



Τεχνολογία Λογισμικού

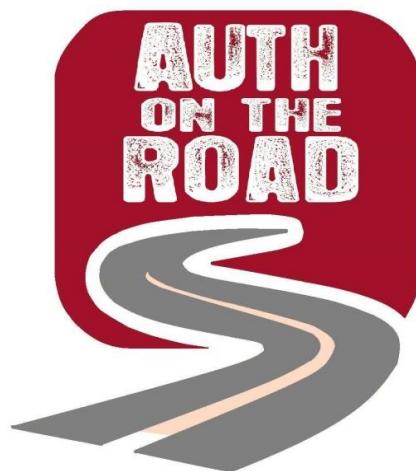
Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη 2016



**Dream, explore, discover!**

## Απαιτήσεις Λογισμικού

### (Προδιαγραφές Λογισμικού)

**Del.1.2**

**Version 0.8**

Αμπαρτσουμιάν Αλεξάνδρα [alexampa@auth.gr](mailto:alexampa@auth.gr)

Δημαρίδου Βαλασία [valia.dimaridou@gmail.com](mailto:valia.dimaridou@gmail.com)

Θεοδωρίδου Ιφιγένεια [ifigtheo@auth.gr](mailto:ifigtheo@auth.gr)

Παπαδόπουλος Βασίλης [vasipapa@auth.gr](mailto:vasipapa@auth.gr)

**10/04/2016**



## Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
A. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers* και του M. Smialek's.	0.1
A. Συμεωνίδης	29/3/2014	Προσαρμογή του εγγράφου.	0.1.3
X. Ζολώτας	2/4/2015	Προσθήκη διαγράμματος καταστάσεων	0.1.4
Auth on the road	6/4/2016	Προσθήκη πακέτων στατικής μοντελοποίησης	0.5
Auth on the road	9/4/2016	Προσθήκη διαγραμμάτων κλάσεων	0.6
Auth on the road	12/4/2016	Προσθήκη δυναμική μοντελοποίησης	0.7
Auth on the road	13/4/2016	Προσθήκη διαγραμμάτων καταστάσεων και τελικές διορθώσεις	0.8

## Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
X. Ζολώτας	*	<a href="mailto:christopherzoltas@issel.ee.auth.gr">christopherzoltas@issel.ee.auth.gr</a>
Αμπαρτσουμιάν Αλεξάνδρα	Auth on the road	<a href="mailto:alexampa@auth.gr">alexampa@auth.gr</a>
Δημαρίδου Βαλασία	Auth on the road	<a href="mailto:valia.dimaridou@gmail.com">valia.dimaridou@gmail.com</a>
Θεοδωρίδου Ιφιγένεια	Auth on the road	<a href="mailto:ifigtheo@auth.gr">ifigtheo@auth.gr</a>
Παπαδόπουλος Βασίλης	Auth on the road	<a href="mailto:vasipapa@auth.gr">vasipapa@auth.gr</a>

\* Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document. Original template is available at: <http://www.processimpact.com/>

## Πίνακας Περιεχομένων

Πίνακας Περιεχομένων .....	3
Λίστα Σχημάτων .....	4
1 Εισαγωγικά .....	5
1.1 Στόχος του Εγγράφου .....	5
1.2 Τυπογραφικές Παραδοχές του Εγγράφου .....	5
1.3 Αναγνωστικό κοινό και τρόπος ανάγνωσης .....	6
1.4 Σκοπός του Έργου .....	7
2 Στατική Μοντελοποίηση .....	8
2.1 Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας .....	8
2.1.1 Πακέτο UserHandle .....	8
2.1.2 Πακέτο ExcursionHandle .....	15
2.2 Πακέτα λεξιλογίου σεναρίων μέσης προτεραιότητας .....	25
2.2.1 Πακέτο EventHandle .....	25
2.2.2 Πακέτο ApplicationFormHandle .....	32
2.3 Πακέτα λεξιλογίου σεναρίων χαμηλής προτεραιότητας .....	40
2.3.1 Πακέτο AudioVisualHandle .....	40
2.3.2 Πακέτο MapHandle .....	48
3 Απαιτήσεις Λογισμικού .....	51
3.1 Δυναμική μοντελοποίηση του συστήματος .....	51
3.1.1 Πακέτο δημιουργίας εκδρομής από το χρήστη .....	51
3.1.2 Πακέτο δημιουργίας εκδήλωσης από το χρήστη .....	52
3.1.3 Πακέτο δημιουργίας τροποποίησης αίτησης συμμετοχής .....	53
3.1.4 Πακέτο αξιολόγησης αιτήσεων εκδρομής .....	54
3.1.5 Πακέτο προβολής οπτικοακουστικού υλικού .....	56
3.1.6 Διάγραμμα καταστάσεων χρήστη .....	57
3.1.7 Διάγραμμα καταστάσεων χρήστη δημιουργού .....	58
3.2 Μη λειτουργικές απαιτήσεις .....	59
3.2.1 Απαιτήσεις επίδοσης .....	59
3.2.2 Απαιτήσεις Χρηστικότητας (Usability) .....	59
3.2.3 Τεχνικές Απαιτήσεις περιβάλλοντος .....	59
Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας .....	61
Παράρτημα ΙΙ – Ανοιχτά Θέματα .....	62



## Λίστα Σχημάτων

Σχήμα 1. Διάγραμμα κλάσεων πακέτου UserHandle.....	14
Σχήμα 2. Διάγραμμα κλάσεων πακέτου ExcursionHandle.....	24
Σχήμα 3. Διάγραμμα κλάσεων πακέτου EventHandle.....	31
Σχήμα 4. Διάγραμμα κλάσεων πακέτου ApplicationFormHandle.....	39
Σχήμα 5. Διάγραμμα κλάσεων πακέτου AudioVisualHandle.....	47
Σχήμα 6. Διάγραμμα κλάσεων πακέτου MapHandle.....	50
Σχήμα 7. Σενάριο δημιουργίας εκδρομής.....	51
Σχήμα 8. Σενάριο δημιουργίας εκδήλωσης.....	52
Σχήμα 9. Σενάριο τροποποίησης αίτησης.....	54
Σχήμα 10. Σενάριο αξιολόγησης αιτήσεων.....	55
Σχήμα 11. Σενάριο προβολής οπτικοακουστικού υλικού.....	56
Σχήμα 12. Διάγραμμα καταστάσεων χρήστη.....	57
Σχήμα 13. Διάγραμμα καταστάσεων χρήστη δημιουργού .....	58



## 1 Εισαγωγικά

### 1.1 Στόχος του Εγγράφου

Το παρόν έγγραφο αποτελεί έναν οδηγό κατανόησης των προδιαγραφών του λογισμικού που συνθέτει το σύστημα. Είναι ένα εργαλείο για την αναγνώριση των χαρακτηριστικών του συστήματος και την πλήρη περιγραφή τους, λαμβάνοντας υπόψη και θέτοντας ως απαραίτητη προϋπόθεση την ικανοποίηση των απαιτήσεων χρηστών που περιγράφηκαν στο προηγούμενο έγγραφο.

Στόχος του εγγράφου είναι ο επακριβής προσδιορισμός της λειτουργίας του συστήματος. Περιγράφεται η συμπεριφορά του συστήματος σε σχέση με τους χρήστες του και τα συνεργαζόμενα συστήματα και παρέχονται λεπτομέρειες σχετικά με την δυναμικότητα του περιβάλλοντος διεπαφής.

Ένας ακόμη σημαντικός στόχος του εγγράφου είναι η αναλυτική περιγραφή των δεδομένων που διαχειρίζεται το σύστημα. Καθορίζονται οι τύποι δεδομένων που ανταλλάσσονται μεταξύ του συστήματος, των χρηστών και των συνεργαζόμενων συστημάτων. Επιπλέον παρέχονται πλήρεις περιγραφές των σελίδων, των φορμών και των μηνυμάτων που επιστρέφει το σύστημα, μέσω των οποίων ανταλλάσσονται δεδομένα.

Προς εξυπηρέτηση των στόχων αυτών είναι απαραίτητη η μοντελοποίηση του συστήματος, η οποία περιλαμβάνει την στατική και την δυναμική μοντελοποίηση.

Η στατική μοντελοποίηση αφορά την μοντελοποίηση των απαιτήσεων του χρήστη. Για ευκολότερη κατανόηση και ανάπτυξη του συστήματος, αυτό χωρίζεται σε πακέτα κλάσεων σύμφωνα με τις επιμέρους λειτουργικότητες του συστήματος. Γίνεται ιεράρχηση των πακέτων αυτών σύμφωνα με την σπουδαιότητα της λειτουργίας που περιγράφουν για την εφαρμογή, σε πακέτα υψηλής, μέσης και χαμηλής προτεραιότητας.

Κάθε πακέτο περιλαμβάνει ένα σύνολο κλάσεων. Οι κλάσεις κατηγοριοποιούνται σε κλάσεις οντοτήτων (entities), οι οποίες αναπαριστούν την μόνιμη πληροφορία του συστήματος, σε οριακές κλάσεις (boundaries) ή κλάσεις διεπαφής χρηστών, οι οποίες αναπαριστούν την αλληλεπίδραση ανάμεσα στον χρήστη και στο σύστημα και αναλαμβάνουν την επικοινωνία του συστήματος με εξωτερικά συστήματα και τέλος σε κλάσεις ελέγχου (controllers) ή ρυθμιστικές κλάσεις, οι οποίες είναι υπεύθυνες για τις εργασίες ελέγχου που εκτελούνται από το σύστημα και τον συντονισμό των διεργασιών των επιμέρους κλάσεων. Επιπλέον δημιουργούνται διαγράμματα που περιγράφουν τις αλληλεπιδράσεις των κλάσεων και των πακέτων.

Στη δυναμική μοντελοποίηση περιλαμβάνονται αφηγήσεις σεναρίων χρήσης, οι οποίες παρουσιάζονται μέσω των ακολουθιακών διαγραμμάτων. Τα σενάρια χρήσης που δημιουργήθηκαν στο προηγούμενο έγγραφο απαιτήσεων χρηστών μετασχηματίζονται σε σενάρια με απεικόνιση κλάσεων και δημιουργούνται επιπλέον διαγράμματα καταστάσεων που περιγράφουν τις δυνατές ενέργειες των χρηστών.

Το παρόν έγγραφο θα αποτελέσει τον οδηγό για την ανάπτυξη του λογισμικού από την ομάδα ανάπτυξης.

### 1.2 Τυπογραφικές Παραδοχές του Εγγράφου

Το κείμενο του εγγράφου τούτου είναι γραμμένο με γραμματοσειρά Callibri, μεγέθους 11pt. Χρησιμοποιείται υπογραμμισμένη πλάγια γραφή για να δειχθεί η επικεφαλίδα ή το χαρακτηριστικό το οποίο αναλύεται από ένα κείμενο.

### 1.3 Αναγνωστικό κοινό και τρόπος ανάγνωσης

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

- Ομάδα ανάπτυξης: Κάθε μέλος της ομάδας ανάπτυξης για να είναι σε θέση να συμβάλει στο σχεδιασμό, την ανάλυση και τον προγραμματισμό του συστήματος πρέπει να έχει κατανοήσει πλήρως τις απαιτήσεις λογισμικού της εφαρμογής. Ανάλογα με την δραστηριότητα ανάπτυξης τα μέλη μπορούν να έχουν διαφορετικούς ρόλους. Επομένως η ομάδα ανάπτυξης μπορεί να χρησιμοποιεί διαφορετικούς τομείς του παρόντος εγγράφου είτε να χρησιμοποιεί διαφορετικές όψεις των ίδιων διαγραμμάτων. Οι ρόλοι σε μία ομάδα ανάπτυξης είναι οι εξής:
  - ✓ Αναλυτής: Ενδιαφέρεται για τις συσχετίσεις ανάμεσα στις κλάσεις στο πεδίο εφαρμογής, και για τις βασικές λειτουργίες των κλάσεων, χωρίς να ενδιαφέρεται για κλάσεις που επιλύουν συγκεκριμένα προβλήματα. Επίσης εκμεταλλεύεται την κληρονομικότητα για να ιεραρχήσει τις κλάσεις. Προς επίτευξη των στόχων αυτών, απαραίτητη προϋπόθεση κρίνεται η μελέτη του δεύτερου και τρίτου κεφαλαίου του παρόντος εγγράφου, δίνοντας μεγαλύτερη έμφαση στα διαγράμματα κλάσεων και τη συσχέτιση των πακέτων.
  - ✓ Σχεδιαστής συστήματος: Ο Σχεδιαστής συστήματος επικεντρώνεται στη λύση του προβλήματος. Η σχεδίαση αποτελείται από πολλές εργασίες, κύρια εργασία είναι ο καθορισμός των διεπαφών. Ο σχεδιαστής περιγράφει τις διεπαφές των κλάσεων και τις διεπαφές των υποσυστημάτων. Προς επίτευξη των στόχων αυτών, απαραίτητη προϋπόθεση κρίνεται η μελέτη του δεύτερου και τρίτου κεφαλαίου του παρόντος εγγράφου, δίνοντας μεγαλύτερη έμφαση στα πακέτα κλάσεων, τη συσχέτιση των πακέτων και στις οριακές κλάσεις (boundaries) ή κλάσεις διεπαφής χρηστών.
  - ✓ Σχεδιαστής αντικειμένων: Ο Σχεδιαστής αντικειμένων επικεντρώνεται στη δημιουργία των μόνιμων οντοτήτων του συστήματος. Απαραίτητη κρίνεται η μελέτη του δεύτερου κεφαλαίου του παρόντος εγγράφου δίνοντας μεγαλύτερη έμφαση στις κλάσεις οντοτήτων (entities).
  - ✓ Προγραμματιστής: Για τα μέλη της ομάδας αυτής κρίνεται απαραίτητη η μελέτη του δεύτερου και τρίτου κεφαλαίου του παρόντος εγγράφου.
- Ομάδα συντήρησης: Η ομάδα συντήρησης και θα αναβάθμισης του συστήματος για να είναι σε θέση να παρέχει βελτιώσεις σε αυτό θα πρέπει να έχει κατανοήσει πρώτα πλήρως την λειτουργικότητα του και τις βασικές οντότητες που δημιουργήθηκαν για τον προγραμματισμό του. Συνεπώς κρίνεται πολύ σημαντική η μελέτη του δεύτερου και τρίτου κεφαλαίου του παρόντος εγγράφου.

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

- ❖ Κεφάλαιο 1<sup>ο</sup>: Αντιπροσωπεύει το εισαγωγικό τμήμα του εγγράφου και περιέχει μια επισκόπηση του στόχου του, τις τυπικές παραδοχές, την ανάλυση του αναγνωστικού κοινού, καθώς και του τρόπου ανάγνωσης του εγγράφου. Επιπρόσθετα, περιέχει ανάλυση του σκοπού του έργου της εφαρμογής.
- ❖ Κεφάλαιο 2<sup>ο</sup>: Πραγματοποιείται η καταγραφή των πακέτων των σεναρίων υψηλής, μέσης και χαμηλής προτεραιότητας. Για κάθε πακέτο γίνεται λεπτομερή ανάλυση των κλάσεων και των συναρτήσεων που περιέχουν και παρουσιάζεται διάγραμμα συσχέτισης των κλάσεων και των πακέτων.
- ❖ Κεφάλαιο 3<sup>ο</sup>: Περιλαμβάνει την δυναμική μοντελοποίηση του συστήματος, που παρουσιάζεται μέσω πακέτων αφήγησης και διαγραμμάτων σεναρίων με απεικόνιση κλάσεων. Στη συνέχεια



περιέχει διαγράμματα καταστάσεων για τους χρήστες του συστήματος. Τέλος παρουσιάζει μία ταξινόμηση των μη λειτουργικών απαιτήσεων.

Παράρτημα Ι : Περιέχει πίνακα ιχνηλασιμότητας, ο οποίος αντιστοιχίζει τα πακέτα των σεναρίων του κεφαλαίου 2 με τις λειτουργικές απαιτήσεις και τα σενάρια χρήσης του εγγράφου των απαιτήσεων χρηστών.

Παράρτημα ΙΙ : Περιέχει μια δυναμική λίστα με ανοιχτά θέματα που αφορούν το λογισμικό.

## 1.4 Σκοπός του Έργου

Η εφαρμογή αφορά την δημιουργία εκπαιδευτικών εκδρομών από μέλη της πανεπιστημιακής κοινότητας, τα οποία επιθυμούν να οργανώσουν εκδρομές με εκπαιδευτικό σκοπό, τόσο καθηγητές όσο και φοιτητές, βοηθώντας στην ενασχόληση των φοιτητών με πραγματικά προβλήματα στους χώρους εργασίας με σκοπό την αύξηση της πρακτικής τους εξάσκησης. Έχει ως σκοπό να δώσει τη δυνατότητα στους φοιτητές να ενημερωθούν για ερευνητικά προγράμματα στο πλαίσιο των εκδρομών και να συμμετέχουν σε αυτά.

Όλα αυτά είναι αναγκαία καθώς παρατηρείται η απουσία έγκαιρης ενημέρωσης των φοιτητών σχετικά με τις εκπαιδευτικές εκδρομές που διοργανώνονται στο πλαίσιο του Πανεπιστημίου και περιορισμένη συμμετοχή των φοιτητών σε αυτές. Ως αποτέλεσμα δεν υπάρχει κίνητρο για διοργάνωση περισσότερων. Επιπλέον παρατηρείται έλλειψη πρακτικής σκέψης των φοιτητών και δυσκολία προσαρμογής στο εργασιακό περιβάλλον.

Οι φοιτητές είναι πλήρως εξοικειωμένοι με την τεχνολογία και επιθυμούν να ενημερώνονται άμεσα για όλα τα θέματα που τους ενδιαφέρουν μέσω εφαρμογών τόσο στα κινητά τηλέφωνα όσο και στους υπολογιστές. Η εφαρμογή αυτή έχει ως στόχο να είναι συμβατή και από τις δύο κατηγορίες συσκευών.

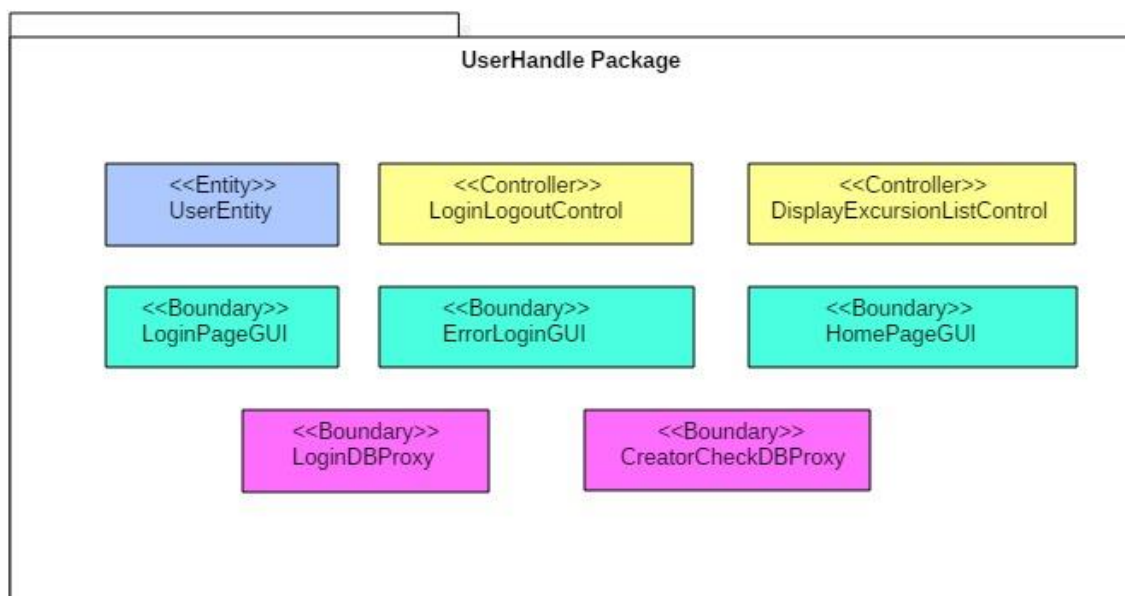
Επομένως μέσω της εφαρμογής, η ομάδα ανάπτυξης επιθυμεί την δημιουργία μίας ελκυστικής και εύχρηστης εφαρμογής με στόχο την πλήρη ενημέρωση των φοιτητών για όλες τις εκπαιδευτικές εκδρομές που πραγματοποιούνται στο πλαίσιο του Πανεπιστημίου, αλλά και την διευκόλυνση των ενδιαφερόμενων σε όλα τα στάδια μέχρι την πραγματοποίηση της, καθώς και την παρουσίαση οπτικοακουστικού υλικού μετά την πραγματοποίησή της.

## 2 Στατική Μοντελοποίηση

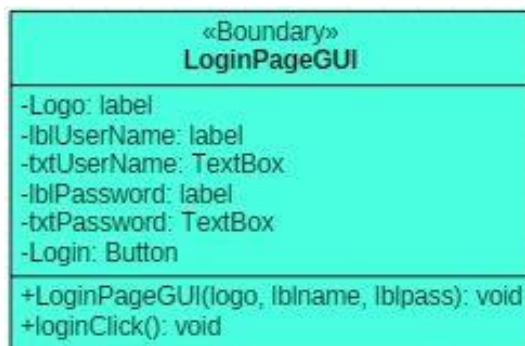
### 2.1 Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας

#### 2.1.1 Πακέτο UserHandle

Το πακέτο αυτό αναφέρεται στις επιλογές login και logout του χρήστη. Επιπροσθέτως, περιλαμβάνει την αρχική οθόνη που βλέπει ο χρήστης κατόπιν της συνδέσεώς του στο σύστημα. Παρακάτω παρουσιάζονται οι κλάσεις που διαχειρίζονται τις παραπάνω επιλογές.



#### Boundary LoginPageGUI



Η κλάση αυτή εκφράζει τη διεπαφή της σελίδας σύνδεσης του χρήστη στο σύστημα της εφαρμογής.



Χαρακτηριστικά της κλάσης:

- ❖ Logo: Λογότυπο της εφαρμογής .
- ❖ lblUserName: Επιγραφή πάνω από το πεδίο συμπλήρωσης ονόματος χρήστη.
- ❖ txtUserName: Πεδίο συμπλήρωσης για το όνομα του πανεπιστημιακού λογαριασμού του χρήστη.
- ❖ lblPassword: Επιγραφή πάνω από το πεδίο συμπλήρωσης κωδικού πρόσβασης του χρήστη.
- ❖ txtPasssword: Πεδίο συμπλήρωσης για τον κωδικό πρόσβασης.
- ❖ Login: Κουμπί για την είσοδο στο σύστημα της εφαρμογής.

Μέθοδοι της κλάσης:

- ❖ LoginPageGUI(logo, lblname, lblpass): Η συνάρτηση αυτή είναι ένας δομητής της κλάσης LoginPageGUI.
- ❖ loginClick(): Η μέθοδος αυτή καλεί τη συνάρτηση checkdata() του ελεγκτή LoginLogoutControl προς έλεγχο των στοιχείων του χρήστη.

Boundary ErrorLoginGUI

«Boundary» <b>ErrorLoginGUI</b>
-Error: string -Back: button
+display(error: string): void +backClick(): void

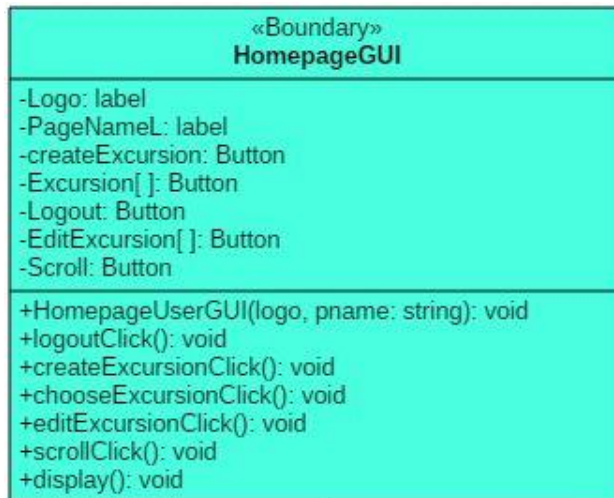
Πρόκειται για τη διεπαφή η οποία είναι υπεύθυνη για την επιστροφή των σφαλμάτων σύνδεσης του χρήστη.

Χαρακτηριστικά της κλάσης:

- ❖ Error: Το μήνυμα σφάλματος που προβάλλεται στο χρήστη.
- ❖ Back: Κουμπί για την επιστροφή στη σελίδα σύνδεσης.

Μέθοδοι της κλάσης:

- ❖ display(error: string): Η μέθοδος αυτή είναι υπεύθυνη για την προβολή του μηνύματος σφάλματος στο χρήστη. Το μήνυμα σφάλματος παρέχεται από τη συνάρτηση returnUrl του ελεγκτή LoginLogoutControl.
- ❖ backClick(): Η μέθοδος καλεί τη συνάρτηση back() του ελεγκτή LoginLogoutControl για επιστροφή στη σελίδα σύνδεσης.

**Boundary HomepageGUI**

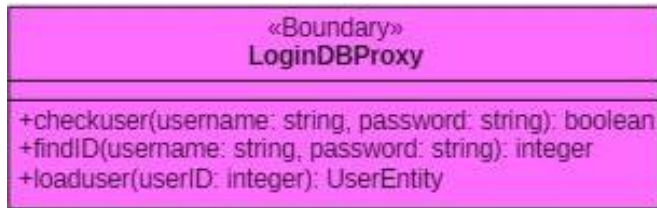
Η κλάση αυτή εκφράζει τη διεπαφή της αρχικής σελίδας του συνδεδεμένου χρήστη. Ξεκινώντας από αυτή μπορούν να πυροδοτηθούν όλες οι λειτουργίες της εφαρμογής.

**Χαρακτηριστικά της κλάσης:**

- ❖ Logo: Λογότυπο της εφαρμογής.
- ❖ PageNameL: Ονομασία της αρχικής σελίδας.
- ❖ createExcursion: Κουμπί για τη δημιουργία νέας εκδρομής.
- ❖ Excursion[]: Λίστα συνδέσμων για την προβολή οποιασδήποτε εκδρομής.
- ❖ Logout: Σύνδεσμος αποσύνδεσης του χρήστη.
- ❖ EditExcursion[]: Λίστα συνδέσμων για την τροποποίηση οποιασδήποτε εκδρομής.
- ❖ Scroll: Ολισθαίνον κουμπί για την προβολή των υπόλοιπων εκδρομών στη λίστα υπαρχουσών εκδρομών.

**Μέθοδοι της κλάσης:**

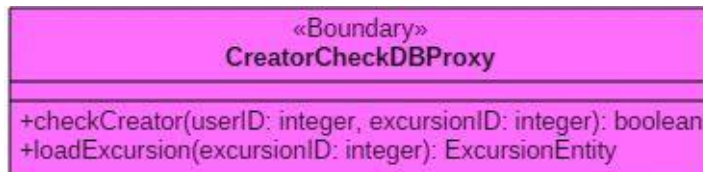
- ❖ HomePageUserGUI(logo,pname:string): Η μέθοδος αυτή είναι η συνάρτηση δόμησης της διεπαφής.
- ❖ logoutClick(): Καλεί τη συνάρτηση logout() του ελεγκτή LoginLogoutControl ο οποίος θα αποσυνδέσει το χρήστη.
- ❖ createExcursionClick(): Καλεί τον ελεγκτή CreateEditExcursionControl ο οποίος είναι υπεύθυνος για την δημιουργία εκδρομής.
- ❖ chooseExcursionClick(): Καλεί τη συνάρτηση loadExcursion() του ελεγκτή CreateEditExcursionControl προς προβολή της σελίδας μιας εκδρομής.
- ❖ editExcursionClick(): Καλεί τη συνάρτηση editExcursion() του ελεγκτή CreateEditExcursionControl, ο οποίος είναι υπεύθυνος και για την τροποποίηση μιας εκδρομής.
- ❖ scrollClick(): Η μέθοδος αυτή πραγματοποιεί ολίσθηση στη λίστα με τις εκδρομές.
- ❖ display(): Η μέθοδος αυτή καλείται από όσους ελεγκτές επιχειρούν να προβάλουν την αρχική σελίδα της εφαρμογής.

**Boundary LoginDBProxy**

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με τη βάση δεδομένων όπου είναι αποθηκευμένα τα στοιχεία των χρηστών.

**Μέθοδοι της κλάσης:**

- ❖ checkuser(username: string, password: string): Η μέθοδος αυτή ελέγχει αν το στοιχείο που έχει δώσει ο χρήστης είναι ορθό. Σε περίπτωση ορθότητας επιστρέφεται η λογική τιμή true, ειδάλλως επιστρέφεται η λογική τιμή false.
- ❖ findID(username: string, password: string): Η μέθοδος αυτή επιστρέφει το μοναδικό αναγνωριστικό του χρήστη, λαμβάνοντας τα στοιχεία του.
- ❖ loaduser(userID: integer): Η μέθοδος αυτή επιστρέφει ένα χρήστη από τη βάση δεδομένων, βάσει του μοναδικού αναγνωριστικού του.

**Boundary CheckCreatorDBProxy**

Η κλάση αυτή επικοινωνεί με τη βάση δεδομένων και χρησιμοποιείται προκειμένου να ελεγχθεί εάν ο συνδεδεμένος χρήστης είναι ο χρήστης-δημιουργός μιας εκδρομής.

**Μέθοδοι της κλάσης:**

- ❖ checkCreator(userID: integer, excursionID: integer): Η μέθοδος αυτή ελέγχει αν ο χρήστης με το συγκεκριμένο αναγνωριστικό είναι ο δημιουργός της εκδρομής που φέρει το συγκεκριμένο αναγνωριστικό. Σε περίπτωση ορθότητας επιστρέφεται η λογική τιμή true, ειδάλλως επιστρέφεται η λογική τιμή false.
- ❖ loadExcursion(excursionID: integer): Η μέθοδος αυτή χρησιμοποιείται στο σώμα της check-Creator() για να είναι δυνατή η εύρεση του χρήστη-δημιουργού της κάθε εκδρομής.

### Entity UserEntity

«Entity» UserEntity
-userID: integer -username: string -password: string
+UserEntity(us, pass): void +setUserID(userID: integer): void +setUsername(username: string): void +setPassword(pass: string): void +getUsername(): string +getPassword(): string +getUserID(): integer

Αυτή είναι η κλάση οντότητας του χρήστη και περιέχει τα στοιχεία για τον ορισμό του.

#### Χαρακτηριστικά της κλάσης:

- ❖ userID: Ο μοναδικός χαρακτηριστικός αριθμός για τον κάθε χρήστη.
- ❖ username: Το όνομα του πανεπιστημιακού του λογαριασμού.
- ❖ password: Ο κωδικός πρόσβασης του πανεπιστημιακού του λογαριασμού.

#### Μέθοδοι της κλάσης:

- ❖ UserEntity(us,pass): Η συνάρτηση αυτή είναι ένας δομητής της οντότητας UserEntity.
- ❖ setUserID(userID: integer): Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο μοναδικό χαρακτηριστικό αριθμό χρήστη του αντικειμένου.
- ❖ setUsername(username: string): Η συνάρτηση αυτή δίνει στο όνομα χρήστη τη συμβολοσειρά που λαμβάνει ως όρισμα.
- ❖ setPassword(pass: string): Η συνάρτηση αυτή θέτει τον κωδικό του χρήστη σύμφωνα με το όρισμα που δέχεται.
- ❖ getUsername(): Η συνάρτηση αυτή επιστρέφει το όνομα του πανεπιστημιακού λογαριασμού του χρήστη.
- ❖ getPassword(): Η συγκεκριμένη συνάρτηση επιστρέφει τον κωδικό του πανεπιστημιακού λογαριασμού του χρήστη.
- ❖ getUserID(): Η συνάρτηση αυτή επιστρέφει το μοναδικό χαρακτηριστικό αριθμό για κάθε χρήστη.

### Controller LoginLogoutControl

«Controller» LoginLogoutControl
-userID: integer
+LoginLogoutControl(userID: integer): void +login(dataflag: boolean): void +logout(): void +checkdata(username: string, password: string): boolean +returnError(dataflag: boolean): string +displayHomepage(): void +back(): void



Ο ελεγκτής αυτός περιέχει όλες τις κατάλληλες συναρτήσεις για την είσοδο του χρήστη στο σύστημα.

Χαρακτηριστικά της κλάσης:

- ❖ userID: Ο μοναδικός χαρακτηριστικός αριθμός για τον κάθε χρήστη.

Μέθοδοι της κλάσης:

- ❖ LoginLogoutControl(userID: integer): Πρόκειται για μέθοδο δόμησης του ελεγκτή.
- ❖ login(dataflag: boolean): Η μέθοδος αυτή συνδέει το χρήστη στο σύστημα εφόσον η boolean μεταβλητή dataflag είναι αληθής.
- ❖ logout(): Η μέθοδος αυτή αποσυνδέει το χρήστη από το σύστημα.
- ❖ checkdata(username:string,password:string): Η μέθοδος αυτή ελέγχει εάν τα στοιχεία που έδωσε ο χρήστης είναι ορθά. Σε περίπτωση ορθότητας καλείται η συνάρτηση login(), ειδικά καλείται η returnError().
- ❖ returnError(dataflag:boolean): Η συνάρτηση αυτή επιστρέφει μήνυμα σφάλματος εύρεσης των στοιχείων του χρήστη στη συνάρτηση displayError() της οριακής κλάσης ErrorLoginGUI. Η συνάρτηση αυτή καλείται εφόσον η μεταβλητή dataflag είναι ψευδής.
- ❖ displayHomepage(): Η μέθοδος αυτή καλείται για την αρχική σελίδα της εφαρμογής.
- ❖ back(): Μέθοδος για την εξυπηρέτηση του κουμπιού "πίσω".

Controller DisplayExcursionListControl

«Controller» DisplayExcursionListControl	
+List: ExcursionEntity	
+displayExcursionListControl(): void +displayEditButton(userID: integer, excursionID: integer): void +displayHomepage(): void	

Ο ελεγκτής αυτός είναι υπεύθυνος για τη διαχείριση της αρχικής σελίδας, ενημερώνοντας τη λίστα εκδρομών και συνδέσμων τροποποίησης που πρέπει να εμφανίζονται σε αυτήν.

Χαρακτηριστικά της κλάσης:

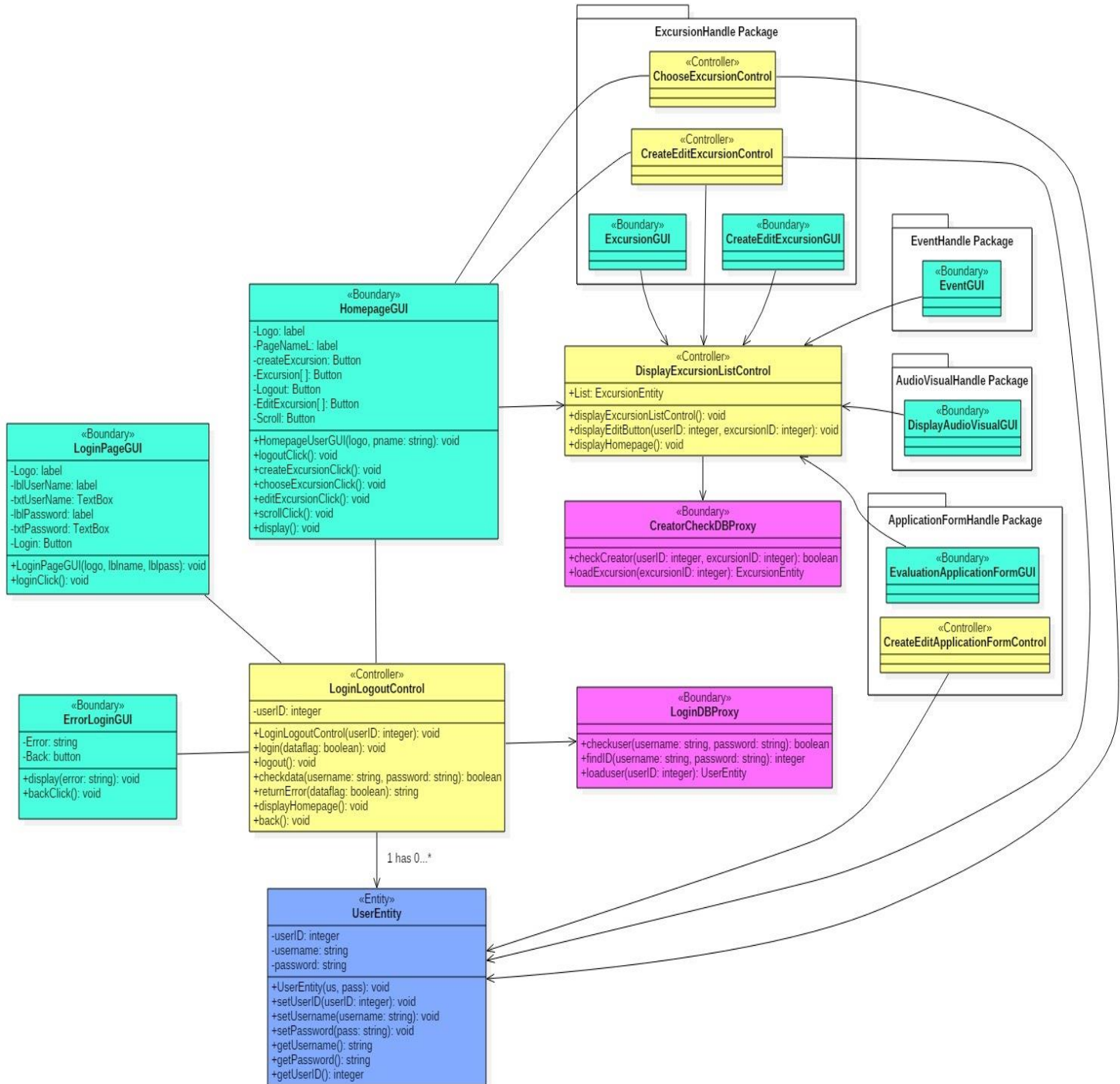
- ❖ List: Λίστα με αντικείμενα τύπου ExcursionEntity.

Μέθοδοι της κλάσης:

- ❖ displayExcursionListControl(): Αποτελεί μέθοδο δόμησης του ελεγκτή.
- ❖ displayEditButton(userID: integer, excursionID: integer): Η μέθοδος αυτή είναι υπεύθυνη για την εμφάνιση των συνδέσμων τροποποίησης στις εκδρομές που έχει δημιουργήσει ο ίδιος χρήστης.
- ❖ displayHomepage(): Η μέθοδος αυτή εμφανίζει την αρχική σελίδα, καλώντας την οριακή κλάση HomepageGUI.

**ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ:**

Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.

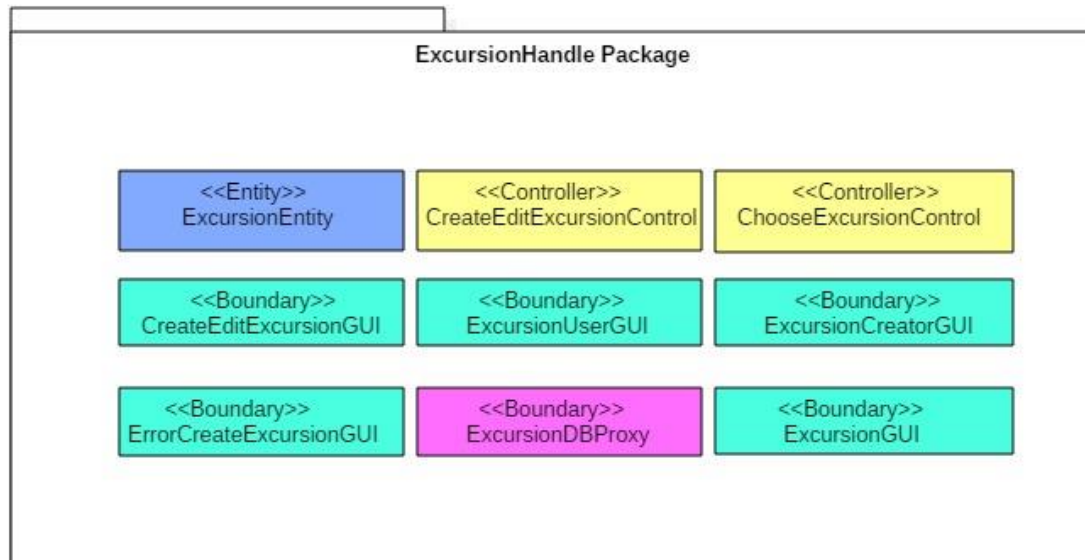


Σχήμα 1. Διάγραμμα κλάσεων πακέτου UserHandle

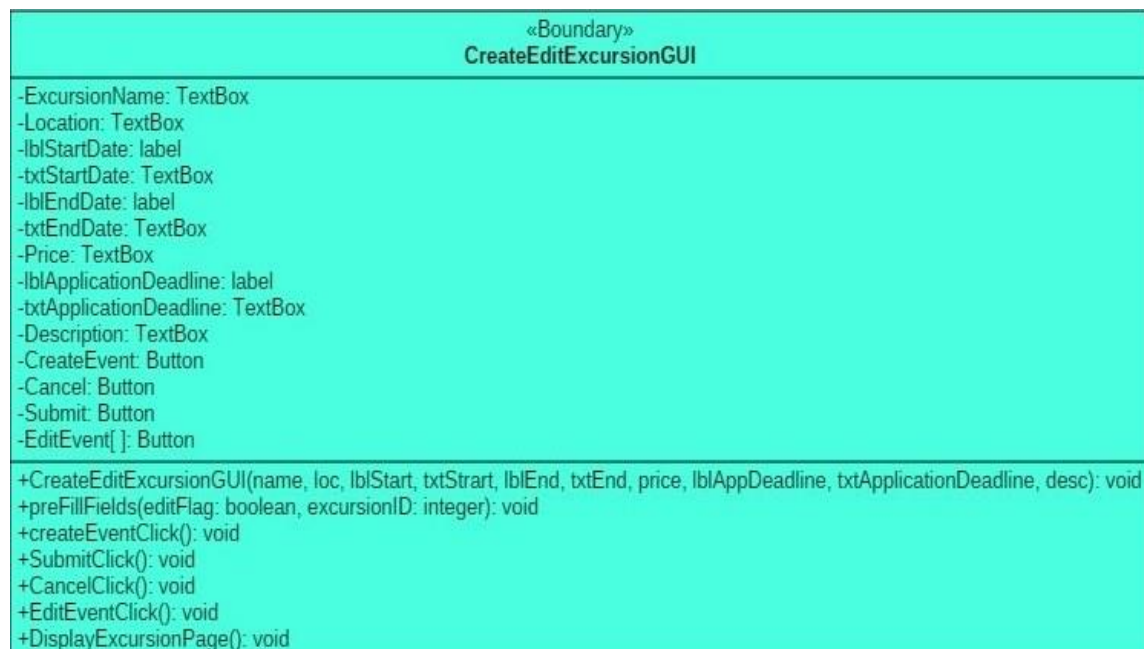


### 2.1.2 Πακέτο ExcursionHandle

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που σχετίζονται με τη διαχείριση των εκδρομών, συμπεριλαμβανομένου της δημιουργίας τους, της τροποποίησής τους και της προβολής τους στο χρήστη, ανάλογα με τα δικαιώματα που ο ίδιος έχει στην καθεμία.



#### Boundary CreateEditExcursionGUI



Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση της σελίδας δημιουργίας ή τροποποίησης μιας εκδρομής.

#### Χαρακτηριστικά της κλάσης:

- ❖ ExcursionName: Πεδίο εισαγωγής του ονόματος της εκδρομής.



- ❖ Location: Πεδίο εισαγωγής της τοποθεσίας της εκδρομής.
- ❖ lblStartDate: Επιγραφή για την ημερομηνία έναρξης.
- ❖ txtStartDate: Πεδίο για την εισαγωγή της ημερομηνίας έναρξης της εκδρομής.
- ❖ lblEnddate: Επιγραφή για την ημερομηνία λήξης της εκδρομής.
- ❖ txtEndDate: Πεδίο για την εισαγωγή της ημερομηνίας λήξης της εκδρομής.
- ❖ Price: Πεδίο για την εισαγωγή του ενδεικτικού κόστους της εκδρομής.
- ❖ lblApplicationDeadline: Επιγραφή για τη λήξη ημερομηνίας δηλώσεων συμμετοχής.
- ❖ txtApplicationDeadline: Πεδίο για την εισαγωγή ημερομηνίας λήξης δηλώσεων συμμετοχής.
- ❖ Description: Πεδίο για την εισαγωγή της γενικής περιγραφής της εκδρομής.
- ❖ CreateEvent: Κουμπί για τη δημιουργία εκδήλωσης για την παρούσα εκδρομή.
- ❖ Cancel: Κουμπί για την ακύρωση δημιουργίας εκδρομής.
- ❖ Submit: Κουμπί για την τελική υποβολή της εκδρομής.
- ❖ EditEvent[]: Λίστα από κουμπιά για την τροποποίηση των εκδηλώσεων που έχουν δημιουργηθεί για την παρούσα εκδρομή. Να σημειωθεί πως η διεπαφή αυτή εμφανίζεται και στην περίπτωση τροποποίησης της εκδρομής.

#### Μέθοδοι της κλάσης:

- ❖ CreateEditExcursion-  
GUI(name,loc,lblStart,txtStart,lblEnd,txtEnd,price,lblAppDeadline,txtApplicationDeadline,desc): Η μέθοδος αυτή αποτελεί την συνάρτηση δόμησης της διεπαφής αυτής.
- ❖ preFillFields(editFlag: boolean, excursionID: integer): Η μέθοδος αυτή σε περίπτωση που η μεταβλητή editFlag έχει την τιμή false, συμπληρώνει τα πεδία με τις αρχικές τιμές τους. Αντίθετα, σε περίπτωση που η μεταβλητή editFlag έχει την τιμή true, σημαίνει ότι η εκδρομή έχει ήδη δημιουργηθεί και άρα πρέπει τα πεδία να προσυμπληρωθούν με τις τιμές της εκδρομής που φέρει το συγκεκριμένο αναγνωριστικό (excursionID).
- ❖ createEventClick(): Η μέθοδος αυτή μεταβαίνει στη σελίδα δημιουργίας εκδήλωσης για την εκδρομή μέσω του ελεγκτή CreateEventControl.
- ❖ SubmitClick(): Καλεί τη συνάρτηση storeExcursion() του ελεγκτή CreateEditExcursionControl και αποθηκεύει την εκδρομή μέσω της ExcursionDBProxy.
- ❖ CancelClick(): Η μέθοδος αυτή ακυρώνει την διαδικασία δημιουργίας εκδρομής και επιστρέφει στην αρχική σελίδα, μέσω της displayHomepage() του ελεγκτή DisplayExcursion-ListControl.
- ❖ EditEventClick(): Καλεί τη συνάρτηση editEvent() του ελεγκτή CreateEventControl για να φορτώσει τα δεδομένα μιας εκδήλωσης από την EventDBProxy, προκειμένου να τα επεξεργαστεί.
- ❖ DisplayExcursionPage(): Η κλήση της συνάρτησης αυτής χρησιμοποιείται για την εμφάνιση της εν λόγω διεπαφής.



### Boundary ErrorCreateExcursionGUI

«Boundary» ErrorCreateExcursionGUI
-Error: string -Back: button
+display(error: string): void +backClick(): void

Πρόκειται για τη διεπαφή η οποία είναι υπεύθυνη για την επιστροφή των σφαλμάτων δημιουργίας μιας εκδρομής.

#### Χαρακτηριστικά της κλάσης:

- ❖ Error: Το μήνυμα σφάλματος που προβάλλεται στο χρήστη.
- ❖ Back: Κουμπί για την επιστροφή στη σελίδα δημιουργίας εκδρομής.

#### Μέθοδοι της κλάσης:

- ❖ display(error: string): Η μέθοδος αυτή είναι υπεύθυνη για την προβολή του μηνύματος σφάλματος στο χρήστη. Το μήνυμα σφάλματος παρέχεται από τη συνάρτηση returnError του ελεγκτή CreateEditExcursionControl.
- ❖ backClick(): Η μέθοδος καλεί τη συνάρτηση back() του ελεγκτή CreateEditExcursionControl για επιστροφή στη σελίδα δημιουργίας εκδρομής.

### Boundary ExcursionGUI

«Boundary» ExcursionGUI
-Excursion: label -Name: label -Date: label -StartDate: label -EndDate: label -lblLocation: label -strLocation: label -lblPrice: label -strPrice: label -lblDescription: label -strDescription: label -AudioVisual[ ]: Button -UploadAudioVisual: Button -Events[]: Button -ApplicationDeadline: label -Home: Button -Scroll: Button
+ExcursionGUI(excur, date, lblloc, lblprice, lbldescr, excursion: ExcursionEntity): void +AudioVisualClick(): void +UploadAudioVisualClick(): void +ChooseEventsClick(): void +HomeClick(): void +ScrollClick(): void +display(): void

Αυτή η κλάση είναι υπεύθυνη για την εμφάνιση της διεπαφής της προβολής εκδρομής στο χρήστη της εφαρμογής. Όσον αφορά μία εκδρομή, οι ιδιότητες του απλού χρήστη και του χρήστη-δημιουργού της διαφοροποιούνται.



Λόγω του γεγονότος αυτού, έχουν δημιουργηθεί δύο ξεχωριστά περιβάλλοντα διεπαφής για την προβολή εκδρομής από τον απλό χρήστη και από το χρήστη-δημιουργό, τα οποία κληρονομούν την κλάση ExcursionGUI.

Χαρακτηριστικά της κλάσης:

- ❖ Excursion: Επιγραφή εμφάνισης της ετικέτας «Εκδρομή».
- ❖ Name: String για την εμφάνιση του ονόματος της εκδρομής.
- ❖ Date: Επιγραφή για την εμφάνιση της ετικέτας «Ημερομηνία».
- ❖ StartDate: String για την εμφάνιση της ημερομηνίας έναρξης της εκδρομής.
- ❖ EndDate: String για την εμφάνιση της ημερομηνίας λήξης της εκδρομής.
- ❖ lblLocation: Επιγραφή για την εμφάνιση της ετικέτας «Τοποθεσία».
- ❖ strLocation: String για την εμφάνιση της τοποθεσίας της εκδρομής.
- ❖ lblPrice: Επιγραφή για την εμφάνιση της ετικέτας «Ενδεικτικό κόστος».
- ❖ strPrice: String για την εμφάνιση του ενδεικτικού κόστους της εκδρομής.
- ❖ lblDescription: Επιγραφή για την εμφάνιση της ετικέτας «Περιγραφή».
- ❖ strDescription: String για την εμφάνιση της περιγραφής της εκδρομής.
- ❖ AudioVisual[]: Λίστα από κουμπιά για την προβολή των βίντεο και εικόνων της εκδρομής.
- ❖ UploadAudioVisual: Κουμπί για το ανέβασμα οπτικοακουστικού υλικού.
- ❖ Events[]: Λίστα κουμπιών για την εμφάνιση των εκδηλώσεων της εκδρομής.
- ❖ ApplicationDeadline: String για την εμφάνιση της ημερομηνίας λήξης υποβολής δηλώσεων συμμετοχής.
- ❖ Home: Κουμπί για την επιστροφή στην αρχική σελίδα.
- ❖ Scroll: Ολισθαίνον κουμπί για την προβολή των υπόλοιπων εικόνων ή βίντεο στη λίστα οπτικοακουστικού υλικού μιας εκδρομής.

Μέθοδοι της κλάσης:

- ❖ ExcursionUserGUI(excur,date,lbllloc,lbllprice,lblldescr,excursion: ExcursionEntity): Πρόκειται για συνάρτηση δόμησης της οριακής κλάσης.
- ❖ AudioVisualClick(): Η μέθοδος αυτή καλεί τη συνάρτηση loadExcursion() του ελεγκτή DisplayAudioVisualListControl για να προβάλει την αντίστοιχη εικόνα ή βίντεο.
- ❖ UploadAudioVisualClick(): Η μέθοδος αυτή καλεί τον ελεγκτή AudioVisualUploadControl για το ανέβασμα ενός αρχείου οπτικοακουστικού υλικού στην AudioVisualDBProxy.
- ❖ ChooseEventsClick(): Καλεί τη συνάρτηση loadEvent() του ελεγκτή CreateEventControl για προβολή της σελίδας μιας εκδήλωσης με χρήση της οριακής κλάσης EventGUI.
- ❖ HomeClick(): Καλεί τη συνάρτηση displayHomepage() του ελεγκτή DisplayExcursionListControl για προβολή της αρχικής σελίδας της εφαρμογής.
- ❖ ScrollClick(): Η συνάρτηση αυτή θέτει σε λειτουργία τον ολισθαίνοντα σύνδεσμο προβολής της λίστας οπτικοακουστικού υλικού.
- ❖ display(): Η συνάρτηση αυτή είναι υπεύθυνη για την εμφάνιση της σελίδας μιας εκδρομής και καλείται κάθε φορά από όσους ελεγκτές επιχειρούν να την προβάλουν.

### Boundary ExcursionUserGUI

«Boundary» <b>ExcursionUserGUI</b>
-CreateEditApplicationForm: Button
+ExcursionUserGUI(): void +CreateEditApplicationFormClick(): void

Πρόκειται για την κλάση που είναι υπεύθυνη για την εμφάνιση της διεπαφής της προβολής εκδρομής στον απλό χρήστη, η οποία κληρονομεί την κλάση ExcursionGUI.

#### Χαρακτηριστικά της κλάσης:

- ❖ CreateEditApplicationForm: Κουμπί για τη δημιουργία ή επεξεργασία της δήλωσης συμμετοχής.

#### Μέθοδοι της κλάσης:

- ❖ ExcursionUserGUI(): Μέθοδος δόμησης της οριακής κλάσης.
- ❖ CreateEditApplicationFormClick(): Η συνάρτηση αυτή καλεί την editApplicationForm() του ελεγκτή CreateEditApplicationFormControl, για δημιουργία ή τροποποίηση μιας αίτησης συμμετοχής.

### Boundary ExcursionCreatorGUI

«Boundary» <b>ExcursionCreatorGUI</b>
-EvaluateApplications: Button
+ExcursionCreatorGUI(): void +EvaluateApplicationsClick(): void

Πρόκειται για την κλάση που είναι υπεύθυνη για την εμφάνιση της διεπαφής της προβολής εκδρομής στο χρήστη-δημιουργό της. Η κλάση αυτή κληρονομεί την κλάση ExcursionGUI.

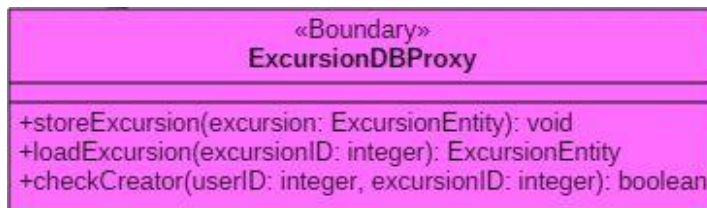
#### Χαρακτηριστικά της κλάσης:

- ❖ EvaluateApplications: Κουμπί για την αξιολόγηση των δηλώσεων συμμετοχής.

#### Μέθοδοι της κλάσης:

- ❖ ExcursionCreatorGUI(): Μέθοδος δόμησης της οριακής κλάσης.
- ❖ EvaluateApplicationsClick(): Η συνάρτηση αυτή καλεί τον ελεγκτή EvaluateApplicationFormControl για την προβολή των αιτήσεων συμμετοχής που έχουν υποβληθεί.

### Boundary ExcursionDBProxy

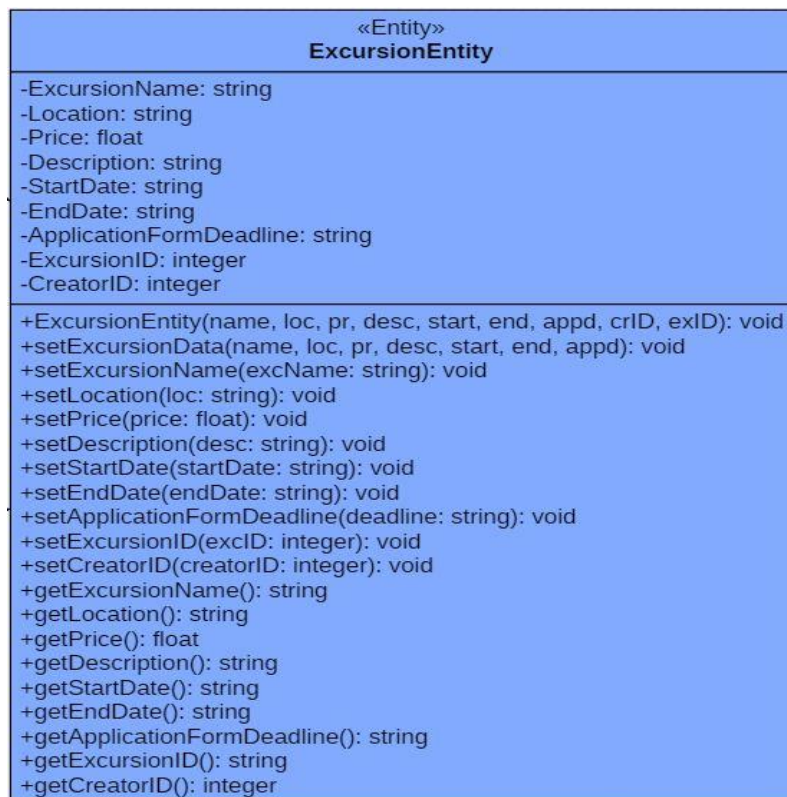


Η κλάση αυτή χρησιμοποιείται για επικοινωνία με τη βάση δεδομένων όπου είναι αποθηκευμένα τα στοιχεία των εκδρομών.

#### Μέθοδοι της κλάσης:

- ❖ storeExcursion(excursion: ExcursionEntity): Η μέθοδος αυτή αποθηκεύει μια εκδρομή, δηλαδή κλάση ExcursionEntity, στη βάση δεδομένων.
- ❖ loadExcursion(excursionID: integer): Η μέθοδος αυτή επιστρέφει μια εκδρομή από τη βάση δεδομένων, μέσω του μοναδικού αναγνωριστικού της.
- ❖ checkCreator(userID: integer, excursionID: integer): Η μέθοδος αυτή ελέγχει αν ο χρήστης της εφαρμογής είναι ο δημιουργός της εκδρομής, η οποία φέρει το αναγνωριστικό που δίνεται ως όρισμα.

### Entity ExcursionEntity



Αυτή είναι η κλάση οντότητας της εκδρομής και περιέχει τα στοιχεία για τον ορισμό της.

Χαρακτηριστικά της κλάσης:

- ❖ ExcursionName: String για το όνομα της εκδρομής
- ❖ Location: String για την τοποθεσία της εκδρομής
- ❖ Price: Το ενδεικτικό κόστος της εκδρομής
- ❖ Description: String για τη γενική περιγραφή της εκδρομής
- ❖ StartDate: String για την ημερομηνία έναρξης της εκδρομής
- ❖ EndDate: String για την ημερομηνία λήξης της εκδρομής
- ❖ ApplicationFormDeadline: String με την ημερομηνία λήξης για την προθεσμία υποβολής αιτήσεων συμμετοχής
- ❖ ExcursionID: Ο μοναδικός χαρακτηριστικός αριθμός για την κάθε εκδρομή
- ❖ CreatorID: Ο μοναδικός χαρακτηριστικός αριθμός για το χρήστη-δημιουργό της εκδρομής

Μέθοδοι της κλάσης:

- ❖ ExcursionEntity(name,loc,pr,desc,start,end,appd,crID,exID): Μέθοδος δόμησης
- ❖ setExcursionData(name,loc,pr,desc,start,end,appd): Μέθοδος η οποία θέτει τιμές στα χαρακτηριστικά της κλάσης.
- ❖ setExcursionName(excName: string): Μέθοδος που θέτει το όνομα της εκδρομής.
- ❖ setLocation(loc: string): Μέθοδος που θέτει την τοποθεσία της εκδρομής.
- ❖ setPrice(price: float): Μέθοδος που θέτει το κόστος της εκδρομής.
- ❖ setDescription(desc: string): Μέθοδος που θέτει τη γενική περιγραφή της εκδρομής.
- ❖ setStartDate(startDate: string): Μέθοδος που θέτει την ημερομηνία έναρξης της εκδρομής.
- ❖ setEndDate(endDate: string): Μέθοδος που θέτει την ημερομηνία λήξης της εκδρομής.
- ❖ setApplicationFormDeadline(deadline: string): Μέθοδος που θέτει την προθεσμία υποβολής αιτήσεων συμμετοχής.
- ❖ setExcursionID(exclD: integer): Μέθοδος που θέτει το μοναδικό αναγνωριστικό της εκδρομής.
- ❖ setCreatorID(creatorID: integer): Μέθοδος που θέτει το μοναδικό αναγνωριστικό του χρήστη-δημιουργού της εκδρομής.
- ❖ getExcursionName(): Η μέθοδος επιστρέφει το όνομα μιας εκδρομής.
- ❖ getLocation(): Η μέθοδος επιστρέφει την τοποθεσία διεξαγωγής μιας εκδρομής.
- ❖ getPrice(): Η μέθοδος επιστρέφει το ενδεικτικό κόστος μιας εκδρομής.
- ❖ getDescription(): Η μέθοδος επιστρέφει τη γενική περιγραφή μιας εκδρομής.
- ❖ getStartDate(): Η μέθοδος επιστρέφει την ημερομηνία έναρξης μιας εκδρομής.
- ❖ getEndDate(): Η μέθοδος επιστρέφει την ημερομηνία λήξης μιας εκδρομής.
- ❖ getApplicationFormDeadline(): Η μέθοδος επιστρέφει την ημερομηνία λήξης υποβολής αιτήσεων μιας εκδρομής.
- ❖ getExcursionID(): Η μέθοδος επιστρέφει το μοναδικό αναγνωριστικό μιας εκδρομής.
- ❖ getCreatorID(): Η μέθοδος αυτή επιστρέφει το μοναδικό χαρακτηριστικό αριθμό του χρήστη-δημιουργού της εκδρομής.

### Controller CreateEditExcursionControl

«Controller» CreateEditExcursionControl
-userID: integer -excursionID: integer -creatorID: integer
+CreateEditExcursionControl(userID, exclD, crID): void +storeExcursion(excursion: ExcursionEntity): void +loadExcursion(excursionID: integer): ExcursionEntity +editExcursion(excursionID: integer): boolean +checkEnteredData(): boolean +returnError(EnteredDataFlag: boolean): string +cancelControl(): void +back(): void

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

#### Χαρακτηριστικά της κλάσης:

- ❖ userID: Ο μοναδικός χαρακτηριστικός αριθμός για τον κάθε χρήστη.
- ❖ excursionID: Ο μοναδικός χαρακτηριστικός αριθμός για την κάθε εκδρομή
- ❖ creatorID: Ο μοναδικός χαρακτηριστικός αριθμός για το χρήστη-δημιουργό της εκδρομής

#### Μέθοδοι της κλάσης:

- ❖ CreateEditExcursionControl(userID, exclD, crID): Πρόκειται για μέθοδο δόμησης του ελεγκτή.
- ❖ storeExcursion(excursion: ExcursionEntity): Η μέθοδος αυτή καλεί την αντίστοιχη συνάρτηση αποθήκευσης εκδρομής στο proxy αντικείμενο με σκοπό την αποθήκευση της αντίστοιχης εκδρομής.
- ❖ loadExcursion(excursionID: integer): Η μέθοδος αυτή καλεί την αντίστοιχη συνάρτηση φόρτωσης εκδρομής στο proxy αντικείμενο, βάσει του μοναδικού αναγνωριστικού της εν λόγω εκδρομής.
- ❖ editExcursion(excursionID: integer): Η μέθοδος καλείται κάθε φορά που ο χρήστης επιθυμεί να δημιουργήσει ή να τροποποιήσει μια εκδρομή.
- ❖ checkEnteredData(): Η συνάρτηση αυτή ελέγχει αν ο χρήστης συμπλήρωσε σωστά τη φόρμα δημιουργίας/τροποποίησης της εκδρομής. Σε περίπτωση ορθότητας επιστρέφει τη λογική τιμή true, ειδάλλως επιστρέφει false.
- ❖ returnError(EnteredDataFlag: boolean): Η συνάρτηση αυτή επιστρέφει μήνυμα σφάλματος σε περίπτωση εσφαλμένης συμπλήρωσης της φόρμας στη συνάρτηση displayError() της οριακής κλάσης ErrorCreateExcursionGUI.
- ❖ cancelControl(): Η συνάρτηση αυτή πραγματοποιεί επιστροφή στη σελίδα CreateEditExcursionGUI.
- ❖ back(): Η συνάρτηση αυτή πραγματοποιεί επιστροφή από τη σελίδα εμφάνισης σφάλματος στη σελίδα CreateEditExcursionGUI.



**Controller ChooseExcursionControl**

«Controller» ChooseExcursionControl	
-userID: integer	
-excursionID: integer	
-creatorID: integer	
+chooseExcursionControl(userID, exclD, crID): void	
+chooseGUI(userID: integer, excursionID: integer): void	

Ο ελεγκτής αυτός είναι υπεύθυνος για την επιλογή του κατάλληλου περιβάλλοντος διεπαφής για μια εκδρομή ανάλογα με την ιδιότητα του χρήστη που την επιλέγει. Εάν ο χρήστης είναι ο δημιουργός της επιλέγεται η οριακή κλάση ExcursionCreatorGUI, ειδάλλως επιλέγεται η Excursion-UserGUI.

Χαρακτηριστικά της κλάσης:

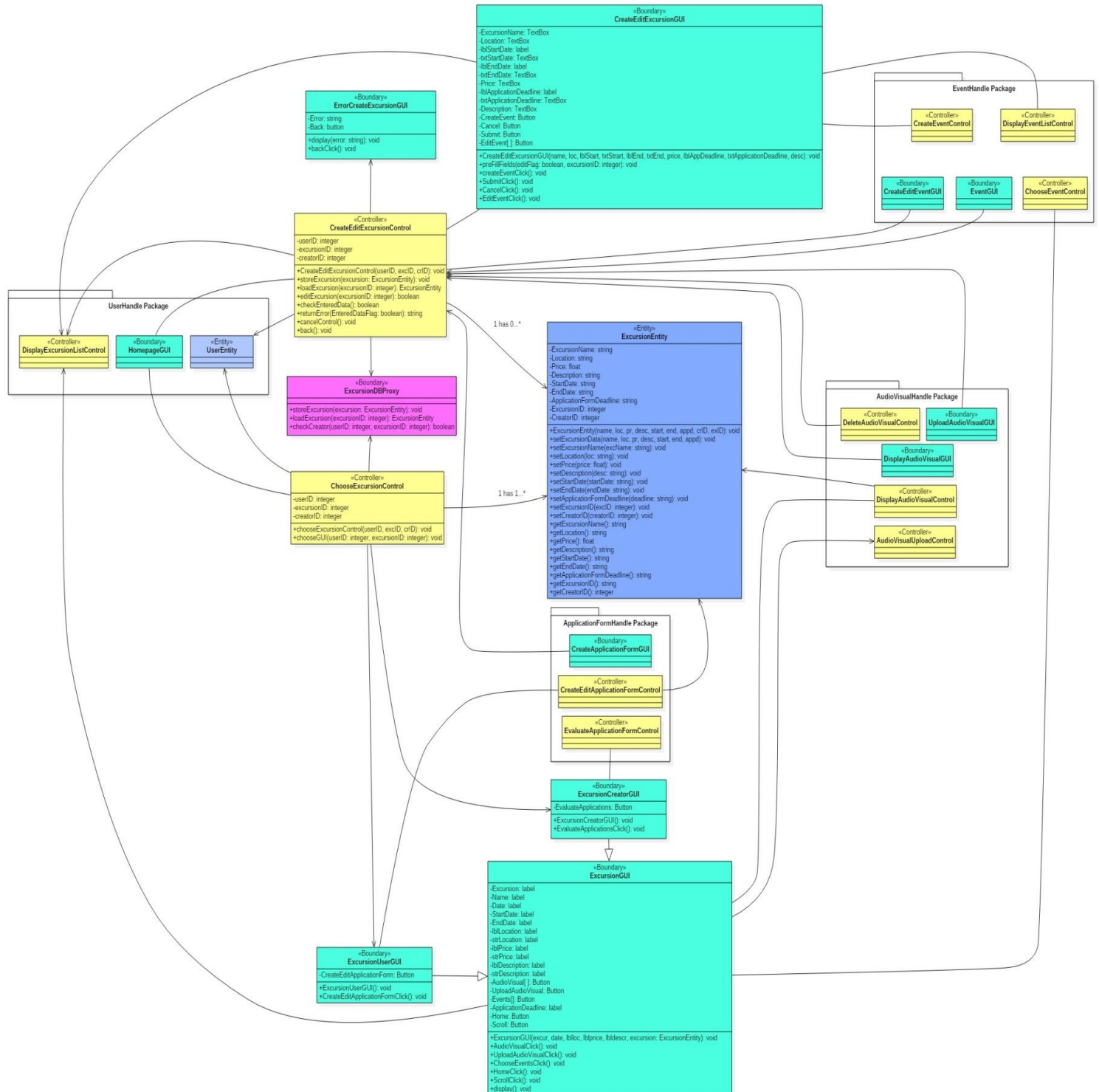
- ❖ userID: Ο μοναδικός χαρακτηριστικός αριθμός για τον κάθε χρήστη.
- ❖ excursionID: Ο μοναδικός χαρακτηριστικός αριθμός για την κάθε εκδρομή
- ❖ creatorID: Ο μοναδικός χαρακτηριστικός αριθμός για το χρήστη-δημιουργό της εκδρομής

Μέθοδοι της κλάσης:

- ❖ ChooseExcursionControl(userID, exclD, crID): Πρόκειται για μέθοδο δόμησης του ελεγκτή.
- ❖ chooseGUI(userID: integer, excursionID: integer): Η μέθοδος αυτή επιλέγει ποιο περιβάλλον διεπαφής μιας εκδρομής θα προβληθεί στο χρήστη, ανάλογα με το εάν αυτός είναι απλός χρήστης ή ο δημιουργός της εκδρομής.

**ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ:**

Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.



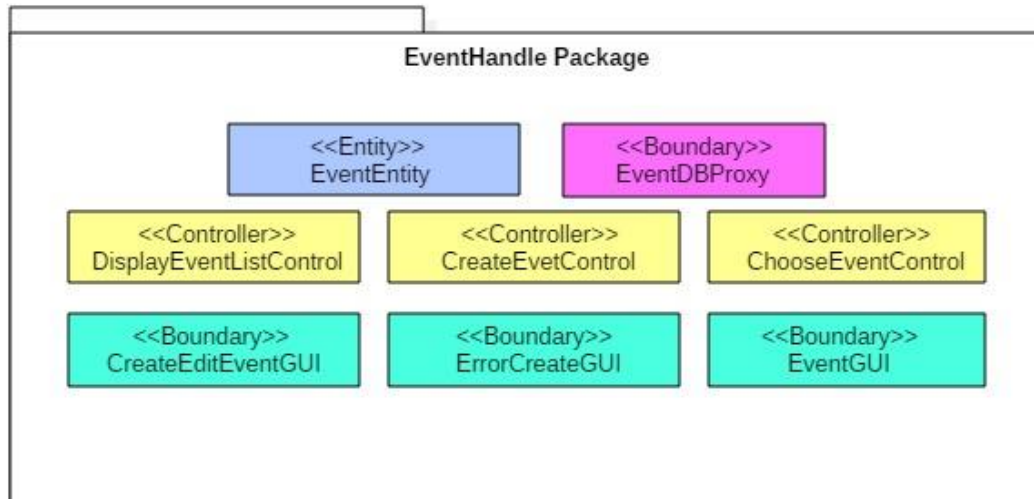
Σχήμα 2. Διάγραμμα κλάσεων πακέτου ExcursionHandle



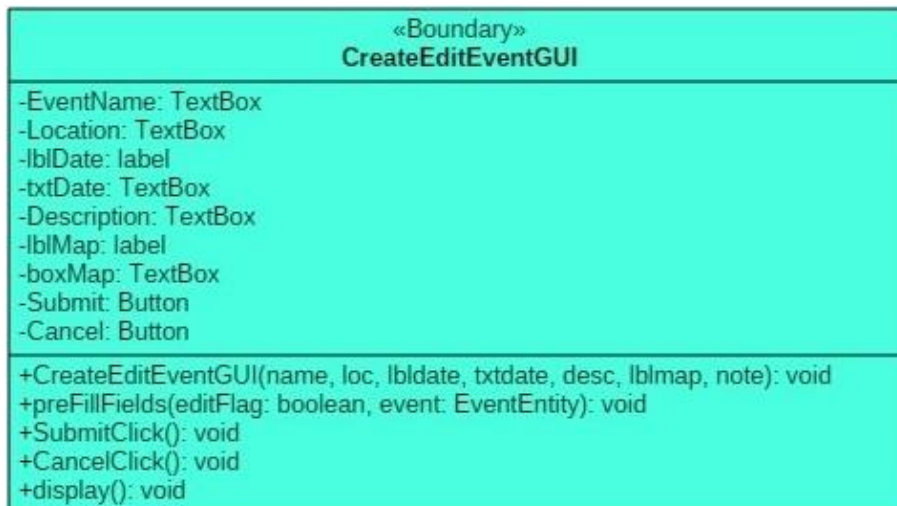
## 2.2 Πακέτα λεξιλογίου σεναρίων μέσης προτεραιότητας

### 2.2.1 Πακέτο EventHandle

Το συγκεκριμένο πακέτο αναφέρεται στις κλάσεις που σχετίζονται με τη διαχείριση των εκδηλώσεων, συμπεριλαμβανομένου της δημιουργίας τους, της τροποποίησής τους και της προβολής τους στο χρήστη.



#### Boundary CreateEditEventGUI



Αυτή η διεπαφή είναι υπεύθυνη για την δημιουργία και επεξεργασία εκδήλωσης.

#### Χαρακτηριστικά της κλάσης:

- ❖ EventName: TextBox για την εισαγωγή του ονόματος της εκδήλωσης.
- ❖ Location: TextBox για την εισαγωγή της τοποθεσίας της εκδήλωσης.
- ❖ lblDate: Πρόκειται για την εμφάνιση της λέξης «έναρξη εκδήλωσης» .
- ❖ txtDate: Textbox για την εισαγωγή της ημερομηνίας διεξαγωγής.
- ❖ Description: TextBox για την εισαγωγή της περιγραφής της εκδήλωσης.



- ❖ lblMap: Πρόκειται για την εμφάνιση της φράσης «Εισάγετε την τοποθεσία στο χάρτη».
- ❖ boxMap: TextBox για την εισαγωγή των συντεταγμένων της τοποθεσίας της εκδήλωσης.
- ❖ Submit: Κουμπί για την τελική υποβολή της εκδήλωσης.
- ❖ Cancel: Κουμπί για την ακύρωση δημιουργίας εκδήλωσης.

#### Μέθοδοι της κλάσης:

- ❖ CreateEditEventGUI(name,loc,lbldate,txtdate,desc,lblmap,note): Πρόκειται για τη συνάρτηση δόμησης της εκδήλωσης.
- ❖ preFillFields(editFlag:Boolean,event:EventEntity): Πρόκειται για τη συνάρτηση που προσυμπληρώνει τα πεδία σε περίπτωση που έχει πατηθεί το edit μίας εκδήλωσης. Πιο συγκεκριμένα όταν πατιέται το κουμπί EditEventClick() από το CreateEditExcursionGUI, τότε καλείται η συνάρτηση editEvent του controller που επιστρέφει τη μεταβλητή Boolean 1, η οποία δείχνει ότι η συνάρτηση έχει ήδη δημιουργηθεί άρα η preFillFields πρέπει να προσυμπληρώσει τα πεδία.
- ❖ SubmitClick(): Από αυτή τη συνάρτηση καλείται ο ελεγκτής CreateEventControl έτσι ώστε να αποθηκευτεί η εκδήλωση και να γίνει επιστροφή στη συνάρτηση της εκδρομής.
- ❖ CancelClick(): Από αυτή τη συνάρτηση καλείται ο ελεγκτής CreateEventControl έτσι ώστε να ακυρωθεί η διαδικασία και να γίνει επιστροφή στην σελίδα της εκδρομής.
- ❖ display(): Εμφανίζει το συγκεκριμένο GUI.

#### Boundary ErrorCreateGUI

«Boundary» ErrorCreateGUI
-Error: string -Back: button
+display(error: string): void +backClick(): void

Αυτή η κλάση διεπαφής εμφανίζει το σφάλμα δημιουργίας εκδήλωσης λόγω μη συμπλήρωσης όλων των πεδίων.

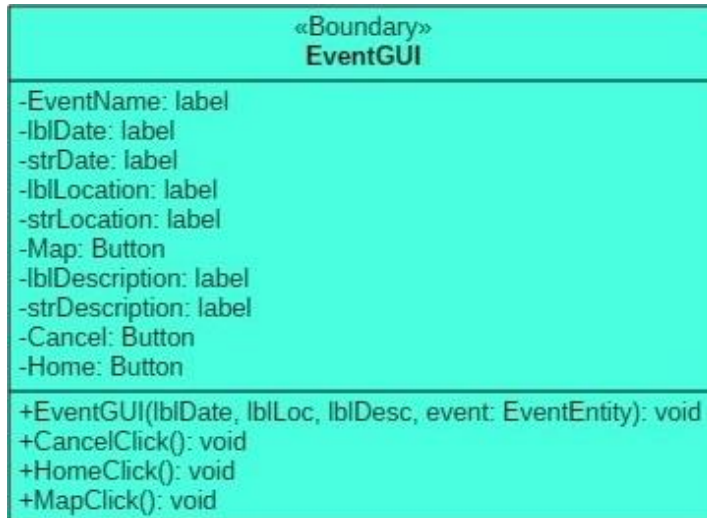
#### Χαρακτηριστικά της κλάσης:

- ❖ Error: Εμφανίζει το σφάλμα που βρέθηκε.
- ❖ Back: Κουμπί για την επιστροφή στην σελίδα δημιουργίας της εκδήλωσης.

#### Μέθοδοι της κλάσης:

- ❖ display(error:string): Πρόκειται για την συνάρτηση που εμφανίζει το μήνυμα λάθους.
- ❖ backClick(): Πρόκειται για τη συνάρτηση που καλείται με το πάτημα του κουμπιού back για έξοδο από την διεπαφή λάθους και επιστροφή στην σελίδα δημιουργίας της εκδήλωσης.

### **Boundary EventGUI**



Αυτή η κλάση είναι η διεπαφή εμφάνισης της εκδήλωσης.

#### Χαρακτηριστικά της κλάσης:

- ❖ EventName: Πρόκειται για την εμφάνιση του ονόματος της εκδρομής.
- ❖ lblDate: Πρόκειται για την εμφάνιση της λέξης «ημερομηνία διεξαγωγής».
- ❖ strDate: Η ημερομηνία διεξαγωγής της εκδήλωσης.
- ❖ lblLocation: Πρόκειται για την εμφάνιση της λέξης «τοποθεσία διεξαγωγής».
- ❖ strLocation: Η τοποθεσία διεξαγωγής της εκδήλωσης.
- ❖ Map: Κουμπί για το άνοιγμα του χάρτη.
- ❖ lblDescription: Πρόκειται για την εμφάνιση της λέξης «πληροφορίες».
- ❖ strDescription: Η περιγραφή της εκδήλωσης.
- ❖ Cancel: Κουμπί για την επιστροφή από τη σελίδα.
- ❖ Home: Κουμπί για την επιστροφή στην αρχική σελίδα.

#### Μέθοδοι της κλάσης:

- ❖ EventGUI(lblDate, lblLoc, lblDesc, event: EventEntity): Πρόκειται για τη συνάρτηση δόμησης της κλάσης.
- ❖ CancelClick(): Καλεί τον ελεγκτή ChooseEventControl για επιστροφή στην σελίδα της εκδρομής.
- ❖ HomeClick(): Επιστρέφει στην αρχική σελίδα μέσω του ελεγκτή displayExcursionListControl.
- ❖ MapClick(): Καλεί τον ελεγκτή MapControl για εμφάνιση του χάρτη.

### **Boundary EventDBProxy**



Αυτή η κλάση χρησιμοποιείται για την επικοινωνία με τη βάση δεδομένων των εκδηλώσεων.

Μέθοδοι της κλάσης:

- ❖ `storeEvent(event:EventEntity)`: Πρόκειται για τη συνάρτηση που αποθηκεύει μία εκδήλωση στη βάση δεδομένων.
- ❖ `loadEvent(eventID: integer)`: Πρόκειται για τη συνάρτηση που φορτώνει μία ζητούμενη εκδήλωση.
- ❖ `getCoordinates(EventID: integer)`: Πρόκειται για τη συνάρτηση που δοσμένου του `id` της εκδήλωσης επιστρέφει τις συντεταγμένες της τοποθεσίας της εκδήλωσης.

Entity EventEntity

«Entity» EventEntity
-EventName: string -Date: string -Description: string -Coordinates: float -EventID: integer
+EventEntity(id, name, date, desc, coord, eventID): void +setEventEntity(name, date, desc, coord, eventID): void +setEventName(name: string): void +setDate(date: string): void +setDescription(desc: string): void +setCoordinates(coord: float): void +setEventID(eventID: integer): void +getEventName(): string +getDate(): string +getDescription(): string +getCoordinates(): float +getEventID(): integer

Αυτή η κλάση περιέχει όλα τα δεδομένα και τις συναρτήσεις μια εκδήλωσης που χρειάζεται ο δημιουργός της.

Χαρακτηριστικά της κλάσης:

- ❖ `EventName`: Πρόκειται για το όνομα της εκδήλωσης.
- ❖ `Date`: Η ημερομηνία διεξαγωγής της εκδήλωσης.
- ❖ `Description`: Πρόκειται για την περιγραφή της εκδήλωσης.
- ❖ `Coordinates`: Οι συντεταγμένες της τοποθεσίας διεξαγωγής της εκδήλωσης.
- ❖ `EventID`: Ο μοναδικός χαρακτηριστικός αριθμός της εκδήλωσης.

Μέθοδοι της κλάσης:

- ❖ `EventEntity(id,name,date,desc,coord,eventID)`: Η συνάρτηση δόμησης της εκδήλωσης.
- ❖ `setEventEntity(name,date,desc,coord,eventID)`: Η συνάρτηση με την οποία μπορούν να μεταβληθούν τα δεδομένα της εκδήλωσης.
- ❖ `setEventName(name: string)`: Μέθοδος που θέτει το όνομα της εκδήλωσης.
- ❖ `setDate(date: string)`: Μέθοδος που θέτει την ημερομηνία διεξαγωγής της εκδήλωσης.



- ❖ setDescription(desc: string): Μέθοδος που θέτει τη γενική περιγραφή της εκδήλωσης.
- ❖ setCoordinates(coord: float): Μέθοδος που θέτει τις συντεταγμένες της τοποθεσίας της εκδήλωσης.
- ❖ setEventID(eventID: integer): Μέθοδος που θέτει το μοναδικό αναγνωριστικό της εκδήλωσης.
- ❖ getEventName(): Η συνάρτηση που επιστρέφει το όνομα της εκδήλωσης.
- ❖ getDate(): Η συνάρτηση που επιστρέφει την ημερομηνία διεξαγωγής της εκδήλωσης.
- ❖ getDescription(): Η συνάρτηση που επιστρέφει την περιγραφή της εκδήλωσης.
- ❖ getCoordinates(): Η συνάρτηση που επιστρέφει τις συντεταγμένες διεξαγωγής της εκδήλωσης.
- ❖ getEventID(): Η συνάρτηση που επιστρέφει το μοναδικό χαρακτηριστικό αριθμό της εκδήλωσης.

#### Controller CreateEventControl

«Controller» CreateEventControl	
-EventID: integer	
+CreateEventControl(evID): void +storeEvent(event: EventEntity): void +checkEnteredData(): boolean +returnError(checkFlag: boolean): string +editEvent(eventID: integer): boolean +loadEvent(eventID: integer): EventEntity +cancel(): void +back(): void	

Αυτός ο ελεγκτής είναι υπεύθυνος για την δημιουργία και επεξεργασία εκδήλωσης.

#### Χαρακτηριστικά της κλάσης:

- ❖ EventID: Πρόκειται για το μοναδικό χαρακτηριστικό της εκδήλωσης που θα χρησιμοποιηθεί από τον ελεγκτή.

#### Μέθοδοι της κλάσης:

- ❖ CreateEventControl(evID): Η συνάρτηση δόμησης του ελεγκτή.
- ❖ storeEvent(event:EventEntity): Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αποθήκευσης μίας εκδήλωσης στο proxy αντικείμενο eventDBProxy με σκοπό την αποθήκευση μίας εκδήλωσης.
- ❖ checkEnteredData(): Πρόκειται για τη συνάρτηση που ελέγχει αν έχουν συμπληρωθεί τα πεδία και επιστρέφει 1 για συμπληρωμένη φόρμα και 0 για μη συμπληρωμένη.
- ❖ returnError(checkFlag:Boolean): Πρόκειται για τη συνάρτηση που επιστρέφει σφάλμα σε περίπτωση μη ολοκληρωμένης φόρμας.
- ❖ editEvent(eventID:integer): Πρόκειται για τη συνάρτηση που σε περίπτωση που έχει πατηθεί το edit button επιστρέφει τα ήδη συμπληρωμένα πεδία για να εμφανιστούν στην φόρμα.
- ❖ loadEvent(eventID:integer): Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση φόρτωσης μίας εκδήλωσης του αντικειμένου eventDBProxy. Αυτή η διαδικασία γίνεται με είσοδο το μοναδικό αναγνωριστικό της κάθε εκδήλωσης.



- ❖ `cancel()`: Πρόκειται για τη συνάρτηση που ακυρώνει τη διαδικασία συμπλήρωσης της φόρμας και επιστρέφει στη δημιουργία εκδρομής.
- ❖ `back()`: Πρόκειται για τη συνάρτηση που επιστρέφει στην δημιουργία της φόρμας, σε περίπτωση σφάλματος.

#### Controller DisplayEventListControl

«Controller» <b>DisplayEventListControl</b>
+List: EventEntity
+DisplayEventListControl(): void +DisplayEditButton(): void

Αυτή η κλάση ελέγχου εμφανίζει την λίστα με τις υπάρχουσες εκδηλώσεις για μία εκδρομή στα GUI που απαιτείται.

#### Χαρακτηριστικά της κλάσης:

- ❖ List: Πρόκειται για μία λίστα από εκδηλώσεις που πρόκειται να εμφανιστούν στα GUI ExcursionGUI και CreateExcursionGUI.

#### Μέθοδοι της κλάσης:

- ❖ `DisplayEventListControl()`: Πρόκειται για συνάρτηση δόμησης του ελεγκτή
- ❖ `DisplayEditButton()`: Η μέθοδος αυτή είναι υπεύθυνη για την κλήση των κατάλληλων συναρτήσεων των ExcursionGUI και CreateExcursionGUI με σκοπό την εμφάνιση των συνδέσμων τροποποίησης στις εκδηλώσεις που έχει δημιουργήσει ο ίδιος χρήστης.

#### Controller ChooseEventControl

«Controller» <b>ChooseEventControl</b>
-EventID: integer
+DisplayOneEvent(eventID: integer): void +cancel(): void

Αυτή η κλάση ελέγχου καλείται από το ExcursionGUI έτσι ώστε εμφανιστεί το GUI της κατάλληλης εκδήλωσης.

#### Χαρακτηριστικά της κλάσης:

- ❖ EventID: πρόκειται για το μοναδικό χαρακτηριστικό της κάθε εκδήλωσης που χρειάζεται να υπάρχει σε αυτή την κλάση ελέγχου.

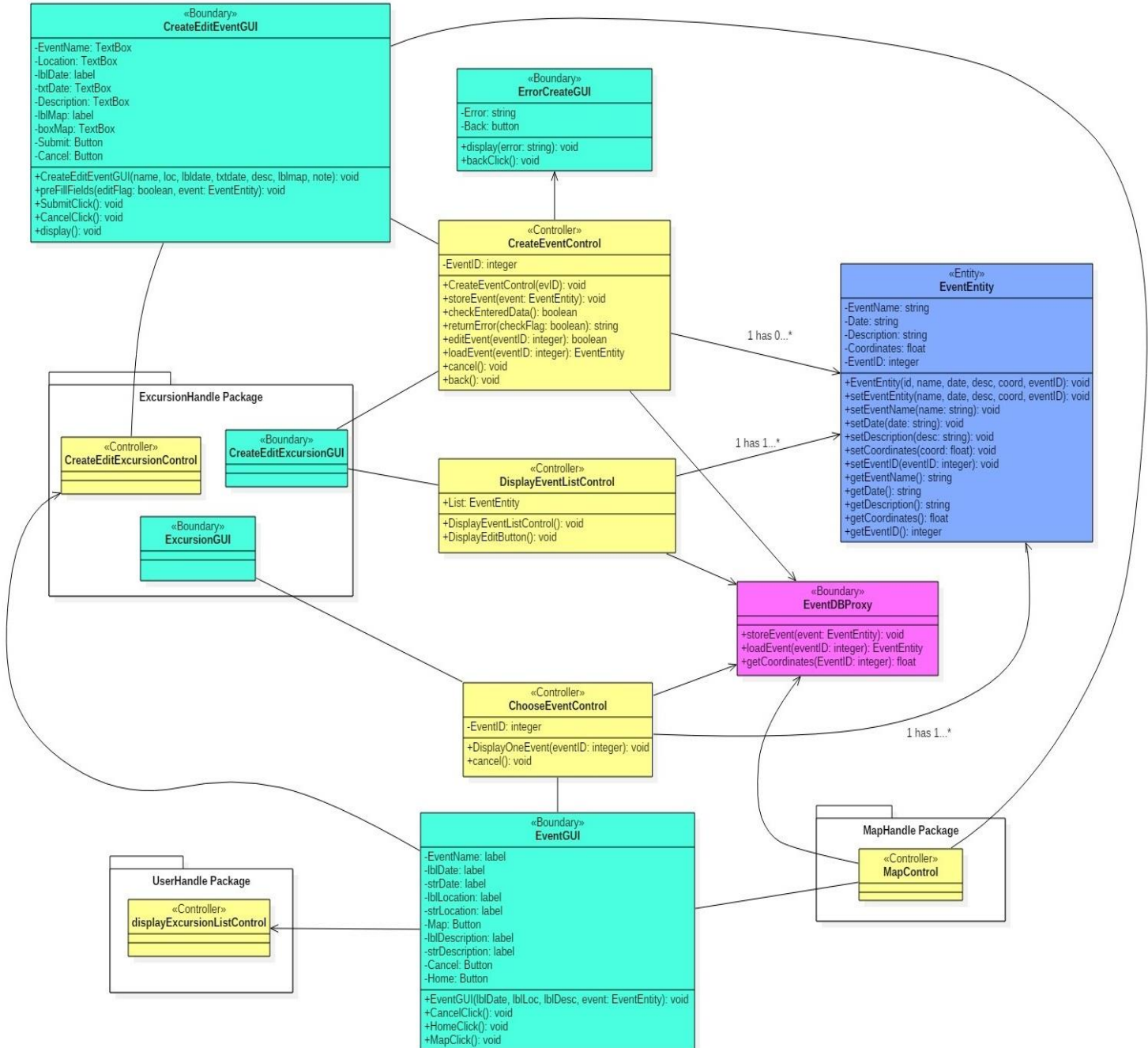
#### Μέθοδοι της κλάσης:

- ❖ `DisplayOneEvent(eventID: integer)`: Πρόκειται για τη συνάρτηση που εμφανίζει μία εκδήλωση. Η συνάρτηση αυτή ενεργοποιείται από το ChooseEventClick και καλεί την συνάρτηση load της βάσης δεδομένων για να φορτωθεί η εκδρομή.
- ❖ `cancel()`: Πρόκειται για τη συνάρτηση που ακυρώνει τη διαδικασία συμπλήρωσης της φόρμας και επιστρέφει στη δημιουργία εκδρομής.



**ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ:**

Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.

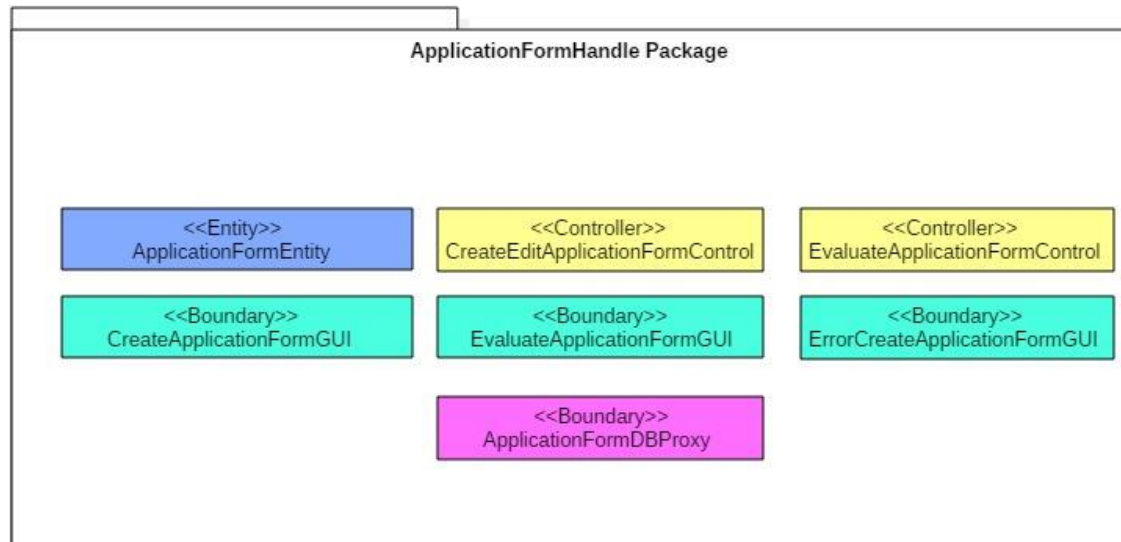


Σχήμα 3. Διάγραμμα κλάσεων πακέτου EventHandle

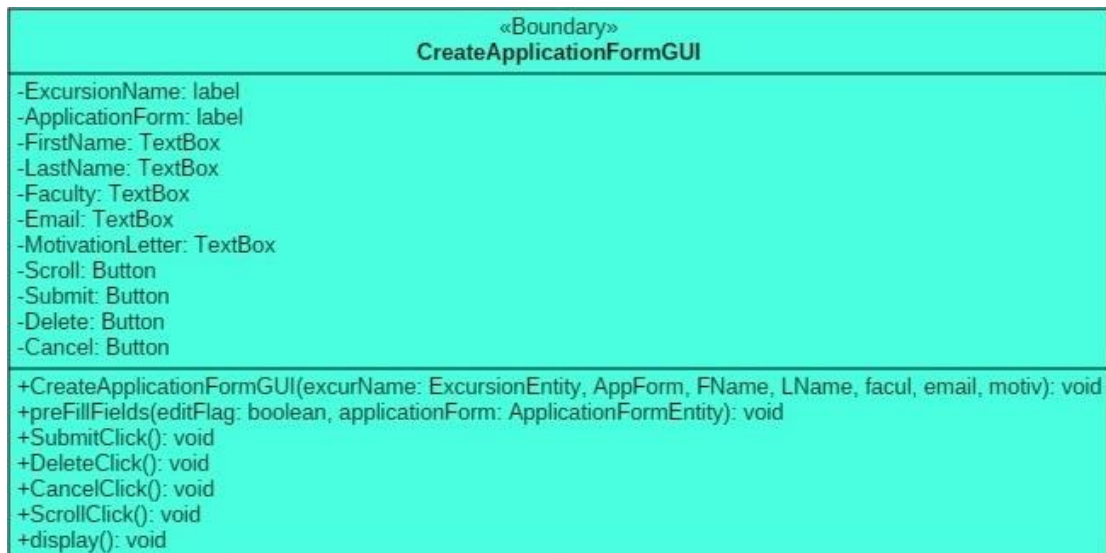


## 2.2.2 Πακέτο ApplicationFormHandle

Η σκοπιμότητα του πακέτου αυτού είναι η διαχείριση των αιτήσεων συμμετοχής των χρηστών σε μια εκδρομή. Οι κλάσεις που το απαρτίζουν αναλαμβάνουν δημιουργία και τροποποίηση μιας αίτησης συμμετοχής από τον απλό χρήστη, καθώς και αξιολόγησή της από το χρήστη-δημιουργό της εκδρομής.



### Boundary CreateApplicationFormGUI



Η παρούσα οθόνη διεπαφής περιέχει την φόρμα αίτησης που θα πρέπει να συμπληρώσει ο χρήστης ώστε να γίνει δεκτός στην εκδρομή.

#### Χαρακτηριστικά της κλάσης:

- ❖ ExcursionName:Ετικέτα με το όνομα της εκδρομής για την οποία κάνει αίτηση ο χρήστης.
- ❖ ApplicationForm:Η ετικέτα με το κείμενο «αίτηση συμμετοχής».
- ❖ FirstName:Ενα κουτί για τη συμπλήρωση του ονόματος του χρήστη που κάνει την αίτηση.





- ❖ LastName: Ένα κουτί για τη συμπλήρωση του επιθέτου του χρήστη που κάνει την αίτηση.
- ❖ Faculty: Ένα κουτί για τη συμπλήρωση του τμήματος που σπουδάζει ο χρήστης που κάνει την αίτηση.
- ❖ Email: Ένα κουτί για τη συμπλήρωση της ηλεκτρονικής διεύθυνσης του χρήστη που κάνει την αίτηση.
- ❖ MotivationLetter: Ένα κουτί για τη συμπλήρωση ενός κειμένου με του λόγους που πρέπει να γίνει δεκτός, ο χρήστης που κάνει την αίτηση, στην εκδρομή.
- ❖ Scroll: Ένα κουμπί που ανεβοκατεβάζει το κείμενο στο MotivationLetter.
- ❖ Submit: Ένα κουμπί που όταν το πατάει ο χρήστης υποβάλλει την αίτηση του.
- ❖ Delete: Ένα κουμπί που όταν το πατάει ο χρήστης διαγράφει την αίτηση του.
- ❖ Cancel: Ένα κουμπί που όταν το πατάει ο χρήστης βγαίνει από τη διεπαφή.

#### Μέθοδοι της κλάσης:

- ❖ CreateApplicationFormGUI(ExcurName:ExcursionEntity,AppForm,FName,LName,facul, email, motiv): Αυτή η συνάρτηση δόμησης της διεπαφής αρχικοποιεί τις μεταβλητές της, με τις τιμές που δίνονται ως όρισμα.
- ❖ preFillFields(editFlag:Boolean,applicationForm:ApplicationFormEntity): Η συνάρτηση αυτή καλείται από ο χρήστης έχει πατήσει τροποποίηση αίτησης και εμφανίζει την αίτηση του με τις ήδη υπάρχουσες τιμές.
- ❖ SubmitClick(): Η συνάρτηση ενεργοποιείται όταν ο χρήστης πατήσει το κουμπί Submit και στέλνει σήμα ώστε να καταχωρηθεί η αίτηση στη βάση δεδομένων.
- ❖ DeleteClick(): Η συνάρτηση ενεργοποιείται όταν ο χρήστης πατήσει το κουμπί Delete και στέλνει σήμα ώστε να διαγραφεί η αίτηση στη βάση δεδομένων.
- ❖ CancelClick(): Η συνάρτηση ενεργοποιείται όταν ο χρήστης πατήσει το κουμπί Cancel και στέλνει σήμα ώστε φύγει ο χρήστης από τη διεπαφή.
- ❖ ScrollClick(): Η συνάρτηση ενεργοποιείται όταν ο χρήστης πατήσει το κουμπί Scroll και στέλνει σήμα ώστε να πάει προς τα πάνω ή κάτω το κείμενο του MotivationLetter.
- ❖ display(): Η συνάρτηση αυτή είναι υπεύθυνη για την εμφάνιση της παρούσας διεπαφής.

#### Boundary ErrorCreateApplicationFormGUI

«Boundary» ErrorCreateApplicationFormGUI	
-Error: string	
-Back: button	
+display(error: string): void	
+backClick(): void	

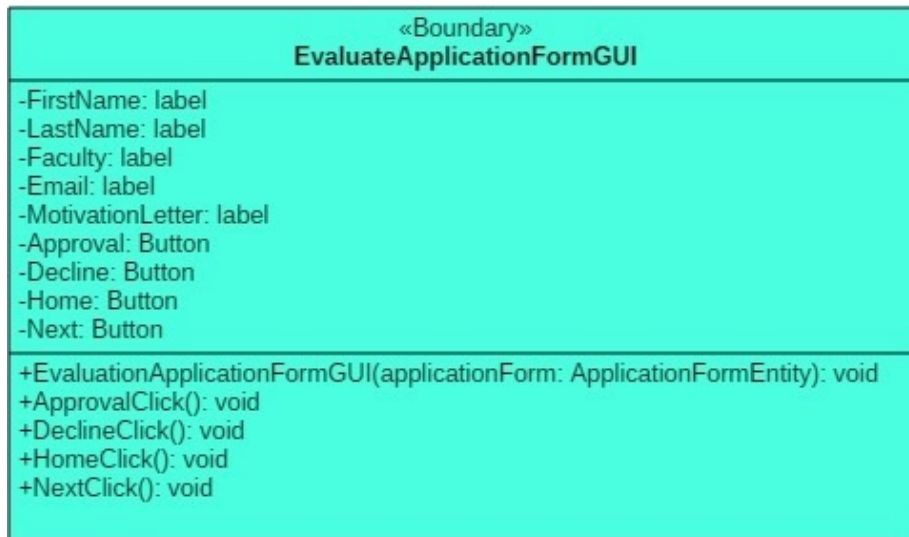
Αυτή η κλάση διεπαφής εμφανίζει το σφάλμα δημιουργίας αιτήσεων συμμετοχής λόγω μη συμπλήρωσης όλων των πεδίων.

#### Χαρακτηριστικά της κλάσης:

- ❖ Error: Εμφανίζει το σφάλμα που βρέθηκε.
- ❖ Back: Button για την επιστροφή στην σελίδα δημιουργίας της εκδήλωσης.

Μέθοδοι της κλάσης:

- ❖ display(error:string): Πρόκειται για την συνάρτηση που εμφανίζει το μήνυμα λάθους.
- ❖ backClick(): Πρόκειται για τη συνάρτηση που καλείται με το πάτημα του κουμπιού back για έξοδο από την διεπαφή λάθους και επιστροφή στην σελίδα της εκδρομής.

BoundaryEvaluateApplicationFormGUI

Αυτή η οθόνη διεπαφής εμφανίζεται στο οργανωτή της εκδρομής με σκοπό αυτός να δει μια αίτηση ώστε να την δεχθεί ή να την απορίψει.

Χαρακτηριστικά της κλάσης:

- ❖ FirstName: Αυτή η ετικέτα περιέχει το όνομα του χρήστη που έκανε την αίτηση.
- ❖ LastName: Αυτή η ετικέτα περιέχει το επίθετο του χρήστη που έκανε την αίτηση.
- ❖ Faculty: Αυτή η ετικέτα περιέχει το τμήμα που σπουδάζει ο χρήστης που έκανε την αίτηση.
- ❖ Email: Αυτή η ετικέτα περιέχει το email του χρήστη που έκανε την αίτηση.
- ❖ MotivationLetter: Αυτή η ετικέτα περιέχει το κείμενο με τους λόγους που θέλει να πάει στην εκδρομή ο χρήστης που έκανε την αίτηση.
- ❖ Approval: Ένα κουμπί που γράφει έγκριση και όταν το πατάει ο οργανωτής αποδέχεται τον χρήστη που έκανε την αίτηση.
- ❖ Decline: Ένα κουμπί που γράφει απόρριψη και όταν το πατάει ο οργανωτής απορρίπτει τον χρήστη που έκανε την αίτηση.
- ❖ Home: Ένα κουμπί που όταν το πατάει ο οργανωτής επιστρέφει στην αρχική σελίδα.
- ❖ Next: Ένα κουμπί που όταν το πατάει ο οργανωτής μεταβαίνει στην επόμενη αίτηση.

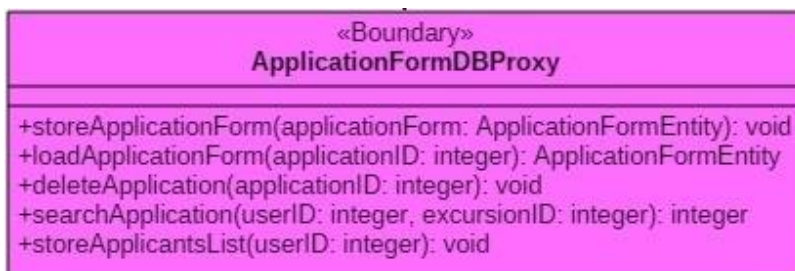
Μέθοδοι της κλάσης:

- ❖ EvaluationApplicationFormGUI(applicationForm:ApplicationFormEntity): Η συνάρτηση δόμησης της διεπαφής αρχικοποιεί τις μεταβλητές της , λαμβάνοντας ως όρισμα την αίτηση για την οποία γίνεται η αξιολόγηση.
- ❖ ApprovalClick(): Ενεργοποιείται όταν πατηθεί το κουμπί Approval και στέλνει σήμα να γίνει δεκτός ο χρήστης που έκανε την αίτηση.



- ❖ DeclineClick(): Ενεργοποιείται όταν πατηθεί το κουμπί Decline και στέλνει σήμα να διαγραφεί η αίτηση.
- ❖ HomeClick(): Ενεργοποιείται όταν πατηθεί το κουμπί Home και στέλνει σήμα να φύγει από αυτήν την διεπαφή.
- ❖ NextClick(): Ενεργοποιείται όταν πατηθεί το κουμπί Next και στέλνει σήμα να ανοίξει η ίδια διεπαφή με τα στοιχεία της επόμενης αίτησης.

#### **Boundary ApplicationFormDBProxy**



Η κλάση αυτή χρησιμοποιείται για την επικοινωνία των χρηστών με τη βάση δεδομένων. Μέσω αυτής γίνεται η αποθήκευση και τροποποίηση μιας αίτησης καθώς και αναζήτηση μιας αίτησης που έχει γίνει και καλείται για αξιολόγηση.

#### **Μέθοδοι της κλάσης:**

- ❖ storeApplicationForm(applicationForm:ApplicationFormEntity):Κάλεται όταν ο χρήστη έχει υποβάλει μια αίτηση και έχει συμπληρώσει σωστά τα στοιχεία του.Η συνάρτηση παίρνει την αίτηση που δέχεται ως όρισμα και την αποθηκεύει στη βάση δεδομένων.
- ❖ loadApplicationForm(applicationID:integer): Επιστρέφει την αίτηση που έχει το συγκεκριμένο applicationID που δέχεται ως όρισμα.
- ❖ deleteApplication(applicationID:integer): Διαγράφει την αίτηση που έχει το συγκεκριμένο applicationID που δέχεται ως όρισμα.
- ❖ searchApplication(userID:integer,excursionID:integer): Επιστρέφει τον κωδικό της αίτησης που έχει κάνει ο χρήστης με το userID για την εκδρομή excursionID.
- ❖ storeApplicantsList(userID: integer): Αποθηκεύει το μοναδικό αναγνωριστικό του χρήστη σε μια λίστα συμμετεχόντων, κατόπιν εγκρίσεως της αίτησης συμμετοχής του.

### Entity ApplicationFormEntity

«Entity» ApplicationFormEntity
-FirstName: string -LastName: string -Faculty: string -Email: string -MotivationLetter: string -UploaderID: integer -ExcursionID: integer -ApplicationFormID: integer
+ApplicationForm(first, last, sch, email, letter, userID, excID, applID): void +setApplicationData(first, last, sch, email, letter): void +setFirstName(first: string): void +setLastName(last: string): void +setFaculty(faculty: string): void +setEmail(email: string): void +setMotivLetter(motivLetter: string): void +setIDs(uploaderID: integer, excID: integer, applID: integer): void +getFirstName(): string +getLastName(): string +getFaculty(): string +getEmail(): string +getMotivationLetter(): string +getUploaderID(): integer +getExcursionID(): integer +getApplicationFormID(): integer

Αυτή η κλάση περιέχει όλα τα δεδομένα και τις συναρτήσεις μια αίτησης που κάνει ο χρήστης , ώστε να παρευρεθεί στην εκδρομή.

#### Χαρακτηριστικά της κλάσης:

- ❖ FirstName: Το όνομα του χρήστη που κάνει την αίτηση.
- ❖ LastName: Το επώνυμο του χρήστη που κάνει την αίτηση.
- ❖ Faculty: Το τμήμα που σπουδάζει ο χρήστης που κάνει την αίτηση.
- ❖ Email: Η επιθυμητή ηλεκτρονική διεύθυνση επικοινωνίας του χρήστη.
- ❖ MotivationLetter: Ένα κείμενο που εισάγει ο χρηστής παραθέτοντας τους λόγους που τον ενδιαφέρει η εκδρομή.
- ❖ UploaderID: Ο κωδικός του χρήστη που υποβάλει την αίτηση.
- ❖ ExcursionID: Ο κωδικός της εκδρομής για την οποία κάνει αίτηση ο χρήστης.
- ❖ ApplicationFormID: Ο κωδικός της αίτησης που κάνει ο χρήστης.

#### Μέθοδοι της κλάσης:

- ❖ ApplicationForm(first,last,sch,email,letter,userID,excID,applID): Η συνάρτηση δόμησης της κλάσης που αρχικοποιεί όλες τις μεταβλητές της κλάσης .
- ❖ setApplicationData(first,last,sch,email,letter): Η συνάρτηση που θέτει τιμές στις μεταβλητές της κλάσης.
- ❖ setFirstName(first: string): Η συνάρτηση που θέτει το όνομα του χρήστη που υποβάλλει την αίτηση.
- ❖ setLastName(last: string): Η συνάρτηση που θέτει το επώνυμο του χρήστη που υποβάλλει την αίτηση.
- ❖ setFaculty(faculty: string): Η συνάρτηση που θέτει τη σχολή φοίτησης του χρήστη.
- ❖ setEmail(email: string): Η συνάρτηση που θέτει το email του χρήστη.



- ❖ setMotivLetter(motivLetter: string): Η συνάρτηση που θέτει το κείμενο με λόγους παλεύρεσης στην εκδρομή του χρήστη.
- ❖ setIDs(uploaderID: integer, excID: integer, applID: integer): Η συνάρτηση που θέτει το μοναδικό αναγνωριστικό του χρήστη, της εκδρομής και της συγκεκριμένης αίτησης συμμετοχής.
- ❖ getFirstName(): Επιστρέφει το όνομα του χρήστη που έκανε την αίτηση.
- ❖ getLastName(): Επιστρέφει το επώνυμο του χρήστη που έκανε την αίτηση.
- ❖ getFaculty(): Επιστρέφει το τμήμα που σπουδάζει ο χρήστης που έκανε την αίτηση.
- ❖ getEmail(): Επιστρέφει την ηλεκτρονική διεύθυνση του χρήστη που έκανε την αίτηση.
- ❖ getMotivationLetter(): Επιστρέφει το κείμενο με λόγους παλεύρεσης στην εκδρομή του χρήστη που έκανε την αίτηση.
- ❖ getUploaderID(): Επιστρέφει το μοναδικό αναγνωριστικό του χρήστη που έκανε την αίτηση.
- ❖ getExcursionID(): Επιστρέφει το μοναδικό αναγνωριστικό της εκδρομής στην οποία ανήκει η αίτηση.
- ❖ getApplicationFormID(): Επιστρέφει το μοναδικό αναγνωριστικό μιας αίτησης συμμετοχής.

#### **Controller CreateEditApplicationFormControl**

«Controller» CreateEditApplicationFormControl	
-uploaderID: integer -excursionID: integer -applicationFormID: integer	
+CreateEditApplicationFormControl(upID, excID, applID): void +storeApplication(applicationForm: ApplicationFormEntity): void +loadApplication(userID: integer, excursionID: integer): void +checkApplicationData(): boolean +returnError(checkFlag: boolean): void +editApplicationForm(): boolean +deleteApplicationForm(userID: integer, excursionID: integer): void +cancel(): void +back(): void	

Ο ελεγκτής αυτός χρησιμοποιείται για τον έλεγχο της δημιουργίας και επεξεργασίας μίας αίτησης.

#### **Χαρακτηριστικά της κλάσης:**

- ❖ uploaderID: Πρόκειται για το μοναδικό αναγνωριστικό του χρήστη που υποβάλλει την αίτηση συμμετοχής.
- ❖ excursionID: Πρόκειται για το μοναδικό αναγνωριστικό της εκδρομής
- ❖ applicationFormID: Πρόκειται για το μοναδικό αναγνωριστικό της αίτησης συμμετοχής.

#### **Μέθοδοι της κλάσης:**

- ❖ CreateEditApplicationFormControl(upID,excID,applID): Πρόκειται για τη συνάρτηση δόμησης του ελεγκτή.
- ❖ storeApplication(applicationForm:ApplicationFormEntity): Αυτή η συνάρτηση δέχεται ως όρισμα ένα entity application Form η αποθήκευση του οποίου γίνεται με την κλήση της αντίστοιχης συνάρτησης αποθήκευσης μίας αίτησης συμμετοχής του αντικειμένου applicationFormDBProxy.
- ❖ loadApplication(userID:integer,excursionID:integer): Αυτή η συνάρτηση φορτώνει μία φόρμα αίτησης χρησιμοποιώντας την αντίστοιχη συνάρτηση load του αντικειμένου application-





FormDBProxy. Αυτή η διαδικασία γίνεται με ορίσματα το ID του χρήστη που βρίσκεται συνδεδεμένος στο σύστημα καθώς και το ID της επιλεγμένης εκδρομής.

- ❖ checkApplicationData(): Η συνάρτηση αυτή ελέγχει εάν έχουν συμπληρωθεί όλα τα πεδία της φόρμας αίτησης.
- ❖ returnError(checkFlag:boolean): Η παρούσα συνάρτηση καλεί το ErrorCreateApplicationFormGUI σε περίπτωση ασυμπλήρωτων πεδίων.
- ❖ editApplicationForm(): Σε περίπτωση που η αίτηση έχει ήδη δημιουργηθεί και επιθυμείται η επεξεργασία της η συνάρτηση αυτή επιστρέφει 1. Σε διαφορετική περίπτωση επιστρέφει 0.
- ❖ deleteApplicationForm(userID:integer,excursionID:integer): Η συνάρτηση αυτή καλεί την αντίστοιχη συνάρτηση διαγραφής μίας αίτησης συμμετοχής του αντικειμένου ApplicationFormDBProxy. Αυτό το κάνει δίνοντας το ID του χρήστη και της επιθυμητής εκδρομής.
- ❖ cancel(): Καλείται με την πίεση του κουμπιού cancel και επιστρέφει στην σελίδα της εκδρομής.
- ❖ back(): Πρόκειται για τη συνάρτηση που καλείται με το πάτημα του κουμπιού back για έξοδο από την διεπαφή λάθους και επιστροφή στη σελίδα της αίτησης.

#### Controller EvaluateApplicationFormControl

«Controller» EvaluateApplicationFormControl	
-uploaderID: integer -excursionID: integer -applicationFormID: integer	
+EvaluateApplicationFormControl(uplID, excID, applID): void +approve(applicationFormID: integer): integer +decline(applicationFormID: integer): void +getNextApplication(applicationID: integer): void	

Ο ελεγκτής αυτός χρησιμοποιείται για τον έλεγχο της αξιολόγησης των αιτήσεων.

#### Χαρακτηριστικά της κλάσης:

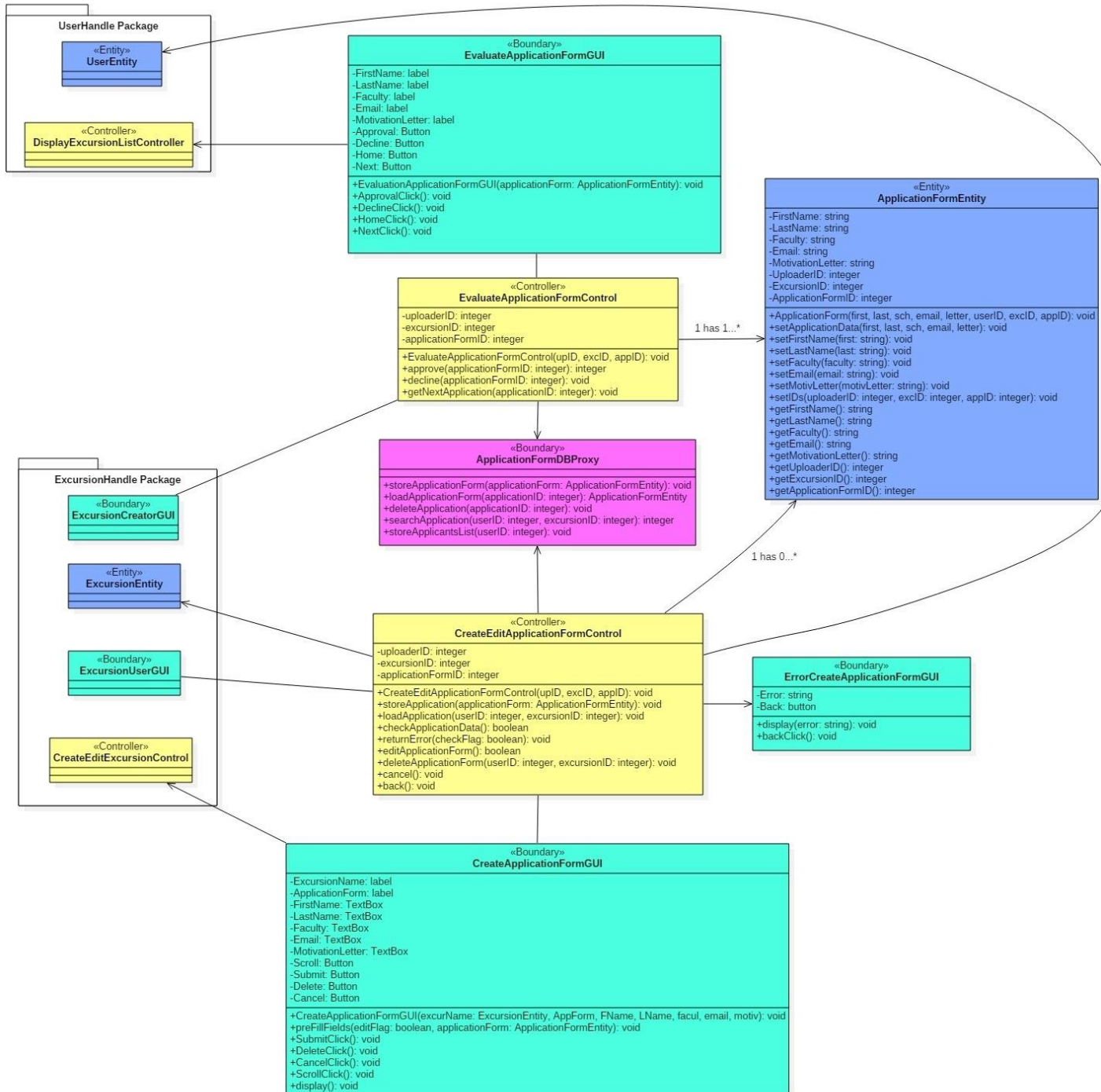
- ❖ uploaderID: Πρόκειται για το μοναδικό αναγνωριστικό του χρήστη που υποβάλλει την αίτηση συμμετοχής.
- ❖ excursionID: Πρόκειται για το μοναδικό αναγνωριστικό της εκδρομής
- ❖ applicationFormID: Πρόκειται για το μοναδικό αναγνωριστικό της αίτησης συμμετοχής.

#### Μέθοδοι της κλάσης:

- ❖ EvaluateApplicationFormControl(uplID,excID,applID): Η συνάρτηση δόμησης του ελεγκτή αυτού.
- ❖ approve(applicationFormID: integer): Η συνάρτηση αυτή καλείται για την αποδοχή μίας αίτησης συμμετοχής με το approvalClick(). Έπειτα φορτώνεται η επόμενη αίτηση συμμετοχής.
- ❖ decline(applicationFormID: integer): Η συνάρτηση αυτή καλείται για την απόρριψη μίας αίτησης συμμετοχής με το DeclineClick(). Έπειτα φορτώνεται η επόμενη αίτηση συμμετοχής. Επιπλέον, σε αυτή την λειτουργία η αίτηση διαγράφεται, μέσω κατάλληλης συνάρτησης της βάσης δεδομένων.
- ❖ getNextApplication(applicationID: integer): Η συνάρτηση αυτή φορτώνει την επόμενη αίτηση συμμετοχής. Η συνάρτηση αυτή ενεργοποιείται με το πάτημα του NextClick.

**ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ:**

Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.

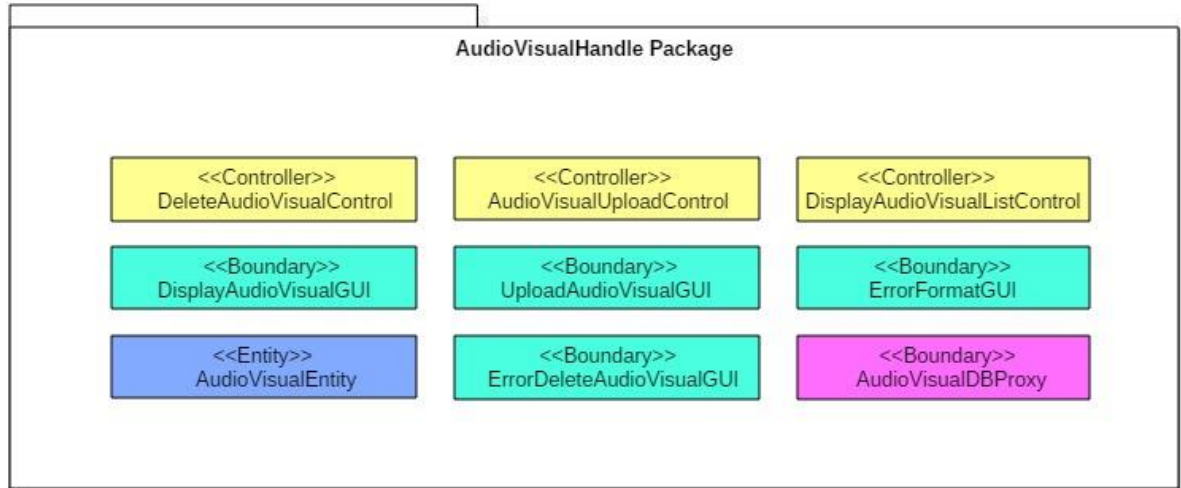


Σχήμα 4. Διάγραμμα κλάσεων πακέτου ApplicationFormHandle

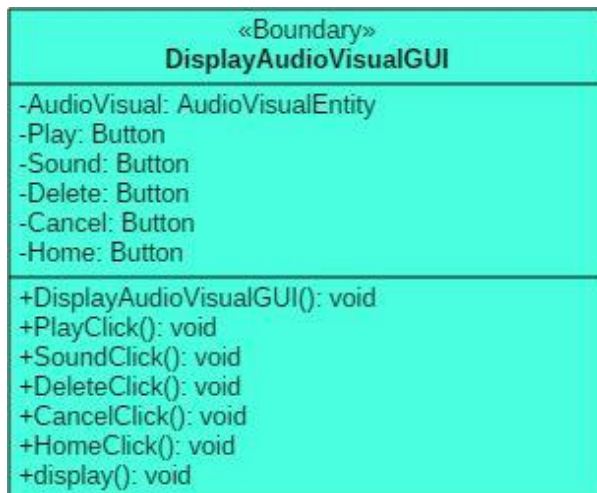
## 2.3 Πακέτο λεξιλογίου σεναρίων χαμηλής προτεραιότητας

### 2.3.1 Πακέτο AudioVisualHandle

Σκοπός του πακέτου αυτού είναι η διαχείριση του οπτικοακουστικού υλικού μιας εκδρομής. Οι κλάσεις που το απαρτίζουν μεριμνούν για την προβολή, το ανέβασμα και τη διαγραφή αρχείων οπτικοακουστικού υλικού.



#### Boundary DisplayAudioVisualGUI



Η κλάση αυτή αντιπροσωπεύει το γραφικό περιβάλλον διεπαφής για την προβολή οπτικοακουστικού υλικού στο χρήστη.

#### Χαρακτηριστικά της κλάσης:

- ❖ AudioVisual: Αρχείο εικόνας/βίντεο τύπου AudioVisualEntity
- ❖ Play: Κουμπί για την αναπαραγωγή αρχείου βίντεο
- ❖ Sound: Κουμπί για τη ρύθμιση ήχου αρχείου βίντεο
- ❖ Delete: Κουμπί για τη διαγραφή αρχείου εικόνας/βίντεο
- ❖ Cancel: Κουμπί για την επιστροφή στη σελίδα προβολής εκδρομής
- ❖ Home: Κουμπί για την επιστροφή στην αρχική σελίδα της εφαρμογής



Μέθοδοι της κλάσης:

- ❖ DisplayAudioVisualGUI(): Αποτελεί μέθοδο δόμησης της οριακής κλάσης.
- ❖ PlayClick(): Η μέθοδος ενεργοποιεί την αναπαραγωγή ενός αρχείου βίντεο.
- ❖ SoundClick(): Η μέθοδος ρυθμίζει την ένταση ήχου ενός αρχείου βίντεο.
- ❖ DeleteClick(): Καλεί τη συνάρτηση deleteAudioVisual() του ελεγκτή DeleteAudioVisualControl για τη διαγραφή του αρχείου οπτικοακουστικού υλικού από την AudioVisualDBProxy. Έπειτα, καλεί τη συνάρτηση loadExcursion() του ελεγκτή CreateEditExcursionControl για προβολή της σελίδας της εκδρομής.
- ❖ CancelClick(): Καλεί τη συνάρτηση loadExcursion() του ελεγκτή CreateEditExcursionControl για προβολή της εκδρομής στην οποία έχει προστεθεί το αρχείο και είναι αποθηκευμένη στην ExcursionDBProxy.
- ❖ HomeClick(): Καλεί τη συνάρτηση displayHomepage() του ελεγκτή DisplayExcursionListControl για προβολή της αρχικής σελίδας της εφαρμογής.
- ❖ display(): Η μέθοδος αυτή είναι υπεύθυνη για την προβολή των περιεχομένων της οριακής κλάσης.

Boundary ErrorDeleteAudioVisualGUI

«Boundary» ErrorDeleteAudioVisualGUI	
-Error: string	
-Back: button	
+displayError(error: string): void	
+backClick(): void	

Η κλάση αυτή αποτελεί το γραφικό περιβάλλον διεπαφής για την εμφάνιση μηνύματος σφάλματος προς το χρήστη σε περίπτωση που δεν έχει δικαίωμα διαγραφής του αρχείου.

Χαρακτηριστικά της κλάσης:

- ❖ Error: Μήνυμα σφάλματος
- ❖ Back: Κουμπί για την επιστροφή από τη σελίδα εμφάνισης σφάλματος

Μέθοδοι της κλάσης:

- ❖ displayError(error: string): Εμφανίζει το μήνυμα σφάλματος που της παρέχεται από τη συνάρτηση returnError() του ελεγκτή DeleteAudioVisualControl.
- ❖ backClick(): Η μέθοδος αυτή καλεί τη συνάρτηση back() του ελεγκτή DeleteAudioVisualControl.

### **Boundary UploadAudioVisualGUI**

«Boundary» UploadAudioVisualGUI
-AudioVisual: TextBox -Upload: Button -Cancel: Button -Browse: Button
+UploadClick(): void +CancelClick(): void +BrowseClick(): void +display(): void

Η κλάση αυτή αντιπροσωπεύει το γραφικό περιβάλλον διεπαφής για το ανέβασμα αρχείων οπτικοακουστικού υλικού σε μια εκδρομή από το χρήστη.

#### **Χαρακτηριστικά της κλάσης:**

- ❖ AudioVisual: Πεδίο προς συμπλήρωση ονόματος επιθυμητού αρχείου εικόνας/βίντεο
- ❖ Upload: Κουμπί για το ανέβασμα επιλεγμένου αρχείου εικόνας/βίντεο
- ❖ Cancel: Κουμπί για την επιστροφή στη σελίδα προβολής εκδρομής
- ❖ Browse: Κουμπί για την επιλογή αρχείου εικόνας/βίντεο από το σύστημα του χρήστη

#### **Μέθοδοι της κλάσης:**

- ❖ UploadClick(): Καλεί τη συνάρτηση storeAudioVisual() του ελεγκτή AudioVisualUploadControl για την αποθήκευση του επιλεγμένου αρχείου στην AudioVisualDBProxy. Έπειτα, καλεί τη συνάρτηση loadExcursion() του ελεγκτή CreateEditExcursionControl για προβολή της σελίδας της εκδρομής.
- ❖ CancelClick(): Καλεί τη συνάρτηση loadExcursion() του ελεγκτή CreateEditExcursionControl για προβολή της εκδρομής στην οποία έχει προστεθεί το αρχείο και είναι αποθηκευμένη στην ExcursionDBProxy.
- ❖ BrowseClick(): Ενεργοποιεί παράθυρο διαλόγου για την εύρεση αρχείου εικόνας/βίντεο από το σύστημα του χρήστη.
- ❖ display(): Η συνάρτηση αυτή είναι υπεύθυνη για την εμφάνιση της σελίδας της οριακής κλάσης.

### **Boundary ErrorFormat**

«Boundary» ErrorFormatGUI
-Error: string -Back: button
+display(error: string): void +backClick(): void

Η κλάση αυτή αποτελεί το γραφικό περιβάλλον διεπαφής για την εμφάνιση μηνύματος σφάλματος προς το χρήστη σε περίπτωση που δεν έχει επιλέξει σωστό τύπο αρχείου.

#### **Χαρακτηριστικά της κλάσης:**

- ❖ Error: Μήνυμα σφάλματος
- ❖ Back: Κουμπί για την επιστροφή από τη σελίδα εμφάνισης σφάλματος

#### Μέθοδοι της κλάσης:

- ❖ `displayError(error: string)`: Εμφανίζει το μήνυμα σφάλματος που της παρέχεται από τη συνάρτηση `returnError()` του ελεγκτή `AudioVisualUploadControl`.
- ❖ `backClick()`: Η μέθοδος αυτή καλεί τη συνάρτηση `back()` του ελεγκτή `AudioVisualUploadControl`.

#### Boundary AudioVisualDBProxy

«Boundary» AudioVisualDBProxy
<code>+storeAudioVisual(audioVisual: AudioVisualEntity): void</code> <code>+loadAudioVisual(audioVisualID: integer): AudioVisualEntity</code> <code>+deleteAudioVisual(audioVisualID: integer): void</code> <code>+searchAudioVisual(ExcursionID: integer): integer</code>

Η κλάση αυτή επικοινωνεί με απομακρυσμένη βάση δεδομένων όπου είναι αποθηκευμένο το οπτικοακουστικό υλικό των εκδρομών.

#### Μέθοδοι της κλάσης:

- ❖ `storeAudioVisual(audioVisual: AudioVisualEntity)`: Η μέθοδος αυτή αποθηκεύει αντικείμενα τύπου `AudioVisualEntity` στην απομακρυσμένη βάση δεδομένων.
- ❖ `loadAudioVisual(audioVisualID: integer)`: Η μέθοδος επιστρέφει αντικείμενα τύπου `AudioVisualEntity`, βάσει του μοναδικού κωδικού τους.
- ❖ `deleteAudioVisual(audioVisualID: integer)`: Η μέθοδος διαγράφει αντικείμενα τύπου `AudioVisualEntity`, βάσει του μοναδικού κωδικού τους.
- ❖ `searchAudioVisual(ExcursionID: integer)`: Η μέθοδος αυτή επιστρέφει το μοναδικό κωδικό ενός αντικειμένου τύπου `AudioVisualEntity` που ανήκει σε μια εκδρομή.

#### Entity AudioVisualEntity

«Entity» AudioVisualEntity
<code>-ImageVideo: ImageVideoType</code> <code>-AudioVisualID: integer</code> <code>-UploaderID: integer</code>
<code>+AudioVisualEntity(imageVideo: ImageVideoType): void</code> <code>+setImageVideo(imVd: ImageVideoType): void</code> <code>+setAudioVisualID(audVisID: integer): void</code> <code>+setUploaderID(uploaderID: integer): void</code> <code>+getImageVideo(): ImageVideoType</code> <code>+getAudioVisualID(): integer</code> <code>+getUploaderID(): integer</code>

Πρόκειται για την κλάση οντότητας οπτικοακουστικού υλικού, που περιλαμβάνεται στη σελίδα προβολής περιεχομένων κάθε εκδρομής.

Χαρακτηριστικά της κλάσης:

- ❖ ImageVideo: Αρχείο εικόνας/βίντεο τύπου ImageVideoType
- ❖ AudioVisualID: Ο μοναδικός κωδικός του αρχείου εικόνας/βίντεο
- ❖ UploaderID: Ο μοναδικός κωδικός του χρήστη που ανέβασε το αρχείο

Μέθοδοι της κλάσης:

- ❖ AudioVisualEntity(imageVideo: ImageVideoType): Πρόκειται για μέθοδο δόμησης της οντότητας.
- ❖ setImageVideo(imVd: ImageVideoType): Μέθοδος που θέτει το αρχείο οπτικοακουστικού υλικού.
- ❖ setAudioVisualID(audVisID: integer): Μέθοδος που θέτει το μοναδικό αναγνωριστικό του αρχείου οπτικοακουστικού υλικού.
- ❖ setUploaderID(uploaderID: integer): Μέθοδος που θέτει το μοναδικό αναγνωριστικό του χρήστη που ανέβασε το αρχείο οπτικοακουστικού υλικού.
- ❖ getImageVideo(): Η μέθοδος αυτή επιστρέφει αντικείμενα τύπου ImageVideoType, δηλαδή αρχεία οπτικοακουστικού υλικού.
- ❖ getAudioVisualID(): Η μέθοδος αυτή επιστρέφει το μοναδικό αναγνωριστικό ενός αρχείου οπτικοακουστικού υλικού.
- ❖ getUploaderID(): Η μέθοδος αυτή επιστρέφει το μοναδικό αναγνωριστικό του χρήστη που ανέβασε το αρχείο οπτικοακουστικού υλικού.

Controller DisplayAudioVisualControl

«Controller» DisplayAudioVisualControl
+List: AudioVisual -audioVisualID: integer -uploaderID: integer
+DisplayAudioVisualControl(aviD, upID): void +loadAudioVisual(ExcursionID: integer): AudioVisualEntity +DisplayAudioVisualListControl(excursionID: integer): void +DisplayOneAudioVisual(audioVisualID: integer): void

Η κλάση αυτή αποτελεί ελεγκτή για την προβολή της λίστας του διαθέσιμου οπτικοακουστικού υλικού κάθε εκδρομής.

Χαρακτηριστικά της κλάσης:

- ❖ List: Λίστα με τα αρχεία οπτικοακουστικού υλικού
- ❖ audioVisualID: Μοναδικό αναγνωριστικό αρχείου εικόνας/βίντεο
- ❖ uploaderID: Μοναδικό αναγνωριστικό χρήστη που ανέβασε το αρχείο

Μέθοδοι της κλάσης:

- ❖ DisplayAudioVisualControl(avid,upID): Μέθοδος δόμησης του ελεγκτή.



- ❖ `loadAudioVisual(ExcursionID: integer)`: Η συγκεκριμένη συνάρτηση επιστρέφει αντικείμενα οπτικοακουστικού υλικού, δηλαδή τύπου `AudioVisualEntity`, σύμφωνα με το μοναδικό κωδικό εκδρομής που λαμβάνει ως όρισμα. Αυτό γίνεται με την κλήση της αντίστοιχης συνάρτησης φόρτωσης που διαθέτει το αντικείμενο `AudioVisualDBProxy`. Καλείται για τη φόρτωση όλων των αρχείων εικόνας/βίντεο μιας εκδρομής από την `AudioVisualDBProxy`.
- ❖ `DisplayAudioVisualListControl(excursionID: integer)`: Η συνάρτηση αυτή καλείται για την προβολή της λίστας του διαθέσιμου οπτικοακουστικού υλικού μιας εκδρομής. Η λίστα είναι αποθηκευμένη στην `AudioVisualDBProxy`, βάσει του μοναδικού κωδικού κάθε εκδρομής.
- ❖ `DisplayOneAudioVisual(audioVisualID: integer)`: Η συγκεκριμένη συνάρτηση ελέγχει κάθε φορά την προβολή ενός από τα αρχεία οπτικοακουστικού υλικού μιας εκδρομής. Καλεί τη συνάρτηση `display()` της οριακής κλάσης `DisplayAudioVisualGUI`.

#### Controller AudioVisualUploadControl

«Controller» AudioVisualUploadControl	
-audioVisualID: integer	
-uploaderID: integer	
+AudioVisualUploadControl(avID, upID): void	
+checkType(): boolean	
+returnError(Typeflag: boolean): string	
+storeAudioVisual(audiovisual: AudioVisualEntity): void	
+back(): void	

Η κλάση αυτή αποτελεί ελεγκτή για το ανέβασμα αρχείων οπτικοακουστικού υλικού σε μια εκδρομή.

#### Χαρακτηριστικά της κλάσης:

- ❖ `audioVisualID`: Μοναδικό αναγνωριστικό αρχείου εικόνας/βίντεο
- ❖ `uploaderID`: Μοναδικό αναγνωριστικό χρήστη που ανέβασε το αρχείο

#### Μέθοδοι της κλάσης:

- ❖ `AudioVisualUploadControl(avID, upID)`: Μέθοδος δόμησης του ελεγκτή.
- ❖ `checkType()`: Η μέθοδος ελέγχει τη συμβατότητα του τύπου του αρχείου που επιλέγει ο χρήστης με τα αρχεία τύπου `ImageVideoType` που υποστηρίζει το σύστημα. Σε περίπτωση συμβατότητας επιστρέφει τη λογική τιμή `true`, ειδάλλως επιστρέφει τη λογική τιμή `false`.
- ❖ `returnError(Typeflag: boolean)`: Η μέθοδος αυτή παρέχει το κατάλληλο `string` στην κλάση `ErrorFormat`. Σε περίπτωση συμβατότητας του επιλεγμένου αρχείου η επιστρεφόμενη μεταβλητή έχει την τιμή `NULL`, ειδάλλως περιέχει το μήνυμα σφάλματος που πρόκειται να εμφανιστεί.
- ❖ `storeAudioVisual(audioVisual: AudioVisualEntity)`: Η συνάρτηση αυτή καλείται για την αποθήκευση του επιλεγμένου αρχείου οπτικοακουστικού υλικού στην `AudioVisualDBProxy`, βάσει του μοναδικού κωδικού της εκδρομής.
- ❖ `back()`: Η συνάρτηση αυτή πραγματοποιεί επιστροφή από τη σελίδα εμφάνισης σφάλματος στη σελίδα `UploadAudioVisualGUI`.

### Controller DeleteAudioVisualControl

«Controller» DeleteAudioVisualControl
-audioVisualID: integer -uploaderID: integer -userID: integer
+DeleteAudioVisualControl(avID, upID, userID): void +deleteAudioVisual(audioVisualID: integer): void +returnError(uploaderFlag: boolean): string +checkUploader(uploaderID: integer, userID: integer): boolean +back(): void

Η κλάση αυτή αποτελεί ελεγκτή για τη διαγραφή αρχείων οπτικοακουστικού υλικού σε μια εκδρομή.

#### Χαρακτηριστικά της κλάσης:

- ❖ audioVisualID: Μοναδικό αναγνωριστικό αρχείου εικόνας/βίντεο
- ❖ uploaderID: Μοναδικό αναγνωριστικό χρήστη που ανέβασε το αρχείο
- ❖ userID: Μοναδικό αναγνωριστικό χρήστη που επιθυμεί να διαγράψει το αρχείο

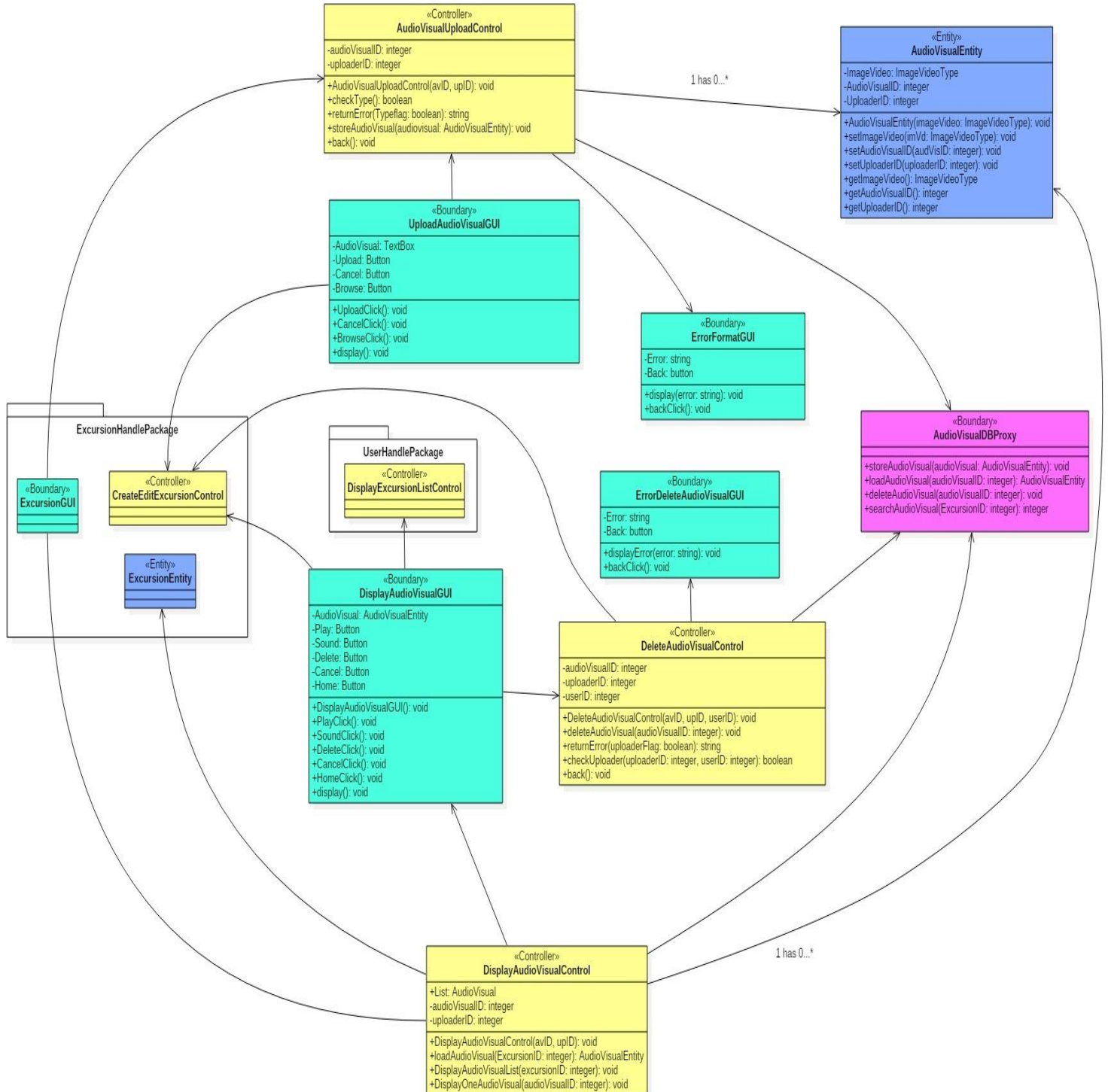
#### Μέθοδοι της κλάσης:

- ❖ DeleteAudioVisualControl(avid,upID,userID): Μέθοδος δόμησης του ελεγκτή.
- ❖ deleteAudioVisual(audioVisualID: integer): Η μέθοδος ελέγχει τη διαγραφή του επιλεγμένου αρχείου οπτικοακουστικού υλικού, που φέρει το audioVisualID που της δίνεται ως όρισμα.
- ❖ returnError(uploaderFlag: boolean): Η μέθοδος αυτή παρέχει το κατάλληλο string στην κλάση ErrorDeleteAudioVisual. Σε περίπτωση που ο χρήστης έχει δικαίωμα διαγραφής του αρχείου η επιστρεφόμενη μεταβλητή έχει την τιμή NULL, ειδάλλως περιέχει το μήνυμα σφάλματος που πρόκειται να εμφανιστεί.
- ❖ checkUploader(uploaderID: integer, userID: integer): Η συνάρτηση αυτή ελέγχει εάν ο χρήστης που ανέβασε το αρχείο ταυτίζεται με το χρήστη που επιθυμεί να το διαγράψει. Στην περίπτωση αυτή επιστρέφει τη λογική τιμή true, ενώ σε αντίθετη περίπτωση επιστρέφει τη λογική τιμή false.
- ❖ back(): Η συνάρτηση αυτή πραγματοποιεί επιστροφή από τη σελίδα εμφάνισης σφάλματος στη σελίδα DisplayAudioVisualGUI.



**ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ:**

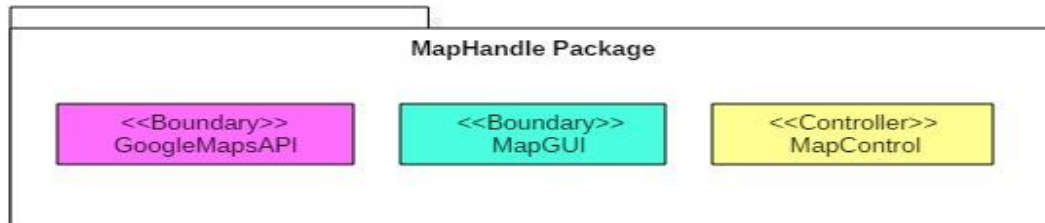
Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.



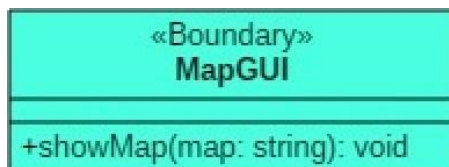
Σχήμα 5. Διάγραμμα κλάσεων πακέτου AudioVisualHandle

## 2.3.2 Πακέτο MapHandle

Η λειτουργικότητα του πακέτου αυτού περιορίζεται στη σύνδεση της εφαρμογής με την εφαρμογή προβολής χαρτών GoogleMaps, με σκοπό την προβολή της τοποθεσίας διεξαγωγής μιας εκδήλωσης.



### Boundary MapGUI

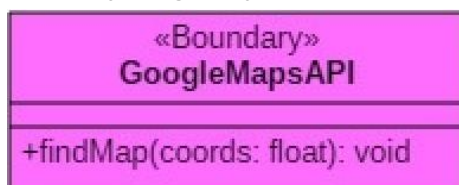


Η κλάση αυτή αποτελεί τη διεπαφή για την προβολή του χάρτη στο χρήστη.

#### Μέθοδος της κλάσης:

- ❖ showMap(map: string): Η μέθοδος αυτή προβάλλει το χάρτη που λαμβάνει από τη συνάρτηση showMap() του ελεγκτή MapControl.

### Boundary GoogleMapsAPI



Η κλάση αυτή επικοινωνεί με την εφαρμογή GoogleMaps με σκοπό την προβολή του χάρτη.

#### Μέθοδος της κλάσης:

- ❖ findMap(coords: float): Η μέθοδος αυτή δέχεται ως ορίσματα τις συντεταγμένες της εκδήλωσης και επιστρέφει την τοποθεσία διεξαγωγής της στο χάρτη.

**Controller MapController**

«Controller» <b>MapControl</b>
-EventID: integer
+MapControl(eventID): void +getCoordinates(EventID: int): float +showMap(map: string): void +getMap(coordinates: float): void

Η κλάση αυτή ελέγχει τη διαδικασία της προβολής του χάρτη στο χρήστη, μεταβιβάζοντας τις συντεταγμένες και τον χάρτη στις αντίστοιχες κλάσεις.

**Χαρακτηριστικά της κλάσης:**

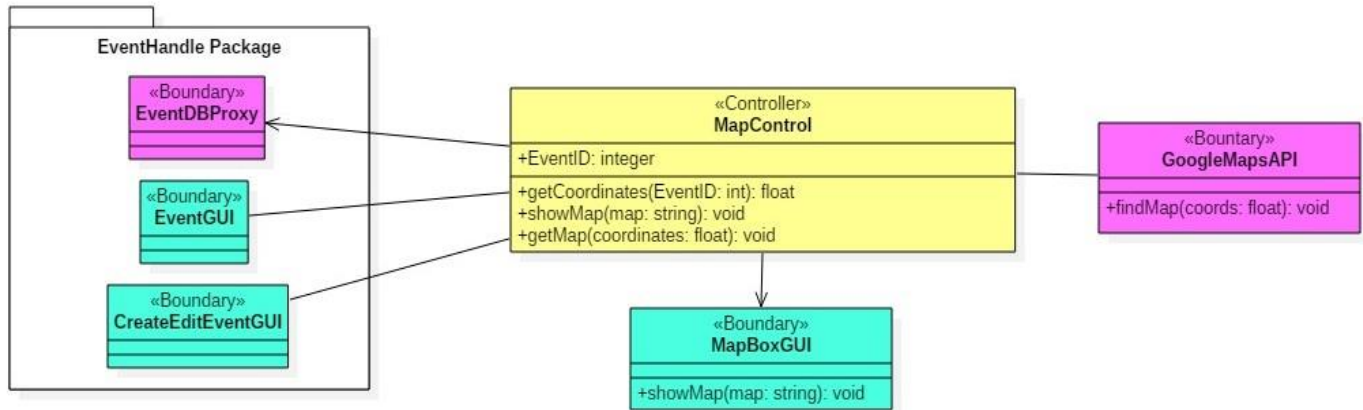
- ❖ EventID: Η μέθοδος αυτή δέχεται ως ορίσματα τις συντεταγμένες της εκδήλωσης και επιστρέφει την τοποθεσία διεξαγωγής της στο χάρτη.

**Μέθοδος της κλάσης:**

- ❖ MapControl(eventID: integer): Πρόκειται για συνάρτηση δόμησης του ελεγκτή.
- ❖ getCoordinates(EventID: integer): Η μέθοδος αυτή επιστρέφει τις συντεταγμένες της εκδήλωσης, η οποία φέρει το μοναδικό αναγνωριστικό που δίνεται ως όρισμα.
- ❖ showMap(map: string): Η μέθοδος αυτή καλεί την οριακή κλάση MapGUI για την προβολή του χάρτη στο χρήστη.
- ❖ getMap(coordinates: float): Η μέθοδος αυτή καλεί την οριακή κλάση GoogleMapsAPI για την εύρεση της τοποθεσίας διεξαγωγής της εκδήλωσης στο χάρτη.

### ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ

Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.



Σχήμα 6. Διάγραμμα κλάσεων πακέτου MapHandle