Έγγραφο απαιτήσεων λογισμικού (SRS)

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

***[Care.Less]***

# Εισαγωγή

### **1.1 Εισαγωγή: σκοπός του λογισμικού**

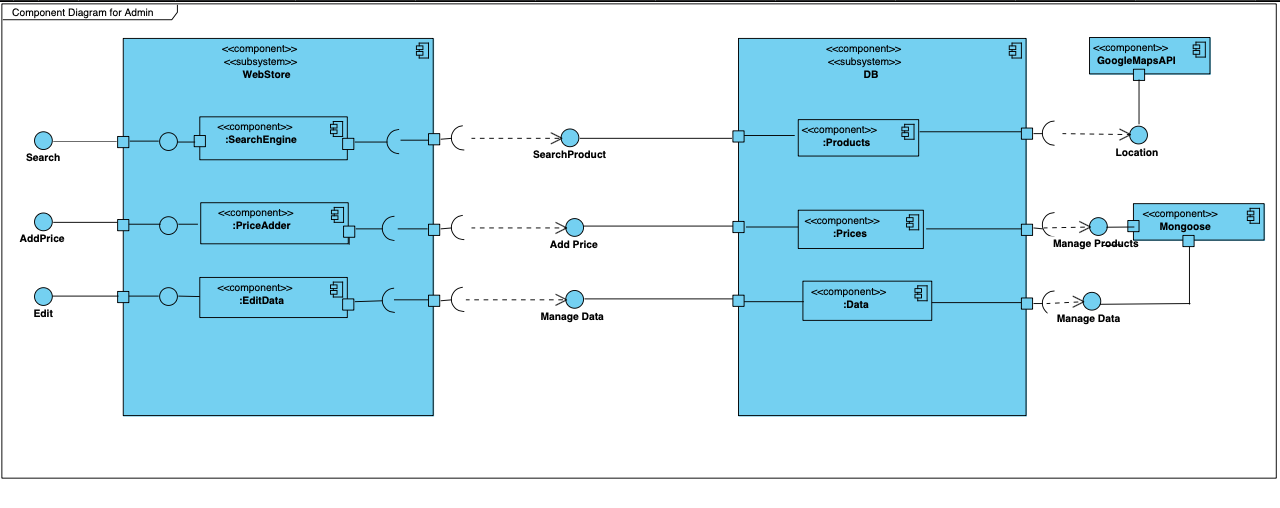
Οριοθέτηση του σκοπού του συστήματος

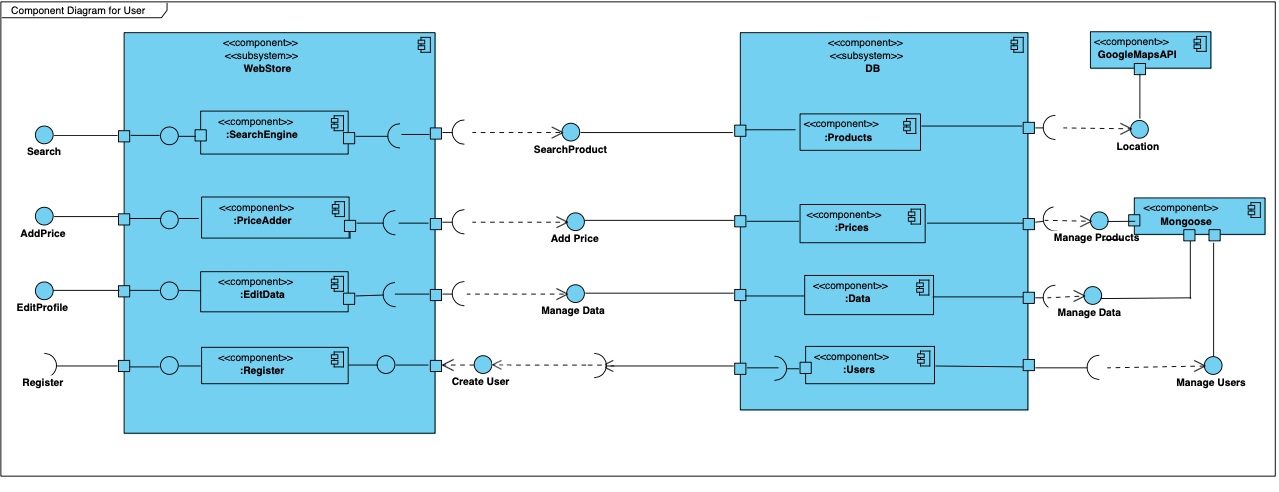
**Η οικονομική κρίση που έχει ξεσπάσει τα τελευταία χρόνια στη χώρα μας, έχει καταστήσει το ψάξιμο για την αγορά κάποιων προϊόντων, αδήριτη ανάγκη. Στόχος του συστήματος είναι να παρέχει στους χρήστες του την ευκαιρία να δουν τις τιμές κάποιων ειδών προσωπικής φροντίδας που τους ενδιαφέρουν από διάφορα καταστήματα, ανανεωμένες με τις πιο πρόσφατες προσφορές ή εκπτώσεις, δίνοντας τους τη δυνατότητα να εξοικονομήσουν χρήματα χωρίς κόπο, επιλέγοντας οι ίδιοι το κατάστημα που τους βολεύει στην καλύτερη τιμή. Το Care.less έρχεται για να δώσει στους χρήστες του μια εύχρηστη web εφαρμογή που θα τους δίνει την ευκαιρία να κάνουν την έρευνα αγοράς τους εύκολα και γρήγορα από το σπίτι.**

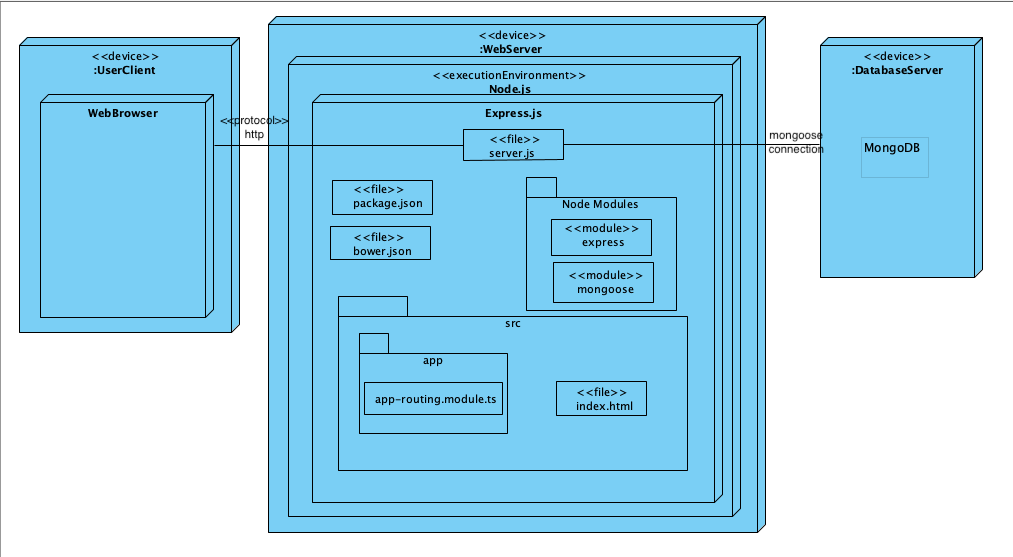
### **1.2 Επισκόπηση του λογισμικού**

Γενική περιγραφή με χρήση διαγράμματος UML . Οι διεπαφές αναφέρονται μόνο ως τίτλοι ή/και σε διάγραμμα.









### **1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού**

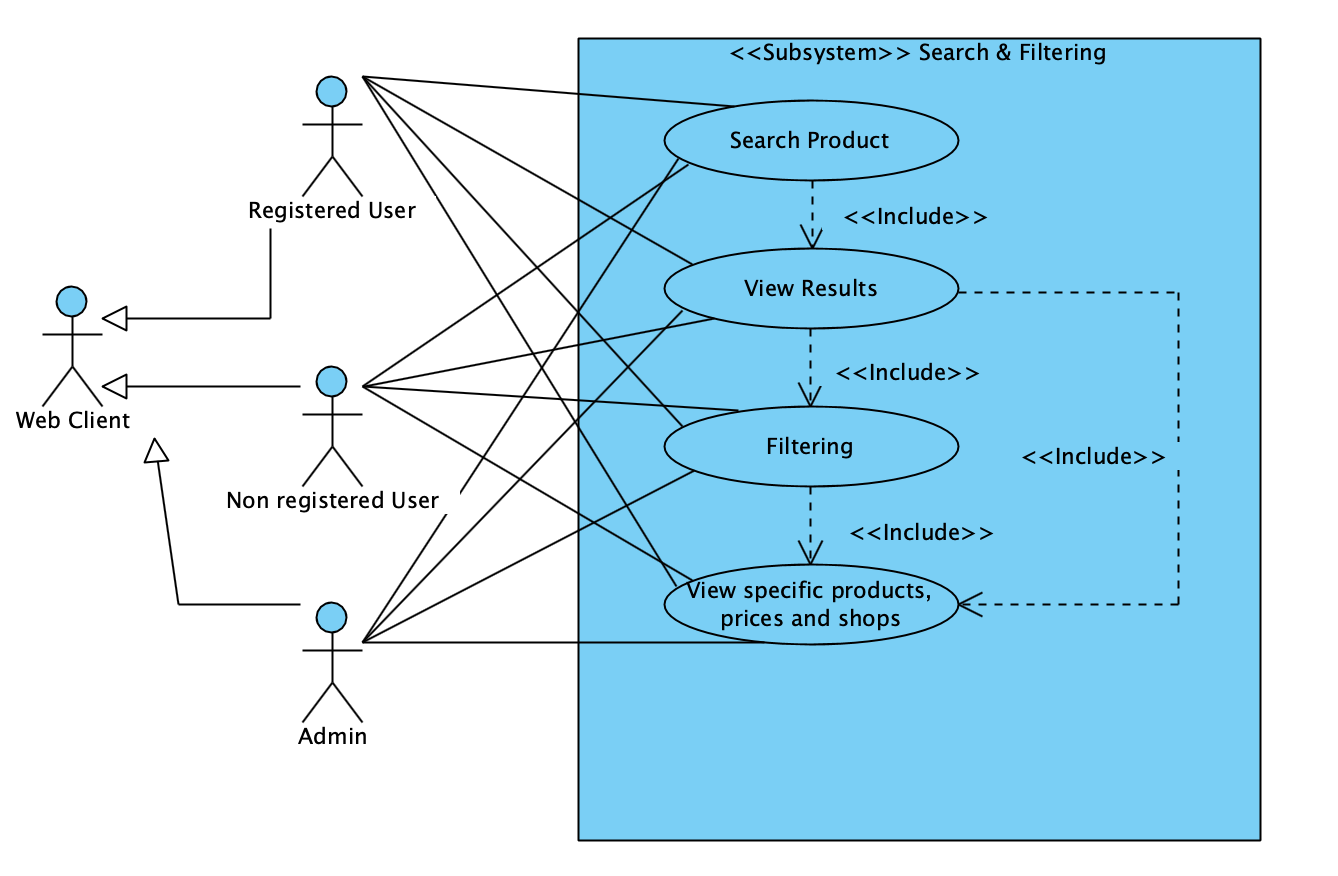
Προδιαγραφή διεπαφών με εξωτερικά συστήματα και λογισμικό, με αναφορά σε πρότυπα ανταλλαγής δεδομένων και κλήσης υπηρεσιών. Χρήση διαγραμμάτων UML.

**Σε αυτή την ενότητα παρουσιάζουμε τις εισαγωγές και εξαγωγές των δεδομένων από το σύστημα. Ο χρήστης μπορεί να κάνει αναζήτηση προϊόντων στην εφαρμογή μας, να γίνει μέλος, να θέσει κάποιες προσφορές ως «μη ισχύουσες», να δει τις τιμές των προϊόντων και τις τοποθεσίες των καταστημάτων στον χάρτη. Επιπλέον μπορεί να δει το προφίλ του κ να επεξεργαστεί τα δεδομένα του. Όλα αυτά γίνονται με τη βοήθεια των διεπαφών με τον χρήστη, όπως θα εξηγηθούν και στην ενότητα 3.2. Επίσης η εφαρμογή μας επικοινωνεί με τη βάση δεδομένων ώστε να δίνει τη δυνατότητα στον χρήστη να μπορεί να δει, να προσθέσει ή και να τροποποιήσει κάποια δεδομένα. Η web εφαρμογή μας παίρνει τα γεωγραφικά δεδομένα ώστε να μπορεί να ξέρει το πού βρίσκεται ο χρήστης τη στιγμή που χρησιμοποιεί την εφαρμογή και με τη βοήθεια του Google Maps API, να αναπαραστήσει γραφικά τα δεδομένα που σχετίζονται με τοποθεσία. Επίσης για την παρουσίαση όλων των δεδομένων ο χρήστης χρειάζεται έναν web browser που να υποστηρίζει HTML5.**

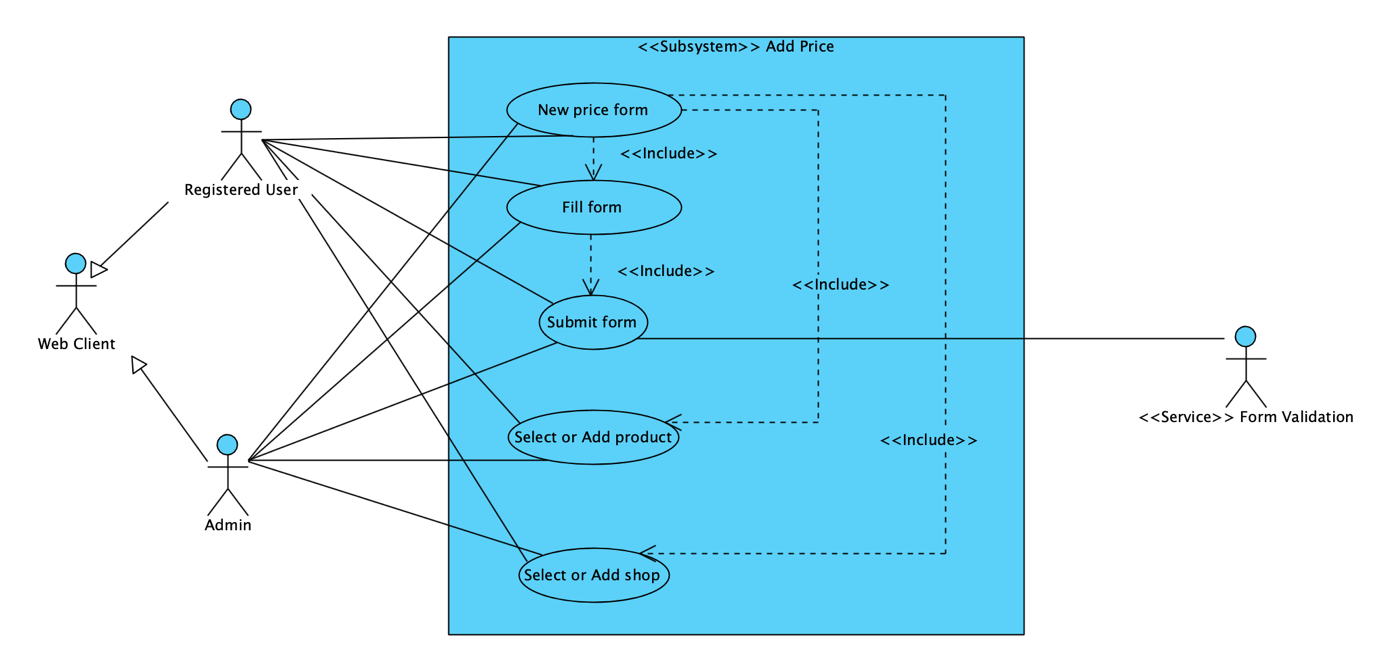
### **1.3.2 Διεπαφές με το χρήστη**

Προδιαγραφή διεπαφών με το χρήστη. Μοντέλο Use Case (UML).

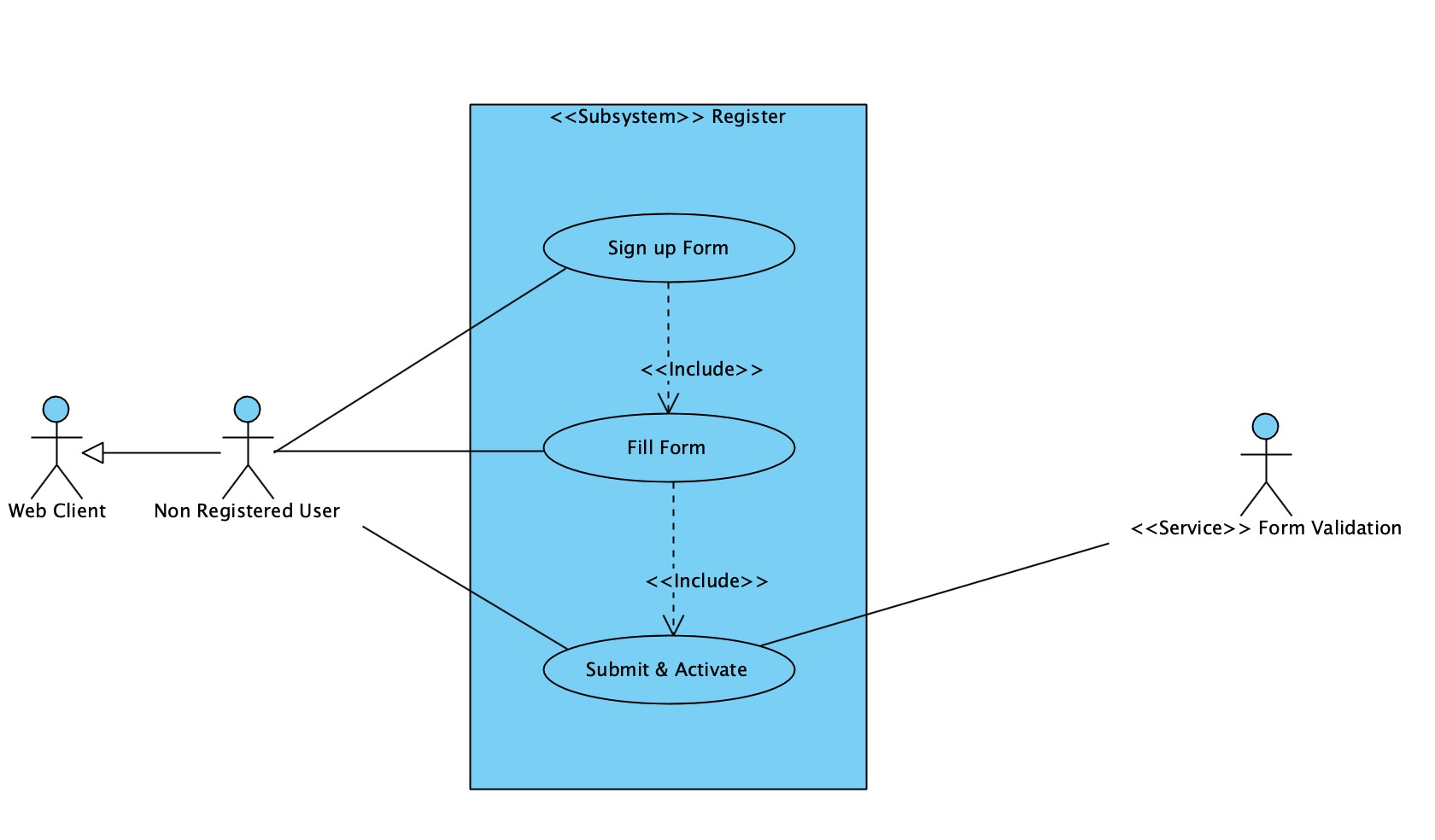
1. **Αναζήτηση-Φιλτράρισμα αποτελεσμάτων**



1. **Προσθήκη νέας τιμής προϊόντος**



1. **Δημιουργία λογαριασμού**



### **1.3.3 Διεπαφές με υλικό**

Προδιαγραφή διεπαφών με υλικό (εφόσον απαιτείται, πχ αναγνώστες κ.ά.)

ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ

**Δεν υπάρχει κάποια άμεση διεπαφή της εφαρμογής μας με υλικό. Το μόνο που μπορούμε να πούμε είναι ότι απαιτείται πρόσβαση στο διαδίκτυο και επιπλέον, ότι η τοποθεσία μας λαμβάνεται με βάση τον υπολογισμό τοποθεσίας που κάνει η Google με βάση τα στοιχεία του router που είμαστε συνδεδεμένοι.**

### **1.3.4 Διεπαφές επικοινωνιών**

Προδιαγραφή διεπαφών επικοινωνιών (αφορά στοιχεία λογισμικού που υλοποιούν τέτοιες διεπαφές, εφόσον υπάρχουν)

ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ

**Ο τρόπος με τον οποίο επικοινωνούν τα διάφορα μέρη του συστήματος μας είναι σημαντικός, αφού εξαρτάται το ένα με βάση το άλλο. Παρ’ολα αυτά ο τρόπος που επιτυγχάνεται αυτή η επικοινωνία δεν είναι σημαντικός για το σύστημα αφού για αυτή τη λειτουργία αναλαμβάνει το λειτουργικό σύστημα.**

# Αναφορές - πηγές πληροφοριών

Αναφορά σε πηγές πληροφοριών στο μέτρο της αναγκαιότητας για την κατανόηση του συστήματος

ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ

# Προδιαγραφές απαιτήσεων λογισμικού

### **3.1 Εξωτερικές διεπαφές**

Λεπτομερής τεχνική προδιαγραφή των διεπαφών που αναφέρονται στην ενότητα 1.3.1.

Προαιρετική χρήση λογισμικού προτυποποίησης διεπαφών χρήστη (mock-up).

**Αρχικά, ο οποιοσδήποτε χρήστης μπορεί να χρησιμοποιήσει την εφαρμογή μας ώστε να κάνει απλή αναζήτηση προϊόντων και καταστημάτων. Μπορεί επίσης να κάνει εγγραφή στο σύστημα μας, ώστε να μπορεί να χρησιμοποιήσει περισσότερες λειτουργίες που προσφέρει η εφαρμογή. Ο εγγεγραμμένος χρήστης έχει τη δυνατότητα να προσθέσει προϊόντα, καταστήματα και να τροποποιήσει την τιμή ενός προϊόντος όταν αντιληφθεί ότι αυτή έχει μεταβληθεί. Επίσης μπορεί να δει τη σελίδα προφίλ του στην οποία φαίνονται τα δεδομένα που έχει καταχωρήσει ο ίδιος στο σύστημα και αν θέλει να τα αλλάξει. Όσον αφορά τις επιπρόσθετες λειτουργίες που προσφέρει το σύστημα, η web εφαρμογή μας παίρνει τα γεωγραφικά δεδομένα ώστε να μπορεί να ξέρει το πού βρίσκεται ο χρήστης τη στιγμή που χρησιμοποιεί την εφαρμογή, ώστε να μπορεί να του προτείνει καταστήματα που βρίσκονται κοντά σε αυτόν ανά πάσα στιγμή, επικοινωνώντας ταυτόχρονα και με τη βάση δεδομένων. Η γραφική αναπαράσταση των δεδομένων των καταστημάτων γίνεται με τη βοήθεια του Google Maps API, το οποίο μας δίνει τη δυνατότητα να δούμε την τοποθεσία των καταστημάτων στον χάρτη. Επίσης κατά την προσθήκη κάποιου καταστήματος στο σύστημα, ο χρήστης μπορεί με τη βοήθεια του Google Maps API, να δηλώσει μόνο τη διεύθυνση του καταστήματος και να το επιβεβαιώσει από τον χάρτη, ή αν δεν γνωρίζει τη διεύθυνση ακριβώς, να βρει στον χάρτη το κατάστημα και η διεύθυνση θα ενημερωθεί άμεσα.**

## 3.2 Λειτουργίες: περιπτώσεις χρήσης

 Λεπτομερής προδιαγραφή των λειτουργιών του λογισμικού σε επίπεδο περιπτώσεων χρήσης.

Για κάθε μία λειτουργία δίνονται τα ακόλουθα.

### ***ΟΜΑΔΑ 6 ΑΤΟΜΩΝ: 3-4 περιπτώσεις χρήσης => Στις επόμενες σελίδες μπορούμε να δούμε τις περιπτώσεις χρήσης:***

### ***Αναζήτηση-Φιλτράρισμα Αποτελεσμάτων***

### ***Προσθήκη νέας τιμής προϊόντος***

### ***Δημιουργία λογαριασμόυ***

### 3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: (***Αναζήτηση-Φιλτράρισμα Αποτελεσμάτων***)

#### 3.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

**Σε αυτή την περίπτωση χρήσης λαμβάνουν μέρος όλοι οι χρήστες (διαχειριστής, εγγεγραμμένος, μη-εγγεγραμμένος).**

#### 3.2.1.2 Προϋποθέσεις εκτέλεσης

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

**Δεν υπάρχει κάποια προϋπόθεση που πρέπει να πληρείται ώστε να μπορεί ο οποιοσδήποτε χρήστης να κάνει αναζήτηση προϊόντος και τιμής και να δει τα αποτελέσματα.**

#### 3.2.1.3 Περιβάλλον εκτέλεσης

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

**Διαδικτυακή διεπαφή χρήστη, βάση δεδομένων καθώς και Google Maps API.**

#### 3.2.1.4 Δεδομένα εισόδου

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

**Οι όροι αναζήτησης καθώς και το φιλτράρισμα των δεδομένων με βάση τις προτιμήσεις του χρήστη. Οποιοσδήποτε όρος αναζήτησης είναι έγκυρος, εκτός κι αν δεν υπάρχει στη βάση προϊόν/κατάστημα που να ικανοποιεί τον όρο αναζήτησης/φιλτραρίσματος.**

#### 3.2.1.5 Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών

**Οποιοσδήποτε όρος αναζήτησης είναι έγκυρος, εκτός κι αν δεν υπάρχει στη βάση προϊόν/κατάστημα που να ικανοποιεί τον όρο αναζήτησης/φιλτραρίσματος.**

#### 3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

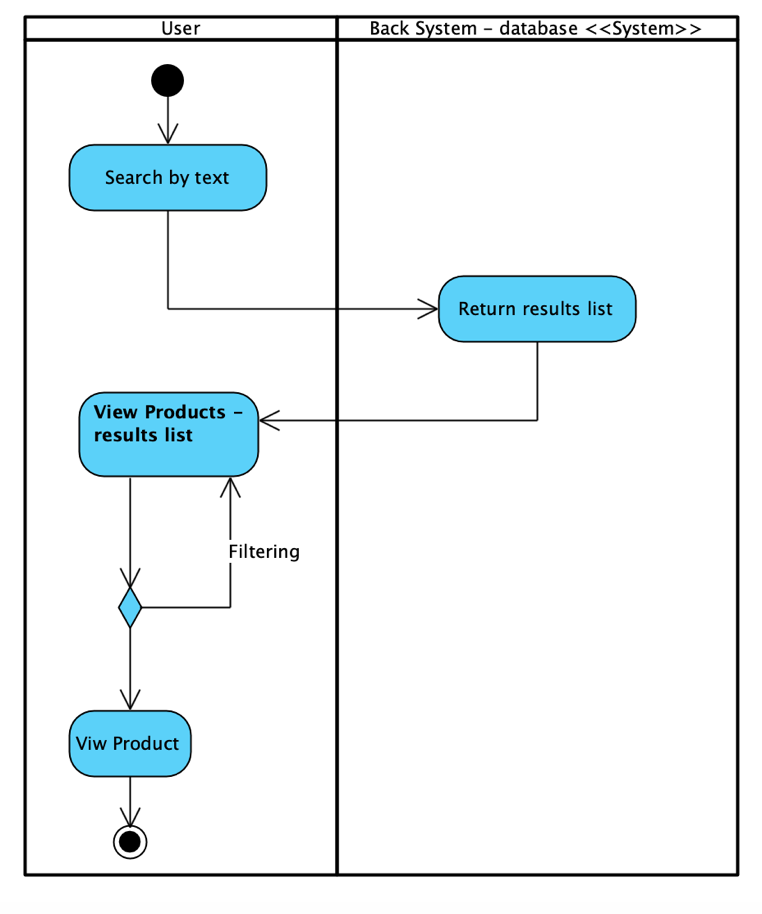
Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

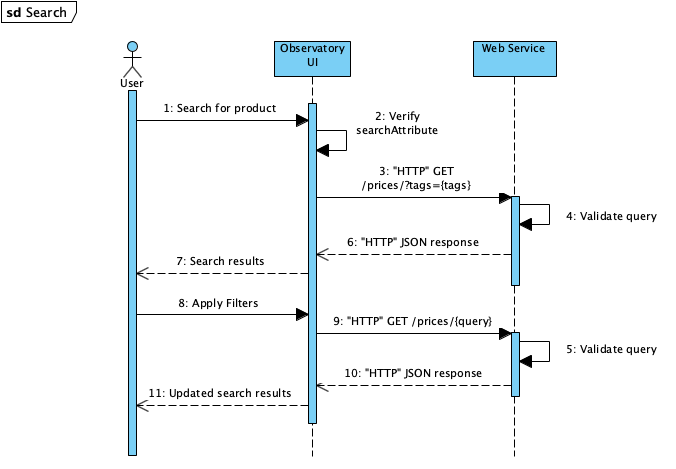
**1. Αναζήτηση προϊόντος**

**2. Επιστροφή λίστας αποτελεσμάτων**

**3. Προβολή αποτελεσμάτων στον χρήστη**

**4. Ο χρήστης μπορεί να επιλέξει να δει ένα προϊόν, ή να φιλτράρει τα αποτελέσματα του, ώστε να τα κάνει πιο συγκεκριμένα και να τα φέρει στα μέτρα του. Στη συνέχεια μπορεί να επιλέξει το προϊόν που θέλει για να δει περισσότερες λεπτομέριες.**





#### 3.2.1.7 Δεδομένα εξόδου

Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

**Δεν υπάρχουν δεδομένα εισόδου αφού ο χρήστης κάνει μόνο αναζήτηση των δεδομένων που υπάρχουν στη βάση.**

#### 3.2.1.8 Παρατηρήσεις

Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει

### 3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: (***Προσθήκη νέας τιμής προϊόντος***)

#### 3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

**Σε αυτή την περίπτωση χρήσης λαμβάνουν μέρος όλοι οι εγγεγραμμένοι χρήστες (διαχειριστής, εγγεγραμμένος) και όχι οι μη-εγγεγραμμένοι.**

#### 3.2.2.2 Προϋποθέσεις εκτέλεσης

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

**Για να εκτελεστεί η συγκεκριμένη περίπτωση χρήσης θα πρέπει ο χρήστης να έχει λογαριασμό στην εφαρμογή ώστε να φαίνεται διαθέσιμη η επιλογή προσθήκης καινούργιας τιμής (καταστήματος ή προϊόντος).**

#### 3.2.2.3 Περιβάλλον εκτέλεσης

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

**Διαδικτυακή διεπαφή χρήστη και βάση δεδομένων.**

#### 3.2.2.4 Δεδομένα εισόδου

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

**Ο χρήστης επιλέγει προϊόν και κατάστημα από την ήδη υπάρχουσα λίστα. Αν δεν υπάρχει στη λίστα είτε προϊόν είτε κατάστημα, τότε καλείται να τα καταχωρήσει. Στη συνέχεια επιλέγει την τιμή και την ημερομηνία της προσφοράς. Τα δεδομένα θεωρούμε πως είναι έγκυρα και καταχωρούνται στο σύστημα. Στη συνέχεια ο διαχειριστής θα έχει την ευχέρεια να διαγράψει ένα προϊόν αν ο χρήστης δεν έχει δώσει σωστές πληροφορίες.**

#### 3.2.2.5 Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών, εφόσον υπάρχουν παράμετροι.

**Θεωρούμε πως τα δεδομένα που καταχωρεί ο χρήστης στο σύστημα όσον αφορά τα καταστήματα και τα προϊόντα κατά τη δημιουργία τους είναι σωστά. Επομένως οι συνθήκες εγκυρότητας αφορούν αποκλειστικά τη νέα τιμή του προϊόντος η οποία πρέπει να είναι σε μορφή αριθμού καθώς και την ημερομηνία για την οποία ισχύει αυτή η τιμή. Ο έλεγχος των παραπάνω γίνονται δυναμικά, οπότε είμαστε σίγουροι ότι ο χρήστης καταχωρίζει στο σύστημα σωστά δεδομένα.**

#### 3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

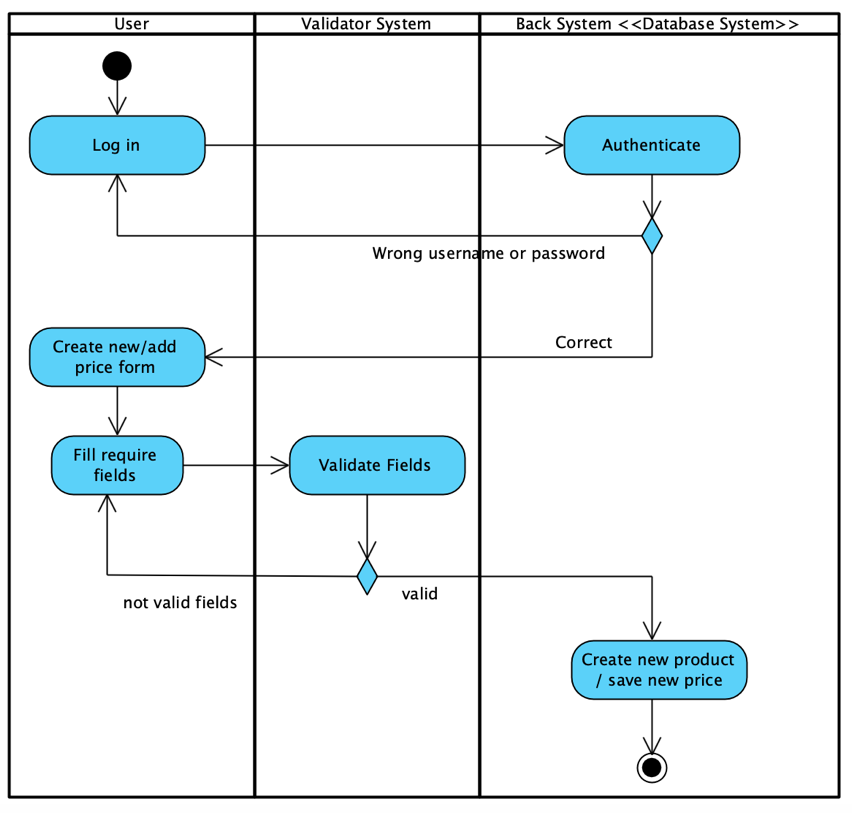
Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

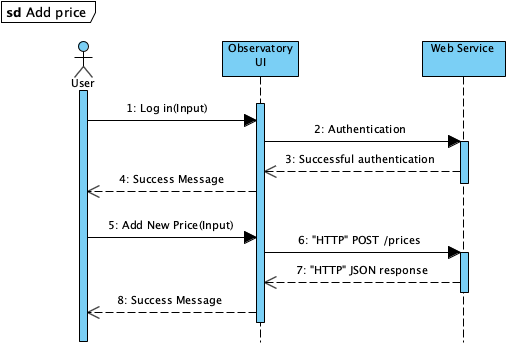
**1. Εισαγωγή στο σύστημα. Κατά τη διάρκεια αυτής της ενέργειας γίνεται ο έλεγχος ορθών δεδομένων από τον χρήστη. Αν ο χρήστης προσπάθησε να εισέλθει στο σύστημα με λανθασμένα στοιχεία, ο χρήστης καλείται να καταχωρήσει τα σωστά.**

**2. Αφού έχει εισέλθει στο σύστημα ο εγγεγραμμένος χρήστης, στη συνέχεια κάνει κλικ στο κουμπί «Προσθήκη τιμής προϊόντος».**

**3. Συμπληρώνει τη φόρμα εισαγωγής τιμής επιλέγοντας τα κατάλληλα στοιχεία (προϊόν, κατάστημα, τιμή, ημερομηνίες)**

**4. Καταχωρίζει την φόρμα στο σύστημα. Αν τα δεδομένα που έχει καταχωρήσει δεν είναι σωστά επιστρέφει στη φόρμα ώστε να αλλάξει τα λανθασμένα πεδία. Αντιθέτως, αν τα στοιχεία που έχει καταχωρήσει είναι σωστά, τότε δημιουργείται η νέα τιμή.**

****



#### 3.2.2.7 Δεδομένα εξόδου

Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

**Σε αυτή την περίπτωση χρήσης έχουμε δεδομένα εξόδου, αφού ο χρήστης έχει τη δυνατότητα να μεταβάλει την τιμή ενός προϊόντος σε κάποιο συγκεκριμένο κατάστημα. Επίσης ο χρήστης μπορεί να δημιουργήσει καινούργια δεδομένα, αφού αν δεν υπάρχει ένα προϊόν ή ένα κατάστημα στη βάση, έχει τη δυνατότητα να το εισάγει στο σύστημα.**

#### 3.2.2.8 Παρατηρήσεις

Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει

### 3.2.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: (***Δημιουργία Λογαριασμού***)

#### 3.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

**Σε αυτή την περίπτωση λαμβάνουν μέρος αποκλειστικά οι μη-εγγεγραμμένοι χρήστες, οι οποίοι στο τέλος της διαδικασίας είναι πλέον εγγεγραμμένοι.**

#### 3.2.3.2 Προϋποθέσεις εκτέλεσης

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

**Η μοναδική συνθήκη που πρέπει να ικανοποιείται είναι, ο μη-εγγεγραμμένος χρήστης ο οποίος θα δημιουργήσει λογαριασμό στην εφαρμογή να μην έχει ήδη λογαριασμό.**

#### 3.2.3.3 Περιβάλλον εκτέλεσης

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

**Διαδικτυακή διεπαφή χρήστη και βάση δεδομένων.**

#### 3.2.3.4 Δεδομένα εισόδου

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

**Ο μη-εγγεγραμμένος χρήστης καταχωρίζει τα δεδομένα του (προσωπικά στοιχεία) στη φόρμα. Τα δεδομένα θεωρούμε πως είναι έγκυρα αφού ο χρήστης εισάγει τα δικά του στοιχεία τα οποία και θα χρησιμοποιεί στη συνέχεια της πλοήγησης του στην εφαρμογή.**

#### 3.2.3.5 Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών

**Οι παράμετροι που μπορούμε να ελέγξουμε την εγκυρότητα τους όσον αφορά τον τύπο δεδομένων, είναι ο κωδικός πρόσβασης (ο οποίος πρέπει να εισαχθεί 2 φορές για επιβεβαίωση κ ο οποίος πρέπει να έχει περισσότερους από 6 χαρακτήρες), η ημερομηνία γέννησης η οποία υποχρεωτικά από τη φόρμα πρέπει να είναι ημερομηνία, όπως επίσης το τηλέφωνο και η διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη.**

#### 3.2.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

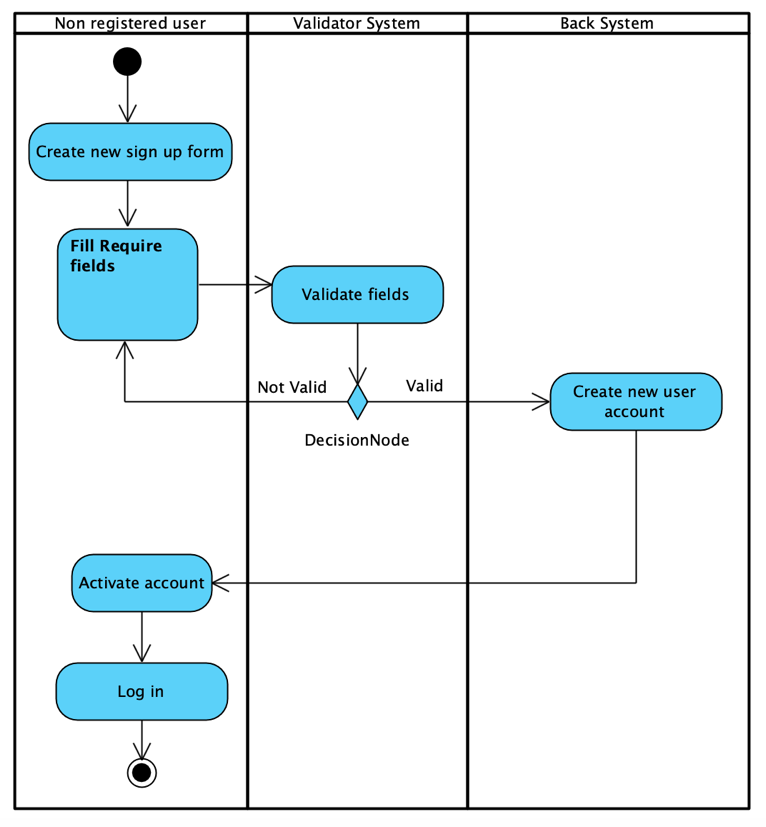
**1. Ο μη εγγεγραμμένος χρήστης πατάει στο κουμπί «Δημιουργία λογαριασμού» και εμφανίζεται η φόρμα εισαγωγής των στοιχείων.**

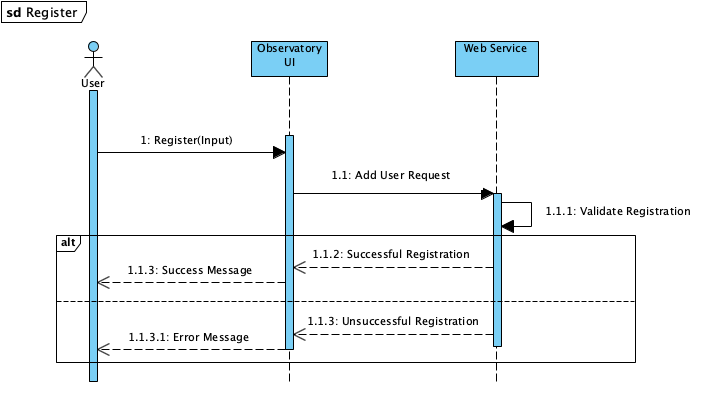
**2. Ο χρήστης συμπληρώνει τα στοιχεία του**

**3. Καταχωρίζει την φόρμα εισαγωγής στο σύστημα.**

**4. Γίνεται ο έλεγχος ορθότητας των πεδίων από την φόρμα εισαγωγής. Αν υπάρχει κάποιο πρόβλημα ο χρήστης καλείται να ξανασυμπληρώσει τη φόρμα εισαγωγής διορθώνοντας τα στοιχεία που πρέπει και καταχωρώντας τη φόρμα ξανά.**

**5. Ο λογαριασμός του χρήστη δημιουργείται και πλέον ο χρήστης είναι εγγεγραμμένος.**





#### 3.2.3.7 Δεδομένα εξόδου

Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

**Τα δεδομένα εξόδου που δημιουργούνται κατά τη διάρκεια αυτής της περίπτωσης χρήσης είναι ο λογαριασμός του πλέον εγγεγραμμένου χρήστη. Κατά την καταχώρηση της φόρμας εγγραφής στο σύστημα, δημιουργείται ο λογαριασμός. Δεν υπάρχει κάποια μεταβολή σε υπάρχοντα δεδομένα κατά την εκτέλεση της δημιουργίας λογαριασμού.**

#### 3.2.3.8 Παρατηρήσεις

Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει

## 3.3 Απαιτήσεις επιδόσεων

Ποσοτική τεκμηρίωση μέτρων και κριτηρίων επιθυμητών επιδόσεων με αναφορά στα ποσοτικά χαρακτηριστικά εισόδων και φορτίου του λογισμικού.

**Το σύστημα είναι διαδραστικό, επομένως οι καθυστερήσεις θα πρέπει να είναι όσο το δυνατόν μικρότερες, και έτσι σε κάθε εντολή-απάντηση δεν πρέπει να υπάρχουν μεγάλες καθυστερήσεις. Σε κάθε εντολή που δίνει ο χρήστης για μεταφορά από μια σελίδα σε άλλη, ο χρόνος απόκρισης θα πρέπει να είναι αρκετά μικρός (πχ max 2δευτερόλεπτα) ώστε να προσφέρει στον χρήστη μια ευχάριστη πλοήγηση. Επιπλέον, η αλληλεπίδραση του συστήματος μας με τη βάση δεδομένων δεν πρέπει να δημιουργεί μεγάλη καθυστέρηση και τα δεδομένα πρέπει να πηγαινοέρχονται σε μικρό χρόνο (πχ max 2δευτερόλεπτα).**

## 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.

Αναλυτική αναφορά στα δεδομένα εισόδου, τα σχετικά πρότυπα δεδομένων και υπηρεσιών, καθώς και σε μετρικές που σχετίζονται με τα δεδομένα (storage capacity planning).

**Το λογισμικό διαχειρίζεται δεδομένα τα οποία είναι ευαίσθητα, όπως τα προσωπικά δεδομένα των χρηστών. Επίσης διαχειρίζεται δεδομένα τα οποία είναι ανοικτά στο ευρύ κοινό. Όλα αυτά τα δεδομένα εισόδου, που προέρχονται από τους χρήστες, καταχωρούνται στη βάση δεδομένων του συστήματος. Επιβάλλεται τα δεδομένα και οι υπηρεσίες να συμμορφώνονται με τα πρότυπα που έχει καθορίσει η IEEE ώστε να μην υπάρχουν διχογνωμίες. Η χωρητικότητα της βάσης πρέπει να είναι επεκτάσιμη, αφού από το έγγραφο StRS, προκύπτει ότι υπάρχουν μεγάλες βλέψεις για την αποδοχή της εφαρμογής από τον κόσμο. Επομένως πρέπει να υπάρχει η ικανότητα να προσαρμοστεί η εφαρμογή στο storage growth.**

### 3.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

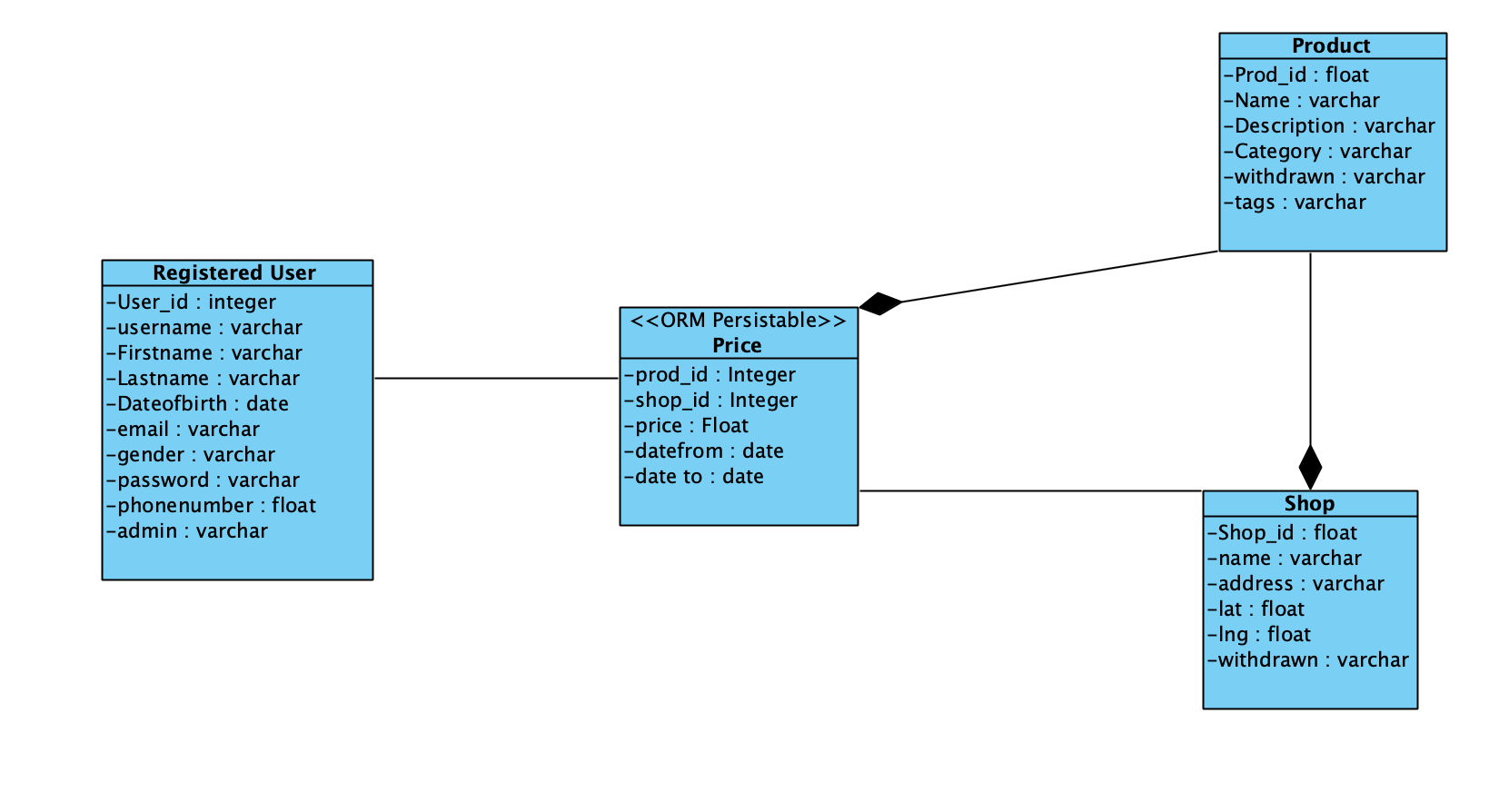
Απαιτήσεις πρόσβασης και περιορισμοί.

**Ο διαχειριστής έχει πρόσβαση σε όλα τα δεδομένα που υπάρχουν στην βάση, έχοντας και τη δυνατότητα να τα επεξεργαστεί (διαγραφή ή αλλαγή κάποιων δεδομένων). Ο εγγεγραμμένος χρήστης έχει τη δυνατότητα να έχει πρόσβαση στα δεδομένα που αφορούν τον λογαριασμό του, καθώς επίσης και στα δεδομένα που υπάρχουν στη βάση και αφορούν τιμές, προϊόντα και καταστήματα. Τέλος, ο μη-εγγεγραμμένος χρήστης έχει πρόσβαση μόνο στα δεδομένα της βάσης που είναι διαθέσιμα σε όλους (τιμές, προϊόντα, καταστήματα).**

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

Μοντέλα δεδομένων UML ή/και ER

**UML Class Diagram:**



**UML ER Diagram: Μια σημείωση που πρέπει να μπει εδώ, είναι το γεγονός ότι λόγω του ότι έγινε χρήση της NoSQL βάσης δεδομένων MongoDB, δεν υπάρχουν σχέσης μεταξύ των οντοτήτων. Επομένως, δημιουργήσαμε το ER Diagram, με βάση τη γενική εικόνα που είχαμε στο μυαλό μας για τη βάση, ώστε να είναι πιο εύκολη και ξεκάθαρη η υλοποίηση της βάσης.**



### 3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Κανόνες ακεραιότητας και εγκυρότητας δεδομένων

**Τα δεδομένα καταχωρούνται στο σύστημα όπως τα έχει καταχωρήσει ο χρήστης στην φόρμα εισαγωγής, χωρίς να υπάρχουν μη-ηθελημένες αλλαγές σε αυτά κατά την εισαγωγή τους στη βάση. Επίσης, θέμα ακεραιότητας δεδομένων προκύπτει και όταν ένας χρήστης χωρίς δικαιοδοσία, καταφέρει να δει/αλλάξει δεδομένα στα οποία δεν δικαιούται να έχει πρόσβαση. Επομένως, πρέπει να διασφαλίζεται από το σύστημα ότι μη εξουσιοδοτημένοι χρήστες δε θα έχουν πρόσβαση σε δεδομένα που δεν είναι προσβάσιμα σε αυτούς.**

### 3.4.5 Προδιαγραφές διατήρησης δεδομένων

Απαιτήσεις διατήρησης δεδομένων σε βάθος χρόνου.

**Τα δεδομένα που χρειάζεται να διατηρούμε σε βάθος χρόνου είναι όσα δεδομένα εισάγονται στο σύστημα, εκτός κι αν ο διαχειριστής αποφασίσει να διαγράψει κάποια. Στοιχεία χρηστών επιβάλλεται να διατηρούνται στο σύστημα σε βάθος χρόνου, εκτός κι αν το ζητήσει ο ίδιος ο χρήστης να διαγραφούν. Επιπλέον, τα δεδομένα που αφορούν τιμές προϊόντων, προϊόντα και καταστήματα, διατηρούνται στο σύστημα μέχρι ο διαχειριστής να αποφασίσει ότι πλέον δε θα μπορούν να χρησιμοποιηθούν για τη βελτίωση των υπηρεσιών της εφαρμογής.**

## 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

**Αρχικά ο πρώτος περιορισμός που έχουμε στη σχεδίαση του συστήματος, αφορά τις γλώσσες προγραμματισμού που έχουμε χρησιμοποιήσει. Για το API χρησιμοποιήθηκε Javascript ενώ για το back-end και το front-end έγινε χρήση της Angular. Επίσης για τη βάση δεδομένων χρησιμοποιήθηκε η MongoDb. Το σύστημα μπορεί να λειτουργήσει σε κάθε λειτουργικό σύστημα που υποστηρίζει κάποιον web-browser, εφόσον υπάρχει βέβαια σύνδεση στο διαδίκτυο. Επίσης, τα γεωγραφικά δεδομένα που λαμβάνονται ώστε να εξυπηρετήσουν διάφορες λειτουργίες της εφαρμογής, δεν είναι πάντα ακριβή, λόγω του γεγονότος ότι λαμβάνονται με βάση τον server στον οποίο είναι συνδεδεμένος ο χρήστης, και όχι μέσω κάποιου υλικού GPS. Επιπλέον, όσον αφορά το API, υπήρχαν αρκετοί περιορισμοί από τον stakeholder που έχει μας έχει αναθέσει την υλοποίηση του συστήματος, αφού δεν μπορούσαμε να αλλάξουμε την πολιτική ονοματολογίας οντοτήτων δεδομένων και πεδίων.**

## 3.6 Λοιπές απαιτήσεις

### 3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Τεκμηρίωση απαιτήσεων διαθεσιμότητας

**Το σύστημα είναι 24/7 διαθέσιμο στους χρήστες. Επιπλέον, αν η σύνδεση διακοπεί ενώ στέλνονται δεδομένα στη βάση, τότε θα πρέπει τα δεδομένα αυτά να ξανασταλούν για επιβεβαίωση ώστε να υπάρχει και συνέπεια. Πρέπει να υπάρχει συνεχής διασύνδεση με το διαδίκτυο ώστε να υπάρχει επικοινωνία της εφαρμογής με τη βάση, όπως επίσης και να δίνεται η ευκαιρία στο να λαμβάνονται τα γεωγραφικά δεδομένα των χρηστών.**

### 3.6.2 Απαιτήσεις ασφάλειας

Τεκμηρίωση απαιτήσεων ασφαλείας

**Ο μεγαλύτερος κίνδυνος ασφαλείας αφορά τα προσωπικά δεδομένα του χρήστη και κυρίως κωδικούς πρόσβασης. Οπότε ο μηχανισμός που χρησιμοποιείται για εγγραφή και εισαγωγή του χρήστη στο σύστημα πρέπει να διασφαλίζει ότι θα γίνει με ασφάλεια χωρίς ο χρήστης να πέσει θύμα υποκλοπής. Επομένως πρέπει τα δεδομένα αυτά να στέλνονται κρυπτογραφημένα, ώστε να διασφαλιστεί η ασφάλεια των δεδομένων των χρηστών.**

### 3.6.3 Απαιτήσεις συντήρησης

Τεκμηρίωση απαιτήσεων συντήρησης

**Το σύστημα είναι εύκολα συντηρήσιμο, αφού οποιοδήποτε λάθος μπορεί να βρεθεί και να διορθωθεί άμεσα. Επιπλέον, οποιαδήποτε απαίτηση από το εξωτερικό περιβάλλον μπορεί να υλοποιηθεί εύκολα και γρήγορα λόγω της προσεγμένης σχεδίασης που έγινε από την ομάδα από την αρχή της υλοποίησης της εφαρμογής, η οποία έγινε με βλέψεις και στο μέλλον. Οποιαδήποτε νέα λειτουργία μπορεί να υλοποιηθεί, αφού ο κώδικας είναι εύκολα συντηρήσιμος και επεκτάσιμος.**

# Παράρτημα

## 4.1 Παραδοχές και εξαρτήσεις

**1. Οι εγγεγραμμένοι χρήστες δίνουν σωστές πληροφορίες για τις τιμές των προϊόντων, επομένως δεν χρειάζεται έγκριση από τους διαχειριστές. Αν υποπέσει στην αντίληψη των διαχειριστών ότι κάτι είναι λάθος, μπορούν απλά να διαγράψουν την καταχώρηση.**

**2. Όταν ένας εγγεγραμμένος χρήστης δημιουργήσει νέο προϊόν ή κατάστημα, θεωρούμε πως τα δεδομένα που έχει δώσει είναι σωστά, οπότε δεν χρειάζεται έγκριση από τον διαχειριστή. Αν υποπέσει στην αντίληψη των διαχειριστών ότι κάτι είναι λάθος, μπορούν απλά να διαγράψουν την καταχώρηση ή ακόμα και τον χρήστη ο οποίος δεν συμπεριφέρεται με βάση τους κανόνες του συστήματος.**

## 4.2 Ακρωνύμια και συντομογραφίες

***24/7: 24 ώρες το 24ώρο, ανά 7 ημέρες την βδομάδα***

## 4.3 Υποστηρικτικά έγγραφα, πρότυπα κ.λπ.