



Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών  
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

# Πρότυπο Παραδοτέου SRS Εγγράφου

Εργασία στο Μάθημα «Ανάλυση & Σχεδίαση Συστημάτων Λογισμικού»

Εαρινό εξάμηνο 2019-20

## ΕΚΔΟΣΗ 2

- Κεφ. 1.2 : δυνατότητα τροποποίησης των απαντήσεων που έχετε ήδη καταθέσει στο ερωτηματολόγιο
- Κεφ. 1.3 και 1.4: διευκρινιστική επαναδιατύ πωση του ζητουμένου
- Προσθήκη πίνακα στο τέλος του εγγράφου, όπου αναφέρεται η βαρύτητα του κάθε ζητουμένου στη βαθμολόγηση της εργασίας

## Ομάδα 1

Κωνσταντίνος Σκορδούλης (1115 201600155), [sdi1600155@di.uoa.gr](mailto:sdi1600155@di.uoa.gr)

Σωτήρης Παυλόπουλος (1115 201100082), [sdi1100082@di.uoa.gr](mailto:sdi1100082@di.uoa.gr)

1	Εισαγωγή .....	2
1.1	Σκοπός του συστήματος.....	2
1.2	Επιχειρησιακό πλαίσιο (ερωτηματολόγιο) .....	2
1.3	Εμβέλεια του συστήματος, λειτουργικές εξαρτήσεις, παραδοχές και περιορισμοί .....	3
1.4	Κρίσιμοι παράγοντες για την επιτυχία του συστήματος.....	4
2	Λειτουργικές Απαιτήσεις και Σενάρια Χρήσης .....	4
2.1	Ρόλοι και δικαιώματα χρηστών.....	4
2.2	Λειτουργικές Απαιτήσεις.....	4
2.3	Σενάριο Χρήσης 1 – Τίτλος .....	4
2.4	Σενάριο Χρήσης 2 - Τίτλος .....	5
2.5	Σενάριο Χρήσης N - Τίτλος .....	5

3	Δομημένη μοντελοποίηση διαδικασιών .....	5
4	Αντικειμενοστρεφής μοντελοποίηση δεδομένων .....	5
5	Αρχιτεκτονική του συστήματος .....	6
5.1	Επίπεδο βάσης δεδομένων .....	6
5.2	Επίπεδο προσπέλασης δεδομένων .....	6
5.3	Επίπεδο επιχειρησιακής λογικής .....	6
5.4	Επίπεδο Υποδοχής Αιτημάτων – Αποστολής Απαντήσεων.....	7
6	Ποιοτικά Χαρακτηριστικά και Μη-Λειτουργικές Απαιτήσεις.....	9
6.1	Γενικά χαρακτηριστικά.....	9
6.2	Απαιτήσεις ευχρηστίας.....	9
6.3	Απαιτήσεις ασφάλειας .....	9
6.4	Απαιτήσεις διαλειτουργικότητας (διασύνδεσης με τρίτα συστήματα).....	9
6.5	Απαιτήσεις απόδοσης.....	9
6.6	Απαιτήσεις διαθεσιμότητας .....	9
6.7	Άλλες Απαιτήσεις .....	9
7	Προδιαγραφές της Backend Εφαρμογής .....	9
7.1	Σχεδιασμός του Επιπέδου Βάσης Δεδομένων.....	9
7.2	Σχεδιασμός του Επιπέδου Προσπέλασης Δεδομένων .....	9
7.3	Σχεδιασμός του Επιπέδου Επιχειρησιακής Λογικής.....	10
7.4	Σχεδιασμός του Επιπέδου Υποδοχής Αιτημάτων & Αποστολής Απαντήσεων.....	10
8	Διεπαφές χρήσης της Frontend Εφαρμογής .....	11
9	Σενάρια και έλεγχοι αποδοχής.....	11
10	Πίνακας Βαρύτητας.....	12

# 1 Εισαγωγή

## 1.1 Σκοπός του συστήματος

Σύντομη περιγραφή του σκοπού του συστήματος.

Ο σκοπός του συστήματός είναι διττός.

- 1) Παρέχει τη δυνατότητα στους εμπορικούς συνεργάτες την προώθηση των προϊόντων τους,
- 2) Υποστηρίζει τη δυνατότητα ηλεκτρονικού πορτοφολιού για την διεκπεραίωση συναλλαγών, με στόχο την αγορά προϊόντων των εμπορικών συνεργατών ( από τους πελάτες/χρήστες της πλατφόρμας)

## 1.2 Επιχειρησιακό πλαίσιο (ερωτηματολόγιο)

Παράθεση των ερωτήσεων / απαντήσεων που έχετε ήδη συμπληρώσει. Αν κάποιες από τις απαντήσεις σας έχουν τροποποιηθεί σε σχέση με το ερωτηματολόγιο που έχετε καταθέσει, συμπληρώστε και τις δύο εκδοχές και αιτιολογήστε το λόγο της τροποποίησης αυτής, εφόσον προέκυψε, στην τελευταία ενότητα του παρόντος κεφαλαίου (συνολικά για όλες τις αλλαγές και όχι για κάθε μία). Δηλαδή, βάζετε την κάθε τροποποίηση στο σημείο που προκύπτει και αιτιολογείτε συνολικά όλες τις αλλαγές στην τελευταία ενότητα (1.2.11).

1.2.1 Ποιοι είναι οι εμπορικοί συνεργάτες της επιχείρησης;

Όλες οι επιχειρήσεις λιανικής πώλησης (Retail), όπως super-market, καταστήματα ηλεκτρονικών ειδών, καταστήματα ένδυσης κλπ.

1.2.2 Γιατί συνεργάζονται με σας; Τι τους προσφέρει η πλατφόρμα;

Η αξία της πλατφόρμας είναι η ευρεία και στοχευμένη προβολή των προϊόντων τους σε ένα μεγάλο αγοραστικό κοινό.

Με άλλα λόγια προσφέροντας τα προϊόντα τους στη πλατφόρμα, διευρύνουν το αγοραστικό τους κοινό, με άμεση συνέπεια την αύξηση του τζίρου τους.

Επιπλέον προσφέρει ένα ανταγωνιστικό περιβάλλον, οπού κάθε επιχείρηση μπορεί να συγκρίνει τις τιμές των προϊόντων τους με άλλες, καθώς και να παρέχει εργαλεία για την ενισχυμένη προώθηση των προϊόντων τους(πχ search categories, price range, popularity etc.).

1.2.3 Ποιον χρεώνει η πλατφόρμα;

Τον Εμπορικό Συνεργάτη

1.2.4 Πώς γίνεται η χρέωση;

Ποσοστό προμήθειας (1%) με minimum χρέωση 1 πόντου. Εννοείται επειδή έχουμε ακέραιους πόντους , στη ποσόστωση θα υπάρξει στρογγυλοποίηση.

**Change:** (προσθήκη)

Η προμήθεια εφαρμόζεται πάνω στην καθαρή αγοραστική αξία.

Η τιμή κάθε προϊόντος διαμορφώνεται ως:

Τιμή = καθαρή αγοραστική αξία + προμήθεια + κόστος μεταφορικής (courier gateway service)

1.2.5 Πώς σχετίζεται το ηλεκτρονικό πορτοφόλι με τη διαδικασία της χρέωσης;

Ο χρήστης αγοράζει πόντους με prepaid card, debit card ή credit card που έχει συνδέσει στο λογαριασμό του.

Ο χρήστης καταναλώνει πόντους αγοράζοντας προϊόντα.

Οι πόντοι αυτοί διανέμονται στα ηλεκτρονικά πορτοφόλια του administrator και του εμπορικού συνεργάτη (καθαρή αγοραστική αξία, χωρίς τη προμήθεια του administrator).

Ο χρήστης μπορεί να εξαργυρώσει τους πόντους του μόνο εάν έχει συνδέσει credit ή debit card.

**Change:**

Ο πελάτης αγοράζει πόντους με χρήση του **Payment Gateway Service** (προσφέρεται από τη πλατφόρμα μας).

Ο πελάτης καταναλώνει πόντους αγοράζοντας προϊόντα (τιμή όπως περιγράφεται παραπάνω)

Οι πόντοι αυτοί διανέμονται στα ηλεκτρονικά πορτοφόλια του administrator και του εμπορικού συνεργάτη. Δηλαδή ο administrator παίρνει την προμήθεια και στον εμπορικό συνεργάτη διανέμεται το υπόλοιπο ποσό.

1.2.6 Ποια είναι η αντιστοιχία πόντων σε ευρώ;

1 πόντος: 0.33\$

Με αυτό το τρόπο, δημιουργούμε την "ψευδαίσθηση" ότι κερδίζει πολύ παραπάνω με την αγορά πόντων.

**Change:**

Η μετατροπή από πόντους σε χρήματα υποστηρίζεται μόνο για τους εμπορικούς συνεργάτες.

Η λειτουργία αυτή παρέχεται από το payment gateway service.

1.2.7 Τιείδους προγράμματα επιβράβευσης συνοδεύουν το ηλεκτρονικό πορτοφόλι;

Τα παρακάτω δίνουν BONUS πόντους στο χρήστη με την ολοκλήρωση τους.

a) Αγορά πακέτων πόντων -> πακέτα 10,20,50 πόντους ==> BONUS 2,5,10 points

b) Milestones: Είτε σε πλήθος αγορών(10,50,100 αγορές), είτε σε διάρκεια χρήσης της πλατφόρμας(ex. 1 year anniversary).

## 1.2.8 Πολιτική ακυρώσεων

Ο user μπορεί να ακυρώσει την παραγγελία πριν ξεκινήσει η διαδικασία αποστολής του προϊόντος.

Αν έχουμε παραλαβή από το κατάστημα, πριν τη παραλαβή του ( και στις 2 περιπτώσεις χωρίς penalty).

### **Change:**

Ο πελάτης μπορεί να ακυρώσει τη παραγγελία οποτεδήποτε, πριν από την παραλαβή του προϊόντος.

Έχουμε λοιπόν 2 σενάρια:

- 1) Ακύρωση παραγγελίας πριν περάσουν 24 ώρες( <= 24). **Επιστρέφεται η καθαρή αγοραστική αξία + cost μεταφορικής** (μετά τις 24 ώρες και ξεκινάει η αποστολή, θα το αναφέρουμε και στις παραδοχές).
- 2) Ακύρωση παραγγελίας μετά τις 24 ώρες. **Επιστρέφεται μόνο η καθαρή αγοραστική αξία** ( το προϊόν έχει παραδοθεί στην μεταφορική εταιρεία/courier gateway service)

Και στις 2 περιπτώσεις, δεν επιστρέφεται η **προμήθεια** της συναλλαγής (το κέρδος της πλατφόρμας)

## 1.2.9 Πολιτική επιστροφών

Ο user ως φυσικό πρόσωπο διαθέτει το δικαίωμα υπαναχώρησης, όπως ορίζεται από το Ευρωπαϊκό Δίκαιο (DOA).

Στην περίπτωση αυτή και εφόσον πληρούνται οι προϋποθέσεις, επιστρέφεται στο user το πλήρες ποσό(πόντων),

αλλά επωμίζεται τα έξοδα επιστροφής του προϊόντος ( cash \$\$\$, η πιθανή υπηρεσία courier δεν έχει καμία σχέση με τη πλατφόρμα μας).

Η πιθανή υπηρεσία courier δεν έχει καμία σχέση με τη πλατφόρμα μας (δεν είναι εμπορικός συνεργάτης).

### **Change:**

Ο χρήστης μπορεί να “δηλώσει” επιστροφή προϊόντος, **το πολύ 7 μέρες** μετά την παραλαβή του.

Ο πελάτης μπορεί να χρησιμοποιήσει το Courier Gateway Service, για την επιστροφή του προϊόντος στο κατάστημα.

**Επιστρέφεται η καθαρή αγοραστική αξία.** Δεν επιστρέφεται ούτε η προμήθεια, ούτε το courier cost.

## 1.2.10 Πολιτική αξιολογήσεων

- a) **Αξιολόγηση Προϊόντος:** Εφόσον δεν έχει ακυρωθεί η παραγγελία, επιτρέπεται η αξιολόγηση (7 μέρες!) μετά την "πληρωμή" του προϊόντος (για να εξασφαλίσουμε την παραλαβή του προϊόντος). Σε περίπτωση ακύρωσης, δεν μπορεί να αξιολογήσει.
- b) **Αξιολόγηση Εμπορικού Συνεργάτη:** Πρέπει να έχει (αξιολογήσει τουλάχιστον 1 προϊόν!) του εμπορικού συνεργάτη.

### Change:

- a) Αξιολόγηση Προϊόντος: Εφόσον δεν έχει ακυρωθεί η παραγγελία, επιτρέπεται η αξιολόγηση μετά την **παραλαβή** του προϊόντος. Σε περίπτωση ακύρωσης, δεν μπορεί να αξιολογήσει. **Δηλαδή το order να είναι complete και το orderline να είναι τουλάχιστον 1 μη ακυρωμένο.**
- b) Αξιολόγηση Εμπορικού Συνεργάτη: Πρέπει να έχει (αξιολογήσει τουλάχιστον 1 προϊόν!) του εμπορικού συνεργάτη.

### 1.2.11 Αιτιολόγηση τροποποιήσεων (εφόσον υπάρχουν)

1.2.4: Διευκρίνιση για το πώς διαμορφώνεται η τιμή του προϊόντος. Σε οποιαδήποτε παραγγελία, η αποστολή των προϊόντων γίνεται μόνο μέσω του courier gateway service, που εμπεριέχει κάποιο μικρό κόστος.

1.2.5: Η χρήση της εξωτερική υπηρεσίας **Payment Gateway Service**, για φόρτιση ηλεκτρονικού πορτοφολιού

1.2.6: Οι πόντοι μας δεν έχουν αξία πέρα από τη πλατφόρμα μας. Οπότε πρέπει να παρέχεται η δυνατότητα μετατροπής πόντων σε χρήματα (με τη βοήθεια του payment gateway service). Έτσι οι εμπορικοί συνεργάτες θα μπορέσουν να απολάβουν τα κέρδη τους.

1.2.8 , 1.2.9 , 1.2.10 : Αναθεώρηση της πολιτικής ακύρωσης και επιστροφής, για να είναι πιο σαφής και κατανοητή. Επιπλέον μικρή διόρθωση στην πολιτική αξιολόγησης.

### 1.3 Εμβέλεια του συστήματος, λειτουργικές εξαρτήσεις, παραδοχές και περιορισμοί

Για να καταλήξετε να οριθετήσετε την εμβέλεια του υπό ανάπτυξη συστήματος (δηλαδή, τι καλύπτει και τι όχι), θα πρέπει να κάνετε αναφορά:

- των εξωτερικών του εξαρτήσεων (ποια εξωτερικά συστήματα ή υπηρεσίες απαιτεί για την ορθή του λειτουργία),
- των παραδοχών που έχετε κάνει για τη λειτουργία του,
- των περιορισμών που έχουν τεθεί από το σχεδιασμό σας.

Για παράδειγμα, μια εξωτερική εξάρτηση σε τρίτο πληροφοριακό σύστημα / υπηρεσία θα μπορούσε να είναι το Payment Gateway Service που χρησιμοποιείται κατά την «φόρτιση» του πορτοφολιού των χρηστών με πόντους. Μια παραδοχή θα μπορούσε να είναι ότι δεν αλλάζει η προγραμματιστική διεπαφή του Service αυτού (δηλαδή δεν καλύπτεται η περίπτωση που η επιχείρησή σας αλλάζει τον Τραπεζικό της συνεργάτη). Αντίστοιχα, ένας περιορισμός θα μπορούσε να είναι ότι υποστηρίζεται μόνο το Ευρώ ως νόμισμα για τις οικονομικές συναλλαγές της πλατφόρμας.

Μπορεί να υπάρχουν και παραδοχές ή περιορισμοί που δεν προκύπτουν από κάποια εξωτερική εξάρτηση. Για παράδειγμα, ένας ακόμα σχεδιαστικός περιορισμός (μπορείτε να το δείτε κι ως κοινή παραδοχή για όλες τις ομάδες) είναι ότι δεν δίνεται έμφαση στον πλήρη χειρισμό των σφαλμάτων

(runtime errors) που μπορεί να προκύψουν κατά τη λειτουργία της εφαρμογής.

#####
#####

#### Λειτουργικές Εξαρτήσεις:

- 1) **Payment Gateway Service.** Είναι ο τραπεζικός συνεργάτης της επιχείρησης (όχι εμπορικός συνεργάτης). Ως είσοδο δέχεται το IBAN/τραπεζικό λογαριασμό του ατόμου που θέλει να διεξάγει χρηματική συναλλαγή με τη πλατφόρμα. Χρειαζόμαστε τον τραπεζικό συνεργάτη για 2 λόγους:
  - a. Για να “φορτίσει” το ηλεκτρονικό πορτοφόλι (χρήματα → πόντους). Δηλαδή στέλνουν τα χρήματα στο λογαριασμό μας (μέσω αυτού του service) και εμείς δίνουμε τους πόντους.
  - b. Για να πληρώσουμε τους εμπορικούς συνεργάτες (πόντοι → χρήματα). Στέλνουν τους πόντους τους στο ηλεκτρονικό πορτοφόλι μας και εμείς στέλνουμε το ανάλογο ποσό χρημάτων (μέσω αυτού του Service). Η υπηρεσία αυτή παρέχεται μόνο σους εμπορικούς συνεργάτες (όχι στους clients).
- 2) **Courier Gateway Service.** Είναι ο συνεργάτης Courier (όχι εμπορικός συνεργάτης). Χρειαζόμαστε αυτή την υπηρεσία για την παράδοση (και πιθανώς επιστροφή) των προϊόντων της πλατφόρμας (των εμπορικών συνεργατών). Δέχεται ως είσοδο 3 πράγματα:
  - a. **IBAN**, για να εισπράξει το κόστος της μεταφοράς (2€, δικιά μας παραδοχή, αναφέρεται και παρακάτω).
  - b. **Διεύθυνση παραλαβής =>** από πού θα παραλάβει το προϊόν.
  - c. **Διεύθυνση αποστολής =>** που θα το παραδώσει.

⇒ Αφού το παραδώσει, επιστρέφει **ID παραγγελίας** (δηλαδή ποια παραγγελία παρέδωσε) και **ημερομηνία και ώρα παράδοσης**.

#### Παραδοχές:

- 1) Δεν αλλάζει η προγραμματιστική διεπαφή των **Payment Gateway** και **Courier Gateway** services.
- 2) Το **Payment Gateway** service χρειάζεται το IBAN (του χρήστη που το εκκινεί) και το ποσό της μεταφοράς (χρήματα ή πόντοι μόνο για εμπορικούς συνεργάτες).
- 3) Το **Courier Gateway** service χρειάζεται τη “source” διεύθυνση (από πού θα το παραλάβει) και τη “destination” διεύθυνση (που θα το παραδώσει) καθώς και το IBAN για να καλύψει κόστος μεταφοράς.
- 4) Η αποστολή/παραλαβή προϊόντων γίνεται μέσω Courier. Δεν καλύπτεται η παραλαβή προϊόντος σε κάποιο σημείο πώλησης.
- 5) Θεωρούμε πως η **τιμή** για αποστολή οποιουδήποτε προϊόντος (οπουδήποτε) χρησιμοποιώντας το **Courier Gateway** είναι **2€** (ή 6 πόντοι) και περιλαμβάνονται στην τιμή πώλησης.
- 6) Μια **σχεδιαστική παραδοχή**, είναι ότι **WalletID** και το **UserID** του **Admin** είναι 0. (Αναφέρεται και στις παραδοχές)

#### Περιορισμοί:

- 1) Υποστηρίζεται μόνο το ευρώ για οικονομικές συναλλαγές της πλατφόρμας.
- 2) Κάθε χρήστης έχει 1 και μόνο ρόλο.
- 3) Δεν δίνεται έμφαση στον πλήρη χειρισμό των σφαλμάτων (runtime errors) που μπορεί να προκύψουν κατά τη λειτουργία της εφαρμογής (από εκφώνηση)

#### 1.4 Κρίσιμοι παράγοντες για την επιτυχία του συστήματος

Βοηθήστε τον αναγνώστη να καταλάβει ποια είναι η ιεράρχηση που πρέπει να δοθεί σε όσα ακολουθούν. Ο αναγνώστης σας είναι ο stakeholder του έργου και το ζητούμενο στην ενότητα αυτή είναι να αναδείξετε τις πλέον καθοριστικές για την επιτυχία του συστήματος λειτουργίες και χαρακτηριστικά του. Μια απλή παράθεση τους αρκεί.

#####

Οι λειτουργίες στις οποίες πρέπει να δώσουμε έμφαση (σε φθίνουσα σειρά) είναι:

- 1) Όλες οι λειτουργίες που αφορούν το **ηλεκτρονικό πορτοφόλι**, για να είναι εφικτές οι αγορές και πωλήσεις μέσω της πλατφόρμας.
- 2) **Αναζήτηση προϊόντων**, αλλιώς δεν θα μπορεί ο χρήστης να έχει πρόσβαση στα προϊόντα της πλατφόρμας.
- 3) **Login/Signup**, για να μπορεί κάθε χρήστης να παίρνει το ρόλο και τα δικαιώματα που του αντιστοιχούν.
- 4) **Διαδικασία ακύρωσης και επιστροφής προϊόντων**.
- 5) **Αξιολόγηση προϊόντων και εμπορικών συνεργατών**
- 6) **Προγράμματα επιβράβευσης**

## 2 Λειτουργικές Απαιτήσεις και Σενάρια Χρήσης

### 2.1 Ρόλοι και δικαιώματα χρηστών

Ποιοι είναι οι διακριτοί ρόλοι στο σύστημα, ποια είναι η δικαιοδοσία τους, πώς τους ανατίθεται ο ρόλος τους κλπ. Εδώ πρέπει να καταλάβει ο αναγνώστης τους βασικούς μηχανισμούς που εισάγετε για να «λάβει» κάθε χρήστης το ρόλο του και τα «όρια» αρμοδιοτήτων που θεσπίζετε μεταξύ των ρόλων, σε φυσική γλώσσα (δε χρειάζεται να τα αναφέρετε όλα εξαντλητικά, αφήστε τις λεπτομέρειες για τις επόμενες ενότητες).

#####
#####

Υπάρχουν 3 ρόλοι στο σύστημα μας:

- a) **Administrator** (admin) -> εγγεγραμμένος χρήστης (signed user).
- b) **Client** -> signed user
- c) **Retailer** (εμπορικός συνεργάτης → λιανικής πώλησης ) -> signed user

// Ο παρακάτω είναι ο ανεπίσημος 4<sup>ος</sup>, δηλαδή μόνο οι 3 πρώτοι αποθηκεύονται στο σύστημα.

**Anonymous** (μη εγγεγραμμένος χρήστης) -> unsigned user

Οι **signed users** αποκτούν τον ρόλο που τους αντιστοιχεί, αφού κάνουν **login**. Το ρόλο τους, τον επιλέγουν κατά την εγγραφή. Εξαίρεση αποτελεί ο ρόλος του Administrator (χρησιμοποιείται μόνο από εμάς). Οι signed users, έχουν όλα τα δικαιώματα που έχει ένας anonymous user, και κάποια παραπάνω. Βέβαια οι 3 ρόλοι έχουν διαφορετικά δικαιώματα μεταξύ τους.

Ο χρήστης με το ρόλο **anonymous** (ή anonymous user) είναι ο **default ρόλος** που έχει κάποιος, πριν κάνει login. Μπορεί να κάνει απλή περιήγηση της πλατφόρμα μας, βλέποντας τα προϊόντα μας και τα profile των εμπορικών συνεργατών. Δεν έχει πρόσβαση στο ηλεκτρονικό πορτοφόλι, οπότε δεν μπορεί να κάνει αγορές.

#### **ΣΗΜΑΝΤΙΚΟ:**

- 1) Χρησιμοποιούμε την έννοια **Signed users** για να αναφερθούμε στους χρήστες που έχουν κάνει login (**Clients, Retailers, Admin**) για να αποφύγουμε να επαναλαμβάνουμε άσκοπα use cases (δεν αποτελεί ρόλος στο σύστημα μου ➔ group ρόλων).
- 2) Όταν γράφουμε **All users**, αναφερόμαστε σε όλους τους χρήστες ανεξαρτήτως ρόλου ( πχ μια λειτουργική απαίτηση είναι ίδια για όλους τους ρόλους).

Ο χρήστης με το ρόλο **Client** έχει πρόσβαση στο ηλεκτρονικό του πορτοφόλι. Μπορεί:

- 1) να **αναζητήσει** (προϊόντα, Retailers)
- 2) να **αγοράσει** προϊόντα
- 3) να **επεξεργαστεί το profile του.**
- 4) να έχει πρόσβαση στο προσωπικό του profile αλλά και στα profile των Retailers (κανένα άλλο profile πέρα από αυτά).
- 5) να **ακυρώσει την παραγγελία του.**
- 6) να δημιουργήσει **αίτηση επιστροφής του προϊόντος** (αν δεν είναι ικανοποιημένος).
- 7) να **αξιολογήσει** τα προϊόντα που έχει λάβει και τους αντίστοιχους εμπορικούς συνεργάτες.
- 8) **Να φορτίσει** το ηλεκτρονικό του πορτοφόλι με πόντους.
- 9) **Να αγοράσει πακέτα πόντων** (Customer Loyalty Award1) ή **να κερδίσει bonus points λόγω επίτευξης κάποιου milestone** (Customer Loyalty Award2).

Ο χρήστης με το ρόλο **Retailer** έχει και αυτός πρόσβαση στο ηλεκτρονικό του πορτοφόλι. Μπορεί:

- 1) να **αναζητήσει** (προϊόντα, Retailers)
- 2) να **επεξεργαστεί το profile** (της επιχείρησης του).
- 3) να έχει πρόσβαση στο προσωπικό του profile αλλά και στα profile των Retailers (κανένα άλλο profile πέρα από αυτά).
- 4) να εξυπηρετεί παραγγελίες από τους χρήστες
- 5) Ανάρτηση προϊόντων, επεξεργασία υπάρχων (για θέματα τιμής, διαθεσιμότητας, προσθήκη έκπτωσης κλπ)
- 6) **Να αποδεχθεί η απορρίψει 1 αίτημα επιστροφής.**
- 7) **Να μετατρέψει points->money**, για να εξαργυρώσει τα κέρδη του.

Ο χρήστης με ρόλο **Administrator** έχει **admin rights**, δηλαδή πλήρη πρόσβαση στην βάση δεδομένων.

Μπορεί να κάνει τα εξής:

- 1) να **αναζητήσει** (προϊόντα, Retailers)
- 2) να **επεξεργαστεί το profile του.**
- 3) Έχει πρόσβαση στα **profile όλων**.
- 4) Μπορεί να **διαγράψει** το profile ενός χρήστη (διαγράφοντας το χρήστη από τη βάση), εκτός του εαυτού του.
- 5) Μπορεί να **δημιουργήσει, επεξεργαστεί ή/και να διαγράψει** ένα πακέτο πόντων.

## 2.2 Λειτουργικές Απαιτήσεις

Σύντομη παράθεση των υποστηριζόμενων λειτουργιών που γίνονται αντιληπτές από τον χρήστη, προδιαγράφοντας τα προαπαιτούμενα (συνθήκες) εκτέλεσης, τα δεδομένα εισόδου και τα δεδομένα εξόδου.

Στην απλή περίπτωση λειτουργιών, όπου τα προαπαιτούμενα και τα δεδομένα εισόδου / εξόδου είναι προφανή, αρκούν 1-2 γραμμές περιγραφής της απαίτησης, σαν να ήταν User Story. Για παράδειγμα, αρκεί η διατύπωση: «Ο χρήστης με το ρόλο του Διαχειριστή θα μπορεί να απενεργοποιήσει το λογαριασμό οποιουδήποτε άλλου χρήστη», καθώς εδώ, προφανώς, η συνθήκη εκτέλεσης είναι ο συνδεδεμένος χρήστης να έχει το ρόλο του διαχειριστή, τα δεδομένα εισόδου είναι το id του υπό απενεργοποίηση χρήστη και δεδομένα εξόδου η τροποποιημένη εγγραφή του -πλέον απενεργοποιημένου- χρήστη. Για πιο σύνθετες λειτουργίες, επεκταθείτε στην περιγραφή έως τις 5-10 γραμμές, αποσαφηνίζοντας τις τρεις αυτές βασικές παραμέτρους (προαπαιτούμενα, δεδομένα εισόδου, δεδομένα εξόδου) όσο / όπου απαιτείται.

Τέλος, καλό είναι να κωδικοποιήσετε τις λειτουργικές απαιτήσεις (π.χ. ΛΑ01, ΛΑ02), ώστε να προσδιορίζονται μοναδικά. Θα μπορούσατε να το δομήσετε με τη μορφή πίνακα δύο στηλών (ID, TEXT).  
#####

### Εγγραφή:

- 1) **(ΛΑ01)** Ο χρήστης με ρόλο **anonymous Client**, συμπληρώνοντας:
  - a. Όνοματεπώνυμο
  - b. Username
  - c. password
  - d. email⇒ Δημιουργείται η αντίστοιχη εγγραφή, στη βάση δεδομένων χρηστών.
- 2) **(ΛΑ02)** Ο χρήστης με ρόλο **anonymous Retailer**, συμπληρώνοντας:
  - a. username
  - b. password
  - c. email
  - d. το όνομα της επιχείρησης
  - e. η κατηγορία που ανήκει
  - f. Διεύθυνση κεντρικού καταστήματος (για την επιστροφή των προϊόντων)
  - g. Περιγραφή της Επιχείρησης⇒ Δημιουργείται η αντίστοιχη εγγραφή, στη βάση δεδομένων χρηστών.

### Login:

- 1) **(ΛΑ03)** Ο χρήστης με ρόλο **anonymous** μπορεί να κάνει **Login** και να αποκτήσει το ρόλο που του αντίστοιχει, συμπληρώνοντας το username και password του.
- 2) **(ΛΑ04)** Ο χρήστης με ρόλο **anonymous**, αν βάλει λάθος στοιχεία θα του εμφανιστεί μήνυμα σφάλματος.

### Sign Out:

- 1) **(ΛΑ05)** Οι **signed users** μπορούν να κάνουν **αποσύνδεση** από το σύστημα, πατώντας την αντίστοιχη επιλογή στο αρχικό μενού (\home).

### Αναζήτηση Εμπορικών συνεργατών:

- 1) **(ΛΑ06)** All users μπορούν να εκτελούν αναζήτηση (με βάση ελεύθερο κείμενο) ➔ auto complete.
- 2) **(ΛΑ07)** All users ταξινομούν τα αποτελέσματα με διάφορα κριτήρια:
  - a. Πρώτα σε αξιολογήσεις (Best Reviews)
  - b. Πρώτα σε πωλήσεις. Δηλαδή εμπορικοί συνεργάτες με τις περισσότερες πωλήσεις προϊόντων (συνολικά).
- 3) **(ΛΑ08)** All users μπορούν να φίλτραρουν τα αποτελέσματα με διάφορα κριτήρια:
  - a. Κατηγορίες που ανήκουν

#### **Αναζήτηση Προϊόντων:**

- 1) **(ΛΑ09)** All users μπορούν να εκτελούν αναζήτηση (με βάση ελεύθερο κείμενο) ➔ autocomplete.
- 2) **(ΛΑ10)** All users ταξινομούν τα αποτελέσματα με διάφορα κριτήρια:
  - a. Best Reviews
  - b. Best Seller
- 3) **(ΛΑ11)** All users μπορούν να φιλτράρουν τα αποτελέσματα με διάφορα κριτήρια:
  - a. Τιμή (maximum)
  - b. Κατηγορίες
  - c. Εμπορικό Συνεργάτη

#### **Προϊόν:**

- 1) **(ΛΑ12)** Ολοι οι users μπορούν να δουν τις πληροφορίες ενός προϊόντος
  - a. Προσαπαιτούμενα: Κανένα! (ένας χρήστης με οποιοδήποτε ρόλο μπορεί να δει τις πληροφορίες του προϊόντος).
  - b. Δεδομένα Εισόδου: Χρειαζόμαστε το **id** του προϊόντος
  - c. Δεδομένα Εξόδου: Πληροφορίες για το συγκεκριμένο προϊόν και συγκεκριμένα:
    - i. ID
    - ii. Τιμή
    - iii. Όνομα προϊόντος
    - iv. Κατηγορία προϊόντος
    - v. Όνομα Εμπορικού συνεργάτη
    - vi. Ποσοστό έκπτωσης (προαιρετικό ➔ minimum final price 1 πόντος)
    - vii. Τεμάχια
- 2) **(ΛΑ13)** Ο χρήστης με ρόλο **Retailer** μπορεί να **τροποποιήσει** τις πληροφορίες του προϊόντος του.
- 3) **(ΛΑ14)** Ο χρήστης με ρόλο **Retailer** μπορεί να **αφαιρέσει** 1 προϊόν του.
- 4) **(ΛΑ15)** All users μπορούν να δουν τις **κριτικές** ενός προϊόντος.

#### **Αγορά Προϊόντος:**

- 1) **(ΛΑ16)** Ο χρήστης με το ρόλο **Client** μπορεί να εισάγει 1 προϊόν στο καλάθι του.
  - a. Προσαπαιτούμενα: Ο συνδεδεμένος χρήστης να έχει ρόλο του **Client**
  - b. Δεδομένα Εισόδου: ID προϊόντος
  - c. Δεδομένα Εξόδου: Δημιουργία καλαθιού με 1 προϊόν και δημιουργία νέας εγγραφής στη βάση δεδομένων καλαθιού.
- 2) **(ΛΑ17)** Ο χρήστης με το ρόλο **Client** μπορεί να προσθέσει επιπλέον προϊόντα στο καλάθι του.
  - a. Προσαπαιτούμενα: Ο συνδεδεμένος χρήστης να έχει ρόλο του **Client**
  - b. Δεδομένα Εισόδου: ID προϊόντος και ID καλαθιού (**basket**)
  - c. Δεδομένα Εξόδου: Τροποποίηση υπάρχοντος καλαθιού ➔ εισαγωγή καινούργιου προϊόντος στο καλάθι
- 3) **(ΛΑ18)** Ο χρήστης με το ρόλο **Client** μπορεί να **αφαιρέσει** προϊόντα από το καλάθι του.
- 4) **(ΛΑ19)** Ο χρήστης με το ρόλο **Client** μπορεί να **αδειάσει** το καλάθι του.
- 5) **(ΛΑ20)** Ο χρήστης με τον ρόλο **Client** μπορεί να οριστικοποιήσει και να υποβάλει την παραγγελία του ➔ **checkout**.
  - a. Προσαπαιτούμενα:
    - i. Ο συνδεδεμένος χρήστης να έχει ρόλο του **Client**
    - ii. Το καλάθι του να μην είναι άδειο
    - iii. Να υπάρχει επαρκές ποσό στο wallet του για την ολοκλήρωση της συναλλαγής
  - b. Δεδομένα Εισόδου: ID καλαθιού, Διεύθυνση παραλαβής, User ID (για να βρούμε το Wallet του)
  - c. Δεδομένα Εξόδου:
    - i. Δημιουργία επιμέρους παραγγελιών(**orders**) ανά **εμπορικό συνεργάτη**
    - ii. Προσθήκη εγγραφών/παραγγελιών στη βάση δεδομένων παραγγελιών(**order**)
    - iii. Προσθήκη εγγραφών/προϊόντων στη βάση δεδομένων **order line**/γραμμή παραγγελίας
    - iv. Μεταφορά του απαραίτημαν ποσού από το wallet του client ➔ στο wallet του εμπορικού συνεργάτη και στο wallet του admin.

Profile:

- 1) Client:
  - a. **(ΛΑ21)** Ο χρήστης με το ρόλο **Client** μπορεί να δει το προσωπικό του profile , καθώς και όλων των εμπορικών συνεργατών.
  - b. **(ΛΑ22)** Ο χρήστης με το ρόλο **Client** μπορεί να τροποποιήσει τις πληροφορίες του προσωπικού του profile και συγκεκριμένα:
    - i. Username
    - ii. Password
    - iii. Email
    - iv. Ονοματεπώνυμο
  - c. **(ΛΑ23)** Ο χρήστης με το ρόλο **Client** μπορεί να δει το **ιστορικό παραγγελιών** του.
  - d. **(ΛΑ24)** Ο χρήστης με το ρόλο **Client** μπορεί να δει τις **ενεργές παραγγελίες** του (Pending/Courier, not yet completed).
  - e. **(ΛΑ25)** Ο χρήστης με το ρόλο **Client** έχει πρόσβαση στα **Αιτήματα επιστροφής** του.
  - f. **(ΛΑ26)** Ο χρήστης με το ρόλο **Client** έχει πρόσβαση στο **ηλεκτρονικό του πορτοφόλι**.
  - g. **(ΛΑ27)** Ο χρήστης με το ρόλο **Client** έχει πρόσβαση στο **καλάθι αγορών** του.
- 2) Retailer:
  - a. **(ΛΑ28)** Ο χρήστης με ρόλο **Retailer** μπορεί να δει το προσωπικό του profile , καθώς και όλων των εμπορικών συνεργατών.
  - b. **(ΛΑ29)** Ο χρήστης με το ρόλο **Retailer** μπορεί να τροποποιήσει τις πληροφορίες του προσωπικού του profile και συγκεκριμένα:
    - i. Username
    - ii. Password
    - iii. Email
    - iv. Όνομα εταιρείας
    - v. Κατηγορίες που ανήκει (σαν επιχείρηση)
    - vi. Διεύθυνση κεντρικού καταστήματος
    - vii. Περιγραφή επιχείρησης
  - c. **(ΛΑ30)** Ο χρήστης με το ρόλο **Retailer** έχει πρόσβαση στα **Αιτήματα επιστροφής** που έχει δεχθεί.
  - d. **(ΛΑ31)** Ο χρήστης με το ρόλο **Retailer** μπορεί να δει τις **ενεργές παραγγελίες** (που δεν έχουν ολοκληρωθεί). ➔ μόνο τα orders που είναι pending , Courier
  - e. **(ΛΑ32)** All users έχουν πρόσβαση στο Product list του **Retailer**, δηλαδή τα προϊόντα που έχει ανεβάσει στη πλατφόρμα.
  - f. **(ΛΑ33)** All users έχουν πρόσβαση στα **Reviews** του **Retailer** (Retailer Reviews).
  - g. **(ΛΑ34)** Ο χρήστης με το ρόλο **Retailer** μπορεί να δει το **ιστορικό παραγγελιών** (δηλαδή τις παραγγελίες που έχουν ολοκληρωθεί) ➔ τα orders που είναι Complete.
  - h. **(ΛΑ35)** Ο χρήστης με το ρόλο **Retailer** έχει πρόσβαση στο **ηλεκτρονικό Πορτοφόλι** του.

Ηλεκτρονικό Πορτοφόλι: Όλες οι λειτουργικές απαιτήσεις προϋποθέτουν τη χρήση του **Payment Gateway Service**.

- 1) **(ΛΑ36)** Όλοι οι **signed users** μπορούν να φορτίσουν το ηλεκτρονικό τους πορτοφόλι με ένα συγκεκριμένο ποσό πόντων.
- 2) **(ΛΑ37)** Ο χρήστης με ρόλο **Retailer** μπορεί να μετατρέψει ένα ποσό των πόντων του σε χρήματα ( points ➔ χρήματα )
- 3) **(ΛΑ38)** Ο χρήστης με ρόλο **Admin** μπορεί να μετατρέψει ένα ποσό των πόντων του σε χρήματα ( points ➔ χρήματα )
- 4) **Customer Loyalty awards:**
  - a. **(ΛΑ39)** Όλοι οι **signed users** μπορούν να αγοράσουν **πακέτα πόντων** (10,20,50) και να κερδίσουν επιπλέον πόντους (2,5,10 αντίστοιχα).
  - b. **(ΛΑ40)** Ο χρήστης με ρόλο **Admin** μπορεί να **προσθέσει** 1 πακέτο πόντων.
  - c. **(ΛΑ41)** Ο χρήστης με ρόλο **Admin** μπορεί να **αφαιρέσει** 1 πακέτο πόντων.
  - d. **(ΛΑ42)** Ο χρήστης με ρόλο **Admin** μπορεί να **τροποποιήσει** 1 πακέτο πόντων.
  - e. **(ΛΑ43)** Όλοι οι **signed users** μπορούν να ελέγξουν αν δικαιούνται **milestone awards** και να κερδίσουν **bonus points**. Συγκεκριμένα τα milestone awards είναι:
    - i. **Πλήθος προϊόντων** που έχουν αγοράσει **συνολικά** ( 10,50,100... αγορές ➔ 20 bonus points)
    - ii. **Anniversary**, δηλαδή αν χρησιμοποιούν την πλατφόρμα μας για εκτεταμένη περίοδο χρόνου (yearly anniversary) ➔ 40 points

### Ιστορικό Παραγγελιών:

- 1) **(ΛΑ44)** Ο χρήστης με ρόλο **Client** ή **Retailer** μπορεί να δει τις παραγγελίες του που έχουν ολοκληρωθεί (**COMPLETED**)
  - a. **Προσαπαιτούμενα:** Ο συνδεδεμένος χρήστης πρέπει να έχει το ρόλο του **Client** ή **Retailer**
  - b. **Δεδομένα Εισόδου:** Χρειαζόμαστε το **id** της παραγγελίας
  - c. **Δεδομένα Εξόδου:** Πληροφορίες για τη συγκεκριμένη παραγγελία και συγκεκριμένα:
    - i. **ID**
    - ii. **Ημερομηνία**
    - iii. **Συνολικό κόστος**
    - iv. Τα **ID** των προϊόντων που αγοράσαμε/πουλήσαμε
    - v. Τις **τιμές** των προϊόντων κατά την αγορά τους (μπορεί να έχουν αλλάξει τώρα).
- 5) **(Πολιτική επιστροφής)**
  - a. **(ΛΑ45)** Ο χρήστης με ρόλο **Client** μπορεί να δημιουργήσει **Αίτημα επιστροφής**, εφόσον δεν έχουν περάσει 7 μέρες από την παραλαβή του, και να έχει παραλάβει τουλάχιστον 1 τεμάχιο του προϊόντος. Επιστρέφει το προϊόν με δικά του έξοδα χρησιμοποιώντας το **Courier Gateway Service**.
    - i. **Προσαπαιτούμενα:** Ο συνδεδεμένος χρήστης να έχει το ρόλο του **Client**, να έχει παραλάβει το προϊόν και να μην έχουν περάσει 7 μέρες από την παραλαβή του
    - ii. **Δεδομένα Εισόδου:** ID καλαθιού, ID παραγγελίας/order, ID προϊόντος/order line και την ποσότητα επιστροφής (πλήθος τεμαχίων που επιστρέφουμε).
    - iii. **Δεδομένα Εξόδου:** Δημιουργία **αιτήματος επιστροφής** (με τις παραπάνω πληροφορίες)
  - b. **(ΛΑ46)** Ο χρήστης με ρόλο **Client** στέλνει το προϊόν πίσω στο **Retailer**, χρησιμοποιώντας το **Courier Gateway Service**.  
(Η συνέχεια στα «Αιτήματα Επιστροφής»)

### Αιτήματα Επιστροφής

- 1) **(ΛΑ47)** Ο χρήστης με το ρόλο **Client** μπορεί να δει τις αιτήσεις που έχει κάνει και σε τι κατάσταση βρίσκονται (**Pending, Accepted, Declined**).
- 2) **(ΛΑ48)** Ο χρήστης με το ρόλο **Retailer** μπορεί να δει τις αιτήσεις που έχει δεχθεί.
- 3) **(ΛΑ49)** Ο χρήστης με ρόλο **Client** μπορεί να ακυρώσει 1 αίτημα επιστροφής.
- 4) **(ΛΑ50)** Ο χρήστης με το ρόλο **Retailer** μπορεί να αποδεχθεί (**accept**) ένα αίτημα, με αποτέλεσμα να επιστρέφεται το πόσο πόντων που του αντιστοιχεί στο πελάτη (για κάθε προϊόν αγοραστική αξία \* πλήθος τεμαχίων) και ενημερώνεται η κατάσταση του αιτήματος (**accepted**).
- 5) **(ΛΑ51)** Ο χρήστης με το ρόλο **Retailer** μπορεί να απορρίψει ένα αίτημα (λόγω κάποιας βλάβης του προϊόντος), με αποτέλεσμα να μην επιστραφούν χρήματα στο **Client** και τέλος ενημερώνεται η κατάσταση του αιτήματος (**declined**).

### Ενεργές Παραγγελίες:

- 1) **(ΛΑ52)**Ο χρήστης με το ρόλο **Client** ή **Retailer** μπορεί να δει τις παραγγελίες που είναι ενεργές (**Pending** ή **Courier**). Δηλαδή αυτές που είτε δεν έχουν εξυπηρετηθεί είτε βρίσκονται καθοδόν(**Courier**).
- 2) **(ΛΑ53)**Ο χρήστης με το ρόλο **Retailer** μπορεί να εξυπηρετήσει τις **PENDING παραγγελίες** και να τις στείλει στο **Courier**, με την προϋπόθεση ότι έχουν περάσει 24 ώρες από την υποβολή της παραγγελίας.
  - a. **Προσπατιούμενα:** Ο συνδεδεμένος χρήστης να έχει το ρόλο του **Retailer**, η παραγγελία να βρίσκεται σε κατάσταση **Pending** και να έχουν περάσει 24 ώρες από την υποβολή της παραγγελίας.
  - b. **Δεδομένα Εισόδου:** ID παραγγελίας, IBAN, Διεύθυνση παραλαβής, Διεύθυνση
  - c. **Δεδομένα Εξόδου:** Τροποποίηση εγγραφής στη βάση δεδομένων **order** ➔ τώρα η παραγγελία είναι σε κατάσταση **COURIER**
- 3) **(ΛΑ54)**Ο χρήστης με τον ρόλο **Retailer** ενημερώνει την παραγγελία/order (ως **Courier**), αφού παραδώσει την παραγγελία στο **Courier**.
- 4) **(ΛΑ55)**Ο χρήστης με τον ρόλο **Retailer** ενημερώνει την παραγγελία/order (ως **Completed**), μόλις ενημερωθεί (από το **Courier**) για την ημερομηνία και ώρα παραλαβής του προϊόντος.
- 5) **(Πολιτική ακύρωσης)**
  - a. **(ΛΑ56)**Ο χρήστης με το ρόλο **Client** μπορεί να ακυρώσει κάποιο προϊόν/**order line** της παραγγελίας του (μια συγκεκριμένη ποσότητα αυτού) πριν περάσουν 24 ώρες και να του επιστραφεί το αντίστοιχο ποσό.
    - i. **Προσπατιούμενα:** Ο χρήστης να έχει το ρόλο **Retailer**, να μην έχουν περάσει 24 ώρες από την υποβολή της παραγγελίας, και η παραγγελία να είναι σε κατάσταση **Pending**.
    - ii. **Δεδομένα Εισόδου:** ID καλαθιού, ID παραγγελίας, ID orderline, User ID (για να έχουμε πρόσβαση στο wallet του Client), αριθμός τεμαχίων προς ακύρωση
    - iii. **Δεδομένα Εξόδου:**
      1. Τροποποίηση εγγραφής στη βάση δεδομένων **orderline** ➔ ενημερώνεται η εγγραφή για πόσα κομμάτια του προϊόντος ακυρώθηκαν.
      2. Επιστροφή χρημάτων ➔ αγοραστική αξία \* πλήθος τεμαχίων + κόστος courier (όχι την προμήθεια)
  - b. **(ΛΑ57)**Ο χρήστης με το ρόλο **Client** μπορεί να ακυρώσει κάποιο προϊόν/**order line** της παραγγελίας του (μια συγκεκριμένη ποσότητα αυτού) αφού περάσουν 24 ώρες και να του επιστραφεί το αντίστοιχο ποσό.
    - i. **Προσπατιούμενα:** Ο χρήστης να έχει το ρόλο **Retailer** και να έχουν περάσει 24 ώρες από την υποβολή της παραγγελίας και η παραγγελία να είναι σε κατάσταση **Courier**.
    - ii. **Δεδομένα Εισόδου:** ID καλαθιού, ID παραγγελίας, ID orderline, User ID (για να έχουμε πρόσβαση στο wallet του Client), αριθμός τεμαχίων προς ακύρωση
    - iii. **Δεδομένα Εξόδου:**
      1. Τροποποίηση εγγραφής στη βάση δεδομένων **orderline** ➔ ενημερώνεται η εγγραφή για πόσα κομμάτια του προϊόντος ακυρώθηκαν.
      2. Επιστροφή χρημάτων ➔ αγοραστική αξία \* πλήθος τεμαχίων (όχι προμήθεια ούτε κόστος courier)

**Πολιτική Αξιολόγησης:**

- 1) **(ΛΑ58)**All users μπορούν να δουν τις κριτικές ενός προϊόντος
- 2) **(ΛΑ59)**All users μπορούν να δουν τις κριτικές ενός εμπορικού συνεργάτη.
- 3) **(ΛΑ60)**Ο χρήστης με το ρόλο **Client** μπορεί να κάνει μόνο 1 αξιολόγηση για κάθε προϊόν και κάθε εμπορικό συνεργάτη.
- 4) **(ΛΑ61)**Ο χρήστης με το ρόλο **Client** μπορεί να τροποποιήσει την αξιολόγηση του.
- 5) **(ΛΑ62)**Ο χρήστης με το ρόλο **Client** μπορεί να αφαιρέσει την αξιολόγηση του.
- 6) **(ΛΑ63)**Ο χρήστης με ρόλο **Client** μπορεί να αξιολογήσει 1 προϊόν/orderline, εφόσον έχει παραλάβει ή επιστρέψει, τουλάχιστον 1 τεμάχιο του προϊόντος.
- 7) **(ΛΑ64)**Ο χρήστης με ρόλο **Client** μπορεί να αξιολογήσει 1 εμπορικό συνεργάτη, εφόσον έχει αξιολογήσει 1 προϊόν του.

**Product list:**

- 1) **(ΛΑ65)**All users μπορούν να επιλέξουν 1 προϊόν από τη λίστα των προϊόντων ενός Retailer
- 2) **(ΛΑ66)**Ο χρήστης με το ρόλο **Retailer** μπορεί να αναρτήσει 1 καινούργιο προϊόν
  - a. Προσαπαιτούμενα: Ο χρήστης να έχει το ρόλο **Retailer**.
  - b. Δεδομένα Εισόδου: Χρειαζόμαστε τα εξής:
    - i. Αγοραστική τιμή (ορισμένη από τον εμπορικό συνεργάτη)
    - ii. Κατηγορία Προϊόντος
    - iii. Όνομα προϊόντος
    - iv. Όνομα Εμπορικού συνεργάτη
    - v. Ποσοστό έκπτωσης (προαιρετικό ➔ minimum final price 1 πόντος)
    - vi. Τεμάχια
  - c. Δεδομένα Εξόδου: Δημιουργία καινούργιας εγγραφής/προϊόντος στην βάση δεδομένων προϊόντων
- 3) **(ΛΑ67)**Ο χρήστης με το ρόλο **Retailer** μπορεί να κατεβάσει/σβήσει 1 υπάρχον προϊόν του.

### Admin:

- 1) **(ΛΑ68)** Ο χρήστης με ρόλο **Admin** μπορεί να μετατρέψει ένα ποσό των πόντων του σε χρήματα (**points->χρήματα**), με τη χρήση του **Payment Gateway Service**.
  - a. Με αυτό τον τρόπο κάνει εξαργύρωση των κερδών του, που οφείλεται στη προμήθεια του transaction/buy product. Εξακολουθεί να κερδίζει χρήματα με την φόρτιση των ηλεκτρονικών πορτοφολιών των υπολοίπων χρηστών (δηλαδή το χρηματικό payment πάει στο λογαριασμό του).
- 2) **(ΛΑ69)** Ο χρήστης με ρόλο **Admin** έχει **admin rights** (δηλαδή πλήρη δικαιώματα read/write) στη βάση δεδομένων της πλατφόρμας. Αυτό σημαίνει ότι μπορεί να δει, τροποποιήσει και να αφαιρέσει εγγραφές από τη βάση δεδομένων.
- 3) **(ΛΑ70)** Ο χρήστης με ρόλο **Admin** έχει πρόσβαση στη λίστα όλων των profile των χρηστών.
- 4) **(ΛΑ71)** Ο χρήστης με ρόλο **Admin** μπορεί να **σβήσει** το profile ενός χρήστη, εκτός από τον εαυτό του (διαγράφοντας τον έτσι, από τη βάση δεδομένων).

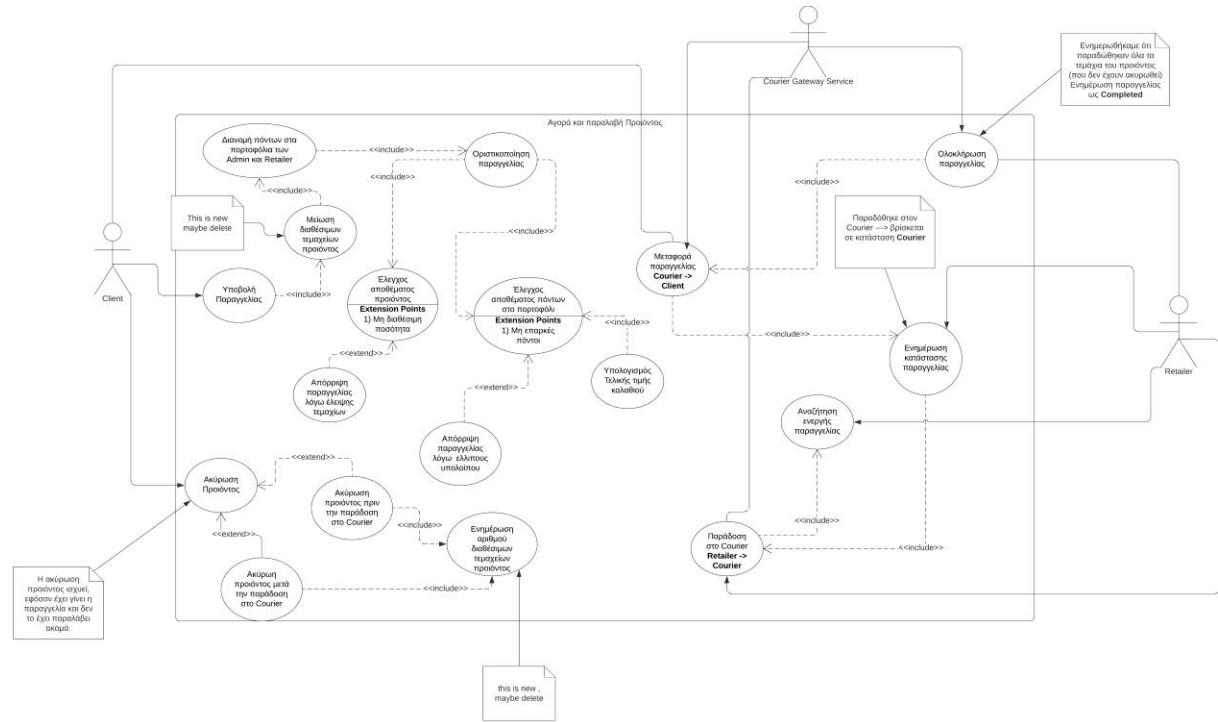
### **2.3 Σενάριο Χρήσης 1 –Αγορά Προϊόντος**

Σύντομη περιγραφή του σεναρίου χρήσης, δίνοντας έναν τίτλο και 1-2 γραμμές περιγραφής του ΚΣΕ (Κύριο Σενάριο Επιτυχίας). Ιδανικά, θα πρέπει να συνδέσετε το ΣΧ και το αντίστοιχο Use-Case diagram με τις συναφείς λειτουργικές απαιτήσεις. Για παράδειγμα, «ΣΧ01: Το σύστημα παρέχει έναν πίνακα ελέγχου (βλ. ΔΠΧ01) για την εκτέλεση των προβλεπόμενων διαχειριστικών ενεργειών (βλ. ΛΑ01, ΛΑ02) από τους χρήστες με το ρόλο του Διαχειριστή».

#####

**Περιγραφή:** Θέλουμε να δείξουμε τι γίνεται από τη στιγμή που ο **Client** πατάει υποβολή παραγγελίας, μέχρι να φτάσει το προϊόν στα χέρια του (μπορεί και όχι αν ακυρώσει την παραγγελία).

### 2.3.1 Διάγραμμα Περιπτώσεων Χρήσης1



### 2.3.2 Χειριστές (Actors)

Ποιο είναι οι πρωτεύοντες και δευτερεύοντες Actors;

#####

Primary Actor: **Client**

Secondary Actors: **Retailer, Courier Gateway Service**

### 2.3.3 Συνθήκη Εισόδου

Ποια συνθήκη πρέπει να ελεγχθεί από το σύστημα πριν εκκινήσει το σενάριο;

#####

Το καλάθι του χρήστη να μην είναι κενό και ο χρήστης να έχει το ρόλο **Client**.

### 2.3.4 Σκανδάλη

Ποιο συμβάν αποτελεί την αιτία εκκίνησης (trigger) του σεναρίου;

#####

Ο **Client** συμπληρώνει τη διεύθυνση του και πατάει υποβολή παραγγελίας => **checkout**.

### 2.3.5 Κύριο Σενάριο Επιτυχίας

Ποια είναι η αριθμημένη αλληλουχία βημάτων που πρέπει να εκτελεστούν στο ΚΣΕ;

#####

- 1) Υπολογισμός τελικής τιμής καλαθιού
- 2) Έλεγχος αποθέματος προϊόντος (αν είναι διαθέσιμη η ποσότητα που ζητάμε)
- 3) Έλεγχος αποθέματος πόντων στο ηλεκτρονικό πορτοφόλι του **Client**.
- 4) Οριστικοποίηση παραγγελίας και δημιουργία **Order** και **Orderline** στο σύστημα μας.
- 5) Διανομή πόντων στα πορτοφόλια του **Client** και του **Admin**. Παραδίδεται η προμήθεια στον admin και το υπόλοιπο ποσό στο **Retailer**.
- 6) Ο **Retailer** βρίσκει την δικιά μας παραγγελία , στο μενού **ενεργών παραγγελιών** του **Retailer**.
- 7) Ο **Retailer** χρησιμοποιεί το **Courier Gateway Service**, για την παράδοση της παραγγελίας.
- 8) Ο **Retailer** ενημερώνει την κατάσταση της παραγγελίας, σε κατάσταση **Courier**. Δηλαδή η παραγγελία είναι στα χέρια του **Courier**.
- 9) Ο **Courier** μεταφέρει το προϊόν και γίνεται παραλαβή από το **Client**.
- 10) Ο **Courier** ενημερώνει τον **Retailer** για την ημερομηνία/ώρα παραλαβής και ο **Retailer** ενημερώνει την κατάσταση παραγγελίας (ως **Complete**).

### 2.3.6 Επεκτάσεις στο ΚΣΕ

Ποια είναι τα εναλλακτικά σενάρια / επεκτάσεις στο ΚΣΕ; Σε ποια βήματα του ΚΣΕ προκύπτουν και υπό ποιες συνθήκες; Πως αλλάζει / επεκτείνεται η αλληλεπίδραση των χρηστών σε κάθε τέτοια περίπτωση (π.χ. εναλλακτική αριθμημένη λίστα βημάτων); Σε ποιο βήμα του ΚΣΕ μεταβαίνει ο χρήστης με την ολοκλήρωση της επέκτασης;

#####

- 2α) **Μη επαρκές απόθεμα προϊόντος.** Απορρίπτεται η παραγγελία και τερματίζει η διαδικασία.
- 3α) **Μη επαρκές απόθεμα πόντων.** Απορρίπτεται η παραγγελία και τερματίζει η διαδικασία.
- 5α) **Ακύρωση τεμαχίων προϊόντος** πριν παραδοθεί στο **Courier** (μέσα σε 24 ώρες). Δηλαδή αφού υποβάλει την παραγγελία ο **Client**, αποφασίζει να ακυρώσει κάποια τεμάχια του προϊόντος που παρήγγειλε (μέσω του μενού **ενεργών παραγγελιών** του Client). Επιστρέφονται **αγοραστική αξία + κόστος Courier**. Τέλος:
  - i. Αν εξακολουθούν να υπάρχουν τεμάχια προς αποστολή, συνεχίζουμε κανονικά από το βήμα 6
  - ii. Αν όχι, ενημερώνεται η παραγγελία (ως **Completed**) από το **Retailer** και η διαδικασία τερματίζει.

8α) **Ακύρωση τεμαχίων προϊόντος** αφού παραδοθεί στο **Courier**(αφού περάσουν 24 ώρες και ο **Retailer** έχει ενημερώσει τη κατάσταση παραγγελίας, ως **Courier**). Επιστρέφεται μόνο η **αγοραστική αξία**. Τέλος:

- i. Αν εξακολουθούν να υπάρχουν τεμάχια προς αποστολή, συνεχίζουμε κανονικά από το βήμα 9
- ii. Αν όχι, ενημερώνεται η παραγγελία (ως **Completed**) από το **Retailer** και η διαδικασία τερματίζει.

### 2.3.7 Εγγυήσεις

Ποια εξασφάλιση παρέχει το σύστημα με το πέρας του ΚΣΕ (εγγύηση επιτυχίας); Σε περίπτωση ενεργοποίησης εναλλακτικού σεναρίου, ποια είναι η ελάχιστη εξασφάλιση που αυτό παρέχει (ελάχιστη εγγύηση);

#####

Εγγύηση Επιτυχίας: Γίνεται σωστά η συναλλαγή των πόντων και ο **Client** παραλαμβάνει τη παραγγελία του.

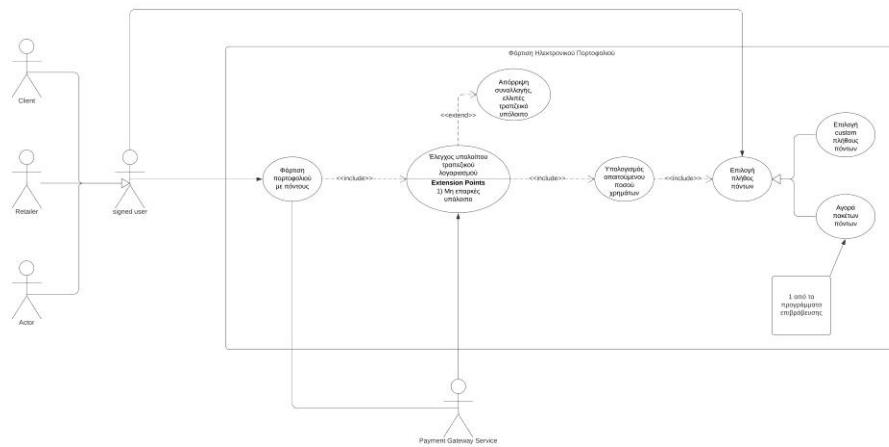
Εγγύηση Εναλλακτικού Σεναρίου:

- 1) Η παραγγελία **απορρίπτεται**, αν η ποσότητα του προϊόντος δεν είναι διαθέσιμη ή ο **Client** δεν έχει τους απαραίτητους πόντους.
- 2) Σε περίπτωση ακύρωσης ποσότητας/τεμαχίων του προϊόντος, του επιστρέφονται οι αντίστοιχοι πόντοι (ανάλογα με τις περιπτώσεις, που προδιαγράψαμε παραπάνω) .

## 2.4 Σενάριο Χρήσης 2 – Φόρτιση ηλεκτρονικού Πορτοφολιού

Περιγραφή: Θέλουμε να δείξουμε τη διαδικασία φόρτισης του ηλεκτρονικού πορτοφολιού του **signed user** ( Client or Retailer or Admin). Ο **Client** δίνει το **IBAN**/τραπεζικό λογαριασμό κατά την εκκίνηση του σεναρίου (ως δεδομένο εισόδου).

### 2.4.1 Διάγραμμα Περιπτώσεων Χρήσης 2



## 2.4.2 Χειριστές (Actors)

Primary Actor: **Signed user**

Secondary Actor: **Payment Gateway Service**

## 2.4.3 Συνθήκη Εισόδου

Ο χρήστης να έχει ρόλο **Client** ή **Retailer** ή **Admin**, δηλαδή να έχει κάνει login (να είναι **signed user**).

## 2.4.4 Σκανδάλη

Ο **signed user** να πατήσει το «φόρτιση ηλεκτρονικού πορτοφολιού» (στο προσωπικό του πορτοφόλι).

## 2.4.5 Κύριο Σενάριο Επιτυχίας

- 1) Ο **signed user** επιλέγει ένα **συγκεκριμένο πλήθος πόντων** προς αγορά και δίνει το **IBAN** του.
- 2) Γίνεται υπολογισμός του απαιτούμενου χρηματικού ποσού.
- 3) Ο **Payment Gateway Service (PGS)** ελέγχει αν υπάρχει το απαραίτητο ποσό στο λογαριασμό του **signed user**.
- 4) Ο **PGS** παίρνει τα χρήματα από το λογαριασμό και φορτίζει το ηλεκτρονικό πορτοφόλι του **signed user**.

## 2.4.6 Επεκτάσεις στο ΚΣΕ

1α) Ο **signed user** αγοράζει **πακέτα πόντων**, που του προσφέρουν **bonus** πόντους (ανάλογα με το πακέτο). Επιλέγει λοιπόν ποσότητα των πακέτων, δίνει πάλι το **IBAN** του και συνεχίζουμε κανονικά από το βήμα 2.

2α) **Μη επαρκές υπόλοιπο λογαριασμού**. Ο **PGS** απορρίπτει την συναλλαγή και η διαδικασία τερματίζει.

## 2.4.7 Εγγυήσεις

Εγγύηση Επιτυχίας: Θα γίνει σωστά η συναλλαγή **χρήματα->πόντους**, και θα φορτιστεί το ηλεκτρονικό πορτοφόλι του **signed user**, με το επιλεγμένο **custom** πλήθος πόντων.

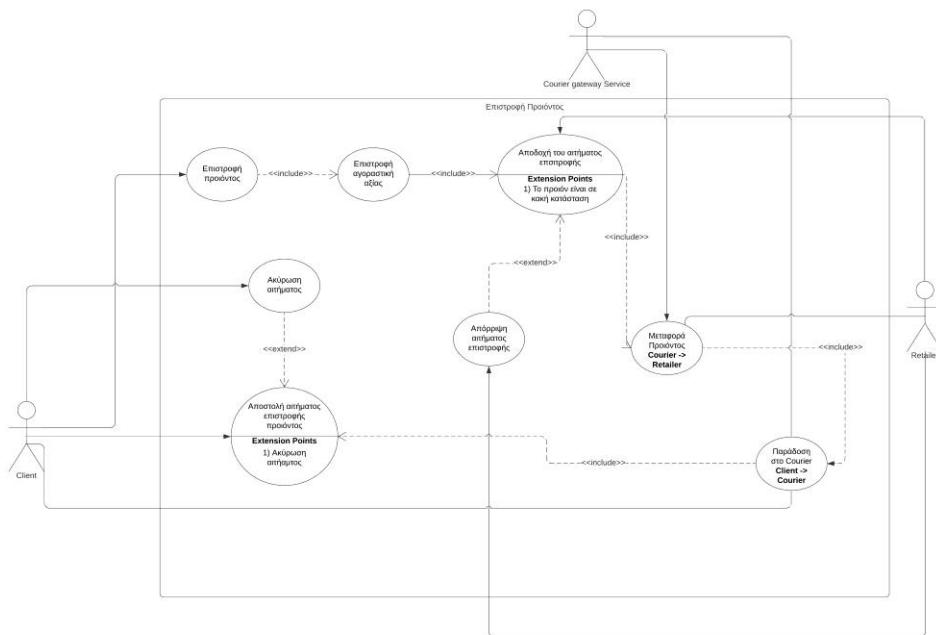
Εγγύηση Εναλλακτικού Σεναρίου:

- 1) Και σε αυτό το σενάριο φορτίζεται το πορτοφόλι του **signed user**, αλλά κερδίζει **bonus** πόντους, λόγω αγοράς **πακέτων πόντων**.
- 2) Αν δεν έχει επαρκές χρηματικό υπόλοιπο στο τραπεζικό λογαριασμό του, τότε ακυρώνεται η συναλλαγή.

## 2.5 Σενάριο Χρήσης 3 –Επιστροφή προϊόντος

Θα περιγράψουμε την διαδικασία όταν ο **Client** θέλει να επιστρέψει 1 προϊόν που έχει παραλάβει. Η διαδικασία αυτή, μπορεί να γίνει αν δεν έχουν περάσει 7 μέρες από την παραλαβή του.

### 2.5.1 Διάγραμμα Περιπτώσεων Χρήσης 3



## 2.5.2 Χειριστές (Actors)

Primary Actors: **Client**

Secondary Actors: **Retailer, Courier Gateway Service**

## 2.5.3 Συνθήκη Εισόδου

Ελέγχουμε αν έχουν περάσει 7 μέρες από την παραλαβή του προϊόντος. Αν ναι, τότε δεν μπορεί να γίνει η επιστροφή του.

## 2.5.4 Σκανδάλη

Ο **Client** πατάει επιστροφή προϊόντος, δίπλα από το προϊόν/**orderline** (που βρίσκεται στο **ιστορικό παραγγελιών** του Client).

## 2.5.5 Κύριο Σενάριο Επιτυχίας

- 1) Ο **Client** κάνει αίτηση επιστροφής του προϊόντος.
- 2) Ο **Client** χρησιμοποιεί το **Courier Gateway Service (CGS)**, και του δίνει το προϊόν προς αποστολή, μαζί με τη διεύθυνση του **Retailer**
- 3) Ο **CGS** παραδίδει το προϊόν στο **Retailer**.
- 4) Ο **Retailer** κάνει **accept** το αίτημα επιστροφής του **Client** και του επιστρέφεται η **αγοραστική αξία**.

## 2.5.6 Επεκτάσεις στο ΚΣΕ

1α) **Ακύρωση αιτήματος επιστροφής.** Ο **Client**, αν αλλάζει γνώμη, μπορεί να ακυρώσει το αίτημα επιστροφής του και η διαδικασία τερματίζει.

4α) **Απόρριψη αιτήματος επιστροφής.** Ο **Retailer** μπορεί να απορρίψει την αίτηση επιστροφής του **Client**, αν το προϊόν είναι σε άσχημη κατάσταση (δηλαδή έχει υποστεί βλάβη). Ο **Client** δεν παίρνει τους πόντους και η διαδικασία τερματίζει.

## 2.5.7 Εγγυήσεις

Εγγύηση Επιτυχίας: Θα επιστραφεί το προϊόν του **Client** και θα του δοθεί η αγοραστική αξία του προϊόντος.

Εγγύηση Εναλλακτικού Σεναρίου:

- 1) Ο **Client** ακυρώνει το αίτημα του, οπότε δεν γίνεται τίποτα.
- 2) Ο **Retailer** **απορρίπτει** την αίτηση και ο **Client** δεν παίρνει τους πόντους.

# 3 Δομημένη μοντελοποίηση διαδικασιών

Με βάση τη μεθοδολογία Δομημένης ανάλυσης και σχεδιασμού, να σχεδιάσετε τα ΔΡΔ της εφαρμογής έως το επίπεδο αποσύνθεσης 2 (χρησιμοποιώντας πολλαπλά επίπεδα ή/και δέντρο ιεραρχίας).

## 4 Αντικειμενοστρεφής μοντελοποίηση δεδομένων

Μοντελοποίηση των δεδομένων της εφαρμογής (application data model) με χρήση του αντικειμενοστρεφούς μοντέλου. Εδώ απαιτείται:

- η περιγραφή των βασικών οντοτήτων που πρέπει να υποστηρίζονται από το σύστημα (για παράδειγμα, Χρήστης, Προϊόν, Καλάθι Αγοράς, κλπ.) και να αποθηκεύονται στη βάση δεδομένων του.

#####

**Σημαντικό:** Το καλάθι αγοράς δεν αποθηκεύεται στη βάση. Είναι μέρος του **UserSession**, και αναλύεται περισσότερο στην ενότητα 7.

Θα παρουσιάσουμε τις οντότητες μόνο με τα χαρακτηριστικά(**attributes**) τους.

1. Wallet:

- int WalletID
- int amountPoints
- int UserID

2. User: Είναι **abstract class** με χαρακτηριστικά:

- int UserID
- Role role ( ο ρόλος του εγγεγραμμένου χρήστη: **Client, Retailer, Admin**)
- String Username
- String Password
- String Email
- Date RegDate (Registration Date, for Customer Loyalty 2 ➔ anniversary)

3. Client (απόγονος του User) έχει τα εξής χαρακτηριστικά(συνολικά):

- int UserID
- Role role ( ο ρόλος του εγγεγραμμένου χρήστη: **Client, Retailer, Admin**)
- String Username
- String Password
- String Email
- Date RegDate (Registration Date, for Customer Loyalty 2 ➔ anniversary)
- String NameSurname (Client-only attribute)

4. Retailer(απόγονος του User) έχει τα εξής χαρακτηριστικά(συνολικά):
  - a. int UserID
  - b. Role role ( ο ρόλος του εγγεγραμμένου χρήστη: **Client, Retailer, Admin**)
  - c. String Username
  - d. String Password
  - e. String Email
  - f. Date RegDate (Registration Date, for Customer Loyalty 2 ➔ anniversary)
  - g. String **CompanyName** (Retailer-only attribute)
  - h. Category **Categories** (οι κατηγορίες των προϊόντων που πουλάει ο **Retailer**, θα εμφανιστούν στο διάγραμμα). ➔ (Retailer-only attribute)
  - i. String **CompanyAddress** (Retailer-only attribute)
  - j. String **Description**
5. Admin (απόγονος του User) έχει τα εξής χαρακτηριστικά(συνολικά):
  - a. int UserID
  - b. Role role ( ο ρόλος του εγγεγραμμένου χρήστη: **Client, Retailer, Admin**)
  - c. String Username
  - d. String Password
  - e. String Email
  - f. Date RegDate (Registration Date, for Customer Loyalty 2 ➔ anniversary)
6. Point Pack. Πακέτα πόντων (**Customer Loyalty1**) που μπορούν να αγοράσουν οι κάτοχοι ηλεκτρονικού πορτοφολιού.
  - a. int PointPackID
  - b. int Price (σε χρήματα €)
  - c. int QuantityPoints (πόσοι πόντοι εμπεριέχονται στο πακέτο).

7. Order: Η παραγγελία μας σε 1 εμπορικό συνεργάτη. Περιέχει 1 ή περισσότερες γραμμές παραγωγής (Orderline).
- int OrderID
  - Date DateOrdered
  - Date DatePickup
  - State OrderState
    - Pending**: δεν έχει παραδοθεί στο Courier.
    - Courier**: παραδόθηκε στο Courier.
    - Completed**: Η παραγγελία ολοκληρώθηκε.
  - int Client\_ID
  - int Retailer\_ID
8. Orderline: Περιέχει το προϊόν και τη ποσότητα που έχει ζητήσει ο Client
- int Orderline\_ID
  - int Order\_ID
  - int Product\_ID
  - int Quantity
  - int ReturnQuantity
  - int CancelQuantity
9. Product:
- int ProductID
  - int Retailer\_ID
  - String ProductName
  - int Price
  - int AvailableAmount (πόσα τεμάχια είναι διαθέσιμα)
  - float Discount
  - Category category (Σε ποια κατηγορία retail ανήκει ➔ όλες οι κατηγορίες θα φανούν στο διάγραμμα).

10. Product Review: Αξιολόγηση προϊόντος

- a. int PReviewID
- b. int ProductID
- c. int ClientID
- d. String Text
- e. ReviewStars Stars ( ➔{1,2,3,4,5} )

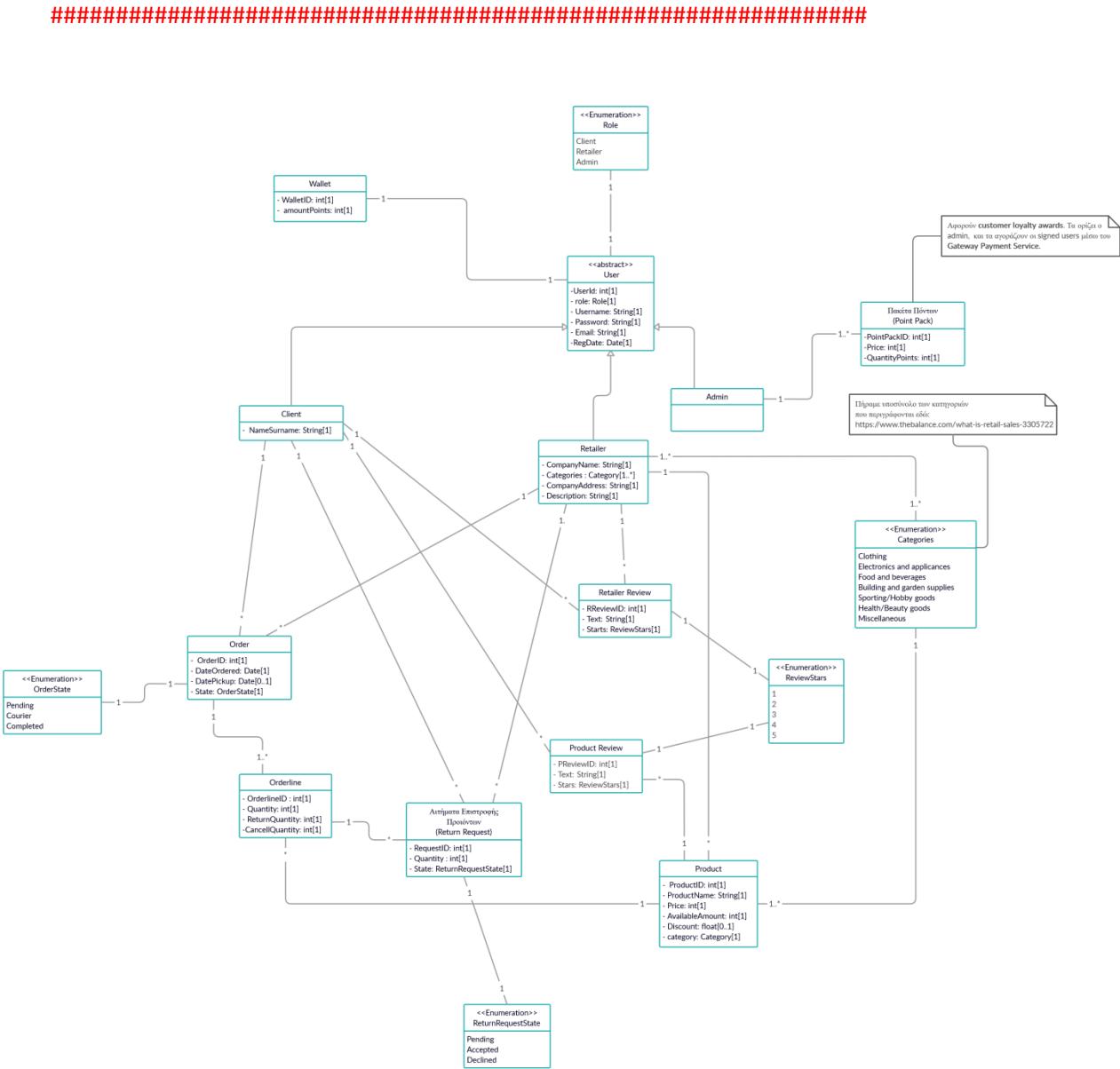
11. Retailer Review: Αξιολόγηση εμπορικού Συνεργάτη

- a. int RReviewID
- b. int ClientID
- c. int RetailerID
- d. String Text
- e. ReviewStars Stars ( ➔{1,2,3,4,5} )

12. Return Request: Αίτημα Επιστροφής προϊόντος. Χρησιμοποιείται κατά την διαδικασία επιστροφής του προϊόντος. Δηλαδή ο Client κάνει 1 αίτημα, στέλνει το προϊόν και περιμένει τον Retailer αν θα το δεχτεί ή όχι.

- a. int RequestID
- b. int OrderlineID
- c. int Quantity
- d. ReturnRequestState State
  - i. **Pending**: δεν το έχει εξυπηρετήσει ακόμα ο Retailer
  - ii. **Accepted**: αποδέχθηκε το αίτημα
  - iii. **Declined**: απέρριψε το αίτημα.

- η σχεδίασή τους ως κλάσεις, με την παράθεση του αντίστοιχου UML Class Diagram.



## 5 Αρχιτεκτονική του συστήματος

Το σύστημα που θα αναπτύξετε βασίζεται στην ευρύτερη Client – Server αρχιτεκτονική, όπου τον ρόλο του Client τον αναλαμβάνει η Frontend εφαρμογή και τον ρόλο του Server η Backend εφαρμογή. Η επικοινωνία μεταξύ των δύο εφαρμογών γίνεται μέσω ενός REST API, το οποίο θα πρέπει να περιγράψετε στη σχετική ενότητα.

Η Backend εφαρμογή βασίζεται στην πολύ-επίπεδη (multi-tier) αρχιτεκτονική, όπου ορίζονται τα εξής επίπεδα (από κάτω προς τα πάνω).

### 5.1 Επίπεδο βάσης δεδομένων

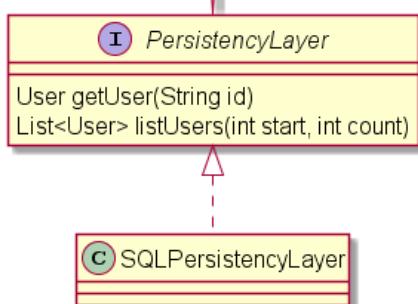
Το επίπεδο, στο οποίο λειτουργεί η ΒΔ του συστήματος. Εδώ κάνουμε την παραδοχή ότι υπάρχει ένα Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS), το οποίο μπορεί να λειτουργεί στο ίδιο ή σε διαφορετικό μηχάνημα σε σχέση με τα υπόλοιπα επίπεδα της Backend εφαρμογής.

### 5.2 Επίπεδο προσπέλασης δεδομένων

Το επίπεδο, το οποίο είναι υπεύθυνο για την προσπέλαση στη ΒΔ. Δηλαδή, στο επίπεδο αυτό έχει ανατεθεί η σχεδιαστική / τεχνική αρμοδιότητα (design / technical concern) της προσπέλασης στα δεδομένα (π.χ. εκτέλεση των SQL επερωτήσεων) και της «μετάφρασης» των αποτελεσμάτων σε αντικείμενα του μοντέλου δεδομένων της εφαρμογής.

Ειδικότερα, μπορείτε να θεωρήσετε ότι στο επίπεδο προσπέλασης δεδομένων υπάρχουν οι κλάσεις που παρουσιάζονται στο ακόλουθο διάγραμμα και, στην αντίστοιχη ενότητα, να ορίσετε επακριβώς το PersistencyLayer interface (απλή παράθεση των μεθόδων του). Οι κλάσεις που θα αναφέρονται ως παράμετροι ή αποτελέσματα στις μεθόδους αυτές θα πρέπει να έχουν παρουσιαστεί στο διάγραμμα κλάσεων του μοντέλου δεδομένων.

Οι λειτουργίες που περιέχονται στο Interface είναι ενδεικτικές, εσείς θα πρέπει να ορίσετε τις μεθόδους αυτές αναλυτικά, με τρόπο που να συνάδει με τις πραγματικές ανάγκες ανάκτησης / αποθήκευσης των δεδομένων της εφαρμογής σας.



### 5.3 Επίπεδο επιχειρησιακής λογικής

Το επίπεδο, το οποίο είναι υπεύθυνο για την επιχειρησιακή λογική της εφαρμογής σας. Δηλαδή, στο επίπεδο αυτό έχει ανατεθεί η σχεδιαστική / τεχνική αρμοδιότητα (design / technical concern) της ενσωμάτωσης των επιχειρησιακών λεπτομερειών της εφαρμογής σας (τρόπος χρέωσης χρηστών, τρόπος εκτέλεσης παραγγελίας, κλπ.).

Για τη διαπίστευση και τη δικαιοδοσία των χρηστών, μπορείτε να θεωρήσετε δεδομένη την κλάση UserSession, η δομή της οποίας φαίνεται στο ακόλουθο διάγραμμα.

 UserSession
static UserSession newSession(String userId) String getUserId() boolean hasRole(String role) void endSession()

Η κλάση αυτή «ξέρει» να διαβάζει και να γράφει τις πληροφορίες του τρέχοντος χρήστη στα αιτήματα και τις απαντήσεις που ανταλλάσσονται μεταξύ της Frontend και της Backend εφαρμογής. Ακολουθεί μια σύντομη περιγραφή των μεθόδων της:

- Η στατική μέθοδος newSession καλείται για να δημιουργήσει ένα νέο session για τον χρήστη με τον δοθέντα μοναδικό προσδιοριστή (άρα πρέπει να κληθεί μετά την επιτυχημένη σύνδεση του χρήστη, δηλαδή μετά το επιτυχές του login, όποτε και με όποιο τρόπο αυτό λάβει χώρα στη Backend εφαρμογή).
- Η μέθοδος getUserId επιστρέφει τον μοναδικό προσδιοριστή του διαπιστευμένου χρήστη.
- Η μέθοδος hasRole ελέγχει αν ο διαπιστευμένος χρήστης έχει τον δοθέντα ρόλο.
- Η μέθοδος endSession τερματίζει το session του διαπιστευμένου χρήστη (άρα πρέπει να κληθεί για να επισημάνει την αποσύνδεση του χρήστη).

#### 5.4 Επίπεδο Υποδοχής Αιτημάτων –Αποστολής Απαντήσεων

Το επίπεδο, το οποίο είναι υπεύθυνο για την υποδοχή των αιτημάτων (REST requests) για συγκεκριμένους πόρους (REST resources) και την παραγωγή των απαντήσεων (REST responses) με τη μορφή JSON (το μόνο REST representation που θα πρέπει να υποστηρίζετε στην εφαρμογή σας). Δεδομένου ότι στο πρότυπο Model-View-Controller, το επίπεδο αυτό αντιστοιχεί στο Controller, μπορείτε να θεωρήσετε δεδομένη την ύπαρξη μιας κλάσης RestController, η οποία είναι δομημένη όπως φαίνεται στο ακόλουθο διάγραμμα.

 RestController
PersistencyLayer database SuccessResponse success ErrorReponse error UserSession session
JsonRepresentation get() JsonRepresentation put() JsonRepresentation post() JsonRepresentation delete()
final String getURLAttribute(String name) final String getURLQueryParam(String name) final Map<String, String> getAllURLQueryParams() final String getRequestBodyParam(String name) final Map<String, String> getAllRequestBodyParams()

Ειδικότερα, η κλάση RestController:

- Παρέχεται από τον Application Server που φιλοξενεί την Backend εφαρμογή της πλατφόρμας και είναι υπεύθυνη για τη δρομολόγηση κάθε αιτήματος εξυπηρέτησης για κάποιο endpoint / route που ζητείται από την Frontend εφαρμογή. Για παράδειγμα, το αίτημα HTTP GET "/admin/users/32" θα δρομολογηθεί στην αντίστοιχη μέθοδο (get) της κατάλληλης υποκλάσης του RestController.
- Περιέχει τέσσερις μεθόδους (get, put, post, delete), κάθε μία από τις οποίες, σχεδιαστικά, αποτελεί το entry-point για την υλοποίηση της «εξυπηρέτησης» του αντίστοιχου HTTP / REST αιτήματος και την παραγωγή του JSON representation που θα επιστραφεί ως απάντηση. Η κλάση JsonRepresentation μπορεί κι αυτή να θεωρηθεί δεδομένη (οι λεπτομέρειες της κωδικοποίησης των δεδομένων της εφαρμογής σε μορφή JSON δεν μας απασχολεί στο έγγραφο).
- Περιέχει πέντε βοηθητικές μεθόδους οι οποίες παρέχουν πρόσβαση στις παραμέτρους του HTTP / REST αιτήματος.
- Περιέχει τέσσερα "έτοιμα προς χρήση" αντικείμενα (τύπου PersistencyLayer, SuccessResponse, ErrorResponse και UserSession), τα οποία τηρούνται αυτόματα από τον Application Server στα αντίστοιχα attributes (database, success, error, session).

Πρακτικά, η κλάση είναι σχεδιασμένη να αποτελέσει το όχημα για την υλοποίηση οποιουδήποτε custom Rest API. Δεδομένου ότι η default υλοποίηση των τεσσάρων βασικών μεθόδων της RestController (get, put, post, delete) περιέχει τον εξής κώδικα:

```
return error.generate(405, "Method not allowed")
```

η κλάση είναι σχεδιασμένη να επεκταθεί ώστε επαναπροσδιοριστούν (override) μία ή περισσότερες από τις βασικές μεθόδους. Για παράδειγμα, έστω ότι η κλάση UserController υλοποιεί την εξυπηρέτηση των αιτήσεων που δρομολογούνται στο "/admin/users/{id}" endpoint. Ο ψευδοκώδικας της κλάσης αυτής παρουσιάζεται στο ακόλουθο σχήμα.

```
@Route("/admin/users/{id}")
class UserController extends RestController {
    @Override
    JsonRepresentation get() {
        if (session == null) {
            return error.generate(401, "Not logged in")
        }
        if(!session.hasRole('ADMIN')) {
            return error.generate(403, "Forbidden")
        }
        // The user is logged in and has the appropriate role
        String userId = getUserIdAttribute("id")
        User user = database.getUser(userId)
        if (user == null) {
            return error.generate(404, "User $id not found")
        }
        return success.generate(user)
    }
}
```

Όπως φαίνεται στο σχήμα, η κλάση UserController επεκτείνει την RestController για να επαναπροσδιορίσει τη συμπεριφορά της μεθόδου get και να υλοποιήσει την ανάκτηση των στοιχείων του χρήστη με το διθέν id κωδικοποιημένο ως URL attribute.

Δηλαδή, επιτρόσθετα των κλάσεων που θα σχεδιάσετε και θα εντάξετε στο επίπεδο της επιχειρησιακής λογικής, θα πρέπει να προδιαγράψετε και τις διάφορες υποκλάσεις της RestController που υλοποιούν τη λήψη των αιτημάτων και την παραγωγή των απαντήσεων για τα REST endpoints του API σας. Επιτρόσθετα, θα πρέπει να προδιαγραφούν και οι μέθοδοι της SuccessResponse κλάσης, ως το κεντρικό σημείο προδιαγραφής των δεδομένων εξόδου της Backend εφαρμογής.

Όλα τα παραπάνω αποτελούν μια επισκόπηση των βασικών σχεδιαστικών αποφάσεων που θεωρούνται κοινές για όλες τις ομάδες / εργασίες. Σε επόμενες ενότητες θα κληθείτε να συμπληρώσετε τα «κενά» με βάση τη δική σας σχεδιαστική προσέγγιση.

## 6 Ποιοτικά Χαρακτηριστικά και Μη-Λειτουργικές Απαιτήσεις

Στην ενότητα αυτή αναφέρετε τα καίρια ποιοτικά χαρακτηριστικά της εφαρμογής και τις μη-λειτουργικές της απαιτήσεις, σε ό,τι αφορά ευχρηστία, απόδοση, ασφάλεια, διαλειτουργικότητα κλπ.

### 6.1 Γενικά χαρακτηριστικά

- Η εφαρμογή μας να υποστηρίζει πολλές γλώσσες.
- Να υποστηρίζει διαφορετικές συσκευές, όπως **υπολογιστής, κινητό, tablet**.
- Να υποστηρίζει διαφορετικούς **browsers**, όπως Mozilla, Chrome, Safari, Edge, TOR.

### 6.2 Απαιτήσεις ευχρηστίας

- Να πληκτρολογεί ο χρήστης όσο το δυνατόν λιγότερο.
- Οι διαδικασίες να αυτοματοποιούνται και να υπάρχει ελάχιστο user input.
- Όλες οι επιλογές του χρήστη, να προκύπτουν από **drop-down menu, pop-up window** ή **buttons**.
- Κάθε φορά να υπάρχουν μέχρι 3 pop-up windows ταυτόχρονα.
- Να υπάρχει **Φωνητική Υποστήριξη** για την αναζήτηση (προϊόντων ή εμπορικών συνεργατών).
- Θέλουμε οι ιστοσελίδες να είναι εύκολα προσβάσιμες. Δηλαδή από την ιστοσελίδα μας, στον προορισμό, να έχουμε το πολύ 3 βήματα (click).

### 6.3 Απαιτήσεις ασφάλειας

- Σύστημα ασφάλειας για την προστασία των username και password. Δηλαδή θα κρυπτογραφεί και θα αποκρύπτει αυτά τα δεδομένα.
- Κάθε χρήστης δεν θα μπορεί να φτιάχνει πάνω από 1 λογαριασμό.
- Σύστημα ασφαλείας που θα εξασφαλίζει τον διαχωρισμό των ρόλων. Δηλαδή 1 χρήστης να μην μπορεί να εκτελέσει ενέργειες που δεν του έχουν παραχωρηθεί.
- Email επιβεβαίωσης στις κινήσεις των Wallet των χρηστών. Δηλαδή, όταν ο χρήστης θέλει να φορτίσει το πορτοφόλι του, θα του έρχεται πρώτα email επιβεβαίωσης, πριν προχωρήσει.

#### 6.4 Απαιτήσεις διαλειτουργικότητας (διασύνδεσης με τρίτα συστήματα)

- Τα εξωτερικά μας συστήματα είναι 2: Payment Gateway Service, Courier Gateway Service.
- Να υπάρχει **έλεγχος εγκυρότητας IBAN** από το Payment Gateway Service.
- Η επικοινωνία της εφαρμογής μας με τα τρίτα συστήματα να είναι κρυπτογραφημένη και συγκεκριμένα να ακολουθεί το πρωτόκολλο του **TOR** (onion routing).
- Θέλουμε έγκαιρη και έγκυρη ενημέρωση των χρηστών για την κατάσταση των προϊόντων τους, από το Courier Gateway Service.

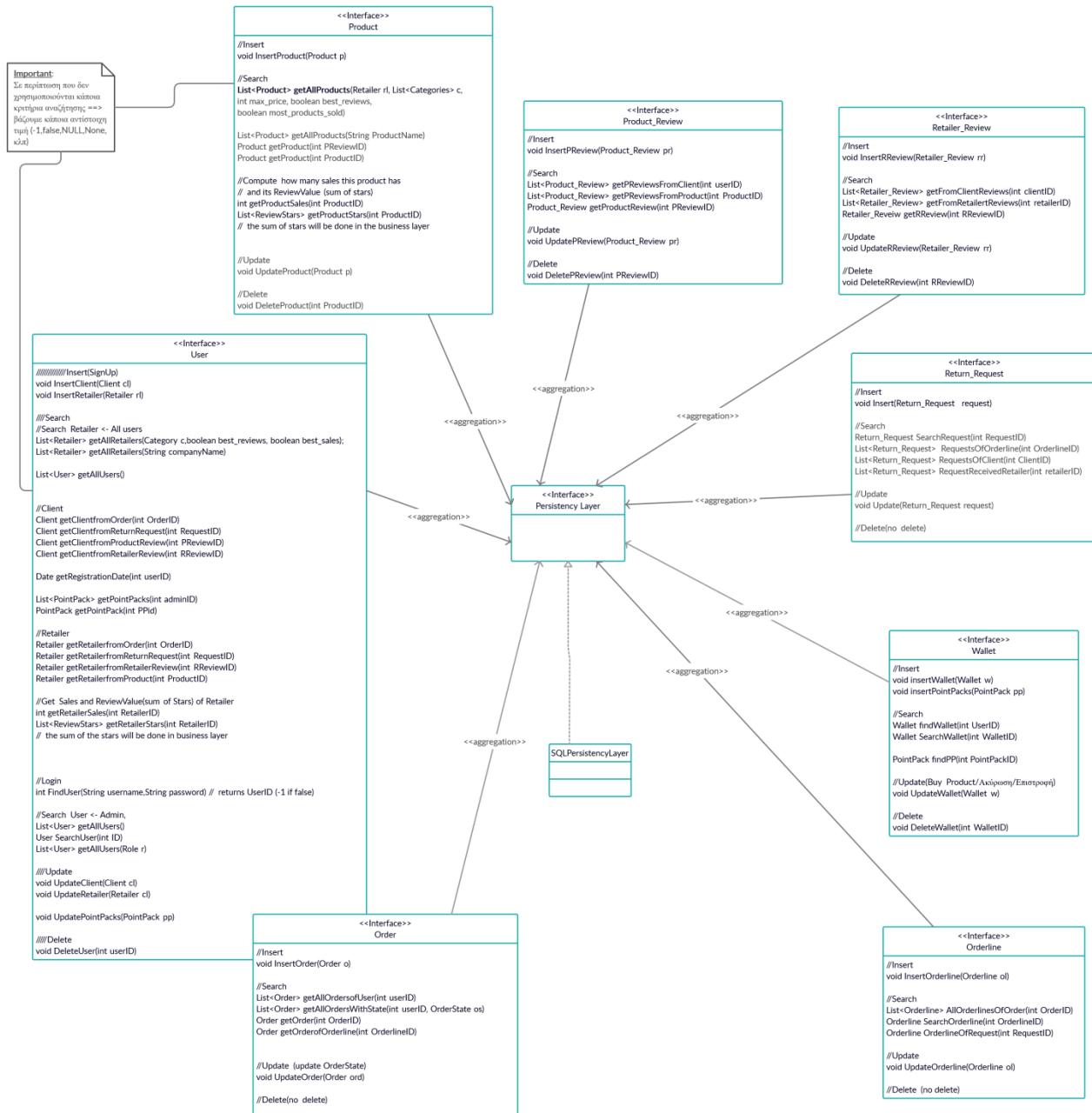
#### 6.5 Άλλες Απαιτήσεις

# 7 Προδιαγραφές της Backend Εφαρμογής

## 7.1 Σχεδιασμός του Επιπέδου Προσπέλασης Δεδομένων

Στην ενότητα αυτή θα πρέπει να παραθέσετε το UML Class diagram για το PersistencyLayer interface (ουσιαστικά, δηλαδή, να προδιαγράψετε τις απαραίτητες για την εφαρμογή σας μεθόδους του).

#####



## 7.2 Σχεδιασμός του Επιπέδου Επιχειρησιακής Λογικής

Στην ενότητα αυτή θα πρέπει να αποτυπώσετε τη σχεδιαστική σας προσέγγιση για την ανάπτυξη της επιχειρησιακής λογικής της εφαρμογής. Συγκεκριμένα, θα πρέπει να παραθέσετε:

1. Μια σύντομη περιγραφή των κλάσεων που απαρτίζουν το επίπεδο της επιχειρησιακής λογικής (εκτός της UserSession, που σας δίνεται έτοιμη). Τα στιγμιότυπα των κλάσεων αυτών -κατά κανόνα- δεν αποθηκεύονται στη ΒΔ (οι κλάσεις αυτές δεν μοντελοποιούν, δηλαδή, τα δεδομένα, αλλά τις διαδικασίες, ενέργειες, ελέγχους κ.λπ. που συναποτελούν το business logic της εφαρμογής).

#####
#####

Παρόλο που κάποιες από τις παρακάτω classes έχουν attributes, δεν «αποθηκεύουν» δεδομένα. Τις βάλαμε απλά για να δείξουμε τα δεδομένα που έχουμε, για την επίλυση του αντίστοιχου προβλήματος.

**Σημαντικό:** Εδώ παρουσιάζουμε το καλάθι αγοράς(**Basket**).

- Το **Basket** αποφασίσαμε να είναι μέρος του **UserSession**, δηλαδή «ζει» όσο βρίσκεται ενεργός ο **Client** στο σύστημα, και δημιουργείται κατά το Login.
- Το καλάθι είναι διαθέσιμο μόνο για τους **Clients**.
- Κάθε χρήστης έχει 1 καλάθι κάθε φορά.
- Μπορεί να προσθέτει, αφαιρεί προϊόντα(**Basket Component**) ή να το αδειάσει πλήρως.
- Μπορεί να κάνει **Checkout**, δηλαδή να καταθέσει τη παραγγελία του. Αφού πληρούνται οι συνθήκες, γίνονται οι εξής 2 ενέργειες:
  - Χωρίζονται τα **Basket Components**, ανά εμπορικό συνεργάτη(int Retailer ID) ➔ φτιάχνονται τα **Orders**.
  - Για κάθε **unique προϊόν** του **Order** ➔ φτιάχνουμε το αντίστοιχο **Orderline**.
  - Τέλος αδειάζουμε το καλάθι.
  - Αναλυτικά θα διατυπωθεί η διαδικασία, με 1 **Sequence Diagram**.

**The Classes:**

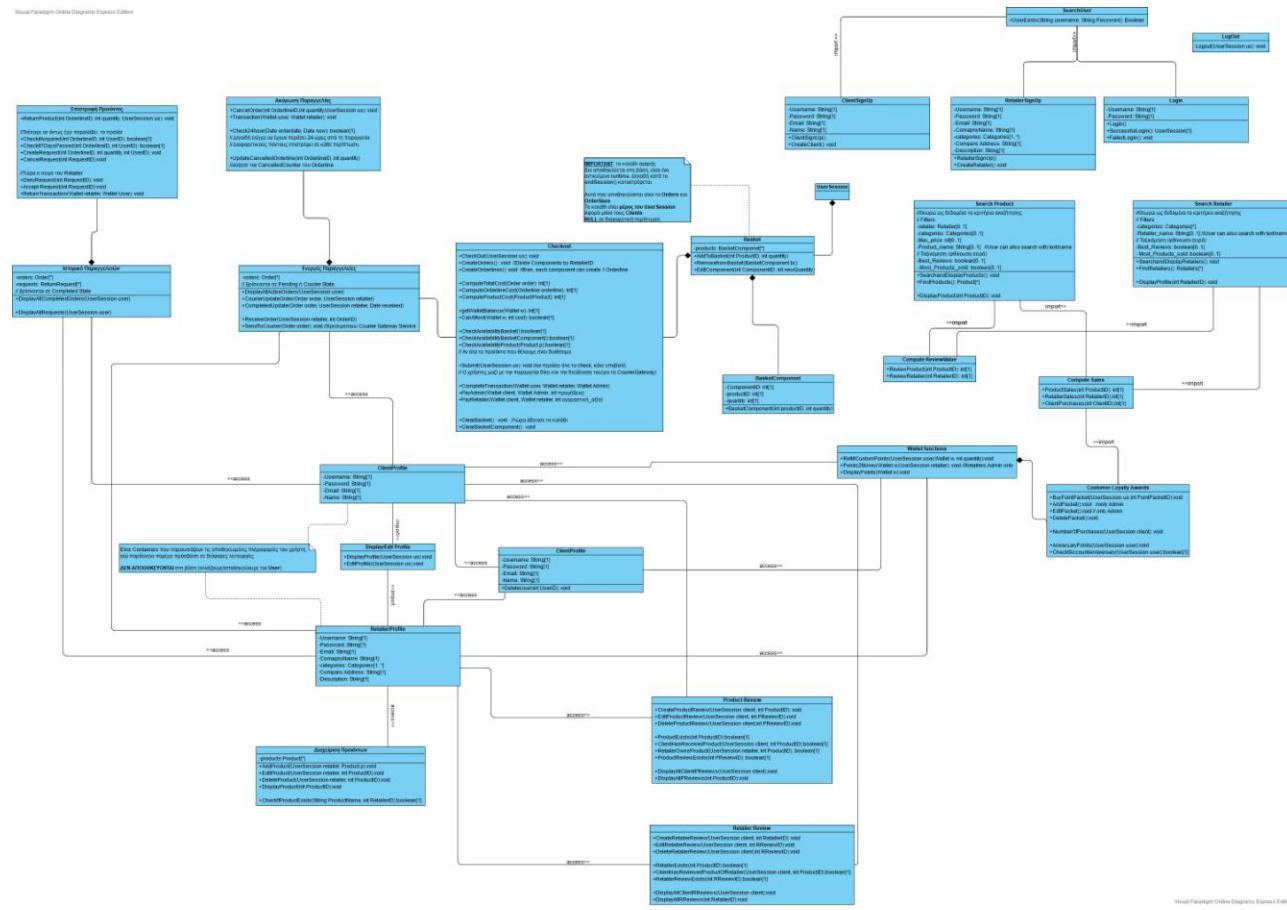
- 1) RetailerSignUp, ClientSignUp, Login: Όπως φαίνεται και από τα ονόματα τους, αφορούν τα προβλήματα **Εγγραφής** (με ρόλο Retailer ή Client) αλλά και **Εισόδου** του χρήστη.
  - a. Ο κύριος έλεγχος που κάνουν και οι 3, είναι το **UserExists(username,password)**, δηλαδή αν είναι ή όχι αποθηκευμένος ο user στο σύστημα μας.

- 2) Search Product: Αφορά την αναζήτηση προϊόντων. Εδώ υπάρχουν φίλτρα καθώς και τρόποι ταξινόμησης.
- Συγκεκριμένα για τους τρόπους ταξινόμησης, και οι 2 βασίζονται σε **παραγόμενες ιδιότητες**(δηλαδή δεν υπάρχουν έτοιμες στη δομή Product), **salesCount(int)** και **reviewValue(int)**.
  - Για αυτό χρησιμοποιούμε τις κλάσεις **ComputeReviewValue** και **ComputeSales**, για να κάνουμε τους υπολογισμούς μας.
- 3) Search Retailer: Αφορά την αναζήτηση εμπορικών συνεργατών. Παρόμοιο πρόβλημα με το παραπάνω , με μικρές διαφορές. Για το **SalesCount(int)** μετράμε πωλήσεις συνολικά για όλα τα προϊόντα. Για το **reviewValue** μετράμε τις αξιολογήσεις του εμπορικού συνεργάτη (και όχι του προιόντος).
- 4) Basket: Έχουμε αναφερθεί πιο πάνω. Δες το «Σημαντικό».
- 5) Active Orders: Εδώ μας ενδιαφέρουν οι ενεργές παραγγελίες (**Pending, Courier**). Συγκεκριμένα:
- Η προβολή και ενημέρωση της κατάστασης των παραγγελιών (**Courier, Completed** ).
  - Η **ακύρωση μέρους του προϊόντος** (ακύρωση κομματιού του Orderline).
    - Γίνεται έλεγχος αν έχουν περάσει 24 ώρες, και επιστρέφονται πόντοι στο **Client** ανάλογα με τη περίπτωση.
- 6) Order History: Εδώ μας ενδιαφέρουν οι ολοκληρωμένες παραγγελίες (**Completed**).  
Συγκεκριμένα:
- Η προβολή όλων των ολοκληρωμένων παραγγελιών.
  - Η δημιουργία **Return Request**, δηλαδή αίτημα επιστροφής κάποιου προϊόντος.
    - Γίνεται έλεγχος αν όντως έχει παραλάβει το προϊόν → **CheckIfAquired()**. Δηλαδή δεδομένου Orderline και completed παραγγελίας, υπάρχει τουλάχιστον 1 μη ακυρωμένο τεμάχιο του προϊόντος, που έχει παραλάβει.
    - Γίνεται έλεγχος αν έχουν περάσει 7 μέρες από τη παραλαβή του. Αν ναι, δεν επιτρέπεται η δημιουργία του.
    - Αποθηκεύεται η αίτηση στο σύστημα, και ο **Retailer** , μπορεί να κάνει **AcceptRequest() or DenyRequest()**.
    - Αν κάνει accept του επιστρέφονται πόντοι, αλλιώς τίποτα.
  - Μπορεί ο **Client** , αν θέλει να ακυρώσει την αίτηση του → **Cancel Request**.
- 7) RetailerProfile, ClientProfile, AdminProfile: Τα profiles είναι containers που παρουσιάζουν τις αποθηκευμένες πληροφορίες του χρήστη, ενώ παράλληλα παρέχει πρόσβαση σε διάφορες λειτουργίες.
- ΔΕΝ αποθηκεύονται στη βάση, αλλά οι πληροφορίες των κλάσεων **Client/Retailer/Admin**.
- 8) Product Management: Αφορά την διαχείριση προϊόντων (άρα χρησιμοποιείται από **Retailers**). Δίνει τη δυνατότητα **προσθήκης, επεξεργασίας, ή/και διαγραφής προϊόντος** (μεταξύ άλλων ενεργειών).

- 9) Product Review: Αφορούν τις αξιολογήσεις των προϊόντων. Συγκεκριμένα:
- Γίνεται έλεγχος αν έχει παραλάβει τουλάχιστον 1 τεμάχιο του προϊόντος.
  - Γίνεται έλεγχος αν έχει κάνει άλλη αξιολόγηση για το ίδιο προϊόν (δεν του επιτρέπεται να φτιάξει 2<sup>η</sup> αξιολόγηση).
  - Μπορεί να δημιουργήσει, επεξεργαστεί ή/και να διαγράψει την αξιολόγησή του.
- 10) Retailer Review: Παρόμοιο με το παραπάνω. Η μόνη διαφορά είναι στη 1<sup>η</sup> συνθήκη, όπου ο Client, πρέπει να έχει αξιολογήσει τουλάχιστον 1 προϊόν του εμπορικού συνεργάτη.
- 11) Wallet functions: Αφορά τις λειτουργίες του ηλεκτρονικού πορτοφολιού. Αρκετές από τις λειτουργίες χρειάζονται την παροχή **IBAN**(τραπεζικού λογαριασμού του χρήστη). Συγκεκριμένα:
- Επιτρέπει την **φόρτιση** του ηλεκτρονικού πορτοφολιού (με την παροχή του **IBAN** του χρήστη).
  - Επιτρέπει την μετατροπή **πόντων ➔ χρήματα** (μόνο για τους Retailers και Admin).
  - Εμπεριέχει λειτουργίες του **Customer Loyalty Awards**:
    - Παρέχει τη δυνατότητα αγοράς **πακέτων πόντων** (ειδικές προσφορές).
    - Παρέχει τη δυνατότητα στον Admin, να **προσθέσει, επεξεργαστεί ή να αφαιρέσει τα πακέτα πόντων**.
    - Γίνεται έλεγχος για την επίτευξη κάποιου **milestone**. Δηλαδή αν είναι τα γενέθλια του λογαριασμού (**anniversary**), ή αν έχουν γίνει παραπάνω από συγκεκριμένο ποσό αγορών. Και στις 2 περιπτώσεις κερδίζει **bonus points**.

2. Ένα UML Class diagram των κλάσεων που απαρτίζουν το επίπεδο της επιχειρησιακής λογικής.

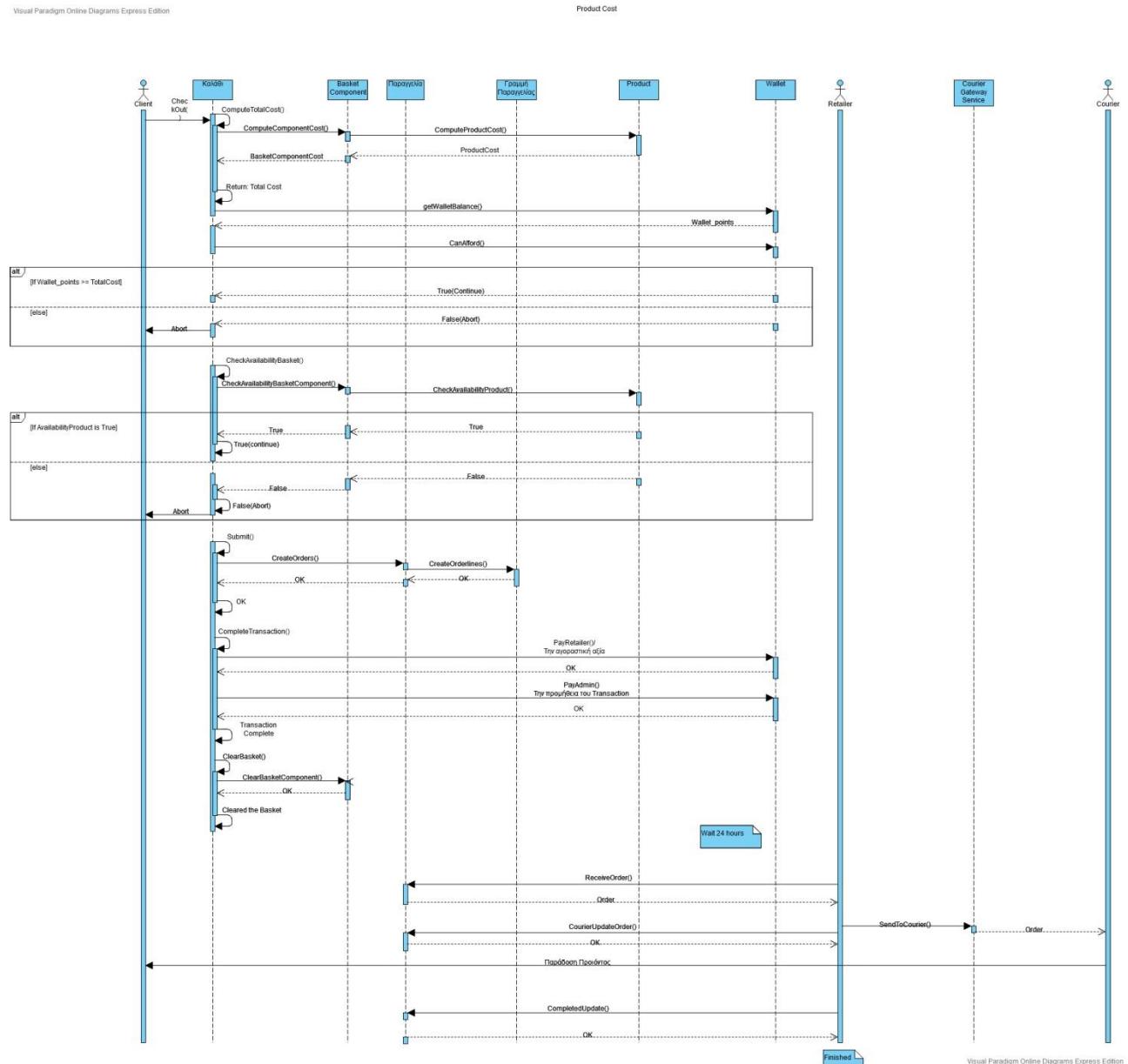
#####



3. Ένα UML Sequence diagram για την εκτέλεση μιας παραγγελίας από έναν χρήστη

#####

Ουσιαστικά θα δείξουμε την διαδικασία, από την εκκίνηση του **Checkout**, μέχρι να φράσει στα χέρια του **Client**.

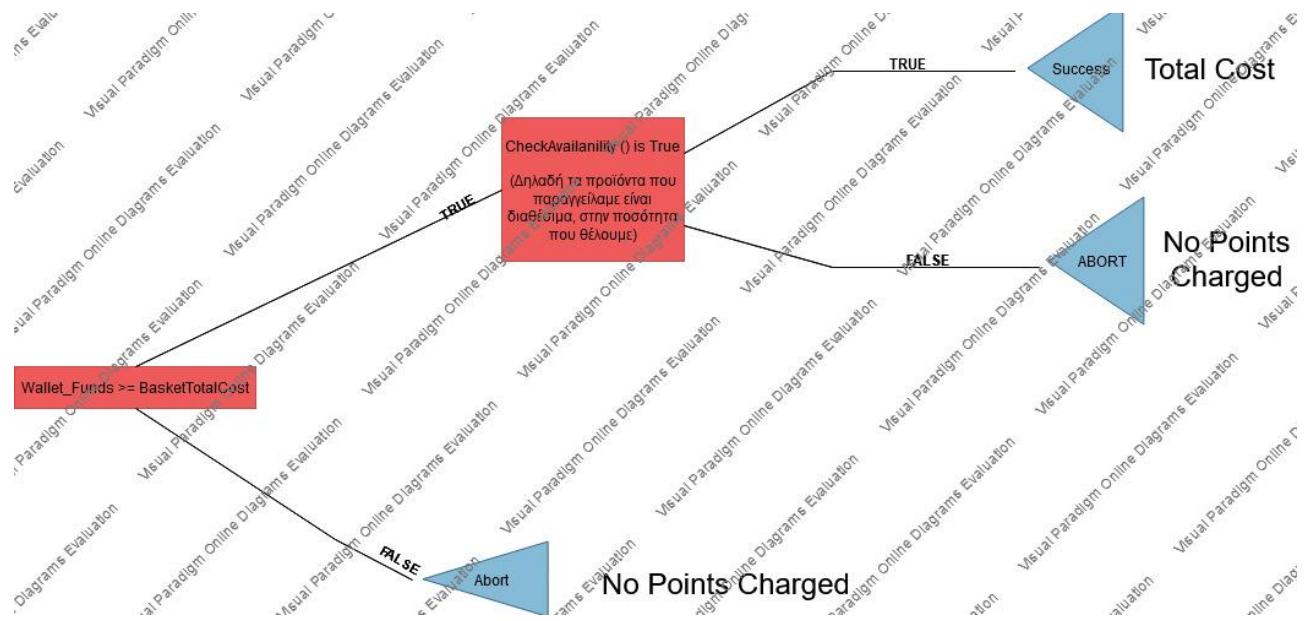


4. Ένα δέντρο απόφασης για τον τρόπο χρέωσης του πορτοφολιού του χρήστη.

#####

Όταν μιλάμε για «χρέωση του πορτοφολιού του χρήστη», μιλάμε για την χρέωση κατά το **Checkout**. Για να γίνει η χρέωση πρέπει να πληρούνται 2 συνθήκες.

(Συγχωρέστε την εικόνα, το Visual Paradigm κάνει aggressive marketing).



### 7.3 Σχεδιασμός του Επιπέδου Υποδοχής Αιτημάτων & Αποστολής Απαντήσεων

Στην ενότητα αυτή θα πρέπει να προδιαγράψετε το REST API της πλατφόρμας σας. Ειδικότερα, θα πρέπει να παραθέσετε:

1. Μια σύντομη περιγραφή του κάθε REST endpoint, τις υποστηριζόμενες HTTP μεθόδους του και τα δεδομένα εισόδου του ανά περίπτωση (π.χ. ονόματα και τύποι των query parameters όταν γίνεται GET, όταν γίνεται PUT κ.ο.κ.). Για την παραγωγή των αποτελεσμάτων, θα πρέπει απλά να αναφέρετε τη λειτουργία ή / λειτουργίες της κλάσης SuccessResponse που χρησιμοποιεί το κάθε REST endpoint (εδώ, μας ενδιαφέρουν μόνο τα signatures των μεθόδων, που όλες θα είναι παραλλαγές της generate).

#####

### Rest Endpoints:

- 1) “/home” **GET**. Η αρχική σελίδα.
  - a. Input: 1 query parameter ➔ **bool logged\_in** (παράγει διαφορετικές καρτέλες στην αρχική σελίδα)
  - b. Success: generate(String msg) ➔ “Welcome to our site”
- 2) “/client\_SignUp” **GET, POST**. Δημιουργία λογαριασμού **client**.
  - a. GET:
    - i. Input: No query parameters
    - ii. Success: generate(String msg) ➔ “Please fill the form below”
  - b. POST:
    - i. Input: Username, Password, Email, NameSurname
    - ii. Success: generate(String msg) ➔ “Client Sign up complete!”
- 3) “/Retailer\_Signup” **GET, POST**. Δημιουργία λογαριασμού **Retailer** (εμπορικού συνεργάτη)
  - a. GET:
    - i. Input: no query parameters
    - ii. Success: generate (String use) -> “please fill the form below”
  - b. POST:
    - i. Input: Username, Password, Email, Company name, Categories, Company address, Description
    - ii. Success: generate (String msg) -> “Retailer Sign up complete”
- 4) “/search\_products” **GET**. Αναζήτηση προϊόντος.
  - a. GET: Εμφάνισε τα προϊόντα
    - i. Input: query parameters
      - Product\_name
      - Retailer
      - Categories
      - Max\_price
      - Best\_reviews
      - Most\_product\_sold
    - ii. Success: generate (List <Product> product)

5) “/profile/{user\_id}/{product\_id}” **GET, PUT, POST, DELETE**

- a. **GET**: Εμφανίζει τις πληροφορίες του προϊόντος
  - o **Input:** no query parameters
  - o **Success:** generate (Product product)
- b. **PUT**: O **Retailer** αλλάζει τις πληροφορίες του προϊόντος
  - o **Input:**
    - Product Name
    - Price
    - Available Amount
    - Category
    - Discount
  - o **Success:** generate (Product updated Product)
- c. **POST**: Add to basket
  - o **Input:**
    - int ProductID,
    - int quantity
- d. **DELETE**: Διαγραφή προϊόντος (μόνο από owner/Retailer)
  - o **Input:** no query parameters
  - o **Success:** generate (String msg) -> “Deletion Complete”

6) “/search\_retailer” **GET**. Αναζήτηση εμπορικού συνεργάτη.

- a. **GET**:
  - i. **Input:** query parameters
    - Categories
    - Retailer\_name
    - Best\_Reviews
    - Most\_Product\_sold
  - ii. **Success:** generate (List <Retailer> retailer )

7) “/log\_out” **POST**. Delete session of logged in User

- a. **POST**:
  - i. **Input:** UserSession
  - ii. **Success:** generate (String msg) -> “Log out Successful”

8) “/login” **POST**.

- a. **Input:** Username, Password
- b. **Success:** generate(String msg) ➔ “Log out Successful”

9) “/profile/{user\_ID}” **GET,PUT,DELETE**. Τα profile των χρηστών. Delete profile → delete user.

a. GET: Γενικά ισχύει ο εξής κανόνας, για το ποιος έχει access σε ποια profile.

Av client\_profile <- owner/Client, admin.

Av retailer\_profile <- all

i. Input: No query parameters

ii. Success: generate (Client client) or generate (Retailer retailer)

b. PUT: Edit profile (only the owner of profile)

i. Input:

- Username
- Password
- Email
- Name <- Client profile only
- Categories <- Retailer profile only
- Company name <- Retailer profile only
- Company Address <- Retailer profile only
- Description <- Retailer profile only

ii. Success: generate (Client updatedClient) or generate (Retailer updatedRetailer)

c. DELETE: Delete Profile (means delete user) <- Only Admin

i. Input: No query parameters

ii. Successful: generate (String msg) -> “User Removed”

10) “/profile/{user\_id}/products” **GET, POST**.

a. GET: Εμφάνισε τη λίστα προϊόντων του Retailer

i. Input: no query parameters

ii. Success: generate (List<Product> products)

b. POST: Add new Product (Retailer)

i. Input:

- ProductName
- Price
- Available Amount
- Discount
- Category

ii. Success: generate (List<Product> updatedProducts)

11) “/profile/{user\_id}/Basket” **GET,DELETE**. Το Basket δεν αποθηκεύεται στη βάση, είναι κομμάτι του UserSession.

- **GET:** Εμφάνισε το περιεχόμενο του καλαθιού
  - **Input:** no query parameters
  - **Success:** generate (List<BasketComponent> basket)
- **DELETE:** Clear Basket (Remove Everything)
  - **Input:** no query parameters
  - **Success:** generate (List<BasketComponent> emptyBasket)

12) “/profile/{user\_id}/Basket/{component\_id}” **GET, PUT, DELETE**.

- **GET:** Εμφάνισε τις πληροφορίες του Basket Component, δηλαδή το ProductID και τη ποσότητα quantity.
  - i. **Input:** no input parameters
  - ii. **Success:** generate (BasketComponent bc)
- **PUT:** Edit quantity of Basket Component
  - i. **Input:**
    - 1. int ComponentID
    - 2. int newQuantity
  - ii. **Success:** generate (BasketComponent updated BC)
- **DELETE:** Remove from Basket
  - i. **Input:** no input parameters
  - ii. **Success:** generate (string msg) -> “Removed from basket”

13) “/profile/{user\_id}/Basket/Checkout” **POST..**

- **POST:** Εκτέλεση διαδικασίας checkout.
  - i. **Input:** Διεύθυνση παραλαβής
  - ii. **Success:** generate (string msg) -> “Checkout complete”

14) “/profile/{user\_id}/ActiveOrders” **GET.**

- **GET:** Εμφάνισε όλες τις ενεργές παραγγελίες
  - **Input :** no query parameters
  - **Success :** generate (List<Order> orders)

15) “/profile/{user\_id}/ActiveOrders/{Order\_id}” **GET, PUT.**

- **PUT:** Αλλαξε την κατάσταση της παραγγελίας (σε courier or completed)
  - **Input:**
    - i. OrderState os
    - ii. int order\_id
  - **Success:** generate (Order updatedOrder)
- **GET:** Δείξε το περιεχόμενο της παραγγελείας
  - **Input:** no query parameters
  - **Success:** generate (Order order)

16) “/profile/{user\_id}/ActiveOrders/{order\_id}/{orderline\_id}” **GET, PUT.**

- **GET:** Εμφάνισε το περιεχόμενο του Orderline
  - **Input :** no query parameters
  - **Success:** generate (Orderline orderline)
- **PUT:** Ακύρωσε μία ποσότητα του orderline/ προϊόντος.
  - **Input:**
    - i. int ordeline\_id
    - ii. int quantity
  - **Success:** generate (Orderline updatedOrderline)

17) “/profile/{user\_id}/OrderHistory” **GET**

- **GET:** Δείχνει όλες τις παραγγελίες που είναι Completed
  - **Input:** no query parameters
  - **Success:** generate (List<Order> orders)

18) “/profile/{user\_id}/OrderHistory/{Order\_id}” **GET.**

- **GET:** Δείχνει όλες τις λεπτομέρειες μιας παραγγελίας
  - **Input:** no query parameters
  - **Success:** generate (Order order)

19) “/profile/{user\_id}/OrderHistory/{OrderLine\_id}” **GET, POST**

- **GET:** Δείχνει όλες τις λεπτομέρειες ενός Orderline.
  - **Input:** no query parameters
  - **Success:** generate (OrderLine orderLine)
- **POST:** Δημιουργία **Return Request**, για επιστροφή ποσότητας του προϊόντος.
  - **Input:**
    - i. int orderline\_id
    - ii. int quantity
  - **Success:** generate (String msg) ➔ “Created Refund Request”

20) “/profile/{user\_id}/ReturnRequests” **GET**

- **GET:** Δείχνει όλες τις αιτήσεις επιστροφής
  - **Input:** no query parameters
  - **Success:** generate (List<Return\_Request> rr)

21) “/profile/{user\_id}/ReturnRequests/{Request\_id}” **GET, PUT, DELETE**

- **GET:** Δείχνει όλες τις πληροφορίες του αιτήματος
  - **Input:** no query parameters
  - **Success:** generate (Return\_Request rr)
- **PUT:** Αποδοχή ή απόρριψη του αιτήματος
  - **Input:** (ReturnRequestState newState)
  - **Success:** generate (ReturnRequest updated)
- **DELETE:** Ακύρωση αιτήματος επιστροφής από το χρήστη (το αίτημα γίνεται **declined**).
  - **Input:** no query parameters
  - **Success:** generate (ReturnRequest updated)

22) “/profile/{user\_id}/products/{Product\_id}/Reviews” **GET, POST**

- **GET:** Εμφάνισε όλα τα reviews του προϊόντος
  - **Input:** no query parameters
  - **Success:** generate (List<ProductReviews> reviews )
- **POST:** Φτιάξε ένα καινούργιο **Product** Review
  - **Input:**
    - i. ReviewStars stars
    - ii. String text
  - **Success:** generate (List<ProductReview> updatedReviews)

23) “/profile/{user\_id}/products/{Product\_id}/Reviews/{Review\_id}” **GET, PUT, DELETE.**

- **GET:** Εμφάνισε όλα τις πληροφορίες του product review
  - **Input:** no query parameters
  - **Success:** generate (ProductReview review )
- **PUT:** Edit review
  - **Input:**
    - i. ReviewStars newstars
    - ii. Text newtext
  - **Success:** Generate (ProductReview updated)
- **DELETE:** Delete Review
  - **Input:** no query parameters
  - **Success:** generated (String msg) -> “Product Review Deleted”

24) “/profile/{user\_id}/Reviews” **GET, POST**

- **GET:** Εμφάνισε όλα τα reviews του Retailer
  - **Input:** no query parameters
  - **Success:** generate (List<RetailerReviews> rr )
- **POST:** Φτιάξε ένα καινούργιο Retailer review
  - **Input:**
    - i. ReviewStars stars
    - ii. String text
  - **Success:** generate (List<RetailerReview> updatedRR)

25) “/profile/{user\_id}/Reviews/{Review\_id}” **GET, PUT, DELETE**

- **GET:** Εμφάνισε όλο το περιεχόμενο του Retailer review
  - **Input:** no query parameters
  - **Success:** generate (Retailer Review rr)
- **PUT:** Edit review
  - **Input:**
    - i. ReviewStars stars
    - ii. String Text
  - **Success:** Generate (RetailerReview updated)
- **DELETE:** Delete Review
  - **Input:** no query parameters
  - **Success:** generated (String msg) -> “Retailer Review Deleted”

26) “/profile/{user\_id}/Wallet/{Wallet\_id}” **GET, PUT, POST**

- **GET:** Δείχνει όλες τις πληροφορίες του πορτοφολιού
  - **Input:** no query parameters
  - **Success:** generate (Wallet wallet)
- **PUT:** φόρτισε/μετέτρεψε σε χρήμα το ηλεκτρονικό πορτοφόλι(custom value)
  - **Input:**
    - i. Int wallet\_id
    - ii. Int num\_of\_points
    - iii. String IBAN
    - iv. Int choice
  - **Success:** generate (Wallet updatedWallet)
- **POST:** Ελέγχει αν ο χρήστης μπορεί να κερδίσει BONUS POINTS (Customer Loyalty 2)
  - **Input:** int wallet\_id
  - **Success:** generate (String msg) -> “Milestone reached => Bonus Points”

27) “/profile/{user\_id}/Wallet/{wallet\_id}/CLA/PointPacks” **GET, POST**

- **GET:** Εμφάνισε όλα τα PointPacks
  - **Input:** no query parameters
  - **Success:** generate (List<PointPacks> pp )
- **POST:** Add pack (μόνο από admin)
  - **Input:**
    - i. int Price
    - ii. int num\_of\_points (in the pack)
  - **Success:** generate (List<PointPack> updatedPP)

28) “/profile/{user\_id}/Wallet/{Wallet\_id} /CLA/PointPacks/{PointPack\_id}” **GET, PUT, POST, DELETE**

- **GET:** Εμφάνισε όλες τις πληροφορίες του PointPack
  - **Input:** no query parameters
  - **Success:** generate (PointPack pack)
- **POST:** Αγορά Πλακέτου.
  - **Input:** no query parameters
  - **Success:** generate(String msg) ➔ “Point Pack bought”
- **PUT:** Edit PointPack (μόνο από ADMIN)
  - **Input:**
    - i. int Price
    - ii. int Points\_Contained
  - **Success:**
    - i. Generate (PointPack updated)
    - ii. Generate (String msg) -> “PointPack updated”
- **DELETE:** Delete PointPack
  - **Input:** no query parameters
  - **Success:** generated (String msg) -> “PointPack Deleted”

29) “/profile” GET.

- GET: Εμφάνισε όλα τα profile ( δηλαδή όλους τους χρήστες).
  - Input: no query parameters
  - Success: generate(List<User> users)

2. Μια ενδεικτική υλοποίηση σε ψευδοκώδικα του Controller που διαχειρίζεται τη σύνδεση των χρηστών (login).

#####

```
@Route("/login")
Class LoginController extends RestController{

    @Override
    JsonRepresentation post(){

        If(session != NULL){
            Return error.generate(403,"Already Logged in")
        }

        String username = getRequestBodyparam("username")
        String password = getRequestBodyparam("password")

        Int userID = database.FindUser(username,password)

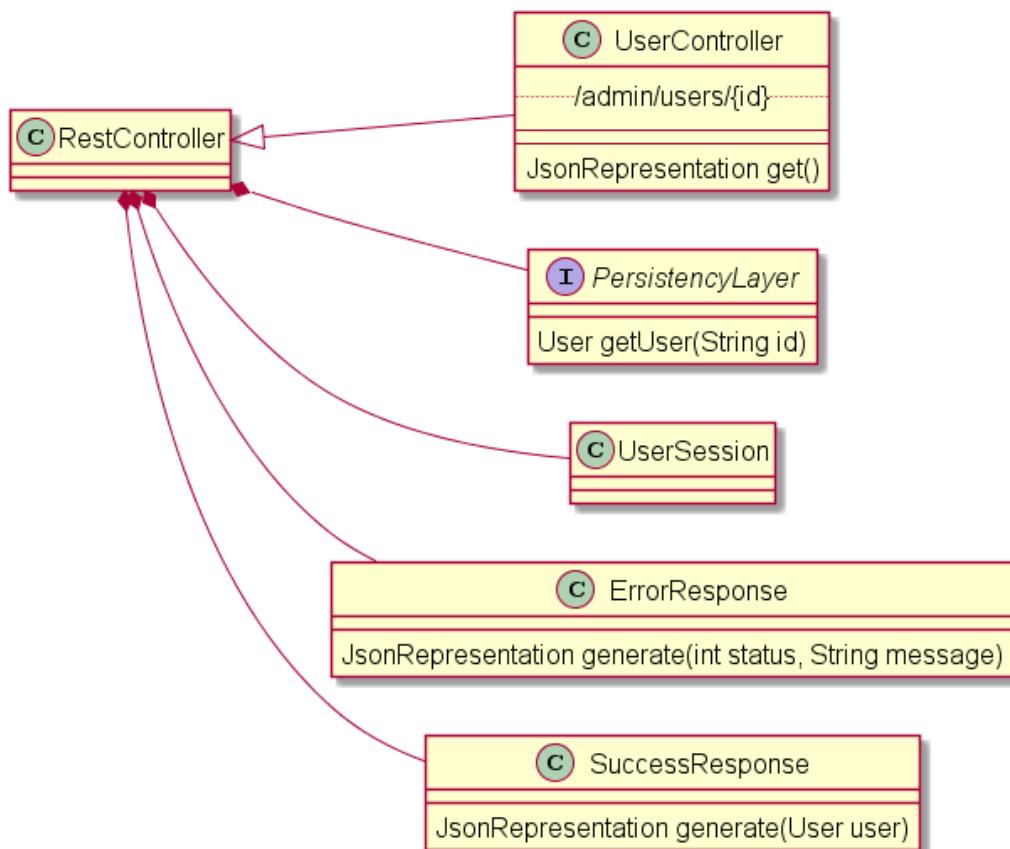
        If( userID == -1){

            Return error.generate(404,"No Use found")
        }
        //else, found User ➔ Ok

        session = session.newSession(userID)
        return success.generate("Login Successful")
    }
}
```

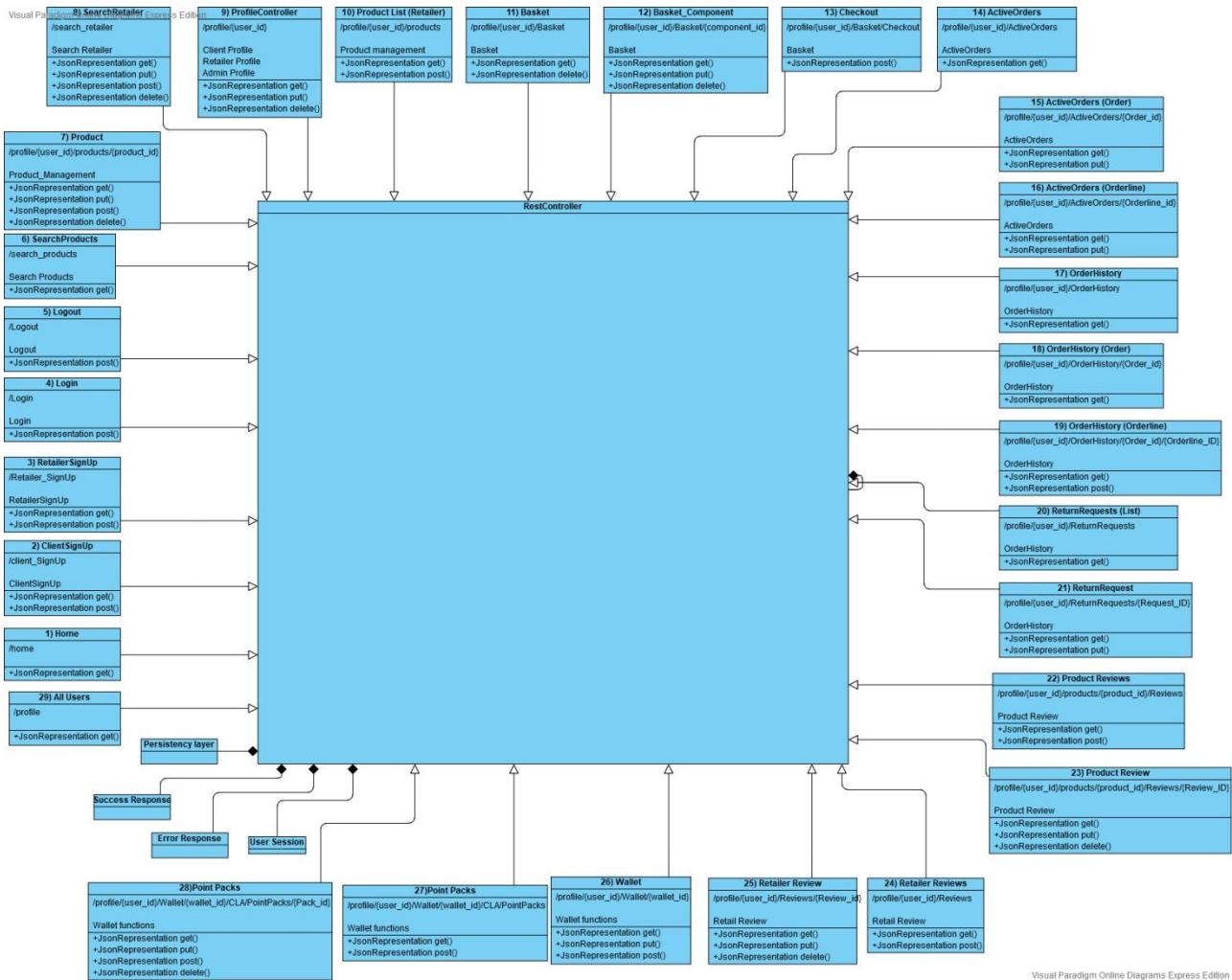
3. Ένα UML Class diagram για το σύνολο των υποστηριζόμενων Controllers (υποκλάσεις της RestController) της εφαρμογής σας, αντιστοιχίζοντας τους με τα σχετικά endpoints/routes. Οι κλάσεις – Controllers θα πρέπει να χρησιμοποιούν τις κλάσεις του επιπέδου της επιχειρησιακής σας λογικής, όπου ανά περίπτωση απαιτούνται (αν προκύψει περιορισμός χώρου, μην ενσωματώσετε τις λειτουργίες στη SuccessResponse, ErrorResponse και παρουσιάστε τις κλάσεις τη Επιχειρησιακής Λογικής ως χαρακτηριστικά των Controllers σας).

Για παράδειγμα, η συνολική αποτύπωση του επιπέδου Υποδοχής Αιτημάτων & Αποστολής Απαντήσεων ως διάγραμμα κλάσεων, για την περίπτωση που υπάρχει μόνο η κλάση UserController, είναι η ακόλουθη (εσείς θα έχετε σίγουρα περισσότερους controllers, πρόσθετες μεθόδους στην κλάση SuccessResponse και πρόσθετες κλάσεις επιπέδου επιχειρησιακής λογικής).



#####

To Class diagram είναι το παρακάτω: (όπως αναφέρει και η εκφώνηση το Success Response, Error Response, UserSession, Persistency Layer, τα βάλαμε απλοποιημένα λόγω έλλειψης χώρου)



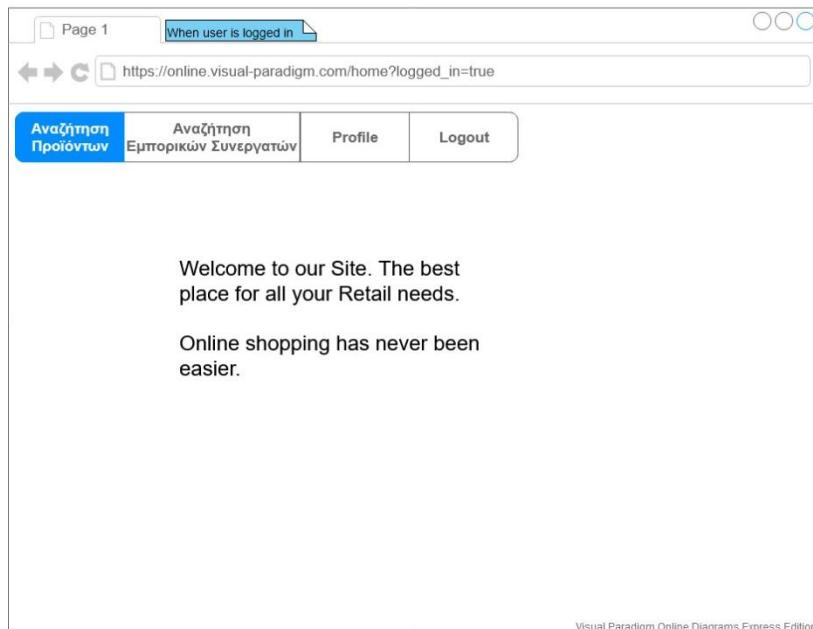
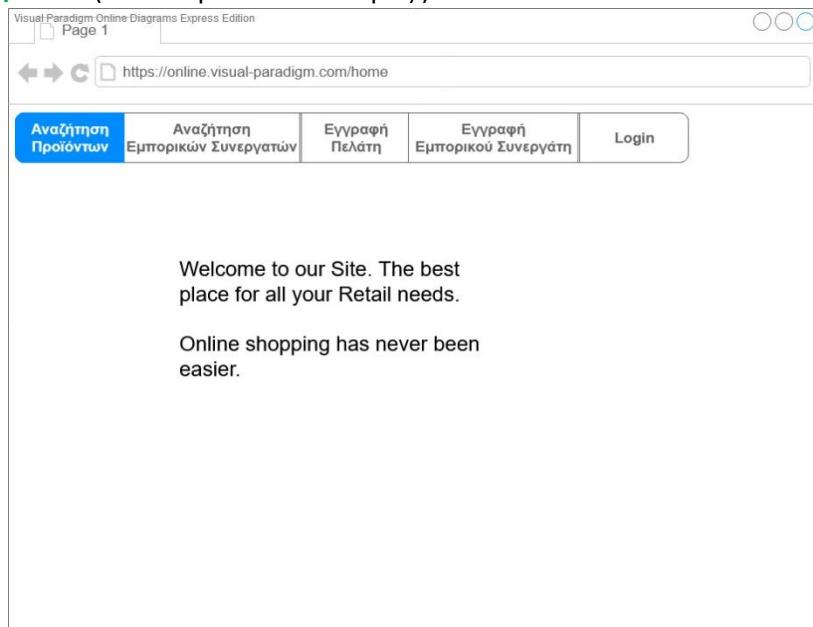
## 8 Διεπαφές χρήστης Frontend Εφαρμογής

Στην ενότητα αυτή θα πρέπει να ενσωματώσετε τα UI Mockups σας, ακολουθώντας, ιδανικά, την ίδια σειρά παράθεσης με τα σενάρια ή τις περιπτώσεις χρήσης. Θα πρέπει, επίσης, να αναφέρετε τα REST endpoints που χρησιμοποιεί η κάθε σελίδα.

#####

Τα rest api θα τα βάλω με πράσινο

## /home ( τα 2 παρακάτω είναι μαζί)



## /client\_signup

Visual Paradigm Online Diagrams Express Edition  
Page 1

https://online.visual-paradigm.com/client\_signup

Username	Password	Email
John	*****	email@mail.com

Name\_Surname  
John Travolta

**Submit**

Visual Paradigm Online Diagrams Express Edition

## /Retailer\_signup

Visual Paradigm Online Diagrams Express Edition  
Page 1

https://online.visual-paradigm.com/Retailer\_signup

Username	Password	Email
John	*****	email@mail.com

Όνομα Εταιρείας Διεύθυνση Εταιρείας  
John Doe Business 21 blabla street

Categories (You can choose more than one)

<input type="checkbox"/> Clothing	<input type="checkbox"/> Sporting/Hobby goods
<input type="checkbox"/> Electronics and appliances	<input type="checkbox"/> Health/Beauty goods
<input type="checkbox"/> Food and Beverages	<input type="checkbox"/> Miscellaneous
<input type="checkbox"/> Building and Garden Supplies	

Description

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

**Submit**

Visual Paradigm Online Diagrams Express Edition

## /Login

Visual Paradigm Online Diagrams Express Edition  
Page 1

Back Forward Stop https://online.visual-paradigm.com/Login

Username

Password

**Login**

Visual Paradigm Online Diagrams Express Edition

## /Log\_out

Visual Paradigm Online Diagrams Express Edition  
Page 1

Back Forward Stop https://online.visual-paradigm.com/log\_out

Log Out Successful

**OK**

Visual Paradigm Online Diagrams Express Edition

## /search\_products

Visual Paradigm Online Diagrams Express Edition  
Page 1

https://online.visual-paradigm.com/search\_products

Search Products	Categories	Results
<input type="text" value="Search"/> 	<input type="checkbox"/> Clothing <input type="checkbox"/> Electronics and appliances <input type="checkbox"/> Food and Beverages <input type="checkbox"/> Building and Garden Supplies <input type="checkbox"/> Sporting/Hobby goods <input type="checkbox"/> Health/Beauty goods <input type="checkbox"/> Miscellaneous	Product 1 Product 2 Product 3 Product 4
<b>Order By:</b> <input checked="" type="radio"/> Best Reviews <input checked="" type="radio"/> Most Sales		
<b>Products of:</b> <input checked="" type="radio"/> Retailer_1 <input checked="" type="radio"/> Retailer_2 <input checked="" type="radio"/> Retailer_3 <input checked="" type="radio"/> Retailer_4 <input checked="" type="radio"/> Retailer_5		
<b>Max Price:</b> 100 		

Visual Paradigm Online Diagrams Express Edition

## /search\_retailer

Visual Paradigm Online Diagrams Express Edition

Page 1

https://online.visual-paradigm.com/search\_retailer

Search Products	Categories	Results
Search <input type="text"/> 	<input type="checkbox"/> Clothing <input type="checkbox"/> Electronics and appliances <input type="checkbox"/> Food and Beverages <input type="checkbox"/> Building and Garden Supplies <input type="checkbox"/> Sporting/Hobby goods <input type="checkbox"/> Health/Beauty goods <input type="checkbox"/> Miscellaneous	Retailer 1 Retailer 2 Retailer 3 Retailer 4

Order By:

Best Reviews  
 Most Sales

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/products/{product\_id}

The image displays three separate browser windows, each showing a different step in a process:

- Screenshot 1 (Left): Product Page**

Product Page  
Product\_ID: 1123  
Retailer\_ID: 231  
Product Name: Jeans  
Price: 100 points  
Available Amount: 20  
Discount: 25%  
Category: Clothing

**Add to Basket**
- Screenshot 2 (Middle): Edit Product**

Product Page  
Product\_ID: 1123  
Retailer\_ID: 231  
Product Name: Jeans  
Price: 100 points  
Available Amount: 20  
Discount: 25%  
Category: Clothing

**Edit** **Delete** **Reviews**

**Add to Basket**

**Edit Product**  
Enter new info:  
Product\_Name:   
Price:   
Available\_Amount:   
Discount:   
Category:  Clothing  
 Electronics  
 Books  
 Food and Beverages  
 Building and Garden Supplies  
 Sporting/Hobby goods  
 Health/Beauty goods  
 Miscellaneous

**Cancel** **OK**
- Screenshot 3 (Right): Delete Confirmation**

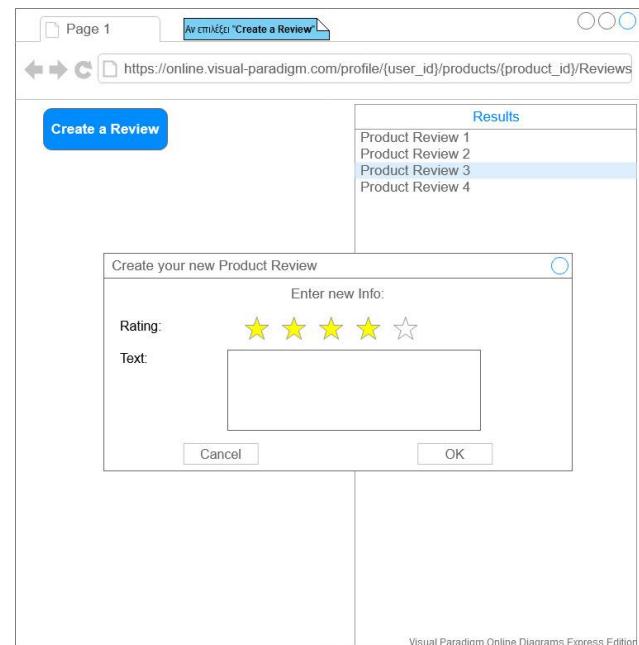
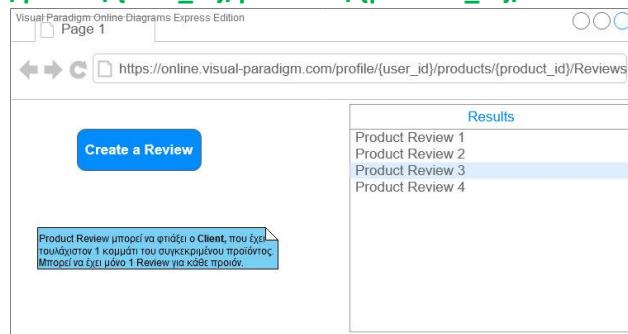
Product Page  
Product\_ID: 1123  
Retailer\_ID: 231  
Product Name: Jeans  
Price: 100 points  
Available Amount: 20  
Discount: 25%  
Category: Clothing

**Edit** **Delete** **Reviews**

**Add to Basket**

**Delete Product?**  
Are you sure you want to delete this product?  
**Cancel** **OK**

## /profile/{user\_id}/products/{product\_id}/Reviews



## /profile/{user\_id}/products/{product\_id}/Reviews/{Review\_ID}

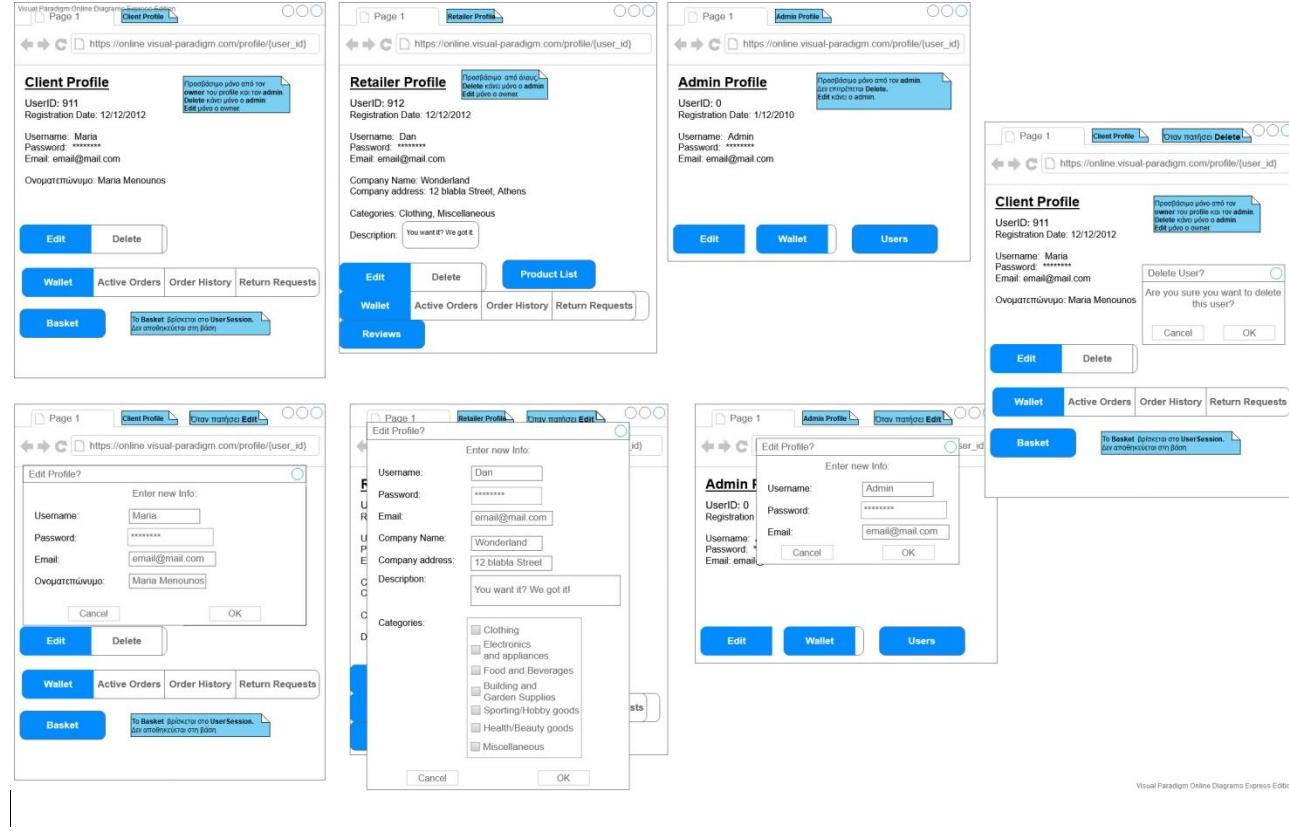
The image displays three sequential screenshots of a web application interface for managing product reviews.

**Screenshot 1: Product Review View**  
A product review is displayed with a rating of 4 stars and the text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat." Below the review are "Edit" and "Delete" buttons.

**Screenshot 2: Edit Product Review**  
A modal window titled "Edit Review?" is open, showing the same review details. It includes fields for "Rating" (set to 4) and "Text". A "Cancel" button is at the bottom left, and an "OK" button is at the bottom right.

**Screenshot 3: Delete Product Review Confirmation**  
A modal window titled "Delete Review?" asks "Are you sure you want to delete this Review?". It contains the same review text and "Cancel" and "OK" buttons.

**/profile/{user\_ID}**



## /profile/{user\_id}/products

Visual Paradigm Online Diagrams Express Edition

Page 1

https://online.visual-paradigm.com/profile/{user\_id}/products

Add New Product

Móvo o Retailer/owner μπορεί να κάνει Add Product

Προσθήκη στό όντος

Results
Product 1
Product 2
Product 3
Product 4

Page 1

https://online.visual-paradigm.com/profile/{user\_id}/products

Add New Product

Móvo o Retailer/owner μπορεί να κάνει Add Product

Προσθήκη στό όντος

Add new Product?

Enter product Info: (the page will refresh after this)

Product\_Name: Jeans

Price: 1123

Available Amount: 1123

Discount: 20%

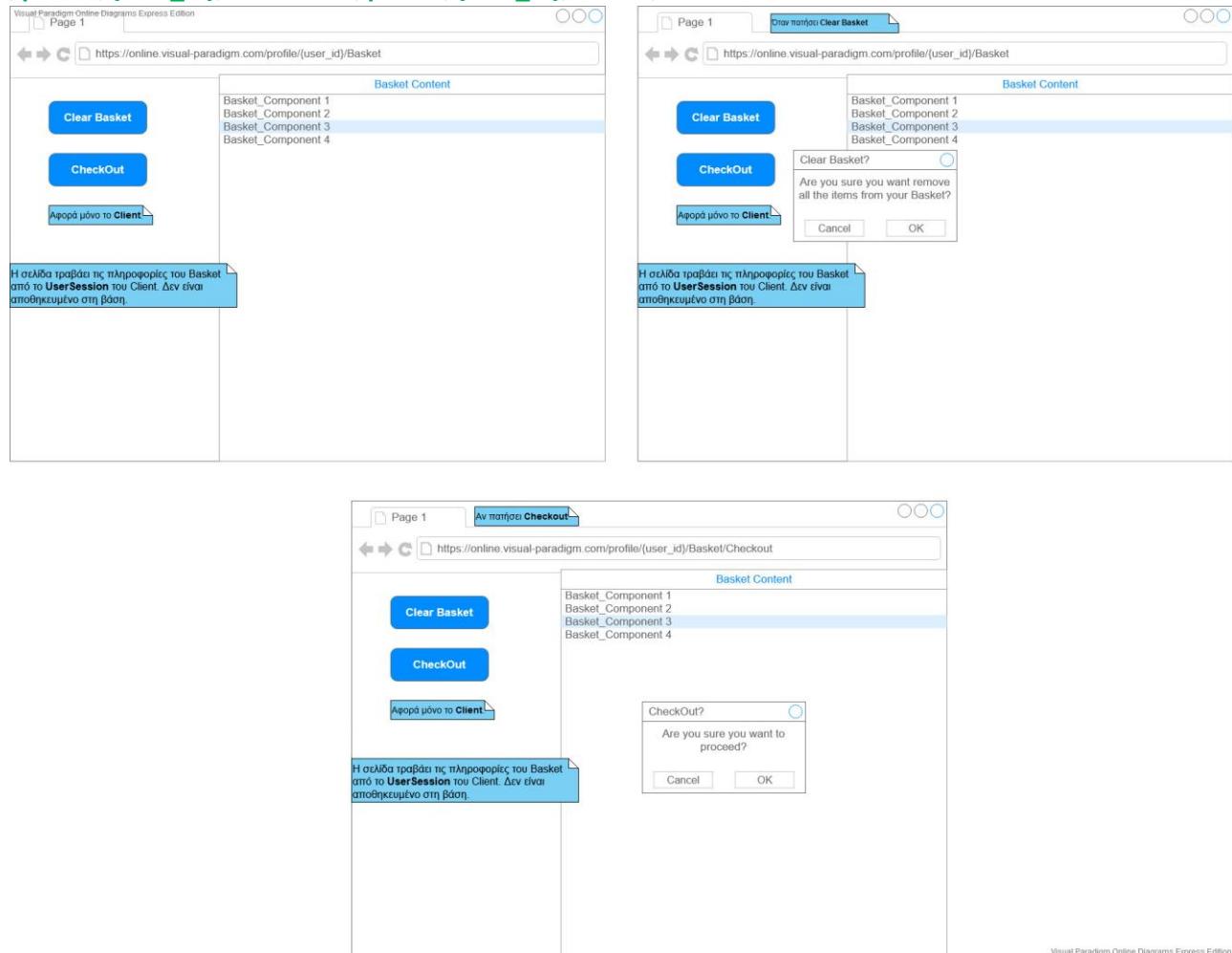
Category:

- Clothing
- Electronics and appliances
- Food and Beverages
- Building and Garden Supplies
- Sporting/Hobby goods
- Health/Beauty goods
- Miscellaneous

Cancel OK

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/Basket και /profile/{user\_id}/Basket/Checkout



Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/Basket/{component\_id}

Visual Paradigm Online Diagrams Express Edition

Page 1

https://online.visual-paradigm.com/profile/{user\_id}/Basket/{Basket\_Component\_ID}

Product_ID:1123 Quantity: 2  Basket_Component_ID: 2	<a href="#">Edit</a>   <a href="#">Delete</a>
--	---

Product\_ID:1123  
Quantity: 2  
Basket\_Component\_ID: 2

Edit Quantity?  
Enter new quantity:  
2  
[Cancel](#) | [OK](#)

Remove from Basket?  
Are you sure you want to remove this from your Basket?  
[Cancel](#) | [OK](#)

Page 1

https://online.visual-paradigm.com/profile/{user\_id}/Basket/{Basket\_Component\_ID}

Product_ID:1123 Quantity: 2  Basket_Component_ID: 2	<a href="#">Edit</a>   <a href="#">Delete</a>
--	---

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/ActiveOrders

Visual Paradigm Online Diagrams Express Edition  
Page 1 Προσβάσιμο από Retailers και Clients

https://online.visual-paradigm.com/profile/{user\_id}/ActiveOrders

Active Order	Status
Active Order 1	Pending
Active Order 2	Courier
Active Order 3	Courier
Active Order 4	Pending

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/ActiveOrders/{Order\_id}

Visual Paradigm Online Diagrams Express Edition  
Page 1

Order\_ID: 1123  
Date Ordered: 12/3/2019  
Date Pickup:  
Client\_ID: 221  
Retailer\_ID: 12  
State: Pending

Change State

To Change State μπορεί να το επιλέξει μόνο ο Retailer, ο οποίος έλαβε την παραγγελία.

Order Lines

- Order Line 1
- Order Line 2
- Order Line 3
- Order Line 4

Page 1

Όταν πατήσει Change State.

Order\_ID: 1123  
Date Ordered: 12/3/2019  
Date Pickup:  
Client\_ID: 221  
Retailer\_ID: 12  
State: Courier

Change State

Change Order State?

Please choose a new State

Pending  
 Courier  
 Completed

Cancel OK

Order Lines

[/profile/{user\\_id}/ActiveOrders/{Order\\_id}/{Orderline\\_id}](https://online.visual-paradigm.com/profile/{user_id}/ActiveOrders/{Order_id}/{Orderline_id})

Visual Paradigm Online Diagrams Express Edition

Page 1

Orderline\_ID: 1123  
Order\_ID: 234

Product\_ID: 1567  
Quantity: 4  
Return Quantity: 0  
Cancel Quantity: 0

To Cancel μπορεί να επιλέξει μόνο  
ο Client, ο οποίος έκανε την παραγγελία.

**Cancel**

Page 1

https://online.visual-paradigm.com/profile/{user\_id}/ActiveOrders/{Order\_id}/{Orderline\_id}

Orderline\_ID: 1123  
Order\_ID: 234

Product\_ID: 1567  
Quantity: 4  
Return Quantity: 0  
Cancel Quantity: 0

To Cancel μπορεί να επιλέξει μόνο  
ο Client, ο οποίος έκανε την παραγγελία.

**Cancel**

Cancel Quantity of Orderline?

Enter Cancel quantity:

**Cancel**      **OK**

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/OrderHistory

Visual Paradigm Online Diagrams Express Edition

Page 1 Προσβάσιμο από Retailers και Clients

https://online.visual-paradigm.com/profile/{user\_id}/OrderHistory

Order	Status
Completed Order 1	Completed
Completed Order 2	Completed
Completed Order 3	Completed
Completed Order 4	Completed

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/OrderHistory/{Order\_ID}

Visual Paradigm Online Diagrams Express Edition

Page 1

Order\_ID:1123

Date Ordered: 12/3/2019

Date Pickup: 15/3/2019

Client\_ID: 221

Retailer\_ID: 12

State: **Completed**

Order Lines

Order Line 1

Order Line 2

Order Line 3

Order Line 4

https://online.visual-paradigm.com/profile/{user\_id}/OrderHistory/{Order\_id}

Visual Paradigm Online Diagrams Express Edition

[/profile/{user\\_id}/OrderHistory/{Order\\_id}/{Orderline\\_id}](https://online.visual-paradigm.com/profile/{user_id}/OrderHistory/{Order_id}/{Orderline_id})

Visual Paradigm Online Diagrams Express Edition

Page 1

Orderline\_ID:1123  
Order\_ID: 234  
Product\_ID: 1567  
Quantity: 4  
Return Quantity: 0  
Cancel Quantity: 0

To Create Return Request μπορεί να επιλέξει μόνο ο Client, ο οποίος έκανε την παραγγελία και έχει παραλάβει τουλάχιστον 1 τεμάχιο του προϊόντος

**Create Return Request**

Page 1

Όταν πατήσει **Create Return Request**

https://online.visual-paradigm.com/profile/{user\_id}/OrderHistory/{Order\_id}/{Orderline\_id}

Orderline\_ID:1123  
Order\_ID: 234  
Product\_ID: 1567  
Quantity: 4  
Return Quantity: 0  
Cancel Quantity: 0

To Create Return Request μπορεί να επιλέξει μόνο ο Client, ο οποίος έκανε την παραγγελία και έχει παραλάβει τουλάχιστον 1 τεμάχιο του προϊόντος

**Create Return Request**

Create Request?

Please enter Return quantity:

2

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/ReturnRequests

Visual Paradigm Online Diagrams Express Edition  
Page 1 Προσβάσιμο από Retailers και Clients

https://online.visual-paradigm.com/profile/{user\_id}/ReturnRequests

Return Request	Status
Return Request 1	Pending
Return Request 2	Accepted
Return Request 3	Declined
Return Request 4	Pending

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/ReturnRequests/{Request\_ID}

Visual Paradigm Online Diagrams Express Edition

Page 1 https://online.visual-paradigm.com/profile/{user\_id}/ReturnRequests/{Request\_ID}

Request\_ID:1123  
Orderline\_ID: 2134  
Quantity: 2

ClientID: 1356  
RetailerID: 6823

State: Pending

**Accept/Decline Request** **Cancel Request**

To Accept/Deny Request μπορεί να επιλέξει μόνο ο Retailer, ο οποίος έχει στέλνει το προϊόν στο Client  
To Cancel Request μόνο ο Client που έκανε την αίτηση

Page 1 **Ανατίναξε Cancel Request** https://online.visual-paradigm.com/profile/{user\_id}/ReturnRequests/{Request\_ID}

Request\_ID:1123  
Orderline\_ID: 2134  
Quantity: 2

ClientID: 1356  
RetailerID: 6823

State: Pending

**Accept/Decline Request** **Cancel Request**

To Accept/Deny Request μπορεί να επιλέξει μόνο ο Retailer, ο οποίος έχει στέλνει το προϊόν στο Client  
To Cancel Request μόνο ο Client που έκανε την αίτηση

Page 1 **Ανατίναξε Accept Deny Request** https://online.visual-paradigm.com/profile/{user\_id}/ReturnRequests/{Request\_ID}

Request\_ID:1123  
Orderline\_ID: 2134  
Quantity: 2

ClientID: 1356  
RetailerID: 6823

State: Pending

**Accept/Decline Request** **Cancel Request**

To Accept/Deny Request μπορεί να επιλέξει μόνο ο Retailer, ο οποίος έχει στέλνει το προϊόν στο Client  
To Cancel Request μόνο ο Client που έκανε την αίτηση

Accept or Deny the Request?  Please make your decision:  
 Accept  
 Decline  
**Cancel** **OK**

To Accept/Deny Request μπορεί να επιλέξει μόνο ο Retailer, ο οποίος έχει στέλνει το προϊόν στο Client  
To Cancel Request μόνο ο Client που έκανε την αίτηση

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/Reviews

Visual Paradigm Online Diagrams Express Edition  
Page 1

https://online.visual-paradigm.com/profile/{user\_id}/Reviews

Create a Review

Retailer Review μπορεί να φτιάξει ο Client, που έχει κάνει τουλάχιστον 1 review σε κάποιο προϊόν του Retailer

Results

- Retailer Review 1
- Retailer Review 2
- Retailer Review 3
- Retailer Review 4

Page 1

Av επιλέξει "Create a Review"

https://online.visual-paradigm.com/profile/{user\_id}/Reviews

Create your new Retailer Review

Enter new Info:

Rating: ★ ★ ★ ★ ☆

Text:

Retailer Review κάνει τουλάχιστον 1 review σε κάποιο προϊόν του Retailer

Cancel OK

Visual Paradigm Online Diagrams Express Edition

## /profile/{user\_id}/Reviews/{Review\_ID}

Visual Paradigm Online Diagrams Express Edition  
Page 1

https://online.visual-paradigm.com/profile/{user\_id}/Reviews/{Review\_ID}

**Retailer Review**

Retailer Review ID: 1562  
Retailer ID: 4422  
Client ID: 1245

★★★★★

**Text:**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

[Edit](#) [Delete](#)

Page 1

https://online.visual-paradigm.com/profile/{user\_id}/Reviews/{Review\_ID}

**Retailer Review**

Retailer Review ID: 1562  
Retailer ID: 4422  
Client ID: 1245

★★★★★

**Rating:**

**Text:**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

[Edit](#) [Delete](#)

Page 1

https://online.visual-paradigm.com/profile/{user\_id}/Reviews/{Review\_ID}

**Retailer Review**

Retailer Review ID: 1562  
Retailer ID: 4422  
Client ID: 1245

★★★★★

**Delete Review?**

Are you sure you want to delete this Review?

[Cancel](#) [OK](#)

## /profile/{user\_id}/Wallet/{Wallet\_ID}

Visual Paradigm Online Diagrams Express Edition  
Page 1

https://online.visual-paradigm.com/Wallet/{Wallet\_ID}

Wallet\_ID:1123  
User\_ID: 2134  
Points: 150

**Money > Points or Points-> Money**  
Access Point Packs

**Customer Loyalty 1**  
Access Point Packs

**Customer Loyalty 2**  
Check for account anniversary or milestones in successful orders

Η δυνατότητα points->Money αφορά μόνο τον Admin και τους Retailers

Page 1 Αν επιλέξει το Money->Points or Points-> Money

https://online.visual-paradigm.com/Wallet/{Wallet\_ID}

Wallet\_ID:1123  
User\_ID: 2134  
Points: 150

**Points or Money?**  
Choose what you want to do and fill the necessary form:

**Choice**  
 Money -> Points  
 Points -> Money

**Number of Points**  
120

**IBAN**  
AZX1111

Cancel OK

**Money > Points or Points-> Money**  
Access Point Packs

**Customer Loyalty 1**  
Access Point Packs

**Customer Loyalty 2**  
Check for account anniversary or milestones in successful orders

Η δυνατότητα points->Money αφορά μόνο τον Admin και τους Retailers

Page 1 Αν επιλέξει το Customer Loyalty 2

https://online.visual-paradigm.com/Wallet/{Wallet\_ID}

Wallet\_ID:1123  
User\_ID: 2134  
Points: 150

**Customer Loyalty 2**  
Do you want to check for anniversary or milestone Bonus Points?  
Cancel OK

**Money -> Points or Points-> Money**  
Access Point Packs

**Customer Loyalty 1**  
Access Point Packs

**Customer Loyalty 2**  
Check for account anniversary or milestones in successful orders

Η δυνατότητα points->Money αφορά μόνο τον Admin και τους Retailers

Visual Paradigm Online Diagrams Express Edition



https://online.visual-paradigm.com/Wallet/{Wallet\_ID}/PointPacks

To Add Pack γίνεται  
μόνο από τον Admin

Add Pack

### Point Packs

- Point Pack 1
- Point Pack 2
- Point Pack 3
- Point Pack 4

Page 1

Όταν πατάει το Add Pack



https://online.visual-paradigm.com/Wallet/{Wallet\_ID}/PointPacks

To Add Pack γίνεται  
μόνο από τον Admin

Add Pack?

Please fill the form below:

Number of Points

Price(€)

Cancel

OK

- Point Pack 1
- Point Pack 2
- Point Pack 3
- Point Pack 4

### Point Packs

## /profile/{user\_id}/Wallet/{Wallet\_ID}/CLA/PointPacks/{Pack\_id}

The screenshots illustrate the PointPacks management interface:

- Screenshot 1 (Edit):** Shows a point pack with ID 1123, Price: 39 €, and Points: 120. Buttons for Edit, Delete, and Buy are present.
- Screenshot 2 (Edit):** A modal window titled "Edit Pack?" asks for "Number of Points" (120) and "Price(€)" (39). Buttons for Cancel and OK are shown.
- Screenshot 3 (Delete):** Shows the same point pack details. A modal window titled "Delete Pack?" asks if you want to delete the pack, with "Cancel" and "OK" buttons.
- Screenshot 4 (Buy):** Shows the same point pack details. A modal window titled "Buy Pack?" asks if you would like to buy this Point pack? and for "IBAN" (AZX1111). Buttons for Cancel and OK are shown.

## /profile

The screenshot shows the User Profiles page:

- Header:** Visual Paradigm Online Diagrams Express Edition, Page 1.
- URL:** https://online.visual-paradigm.com/profile
- Message Bar:** Αυτή η σελίδα/δυνατότητα είναι προσβάσιμη μόνο από τον Admin.
- User Profiles List:** A table titled "User Profiles" lists four users: User 1, User 2, User 3, and User 4. User 3 is highlighted with a blue background.
- Page Footer:** Visual Paradigm Online Diagrams Express Edition.

## 9 Πίνακας Βαρύτητας

Ενότητα εγγράφου	Ομάδες 2 ατόμων	Ομάδες 3 ατόμων
1. Εισαγωγή	5%	5%
2. Λειτουργικές απαιτήσεις και σενάρια χρήσης	20%	18%
3. Δομημένη μοντελοποίηση διαδικασιών	10%	10%
4. Αντικειμενοστρεφής μοντελοποίηση δεδομένων	10%	10%
5. Αρχιτεκτονική του συστήματος	-	-
6. Ποιοτικά χαρακτηριστικά και Μη-λειτουργικές απαιτήσεις	10%	10%
7. Προδιαγραφές της Back-end εφαρμογής	20%	18%
8. Διεπαφές χρήσης της Front-end εφαρμογής	15%	15%
9. Σενάρια και έλεγχοι αποδοχής	-	4%
<b>Σύνολο*</b>	<b>90%</b>	<b>90%</b>

\* Η γενική εικόνα του εγγράφου (σαφήνεια, πληρότητα, συντακτική αρτιότητα) βαραίνει κατά 10% τον τελικό βαθμό.