



ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ

---

Έγγραφο απαιτήσεων λογισμικού

---

saiko-killers

# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>3</b>
1.1	Εισαγωγή: σκοπός του λογισμικού	3
1.2	Επισκόπηση του λογισμικού	3
1.3	Διεπαφές	3
1.3.1	Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού	3
1.3.2	Διεπαφές με το χρήστη	5
<b>2</b>	<b>Αναφορές - πηγές πληροφοριών</b>	<b>6</b>
<b>3</b>	<b>Προδιαγραφές απαιτήσεων λογισμικού</b>	<b>7</b>
3.1	Εξωτερικές διεπαφές	7
3.2	Λειτουργίες: περιπτώσεις χρήσης	7
3.2.1	ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Αναζήτηση από Απλό Χρήστη-Παρατηρητή	7
3.2.1.1	Χρήστες (ρόλοι) που εμπλέκονται	7
3.2.1.2	Προϋποθέσεις εκτέλεσης	7
3.2.1.3	Περιβάλλον εκτέλεσης	8
3.2.1.4	Δεδομένα εισόδου	8
3.2.1.5	Παράμετροι	8
3.2.1.6	Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά	8
3.2.1.7	Δεδομένα εξόδου	8
3.2.1.8	Παρατηρήσεις	8
3.2.2	ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Καταγραφή τιμής ενός προϊόντος από εθελοντή	11
3.2.2.1	Χρήστες (ρόλοι) που εμπλέκονται	11
3.2.2.2	Προϋποθέσεις εκτέλεσης	11
3.2.2.3	Περιβάλλον εκτέλεσης	11
3.2.2.4	Δεδομένα εισόδου	11
3.2.2.5	Παράμετροι	11
3.2.2.6	Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά	11
3.2.2.7	Δεδομένα εξόδου	12
3.2.3	ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: ADMIN	14
3.2.3.1	Χρήστες (ρόλοι) που εμπλέκονται	14
3.2.3.2	Προϋποθέσεις Εκτέλεσης	14
3.2.3.3	Περιβάλλον Εκτέλεσης	15
3.2.3.4	Δεδομένα Εισόδου	15
3.2.3.5	Παράμετροι	15
3.2.3.6	Αλληλουχία Ενεργειών - Επιθυμητή Συμπεριφορά	15
3.2.3.7	Δεδομένα εξόδου	16
3.3	Απαιτήσεις επιδόσεων	20
3.4	Απαιτήσεις οργάνωσης δεδομένων	21
3.4.1	Τεχνική περιγραφή των δεδομένων που διαχειρίζεται το λογισμικό και των σχετικών μετρικών φορτίου δεδομένων εισόδου, επεξεργασίας κ.λπ.	21
3.4.2	Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα	21
3.4.3	Μοντέλο δεδομένων (μοντέλο κλάσεων UML ή/και μοντέλο ER)	21
3.4.4	Προδιαγραφές ακεραιότητας δεδομένων	22
3.4.5	Προδιαγραφές διατήρησης δεδομένων	23
3.5	Περιορισμοί σχεδίασης	23
3.6	Λοιπές απαιτήσεις	23
3.6.1	Απαιτήσεις Διαθεσιμότητας Λογισμικού	23
3.6.2	Απαιτήσεις Ασφάλειας	24
3.6.3	Απαιτήσεις Συντήρησης	24
<b>4</b>	<b>Παράρτημα</b>	<b>25</b>

4.1	Παραδοχές και εξαρτήσεις . . . . .	25
4.2	Αχρονύμια και συντομογραφίες . . . . .	25
4.3	Υποστηρικτικά έγγραφα, πρότυπα κ.λπ . . . . .	25

## UML Διαγράμματα

1	Deployment Diagram . . . . .	3
2	Component Diagram . . . . .	4
3	User UI . . . . .	5
4	Simple User: Activity Diagram . . . . .	9
5	Simple User: Sequence Diagram . . . . .	10
6	User: Activity Diagram . . . . .	12
7	User Login: Sequence Diagram . . . . .	13
8	User Add Product: Sequence Diagram . . . . .	14
9	Admin: Activity Diagram . . . . .	16
10	Admin Login: Sequence Diagram . . . . .	17
11	Admin Delete User: Sequence Diagram . . . . .	18
12	Admin Delete Store: Sequence Diagram . . . . .	18
13	Admin Update User: Sequence Diagram . . . . .	19
14	Entity-Relationship Diagram (ERD) . . . . .	22
15	Class Diagram . . . . .	22

# 1 Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Ο σκοπός του συστήματος που υλοποιούμε είναι η δημιουργία μιας ενιαίας και εύκολης στη χρήση πλατφόρμας που ενθαρρύνει τη συνεργατική παρατήρηση τιμών πρατηρίων υγρών καυσίμων και παρέχει αυτές τις πληροφορίες δωρεάν σε κάθε χρήστη. Καθώς τα υγρά καύσιμα αποτελούν βασικό και απαραίτητο αγαθό της σύγχρονης δυτικής κοινωνίας θεωρείται αρκετά πιθανό το παραπάνω σύστημα να προσεγγίσει μεγάλο μέρος πληθυσμού γεγονός που θα ωφελήσει οικονομικά ένα ευρύ μέρος της κοινωνίας με την εύρεση καυσίμων σε χαμηλότερες τιμές.

## 1.2 Επισκόπηση του λογισμικού

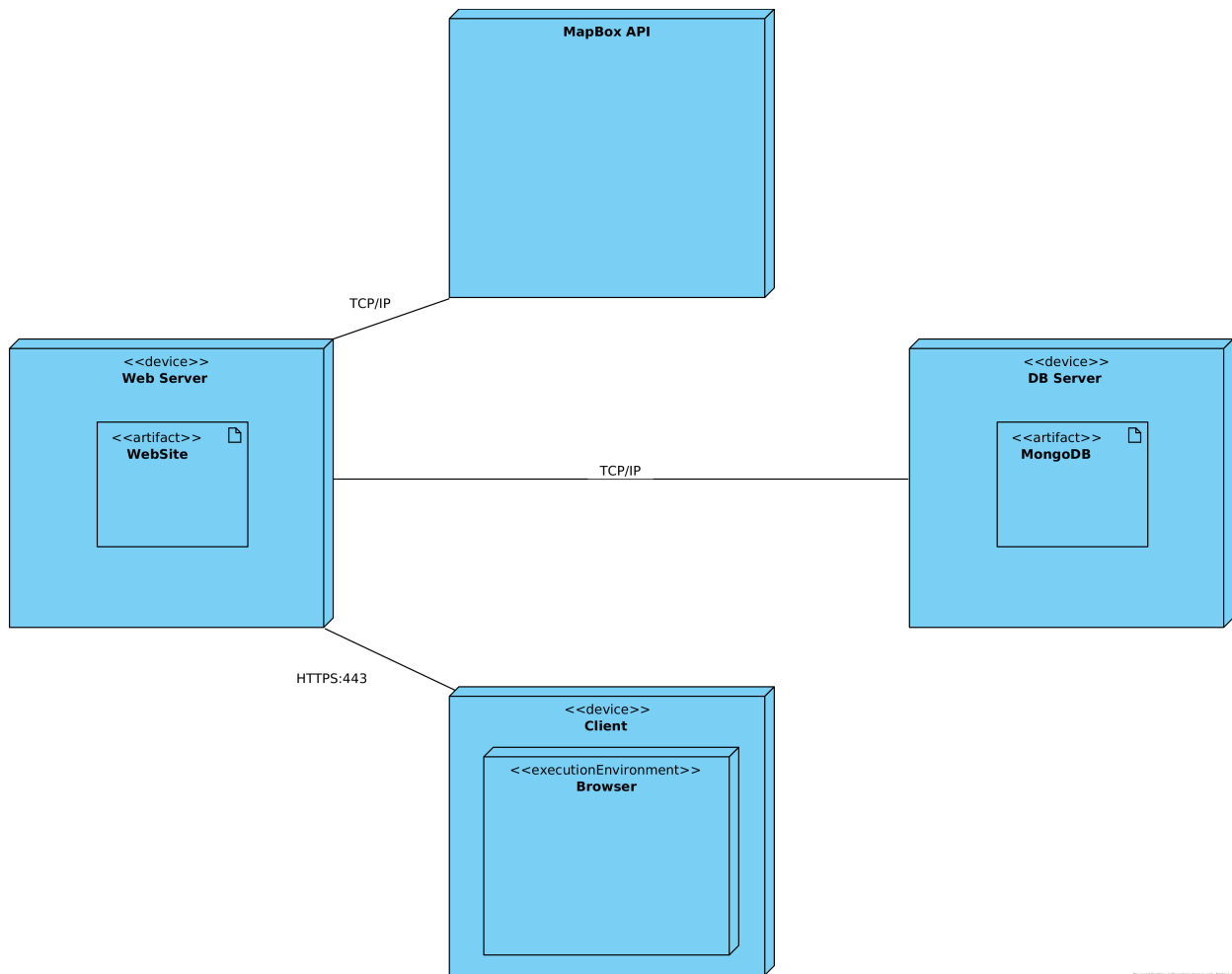


Figure 1: Deployment Diagram

## 1.3 Διεπαφές

### 1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού

Το μοναδικό εξωτερικό σύστημα που χρησιμοποιείται είναι το API του MapBox για τη χρήση του χάρτη και των λειτουργιών του. 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 API του

MapBox.

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 τη 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.

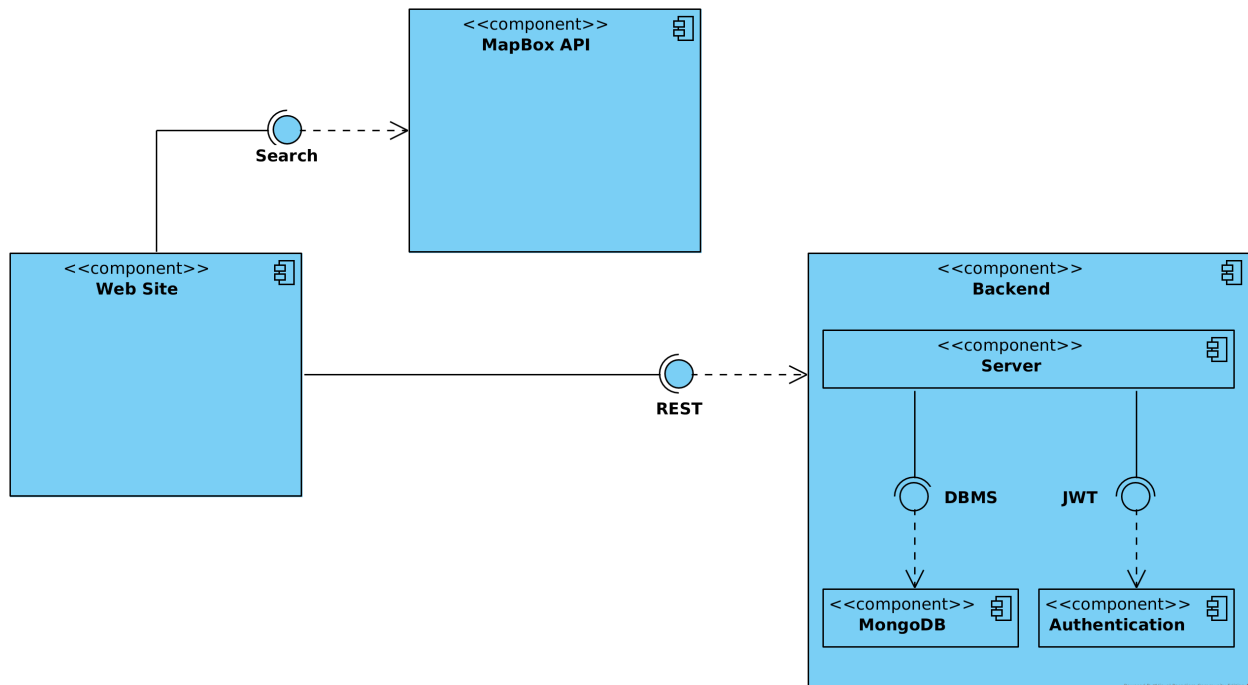


Figure 2: Component Diagram

## 1.3.2 Διεπαφές με το χρήστη

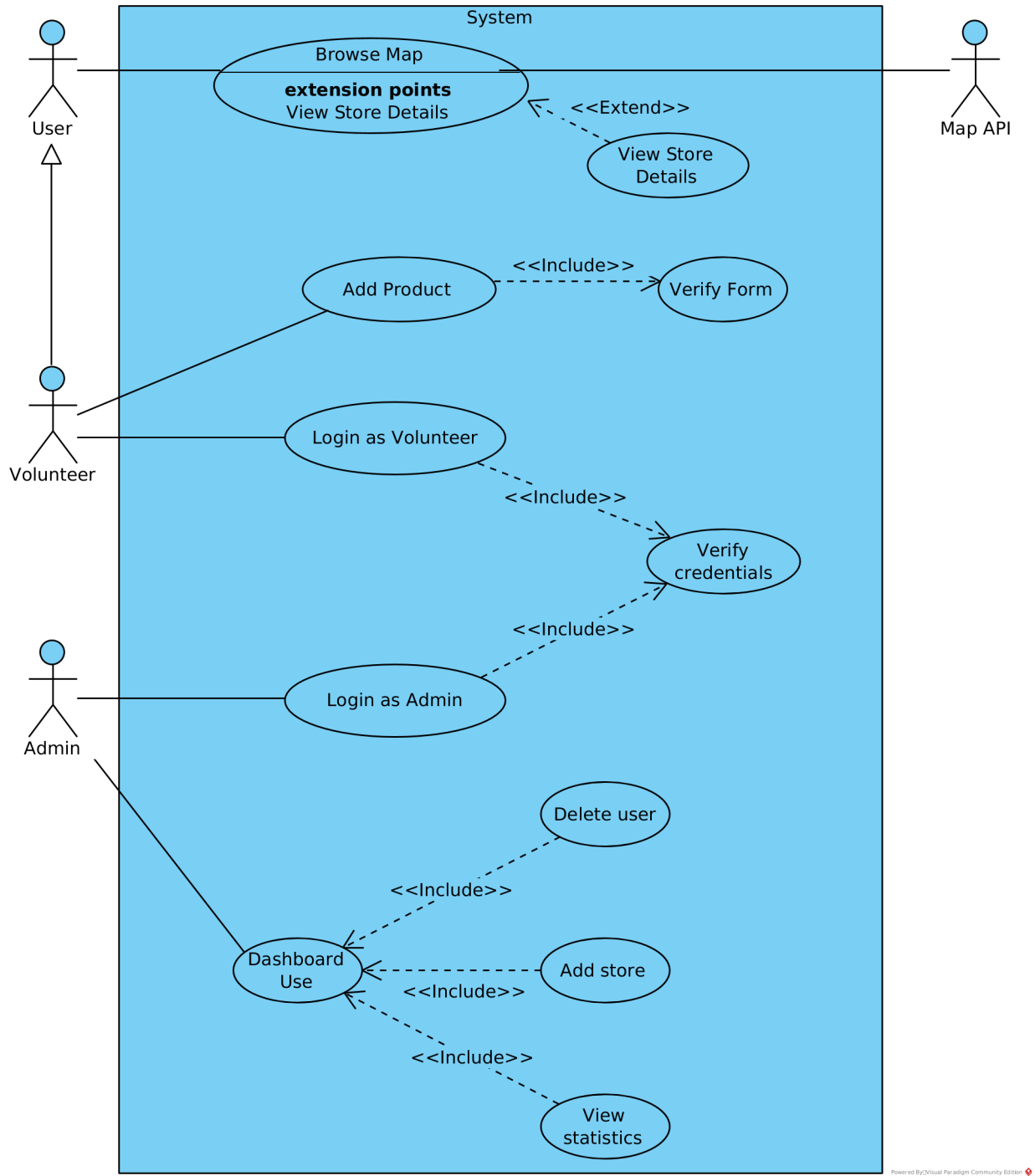


Figure 3: User UI

## 2 Αναφορές - πηγές πληροφοριών

---

Χρησιμοποιήθηκαν οι ακόλουθες πηγές πληροφοριών καθ' όλο το έγγραφο:

1. <https://searchstorage.techtarget.com/video/Most-valuable-storage-metrics-describe-performance-capacity>
2. <https://www.mongodb.com/cloud/atlas/pricing>
3. <https://www.mssqltips.com/sqlservertutorial/2210/maintenance-tasks-for-sql-server/>
4. [https://medium.com/@tabu\\_craig/key-front-end-performance-metrics-and-how-to-capture-them-cae067dada7f](https://medium.com/@tabu_craig/key-front-end-performance-metrics-and-how-to-capture-them-cae067dada7f)
5. <https://techbeacon.com/app-dev-testing/understanding-front-end-vs-back-end-performance-metrics-mobile-apps>
6. <https://sqlperformance.com/2015/05/io-subsystem/analyzing-io-performance-for-sql-server>
7. [https://en.wikipedia.org/wiki/Data\\_integrity](https://en.wikipedia.org/wiki/Data_integrity)
8. <https://www.visual-paradigm.com/>

## 3 Προδιαγραφές απαιτήσεων λογισμικού

### 3.1 Εξωτερικές διεπαφές

Σε αυτή την ενότητα θα περιγράψουμε τις τεχνικές προδιαγραφές των εξωτερικών διεπαφών.

Όσον αφορά το MapBox API, για να μπορέσει να αποτυπώσει σωστά τα καταστήματα στο χάρτη θα χρειαστεί να παρέχουμε τις πληροφορίες με ένα εξειδικευμένο τύπο JSON, τον οποίο το MapBox αποκαλεί GEOJSON. Αυτά είναι της μορφής:

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [lng, lat]
  },
  "properties": {
    "name": name
  }
}
```

Έπειτα, το REST API βασίζεται στο δομότυπο JSON, με τον οποίο η επικοινωνία Frontend και Backend γίνεται πιο εύκολη. Πιο συγκεκριμένα το Frontend κάνει συγκεκριμένα HTTP αιτήματα (στα οποία για τις περιπτώσεις PUT / POST περιέχουν και JSON αντικείμενα) στον Server του Backend, ο οποίος απαντά με ένα JSON Response. Η μορφή ενός τέτοιου μηνύματος είναι η εξής:

```
{
  "name": name,
  "lng": lng,
  "lat": lat,
  "brand": brand,
  ...
}
```

Επιπλέον, για να βεβαιωθούμε ότι κάποιες λειτουργίες μπορούν να γίνουν μόνο από εγγεγραμμένο χρήστη, ενώ κάποιες άλλες μόνο από διαχειριστή, χρησιμοποιούμε JWT Authentication. Αυτό δημιουργεί σε κάθε επιτυχημένο login ένα token το οποίο επιστρέφεται μία φορά κατά το login στο Frontend, το οποίο είναι υποχρεωμένο να το κρατήσει και να το στέλνει στην επικεφαλίδα των HTTP αιτημάτων.

### 3.2 Λειτουργίες: περιπτώσεις χρήσης

#### 3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Αναζήτηση από Απλό Χρήστη-Παρατηρητή

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

Στη συγκεκριμένη περίπτωση χρήσης υπάρχει ουσιαστικά μόνο ο Απλός Χρήστης-Παρατηρητής, ο οποίος εισέρχεται στη σελίδα προκειμένου να παρατηρήσει τις πλέον πρόσφατες τιμές των πρατηρίων υγρών καυσίμων στην περιοχή που επιθυμεί.

##### 3.2.1.2 Προϋποθέσεις εκτέλεσης

Οι συνθήκες οι οποίες πρέπει να ισχύουν είναι οι εξής:

1. Σύνδεση στο διαδίκτυο
2. Η βάση δεδομένων του backend είναι online



3. Η διαδικτυακή διεπαφή λειτουργεί
4. Η διεπαφή με το API του MapBox είναι ενεργή

#### 3.2.1.3 Περιβάλλον εκτέλεσης

Το περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης είναι η διαδικτυακή διεπαφή χρήστη, είτε από κάποιο φυλλομετρητή, είτε από κάποιο smartphone.

#### 3.2.1.4 Δεδομένα εισόδου

Δεν υπάρχουν δεδομένα εισόδου τύπου log-in, μιας και ο απλός χρήστης-παρατηρητής δε χρειάζεται να συνδεθεί. Ωστόσο, μπορούμε να θεωρήσουμε σαν δεδομένο εισόδου τη γεωγραφική τοποθεσία του απλού χρήστη-παρατηρητή που παραχωρείται στο σύστημα εφόσον αυτός το επιθυμεί. Με την είσοδο αυτή, ο χάρτης εντοπίζει την τρέχουσα τοποθεσία του χρήστη. Επίσης, ως δεδομένο εισόδου θεωρούμε την είσοδο στη φόρμα αναζήτησης. Συνθήκη Εγκυρότητας για τα παραπάνω είναι η συσκευή που χρησιμοποιεί ο χρήστης να επιτρέπει την παροχή ακριβούς γεωγραφικής τοποθεσίας, καθώς και να παρέχει συμβατή με το σύστημα κωδικοποίηση.

Ως εξόδο θεωρούμε την εμφάνιση των αποτελεσμάτων αναζήτησης ή των κοντινών στο χρήστη πρατηρίων υγρών καυσίμων στο χάρτη. Ωστόσο, θα πρέπει να τονίσουμε πως δεν υπάρχουν δεδομένα εξόδου σύμφωνα με τον τυπικό ορισμό.

Συνθήκη Εγκυρότητας για τα παραπάνω είναι οι λεπτομέρειες των πρατηρίων υγρών καυσίμων που παρέχονται στο χρήστη να είναι οι πλέον πρόσφατες. Επίσης, το σύστημα να εντοπίζει πράγματι την τοποθεσία του χρήστη. Τα διαγράμματα UML που απεικονίζουν την αλληλουχία ενεργειών του απλού χρήστη-παρατηρητή παρουσιάζονται παρακάτω (βλ. διαγράμματα 4 και 5).

#### 3.2.1.5 Παράμετροι

Μπορούμε να θεωρήσουμε σαν παράμετρο τη γεωγραφική τοποθεσία του απλού χρήστη-παρατηρητή που παραχωρείται στο σύστημα εφόσον αυτός το επιθυμεί. Με τον τρόπο αυτό, το backend μπορεί να απαντήσει στα εισερχόμενα queries που έχουν ως παράμετρο τη γεωγραφική τοποθεσία του απλού χρήστη-παρατηρητή.

#### 3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Είσοδος στην ιστοσελίδα
2. Επιλογή με το αν ο χρήστης επιθυμεί να δώσει την τοποθεσία του στο σύστημα
3. Σε περίπτωση συγκατάθεσης του χρήστη, ο χάρτης κάνει zoom στο σημείο του χρήστη και εμφανίζει τα γειτονικά σε αυτόν πρατήρια υγρών καυσίμων.
4. Σε κάθε περίπτωση ο χρήστης μπορεί να πλοηγηθεί στο χάρτη για να βρει τα επιθυμητά σε αυτόν πρατήρια υγρών καυσίμων
5. Σε κάθε περίπτωση ο χρήστης μπορεί να πραγματοποιήσει αναζήτηση στο χάρτη για να βρει τα επιθυμητά σε αυτόν πρατήρια υγρών καυσίμων.

#### 3.2.1.7 Δεδομένα εξόδου

Στην περίπτωση του απλού χρήστη - παρατηρητή δεν έχουμε δεδομένα εξόδου.

#### 3.2.1.8 Παρατηρήσεις

Θα πρέπει να τονίσουμε πως η παραπάνω διαδικασία αναζήτησης υπάγεται προφανώς και στην περίπτωση του Εθελοντή.

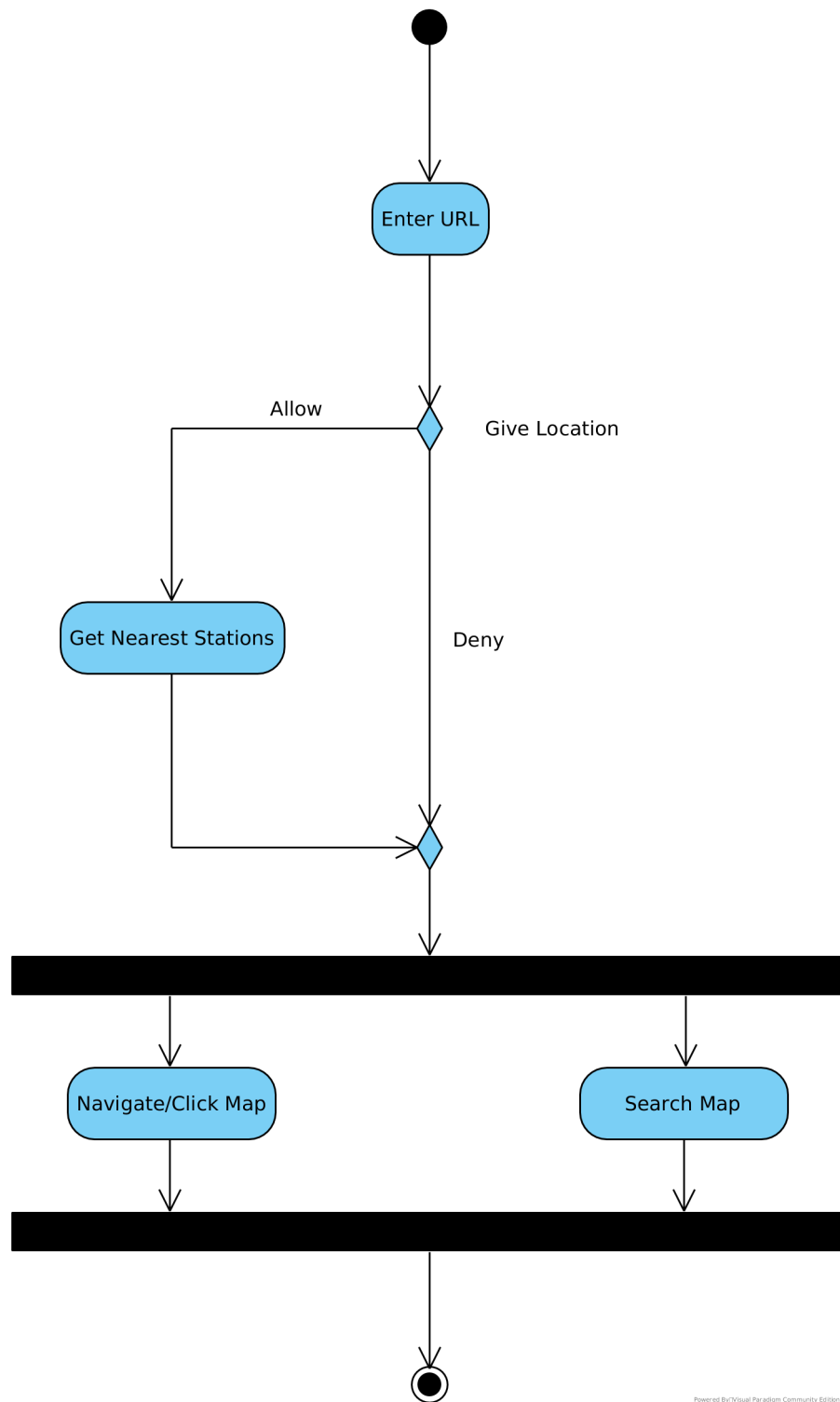


Figure 4: Simple User: Activity Diagram

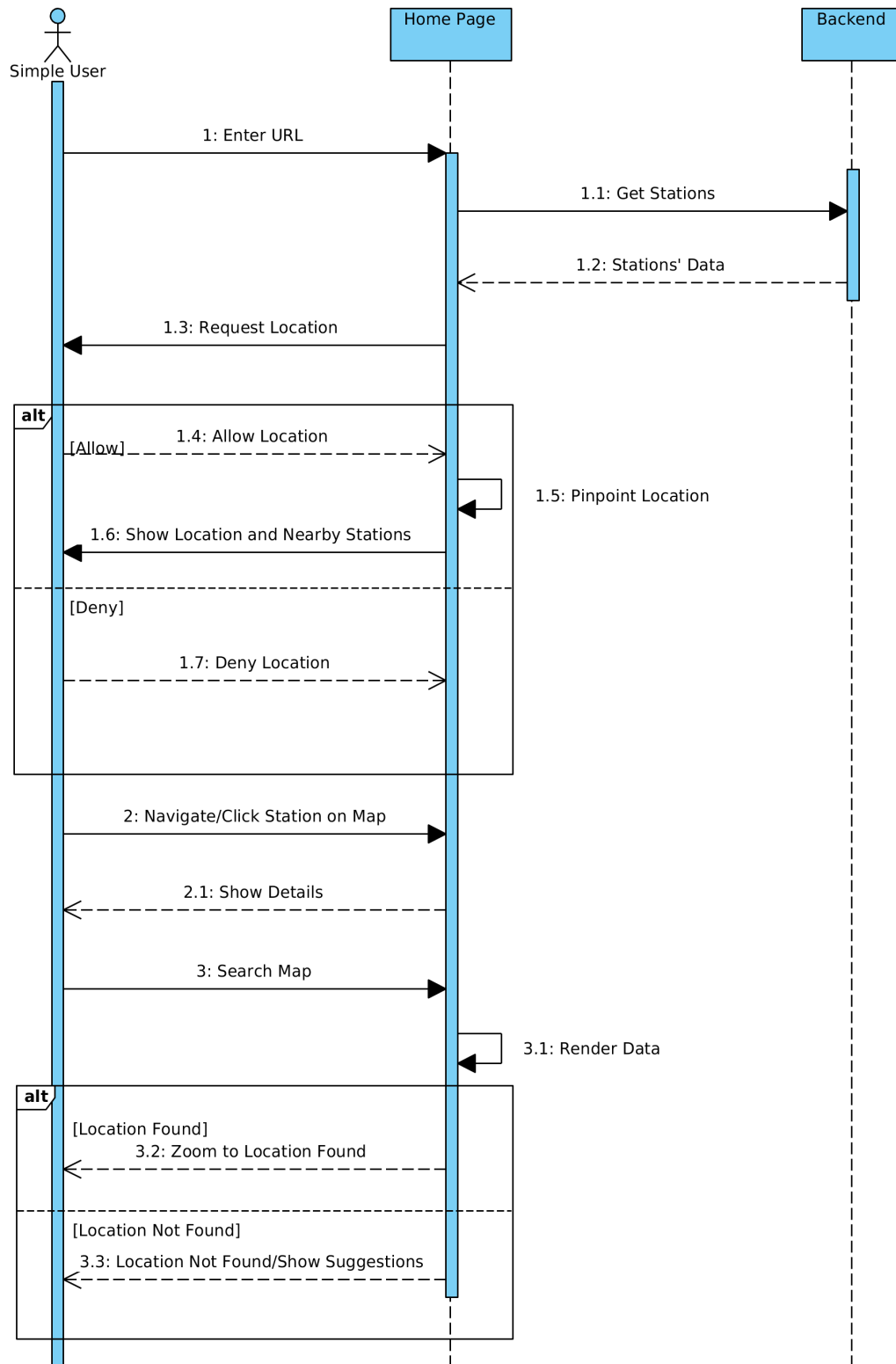


Figure 5: Simple User: Sequence Diagram

### 3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Καταγραφή τιμής ενός προϊόντος από εθελοντή

#### 3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Ο μόνος που εμπλέκεται σε αυτήν την περίπτωση χρήσης είναι ο εθελοντής που επιθυμεί να πραγματοποιήσει την καταγραφή της πλέον πρόσφατης τιμής ενός προϊόντος κάποιου πρατηρίου υγρών καυσίμων.

#### 3.2.2.2 Προϋποθέσεις εκτέλεσης

Για την επιτυχή εκτέλεση του παραπάνω σεναρίου χρήσης, θα πρέπει να ικανοποιούνται οι παρακάτω προϋποθέσεις:

1. Σύνδεση στο διαδίκτυο
2. Η βάση δεδομένων του backend είναι online
3. Η διαδικτυακή διεπαφή λειτουργεί
4. Ο χρήστης πρέπει να είναι συνδεδεμένος στο λογαριασμό του
5. Το κατάστημα το οποίο διαθέτει το προϊόν να βρίσκεται στη βάση
6. Ο τύπος του προϊόντος να είναι διαθέσιμος

#### 3.2.2.3 Περιβάλλον εκτέλεσης

Το περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης είναι η διαδικτυακή διεπαφή χρήστη, είτε από κάποιο φυλλομετρητή, είτε από κάποιο smartphone.

#### 3.2.2.4 Δεδομένα εισόδου

Σαν δεδομένα εισόδου θεωρούνται τα στοιχεία που δίνει ο εθελοντής-χρήστης προκειμένου να πραγματοποιηθεί το login, καθώς και τα στοιχεία που απαιτούνται για την εν λόγω καταχώρηση.

Οι συνθήκες εγκυρότητας για τα παραπάνω είναι η ταυτοποίηση του χρήστη και η ύπαρξη του συγκεκριμένου πρατηρίου υγρών καυσίμων (μέσω της βάσης δεδομένων). Επίσης, η συσκευή που χρησιμοποιεί ο χρήστης να είναι συμβατή με την κωδικοποίηση κειμένου του συστήματος (utf8).

Ως δεδομένο εξόδου θεωρείται η ενημέρωση της βάσης δεδομένων για την τιμή του προϊόντος που καταχώρησε ο χρήστης.

Οι συνθήκες εγκυρότητας για το παραπάνω είναι ο χρήστης να συμπληρώσει όλα τα απαραίτητα πεδία για την επιθυμητή καταχώρηση καθώς και η ύπαρξη του προϊόντος στη βάση δεδομένων.

#### 3.2.2.5 Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών

#### 3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Τα βήματα για την πραγματοποίηση της καταχώρησης είναι τα εξής:

1. Είσοδος στην ιστοσελίδα
2. Επιλογή του sign in
3. Εισαγωγή στοιχείων για την πραγματοποίηση του login
4. Εμφάνιση homepage, η οποία θα περιέχει το χάρτη
5. Επιλογή πρατηρίου από τον χάρτη
6. Επιλογή κουμπιού καταχώρησης τιμής στο pop-up που εμφανίζεται από την επιλογή του πρατηρίου
7. Συμπλήρωση απαραίτητων πεδίων για την καταχώρηση της τιμής

Το αντίστοιχο διάγραμμα είναι το 7

**3.2.2.7 Δεδομένα εξόδου**

Εξετάζεται η περίπτωση προσθήκης προϊόντος στο διάγραμμα 8.

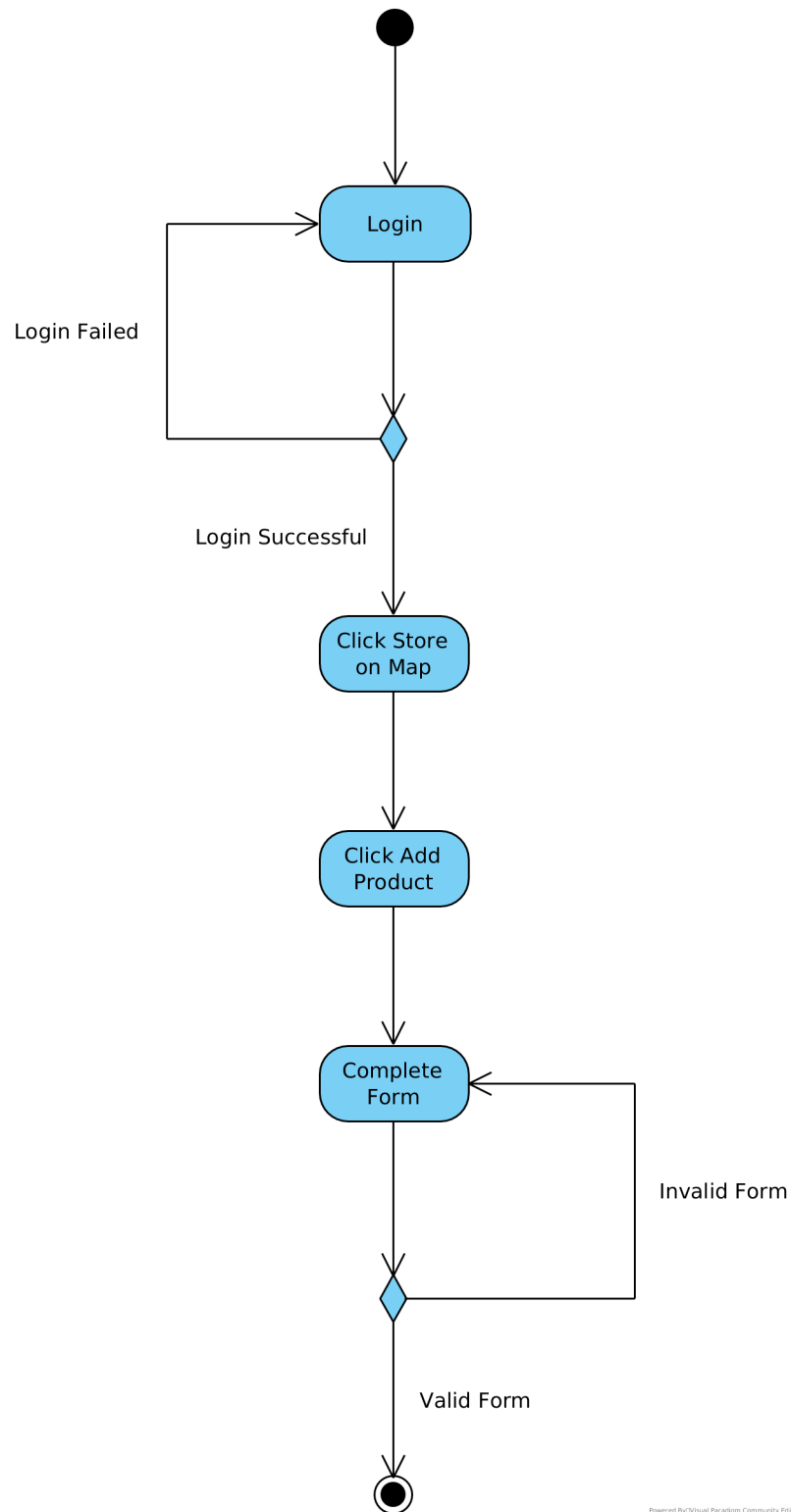


Figure 6: User: Activity Diagram

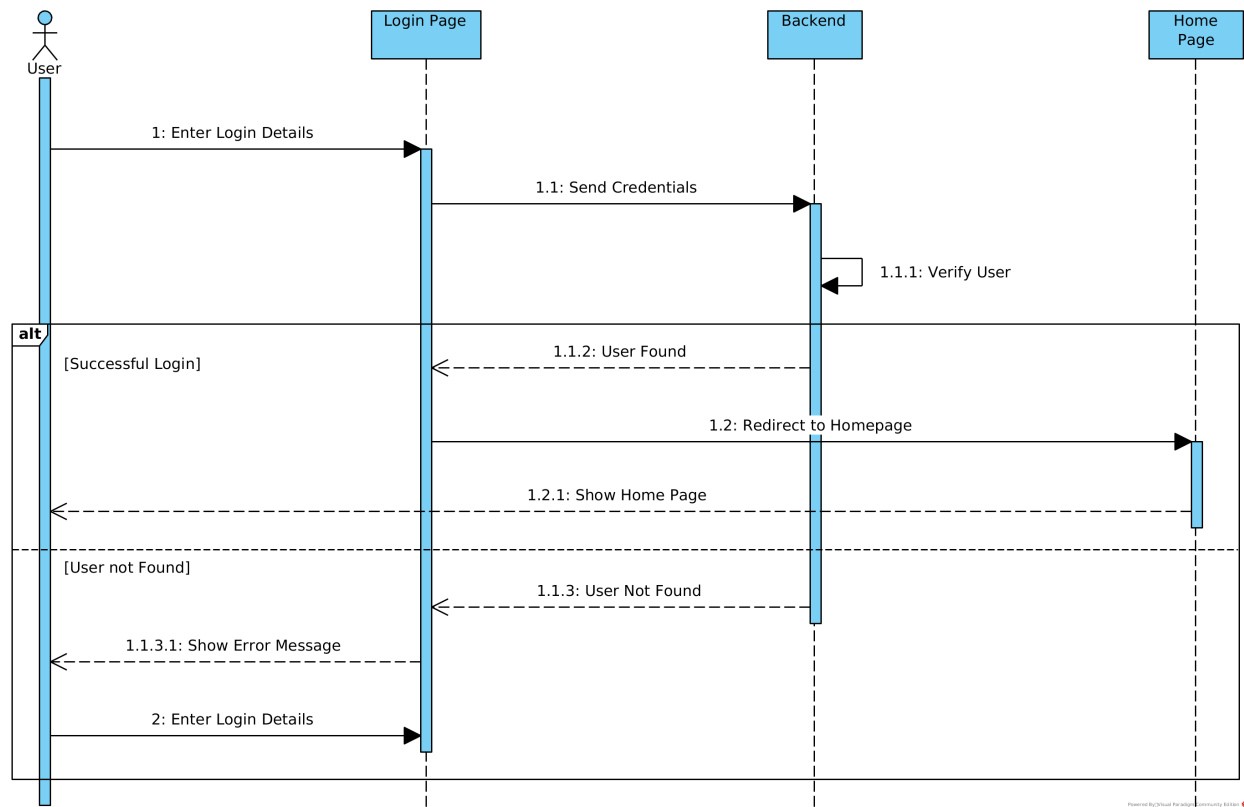


Figure 7: User Login: Sequence Diagram

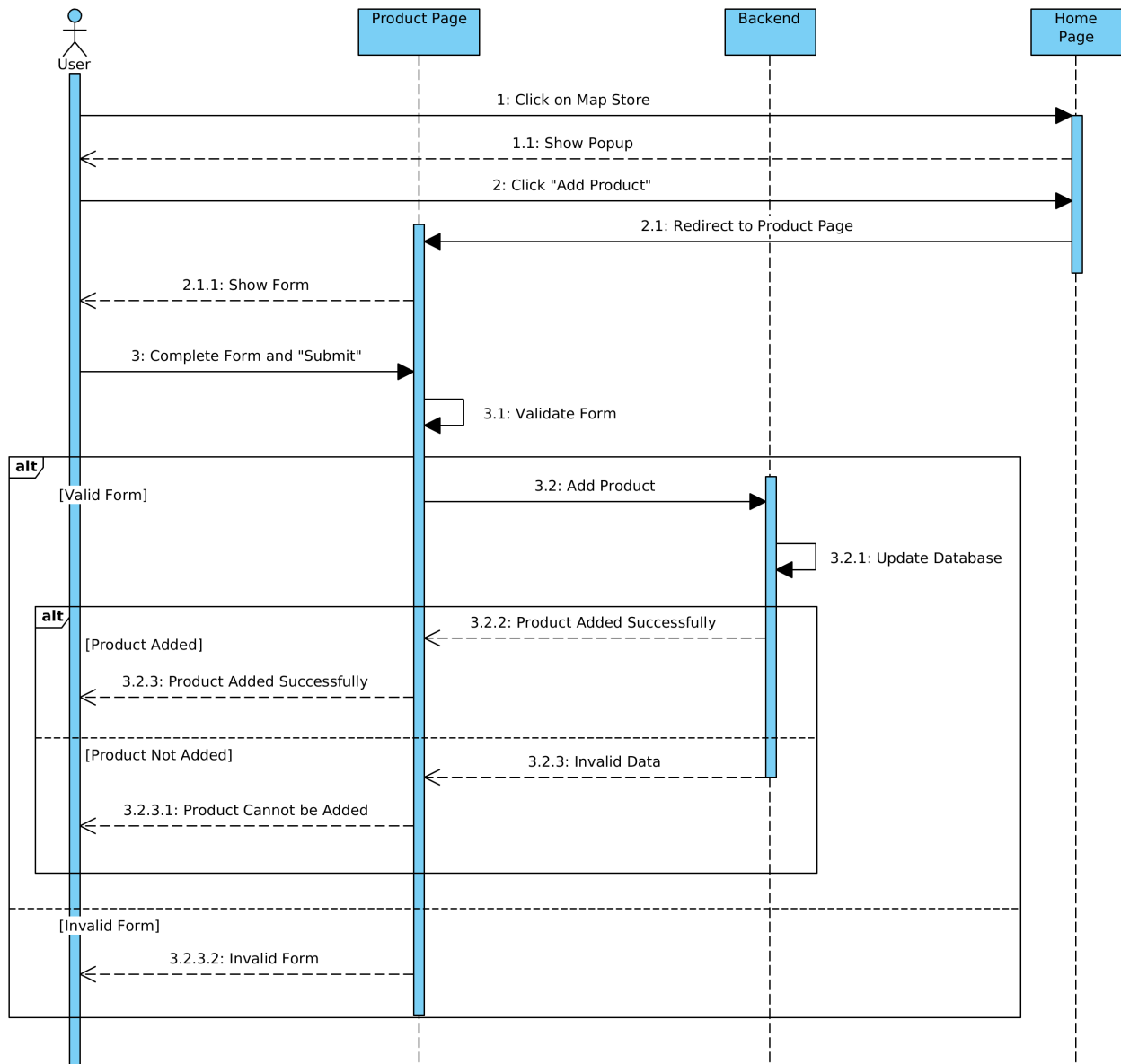


Figure 8: User Add Product: Sequence Diagram

### 3.2.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: ADMIN

#### 3.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται

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

#### 3.2.3.2 Προϋποθέσεις Εκτέλεσης

Οι συνθήκες οι οποίες πρέπει να πληρούνται είναι:

1. σύνδεση στο διαδίκτυο
2. η βάση δεδομένων του backend είναι online
3. η διαδικτυακή διεπαφή λειτουργεί
4. η διεπαφή με το API του MapBox είναι ενεργή

5. ο χρήστης που προσπαθεί να κάνει login πρέπει να έχει credentials διαχειριστή

#### 3.2.3.3 Περιβάλλον Εκτέλεσης

Το περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης είναι η διαδικτυακή διεπαφή χρήστη, το οποίο μετατρέπεται σε dashboard μετά το επιτυχημένο login, είτε από κάποιο φυλλομετρητή, είτε από κάποιο smartphone.

#### 3.2.3.4 Δεδομένα Εισόδου

Τα δεδομένα εισόδου είναι:

1. username
2. password

Η εγκυρότητα των άνω έγκειται στο γεγονός ότι τα credentials αντιστοιχούν σε ένα χρήστη που έχει ρόλο διαχειριστή. Τα δεδομένα εξόδου είναι:

1. επιτυχής διαγραφή χρήστη
2. επιτυχής αναβάθμιση χρήστη σε διαχειριστή
3. επιτυχής διαγραφή καταστήματος

Η εγκυρότητα των άνω έγκειται στο γεγονός ότι ο διαχειριστής χρησιμοποιεί σωστά τις λειτουργικότητες που παρέχει η διαδικτυακή διεπαφή, ενώ ταυτόχρονα τα δεδομένα που εμφανίζονται από αυτή είναι συνεπή με τη βάση δεδομένων μας.

#### 3.2.3.5 Παράμετροι

Δεν υπάρχουν παράμετροι

#### 3.2.3.6 Αλληλουχία Ενεργειών - Επιθυμητή Συμπεριφορά

Τα βήματα που απαιτούνται για την περίπτωση χρήσης είναι:

1. εισαγωγή username και password για σύνδεση διαχειριστή
2. επιβεβαίωση ρόλου διαχειριστή
3. φόρτωση δεδομένων εμφάνισης
4. εμφάνιση dashboard
5. επιτέλεση λειτουργίας
  - (a) Διαγραφή Χρήστη
    - κλικ "Χρήστες"
    - εμφάνιση χρηστών σε λίστα
    - κλικ "Διαγραφή" σε κάποιο χρήστη
  - (b) Διαγραφή Καταστήματος
    - κλικ "Καταστήματα"
    - εμφάνιση καταστημάτων σε λίστα
    - κλικ "Διαγραφή" σε κάποιο κατάστημα
  - (c) Προαγωγή Χρήστη σε Admin
    - κλικ "Χρήστες"
    - εμφάνιση χρηστών σε λίστα
    - κλικ "Προαγωγή" σε κάποιο χρήστη



Ένα γενικό Activity Diagram για τις ενέργειες του διαχειριστή είναι το 9. Το Sequence Diagram για το login as administrator και την εμφάνιση του dashboard είναι το 10. (Οι υπόλοιπες ενέργειες αναφέρονται στα **Δεδομένα εξόδου**)

### 3.2.3.7 Δεδομένα εξόδου

Εξετάζονται οι ακόλουθες περιπτώσεις για τον "Διαχειριστή":

- Διαγραφή χρήστη, βλ. 11
- Διαγραφή καταστήματος, βλ. 12
- Ενημέρωση χρήστη. βλ, 13

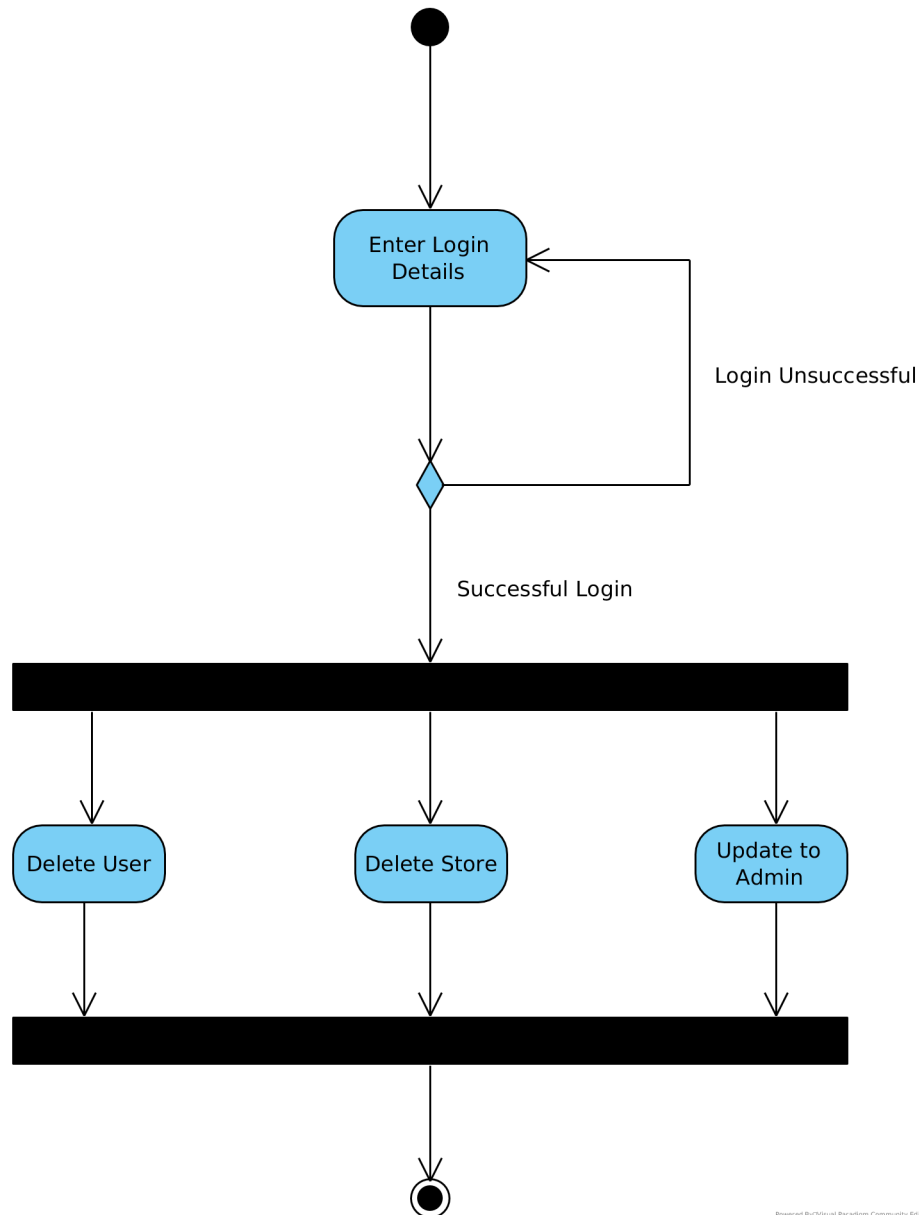


Figure 9: Admin: Activity Diagram

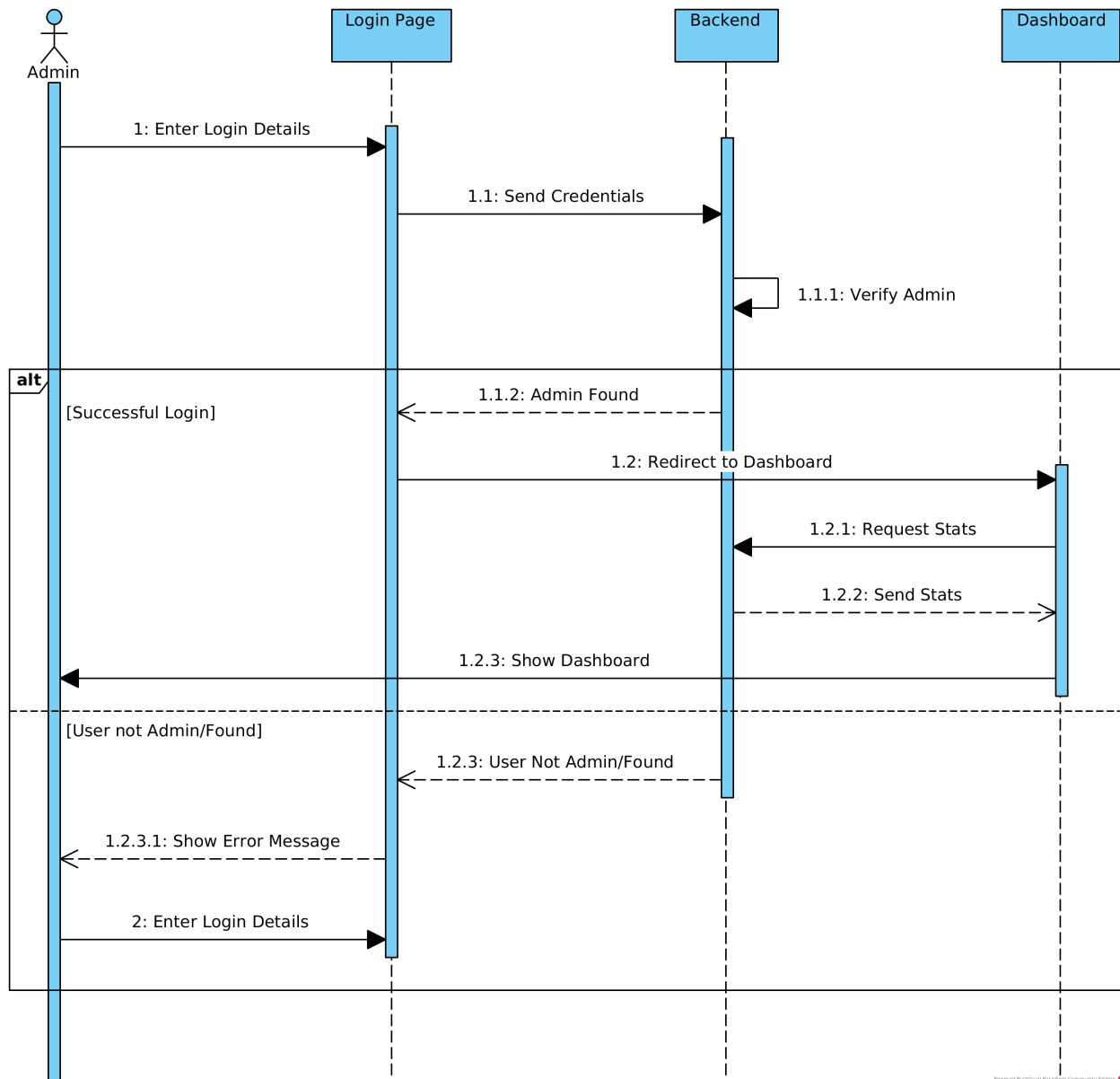
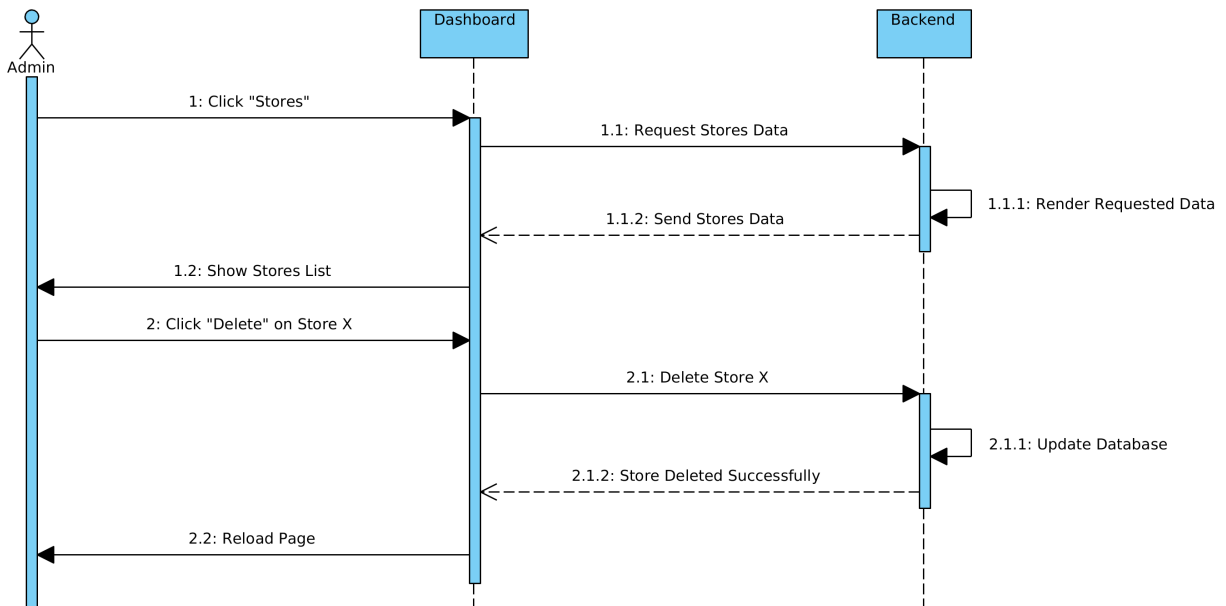
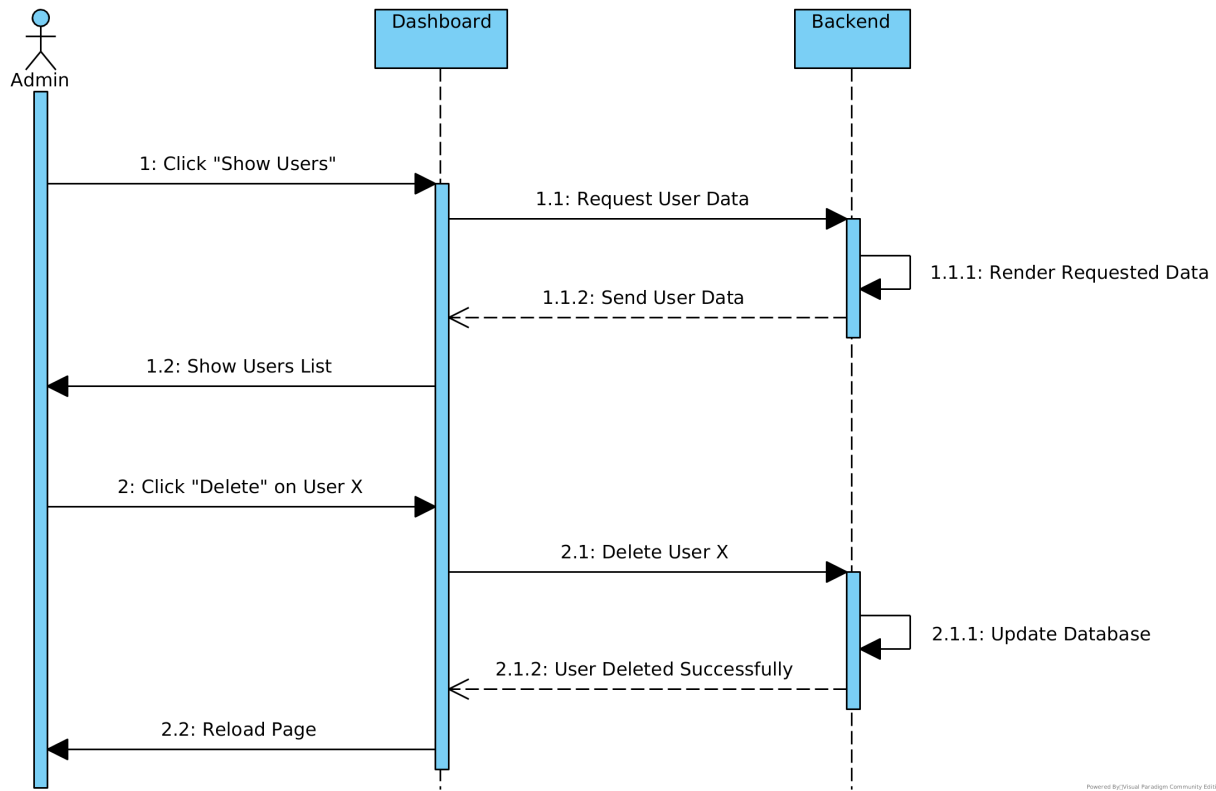


Figure 10: Admin Login: Sequence Diagram



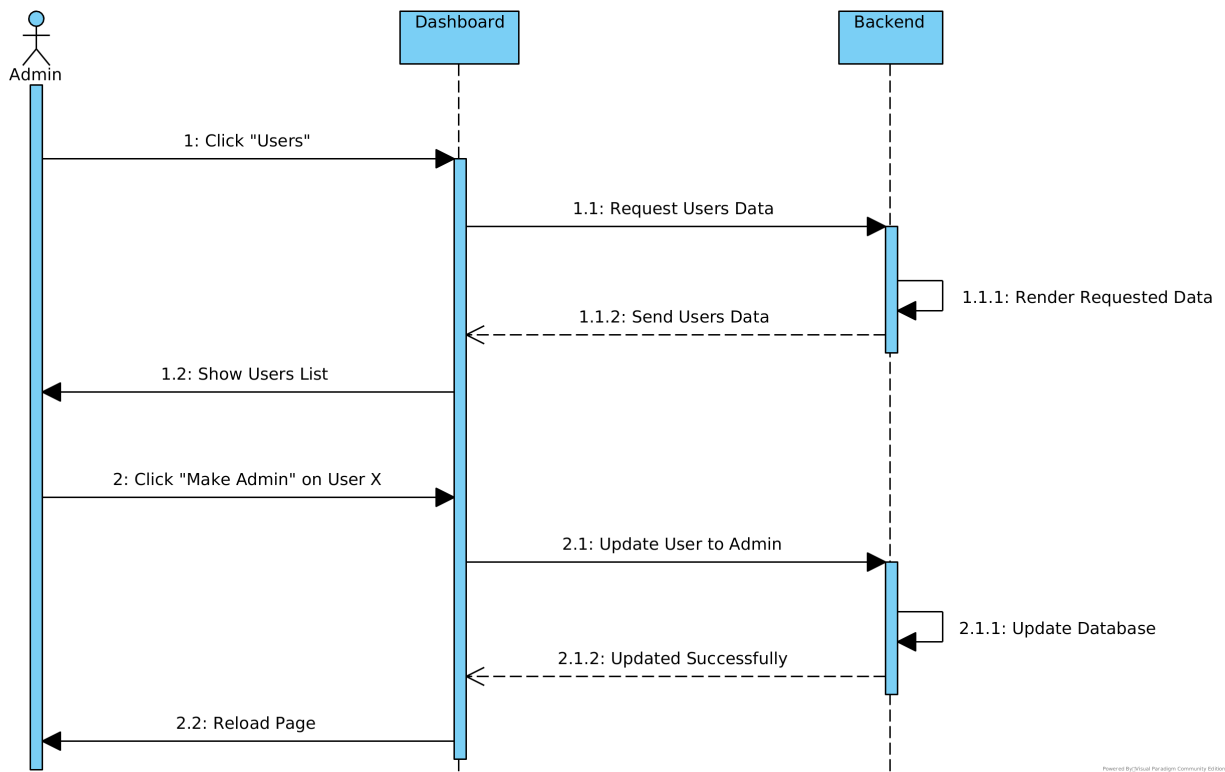


Figure 13: Admin Update User: Sequence Diagram

### 3.3 Απαιτήσεις επιδόσεων

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

Θα παρουσιάσουμε τις απαιτήσεις επιδόσεων για το σύστημα. Θα διαχωρίσουμε τα μέτρα σε back-end και front-end.

#### Back-end

1. **Υψηλή ταχύτητα φόρτωσης δεδομένων:** υποδεικνύει την ταχύτητα με την οποία φορώνονται και είναι διαθέσιμα τα δεδομένα
2. **Latency:** υποδεικνύει το χρόνο ολοκλήρωσης μίας λειτουργίας I/O
3. **Υψηλό IOPS<sup>1</sup>:** πρόκειται για το πλήθος των λειτουργιών I/O που εκτελούνται ανά δευτερόλεπτο. Σχετίζεται άμεσα με το Latency.
4. **Sequential Throughput:** πρόκειται για το 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 τα χρησιμοποιούμε φειδωλά.

Όλα τα παραπάνω να ισχύουν και σε συνθήκες υψηλού φορτίου (workload): αναλυτικότερα επιθυμούμε το σύστημα

#### Front-end

1. **First Meaningful Paint:** πρόκειται για το χρόνο που απαιτείται για να φορτωθεί το πρώτο στοιχείο της σελίδας
2. **Time to Interact:** υποδεικνύει πότε η σελίδα θα είναι responsive προκειμένου ο χρήστης να αλληλεπιδράσει μαζί της
3. **Interaction frame rate:** επιθυμούμε το frame rate της ιστοσελίδας να είναι υψηλότερο από αυτό της οθόνης του χρήστη (οι περισσότερες οθόνες λειτουργούν στα 60 fps)
4. **Page rendering:** σχετίζεται με τη μεταφόρτωση πόρων (όπως εικόνες κ.α.) στην ιστοσελίδα. Όσο λιγότεροι οι απαιτούμενοι πόροι τόσο πιο χαμηλό το page rendering.

---

<sup>1</sup>Input/Output Operations per Second

### 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 δεδομένου εισόδου είναι τα ακόλουθα:

1. προσωπικά δεδομένα / πληροφορίες χρηστών
2. γεωγραφικές πληροφορίες (τοποθεσία χρήστη, πρατήρια υγρών καυσίμων)
3. πληροφορίες πρατηρίων υγρών καυσίμων
4. πληροφορίες προϊόντων
5. πληροφορίες τιμών
6. API call parameters

Όσον αφορά τα πρότυπα δεδομένων και υπηρεσιών, χρησιμοποιούνται τα ακόλουθα:

- HTTP
- TCP/IP

Τέλος, όσον αφορά τις μετρικές που σχετίζονται με τα δεδομένα (storage capacity planning) έχουμε τις ακόλουθες μετρικές:

- **Performance:** Σχετικές μετρικές είναι:
  - **Latency:** το response time μίας λειτουργίας
  - **Bandwidth:** το μέγεθος των δεδομένων που μεταφέρεται σε ένα κβάντο χρόνου
- **Availability:** η διαθεσιμότητα των δεδομένων. Η μετρική αυτή μελετά, μεταξύ άλλων, τη συμπεριφορά του συστήματος στην περίπτωση διακοπής δικτύου
- **Capacity:** υποδεικνύει τις δυνατότητες αποθήκευσης δεδομένων
- **Economics:** υποδεικνύει το κόστος λειτουργίας της ιστοσελίδας, όπως το κόστος φιλοξενίας της Βάσης Δεδομένων στο MongoDB cloud.

#### 3.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

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

- **Απλός Χρήστης - Παρατηρητής:** Ο Απλός Χρήστης - Παρατηρητής έχει τη δυνατότητα να παρακολουθεί δεδομένα που αφορούν πρατηρίων υγρών καυσίμων και προϊόντα.
- **Εθελοντής:** Ο Εθελοντής πέρα από τις δυνατότητες του Απλού Χρήστη - Παρατηρητή, έχει επιπλέον τη δυνατότητα να καταχωρεί πληροφορίες σχετικά με προϊόντα στο σύστημα
- **Administrator:** Ο Administrator έχει τη δυνατότητα να παρακολουθεί και να διαμορφώνει όλα τα δεδομένα της βάσης δεδομένων.

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

Παρατίθενται παρακάτω τα ζητούμενα.

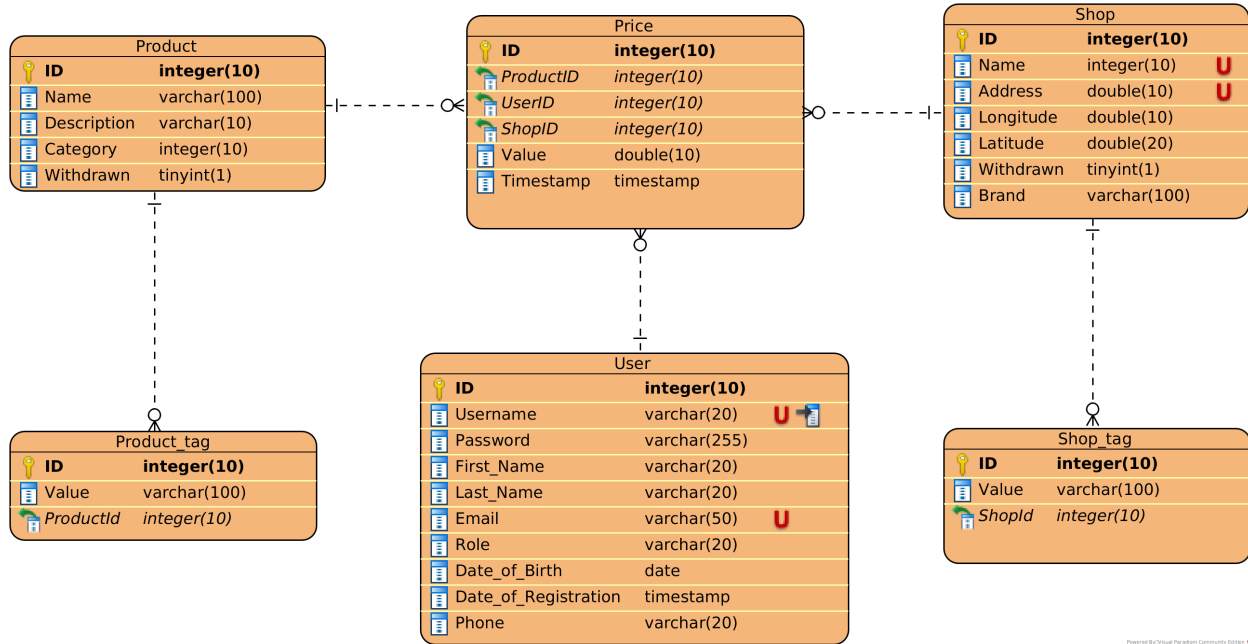


Figure 14: Entity-Relationship Diagram (ERD)

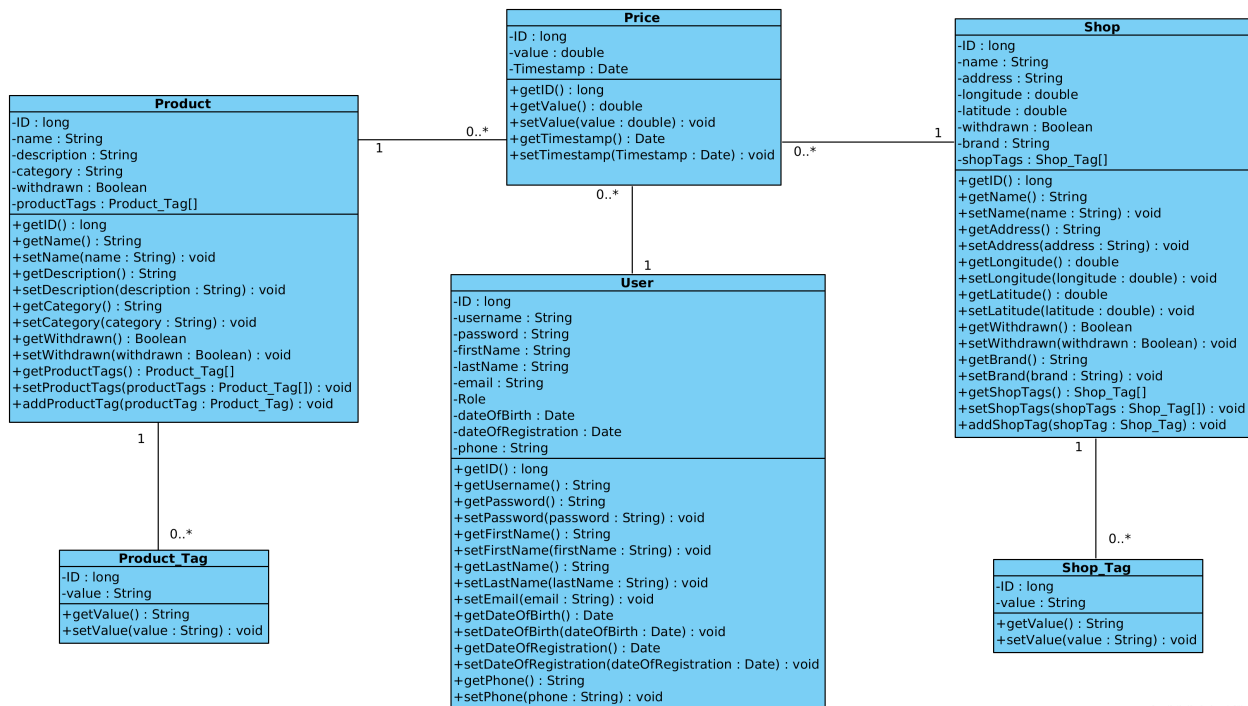


Figure 15: Class Diagram

### 3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Παρακάτω παρουσιάζονται οι κανόνες ακεραιότητας και εγκυρότητας δεδομένων τόσο σε φυσικό όσο και σε λογικό επίπεδο.

### Φυσικό Επίπεδο

1. Χρήση συστοιχιών RAID για αποφυγή σφαλμάτων υλικού στη βάση δεδομένων.
2. Χρήση error detecting algorithms ή αλλιώς error-correcting codes (π.χ. parity bit)

### Λογικό Επίπεδο

1. Αρχές Μεθοδολογίας ACID: Η βάση δεδομένων μας θα πρέπει να ακολουθεί κανόνες ACID για τις δοσοληψίες της.
2. Entity Integrity: Ύπαρξη μοναδικών primary keys για κάθε οντότητα.
3. Referential integrity: Ύπαρξη foreign keys μεταξύ οντοτήτων.
4. Domain integrity: Κάθε πεδίο πρέπει να έχει ένα συγκεκριμένο τύπο δεδομένων.
5. User-defined integrity: Χρήση user defined rules (π.χ. NOT NULL κ.τ.λ)

#### 3.4.5 Προδιαγραφές διατήρησης δεδομένων

Παρακάτω παρουσιάζονται προδιαγραφές διατήρησης δεδομένων

- Δημιουργία αντιγράφων backup της βάσης δεδομένων και των log files της.
- Ανανέωση στατιστικών στοιχείων στη βάση δεδομένων για optimized λειτουργία.
- Τακτικοί έλεγχοι για τη συνέπεια της βάσης δεδομένων.
- Τακτικό rebuild των indices της βάσης δεδομένων
- Τακτικά clean up tasks
- Shrinking δεδομένων και αφαίρεση άδειων σελίδων από τη βάση δεδομένων.
- Συνεχή updates για τη χρήση των πλέον πρόσφατων εκδόσεων λογισμικού.

### 3.5 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 διότι 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 τη βάση δεδομένων είναι το MongoDB. Σημειώνουμε επίσης ότι επιλέχθηκε και ο online server που παρέχει αυτό το εργαλείο. Τέλος, για τη δυναμική υλοποίηση του frontend, χρησιμοποιούμε TypeScript, HTML και CSS μέσω της Angular.

Όσον αφορά το backend, για την ονοματολογία των οντοτήτων χρησιμοποιούμε κεφαλαίο στο πρώτο γράμμα (π.χ. Price, User, κτλ) και για τις συναρτήσεις μικρό το πρώτο γράμμα, ενώ όταν υπάρχουν πολλές λέξεις, η κάθε μια ξεκινά με κεφαλαίο γράμμα (π.χ. Price.findById). Αντίστοιχα, στο frontend η μόνη διαφορά σε σχέση με πριν είναι ότι τα ονόματα των οντοτήτων ξεκινούν με μικρό γράμμα.

Η χρήση του εργαλείου MongoDB για την επικοινωνία του backend με τη βάση δεδομένων περιέχει τον περιορισμό των 512 MB storage, καθώς επιλέξαμε το free plan για το εργαλείο αυτό. Ανάλογα βέβαια με την επιτυχία της εφαρμογής μας, μπορούμε, αν χρειαστεί, να χρησιμοποιήσουμε κάποιο αναβαθμισμένο plan του εργαλείου αυτού, προκειμένου να επεκταθεί το μέγεθος του storage.

Σχετικά με τη συμμόρφωση σε πρότυπα, για τις συντεταγμένες των χρηστών και των καταστημάτων χρησιμοποιείται το σύστημα συντεταγμένων WGS84. Επιπλέον, οι κωδικοί για το login των χρηστών αποθηκεύονται σε hashed μορφή, το οποίο εξασφαλίζει ότι είναι κωδικοποιημένοι.

### 3.6 Λοιπές απαιτήσεις

#### 3.6.1 Απαιτήσεις Διαθεσιμότητας Λογισμικού

Απαιτούμε τόσο η διαδικτυακή διεπαφή, όσο και η διεπαφή με τη βάση δεδομένων να είναι ενεργές καθόλη τη διάρκεια της ημέρας, καθώς και σε βάθος χρόνου. Με αυτό τον τρόπο, θα εξασφαλίσουμε την καθολική



διαθεσιμότητα της πλατφόρμας "SaikoFuel", με σκοπό την καλύτερη εξυπηρέτηση των χρηστών.

### 3.6.2 Απαιτήσεις Ασφάλειας

Η διαδικτυακή μας πλατφόρμα χρησιμοποιεί το πρωτόκολλο HTTPS, ώστε να σιγουρευτούμε πως τα δεδομένα των χρηστών προστατεύονται από επιθέσεις. Επίσης, τα δεδομένα της βάσης δεδομένων απαιτούμε να είναι κρυφά προς τον έξω κόσμο και να μπορούν μόνο οι διαχειριστές να έχουν πρόσβαση σε αυτά. Ως επιπλέον θυρίδα ασφαλείας, χρησιμοποιούμε ταυτοποίηση με hash authentication, ώστε να βεβαιωθούμε ότι δεν μπορεί να γίνει reverse engineering σε αυτή την τεχνική.

### 3.6.3 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
- 4.2 Ακρωνύμια και συντομογραφίες
- 4.3 Υποστηρικτικά έγγραφα, πρότυπα κ.λπ