
ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2022-2023

ΟΜΑΔΑ 5339-5355-5412

ΑΙΚΑΤΕΡΙΝΑ ΡΑΛΛΗ, ΑΜ:5339

ΑΘΗΝΑ ΣΤΑΣΙΝΟΥ, ΑΜ:5355

ΔΗΜΗΤΡΙΟΣ-ΠΑΝΑΓΙΩΤΗΣ
ΣΙΑΜΑΛΕΚΑΣ, ΑΜ:5412

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2022

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφείς
2022/11/15	v.01	(a) use cases + (b) the OREOS part for the test cases + (c) a first version of the class diagram(s) with the design of the system.	Αικατερίνα Ράλλη, Αθηνά Στασινού, Δημήτριος- Παναγιώτης Σιαμαλέκας
2022/12/11	v.02	ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ	Αικατερίνα Ράλλη, Αθηνά Στασινού, Δημήτριος- Παναγιώτης Σιαμαλέκας

1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USE CASES

ΕΜΦΑΝΙΣΕ ΠΡΟΪΟΝΤΑ ΚΑΤΑΣΤΗΜΑΤΟΣ

ID: UC 1

DESCRIPTION AND GOAL

Φορτώνει από απλό delimited κείμενο το οποίο περιέχει tasks τα οποία θα ταξινομηθούν και θα χρησιμοποιηθούν για να δημιουργηθεί ένα Gantt Diagram.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Πρέπει έχει πάρει όρισμα ένα valid αρχείο.

BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από μενού "File" το "Load TSV".
2. Αφού επιλέξει ένα valid αρχείο.
3. Το σύστημα φορτώνει το αρχείο.

EXTENSIONS / VARIATIONS

Στην περίπτωση κατά την οποία ο χρήστης δεν έχει δώσει σωστό τύπο αρχείου, εμφανίζεται το μήνυμα "FILE NOT FOUND" στο terminal.

POST CONDITIONS

Το σύστημα εμφανίζει το διάγραμμα Gantt στην οθόνη και τα Tasks ταξινομημένα σε μορφή πίνακα.

ΑΝΑΚΤΗΣΗ ΤΩΝ ΕΡΓΑΣΙΩΝ ΚΟΡΥΦΑΙΟΥ ΕΠΙΠΕΔΟΥ

ID: UC 2

DESCRIPTION AND GOAL

Παρουσιάζει τα Tasks κορυφαίου επιπέδου μαζί με τα στοιχεία που τα χαρακτηρίζουν.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Πρέπει να έχουν φορτωθεί τα Tasks και να έχει γίνει ο διαχωρισμός ανάμεσα σε top level εργασίες και subtask εργασίες.

BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού "Filters" το "Top Level Tasks".
2. Ανακτώνται οι εργασίες κορυφαίου επιπέδου.
3. Εμφανίζονται τα top level tasks σε μορφή Gantt Diagram στην οθόνη του χρήστη.

EXTENSIONS / VARIATIONS

Στην περίπτωση στην οποία δεν υπάρχουν top level tasks , εμφανίζεται το μήνυμα "Top level task not found" στο terminal.

POST CONDITIONS

Εμφανίζεται το διάγραμμα Gantt στην οθόνη και τα Tasks κορυφαίου επιπέδου σε μορφή πίνακα.

ΑΝΑΚΤΗΣΗ ΟΛΩΝ ΤΩΝ ΕΡΓΑΣΙΩΝ ΠΟΥ ΠΛΗΡΟΥΝ ΤΟ ΙΔΙΟ ΚΡΙΤΗΡΙΟ ΠΕΡΙΓΡΑΦΗΣ.

ID: UC 3

DESCRIPTION AND GOAL

Ανακτά ταξινομημένες μόνο τις εργασίες που η περιγραφή τους ξεκινά με τα γράμματα που δόθηκαν από τον χρήστη.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Να υπάρχουν φορτωμένα Tasks.

BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού "Filters" το "Filter by Prefix".
2. Ο χρήστης πληκτρολογεί την σειρά γραμμάτων με την οποία θέλει να ξεκινάει η περιγραφή του Task.
3. Φορτώνεται ο πίνακας με τα ζητούμενα Tasks.
4. Εμφανίζονται τα ζητούμενα Tasks σε μορφή Table στην οθόνη του χρήστη.

EXTENSIONS / VARIATIONS

Στην περίπτωση στην οποία δεν υπάρχουν tasks με το κριτήριο περιγραφής που ζητήθηκε, εμφανίζεται το μήνυμα "Prefix tasks not found" στο terminal.

POST CONDITIONS

Εμφανίζεται το Table στην οθόνη με τα Tasks που ταιριάζουν με την περιγραφή του χρήστη.

ΑΝΑΚΤΗΣΗ ΕΡΓΑΣΙΑΣ ΜΕ ΣΥΓΚΕΚΡΙΜΕΝΟ TaskId

ID: UC 4

DESCRIPTION AND GOAL

Ανακτά ταξινομημένες μόνο τις εργασίες που το TaskId τους ταιριάζει με αυτό που έχει δωθεί από τον χρήστη.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Να υπάρχουν φορτωμένα Tasks.

BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού "Filters" το "Filter by Id".
2. Ο χρήστης πληκτρολογεί τον αριθμό με τον οποίο θέλει να ταιριάζει το TaskId του Task που αναζητά.
3. Φορτώνεται ο πίνακας με τα ζητούμενα Tasks.
4. Εμφανίζονται το ζητούμενο Task σε μορφή table στην οθόνη του χρήστη.

EXTENSIONS / VARIATIONS

Στην περίπτωση στην οποία δεν υπάρχουν tasks με το Id που ζητήθηκε, εμφανίζεται το μήνυμα "Id not found" στο terminal.

POST CONDITIONS

Εμφανίζεται το table στην οθόνη και το Task που ταιριάζει με το ζητούμενο TaskId.

ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΑΠΛΟ ΚΕΙΜΕΝΟ, HTML ΚΑΙ MARKDOWN

ID: UC 5

DESCRIPTION AND GOAL

Αποθηκεύει μια αναφορά με τρεις εναλλακτικούς τρόπους απλό κείμενο , html και markdown.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Να υπάρχουν φορτωμένα Tasks και να έχει δοθεί σωστός τύπος αποθήκευσης.

BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού "Report" μία από τις τρεις επιλογές report txt(απλό αρχείο κειμένου όπου οι στήλες χωρίζονται με tabs και οι γραμμές με αλλαγή γραμμής) , md(αρχείο κειμένου στο οποίο οι στήλες χωρίζονται με space, οι γραμμές με αλλαγή γραμμής, η γραμματοσειρά Italics με ένα αστεράκι (*) και η γραμματοσειρά bold με δύο αστεράκια (**) , html(αρχείο κειμένου το οποίο αποτελείται από μια σειρά στοιχείων τα οποία είναι ο σκελετός ενός web page, δηλαδή τύπος αρχείου που εμφανίζεται στον browser).
2. Αφού γίνει η επιλογή τύπου αρχείου, ο χρήστης επιλέγει ονομασία και θέση αποθήκευσης του.
3. Το αρχείο δημιουργείται με τα επιλεγμένα στοιχεία.

EXTENSIONS / VARIATIONS

-

POST CONDITIONS

Δημιουργείται ένα αρχείο στη μορφή που επέλεξε ο χρήστης.

ΕΞΟΔΟΣ

ID: UC 6

DESCRIPTION AND GOAL

Γίνεται έξοδος από το πρόγραμμα.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Να έχει πατηθεί από τον χρήστη το κουμπί της εξόδου

BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού "Files" το "Exit".
2. Εμφανίζεται ένα μήνυμα ερώτησης για τον αν θέλει να κλείσει το πρόγραμμα
3. Αν η απάντηση είναι θετική τότε το πρόγραμμα κλείνει, αν είναι αρνητική τότε παραμένει στο πρόγραμμα.

EXTENSIONS / VARIATIONS

-

POST CONDITIONS

-

2 ΣΧΕΔΙΑΣΗ ΕΛΕΓΧΩΝ

2.1 ΕΛΕΓΧΟΣ USE CASES VIA SYSTEM TESTS

2.1.1 USE CASE UC1: LOAD FROM TEXT

Test cases

<i>Description</i>	<i>ON</i>	<i>Load a specific txt delimited file with tasks</i>
	<i>RECEIVING</i>	<i>A txt delimited file</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>A Gantt diagram and a table with sorted tasks</i>
	<i>SUCH THAT</i>	<i>State is intact</i>

ID	T1_V0_01	HappyDayScenario for TaskService.dom()
Pre-cond.		A valid input file must be loaded
Input		input_Shop.tsv, a small file with less than 10 entries, all valid
Output		a board with the same entries of input_Shop.tsv sorted by Id and start date
Post-cond.		Creates a List of sorted tasks
Method To test		TaskService.load(String filename, String delimiter)

ID	T1_V0_02	HappyDayScenario for MainController.backend()
Pre-cond.		A valid input file must be loaded
Input		input_Shop.tsv, a small file with less than 10 entries, all valid
Output		a board with the same entries of input_Shop.tsv sorted by Id and start date
Post-cond.		Creates a SimpleTableModel object with the list of the sorted tasks
Method To test		MainController.load(String filename, String delimiter)

ID	T1_V0_03	RainyDayScenario for TaskService.dom()
Pre-cond.		A non valid input file must be loaded
Input		input_fake.tsv, a file that doesn't exist
Output		ERROR: FILE NOT FOUND
Post-cond.		Load method doesn't work
Method To test		TaskService.load(String filename, String delimiter)

Involved methods

```
FileParser.parse(filename,delimiter)
```

```
TaskService.Load(filename,delimiter)
```

```
TaskConverter.convert(String tableName,List<Task> tasks)
```

2.1.2 USE CASE UC2: GET TOP LEVEL TASKS

Test cases

<i>Description</i>	<i>ON</i>	<i>All tasks must be loaded and sorted to simple and complex tasks</i>
	<i>RECEIVING</i>	<i>A list of tasks</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>A Gantt diagram only with top level tasks and a board with top level sorted tasks</i>
	<i>SUCH THAT</i>	<i>Display Gantt diagram on the screen</i>

ID	T2_V0_01	HappyDayScenario for TaskService.dom()
Pre-cond.		A valid input file must be loaded
Input		input_Shop.tsv, a small file with less than 10 entries, all valid
Output		A list of topLevelTasks
Post-cond.		Creates a list of topLevelTasks
Method To test		TaskService.getTopLevelTasks()

ID	T2_V0_02	HappyDayScenario for MainController.backend()
Pre-cond.		A valid input file must be loaded
Input		input_Shop.tsv, a small file with less than 10 entries, all valid
Output		Creates a SimpleTableModel object with sorted top level tasks only
Post-cond.		Creates a SimpleTableModel object with sorted top level tasks only
Method To test		MainController.getTopLevelTasks()

Involved methods

`Task.getMamaId()`

`TaskService.getTopLevelTasks()`

`TaskConverter.convert(String tableName, List<Task> tasks)`

2.1.3 USE CASE UC3: GET TASKS BY PREFIX

Test cases

<i>Description</i>	<i>ON</i>	<i>All tasks must be loaded</i>
	<i>RECEIVING</i>	<i>A specific string</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>Tasks that their description starts with the input string</i>
	<i>SUCH THAT</i>	<i>A table with the chosen tasks</i>

ID	T3_V0_01	HappyDayScenario for TaskService.dom()
Pre-cond.		A valid input file must be loaded
Input		A string
Output		A List of tasks that their description starts with the input String
Post-cond.		Creates a list of prefixTasks
Method To test		TaskService.getTasksByPrefix(String)

ID	T3_V0_02	HappyDayScenario for MainController.backend()
Pre-cond.		A valid input file must be loaded
Input		A string
Output		Creates a SimpleTableModel object with sorted prefixTasks
Post-cond.		Creates a SimpleTableModel object with sorted prefixTasks
Method To test		mainController.getTasksByPrefix(String)

ID	T3_V0_03	RainyDayScenario for TaskService.dom()
Pre-cond.		A non-valid input file must be loaded
Input		Input a String
Output		The String "Prefix Tasks not found" in terminal
Post-cond.		There aren't tasks that their tasksText starts with the input String
Method To test		TaskService.getTasksByPrefix(String prefix)

Involved methods

Task.getTaskText()

String.startsWith(String)

TaskService.getTasksByPrefix(String)

TaskConverter.convert(String tableName,List<Task> tasks)

2.1.4 USE CASE UC4: GET TASKS BY TASKID

Test cases

<i>Description</i>	<i>ON</i>	<i>All tasks must be loaded</i>
	<i>RECEIVING</i>	<i>Specific number</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>The tasks that their taskId matches the inout number</i>
	<i>SUCH THAT</i>	<i>A table of the chosen Tasks</i>

ID	T4_V0_01	HappyDayScenario for TaskService.dom()
Pre-cond.		A valid input file must be loaded
Input		A number
Output		A task with the chosen Id
Post-cond.		The task with the chosen Id
Method To test		TaskService.getTasksById(int)

ID	T4_V0_02	HappyDayScenario for MainController.backend()
Pre-cond.		A valid input file must be loaded
Input		A number
Output		Creates a SimpleTableModel object with the chosen id task
Post-cond.		Creates a SimpleTableModel object with the chosen id task
Method To test		mainController.getTasksById(int)

ID	T4_V0_03	RainyDayScenario for TaskService.dom()
Pre-cond.		A non-valid input file must be loaded
Input		A number
Output		The String "Id not found" in terminal
Post-cond.		There isn't a task with this Id
Method To test		TaskService.getTasksById(int)

Involved methods

Task.getTaskId()

TaskService.getTaskById(int)

TaskConverter.convert(String tableName, List<Task> tasks)

2.1.5 USE CASE UC5: CREATE REPORT BASED ON REPORT TYPES

Test cases

<i>Description</i>	<i>ON</i>	<i>All tasks must be loaded</i>
	<i>RECEIVING</i>	<i>Specific report type</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>Creates a report in the chosen type</i>
	<i>SUCH THAT</i>	<i>Saves the report in the chosen type</i>

ID	T5_V0_01	HappyDayScenario for TextReporter.reporter()
Pre-cond.		A valid input file must be loaded
Input		A list of tasks
Output		String with the content of the file
Post-cond.		The content that is going to be saved in a txt file
Method To test		TextReporter.prepareData(List<Task>)

ID	T5_V0_02	HappyDayScenario for HtmlReporter.reporter()
Pre-cond.		A valid input file must be loaded
Input		A list of tasks
Output		String with the content of the file
Post-cond.		The content that is going to be saved in a html file
Method To test		HtmlReporter.prepareData(List<Task>)

ID	T5_V0_03	HappyDayScenario for MarkdownReporter.reporter()
Pre-cond.		A valid input file must be loaded
Input		A list of tasks
Output		String with the content of the file
Post-cond.		The content that is going to be saved in a md file
Method To test		MarkDownReporter.prepareData(List<Task>)

Involved methods

Reporter.prepareData(List<Task>)

String.getBytes()

Files.write()

Paths.get()

2.1.6 USE CASE UC4: EXIT

Test cases

<i>Description</i>	<i>ON</i>	<i>The "EXIT" button must be pressed</i>
	<i>RECEIVING</i>	<i>"YES" OR "NO"</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>The program exits</i>
	<i>SUCH THAT</i>	<i>The program close</i>

2.2 TRACEABILITY MATRIX

UC1	Load Data
UC2	Get top Level Tasks
UC3	Get Tasks by Prefix
UC4	Get Task by Id
UC5	Create report Based on report types

Πίνακας 1 Σύνοψη use cases και των id's τους

	UC1	UC2	UC3	UC4	UC5
T1_V0_01	X				
T1_V0_02	X				
T1_V0_03	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
T6_V0_03					X

Πίνακας 2 Traceability matrix between use cases and tests

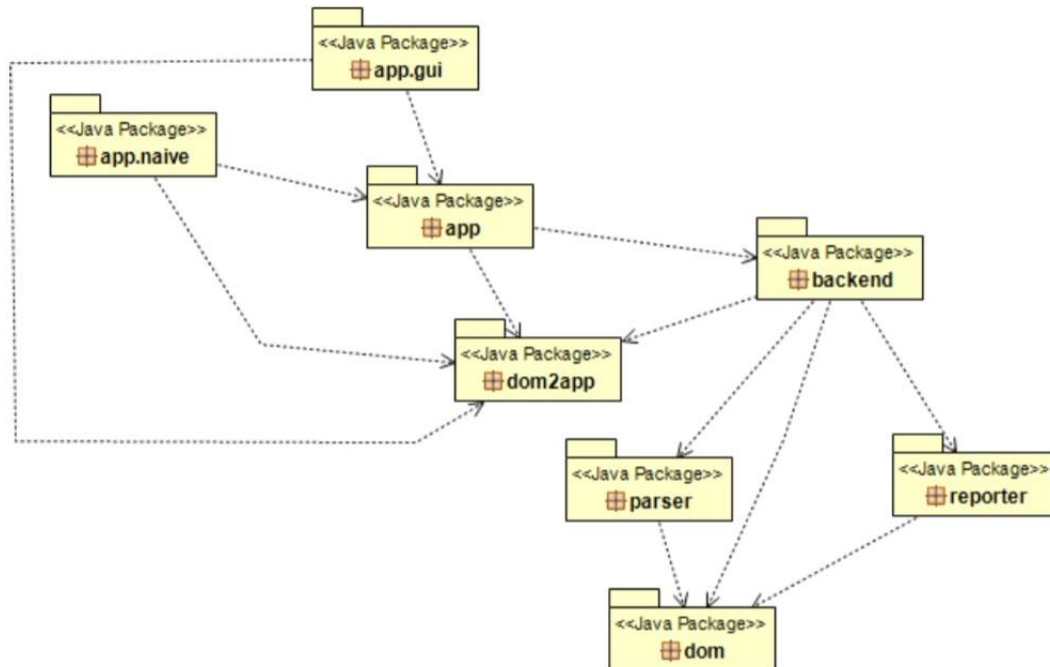
2.3 ΕΚΚΡΕΜΟΤΗΤΕΣ (TODO)

-

3 ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

3.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ

Διάγραμμα των πακέτων του συστήματος.



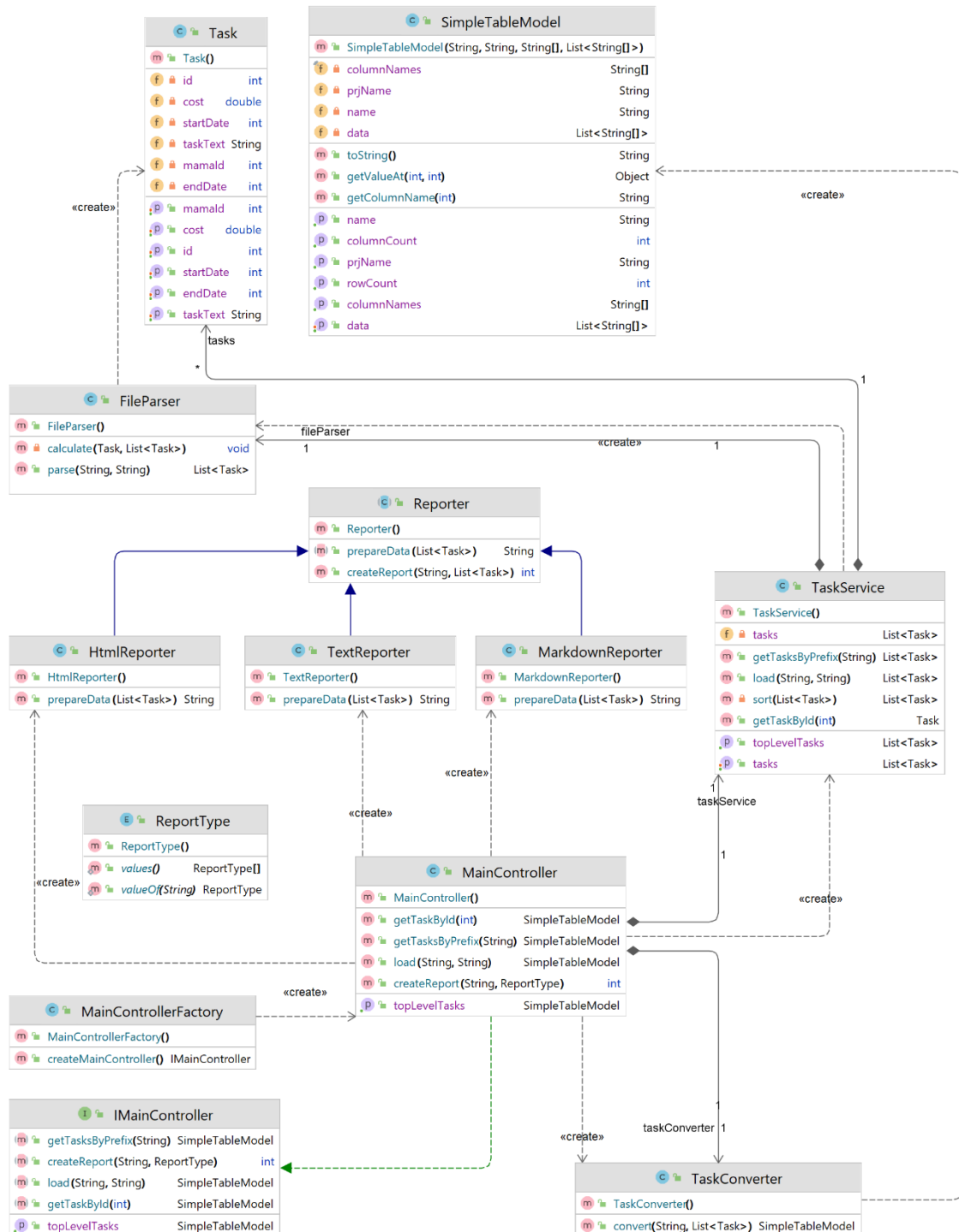
ΠΑΚΕΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

app	Περιέχει τις boundary classes που είναι υπεύθυνες για την αλληλεπίδραση με το χρήστη
backend	Κεντρική business logic engine
dom2app	Υποσύστημα που συνδέει τις domain classes με τις boundary classes
parser	Υποσύστημα φόρτωσης δεδομένων στο σύστημα
report	Υποσύστημα παραγωγής αναφορών
dom	Domain classes of the system

Πίνακας 3. Συνοπτική περιγραφή πακέτων συστήματος

3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

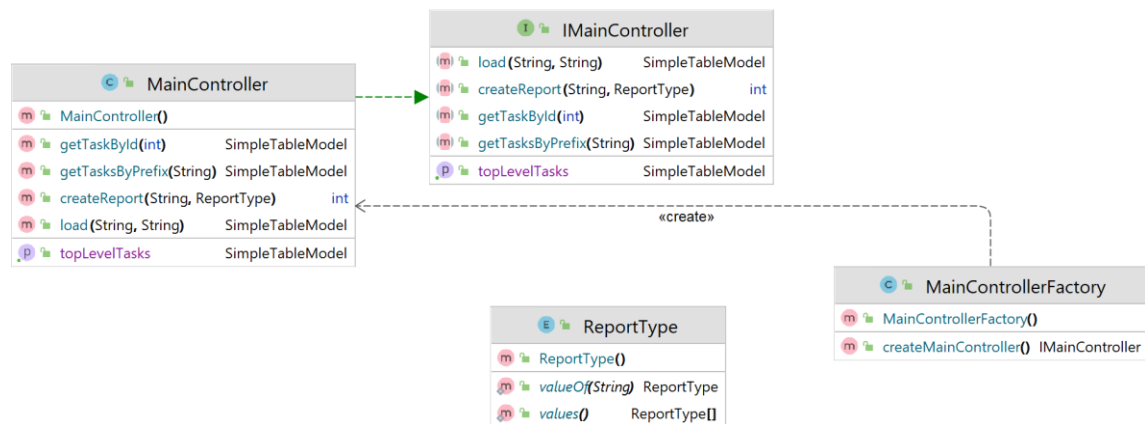
BACK-END



Σχήμα 1. Διάγραμμα κλάσεων για το back-end της εφαρμογής

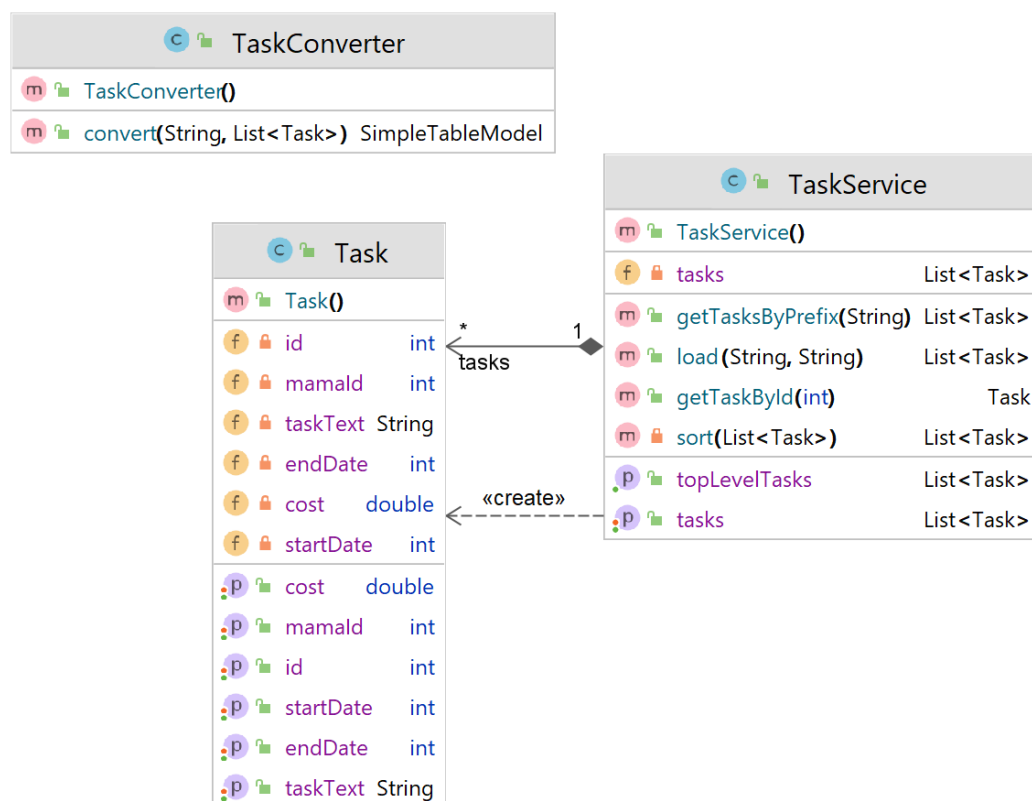
```
(packages: backend, dom, parser, reporter)
```


package backend;



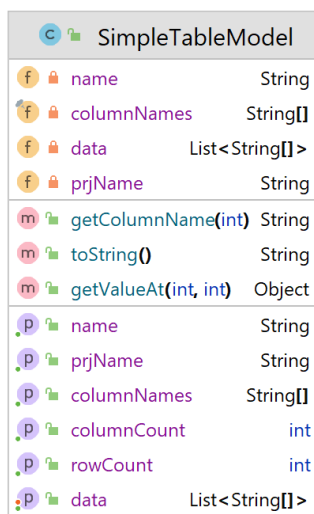
Σχήμα 1. Διάγραμμα κλάσεων για το πακέτο backend

package dom;



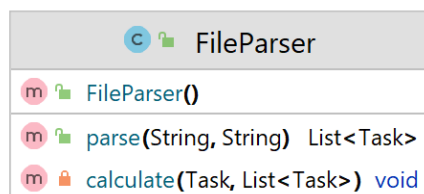
Σχήμα 2. Διάγραμμα κλάσεων για το πακέτο dom

```
package dom2app;
```



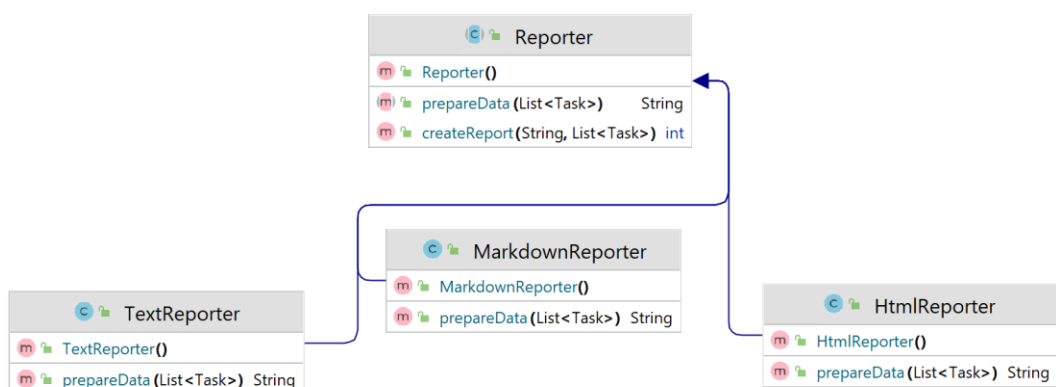
Σχήμα 4. Διάγραμμα κλάσεων για το πακέτο dom2app

```
package parser;
```



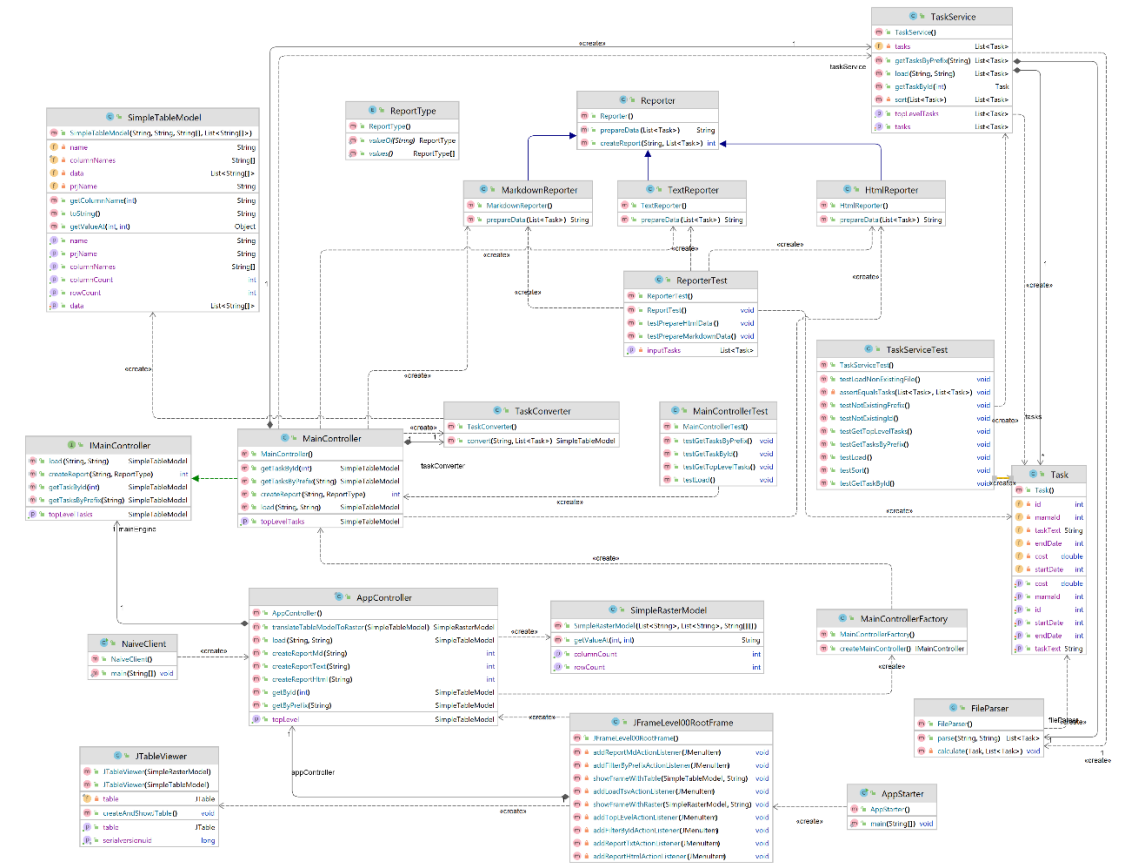
Σχήμα 5. Διάγραμμα κλάσεων για το πακέτο parser

```
package reporter;
```



Σχήμα 6. Διάγραμμα κλάσεων για το πακέτο reporter

CLASS DIAGRAM FOR THE GANTT DIAGRAM APP



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.

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 σχεδιάσής σας.

3.3.1 DOMAIN CLASSES

Package dom	<ul style="list-style-type: none"> Task, μια κλάση που τα αντικείμενα της είναι το δομικό στοιχείο του προγράμματος. Task Service, εκτελεί βασικές λειτουργίες με αντικείμενα Task. TaskConverter, δομή μετατροπής των Task.
-------------	---

3.3.2 BUSINESS LOGIC CLASSES

Package backend,	<ul style="list-style-type: none"> IMainController interface για την υλοποίηση της MainController. MainController για την υλοποίηση όλων των use cases στο back-end. MainControllerFactory για τη δημιουργία ενός MainController αντικειμένου. ReportType που περιέχει τις επιλογές για το report.
Package dom2app	<ul style="list-style-type: none"> SimpleTableModel, δομή αλληλεπίδρασης back-end και front-end.

3.3.3 BOUNDARY CLASSES

Package app	<ul style="list-style-type: none"> AppStarter, περιέχει την main method που εκκινεί το πρόγραμμα. AppController, δομή που εκτελεί βασικές λειτουργίες. NaiveClient, μια client class για την αλληλεπίδραση με το χρήστη. SimpleRasterModel, JFrameLevel100RootFrame, JTableModel, για το graphical user interface με το χρήστη (οι handler/controller classes χειρίζονται τα events από τη γραφική διαπροσωπεία).
-------------	---

3.3.4 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
Φόρτωση από απλό κείμενο	<code>MainController.Load(filename, delimiter)</code>	<code>AppController.Load(filename, delimiter)</code>
Ανάκτηση εργασιών κορυφαίου επιπέδου	<code>MainController.getTopLevelTasks()</code>	<code>AppController.getTopLevelTasks()</code>
Ανάκτηση των εργασιών που πληρούν το ίδιο κριτήριο περιγραφής	<code>MainController.getTasksByPrefix(String prefix)</code>	<code>AppController.getTasksByPrefix(String prefix)</code>
Ανάκτηση εργασίας με συγκεκριμένο TaskId	<code>MainController.getTaskById(int id)</code>	<code>AppController.getTaskById(int id)</code>
Αποθήκευση σε απλό κείμενο,html και markdown	<code>MainController.createReport(String path, Report type)</code>	<code>AppController.createReportText(String path)</code> <code>AppController.createReportHtml(String path)</code> <code>AppController.createReportMd(String path)</code>

Πίνακας 4 Επαλήθευση απεικόνισης use cases σε μεθόδους

4 ΛΟΙΠΑ ΣΧΟΛΙΑ

4.1 ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΠΟΦΑΣΕΙΣ

Στην εκφώνηση μας ζητείται use case exit αλλά κατά τη δημιουργία του κώδικα συμπεράναμε ότι υπάρχει ήδη στα δοθέντα αρχεία.

4.2 ΣΗΜΕΙΑ ΚΙΝΔΥΝΟΥ

Σημεία κινδύνου μπορούν να θεωρηθούν η εισαγωγή μη υπάρχοντος αρχείου και ανάκτηση εργασίας με μη υπάρχον Id ή prefix.

