Έγγραφο απαιτήσεων λογισμικού (SRS)

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

[TouloumbEnergy]

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

*Το σύστημα που αναπτύσουμε συνιστά μια πλατφόρμα στην οποία ο χρήστης μπορεί να λαμβάνει δεδομένα σχετικά με την κατανάλωση ηλεκτρικής ενέργειας στην Ευρώπη. Στόχος του συστήματος είναι η μεγάλη ανάγκη για την επισκόπιση και την κατανόηση της χρήσης της ηλεκτρικής ενέργειας, τόσο από οργανισμούς, όσο και από ιδιότες. Με την αλλαγή των συνθηκών στον τομέα της παραγωγής ενέργειας και την εμφάνιση νέων μορφών ηλεκτρικής ενέργειας καθίσταται σημαντικό ο καθένας να γνωρίζει από που προέρχεται η παραγωγή για την περιφέρειά του. Την ανάγκη αυτή στοχεύει να καλύψει η εφαρμογή μας.*

## 1.2 Επισκόπηση του λογισμικού

*Στην εικόνα που ακολουθεί πατουσιάζονται οι αλληλεπιδράσεις που δύναται να έχει κάποιος χρήστης της εφαρμογής μας. Ως χρήστες ξεχωρίζουμε δύο κατηγορίες:*

* *Εγγεγραμμένους χρήστες (δυνατότητα προβολής των δεδομένων)*
* *Διαχειρηστής (δυνατότητα διαχείρησης χρηστών).*

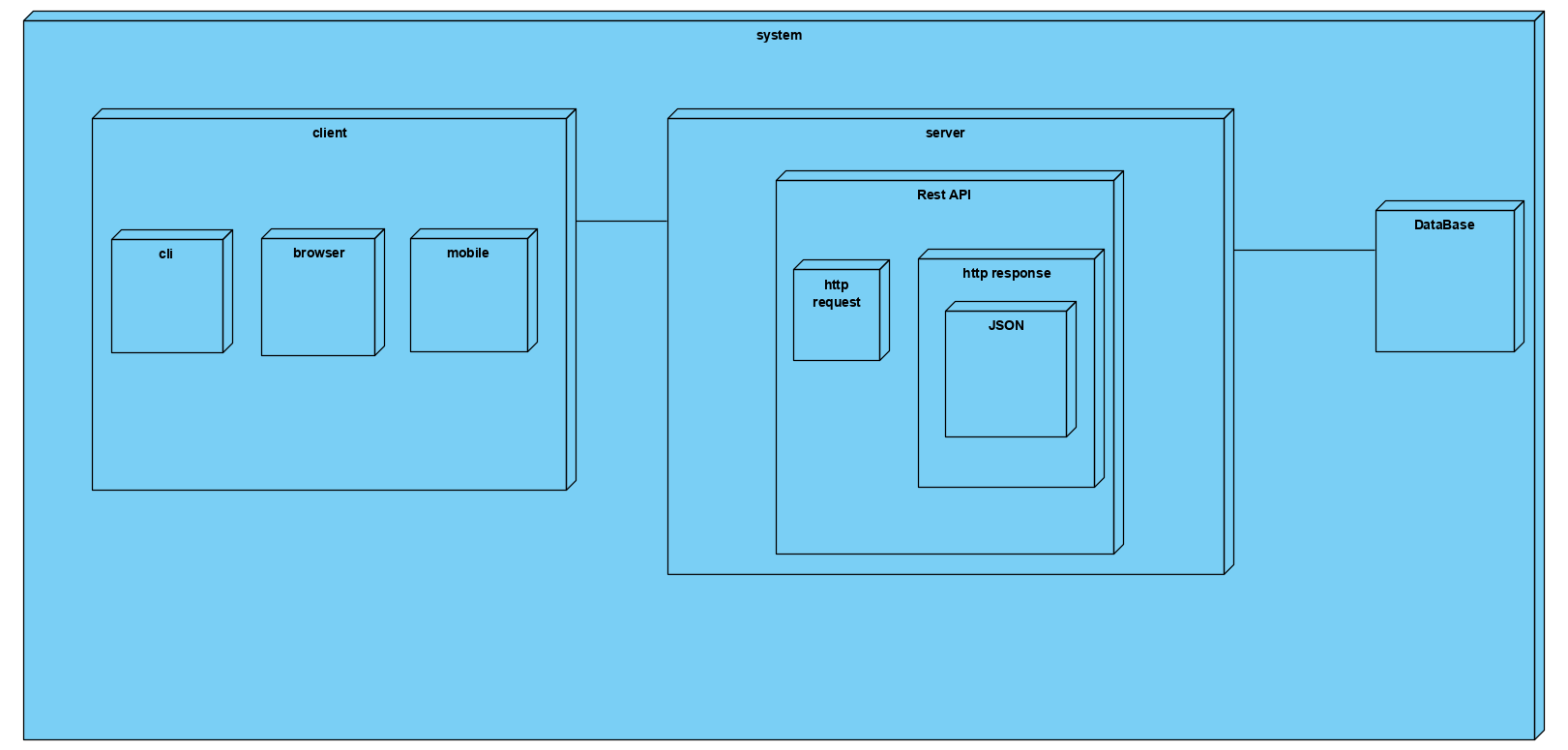
*(MORE TBA).*

### 1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού

*Ακολουθόντας σε μεγάλο βαθμό το μοντέλο MVC και ενσωματώνοντας την αρχιτεκτονική REST, μια επισκόπηση της δομής του λογισμικού μας φαίνεται από το Deployment Diagram της εικόνας 2. Οι εξωτερικές διεπαφές του συστέματος αφορούν:*

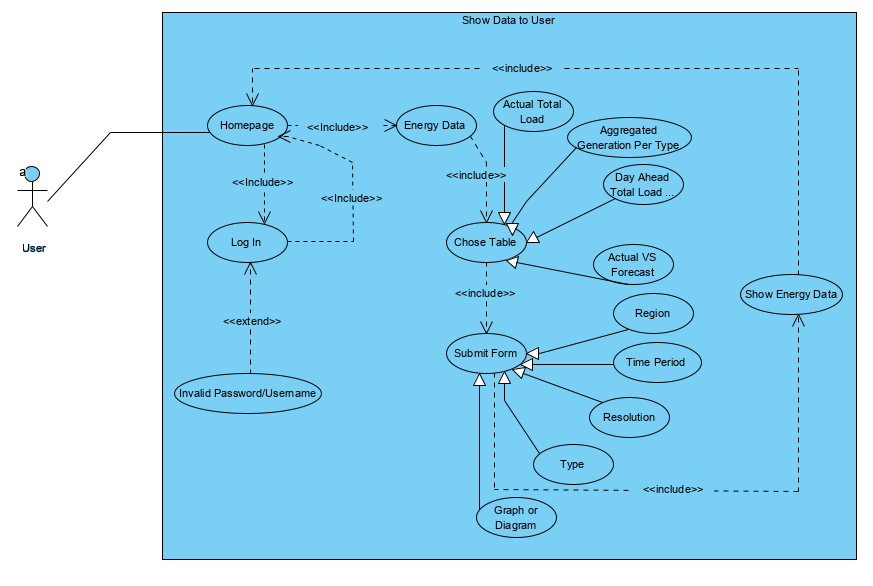
* *Τα άκρα της σύνδεσης στον browser με τον εξυπηρετητή που φιλλοξενεί το API της εφαρμογής.*
* *Την επικοινωνία ανάμεσα στο User Interface και REST API για τη σωστή ενημέρωση του περιεχομένου που ζητείται από το χρήστη.*
* *Τις εξόδους του REST API προς τη βάση δεδομενων.*
* *Η επικοινωνία μέσω του CLI.*
* *Η επικοινωνία μέσω του mobile app.*

*Ο Client επικοινωνεί μέσω http requests με το API. Δέχεται απάντηση από το API http responses σε μορφή JSON με τις πληροφορίες που ζητήθηκαν. Το API από τη μεριά του είναι υπεύθηνο για την επικοινωνία με τη βάση δεδομένων μέσω SQL requests & responses. Ανάλογα με τα αιτήματα του χρήστη πραγματοποιεί την απαραίτητη επεξεργασία δεδομένων, ενημερώνει ή ζητά αποτελέσματα από τη βάση και, τέλος, επιστρέφει αποτελέσματα στον φυλλομετρητή του Client με μορφοποίηση JSON.*

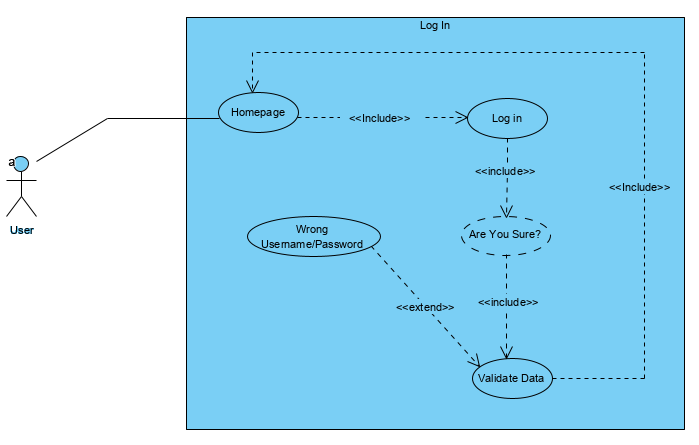
*Εικόνα 2. Deployment Diagram του Συστήματος*

### 1.3.2 Διεπαφές με το χρήστη

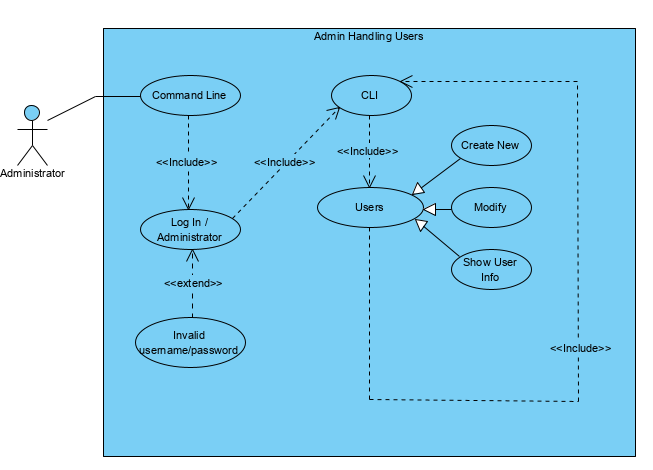
*Στις επόμενες εικόνες περιγράφουμε τα σενάρια διεπαφής του χρήστη με την εφαρμογή.*

**

*Εικόνα 3. Εμφάνιση δεδομένων στο χρήστη από το Web App.*

**

*Εικόνα 4. Είσοδος χρήστη από το Web App.*

**

*Εικόνα 5. Επεξεργασία χρηστών από τον Administrator.*

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Εξωτερικές διεπαφές

*Στην προηγούμενη ενότητα 1.3.1 περιγράψαμε τη δομή των διεπαφών της εφαρμογής μας. Σε αυτή την ενότητα θα περιγράψουμε την εμπειρία του χρήστη μέσα από τις γραφικές διεπαφές (Browser UI και Mobile UI) καθώς και τις μη γραφικές διεπαφές (CLI) με τις οποίες αλληλοεπιδρά.*

*Αναφορικά με το web app UI, ο πρώτος κόμβος επίσκεψης κάθε χρήστη καθορίζεται δυναμικά με το εάν έχει συνδεθεί ή όχι. Αν ο χρήστης δεν είναι συνδεδεμένος, σε όποια σελίδα της εφαρμογής προσπαθήσει να αποκτήσει πρόσβαση θα γίνει redirect στη σελίδα σύνδεσης (Login Page). Αντίθετα, αν ο χρήστης είναι συνδεδεμένος ή συνδέεται, μεταφέρεται στην κεντρική σελίδα της εφαρμογής (Home Page). Αυτή είναι η κύρια σελίδα της εφαρμογής και αποτελείται από τα 3 βασικά μέρη : Header, Body, Footer. Στο header της σελίδας εμφανίζεται το logo της ομάδας, καθώς και links για μετάβαση στην αρχική σελίδα και για Log Out. Στο footer, έχουμε το όνομα της ομάδας μας.*

*Το κυρίως σώμα της εφαρμογής χωρίζεται σε 2 μέρη: Στα πεδία αναζήτησης που βρίσκονται στο αριστερό μέρος*

## 3.2 Λειτουργίες: περιπτώσεις χρήσης

*Λεπτομερής προδιαγραφή των λειτουργιών του λογισμικού σε επίπεδο περιπτώσεων χρήσης.*

*Για κάθε μία λειτουργία δίνονται τα ακόλουθα.*

### 3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: (Web log in)

#### 3.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται

*Αυτή η χρήση αφορά τον χρήστη που θέλει να συνδεθεί μέσω του Web App ώστε να καταναλωσει ττα δεδομένα του API.*

#### 3.2.1.2 Προϋποθέσεις εκτέλεσης

*Για να μπορεί ο χρήστης να συνδεθεί, πρέπει να είναι ήδη εγγεγραμένος στο σύστημα της βάσης δεδομένων. Η λειτουργία αυτή γίνεται από τον διαχειριστή.*

#### 3.2.1.3 Περιβάλλον εκτέλεσης

*Για τη συγκεκριμένη περίπτωση χρήσης, θα εξετάσουμε μόνο το περιβάλλον της εφαρμογής Web.*

#### 3.2.1.4 Δεδομένα εισόδου

*Τα δεδομένα εισόδου είναι τα ατομικά στοιχεία του εγγεγραμένου χρήστη (username, password).*

#### 3.2.1.5 Παράμετροι

*Για την επιτυχή είσοδο του χρήστη απαιτείται σωστός συνδυασμός username και password.*

#### 3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

*Βήμα 1: Είσοδος στο login page*

*Βήμα 2: Εισαγωγή username και password από τον χρήστη*

*Βήμα 3: Καταχώριση της φόρμας και επιβεβαίωση στο popup window*

*Βήμα 4: Αν είναι σωστά τα credentials, ο user συνδέεται και μεταφέρεται στο homepage, αλλιώς βήμα 4.*

*Στις επόμενες εικόνες διατίθεται διάγραμμα UML για το παραπάνω Use Case.*

#### 3.2.1.7 Δεδομένα εξόδου

*Δεν υπάρχει κάποιο δεδομένο εξόδου προς τον χρήστη, παρατηρεί πως η σύνδεση επιτεύχθηκε από την αλλαγή του Navigation Bar, στο οποίο τώρα αναφέρεται η ένδειξη Logout.*

(UML 1ΟΥ USE CASE ΕΔΩ)

### 3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: (Admin Handling Users)

#### 3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

*Η συγκεκριμένη περίπτωση χρήσης αφορά τους Administrators.*

#### 3.2.2.2 Προϋποθέσεις εκτέλεσης

*O χρήστης πρέπει να έχει δικαιώματα admin και το σωστό password ώστε να μπορέσει να συνδεθεί.*

#### 3.2.2.3 Περιβάλλον εκτέλεσης

*Η επεξεργασία των χρηστών γίνεται μόνο μέσω του command line interface, και όχι μέσω web app ή mobile app.*

#### 3.2.2.4 Δεδομένα εισόδου

*Τα δεδομένα εισόδου είναι το username (admin) και ο κωδικός πρόσβασης password. Επιπλέον, απαιτείται η επιλογή του είδους της επεξεργασίας (Create New/ Modify/ Show User Info).*

#### 3.2.2.5 Παράμετροι

*Για την επιτυχή είσοδο του διαχειρηστή απαιτείται ο σωστός κωδικός πρόσβασης από αυτόν.*

#### 3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

*Βήμα 1. Είσοδος στο CLI.*

*Βήμα 2. Login ως admin και εισαγωγή του κωδικού πρόσβασης. Αν δωθεί λανθασμένος κωδικός, εμφανίζεται μήνυμα λάθους και επιστρέφουμε στο βήμα 2.*

*Βήμα 3. Επιλογή του τύπου επεξεργασίας. Create New/ Modify/ Show User Info. Για τις πρώτες δύο περιπτώσεις, ακολουθούμε στο βήμα 4, για την Τρίτη στο βήμα 5 .*

*Βήμα 4. Εισαγωγή δεδομένων προς επεξεργασία. Επιστροφή στο βήμα 3.*

*Βήμα 5. Εμφάνιση δεδομένων του χρήστη, επιστροφή στο βήμα 3.*

#### 3.2.2.7 Δεδομένα εξόδου

*Ως δεδομένα εξόδου στην περίπτωση Create New/ Modify έχουμε μόνο Confirmation πως οι αλλαγές έγιναν σωστά. Στην περίπτωση Show User Info, έχουμε ως έξοδο τα δεδομένα του χρήστη.*

*(UML SEQUENCE ΕΔΩ)*

### 3.2.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: (User Fetching Data)

#### 3.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται

*Η συγκεκριμένη περίπτωση χρήσης αφορά τους Users που είναι εγγεγραμμένοι.*

#### 3.2.3.2 Προϋποθέσεις εκτέλεσης

*Ο χρήστης πρέπει να είναι εγγεγραμμένος και να διαθέτει το σωστό συνδυασμό username/ password.*

#### 3.2.3.3 Περιβάλλον εκτέλεσης

*Το περιβάλλον εκτέλεσης είναι η web εφαρμογή, αλλά ίδια λογική ακολουθείται και στη mobile εφαρμογή.*

#### 3.2.3.4 Δεδομένα εισόδου

*Τα δεδομένα εισόδου είναι το username και ο κωδικός πρόσβασης password. Επιπλέον, απαιτείται η επιλογή του είδους της επεξεργασίας επιλογή του τύπου των δεδομένων που ζητάει (Πρόβλεψη/ Τύπος ενέργειας/ Πραγματικό Φορτίο). Τέλος, πρέπει να προσδιοριστούν οι παράμετροι που αναφέρονται στο επόμενο ερώτημα.*

#### 3.2.3.5 Παράμετροι

*Οι παράμετροι είναι ο συνδυασμός username/password, ο οποίος πρέπει να είναι σωστός, ο τύπος δεδομένων που ζητείται όπως αναφέρθηκε στο 3.2.3.4, καθώς και οι παράμετροι Region, Time Period (From – Until), Resolution (διαστήματα χρόνου) και επιλογή Graph/ Table.*

#### 3.2.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

*Βήμα 1. Είσοδος στο login Page.*

*Βήμα 2. Login με εισαγωγή του ονόματος χρήστη και του κωδικού πρόσβασης. Αν δωθεί λανθασμένος κωδικός, εμφανίζεται μήνυμα λάθους και επιστρέφουμε στο βήμα 2.*

*Bήμα 3. Μετάβαση στο Homepage.*

*Βήμα 4. Επιλογή των παραμέτρων όπως προσδιορίζονται στην παράγραφο 3.2.3.5 και επιλογή Submit .*

*Βήμα 5. Αναμονή για απάντηση, αν το αίτημα είναι δεκτό, πήγαινε στο βήμα 6, αλλιώς πήγαινε στο βήμα 7.*

*Βήμα 6. Εμφάνιση των αποτελεσμάτων που ζητήθηκαν. Επιλογή να κάνει ο χρήστης reload, οπότε έχουμε επιστροφή στο βήμα 6, είτε να πατήσει back, οπότε έχουμε επιστροφή στο βήμα 3.*

*Βήμα 7. Εμφάνιση μηνύματος λάθους στον χρήστη και μετάβαση στο βήμα 3.*

#### 3.2.3.7 Δεδομένα εξόδου

*Ως δεδομένα εξόδου σε επιτυχή περίπτωση έχουμε τα ζητούμενα δεδομένα στην μορφή που ζητήθηκαν. Αλλιώς έχουμε ως έξοδο το κατάλληλο μήνυμα σφάλματος προς τον χρήστη.*

## 3.3 Απαιτήσεις επιδόσεων

Το σύστημά μας διαθέτει βάση δεδομένων η οποία περιέχει πληροφορίες για τους χρήστες, τα καταστήματα, τα προϊόντα και τις τιμές, καθώς και API και UI, ρόλος των οποίων είναι η παροχή πρόσβασης χρηστών, τρίτων εφαρμογών και γενικά εξωτερικών συστημάτων στις λειτουργίες του παρατηρητηρίου. Με βάση, λοιπόν, την αρχιτεκτονική αυτή, τα κύρια μεγέθη που αφορούν το λογισμικό μας είναι τα εξής δύο:

* Το μέγεθος των δεδομένων που είναι αποθηκευμένα στη βάση μας και αφορούν χρήστες, καταστήματα, προϊόντα και τιμές (το οποίο μέγεθος μετράται σε GB)
* Το πλήθος των ενεργών συνδέσεων με το API (είτε από το δικό μας UI, είτε από τρίτες εφαρμογές)
* Η ταχύτητα ανταλλαγής δεδομένων μέσω αυτών των συνδέσεων (από και προς το API, από και προς τη βάση δεδομένων)

Έχοντας, οπότε, υπόψη μας τα παραπάνω βασικά μεγέθη, κρίσιμες μετρικές της επιθυμητής απόδοσης του λογισμικού μας είναι οι ακόλουθες:

* Η αύξηση του μεγέθους των δεδομένων μας με το χρόνο (ή, καλύτερα, το ποσοστό αύξησης π.χ. ανά εβδομάδα, μήνα και έτος), η οποία μας βοηθά στη κατανόηση των τεχνικών αναγκών της εφαρμογής.
* Χρόνος απόκρισης σε ένα αίτημα χρήστη σε ms. Είναι πολύ σημαντικό ο συνολικός χρόνος επεξεργασίας, καθώς και το latency του δικτύου να μην επηρεάζουν αρνητικά την εμπειρία του χρήστη.
* Throughput εξυπηρέτησης αιτημάτων σε βάση δεδομένων και API. Ενδιαφέρει τόσο το πλήθος των αιτημάτων όσο και το μέγεθός τους (επιτρέπονται κλήσεις στο API που απαιτούν πολλή επεξεργασία και επιστρέφουν μεγάλο πλήθος δεδομένων).
* Η κλιμακωσιμότητα, θέλουμε δηλαδή η οποιαδήποτε αύξηση των δεδομένων λόγω της δραστηριότητας της εφαρμογής να αντισταθμίζεται γραμμικά από μία αντίστοιχη οριζόντια αύξηση των διαθέσιμων πόρων.
* Το πλήθος των σφαλμάτων που επιστρέφουν η βάση και το API, καθώς και το ποσοστό σφαλμάτων σε σχέση με το συνολικό πλήθος των αιτημάτων, με τη πάροδο του χρόνου. Περαιτέρω διάκριση των σφαλμάτων είναι επίσης σημαντική για τη συντήρηση αλλά και για τη προστασία του συστήματος από λογικά λάθη της εφαρμογής, λανθασμένη χρήση ή ακόμα και κακόβουλες επιθέσεις.
* Η διεκπεραιωτική ικανότητα της εφαρμογής, και, πιο συγκεκριμένα, τόσο το bandwidth του ίδιου του API, όσο και της βάσης, καθώς τα δύο αυτά δεν είναι άρρηκτα συνδεδεμένα.

## 3.4 Απαιτήσεις οργάνωσης δεδομένων

### 3.4.1 Τεχνική περιγραφή των δεδομένων που διαχειρίζεται το λογισμικό και των σχετικών μετρικών φορτίου δεδομένων εισόδου, επεξεργασίας κ.λπ.

*Τα δεδομένα που διαχειρίζεται το λογισμικό μας είναι κατά κύριο λόγο (σχεδόν αποκλειστικά αν εξεραιθούν τοπικά αποθηκευμένα δεδομένα που αφορούν πχ τα credentials του χρήστη σε ένα session) τα δεδομένα που τελικά θα αποθηκευτούν στη βάση δεδομένων μας.*

*Τα δεδομένα της βάσης παρουσιάζονται στο E-R διάγραμμα που παρατέθηκε προηγουμένως. Συγκεκριμένα, τα δεδομένα μας αναφέρονται σε:*

* *Δεδομένα που αφορούν τους χρήστες (admin και εγγεγραμμένοι)*
* *Δεδομένα που αφορούν την πραγματική κατανάλωση ενέργειας*
* *Δεδομένα που αφορούν προβλέψεις κατανάλωσης της επόμενης ημέρας*
* *Δεδομένα που αφορύν την παραγωγή ενέργειας ανά είδος*

*Τα δεδομένα αυτά παρέχονται στο λογισμικό μας είτε από τον οργανισμό entso, είτε από τον admin, ο οποίος μπορεί να δημιουργεί νέους χρήστες και να επεξεργάζεται τα δεδομένα που αφορούν την ενέργεια.*

*Τέλος, κάποιες μετρικές που μας ενδιαφέροτν για τη διαδικασία του storage capacity planning είιναι οι:*

* *Το μέγεθος των δεδομένων που αποθηκεύουμε στη βάση δεδομένων. Πρόκειται για την πιο σημαντική μετρική, η οποία είναι εύκολο και σημαντικό να καταγράφεται κατά την εξέλιξη της εφαρμογής, ώστε να έχουμε μια καλή εικόνα για τις απαιτήσεις του λογισμικού σε αποθηκευτικό χώρο.*
* *Το πλήθος των εσφαλμένων δεδομένων, το οποίο δύσκολα αναγνωρίζεται και είναι δυνατόν μόνο μέσω feedback από τους χρήστες.*
* *Το σύνολο των άκυρων/ελλειπών δεδομένων, δηλαδή αυτών που δεν διαθέτουν όλα τα απαιτούμενα πεδία που μας χρειάζονται. Αν και δεν περιμένουμε να έχουμε τέτοια δεδομένα από τον οργανισμό entso, λόγω της δυνατότητας επεξεργασίας τους μπορεί να δημιουργηθούν κενά.*

### 3.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

*Από την αρχή του έργου έχουμε αποφασίσει τη σχεδίαση και υλοποίηση ενός REST API. Επομένως, έχουμε ήδη προδιαγράψει πως κάποιος εξωτερικός χρήστης δεν θα έχει ποτέ άμεση πρόσβαση στα δεδομένα που διατηρεί η βαση δεδομένων και το λογισμικό. Η πρόσβαση σε αυτά γίνεται μόνο από τα γνωστά endpoints του API.*

*Εκτός από αυτό, είναι απαραίτητο να μην υπάρχει πρόσβαση στα δεδομένα που αφορούν τον χρήστη από άλλους. Για το λόγο αυτό, κάθε χρήστης της εφαρμογής απαιτείται να έχει ένα username και ένα κωδικό πρόσβασης.*

### 3.4.3 Μοντέλο δεδομένων (μοντέλο κλάσεων UML ή/και μοντέλο ER)

*Μοντέλα δεδομένων UML ή/και ER*

### 3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

### 3.4.5 Προδιαγραφές διατήρησης δεδομένων

*Απαιτήσεις διατήρησης δεδομένων σε βάθος χρόνου.*

## 3.5 Περιορισμοί σχεδίασης

*Λεπτομερής τεχνική τεκμηρίωση των περιορισμών σχεδίασης οι οποίοι επιβάλλονται από απαιτήσεις συμμόρφωσης σε πρότυπα, κανονισμούς, ή άλλους περιορισμούς του έργου. Περιλαμβάνεται η πολιτική ονοματολογίας οντοτήτων δεδομένων και πεδίων. Τέτοιοι περιορισμοί μπορεί να επιβάλλονται από τη χρήση βιβλιοθηκών, frameworks, περιβαλλόντων ανάπτυξης κλπ*

## 3.6 Λοιπές απαιτήσεις

### 3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

*Τεκμηρίωση απαιτήσεων διαθεσιμότητας*

### 3.6.2 Απαιτήσεις ασφάλειας

*Τεκμηρίωση απαιτήσεων ασφαλείας*

### 3.6.3 Απαιτήσεις συντήρησης

*Τεκμηρίωση απαιτήσεων συντήρησης*

# Παράρτημα

## 4.1 Παραδοχές και εξαρτήσεις

## 4.2 Ακρωνύμια και συντομογραφίες

## 4.3 Υποστηρικτικά έγγραφα, πρότυπα κ.λπ.