Έγγραφο απαιτήσεων λογισμικού (SRS)

**ΟΜΑΔΑ 16**

Ειρήνη Παπαδοπούλου ΑΜ: 03115703

Νεόφυτος Παπασάββας ΑΜ: 03115702

Μηνάς Τράττου ΑΜ: 03115710

Europe Energy

# Εισαγωγή

## Εισαγωγή: σκοπός του λογισμικού

Ο σκοπός του λογισμικού είναι να παρέχει στους χρήστες του δεδομένα σχετικά με την αγορά ηλεκτρικής ενέργειας στην Ευρώπη. Ο χρήστης θα μπορεί να δει δεδομένα που αφορούν τις εξής τρεις κατηγορίες με βάση χώρα, περίοδο, τρόπο παραγωγής και ανά χρονικά διαστήματα:

* **ΑctualTotalLoad**. Περιέχει την πραγματική ενέργεια που καταναλώθηκε (φορτίο) σε μια χωρική εμβέλεια στην οποία αναφέρεται (χώρα ή ζώνη αγοράς). Οι τιμές που περιέχει (MWh) έχουν προκύψει από μετρήσεις του συστήματος διανομής ηλεκτρικής ενέργειας μετά την κατανάλωση, πχ την επόμενη μέρα.
* **DayAheadTotalLoadForecast**. Περιέχει μια πρόβλεψη για τη συνολική ζήτηση σε ενέργεια κατά την επόμενη μέρα, σε μια χωρική εμβέλεια στην οποία αναφέρεται (χώρα ή ζώνη αγοράς). Οι τιμές που περιέχει (MWh) υπολογίζονται από μαθηματικά μοντέλα πρόβλεψης.
* **AggregatedGenerationPerType**. Αφορά την ανάλυση της πραγματικής παραγωγής ενέργειας στη χώρα ή ζώνη στην οποία αναφέρεται, ανάλογα με τον τρόπο παραγωγής (θερμική, αιολική, ηλιακή, πυρηνική κ.ά.). Περιέχει τιμές (MWh) που έχουν αθροιστεί ανά τρόπο παραγωγής ηλεκτρικής ενέργειας.

Αυτό θα γίνεται μέσω ενός REST API, το οποίο θα λειτουργεί ως μεσάζοντας μεταξύ αυτών και της βάσης δεδομένων του συστήματος, ούτως ώστε να γίνεται εύκολα αυτή η διαδικασία, χωρίς να χρειάζεται ο χρήστης να γνωρίζει το πώς συνδέετε στη βάση και πως εξάγει πληροφορίες, αλλά και με ασφάλεια, με κρυπτογράφηση δεδομένων. Επίσης, για να χρησιμοποιήσει κάποιος το λογισμικό, πρέπει να έχει λογαριασμό και να συνδεθεί. Κάθε χρήστης έχει συγκεκριμένο αριθμό quotas που του επιτρέπουν να ζητήσει τιμές από τη βάση, και ανανεώνονται κάθε μέρα.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

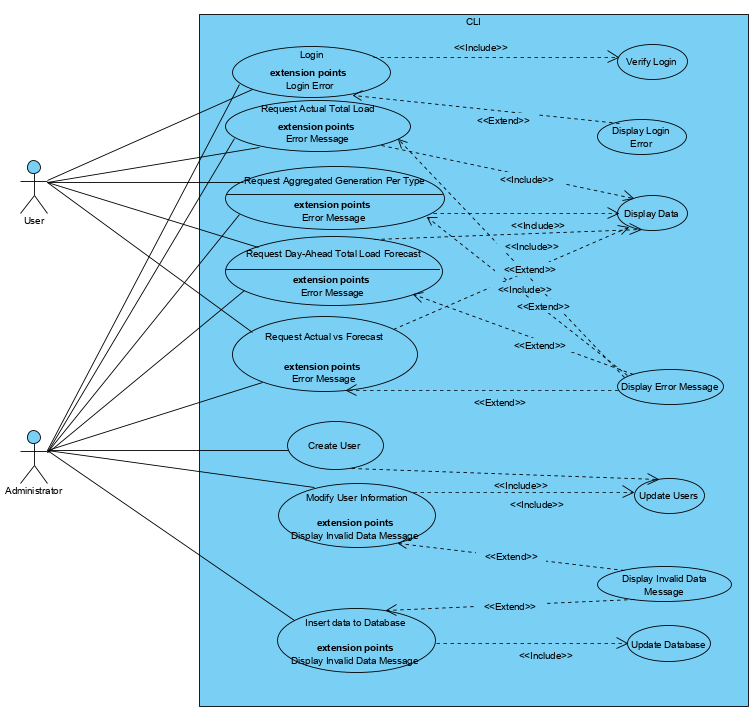
Δημιουργήσαμε τα αντίστοιχα διαγράμματα UML, βρίσκονται στο **SoftwareEngineering2019\_group16.vpp**

### 1.2.2 Διεπαφές με το χρήστη

Υπάρχουν 2 χρήστες, ο κοινός χρήστης και ο διαχειριστής. Και οι 2 χρήστες αλληλοεπιδρούν με το CLI (Command Line Interface) και ανάλογα με το ποιος είναι συνδεδεμένος, έχει και τα ανάλογα δικαιώματα.

Ο κοινός χρήστης, αφού συνδεθεί με το λογαριασμό του, μπορεί να ζητήσει από το σύστημα πληροφορίες, και το σύστημα αφού επεξεργαστεί το αίτημα, θα του απαντήσει είτε με τις πληροφορίες που ζητήθηκαν, είτε με το αντίστοιχο μήνυμα λάθους.

Ο διαχειριστής, αφού συνδεθεί με το λογαριασμό του, μπορεί να κάνει αυτά που κάνει και ο κοινός χρήστης, αλλά επιπλέον μπορεί να προσθέσει/τροποποιήσει/εμφανίσει τους κοινούς χρήστες. Έχει τη δυνατότητα επίσης να προσθέσει δεδομένα στη βάση.



# Αναφορές - πηγές πληροφοριών

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: **Ο χρήστης ζητά δεδομένα από Actual Total Load**

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Ο κοινός χρήστης, το CLI, η βάση δεδομένων και το REST API.

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

Ο server πρέπει να τρέχει για να απαντήσει σε αιτήματα. Επίσης ο χρήστης πρέπει να είναι ήδη εγγεγραμμένος στο σύστημα από τον διαχειριστή που είναι ο μόνος που μπορεί να δημιουργήσει νέο χρήστη.

#### 3.1.1.3 Περιβάλλον εκτέλεσης

Ο κοινός χρήστης, το CLI, η βάση δεδομένων και το REST API.

#### 3.1.1.4 Δεδομένα εισόδου

Είσοδος: username, password, η εντολή με τις παραμέτρους για τα δεδομένα που ζητά (της μορφής energy\_group16 ActualTotalLoad –area areaValue –timeres timeresValue – date dateValue και το format(optional))

Έξοδος: Τα δεδομένα που ζήτησε ο χρήστης αν υπάρχουν (σε μορφή json), διαφορετικά κενό (πχ json object αν αυτό όρισε στο format). Επίσης το token που επιστρέφει το σύστημα όταν κάνει Login ένας χρήστης, αποθηκευμένο σε ένα αρχείο με όνομα “softeng19bAPI.token”.

*Συνθήκες εγκυρότητας δεδομένων:*

username (αλφαριθμητικό, unique)

password (String χωρίς κενά, unique)

areaValue (Οι distinct τιμές του πεδίου AreaName στο αντίστοιχο dataset)

timeresValue (PT15M or PT30M or PT60M)

dateValue (YYYY-MM-DD)

format (json, csv)

#### Παράμετροι

Καταγράφονται οι παράμετροι που θα χρησιμοποιηθούν στο CLI:

--username –passw (για να κάνει Login o χρήστης)

--area --timeres --date (για την εκτέλεση της εντολής και απεικόνιση των δεδομένων)

#### Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

**Βήμα 1**: Ο χρήστης κάνει πρώτα login. Σε οποιαδήποτε άλλη περίπτωση που δώσει κάποια εντολή ενώ είναι εκτός σύνδεσης, παίρνει μήνυμα λάθους “401 not authorized”.

**Βήμα 2:** O χρήστης δίνει το username και το password του. Τα στοιχεία προωθούνται από το CLI στο RestAPI και ελέγχονται μέσω του DBMS. Εάν τα στοιχεία είναι λάθος, ο χρήστης λαμβάνει μήνυμα λάθους “401 Not Authorized”. Εάν τα στοιχεία είναι σωστά, ο χρήστης λαμβάνει token στη μορφή {"token":"JXfEVaipTnC4LAP8MZSW"}. Το token δημιουργείται από το RestAPI.

**Βήμα 3**: Ο χρήστης δίνει την εντολή για Actual Total Load μαζί με τις παραμέτρους. Το CLI παίρνει την εντολή αυτή, δημιουργεί ένα URL και το στέλνει στο RestAPI. Στο header αυτού του url με όνομα X-OBSERVATORY-AUTH βρίσκεται και το token του χρήστη.

**Βήμα 4**: Το RestAPI ελέγχει εάν υπάρχει token και εάν το token ανήκει στο συγκεκριμένο χρήστη.

**!!!!!!!!!!!!!Βήμα 5**: Ελέγχεται η τελευταία ημερομηνία που συνδέθηκε ο χρήστης και ενημερώνεται η τιμή του quotas αναλόγως στο DBMS.

**Βήμα 6**: Γίνεται έλεγχος στο RestAPI αν ο χρήστης έχει ξεπεράσει τον αριθμό των κλήσεων που δικαιούται (ο αριθμός των quotas). Αν τις έχει ξεπεράσει το CLI παίρνει μήνυμα “402 out of quotas” και το εμφανίζει στο χρήστη. Αν δεν έχουν ξεπεραστεί, γίνεται έλεγχος από το CLI ότι η εντολή είναι σωστά δομημένη μαζί με τις παραμέτρους της και δημιουργείται url.

**Βήμα 7**: Στη συνέχεια, αν είναι σωστά τα δεδομένα που δόθηκαν από το χρήστη, ελέγχεται αν υπάρχουν δεδομένα στη βάση. Το RestAPI δίνει στο DBMS date, area name, time resolution και το DBMS επιστρέφει τα δεδομένα που ζητήθηκαν. Αν δεν υπάρχουν δεδομένα για να εμφανιστούν στο χρήστη, τότε το RestAPI επιστρέφει κενό (πχ json object αν το format = json), το ποιο εμφανίζεται στο χρήστη. Αν υπάρχουν τα δεδομένα, τότε επιστρέφονται στο χρήστη (σε μορφή json ή csv, ανάλογα με το τι ζήτησε).

**Βήμα 8**: Ενημερώνεται ο αριθμός των quotas του χρήστη από το RestAPI.

Tα activity και sequence UML διαγράμματα βρίσκονται στο αρχείο **SoftwareEngineering2019\_group16.vpp** με όνομα «Sequence Diagram for Actual Total Load (1η Περίπτωση)» και «Activity Diagram for Actual Total Load (1η Περίπτωση)».

#### Δεδομένα εξόδου

Τo UML sequence διάγραμμα για τα δεδομένα εξόδου με όνομα «Sequence Diagram για δεδομένα εξόδου (1η Περίπτωση)» βρίσκεται στο αρχείο **SoftwareEngineering2019\_group16.vpp**

#### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: **Εγγραφή νέου χρήστη από τον Admin**

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Ο κοινός χρήστης, το CLI, η βάση δεδομένων, το REST API και ο Admin.

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Ο server πρέπει να τρέχει για να απαντήσει σε αιτήματα.

#### 3.1.2.3 Περιβάλλον εκτέλεσης

Το CLI, η βάση δεδομένων και το REST API.

#### 3.1.2.4 Δεδομένα εισόδου

Είσοδος: Username(αλφαριθμητικό, unique), password(String χωρίς κενά, unique), email(String διεύθυνση email, unique), quotas(int) του νέου χρήστη

Έξοδος: token του νέου χρήστη

#### 3.1.2.5 Παράμετροι

Καταγράφονται οι παράμετροι που θα χρησιμοποιηθούν στο CLI:

--username --passw (για να κάνει Login o Admin)

--newuser --passw –email --quota (για την προσθήκη νέου χρήστη)

#### 3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

**Βήμα 1**: Ο admin κάνει πρώτα login. Σε οποιαδήποτε άλλη περίπτωση που δώσει κάποια εντολή ενώ είναι εκτός σύνδεσης, παίρνει μήνυμα λάθους “401 not authorized”.

**Βήμα 2:**. Τα στοιχεία προωθούνται από το CLI στο RestAPI και ελέγχονται μέσω του DBMS. Εάν τα στοιχεία είναι λάθος, ο Admin επιστρέφει στην προηγούμενη σελίδα για να βάλει τα σωστά στοιχεία. Εάν τα στοιχεία είναι σωστά, ο χρήστης λαμβάνει token το οποίο αποθηκεύεται στο αρχείο “softeng19bAPI.token”. Το token δημιουργείται από το RestAPI.

**Βήμα 3**: O Admin δίνει την εντολή για να δημιουργήσει νέο χρήστη και τις κατάλληλες παραμέτρους (username, password, email, quotas), δημιουργείται ένα URL και στέλνεται στο RestAPI. Το RestAPI ελέγχει εάν υπάρχει token και εάν το token ανήκει στον Admin. Εάν όχι, εμφανίζεται στο χρήστη το μήνυμα “401 not authorized”.

**Βήμα 4**: Γίνεται έλεγχος από το RestAPI ότι η εντολή είναι σωστά δομημένη μαζί με τις παραμέτρους της. Αν δεν είναι σωστή, το RestAPI δίνει στο CLI το μήνυμα “400 bad request” το οποίο εμφανίζεται στον Admin.

**Βήμα 6**: Στη συνέχεια, αν είναι σωστή η εντολή που δόθηκε από το Admin, ελέγχεται αν υπάρχει ήδη ο χρήστης . Το RestAPI δίνει στο DBMS username, password, email, quotas και δημιουργείται ο νέος χρήστης.

Tα activity και sequence UML διαγράμματα βρίσκονται στο αρχείο **SoftwareEngineering2019\_group16.vpp** με όνομα «Sequence Diagram for Creating New User (2η Περίπτωση)» και «Activity Diagram for Creating New User (2η Περίπτωση)».

#### 3.1.2.7 Δεδομένα εξόδου

Το UML sequence διάγραμμα με όνομα «Sequence Diagram για δεδομένα εξόδου (2η Περίπτωση)» βρίσκεται στο αρχείο **SoftwareEngineering2019\_group16.vpp.**

## 3.2 Απαιτήσεις επιδόσεων

Το σύστημα θα απαντά σε κάθε αίτημα του χρήστη σε μικρό χρονικό διάστημα. Επίσης οι απαιτήσεις συστήματος (υπολογιστή) θα είναι πολύ λίγες, ούτως ώστε να τρέχει σε κάθε σύστημα. Τέλος το σύστημα μπορεί να έχει μόνο ένα χρήστη συνδεδεμένο.

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Ο κοινός χρήστης μπορεί να ζητήσει να δει δεδομένα μόνο αν είναι συνδεδεμένος στο σύστημα με τον λογαριασμό του και έχει διαθέσιμα ημερήσια quotas. Δεν έχει δικαίωμα να προσθέσει/διαγράψει/τροποποιήσει άλλους χρήστες, ούτε να δει ποιοι είναι εγγεγραμμένοι. Επίσης δεν έχει δικαίωμα να προσθέσει/διαγράψει/τροποποιήσει κάποια εγγραφή στη βάση δεδομένων.

Ο διαχειριστής μπορεί να κάνει ότι και ο χρήστης. Μπορεί επιπλέον να προσθέσει/τροποποιήσει χρήστες στη βάση δεδομένων και να δει τους ήδη υπάρχοντες. Επίσης μπορεί να προσθέσει δεδομένα στη βάση στους πίνακες ActualTotalLoad, AggregatedGenerationPerType και DayAheadTotalLoadForecast.

## 3.4 Περιορισμοί σχεδίασης

* Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Java με τη χρήση του spring framework.
* Η γλώσσα των διεπαφών χρήστη στην εφαρμογή CLI είναι η αγγλική
* Για την γλώσσα Java θα γίνει χρήση του εργαλείου build automation Gradle.
* Το back-end υποσύστημα πρέπει να παρέχει κατάλληλο REST API για τη διασύνδεσή του τις υπόλοιπες εφαρμογές.
* Η πρόσβαση στο REST API θα πρέπει να γίνεται με λογαριασμούς χρηστών. Κάθε λογαριασμός χρήστη περιέχει όριο χρήσης (quota) ώστε να περιορίζεται η κατανάλωση πόρων του API.
* Πρέπει να υποστηρίζεται το πρωτόκολλο HTTPS για όλες τις χρηστικές ή προγραμματιστικές διεπαφές μέσω self-signed certificate.
* Η ονοματολογία οντοτήτων δεδομένων και πεδίων καθορίζεται μαζί με τη βάση δεδομένων που δίνεται έτοιμη.
* Το RestAPI επιστρέφει μόνο τους μορφότυπους .json και .csv για τα δεδομένα.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Το λογισμικό θα είναι διαθέσιμο σε όλο το κοινό. Όλοι μπορούν να δημιουργήσουν λογαριασμό. Ο μόνος περιορισμός είναι το ότι το λογισμικό θα τρέχει σε τοπικό server, οπότε δε θα είναι διαθέσιμο στο διαδίκτυο.

### 3.5.2 Απαιτήσεις ασφάλειας

Τα δεδομένα θα μεταφέρονται κρυπτογραφημένα, για αποφυγή υποκλοπής δεδομένων των χρηστών. Επίσης οι αναζητήσεις που έκανε ο κάθε χρήστης δε θα είναι διαθέσιμες παρά μόνο στους ίδιους χρήστες, ούτως ώστε να προστατεύονται οι χρήστες από άλλους με διαφορετικά συμφέροντα.

### 3.5.3 Απαιτήσεις συντήρησης

Το λογισμικό πρέπει να αντιμετωπίζει τυχόν λάθος εισόδους από τους χρήστες επί τόπου με αντίστοιχα μηνύματα λάθους. Σε περίπτωση κάποιου λάθους που θα προκαλέσει βλάβη στο σύστημα, θα πρέπει η επιδιόρθωσή του να μπορεί να γίνει άμεσα, λόγω της δομής της υλοποίησης του κώδικα που είναι ευανάγνωστος.