

# Έγγραφο απαιτήσεων λογισμικού (SRS)

[Stingy]

## 1. Εισαγωγή

### 1.1 Εισαγωγή: σκοπός του λογισμικού

Το εγχείρημα αφορά την ανάπτυξη μιας λειτουργικής και εύχρηστης πλατφόρμας/λογισμικού (ένα online παρατηρητήριο τιμών), που έχει ως απώτερο σκοπό την αντιμετώπιση της διαδικτυακής – και μη – υπερπληροφορίας για τεχνολογικά προϊόντα.

Στην κατεύθυνση αυτή, επιλέγονται *εθελοντές* για το σύστημα, οι οποίοι καταγράφουν χαρακτηριστικά τεχνολογικών προϊόντων (desktops, laptops, tablets κλπ) για *εταιρείες* που εμφανίζουν σημαντική online και offline ποικιλομορφία και έχουν περαιτέρω ταυτοποιηθεί και ελεγχθεί από τον *διαχειριστή* του συστήματος.

Παράλληλα, προσφέρονται στους τελικούς *χρήστες* δυνατότητες αναζήτησης προϊόντων και αξιολόγησης (εγγεγραμμένοι *χρήστες*) με απόρροια την δημιουργία ενός δικτύου ενημέρωσης (crowdsourcing). Τα διαδικτυακά αυτά ίχνη μετατρέπονται σε analytics, με την μορφή πακέτου υπηρεσιών για το υποσύνολο *εταιρειών* που αποζητούν συστηματική προώθηση των προϊόντων τους και επίτευξη κέρδους.

Τέλος, σημειώνεται η δημιουργία ενός συστήματος επιβράβευσης των *εθελοντών* βάσει ποιότητας καταγραφών, με μορφή πόντων για εξαργύρωση στο υποσύνολο αυτό των *εταιρειών*.

### 1.2 Επισκόπηση του λογισμικού

Αυτό το section δίνει μια επισκόπηση του συνολικού συστήματος. Για το σύστημα θα αναδειχθεί πλήρως πως αλληλεπιδρά με άλλα συστήματα και θα εισαχθούν οι απαιτήσεις αυτού. Επίσης μέσω διαγραμμάτων θα αποφανθούν πλήρως και οι αλληλεπιδράσεις των stakeholders με αυτό καθώς και οι περιπτώσεις χρήσης αυτών. Παράλληλα γίνονται ευδιάκριτοι τόσο οι περιορισμοί όσο και οι υποθέσεις για το σύστημα.

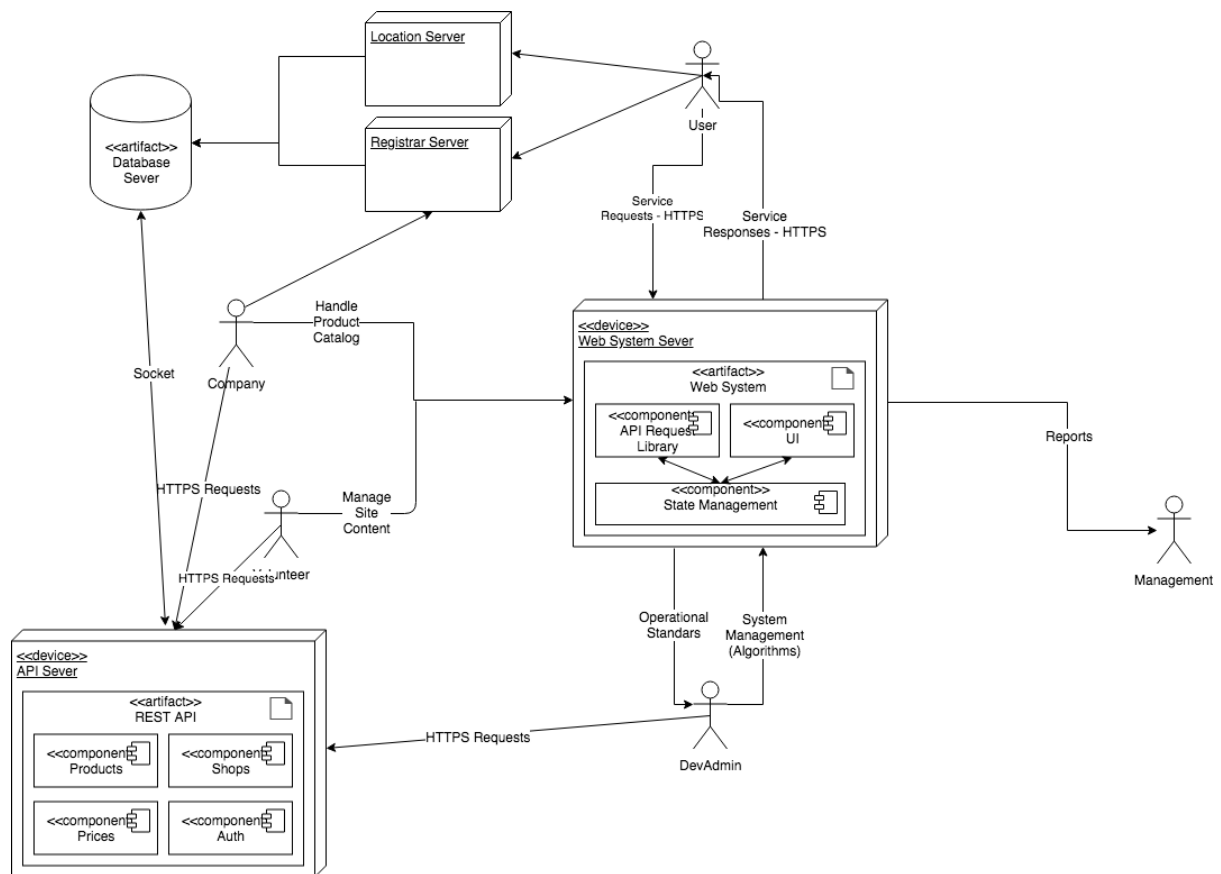
Συγκεκριμένα το section 1.2, γίνεται δημιουργία ενός UML context diagram που παρουσιάζει το σύστημα ως μία high-level διαδικασία και δείχνει τις σχέσεις με εξωτερικά entities (άλλα συστήματα, εξωτερικά data stores κλπ). Αντιθέτως δεν παρέχει καμία πληροφορία σχετικά με timing, sequencing ή synchronization διαδικασιών και άρα δεν πρέπει να συγχέεται με κάποιου είδους flowchart.

Χρησιμοποιήθηκε γιατί:

1. Αναδεικνύει πλήρως την επισκόπηση του συστήματος και τα όρια του
2. Καμία τεχνική γνώση δεν απαιτείται για την κατανόηση του
3. Εύκολη επέκταση του σε περίπτωση ανάγκης
4. Ευνοεί όλους τους stakeholders, business analysts, data analysts κλπ

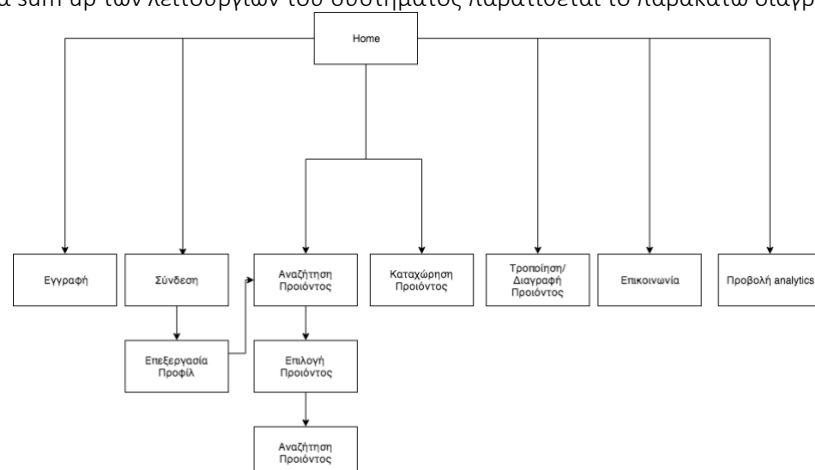
Οι εμπλεκόμενοι Δράστες (Actors) στο σύστημα είναι:

- Web System Server
- Database Server
- Location Server
- Registrar Server
- API Server



- (Web System Server): Το σύστημα είναι GPS-based web application που βοηθάει στην αναζήτηση, σύγκριση, αξιολόγηση προϊόντων και στην εύρεση κοντινότερων καταστημάτων βάσει της τοποθεσίας του χρήστη.
- (API Server): Οι εθελοντές καταγράφουν τις απαραίτητες πληροφορίες των προϊόντων επιλεγμένων εταιρειών. Όσες επιλέξουν το πακέτο προσφορών έχουν δυνατότητες σχολιασμού και τροποποίησης των πληροφοριών. Γενικά οι τελικά διαμορφωμένες πληροφορίες λειτουργούν ως βάση για αποτελέσματα αναζήτησης που εμφανίζονται στους χρήστες. Ο διαχειριστής μπορεί επίσης να κάνει χρήση του API Server με σκοπό την ομαλή διαχείριση του συστήματος και την ακρίβεια των πληροφοριών. Επίσης η χρήση του API δίνει δυνατότητες χρήσης των προϊόντων που θα υπάρχουν στο αποθετήριο, από άλλες εφαρμογές. (επικοινωνία μεταξύ τους)
- (Database Server): Επιπροσθέτως, το σύστημα χρειάζεται Internet και GPS σύνδεση για το login χρηστών και εταιρειών καθώς και την εμφάνιση αποτελεσμάτων αναζήτησης. Όλες οι πληροφορίες διατηρούνται στο Database Server.

Επιπροσθέτως για sum up των λειτουργιών του συστήματος παρατίθεται το παρακάτω διάγραμμα



### 1.3.1 Διεπαφές με εξωτερικά συστήματα και εφαρμογές λογισμικού

Η web πλατφόρμα μας επικοινωνεί με το GPS application – Location Server με σκοπό την άντληση των γεωγραφικών δεδομένων της τοποθεσίας του χρήστη και την visual αναπαράσταση αυτού, και με το Database Server με σκοπό την άντληση πληροφοριών για την τοποθεσία των καταστημάτων.

Η επικοινωνία μεταξύ του Database Server και του API Server περιλαμβάνει τις διαδικασίες ανάγνωσης (get) και τροποποίησης δεδομένων, ενώ η επικοινωνία μεταξύ του Database Server και του Web System Server περιλαμβάνει μόνο την διαδικασία ανάγνωσης (get).

#### Web Client

- ✓ Web Browser (Any Web Browser)

#### Web Server

- ✓ Jetty

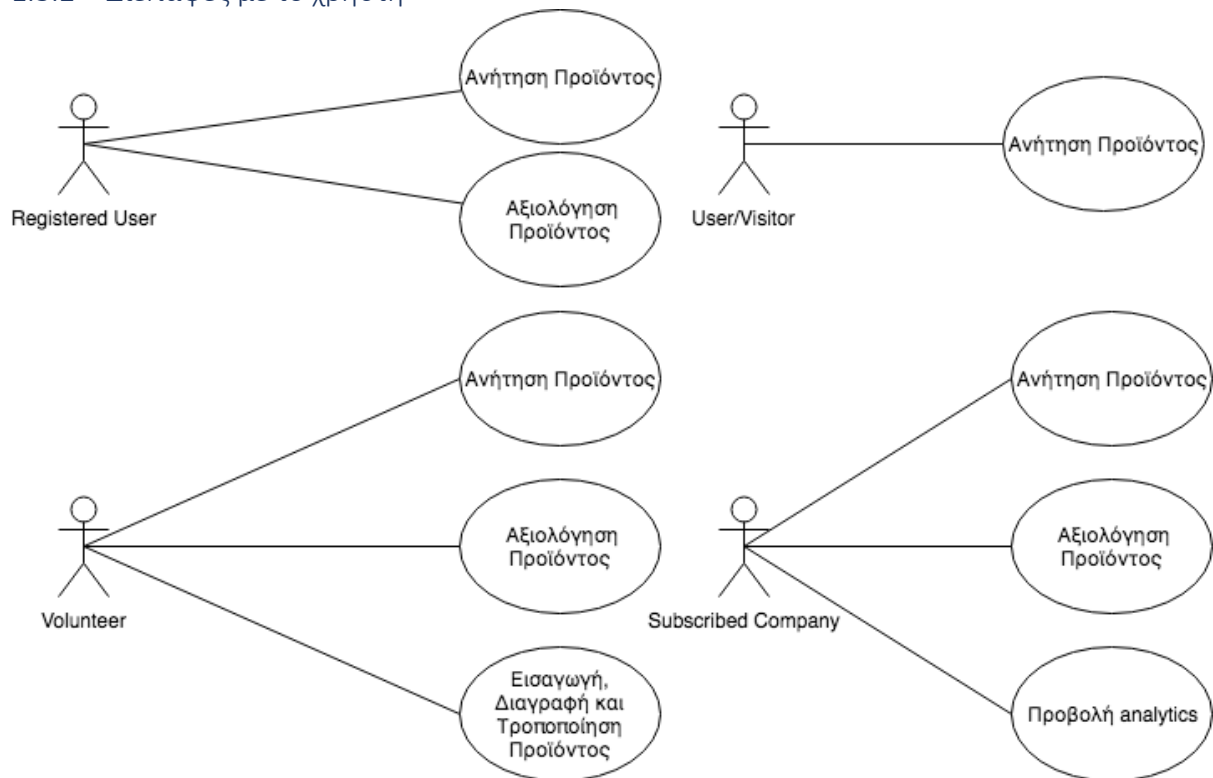
#### Database

- ✓ MySQL Database Server

#### Development

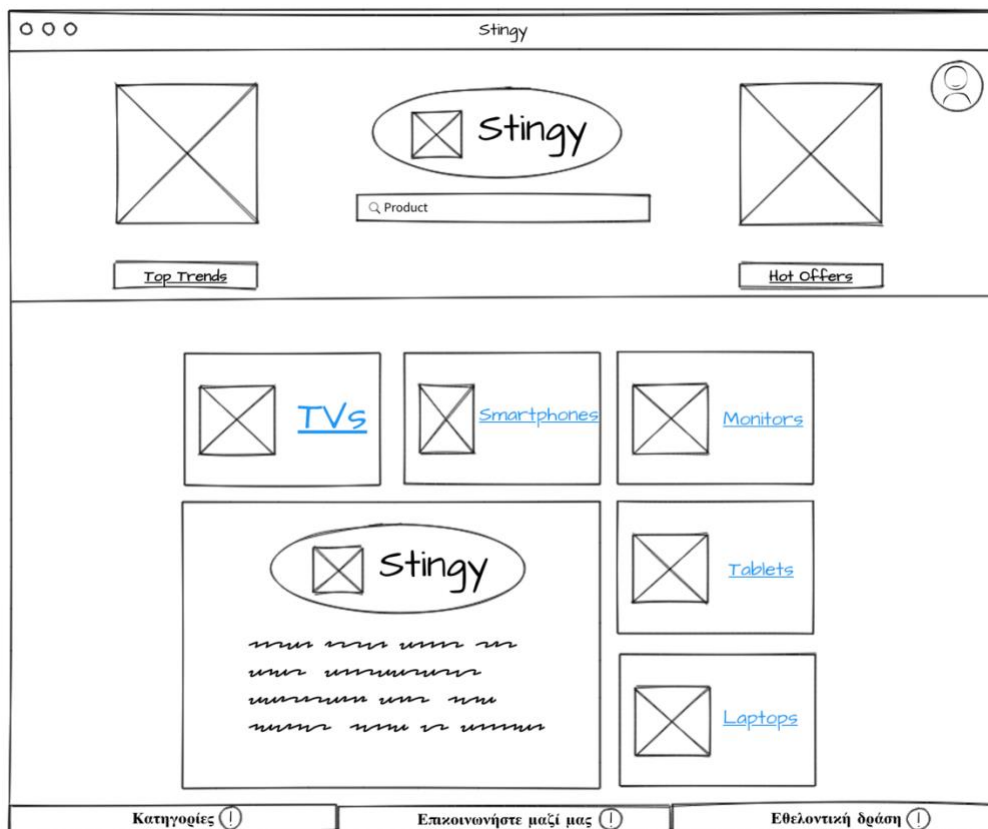
- ✓ Java 8 in Eclipse IDE

### 1.3.2 Διεπαφές με το χρήστη

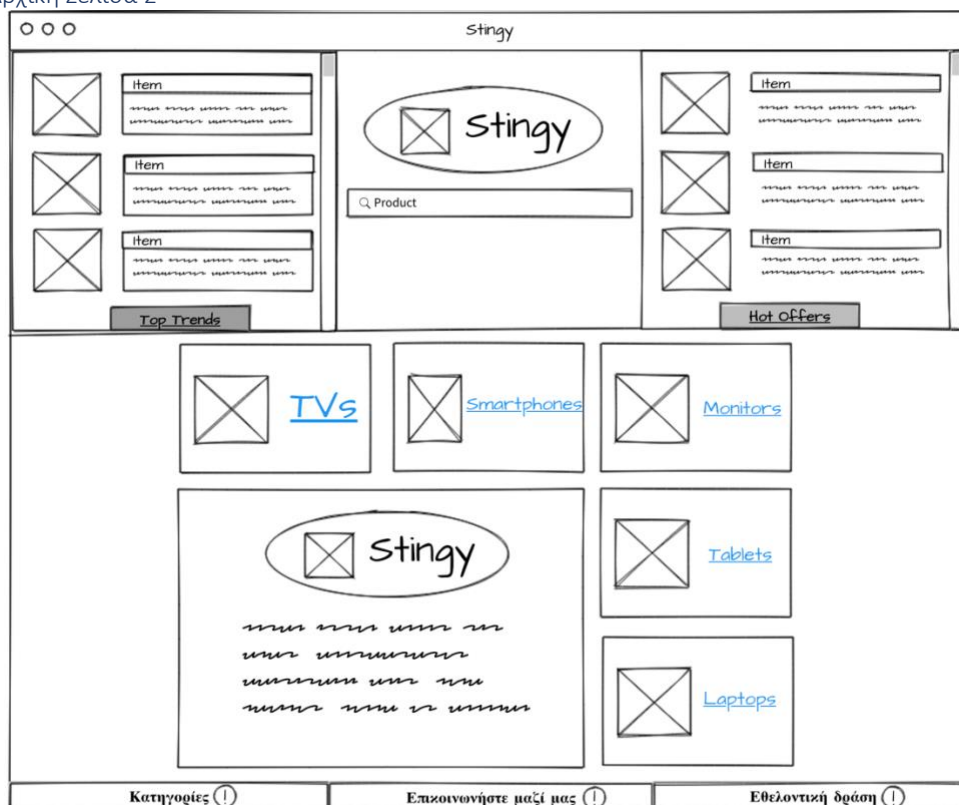


Ένα από τα πλεονεκτήματα του συστήματος είναι τα user interface του, δεδομένου ότι είναι εξαιρετικά λειτουργικά, εύκολα να χρησιμοποιηθούν και μπορούν να αντιμετωπιστούν από τους χρήστες ακόμα και με ελάχιστη γνώση και εμπειρία σχετικά με τη χρήση υπολογιστών. Τα user interfaces λοιπόν περιλαμβάνουν

#### 1.3.2.1 Αρχική Σελίδα



### 1.3.2.2 Αρχική Σελίδα 2



### 1.3.2.3 Είσοδος και Εγγραφή Χρήστη

The wireframe shows a web browser window titled 'Stingy'. The header includes the 'Stingy' logo, a search bar with the placeholder 'Product', and a user profile icon. The main content area features a central form with two tabs: 'Log in' and 'Sign up'. The 'Log in' tab has fields for 'Email/Username\*' and 'Password\*', a 'Forgot your Password?' link, a 'Keep me signed in' checkbox, and an 'Enter' button. The 'Sign up' tab has fields for 'Email/Username\*', 'Password\*', and 'Confirm Password\*', a checkbox for 'By signing up, I accept the Terms & Privacy policies', and a 'Register' button. The footer contains three links: 'Κατηγορίες', 'Επικοινωνήστε μαζί μας', and 'Εθελοντική δράση'.

### 1.3.4.4 Αναζήτηση Προϊόντων

The wireframe shows a web browser window titled 'Stingy'. The header includes the 'Stingy' logo, a search bar with the placeholder 'Product', and a user profile icon. The main content area is divided into a left sidebar and a right main area. The sidebar, titled 'Filters', includes sections for 'Categories' (with three checkboxes), 'Price' (with a slider from 0€ to 1500€), 'Distance' (with a slider from 0 km to 15 km), 'Average Customer Review' (with a star rating of 4 stars and up), and 'Brand' (with three checkboxes). The main area displays a grid of four product cards. Each card has a placeholder image, a title 'Item', a price (25€, 500€, 750€, and 760€ respectively), and a star rating. Each card also has a checkbox labeled 'Compare it / Add to Favourites'. The footer contains three links: 'Κατηγορίες', 'Επικοινωνήστε μαζί μας', and 'Εθελοντική δράση'.

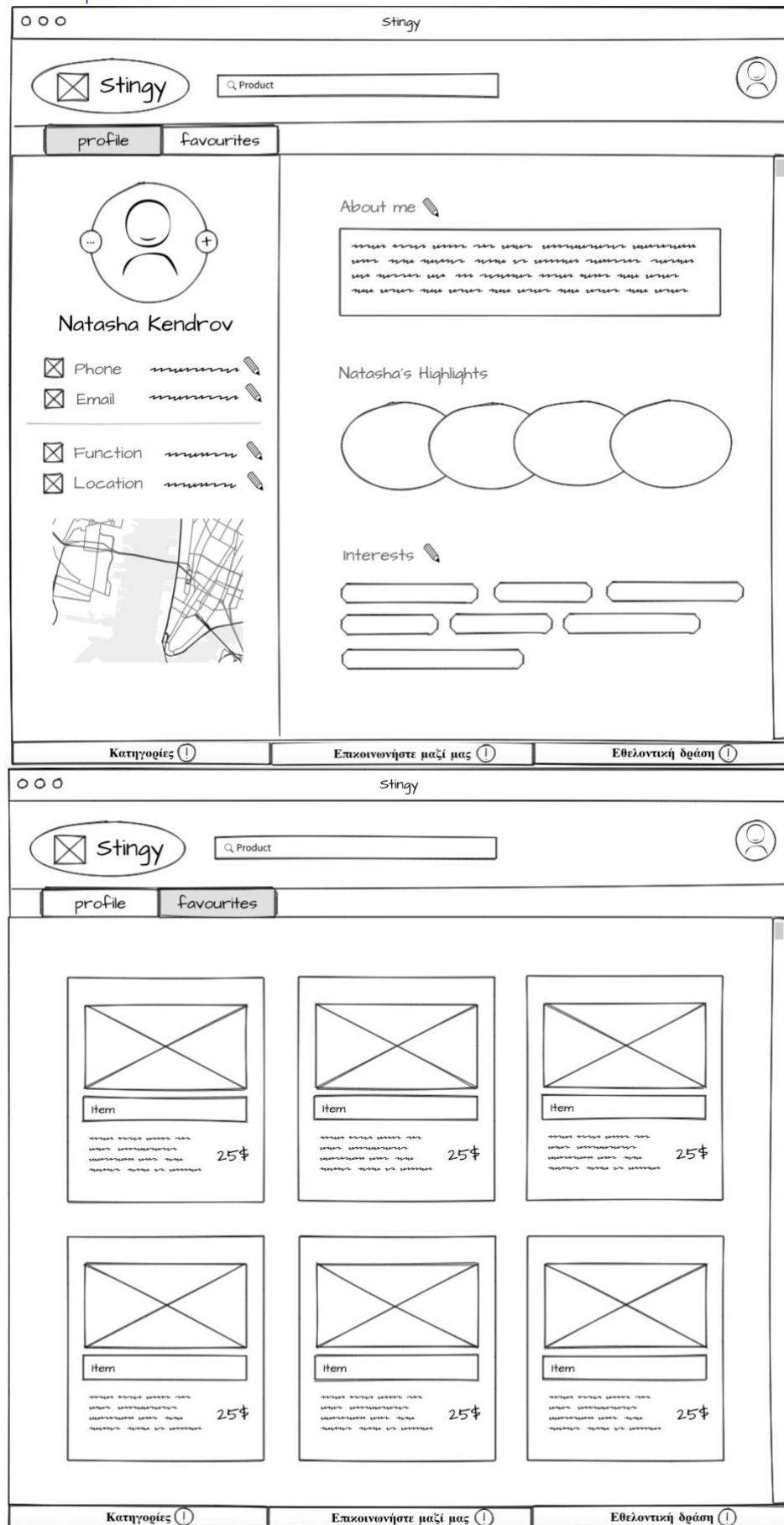
### 1.3.4.5 Προβολή Προϊόντος

The wireframe shows a product page for 'Stingy'. At the top is a navigation bar with the 'Stingy' logo, a search bar labeled 'Q Product', and a user profile icon. Below the navigation bar, there are two main columns. The left column features a large placeholder for a product image with a diagonal 'X' and a smaller carousel of images below it. Above the main image is a link 'Compare it / Add to Favourite'. The right column contains product details: 'PRODUCT TITLE' with a 5-star rating and 'BY PRODUCER NAME', 'PRICE: 500\$', 'COLOR' with three colored squares, 'CAPACITY' with '250 GB' and '500 GB' buttons, and a 'DESCRIPTION' section with placeholder text and a 'Show More' link. Below the product details, there are two sections: 'Customer Questions & Answers' with a search bar and a table of questions/answers, and 'XXX Customer Reviews' with a 4-star rating and a 'Write a review' button. At the bottom is a footer with three links: 'Κατηγορίες', 'Επικοινωνήστε μαζί μας', and 'Εθελοντική δράση'.

### 1.3.4.5 Φόρμα Επικοινωνίας

The wireframe shows a contact form for 'Stingy'. At the top is a navigation bar with the 'Stingy' logo, a search bar labeled 'Q Product', and a user profile icon. Below the navigation bar, there is a large text input field labeled 'Contact with us'. Below this is a form with a title 'Choose Subject\*' and a dropdown menu. Below the dropdown is a text input field labeled 'Message\*'. Below the message field are four text input fields labeled 'Name\*', 'Email\*', 'Phone Number\*', and 'Type your phone number..'. To the right of these fields is a large placeholder for a product image with a diagonal 'X'. At the bottom of the form is a 'Send >' button. At the bottom of the page is a footer with three links: 'Κατηγορίες', 'Επικοινωνήστε μαζί μας', and 'Εθελοντική δράση'.

#### 1.3.4.5 Σελίδα user profile



### 1.3.3 Διεπαφές με υλικό

Μια έμμεση διεπαφή υλικού:

- **Hardware χρηστών:**
  - ο Προσωπικοί υπολογιστές, laptops ή smartphones που χρησιμοποιούνται από τελικούς χρήστες, εθελοντές και εταιρείες
- **Network:**
  - ο **Internet**, που είναι το παγκόσμιο network που χρησιμοποιείται για όλες τις απαιτήσεις.
  - ο **Local Area Network**, που είναι το τοπικό network που χρησιμοποιείται από τους εθελοντές για άμεση επικοινωνία και ανταλλαγή αρχείων.

Επειδή η πλατφόρμα δεν έχει κάποιο αναγκαστικά σχεδιασμένο hardware για την λειτουργία της, μπορούμε να θεωρήσουμε ότι δεν υπάρχει άμεση διεπαφή υλικού. Η διαχείριση της φυσική τοποθεσία GPS γίνεται μέσω του GPS application στο εκάστοτε hardware χρήστη ενώ την διαχείριση της σύνδεσης του υλικού στον database server αναλαμβάνει το αντίστοιχο λειτουργικό σύστημα του hardware χρήστη και του web system server.

### 1.3.4 Διεπαφές επικοινωνιών

Η επικοινωνία μεταξύ των διαφόρων κομματιών του συστήματος είναι απαραίτητη εφόσον εξαρτώνται το ένα από το άλλο. Παρόλα αυτά με ποιον τρόπο επιτυγχάνεται η επικοινωνία αυτή δεν είναι ζωτικής σημασίας για το σύστημα και επομένως γίνεται διαχείριση αυτή από τα αντίστοιχα λειτουργικά συστήματα για την web πλατφόρμα μας.

- ✓ Σημειώνεται η χρήση standard πρωτοκόλλων HTTP/HTTPS.
- ✓ Web browser άδεια
- ✓ Network Server TCP/IP επικοινωνία με το website server
- ✓ JDBC επικοινωνία με την βάση δεδομένων

## 2. Αναφορές – πηγές πληροφοριών

Το παρόν έγγραφο αναφέρεται ή συμμορφώνεται με τα ακόλουθα έγγραφα:

- Διακήρυξη Ανοικτού Διεθνούς Ηλεκτρονικού Διαγωνισμού για το έργο: «Προδιαγραφές των αναγκαίων πληροφοριακών συστημάτων»
  - [https://www.eetaa.gr/diadikasiaes/04072018\\_sxedio\\_diakirixis\\_aploysteysis\\_diadikasion.pdf](https://www.eetaa.gr/diadikasiaes/04072018_sxedio_diakirixis_aploysteysis_diadikasion.pdf)
- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- Προδιαγραφές Άσκησης Εργαστηρίου
  - *SRS\_template\_2018\_v2*
- OPEN Process Framework (OPF) Conventions:
  - Use Case Modeling Guidelines, που υπαγορεύει τα guidelines που χρησιμοποιήθηκαν για την ανάπτυξη των σεναρίων χρήσης συγκεκριμενοποιώντας τις λειτουργικές απαιτήσεις στο παρόν έγγραφο
  - System Requirements Specification Content and Format Standard, που υπαγορεύει το περιεχόμενο και την δομή του παρόντος εγγράφου
  - System Requirements Specification Inspection Checklist, που χρησιμοποιείτε κατά την διάρκεια επιθεώρησης του παρόντος εγγράφου
  - System Requirements Specification Template, που παρέχει τον σκελετό του παρόντος εγγράφου.

Επιπροσθέτως γίνεται άντληση πληροφοριών από υπάρχουσες υπηρεσίες στον χώρο αναζήτησης τεχνολογικών προϊόντων στο διαδίκτυο.



### 3. Προδιαγραφές απαιτήσεων λογισμικού

#### 3.1 Εξωτερικές διεπαφές

Στο σύστημα χρησιμοποιούνται εξωτερικές διεπαφές:

- Google Maps:
  - ο Όταν ο εθελοντής εισέρχεται στην σελίδα καταχώρησης προϊόντος, συμπληρώνει εκτός των άλλων την γεωγραφική τοποθεσία του shop κάποιας εταιρείας. Συγκεκριμένα θα δίνεται ένας χάρτης Google μεσαίου μεγέθους με τις βασικές λειτουργίες zoom-in / zoom-out και pin επιθυμητών σημείων. Θα απαιτείται από τον εθελοντή η επιλογή του σημείου του χάρτη όπου βρίσκεται το κατάστημα. Στην συνέχεια μέσω back end αυτό μεταφράζεται σε γεωγραφικά σημεία τύπου float τα οποία καταχωρούνται στην βάση μαζί με τα υπόλοιπα στοιχεία, μετά την επιτυχή καταχώρηση της εγγραφής.
  - ο Αντίστοιχη δυνατότητα δίνεται στον χρήστη, με την έννοια του καθορισμού της τοποθεσίας του (με παρόμοιο τρόπο), στο edit profile. Συνδυαστικά με τις τοποθεσίες των καταστημάτων μπορούν να χρησιμοποιηθούν ως extra φίλτρο αναζήτησης (πχ. Καταστήματα σε απόσταση 12 km)
- Βάση Δεδομένων MySQL: Γίνεται αποθήκευση δεδομένων μέσω UPDATE, PUT, PATCH από το backend μέσω κατάλληλων φορμών που τις διαχειρίζεται ο εθελοντής. Διαφορά με εγγραφόμενο χρήστη είναι ότι το κουμπί τροποποίηση δίπλα από τα αντίστοιχα χαρακτηριστικά των προϊόντων είναι ανενεργό. Για το INSERT θα εμφανίζεται extra φόρμα (ως popup) συμπλήρωση νέου προϊόντος και των χαρακτηριστικών του (την τοποθεσία συγκεκριμένα όπως αναλύθηκε προηγουμένως). Επιπλέον η βάση παρέχει τα κατάλληλα κλειδιά/ευρετήρια ώστε να προσδιορίζονται αποδοτικά όλα της τα tuples καθώς και triggers & views , ώστε να γίνονται αυτόματα ενημερώσεις δεικτών/μετρητών και να αυτοματοποιούνται ορισμένοι χρήσιμοι συνδυασμοί δεδομένων.
- STMP Server: Ο χρήστης μέσω του contact form θα μπορεί να στείλει email στους admin. Το email θα εμφανίζεται τόσο στην σελίδα των διαχειριστών όσο και στα email τους. Ακόμα κατά την εγγραφή ενός χρήστη θα του αποστέλλεται email επιβεβαίωσης εγγραφής. Τέλος, μέσω αυτής της διεπαφής θα αποστέλλεται στο χρήστη email σε περίπτωση που έχει ξεχάσει τον κωδικό του.

#### 3.2 Λειτουργίες: περιπτώσεις χρήσης

##### Use case Index

ID	Use Case	PRIMARY	COMPLEXITY	PRIORITY
1	Ενημέρωση τιμής προϊόντος από εθελοντή	Volunteer	low	1
2	Διαγραφή καταστήματος από εθελοντή	Volunteer	high	3
3	Απλός χρήστης κάνει αναζήτηση προϊόντος	User	med	1
4	Ενημέρωση τιμής προϊόντος από εταιρεία	Company	med	2

5	Διαχείριση λογαριασμών από τον διαχειριστή	Dev-Admin	med	1
---	--	-----------	-----	---

### 3.2.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: (Ενημέρωση τιμής προϊόντος από εθελοντή)

#### 3.2.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Ένας actor έχει υποχρεώσεις απέναντι στο σύστημα (δεδομένα εισόδου) αλλά και προσδοκίες από αυτό (δεδομένα εξόδου)

Volunteer: Εγγεγραμμένος στο σύστημα User, με δικαιώματα – όπως έχουν αυτά καθοριστεί από τον dev-admin – εισαγωγής, τροποποίησης και διαγραφής χαρακτηριστικών των προϊόντων του παρατηρητηρίου.

Dev-Admin

#### 3.2.1.2 Προϋποθέσεις εκτέλεσης

Απαραίτητες Προϋποθέσεις:

- Ο Web System Server να είναι ενεργός
- Ο απλός user πρέπει να έχει κάνει την πρώτη εγγραφή
- Ο εγγεγραμμένος πλέον user πρέπει να έχει επικοινωνήσει το ενδιαφέρον του στον dev-admin για ενασχόληση εθελοντικά και βοηθητικά με τις δραστηριότητες της πλατφόρμας.
- Ομαλή λειτουργία συστήματος
- Επιτυχημένη εισαγωγή διαπιστευτηρίων και σύνδεση στο σύστημα
- Για την εισαγωγή μετέπειτα δεν υπάρχει κάποια προϋπόθεση. Αντιθέτως για την τροποποίηση ή την διαγραφή απαιτείται η ύπαρξη του προϊόντος στο αποθετήριο

#### 3.2.1.3 Περιβάλλον εκτέλεσης

Γενικότερα η συμπεριφορά του συστήματος είναι εφικτή από την σκοπιά των συνολικών λειτουργιών. Υπάρχουν περιορισμένες εφαρμογές του λογισμικού και άρα η συμπεριφορά δεν πρόκειται να αλλάξει για αρκετά χρόνια. Επιπλέον το λογισμικό μπορεί να αναβαθμιστεί εάν κριθεί απαραίτητο. Το περιβάλλον εκτέλεσης παρέχεται από τους hosts και ο πελάτης χρειάζεται αποκλειστικά έναν browser που είναι δωρεάν με οποιοδήποτε λειτουργικό σύστημα και κατ' επέκταση συμβατό με σχεδόν όλα τα hardware του χρήστη.

Το περιβάλλον εκτέλεσης είναι η διαδικτυακή διεπαφή χρήστη (Web System Server) που αλληλεπιδρά με το API Server που αλληλεπιδρά με τη βάση δεδομένων (Database Server)

#### 3.2.1.4 Δεδομένα εισόδου

Δεδομένα εισόδου αποτελούν τα fields που τροποποιούνται από τον εθελοντή οποιουδήποτε υπάρχοντος προϊόντος. Συγκεκριμένα:

- Name (sN)
- Tags (sT)
- Description (sD)
- Extra Data (sED)
- Price (sP) = product's price

Στο συγκεκριμένο Use Case μελετάμε την τροποποίηση τιμής (sP) που χωρίς καθορισμό παραμέτρων η ασφάλεια των προϊόντων και κατ' επέκταση του συστήματος υπονομεύεται.

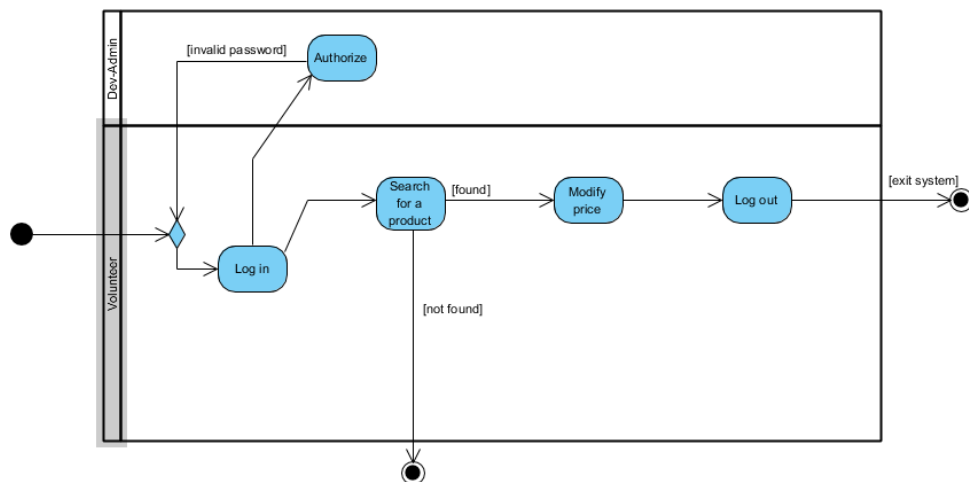
#### 3.2.1.5 Παράμετροι

- #tags >= 2
- If(Old Price != nil) New price = -80%Old Price έως Old Price (σε περιπτώσεις προσφορών)

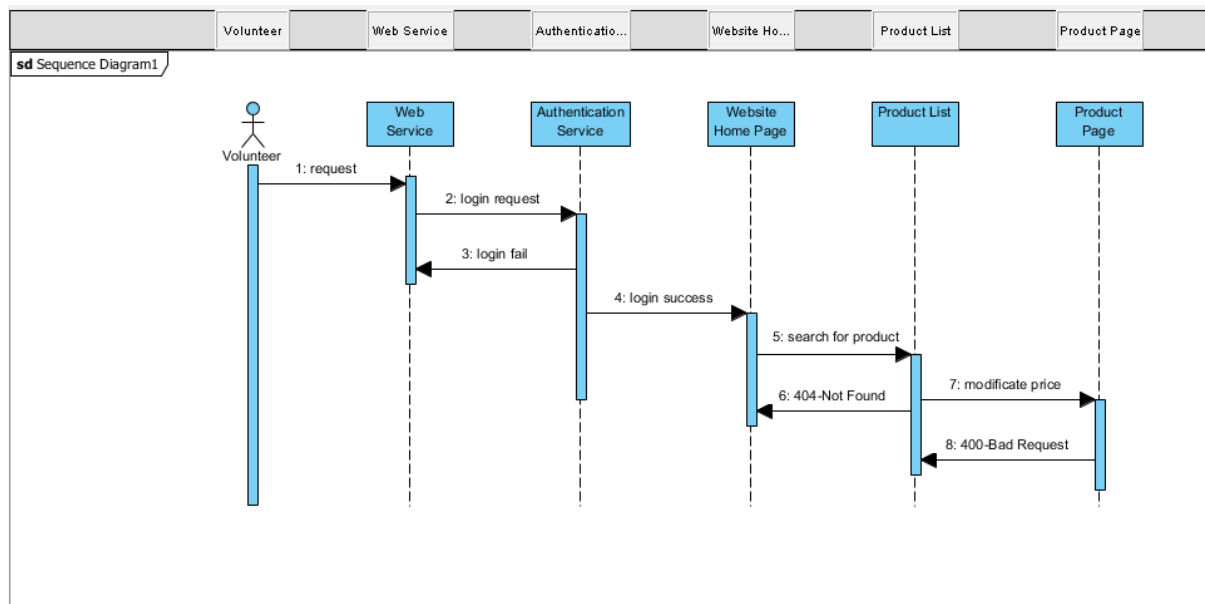
### 3.2.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Ο χρήστης εισέρχεται στο σύστημα, κάνει εγγραφή και ζητάει να γίνει εθελοντής πατώντας στο αντίστοιχο κουμπί του homepage.
2. Ο χρήστης εισέρχεται στο σύστημα και εφόσον δώσει σωστά τα διαπιστευτήρια του, του δίνεται η δυνατότητα επεξεργασίας στο σύστημα.
3. Ο χρήστης επιλέγει τη λειτουργία αναζήτησης και συμπληρώνει το πεδίο με τα αντίστοιχα κλειδιά για εύρεση του προς τροποποίηση προϊόντος
4. Εφόσον βρεθεί το προϊόν που θέλει να ενημερώσει, καταχωρεί τα νέα στοιχεία.
5. Ο εθελοντής αποσυνδέεται από το σύστημα.

#### Activity UML



#### Sequence UML



### 3.2.1.7 Δεδομένα εξόδου

Δεδομένα εξόδου αποτελούν το τροποποιημένα από τον εθελοντή fields υπάρχοντος προϊόντος. Συγκεκριμένα:

- Name (sKN)
- Tags (sKT)
- Description (sKD)

- *Extra Data (sEKD)*
- *Price (sKP) = computed search key of product's price*

Στο συγκεκριμένο Use Case εξετάζουμε τροποποιημένη τιμή (sKP) ως δεδομένο εξόδου.

### 3.2.1.8 Παρατηρήσεις

- Το βήμα 1 της αλληλουχίας ενεργειών παραλείπεται για τα activity και sequence model. Ο λόγος είναι ότι ο εθελοντής ως actor, αποτελεί κατά βάση το αποτέλεσμα του βήματος 1 και καθολικά χρησιμοποιείται σε οποιοδήποτε έγγραφο.
- Η τελευταία προϋπόθεση του 3.2.1.2 αν και προφανής αναφέρεται γιατί μπορεί παλιότερα να υπήρχε κάποιο προϊόν που πλέον δεν υπάρχει, και λόγω κακής διαχείρισης δεδομένων της βάσης, το url `https://.../product=qXXX` να επέστρεφε το view page του προϊόντος, ενώ μέσω ομαλής χρήσης του UI να μην δινόταν η δυνατότητα ανεύρεσης του.

## 3.2.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: (Διαγραφή καταστήματος από εθελοντή)

### 3.2.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Volunteer: Εγγεγραμμένος στο σύστημα User, με δικαιώματα – όπως έχουν αυτά καθοριστεί από τον dev-admin – εισαγωγής, τροποποίησης και διαγραφής χαρακτηριστικών των προϊόντων του παρατηρητηρίου.

Dev-Admin

### 3.2.2.2 Προϋποθέσεις εκτέλεσης

Απαραίτητες Προϋποθέσεις:

- Ο Web System Server να είναι ενεργός
- Ο απλός user πρέπει να έχει κάνει την πρώτη εγγραφή
- Ο εγγεγραμμένος πλέον user πρέπει να έχει επικοινωνήσει το ενδιαφέρον του στον dev-admin για ενασχόληση εθελοντικά και βοηθητικά με τις δραστηριότητες της πλατφόρμας.
- Ομαλή λειτουργία συστήματος
- Επιτυχημένη εισαγωγή διαπιστευτηρίων και σύνδεση στο σύστημα
- Για την διαγραφή απαιτείται η ύπαρξη της εταιρείας στο αποθετήριο
- Η ολοκλήρωση (ή όχι) της διαγραφής λαμβάνει χώρα από τον διαχειριστή του συστήματος

### 3.2.2.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης είναι η διαδικτυακή διεπαφή χρήστη (Web System Server) που αλληλεπιδρά με το API Server που αλληλεπιδρά με την βάση δεδομένων (Database Server)

### 3.2.2.4 Δεδομένα εισόδου

- Click του button delete shop όταν ο εθελοντής βρίσκεται στο profile page της εταιρείας.

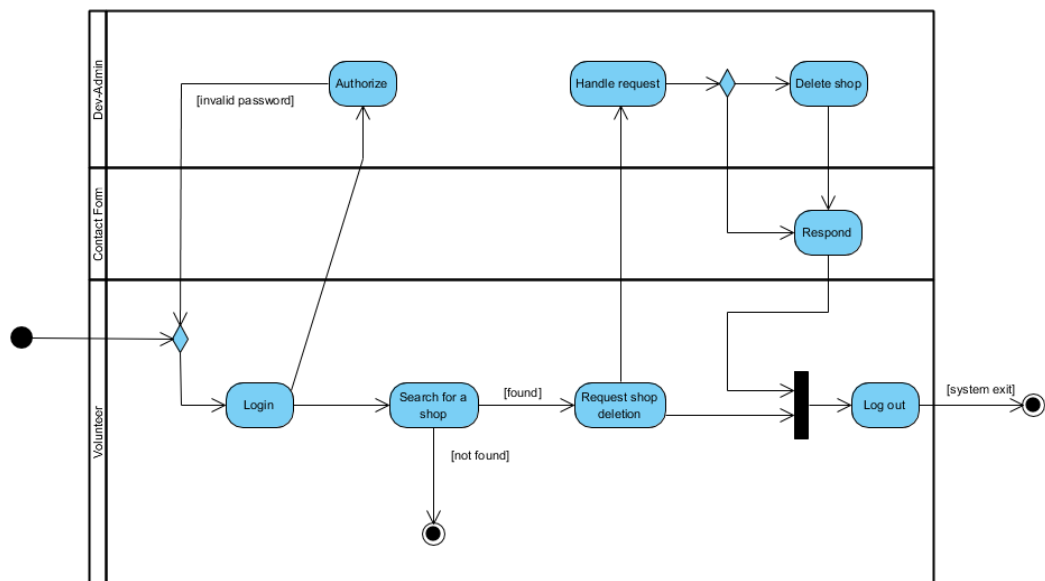
### 3.2.2.5 Παράμετροι

- Απο τεχνικής και πρακτικής απόψεως δεν υπάρχουν καθώς πρόκειται για ένα click. Παρόλα αυτά αν η ενέργεια της διαγραφής έγινε με κακόβουλο σκοπό ή από λάθος τότε από θεωρητικής απόψεως ο διαχειριστής του συστήματος πρέπει να ακυρώσει το αίτημα.

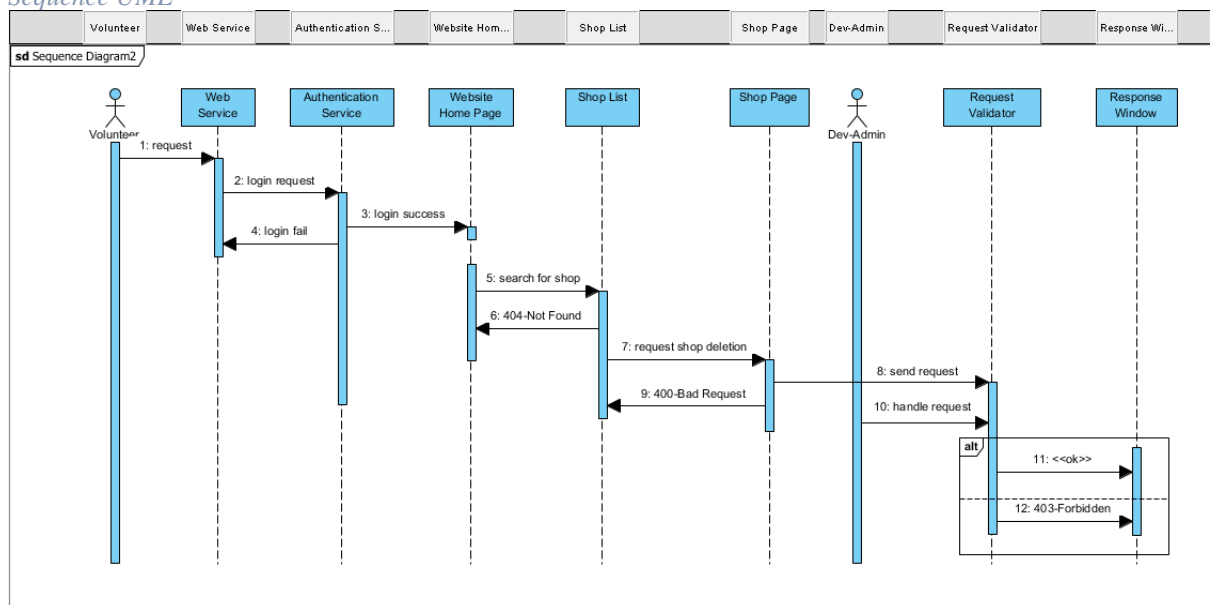
### 3.2.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Ο χρήστης εισέρχεται στο σύστημα, κάνει εγγραφή και ζητάει να γίνει εθελοντής πατώντας στο αντίστοιχο κουμπί του homepage.
2. Ο χρήστης εισέρχεται στο σύστημα και εφόσον δώσει σωστά τα διαπιστευτήρια του, του δίνεται η δυνατότητα επεξεργασίας στο σύστημα
3. Κάνει αναζήτηση του καταστήματος/εταιρείας που θέλει να διαγράψει.
4. Επιλέγει το profile page της εταιρείας και κάνει κλικ στο delete shop
5. Ο διαχειριστής λαμβάνει το αίτημα
6. Είτε διαγράφει το κατάστημα, είτε απαντάει αρνητικά στον εθελοντή/χρήστη.
7. Ο χρήστης αποσυνδέεται από το λογαριασμό του και φεύγει από το σύστημα.

## Activity UML



## Sequence UML



### 3.2.2.7 Δεδομένα εξόδου

- Σε περίπτωση έγκρισης του αιτήματος διαγραφής το παρατηρητήριο τιμών περιλαμβάνει 1 λιγότερο εταιρεία και k λιγότερα προϊόντα, ενώ στην αντίθετη περίπτωση δεν γίνεται καμία μεταβολή του συστήματος.

### 3.2.2.8 Παρατηρήσεις

- Τα στοιχεία που αφορούν μια εταιρεία είναι αρκετά sensitive, ο έλεγχος σε περιπτώσεις τροποποίησης αυτών καθίσταται δύσκολο εγχείρημα ακόμα και με την χρήση περιορισμών, οπότε δεν

προσφέρεται ως δυνατότητα στους εθελοντές για αποφυγή κακόβουλης χρήστης. Αντιθέτως προσφέρεται μόνο η δυνατότητα διαγραφής κάποιας εταιρείας κατόπιν έγκρισης από τον διαχειριστή.

- Η τελευταία προϋπόθεση του 3.2.2.2 αν και προφανής αναφέρεται γιατί μπορεί παλιότερα να υπήρχε κάποια εταιρεία που πλέον δεν υπάρχει, και λόγω κακής διαχείρισης δεδομένων της βάσης, το url `https://.../shop=qXXX` να επέστρεφε το προφίλ της εταιρείας, ενώ μέσω ομαλής χρήσης του UI να μην δινόταν η δυνατότητα ανεύρεσης της.

### 3.2.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: (Αναζήτηση προϊόντος από απλό χρήστη)

#### 3.2.3.1 Χρήστες (ρόλοι) που εμπλέκονται

User: Ο μόνος ρόλος που εμπλέκεται στην περίπτωση χρήσης είναι αυτός του απλού χρήστη είτε ως visitor είτε ως εγγεγραμμένος.

#### 3.2.3.2 Προϋποθέσεις εκτέλεσης

Απαραίτητες Προϋποθέσεις:

- Ο Web System Server να είναι ενεργός
- Ο απλός user πρέπει να έχει κάνει την πρώτη εγγραφή
- Να υπάρχουν  $\geq 1$  προϊόν στην λίστα προϊόντων ως αποτέλεσμα είτε αναζήτησης με keyboard είτε αναζήτησης με φίλτρων
- Να γίνει επιτυχημένη ανακατεύθυνση στην ιστοσελίδα του προϊόντος

#### 3.2.3.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης είναι η διαδικτυακή διεπαφή χρήστη (Web System Server) που αλληλεπιδρά με τη βάση δεδομένων (Database Server)

#### 3.2.3.4 Δεδομένα εισόδου

Δεδομένα εισόδου αποτελούν:

- Λέξεις κλειδιά σε περίπτωση αναζήτησης με keyboard (αυτά μπορεί να είναι κατηγορίες προϊόντων, ονόματα καταστημάτων, διεύθυνση καταστημάτων, κλπ)
- Σε περίπτωση φίλτρων δεν υπάρχει η παραπάνω ελευθερία στην αναζήτηση, αντιθέτως γίνεται πιο περιοριστικά σε επίπεδο UI:
  - Categories (sC): [laptops, desktops, tablets, TVs,...etc]
  - Price (sP): από 0 έως X \$
  - Distance (sD): από 0.0 έως X.Y km
  - Average Review (sAR): από 0 έως 5 αστέρια
  - Brand (sB): [Apple, Kotsovolos,...]

#### 3.2.3.5 Παράμετροι

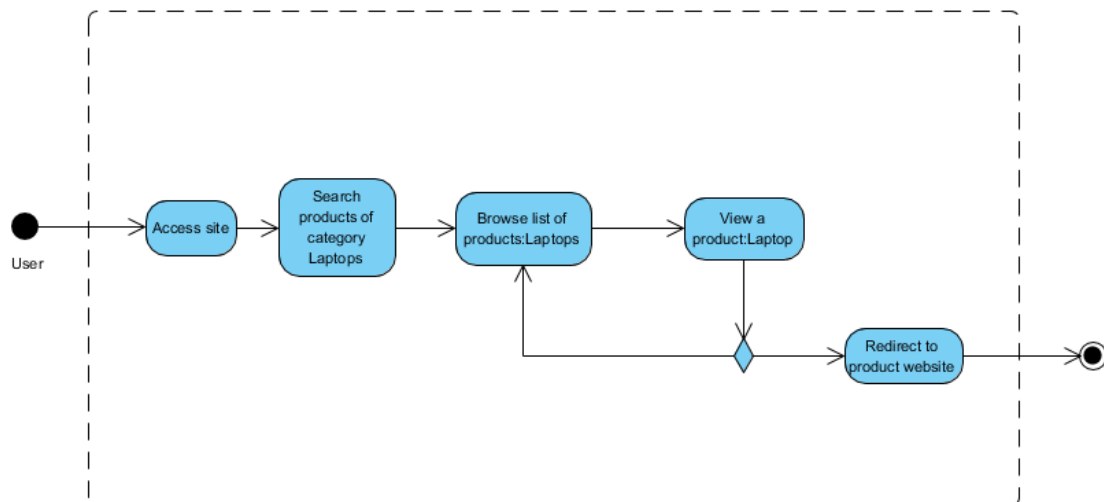
- Συνθήκες εγκυρότητας δεν υπάρχουν, αλλά με σκοπό η αναζήτηση να οδηγήσει σε τουλάχιστον  $\geq 1$  αποτέλεσμα ως λίστα προϊόντων
  - Στην περίπτωση της αναζήτησης με keyboard αρκεί να δοθεί κάποια εταιρεία, όνομα προϊόντος ή κατηγορία που να υπάρχει
  - Στην περίπτωση αναζήτησης με φίλτρα αρκεί να γίνει χρήση ενός φίλτρου (προφανές αφού τα όρια των φίλτρων έχουν καθοριστεί ανεξάρτητα βάσει των προδιαγραφών των προϊόντων) ή εστιασμένου συνδυασμού αυτών που να έχει νόημα,

#### 3.2.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

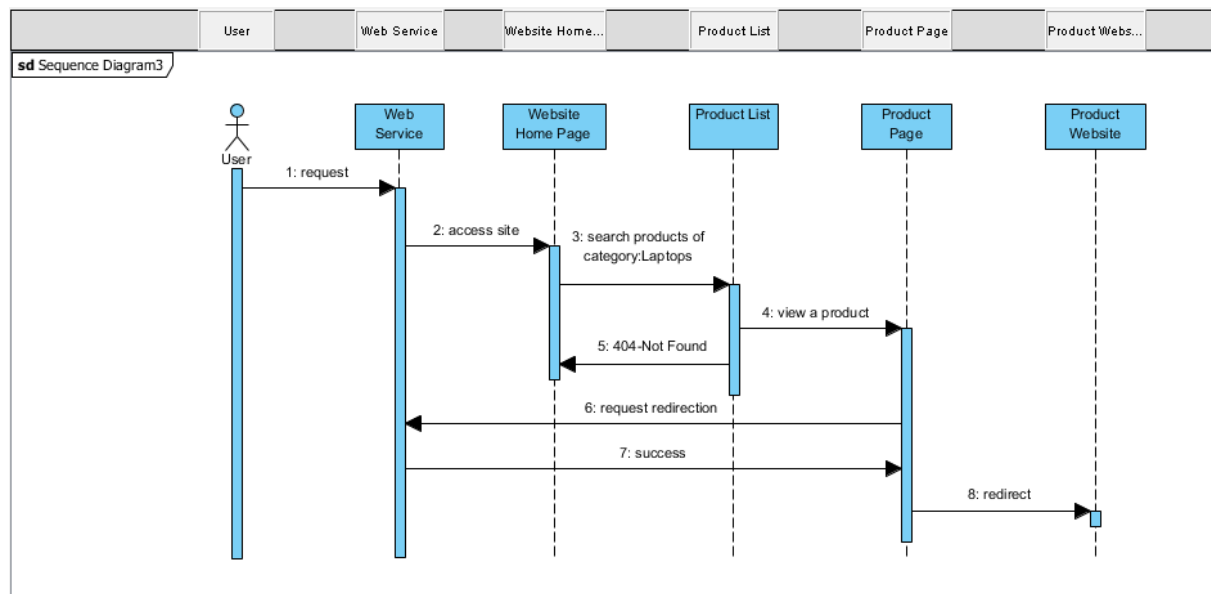
1. Ο χρήστης εισέρχεται στο σύστημα και είτε παραμένει visitor, είτε δίνει σωστά τα διαπιστευτήρια του είτε εγγράφεται για πρώτη φορά
2. Κάνει **αρχική** αναζήτηση προϊόντος μέσω keyboard (λέξεις κλειδιά κατηγορία, tags ή κατάσταση προϊόντος)

3. Περιηγείται σε όλα τα προϊόντα που του έχει εμφανίσει το σύστημα ενώ ταυτόχρονα με ενεργά κλικ (χωρίς επιπρόσθετο πάτημα σε κάποιο search button) στα αντίστοιχα φίλτρα αναζήτησης, μεταβάλλεται η λίστα αποτελεσμάτων.
4. Επιλέγει κάποιο προϊόν που τον ενδιαφέρει ή που θέλει να αξιολογήσει (επειδή μπορεί να το έχει αγοράσει στο παρελθόν) με σκοπό την εμφάνιση των αναλυτικών πληροφοριών αυτού)
5. Ο χρήστης έχει τη δυνατότητα είτε να επιστρέψει στην περιήγηση ώστε να δει και άλλα σχετικά προϊόντα (βήμα 3) είτε να γίνει ανακατεύθυνση στην ιστοσελίδα του καταστήματος που πουλάει το συγκεκριμένο προϊόν όπου και ολοκληρώνεται η αλληλεπίδραση με το σύστημα.

### Activity UML



### Sequence UML



### 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)

### 3.2.3.8 Παρατηρήσεις

- Η επιτυχία του παραπάνω use case μπορεί να θεωρηθεί ακόμα και σε περίπτωση αποτυχίας της τελευταίας προϋπόθεσης του 3.2.3.2, απλώς αυτή σημειώνεται για λόγους πληρότητας.

## 3.2.4 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 4 (Εταιρία έρχεται σε επαφή με τον διαχειριστή)

### 3.2.4.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι εμπλεκόμενοι ρόλοι είναι οι εγγεγραμμένες εταιρίες και ο διαχειριστής.

### 3.2.4.2 Προϋποθέσεις εκτέλεσης

### 3.2.4.3 Περιβάλλον εκτέλεσης

Αποτελεί διαδικτυακή διεπαφή χρήστη που αλληλεπιδρά με τη Βάση Δεδομένων.

### 3.2.4.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου είναι τα διαπιστευτήρια και η τιμή που θέλει η εταιρία να τροποποιήσει. Τα διαπιστευτήρια ελέγχονται από το σύστημα και αν είναι λάθος επιστρέφει σχετικό μήνυμα ενώ αν είναι σωστά δίνεται πρόσβαση στο σύστημα. Η τιμή δεν είναι δυνατόν να ελεγχθεί ούτε από το σύστημα ούτε από τον διαχειριστή.

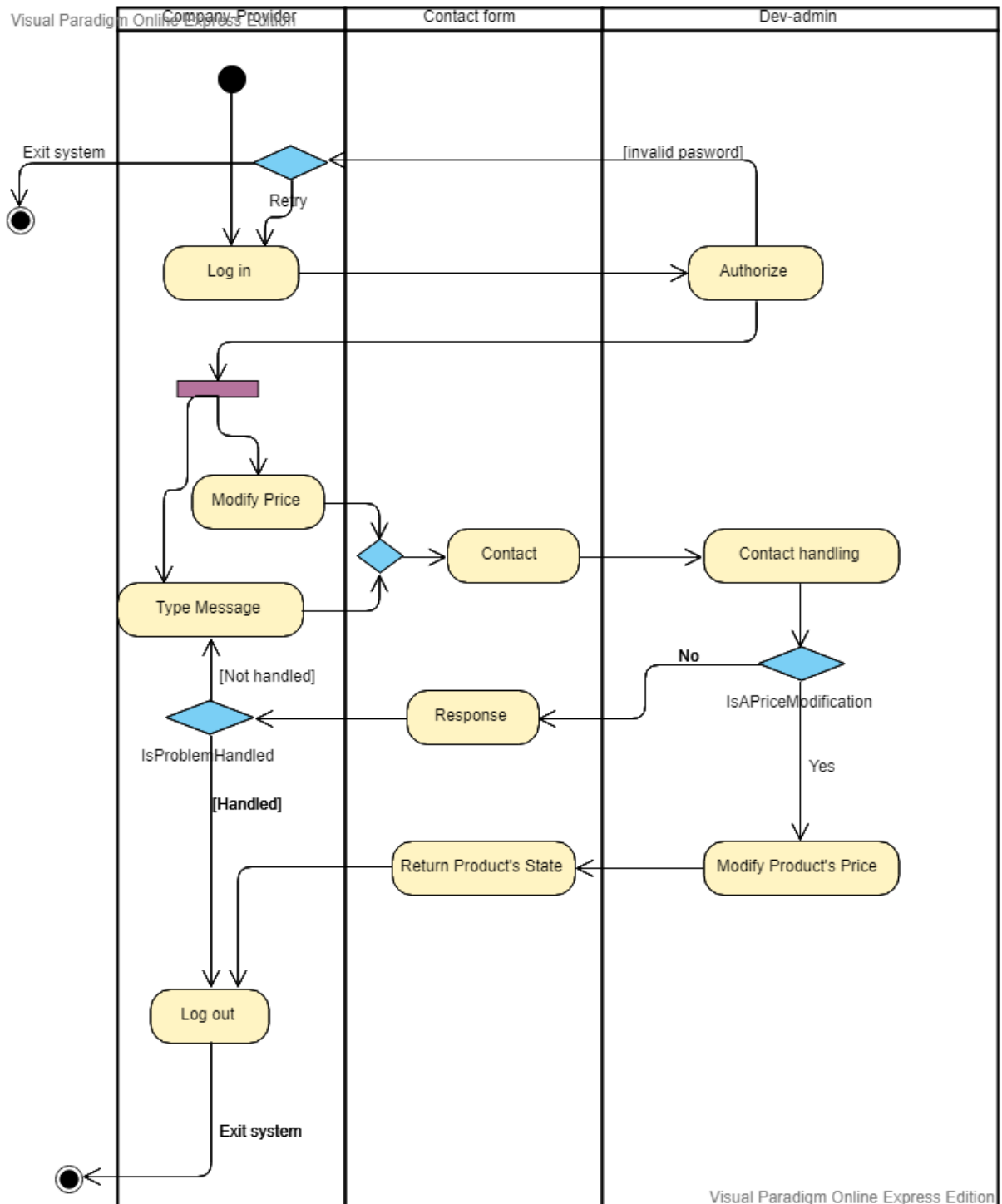
### 3.2.4.5 Παράμετροι

### 3.2.4.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

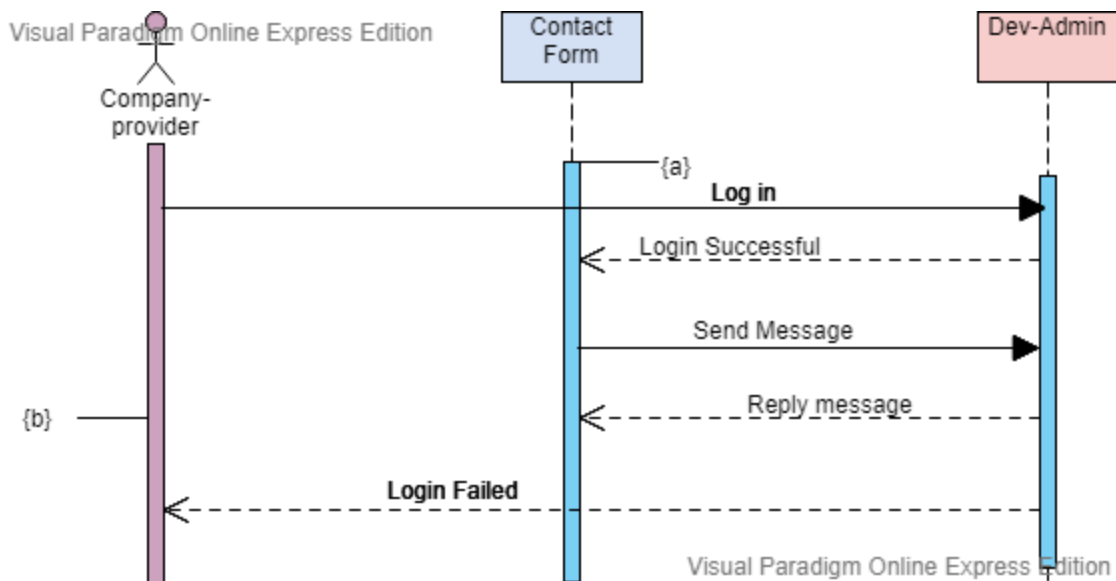
1. Η εταιρία εισέρχεται με τα διαπιστευτήρια της και εφόσον είναι σωστά μπορεί να συνεχίσει.
2. Η εταιρία κάνει αίτημα είτε για τροποποίηση της τιμής ενός προϊόντος είτε για αποστολή ενός μηνύματος.
3. Ο διαχειριστής στην περίπτωση που το αίτημα δεν είναι τροποποίηση τιμής επιστρέφει μια απάντηση στην εταιρία ενώ αν είναι τροποποίηση πραγματοποιεί την αλλαγή.
4. Η εταιρία αν το αίτημά της δεν έχει πραγματοποιηθεί μπορεί να το ξανακάνει αλλιώς κάνει αποσύνδεση από το σύστημα.

Activity UML





Sequence UML



#### 3.2.4.7 Δεδομένα εξόδου

#### 3.2.3.8 Παρατηρήσεις

### 3.2.5 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 5 (Ο διαχειριστής διαχειρίζεται λογαριασμούς χρηστών και εταιριών)

#### 3.2.5.1 Χρήστες (ρόλοι) που εμπλέκονται

Ο σταθερός ρόλος που εμπλέκεται στην περίπτωση αυτή είναι αυτός του διαχειριστή και αναλόγως το ποιος έχει κάνει το αίτημα μπορεί να εμπλακεί είτε ένας χρήστης/εθελοντής, είτε μια εταιρία.

#### 3.2.5.2 Προϋποθέσεις εκτέλεσης

Η συνθήκη που πρέπει να ισχύει είναι να είναι λειτουργικό το σύστημα.

#### 3.2.5.3 Περιβάλλον εκτέλεσης

Αποτελεί διαδικτυακή διεπαφή του διαχειριστή με τη Βάση Δεδομένων αναζητώντας, δημιουργώντας ή αφαιρώντας από αυτή, λογαριασμούς.

#### 3.2.5.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου ελέγχονται, από τον διαχειριστή και σε περίπτωση μη έγκυρων δεδομένων επιστρέφεται μηνύματα στον χρήστη ώστε να ξαναδώσει τα δεδομένα από την αρχή.

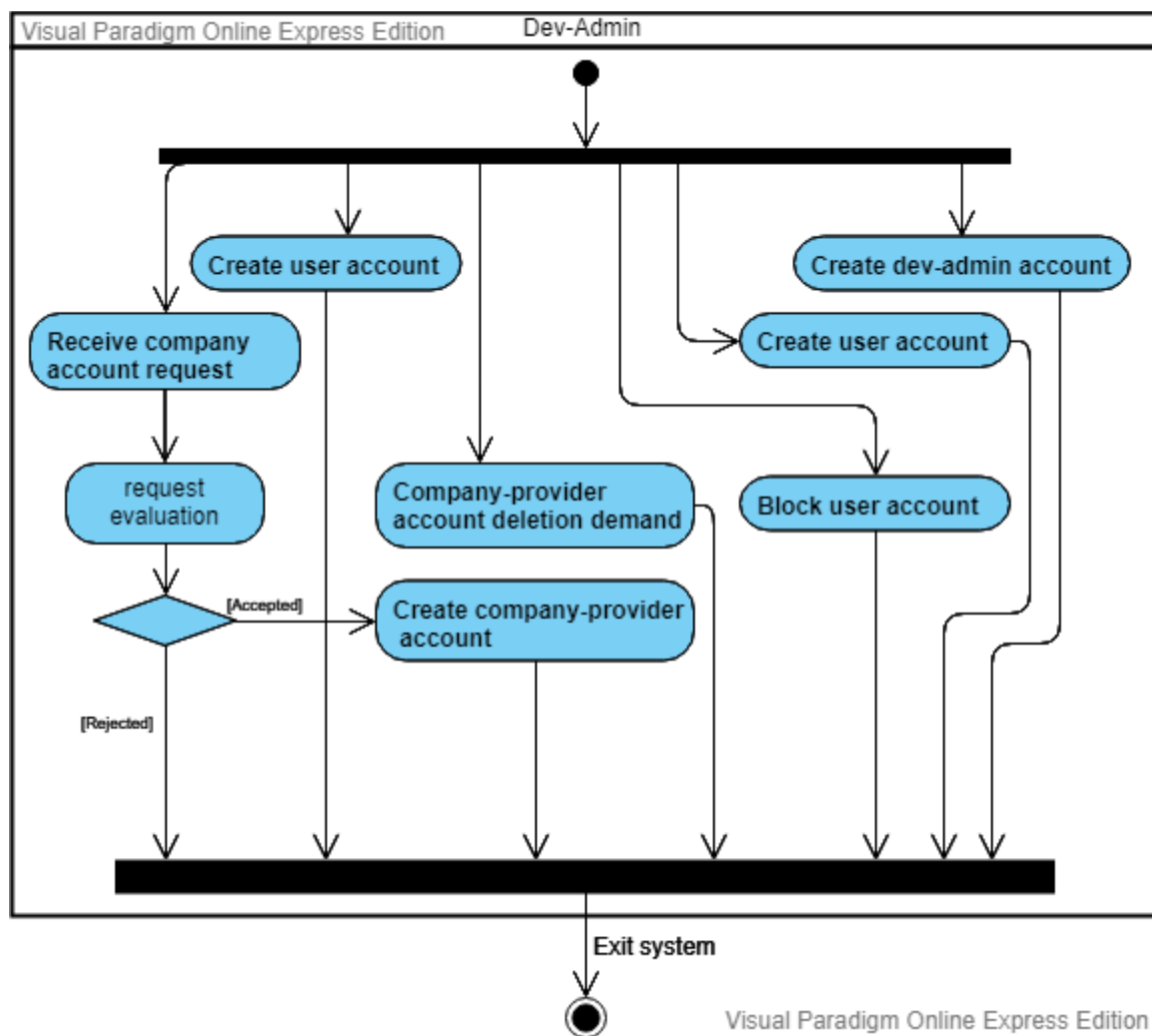
#### 3.2.5.5 Παράμετροι

Οι παράμετροι που επηρεάζουν την ορθότητα των δεδομένων παράγονται αυτόματα από το σύστημα και άρα δεν τίθεται θέμα εγκυρότητας δεδομένων, με δεδομένη την ορθότητα του συστήματος.

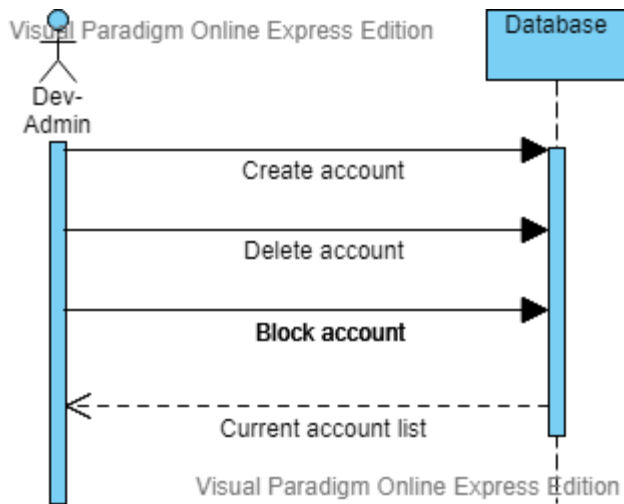
#### 3.2.5.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Ο διαχειριστής είτε δημιουργεί λογαριασμό διαχειριστή, είτε λογαριασμό χρήστη/εθελοντή, είτε μπλοκάρει έναν λογαριασμό χρήστη, είτε διαγράφει έναν εταιρικό λογαριασμό, είτε δέχεται αίτημα από εταιρία για δημιουργία λογαριασμού.
2. Στην περίπτωση του αιτήματος από εταιρία κάνει εκτίμηση και είτε το απορρίπτει είτε το αποδέχεται και δημιουργεί εταιρικό λογαριασμό
3. Αποσυνδέεται από το σύστημα. Σε όλες τις άλλες περιπτώσεις εκτός από το εταιρικό αίτημα για νέο λογαριασμό το βήμα 2 παραλείπεται.

#### Activity UML



#### Sequence UML



### 3.2.5.7 Δεδομένα εξόδου

### 3.2.3.8 Παρατηρήσεις

## 3.3 Απαιτήσεις επιδόσεων

Οι απαιτήσεις σε αυτό το section παρέχουν αναλυτική προδιαγραφή της αλληλεπίδρασης του χρήστη με το λογισμικό και τις μετρήσεις που γίνονται στην απόδοση του συστήματος. Η εφαρμογή τρέχει από κάποιον Web Server. Το αρχικό loadtime εξαρτάται από την σύνδεση στο internet, η οποία εξαρτάται από το hardware του χρήστη που χρησιμοποιείται. Άρα η γενικότερη απόδοση είναι άμεσα εξαρτώμενη με το hardware του εκάστοτε χρήστη.

### 3.3.1 Κυρίαρχο feature αναζήτησης

ID: QR1

TITLE: Κυρίαρχο feature αναζήτησης

DESC: Το feature της αναζήτησης πρέπει να είναι κυρίαρχο και εύκολα προσβάσιμο από τον χρήστη.

RAT: Με σκοπό την εύκολη εύρεση από τον χρήστη του feature της αναζήτησης

DEP: -

### 3.3.2 Χρήση του feature της αναζήτησης

ID: QR2

TITLE: Χρήση του feature της αναζήτησης

DESC: Οι διαφορετικές επιλογές (φίλτρα) πρέπει να είναι εμφανή, απλά και ευκολοκατανόητα

RAT: Με σκοπό ο χρήστης να αναζητά με ευκολία

DEP: -

### 3.3.3 Χρησιμότητα των αποτελεσμάτων στην λίστα προβολής

ID: QR3

TITLE: Χρησιμότητα των αποτελεσμάτων στην λίστα προβολής

DESC: Τα αποτελέσματα-προϊόντα που εμφανίζονται στην λίστα προβολής πρέπει να είναι user friendly και ευκολοκατανόητα. Η επιλογή ενός αποτελέσματος από την λίστα πρέπει να γίνεται με ένα κλικ.

RAT: Με σκοπό ο χρήστης να χρησιμοποιεί την λίστα προβολής αποτελεσμάτων με ευκολία

DEP: -

### 3.3.4 Χρησιμότητα του information συνδέσμου

ID: QR4

TITLE: Χρησιμότητα του information συνδέσμου

DESC: Ο information σύνδεσμος πρέπει να είναι κυριάρχος και πρέπει να γίνεται εμφανές ότι είναι ένας χρήσιμος σύνδεσμος.

RAT: Με σκοπό ο χρήστης να χρησιμοποιεί τον information σύνδεσμο με ευκολία

DEP: -

### 3.3.5 Χρόνος Απόκρισης

ID: QR5

TAG: ResponseTime

GIST: Η ταχύτητα αναζήτησης

SCALE: Ο χρόνος απόκρισης μίας αναζήτησης (κυρίως με keyboard που αντιπροσωπεύει τα worst case σενάρια)

METER: Μετρήσεις όπως αυτές προέκυψαν από 1000 αναζητήσεις κατά την διάρκεια δοκιμών

MUST: όχι περισσότερο από 2 seconds 100% του χρόνου

WISH: όχι από 1 second 100% του χρόνου

### 3.3.6 Εξάρτηση Συστήματος

ID: QR6

TAG: SystemDependency

GIST: Η ανεκτικότητα λαθών του συστήματος

SCALE: Εάν το σύστημα χάσει την σύνδεση στο Internet ή στον GPS Server (Location Server), ο χρήστης πρέπει να ενημερώνεται άμεσα.

METER: Μετρήσεις όπως αυτές προέκυψαν από 1000 ώρες κατά την διάρκεια δοκιμών

MUST: 100% του χρόνου

### 3.3.7 Extra Απαιτήσεις

- ✓ Αριθμός αναζητήσεων με μηδενικά αποτελέσματα
- ✓ Αριθμός εταιρειών που έχουν επιλέξει το πακέτο υπηρεσιών
- ✓ Ποσότητα καταχωρημένων προϊόντων και ανά shop
- ✓ Επισκεψιμότητα ιστοσελίδας, γεωγραφική διαστρωμάτωση επισκέψεων
- ✓ Αριθμός dead links – error pages
- ✓ Αριθμός εγγεγραμμένων χρηστών
- ✓ Κλικ ανά απλό χρήστη/επισκέπτη

## 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.

- User: username/varchar(45), firstName/varchar(45), lastName/varchar(45), email/varchar(45), password/varchar(45) κλπ
- Product: name/varchar(255), description/MEDIUMTEXT, category/varchar(128)
- Shop: name/varchar(255), lng/float(13,10), lat/float(13,10), tags/MEDIUMTEXT

Μέσω τον προσδιορισμό συγκεκριμένων μετρικών επιδιώκουμε την απόκτηση εικόνας για τον απαιτούμενο χώρο των διαφόρων οντοτήτων της βάσης. Στην βάση γίνεται διατήρηση κατάλληλων κλειδιών για προσδιορισμό μοναδικά κάθε εγγραφής και αποφυγή διπλότυπων πληροφοριών (ή ημιτελών)

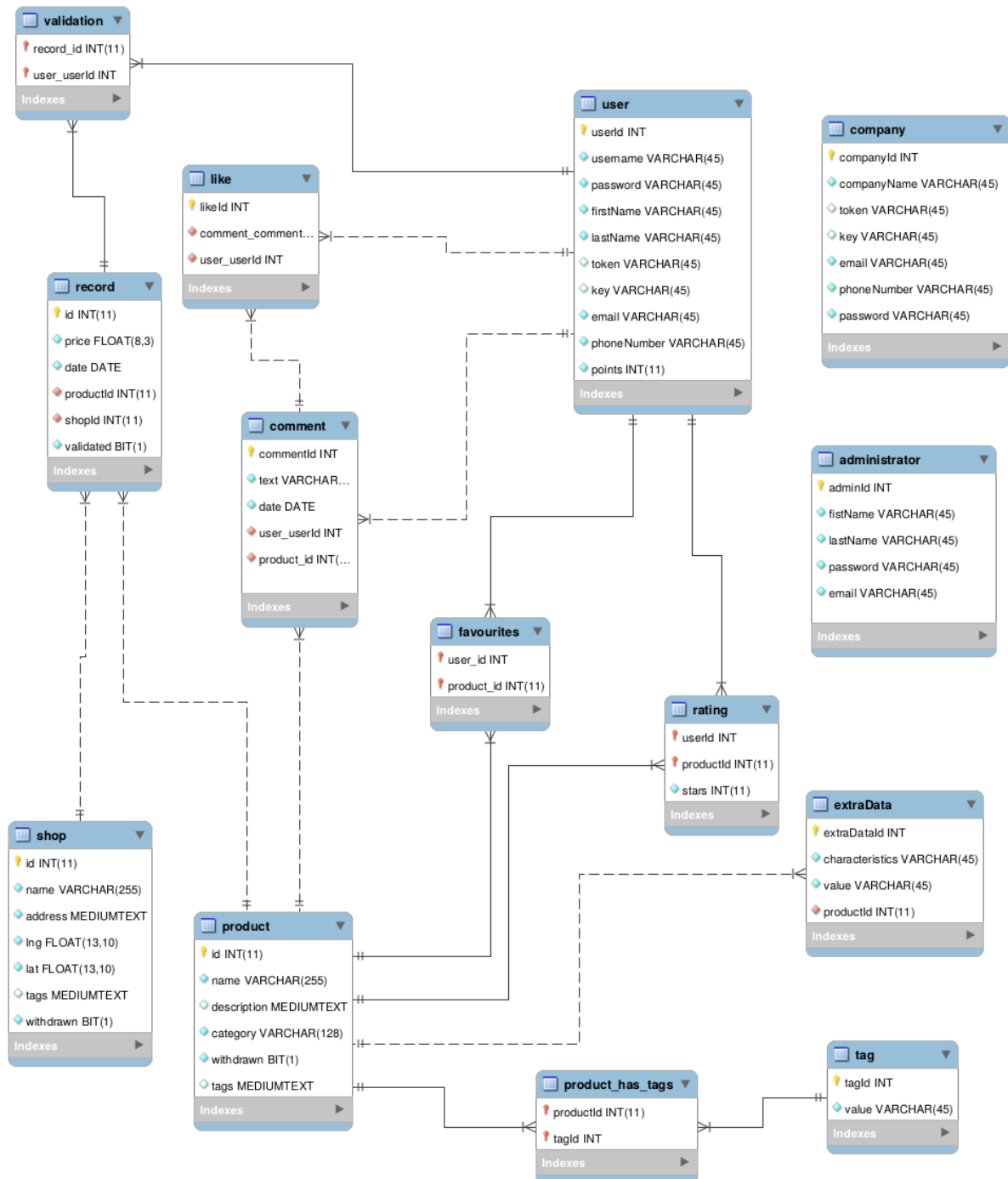
### 3.4.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

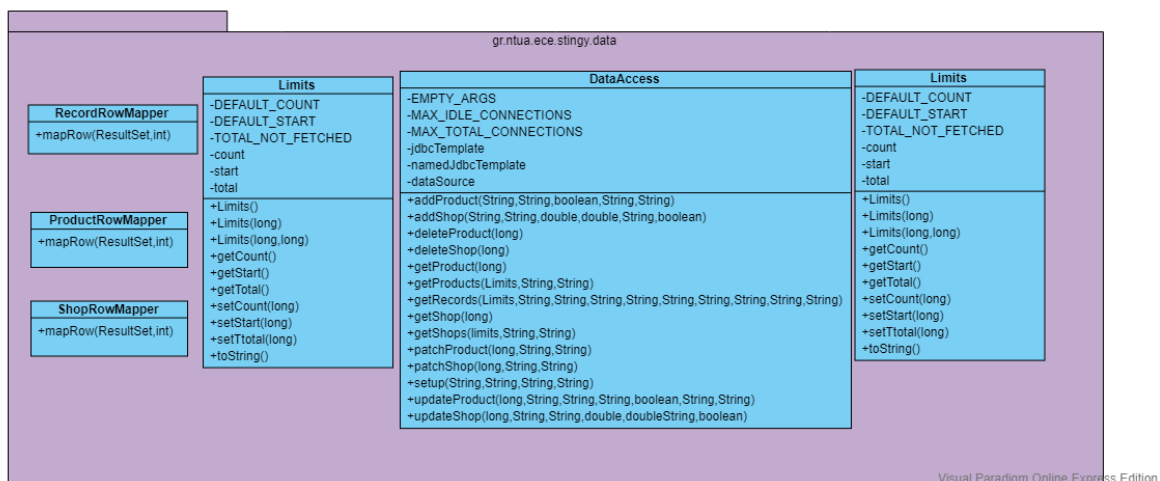
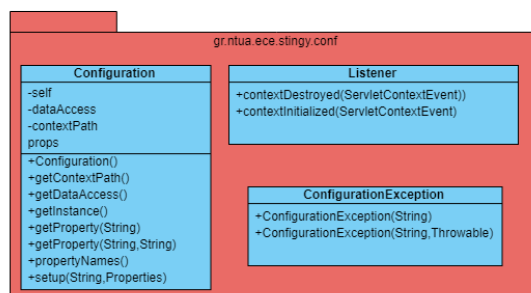
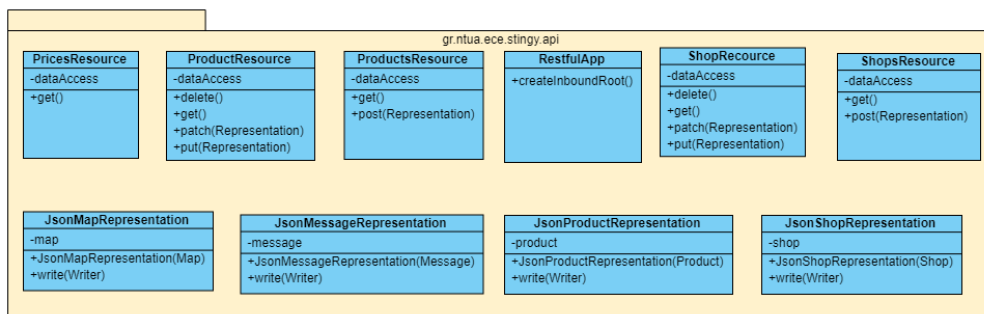
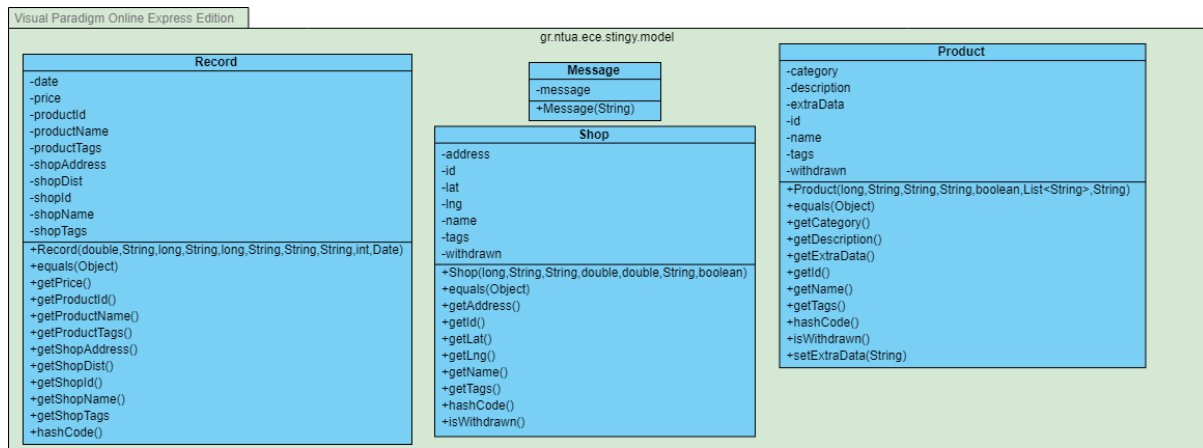
- Εφαρμογή του GDPR και προστασία των δεδομένων των χρηστών.
- Τα shops δεν έχουν την δυνατότητα παρακολούθησης πωλήσεων των προϊόντων μέσω της πλατφόρμας παρά μόνο της κριτικής που υφίστανται (δυνατότητα ανταπαντήσεων σε εταιρείες που έχουν επιλέξει το πακέτο υπηρεσιών), καθώς η εφαρμογή δεν έχει αγοροπωλησιακό περιεχόμενο
- Οι απλοί και οι εγγεγραμμένοι χρήστες δεν έχουν δυνατότητα πρόσβασης σε ευαίτητα προσωπικά δεδομένα άλλων χρηστών (πχ τοποθεσία)
- Οι απλοί χρήστες δεν έχουν την δυνατότητα καταγραφής & τροποποίησης προϊόντων, υποβολής σχολίων, ερωταπαντήσεων και επικοινωνίας με τους διαχειριστές,
- Οι εγγεγραμμένοι χρήστες έχουν δυνατότητες υποβολής σχολίων, ερωταπαντήσεων και επικοινωνίας με τους διαχειριστές αλλά όχι καταγραφής & τροποποίησης προϊόντων

- ο Οι εθελοντές έχουν όλες τις παραπάνω δυνατότητες
- ο Οι εταιρείες με πακέτο υπηρεσιών έχουν τις δυνατότητες των εγγεγραμμένων χρηστών και πρόσβασης σε analytics, ενώ όσες δεν το έχουν επιλέξει θεωρούνται απλοί χρήστες.

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

*Μοντέλα δεδομένων UML ή/και ER*





### 3.4.4 Προδιαγραφές ακεραιότητας δεδομένων

Αφορά κυρίως τον σχεδιασμό της βάσης δεδομένων:

- ✓ Ανάθεση εξ' ορισμού τιμές σε κάποιες κolumnες για εξασφάλιση της ομαλής λειτουργίας ορισμένων ενεργειών του χρήστη (πχ. Εισαγωγή δεδομένων κατά την εγγραφή ή τροποποίηση αυτών στο edit profile)
- ✓ Εφαρμογή διαφόρων τύπων περιορισμών στα δεδομένα ενός πίνακα ή μιας κolumnας, όπως πχ. NOT NULL και triggers.

- ✓ Δημιουργούμε συσχετίσεις που ορίζονται από τα κύρια και δευτερεύοντα κλειδιά τα οποία «δένουν» την πληροφορία των πινάκων μιας σχεσιακής βάσης δεδομένων όπως η MySQL.
- ✓ Γενικότερη δημιουργία triggers για διατήρηση ακεραιότητας δεδομένων

Με τους παραπάνω τρόπους εξασφαλίζεται η ακεραιότητα, η πληρότητα και η ορθότητα τους, ώστε να γίνει ενίσχυση της ομαλής λειτουργίας της πλατφόρμας. Γενικότερα επιβάλλουμε διαφόρων τύπων περιορισμούς με σκοπό την διασφάλιση της δομής και της εγκυρότητας της βάσης δεδομένων. Η χρήση κύριων και ξένων κλειδιών διασφαλίζουν την σχεσιακή δομή της βάσης. Μέσω των κανόνων δημιουργείται και συνοχή των δεδομένων των εμπλεκόμενων πινάκων.

### 3.4.5 Προδιαγραφές διατήρησης δεδομένων

Στην κατεύθυνση της αποθήκευσης δεδομένων και backup αυτών το σύστημα έχει μία πολύ κομψή προσέγγιση. Συγκεκριμένα, βάσει των διαδικτυακών ιχνών visitors και εγγεγραμμένων χρηστών δημιουργούνται analytics (heatmaps, etc) τα οποία προσφέρονται μόνο στις εταιρείες που έχουν επιλέξει το πακέτο υπηρεσιών. Παρόλα αυτά:

- ✓ Τα analytics καθώς και τα τόσα πεδία των κλάσεων Product, Shops και Record αποτελούν βασική πηγή ιστορικού για δεδομένα σε επίπεδο περιεχομένου της πλατφόρμας
- ✓ Θεωρούμε ότι τα reports που παρέχονται στον διαχειριστή του συστήματος σε περιοδική βάση (μπορεί να μην είναι dev-admin, βλέπε 1.2), βάσει προεπεξεργασίας των παραπάνω δεδομένων, αποτελούν καλή πηγή για backup μιας πιο business πτυχής της πορείας της πλατφόρμας στο βάθος χρόνου
- ✓ Γίνεται συντήρηση του hardware του server που στήνεται κάθε φορά η πλατφόρμα (ίσως και αχρείασ λόγω του portability που παρουσιάζει)
- ✓ Προγραμματισμένα system patches.

## 3.5 Περιορισμοί σχεδίασης

### ΑΡΧΕΣ ΠΕΡΙΟΡΙΣΜΩΝ

- ✓ Το Stingy αποτελεί σπουδαστική εργασία και θα βρίσκεται αναρτημένο στον github ενώ τα αρχεία είναι δομημένα και ιεραρχημένα βάσει προδιαγραφών που δόθηκαν. Στην παρούσα φάση τα προϊόντα που θα τοποθετηθούν στην βάση έγινε χειροκίνητα, παρόλα αυτά αξίζει να σημειωθεί πως έγινε στοιχειώδες web scrapping για εξόρυξη δεδομένων για τεχνολογικά προϊόντα από διάφορες ελληνικές εταιρείες. Στο μέλλον, αν κριθεί απαραίτητο θα γίνει αποδοτική χρήση αυτών.
- ✓ Η ομαλή λειτουργία της πλατφόρμας επιτυγχάνεται εφόσον πληρούνται οι λειτουργικές απαιτήσεις που έχουν αναλυθεί.
- ✓ Ο κώδικας είναι επαρκώς δομημένος και περιέχει σχόλια τα οποία εξηγούν αναλυτικά το κάθε κομμάτι με απώτερο σκοπό την εύκολη τροποποίηση του.
- ✓ Η εφαρμογή πρέπει να έχει responsive design για εύκολη αξιοποίηση από mobile συσκευές.

### ΠΟΛΙΤΙΚΗ ΟΝΟΜΑΤΟΛΟΓΙΑΣ

Τηρούνται τα name constraints της γλώσσας (πχ όνομα αρχείου = όνομα class).

### BACKEND

Αποφασίσαμε να κάνουμε χρήση java για το backend αντί για node.js που χρησιμοποιεί event driven και nonblocking IO μοντέλο.

Για node.js (+): Επειδή είναι javascript runtime ουσιαστικά θα χρησιμοποιούσαμε την ίδια γλώσσα προγραμματισμού σε frontend και backend, γεγονός που θα επέτρεπε και στα 7 μέλη της ομάδας να λειτουργούμε ως full stack engineers. Μάλιστα το express είναι ένα flexible web app framework της node.js που επιτρέπει τον σχεδιασμό της εφαρμογής με rapid prototyping, ενώ παράλληλα δεν είναι περιοριστικό ως προς την αρχιτεκτονική που θα χρησιμοποιήσουμε.

Για node.js (-) & java(+): Παρ'όλα αυτά η node.js δεν είναι ιδανική όταν πρόκειται για project μεγάλης κλίμακας που περιλαμβάνει threads. Δεν χειρίζεται threads τόσο καλά όπως η java και αυτό είναι τα πιο ελαττωματικά σημεία του framework. Επιπροσθέτως το πλήθος των βιβλιοθηκών που είναι διαθέσιμες στην java δεν



συγκρίνονται με τα ελάχιστα της node.js. Γενικά η Java έχει ένα rock-solid οικοσύστημα βιβλιοθηκών και εργαλείων. Κρίσιμα project απαιτούν την χρήση της, καθώς δημιουργεί μια αίσθηση σιγουριάς ότι το project θα υλοποιηθεί επιτυχώς με ελαχιστοποιημένες ανάγκες για error handling. Επίσης υπάρχει μεγάλη ποικιλομορφία σε χρήση των IDE. Εμείς επιλέξαμε το eclipse. Τέλος, η χρήση του jvm είναι τόσο ωφέλιμη, καθώς επιτρέπεται την ανεξαρτησία του κώδικα από το λειτουργικό σύστημα και το τρέξιμο του κώδικα σε μεγαλύτερη ταχύτητα.

Οι τεχνολογίες που χρησιμοποιούμε δεν μας επιτρέπουν rapid prototyping, παρ' όλα αυτά με χρήση του JVM επιταχύνονται οι διαδικασίες μεταγλώττισης που απαιτούνται. Ενώ στην node.js ορίζονται τα events που αντιστοιχούν σε routes της πλατφόρμας μας και με http requests του χρήστη γίνονται ασύγχρονα τα callbacks (για rendering το κατάλληλο html), στην java γίνεται χρήση των interfaces που παρέχουν μηχανισμό ισοδύναμο με callbacks.

Έτσι λοιπόν στα πλαίσια του project που αποζητούμε μια solid διεκπεραίωση της εγκυρότητας του κώδικα, αδιαφορώντας για το building time, η χρήση της java κρίθηκε καταλληλότερη, ειδικά εφόσον έγιναν και ειδικά εργαστήρια πάνω σε αυτήν. Σε επίπεδο αρχιτεκτονικής υλοποιήσαμε web εφαρμογή πολλαπλών σελίδων η οποία υλοποιεί serverside rendering.

## FRONTEND

Κατάλληλη χρήση node.js, HTML, CSS, JS.

Στο frontend για να υλοποιήσουμε το responsive design αποφασίσαμε να χρησιμοποιήσουμε το bootstrap. Με αυτόν τον τρόπο ικανοποιούμε την απαίτηση υποστήριξης χρηστών που χρησιμοποιούν συσκευή με οποιοδήποτε μέγεθος οθόνης. Χρησιμοποιούμε επίσης jquery, μια βιβλιοθήκη της javascript που επιτρέπει την γραφή σύντομου κώδικα για dom manipulation και event handling.

## ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Χρησιμοποιούμε ως κύρια βάση δεδομένων την MySQL, και όχι PostgreSQL το DMBS της οποίας υλοποιεί ACID transactions, καθώς δεν προέκυψε αναγκαιότητα επέκτασης του παρατηρητήριου σε eshop με δυνατότητες ηλεκτρονικών συναλλαγών. Χρήση των αντίστοιχων constraints των σχεσιακών δομών (πχ. NOT NULL τιμές)

## 3.6 Λοιπές απαιτήσεις

### 3.6.1 Απαιτήσεις διαθεσιμότητας λογισμικού

ID: QR6

TAG: SystemAvailability

GIST: Η διαθεσιμότητα του συστήματος όταν χρησιμοποιείται

SCALE: Η μέση διαθεσιμότητα του συστήματος (χωρίς να λαμβάνουμε υπόψιν τις αποτυχίες σύνδεσης δικτύου)

METER: Μετρήσεις όπως αυτές προέκυψαν από 1000 ώρες χρήσης κατά την διάρκεια δοκιμών

MUST: Περισσότερο από 98% του χρόνου

PLAN: Περισσότερο από 99% του χρόνου

WISH: 100% του χρόνου

ID: QR7

TITLE: Internet Connection

DESC: Η Web εφαρμογή πρέπει να είναι συνδεδεμένη στο internet

RAT: Με σκοπό η εφαρμογή να επικοινωνεί με την βάση δεδομένων

DEP: -

ID: QR8

TITLE: GPS Connection

DESC: Η Web εφαρμογή πρέπει να είναι συνδεδεμένη στην GPS συσκευή (Location Server)

RAT: Με σκοπό η εφαρμογή να λαμβάνει τις τοποθεσίες των χρηστών, τον χάρτη και να υπολογίζει τις αποστάσεις με τις εταιρείες. Η απόσταση χρησιμοποιείται και ως φίλτρο αναζήτησης.  
DEP: -

Παρατηρήσεις:

1. Testing και δοκιμές σεναρίων και ανάκτησης της εφαρμογής, δοκιμές με REST API client.
2. Αντίγραφα ασφαλείας ανά τακτά χρονικά διαστήματα.

### 3.6.2 Απαιτήσεις ασφάλειας

**ID:** QR9

**TAG:** CommunicationSecurity

**GIST:** Ασφάλεια της επικοινωνίας μεταξύ συστήματος και server

**SCALE:** Τα μηνύματα πρέπει να κρυπτογραφούνται για επικοινωνίες log-in, ώστε κανένας άλλος να μην μπορεί να γνωρίζει email ή κωδικούς μέσω αυτών των μηνυμάτων.

**METER:** Προσπάθειες για λήψη email κωδικού από λαμβανόμενα μηνύματα σε 1000 log-in sessions

**MUST:** 100% των μηνυμάτων επικοινωνίας σε log-in sessions πρέπει να είναι κρυπτογραφημένα

Ορισμός Μηνυματος επικοινωνίας: Οποιαδήποτε ανταλλαγή πληροφορίας μεταξύ client και server

**ID:** QR10

**TAG:** CompanyLoginAccountSecurity

**GIST:** Ασφάλεια λογαριασμών

**SCALE:** Εάν κάποια εταιρεία που έχει επιλέξει το πακέτο υπηρεσιών και άρα έχει account στην πλατφόρμα (με τι αντίστοιχες δυνατότητες), προσπαθήσει να κάνει log in με έναν μη υπάρχων λογαριασμό, τότε δεν πρέπει να συνδεθεί. Η εταιρεία πρέπει να ειδοποιηθεί για το log-in failure

**METER:** 1000 προσπάθειες για log-in με μη υπάρχοντα λογαριασμό κατά την διάρκεια δοκιμών

**MUST:** 100% του χρόνου

**ID:** QR11

**TAG:** AdminLoginAccountSecurity

**GIST:** Ασφάλεια λογαριασμών

**SCALE:** Εάν κάποιος admin προσπαθήσει να κάνει log in με έναν μη υπάρχων λογαριασμό, τότε δεν πρέπει να συνδεθεί. Ο admin πρέπει να ειδοποιηθεί για το log-in failure

**METER:** 1000 προσπάθειες για log-in με μη υπάρχοντα λογαριασμό κατά την διάρκεια δοκιμών

**MUST:** 100% του χρόνου

**ID:** QR12

**TAG:** CompanyAccountSecurity

**GIST:** Ασφάλεια λογαριασμών των εταιρειών

**SCALE:** Μια εμπλεκόμενη εταιρεία και μία διεύθυνση IP δεν πρέπει να έχει την δυνατότητα log in για συγκεκριμένο χρονικό διάστημα ύστερα από 3 αποτυχημένες log-in προσπάθειες

**METER:** 1000 προσπάθειες για log-in κατά την διάρκεια περιόδου «κλειδώματος» μετά δηλαδή από τις 3 αποτυχημένες σε σειρά log-in προσπάθειες.

**MUST:** Η περίοδος «κλειδώματος» πρέπει να είναι μισή ώρα, και κατά την διάρκεια αυτή η δυνατότητα log-in πρέπει να είναι ανενεργή

**ID:** QR13

**TAG:** AdminAccountSecurity

**GIST:** Ασφάλεια λογαριασμών admin

**SCALE:** Ένας admin και μία διεύθυνση IP δεν πρέπει να έχει την δυνατότητα log in για συγκεκριμένο χρονικό διάστημα ύστερα από 3 αποτυχημένες log-in προσπάθειες

**METER:** 1000 προσπάθειες για log-in κατά την διάρκεια περιόδου «κλειδώματος» μετά δηλαδή από τις 3 αποτυχημένες σε σειρά log-in προσπάθειες.

**MUST:** Η περίοδος «κλειδώματος» πρέπει να είναι μισή ώρα, και κατά την διάρκεια αυτή η δυνατότητα log-in πρέπει να είναι ανενεργή

**ID:** QR14

**TAG:** UserCreateAccountSecurity

GIST: Η ασφάλεια που αφορά την δημιουργία λογαριασμού για χρήστες του συστήματος  
SCALE: Εάν κάποιος χρήστης επιθυμεί να δημιουργήσει account και να εγγραφεί στην πλατφόρμα, και το επιθυμητό username είναι κατειλημμένο, ο χρήστης πρέπει να επιλέξει διαφορετικό όνομα  
METER: Μετρήσεις όπως αυτές προέκυψαν από 1000 ώρες χρήσης κατά την διάρκεια δοκιμών  
MUST: 100% του χρόνου

**ID:** QR15

**TAG:** HandlingAccountSecurity

GIST: Η ασφάλεια που αφορά την διαχείριση λογαριασμών χρηστών, εθελοντών και εταιρειών

SCALE: Δεν υπάρχει κάποιος άλλος χρήστης πέραν των dev-admin με διαχειριστικά δικαιώματα πάνω στους λογαριασμούς.

METER: 1000 προσπάθειες για τροποποίηση ή διαγραφή στοιχείων για λογαριασμούς από unauthorized users.

MUST: Ban του αντίστοιχου account (αν η κακόβουλη προσπάθεια έγινε από εγγεγραμμένο χρήστη) από τον dev-admin

#### Παρατηρήσεις:

1. Όπως προκύπτει από την διαμόρφωση του συστήματος, δεν απαιτείται απαίτηση ασφάλειας μορφής «CompanyCreateAccountSecurity», καθώς τα accounts – profile των εταιρειών δημιουργούνται από τους εθελοντές του συστήματος ή τον admin, ενώ μόνο όσες εταιρείες επιλέξουν το πακέτο προσφορών τους δίνεται πρόσβαση σε αυτά (με τι αντίστοιχες δυνατότητες όπως έχουν αυτές περιγραφεί στο κατάλληλο StRS-Companies)
2. Έλεγχος CAPTCHA
3. Απαίτηση από χρήστες να βάζουν κωδικό με constrained επιλογές (ελάχιστο μήκος, απαίτηση αριθμών, γραμμάτων κλπ)

### 3.6.3 Απαιτήσεις συντήρησης

**ID:** QR16

**TITLE:** Application Extendibility

DESC: Η Web app πρέπει να είναι εύκολα επεκτάσιμη. Ο κώδικας πρέπει να είναι γραμμένος με τρόπο τέτοιο ώστε να ευνοεί την υλοποίηση νέων λειτουργιών.

RAT: Με σκοπό την εύκολη υλοποίηση μελλοντικών λειτουργιών στην εφαρμογή.

DEP: -

**ID:** QR17

**TITLE:** Application Testability

DESC: Περιβάλλοντα δοκιμών πρέπει να χτιστούν για την Web App που να επιτρέπουν την δοκιμή διαφορετικών λειτουργιών της εφαρμογής.

RAT: Με σκοπό την δοκιμή της εφαρμογής

DEP: -

### 3.6.44 Απαιτήσεις αξιοπιστίας

**ID:** QR18

**TITLE:** System Messaging Errors

DESC: Το σύστημα πρέπει να εμφανίζει κατάλληλα μηνύματα όταν το σύστημα είναι down.

RAT: Με σκοπό την καθυσύχαση των χρηστών

DEP: -

**ID:** QR19

**TITLE:** System Invalid Data Errors

DESC: Το σύστημα πρέπει να εμφανίζει κατάλληλα μηνύματα λάθους όταν κάποιος χρήστης προσπαθεί να εισαγάγει invalid data. Αυτά καθορίζονται βάσει των περιορισμών, για ορισμένα use cases αυτοί έχουν δοθεί.

RAT: Με σκοπό την εύκολη κατανόηση του λόγου για την δημιουργία του error.

DEP: -

#### Παρατηρήσεις:

1. Τακτικός έλεγχος του περιεχομένου από τον admin και ανανέωση των χαρακτηριστικών ανάλογα με τις τάσεις στην αξιοποίηση τους από τους χρήστες.

### 3.6.4 Extra Απαιτήσεις

- ✓ Η web εφαρμογή είναι ευέλικτη για χρήση τόσο για τον admin όσο και για τους χρήστες της εφαρμογής
- ✓ Είναι portable, απαιτείται απλά σύνδεση στο internet
- ✓ Admin-developers, testers και χρήστες μπορούν να την χρησιμοποιήσουν εύκολα για τους σκοπούς που περιγράφηκαν στα StRS, και να την συντηρούν.

## 4. Παράρτημα

### 4.1 Παραδοχές και εξαρτήσεις

#### Για τον χρήστη

- ✓ Ο χρήστης έχει προηγούμενη γνώση σε βασικές λειτουργίες λειτουργικού συστήματος όπως ενεργοποίηση της υπολογιστικής μονάδας, login, χρήσης και περιήγησης σε κάποιο browser με κατάληξη την διατύπωση της σωστής διεύθυνσης url της πλατφόρμας μας.

#### Για τον εθελοντή

- ✓ Διατίθεται pc lab x υπολογιστικών μονάδων για όσους εθελοντές επιθυμούν την συστηματικοποίηση της εργασίας τους και χρήση του LAN network

#### Για το σύστημα

- ✓ Υποθέτουμε ότι το Stingy τρέχει σε ένα πλήρως λειτουργικό web server και database system με δυνατότητα σύνδεσης στο internet που επιτρέπει την χρήση του admin UI για τις λειτουργίες όπως αυτές έχουν αναλυθεί στο use case 5, + καθώς και την επικοινωνία με άλλους χρήστες όταν κριθεί αναγκαίο (περιπτώσεις απόρριψης αιτήματος εθελοντή κλπ)

#### Υποθέσεις

- ✓ Ενώ με αποσύνδεση κάποιου χρήστη η λίστα αγαπημένων προϊόντων παραμένει ως έχει, δεν υπάρχει αντίστοιχη για το ιστορικό αναζήτησης του σε επίπεδο UI για τον ίδιο (σε επίπεδο analytics ναι)
- ✓ Οποιοσδήποτε χρήστης μπορεί να κάνει επεξεργασία των προσωπικών του πληροφοριών (το οποίο επίσης έχει νόημα σε επίπεδο analytics)
- ✓ GPS components σε όλα τα κινητά δουλεύουν με τον ίδιο τρόπο για αποφυγή αναγκαστικών τροποποιήσεων

### 4.2 Ακρωνύμια και συντομογραφίες

- ✓ GDPR: General Data Protection Regulation. Ευρωπαϊκός κανονισμός για την προστασία προσωπικών δεδομένων των ιδιωτών στα πλαίσια της Ευρωπαϊκής Ένωσης.
- ✓ IEEE: Institute of Electrical and Electronics Engineers
- ✓ XML: Extensive Markup Language – Γλώσσα που αφορά το front end
- ✓ HTTP, HTTPS: Πρωτόκολλο internet για ασφαλή μεταφορά δεδομένων στο δίκτυο
- ✓ Web scrapping: Η διαδικασία εξαγωγής συγκεκριμένων δεδομένων από ένα site.
- ✓ DB: Database
- ✓ MVC: Model View Controller
- ✓ GUI: Graphical User Interface – κυριώς για τις διεπαφές χρήστη.

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

Στον φάκελο docs περιέχονται τα αρχεία StRS για κάθε ομάδα χρηστών που λαμβάναμε υπόψιν κατά τον σχεδιασμό της εφαρμογής. Επιπροσθέτως περιλαμβάνονται τα αρχεία των διαγραμμάτων του παρόντος εγγράφου μαζί με το deployment diagram