



Make your place better

Απαιτήσεις Χρηστών και Προδιαγραφές Συστήματος

Del.1.1

Version 1.0

Καρατζάς Μιχάλης mikalakis@ece.auth.gr
Μαραντίδης Κυριάκος kyrmarkos@ece.auth.gr
Εμμανουήλ Μιχάλινας michalain@ece.auth.gr
Ιωάννης Χαρισιάδης charisiad@ece.auth.gr

8/04/2010



Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Καρατζάς Μιχάλης	myDimos	mikalakis@ece.auth.gr
Εμμανουήλ Μιχάλαϊνας	myDimos	michalain@ece.auth.gr
Μαραντίδης Κυριάκος	myDimos	kyrmarkos@ece.auth.gr
Ιωάννης Χαρισιάδης	myDimos	charisiad@ece.auth.gr



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	3
Λίστα Σχημάτων	5
1 Απαιτήσεις Συστήματος.....	6
1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης)	6
1.2 Χρήστες και εξωτερικά συστήματα.....	8
1.2.1 Κατηγορίες χρηστών.....	8
1.2.2 Εξωτερικά Συστήματα.....	9
1.3 Σημαντικές μη λειτουργικές απαιτήσεις.....	9
1.3.1 Απαιτήσεις ασφάλειας (Security).....	9
1.3.2 Απαιτήσεις Φορητότητας (Portability)	10
1.3.3 Απαιτήσεις Αξιοπιστίας (Reliability)	10
1.3.4 Τεχνικές Απαιτήσεις περιβάλλοντος	10
1.3.5 Απαιτήσεις πολιτικής (Policy - legal, corporate)	11
1.4 Λεξικογραφικοί Προσδιορισμοί.....	11
2 Σενάρια Χρήσης.....	13
2.1 Διάγραμμα σεναρίων χρήσης	13
2.2 Feature: Issue submission.....	13
2.3 Feature: View issues list (user).....	14
2.4 Feature: Issue upvote	15
2.5 Feature: Issue content view (user)	16
2.6 Feature: Issues filtering(user).....	17
2.7 Feature: View my pending issues	18
2.8 Feature: Issue management (user).....	19
2.9 Feature: View issue list (admin)	21
2.10 Feature: Issue filtering (admin).....	22
2.11 Feature: Category management.....	23
2.12 Feature: Issue management (admin)	25
3 Επιδεικτικά γραφικά παράθυρα διεπαφής	27
3.1 Αρχική σελίδα χρήστη.....	27
3.2 Σελίδα Εφαρμογής φίλτρων χρήστη.....	29
3.3 Σελίδα Καταχώρησης ζητήματος.....	30
3.4 Σελίδα Ζητήματος.....	31
3.5 Σελίδα Προβολής εκκρεμών ζητημάτων χρήστη.....	33
3.6 Σελίδα Επεξεργασίας Εκκρεμούς ζητήματος - χρήστη.....	34
3.7 Αρχική οθόνη διαχειριστή - λίστα ζητημάτων διαχειριστή	35
3.8 Σελίδα διαχείρισης ζητήματος – διαχειριστής.....	36
3.9 Σελίδα κατηγοριών – διαχειριστής.....	37
3.10 Σελίδα διαχείρισης κατηγορίας – διαχειριστής.....	38
4 Στατική Μοντελοποίηση	39



4.1	- Πακέτο(α) λεξιλογίου σεναρίων υψηλής προτεραιότητας	39
4.1.1	Πακέτο UserHomePage.....	39
4.1.1.1	Ορισμοί- Πακέτο UserHomePage	39
4.1.1.2	Διαγράμματα κλάσεων - Πακέτο UserHomePage	45
4.1.2	Πακέτο UserIssuesHandle	45
4.1.2.1	Ορισμοί - Πακέτο UserIssuesHandle.....	46
4.1.2.2	Διαγράμματα κλάσεων - Πακέτο UserIssuesHandle	55
4.1.3	Πακέτο AdminCore	55
4.1.3.1	Ορισμοί – Πακέτο AdminCore.....	56
4.1.3.2	Διαγράμματα κλάσεων – Πακέτο AdminCore.....	63
4.2	Πακέτο λεξιλογίου σεναρίων μέσης προτεραιότητας.....	64
4.2.1	Πακέτο CategoryManagement.....	64
4.2.1.1	Ορισμοί – Πακέτο CategoryManagement.....	64
4.2.1.2	Διαγράμματα Κλάσεων – Πακέτο CategoryManagement	68
4.3	Πακέτα λεξιλογίου σεναρίων χαμηλής προτεραιότητας – Συμπληρωματικά πακέτα.....	69
4.3.1	Πακέτο GpsSystemHandle	69
4.3.1.1	Ορισμοί - Πακέτο GpsSystemHandle	69
4.3.1.2	Διαγράμματα κλάσεων - Πακέτο GpsSystemHandle.....	70
4.3.2	Πακέτο UserDeviceHandle	70
4.3.2.1	Ορισμοί - Πακέτο GpsSystemHandle	71
4.3.2.2	Διαγράμματα κλάσεων - Πακέτο UserDeviceHandle	72
	Παράρτημα Ι – Γλωσσάριο	73
	Παράρτημα ΙΙ – Ανοιχτά Θέματα.....	74



Λίστα Σχημάτων

Σχήμα 1. Λεζάντα Σχήματος

Σχήμα 2.1: Διάγραμμα σεναρίων χρήσης	13
Σχήμα 2.2: Ο χρήστης καταχωρεί ζήτημα	15
Σχήμα 2.3: Ο χρήστης βλέπει, υπερψηφίζει και προβάλλει τα ζητήματα.	17
Σχήμα 2.4: Ο χρήστης φιλτράρει τα δημοσιευμένα ζητήματα.	19
Σχήμα 2.5: Ο χρήστης βλέπει και επεξεργάζεται τα εκκρεμή του ζητήματα.	21
Σχήμα 2.6: Ο διαχειριστής βλέπει και φιλτράρει τα ζητήματα του.....	23
Σχήμα 2.7: Ο διαχειριστής διαχειρίζεται μία κατηγορία.	24
Σχήμα 2.8: Ο διαχειριστής διαχειρίζεται ένα ζήτημα	26
Σχήμα 3.1: (Αριστερά) Η αρχική σελίδα του Χρήστη - Χωρίς Ψηφισμένο ζήτημα	27
Σχήμα 3.2: (Δεξιά) Η αρχική σελίδα του Χρήστη - Ψηφισμένο το πρώτο ζήτημα.....	27
Σχήμα 3.3: Η αρχική σελίδα του Χρήστη - Λυμένο το πρώτο ζήτημα.....	28
Σχήμα 3.4: Σελίδα εφαρμογής φίλτρων χρήστη.....	29
Σχήμα 3.5: Σελίδα Καταχώρησης ζητήματος.....	30
Σχήμα 3.6: (Αριστερά) Σελίδα μη επιλυμένου ζητήματος	31
Σχήμα 3.7: (Δεξιά) Σελίδα επιλυμένου ζητήματος	31
Σχήμα 3.8: Σελίδα υπερψηφισμένου ζητήματος.....	32
Σχήμα 3.9: Σελίδα προβολής εκκρεμών ζητημάτων χρήστη	33
Σχήμα 3.10: Σελίδα προβολής εκκρεμών ζητημάτων χρήστη - κενή	33
Σχήμα 3.11: Σελίδα επεξεργασίας εκκρεμούς ζητήματος - χρήστης	34
Σχήμα 3.12: Αρχική οθόνη διαχειριστή - λίστα ζητημάτων διαχειριστή	35
Σχήμα 3.13: Σελίδα διαχείρισης ζητήματος-διαχειριστής	36
Σχήμα 3.14: Σελίδα κατηγοριών-διαχειριστής	37
Σχήμα 3.15: Σελίδα επεξεργασίας κατηγορίας -διαχειριστής.....	38

1 Απαιτήσεις Συστήματος

1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης)

<ΛΑ- 1> Ο χρήστης πρέπει να μπορεί να **καταχωρεί ένα ζήτημα** στο σύστημα.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να καταχωρεί ένα ζήτημα αρμοδιότητας του δήμου, στο σύστημα, συμπληρώνοντας τα στοιχεία μιας φόρμας καταχώρησης.

User Priority(5/5): Η καταχώρηση ζητήματος αποτελεί πρωταρχικής σημασίας λειτουργικότητα για το χρήστη και κύριο λόγο ύπαρξης του συστήματος μας.

Technical Priority(5/5): Η απαίτηση αυτή είναι ιδιαίτερα σημαντική και για το σύστημα καθώς η ανυπαρξία καταχωρημένων ζητημάτων αφαιρεί λειτουργικότητα και το καθιστά κενό και άχρηστο.

<ΛΑ- 2> Ο χρήστης πρέπει να μπορεί να βλέπει τα **δημοσιευμένα ζητήματα**.

Περιγραφή: Ο χρήστης θα έπρεπε να μπορεί να βλέπει μια λίστα με όλα τα δημοσιευμένα ζητήματα, όπου επιπλέον εμφανίζονται οι ψήφοι κάθε ζητήματος καθώς και πληροφορία σχετική με την κατάσταση του ζητήματος, την τοποθεσία του και την απόσταση από τον χρήστη.

User Priority(3/5): Η προβολή των δημοσιευμένων ζητημάτων αποτελεί επιθυμητή λειτουργικότητα για τον χρήστη, καθώς έτσι παρακολουθεί τα ζητήματα που τον ενδιαφέρουν.

Technical Priority(2/5): Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, επειδή η λειτουργία και η κατάσταση του είναι ανεξάρτητες από αυτήν.

<ΛΑ-3> Ο χρήστης θα έπρεπε να μπορεί να **υπερψηφίσει** ένα δημοσιευμένο **ζήτημα**.

Περιγραφή: Ο χρήστης θα έπρεπε να μπορεί να υπερψηφίσει ένα δημοσιευμένο ζήτημα προκειμένου να αυξήσει την σημαντικότητά του. Δηλαδή ζητήματα με περισσότερες ψήφους είναι πιο σημαντικά για τους δημότες.

User Priority(3/5): Η δυνατότητα υπερψηφίσης ενός ζητήματος αποτελεί επιθυμητή λειτουργικότητα για τον χρήστη, καθώς έτσι μπορεί να αναδείξει την σημαντικότητά του.

Technical Priority(2/5): Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, καθώς εξυπηρετεί μόνο την ταξινόμηση των ζητημάτων.

<ΛΑ-4> Ο χρήστης θα έπρεπε να μπορεί να **προβάλλει** το περιεχόμενο ενός δημοσιευμένου ζητήματος.

Περιγραφή: Ο χρήστης θα έπρεπε να μπορεί να προβάλλει το πλήρες περιεχόμενο ενός δημοσιευμένου ζητήματος, επιλέγοντας το από την λίστα με τα δημοσιευμένα ζητήματα.

User Priority(3/5): Η προβολή του πλήρους περιεχομένου ενός ζητήματος που επιλέγει ο χρήστης αποτελεί επιθυμητή λειτουργικότητα για τον χρήστη, καθώς έτσι μπορεί να δει λεπτομέρειες για ένα ζήτημα που τον ενδιαφέρει.

Technical Priority(2/5): Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, επειδή η λειτουργία και η κατάσταση του είναι ανεξάρτητες από αυτήν.

<ΛΑ-5> Ο χρήστης θα έπρεπε να μπορεί να **φιλτράρει** τα **δημοσιευμένα ζητήματα**.



Περιγραφή: Ο χρήστης θα έπρεπε να μπορεί να εφαρμόζει μία σειρά από διαθέσιμα φίλτρα προκειμένου να περιορίσει την λίστα των δημοσιευμένων ζητημάτων που εμφανίζονται στην οθόνη του/της.

User Priority(3/5): Η δυνατότητα επιλεκτικής προβολής της λίστας των δημοσιευμένων ζητημάτων, αποτελεί επιθυμητή λειτουργικότητα για τον χρήστη.

Technical Priority(1/5): Η απαίτηση αυτή δεν είναι σημαντική για το σύστημα, επειδή η λειτουργία και η κατάσταση του είναι ανεξάρτητες από αυτήν.

<ΛΑ- 6> Ο χρήστης θα έπρεπε να μπορεί να **βλέπει τα εκκρεμή ζητήματα που έχει καταχωρήσει.**

Περιγραφή: Ο χρήστης θα έπρεπε να μπορεί να βλέπει μια λίστα με τα εκκρεμή ζητήματα που έχει καταχωρήσει .

User Priority(3/5): Η προβολή των ζητημάτων που έχει καταχωρήσει ένας χρήστης και αναμένουν έγκριση από τους διαχειριστές, αποτελεί επιθυμητή λειτουργικότητα για τον χρήστη.

Technical Priority(2/5): Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, επειδή η λειτουργία και η κατάσταση του είναι ανεξάρτητες από αυτήν.

<ΛΑ-7> Ο χρήστης πρέπει να μπορεί να **διαχειριστεί ένα εκκρεμές ζήτημα που έχει καταχωρήσει.**

Περιγραφή: Ο χρήστης πρέπει να μπορεί να επεξεργαστεί ή να διαγράψει ένα ζήτημα το οποίο έχει καταχωρήσει ο ίδιος στο σύστημα.

User Priority(4/5): Η διαχείριση ενός ζητήματος από τον χρήστη που το έχει καταχωρήσει, αποτελεί αρκετά σημαντική λειτουργικότητα για αυτόν, καθώς έτσι μπορεί να διορθώσει τυχόντα σφάλματα του.

Technical Priority(2/5): Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, καθώς η κατάσταση και η λειτουργία του δεν επηρεάζονται σημαντικά από την επεξεργασία ή διαγραφή ορισμένων καταχωρήσεων.

<ΛΑ-8> Ο διαχειριστής θα έπρεπε να μπορεί να **βλέπει τα ζητήματα.**

Περιγραφή: Ο διαχειριστής θα πρέπει να μπορεί να βλέπει μια λίστα με όλα τα ζητήματα , όπου επιπλέον εμφανίζεται το πλήρες περιεχόμενο κάθε ζητήματος .

User Priority(5/5): Η προβολή των ζητημάτων του δήμου, που έχουν καταχωρηθεί στο σύστημα, από τους διαχειριστές, αποτελεί πρωταρχικής σημασίας λειτουργικότητα για αυτούς.

Technical Priority(2/5): Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, επειδή η λειτουργία και η κατάσταση του είναι ανεξάρτητες από αυτήν.

<ΛΑ-9> Ο διαχειριστής θα έπρεπε να μπορεί να **φιλτράρει τα ζητήματα.**

Περιγραφή: Ο διαχειριστής θα πρέπει να μπορεί να εφαρμόζει μία σειρά από διαθέσιμα φίλτρα προκειμένου να περιορίσει την λίστα των δημοσιευμένων ζητημάτων που θα εμφανίζονται στην οθόνη του/της.

User Priority(4/5): Η δυνατότητα επιλεκτικής προβολής της λίστας των ζητημάτων, αποτελεί αρκετά σημαντική λειτουργικότητα για έναν διαχειριστή.

Technical Priority(1/5): Η απαίτηση αυτή δεν είναι σημαντική για το σύστημα, επειδή η λειτουργία και η κατάσταση του είναι ανεξάρτητες από αυτήν.

<ΛΑ-10> Ο διαχειριστής θα έπρεπε να μπορεί να **διαχειριστεί τις κατηγορίες**.

Περιγραφή: Ο διαχειριστής πρέπει να μπορεί να δημιουργήσει μία νέα κατηγορία, να επεξεργαστεί ή να διαγράψει μία υπάρχουσα.

User Priority(4/5): Η διαχείριση μιας κατηγορίας, αποτελεί αρκετά σημαντική λειτουργικότητα για τον διαχειριστή, καθώς έτσι αυτός προσαρμόζει το σύστημα στις ανάγκες του δήμου.

Technical Priority(3/5): Η απαίτηση αυτή είναι μέτριας σημαντικότητας για το σύστημα, καθώς η λειτουργία του και κατάσταση του επηρεάζονται ελαφρώς από αυτή.

<ΛΑ-11> Ο διαχειριστής πρέπει να μπορεί να **διαχειριστεί ένα ζήτημα**.

Περιγραφή: Ο διαχειριστής θα πρέπει να μπορεί να επεξεργαστεί ένα ζήτημα, να δει τα στοιχεία επικοινωνίας του χρήστη που το καταχώρησε, να αλλάξει την κατάσταση του ζητήματος ή να το διαγράψει από το σύστημα.

User Priority(5/5): Η διαχείριση ενός ζητήματος είναι πρωταρχικής σημασίας λειτουργικότητα για τον διαχειριστή.

Technical Priority(5/5): Η δυνατότητα διαχείρισης ζητημάτων από έναν διαχειριστή είναι εξίσου σημαντική και για το σύστημα, καθώς η απουσία της το καθιστά άκαμπτο και ευάλωτο σε ακατάλληλο περιεχόμενο.

1.2 Χρήστες και εξωτερικά συστήματα

Το σύστημα myDimos χρησιμοποιείται από κάτοικους ενός δήμου που επιθυμούν να καταχωρούν ζητήματα αρμοδιότητας των υπηρεσιών του δήμου προς επίλυση. Το σύστημα προϋποθέτει ότι έχουν οριστεί αρμόδιοι χρήστες (διαχειριστές) από τον εκάστοτε δήμο, ώστε να επεξεργάζονται τα καταχωρημένα ζητήματα και να διαμορφώνουν το σύστημα, με βάση τις ανάγκες του δήμου. Επιπλέον, προϋποτίθεται ότι οι δημότες έχουν εγγραφεί ως χρήστες και ταυτοποιηθεί από το σύστημα προτού το χρησιμοποιήσουν.

1.2.1 Κατηγορίες χρηστών

➤ <Χρήστης 1> Απλός χρήστης(user)

Σκοπός αυτού του χρήστη είναι να καταχωρεί ζητήματα στο σύστημα που αφορούν προβλήματα του δήμου του, που επηρεάζουν την καθημερινή ζωή του και χρειάζονται επίλυση. Ο χρήστης αυτός έχει το δικαίωμα να καταχωρεί ζητήματα, καθώς επίσης και να υπερψηφίζει ζητήματα, ώστε να δείξει με ένα μέτρο την σημασία τους και την αναγκαιότητα να επιλυθούν. Ο χρήστης αυτός, πρόσθετα, μπορεί να βλέπει τα δημοσιευμένα και επιλυμένα ζητήματα, να τα οπτικοποιεί με χρήση χάρτη και έτσι να παρακολουθεί την πορεία επίλυσης τους. Επίσης, μπορεί να επεξεργάζεται τα εκκρεμή ζητήματα που έχει καταχωρήσει. Σημειώνεται ότι η χρήση της εφαρμογής είναι επιτρεπτή μόνο από ταυτοποιημένους χρήστες και όχι επισκέπτες.

➤ <Χρήστης 2> Διαχειριστής(admin)

Ο αριθμός αυτών των χρηστών είναι περιορισμένος καθώς αυτοί πρέπει να είναι εξουσιοδοτημένοι εργαζόμενοι του δήμου. Ο σκοπός τους είναι η διαχείριση των καταχωρημένων ζητημάτων. Ο διαχειριστής αποφασίζει να δημοσιεύσει ένα ζήτημα, όταν κρίνει ότι πληροί τα **κριτήρια καταλληλότητας**. Ο διαχειριστής χρησιμοποιεί την λειτουργικότητα του συστήματος για να οργανώσει ένα σχέδιο επίλυσης των δημοσιευμένων ζητημάτων και να προβεί στις απαραίτητες ενέργειες (π.χ. επικοινωνία με τον χρήστη, προώθηση του αιτήματος στις αρμόδιες υπηρεσίες του δήμου). Οι διαχειριστές έχουν το δικαίωμα προβολής, οπτικοποίησης με χρήση χαρτών και διαχείρισης των ζητημάτων. Επιπλέον, διαχειρίζονται τις κατηγορίες των ζητημάτων, ώστε να διαμορφώνουν κατάλληλα το σύστημα για τις ανάγκες του δήμου τους.

1.2.2 Εξωτερικά Συστήματα

➤ <Εξωτερικό Σύστημα 1> Βάση δεδομένων(Database)

Η βάση δεδομένων αποθηκεύει τα παρακάτω στοιχεία:

- ✓ Στοιχεία των ταυτοποιημένων χρηστών:
 - User ID
 - Λίστα με ID των ζητημάτων του χρήστη
- ✓ Δεδομένα των καταχωρημένων ζητημάτων
 - ID ζητήματος
 - Τίτλος
 - Συνοπτική περιγραφή
 - Τοποθεσία
 - Οπτικοακουστικό υλικό
 - Κατηγορία
 - Κατάσταση ζητήματος
 - ID του χρήστη που το καταχώρησε
 - Αριθμός ψήφων
 - Ημερομηνία καταχώρησης
- ✓ Δεδομένα των δημιουργημένων κατηγοριών
 - ID κατηγορίας
 - Όνομα κατηγορίας
- ✓ Δεδομένα του χρήστη Διαχειριστή
 - ID Διαχειριστή

➤ <Εξωτερικό Σύστημα 2> Εφαρμογή (Σύστημα) διαχείρισης ιχών GPS

Για την υποστήριξη των λειτουργικών απαιτήσεων που σχετίζονται με τοποθεσίες και διευθύνσεις, το σύστημα πρέπει να μπορεί να συνδέεται με το εξωτερικό σύστημα Google maps το οποίο επιτρέπει:

- Αντιστοίχιση στιγμάτων GPS με ταχυδρομική διεύθυνση.
- Υπολογισμός της απόστασης της τοποθεσίας των ζητημάτων από το στίγμα του χρήστη.

1.3 Σημαντικές μη λειτουργικές απαιτήσεις

1.3.1 Απαιτήσεις ασφάλειας (Security)

<ΜΛΑ-1> Το σύστημα πρέπει να πληροί προδιαγραφές ασφάλειας.

Περιγραφή: Το σύστημα πρέπει να μπορεί να αντιμετωπίσει κακόβουλες επιθέσεις, ώστε να επιτυγχάνεται η ασφάλεια των προσωπικών δεδομένων των χρηστών, διασφαλίζεται η φήμη των φορέων που το χρησιμοποιούν καθώς και η ομαλή του λειτουργία.

User Priority: (5/5) Η απαίτηση αυτή είναι ιδιαίτερα σημαντική για τον χρήστη, καθώς θα ήταν διστακτικός απέναντι στη χρήση του, αν δεν υπήρχε αξιόπιστη ασφάλεια.

Technical Priority: (5/5) Η απαίτηση αυτή είναι εξίσου σημαντική για το σύστημα, καθώς εάν δεν μπορεί, να αντιμετωπίσει με επιτυχία διάφορες απόπειρες επίθεσης, κινδυνεύει να καταρρεύσει.



Stability: (3/5) Οι βασικές προδιαγραφές ασφαλείας θα παραμείνουν υλοποιημένες, ωστόσο απαιτείται ενημέρωση του συστήματος ώστε να παραμένει ασφαλές από νεότερες απειλές.

1.3.2 Απαιτήσεις Φορητότητας (Portability)

<ΜΛΑ-2> Το σύστημα πρέπει να είναι συμβατό με τις πλέον δημοφιλείς συσκευές.

Περιγραφή: Το σύστημα πρέπει να μπορεί να χρησιμοποιηθεί από web browsers και συσκευές που λειτουργούν με Android ή iOS . Με αυτόν τον τρόπο, όλο και περισσότεροι χρήστες θα μπορούν να το χρησιμοποιούν.

User Priority: (4/5) Η απαίτηση αυτή είναι αρκετά σημαντική για τους χρήστες, καθώς έτσι διευκολύνεται η πρόσβαση τους στο σύστημα.

Technical Priority: (2/5) Η απαίτηση αυτή δεν είναι σημαντική για το σύστημα, επειδή η λειτουργία και η κατάσταση του δεν επηρεάζονται σημαντικά ,από το αν αυτό είναι συμβατό με ορισμένους τύπους συσκευών ή με ορισμένα λειτουργικά συστήματα.

Stability: (4/5) Η δυνατότητα χρήσης του συστήματος από διάφορες συσκευές και περιβάλλοντα είναι κάτι που δεν προβλέπεται να αλλάξει. Ωστόσο η εμφάνιση νέων ηλεκτρονικών συσκευών μπορεί να οδηγήσει στην ανάπτυξη εκδόσεων συμβατών με αυτές τις συσκευές.

1.3.3 Απαιτήσεις Αξιοπιστίας (Reliability)

<ΜΛΑ-3> Το σύστημα θα έπρεπε να μπορεί να διαχειρίζεται στίγματα GPS, μέσω του συστήματος google maps.

Περιγραφή: Το σύστημα θα έπρεπε να μπορεί να υπολογίζει και να εμφανίζει , κατά προσέγγιση την απόσταση ενός χρήστη από τα δημοσιευμένα ζητήματα , χρησιμοποιώντας το GPS στίγμα του. Επίσης θα πρέπει να αντιστοιχεί το στίγμα ενός χρήστη σε διεύθυνση , όταν το χρησιμοποιεί ο χρήστης στην καταχώρηση ενός ζητήματος.

User Priority: (3/5) Η απαίτηση αυτή είναι μέτριας σημαντικότητας για τον χρήστη καθώς απλά του επιτρέπει να δει την απόστασή του από τα δημοσιευμένα ζητήματα και να δηλώνει την τοποθεσία του μέσω του στίγματός του όταν καταχωρεί ένα ζήτημα. Αντίθετα για έναν διαχειριστή είναι αρκετά σημαντική, καθώς του δίνει την δυνατότητα να προσδιορίσει τον τόπο του προβλήματος.

Technical Priority: (3/5) Η απαίτηση αυτή είναι μέτριας σημαντικότητας για το σύστημα, καθώς η λειτουργία του και η κατάστασή του επηρεάζονται σε μικρό βαθμό από αυτή.

Stability: (5/5) Η δυνατότητα διαχείρισης στίγμάτων GPS , είναι κάτι που δεν πρόκειται να αλλάξει από το σύστημά μας.

1.3.4 Τεχνικές Απαιτήσεις περιβάλλοντος

<ΜΛΑ- 4> Το σύστημα θα έπρεπε να υποστηρίζει πολυμέσα υψηλής ποιότητας.

Περιγραφή: Το σύστημα θα έπρεπε να μπορεί να προβάλλει βίντεο 1080p και εικόνες υψηλής ανάλυσης.

User Priority: (3/5) Η απαίτηση αυτή είναι μέτριας σημαντικότητας για τον χρήστη, καθώς διευκολύνεται στην ανάδειξη και οπτικοποίηση των προβλημάτων εάν μπορούν να ανεβάσουν και να προβάλλουν πολυμέσα υψηλής ευκρίνειας. Εξίσου σημαντική είναι και για τους διαχειριστές, καθώς τα πολυμέσα υψηλής ποιότητας, τους παρέχουν καλύτερη εποπτεία ενός ζητήματος.



Technical Priority: (3/5) Η μη ικανοποίηση αυτής της απαίτησης δεν επηρεάζει σε μεγάλο βαθμό τη λειτουργία του συστήματος. Ωστόσο πρέπει να αναπτυχθεί με τρόπο ώστε να υποστηρίζει την απαίτηση αυτή.

Stability: (4/5) Η απαίτηση αυτή ενδέχεται να μεταβληθεί στο μέλλον καθώς τα κριτήρια υψηλής ποιότητας πολυμέσων αλλάζουν.

1.3.5 Απαιτήσεις πολιτικής (Policy - legal, corporate)

<ΜΛΑ-5> Το σύστημα πρέπει να πληροί τον κανονισμό GDPR.

Περιγραφή: Τα δεδομένα των χρηστών πρέπει να διατηρούνται σύμφωνα με τον κανονισμό GDPR και οι χρήστες να ενημερώνονται και να συμφωνούν με την πολιτική διαχείρισης των δεδομένων τους.

User Priority (4/5): Η απαίτηση αυτή είναι αρκετά σημαντική για τον χρήστη, καθώς ανησυχεί για τον τρόπο διαχείρισης των προσωπικών του δεδομένων.

Technical Priority(3/5): Το σύστημα δεν επηρεάζεται άμεσα από την ικανοποίηση ή όχι αυτής της απαίτησης. Ωστόσο πρέπει να δίνεται προσοχή στην τήρηση τους, καθώς τυχούσα παραβίαση της μπορεί να οδηγήσει σε νομικά ζητήματα που πιθανόν να υποχρεώσουν τον τερματισμό της λειτουργίας του συστήματος.

Stability: (4/5) Η συμφωνία με τις επικρατούσες πολιτικές προστασίας δεδομένων είναι πρωταρχικό μέλημα για το σύστημά μας, συνεπώς εάν αυτές μεταβληθούν, θα ενημερωθεί ανάλογα και το σύστημα.

1.4 Λεξικογραφικοί Προσδιορισμοί

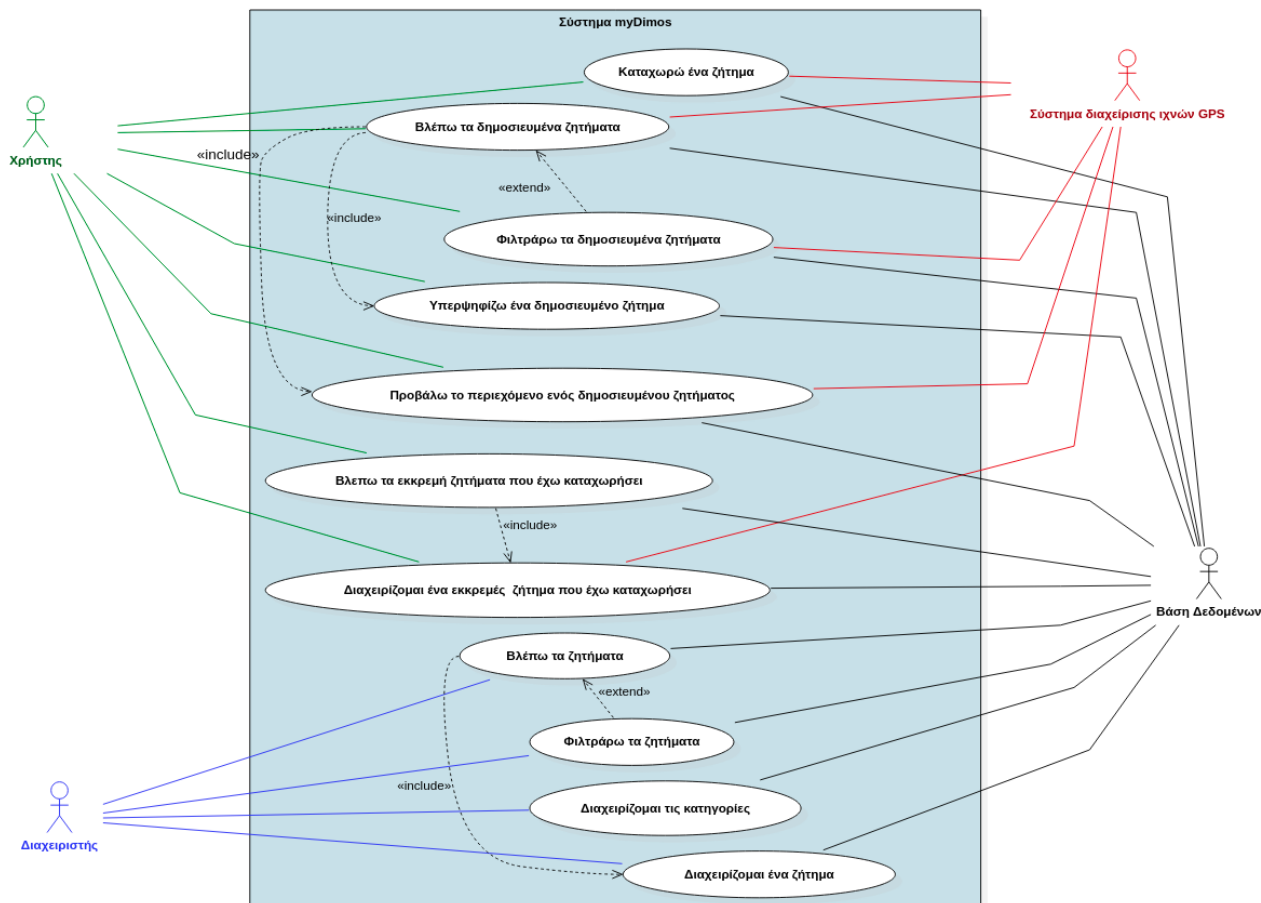
- **Ζήτημα(Issue):** Η τυποποιημένη μορφή γνωστοποίησης ενός προβλήματος του δήμου στο σύστημα από το χρήστη, π.χ. λακκούβα στον δρόμο, σπασμένη αποχέτευση κλπ.
- **Καταχωρώ ζήτημα(Sumbit issue):** Συμπληρώνω και υποβάλλω μία ηλεκτρονική φόρμα καταχώρησης με σκοπό την γνωστοποίηση ενός προβλήματος του δήμου στο σύστημα. Απώτερος στόχος είναι η ενημέρωση του δήμου και η επίλυση του προβλήματος από αυτόν.
- **Φόρμα καταχώρησης(Sumbission form):** Μία φόρμα με τα στοιχεία που απαιτούνται για την καταχώρηση ενός ζητήματος στο σύστημα: Τίτλος, επιλογή κατηγορίας, Σύντομη περιγραφή του προβλήματος, Διεύθυνση που εντοπίζεται και οπτικοακουστικό υλικό που αποδεικνύει την ύπαρξη του.
- **Δημοσιευμένο ζήτημα(Published Issues):** Το ζήτημα το οποίο έχει περάσει τον έλεγχο των διαχειριστών για την ικανοποίηση των κριτηρίων καταλληλότητας και είναι ορατό και στους χρήστες.
- **Φιλτράρω(Filter):** Περιορίζω τα ζητήματα τα οποία βλέπω εφαρμόζοντας φίλτρα βάσει ρητής επιλογής (μαρκάρισμα), αλφαριθμητικού(αναζήτηση), κατηγορίας, κατάστασης, γεωγραφικής απόστασης-τοποθεσίας, χρόνου καταχώρησης και ψήφων.
- **Κατάσταση(Status):** Κατάσταση ονομάζεται το στάδιο διαχείρισης στο οποίο βρίσκεται ένα ζήτημα. Υπάρχουν τρεις πιθανές καταστάσεις: εκκρεμές, δημοσιευμένο και επιλυμένο.
- **Εκκρεμές ζήτημα(Pending issue):** Ένα ζήτημα που έχει καταχωρηθεί από έναν χρήστη και περιμένει την έγκριση κάποιου διαχειριστή.
- **Επιλυμένο ζήτημα(Solved issue):** Ένα ζήτημα το πρόβλημα του οποίου έχει επιλυθεί.
- **Κατηγορία(Category):** Ομάδα ζητημάτων των οποίων τα προβλήματα σχετίζονται μεταξύ τους και συνήθως απαιτούν παρόμοια αντιμετώπιση, από τις ίδιες υπηρεσίες του δήμου. Οι κατηγορίες διαμορφώνονται αποκλειστικά από τους διαχειριστές του συστήματος με βάση τις ανάγκες του κάθε δήμου.
- **Διαχειρίζομαι ένα ζήτημα(Manage an issue):** επεξεργάζομαι τον τίτλο, ή/και την περιγραφή, το οπτικοακουστικό υλικό, την τοποθεσία, την κατάσταση (μόνο αν είμαι διαχειριστής), την κατηγορία ενός ζητήματος, ή το διαγράψω από το σύστημα.



- **Διαχειρίζομαι μία κατηγορία(Manage a category):** Δημιουργώ μία νέα κατηγορία, επεξεργάζομαι ή διαγράφω μία υπάρχουσα.
- **Χρόνος καταχώρησης(Sumbission time):** Η χρονική στιγμή όπου ο χρήστης καταχωρεί ένα ζήτημα στο σύστημα.
- **Υπερψηφίζω(Uprvote):** Ψηφίζω ένα ζήτημα με την χρήση κάποιου like button με σκοπό να δείξω ότι επιβεβαιώνω την ύπαρξη και την σημαντικότητα του αντίστοιχου προβλήματος.
- **Ψήφοι(Votes):** Είναι το καθοριζόμενο από τους χρήστες μέτρο της σημαντικότητας ενός ζητήματος.
- **Κριτήρια καταλληλότητας(Appropriateness crietiria):** Τα κριτήρια που πρέπει να πληροί ένα ζήτημα για να δημοσιοποιηθεί. Το ζήτημα δεν θα πρέπει να περιέχει περιεχόμενο παράνομο, άσεμνο, άσχετο, προσβλητικό (ρατσιστικό και σεξιστικό), spam, ή περιεχόμενο που έχει ήδη καταχωρηθεί.

2 Σενάρια Χρήσης

2.1 Διάγραμμα σεναρίων χρήσης



Σχήμα 2.1: Διάγραμμα σεναρίων χρήσης

2.2 Feature: Issue submission

As a user

So that my problems are reported to and solved by the municipal authorities

I want to submit an issue

Scenario: Open the issue submission form

Given I am in my home page

When I press the “Submit an issue” button

Then the issue submission form shows up

And all the fields are blank

Scenario: Submit an issue

Given I have filled in all the issue’s details in the submission form

| Title | ‘chuckhole’ | Max 80 chars |

| Description | ‘A big chuckhole is on the right side of the road’ | Max 500 chars |



Category	'Road map issue'	Drop down list
Address	'Tsimiski 39'	Can be inputted by GPS
Audiovisual material	Photograph or video	Max 100MB size

When I press the "Submit" button

Then I return to my home page

Scenario: Details missing or invalid

Given I am filling up the submission form

But a field is empty or has invalid content

When I press the "Submit" button

Then the field turns red

And next to it I see a message "Please fill in this field with a valid value"

And next to it I see a hint of the validation criteria

And I can continue filling in the submission form

Scenario: Cancel the submission

Given I am filling in the submission form

When I press the cancel button, or I attempt to close the form

Then I see the message "Do you wish to close the submission form? Your data will be lost"

And I am offered the choice to resume filling in the form or close the form.

Scenario: Resume filling in the submission form

Given The leave confirmation dialog has popped up

When I press "No"

Then I can continue filling in the submission form

Scenario: Leave the submission form

Given The leave confirmation dialog has popped up

When I press "Yes"

Then I return to my home page

2.3 Feature: View issues list (user)

As a user

So that I can track the problems of my municipality

I want to see a list of all published issues

Scenario: The system contains published issues

Given I am logged in

And the system contains published issues

When I go to my home page

Then I see a list with every published issue with the details

| Title | 'chuckhole' | Max 80 chars |

| Status | ['Pending', 'Published', 'Solved'] |

| Votes | 341 |

| Category | 'Road map issue' | Drop down list |

| Address | 'Tsimiski 39' | Can be inputted by GPS |

| Distance from user | 3.7km | Only visible if GPS is enabled |

And I see the number of published issues

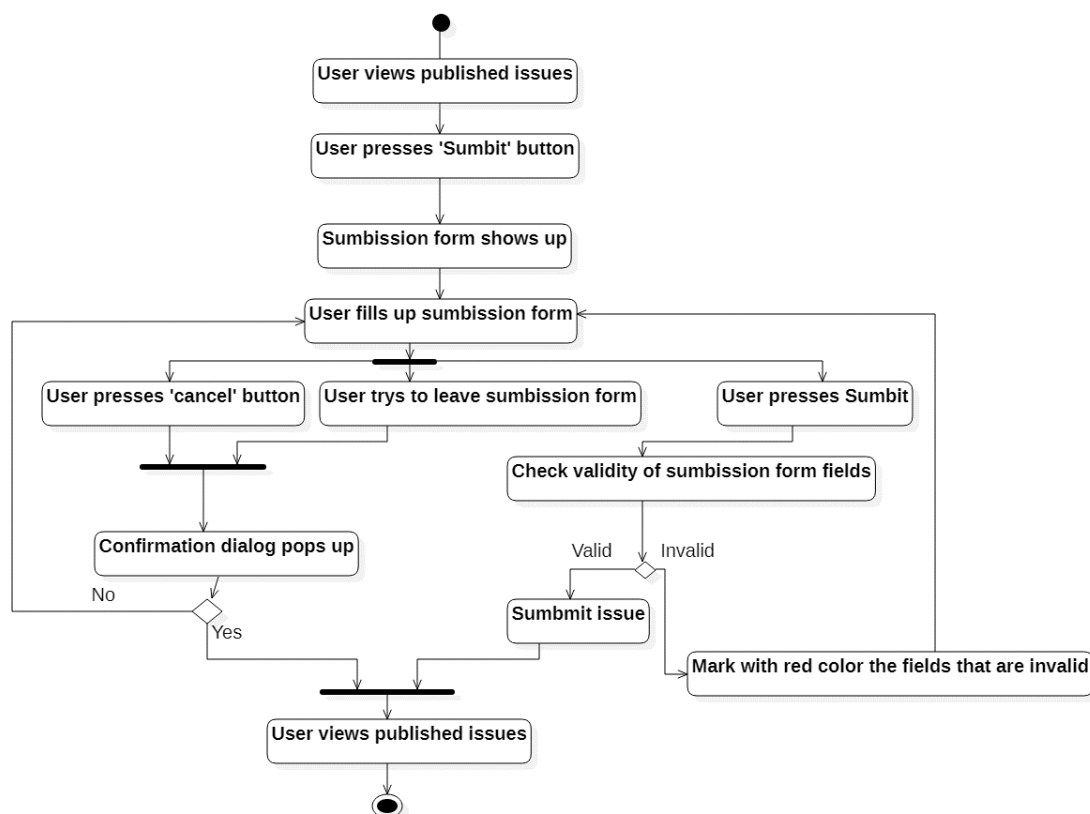
Scenario: The system contains no published issues

Given I am logged in

And the system doesn't contain any published issue

When I go to my home page

Then I see a text "No issues exist" in the list's place



Σχήμα 2.2: Ο χρήστης καταχωρεί ζήτημα

2.4 Feature: Issue upvote

As a user

So that I can point out the importance of an issue

I want to upvote an issue

Scenario: Upvoting in list view

Given I see a list of issues

And I see an issue that I haven't upvoted with <old> votes

When I press the upvote button of the issue

Then the issue is upvoted by me

And the issue has <new> votes

Examples:

| old | new |



| 15 | 16 |

Scenario: Cancelling vote in list view

Given I see a list of issues

And I see an issue that I have upvoted with <old> votes

When I press the upvote button of the issue

Then the issue is not upvoted by me

And I see the <new> number of votes

Examples:

| old | new |

| 16 | 15 |

Scenario: Upvoting in content view

Given I have opened the detailed view of an issue

And I haven't upvoted the issue

When I press the upvote button

Then the issue is upvoted by me

And I see the new number of votes

Scenario: Cancelling vote in content view

Given I have opened the detailed view of an issue

And I have upvoted the issue

When I press the upvote button

Then the issue is not upvoted by me

And I see the new number of votes

2.5 Feature: Issue content view (user)

As a user

So that I learn more about a specific issue

I want to view the full information associated with an issue

Scenario: open the content view page

Given I see a list of issues

When I press an issue

Then the content view page opens

And I can see the full information of an issue

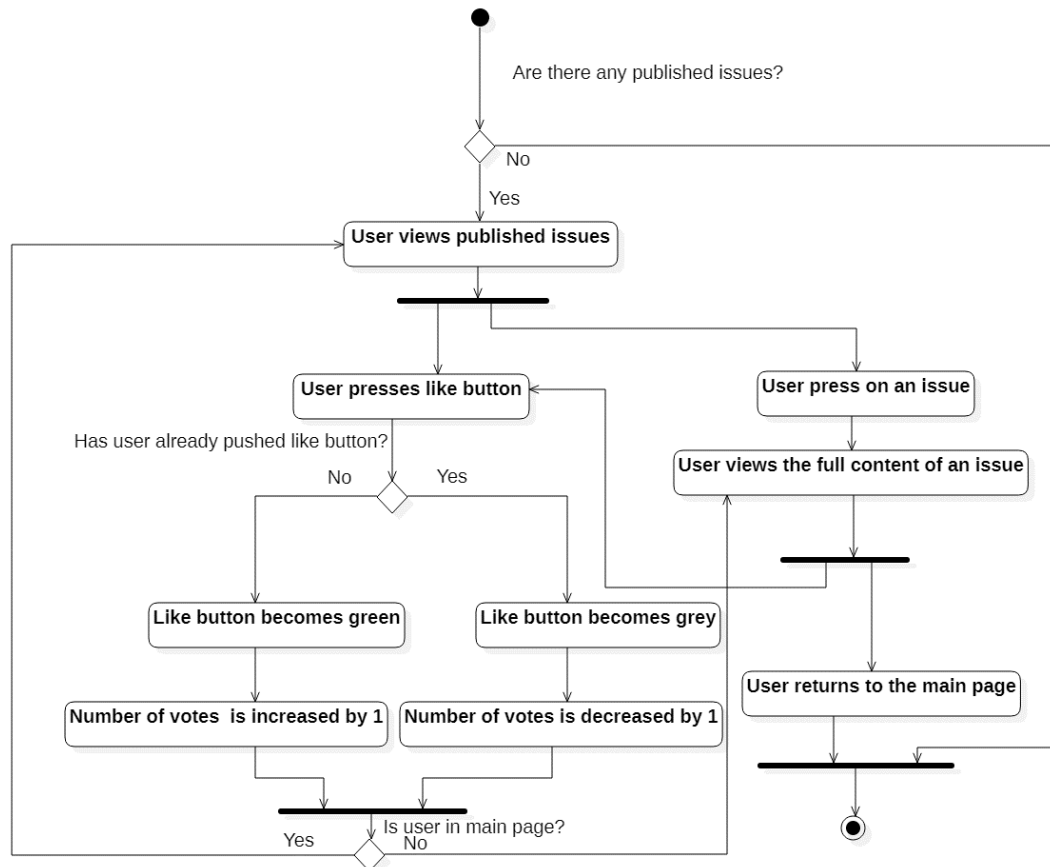
Scenario: close the content view page

Given I am in content view

When I press the "Back" button

Then the content view page closes

And I return to the previous issue list



Σχήμα 2.3: Ο χρήστης βλέπει, υπερψηφίζει και προβάλει τα ζητήματα.

2.6 Feature: Issues filtering(user)

Feature: Issues filtering(user)

As a user

So that I can search for specific issues

I want to filter the issues visible in my list

Scenario: Search by string

Given I see a list of issues

And I have inputted a string in the search box

When I press the “magnifying glass” icon

Then the search string filter is added

And the list is updated

Scenario: Open the filters page

Given I am in my home page

When I press the “filters” icon

Then I go to the filters page

And The filters have their previous state

And I see the same search string as in my home page

Scenario: Change filters



Given I am in the filters page

And I have added or removed search filters:

| Category | 'Road map issues' | checkbox |

| Status | ['Published', 'Solved'] | checkbox |

| Distance radius | 2.5km | spinner |

| Submitted from | 23/11/2019 | date selection calendar |

| Submitted until | 15/2/2020 | date selection calendar |

| Search string | 'Traffic light' | |

When I press "Apply filters"

Then I return to my home page

And I see the same search string as in the filters page

And I see the updated list of issues

Scenario: There are issues matching my filters

Given I am in my home page

And there are issues matching my filters

Then I see a list of issues matching my new filters

And I see the number of the issues

Scenario: No issues match the filters

Given I am in my home page

And no issues match my new filters

Then I see a message "No issues found" in the list's place

Scenario: Remove every filter

Given I am in the filters page

When I press the "Cancel" button

Then Every filter is blank

2.7 Feature: View my pending issues

As a user

So that I can revise my pending issues

I want to view a list of all the pending issues I have submitted

Scenario: Go to my pending issues page

Given I am in my home page

When I press the "My pending issues" button

Then I go to my pending issues page

Scenario: I have pending issues

Given I am in my pending issues page

And the systems contains issues I submitted and have "pending" status

Then I see a list of all the issues I have submitted and have "pending" status

Scenario: I have no pending issues

Given I am in my pending issues page

And the systems doesn't contain issues I submitted and have "pending" status

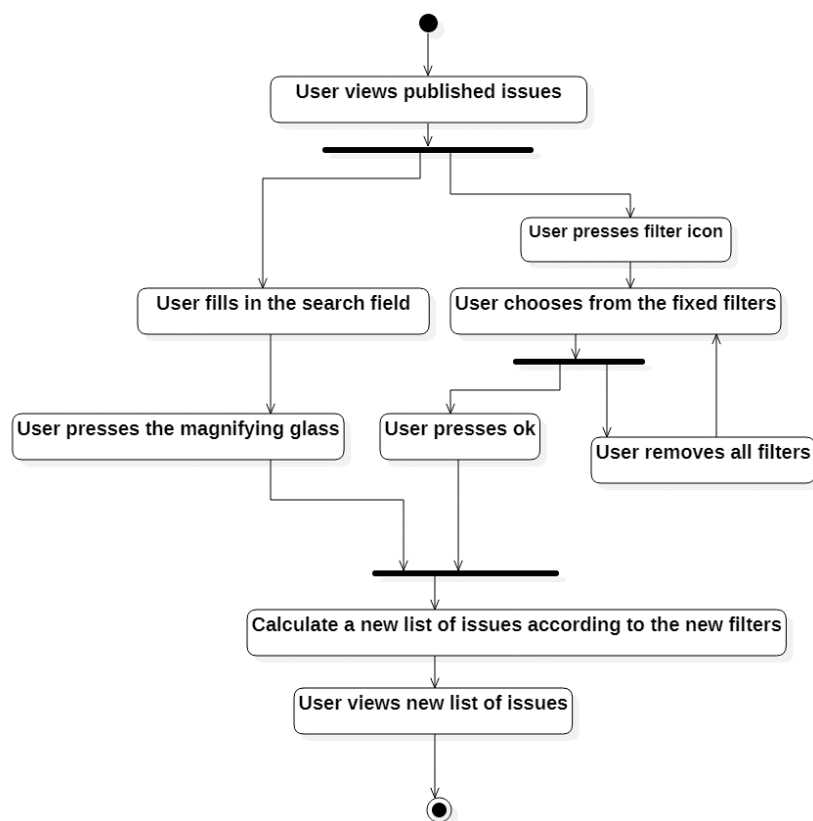
Then I see a message “No pending issues” in the place of the list.

Scenario: Leave the my pending issue page

Given I am in my pending issues page

When I press “Back”

Then I go to my home page



Σχήμα 2.4: Ο χρήστης φιλτράρει τα δημοσιευμένα ζητήματα.

2.8 Feature: Issue management (user)

As a user

So that I can correct or change the issues I have submitted

I want to manage an issue I have submitted and has “pending” status

Scenario: open the issue management page

Given I see a non-empty list of issues I have submitted and have “pending” status

When I press an issue

Then the issue management page opens

Scenario: Delete an issue

Given I am in the issue management page

When I press “Delete issue”

Then I see a dialog “Are you sure you want to delete the issue?”

And I am offered the choice to confirm deletion or not

Scenario: Confirm deletion



Given I see the deletion confirmation dialog

When I press “Yes”

Then I return to my pending issues page

And the issue is deleted

And I see a message “Issue was deleted”

Scenario: Abort deletion

Given I see the deletion confirmation dialog

When I press “No”

Then I return to my pending issues page

Scenario: Successful editing

Given I am in the issue management page

And I have edited some fields of the issue

And all the fields are valid

When I press “Save”

Then I return to my pending issues page

And the issue is edited

And I see a message “Issue was edited”

Scenario: Unsuccessful editing

Given I am in the issue management page

And I have edited a field of the issue

But the field is empty or has invalid content

When I press the “Save” button

Then the field turns red

And next to it I see a message “Please fill in this field with a valid value”

And next to it I see a hint of the validation criteria

And I can continue editing the issue

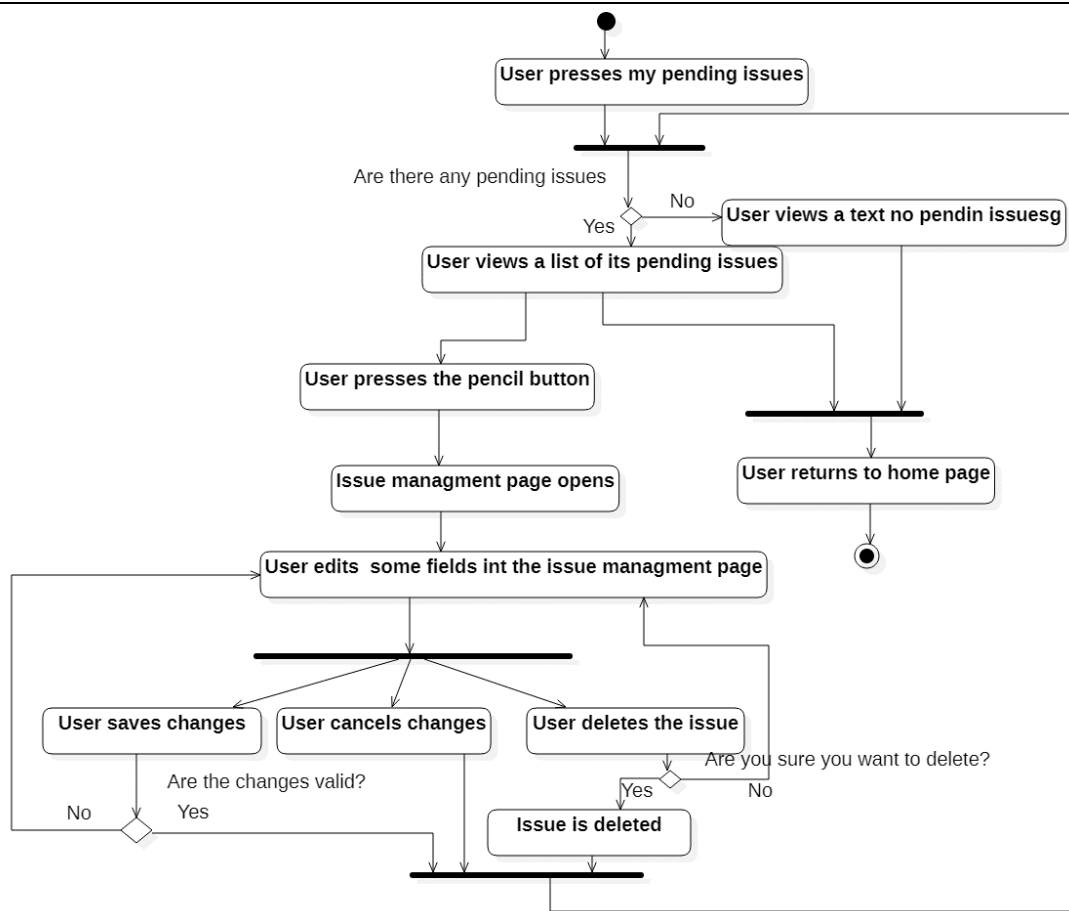
Scenario: Cancel editing

Given I am in the issue management page

When I press “Cancel”

Then the issue is not changed

And I return to my pending issues page



Σχήμα 2.5: Ο χρήστης βλέπει και επεξεργάζεται τα εκκρεμή του ζητήματα.

2.9 Feature: View issue list (admin)

As an admin

So that I can track and manage the problems of my municipality

I want to see a list of all issues

Scenario: The system contains issues

Given I am logged in

And the system contains issues

When I go to my home page

Then I see a list with every issue with the details

| Title | 'chuckhole' | Max 80 chars |

| Status | ['Pending', 'Published', 'Solved'] |

| Votes | 341 |

| Description | 'A big chuckhole is on the right side of the road' | Max 500 chars |

| Category | 'Road map issue' | Drop down list |

| Address | 'Tsimiski 39' | Can be inputted by GPS |

| Media | Image |

And I see the number of issues



Scenario: The system contains no issues

Given I am logged in

And the system doesn't contain any issue

When I go to my home page

Then I see a message "No issues exist" in the list's place

2.10 Feature: Issue filtering (admin)

As an admin

So that I can concentrate on managing a subset of issues

I want to filter the issues in my list

Scenario: Search by string

Given I see a list of issues

And I have inputted a string in the search box

When I press the "magnifying glass" icon

Then the search string filter is added

And the list is updated

Scenario: There are issues matching my filters

Given I am in my home page

When I add an extra or remove a search filter:

| Category | 'Road map issues' | checkbox |

| Status | ['Pending', 'Published', 'Solved'] | checkbox |

| Submitted from | 23/11/2019 | date selection calendar |

| Submitted until | 15/2/2020 | date selection calendar |

| Address | 'Tsimiski 39' |

| Search string | 'Traffic light' | |

And there are issues matching my new filters

Then I see a list of issues matching my new filters

And I see the number of the issues

Scenario: No issues match the filters

Given I am in my home page

When I add an extra or remove a search filter

And no issues match my new filters

Then I see a message "No issues found" in the list's place

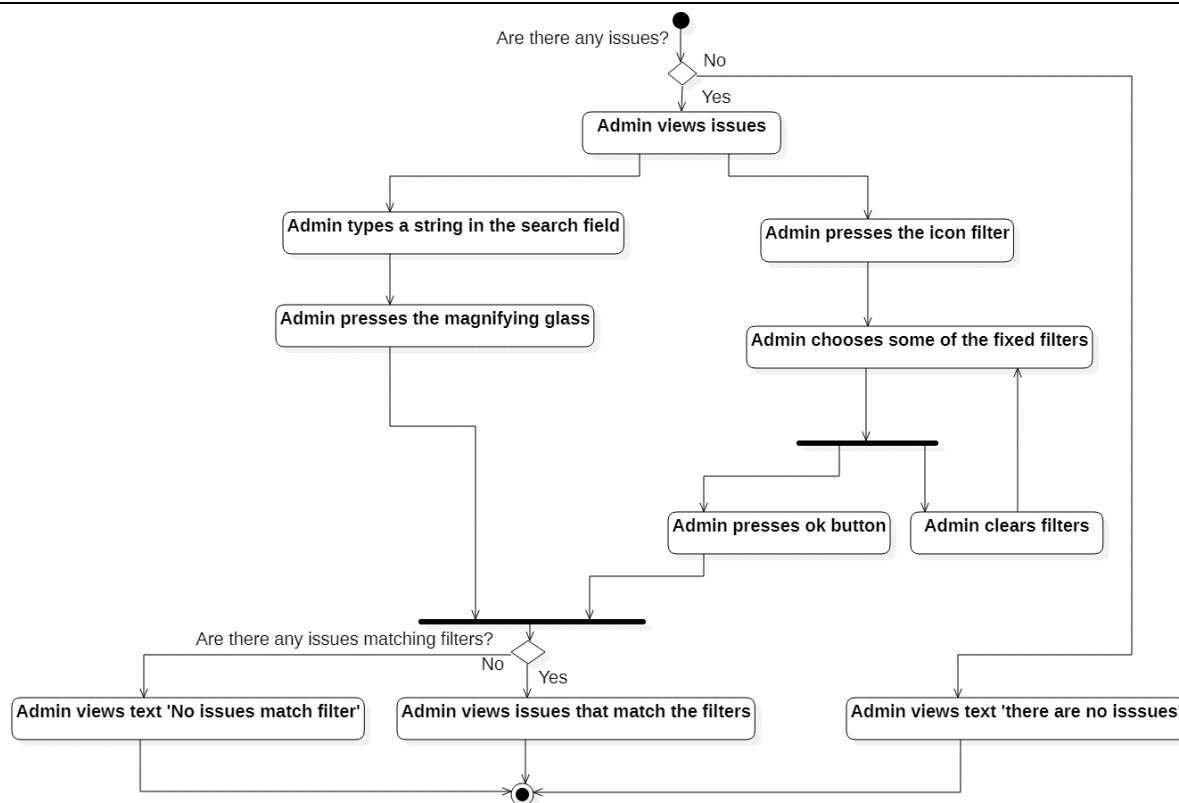
Scenario: Remove every filter

Given I am in my home page

When I press the "Clear filters" button

Then I see a list of all issues

And I see the number of the issues



Σχήμα 2.6: Ο διαχειριστής βλέπει και φιλτράρει τα ζητήματα του

2.11 Feature: Category management

As an admin

So that I adjust the system to my municipality's needs

I want to manage a category

Scenario: Go to the category management page

Given I am in my home page

When I press "Manage categories"

Then I go to the category management page

Scenario: Create a category

Given I am in the category management page

When I press "New Category"

Given I give a unique name for the new category

Then I return to the updated category management page

And I see a message "Category created"

Scenario: Edit an existing category

Given I am in the category management page

And I have selected a category

When I press "Edit" Button

Given I give a unique name for the new category

Then I return to the updated category management page

And I see a message “Category edited”

Scenario: Duplicate name

Given I am creating a new or editing an existing category

But I give a non-unique name

Then I see a message “This name already exists”

And I can give another name.

Scenario: Delete a category

Given I am in the category management page

And I have selected a category

When I press “Delete”

Then I see a dialog “Are you sure you want to delete the category <category name>? Every issue of this category will then belong the “Other” category.

And I have the choice of confirming or aborting the deletion

Scenario: Confirm deletion

Given I am at the category deletion confirmation dialog

When I press “Yes”

Then the category is deleted

And I return to the category management page

And I see a message “Category deleted”

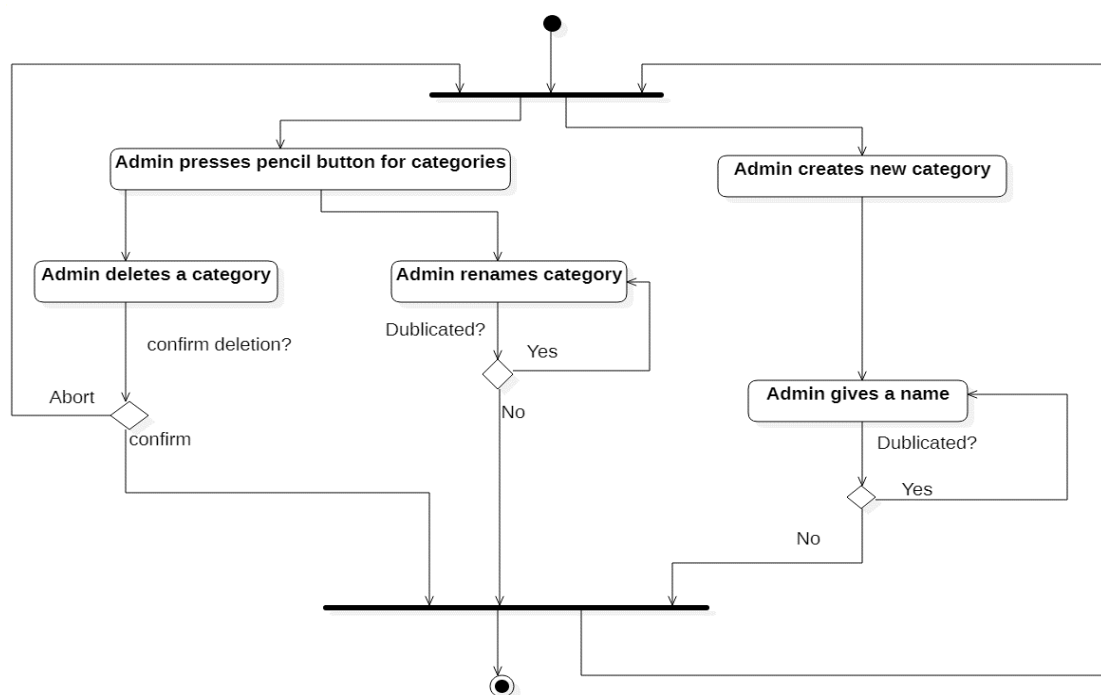
And every issue of that category belongs to the “Other” category.

Scenario: Abort deletion

Given I see the deletion confirmation dialog

When I press “No”

Then I return to the category management page



Σχήμα 2.7: Ο διαχειριστής διαχειρίζεται μία κατηγορία.



2.12 Feature: Issue management (admin)

As an admin

So that I can manage the problems of my municipality

I want to manage an issue

Scenario: open the issue management page

Given I see a non-empty list of issues

When I press the “Edit” button on an issue

Then the issue management page opens

Scenario: Delete an issue

Given I am in the issue management page

When I press “Delete issue”

Then I see a dialog “Are you sure you want to delete the issue?”

And I am offered the choice to confirm deletion or not

Scenario: Confirm deletion

Given I see the deletion confirmation dialog

When I press “Yes”

Then I return to my pending issues page

And the issue is deleted

And I see a message “Issue was deleted”

Scenario: Abort deletion

Given I see the deletion confirmation dialog

When I press “No”

Then I return to the issue management page

Scenario: Successful editing

Given I am in the issue management page

And I have edited some fields of the issue

And there are no blank or invalid fields

When I press “Save”

Then I see my previous issue list

And the issue is edited

And I see a message “Issue was edited”

Scenario: Unsuccessful editing

Given I am in the issue management page

And I have edited a field of the issue

But the field is empty or has invalid content

When I press the “Save” button

Then the field turns red

And next to it I see a message “Please fill in this field with a valid value”

And next to it I see a hint of the validation criteria

And I can continue editing the issue

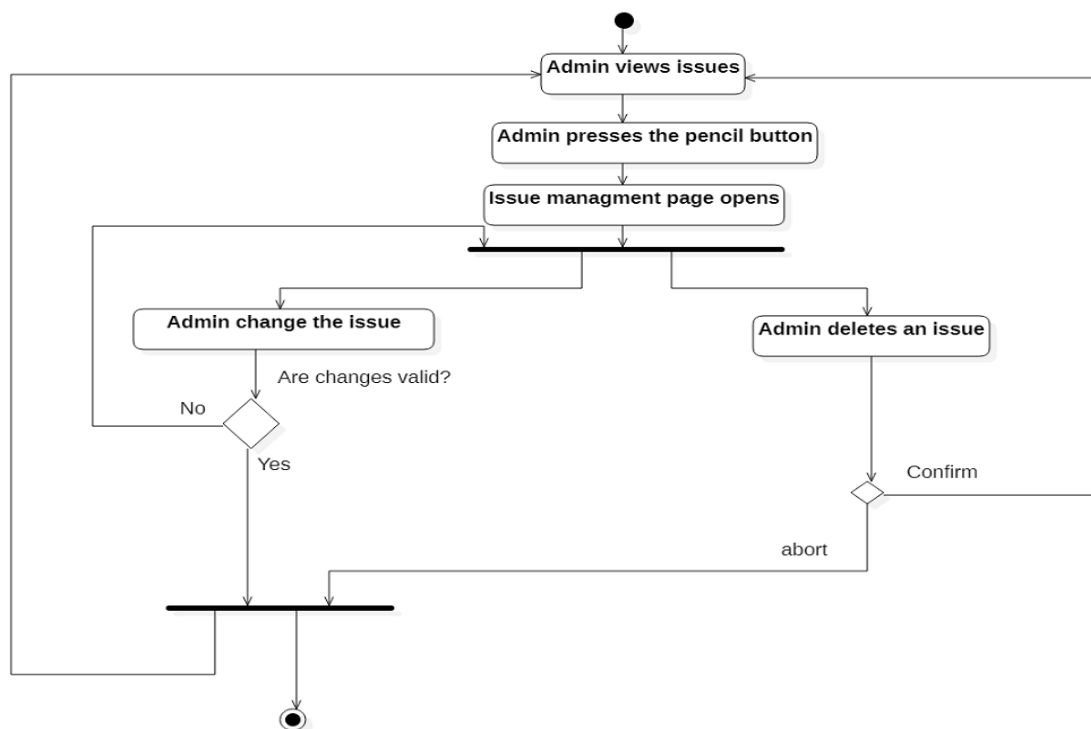
Scenario: Cancel editing

Given I am in the issue management page

When I press “Cancel”

Then The issue is not changed

And I see my previous issue list



Σχήμα 2.8: Ο διαχειριστής διαχειρίζεται ένα ζήτημα

3 Επιδεικτικά γραφικά παράθυρα διεπαφής

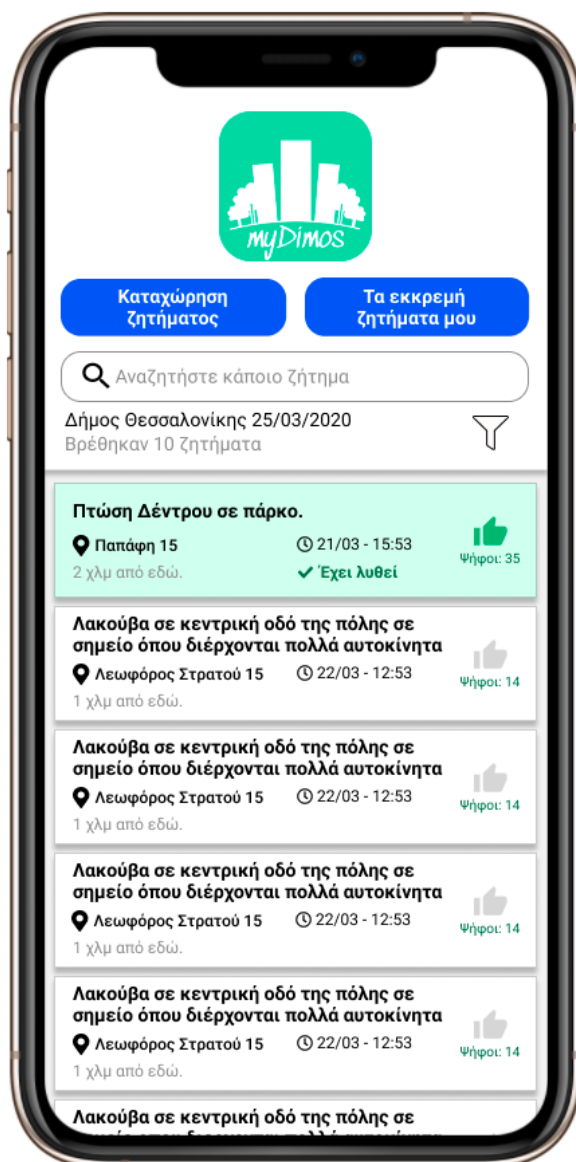
3.1 Αρχική σελίδα χρήστη



Σχήμα 3.1: (Αριστερά) Η αρχική σελίδα του Χρήστη - Χωρίς Ψηφισμένο ζήτημα

Σχήμα 3.2: (Δεξιά) Η αρχική σελίδα του Χρήστη - Ψηφισμένο το πρώτο ζήτημα

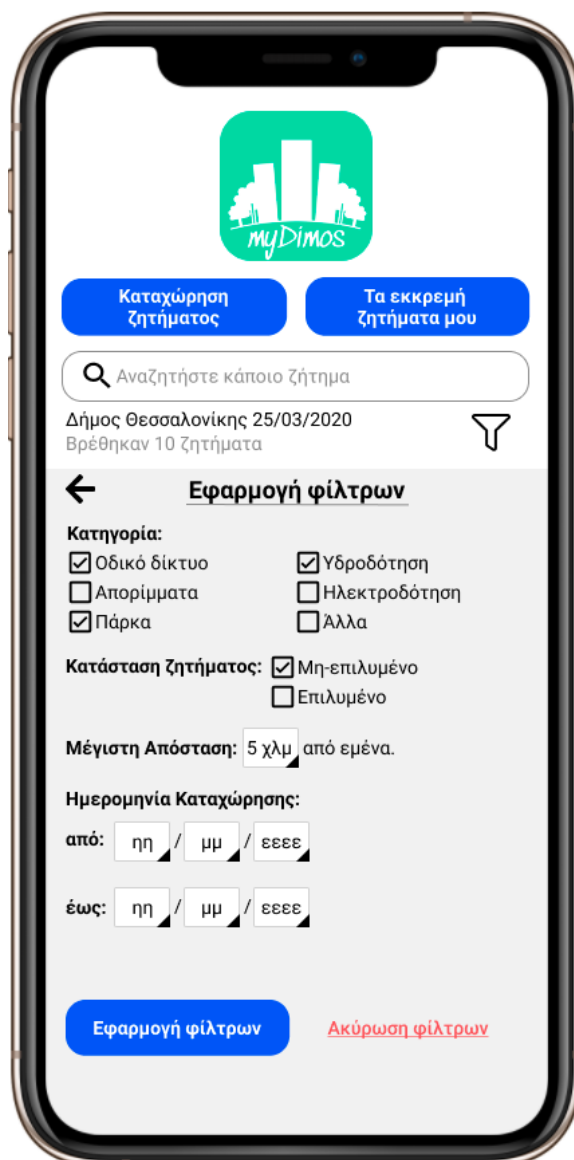
Οι παραπάνω οθόνες δείχνουν την αρχική σελίδα που βλέπει ο χρήστης μόλις συνδεθεί στην εφαρμογή myDimos. Ο χρήστης βλέπει την λίστα των δημοσιευμένων ζητημάτων, μαζί με κάποια πληροφορία σχετική με αυτές. Κανένα ζήτημα από αυτά που φαίνονται δεν είναι επιλυμένο. Η δεύτερη οθόνη προκύπτει στην περίπτωση που ο χρήστης έχει υπερψηφίσει το πρώτο ζήτημα.



Σχήμα 3.3: Η αρχική σελίδα του Χρήστη - Λυμένο το πρώτο ζήτημα

Η παραπάνω οθόνη(Σχήμα 3.3) αποτελεί την αρχική σελίδα που βλέπει ο χρήστης μόλις συνδεθεί στην εφαρμογή myDimos. Συγκεκριμένα στην παραπάνω οθόνη ο χρήστης βλέπει την λίστα των δημοσιευμένων ζητημάτων, μαζί με κάποια πληροφορία τους. Επίσης, το πρώτο ζήτημα έχει λυθεί (έχει κατάσταση επιλυμένο), ενώ έχει υπερψηφιστεί και από τον χρήστη.

3.2 Σελίδα Εφαρμογής φίλτρων χρήστη



Σχήμα 3.4: Σελίδα εφαρμογής φίλτρων χρήστη

Το σχήμα 3.4 απεικονίζει την οθόνη-σελίδα, μέσω της οποίας ο χρήστης μπορεί να εφαρμόσει φίλτρα ώστε να περιορίσει τα ζητήματα που βλέπει στην αρχική του σελίδα.

3.3 Σελίδα Καταχώρησης Ζητήματος



Δήμος Θεσσαλονίκης 25/03/2020

← **Καταχώρηση νέου ζητήματος**

***Τίτλος:**
Εισάγετε έναν τίτλο για το ζήτημα.

***Κατηγορία:**
--Επιλέξτε Κατηγορία--

***Περιγραφή:**
Εισάγετε μια σύντομη περιγραφή για το ζήτημα.

***Διεύθυνση:**
Εισάγετε την διεύθυνση του ζητήματος.
ή εισάγετε το στίγμα απο το GPS σας
Εισαγωγή στίγματος GPS

Πολυμέσα:
Επιλέξτε ένα αρχείο εικόνας ή βιντεο σχετικό με το ζήτημα.
Επιλογή αρχείου Δεν έχουν επιλεγεί αρχεία.

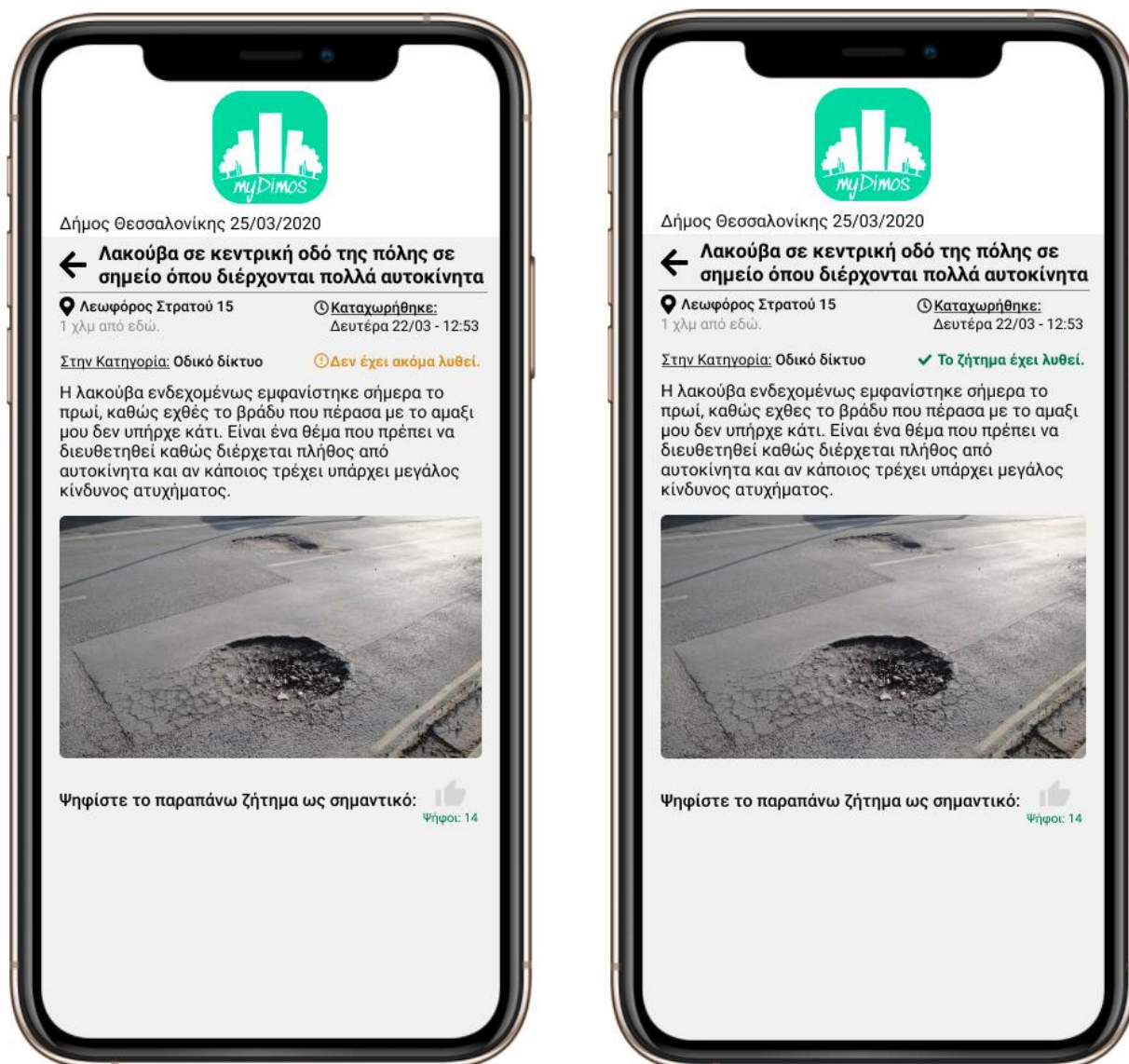
*** Υποχρεωτικά πεδία.**

Υποβολή

Σχήμα 3.5: Σελίδα Καταχώρησης ζητήματος

Στο σχήμα 3.5 φαίνεται η οθόνη που αποτελεί την σελίδα μέσω της οποίας ο χρήστης μπορεί να καταχωρήσει ένα νέο ζήτημα στο σύστημα.

3.4 Σελίδα Ζητήματος



Σχήμα 3.6: (Αριστερά) Σελίδα μη επιλυμένου ζητήματος

Σχήμα 3.7: (Δεξιά) Σελίδα επιλυμένου ζητήματος

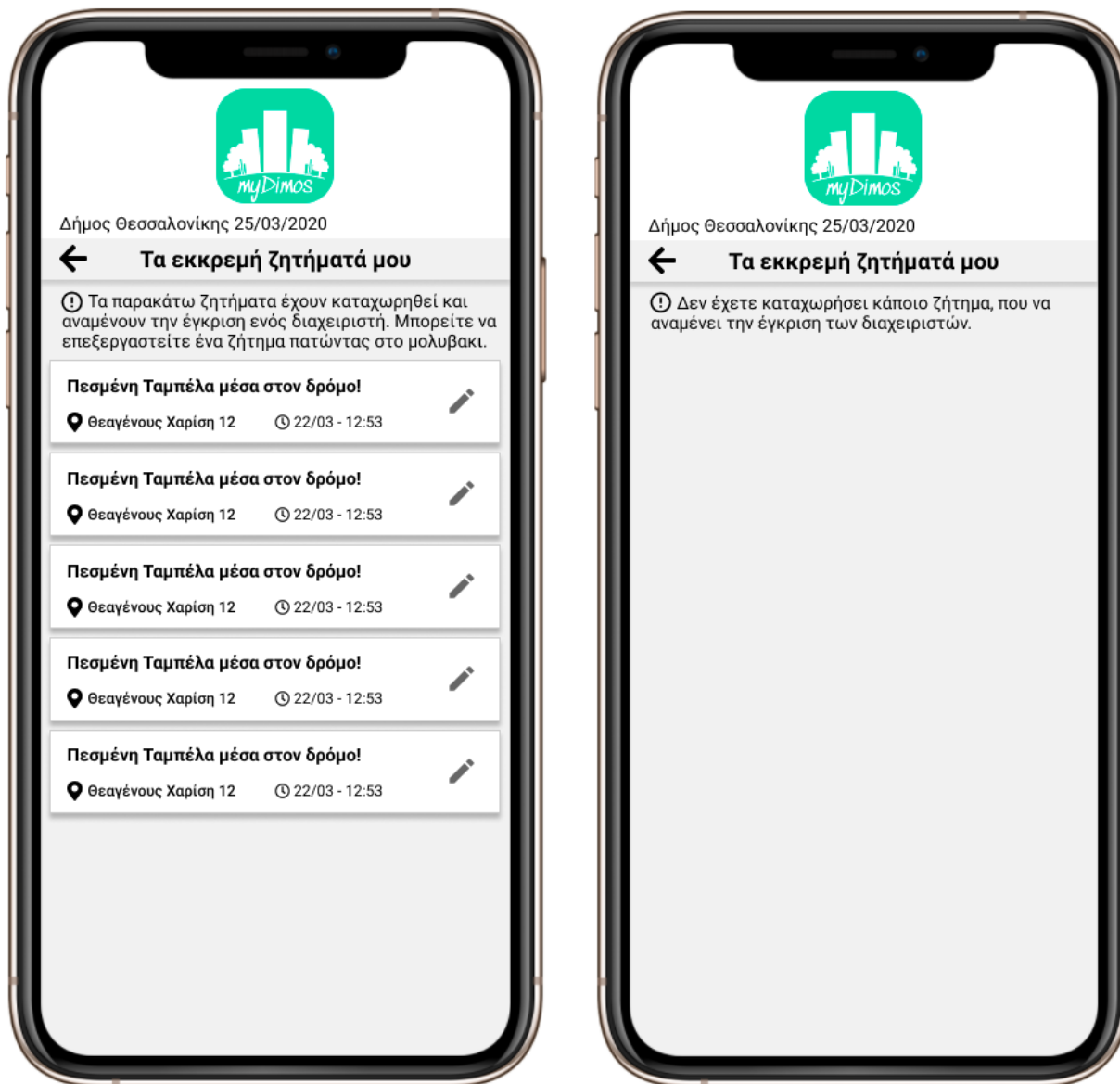
Στο σχήμα 3.6 φαίνεται η σελίδα μέσω της οποίας ο χρήστης μπορεί να δει το πλήρες περιεχόμενο ενός δημοσιευμένου ζητήματος και το ζήτημα δεν έχει ακόμα επιλυθεί. Στο σχήμα 3.7 φαίνεται η σελίδα μέσω της οποίας ο χρήστης μπορεί να δει το πλήρες περιεχόμενο ενός δημοσιευμένου ζητήματος και το ζήτημα έχει επιλυθεί.



Σχήμα 3.8: Σελίδα υπερψηφισμένου ζητήματος

Στο σχήμα 3.8 το ζήτημα δεν έχει επιλυθεί ακόμα, το έχει όμως υπερψηφίσει ο χρήστης.

3.5 Σελίδα Προβολής εκκρεμών ζητημάτων χρήστη



Σχήμα 3.9: Σελίδα προβολής εκκρεμών ζητημάτων χρήστη

Σχήμα 3.10: Σελίδα προβολής εκκρεμών ζητημάτων χρήστη - κενή

Οι παραπάνω οθόνες δείχνουν την σελίδα προβολής εκκρεμών ζητημάτων, όταν ο χρήστης έχει καταχωρήσει ζητήματα που περιμένουν την έγκριση κάποιου διαχειριστή (σχήμα 3.9) και όταν δεν υπάρχουν τέτοια ζητήματα (σχήμα 3.10).

3.6 Σελίδα Επεξεργασίας Εκκρεμούς Ζητήματος - χρήστης



Δήμος Θεσσαλονίκης 25/03/2020

← **Επεξεργασία Ζητήματος**

*Τίτλος:
Πεσμένη Ταμπέλα μέσα στον δρόμο!

*Κατηγορία:
Οδικό δίκτυο

*Περιγραφή:
Η ταμπέλα, που βρισκόταν δεξιά του δρόμου έχει σπάσει και έχει πέσει στο πλάι του δρόμου, καθιστώντας δύσκολη την διέλευση αμαξιών.

*Διεύθυνση:
Θεαγένους Χαρίση 12
ή εισαγάγετε το στίγμα απο το GPS σας
Εισαγωγή στίγματος GPS

Πολυμέσα:
Επιλέξτε ένα αρχείο εικόνας ή βίντεο σχετικό με το ζήτημα.
Επιλογή αρχείου 123Image_1232020.jpg

* Υποχρεωτικά πεδία.

Αποθήκευση Διαγραφή

Σχήμα 3.11: Σελίδα επεξεργασίας εκκρεμούς ζητήματος - χρήστης

Η παραπάνω οθόνη αποτελεί την σελίδα, μέσω της οποίας ο χρήστης μπορεί να διαχειριστεί ένα εκκρεμές ζήτημα που έχει καταχωρήσει ο ίδιος στο σύστημα.



3.7 Αρχική οθόνη διαχειριστή - λίστα ζητημάτων διαχειριστή

Σελίδα Διαχείρισης Ζητημάτων
Δήμος Θεσσαλονίκης 25/03/2020

Βρέθηκαν 15 ζητήματα

Τίτλος: Λακούβα σε κεντρική οδό της πόλης σε σημείο όπου διερχονται πολλά αυτοκίνητα

Περιγραφή: Η λακούβα ενδεχομένως εμφανίστηκε σήμερα το πρωί, καθώς εχθες το βράδυ που πέρασα με το αμάξι μου δεν υπήρχε κάτι. Είναι ένα θέμα που πρέπει να διευθετηθεί καθώς διέρχεται πλήθος από αυτοκίνητα και αν κάποιος τρέχει υπάρχει μεγάλος κίνδυνος ατυχήματος.

Αριθμός Ψήφων: 14

Κατηγορία: Οδικό δίκτυο

Κατάσταση: Δημοσιευμένο

Διεύθυνση: Λεωφόρος Στρατού 15

Καταχωρήθηκε: Δευτέρα 22/03 - 12:53

Πολυμέσα:

[Επεξεργασία](#)

Τίτλος: Λακούβα σε κεντρική οδό της πόλης σε σημείο όπου διερχονται πολλά αυτοκίνητα

Περιγραφή: Η λακούβα ενδεχομένως εμφανίστηκε σήμερα το πρωί, καθώς εχθες το βράδυ που πέρασα με το αμάξι μου δεν υπήρχε κάτι. Είναι ένα θέμα που πρέπει να διευθετηθεί καθώς διέρχεται πλήθος από αυτοκίνητα και αν κάποιος τρέχει υπάρχει μεγάλος κίνδυνος ατυχήματος.

Αριθμός Ψήφων: 14

Κατηγορία: Οδικό δίκτυο

Κατάσταση: Δημοσιευμένο

Διεύθυνση: Λεωφόρος Στρατού 15

Καταχωρήθηκε: Δευτέρα 22/03 - 12:53

Πολυμέσα:

[Επεξεργασία](#)

Τίτλος: Λακούβα σε κεντρική οδό της πόλης σε σημείο όπου διερχονται πολλά αυτοκίνητα

Περιγραφή: Η λακούβα ενδεχομένως εμφανίστηκε σήμερα το πρωί, καθώς εχθες το βράδυ που πέρασα με το αμάξι μου δεν υπήρχε κάτι. Είναι ένα θέμα που πρέπει να διευθετηθεί καθώς διέρχεται πλήθος από αυτοκίνητα και αν κάποιος τρέχει υπάρχει μεγάλος κίνδυνος ατυχήματος.

Αριθμός Ψήφων: 14

Κατηγορία: Οδικό δίκτυο

Κατάσταση: Δημοσιευμένο

Διεύθυνση: Λεωφόρος Στρατού 15

Καταχωρήθηκε: Δευτέρα 22/03 - 12:53

Πολυμέσα:

[Επεξεργασία](#)

Διαχείριση Κατηγοριών

Φίλτρα

Τίτλος- Περιγραφή:
Εισάγεται μία λέξη ή φράση σχετική με ένα ζήτημα

Κατηγορία:

<input checked="" type="checkbox"/> Οδικό δίκτυο	<input checked="" type="checkbox"/> Υδροδότηση
<input type="checkbox"/> Απορρίμματα	<input type="checkbox"/> Ηλεκτροδότηση
<input checked="" type="checkbox"/> Πάρκα	<input type="checkbox"/> Άλλα

Κατάσταση Ζητήματος:

<input checked="" type="checkbox"/> Δημοσιευμένο
<input type="checkbox"/> Επιλυμένο
<input type="checkbox"/> Εκκρεμές

Διεύθυνση:

Ημερομηνία Καταχώρησης:

από: ηη / μμ / έεεε


έως: ηη / μμ / έεεε

[Εφαρμογή φίλτρων](#) [Καθαρισμός φίλτρων](#)

Σχήμα 3.12: Αρχική οθόνη διαχειριστή - λίστα ζητημάτων διαχειριστή

Η παραπάνω οθόνη αποτελεί την σελίδα, μέσω της οποίας ο διαχειριστής μπορεί να δει τα ζητήματα που έχουν καταχωρηθεί από τους χρήστες και να εφαρμόσει φίλτρα ώστε να περιορίσει τα ζητήματα που βλέπει.

3.8 Σελίδα διαχείρισης ζητήματος – διαχειριστής



Σελίδα διαχείρισης ζητήματος

Δήμος Θεσσαλονίκης 25/03/2020

Βρέθηκαν 15 ζητήματα

Τίτλος: Λακούβα σε κεντρική οδό της πόλης σε σημείο όπου διερχονται πολλά αυτοκίνητα


Περιγραφή: Η λακούβα ενδεχομένως εμφανίστηκε σήμερα το πρωί, καθώς εχθες το βράδυ που πέρασα με το αμάξι μου δεν υπήρχε κάτι. Είναι ένα θέμα που πρέπει να διευθετηθεί καθώς διέρχεται πλήθος από αυτοκίνητα και αν κάποιος τρέχει υπάρχει μεγάλος κίνδυνος ατυχήματος.

Αριθμός Ψήφων: 14

Κατηγορία: Οδικό δίκτυο

Κατάσταση: Δημοσιευμένο

Πολυμέσα:



Διεύθυνση: Λεωφόρο Στρατού 15

Καταχωρήθηκε: Δευτέρα 22/03 - 12:53

✓ Αποθήκευση

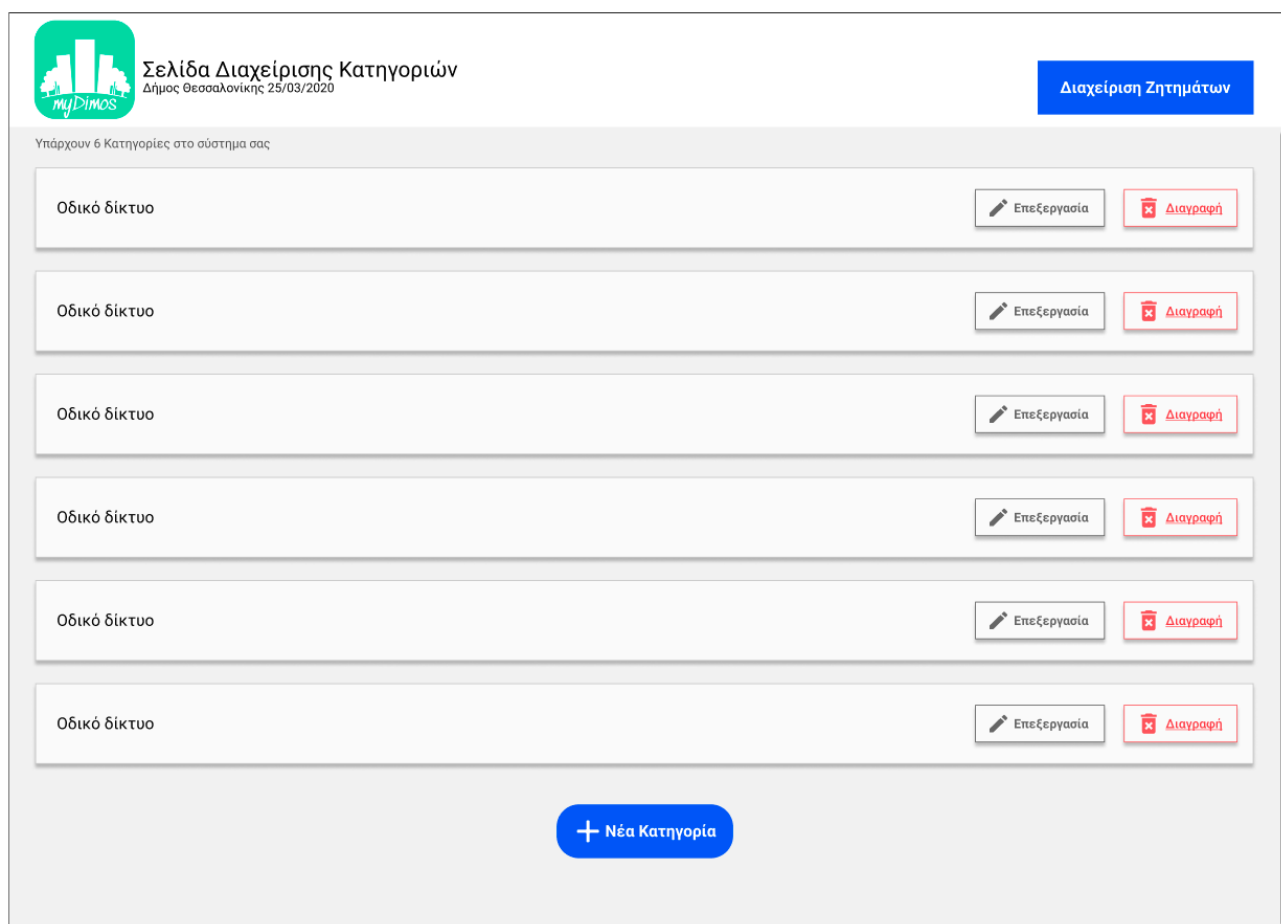
Ακύρωση

Διαγραφή ζητήματος

Σχήμα 3.13: Σελίδα διαχείρισης ζητήματος-διαχειριστής

Η παραπάνω οθόνη αποτελεί την σελίδα, μέσω της οποίας ο διαχειριστής μπορεί να επεξεργαστεί και να διαγράψει ένα ζήτημα .

3.9 Σελίδα κατηγοριών – διαχειριστής



Σελίδα Διαχείρισης Κατηγοριών
Δήμος Θεσσαλονίκης 25/03/2020

Διαχείριση Ζητημάτων

Υπάρχουν 6 Κατηγορίες στο σύστημα σας

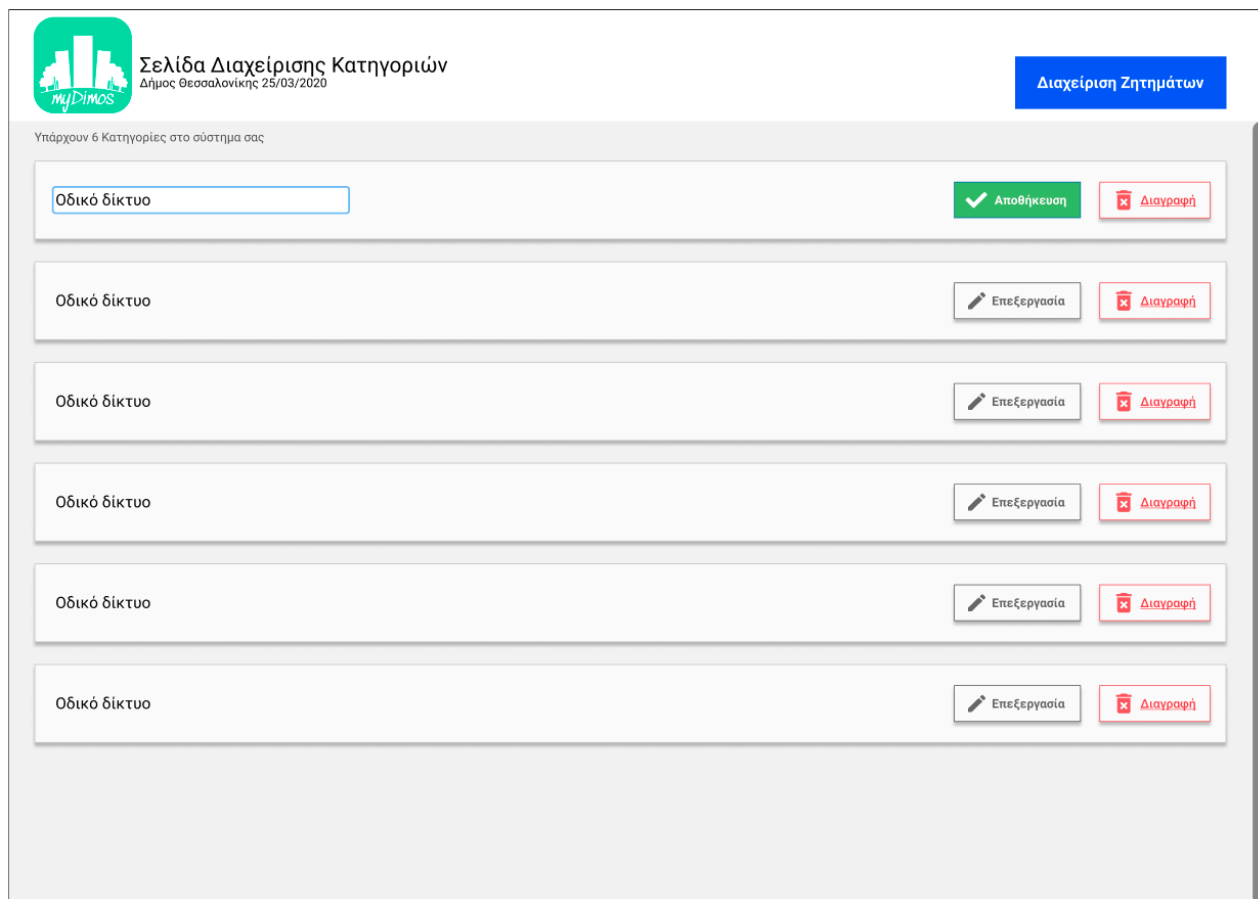
Οδικό δίκτυο	Επεξεργασία	Διαγραφή
Οδικό δίκτυο	Επεξεργασία	Διαγραφή
Οδικό δίκτυο	Επεξεργασία	Διαγραφή
Οδικό δίκτυο	Επεξεργασία	Διαγραφή
Οδικό δίκτυο	Επεξεργασία	Διαγραφή
Οδικό δίκτυο	Επεξεργασία	Διαγραφή

+ Νέα Κατηγορία

Σχήμα 3.14: Σελίδα κατηγοριών-διαχειριστής

Η παραπάνω οθόνη αποτελεί την σελίδα, μέσω της οποίας ο διαχειριστής μπορεί να δει και να διαχειριστεί τις κατηγορίες του συστήματος.

3.10 Σελίδα διαχείρισης κατηγορίας – διαχειριστής



myDimos

Σελίδα Διαχείρισης Κατηγοριών
Δήμος Θεσσαλονίκης 25/03/2020

Διαχείριση Ζητημάτων

Υπάρχουν 6 Κατηγορίες στο σύστημα σας

<input type="text" value="Οδικό δίκτυο"/>	✓ Αποθήκευση	✗ Διαγραφή
Οδικό δίκτυο	✎ Επεξεργασία	✗ Διαγραφή
Οδικό δίκτυο	✎ Επεξεργασία	✗ Διαγραφή
Οδικό δίκτυο	✎ Επεξεργασία	✗ Διαγραφή
Οδικό δίκτυο	✎ Επεξεργασία	✗ Διαγραφή
Οδικό δίκτυο	✎ Επεξεργασία	✗ Διαγραφή

Σχήμα 3.15: Σελίδα επεξεργασίας κατηγορίας -διαχειριστής

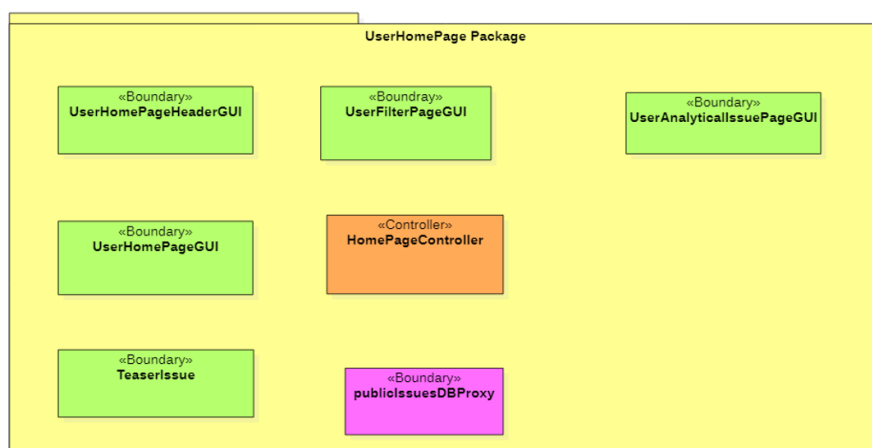
Η παραπάνω οθόνη αποτελεί την σελίδα, μέσω της οποίας ο διαχειριστής μπορεί να επεξεργαστεί (μετονομάσει) μια συγκεκριμένη κατηγορία.



4 Στατική Μοντελοποίηση - Πακέτο(α) λεξιλογίου σεναρίων υψηλής προτεραιότητας

4.1.1 Πακέτο UserHomePage

Σκοπός του πακέτου είναι η προβολή της αρχικής σελίδας καθώς επίσης της σελίδας των φίλτρων και αυτής της αναλυτικής προβολής ενός ζητήματος. Αν και το πακέτο δεν σχετίζεται άμεσα με τις λειτουργικές απαιτήσεις υψηλότερης προτεραιότητας που έχουν να κάνουν με τον χρήστη, παρουσιάζεται πρώτο, καθώς μέσω αυτού, είναι εφικτή η ανακατεύθυνση του χρήστη μεταξύ των σελίδων και η περιήγησή του στην εφαρμογή.



4.1.1.1 Ορισμοί- Πακέτο UserHomePage

Boundary UserHomePageHeaderGUI

«Boundary» UserHomePageHeaderGUI
-Logo: image -btnSubmitIssue: Button -btnPendingIssues: Button -btnMagnifyingGlass: Button -txtSearch: TextBox -btnFilter: Button -infoAndDate: string -results: string
+UserHomePageGUI(): void +UserHomePageGUI(Date: string): void +setTxtResults(val: string): void +getTextBoxInput(): string +btnMagnifyingGlassClick(): void +btnSubmitIssueClick(): void +btnPendingIssuesClick(): void +btnFilterClick(): void

Αυτή η κλάση είναι υπεύθυνη για την εμφάνιση της διεπαφής της αρχικής σελίδας του χρήστη.

Χαρακτηριστικά της κλάσης:

- Logo: Μια εικόνα με το logo της εφαρμογής
- btnSubmitIssue: Ένα Button που θα οδηγεί τον χρήστη στην σελίδα της εφαρμογής με την φόρμα καταχώρησης όταν το πατήσει.



- btnPendingIssues: Ένα Button που θα οδηγεί τον χρήστη στην σελίδα της εφαρμογής με τα εκκρεμή του ζητήματα όταν το πατήσει.
- btnMagnifyingGlass: Ένα Button που θα εκκινεί την διαδικασία φιλτραρίσματος.
- txtSearch: Ένα textBox όπου θα γράφει ο χρήστης για αναζήτηση.
- btnFilter: Ένα button που θα οδηγεί στην σελίδα με τα φίλτρα.
- infoAndDate: Ένα string που θα εμφανίζει την τρέχουσα ημερομηνία.
- results: Ένα string που θα εμφανίζει τον αριθμό των αποτελεσμάτων αναζήτησης

Μέθοδοι της κλάσης:

- UserHomePageGUI(): Constructor χωρίς ορίσματα.
- UserHomePageGUI(Date:string): Constructor με όρισμα ένα string με την ημερομηνία.
- setTxtResults(val: string): Ένας setter για την μεταβλητή results.
- getTextInput(): Η συνάρτηση επιστρέφει την τιμή του textBox με το οποίο θα έχει αλληλεπιδράσει ο χρήστης. Η τιμή είναι τύπου string.
- btnMagnifyingGlassClick(): Η συνάρτηση αυτή καλεί την calculateFilteredIssueList(categories:list<string>, status:list<string>, searchString:string, in dateFrame:list<string>, maxDistance:integer): list<Issues> και μετά την showUserHomePageGUI().
- btnSubmitIssueClick(): Η συνάρτηση αυτή καλεί τη showUserIssueCreationPageGUI() συνάρτηση στον HomeController.
- btnPendingIssuesClick(): Η συνάρτηση αυτή καλεί την showPendingIssuePageGUI() συνάρτηση στον HomeController.
- btnFilterClick(): Η συνάρτηση αυτή καλεί την showUserFilterPageGUI() συνάρτηση στον HomeController.

UserFilterPageGUI

«Boundary» UserFilterPageGUI
-btnBack: Button -lblTitle: label -lblCategory: label -categoriesCheckBoxes: list<CheckBox> -lblIssueStatus: label -statusCheckBoxes: list<CheckBox> -lblMaxDistance: label -lblDateSubmitted: label -lblFrom: label -lblUntil: label -dateComboBoxes: list<ComboBox> -btnApply: Button -btnCancel: Button -lblSlash: label
+UserFilterPageGUI(): void +UserFilterPageGUI(categoryList: list<Category>): void +getCategoriesChecked(): list<string> +getStatusChecked(): list<string> +getDateFrame(): list<string> +getMaxDistance(): integer +btnBackClick(): void +btnApplyClick(): void +btnCancelClick(): void

Αυτή η κλάση είναι υπεύθυνη για την εμφάνιση της διεπαφής των φίλτρων και κληρονομεί την UserHomePageHeaderGUI.

Χαρακτηριστικά της κλάσης:

- btnBack: Ένα Button που θα επιστρέφει τον χρήστη στην αρχική σελίδα όταν πατηθεί.



- **lblTitle:** Ένα label για τον τίτλο που θα έχει η διεπαφή.
- **lblCategory:** Ένα label για να εμφανίζεται η λέξη 'Κατηγορία' και έπειτα να ακολουθούν τα checkBox με τις επιλογές που έχει ο χρήστης.
- **categoriesCheckBoxes:** Μία λίστα από CheckBox που θα είναι υπεύθυνα για την εμφάνιση των επιλογών του χρήστη για φιλτράρισμα με βάση την κατηγορία.
- **lblIssueStatus:** Ένα label για να εμφανίζεται η λέξη 'Κατάσταση' και έπειτα τα checkBox με τις επιλογές που έχει ο χρήστης για την κατάσταση του ζητήματος.
- **statusCheckBoxes:** Μία λίστα από CheckBox που θα είναι υπεύθυνα για την εμφάνιση των επιλογών του χρήστη για φιλτράρισμα με βάση την κατάσταση του ζητήματος.
- **lblMaxDistance:** Ένα label που εμφανίζει την φράση 'Μέγιστη απόσταση:'.
- **lblFrom:** Ένα label για να εμφανίζει την λέξη 'From'.
- **lblUntill:** Ένα label για να εμφανίζει την λέξη 'Untill'.
- **dateComboBoxes:** Μία λίστα από ComboBox που θα χρησιμοποιούνται για να επιλέξει ημερομηνία από και έως ο χρήστης.
- **btnApply:** Ένα button που όταν πατηθεί θα εφαρμόζει όλα τα φίλτρα και θα επιστρέφει ο χρήστης στην αρχική βλέποντας μια νέα λίστα.
- **btnCancel:** Ένα button που θα καθαρίζει όλα τα φίλτρα.
- **lblSlash:** Ένα label που θα εμφανίζει το σύμβολο '/' και θα χρησιμοποιείται ως διαχωριστικό ανάμεσα στα ComboBox.

Μέθοδοι της κλάσης:

- **UserFilterPageGUI():** Constructor χωρίς ορίσματα.
- **UserFilterPageGUI (categoryList: list<Category>):** Constructor με όρισμα μία λίστα με αντικείμενα Category.
- **getCategoriesChecked():** Η συνάρτηση αυτή επιστρέφει μία λίστα με string από τις κατηγορίες που έχουν επιλεγεί.
- **getStatusChecked():** Η συνάρτηση αυτή επιστρέφει μία λίστα από string από τις καταστάσεις ζητήματος που έχουν επιλεγεί.
- **getDateFrame():** Η συνάρτηση αυτή επιστρέφει μία λίστα string με την χρονική περίοδο από-εώς που έχει επιλέξει ο χρήστης.
- **getMaxDistance():** Η συνάρτηση αυτή επιστρέφει έναν integer για την μέγιστη απόσταση που θέλει ο χρήστης να απέχει ένα ζήτημα από αυτόν για να εμφανιστεί.
- **btnBackClick():** Η συνάρτηση αυτή καλεί την `showUserHomePageGUI()` συνάρτηση του `HomePageController`.
- **btnApplyClick():** Η συνάρτηση αυτή καλεί την `calculateFilteredIssueList(categories:list<string>, status:list<string>, searchString:string, in dateFrame:list<string>, maxDistance:integer): list<Issues>` και μετά την `showUserHomePageGUI()`.
- **btnCancelClick():** Διαγράφει όλες τις επιλογές του χρήστη στα φίλτρα.

HomePageTeaserIssue



«Boundary» TesearIssue
-title: string -imgGpsPin: icon -address: string -btnLike: Button -votes: string -distance: string -imgClock: icon -dateSumbited: string -status: string
+HomePageTesearIssue(): void +HomePageTesearIssue(issueToTesea: Issue): void +click(uniqueId: integer): void +calcUserDistanceFromIssue(address: string): integer +btnLikeClick(issueID: integer): void

Αυτή η κλάση είναι υπεύθυνη για την εμφάνιση της διεπαφής HomePageTeaserIssue και προβάλλει συνοπτικά το περιεχόμενο της κλάσης Issue.

Χαρακτηριστικά της κλάσης:

- title: Ένα string που θα εμφανίζει το τίτλο από το ζήτημα που προβάλλει συνοπτικά.
- imgGpsPin: Μία μεταβλητή για την εμφάνιση μιας ring εικόνας.
- address: Ένας string υπεύθυνος για την εμφάνιση της διεύθυνσης του ζητήματος που προβάλλει συνοπτικά.
- btntLike: Ένα Button που θα μπορεί να πατάει ο χρήστης για να υπερψηφίσει ένα ζήτημα.
- votes: Ένα string που θα εμφανίζει τις ψήφους που έχει ένα ζήτημα.
- distance: Ένα string που θα εμφανίζει την απόσταση που θα απέχει ο χρήστης από ένα ζήτημα.
- imgClock: Μία μεταβλητή για την εμφάνιση της εικόνας ενός ρολογιού.
- dateSumbited: Ένα string για να εμφανίζει την ημερομηνία που καταχωρήθηκε το ζήτημα.
- status: Ένα string για να εμφανίζει την κατάσταση του ζητήματος.

Μέθοδοι της κλάσης:

- HomePageTesearIssue(): Constructor χωρίς ορίσματα.
- HomePageTesearIssue (issueToTesea: Issue): Constructor με όρισμα αντικείμενο Issue.
- click(uniqueId: integer): Αυτή η συνάρτηση καλεί τη showAnalyticalIssuePageGUI (issueId: integer) από τον HomeController.
- calcUserDistanceFromIssue(address: string): Η συνάρτηση αυτή υπολογίζει την απόσταση του χρήστη από την διεύθυνση του συγκεκριμένου ζητήματος. Επικοινωνεί με το UserDeviceHandle Package προκειμένου να πάρει την θέση του χρήστη και να υπολογισθεί η απόσταση.
- btnLikeClick(issueId: integer): Καλείται όταν πατάει το btnLike ο χρήστης και καλεί την συνάρτηση updateIssueVotes(issueId: integer) του HomeController.

UserHomePageGUI

«Boundary» UserHomePageGUI
+listTeaserIssue: list<TeaserIssue>
+UserHomePageGUI(issues: list<Issue>): void
+onSlide()

Αυτή η κλάση είναι υπεύθυνη για την εμφάνιση της διεπαφής UserHomePageGUI και είναι η αρχική σελίδα της εφαρμογής. Κληρονομεί την UserHomePageHeaderGUI και αποτελεί συνάθροιση της TesearIssue.

**Χαρακτηριστικά της κλάσης:**

- listTesearIssue: Μία λίστα με αντικείμενα τύπου TesearIssue για να προβάλλονται σε μία λίστα.

Μέθοδοι της κλάσης:

- UserHomePageGUI(issues: Issues): Constructor με όρισμα μία λίστα με Issue αντικείμενα.
- onSlide(): Η συνάρτηση αυτή καλείται όταν κάνει scroll ο χρήστης για να εμφανίζει τα ανάλογα HomePageTesearIssue.

UserAnalyticalIssuePageGUI

«Boundary» UserAnalyticalIssuePageGUI
-Logo: image -date: string -btnBack: Button -title: string -ping: icon -address: string -distance: string -clock: icon -lblSumbited: label -dateSumbited: string -category: string -status: string -description: string -media: audioVisual -lblVoteIssue: label -btnLike: Button -numberOfVotes: string +UserAnalyticalIssuePageGUI(issue: Issue): void +btnBackClick(): void +btnLikeClick(issueID: integer): void

Αυτή η κλάση είναι υπεύθυνη για την εμφάνιση της διεπαφής UserAnalyticalIssuePageGUI και προβάλλει αναλυτικά το περιεχόμενο της κλάσης Issue.

Χαρακτηριστικά της κλάσης:

- Logo: Μια εικόνα με το logo της εφαρμογής.
- date: Ένα string που εμφανίζει την τρέχουσα ημερομηνία.
- btnBack: Ένα Button για να επιστρέφει ο χρήστης στην αρχική σελίδα.
- title: Ένα string που θα εμφανίζει το τίτλο του ζητήματος.
- ping: Μία μεταβλητή για την εμφάνιση μιας ping εικόνας.
- address: Μία μεταβλητή για την εμφάνιση της διεύθυνσης.
- distance: Μία μεταβλητή για την εμφάνιση της απόστασης του ζητήματος από την θέση του χρήστη.
- clock: Μία μεταβλητή για την εμφάνιση της εικόνας ενός ρολογιού.
- lblSumbitted: Ένα label για την εμφάνιση της φράσης 'Καταχωρήθηκε:'.
- dateSumbited: Ένα string για να εμφανίζει την ημερομηνία που καταχωρήθηκε το ζήτημα.
- category: Ένα string που θα εμφανίζει την κατηγορία του ζητήματος.
- status: Ένα string που θα εμφανίζει την κατάσταση του ζητήματος.
- description: Ένα string που θα εμφανίζει την περιγραφή του ζητήματος.
- media: Μία μεταβλητή υπεύθυνη για το οπτικοακουστικό υλικό του ζητήματος.
- lblVoteIssue: Ένα label που θα εμφανίζει την φράση 'Ψηφίστε το παραπάνω ζήτημα ως σημαντικό'.
- btnLike: Ένα Button για να υπερψηφίζει ο χρήστης το ζήτημα.
- numberOfVotes: Ένα string για να εμφανίζει τον αριθμό των ψήφων ενός ζητήματος.

Μέθοδοι της κλάσης:

- UserAnalyticalIssuePageGUI (issue: Issue): Constructor με όρισμα ένα αντικείμενο Issue.



- `btnBackClick(uniqueId: integer)`: Αυτή η συνάρτηση καλεί τη `showUserHomePageGUI()` από τον `HomePageController`.
- `btnLikeClick(issueID: integer)`: Αυτή η συνάρτηση καλεί τη `updateIssueVotes(issueID: integer)` από τον `HomePageController`.

HomePageController

«Controller» HomePageController
<code>+userID: integer</code> <code>+HomePageController(): void</code> <code>+showUserHomePageGUI(): void</code> <code>+showAnalyticalIssuePageGUI(issueID: integer): void</code> <code>+showUserFilterPageGUI(): void</code> <code>+showUserIssueCreationPageGUI(): void</code> <code>+showPendingIssuePageGUI(userID: integer): void</code> <code>+calculateFilteredIssueList(categories: list<string>, status: list<string>, searchString: string, dateFrame: list<string>, maxDistance: integer): list<Issues></code> <code>+updateIssueVotes(issueID: integer): void</code>

Αυτή η κλάση είναι υπεύθυνη για τον χειρισμό όλων των κλάσεων του πακέτου.

Χαρακτηριστικά της κλάσης:

- `userID`: Ένας μοναδικός `integer` για να αναγνωρίζει το σύστημα ποιος χρήστης το χρησιμοποιεί.

Μέθοδοι της κλάσης:

- `HomePageController()`: Constructor χωρίς ορίσματα
- `showUserHomePageGUI()`: Αυτή η συνάρτηση εμφανίζει την διεπαφή που εκφράζει η κλάση `UserHomePageGUI`.
- `showAnalyticalIssuePageGUI(issueID: integer)`: Αυτή η συνάρτηση εμφανίζει την διεπαφή που εκφράζει η κλάση `AnalyticalIssuePageGUI` με το όρισμα να δείχνει πιο ζήτημα πρέπει να προβληθεί αναλυτικά.
- `showUserFilterPageGUI()`: Αυτή η συνάρτηση εμφανίζει την διεπαφή που εκφράζει η κλάση `UserFilterPageGUI`.
- `showUserIssueCreationPageGUI()`: Αυτή η συνάρτηση εμφανίζει την διεπαφή που εκφράζει η κλάση `SubmissionFormGUI` του πακέτου `UserIssueHandle Package`.
- `showPendingIssuePageGUI(userID: integer)`: Αυτή η συνάρτηση εμφανίζει την διεπαφή που εκφράζει η κλάση `PendingIssueGUI` του πακέτου `UserIssueHandle Package`.
- `calculateFilteredIssueList(categories: list<string>, status: list<string>, searchString: string, in dateFrame: list<string>, maxDistance: integer)`: Η συνάρτηση αυτή καλεί την `LoadIssueList(categories: list<string>, status: list<string>, searchString: string, in dateFrame: list<string>, maxDistance: integer)` της κλάσης `publicIssuesDBProxy` και επιστρέφει μια λίστα από `Issues`.
- `updateIssueVotes(issueID: integer)`: Η συνάρτηση αυτή καλεί την `writeNewVotes(issueID: integer, val: string)` της κλάσης `publicIssuesDBProxy`.

publicIssuesDBProxy

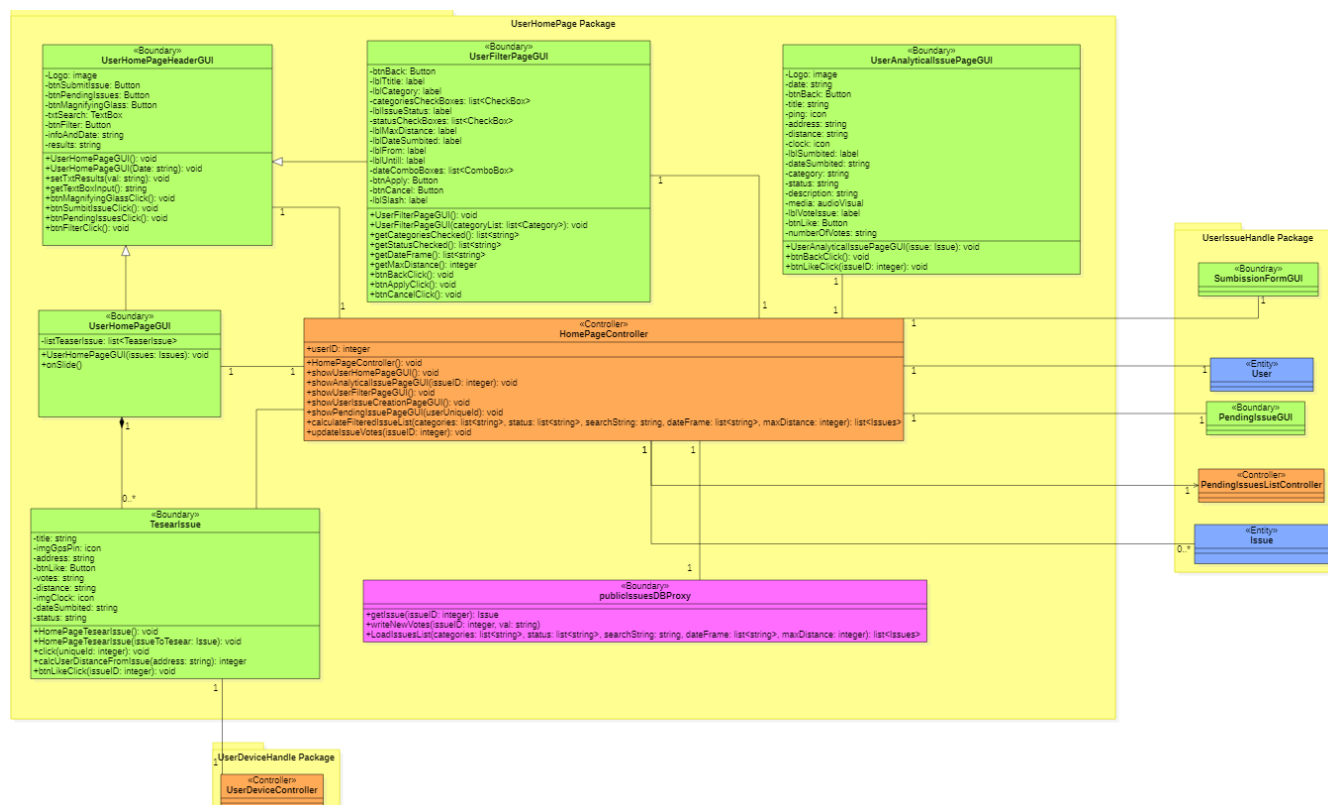
«Boundary» publicIssuesDBProxy
<code>+getIssue(issueID: integer): Issue</code> <code>+writeNewVotes(issueID: integer, val: string)</code> <code>+LoadIssueList(categories: list<string>, status: list<string>, searchString: string, dateFrame: list<string>, maxDistance: integer): list<Issues></code>

Αυτή η κλάση είναι υπεύθυνη για τον χειρισμό όλων των κλάσεων του πακέτου.

Μέθοδοι της κλάσης:

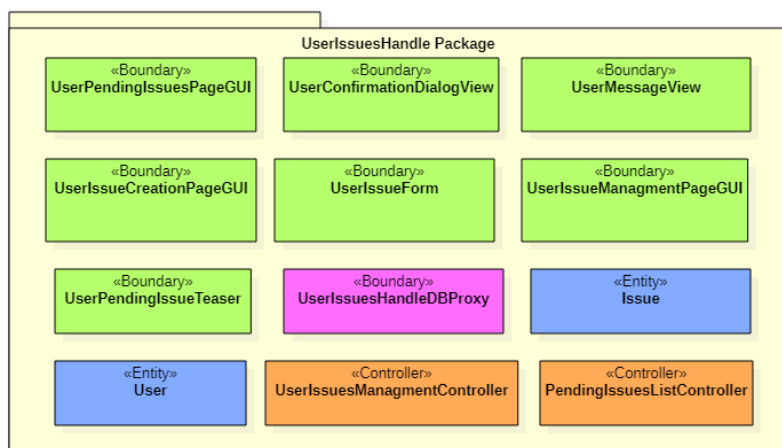
- `getIssue(issueId: integer)`: Η συνάρτηση αυτή επικοινωνεί με την βάση δεδομένων και επιστρέφει ένα αντικείμενο τύπου `Issue` με βάση το `issueId` που δόθηκε ως όρισμα.
- `writeNewVotes(issueId: integer, val: string)`: Η συνάρτηση γράφει στην βάση δεδομένων μια νέα τιμή για της ψήφους του ζητήματος με `issueId`.
- `loadIssueList(categories:list<string>,status:list<string>,searchString:string,dateFrame:list<string>,maxDistance:integer)`: Η συνάρτηση λαμβάνει όλα τα όρια υπόψιν και επιστρέφει μια λίστα με `Issues` που τα ικανοποιούν.

4.1.1.2 Διαγράμματα κλάσεων - Πακέτο UserHomePage



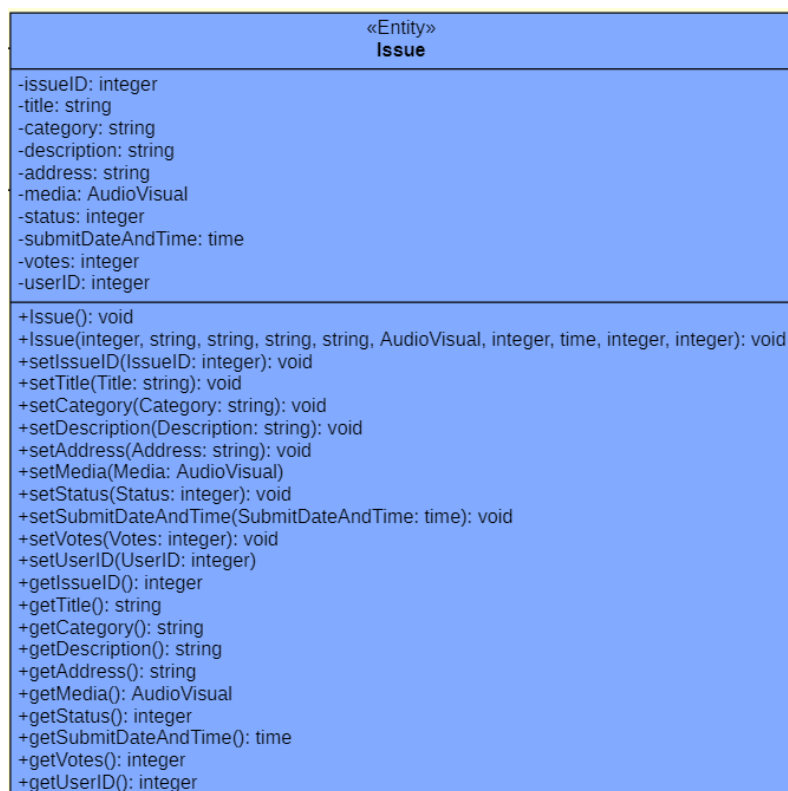
4.1.2 Πακέτο UserIssuesHandle

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που σχετίζονται με τη διαχείριση των ζητημάτων από ένα χρήστη. Συγκεκριμένα περιλαμβάνει την δημιουργία ενός νέου ζητήματος καθώς και την προβολή, επεξεργασία και διαγραφή των εκκρεμών ζητημάτων του χρήστη. Επίσης λόγω σημαντικότητας των λειτουργιών του συγκεκριμένου πακέτου για τον χρήστη, σε αυτό ορίζονται οι κλάσεις οντοτήτων του χρήστη (απλός χρήστης) και του ζητήματος.



4.1.2.1 Ορισμοί - Πακέτο UserIssuesHandle

Entity Issue



Η κλάση αυτή εμπεριέχει όλες τις ιδιότητες και τις μεθόδους που απαιτούνται για να οριστεί πλήρως η οντότητα των ζητημάτων.

Ιδιότητες της Κλάσης:

- **issueID:** Ο μοναδικός χαρακτηριστικός αριθμός ενός ζητήματος.
- **title:** Ο τίτλος του ζητήματος
- **category:** Η κατηγορία στην οποία ανήκει το ζήτημα.
- **description:** Η περιγραφή του ζητήματος.
- **address:** Η διεύθυνση όπου λαμβάνει χώρα το πρόβλημα που παρουσιάζεται στο ζήτημα.
- **media:** Οπτικοακουστικό υλικό σχετικό με το ζήτημα.
- **status:** Η κατάσταση του ζητήματος (επιλυμένο ή όχι).



- `submitDateAndTime`: Η ημερομηνία και ώρα καταχώρησης του ζητήματος.
- `Votes`: Οι ψήφοι που έχει λάβει συνολικά το ζήτημα.
- `userID`: Το `userID` (μοναδικός χαρακτηριστικός αριθμός του χρήστη), που έχει καταχωρήσει το ζήτημα.

Μέθοδοι της Κλάσης:

- `Issue()`, `Issue(integer, string, string, string, string, AudioVisual, integer, time, integer, integer)`: Οι μέθοδοι δόμησης μιας οντότητας της κλάσης.
- `setIssueID(IssueID: integer)`: Θέτει την τιμή του μοναδικού χαρακτηριστικού αριθμού ενός ζητήματος.
- `setTitle(Title: string)`: Θέτει τον τίτλο του ζητήματος.
- `setCategory(Category: string)`: Θέτει την κατηγορία του ζητήματος.
- `setDescription(Description: string)`: Θέτει την περιγραφή του ζητήματος.
- `setAddress(Address: string)`: Θέτει την διεύθυνση του ζητήματος.
- `setMedia(Media: AudioVisual)`: Θέτει το οπτικοακουστικό υλικό ενός ζητήματος.
- `setStatus(Status: integer)`: Θέτει την κατάσταση ενός ζητήματος (εκκρεμές, δημοσιευμένο, επιλυμένο).
- `setSubmitDateAndTime(SubmitDateAndTime: time)`: Θέτει την ημερομηνία και την ώρα καταχώρησης του ζητήματος.
- `setVotes(Votes: integer)`: Θέτει την τιμή του αριθμού των ψήφων του ζητήματος.
- `setUserID(UserID: integer)`: Θέτει τον μοναδικό χαρακτηριστικό αριθμό (ID) του χρήστη που καταχώρησε το ζήτημα.
- `getIssueID()`: Επιστρέφει τον μοναδικό χαρακτηριστικό αριθμό του ζητήματος.
- `getTitle()`: Επιστρέφει τον τίτλο του ζητήματος.
- `getCategory()`: Επιστρέφει την κατηγορία του ζητήματος.
- `getDescription()`: Επιστρέφει την περιγραφή του ζητήματος.
- `getAddress()`: Επιστρέφει την διεύθυνση στην οποία λαμβάνει χώρα το ζήτημα.
- `getMedia()`: Επιστρέφει το οπτικοακουστικό υλικό του ζητήματος.
- `getStatus()`: Επιστρέφει την κατάσταση του ζητήματος.
- `getSubmitDateAndTime()`: Επιστρέφει την ημερομηνία και την ώρα καταχώρησης του ζητήματος.
- `getVotes()`: Επιστρέφει τον αριθμό των ψήφων του ζητήματος.
- `getUserID()`: Επιστρέφει τον μοναδικό χαρακτηριστικό αριθμό (ID) του χρήστη που καταχώρησε το ζήτημα.

Entity User

«Entity» User
-userID: integer -issuesIDs: list<integer> -username: string -password: string
+User(): void +User(integer, issuesIDs: list<integer>, string, string): void +setUserID(UserID: integer): void +setIssuesIDs(IssuesIDs: list<integer>): void +setUsername(Username: string): void +setPassword>Password: string): void +getUserID(): integer +getIssuesIDs(): list<integer> +getUsername(): string +getPassword(): string

Η κλάση αυτή εμπεριέχει όλες τις ιδιότητες και τις μεθόδους που απαιτούνται για να οριστεί πλήρως η οντότητα των χρηστών (απλών χρηστών).

*Ιδιότητες της Κλάσης:*

- userID: Ο μοναδικός χαρακτηριστικός αριθμός ενός χρήστη.
- issuesIDs: Οι μοναδικοί χαρακτηριστικοί αριθμοί (ID) των ζητημάτων που έχει καταχωρήσει ο χρήστης.
- username: Το username του χρήστη.
- password: Ο κωδικός του χρήστη.

Μέθοδοι της Κλάσης:

- User() , User(integer, issuesIDs: list<integer>, string, string) : Οι μέθοδοι δόμησης μιας οντότητας της κλάσης .
- setUserIDs(UserID: integer): Θέτει τον μοναδικό χαρακτηριστικό αριθμό (ID) του χρήστη.
- setIssuesIDs(IssuesIDs: list<integer>): Θέτει τα ID των ζητημάτων που έχει καταχωρήσει ο χρήστης
- setUsername(Username:string) : Θέτει το username του χρήστη.
- setPassword(Password:string) : Θέτει τον κωδικό του χρήστη.
- getUserID(): Επιστρέφει το ID του χρήστη.
- getIssuesIDs() : Επιστρέφει τα IDs των ζητημάτων που έχει καταχωρήσει ο χρήστης
- getUsername: Επιστρέφει το username του χρήστη.
- getPassword() : Επιστρέφει τον κωδικό του χρήστη.

Boundary UserIssueForm

«Boundary» UserIssueForm
-Logo: Image -infoAndDate: string -btnBack: Button -PageTitle: string -lblIssueTitle: label -txtIssueTitle: TextBox -lblCategory: label -cmbCategory: ComboBox -lblDescription: label -txtDescription: TextBox -lblAddress: label -txtAddress: TextBox -lblMessGps: label -btnGpsPinImport: Button -lblMedia: label -lblMessMedia: label -btnSelectFile: Button -isFileUploaded: Boolean -fileName: link -btnDeleteFile: Button -lblRequiredFields: label
+UserIssueForm(): void +btnBackClick(): void +btnGpsPinImportClick(): void +autoCompleteAddress(Address: string): void +btnSelectFileClick() +btnDeleteFileClick(fileName: link): void +fileNameClick()

Η κλάση αυτή αποτελεί την φόρμα που βλέπει ο χρήστης για την καταχώρηση ή επεξεργασία ενός ζητήματος. Εμπεριέχει όλες τις ιδιότητες και τις μεθόδους που απαιτούνται για να ορίσει ο χρήστης ένα ζήτημα. Κληρονομείται από τις κλάσεις **UserIssueCreationPageGUI** και **UserIssueManagementPageGUI** , οι οποίες αποτελούν τις σελίδες δημιουργίας νέου ζητήματος και επεξεργασίας ενός καταχωρημένου εκκρεμούς ζητήματος αντιστοίχως.

**Ιδιότητες της Κλάσης:**

- **Logo:** Το λογότυπο της εφαρμογής MyDimos.
- **infoAndDate:** Κείμενο που απεικονίζει το όνομα του δήμου καθώς και την σημερινή ημερομηνία.
- **btnBack:** Κουμπί για επιστροφή σε προηγούμενη οθόνη.
- **lblPageTitle:** Ο τίτλος της σελίδας.
- **lblIssueTitle:** Ετικέτα που εμφανίζει την λέξη «Τίτλος».
- **txtIssueTitle:** Πλαίσιο για την συμπλήρωση του τίτλου του ζητήματος.
- **lblCategory:** Ετικέτα που εμφανίζει την λέξη «Κατηγορία»
- **cmbCategory:** Ένας επιλογέας για τη επιλογή της κατηγορίας του ζητήματος.
- **lblDescription:** Ετικέτα που εμφανίζει την λέξη «Περιγραφή».
- **txtDescription:** Πλαίσιο για την συμπλήρωση της περιγραφής του ζητήματος.
- **lblAddress:** Ετικέτα που εμφανίζει την λέξη «Διεύθυνση»
- **txtAddress:** Πλαίσιο για την συμπλήρωση της διεύθυνσης του ζητήματος.
- **lblMessGps:** Μήνυμα για την εισαγωγή του στίγματος GPS.
- **btnGpsPinImport:** Κουμπί για την εισαγωγή του στίγματος GPS του χρήστη.
- **lblMedia:** Ετικέτα που εμφανίζει την λέξη «Πολυμέσα».
- **lblMessMedia:** Μήνυμα σχετικό με την επιλογή οπτικοακουστικού υλικού.
- **btnSelectFile:** Κουμπί για την επιλογή αρχείου από τον χρήστη.
- **isFileUploaded:** Boolean μεταβλητή που υποδεικνύει αν έχει ανέβει οπτικοακουστικό υλικό σχετικό με το ζήτημα ή όχι.
- **filename:** Link που δείχνει στο αρχείο το οποίο έχει επιλέξει ο χρήστης ως οπτικοακουστικό υλικό. Πρόκειται για το αρχείο που είναι αποθηκευμένο στην συσκευή του χρήστη σε περίπτωση που ο χρήστης δημιουργεί νέο ζήτημα(δεν έχει αποθηκευτεί) . Ή για το αποθηκευμένο στην βάση αρχείο όταν , ο χρήστης επεξεργάζεται ένα προηγουμένως καταχωρημένο ζήτημα.
- **btnDeleteFile:** Κουμπί για την διαγραφή του αρχείου.
- **btnDeleteFileVisibility:** Δυαδική τιμή σχετική με την ορατότητα(εμφάνιση) του κουμπιού διαγραφής αρχείου ή όχι. Έχει τιμή 1 όταν υπάρχει αρχείο και τιμή 0 διαφορετικά.
- **lblRequiredFields:** Ετικέτα που εμφανίζει το μήνυμα «* Υποχρεωτικά πεδία»

Μέθοδοι της Κλάσης:

- **UserIssueForm():** Μέθοδος δόμησης μιας οντότητας της κλάσης .
- **btnBackClick():** Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί για την επιστροφή σε προηγούμενη οθόνη, μεταφέρει τον έλεγχο στην συνάρτηση *showPreviousPage()* του controller *UserIssuesManagementController* .
- **btnGpsPinImportClick():** Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί για την εισαγωγή του στίγματος GPS του χρήστη. Μεταφέρει τον έλεγχο στην συνάρτηση *getGpsPinFromDevice()* του controller *UserDeviceController* που ορίζεται στο πακέτο *UserDeviceHandle*.
- **autoCompleteAddress():** Συμπληρώνει στο πλαίσιο της διεύθυνσης, όταν ο χρήστης εισάγει την θέση του μέσω του Gps Pin του, την διεύθυνση που επιστρέφεται από την συνάρτηση *printAddressToIssueForm()* του controller *UserDeviceController* που ορίζεται στο πακέτο *UserDeviceHandle*.
- **btnSelectFileClick():** Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί για την επιλογή αρχείου, μεταφέρει τον έλεγχο στον *UserDeviceController* και συγκεκριμένα στις συναρτήσεις *showDeviceGallery()* και *selectFile()* .
- **getBtnDeleteFileVisibility():** επιστρέφει την δυαδική τιμή της μεταβλητής *btnDeleteFileVisibility*.
- **setBtnDeleteFileVisibility(Visibility:Boolean) :** θέτει την δυαδική τιμή της μεταβλητής *btnDeleteFileVisibility*.



- `btnDeleteFileClick(fileName:link)` : Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί διαγραφής του αρχείου και έχει ως αποτέλεσμα την διαγραφή της σύνδεσης με το αρχείο .
- `fileNameClick()`: Συνάρτηση που εκτελείται μόλις πατήσουμε πάνω στο όνομα του αρχείου (link) και έχει ως αποτέλεσμα να προβάλλεται το αρχείο στο οποίο δείχνει ο σύνδεσμος. Μεταφέρει τον έλεγχο στον Controller *UserIssuesManagmentController* ή στον *UserDeviceController* , ανάλογα με το αν το αρχείο είναι αποθηκευμένο ή όχι.

Boundary UserIssueManagmentPageGUI

«Boundary» UserIssueManagmentPageGUI
-btnSave: Button -btnDelete: Button
+UserIssueManagmentPageGUI(Button, Button): void +btnSaveClick(): void +btnDeleteClick(): void

Η κλάση αυτή αποτελεί την σελίδα επεξεργασίας ενός εκκρεμούς ζητήματος του χρήστη. Κληρονομεί την κλάση *UserIssueForm* που παρουσιάστηκε παραπάνω.

Ιδιότητες της Κλάσης:

- `btnSave`: Κουμπί για την αποθήκευση των αλλαγών στο εκκρεμές ζήτημα.
- `btnDelete`: Κουμπί για την διαγραφή του εκκρεμούς ζητήματος .

Μέθοδοι της Κλάσης:

- `UserIssueManagmentPageGUI(Button, Button)` : Μέθοδος δόμησης μιας οντότητας της κλάσης .
- `btnSaveClick()`: Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί *btnSave*. Μεταφέρει τον έλεγχο στον controller *UserIssuesManagmentController* όπου μέσω της *checkFormData()*, ελέγχεται το περιεχόμενο της φόρμας και στην συνέχεια ο controller «δρα» ανάλογα.
- `btnDeleteClick()`: Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί *btnDelete*. Μεταφέρει τον έλεγχο στην συνάρτηση *displayConfirmationDialog()* του controller *UserIssuesManagmentController* .

Boundary UserIssueCreationPageGUI

«Boundary» UserIssueCreationPageGUI
-btnSubmit: Button
+UserIssueCreationPageGUI(Button): void +btnSubmitClick(): void

Η κλάση αυτή αποτελεί την σελίδα καταχώρησης ενός νέου ζητήματος. Κληρονομεί την κλάση *UserIssueForm* που παρουσιάστηκε παραπάνω.

Ιδιότητες της Κλάσης:

- `btnSubmit`: Κουμπί για την υποβολή του νέου ζητήματος στο σύστημα.

Μέθοδοι της Κλάσης:

- `UserIssueCreationPageGUI (Button)` : Μέθοδος δόμησης μιας οντότητας της κλάσης .
- `btnSubmitClick()`: Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί *btnSubmit*. Μεταφέρει τον έλεγχο στον controller *UserIssuesManagmentController* όπου μέσω της *checkFormData()*, ελέγχεται το περιεχόμενο της φόρμας και στην συνέχεια ο controller «δρα» ανάλογα.

Controller UserIssuesManagmentController



«Controller» UserIssuesManagmentController
-issueID: integer -userID: integer +UserIssuesManagmentController(): void +createIssue(title: string, category: string, description: string, address: string, media: audioVisual, status: integer, issueID: integer, userID: integer): Issue +storeIssue(issue: Issue): void +loadIssue(issueID: integer): Issue +updateIssue(title: string, category: string, description: string, address: string, media: audioVisual, status: integer, issueID: integer, userID: integer): void +deleteIssue(issueID: integer): void +showHomePage() +showPreviousPage(): void +showUserPendingIssuesPage(userID: integer): void +showUserIssueManagmentPage(): void +checkFormData(): boolean +displayMessage(FormDataFlag: boolean): UserMessageView +displayConfirmationDialog(FormDataFlag: boolean): UserConfirmationDialogView

Η κλάση αυτή αποτελεί τον βασικό ελεγκτή(controller) του πακέτου userIssueHandle. Είναι υπεύθυνος για την περιήγηση από τις οθόνες που σχετίζονται με το πακέτο, για την επικοινωνία με την κλάση *UserIssuesHandleDBProxy* και σχετίζεται με τον controller *PendingIssuesListController*.

Ιδιότητες της Κλάσης:

- issueID: Ο μοναδικός χαρακτηριστικός αριθμός (ID) ενός.
- userID: Ο μοναδικός χαρακτηριστικός αριθμός (ID) του χρήστη, που είναι συνδεδεμένος εκείνη την στιγμή στο σύστημα.

Μέθοδοι της Κλάσης:

- UserIssuesManagmentController(integer, integer): Μέθοδος δόμησης μιας οντότητας της κλάσης.
- checkFormData(): Η μέθοδος αυτή καλείται από τις κλάσεις *UserIssueManagmentPageGUI*, *UserIssueCreationPageGUI*, κατά το πάτημα των κουμπιών αποθήκευσης μεταβολών και υποβολής στην φόρμα ενός ζητήματος αντίστοιχα και στην συνέχεια ελέγχει την εγκυρότητα των στοιχείων της φόρμας.
- createIssue(...): Συνάρτηση που εκτελείται αφού επαληθευτεί ο έλεγχος εγκυρότητας στην φόρμα της σελίδας *UserIssueCreationPageGUI* από την συνάρτηση *checkFormData()* και αποσκοπεί στην δημιουργία ενός νέου ζητήματος με τα στοιχεία της φόρμας στο σύστημα. Επίσης καλεί την συνάρτηση *storeNewIssue(issue: Issue)* της κλάσης *UserIssuesHandleDBProxy*, για την αποθήκευση του νέου ζητήματος στη βάση.
- loadIssue(issueID: integer): Συνάρτηση που φορτώνει τα στοιχεία ενός καταχωρημένου εκκρεμούς ζητήματος στην σελίδα επεξεργασίας ζητήματος *UserIssueManagmentPageGUI*. Καλεί την ομώνυμη συνάρτηση της κλάσης *UserIssuesHandleDBProxy*, για να λάβει τα στοιχεία του ζητήματος από την βάση.
- updateIssue(...): Συνάρτηση που εκτελείται αφού επαληθευτεί ο έλεγχος εγκυρότητας στην φόρμα της σελίδας *UserIssueManagmentPageGUI* από την συνάρτηση *checkFormData()* και αποσκοπεί στην ενημέρωση ενός καταχωρημένου εκκρεμούς ζητήματος, με βάση τα στοιχεία της φόρμας. Επίσης καλεί την ομώνυμη συνάρτηση της κλάσης *UserIssuesHandleDBProxy*, για την ενημέρωση των στοιχείων του ζητήματος στη βάση.
- deleteIssue(issueID: integer): Συνάρτηση για την διαγραφή ενός στοιχείου από το σύστημα. Κατά την εκτέλεση της καλεί την ομώνυμη συνάρτηση της κλάσης *UserIssuesHandleDBProxy*, για την διαγραφή του ζητήματος από την βάση.
- showHomePage(): Εμφανίζει την αρχική σελίδα του χρήστη.
- showPreviousPage(): Εμφανίζει την σελίδα στην οποία βρισκόταν πριν ο χρήστης.
- showUserPendingIssuesPage(): Εμφανίζει την σελίδα με τα εκκρεμή ζητήματα του χρήστη, ενώ ταυτόχρονα μεταφέρει τον έλεγχο στην κλάση *PendingIssuesListController*, που είναι υπεύθυνη για την σελίδα αυτή.
- showUserIssueManagmentPage(): Εμφανίζει την σελίδα επεξεργασίας ενός εκκρεμούς ζητήματος από τον χρήστη.



- `displayMessage(FormDataFlag:boolean)`: Εμφανίζει ένα μήνυμα της κλάσης `UserMessageView` (μήνυμα επιτυχίας ή αποτυχίας μιας ενέργειας του χρήστη).
- `displayConfirmationDialog()`: Εμφανίζει ένα παράθυρο της κλάσης `UserConfirmationDialogView` (παράθυρο επιβεβαίωσης για μια ενέργεια του χρήστη).

Boundary UserIssuesHandleDBProxy

«Boundary» UserIssuesHandleDBProxy
<pre>+UserIssuesHandleDBProxy(): void +returnIssuesFromUserID(userID: integer): list<Issue> +storeNewIssue(issue: Issue): void +loadIssue(issueID: integer): Issue +updateIssue(title: string, category: string, description: string, address: string, media: audioVisual, status: integer, issueID: integer, userID: integer): boolean +deleteIssue(issueID: integer): void</pre>

Η κλάση αυτή χρησιμοποιείται για την επικοινωνία με την βάση δεδομένων. Μέσω αυτής πραγματοποιούνται όλες οι δράσεις της βάσης που σχετίζονται με τις λειτουργίες του πακέτου `UserIssuesHandle`.

Μέθοδοι της Κλάσης:

- `UserIssuesHandleDBProxy()`: Μέθοδος δόμησης μιας οντότητας της κλάσης .
- `returnIssuesFromUserID(userID:integer)`: Συνάρτηση που επιστρέφει τα εκκρεμή ζητήματα από την βάση δεδομένων που έχουν καταχωρηθεί από τον χρήστη λαμβάνοντας ως όρισμα το ID του.
- `storeNewIssue(issue:Issue)`: Συνάρτηση για την καταχώρηση ενός νέου ζητήματος στην βάση δεδομένων.
- `loadIssue(issueID: integer)`: Συνάρτηση που φορτώνει ένα ζήτημα από την βάση δεδομένων.
- `updateIssue(...)`: Συνάρτηση για την ενημέρωση των στοιχείων ενός ζητήματος που είναι καταχωρημένο στην βάση δεδομένων, με βάση τα στοιχεία που δίνονται ως όρισμα.
- `deleteIssue(issueID: integer)`: Συνάρτηση για την διαγραφή ενός ζητήματος από την βάση δεδομένων.

Boundary UserPendingIssueTeaser

«Boundary» UserPendingIssueTeaser
<pre>-issueTitle: string -imgGpsPin: icon -issueAddress: string -imgClock: icon -submitDate: string -btnEdit: Button</pre>
<pre>+UserPendingIssueTeaser(string, icon, string, icon, string, Button): void +btnEditClock(): void</pre>

Η κλάση αυτή αποτελεί την συνοπτική παρουσίαση ενός ζητήματος στην σελίδα όπου εμφανίζονται τα εκκρεμή ζητήματα του χρήστη.

Ιδιότητες της Κλάσης:

- `issueTitle`: Ο τίτλος του ζητήματος
- `imgGpsPin`: Εικονίδιο Gps Pin.
- `issueAddress`: Η διεύθυνση όπου δηλώθηκε το ζήτημα.
- `imgClock`: Εικονίδιο ρολόι.
- `submitDateAndTime`: Ημερομηνία και ώρα καταχώρησης του ζητήματος.
- `btnEdit`: Κουμπί επεξεργασίας ενός ζητήματος.

Μέθοδοι της Κλάσης:

- `UserPendingIssueTeaser()`: Μέθοδος δόμησης μιας οντότητας της κλάσης .



- `btnEditClick`: Συνάρτηση που εκτελείτε όταν πατηθεί το κουμπί επεξεργασίας ενός ζητήματος. Μεταφέρει τον έλεγχο στις συναρτήσεις `showUserIssueManagmentPage()` και `loadIssue(issueID:integer)` της κλάσης `UserIssuesManagmentController`.

Boundary UserPendingIssuesPageGUI

«Boundary» UserPendingIssuesPageGUI
-Logo: Image -infoAndDate: String -btnBack: Button -lblPageTitle: label -imgAttention: icon -lblMessage: label -pendingIssuesList: list <UserPendingIssueTeaser>
+UserPendingIssuesPageGUI(): void +btnBackClick()

Η κλάση αυτή αποτελεί την σελίδα με τα εκκρεμή ζητήματα του χρήστη.

Ιδιότητες της Κλάσης:

- `Logo`: Το λογότυπο της εφαρμογής MyDimos.
- `infoAndDate`: Κείμενο που απεικονίζει το όνομα του δήμου καθώς και την σημερινή ημερομηνία.
- `btnBack`: Κουμπί για επιστροφή σε προηγούμενη οθόνη.
- `lblPageTitle`: Ο τίτλος της σελίδας.
- `imgAttention`: Εικονίδιο έμφασης (κύκλος με θαυμαστικό).
- `lblMessage`: Μήνυμα σχετικό με τα εκκρεμή ζητήματα του χρήστη.
- `pendingIssuesList`: Μια λίστα με αντικείμενα τύπου `UserPendingIssueTeaser` (συνοπτική παρουσίαση εκκρεμούς ζητήματος)

Μέθοδοι της Κλάσης:

- `UserPendingIssuesPageGUI ()` : Μέθοδος δόμησης μιας οντότητας της κλάσης.
- `btnBackClick()`: Συνάρτηση που εκτελείτε μόλις πατηθεί το κουμπί `btnBack`, μεταφέρει τον έλεγχο στην συνάρτηση `showPreviousPage()` της κλάσης `UserIssuesManagmentController`.

Boundary UserMessageView

«Boundary» UserMessageView
-type: string -imgIcon: icon -message: string -nColor: integer -isVisible: boolean -btnCloseMessage: Button
+UserMessageView(): void +UserMessageView(string, icon, string, integer, boolean, Button): void +setIsVisible(Vis: boolean): void +getIsVisible(): boolean +btnCloseMessageClick(): void

Η κλάση αυτή αποτελεί κλάση για την δημιουργία μηνυμάτων σχετικών με την επιτυχία ή αποτυχία μιας ενέργειας του χρήστη.

Ιδιότητες της Κλάσης:

- `type`: Ο τύπος του μηνύματος.
- `imgIcon`: Εικονίδιο σχετικό με το κείμενο.
- `message`: Το μήνυμα προς τον χρήστη.
- `nColor`: Το χρώμα του μηνύματος.
- `isVisible`: Δυαδική μεταβλητή για την ορατότητα του μηνύματος.



- `btnCloseMessage`: Κουμπί για το κλείσιμο του μηνύματος .

Μέθοδοι της Κλάσης:

- `UserMessageView()` , `UserMessageView(string, icon, string, Boolean, Button)` : Μέθοδοι δόμησης μιας οντότητας της κλάσης.
- `setVisible(Vis:Boolean)`: Θέτει την μεταβλητή `isVisible` .
- `getVisible()`: Επιστρέφει την τιμή της μεταβλητής `isVisible`.
- `btnCloseMessageClick()`: Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί `btnCloseMessage` και κλείνει το παράθυρο του μηνύματος.

Boundary UserConfirmationDialogView

«Boundary» UserConfirmationDialogView
-imgIcon: icon -dialogMessage: string -btnApply: Button -btnDiscard: Button
+UserConfirmationDialogView(): void +UserConfirmationDialogView(icon, string, Button, Button): void +btnApplyClick(): void +btnDiscardClick(): void

Η κλάση αυτή αποτελεί κλάση για την δημιουργία παραθύρων επιβεβαίωσης των ενεργειών του χρήστη.

Ιδιότητες της Κλάσης:

- `imgIcon`: Εικονίδιο σχετικό με το κείμενο.
- `dialogMessage`: Μήνυμα σχετικό με το παράθυρο επιβεβαίωσης.
- `btnApply`: Κουμπί επιβεβαίωσης ενέργειας .
- `btnDiscard`: Κουμπί απόρριψης ενέργειας .

Μέθοδοι της Κλάσης:

- `UserConfirmationDialogView()`, `UserConfirmationDialogView(icon, string, Button, Button)`: Μέθοδοι δόμησης μιας οντότητας της κλάσης.
- `setVisible(Vis:Boolean)`: Θέτει την μεταβλητή `isVisible` .
- `btnApplyClick()`: Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί `btnApply` και σηματοδοτεί την επιβεβαίωση της ενέργειας από την χρήστη. Μεταφέρει τον έλεγχο στην συνάρτηση `showUserPendingIssuesPage()` της κλάσης `UserIssuesManagementController`.
- `btnDiscardClick()`: Συνάρτηση που εκτελείται μόλις πατηθεί το κουμπί `btnDiscard` και σηματοδοτεί την επιβεβαίωση της ενέργειας από την χρήστη. Μεταφέρει τον έλεγχο στην συνάρτηση `showUserPendingIssuesPage()` της κλάσης `UserIssuesManagementController`.

Controller PendingIssuesListController

«Controller» PendingIssuesListController
-userID: integer -userIssuesEntities: list<Issues>
+PendingIssuesListController(): void +checkPendingIssuesExistence(userID): IblMessage +getPendingIssuesFromUserID(userID: integer): list<Issues> +printIssueAsTeaser(issue: Issue): UserPendingIssueTeaser

Η κλάση αυτή αποτελεί τον ελεγκτή(controller) που είναι υπεύθυνος για την απεικόνιση των οντοτήτων `UserPendingIssueTeaser` στην σελίδα όπου εμφανίζονται τα εκκρεμή ζητήματα του χρήστη.

Ιδιότητες της Κλάσης:

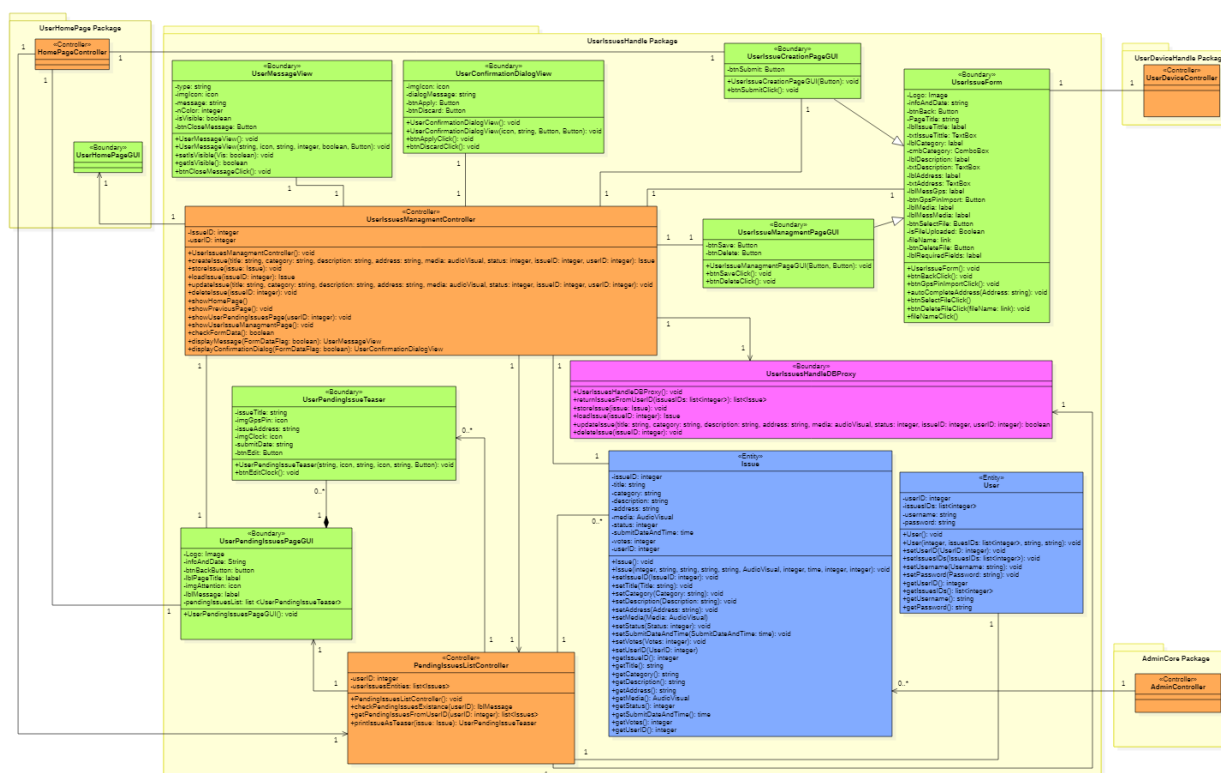
- `userID`: Ο μοναδικός χαρακτηριστικός αριθμός (ID) του χρήστη.

- **userIssuesEntities:** Η λίστα των ζητημάτων των οποίων οι συντομεύσεις θα εμφανιστούν στην λίστα με τα εκκρεμή ζητήματα του χρήστη.

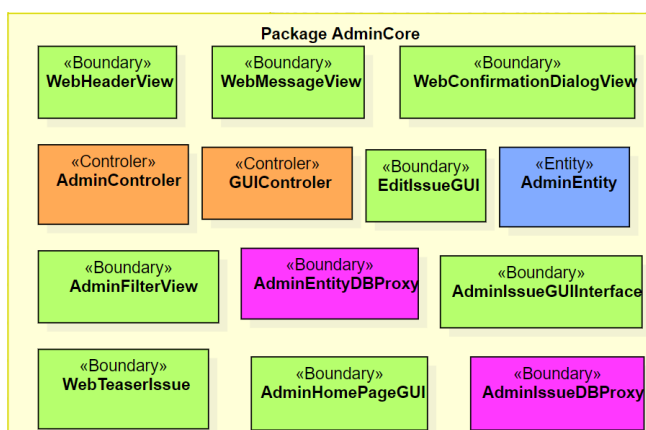
Μέθοδοι της Κλάσης:

- PendingIssuesListController() : Μέθοδος δόμησης μιας οντότητας της κλάσης.
- checkPendingIssuesExistence(userID): Συνάρτηση που ελέγχει εάν υπάρχουν καταχωρημένα εκκρεμή ζητήματα από τον χρήστη με ID userID. Επιστρέφοντας ανάλογο μήνυμα στην σελίδα με τα εκκρεμή ζητήματα του χρήστη (UserPendingIssuesPageGUI).
- getPendingIssuesFromUserID(userID: integer): Συνάρτηση που επιστρέφει στα τα εκκρεμή ζητήματα του χρήστη (εάν υπάρχουν).
- printIssueAsTeaser(issue: issue): Συνάρτηση που λαμβάνει ένα ζήτημα από την λίστα των εκκρεμών ζητημάτων που λαμβάνει η παραπάνω συνάρτηση και το εκτυπώνει ως UserPendingIssueTeaser.

4.1.2.2 Διαγράμματα κλάσεων - Πακέτο UserIssuesHandle



4.1.3 Πακέτο AdminCore



Το πακέτο αυτό περιέχει τις βασικές κλάσεις που απαιτούνται για την λειτουργικότητα του διαχειριστή.

4.1.3.1 Ορισμοί – Πακέτο AdminCore

Κλάση Boundary WebHeaderView

«Boundary» WebHeaderView
-logo: image -municipality: string -date: string -title: string
+WebHeaderView() +WebHeaderView(image, string, string, string): void

Η κλάση αυτή εκφράζει την κεφαλίδα σε όλες τις σελίδες του διαχειριστή.

Χαρακτηριστικά της κλάσης:

- logo: Λογότυπο της εφαρμογής.
- municipality: Αλφαριθμητικό που περιέχει το όνομα του οργανισμού (δήμου) που χρησιμοποιεί την εφαρμογή
- date: Ημερομηνία
- title: Αλφαριθμητικό με τον τίτλο της σελίδας που βρίσκεται ο διαχειριστής

Μέθοδοι της κλάσης:

- WebHeaderView(): default constructor
- WebHeaderView(image, string, string, string): Constructor που δέχεται τιμές για τα αντίστοιχα χαρακτηριστικά της κλάσης.

Κλάση Boundary WebMessageView

«Boundary» WebMessageView
-srtType: string -strMessage: string -imgIcon: icon -nColor: integer -isVisible: boolean -btnCloseMessage: Button
+WebMessageView() +WebMessageView(string, string, icon, integer, boolean) +setVisible(boolean): void +getVisible(): boolean +setMessage(string, string, icon, integer): void +btnCloseMessageClick(): void

Η κλάση αυτή αντιπροσωπεύει τα διάφορα μηνύματα που θα δείχνει η εφαρμογή στον χρήστη.

Χαρακτηριστικά της κλάσης:

- strType: Αλφαριθμητικό με το είδος μηνύματος (σφάλμα, μήνυμα, προσοχή)
- strMessage: Το κείμενο του μηνύματος.
- imgIcon: Εικονίδιο ανάλογα με τον τύπο του σφάλματος.
- nColor: Αριθμός του χρώματος που θα έχει το πλαίσιο του μηνύματος.
- isVisible: Boolean που ελέγχει την ορατότητα του μηνύματος
- btnCloseMessage: Κουμπί που κλείνει το μήνυμα

Μέθοδοι της κλάσης:

- WebMessageView: default constructor



- `WebMessageView(string, string, icon, integer)`: Constructor που δέχεται τιμές για τα αντίστοιχα χαρακτηριστικά της κλάσης.
- `setVisible(boolean)`: Θέτει την ορατότητα του πλαισίου μηνύματος
- `getVisible()`: Επιστρέφει την κατάσταση ορατότητας του πλαισίου μηνύματος
- `setMessage(string, string, icon, integer)`: Θέτει τα αντίστοιχα χαρακτηριστικά του μηνύματος
- `btnCloseMessageClick()`: συνάρτηση που εκτελείται όταν πατηθεί το κουμπί `btnCloseMessage`

Κλάση `WebConfirmationDialogView`

«Boundary» WebConfirmationDialogView
-imgIcon: icon -dialogMessage: string -btnApply: Button -btnDiscard: Button
+WebConfirmationDialogView(): void +WebConfirmationDialogView(icon, string, Button, Button): void +btnApplyClick(): void +btnDiscardClick(): void

Η κλάση αυτή αντιπροσωπεύει τον διάλογο επιβεβαίωσης που εμφανίζεται όταν ο διαχειριστής κάνει κρίσιμες ενέργειες.

Χαρακτηριστικά της κλάσης:

- `imgIcon`: εικονίδιο στο παράθυρο διαλόγου
- `dialogMessage`: Το μήνυμα του διαλόγου
- `btnApply`: Το κουμπί επιβεβαίωσης της ενέργειας
- `btnDiscard`: Το κουμπί ακύρωσης της ενέργειας

Μέθοδοι της κλάσης:

- `WebConfirmationDialogView()`: default constructor
- `WebConfirmationDialogView(icon, string, Button, Button)`: constructor που αρχικοποιεί τα χαρακτηριστικά του αντικειμένου
- `btnApplyClick()`: Η συνάρτηση που εκτελείται όταν πατηθεί το κουμπί επιβεβαίωσης.
- `btnDiscardClick()`: Η συνάρτηση που εκτελείται όταν πατηθεί το κουμπί ακύρωσης.

Κλάση Boundary `AdminHomePageGUI`

«Boundary» AdminHomePageGUI
-btnEditCategory: button -teaserIssuesIds: integer[0..*] -lblFound: Label
+AdminHomePageGUI() +btnEditCategoryClick(): void +setTeaserIssuesIds(new: integer[0..*]): void +getTeaserIssuesIds(): integer[0..*]

Η κλάση αυτή αντιπροσωπεύει την αρχική σελίδα του διαχειριστή.

Χαρακτηριστικά της κλάσης:

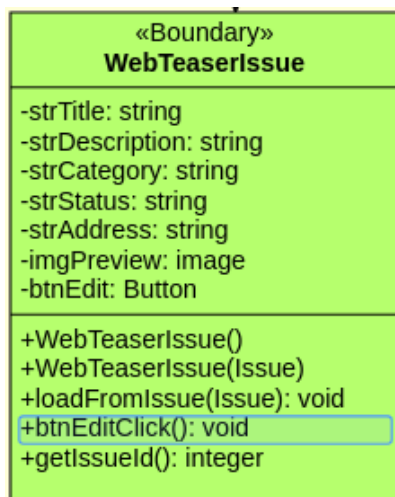
- `btnEditCategory`: κουμπί που οδηγεί στην σελίδα διαχείρισης κατηγοριών.
- `TeaserIssuesIds`: πίνακας με τα id των ζητημάτων που προβάλλονται.
- `LblFound`: Κείμενο "Βρέθηκαν χ ζητήματα"

Μέθοδοι της κλάσης:



- AdminHomePageGUI() : default constructor
- btnEditCategoryClick() : η συνάρτηση που εκτελείται όταν πατηθεί το κουμπί
- setTeaserIssuesIds(new): θέτει τον πίνακα με τα id των ζητημάτων που βλέπει ο διαχειριστής
- getTeaserIssuesIds(new): επιστρέφει τον πίνακα με τα id των ζητημάτων που βλέπει ο διαχειριστής

Κλάση Boundary WebTeaserIssue



Αυτή η κλάση είναι υπεύθυνη για την προβολή ενός ζητήματος στην λίστα.

Χαρακτηριστικά της κλάσης:

- strTitle, strDescription, ..., strSubmitDate: Αλφαριθμητικά που περιέχουν τα δεδομένα του ζητήματος που προβάλλει η κλάση.
- imgPreview: Μια προεπισκόπηση, εικόνα μικρού μεγέθους, του οπτικοακουστικού υλικού του προβλήματος.
- btnEdit: κουμπί επεξεργασίας ζητήματος

Μέθοδοι της κλάσης:

- WebTeaserIssue(): default constructor
- WebTeaserIssue(Issue): constructor που δέχεται μια αναφορά σε αντικείμενο issue και φορτώνει τα δεδομένα του για να τα προβάλλει
- loadFromIssue(Issue): δέχεται μια αναφορά σε αντικείμενο issue και φορτώνει τα δεδομένα του για να τα προβάλλει
- btnEditClick(): συνάρτηση που εκτελείται όταν πατηθεί το κουμπί επεξεργασίας ζητήματος
- getIssueId(): επιστρέφει το id του ζητήματος που προβάλλεται

Κλάση AdminBoundaryFilterView



«Boundary» AdminFilterView
-lblHeader: label -lblSearch: label -lblCategory: label -lblIssueStatus: label -lblAddress: label -lblSubmissionDate: label -lblFrom: label -lblUntil: label -txtSearch: TextBox -txtAddress: TextBox -btnApply: Button -btnCancelFilters: Button -strFilters: string
+AdminFilterView() +AdminFilterView(activeFilters: string) +btnApplyClick(): void +btnCancelFiltersClick(): void +getActiveFilters(): string +setActiveFilters(string): void

Η κλάση αυτή αντιπροσωπεύει την διεπαφή χρήση για τα φίλτρα αναζήτησης.

Χαρακτηριστικά της κλάσης:

- lblHeader, ..., lblUntil: labels με κείμενο τα ονόματα των διαθέσιμων φίλτρων
- txtSearch: textbox για εισαγωγή αλφαριθμητικού αναζήτησης
- txtAddress: textbox για εισαγωγή αλφαριθμητικού διεύθυνσης
- btnApply: κουμπί που εφαρμόζει τα επιλεγμένα φίλτρα
- btnCancelFilters: κουμπί που αφαιρεί όλα τα φίλτρα
- strFilters: αλφαριθμητικό που κωδικοποιεί όλα τα εφαρμοσμένα φίλτρα.

Μέθοδοι της κλάσης:

- AdminFilterView(): default constructor
- AdminFilterView(activeFilters): constructor που δέχεται αλφαριθμητικό με φίλτρα που έχουν εφαρμοστεί και αρχικοποιεί κατάλληλα τα γραφικά στοιχεία της κλάσης.
- btnApplyClick(): η συνάρτηση που εκτελείται όταν πατηθεί το κουμπί εφαρμογής φίλτρων.
- btnCancelFiltersClick(): η συνάρτηση που εκτελείται όταν πατηθεί το κουμπί ακύρωσης φίλτρων.
- setActiveFilters(string): θέτει τα ενεργά φίλτρα αναζήτησης
- getActiveFilters(string): επιστρέφει τα ενεργά φίλτρα αναζήτησης

Κλάση Controller GUIController

«Controler» GUIController
-nAdminId: integer -nEditIssueId: integer
+GUIController(nAdminId: integer) +createErrorMessage(string): WebMessageView +createWarningMessage(string): WebMessageView +createSuccessMessage(string): WebMessageView +gotoHome(): AdminHomePageGUI +gotoEditPage(issueId: integer): null +createConfirmationDialog(string): WebConfirmationDialogView

Αυτή η κλάση είναι υπεύθυνη για τον χειρισμό των γραφικών διεπαφών του διαχειριστή.

Χαρακτηριστικά της κλάσης:

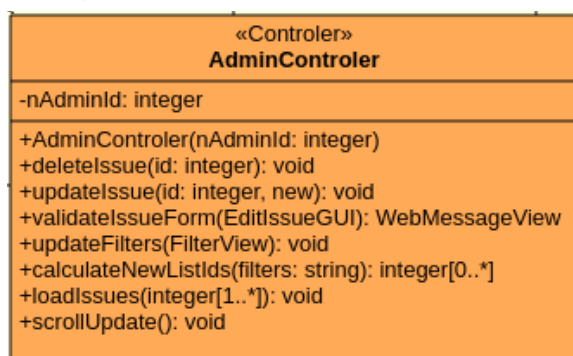


- nAdminId: Το id του διαχειριστή
- nEditIssueId: το id του ζητήματος υπό επεξεργασία (εάν ο διαχειριστής επεξεργάζεται κάποιο ζήτημα).

Μέθοδοι της κλάσης:

- GUIController(): default constructor
- createErrorMessage(string): δημιουργεί ένα μήνυμα σφάλματος
- createWarningMessage(string): δημιουργεί ένα μήνυμα προειδοποίησης
- createSuccessMessage(string): δημιουργεί μια αναφορά επιτυχίας
- gotoHome(): κατευθύνει τον διαχειριστή στην αρχική του σελίδα
- gotoEditPage(issueId): κατευθύνει τον διαχειριστή στην σελίδα διαχείρισης ενός ζητήματος
- createConfirmationDialog(string): δημιουργεί έναν διάλογο επιβεβαίωσης

Κλάση Controller AdminController



Αυτή η κλάση επιτελεί τις βασικές λειτουργίες του διαχειριστή

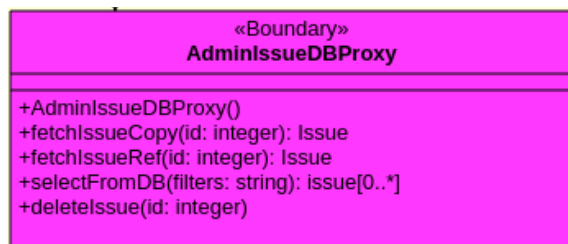
Χαρακτηριστικά της κλάσης:

- nAdminId: το id του διαχειριστή

Μέθοδοι της κλάσης:

- AdminController(id): constructor που δέχεται το id του διαχειριστή
- deleteIssue(id): αναλαμβάνει την διαγραφή του ζητήματος με το συγκεκριμένο id
- updateIssue(id, new): αναλαμβάνει την ανανέωση των πληροφοριών ενός ζητήματος, μετά από επεξεργασία. Δέχεται το id του ζητήματος και μια αναφορά για την διεπαφή επεξεργασίας, από την οποία αντλεί τα καινούρια δεδομένα.
- validateIssueForm(EditIssueGUI): δέχεται μια αναφορά για την διεπαφή επεξεργασίας και ελέγχει την εγκυρότητα των καινούριων δεδομένων του ζητήματος, πριν γίνει η καταχώρηση στη βάση δεδομένων.
- updateFilters(FilterView): Εξετάζει τις αλλαγές φίλτρων που έκανε ο χρήστης και ανανεώνει κατάλληλα τα ζητήματα της λίστας
- calculateNewListIds(filters): Δέχεται μια συμβολοσειρά που αντιπροσωπεύει τα φίλτρα της αναζήτησης και επιστρέφει ένα πίνακα με τα id των ζητημάτων που ταιριάζουν.
- scrollUpdate(): ανανεώνει την λίστα όταν ο χρήστης scrollάρει
- loadIssues(integer[1..*]): εκκινεί την διαδικασία φόρτωσης για τα teasers με τα id ορίσματα που δέχεται.

Κλάση Boundary AdminIssueDBProxy

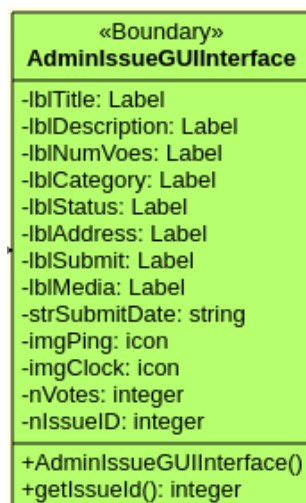


Αναλαμβάνει την μεσολάβηση της επικοινωνίας των κλάσεων της εφαρμογής με την βάση δεδομένων για τα ζητήματα.

Μέθοδοι της κλάσης:

- IssueDBProxy(): default constructor
- fetchIssueCopy(id): επιστρέφει ένα αντίγραφο αντικειμένου κλάσης Issue
- fetchIssueRef(id): επιστρέφει μια αναφορά για το αντικείμενο της κλάσης Issue, με στόχο την επεξεργασία του.
- selectFromDB(filters): επιστρέφει πίνακα από αντικείμενα που ταιριάζουν στα δοσμένα φίλτρα
- deleteIssue(id): διαγράφει το ζήτημα με το δοσμένο id από την βάση δεδομένων.

Κλάση Boundary AdminIssueGUIInterface



Βασική κλάση (interface) για προβολή ενός ζητήματος. Κληρονομείται από την EditIssueGUI, η οποία επιτρέπει την επεξεργασία ενός ζητήματος, και από την WebTeaserView, η οποία προβάλλει τα δεδομένα ενός ζητήματος.

Χαρακτηριστικά της κλάσης:

- lblTitle, ..., lblMedia: Ετικέτες με κείμενο που συνοδεύουν κάθε δεδομένο του ζητήματος (βλέπε επιδεικτικό παράθυρο)
- strSubmitDate: η ημερομηνία της καταχώρησης του ζητήματος στο σύστημα
- imgPing, imgClock: εικονίδια τοποθεσίας και ρολογιού αντίστοιχα (βλ επιδεικτικό παράθυρο)
- nVotes: ο αριθμός των ψήφων που έχει το ζήτημα
- nIssueID: ο αριθμός id του ζητήματος

Μέθοδοι της κλάσης:

- AdminIssueGUIInterface(): default constructor
- getIssueId: επιστρέφει το id του ζητήματος που προβάλλεται/είναι υπό επεξεργασία,

Κλάση Boundary EditIssueGUI



«Boundary» EditIssueGUI
-txtTitle: TextBox -txtDescription: TextBox -spnCategory: Spinner -spnStatus: Spinner -textAddress: TextBox -media: Audiovisual -btnSave: Button -btnCancel: Button -btnDelete: Button
+EditIssueGUI() +EditIssueGUI(Issue) +btnSaveClick(): void +btnCancelClick(): void +btnDeleteClick(): void

Αντιπροσωπεύει την σελίδα διαχείρισης ενός ζητήματος.

Χαρακτηριστικά της κλάσης:

- txtTitle, txtDescription, txtAddress: πλαίσια για την συμπλήρωση του τίτλου, της περιγραφής και της διεύθυνσης του ζητήματος αντίστοιχα
- spnCategory, spnStatus: spinners για την επιλογή της κατηγορίας και της κατάστασης του ζητήματος αντίστοιχα.
- media: το οπτικοακουστικό υλικό που συνοδεύει το ζήτημα
- btnSave, btnCancel, btnDelete: κουμπιά αποθήκευσης, ακύρωσης επεξεργασίας και διαγραφής ζητήματος αντίστοιχα.

Μέθοδοι της κλάσης:

- EditIssueGUI(): default constructor
- EditIssueGUI(Issue): constructor που δέχεται μια αναφορά σε ένα ζήτημα και αντλεί τις πληροφορίες από αυτό
- btnSaveClick(), btnCancelClick(), btnDeleteClick(): οι συνάρτησεις που εκτελούνται όταν πατηθούν τα κουμπιά αποθήκευσης, ακύρωσης επεξεργασίας και διαγραφής ζητήματος αντίστοιχα.

Entity AdminEntity

AdminEntity
-adminID: integer -adminUsername: string -adminPassword: string
+AdminEntity(us, pass): void +setAdminID(ID: integer): void +setAdminUsername(us: string): void +setAdminPass(Pass: string): void +getAdminID(): integer +getAdminUsername(): string +getAdminPassword(): string

Η κλάση παρουσιάζει τον διαχειριστή ως οντότητα.

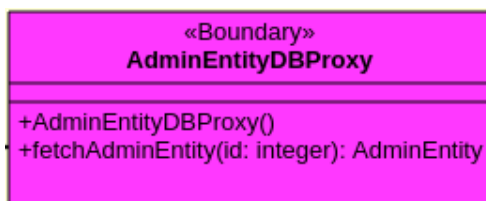
Χαρακτηριστικά της κλάσης

- adminID : Ο χαρακτηριστικός αριθμός για κάθε διαχειριστή.
- adminUsername: Το username για κάθε διαχειριστή.
- adminPassword: Ο κωδικός πρόσβασης για κάθε διαχειριστή.

Μέθοδοι της κλάσης

- AdminEntity(us,password):Constructor της κλάσης
- setAdminID(ID:integer): Setter για τη μεταβλητή adminID του Entity.
- setAdminUsername(us:string): Setter για τη μεταβλητή AdminUsername του Entity.
- setAdminPass(Pass:string):Setter για τη μεταβλητή AdminPass του Entity
- getAdminID(): Getter για τη μεταβλητή adminID του Entity.
- getAdminUsername(): Getter για τη μεταβλητή AdminUsername του Entity.
- getAdminPass():Getter για τη μεταβλητή AdminPass του Entity

Κλάση AdminEntityDBProxy

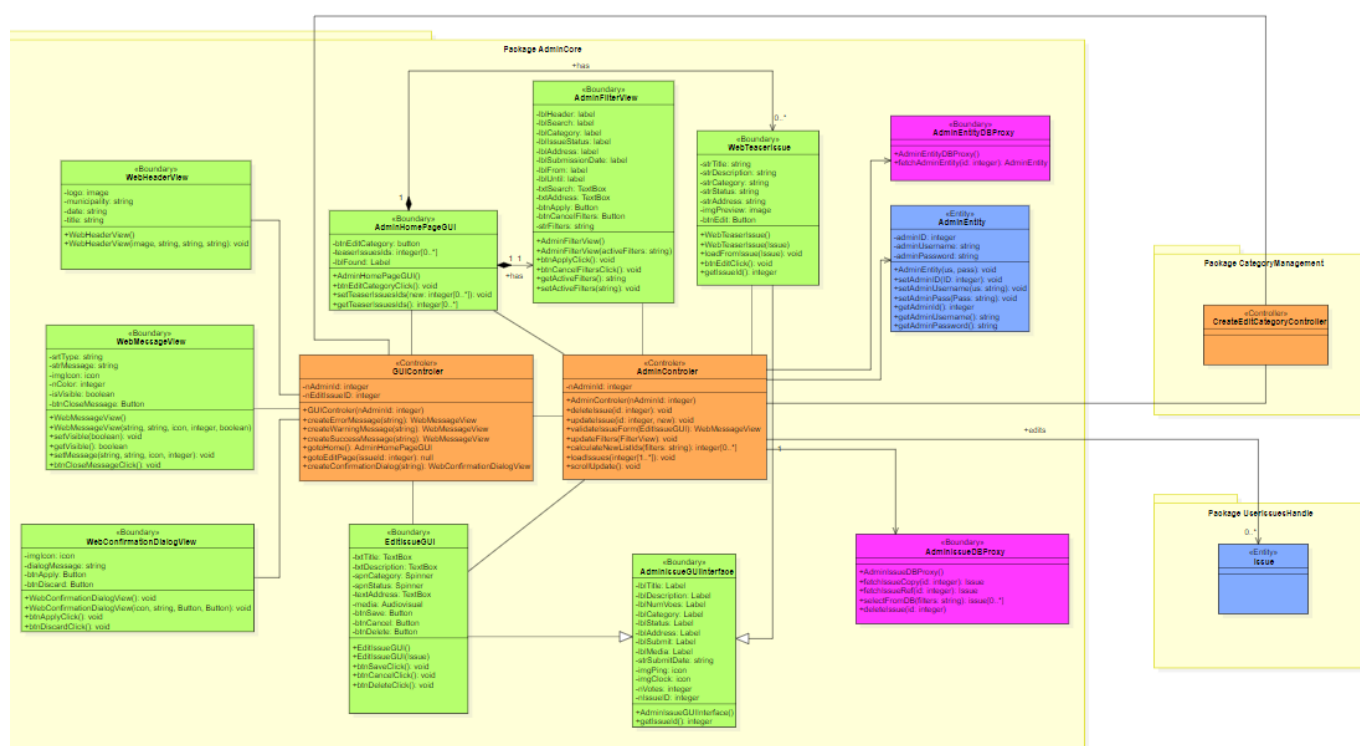


Αναλαμβάνει την μεσολάβηση της επικοινωνίας των κλάσεων της εφαρμογής με την βάση δεδομένων για τους διαχειριστές.

Μέθοδοι της κλάσης:

- AdminEntityDBProxy(): default constructor
- fetchAdminEntity(id:integer): φορτώνει από την βάση δεδομένων το αντικείμενο AdminEntity με το δοσμένο id.

4.1.3.2 Διαγράμματα κλάσεων – Πακέτο AdminCore

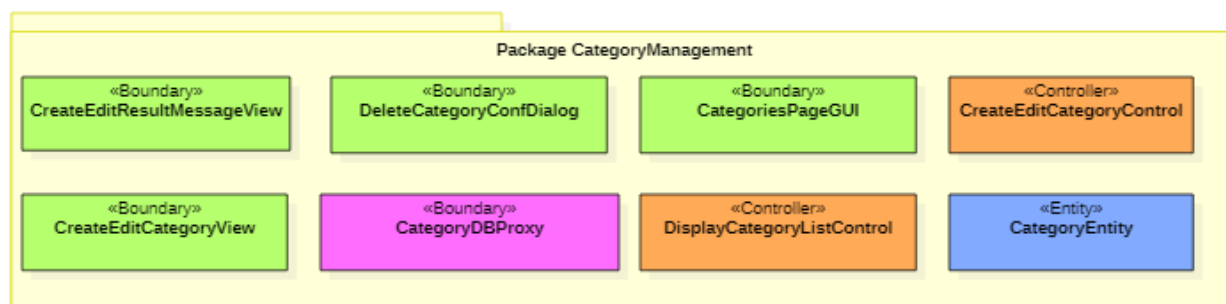




4.2 Πακέτο λεξιλογίου σεναρίων μέσης προτεραιότητας

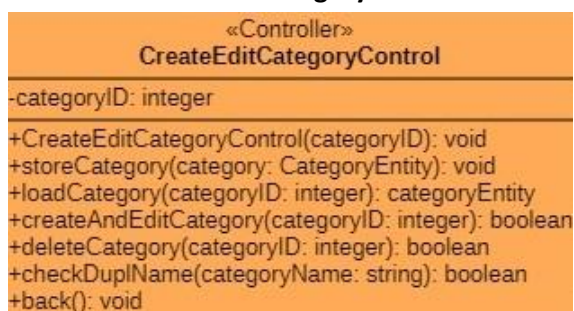
4.2.1 Πακέτο CategoryManagement

Το πακέτο αυτό αναφέρεται στην δυνατότητα του admin, να διαχειριστεί τις κατηγορίες των ζητημάτων. Αναλυτικότερα, του δίνεται η δυνατότητα να προσθέσει μια κατηγορία, να τροποποιήσει μια υπάρχουσα κατηγορία και να διαγράψει όποια κατηγορία θελήσει. Παρακάτω, αναλύονται οι κλάσεις που ανήκουν στο συγκεκριμένο πακέτο.



4.2.1.1 Ορισμοί – Πακέτο CategoryManagement

Controller CreateEditCategoryControl



Ο ελεγκτής αυτός περιέχει τις κατάλληλες συναρτήσεις για τη δημιουργία και την τροποποίηση μιας κατηγορίας.

Χαρακτηριστικά της κλάσης

- categoryID : Ο χαρακτηριστικός αριθμός για κάθε κατηγορία.

Μέθοδοι της κλάσης

- CreateEditCategoryControl(categoryID) : Συνάρτηση δόμησης της κλάσης.
- storeCategory(category:CategoryEntity) : Η μέθοδος αυτή καλεί την αντίστοιχη συνάρτηση αποθήκευσης κατηγορίας στο proxy αντικείμενο.
- loadCategory(categoryID:integer): Η μέθοδος αυτή καλεί την αντίστοιχη συνάρτηση φόρτωσης κατηγορίας στο proxy αντικείμενο.
- createAndEditCategory(categoryID:integer): Η μέθοδος καλείται όταν ο admin θέλει να δημιουργήσει ή να τροποποιήσει μια κατηγορία.
- deleteCategory(categoryID:integer): Η μέθοδος καλείται όταν ο admin θέλει να διαγράψει μια κατηγορία.
- checkDuplName(categoryName:string): Η μέθοδος καλείται για να ελεγχθεί αν το όνομα που δίνει ο admin σε μια νέα κατηγορία, έχει ήδη δοθεί.
- back(): Η μέθοδος επιστρέφει τον admin στη σελίδα κατηγοριών.

Boundary CreateEditResultMessageView



«Boundary» CreateEditResultMessageView
-result: string -btnBack: Button
+CreateEditResultMessageView(string, string): void +displayResult(result): void +btnBackClick(): void

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση μηνύματος επιτυχίας ή αποτυχίας τροποποίησης ή δημιουργίας κατηγορίας.

Χαρακτηριστικά της κλάσης

- result : Μήνυμα σχετικό με την επιτυχία ή αποτυχία δημιουργίας ή τροποποίησης μιας κατηγορίας.
- btnBack: Κουμπί για το κλείσιμο του μηνύματος και για την επιστροφή στη σελίδα κατηγοριών.

Μέθοδοι της κλάσης

- CreateEditResultMessageView(string, string): Μέθοδος δόμησης της κλάσης.
- displayResult(result): Συνάρτηση που εκτυπώνει το σχετικό μήνυμα.
- btnBackClick(): Συνάρτηση που εκτελείται όταν πατηθεί το κουμπί btnBack, καλεί τη συνάρτηση back() του EditAndCreateController.

Boundary DeleteCategoryConfDialog

«Boundary» DeleteCategoryConfDialog
-categoryID: integer -btnCancel: Button -btnConfirm: Button
+DeleteCategoryConfDialog(catName): void +btnCancelClick(): void +btnConfirmClick(): void +displayMessage(categoryID: integer): void

Η κλάση αυτή αποτελεί παράθυρο για την επιβεβαίωση της διαγραφής μιας κατηγορίας.

Χαρακτηριστικά της κλάσης

- categoryID : Ο χαρακτηριστικός αριθμός για κάθε κατηγορία.
- btnCancel: Το κουμπί αυτό ακυρώνει τη διαγραφή κατηγορίας.
- btnConfirm: Το κουμπί αυτό επικυρώνει τη διαγραφή κατηγορίας.

Μέθοδοι της κλάσης

- DeleteCategoryConfDialog(catName): Συνάρτηση δόμησης της κλάσης.
- btnCancelClick(): Η μέθοδος εκτελείτε μόλις πατηθεί το κουμπί btnCancel , ακυρώνει την διαγραφή της κατηγορίας και επιστρέφει τον admin μέσω του CreateAndEditControl στη σελίδα των κατηγοριών.
- btnConfirmClick(): Η μέθοδος εκτελείτε μόλις πατηθεί το κουμπί btnConfirm , επιβεβαιώνει την διαγραφή της κατηγορίας και καλεί τον AdminController του πακέτου AdminCore ώστε να αλλάξει η κατηγορία των ζητημάτων που ανήκαν στην κατηγορία που διαγράφεται.
- displayMessage(categoryID): Η μέθοδος εμφανίζει μήνυμα στο admin αν θέλει να διαγράψει την κατηγορία.

**Entity CategoryEntity**

«Entity» CategoryEntity
-categoryID: integer -CategoryName: string
+CategoryEntity(ID, Name): void +setCategoryID(ID: integer): void +setCategoryName(Name: string): void +getCategoryID(): integer +getCategoryName(): string

Η κλάση της οντότητας της κατηγορίας.

Χαρακτηριστικά της κλάσης

- categoryID : Ο μοναδικός χαρακτηριστικός αριθμός της κατηγορίας.
- CategoryName: Το όνομα της κατηγορίας.

Μέθοδοι της κλάσης

- CategoryEntity(ID,Name): Μέθοδος δόμησης της κλάσης
- setCategoryID(ID:integer): Θέτει τη μεταβλητή *categoryID*.
- setCategoryName(Name:string): Θέτει τη μεταβλητή *CategoryName*.
- getCategoryID(): Επιστρέφει την τιμή της μεταβλητής *CategoryID*.
- getCategoryName(Name:string): Επιστρέφει την τιμή της μεταβλητής *CategoryName*.

Boundary CreateEditCategoryView

«Boundary» CreateEditCategoryView
-categoryName: string -btnSubmit: Button -btnCancel: Button
+CreateEditCategoryView(categoryName): void +btnSubmitClick(): void +btnCancelClick(): void

Η κλάση αυτή αναπαριστά τη διεπαφή για τη δημιουργία ή τροποποίηση κατηγορίας στη σελίδα της λίστας κατηγοριών.

Χαρακτηριστικά της κλάσης.

- categoryName: Η ονομασία που θέλει να δώσει ο admin σε νέα ή τροποποιημένη κατηγορία.
- btnSubmit: Το κουμπί για την υποβολή αίτησης νέας ή τροποποιημένης κατηγορίας.
- btnCancel: Το κουμπί για ακύρωση προσθήκης ή τροποποίησης κατηγορίας.

Μέθοδοι της κλάσης

- CreateEditCategoryView(categoryName): Constructor της κλάσης
- btnSubmitClick(): Η μέθοδος εκτελείται μόλις πατηθεί το κουμπί btnSubmit, καλεί την *checkDuplName()* του controller για να ελέγξει αν το νέο όνομα είναι έγκυρο.
- btnCancelClick(): Η μέθοδος εκτελείται μόλις πατηθεί το κουμπί btnCancel, καλεί την *back()* του *createEditCategoryController* για να εμφανιστεί η λίστα των κατηγοριών.

**Controller DisplayCategoryListControl**

«Controller» DisplayCategoryListControl
+CategoryList: list<CategoryEntity>
+displayCategoryListControl(): void +displayCategoryList(): void +displayHomepage(): void

Ο ελεγκτής αυτός περιέχει τις κατάλληλες συναρτήσεις για την εμφάνιση της λίστας κατηγοριών.

Χαρακτηριστικά της κλάσης

- CategoryList : Η λίστα με τις κατηγορίες που έχουν δημιουργηθεί (αντικείμενα τύπου Entity).

Μέθοδοι της κλάσης

- displayCategoryListControl(): Μέθοδος δόμησης της κλάσης
- displayCategoryList(): Η μέθοδος αυτή εμφανίζει τη σελίδα των κατηγοριών στον admin.
- displayHomePage(): Η μέθοδος αυτή εμφανίζει την αρχική σελίδα του admin.

Proxy CategoryDBProxy

«Boundary» CategoryDBProxy
+loadCategory(categoryID: integer): CategoryEntity +storeCategoryEntity(category: categoryEntity): void +deleteCategory(categoryID): void

Αυτή η κλάση χρησιμοποιείται για την επικοινωνία με τη βάση δεδομένων με στόχο την διαχείριση των οντοτήτων των κατηγοριών που είναι καταχωρημένες σε αυτή.

Μέθοδοι της κλάσης

- storeCategory(category:categoryEntity): Μέθοδος αποθήκευσης κατηγορίας στη βάση δεδομένων.
- loadCategory(categoryID:integer): Μέθοδος φόρτωσης κατηγορίας από τη βάση δεδομένων.
- deleteCategory(categoryID:integer): Μέθοδος για την διαγραφή μιας κατηγορίας από την βάση δεδομένων.

Boundary CategoriesPageGUI

«Boundary» CategoriesPageGUI
-Logo: Image -btnIssuesManagement: Button -CategoryList: list<CategoryEntity> -MunicipalIbl: label -btnsEdit: list<Button> -btnsDeleteCategory: list<Button> -btnCreateCategory: Button
+CategoriesPageGUI(List: CategoryEntity): void +btnIssueManageClick(): void +btnEditCategoryClick(): void +btnDeleteCategoryClick(): void +btnCreateCategoryClick(): void

Η κλάση αυτή αναπαριστά την διεπαφή για την εμφάνιση των κατηγοριών και για την επεξεργασία τους.



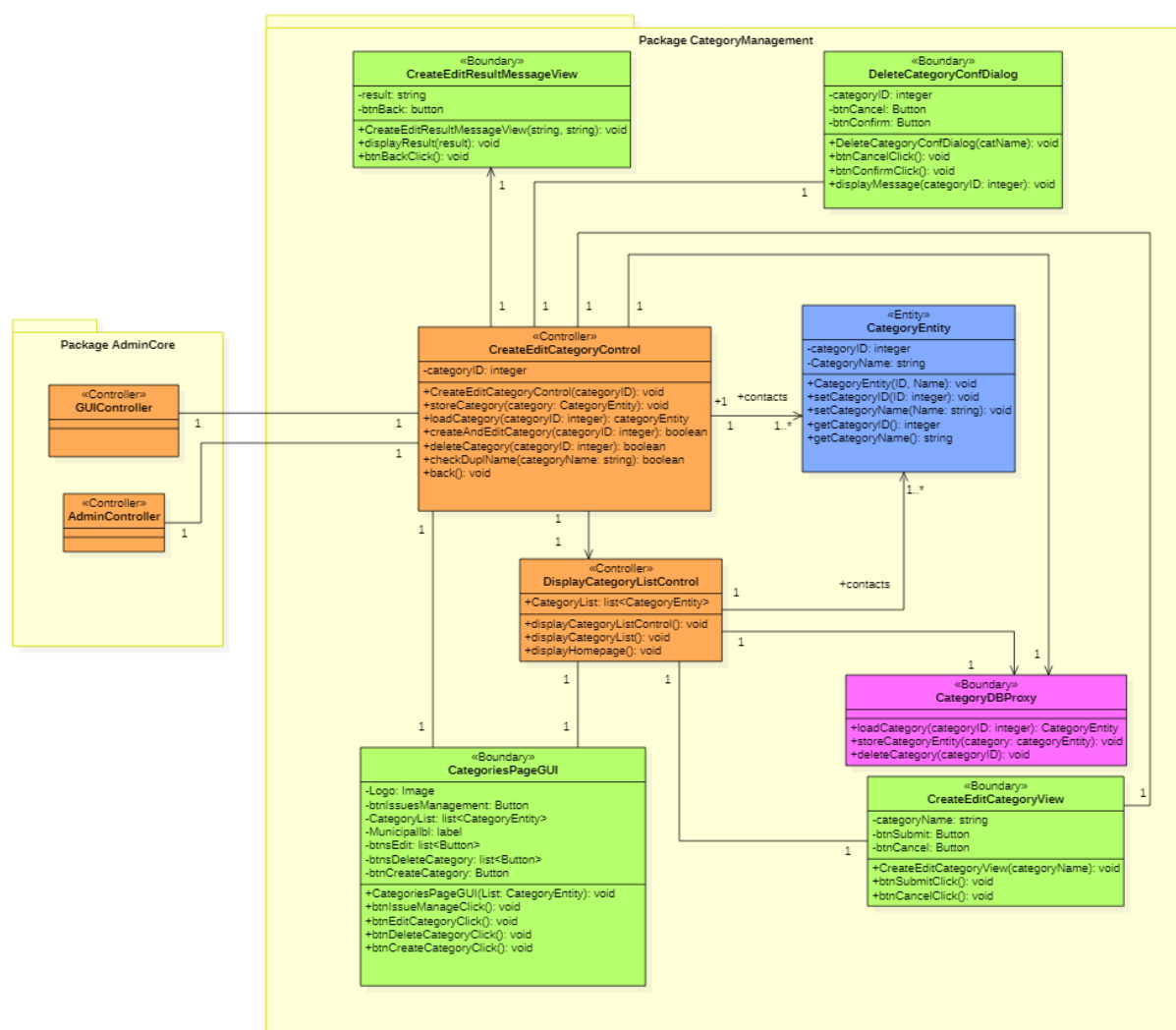
Χαρακτηριστικά της κλάσης

- Logo: Το Logo της εφαρμογής myDimos.
- btnIssuesManagement: Κουμπί για μεταφορά στη σελίδα διαχείρισης ζητημάτων.
- CategoryList: Η λίστα των κατηγοριών που βλέπει ο διαχειριστής
- Municipallbl: Ετικέτα που αναφέρει το όνομα του δήμου.
- btnsEdit : Λίστα από κουμπιά για την τροποποίηση των κατηγοριών της λίστας.
- btnsDeleteCategory: Λίστα από κουμπιά για την διαγραφή των κατηγοριών της λίστας.
- btnCreateCategory: Κουμπί για την δημιουργία κατηγορίας.

Μέθοδοι της κλάσης

- CategoryPageGUI(List:CategoryEntity): Συνάρτηση δόμησης της κλάσης.
- btnIssueManageClick(): Η μέθοδος καλεί τη συνάρτηση *back()* του *CreateAndEditController* για επιστροφή στη σελίδα διαχείρισης ζητημάτων που είναι η αρχική σελίδα του admin.
- btnEditCategoryClick(): Η μέθοδος, εκτελείται μόλις πατηθεί ένα κουμπί από την λίστα κουμπιών *btnsEditCategory*.
- btnDeleteCategoryClick(): Η μέθοδος, εκτελείται μόλις πατηθεί ένα κουμπί από την λίστα κουμπιών *btnsDeleteCategory* και καλεί τη συνάρτηση *deleteCategory()* του *CreateAndEditController*.
- btnCreateCategoryClick() Η μέθοδος καλεί τη συνάρτηση *CreateAndEditCategory()* του *CreateAndEditController*.

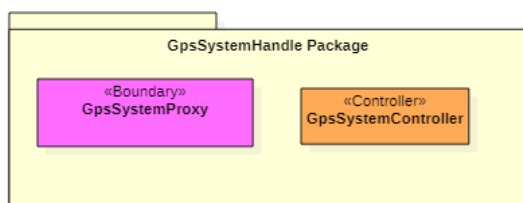
4.2.1.2 Διαγράμματα Κλάσεων – Πακέτο CategoryManagement



4.3 Πακέτα λεξιλογίου σεναρίων χαμηλής προτεραιότητας – Συμπληρωματικά πακέτα.

4.3.1 Πακέτο GpsSystemHandle

Η λειτουργικότητα του πακέτου αυτού περιορίζεται στην επικοινωνία της εφαρμογής μας με την συσκευή του χρήστη.



4.3.1.1 Ορισμοί - Πακέτο GpsSystemHandle

Controller GpsSystemController

«Controller» GpsSystemController
-deviceGpsPin: Array<float> -address: string -addressAndPinDistance: integer
+GpsSystemController(): void +getAddressFromPin(deviceGpsPin: Array<float>): string +returnIssueAddress(): string +getAddressAndPinDistance(deviceGpsPin: Array<float>, string: Address): integer +returnAddressAndPinDistance(): integer

Η κλάση αυτή αποτελεί τον controller του πακέτου GpsSystemHandle. Αν και οι περισσότερες συναρτήσεις υπάρχουν και στον *UserDeviceController* του πακέτου *UserDeviceHandle*, ορίζονται και εδώ για λόγους πληρότητας και για καλύτερο σχεδιασμό του πακέτου επικοινωνίας με το εξωτερικό σύστημα διαχείρισης ιχνών GPS.

Ιδιότητες της Κλάσης:

- **deviceGpsPin:** Οι συντεταγμένες που αντιστοιχούν στο Gps στίγμα της συσκευής του χρήστη.
- **address:** Η διεύθυνση που αντιστοιχεί στις συντεταγμένες του deviceGpsPin.
- **addressAndPinDistance:** Η απόσταση μεταξύ της διεύθυνσης ενός ζητήματος και του στίγματος Gps του χρήστη.

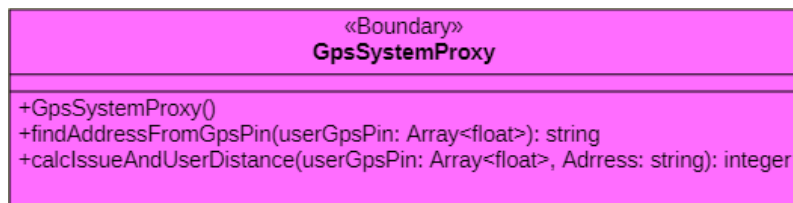
Μέθοδοι της Κλάσης:

- **GpsSystemController ()** : Μέθοδος δόμησης μιας οντότητας της κλάσης .
- **getAddressFromPin(deviceGpsPin: Array<float>): string** : Συνάρτηση για την εύρεση της διεύθυνσης που αντιστοιχεί στο στίγμα Gps του χρήστη, καλεί την συνάρτηση *findAddressFromGpsPin(in userGpsPin:Array<float>)* της κλάσης *GpsSystemProxy*.
- **returnIssueAddress(): string** : Επιστρέφει την διεύθυνση του ζητήματος που υπολογίστηκε από την αμέσως προηγούμενη συνάρτηση, στην συνάρτηση *printAddressToIssueForm()* του *userDeviceController* του πακέτου *userDeviceHandle*.
- **getAddressAndPinDistance(deviceGpsPin:Array<float>, string:Address) : string** : Συνάρτηση για την εύρεση της απόστασης μεταξύ του στίγματος Gps του χρήστη (deviceGpsPin) και της διεύθυνσης ενός ζητήματος(Address) .Καλεί την συνάρτηση *calcIssueAndUserDistance(userGpsPin:Array<float>, Address:string)* της κλάσης *GpsSystemProxy*.



- `returnAddressAndPinDistance()`: Επιστρέφει την απόσταση που υπολογίστηκε στην παραπάνω συνάρτηση, στην συνάρτηση `printDistanceToTeaser()` του `userDeviceController` του πακέτου `userDeviceHandle`.

Boundary GpsSystemProxy

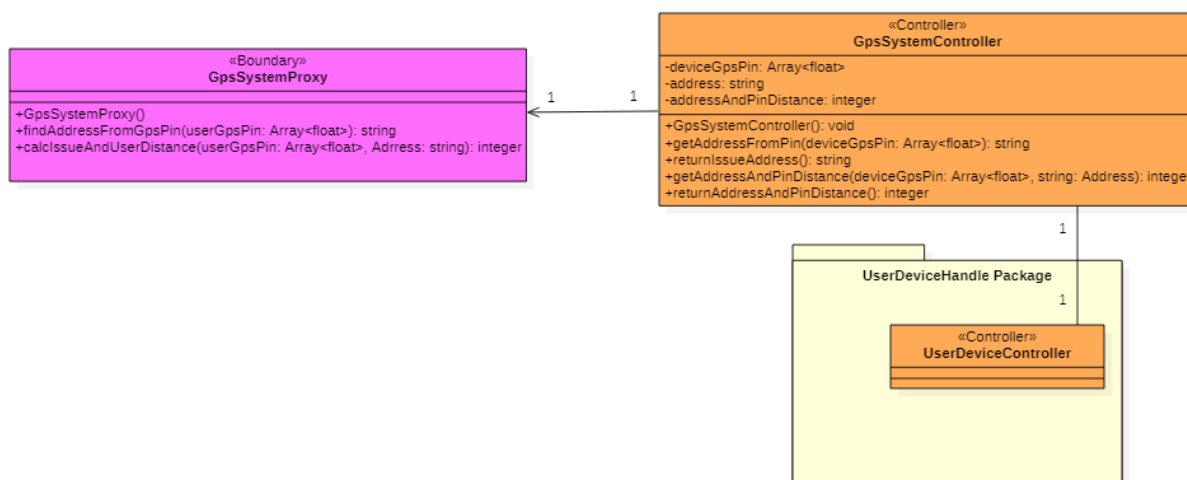


Η οριακή κλάση μέσω της οποίας επιτυγχάνεται η επικοινωνία με το σύστημα διαχείρισης ιχνών Gps.

Μέθοδοι της Κλάσης:

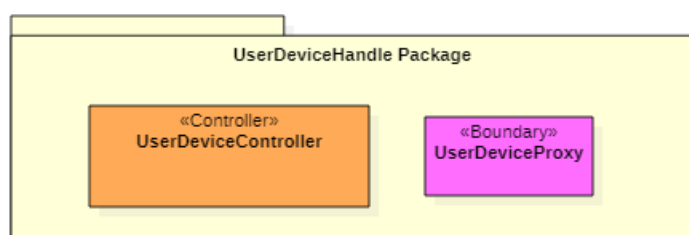
- `GpsSystemProxy()` : Μέθοδος δόμησης μιας οντότητας της κλάσης .
- `findAddressFromGpsPin(userGpsPin: Array<float>): string`: Συνάρτηση που επιστρέφει την διεύθυνση η οποία αντιστοιχεί στο Gps στίγμα που της δίνεται.
- `calcIssueAndUserDistance(userGpsPin: Array<float>, Address: string): integer`: Συνάρτηση που επιστρέφει την απόσταση μεταξύ του Gps Στίγματος και της διεύθυνσης που της δίνονται.

4.3.1.2 Διαγράμματα κλάσεων - Πακέτο GpsSystemHandle



4.3.2 Πακέτο UserDeviceHandle

Η λειτουργικότητα του πακέτου αυτού περιορίζεται στην επικοινωνία της εφαρμογής μας με την συσκευή του χρήστη.





4.3.2.1 Ορισμοί - Πακέτο GpsSystemHandle

Controller UserDeviceController

«Controller» UserDeviceController
-deviceGpsPin: Array<float> -address: string -addressAndPinDistance: integer
+UserDeviceController(): void +getGpsPinFromDevice(): Array<float> +getAddressFromGps(deviceGpsPin: Array<float>): string +printAddressToIssueForm(address: string) +getAddressAndPinDistance(deviceGpsPin: Array<float>, string: Address): integer +printDistanceToTeaser(addressAndPinDistance: integer) +showDeviceGallery(): void +selectFile(): void +loadFile(): link

Η κλάση αυτή αποτελεί τον controller του πακέτου UserDeviceHandle .

Ιδιότητες της Κλάσης:

- deviceGpsPin: Οι συντεταγμένες που αντιστοιχούν στο Gps στίγμα της συσκευής του χρήστη.
- address: Η διεύθυνση που αντιστοιχεί στις συντεταγμένες του deviceGpsPin.
- addressAndPinDistance: Η απόσταση μεταξύ της διεύθυνσης ενός ζητήματος και του στίγματος Gps του χρήστη.

Μέθοδοι της Κλάσης:

- UserDeviveController() : Μέθοδος δόμησης μιας οντότητας της κλάσης.
- getGpsPinFromDevice(): Συνάρτηση για την εύρεση του στίγματος Gps του χρήστη, καλεί την συνάρτηση *returnGpsPin()* της κλάσης *userDeviceProxy*.
- getAddressFromPin(deviceGpsPin: Array<float>): Συνάρτηση για την εύρεση της διεύθυνσης από το στίγμα Gps του χρήστη. Καλεί την συνάρτηση *getAddressFromPin()* της κλάσης *GpsSystemController*.
- printAddressToIssueForm(): Εκτυπώνει την διεύθυνση που αντιστοιχεί στο στίγμα Gps του χρήστη στην κλάση *UserIssueForm* του πακέτου *UserIssuesHandle*.
- getAddressAndPinDistance(deviceGpsPin: Array<float>, string: Address) : Συνάρτηση για την εύρεση της απόστασης μεταξύ του στίγματος Gps του χρήστη (deviceGpsPin) και της διεύθυνσης ενός ζητήματος(Address) .Καλεί την ομώνυμη συνάρτηση της κλάσης *GpsSystemController*.
- printDistanceToTeaser(addressAndPinDistance): Εκτυπώνει την τιμή εύρεση της απόστασης μεταξύ του στίγματος Gps του χρήστη και της διεύθυνσης ενός ζητήματος, στην κλάση *TesearIssue* του πακέτου *UserHomePage*.
- showDeviceGallery(): Συνάρτηση για την εμφάνιση της συλλογής οπτικοακουστικού υλικού της συσκευής του χρήστη. Καλεί την συνάρτηση *loadDeviceGallery()* της κλάσης *UserDeviceProxy*.

Boundary UserDeviceProxy

«Boundary» UserDeviceProxy
+UserDeviceProxy() +returnGpsPin(): Array<float> +loadDeviceGallery(): void

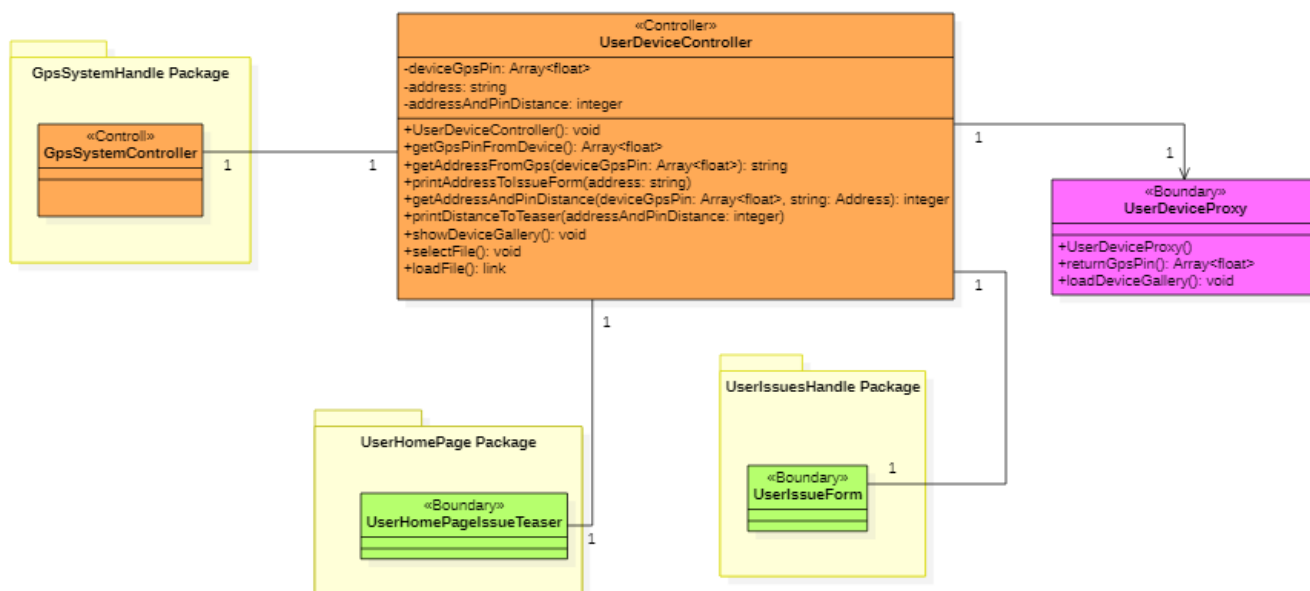
Η οριακή κλάση μέσω της οποίας επιτυγχάνεται η επικοινωνία με την συσκευή του χρήστη.

Μέθοδοι της Κλάσης:



- UserDeviceProxy() : Μέθοδος δόμησης μιας οντότητας της κλάσης.
- returnGpsPin(): Επιστρέφει το στίγμα Gps της συσκευής του χρήστη.
- loadDeviceGallery(): Φορτώνει την συλλογή οπτικοακουστικού υλικού της συσκευής του χρήστη.

4.3.2.2 Διαγράμματα κλάσεων - Πακέτο UserDeviceHandle





Παράρτημα Ι – Γλωσσάριο

Del.x.x	Deliverable x.x
ΟΑ	Ομάδα Ανάπτυξης
ΛΑ-xx	Λειτουργική Απαίτηση xx
ΜΛΑ-xx	Μη Λειτουργική Απαίτηση xx
GUI	Graphical user interface
Admin	Administrator



Παράρτημα II – Ανοιχτά Θέματα

- Εμφάνιση Χάρτη κυρίως στον διαχειριστή και δευτερευόντως στον απλό χρήστη με στίγματα από τις διευθύνσεις των ζητημάτων .
- Προσβασιμότητα της εφαρμογής σε άτομα με ειδικές ανάγκες , συμφωνία με πρότυπο WCAG 2.0.
- Άμεση σύνδεση της εφαρμογής με φορείς αρμόδιους για την επίλυση των ζητημάτων .