**Έγγραφο απαιτήσεων λογισμικού (SRS)**

**1. Εισαγωγή**

**1.1 Εισαγωγή: σκοπός του λογισμικού**

Κεντρικός στόχος του συστήματος που υλοποιούμε είναι η δημιουργία μια ενιαίας και εύκολης στη χρήση πλατφόρμας που ενθαρρύνει τη συνεργατική παρατήρηση των δεδομένων της αγοράς ηλεκτρικής ενέργειας στην Ευρώπη και παρέχει αυτές τις πληροφορίες δωρεάν σε κάθε χρήστη. Επιπλέον, θα παρέχεται η δυνατότητα πολυκριτηριακής και ομαδοποιημένης αναζήτησής τους.

(επισυναψη deployment diagram;;)

**1.2 Διεπαφές(interfaces)**

**1.2.1 Διεπαφές με εξωτερικά συστήματα**

Το μοναδικό εξωτερικό σύστημα που χρησιμοποιείται είναι ο δικτυακός τόπος transparency.entsoe.eu ο οποίος διαθέτει τα δεδομένα, με ελεγχόμενη πρόσβαση μέσω ενός RESTful Web API. Η κλήση του API είναι ήδη προσαρμοσμένη στο Framework που χρησιμοποιείται, μέσω συγκεκριμένων κλάσεων και μεθόδων αυτών, αλλά χρειάζεται και η μεταφόρτωση δεδομένων από το RESTful Web API.

Η επικοινωνία μεταξύ του Frontend και του Backend γίνεται μέσω ενός REST API, το οποίο βοηθά στην παράλληλη εξέλιξη των δύο μέσω προδιαγραφών και μόνο. Αυτά γίνονται με HTTPS αιτήματα τύπου GET/POST/PUT/PATCH/DELETE.

Επίσης, το Backend επικοινωνεί με την βάση μας (MongoDB), μέσω DBMS statements τα οποία είτε παρέχονται έτοιμα από βιβλιοθήκη, είτε δημιουργήθηκαν από εμάς για πιο εξειδικευμένα queries. Έτσι λαμβάνει τα απαραίτητα δεδομένα που ζητούνται μέσω του REST API.

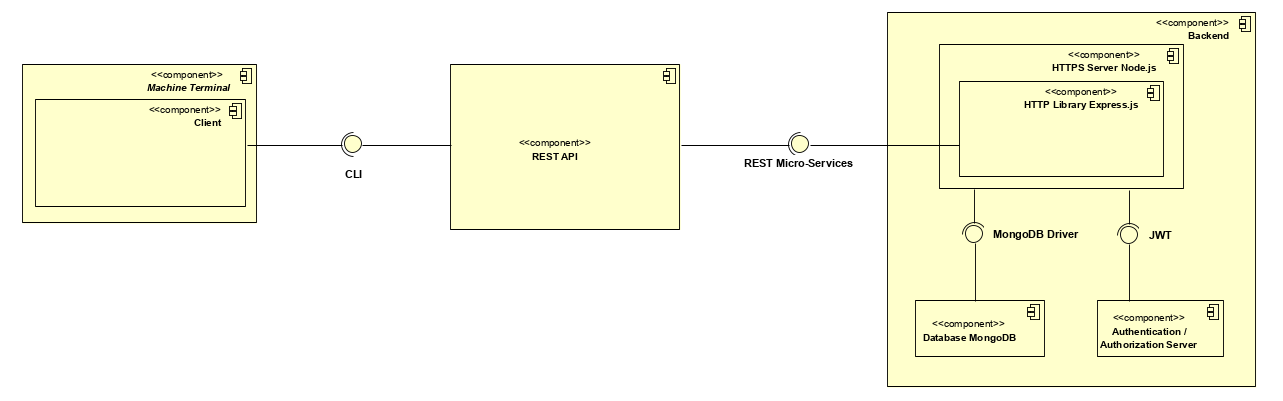


Figure 1: UML component diagram

**1.2.2 Διεπαφές με το χρήστη**

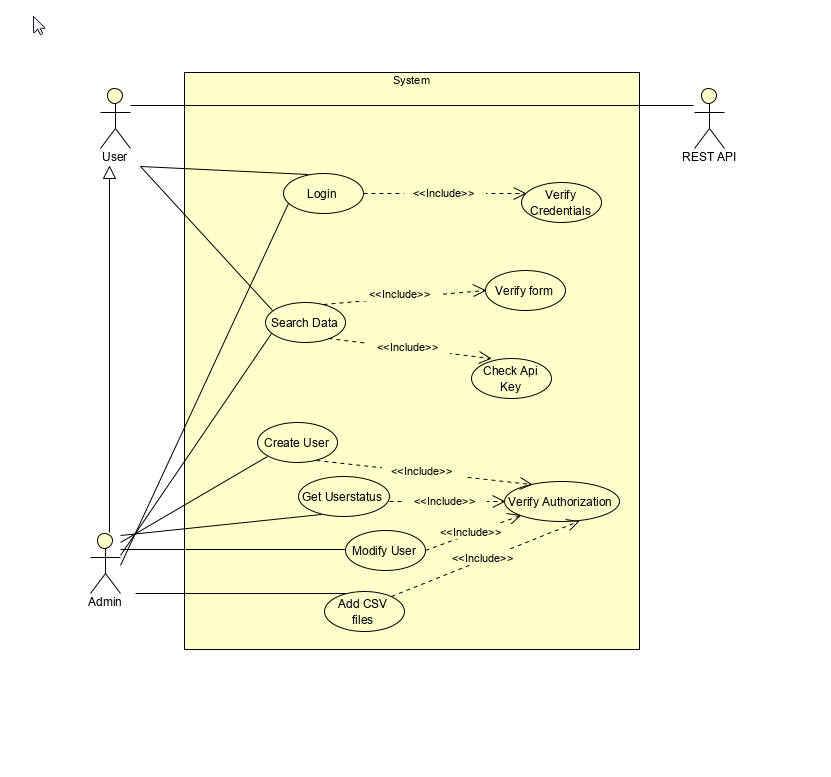
****

Figure 2: UML use case diagram

**2. Αναφορές - πηγές πληροφοριών**

• <https://transparency.entsoe.eu/>

• <https://www.visual-paradigm.com/>

• <https://www.visual-paradigm.com/features/visual-api-designer/>

• <https://www.mongodb.com/>

**3. Προδιαγραφές απαιτήσεων λογισμικού**

**3.1 Περιπτώσεις χρήσης**

**3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Log in του χρήστη και αναζήτηση δεδομένων**

**3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται**

Στη συγκεκριμένη περίπτωση χρήσης εμπλέκεται μόνο ο χρήστης που επιθυμεί να παρατηρήσει τα δεδομένα παραγωγής ηλεκτρικής ενέργειας στην χώρα ή την ζώνη της Ευρώπης που επιθυμεί.

**3.1.1.2 Προϋποθέσεις εκτέλεσης**

* Ο χρήστης πρέπει να είναι συνδεδεμένος στον λογαριασμό του.
* Αναζήτηση δεδομένων με σωστό API Key.
* Να υπάρχουν στην βάση τα δεδομένα που αναζητάμε(πχ σωστή χώρα ή ζώνη αγοράς)

**3.1.1.3 Περιβάλλον εκτέλεσης**

Η εν λόγω αναζήτηση γίνεται μέσω του terminal του υπολογιστή κάθε χρήστη όπου βάζει την κατάλληλη εντολή ανάλογα με την ενέργεια που θέλει να εκτελέσει.

**3.1.1.4 Δεδομένα εισόδου**

Δεδομένα εισόδου θεωρούνται τα στοιχεία που δίνει ο χρήστης προκειμένου να πραγματοποιηθεί το Login και το API Key που χρησιμοποιεί.

**3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά**

1. Επιτυχές login
2. Επιστροφή Public ApiKey στον χρήστη
3. Αναζήτηση με API Key
4. Προβολή αποτελεσμάτων

Συμπεριφορά σε απρόβλεπτες καταστάσεις: Σε περίπτωση εισαγωγής λάθος δεδομένων από τον χρήστη επιστρέφει error message.

**3.1.1.6 Δεδομένα εξόδου**

Έξοδος της αναζήτησης είναι το Public ApiKey και το σύνολο των δεδομένων που έχουμε αναζητήσει στην βάση.

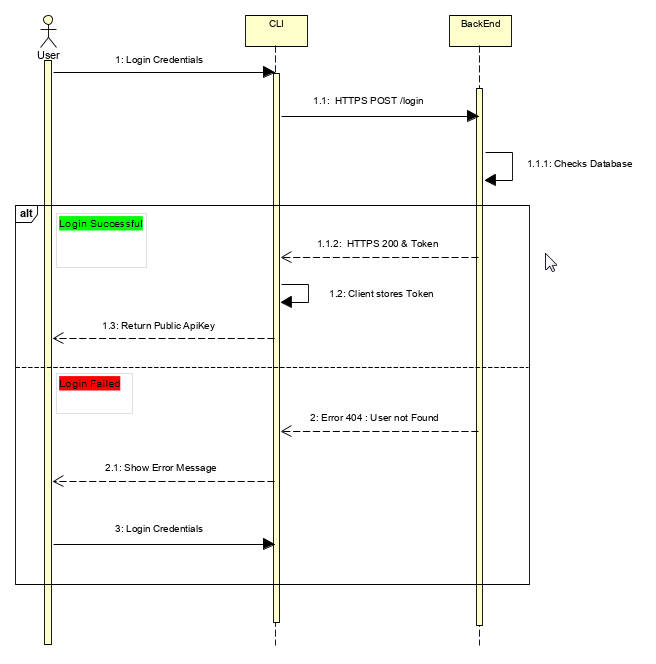


Figure 3: UML sequence diagram για login

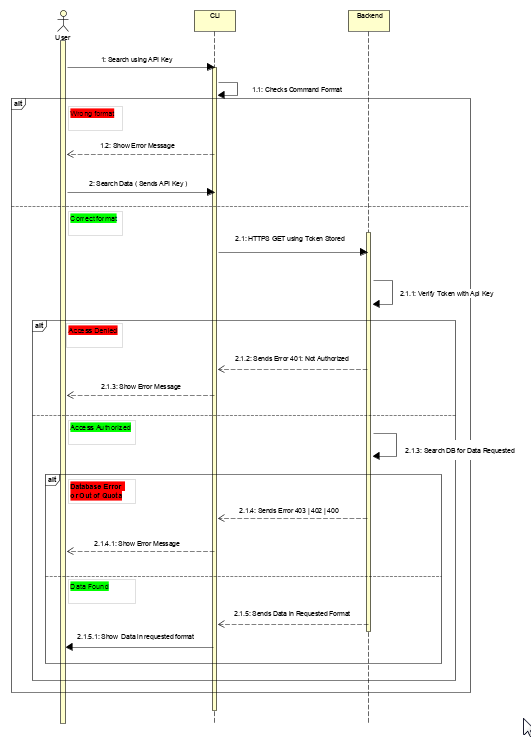


Figure 4: UML sequence diagram για search data

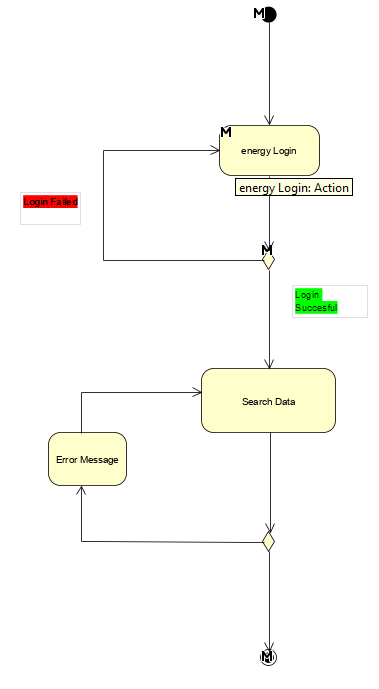


Figure 5: UML activity diagram για user search data

**3.1.1.7 Παρατηρήσεις**

Ν/Α

**3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Ο ADMIN κάνει login και add user**

**3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται**

Στη συγκεκριμένη περίπτωση εμπλέκεται ο ¨Διαχειριστής¨, ο οποίος αλληλεπιδρά με το ειδικό dashboard που εμφανίζεται λόγω της ιδιότητάς του.

**3.1.2.2 Προϋποθέσεις εκτέλεσης**

* Είσοδος του admin με τα διαπιστευτήριά του (username και password).
* Ο user να μην υπάρχει ήδη στην βάση

**3.1.2.3 Περιβάλλον εκτέλεσης**

Η εν λόγω αναζήτηση γίνεται μέσω του terminal του υπολογιστή του admin όπου βάζει energy την κατάλληλη εντολή ανάλογα με την ενέργεια που θέλει να εκτελέσει.

**3.1.2.4 Δεδομένα εισόδου**

Δεδομένα εισόδου θεωρούνται τα στοιχεία (login credentials) που δίνει ο admin προκειμένου να πραγματοποιηθεί το Login του και τα user credentials και quota που αυτός προσθέτει προκειμένου να προσθέσει έναν νέο user.

**3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά**

1. Επιτυχές login(με τα σωστά credentials για να έχει priviledges admin)
2. Επιστροφή Public ApiKey στον χρήστη
3. Προσθήκη credentials και quota του user που θέλει να κάνει add

Συμπεριφορά σε απρόβλεπτες καταστάσεις: Σε περίπτωση εισαγωγής λάθος δεδομένων από τον admin (είτε λάθος στο Login είτε ο χρήστης υπάρχει ηδη στην βάση) επιστρέφει error message.

**3.1.2.6 Δεδομένα εξόδου**

Έξοδος της αναζήτησης είναι το Public ApiKey του admin.

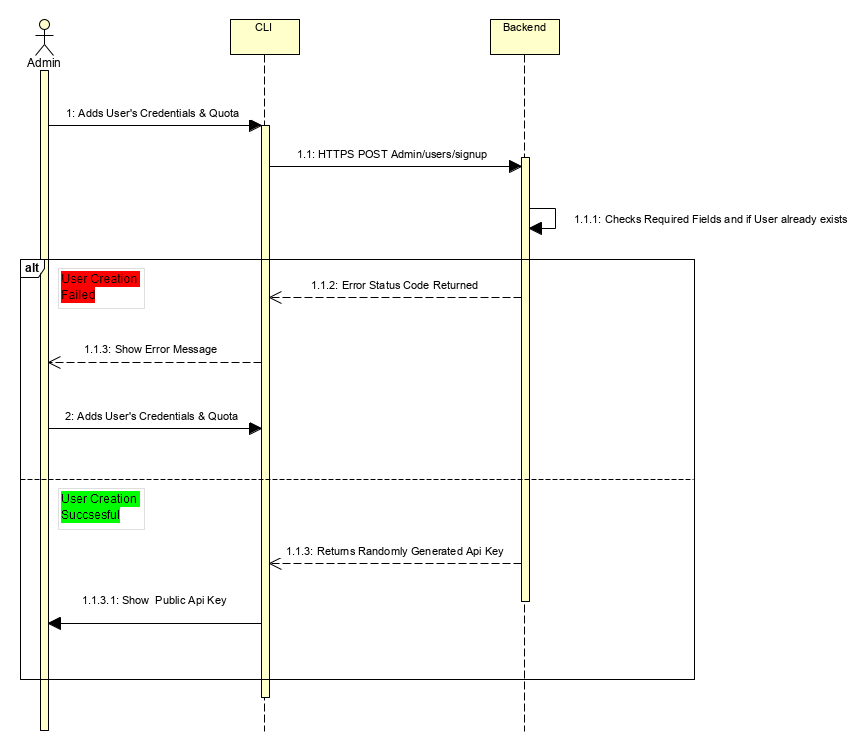


Figure 6: UML activity diagram για create user από admin

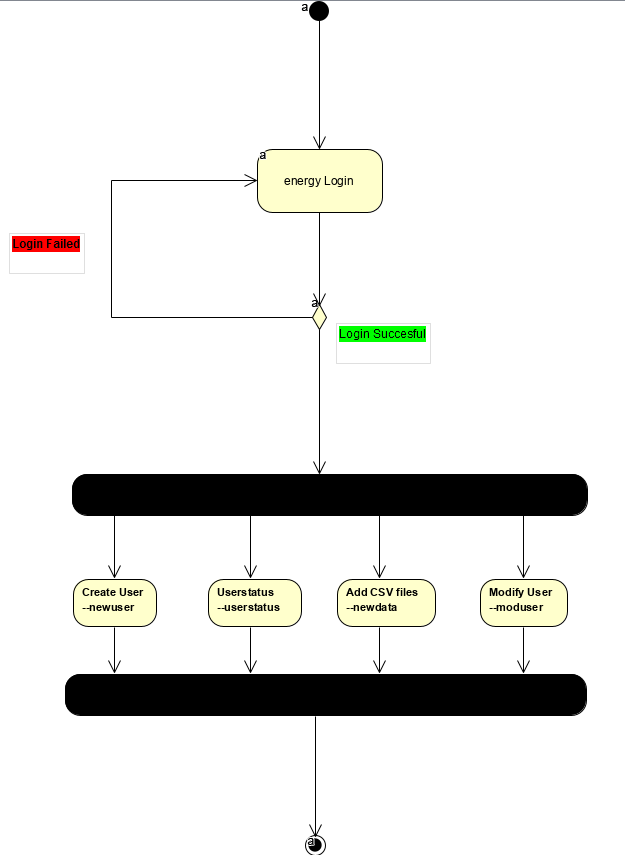


Figure 7: UML activity diagram για admin

**3.1.2.7 Παρατηρήσεις**

Ν/Α

**3.2 Απαιτήσεις επιδόσεων**

Τα δεδομένα εισόδου που εισάγει ο admin είναι μικρά σε όγκο. Ωστόσο απαιτείται η ταυτόχρονη εξυπηρέτηση μεγάλου πλήθους αιτήσεων. Μέτρο επίδοσης αποτελεί ο χρόνος απόκρισης της εφαρμογής σε αιτήματα, ο οποίος οφείλει να είναι τάξης μεγέθους μικρότερης του δευτερολέπτου με πλήθος αιτήσεων σε φυσιολογικά επίπεδα. Σε αυξημένη κίνηση μέτρο αποτελεί η διαθεσιμότητα, με κριτήριο επίδοσης το πλήθος αιτημάτων που εξυπηρετούνται. Σε αυτή την περίπτωση προτιμάται η εξασφάλιση της διαθεσιμότητας με τίμημα αύξησης του χρόνου απόκρισης.

Τα μέτρα του Back-end είναι τα εξής:

1. **Υψηλή ταχύτητα φόρτωσης δεδομένων:** υποδεικνύει την ταχύτητα με την οποία φορτώνονται και είναι διαθέσιμα τα δεδομένα.
2. **Latency:** υποδεικνύει τον χρόνο ολοκλήρωσης μίας λειτουργίας I/O.
3. **Υψηλό IOPS:** πρόκειται για το πλήθος λειτουργιών I/O που εκτελούνται ανά δευτερόλεπτο.
4. **Sequential Throughput:** πρόκειται για τον ρυθμό μεταφοράς δεδομένων. Μονάδα μέτρησης είναι τα (M/G)bytes/sec. Ισούται με IOPS\*TransferSize.
5. **HTTPS calls:** όσο λιγότερα τόσο χαμηλότερο το workload. Οφείλουμε να τυα χρησιμοποιούμε φειδωλά.

**3.3 Απαιτήσεις οργάνωσης δεδομένων**

**3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα**

Ανάλογα με την κατηγορία κάθε χρήστη υπάρχουν διαφορετικές απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα. Πιο συγκεκριμένα:

* Administrator: Ο administrator έχει τη δυνατότητα να παρακολουθεί και να διαμορφώνει όλα τα δεδομένα της βάσης δεδομένων.
* Χρήστης: Ο χρήστης έχει τη δυνατότητα να παρακολουθεί τα σύνολα των δεδομένων της αγοράς ηλεκτρικής ενέργειας στην Ευρώπη.

Μοντέλο κλάσεων UML:

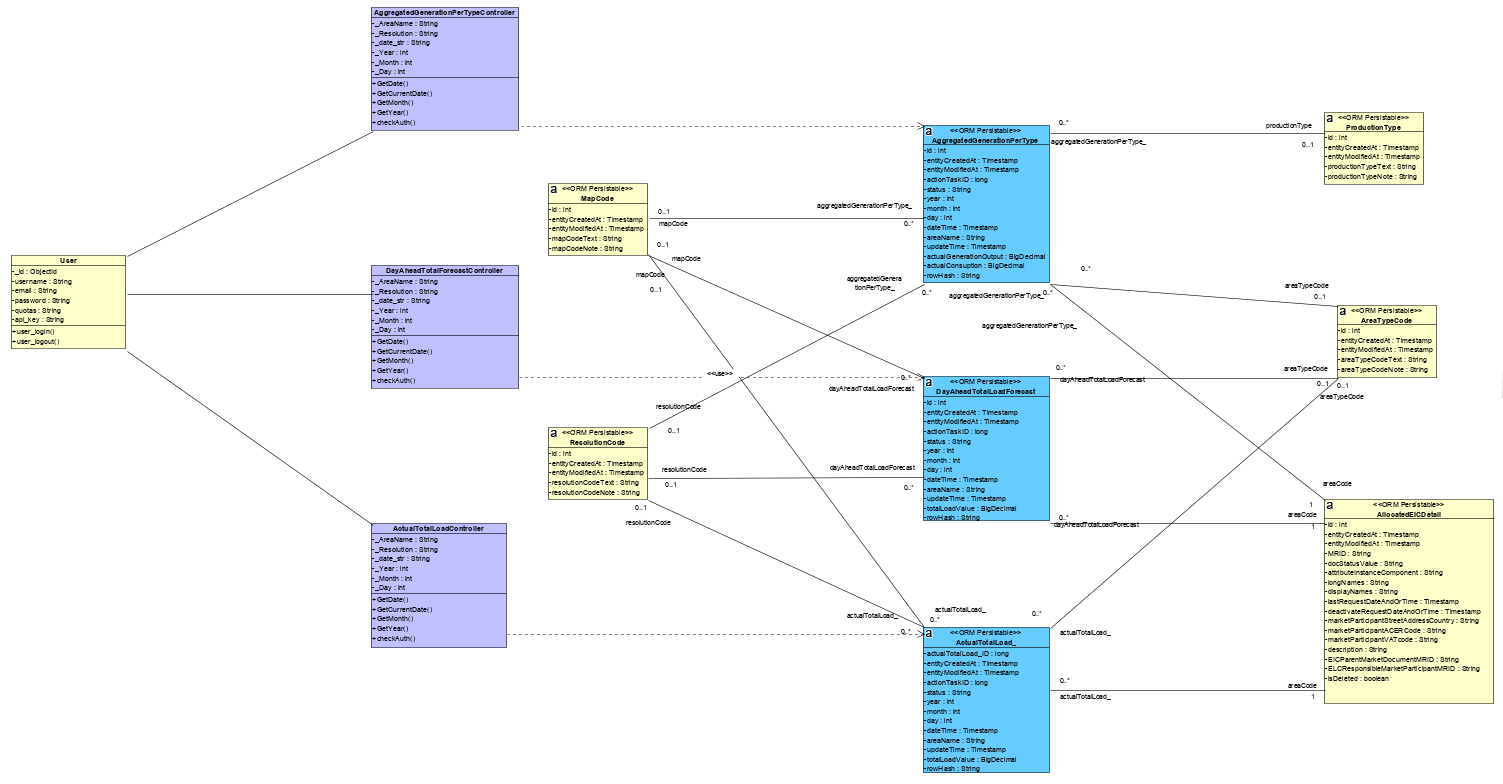


Figure 8: UML class diagram

**3.4 Περιορισμοί σχεδίασης**

Αρχικά αναφέρουμε τα σχεδιαστικά εργαλεία που χρησιμοποιούμε. Για την υλοποίηση του backend χρησιμοποιούμε την JavaScript με το NodeJS framework. Η επιλογή αυτού του framework έγινε γιατί είναι αρκετά διαδεδομένο για την ευελιξία που παρέχει στον προγραμματιστή. Για build automation χρησιμοποιούμε το npm. Το εργαλείο που χρησιμοποιείται για την επικοινωνία του backend με την βάση είναι το MongoDB. Η χρήση του εργαλείου MongoDB για την επικοινωνία του backend με τη βάση δεδομένων περιέχει τον περιορισμό των 512 MB storage, καθώς επιλέξαμε free plan για το εργαλείο αυτό. Μπορούμε βεβαια στο μέλλον να χρησιμοποιήσουμε αναβαθμισμένο plan του εργαλείου αυτού, προκειμένου να επεκταθεί το μέγεθος του storage.

**3.5 Λοιπές απαιτήσεις**

**3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού**

Απαιτούμε η διεπαφή με την βάση δεδομένων να είναι ενεργή καθ’όλη την διάρκεια της ημέρας, καθώς και σε βάθος χρόνου. Με αυτό το τρόπο θα εξασφαλίσουμε την καθολική διαθεσιμότητα της εφαρμογής μας, με σκοπό την καλύτερη εξυπηρέτηση των χρηστών.

**3.5.2 Απαιτήσεις ασφάλειας**

Παροχή κρυπτογραφημένης ανταλλαγής δεδομένων μεταξύ χρήστη-εξυπηρετητή μέσω https. Σε ό,τι αφορά τους λογαριασμούς βασική απαίτηση αποτελεί η κρυπτογράφηση των passwords των χρηστών και η αποθήκευση της αντίστοιχης hash τιμής τους μετά από αλλεπάλληλες κρυπτογραφήσεις στη βάση.

**3.5.3 Απαιτήσεις συντήρησης**

Θα εφαρμόζονται συχνά εργασίες συντήρησης στους κύριους διακομιστές της εφαρμογής μας, κατά τη διάρκεια των οποίων θα χρησιμοποιούνται backup διακομιστές, ώστε όντως να εξασφαλίζουμε 24/7 εξυπηρέτηση χρηστών. Παράλληλα, θα γίνονται συχνές αναβαθμίσεις τόσο στο λογισμικό όσο και στο σύστημα της βάσης δεδομένων μας, προκειμένου να εξασφαλίσουμε συμβατότητα με τις τελευταίες εκδόσεις και πρότυπα που επικρατούν στην αγορά.

**4. Παράρτημα**

**4.1 Παραδοχές και εξαρτήσεις**

N/A

**4.2 Ακρωνύμια και συντομογραφίες**

* UML: Unified Modeling Language
* IOPS: Input/Output Operations per Second

**4.3 Υποστηρικτικά έγγραφα, πρότυπα κ.λπ.**

N/A