Έγγραφο απαιτήσεων λογισμικού (SRS)

ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΕΓΓΡΑΦΟΥ ΤΟΥ ΠΡΟΤΥΠΟΥ ISO/IEC/IEEE 29148:2011

**ENERGY DATA PLATFORM**

# Εισαγωγή

## Εισαγωγή: σκοπός του λογισμικού

Σκοπός του λογισμικού μας είναι η παροχή πληροφοριών σε χρήστες άμεσα ενδιαφερόμενους για την παραγωγή ενέργειας.

Πιο συγκεκριμένα ενημερώνονται οι εταιρείες παραγωγής ενέργειας, οι ιδιώτες παραγωγοί, οι σχετικοί κρατικοί οργανισμοί και οποιοσδήποτε άλλος επιθυμεί να αντλήσει δεδομένα σχετικά με την κατανάλωση και την παραγωγή ενέργειας για προσωπική χρήση (δηλαδή το απλό κοινό όπως φοιτητές).

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

Το λογισμικό μας δεν έχει κάποια διεπαφή με εξωτερικά συστήματα, δεδομένου ότι δεν παίρνει απευθείας δεδομένα από τον ιστότοπο του entso-e αλλά έχει πρόσβαση σε μία σχετική βάση δεδομένων (μέσω των datasets) που του παρέχει τις πληροφορίες που χρειαζόμαστε.

### 1.2.2 Διεπαφές με το χρήστη

Προδιαγραφή διεπαφών με το χρήστη. Μοντέλο Use Case (UML).

Έχουμε μία τέτοια διεπαφή με τον χρήστη ώστε να μπορούμε να παρέχουμε διαφορετικές δυνατότητες (περιγράφονται παρακάτω) ανάλογα με το ποιος χρήστης έχει εισέλθει στο σύστημά μας.

# Αναφορές - πηγές πληροφοριών

<https://transparency.entsoe.eu/>

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

* Χρήση από administrator- Εγγραφή χρηστών
* Χρήση από administrator και εγγεγραμμένο χρήστη-πρόσβαση στα δεδομένα
* Χρήση από administrator- Ενημέρωση χρηστών

### **3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1:**

### **Administrator-Εγγραφή χρηστών**

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Η περίπτωση χρήσης αυτή αφορά μόνο τον χρήστη που είναι ο administrator του συστήματος. Ο administrator μπορεί να εκτελέσει εντολές στο CLI αλλά και στο front-end (σε κατάλληλα διαμορφώμενη σελίδα στην όποια έχει πρόσβαση μόνο αυτός) και να πραγματοποιήσει εισαγωγή χρηστών στη βάση.

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

Πρέπει να υπάρχουν στην βάση δεδομένα, να είναι ενεργή η σύνδεση του δικτύου και να είναι δυνατή η σύνδεση στο local host. Επίσης, ο χρήστης πρέπει να αναγνωρίζεται από το σύστημα ότι είναι ο administrator.

#### 3.1.1.3 Περιβάλλον εκτέλεσης

Η εγγραφή των χρηστών από τον administrator γίνεται από το CLI καθώς και από το front-end μέσω κατάλληλων κλήσεων στο REST-API του back-end .

#### 3.1.1.4 Δεδομένα εισόδου

O administrator για να εγγράψει κάποιον χρήστη στη βάση δεδομένων του συστήματος μέσω του CLI πρέπει να εισάγει την παράμετρο newuser με τιμή το username του και να δώσει ως παραμέτρους το password, το email και κάποιο quota που θα τον αφορά.

O administrator για να εγγράψει κάποιον χρήστη στη βάση δεδομένων του συστήματος μέσω του front-end πρέπει να επιλέξει την κατάλληλη σελίδα για να μεταβεί ώστε να εισάγει τα στοιχεία του χρήστη που θέλει να εγγράψει, δηλαδή το username, το password, το email και το quota που θα του αντιστοιχούν.

#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Για το CLI:

Βήμα 1

Γράφει την παράμετρο newuser.

Βήμα 2

Εισάγει σαν τιμή της παραμέτρου το username του χρήστη που θέλει να εγγράψει.

Βήμα 3

Στη συνέχεια δίνονται ως παράμετροι το password, το email και το quota.

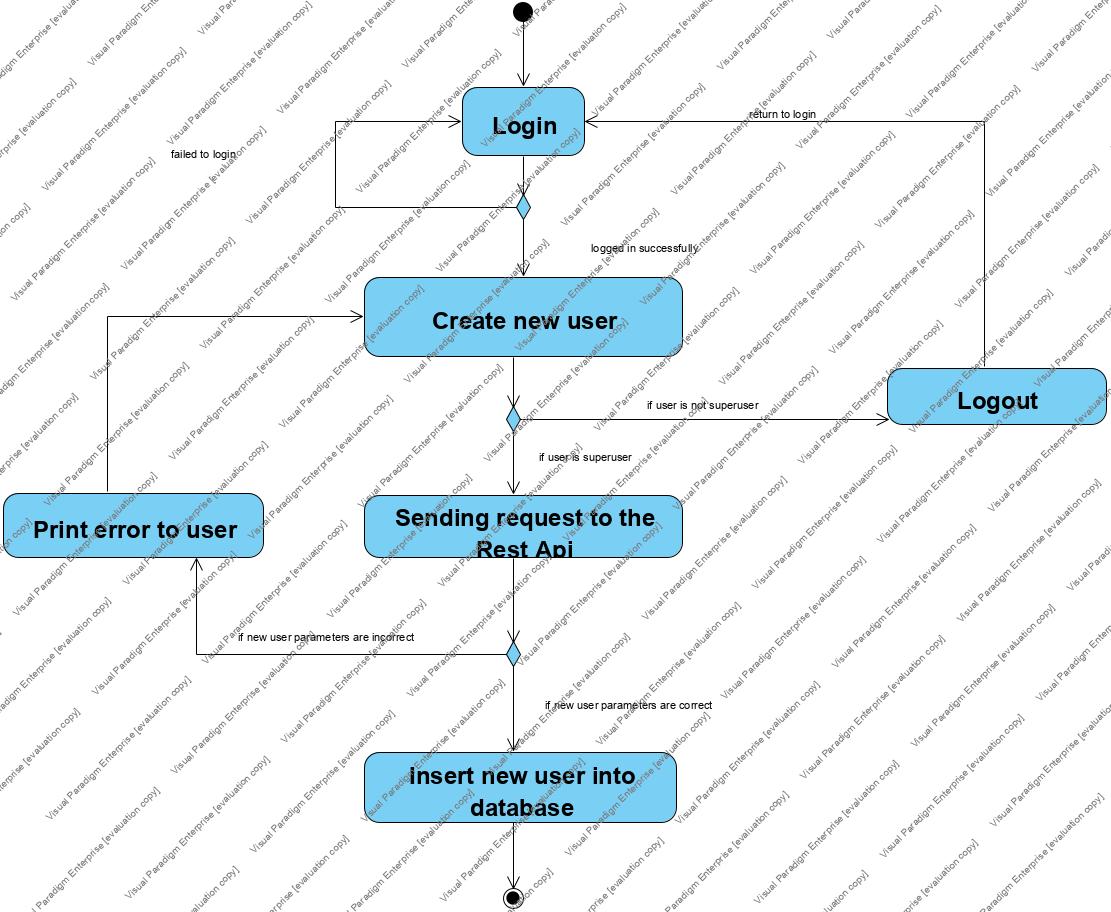
Βήμα 4

Δημιουργείται ένας νέος χρήστης.

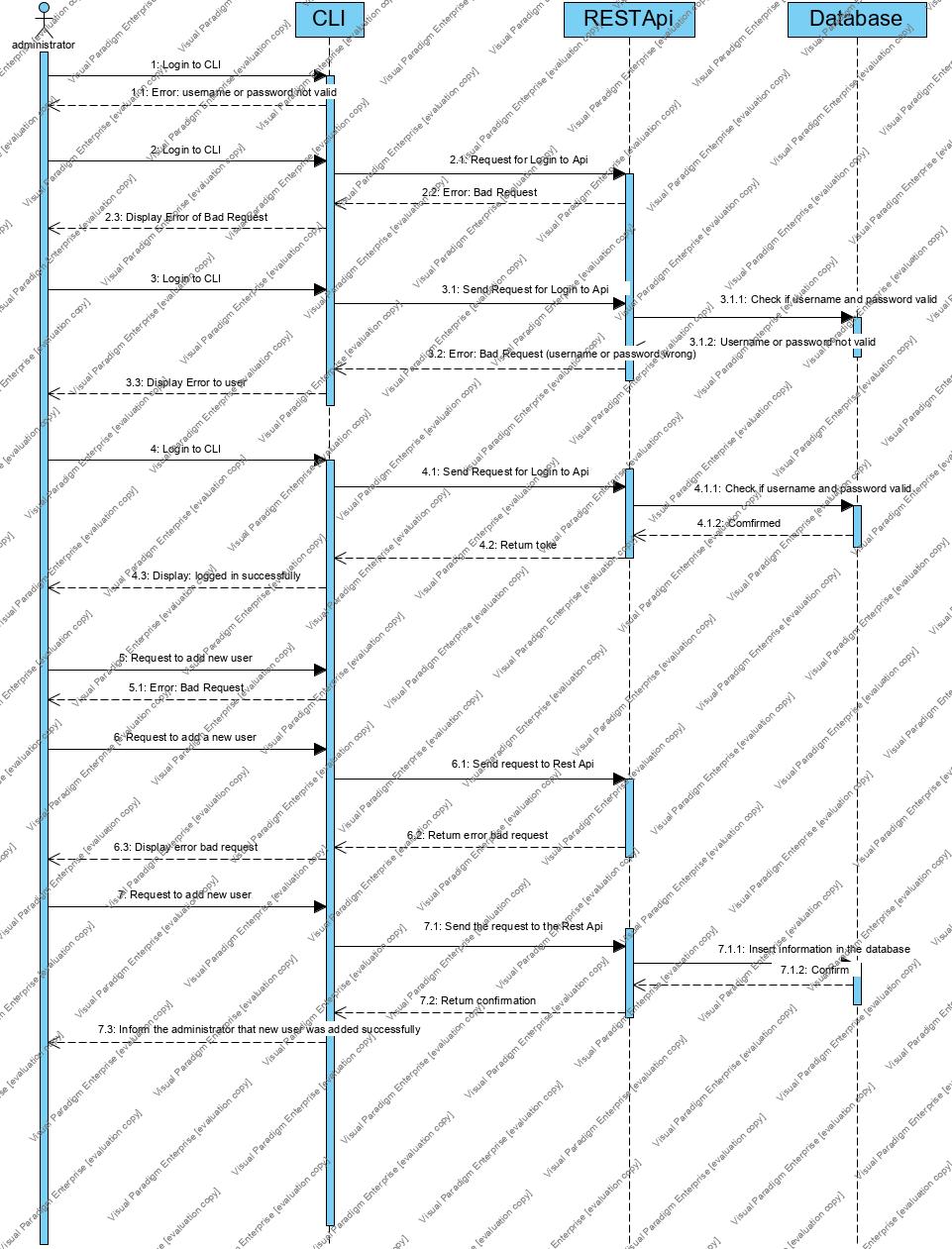
Βήμα 5

Αποτέλεσμα της δημιουργίας είναι ένα API key.

**Activity diagram**



**Sequence diagram**



#### 3.1.1.7 Δεδομένα εξόδου

Με τη διαδικασία εισαγωγής ένος νέου χρήστη δίνεται γι αυτόν ένα API key. Έτσι, εισάγεται ένας νέος χρήστης στο σύστημα, συνεπώς αν αυτός επιχειρήσει να συνδεθεί τώρα θα βρεθεί στο σύστημα κι άρα θα έχει πρόσβαση στα δεδομένα. Διαφορετικά δεν θα βρισκόταν στη βάση και θα του εμφανιζόταν κατάλληλο μήνυμα λάθους κι άρα δεν θα είχε δυνατότητα πρόσβασης.

#### 3.1.1.8 Παρατηρήσεις

Το API key που δημιουργείται κατά την εγγραφή ενός χρήστη είναι αυτό που καθορίζει την ταυτοτητά του.

Εναλλακτικά ο administrator στο CLI μπορεί να γράψει την παράμετρο moduser με τη τιμή του username και να τροποποιήσει κάποιες από τις λοιπές παραμέτρους. Στην περίπτωση αυτή δεν επιστρέφεται κάποιο API key.

Επίσης, ο administrator στο CLI μπορεί να ζητήσει να του εμφανιστουύν οι πληροφορίες που συνοδεύουν ένα χρήστη, δηλαδή την κατάσταση που βρίσκεται. Αυτό γίνεται μέσω της παραμέτρου userstatus με τιμή το username του χρήστη και αποτέλεσμα το username, το email, το API key και τα quotas.

Τις δύο παραπάνω ενέργειες μπορεί να τις πραγματοποιήσει ο administrator και μέσω του front-end εφόσον μεταβεί στις αντίστοιχες σελίδες. Για την τροποποίηση των στοιχείων ενός χρήστη εισάγει το username του καθώς και το email και το quota που πιθανόν να τροποιήσουν. Για την πρόσβαση στην κατάσταση κάποιου χρήστη αρκεί να δοθεί το username του κι έτσι θα εμφανιστούν όλα τα δεδομένα που τον αφορούν.

### 3.1.2 **ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2:**

### **Administrator/Εγγεγραμένος χρήστης- Πρόσβαση στα δεδομένα**

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Η συγκεκριμένη περίπτωση χρήσης αφορά όλους τους χρήστες που έχουν πρόσβαση στην εφαρμογή μας. Καθένας από τους χρήστες μορεί να κάνει log in, ώστε να συνδεθεί και να μπορέσει να λάβει δεδομένα αναζητώντας τα από τους τρεις πίνακες. Επίσης, μπορέι στην συνέχεια να κάνει log out για να αποσυνδεθεί από τον “λογαριασμό” του και να εξέλθει από το σύστημά μας.

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Πρέπει να υπάρχουν στην βάση δεδομένα, να είναι ενεργή η σύνδεση του δικτύου και να είναι δυνατή η σύνδεση στο local host. Επίσης, ο χρήστης πρέπει να αναγνωρίζεται από το σύστημα και να του επιστρέφεται το αντίστοιχο token που του δίνει την πρόσβαση στην εφαρμογή και άρα στα στα δεδομένα.

#### 3.1.2.3 Περιβάλλον εκτέλεσης

Η είσοδος στο σύστημα για την λήψη δεδομένων γίνεται από το CLI και από το front-end αυτού μέσω κατάλληλων κλήσεων στο REST-API του back-end ..

#### 3.1.2.4 Δεδομένα εισόδου

Ο κάθε χρήστης που εισέρχεται στο σύστημα έχει πρόσβαση σε δεδομένα όπως είναι το ActualTotalLoad. Ενδιαφέρεται για συγκεκριμένα τέτοια δεδομένα ανάλογα με τη χώρα και τη χρονική περίδοδο ενδιαφέροντος. Συνεπώς, σαν είσοδος δίνεται ο πίνακας που επιθυμέιται να προσπελαστεί, εδώ το ActualTotalLoad, και συνοδεύεται από τις αντίστοιχες παραμέτρους area, timeres, day, month και year που αφορούν την χώρα αναφοράς, την ημερομηνία ενδιαφέροντος καθώς και τον μορφότυπο εμφάνισης των δεδομένων, δηλαδή τη χρονική ανάλυση αυτών ανά ώρα, ανά μισή ώρα ή ανά τέταρτο αυτής.

#### 3.1.2.5 Παράμετροι

Όταν ζητούνται δεδομένα από το συγκεκριμένο dataset, τότε πρέπει η κλήση του να συνοδεύεται από ορισμένες παραμέτρους ώστε να αναγνωριστούν και να επιστραφούν τα κατάλληλα δεδομένα, διαφορετικά θα υπάρξει σφάλμα. Για παράδειγμα στο CLI η κλήση για την εμφάνιση δεδομένων με βάση τη χρονική ανάλυσή τους σε χρονοσειρά γίνεται με την παράμετρο scope και την τιμή ActualTotalLoad που συνοδεύονται από τις παραμέτρους area, timeres, day, month, year.

Στο front-end η κλήση για την εμφάνιση δεδομένων με βάση τη χρονική ανάλυσή τους σε χρονοσειρά γίνεται με τη συμπλήρωση των κατάλληλων πεδίων-παραμέτρων στην αντίστοιχη σελίδα αυτού. Έτσι, εισάγοντας για το dataset ActualTotalLoad τις τιμές που σχετίζονται με την περιοχή ενδιαφέροντος, την χρονική ανάλυση, τον χρόνο, τον μήνα και την ημέρα λαμβάνονται τα καταγγρεγραμμένα αποτελέσματα.

#### 3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Για το CLI:

Βήμα 1

Γράφει την παράμετρο scope.

Βήμα 2

Εισάγει σαν τιμή της παραμέτρου το τίτλο του πίνακα για τον οποίο ενδιαφέρεται, εδώ το ActualTotalLoad.

Βήμα 3

Στη συνέχεια δίνονται ως παράμετροι το area, το timeres, το date, το month και το year.

Βήμα 4

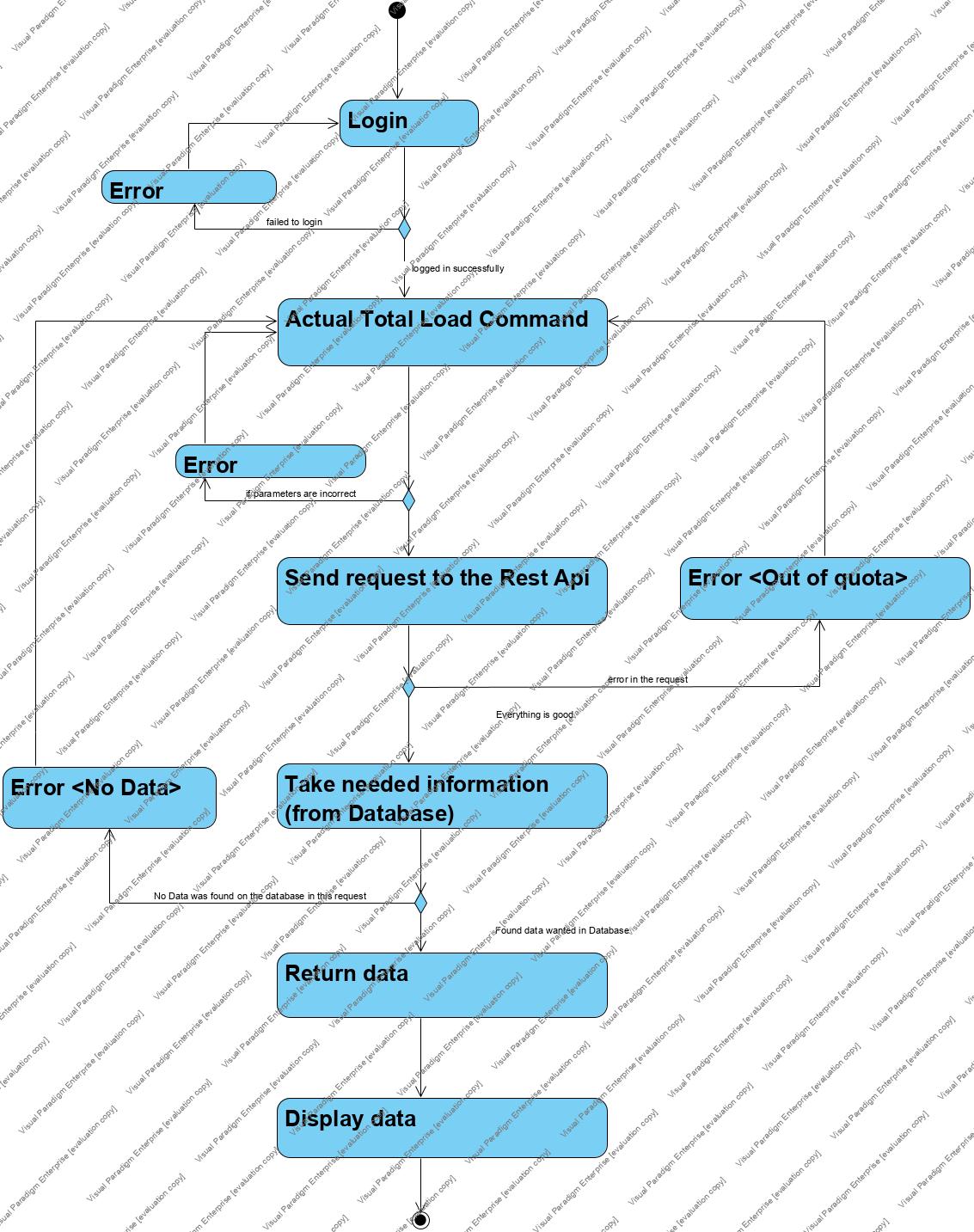
Εκτελέιται το κατάλληλο query στον κώδικα ώστε να παραχθούν τα δεδομένα που ζητούνται.

Βήμα 5

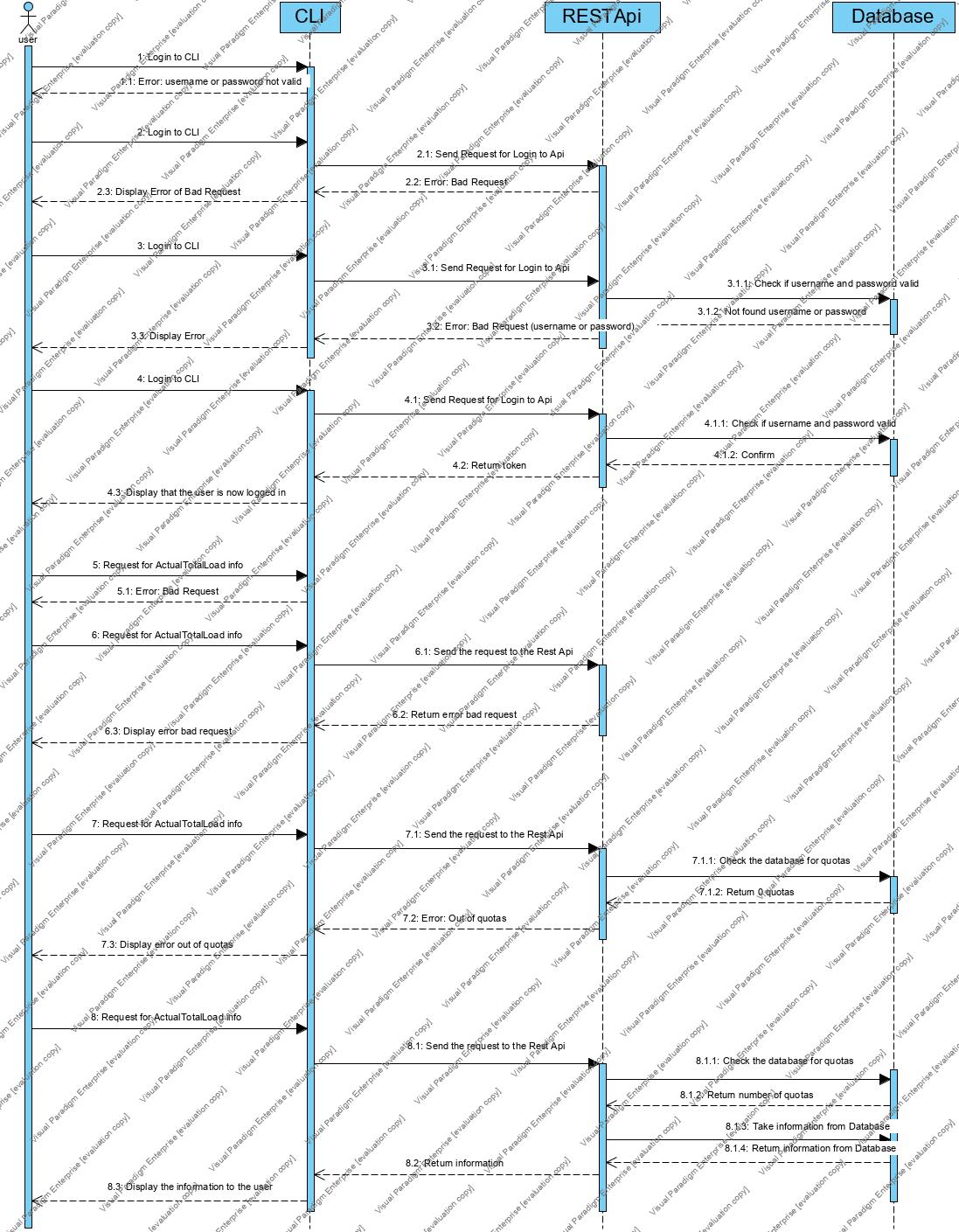
Αποτέλεσμα η χρονοσειρά με χρονική ανάλυση των καταγεγραμμένων τιμών φορτίου της περιοχής που δόθηκε ως παράμετρος για την ημέρα, τον μήνα και τον έτος που επίσης προσδιορίστηκαν από τις παραμέτρους.

Σημειώνται ότι αν δεν δοθούν οι παράμετροι της ημερομηνίας λαμβάνεαι αυτόματα η τρέχουσα.

**Activity diagram**



**Sequence diagram**



#### 3.1.2.7 Δεδομένα εξόδου

Σαν έξοδος παράγεται η χρονοσειρά με χρονική ανάλυση των καταγεγραμμένων τιμών φορτίου της περιοχής που δόθηκε ως παράμετρος για την ημέρα, τον μήνα και τον έτος που επίσης προσδιορίστηκαν από τις παραμέτρους. Σημειώνεται ότι μπορεί να δοθούν ως παράμετροι μονάχα ο χρόνος, ή ο χρόνος και ο μήνας ενδιαφέροντος οπότε και να εμφανιστούν κατάλληλα ομαδοποιημένα τα αντίστοιχα δεδομένα.

#### 3.1.2.8 Παρατηρήσεις

Όλα όσα αναφέρθηκαν παραπάνω πραγματοποιούνται για καθένα από τα τρία dataset που χρησιμοποιούμε με τιμή παραμέτρου ActualTotalLoad, AggregatedGenerationPerType, DayAheadTotalLoadForecast, ActualvsForecast συνοδευόμενη από τα κατάλληλα πεδία παραμέτρων ώστε να εμφανιστούν τα ζητούμενα δεδομένα. Για καθέναν από αυτούς τους πίνακες υποστηρίζεται η λήψη δεδομένων για κάθε μήνα ενός χρόνου, για κάθε ημέρα ενός μήνα ενός χρόνου και για κάποια ορισμένη χρονική ανάλυση κάποιας ημέρας.

### **3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3:**

### **Administrator-Ενημέρωση χρηστών**

#### 3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται

Η περίπτωση χρήσης αυτή αφορά μόνο τον χρήστη που είναι ο administrator του συστήματος. Ο administrator μπορεί να εκτελέσει εντολές στο CLI αλλά και στο front-end (σε κατάλληλα διαμορφώμενη σελίδα στην όποια έχει πρόσβαση μόνο αυτός) και να πραγματοποιήσει ενημέρωση των στοιχέιων των χρηστών στη βάση.

#### 3.1.3.2 Προϋποθέσεις εκτέλεσης

Πρέπει να υπάρχουν στην βάση δεδομένα, να είναι ενεργή η σύνδεση του δικτύου και να είναι δυνατή η σύνδεση στο local host. Επίσης, ο χρήστης πρέπει να αναγνωρίζεται από το σύστημα ότι είναι ο administrator.

#### 3.1.3.3 Περιβάλλον εκτέλεσης

Η ενημέρωση των χρηστών από τον administrator γίνεται από το CLI καθώς και από το front-end μέσω κατάλληλων κλήσεων στο REST-API του back-end .

#### 3.1.3.4 Δεδομένα εισόδου

O administrator για να ενημερώσει τα στοιχεία κάποιου χρήστη στη βάση δεδομένων του συστήματος μέσω του CLI πρέπει να εισάγει την παράμετρο moduser με τιμή το username του και να δώσει ως παραμέτρους το password, το email και κάποιο quota που θα τον αφορά, ώστε να τα τροποποιήσει.

O administrator για να ενημερώσει κάποιον χρήστη στη βάση δεδομένων του συστήματος μέσω του front-end πρέπει να επιλέξει την κατάλληλη σελίδα για να μεταβεί ώστε να εισάγει το username του χρήστη τα στοιχέια του οποίου θέλει να τροποποιήσει, καθώς και τα στοιχεία που θα ενημερώσει, δηλαδή το email και το quota.

#### 3.1.3.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Για το CLI:

Βήμα 1

Γράφει την παράμετρο moduser.

Βήμα 2

Εισάγει σαν τιμή της παραμέτρου το username του χρήστη που θέλει να ενημερώσει.

Βήμα 3

Στη συνέχεια δίνονται ως παράμετροι το password, το email και το quota, ανάλογα με το τι θέλει να αλλάξει.

Βήμα 4

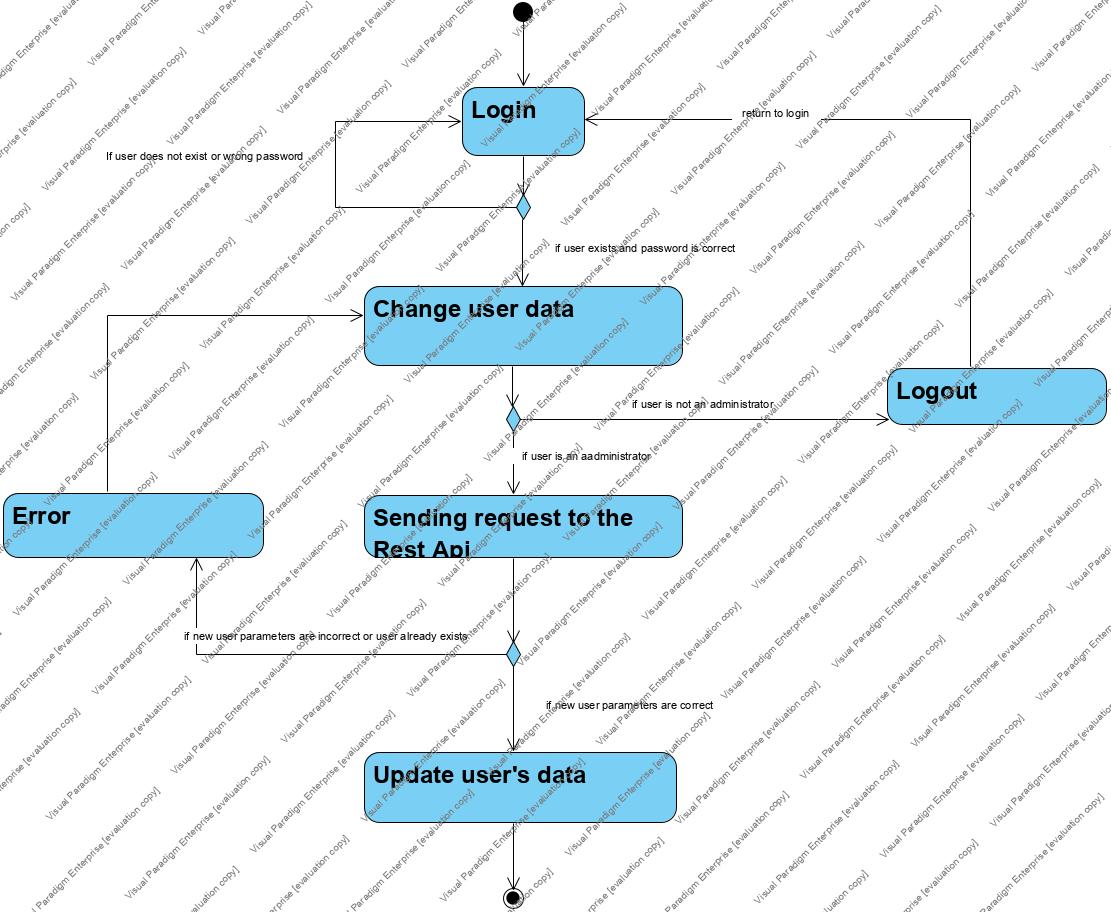
Ενημερώνονται τα στοιχεία του αντίστοιχου χρήστη.

Βήμα 5

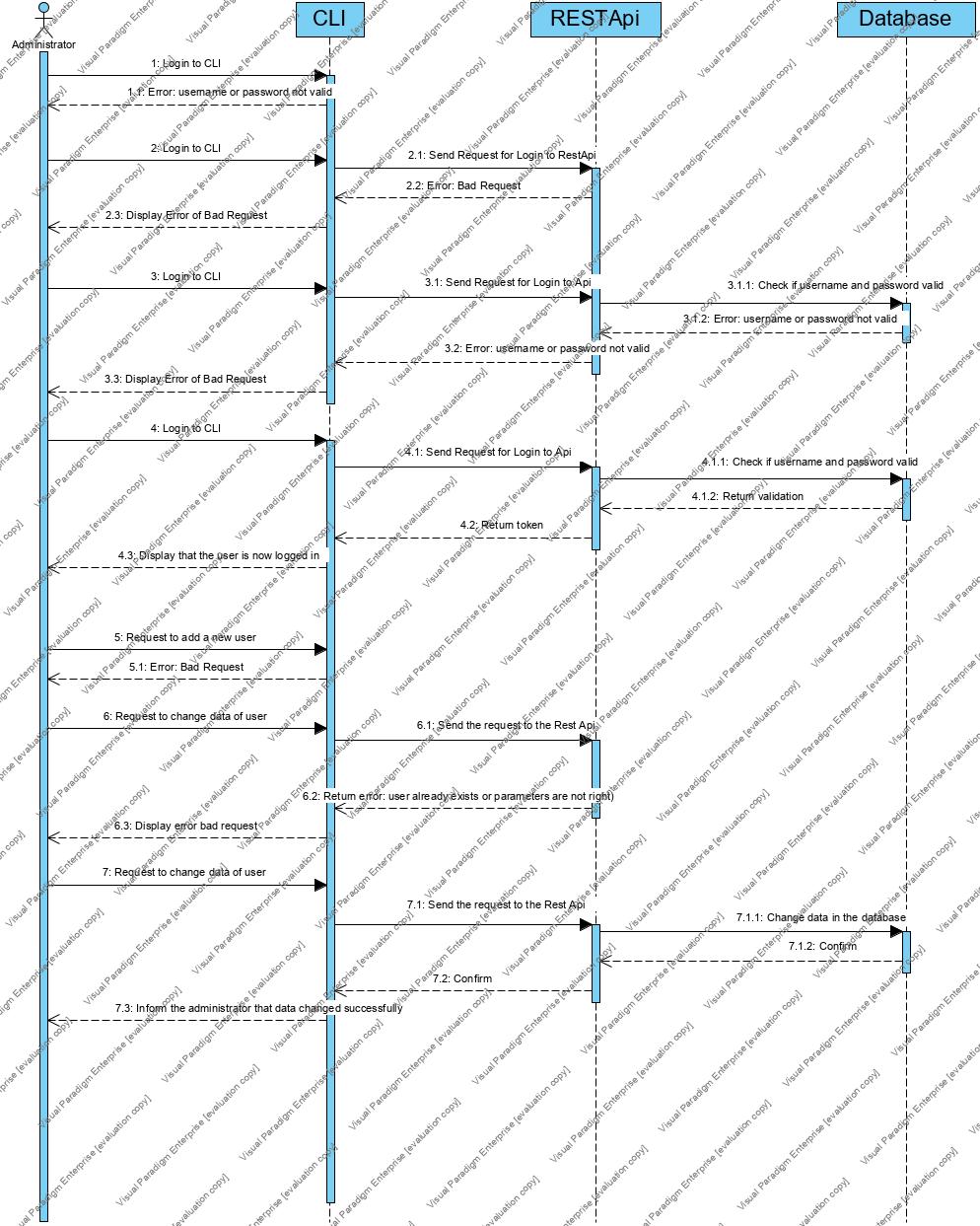
Αποτέλεσμα εκτέλεσης.

Εναλλακτικά, ο administrator μπορεί να ζητήσει να του εμφανιστουύν οι πληροφορίες που συνοδεύουν ένα χρήστη, δηλαδή την κατάσταση που βρίσκεται. Αυτό γίνεται μέσω της παραμέτρου userstatus με τιμή το username του χρήστη και αποτέλεσμα το username, το email, το API key και τα quotas.

**Activity diagram**

****

**Sequence diagram**



#### 3.1.3.7 Δεδομένα εξόδου

Με τη διαδικασία ενημέρωσης ένος χρήστη δεν επιστρέφεται κάποια έξοδος, απλά πραγματοποιείται η συγκεκριμένη ενέργεια και ολοκληρώνεται η διαδικασία της ενημέρωσης. Έτσι, για αυτόν τον χρήστη έχουν τροποποιηθεί ορισμένες, ακόμη κι όλες, οι πληροφορίες που τον συνοδεύουν για την πρόσβαση του στο σύστημα των δεδομένων.

#### 3.1.3.8 Παρατηρήσεις

Ο administrator έχει τη διακαιοδοσία να εισάγει νέους χρήστες στο σύστημα, να ενημερώνει τα στοιχέια τους και να λαμβάνει γνώση σχετικά με την «κατάστασή» τους, δηλαδή τα διάφορα στοιχέια που τους χαρακτηρίζουν μέσα στο σύστημα. Αυτές τις ενέργειες μπορεί να τις πραγματοποιήσει και από το CLI και από το front-end.

## 3.2 Απαιτήσεις επιδόσεων

Θέλουμε να ανταποκρίνεται όσο το δυνατόν πιο γρήγορα και με μεγαλύτερη ακρίβεια στις αναζητήσεις, δεδομένου του μεγάλου όγκου πληροφοριών που διαχειρίζεται.

Επείδη ακριβώς ενδιαφερόμαστε για την επίδοση του συστήματος μας έχει αναπτυχθεί το benchmarking με το οποίο ελέγχεται η επίδοση του REST-API. Έτσι, μέσω της εφαρμογής Apache JMeter φτιάχνονται εικονικοί χρήστες και στέλνονται πολλά requests ταυτόχρονα ώστε να λαμβάνονται οι χρόνοι απόκρισης του συστήματος, το πλήθος των bytes που στάλθηκαν και επιστράφηκαν, καθώς και των επιτυχημένων κλήσεων που έγιναν. Με αυτόν τον τρόπο παρακολουθείται η επίδοση του συστήματος ώστε να μπόρει αυτό να προσαρμοστεί κατάλληλα στις απαιτήσεις.

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Για να έχει κάποιος χρήστης πρόσβαση στα δεδομένα που παρέχονται απαιτείται να διαθέτει API key, δηλαδή να είναι εγγεγραμμένος στη βάση δεδομένων.

Τα δεδομένα είναι ομαδοποιημένα κατάλληλα κατά χρόνο, μήνα και ημέρα και επομένως με κατάλληλες κλήσεις μπορούν να ληφθούν επαρκώς και ακριβώς.

## 3.4 Περιορισμοί σχεδίασης

Ο βασικός περιορισμός αφορά τα δεδομένα και σχετίζεται με το ποια δεδομένα λαμβάνει το λογισμικό μας και ποιες πράξεις μπορεί να πραγαματοποιήσει σ’ αυτά, οι οποίες με τη σειρά τους πηγάζουν από τις προδιαγραφές του λογισμικού μας.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Ανάλογα με τον κάθε stakeholder διαφέρει η διαθεσιμότητα των δεδομένων του λογισμικού. Για μία εταιρεία, για παράδειγμα, παραγωγής ενέργειας το λογισμικό πρέπει να είναι διαθέσιμο συνεχώς ώστε να υπάρχει διαρκής τροφοδότηση. Αντίθετα για κάποιον δημοσιογράφο τον ενδιαφέρει να βρει αποτελέσματα μία οποιαδήποτε χρονική στιγμή χωρίς να είναι απαραίτητο να είναι διαθέσιμο το σύστημά μας όλο το 24ωρο.

### 3.5.2 Απαιτήσεις ασφάλειας

Δημιουργούμε ένα ασφαλές δίκτυο για τους χρήστες ως προς την άντληση των δεδομένων για να αποφύγουμε τυχόν αλλοιώσεις από τρτίτους. Όμως τα αποτελέσματα είναι δωρεάν και μπορούν πρόσβαση σε αυτά να έχουν όλοι όσοι το επιθυμούν.

### 3.5.3 Απαιτήσεις συντήρησης

Το δίκτυό μας να συντηρείται ανά τακτά χρονικά διαστήματα ώστε να διασφαλίζεται η ορθότητα των αποτελεσμάτων, να μην υπολειτουργεί και να συμβαδίζει με τα εκάστοτε απαιτήσεις.