ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Τεχνολογία Λογισμικού**

Έγγραφο απαιτήσεων λογισμικού (SRS)

Version 1.0

21 Φεβρουαρίου, 2019

Κακό Λογοπαίγνιο

Μενής - Μαστρομιχαλάκης Ορφέας

Αφουξενίδης Ευστάθιος

Γερακάρης Απόστολος

Χαλλιορή Μαρκέλλα

Μηναΐδη Μαρία Νεκταρία

Χανής Γεώργιος

**Περιεχόμενα**

**1.0 Introduction………………………………………………………………...4**

1.1 Purpose.........................................................................................................4  
 1.2 Overview .....................................................................................................4  
 1.3 External Interface Requirements…………………………………………..4

1.3.1 Hardware Interfaces................................................................................4

1.3.2 User Interfaces .......................................................................................5

1.4 References....................................................................................................5

**3.0 External Interface Requirements………………………………………..15**

3.1 External Interfaces……………………………………………………….45

3.2 System Functions: Use Cases…………………………………………....89

3.2.1 Use Case 1: List new Gas Station and add product price…..………...89

3.2.2 Use Case 2: Register and search…………………………………….88

3.2.3 Use Case 3: Block unwanted user……………………………………..

3.3 Performance Requirements………………………………………………22

3.4 Data Organisation Requirements………………………………………...33

3.4.1 Technical Description of data managed by the software and related metric data input etc……………………………………………………………33

3.4.2 Access Requirements and Constrictions……………………………...22

3.4.3 Data Model……………………………………………………………09

3.4.4 Data Integrity Specifications………………………………………….33

3.4.5 Data Retention Standards……………………………………………..77

3.5 Design Constraints……………………………………………………….33

3.6 Other Requirements……………………………………………………...33

3.6.1 Software Availability Requirements………………………………….22

3.6.2 Security Requirements………………………………………………..33

3.6.3 Maintenance Requirements…………………………………………...33

**4.0 Appendix…………………………………………………………………..33**

4.1 Assumptions and Dependencies……………………………………….44

4.2 Acronyms and Abbreviations…………………………………………...55

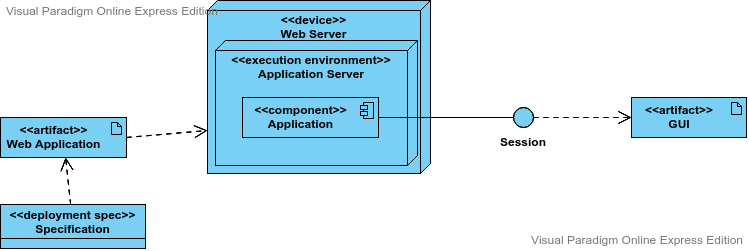
4.3 Supporting Documents, Standards etc…………………………………..77

**1.0 Εισαγωγή**

## **1.1 Purpose**

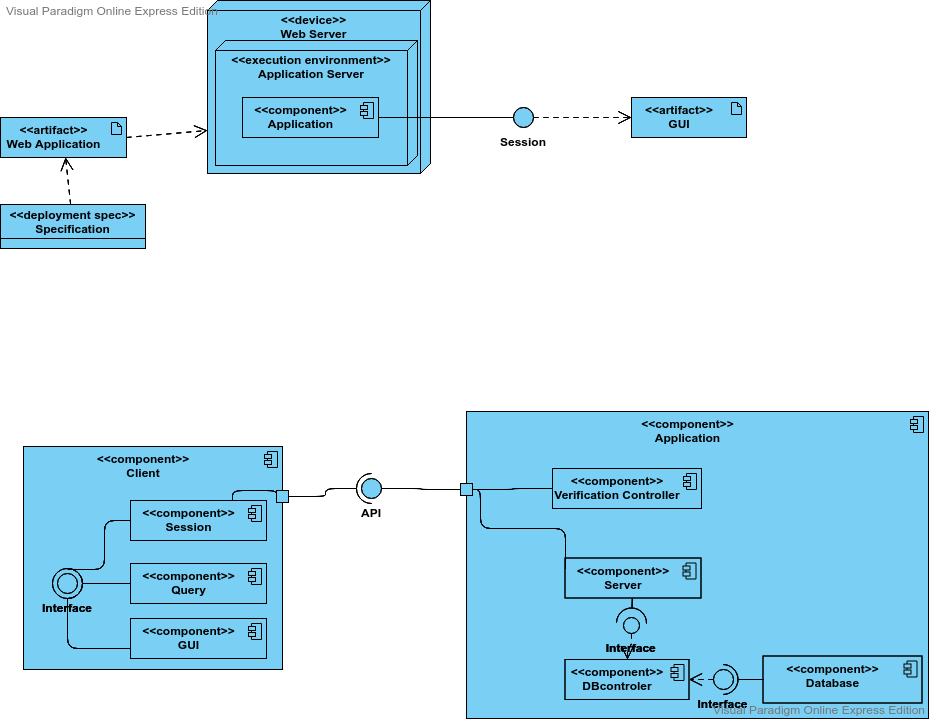
Σκοπός του συστήματος είναι να προσφέρει ένα εύχρηστο, χρήσιμο και γρήγορο εργαλείο, με στόχο την διευκόλυνση του καταναλωτή στην έρευνα αγοράς του, με συνέπεια άμεσα και έμμεσα κέρδη για τους εμπλεκόμενους. Μέσω του συστήματός μας ο χρήστης θα έχει την δυνατότητα να βρει εύκολα το επιθυμητό προϊόν με τη χαμηλότερη τιμή στην αντίστοιχη περιοχή ενδιαφέροντος.

## **1.2 Overview**



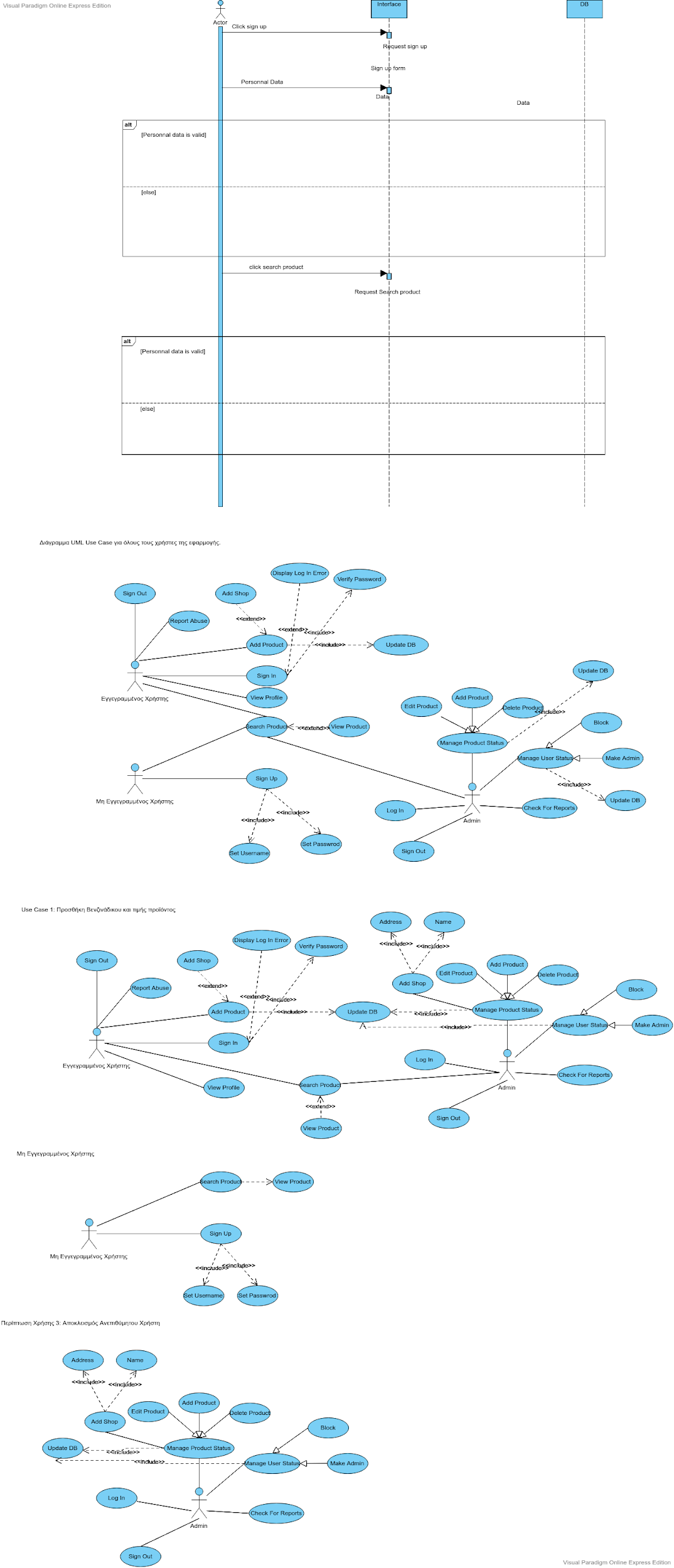
**1.3 External Interface Requirements**

**1.3.1 Hardware Interfaces**

****

**1.3.2 User Interfaces**

Στο παρακάτω διάγραμμα φαίνονται οι περιπτώσεις χρήσης όλων των χρηστών της εφαρμογής.

**

# **1.4 References**

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

*ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ*

*IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.*

IEEE Computer Society, 1998.

**2.0 External Interface Requirements**

## **2.1 External Interfaces**

*Λεπτομερής τεχνική προδιαγραφή των διεπαφών που αναφέρονται στην ενότητα 1.3.1.*

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

Περιγραφή του component.

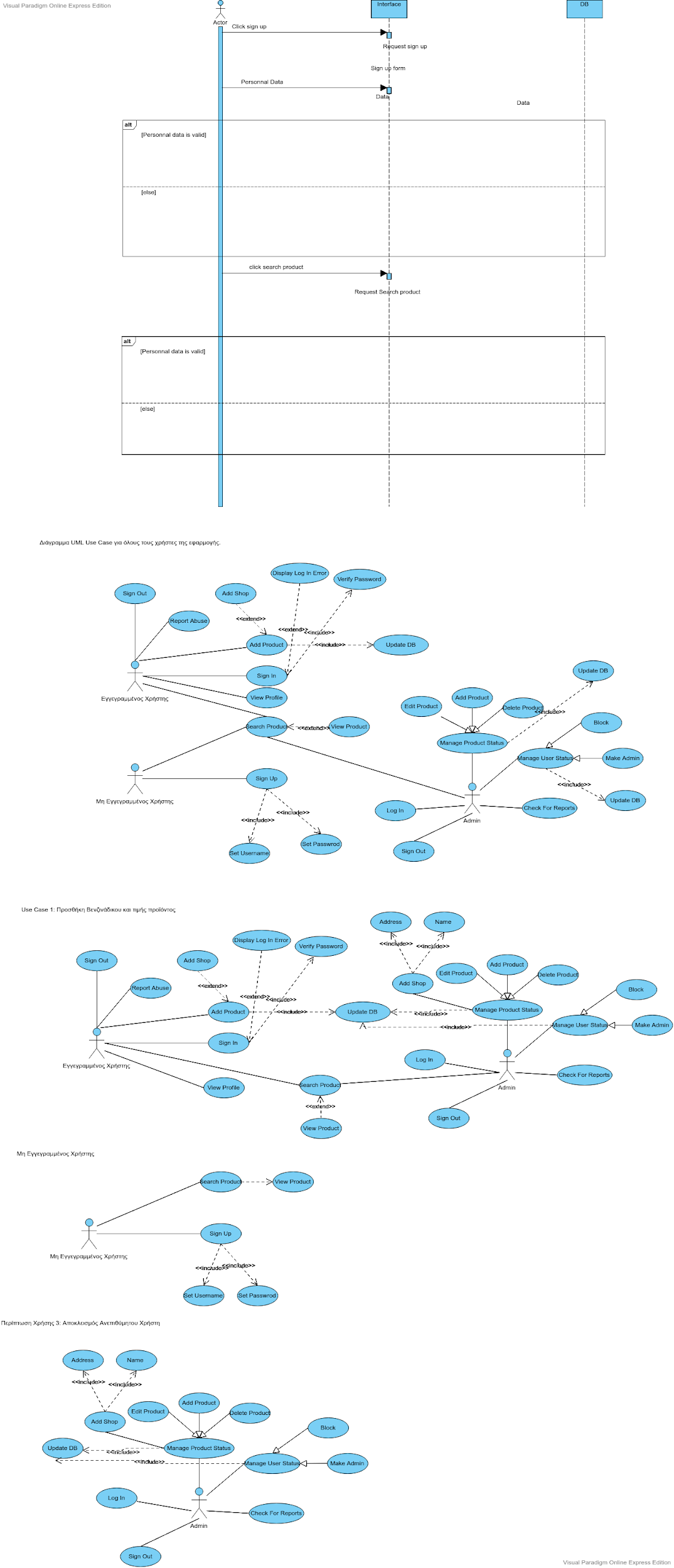
**3.2 System Functions: Use Cases**

### **3.2.1 Use Case 1:** Προσθήκη βενζινάδικου και τιμής προϊόντος

#### **3.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται**

Οι χρήστες που εμπλέκονται είναι ο εγγεγραμμένος χρήστης. Συγκεκριμένα ο εγγεγραμμένος χρήστης εμπλέκεται ,καθώς αυτός είναι που προσθέτει στον χάρτη το βενζινάδικο και την αντίστοιχη τιμή του.

Παρακάτω φαίνεται και το αντίστοιχο Use Case διάγραμμα για την πρώτη περίπτωση χρήσης.



#### **3.2.1.2 Προϋποθέσεις εκτέλεσης**

Για να μπορεί να εκτελεστεί η συγκεκριμένη περίπτωση χρήσης θα πρέπει ο χρήστης να κάνει log in για να μπει στο περιβάλλον του εγγεγραμμένου χρήστη.

#### **3.2.1.3 Περιβάλλον εκτέλεσης**

Το περιβάλλον στο οποίο εκτελείται η προσθήκη βενζινάδικου και τιμής, είναι η διαδικτυακή διεπαφή του εγγεγραμμένου χρήστη.

#### **3.2.1.4 Data Input**

Τα δεδομένα εισόδου είναι τα στοιχεία που συμπληρώνει ο εγγεγραμμένος χρήστης στα κελιά. Συγκεκριμένα το όνομα χρήστη, ή τη διεύθυνση ηλεκτρονικού ταχυδρομείου του, και τον κωδικό του.

#### **3.2.1.5 Parameters**

*Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών*

Θεωρούμε ότι η συγκεκριμένη ενέργεια μπορεί να εκτελεστεί είτε από έναν εγγεγραμμένο χρήστη, είτε από τον κάποιον administrator. Σε οποιαδήποτε από τις δύο προηγούμενες περιπτώσεις ο χρήστης θα πρέπει να δώσει την ονομασία του βενζινάδικου, τη διεύθυνσή του, και το είδος της βενζίνης. Η διεύθυνση θα πρέπει να έχει τη μορφή <Δρόμος>, <Αριθμός>, <Περιοχή>, <ΤΚ> προκειμένου να θεωρείται έγκυρη.

#### **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).*

Βημα 1: Ο χρήστης κάνει log in με τα προσωπικά του στοιχεία.

1. 1.Τα δεδομένα που πληκτρολόγησε είναι έγκυρα, οπότε το σύστημα μεταφέρει τον χρήστη στο περιβάλλον του εγγεγραμμένου χρήστη.
2. Τα δεδομένα που πληκτρολόγησε δεν είναι έγκυρα, οπότε το σύστημα ενημερώνει τον χρήστη με αντίστοιχο μήνυμα και του ζητάει να πληκτρολογήσει ξανά τα στοιχεία του.

Βήμα 2: Ο χρήστης κάνει κλικ στο κουμπί Add Price και μεταφέρεται σε μία σελίδα όπου θα προσθέσει το προϊόν, το βενζινάδικο και την τιμή του. Σε αυτό το σημείο ο χρήστης επιλέγει με όποια σειρά θέλει να διαλέξει τι θα συμπληρώσει. Οι επιλογές του είναι 3:

1. Να επιλέξει το είδος του προϊόντος από μια drop-down list από τα ήδη υπάρχοντα προϊόντα της βάσης και θα έχουμε 2 υποπεριπτώσεις:
2. Να υπάρχει το προϊόν στην λίστα και να το επιλέξει και να συνεχίσει την υποβολή των άλλων στοιχείων.
3. Να μην υπάρχει το προϊόν στην λίστα και να πατήσει στο κουμπί Add Product, έτσι ώστε να προσθέσει ένα νέο προϊόν και να το περιγράψει. Ύστερα συνεχίζει στην υποβολή των άλλων στοιχείων.

2. Να επιλέξει το βενζινάδικο από μια drop-down list από τα ήδη υπάρχοντα βενζινάδικα της βάσης και

έχουμε πάλι δύο περιπτώσεις:

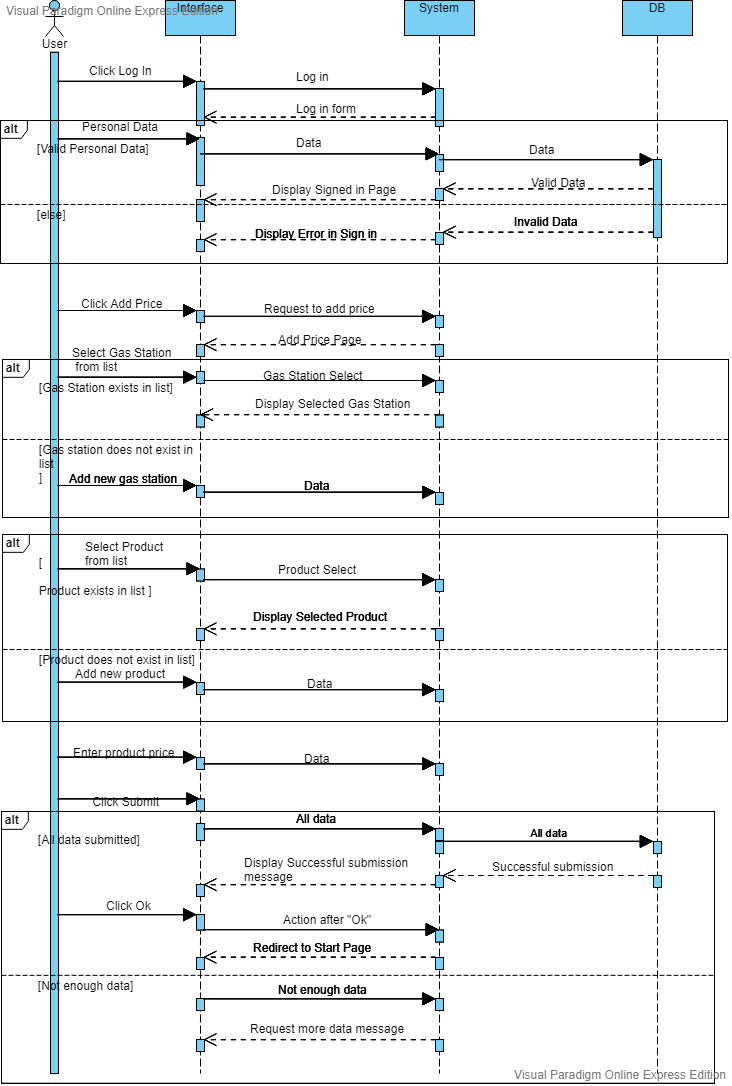
1. Να υπάρχει το βενζινάδικο στη λίστα, να το επιλέξει και να συνεχίσει την υποβολή των άλλων στοιχείων.
2. Να μην υπάρχει το βενζινάδικο στην λίστα και να πατήσει στο κουμπί Add Shop, έτσι ώστε να προσθέσει ένα νέο βενζινάδικο με τα στοιχεία του. Έπειτα θα συνεχίσει με την υποβολή των άλλων στοιχείων.

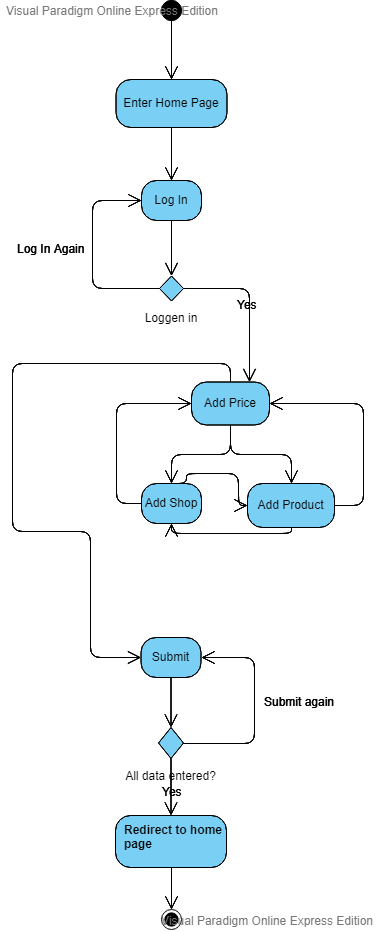
3. Να συμπληρώσει την τιμή του προϊόντος που επιθυμεί στο αντίστοιχο πεδίο και να συνεχίσει με την υποβολή των υπόλοιπων στοιχείων.

Βήμα 3: Ο χρήστης αφού συμπληρώσει όλα τα πεδία που είναι απαραίτητα πατάει το κουμπί Submit και υποβάλλει το προϊόν μαζί με την τιμή του και το βενζινάδικο που βρίσκεται.Σε περίπτωση που δεν έχουν συμπληρωθεί όλα τα πεδία, αποστέλλεται μήνυμα λάθους.

Βήμα 4: Εμφανίζεται μήνυμα επιτυχούς υποβολής στο οποίο ο χρήστης πατάει το κουμπί ΟΚ και τον ανακατευθύνει αυτόματα στην αρχική σελίδα.

Παρακάτω παρατίθεται το αντίστοιχο sequence diagram.





#### **3.2.1.7 Data Output**

*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 εισόδου είναι τα στοιχεία που συμπληρώνει ο εγγεγραμμένος χρήστης στα κελιά. Συγκεκριμένα συμπληρώνει το όνομα, τη διεύθυνση (και συμπληρώνονται οι συντεταγμένες αυτόματα μέσω του χάρτη) του βενζινάδικου και tags που προσδιορίζουν το κατάστημα.Επιπλέον συμπληρώνει το είδος βενζίνης/προϊόντος και εκεί ανάλογα έχουμε 2 περιπτώσεις:

1. αν υπάρχει το προϊόν στην drop-down list, το επιλέγει και συμπληρώνει την τιμή του προϊόντος και κάποια σχόλια.
2. αν δεν υπάρχει το προϊόν, προσθέτει το καινούργιο προϊόν και συμπληρώνει την τιμή προϊόντος και κάποια σχόλια.

Παρακάτω παρατίθεται το διάγραμμα αλληλουχίας για τα δεδομένα εξόδου.

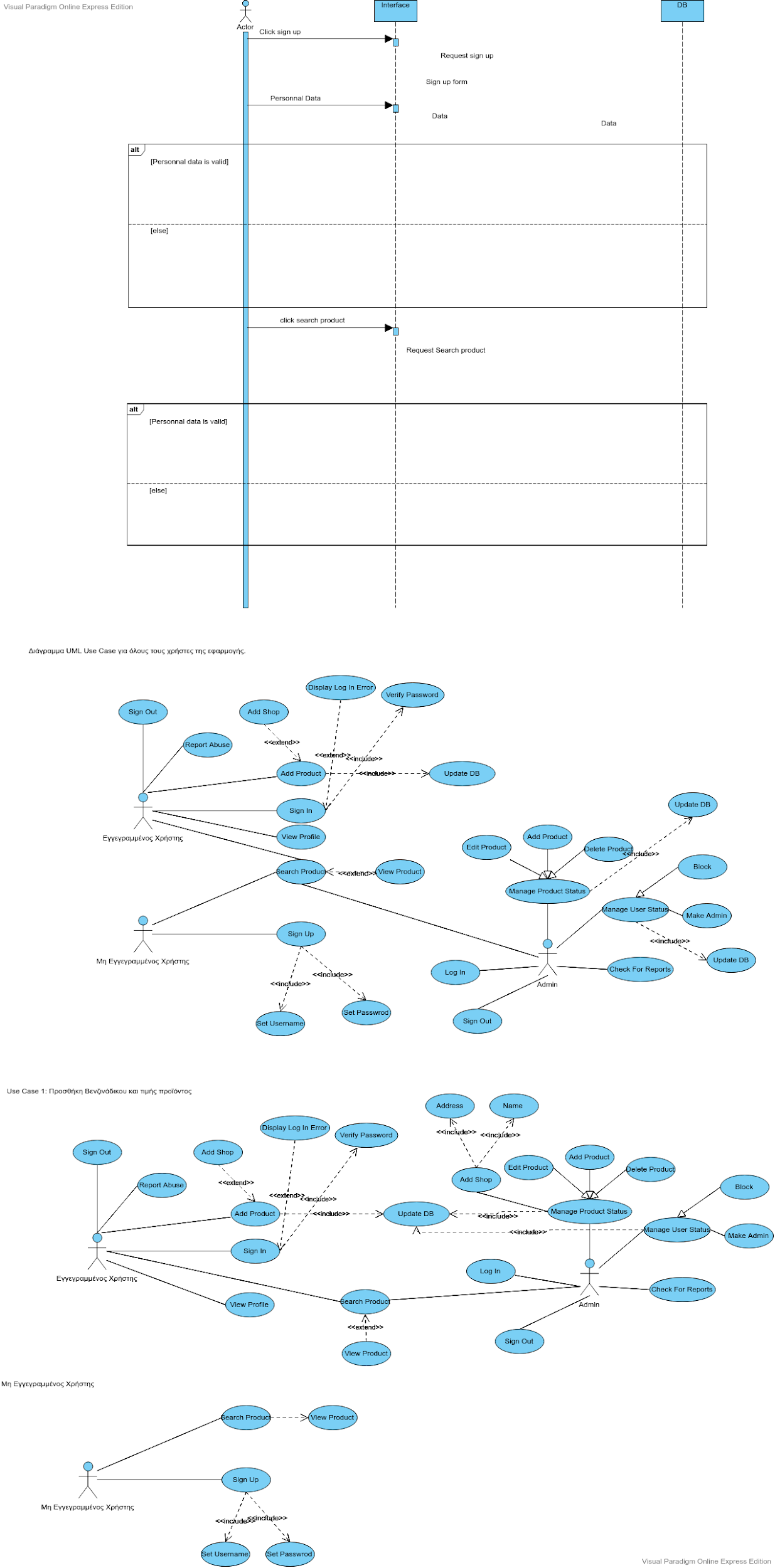
### **3.2.2 Use Case 2: Registration and Search**

#### **3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται**

*Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης*

Η περίπτωση αυτή αφορά τους μη εγγεγραμμένους χρήστες.

Παρακάτω φαίνεται το διάγραμμα για τον μη εγγεγραμμένο χρήστη.



#### **3.2.2.2 Προϋποθέσεις εκτέλεσης**

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

Το σύστημα δεν θέτει συγκεκριμένες προϋποθέσεις για την είσοδο του μη εγγεγραμμένου χρήστη ώστε αυτός είτε να εγγραφεί είτε να αναζητήσει ό,τι χρειάζεται, πέρα από τη σύνδεση στο διαδίκτυο και την επίσκεψη στην ιστοσελίδα μας.

#### **3.2.2.3 Περιβάλλον εκτέλεσης**

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

Το περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης είναι η διαδικτυακή διεπαφή χρήστη. Στην αρχική σελίδα ο χρήστης θα μπορεί να δει την αρχική σελίδα και να αναζητήσει το προϊόν που ψάχνει, καθώς θα έχει την επιλογή να κάνει εγγραφή συμπληρώνοντας τα απαραίτητα πεδία.

#### **3.2.2.4 Δεδομένα εισόδου**

*Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.*

Ένας μη εγγεγραμμένος χρήστης μπορεί αρχικά να κάνει εγγραφή, οπότε δεδομένα εισόδου θα είναι τα προσωπικά στοιχεία του χρήστη. Τα δεδομένα εγγραφής λοιπόν θεωρούνται μη έγκυρα, μόνο στην περίπτωση που το username ή το email, ανήκουν ήδη σε κάποιο εγγεγραμμένο χρήστη και εμφανίζεται σχετικό μήνυμα σφάλματος. Στην περίπτωση που ο χρήστης κάνει αναζήτηση, δεδομένα εισόδου είναι τα κλειδιά αναζήτησης που επιλέγει οπότε δεδομένα εξόδου είναι μία λίστα με τα σχετικά αποτελέσματα. Λόγω σχεδιασμού βάσης, τα δεδομένα που θα επιστρέφονται στο χρήστη ως αποτελέσματα είναι συνεπή σε σχέση με τον όρο που έχει επιλέξει να αναζητήσει.

#### **3.2.2.5 Παράμετροι**

*Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών, εφόσον υπάρχουν παράμετροι.*

Παράμετρος στη συγκεκριμένη περίπτωση αποτελεί η διεύθυνση ηλεκτρονικού ταχυδρομείου του μη εγγεγραμμένου χρήστη και ο κωδικός που θα πληκτρολογήσει. Συνθήκη εγκυρότητας για το e-mail αποτελεί η απαίτηση να έχει την μορφή <username>@<domain>.

#### **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).*

Για την εγγραφή:

Βήμα 1: Ο χρήστης ζητά άδεια για να κάνει εγγραφή

Βήμα 2: Το σύστημα επιστρέφει την φόρμα εγγραφής

Βήμα 3: Ο χρήστης συμπληρώνει τα στοιχεία του τα οποία μεταφέρονται στη βάση δεδομένων

1η Περίπτωση: Τα δεδομένα εγγραφής είναι έγκυρα, δηλαδή δεν χρησιμοποιούνται ήδη από εγγεγραμμένο χρήστη. → Γίνεται επιβεβαίωση εγγραφής.

2η Περίπτωση: Τα δεδομένα δεν είναι έγκυρα. → Αποστέλλεται μήνυμα λάθους.

Για την αναζήτηση:

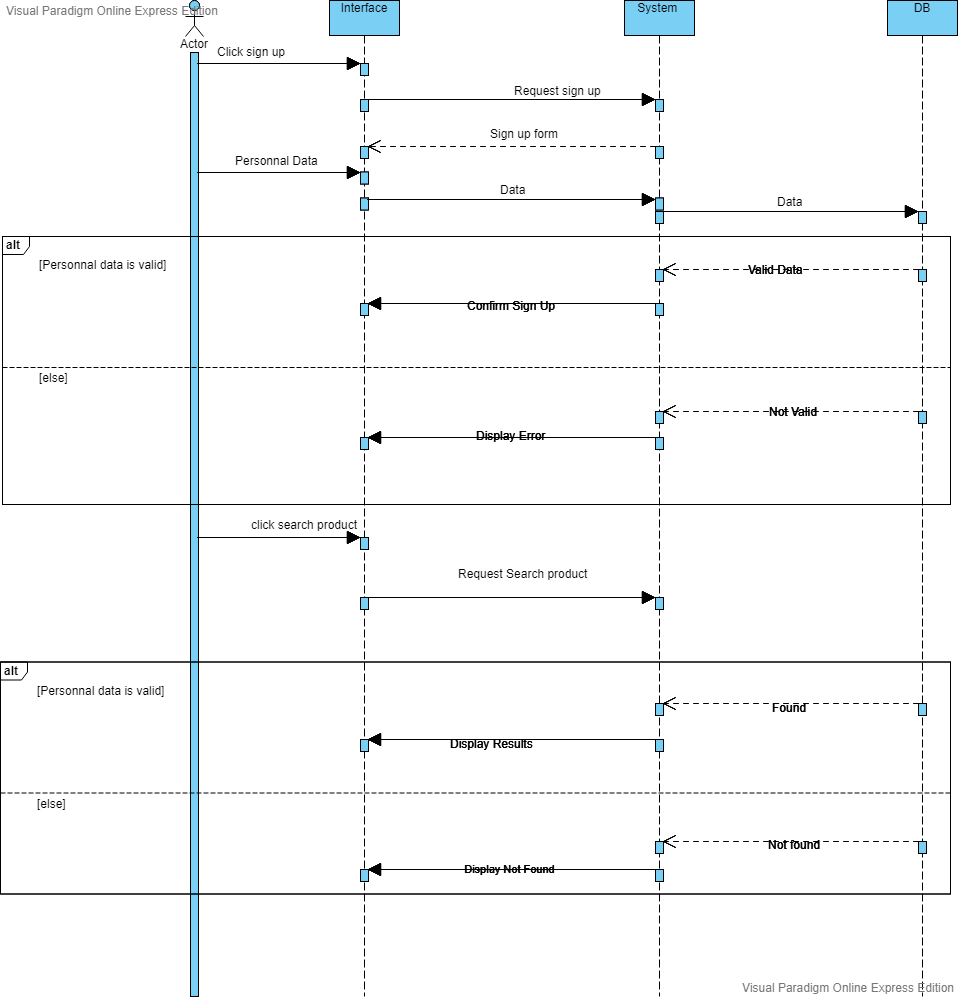
Βήμα 1: Ο χρήστης ζητά άδεια για να κάνει αναζήτηση

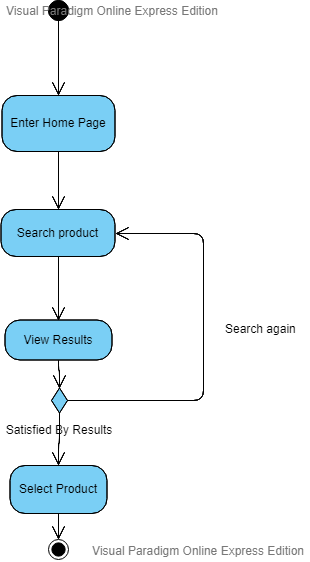
Βήμα 2: Το σύστημα επιστρέφει την φόρμα αναζήτησης (home page)

Βήμα 3: Ο χρήστης πληκτρολογεί το κλειδί αναζήτησης

1η Περίπτωση: Τα δεδομένα υπάρχουν στη βάση . → Επιστρέφεται η λίστα με τα αποτελέσματα της αναζήτησης.

2η Περίπτωση: Τα δεδομένα δεν υπάρχουν στη βάση. → Επιστρέφεται αντίστοιχο μήνυμα.

**

**

#### **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)*

Στην περίπτωση αυτή δεδομένα εξόδου είναι τα προσωπικά δεδομένα με τα οποία ο χρήστης δημιουργεί καινούριο λογαριασμό (sign up). Αυτά είναι ο κωδικός, το e-mail και το user

#### **3.2.2.8 Παρατηρήσεις**

*Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει*

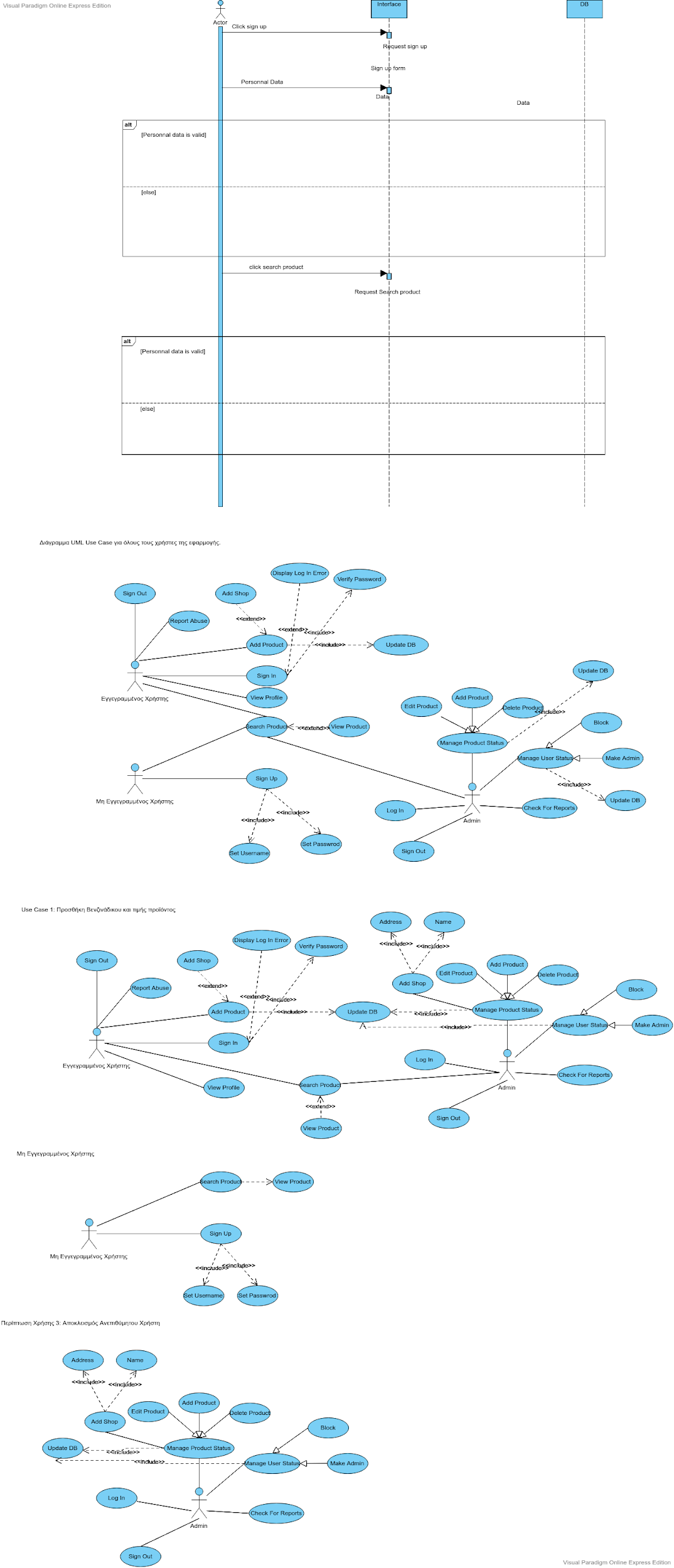
*. . .*

### **3.2.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Block unwanted user**

#### **3.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται**

Οι χρήστες που εμπλέκονται είναι o admin καθώς είναι ο μόνος που μπορεί να blockάρει κάποιον εγγεγραμμένο χρήστη.

Παρακάτω φαίνεται το διάγραμμα use case για την περίπτωση χρήσης 3.



**3.2.3.2 Προϋποθέσεις εκτέλεσης**

Για να μπορεί να εκτελεστεί η συγκεκριμένη περίπτωση χρήσης θα πρέπει ο χρήστης να κάνει log in για να μπει στο περιβάλλον του admin.

#### **3.2.3.3 Περιβάλλον εκτέλεσης**

Το περιβάλλον στο οποίο εκτελείται το block κάποιου εγγεγραμμένου χρήστη είναι μια σελίδα όπου εμφανίζονται όλοι οι εγγεγραμμένοι χρήστες και ο admin μπορεί να ελέγξει τα στοιχεία τους καθώς και την ισχύουσα κατάσταση τους(ενεργός, banned, blocked κλπ).

#### **3.2.3.4 Δεδομένα εισόδου**

Ο admin στην λίστα με τους χρήστες θα μπορεί να κάνει search ώστε να βρει τον χρήστη που θέλει και στη συνέχεια από ένα menu που θα υπάρχει δίπλα στο όνομα του χρήστη να αλλάξει την κατάσταση του, για παράδειγμα από “Ενεργός” σε “Blocked”. Δεδομένα εισόδου στην συγκεκριμένη περίπτωση θα είναι επίσης ο κωδικός του admin, διότι θα του ζητείται από το σύστημα ώστε να επικυρώσει την ενέργεια του σε σχέση με την αλλαγή κατάστασης ενός χρήστη.

#### **3.2.3.5 Παράμετροι**

*Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών, εφόσον υπάρχουν παράμετροι.*

#### **3.2.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά**

Βημα 1: Ο admin κάνει log in με τα προσωπικά του στοιχεία.

1. Τα δεδομένα που πληκτρολόγησε είναι έγκυρα, οπότε το σύστημα μεταφέρει τον χρήστη στο περιβάλλον του admin.
2. Τα δεδομένα που πληκτρολόγησε δεν είναι έγκυρα, οπότε το σύστημα ενημερώνει τον χρήστη με αντίστοιχο μήνυμα και του ζητάει να πληκτρολογήσει ξανά τα στοιχεία του.

Βήμα 2: Ο admin κάνει κλικ στο κουμπί User List και μεταφέρεται στην σελίδα όπου εμφανίζονται όλοι οι εγγεγραμμένοι χρήστες. Οι επιλογές του admin είναι εξής:

1. Να κάνει κλικ στο πεδίο “Search User” και να πληκτρολογήσει το όνομα χρήστη του οποίου θέλει να ελέγξει την δραστηριότητα ή/και τα στοιχεία του και να αλλάξει πιθανώς την κατάσταση του. Εδώ φυσικά υπάρχουν δύο περιπτώσεις:

1. Ο συγκεκριμένος χρήστης να υπάρχει στην λίστα με τους εγγεγραμμένους.
2. Ο συγκεκριμένος χρήστης να μην υπάρχει οπότε και να εμφανίζεται “No result” και ο admin να πρέπει να επαναλάβει την εν λόγω διαδικασία

2. Αν ο admin επιλέξει τον συγκεκριμένο χρήστη που έψαχνε κάνει κλικ στο όνομά του οπότε και εμφανίζονται τα στοιχεία του χρήστη, η δραστηριότητα του και η κατάσταση του.

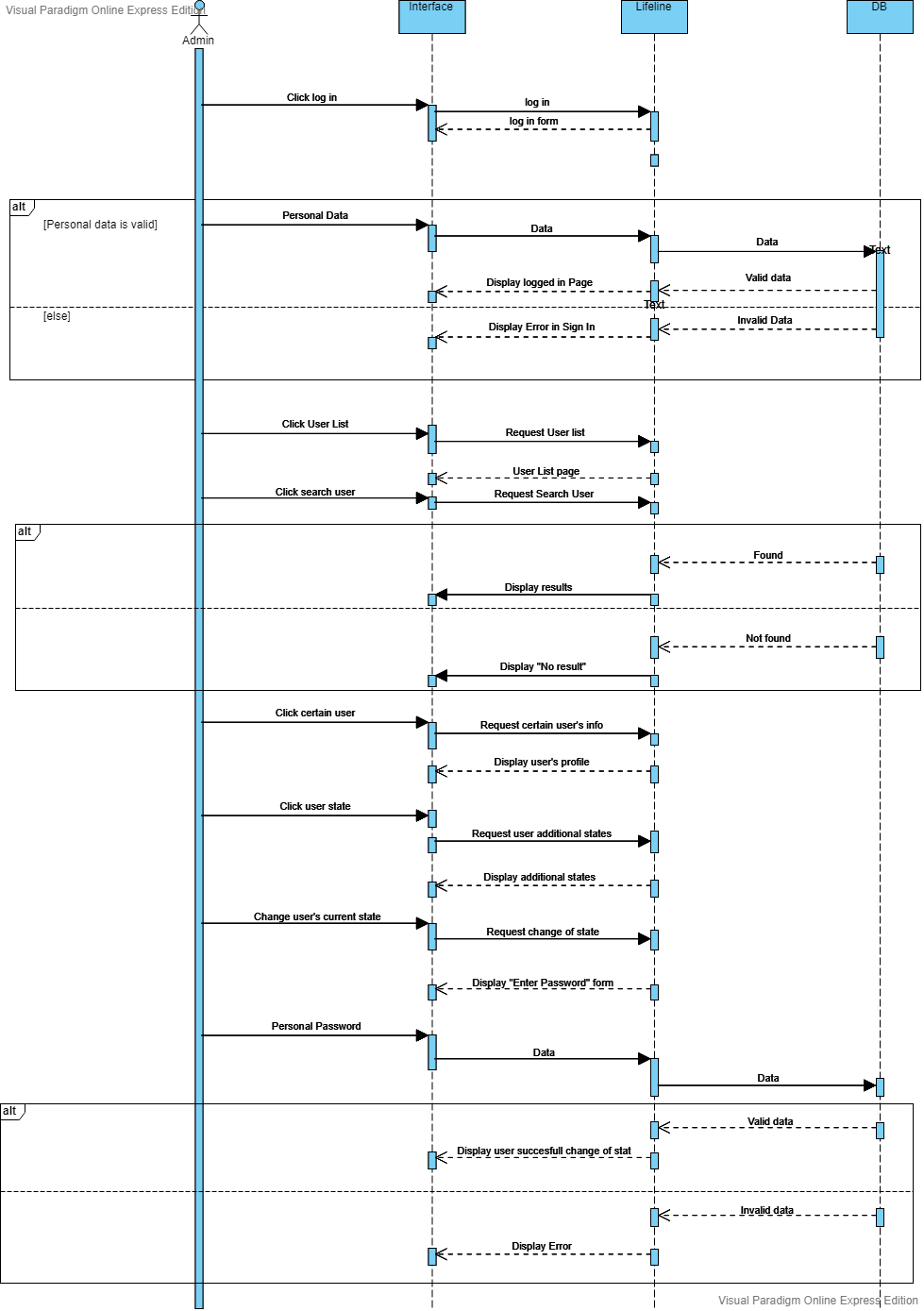
3. Ο admin κάνει κλικ στην υπάρχουσα κατάσταση του χρήστη και εμφανίζεται η υπολίστα με τις διαθέσιμες καταστάσεις που μπορεί να βάλει τον χρήστη.

4. Ο admin επιλέγει τι θέλει να κάνει με τον συγκεκριμένο χρήστη και ακολούθως διαλέγει αν θα τον κάνει permanent ban, block για καταχώρηση πρατηρίου κλπ. Εν συνεχεία κλικάρει την επιθυμητή κατάσταση.

5. Ο admin κλικάρει “Done” και το σύστημα εμφανίζει ένα πεδίο ώστε ο admin να υποβάλει τον κωδικό του ώστε να αποφευχθεί τυχόν υποκλοπή token από τρίτους και επιχείρηση αλλοίωσης του συστήματος.

6. Ο admin πληκτρολογεί τον κωδικό του στο πεδίο και πατάει “Submit”. Πλέον οι περιπτώσεις είναι οι εξής:

1. Ο κωδικός είναι έγκυρος και η κατάσταση του εγγεγραμμένου χρήστη έχει αλλάξει επιτυχώς.
2. Ο κωδικός είναι εσφαλμένος και ο admin πρέπει να επαναλάβει την διαδικασία.



#### 

#### 

#### **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)*

Σε αυτή την περίπτωση το μόνο δεδομένο εξόδου που έχουμε είναι η αλλαγή κατάστασης του χρήστη σε Blocked.

## **3.3 Απαιτήσεις επιδόσεων**

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

→ Εφόσον πρόκειται για μία εφαρμογή crowdsourcing πρέπει το λογισμικό μας να είναι σε θέση να εξυπηρετήσει όλες τις ταυτόχρονες αιτήσεις για είσοδο, αναζήτηση, ενημέρωση της βάσης κλπ. Πρέπει να εξασφαλισθεί η συνεπής παρουσίαση των αποτελεσμάτων.

→ Οι ενέργειες του χρήστη πρέπει να επιτελούνται σύντομα, δηλαδή το σύστημά μας να είναι άμεσα αποκρίσιμο.

→ Κριτήρια επιθυμητών επιδόσεων για το λογισμικό μας μπορούν να θεωρηθούν ο αριθμός χρηστών που επισκέπτονται τη σελίδα μας, ο χρόνος παραμονής αυτών, ο αριθμός των καινούριων καταχωρήσεων, η συνολική μέση βαθμολογία των καταχωρήσεων, ο αριθμός report κακής χρήσης κλπ.

## **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).*

### **3.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα**

*Απαιτήσεις πρόσβασης και περιορισμοί.*

→ Μη εγγεγραμμένος χρήστης: Δεν απαιτείται κανένα στοιχείο ταυτοποίησης και ο χρήστης έχει πρόσβαση στην αρχική μας σελίδα όπου μπορεί να κάνει εγγραφή ή να αναζητήσει καύσιμα και βενζινάδικα.

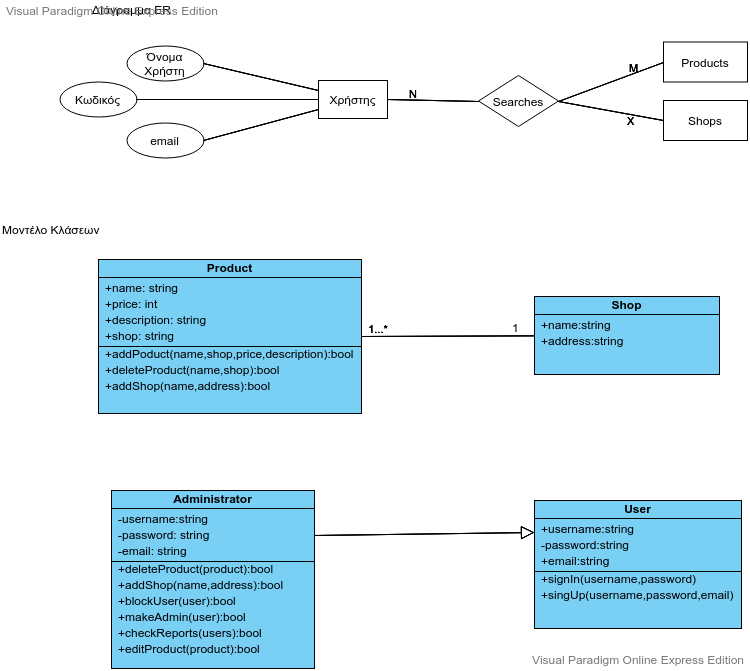
→ Εγγεγραμμένος χρήστης: Συμπληρώνοντας τα στοιχεία του, ο χρήστης μπαίνει στο σύστημα, και έχει τη δυνατότητα πέρα από το να κάνει αναζήτηση, να καταχωρήσει δεδομένα που θα προστεθούν στη βάση μας, να βαθμολογήσει καταχωρήσεις, να δει προφίλ άλλων χρηστών κλπ.

→ Administrator: Ο administrator έχει τις ίδιες απαιτήσεις πρόσβασης με τον εγγεγραμμένο χρήστη. Μέσα στο σύστημα μπορεί να εκτελέσει τις ίδιες δραστηριότητες με τον εγγεγραμμένο χρήστη, μόνο που επιπλέον μπορεί και να επεξεργαστεί τη λίστα των χρηστών και να μπλοκάρει προφίλ τα οποία έχουν δεχθεί ανάλογες αναφορές.

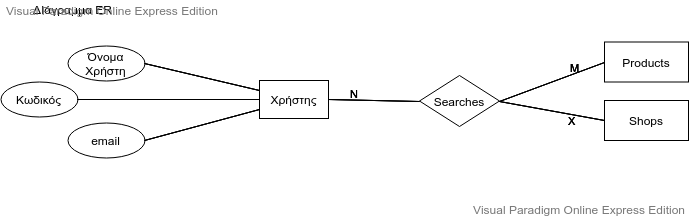
### **3.4.3 Μοντέλο δεδομένων (μοντέλο κλάσεων UML ή/και μοντέλο ER)**

*Μοντέλα δεδομένων UML ή/και ER*

Μοντέλο κλάσεων:



Μοντέλο ΕR:



### **3.4.4 Προδιαγραφές ακεραιότητας δεδομένων**

*Κανόνες ακεραιότητας και εγκυρότητας δεδομένων*

Οποιαδήποτε δεδομένα εισάγει ο χρήστης θεωρούνται έγκυρα. Αυτό ισχύει μέχρις ότου ο administrator ανακαλύψει δεδομένα τα οποία δεν συνάδουν με την πραγματικότητα. Εάν συμβεί αυτό ο administrator έχει την δυνατότητα να τα διορθώσει ή να τα διαγράψει.

### **3.4.5 Προδιαγραφές διατήρησης δεδομένων**

*Απαιτήσεις διατήρησης δεδομένων σε βάθος χρόνου.*

## **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*

## **3.6 Λοιπές απαιτήσεις**

### **3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού**

*Τεκμηρίωση απαιτήσεων διαθεσιμότητας*

### **3.6.2 Απαιτήσεις ασφάλειας**

*Τεκμηρίωση απαιτήσεων ασφαλείας*

### **3.6.3 Απαιτήσεις συντήρησης**

*Τεκμηρίωση απαιτήσεων συντήρησης*

# **4.0 Παράρτημα**

## **4.1 Παραδοχές και εξαρτήσεις**

## **4.2 Ακρωνύμια και συντομογραφίες**

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