

# Έγγραφο απαιτήσεων λογισμικού (SRS)

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

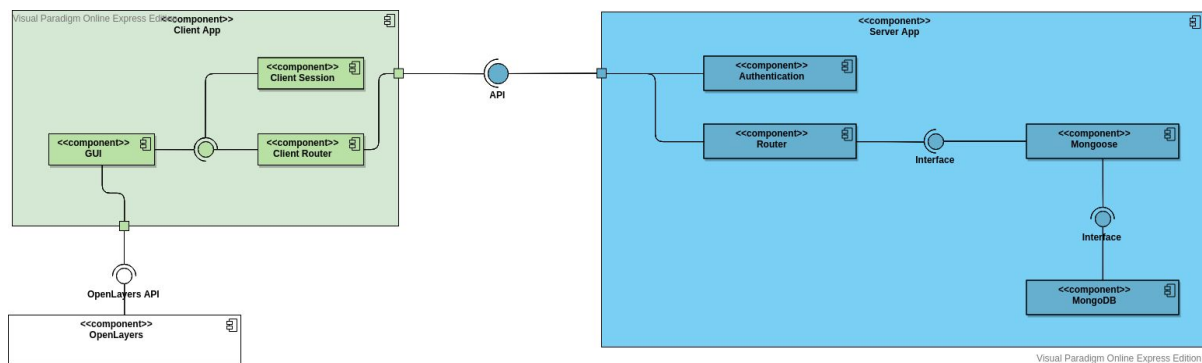
Definitely Not Skrutz (DNS)

## 1. Εισαγωγή

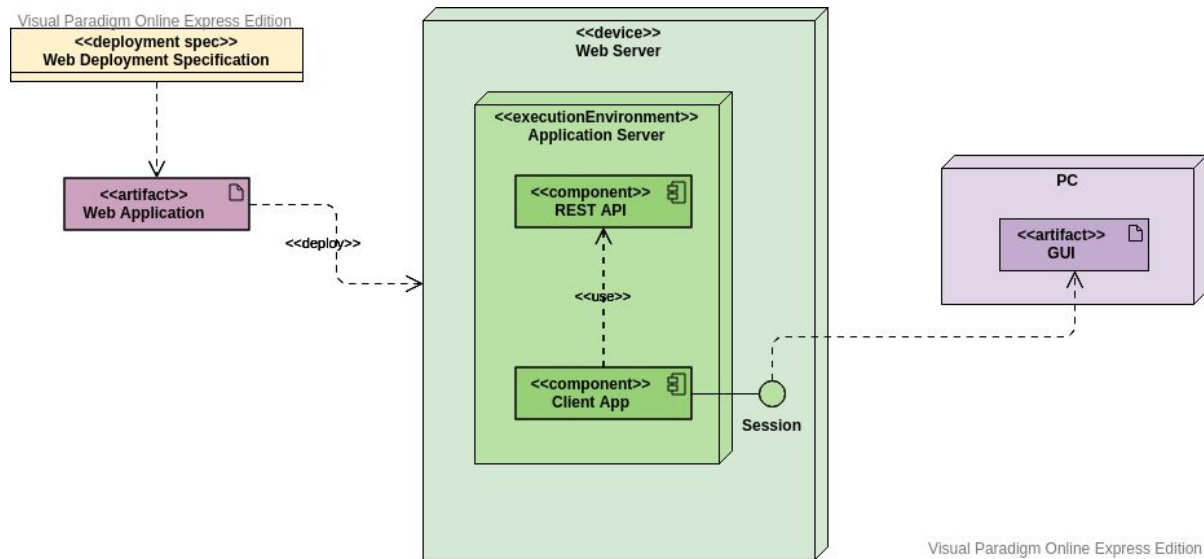
### 1.1 Εισαγωγή: σκοπός του λογισμικού

Το σύστημα αποτελεί ένα ηλεκτρονικό παρατηρητήριο τιμών για ένα εύρος προϊόντων, που θα βασίζεται στη μέθοδο του πληθοπορισμού (crowdsourcing). Σκοπός του είναι η ενημέρωση των καταναλωτών για τις τιμές των προϊόντων, καθώς και η δυνατότητα σύγκρισης των προϊόντων.

### 1.2 Επισκόπηση του λογισμικού



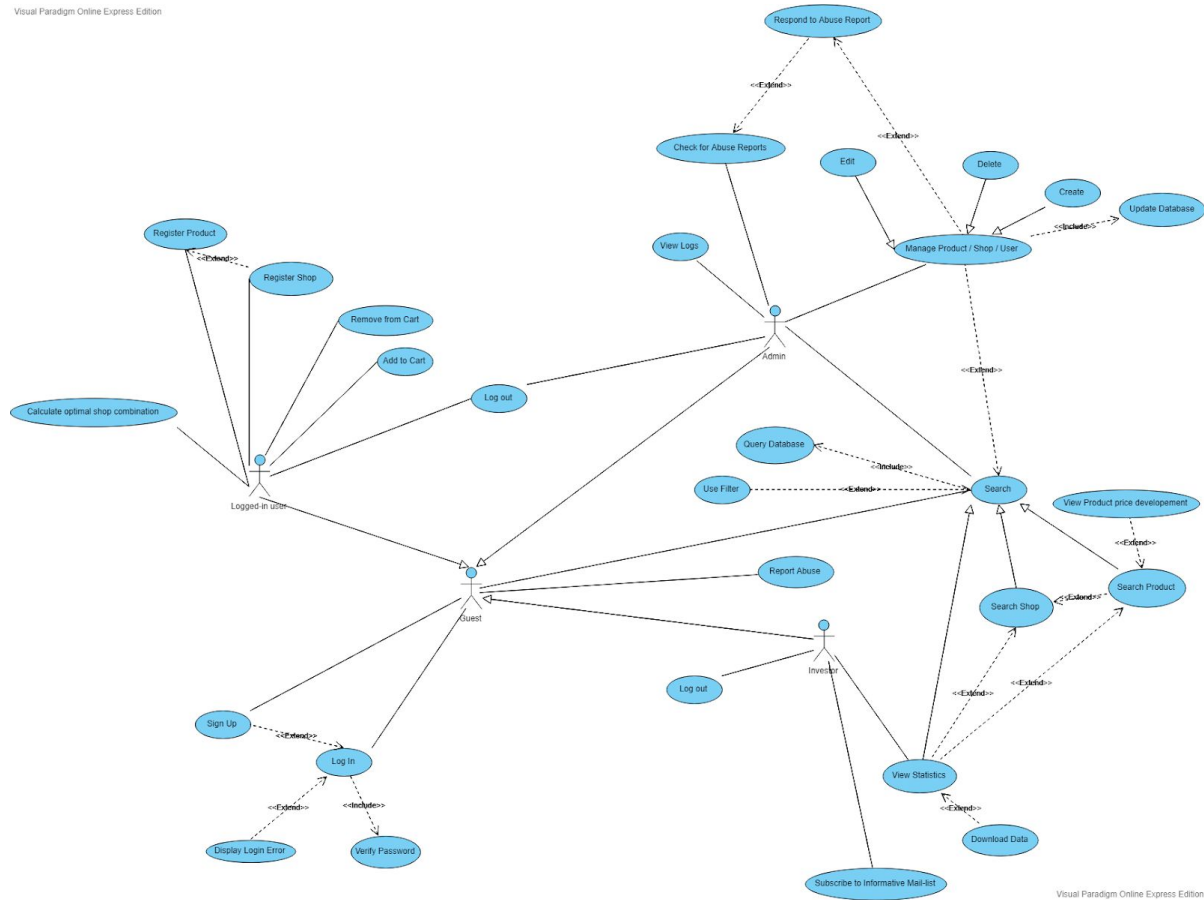
### 1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού



Ο Web Server επικοινωνεί με το PC με μηνύματα HTTPS.

### 1.3.2 Διεπαφές με το χρήστη

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



## 2. Αναφορές - πηγές πληροφοριών

## 3. Προδιαγραφές απαιτήσεων λογισμικού

### 3.1 Εξωτερικές διεπαφές

Το client app χρησιμοποιεί το OpenLayers API για την παρουσίαση γεωγραφικών δεδομένων.

## 3.2 Λειτουργίες: περιπτώσεις χρήσης

### 3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: (User Use Case)



#### 3.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Guest: Επισκέπτης στην ιστοσελίδα, για τον οποίο δεν απαιτείται κάποια επιβεβαίωση ταυτότητας

Logged-in User: Χρήστης για τον οποίο έχει γίνει επιβεβαίωση ταυτότητας

#### 3.2.1.2 Προϋποθέσεις εκτέλεσης

Μοναδική προϋπόθεση είναι ο χρήστης να βρίσκεται στο περιβάλλον της διεπαφής.

#### 3.2.1.3 Περιβάλλον εκτέλεσης

Το συγκεκριμένο σενάριο χρήσης εκτελείται μόνο στη διαδικτυακή διεπαφή χρήστη.

#### 3.2.1.4 Δεδομένα εισόδου

Σε περίπτωση που ο χρήστης επιθυμεί να εγγραφεί στο σύστημα, θα πρέπει να αποθηκεύονται με ασφάλεια τα προσωπικά του στοιχεία (username, password, email).

Επίσης, όταν κάνει log in, τα δεδομένα εισόδου του (username, password) θα μεταβιβάζονται στο σύστημα. Το συγκεκριμένο ζεύγος username, password θα πρέπει να είναι ορθά ώστε να τα δεχθεί το σύστημα.

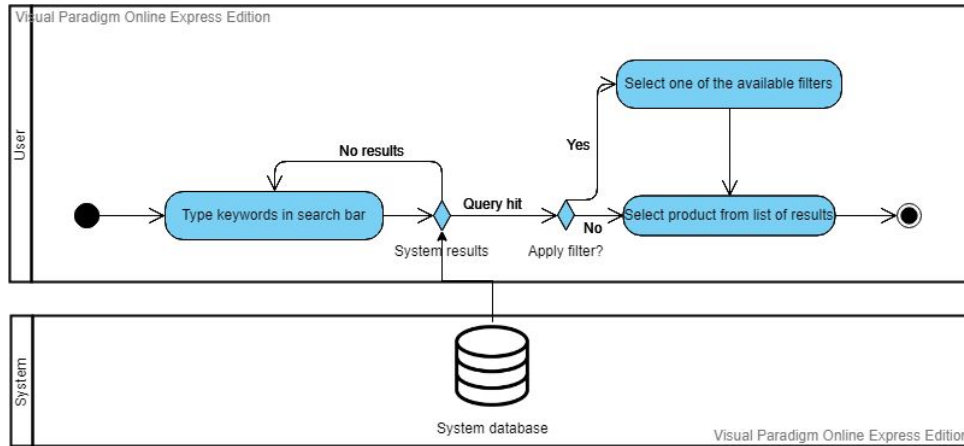
Ως δεδομένα εξόδου θα έχουμε όλα τα δεδομένα των καταστημάτων και των προϊόντων που μπορεί να αναζητήσει ο χρήστης, ενώ όταν ο χρήστης εκτελεί μία αναζήτηση, καταχωρεί ως δεδομένα εισόδου τα keywords που επιθυμεί να αναζητήσει, τα οποία πρέπει να υπάρχουν στη βάση ώστε να θεωρούνται έγκυρα.

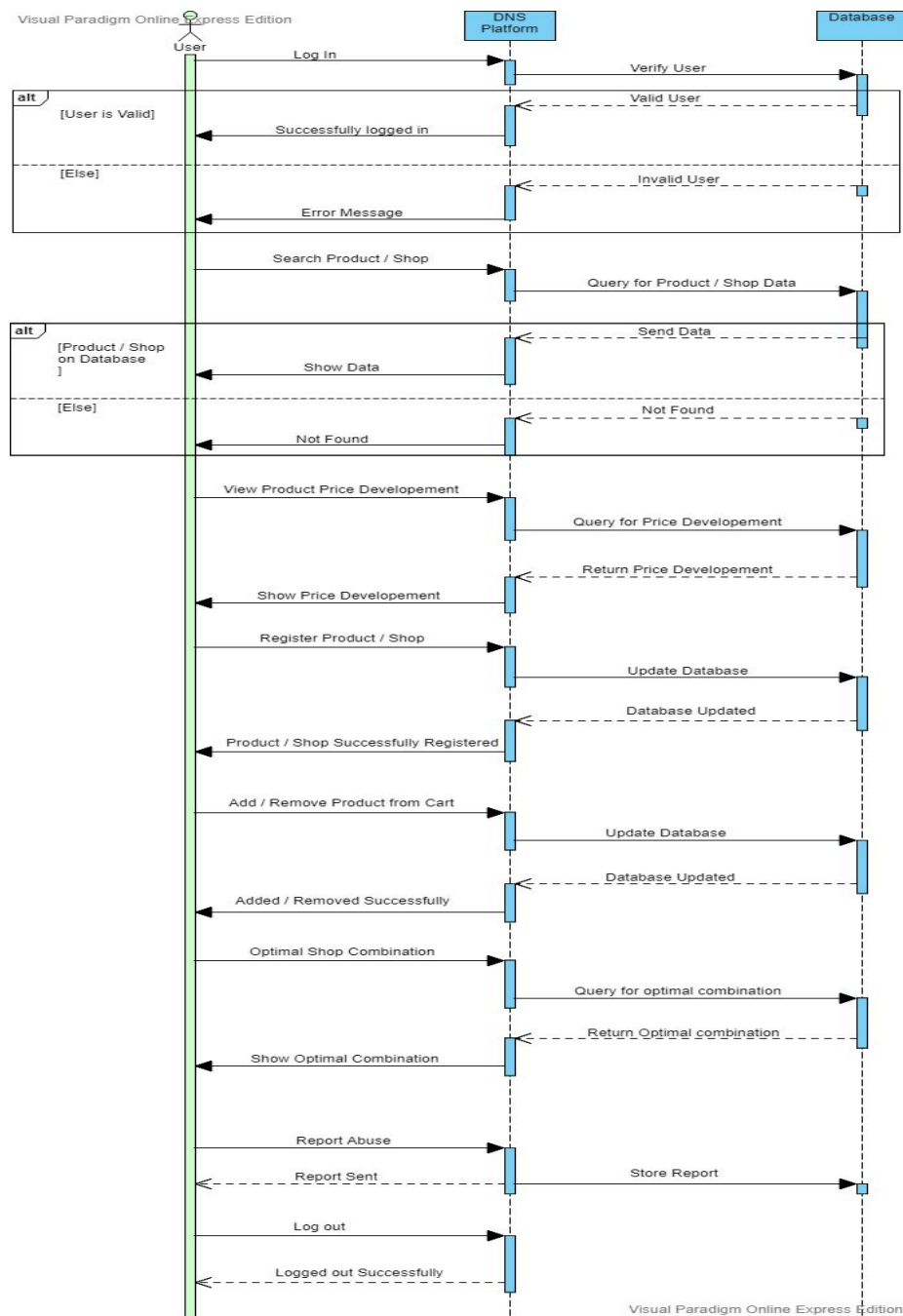
#### 3.2.1.5 Παράμετροι

Θεωρούμε πως ο χρήστης μπορεί να είναι είτε εγγεγραμμένος στο σύστημα, είτε επισκέπτης. Κατά την περίπτωση που ένας χρήστης θέλει να εγγραφεί, θα πρέπει τα δεδομένα που θα δώσει στα ανάλογα πεδία εισόδου να ικανοποιούν συγκεκριμένες συνθήκες. Το email που θα δώσει θα πρέπει να έχει μορφή <username>@<domain> ώστε να θεωρείται έγκυρο, ενώ και ο κωδικός που θα παρέχει θα πρέπει να είναι μεγαλύτερος από ένα όριο μεγέθους ώστε να γίνεται αποδεκτός. Ακόμα, κατά την είσοδο ενός προηγουμένως εγγεγραμμένου χρήστη, είναι προφανής απαίτηση το username και ο κωδικός πρόσβασης να ταυτίζονται με το αντίστοιχο ζεύγος που θα υπάρχει αποθηκευμένο στη βάση.

#### 3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Τα βήματα τα οποία ακολουθούνται σε κάθε περίπτωση είναι παρόμοια. Αρχικά, ο χρήστης επικοινωνεί με την διεπαφή μεταβιβάζοντάς της ένα μήνυμα. Στη συνέχεια, αυτή έρχεται σε επαφή με την βάση, μεταφέροντας το αίτημα του χρήστη με το αντίστοιχο query στη βάση. Η βάση με τη σειρά της επεξεργάζεται το αίτημα και επιστρέφει τα αντίστοιχα δεδομένα στην πλατφόρμα, η οποία ενημερώνει αντίστοιχα τον χρήστη. Σε περίπτωση που το αίτημα προς την βάση δεν ικανοποιηθεί, βρισκόμαστε σε εναλλακτικό σενάριο, στο οποίο επιστρέφεται στον χρήστη ανάλογο μήνυμα σφάλματος. Τέλος, έχουμε μία περίπτωση στην οποία ο χρήστης επιθυμεί να κάνει αναφορά κακής χρήσης, όπου και στέλνει το αίτημά του στην πλατφόρμα, εκείνη το μεταφέρει στη βάση και στη συνέχεια στέλνει επιβεβαίωση αποστολής στον χρήστη, χωρίς να περιμένει απάντηση από τη βάση, καθώς δεν απαιτείται στη συγκεκριμένη περίπτωση.

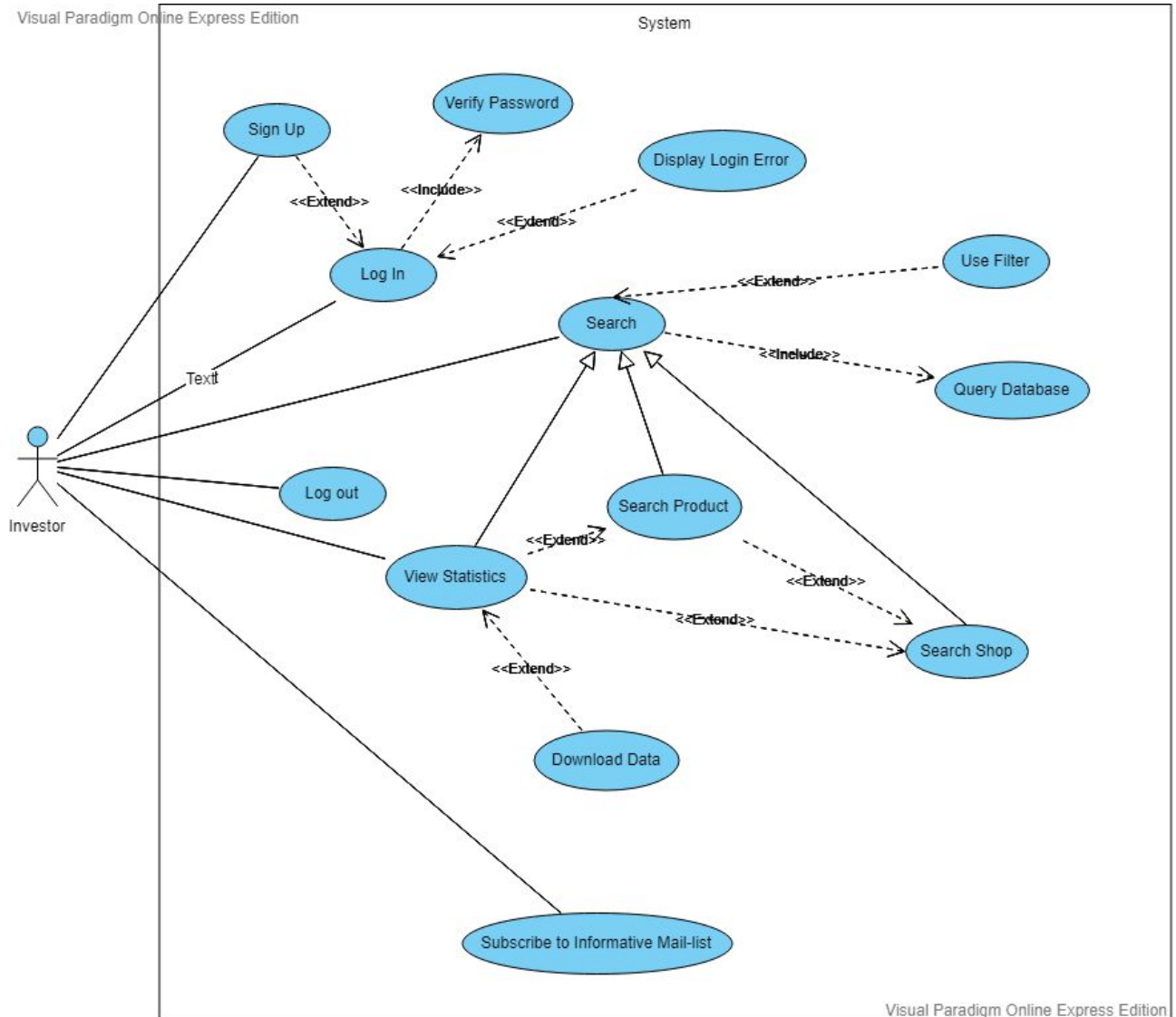




### 3.2.1.7 Δεδομένα εξόδου

Δεδομένα εξόδου στην περίπτωση του χρήστη έχουμε κατά την δημιουργία καλαθιού αγορών (shopping cart), κατά τη διαδικασία δημιουργίας λογαριασμού (sign up) αλλά και κατά την αναφορά κάποιας κακής χρήσης (report abuse).

### 3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: (Investor Use Case)



### 3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Σε αυτήν την περίπτωση μοναδικός εμπλεκόμενος είναι ο επενδυτής (investor).

### 3.2.2.2 Προϋποθέσεις εκτέλεσης

Μοναδική προϋπόθεση είναι ο επενδυτής να βρίσκεται στο περιβάλλον της διεπαφής και να έχει κάνει log in με τα αντίστοιχα credentials τα οποία τον αναγνωρίζουν ως επενδυτή

### 3.2.2.3 Περιβάλλον εκτέλεσης

Η περίπτωση χρήσης εκτελείται αποκλειστικά στην διαδικτυακή διεπαφή χρήστη.

### 3.2.2.4 Δεδομένα εισόδου

Το σύστημα δέχεται ως δεδομένα εισόδου τα credentials του επενδυτή ώστε να εισέλθει στον λογαριασμό του. Ακόμα, δέχεται όρους τους οποίους επιθυμεί να αναζητήσει αυτός στην πλατφόρμα, οι οποίοι μπορούν να αφορούν καταστήματα ή προϊόντα.

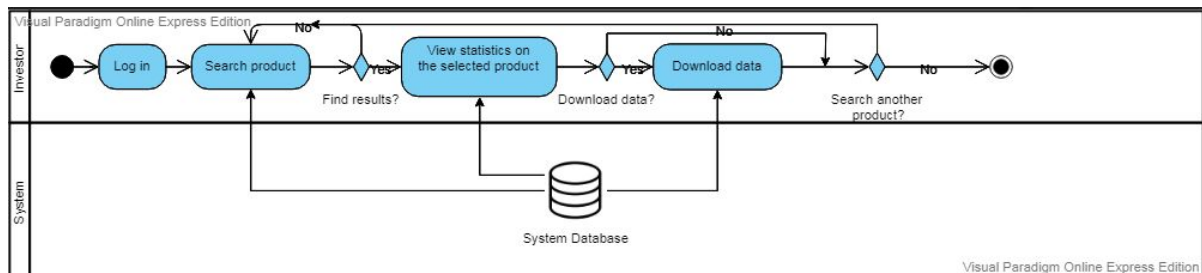


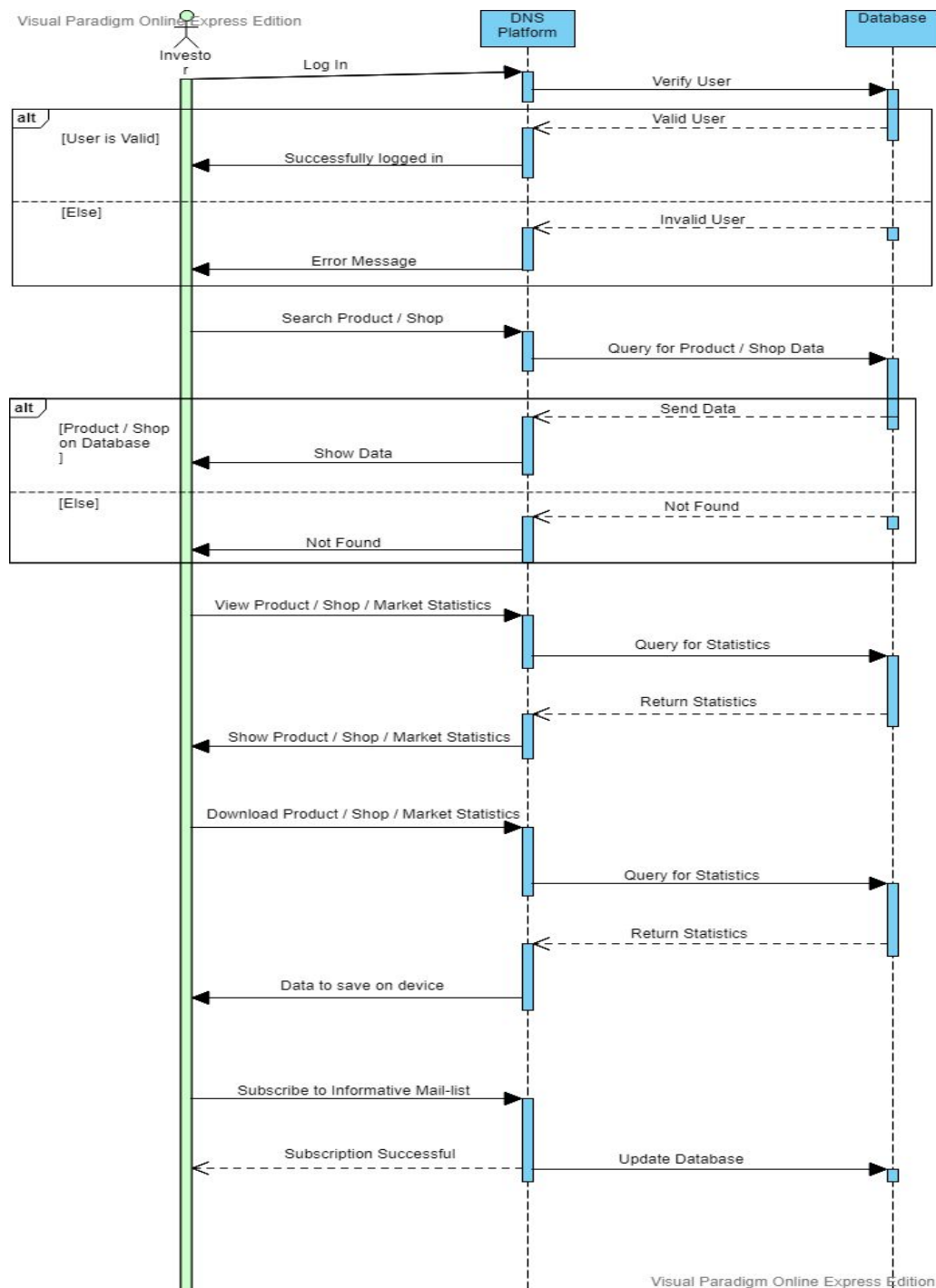
### 3.2.2.5 Παράμετροι

Κατά την είσοδο του επενδυτή στην πλατφόρμα, θα πρέπει το username και ο κωδικός πρόσβασης να ταυτίζονται με το αντίστοιχο ζεύγος που θα υπάρχει αποθηκευμένο στη βάση, το οποίο και θα υποδεικνύει ότι ο εν λόγω χρήστης είναι επενδυτής.

### 3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Όπως και στο σενάριο του χρήστη, έτσι και εδώ οι περισσότερες ενέργειες του επενδυτή αποτελούνται από 4 βήματα, όπως φαίνεται και από το παρακάτω sequence diagram. Αρχικά ο επενδυτής μεταβιβάζει ένα μήνυμα στην πλατφόρμα, η οποία κάνει το αντίστοιχο ερώτημα στη βάση. Η βάση στη συνέχεια δίνει την αντίστοιχη απάντηση στην πλατφόρμα και αυτή ενημερώνει κατάλληλα τον επενδυτή.



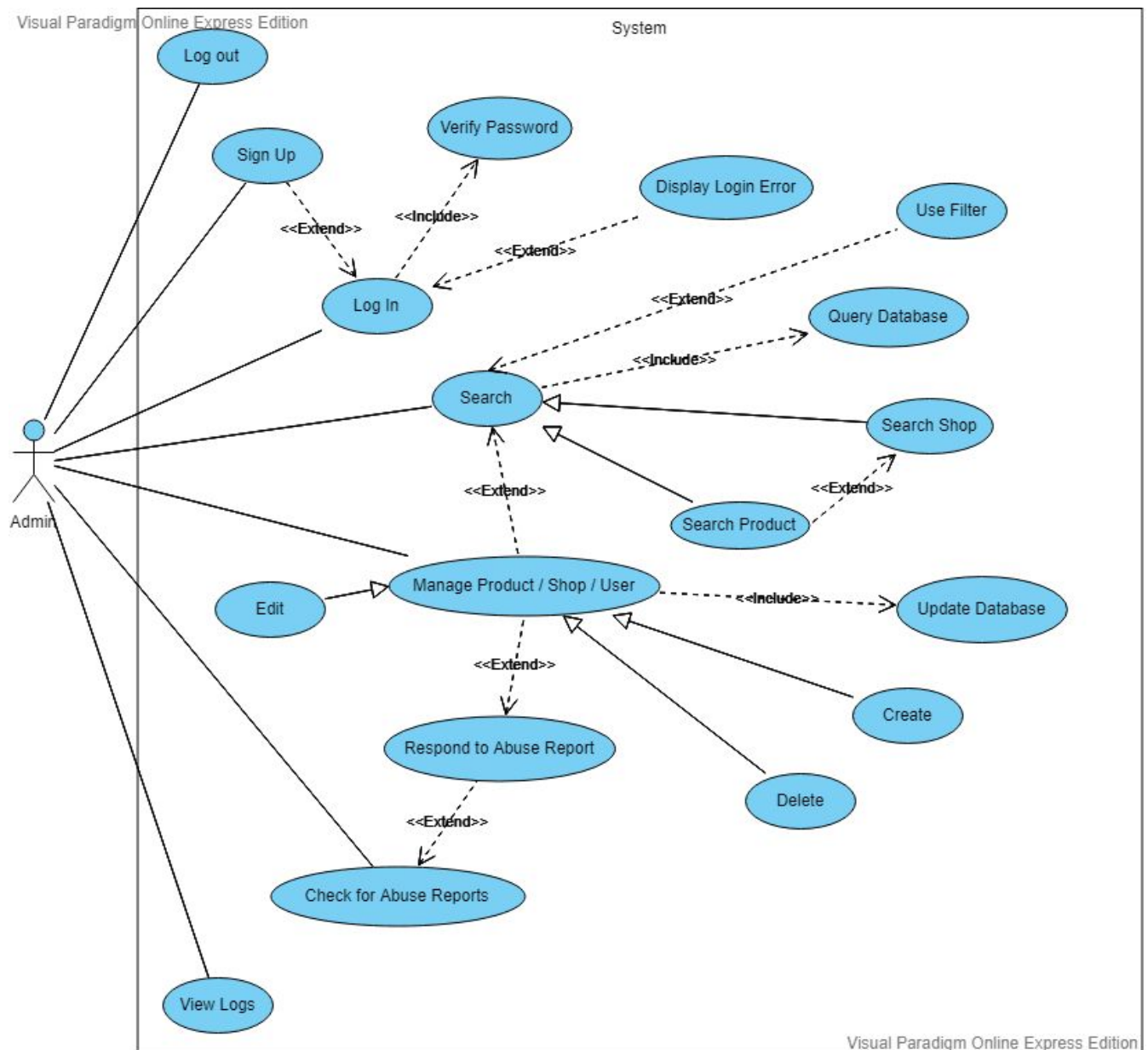


### 3.2.2.7 Δεδομένα εξόδου

Το σύστημα θα επιστρέφει στον επενδυτή τα στατιστικά τα οποία εκείνος θα ζητάει με τις αναζητήσεις του (και να ανάλογα queries του συστήματος στη βάση), τα οποία θα μπορεί είτε να τα προβάλει είτε να τα κατεβάσει.

...

### 3.2.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: (Administrator Use Case)



#### 3.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται

Σε αυτήν την περίπτωση εμπλέκεται ο διαχειριστής (administrator).

#### 3.2.3.2 Προϋποθέσεις εκτέλεσης

Μοναδική προϋπόθεση είναι ο διαχειριστής να βρίσκεται στο περιβάλλον της διεπαφής και να έχει κάνει log in με τα αντίστοιχα credentials τα οποία τον αναγνωρίζουν ως διαχειριστή.

#### 3.2.3.3 Περιβάλλον εκτέλεσης

Το σενάριο εκτελείται στην διαδικτυακή διεπαφή χρήστη.

#### 3.2.3.4 Δεδομένα εισόδου

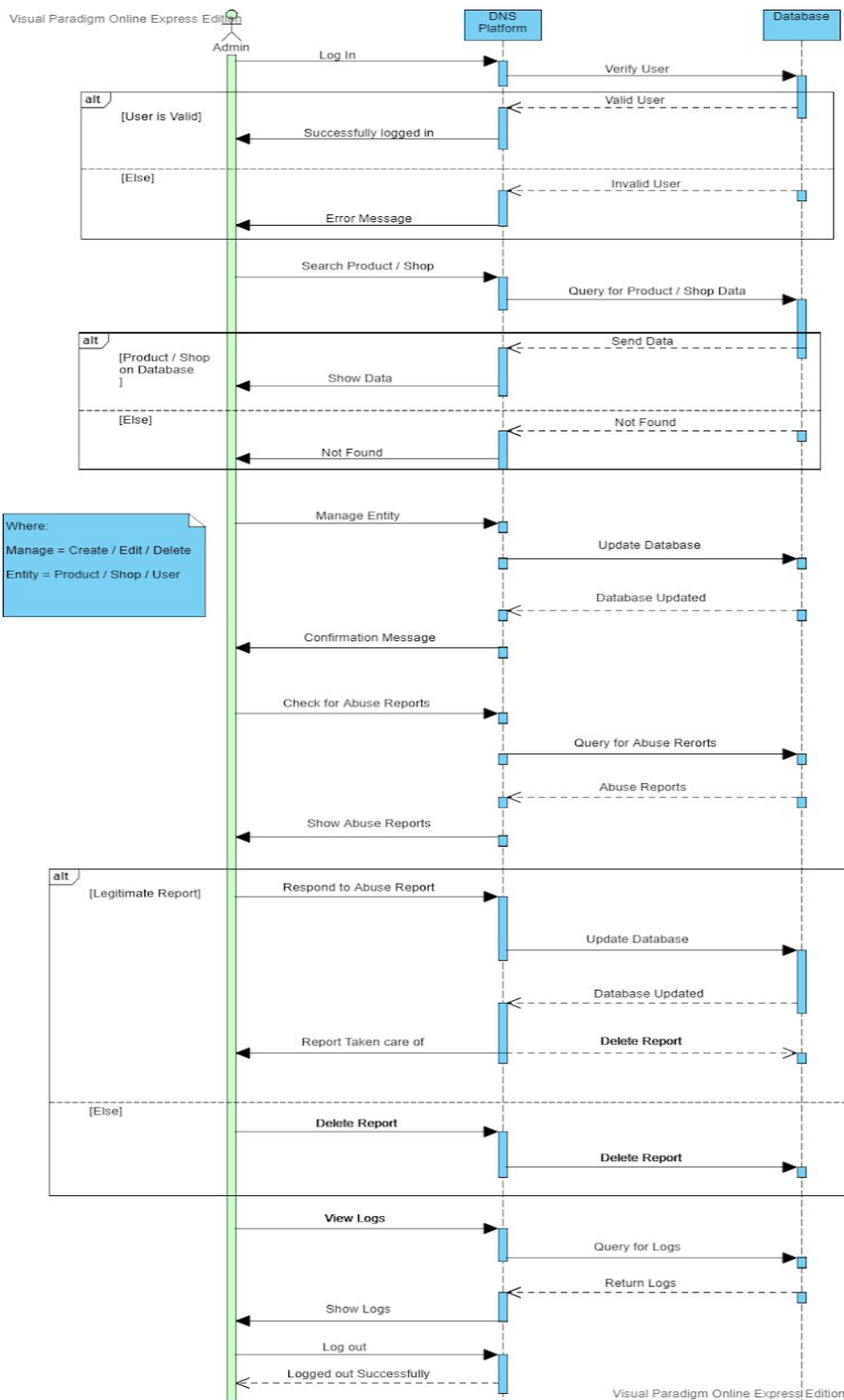
Δεδομένα εισόδου, πέρα από τα στοιχεία που καταχωρεί ο διαχειριστής για να εισέλθει στο σύστημα, τα οποία αποθηκεύονται όπως και στις προηγούμενες περιπτώσεις χρήσης, ο διαχειριστής μπορεί να εισάγει δεδομένα στο σύστημα κατά την διαχείριση των καταστημάτων/ προϊόντων (προσθήκη νέου ή διαχείριση υπάρχοντος), αλλά και κατά τη διαχείριση αναφορών κακής χρήσης.

#### 3.2.3.5 Παράμετροι

Βασική παράμετρος είναι ξανά το ζεύγος username-password του διαχειριστή, το οποίο απαιτείται να είναι έγκυρο ώστε να μπορεί αυτός να εισέλθει στην πλατφόρμα.

#### 3.2.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Τα βήματα που ακολουθούνται για κάθε μία από τις ενέργειες του διαχειριστή είναι αντίστοιχα με αυτά που περιγράφονται στις προηγούμενες περιπτώσεις χρήσης. Αναλυτικότερα, μπορεί να γίνει εγγραφή ή είσοδος στο σύστημα, όπου γίνεται ανάλογος χειρισμός λανθασμένης εισαγωγής στοιχείων, όπως φαίνεται στο διάγραμμα αλληλουχίας στα εναλλακτικά σενάρια. Ακόμα, ο διαχειριστής έχει την δυνατότητα να διαχειριστεί (προσθήκη, διαγραφή ή τροποποίηση) οντότητες (προϊόντα, καταστήματα ή χρήστες), να διαχειριστεί αναφορές χρηστών (και να δράσει ανάλογα, εξετάζοντας κάθε φορά την ορθότητα των αναφορών) αλλά και να δει τα σχετικά στοιχεία (logs) που έχουν καταγραφεί στην πλατφόρμα.



### 3.2.3.7 Δεδομένα εξόδου

Δεδομένα εξόδου του διαχειριστή προκύπτουν από την μεταβολή των στοιχείων ήδη υπαρχόντων καταστημάτων/ προϊόντων/ χρηστών ή την προσθήκη νέων.

### 3.3 Απαιτήσεις επιδόσεων

Το λογισμικό να είναι σε θέση να εξυπηρετήσει το αναμενόμενο πλήθος χρηστών (crowdsources), δηλαδή να εξασφαλίζει την ομαλή περιήγησή τους στο περιβάλλον της εφαρμογής.

Το πλήθος των εισόδων στην εφαρμογή (πχ οι χρήστες που κάνουν browse) ή τα νέα προϊόντα που μπαίνουν στη βάση είναι ένας δείκτης για την επίδοση της εφαρμογής. Μεγάλο πλήθος των παραπάνω στοιχείων δείχνει ότι η εφαρμογή έχει πετύχει τον σκοπό της και προσφέρει στους χρήστες αυτό που τους υποσχέθηκε.

### 3.4 Απαιτήσεις οργάνωσης δεδομένων

#### 3.4.1 Τεχνική περιγραφή των δεδομένων που διαχειρίζεται το λογισμικό και των σχετικών μετρικών φορτίου δεδομένων εισόδου, επεξεργασίας και άλλα.

Για την μορφή των δεδομένων που απαιτούν συγκεκριμένη μορφή χρησιμοποιούνται γνωστά πρότυπα. Για τις ημερομηνίες χρησιμοποιείται το πρότυπο ISO 8601. Για τις γεωγραφικές συντεταγμένες το πρότυπο WGS84.

Η επιλεγμένη τεχνολογία για την βάση δεδομένων (MongoDB) είναι εύκολα οριζόντια κλιμακώσιμη. Όταν χρειάζεται, μπορούν να αυξάνονται τα nodes δεδομένων. Κάτι τέτοιο είναι πιθανό καθώς μεγαλώνει η υπηρεσία και διαχειρίζεται ολοένα και μεγαλύτερο πλήθος queries.

*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

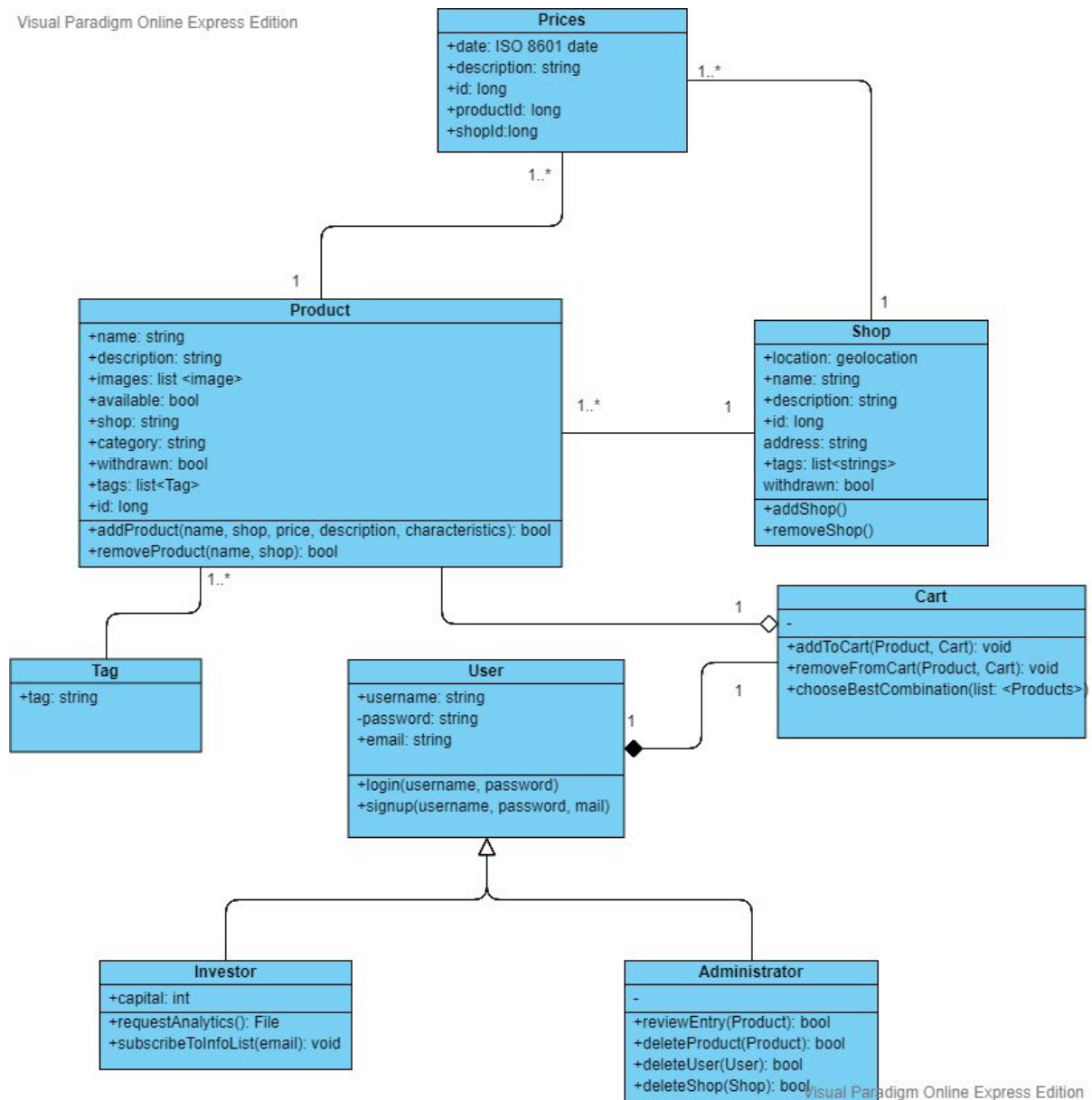
Κάθε χρήστης θα έχει πρόσβαση σε διαφορετικά δεδομένα. Ο επισκέπτης (guest) θα έχει την μικρότερη πρόσβαση, καθώς θα μπορεί να προβάλει όλα τα δεδομένα των προϊόντων και των καταστημάτων. Ο συνδεδεμένος χρήστης (logged-in user) θα έχει πρόσβαση σε ό,τι δεδομένα έχει πρόσβαση ο επισκέπτης, με την επιπλέον δυνατότητα επεξεργασία συγκεκριμένων πεδίων, πράγμα το οποίο δεν μπορεί να κάνει ο επισκέπτης, καθώς περιορίζεται μόνο στην προβολή τους.

Ο επενδυτής (investor) θα μπορεί, εκτός από τις προβολή των δεδομένων των καταστημάτων και των προϊόντων, να έχει πρόσβαση και σε χρήσιμα στατιστικά στοιχεία, σε μορφή πινάκων αλλά και γραφημάτων. Αυτά τα δεδομένα θα μπορεί να τα προβάλει στην πλατφόρμα ή να τα κατεβάσει για εξωτερική χρήση.

Τέλος, ο διαχειριστής (administrator) θα έχει την μεγαλύτερη πρόσβαση από όλους τους χρήστες, καθώς θα μπορεί να προβάλει και να επεξεργαστεί όλα τα δεδομένα που μπορούν και οι προηγούμενοι, θα έχει πρόσβαση στην λίστα και σε όλα τα ανάλογα πεδία των εγγεγραμμένων χρηστών και των επενδυτών, αλλά και στα logs που θα έχουν αποθηκευτεί στην βάση.

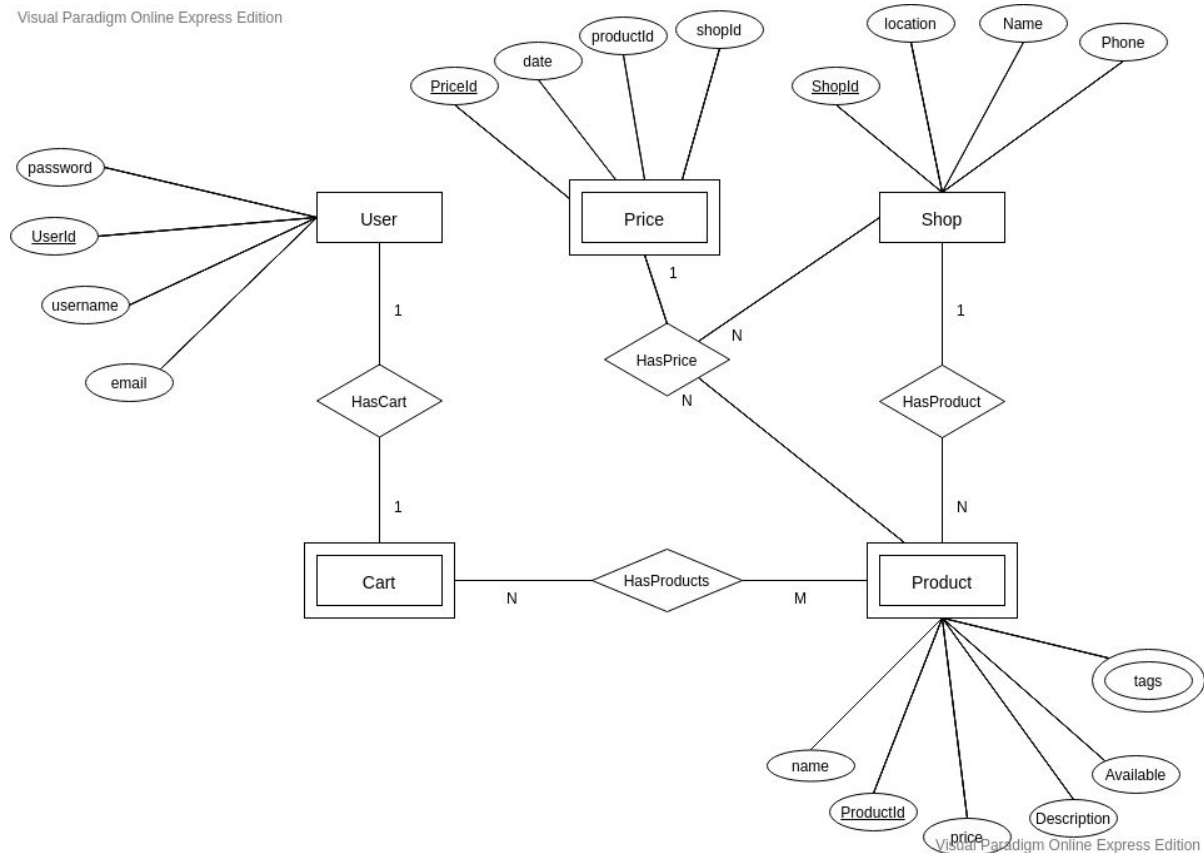
### 3.4.3 Μοντέλο δεδομένων (μοντέλο κλάσεων UML ή/και μοντέλο ER)

Visual Paradigm Online Express Edition



Visual Paradigm Online Express Edition





### 3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Τα δεδομένα που εισάγει ο χρήστης θεωρούνται έγκυρα μέχρι αποδείξεως του αντιθέτου. Όταν ένας χρήστης εισάγει κάποια νέα δεδομένα, από μια χρονική στιγμή και έπειτα όλοι οι υπόλοιποι χρήστες βλέπουν αυτήν την αλλαγή (eventually consistent) και συμφωνούν στα νέα δεδομένα.

### 3.4.5 Προδιαγραφές διατήρησης δεδομένων

Στο σύστημα μας απαιτείται το αναγκαίο redundancy ώστε να διαβεβαιώνουμε ότι σε βάθος χρόνου θα διατηρούνται όλα τα δεδομένα

## 3.5 Περιορισμοί σχεδίασης

Με βάση το workflow και τα frameworks που θα χρησιμοποιηθούν στην σχεδίαση και την ανάπτυξη της εφαρμογής, τίθενται οι παρακάτω περιορισμοί.

Όσον αφορά την ονοματολογία:

- Use lowercase and separate words with an underscore for branch names.
- Use lowercase for directories (do not capitalize first), unless it's a class specific.



- Use PascalCase for class file names (otherwise preferably lowercase).
- Use whole words, not abbreviations, in names.
- Use PascalCase for type names and public methods.
- Use camelCase for local variables and private properties.

## 3.6 Λοιπές απαιτήσεις

### 3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Η εφαρμογή δεν θα είναι σε θέση να απαντήσει με δεδομένα σε περίπτωση μη διαθεσιμότητας του REST API.

### 3.6.2 Απαιτήσεις ασφάλειας

Η σύνδεση του χρήστη στην σελίδα της εφαρμογής θα πρέπει να είναι ασφαλής σύμφωνα με το ανάλογο πρωτόκολλο (https), ενώ και οι κωδικοί των χρηστών θα πρέπει να αποθηκεύονται με ασφαλή τρόπο στη βάση δεδομένων

### 3.6.3 Απαιτήσεις συντήρησης

Η διαχείριση του συστήματος θα είναι υπεύθυνη για την ενημέρωση και συντήρηση του. Οι απαιτήσεις αυτές έγκειται κυρίως στα διαχειριστικά (διασφάλιση ορθής πληροφορίας και ενασχόληση με θέματα των clients) αλλά και σε θέματα ενημέρωσης του ίδιου του λογισμικού του συστήματος.

## 4. Παράρτημα

### 4.1 Παραδοχές και εξαρτήσεις

Παραδεχόμαστε ότι έχουμε γράψει τα μισά στην τύχη και ότι μάλλον θα αλλάξουν πολλά πράγματα μέχρι την άλλη Τρίτη.

### 4.2 Ακρωνύμια και συντομογραφίες

API: Application Programming Interface

HTTPS: Hypertext Transfer Protocol Secure

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

- Το specification του REST API.



