ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2022-2023

ΟΜΑΔΑ 5212-5387-5388

ΓΕΩΡΓΙΟΣ ΔΗΜΟΥΔΗΣ, ΑΜ:5212

ΧΑΡΙΛΑΟΣ ΧΑΤΖΗΔΗΜΗΤΡΙΟΥ, ΑΜ:5387

ΟΜΗΡΟΣ ΧΑΤΖΗΙΟΡΔΑΝΗΣ, ΑΜ:5388

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2022

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφείς
2022/11/15	v.01	Οργάνωση απαιτήσεων σε use cases, πρώτη σχεδίαση test cases και δημιουργία του πρώτου class diagram	5212,5387,5388
2022/12/13	v.02	ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ	5212,5387,5388

1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USE CASES

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

LOAD

Use case 1:

Description and Goal:

The system loads to the GUI all the tasks from a text file that is given by the client

Actor:

The client

Preconditions:

There must be a text in a valid path, in a string format. The user must initiate this process when he starts the program

Flow:

1. The system takes the path in the arguments
2. While the file has a next line
    a. The system reads the line of the file
        i. The system takes each word between the tabs
        ii. The system adds them into a string array list
        iii. The system makes a new task object with the elements of the array
        iv. The system stores the task into its task list
    b. The system goes to the next line
3. The system closes the file

4. The system sorts its task list by Id at first and between the subtasks of one top-level task by the start point
5. The system returns the task list that it created in the previous steps

Extensions/Variations:

2a-iv. If a task is a sub-task of another one, we call the method addSubTask(subTask) to the mama-task

Post Conditions:

The system has a List of tasks to process, and the client has a graph of all the tasks.

Special Requirements, Issues, Risks and Other Comments:

The system might encounter errors if the file or the path has problems or they're not valid

GETTOPLEVELTASKS

Use case 2:

Description and Goal:

The system takes all the top-level tasks from the task list and returns them to the GUI.

Actor:

The Client.

Preconditions:

- The system must have a list of tasks that the system has created from UC1.
- The client must initiate this process through the GUI.

Basic Flow:

1. The system creates a temporary list for the return.
2. The system runs through the whole task list.
   a. If a task has a MamaId equal to 0
      i. The system adds the task to the return list
3. The system returns the return list.

Extensions / Variations:

Post Conditions:

The system has a list of top-level tasks that it presents to the client through the GUI.

Special Requirements, Issues, Risks and Other Comments:

UC1 must have been initialized.

GETTASKSBYPREFIX

Use case 3:

Description and Goal:
The system returns to the client tasks that match the prefix that the client gives

Actors:

The client

Preconditions:

The client must give a string and he must initiate the process from the corresponding button.

Flow:

1. The system reads the input of the client
2. The system creates a return task list
3. The system goes through the task list
    a. The system accesses the name of the current task
    b. If the name contains the input of the user in the beginning
        i. The system inserts the task into the return task list
4. The system returns the return task list

Extensions/Variations:

Post Conditions:

The system presents to the client a list of tasks that match his input

Special Requirements, Issues, Risks and Other Comments:

UC1 must have been initialized.

GETTASKBYID

Use case 4:

Description and Goal:
The system returns to the client task that match the taskId that the client gives

Actors:

The client

Preconditions:

The client must give a string and he must initiate the process from the corresponding button.

Flow:

1. The system reads the input of the client
2. The system goes through the task list
    a. The system accesses the taskId of the current task
    b. If the taskId matches the input of the user
        i. The system returns the task

Extensions/Variations:

Post Conditions:

The system presents to the client a task that matches his input

Special Requirements, Issues, Risks and Other Comments:

UC1 must have been initialized.

CREATEREPORT

Use case 5:

Description and Goal:
The system creates a simple text, html and or markdown file depending on what the client chooses

Actors:

The client

Preconditions:

The client must initiate the process from the corresponding button and he must have given the appropriate file for the tasks.

Flow:

1. The system reads the input of the client for his choice
2. The system creates the file type that the client chose with all the tasks and their information in it.

Extensions/Variations:

Post Conditions:

The system has created a simple text, html and or markdown file with the report of the tasks for the client

Special Requirements, Issues, Risks and Other Comments:

The client must specify the specific location that he wants the system to save the file

EXIT

Use case 6:

Description and Goal:
The system shuts down after the client requests it

Actors:

The client

Preconditions:

The client must initiate the process from the corresponding button.

Flow:

1. The system asks the client if he is sure for the procedure
2. If the input from the GUI is yes
    a. The system shuts down

Extensions/Variations:

2b. if the input is no, the system continues to operate

Post Conditions:

The system is shut down

Special Requirements, Issues, Risks and Other Comments:

2 ΣΧΕΔΙΑΣΗ ΕΛΕΓΧΩΝ

Οι έλεγχοι που σχεδιάσθηκαν 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

2.1.1 USE CASE UC1: LOAD

Test cases

Description	*ON*	*Any context*
	RECEIVING	*Request to parse a specific txt file *with a valid delimiter**
	ENSURE	*That the System*
	OUTPUTS	*A task list with the correct size and no offending task value pairs*
	SUCH THAT	*state is intact*

ID	testLoadHappy	HappyDayScenario for TaskManager.load()
Pre-cond.		A TaskManager object with a file path and a valid delimiter
Input		Eggs.tsv, a small file with 14 entries, all valid
Output		a SimpleTableModel with the same #entries as the contexts of Eggs.tsv , sorted and no offending values
Post-cond.		No state properties tested
Method To test		FileManager.giveTasks(), TaskManager.load()

ID	testReadFileHappy	HappyDayScenario for FileManager.giveTasks()
Pre-cond.		A FileManager object with a file path and a valid delimiter
Input		Eggs.tsv, a small file with less than 14 entries, all valid
Output		a task list with the same #entries as the contexts of Eggs.tsv, sorted and no offending values
Post-cond.		No state properties tested
Method To test		FileManager.giveTasks()

ID	testLoadRainy	RainyDayScenario for TaskManager.load()
Pre-cond.		A TaskManager object with a file path and a non-valid delimiter
Input		DoesNotExist.tsv, a file that doesn't exist
Output		That the SimpleTableModel size is 0
Post-cond.		No state properties tested
Method To test		FileManager.giveTasks(), TaskManager.load()

ID	testReadFileRainy	RainyDayScenario for FileManager.giveTasks()
Pre-cond.		A FileManager object with a file path and a non-valid delimiter
Input		DoesNotExist.tsv, a file that doesn't exist
Output		That the task list isn't null and that its size is 0
Post-cond.		No state properties tested
Method To test		FileManager.giveTasks()

Involved methods

TaskManager.load()

FileManager.giveTasks()

2.1.2 USE CASE UC2: GET TOP LEVEL TASKS

Test cases

Description	*ON*	*A an already created task list*
	RECEIVING	*Request to output all top-level tasks*
	ENSURE	*That the System*
	OUTPUTS	*a list of top-level tasks*
	SUCH THAT	*state is intact*

ID	testGetTopLevelTasksHappy	HappyDayScenario for TaskManager.getTopLevelTasks()
Pre-cond.		A TaskManager object with a file path and a valid delimiter
Input		Eggs.tsv, a small file with less than 14 entries, all valid
Output		a SimpleTableModel with the same #entries as the top-level contexts of Eggs.tsv, sorted and no offending values
Post-cond.		No state properties tested
Method To test		TaskManager.getTopLevelTasks()

ID	testGetTopLevelTasksRainy	RainyDayScenario for TaskManager. getTopLevelTasks ()
Pre-cond.		A TaskManager object with a file path and a non-valid delimiter
Input		DoesNotExist.tsv, a file that doesn't exist
Output		That the SimpleTableModel size is 0
Post-cond.		No state properties tested
Method To test		TaskManager. getTopLevelTasks ()

Involved methods

TaskManager.load()

FileManager.giveTasks()

TaskManager.getTopLevelTasks()

2.1.3 USE CASE UC3: GET TASKS BY PREFIX

Test cases

Description	*ON*	*A an already created task list*
	RECEIVING	*Request to output all tasks that the first letters of their name match the input of the user*
	ENSURE	*That the System*
	OUTPUTS	*An appropriate task list that their names include the prefix in the beginning*
	SUCH THAT	*state is intact*

ID	testGetTasksByPrefixHappy	HappyDayScenario for TaskManager.getTasksByPrefix(String Prefix)
Pre-cond.		A TaskManager object with a file path and a valid delimiter
Input		Eggs.tsv, a small file with 14 entries, all valid and a valid prefix
Output		A correct sorted SimpleTableModel with the tasks that match the prefix
Post-cond.		No state properties tested
Method To test		TaskManager.getTasksByPrefix(String Prefix)

ID	testGetTasksByPrefixRainy	RainyDayScenario for TaskManager.getTasksByPrefix(String Prefix)
Pre-cond.		A TaskManager object with a file path and a valid delimiter
Input		Eggs.tsv, a small file with 14 entries, all valid and a non-valid prefix
Output		That the SimpleTableModel size is 0
Post-cond.		No state properties tested
Method To test		TaskManager.getTasksByPrefix(String Prefix)

Involved methods

TaskManager.load()

FileManager.giveTasks()

TaskManager.getTasksByPrefix(String Prefix)

2.1.4 USE CASE UC4: GET TASK BY ID

Test cases

Description	*ON*	*A an already created task list*
	RECEIVING	*Request to output the task that their taskId matches the input of the user*
	ENSURE	*That the System*
	OUTPUTS	*An appropriate task that its id matches the input*
	SUCH THAT	*state is intact*

ID	testGetTaskByIdHappy	HappyDayScenario for TaskManager.getTaskById(int id)
Pre-cond.		A TaskManager object with a file path and a valid delimiter
Input		Eggs.tsv, a small file with 14 entries, all valid
Output		A correct SimpleTableModel with the task that matches
Post-cond.		No state properties tested
Method To test		TaskManager. getTaskById(int id)

ID	testGetTaskByIdRainy	RainyDayScenario for TaskManager.getTaskById(int id)
Pre-cond.		A TaskManager object with a file path and a non-valid delimiter
Input		DoesNotExist.tsv, a file that doesn't exist
Output		That the SimpleTableModel size is 0
Post-cond.		No state properties tested
Method To test		TaskManager. getTaskById(int id)

Involved methods

TaskManager.load()

FileManager.giveTasks()

TaskManager.getTaskById()

2.1.5 USE CASE UC5: CREATE REPORT

Test cases

Description	*ON*	*A an already created task list*
	RECEIVING	*Request to create a file with all the tasks*
	ENSURE	*That the System*
	OUTPUTS	*An appropriate file with the matching format with the tasks that have been created*
	SUCH THAT	*state is intact*

ID	testCreateReportHappy	HappyDayScenario for TaskManager.createReport()
Pre-cond.		A TaskManager object with a file path and a valid delimiter
Input		Eggs.tsv, a small file with 14 entries, all valid
Output		A file with the tasks sorted
Post-cond.		No state properties tested
Method To test		TaskManager.createReport(),Reporter.makeReportTXT(), Reporter.makeReportHTML(),Reporter.makeReportMD()

ID	testCreateReportRainy	RainyDayScenario for TaskManager.createReport ()
Pre-cond.		A TaskManager object with a file path and a non-valid delimiter
Input		DoesNotExist.tsv, a file that doesn't exist
Output		A file including only the column names and elements that are needed for the file format
Post-cond.		No state properties tested
Method To test		TaskManager.createReport(),Reporter.makeReportTXT(), Reporter.makeReportHTML(),Reporter.makeReportMD()

Involved methods

TaskManager.load()

FileManager.giveTasks()

TaskManager.createReport()

Reporter.makeReportTXT()

Reporter.makeReportHTML()

Reporter.makeReportMD()

2.1.6 USE CASE UC6: EXIT

Test cases

Description	*ON*	*Anything*
	RECEIVING	*Request to exit the program*
	ENSURE	*That the System*
	OUTPUTS	*Exits*
	SUCH THAT	*state is intact*

We didn't make any test cases for the UC6 because it is handled by classes in the GUI and the professor said it isn't necessary

2.2 TRACEABILITY MATRIX

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

UC1	Load
UC2	Get Top Level Tasks
UC3	Get Tasks By Prefix
UC4	Get Task By Id
UC5	Create Report
UC6	Exit

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	UC6
testLoadHappy	X					
testReadFileHappy	X					
testGetTopLevelTasksHappy	X	X				
testGetTasksByPrefixHappy	X		X			
testGetTaskByIdHappy	X			X		
testCreateReportHappy	X				X	
testLoadHappyRainy	X					
testReadFileRainy	X					
testGetTopLevelTasksRainy	X	X				
testGetTasksByPrefixRainy	X		X			
testGetTaskByIdRainy	X			X		
testCreateReportRainy	X				X	

Πίνακας 2 Traceability matrix between use cases and tests

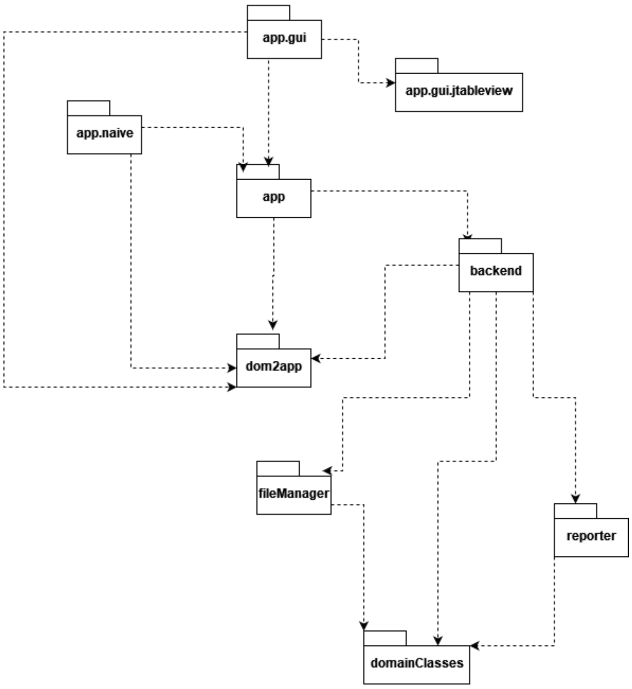
2.3 ΕΚΚΡΕΜΟΤΗΤΕΣ (TODO)

3 ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

3.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ

Η ανάλυση του κώδικα σε υποσυστήματα και πακέτα έχει νόημα μόνο όταν το μέγεθος και η πολυπλοκότητα του κώδικα επιτάσσουν την εν λόγω διαίρεση.

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

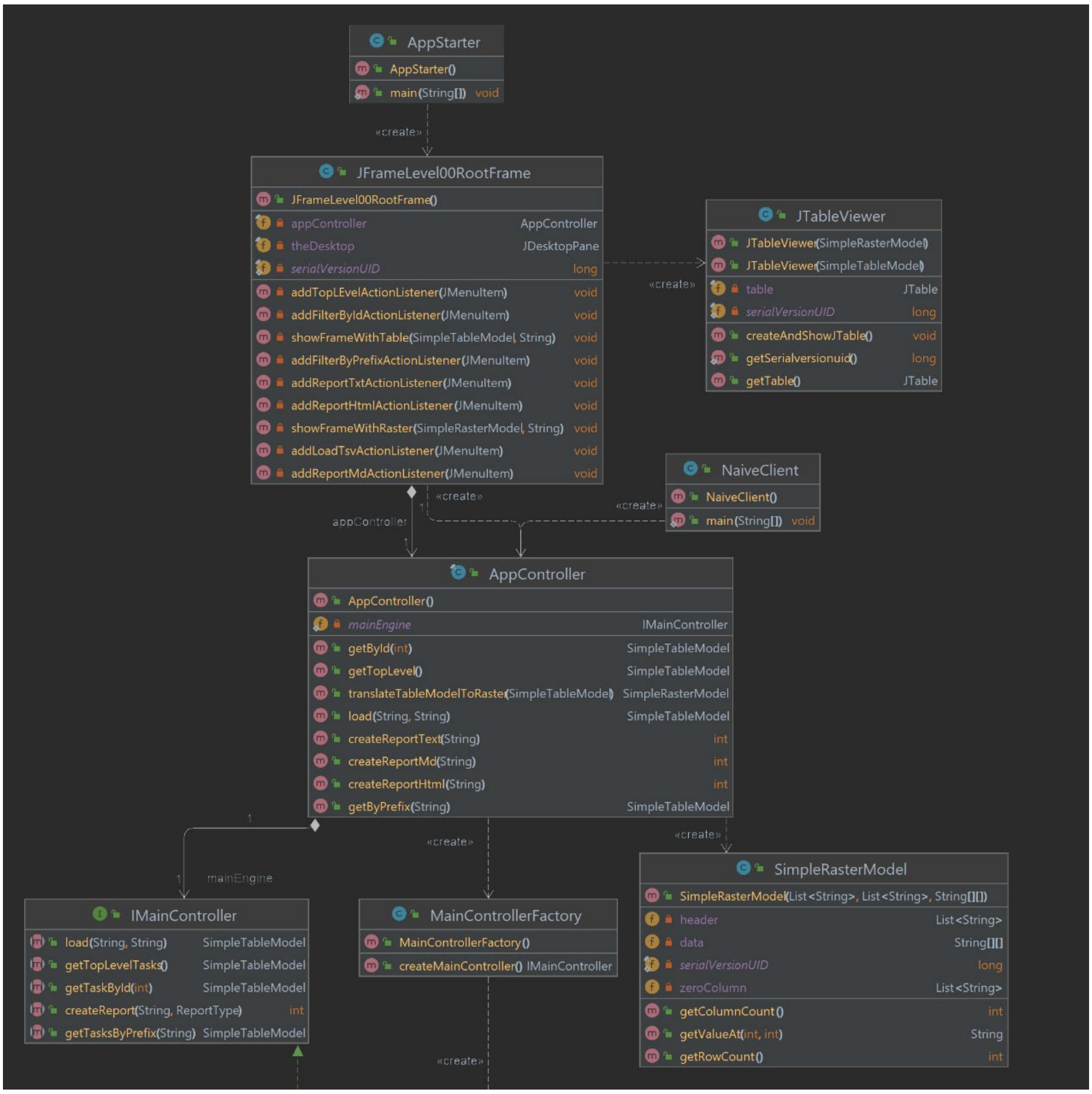
`app, app.gui`
`app.gui.jtableview,`
`app.naive`
	Περιέχουν τις boundary classes που είναι υπεύθυνες για την αλληλεπίδραση με το χρήστη

`backend`	Κεντρική business logic engine, along with the necessary interface to export to the boundary classes
`fileManager`	Υποσύστημα αλληλεπίδρασης με τα αρχεία δεδομένων, για την ανάκτησή τους από το σύστημα
`reporter`	Υποσύστημα παραγωγής αναφορών
`domainClasses`	Domain classes of the system

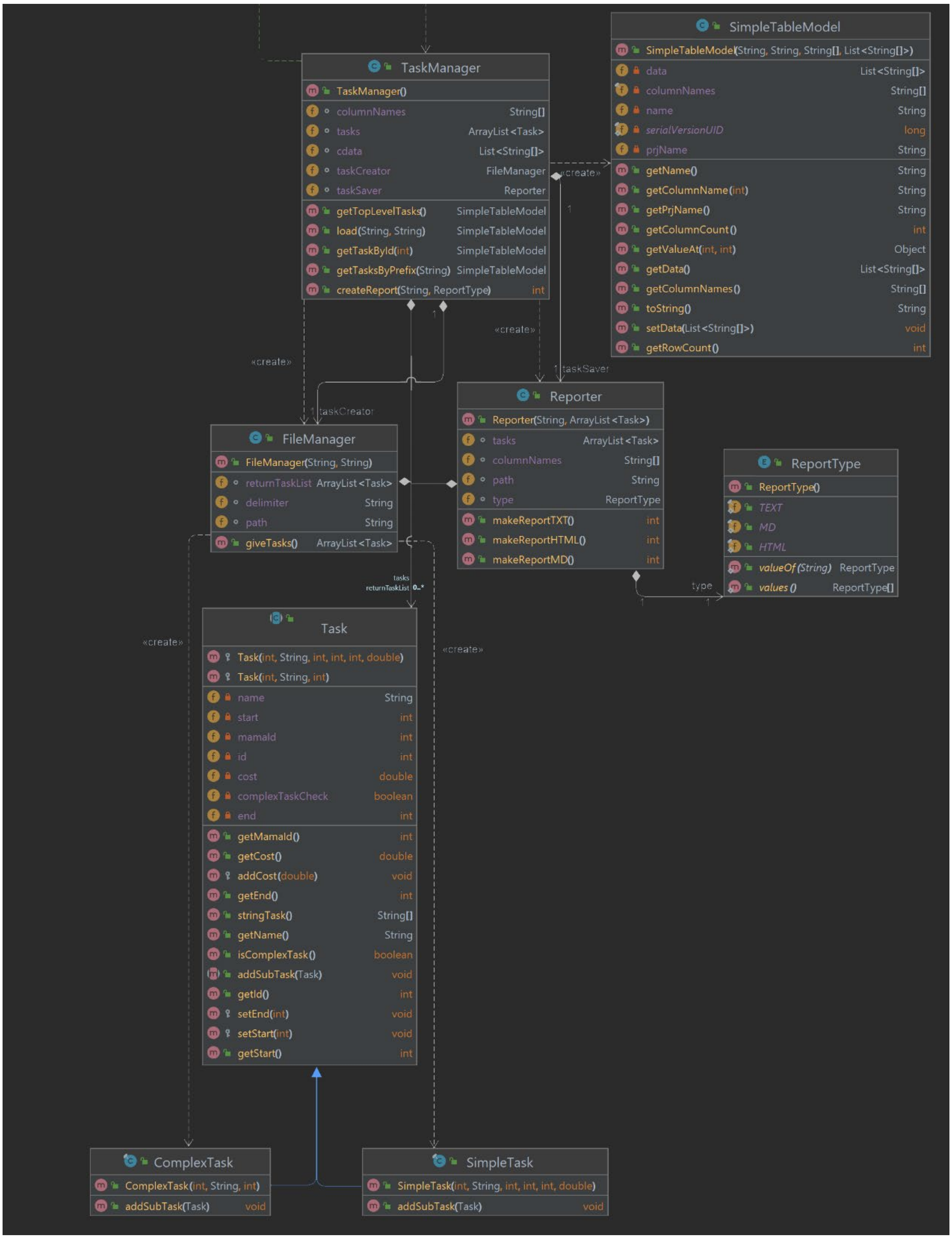
Πίνακας 3. Συνοπτική περιγραφή πακέτων συστήματος (εδώ: από την αξιολόγηση εστιατορίου)

3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

Στην παρούσα υποενότητα, παρατίθενται τα διαγράμματα κλάσεων και ακολουθιών.


AppStarter
AppStarter()
main(String[]) void
«create»
JFrameLevel00RootFrame
JFrameLevel00RootFrame()
appController AppController
theDesktop JDesktopPane
serialVersionUID long
addToplEvelActionListener(JMenuItem) void
addFilterByIdActionListener(JMenuItem) void
showFrameWithTable(SimpleTableModel, String) void
addFilterByPrefixActionListener(JMenuItem) void
addReportTxtActionListener(JMenuItem) void
addReportHtmlActionListener(JMenuItem) void
showFrameWithRaster(SimpleRasterModel, String) void
addLoadTsvActionListener(JMenuItem) void
addReportMdActionListener(JMenuItem) void
«create»
JTableViewer
JTableViewer(SimpleRasterModel)
JTableViewer(SimpleTableModel)
table JTable
serialVersionUID long
createAndShowJTable() void
getSerialversionuid() long
getTable() JTable
NaiveClient
NaiveClient()
main(String[]) void
«create»
«create»
appController
1
1
AppController
AppController()
mainEngine IMainController
getById(int) SimpleTableModel
getTopLevel() SimpleTableModel
translateTableModelToRaster(SimpleTableModel) SimpleRasterModel
load(String, String) SimpleTableModel
createReportText(String) int
createReportMd(String) int
createReportHtml(String) int
getByPrefix(String) SimpleTableModel
1
mainEngine
1
«create»
«create»
IMainController
load(String, String) SimpleTableModel
getTopLevelTasks() SimpleTableModel
getTaskById(int) SimpleTableModel
createReport(String, ReportType) int
getTasksByPrefix(String) SimpleTableModel
MainControllerFactory
MainControllerFactory()
createMainController() IMainController
«create»
SimpleRasterModel
SimpleRasterModel(List<String>, List<String>, String[][])
header List<String>
data String[][]
serialVersionUID long
zeroColumn List<String>
getColumnCount() int
getValueAt(int, int) String
getRowCount() int

Σχήμα 2-1. Διάγραμμα κλάσεων



Σχήμα 2-2. Διάγραμμα κλάσεων (συνέχεια)

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 domainClasses

SimpleTask, ComplexTask και μια αφηρημένη κλάση Task (A) για αυτές τα δύο.

3.3.2 BUSINESS LOGIC CLASSES

Packages backend, fileManager, reporter, dom2app

TasksManager, για την υλοποίηση όλων των use cases στο back-end. TaskManager:

- Interfaces with domain classes via Task.
- Interfaces with boundary classes via IMainController and IMainControllerFactory.

FileManager, για την δημιουργία των Tasks από το αρχείο που δίνει ο client:

- Interfaces with domain classes via Task, SimpleTask and ComplexTask.
- Interfaces with boundary classes via TaskManager

Reporter, για την δημιουργία αρχείων αναφοράς των tasks:

- Interfaces with domain classes via Task.
- Interfaces with boundary classes via TaskManager

SimpleTableModel, για την εμφάνιση των tasks στο GUI

IMainController, interface για την επικοινωνία backend με frontend

IMainControllerFactory, για τη δημιουργία αντικειμένου κλάσης TaskManager που χρησιμοποιείται από το GUI

3.3.3 BOUNDARY CLASSES

Packages app, app.gui app.gui.jtableview, app.naive

NaiveClient, AppStarter, AppController, client classes για την αλληλεπίδραση με το χρήστη μέσω κονσόλας.

JFrameLevel00RootFrame, JTableViewer, SimpleRasterModel, για το graphical user interface με το χρήστη.

3.3.4 ΑΠΕΙΚΟΝΙΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΕ ΜΕΘΟΔΟΥΣ

Υπάρχουν 6 use cases για το διάγραμμα Gantt: Load, Get Top Level Tasks, Get Tasks By Prefix, Get Tasks By Id, Create Report και Exit

ΑΠΕΙΚΟΝΙΣΗ USE CASES ΣΕ ΜΕΘΟΔΟΥΣ

Use case	Back-end methods	Front-end methods

Load	TaskManager.load()
	FileManager.giveTasks()
		AppController.load()
		IMainController.load()

Get Top Level Tasks	TaskManager.getTopLevelTasks()	AppController.getTopLevelTasks() IMainController.getTopLevelTasks ()

Get Tasks By Prefix	`TaskManager.getTasksByPrefix()`	`AppController.getTasksByPrefix()`
		`IMainController.getTasksByPrefix()`

Get Task By Id	`TaskManager.getTaskById()`	`AppController.getTaskById()`
		`IMainController.getTaskById()`

Create Report	`TaskManager.createReport()`
	`Reporter.makeReportTXT()`
	`Reporter.makeReportHTML()`
	`Reporter.makeReportMD()`
		`AppController.createReportText()`
		`AppController.createReportMd()`
		`AppController.createReportHtml()`
		`IMainController.createReport()`

Exit		

Πίνακας 4 Επαλήθευση απεικόνισης use cases σε μεθόδους

4 ΛΟΙΠΑ ΣΧΟΛΙΑ

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.

4.1 ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΠΟΦΑΣΕΙΣ

Στη σχεδίαση μας δεν υπάρχει υλοποίηση του UC6 στο κομμάτι του κώδικα που έχουμε αναλάβει, διότι ο καθηγητής το έχει εφαρμόσει στο GUI και δεν μας δίνει πρόσβαση σε αυτό

4.2 ΣΗΜΕΙΑ ΚΙΝΔΥΝΟΥ

4.3 ΕΚΚΡΕΜΟΤΗΤΕΣ (TODO)