
ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2020-2021

ΟΜΑΔΑ 3088

ΤΖΩΡΤΖΗΣ ΕΥΑΓΓΕΛΟΣ, ΑΜ:3088

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2020

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφείς
2020/11/22	v.01	Οργάνωση απαιτήσεων σε use cases	3088
2020/11/22	v.02	Αρχική σχεδίαση κλάσεων και ελέγχων	3088
2020/12/20	v.03	Διορθώσεις στις use cases, επεκτάσεις στη σχεδίαση κλάσεων και ελέγχων	3088
2020/12/20	v.03	ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ	3088

1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USECASES

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

ΕΙΣΗΓΑΓΕΑΡΧΕΙΟ

ID: UC 1

DESCRIPTION AND GOAL

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

ACTORS (ESP. PRIMARYACTOR)

Ο στοιχειοθέτης/χρήστης.

PRECONDITIONS

-

BASICFLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από μενού την επιλογή «Εισαγωγή αρχείου».
2. Ο χρήστης επιλεγεί το path του αρχείου που επιθυμεί.
3. Ο χρήστης επιλεγεί το τύπο του αρχείου (πρωτογενές ή επισημειωμένο).
4. Ο χρήστης εισάγει ένα φιλικό όνομα (alias) για το αρχείο.
5. Το σύστημα αποθηκεύει το path, το τύπο και το alias που έδωσε ο χρήστης.

EXTENSIONS / VARIATIONS

2. Στην περίπτωση κατά την οποία το path δεν ισχύει/ είναι λάθος, το σύστημα ενημερώνει τον χρήστη και ακυρώνει την εισαγωγή αρχείου.
4. Αν το alias δεν είναι επιτρεπτό όνομα το σύστημα ζητάει πάλι από τον χρήστη να εισάγει φιλικό όνομα για το αρχείο.

POSTCONDITIONS

Το σύστημα έχει αποθηκεύσει τη τοποθεσία του αρχείου, το τύπο του και ένα φιλικό όνομα.

ΚΑΤΕΓΡΑΨΕΚΑΝΟΝΕΣΓΙΑΠΡΩΤΟΓΕΝΗΑΡΧΕΙΑ

ID: UC 2

DESCRIPTION AND GOAL

Η use case «ΚατέγραψεΚανόνεςΓιαΠρωτογενήΑρχεία» καταγράφει τους κανόνες που πρόκειται να εφαρμοστούν στο πρωτογενές κείμενο εισόδου.

ACTORS (ESP. PRIMARYACTOR)

Ο στοιχειοθέτης/χρήστης.

PRECONDITIONS

Πρέπει να έχει εισαχθεί ένα αρχείο από τον χρήστη.

BASICFLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από μενού την επιλογή «Πρόσθεση κανόνων».
2. Ο χρήστης επιλέγει τους κανόνες που επιθυμεί επαναληπτικά.
 - 2.1. Πρώτα επιλέγει το τύπο του κανόνα (omit, h1, h2, bold, italics).
 - 2.2. Μετά επιλέγει που θα εφαρμοστεί ο κανόνας (positions, starts_with, all_caps).
 - 2.2.1. Εάν επιλέξει positions, τότε εισάγει τις θέσεις των παραγράφων που επιθυμεί να εφαρμόσει τον κανόνα.
 - 2.2.2. Εάν επιλέξει starts_with, τότε εισάγει τη γραμματοσειρά με την οποία ξεκινά η παράγραφος που επιθυμεί να εφαρμόσει τον κανόνα.
 - 2.3. Το σύστημα προσθέτει τον κανόνα στη λίστα με τους κανόνες.
3. Ο χρήστης επιλέγει από το μενού «Επιβεβαίωση κανόνων».
4. Το σύστημα καταγράφει τους κανόνες.

EXTENSIONS / VARIATIONS

- 2.2 Στην περίπτωση που ο χρήστης επιλέξει all-caps, ο κανόνας δεν χρειάζεται παραπάνω πληροφορίες και προστίθεται στη λίστα με το τρίτο στοιχείο κενό.
4. Στην περίπτωση κατά την οποία δεν έχουν προστεθεί κανόνες, το σύστημα ενημερώνει το χρήστη.

POSTCONDITIONS

Οι κανόνες έχουν καταγραφεί.

ΚΑΤΕΓΡΑΨΕΚΑΝΟΝΕΣΓΙΑΕΠΙΣΗΜΕΙΩΜΕΝΑΑΡΧΕΙΑ

ID: UC 3

DESCRIPTION AND GOAL

Η use case «ΚατέγραψεΚανόνεςΓιαΕπισημειώμεναΑρχεία» καταγράφει τους κανόνες που πρόκειται να εφαρμοστούν στο επισημειωμένο κείμενο εισόδου.

ACTORS (ESP. PRIMARYACTOR)

Ο στοιχειοθέτης/χρήστης.

PRECONDITIONS

Πρέπει να έχει εισαχθεί ένα αρχείο από τον χρήστη.

BASICFLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από μενού την επιλογή «Πρόσθεση κανόνων».
2. Ο χρήστης επιλέγει τους κανόνες που επιθυμεί επαναληπτικά.
 - 2.1. Πρώτα επιλέγει το τύπο του κανόνα (omit, h1, h2, bold, italics).
 - 2.2. Ύστερα εισάγει τη γραμματοσειρά-πρόθεμα (prefix) με την οποία ξεκινά η παράγραφος, που επιθυμεί να εφαρμόσει τον κανόνα.
 - 2.3. Το σύστημα προσθέτει τον κανόνα στη λίστα με τους κανόνες και το prefix στη λίστα με τα prefixes.
3. Ο χρήστης επιλέγει από το μενού «Επιβεβαίωση κανόνων».
4. Το σύστημα καταγράφει τους κανόνες, περιλαμβάνοντας και τα prefixes.

EXTENSIONS / VARIATIONS

4. Στην περίπτωση κατά την οποία δεν έχουν προστεθεί κανόνες, το σύστημα ενημερώνει το χρήστη.

POSTCONDITIONS

Οι κανόνες έχουν καταγραφεί.

ΕΠΕΞΕΡΓΑΣΟΥ ΑΡΧΕΙΟ

ID: UC 4

DESCRIPTION AND GOAL

Η use case «Επεξεργασίας Αρχείου» επεξεργάζεται το αρχείο εισόδου με βάση τους κανόνες που έχουν καταγραφεί.

ACTORS (ESP. PRIMARYACTOR)

Ο στοιχειοθέτης/χρήστης.

PRECONDITIONS

Πρέπει να έχει εισαχθεί ένα αρχείο από τον χρήστη και να έχουν καταγραφεί κανόνες.

BASICFLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από το μενού την επιλογή «Επεξεργασία αρχείου».
2. Το σύστημα φορτώνει το αρχείο και χαρακτηρίζει τις παράγραφους του με βάση τους κανόνες που καταγραφήκαν.

EXTENSIONS / VARIATIONS

-

POSTCONDITIONS

Το αρχείο έχει χαρακτηριστεί.

ΠΑΡΗΓΑΓΕΣΤΑΤΙΣΤΙΚΑΣΕΑΝΑΦΟΡΑ

ID: UC 5

DESCRIPTION AND GOAL

Η use case «Παρήγαγε Στατιστικά Σε Αναφορά» παράγει μια αναφορά με διάφορα στατιστικά για το κείμενο.

ACTORS (ESP. PRIMARYACTOR)

Ο στοιχειοθέτης/χρήστης.

PRECONDITIONS

Πρέπει να έχει φορτωθεί και χαρακτηριστεί το κείμενο.

BASICFLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από μενού την επιλογή «Εμφάνιση αναφοράς».
2. Το σύστημα υπολογίζει στατιστικά, όπως τον αριθμό των παραγράφων, τον αριθμό των λέξεων και τον αριθμό λέξεων ανά γραμμή.
3. Το σύστημα επιστρέφει μια λίστα με τα στατιστικά.

EXTENSIONS / VARIATIONS

-

POSTCONDITIONS

Εμφανίζεται η λίστα με τα στατιστικά.

ΕΞΗΓΑΓΕΑΡΧΕΙΟΣΕMARKDOWN

ID: UC 6

DESCRIPTION AND GOAL

Η usecase «ΕξήγαγεΑρχείοΣεMarkdown» εξάγει το χαρακτηρισμένο αρχείο σε markdown.

ACTORS (ESP. PRIMARYACTOR)

Ο στοιχειοθέτης/χρήστης.

PRECONDITIONS

Πρέπει να έχει φορτωθεί και χαρακτηριστεί το κείμενο.

BASICFLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από μενού την επιλογή «Εξαγωγή σε markdown».
2. Το σύστημα εξάγει το κείμενο σε markdown.

EXTENSIONS / VARIATIONS

-

POSTCONDITIONS

Πλέον υπάρχει ένα αρχείο markdown.

ΕΞΗΓΑΓΕΑΡΧΕΙΟΣΕPDF

ID: UC 7

DESCRIPTION AND GOAL

Η use case «ΕξήγαγεΑρχείοΣεPDF» εξάγει το χαρακτηρισμένο αρχείο σε PDF.

ACTORS (ESP. PRIMARYACTOR)

Ο στοιχειοθέτης/χρήστης.

PRECONDITIONS

Πρέπει να έχει φορτωθεί και χαρακτηριστεί το κείμενο.

BASICFLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από μενού την επιλογή «Εξαγωγή σε PDF».
2. Το σύστημα εξάγει το κείμενο σε PDF.

EXTENSIONS / VARIATIONS

-

POSTCONDITIONS

Πλέον υπάρχει ένα αρχείο PDF.

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 USECASES VIA SYSTEM TESTS

2.1.1 USE CASE UC1: ΕΙΣΗΓΑΓΕ ΑΡΧΕΙΟ

Test cases

Description	ON	any context
	RECEIVING	Ένα path αρχείου, το τύπο του και ένα alias, που εισάγονται από τον χρήστη
	ENSURE	That the System
	OUTPUTS	Αποθηκεύει τα δεδομένα
	SUCH THAT	state is intact

ID	T1_V0_01	HappyDayScenario for ClientUI.getFileInput()
Pre-cond.		No specific precondition constructed
Input		A file path
Output		Input is saved
Post-cond.		Input is passed as parameters to a new Engine() object
Method To test		ClientUI.getFileInput()

ID	T1_V1_01	HappyDayScenario for ClientUI.printTypeMenu()
Pre-cond.		No specific precondition constructed
Input		A file type
Output		Input is saved
Post-cond.		Input is passed as parameters to a new Engine() object
Method To test		ClientUI.printTypeMenu()

ID	T1_V2_01	HappyDayScenario for ClientUI.getAlias()
Pre-cond.		No specific precondition constructed
Input		An alias
Output		Input are saved
Post-cond.		Input is passed as parameters to a new Engine() object
Method To test		ClientUI.getAlias()

Involved methods

ClientUI.getFileInput(),

ClientUI.printTypeMenu(),

ClientUI.getAlias()

2.1.2 USE CASE UC2: ΚΑΤΕΓΡΑΨΕ ΚΑΝΟΝΕΣ ΓΙΑ ΠΡΩΤΟΓΕΝΗ ΑΡΧΕΙΑ

Test cases

<i>Description</i>	<i>ON</i>	<i>Ένα αρχείο έχει εισαχθεί στο πρόγραμμα</i>
	<i>RECEIVING</i>	<i>Μια λίστα με τους κανόνες</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>Καταγραφεί τους κανόνες</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T2_V0_01	HappyDayScenario for registerInputRuleSetForPlainFiles()
Pre-cond.		-
Input		A list with rule specification
Output		A list with the rules
Post-cond.		The rules have been registered
Method To test		Engine.registerInputRuleSetForPlainFiles(List<List<String>> inputSpec)

ID	T2_V1_01	HappyDayScenario for RuleSetCreator.createRuleSet()
Pre-cond.		-
Input		A list with rule specification
Output		A list with the rules
Post-cond.		The rules have been registered
Method To test		RuleSetCreator.createRuleSet()

ID	T2_V1_02	Wrong rule syntax for RuleSetCreator.createRuleSet()
Pre-cond.		-
Input		A list with rule specification
Output		An error message
Post-cond.		The program exits
Method To test		RuleSetCreator.createRuleSet()

ID	T2_V1_03	Null rule for RuleSetCreator.createRuleSet()
Pre-cond.		-
Input		A list with rule specification with a null rule
Output		An error message
Post-cond.		The program exits
Method To test		RuleSetCreator.createRuleSet()

ID	T2_V2_01	HappyDayScenario for ConcreteRuleFactory.createRuleInPosition()
Pre-cond.		The rule is of position type
Input		The lineblocks and the list of positions to apply the rule
Output		-
Post-cond.		A new RuleInPosition() object is created
Method To test		ConcreteRuleFactory.createRuleInPosition()

ID	T2_V2_02	Null position for ConcreteRuleFactory.createRuleInPosition()
Pre-cond.		The rule is of position type
Input		The lineblocks and a null list of positions to apply the rule
Output		An error message
Post-cond.		The program exits
Method To test		ConcreteRuleFactory.createRuleInPosition()

ID	T2_V3_01	HappyDayScenario for ConcreteRuleFactory.createRuleStartWith()
Pre-cond.		The rule is of starts_with type
Input		The prefixes to apply the rule
Output		-
Post-cond.		A new RuleStartsWith() object is created
Method To test		ConcreteRuleFactory.createStartWith()

ID	T2_V3_02	Null rule starts_with for ConcreteRuleFactory.createRuleStartWith()
Pre-cond.		The rule is of starts_with type and null prefixes
Input		Null prefixes to apply the rule
Output		An error message
Post-cond.		The program exits
Method To test		ConcreteRuleFactory.createStartWith()

Involved methods

```

Engine.registerInputRuleSetForPlainFiles(List<List<String>> inputSpec),
RuleSetCreator.createRuleSet(),
ConcreteRuleFactory.createRuleInPosition(),
ConcreteRuleFactory.createStartWith(),
ConcreteRuleFactory.createRuleUndefined(),
ConcreteRuleFactory.createRuleAllCaps()

```

2.1.3 USE CASE UC3: ΚΑΤΕΓΡΑΨΕ ΚΑΝΟΝΕΣ ΓΙΑ ΕΠΙΣΗΜΕΙΩΜΕΝΑ ΑΡΧΕΙΑ

Test cases

<i>Description</i>	<i>ON</i>	<i>Ένα αρχείο έχει εισαχθεί στο πρόγραμμα</i>
	<i>RECEIVING</i>	<i>Μια λίστα με τους κανόνες και μια λίστα με τα prefixes</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>Καταγραφεί τους κανόνες</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T3_V0_01	HappyDayScenario for registerInputRuleSetForAnnotatedFiles()
Pre-cond.		-
Input		A list with rule specifications and a list with prefixes
Output		A set with the rules
Post-cond.		The rules have been registered
Method To test		Engine.registerInputRuleSetForAnnotatedFiles()

ID	T3_V0_02	Rule is not Starts_with for registerInputRuleSetForAnnotatedFiles()
Pre-cond.		-
Input		A list with rule specifications that contain a rule of other than STARTS_WITH and a list with prefixes
Output		An error message and a null
Post-cond.		The rule set is null
Method To test		Engine.registerInputRuleSetForAnnotatedFiles()

Involved methods

Engine.registerInputRuleSetForAnnotatedFiles(),

RuleSetCreator.createRuleSet(),

ConcreteRuleFactory.createStartWith(),

ConcreteRuleFactory.createRuleUndefined()

2.1.4 USE CASE UC4: ΕΠΕΞΕΡΓΑΣΟΥ ΑΡΧΕΙΟ

Test cases

<i>Description</i>	<i>ON</i>	<i>Ένα path έχει εισαχθεί</i>
	<i>RECEIVING</i>	<i>Το αρχείο (και πιθανώς καταγεγραμμένους κανόνες)</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>Φορτώνει το αρχείο (και το χαρακτηρίζει εφόσον κανόνες έχουν καταγραφεί)</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T4_V0_01	HappyDayScenario for Engine.loadFileAndCharacterizeBlocks ()
Pre-cond.		A file has been given (and possibly a rule set)
Input		-
Output		The count of paragraphs
Post-cond.		The file has been loaded and if rules were given, it is characterized too
Method To test		Engine.loadFileAndCharacterizeBlocks ()

ID	T4_V1_01	HappyDayScenario for Engine.loadRawDocument()
Pre-cond.		A file has been given
Input		The file path
Output		The count of paragraphs
Post-cond.		The file has been loaded
Method To test		Engine.loadRawDocument ()

ID	T4_V2_01	HappyDayScenario for RawFileLineLoader.load()
Pre-cond.		A file has been given
Input		The file path and the lineblocks list
Output		The count of paragraphs
Post-cond.		The file has been loaded
Method To test		RawFileLineLoader.load()

ID	T4_V2_02	File does not exist scenario for RawFileLineLoader.load()
Pre-cond.		A wrong file has been given
Input		The wrong file path and the lineblocks list
Output		An error message
Post-cond.		The program notifies the user with an error message
Method To test		RawFileLineLoader.load()

ID	T4_V3_01	HappyDayScenario for Engine.characterizeLineblocks()
Pre-cond.		A file and a rule set have been given and there are lineblocks
Input		A document object and a rule set
Output		-
Post-cond.		The file is characterized, each lineblock has a format and a style applied to it based on the rules
Method To test		Engine.characterizeLineblocks()

Involved methods

`Engine.loadFileAndCharacterizeBlocks()`,

`Engine.loadRawDocument(String fileName)`,

`RawFileLineLoader.load(String filePath, List<LineBlock> lineblocks)`,

`Engine.characterizeLineblocks(Document document, RuleSet ruleSet)`

2.1.5 USE CASE UC5: ΠΑΡΗΓΑΓΕ ΣΤΑΤΙΣΤΙΚΑ ΣΕ ΑΝΑΦΟΡΑ

Test cases

<i>Description</i>	<i>ON</i>	<i>Ένα αρχείο έχει επεξεργαστεί</i>
	<i>RECEIVING</i>	<i>Το αρχείο</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>Μια λίστα με στατιστικά</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T5_V0_01	HappyDayScenario for reportWithStats()
Pre-cond.		A file is loaded
Input		-
Output		A list with statistics report
Post-cond.		The report list is created
Method To test		Engine.reportWithStats()

Involved methods

Engine.reportWithStats(),

Engine.loadFileAndCharacterizeBlocks ()

2.1.6 USE CASE UC6: ΕΞΗΓΑΓΕ ΑΡΧΕΙΟ ΣΕ MARKDOWN

Test cases

<i>Description</i>	<i>ON</i>	<i>Ένα αρχείο έχει εισαχθεί και επεξεργαστεί</i>
	<i>RECEIVING</i>	<i>Το αρχείο και το όνομα του αρχείου εξόδου</i>
	<i>ENSURE</i>	<i>That the System</i>
	<i>OUTPUTS</i>	<i>Εξάγει το αρχείο σε markdown</i>
	<i>SUCH THAT</i>	<i>state is intact</i>

ID	T6_V0_01	HappyDayScenario for Engine.exportMarkdown()
Pre-cond.		-
Input		A name for the output file
Output		The number of paragraphs
Post-cond.		The file is created in markdown form
Method To test		Engine.exportMarkdown()

Involved methods

Engine.exportMarkdown(String outputFileName),

Engine.loadFileAndCharacterizeBlocks ()

2.1.7 USE CASE UC7: ΕΞΗΓΑΓΕ ΑΡΧΕΙΟ ΣΕ PDF

Test cases

Description	ON	Ένα αρχείο έχει εισαχθεί και επεξεργαστεί
	RECEIVING	Το αρχείο και το όνομα του αρχείου εξόδου
	ENSURE	That the System
	OUTPUTS	Εξάγει το αρχείο σε PDF
	SUCH THAT	state is intact

ID	T7_V0_01	HappyDayScenario for Engine.exportPdf()
Pre-cond.		-
Input		A name for the output file
Output		The number of paragraphs
Post-cond.		The file is created in markdown form
Method To test		Engine.exportPdf()

Involved methods

Engine.exportPdf(String outputFileName),

Engine.loadFileAndCharacterizeBlocks()

2.2 TRACEABILITY MATRIX

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

UC1	Εισήγαγε Αρχείο
UC2	Κατέγραψε Κανόνες Για Πρωτογενή Αρχεία
UC3	Κατέγραψε Κανόνες Για Επισημειώμενα Αρχεία
UC4	Επεξεργάσου Αρχείο
UC5	Παρήγαγε Στατιστικά Σε Αναφορά
UC6	Εξήγαγε Αρχείο Σε Markdown
UC7	Εξήγαγε Αρχείο Σε PDF

Πίνακας1 Σύνοψη use cases και των id'ς 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	UC7
T1_V0_01	X			X			
T1_V1_01	X			X			
T1_V2_01	X						
T2_V0_01		X					
T2_V1_01		X	X				
T2_V1_02		X	X				
T2_V1_03		X	X				
T2_V2_01		X	X				
T2_V2_02		X	X				
T2_V3_01		X	X				
T2_V3_02		X	X				
T3_V0_01			X				
T3_V0_02			X				
T4_V0_01				X	X	X	X
T4_V1_01				X	X	X	X
T4_V2_01				X	X	X	X
T4_V2_02				X			
T4_V3_01				X	X	X	X
T5_V0_01					X		
T6_V0_01						X	
T7_V0_01							X

Πίνακας1 Traceability matrix between use cases and tests

2.3 ΕΚΚΡΕΜΟΤΗΤΕΣ (TODO)

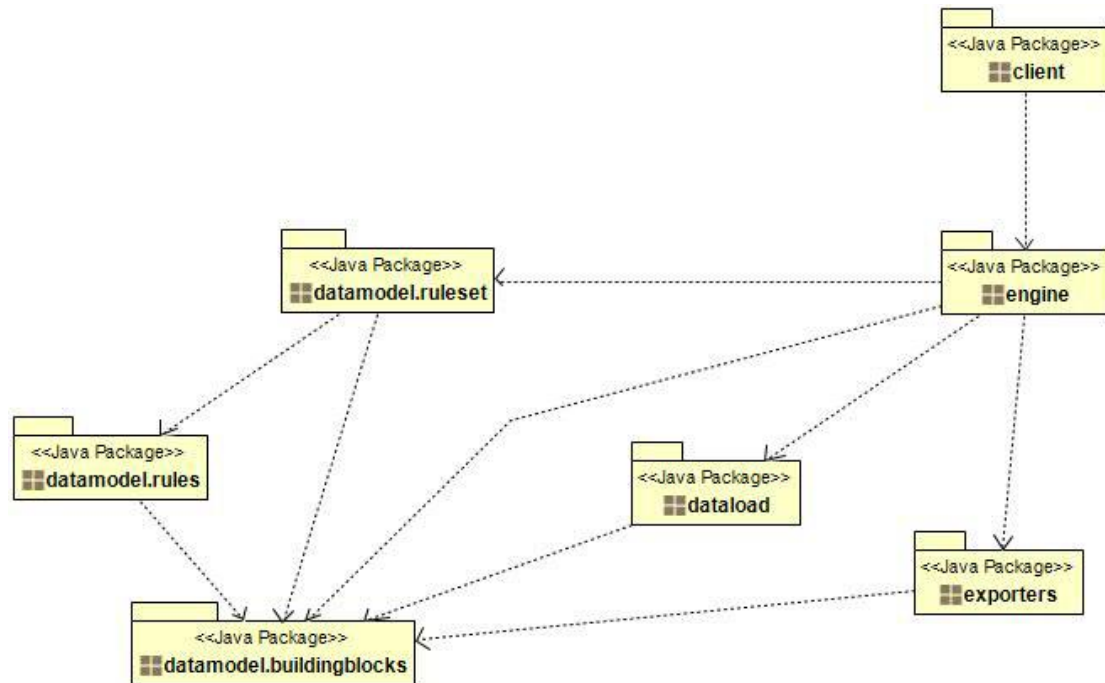
Εκκρεμούν μη υλοποιημένοι έλεγχοι ως ακολούθως:

-

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.

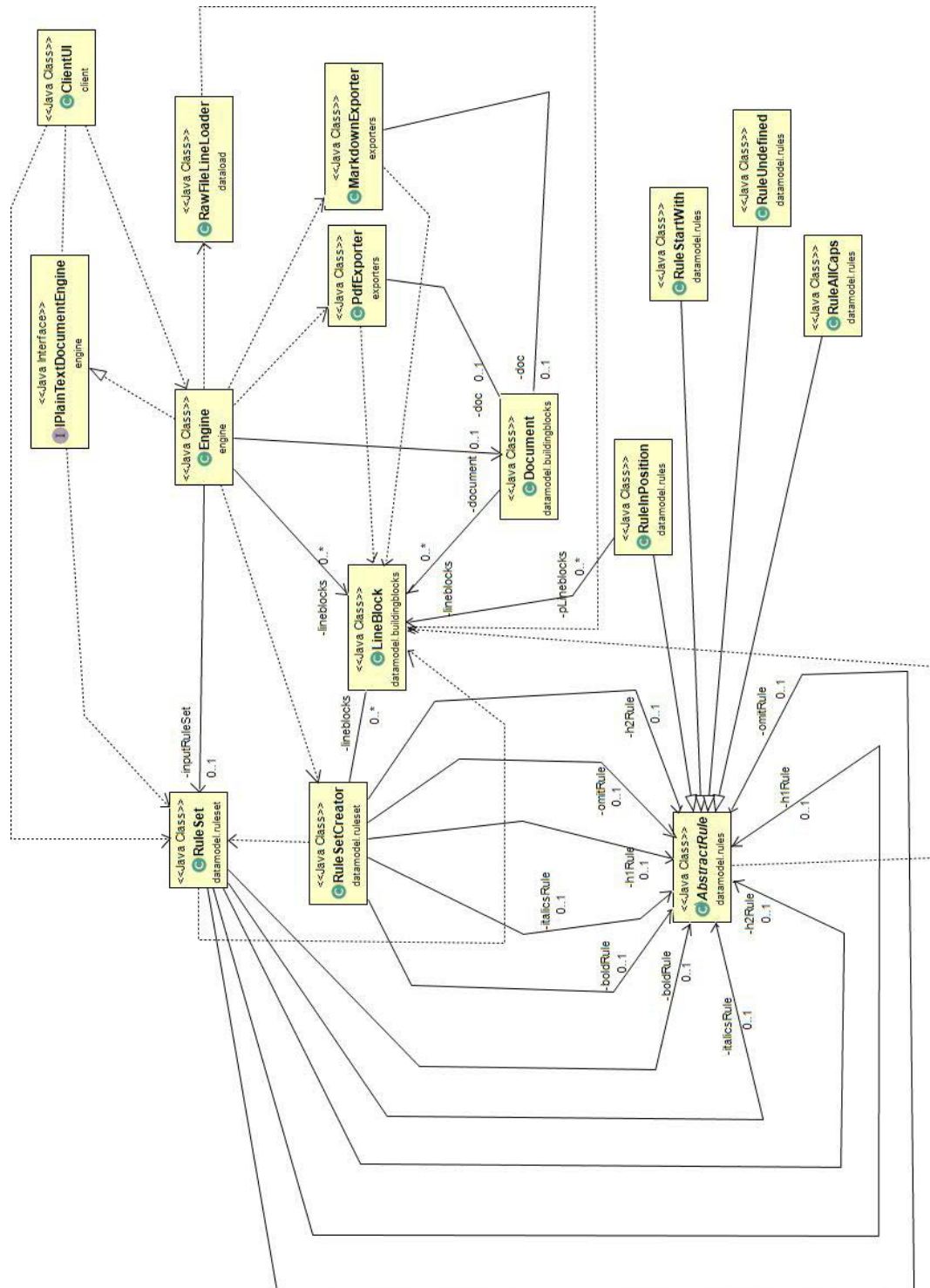
ΠΑΚΕΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

client	Περιέχει τη boundary class που είναι υπεύθυνη για την αλληλεπίδραση με το χρήστη
engine	Κεντρική business logic engine, along with the necessary interface to export to the boundary classes. For each file to process, we will launch a dedicated Engine object – it is the responsibility of the client (if deemed necessary) to handle multiple such engine instances. At the serverside, each engine instance handles a single input file.
dataload	The package responsible for reading the input file and loading it in the appropriate collection(s) in memory.
exporters	The package responsible for creating output files with exporter classes for PDF and Markdown.
datamodel	Domain classes of the system

Πίνακας 2. Συνοπτική περιγραφή πακέτων συστήματος.

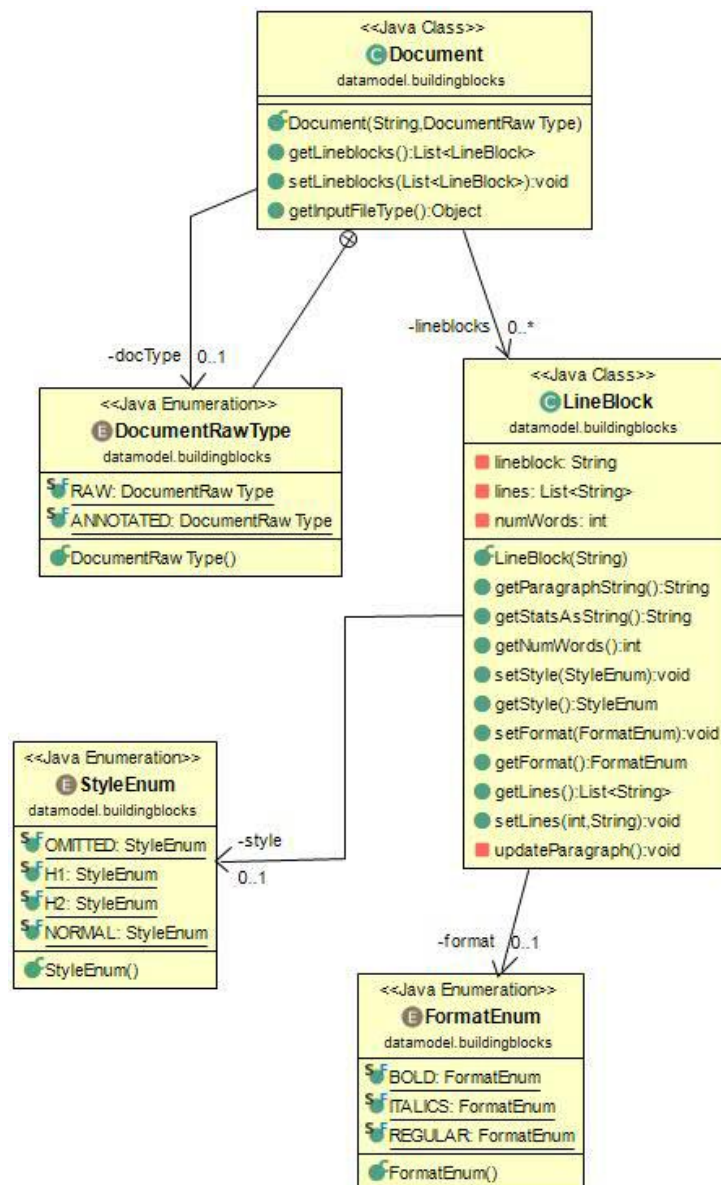
3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

Στην παρούσα υποενότητα, παρατίθενται τα διαγράμματα κλάσεων και ακολουθιών.



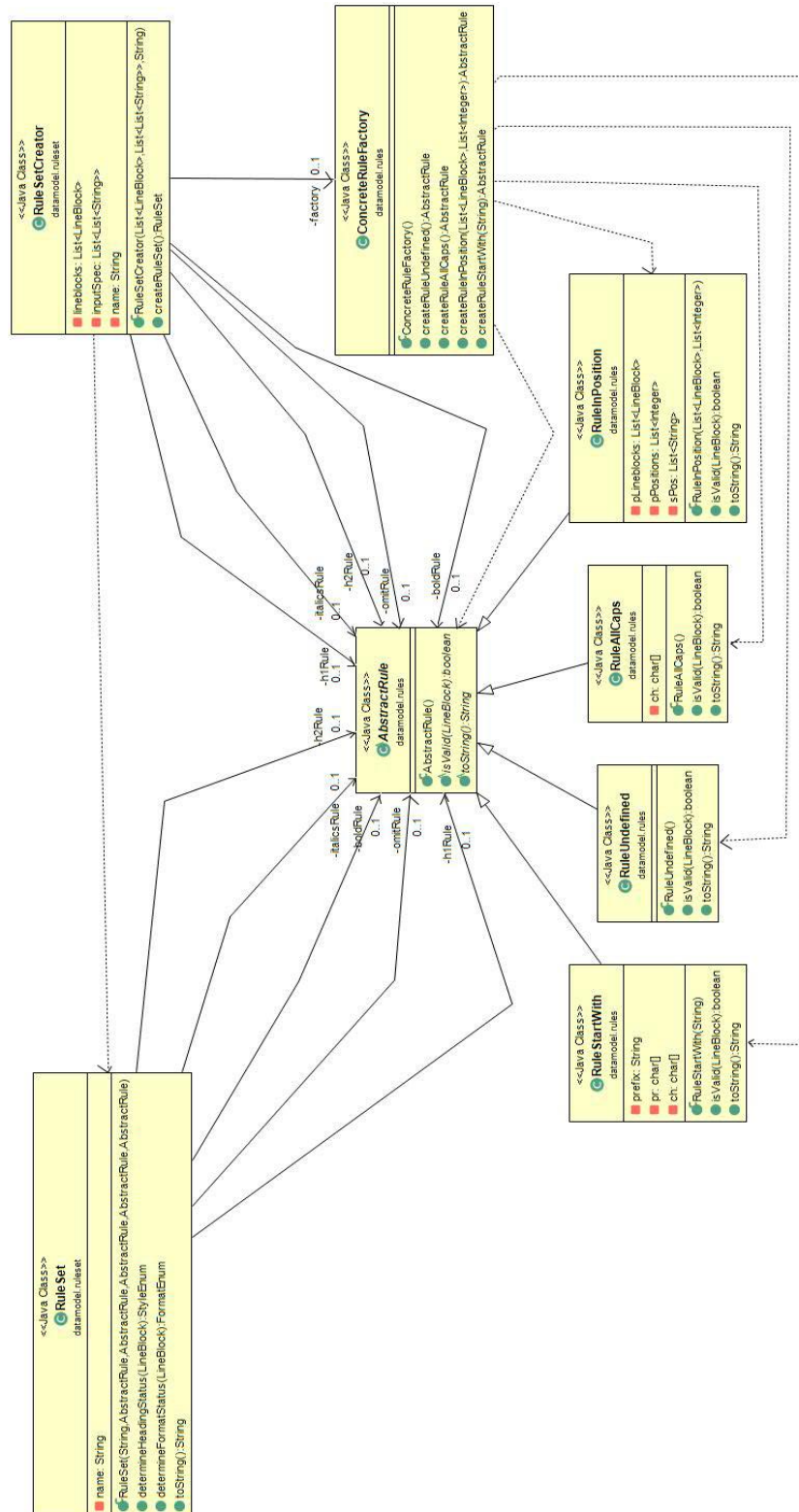
Σχήμα 2. Συνοπτικό διάγραμμα κλάσεων για τη κύρια λειτουργία του προγράμματος, περιλαμβάνοντας κλάσεις από διαφορετικά πακέτα.

Package datamodel.buildingblocks;



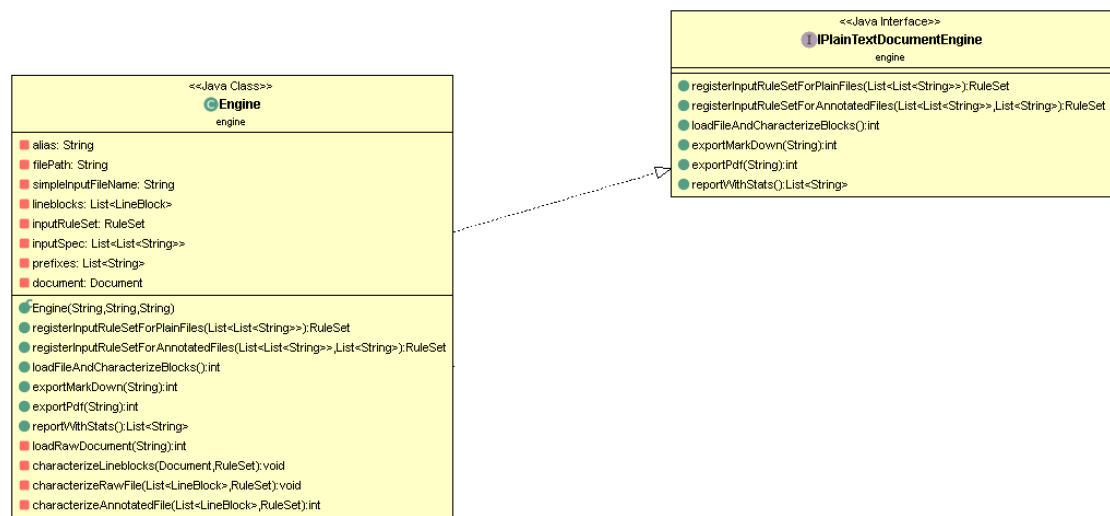
Σχήμα3. Διάγραμμα κλάσεων για το πακέτο `datamodel.buildingblocks`.

Package datamodel.ruleset;
Package datamodel.rules;



Σχήμα 4. Διάγραμμα κλάσεων για τον ορισμό κανόνων.

Package engine;



Σχήμα 5. Διάγραμμα κλάσεων για την κύρια κλάση engine.

3.3 ΑΝΑΛΥΣΗ ΚΛΑΣΕΩΝ ΚΑΙ ΣΥΝΕΠΕΙΑ ΠΡΟΣ ΤΙΣ ΑΠΑΙΤΗΣΕΙΣ

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

3.3.1 DOMAIN CLASSES

Package: datamodel.buildingblocks	Document περιέχει μια λίστα με lineblocks, LineBlock έχει μεθόδους σχετικά με μια παράγραφο, FormatEnum, StyleEnum enumerators για στυλ και γραμματοσειρά
Package: datamodel.ruleset	RuleSetCreator δημιουργεί το ruleset, RuleSet περιέχει τους κανόνες
Package: datamodel.rules	AbstractRule μια αφηρημένη κλάση κανόνων, ConcreteRuleFactory ένα factory που επιστρέφει κανόνες, RuleAllCaps, RuleInPosition, RuleStartWith, RuleUndefined κλάσεις κανόνων για διάφορους τύπους κανόνων.

3.3.2 BUSINESS LOGIC CLASSES

Package: engine	Engine υλοποιεί όλες τις use cases: <ul style="list-style-type: none">- Διεπαφή με όλες τις domain classes μέσω των Document, LineBlock, RuleSetCreator, Ruleset- Διεπαφή με τις boundary classes μέσω των RawFileLoader, MarkdownExporter, PdfExporter, ClientUI
--------------------	--

3.3.3 BOUNDARY CLASSES

Package: client	ClientUI υλοποιεί τη διεπαφή με το χρήστη.
Package: exporters	MarkdownExporter εξάγει σε αρχείο markdown, PdfExporter εξάγει σε αρχείο pdf.
Package: dataload	RawFileLineLoader φορτώνει ένα αρχείο εισόδου σε lineblocks

3.3.4 ΑΠΕΙΚΟΝΙΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΕ ΜΕΘΟΔΟΥΣ

ΑΠΕΙΚΟΝΙΣΗ USE CASES ΣΕ ΜΕΘΟΔΟΥΣ

Use case	Back-end methods	Front-end methods
ΕισήγαγεΑρχείο	Engine()	ClientUI.main()
ΚατέγραψεΚανόνεςΓια ΠρωτογενήΑρχεία	Engine. registerInputRuleSetForPlainFiles()	ClientUI.main()
ΚατέγραψεΚανόνεςΓια ΕπισημειωμέναΑρχεία	Engine. registerInputRuleSetForAnnotatedFiles()	ClientUI.main()
ΕπεξεργάσουΑρχείο	Engine.loadFileAndCharacterizeBlocks()	ClientUI.main()
ΠαρήγαγεΣτατιστικά ΣεΑναφορά	Engine.reportWithStats()	ClientUI.main()
ΕξήγαγεΑρχείο ΣεMarkdown	Engine.exportMarkDown()	ClientUI.main()
ΕξήγαγεΑρχείοΣεPDF	Engine.exportPdf()	ClientUI.main()

Πίνακας 3 Επαλήθευση απεικόνισης usecases σε μεθόδους

3.4 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΩΝ

Αν ζητούνται / υπάρχουν

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

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

4.2 ΣΗΜΕΙΑ ΚΙΝΔΥΝΟΥ

Αν υπάρχουν.

4.3 ΕΚΚΡΕΜΟΤΗΤΕΣ (TODO)

-