

# Έγγραφο απαιτήσεων λογισμικού (SRS)

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

## Energastalker app

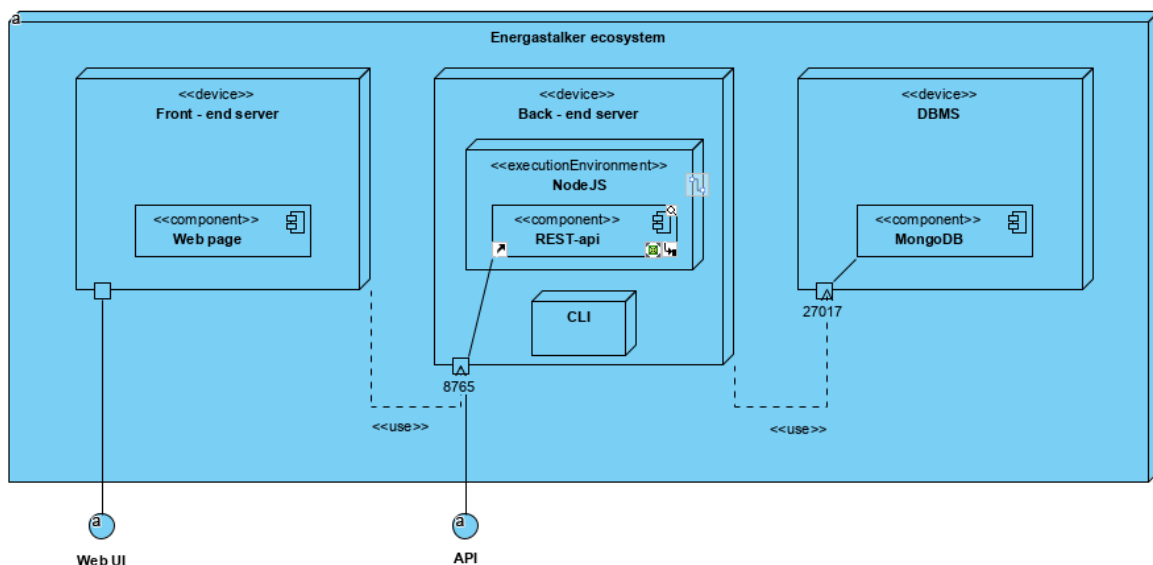
### 1. Εισαγωγή

#### 1.1 Εισαγωγή: σκοπός του λογισμικού

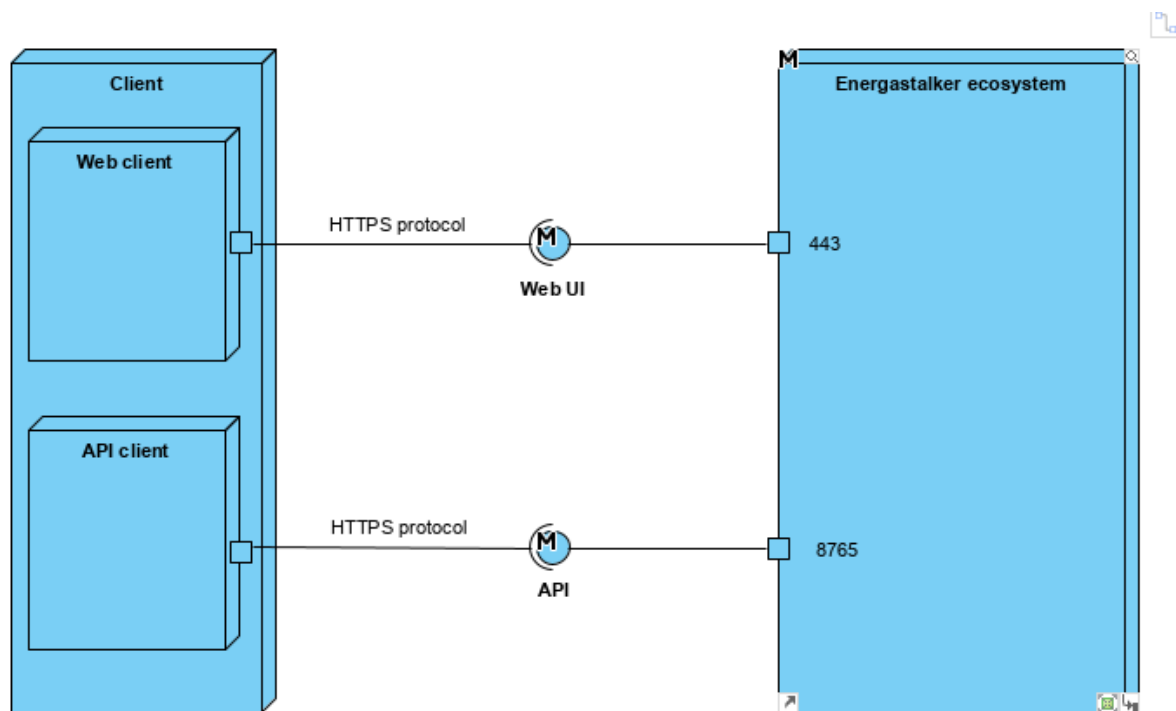
Σκοπός του πληροφοριακού συστήματος “Energastalker”, εφεξής αναφερόμενο ως έργο ή/και project, είναι η ανάπτυξη ενός διαδικτυακού παρατηρητηρίου ενεργειακών δεδομένων, όπου οι χρήστες θα είναι σε θέση να διαχειριστούν και να επεξεργαστούν τα δεδομένα αυτά της αγοράς ηλεκτρικής ενέργειας στην Ευρώπη, τα οποία διατίθενται ως ανοικτά δεδομένα (Open Data) από το δικτυακό τόπο [transparency.entsoe.eu](http://transparency.entsoe.eu). Τα δεδομένα αυτά είναι κρίσιμα για τη διαφάνεια στη λειτουργία της αγοράς ηλεκτρικής ενέργειας και η διάθεση, οπτικοποίηση και ανάλυσή τους είναι απαραίτητες υπηρεσίες προς τους συμμετέχοντες στην αγορά. Την ανάγκη αυτή έχει την προσδοκία να ικανοποιήσει η “Ομάδα Πύραυλος” με το παρόν project.

#### 1.2 Διεπαφές (interfaces)

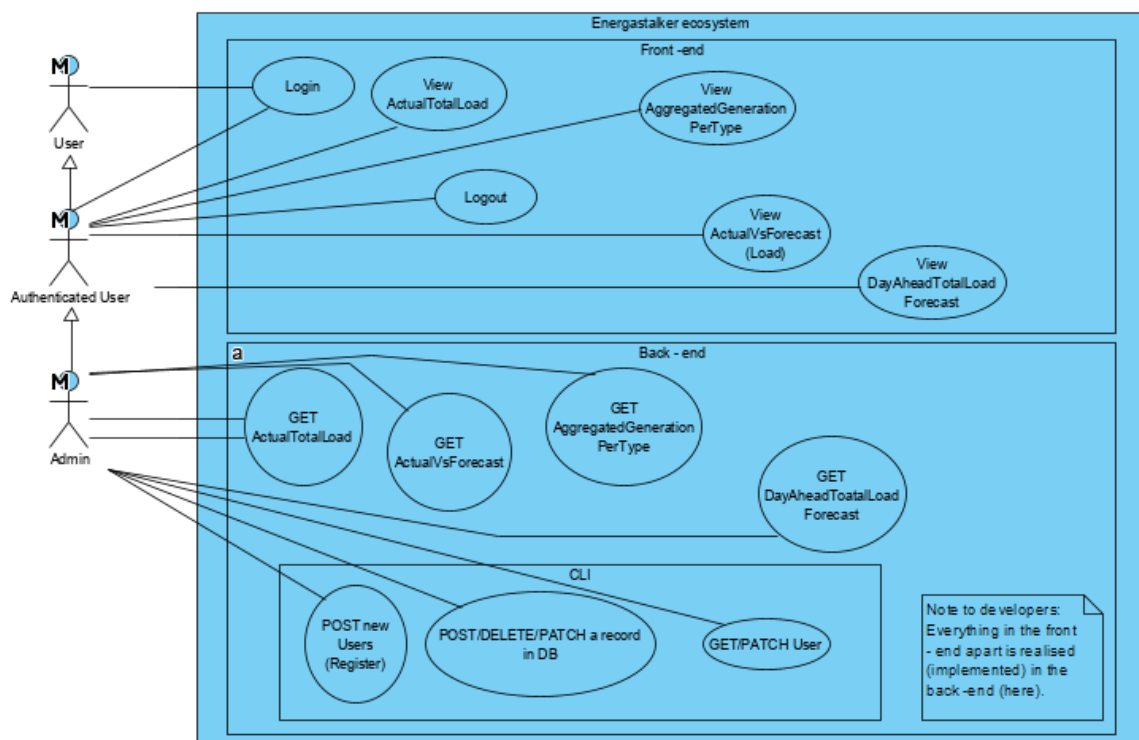
Η επικοινωνία του χρήστη με την εφαρμογή και η πρόσβασή του στα δεδομένα που τον ενδιαφέρουν γίνεται μέσω ενός RESTful API. Προσδιορίζοντας τις αντίστοιχες παραμέτρους μπορούμε να διαχειριζόμαστε μια συγκεκριμένη πληροφορία, προερχόμενη από τη βάση δεδομένων μας. Για φιλικότερη ως προς τον χρήστη επικοινωνία με το Energastalker app, χρησιμοποιείται ένα UI, μέσω ενός εύχρηστου dashboard.



### 1.2.1 Διεπαφές με εξωτερικά συστήματα



### 1.2.2 Διεπαφές με το χρήστη



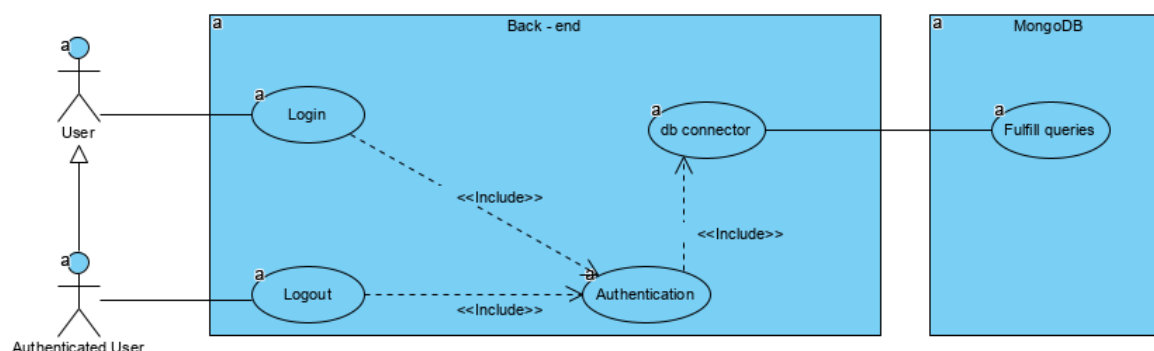
## 2. Αναφορές - πηγές πληροφοριών

- <https://transparency.entsoe.eu/> (εναποθετήριο open data που θα εισαχθούν στη βάση δεδομένων του project)
- [https://transparency.entsoe.eu/content/static\\_content/Static%20content/knowledge%20base/SFTP-Transparency\\_Docs.html](https://transparency.entsoe.eu/content/static_content/Static%20content/knowledge%20base/SFTP-Transparency_Docs.html)

## 3. Προδιαγραφές απαιτήσεων λογισμικού

### 3.1 Περιπτώσεις χρήσης

#### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: ΣΥΝΔΕΣΗ ΚΑΙ ΑΠΟΣΥΝΔΕΣΗ ΧΡΗΣΤΩΝ



##### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Η περίπτωση χρήσης αυτή αφορά όλες τις κατηγορίες χρηστών (κανονικούς και premium). Οι χρήστες θα έχουν την δυνατότητα εγγραφής στην πλατφόρμα μας κάνοντας αίτηση απόκτησης λογαριασμού στους διαχειριστές, ενώ οι ήδη εγγεγραμμένοι θα μπορούν να προβούν σε ενέργειες Σύνδεσης ή Αποσύνδεσης.

##### 3.1.1.2 Προϋποθέσεις εκτέλεσης

Προκειμένου να εκτελεστεί η συγκεκριμένη περίπτωση χρήσης, θα πρέπει να υπάρχουν εγγεγραμμένοι χρήστες στην πλατφόρμα μας, ώστε να προσπαθήσει ένας εξ αυτών να συνδεθεί, ή να είναι κάποιος ήδη συνδεδεμένος ώστε να αποσυνδεθεί.

##### 3.1.1.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης είναι αρχικά το UI (frontend) μέσω του οποίου ένας χρήστης μπορεί να συνδεθεί/αποσυνδεθεί. Για να εκτελεστούν οι ενέργειες αυτές, στέλνονται τα ανάλογα post queries μέσω του backend προς τη dbms, και αναλόγως με την ενέργεια που θέλουμε να εκτελεστεί η βάση επιστρέφει στον χρήστη ανάλογα μηνύματα ή σφάλματα. Για παράδειγμα, κατά την σύνδεση ενός χρήστη, αν το Username που υποβλήθηκε δεν υπάρχει, η dbms θα επιστρέψει το αποτέλεσμα της αναζήτησης στο backend και εκείνο με τη σειρά του θα μας εμφανίσει σχετικό σφάλμα (λάθος στοιχεία σύνδεσης).

#### 3.1.1.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου στις περιπτώσεις σύνδεσης ή εγγραφής στέλνονται με τον ίδιο τρόπο, μέσω Json ή csv. Στην περίπτωση αποσύνδεσης, στέλνεται με Post ένα Logout query το οποίο θα πρέπει να περιέχει και το authorization token του συνδεδεμένου χρήστη.

#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

##### Για τη σύνδεση:

Βήμα 1: Ο χρήστης εισάγει τα στοιχεία του στην φόρμα.

Βήμα 2: Το backend κάνει post τα στοιχεία του χρήστη στην dbms.

Βήμα 3: Η dbms αναζητά το Username. Αν δεν τον βρει επιστρέφει το αποτέλεσμα της αναζήτησης ως null. Το backend με τη σειρά του μας επιστρέφει σφάλμα, πως το Username δεν υπάρχει.

Βήμα 4: Αν η αναζήτηση είναι επιτυχής, ελέγχεται το password και αν ταιριάζει η dbms επιστρέφει ένα authorization token προς το backend. Το backend, μας επιστρέφει το μήνυμα πως η σύνδεση ήταν επιτυχής.

Βήμα 5: Αν το password δεν ταιριάζει με το ήδη αποθηκευμένο στην βάση, το backend μας επιστρέφει σφάλμα.

##### Για την αποσύνδεση:

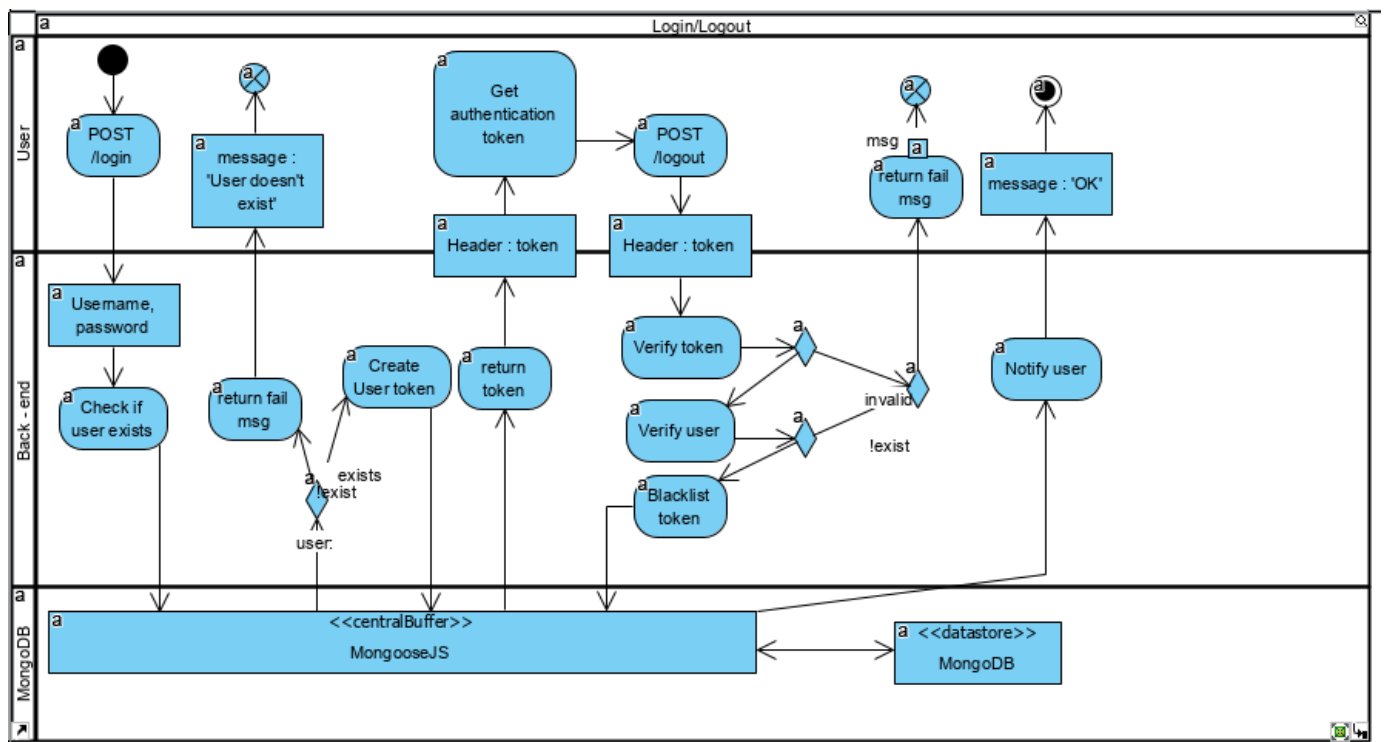
Βήμα 1: Ο χρήστης επιλέγει την αποσύνδεση από την πλατφόρμα μέσω του αντίστοιχου συνδέσμου στο frontend.

Βήμα 2: Το backend στέλνει ένα logout post query μαζί με το username και το authorization token του συνδεδεμένου χρήστη.

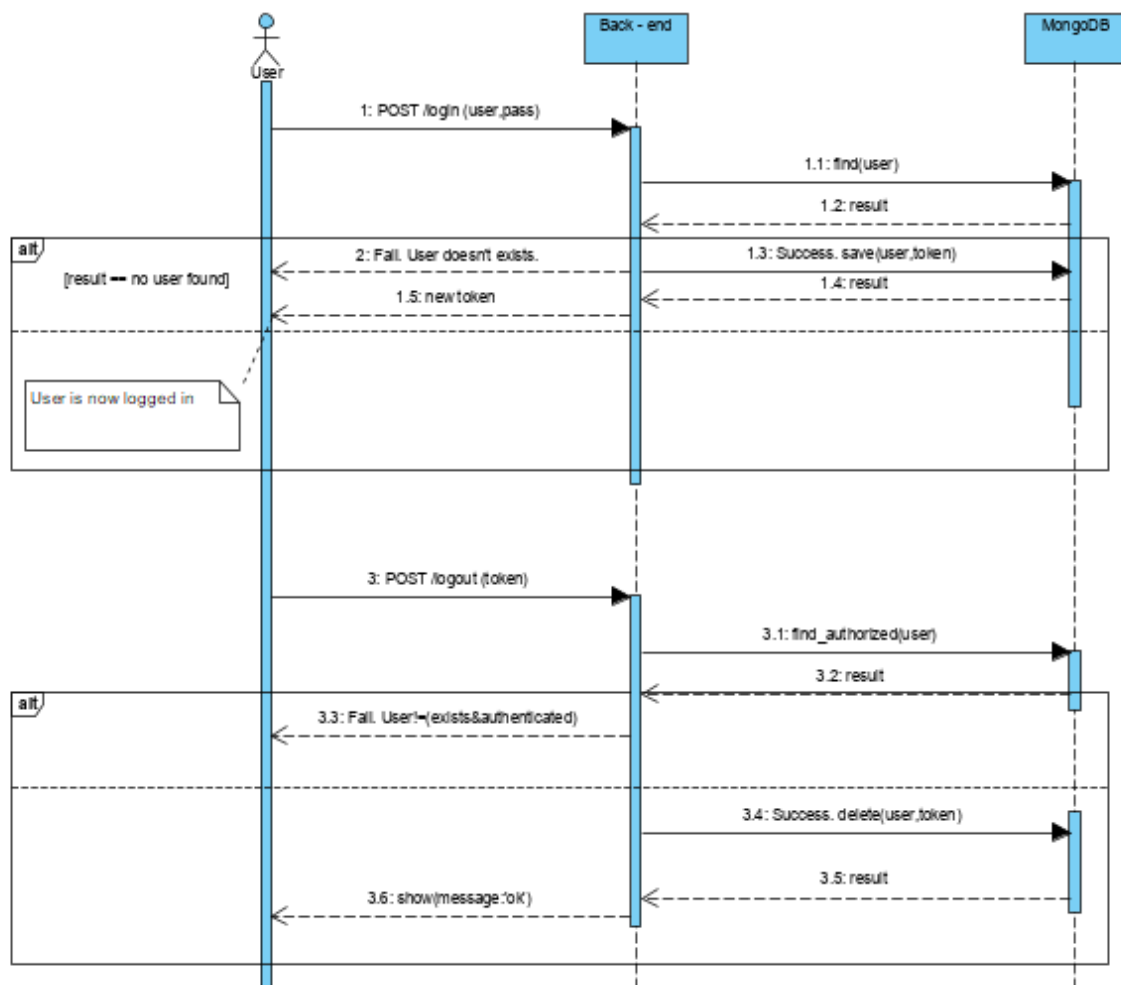
Βήμα 3: Η dbms αναζητά το συνδυασμό (Username, Token). Αν δεν τον βρει, το backend επιστρέφει σφάλμα πως ο χρήστης δεν βρέθηκε.

Βήμα 4: Αν η αναζήτηση είναι επιτυχής, η dbms διαγράφει το Authorization Token και το backend μας επιστρέφει το μήνυμα πως έχουμε αποσυνδεθεί.

## Uml Activity Diagram



## Uml Sequence Diagram



### 3.1.1.6 Δεδομένα εξόδου

Τα δεδομένα εξόδου παρουσιάστηκαν όπως αυτά ζητούνται στα Uml διαγράμματα της ενότητας 3.1.1.5.

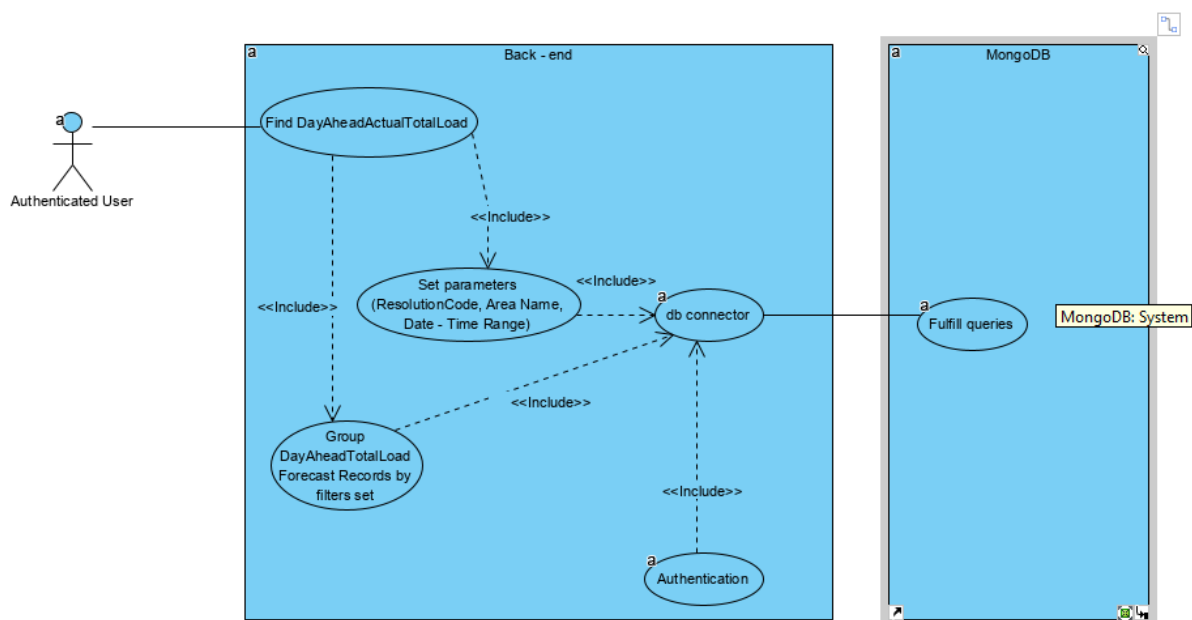
### 3.1.1.7 Παράμετροι

Προκειμένου τα δεδομένα που στέλνονται να είναι έγκυρα, θα πρέπει οι τύποι των μεταβλητών να είναι αυστηρά καθορισμένοι χωρίς περιθώριο λαθών. Για παράδειγμα, το username και το password του χρήστη θα πρέπει να είναι κωδικοποιημένα σε application/x-www-form-urlencoded και να υπάρχουν ήδη στο database και να είναι έγκυρα.

### 3.1.1.8 Παρατηρήσεις

N/A

## 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: ΕΥΡΕΣΗ DayAheadTotalLoadForecast



### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι χρήστες που εμπλέκονται στη συγκεκριμένη περίπτωση χρήσης είναι οι εγγεγραμμένοι χρήστες, δηλαδή αυτοί που έχουν αποκτήσει λογαριασμό από τους διαχειριστές. Επιπλέον για να εκτελέσουν τη συγκεκριμένη ενέργεια, πρέπει να είναι διαπιστευμένοι, δηλαδή να έχουν συνδεθεί στην υπηρεσία με τα στοιχεία τους.

### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Προκειμένου ένας χρήστης να αναζητήσει records για DayAheadTotalLoadForecast, πρέπει να έχει συνδεθεί στον λογαριασμό του, για να εκτελέσει τα queries προς το σύστημα. Επιπλέον, θα πρέπει ο χρήστης να έχει εναπομείναντα quotas, αν θέλει να εκτελέσει αναζήτηση, αλλιώς το back-end επιστρέφει αρνητικό reply.

### 3.1.2.3 Περιβάλλον εκτέλεσης

Ο χρήστης θα μπορεί να αναζητήσει DayAheadTotalLoadForecast μέσω του User Interface. Κατά την επεξεργασία της κλήσης, το Api θα αποστείλει το αίτημα στο Dbms και εκείνο με τη σειρά του θα επιστρέψει μέσω του back-end και θα εμφανίσει στον χρήστη μέσω του UI τα σχετικά αποτελέσματα.

### 3.1.2.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου στην περίπτωση αναζήτησης DayAheadTotalLoadForecast στέλνονται με τον ίδιο τρόπο, μέσω Json ή csv, αναλόγως του format που προτιμά ο χρήστης. Στην περίπτωσή μας, στέλνεται με GET ένα query το οποίο θα πρέπει να περιέχει και το api-key του συνδεδεμένου χρήστη.

### 3.1.2.5 Παράμετροι

Προκειμένου τα δεδομένα που στέλνονται να είναι έγκυρα, θα πρέπει οι τύποι των μεταβλητών να είναι αυστηρά καθορισμένοι χωρίς περιθώριο λαθών. Για παράδειγμα, το AreaName θα πρέπει να είναι τύπου String, ενώ το DateTimeUTC, τύπου timestamp και να υπάρχουν ήδη στο database και να είναι έγκυρα.

### 3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

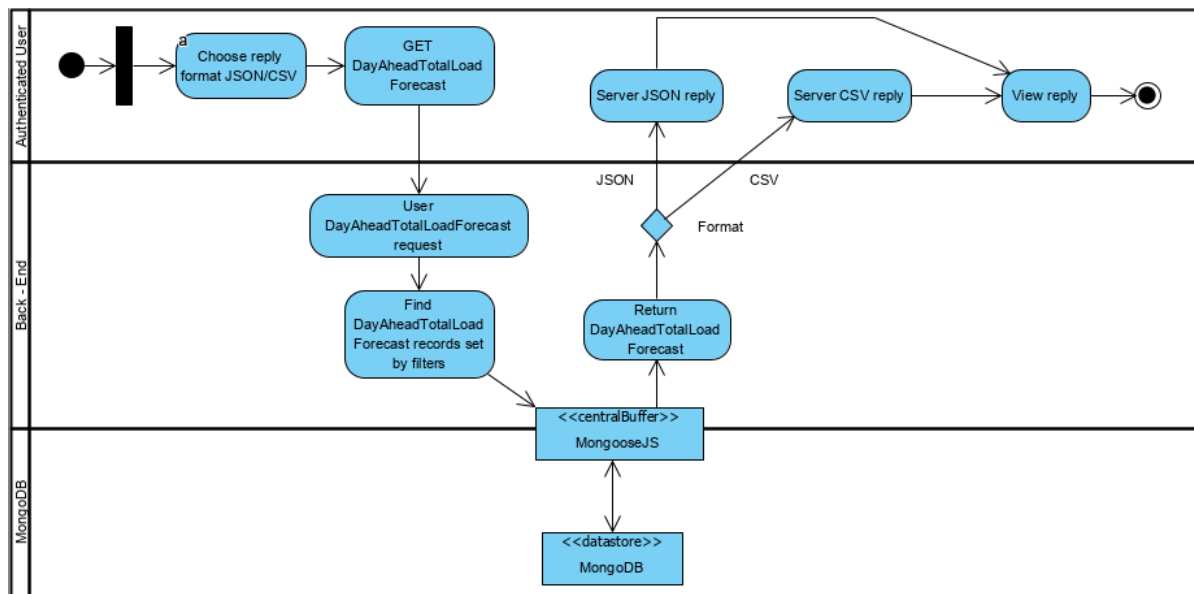
Βήμα 1: Ο χρήστης μπορεί να αναζητεί χρονοσειρές DayAheadTotalLoadForecast από την πλατφόρμα μέσω του αντίστοιχου συνδέσμου στο frontend, θέτοντας ως παραμέτρους εισόδου την περιοχή που τον ενδιαφέρει {AreaName}, την ημέρα {DD}, τον μήνα {MM}, και το έτος {YYYY}.

Βήμα 2: Το backend στέλνει ένα get query μαζί με το api-key του συνδεδεμένου χρήστη.

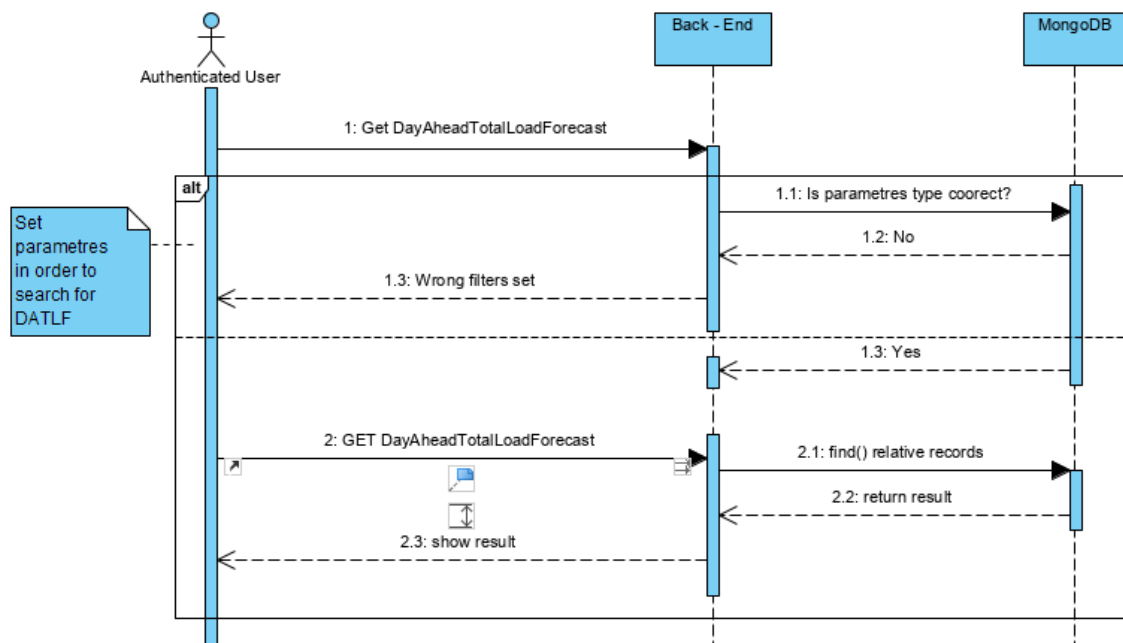
Βήμα 3: Η dbms αναζητά σχετικές εγγραφές, που αντιστοιχούν στις παραμέτρους αναζήτησης που έθεσε ο χρήστης.

Βήμα 4: Αν η αναζήτηση είναι επιτυχής, η dbms επιστρέφει τα αποτελέσματα μέσω του back-end και τα εμφανίζει στον χρήστη στο UI με χρονική ανάλυση και στο format που προτιμά.

## Uml Activity Diagram



## Uml Sequence Diagram



### 3.1.2.7 Δεδομένα εξόδου

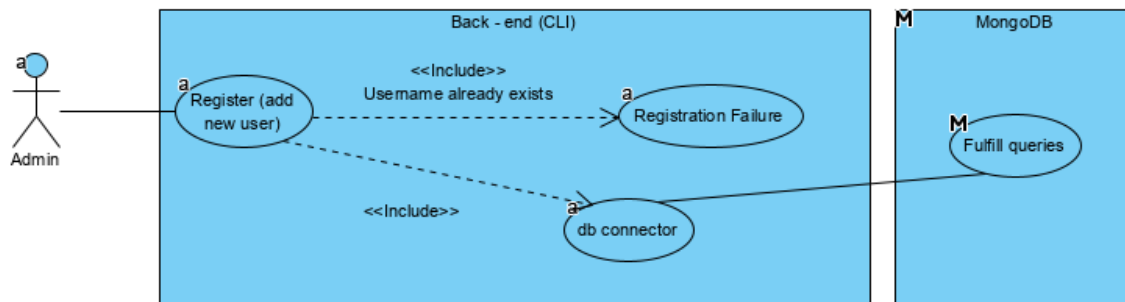
Τα δεδομένα εξόδου παρουσιάστηκαν όπως αυτά ζητούνται στα Uml διαγράμματα της ενότητας 3.1.2.6.

### 3.1.2.8 Παρατηρήσεις

N/A



### 3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: ΕΓΓΡΑΦΗ ΧΡΗΣΤΩΝ



#### 3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται

Ο διαχειριστής και μόνο αυτός θα μπορεί να δημιουργεί νέους λογαριασμούς χρηστών (register) μέσω του CLI, το οποίο θα φιλοξενείται στο μηχάνημα που φιλοξενεί την υπηρεσία και δεν θα είναι προσβάσιμο από τον χρήστη. Το CLI παρακάμπτοντας του front-end θα αποστέλλει το αίτημα POST (signup) στο Dbms και εκείνο με τη σειρά του θα επιστρέψει μέσω του back-end και θα εμφανίσει στον διαχειριστή μέσω του CLI τα σχετικά αποτελέσματα, αν η εγγραφή είναι επιτυχής.

#### 3.1.3.2 Προϋποθέσεις εκτέλεσης

Για την περίπτωση που κάποιος διαχειριστής θέλει απλώς να εγγράψει έναν νέο χρήστη, η μόνη προϋπόθεση που υπάρχει είναι να μην υπάρχει ήδη εγγεγραμμένος χρήστης με το ίδιο username, και ο διαχειριστής να εισάγει τις απαιτούμενες παραμέτρους σωστά (σύμφωνα με τους προκαθορισμένους τύπους).

#### 3.1.3.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης είναι το CLI μέσω του οποίου ένας διαχειριστής μπορεί να δημιουργήσει έναν νέο λογαριασμό χρήστη. Για να εκτελεστούν οι ενέργειες αυτές, στέλνονται τα ανάλογα post queries μέσω του backend προς τη dbms, και η βάση επιστρέφει στον διαχειριστή είτε το API-KEY, σε περίπτωση επιτυχούς εγγραφής, είτε μήνυμα σφάλματος. Για παράδειγμα, κατά την εγγραφή ενός χρήστη, αν το username που υποβλήθηκε υπάρχει ήδη, η dbms θα επιστρέψει το αποτέλεσμα της αναζήτησης στο backend και εκείνο με τη σειρά του θα εμφανίσει σφάλμα στο CLI, καθώς το username χρησιμοποιείται ήδη.

#### 3.1.3.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου στην περίπτωση εγγραφής στέλνονται με τον ίδιο τρόπο, μέσω Json ή csv. Στέλνεται με Post ένα query το οποίο θα πρέπει να περιέχει το username και το password, το email και τα διαθέσιμα quota που θα έχει ο νέος χρήστης.

#### 3.1.3.5 Παράμετροι

Προκειμένου τα δεδομένα που στέλνονται να είναι έγκυρα, θα πρέπει οι τύποι των μεταβλητών να είναι αυστηρά καθορισμένοι χωρίς περιθώριο λαθών. Έτσι για παράδειγμα σε περίπτωση εγγραφής χρήστη, το Username και το Password ελέγχεται αν έχει τη σωστή μορφή application/x-www-form-urlencoded, ενώ το password στέλνεται hashed για λόγους ασφάλειας.

#### 3.1.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Για την εγγραφή:

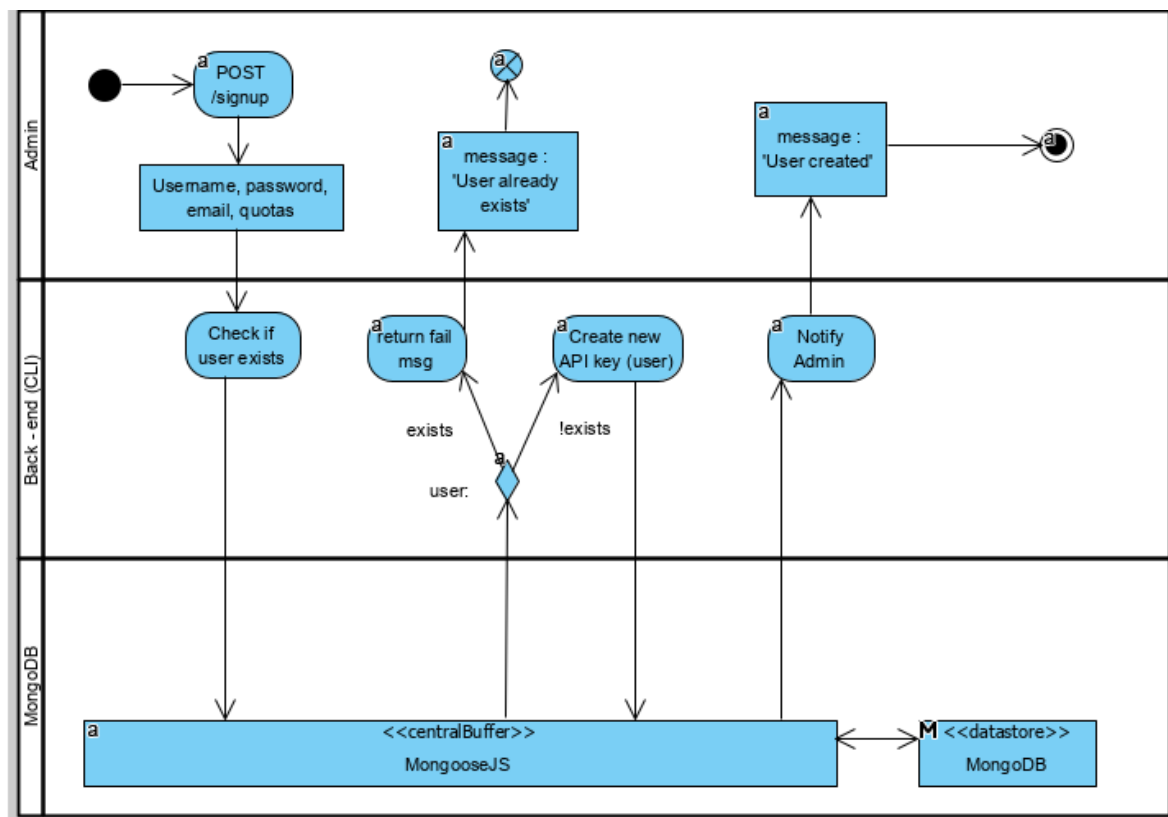
Βήμα 1: Ο διαχειριστής με κατάλληλες εντολές μέσω του CLI, πληκτρολογεί το username και το password που επιθυμεί και το καταχωρεί. Παράλληλα, πρέπει να συμπληρώσει το mail του χρήστη και τα quota που θα του διαθέσει.

Βήμα 2: Το backend στέλνει Post query προς το dbms, με τα παραπάνω στοιχεία.

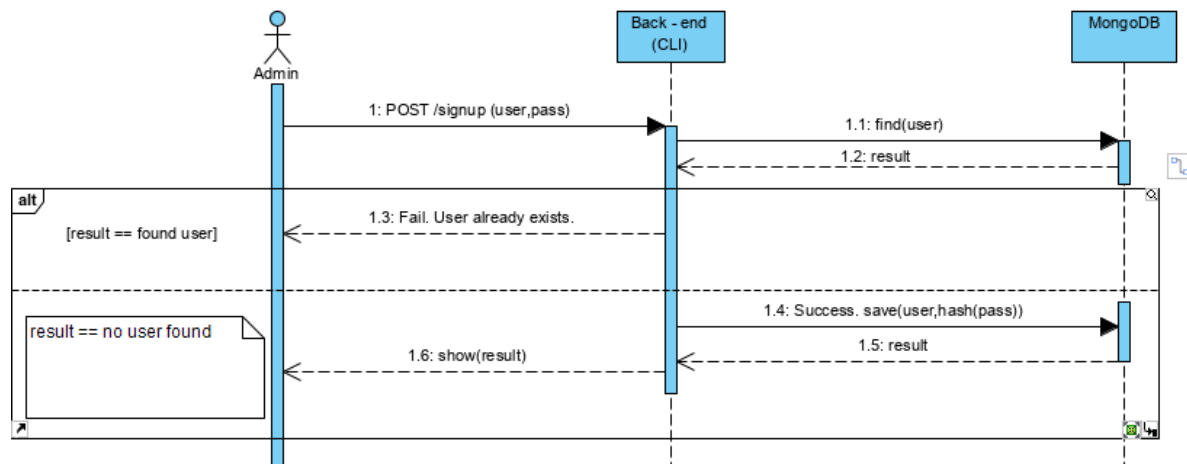
Βήμα 3: Η db αναζητά το username. Σε περίπτωση που η αναζήτηση επιστρέψει αποτέλεσμα, το backend μας επιστρέφει σφάλμα πως ο χρήστης υπάρχει ήδη.

Βήμα 4: Εάν η αναζήτηση είναι ανεπιτυχής, η dbms αποθηκεύει το νέο συνδυασμό και το backend επιστρέφει μήνυμα για την επιτυχημένη εγγραφή του χρήστη, μαζί με το API-KEY του, μέσω του CLI.

### Uml Activity Diagram



## Uml Sequence Diagram



### 3.1.3.7 Δεδομένα εξόδου

Τα δεδομένα εξόδου παρουσιάστηκαν όπως αυτά ζητούνται στα Uml διαγράμματα της ενότητας 3.1.3.6. Εν προκειμένου το αποτέλεσμα μιας επιτυχημένης εγγραφής είναι η δημιουργία ενός API-KEY.

### 3.1.3.8 Παρατηρήσεις

N/A

## 3.2 Απαιτήσεις επιδόσεων

Για να είναι ανταγωνιστικό και εύχρηστο το λογισμικό που αναπτύσσουμε, θα πρέπει αυτό να ικανοποιεί κάποιες απαιτήσεις που σχετίζονται με:

- Την ταχύτητα αποκρισμότητας: κάθε αίτημα θα πρέπει να λαμβάνει απάντηση σε λιγότερο από 150msec (<150msec).
- Την επεκτασιμότητα του υλικού: διπλασιασμός των υλικών πόρων του συστήματος οδηγεί σε διπλασιασμό των αιτημάτων που μπορεί να ικανοποιηθούν.
- Αξιοπιστία, ασφάλεια, διαθεσιμότητα και μεταφερσιμότητα των δεδομένων (αναλύεται στη συνέχεια).

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Για να έχει ένας χρήστης δικαίωμα αναζήτησης (get) στην εφαρμογή, θα πρέπει πρώτα να έχει εγγραφεί, δηλαδή να έχει αιτηθεί τη δημιουργία λογαριασμού στους διαχειριστές, ώστε να είναι καταχωρημένος στη βάση δεδομένων και να του εκχωρηθούν τα αντίστοιχα δικαιώματα. Εφόσον μετά το register κάνει επιτυχές login (εισάγοντας σωστό username και password), αποκτά πρόσβαση στην πλατφόρμα και στα δεδομένα της βάσης δεδομένων, εκτελώντας κατάλληλα queries. Τέλος, ένας

admin, αποκτά πρόσβαση στην πλατφόρμα με όμοιο τρόπο με έναν εθελοντή, αλλά έχει κάθε δικαίωμα στα δεδομένα (post, patch, get, delete).

### 3.4 Περιορισμοί σχεδίασης

#### 3.4.1 Περιορισμός προσβασιμότητας εφαρμογής

Το έργο στην πιο εύχρηστη έκδοσή του, θα απαιτεί πρόσβαση στο διαδίκτυο είτε μέσω παρεχόμενου wifi είτε μέσω σταθερής σύνδεσης.

#### 3.4.2 Περιορισμός χρήσης πόρων της υπηρεσίας

Ο χρήστης έχει τη δυνατότητα εκτέλεσης ερωτημάτων μέσω της εφαρμογής στη βάση δεδομένων, μέχρι να εξαντλήσει τα quotas που του παρέχονται κατά τη σύνδεσή του.

#### 3.4.3 Περιορισμός συνδεσιμότητας χρήστη

Ο χρήστης θα μπορεί να συνδεθεί στην υπηρεσία αν και μόνο αν οι διαχειριστές έχουν προχωρήσει στη δημιουργία του σχετικού λογαριασμού για την εγγραφή του εκάστοτε χρήστη.

#### 3.4.4 Περιορισμός ως προς το χρόνο ολοκλήρωσης του έργου

Το project θα πρέπει να έχει ολοκληρωθεί μέχρι τα το τέλος του Φλεβάρη του έτους 2020.

#### 3.4.5 Περιορισμός χρήσης πόρων της υπηρεσίας – premium account

Ο δημοσιογράφος έχει τη δυνατότητα εκτέλεσης απεριόριστων ερωτημάτων μέσω της εφαρμογής στη βάση δεδομένων, αν και εφόσον έχει premium συνδρομή στην υπηρεσία.

#### 3.4.6 Περιορισμός ως προς τον τύπο των εισαγόμενων δεδομένων

Τα δεδομένα που εισάγονται από τους διαχειριστές πρέπει να τηρούν τον τύπο, καθώς και κάθε περιορισμό που έχει τεθεί για τη μορφή τους στα αντίστοιχα μοντέλα.

### 3.5 Λοιπές απαιτήσεις

#### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Προκειμένου, η εφαρμογή μας να είναι διαρκώς διαθέσιμη στον χρήστη θα χρησιμοποιούνται εφεδρικοί servers, ώστε σε περίπτωση που κάποιος master κρασάρει, τα αιτήματα να ικανοποιούνται από έναν δευτερεύον. Επιπλέον, η πρόσβαση στα δεδομένα ανα πάσα στιγμή είναι εφικτή χάρη στην κατανομημένη αρχιτεκτονική του MongoDB.

#### 3.5.2 Απαιτήσεις ασφάλειας

Η εφαρμογή μας εγγυάται μια ασφαλή εμπειρία στον χρήστη, καθώς η πρόσβαση σε αυτήν γίνεται μέσω SSL πρωτοκόλλου, με πιστοποιητικό υπογεγραμμένο από έμπιστη αρχή πιστοποίησης. Δηλαδή υποστηρίζεται HTTPS. Επιπλέον, τα προσωπικά δεδομένα των χρηστών διαφυλάσσονται αποτελεσματικά, καθώς τα password κρυπτογραφούνται με σχετικά hash functions.

### 3.5.3 Απαιτήσεις συντήρησης

Όπως κάθε ολοκληρωμένο και ανταγωνιστικό έργο λογισμικού, η εφαρμογή Energastalker θα ενημερώνεται με τακτικά updates, που στόχο θα έχουν την πιθανή επίλυση bugs, την βελτίωση της εμπειρίας του χρήστη μέσω του εμπλουτισμού του user interface με καινούριες υπηρεσίες, καθώς και αποτελεσματικότερους τρόπους διαχείρισης των δεδομένων, ενώ ο όγκος τους θα αυξάνεται διαρκώς. Πιθανότατα, ανάλογα με τον φόρτο επισκεψιμότητας, θα μπορούσε να χρειαστεί και επέκταση υλικού σε ότι αφορά τους servers, ώστε να ικανοποιείται ο αυξημένος αριθμός αιτημάτων.