιου String.
3. Ο χρήστης πληκτρολογεί κάποιο string στην είσοδο με το οποίο θέλει να ξεκινάνε οι εργασίες.
4. Το σύστημα εμφανίζει ταξινομημένες τις εργασίες με το συγκεκριμένο πρόθεμα.

### EXTENSIONS / VARIATIONS

- 3.1 Στην περίπτωση κατά την οποία καμία εργασία δεν ξεκινά με το συγκεκριμένο πρόθεμα, το σύστημα εμφανίζει κενή λίστα.

### POST CONDITIONS

-

# FILTERBYTASKID

---

## ID: UC 4

### DESCRIPTION AND GOAL

Η use case «FilterByTaskID» εμφανίζει εργασίες που έχουν ένα συγκεκριμένο taskID.

### ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης της εφαρμογής.

### PRECONDITIONS

Θα πρέπει το σύστημα να έχει φορτώσει ένα αρχείο κειμένου

### BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει «Filter By ID» από το μενού «Filter».
2. Το σύστημα εμφανίζει ένα παράθυρο για εισαγωγή κάποιου ακεραίου.
3. Ο χρήστης πληκτρολογεί έναν τριψήφιο ακέραιο αριθμό, με το οποίο θέλει να ξεκινάει η εργασία που αναζητεί.
4. Το σύστημα εμφανίζει την εργασία με το συγκεκριμένο TaskID

### EXTENSIONS / VARIATIONS

- 3.1 Στην περίπτωση κατά την οποία καμία εργασία δεν έχει το συγκεκριμένο taskID , το σύστημα εμφανίζει κενή λίστα .

### POST CONDITIONS

-

# CREATEREPORT

## ID: UC 5

### DESCRIPTION AND GOAL

Η use case «CreateReport» αποθηκεύει την αναφορά των εργασιών στη μορφή που επιλέγει ο χρήστης.

### ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης της εφαρμογής.

### PRECONDITIONS

Θα πρέπει το σύστημα να έχει φορτώσει ένα αρχείο κειμένου

### BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει «Report» από το μενού.
2. Το σύστημα εμφανίζει μια λίστα με επιλογές αποθήκευσης: απλό tab-delimited κείμενο, markdown ή html.
3. Αν ο χρήστης επιλέξει «Report txt» .
  - 3.1. Το σύστημα εμφανίζει ένα παραθυράκι με τα αρχεία του υπολογιστή.
  - 3.2. Ο χρήστης επιλέγει τον φάκελο στον οποίο θέλει να αποθηκεύσει το αρχείο.
  - 3.3. Το σύστημα αποθηκεύει ένα αρχείο κειμένου στον φάκελο.
4. Αν ο χρήστης επιλέξει «Report md» .
  - 4.1. Το σύστημα εμφανίζει ένα παραθυράκι με τα αρχεία του υπολογιστή.
  - 4.2. Ο χρήστης επιλέγει τον φάκελο στον οποίο θέλει να αποθηκεύσει το αρχείο.
  - 4.3. Το σύστημα αποθηκεύει ένα αρχείο markdown στον φάκελο.
5. Αν ο χρήστης επιλέξει «Report html» .
  - 5.1. Το σύστημα εμφανίζει ένα παραθυράκι με τα αρχεία του υπολογιστή.
  - 5.2. Ο χρήστης επιλέγει τον φάκελο στον οποίο θέλει να αποθηκεύσει το αρχείο.
  - 5.3. Το σύστημα αποθηκεύει ένα αρχείο html στον φάκελο.

## EXTENSIONS / VARIATIONS

-

## POST CONDITIONS

Το αρχείο αποθηκεύεται στον χώρο αποθήκευσης του χρήστη.



# EXIT

## ID: UC 6

### DESCRIPTION AND GOAL

Η use case «Exit» ζητάει διαβεβαίωση ότι ο χρήστης θέλει να αποχωρήσει από το πρόγραμμα και αν επιλέξει «Ναι» τότε το πρόγραμμα κλείνει.

### ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης της εφαρμογής.

### PRECONDITIONS

-

### BASIC FLOW

1. Το use case ξεκινά όταν ο χρήστης επιλέξει «Έξοδος» από το μενού.
2. Το σύστημα εμφανίζει ένα μήνυμα ρωτώντας τον χρήστη αν θέλει να φύγει από το πρόγραμμα.
  - 2.1. Αν ο χρήστης επιλέξει «Ναι» τότε το πρόγραμμα τερματίζεται.
  - 2.2. Αν ο χρήστης επιλέξει «Όχι» τότε συνεχίζεται η κανονική ροή του προγράμματος.

### EXTENSIONS / VARIATIONS

-

### POST CONDITIONS

Αν ο χρήστης επιλέξει «Ναι», το πρόγραμμα τερματίζει.

## 2. ΣΧΕΔΙΑΣΗ ΕΛΕΓΧΩΝ

Οι έλεγχοι που σχεδιάσθηκαν και εντάχθηκαν στην υλοποίηση περιγράφονται παρακάτω.

### 2.1 ΕΛΕΓΧΟΣ USE CASES VIA SYSTEM TESTS

#### 2.1.1 USE CASE UC1: LOAD

##### Test cases

<i>Description</i>	<i>ON</i>	<i>Any context</i>
	<i>RECEIVING</i>	<i>A tab delimited text file which contains the tasks</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>A Gantt diagram of the tasks and a table with the tasks sorted</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T1_V0_01	HappyDayScenario for myLoader.parseFile()
Pre-cond.		No specific precondition constructed
Input		Eggs.tsv file, tab delimited with the tasks
Output		The correct number of lines in the file, each line being a task
Post-cond.		No state properties tested
Method To test		myLoader.parseFile(path,delimiter)

ID	T1_V0_02	RainyDayScenario for myLoader.parseFile()
Pre-cond.		No specific precondition constructed
Input		A wrong file path that does not exist
Output		Zero, as there is no such path
Post-cond.		No state properties tested
Method To test		myLoader.parseFile(path,delimiter)

##### Involved methods

```
myLoader.parseFile(filePath," ");
```

---

### 2.1.2 USE CASE UC2: FILTER BY TOP LEVEL

---

#### Test cases

<i>Description</i>	<i>ON</i>	<i>A tab delimited text file having been loaded into the system</i>
	<i>RECEIVING</i>	<i>Request to filter tasks by top level</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>A Gantt diagram of the top-level tasks and a table with the top-level tasks sorted</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T2_V0_01	HappyDayScenario for myFilter.filterByTopLevelTasks()
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		The abovementioned file
Output		The correct number of top-level tasks
Post-cond.		No state properties tested
Method To test		myFilter.filterByTopLevelTasks(myLoader)

ID	T2_V0_02	RainyDayScenario for myFilter.filterByTopLevelTasks()
Pre-cond.		File not loaded
Input		An empty task list
Output		Zero, as there are no tasks
Post-cond.		No state properties tested
Method To test		myFilter.filterByTopLevelTasks(wrongLoader);

#### Involved methods

```
myLoader.parseFile(filePath," ");  
  
myLoader.fillTable();  
  
myFilter.filterByTopLevelTasks(FileLoader);
```

### 2.1.3 USE CASE UC3: FILTER BY PREFIX

#### Test cases

<i>Description</i>	<i>ON</i>	<i>A tab delimited text file having been loaded into the system</i>
	<i>RECEIVING</i>	<i>A string as an input</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>A sorted table of the tasks starting with the requested prefix</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T3_V0_01	HappyDayScenario for myFilter.filterByPrefix()
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		A string
Output		The correct number of the tasks starting with the requested prefix
Post-cond.		No state properties tested
Method To test		myFilter.filterByPrefix(myLoader, string)

ID	T3_V0_02	RainyDayScenario for myFilter.filterByPrefix()
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		Nothing as input
Output		The number of all the tasks
Post-cond.		No state properties tested
Method To test		myFilter.filterByPrefix(myLoader, "");

ID	T3_V0_03	RainyDayScenario for myFilter.filterByPrefix()
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		A string that no tasks start with
Output		Zero
Post-cond.		No state properties tested
Method To test		myFilter.filterByPrefix(myLoader, wrong prefix);

#### Involved methods

```
myLoader.parseFile(filePath, " ");
```

```
myLoader.fillTable();
```

```
myFilter.filterByPrefix(FileLoader,String);
```

#### 2.1.4 USE CASE UC4: FILTER BY TASK ID

##### Test cases

<i>Description</i>	<i>ON</i>	<i>A tab delimited text file having been loaded into the system</i>
	<i>RECEIVING</i>	<i>A three-digit integer number as an input</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>The task with the ID that matches exactly the number the user entered</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T4_V0_01	HappyDayScenario for myFilter.filterByID();
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		A three-digit integer number
Output		The task that matches with the id given
Post-cond.		No state properties tested
Method To test		myFilter.filterByID(myLoader,int);

ID	T4_V0_02	RainyDayScenario for myFilter.filterByID();
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		A wrong number that doesn't match any task
Output		Zero
Post-cond.		No state properties tested
Method To test		myFilter.filterByID(myLoader,wrong number);

##### Involved methods

```
myLoader.parseFile(filePath,"");
```

```
myLoader.fillTable();
```

```
myFilter.filterByID(myLoader,int);
```

## 2.1.5 USE CASE UC5: CREATE REPORT

### Test cases

<i>Description</i>	<i>ON</i>	<i>A tab delimited text file having been loaded into the system</i>
	<i>RECEIVING</i>	<i>Request to save the file in a specific format (txt, html or markdown)</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>The file in the form the user selected</i>
	<i>SUCH THAT</i>	<i>The file has been successfully saved in the user's storage space</i>

ID	T5_V0_01	HappyDayScenario for myReporter.WriteTxtFile()
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		The abovementioned file
Output		True if the file exists, its path and the number of lines in the file
Post-cond.		No state properties tested
Method To test		myReporter.WriteTxtFile(filePath, myLoader);

ID	T5_V0_02	RainyDayScenario for myReporter.WriteTxtFile()
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		An empty task list
Output		True if the file exists and the number of lines for the header
Post-cond.		No state properties tested
Method To test		myReporter.WriteTxtFile(filePath, myLoader);

ID	T5_V0_03	RainyDayScenario for myReporter.WriteTxtFile()
Pre-cond.		Tab delimited file Eggs.tsv having been loaded into the system
Input		A wrong file path
Output		False, as the file does not exist
Post-cond.		No state properties tested
Method To test		myReporter.WriteTxtFile(wrong filePath, myLoader);

ID	T5_V0_04	HappyDayScenario for myReporter.WriteHtmlFile()
		...identical setup with T5_V0_01
Method To test		myReporter.WriteHtmlFile(filePath, myLoader);

ID	T5_V0_05	RainyDayScenario for myReporter.WriteHtmlFile()
		...identical setup with T5_V0_02
Method To test		myReporter.WriteHtmlFile(filePath, myLoader);

ID	T5_V0_06	HappyDayScenario for myReporter.WriteHtmlFile()
		...identical setup with T5_V0_03
Method To test		myReporter.WriteHtmlFile(wrong filePath, myLoader);

ID	T5_V0_07	HappyDayScenario for myReporter.WriteMarkdownFile()
		...identical setup with T5_V0_01
Method To test		myReporter.WriteMarkdownFile(filePath, myLoader);

ID	T5_V0_08	RainyDayScenario for myReporter.WriteMarkdownFile()
		...identical setup with T5_V0_02
Method To test		myReporter.WriteMarkdownFile(filePath, myLoader);

ID	T5_V0_09	HappyDayScenario for myReporter.WriteMarkdownFile()
		...identical setup with T5_V0_03
Method To test		myReporter.WriteMarkdownFile(wrong filePath, myLoader);

### Involved methods

```

myLoader.parseFile(filePath," ");
myLoader.fillTable();
myReporter.WriteTxtFile(filePath,FileLoader);
myReporter.WriteHtmlFile(filePath,FileLoader);
myReporter.WriteMarkdownFile(filePath,FileLoader);

```

---

### 2.1.6 USE CASE UC6: EXIT

---

### Test cases

Description	ON	Any context
	RECEIVING	Request to exit the program
	ENSURE	That the System
	OUTPUTS	A message to ensure that the user wants to leave. If the user selects "yes", the program terminates
	SUCH THAT	The program has been terminated successfully

No test cases implemented

## 2.2 TRACEABILITY MATRIX

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

UC1	Load
UC2	Filter By TopLevel
UC3	Filter By Prefix
UC4	Filter By Task ID
UC5	Save Report
UC6	Exit

Πίνακας 1 Σύνοψη use cases και των id's τους

Πίνακας 2 είναι ο traceability matrix για τους ελέγχους μας.

	UC1	UC2	UC3	UC4	UC5
T1_V0_01	X				
T1_V0_02	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	
T5_V0_01					X
T5_V0_02					X
T5_V0_03					X
T5_V0_04					X
T5_V0_05					X
T5_V0_06					X
T5_V0_07					X
T5_V0_08					X
T5_V0_09					X

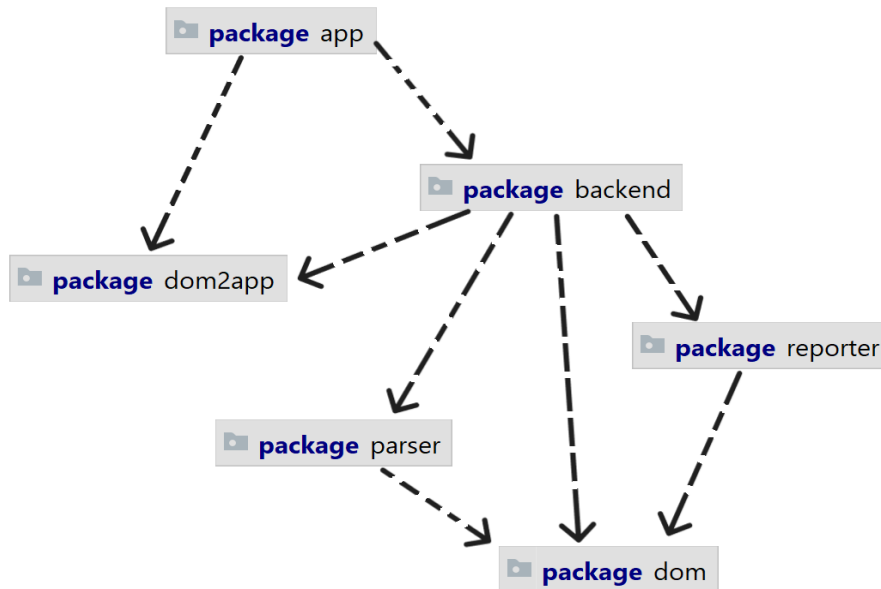
Πίνακας 2 Traceability matrix between use cases and tests



### 3 ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

#### 3.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ

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



Σχήμα 1. Διάγραμμα πακέτων

Ακολουθεί μια συνοπτική περιγραφή των πακέτων του συστήματος.

#### ΠΑΚΕΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

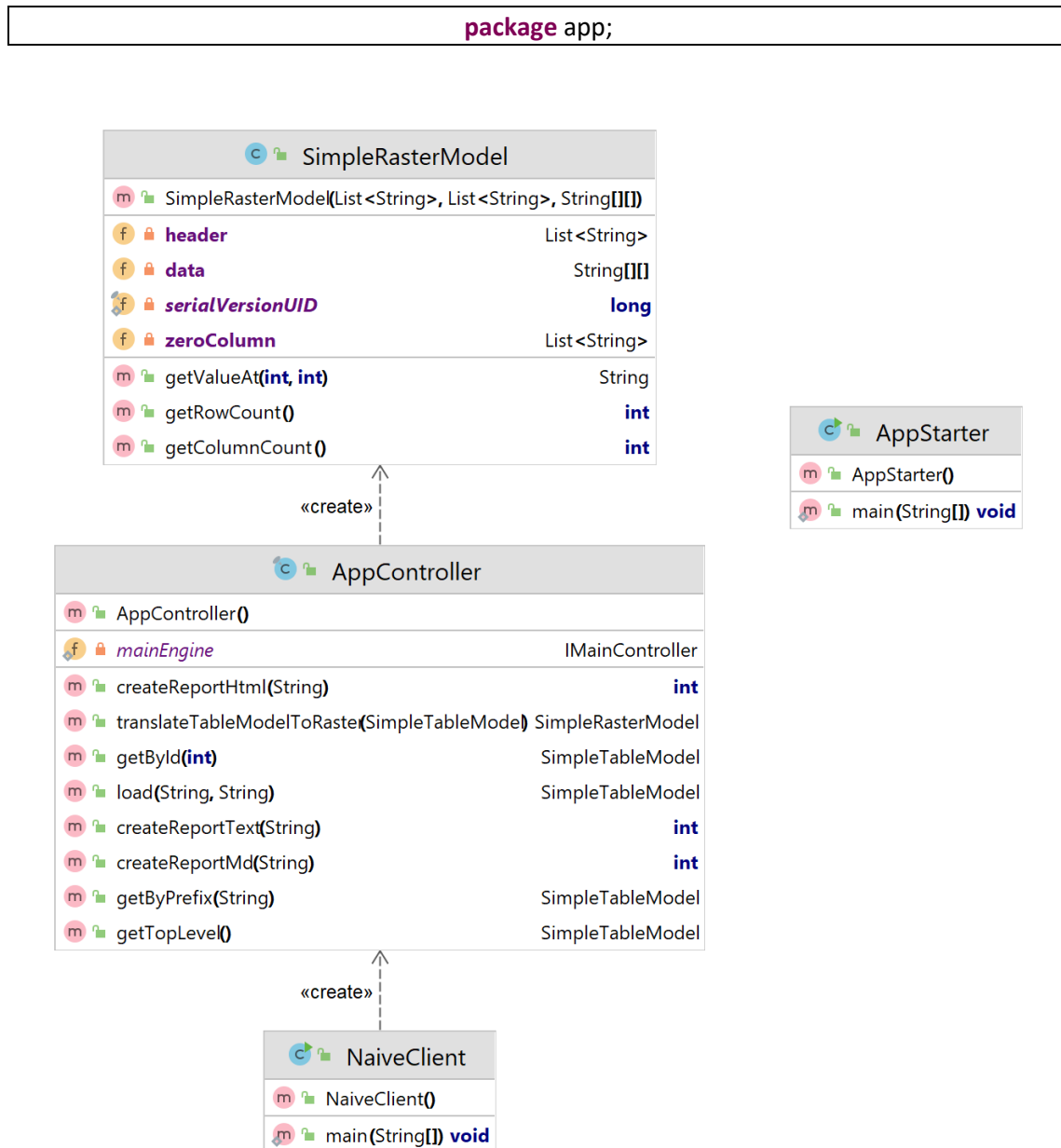
app	Περιέχει boundary classes που είναι υπεύθυνες για την αλληλεπίδραση με το χρήστη, την εκτέλεση της εφαρμογής και του gui.
dom2app	Η γέφυρα μεταξύ domain και gui.
backend	Υπεύθυνο για την διεκπεραίωση των use cases που προκύπτουν από την αρχική περιγραφή.
parser	Υποσύστημα αλληλεπίδρασης με τα αρχεία δεδομένων, για την ανάκτησή τους από το σύστημα.
dom	Οι domain classes.
reporter	Υποσύστημα παραγωγής αναφορών.

Πίνακας 3. Συνοπτική περιγραφή πακέτων συστήματος

### 3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

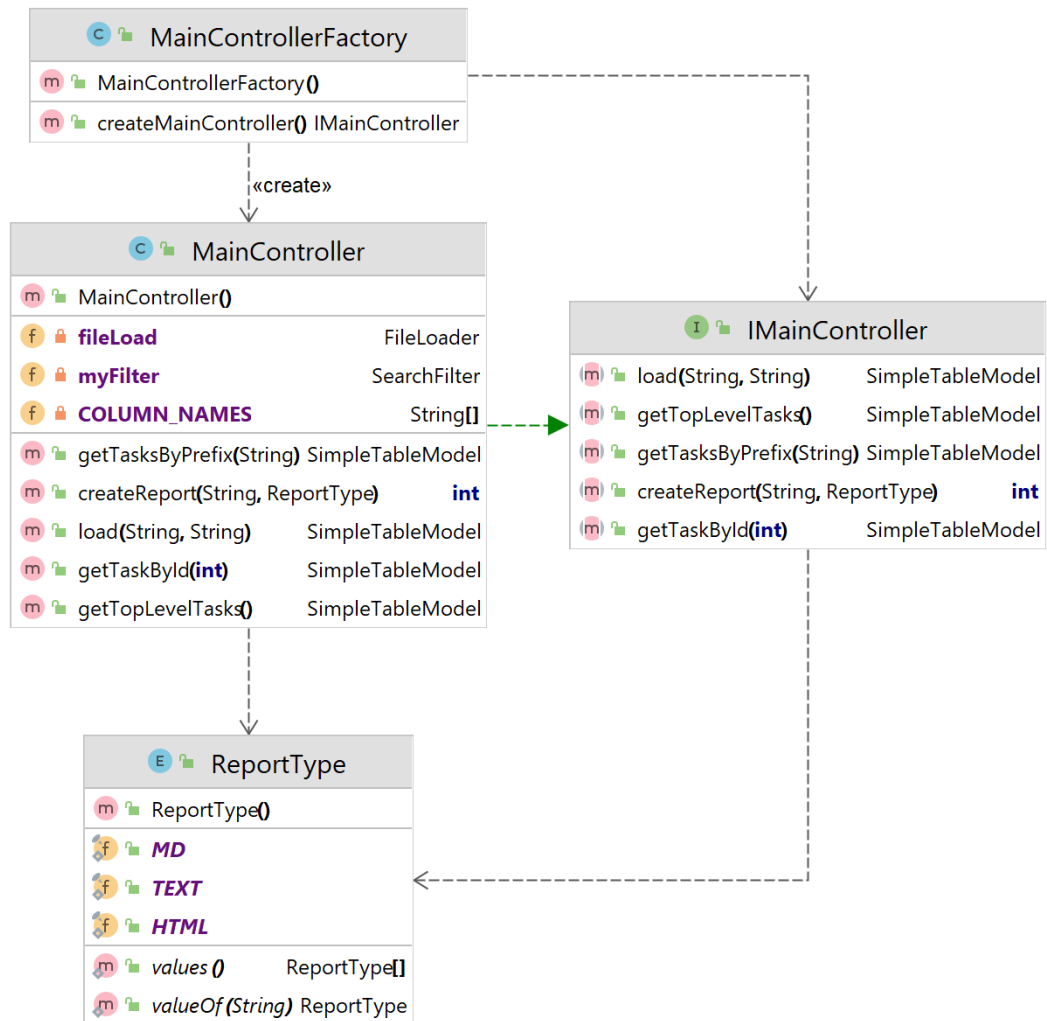
Στην παρούσα υποενότητα, παρατίθενται τα διαγράμματα κλάσεων.

Για τη σχεδίασή τους χρησιμοποιήσαμε το περιβάλλον του IntelliJ IDEA επειδή το ObjectAid του Eclipse είχε κάποιο πρόβλημα και δεν λειτουργούσε.



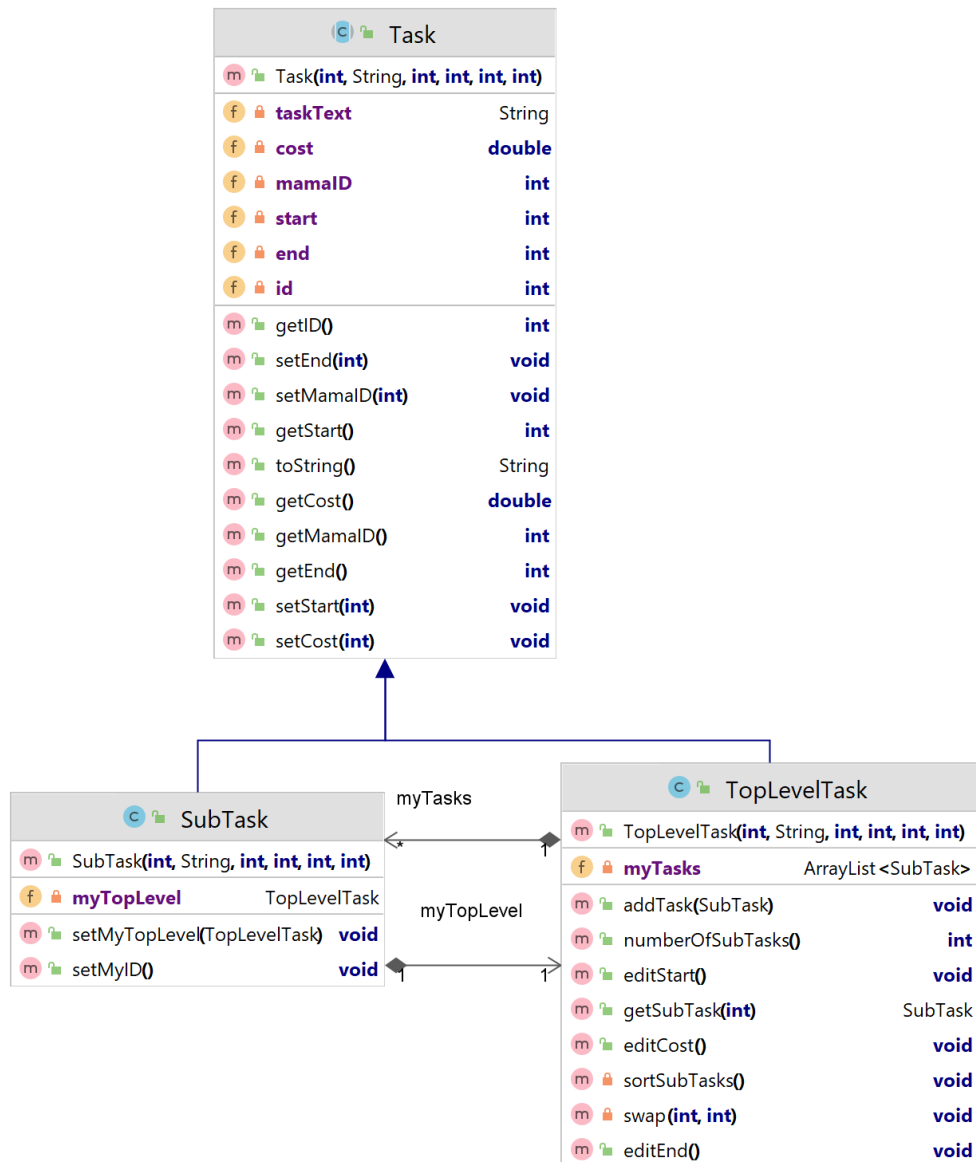
Σχήμα 2. Διάγραμμα κλάσεων για το πακέτο app

```
package backend;
```



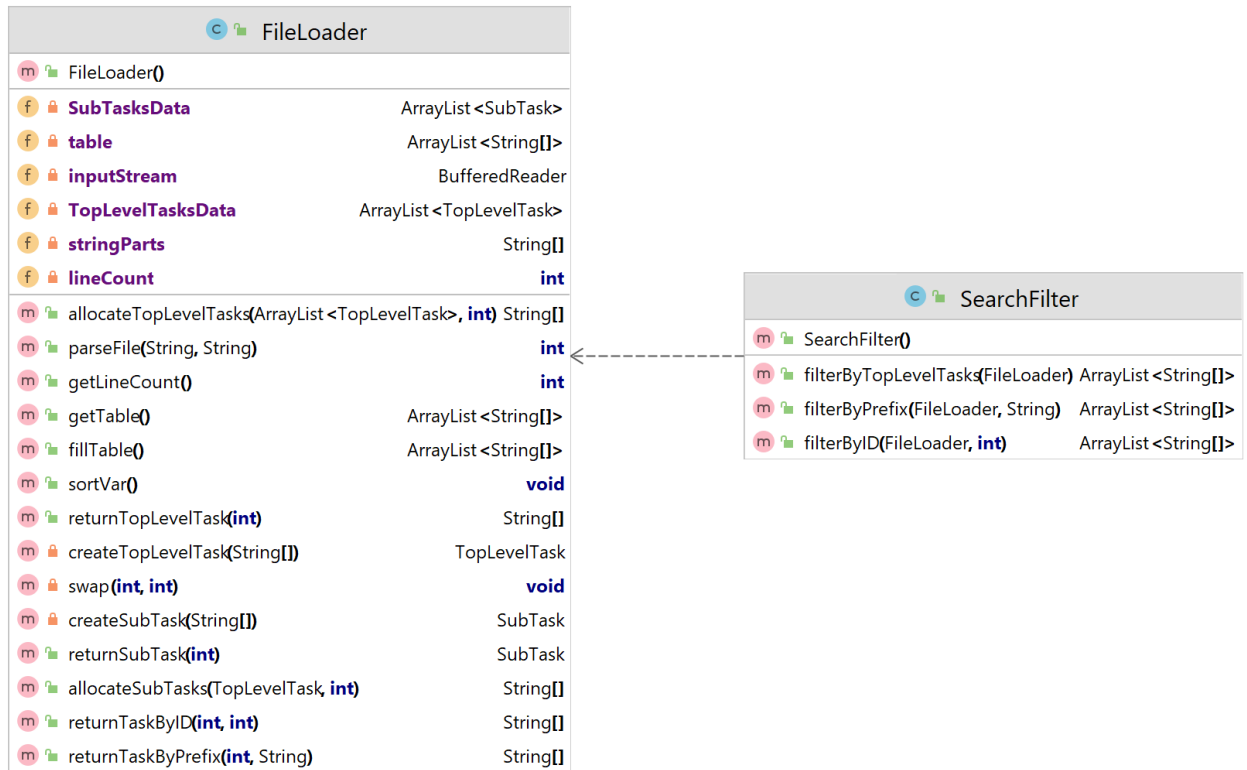
Σχήμα 3. Διάγραμμα κλάσεων για το πακέτο backend

package dom;



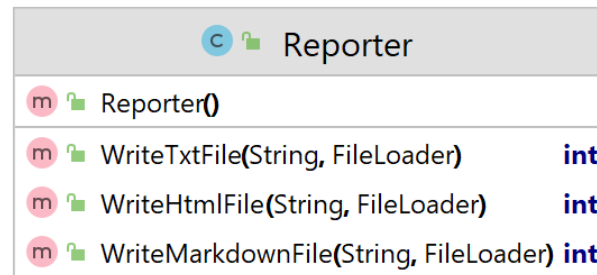
Σχήμα 4. Διάγραμμα κλάσεων για το πακέτο dom

```
package parser;
```



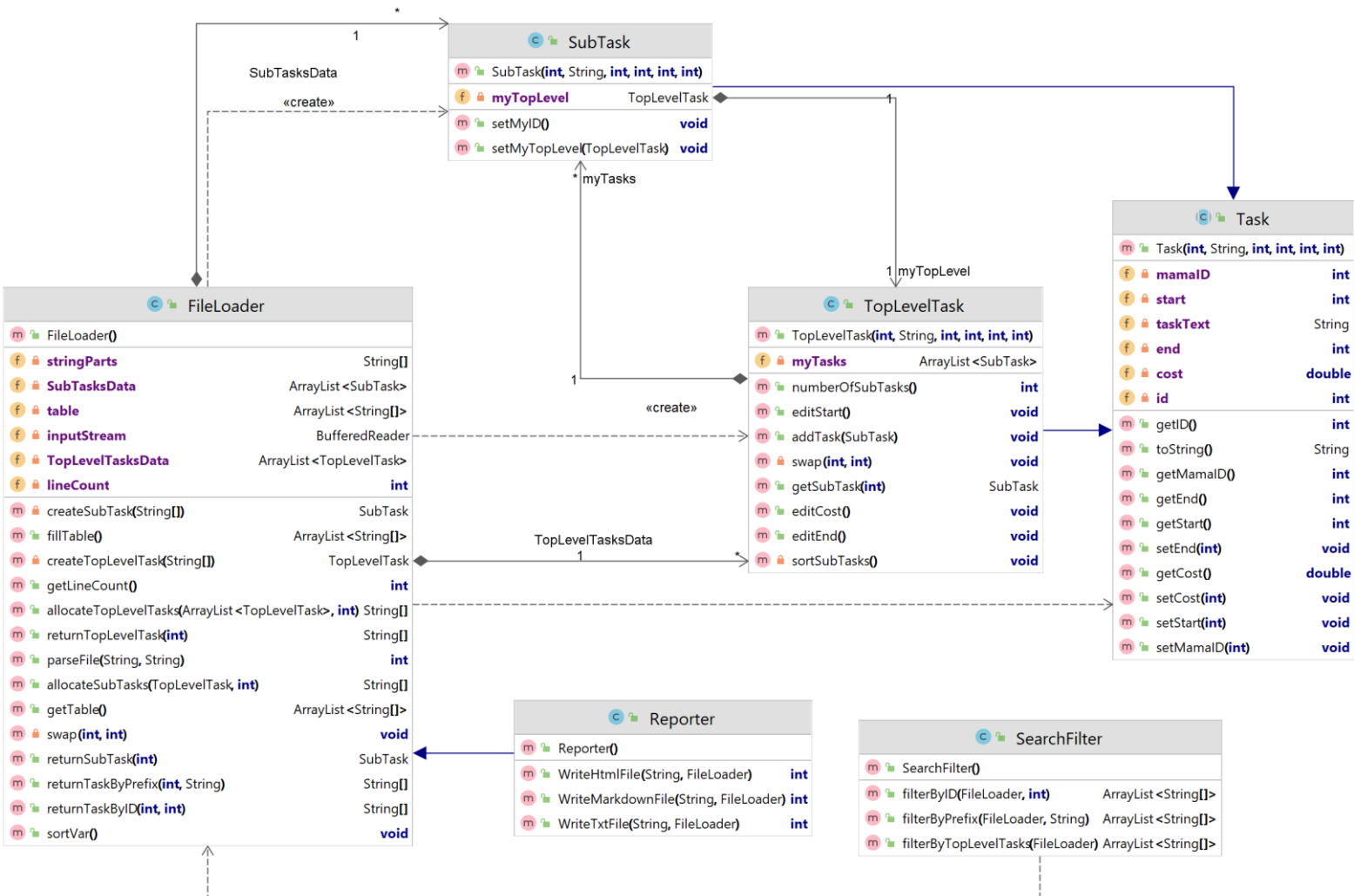
Σχήμα 5. Διάγραμμα κλάσεων για το πακέτο parser

**package reporter;**



Σχήμα 6. Διάγραμμα κλάσεων για το πακέτο reporter

### Συνεργαζόμενες Κλάσεις



Σχήμα 7. Διάγραμμα συνεργαζόμενων κλάσεων

### 3.3 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 .

#### 3.3.1 DOMAIN CLASSES

Package dom SubTask, TopLevelTask, μια αφηρημένη κλάση Task(A) για αυτές τις δύο

#### 3.3.2 BUSINESS LOGIC CLASSES

Package backend, parser, reporter MainController, η οποία κάνει implement το ImainController για την υλοποίηση όλων των use cases στο back-end και MainControllerFactory για τη δημιουργία του MainController  
FileLoader, SearchFilter, Reporter για την ανάκτηση του αρχείου, την υλοποίηση των φίλτρων και την παραγωγή αναφορών

#### 3.3.3 BOUNDARY CLASSES

Package app, dom2app SimpleTableModel, JTableViewer για την υλοποίηση του gui και την δημιουργία των διαγραμμάτων στην εφαρμογή  
AppController, AppStarter για την αλληλεπίδραση με το χρήστη και την εκκίνηση της εφαρμογής

#### 3.3.4 ΑΠΕΙΚΟΝΙΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΕ ΜΕΘΟΔΟΥΣ

Υπάρχουν 6 use cases: φόρτωση αρχείου, ανάκτηση εργασιών κορυφαίου επιπέδου, ανάκτηση εργασιών με id, ανάκτηση εργασιών με prefix, αποθήκευση αναφορά και έξοδος.

#### ΑΠΕΙΚΟΝΙΣΗ USE CASES ΣΕ ΜΕΘΟΔΟΥΣ

Use case	Back-end methods	Front-end methods
Φόρτωση αρχείου	FileLoader.parseFile()	mainEngine.load()
Ανάκτηση εργασιών κορυφαίου επιπέδου	searchFilter.filterByTopLevelTasks()	mainEngine.getTopLevelTasks()
Ανάκτηση εργασιών με id	searchFilter.filterById()	mainEngine.getTaskById()
Ανάκτηση εργασιών με prefix	searchFilter.filterByPrefix()	mainEngine.getTasksByPrefix()
Αποθήκευση αναφορά	Report.WriteTxtFile() Report.WriteHtmlFile() Report.WriteMarkdownFile()	mainEngine.createReport()

Πίνακας 4 Επαλήθευση απεικόνισης use cases σε μεθόδους