



Uni-Food

Because food matters

Απαιτήσεις Λογισμικού (Προδιαγραφές Λογισμικού)

Del.1.2

Version 1.0.6

Μαυροδής Κωνσταντίνος (kmavrodis@outlook.com)

Νήρας Δημήτρης (niras.94@gmail.com)

Σχινάς Γιώργος (schinas.georgios@outlook.com)

Χατζηθωμά Ανδρέας (andreas_h92@hotmail.com)

09 / 06 / 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	29/03/2014	Προσαρμογή του εγγράφου.	0.1.3
X. Ζολώτας	02/04/2015	Προσθήκη διαγράμματος καταστάσεων	0.1.4
Uni-Food	06/04/2016	Προσθήκη πακέτων στατικής μοντελοποίησης	0.2
Uni-Food	07/04/2016	Προσθήκη διαγραμμάτων κλάσεων	0.2.1
Uni-Food	09/04/2016	Προσθήκη δυναμικής μοντελοποίησης	0.3
Uni-Food	11/04/2016	Αλλαγή κλάσης UniFoodDBProxy και διορθώσεις πακέτων στατικής μοντελοποίησης	0.5
Uni-Food	11/04/2016	Διορθώσεις διαγραμμάτων κλάσεων και δυνα- μική μοντελοποίησης	0.5.1
Uni-Food	15/04/2016	Διορθώσεις στα διαγράμματα καταστάσεων	0.6
Uni-Food	21/05/2016	Διορθώσεις στο Entity Comment και στο διά- γραμμα κλάσεων πακέτου CommentsAdmin- istration	1.0.1
Uni-Food	01/06/2016	Διορθώσεις στον τρόπο εμφάνισης των πακέτων	1.0.2
Uni-Food	08/06/2016	Διόρθωση Τυπογραφικών Παραδοχών και Ανα- γνωστικού Κοινού	1.0.3
Uni-Food	08/06/2016	Προσθήκη συνάρτησης στο ResultsGUI	1.0.4
Uni-Food	08/06/2016	Διόρθωση Σεναρίου Ψηφοφορίας Χρήστη και Σε- ναρίου Διαχείρισης Προγράμματος σίτισης	1.0.5
Uni-Food	08/06/2016	Διόρθωση κατανομής ΜΛΑ	1.0.6



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2016

Μέλη της Ομάδας Ανάπτυξης

Όνομα	OA	Email
Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Χ. Ζολώτας	*	christopherzoltas@issel.ee.auth.gr
Μαυροδής Κωνσταντίνος	Uni-Food	kmavrodis@outlook.com
Νήρας Δημήτρης	Uni-Food	niras.94@gmail.com
Σχινάς Γιώργος	Uni-Food	schinas.georgios@outlook.com
Χατζηθωμά Ανδρέας	Uni-Food	andreas_h92@hotmail.com



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	4
Λίστα Σχημάτων	5
1 Εισαγωγικά	6
1.1 Στόχος του Εγγράφου	6
1.2 Τυπογραφικές Παραδοχές του Εγγράφου	6
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 VotingCapabilities	8
2.1.2 Πακέτο ScheduleCapabilities	19
2.2 Πακέτο λεξιλογίου σεναρίων μέσης προτεραιότητας	26
2.2.2 Πακέτο TimeEstimation	26
2.3 Πακέτο λεξιλογίου σεναρίων χαμηλής προτεραιότητας	31
2.3.1 Πακέτο CommentsAdministration	31
3 Απαιτήσεις Λογισμικού	36
3.1 Δυναμική μοντελοποίηση του συστήματος	36
3.1.1 Πακέτο ψηφοφορίας χρήστη	36
3.1.2 Πακέτο διαχείρισης προγράμματος σίτισης από τον διαχειριστή	38
3.1.3 Πακέτο διαχείρισης προγράμματος σίτισης από τον διαχειριστή	40
3.2 Διαγράμματα Καταστάσεων	41
3.2.1 Διάγραμμα Καταστάσεων χρήστη	41
3.2.2 Διάγραμμα Καταστάσεων διαχειριστή	41
3.3 Μη λειτουργικές απαιτήσεις	42
3.3.1 Απαιτήσεις Φορητότητας (Portability)	42
3.3.2 Τεχνικές Απαιτήσεις Περιβάλλοντος	42
3.3.3 Απαιτήσεις πολιτικής (Policy - legal, corporate)	42
Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας	43
Παράρτημα ΙΙ – Ανοιχτά Θέματα	44



Λίστα Σχημάτων

Σχήμα 1 – Διάγραμμα κλάσεων πακέτου VotingCababilities	18
Σχήμα 2 – Διάγραμμα κλάσεων πακέτου ScheduleCababilities	25
Σχήμα 3 – Διάγραμμα κλάσεων πακέτου TimeEstimation	30
Σχήμα 4 – Διάγραμμα κλάσεων πακέτου CommentsAdministration	35
Σχήμα 5 – Σενάριο ψηφοφορίας χρήστη	37
Σχήμα 6 – Σενάριο διαχείρισης του προγράμματος σίτισης από τον διαχειριστή	39
Σχήμα 7 – Σενάριο διαχείρισης του προγράμματος σίτισης από τον διαχειριστή	40
Σχήμα 8 – Διάγραμμα καταστάσεων χρήστη	41
Σχήμα 9 – Διάγραμμα καταστάσεων διαχειριστή	41



1 Εισαγωγικά

1.1 Στόχος του Εγγράφου

Έχοντας περιγράψει με ακρίβεια τις απαιτήσεις των χρηστών στο «Έγγραφο Απαιτήσεων Χρηστών», το επόμενο βήμα είναι η λεπτομερής περιγραφή των απαιτήσεων του συστήματος. Το παρόν έγγραφο αποτελεί το «Έγγραφο Απαιτήσεων Συστήματος» και στόχος του είναι να μοντελοποιήσει τις απαιτήσεις των χρηστών με τρόπο σαφή και ακριβή, ώστε να αποτελέσει οδηγό για τους ανθρώπους που θα αναλάβουν την υλοποίηση του συστήματος.

Η ανάλυση λοιπόν που θα πραγματοποιήσουμε θα επικεντρωθεί στον διαχωρισμό του συστήματος σε διακριτά μέρη. Αυτά είναι οι *κλάσεις οντοτήτων (entities)* που αναπαριστούν τη μόνιμη πληροφορία η οποία ανιχνεύεται από το σύστημα, οι *κλάσεις ελέγχου (controllers)* που ελέγχουν και συντονίζουν τη συμπεριφορά του συστήματος και οι *οριακές κλάσεις (boundaries)* που μοντελοποιούν την αλληλεπίδραση ανάμεσα στο εσωτερικό και το εξωτερικό του συστήματος. Ο σαφής διαχωρισμός των παραπάνω κλάσεων και ο ακριβής καθορισμός των συστατικών τους στοιχείων (χαρακτηριστικών και μεθόδων) βοηθά στην ορθή ανάπτυξη του συστήματος αλλά και στην ομαλή λειτουργία και συντήρησή του.

Τέλος σημαντικό κομμάτι της ανάλυσης αποτελεί η λεπτομερής δυναμική μοντελοποίηση του συστήματος. Αυτή επιτυγχάνεται με την παρουσίαση των ροών οι οποίες περιγράφουν το πως οι κλάσεις που αναπτύχθηκαν κατά τη διαδικασία της στατικής μοντελοποίησης ικανοποιούν τα σενάρια χρήσης που συγκροτήθηκαν στο «Έγγραφο Απαιτήσεων Χρηστών». Δημιουργούνται δηλαδή διαγράμματα ακολουθιών που περιγράφουν πλήρως τη συμπεριφορά του συστήματος κατά την εκτέλεση των διαφόρων λειτουργιών του και κατά την αλληλεπίδρασή του με τους χρήστες και τα εξωτερικά συστήματα.

1.2 Τυπογραφικές Παραδοχές του Εγγράφου

Το κείμενο του εγγράφου αυτού είναι γραμμένο με γραμματοσειρά Calibri, μεγέθους 11pt. Ο τίτλος κάθε κεφαλαίου είναι γραμμένος σε μέγεθος 18pt με έντονη γραφή και οι υπότιτλοι σε μέγεθος 14 με έντονη γραφή. Οι λέξεις – φράσεις που σημειώνονται με πλάγια γραφή αποτελούν σημαντικά σημεία στα οποία ο αναγνώστης πρέπει να δώσει τη δέουσα προσοχή. Επίσης, οι σημειώσεις στο υποσέλιδο χρησιμοποιούνται για να προσδιοριστεί καλύτερα κάποιος όρος του κειμένου. Σε περίπτωση που κριθεί αναγκαία η αναφορά κάποιου αγγλικού όρου, αυτός αναγράφεται μετά από τον ελληνικό μέσα σε παρενθέσεις. Τέλος για τις λεζάντες των διαγραμμάτων χρησιμοποιείται γραμματοσειρά Calibri, μεγέθους 11pt και πλάγια γραφή.

Στα διαγράμματα κλάσεων του δευτέρου κεφαλαίου ακολουθούνται οι εξής προδιαγραφές. Όταν οι σχέσεις ανάμεσα στις κλάσεις είναι 1 προς 1 τότε δεν εμφανίζονται οι αριθμοί πάνω από τα βέλη. Σε αντίθετη περίπτωση εμφανίζεται στα άκρα τους η μορφή της συσχέτισης που υπάρχει.

1.3 Αναγνωστικό κοινό και τρόπος ανάγνωσης

Το αναγνωστικό κοινό του εγγράφου απαιτήσεων λογισμικού αποτελείται από:

- Τους αρμόδιους μηχανικούς του Κέντρου Ηλεκτρονικής Διακυβέρνησης (ΚΗΔ), ώστε να κρίνουν την εγκυρότητα του έργου και τον αποτελεσματικό τρόπο αντιμετώπισης των προβλημάτων του σύγχρονου πανεπιστημίου. Παράλληλα το ΚΗΔ είναι υπεύθυνο για την παροχή πρόσβασης στα στοιχεία φοιτητών, συνεπώς είναι απαραίτητη η συνεργασία του με τους μηχανικούς ανάπτυξης για την επιτυχή και αποδοτική σύνδεση με τις βάσεις δεδομένων του πανεπιστημίου.



- Οι μηχανικοί λογισμικού που θα αναλάβουν την υλοποίηση του συστήματος.
- Οι μηχανικοί λογισμικού που θα αναλάβουν την συντήρηση και εύρυθμη λειτουργία του συστήματος.

Η ανάγνωση του εγγράφου πρέπει να πραγματοποιείται ακολουθώντας την ορθή σειρά των κεφαλαίων, ώστε να γίνεται πλήρως κατανοητή η λειτουργικότητα του συστήματος. Παράλληλα με το κείμενο υπάρχει μια πληθώρα πινάκων και διαγραμμάτων που συμβάλουν στην πληρέστερη αντίληψη των διάφορων λειτουργιών του συστήματος αλλά και των αλληλεπιδράσεων που υπάρχουν μεταξύ τους.

1.4 Σκοπός του Έργου

Είναι κατά γενική ομολογία αποδεκτό ότι οι φοιτητές του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης δεν είναι αρκετά ευχαριστημένοι με τις υπηρεσίες που προσφέρει η Φοιτητική Λέσχη σήμερα. Μερικοί από τους βασικότερους παράγοντες που επηρεάζουν προς αυτήν την κατεύθυνση περιγράφονται παρακάτω.

Οι φοιτητές δεν είναι ευχαριστημένοι με τα γεύματα που προσφέρονται από την λέσχη, καθώς δεν έχουν καμία δικαιοδοσία στο να καθορίσουν ή έστω να επηρεάσουν την ποικιλομορφία του φαγητού που παρέχεται. Η διαδικασία επιλογής φαγητού, λοιπόν, μέχρι σήμερα κάθε άλλο παρά δημοκρατική είναι, μιας και ο διαχειριστής της λέσχης είναι ο μόνος υπεύθυνος για αυτήν. Παράλληλα, ο τελευταίος δεν έχει τη δυνατότητα να πάρει feedback από τους φοιτητές για το τί τους αρέσει και τί όχι, με αποτέλεσμα ακόμα και αν θέλει να λαμβάνει υπόψιν την γνώμη τους να μην μπορεί να το κάνει.

Επιπροσθέτως, ένα ακόμη πρόβλημα που αντιμετωπίζουν οι φοιτητές είναι οι εκτεταμένοι χρόνοι α-ναμονής στις ουρές των διανομών. Μοιάζει σχεδόν θέμα τύχης το πότε ο φοιτητής θα πρέπει να περιμένει 1 λεπτό και πότε θα πρέπει να περιμένει πάνω από 30 λεπτά για μία μερίδα φαγητού. Μοιάζει θέμα τύχης μιας και εξαρτάται από ένα σύνολο παραγόντων δύσκολων ως προς τη μέτρηση ή και την πρόβλεψη.

Τέλος, οι διαχειριστές της λέσχης δεν έχουν στη διάθεσή τους ένα οργανωμένο σύστημα διαχείρισης του εβδομαδιαίου προγράμματος με αποτέλεσμα η ορθή οργάνωση του να είναι πρακτικά αδύνατη λόγω έλλειψης κατάλληλων εργαλείων.

Έχουμε, λοιπόν, ένα σύνολο από προβλήματα που αφορούν φοιτητές και διαχειριστές της λέσχης και επηρεάζουν αρνητικά την ορθή λειτουργία της λέσχης, όπως και την ψυχολογία και την στάση του πανεπιστημιακού κόσμου απέναντι της.

Το Uni-Food είναι μια επαναστατική, για τα δεδομένα του πανεπιστημίου, πρωτοβουλία που έρχεται να δώσει λύση σε αυτά τα προβλήματα με τον βέλτιστο δυνατό τρόπο. Στοχεύουμε τόσο στην εκδημοκράτιση της απολαβής ενός από τα σημαντικότερα αγαθά, του φαγητού, στους χώρους του πανεπιστημίου, όσο και στην πρακτική βελτίωση της εμπειρίας χρήσης της λέσχης, ενώ δεν ξεχνάμε και την διευκόλυνση της λειτουργίας της για τους διαχειριστές της. Απώτερος στόχος είναι η δημιουργία άριστου κλίματος και ονόματος για την λέσχη του ΑΠΘ, έτσι ώστε να γίνει πρότυπο λέσχης για εκπαιδευτικά ιδρύματα τόσο του εσωτερικού όσο και του εξωτερικού, αποτελώντας δυνατό όπλο στη φάρετρα του ΑΠΘ, αλλά και στην βελτίωση της σχέσης των φοιτητών με το ίδιο το πανεπιστήμιο και την πανεπιστημιακή κοινότητα.

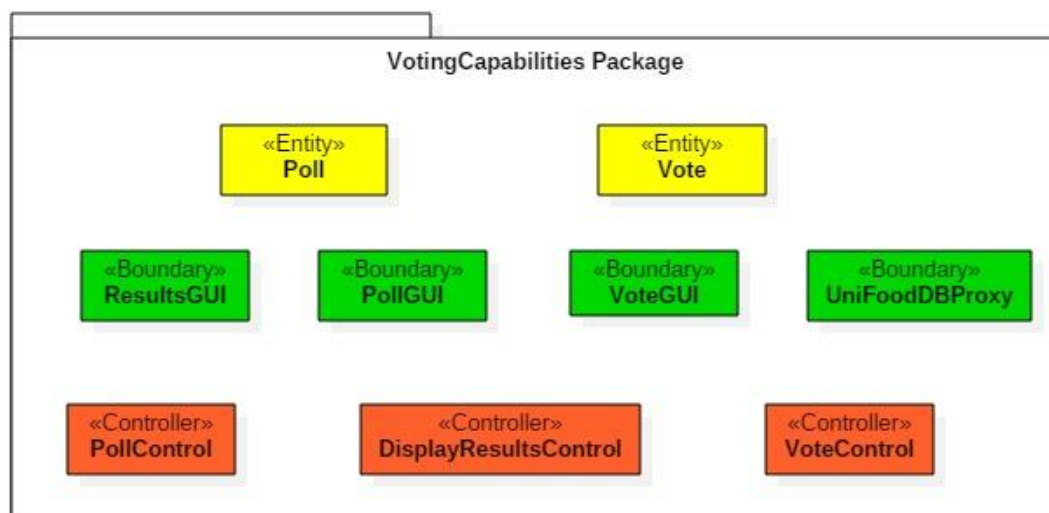


2 Στατική Μοντελοποίηση

2.1 Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας

2.1.1 Πακέτο VotingCapabilities

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που απαιτούνται για την καταχώρηση μιας ψηφοφορίας από τον διαχειριστή και την εμφάνιση του ιστορικού των ψηφοφοριών. Επίσης περιλαμβάνει την κλάση που απαιτείται για την καταχώρηση μιας ψήφου από τον χρήστη. Επιπλέον περιλαμβάνονται και οι κλάσεις με τα αντίστοιχα γραφικά περιβάλλοντα της εφαρμογής.





Entity Poll

«Entity» Poll
-firstChoice: String -secondChoice: String -thirdChoice: String -fourthChoice: String -fifthChoice: String -sixthChoice: String -date: String
+setFirstChoice(firstChoice: String): void +setSecondChoice(secondChoice: String): void +setThirdChoice(thirdChoice: String): void +setFourthChoice(fourthChoice: String): void +setFifthChoice(fifthChoice: String): void +setSixthChoice(sixthChoice: String): void +getFirstChoice(): String +getSecondChoice(): String +getThirdChoice(): String +getFourthChoice(): String +getFifthChoice(): String +getSixthChoice(): String +Poll(choice1: String, choice2: String, choice3: String, choice4: String, choice5: String, choice6: String): void +setDate(date: String): void +getDate(): String

Η κλάση Poll περιέχει όλα τα στοιχεία τα οποία χρειάζονται για την υποβολή μίας καινούριας ψηφοφορίας.

Οι μεταβλητές της κλάσης αυτής είναι οι εξής:

- **firstChoice**: Μια μεταβλητή τύπου String που περιέχει την 1^η επιλογή φαγητού του διαχειριστή.
- **secondChoice**: Μια μεταβλητή τύπου String που περιέχει την 2^η επιλογή φαγητού του διαχειριστή.
- **thirdChoice**: Μια μεταβλητή τύπου String που περιέχει την 3^η επιλογή φαγητού του διαχειριστή.
- **fourthChoice**: Μια μεταβλητή τύπου String που περιέχει την 4^η επιλογή φαγητού του διαχειριστή.
- **fifthChoice**: Μια μεταβλητή τύπου String που περιέχει την 5^η επιλογή φαγητού του διαχειριστή.
- **sixthChoice**: Μια μεταβλητή τύπου String που περιέχει την 6^η επιλογή φαγητού του διαχειριστή.
- **date**: Μια μεταβλητή τύπου String η οποία περιέχει την ημερομηνία στην οποία αναφέρεται η συγκεκριμένη ψηφοφορία.

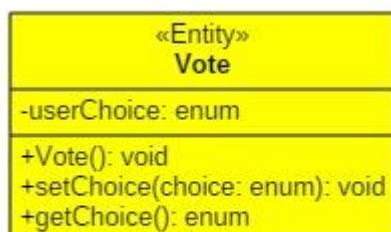
Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **setFirstChoice(firstChoice)**: Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή firstChoice.
- **setSecondChoice(secondChoice)**: Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή secondChoice.



- **setThirdChoice(thirdChoice):** Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή thirdChoice.
- **setFourthChoice(fourthChoice):** Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή fourthChoice.
- **setFifthChoice(fifthChoice):** Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή fifthChoice.
- **setSixthChoice(sixthChoice):** Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή sixthChoice.
- **getFirstChoice():** Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής firstChoice.
- **getSecondChoice():** Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής secondChoice.
- **getThirdChoice():** Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής thirdChoice.
- **getFourthChoice():** Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής fourthChoice.
- **getFifthChoice():** Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής fifthChoice.
- **getSixthChoice():** Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής sixthChoice.
- **Poll(choice1, choice2, choice3, choice4, choice5, choice6):** Η μέθοδος αυτή αποτελεί τον Constructor της κλάσης, στην οποία μέσω των ορισμάτων της η κλάση λαμβάνει όλα τα απαραίτητα στοιχεία για την δημιουργία ψηφοφορίας.
- **setDate(date):** Είναι η μέθοδος μέσω της οποίας μπορούμε να καθορίσουμε την ημερομηνία για την οποία έχουμε θέσει την συγκεκριμένη ψηφοφορία.
- **getDate():** Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής date.

Entity Vote



Η κλάση Vote περιέχει τις μεταβλητές και μεθόδους που είναι απαραίτητες για τον ορισμό μίας ψηφου ενός χρήστη.



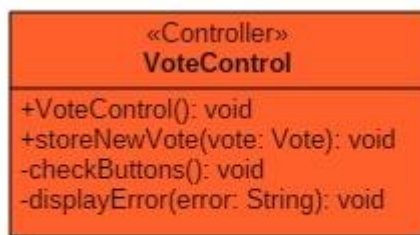
Οι μεταβλητές της κλάσης αυτής είναι οι εξής:

- **userChoice**: Μια μεταβλητή τύπου `enum` η οποία περιέχει την επιλογή του χρήστη από τις διαθέσιμες επιλογές του μενού.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **Vote()**: Η μέθοδος αυτή αποτελεί τον Constructor της κλάσης μέσω του οποίου ορίζεται μία καινούρια ψήφος ενός χρήστη.
- **setChoice(choice)**: Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή `userChoice`.
- **getChoice()**: Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής `userChoice`.

Controller VoteControl



Η κλάση αυτή ελέγχει τη διαδικασία καταχώρησης νέας ψήφου από τον χρήστη. Εμφανίζει το γραφικό περιβάλλον στο οποίο ο χρήστης επιλέγει το φαγητό της προτίμησης του και ελέγχει την εγκυρότητα της επιλογής του. Στη συνέχεια συνδέεται με τη Βάση Δεδομένων και καταχωρεί την επιλογή του χρήστη ως ένα αντικείμενο τύπου `Vote` ή εμφανίζει το κατάλληλο μήνυμα σφάλματος.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **VoteControl()**: Η μέθοδος της κλάσης αυτής αποτελεί τον Constructor της κλάσης μέσω του οποίου δημιουργείται ένα καινούριο αντικείμενο.
- **storeNewVote(vote)**: Είναι η μέθοδος μέσω της οποίας καταχωρούμε την ψήφο του χρήστη στη Βάση Δεδομένων.
- **checkButtons()**: Είναι η μέθοδος μέσω της οποίας ελέγχεται αν έχει πατηθεί κάποιο από τα `RadioButtons` από τον χρήστη.
- **displayError(error)**: Είναι η μέθοδος μέσω της οποίας εμφανίζεται η οθόνη σφάλματος σε περίπτωση που δεν έχει πατηθεί κανένα από τα `RadioButtons`.



Controller PollControl

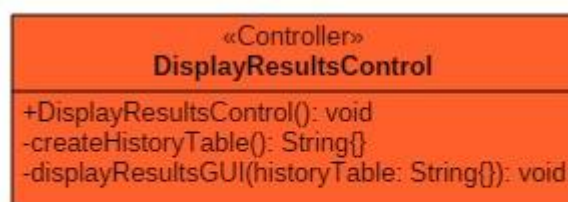


Η κλάση αυτή ελέγχει τη διαδικασία καταχώρησης νέας ψηφοφορίας από τον διαχειριστή. Εμφανίζει το γραφικό περιβάλλον στο οποίο ο διαχειριστής πληκτρολογεί το προτεινόμενο μενού και ελέγχει την εγκυρότητα των στοιχείων. Στη συνέχεια συνδέεται με τη Βάση Δεδομένων και καταχωρεί την ψηφοφορία ως ένα αντικείμενο τύπου Poll ή εμφανίζει το κατάλληλο μήνυμα σφάλματος.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **PollControl():** Η μέθοδος της κλάσης αυτής αποτελεί τον Constructor της κλάσης μέσω του οποίου δημιουργείται ένα καινούριο αντικείμενο.
- **storeNewPoll(poll):** Είναι η μέθοδος μέσω της οποίας καταχωρούμε την ψηφοφορία του διαχειριστή στη Βάση Δεδομένων.
- **checkEnteredData():** Είναι η μέθοδος μέσω της οποίας ελέγχονται τα εισαχθέντα στοιχεία για τυχόν σφάλματα.
- **displayError(error):** Είναι η μέθοδος μέσω της οποίας εμφανίζεται η οθόνη σφάλματος σε περίπτωση που δεν έχει συμπληρωθεί κάποιο πεδίο ή έχουν συμπληρωθεί με το ίδιο φαγητό.

Controller DisplayResultsControl



Η κλάση αυτή ελέγχει τη διαδικασία εμφάνισης του ιστορικού των ψηφοφοριών. Είναι υπεύθυνη για τη δημιουργία ενός πίνακα καταχώρησης των ψηφοφοριών και για την εμφάνιση τους στο κατάλληλο γραφικό περιβάλλον.

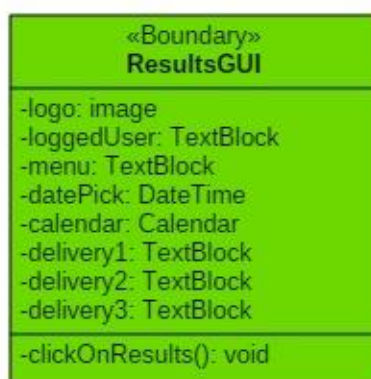
Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **DisplayResultsControl():** Η μέθοδος της κλάσης αυτής αποτελεί τον Constructor της κλάσης μέσω του οποίου δημιουργείται ένα καινούριο αντικείμενο.



- **createHistoryTable():** Είναι η μέθοδος μέσω της οποίας δημιουργείται ο πίνακας του ιστορικού που περιέχει όλες τις παλαιότερες ψηφοφορίες, καθώς και τα αποτελέσματα αυτών. Τα στοιχεία αυτά φορτώνονται από τη Βάση Δεδομένων μέσω της κλάσης VotingDBProxy.
- **displayResultsGUI(historyTable):** Είναι η μέθοδος μέσω της οποίας δημιουργείται το γραφικό περιβάλλον απεικόνισης του ιστορικού έχοντας ως όρισμα τον πίνακα που δημιουργήθηκε μέσω της συνάρτησης createHistoryTable().

Boundary ResultsGUI



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος του ιστορικού των ψηφοφοριών.

Οι μεταβλητές της κλάσης αυτής είναι οι εξής:

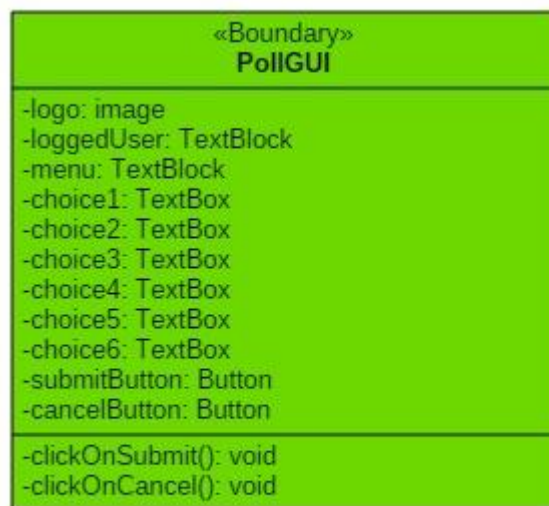
- **logo:** Το λογότυπο της εφαρμογής.
- **loggedUser:** Ο τίτλος που απεικονίζει την ιδιότητα του συνδεδεμένου χρήστη.
- **menu:** Ο τίτλος που απεικονίζει το μενού στο οποίο βρισκόμαστε.
- **datePick:** Η μεταβλητή στην οποία ορίζουμε την ημερομηνία της οποίας θέλουμε να δούμε την ψηφοφορία.
- **calendar:** Ένα ημερολόγιο στο οποίο μπορούμε να επιλέξουμε την ημερομηνία της οποίας θέλουμε να δούμε την ψηφοφορία.
- **delivery1:** Ο τίτλος που απεικονίζει τη διανομή 1.
- **delivery2:** Ο τίτλος που απεικονίζει τη διανομή 2.
- **delivery3:** Ο τίτλος που απεικονίζει τη διανομή 3.

Η μέθοδος της κλάσης αυτής είναι η εξής:

- **clickOnSubmit():** Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί resultsButton, καλείται ο controller DisplayResultsControl ο οποίος είναι υπεύθυνος για την εμφάνιση του ιστορικού των ψηφοφοριών.



Boundary PollGUI



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος στο οποίο ο διαχειριστής προσθέτει μία καινούρια ψηφοφορία.

Οι μεταβλητές της κλάσης αυτής είναι οι εξής:

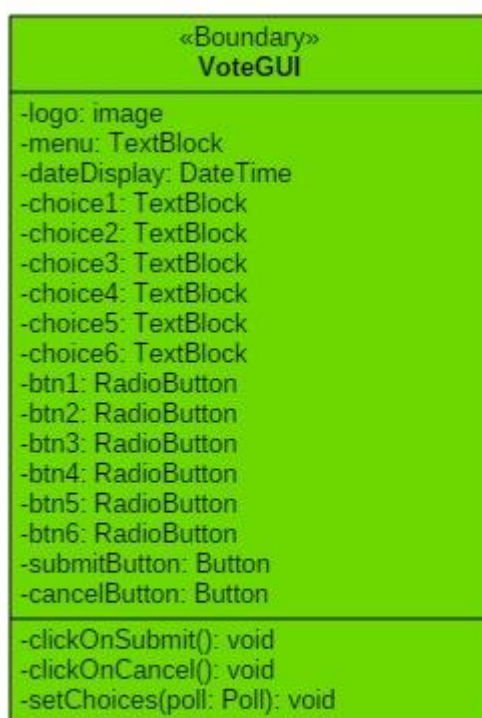
- **logo**: Το λογότυπο της εφαρμογής.
- **loggedUser**: Ο τίτλος που απεικονίζει την ιδιότητα του συνδεδεμένου χρήστη.
- **menu**: Ο τίτλος που απεικονίζει το μενού στο οποίο βρισκόμαστε.
- **choice1**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την προτεινόμενη επιλογή.
- **choice2**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την προτεινόμενη επιλογή.
- **choice3**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την προτεινόμενη επιλογή.
- **choice4**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την προτεινόμενη επιλογή.
- **choice5**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την προτεινόμενη επιλογή.
- **choice6**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την προτεινόμενη επιλογή.
- **submitButton**: Μία μεταβλητή τύπου Button το οποίο όταν πατηθεί αποστέλλει την ψηφοφορία στη Βάση Δεδομένων ή εμφανίζει μήνυμα σφάλματος εάν ο έλεγχος των στοιχείων αποτύχει.
- **cancelButton**: Μία μεταβλητή τύπου Button το οποίο όταν πατηθεί το σύστημα επιστρέφει στην αρχική σελίδα της εφαρμογής.



Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **clickOnSubmit():** Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί submitButton καλείται ο Controller PollControl για τον έλεγχο των πεδίων και την αποστολή της ψηφοφορίας στη Βάση Δεδομένων.
- **clickOnCancel():** Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί cancelButton το σύστημα επιστρέφει στην αρχική σελίδα της εφαρμογής.

Boundary VoteGUI



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος στο οποίο ο χρήστης επιλέγει το φαγητό της προτίμησης του.

Οι μεταβλητές της κλάσης αυτής είναι οι εξής:

- **logo:** Το λογότυπο της εφαρμογής.
- **menu:** Ο τίτλος που απεικονίζει το μενού στο οποίο βρισκόμαστε.
- **choice1:** Η μεταβλητή στην οποία περιέχεται η πρώτη επιλογή του διαχειριστή.
- **choice2:** Η μεταβλητή στην οποία περιέχεται η δεύτερη επιλογή του διαχειριστή.
- **choice3:** Η μεταβλητή στην οποία περιέχεται η τρίτη επιλογή του διαχειριστή.
- **choice4:** Η μεταβλητή στην οποία περιέχεται η τέταρτη επιλογή του διαχειριστή.
- **choice5:** Η μεταβλητή στην οποία περιέχεται η πέμπτη επιλογή του διαχειριστή.
- **choice6:** Η μεταβλητή στην οποία περιέχεται η έκτη επιλογή του διαχειριστή.

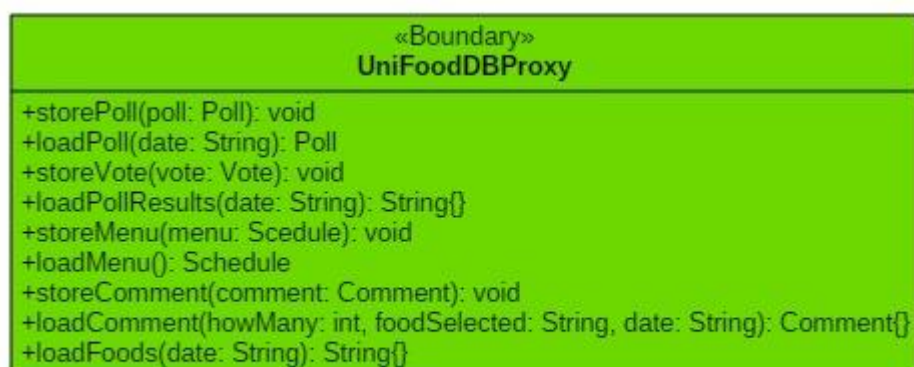


- **btn1**: Μία μεταβλητή τύπου RadioButton η οποία αντιπροσωπεύει την επιλογή του χρήστη για την πρώτη επιλογή φαγητού.
- **btn2**: Μία μεταβλητή τύπου RadioButton η οποία αντιπροσωπεύει την επιλογή του χρήστη για τη δεύτερη επιλογή φαγητού.
- **btn3**: Μία μεταβλητή τύπου RadioButton η οποία αντιπροσωπεύει την επιλογή του χρήστη για την τρίτη επιλογή φαγητού.
- **btn4**: Μία μεταβλητή τύπου RadioButton η οποία αντιπροσωπεύει την επιλογή του χρήστη για την τέταρτη επιλογή φαγητού.
- **btn5**: Μία μεταβλητή τύπου RadioButton η οποία αντιπροσωπεύει την επιλογή του χρήστη για την πέμπτη επιλογή φαγητού.
- **btn6**: Μία μεταβλητή τύπου RadioButton η οποία αντιπροσωπεύει την επιλογή του χρήστη για την έκτη επιλογή φαγητού.
- **submitButton**: Μία μεταβλητή τύπου Button το οποίο όταν πατηθεί αποστέλλει την ψηφο του χρήστη στη Βάση Δεδομένων ή εμφανίζει μήνυμα σφάλματος εάν ο έλεγχος των στοιχείων αποτύχει.
- **cancelButton**: Μία μεταβλητή τύπου Button το οποίο όταν πατηθεί το σύστημα επιστρέφει στην αρχική σελίδα της εφαρμογής.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **clickOnSubmit()**: Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί submitButton καλείται ο Controller VoteControl για τον έλεγχο των πεδίων και την αποστολή της ψηφου στη Βάση Δεδομένων.
- **clickOnCancel()**: Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί cancelButton το σύστημα επιστρέφει στην αρχική σελίδα της εφαρμογής.
- **setChoices(poll)**: Είναι η μέθοδος μέσω της οποίας παίρνουν τιμές οι αντίστοιχες choice1 - choice6 μεταβλητές σύμφωνα με της επιλογές που έχει δώσει ο διαχειριστής, οι οποίες βρίσκονται στο αντικείμενο poll που έχει ως όρισμα η μέθοδος και το οποίο φορτώνεται από τη Βάση Δεδομένων.

Boundary UniFoodDBProxy





Η κλάση αυτή είναι υπεύθυνη για την επικοινωνία με τη Βάση Δεδομένων δίνοντας τη δυνατότητα αποθήκευσης ψηφοφοριών και ψήφων, φόρτωσης συγκεκριμένων αποτελεσμάτων και ψηφοφοριών, αποθήκευσης και φόρτωσης των σχολίων των χρηστών και αποθήκευσης και φόρτωσης του προγράμματος σίτισης.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **storePoll(poll):** Είναι η μέθοδος μέσω της οποίας αποθηκεύεται μια ψηφοφορία, ένα αντικείμενο δηλαδή τύπου Poll στη Βάση Δεδομένων.
- **loadPoll(date):** Είναι η μέθοδος μέσω της οποίας φορτώνεται από τη Βάση Δεδομένων μία συγκεκριμένη ψηφοφορία, ένα αντικείμενο τύπου Poll δηλαδή, βάσει συγκεκριμένης ημερομηνίας η οποία δίνεται ως όρισμα.
- **storeVote(vote):** Είναι η μέθοδος μέσω της οποίας αποθηκεύεται μία ψήφος ενός χρήστη, ένα αντικείμενο τύπου Vote δηλαδή στη Βάση Δεδομένων.
- **loadPollResults(date):** Είναι η μέθοδος μέσω της οποίας φορτώνονται από τη Βάση Δεδομένων τα αποτελέσματα μίας ψηφοφορίας βάσει συγκεκριμένης ημερομηνίας η οποία δίνεται ως όρισμα.
- **storeMenu(menu):** Είναι η μέθοδος μέσω της οποίας αποθηκεύεται το πρόγραμμα σίτισης υπό τη μορφή ενός Schedule αντικειμένου, στη Βάση Δεδομένων.
- **loadMenu():** Είναι η μέθοδος μέσω της οποίας φορτώνεται από τη Βάση Δεδομένων το πρόγραμμα σίτισης.
- **storeComment(comment):** Είναι η μέθοδος μέσω της οποίας αποθηκεύεται ένα σχόλιο στη Βάση Δεδομένων.
- **loadComments(howMany, foodSelected, date):** Είναι η μέθοδος μέσω της οποίας επιστρέφονται τα πιο πρόσφατα σχόλια για το φαγητό που δόθηκε ως όρισμα.
- **loadFoods(date):** Είναι η μέθοδος μέσω της οποίας επιστρέφονται τα φαγητά για τα οποία υπάρχουν σχόλια για μία συγκεκριμένη ημερομηνία, η οποία δίνεται ως όρισμα.

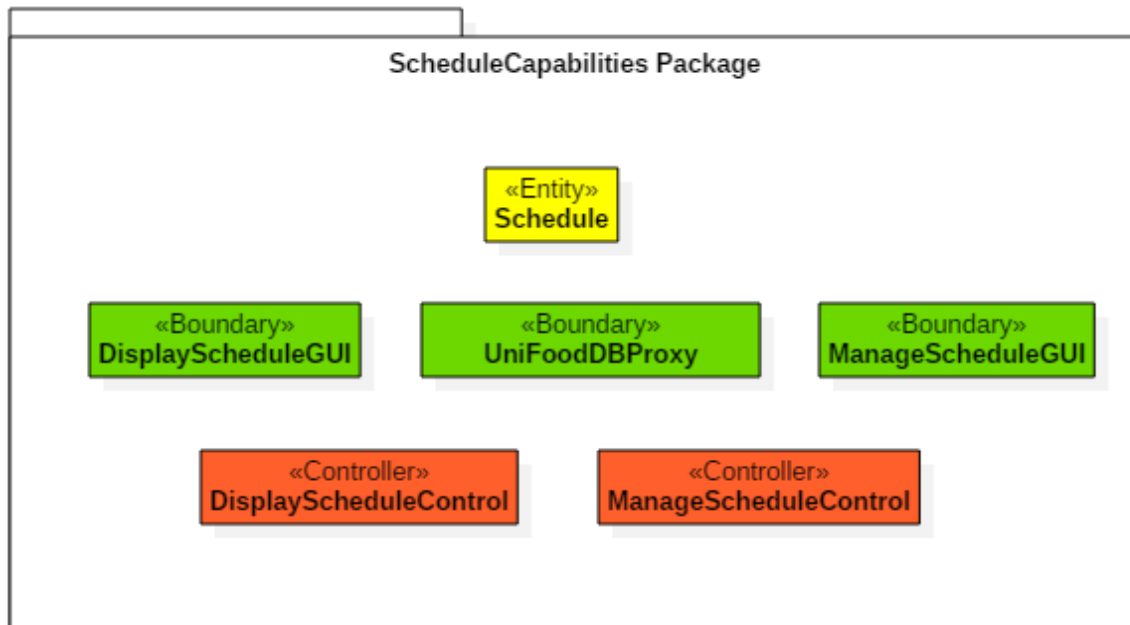


Σχήμα 1 – Διάγραμμα κλάσεων πακέτου VotingCababilities

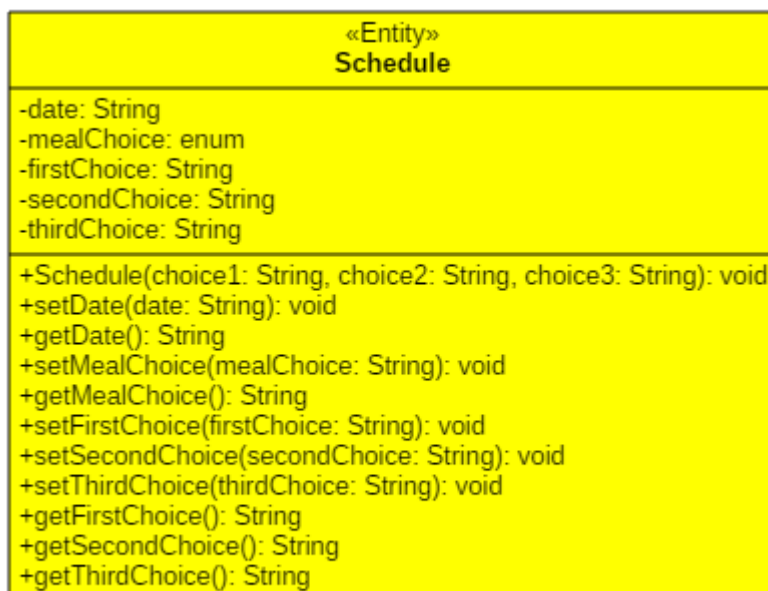


2.1.2 Πακέτο ScheduleCapabilities

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που απαιτούνται για την καταχώρηση και επεξεργασία του προγράμματος σίτισης από τον διαχειριστή καθώς και την εμφάνιση του εβδομαδιαίου προγράμματος σίτισης.



Entity Schedule



Η κλάση Schedule περιέχει όλα τα στοιχεία τα οποία χρειάζονται για την καταχώρηση και τροποποίηση του προγράμματος σίτισης.

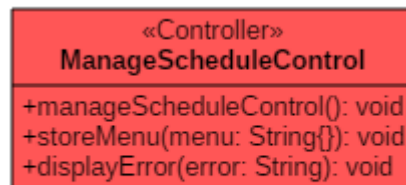


Οι μεταβλητές της κλάσης αυτής είναι οι εξής:

- **date**: Μια μεταβλητή τύπου String η οποία περιέχει την ημερομηνία στην οποία αναφέρεται η συγκεκριμένη καταχώρηση/τροποποίηση του προγράμματος σίτισης.
- **mealChoice**: Μια μεταβλητή τύπου enum η οποία περιέχει την επιλογή του διαχειριστή σχετικά με το γεύμα (πρωινό, μεσημεριανό, βραδινό).
- **firstChoice**: Μια μεταβλητή τύπου String η οποία περιέχει την 1^η διαθέσιμη επιλογή φαγητού που τοποθετήθηκε από το διαχειριστή.
- **secondChoice**: Μια μεταβλητή τύπου String η οποία περιέχει την 2^η διαθέσιμη επιλογή φαγητού που τοποθετήθηκε από το διαχειριστή.
- **thirdChoice**: Μια μεταβλητή τύπου String η οποία περιέχει την 3^η διαθέσιμη επιλογή φαγητού που τοποθετήθηκε από το διαχειριστή.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **Schedule(meal, choice1, choice2, choice3)**: Η μέθοδος αυτή αποτελεί τον Constructor της κλάσης μέσω του οποίου ορίζεται μία καινούρια καταχώρηση προγράμματος σίτισης για ένα συγκεκριμένο γεύμα (πρωινό, μεσημεριανό, βραδινό).
- **setDate(date)**: Είναι η μέθοδος μέσω της οποίας μπορούμε να καθορίσουμε την ημερομηνία για την οποία έχουμε θέσει την συγκεκριμένη τροποποίηση του προγράμματος σίτισης.
- **getDate()**: Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής date.
- **setMealChoice(mealChoice)**: Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή mealChoice.
- **getMealChoice()**: Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής mealChoice.
- **setFirstChoice(firstChoice)**: Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή firstChoice.
- **setSecondChoice(secondChoice)**: Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή secondChoice.
- **setThirdChoice(thirdChoice)**: Είναι η μέθοδος μέσω της οποίας μπορούμε να δώσουμε τιμή στη μεταβλητή thirdChoice.
- **getFirstChoice()**: Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής firstChoice.
- **getSecondChoice()**: Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής secondChoice.
- **getThirdChoice()**: Είναι η μέθοδος μέσω της οποίας μπορούμε να πάρουμε το περιεχόμενο της μεταβλητής thirdChoice.

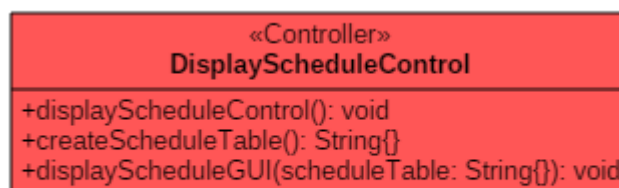


Η κλάση αυτή ελέγχει τη διαδικασία καταχώρησης και τροποποίησης του προγράμματος σίτισης από τον διαχειριστή. Εμφανίζει το γραφικό περιβάλλον στο οποίο ο διαχειριστής τροποποιεί ή καταχωρεί το πρόγραμμα σίτισης και ελέγχει την εγκυρότητα των αλλαγών του. Στη συνέχεια συνδέεται με τη Βάση Δεδομένων και καταχωρεί το πρόγραμμα σίτισης ως ένα αντικείμενο τύπου Schedule ή εμφανίζει το κατάλληλο μήνυμα σφάλματος.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **manageScheduleControl(date):** Η μέθοδος της κλάσης αυτής αποτελεί τον Constructor της κλάσης μέσω του οποίου δημιουργείται ένα καινούριο αντικείμενο.
- **storeMenu(menu):** Είναι η μέθοδος μέσω της οποίας καταχωρούμε το πρόγραμμα σίτισης στη Βάση Δεδομένων.
- **checkEnteredData():** Είναι η μέθοδος μέσω της οποίας η πληρότητα των εισαχθέντων στοιχείων.
- **displayError(error):** Είναι η μέθοδος μέσω της οποίας εμφανίζεται η οθόνη σφάλματος σε περίπτωση που έχει πατηθεί «Υποβολή» και όλα τα πεδία TextBox είναι κενά.

Controller DisplayScheduleControl



Η κλάση αυτή ελέγχει τη διαδικασία εμφάνισης του προγράμματος σίτισης. Είναι υπεύθυνη για τη δημιουργία ενός πίνακα με τα διαθέσιμα γεύματα και για την εμφάνιση τους στο κατάλληλο γραφικό περιβάλλον.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

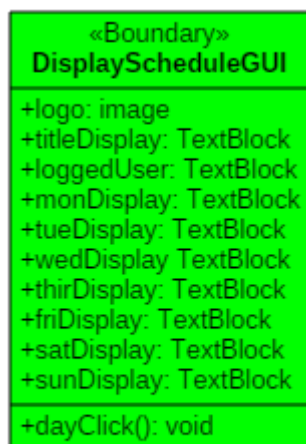
- **displayScheduleControl():** Η μέθοδος της κλάσης αυτής αποτελεί τον Constructor της κλάσης μέσω του οποίου δημιουργείται ένα καινούριο αντικείμενο.
- **createScheduleTable():** Είναι η μέθοδος μέσω της οποίας δημιουργείται ο πίνακας του προγράμματος σίτισης που περιέχει όλα τα διαθέσιμα φαγητά του εβδομαδιαίου προγράμματος



σίτισης. Τα στοιχεία αυτά φορτώνονται από τη Βάση Δεδομένων μέσω της κλάσης Scheduling-DBProxy.

- **displayScheduleGUI(scheduleTable):** Είναι η μέθοδος μέσω της οποίας δημιουργείται το γραφικό περιβάλλον απεικόνισης του προγράμματος σίτισης έχοντας ως όρισμα τον πίνακα που δημιουργήθηκε μέσω της συνάρτησης createSchedulingTable().

Boundary DisplayScheduleGUI



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος του προγράμματος σίτισης.

Οι μεταβλητές της κλάσης αυτής είναι οι εξής:

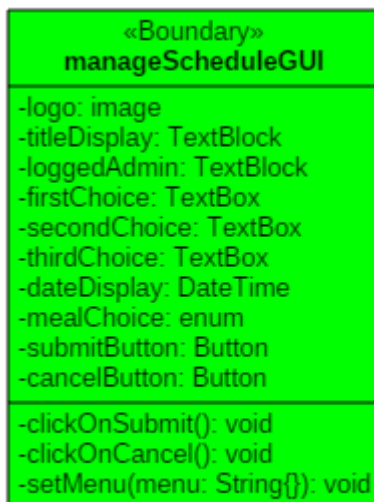
- **logo:** Το λογότυπο της εφαρμογής.
- **titleDisplay:** Ο τίτλος που απεικονίζει την λειτουργικότητα της σελίδας.
- **loggedUser:** Ο τίτλος που απεικονίζει την ιδιότητα του συνδεδεμένου ή μη χρήστη.
- **mealChoice:** Ο τίτλος που απεικονίζει την επιλογή γεύματος (πρωινό, μεσημεριανό, βραδινό).
- **monDisplay:** Ο τίτλος που απεικονίζει την ημέρα Δευτέρα καθώς και την ημερομηνία.
- **tueDisplay:** Ο τίτλος που απεικονίζει την ημέρα Τρίτη καθώς και την ημερομηνία.
- **wedDisplay:** Ο τίτλος που απεικονίζει την ημέρα Τετάρτη καθώς και την ημερομηνία.
- **thirDisplay:** Ο τίτλος που απεικονίζει την ημέρα Πέμπτη καθώς και την ημερομηνία.
- **friDisplay:** Ο τίτλος που απεικονίζει την ημέρα Παρασκευή καθώς και την ημ/νία.
- **satDisplay:** Ο τίτλος που απεικονίζει την ημέρα Σάββατο καθώς και την ημερομηνία.
- **sunDisplay:** Ο τίτλος που απεικονίζει την ημέρα Κυριακή καθώς και την ημερομηνία.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **dayClick(mealChoice):** Είναι η μέθοδος μέσω της οποίας όταν πατηθεί κάποια από τις ημέρες τότε εμφανίζεται το αντίστοιχο μενού της ημέρας αυτής βάσει το επιλεγμένο γεύμα.



Boundary ManageScheduleGUI



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος στο οποίο ο διαχειριστής καταχωρεί ή τροποποιεί το πρόγραμμα σίτισης.

Οι μεταβλητές της κλάσης αυτής είναι οι εξής:

- **logo**: Το λογότυπο της εφαρμογής.
- **titleDisplay**: Ο τίτλος που απεικονίζει την λειτουργικότητα της σελίδας.
- **loggedAdmin**: Ο τίτλος που απεικονίζει την ιδιότητα του συνδεδεμένου χρήστη.
- **firstChoice**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την 1^η διαθέσιμη επιλογή φαγητού.
- **secondChoice**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την 2^η διαθέσιμη επιλογή φαγητού.
- **thirdChoice**: Μία μεταβλητή τύπου TextBox στο οποίο ο διαχειριστής συμπληρώνει την 3^η διαθέσιμη επιλογή φαγητού.
- **dateDisplay**: Η μεταβλητή στην οποία ο διαχειριστής ορίζει την ημερομηνία της οποίας θέλει να τροποποιήσει ή να καταχωρήσει το πρόγραμμα σίτισης.
- **mealChoice**: Ο τίτλος που απεικονίζει την επιλογή γεύματος (πρωινό, μεσημεριανό, βραδινό).
- **submitButton**: Μία μεταβλητή τύπου Button το οποίο όταν πατηθεί αποστέλλει την ψηφοφορία στη Βάση Δεδομένων ή εμφανίζει μήνυμα σφάλματος εάν ο έλεγχος των στοιχείων αποτύχει.
- **cancelButton**: Μία μεταβλητή τύπου Button το οποίο όταν πατηθεί το σύστημα επιστρέφει στην αρχική σελίδα της εφαρμογής.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **clickOnSubmit()**: Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί submitButton καλείται ο Controller ManageScheduleControl για τον έλεγχο των πεδίων και την αποστολή της ψηφοφορίας στη Βάση Δεδομένων.



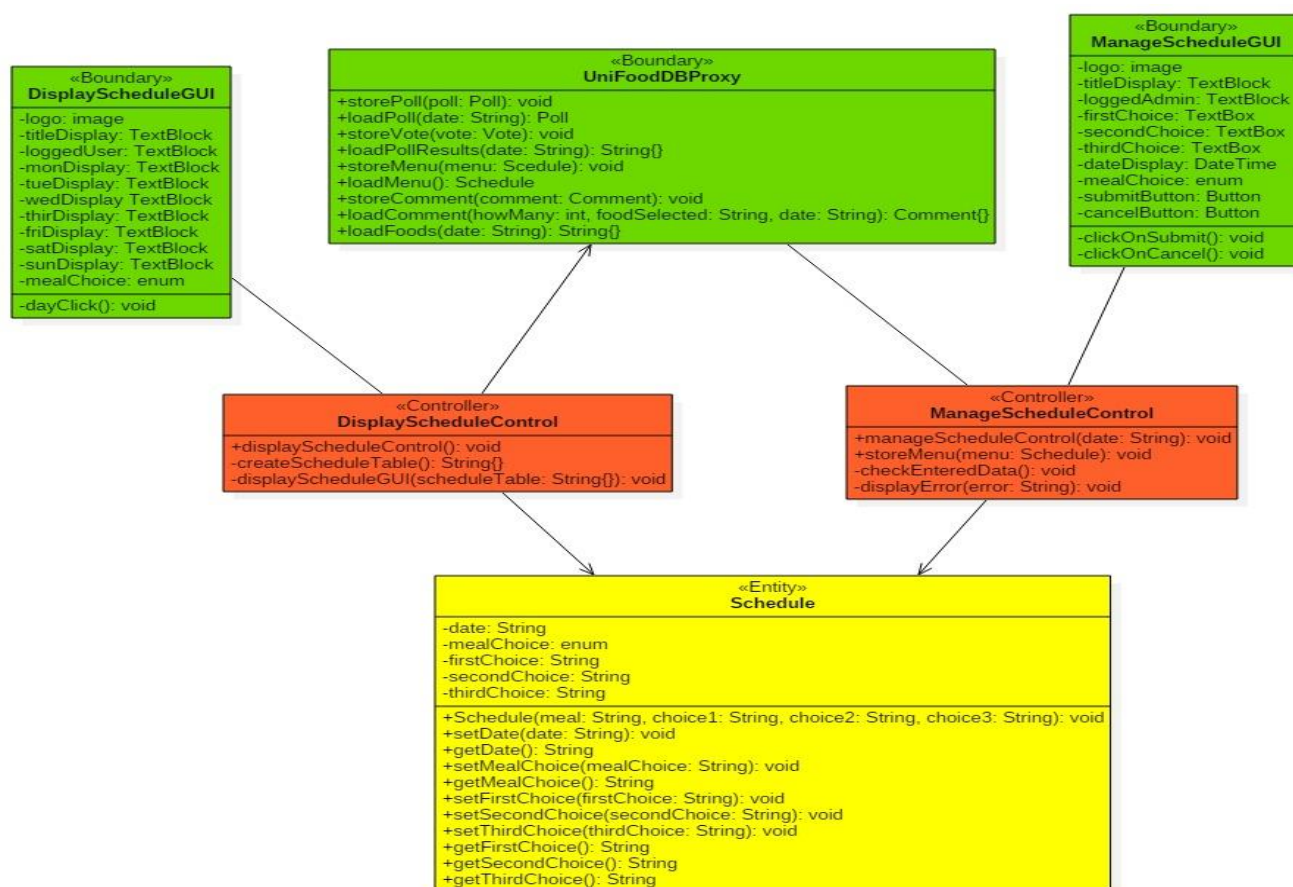
- **clickOnCancel()**: Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί cancelButton το σύστημα επιστρέφει στην αρχική σελίδα της εφαρμογής.

Boundary UniFoodDBProxy

«Boundary» UniFoodDBProxy
+storePoll(poll: Poll): void +loadPoll(date: String): Poll +storeVote(vote: Vote): void +loadPollResults(date: String): String[] +storeMenu(menu: Scedule): void +loadMenu(): Schedule +storeComment(comment: Comment): void +loadComment(howMany: int, foodSelected: String, date: String): Comment[] +loadFoods(date: String): String[]

Η κλάση αυτή είναι υπεύθυνη για την επικοινωνία με τη Βάση Δεδομένων δίνοντας τη δυνατότητα αποθήκευσης και φόρτωσης του προγράμματος σίτισης.

Η κλάση αυτή είναι ίδια με τη κλάση UniFoodDBProxy του πακέτου VotingCapabilities και η επεξήγηση των μεθόδων βρίσκεται στην παράγραφο 2.1.1.



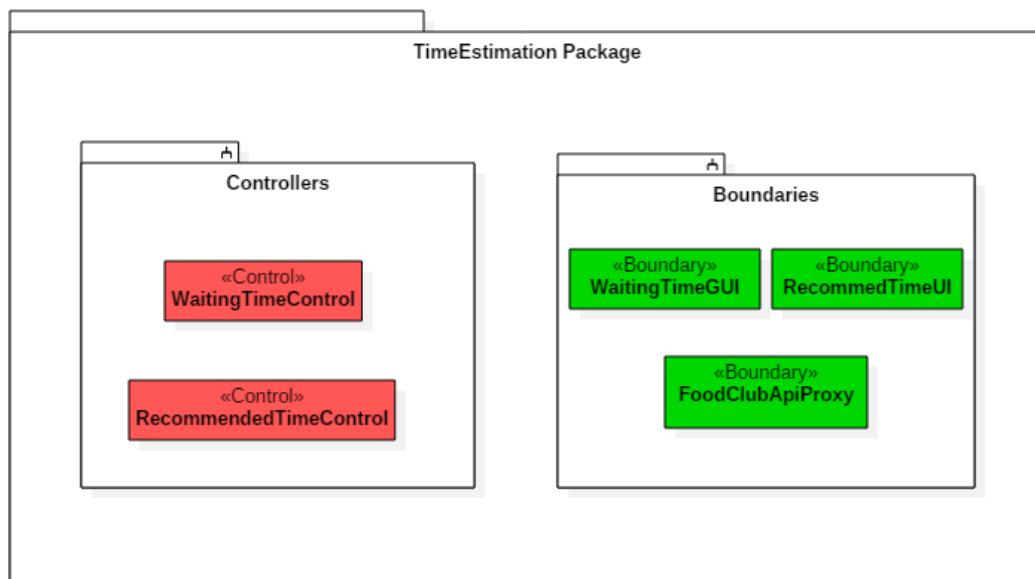
Σχήμα 2 – Διάγραμμα κλάσεων πακέτου ScheduleCababilities



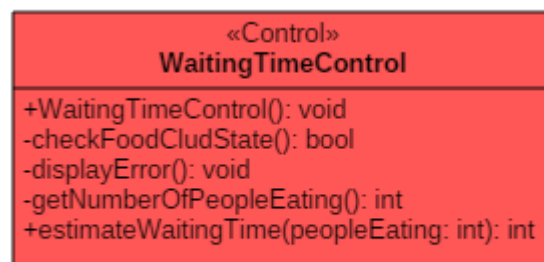
2.2 Πακέτο λεξιλογίου σεναρίων μέσης προτεραιότητας

2.2.2 Πακέτο TimeEstimation

Το πακέτο αυτό περιλαμβάνει την λειτουργικότητα που είναι απαραίτητη για την εμφάνιση του εκτιμώμενου χρόνου αναμονής στη λέσχη αλλά και της καλύτερης ώρας για να επισκεφτεί κανείς τη λέσχη.



Controller WaitingTimeControl



Η κλάση αυτή ελέγχει τη διαδικασία εμφάνισης του εκτιμώμενου χρόνου αναμονής στη λέσχη από το χρήστη τη δεδομένη χρονική στιγμή. Συνοπτικά, ελέγχει το αν η λέσχη είναι ανοιχτή, χρησιμοποιεί το API της πανεπιστημιακής λέσχης για να αναγνωρίσει το πόσοι φοιτητές σιτίζονται σε πραγματικό χρόνο, εκτελεί ειδικό αλγόριθμο και τελικά εμφανίζει τον εκτιμώμενο χρόνο αναμονής.

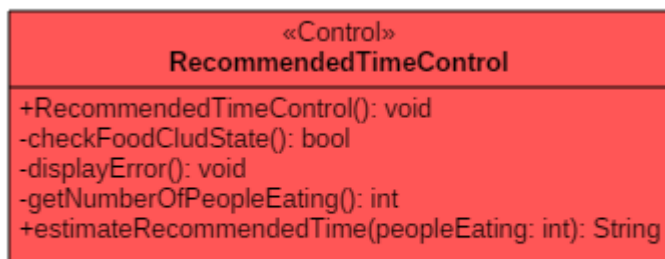
Οι μέθοδοι αυτής της κλάσης είναι οι εξής:

- **WaitingTimeControl():** Η μέθοδος αυτή αποτελεί τον Constructor της κλάσης μέσω του οποίου δημιουργείται ένα καινούργιο αντικείμενο.



- **checkFoodClubState():** Η μέθοδος αυτή ελέγχει το αν είναι ανοιχτή η λέσχη ή όχι τη δεδομένη χρονική στιγμή. Επιστρέφει true ή false αντιστοίχως.
- **displayError():** Η μέθοδος αυτή εμφανίζει μήνυμα σφάλματος σε περίπτωση που ο χρήστης επιχειρήσει να δει τον εκτιμώμενο χρόνο αναμονής και η λέσχη είναι κλειστή.
- **getNumberOfPeopleEating():** Η μέθοδος αυτή χρησιμοποιεί το API της λέσχης για να πάρει τον αριθμό των σιτιζομένων στη λέσχη τη δεδομένη χρονική στιγμή. Επιστρέφει τον αριθμό αυτό ως έναν integer.
- **estimateEatingTime(peopleEating):** Η μέθοδος αυτή υλοποιεί το αλγόριθμο που δέχεται τον αριθμό των σιτιζομένων και παράγει και επιστρέφει τον χρόνο αναμονής στην ουρά της λέσχης ως έναν integer.

Controller RecommendedTimeControl



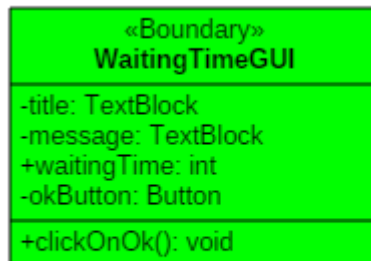
Η κλάση αυτή ελέγχει τη διαδικασία εμφάνισης της προτεινόμενης ώρας για να επισκεφτεί κανείς τη λέσχη τη συγκεκριμένη μέρα. Συνοπτικά, ελέγχει το αν η λέσχη είναι ανοιχτή, χρησιμοποιεί το API της πανεπιστημιακής λέσχης για να αναγνωρίσει το πόσοι φοιτητές σιτίζονται σε πραγματικό χρόνο, εκτελεί ειδικό αλγόριθμο και τελικά εμφανίζει προτεινόμενη ώρα για να πάει ο φοιτητής στη λέσχη.

Οι μέθοδοι αυτής τη κλάσης είναι οι εξής:

- **WaitingTimeControl():** Η μέθοδος αυτή αποτελεί τον Constructor της κλάσης μέσω του οποίου δημιουργείται ένα καινούργιο αντικείμενο.
- **checkFoodClubState():** Η μέθοδος αυτή ελέγχει το αν είναι ανοιχτή η λέσχη ή όχι τη δεδομένη χρονική στιγμή. Επιστρέφει true ή false αντιστοίχως.
- **displayError():** Η μέθοδος αυτή εμφανίζει μήνυμα σφάλματος σε περίπτωση που ο χρήστης επιχειρήσει να δει την προτεινόμενη ώρα και η λέσχη είναι κλειστή.
- **getNumberOfPeopleEating():** Η μέθοδος αυτή χρησιμοποιεί το API της λέσχης για να πάρει τον αριθμό των σιτιζομένων στη λέσχη τη δεδομένη χρονική στιγμή. Επιστρέφει τον αριθμό αυτό ως έναν integer.
- **estimateEatingTime(peopleEating):** Η μέθοδος αυτή υλοποιεί το αλγόριθμο που δέχεται τον αριθμό των σιτιζομένων και παράγει και επιστρέφει την προτεινόμενη ώρα να επισκεφτεί την Λέσχη ως ένα String.



Boundary WaitingTimeGUI



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος υπό τη μορφή απλού μηνύματος που δείχνει τον εκτιμώμενο χρόνο αναμονής στην λέσχη.

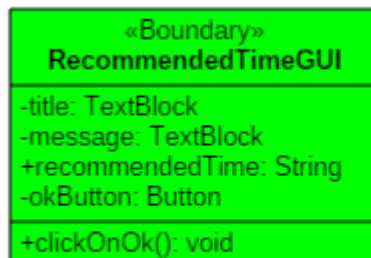
Οι μεταβλητές της κλάσης αυτής είναι:

- **title**: Μία σταθερή μεταβλητή τύπου TextBlock που αποτελεί τον τίτλο του μηνύματος.
- **message**: Μία σταθερή μεταβλητή τύπου TextBlock που αποτελεί το περιεχόμενο του μηνύματος.
- **waitingTime**: Μία μεταβλητή τύπου integer η οποία κρατά τα λεπτά που αναμένεται να περιμένει ο χρήστης στην ουρά της λέσχης.
- **okButton**: Μία μεταβλητή τύπου Button το οποίο όταν πατηθεί κλείνει το παράθυρο μηνύματος επιστρέφοντας στην αρχική σελίδα.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **clickOnOk()**: Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί okButton, το μήνυμα κλείνει και το σύστημα επιστρέφει στην αρχική σελίδα της εφαρμογής.

Boundary RecommendedTimeGUI



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος υπό τη μορφή απλού μηνύματος που δείχνει την προτεινόμενη ώρα για να επισκεφτεί κανείς τη λέσχη.



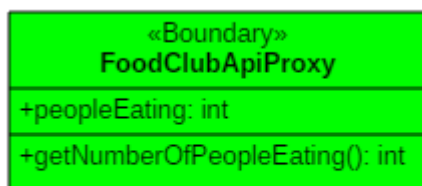
Οι μεταβλητές της κλάσης αυτής είναι:

- **title**: Μία σταθερή μεταβλητή τύπου TextBlock που αποτελεί τον τίτλο του μηνύματος.
- **message**: Μία σταθερή μεταβλητή τύπου TextBlock που αποτελεί το περιεχόμενο του μηνύματος.
- **recommendedTime**: Μία μεταβλητή τύπου String η οποία κρατά την προτεινόμενη ώρα για να επισκεφτεί κανείς τη λέσχη.
- **okButton**: Μία μεταβλητή τύπου Button το οποίο όταν πατηθεί κλείνει το παράθυρο μηνύματος επιστρέφοντας στην αρχική σελίδα.

Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

- **clickOnOk()**: Είναι η μέθοδος μέσω της οποίας όταν πατηθεί το κουμπί okButton, το μήνυμα κλείνει και το σύστημα επιστρέφει στην αρχική σελίδα της εφαρμογής.

Boundary FoodClubApiProxy



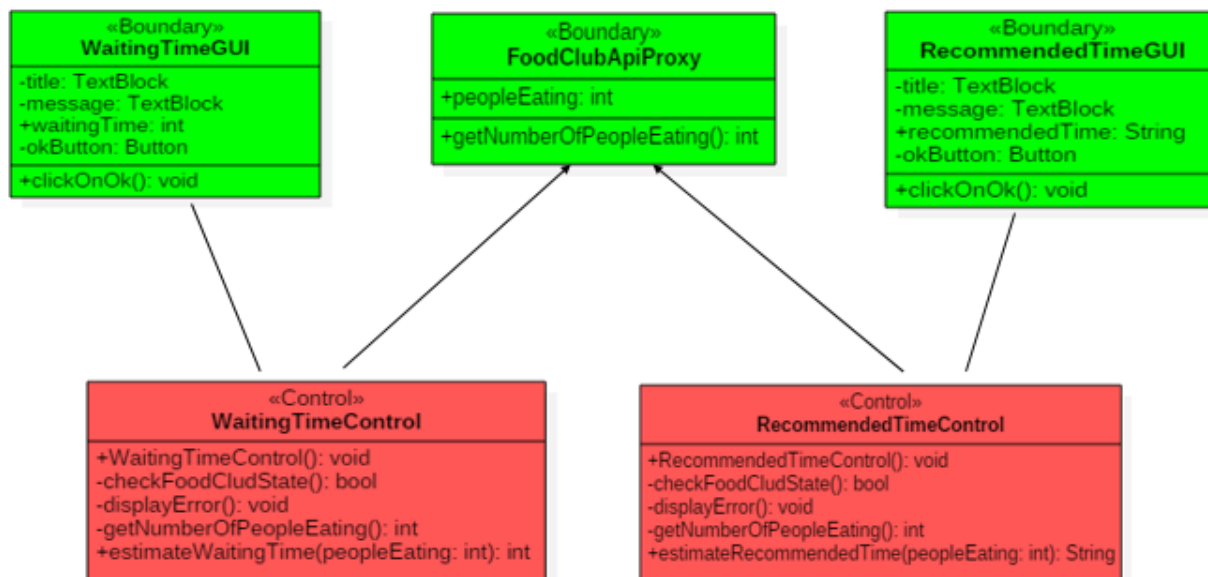
Η κλάση αυτή είναι υπεύθυνη για την επικοινωνία με το εξωτερικό συστήματα της Πανεπιστημιακής Λέσχης μέσω του API που διαθέτει, έτσι ώστε να το σύστημα του UniFood να έχει πρόσβαση στον αριθμό των σιτιζομένων ανά πάσα χρονική στιγμή.

Η μεταβλητή της κλάσης είναι:

- **peopleEating**: Είναι η μεταβλητή που κρατάει τον αριθμό των ατόμων που σιτίζονται στη λέσχη.

Η μέθοδος της κλάσης είναι:

- **getNumberOfPeopleEating()**: Η μέθοδος αυτή επικοινωνεί μέσω του API με το σύστημα της λέσχης και διαβάζει τον αριθμό των σιτιζομένων.



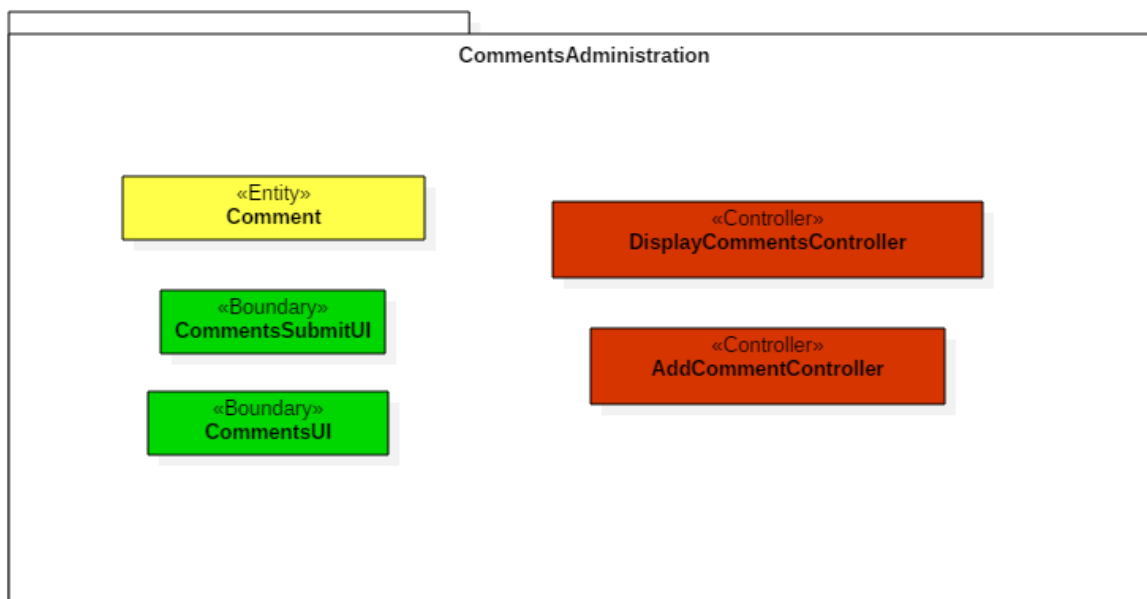
Σχήμα 3 – Διάγραμμα κλάσεων πακέτου TimeEstimation



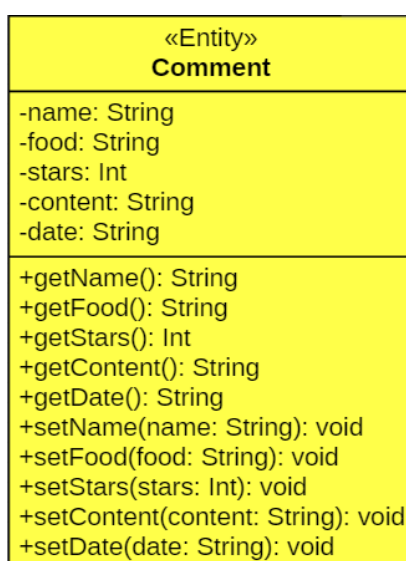
2.3 Πακέτο λεξιλογίου σεναρίων χαμηλής προτεραιότητας

2.3.1 Πακέτο CommentsAdministration

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που απαιτούνται για την καταχώρηση ενός σχολίου από κάποιον χρήστη καθώς και αυτών που απαιτούνται για την προβολή αυτών των σχολίων. Επιπλέον, περιλαμβάνονται και οι κλάσεις με τα αντίστοιχα γραφικά περιβάλλοντα της εφαρμογής.



Entity Comment



Η κλάση **Comment** περιέχει τις απαραίτητες μεταβλητές που χρειάζονται για την καταχώρηση των σχολίων από τους χρήστες.



Οι μεταβλητές είναι οι εξής:

- **name:** μια μεταβλητή τύπου string όπου περιέχει το όνομα του χρήστη που έχει καταθέσει το συγκεκριμένο σχόλιο
- **food:** μια μεταβλητή τύπου string όπου περιέχει το φαγητό για το οποίο ο χρήστης καταθέτει το σχόλιο
- **stars:** μια μεταβλητή τύπου int η οποία περιέχει την πληροφορία του με πόσα αστεράκια αξιολογεί ο χρήστης την εφαρμογή
- **content:** μια μεταβλητή τύπου string η οποία περιέχει το σχόλιο του χρήστη.
- **date:** μια μεταβλητή τύπου string η οποία περιέχει με κατάλληλο τρόπο την ημερομηνία δημιουργίας του αντικειμένου της κλάσης

Controller AddCommentController

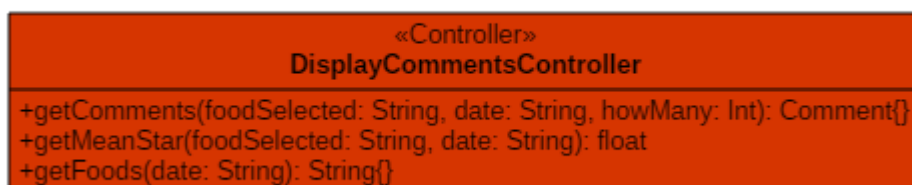


Η κλάση αυτή περιλαμβάνει την απαραίτητη μέθοδο για την καταχώρηση ενός σχολίου από τον χρήστη.

Η συνάρτηση αυτή περιγράφεται παρακάτω:

- **submitCommentControl(Comment):** Είναι η μέθοδος η οποία δέχεται ως όρισμα το σχόλιο του χρήστη και καλεί την κατάλληλη μέθοδο για την αποθήκευση του σχολίου στην βάση δεδομένων.

Controller DisplayCommentsController



Η κλάση αυτή περιλαμβάνει τις απαραίτητες μεθόδους για να ανακληθούν σχόλια και να δοθούν στην κατάλληλη κλάση γραφικού περιβάλλοντος ώστε να προβληθούν στον χρήστη.

Οι συναρτήσεις αυτές είναι οι εξής:

- **getComment(foodSelected: String, date: String, howMany: Integer): Comment[]:** Πρόκειται για την μέθοδο η οποία επιστρέφει τα πιο πρόσφατα σχόλια ("howMany") για το φαγητό που



επέλεξε νωρίτερα ο χρήστης ("foodSelected").

- **getMeanStar(foodSelected: String, date: String): float:** Πρόκειται για την συνάρτηση που επιστρέφει το μέσο όρο αστεριών που έχει λάβει το φαγητό "food".
- **getFoods(date:String):String[] :** Πρόκειται για την συνάρτηση που επιστρέφει τα διαθέσιμα φαγητά της ημερομηνίας "date".

Boundary CommentsUI

«Boundary» CommentsUI
-date: string -food: Enum -StarReview: Image -comments: Textbox
+showComment(foodSelected: String, date: String, howMany: Int): void +showRating(foodSelected: String, date: String): void +chooseDate(date: String): void

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος όπου ο χρήστης θα μπορεί να βλέπει τα σχόλια των άλλων χρηστών.

Μεταβλητές της κλάσης αυτής είναι:

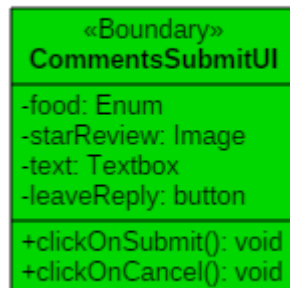
- **date:** Η ημερομηνία για την οποία ζητά ο χρήστης να δει σχόλια.
- **food:** Το φαγητό για το οποίο ο χρήστης ενδιαφέρεται να δει σχόλια.
- **starReview:** Ο μέσος όρος αξιολογήσεων που έλαβε το συγκεκριμένο φαγητό την συγκεκριμένη ημέρα.
- **comments:** Το σχόλιο του χρήστη που εμφανίζεται.

Συναρτήσεις της κλάσης αυτής είναι:

- **showRating(foodSelected:String, date:String):** η μέθοδος αυτή καλεί την μέθοδο getMean της κλάσης DisplayCommentsController και σχεδιάζει στο γραφικό περιβάλλον την εικόνα starReview που αντιστοιχεί στον μέσο όρο των αξιολογήσεων.
- **showComment(food:String, date:String, number:Integer):** η μέθοδος αυτή καλεί την getComments της κλάσης DisplayCommentsController και αποτυπώνει στην οθόνη τα τελευταία σχόλια των χρηστών για το συγκεκριμένο φαγητό.
- **chooseDate(date: String):** Καταθέτει την ημερομηνία για να επιστραφούν τα φαγητά που ήταν διαθέσιμα εκείνη την ημέρα.



Boundary CommentsSubmitUI



Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του γραφικού περιβάλλοντος όπου ο χρήστης θα μπορεί να καταχωρήσει ένα νέο σχόλιο για κάποιο φαγητό.

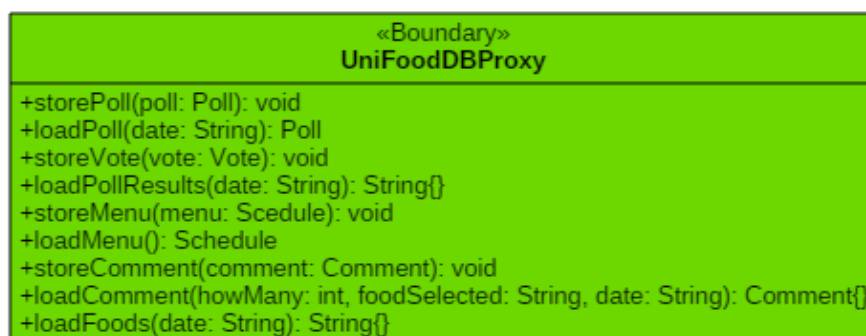
Μεταβλητές της κλάσης αυτής είναι:

- **food:** Πρόκειται για το φαγητό για το οποίο ο χρήστης επιθυμεί να αφήσει κάποιο σχόλιο.
- **starReview:** Η βαθμολογικού τύπου αξιολόγηση του φαγητού
- **text:** Το κείμενο του σχολίου του χρήστη.
- **leaveReply:** Το κουμπί για την κατάθεση σχολίου.
- **cancel:** Το κουμπί για την ακύρωση της διαδικασίας.

Συναρτήσεις της κλάσης αυτής είναι:

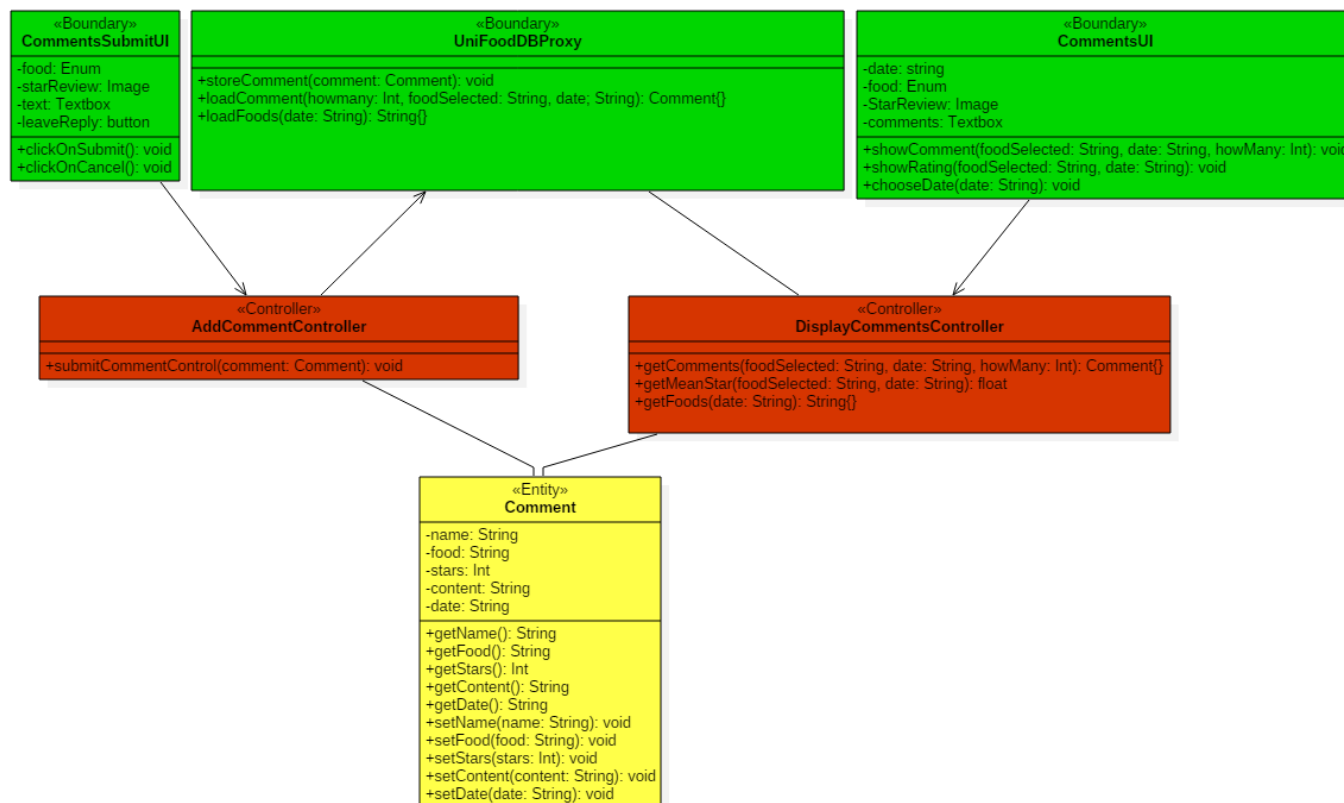
- **clickOnSubmit():** το κουμπί για την δημιουργία ενός αντικειμένου comment και την κατάθεση του ως σχολίου
- **clickOnCancel():** το κουμπί για την ακύρωση της διαδικασίας

Boundary UniFoodDBProxy



Η κλάση αυτή είναι υπεύθυνη για την επικοινωνία με τη Βάση Δεδομένων δίνοντας τη δυνατότητα αποθήκευσης και φόρτωσης των σχολίων των χρηστών.

Η κλάση αυτή είναι ίδια με τη κλάση UniFoodDBProxy του πακέτου VotingCapabilities και η επεξήγηση των μεθόδων βρίσκεται στην παράγραφο 2.1.1.



Σχήμα 4 – Διάγραμμα κλάσεων πακέτου CommentsAdministration



3 Απαιτήσεις Λογισμικού

3.1 Δυναμική μοντελοποίηση του συστήματος

3.1.1 Πακέτο ψηφοφορίας χρήση

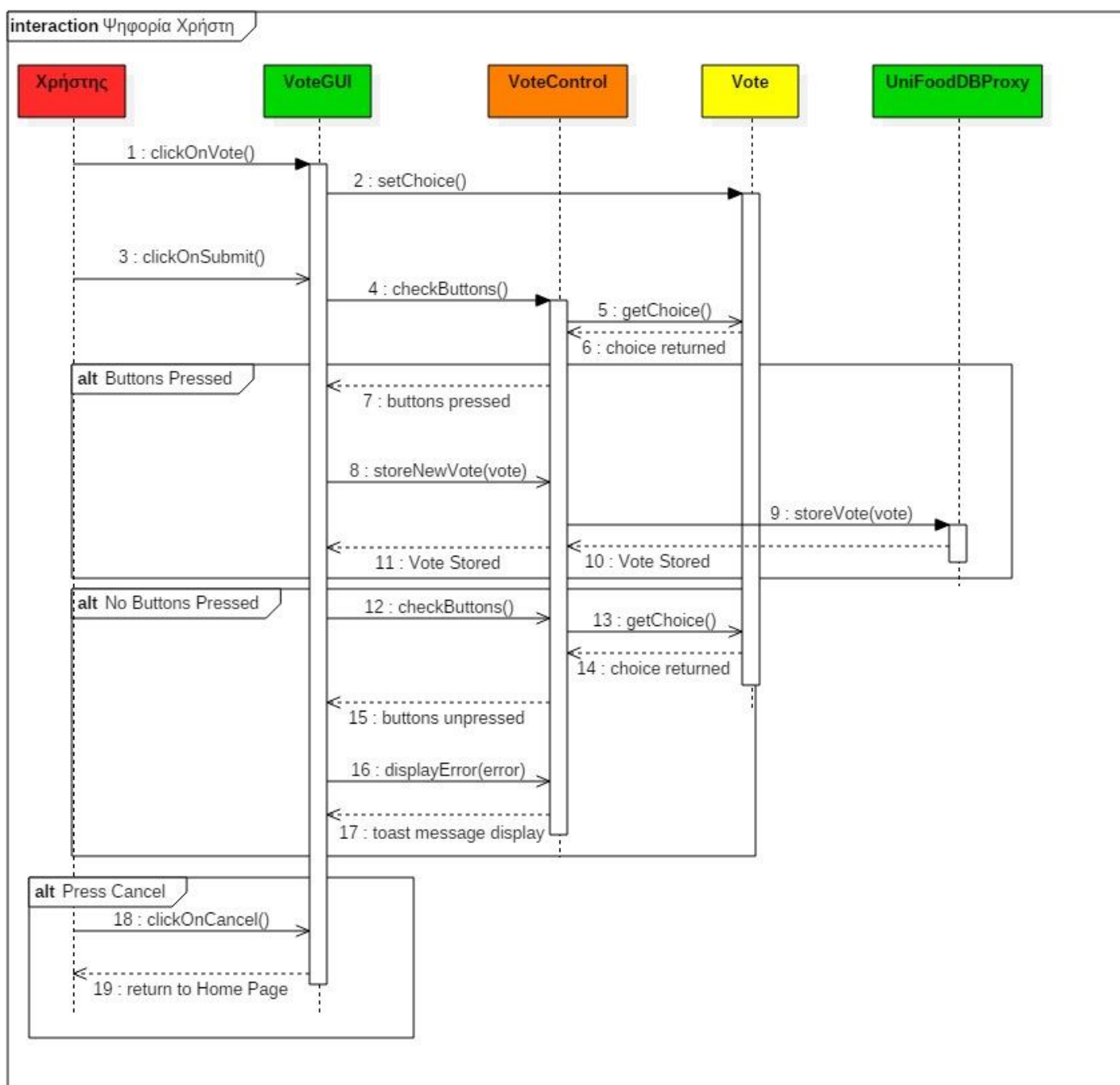
Το πακέτο αυτό αναφέρεται στη δυνατότητα του χρήστη να ψηφίζει για το πρόγραμμα σίτισης βάσει των επιλογών που έχουν δοθεί από τον διαχειριστή. Στη σελίδα ψηφοφορίας δίνεται η δυνατότητα να επιλέξει ο χρήστης ένα από τα έξι προτεινόμενα φαγητά. Αν ο χρήστης πατήσει «Υποβολή» δίχως να έχει επιλέξει κανένα από τα διαθέσιμα φαγητά εμφανίζεται μήνυμα σφάλματος και το σύστημα παραμένει στη σελίδα ψηφοφορίας για επιλογή του προτιμώμενου φαγητού. Τέλος αν ο χρήστης επιλέξει «Άκυρο» το σύστημα δεν αποθηκεύει την επιλογή του χρήστη και επιστρέφει στην αρχική σελίδα της εφαρμογής.

Η λειτουργική απαίτηση που σχετίζεται με το συγκεκριμένο πακέτο είναι η εξής:

- <ΛΑ-6> Ο χρήστης πρέπει να μπορεί να ψηφίζει για το πρόγραμμα σίτισης της επόμενης εβδομάδας για την αντίστοιχη μέρα.

Αφήγηση Σεναρίου

Ο χρήστης βρίσκεται στη σελίδα ψηφοφορίας. Στη συνέχεια ο χρήστης επιλέγει ένα από τα έξι διαθέσιμα φαγητά που βρίσκονται στη σελίδα. Μόλις ο χρήστης πατήσει το αντίστοιχο κουμπί η κλάση Vote καλεί τη συνάρτηση `setChoice(choice)`, η οποία αποθηκεύει προσωρινά την επιλογή του χρήστη. Όταν ο χρήστης πατήσει «Υποβολή» για αποθήκευση της ψήφου του καλείται η συνάρτηση `clickOnSubmit()` της κλάσης `VoteGUI` η οποία με τη σειρά της καλεί τη συνάρτηση `checkButtons()` του ελεγκτή `VoteControl`. Η συνάρτηση αυτή ελέγχει εάν έχει πατηθεί κάποιο κουμπί από τα έξι διαθέσιμα, αφού πρώτα καλέσει τη συνάρτηση `getChoice()` της κλάσης `Vote` η οποία επιστρέφει την επιλογή του χρήστη. Στη συνέχεια ο ελεγκτής `VoteControl` καλεί τη συνάρτηση `storeNewVote()` για αποθήκευση της ψήφου, η οποία με τη σειρά της καλεί τη συνάρτηση `storeVote()` του οριακού αντικειμένου `UniFoodDBProxy`, με σκοπό την αποθήκευση της ψήφου στη Βάση Δεδομένων. Εάν ο χρήστης δεν έχει επιλέξει κανένα από τα έξι κουμπιά, τότε η συνάρτηση `checkButtons()` καλεί με τη σειρά της τη συνάρτηση `displayError(error)` του ελεγκτή `VoteControl`, η οποία εμφανίζει μήνυμα σφάλματος στον χρήστη και επιστρέφει το σύστημα στην σελίδα ψηφοφορίας για εκ νέου επιλογή του χρήστη. Τέλος αν ο χρήστης σε οποιαδήποτε στιγμή πατήσει το κουμπί «Άκυρο» τότε καλείται η συνάρτηση `clickOnCancel()` της κλάσης `VoteGUI` και το σύστημα επιστρέφει στην αρχική σελίδα.



Σχήμα 5 – Σενάριο ψηφοφορίας χρήστη



3.1.2 Πακέτο διαχείρισης προγράμματος σίτισης από τον διαχειριστή

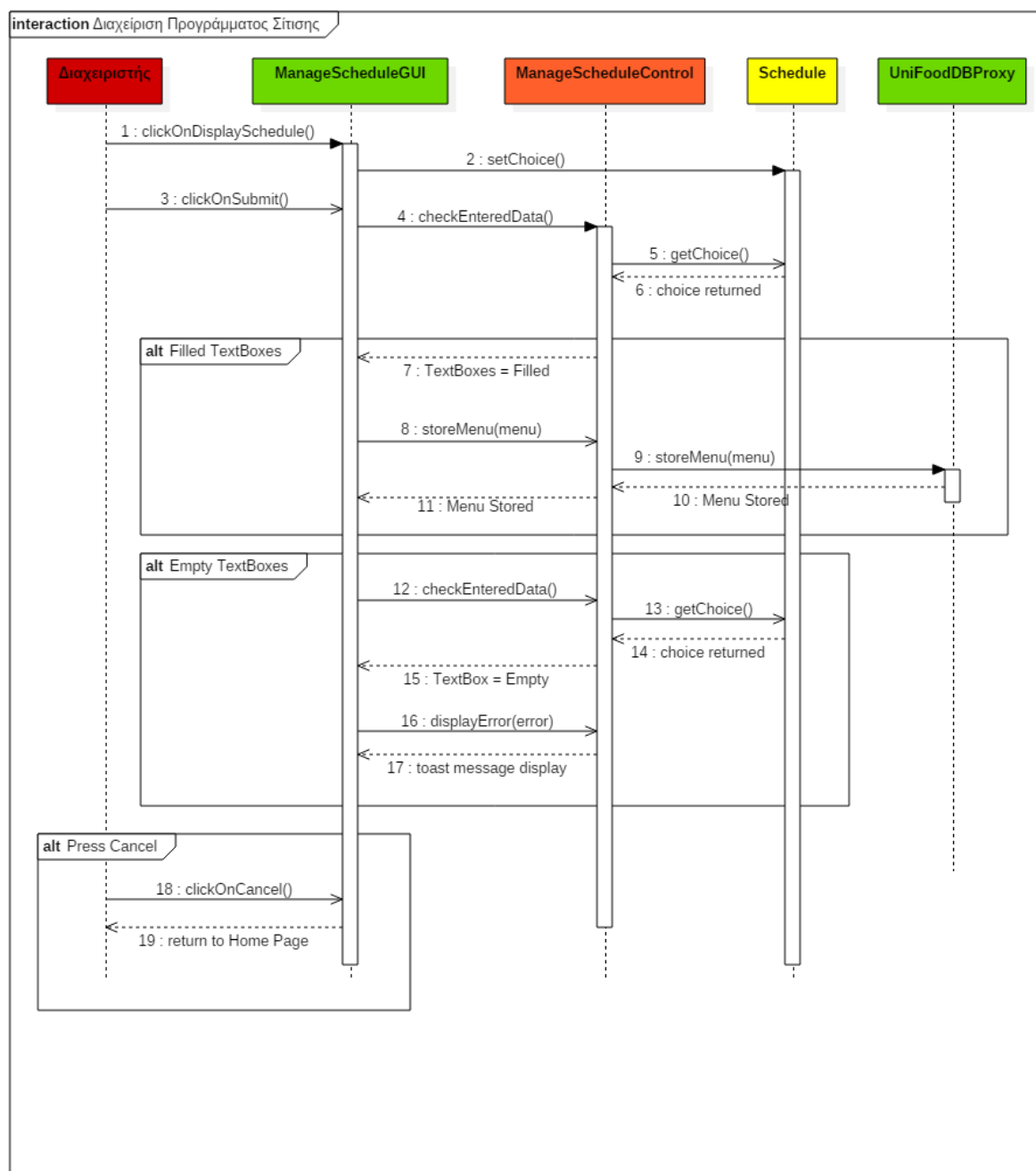
Το πακέτο αυτό αναφέρεται στη δυνατότητα του διαχειριστή να καταχωρεί ή να τροποποιεί το πρόγραμμα σίτισης βάσει των αποτελεσμάτων της ψηφοφορίας. Στη σελίδα της καταχώρησης/τροποποίησης του προγράμματος σίτισης δίνεται η δυνατότητα να επιλέξει ο διαχειριστής την επιθυμητή ημερομηνία αλλά και το γεύμα. Ακολουθώντας καταχωρεί τις επιλογές στα 3 TextBoxes. Αν ο διαχειριστής πατήσει «Υποβολή» αφήνοντας όλα τα TextBoxes άδεια, τότε εμφανίζεται μήνυμα σφάλματος και το σύστημα παραμένει στη σελίδα καταχώρησης/τροποποίησης του προγράμματος σίτισης. Τέλος αν ο διαχειριστής επιλέξει «Άκυρο» το σύστημα δεν αποθηκεύει την καταχώρηση του διαχειριστή και επιστρέφει στην αρχική σελίδα της εφαρμογής.

Η λειτουργική απαίτηση που σχετίζεται με το συγκεκριμένο πακέτο είναι η εξής:

- <ΛΑ-1> Ο διαχειριστής της λέσχης πρέπει να διαχειρίζεται το πρόγραμμα σίτισης.

Αφήγηση Σεναρίου

Ο διαχειριστής βρίσκεται στη σελίδα διαχείρισης. Στη συνέχεια ο διαχειριστής επιλέγει την ημερομηνία που επιθυμεί να καταχωρήσει/τροποποιήσει το πρόγραμμα σίτισης και ακολούθως επιλέγει το γεύμα. Μόλις ο διαχειριστής καταχωρήσει κάποιο φαγητό, η κλάση Schedule καλεί τη συνάρτηση `setChoice(choice)`, η οποία αποθηκεύει προσωρινά την επιλογή του διαχειριστή. Όταν ο διαχειριστής πατήσει «Υποβολή» για να αποθηκεύσει το τις καταχωρήσεις που μόλις έκανε, καλείται η συνάρτηση `clickOnSubmit()` της κλάσης `ManageScheduleGUI` η οποία με τη σειρά της καλεί τη συνάρτηση `checkEnteredData()` του ελεγκτή `ManageScheduleControl`. Η συνάρτηση αυτή ελέγχει εάν ο διαχειριστής καταχώρησε κάποιο φαγητό, αφού πρώτα καλέσει τη συνάρτηση `getChoice()` της κλάσης `Schedule` η οποία επιστρέφει την καταχώρηση του διαχειριστή. Στη συνέχεια ο ελεγκτής `ManageScheduleControl` καλεί τη συνάρτηση `storeMenu(menu)` για αποθήκευση του μενού που καταχώρησε ο διαχειριστής, η οποία με τη σειρά της καλεί τη συνάρτηση `storeMenu(menu)` του οριακού αντικειμένου `UniFoodDBProxy`, με σκοπό την αποθήκευση της καταχώρησης του προγράμματος σίτισης στη Βάση Δεδομένων. Εάν ο διαχειριστής δεν έχει καταχωρήσει καμία επιλογή, τότε η συνάρτηση `checkEnteredData()` καλεί με τη σειρά της τη συνάρτηση `displayError(error)` του ελεγκτή `ManageScheduleControl`, η οποία εμφανίζει μήνυμα σφάλματος στον διαχειριστή και το σύστημα παραμένει στην σελίδα διαχείρισης του προγράμματος σίτισης μέχρι ο διαχειριστής να δώσει έστω και μια καταχώρηση ή να επιλέξει «Άκυρο». Εάν ο διαχειριστής σε οποιαδήποτε στιγμή επιλέξει «Άκυρο» τότε καλείται η συνάρτηση `clickOnCancel()` της κλάσης `ManageScheduleGUI` και το σύστημα επιστρέφει στην αρχική σελίδα.



Σχήμα 6 – Σενάριο διαχείρισης του προγράμματος σίτισης από τον διαχειριστή



3.1.3 Πακέτο διαχείρισης προγράμματος σίτισης από τον διαχειριστή

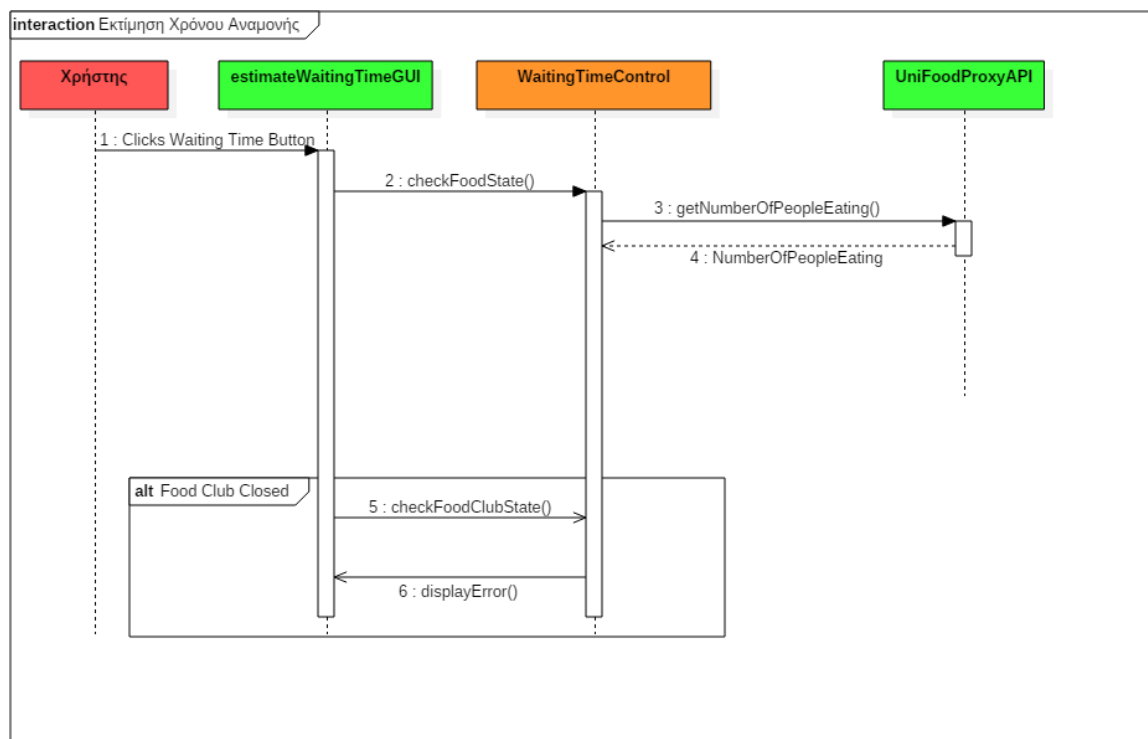
Το πακέτο αυτό αναφέρεται στη δυνατότητα του χρήστη να διαβάσει τον εκτιμώμενο χρόνο αναμονής στην ουρά της Λέσχης σε περίπτωση που φτάσει την εκάστοτε χρονική στιγμή. Ο χρόνος εμφανίζεται με τη μορφή “pop-up” μηνύματος, το οποίο διαθέτει μόνο την επιλογή “OK”, η οποία επαναφέρει τον χρήστη στην αρχική οθόνη. Σε περίπτωση που ζητήσει να διαβάσει τον εκτιμώμενο χρόνο σε χρονική στιγμή που δεν είναι ανοιχτή η Λέσχη τότε αντί για το παραπάνω μήνυμα, δέχεται μήνυμα λάθους και επιστρέφει και πάλι στην αρχική οθόνη.

Η λειτουργική απαίτηση που σχετίζεται με το συγκεκριμένο πακέτο είναι η εξής:

- <ΛΑ-7> Ο χρήστης πρέπει να βλέπει τον εκτιμώμενο χρόνο αναμονής στην Λέσχη.

Αφήγηση Σεναρίου

Από τη στιγμή που ο χρήστης επιλέξει να δει τον εκτιμώμενο χρόνο αναμονής στη λέσχη συμβαίνουν τα ακόλουθα. Το σύστημα ελέγχει αν την δεδομένη χρονική στιγμή η Λέσχη είναι ανοιχτή ή όχι μέσω της συνάρτησης `checkFoodState()`. Αν η λέσχη είναι κλειστή τότε εκτελείται η συνάρτηση `displayError()`, εμφανίζεται μήνυμα λάθους και επιστρέφει στην αρχική οθόνη. Αν η Λέσχη είναι ανοιχτή τότε γίνεται επικοινωνία με το εξωτερικό σύστημα ελέγχου των φοιτητών που μπαίνουν στη λέσχη μέσω της συνάρτησης `getNumberOfPeopleEating()`, εκτελείται ο αλγόριθμος πρόβλεψης χρόνου αναμονής που βρίσκεται στη συνάρτηση `estimateWaitingTime()` και τέλος εμφανίζεται ο εκτιμώμενος χρόνος αναμονής υπό τη μορφή μηνύματος. Τέλος αν ο χρήστης επιλέξει “OK” τότε μέσω της `clickOnOk()` επιστρέφει στην αρχική σελίδα.

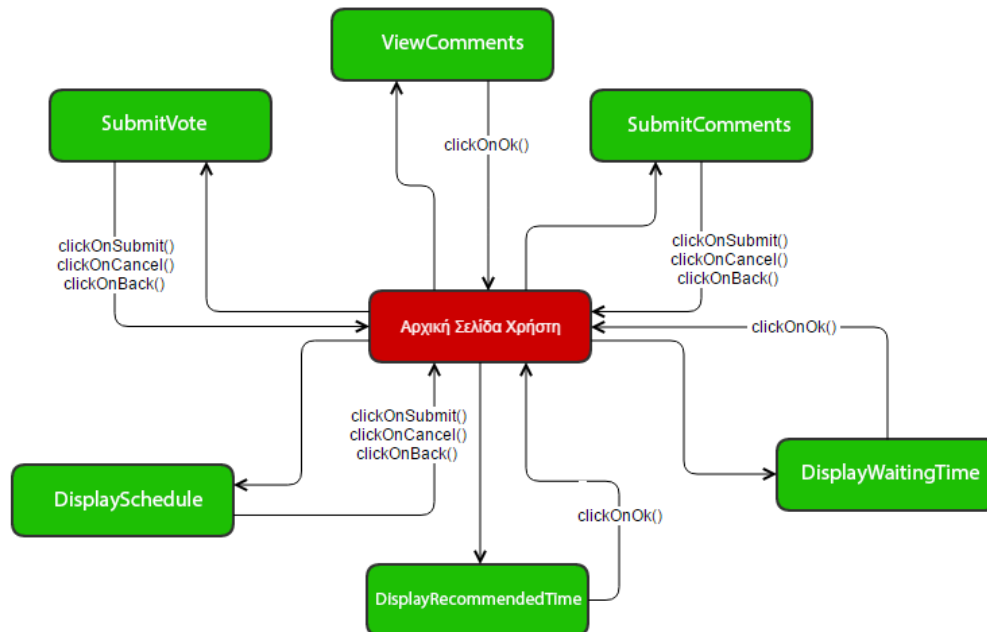


Σχήμα 7 – Σενάριο διαχείρισης του προγράμματος σίτισης από τον διαχειριστή



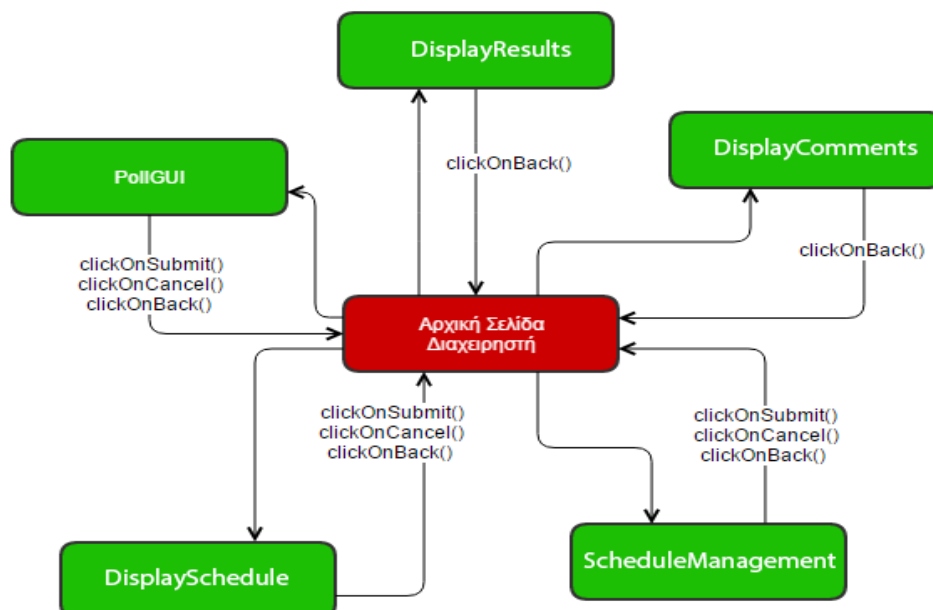
3.2 Διαγράμματα Καταστάσεων

3.2.1 Διάγραμμα Καταστάσεων χρήστη



Σχήμα 8 – Διάγραμμα καταστάσεων χρήστη

3.2.2 Διάγραμμα Καταστάσεων διαχειριστή



Σχήμα 9 – Διάγραμμα καταστάσεων διαχειριστή



3.3 Μη λειτουργικές απαιτήσεις

3.3.1 Απαιτήσεις Φορητότητας (Portability)

<ΜΛΑ-4> Η εφαρμογή θα πρέπει να είναι διαθέσιμη για τα λειτουργικά Android, iOS και Windows 10 Universal.
User Priority: (5/5)
Technical Priority: (5/5)
Stability: (1/5)

3.3.2 Τεχνικές Απαιτήσεις Περιβάλλοντος

<ΜΛΑ-1> Ο χρήστης πρέπει να μπορεί να συνδέεται στο σύστημα μέσω του ιδρυματικού λογαριασμού του.
User Priority: (1/5)
Technical Priority: (4/5)
Stability: (2/5)

<ΜΛΑ-2> Το σύστημα πρέπει να επικοινωνεί με το σύστημα αναγνώρισης ακαδημαϊκής ταυτότητας στην είσοδο της λέσχης.
User Priority: (2/5)
Technical Priority: (5/5)
Stability: (3/5)

3.3.3 Απαιτήσεις πολιτικής (Policy - legal, corporate)

<ΜΛΑ-3> Το σύστημα πρέπει να διασφαλίζει ότι το πρόγραμμα της Λέσχης θα συμβαδίζει με τους διατροφικούς κανόνες που ορίζει το Πανεπιστήμιο σύμφωνα με το άρθρο 23 του Ν.250/1999.
User Priority: (3/5)
Technical Priority: (1/5)
Stability: (4/5)



Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας

- Πακέτο **VotingCapabilities** (Κεφάλαιο 2.1.1):

Λειτουργικές Απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-2>, <ΛΑ-3>, <ΛΑ-4>, <ΛΑ-6>

Σενάρια Χρήσης στα οποία απευθύνεται: <ΣΧ-2>, <ΣΧ-3>, <ΣΧ-4>, <ΣΧ-6>

- Πακέτο **ScheduleCapabilities** (Κεφάλαιο 2.1.2):

Λειτουργικές Απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-1>, <ΛΑ-5>

Σενάρια Χρήσης στα οποία απευθύνεται: <ΣΧ-1>, <ΣΧ-5>

- Πακέτο **TimeEstimation** (Κεφάλαιο 2.2.1):

Λειτουργικές Απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-7>, <ΛΑ-8>, <ΛΑ-11>

Σενάρια Χρήσης στα οποία απευθύνεται: <ΣΧ-7>, <ΣΧ-8>

- Πακέτο **CommentsAdministration** (Κεφάλαιο 2.3.1):

Λειτουργικές Απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-9>, <ΛΑ-10>

Σενάρια Χρήσης στα οποία απευθύνεται: <ΣΧ-9>, <ΣΧ-10>



Παράρτημα II – Ανοιχτά Θέματα

- Είναι πιθανό κατά την υλοποίηση του συστήματος να προστεθούν επιπλέον μεταβλητές οι οποίες θα διευκολύνουν τον προγραμματιστή.
- Κατά την υλοποίηση των αναφερθέντων συναρτήσεων ενδέχεται να αντιληφθούμε την έλλειψη ορισμένων άλλων συναρτήσεων, απαραίτητων για την υλοποίηση της εφαρμογής, και συνεπώς να χρειαστεί να προστεθούν στο παρών παραδοτέο.
- Η υλοποίηση που παρουσιάστηκε έγινε στα πρότυπα της γλώσσας προγραμματισμού C++. Αν η υλοποίηση του συστήματος γίνει με κάποια άλλη αντικειμενοστραφή γλώσσα προγραμματισμού είναι πιθανό να χρειαστούν κάποιες τροποποιήσεις στις κλάσεις που παρουσιάστηκαν.
- Χρησιμοποιώντας την πλατφόρμα Xamarin η οποία πλέον διατίθεται δωρεάν θα μπορέσουμε να αναπτύξουμε την εφαρμογή μας για όλες τις πλατφόρμες χρησιμοποιώντας την ίδια γλώσσα προγραμματισμού, αποφεύγοντας τον κίνδυνο να αλλάξει η δομή της εφαρμογής μας εξαιτίας διαφορετικής γλώσσας υλοποίησης.