

Έγγραφο απαιτήσεων λογισμικού (SRS)

ΑΚΟΛΟΥΘΩΝΤΑΣ ΤΟ ΠΡΟΤΥΠΟ ISO/IEC/IEEE 29148:2011

Electromarket

1. Εισαγωγή

1.1 Σκοπός του λογισμικού

Σκοπός του project 48HourProject είναι η ανάπτυξη λογισμικού που αναλαμβάνει τη διαχείριση δεδομένων της αγοράς ηλεκτρικής ενέργειας στην Ευρώπη. Τα δεδομένα αυτά διατίθενται ελεύθερα (Open Data) από το δικτυακό τόπο transparency.entsoe.eu, ενώ είναι κρίσιμα για τη διαφάνεια στη λειτουργία της αγοράς ηλεκτρικής ενέργειας. Η διάθεση, οπτικοποίηση και ανάλυσή τους είναι απαραίτητες υπηρεσίες προς τους συμμετέχοντες στην αγορά, ανεξάρτητα από το ρόλο τους (παραγωγοί, πωλητές, συνεταιρισμοί κ.ά.).

Με το παρών λογισμικό παρέχεται στο χρήστη ένα εύχρηστο εργαλείο για την προβολή των δεδομένων, το οποίο διασφαλίζει μία ευχάριστη εμπειρία χρήσης. Για το σχεδιασμό του site έχει δοθεί ιδιαίτερη προσοχή στο user experience, ώστε να είναι εύκολη αλληλεπίδραση με το χρήστη.

1.2 Διεπαφές (interfaces)

1.2.1 Διεπαφές με εξωτερικά συστήματα

Για την καλύτερη εμπειρία των χρηστών το λογισμικό χρησιμοποιεί και εξωτερικά API's, τα οποία αναφέρονται παρακάτω:

- **Google Maps Platform API:** Η χρήση του συγκεκριμένου API παρέχει στο χρήστη μία γραφική αναπαράσταση της περιοχής για την οποία έγινε η αναζήτηση.

Το σύστημα λειτουργεί με την υποστήριξη των παρακάτω εσωτερικών διεπαφών:

- **MySQL RDBMS Server:** Το σύστημα που επιλέχθηκε για τη διαχείριση των δεδομένων του συστήματος.

Η διεπαφή με τον χρήστη παρέχεται με δύο διαφορετικούς τρόπους:

1. **RESTful API:** Βασίζεται στο πρότυπο αρχιτεκτονικής σχεδίασης REST(REpresentational State Transfer) και χρησιμοποιεί HTTP requests (GET, PUT, POST, DELETE, PATCH). Ο χρήστης δύναται να "χτυπήσει" το επιθυμητό url και να προκύψουν τα αποτελέσματα. Η διαδικασία αυτή αυτοματοποιείται με τη χρήση του CLI.

- 2. Frontend:** Μέσα στον ιστότοπο της εφαρμογής, με τη χρήση browser, ο χρήστης θα έχει τη δυνατότητα εγγραφής και προβολής γραφικής αναπαράστασης των δεδομένων

UML

1.2.2 Διεπαφές με το χρήστη

Οι χρήστες διακρίνονται σε 3 κατηγορίες:

- 1. Επισκέπτης:** Πρόκειται για χρήστη που χρησιμοποιεί ανώνυμα το front-end μέσω web-browser.
- 2. Εγγεγραμμένος χρήστης:** Πρόκειται για χρήστες που έχουν κάνει log-in και έχουν πρόσβαση τόσο στο front-end όσο και στο cli.
- 3. Διαχειριστής:** Πρόκειται για χρήστη που είναι υπεύθυνος για τους χρήστες και έχει δικαίωμα να αλλάξει ρόλους και χαρακτηριστικά των χρηστών, αλλά και να διαχειριστεί τις εγγραφές στη βάση δεδομένων.

2. Αναφορές - πηγές πληροφοριών

3. Προδιαγραφές απαιτήσεων λογισμικού

3.1 Περιπτώσεις χρήσης

3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1:

Αναζήτηση πραγματικής κατανάλωσης σε συγκεκριμένη περιοχή και χρονική περίοδο

3.1.1.1 Χρήστες που εμπλέκονται

Στη λειτουργία αυτή έχει πρόσβαση οποιοσδήποτε χρήστης, εγγεγραμμένος ή μη.

3.1.1.2 Προϋποθέσεις εκτέλεσης

Πρόσβαση στο διαδίκτυο, ο web server ενεργός, ύπαρξη καταχωρήσεων στη βάση.

3.1.1.3 Περιβάλλον εκτέλεσης

- REST API
- Web application

3.1.1.4 Δεδομένα εισόδου

Ως φίλτρα της αναζήτησης τίθενται τα παρακάτω:

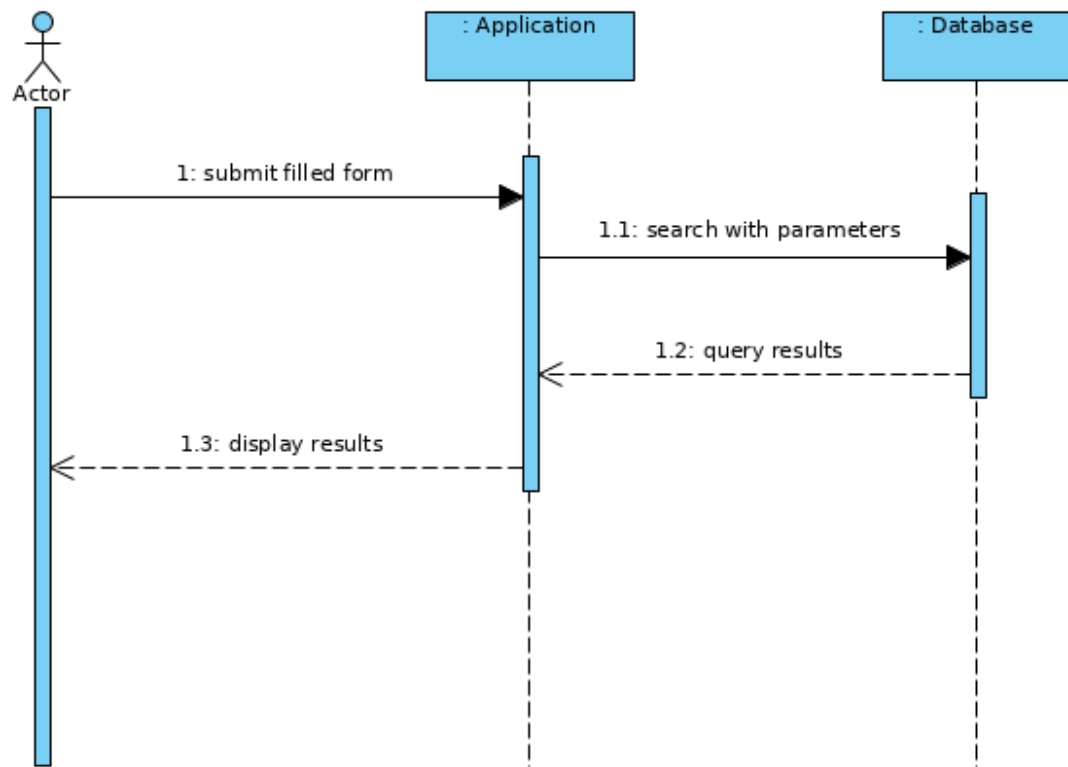
- Year
- Month
- Day
- AreaCodeText

3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

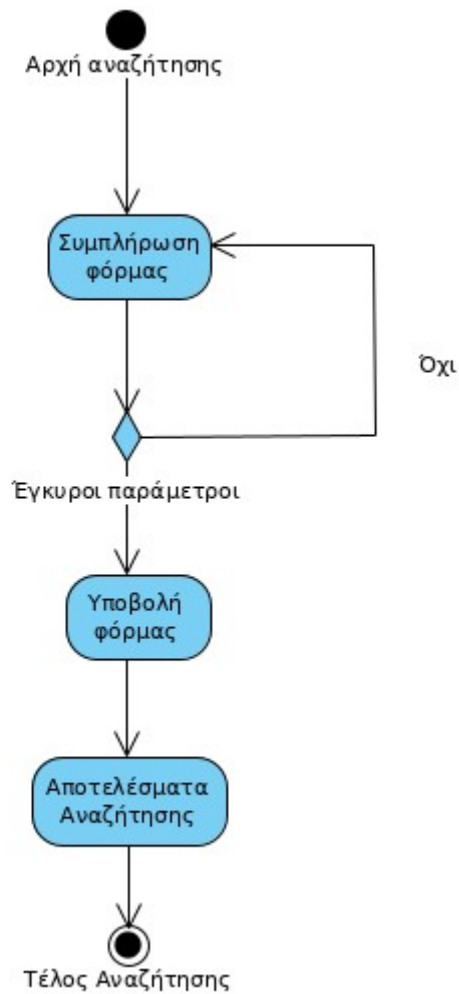
1. Εισαγωγή από τον χρήστη τις τιμές για τις παραμέτρους αναζήτησης.
2. Υποβολή φόρμας.
3. Επιστροφή δεδομένων που αντιστοιχούν στα φίλτρα της αναζήτησης.

Ακολουθούν τα UML sequence και activity διαγράμματα αντίστοιχα:

Sequence diagram



Activity diagram



3.1.1.7 Δεδομένα εξόδου

Ως έξοδος θα είναι τα αποτελέσματα της αναζήτησης που πραγματοποίησε ο χρήστης.

3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Διαχείριση πλατφόρμας(Edit)

3.1.2.1 Χρήστες που εμπλέκονται

- Διαχειριστής

3.1.2.2 Προϋποθέσεις εκτέλεσης

- Προσθήκη/Επεξεργασία/Έγκυρη ενημέρωση δεδομένων

3.1.2.3 Περιβάλλον εκτέλεσης

Web application (Admin page)

3.1.2.4 Δεδομένα εισόδου

Ο διαχειριστής συνδέεται στην πλατφόρμα με τα μοναδικά username και password του ,που του δίνουν επαυξημένα δικαιώματα .

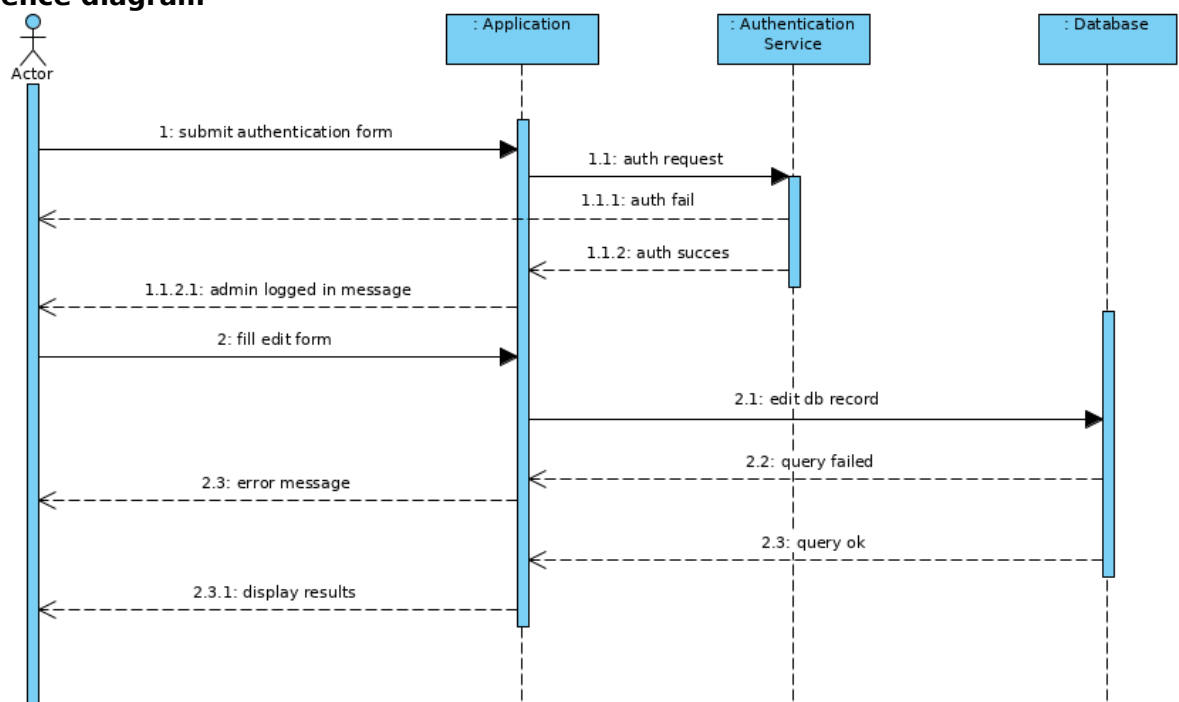
3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Αρχικά ο διαχειριστής εισάγει τα στοιχεία του και ταυτοποιείται. Μετά τη σύνδεση έχει δικαίωμα να εφαρμόσει τροποποιήσεις.

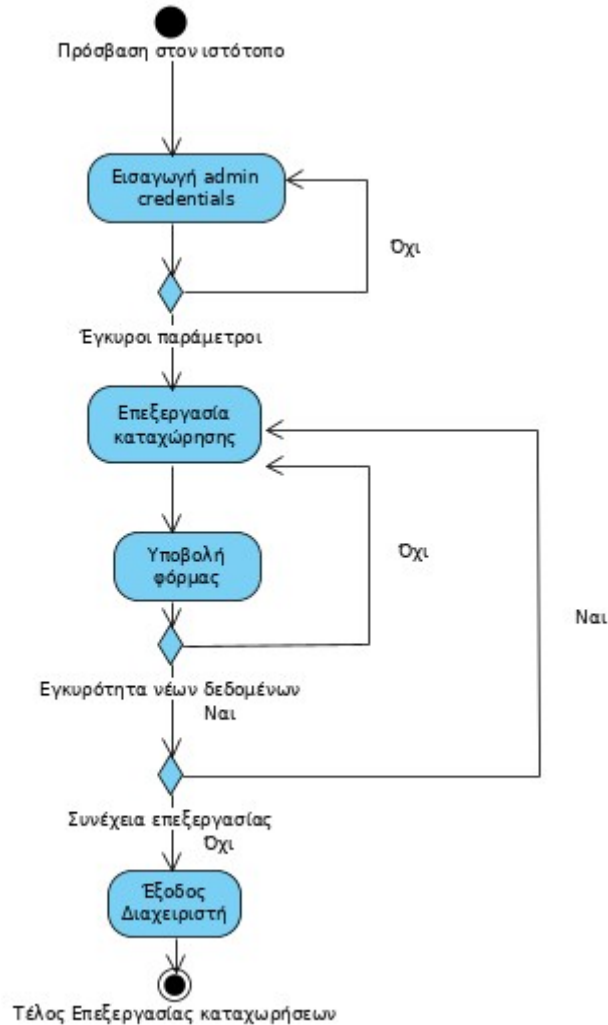
Στην συνέχεια, επεξεργάζεται τα δεδομένα και τελικά αποσυνδέεται.

Ακολουθούν τα αντίστοιχα sequence και activity diagrams.

Sequence diagram



Activity Diagram



3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Εισαγωγή δεδομένων απο διαχειριστή(Insert)

3.1.3.1 Χρήστες που εμπλέκονται

- Διαχειριστές

3.1.3.2 Προϋποθέσεις εκτέλεσης

Κάποιος admin (με δικαιώματα read & write) κάνει login με τα μοναδικά στοιχεία του. Συμπληρώνει τα δεδομένα για τη ζήτηση και πώληση ενέργειας σε συγκεκριμένη ζώνη αγοράς και τα κάνει διαθέσιμα στην πλατφόρμα. Έπειτα αποσυνδέεται.

3.1.3.3 Περιβάλλον εκτέλεσης

Web application (Admin page)

3.1.3.4 Δεδομένα εισόδου

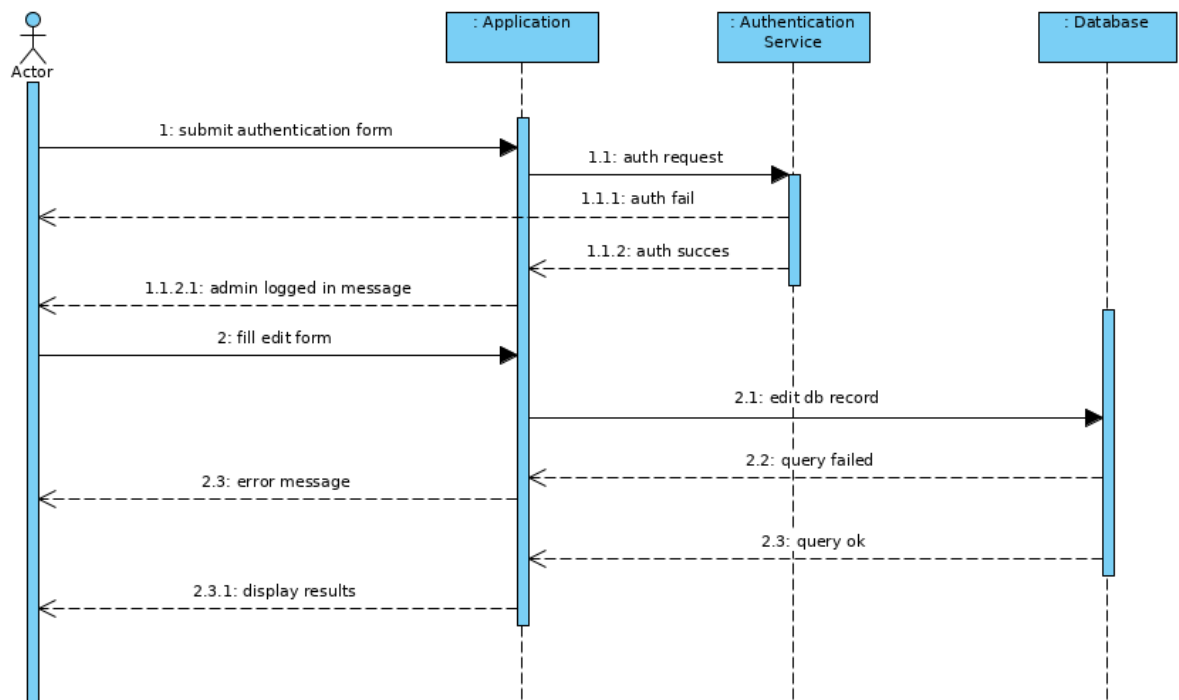
Ο διαχειριστής συνδέεται στην πλατφόρμα με τα μοναδικά username και password του ,που του δίνουν επαυξημένα δικαιώματα .

3.1.3.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

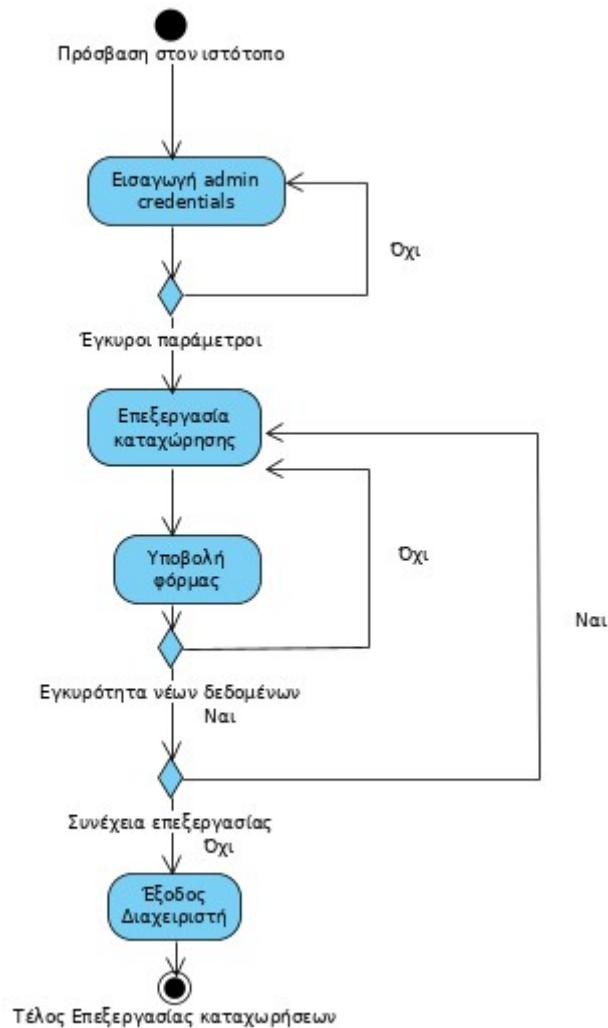
- Είσοδος (login)
- Έλεγχος στοιχείων (username/password)
- Συμπλήρωση δεδομένων
- Upload στη βάση και διαθέσιμα στην πλατφόρμας

Ακολουθούν τα αντίστοιχα sequence και activity diagrams.

Sequence Diagram



Activity Diagram



3.2 Απαιτήσεις επιδόσεων

Η πλατφόρμα θα πρέπει να είναι συνεχώς διαθέσιμη προς τους χρήστες με σωστά και έγκυρα δεδομένα. Οφείλει να μπορεί να “αντέξει” αρκετούς επισκέπτες ταυτόχρονα.

3.3 Απαιτήσεις οργάνωσης δεδομένων

Γενικά: Τα δεδομένα που παρέχει το λογισμικό μας είναι αποθηκευμένα σε βάση δεδομένων. Παρέχονται δεδομένα για γεωγραφικές περιοχές και χρονικές περιόδους.

3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Οι διαχειριστές : Έχουν το δικαίωμα να διαβάσουν τα δεδομένα(Read) αλλά και να τροποποιήσουν αυτά (WRITE) αν έχει προκύψει κάποιο λάθος

Οι χρήστες : Περιορισμένα δικαιώματα που αφορούν μόνο την ανάγνωση δεδομένων.

Συγκεκριμένα μπορούν να δουν τα δεδομένα που προσφέρονται για περιοχές/αγορές και χρονικές περιόδους που τους ενδιαφέρουν.

3.4 Περιορισμοί σχεδίασης

Ολοκληρωμένες οι απαιτήσεις θα προκύψουν μετά την ολοκλήρωση της κατασκευής του λογισμικού

3.5 Λοιπές απαιτήσεις

3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Θα χρησιμοποιηθεί το framework Flask για την υλοποίηση του back-end συστήματος και το framework Angular για το front-end σύστημα.

3.5.2 Απαιτήσεις ασφάλειας

3.5.3 Απαιτήσεις συντήρησης