



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Έγγραφο απαιτήσεων λογισμικού (SRS)

Σύμφωνα με το πρότυπο ISO/IEC/IEEE 29148:2011

Ομάδα: WeAreBack
Σταύρος Σταύρου
Λένος Τσοκκής
Κωνσταντίνα Φουντουραδάκη

Περιεχόμενα

1	Εισαγωγή	4
1.1	Σκοπός του λογισμικού	4
1.2	Διεπαφές (interfaces)	4
1.2.1	Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού .	4
1.2.2	Διεπαφές με το χρήστη	5
2	Αναφορές	5
3	Προδιαγραφές απαιτήσεων λογισμικού	6
3.1	Περιπτώσεις χρήσης	6
3.1.1	ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Ανάκτηση δεδομένων από Απλό Χρήστη	6
3.1.1.1	Χρήστες (ρόλοι) που εμπλέκονται	6
3.1.1.2	Προϋποθέσεις εκτέλεσης	6
3.1.1.3	Περιβάλλον εκτέλεσης	6
3.1.1.4	Δεδομένα εισόδου	6
3.1.1.5	Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά . .	6
3.1.1.6	Δεδομένα εξόδου	9
3.1.2	ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Εμφάνιση κατάστασης ενός χρήστη από τον Διαχειριστή	10
3.1.2.1	Χρήστες (ρόλοι) που εμπλέκονται	10
3.1.2.2	Προϋποθέσεις εκτέλεσης	10
3.1.2.3	Περιβάλλον εκτέλεσης	10
3.1.2.4	Δεδομένα εισόδου	10
3.1.2.5	Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά . .	10
3.2	Απαιτήσεις επιδόσεων	13
3.3	Απαιτήσεις οργάνωσης δεδομένων	13
3.3.1	Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα	13
3.4	Περιορισμοί σχεδίασης	14
3.5	Λοιπές απαιτήσεις	14
3.5.1	Απαιτήσεις διαθεσιμότητας λογισμικού	14
3.5.2	Απαιτήσεις ασφάλειας	14
3.5.3	Απαιτήσεις συντήρησης	14
4	Παράρτημα: ακρωνύμια και συντομογραφίες	15

Διαγράμματα UML

1	Deployment diagram	4
2	Component diagram	5
3	Activity diagram Απλού Χρήστη	7
4	Sequence diagram εισόδου Απλού Χρήστη	8
5	Sequence diagram ανάκτησης δεδομένων από Απλό Χρήστη	9
6	Activity diagram Διαχειριστή, όπου ζητάει πληροφορίες για κάποιο χρήστη	11
7	Sequence diagram ανάκτησης πληροφοριών για κάποιο χρήστη από τον Διαχειριστή	12

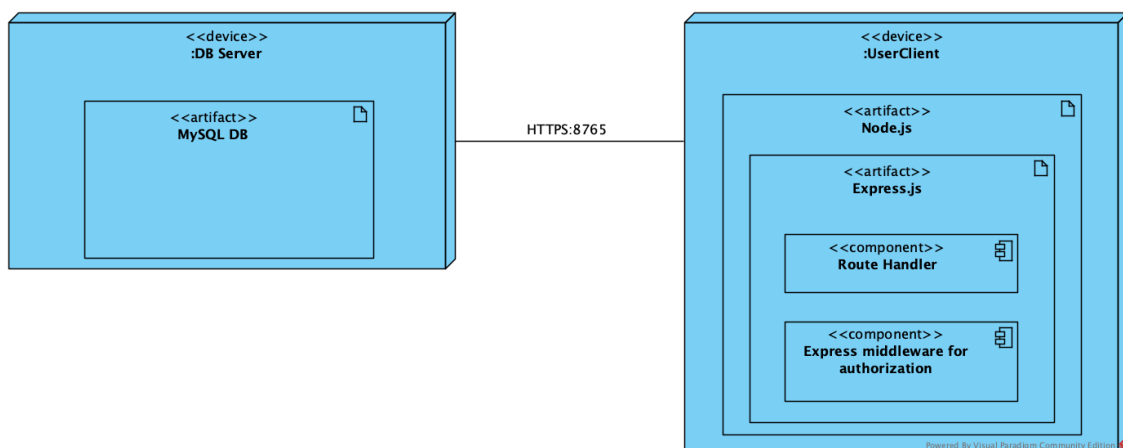
1 Εισαγωγή

1.1 Σκοπός του λογισμικού

Ο σκοπός της ομάδας WeAreBack είναι η υλοποίηση μιας πλατφόρμας παρατήρησης κατανάλωσης/παραγωγής ενέργειας στην οποία μπορεί οποιοσδήποτε να έχει ελεύθερη πρόσβαση. Απευθύνεται κυρίως σε εταιρίες παραγωγής ή παροχής ενέργειας, σε Υπουργεία Ενέργειας εντός Ευρωπαϊκής Ένωσης, στον απλό πολίτη και στους δημοσιογράφους. Η πλατφόρμα θα προσφέρει στους χρήστες υπηρεσίες όπως πρόσβαση σε ένα μεγάλο σύνολο δεδομένων παραγωγής και κατανάλωσης ενέργειας, προβλέψη της μελλοντικής κατανάλωσης ανά τύπο ενέργειας και σύγκριση της προβλεπόμενης με την πραγματική κατανάλωση. Επιπλέον, πρωτεύοντες στόχοι είναι η ευχρηστία πλατφόρμας, η τεκμηρίωση και η συντήρηση του κώδικα. Τέλος, αφορμή για την υλοποίηση του συστήματος συνιστούν η αναγκαιότητα της ενέργειας και η διαχείριση της, λόγοι για τους οποίους η πλατφόρμα θα μπορεί να διαδοθεί ευρέως.

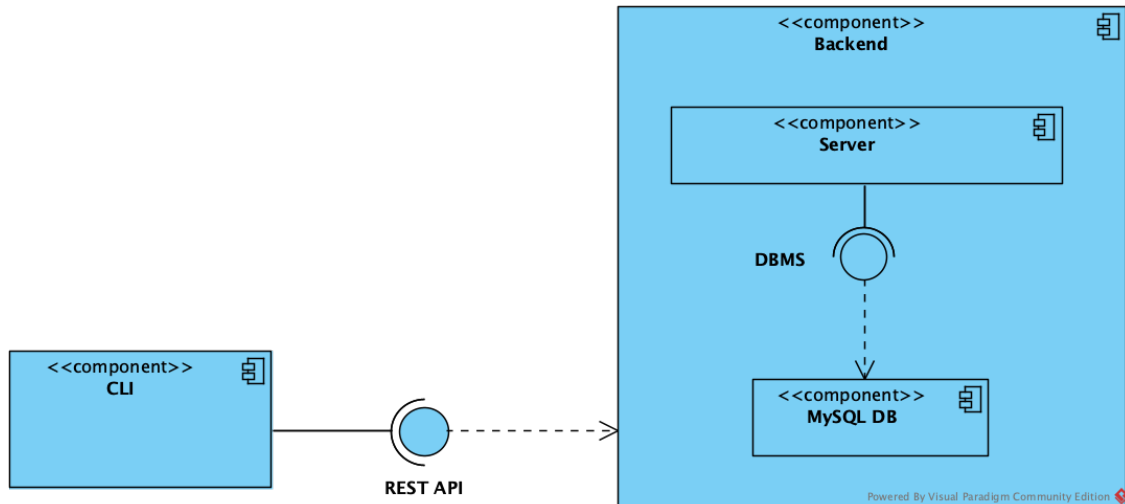
1.2 Διεπαφές (interfaces)

1.2.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού



Σχήμα 1: Deployment diagram

1.2.2 Διεπαφές με το χρήστη



Σχήμα 2: Component diagram

2 Αναφορές

- [1] ISO/IEC 29148:2011 (IEEE Std 29148-2011), Systems and software engineering - Life cycle processes - Requirements engineering

Πηγές Πληροφοριών

<https://www.visual-paradigm.com/guide/>

<https://swagger.io/specification/>

<https://medium.com/@mihaigeorge.c/web-rest-api-benchmark-on-a-real-life-application-ebb743a5d7a3>

<https://medium.com/hackernoon/back-end-performance-those-metrics-we-care-about-ade678e87969>

3 Προδιαγραφές απαιτήσεων λογισμικού

3.1 Περιπτώσεις χρήσης

3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Ανάκτηση δεδομένων από Απλό Χρήστη

3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Στη συγκεκριμένη περίπτωση εμπλέκεται μόνο ο Απλός Χρήστης, ο οποίος συνδέεται στην πλατφόρμα και ανακτά δεδομένα ενέργειας από τη βάση δεδομένων.

3.1.1.2 Προϋποθέσεις εκτέλεσης

Προϋποθέσεις πρόσβασης του Χρήστη στα δεδομένα που επιθυμεί, είναι η σύνδεση του στην πλατφόρμα, η διαπίστευση του (user access token), να έχει επαρκή quota και να έχει διατυπώσει σωστά την εντολή στο CLI. Μια επιπλέον προϋπόθεση είναι η πλήρης συνδεσιμότητα του χρήστη με τη βάση δεδομένων, που μπορεί να ελεγχθεί μέσω του endpoint HealthCheck.

3.1.1.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης της περίπτωσης χρήσης είναι το CLI, το οποίο είναι διαθέσιμο μόνο από την κονσόλα (command line, ssh) του συστήματος που φιλοξενεί την εφαρμογή.

3.1.1.4 Δεδομένα εισόδου

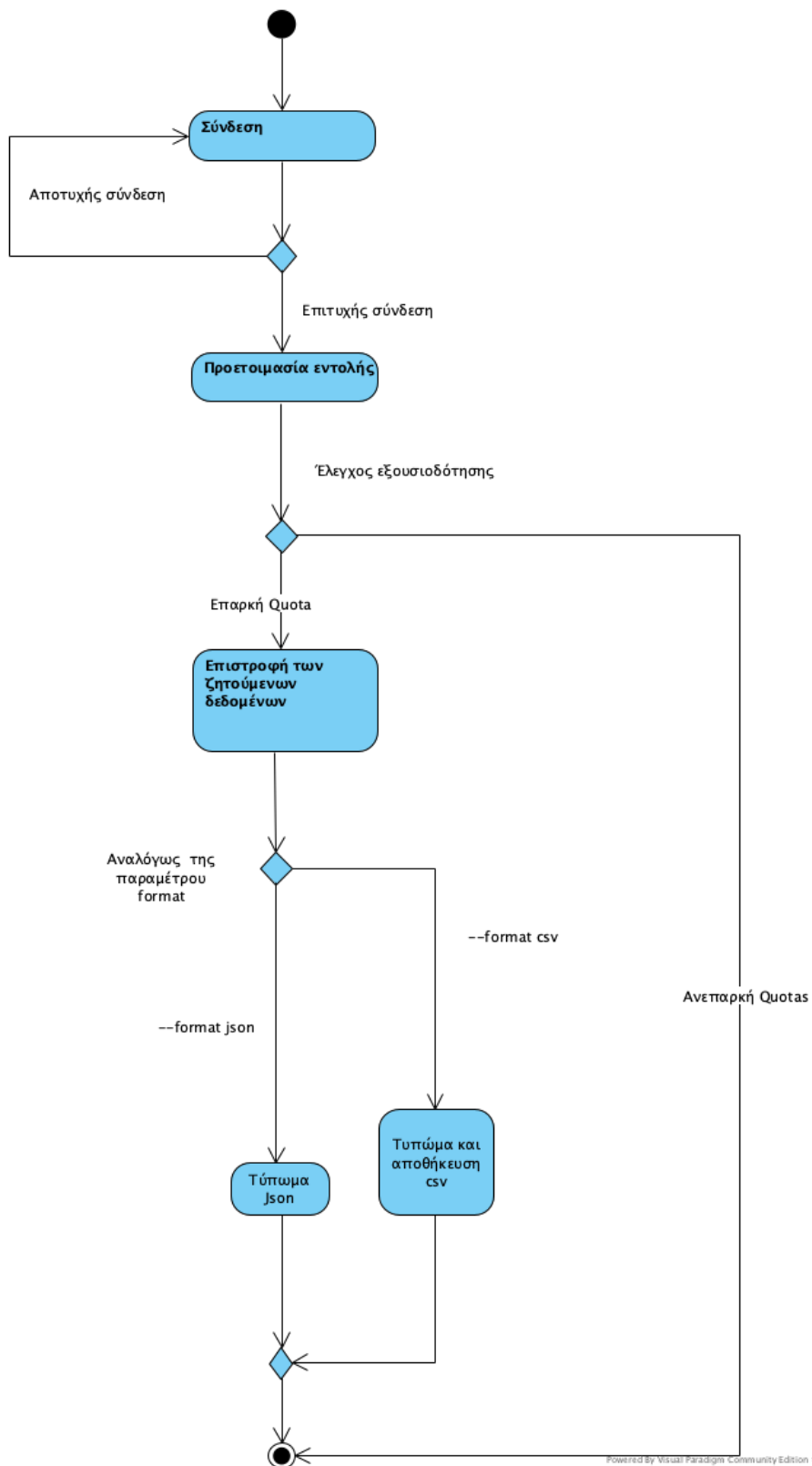
Ως δεδομένα εισόδου θεωρούμε το username και password που εισάγει ο χρήστης προκειμένου να συνδεθεί στην πλατφόρμα, καθώς και η εντολή που δίνει στο CLI.

Οι συνθήκες εγκυρότητας για τα παραπάνω είναι η ταυτοποίηση του χρήστη και σωστή χρήση των παραμέτρων που δίνονται σε μία κλήση (διαφορετικά το API επιστρέφει HTTP Status Code 400, "Bad Request"). Οι συνθήκες εγκυρότητας εξόδου είναι η ύπαρξη των δεδομένων που ζήτησε ο χρήστης (διαφορετικά το API επιστρέφει HTTP Status Code 403, "No data").

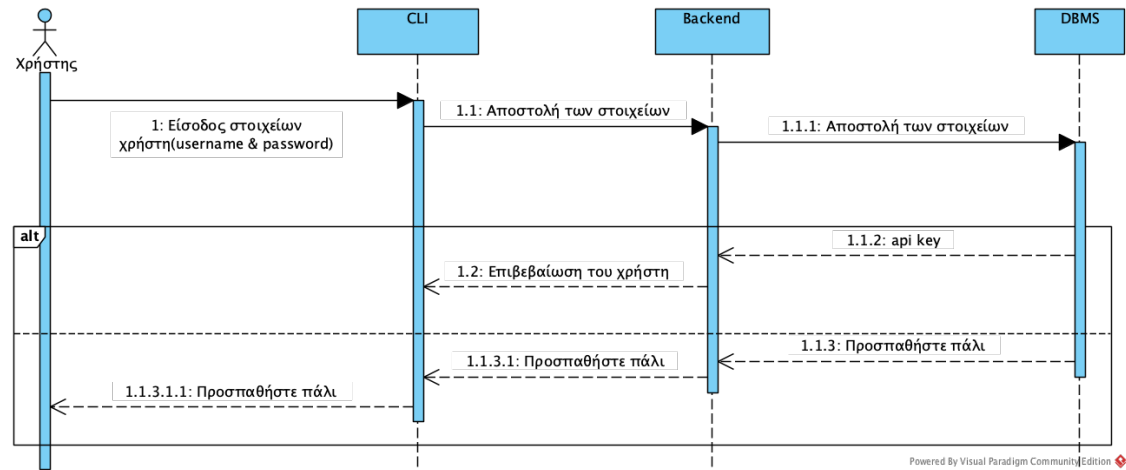
3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Αλληλουχία ενεργειών:

1. Πρόσβαση στο CLI
2. Εντολή εισόδου συνοδευόμενη από τα στοιχεία εισόδου
3. Επιστρέφεται το τοκεν στον χρήστη
4. Εντολή για προσπέλαση δεδομένων
5. Επιστρέφονται στον χρήστη τα ζητούμενα δεδομένα είτε με μορφή JSON είτε csv



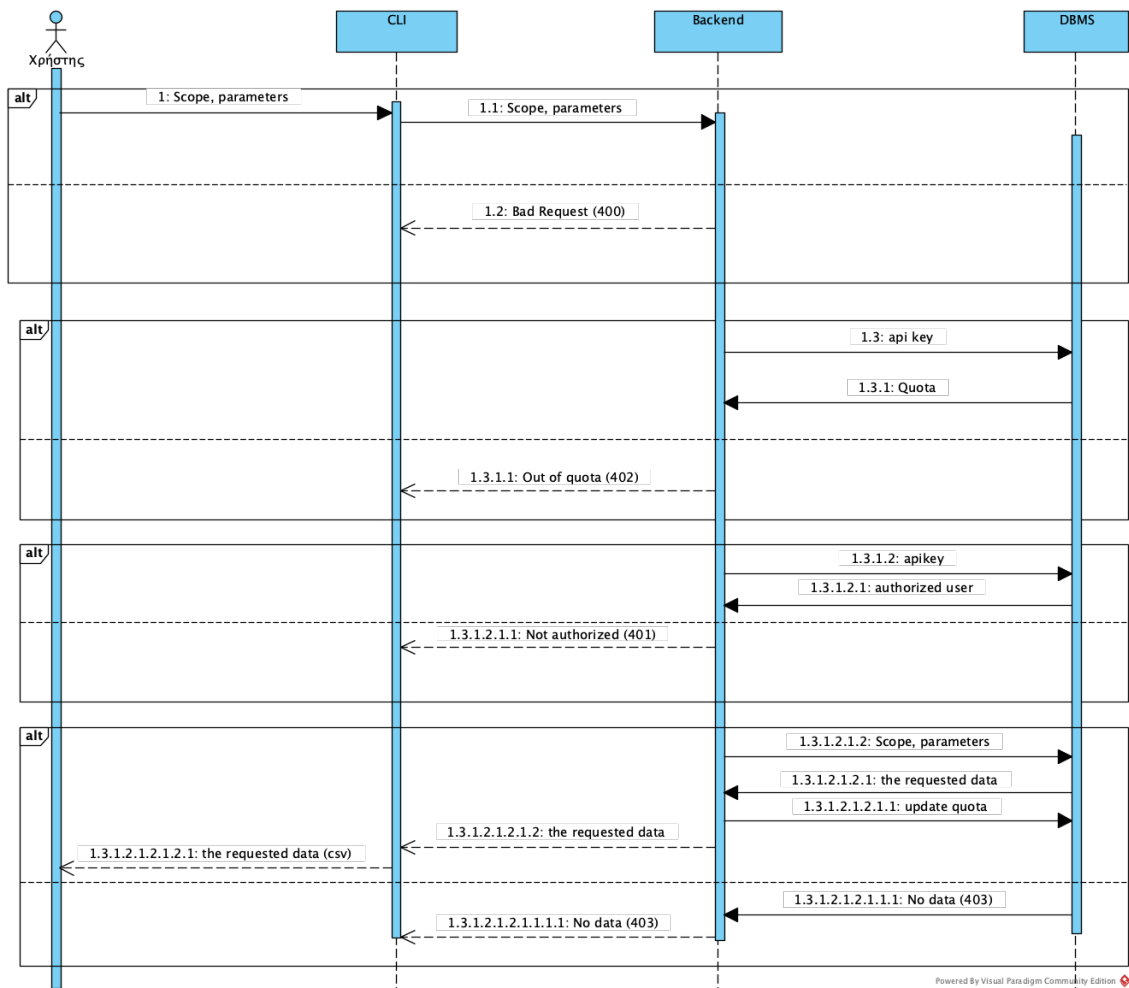
Σχήμα 3: Activity diagram Απλού Χρήστη



Σχήμα 4: Sequence diagram εισόδου Απλού Χρήστη

Αλληλουχία ενεργειών για προσπέλαση δεδομένων:

1. Εντολή για προσπέλαση δεδομένων
2. Ελέγχεται αν η μορφή της εντολής είναι έγκυρη
3. Ελέγχεται αν ο χρήστης έχει διαθέσιμα quota
4. Ελέγχεται αν ο χρήστης είναι εξουσιοδοτημένος για αυτή την ενέργεια
5. Επιστρέφονται στον χρήστη τα ζητούμενα δεδομένα είτε με μορφή JSON είτε csv, αν υπάρχουν και μειώνεται το quota



Σχήμα 5: Sequence diagram ανάκτησης δεδομένων από Απλό Χρήστη

3.1.1.6 Δεδομένα εξόδου

Εάν η παράμετρος format της εντολή στο CLI, έχει οριστεί ως csv τότε δημιουργείται ένα αρχείο .csv που περιέχει τα δεδομένα που ζήτησε ο χρήστης.

3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Εμφάνιση κατάστασης ενός χρήστη από τον Διαχειριστή

3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Στη συγκεκριμένη περίπτωση εμπλέκεται ο Διαχειριστής, ο οποίος συνδέεται στην πλατφόρμα και ο Χρήστης του οποίου τα στοιχεία κατάστασης θέλει να προσπελάσει ο Διαχειριστής από τη βάση δεδομένων.

3.1.2.2 Προϋποθέσεις εκτέλεσης

Προϋποθέσεις πρόσβασης του Διαχειριστή στη βάση δεδομένων για να ανακτήσει την κατάσταση κάποιου χρήστη, είναι η σύνδεση του στην πλατφόρμα, η διαπίστευση του ως διαχειριστή (user access token) και να έχει διατυπώσει σωστά την εντολή στο CLI. Μια επιπλέον προϋπόθεση είναι η πλήρη συνδεσιμότητα του χρήστη με τη βάση δεδομένων, που μπορεί να ελεγχθεί μέσω του endpoint HealthCheck.

3.1.2.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης της περίπτωσης χρήσης είναι το CLI, το οποίο είναι διαθέσιμο μόνο από την κονσόλα (command line, ssh) του συστήματος που φιλοξενεί την εφαρμογή.

3.1.2.4 Δεδομένα εισόδου

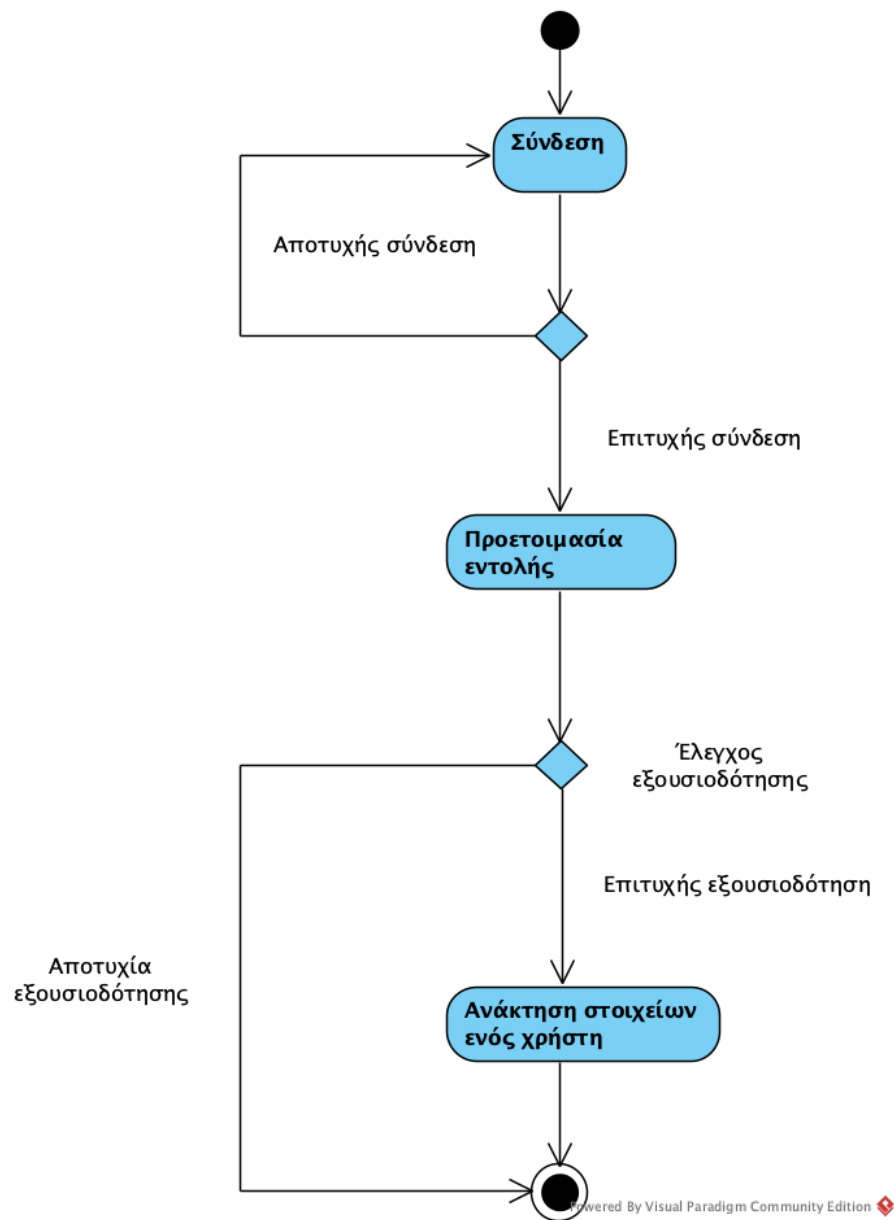
Ως δεδομένα εισόδου θεωρούμε το username και password που εισάγει ο Διαχειριστής προκειμένου να συνδεθεί στην πλατφόρμα, καθώς και η εντολή που δίνει στο CLI.

Οι συνθήκες εγκυρότητας για τα παραπάνω είναι η ταυτοποίηση του ως Διαχειριστή και σωστή χρήση των παραμέτρων που δίνονται σε μία κλήση (διαφορετικά το API επιστρέφει HTTP Status Code 400, "Bad Request"). Οι συνθήκες εγκυρότητας εξόδου είναι η ύπαρξη του χρήστη που αντιστοιχεί στο δοθέν username (διαφορετικά το API επιστρέφει HTTP Status Code 403, "No data").

3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Αλληλουχία ενεργειών:

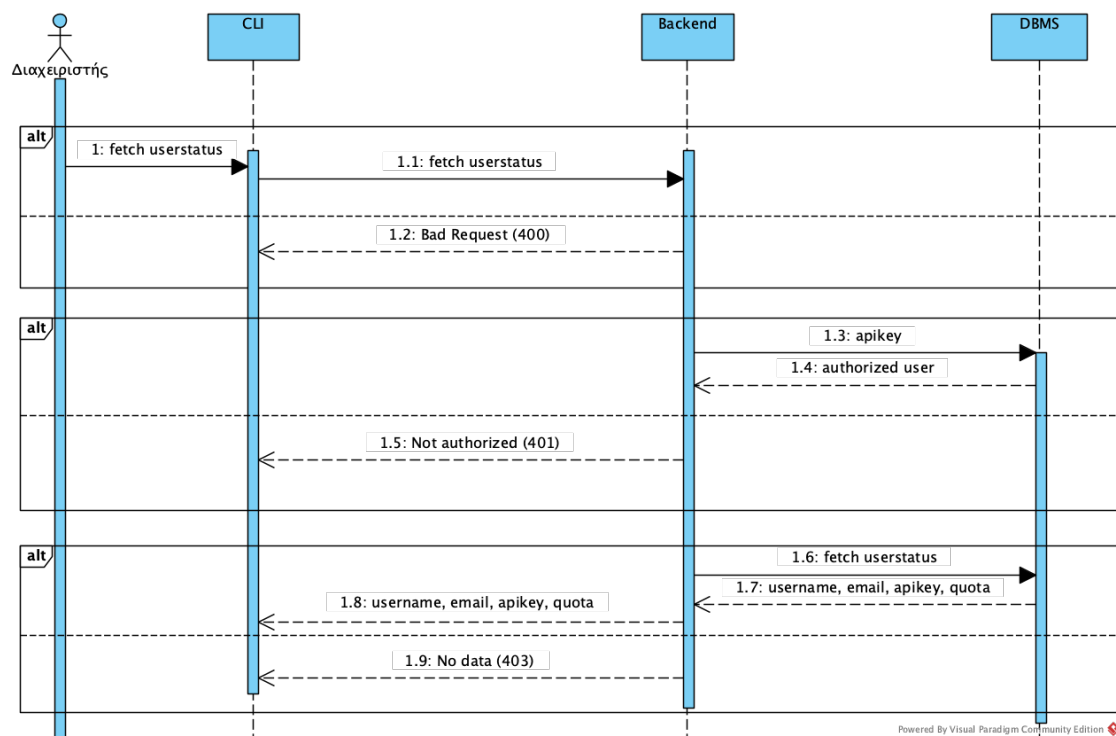
1. Πρόσβαση στο CLI
2. Εντολή εισόδου συνοδευόμενη από τα στοιχεία εισόδου
3. Επιστρέφεται το τοκεν στον χρήστη
4. Εντολή για προσπέλαση των στοιχείων κάποιου χρήστη
5. Επιστρέφονται στον διαχειριστή τα ζητούμενα δεδομένα



Σχήμα 6: Activity diagram Διαχειριστή, όπου ζητάει πληροφορίες για κάποιο χρήστη

Αλληλουχία ενεργειών για προσπέλαση δεδομένων ενός χρήστη:

1. Εντολή για προσπέλαση δεδομένων χρήστη
2. Ελέγχεται αν η μορφή της εντολής είναι έγκυρη
3. Ελέγχεται αν είναι όντως ο διαχειριστής, δηλαδή αν είναι εξουσιοδοτημένος για αυτή την ενέργεια
4. Επιστρέφονται στο διαχειριστή τα ζητούμενα δεδομένα, αν υπάρχουν



Σχήμα 7: Sequence diagram ανάκτησης πληροφοριών για κάποιο χρήστη από τον Διαχειριστή

3.2 Απαιτήσεις επιδόσεων

Έχοντας κατά νου ότι το λογισμικό μας θα είναι συνεχώς διαθέσιμο στους χρήστες και πως σε μια εφαρμογή πραγματικού κόσμου σχεδόν όλα τα αιτήματα αλληλεπιδρούν με τη βάση δεδομένων, υλοποιήσαμε την πλατφόρμα χρησιμοποιώντας Node.js με Express JS που έχουν πολύ υψηλή απόδοση ακόμα για πολλά αιτήματα ανά δευτερόλεπτο. Ενδεικτικά, ένας διακομιστής με καλή σχέση κόστους-απόδοσης θα ήταν με 4 πυρήνες και 8 GB μνήμη¹.

Ακολουθούν οι μετρικές που γενικά μας ενδιαφέρουν για την αξιολόγηση του back-end.

- **Throughput:** υποδεικνύει τον όγκο της κυκλοφορίας που διαχειρίζεται μια υπηρεσία. Γενικά υπό υψηλή πίεση, ένας διακομιστής είναι γενικά βραδύτερος από αυτόν σε κανονικό φορτίο.
- **Latency:** υποδεικνύει το χρόνο ολοκλήρωσης μίας λειτουργίας I/O. Στην περίπτωση μας, αναμένουν ανατροφοδότηση σε πραγματικό χρόνο μετά από κάθε είσοδο ενός χαρακτήρα. Σχετίζεται με το throughput, δηλαδή η αύξηση του throughput συνοδεύεται από παράλληλη αύξηση του latency.
- **Error rate:** τα error που δημιουργούνται από ζητήματα δικτύου (π.χ. συμφόρηση) δεν πρέπει να υπερβαίνουν το 5% των συνολικών αιτημάτων και τα error που προκλήθηκαν από την εφαρμογή δεν πρέπει να υπερβαίνουν το 1%.

3.3 Απαιτήσεις οργάνωσης δεδομένων

3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Υπάρχουν δύο διαφορετικές κατηγορίες χρηστών, εκάστη με τις δικές της απαιτήσεις και περιορισμούς πρόσβασης.

- **Απλός Χρήστης:** Ο Απλός Χρήστης έχει πρόσβαση στα δεδομένα παραγωγής, κατανάλωσης, εκτιμώμενης κατανάλωσης και στις συγκρίσεις τους. Όμως η πρόσβαση του στα δεδομένα περιορίζεται σε συγκεκριμένο πλήθος κλήσεων της εφαρμογής ανά ημέρα (quota).
- **Διαχειριστής:** Ο Διαχειριστής έχει τη δυνατότητα να παρακολουθεί και να διαμορφώνει όλα τα δεδομένα στη βάση δεδομένων, είτε πρόκειται για εγγραφές ενέργειας είτε χρηστών.

¹<https://medium.com/@mihaigeorge.c/web-rest-api-benchmark-on-a-real-life-application-ebb743a5d7a3>

3.4 Περιορισμοί σχεδίασης

Αρχικά αναφέρουμε τα εργαλεία που χρησιμοποιήσαμε για την υλοποίηση της πλατφόρμας.

- **MySQL** για τη σχεσιακή βάση δεδομένων
- **JavaScript**
- **Node.js** framework της JavaScript
- **Express** framework για το Node.js
- **npm** για το build automation
- **SSH** για πρόσβαση στον σερερ

3.5 Λοιπές απαιτήσεις

3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Οι απλοί χρήστες, που είναι εγγεγραμμένοι, έχουν δυνατότητες προσπέλασης δεδομένων και ελέγχου της συνδεσιμότητας (end-to-end connectivity) τους με τη βάση δεδομένων (HealthCheck). Ο διαχειριστής του συστήματος έχει δυνατότητες δημιουργίας και επεξεργασίας χρήστη, εμφάνισης κατάστασης δοσμένου χρήστη, προσθήκης νέων δεδομένων στη βάση από έγγραφο .csv, «εκκαθάρισης» της βάσης από κάθε δεδομένο πέραν του διαχειριστικού λογαριασμού (reset) και ελέγχου της συνδεσιμότητας (end-to-end connectivity) του με τη βάση δεδομένων (HealthCheck).

3.5.2 Απαιτήσεις ασφάλειας

Το σύστημα υποστηρίζει το πρωτόκολλο HTTPS για όλες τις χρηστικές και προγραμματιστικές διεπαφές μέσω self-signed certificate, για να διασφαλίσει ότι τα προσωπικά δεδομένα των χρηστών είναι απροσπέλαστα. Χρησιμοποιεί επίσης ένα επιπλέον επίπεδο ασφαλείας, με hash authentication, για τους κωδικούς πρόσβασης των χρηστών. Τέλος, αν και τα δεδομένα που επιστρέφονται από τα endpoints ανάκτησης και αναζήτησης δεδομένων του API προσφέρονται ως ανοικτά δεδομένα (Open Data), για λόγους ελέγχου της κατανάλωσης πόρων του συστήματος που τα διαθέτει, για τη χρήση του API θα απαιτείται διαπίστευση των χρηστών. Συνεπώς, κατά την κλήση του API παρέχεται το user access token, κωδικοποιημένο μέσα σε custom HTTP Header, το όνομα του οποίου είναι XOBSERVATORY-AUTH.

3.5.3 Απαιτήσεις συντήρησης

Πρέπει ανά τακτά διαστήματα να ενημερώνεται το λογισμικό και η βάση δεδομένων, ώστε το σύστημα να παραμένει συμβατό με τις νέες εκδόσεις των γλωσσών και των τεχνολογικών εργαλείων που χρησιμοποιεί. Ακόμη, απαιτείται η ύπαρξη ενός εφεδρικού διακομιστή ώστε η πλατφόρμα να είναι διαρκώς διαθέσιμη, ακόμα και όταν ο server χρήζει συντήρησης.

4 Παράρτημα: αββωνύμια και συντομογραφίες

API Application Programming Interface

REST API RESTful Application Programming Interface

CLI Command Line Interface

JSON JavaScript Object Notation

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

SSH Secure Shell

SQL Structured Query Language

JS JavaScript

npm Node Package Manager

UML Unified Modeling Language

SRS Software Requirements Specification