



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2019



E-market Ψώνια...με το πάτημα ενός κουμπιού!

Απαιτήσεις Λογισμικού (Προδιαγραφές Λογισμικού)

Del.1.2

Version 0.1.4

ΘΕΟΔΩΡΑΚΑΚΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ tkostas13@gmail.com

ΚΑΓΙΑΦΑΣ ΝΙΚΟΛΑΟΣ nikoskagiafas23@gmail.com

ΚΑΡΑΜΟΥΖΑΣ ΔΙΟΝΥΣΙΟΣ dion.kara.auth@gmail.com

ΚΥΠΑΡΙΣΣΗΣ ΟΔΥΣΣΕΑΣ odyskypa@ece.auth.gr

7/5/2019

**Ιστορικό Αλλαγών**

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
e-market	12/4/2019	Ευρεση κλάσεων/δημιουργία εγγράφου	0.1
e-market	15/4/2019	Εύρεση κλάσεων	0.2
e-market	16/4/2019	Χωρισμός κλάσεων σε πακέτα	0.3
e-market	17/4/2019	Χωρισμός κλάσεων σε πακέτα	0.4
e-market	19/4-5/5	Υλοποίηση των πακέτων	0.5
e-market	6/5/2019	Εύρεση προτύπων	0.6
e-market	7/5/2019	Έλεγχος	0.7

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Θεοδωρακάκος Κων/νος	19	tkostas13@gmail.com
Καγιαφας Νικόλαος	19	nikoskagiafas23@gmail.com
Καραμούζας Διονύσιος	19	dion.kara.auth@gmail.com
Κυπαρίσσης Οδυσσέας	19	odyskypa@ece.auth.gr



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	2
Λίστα Σχημάτων	2
1 Εισαγωγικά	3
1.1 Στόχος του Εγγράφου	3
1.2 Τυπογραφικές Παραδοχές του Εγγράφου	3
1.3 Αναγνωστικό κοινό και τρόπος ανάγνωσης	3
1.4 Σκοπός του Έργου	4
2 Στατική Μοντελοποίηση	5
2.1 <Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας>	5
2.1.1 Πακέτο UserHandle	5
2.1.2 Διάγραμμα κλάσεων	15
2.1.3 Πακέτο ProductHandle	16
2.1.4 Διάγραμμα κλάσεων	25
2.2 <Πακέτα λεξιλογίου σεναρίων μέσης προτεραιότητας>	26
2.2.1 Πακέτο CartHandle	26
2.2.2 Διάγραμμα κλάσεων	35
2.2.3 Πακέτο OrderHandle	36
2.2.4 Διάγραμμα κλάσεων	42
3 Μη λειτουργικές απαιτήσεις	43
3.1 Απαιτήσεις επίδοσης	43
3.2 Απαιτήσεις Χρηστικότητα (Usability)	43
3.3 Απαιτήσεις Αξιοπιστίας (Reliability)	43
3.4 Τεχνικές Απαιτήσεις Περιβάλλοντος	44
3.5 Απαιτήσεις Πολιτικής (Policy - legal , corporate)	45
4 Πρότυπα Σχεδιασμού που υιοθετήθηκαν	45
4.1 Δομικά πρότυπα	45
Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας	46
Παράρτημα ΙΙ – Ανοιχτά Θέματα	47

Λίστα Σχημάτων

Σχήμα 1. Διάγραμμα κλάσεων πακέτου UserHandle.	15
Σχήμα 2. Διάγραμμα κλάσεων πακέτου ProductHandle.	25
Σχήμα 3. Διάγραμμα κλάσεων πακέτου CartHandle.	35
Σχήμα 4. Διάγραμμα κλάσεων πακέτου OrderHandle.	42
Σχήμα 5. Διάγραμμα πρότυπου σχεδίασης bridge pattern για το SuppliesServiceProxy.	45
Σχήμα 6. Διάγραμμα πρότυπου σχεδίασης bridge pattern για το PaymentProxy.	46
Σχήμα 7. Διάγραμμα πρότυπου σχεδίασης bridge pattern.	46



1 Εισαγωγικά

1.1 Στόχος του Εγγράφου

Το παρόν έγγραφο αποτελεί έναν οδηγό κατανόησης των προδιαγραφών του λογισμικού που συνθέτει το σύστημα. Είναι ένα εργαλείο για την αναγνώριση των χαρακτηριστικών του συστήματος και την πλήρη περιγραφή τους, λαμβάνοντας υπόψη και θέτοντας ως απαραίτητη προϋπόθεση την ικανοποίηση των απαιτήσεων χρηστών που περιγράφηκαν στο προηγούμενο έγγραφο. Στόχος του εγγράφου είναι ο ακριβής προσδιορισμός της λειτουργίας του συστήματος. Περιγράφεται η συμπεριφορά του συστήματος σε σχέση με τους χρήστες του και τα συνεργαζόμενα συστήματα. Ένας ακόμη σημαντικός στόχος του εγγράφου είναι η αναλυτική περιγραφή των δεδομένων που διαχειρίζεται το σύστημα. Καθορίζονται οι τύποι δεδομένων που ανταλλάσσονται μεταξύ του συστήματος, των χρηστών και των συνεργαζόμενων συστημάτων. Επιπλέον, παρέχονται πλήρεις περιγραφές των σελίδων, των φορμών και των μηνυμάτων που επιστρέφει το σύστημα, μέσω των οποίων ανταλλάσσονται δεδομένα. Προς εξυπηρέτηση των στόχων αυτών είναι απαραίτητη η μοντελοποίηση του συστήματος, η οποία περιλαμβάνει την στατική μοντελοποίηση. Η στατική μοντελοποίηση αφορά την μοντελοποίηση των απαιτήσεων του χρήστη. Για ευκολότερη κατανόηση και ανάπτυξη του συστήματος, αυτό χωρίζεται σε πακέτα κλάσεων σύμφωνα με τις επιμέρους λειτουργικότητες του συστήματος. Γίνεται ιεράρχηση των πακέτων αυτών σύμφωνα με την σπουδαιότητα της λειτουργίας που περιγράφουν για την εφαρμογή, σε πακέτα υψηλής και μέσης προτεραιότητας. Κάθε πακέτο περιλαμβάνει ένα σύνολο κλάσεων. Οι κλάσεις κατηγοριοποιούνται σε κλάσεις οντοτήτων (entities), οι οποίες αναπαριστούν την μόνιμη πληροφορία του συστήματος, σε οριακές κλάσεις (boundaries) ή κλάσεις διεπαφής χρηστών, οι οποίες αναπαριστούν την αλληλεπίδραση ανάμεσα στον χρήστη και στο σύστημα και αναλαμβάνουν την επικοινωνία του συστήματος με εξωτερικά συστήματα και τέλος σε κλάσεις ελέγχου (controllers) ή ρυθμιστικές κλάσεις, οι οποίες είναι υπεύθυνες για τις εργασίες ελέγχου που εκτελούνται από το σύστημα και τον συντονισμό των διεργασιών των επιμέρους κλάσεων. Επιπλέον, δημιουργούνται διαγράμματα που περιγράφουν τις αλληλεπιδράσεις των κλάσεων και των πακέτων. Το παρόν έγγραφο θα αποτελέσει τον οδηγό για την ανάπτυξη του λογισμικού από την ομάδα ανάπτυξης.

1.2 Τυπογραφικές Παραδοχές του Εγγράφου

Το κείμενο του εγγράφου τούτου είναι γραμμένο με γραμματοσειρά Calibri, μεγέθους 10pt. Χρησιμοποιείται έντονη γραμματοσειρά Calibri, μεγέθους 12pt, για να δειχθεί η επικεφαλίδα ή το χαρακτηριστικό το οποίο αναλύεται από ένα κείμενο.

1.3 Αναγνωστικό κοινό και τρόπος ανάγνωσης

Το έγγραφο απευθύνεται στις παρακάτω ομάδες ανθρώπων, οι οποίες καλούνται να το μελετήσουν προκειμένου να μοντελοποιήσουν και να σχεδιάσουν τα χαρακτηριστικά του συστήματος, να προγραμματίσουν και να υλοποιήσουν την εφαρμογή.

- Ομάδα ανάπτυξης: Κάθε μέλος της ομάδας ανάπτυξης για να είναι σε θέση να συμβάλει στο σχεδιασμό, την ανάλυση και τον προγραμματισμό του συστήματος πρέπει να έχει κατανοήσει πλήρως τις απαιτήσεις λογισμικού της εφαρμογής. Ανάλογα με την δραστηριότητα ανάπτυξης τα μέλη μπορούν να έχουν διαφορετικούς ρόλους. Επομένως η ομάδα ανάπτυξης μπορεί να χρησιμοποιεί διαφορετικούς τομείς του παρόντος εγγράφου είτε να χρησιμοποιεί διαφορετικές όψεις των ίδιων διαγραμμάτων. Οι ρόλοι σε μία ομάδα ανάπτυξης είναι οι εξής:
- Αναλυτής: Ενδιαφέρεται για τις συσχετίσεις ανάμεσα στις κλάσεις στο πεδίο εφαρμογής και
- για τις βασικές λειτουργίες των κλάσεων, χωρίς να ενδιαφέρεται για κλάσεις που επιλύουν συγκεκριμένα προβλήματα. Επίσης, εκμεταλλεύεται την κληρονομικότητα για να ιεραρχήσει τις



κλάσεις. Προς επίτευξη των στόχων αυτών, απαραίτητη προϋπόθεση κρίνεται η μελέτη του δεύτερου και τρίτου κεφαλαίου του παρόντος εγγράφου, δίνοντας μεγαλύτερη έμφαση στα διαγράμματα κλάσεων και την συσχέτιση των πακέτων.

- **Σχεδιαστής συστήματος:** Ο Σχεδιαστής συστήματος επικεντρώνεται στη λύση του προβλήματος. Η σχεδίαση αποτελείται από πολλές εργασίες, κύρια εργασία είναι ο καθορισμός των διεπαφών. Ο σχεδιαστής περιγράφει τις διεπαφές των κλάσεων και τις διεπαφές των υποσυστημάτων. Προς επίτευξη των στόχων αυτών, απαραίτητη προϋπόθεση κρίνεται η μελέτη του δεύτερου και τρίτου κεφαλαίου του παρόντος εγγράφου, δίνοντας μεγαλύτερη έμφαση στα πακέτα κλάσεων, τη συσχέτιση των πακέτων και στις οριακές κλάσεις (boundaries) ή κλάσεις διεπαφής χρηστών.
- **Σχεδιαστής αντικειμένων:** Ο Σχεδιαστής αντικειμένων επικεντρώνεται στη δημιουργία των μόνιμων οντοτήτων του συστήματος. Απαραίτητη κρίνεται η μελέτη του δεύτερου κεφαλαίου του παρόντος εγγράφου δίνοντας μεγαλύτερη έμφαση στις κλάσεις οντοτήτων (entities).
- **Προγραμματιστής:** Για τα μέλη της ομάδας αυτής κρίνεται απαραίτητη η μελέτη του δεύτερου και τρίτου κεφαλαίου του παρόντος εγγράφου.
- **Ομάδα συντήρησης:** Η ομάδα συντήρησης και θα αναβάθμισης του συστήματος για να είναι σε θέση να παρέχει βελτιώσεις σε αυτό θα πρέπει να έχει κατανοήσει πρώτα πλήρως την λειτουργικότητα του και τις βασικές οντότητες που δημιουργήθηκαν για τον προγραμματισμό του. Συνεπώς, κρίνεται πολύ σημαντική η μελέτη του δεύτερου και τρίτου κεφαλαίου του παρόντος εγγράφου.

Η δομή του εγγράφου ακολουθεί συγκεκριμένα πρότυπα, προκειμένου το ίδιο να είναι ευανάγνωστο για κάθε ενδιαφερόμενο, αλλά και να πληροί τις προϋποθέσεις που επιβάλλει το τεχνικά εξοικειωμένο κοινό. Το έγγραφο έχει τις εξής ενότητες:

- **Κεφάλαιο 1^ο :** Αντιπροσωπεύει το εισαγωγικό τμήμα του εγγράφου και περιέχει μια επισκόπηση του στόχου του, τις τυπογραφικές παραδοχές, την ανάλυση του αναγνωστικού κοινού, καθώς και του τρόπου ανάγνωσης του εγγράφου. Επιπρόσθετα, περιέχει ανάλυση του σκοπού του έργου της εφαρμογής.
- **Κεφάλαιο 2^ο :** Πραγματοποιείται η καταγραφή των πακέτων των σεναρίων υψηλής και μέσης προτεραιότητας. Για κάθε πακέτο, γίνεται λεπτομερής ανάλυση των κλάσεων και των συναρτήσεων που περιέχουν και παρουσιάζεται το διάγραμμα συσχέτισης των κλάσεων και των πακέτων.
- **Κεφάλαιο 3^ο :** Αναγράφονται οι μη λειτουργικές απαιτήσεις του συστήματος και κατηγοριοποιούνται.
- **Κεφάλαιο 4^ο :** Πραγματοποιείται η καταγραφή των προτύπων που χρησιμοποιήθηκαν στο σύστημά μας, προκειμένου να επιλυθούν ορισμένα προβλήματα και να ικανοποιηθούν ορισμένες μη λειτουργικές απαιτήσεις.
- **Παράρτημα I :** Περιέχει τον πίνακα ιχνηλασιμότητας, ο οποίος περιέχει προσθήκες στοιχείων του 2ου παραδοτέου που είχαν παραλειφθεί στο 1ο.
- **Παράρτημα II :** Περιέχει μια λίστα με ανοιχτά θέματα που αφορούν το λογισμικό.

1.4 Σκοπός του Έργου

Η start-up που αναπτύσσεται έχει ως σκοπό την δημιουργία ενός ηλεκτρονικού πολυκαταστήματος (e-



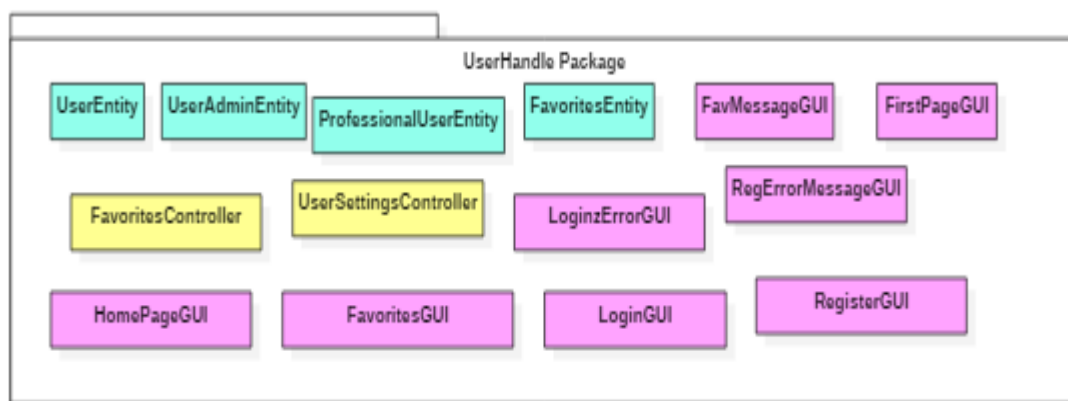
market). Σε συνεργασία με άλλα πολυκαταστήματα, προμηθευτές και μία υπηρεσία delivery η εφαρμογή δίνει την δυνατότητα στον χρήστη να επιλέξει ποια προϊόντα του καταστήματος επιθυμεί να αγοράσει και να δημιουργήσει μια τελική παραγγελία. Τού δίνεται επίσης η δυνατότητα να διαλέξει ανάμεσα σε πολλαπλές μεθόδους πληρωμής αυτήν που τον διευκολύνει περισσότερο αλλά και να εισάγει τα στοιχεία που αφορούν τον χώρο και τον χρόνο παράδοσης της παραγγελίας του. Το σύστημα τον ενημερώνει επιπλέον για τις τρέχουσες προσφορές στα διάφορα διαφημιζόμενα προϊόντα αλλά και προβάλλει κριτικές και αξιολογήσεις προηγούμενων χρηστών της εφαρμογής για την ποιότητα της εξυπηρέτησης που δέχτηκαν. Η εφαρμογή μπορεί ακόμα να δεχτεί επαγγελματίες εγγεγραμμένους χρήστες, οι οποίοι αγοράζουν πιο μαζικά και τακτικά από αυτήν και να τους παρέχει την κατάλληλη μεταχείριση, όπως παράδοση των προϊόντων ταχύτατα και με μεγαλύτερες και πιο συμφέρουσες προσφορές. Ο πελάτης συνεπώς με την χρήση της εφαρμογής θα έχει με το πάτημα μόνο λίγων κουμπιών την παραγγελία του στον χώρο του, όποτε εκείνος το επιθυμήσει.

2 Στατική Μοντελοποίηση

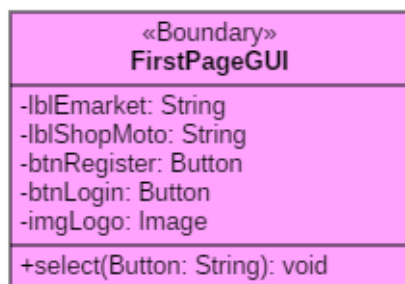
2.1 <Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας>

2.1.1 Πακέτο UserHandle

Είναι το πακέτο που αναφέρεται στην είσοδο-έξοδο του χρήστη από το σύστημα, την αρχική σελίδα, την σελίδα Home καθώς και την σελίδα αγαπημένα, αφού είναι ξεχωριστή για κάθε χρήστη.



Boundary FirstPageGUI



Η κλάση αυτή είναι η αρχική σελίδα της εφαρμογής όπου ο χρήστης διαλέγει αν θα κάνει «login» η «register».

Χαρακτηριστικά:

- lblEmarket: Το όνομα της εφαρμογής.
- lblShopMoto: Το «moto» της εφαρμογής.
- btnRegister : Ένα κουμπί που μεταφέρει τον χρήστη στη σελίδα «register».
- btnLogin: Ένα κουμπί που μεταφέρει τον χρήστη στη σελίδα «login».
- imgLogo: Το «logo» της εφαρμογής.



Μέθοδοι:

- select(Button: String): Καλεί τον controller UserSettingsController που ελέγχει ποιο κουμπί πατήθηκε και εμφανίζει την κατάλληλη οθόνη.

Boundary RegisterGUI

Η κλάση αυτή εκφράζει την σελίδα, όπου ένας καινούργιος χρήστης συμπληρώνει τα στοιχεία του, για να καταχωρηθεί σαν χρήστης του συστήματος.

«Boundary» RegisterGUI
-lblRegister: String -lblName: String -lblSurname: String -lblUsername: String -lblAddress: String -lblFloor: String -lblPostCode: String -lblPhone: String -lblEmail: String -lblPassword: String -lblQuestion: String -btnRegister: Button +txtName: textbox +txtSurname: textbox +txtUsername: textbox +txtAddress: textbox +txtFloor: textbox +txtPostCode: textbox +txtPhone: textbox +txtEmail: textbox +txtPassword: textbox +tbxQuestion: tickbox -btnReturnFirstPage: Button
+select(Button: String): void +tickboxClick(): void

Χαρακτηριστικά:

- lblRegister: Ετικέτα με το μήνυμα "Register".
- lblName: Ετικέτα με το μήνυμα "Όνομα".
- lblSurname: Ετικέτα με το μήνυμα "Επώνυμο".
- lblUsername: Ετικέτα με το μήνυμα "Όνομα Χρήστη".
- lblAddress: Ετικέτα με το μήνυμα "Διεύθυνση".
- lblFloor: Ετικέτα με το μήνυμα "Ορόφος".
- lblPostCode: Ετικέτα με το μήνυμα "Τ.Κ.".
- lblPhone: Ετικέτα με το μήνυμα "Τηλέφωνο".
- lblEmail: Ετικέτα με το μήνυμα "email".
- lblPassword: Ετικέτα με το μήνυμα "Password".
- lblQuestion: Ετικέτα με το μήνυμα "Είσαι επαγγελματίας χρήστης;".
- btnRegister: Ένα κουμπί που καταχωρεί τον χρήστη και τον μεταφέρει στην σελίδα «Home».
- txtName: Πεδίο συμπλήρωσης του ονόματος.
- txtSurname: Πεδίο συμπλήρωσης του επώνυμου.
- txtUsername: Πεδίο συμπλήρωσης του ονόματος χρήστη.
- txtAddress: Πεδίο συμπλήρωσης της διεύθυνσης.
- txtFloor: Πεδίο συμπλήρωσης του ορόφου.
- txtPostCode: Πεδίο συμπλήρωσης του ταχυδρομικού κώδικα.
- txtPhone: Πεδίο συμπλήρωσης του τηλεφώνου.
- txtEmail: Πεδίο συμπλήρωσης του "email".
- txtPassword: Πεδίο συμπλήρωσης του "Password".
- tbxQuestion: Κουμπί επιβεβαίωσης σε περίπτωση που είναι επαγγελματίας χρήστης.

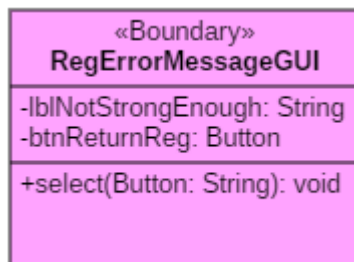


- btnReturnFirstPage: Κουμπί που επιστρέφει στην αρχική σελίδα.

Μέθοδοι:

- select(Button: String): Καλεί τον controller UserSettingsController που ελέγχει ποιο κουμπί πατήθηκε και εμφανίζει την κατάλληλη οθόνη.
- tickboxClick(): Καταχωρεί το πάτημα του κουμπιού επιβεβαίωσης.

Boundary RegErrorMessageGUI



Η κλάση αυτή αντιπροσωπεύει το παράθυρο που εμφανίζεται όταν κάποιος βάλει μικρό "Password" και προσπαθήσει να κάνει "register".

Χαρακτηριστικά:

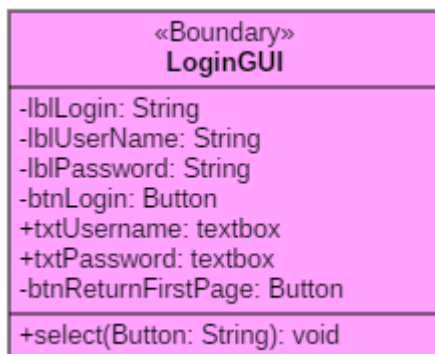
- lblNotStrongEnough : Ετικέτα με το μήνυμα "Το password δεν είναι αρκετά ισχυρό".
- btnReturnReg: Κουμπί που επιστρέφει στην σελίδα "register".

Μέθοδοι:

- select(Button: String): Καλεί τον controller UserSettingsController που ελέγχει ποιο κουμπί πατήθηκε και εμφανίζει την κατάλληλη οθόνη.

Boundary LoginGUI

Η συνάρτηση αυτή αντιπροσωπεύει την σελίδα όπου ένας εγγεγραμμένος χρήστης κάνει «login».



Χαρακτηριστικά:

- lblLogin: Ετικέτα με το μήνυμα "Log in".
- lblUsername: Ετικέτα με το μήνυμα "Όνομα Χρήστη".
- lblPassword: Ετικέτα με το μήνυμα "Password".
- btnLogin: Κουμπί που εισάγει τον χρήστη στο σύστημα και τον στέλνει στη σελίδα «Home».
- txtUsername: Πεδίο συμπλήρωσης του ονόματος χρήστη.
- txtPassword: Πεδίο συμπλήρωσης "Password".
- btnReturnFirstPage: Κουμπί που επιστρέφει τον χρήστη στην αρχική σελίδα.

Μέθοδοι:

- select(Button: String): καλεί τον controller UserSettingsController που ελέγχει ποιο κουμπί πατήθηκε



και εμφανίζει την κατάλληλη οθόνη.

Boundary LoginErrorMessageGUI

«Boundary» LoginErrorMessageGUI
-lblWrong: String -btnReturnLog: Button
+select(Button: String): void

Χαρακτηριστικά:

- lblWrong: Ετικέτα με το μήνυμα “Λάθος συμπλήρωσή στοιχείων”.
- btnReturnLog: Κουμπί που επιστρέφει στην σελίδα login.

Μέθοδοι:

- select(Button: String): Καλεί τον controller UserSettingsController που ελέγχει ποιο κουμπί πατήθηκε και εμφανίζει την κατάλληλη οθόνη.

Boundary HomePageGUI

Η κλάση αυτή αντιπροσωπεύει την σελίδα «home».

«Boundary» HomePageGUI
-lblEmarket: String -lblProducts: String -lblCategories: String +homeProductList[]: ProductEntity -btnListProdPage[]: Button -btnFavorites: Button -btnOffers: Button -btnStock: Button +srcSearch: Searchbox -btnUser: Button -btnCart: Button -btnHistory: Button -btnLogout: Button -btnAd: Button -categoryButtonList[]: Button -imgLogo: Image
+select(Button: string): void +srcSearchpress(text): void

Χαρακτηριστικά:

- lblEmarket: Ετικέτα με το όνομα της σελίδας.
- lblProducts: Ετικέτα με το μήνυμα Προϊόντα.
- lblCategories: Ετικέτα με το μήνυμα κατηγορίες .
- homeProductList[]: Λίστα προϊόντων που εμφανίζονται στην σελίδα.
- btnListProductPage[]: Μια λίστα με κουμπιά που οδηγούν στις σελίδες του κάθε προϊόντος.
- btnFavorites: Ένα κουμπί που οδηγεί στην σελίδα «αγαπημένα».
- btnOffers: Ένα κουμπί που οδηγεί στην σελίδα «προσφορές».
- btnStock: Ένα κουμπί που οδηγεί στην σελίδα “Stock”.
- srcSearchbox : Ένα κουμπί αναζήτησης προϊόντος.
- btnUser: Ένα κουμπί που οδηγεί στην σελίδα userSettings.
- btnCart: Ένα κουμπί που οδηγεί στη σελίδα καλάθι αγορών.
- btnHistory: Ένα κουμπί που οδηγεί στο ιστορικό παραγγελιών.



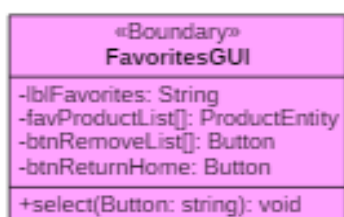
- btnLogout: Ένα κουμπί που αποσυνδέει τον χρήστη.
- btnAd: Ένα κουμπί που στέλνει το χρήστη στο διαφημιζόμενο προϊόν.
- categoryButtonList[]: Μια λίστα κουμπιών για κάθε κατηγορία που ανανεώνουν την αρχική σελίδα.
- imgLogo: Το "logo" της σελίδας.

Μέθοδοι:

- select(Button: String): Καλεί τον controller UserSettingsController που ελέγχει ποιο κουμπί πατήθηκε και εμφανίζει την κατάλληλη οθόνη.
- srcSearchPress(text): Καλεί τον controller UserStettingsController και συγκεκριμένα την συνάρτηση searchpress(text).

Boundary FavoritesGUI

Η κλάση αυτή αντιπροσωπεύει την σελίδα αγαπημένα.



Χαρακτηριστικά

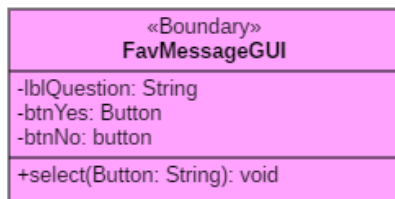
- lblFavorites : Ετικέτα με το μήνυμα Αγαπημένα.
- favProductList[]:Λίστα με τα προϊόντα που ανήκουν στα αγαπημένα.
- btnRemoveList[]: Λίστα κουμπιών που αφαιρούν το αντίστοιχο προϊόν.
- btnReturnHome: Κουμπί που επιστρέφει στην σελίδα home.

Μέθοδοι

- select(Button: String): Καλεί τον controller FavoritesController η UserSettingsController που ελέγχει ποιο κουμπί πατήθηκε και εμφανίζει την κατάλληλη οθόνη.

Boundary FavMessageGUI

Η κλάση αυτή αντιπροσωπεύει το μήνυμα επιβεβαίωσης που εμφανίζεται όταν κάποιος προσθέτει η αφαιρεί από τα αγαπημένα.



Χαρακτηριστικά:

- lblQuestion: Ετικέτα με το μήνυμα "είστε σίγουρος?".
- btnYes: Κουμπί επιβεβαίωσης ενέργειας.
- btnNo: Κουμπί ακύρωσης ενέργειας.

Μέθοδοι:

- select(Button: String): καλεί τον controller FavoritesController ή UserSettingsController που ελέγχει ποιο κουμπί πατήθηκε και εμφανίζει την κατάλληλη οθόνη.



Entity UserEntity

Η κλάση αυτή ορίζει τον χρήστη του συστήματος και τα στοιχεία του.

«Entity» UserEntity
-name: String -surname: String ~username: String -postCode: Int -address: String -floor: Int -phoneNumber: Int -email: String ~password: String -userType: Int
+setName(name: String): void +getName(): String +setUsername(username: String): void +getUsername(): String +setPostCode(postCode: Int): void +getPostCode(): Int +setAddress(address: String): void +getAddress(): String +setFloor(floor: Int): void +getFloor(): Int +setPhoneNumber(phoneNumber: Int): void +getPhoneNumber(): Int +setEmail(email: String): void +getEmail(): String +setPassword(password: String): void +getPassword(): String +setUserType(userType: Int): void +getUserType(): Int

Χαρακτηριστικά:

- Name: Το όνομα του χρήστη.
- Surname: Το επώνυμο του χρήστη.
- Username: Το «όνομα χρήστη».
- PostCode: Ο ταχυδρομικός κώδικας του χρήστη.
- Address: Η διεύθυνση του χρήστη.
- Floor: Ο όροφος του χρήστη.
- phoneNumber: Το τηλέφωνο του χρήστη.
- email: Το "email" του χρήστη.
- Password: το "Password" του χρήστη.
- userType: Ένας int αριθμός που δηλώνει τον τύπο του χρήστη: 1 για απλό χρήστη, 2 για επαγγελματία, 3 για admin.

Μέθοδοι:

- setName(name: String): Η συνάρτηση αυτή δέχεται ένα string και το αποθηκεύει ως όνομα του χρήστη.
- getName(): Η συνάρτηση αυτή επιστρέφει το όνομα του χρήστη.
- setSurname(Surname: String): Η συνάρτηση αυτή δέχεται ένα string και το αποθηκεύει ως επώνυμο του χρήστη.
- getSurname(): Η συνάρτηση αυτή επιστρέφει το επώνυμο του χρήστη.
- setUsername(Username: String): Η συνάρτηση αυτή δέχεται ένα string και το αποθηκεύει ως «όνομα χρήστη».



- getUsername(): Η συνάρτηση αυτή επιστρέφει το «όνομα χρήστη».
- setPostCode(PostCode: int): Η συνάρτηση αυτή δέχεται ένα int και το αποθηκεύει ως ταχυδρομικό κώδικα του χρήστη.
- getPostCode(): Η συνάρτηση αυτή επιστρέφει τον ταχυδρομικό κώδικα του χρήστη.
- setAddress(Address: String): Η συνάρτηση αυτή δέχεται ένα string και το αποθηκεύει ως διεύθυνση του χρήστη.
- getAddress(): Η συνάρτηση αυτή επιστρέφει την διεύθυνση του χρήστη.
- setFloor(floor: int): Η συνάρτηση αυτή δέχεται ένα int και το αποθηκεύει ως όροφο του χρήστη.
- getFloor(): Η συνάρτηση αυτή επιστρέφει τον όροφο του χρήστη.
- setPhoneNumber(phone: int): Η συνάρτηση αυτή δέχεται ένα int και το αποθηκεύει ως αριθμό τηλεφώνου του χρήστη.
- getPhoneNumber(): Η συνάρτηση αυτή επιστρέφει το τηλέφωνο του χρήστη.
- setEmail(email: String): Η συνάρτηση αυτή δέχεται ένα string και το αποθηκεύει ως email του χρήστη.
- getEmail(): Η συνάρτηση αυτή επιστρέφει το email του χρήστη.
- setPassword(password: int): Η συνάρτηση αυτή δέχεται ένα string και το αποθηκεύει ως password του χρήστη.
- getPassword(): Η συνάρτηση αυτή επιστρέφει το "Password" του χρήστη.
- setUserType(UserType: int): Η συνάρτηση αυτή δέχεται ένα int και το αποθηκεύει ως τύπο του χρήστη.
- getUserType(): Η συνάρτηση αυτή επιστρέφει τον τύπο του χρήστη.

Entity ProfessionalUserEntity

Η κλάση αυτή ορίζει τον επαγγελματία χρήστη του συστήματος

«Entity» ProfessionalUserEntity
-companyId: int
+ProfessionalUserEntity(cmpID: int): void +setCompanyId(companyID: int): void +getCompanyId(): int

Χαρακτηριστικά:

- companyId: Ο μοναδικός εταιρικός αριθμός του επαγγελματία χρήστη.

Μέθοδοι:

- ProfesionalUserEnnity(cmpID: int): Ο constructor της συνάρτησης.
- setCompanyId(companyID: int): Η συνάρτηση αυτή δέχεται ένα int και τον καταχωρεί σαν εταιρικό αριθμό του επαγγελματία χρήστη.
- getCompanyId(): Η συνάρτηση αυτή επιστρέφει τον εταιρικό αριθμό του χρήστη.



Entity AdminUserEntity

Η κλάση αυτή ορίζει τον χρήστη «admin» του συστήματος.

«Entity» UserAdminEntity
-adminName: String -adminPassword: String
+UserAdminEntity(adName: String, adPassword): void +setAdminName(adminName: String): void +getAdminName(): String +setAdminPassword(adminPassword: String): void +getAdminPassword(): String

Χαρακτηριστικά:

- adminName: Το όνομα του χρήστη «admin».
- adminPassword: Το password του χρήστη «admin».

Μέθοδοι:

- UserAdminEntity(adName:String,AdPassword:String): Ο constructor της συνάρτησης.
- setAdminName(adminName:String): Η συνάρτηση αυτή παίρνει ένα string και το καταχωρεί όνομα του χρήστη «admin».
- getAdminName(): η συνάρτηση αυτή επιστρέφει το όνομα του χρήστη «admin».
- setAdminPassword(adminPassword:String): Η συνάρτηση αυτή παίρνει ένα string και το καταχωρεί password του χρήστη «admin».
- getAdminPassword(): η συνάρτηση αυτή επιστρέφει το password του χρήστη «admin».

Entity FavoritesEntity

Η κλάση αυτή ορίζει τα αγαπημένα.

«Entity» FavoritesEntity
-favProductList[]: ProductEntity
+FavoritesEntity(favProductList: ProductEntity): void +setFavProductList(favProductList: ProductEntity): void +getFavProductList(): favProductList

Χαρακτηριστικά:

- favProductList[]: Η λίστα με τα προϊόντα των αγαπημένων.

Μέθοδοι:

- FavoritesEntity(favProductList: ProductEntity): Ο constructor της συνάρτησης.
- setFavProductList(favProductList: ProductEntity): Η συνάρτηση αυτή δέχεται σαν όρισμα μια λίστα από προϊόντα και την καταχωρεί ως την λίστα των αγαπημένων.
- getFavProductList(): Επιστρέφει την λίστα προϊόντων των αγαπημένων.

Controller FavoritesController

Ο controller με τις κατάλληλες συναρτήσεις για την διαχείριση των αγαπημένων.

«Controller» FavoritesController
+removeFromFavorites(product: ProductEntity, checkMesButton: boolean): void +addToFavorites(product: ProductEntity, checkMesButton): void +displayMessage(): void +checkMesButton(): boolean +stayOnPage(CheckMesButton: boolean): void +displayFavorites(): void

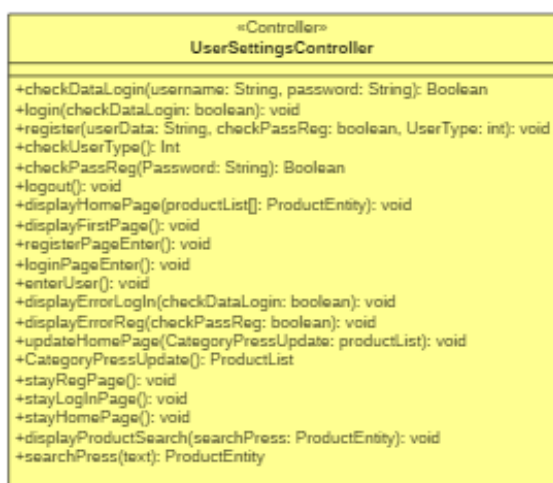


Μέθοδοι:

- `removeFromFavorites(product:ProductEntity,CheckMesButton:boolean)`: Η συνάρτηση δέχεται ένα προϊόν καθώς και την επιβεβαίωση/απόρριψη του χρήστη και αν το επιβεβαίωσε το αφαιρεί από τη λίστα αγαπημένα.
- `addToFavorites(product:ProductEntity,checkMesButton:boolean)`: Η συνάρτηση δέχεται ένα προϊόν καθώς και την επιβεβαίωση/απόρριψη του χρήστη και αν το επιβεβαίωσε το προσθέτει στη λίστα αγαπημένα.
- `displayMessage()`:εμφανίζει το μήνυμα επιβεβαίωσης.
- `checkMesButton()`: ελέγχει αν επιβεβαίωσε ή απέρριψε την ενέργεια ο χρήστης και το καταχωρεί σαν Boolean.
- `stayOnPage(CheckMesButton:Boolean)`: σε περίπτωση που ο χρήστης απέρριψε την ενέργεια το σύστημα παραμένει στην σελίδα από την οποία ήρθε το αίτημα της ενέργειας.
- `displayFavorites()`:Η συνάρτηση αυτή εμφανίζει την σελίδα αγαπημένα.

ControllerUserHandleController

Ο controller που έχει τις κατάλληλες συναρτήσεις για την είσοδο του χρήστη στο σύστημα και την διαχείριση της σελίδας Home.



Μέθοδοι:

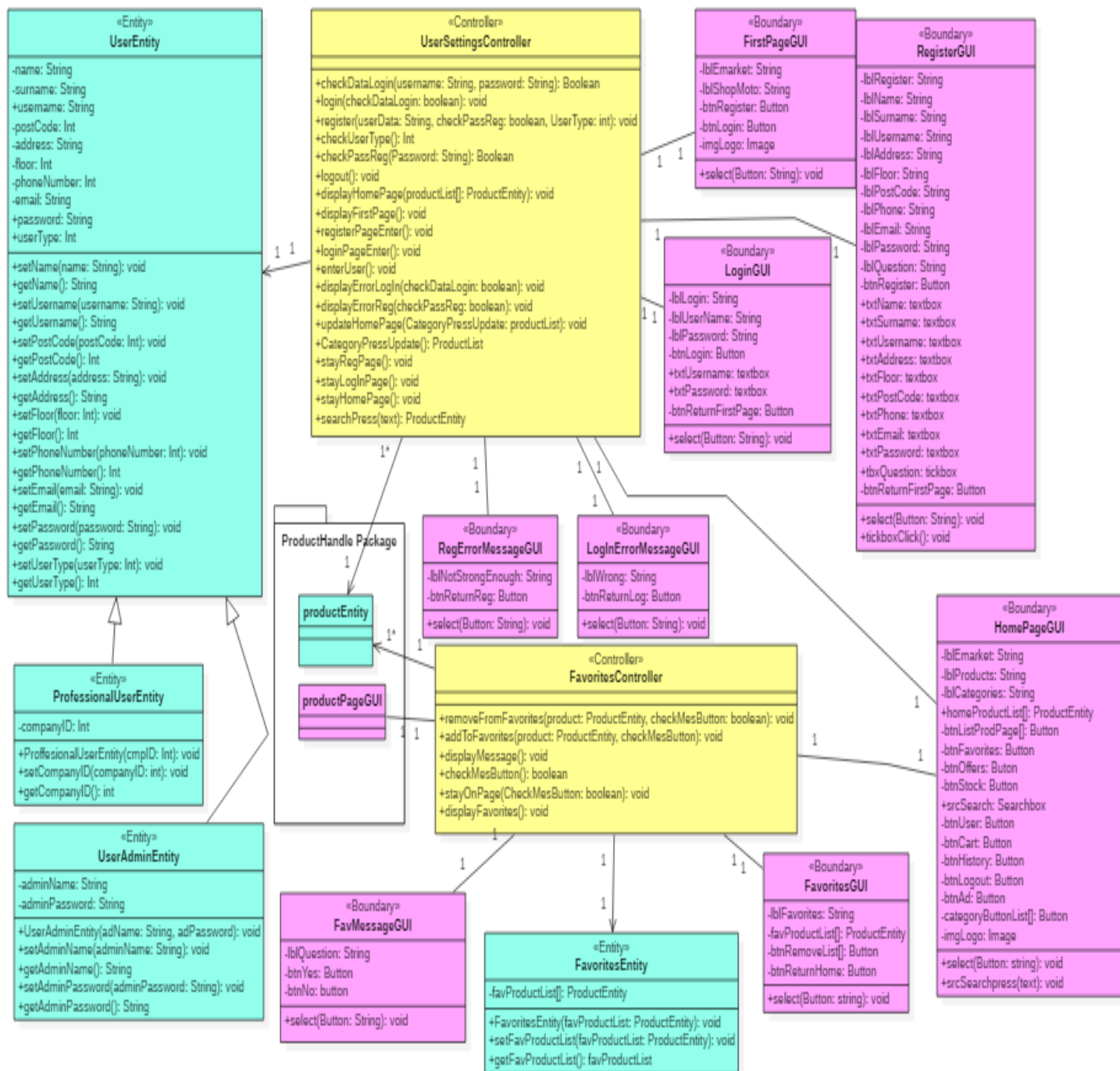
- `checkDataLogin(username:String,Password:String)`: Η συνάρτηση αυτή δέχεται σαν όρισμα 2 string και αν υπάρχουν καταχωρημένα στη βάση δεδομένων ως username, password χρήστη επιστρέφει true αλλιώς false.
- `login(checkDataLogin():boolean)`: Αν το checkDataLogin επέστρεψε true τότε συνδέει τον χρήστη.
- `displayErrorLogin(checkDataLogin:boolean)`: Η συνάρτηση αυτή έχει ως όρισμα το αποτέλεσμα της checkDataLogin και αν έχει επιστρέψει false, τότε εμφανίζει το μήνυμα σφάλματος σύνδεσης "errorLogin".
- `checkPasswordReg(Password:String)`: Η συνάρτηση αυτή ελέγχει το μέγεθος του "Password" και επιστρέφει true αν είναι μεγαλύτερο των 5 χαρακτήρων .
- `CheckUserType()`: επιστρέφει τον τύπο του χρήστη.
- `Register(UserData:String,checkPasswordReg:Boolean,getUserType:int)`: Η συνάρτηση αυτή έχει ορίσματα τα στοιχεία χρήστη το αποτέλεσμα των 2 παραπάνω συναρτήσεων και αν η checkPasswordReg επέστρεψε true καταχωρεί τον χρήστη στη βάση δεδομένων .
- `displayErrorReg(checkPassReg:boolean)`: Η συνάρτηση αυτή έχει όρισμα το αποτέλεσμα της συνάρτησης checkPassReg και αν είναι false τότε εμφανίζει το μήνυμα σφάλματος .



- Logout(): Η συνάρτηση αυτή αποσυνδέει τον χρήστη από το σύστημα και καλεί την displayFirstPage().
- displayHomePage(productList[:ProductEntity): Η συνάρτηση αυτή έχει όρισμα μια λίστα από προϊόντα και εμφανίζει την σελίδα home με τα προϊόντα αυτά.
- displayFirstPage(): Η συνάρτηση αυτή εμφανίζει την αρχική σελίδα.
- registerPageEnter(): Η συνάρτηση αυτή εμφανίζει την σελίδα register.
- loginPageEnter(): Η συνάρτηση αυτή εμφανίζει την σελίδα login.
- enterUser(): Η συνάρτηση αυτή εμφανίζει την σελίδα userSettings.
- categoryPressUpdate(): Η συνάρτηση αυτή ενημερώνει την productList της home ανάλογα με την κατηγορία που πατήθηκε.
- updateHomePage(categoryPressUpdate():ProductList[]): Η συνάρτηση αυτή ενημερώνει την home σελίδα με την λίστα της categoryPressUpdate.
- stayRegPage(): Η συνάρτηση αυτή παραμένει στην σελίδα register με τα προηγούμενα ορίσματα .
- stayLoginPage(): Η συνάρτηση αυτή παραμένει στην σελίδα login με τα προηγούμενα ορίσματα.
- stayHomePage(): Η συνάρτηση αυτή παραμένει στην σελίδα home με τα προηγούμενα ορίσματα.
- searchPress(text): Η συνάρτηση αυτή αναγνωρίζει ποιο προϊόν έχει γράψει ο χρήστης στην αναζήτηση και καλεί την displayProductPage.



2.1.2 Διάγραμμα κλάσεων.

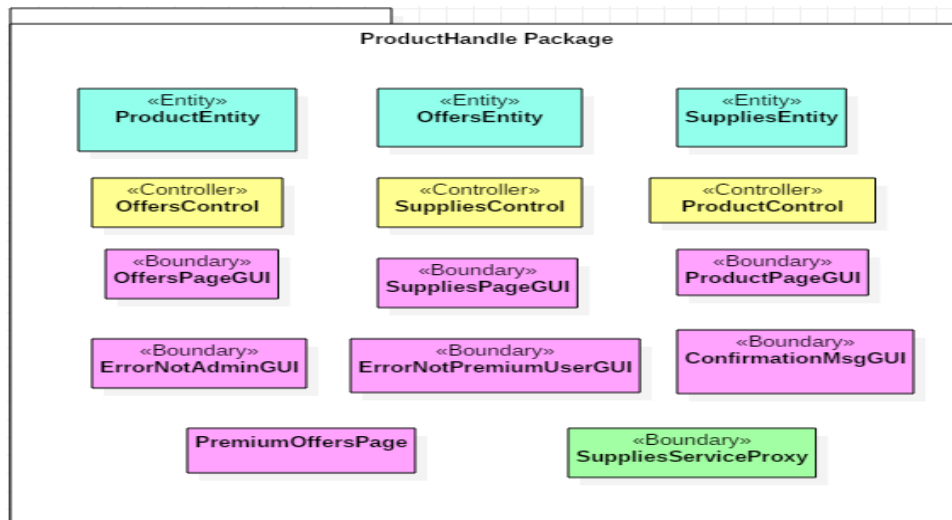


Σχήμα 1. Διάγραμμα κλάσεων πακέτου UserHandle.

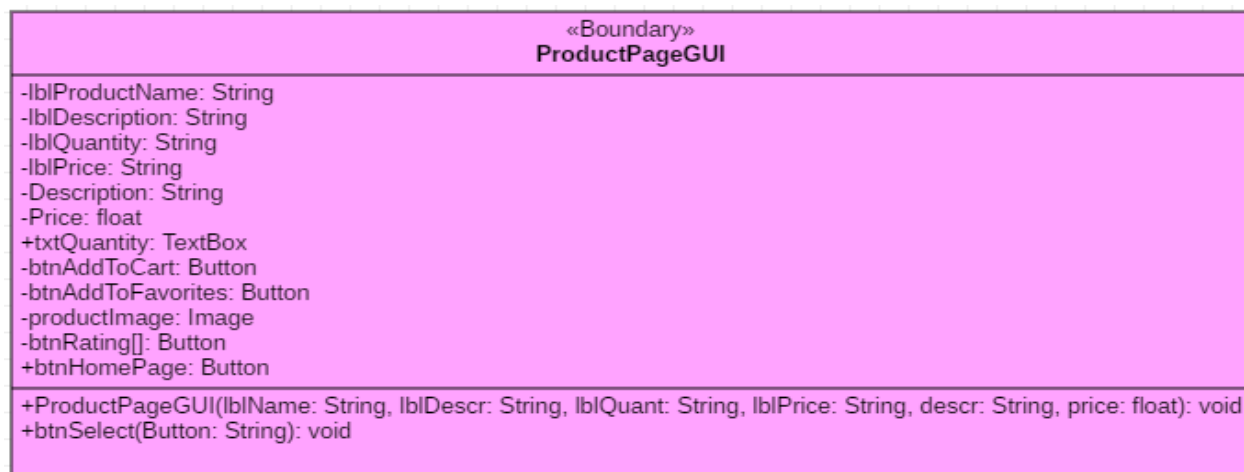


2.1.3 Πακέτο ProductHandle

Το πακέτο αυτό διαχειρίζεται τα προϊόντα. Περιλαμβάνει τη σελίδα προϊόντος, διαχειρίζεται τις προσφορές και τις προμήθειες.



Boundary ProductPageGUI



Η κλάση αυτή είναι η σελίδα του προϊόντος. Ο χρήστης μεταφέρεται εδώ πατώντας το σύνδεσμο του προϊόντος στην αρχική σελίδα του συστήματος.

Χαρακτηριστικά της κλάσης:

- lblProductName: Επιγραφή του ονόματος του προϊόντος.
- lblDescription: Επιγραφή «Περιγραφή» πάνω από την περιγραφή του προϊόντος.
- lblQuantity: Επιγραφή «Ποσότητα» πάνω από το πεδίο συμπλήρωσης της ποσότητας.
- lblPrice: Επιγραφή «Τιμή» πάνω από την τιμή του προϊόντος.
- Description: Κείμενο για την περιγραφή του προϊόντος.
- Price: Τιμή του προϊόντος.



- txtQuantity: Πεδίο συμπλήρωσης της επιθυμητής ποσότητας του προϊόντος προς αγορά.
- btnAddToCart: Κουμπί προσθήκης του προϊόντος στο καλάθι.
- btnAddToFavorites: Κουμπί προσθήκης του προϊόντος στην λίστα αγαπημένων του χρήστη.
- productImage: Εικόνα του προϊόντος.
- btnRating[]: 3 κουμπιά αξιολόγησης του προϊόντος από το 1 έως το 3 αντίστοιχα.
- btnHomePage: Κουμπί επιστροφής στην αρχική σελίδα του συστήματος.

Μέθοδοι της κλάσης:

- ProductPageGUI(): Η συνάρτηση αυτή είναι ένας δομητής της κλάσης ProductPageGUI.
- btnSelect(Button:String): Η συνάρτηση αυτή ανάλογα με ποιο κουμπί πάτησε ο χρήστης καλεί την αντίστοιχη συνάρτηση του εκάστοτε controller.

Boundary OffersPageGUI

«Boundary» OffersPageGUI
-lblPageName: String -lblProductNameList []: String -lblReducedPriceList []: float -productImageList []: Image -lblDiscountList[]: String -btnProductButtonList []: Button -btnPremiumOffers: Button +btnHomePage: Button
+OffersPageGUI(lblPaName: String, lblPrNameList []: String, lblRedPriceList []: float, lblDiscList[]: String) +btnSelect(Button: String): void

Η κλάση αυτή είναι η σελίδα με τις προσφορές προϊόντων. Ο χρήστης μεταφέρεται εδώ πατώντας το σύνδεσμο «Προσφορές» στην αρχική σελίδα.

Χαρακτηριστικά της κλάσης:

- lblPageName: Επιγραφή του ονόματος της σελίδας - «Προσφορές».
- lblProductNameList[]: Λίστα από επιγραφές με τα ονόματα του κάθε προϊόντος πάνω από την εικόνα κάθε προϊόντος.
- lblReducedPriceList[]: Λίστα από επιγραφές πάνω από τη μειωμένη τιμή κάθε προϊόντος.
- productImageList[]: Λίστα από εικόνες για κάθε προϊόν.
- lblDiscountList[]: Λίστα από επιγραφές του ποσοστού έκπτωσης για κάθε προϊόν.
- btnProductButtonList []: Λίστα από συνδέσμους προϊόντων κάτω από την εικόνα του κάθε προϊόντος.
- btnPremiumOffers: Σύνδεσμος που οδηγεί στην σελίδα προσφορών για επαγγελματίες χρήστες.
- btnHomePage: Σύνδεσμος που επιστρέφει τον χρήστη στην αρχική σελίδα.

Μέθοδοι της κλάσης:

- OffersPageGUI(): Η συνάρτηση αυτή αποτελεί τον δομητή της κλάσης OffersPageGUI.
- btnSelect(Button:String): Η συνάρτηση αυτή ανάλογα με ποιο κουμπί πάτησε ο χρήστης καλεί την αντίστοιχη συνάρτηση του εκάστοτε controller.



Boundary PremiumOffersPage

«Boundary» PremiumOffersPage
-lblPageName: String -lblProductNameList []: String -lblReducedPriceList []: float -productImageList []: Image -lblDiscountList[]: String -btnProductButtonList []: Button +btnHomePage: Button
+PremiumOffersPageGUI(): void +btnSelect(Button: String): void

Η κλάση αυτή είναι η σελίδα με τις προσφορές προϊόντων για premium users. Ο χρήστης μεταφέρεται εδώ πατώντας το σύνδεσμο «Προσφορές για επαγγελματίες» στην αρχική σελίδα.

Χαρακτηριστικά της κλάσης:

- lblPageName: Επιγραφή του ονόματος της σελίδας - «Προσφορές για επαγγελματίες».
- lblProductNameList[]: Λίστα από επιγραφές με τα ονόματα του κάθε προϊόντος πάνω από την εικόνα κάθε προϊόντος.
- lblReducedPriceList[]: Λίστα από επιγραφές πάνω από τη μειωμένη τιμή κάθε προϊόντος.
- productImageList[]: Λίστα από εικόνες για κάθε προϊόν.
- lblDiscountList[]: Λίστα από επιγραφές του ποσοστού έκπτωσης για κάθε προϊόν.
- btnProductButtonList []: Λίστα από συνδέσμους προϊόντων κάτω από την εικόνα του κάθε προϊόντος.
- btnHomePage: Σύνδεσμος που επιστρέφει τον χρήστη στην αρχική σελίδα.

Μέθοδοι της κλάσης:

- PremiumOffersPage(): Η συνάρτηση αυτή αποτελεί τον δομητή της κλάσης PremiumOffersPage.
- btnSelect(Button:String): Η συνάρτηση αυτή ανάλογα με ποιό κουμπί πάτησε ο χρήστης καλεί την αντίστοιχη συνάρτηση του εκάστοτε controller.

Boundary SuppliesPageGUI

«Boundary» SuppliesPageGUI
-lblPageName: String -lblChooseProd: String -lblChooseQuant: String -lblChooseSupplier: String +txtProductID: TextBox +txtProductQuant: TextBox +txtProductSupplier: TextBox -btnSubmitSupplyOrder: Button +btnHomePage: Button
+SuppliesPageGUI(lblName: String, lblProd: String, lblQuant: String, lblSup: String): void +btnSelect(Button: String): void

Η κλάση αυτή είναι η σελίδα από την οποία ο χρήστης admin δημιουργεί παραγγελίες προϊόντων προς τους προμηθευτές όταν εξαντλείται το εμπόρευμα του καταστήματος. Ο admin μεταφέρεται εδώ πατώντας το σύνδεσμο «Ενημέρωση Stock» στην αρχική σελίδα.



Χαρακτηριστικά της κλάσης:

- lblPageName: Επιγραφή του ονόματος της σελίδας - «Ενημέρωση Stock».
- lblChooseProd: Επιγραφή πάνω από το πεδίο εισαγωγής προϊόντος προς παραγγελία.
- lblChooseQuant: Επιγραφή πάνω από το πεδίο εισαγωγής ποσότητας του προϊόντος.
- lblChooseSupplier: Επιγραφή πάνω από το πεδίο εισαγωγής προμηθευτή.
- txtProductID: Πεδίο εισαγωγής του id του προϊόντος προς παραγγελία.
- txtProductQuant: Πεδίο εισαγωγής της ποσότητας του προϊόντος προς παραγγελία.
- txtProductSupplier: Πεδίο εισαγωγής του προμηθευτή στον οποίο θα σταλεί η παραγγελία.
- btnSubmitSupplyOrder: Κουμπί υποβολής της παραγγελίας του admin προς τον προμηθευτή.
- btnHomePage: Κουμπί που επιστρέφει τον χρήστη στην αρχική σελίδα.

Μέθοδοι της κλάσης:

- SuppliesPageGUI(): Η συνάρτηση αυτή αποτελεί τον δομητή της κλάσης SuppliesPageGUI.
- btnSelect(Button:String): Η συνάρτηση αυτή ανάλογα με ποιό κουμπί πάτησε ο χρήστης καλεί την αντίστοιχη συνάρτηση του εκάστοτε controller.

Boundary ErrorNotPremiumUserGUI

«Boundary» ErrorNotPremiumUserGUI
-lblShowError: String -btnBack: Button
+ErrorNotPremiumUserGUI(): void +btnSelect(Button: String): void

Η κλάση αυτή είναι το παράθυρο που εμφανίζεται στον απλό χρήστη όταν πατήσει το σύνδεσμο προσφορές για επαγγελματίες που βρίσκεται στη σελίδα των προσφορών.

Χαρακτηριστικά της κλάσης:

- lblShowError: Μήνυμα σφάλματος.
- btnBack: Κουμπί που επιστρέφει τον χρήστη στη σελίδα των προσφορών.

Μέθοδοι της κλάσης:

- ErrorNotPremiumUserGUI(): Η συνάρτηση αυτή αποτελεί τον δομητή της κλάσης ErrorNotPremiumUserGUI.
- btnSelect(Button:String): Η συνάρτηση αυτή ανάλογα με ποιό κουμπί πάτησε ο χρήστης καλεί την αντίστοιχη συνάρτηση του εκάστοτε controller.

Boundary ErrorNotAdminGUI

«Boundary» ErrorNotAdminGUI
-lblShowError: String -btnBack: Button
+ErrorNotAdminGUI(): void +btnSelect(Button: String): void



Η κλάση αυτή είναι το παράθυρο που εμφανίζεται στο χρήστη όταν πατήσει το σύνδεσμο ενημέρωση Stock στην αρχική σελίδα και δεν είναι ο admin.

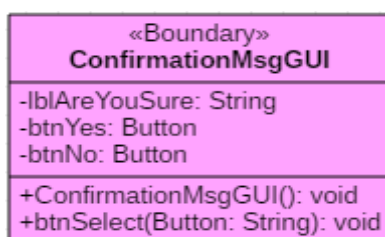
Χαρακτηριστικά της κλάσης:

- lblShowError: Μήνυμα σφάλματος.
- btnBack: Κουμπί που επιστρέφει τον χρήστη στην αρχική σελίδα.

Μέθοδοι της κλάσης:

- ErrorNotAdminGUI(): Η συνάρτηση αυτή αποτελεί τον δομητή της κλάσης ErrorNotAdminGUI.
- btnSelect(Button:String): Η συνάρτηση αυτή ανάλογα με ποιο κουμπί πάτησε ο χρήστης καλεί την αντίστοιχη συνάρτηση του εκάστοτε controller.

Boundary ConfirmationMsgGUI



Η κλάση αυτή είναι το παράθυρο που εμφανίζεται στον admin όταν πατήσει το κουμπί submit στη σελίδα ενημέρωσης stock για να επιβεβαιώσει την απόφασή του.

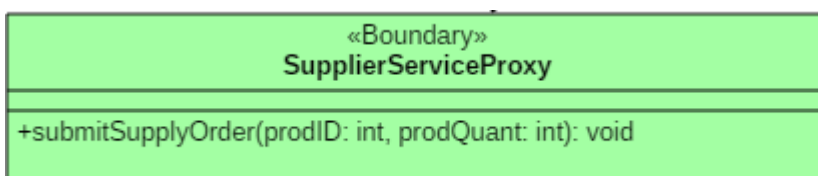
Χαρακτηριστικά της κλάσης:

- lblAreYouSure: Μήνυμα επιβεβαίωσης.
- btnYes: Κουμπί που κατοχυρώνει την παραγγελία και ξεκινά τη διαδικασία αποστολής της στους προμηθευτές.
- btnNo: Κουμπί που επιστρέφει τον admin στην σελίδα ενημέρωσης stock.

Μέθοδοι της κλάσης:

- ConfirmationMsgGUI(): Η συνάρτηση αυτή αποτελεί τον δομητή της κλάσης ConfirmationMsgGUI.
- btnSelect(Button:String): Η συνάρτηση αυτή ανάλογα με ποιο κουμπί πάτησε ο χρήστης καλεί την αντίστοιχη συνάρτηση του εκάστοτε controller.

Boundary SupplierServiceProxy



Είναι η κλάση με την οποία γίνεται υποβολή της παραγγελίας του admin στον κατάλληλο προμηθευτή.

Μέθοδοι της κλάσης:

- submitSupplyOrder(prodID : int, prodQuant : int): Η συνάρτηση αυτή παίρνει σαν ορίσματα το id και την ποσότητα του προϊόντος προς παραγγελία και τα καταχωρεί στη βάση δεδομένων του επιλεγμένου προμηθευτή.



Entity ProductEntity

«Entity» ProductEntity
-productName: String -productPrice: float -productQuantAvail: int -productID: int -productRating: int -productDiscr: String
+ProductEntity(name: string, price: float, availquant: int, id: int, rating: int, discr: String): void +setProductName(name : String): void +setProductPrice(price : float): void +setProductQuantAvail(quant : int): void +setProductID(id : int): void +setProductRating(rating : int): void +setProductDiscr(discr : String): void +getProductName(): String +getProductPrice(): float +getProductQuantAvail(): int +getProductID(): int +getProductRating(): int +getProductDiscr(): String

Είναι η κλάση με την οποία δημιουργούνται όλα τα προϊόντα-objects με τα χαρακτηριστικά τους.

Χαρακτηριστικά της κλάσης:

- productName: Όνομα προϊόντος.
- productPrice: Η τιμή πώλησης του προϊόντος.
- productQuantAvail: Η διαθέσιμη στο stock ποσότητα του προϊόντος.
- productID: Ένας int μοναδικός για κάθε προϊόν.
- productRating: Ένας int από το 1 έως το 3 για την αξιολόγηση του προϊόντος.
- productDiscr: Περιγραφή του προϊόντος – κείμενο.

Μέθοδοι της κλάσης:

- ProductEntity(): Η συνάρτηση αυτή είναι ο δομητής της κλάσης.
- setProductName(name :String): Θέτει το όνομα του προϊόντος σύμφωνα με το όρισμα.
- setProductPrice(price :float): Θέτει την τιμή του προϊόντος σύμφωνα με το όρισμα.
- setProductQuantAvail(quant :int): Θέτει την διαθέσιμη στο stock ποσότητα σύμφωνα με το όρισμα.
- setProductID(id :int): Θέτει το id του προϊόντος σύμφωνα με το όρισμα.
- setProductRating(rating :int): Θέτει το rating του προϊόντος σύμφωνα με το όρισμα.
- setProductDiscr(discr :String): Θέτει την περιγραφή του προϊόντος σύμφωνα με το όρισμα.
- getProductName(): Επιστρέφει το όνομα προϊόντος.
- getProductPrice(): Επιστρέφει την τιμή προϊόντος.
- getProductQuantAvail(): Επιστρέφει την διαθέσιμη στο stock ποσότητα του προϊόντος.
- getProductID(): Επιστρέφει το id του προϊόντος.
- getProductRating(): Επιστρέφει το rating του προϊόντος.
- getProductDiscr(): Επιστρέφει την περιγραφή του προϊόντος.



Entity OffersEntity

«Entity» OffersEntity
-productList []: ProductEntity -discountList []: float
+OffersEntity(prodList []: ProductEntity, discList []: float): void +setProductList(prodList []: ProductEntity): void +setDiscountList(discList []: float): void +getProductList(): ProductEntity +getDiscountList(): float

Είναι η κλάση από την οποία δημιουργούνται τα object - προσφορές.

Χαρακτηριστικά της κλάσης:

- productList []: Λίστα με τα προϊόντα – objects που είναι σε προσφορά.
- discountList []: Λίστα με τους συντελεστές έκπτωσης του κάθε προϊόντος σε προσφορά.

Μέθοδοι της κλάσης:

- OffersEntity(): Συνάρτηση – δομητής της κλάσης.
- setProductList(prodList []:ProductEntity): Θέτει τη λίστα με τα προϊόντα σε προσφορά σύμφωνα με το όρισμα.
- setDiscountList(discList []:float) : Θέτει τη λίστα με τους συντελεστές έκπτωσης σύμφωνα με το όρισμα.
- getProductList(): Επιστρέφει τη λίστα με τα προϊόντα σε προσφορά.
- getDiscountList(): Επιστρέφει τη λίστα με τους συντελεστές έκπτωσης.

Entity SuppliesEntity

«Entity» SuppliesEntity
-prodSupplyID: int -prodSupplyQuant: int -prodSupplier: String
+SuppliesEntity(prodID: int, prodQuant: int, prodSupplier: String): void +setProdSupplyID(prodID: int): void +setProdSupplyQuant(prodQuant: int): void +setProdSupplier(prodSupplier: String): void +getProdSupplyID(): int +getProdSupplyQuant(): int +getProdSupplier(): String

Είναι η κλάση με την οποία δημιουργούνται objects – παραγγελίες το καθένα από τα οποία αναφέρεται σε ένα προϊόν. Οι παραγγελίες αυτές δημιουργούνται από τον admin και προορίζονται για αποστολή στον προμηθευτή.

Χαρακτηριστικά της κλάσης:



- prodSupplyID: Το id του προϊόντος προς παραγγελία.
- prodSupplyQuant: Η επιθυμητή ποσότητα του προϊόντος προς παραγγελία.
- prodSupplier: Το όνομα του προμηθευτή του προϊόντος.

Μέθοδοι της κλάσης:

- SuppliesEntity(): Συνάρτηση – δομητής της κλάσης.
- setProdSupplyID(prodID:int): Θέτει το id του προϊόντος προς παραγγελία σύμφωνα με το όρισμα.
- setProdSupplyQuant(prodQuant:int): Θέτει την επιθυμητή ποσότητα του προϊόντος προς παραγγελία.
- setProdSupplier(prodSupplier:String): Θέτει το όνομα του προμηθευτή του προϊόντος.
- getProdSupplyID(): Επιστρέφει το id του προϊόντος προς παραγγελία.
- getProdSupplyQuant(): Επιστρέφει την επιθυμητή ποσότητα του προϊόντος προς παραγγελία.
- getProdSupplier(): Επιστρέφει το όνομα του προμηθευτή του προϊόντος.

Controller OffersController

«Controller» OffersController
+displayOffersPage(offers : OffersEntity): void +checkUserType(userType : int): boolean +displayPremium(checkType: boolean): void +displayErrorNotPremiumUserGUI(): void +displayPremiumOffersPage(offers: OffersEntity): void +stayOnOffersPage(): void +returnToHomePage(): void

Είναι η κλάση που διαχειρίζεται με τις μεθόδους της τις προσφορές.

Μέθοδοι της κλάσης:

- displayOffersPage(offers : OffersEntity): Εμφανίζει την σελίδα προσφορών αφού του δώσουμε σαν είσοδο ένα object τύπου OffersEntity. Η συνάρτηση αυτή καλείται από το HomePageGUI όταν ο χρήστης πατήσει τον σύνδεσμο «Προσφορές» στην αρχική σελίδα.
- checkUserType(userType : int): Παίρνει σαν είσοδο το userType και αν ο αριθμός αυτός αντιστοιχεί σε επαγγελματία – premium χρήστη επιστρέφει true. Η συνάρτηση αυτή καλείται από το OffersPageGUI όταν ο χρήστης πατήσει το σύνδεσμο «Προσφορές για επαγγελματίες» στη σελίδα προσφορών.
- displayPremium(checkType:boolean): Παίρνει σαν είσοδο την έξοδο της προηγούμενης συνάρτησης. Αν αυτή είναι false δηλαδή ο χρήστης δεν είναι premium, καλεί την συνάρτηση displayErrorNotPremiumUserGUI() αλλιώς καλεί την displayPremiumOffersPage(offers:OffersEntity). Η displayPremium καλείται αμέσως μετά την checkUserType.
- displayErrorNotPremiumUserGUI(): Εμφανίζει το παράθυρο σφάλματος σε περίπτωση που ένας μη επαγγελματίας χρήστης πατήσει το σύνδεσμο «Προσφορές για επαγγελματίες».
- displayPremiumOffersPage(offers:OffersEntity): Εμφανίζει τη σελίδα προσφορών για επαγγελματίες παίρνοντας σαν όρισμα ένα object τύπου OffersEntity.
- stayOnOffersPage(): Γυρίζει τον χρήστη πίσω στη σελίδα προσφορών. Η συνάρτηση αυτή καλείται από το ErrorNotPremiumUserGUI όταν ο χρήστης πατήσει το κουμπί back.
- returnToHomePage(): Η συνάρτηση αυτή καλείται όταν πατήσει ο χρήστης το κουμπί HomePage στη σελίδα προσφορών ή στη σελίδα προσφορές για επαγγελματίες.



Controller SuppliesController

«Controller» SuppliesController
<pre>+displaySupplies(checkType: boolean): void +displaySuppliesPageGUI(): void +displayErrorNotAdminGUI(): void +checkUserType(userType: int): boolean +sendSuppliesOrder(prodID: int, prodQuant: int, prodSupplier: String): void +displayConfirmationMsgGUI(): void +stayOnHomePage(): void +stayOnSuppliesPage(): void +returnToHomePage(): void</pre>

Είναι η κλάση που διαχειρίζεται με τις μεθόδους της την ενημέρωση του stock.

Μέθοδοι της κλάσης:

- `displaySupplies(checkType:boolean)`: Η συνάρτηση αυτή παίρνει σαν όρισμα την έξοδο της `checkUserType`. Αν αυτή είναι `true` καλεί την συνάρτηση `displaySuppliesPage` αλλιώς καλεί την `displayErrorNotAdminGUI`. Η `displaySupplies` καλείται αμέσως μετά την `checkUserType`.
- `displaySuppliesPage()`: Η συνάρτηση αυτή εμφανίζει τη σελίδα ενημέρωσης stock.
- `displayErrorNotAdminGUI()`: Η συνάρτηση αυτή εμφανίζει το παράθυρο σφάλματος σε περίπτωση που ένας μη admin χρήστης πατήσει το σύνδεσμο «Ενημέρωση stock».
- `checkUserType(userType:int)`: Η συνάρτηση αυτή παίρνει σαν όρισμα τον ακέραιο `userType` και αν αυτός είναι εκείνος που αντιστοιχεί στον admin επιστρέφει `true` αλλιώς `false`. Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει τον σύνδεσμο «Ενημέρωση stock» στην αρχική σελίδα.
- `sendSuppliesOrder(prodID:int, prodQuant:int, prodSupplier:String): void`. Η συνάρτηση αυτή καλείται από το `ConfirmationMsgGUI` αν ο admin πατήσει το κουμπί Yes. Η συνάρτηση αυτή παίρνει σαν ορίσματα το όνομα του supplier, το id και την ποσότητα του προϊόντος προς παραγγελία που εισήγαγε ο admin κατά την συμπλήρωση της φόρμας παραγγελίας και καλεί ανάλογα με το όνομα του προμηθευτή την `submitSupplyOrder()` από το `SupplierProxy`.
- `displayConfirmationMsgGUI()`: Η συνάρτηση αυτή καλείται μόλις πατηθεί από τον admin το κουμπί `SubmitSupplyOrder` στη σελίδα ενημέρωσης stock. Εμφανίζει παράθυρο με μήνυμα επιβεβαίωσης.
- `stayOnHomePage()`: Η συνάρτηση αυτή καλείται από το `ErrorNotAdminGUI` όταν πατηθεί το κουμπί back. Επιστρέφει το χρήστη στην αρχική σελίδα.
- `stayOnSuppliesPage()`: Η συνάρτηση αυτή καλείται από το `ConfirmationMsgGUI` όταν πατηθεί το κουμπί No. Επιστρέφει το χρήστη στην σελίδα ενημέρωσης stock.
- `returnToHomePage()`: Επιστρέφει το χρήστη στην αρχική σελίδα. Καλείται από το `SuppliesPageGUI` όταν πατηθεί το κουμπί `HomePage`.

Controller ProductController

«Controller» ProductController
<pre>+displayProductPage(product : ProductEntity): void +rateProduct(rating: int): void +returnToHomePage(): void</pre>

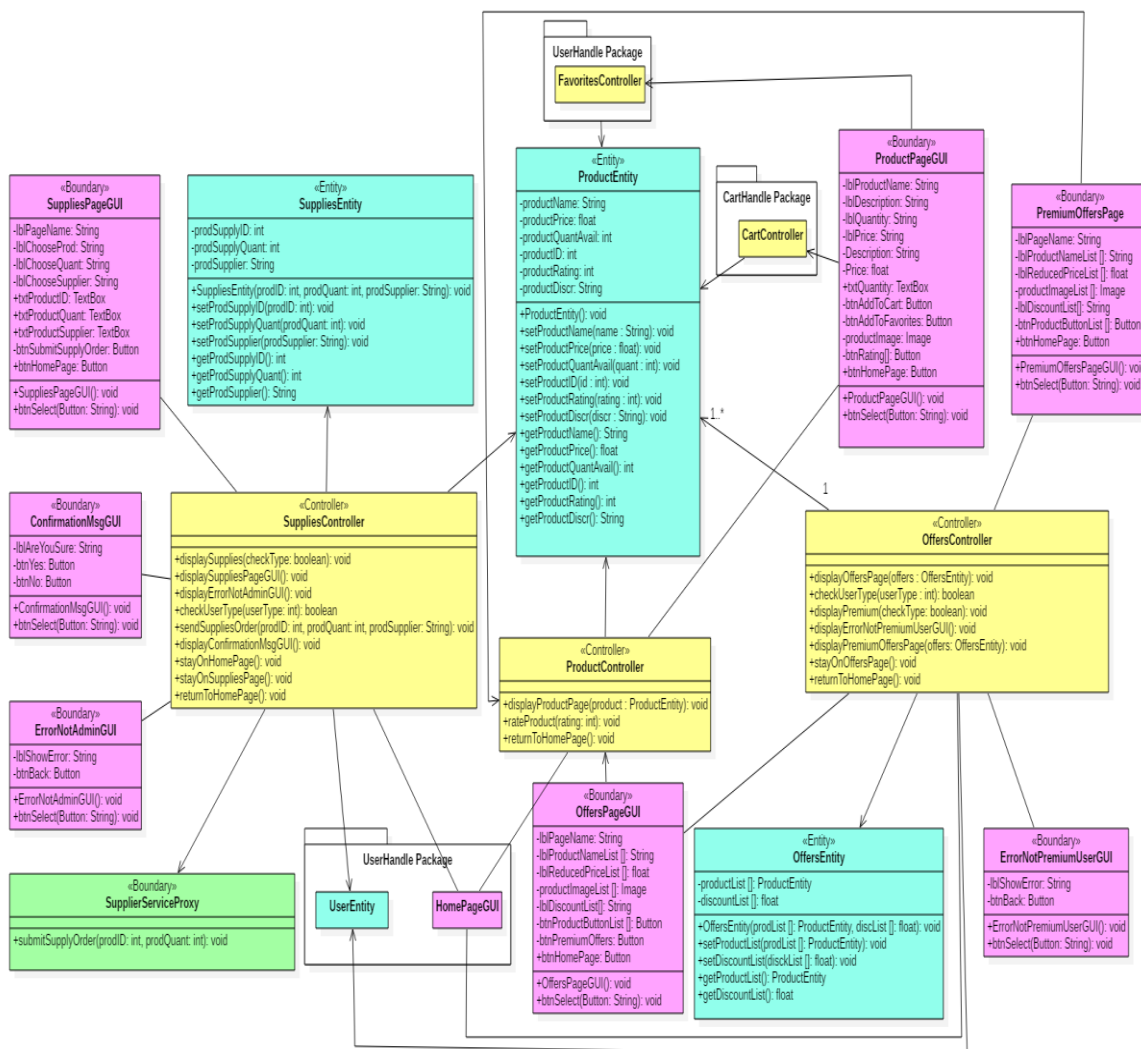
Είναι η κλάση που διαχειρίζεται με τις μεθόδους της τα προϊόντα.



Μέθοδοι της κλάσης:

- **displayProductPage(product :ProductEntity):** Εμφανίζει την σελίδα του προϊόντος αφού του δώσουμε σαν είσοδο ένα object τύπου ProductEntity. Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει τον σύνδεσμο του προϊόντος στην αρχική σελίδα, στη σελίδα προσφορών ή στη σελίδα προσφορών για επαγγελματίες.
- **rateProduct(rating:int):** Η συνάρτηση αυτή καλείται όταν ο χρήστης από την σελίδα του προϊόντος πατήσει ένα από τα κουμπιά αξιολόγησης του προϊόντος. Ανάλογα με ποιο κουμπί πατήθηκε περνάει το αντίστοιχο όρισμα στη συνάρτηση η οποία ανανεώνει κατάλληλα την τιμή του productRating.
- **returnToHomePage():** Επιστρέφει το χρήστη στην αρχική σελίδα. Καλείται από το ProductPageGUI όταν πατηθεί το κουμπί HomePage.

2.1.4 Διάγραμμα κλάσεων.



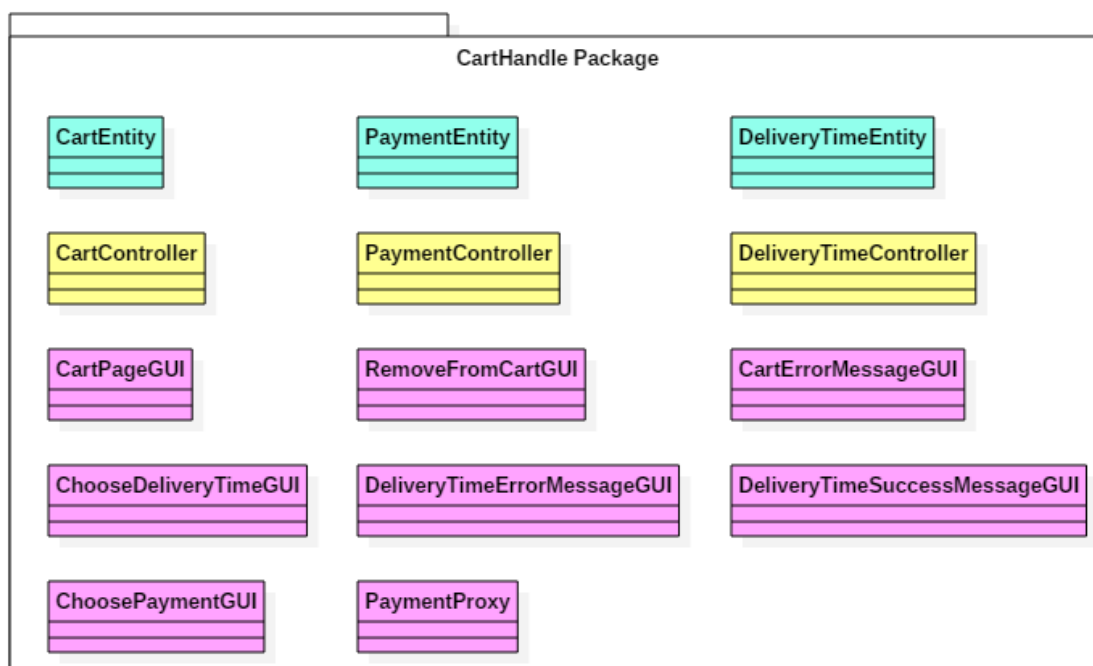
Σχήμα 2. Διάγραμμα κλάσεων πακέτου ProductHandle.



2.2 <Πακέτα λεξιλογίου σεναρίων μέσης προτεραιότητας>

Το πακέτο αυτό περιλαμβάνει τις κλάσεις διαχείρισης για το καλάθι αγορών, τον τρόπο πληρωμής για την διεκπεραίωση της συναλλαγής καθώς και για την επιλογή της ώρας παράδοσης της παραγγελίας.

2.2.1 Πακέτο CartHandle



Boundary CartPageGUI:

«Boundary» CartPageGUI
-lblCart: String -lblQuantity: String -lblPrice: String -lblProductName: String -btnRemoveList[]: Button -btnDeliveryTime: String -btnPayment: String -btnSubmitOrder: String
+select(Button: String): void

Η κλάση αυτή συνιστά το γραφικό περιβάλλον διεπαφής για το καλάθι αγορών.

Χαρακτηριστικά της κλάσης:

- lblCart: Μια ετικέτα που αναγράφει «Καλάθι αγορών».
- lblQuantity: Μια ετικέτα που αναγράφει «Ποσότητα».
- lblPrice: Μια ετικέτα που αναγράφει «Τιμή».
- lblProductName: Μια ετικέτα που αναγράφει προϊόν.
- btnRemoveList[]: Μία λίστα με κουμπιά «Αφαίρεση», που αντιστοιχεί σε κάθε ένα προϊόν.
- btnDeliveryTime: Ένα κουμπί που μεταφέρει τον χρήστη στην σελίδα ChooseDeliveryTimeGUI.
- btnPayment: Ένα κουμπί που μεταφέρει τον χρήστη στην σελίδα ChoosePaymentGUI.



- `btnSubmitOrder`: Ένα κουμπί που οδηγεί στον controller `OrderController`, ο οποίος εκτελεί τις εξής συναρτήσεις την μία μετά την άλλη: `createOrder`, `calculateDeliveryCost`, `displaySubmitOrderPageGUI`.

Μέθοδοι της κλάσης:

- `select(Button: String)`: Καλείται όταν πατηθεί κάποιο από τα παραπάνω κουμπιά: `btnRemoveList[]`, `btnDeliveryTime`, `btnPayment`, `btnSubmitOrder`. Η λειτουργικότητα αυτών παρουσιάζεται παραπάνω.

Boundary RemoveFromCartGUI:

«Boundary» RemoveFromCartGUI
-lblSelectQuantityToRemove: String -btnContinue: String -txtQuantity: Textbox
+select(Button: String): void

Πρόκειται για το παράθυρο που εμφανίζεται, όταν ο χρήστης επιθυμεί να αφαιρέσει ένα προϊόν από το καλάθι αγορών.

Χαρακτηριστικά της κλάσης:

- `lblSelectQuantityToRemove`: Μία ετικέτα που εμφανίζει το μήνυμα «Επιλογή ποσότητας προϊόντος προς αφαίρεση».
- `btnContinue`: Ένα κουμπί που υποβάλλει το αίτημα για την μείωση της ποσότητας, καλεί τον controller `CartController`, ελέγχει εάν η ποσότητα προς αφαίρεση είναι έγκυρη και εάν είναι εμφανίζει την σελίδα `CartPageGUI`, εάν όχι εμφανίζει την σελίδα `CartErrorMessageGUI`.
- `txtQuantity`: Textbox στο οποίο εισάγεται η επιθυμητή ποσότητα προϊόντος προς αφαίρεση.

Μέθοδοι της κλάσης:

- `select(Button: String)`: Καλείται όταν πατηθεί το μοναδικό κουμπί «Συνέχεια», καλεί τον controller `CartController`, ο οποίος ελέγχει το αίτημα για την μείωση της ποσότητας και αναλόγως εμφανίζει την σελίδα `CartPageGUI` ή `CartErrorMessageGUI`.

Boundary CartErrorMessageGUI:

«Boundary» CartErrorMessageGUI
-lblErrorSelectedQuantityMessage: String -btnReturnToPreviousPage: String
+select(Button: String): void

Πρόκειται για το αναδυόμενο μήνυμα, σε περίπτωση λανθασμένης επιλογής ποσότητας προσθαφαίρεσης προϊόντος από το καλάθι αγορών.

Χαρακτηριστικά της κλάσης:

- `lblErrorSelectedQuantityMessage`: Μία ετικέτα που εμφανίζει το μήνυμα «Λανθασμένη επιλογή ποσότητας προσθήκης ή αφαίρεσης προϊόντος στο καλάθι αγορών».
- `btnReturnToPreviousPage`: Ένα κουμπί που όταν πατηθεί, μεταφέρει τον χρήστη στην προηγούμενη σελίδα στην οποία βρισκόταν, δηλαδή στην σελίδα προϊόντος αν πάτησε προσθήκη με αιτούμενη ποσότητα μεγαλύτερη από την διαθέσιμη ή στην σελίδα επιλογής ποσότητας προς αφαίρεση αν ζήτησε να αφαιρεθεί ποσότητα μεγαλύτερη από αυτήν που ήδη υπήρχε για το προϊόν στο καλάθι αγορών.



Μέθοδοι της κλάσης:

- select(Button: String): Καλείται όταν πατηθεί το μοναδικό κουμπί επιστροφής στην προηγούμενη σελίδα btnReturnToPreviousPage, του οποίου η λειτουργικότητα αναλύθηκε ακριβώς από πάνω.

Boundary ChooseDeliveryTimeGUI:

«Boundary» ChooseDeliveryTimeGUI
-lblAvailableDeliveryTime: String -lblPleaseChooseDeliveryTime: String -lblDeliveryTimeList[]: String -btnCheckDeliveryTimeList[]: Button -btnSaveMyChoice: String
+select(Button: String): void

Η κλάση αυτή ορίζει το γραφικό περιβάλλον διεπαφής για την επιλογή από τον χρήστη της ώρας παράδοσης της παραγγελίας του.

Χαρακτηριστικά της κλάσης:

- lblAvailableDeliveryTime: Μία ετικέτα που γράφει «Διαθέσιμες ώρες παράδοσης».
- lblPleaseChooseDeliveryTime: Μία ετικέτα που γράφει «Παρακαλώ, επιλέξτε μία από τις παρακάτω».
- lblDeliveryTimeList[]: Μία λίστα με ετικέτες που δηλώνουν την ώρα παράδοσης προς επιλογή.
- btnCheckDeliveryTimeList[]: Μία λίστα με κουμπιά επιλογής δίπλα από την λίστα με τις παραπάνω ετικέτες.
- btnSaveMyChoice: Ένα κουμπί «Αποθήκευσης της επιλογής μου», ώστε να καταχωρηθεί η τελική επιλογή του χρήστη.

Μέθοδοι της κλάσης:

- select(Button: String): Εκτελείται όταν πατηθεί κάποιο από τα παραπάνω κουμπιά. Πιο συγκεκριμένα, το κουμπί btnSaveMyChoice εμφανίζεται, μόνο αν ο πελάτης έχει επιλέξει ώρα παράδοσης πρώτα. Άρα, αν ο πελάτης πατήσει ένα από τα κουμπιά επιλογής ώρας παράδοσης, τότε καλείται ο DeliveryTimeController και ελέγχει ανάλογα με τον τύπο του χρήστη την διαθεσιμότητα ή μη της συγκεκριμένης ώρας. Αν πατηθεί το κουμπί btnSaveMyChoice, τότε καλείται ο CartController και προβάλλει την σελίδα CartPageGUI.

Boundary DeliveryTimeErrorMessageGUI:

«Boundary» DeliveryTimeErrorMessageGUI
-lblErrorSelectedTimeMessage: String -btnReturnToChooseDeliveryTimeGUI: String
+select(Button: String): void

Πρόκειται για το μήνυμα λάθους που εμφανίζεται, σε περίπτωση επιλογής μη διαθέσιμης ώρας παράδοσης.

Χαρακτηριστικά της κλάσης:

- lblErrorSelectedTimeMessage: Μία ετικέτα που γράφει «Συγγνώμη, δεν μπορούμε να σας εξυπηρετήσουμε την συγκεκριμένη ώρα».



- `btnReturnToChooseDeliveryTimeGUI`: Κουμπί που επιστρέφει στην σελίδα επιλογής ώρας προς παράδοση `ChooseDeliveryTimeGUI`.

Μέθοδοι της κλάσης:

- `select(Button: String)`: Εκτελείται όταν πατηθεί το κουμπί `btnReturnToChooseDeliveryTimeGUI` και καλεί τον controller `DeliveryTimeController` προκειμένου να γίνει επιστροφή στην σελίδα επιλογής ώρας παράδοσης.

Boundary `DeliveryTimeSuccessMessageGUI`:

«Boundary» DeliveryTimeSuccessMessageGUI
-lblSuccessfullySelectedTimeMessage: String -btnReturnToChooseDeliveryTimeGUI: String
+select(Button: String): void

Πρόκειται για το μήνυμα επιτυχούς επιλογής από τον χρήστη της ώρας παράδοσης της παραγγελίας του.

Χαρακτηριστικά της κλάσης:

- `lblSuccessfullySelectedTimeMessage`: Μία ετικέτα που γράφει «Επιλέξατε επιτυχώς ώρα παράδοσης της παραγγελίας σας».
- `btnReturnToChooseDeliveryTimeGUI`: Κουμπί που επιστρέφει στην σελίδα επιλογής ώρας προς παράδοση `ChooseDeliveryTimeGUI`.

Μέθοδοι της κλάσης:

- `select(Button: String)`: Εκτελείται όταν πατηθεί το κουμπί `btnReturnToChooseDeliveryTimeGUI` και καλεί τον controller `DeliveryTimeController`, προκειμένου να γίνει επιστροφή στην σελίδα επιλογής ώρας παράδοσης.

Boundary `ChoosePaymentGUI`:

«Boundary» ChoosePaymentGUI
-lblChoosePayment: String -btnContinue: String -lblPaymentList[]: String -btnCheckPaymentList[]: Button
+select(Button: String): void

Η κλάση αυτή συνιστά το γραφικό περιβάλλον διεπαφής για την επιλογή του τρόπου πληρωμής για την διεκπεραίωση της συναλλαγής.

Χαρακτηριστικά της κλάσης:

- `lblChoosePayment`: Μία ετικέτα, η οποία γράφει «Επιλογή τρόπου πληρωμής».
- `btnContinue`: Κουμπί «Συνέχεια», για την καταχώρηση της επιλογής του τρόπου πληρωμής.
- `lblPaymentList[]`: Λίστα με τις διαθέσιμες μεθόδους πληρωμής.
- `btnCheckPaymentList[]`: Μία λίστα με κουμπιά για την επιλογή τρόπου πληρωμής.



Μέθοδοι της κλάσης:

- `select(Button: String)`: Εκτελείται όταν πατηθεί κάποιο από τα παραπάνω κουμπιά. Αν πατηθεί το `btnContinue`, τότε καλείται ο `PaymentController` και οδηγεί τον χρήστη στο `PaymentProxy`. Αυτό το κουμπί εμφανίζεται όμως και μπορεί να πατηθεί μόνο αν πρώτα έχει επιλεγεί τρόπος πληρωμής, δηλαδή αν πατηθεί ένα από τα κουμπιά της λίστας `btnCheckPaymentList[]`, οπότε καλείται αυτή η συνάρτηση και στην συνέχεια καλείται ο `PaymentController`, για να ρυθμίσει τα στοιχεία πληρωμής ανάλογα με το ποιο κουμπί της λίστας πατήθηκε.

Boundary PaymentProxy:

«Boundary» PaymentProxy
+chargeAccount(orderCost: Float): void

Πρόκειται για το οριακό αντικείμενο που δρομολογεί την διαδικασία πληρωμή

Μέθοδοι της κλάσης:

- `chargeAccount(orderCost: Float)`: Ανάλογα με την μέθοδο πληρωμής που επιλέχθηκε, καλείται από τον `PaymentController` η συνάρτηση αυτή με στόχο την διεκπεραίωση της συναλλαγής, με συνολικό κόστος `orderCost`.

Entity CartEntity:

«Entity» CartEntity
-cartProductList[]: ProductEntity -numberOfCartProducts: Int -totalCartProductPrice: Float -numberOfProductKindList[]: Int -priceOfProductKindList[]: Float
+setCartProductList(cartProductList[]: ProductEntity): void +getCartProductList(): cartProductList +setNumberOfCartProducts(numberOfCartProducts: Int): void +getNumberOfCartProducts(): Int +setTotalCartProductPrice(totalCartProductPrice: Float): void +getTotalCartProductPrice(): Float +setNumberOfProductKindList(numberOfProductKindList[]: Int): void +getNumberOfProductKindList(): numberOfProductKindList +setPriceOfProductKindList(priceOfProductKindList[]: Float): void +getPriceOfProductKindList(): priceOfProductKindList

Η κλάση αυτή ορίζει την οντότητα που συνιστά το καλάθι αγορών, στο οποίο ο χρήστης προσθέτει ή αφαιρεί προϊόντα ανάλογα με το τι επιθυμεί να αγοράσει.

Χαρακτηριστικά της κλάσης:

- `cartProductList[]`: Μία λίστα με τα προϊόντα τύπου `ProductEntity` που υπάρχουν στο καλάθι αγορών και έχουν επιλεγεί από τον πελάτη για να τα αγοράσει.
- `numberOfCartProducts`: Ο συνολικός αριθμός προϊόντων που υπάρχουν στο καλάθι αγορών.
- `totalCartProductPrice`: Η συνολική τιμή των προϊόντων που έχουν προστεθεί από τον πελάτη στο καλάθι αγορών.



- `numberOfProductKindList[]`: Μία λίστα με τους αριθμούς συσκευασιών για το κάθε είδος προϊόντος που έχει προσθέσει ο πελάτης στο καλάθι αγορών.
- `priceOfProductKindList[]`: Μία λίστα με τις τιμές για τον συνολικό αριθμό συσκευασιών του κάθε είδους προϊόντος που έχει προσθέσει ο πελάτης στο καλάθι αγορών.

Μέθοδοι της κλάσης:

- `setCartProductList(cartProductList[]: ProductEntity)`: Θέτει την λίστα των προϊόντων `cartProductList[]` που θα βρίσκονται μέσα στο καλάθι αγορών.
- `getCartProductList()`: Επιστρέφει την λίστα των προϊόντων που βρίσκονται στο καλάθι αγορών την δεδομένη στιγμή.
- `setNumberOfCartProducts(numberOfCartProducts: Int)`: Θέτει τον συνολικό αριθμό προϊόντων που υπάρχουν στο καλάθι αγορών.
- `getNumberOfCartProducts()`: Επιστρέφει τον συνολικό αριθμό προϊόντων που υπάρχουν στο καλάθι αγορών.
- `setTotalCartProductPrice(totalCartProductPrice: Float)`: Θέτει την συνολική τιμή των προϊόντων που έχουν προστεθεί από τον πελάτη στο καλάθι αγορών.
- `getTotalCartProductPrice()`: Επιστρέφει την συνολική τιμή των προϊόντων που έχουν προστεθεί από τον πελάτη στο καλάθι αγορών.
- `setNumberOfProductKindList(numberOfProductKindList[]: Int)`: Θέτει την λίστα με τους αριθμούς συσκευασιών για το κάθε είδος προϊόντος που έχει προσθέσει ο πελάτης στο καλάθι αγορών.
- `getNumberOfProductKindList()`: Επιστρέφει την λίστα με τους αριθμούς συσκευασιών για το κάθε είδος προϊόντος που έχει προσθέσει ο πελάτης στο καλάθι αγορών.
- `setPriceOfProductKindList(priceOfProductKindList[]: Float)`: Θέτει την λίστα με τις τιμές για τον συνολικό αριθμό συσκευασιών του κάθε είδους προϊόντος που έχει προσθέσει ο πελάτης στο καλάθι αγορών.
- `getPriceOfProductKindList()`: Επιστρέφει την λίστα με τις τιμές για τον συνολικό αριθμό συσκευασιών του κάθε είδους προϊόντος που έχει προσθέσει ο πελάτης στο καλάθι αγορών.

Entity PaymentEntity:

«Entity» PaymentEntity
-paymentMethod: Int
+setPaymentMethod(paymentMethod: Int): void +getPaymentMethod(): Int

Η κλάση αυτή ορίζει την οντότητα που αποτελεί τον τρόπο πληρωμής, με τον οποίο διεκπεραιώνεται η συναλλαγή.

Χαρακτηριστικά της κλάσης:

- `paymentMethod`: Ακέραιος αριθμός που προσδιορίζει τον τρόπο πληρωμής με τον οποίο θα διεκπεραιωθεί η συναλλαγή. Πιο συγκεκριμένα, το 0 αντιστοιχεί στην αντικαταβολή, το 1 στο Paypal και το 2 στην πιστωτική κάρτα.

Μέθοδοι της κλάσης:



- `setPaymentMethod(paymentMethod: Int)`: Θέτει την τιμή του ακεραίου αριθμού `paymentMethod` που προσδιορίζει τον τρόπο πληρωμής.
- `getPaymentMethod()`: Επιστρέφει την τιμή του ακεραίου αριθμού `paymentMethod` που προσδιορίζει τον τρόπο πληρωμής.

Entity DeliveryTimeEntity:

«Entity» DeliveryTimeEntity
-deliveryTimeSimpleUser: Int -deliveryTimeProfessionalUser: Int
+setDeliveryTimeSimpleUser(deliveryTimeSimpleUser: Int): void +getDeliveryTimeSimpleUser(): Int +setDeliveryTimeProfessionalUser(deliveryTimeProfessionalUser: Int): void +getDeliveryTimeProfessionalUser(): Int

Η κλάση αυτή ορίζει την οντότητα που αποτελεί την επιλογή του χρονικού διαστήματος παράδοσης της παραγγελίας, είτε για απλό είτε για επαγγελματία χρήστη.

Χαρακτηριστικά της κλάσης:

- `deliveryTimeSimpleUser`: Ακέραιος αριθμός που προσδιορίζει την ώρα παράδοσης (για παράδειγμα 16:00-17:00), με βάση την λίστα με τις διαθέσιμες ώρες παράδοσης για απλό χρήστη.
- `deliveryTimeProfessionalUser`: Ακέραιος αριθμός που προσδιορίζει την ώρα παράδοσης (για παράδειγμα 19:00-20:00), με βάση την λίστα με τις διαθέσιμες ώρες παράδοσης για επαγγελματία χρήστη.

Μέθοδοι της κλάσης:

- `setDeliveryTimeSimpleUser(deliveryTimeSimpleUser: Int)`: Θέτει τον αριθμό που προσδιορίζει την ώρα παράδοσης για την περίπτωση του απλού χρήστη.
- `getDeliveryTimeSimpleUser()`: Επιστρέφει τον αριθμό που προσδιορίζει την ώρα παράδοσης για την περίπτωση του απλού χρήστη.
- `setDeliveryTimeProfessionalUser(deliveryTimeProfessionalUser: Int)`: Θέτει τον αριθμό που προσδιορίζει την ώρα παράδοσης για την περίπτωση του επαγγελματία χρήστη.
- `getDeliveryTimeProfessionalUser()`: Επιστρέφει τον αριθμό που προσδιορίζει την ώρα παράδοσης για την περίπτωση του επαγγελματία χρήστη.

Controller CartController:

«Controller» CartController
+addProductToCart(productId: Int, quantity: Int): void +checkAddingQuantityAvailable(productId: Int, quantity: Int, availableQuantity: Int): boolean +removeProductFromCart(productId: Int, quantity: Int): void +checkRemovingQuantity(removingQuantity: Int, cartQuantity: Int): boolean +displayCartPageGUI(productList[]: ProductEntity): void +displayCartErrorMessageGUI(): void +displayRemoveFromCartGUI(productId: Int): void +displayChooseDeliveryTimeGUI(typeOfUser: Int): void +displayChoosePaymentGUI(): void +returnToProductPage(productId: Int): void

Πρόκειται για τον ελεγκτή που χρησιμοποιείται στο καλάθι αγορών, προκειμένου να δρομολογούνται οι ενέργειες του χρήστη σχετικά με την προσθαφαίρεση προϊόντων σε αυτό ή με την προβολή σελίδων στις οποίες εκείνος επιθυμεί να μεταβεί.

Μέθοδοι της κλάσης:



- `addProductToCart(productID: Int, quantity: Int)`: Καλείται όταν ο χρήστης πατήσει «Προσθήκη στο καλάθι» και προσθέτει στην λίστα των προϊόντων που υπάρχουν στο καλάθι αγορών το συγκεκριμένο προϊόν με βάση το αναγνωριστικό του, δηλαδή το `productID`, στην δοσμένη ποσότητα `quantity`, αφού πρώτα έχει ελεγχθεί ότι η ζητούμενη ποσότητα είναι διαθέσιμη.
- `checkAddingQuantityAvailable(productID: Int, quantity: Int, availableQuantity: Int)`: Ελέγχει εάν το ζητούμενο προϊόν με αναγνωριστικό `productID` βρίσκεται σε διαθέσιμη ποσότητα `availableQuantity` που είναι επαρκής σχετικά με την ζητούμενη ποσότητα `quantity` και επιστρέφει `true` ή `false` αναλόγως.
- `removeProductFromCart(productID: Int, quantity: Int)`: Αφαιρεί ποσότητα `quantity` από το προϊόν με αναγνωριστικό προϊόντος `productID`, αφού πρώτα έχει ελεγχθεί ότι η αφαιρούμενη ποσότητα `quantity` είναι μικρότερη ή ίση με την ποσότητα του προϊόντος που βρίσκεται την δεδομένη στιγμή στο καλάθι αγορών.
- `checkRemovingQuantity(removingQuantity: Int, cartQuantity: Int)`: Ελέγχει εάν η αφαιρούμενη ποσότητα `removingQuantity` είναι μικρότερη ή ίση με την ποσότητα του προϊόντος `cartQuantity` που βρίσκεται την δεδομένη στιγμή στο καλάθι αγορών και επιστρέφει `true` ή `false` αναλόγως.
- `displayCartPageGUI(productList[:]: ProductEntity)`: Προβάλλει την σελίδα που αφορά το καλάθι αγορών με την λίστα των προϊόντων `productList[]` που αυτό περιλαμβάνει, όταν πατηθεί ο σύνδεσμος «Καλάθι αγορών».
- `displayCartErrorMessageGUI()`: Προβάλλει την σελίδα `CartErrorMessageGUI`, είτε όταν ο χρήστης προσπαθήσει να προσθέσει στο καλάθι προϊόν με ποσότητα που υπερβαίνει την διαθέσιμη, είτε όταν προσπαθήσει να αφαιρέσει ποσότητα προϊόντος παραπάνω από αυτήν που βρίσκεται στο καλάθι.
- `displayRemoveFromCartGUI(productID: Int)`: Προβάλλει την σελίδα `RemoveFromCartGUI`, για να μειώσει ο χρήστης την ποσότητα που επιθυμεί για το προϊόν με αναγνωριστικό `productID`, όταν μέσα στο καλάθι αγορών πατήσει δίπλα από αυτό το προϊόν τον σύνδεσμο «Αφαίρεση».
- `displayChooseDeliveryTimeGUI(typeOfUser: Int)`: Προβάλλει την σελίδα `ChooseDeliveryTimeGUI`, τέτοια ώστε να ανταποκρίνεται στον τύπο του χρήστη `typeOfUser`, εάν είναι δηλαδή απλός ή επαγγελματίας.
- `displayChoosePaymentGUI()`: Προβάλλει την σελίδα `ChoosePaymentGUI`, στην οποία ο χρήστης επιλέγει τον τρόπο πληρωμής για την διεκπεραίωση της συναλλαγής, αφού πατηθεί ο σύνδεσμος «Τρόπος πληρωμής» από τον χρήστη στην σελίδα `CartPageGUI`.
- `returnToProductPage(productID: Int)`: Επιστρέφει στην σελίδα του προϊόντος με αναγνωριστικό `productID`.

PaymentController:

«Controller» PaymentController
+checkPaymentButtonPressed(Button: String): Int +displayPaymentProxyPage(paymentMethod: Int): void +returnToCartPageGUI(): void +payWithPayPal(orderCost: Float): void +payWithCreditCard(orderCost: Float): void +payWithCash(orderCost: Float): void

Η κλάση αυτή ορίζει τον ελεγκτή που ρυθμίζει τις ενέργειες του χρήστη για την επιλογή του τρόπου πληρωμής της παραγγελίας του.

Μέθοδοι της κλάσης:



- `checkPaymentButtonPressed(Button: String)`: Ελέγχει ποιο κουμπί πατήθηκε από την σελίδα `ChoosePaymentGUI` και αναλόγως επιστρέφει έναν ακέραιο αριθμό από το 0 έως και το 2, αν πρόκειται για αντικαταβολή, Paypal ή πιστωτική κάρτα αντίστοιχα.
- `displayPaymentProxyPage(paymentMethod: Int)`: Προβάλλει την σελίδα στην οποία πραγματοποιείται η διενέργεια της συναλλαγής ανάλογα με την μέθοδο `paymentMethod` που επιλέχθηκε.
- `returnToCartPageGUI()`: Επιστρέφει στην σελίδα του καλαθιού αγορών `CartPageGUI`.
- `payWithPayPal(orderCost: Float)`: Διενέργεια συναλλαγής μέσω του γραφικού περιβάλλοντος της `PayPal`.
- `payWithCreditCard(orderCost: Float)`: Διενέργεια τραπεζικής συναλλαγής μέσω πιστωτικής κάρτας.
- `payWithCash(orderCost: Float)`: Καταχώρηση στο σύστημα ότι η πληρωμή θα γίνει την ώρα της παράδοσης με μετρητά.

Controller `DeliveryTimeController`:

«Controller» <code>DeliveryTimeController</code>
<code>+checkSelectedDeliveryTimeSimpleUser(Button: String): Int</code> <code>+checkSelectedDeliveryTimeProfessionalUser(Button: String): Int</code> <code>+displayDeliveryTimeErrorMessageGUI(): void</code> <code>+displayDeliveryTimeSuccessMessageGUI(): void</code> <code>+displayChooseDeliveryTimeGUI(): void</code>

Πρόκειται για τον ελεγκτή, ο οποίος ανάλογα με την επιλογή του χρήστη για την ώρα παράδοσης της παραγγελίας του, τσεκάρει για την διαθεσιμότητά της και οδηγεί στην εκτύπωση αντίστοιχου μηνύματος επιτυχίας ή αποτυχίας επιλογής.

Μέθοδοι της κλάσης:

- `checkSelectedDeliveryTimeSimpleUser(Button: String)`: Ελέγχει ποιο κουμπί πατήθηκε από την σελίδα `ChooseDeliveryTimeGUI` για απλό χρήστη και αναλόγως επιστρέφει έναν ακέραιο αριθμό που δηλώνει την ώρα που επιλέχθηκε.
- `checkSelectedDeliveryTimeProfessionalUser(Button: String)`: Ελέγχει ποιο κουμπί πατήθηκε από την σελίδα `ChooseDeliveryTimeGUI` για επαγγελματία χρήστη και αναλόγως επιστρέφει έναν ακέραιο αριθμό που δηλώνει την ώρα που επιλέχθηκε.
- `displayDeliveryTimeErrorMessageGUI()`: Προβάλλει την σελίδα `DeliveryTimeErrorMessageGUI`, εάν ο χρήστης, απλός ή επαγγελματίας, επιλέξει μία ώρα, κατά την οποία το σύστημα δεν μπορεί να τον εξυπηρετήσει λόγω άλλων παραγγελιών που έχουν ήδη δρομολογηθεί να παραδοθούν την συγκεκριμένη ώρα.
- `displayDeliveryTimeSuccessMessageGUI()`: Προβάλλει την σελίδα `DeliveryTimeSuccessMessageGUI`, η οποία διαβεβαιώνει τον χρήστη ότι η ώρα παράδοσης επιλέχθηκε επιτυχώς.
- `displayChooseDeliveryTimeGUI()`: Προβάλλει την σελίδα `ChooseDeliveryTimeGUI`, όταν πατηθεί το κατάλληλο κουμπί από τα περιβάλλοντα `DeliveryTimeErrorMessageGUI` ή `DeliveryTimeSuccessMessageGUI`.



2.2.2 Διάγραμμα κλάσεων

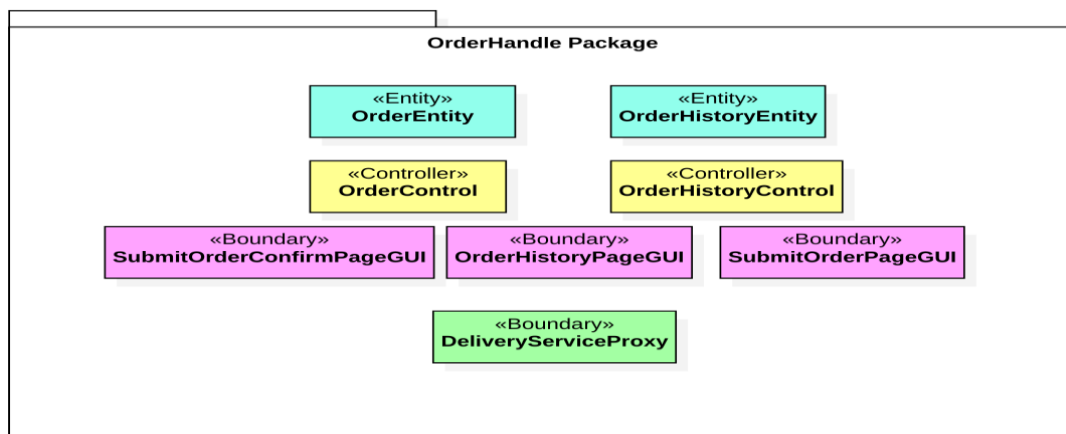


Σχήμα 3. Διάγραμμα κλάσεων πακέτου CartHandle.



2.2.3 Πακέτο OrderHandle

Το πακέτο αυτό αναφέρεται στην συνολική διαχείριση της παραγγελίας. Περιλαμβάνει την σελίδα της τελικής υποβολής παραγγελίας και του ιστορικού παραγγελιών, καθώς και συνδέεται με την εξωτερική υπηρεσία Delivery.



Boundary SubmitOrderPageGUI :

«Boundary» SubmitOrderPageGUI
-lblPageName: String -lblCartProductList[]: String -lblCartProductPriceList[]: float -lblDeliveryCost: String -lblTotalCost: String -lblTotalCostValue: float -lblDeliveryCostValue: float -lblDeliveryTime: String -lblDeliveryTimeValue: String +btnSubmitOrder: Button
+SubmitOrderPageGUI(): void +btnSelect(Button: String): void

Η κλάση αυτή είναι η σελίδα της τελικής υποβολής παραγγελίας. Ο χρήστης μεταφέρεται εδώ πατώντας τον σύνδεσμο «Submit Order» στην σελίδα του καλαθιού.

Χαρακτηριστικά της κλάσης :

- lblPageName : Επιγραφή που εμφανίζει την συμβολοσειρά "Υποβολή Παραγγελίας".
- lblCartProductList[] : Λίστα επιγραφών που παρουσιάζει τα ονόματα των προϊόντων προς παραγγελία.
- lblCartProductPriceList[] : Λίστα επιγραφών που παρουσιάζει τις τιμές των προϊόντων προς παραγγελία.
- lblDeliveryCost : Επιγραφή που εμφανίζει την συμβολοσειρά "Κόστος Μεταφορικών".
- lblTotalCost : Επιγραφή που εμφανίζει την συμβολοσειρά "Συνολικό Κόστος".
- lblTotalCostValue : Επιγραφή η οποία παρουσιάζει την συνολική τιμή της παραγγελίας.
- lblDeliveryCostValue : Επιγραφή η οποία παρουσιάζει το κόστος των μεταφορικών.



- lblDeliveryTime : Επιγραφή που εμφανίζει την συμβολοσειρά “Ωρα Παράδοσης”.
- lblDeliveryTimeValue : Επιγραφή η οποία παρουσιάζει την ώρα παράδοσης που επιλέχθηκε.
- btnSubmitOrder : Κουμπί για την ολοκλήρωση της παραγγελίας.

Μέθοδοι της κλάσης :

- SubmitOrderPageGUI() : Η συνάρτηση αυτή αποτελεί τον «constructor» της κλάσης αυτής και πρέπει να δέχεται σαν όρισμα όλα τα χαρακτηριστικά που αναφέρθηκαν παραπάνω .
- btnSelect(Button : String) : Η μέθοδος αυτή καλεί την συνάρτηση displayConfirmationMsg () του ελεγκτή OrderController για επιβεβαίωση της υποβολής τελικής παραγγελίας.

Boundary OrderHistoryPageGUI :

«Boundary» OrderHistoryPageGUI
-lblPageName: String -lblOrderIDList[]: int -lblOrderDateList[]: String -lblOrderTotalPriceList[]: float -btnScrollUp: Button -btnScrollDown: Button -btnHomePage: Button
+OrderHistoryPageGUI(): void +btnSelect(Button: Image): void

Η κλάση αυτή είναι η σελίδα του ιστορικού παραγγελιών . Ο χρήστης μεταφέρεται εδώ πατώντας τον σύνδεσμο «Ιστορικό παραγγελιών» στην αρχική σελίδα .

Χαρακτηριστικά της κλάσης :

- lblPageName : Επιγραφή που εμφανίζει την συμβολοσειρά “Ιστορικό Παραγγελιών”.
- lblOrderIDList[] : Λίστα επιγραφών που παρουσιάζει το «ID» των τελευταίων παραγγελιών που έχουν πραγματοποιηθεί .
- lblOrderDateList[] : Λίστα επιγραφών που παρουσιάζει τις ημερομηνίες των τελευταίων παραγγελιών που έχουν πραγματοποιηθεί .
- lblOrderTotalPriceList[] : Λίστα επιγραφών που παρουσιάζει την συνολική τιμή των τελευταίων παραγγελιών που έχουν πραγματοποιηθεί .
- btnScrollUp : Κουμπί για ολίσθηση της σελίδας προς τα πάνω.
- btnScrollDown : Κουμπί για ολίσθηση της σελίδας προς τα κάτω.
- btnHomePage : Κουμπί για επιστροφή στην αρχική σελίδα της εφαρμογής.

Μέθοδοι της κλάσης :

- OrderHistoryPageGUI() : Η συνάρτηση αυτή αποτελεί τον «constructor» της κλάσης αυτής και πρέπει να δέχεται σαν όρισμα όλα τα χαρακτηριστικά που αναφέρθηκαν παραπάνω .
- btnSelect(Button : Image) : Η συνάρτηση αυτή ελέγχει ποιο απο τα τρία πιθανά κουμπιά της σελίδας έχει επιλεγθεί και πράττει ανάλογα. Αν επιλεγθεί το «btnScrollUp» , καλείται η συνάρτηση scrollUp() του ελεγκτή OrderHistoryController , ομοίως αν επιλεγθεί το «btnScrollDown» καλείται η συνάρτηση scrollDown() του ελεγκτή OrderHistoryController και τέλος με την επιλογή του «btnHomePage» καλείται η συνάρτηση returnToHomePage() του ελεγκτή OrderHistoryController .

**Boundary SubmitOrderConfirmPageGUI :**

SubmitOrderConfirmPageGUI
+lblConfirmation: String +btnYes: Button +btnNo: Button
+SubmitOrderComfirmPageGUI(label: String, btnY: Button, btnN: Button) +btnSelect(Button: String): void

Η κλάση αυτή είναι η σελίδα επιβεβαίωσης για την οριστική υποβολή της τελικής παραγγελίας . Ο χρήστης μεταφέρεται εδώ πατώντας τον σύνδεσμο «Submit Order» στη σελίδα της τελικής υποβολής παραγγελίας.

Χαρακτηριστικά της κλάσης :

- lblConfirmation : Επιγραφή με το όνομα της συγκεκριμένης σελίδας.
- btnYes : Κουμπί επιβεβαίωσης .
- btnNo : Κουμπί ακύρωσης .

Μέθοδοι της κλάσης :

- SubmitOrderComfirmPageGUI (label : String , btnY : Button , btnN : Button) : Η συνάρτηση αυτή αποτελεί τον «constructor» της κλάσης αυτής και πρέπει να δέχεται σαν όρισμα όλα τα χαρακτηριστικά που αναφέρθηκαν παραπάνω .
- btnSelect(Button : String) : Η συνάρτηση αυτή ελέγχει ποια απο τα δύο πιθανά κουμπιά της σελίδας έχει επιλεγεί και πράττει ανάλογα . Με την επιλογή του «btnYes» , αρχικά καλείται η συνάρτηση informOrderHistory() του ελεγκτή OrderController και έπειτα καλείται η συνάρτηση sendOrder(delivTime:DeliveryTimeEntity, cart : CartEntity, pay : PaymentEntity, user : UserEntity, order : OrderEntity) του ελεγκτή OrderController . Διαφορετικά καλείται η συνάρτηση stayOnSubmitOrderPage() του ελεγκτή OrderController.

Boundary DeliveryServiceProxy :

«Boundary» DeliveryServiceProxy
+completeOrder(delivTime: DeliveryTimeEntity, cart: CartEntity, pay: PaymentEntity, user: UserEntity, order: OrderEntity): void

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με την υπηρεσία «Delivery» , ώστε να αποστέλλεται η τελική παραγγελία στην υπηρεσία αυτή για να την ολοκληρώσει.

Μέθοδοι της κλάσης :

- completeOrder(delivTime : DeliveryTimeEntity , cart : CartEntity, pay : PaymentEntity, user : UserEntity, order : OrderEntity) : Η συνάρτηση αυτή αποστέλλει όλες τις σημαντικές πληροφορίες που δέχεται ως ορίσματα , στην υπηρεσία «Delivery» , για την προετοιμασία και μεταφορά της παραγγελίας στον επιλεγμένο χώρο του χρήστη. Πιο συγκεκριμένα αποστέλλονται οι εξής πληροφορίες : Η ώρα παράδοσης που έχει επιλεγθεί , το σύνολο των προϊόντων που επιθυμεί να αγοράσει ο πελάτης , τα οποία βρίσκονται στην οντότητα του καλαθιού , τον τρόπο πληρωμής που έχει επιλεγθεί , τα στοιχεία του χρήστη , ώστε να γνωρίζει η υπηρεσία τον χώρο παράδοσης , ονοματεπώνυμο κτλ , και τέλος τον χαρακτηριστικό αριθμό της κάθε παραγγελίας ώστε να υπάρχει εύκολη διαχείριση των παραγγελιών από την υπηρεσία .



Entity OrderEntity :

«Entity» OrderEntity
-orderId: int -orderDate: String -deliveryCost: float -prodList []: ProductEntity -totalCost: float
+OrderEntity(id: int, date: String, delivCost: float, prodList []: ProductEntity, ttlCost: float): void +setOrderID(id: int): void +setOrderDate(date: String): void +setDeliveryCost(delivCost: float): void +setProdList(prodList []: ProductEntity): void +setTotalCost(total: float): void +getOrderID(): int +getOrderDate(): String +getDeliveryCost(): float +getProdList(): ProductEntity +getTotalCost(): float

Αυτή είναι η κλάση οντότητας της παραγγελίας , η οποία περιέχει τα χαρακτηριστικά της για την δημιουργία της.

Χαρακτηριστικά της κλάσης :

- orderId : Ο μοναδικός χαρακτηριστικός αριθμός για την κάθε παραγγελία .
- orderDate: Η ημερομηνία πραγματοποίησης της παραγγελίας .
- deliveryCost : Κόστος μεταφορικών της παραγγελίας .
- prodList [] : Λίστα των προϊόντων προς παραγγελία.
- totalCost : Το συνολικό κόστος της παραγγελίας.

Μέθοδοι της κλάσης :

- OrderEntity(id : int , date : String , delivCost : float , prodList [] : ProductEntity): Η συνάρτηση αυτή αποτελεί τον «constructor» της οντότητας OrderEntity.
- setOrderID(id : int): Η συνάρτηση αυτή δίνει την τιμή του ορίσματος στον μοναδικό χαρακτηριστικό αριθμό της παραγγελίας .
- setOrderDate(date : String): Η συνάρτηση αυτή δίνει την τιμή του ορίσματος στην ημερομηνία πραγματοποίησης της παραγγελίας .
- setDeliveryCost(delivCost : float): Η συνάρτηση αυτή δίνει την τιμή του ορίσματος στο κόστος μεταφορικών της παραγγελίας .
- setProdList (prodList [] : ProductEntity) : Η συνάρτηση αυτή δίνει τις τιμές της λίστα του ορίσματος στη λίστα των προϊόντων προς παραγγελία .
- setTotalCost (total : float) : Η συνάρτηση αυτή δίνει την τιμή του ορίσματος στη μεταβλητή που αναφέρεται στο συνολικό κόστος της παραγγελίας.
- getOrderID(): Η συνάρτηση αυτή επιστρέφει τον μοναδικό χαρακτηριστικό αριθμό της παραγγελίας .
- getOrderDate(): Η συνάρτηση αυτή επιστρέφει την ημερομηνία πραγματοποίησης της παραγγελίας.
- getDeliveryCost() : Η συνάρτηση αυτή επιστρέφει το κόστος μεταφορικών της παραγγελίας.
- getProdList() : Η συνάρτηση αυτή επιστρέφει την λίστα των προϊόντων προς παραγγελία.
- getTotalCost() : Η συνάρτηση αυτή επιστρέφει το συνολικό κόστος της παραγγελίας.

**Entity OrderHistoryEntity :**

«Entity» OrderHistoryEntity
-orderHistoryList[]: OrderEntity
+OrderHistoryEntity(historyList[]: OrderEntity): void +setOrderHistoryList(historyList[]: OrderEntity): void +getOrderHistoryList(): OrderEntity

Αυτή είναι η κλάση οντότητας του ιστορικού παραγγελιών , η οποία περιέχει μια λίστα παραγγελιών για την δημιουργία του ιστορικού.

Χαρακτηριστικά της κλάσης :

- orderHistoryList[] : Η λίστα των τελευταίων παραγγελιών προς εμφάνιση.

Μέθοδοι της κλάσης :

- OrderHistoryEntity(historyList[] : OrderEntity) : Η συνάρτηση αυτή αποτελεί τον «constructor» της οντότητας OrderHistoryEntity .
- setOrderHistoryList(historyList[] : OrderEntity) : Η συνάρτηση αυτή δίνει τις τιμές της λίστα του ορίσματος στή λίστα των παραγγελιών προς εμφάνιση.
- getOrderHistoryList() : Η συνάρτηση αυτή επιστρέφει την λίστα των τελευταίων παραγγελιών προς εμφάνιση.

Controller OrderController :

«Controller» OrderController
+createOrder(cart: CartEntity, delivTime: DeliveryTimeEntity): void +displaySubmitOrderPageGUI(cart: CartEntity, delivTime: DeliveryTimeEntity, order: OrderEntity): void +calculateDeliveryCost(user: UserEntity, cart: CartEntity, order: OrderEntity): void +sendOrder(delivTime: DeliveryTimeEntity, cart: CartEntity, pay: PaymentEntity, user: UserEntity, order: OrderEntity): void +informOrderHistory(order: OrderEntity): void +displayConfirmationMsg(): void +stayOnSubmitOrderPage(): void +returnToHomePage(): void

Η κλάση αυτή αποτελεί τον ελεγκτή της παραγγελίας , ο οποίος περιέχει όλες τις κατάλληλες συναρτήσεις για την επεξεργασία και αποστολή της παραγγελίας .

Μέθοδοι της κλάσης :

- createOrder(cart : CartEntity , delivTime : DeliveryTimeEntity) : Η συνάρτηση αυτή δέχεται ως όρισμα την οντότητα του καλαθιού , η οποία περιέχει όλα τα επιθυμητά προϊόντα και τις τιμές τους , καθώς και την οντότητα DeliveryTimeEntity , η οποία περιέχει την πληροφορία για την ώρα παράδοσης . Με την κλήση αυτής της συνάρτησης δημιουργείται η οντότητα OrderEntity μέσω του «constructor» της .
- displaySubmitOrderPageGUI(cart : CartEntity, delivTime : DeliveryTimeEntity, order : OrderEntity): Η συνάρτηση αυτή δέχεται ως όρισμα την οντότητα του καλαθιού η οποία περιέχει τα επιλεγθέντα προϊόντα , την οντότητα του χρόνου παράδοσης και την οντότητα της παραγγελίας που περιέχει το κόστος μεταφοράς , ώστε να πραγματοποιηθεί η δημιουργία και η εμφάνιση της σελίδας



- SubmitOrderPageGUI μέσω του «constructor» της .
- calculateDeliveryCost(user : UserEntity, cart : CartEntity, order : OrderEntity): Η συνάρτηση αυτή δέχεται ως όρισμα τα στοιχεία του χρήστη, ώστε να γνωρίζει που ακριβώς πρέπει να μεταφερθεί η παραγγελία, τα προϊόντα που έχουν προστεθεί στο καλάθι, ώστε να μπορεί να υπολογιστεί το βάρος της παραγγελίας, και τα στοιχεία της παραγγελίας (orderId) ώστε να γνωρίζει για ποιά ακριβώς παραγγελία πρόκειται. Σύμφωνα με τα παραπάνω στοιχεία υπολογίζεται το κόστος μεταφοράς και ενημερώνεται η αντίστοιχη μεταβλητή της οντότητας OrderEntity.
 - sendOrder(delivTime : DeliveryTimeEntity, cart : CartEntity, pay : PaymentEntity, user : UserEntity, order : OrderEntity): Η συνάρτηση αυτή μεταφέρει όλη την απαραίτητη πληροφορία για την παραγγελία στην υπηρεσία Delivery, καλώντας της συνάρτηση completeOrder(delivTime : DeliveryTimeEntity, cart : CartEntity, pay : PaymentEntity, user : UserEntity, order : OrderEntity) του εξωτερικού συστήματος DeliveryServiceProxy.
 - informOrderHistory(order : OrderEntity) : Η συνάρτηση αυτή δέχεται ως όρισμα την οντότητα της παραγγελίας που δημιουργήθηκε με την κλήση της createOrder(cart : CartEntity, delivTime : DeliveryTimeEntity) του ελεγκτή OrderController και ενημερώνει την λίστα των παραγγελιών της οντότητας του ιστορικού παραγγελιών(OrderHistoryEntity).
 - displayConfirmationMsg(): Η συνάρτηση αυτή εμφανίζει την σελίδα SubmitOrderConfirmPageGUI.
 - stayOnSubmitOrderPage(): Η συνάρτηση αυτή ακυρώνει την υποβολή της τελικής παραγγελίας και επιστρέφει τον χρήστη στην σελίδα SubmitOrderPageGUI στην κατάσταση ακριβώς πριν πατήσει το κουμπί «SubmitOrder».
 - returnToHomePage(): Η συνάρτηση αυτή επιστρέφει τον χρήστη στην αρχική σελίδα (HomePageGUI).

Controller OrderHistoryController :

«Controller» OrderHistoryController
+displayOrderHistoryPage(orderHistoryList[]: OrderEntity): void +scrollUp(): void +scrollDown(): void +returnToHomePage(): void

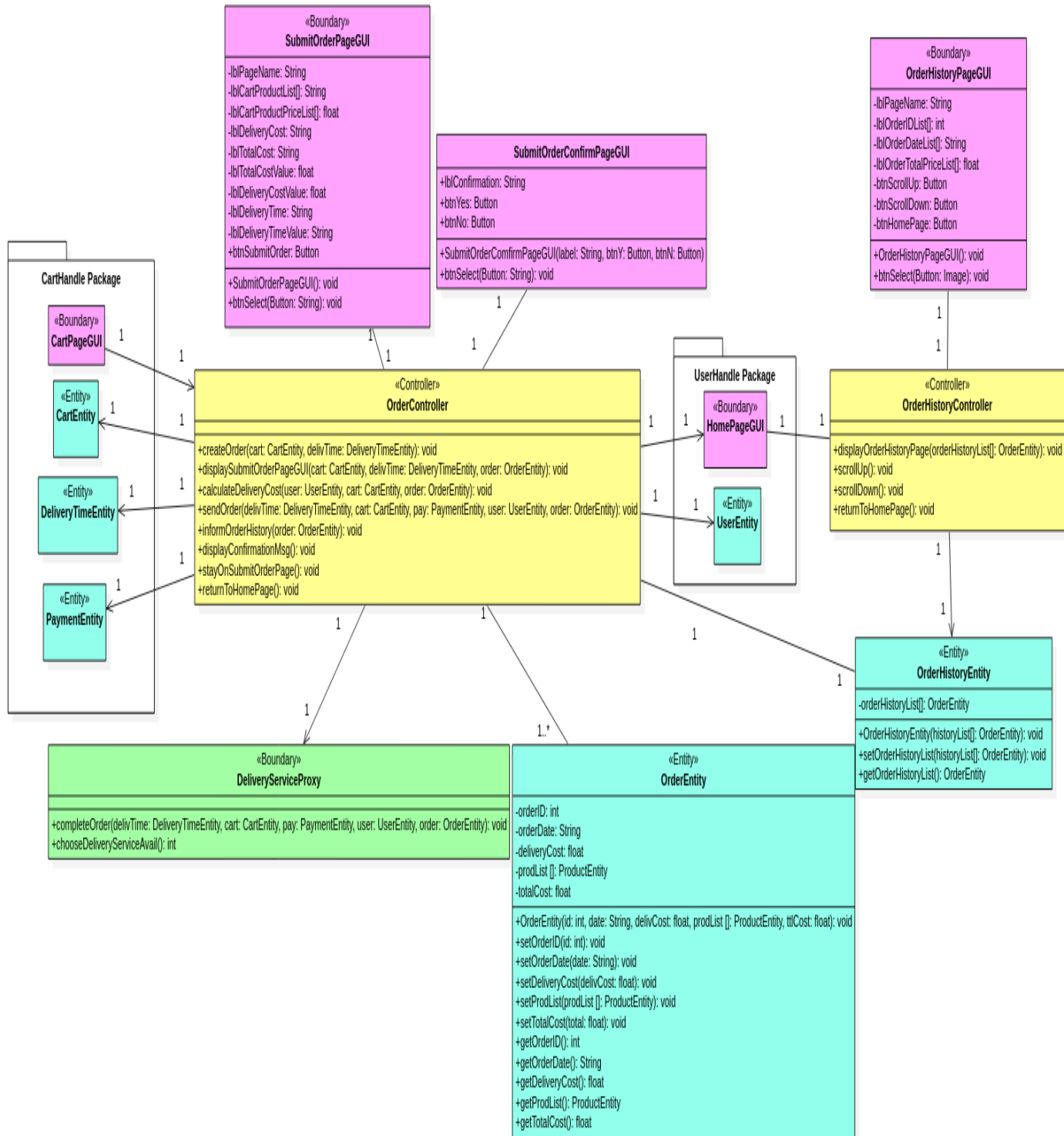
Η κλάση αυτή αποτελεί τον ελεγκτή του ιστορικού παραγγελιών και περιέχει τις κατάλληλες μεθόδους για την προβολή του ιστορικού παραγγελιών.

Μέθοδοι της κλάσης :

- displayOrderHistoryPage(orderHistoryList[] : OrderEntity) : Η συνάρτηση αυτή δέχεται ως όρισμα την λίστα με τις τελευταίες παραγγελίες προς εμφάνιση, και δημιουργεί και εμφανίζει την σελίδα του ιστορικού παραγγελιών (OrderHistoryPageGUI) μέσω του «constructor» της.
- scrollUp() : Η συνάρτηση αυτή ολισθαίνει προς τα πάνω στην σελίδα που εμφανίζεται εκείνη την στιγμή.
- scrollDown() : Η συνάρτηση αυτή ολισθαίνει προς τα κάτω στην σελίδα που εμφανίζεται εκείνη την στιγμή.
- returnToHomePage() : Η συνάρτηση αυτή επιστρέφει τον χρήστη στην αρχική σελίδα (HomePageGUI).



2.2.4 Διαγράμματα κλάσεων



Σχήμα 4. Διάγραμμα κλάσεων πακέτου OrderHandle.



3 Μη λειτουργικές απαιτήσεις

3.1 Απαιτήσεις επίδοσης

<ΜΛΑ- 2>

Το σύστημα πρέπει να έχει χρόνο απόκρισης μικρότερο από 200 ms.

Περιγραφή: Κάθε ενέργεια της εφαρμογής θα πρέπει να πραγματοποιείται σε χρόνο μικρότερο των 200 ms , ώστε η εφαρμογή να μην είναι κουραστική για τον χρήστη .

User Priority (4/5) : Ο χρόνος απόκρισης μιας εφαρμογής είναι υψηλής σημασίας για το μέσο χρήστη, καθώς κάθε χρήστης επιθυμεί μία γρήγορη εφαρμογή.

Technical Priority (5/5): Για το σύστημα ο χρόνος απόκρισης είναι πολύ σημαντικό χαρακτηριστικό. Καθώς ο χρόνος απόκρισης εξαρτάται αφενός από την ταχύτητα αποστολής και παραλαβής δεδομένων από το σύστημα και αφετέρου από την ταχύτητα επεξεργασίας στην κεντρική μονάδα. Είναι μεγάλης σημασίας το σύστημα να είναι εκσυγχρονισμένο και επιλεγμένο κατάλληλα για να ανταπεξέλθει στις υπολογιστικές απαιτήσεις του συστήματος.

Stability: (3/5) Ο αναμενόμενος χρόνος απόκρισης του συστήματος ενδέχεται να αλλάξει σύμφωνα με τις νέες απαιτήσεις των χρηστών καθώς και την εξέλιξη της υπολογιστικής ισχύς των σύγχρονων συστημάτων.

3.2 Απαιτήσεις Χρηστικότητα (Usability)

<ΜΛΑ- 4>

Το σύστημα πρέπει να υποστηρίζει Ελληνικά και Αγγλικά

Περιγραφή: Η υποστήριξη Αγγλικών από το σύστημα επιτρέπει την χρήση του από ανθρώπους που δεν γνωρίζουν ελληνικά και θέλουν να χρησιμοποιήσουν την εφαρμογή. Επίσης κάποια προϊόντα εξωτερικού έχουν αγγλική ονομασία

User Priority (5/5) : Για κάποιον χρήστη που δεν γνωρίζει ελληνικά είναι βασικό να υποστηρίζει αγγλικά το σύστημα .

Technical Priority (3/5): Μεσαίας σημασίας για το σύστημα.

Stability: (4/5) Η απαίτηση αυτή ενδέχεται να αλλάξει στο μέλλον αν χρειαστεί να προστεθούν καινούργιες γλώσσες.

<ΜΛΑ- 7>

Το σύστημα πρέπει να παρέχει κατηγοριοποίηση των διαθέσιμων προϊόντων.

Περιγραφή: Η μη λειτουργική απαίτηση αυτή θα βελτιώσει την χρηστικότητα της εφαρμογής και θα διευκολύνει την περιήγηση του χρήστη σε περίπτωση που ψάχνει κάποιο συγκεκριμένο προϊόν να αγοράσει. Θα έχει επίσης την δυνατότητα να βλέπει τις διάφορες εναλλακτικές επιλογές σε ένα προϊόν και θα επιλέγει αυτό που τον καλύπτει περισσότερο.

User Priority (3/5): Αρκετά σημαντικό χαρακτηριστικό για τον χρήστη και για την διευκόλυνσή του στο να δώσει μία παραγγελία.

Technical Priority (3/5): Αποτελεί κριτήριο του τρόπου μορφοποίησης και δόμησης του συστήματος, οπότε σε τεχνικό επίπεδο κατέχει έναν σημαντικό βαθμό στην σειρά υλοποίησης.

Stability: (5/5) Η απαίτηση αυτή δεν θα αλλάξει στο μέλλον καθώς θα έχει πάντα κατηγορίες.

3.3 Απαιτήσεις Αξιοπιστίας (Reliability)

<ΜΛΑ- 6>

Το σύστημα πρέπει να παρέχει κριτικές χρηστών και αξιολογήσεις για την λειτουργία της εφαρμογής.

Περιγραφή: Το σύστημα θα δίνει την δυνατότητα στον χρήστη να διατυπώσει την άποψή του για την ποιότητα των προϊόντων αλλά και να εκφράσει την ικανοποίηση ή την δυσαρέσκειά του για την εξυπηρέτηση που δέχτηκε. Με αυτόν τον τρόπο, οι νέοι χρήστες θα γνωρίζουν τις απόψεις και τα παράπονα των προηγούμενων για ορισμένα προϊόντα ή για την λειτουργία του συστήματος. Ο τρόπος των αξιολογήσεων θα γίνεται με αστεράκια. Τα πέντε αστεράκια δείχνουν το άριστο και το ένα το χειρίστο.



User Priority (4/5): Η διατύπωση της γνώμης αποτελεί για τον χρήστη ένα αρκετά σημαντικό χαρακτηριστικό της εφαρμογής, καθώς αποτελεί αναφαίρετο δικαίωμά του. Μόνο έτσι η εφαρμογή δύναται να γίνει καλύτερη και πιο φιλική προς αυτόν.

Technical Priority (2/5): Η υποστήριξη ενός τέτοιου forum ανταλλαγής απόψεων δεν συνιστά άμεση προτεραιότητα στην υλοποίηση του συστήματος και μπορεί να δομηθεί ανεξάρτητα στα τελικά στάδια της εφαρμογής

Stability: (5/5) Η απαίτηση αυτή δεν θα αλλάξει στο μέλλον .

3.4 Τεχνικές Απαιτήσεις Περιβάλλοντος

<ΜΛΑ- 1>

Το σύστημα πρέπει να υποστηρίζει πληρωμή μέσω PayPal και με χρεωστική/πιστωτική κάρτα

Περιγραφή: Το σύστημα πρέπει να υποστηρίζει τα δημοφιλή συστήματα πληρωμής Paypal και χρεωστικής/πιστωτικής κάρτας, για να παρέχει στον χρήστη την δυνατότητα να επιλέξει, ανάμεσα σε πολλούς τρόπους πληρωμής, αυτόν που εκείνος επιθυμεί.

User Priority (4/5) : Ο χρήστης αισθάνεται την ανάγκη, για την δική του άνεση, να έχει πολλούς εναλλακτικούς τρόπους πληρωμής.

Technical Priority (5/5): Το στάδιο αυτό είναι βασικό για την διεκπεραίωση της συναλλαγής.

Stability: (4/5) Η απαίτηση αυτή ενδέχεται να αλλάξει στο μέλλον αν χρειαστεί να προστεθούν καινούργιοι τρόποι πληρωμής.

<ΜΛΑ- 5>

Το σύστημα πρέπει να υποστηρίζει γραφική και φωτογραφική απεικόνιση προϊόντων με ανάλυση 1080p.

Περιγραφή: Η γραφική και φωτογραφική απεικόνιση προϊόντων κάνει την εφαρμογή πιο εύχρηστη και πιο ελκυστική για τον μέσο χρήστη

User Priority (5/5) : Η γραφική και φωτογραφική απεικόνιση των προϊόντων είναι σημαντική για το χρήστη , διότι διαφορετικά η εφαρμογή θα είναι δύσχρηστη , με αποτέλεσμα να αποθαρρύνει τους χρήστες από το να την χρησιμοποιούν στην καθημερινότητα τους

Technical Priority (5/5): Υψηλής σημασίας για το σύστημα.

Stability: (4/5) Καθώς το όριο επιτρεπτού ήχου όπως και η θεωρούμενη υψηλή ανάλυση είναι ευμετάβλητες ποσότητες, ίσως αυτή η απαίτηση να μεταβληθεί στο μέλλον.

<ΜΛΑ- 8>

Το σύστημα πρέπει να υποστηρίζει δυνατότητα παραγγελίας προμηθειών από δύο διαφορετικούς προμηθευτές.

Περιγραφή: Το σύστημα πρέπει να υποστηρίζει δύο διαφορετικούς προμηθευτές έτσι ώστε να υπάρχει εναλλακτική λύση σε περίπτωση που ένας από τους δύο δεν διαθέτει το ζητούμενο προϊόν.

User Priority (4/5) : Ο χρήστης θέλει να βλέπει συνεχώς διαθέσιμα τα προϊόντα.

Technical Priority (2/5): Το στάδιο αυτό είναι χαμηλής προτεραιότητας για το σύστημα.

Stability: (3/5) Η απαίτηση αυτή ενδέχεται να αλλάξει στο μέλλον αν χρειαστεί να προστεθούν καινούργιοι προμηθευτές.

3.5 Απαιτήσεις Πολιτικής (Policy - legal, corporate)

<ΜΛΑ- 3>

Το σύστημα πρέπει να ακολουθεί όλες τις νομοθετικές διατάξεις προστασίας προσωπικών δεδομένων των χρηστών.



Περιγραφή: Το σύστημα οφείλει να ακολουθεί όλες τις νομοθετικές διατάξεις του κράτους

User Priority (5/5) : Ο χρήστης θέλει να είναι σίγουρος ότι τα προσωπικά δεδομένα του λογαριασμού του είναι καλά προστατευμένα.

Technical Priority (2/5): Μικρής σημασίας για το σύστημα.

Stability: (5/5) Όσο συχνά και να αλλάζει η νομοθεσία του κράτους η εφαρμογή είναι υποχρεωμένη να συμμορφώνεται με βάση τους νόμους.

4 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

4.1 Δομικά πρότυπα

Τα δομικά πρότυπα χρησιμοποιούνται για την μείωση της σύζευξης ανάμεσα σε κλάσεις, εμπεριέχουν αρκετά σύνθετες δομές και εισάγουν μια abstract κλάση για μελλοντικές επεκτάσεις του συστήματος.

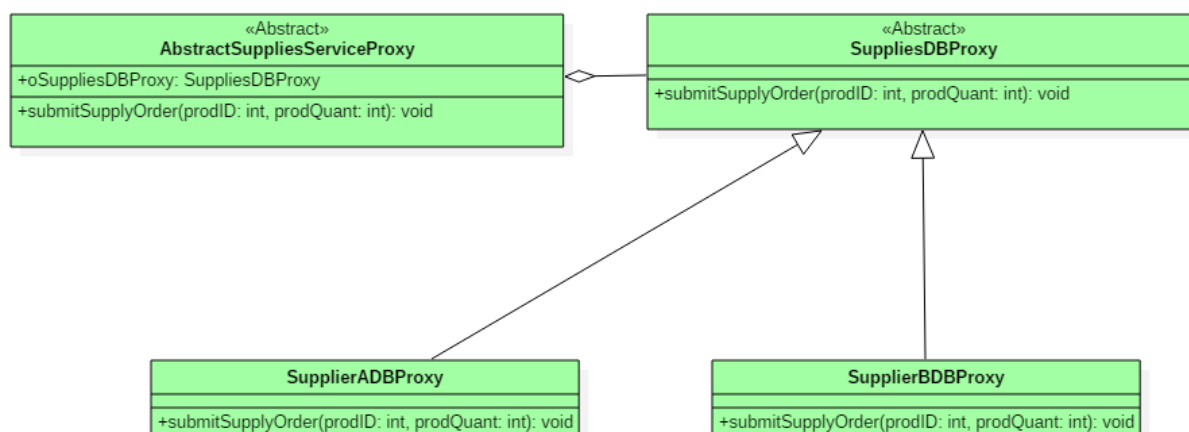
Bridge Design Pattern

Το πρότυπο της γέφυρας επιτρέπει την αποσύζευξη μιας αφαίρεσης από την υλοποίησή της, ώστε να μπορούν να υπάρχουν ανεξάρτητα. Επομένως, χρησιμοποιείται για να παρέχει πολλές υλοποιήσεις κάτω από την ίδια διεπαφή και εφαρμόζεται εκ των προτέρων για να διαχωρίσει τις αφαιρέσεις από τις υλοποιήσεις. Κατ' αυτό τον τρόπο, το σύστημά μας καθίσταται συμβατό με εξωτερικά συστήματα.

1ο Πρόβλημα που αντιμετωπίστηκε: Το πρότυπο αυτό χρησιμοποιήθηκε για να σχεδιαστεί η διεπαφή του συστήματός μας με το εξωτερικό σύστημα της βάσης δεδομένων των προμηθευτών, όπου υποβάλλονται οι παραγγελίες προϊόντων από τον admin. Ο λόγος που υιοθετήθηκε το πρότυπο αυτό ήταν η δυνατότητα αποστολής παραγγελίας σε δύο διαφορετικούς προμηθευτές εξασφαλίζοντας την εύκολη αλλαγή ενός από τους δύο σε περίπτωση που για κάποιο λόγο σταματήσει η συνεργασία μας με αυτόν. Μέσω της εφαρμογής του προτύπου αυτού, επιτυγχάνεται η ικανοποίηση της ΜΛΑ-8.

2ο Πρόβλημα που αντιμετωπίστηκε: Επίσης, χρησιμοποιήσαμε το πρότυπο αυτό, για να σχεδιαστεί η διεπαφή του συστήματός μας με τα εξωτερικά συστήματα που διατίθενται σαν πιθανοί εναλλακτικοί τρόποι πληρωμής, όπως το Payral και η ηλεκτρονική συναλλαγή μέσω τράπεζας, προς ευκολία του χρήστη. Ο λόγος που υιοθετήθηκε αυτό το πρότυπο ήταν, διότι θα πρέπει να εξασφαλίζεται η επικοινωνία με τα παραπάνω περιβάλλοντα για την επιτυχή διεξαγωγή της διαδικασίας πληρωμής. Μέσω της εφαρμογής αυτού του προτύπου, επιτυγχάνεται η ικανοποίηση της ΜΛΑ-1.

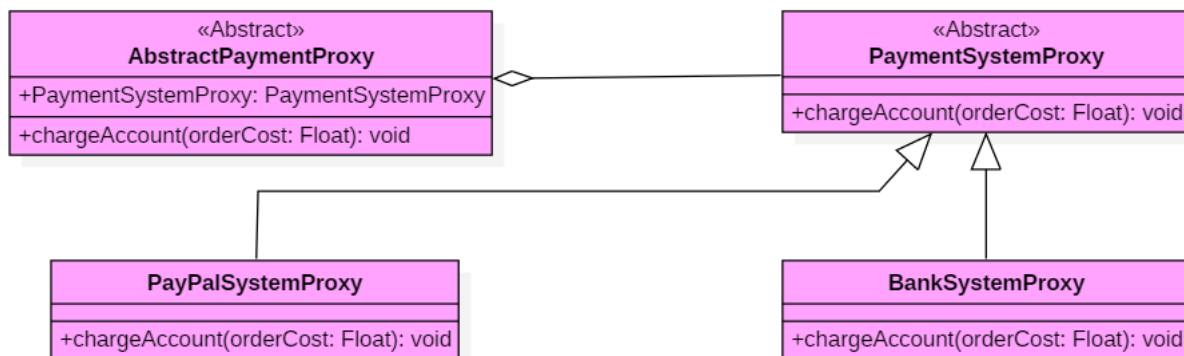
1ο υποσύστημα που αναδιοργανώθηκε: ProductHandle



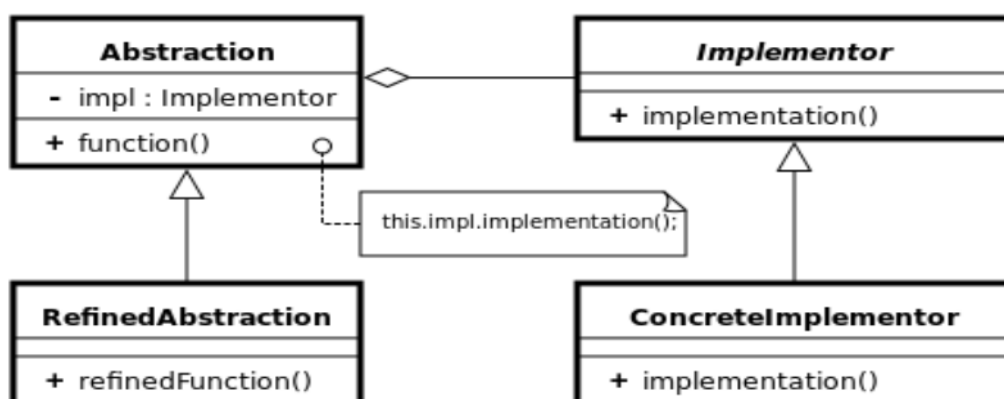
Σχήμα 5. Διάγραμμα πρότυπου σχεδίασης bridge pattern για το SuppliesServiceProxy.



2ο υποσύστημα που αναδιοργανώθηκε: CartHandle



Σχήμα 6. Διάγραμμα πρότυπου σχεδίασης bridge pattern για το PaymentProxy.



Σχήμα 7. Διάγραμμα πρότυπου σχεδίασης bridge pattern.

Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας

Έγιναν κάποιες προσθήκες στο 2^ο παραδοτέο καθώς το 1^ο φάνηκε ελλιπές. Οι προσθήκες αυτές παρουσιάζονται παρακάτω.

<ΜΛΑ- 8>



Το σύστημα πρέπει να υποστηρίζει δυνατότητα παραγγελίας προμηθειών από δύο διαφορετικούς προμηθευτές.

Περιγραφή: Το σύστημα πρέπει να υποστηρίζει δύο διαφορετικούς προμηθευτές έτσι ώστε να υπάρχει εναλλακτική λύση σε περίπτωση που ένας από τους δύο δεν διαθέτει το ζητούμενο προϊόν.

User Priority (4/5) : Ο χρήστης θέλει να βλέπει συνεχώς διαθέσιμα τα προϊόντα.

Technical Priority (2/5): Το στάδιο αυτό είναι χαμηλής προτεραιότητας για το σύστημα.

Stability: (3/5) Η απαίτηση αυτή ενδέχεται να αλλάξει στο μέλλον αν χρειαστεί να προστεθούν καινούργιοι προμηθευτές.

Προστέθηκε το PremiumOffersPage GUI του οποίου η ύπαρξη σαν οντότητα είχε αναφερθεί στο 1^ο παραδοτέο απλά δεν είχε σχεδιαστεί το αντίστοιχο mockup.

Παράρτημα II – Ανοιχτά Θέματα

Πρέπει να υλοποιηθεί το UserSettingsGUI για να μπορεί ο χρήστης να αλλάξει τα στοιχεία του