



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ

8<sup>ο</sup> ΕΞΑΜΗΝΟ 2021



# PET MY PET

*TAKING CARE OF YOUR PETS*

***Απαιτήσεις Χρηστών και  
Προδιαγραφές Συστήματος***

ΟΜΑΔΑ 12

ΤΣΟΥΣΗΣ ΠΑΝΑΓΙΩΤΗΣ - [pitsousis@ece.auth.gr](mailto:pitsousis@ece.auth.gr)

ΣΑΜΑΡΑΣ ΔΗΜΗΤΡΙΟΣ - [dzsamaras@ece.auth.gr](mailto:dzsamaras@ece.auth.gr)

ΚΑΣΙΜΙΑΔΗΣ ΓΕΩΡΓΙΟΣ - [gkasimiad@ece.auth.gr](mailto:gkasimiad@ece.auth.gr)

ΠΟΥΓΑΡΙΔΗΣ ΝΙΚΟΛΑΟΣ - [pougaridi@ece.auth.gr](mailto:pougaridi@ece.auth.gr)



## Περιεχόμενα

Λίστα σχημάτων .....	4
1.Απαιτήσεις Συστήματος .....	5
1.1 Λειτουργικές απαιτήσεις .....	5
1.2 Χρήστες και εξωτερικά συστήματα .....	7
1.3 Σημαντικές μη λειτουργικές απαιτήσεις .....	7
2. Σενάρια Χρήσης .....	10
2.1 Διάγραμμα σεναρίων χρήσης .....	10
2.2 Feature 1: Login (Feature με βάση την ορολογία Gherkin) .....	10
2.3 Feature 2: Sign up (Feature με βάση την ορολογία Gherkin) .....	11
2.4 Feature 3: Edit user information (Feature με βάση την ορολογία Gherkin) .....	12
2.5 Feature 4: Evaluate provider (Feature με βάση την ορολογία Gherkin) .....	14
2.6 Feature 5: See evaluations (Feature με βάση την ορολογία Gherkin) .....	15
2.7 Feature 6: Search Provider (Feature με βάση την ορολογία Gherkin) .....	16
2.8 Feature 7: Select Provider (Feature με βάση την ορολογία Gherkin) .....	18
2.9 Feature 8: Post a Service (Feature με βάση την ορολογία Gherkin) .....	19
2.10 Feature 9: Delete a Service (Feature με βάση την ορολογία Gherkin) .....	20
2.11 Feature 10: Pay Provider (Feature με βάση την ορολογία Gherkin) .....	21
2.12 Feature 11: Upload Photo (Feature με βάση την ορολογία Gherkin) .....	22
2.13 Feature 12: See Order (Feature με βάση την ορολογία Gherkin) .....	23
2.14 Feature 13: Delete Order (Feature με βάση την ορολογία Gherkin) .....	24
3 Επιδεικτικά γραφικά παράθυρα διεπαφής .....	24
3.1 Αρχική οθόνη .....	24
3.2 Οθόνη εισαγωγής .....	25
3.3 Οθόνη εγγραφής .....	25
3.4 Κεντρική Οθόνη .....	26
3.5 Οθόνη προσωπικών στοιχείων .....	26
3.6 Οθόνη δημιουργίας υπηρεσίας .....	27
3.7 Οθόνη αναζήτησης υπηρεσίας .....	27
3.8 Οθόνη επιλογής υπηρεσίας .....	28
3.9 Οθόνη ενεργούς παραγγελίας .....	28
3.10 Οθόνες αξιολόγησης .....	29
4 Στατική Μοντελοποίηση .....	30
4.1 <Πακέτο(α) λεξιλογίου σεναρίων υψηλής προτεραιότητας> .....	30
4.1.1 Πακέτο Landing Package: .....	30



4.1.2 Πακέτο Login Package: .....	32
4.1.3 Πακέτο Sign Up Package: .....	35
4.1.4 Πακέτο HomePage Package:.....	38
4.1.5 Πακέτο CreateService Package.....	40
4.1.6 Πακέτο SearchSelectService Package .....	45
4.1.7 Πακέτο Pay Package: .....	52
4.1.8 Πακέτο Order Package:.....	54
4.2 <Πακέτο(α) λεξιλογίου σεναρίων μέσης προτεραιότητας> .....	57
4.2.1 Πακέτο Profile Package: .....	57
4.3 <Πακέτο(α) λεξιλογίου σεναρίων χαμηλής προτεραιότητας > .....	59
4.3.1 Πακέτο Evaluation Package:.....	59
<b>Παράρτημα Ι – Γλωσσάριο .....</b>	<b>67</b>



## **Λίστα σχημάτων**

- Figure 1: Αρχική οθόνη με τα κουμπιά εισόδου και εγγραφής στην εφαρμογή.
- Figure 2: Οθόνη εισαγωγής προσωπικών στοιχείων.
- Figure 3: Οθόνη μετά την λανθασμένη εισαγωγή στοιχείων.
- Figure 4: Οθόνη εγγραφής στην εφαρμογή.
- Figure 5: Οθόνη εγγραφής μετά από λάθος συμπλήρωση στοιχείων.
- Figure 6: Κεντρική οθόνη για χρήστες Provider.
- Figure 7: Κεντρική οθόνη για χρήστες Owner.
- Figure 8: Οθόνη επεξεργασίας προσωπικών δεδομένων χρήστη.
- Figure 9: Οθόνη δημιουργίας μιας καινούριας υπηρεσίας.
- Figure 10: Οθόνη για την αναζήτηση της επιθυμητής υπηρεσίας.
- Figure 11: Οθόνη προβολής των διαθέσιμων Provider για την υπηρεσία.
- Figure 12: Οθόνη προβολής στοιχείων και επιλογής συγκεκριμένου Provider
- Figure 13: Οθόνη λεπτομερειών μια ενεργής υπηρεσίας για την προβολή ή την διαγραφή της.
- Figure 14: Οθόνη προβολής όλων των αξιολογήσεων ενός Provider.
- Figure 15: Οθόνη προβολής όλων των Provider που μπορούν να αξιολογηθούν από τον Owner.
- Figure 16: Οθόνη προβολής της αξιολόγησης των Owner για τον Provider.
- Figure 17: Οθόνη για την πραγματοποίηση μιας αξιολόγησης από τους Owner.



## 1.Απαιτήσεις Συστήματος

### 1.1 Λειτουργικές απαιτήσεις

#### (Λ.Α. - 1)

Ο User πρέπει να μπορεί να κάνει login στην εφαρμογή.

**Περιγραφή:** Ο User πρέπει να μπορεί να κάνει login στην εφαρμογή, εισάγοντας το email του και τον κωδικό.

**User Priority {5/5}:** Είναι σημαντικό για τον User, για να έχει πρόσβαση στον προσωπικό του λογαριασμό και τις υπηρεσίες της εφαρμογής.

**Technical Priority {5/5}:** Η εφαρμογή δεν μπορεί να λειτουργήσει χωρίς το login.

#### (Λ.Α. - 2)

Ο User πρέπει να μπορεί να κάνει sign up στην εφαρμογή.

**Περιγραφή:** Ο User πρέπει να μπορεί να κάνει sign up στην εφαρμογή, εισάγοντας τα προσωπικά στοιχεία του (email, όνομα, επώνυμο, κωδικό, τηλέφωνο και τη διεύθυνσή του) και επιλογή provider ή owner.

**User Priority {5/5}:** Είναι σημαντικό για τον User, για να δημιουργήσει προσωπικό λογαριασμό και να έχει πρόσβαση στις υπηρεσίες της εφαρμογής.

**Technical Priority {5/5}:** Η εφαρμογή δεν μπορεί να λειτουργήσει χωρίς το sign up.

#### (Λ.Α. - 3)

Ο User πρέπει να μπορεί να τροποποιεί τα προσωπικά του στοιχεία.

**Περιγραφή :** Ο User πρέπει να μπορεί να τροποποιεί το email, το όνομα, το επίθετο, τον κωδικό, την διεύθυνση και το τηλέφωνό του.

**User Priority {5/5}:** Είναι σημαντικό για το χρήστη γιατί ο ίδιος θέλει να βλέπει την αλλαγή στα προσωπικά του στοιχεία.

**Technical Priority {3/5}:** Τα προσωπικά στοιχεία του χρήστη δεν είναι απαραίτητα για τη λειτουργία του συστήματος.

#### (Λ.Α. - 4)

Ο owner πρέπει να μπορεί να αξιολογήσει τον provider.

**Περιγραφή:** Ο owner θα έπρεπε να μπορεί να κάνει μια αξιολόγηση του παρόχου με άριστα το 5 και να αφήσει ένα σχόλιο σχετικά με την επίδοσή του.

**User Priority {4/5}:** Είναι σημαντικό για τον owner, γιατί ο ίδιος θέλει να ξέρει αν το άτομο που θα φροντίσει το κατοικίδιό του χρήζει εμπιστοσύνης.

**Technical Priority {3/5}:** Η δυνατότητα αξιολόγησης δεν είναι απαραίτητη για τη βασική λειτουργία του συστήματος

#### (Λ.Α. - 5)

Ο provider πρέπει να μπορεί να δει τις αξιολογήσεις του.

**Περιγραφή :** Ο provider θα πρέπει να μπορεί να βλέπει τις αξιολογήσεις και τα σχόλια που του κάνουν οι owners.

**User Priority {5/5}:** Είναι σημαντικό για τον provider να γνωρίζει την γνώμη των owners έτσι ώστε να επιβεβαιώνεται για την δουλειά του ή να διορθώνει τα λάθη του.

**Technical Priority {3/5}:** Η δυνατότητα ανάγνωσης μιας αξιολόγησης δεν είναι απαραίτητη για τη βασική λειτουργία του συστήματος

#### (Λ.Α. - 6)

Ο owner πρέπει να μπορεί να αναζητήσει provider.

**Περιγραφή :** Ο owner θα πρέπει να επιλέξει κατηγορία και μέγεθος κατοικιδίου, κατηγορία υπηρεσίας, ημερομηνία και ώρα, ώστε να αναζητήσει τους providers που είναι διαθέσιμοι.



**User Priority {5/5}** : Είναι σημαντικό για τον owner γιατί είναι απαραίτητο να βρει τον κατάλληλο Provider για την παροχή της υπηρεσίας που χρειάζεται.

**Technical Priority {5/5}** : Είναι σημαντικό για το σύστημα γιατί είναι μια από τις βασικές του λειτουργίες.

#### (Λ.Α. - 7)

Ο owner πρέπει να μπορεί να επιλέξει τον provider που επιθυμεί.

**Περιγραφή** : Ο owner θα πρέπει να επιλέξει τον κατάλληλο provider, σύμφωνα με τα χαρακτηριστικά που έχει επιλέξει στην αναζήτηση.

**User Priority: {5/5}** : Είναι σημαντικό για τον owner γιατί είναι απαραίτητη η επιλογή του provider που επιθυμεί.

**Technical Priority: {5/5}**: Είναι σημαντικό για το σύστημα να ξέρει ποιος provider έχει επιλεγεί για την συνέχεια της διαδικασίας.

#### (Λ.Α. - 8)

Ο provider πρέπει να μπορεί να κάνει δημοσίευση υπηρεσίας

**Περιγραφή**: Ο provider πρέπει να μπορεί να κάνει μια δημοσίευση υπηρεσίας που θα παρέχει.

**User Priority {5/5}**: Είναι σημαντικό για τον owner να υπάρχουν οι δημοσιεύσεις των provider ώστε να επιλέξει αυτή που επιθυμεί.

**Technical Priority {2/5}**: Η δημιουργία των δημοσιεύσεων δεν είναι απαραίτητη για τη λειτουργία του συστήματος.

#### (Λ.Α. - 9)

Ο provider πρέπει να μπορεί να διαγράψει τη δημοσίευσή του.

**Περιγραφή**: Ο provider πρέπει να μπορεί να διαγράψει τη δημοσίευση υπηρεσίας που έχει αναρτήσει.

**User Priority {5/5}**: Είναι σημαντικό για τον Provider να μπορεί επαναδιαμορφώσει το πρόγραμμά του.

**Technical Priority {2/5}**: Η διαγραφή της δημοσίευσης δεν είναι απαραίτητη για τη λειτουργία του συστήματος.

#### (Λ.Α. - 10)

Ο owner πρέπει να μπορεί να πληρώσει τον provider.

**Περιγραφή** : Ο owner θα πρέπει να έχει την δυνατότητα αφού επιλέξει τον επιθυμητό Provider , να μπορεί να τον πληρώσει για την παροχή των υπηρεσιών του.

**User Priority {5/5}** : Είναι σημαντικό για τον χρήστη να προβεί στην πληρωμή του Provider για την υπηρεσία που έχει λάβει.

**Technical Priority {5/5}** : Η πληρωμή του provider από τον owner είναι σημαντική για την λειτουργία του συστήματος, για να ολοκληρωθεί η διαδικασία επιλογής υπηρεσίας.

#### (Λ.Α. - 11)

Οι χρήστες πρέπει να μπορούν να αναρτήσουν φωτογραφίες.

**Περιγραφή**: Ο User πρέπει να μπορεί να κάνει upload φωτογραφίες.

**User Priority {5/5}**: Είναι σημαντικό και για τους Users να υπάρχει η φωτογραφία.

**Technical Priority {3/5}**: Τα προσωπικά στοιχεία του Users δεν είναι απαραίτητα για τη λειτουργία του συστήματος.

#### (Λ.Α. - 12)

Ο provider πρέπει να μπορεί να δει την τρέχουσα παραγγελία του.

**Περιγραφή** : Ο provider πρέπει να μπορεί να δει την παραγγελία του, δηλαδή τις πληροφορίες του owner και της υπηρεσίας που επέλεξε.

**User Priority {5/5}**: Η δυνατότητα προβολής της παραγγελίας είναι απαραίτητη για τον provider, ώστε να ξέρει ακριβώς την υπηρεσία που πρέπει να παρέχει.



**Technical Priority {5/5}:** Η δυνατότητα προβολής της παραγγελίας είναι απαραίτητη για την λειτουργία της εφαρμογής.

**(Λ.Α. - 13)**

Ο provider πρέπει να μπορεί να διαγράψει την παραγγελία.

**Περιγραφή :** Ο provider πρέπει να έχει τη δυνατότητα να διαγράψει την παραγγελία αφού την ολοκληρώσει, ώστε να μπορεί να δει τις πληροφορίες της επόμενης παραγγελίας και να ξέρει ότι δεν έχει εκκρεμότητες.

**User Priority {5/5}:** Η δυνατότητα διαγραφής της παραγγελίας είναι απαραίτητη για τον provider, ώστε να μπορεί να δει τις πληροφορίες της επόμενης παραγγελίας.

**Technical Priority {3/5}:** Η διαγραφή της παραγγελίας θα μπορούσε να γίνεται αυτόματα από το σύστημα με την εμφάνιση μιας νέας παραγγελίας.

## **1.2 Χρήστες και εξωτερικά συστήματα**

### **1.2.1 <Unregistered User>**

Αποτελεί την κατηγορία χρηστών που δεν έχουν λογαριασμό στην εφαρμογή.

### **1.2.2 <Owner>**

Αποτελεί την κατηγορία χρηστών που έχουν εγγραφεί ως owners.

### **1.2.3 <Provider>**

Αποτελεί την κατηγορία χρηστών που έχουν εγγραφεί ως providers.

### **1.2.4 <Not Logged-in User>**

Αποτελεί την κατηγορία των χρηστών που δεν έχουν κάνει log in στην εφαρμογή.

### **1.2.5 <Βάση Δεδομένων ServicesDB>**

Για τη λειτουργία της εφαρμογής είναι αναγκαίο να υπάρχει η βάση δεδομένων ServicesDB στην οποία αποθηκεύονται οι διαθέσιμες υπηρεσίες που έχουν δημοσιεύσει οι providers.

### **1.2.6 <Βάση Δεδομένων UsersDB>**

Για τη λειτουργία της εφαρμογής είναι αναγκαίο να υπάρχει η βάση δεδομένων UsersDB στην οποία αποθηκεύονται τα προσωπικά στοιχεία των users της εφαρμογής.

### **1.2.7 <Google Maps Proxy>**

Για τη λειτουργία της εφαρμογής είναι αναγκαίο να υπάρχει η πρόσβαση σε ένα εξωτερικό σύστημα της Google, το Google Maps, για τον ακριβή προσδιορισμό της απόστασης μεταξύ owner και provider.

### **1.2.8 <Bank Proxy>**

Για τη λειτουργία της εφαρμογής είναι αναγκαίο να υπάρχει η πρόσβαση στο ασφαλές εξωτερικό περιβάλλον μιας τράπεζας, ώστε να μπορεί να πραγματοποιηθεί η συναλλαγή χρημάτων μεταξύ owner και provider.

## **1.3 Σημαντικές μη λειτουργικές απαιτήσεις**

### **(Μ.Λ.Α. - 1) <Απαιτήσεις Λειτουργίας σε Διαφορετικά Λογισμικά>**

Το σύστημα πρέπει να μπορεί να λειτουργεί σε περιβάλλον Android και Windows (7,8,1,10).

**Περιγραφή:** Το σύστημα θα πρέπει να λειτουργεί και στα 2 λογισμικά (Android και Windows).



**User Priority {5/5}:** Είναι σημαντικό προκειμένου ο User να μπορεί να χρησιμοποιήσει το σύστημα είτε από κινητό είτε από υπολογιστή.

**Technical Priority {5/5}:** Είναι σημαντικό για το σύστημα για να μπορεί να ρυθμίζει την εμφάνισή του.

#### **(Μ.Λ.Α. - 2) <Απαιτήσεις Ασφάλειας-Πολιτικής>**

Το σύστημα πρέπει να συμβαδίζει με τους κανονισμούς ασφαλείας και προσωπικών δεδομένων, όπως αναφέρονται από το GDPR.

**Περιγραφή:** Σύμφωνα με τους ισχύοντες νόμους περί προστασίας προσωπικών δεδομένων και ασφαλείας χρηστών, το σύστημα είναι σχεδιασμένο ώστε να τους ικανοποιεί πλήρως.

**User Priority {5/5}:** Είναι σημαντικό για τον χρήστη να γνωρίζει για την ασφάλεια των δεδομένων του, ώστε να μην ανησυχεί για τυχόν υποκλοπές και παραβιάσεις??

**Technical Priority {5/5}:** Για να μπορεί το σύστημα να υπάρχει στην αγορά, πρέπει να ακολουθεί τους ισχύοντες νόμους.

#### **(Μ.Λ.Α. - 3) <Google Maps Proxy>**

Το σύστημα θα πρέπει να επικοινωνεί με το περιβάλλον της google.

**Περιγραφή:** Το σύστημα μας θα πρέπει να μπορεί να επικοινωνήσει με το περιβάλλον της google και συγκεκριμένα το περιβάλλον της εφαρμογής google maps με σκοπό να πάρει δεδομένα τοποθεσίας.

**User Priority {5/5}:** Είναι σημαντικό για τους users για να βρίσκονται στην επιθυμητή απόσταση, που θα ορίζει ο provider.

**Technical Priority {5/5}:** Είναι σημαντικό για το σύστημα για τη εμφάνιση των διαθέσιμων υπηρεσιών στον owner.

#### **(Μ.Λ.Α. - 4) <Bank Payment Proxy>**

Το σύστημα θα πρέπει να επικοινωνεί με το περιβάλλον της τράπεζας

**Περιγραφή:** Το σύστημα μας θα πρέπει να μπορεί να επικοινωνήσει με το περιβάλλον της τράπεζας έτσι ώστε να πραγματοποιηθούν οι συναλλαγές που χρειάζονται.

**User Priority {5/5}:** Είναι σημαντικό για τον χρήστη για να ολοκληρώσει την πληρωμή της υπηρεσίας

**Technical Priority {5/5}:** Είναι σημαντικό για το σύστημα για την ολοκλήρωση της συναλλαγής

#### **(Μ.Λ.Α. - 5) - <Λειτουργία σε πραγματικό χρόνο>**

Το σύστημα θα πρέπει να ενημερώνει τις βάσεις δεδομένων σε χρόνο μικρότερο από 100 ms.

**Περιγραφή:** Το σύστημα μας θα πρέπει να μπορεί να ενημερώσει τις βάσεις δεδομένων σε χρόνο μικρότερο των 100 ms όταν γίνεται κάποια αλλαγή, όπως η διαγραφή μιας υπηρεσίας.

**User Priority {5/5}:** Είναι σημαντικό για τον χρήστη για να γνωρίζει πως οι υπηρεσίες που βλέπει στην οθόνη του είναι πραγματικά διαθέσιμες.

**Technical Priority {5/5}:** Είναι σημαντικό για το σύστημα για να μην υπάρχουν αιτήματα που δημιουργούν σύγκρουση (π.χ να μην είναι δυνατό να προσπαθήσει ένας provider να διαγράψει μια υπηρεσία που έχει ήδη διαγραφεί).





#### **1.4 Λεξικογραφικοί Προσδιορισμοί**

User : Ο άνθρωπος που χρησιμοποιεί την εφαρμογή, είτε ως owner είτε ως provider.

Owner : Ο User της εφαρμογής, που έχει δημιουργήσει λογαριασμό επιζητώντας υπηρεσία για κάποιο κατοικίδιο του και είναι logged in.

Provider : Ο User της εφαρμογής, που έχει δημιουργήσει λογαριασμό προσφέροντας υπηρεσίες για κατοικίδια και είναι logged in..

Υπηρεσία / Service : Ενέργεια σχετικά με κάποιο κατοικίδιο, πιο συγκεκριμένα βόλτα, φύλαξη, μετάβαση σε κτηνίατρο και εκπαίδευση.

Login : Είσοδος του User στην εφαρμογή.

Sign up: Εγγραφή του User στην εφαρμογή.

Προσωπικά στοιχεία: Email, όνομα, επώνυμο, κωδικό,τηλέφωνο και διεύθυνση ενός User.

Αξιολόγηση / Evaluation: Η βαθμολόγηση και ο σχολιασμός μιας υπηρεσίας.

Δημοσίευση υπηρεσίας: Η προσθήκη υπηρεσίας στη βάση δεδομένων των διαθέσιμων υπηρεσιών.

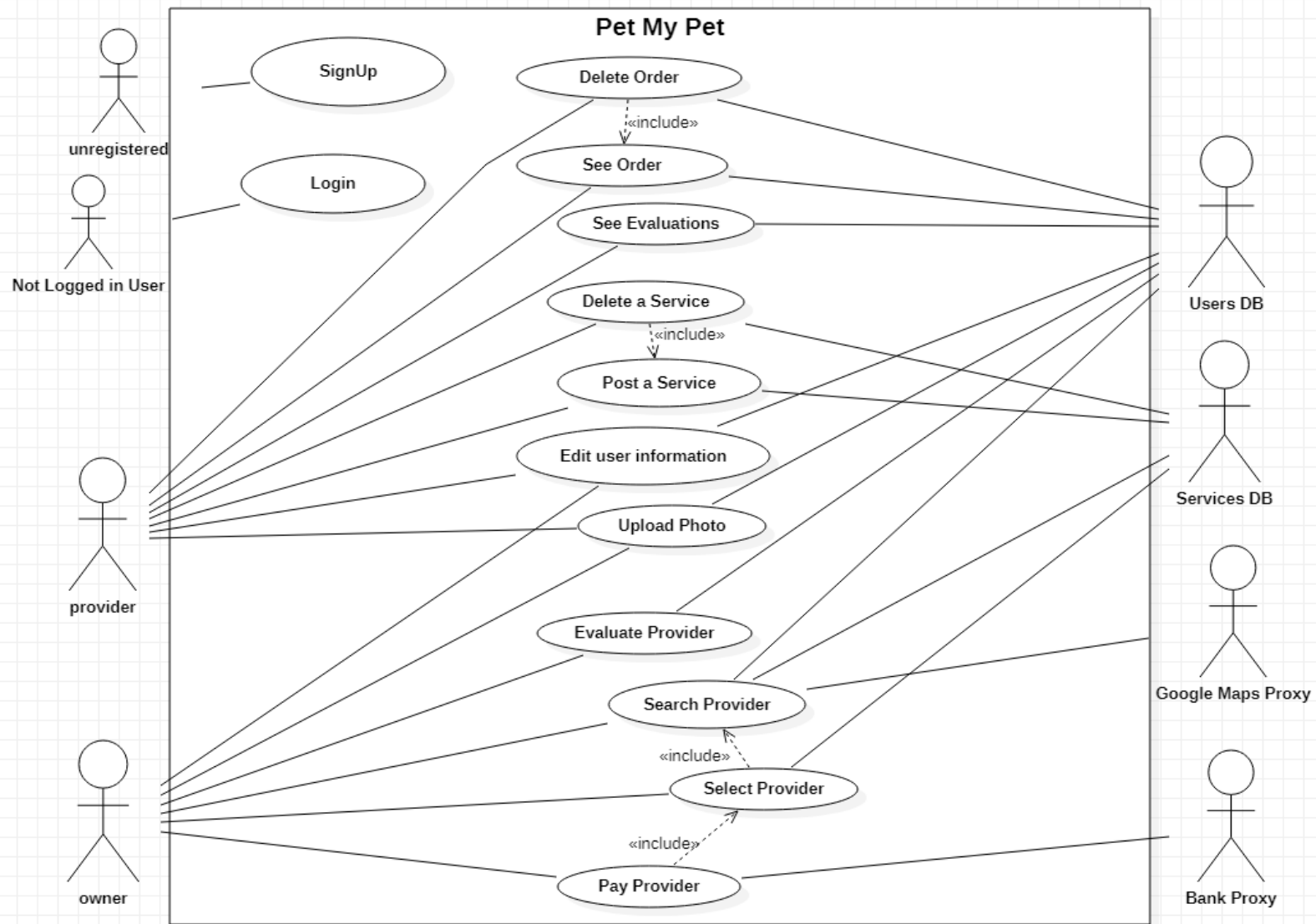
Διαγραφή δημοσίευσης υπηρεσίας: Η διαγραφή υπηρεσίας από τη βάση δεδομένων των διαθέσιμων υπηρεσιών.

Upload: Η δυνατότητα υποβολής εικόνων ως προφίλ.



## 2. Σενάρια Χρήσης

### 2.1 Διάγραμμα σεναρίων χρήσης



### 2.2 Feature 1: Login (Feature με βάση την ορολογία Gherkin)

- As a User
- So that I can access my account and the services of the application
- I want to login

**Feature:** Login

**Background:**

**Given** that I am signed up.

**Scenario:** Successful login

**When** I click "login" on the landing page

**Then** I should be directed to the login page

**And** I should be able to enter my personal information



**When** I click “login” on the login page

**Then** I should be directed to home page

**Scenario:** Unsuccessful login

**When** I click “login” on the landing page

**Then** I should be directed to the login page

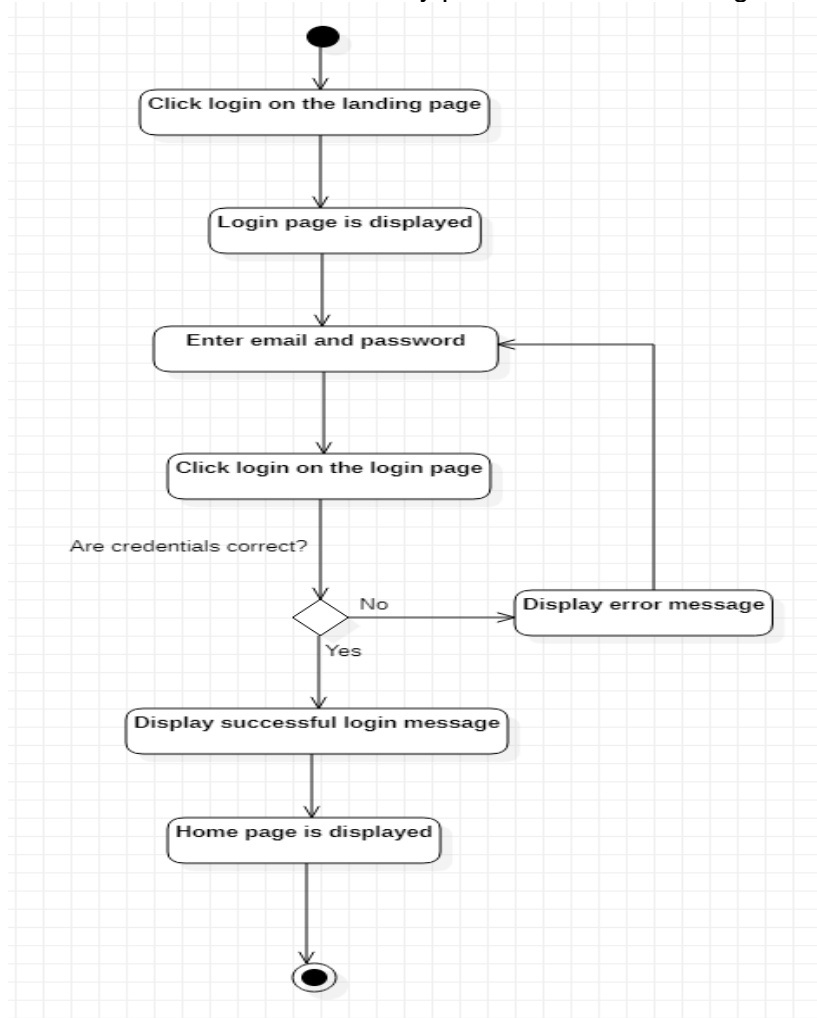
**And** I should be able to enter my personal information

**But** the email or password is incorrect

**When** I click “login” on the login page

**Then** I should see a message saying “Wrong email or password. Please try again”

**And** I should be able to enter my personal information again



### 2.3 Feature 2: Sign up (Feature με βάση την ορολογία Gherkin)

- As an unregistered User
- So that I can create an account for the application
- I want to sign up

**Feature:** Sign up

**Background:**

**Given** that I am not signed up

**Scenario:** Successful sign up

**When** I click “sign up” on the landing page

**Then** I should be directed to the sign up page



**And** I should be able to enter my personal information

**And** I should be able to choose between provider or owner account

**When** I click “sign up” on the sign up page

**Then** I should see a message saying “Successful sign up”

**And** I should be directed to the landing page

**Scenario:** Unsuccessful sign up

**When** I click “sign up” on the landing page

**Then** I should be directed to the sign up page

**And** I should be able to enter my personal information

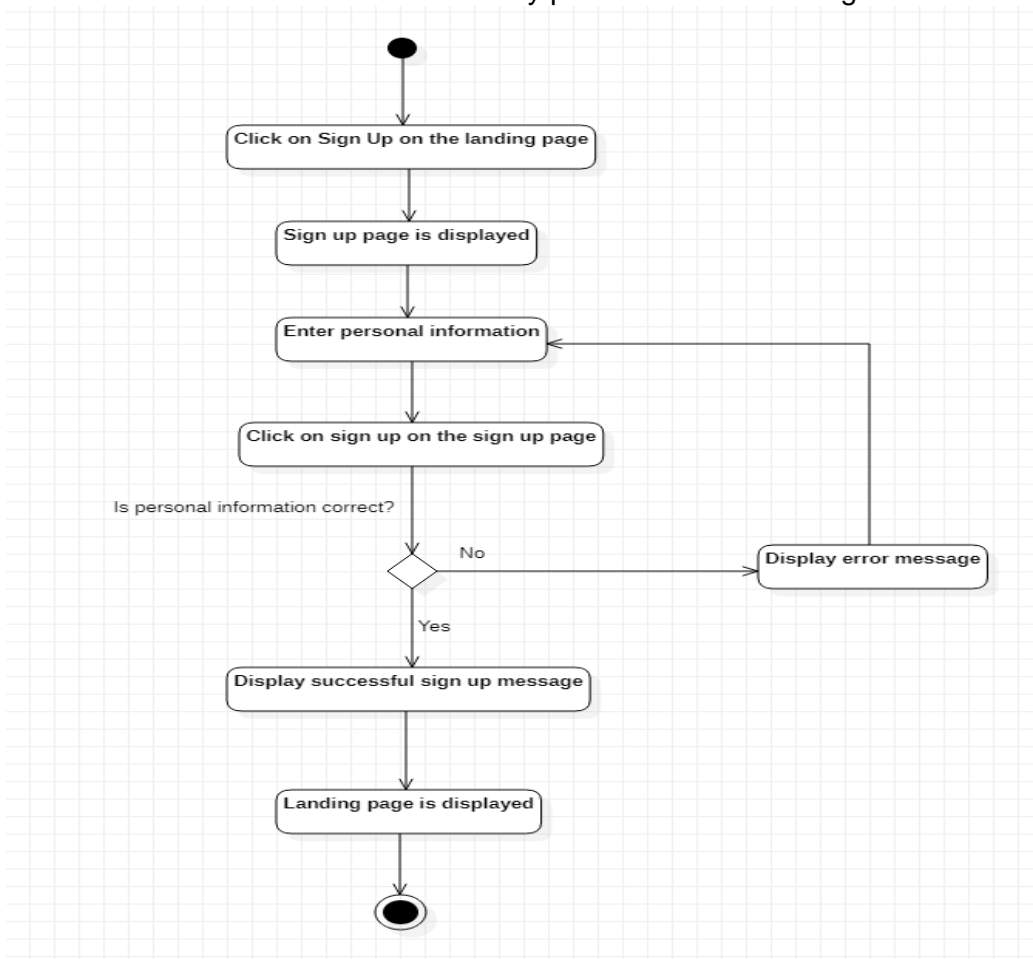
**And** I should be able to choose between provider or owner account

**But** some of the information is missing

**When** I click “sign up” on the sign up page

**Then** I should see a message saying “Missing information. Please try again”

**And** I should be able to enter my personal information again



## 2.4 Feature 3: Edit user information (Feature με βάση την ορολογία Gherkin)

- As a user
- So that I can edit my personal information
- I want to edit my personal details

**Feature:** Edit personal details

**Background:**

**Given** that I am the user

**And** I am logged in



**And** I have entered my user information

**Scenario:** Successfully editing user information

**When** I click “edit user information”

**Then** I should be asked to enter my email, name, surname, password, address and phone number

**When** I enter my email, name, surname, password, address and phone number

**And** the phone number has 10 digits

**Then** I should see a message saying “Information successfully updated.”

**And** I should be able to view the information

**And** I should be able to go back to the home page

**Scenario:** Unsuccessfully editing user information

**When** I click edit user information

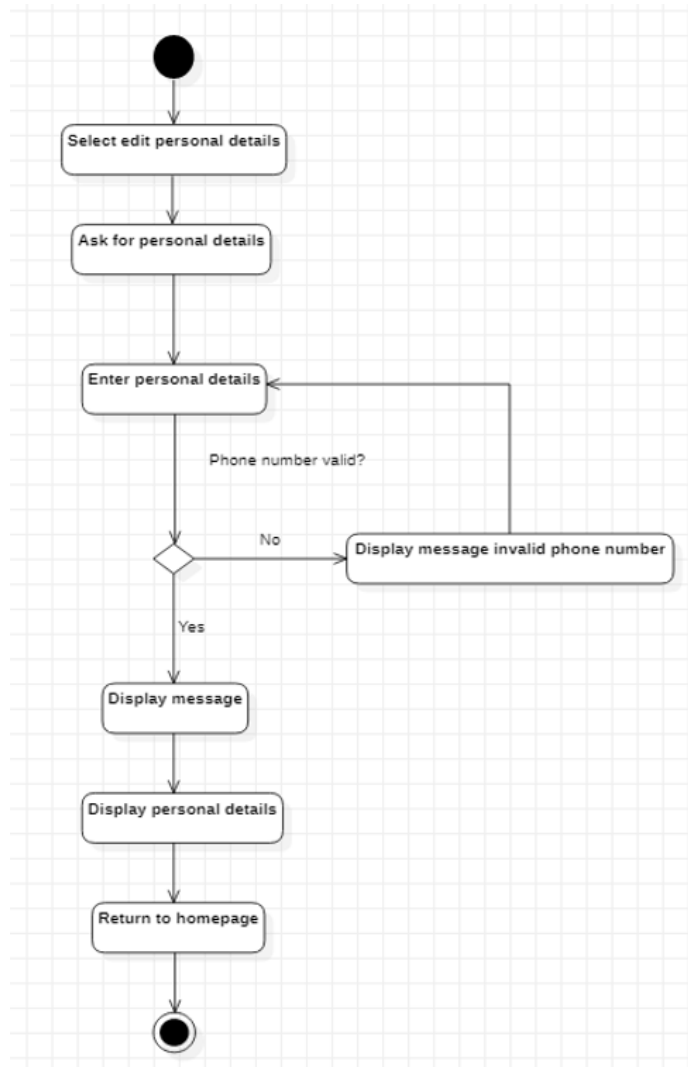
**Then** I should be asked to enter my email, name, surname, password, address and phone number

**When** I enter my email, name, surname, password, address and phone number

**But** the phone number is invalid

**Then** I should see a message saying "Invalid phone number. Please enter phone number again."

**And** I should be prompted to enter my phone number again



## 2.5 Feature 4: Evaluate provider (Feature με βάση την ορολογία Gherkin)

- As an owner
- So that I can evaluate a provider
- I want to create an evaluation of provider

**Feature:** Evaluate provider

### **Background:**

**Given** that I am logged in  
**And** that I am an owner  
**And** that a service has been completed

### **Scenario:** Successful evaluation

**When** I click “evaluate” on the home page  
**Then** I should be directed to the evaluation page 1  
**And** a list of completed services must be displayed  
**When** I click on a completed service  
**Then** I should be directed to the evaluation page 2  
**And** I should be able to evaluate the service with a mark  
**And** the mark is:  
 |[0-5]|



**And** I should be able to write a comment.

**When** I click “submit”

**Then** I should be directed to the home page.

**Scenario:** Unsuccessful evaluation

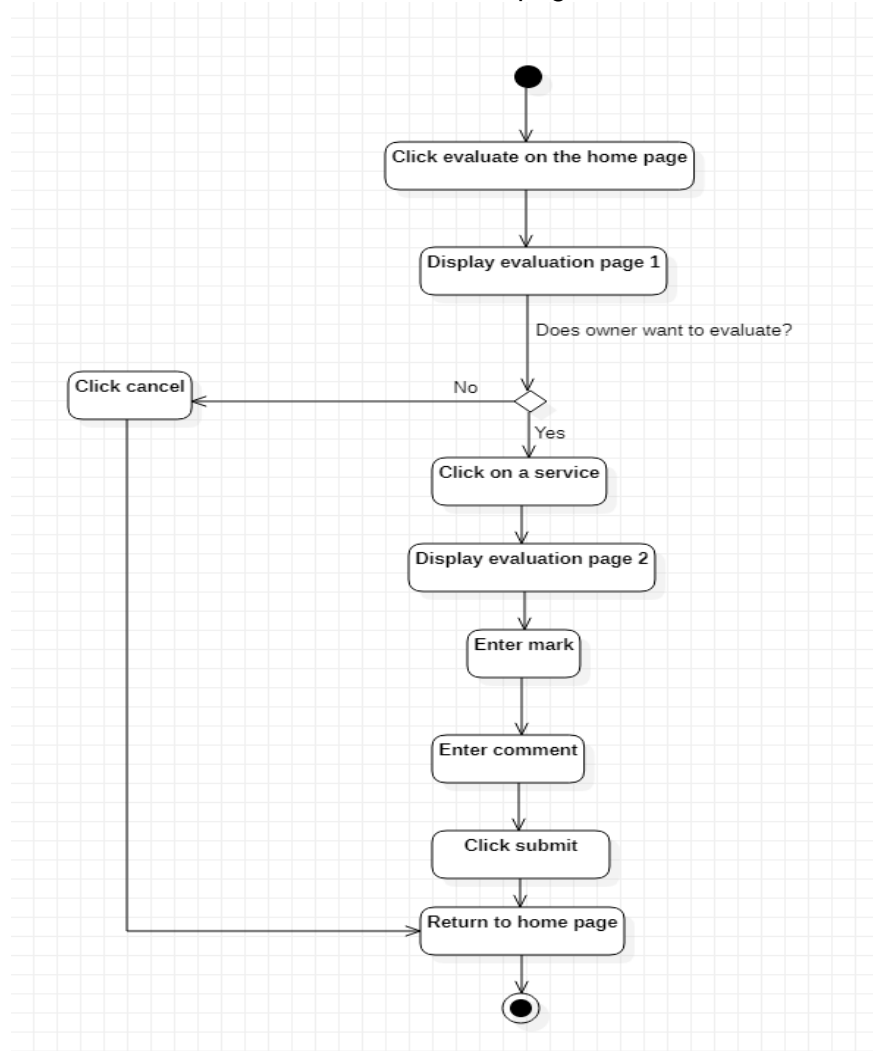
**When** I click “evaluate” on the home page

**Then** I should be directed to the evaluation page 1

**But** I change my mind

**When** I click “cancel”

**Then** I should be directed to the home page



## 2.6 Feature 5: See evaluations (Feature με βάση την ορολογία Gherkin)

- As a provider
- So that I can see the evaluations from the owners
- I want to see my evaluations

**Feature :** See evaluations

**Background:**

**Given** that I am logged in

**And** I am a provider

**Scenario :** Successful see evaluations

**When** I click on "evaluations"



**Then** I should be directed to the evaluation page

**And** I should see a list of my evaluations

**When** I select a specific evaluation

**Then** I should see more information about the evaluation

**And** I should be able to return to the home page

**Scenario** : Unsuccessful search for provider

**When** I click on "evaluations"

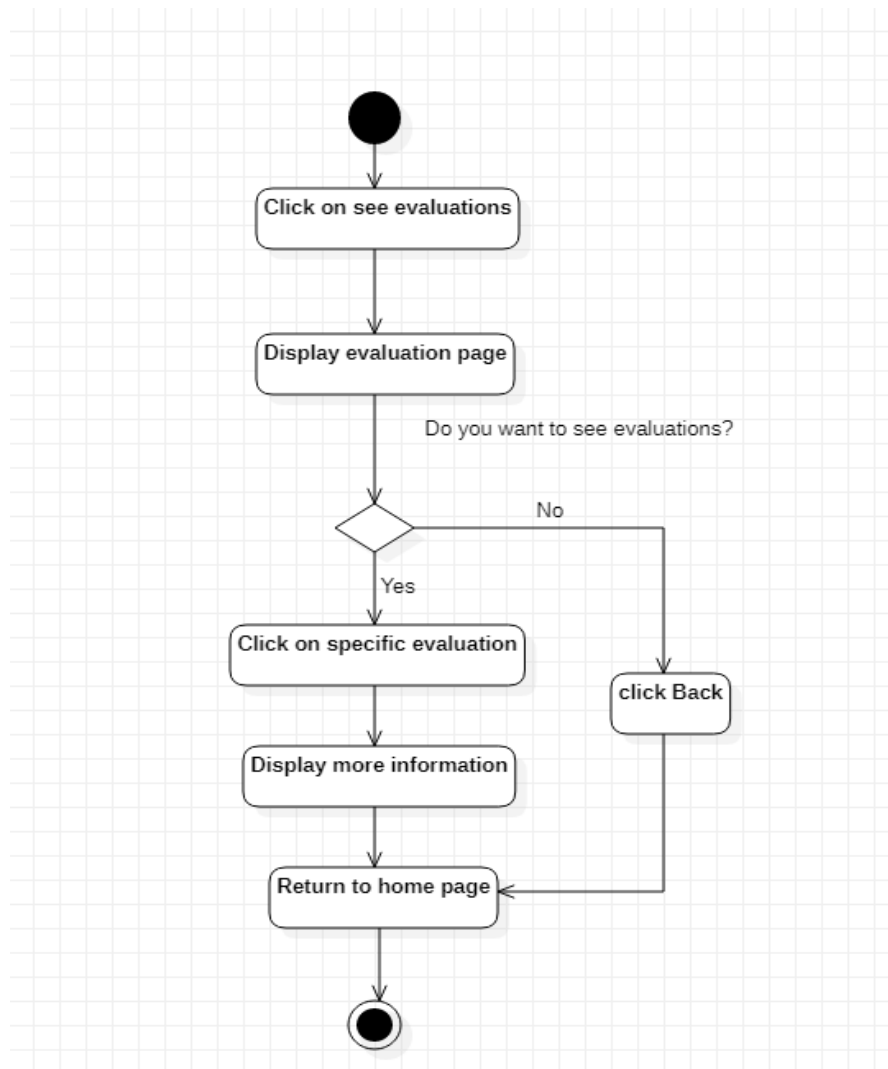
**Then** I should be directed to the evaluation page

**And** I should see a list of my evaluations

**But** I change my mind

**When** I click "back"

**Then** I should be able to return to the home page



## 2.7 Feature 6: Search Provider (Feature με βάση την ορολογία Gherkin)

- As an owner
- So that I can search for a provider
- I want to search for a provider





**Feature** : Search for Provider

**Background:**

**Given** that I am logged in

**And** I am an owner

**Scenario** : Successful search for provider

**When** I click on "search for a provider"

**Then** I should be directed to search for provider page 1

**And** I should be asked to select pet type, pet size, service type, day and time from dropdown menus.

**When** I select pet type, pet size, service, day and time

**And** I click "Search"

**Then** I should be directed to search for provider page 2

**And** I should see a list of all the available Providers

**Scenario** : Unsuccessful search for provider

**When** I click on "search for provider"

**Then** I should be directed to search for provider page 1

**And** I should be asked to select pet type, pet size, service, day and time from dropdown menus.

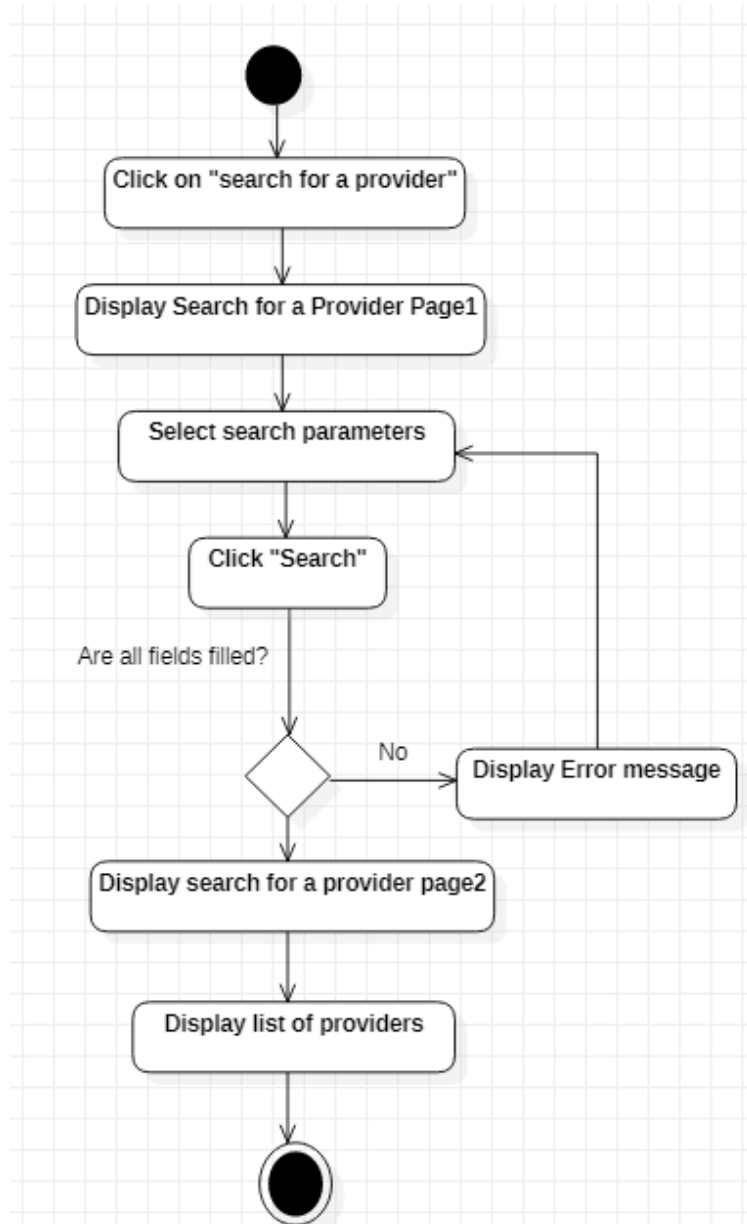
**When** I select pet type, pet size, service, day and time`

**And** I click "Search"

**But** some of the fields are not filled

**Then** I should see a message saying "Error some fields not filled"

**And** I should be prompted to select search parameters again



## **2.8 Feature 7: Select Provider (Feature με βάση την ορολογία Gherkin)**

- As an owner
- So that I can select a provider
- I want to select a provider

**Feature :** Select Provider

**Background:**

**Given** that I am the user

**And** I am logged in

**And** I have entered my user information

**Scenario:** Successfully selecting preferable Provider

**When** I click select Provider

**Then** I should be able to see a message saying " Provider Selected"

**And** I should be able to view the information

**And** I should be able to go back to the home page



## **2.9 Feature 8: Post a Service (Feature με βάση την ορολογία Gherkin)**

- As a provider
- So that I can post a service
- I want to post a service

### **Background:**

**Given** that I am logged in.

**And** that I am a Provider.

### **Scenario:** Successfully posting a service

**When** I click on "service".

**Then** I will be able to select a type of service, a pet type, a pet size, a maximum service range, the day and time and price.

**And** I should be able to click on "post", to post the service.

**Then** a pop-up message will appear saying: "Service Posted Successfully".

**Then** I will be prompted to the HomePage.

### **Scenario:** Unsuccessfully posting a service

**When** I click on "service".

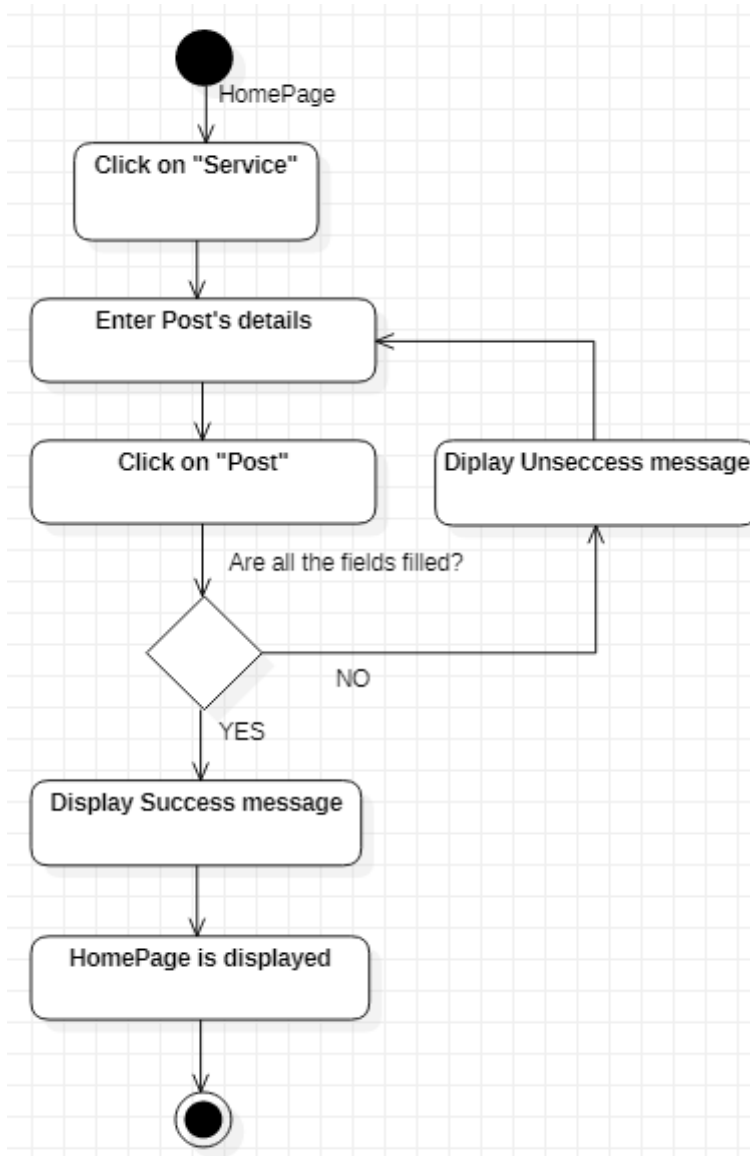
**Then** I will be able to select a type of service, a pet type, a pet size, a maximum service range, and the day and time.

**Then** I should be able to click on "post", to post the service.

**But** some of the fields are not filled.

**Then** a pop-up message will appear saying: "Service was not posted. Please fill all the fields.".

**Then** I will be asked to select the post details again.



### 2.10 Feature 9: Delete a Service (Feature με βάση την ορολογία Gherkin)

- As a provider
- In order to delete my service
- I want to delete a service

#### **Background:**

**Given** that I am logged in  
**And** that I am a Provider  
**And** I have posted a service

#### **Scenario:** Successfully deleting a service

**When** I click on "service"  
**Then** I should be directed to Post Service page  
**And** I will be able to click on "delete"  
**Then** a pop-up window will appear and ask for my confirmation  
**Then** I should be able to press "YES"  
**Then** I will be prompted to Post Service page

#### **Scenario:** Unsuccessfully deleting a service

**When** I click on "service"



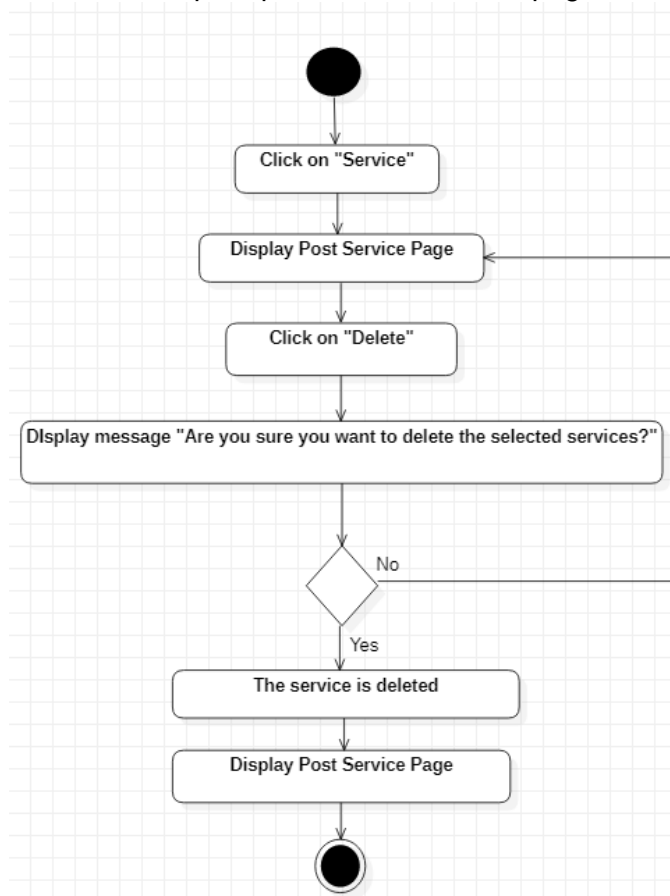
**Then** I should be directed to Post Service page

**And** I will be able to click on “delete”

**Then** a pop-up window will appear and ask for my confirmation

**Then** I should be able to press “NO”.

**Then** I will be prompted to Post Service page.



## 2.11 Feature 10: Pay Provider (Feature με βάση την ορολογία Gherkin)

- As an owner
- So that I can pay a provider
- I want to pay a provider

**Feature:** Pay provider

**Background:**

**Given** that I am logged in

**And** that I am an owner

**And** that the service has been selected

**Scenario:** Successful payment

**When** I click “Select&Pay” on Selected Provider page

**Then** I should be directed to a supported from a bank environment

**And** the payment was successful

**Then** I should be able to see a display message “Successful Payment”

**And** I should be directed to the home page.

**Scenario:** Unsuccessful payment

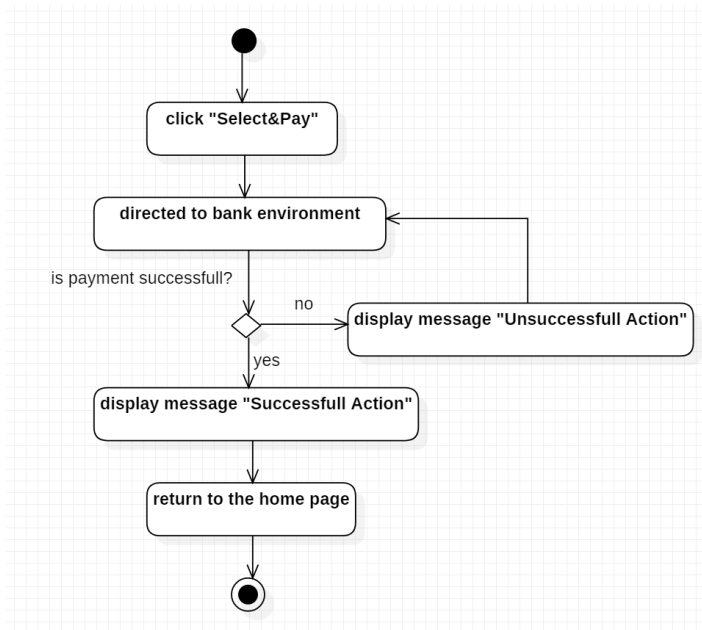
**When** I click “Select&Pay” on Selected Provider page

**Then** I should be directed to a supported from a bank environment

**But** the payment was not successful



**Then** I should be able to see a display message “Unsuccessful Payment”  
**And** I should be directed to a supported from a bank environment again



## **2.12 Feature 11: Upload Photo (Feature με βάση την ορολογία Gherkin)**

- As a User
- So that I can upload a photo.
- I want to upload a photo

**Feature** : Upload Photo

**Background:**

**Given** that I am logged in.

**And** I have entered my user information.

**And** that I am at my account page.

**Scenario:** Successfully entering photo

**When** I click on “upload photo”.

**Then** I must be asked to select the photo.

**And** I should be able to upload a photo.

**And** I should be able to save the photo.

**Then** I should be directed to the account page.

**Scenario:** Unsuccessfully entering photo

**When** I click on “upload photo”.

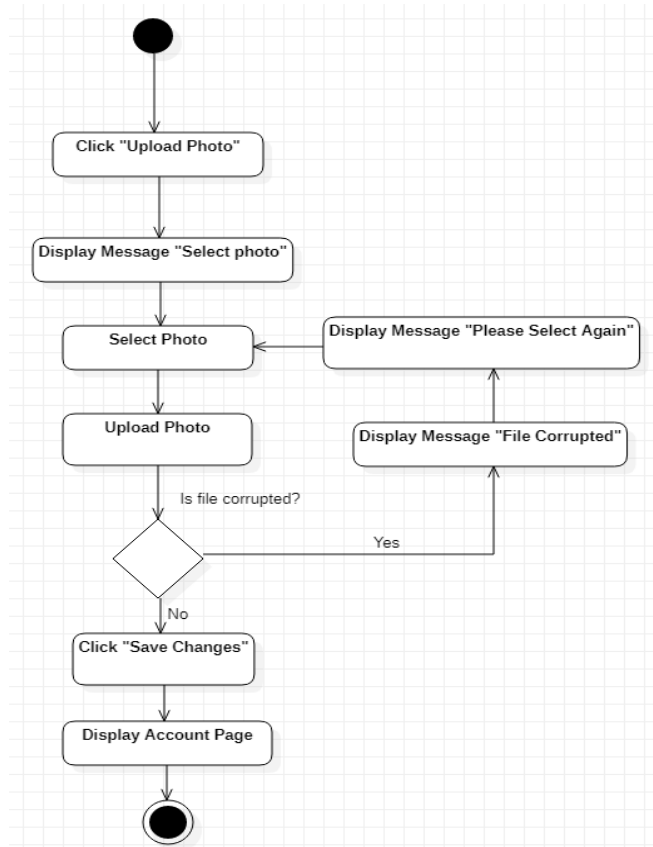
**Then** I must be asked to select the photo.

**And** I should be able to upload a photo.

**But** the file is corrupted.

**Then** I should see a message saying: "Corrupted file."

**And** I will be prompted to upload my file again.



### 2.13 Feature 12: See Order(Feature με βάση την ορολογία Gherkin)

- As a Provider
- So that I can see my order
- I want to see my active order

**Feature:** See order

**Background:**

**Given** that I am a provider

**And** I am logged in

**And** posted a service

**And** an owner has selected my service

**Scenario:** Successfully see my order

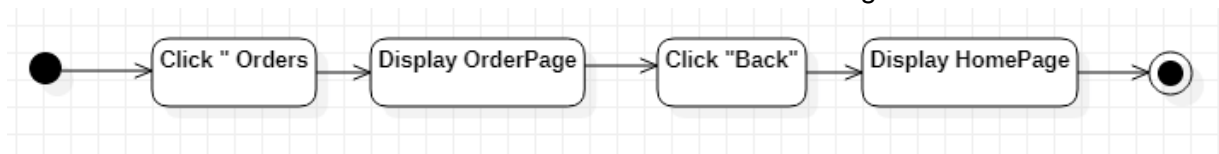
**When** I click "Orders"

**Then** I should be directed to the Order Page

**And** I should see the information of my active order

**When** I click "Back"

**Then** I should be directed to the Home Page





### 2.14 Feature 13: Delete Order (Feature με βάση την ορολογία Gherkin)

- As a Provider
- So that I can delete my order
- I want to delete my active order

**Feature:** Delete order

**Background:**

**Given** that I am a provider

**And** I am logged in

**And** I have an active order

**Scenario:** Successfully deleting active order

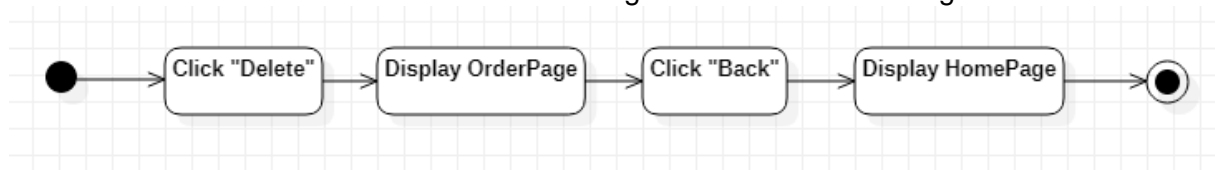
**When** I click delete on order page

**Then** all the active order details should be erased

**And** I should be directed to the Order Page

**When** I click back on Order Page

**Then** I should be able to go back to the Home Page



## 3 Επιδεικτικά γραφικά παράθυρα διεπαφής

### 3.1 Αρχική οθόνη

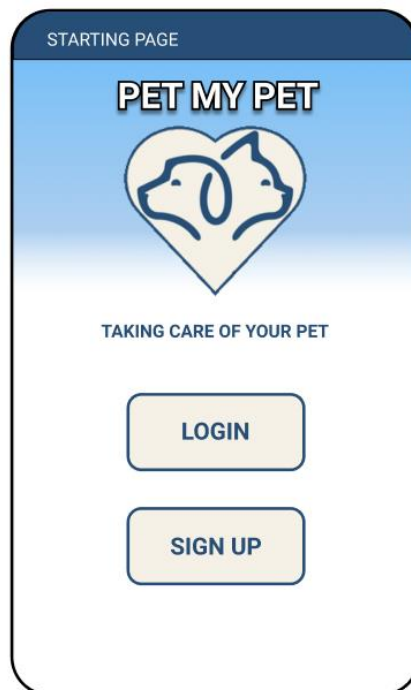


Figure 1: Αρχική οθόνη με τα κουμπιά εισόδου και εγγραφής στην εφαρμογή.





### 3.2 Οθόνη εισαγωγής

Logo: Two dog heads inside a heart.

Email

Password

LOGIN

Figure 2 : Οθόνη εισαγωγής προσωπικών στοιχείων.

Logo: Two dog heads inside a heart.

Email

Password

Combination of email and password is incorrect. Please try again.

LOGIN

Figure 3: Οθόνη μετά την λανθασμένη εισαγωγή στοιχείων.

### 3.3 Οθόνη εγγραφής

REGISTER PAGE

Logo: Two dog heads inside a heart.

Full Name \*

Email \*

Password \*

Phone \*

Address \*

☐ Owner ☐ Provider

SIGN UP

Figure 4: Οθόνη εγγραφής στην εφαρμογή.

REGISTER PAGE

Logo: Two dog heads inside a heart.

Full Name \*

WARNING

-All fields must be filled  
or  
-Email already in use

Address \*

☐ Owner ☐ Provider

SIGN UP

Figure 5: Οθόνη εγγραφής μετά από λάθος συμπλήρωση στοιχείων.



### 3.4 Κεντρική Οθόνη

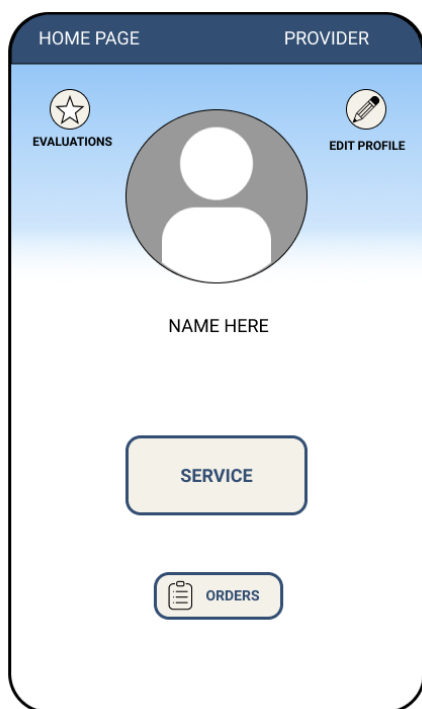


Figure 6:Κεντρική οθόνη για χρήστες Provider.

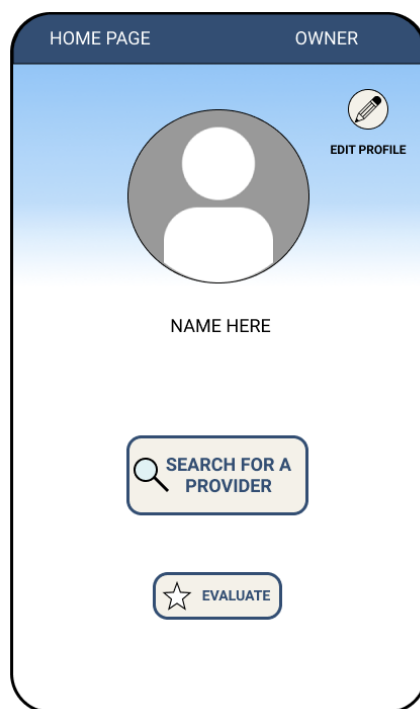


Figure 7:Κεντρική οθόνη για χρήστες Owner.

### 3.5 Οθόνη προσωπικών στοιχείων

MY PROFILE

UPLOAD PHOTO

E-MAIL:

FULL NAME:

PASSWORD:

ADDRESS:

PHONE:

SAVE CANCEL

Figure 8:Οθόνη επεξεργασίας προσωπικών δεδομένων χρήστη.



### 3.6 Οθόνη δημιουργίας υπηρεσίας

POST A SERVICE PAGE

**SERVICE:**

**PET TYPE:**

**PET SIZE:**

**DAYS:**

☐ MONDAY ☐ FRIDAY

☐ TUESDAY ☐ SATURDAY

☐ WEDNESDAY ☐ SUNDAY

☐ THURSDAY

**TIME:**  -

**RANGE: (KM)**

**PRICE:**

Figure 9: Οθόνη δημιουργίας μιας καινούριας υπηρεσίας.

### 3.7 Οθόνη αναζήτησης υπηρεσίας

SEARCH SERVICE

**SEARCH FOR A SERVICE**

**SERVICE:**

**PET TYPE:**

**PET SIZE:**

**DATE:**

December 2014

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27

**TIME:**

Figure 10: Οθόνη για την αναζήτηση της επιθυμητής υπηρεσίας.



SELECT SERVICE

Available Providers:

	Provider's Name	☆☆☆☆☆	XX €
	Provider's Name	☆☆☆☆☆	XX €
	Provider's Name	☆☆☆☆☆	XX €
	Provider's Name	☆☆☆☆☆	XX €

BACK

Figure 11:Οθόνη προβολής των διαθέσιμων Provider για την υπηρεσία.

SELECT SERVICE

Provider's Name

Price: xx€

Rating: ☆☆☆☆☆

x.x / 5.0

SELECT & PAY BACK

Figure 12:Οθόνη προβολής στοιχείων και επιλογής συγκεκριμένου Provider

**3.9 Οθόνη ενεργούς παραγγελίας**

ORDER PAGE

**ACTIVE ORDER**

OWNER NAME:

OWNER ADDRESS:

DATE SELECTED:

TIME SELECTED:

SERVICE TYPE:

PET TYPE:

PET SIZE:

PRICE:

DELETE BACK

Figure 13:Οθόνη λεπτομερειών μια ενεργής υπηρεσίας για την προβολή ή την διαγραφή της.



## ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΤΩΝ

### 3.10 Οθόνες αξιολόγησης

## TAKING CARE OF YOUR PETS

EVALUATION PAGE PROVIDER

**Services**

OVERALL RATING  
X.X/5.0

Owner's Name  
Service Type  
dd/MM/yyyy hh:mm

Owner's Name  
Service Type  
dd/MM/yyyy hh:mm

Owner's Name  
Service Type  
dd/MM/yyyy hh:mm

Owner's Name  
Service Type  
dd/MM/yyyy hh:mm

Owner's Name  
Service Type  
dd/MM/yyyy hh:mm

Owner's Name  
Service Type  
dd/MM/yyyy hh:mm

CANCEL

Figure 14: Οθόνη προβολής όλων των αξιολογήσεων ενός Provider.

EVALUATION PAGE OWNER

**Services**

Evaluating a service, improves the performance of our partners.

Provider's Name  
Service Type  
dd/MM/yyyy hh:mm

Provider's Name  
Service Type  
dd/MM/yyyy hh:mm

Provider's Name  
Service Type  
dd/MM/yyyy hh:mm

Provider's Name  
Service Type  
dd/MM/yyyy hh:mm

Provider's Name  
Service Type  
dd/MM/yyyy hh:mm

Provider's Name  
Service Type  
dd/MM/yyyy hh:mm

CANCEL

Figure 15: Οθόνη προβολής όλων των Provider που μπορούν να αξιολογηθούν από τον Owner.

EVALUATION PAGE PROVIDER

Owner's Name  
Service Type  
dd/MM/yyyy hh:mm

**Star Rating:**

★ ☆ ☆ ☆ ☆

**Comments:**

COMMENTS HERE!

BACK

Figure 16: Οθόνη προβολής της αξιολόγησης των Owner για τον Provider.

EVALUATION PAGE OWNER

Provider's Name  
Service Type  
dd/MM/yyyy hh:mm

**Star Rating:**

☆ ☆ ☆ ☆ ☆

**Comments:**

COMMENTS HERE!

SUBMIT CANCEL

Figure 17: Οθόνη για την πραγματοποίηση μιας αξιολόγησης από τους Owner.

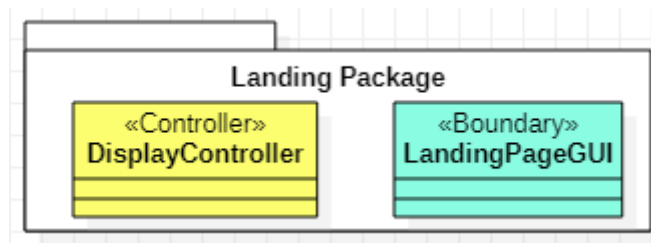


## 4 Στατική Μοντελοποίηση

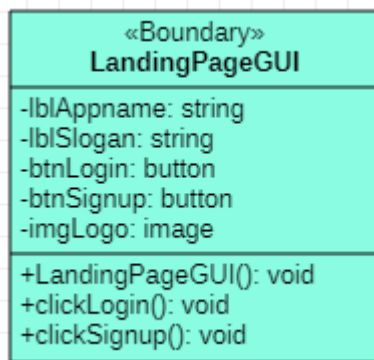
### 4.1 <Πακέτο(α) λεξιλογίου σεναρίων υψηλής προτεραιότητας>

#### 4.1.1 Πακέτο Landing Package:

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που έχουν σχέση με την οθόνη καλωσορίσματος, καθώς και τον ελεγκτή που είναι υπεύθυνος για την πλοήγηση του χρήστη στο σύστημα.



#### Boundary LandingPageGUI:



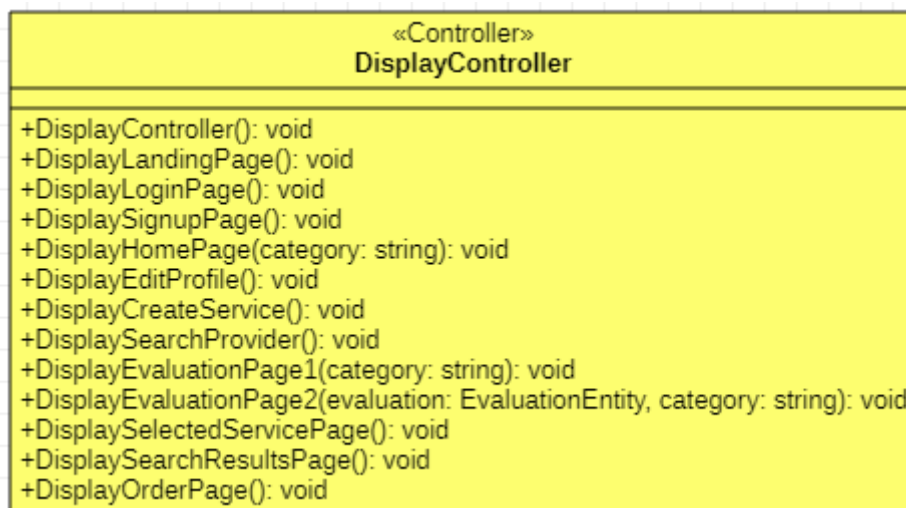
Η κλάση αυτή αναπαριστά τη διεπαφή της σελίδας προβολής του landing page.

#### Χαρακτηριστικά κλάσης:

- lblAppname: Επιγραφή του ονόματος της εφαρμογής.
- lblSlogan: Επιγραφή του slogan της εφαρμογής.
- btnLogin: Κουμπί για μετάβαση στη σελίδα login.
- btnSignup: Κουμπί για μετάβαση στη σελίδα Signup.
- imgLogo: Εικόνα με το logo της εφαρμογής.

#### Μέθοδοι Κλάσης:

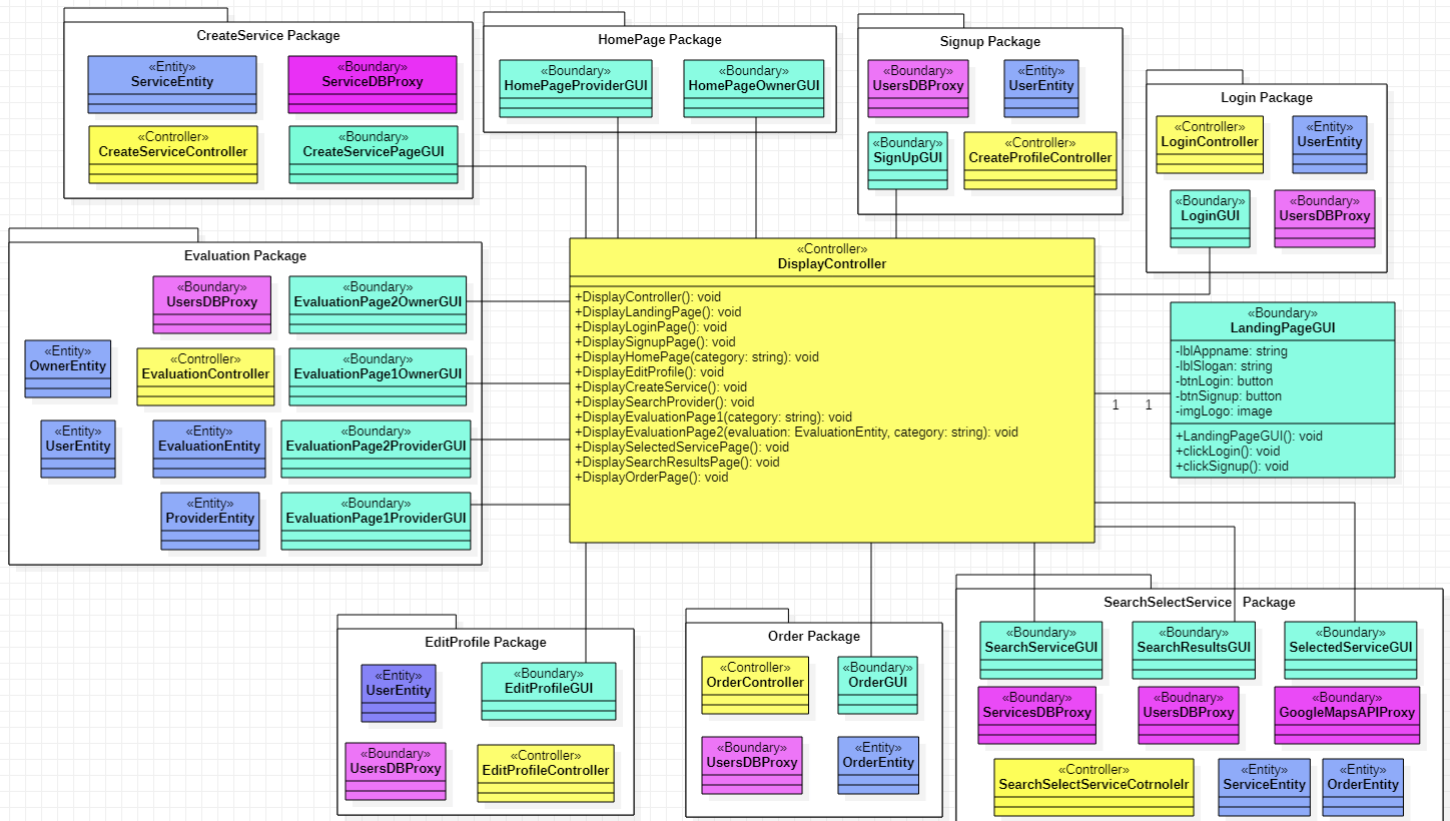
- LandingPageGUI(): Μέθοδος δόμησης της κλάσης.
- clickLogin(): Καλεί τη συνάρτηση DisplayLoginPage() του ελεγκτή DisplayController.
- clickSignup(): Καλεί τη συνάρτηση DisplaySignupPage() του ελεγκτή DisplayController.

**Controller DisplayController:**

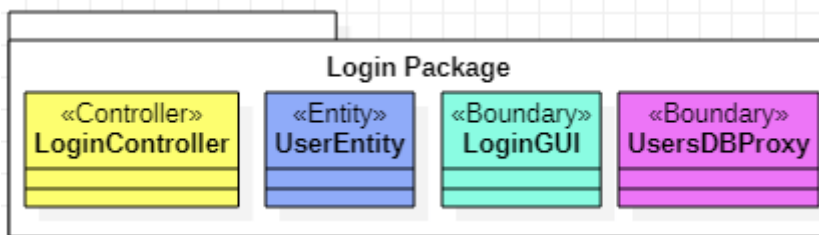
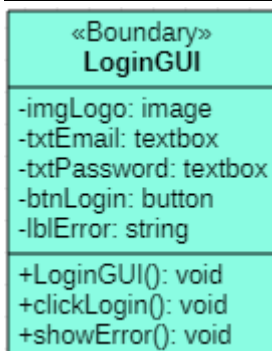
Η κλάση αυτή είναι ο ελεγκτής που έχει τις απαραίτητες συναρτήσεις για την πλοήγηση του χρήστη στις σελίδες του συστήματος.

**Μέθοδοι Κλάσης:**

- *DisplayController()*: Μέθοδος δόμησης της κλάσης.
- *DisplayLandingPage()*: Προβάλλει τη σελίδα καλωσορίσματος, δημιουργώντας ένα αντικείμενο LandingPageGUI.
- *DisplayLoginPage()*: Προβάλλει τη σελίδα login, δημιουργώντας ένα αντικείμενο LoginPageGUI.
- *DisplaySignupPage()*: Προβάλλει τη σελίδα signup, δημιουργώντας ένα αντικείμενο SignUpGUI.
- *DisplayHomePage(category: string)*: Προβάλλει τη σελίδα home page, δημιουργώντας ένα αντικείμενο HomePageProviderGUI ή HomePageOwnerGUI ανάλογα το category.
- *DisplayEditProfile()*: Προβάλλει τη σελίδα edit profile, δημιουργώντας ένα αντικείμενο EditProfileGUI.
- *DisplayCreateService()*: Προβάλλει τη σελίδα post a service page, δημιουργώντας ένα αντικείμενο CreateServicePageGUI.
- *DisplaySearchProvider()*: Προβάλλει τη σελίδα search service page, δημιουργώντας ένα αντικείμενο SearchServiceGUI.
- *DisplayEvaluationPage1(category: string)*: Προβάλλει τη σελίδα evaluation page 1, δημιουργώντας ένα αντικείμενο EvaluationPage1ProviderGUI ή EvaluationPage1OwnerGUI, ανάλογα με το όρισμα category.
- *DisplayEvaluationPage2(evaluation: EvaluationEntity, category: string)*: Προβάλλει τη σελίδα evaluation page 2, δημιουργώντας ένα αντικείμενο EvaluationPage2ProviderGUI ή EvaluationPage2OwnerGUI, ανάλογα με το όρισμα category.
- *DisplaySelectedServicePage()*: Προβάλλει τη σελίδα selected service page που εμφανίζεται αφού ο owner επιλέξει το επιθυμητό service, δημιουργώντας ένα αντικείμενο SelectedServiceGUI.
- *DisplaySearchResultsPage()*: Προβάλλει τη σελίδα search results page που εμφανίζεται αφού ο owner κάνει αναζήτηση υπηρεσίας, δημιουργώντας ένα αντικείμενο SearchResultsGUI.
- *DisplayOrderPage()*: Προβάλλει τη σελίδα order page, δημιουργώντας ένα αντικείμενο OrderGUI.

**Διάγραμμα κλάσεων:****4.1.2 Πακέτο Login Package:**

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που σχετίζονται με το login του χρήστη στο σύστημα.

**Boundary LoginGUI:**

Η κλάση αυτή αναπαριστά τη διεπαφή της σελίδας προβολής του login.





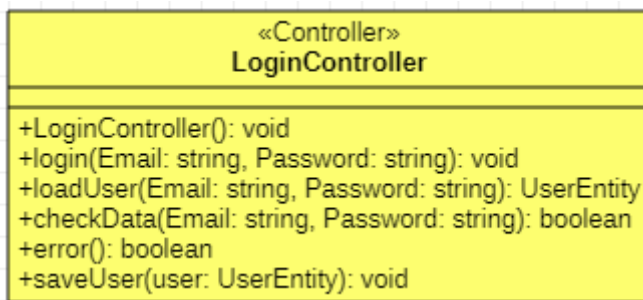
### Χαρακτηριστικά κλάσης:

- *imgLogo*: Εικόνα με το logo της εφαρμογής.
- *txtEmail*: Πεδίο εισαγωγής του email του χρήστη.
- *txtPassword*: Πεδίο εισαγωγής του κωδικού του χρήστη.
- *btnLogin*: Κουμπί υποβολής στοιχείων.
- *lblError*: Επιγραφή του μηνύματος σφάλματος.

### Μέθοδοι Κλάσης:

- *LoginGUI()*: Μέθοδος δόμησης της κλάσης
- *clickLogin()*: Καλεί τη συνάρτηση login() του ελεγκτή LoginController ώστε να γίνει η καταχώρηση και επαλήθευση των στοιχείων.
- *showError()*: Κάνει το μήνυμα σφάλματος ορατό στο χρήστη.

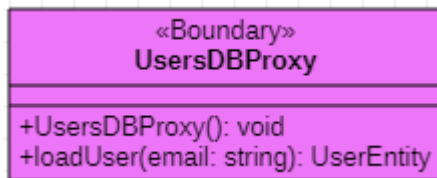
### Controller LoginController:



Η κλάση αυτή είναι ο ελεγκτής που περιέχει τις απαραίτητες συναρτήσεις για τον έλεγχο των στοιχείων του χρήστη.

### Μέθοδοι Κλάσης:

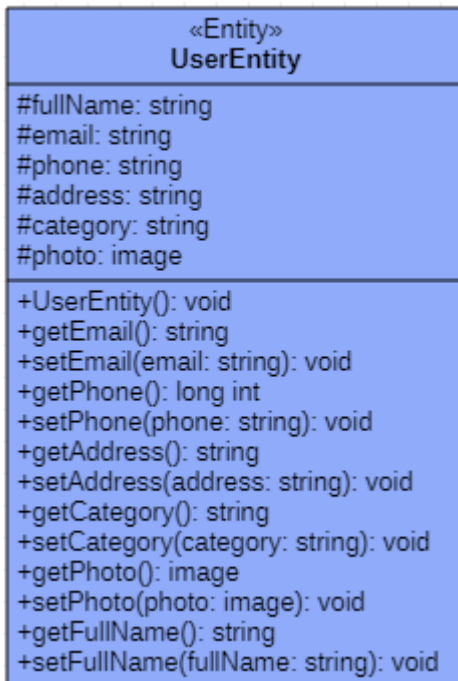
- *LoginController()*: Μέθοδος δόμησης της κλάσης
- *login(Email: string, Password: string)*: Η συνάρτηση αυτή δέχεται ως ορίσματα το email και το password που έβαλε ο χρήστης, βρίσκει το χρήστη από το UsersDB με τη συνάρτηση loadUser(Email: string, Password: string), ελέγχει αν τα στοιχεία είναι σωστά με τη συνάρτηση checkData(Email: string, Password: string) και στη συνέχεια, σε περίπτωση επιτυχίας, καλεί την DisplayHomePage(category: string) ανάλογα με το είδος του χρήστη από την κλάση DisplayController και την saveUser(user: UserEntity) για την αποθήκευση των πληροφοριών του χρήστη εσωτερικά στο σύστημα, ή καλεί την error() σε περίπτωση αποτυχίας.
- *loadUser(Email: string, Password: string)*: Καλεί την loadUser(email: string) της κλάσης UsersDBProxy και επιστρέφει ένα αντικείμενο τύπου UserEntity, με βάση το email και τον κωδικό.
- *checkData(Email: string, Password: string)*: Κάνει έλεγχο μεταξύ των στοιχείων που δόθηκαν από το χρήστη και των στοιχείων που ανακτήθηκαν από τη βάση δεδομένων.
- *error()*: Καλεί τη συνάρτηση showError() της κλάσης LoginGUI.
- *saveUser(User: UserEntity)*: Αποθηκεύει τα στοιχεία του χρήστη (ως UserEntity) εσωτερικά του συστήματος, ώστε να έχουν access στις βασικές πληροφορίες του χρήστη οι υπόλοιπες κλάσεις.

**Boundary UsersDBProxy:**

Η κλάση αυτή είναι η κλάση επικοινωνίας με τη βάση δεδομένων UsersDB, ώστε να είναι δυνατή η ανάκτηση, αποθήκευση και ενημέρωση των δεδομένων της.

**Μέθοδοι Κλάσης:**

- *UsersDBProxy()*: Μέθοδος δόμησης της κλάσης
- *loadUser(email: string)*: Αναζητεί τον χρήστη με το συγκεκριμένο email στη βάση δεδομένων και επιστρέφει το UserEntity που του αντιστοιχεί.

**Entity UserEntity:**

Η κλάση αυτή είναι ένας γενικός τύπος δεδομένων για τη μοντελοποίηση των βασικών στοιχείων ενός χρήστη, με σκοπό την αποθήκευση και ανάκτησή τους.

**Χαρακτηριστικά κλάσης:**

- *fullName*: Το ονοματεπώνυμο του χρήστη
- *email*: Το email του χρήστη
- *phone*: Το τηλέφωνο του χρήστη
- *address*: Η διεύθυνση του χρήστη
- *category*: Η κατηγορία owner ή provider του χρήστη
- *photo*: Η φωτογραφία προφίλ του χρήστη

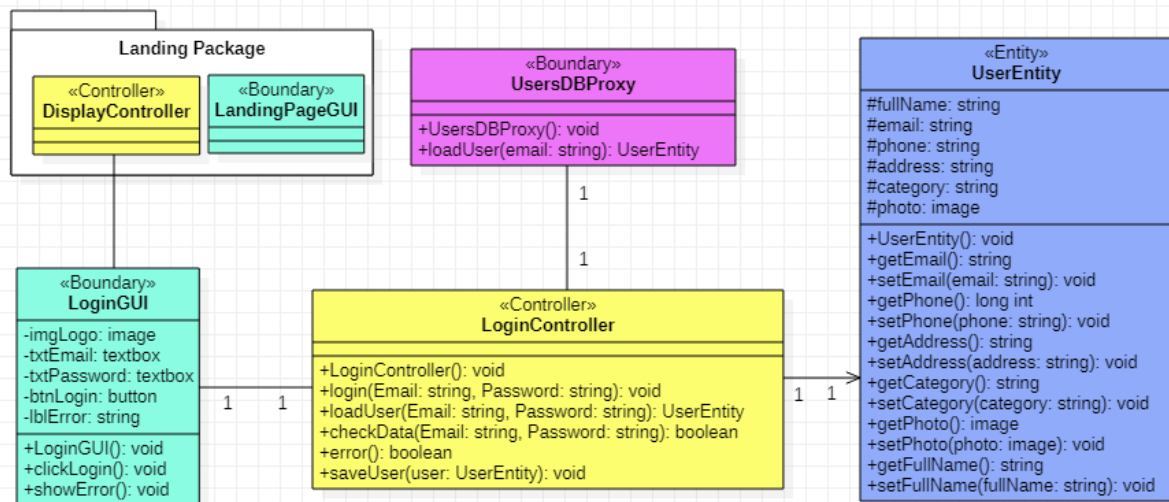
**Μέθοδοι Κλάσης:**

- *UserEntity()*: Μέθοδος δόμησης της κλάσης.
- *getFullName()*: Η μέθοδος αυτή επιστρέφει τη μεταβλητή fullName.
- *setFullName(fullName: string)*: Η μέθοδος αλλάζει τη μεταβλητή fullName στο όρισμα που της δόθηκε.



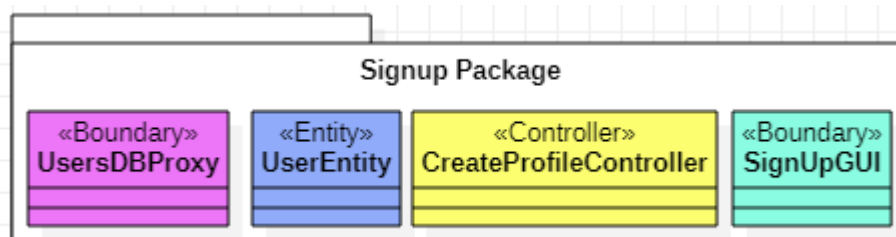
- *getEmail()*: Η μέθοδος αυτή επιστρέφει τη μεταβλητή email.
- *setEmail(email:string)*: Η μέθοδος αλλάζει τη μεταβλητή email στο όρισμα που της δόθηκε.
- *getPhone()*: Η μέθοδος αυτή επιστρέφει τη μεταβλητή phone.
- *setPhone(phone: string)*: Η μέθοδος αλλάζει τη μεταβλητή phone στο όρισμα που της δόθηκε.
- *getAddress()*: Η μέθοδος αυτή επιστρέφει τη μεταβλητή address.
- *setAddress(address: string)*: Η μέθοδος αλλάζει τη μεταβλητή address στο όρισμα που της δόθηκε.
- *getCategory()*: Η μέθοδος αυτή επιστρέφει τη μεταβλητή category.
- *setCategory(in category:string)*: Η μέθοδος αλλάζει τη μεταβλητή category στο όρισμα που της δόθηκε.
- *getPhoto()*: Η μέθοδος αυτή επιστρέφει τη μεταβλητή photo.
- *setPhoto(in photo:image)*: Η μέθοδος αλλάζει τη μεταβλητή photo στο όρισμα που της δόθηκε.

### Διάγραμμα κλάσεων:



### 4.1.3 Πακέτο Sign Up Package:

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που έχουν σχέση με την εγγραφή του χρήστη στο σύστημα και την προσθήκη των προσωπικών του στοιχείων.

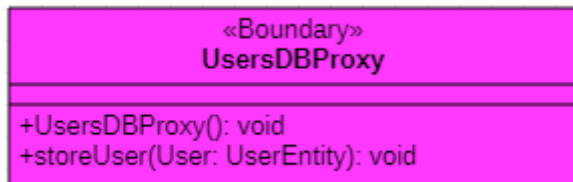


### Entity UserEntity:

Η κλάση αυτή έχει περιγραφεί στο Login Package.



### **Boundary UsersDBProxy:**

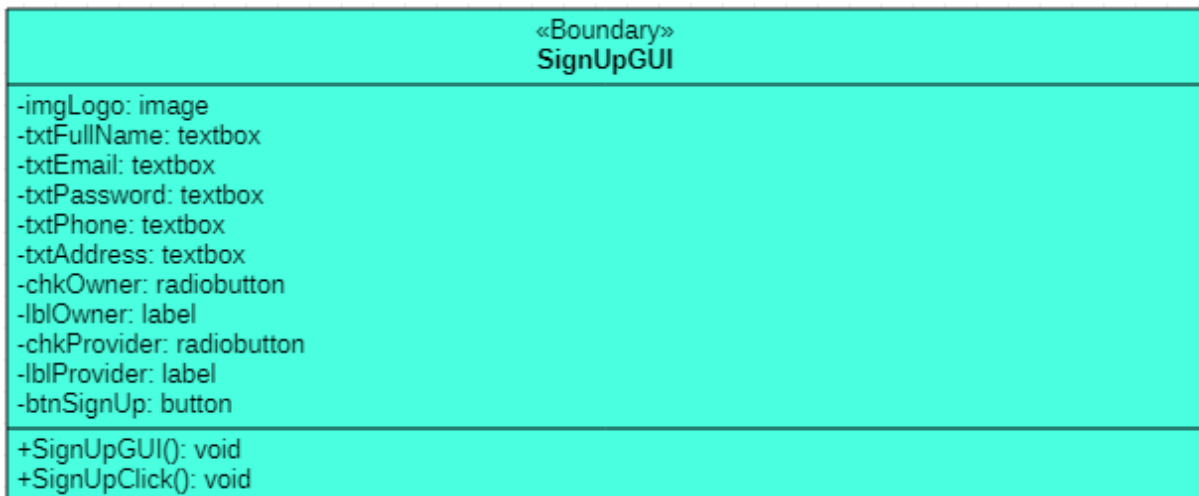


Η κλάση αυτή χρησιμοποιείται για την επικοινωνία με τη βάση δεδομένων της εφαρμογής, ώστε να αποθηκεύονται ή να ανακτώνται πληροφορίες σχετικά με τα προσωπικά στοιχεία του χρήστη.

### **Μέθοδοι κλάσης:**

- *UsersDBProxy()*: Αποτελεί συνάρτηση δόμησης της κλάσης.
- *storeUser(User: UserEntity)*: Η μέθοδος αυτή αποθηκεύει τα προσωπικά στοιχεία που καταχωρεί ο χρήστης, δηλαδή ένα αντικείμενο UserEntity, στη βάση δεδομένων της εφαρμογής.

### **Boundary SignUpGUI:**



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση της διεπαφής της προβολής του Sign Up Page στον χρήστη, που περιέχει τα προσωπικά στοιχεία τα οποία πρέπει να εισάγει.

### **Χαρακτηριστικά κλάσης:**

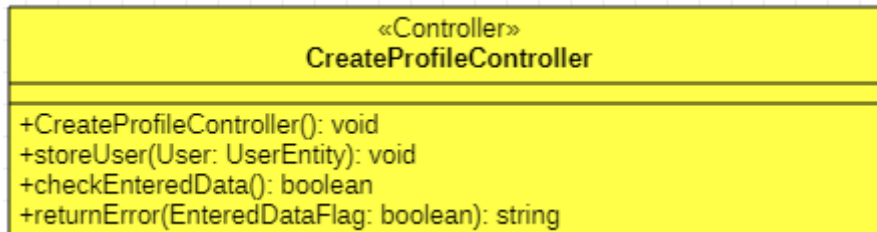
- *imgLogo*: Εικόνα με το λογότυπο της εφαρμογής.
- *txtFullName*: Πεδίο εισαγωγής του ονοματεπώνυμου του χρήστη.
- *txtEmail*: Πεδίο εισαγωγής του email του χρήστη.
- *txtPassword*: Πεδίο εισαγωγής του κωδικού του χρήστη.
- *txtPhone*: Πεδίο εισαγωγής του τηλεφώνου του χρήστη.
- *txtAddress*: Πεδίο εισαγωγής της διεύθυνσης του χρήστη.
- *chkOwner*: Radio button επιλογής της ιδιότητας Owner.
- *lblOwner*: Επιγραφή για επιλογή Owner.
- *chkProvider*: Radio button επιλογής της ιδιότητας Provider.
- *lblProvider*: Επιγραφή για επιλογή Provider.
- *btnSignUp*: Κουμπί υποβολής στοιχείων.



### Μέθοδοι κλάσης:

- *SignUpGUI()*: Αποτελεί συνάρτηση δόμησης της διεπαφής αυτής.
- *SignUpClick()*: Η μέθοδος αυτή καλεί την συνάρτηση *storeUser(User: UserEntity)* του ελεγκτή *CreateProfileController*, έτσι ώστε να αποθηκευτεί η καταχώρηση και την *DisplayLoginPage()* του *DisplayController* για να γίνει επιστροφή στη σελίδα του Login.

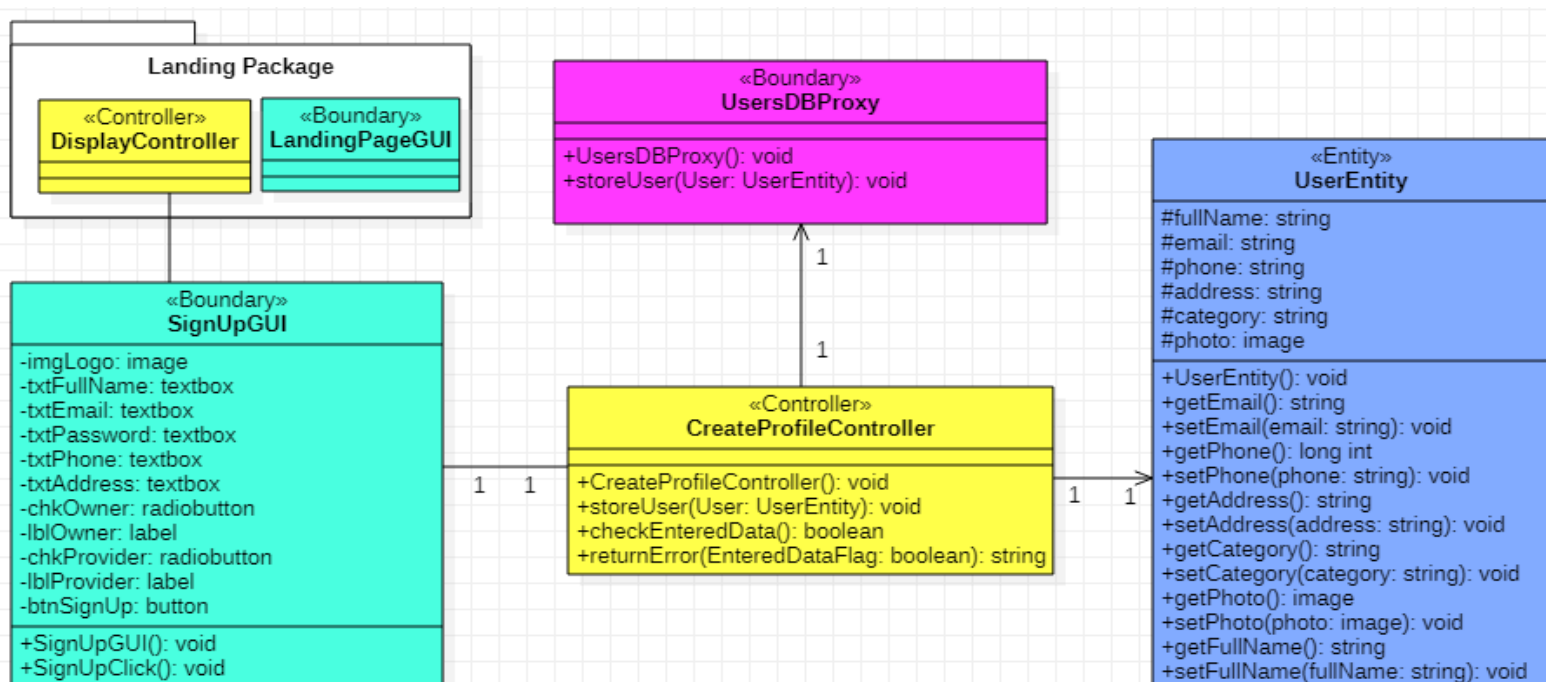
### Controller *CreateProfileController*:



### Μέθοδοι κλάσης:

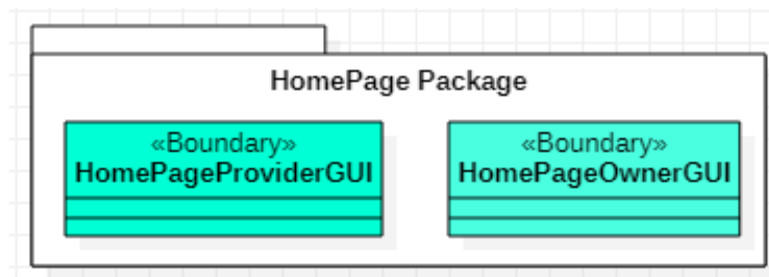
- *CreateProfileController()*: Αποτελεί συνάρτηση δόμησης του ελεγκτή.
- *storeUser(User: UserEntity)*: Η μέθοδος αυτή δημιουργεί ένα *OwnerEntity* ή *ProviderEntity* ανάλογα με την τιμή της μεταβλητής *category* και έπειτα καλεί την αντίστοιχη συνάρτηση αποθήκευσης μιας καταχώρησης προσωπικών στοιχείων στο *UsersDBProxy*, ώστε να αποθηκευτεί η πληροφορία.
- *CheckEnteredData()*: Η μέθοδος αυτή ελέγχει εάν ο χρήστης έχει συμπληρώσει σωστά τη φόρμα προσθήκης προσωπικών στοιχείων. Σε περίπτωση ορθότητας επιστρέφει τη λογική τιμή *true*, ενώ στην αντίθετη περίπτωση επιστρέφει τη λογική τιμή *false*.
- *returnError(EnteredDataFlag: boolean)*: Η συνάρτηση αυτή επιστρέφει μήνυμα σφάλματος σε περίπτωση μη ορθής συμπλήρωσης της φόρμας προσθήκης προσωπικών στοιχείων.

### Διάγραμμα κλάσεων:

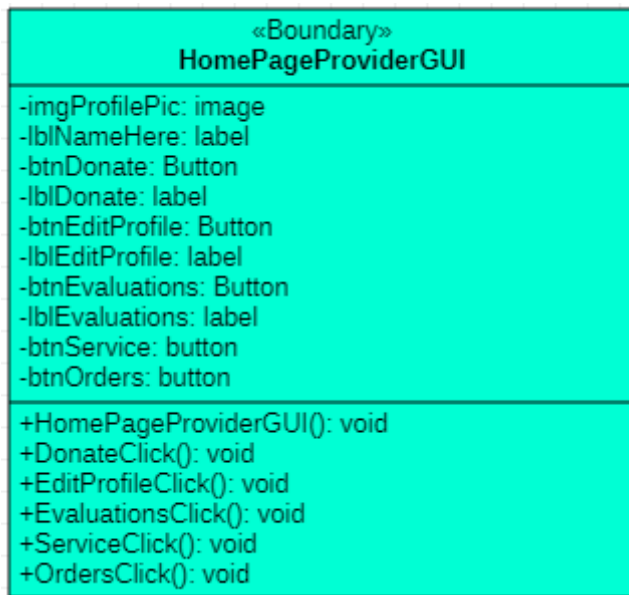




#### 4.1.4 Πακέτο HomePage Package:



#### Boundary HomePageProviderGUI:



Η κλάση αυτή αναπαριστά τη διεπαφή της αρχικής σελίδας του Provider.

#### Χαρακτηριστικά κλάσης:

- *imgProfilePic* : Εικόνα λογαριασμού του χρήστη.
- *lblNameHere* : Επιγραφή ονομασίας του χρήστη.
- *btnDonate* : Κουμπί για τη διεπαφή της σελίδας δωρεών.
- *lblDonate* : Επιγραφή για τη διεπαφή της σελίδας δωρεών.
- *bntEditProfile* : Κουμπί για τη διεπαφή της σελίδας επεξεργασίας δεδομένων.
- *lblEditProfile* : Επιγραφή για τη διεπαφή της σελίδας επεξεργασίας δεδομένων.
- *btnEvaluations* : Κουμπί για τη διεπαφή της σελίδας αξιολογήσεων.
- *lblEvaluations* : Επιγραφή για τη διεπαφή της σελίδας αξιολογήσεων.
- *btnService* : Κουμπί για τη διεπαφή της σελίδας υπηρεσιών.
- *btnOrders* : Κουμπί για τη διεπαφή της σελίδας παραγγελιών.

#### Μέθοδοι κλάσης:

- *HomePageProviderGUI()* : Αποτελεί συνάρτηση δόμησης της κλάσης.
- *DonateClick()* : Καλεί τη συνάρτηση *loadBankEnvironment(email: string)* του ελεγκτή *DonateController* ώστε να γίνει η ανακατεύθυνση στο περιβάλλον πληρωμής της τράπεζας.
- *EditProfileClick()* : Καλεί την συνάρτηση *DisplayEditProfile()* του ελεγκτή *DisplayController* ώστε να γίνει προβολή της σελίδας επεξεργασίας δεδομένων χρήστη.
- *EvaluationClick()* : Καλεί την συνάρτηση *DisplayEvaluationPage1(category: string)* του ελεγκτή *DisplayController* ώστε να γίνει προβολή της σελίδας αξιολογήσεων.





- *ServiceClick()* : Καλεί την συνάρτηση DisplayCreateService() του ελεγκτή DisplayController ώστε να γίνει προβολή της σελίδας υπηρεσιών.
- *OrdersClick()*: Καλεί τη συνάρτηση DisplayOrders() του ελεγκτή DisplayController ώστε να γίνει η προβολή της σελίδας παραγγελιών του provider.

### **Boundary HomePageOwnerGUI:**

«Boundary» HomePageOwnerGUI
-imgProfilePic: image -lblNameHere: label -btnDonate: Button -lblDonate: label -btnEditProfile: Button -lblEditProfile: label -btnSearchForAProvider: Button -btnEvaluate: Button
+HomePageOwnerGUI(): void +DonateClick(): void +EditProfileClick(): void +EvaluateClick(): void +SearchForAProviderClick(): void

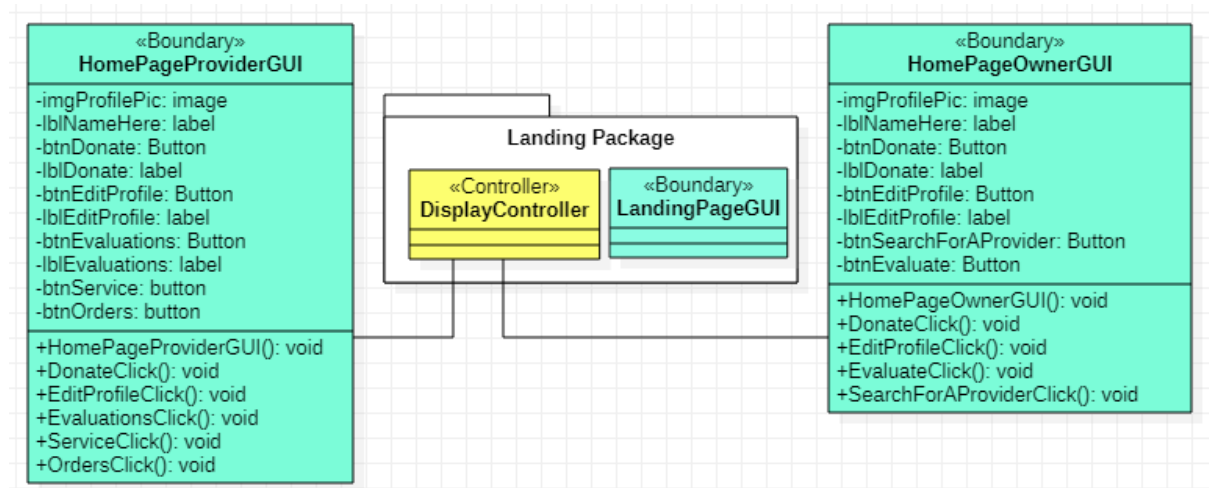
Η κλάση αυτή αναπαριστά τη διεπαφή της αρχικής σελίδας του Owner.

### **Χαρακτηριστικά κλάσης:**

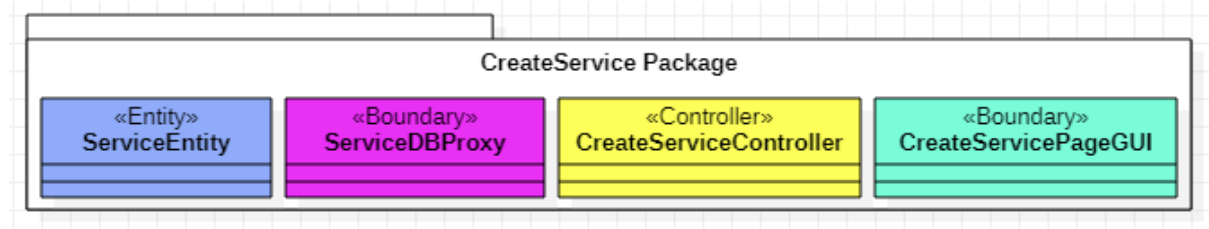
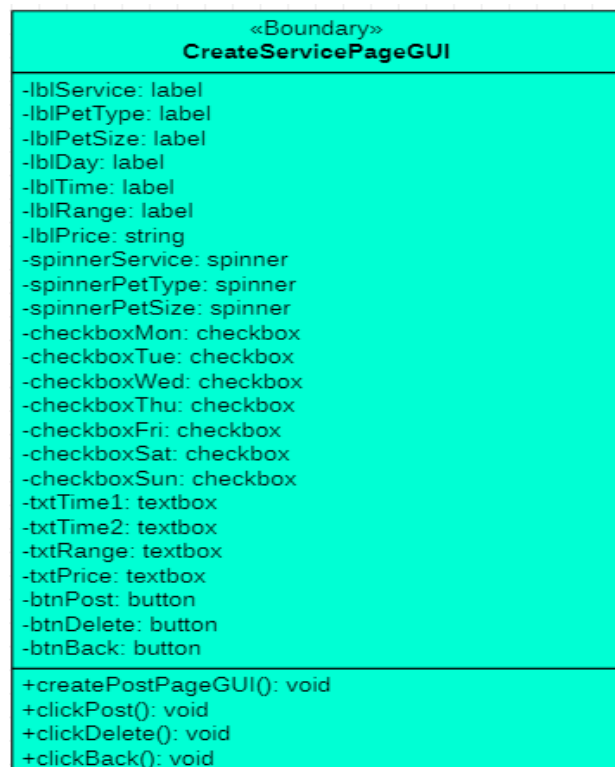
- *imgProfilePic* : Εικόνα λογαριασμού του χρήστη.
- *lblNameHere* : Επιγραφή ονόματος χρήστη.
- *btnDonate* : Κουμπί για τη διεπαφή της σελίδας δωρεών.
- *lblDonate* : Επιγραφή της διεπαφής της σελίδας δωρεών.
- *btnEditProfile* : Κουμπί για τη διεπαφή της σελίδας επεξεργασίας δεδομένων χρήστη.
- *lblEditProfile* : Επιγραφή της διεπαφής της σελίδας επεξεργασίας δεδομένων χρήστη.
- *btnSearchForAProvider* : Κουμπί για τη διεπαφή της σελίδας αναζήτησης Provider.
- *btnEvaluate* : Κουμπί για τη διεπαφή της σελίδας αξιολόγησης Provider.

### **Μέθοδοι κλάσης:**

- *HomePageOwnerGUI()* : Αποτελεί τη συνάρτηση δόμησης της κλάσης.
- *DonateClick()* : Καλεί τη συνάρτηση loadBankEnvironment(email: string) του ελεγκτή DonateController ώστε να γίνει η ανακατεύθυνση στο περιβάλλον πληρωμής της τράπεζας.
- *EditProfileClick()* : Καλεί την συνάρτηση DisplayEditProfile() του ελεγκτή DisplayController ώστε να γίνει προβολή της σελίδας επεξεργασίας δεδομένων χρήστη.
- *EvaluateClick()* : Καλεί την συνάρτηση DisplayEvaluationPage1(category: string) του ελεγκτή DisplayController ώστε να γίνει προβολή της σελίδας αξιολόγησης.
- *SearchForAProviderClick()* : Καλεί την συνάρτηση DisplaySearchProvider() του ελεγκτή DisplayController ώστε να γίνει προβολή της σελίδας αναζήτησης Provider.

**Διάγραμμα κλάσεων:****4.1.5 Πακέτο CreateService Package**

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που έχουν σχέση με την δημιουργία υπηρεσιών από τον Provider.

**Boundary CreateServicePageGUI:**





Η κλάση αυτή εκφράζει τη διεπαφή, με τη σελίδα δημιουργίας υπηρεσιών του provider.

### **Χαρακτηριστικά κλάσης:**

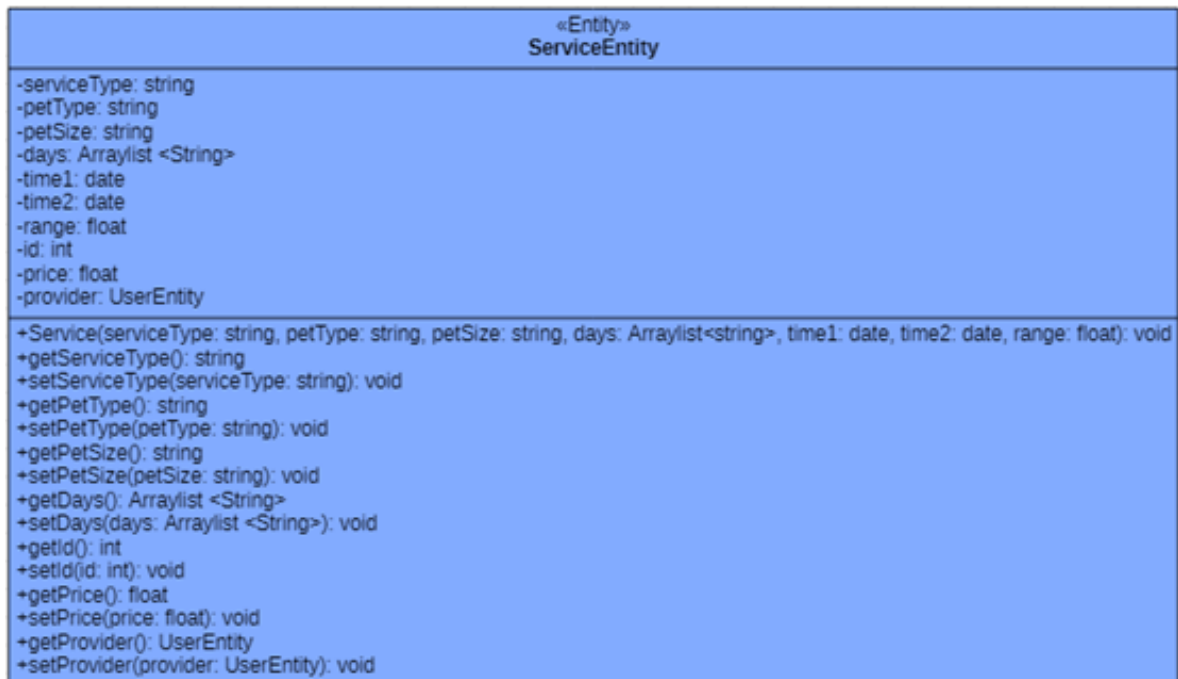
- *lblService*: Επιγραφή «SERVICE»
- *lblPetType*: Επιγραφή «PET TYPE»
- *lblPetSize*: Επιγραφή «PET SIZE»
- *lblDay*: Επιγραφή «DAYS»
- *lblTime*: Επιγραφή «TIME»
- *lblRange*: Επιγραφή «RANGE (KM) »
- *lblPrice* : Επιγραφή «Price»
- *spinnerService*: DropDown λίστα, με τις επιλογές παροχής υπηρεσίας
- *spinnerPetType*: DropDown λίστα, με τις επιλογές ειδών κατοικιδίων
- *spinnerPetSize*: DropDown λίστα, με τις επιλογές μεγέθους κατοικιδίων
- *checkboxMon*: Κουμπί επιλογής της ημέρας, με επιγραφή «Monday»
- *checkboxTue*: Κουμπί επιλογής της ημέρας, με επιγραφή «Tuesday»
- *checkboxWed*: Κουμπί επιλογής της ημέρας, με επιγραφή «Wednesday»
- *checkboxThu*: Κουμπί επιλογής της ημέρας, με επιγραφή «Thursday»
- *checkboxFri*: Κουμπί επιλογής της ημέρας, με επιγραφή «Friday»
- *checkboxSat*: Κουμπί επιλογής της ημέρας, με επιγραφή «Saturday»
- *checkboxSun*: Κουμπί επιλογής της ημέρας, με επιγραφή «Sunday»
- *txtTime1*: Πεδίο συμπλήρωσης της ώρας έναρξης της υπηρεσίας
- *txtTime2*: Πεδίο συμπλήρωσης της ώρας λήξης της υπηρεσίας
- *txtRange*: Πεδίο συμπλήρωσης της ακτίνας απόστασης από τη διεύθυνση του Provider, μέσα στην οποία θα παρασχεθεί η υπηρεσία
- *txtPrice*: Πεδίο συμπλήρωσης του κόστους της υπηρεσίας
- *btnPost*: Κουμπί δημιουργίας και δημοσίευσης της υπηρεσίας που θα παρασχεθεί
- *btnDelete*: Κουμπί διαγραφής της υπάρχουσας υπηρεσίας
- *btnBack* : Κουμπί μετάβασης στην προηγούμενη σελίδα, δηλαδή στο HomePage

### **Μέθοδοι κλάσης:**

- *createPostPageGUI()* : Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *clickPost()* : Μέθοδος που καλεί, πρώτα την *storeService(service :ServiceEntity)* του controller, και μετά την *DisplayHomePage(category : string) : void* , του *DisplayController*
- *clickDelete()* : Μέθοδος που καλεί, πρώτα την *deleteService(id : int): void* του controller, και μετά την *DisplayHomePage(category : string) : void*, του *DisplayController*
- *clickBack()* : Μέθοδος που καλεί την *DisplayHomePage(category : string) : void*, του *DisplayController*



### Entity ServiceEntity:



Η κλάση αυτή εκφράζει τα αντικείμενα τύπου ServiceEntity.

### Χαρακτηριστικά κλάσης:

- *serviceType* : Το είδος της υπηρεσίας
- *petType*: Το είδος του κατοικιδίου
- *petSize*: Το μέγεθος του κατοικιδίου
- *days*: Μια λίστα με τις επιλεγμένες μέρες
- *time1*: Η ώρα έναρξης της διαθεσιμότητας της υπηρεσίας
- *time2*: Η ώρα λήξης της διαθεσιμότητας της υπηρεσίας
- *range*: Η μέγιστη ακτίνα εξυπηρέτησης, με αρχή τη διεύθυνση του provider
- *id*: Μοναδικός αριθμός που αποδίδεται από το σύστημα σε κάθε υπηρεσία
- *price*: Το κόστος της υπηρεσίας
- *provider*: Ο provider που έχει δημοσιεύσει την υπηρεσία αυτή.

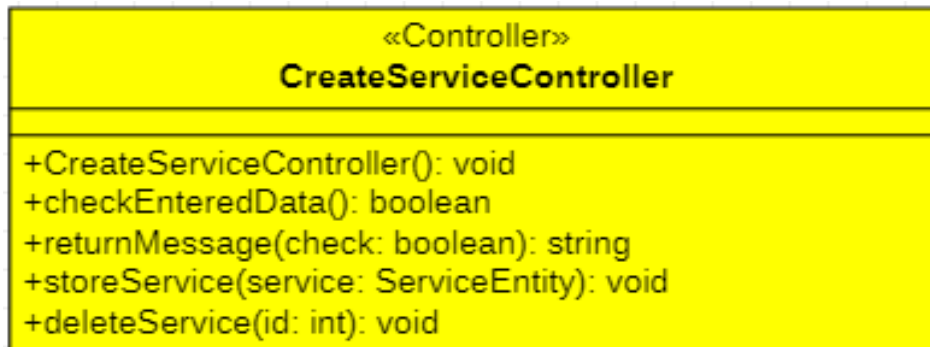
### Μέθοδοι κλάσης:

- *Service(serviceType : string, petType : string, petSize : string, days : ArrayList<string>, time1 : date, time2 : date, range : float)* : Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *getServiceType()*: Η μέθοδος επιστρέφει το είδος της υπηρεσίας
- *setServiceType(serviceType:string)*: Η μέθοδος αυτή θέτει το είδος της υπηρεσίας
- *getPetType()*: Η μέθοδος επιστρέφει το είδος του κατοικιδίου
- *setPetType(petType : string)*: Η μέθοδος αυτή θέτει το είδος του κατοικιδίου
- *getPetSize()*: Η μέθοδος επιστρέφει το μέγεθος του κατοικιδίου
- *setPetSize(petSize : string)*: Η μέθοδος αυτή θέτει το μέγεθος του κατοικιδίου
- *getDays()*: Η μέθοδος επιστρέφει τις επιλεγμένες μέρες της υπηρεσίας
- *setDays(days : ArrayList<String>)*: Η μέθοδος αυτή θέτει τις επιλεγμένες μέρες της υπηρεσίας
- *getId()*: Η μέθοδος επιστρέφει το μοναδικό αριθμό της υπηρεσίας



- *setId(id : int)*: Η μέθοδος αυτή θέτει το μοναδικό αριθμό της υπηρεσίας
- *getPrice()*: Η μέθοδος επιστρέφει το κόστος της υπηρεσίας
- *setPrice(price : float)*: Η μέθοδος αυτή θέτει το κόστος της υπηρεσίας
- *getProvider()*: Η μέθοδος επιστρέφει αντικείμενο *userEntity*, του provider που δημιούργησε την υπηρεσία.
- *setProvider(provider : UserEntity)*: Η μέθοδος αυτή θέτει αντικείμενο *userEntity*, του provider που δημιούργησε την υπηρεσία

### **Controller CreateServiceController:**

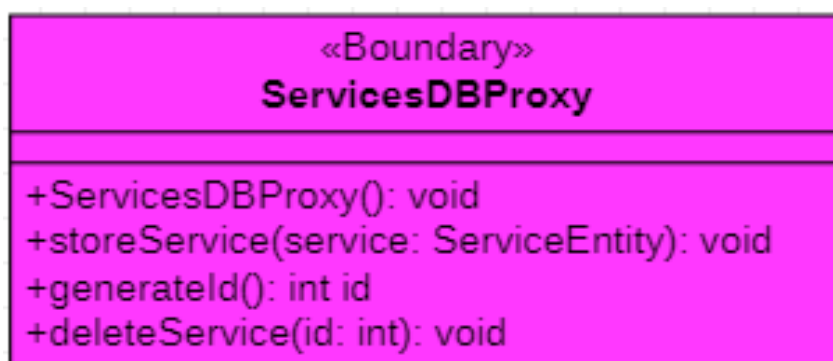


Η κλάση αυτή είναι ο ελεγκτής που περιέχει τις απαραίτητες συναρτήσεις για τη δημιουργία μιας υπηρεσίας από τον provider.

### **Μέθοδοι κλάσης:**

- *CreateServiceController()*: Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *checkEnteredData()* : Μέθοδος που ελέγχει αν έχουν συμπληρωθεί όλα τα πεδία με το σωστό τρόπο, και επιστρέφει true, όταν είναι σωστά, αλλιώς false
- *returnMessage(check : boolean)* : Μέθοδος που εμφανίζει το αντίστοιχο μήνυμα στο χρήστη, σχετικά με την επιτυχία(true) ή μη(false), δημοσίευση της υπηρεσίας του
- *storeService(service : ServiceEntity)* : Μέθοδος που καλεί την μέθοδο *storeService(service : ServiceEntity)* του *ServicesDBProxy*, για την αποθήκευση υπηρεσίας στη βάση δεδομένων
- *deleteService(id : int)* : Μέθοδος που καλεί την μέθοδο *deleteService(id: int)* του *ServicesDBProxy*, για τη διαγραφή υπηρεσίας από τη βάση δεδομένων

### **Boundary ServicesDBProxy:**



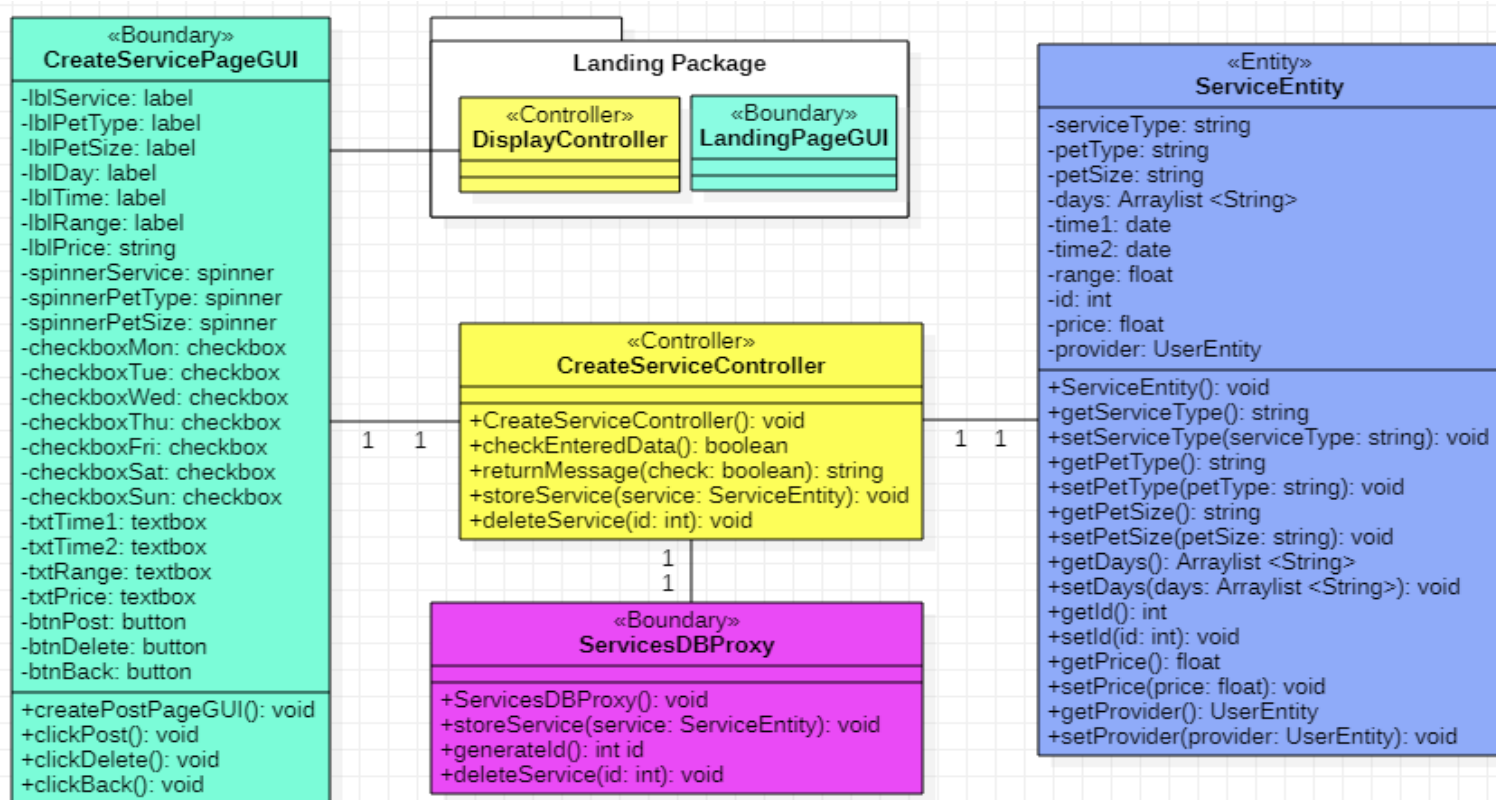


Η κλάση αυτή είναι η κλάση επικοινωνίας με τη βάση δεδομένων ServicesDB, ώστε να είναι δυνατή η ανάκτηση, αποθήκευση και ενημέρωση των δεδομένων της.

### Μέθοδοι κλάσης:

- *ServicesDBProxy()*: Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *storeService(service : ServiceEntity)*: Μέθοδος αποθήκευσης αντικειμένου τύπου ServiceEntity, στη βάση δεδομένων των υπηρεσιών
- *generateId()* : Μέθοδος δημιουργίας μοναδικού αριθμού για κάθε υπηρεσία
- *deleteService(id : int)* : Μέθοδος διαγραφής υπάρχουσας υπηρεσίας, με id αυτό του ορίσματος της μεθόδου.

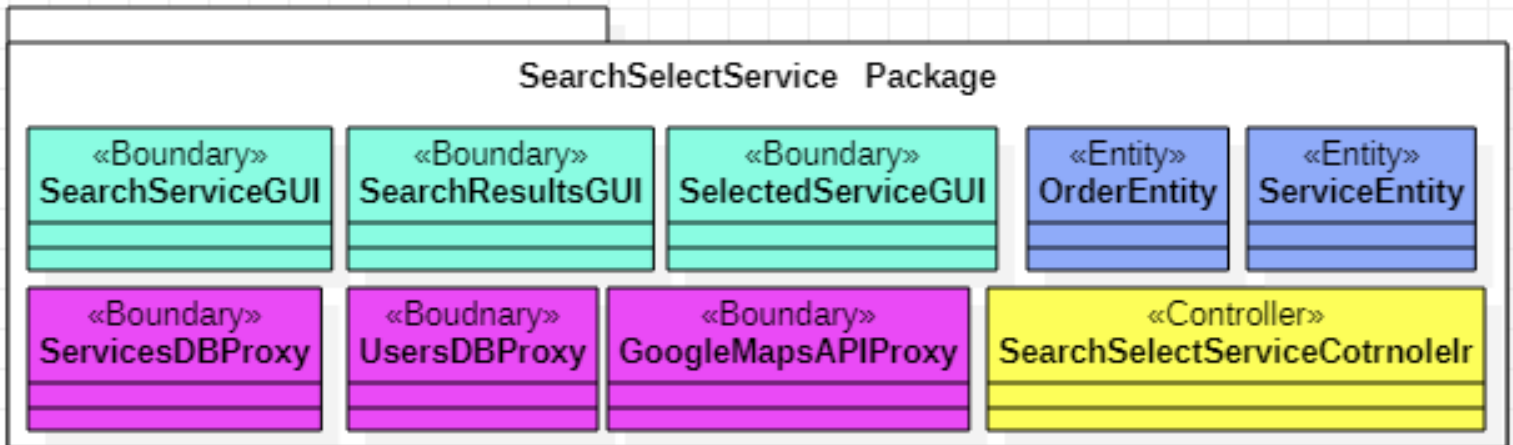
### Διάγραμμα κλάσεων:



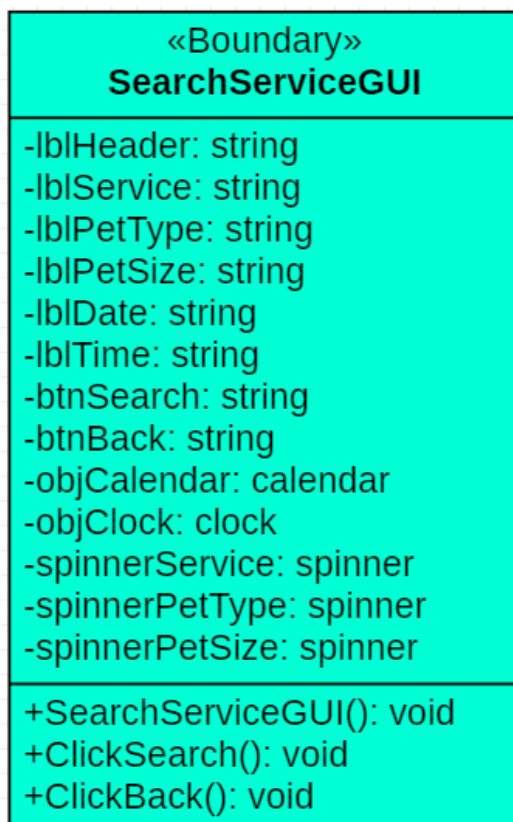


#### 4.1.6 Πακέτο SearchSelectService Package

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που έχουν σχέση με αναζήτηση, εμφάνιση και επιλογή των διαθέσιμων υπηρεσιών από τον owner.



##### Boundary SearchServiceGUI:



Η κλάση αυτή εκφράζει τη διεπαφή της σελίδας δημιουργίας υπηρεσιών του Provider.

##### Χαρακτηριστικά κλάσης:

- *lblHeader*: Επιγραφή «SEARCH FOR A SERVICE»
- *lblService*: Επιγραφή «SERVICE»
- *lblPetType*: Επιγραφή «PET TYPE»

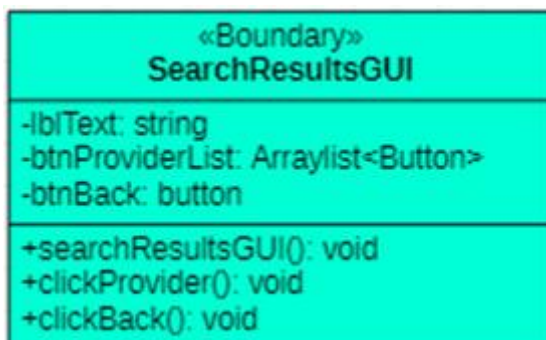


- *lblPetSize*: Επιγραφή «PET SIZE»
- *lblDate*: Επιγραφή «DATE»
- *lblTime*: Επιγραφή «TIME»
- *spinnerService*: DropDown λίστα, με τις επιλογές παροχής υπηρεσίας
- *spinnerPetType*: DropDown λίστα, με τις επιλογές ειδών κατοικιδίων
- *spinnerPetSize*: DropDown λίστα, με τις επιλογές μεγέθους κατοικιδίων
- *objCalendar*: Αντικείμενο calendar
- *objClock*: Αντικείμενο clock
- *btnSearch*: Κουμπί εκκίνησης της διαδικασίας αναζήτησης υπηρεσιών
- *btnBack*: Κουμπί μετάβασης στην προηγούμενη σελίδα, δηλαδή στο HomePage

### **Μέθοδοι κλάσης:**

- *SearchServiceGUI()* : Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *clickSearch()* : Μέθοδος που καλεί, πρώτα την *checkEnteredData()* : boolean, και αν επιστραφεί true, καλεί *checkRange(addressOwner : string) : ArrayList<int>*, η οποία επιστρέφει ένα ArrayList από ακεραίους, που αντιπροσωπεύουν τα ids των ServiceEntities που θα εμφανιστούν.
- *clickBack()* : Μέθοδος που καλεί την *DisplayHomePage(category : string) : void*, του DisplayController

### **Boundary SearchResultsGUI:**



Η κλάση αυτή εκφράζει τη διεπαφή της σελίδας προβολής αποτελεσμάτων της αναζήτησης Provider στον χρήστη Owner.

### **Χαρακτηριστικά κλάσης:**

- *lblText*: Επιγραφή «Available Providers»
- *btnProviderList*: Κουμπιά-λίστα των διαθέσιμων Provider.
- *btnBack*: Κουμπί μετάβασης στην προηγούμενη σελίδα, δηλαδή στο SearchServicePage

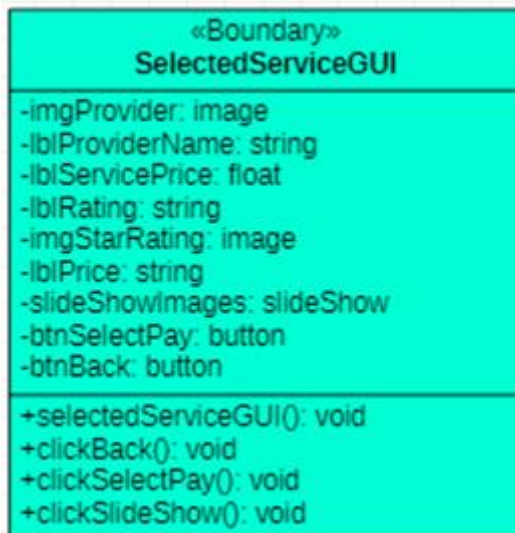
### **Μέθοδοι κλάσης:**

- *SearchResultsGUI()*: Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *clickProvider()*: Μέθοδος που εμφανίζει τα στοιχεία της υπηρεσίας που επιλέγουμε, καλώντας πρώτα την *saveSelectedService(service : ServiceEntity) : void* και μετά την *displaySelectedServicePage():void*
- *clickBack()*: Μέθοδος που καλεί την *DisplaySearchProvider() : void*, του DisplayController





### **Boundary SelectedServiceGUI:**



Η κλάση αυτή εκφράζει τη διεπαφή της σελίδας προβολής της επιλεγμένης υπηρεσίας, στον Owner

### **Χαρακτηριστικά κλάσης:**

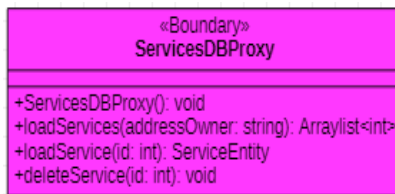
- *imgProvider*: Εικόνα προφίλ του χρήστη Provider.
- *lblProviderName*: Επιγραφή του ονόματος του χρήστη Provider.
- *lblServicePrice*: Επιγραφή της τιμής για τη συγκεκριμένη υπηρεσία.
- *lblRating*: Επιγραφή «Rating:»
- *imgStarRating*: Εικόνα εμφάνισης της βαθμολογίας του χρήστη Provider
- *lblPrice*: Επιγραφή τιμής για την υπηρεσία.
- *slideShowImages*: Slide show με εικόνες του χρήστη Provider
- *btnSelectPay*: Κουμπί μετάβασης στο περιβάλλον της τράπεζας για πληρωμή της υπηρεσίας
- *btnBack*: Κουμπί μετάβασης στην προηγούμενη σελίδα, δηλαδή στο SearchResultsPage

### **Μέθοδοι κλάσης:**

- *selectedServiceGUI()* : Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *clickBack()* : Μέθοδος που καλεί την DisplaySearchResultsPage() : void, του DisplayController
- *clickSelectPay()* : Μέθοδος που καλεί αρχικά την sendOrder(order : OrderEntity) : void, μετά την loadBankEnvironment(ProviderEmail : string, Price : float) : void από το Pay Package και τέλος τις deleteService(id : int) από την κλάση ServiceDBProxy και clearPostedServiceId(email : string) από την κλάση UsersDBProxy
- *clickSlideShow()* : Μέθοδος που διαχειρίζεται την εναλλαγή των εικόνων του Provider, είτε αυτόματα είτε με “Click”, που έχει δημιουργήσει την υπηρεσία μέσω της κλήσης της συνάρτησης slideshowActionListener()



### **Boundary ServicesDBProxy:**

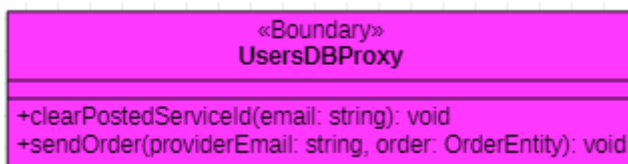


Η κλάση αυτή είναι η διεπαφή με την βάση δεδομένων των υπηρεσιών.

### **Μέθοδοι κλάσης:**

- *ServicesDBProxy()* : Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *loadServices(addressOwner : string)* : Μέθοδος που παίρνει σαν όρισμα τη διεύθυνση του Owner και καλεί την *checkRange(range float, addressOwner : string, addressProvider : string)* του *GoogleMapsAPIProxy*. Η συνάρτηση επιστρέφει ένα *ArrayList* ακεραίων με τα *id* των υπηρεσιών που επαληθεύουν τις παραμέτρους αναζήτησης.
- *loadService(id : int)* : Μέθοδος που επιστρέφει αντικείμενο τύπου *ServiceEntity*, παίρνοντας σαν όρισμα το *id* του *Entity* αυτού.
- *deleteService(id : int)* : Μέθοδος που διαγράφει αντικείμενο τύπου *ServiceEntity*, παίρνοντας σαν όρισμα το *id* του *Entity* αυτού.

### **Boundary UsersDBProxy:**

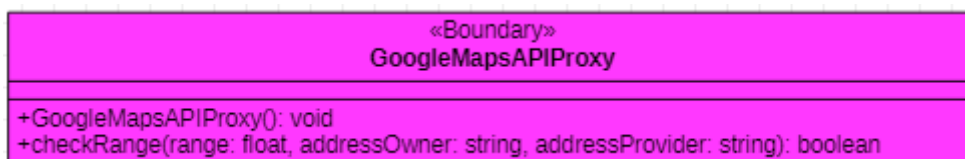


Η κλάση αυτή είναι η διεπαφή με την βάση δεδομένων των χρηστών.

### **Μέθοδοι κλάσης:**

- *clearPostedServiceId(email : string)* : Μέθοδος που διαγράφει το περιεχόμενο της μεταβλητής *PostedServiceId* του *Provider* που αντιστοιχεί στο *email* του ορίσματος.
- *sendOrder(providerEmail : string, order : OrderEntity)* : Μέθοδος που αρχικοποιεί την μεταβλητή *order* του *User* που αντιστοιχεί στο *email* του ορίσματος *providerEmail*, με ένα αντικείμενο τύπου *OrderEntity*.

### **Boundary GoogleMapsAPIProxy:**



Η κλάση αυτή είναι η διεπαφή με το API των χαρτών της Google.

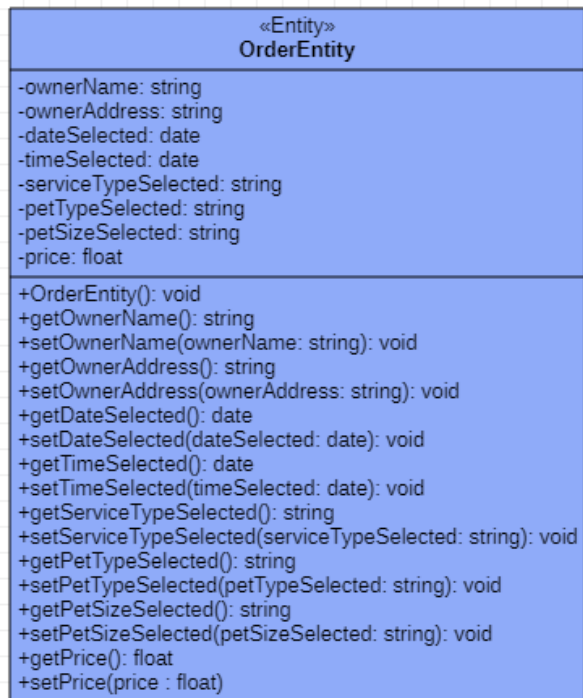




### Μέθοδοι κλάσης:

- *GoogleMapsAPIProxy()* : Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *checkRange(range : float, addressOwner : string, addressProvider : string)* : Μέθοδος που δέχεται σαν ορίσματα μια απόσταση range τύπου float και τις διευθύνσεις των Owner και Provider. Μέσω του API googleMaps, θα υπολογίσει αν οι 2 διευθύνσεις απέχουν λιγότερο από το range, και αν ναι επιστρέφει true, αλλιώς false.

### Entity OrderEntity:



Η κλάση δημιουργίας των αντικειμένων τύπου OrderEntity.

### Χαρακτηριστικά κλάσης:

- *ownerName* : Το όνομα του Owner που πλήρωσε για την υπηρεσία
- *ownerAddress* : Η διεύθυνση του Owner που πλήρωσε για την υπηρεσία
- *dateSelected* : Η ημερομηνία που επιλέχθηκε από τον Owner για την εκτέλεση της υπηρεσίας
- *timeSelected* : Η ώρα που επιλέχθηκε από τον Owner για την εκτέλεση της υπηρεσίας
- *serviceTypeSelected* : Το είδος της υπηρεσίας που έχει επιλεγεί
- *petTypeSelected* : Το είδος του κατοικιδίου που έχει επιλεγεί
- *petSizeSelected* : Το μέγεθος του κατοικιδίου που έχει επιλεγεί
- *price* : Το κόστος της υπηρεσίας

### Μέθοδοι κλάσης:

- *OrderEntity()*: Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *getOwnerName()*: Η μέθοδος αυτή επιστρέφει το όνομα του Owner που πλήρωσε για την υπηρεσία

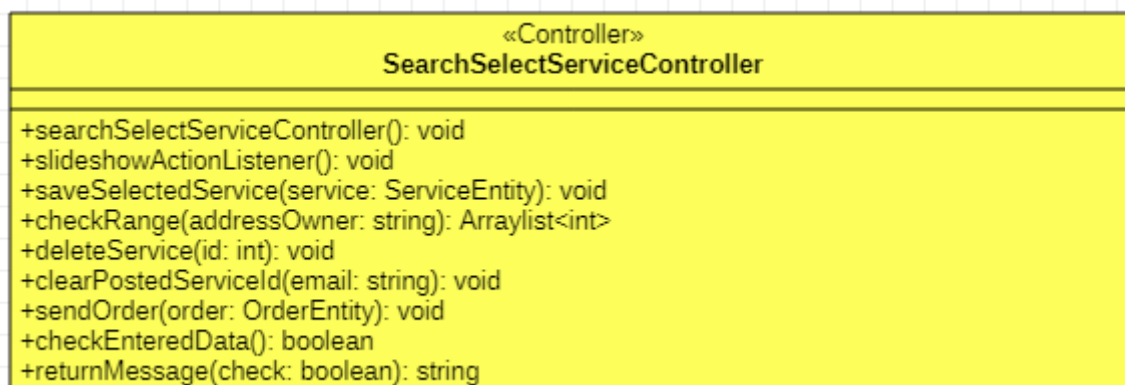


- *setOwnerName(ownerName : string)*: Η μέθοδος αυτή θέτει το όνομα του Owner που πλήρωσε για την υπηρεσία
- *getOwnerAddress()*: Η μέθοδος αυτή επιστρέφει τη διεύθυνση του Owner που πλήρωσε για την υπηρεσία
- *setOwnerAddress(ownerAddress : string)*: Η μέθοδος αυτή θέτει τη διεύθυνση του Owner που πλήρωσε για την υπηρεσία
- *getDateSelected()*: Η μέθοδος αυτή επιστρέφει την ημερομηνία που θα εκτελεστεί η υπηρεσία
- *setDateSelected(dateSelected : date)*: Η μέθοδος αυτή θέτει την ημερομηνία που θα εκτελεστεί η υπηρεσία
- *getTimeSelected()*: Η μέθοδος αυτή επιστρέφει την ώρα που θα εκτελεστεί η υπηρεσία
- *setTimeSelected(timeSelected : date)*: Η μέθοδος αυτή θέτει την ώρα που θα εκτελεστεί η υπηρεσία
- *getServiceTypeSelected()*: Η μέθοδος αυτή επιστρέφει το είδος της υπηρεσίας
- *setServiceTypeSelected(serviceTypeSelected : string)*: Η μέθοδος αυτή θέτει το είδος της υπηρεσίας
- *getPetTypeSelected()*: Η μέθοδος αυτή επιστρέφει το είδος του κατοικιδίου της επιλεγμένης υπηρεσίας
- *setPetTypeSelected(petTypeSelected : string)*: Η μέθοδος αυτή θέτει το είδος του κατοικιδίου της επιλεγμένης υπηρεσίας
- *getPetSizeSelected()*: Η μέθοδος αυτή επιστρέφει το μέγεθος του κατοικιδίου της επιλεγμένης υπηρεσίας
- *setPetSizeSelected(petSizeSelected : string)*: Η μέθοδος αυτή θέτει το μέγεθος της επιλεγμένης υπηρεσίας
- *getPrice()*: Η μέθοδος αυτή επιστρέφει το κόστος της επιλεγμένης υπηρεσίας
- *setPrice(price : float)*: Η μέθοδος αυτή θέτει το κόστος της επιλεγμένης υπηρεσίας

### **Entity ServiceEntity:**

Η κλάση αυτή έχει περιγραφεί στο CreateService Package.

### **Controller SearchSelectServiceController:**



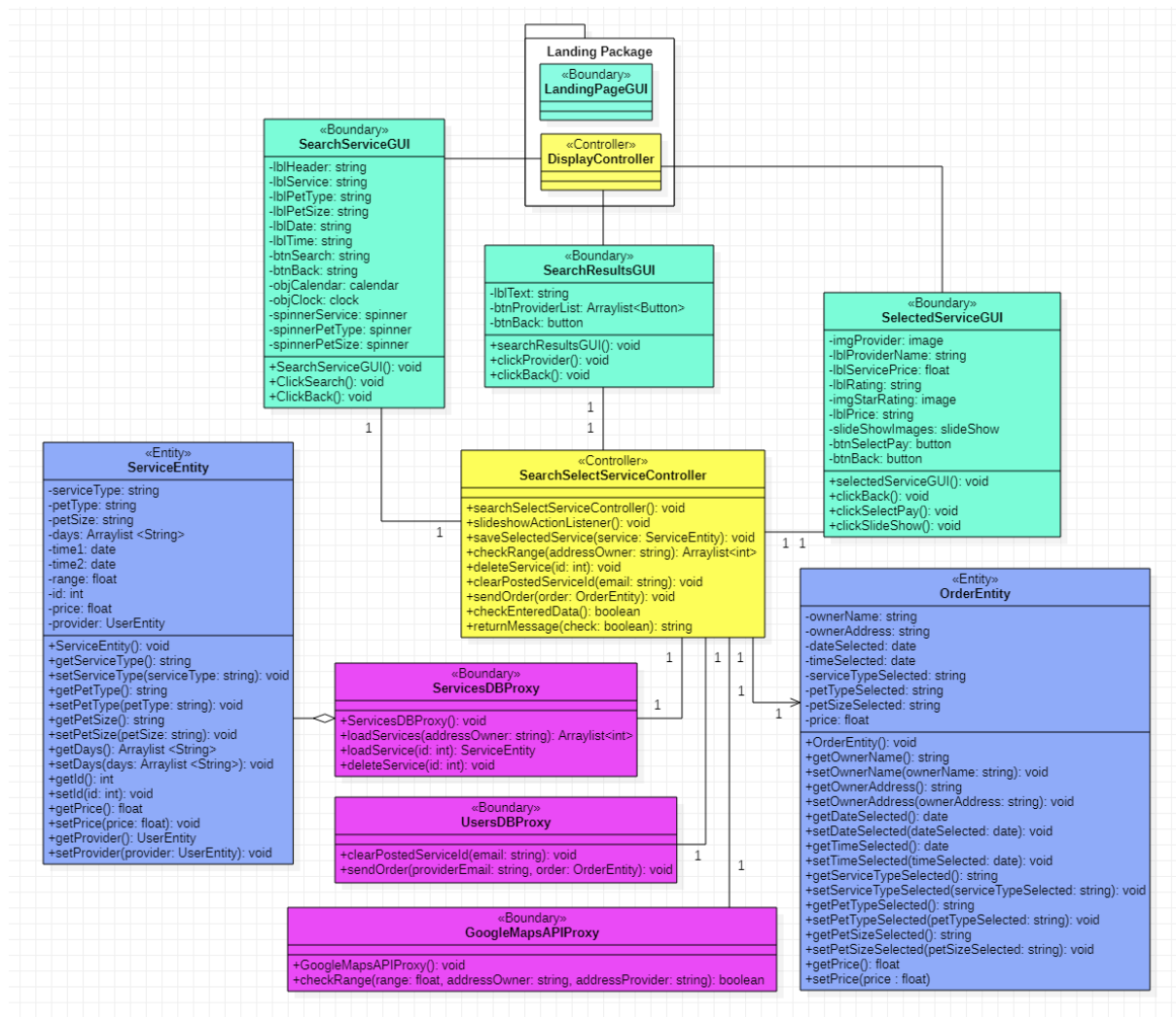
Η κλάση αυτή είναι ο ελεγκτής που περιέχει τις απαραίτητες συναρτήσεις για την αναζήτηση, εμφάνιση και επιλογή των υπηρεσιών από τον Owner.

**Μέθοδοι κλάσης:**

- *searchSelectServiceController()* : Μέθοδος δημιουργίας των αντικειμένων της κλάσης (constructor)
- *slideshowActionListener()* : Μέθοδος λειτουργίας του slideshow για την εναλλαγή εικόνων.
- *saveSelectedService(service : ServiceEntity)* : Μέθοδος για αποθήκευση της επιλεγμένης υπηρεσίας στο σύστημα για εσωτερική χρήση
- *checkRange(addressOwner : string)* : Μέθοδος που καλεί τη συνάρτηση *loadServices(address : string)* του *ServiceDBProxy*, και επιστρέφει ένα *ArrayList* ακεραίων με τα ids των διαθέσιμων υπηρεσιών.
- *deleteService(id : int)* : Μέθοδος που καλεί την *deleteService(id : int)* από το *ServiceDBProxy*
- *clearPostedServiceId(email : string)* : Μέθοδος που καλεί τη συνάρτηση *clearPostedServiceId(email : string)* από το *UsersDBProxy*.
- *sendOrder(order : OrderEntity)* : Μέθοδος που καλεί τη συνάρτηση *sendOrder(providerEmail : string, order : OrderEntity)* από το *UsersDBProxy*.
- *checkEnteredData()* : Μέθοδος που ελέγχει αν έχουν συμπληρωθεί όλα τα πεδία με το σωστό τρόπο, και επιστρέφει *true*, όταν είναι σωστά, αλλιώς *false*, και καλεί την συνάρτηση *returnMessage(check : Boolean)*.
- *returnMessage(check : Boolean)* : Μέθοδος που εμφανίζει το αντίστοιχο μήνυμα στο χρήστη, σχετικά με την επιτυχία(*true*) ή μη(*false*), αναζήτηση υπηρεσίας

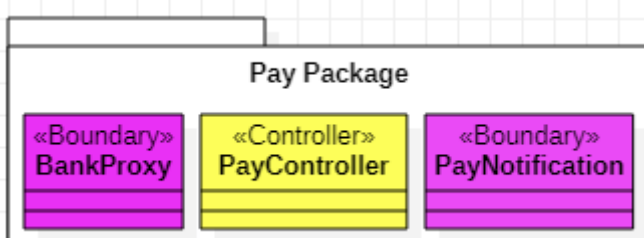


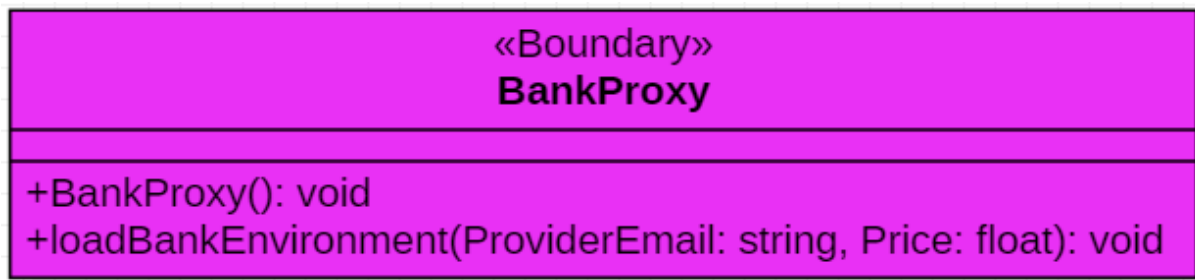
### Διάγραμμα κλάσεων:



#### 4.1.7 Πακέτο Pay Package:

Σκοπός αυτού του πακέτου είναι η διαχείριση των πληρωμών υπηρεσιών στους Providers από τους Owners καθώς και οι ηχητικές ειδοποιήσεις που αποστέλλονται στον Owner και αφορούν την επιτυχή πληρωμή της υπηρεσίας.

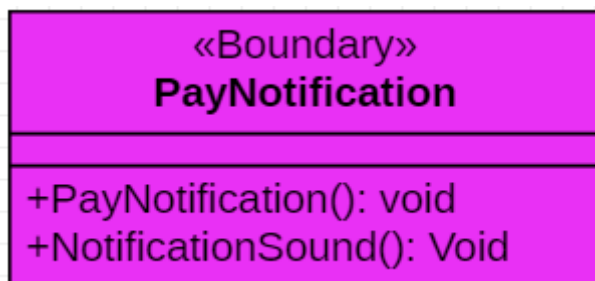


**Boundary BankProxy:**

Η κλάση αυτή επικοινωνεί με το περιβάλλον της τράπεζας και χρησιμοποιείται για να πραγματοποιηθεί μια πληρωμή υπηρεσίας.

**Μέθοδοι κλάσης:**

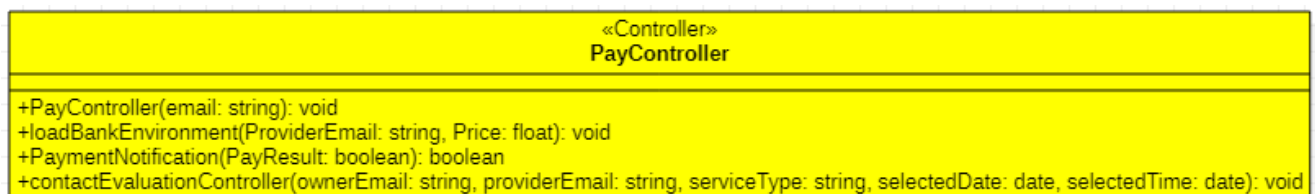
- *BankProxy(): void* : Είναι η συνάρτηση δόμησης της κλάσης.
- *loadBankEnvironment(ProviderEmail:string, Price:float):void* : Η μέθοδος αυτή επιστρέφει το email του Provider και την τιμή της υπηρεσίας στο περιβάλλον της τράπεζας για την πληρωμή αυτής.

**Boundary PayNotification:**

Η κλάση αυτή επικοινωνεί μέσω του ελεγκτή PayController με το περιβάλλον της τράπεζας και ειδοποιεί ηχητικά τον χρήστη μετά την επιτυχή πληρωμή της υπηρεσίας.

**Μέθοδοι κλάσης:**

- *PayNotification():* Είναι η συνάρτηση δόμησης της κλάσης.
- *NotificationSound():* Η μέθοδος αυτή χρησιμοποιείται για την ηχητική ειδοποίηση του χρήστη μετά από επιτυχή πληρωμή.

**Controller PayController:**

Ο ελεγκτής αυτός περιέχει όλες τις κατάλληλες συναρτήσεις για την διαδικασία πληρωμής μιας υπηρεσίας.

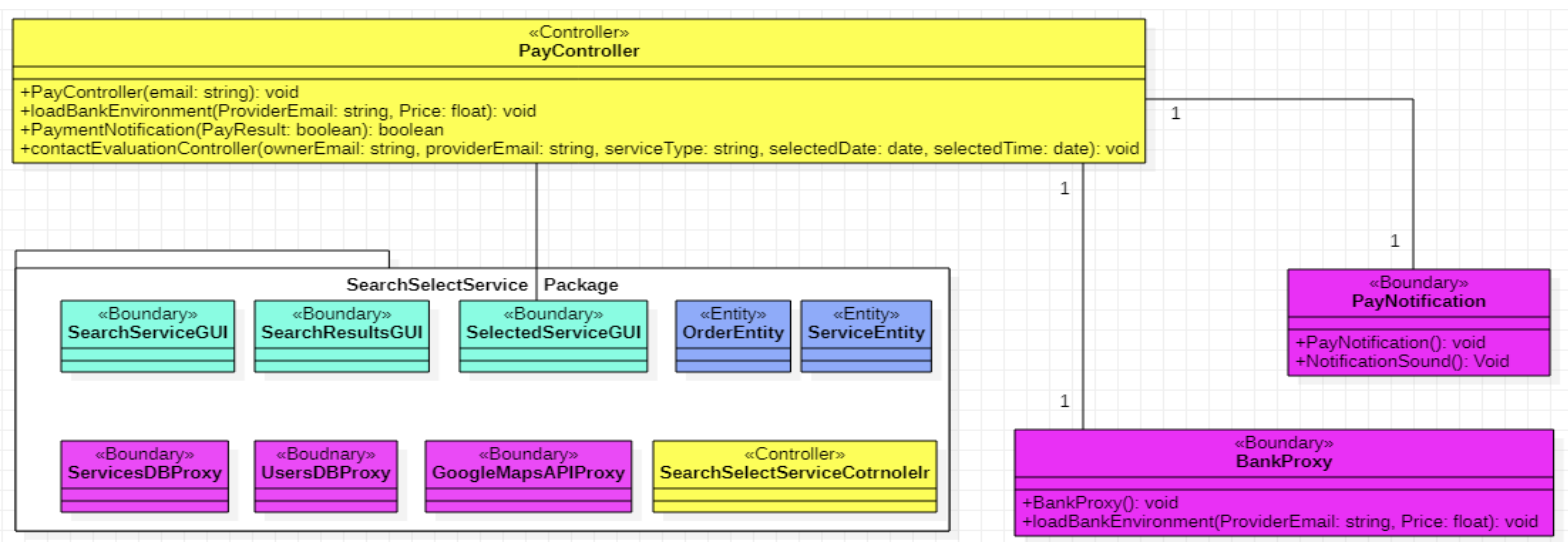
**Μέθοδοι κλάσης:**

- *PayController(email)* : Αποτελεί συνάρτηση δόμησης του ελεγκτή.



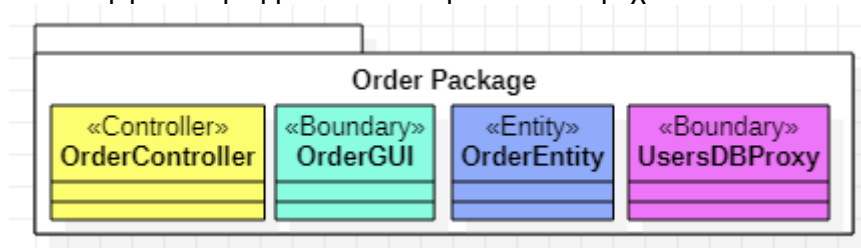
- *loadBankEnvironment(ProviderEmail:string, Price:float)*: Καλεί την αντίστοιχη συνάρτηση φόρτωσης του περιβάλλοντος τράπεζας στο BankProxy, με σκοπό την πληρωμή υπηρεσίας.
- *PaymentNotification(PayResult:boolean)*: Η μέθοδος αυτή ελέγχει αν έχει γίνει η πληρωμή επιτυχώς.
- *contactEvaluationController(ownerEmail:string, providerEmail:string, serviceType:string, currentDate:date, currentTime:date)*: Καλείται από τη συνάρτηση clickSelectPay() του SelectedServiceGUI. Καλεί τη συνάρτηση createEvaluation(ownerEmail: string, providerEmail: string, serviceType: string, selectedDate: date, selectedTime: date) του EvaluationController.

### Διάγραμμα κλάσεων:



#### 4.1.8 Πακέτο Order Package:

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που χρειάζονται για την προβολή και επεξεργασία των ενεργών παραγγελιών που πρέπει να παρέχει ο Provider.





### **Boundary OrderGUI:**

«Boundary» OrderGUI
-lblHeader: string -lblOwnerName: string -lblOwnerAddress: string -lblDateSelected: string -lblTimeSelected: string -lblServiceType: string -lblPetType: string -lblPetSize: string -lblPrice: string -txtOwnerName: textbox -txtOwnerAddress: textbox -txtDateSelected: textbox -txtTimeSelected: textbox -txtServiceType: textbox -txtPetType: textbox -txtPetSize: textbox -txtPrice: textbox -btnDelete: button -btnBack: button
+OrderGUI(): void +clickDelete(): void +clickBack(): void

Η κλάση αυτή αναπαριστά τη διεπαφή της σελίδας προβολής του order.

### **Χαρακτηριστικά κλάσης:**

- *lblHeader*: Επιγραφή 'Active Order'.
- *lblOwnerName*: Επιγραφή 'Owner Name'.
- *lblOwnerAddress*: Επιγραφή 'Owner Address'.
- *lblDateSelected*: Επιγραφή 'Date Selected'.
- *lblTimeSelected*: Επιγραφή 'Time Selected'.
- *lblServiceType*: Επιγραφή 'Service Type'.
- *lblPetType*: Επιγραφή 'Pet Type'.
- *lblPetSize*: Επιγραφή 'Pet Size'.
- *lblPrice*: Επιγραφή 'Price'.
- *txtOwnerName*: Πεδίο προβολής ονόματος owner.
- *txtOwnerAddress*: Πεδίο προβολής διεύθυνσης owner.
- *txtDateSelected*: Πεδίο προβολής επιλεγμένης ημερομηνίας από τον owner.
- *txtTimeSelected*: Πεδίο προβολής επιλεγμένης ώρας από τον owner.
- *txtServiceType*: Πεδίο προβολής είδους υπηρεσίας επιλεγμένη από τον owner.
- *txtPetType*: Πεδίο προβολής είδους κατοικιδίου επιλεγμένο από τον owner.
- *txtPetSize*: Πεδίο προβολής μεγέθους κατοικιδίου επιλεγμένο από τον owner.
- *txtPrice*: Πεδίο προβολής κόστους της υπηρεσίας.
- *btnDelete*: Κουμπί διαγραφής των στοιχείων της παραγγελίας.
- *btnBack*: Κουμπί επιστροφής στο home page.

### **Μέθοδοι Κλάσης:**

- *OrderGUI()*: Μέθοδος δόμησης της κλάσης.
- *clickDelete()*: Καλεί τη συνάρτηση `deleteOrder(order:OrderEntity)` του ελεγκτή `OrderController` ώστε να γίνει η διαγραφή των στοιχείων του τρέχον order.



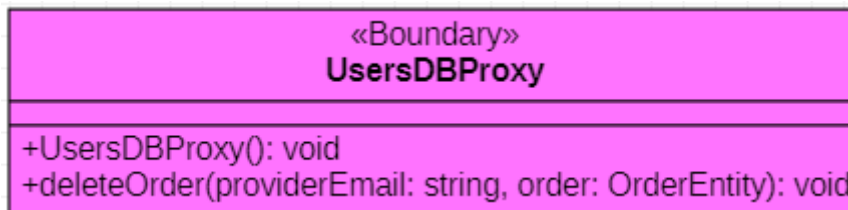


- *clickBack()*: Καλεί τη συνάρτηση *DisplayHomePage(category:string)* του ελεγκτή *DisplayController*.

### Entity OrderEntity:

Έχει περιγραφεί στο *SearchSelectService Package*.

### Boundary UsersBDProxy:

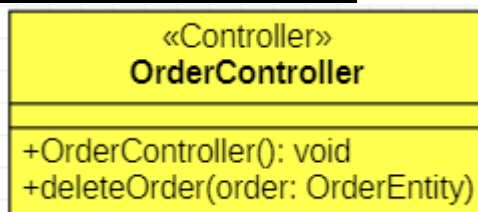


Η κλάση αυτή είναι υπεύθυνη για τη διαγραφή της παραγγελίας από το *OrderEntity* ενός *ProviderEntity* από τη βάση δεδομένων *UsersDB*.

### Μέθοδοι Κλάσης:

- *UsersDBProxy()*: Μέθοδος δόμησης της κλάσης
- *deleteOrder(providerEmail:string, order:OrderEntity)*: Διαγράφει τα περιεχόμενα της μεταβλητής *order* με βάση το δοσμένο *email provider*.

### Cotroller OrderController:

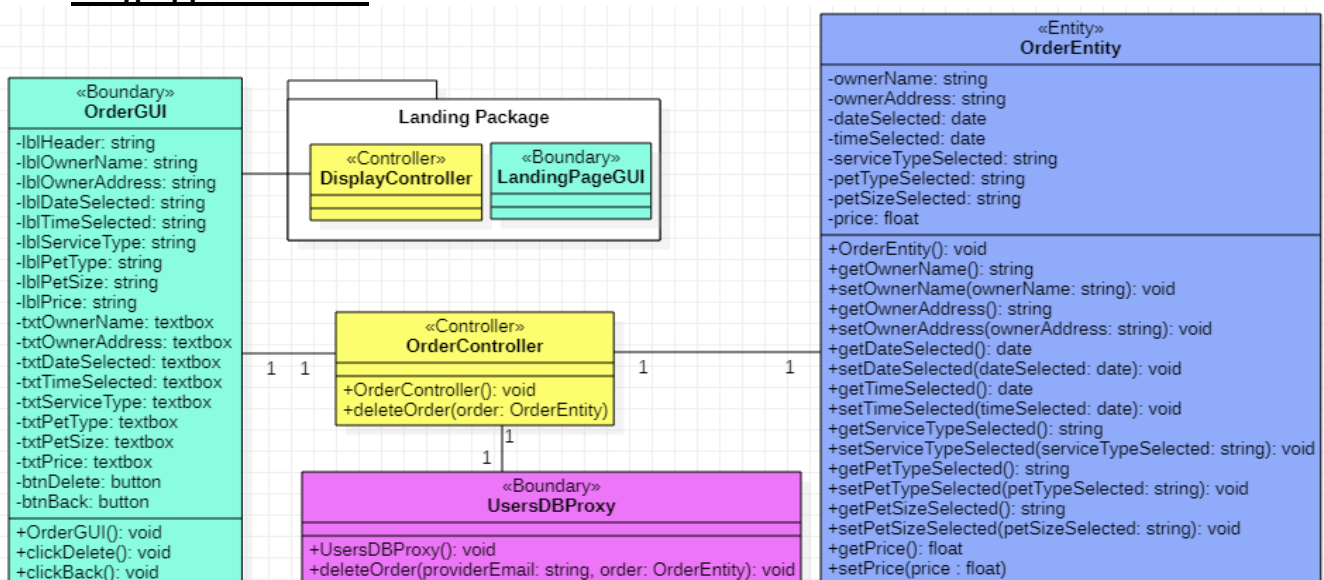


Η κλάση αυτή είναι ο ελεγκτής που περιέχει τις απαραίτητες συναρτήσεις για τη διαγραφή του *OrderEntity* ενός *provider*.

### Μέθοδοι Κλάσης:

- *OrderController()*: Μέθοδος δόμησης της κλάσης
- *deleteOrder(order:OrderEntity)*: Καλεί τη συνάρτηση *deleteOrder(providerEmail:string, order:OrderEntity)* του *UsersDBProxy*.

### Διάγραμμα κλάσεων:



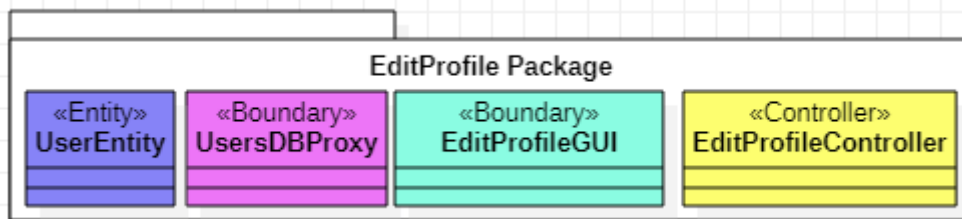




## 4.2 <Πακέτο(α) λεξιλογίου σεναρίων μέσης προτεραιότητας>

### 4.2.1 Πακέτο Profile Package:

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που σχετίζονται με την τροποποίηση των προσωπικών στοιχείων του χρήστη.



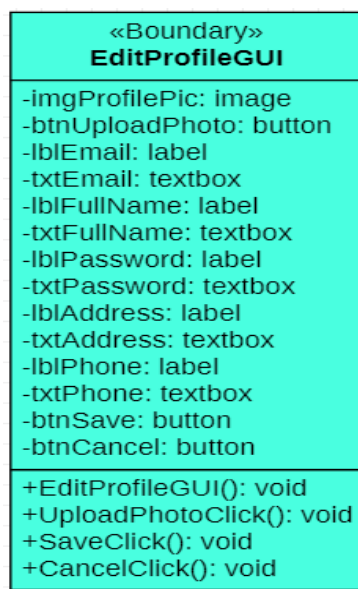
#### Entity UserEntity:

Έχει περιγραφεί στο πακέτο Login Package.

#### Boundary UsersDBProxy:

Έχει περιγραφεί στα πακέτα Signup Package και Login Package.

#### Boundary EditProfileGUI:



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση της διεπαφής της επεξεργασίας των προσωπικών στοιχείων του χρήστη.

#### Χαρακτηριστικά κλάσης:

- *imgProfilePic*: Η φωτογραφία Profile του χρήστη.
- *btnUploadPhoto*: Κουμπί για προσθήκη φωτογραφίας.
- *lblEmail*: Επιγραφή για το email του χρήστη.
- *txtEmail*: Πεδίο για την εισαγωγή του email επικοινωνίας του χρήστη.
- *lblFullName*: Επιγραφή για το ονοματεπώνυμο του χρήστη.
- *txtFullName*: Πεδίο για την εισαγωγή του ονοματεπώνυμου του χρήστη.
- *lblPassword*: Επιγραφή για τον κωδικό του χρήστη.



- *txtPassword*: Πεδίο για την εισαγωγή του κωδικού του χρήστη.
- *lblAddress*: Επιγραφή για την διεύθυνση του χρήστη.
- *txtAddress*: Πεδίο για την εισαγωγή της διεύθυνσης του χρήστη.
- *lblPhone*: Επιγραφή για το τηλέφωνο του χρήστη.
- *txtPhone*: Πεδίο για την εισαγωγή του τηλεφώνου του χρήστη.
- *btnSave*: Κουμπί για υποβολή των νέων προσωπικών στοιχείων του χρήστη.
- *btnCancel*: Κουμπί για την επιστροφή στην προηγούμενη σελίδα της εφαρμογής.

#### Μέθοδοι κλάσης:

- *EditProfileGUI()*: Αποτελεί συνάρτηση δόμησης της διεπαφής αυτής.
- *UploadPhotoClick()*: Η συνάρτηση αυτή επιτρέπει την μεταφόρτωση φωτογραφιών.
- *SaveClick()*: Η μέθοδος αυτή καλεί την συνάρτηση *storeUser(User: UserEntity)* του ελεγκτή *EditProfileController*, έτσι ώστε να αποθηκευτεί η νέα καταχώρηση και να γίνει επιστροφή στη σελίδα προβολής των προσωπικών στοιχείων του χρήστη.
- *CancelClick()*: Καλεί τη συνάρτηση *DisplayHomePage(category: string)* του ελεγκτή *DisplayController*, ώστε να γίνει προβολή της αρχικής σελίδας της εφαρμογής.

#### Controller EditProfileController:

«Controller» <b>EditProfileController</b>
<b>+EditProfileController(): void</b> <b>+storeUser(User: UserEntity): void</b> <b>+loadUser(email: string): UserEntity</b> <b>+editProfile(email: string): boolean</b> <b>+checkEnteredData(): boolean</b> <b>+returnError(EnteredDataFlag: boolean): string</b> <b>+Cancel(): void</b>

Ο ελεγκτής αυτός περιέχει τις κατάλληλες συναρτήσεις-μεθόδους για την τροποποίηση των προσωπικών στοιχείων ενός χρήστη.

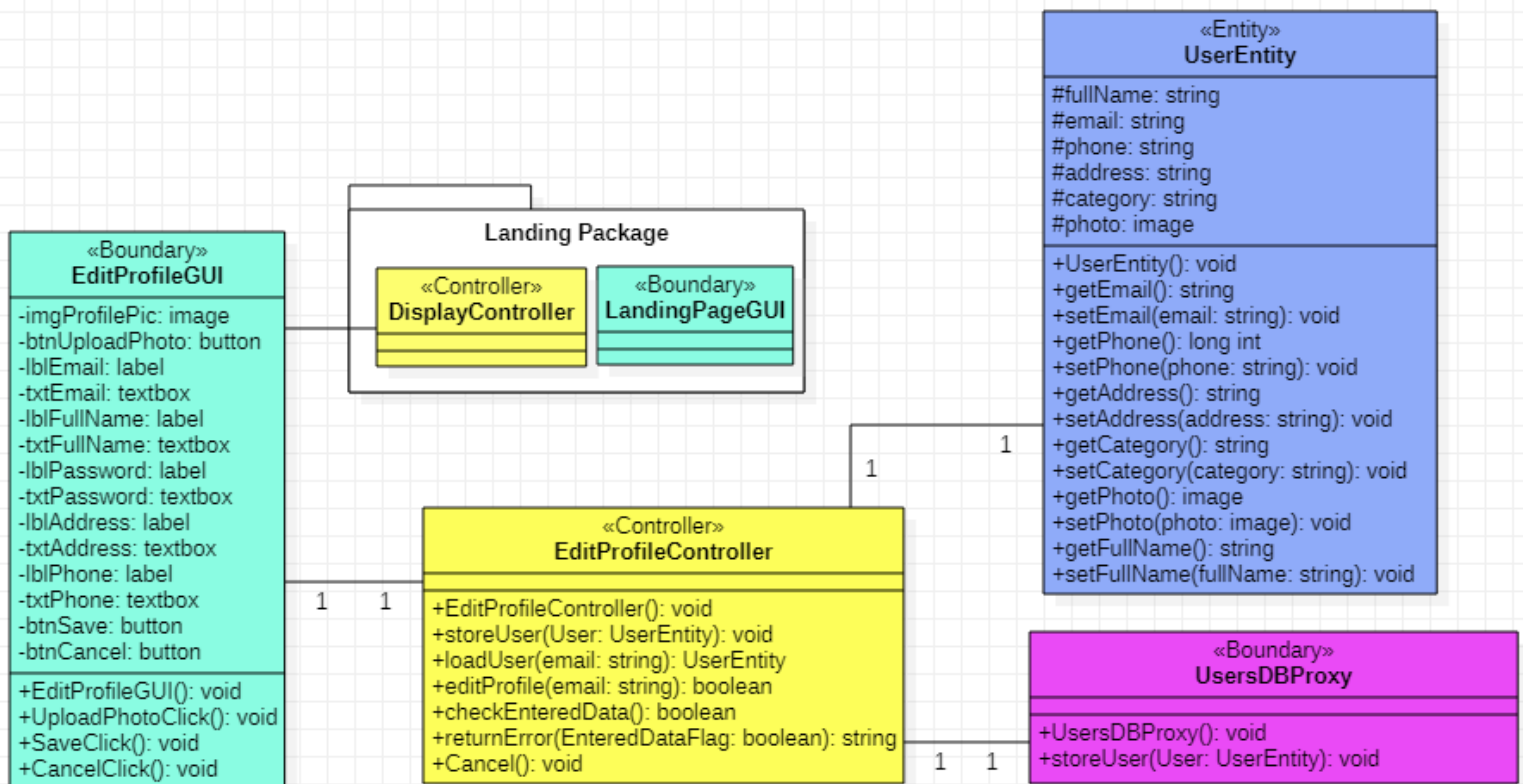
#### Μέθοδοι κλάσης:

- *EditProfileController*: Αποτελεί συνάρτηση δόμησης του ελεγκτή.
- *storeUser(User: UserEntity)*: Η μέθοδος αυτή καλεί την αντίστοιχη συνάρτηση αποθήκευσης μιας καταχώρησης προσωπικών στοιχείων στο *UsersDBProxy*, ώστε να αποθηκευτεί η πληροφορία.
- *loadUser(email: string)*: Η μέθοδος αυτή καλεί την αντίστοιχη συνάρτηση φόρτωσης μιας καταχώρησης προσωπικών στοιχείων στο *UsersDBProxy*, βάσει του email του χρήστη.
- *editProfile(email: string)*: Η συνάρτηση αυτή καλείται κάθε φορά που ο χρήστης επιθυμεί να αποθηκεύσει τις τροποποιήσεις στα προσωπικά του στοιχεία.
- *checkEnteredData()*: Η συνάρτηση αυτή ελέγχει την ορθότητα των καταχωρήσεων του χρήστη.



- *returnError(EnteredDataFlag: boolean)*: Η συνάρτηση αυτή επιστρέφει μήνυμα σφάλματος σε περίπτωση μη ορθής συμπλήρωσης της φόρμας τροποποίησης προσωπικών στοιχείων.
- *Cancel()*: Η μέθοδος αυτή μεταβαίνει στην αρχική σελίδα της εφαρμογής(HomePageGUI).

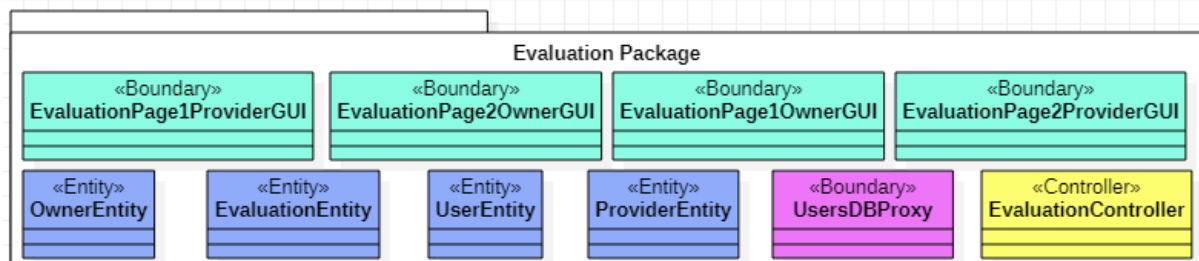
### Διάγραμμα κλάσεων:



## 4.3 <Πακέτο(α) λεξιλογίου σεναρίων χαμηλής προτεραιότητας >

### 4.3.1 Πακέτο Evaluation Package:

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που σχετίζονται με την δημιουργία και αποθήκευση των αξιολογήσεων των υπηρεσιών.





### **Boundary EvaluationPage1ProviderGUI:**

«Boundary» EvaluationPage1ProviderGUI
-lblRating: label -btnEvaluation: ArrayList <button> -lblServices: label -btnCancel: btn
+EvaluationPage1ProviderGUI(): void +clickService(): void +clickCancel(): void

Η κλάση αυτή αναπαριστά τη διεπαφή της πρώτης σελίδας προβολής των αξιολογήσεων για τον provider.

#### **Χαρακτηριστικά κλάσης:**

- *lblRating*: Επιγραφή με το μέσο όρο των αξιολογήσεων του provider.
- *btnEvaluation*: Πολλαπλά, scrollable κουμπιά με τις πληροφορίες της υπηρεσίας που έχει αξιολογηθεί.
- *lblServices*: Επιγραφή της κεφαλίδας “Services” της σελίδας.
- *btnCancel*: Κουμπί που επιτρέπει την επιστροφή στη σελίδα home page.

#### **Μέθοδοι Κλάσης:**

- *EvaluationPage1ProviderGUI()*: Μέθοδος δόμησης της κλάσης
- *clickService()*: Καλεί τη συνάρτηση `displayEvaluationPage2(evaluation:EvaluationEntity,category: string)` του ελεγκτή `EvaluationController` ώστε να γίνει η εμφάνιση των στοιχείων μίας συγκεκριμένης αξιολόγησης στη σελίδα `EvaluationPage2ProviderGUI`.
- *clickCancel()*: Καλεί τη συνάρτηση `DisplayHomePage(category: string)` του `LoginController` για την εμφάνιση του home page, σε περίπτωση που ο provider θέλει να πάει πίσω.

### **Boundary EvaluationPage2ProviderGUI:**

«Boundary» EvaluationPage2ProviderGUI
-lblOwnerName: label -lblServiceType: label -lblDate: label -lblTime: label -lblStarRating: label -sldRating: slider -lblComments: label -lblRealComment: label -btnBack: button
+EvaluationPage2ProviderGUI(): void +clickBack(): void

Η κλάση αυτή αναπαριστά τη διεπαφή της δεύτερης σελίδας προβολής των αξιολογήσεων για τον provider.

#### **Χαρακτηριστικά κλάσης:**

- *lblOwnerName*: Επιγραφή με το όνομα του owner που άφησε την αξιολόγηση.
- *lblServiceType*: Επιγραφή με το είδος της υπηρεσίας που παρείχε ο provider.
- *lblDate*: Επιγραφή της ημερομηνίας που έγινε η αγορά της υπηρεσίας.
- *lblTime*: Επιγραφή της ώρας που έγινε η αγορά της υπηρεσίας.
- *lblStarRating*: Επιγραφή της κεφαλίδας “Star Rating:”
- *sldRating*: Απενεργοποιημένο slider που δείχνει τη βαθμολογία του provider.



- *lblComments*: Επιγραφή της κεφαλίδας “Comments”.
- *lblRealComment*: Το σχόλιο που άφησε ο owner κατά την αξιολόγηση του provider.
- *btnBack*: Κουμπί που επιτρέπει την επιστροφή στη σελίδα evaluation page 1.

**Μέθοδοι Κλάσης:**

- *EvaluationPage2ProviderGUI()*: Μέθοδος δόμησης της κλάσης
- *clickBack()*: Καλεί τη συνάρτηση DisplayEvaluations() του HomePageController για την εμφάνιση του evaluation page 1 , σε περίπτωση που ο provider θέλει να πάει πίσω.

**Boundary EvaluationPage1OwnerGUI:**

«Boundary» EvaluationPage1OwnerGUI
-lblService: label -lblServiceComment: label -btnServices: ArrayList <button> -btnCancel: button
+EvaluationPage1OwnerGUI(): void +clickService(): void +clickCancel(): void

Η κλάση αυτή αναπαριστά τη διεπαφή της πρώτης σελίδας προβολής των διαθέσιμων αξιολογήσεων για τον owner.

**Χαρακτηριστικά κλάσης:**

- *lblServiceComment*: Επιγραφή με το σχόλιο κάτω από την κεφαλίδα “Services”.
- *btnEvaluation*: Πολλαπλά, scrollable κουμπιά με τις πληροφορίες της υπηρεσίας που είναι διαθέσιμη για αξιολόγηση.
- *lblService*: Επιγραφή της κεφαλίδας “Services” της σελίδας.
- *btnCancel*: Κουμπί που επιτρέπει την επιστροφή στη σελίδα home page.

**Μέθοδοι Κλάσης:**

- *EvaluationPage1OwnerGUI()*: Μέθοδος δόμησης της κλάσης
- *clickService()*: Καλεί τη συνάρτηση displayEvaluationPage2(evaluation:EvaluationEntity, category: string) του ελεγκτή EvaluationController ώστε να γίνει η εμφάνιση των στοιχείων μίας συγκεκριμένης αξιολόγησης στη σελίδα EvaluationPage2OwnerGUI.
- *clickCancel()*: Καλεί τη συνάρτηση DisplayHomePage(category: string) του LoginController για την εμφάνιση του home page, σε περίπτωση που ο owner θέλει να πάει πίσω.



### **Boundary EvaluationPage2OwnerGUI:**

«Boundary» EvaluationPage2OwnerGUI
-lblProviderName: label -lblServiceType: label -lblDate: label -lblTime: label -lblStarRating: label -sldStarRating: slider -lblComments: label -lblRealComment: label -btnSubmit: button -btnCancel: button
+EvaluationPage2OwnerGUI(): void +clickSlider(): void +clickSubmit(): void +clickCancel(): void

Η κλάση αυτή αναπαριστά τη διεπαφή της δεύτερης σελίδας προβολής των διαθέσιμων αξιολογήσεων για τον owner, όπου μπορεί να πραγματοποιήσει την αξιολόγηση.

### **Χαρακτηριστικά κλάσης:**

- *lblProviderName*: Επιγραφή με το όνομα του provider που παρείχε την υπηρεσία.
- *lblServiceType*: Επιγραφή με το είδος της υπηρεσίας που παρείχε ο provider.
- *lblDate*: Επιγραφή της ημερομηνίας που έγινε η αγορά της υπηρεσίας.
- *lblTime*: Επιγραφή της ώρας που έγινε η αγορά της υπηρεσίας.
- *lblStarRating*: Επιγραφή της κεφαλίδας “Star Rating:”
- *sldRating*: Απενεργοποιημένο slider που δείχνει τη βαθμολογία του provider.
- *lblComments*: Επιγραφή της κεφαλίδας “Comments:”.
- *lblRealComment*: Το σχόλιο που άφησε ο owner κατά την αξιολόγηση του provider.
- *btnBack*: Κουμπί που επιτρέπει την επιστροφή στη σελίδα evaluation page 1.

### **Μέθοδοι Κλάσης:**

- *EvaluationPage2ProviderGUI()*: Μέθοδος δόμησης της κλάσης
- *clickCancel()*: Καλεί τη συνάρτηση DisplayEvaluate() του HomePageController για την εμφάνιση του evaluation page 1 , σε περίπτωση που ο owner θέλει να πάει πίσω.
- *clickSlider()*: Ρυθμίζει τη βαθμολογία που θέτει ο owner στο σημείο που θα κάνει click πάνω στα αστεράκια.
- *clickSubmit()*: Καλεί τη συνάρτηση storeEvaluation(ownerOrProviderEmail:string, evaluation: EvaluationEntity), μετά την updateOwnerEvaluation() και στην συνέχεια την updateProviderRating(emailProvider: string) του EvaluationController.

**Entity EvaluationPage1ProviderGUI:**

«Entity» EvaluationEntity
-emailOwner: string -emailProvider: string -serviceType: string -datePurchased: date -timePurchased: date -rating: float -comment: string -evaluated: boolean
+EvaluationEntity(): void +getEmailOwner(): string +setEmailOwner(ownerEmail: string): void +getEmailProvider(): string +setEmailProvider(providerEmail: string): void +getServiceType(): string +setServiceType(service: string): void +getDatePurchased(): date +setDatePurchased(date: date): void +getTimePurchased(): date +setTimePurchased(time: date): void +getRating(): float +setRating(rating: float): void +getComment(): string +setComment(comment: string): void +getEvaluated(): boolean +setEvaluated(evaluated: boolean): void

Η κλάση αυτή αποτελεί μια ομαδοποίηση των στοιχείων μιας αξιολόγησης. Αρχικοποιείται και αποδίδεται ως διαθέσιμη αξιολόγηση στην κλάση OwnerEntity ενός owner όταν κάνει μια αγορά υπηρεσίας. Αποδίδεται στην κλάση ProviderEntity του αντίστοιχου provider, όταν συμπληρωθεί και υποβληθεί από τον owner.

**Χαρακτηριστικά κλάσης:**

- *emailOwner*: Το email του owner που αγόρασε την υπηρεσία.
- *emailProvider*: Το email του provider που παρείχε την υπηρεσία.
- *serviceType*: Ο τύπος υπηρεσίας που παρείχε ο provider.
- *datePurchased*: Η ημέρα που έγινε η αγορά.
- *timePurchased*: Η ώρα που έγινε η αγορά.
- *rating*: Η βαθμολογία που έθεσε ο owner.
- *comment*: Το σχόλιο που έκανε ο owner.
- *evaluated*: Μεταβλητή που αποθηκεύει αν μια αξιολόγηση έχει υποβληθεί ή όχι.

**Μέθοδοι Κλάσης:**

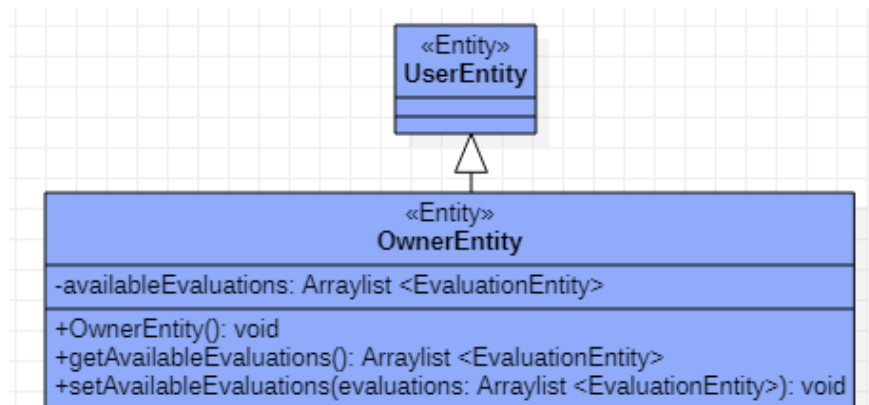
- *EvaluationEntity()*: Μέθοδος δόμησης της κλάσης.
- *getEmailOwner()*: Επιστρέφει τη μεταβλητή emailOwner.
- *setEmailOwner(ownerEmail: string)*: Αλλάζει τη μεταβλητή emailOwner στο όρισμα που της δόθηκε.
- *getEmailProvider()*: Επιστρέφει τη μεταβλητή emailProvider.
- *setEmailProvider(emailProvider: string)*: Αλλάζει τη μεταβλητή emailProvider στο όρισμα που της δόθηκε.
- *getServiceType()*: Επιστρέφει τη μεταβλητή serviceType.
- *setServiceType(service: string)*: Αλλάζει τη μεταβλητή serviceType στο όρισμα που της δόθηκε.
- *getDatePurchased()*: Επιστρέφει τη μεταβλητή datePurchased.
- *setDatePurchased(date: date)*: Αλλάζει τη μεταβλητή datePurchased στο όρισμα που της δόθηκε.





- *getTimePurchased()*: Επιστρέφει τη μεταβλητή *timePurchased*.
- *setTimePurchased(time: date)*: Αλλάζει τη μεταβλητή *timePurchased* στο όρισμα που της δόθηκε.
- *getRating()*: Επιστρέφει τη μεταβλητή *rating*.
- *setRating(rating: float)*: Αλλάζει τη μεταβλητή *rating* στο όρισμα που της δόθηκε.
- *getComment()*: Επιστρέφει τη μεταβλητή *comment*.
- *setComment(comment: string)*: Αλλάζει τη μεταβλητή *comment* στο όρισμα που της δόθηκε.
- *getEvaluated()*: Επιστρέφει τη μεταβλητή *evaluated*.
- *setEvaluated(evaluated: boolean)*: Αλλάζει τη μεταβλητή *evaluated* στο όρισμα που της δόθηκε.

### **Entity OwnerEntity:**



Η κλάση αυτή είναι υποκλάση του *UserEntity* και προσθέτει τα απαραίτητα χαρακτηριστικά και μεθόδους σχετικά με τις αξιολογήσεις.

### **Χαρακτηριστικά κλάσης:**

- *availableEvaluations*: *ArrayList* που περιέχει όλες τις διαθέσιμες αξιολογήσεις που μπορεί να κάνει ο *owner*.

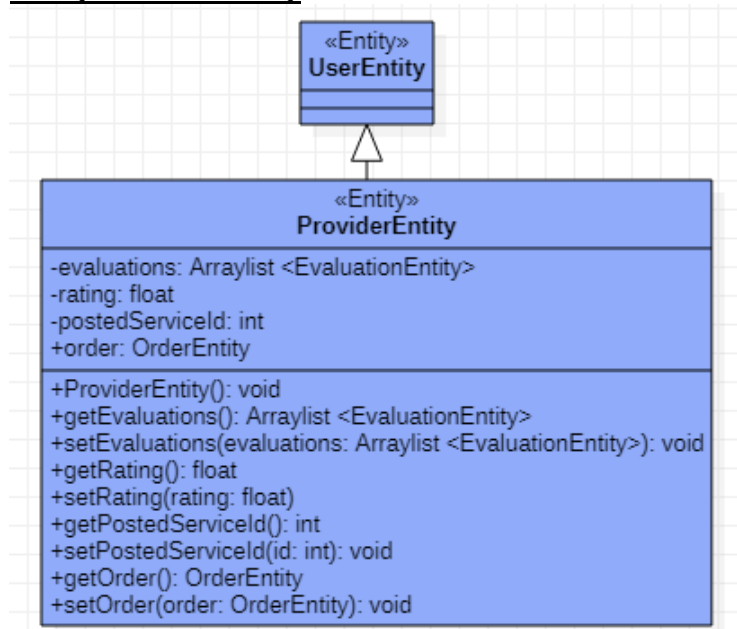
### **Μέθοδοι Κλάσης**

- *OwnerEntity()*: Μέθοδος δόμησης της κλάσης.
- *getAvailableEvaluations()*: Επιστρέφει το *arraylist* με όλες τις διαθέσιμες αξιολογήσεις του *owner*.
- *setAvailableEvaluations( evaluations: ArrayList <EvaluationEntity> )*: Αλλάζει τη μεταβλητή *evaluations* στο όρισμα που της δόθηκε.





### Entity ProviderEntity:



Η κλάση αυτή είναι υποκλάση του UserEntity και προσθέτει τα απαραίτητα χαρακτηριστικά και μεθόδους σχετικά με τις αξιολογήσεις, τη δημοσιευμένη υπηρεσία του provider και την παραγγελία για μια υπηρεσία από κάποιον Owner.

### Χαρακτηριστικά κλάσης:

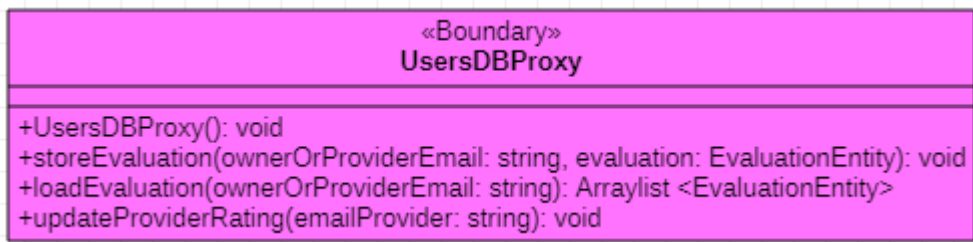
- *evaluations*: ArrayList τύπου EvaluationEntity που περιέχει όλες τις αξιολογήσεις του provider.
- *rating*: Ο μέσος όρος του βαθμού των αξιολογήσεων του provider.
- *postedServiceId*: Το ID της υπηρεσίας που έχει δημοσιεύσει ένας provider.
- *order*: Η υπηρεσία του provider που έχει αγοραστεί από έναν owner και πρέπει να εκπληρωθεί.

### Μέθοδοι Κλάσης:

- *ProviderEntity()*: Μέθοδος δόμησης της κλάσης.
- *getEvaluations()*: Επιστρέφει το ArrayList evaluations με όλα τα αντικείμενα αξιολογήσεων που έχει ένας provider.
- *setEvaluations(evaluations: ArrayList <EvaluationEntity>)*: Αλλάζει το ArrayList με όλα τα αντικείμενα αξιολογήσεων evaluations που έχει ένας provider.
- *getRating()*: Επιστρέφει τη μεταβλητή rating του provider.
- *setRating(rating: float)*: Αλλάζει τη μεταβλητή rating του provider στο όρισμα που δόθηκε.
- *getPostedServiceId()*: Επιστρέφει τη μεταβλητή postedServiceId.
- *setPostedServiceId()*: Αλλάζει τη μεταβλητή postedServiceId στο όρισμα που δόθηκε.
- *getOrder()*: Επιστρέφει τη μεταβλητή order.
- *setOrder(order: OrderEntity)*: Αλλάζει τη μεταβλητή order στο όρισμα που της δόθηκε.



### **Boundary UsersDBProxy:**

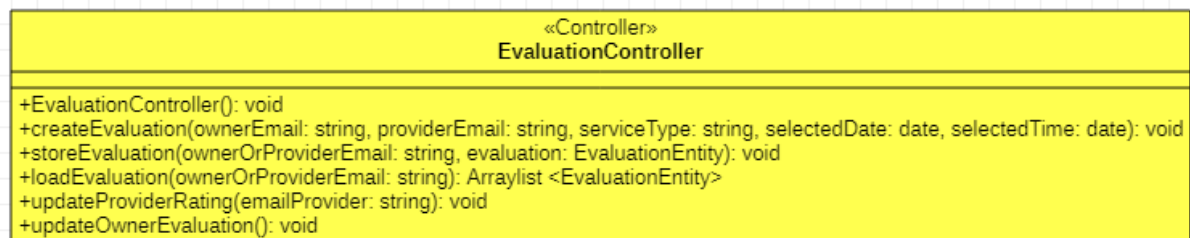


Η κλάση αυτή είναι η κλάση επικοινωνίας με τη βάση δεδομένων UsersDB, ώστε να είναι δυνατή η ανάκτηση, αποθήκευση και ενημέρωση των δεδομένων της.

#### **Μέθοδοι Κλάσης:**

- *UsersDBProxy()*: Μέθοδος δόμησης της κλάσης.
- *storeEvaluation(ownerOrProviderEmail:string,evaluation:EvaluationEntity)*: Αποθηκεύει στο OwnerEntity ή ProviderEntity με το δοσμένο email το EvaluationEntity στη λίστα availableEvaluations ή evaluations ανάλογα με το category του.
- *loadEvaluation(ownerOrProviderEmail:string)*: Επιστρέφει τη λίστα με όλα τα EvaluationEntity του OwnerEntity ή ProviderEntity στο οποίο αντιστοιχεί το δοσμένο email.
- *updateProviderRating(emailProvider: string)*: Υπολογίζει το μέσο όρο των ratings στα evaluations του provider με το δοσμένο email και ανανεώνει την μεταβλητή rating του.

### **Controller EvaluationController:**



Η κλάση αυτή είναι ο ελεγκτής με τις απαραίτητες μεθόδους για τον έλεγχο της ροής των αξιολογήσεων.

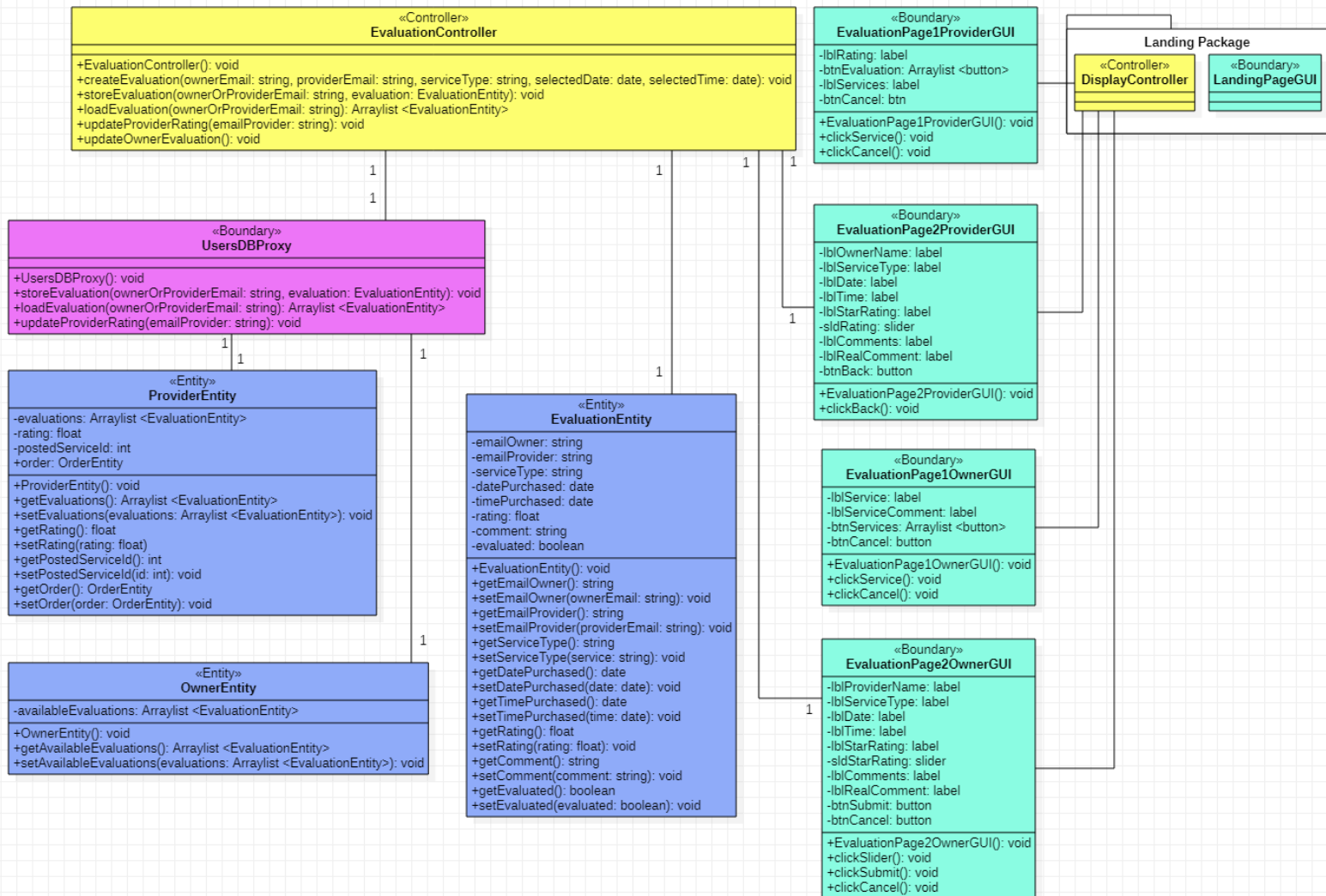
#### **Μέθοδοι Κλάσης:**

- *EvaluationController()*: Μέθοδος δόμησης της κλάσης.
- *createEvaluation(ownerEmail:string, providerEmail:string, serviceType:string, selectedDate:date, selectedTime:date)*: Δημιουργεί ένα EvaluationEntity, στο οποίο αρχικοποιεί όλες τις μεταβλητές εκτός από το rating και το comment. Στη συνέχεια, καλεί τη συνάρτηση storeEvaluation(ownerOrProviderEmail:string, evaluation:EvaluationEntity) του UsersDBProxy με το email του owner που έκανε την αγορά και του αποδίδει το EvaluationEntity με τις απαραίτητες πληροφορίες της αγοράς.
- *storeEvaluation(ownerOrProviderEmail:string,evaluation:EvaluationEntity)*: Καλεί τη συνάρτηση storeEvaluation(ownerOrProviderEmail:string, evaluation:EvaluationEntity) του UsersDBProxy.
- *loadEvaluation(in ownerOrProviderEmail:string)*: Καλεί τη συνάρτηση loadEvaluation(in ownerOrProviderEmail:string) από το UsersDBProxy.
- *updateProviderRating(emailProviderRating: string)*: Καλεί τη συνάρτηση updateProviderRating(emailProviderRating: string) από το UsersDBProxy.



- *updateOwnerEvaluation()*: Αλλάζει τα χαρακτηριστικά rating και comment του τρέχον αντικειμένου EvaluationEntity με βάση τα πεδία του αντικειμένου EvaluationPage2OwnerGUI.

### Διάγραμμα κλάσεων:



## Παράρτημα Ι – Γλωσσάριο

Το σετ των ακρωνυμίων που χρησιμοποιείται στο έγγραφο

ΛΑ-xxxx Λειτουργική Απαίτηση xxxx

ΜΛΑ-xx Μη Λειτουργική Απαίτηση