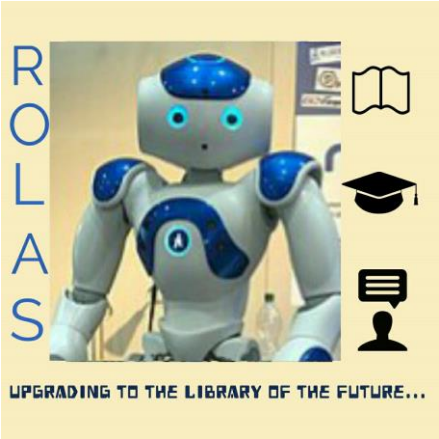
Τεχνολογία Λογισμικού
Τομέας Ηλεκτρονικής και Υπολογιστών
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Α.Π.Θ.

8ο Εξάμηνο
Άνοιξη 2018



ROboLibrarian Assistant System

Απαιτήσεις Χρηστών

Del.2.1

Version 2.2

Καλαϊτζής Γεώργιος kalaitzg@ece.auth.gr
Καμπελής Ελιέζερ Σολομών eliekamp@ece.auth.gr
Παπαδόπουλος Κωνσταντίνος konserpap@ece.auth.gr
Τοπαλίδης Ευθύμιος eatopalid@ece.auth.gr
Ομάδα 19
02/05/2018

Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Α. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers[♣] και του M. Smialek's.	0.1
Α. Συμεωνίδης	29/3/2014	Προσαρμογή του εγγράφου.	0.1.3

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Καλαϊτζής Γεώργιος	RO.L.A.S.	kalaitzg@ece.auth.gr
Καμπελής Ελιέζερ Σολομών	RO.L.A.S.	eliekamp@ece.auth.gr
Παπαδόπουλος Κωνσταντίνος	RO.L.A.S.	konserpap@ece.auth.gr
Τοπαλίδης Ευθύμιος	RO.L.A.S.	eatopalid@ece.auth.gr

*** ΕΡΓΑΣΤΗΡΙΑΚΗ ΟΜΑΔΑ 19**

Πίνακας Περιεχομένων

Εισαγωγικά

Στόχος του Εγγράφου

Σε αυτό το δεύτερο παραδοτέο απαιτήσεων λογισμικού περνάμε από το ΤΙ στο ΠΩΣ. Δηλαδή ενώ στο πρώτο παραδοτέο περιγράψαμε τα λειτουργικά και μη χαρακτηριστικά του συστήματος μας, τώρα σε αυτή τη φάση προχωρούμε στην υλοποίηση τους και τι είδους δεδομένα θα ανταλλάσσονται μεταξύ τους. Επίσης προβάλλεται το πως επικοινωνεί το σύστημά μας με τα εξωτερικά συστήματα που έχουν οριστεί αλλά και με τους διάφορους χρήστες. Πρόκειται για τη στατική φάση ανάπτυξης του συστήματος μας, όπου περιέχονται τα πακέτα κλάσεων που δίνουν φυσική υπόσταση στις λειτουργικές απαιτήσεις του συστήματος μας και τα διαγράμματα κλάσεων που δείχνουν πως αλληλοεπιδρούν οι διάφορες κλάσεις μεταξύ τους.

Κάθε πακέτο κλάσεων περιλαμβάνει κλάσεις που χωρίζονται στις εξής κατηγορίες: 1)Οριακές κλάσεις(Boundary) ή κλάσεις διεπαφής χρηστών , 2)Κλάσεις Οντοτήτων(Entities) και 3)Κλάσεις Ελέγχου(Controllers). Οι οριακές κλάσεις πραγματοποιούν την αλληλεπίδραση ανάμεσα στο χρήστη και στο σύστημα καθώς και την επικοινωνία του συστήματος με εξωτερικά συστήματα, όπως το ρομπότ NAO και την LibDatabase. Οι κλάσεις οντοτήτων περιέχουν τις ιδιότητες των αντικειμένων που κυριαρχούν στην περιγραφή των λειτουργικών απαιτήσεων του εγγράφου απαιτήσεων χρηστών. Τέλος οι κλάσεις ελέγχου έχουν ως σκοπό να συντονίσουν την συνεργασία μεταξύ των κλάσεων και να διασφαλίσουν την ομαλή λειτουργία του συστήματος.

Στο τέλος του εγγράφου καθορίζουμε ποια από τα βασικά πρότυπα σχεδίασης χρησιμοποιήσαμε ώστε να φτιάξουμε όσο το δυνατόν πιο αποδοτικό και ταυτόχρονα αξιόπιστο σύστημα που να ικανοποιεί και τις μη λειτουργικές απαιτήσεις που έχουν ορισθεί στο έγγραφο απαιτήσεων χρηστών.

Αναγνωστικό κοινό και τρόπος ανάγνωσης

Στόχος των διαγραμμάτων κλάσεων είναι η περιγραφή των στατικών χαρακτηριστικών του συστήματος.

Οι κυριότερες **κατηγορίες χρηστών** που χρησιμοποιούν τα διαγράμματα κλάσεων είναι:

– Οι **ειδικοί του πεδίου εφαρμογής** που χρησιμοποιούν τα διαγράμματα κλάσεων για να μοντελοποιήσουν το πεδίο εφαρμογής (συμπεριλαμβανομένων των ταξινομιών)

– Η **ομάδα ανάπτυξης** που χρησιμοποιεί τα διαγράμματα κλάσεων κατά τις φάσεις ανάλυσης, σχεδίασης του συστήματος, σχεδίασης των αντικειμένων, και προγραμματισμού.

Ανάλογα με τη δραστηριότητα ανάπτυξης, τα **μέλη της ομάδας ανάπτυξης** μπορούν να έχουν διαφορετικούς ρόλους:

– **Αναλυτή**

Ο αναλυτής ενδιαφέρεται για κλάσεις εφαρμογής: οι συσχετίσεις ανάμεσα στις κλάσεις είναι σχέσεις ανάμεσα στις αφαιρέσεις στο πεδίο εφαρμογής.

Επιδιώκει να εκμεταλλευτεί την κληρονομικότητα για να αναπαραστήσει τις ταξινομίες στο πεδίο εφαρμογής.

– **Σχεδιαστής Συστήματος**

Ο σχεδιαστής επικεντρώνεται στη λύση του προβλήματος, στη λύση του πεδίου εφαρμογής.

Η σχεδίαση αποτελείται από πολλές εργασίες (αποσύνθεση συστήματος, επιλογή της κατάλληλης πλατφόρμας υλικού, επιλογή συστήματος διαχείρισης δεδομένων, κτλ.)

– **Σχεδιαστής Αντικειμένων**

Ο σχεδιαστής αντικειμένων επικεντρώνεται στη δημιουργία των μόνιμων οντοτήτων του συστήματος.

– **Προγραμματιστής**

Ο προγραμματιστής είναι υπεύθυνος για την υλοποίηση σε κώδικα του τελικού έργου λογισμικού.

1.4 Σκοπός του Έργου

Οι βιβλιοθήκες αποτελούν χώρους πολιτισμού, διαφύλαξης και διάδοσης της γνώσης. Από μικρές βιβλιοθήκες σε δημοτικά σχολεία μέχρι μεγάλες πανεπιστημιακές (κρατικές ή δημοτικές) βιβλιοθήκες, τα συγκεκριμένα ιδρύματα έχουν μεγάλη σημασία για την παιδεία.

Η εισαγωγή λοιπόν της τεχνολογίας σε αυτές συνιστά μία αναγκαία και αρκετά σημαντική πρωτοβουλία.

Η εφαρμογή της ομάδας μας αποσκοπεί στη δημιουργία ενός έξυπνου βοηθού που θα συμπληρώσει και θα δώσει ένα νέο ρόλο σε αυτόν της/του βιβλιοθηκάριου. Ο NAO RoboLibrarian μεταξύ άλλων θα καθοδηγεί τον επισκέπτη στο κατάλληλο ράφι, θα αναλαμβάνει οργανωτικό ρόλο π.χ. ταξινομώντας τα βιβλία, καθώς και πιο σύνθετες λειτουργίες όπως το να προτείνει λίστες ανάγνωσης με βάση τις προτιμήσεις του αναγνώστη.

Παράρτημα I - Πίνακας Ιχνηλασιμότητας

- Δεν πραγματοποιήθηκαν αλλαγές που να επηρεάζουν το προηγούμενο παραδοτέο.

Παράρτημα II – Ανοιχτά Θέματα

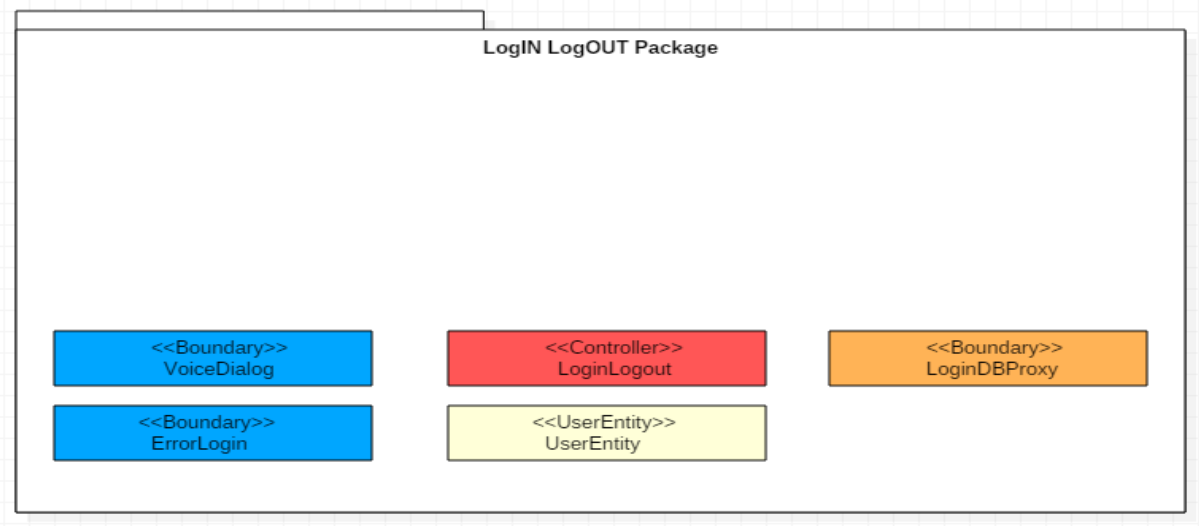
- Η τιμή των 3 δευτερολέπτων απόκρισης του συστήματος μετά από την φωνητική εντολή του χρήστη, μπορεί να μεταβληθεί και να βελτιωθεί (όπως σημειώθηκε και στην παράγραφο 2.1) εάν ο δρομολογητής του δικτύου αναβαθμιστεί.

1 Στατική Μοντελοποίηση

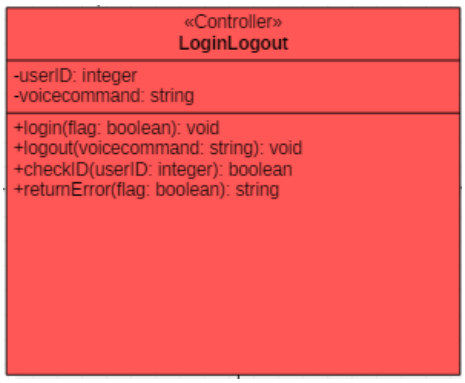
1.1Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας

1.1.1 LogIN LogOUT Package

Το πακέτο αυτό αναφέρεται στις επιλογές login και logout του εγγεγραμμένου χρήστη. Παρακάτω παρουσιάζονται οι κλάσεις που χρησιμοποιούνται για την υλοποίηση αυτού του πακέτου.



Controller LoginLogout



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

Χαρακτηριστικά της κλάσης:

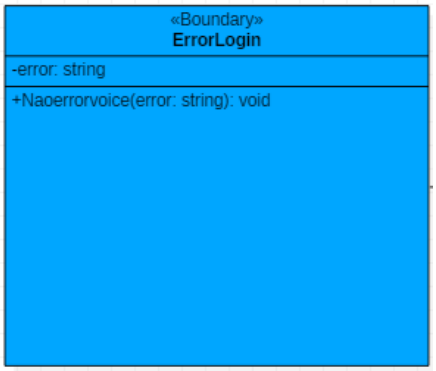
- userID : Χαρακτηριστικός αριθμός του κάθε εγγεγραμμένου χρήστη
- voicecommand : Η ιδιότητα αυτή περιέχει την φωνητική εντολή που εκφωνεί το σύστημα όταν ο εγγεγραμμένος χρήστης αποφασίσει να αποσυνδεθεί από το σύστημα.

Μέθοδοι της κλάσης:

- login(flag:boolean) : Η μέθοδος αυτή συνδέει τον εγγεγραμμένο χρήστη στο σύστημα της εφόσον η μεταβλητή flag είναι αληθής.
- logout(voicecommand:string) : Η μέθοδος αυτή αποσυνδέει τον εγγεγραμμένο χρήστη από το σύστημα και εκφωνεί προς αυτόν συγκεκριμένο μήνυμα μετά την κλήση της κλάσης VoiceDialog μέσω της μεθόδου logoutmethod().

- checkID(userID:integer) : Η μέθοδος αυτή ελέγχει αν τα στοιχεία ελέγχει αν τα στοιχεία που έδωσε ο χρήστης είναι ορθά. Σε περίπτωση που τα στοιχεία είναι ορθά καλείται η συνάρτηση login() σε κάθε άλλη περίπτωση καλείται η συνάρτηση returnError().
- returnError(flag:Boolean) : Η συνάρτηση αυτή επιστρέφει μήνυμα σφάλματος εύρεσης των στοιχείων του εγγεγραμμένου χρήστη στην Naoerrorvoice() της οριακής κλάσης ErrorLogin.Η διαδικασία αυτή εκτελείται εφόσον η flag είναι ψευδής.

Boundary ErrorLogin



Η οριακή αυτή κλάση ενεργοποιείται στην περίπτωση όπου ο χρήστης του συστήματος εισάγει στο σύστημα μη έγκυρο χαρακτηριστικό αριθμό εγγεγραμμένου χρήστη. Σε αυτή την περίπτωση μέσω της μεθόδου Naoerrorvoice() εκφωνείτε από το σύστημα σχετικό μήνυμα.

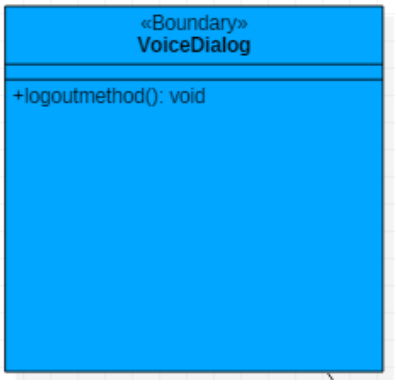
Χαρακτηριστικά της κλάσης:

- error : Η ιδιότητα αυτή εμπεριέχει το σχετικό μήνυμα που θα εκφωνηθεί από το σύστημα σε περίπτωση λάθους χαρακτηριστικού αριθμού χρήστη.

Μέθοδοι της κλάσης:

- Naoerrorvoice(error:string) : Η συνάρτηση αυτή εκφωνεί το σχετικό μήνυμα λάθους. Η λειτουργία της πυροδοτείτε από την returnError() (που ανήκει στην κλάση LoginLogout).

Boundary VoiceDialog

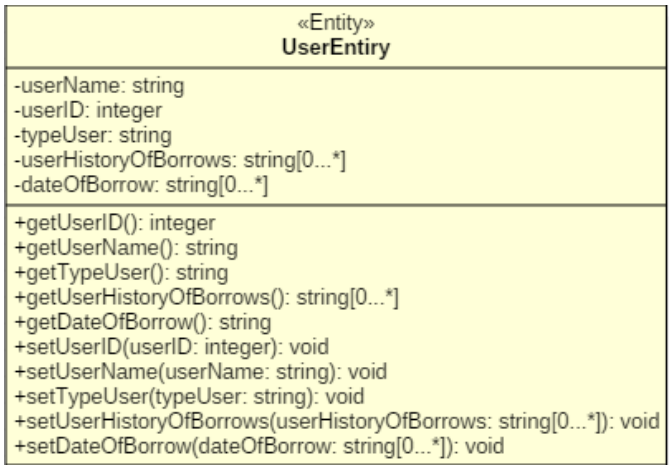


Η οριακή αυτή κλάση ενεργοποιείται στην περίπτωση όπου ο χρήστης του συστήματος ζητήσει να αποσυνδεθεί από το σύστημα. Ουσιαστικά η κλάση αυτή καλείται-ενεργοποιείται από την μέθοδο logout(voicecommand:string) που ανήκει στον Controller “LoginLogout”.

Μέθοδοι της κλάσης:

- logoutmethod(): Η μέθοδος αυτή καλείται από τον controller “LoginLogout” και εκφωνεί σχετικό μήνυμα για την αποσύνδεση του «εγγεγραμμένου χρήστη» από το σύστημα.

Entity UserEntity



Αυτή είναι η κλάση οντότητας του εγγεγραμμένου χρήστη που περιέχει και τα στοιχεία ορισμού του

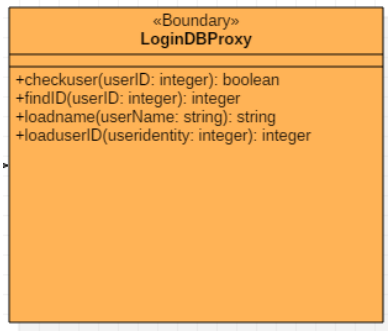
Χαρακτηριστικά της κλάσης:

- userName : Το όνομα του εγγεγραμμένου χρήστη
- userID : Χαρακτηριστικός αριθμός του κάθε εγγεγραμμένου χρήστη
- typeUser : Τύπος χρήστη εγγεγραμμένος ή μη στο σύστημα της ηλεκτρονικής βιβλιοθήκης
- userHistoryOfBorrows: Η μεταβλητή αυτή παίρνει την τιμή 1 (αληθής) όταν ο εγγεγραμμένος χρήστης έχει δανειστεί ενώ παίρνει την τιμή 0 όταν δεν έχει δανειστεί κάποιο βιβλίο.
- dateOfBorrow : Ημερομηνία δανεισμού βιβλίου από τον εγγεγραμμένο χρήστη

Μέθοδοι της κλάσης:

- setUserID(userID:integer): Η συνάρτηση αυτή θέτει τον χαρακτηριστικό αριθμό χρήστη σύμφωνα με το όρισμα που δέχεται.
- setUserName(userName:string): Η συνάρτηση αυτή θέτει το όνομα χρήστη σύμφωνα με την συμβολοσειρά που λαμβάνει ως όρισμα.
- setTypeUser(typeUser:string): Η συνάρτηση αυτή θέτει τον τύπο του χρήστη που αλληλεπιδρά με το σύστημα (εγγεγραμμένος ή μη)
- setUserHistoryOfBorrows(userHistoryOfBorrows:boolean): Η συνάρτηση αυτή ανάλογα με την τιμή της εισόδου που δέχεται, ανιχνεύει αν ο εγγεγραμμένος χρήστης έχει δανειστεί κάποιο βιβλίο (userHistoryOfBorrows=1) και επιστρέφει την λίστα των δανεισμών του ή σε περίπτωση που δεν έχει δανειστεί κάποιο βιβλίο επιστρέφει ανάλογο μήνυμα.
- setDateOfBorrow(dateOfBorrow:string): Η συνάρτηση αυτή θέτει την ημερομηνία δανεισμού του βιβλίου σύμφωνα με το όρισμα που δέχεται ως είσοδο.
- getUserID() : Η συνάρτηση αυτή επιστρέφει τον χαρακτηριστικό αριθμό του εγγεγραμμένου χρήστη.
- getUserName(): Η συνάρτηση αυτή επιστρέφει το όνομα του εγγεγραμμένου χρήστη
- getTypeUser(): Η συνάρτηση αυτή επιστρέφει τον τύπο του χρήστη που αλληλοεπιδρά με το σύστημα.
- getUserHistoryOfBorrows(): Η συνάρτηση αυτή επιστρέφει το ιστορικό δανεισμού βιβλίων του εγγεγραμμένου χρήστη εφόσον ο εγγεγραμμένος χρήστης έχει δανειστεί βιβλία(δηλαδή η μεταβλητή userHistoryOfBorrows έχει τιμή 1) .
- getDateOfBorrow(): Η συνάρτηση αυτή επιστρέφει την ημερομηνία δανεισμού των βιβλίων από τον εγγεγραμμένο χρήστη.

Boundary LoginDBProxy

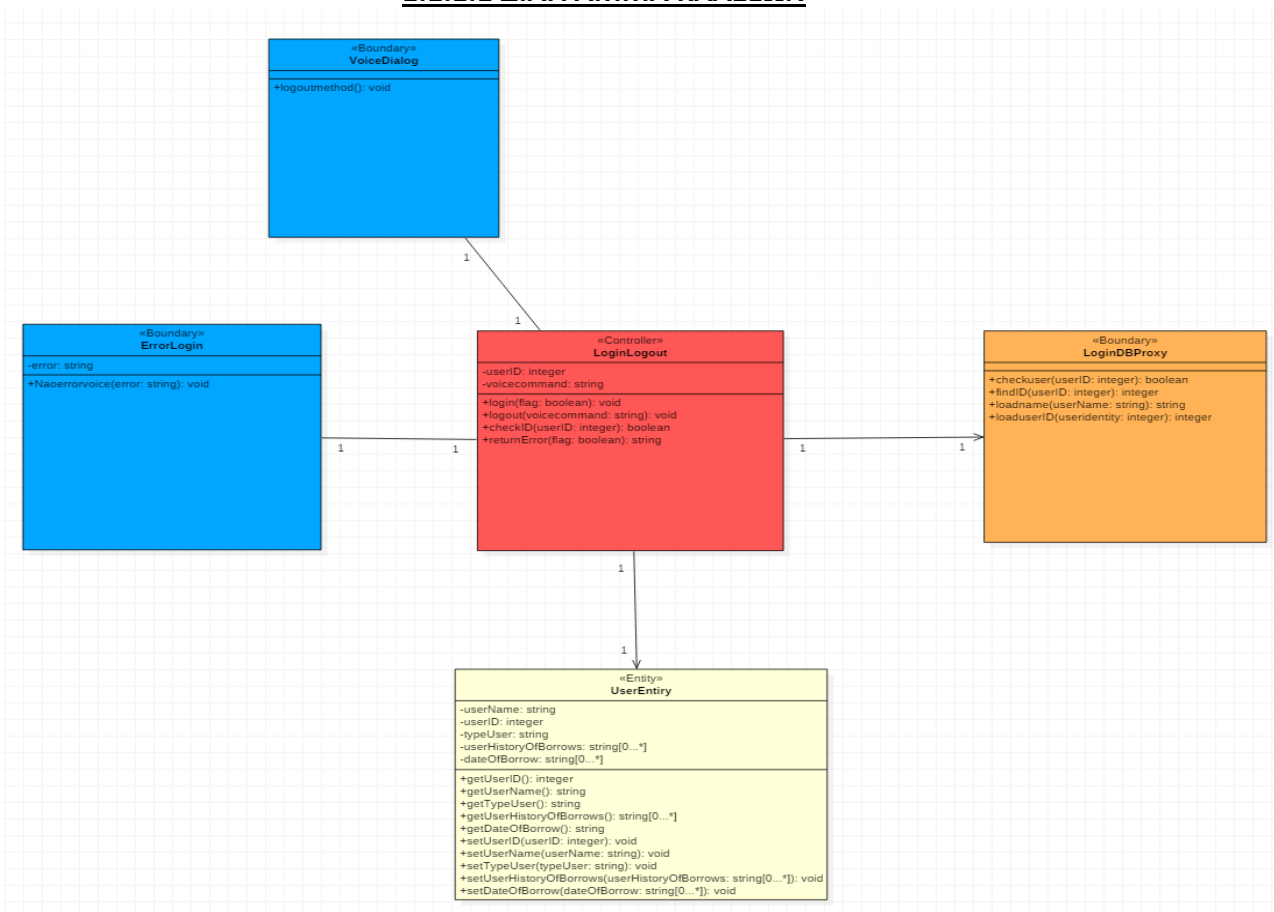


Η κλάση αυτή χρησιμοποιείται για την επικοινωνία με τη βάση δεδομένων του συστήματος όπου είναι αποθηκευμένα τα ονόματα και οι χαρακτηριστικοί αριθμοί των εγγεγραμμένων χρηστών του συστήματος.

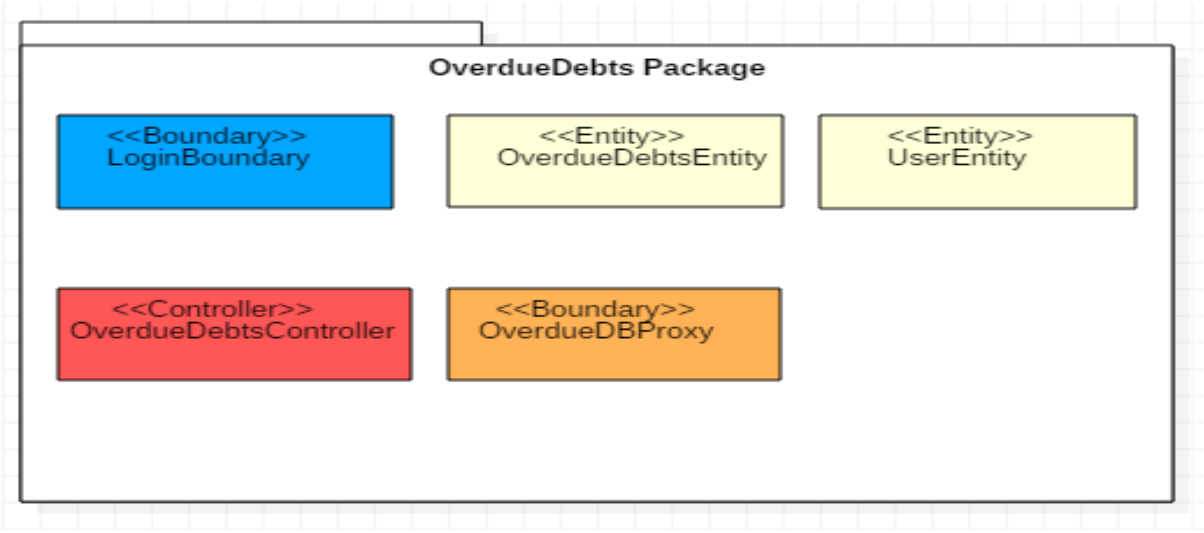
Μέθοδοι της κλάσης:

- checkuser(userID:integer): Η μέθοδος αυτή ελέγχει την εγκυρότητα του χαρακτηριστικού κωδικού που έχει δοθεί στο σύστημα από τον χρήστη.
- findID(userID:integer): Η μέθοδος αυτή ψάχνει στην βάση δεδομένων το συγκεκριμένο χαρακτηριστικό κωδικό.
- loadname(userName:string): Μετά την αναζήτηση του χαρακτηριστικού κωδικού από την findID() μέσω της μεθόδου loadname() φορτώνεται το όνομα του χρήστη που αντιστοιχεί στον συγκεκριμένο κωδικό.
- loaduserID(useridentity:integer): Μετά την εύρεση του αναζήτηση κωδικού από την findID() μέσω της μεθόδου loaduserID() φορτώνεται ο χαρακτηριστικός κωδικός του εγγεγραμμένου χρήστη του χρήστη που αντιστοιχεί στον συγκεκριμένο κωδικό.

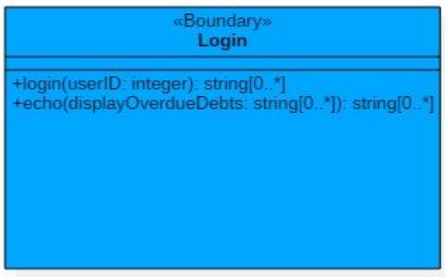
1.1.1.1 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ



1.1.2 OverdueDebts Package



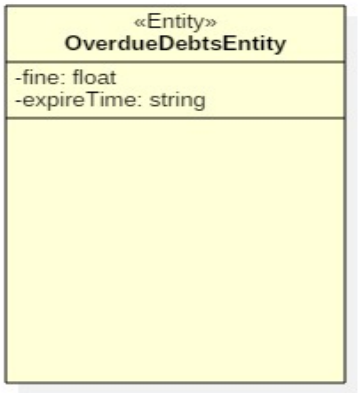
Boundary Login



Μέθοδοι της κλάσης:

- login(userID: integer): string[0..*] : Η συνάρτηση αυτή πραγματοποιεί τη σύνδεση του χρήστη στο σύστημα και επιστρέφει έναν πίνακα από συμβολοσειρές εκ των οποίων το ένα χαρακτηριστικό είναι το dateOfBorrow στο UserEntity μέσω του οποίου γίνεται ο έλεγχος για τις ληξιπρόθεσμες οφειλές.
- echo(displayOverdueDebts: string[0..*]): string[0..*] : Η συνάρτηση αυτή δίνει τη δυνατότητα στο ΝΑΟ να εκφωνήσει τις ληξιπρόθεσμες οφειλές στο χρήστη, αν βέβαια υπάρχουν, με το που συνδεθεί στο σύστημα.

Entity OverdueDebtsEntity



Αυτή είναι η κλάση ορισμού των ληξιπρόθεσμων οφειλών με τα αντίστοιχα χαρακτηριστικά της.

Χαρακτηριστικά της κλάσης:

- fine : Είναι η ποσότητα του προστίμου της ληξιπρόθεσμης οφειλής.
- expireTime : Προσδιορίζει το χρόνο που έχει περάσει από την ημερομηνία επιστροφής του εκάστοτε βιβλίου.

Controller OverdueDebtsController

Αυτός ο ελεγκτής εκκινεί εκείνες τις διαδικασίες για αναζήτηση των ληξιπρόθεσμων οφειλών και ανακοίνωση του αποτελέσματος και ταυτόχρονα ενημέρωση του OverdueDebtsEntity.

Μέθοδοι της κλάσης:

- validCheckForOverdueDebts(loginDetails: string[0..*], dateOfBorrow:string[0..*]): Η συνάρτηση αυτή δέχεται τα στοιχεία εισόδου του χρήστη και τις ημερομηνίες δανεισμών των βιβλίων. Επιστρέφει μία λογική τιμή 1 ή 0 αν έχει νόημα η αναζήτηση ή όχι στη βάση δεδομένων των ληξιπρόθεσμων οφειλών.
- getLoginUserDetails(): Επιστρέφει τα στοιχεία του χρήστη που συνδέεται.
- setOverdueDebts(overdueDetails: OverdueEntity): Ενημερώνει την οντότητα UserEntity ως προς τα χαρακτηριστικά fine και expireTime.
- resultAnnouncement(overdueDetails:OverdueEntity): Αποθηκεύει το αποτέλεσμα που θα ανακοινωθεί από το NAO όπως το ποσό του προστίμου και τις ημέρες που έχουν περάσει από τις ημερομηνίες επιστροφής των βιβλίων.

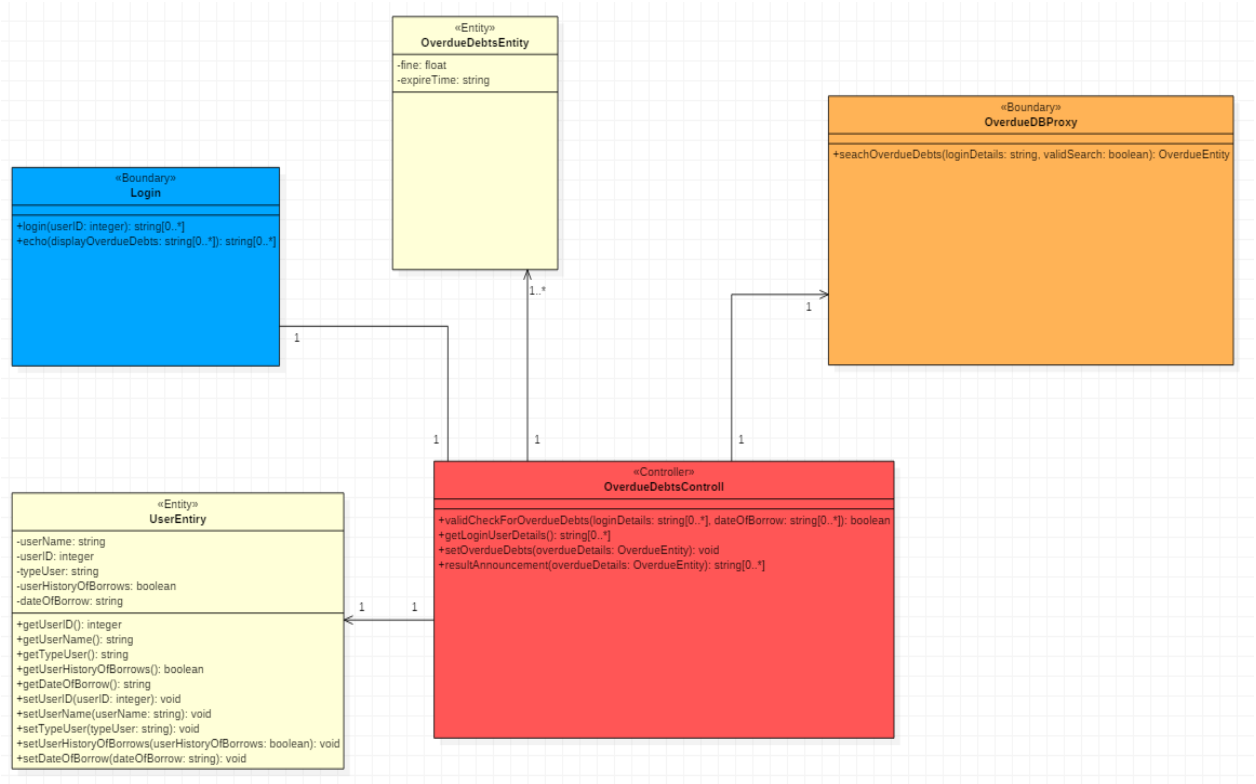
Boundary OverdueDBProxy

Η κλάση αυτή πραγματοποιεί την αναζήτηση των ληξιπρόθεσμων οφειλών στη βάση δεδομένων.

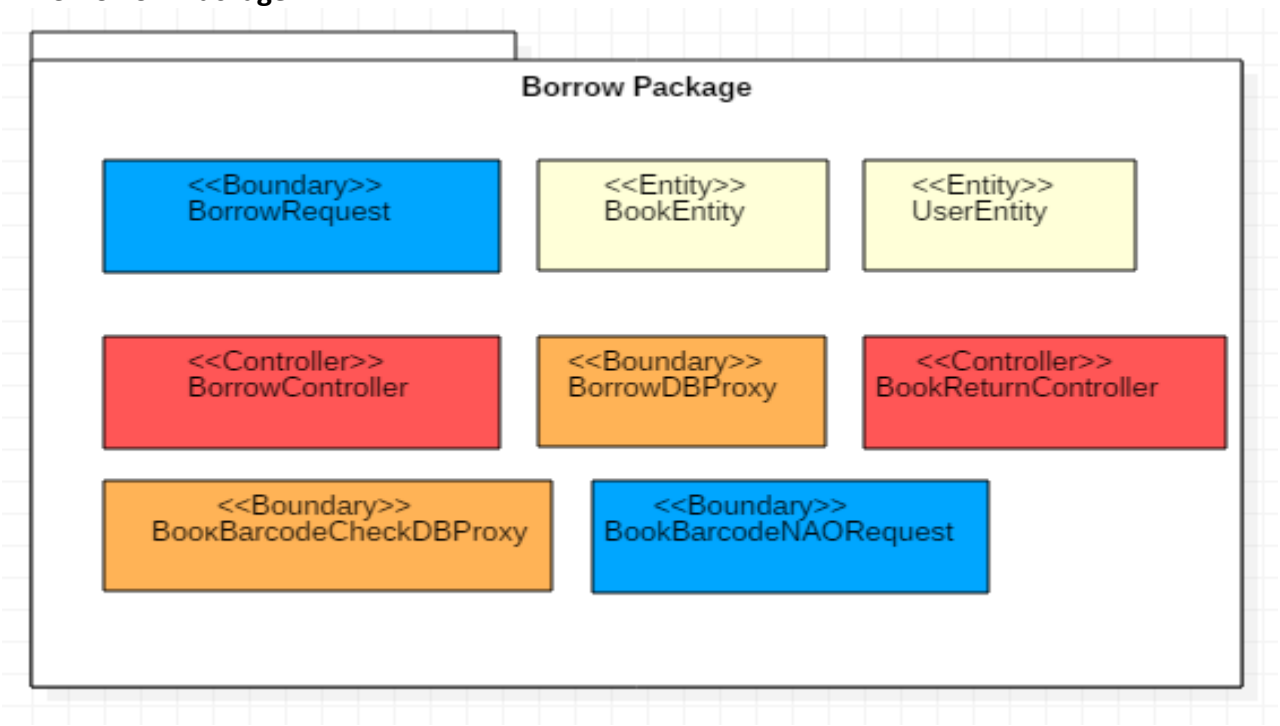
Μέθοδοι της κλάσης:

- searchOverdueDebts(loginDetails: string[0..*], validSearch: boolean): Η συνάρτηση πραγματοποιεί την αναζήτηση των ληξιπρόθεσμων οφειλών στη βάση δεδομένων.

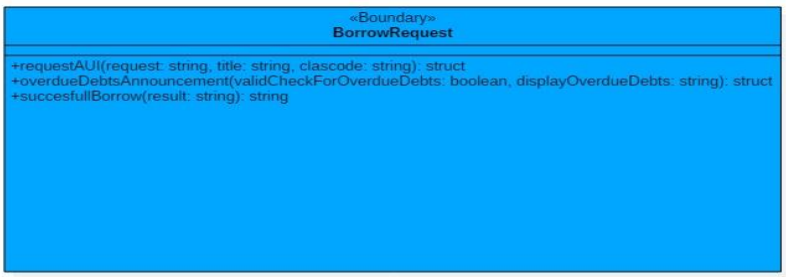
1.1.2.1 Διαγράμματα Κλάσεων



1.1.3 Borrow Package



Boundary BorrowRequest

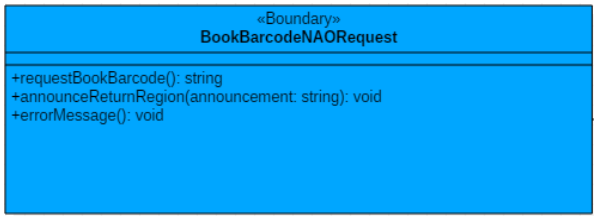


Η κλάση αυτή μεταφέρει στο NAO το αίτημα του χρήστη για δανεισμό βιβλίου.

Μέθοδοι της κλάσης:

- requestAUI(request:string, title:string, clascode:string): Η συνάρτηση αυτή δέχεται από το χρήστη το αίτημα δανεισμού και έπειτα τον τίτλο ή τον Ταξινομικό Κωδικό αυτού. Η συνάρτηση επιστρέφει ένα struct με αυτές τις πληροφορίες.
- overdueDebtsAnnouncement(validCheckForOverdueDebts: boolean, displayOverdueDebts: string): Εφόσον δεν υπάρχουν ληξιπρόθεσμες οφειλές, κάτι το οποίο ελέγχεται από τη μεταβλητή validCheckForOverdueDebts, ανακοινώνονται στο χρήστη μέσω του NAO οι λεπτομέρειες των οφειλών του αποτρέποντας του ταυτόχρονα να πραγματοποιήσει κάποιο δανεισμό.
- succesfullBorrow(result:string): Η μέθοδος αυτή ανακοινώνει, μέσω του NAO, στο χρήστη μήνυμα επιτυχούς δανεισμού.

BookBarcodeNAORequest

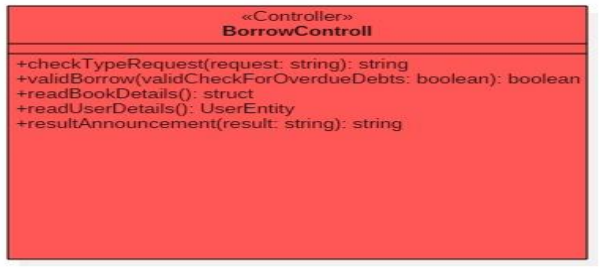


Η κλάση αυτή αποτελεί τη διεπαφή του χρήστη κατά τη διαδικασία επιστροφής δανεισμένου βιβλίου.

Μέθοδοι της κλάσης:

- requestBookBarcode( ): μέθοδος η οποία διαβάζει τον ραβδοκώδικα του βιβλίου. Επιστρέφει μία συμβολοσειρά.
- announceReturnRegion(): μέθοδος η οποία ανακοινώνει την επιτυχή επιστροφή του βιβλίου μέχω του NAO και εκφωνεί τον χώρο εναπόθεσης του βιβλίου στον φυσικό χώρο της βιβλιοθήκης.
- errorMessage(): μέθοδος η οποία εμφανίζει μήνυμα σφάλματος μη ύπαρξης του κωδικού του βιβλίου στη βάση δεδομένων.

Controller BorrowControll

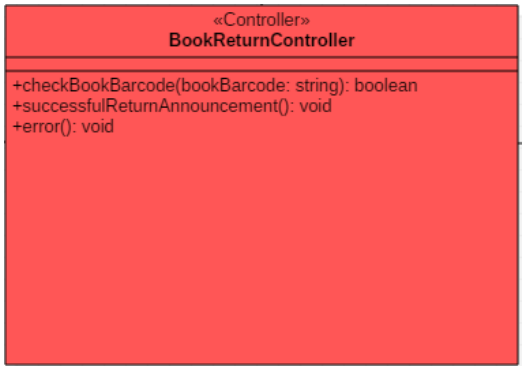


Ο ελεγκτής αυτός διαθέτει εκείνες τις μεθόδους που ελέγχουν τον τύπο του αιτήματος που έχει κατατεθεί από το χρήστη, αν είναι δυνατός ο δανεισμός κάποιου βιβλίου, διαβάζει τα χαρακτηριστικά του χρήστη και του βιβλίου που πρόκειται να δανειστεί και τέλος αποθηκεύει το αποτέλεσμα για επιτυχή δανεισμό ή μη.

Μέθοδοι της κλάσης:

- checkTypeRequest(request:string): Η μέθοδος αυτή ελέγχει τον τύπο του αιτήματος που έχει διατυπώσει ο χρήστης στο NAO και το αποθηκεύει ώστε να χρησιμοποιηθεί αργότερα από τη βάση δεδομένων.
- validBorrow(validCheckForOverdueDebts:boolean) : Η μέθοδος αυτή δίνει τη δυνατότητα στο χρήστη να δανειστεί ή όχι με βάση την τιμή της μεταβλητής validCheckForOverdueDebts 1 ή 0.
- readBookDetails(): Η μέθοδος αυτή διαβάζει τα χαρακτηριστικά του βιβλίου που έχει δώσει ο χρήστης στο NAO, στην κλάση BorrowRequest.
- readUserDetails(): Διαβάζει τα χαρακτηριστικά του χρήστη ώστε αυτά να χρησιμοποιηθούν από τη βάση δεδομένων για να καταχωρηθεί ο δανεισμός.
- resultAnnouncement(result:string): Αποθηκεύεται το μήνυμα επιτυχούς δανεισμού που αργότερα εκφωνείται από το NAO, στην κλάση BorrowRequest.

Controller BookReturnController

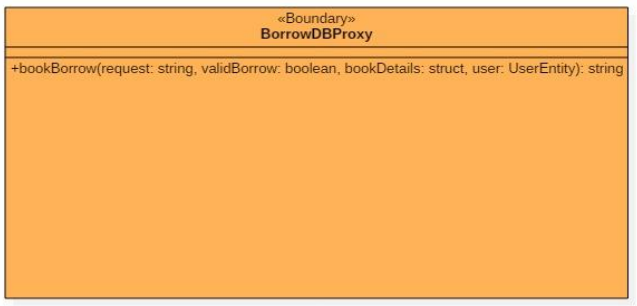


Η κλάση αυτή ελέγχει την διαδικασία επιστροφής δανεισμένου βιβλίου.

Μέθοδοι της κλάσης:

- checkBookBarcode(bookBarcode: string): δέχεται ως όρισμα μία συμβολοσειρά που αντιπροσωπεύει το barcode που διαβάστηκε και καλεί την bookBarcodeCheck() της BookBarcodeCheckDBProxy.Επιστρέφει αληθές όταν το βιβλίο υπάρχει στην βάση δεδομένων , ενώ αντίθετα επιστρέφει ψευδές.
- SuccessfulReturnAnnouncement(): μέθοδος η οποία καλεί την announceReturnRegion της BookBarcodeNAORequest.
- error(): καλεί την errorMessage της BookBarcodeNAORequest.

Boundary BorrowDBProxy

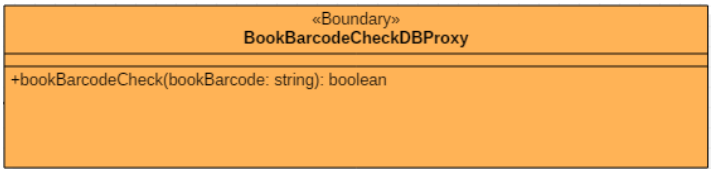


Η κλάση αυτή πραγματοποιεί τη σύνδεση του συστήματος με τη βάση δεδομένων και την μεταφορά των κατάλληλων παραμέτρων για να επιτευχθεί ο δανεισμός.

Μέθοδοι της κλάσης:

- bookBorrow(request:string,validBorrow:boolean,bookDetails:struct,user:UserEntity): Η μέθοδος αυτή αφού δεχτεί το αίτημα του χρήστη για δανεισμό, την επιβεβαίωση για πραγματοποίηση δανεισμού, τις λεπτομέρειες του βιβλίου και τα στοιχεία του χρήστη, αναζητά στη βάση δεδομένων το βιβλίο και κατοχυρώνει το δανεισμό του στο χρήστη αν αυτό καθίσταται εφικτό.

BookBarcodeCheckDBProxy

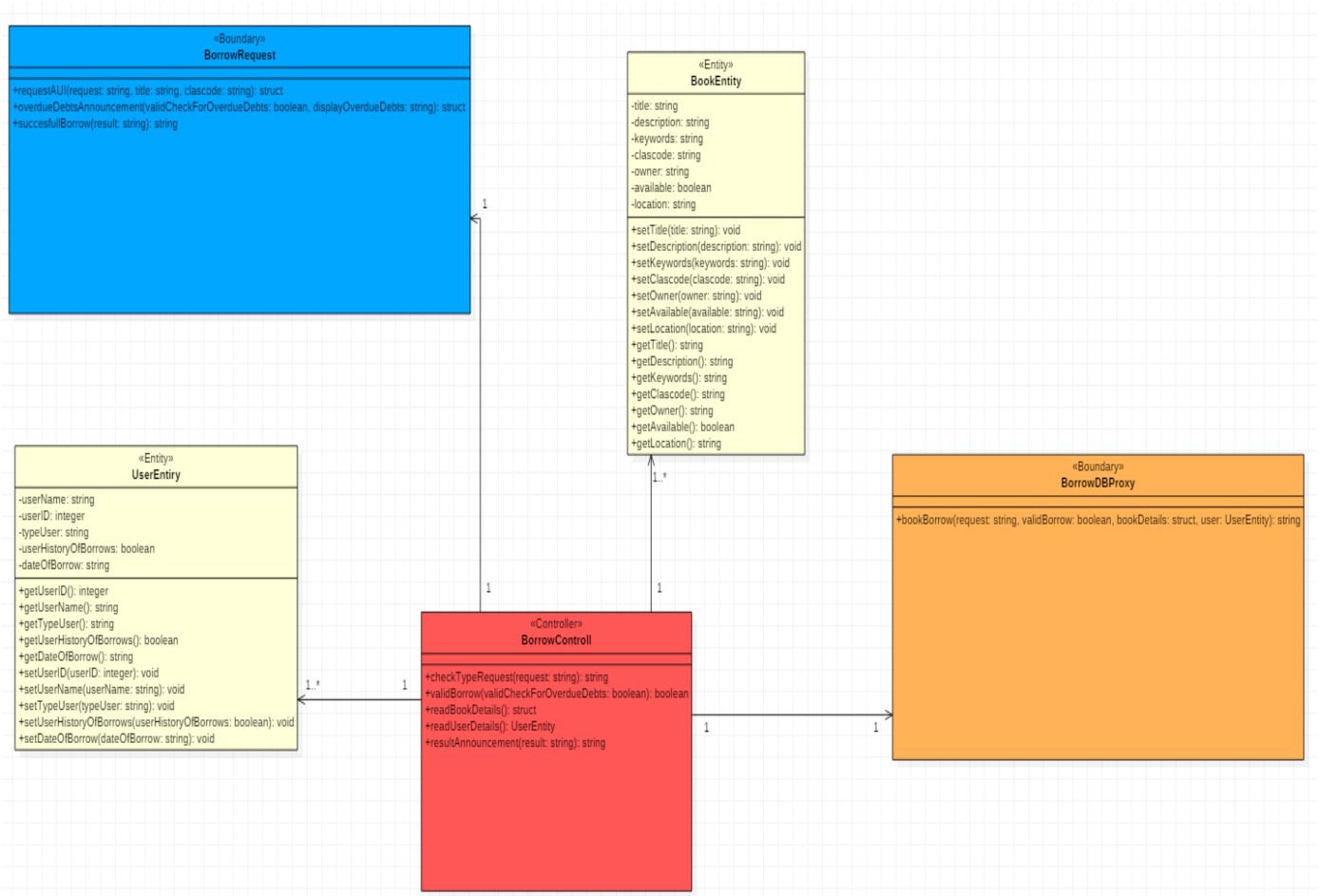


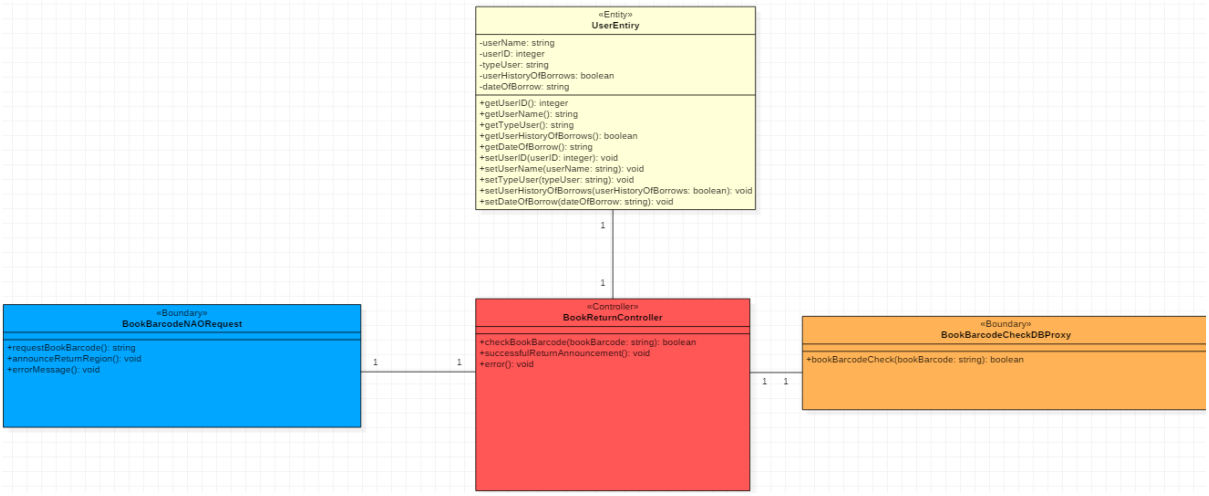
Η κλάση αυτή ελέγχει την διαδικασία επιστροφής δανεισμένου βιβλίου.

Μέθοδοι της κλάσης:

- bookBarcodeCheck(bookBarcode: string): μέθοδος η οποία δέχεται το ραβδοκώδικα του βιβλίου ως όρισμα και ελέγχει την ύπαρξη του στη βάση δεδομένων. Επιστρέφει αληθές όταν βρεθεί το βιβλίο στην βάση δεδομένων, αντίθετα επιστρέφει ψευδές.

1.1.3.1 Διαγράμματα Κλάσεων

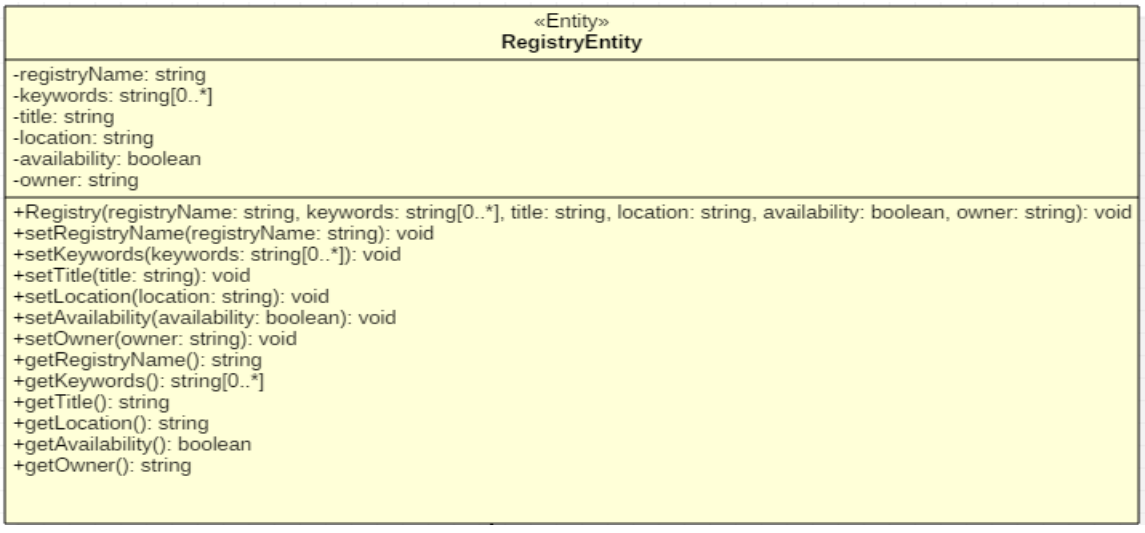




1.1.4-1.1.6 SearchAddEditRegistryPackage

1.1.4 SearchPackage

RegistryEntity



Η οντότητα αυτή περιέχει όλες τις πληροφορίες , οι οποίες σχετίζονται με το βιβλίο και είναι απαραίτητες για την λειτουργία του συστήματος.

Χαρακτηριστικά της κλάσης :

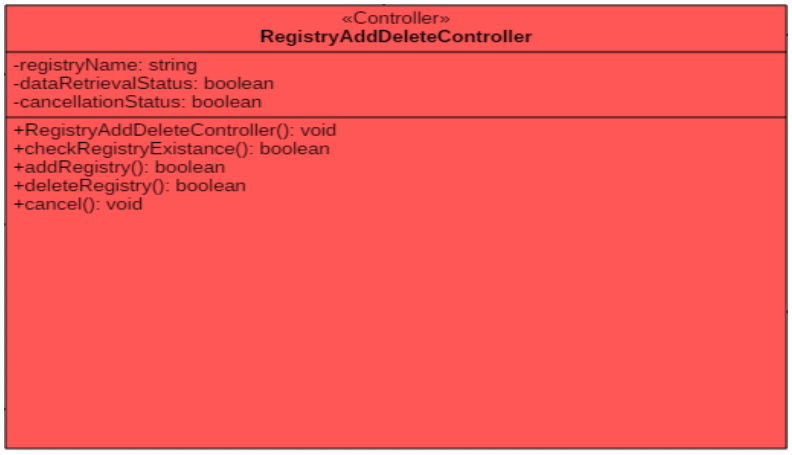
- registryName : string , το όνομα της εγγραφής.
- keywords: string[0..*] , μία συλλογή συμβολοσειρών οι οποίες περιέχουν λέξεις-κλειδιά που σχετίζονται με το βιβλίο που περιγραφεί η τρέχουσα εγγραφή.
- title: string , ο τίτλος του βιβλίου.
- location: string , μία συμβολοσειρά η οποία περιέχει τον ταξινομικό κωδικό του βιβλίου και επακόλουθα προσδιορίζει τη φυσική θέση του βιβλίου.

- availability: boolean , μία λογική μεταβλητή, η οποία προσδιορίζει την διαθεσιμότητα του βιβλίου προς δανεισμό (εάν η τιμή της είναι αληθής τότε είναι διαθέσιμο προς δανεισμό το βιβλίο , ενώ αν είναι ψευδής τότε το βιβλίο είναι ήδη δανεισμένο).
- owner: string , μία συμβολοσειρά , η οποία προσδιορίζει στην κατοχή του το βιβλίο.

Μέθοδοι της κλάσης :

- Registry(in registryName:string, in keywords:string[0..*], in title:string, in location:string, in availability:boolean, in owner:string): void , μέθοδος δόμησης της κλάσης RegistryEntity , δέχεται ως ορίσματα το όνομα,τις λέξεις-κλειδιά,τον τίτλο, την τοποθεσία, την διαθεσιμότητα και τον ιδιοκτήτη.
- setRegistry(registryName: string): void , μέθοδος η οποία θέτει την όνομα χρήστη.
- setKeywords(keywords: string[0..*]): void , μέθοδος η οποία θέτει τις λέξεις-κλειδιά.
- setTitle(title:string):void , μέθοδος η οποία θέτει τον τίτλο του βιβλίου.
- setLocation(location: string): void , μέθοδος η οποία θέτει το όνομα τον ταξινομικό κωδικό της τοποθεσίας του βιβλίου.
- setAvailability(availability: boolean): void , μέθοδος η οποία θέτει την διαθεσιμότητα του βιβλίου.
- setOwner(owner: string): void , μέθοδος η οποία θέτει το όνομα του χρήστη που έχει στην κατοχή του το βιβλίο.
- getRegistryName(): string , μέθοδος η οποία επιστρέφει το όνομα της εγγραφής.
- getKeywords(): string[0..*] , μέθοδος η οποία επιστρέφει τις λέξεις-κλειδιά του βιβλίου.
- getTitle(): string , μέθοδος η οποία επιστρέφει τον τίτλο του βιβλίου.
- getLocation(): string , μέθοδος η οποία επιστρέφει την τοποθεσία του βιβλίου.
- getAvailability(): boolean , μέθοδος η οποία επιστρέφει την διαθεσιμότητα του βιβλίου.
- getOwner(): string , μέθοδος η οποία επιστρέφει το όνομα του χρήστη που έχει στην κατοχή του βιβλίο.

RegistryAddDeleteController



Ο ελεγκτής είναι υπεύθυνος για την προσθήκη και αφαίρεση εγγραφών στην εξωτερική βάση δεδομένων.

Χαρακτηριστικά της κλάσης :

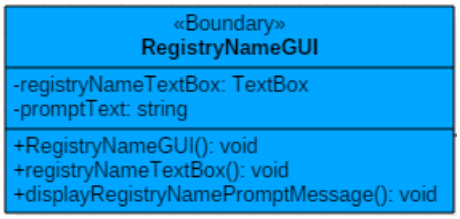
- registryName : string , το όνομα της εγγραφής.
- dataRetrievalStatus: boolean , η μεταβλητή αυτή παίρνει τη λογική τιμή αληθής όταν τα δεδομένα έχουν ληφθεί από την βάση δεδομένων ενώ σε αντίθετη περίπτωση λαμβάνει την τιμή ψευδής.
- cancellationStatus: boolean , η μεταβλητή αυτή παίρνει τη λογική τιμή αληθής όταν η διαδικασία προσθήκης/αφαίρεσης τεθεί προς ακύρωση ενώ σε αντίθετη περίπτωση λαμβάνει την τιμή ψευδής.

Μέθοδοι της κλάσης :

- RegistryAddDeleteController(): void , μέθοδος δόμησης του ελεγκτή RegistryAddDeleteController.

- checkRegistryExistance():boolean, η μέθοδος αυτή ελέγχει την ύπαρξη της εγγραφής στην βάση δεδομένων καλώντας τη μέθοδο checkRegistry της οριακής κλάσης RegistryCheckDBProxy. Επιστρέφει την λογική τιμή αληθής εάν η διαδικασία ανάκτησης της εγγραφής ήταν επιτυχημένη, ενώ ψευδής σε αντίθετη περίπτωση.
- addRegistry(): boolean , η μέθοδος αυτή προσθέτει στην βάση δεδομένων την εγγραφή με το όνομα που εισήγαγε ο χρήστης ,εάν η μεταβλητή dataRetrievalStatus έχει την τιμή αληθής , καλώντας τη μέθοδο addRegistryDB της οριακής κλάσης RegistryAddDeleteDBProxy. Επιστρέφει την λογική τιμή αληθής εάν η διαδικασία προσθήκης της εγγραφής ήταν επιτυχημένη, ενώ ψευδής σε αντίθετη περίπτωση.
- deleteRegistry(): boolean , η μέθοδος αυτή αφαιρεί από την βάση δεδομένων την εγγραφή με το όνομα που εισήγαγε ο χρήστης ,εάν η μεταβλητή dataRetrievalStatus έχει την τιμή ψευδής , καλώντας τη μέθοδο deleteRegistryDB της οριακής κλάσης RegistryAddDeleteDBProxy. Επιστρέφει την λογική τιμή αληθής εάν η διαδικασία διαγραφής της εγγραφής ήταν επιτυχημένη, ενώ ψευδής σε αντίθετη περίπτωση.
- cancel(): void , η μέθοδος αυτή ακυρώνει την διαδικασία προσθήκης/αφαίρεσης εγγραφών.

RegistryNameGUI



Η οριακή κλάση αυτή είναι υπεύθυνη για την παροχή διεπαφής με σκοπό την εισαγωγή του ονόματος της εγγραφής προς προσθήκη/αφαίρεση από τον διαχειριστή.

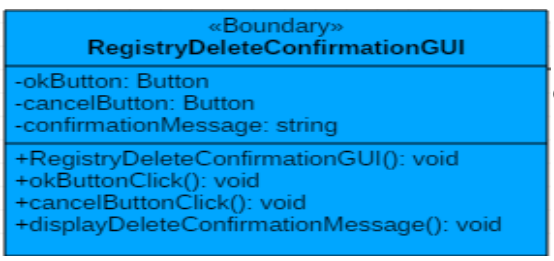
Χαρακτηριστικά της κλάσης :

- registryNameTextBox: TextBox , πεδίο κειμένου προς εισαγωγή του ονόματος εγγραφής.
- promptText: string , προτρεπτικό μήνυμα για την εισαγωγή ονόματος στην γραφική διεπαφή.

Μέθοδοι της κλάσης :

- RegistryNameGUI(): void , μέθοδος δόμησης της RegistryNameGUI.
- registryNameTextBox(): void, μέθοδος η οποία εμφανίζει το πεδίο κειμένου προς εισαγωγή ονόματος και καλεί την checkRegistryExistance του RegistryAddDeleteController.
- displayRegistryNamePromptMessage(): void, η μέθοδος αυτή προβάλλει το περιεχόμενο του promptText στην γραφική διεπαφή.

RegistryDeleteConfirmationGUI



Η οριακή αυτή κλάση ζητάει από τον διαχειριστή την συγκατάθεση του για την οριστική αφαίρεση της εγγραφής από την βάση δεδομένων.

Χαρακτηριστικά της κλάσης :

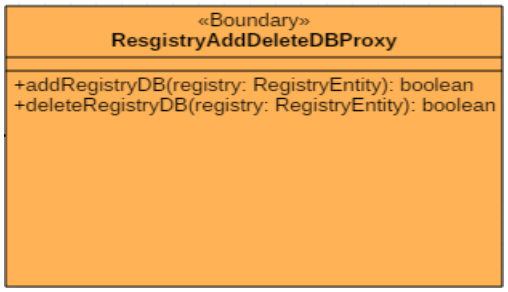
- okButton: Button , το κουμπί αποδοχής αφαίρεσης της εγγραφής.
- cancelButton: Button , το κουμπί αποδοχής ακύρωσης της αφαίρεσης της εγγραφής.

- confirmationMessage: string , συμβολοσειρά η οποία περιέχει το σχετικό μήνυμα για την οριστική αφαίρεση της εγγραφής ή την ακύρωση της αφαίρεσης.

Μέθοδοι της κλάσης :

- RegistryDeleteConfirmationGUI(): void , μέθοδος δόμησης της RegistryDeleteConfirmationGUI.
- okButtonClick(): void, μέθοδος η οποία παρακολουθεί την κατάσταση του κουμπιού αποδοχής της οριστικής διαγραφής της εγγραφής .
- cancelButtonClick(): void, μέθοδος η οποία παρακολουθεί την κατάσταση του κουμπιού ακύρωσης της οριστικής διαγραφής της εγγραφής .
- displayDeleteConfirmationMessage():void , μέθοδος η οποία προβάλλει το περιεχόμενο της confirmationMessage στην γραφική διεπαφή.

RegistryAddDeleteDBProxy

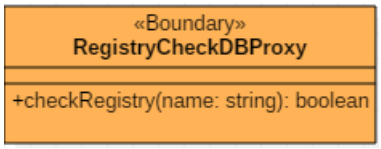


Η οριακή αυτή κλάση προσθέτει/αφαιρεί εγγραφές στην βάση δεδομένων.

Μέθοδοι της κλάσης :

- addRegistryDB(registry:RegistryEntity): boolean, μέθοδος η οποία προσθέτει την εγγραφή που της δόθηκε σαν όρισμα στην βάση δεδομένων. Επιστρέφει την λογική τιμή αληθές όταν η προσθήκη είναι επιτυχής, αντίθετα επιστρέφει ψευδές.
- deleteRegistryDB(registry:RegistryEntity): boolean, μέθοδος η οποία αφαιρεί την εγγραφή που της δόθηκε σαν όρισμα στην βάση δεδομένων. Επιστρέφει την λογική τιμή αληθές όταν η αφαίρεση είναι επιτυχής ,αντίθετα επιστρέφει ψευδές.

RegistryCheckDBProxy

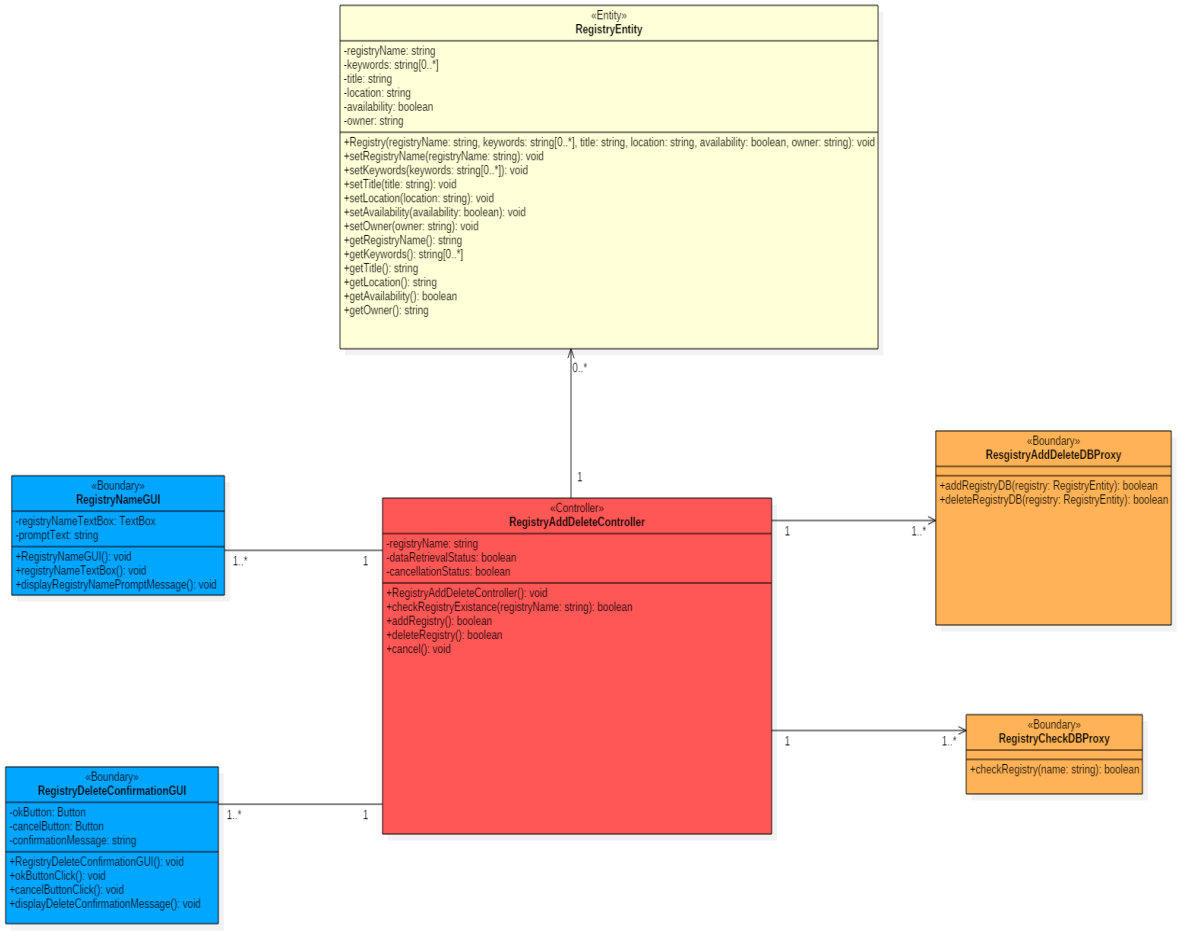


Η οριακή αυτή κλάση έχει ως ρόλο τον έλεγχο ύπαρξης της εγγραφής στη βάση δεδομένων.

Μέθοδοι της κλάσης :

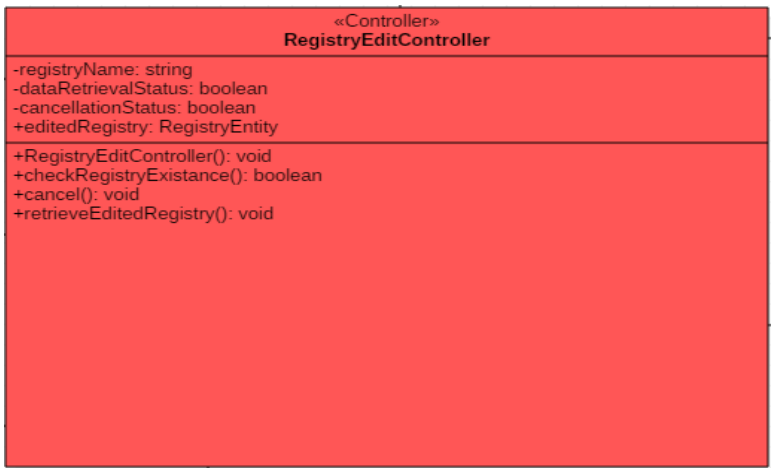
- checkRegistry(name : string): boolean, μέθοδος η οποία ελέγχει την ύπαρξη της εγγραφής στη βάση δεδομένων. Επιστρέφει την λογική τιμή αληθές όταν η εγγραφή βρεθεί, αντίθετα επιστρέφει ψευδές.

1.1.4.1 Διαγράμματα Κλάσεων



1.1.5 AddPackage

RegistryEditController



Ο ελεγκτής είναι υπεύθυνος για την επεξεργασία των εγγραφών που βρίσκονται αποθηκευμένες στην εξωτερική βάση δεδομένων.

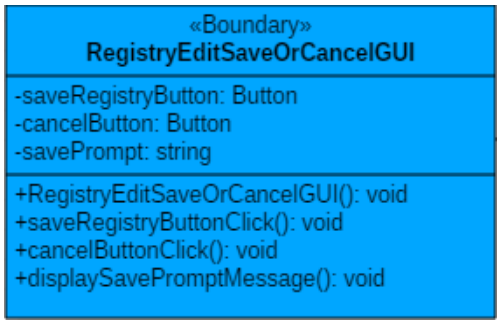
Χαρακτηριστικά της κλάσης :

- registryName : string , το όνομα της εγγραφής.
- dataRetrievalStatus: boolean , η μεταβλητή αυτή παίρνει τη λογική τιμή αληθής όταν τα δεδομένα έχουν ληφθεί από την βάση δεδομένων ενώ σε αντίθετη περίπτωση λαμβάνει την τιμή ψευδής.
- cancellationStatus: boolean , η μεταβλητή αυτή παίρνει τη λογική τιμή αληθής όταν η διαδικασία προσθήκης/αφαίρεσης τεθεί προς ακύρωση ενώ σε αντίθετη περίπτωση λαμβάνει την τιμή ψευδής.
- editedRegistry: RegistryEntity , αντίγραφο της εγγραφής που επεξεργάστηκε ο χρήστης.

Μέθοδοι της κλάσης :

- RegistryEditController(): void , μέθοδος δόμησης του ελεγκτή RegistryEditController.
- checkRegistryExistance():boolean, η μέθοδος αυτή ελέγχει την ύπαρξη της εγγραφής στη βάση δεδομένων καλώντας τη μέθοδο checkRegistry της οριακής κλάσης RegistryCheckDBProxy. Επιστρέφει την λογική τιμή αληθής εάν η διαδικασία ανάκτησης της εγγραφής ήταν επιτυχημένη, ενώ ψευδής σε αντίθετη περίπτωση.
- cancel(): void , η μέθοδος αυτή ακυρώνει την διαδικασία επεξεργασίας της εγγραφής.
- retrieveEditedRegistry(): void , μέθοδος η οποία ανακτά την επεξεργασμένη εγγραφή.

RegistryEditSaveOrCancelGUI



Η οριακή αυτή κλάση ζητάει από τον διαχειριστή την συγκατάθεση του για την οριστική αποθήκευση της επεξεργασμένης εγγραφής στην βάση δεδομένων.

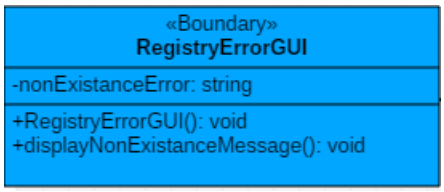
Χαρακτηριστικά της κλάσης :

- saveButton: Button , το κουμπί αποθήκευσης της επεξεργασμένης εγγραφής.
- cancelButton: Button , το κουμπί αποδοχής ακύρωσης της επεξεργασμένης της εγγραφής.
- savePrompt: string , συμβολοσειρά η οποία περιέχει προτρεπτικό μήνυμα για την οριστική αποθήκευση της εγγραφής.

Μέθοδοι της κλάσης :

- RegistryEditSaveOrCancelGUI(): void , μέθοδος δόμησης της RegistryEditSaveOrCancelGUI.
- saveRegistryButtonClick(): void, μέθοδος η οποία παρακολουθεί την κατάσταση του κουμπιού της οριστικής αποθήκευσης της εγγραφής και καλεί την retrieveEditedRegistry του RegistryEditController όταν πατηθεί .
- cancelButtonClick(): void, μέθοδος η οποία παρακολουθεί την κατάσταση του κουμπιού της οριστικής αποθήκευσης της εγγραφής και καλεί την μέθοδο cancel του RegistryEditController όταν πατηθεί .
- displaySavePrompt():void , μέθοδος η οποία προβάλλει το περιεχόμενο της savePrompt στην γραφική διεπαφή.

RegistryErrorGUI



Η οριακή κλάση αυτή προβάλλει στο χρήστη σχετικό μήνυμα στην περίπτωση μη εύρεσης της εγγραφής στην βάση δεδομένων.

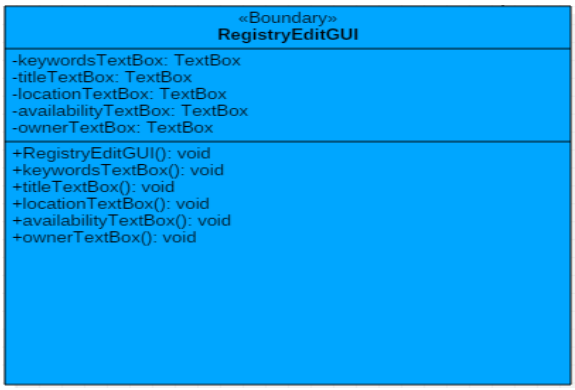
Χαρακτηριστικά της κλάσης :

- nonExistanceError: string , συμβολοσειρά που περιέχει το μήνυμα που σφάλματος εύρεσης της εγγραφής.

Μέθοδοι της κλάσης :

- RegistryErrorGUI(): void , μέθοδος δόμησης της RegistryErrorGUI.
- registryNameTextBox(): void, μέθοδος η οποία εμφανίζει το πεδίο κειμένου προς εισαγωγή ονόματος και καλεί την checkRegistryExistance του RegistryAddDeleteController.
- displayNonExistanceMessage(): void, η μέθοδος προβάλλει το περιεχόμενο της συμβολοσειράς nonExistanceError στη γραφική διεπαφή.
-

RegistryEditGUI



Η οριακή αυτή κλάση προβάλλει στον διαχειριστή τα πεδία των εγγραφών τα οποία επιθυμεί να επεξεργαστεί(λέξεις-κλειδιά, τίτλος, θέση, διαθεσιμότητα, κάτοχος).

Χαρακτηριστικά της κλάσης :

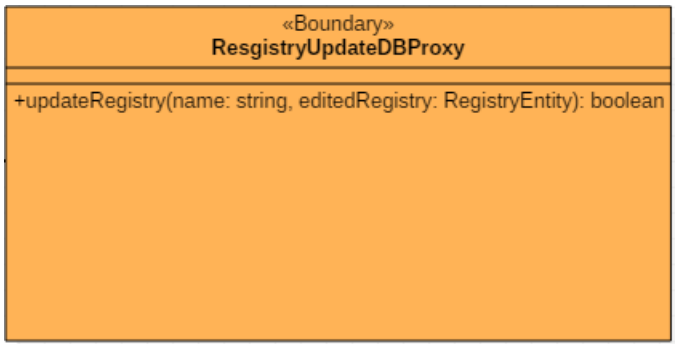
- keywordsTextBox: TextBox , πεδίο κειμένου για εισαγωγή των λέξεων-κλειδιών του βιβλίου.
- titleTextBox: TextBox , πεδίο κειμένου για εισαγωγή του τίτλου του βιβλίου.
- locationTextBox: TextBox , πεδίο κειμένου για εισαγωγή της θέσης του βιβλίου.
- availabilityTextBox: TextBox , πεδίο κειμένου για εισαγωγή της διαθεσιμότητας του βιβλίου.
- ownerTextBox: TextBox , πεδίο κειμένου για εισαγωγή του ονόματος του κατόχου του βιβλίου.

Μέθοδοι της κλάσης :

- RegistryEditGUI(): void , μέθοδος δόμησης της RegistryEditGUI.
- keywordsTextBox(): void , μέθοδος η οποία εμφανίζει το πεδίο κειμένου για εισαγωγή των λέξεων-κλειδιών του βιβλίου , οι οποίες θα σταλεί στον RegistryEditController καλώντας την retrieveEditedRegistry.
- titleTextBox(): void , μέθοδος η οποία εμφανίζει το πεδίο κειμένου για εισαγωγή του τίτλου του βιβλίου , ο οποίος θα σταλεί στον RegistryEditController καλώντας την retrieveEditedRegistry.

- locationTextBox(): void , μέθοδος η οποία εμφανίζει το πεδίο κειμένου για εισαγωγή θέσης του βιβλίου η οποία θα σταλεί στον RegistryEditController καλώντας την retrieveEditedRegistry.
- availabilityTextBox(): void , μέθοδος η οποία εμφανίζει το πεδίο κειμένου για εισαγωγή της διαθεσιμότητας του βιβλίου η οποία θα σταλεί στον RegistryEditController καλώντας την retrieveEditedRegistry.
- ownerTextBox(): void , μέθοδος η οποία εμφανίζει το πεδίο κειμένου για εισαγωγή του κατόχου του βιβλίου , η πληροφορία θα σταλεί στον RegistryEditController καλώντας την retrieveEditedRegistry.

RegistryUpdateDBProxy

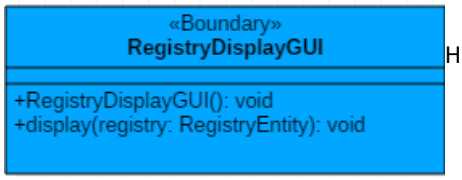


Η οριακή αυτή κλάση αποθηκεύει την εγγραφή που επεξεργάστηκε ο διαχειριστής στη βάση δεδομένων .

Μέθοδοι της κλάσης :

- updateRegistry(name: string , editedRegistry: RegistryEntity): boolean, μέθοδος η οποία αποθηκεύει στη βάση δεδομένων την επεξεργασμένη εγγραφή. Δέχεται ως ορίσματα το όνομα της εγγραφής (name:string) και την προς αποθήκευση εγγραφή (editedRegistry:RegistryEntity). Επιστρέφει την λογική τιμή αληθές όταν η αποθήκευση είναι επιτυχής, αντίθετα επιστρέφει ψευδές.

RegistryDisplayGUI

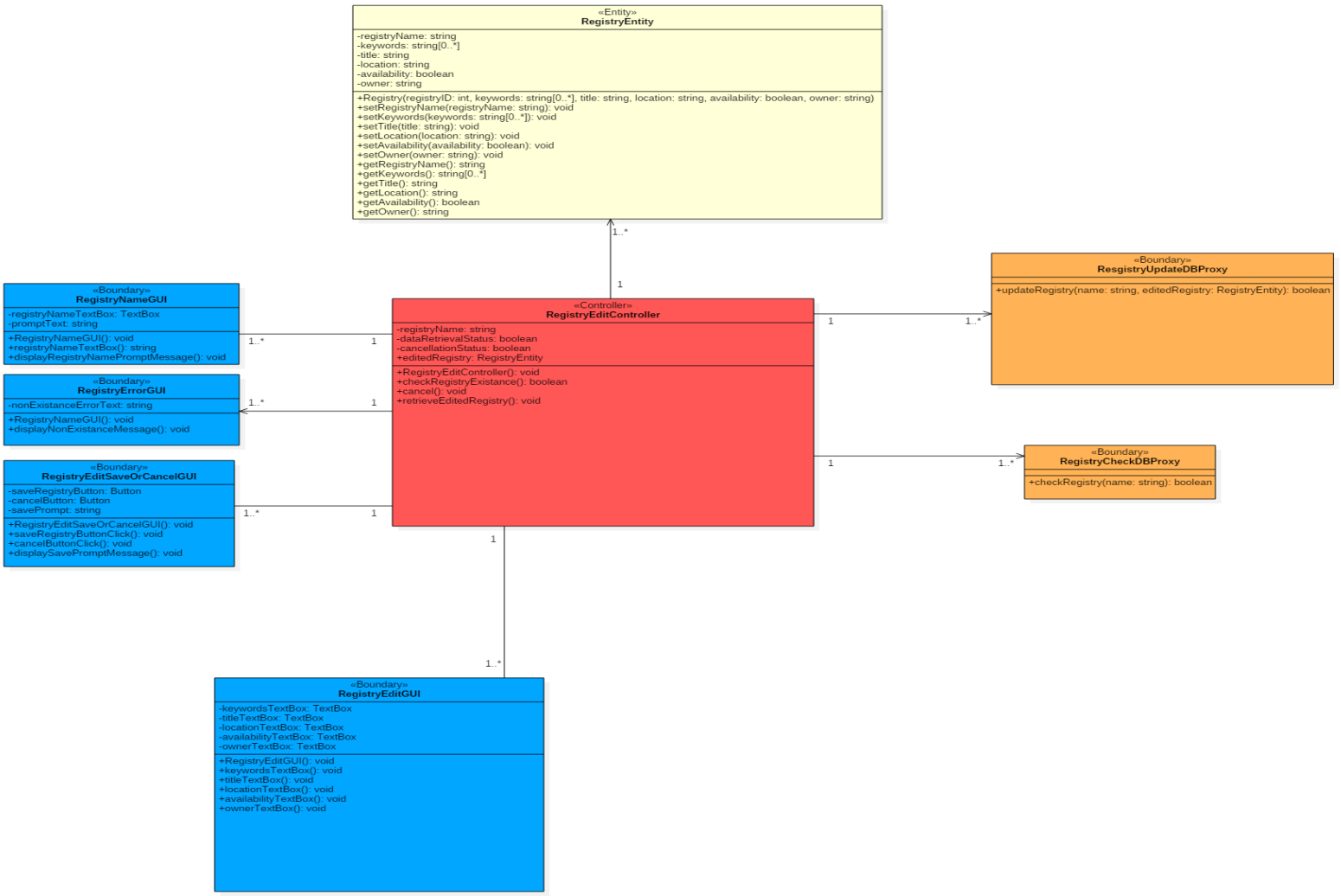


Η οριακή αυτή κλάση προβάλλει στον διαχειριστή τα περιεχόμενα της εγγραφής.

Μέθοδοι της κλάσης :

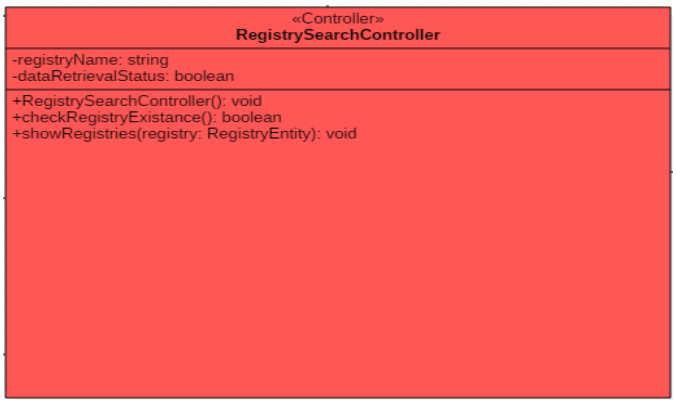
- RegistryDisplayGUI(name: string , editedRegistry: RegistryEntity): void, μέθοδος δόμησης της RegistryDisplayGUI.
- display(registry:RegistryEntity): void , μέθοδος η οποία προβάλλει μία σελίδα η οποία περιέχει τις πληροφορίες της εγγραφής που δόθηκε ως όρισμα. Οι πληροφορίες που προβάλλονται είναι το όνομα της εγγραφής, οι λέξεις-κλειδιά,η τοποθεσία,η διαθεσιμότητα , ο τίτλος και ο κάτοχος του βιβλίου.

1.1.5.1 Διάγραμμα Κλάσεων



1.1.6 EditPackage

RegistrySearchController



Ο ελεγκτής είναι υπεύθυνος για την αναζήτηση των εγγραφών που βρίσκονται αποθηκευμένες στην εξωτερική βάση δεδομένων.

Χαρακτηριστικά της κλάσης :

- registryName : string , το όνομα της εγγραφής.
- dataRetrievalStatus: boolean , η μεταβλητή αυτή παίρνει τη λογική τιμή αληθής όταν τα δεδομένα έχουν ληφθεί από την βάση δεδομένων ενώ σε αντίθετη περίπτωση λαμβάνει την τιμή ψευδής.

Μέθοδοι της κλάσης :

- RegistrySearchController(): void , μέθοδος δόμησης του ελεγκτή RegistrySearchController.
- checkRegistryExistance():boolean, η μέθοδος αυτή ελέγχει την ύπαρξη της εγγραφής στη βάση δεδομένων καλώντας τη μέθοδο checkRegistry της οριακής κλάσης RegistryCheckDBProxy. Επιστρέφει την λογική τιμή αληθής εάν η διαδικασία ανάκτησης της εγγραφής ήταν επιτυχημένη, ενώ ψευδής σε αντίθετη περίπτωση.Εάν δεν βρεθεί τότε καλεί την μέθοδο displayNonExistanceMessage της RegistryErrorGUI.
- showRegistries(): void , μέθοδος η οποία καλεί την μέθοδο display της RegistryDisplayGUI για να προβάλλει τις πληροφορίες των εγγραφών.
- retrieve(): void , μέθοδος η οποία καλεί την retrieveRegistry της RegistryRetrieveDBProxy για να ανακτήσει από την βάση δεδομένων την ζητούμενη εγγραφή.

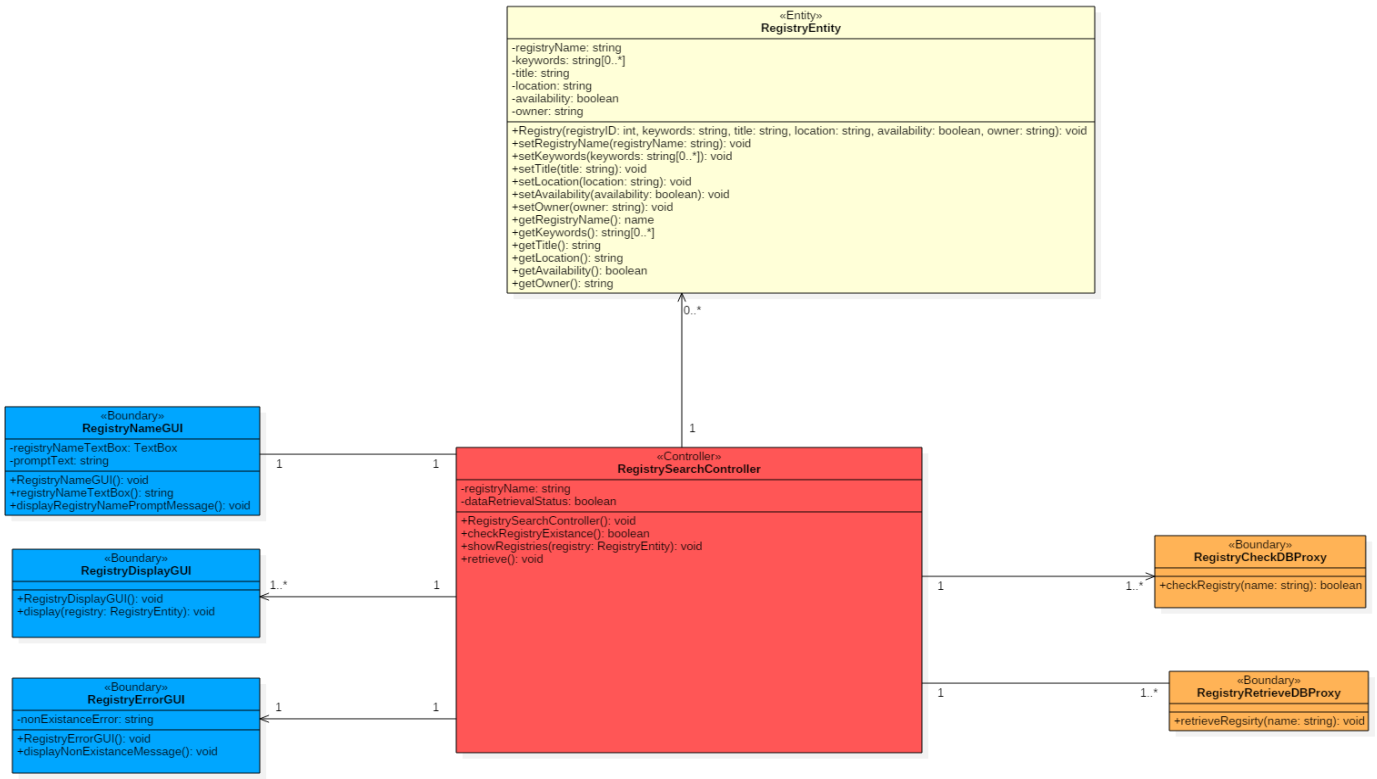
RegistryRetrieveDBProxy

Η οριακή αυτή κλάση ανακτά από την βάση δεδομένων εγγραφές.

Μέθοδοι της κλάσης :

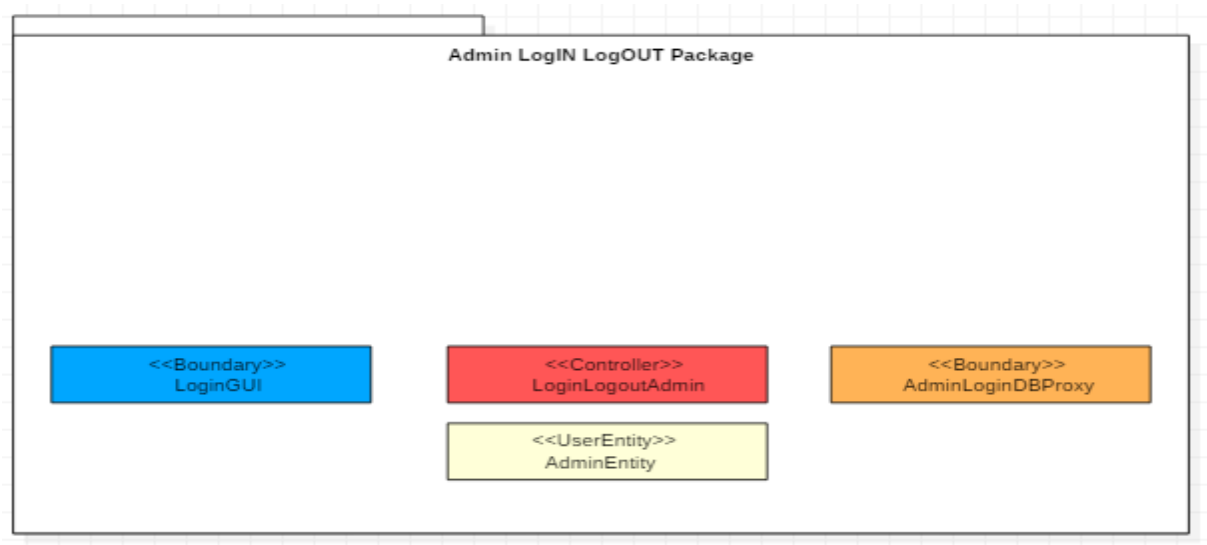
- retrieve(name: string): void, μέθοδος η οποία αναζητά εγγραφές στην βάση δεδομένων με το όνομα που δόθηκε ως όρισμα και τις ανακτά.

1.1.6.1 Διάγραμμα Κλάσεων

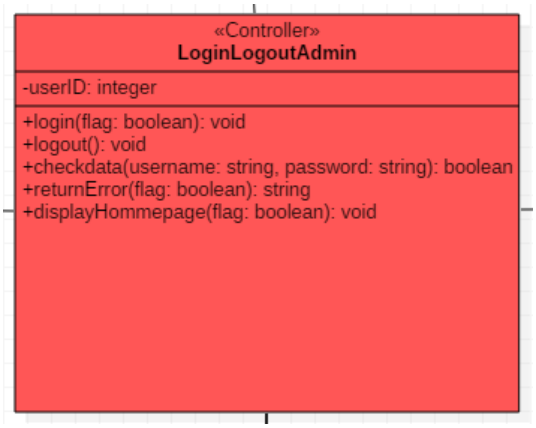


1.1.7 Admin LogIN LogOUT Package

Το πακέτο αυτό αναφέρεται στις επιλογές login και logout του διαχειριστή του συστήματος. Παρακάτω παρουσιάζονται οι κλάσεις που χρησιμοποιούνται για την υλοποίηση του πακέτου αυτού.



Controller LoginLogoutAdmin



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

Χαρακτηριστικά της κλάσης:

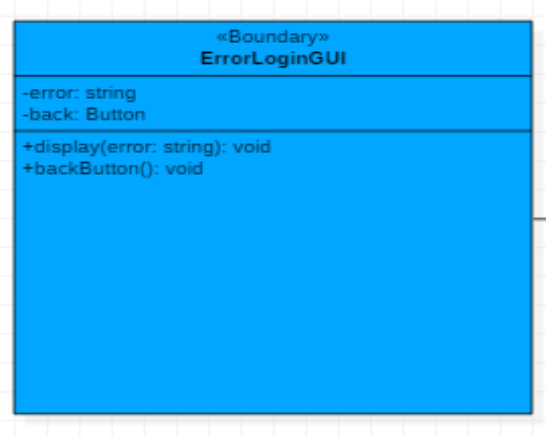
- userID : Μοναδικός χαρακτηριστικός αριθμός των διαχειριστών του συστήματος.

Μέθοδοι της κλάσης:

- login(flag:boolean) : Η μέθοδος αυτή συνδέει το διαχειριστή στο σύστημα της εφόσον η μεταβλητή flag είναι αληθής.
- logout() : Η μέθοδος αυτή αποσυνδέει το διαχειριστή από το σύστημα.
- checkdata(username:string ,password:string) : Η μέθοδος αυτή ελέγχει αν τα στοιχεία ελέγχει αν τα στοιχεία που έδωσε ο διαχειριστής είναι ορθά. Σε περίπτωση που τα στοιχεία είναι ορθά καλείται η συνάρτηση login() σε κάθε άλλη περίπτωση καλείται η συνάρτηση returnError().
- returnError(flag:boolean) : Η συνάρτηση αυτή επιστρέφει μήνυμα σφάλματος εύρεσης των στοιχείων του εγγεγραμμένου χρήστη στην display() της οριακής κλάσης ErrorLoginGUI.Η διαδικασία αυτή εκτελείται εφόσον η flag είναι ψευδής.

- displayHommepage(flag: boolean): Η μέθοδος αυτή καλείται σε περίπτωση που η μεταβλητή flag είναι ψευδής για να επιστρέψει στην αρχική σελίδα συμπλήρωσης των στοιχείων ώστε ο διαχειριστής να πληκτρολογήσει σωστά το όνομα χρήστη και τον κωδικό του.

ErrorLoginGUI



Η οριακή κλάση αυτή είναι υπεύθυνη για την επιστροφή των σφαλμάτων σύνδεσης του διαχειριστή στο σύστημα.

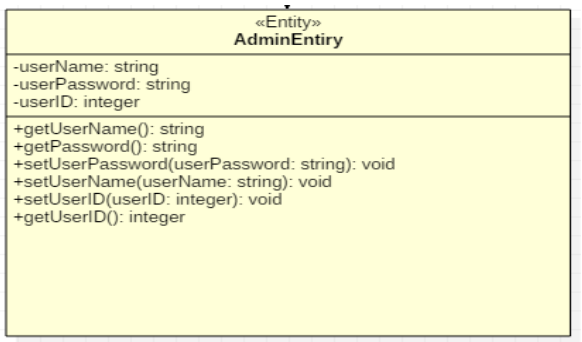
Χαρακτηριστικά της κλάσης:

- error: Το μήνυμα σφάλματος που προβάλλονται στον διαχειριστή.
- back : Κουμπί για την επιστροφή στην σελίδα αρχικής συμπλήρωσης των στοιχείων εισόδου του διαχειριστή.

Μέθοδοι της κλάσης:

- display(error: string) : Η μέθοδος αυτή είναι υπεύθυνη για την προβολή μηνύματος σφάλματος.Το μήνυμα σφάλματος δίνεται από τη συνάρτηση returnError() του ελεγκτή LoginLogoutAdmin.
- backButton() : Η μέθοδος καλεί τη συνάρτηση displayHommepage() του ελεγκτή LoginLogoutAdmin() για επιστροφή στην αρχική οθόνη συμπλήρωσης των στοιχείων εισόδου του διαχειριστή.

AdminEntity



Αυτή είναι η κλάση οντότητας του διαχειριστή που περιέχει και τα στοιχεία ορισμού του.

Χαρακτηριστικά της κλάσης:

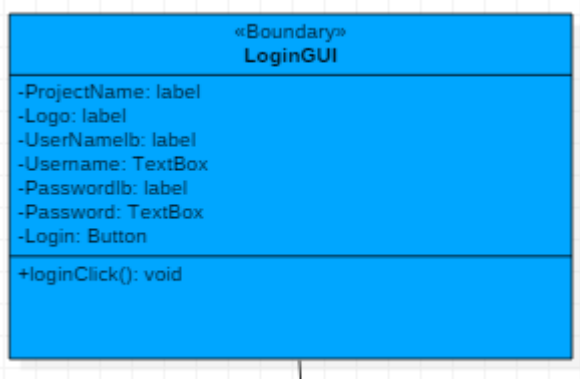
- userName : Το όνομα του διαχειριστή.
- userPassword: Ο κωδικός του διαχειριστή.
- userID : Χαρακτηριστικός αριθμός του διαχειριστή.

Μέθοδοι της κλάσης:

- setUserID(userID:integer): Η συνάρτηση αυτή θέτει τον χαρακτηριστικό αριθμό του διαχειριστή σύμφωνα με το όρισμα που δέχεται.
- setUserName(userName:string): Η συνάρτηση αυτή θέτει το όνομα διαχειριστή σύμφωνα με την συμβολοσειρά που λαμβάνει ως όρισμα.
- setUserPassword(userPassword:string): Η συνάρτηση αυτή θέτει το κωδικό πρόσβασης διαχειριστή σύμφωνα με την συμβολοσειρά που λαμβάνει ως όρισμα.

- getUserID() : Η συνάρτηση αυτή επιστρέφει τον χαρακτηριστικό αριθμό του διαχειριστή.
- getUserName(): Η συνάρτηση αυτή επιστρέφει το όνομα του διαχειριστή.

LoginGUI



Η κλάση αυτή εκφράζει την διεπαφή της σελίδας σύνδεσης του διαχειριστή με το σύστημα.

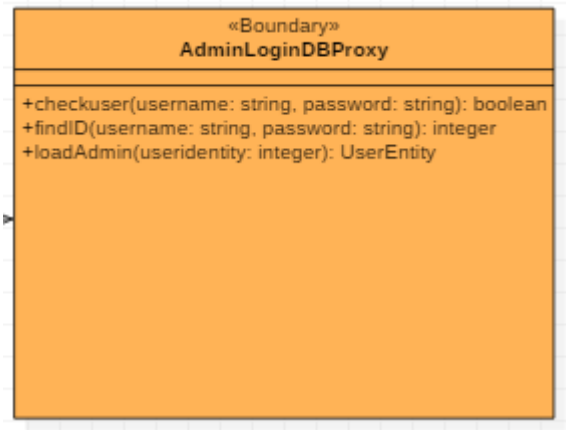
Χαρακτηριστικά της κλάσης:

- ProjectName: Τίτλος της εφαρμογής.
- Logo: Λογότυπο της εφαρμογής.
- login : Κουμπί για την είσοδο του διαχειριστή στην εφαρμογή του συστήματος
- UserNamelb: Επιγραφή πάνω από το πεδίο συμπλήρωσης του ονόματος χρήστη του διαχειριστή.
- Username : Πεδίο συμπλήρωσης του ονόματος χρήστη του διαχειριστή.
- Passwordlb: Επιγραφή πάνω από το πεδίο συμπλήρωσης του κωδικού πρόσβασης του διαχειριστή.
- Password: Πεδίο συμπλήρωσης του κωδικού πρόσβασης του διαχειριστή.

Μέθοδοι της κλάσης:

- loginClick(): Η μέθοδος αυτή καλεί τη συνάρτηση checkdata() του ελεγκτή LoginLogoutAdmin προκειμένου να γίνει ο έλεγχος των στοιχείων που έχει δώσει ο διαχειριστής.

AdminLogicDBProxy

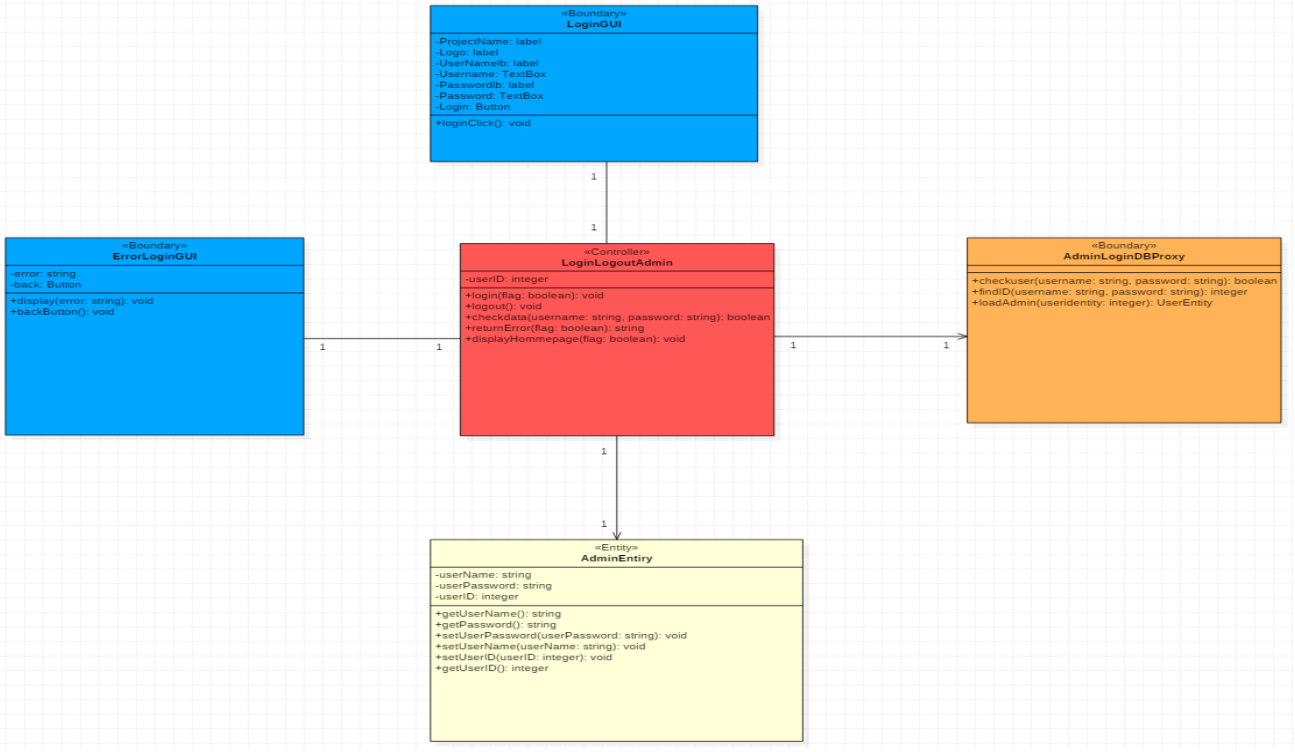


Η κλάση αυτή χρησιμοποιείται για την επικοινωνία με τη βάση δεδομένων του συστήματος όπου είναι αποθηκευμένα τα ονόματα και οι χαρακτηριστικοί αριθμοί των διαχειριστών του συστήματος.

Μέθοδοι της κλάσης:

- checkuser(username:string, password:string): Η μέθοδος αυτή ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης που έχει δοθεί στο σύστημα από τον διαχειριστή.
- findID(username:string, password:string): Η μέθοδος αυτή ψάχνει και επιστρέφει το συγκεκριμένο χαρακτηριστικό κωδικό με βάση τα στοιχεία που δέχεται ως όρισμα.
- loadAdmin(useridentity:integer): Η μέθοδος αυτή επιστρέφει έναν διαχειριστή-χρήστη του συστήματος με βάση το μοναδικό αναγνωριστικό που δέχεται ως όρισμα.

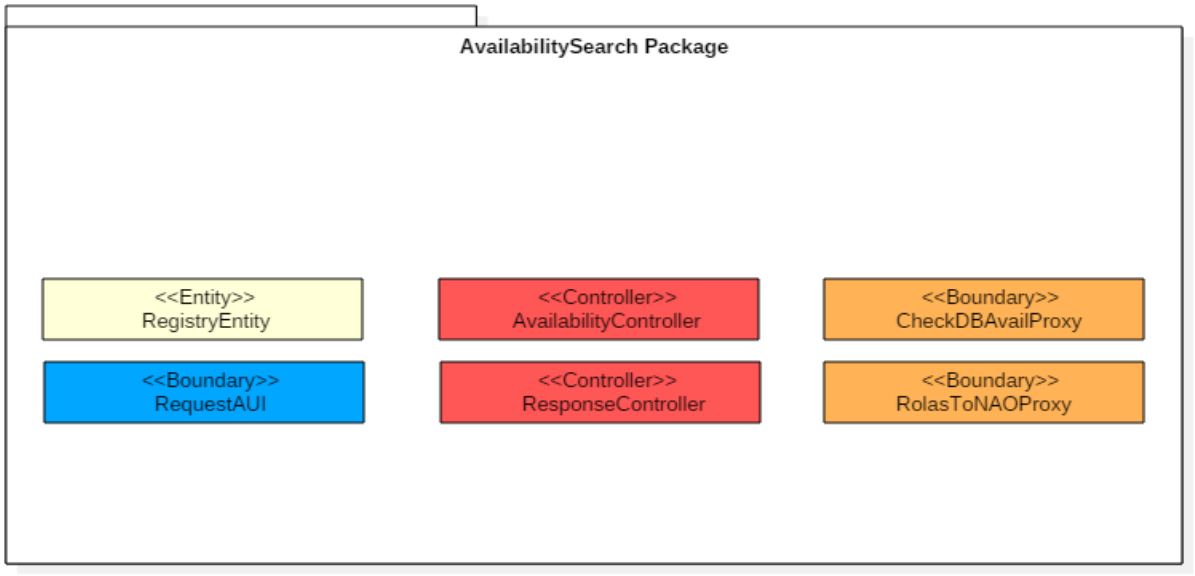
1.1.7.1 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ



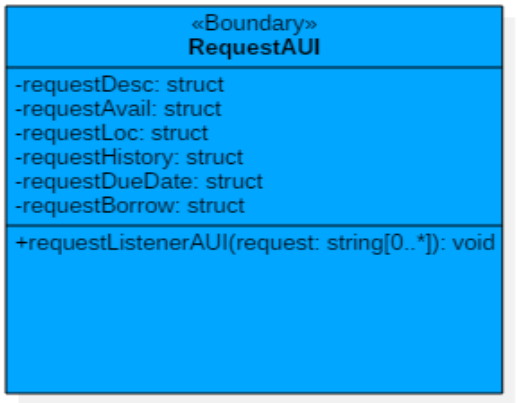
1.2 Πακέτα λεξιλογίου σεναρίων μέσης προτεραιότητας

1.2.1 AvailabilitySearch Pachage

Το πακέτο αυτό αναφέρεται στη διαδικασία αναζήτησης της διαθεσιμότητας ενός βιβλίου, στο χώρο της βιβλιοθήκης, από εγγεγραμμένους χρήστες ή επισκέπτες. Στη συνέχεια παρουσιάζονται οι κλάσεις οντοτήτων, οριακές και ελέγχου που εμπλέκονται στο συγκεκριμένο σενάριο μέσης προτεραιότητας και οι συσχετίσεις τους.



Boundary RequestAUI



Η κλάση αυτή εκφράζει την ακουστική διεπαφή χρήστη-Audio User Interface (AUI) στο σύστημα του προγράμματος.

Χαρακτηριστικά της κλάσης:

- requestDesc:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείτε (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Περιγραφής (στο activation).
- requestAvail:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείτε (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Διαθεσιμότητας (στο activation).
- requestLoc:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείτε (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Τοποθεσίας (στο activation).

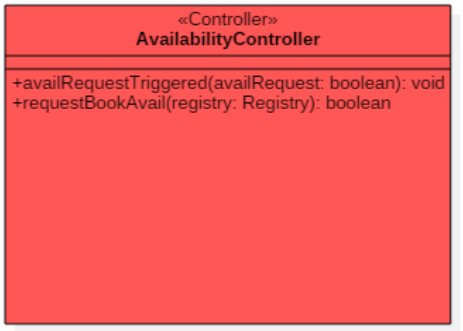
- requestHistory:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείτε (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Ενημέρωση Ιστορικού Δανεισμών (στο activation).
- requestDueDate:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείτε (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Ενημέρωση Ληξιπρόθεσμων Οφειλών (στο activation).
- requestBorrow:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείτε (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αίτημα Δανεισμού (στο activation).

*Ο τύπος struct σε αυτήν την περίπτωση έχει τη συγκεκριμένη δομή:
title:string[], activation:boolean

Μέθοδοι της κλάσης:

- requestListenerAUI(request:string[]):void : Η μέθοδος αυτή δέχεται ως όρισμα μία από τις πιθανές εντολές του χρήστη και ενημερώνει τα χαρακτηριστικά της κλάσης, ώστε να γίνεται γνωστό ποια επιθυμία του χρήστη ζητείται κάθε φορά και με ποια συγκεκριμένα χαρακτηριστικά.

Controller AvailabilityController

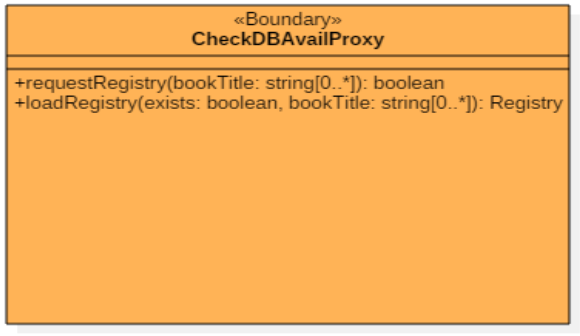


Ο ελεγκτής αυτός περιέχει συναρτήσεις για τον έλεγχο διαθεσιμότητας ενός βιβλίου.

Μέθοδοι της κλάσης:

- availRequestTriggered(availRequest:boolean):void : Η συνάρτηση αυτή ελέγχει αν ενεργοποιήθηκε από τον χρήστη αίτημα για αναζήτηση διαθεσιμότητας ενός βιβλίου.
- requestBookAvail(registry:Registry):boolean : Η συνάρτηση αυτή ελέγχει την εγγραφή για να διαπιστώσει αν το βιβλίο που αναζητείται είναι διαθέσιμο.

Boundary CheckDBAvailProxy

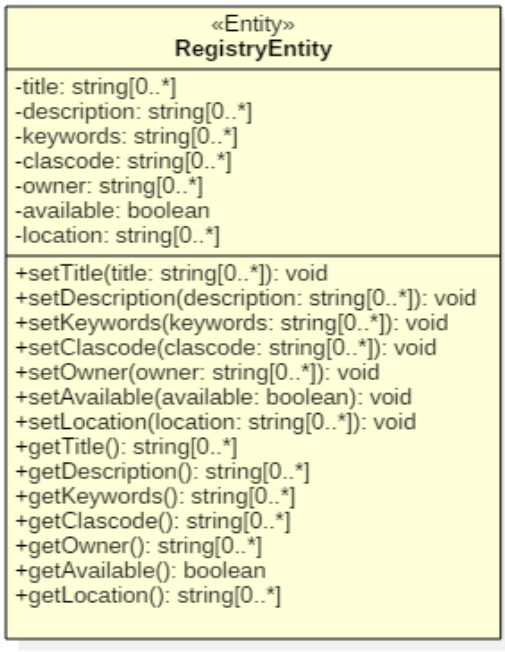


Η συγκεκριμένη κλάση επικοινωνεί με τη βάση δεδομένων και χρησιμοποιείται προκειμένου να αναζητήσει τη διαθεσιμότητα ενός βιβλίου σε αυτή.

Μέθοδοι της κλάσης:

- requestRegistry(bookTitle:string[]):boolean : Η συνάρτηση αυτή επιστρέφει αν το βιβλίο που αναζητείται υπάρχει ή όχι στη βάση δεδομένων της βιβλιοθήκης.
- loadRegistry(exists:boolean, bookTitle:string[]):Registry : Η συνάρτηση αυτή επιστρέφει την εγγραφή που υπάρχει στη βάση δεδομένων της βιβλιοθήκης και αναζητείται με βάση τον τίτλο του βιβλίου (αν η εγγραφή υπάρχει).

Entity RegistryEntity



Η συγκεκριμένη κλάση οντότητας περιέχει όλες τις ιδιότητες που περιέχει και μια εγγραφή στη Lib_Database, καθώς και μεθόδους για τον προσδιορισμό των τιμών των ιδιοτήτων αυτών.

Χαρακτηριστικά της κλάσης:

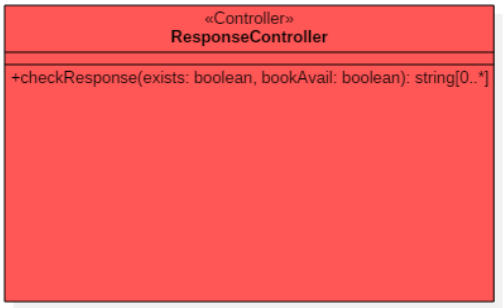
- title:string[] : Περιέχει τον τίτλο του βιβλίου που αντιστοιχεί η εγγραφή.
- description:string[] : Περιέχει την περιγραφή του βιβλίου.
- keywords:string[] : Περιέχει λέξεις κλειδιά του βιβλίου.
- clascode:string[] : Περιέχει τον Τ.Κ. (Ταξινομικό Κωδικό) του βιβλίου.
- owner:string[] : Περιέχει το χρήστη που έχει δανειστεί το βιβλίο.
- available:boolean : Περιέχει πληροφορία για το αν έχει δανειστεί ή όχι το βιβλίο.
- location:string[] : Περιέχει την τοποθεσία του βιβλίου στο χώρο της βιβλιοθήκης.

Μέθοδοι της κλάσης:

- setTitle(title:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του τίτλου του βιβλίου που αντιστοιχεί η εγγραφή.
- setDescription(description:string[]):void : Η συνάρτηση αυτή θέτει την τιμή της περιγραφής του βιβλίου.
- setKeywords(keywords:string[]):void : Η συνάρτηση αυτή θέτει την τιμή των λέξεων κλειδιών του βιβλίου.
- setClascode(clascode:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του Τ.Κ. του βιβλίου.

- setOwner(owner:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του δανειστή του βιβλίου.
- setAvailable(available:boolean):void : Η συνάρτηση αυτή θέτει την τιμή της διαθεσιμότητας του βιβλίου.
- setLocation(location:string[]):void : Η συνάρτηση αυτή θέτει την τιμή της τοποθεσίας του βιβλίου.
- getTitle():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του τίτλου του βιβλίου που αντιστοιχεί η εγγραφή.
- getDescription():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή της περιγραφής του βιβλίου.
- getKeywords():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή των λέξεων κλειδιών του βιβλίου.
- getClascode():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του Τ.Κ. του βιβλίου.
- getOwner():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του δανειστή του βιβλίου.
- getAvailable():boolean : Η συνάρτηση αυτή επιστρέφει την τιμή της διαθεσιμότητας του βιβλίου.
- getLocation():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή της τοποθεσίας του βιβλίου.

Controller ResponseController

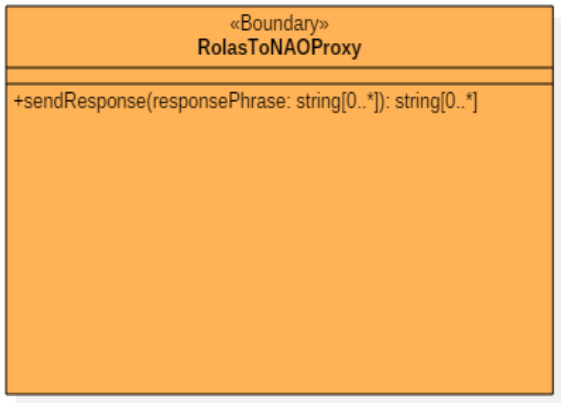


Ο ελεγκτής αυτός αποφασίζει ποια φράση θα εκφωνηθεί από το σύστημα.

Μέθοδοι της κλάσης:

- checkResponse(exists:boolean, bookAvail:boolean):string[] : Η συνάρτηση αυτή ελέγχει ποια είναι η κατάσταση του βιβλίου και αναλόγως στέλνει και στην περιφερειακή κλάση την κατάλληλη φράση για να εκφωνηθεί από το NAO.

Boundary RolasToNAOProxy

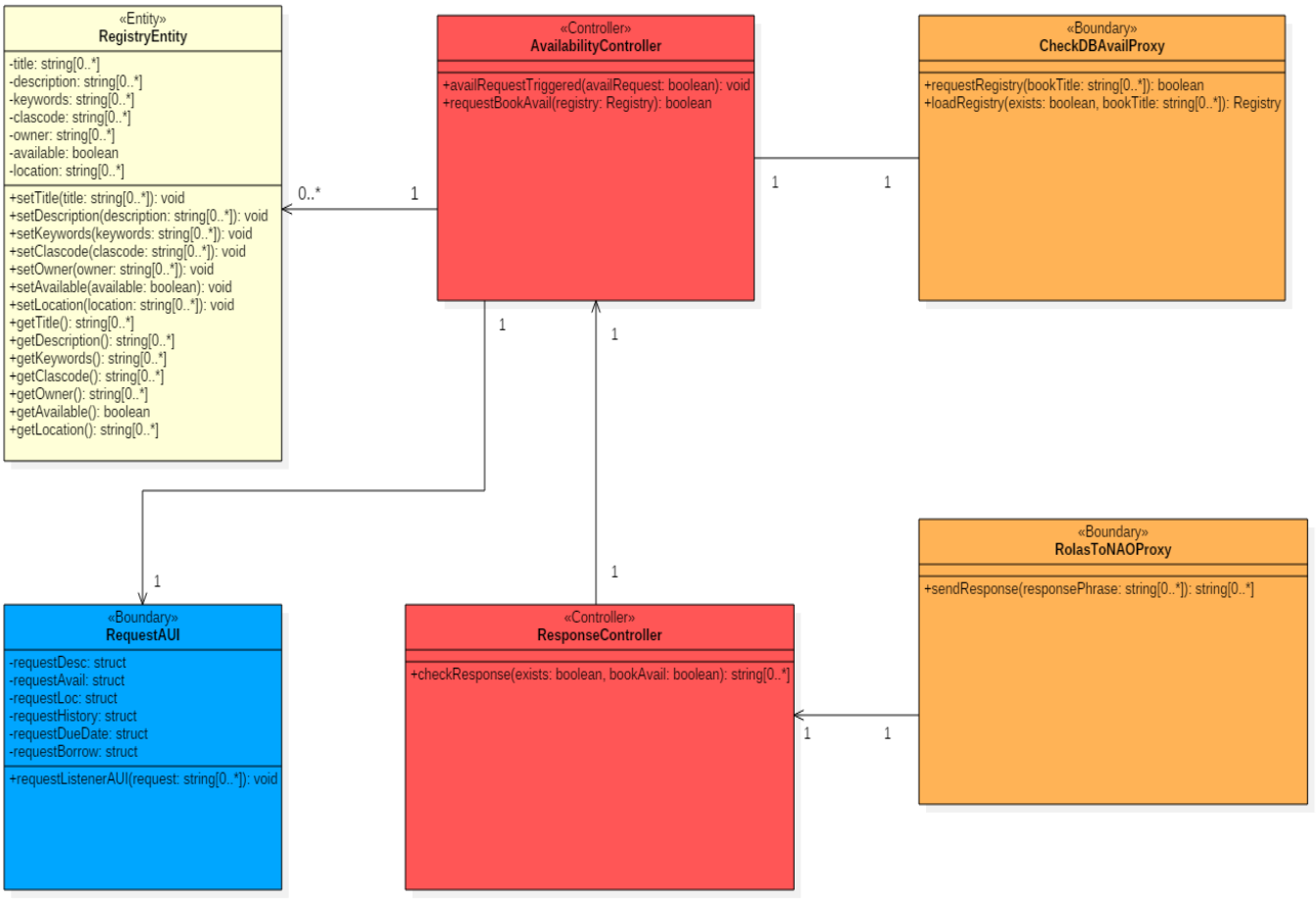


Ο ελεγκτής αυτός στέλνει στο NAO την φράση που θα εκφωνηθεί.

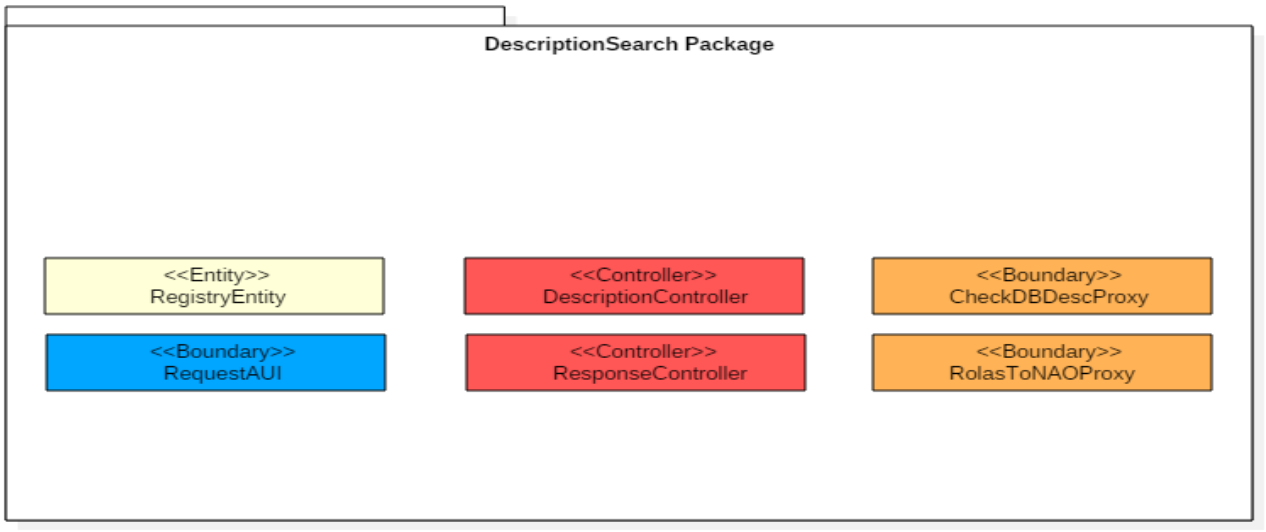
Μέθοδοι της κλάσης:

- sendResponse(responsePhrase:string[]):string[] : Η συνάρτηση αυτή στέλνει στο εξωτερικό σύστημα NAO τη φράση που θα κληθεί να εκφωνήσει στο χρήστη.

1.2.1.1 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ

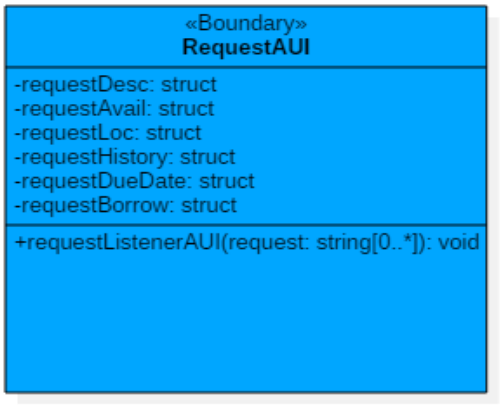


1.2.2 DescriptionSearch Package



Το πακέτο αυτό αναφέρεται στη διαδικασία αναζήτησης της περιγραφής ενός βιβλίου, στο χώρο της βιβλιοθήκης, από εγγεγραμμένους χρήστες ή επισκέπτες. Στη συνέχεια παρουσιάζονται οι κλάσεις οντοτήτων, οριακές και ελέγχου που εμπλέκονται στο συγκεκριμένο σενάριο μέσης προτεραιότητας και οι συσχετίσεις τους.

Boundary RequestAUI



Η κλάση αυτή εκφράζει την ακουστική διεπαφή χρήστη-Audio User Interface (AUI) στο σύστημα του προγράμματος.

Χαρακτηριστικά της κλάσης:

- requestDesc:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Περιγραφής (στο activation).
- requestAvail:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Διαθεσιμότητας (στο activation).
- requestLoc:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Τοποθεσίας (στο activation).
- requestHistory:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Ενημέρωση Ιστορικού Δανεισμών (στο activation).
- requestDueDate:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Ενημέρωση Ληξιπρόθεσμων Οφειλών (στο activation).
- requestBorrow:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αίτημα Δανεισμού (στο activation).

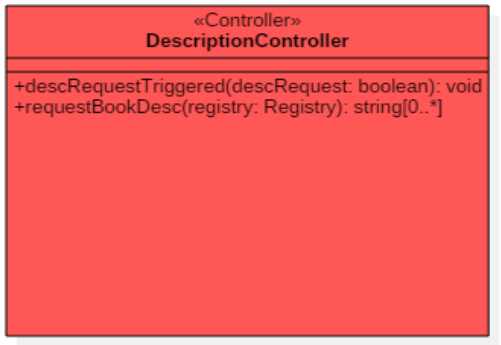
*Ο τύπος struct σε αυτήν την περίπτωση έχει τη συγκεκριμένη δομή:

title:string[], activation:boolean

Μέθοδοι της κλάσης:

- requestListenerAUI(request:string[]):void : Η μέθοδος αυτή δέχεται ως όρισμα μία από τις πιθανές εντολές του χρήστη και ενημερώνει τα χαρακτηριστικά της κλάσης, ώστε να γίνεται γνωστό ποια επιθυμία του χρήστη ζητείται κάθε φορά και με ποια συγκεκριμένα χαρακτηριστικά.

Controller DescriptionController

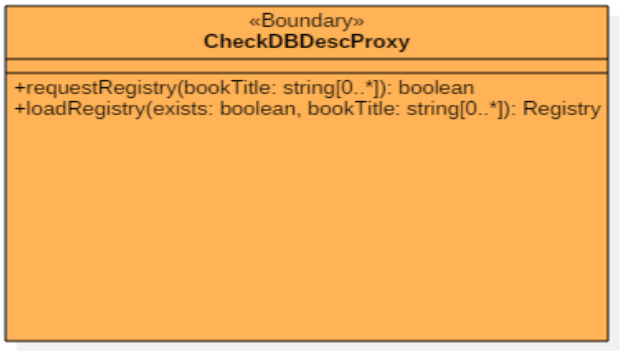


Ο ελεγκτής αυτός περιέχει συναρτήσεις για την αναζήτηση της περιγραφής ενός βιβλίου.

Μέθοδοι της κλάσης:

- availRequestTriggered(descRequest:boolean):void : Η συνάρτηση αυτή ελέγχει αν ενεργοποιήθηκε από τον χρήστη αίτημα για αναζήτηση περιγραφής ενός βιβλίου.
- requestBookAvail(registry:Registry):string[] : Η συνάρτηση αυτή ελέγχει την εγγραφή για να ανακτήσει την περιγραφή του βιβλίου που αναζητείται.

Boundary CheckDBDescProxy

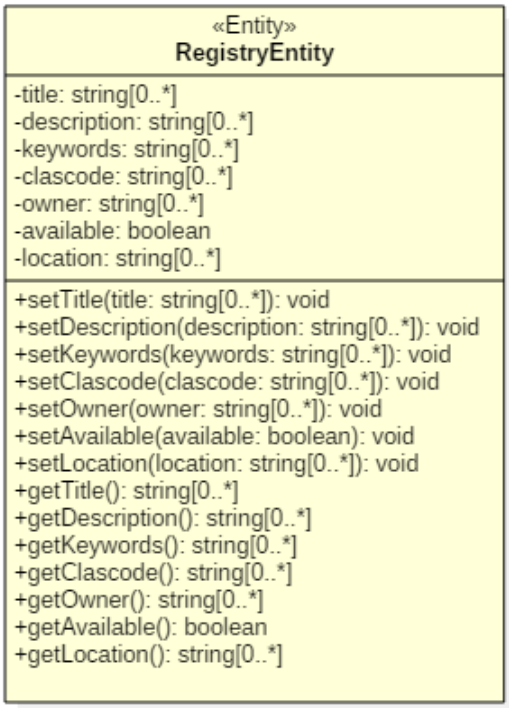


Η συγκεκριμένη κλάση επικοινωνεί με τη βάση δεδομένων και χρησιμοποιείται προκειμένου να αναζητήσει την περιγραφή ενός βιβλίου.

Μέθοδοι της κλάσης:

- requestRegistry(bookTitle:string[]):boolean : Η συνάρτηση αυτή επιστρέφει αν το βιβλίο που αναζητείται υπάρχει ή όχι στη βάση δεδομένων της βιβλιοθήκης.
- loadRegistry(exists:boolean, bookTitle:string[]):Registry : Η συνάρτηση αυτή επιστρέφει την εγγραφή που υπάρχει στη βάση δεδομένων της βιβλιοθήκης και αναζητείται με βάση τον τίτλο του βιβλίου (αν η εγγραφή υπάρχει).

Entity RegistryEntity



Η συγκεκριμένη κλάση οντότητας περιέχει όλες τις ιδιότητες που περιέχει και μια εγγραφή στη Lib_Database, καθώς και μεθόδους για τον προσδιορισμό των τιμών των ιδιοτήτων αυτών.

Χαρακτηριστικά της κλάσης:

- title:string[] : Περιέχει τον τίτλο του βιβλίου που αντιστοιχεί η εγγραφή.
- description:string[] : Περιέχει την περιγραφή του βιβλίου.
- keywords:string[] : Περιέχει λέξεις κλειδιά του βιβλίου.
- clascode:string[] : Περιέχει τον Τ.Κ. (Ταξινομικό Κωδικό) του βιβλίου.
- owner:string[] : Περιέχει το χρήστη που έχει δανειστεί το βιβλίο.
- available:boolean : Περιέχει πληροφορία για το αν έχει δανειστεί ή όχι το βιβλίο.
- location:string[] : Περιέχει την τοποθεσία του βιβλίου στο χώρο της βιβλιοθήκης.

Μέθοδοι της κλάσης:

- setTitle(title:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του τίτλου του βιβλίου που αντιστοιχεί η εγγραφή.
- setDescription(description:string[]):void : Η συνάρτηση αυτή θέτει την τιμή της περιγραφής του βιβλίου.
- setKeywords(keywords:string[]):void : Η συνάρτηση αυτή θέτει την τιμή των λέξεων κλειδιών του βιβλίου.
- setClascode(clascode:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του Τ.Κ. του βιβλίου.
- setOwner(owner:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του δανειστή του βιβλίου.
- setAvailable(available:boolean):void : Η συνάρτηση αυτή θέτει την τιμή της διαθεσιμότητας του βιβλίου.
- setLocation(location:string[]):void : Η συνάρτηση αυτή θέτει την τιμή της τοποθεσίας του βιβλίου.
- getTitle():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του τίτλου του βιβλίου που αντιστοιχεί η εγγραφή.
- getDescription():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή της περιγραφής του βιβλίου.
- getKeywords():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή των λέξεων κλειδιών του βιβλίου.
- getClascode():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του Τ.Κ. του βιβλίου.

- getOwner():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του δανειστή του βιβλίου.
- getAvailable():boolean : Η συνάρτηση αυτή επιστρέφει την τιμή της διαθεσιμότητας του βιβλίου.
- getLocation():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή της τοποθεσίας του βιβλίου.

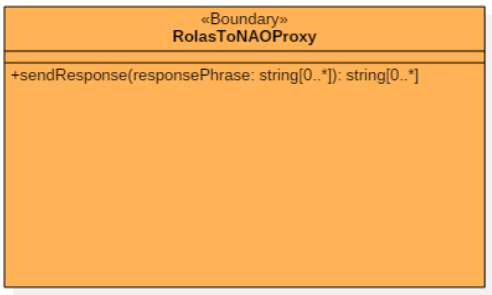
Controller ResponseController

Ο ελεγκτής αυτός αποφασίζει ποια φράση θα εκφωνηθεί από το σύστημα.

Μέθοδοι της κλάσης:

checkResponse(exists:boolean, bookDesc:string[]):string[] : Η συνάρτηση αυτή ελέγχει ποια είναι η κατάσταση του βιβλίου και αναλόγως στέλνει και στην περιφερειακή κλάση την κατάλληλη φράση για να εκφωνηθεί από το NAO.

Boundary RolasToNAOProxy

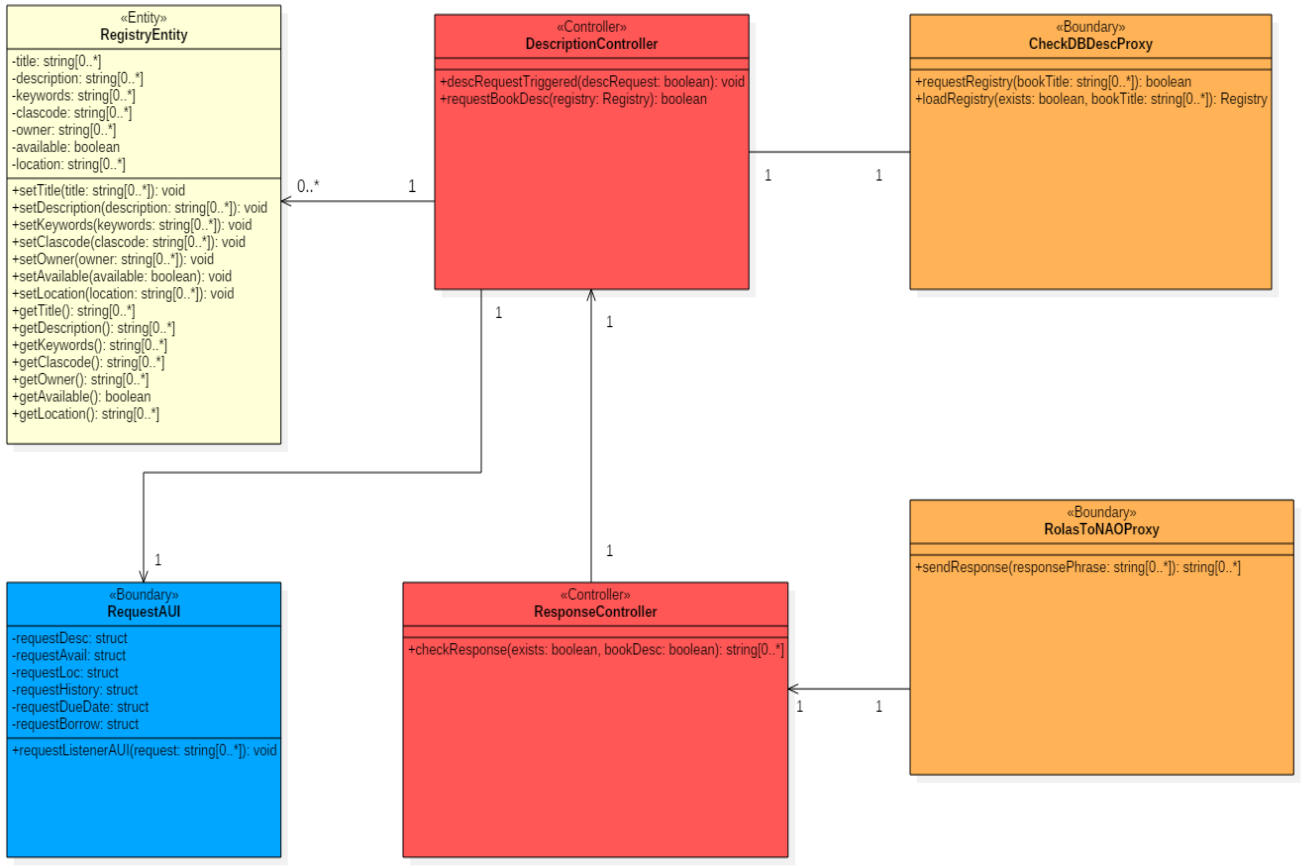


Ο ελεγκτής αυτός στέλνει στο NAO την φράση που θα εκφωνηθεί.

Μέθοδοι της κλάσης:

- sendResponse(responsePhrase:string[]):string[] : Η συνάρτηση αυτή στέλνει στο εξωτερικό σύστημα NAO τη φράση που θα κληθεί να εκφωνήσει στο χρήστη.

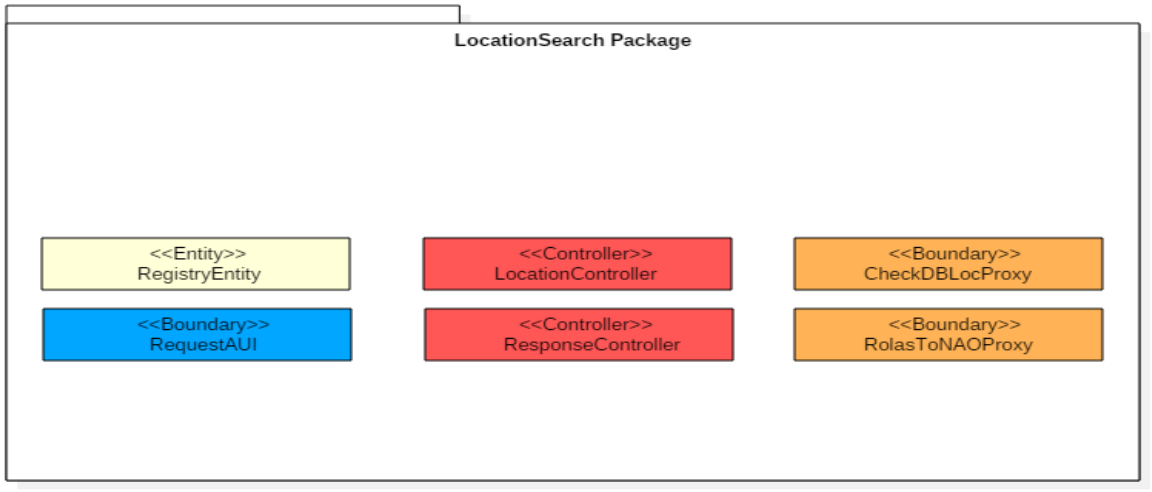
1.2.2.1 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ



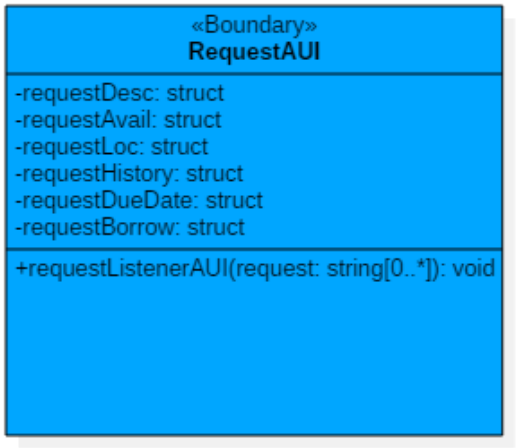
1.3 Πακέτα λεξιλογίου σεναρίων χαμηλής προτεραιότητας

1.3.1 LocationSearch Package

Το πακέτο αυτό αναφέρεται στη διαδικασία αναζήτησης της τοποθεσίας ενός βιβλίου, στο χώρο της βιβλιοθήκης, από εγγεγραμμένους χρήστες ή επισκέπτες. Στη συνέχεια παρουσιάζονται οι κλάσεις οντοτήτων, οριακές και ελέγχου που εμπλέκονται στο συγκεκριμένο σενάριο χαμηλής προτεραιότητας και οι συσχετίσεις τους.



Boundary RequestAUI



Η κλάση αυτή εκφράζει την ακουστική διεπαφή χρήστη-Audio User Interface (AUI) στο σύστημα του προγράμματος.

Χαρακτηριστικά της κλάσης:

- requestDesc:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Περιγραφής (στο activation).
- requestAvail:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Διαθεσιμότητας (στο activation).
- requestLoc:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αναζήτηση Τοποθεσίας (στο activation).
- requestHistory:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Ενημέρωση Ιστορικού Δανεισμών (στο activation).
- requestDueDate:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Ενημέρωση Ληξιπρόθεσμων Οφειλών (στο activation).
- requestBorrow:struct : Παίρνει τον τίτλο του βιβλίου που επιθυμείται (στο title) και ενημερώνεται ότι ενεργοποιήθηκε η εντολή Αίτημα Δανεισμού (στο activation).

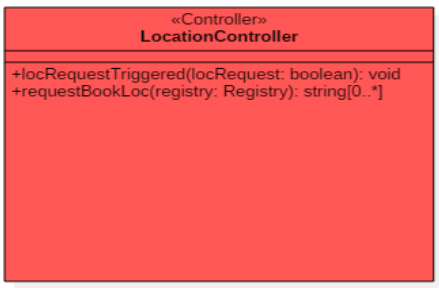
*Ο τύπος struct σε αυτήν την περίπτωση έχει τη συγκεκριμένη δομή:

title:string[], activation:boolean

Μέθοδοι της κλάσης:

requestListenerAUI(request:string[]):void : Η μέθοδος αυτή δέχεται ως όρισμα μία από τις πιθανές εντολές του χρήστη και ενημερώνει τα χαρακτηριστικά της κλάσης, ώστε να γίνεται γνωστό ποια επιθυμία του χρήστη ζητείται κάθε φορά και με ποια συγκεκριμένα χαρακτηριστικά.

Controller LocationController

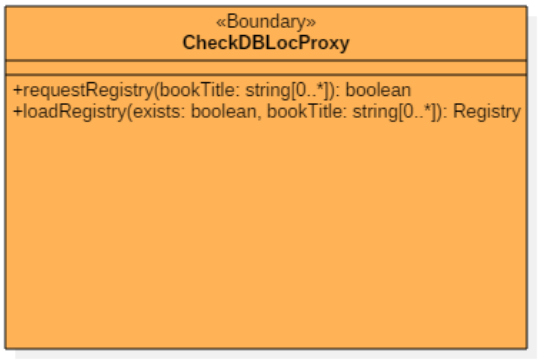


Ο ελεγκτής αυτός περιέχει συναρτήσεις για την αναζήτηση της τοποθεσίας ενός βιβλίου.

Μέθοδοι της κλάσης:

- locRequestTriggered(locRequest:boolean):void : Η συνάρτηση αυτή ελέγχει αν ενεργοποιήθηκε από τον χρήστη αίτημα για αναζήτηση της τοποθεσίας ενός βιβλίου.
- requestBookLoc(registry:Registry):string[] : Η συνάρτηση αυτή ελέγχει την εγγραφή για να ανακτήσει την τοποθεσία του βιβλίου που αναζητείται.

Boundary CheckDBLocProxy

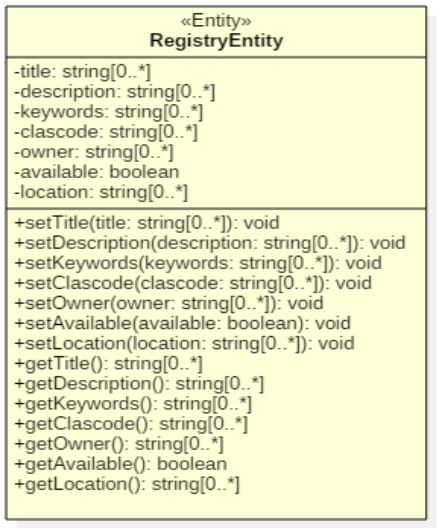


Η συγκεκριμένη κλάση επικοινωνεί με τη βάση δεδομένων και χρησιμοποιείται προκειμένου να αναζητήσει τη τοποθεσία ενός βιβλίου.

Μέθοδοι της κλάσης:

- requestRegistry(bookTitle:string[]):boolean : Η συνάρτηση αυτή επιστρέφει αν το βιβλίο που αναζητείται υπάρχει ή όχι στη βάση δεδομένων της βιβλιοθήκης.
- loadRegistry(exists:boolean, bookTitle:string[]):Registry : Η συνάρτηση αυτή επιστρέφει την εγγραφή που υπάρχει στη βάση δεδομένων της βιβλιοθήκης και αναζητείται με βάση τον τίτλο του βιβλίου (αν η εγγραφή υπάρχει).

Entity RegistryEntity



Η συγκεκριμένη κλάση οντότητας περιέχει όλες τις ιδιότητες που περιέχει και μια εγγραφή στη Lib_Database, καθώς και μεθόδους για τον προσδιορισμό των τιμών των ιδιοτήτων αυτών.

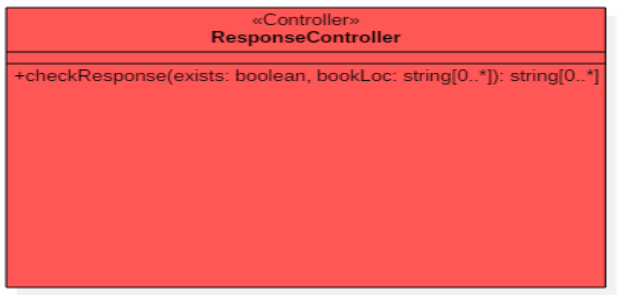
Χαρακτηριστικά της κλάσης:

- title:string[] : Περιέχει τον τίτλο του βιβλίου που αντιστοιχεί η εγγραφή.
- description:string[] : Περιέχει την περιγραφή του βιβλίου.
- keywords:string[] : Περιέχει λέξεις κλειδιά του βιβλίου.
- clascode:string[] : Περιέχει τον Τ.Κ. (Ταξινομικό Κωδικό) του βιβλίου.
- owner:string[] : Περιέχει το χρήστη που έχει δανειστεί το βιβλίο.
- available:boolean : Περιέχει πληροφορία για το αν έχει δανειστεί ή όχι το βιβλίο.
- location:string[] : Περιέχει την τοποθεσία του βιβλίου στο χώρο της βιβλιοθήκης.

Μέθοδοι της κλάσης:

- setTitle(title:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του τίτλου του βιβλίου που αντιστοιχεί η εγγραφή.
- setDescription(description:string[]):void : Η συνάρτηση αυτή θέτει την τιμή της περιγραφής του βιβλίου.
- setKeywords(keywords:string[]):void : Η συνάρτηση αυτή θέτει την τιμή των λέξεων κλειδιών του βιβλίου.
- setClascode(clascode:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του Τ.Κ. του βιβλίου.
- setOwner(owner:string[]):void : Η συνάρτηση αυτή θέτει την τιμή του δανειστή του βιβλίου.
- setAvailable(available:boolean):void : Η συνάρτηση αυτή θέτει την τιμή της διαθεσιμότητας του βιβλίου.
- setLocation(location:string[]):void : Η συνάρτηση αυτή θέτει την τιμή της τοποθεσίας του βιβλίου.
- getTitle():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του τίτλου του βιβλίου που αντιστοιχεί η εγγραφή.
- getDescription():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή της περιγραφής του βιβλίου.
- getKeywords():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή των λέξεων κλειδιών του βιβλίου.
- getClascode():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του Τ.Κ. του βιβλίου.
- getOwner():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή του δανειστή του βιβλίου.
- getAvailable():boolean : Η συνάρτηση αυτή επιστρέφει την τιμή της διαθεσιμότητας του βιβλίου.
- getLocation():string[] : Η συνάρτηση αυτή επιστρέφει την τιμή της τοποθεσίας του βιβλίου.

Controller ResponseController

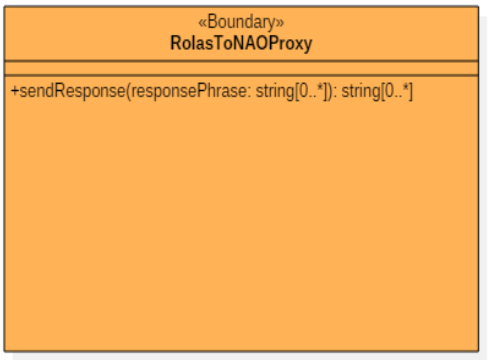


Ο ελεγκτής αυτός αποφασίζει ποια φράση θα εκφωνηθεί από το σύστημα.

Μέθοδοι της κλάσης:

- checkResponse(exists:boolean, bookLoc:string[]):string[] : Η συνάρτηση αυτή ελέγχει ποια είναι η κατάσταση του βιβλίου και αναλόγως στέλνει και στην περιφερειακή κλάση την κατάλληλη φράση για να εκφωνηθεί από το NAO.

Boundary RolasToNAOProxy

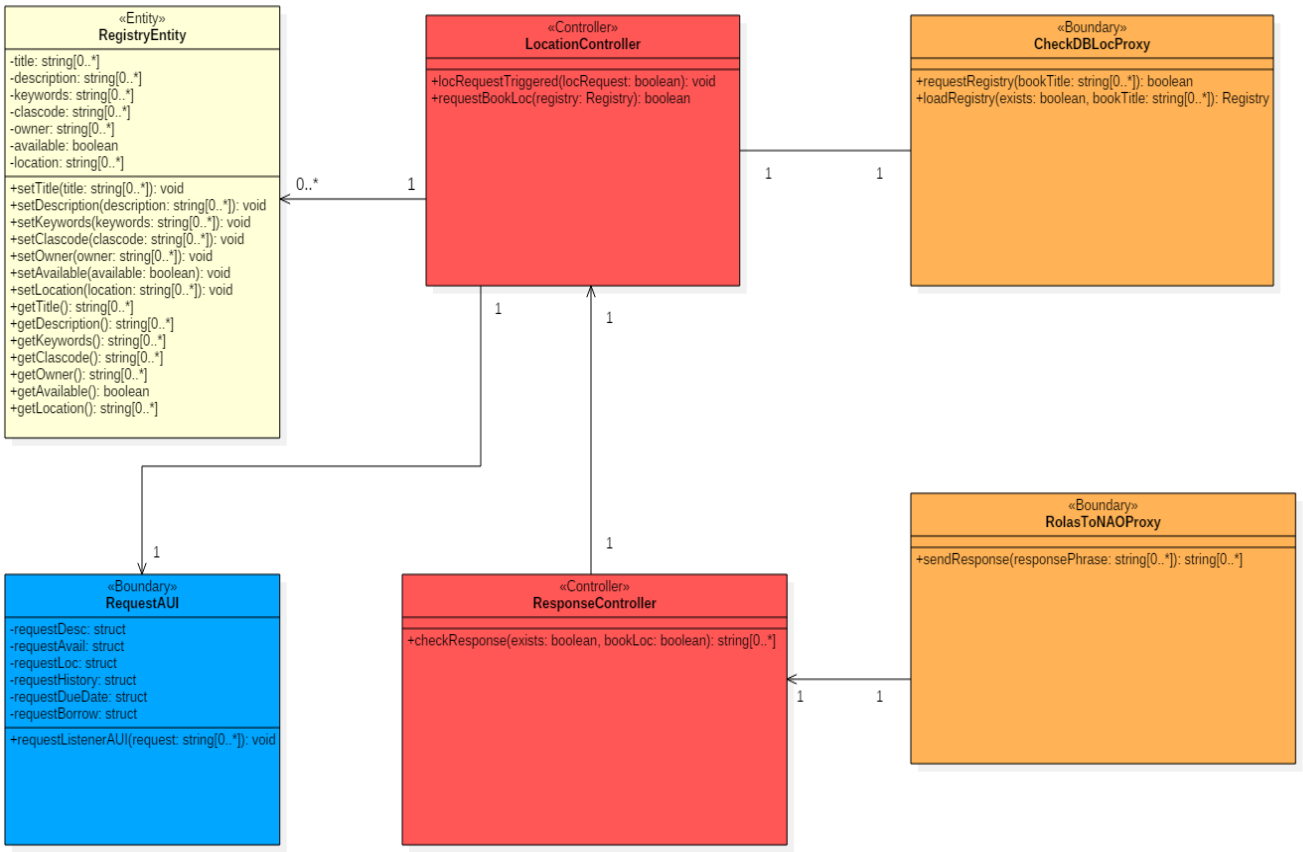


Ο ελεγκτής αυτός στέλνει στο NAO την φράση που θα εκφωνηθεί.

Μέθοδοι της κλάσης:

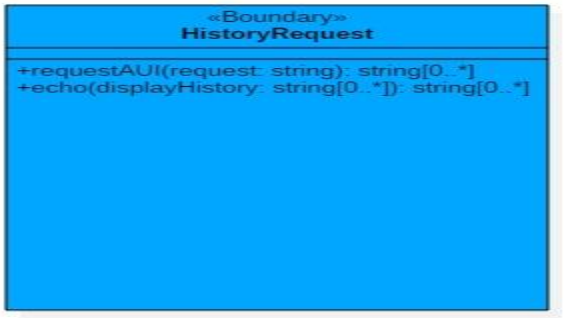
- sendResponse(responsePhrase:string[]):string[] : Η συνάρτηση αυτή στέλνει στο εξωτερικό σύστημα NAO τη φράση που θα κληθεί να εκφωνήσει στο χρήστη.

1.3.1.1 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ



1.3.2 HistoryOfBorrows Package

Boundary HistoryRequest



Η κλάση αυτή πραγματοποιεί την ενημέρωση του χρήστη από το ΝΑΟ για το ιστορικό δανεισμών του.

Μέθοδοι της κλάσης:

- requestAUI(request: string): string[0..*] : Η συνάρτηση αυτή αναμένει ως όρισμα την εκφώνηση της φωνητικής εντολής <<ΠΡΟΒΟΛΗ ΙΣΤΟΙΚΟΥ>> την οποία επιστρέφει ως string για να χρησιμοποιθεί σε επόμενες κλάσεις.
- echo(displayHistory:string[0..*]): string[0..*] : Η συνάρτηση αυτή πραγματοποιεί την ανακοίνωση του ιστορικού στον χρήστη.

Controller HistoryOfBorrowsControll

Ο ελεγκτής καλεί εκείνες τις συναρτήσεις που εκκινούν τις διαδικασίες για αναζήτηση του ιστορικού δανεισμών, ελέγχοντας ταυτόχρονα για ύπαρξη ή μη προηγούμενων δανεισμών.

Μέθοδοι της κλάσης:

- checkTypeRequest(request:string): string : Η συνάρτηση αυτή διαβάζει τον τύπο του αιτήματος που έχει καταχωρηθεί από το χρήστη .
- ReadUserDetails(): UserEntity : Η συνάρτηση αυτή επιστρέφει τα στοιχεία του χρήστη για τον οποίο αργότερα γίνεται η προβολή ιστορικού δανεισμών.
- resultAnnouncement(result: string[0..*]): string[0..*] : Η συνάρτηση αυτή αποθηκεύει το ιστορικό που έχει αναζητηθεί στη βάση δεδομένων, για να το κάνει αργότερα διαθέσιμο στο ΝΑΟ ώστε αυτός με τη σειρά του να το εκφωνήσει στο χρήστη.

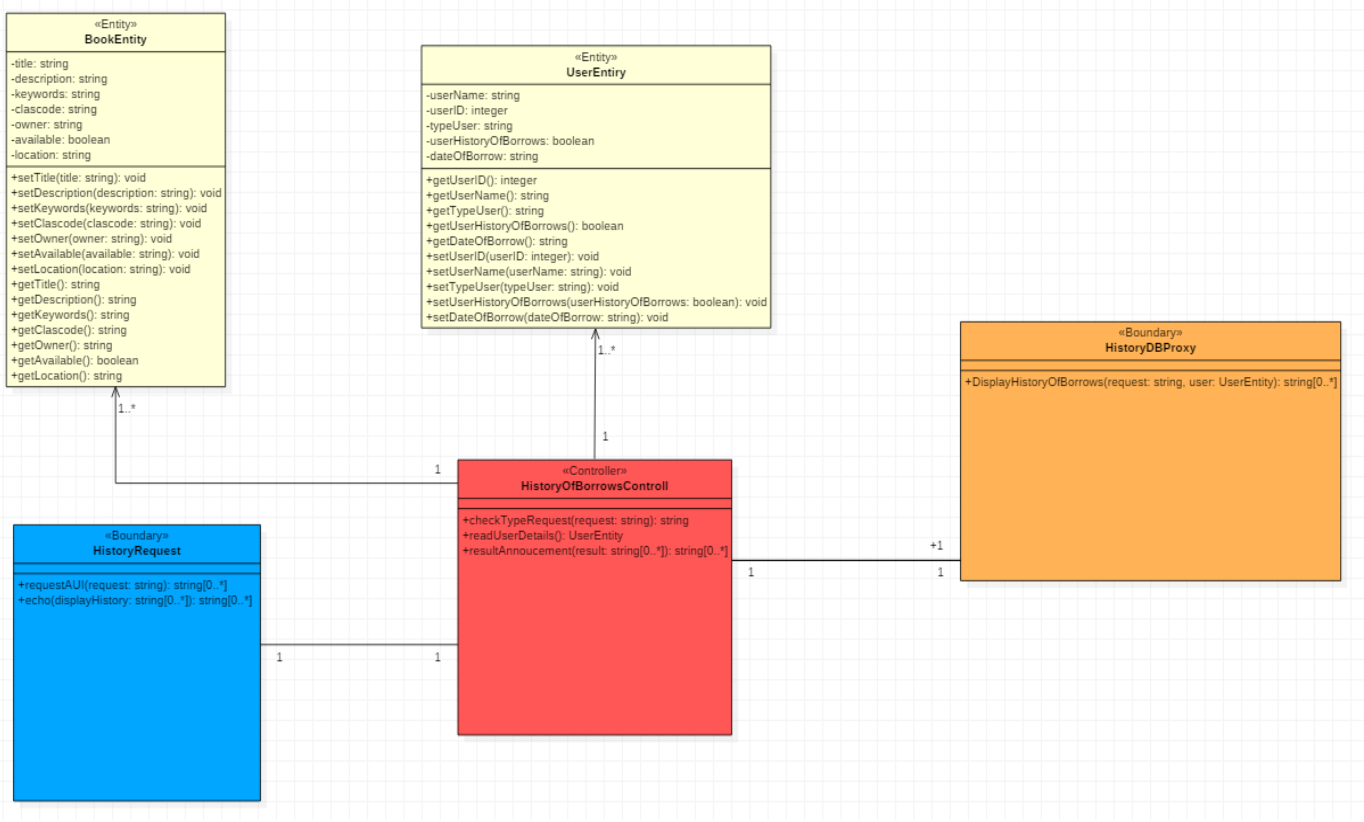
Boundary HistoryDBProxy

Η κλάση αυτή πραγματοποιεί με βάση τα στοιχεία που της δίνει ο ελεγκτής την αναζήτηση, αν αυτή είναι δυνατό να πραγματοποιηθεί, του ιστορικού δανεισμών.

Μέθοδοι της κλάσης:

- displayHistoryOBorrows(request: string, user: UserEntity): string[0..*] : Με βάση τα ορίσματα που δέχεται αυτή η μέθοδος πραγματοποιεί την αναζήτηση του ιστορικού στη βάση δεδομένων. Αυτή βέβαια η αναζήτηση θα γίνει εφόσον το χαρακτηριστικό userHistoryOfBorrows: boolean έχει τιμή 1, που δηλώνει πως έχουν γίνει προηγούμενοι δανεισμοί.

1.3.2.1 Διάγραμμα Κλάσεων



2 Μη λειτουργικές απαιτήσεις

2.1 Απαιτήσεις Επίδοσης

<ΜΛΑ-4>

Το σύστημα πρέπει να αποκρίνεται εντός 3 δευτερολέπτων μετά από κάθε φωνητική εντολή ενός χρήστη (εγγεγραμμένου ή μη χρήστη).

UserPriority: Η απαίτηση αυτή είναι πολύ σημαντική για τον χρήστη του συστήματος καθώς κάθε χρήστης επιθυμεί γρήγορη και άμεση απόκριση από το σύστημα με το οποίο αλληλοεπιδρά.

TechnicalPriority: Είναι πολύ σημαντικό το σύστημα να έχει μικρό χρόνο απόκρισης όπου εδώ ορίζεται στα 3 δευτερόλεπτα. Ο χρόνος απόκρισης του συστήματος εξαρτάται από την ταχύτητα αποστολής και παραλαβής των δεδομένων ανάμεσα στο ΝΑΟ και την Lib_Database. Οι υπολογιστικές ικανότητες του ΝΑΟ είναι προκαθορισμένες παρόλα αυτά το σύστημα διακομιστή θα πρέπει να εξοπλισμένο με χαρακτηριστικά που να επιτρέπουν ταχύτητες απόκρισης, οι οποίες να πληρούν τις προδιαγραφές του συστήματος .

Stability: Ο χρόνος απόκρισης του συστήματος μπορεί να μεταβληθεί η μεταβολή του αυτή εξαρτάται μόνο από τις επιδόσεις του συστήματος διακομιστή (καθώς οι επιδώσεις του ΝΑΟ είναι απόλυτα προκαθορισμένες). Οπότε σε περίπτωση αναβάθμισης του συστήματος του διακομιστή ο χρόνος των 3 δευτερολέπτων μπορεί να ελαττωθεί.

2.2 Απαιτήσεις Ασφάλειας (Safety)

<ΜΛΑ-5>

Το σύστημα πρέπει να αποσυνδέει τον εγγεγραμμένο χρήστη, που έχει συνδεθεί στο σύστημα, μετά από 15 συνεχόμενα δευτερόλεπτα κατά τα οποία δεν έχει δεχτεί κάποια εντολή από τον εγγεγραμμένο χρήστη.

UserPriority: Η απαίτηση αυτή είναι σημαντική για την ασφάλεια του εγγεγραμμένου χρήστη, που έχει συνδεθεί στο σύστημα , καθώς έτσι διασφαλίζεται ότι κανένας από τους υπόλοιπους επισκέπτες δεν θα προσπαθήσει να παραβιάσει το λογαριασμό του σε περίπτωση όπου ο πρώτος απομακρυνθεί από την εμβέλεια του ΝΑΟ χωρίς να κάνει αποσύνδεση. Για παράδειγμα θα μπορούσε να συμβεί στην περίπτωση όπου ένας εγγεγραμμένος που έχει κάνει εισαγωγή στο σύστημα φύγει από την βιβλιοθήκη χωρίς να κάνει αποσύνδεση από το σύστημα τότε κάποιος άλλος επισκέπτης της βιβλιοθήκης θα μπορούσε να δανειστεί κάποιο βιβλίο το οποίο θα χρέωνε στο λογαριασμό του πρώτου.

TechnicalPriority: Είναι πολύ σημαντικό χαρακτηριστικό του συστήματος καθώς έτσι διασφαλίζεται η ασφάλεια των εγγεγραμμένων χρηστών όταν αυτοί είναι συνδεδεμένοι στο σύστημα . Μέσω αυτού του μηχανισμού του συστήματος αποφεύγονται περιπτώσεις όπου μπορεί διαφορετικός εγγεγραμμένος χρήστης να δανειστεί βιβλίο το οποίο θα χρεωθεί σε λογαριασμό άλλου εγγεγραμμένου χρήστη, αλλοιώνοντας με αυτό τον τρόπο τις εγγραφές της Lib_Database.

Stability: Το χαρακτηριστικό αυτό είναι βασικό για το σύστημα καθώς διασφαλίζει την ασφάλεια των χρηστών που το χρησιμοποιούν. Η λειτουργία αυτή θα μπορούσε να παρέχει μεγαλύτερη ασφάλεια αν στην υλοποίηση της μελλοντικά χρησιμοποιούνταν προσθετικά και η δυνατότητα της τεχνητής όρασης του ΝΑΟ. Αυτό θα γινόταν με σκοπό το σύστημα αυτόματα να αποσυνδέει τον εγγεγραμμένο χρήστη, σε περίπτωση όπου απομακρυνθεί από το ΝΑΟ.

2.3 Απαιτήσεις Χρηστικότητας (Usability)

<ΜΛΑ- 1>

Το σύστημα πρέπει να μπορεί να προτείνει στον εγγεγραμμένο χρήστη βιβλία για να δανειστεί με βάση το ιστορικό δανεισμών του.

UserPriority: Είναι πολύ σημαντικό για τον εγγεγραμμένο χρήστη οι προτάσεις δανεισμού βιβλίων να είναι συναφείς με τη θεματολογία των βιβλίων που έχει δανειστεί κατά το παρελθόν προκειμένου η πιθανότητα του να τα δανειστεί να είναι μεγάλη. Επίσης σημαντικό είναι το σύστημα να μην προτείνει στον εγγεγραμμένο χρήστη βιβλία τα οποία έχει ήδη δανειστεί- διαβάσει.

TechnicalPriority: Η λειτουργία αυτή του συστήματος είναι πολύ χρηστική για τους εγγεγραμμένους χρήστες της βιβλιοθήκης. Για το λόγο αυτό η πρόταση σωστών βιβλίων προς δανεισμό, δηλαδή η αντιστοίχιση τους με παρόμοια βιβλία που ο εγγεγραμμένος χρήστης έχει δανειστεί κατά το παρελθόν είναι πολύ σημαντική και πρέπει να εκτελείται σε χρονικό διάστημα 3 δευτερολέπτων το σύστημα πρέπει να έχει αρχίσει να εκφωνεί στον εγγεγραμμένο χρήστη τη λίστα των προτεινόμενων βιβλίων που έχει δημιουργήσει για αυτόν. Αυτό επιτυγχάνεται μέσω των πληροφοριών περιγραφής του βιβλίου που είναι αποθηκευμένες στη Lib_Database σε συνδυασμό με το ιστορικό δανεισμού βιβλίων του εγγεγραμμένου χρήστη.

Stability: Μέσα από τη συνεχή βελτίωση του συστήματος ενδέχεται ο τρόπος περιγραφής των πληροφοριών των βιβλίων να γίνει πιο λεπτομερής και ο αλγόριθμος πρότασης βιβλίων να εξελιχθεί, με αποτέλεσμα οι «προτάσεις δανεισμού» του συστήματος να γίνουν πιο στοχευμένες και συνεπώς περισσότερο χρηστικές για τον εγγεγραμμένο χρήστη.

<ΜΛΑ-2>

Το σύστημα πρέπει να μπορεί να εκφωνεί στον εγγεγραμμένο χρήστη, μετά την ολοκλήρωση του δανεισμού του, τα 5 πιο συχνά δανειζόμενα βιβλία της βιβλιοθήκης για την περασμένη εβδομάδα.

UserPriority: Η λειτουργία αυτή είναι χαμηλής προτεραιότητας καθώς προσδίδει στο χρήστη μια επιπλέον λειτουργικότητα η οποία δεν είναι απαραίτητη, αλλά δίνει στον εγγεγραμμένο χρήστη επιπλέον δυνατότητες.

TechnicalPriority: Η απαίτηση αυτή είναι σημαντικό αμέσως μετά την ολοκλήρωση του δανεισμού το σύστημα να εκφωνεί τα 5 δημοφιλέστερα βιβλία (εντός 5 δευτερολέπτων).

Stability: Δεν ενδέχεται να μεταβληθεί αυτή η μη λειτουργική απαίτηση.

<ΜΛΑ-3>

Το σύστημα μπορεί να ενημερώνει τον εγγεγραμμένο χρήστη για την κατάσταση των βιβλίων που έχει δανειστεί μέσω μηνύματος ηλεκτρονικού ταχυδρομείου.

UserPriority: Η απαίτηση αυτή είναι σημαντική για τον εγγεγραμμένο χρήστη καθώς το σύστημα του δίνει τη δυνατότητα ενημέρωσης πριν λήξει ο δανεισμός των βιβλίων του οπότε τον προτρέπει να τα επιστρέψει αλλά και να γλιτώσει το επικείμενο πρόστιμο που σε περίπτωση καθυστερημένης επιστροφής του βιβλίου η βιβλιοθήκη θα του επιβάλει.

TechnicalPriority: Το σύστημα πρέπει να μπορεί να αποστείλει μήνυμα ηλεκτρονικού ταχυδρομείου στο εγγεγραμμένο χρήστη 3 ημέρες πριν τη λήξη των δανεισμένων βιβλίων του.

Stability: Ενδεχομένως στο μέλλον η ενημέρωση του εγγεγραμμένου χρήστη για την κατάσταση των δανεισμένων βιβλίων να γίνει πιο αποτελεσματική αν το σύστημα τον ενημερώνει με περισσότερα από ένα μηνύματα ηλεκτρονικού ταχυδρομείου στο διάστημα 5 ημερών πριν από την λήξη των δανεισμών του.

3 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

Δεν υπάρχει μία μοναδική σχεδίαση για ένα σύστημα λογισμικού. Η μελέτη διαφορετικών επιλογών και διαφορετικών προτύπων σχεδίασης είναι αναγκαία. Σε αυτό το μέρος του εγγράφου καθορίζουμε τα πρότυπα που χρησιμοποιούμε, ώστε να αντιμετωπίσουμε συγκεκριμένα προβλήματα στην αρχιτεκτονική μας.

Ένα πρότυπο σχεδίασης περιγράφει ένα πρόβλημα το οποίο συμβαίνει ξανά και ξανά στο περιβάλλον μας. Επίσης, περιγράφει τον πυρήνα της λύσης στο πρόβλημα αυτό με τέτοιο τρόπο, ώστε να μπορεί η λύση να επαναχρησιμοποιηθεί, χωρίς να ξανα-υλοποιηθεί.

3.1 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

3.1.1 Δομικά πρότυπα

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

Proxy Design Pattern

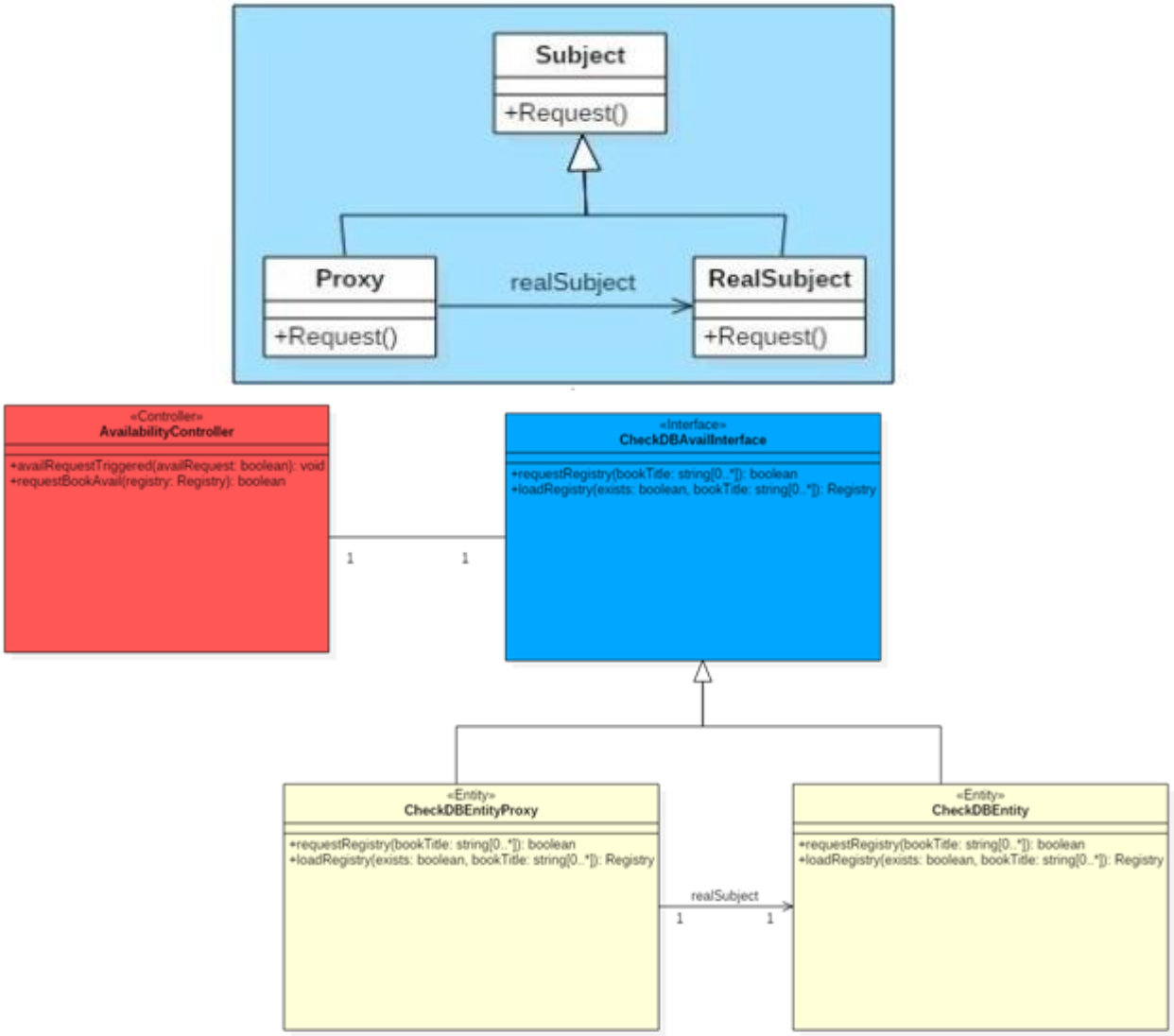
Το συγκεκριμένο μοτίβο παρέχει διαφάνεια ως προς την τοποθεσία και μειώνει το κόστος προσπέλασης αντικειμένων.

Χρησιμοποιείται ένα άλλο αντικείμενο, “το proxy”, το οποίο λειτουργεί ως αντικαταστάτης του πραγματικού αντικειμένου. Ο proxy δημιουργεί το πραγματικό αντικείμενο μόνο όταν το ζητήσει το χρήστης.

Πρόβλημα που αντιμετωπίστηκε:

Το σύστημα πρέπει να μπορεί να εκφωνεί στον εγγεγραμμένο χρήστη, μετά την ολοκλήρωση του δανεισμού του, τα 5 πιο συχνά δανειζόμενα βιβλία της βιβλιοθήκης για την περασμένη εβδομάδα.

Για να επιτυγχάνεται αυτή η Μη Λειτουργική Απαίτητηση χρησιμοποιούμε proxy και μειώνουμε το κόστος προσπέλασης των αντικειμένων, έχοντας το αντικείμενο CheckDBAvailability Proxy.



3.2.1 Δομικά πρότυπα

Adapter Design Pattern

Το πρότυπο του προσαρμογέα επιτρέπει τη συνεργασία κλάσεων που αλλιώς δεν θα μπορούσαν να συνεργαστούν, λόγω ασύμβατων διεπαφών. Χρησιμοποιείται για να παρέχει μια νέα διεπαφή σε υπάρχοντα συστήματα - γνωστό και ως wrapper (legacy components).

Συγκεκριμένα ο προσαρμογέας κλάσης (Class adapter):

Εφαρμόζει πολλαπλή κληρονομικότητα για να προσαρμόσει μια διεπαφή σε μια άλλη.

Πρόβλημα που αντιμετωπίστηκε:

Το πρόβλημα που αντιμετωπίστηκε ήταν η προσπέλαση της βάσης δεδομένων Lib_Database. Μεσώ της BorrowDBProxy, δινόταν ένα struct για να αλλάξει η κατάσταση δανεισμού στη βάση δεδομένων στη θέση της οποίας χρησιμοποιήθηκε μία νέα κλάση Interface, BookBorrowInterface. Το δομικό πρότυπο Adapter επέτρεψε τη αλληλεπίδραση των δύο διαφορετικών αυτών συστημάτων μέσω της κλάσης Adapter.