ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019-2020

ΟΜΑΔΑ 4387-4508

ΚΙΚΙΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ, ΑΜ:4387

ΤΣΑΜΠΙΡΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ, ΑΜ:4508

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2019

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφείς
2019/11/03	v.01	Οργάνωση απαιτήσεων σε use cases	4387-4508
2019/11/24	v.02	Αρχική σχεδίαση κλάσεων και ελέγχων	4387-4508
2019/12/13	v.03	Διορθώσεις στις uses cases, επεκτάσεις στη σχεδίαση κλάσεων και ελέγχων	4387-4508
2019/12/14	v.04	ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ	4387-4508

1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USE CASES

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

LOADDATA

ID: UC1

DESCRIPTION AND GOAL

The UC LoadData, reads the data from the given file and stores them.

ACTORS (ESP. PRIMARY ACTOR)

Analyst

PRECONDITIONS

-

BASIC FLOW

1. The UC starts when the actor selects the 1st option from the menu.
2. The system asks for the file path, the delimiter used and if the file has a header line.

EXTENSIONS / VARIATIONS

2a. In case of wrong file path, the system asks for a valid file path.

2b. In case of wrong delimiter, the system asks for a valid delimiter.

POST CONDITIONS

The system returns the number of valid lines that have been loaded.

SPECIAL REQUIREMENTS, ISSUES, RISKS AND OTHER COMMENTS

The input file must have the correct format, with 9 columns.

AGGREGATEBYTIMEUNIT

ID: UC2

DESCRIPTION AND GOAL

The UC AggregateByTimeUnit, aggregates measurements by a time unit, e.g., month, day of week, period of day etc.

ACTORS (ESP. PRIMARY ACTOR)

Analyst

PRECONDITIONS

Data must have been loaded first.

BASIC FLOW

1. The UC starts when the actor selects the 2nd option from the menu.
2. The system asks the actor for a description of the data, in order to show it on the final report, the type of aggregation (season/month/day of week/period of day), the aggregate function (avg/sum).

EXTENSIONS / VARIATIONS

2a. In case of wrong type of aggregation, the system asks for a valid one.

2b. In case of wrong aggregate function, the system asks for a valid one.

POST CONDITIONS

The data is have been aggregated correctly.

SPECIAL REQUIREMENTS, ISSUES, RISKS AND OTHER COMMENTS

-

REPORTRESULTSINFILE

ID: UC3

DESCRIPTION AND GOAL

The UC ReportResultInFile, reports the contents of an aggregate result to a file.

ACTORS (ESP. PRIMARY ACTOR)

Analyst

PRECONDITIONS

Data must have been loaded and aggregated first.

BASIC FLOW

1. The UC starts when the actor selects the 3rd option from the menu.
2. The system asks the actor for the file format of the output file (html/txt/md) and for the output directory path.

EXTENSIONS / VARIATIONS

2a. In case of wrong file format, the system asks for a valid one.

2b. In case of wrong output directory, the system asks for a valid one.

POST CONDITIONS

The system returns 0 and the place where the report has been saved, with filename FinalReport.

SPECIAL REQUIREMENTS, ISSUES, RISKS AND OTHER COMMENTS

After a completed export, the system saves all the metadata on a list for future reference (lost if the application terminates).

SHOWHISTORY

ID: UC4

DESCRIPTION AND GOAL

The UC ShowHistory, shows the types of the reports requested previously from the actor.

ACTORS (ESP. PRIMARY ACTOR)

Analyst

PRECONDITIONS

There's already at least one completed report.

BASIC FLOW

1. The UC starts when the actor selects the 4th option from the menu.
2. The system returns a list with the metadata of previously requested reports.

EXTENSIONS / VARIATIONS

2a. If there's no report, the system returns an error message.

POST CONDITIONS

-

SPECIAL REQUIREMENTS, ISSUES, RISKS AND OTHER COMMENTS

The history will be deleted when the application terminates.

EXIT

ID: UC5

DESCRIPTION AND GOAL

The UC Exit, terminates the application.

ACTORS (ESP. PRIMARY ACTOR)

Analyst

PRECONDITIONS

-

BASIC FLOW

1. The UC starts when the actor selects the 5th option.
2. The system terminates.

EXTENSIONS / VARIATIONS

-

POST CONDITIONS

-

SPECIAL REQUIREMENTS, ISSUES, RISKS AND OTHER COMMENTS

-

2 ΣΧΕΔΙΑΣΗ ΕΛΕΓΧΩΝ

Οι έλεγχοι που σχεδιάσθηκαν και εντάχθηκαν στην υλοποίηση περιγράφονται παρακάτω.

2.1 ΕΛΕΓΧΟΣ USE CASES VIA SYSTEM TESTS

2.1.1 TRACEABILITY MATRIX

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

UC1	Load data
UC2	Aggregate by time unit
UC3	Report results in file
UC4	Show history
UC5	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
T1_V0_01	X				
T2_V0_01		X			
T3_V0_01			X		

Πίνακας 2 Traceability matrix between use cases and tests

2.1.2 USE CASE UC1: LOAD DATA

Involved methods

Loader.load(), MainEngine.loadData(), (MainApplication, switch case1)

Test cases

ID	T1_V0_01	HappyDayScenario for load()
Description	ON	any context
	RECEIVING	The file path of a file with the data
	ENSURE	That the System
	OUTPUTS	The correct formatting of the file
	SUCH THAT	The state is intact.
Pre-cond.		No specific precondition
Input		The file path of the file
Output		The organized data in collection
Post-cond.		The data has been loaded correctly
Method To test		`load(), loadData()`

2.1.3 USE CASE UC2: AGGREGATE BY TIME UNIT

Involved methods

`Aggregator.aggregateByTimeUnit(), MainEngine.aggregateByTimeUnit(), (MainApplication, switch case2)`

Test cases

ID	T2_V0_01	HappyDayScenario for aggregateByTimeUnit()
Description	ON	Any context
	RECEIVING	The loaded data and the aggregation type
	ENSURE	That the System
	OUTPUTS	The aggregated data on an IResult object
	SUCH THAT	The state is intact.
Pre-cond.		Data must have been loaded first.
Input		The loaded data and the aggregation function
Output		The IResult object
Post-cond.		The data has been aggregated correctly
Method To test		`aggregateByTimeUnit()`

2.1.4 USE CASE UC3: REPORT RESULTS IN FILE

Involved methods

`ResultReporter.reportResultInFile(), MainEngine.reportResultInFile(), (MainApplication,switch case3)`

Test cases

ID	T3_V0_01	HappyDayScenario for reportResultInFile()
Description	ON	A collection of records having being aggregated
	RECEIVING	The IResult object and the output location
	ENSURE	That the System
	OUTPUTS	The Final Report file
	SUCH THAT	The state is intact
Pre-cond.		Data must have been loaded and aggregated first
Input		The output directory
Output		The Final Report
Post-cond.		Return 0 and the file output location
Method To test		reportResultInFile()

2.2 UNIT TESTS

Class Loader

ON any context RECEIVING The file path of a file with the data ENSURE That the System OUTPUTS The correct formatting of the file SUCH THAT The state is intact.

Class Aggregator

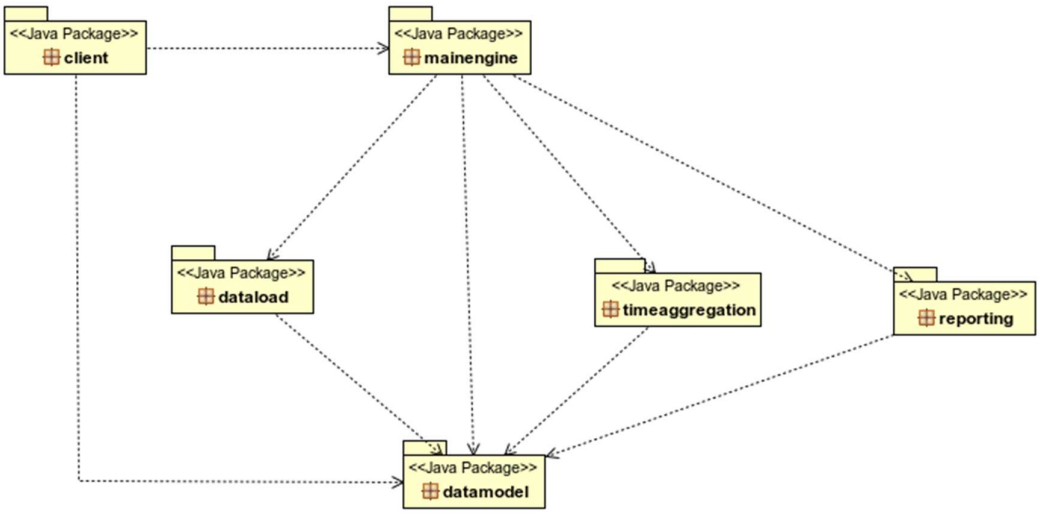
ON Any context RECEIVING The loaded data and the aggregation type ENSURE That the System OUTPUTS The aggregated data on an IResult object SUCH THAT The state is intact.

Class ResultReporter

ON A collection of records having being aggregated RECEIVING The IResult object and the output location ENSURE That the System OUTPUTS The Final Report file SUCH THAT The state is intact.

3 ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

3.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ



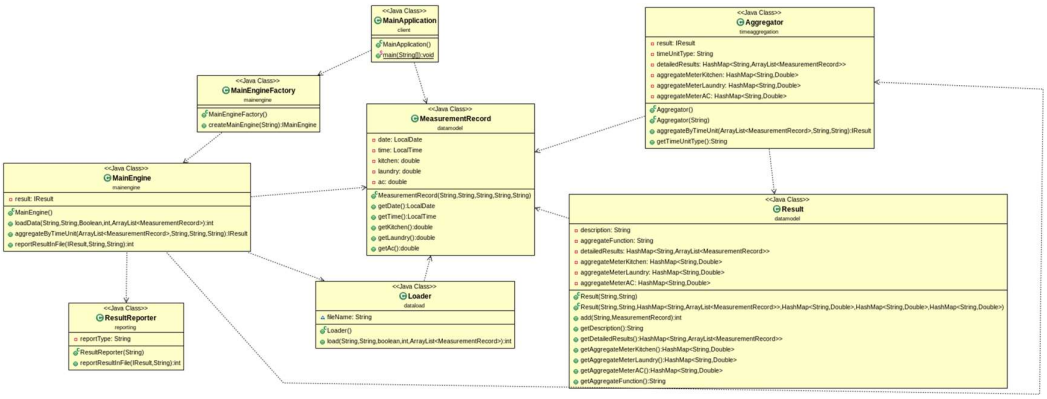
ΠΑΚΕΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

`client`	Περιέχει την boundary class που είναι υπεύθυνη για την αλληλεπίδραση με το χρήστη.
`mainengine`	Κεντρική business logic engine, με την απαραίτητη διεπαφή για εξαγωγή στην boundary class.
`dataload`	Υποσύστημα αλληλεπίδρασης με τα αρχεία δεδομένων, για την ανάκτησή τους από το σύστημα.
`timeaggregation`	Υποσύστημα ανάλυσης των δεδομένων των Domain classes.
`reporting`	Υποσύστημα παραγωγής αναφορών.
`datamodel`	Domain classes of the system.

Πίνακας 3. Συνοπτική περιγραφή πακέτων συστήματος

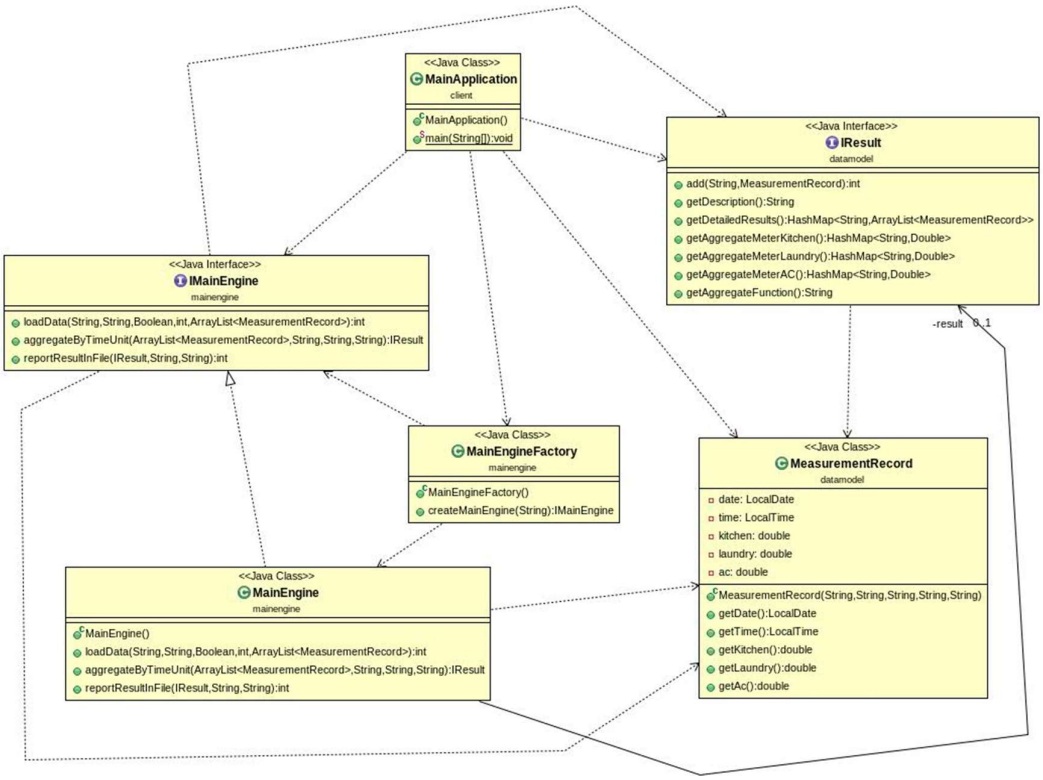
3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

Στην παρούσα υποενότητα, παρατίθενται τα διαγράμματα κλάσεων και ακολουθιών.

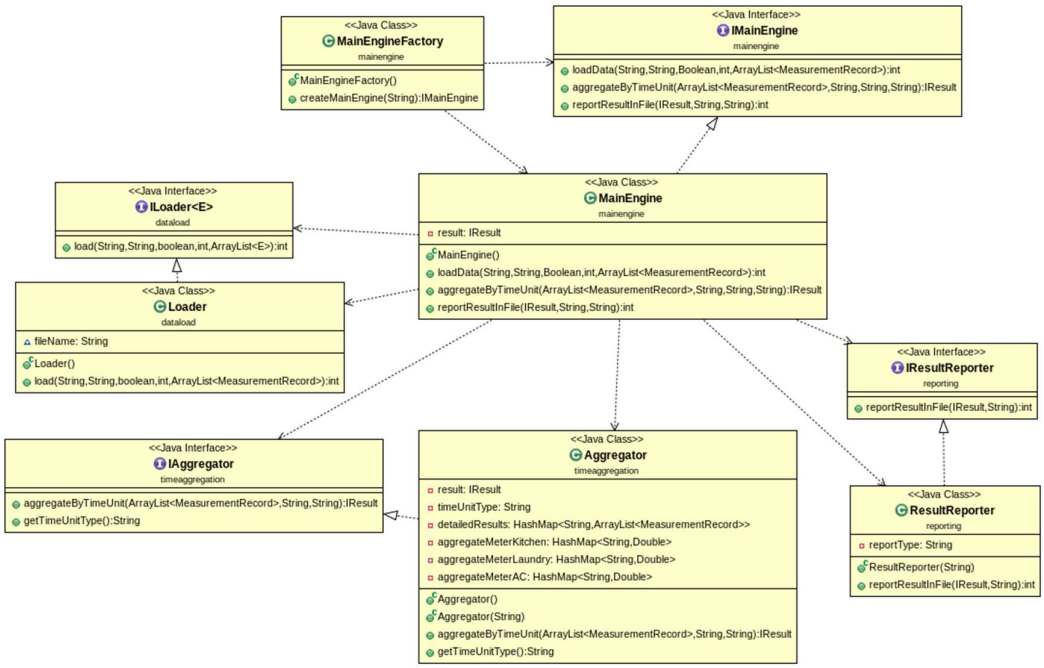


Σχήμα 1. Διάγραμμα κλάσεων για όλα τα πακέτα (παρατίθεται ευκρινέστερο διάγραμμα στο τέλος της αναφοράς)

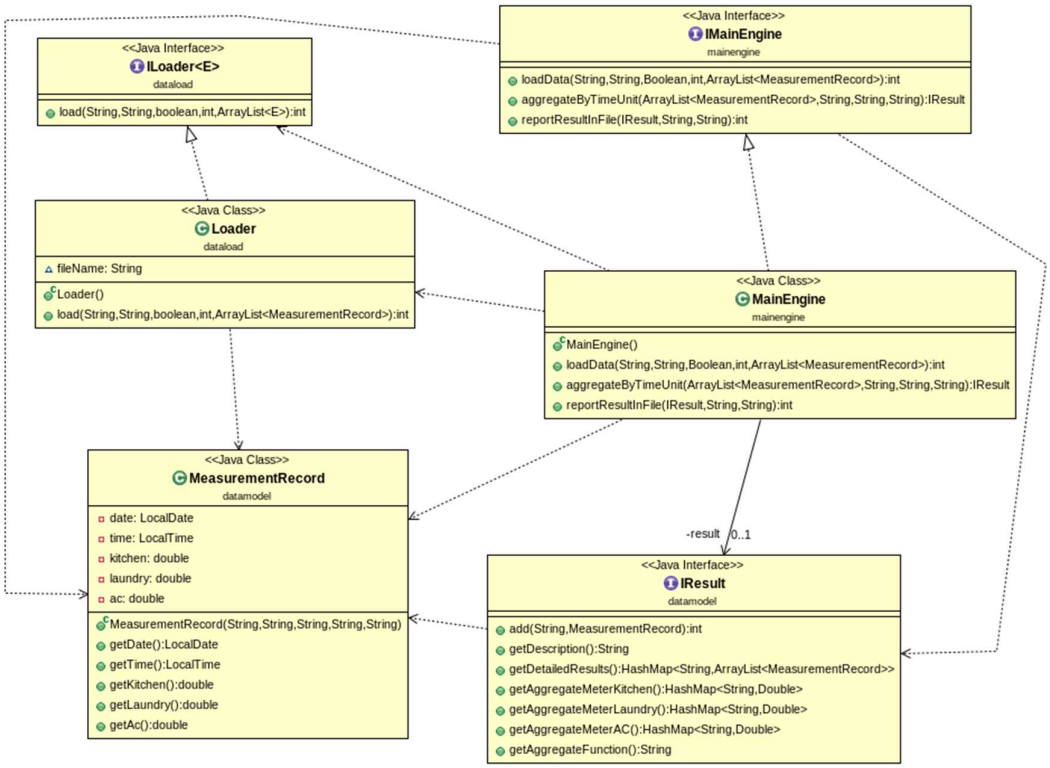
package client;
(και οι άμεσα συνεργαζόμενες κλάσεις)



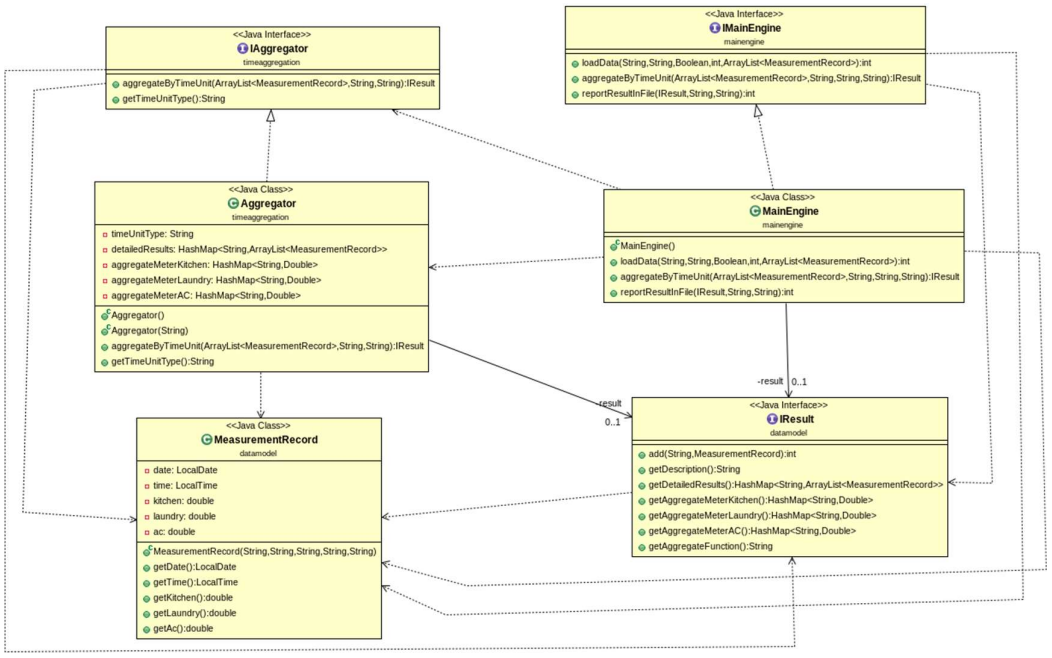
package mainengine;
(και οι άμεσα συνεργαζόμενες κλάσεις)



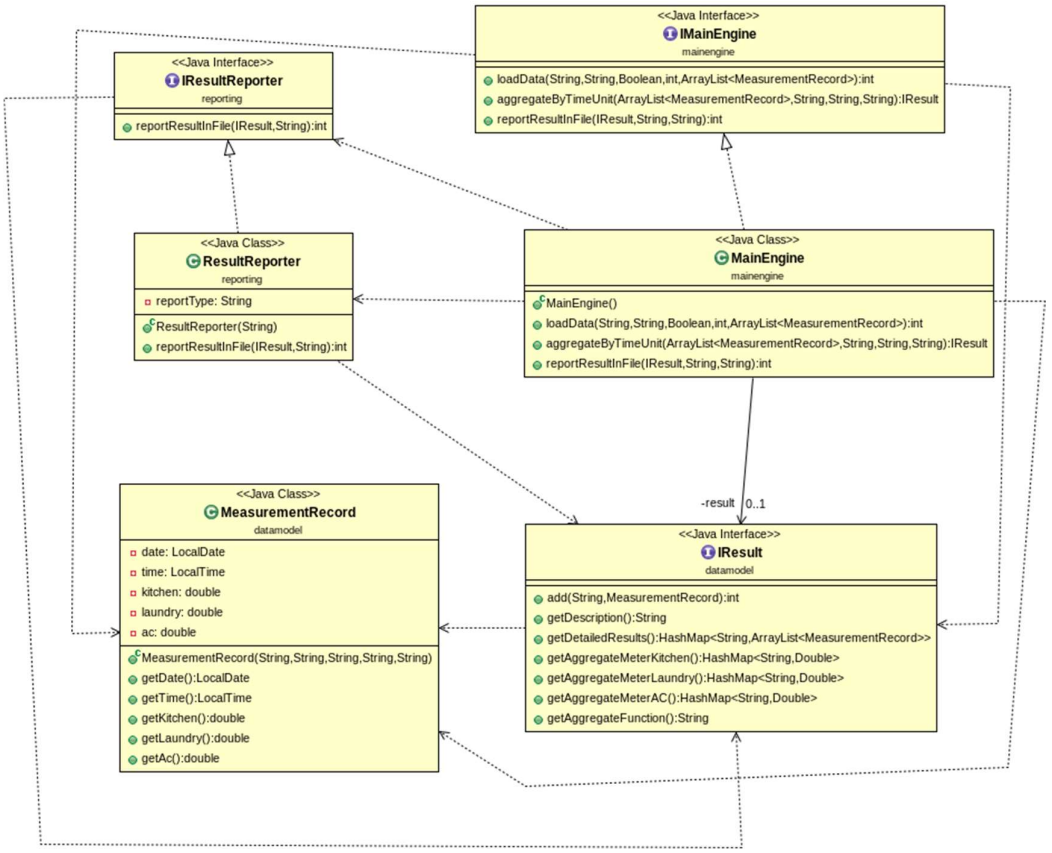
package dataload;
(και οι άμεσα συνεργαζόμενες κλάσεις)



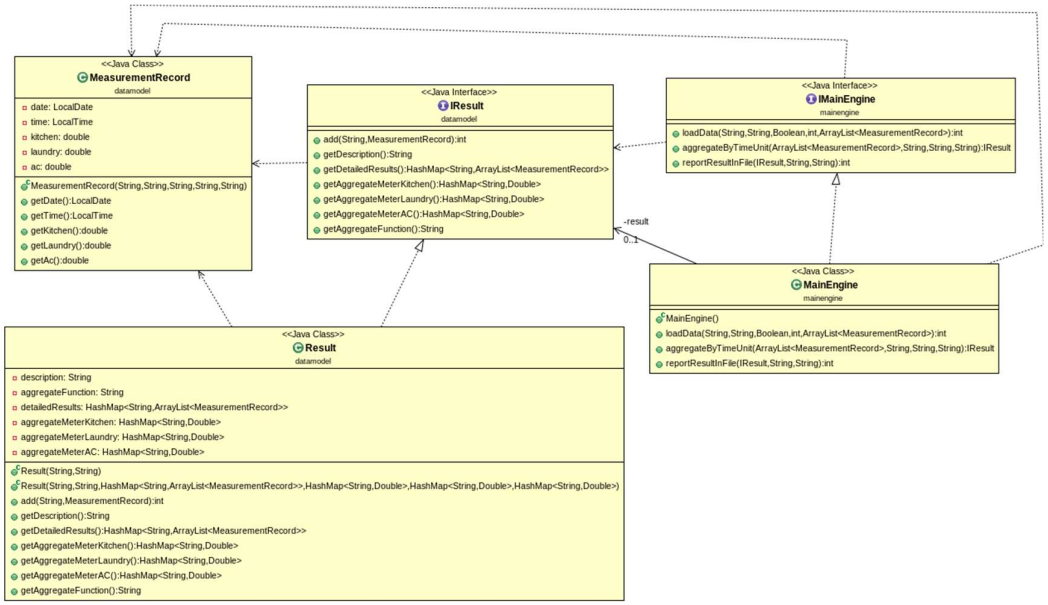
package timeaggregation;
(και οι άμεσα συνεργαζόμενες κλάσεις)



package reporting;
(και οι άμεσα συνεργαζόμενες κλάσεις)



package datamodel;
(και οι άμεσα συνεργαζόμενες κλάσεις)



3.3 ΑΝΑΛΥΣΗ ΚΛΑΣΕΩΝ ΚΑΙ ΣΥΝΕΠΕΙΑ ΠΡΟΣ ΤΙΣ ΑΠΑΙΤΗΣΕΙΣ

3.3.1 DOMAIN CLASSES

Package `datamodel`	Result, MeasurementRecord και ένα interface IResult. Οι κλάσεις αυτές θα κρατούν δεδομένα, τα αποτελέσματα της ανάλυσης του εισαχθέντος αρχείου και τα δεδομένα των γραμμών του εισαχθέντος αρχείου αντίστοιχα.

3.3.2 BUSINESS LOGIC CLASSES

Package `mainengine`	MainEngine, MainEngineFactory, για την επικοινωνία με τις άλλες business logic κλάσεις. Το Interface ImainEngine αναλαμβάνει την επικοινωνία με την boundary class.
Package `dataload`	`Loader, που αναλαμβάνει την φόρτωση του αρχείου και το Interface Iloader.`
Package `timeaggregation`	`Aggregator, που αναλαμβάνει την ανάλυση των δεδομένων που φορτώθηκαν και το Interface Iaggregator.`
Package `reporting`	`ResultReporter, που αναλαμβάνει την δημιουργία και την εξαγωγή της τελικής αναφοράς και το Interface IresultReporter.`

3.3.3 BOUNDARY CLASSES

Package `client`	`MainApplication`, μια client class για την αλληλεπίδραση με το χρήστη μέσω κονσόλας.

3.3.4 ΑΠΕΙΚΟΝΙΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΕ ΜΕΘΟΔΟΥΣ

ΑΠΕΙΚΟΝΙΣΗ USE CASES ΣΕ ΜΕΘΟΔΟΥΣ

Use case	Back-end methods	Front-end methods
LoadData	`Loader.load()`	`MainApplication.(switch-case1)`
AggregateByTimeUnit	`Aggregator.aggregateByTimeUnit()`	`MainApplication.(switch-case2)`
ReportResultsInFile	`ResultReporter.reportResultInFile()`	`MainApplication.(switch-case3)`
ShowHistory	`--- Front-end implementation ---`	`MainApplication.(switch-case4)`
Exit	`--- Front-end implementation ---`	`MainApplication.(switch-case5)`

Πίνακας 4 Επαλήθευση απεικόνισης use cases σε μεθόδους