

Έγγραφο απαιτήσεων λογισμικού (SRS)

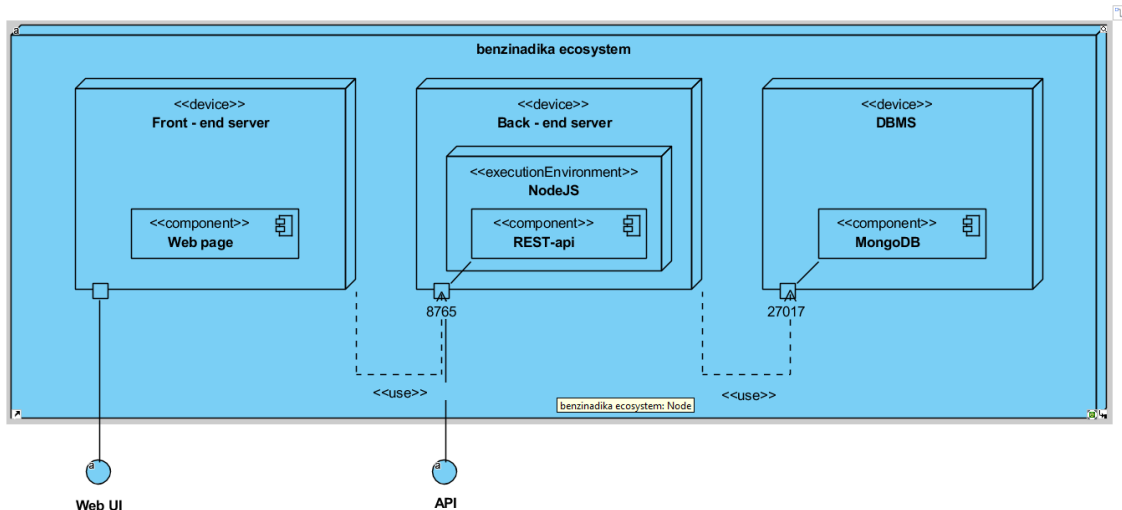
benzinadika app

1. Εισαγωγή

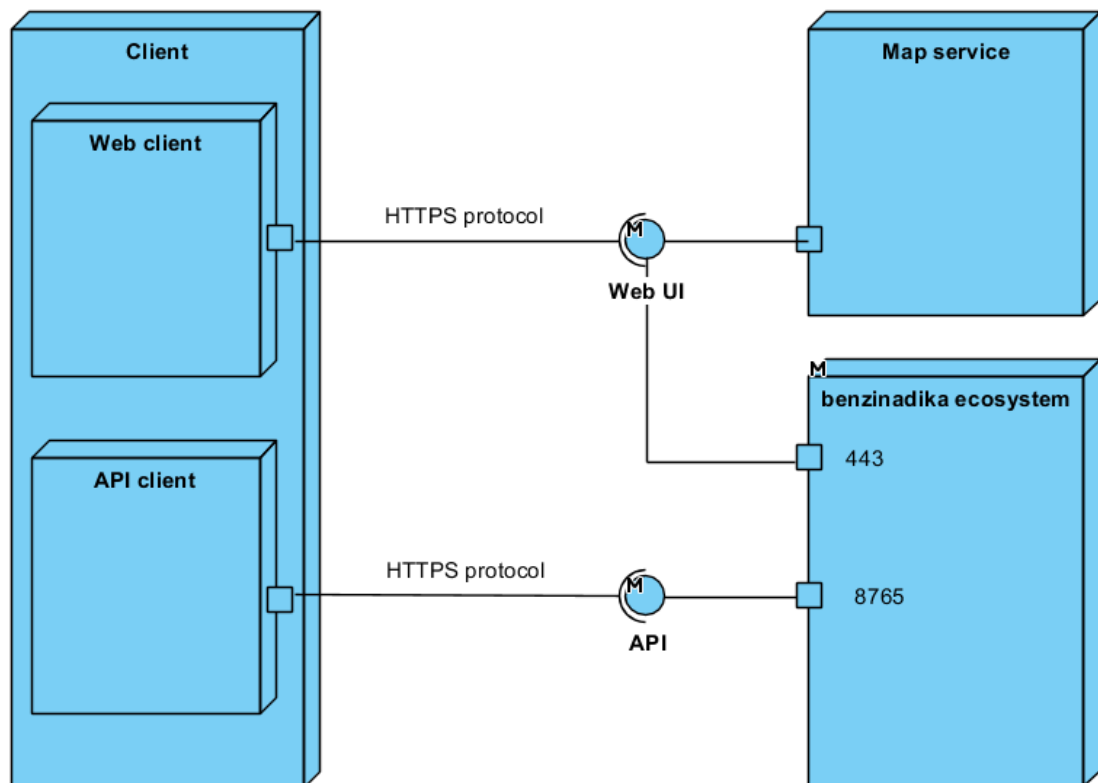
1.1. Εισαγωγή: σκοπός του λογισμικού

Σκοπός του παρόντος έργου λογισμικού “benzinadika”, είναι η υλοποίηση ενός διαδικτυακού παρατηρητηρίου τιμών για πρατήρια καυσίμων, όπου οι χρήστες θα είναι σε θέση να προσθέσουν πρατήρια και τιμές για κάθε είδος καύσιμου, να βλέπουν και να συγκρίνουν τις τιμές καθώς και να πλοηγούνται στα πρατήρια. Στόχος είναι η απλή και χρήσιμη ενημέρωση των χρηστών για τον οικονομικό ανεφοδιασμό των οχημάτων τους.

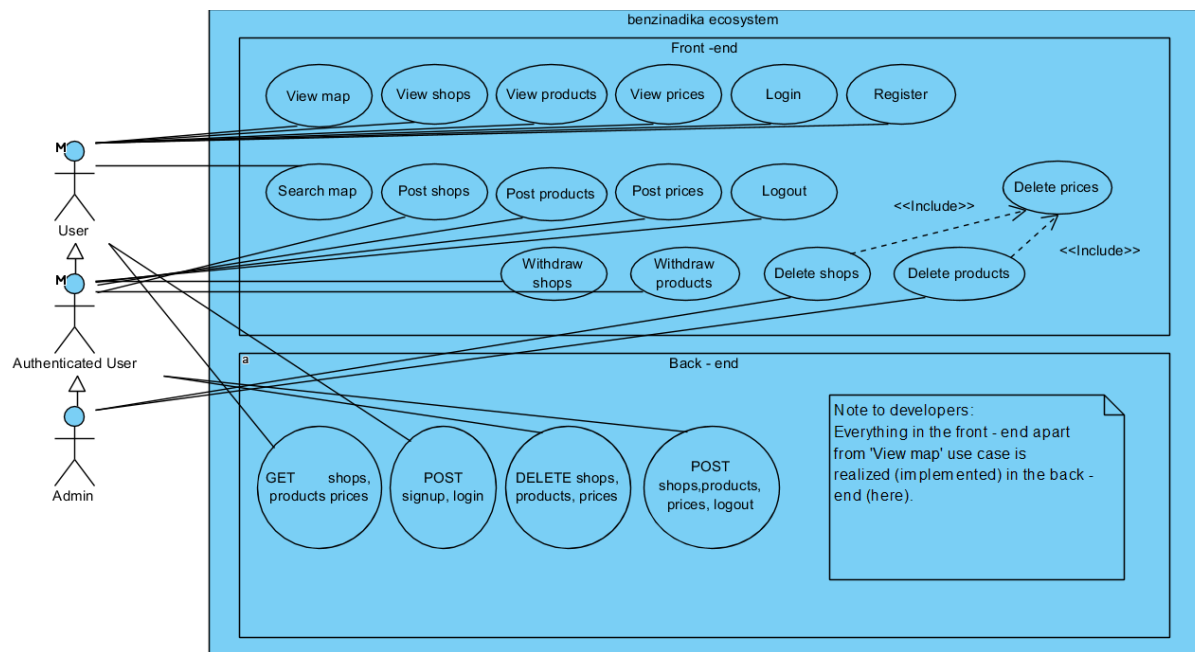
1.2. Επισκόπηση του λογισμικού



1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού



1.3.2 Διεπαφές με το χρήστη



1.3.3 Διεπαφές με υλικό

N/A

1.3.4 Διεπαφές επικοινωνιών

N/A

2. Αναφορές - πηγές πληροφοριών

N/A

3. Προδιαγραφές απαιτήσεων λογισμικού

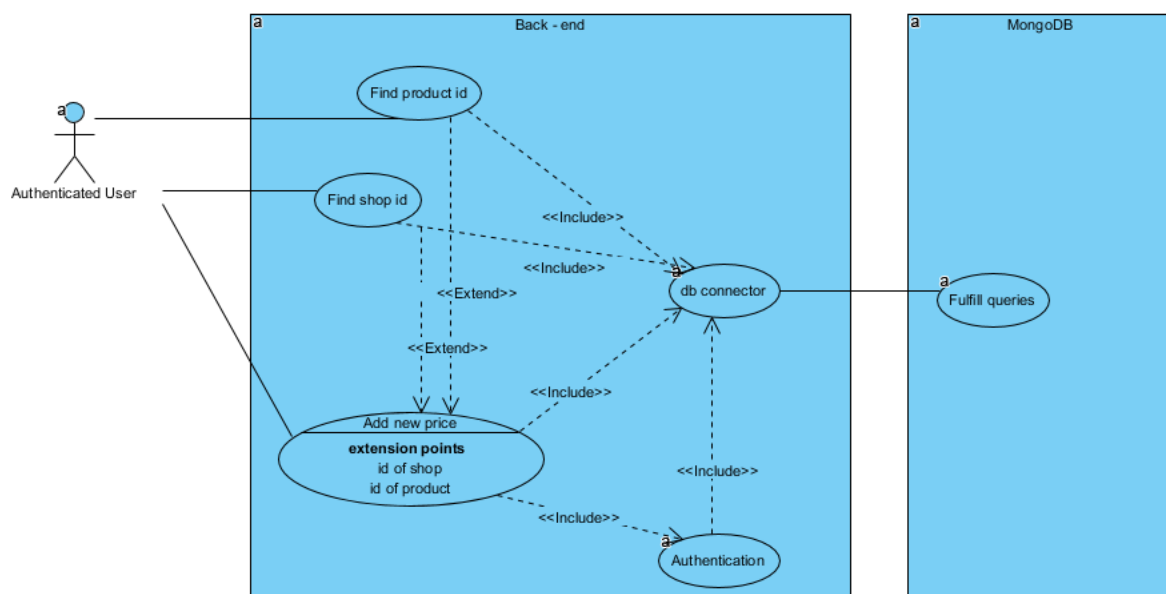
3.1 Εξωτερικές διεπαφές

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

3.2 Λειτουργίες: περιπτώσεις χρήσης

Η πρώτη περίπτωση χρήσης είναι η προσθήκη νέας τιμής προϊόντος από κάποιον εγγεγραμμένο χρήστη ή η διόρθωση μίας υπάρχουσας. Η δεύτερη περίπτωση θα είναι η εγγραφή, η σύνδεση και η αποσύνδεση κάποιου χρήστη στην πλατφόρμα.

3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Προσθήκη/Ενημέρωση Τιμής Προϊόντος



3.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι χρήστες που εμπλέκονται στη συγκεκριμένη περίπτωση χρήσης είναι οι «Εγγεγραμμένοι Χρήστες» (Οι εθελοντές δηλαδή) χάρις στους οποίους θα ανανεώνονται συνεχώς οι τιμές των πρατηρίων καθώς και οι Developers που έχουν την δυνατότητα τροποποίησης ή διαγραφής, οποιασδήποτε τιμής επιθυμούν.

3.2.1.2 Προϋποθέσεις εκτέλεσης

Προκειμένου ένας χρήστης να προσθέσει ή να τροποποιήσει μία τιμή, θα πρέπει να είναι συνδεδεμένος με τον λογαριασμό του είτε ως χρήστης εθελοντής είτε ως Developer. Έτσι κατά την προσθήκη μίας τιμής, το σύστημα θα προτρέψει τον χρήστη αν δεν είναι συνδεδεμένος, να εισάγει το Username και το Password του, προκειμένου να αποκτήσει επαρκή δικαιώματα για την εκτέλεση των λειτουργιών αυτών.

3.2.1.3 Περιβάλλον εκτέλεσης

Ο χρήστης θα μπορεί να προσθέσει ή να τροποποιήσει κάποια τιμή μέσω του User Interface. Κατά την επιβεβαίωση της καταχώρησης το Api θα αποστείλει τις ανανεωμένες τιμές στο Dbms και εκείνο με τη σειρά του θα αποθηκεύσει τις αλλαγές στη βάση, ώστε να μπορούν να είναι προσβάσιμες και από τους υπόλοιπους χρήστες.

3.2.1.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου που θα σταλθούν προς το backend της εφαρμογής είναι ένα Json αρχείο, που θα περιέχει απλώς τις τροποποιημένες/νέες τιμές των πρατηρίων. Ενώ, θα μας επιστρέφεται πίσω πάλι ένα Json που θα μας ενημερώνει για την επιτυχημένη καταχώρηση τιμών.

3.2.1.5 Παράμετροι

Προκειμένου τα δεδομένα που στέλνονται να είναι έγκυρα, θα πρέπει οι τύποι των μεταβλητών να είναι αυστηρά καθορισμένοι χωρίς περιθώριο λαθών. Για παράδειγμα, η τιμή των προϊόντων να είναι float, τα product id και shop id να υπάρχουν ήδη στο database και να είναι έγκυρα κ.ο.κ

3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Βήμα 1: Ο εγγεγραμμένος χρήστης αναζητά κάποιο υπάρχον πρατήριο.

Βήμα 2: Εφόσον το backend, μας επιστρέψει το πρατήριο που αναζητήσαμε, επιλέγουμε το είδος του προϊόντος που θέλουμε να προσθέσουμε ή να τροποποιήσουμε.

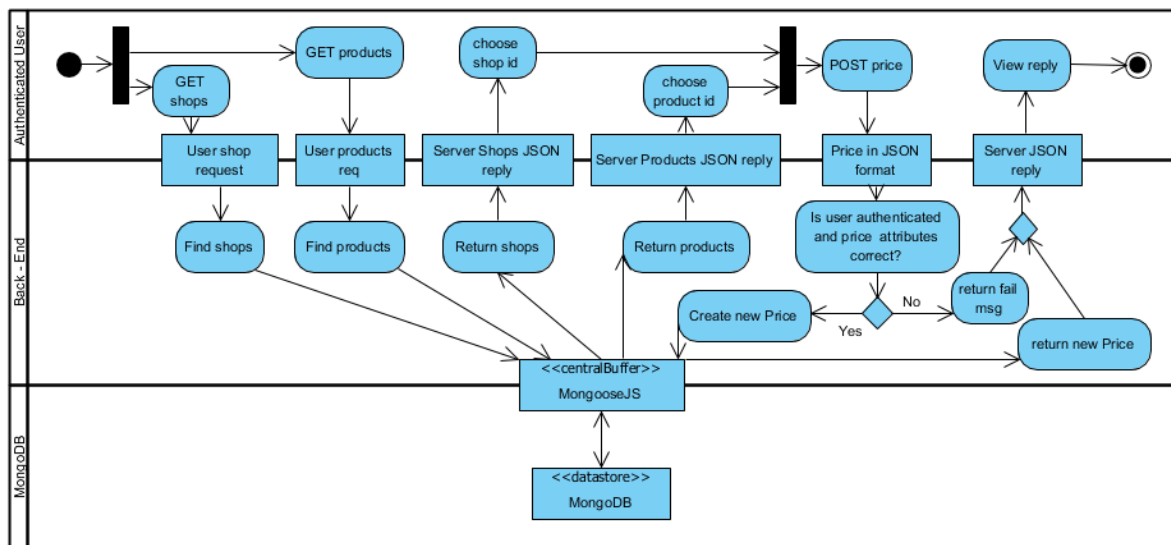
Βήμα 3: Καταγράφουμε την νέα τιμή του προϊόντος που έχει επιλεγεί και την καταχωρούμε μέσω Post στο backend.

Βήμα 4: Το backend, πρώτου καταχωρήσει τη νέα τιμή στη Database, στέλνει query προς την db, προκειμένου να διαπιστώσει αν ο χρήστης είναι authenticated ή όχι. Εάν ο χρήστης δεν είναι συνδεδεμένος, το backend επιστρέφει σφάλμα καθώς ο χρήστης δεν έχει δικαιώματα εκτέλεσης της ενέργειας αυτής.

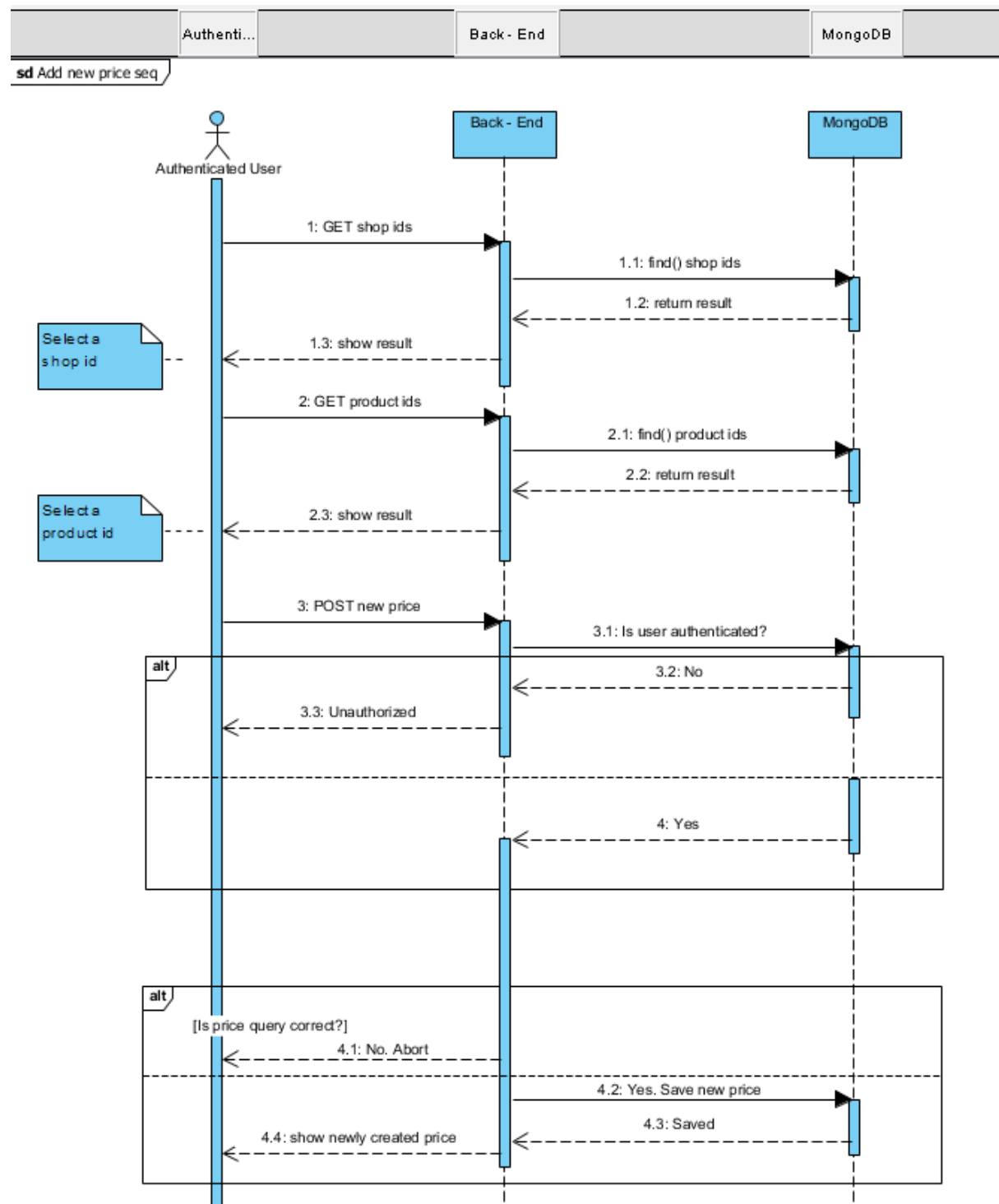
Βήμα 5: Αν ο χρήστης είναι συνδεδεμένος, το backend ξαναστέλνει query προς τη db, ώστε να βεβαιωθεί πως το price query που αποστείλαμε αρχικά είναι έγκυρο και περιέχει σωστούς τύπους δεδομένων. Αν όχι, επιστρέφει σφάλμα.

Βήμα 6: Αν το price query είναι σωστό, καταχωρείται η νέα τιμή στο database και μετά την αποθήκευσή της, μας επιστρέφεται η νέα τιμή από το backend, η οποία πλέον είναι εμφανής και στο frontend.

Uml Activity Diagram



Uml Sequence Diagram



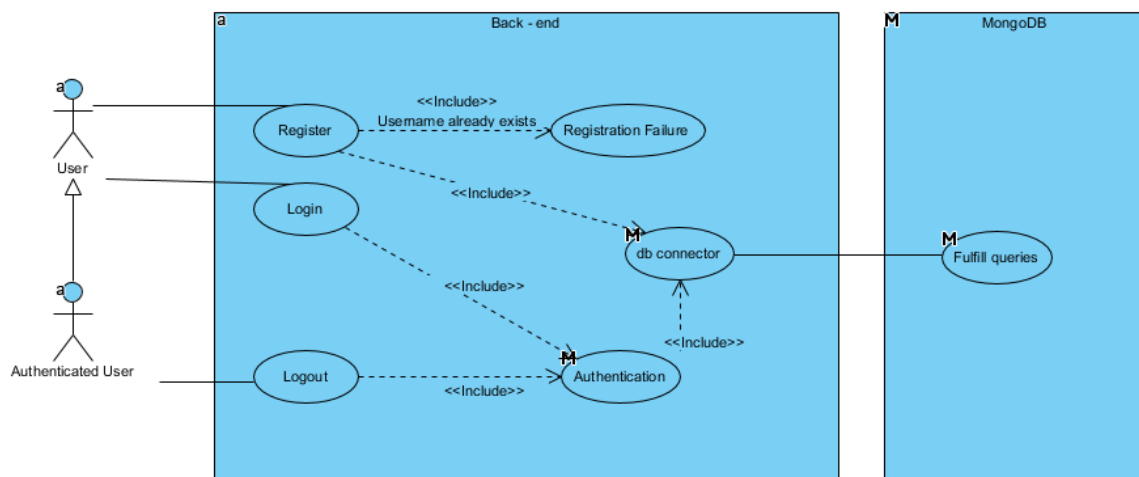
3.2.1.7 Δεδομένα εξόδου

Τα δεδομένα εξόδου παρουσιάστηκαν όπως αυτά ζητούνται στα Uml διαγράμματα της ενότητας 3.2.1.6.

3.2.1.8 Παρατηρήσεις

N/A

3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Εγγραφή, Σύνδεση και Αποσύνδεση Χρηστών



3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Η περίπτωση χρήσης αυτή αφορά όλες τις κατηγορίες χρηστών. Οι απλοί χρήστες θα έχουν την δυνατότητα εγγραφής στην πλατφόρμα μας, ενώ οι ήδη εγγεγραμμένοι θα μπορούν να προβούν σε ενέργειες Σύνδεσης ή Αποσύνδεσης

3.2.2.2 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. Για την περίπτωση που κάποιος χρήστης θέλει απλώς να εγγραφεί, η μόνη προϋπόθεση που υπάρχει είναι να μην υπάρχει ήδη εγγεγραμμένος χρήστης με το ίδιο Username.

3.2.2.3 Περιβάλλον εκτέλεσης

Όπως και στην πρώτη περίπτωση χρήσης, το περιβάλλον εκτέλεσης είναι αρχικά το UI (frontend) μέσω του οποίου ένας χρήστης μπορεί είτε να εγγραφεί είτε 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 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 μας εμφανίσει σφάλμα, καθώς το Username χρησιμοποιείται ήδη.

3.2.2.4 Δεδομένα εισόδου

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

3.2.2.5 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 ελέγχεται αν έχει τη σωστή μορφή char variable, ενώ το password στέλνεται hashed για λόγους ασφάλειας και ελέγχεται κατά την πληκτρολόγηση αν έχει έγκυρη μορφή (πάνω από 8 χαρακτήρες, χρήση γραμμάτων και αριθμών κλπ.)

3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Για την εγγραφή:

Βήμα 1: Ο χρήστης πληκτρολογεί το Username και το Password που επιθυμεί και το καταχωρεί.

Βήμα 2: Το backend στέλνει Post query προς το dbms, με το Username και Password.

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 μας επιστρέφει μήνυμα για την επιτυχημένη εγγραφή του χρήστη.

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:

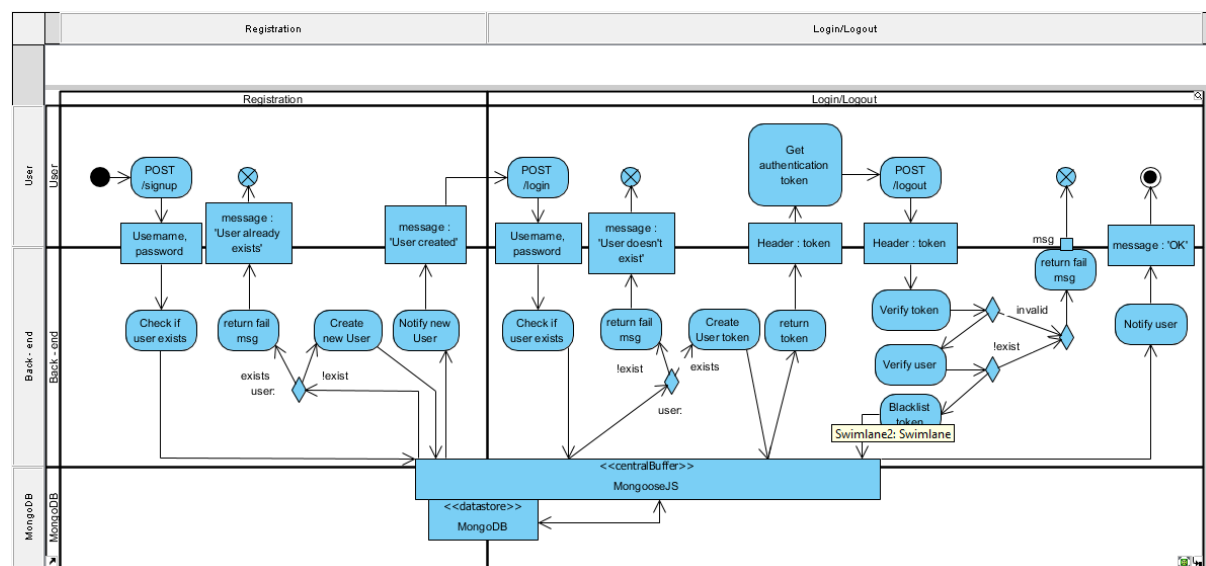
Βήμα 1: Ο χρήστης επιλέγει την αποσύνδεση από την πλατφόρμα μέσω του αντίστοιχου συνδέσμου στο frontend.

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

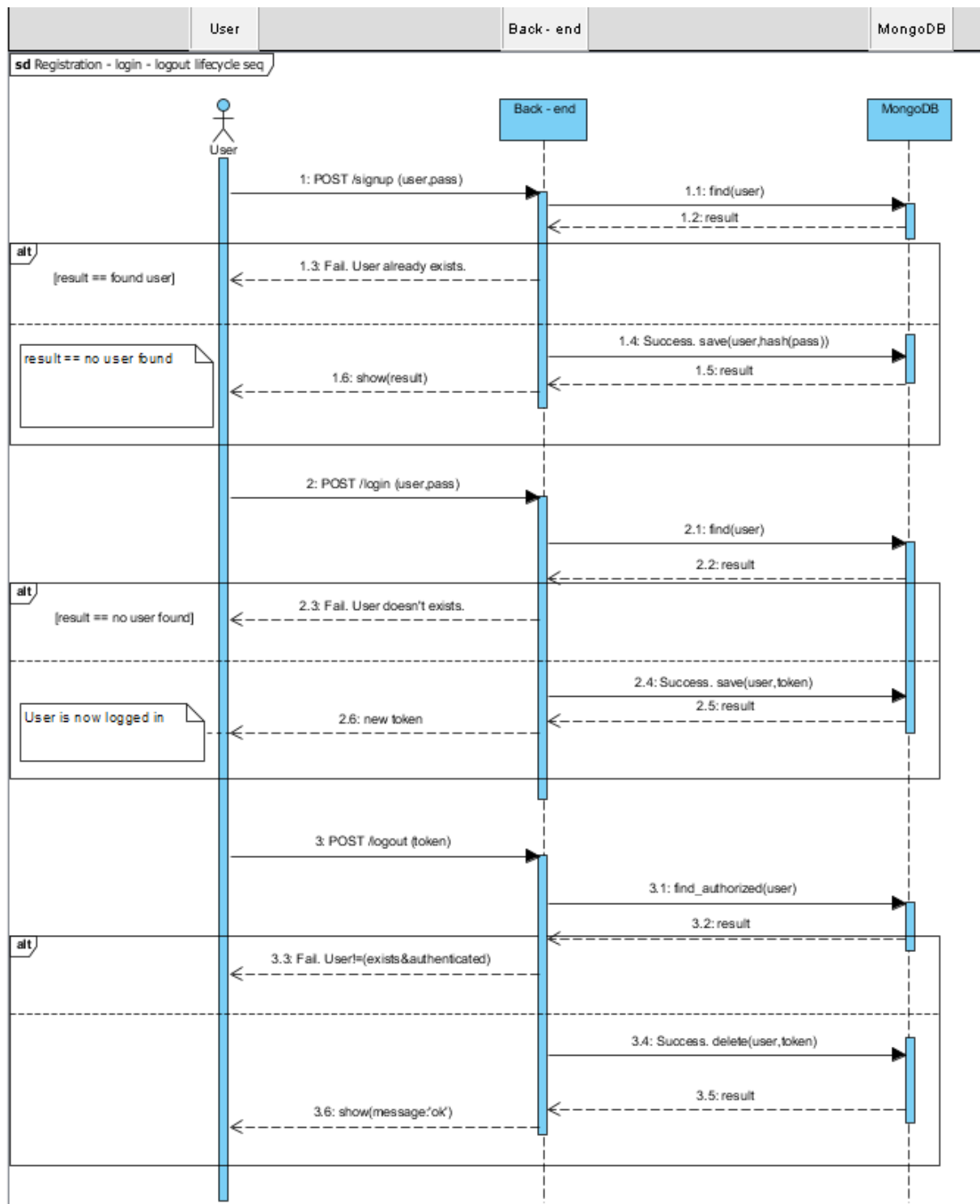
Βήμα 3: Η dbms αναζητά το συνδυασμό (Username, Token). Αν δεν τον βρει, το backend επιστρέφει σφάλμα πως ο χρήστης δεν βρέθηκε ή δεν είναι εξουσιοδοτημένος (έληξε το token).

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



3.2.2.7 Δεδομένα εξόδου

Τα δεδομένα εξόδου παρουσιάστηκαν όπως αυτά ζητούνται στα Uml διαγράμματα της ενότητας 3.2.2.6.

3.2.2.8 Παρατηρήσεις

N/A

3.3 Απαιτήσεις επιδόσεων

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

- Την ταχύτητα αποκρισιμότητας: 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 μεταφερισιμότητα των δεδομένων (αναλύεται στη συνέχεια).

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 βασικά δεδομένα που αποθηκεύουμε στην βάση δεδομένων μας, αφορούν τα καταστήματα (πρατήρια καυσίμων), τα προϊόντα (καύσιμα), τις τιμές των προϊόντων, καθώς και τους χρήστες οι οποίοι επισκέπτονται την εφαρμογή.

Για τα καταστήματα, αποθηκεύεται το shopID ως primary key, το όνομα, η διεύθυνση, καθώς και δύο επιπλέον μετρικές, το γεωγραφικό μήκος και πλάτος στο οποίο βρίσκεται το πρατήριο, με σκοπό να μπορεί να εντοπιστεί από τον χρήστη μέσω της ενσωμάτωσης των geolocation services της google.

Για τα προϊόντα αποθηκεύεται το productID ως primary key, το όνομα του προϊόντος, η κατηγορία στην οποία ανήκει (π.χ. βενζίνη, πετρέλαιο κλπ), μια περιγραφή του προϊόντος και άλλα χρήσιμα χαρακτηριστικά.

Για τις τιμές των προϊόντων αποθηκεύεται το id του προϊόντος το οποίο αφορά η τιμή, καθώς και το id του πρατηρίου το οποίο διαθέτει το συγκεκριμένο προϊόν. Επιπλέον καταγράφεται η ημερομηνία της καταχώρησης της εκάστοτε τιμής.

Για τους users αποθηκεύεται, το userID ως primary key, το username και το password τους, εφόσον είναι εγγεγραμμένοι.

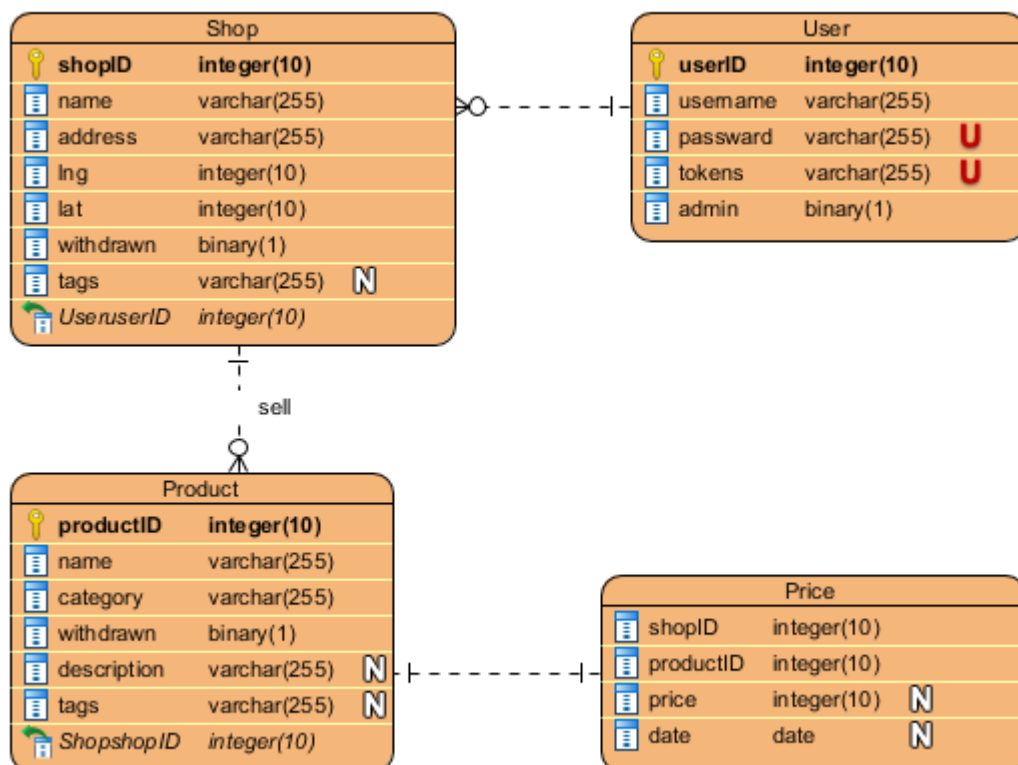
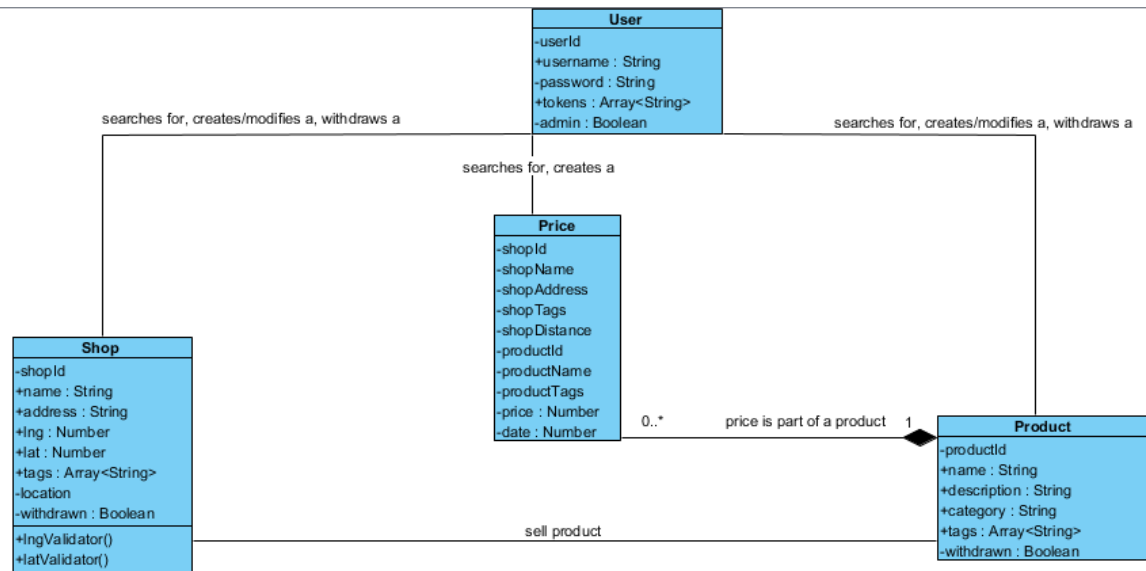
3.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Για να έχει ένας χρήστης δικαίωμα ενημέρωσης προϊόντων και των τιμών τους στην εφαρμογή, θα πρέπει πρώτα να έχει εγγραφεί, δηλαδή να έχει κάνει register ως εθελοντής, ώστε να είναι καταχωρημένος στη βάση δεδομένων και να του εκχωρηθούν τα αντίστοιχα δικαιώματα post, patch. Εφόσον μετά το register κάνει επιτυχές login (εισάγοντας σωστό username και password), αποκτά πρόσβαση στην πλατφόρμα με δικαιώματα volunteer.

Ένας χρήστης που δεν είναι admin αλλά ούτε και εθελοντής, δεν χρειάζεται να κάνει login, αλλά διαθέτει περιορισμένα δικαιώματα (μόνο τύπου get) στα δεδομένα.

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

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



3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Τα προσωπικά δεδομένα των χρηστών διαφυλάσσονται αποτελεσματικά, καθώς τα password κρυπτογραφούνται με σχετικά hash functions. Τέλος, σε κάθε session αποδίδεται στον εκάστοτε συνδεδεμένο χρήστη (admin ή volunteer) μοναδικά jwt tokens. Έτσι, πιστοποιείται η εγκυρότητα και η ακεραιότητα των δεδομένων, αφού αυτά προέρχονται από αποκλειστικά πιστοποιημένους χρήστες.

3.4.5 Προδιαγραφές διατήρησης δεδομένων

Όλα τα απαραίτητα για την εφαρμογή δεδομένα αποθηκεύονται σε μια MongoDB βάση δεδομένων, υπό την μορφή json αρχείων (NoSQL). Το σύστημα αυτό εγγυάται τη δυνατότητα επεκτασιμότητας (scalability) της εφαρμογής μας, τη μόνιμη αποθήκευση των δεδομένων, καθώς και την γρήγορη ανάκτησή τους, όποτε αυτή είναι απαραίτητη.

3.5 Περιορισμοί σχεδίασης

- Το έργο οφείλει να έχει ολοκληρωθεί το αργότερο έως τα τέλη Φεβρουαρίου 2019.
- Το έργο στην πιο εύχρηστη εκδοχή του, δηλαδή την εισαγωγή τιμών καυσίμων κατά την στιγμή ανεφοδιασμού, θα απαιτεί πρόσβαση στο διαδίκτυο είτε μέσω παρεχόμενου wifi από το πρατήριο είτε πιθανότερα μέσω των δεδομένων κινητής του ιδίου του χρήστη.
- Προκειμένου ένας χρήστης να προσθέσει προϊόντα και τιμές θα πρέπει πρώτα να έχει εγγραφεί ως εθελοντής και να έχει κάνει επιτυχές login, αλλιώς δεν του παρέχονται τα δικαιώματα για post και patch.
- Τα δεδομένα που εισάγονται από τους volunteers μέσω του UI πρέπει να τηρούν τον τύπο, καθώς και κάθε περιορισμό που έχει τεθεί για τη μορφή τους στα αντίστοιχα μοντέλα. Για παράδειγμα, το password που διαλέγει ο χρήστης κατά την εγγραφή του, πρέπει να αποτελείται από τουλάχιστον 5 στοιχεία.

3.6 Λοιπές απαιτήσεις

3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

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

3.6.2 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. Τέλος, σε κάθε session αποδίδεται στον εκάστοτε συνδεδεμένο χρήστη (admin ή volunteer) μοναδικά jwt tokens. Έτσι, πιστοποιείται η εγκυρότητα και η γνησιότητα των δεδομένων, αφού αυτά προέρχονται από αποκλειστικά πιστοποιημένους χρήστες.

3.6.3 Απαιτήσεις συντήρησης

Όπως κάθε ολοκληρωμένο και ανταγωνιστικό έργο λογισμικού, τα benzinadika θα ενημερώνονται 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.

4. Παράρτημα

4.1 Παραδοχές και εξαρτήσεις

N/A

4.2 Ακρωνύμια και συντομογραφίες

N/A

4.3 Υποστηρικτικά έγγραφα, πρότυπα κ.λπ.

N/A