



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπο-  
λογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017



# Απαιτήσεις Λογισμικού

## Del.1.2

Version 0.1.4

Μπεκιάρης Θεοφάνης (8200)  
Πεχλιβάνης Νίκος (8228)  
Κελεσάκης Δημήτριος (7898)  
Μπρούζος Ραφαήλ (7945)

[theompek@auth.gr](mailto:theompek@auth.gr)  
[nikopecth@auth.gr](mailto:nikopecth@auth.gr)  
[dkelesak@auth.gr](mailto:dkelesak@auth.gr)  
[mprouzos@auth.gr](mailto:mprouzos@auth.gr)

22/04/2017



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

## Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
A. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegerts <sup>□</sup> και του M. Smialek's.	0.1
A. Συμεωνίδης	29/3/2014	Προσαρμογή του εγγράφου.	0.1.3
Χ. Ζολώτας	2/4/2015	Προσθήκη διαγράμματος καταστάσεων	0.1.4

## Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
A. Συμεωνίδης	*	<a href="mailto:asymeon@issel.ee.auth.gr">asymeon@issel.ee.auth.gr</a>
Μπεκιάρης Θεοφάνης	Reporter 007	<a href="mailto:theompek@auth.gr">theompek@auth.gr</a>
Μπρούζος Ραφαήλ	Reporter 007	<a href="mailto:mprouzos@auth.gr">mprouzos@auth.gr</a>
Κελεσάκης Δημήτριος	Reporter 007	<a href="mailto:dkelesak@auth.gr">dkelesak@auth.gr</a>
Πεχλιβάνης Νίκος	Reporter 007	<a href="mailto:nikopech@auth.gr">nikopech@auth.gr</a>



## Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	3
Λίστα Σχημάτων - Διαγραμμάτων.....	4
1 Στατική Μοντελοποίηση.....	5
1.1 Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας.....	5
1.1.1 Πακέτο UserHandle.....	5
1.1.2 Πακέτο DataAnalysis.....	12
1.2 Πακέτα λεξιλογίου σεναρίων μέσης προτεραιότητας.....	22
1.2.1 Πακέτο DiagramHandle.....	22
1.2.1 Πακέτο QuestionnaireHandle.....	31
1.3 Πακέτο λεξιλογίου σεναρίων χαμηλής προτεραιότητας.....	41
1.3.1 Πακέτο DataHandle.....	41
2 Δυναμική μοντελοποίηση του συστήματος.....	50
2.1 Πακέτο Ανάλυσης Δεδομένων.....	50
2.1.1 Αφήγηση σεναρίου δημιουργίας ρυθμίσεων.....	50
2.1.2 Αφήγηση σεναρίου διαγραφής ρυθμίσεων.....	52
2.2 Πακέτο δημιουργίας διαγράμματος στατιστικών.....	53
2.2.1 Αφήγηση σεναρίου Δημιουργίας Διαγράμματος.....	53
2.3 Πακέτο αναζήτησης δεδομένων και αποθήκευσης δεδομένων.....	54
2.3.1 Αφήγηση σεναρίου Αναζήτησης Δεδομένων.....	54
2.3.2 Αφήγηση σεναρίου Αποθήκευσης Δεδομένων.....	55
2.4 Πακέτο Δημιουργίας και επεξεργασίας ερωτηματολογίων από το χρήστη.....	56
2.4.1 Αφήγηση σεναρίου Δημιουργίας ερωτηματολογίου.....	56
2.4.2 Αφήγηση σεναρίου Διαγραφής ερωτηματολογίου.....	57
3 Μη λειτουργικές απαιτήσεις.....	59
3.1 Απαιτήσεις Χρηστικότητας (Usability).....	59
3.2 Τεχνικές Απαιτήσεις περιβάλλοντος.....	59
Παράρτημα Ι - Πίνακας Ιχνηλασιμότητας.....	61
Παράρτημα ΙΙ - Ανοιχτά Θέματα.....	62

## Λίστα Σχημάτων - Διαγραμμάτων

Εικόνα 1: Διάγραμμα κλάσεων πακέτου UserHandle.....	11
Εικόνα 2: Διάγραμμα κλάσεων πακέτου DataAnalysis.....	21



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπο-  
λογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

---

Εικόνα 3: Διάγραμμα κλάσεων πακέτου DiagramHandle.....	31
Εικόνα 4: Διάγραμμα κλάσεων του πακέτου QuestionnaireHandle.....	40
Εικόνα 5: Διάγραμμα κλάσεων του πακέτου DataHandle.....	49
Εικόνα 6: Διάγραμμα ακολουθιών σεναρίου δημιουργίας ρυθμίσεων.....	51
Εικόνα 7: Διάγραμμα ακολουθιών σεναρίου διαγραφής ρυθμίσεων.....	52
Εικόνα 8: Διάγραμμα ακολουθιών σεναρίου δημιουργίας διαγράμματος.....	54
Εικόνα 9: Διάγραμμα ακολουθιών σεναρίου αναζήτησης δεδομένων.....	55
Εικόνα 10: Διάγραμμα ακολουθιών σεναρίου αποθήκευσης δεδομένων.....	56
Εικόνα 11: Διάγραμμα ακολουθιών σεναρίου δημιουργίας ερωτηματολογίου.....	57
Εικόνα 12: Διάγραμμα ακολουθιών σεναρίου διαγραφής ερωτηματολογίου.....	58

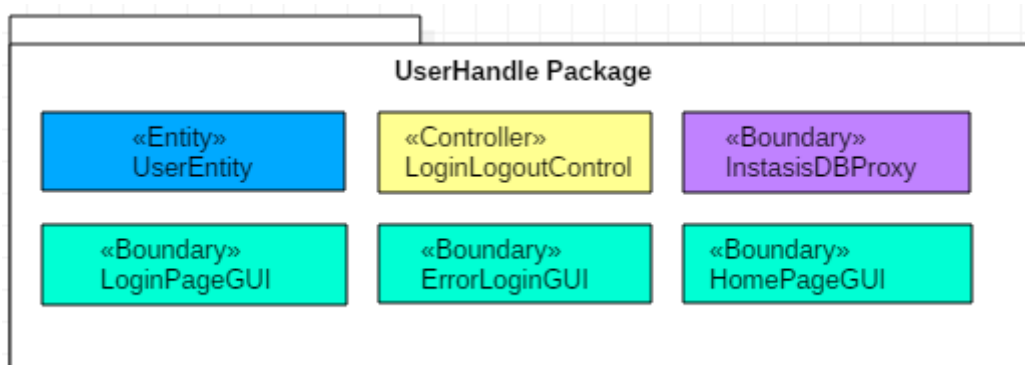


# 1 Στατική Μοντελοποίηση

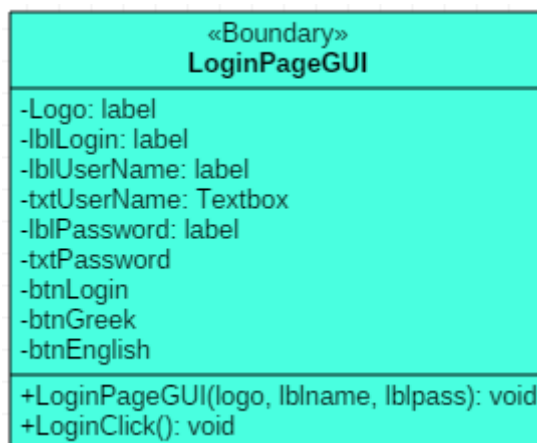
## 1.1 Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας

### 1.1.1 Πακέτο UserHandle

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



### LoginPageGUI



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

Χαρακτηριστικά της κλάσης:

- Logo: Λογότυπο της εφαρμογής .
- lblLogin: Η επιγραφή "login".



- lblUserName: Επιγραφή πάνω από το πεδίο συμπλήρωσης ονόματος χρήστη.
- txtUserName: Πεδίο συμπλήρωσης για το όνομα του πανεπιστημιακού λογαριασμού του χρήστη.
- lblPassword: Επιγραφή πάνω από το πεδίο συμπλήρωσης κωδικού πρόσβασης του χρήστη.
- txtPasssword: Πεδίο συμπλήρωσης για τον κωδικό πρόσβασης.
- btnLogin: Κουμπί για την είσοδο στο σύστημα της εφαρμογής.
- btnGreek: Κουμπί για την ρύθμιση της εμφάνισης όλων των επιλογών και των διαγραμμάτων στην ελληνική γλώσσα.
- btnEnglish: Κουμπί για την ρύθμιση της εμφάνισης όλων των επιλογών και των διαγραμμάτων στην αγγλική γλώσσα.

Μέθοδοι της κλάσης:

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

### ErrorLoginGUI

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

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

Χαρακτηριστικά της κλάσης:

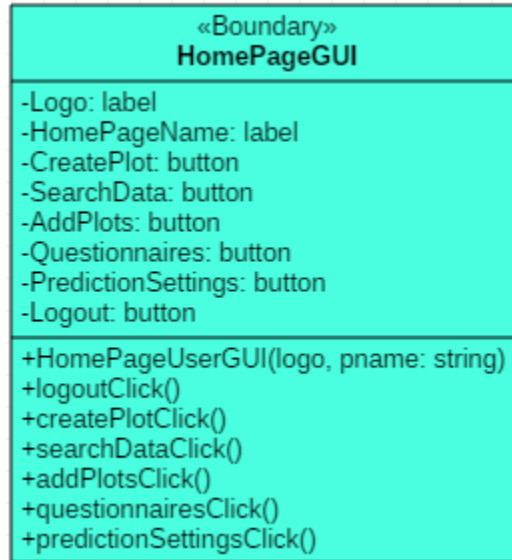
- Error: Το μήνυμα σφάλματος που προβάλλεται στο χρήστη.
- Back: Κουμπί για την επιστροφή στη σελίδα σύνδεσης.

Μέθοδοι της κλάσης:

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



### HomepageGUI



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

Χαρακτηριστικά της κλάσης:

- Logo: Λογότυπο της εφαρμογής.
- HomePageName: Ονομασία της αρχικής σελίδας.
- CreatePlot: Κουμπί για τη δημιουργία νέου διαγράμματος.
- SearchData: Κουμπί για την αναζήτηση δεδομένων.
- AddPlots: Κουμπί για τη προσθήκη διαγραμμάτων.
- Questionnaires: Κουμπί για τη δημιουργία και διαχείριση των ερωτηματολογίων.
- PredictionSettings: Κουμπί για τη ρύθμιση και τη τροποποίηση των ρυθμίσεων προβλέψεων.
- Logout: Κουμπί για την αποσύνδεση του χρήστη.

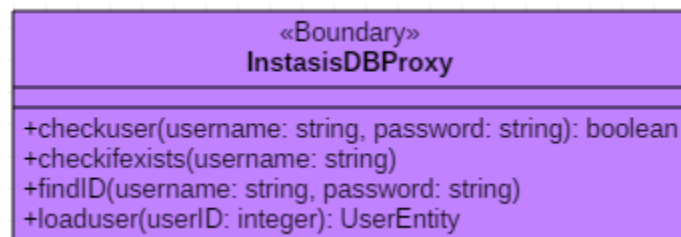
Μέθοδοι της κλάσης:

- HomePageUserGUI(logo,pname:string): Η μέθοδος αυτή είναι η συνάρτηση δόμησης της διεπαφής.
- logoutClick(): Καλεί τη συνάρτηση logout() του ελεγκτή LoginLogoutControl ο οποίος θα αποσυνδέσει το χρήστη.



- `createPlotClick()`: Καλεί τη συνάρτηση `createDiagram()` του ελεγκτή `DiagramHandlerController` από το πακέτο `DiagramHandlerPackage` ο οποίος είναι υπεύθυνος για τη δημιουργία του διαγράμματος.
- `searchDataClick()`: Καλεί τη συνάρτηση `searchData()` του ελεγκτή `DataHandlerController` από το πακέτο `DataHandlerPackage` για την αναζήτηση δεδομένων.
- `addPlotsClick()`: Καλεί τη συνάρτηση του ελεγκτή `uploadDiagram()` του ελεγκτή `DiagramHandlerController` από το πακέτο `DiagramHandlerPackage` ο οποίος είναι υπεύθυνος για το ανέβασμα(upload) διαγραμμάτων.
- `questionnairesClick()`: Η μέθοδος αυτή καλεί τη συνάρτηση `displayQuestionnairePageGUI()` του ελεγκτή `QuestionnaireHandlerControl` από το πακέτο `Questionnaire HandlerPackage` για τη δημιουργία ερωτηματολογίων.
- `predictionSettingsClick()`: Η μέθοδος αυτή καλεί τη συνάρτηση `displayPublishOption()` του ελεγκτή `PublishOptionController` από το πακέτο `DataAnalysisPackage` η οποία χρησιμεύει για τον καθορισμό των ρυθμίσεων προβλέψεων.

### InstasisDBProxy



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

Μέθοδοι της κλάσης:

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





## **Τεχνολογία Λογισμικού**

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπο-  
λογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017



### UserEntity

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

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

Χαρακτηριστικά της κλάσης:

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

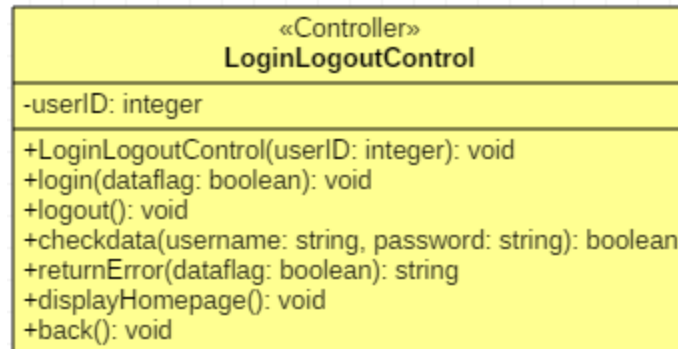
Μέθοδοι της κλάσης:

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



---

### LoginLogoutControl



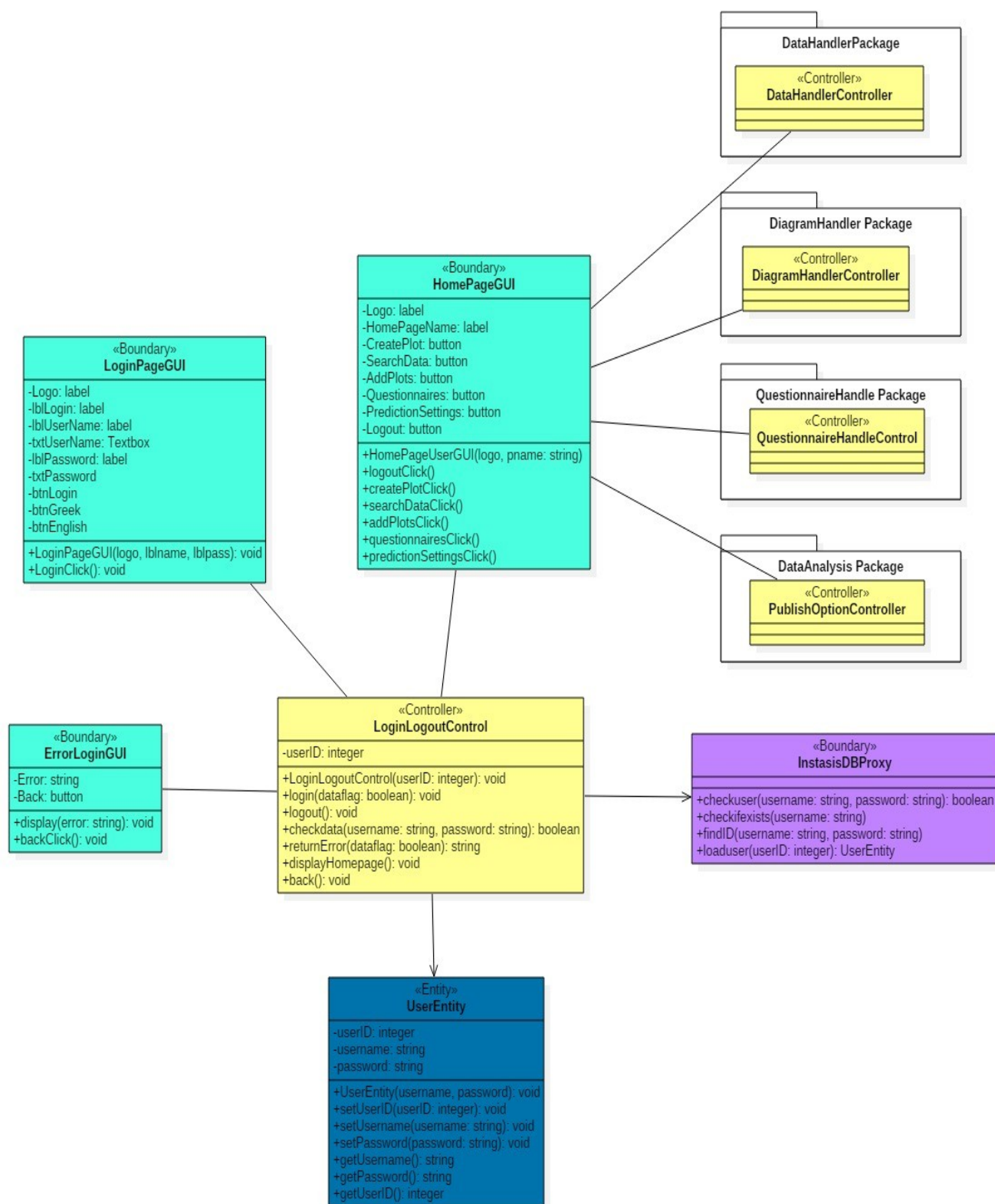
Ο ελεγκτής αυτός περιέχει όλες τις κατάλληλες συναρτήσεις για την είσοδο του χρήστη στο σύστημα.

Χαρακτηριστικά της κλάσης:

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

Μέθοδοι της κλάσης:

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

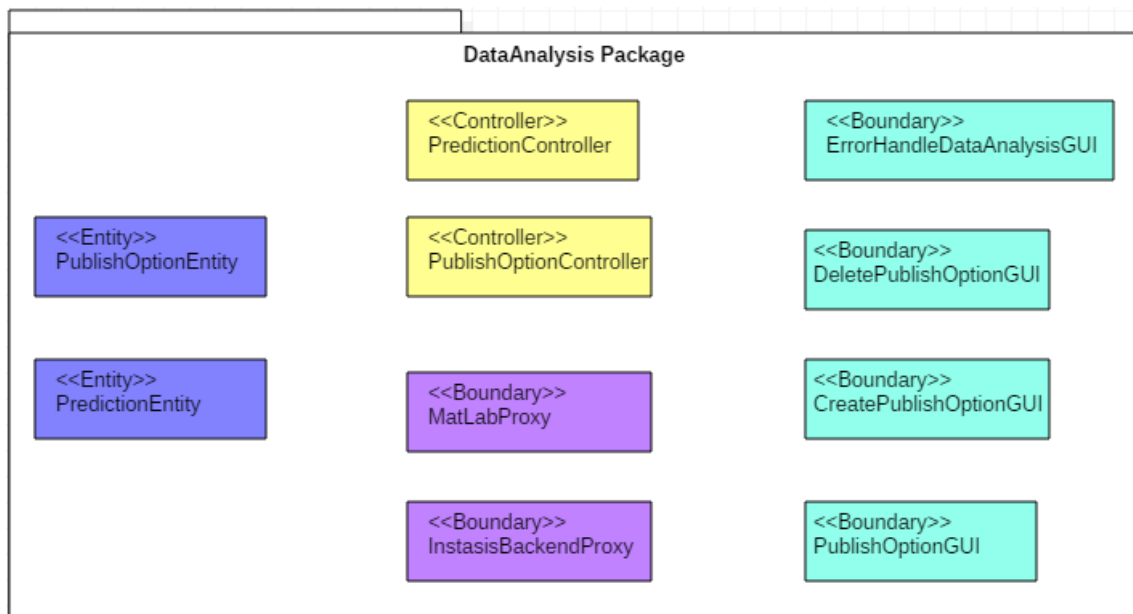


Εικόνα 1: Διάγραμμα κλάσεων πακέτου UserHandle

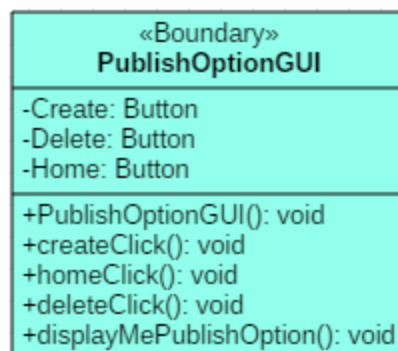


### 1.1.2 Πακέτο DataAnalysis

Το πακέτο αυτό περιλαμβάνει ότι έχει να κάνει με τις προβλέψεις που γίνονται στην εφαρμογή. Περιλαμβάνει, δηλαδή, τις κλάσεις που έχουν να κάνουν με την πρόβλεψη του χρόνου απάντησης και της πιθανότητας αποδοχής κάποιου αιτήματος για ραντεβού, με την πρόταση μιας άλλης κοντινής μέρας και ώρας για την βελτίωση του αιτήματος καθώς και με την έκδοση και ρύθμιση διαγραμμάτων πρόβλεψης απασχόλησης της γραμματείας.



#### PublishOptionGUI:



Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση της σελίδας Ρύθμιση Προβλέψεων.



Χαρακτηριστικά της κλάσης:

- Create: Κουμπί για την δημιουργία Νέας Ρύθμισης Πρόβλεψης.
- Delete: Κουμπί για την διαγραφή κάποιας Ρύθμισης Πρόβλεψης.
- Home: Κουμπί για την επιστροφή στην αρχική σελίδα.

Μέθοδοι της κλάσης:

- PublishOptionGUI(): Είναι η συνάρτηση δόμησης της οριακής κλάσης.
- createClick(): Η μέθοδος αυτή μεταβαίνει στη σελίδα δημιουργίας Ρύθμισης Πρόβλεψης μέσω της συνάρτησης displayCreatePublishOption() ελεγκτή PublishOptionController.
- homeClick(): Καλεί τη συνάρτηση displayHome() του ελεγκτή HomepageController για προβολή της αρχικής σελίδας της εφαρμογής.
- deleteClick(): Η μέθοδος αυτή μεταβαίνει στη σελίδα διαγραφής Ρύθμισης Πρόβλεψης DeletePublishOptionGUI, καλώντας την displayDeletePublishOption() του ελεγκτή PublishOptionController.

### DeletePublishOptionGUI:

«Boundary» DeletePublishOptionGUI
-Confirm: Button -Cancel: Button -OptionList: Listbox
+DeletePublishOptionGUI(optionList): void +cancelClick(): void +confirmClick(): void +displayMeDeleteOption(): void

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση της σελίδας διαγραφής μιας Ρύθμισης Πρόβλεψης.

Χαρακτηριστικά της κλάσης:

- Confirm: Κουμπί για την επιβεβαίωση της Διαγραφής της Ρύθμισης Πρόβλεψης.
- Cancel: Κουμπί για την ακύρωση της διαγραφής κάποιας Ρύθμισης Πρόβλεψης.
- OptionList: Λίστα με τις διαθέσιμες Ρυθμίσεις Πρόβλεψης για επιλογή μίας.

Μέθοδοι της κλάσης:

- DeletePublishOptionGUI(optionList): Είναι η συνάρτηση δόμησης της οριακής κλάσης.



- `cancelClick()`: Ακυρώνει την διαδικασία και καλεί τη συνάρτηση `displayPublishOption()` του ελεγκτή `PublishOptionController` για προβολή της σελίδας Ρύθμιση Προβλέψεων.
- `confirmClick()`: Η μέθοδος αυτή για να διαγράψει την επιλεγμένη ρύθμιση καλεί την `deleteOption(OptionID)` του ελεγκτή `PublishOptionController`.
- `displayMeCreatePublishOption()`: Η μέθοδος αυτή εμφανίζει την διεπαφή της κλάσης στην οθόνη.

### CreatePublishOptionGUI:

«Boundary» CreatePublishOptionGUI
-Timescale: Listbox -Home: Button -Cancel: Button -StartTime: Textbox -StopTime: Textbox -Period: Listbox -Confirm: Button -Name: Textbox
+CreatePublishOptionGUI(scale, start, stop, period, name): void +cancelClick(): void +homeClick(): void +confirmClick(): void +displayMeCreatePublishOption(): void

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση της σελίδας δημιουργίας μιας Ρύθμισης Πρόβλεψης.

Χαρακτηριστικά της κλάσης:

- Timescale: Λίστα με τις επιλογές για την κλίμακα χρόνου.
- Home: Κουμπί για την επιστροφή στην αρχική σελίδα.
- Cancel: Κουμπί για την ακύρωση της δημιουργίας κάποιας Ρύθμισης Πρόβλεψης.
- StartTime: Πεδίο συμπλήρωσης για την απόσταση πρόβλεψης.
- StopTime: Πεδίο συμπλήρωσης για το πλήθος πρόβλεψης.
- Period: Λίστα με τις επιλογές για την περιοδικότητα.
- Confirm: Κουμπί για την επιβεβαίωση δημιουργία Νέας Ρύθμισης Πρόβλεψης.
- Name: Πεδίο συμπλήρωσης για το όνομα Ρύθμισης Πρόβλεψης.

Μέθοδοι της κλάσης:

- `CreatePublishOptionGUI(scale, start, stop, period, name)`: Είναι η συνάρτηση δόμησης της οριακής κλάσης.



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

- `cancelClick()`: Ακυρώνει την διαδικασία και καλεί τη συνάρτηση `displayPublishOption()` του ελεγκτή `PublishOptionController` για προβολή της σελίδας Ρύθμιση Προβλέψεων.
- `homeClick()`: Καλεί τη συνάρτηση `displayHome()` του ελεγκτή `HomepageController` για προβολή της αρχικής σελίδας της εφαρμογής.
- `confirmClick()`: Η μέθοδος αυτή για να δημιουργήσει την νέα ρύθμιση καλεί την `createOption(OptionID, scale, start, stop, period, odate, name)` του ελεγκτή `PublishOptionController`.
- `DisplayMeCreatePublish()`: Η μέθοδος αυτή εμφανίζει την διεπαφή της κλάσης στην οθόνη.

### ErrorHandleDataAnalysisGUI:

«Boundary»
<b>ErrorHandleDataAnalysisGUI</b>
-Back: Button
-Error: string
+backClick(): void
+showError(Error): void

Η κλάση αυτή αποτελεί το γραφικό περιβάλλον διεπαφής για την εμφάνιση μηνύματος σφάλματος προς το χρήστη.

Χαρακτηριστικά της κλάσης:

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

Μέθοδοι της κλάσης:

- `backClick()`: Η μέθοδος αυτή καλεί την κατάλληλη συνάρτηση του ελεγκτή `PublishOptionController` για επιστροφή στην κατάλληλη σελίδα.
- `showError(error: string)`: Εμφανίζει το μήνυμα σφάλματος που της παρέχεται από τη συνάρτηση `returnOptionError()` του ελεγκτή `PublishOptionController`.





**PublishOptionEntity:**

«Entity» PublishOptionEntity
-Timescale: integer -StartTime: integer -StopTime: integer -Period: integer -OptionDate: string -OptionID: integer -Name: string
+PublishOptionEntity(scale, start, stop, period, odate, id, name): void +getTimescale(): void +getStartTime(): void +getStopTime(): void +getPeriod(): void +getOptionDate(): void +getOptionID(): void +getName(): void +setTimescale(Timescale): void +setStartTime(StartTime): void +setStopTime(StopTime): void +setPeriod(Period): void +setOptionDate(OptionDate): void +setOptionID(OptionID): void +setName(Name): void

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

Χαρακτηριστικά της κλάσης:

- Timescale: Ο κωδικός της κλίμακα χρόνου της Ρύθμισης Πρόβλεψης.
- StartTime: Η απόσταση πρόβλεψης της Ρύθμισης Πρόβλεψης.
- StopTime: Το πλήθος πρόβλεψης της Ρύθμισης Πρόβλεψης.
- Period: Η περιοδικότητα της Ρύθμισης Πρόβλεψης.
- OptionDate: Η ημερομηνία της Ρύθμισης Πρόβλεψης.
- OptionID: Ο μοναδικός κωδικός που καθορίζει την Ρύθμιση Πρόβλεψης.
- Name: String για το όνομα της Ρύθμισης Πρόβλεψης.

Μέθοδοι της κλάσης:

- PublishOptionEntity(scale, start, stop, period, odate, name): Είναι η συνάρτηση δόμησης της κλάσης.
- getTimescale(): Μέθοδος που επιστρέφει τον κωδικό της κλίμακας χρόνου.
- getStartTime(): Μέθοδος που επιστρέφει την απόσταση πρόβλεψης.
- getStopTime(): Μέθοδος που επιστρέφει το πλήθος.
- getPeriod(): Μέθοδος που επιστρέφει την περιοδικότητα.
- getOptionDate(): Μέθοδος που επιστρέφει την ημερομηνία.



- `getOptionID()`: Μέθοδος που επιστρέφει τον κωδικό ρύθμισης.
- `getName()`: Μέθοδος που επιστρέφει το όνομα.
- `setTimescale(Timescale)`: Μέθοδος που θέτει τον κωδικό της κλίμακας χρόνου.
- `setStartTime(StartTime)`: Μέθοδος που θέτει την απόσταση.
- `setStopTime(StopTime)`: Μέθοδος που θέτει το πλήθος.
- `setPeriod(Period)`: Μέθοδος που θέτει την περιοδικότητα.
- `setOptionDate(OptionDate)`: Μέθοδος που θέτει την ημερομηνία.
- `setOptionID(OptionID)`: Μέθοδος που θέτει τον κωδικό ρύθμισης.
- `setName(Name)`: Μέθοδος που θέτει το όνομα.

**PredictionEntity:**

«Entity» PredictionEntity
-WaitTime: float -Probability: float -OfferDate: string -OfferTimeslot: integer -PredictionID: integer
+PredictionEntity(wait, prob, odate, otime, id): void +getWaitTime(): void +getProbability(): void +getOfferDate(): void +getOfferTimeslot(): void +getPredictionID(): void +setWaitTime(WaitTime): void +setProbability(Probability): void +setOfferDate(OfferDate): void +setOfferTimeslot(OfferTimeslot): void +setPredictionID(PredictionID): void

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

Χαρακτηριστικά της κλάσης:

- `WaitTime`: Εκτίμηση χρόνου απάντησης στο αίτημα.
- `Probability`: Εκτίμηση πιθανότητας αποδοχής στο αίτημα.
- `OfferDate`: Προτεινόμενη ημερομηνία για το αίτημα.
- `OfferTimeslot`: Προτεινόμενη ώρα για το αίτημα.



- PredictionID: Μοναδικός κωδικός πρόβλεψης αιτήματος.

Μέθοδοι της κλάσης:

- PredictionEntity(wait, prob, odate, otime, id): Είναι η συνάρτηση δόμησης της κλάσης.
- getWaitTime(): Μέθοδος που επιστρέφει τον χρόνο απάντησης.
- getProbability(): Μέθοδος που επιστρέφει την πιθανότητα αποδοχής.
- getOfferDate(): Μέθοδος που επιστρέφει την προτεινόμενη ημερομηνία.
- getOfferTimeslot(): Μέθοδος που επιστρέφει την προτεινόμενη ώρα.
- getPredictionID(): Μέθοδος που επιστρέφει τον κωδικό πρόβλεψης.
- setWaitTime(WaitTime): Μέθοδος που θέτει τον χρόνο απάντησης.
- setProbability(Probability): Μέθοδος που θέτει την πιθανότητα αποδοχής.
- setOfferDate(OfferDate): Μέθοδος που θέτει την προτεινόμενη ημερομηνία.
- setOfferTimeslot(OfferTimeslot): Μέθοδος που θέτει την προτεινόμενη ώρα.
- setPredictionID(PredictionID): Μέθοδος που θέτει τον κωδικό πρόβλεψης.

### PredictionController:

«Controller» PredictionController	
-PredictionID: integer	
+PredictionController(PredictionID): void +predictionDataMatLab(PredictionOn: Request, PredictionID: integer): PredictionEntity +returnPredictionError(string): void	

Ο ελεγκτής αυτός χρησιμοποιείται για τον έλεγχο των προβλέψεων για κάποιο αίτημα.

Χαρακτηριστικά της κλάσης:

- PredictionID: Μοναδικός κωδικός πρόβλεψης αιτήματος.

Μέθοδοι της κλάσης:

- PredictionController(PredictionID): Είναι η συνάρτηση δόμησης της κλάσης.
- predictionDataMatLab(PredictionOn:Request, PredictionID:integer): Καλεί την συνάρτηση predictionData(PredictionOn:Request) της κλάσης MatLab-Proxy για να στείλει τα δεδομένα του αιτήματος στο MatLab, και να επιστρέψει τις προβλέψεις. Επιστρέφει στο InstasisBackendProxy αυτές τις προβλέψεις, ή καλεί την returnPredictionError() στην περίπτωση σφάλματος.



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπο-  
λογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

- 
- `returnPredictionError(string)`: Καλεί την `returnError` της κλάσης `Instasis-BackendProxy`.

**PublishOptionController:**

«Controller» PublishOptionController
-OptionID: integer
+PublishOptionController(OptionalID): void +deleteOption(OptionalID): void +publishDiagramFromMatLab(PublishThis: PublishOptionEntity): Image +retrieveOptionNamesList(): Listbox +sendOptionToInstasis(PublishOption: PublishOptionEntity): void +createPublishOption(scale, start, stop, period, odate, id, name): void +displayCreatePublishOption(): void +displayDeletePublishOption(): void +displayPublishOption(): void +displayError(Error: string): void +returnOptionError(string): void

Ο ελεγκτής αυτός χρησιμοποιείται για τον έλεγχο των ρυθμίσεων και εκδόσεων διαγραμμάτων πρόβλεψης της γραμματείας.

Χαρακτηριστικά της κλάσης:

- OptionID: Ο μοναδικός κωδικός που καθορίζει την Ρύθμιση Πρόβλεψης.

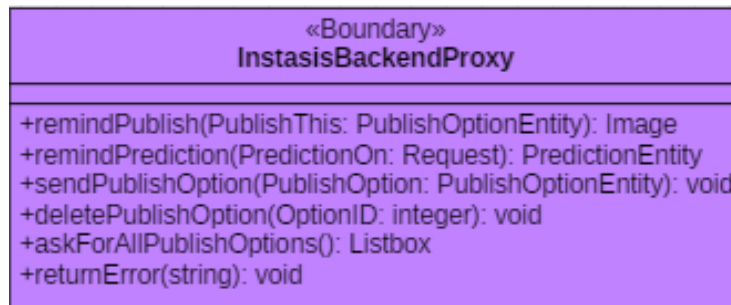
Μέθοδοι της κλάσης:

- PublishOptionController(OptionalID): Είναι η συνάρτηση δόμησης της κλάσης.
- deleteOption(OptionalID): Καλεί την deletePublishOption(OptionalID:integer) της κλάσης InstasisBackendProxy για να διαγραφεί η ρύθμιση πρόβλεψης απασχόλησης, της οποίας τον κωδικό έχει λάβει ως όρισμα.
- publishDiagramFromMatLab(PublishThis:PublishOptionEntity): Καλεί την predictionDiagram(PublishOption:PublishOptionEntity) της κλάσης MatLabProxy για να στείλει τις ρυθμίσεις του διαγράμματος πρόβλεψης και να λάβει και να επιστρέψει το διάγραμμα αυτό. Σε περίπτωση που λάβει μήνυμα σφάλματος καλεί την returnOptionError() για να επιστρέψει το μήνυμα σφάλματος.
- retrieveOptionNamesList(): Καλεί την συνάρτηση askForAllPublishOptions() της κλάσης InstasisBackendProxy και λαμβάνει (μέσω αυτής) και επιστρέφει την λίστα με τις διαθέσιμες ρυθμίσεις διαγραμμάτων πρόβλεψης, για να εμφανιστούν αυτές στην διεπαφή DeletePublishOption.
- sendOptionToInstasis(PublishOption:PublishOptionEntity): Καλεί την συνάρτηση sendPublishOption(PublishOption:PublishOptionEntity) της κλάσης InstasisBackendProxy για να σταλεί μια νέα ρύθμιση διαγραμμάτων πρόβλεψης που πήρε ως όρισμα.
- createPublishOption(scale, start, stop, period, odate, id, name): Δημιουργεί ένα καινούριο αντικείμενο της κλάσης οντότητας PublishOptionEntity καλώντας την συνάρτηση δόμησης της κλάσης. Εν συνεχεία καλεί την sendOptionToInstasis(PublishOption:PublishOptionEntity) με όρισμα το νέο αντικείμενο για να σταλεί αυτό στο InstasisBackend.



- `displayCreatePublishOption()`: Η συνάρτηση αυτή καλεί την `displayMeCreatePublishOption` από το γραφικό περιβάλλον διεπαφής `CreatePublishOptionGUI` για να εμφανιστεί αυτό.
- `displayDeletePublishOption()`: Η συνάρτηση αυτή καλεί την `displayMeDeletePublishOption` από το γραφικό περιβάλλον διεπαφής `DeletePublishOptionGUI` για να εμφανιστεί αυτό.
- `displayPublishOption()`: Η συνάρτηση αυτή καλεί την `displayMePublishOption` από το γραφικό περιβάλλον διεπαφής `PublishOptionGUI` για να εμφανιστεί αυτό.
- `displayError(Error: string)`: Καλεί την `showError(Error)` από την κλάση διεπαφής `ErrorHandleDataAnalysisGUI` για να εμφανίσει ένα μήνυμα σφάλματος στον χρήστη.
- `returnOptionError(string)`: Η συνάρτηση αυτή καλείται όταν ο ελεγκτής λάβει κάποιο σφάλμα. Καλεί την `returnError(string)` για να περάσει ως όρισμα σε αυτήν το μήνυμα σφάλματος.

### InstasisBackendProxy:



Η κλάση αυτή χρησιμοποιείται για επικοινωνία με το `InstasisBackend`. Στέλνει και λαμβάνει μηνύματα και αιτήματα μέσω των συναρτήσεων της.

Μέθοδοι της κλάσης:

- `remindPublish(PublishThis:PublishOptionEntity)`: Καλείται μέσω του `InstasisBackend` και αποτελεί ένα αίτημα για δημιουργία Διαγράμματος πρόβλεψης απασχόλησης. Τα στοιχεία του διαγράμματος δίνονται σαν όρισμα. Με την σειρά της καλεί την `publishDiagramFromMatLab(PublishThis:PublishOptionEntity)` του ελεγκτή `PublishOptionController` και στο τέλος επιστρέφει το διάγραμμα που ζητήθηκε.
- `remindPrediction(PredictionOn:Request)`: Καλείται μέσω του `InstasisBackend` και αποτελεί ένα αίτημα για δημιουργία πρόβλεψης πάνω σε κάποιο αίτημα. Τα στοιχεία του αιτήματος δίνονται σαν όρισμα. Με την σειρά της καλεί την `predictionDataMatLab(PredictionOn:Request, PredictionID:integer)` του ελεγκτή `PredictionController` και στο τέλος επιστρέφει το διάγραμμα που ζητήθηκε.
- `sendPublishOption(PublishOption:PublishOptionEntity)`: Καλείται για να



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

στείλει στο InstasisBackend μια νέα ρύθμιση διαγράμματος πρόβλεψης που δημιουργήθηκε ώστε να αποθηκευτεί. Την νέα αυτή ρύθμιση τη λαμβάνει ως όρισμα.

- `deletePublishOption(OptionID:integer)`: Καλείται για να στείλει στο InstasisBackend ένα αίτημα διαγραφής μιας ρύθμιση διαγράμματος πρόβλεψης που προσδιορίζεται μέσω του μοναδικού κωδικού της. Τον κωδικό αυτό τον λαμβάνει ως όρισμα.
- `askForAllPublishOptions()`: Καλείται για να λάβει απο InstasisBackend την λίστα με τις αποθηκευμένες ρυθμίσεις διαγραμμάτων απασχόλησης.
- `returnError(string)`:

### MatLabProxy:



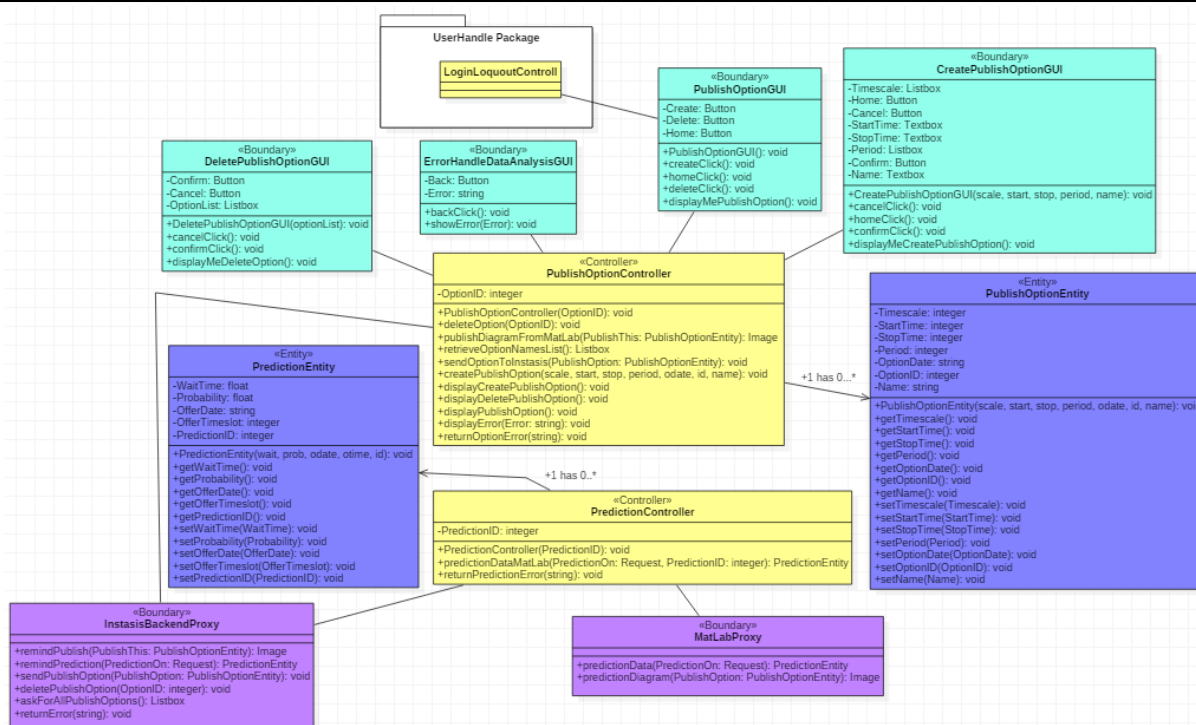
Η κλάση αυτή χρησιμοποιείται για επικοινωνία με το MatLab. Στέλνει και λαμβάνει μηνύματα και αιτήματα μέσω των συναρτήσεων της.

Μέθοδοι της κλάσης:

- `predictionData(PredictionOn:Request)`: Στέλνει στο MatLab τα στοιχεία ενός αιτήματος και επιστρέφει πίσω ένα αντικείμενο πρόβλεψης πάνω σε αυτό το αίτημα. Σε περίπτωση σφάλματος επιστρέφει το σφάλμα.
- `predictionDiagram(PublishOption:PublishOptionEntity)`: Στέλνει στο MatLab τα στοιχεία ενός διαγράμματος και επιστρέφει πίσω το διάγραμμα σε εικόνα. Σε περίπτωση σφάλματος επιστρέφει το σφάλμα.

### Διάγραμμα κλάσεων:



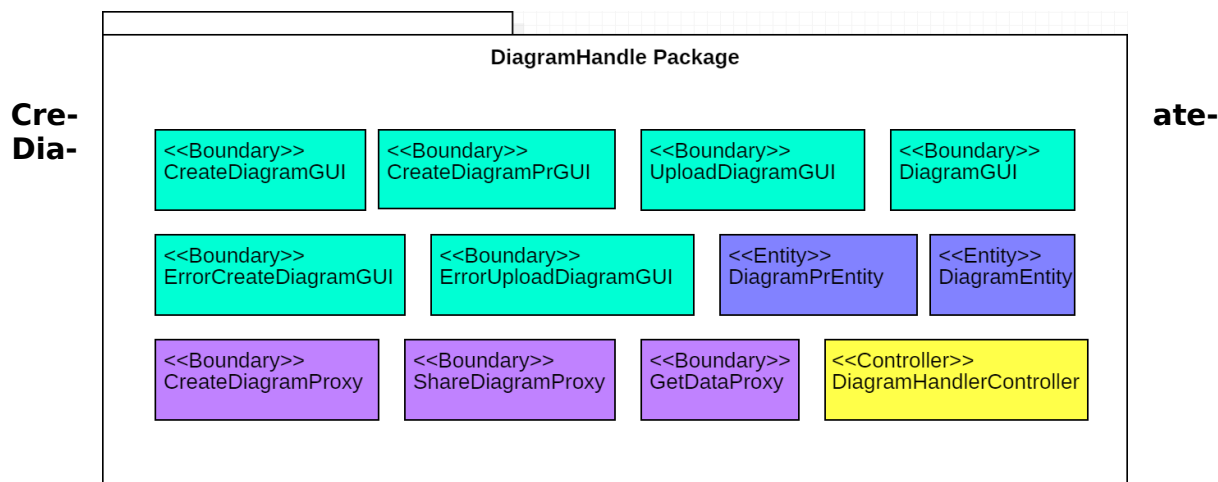


Εικόνα 2: Διάγραμμα κλάσεων πακέτου DataAnalysis

## 1.2 Πακέτα λεξιλογίου σεναρίων μέσης προτεραιότητας

### 1.2.1 Πακέτο DiagramHandle

Το πακέτο αυτό αναφέρεται στις επιλογές δημιουργίας ενός διαγράμματος και αποστολής του στο σύστημα instasis-backend. Ακόμη στο πακέτο αυτό περιλαμβάνεται και η λειτουργία επισύναψης και αποστολής ενός διαγράμματος από τον προσωπικό υπολογιστή του χρήστη στο σύστημα instasis-backend.







---

**gramGUI:**

«Boundary» <b>CreateDiagramGUI</b>
-StartDate: TextBox -EndDate: TextBox -ApplicationType: ListBox -Next: Button
+CreateDiagramGUI(): void +nextClick(): void +display(): void

Η κλάση αυτή αποτελεί την πρώτη διεπαφή-φόρμα που πρέπει να συμπληρώσει ο χρήστης για την δημιουργία διαγράμματος.

Χαρακτηριστικά κλάσης:

- StartDate: Πεδίο εισαγωγής της ημερομηνίας από την οποία θα ξεκινήσει η ανάλυση δεδομένων για το διάγραμμα.
- EndDate: Πεδίο εισαγωγής της ημερομηνίας μέχρι την οποία θα γίνει η ανάλυση δεδομένων για το διάγραμμα.
- ApplicationType: Λίστα επιλογής του τύπου δεδομένων που θα αναλυθούν.
- Next: Κουμπί για το επόμενο στάδιο συμπλήρωσης των παραμέτρων του διαγράμματος.

Συναρτήσεις της κλάσης:

- CreateDiagramGUI(): Συνάρτηση δόμησης της κλάσης.
- nextClick(): Η μέθοδος αυτή επιστρέφει τον έλεγχο στον DiagramHandler-Controller για άνοιγμα της επόμενης φόρμας.
- display(): Η μέθοδος αυτή προβάλλει την εν λόγω διεπαφή.

**CreateDiagramPrGUI:**

«Boundary» <b>CreateDiagramPrGUI</b>
-yAxis: ListBox -xAxis: ListBox -DiagramType: ListBox -Cancel: Button -Diagram: Button
+CreateDiagramPrGUI(): void +cancelClick(): void +diagramClick(): void +display(): void

Η κλάση αυτή αποτελεί την πρώτη διεπαφή-φόρμα που πρέπει να συμπληρώσει ο χρήστης για την δημιουργία διαγράμματος.



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπο-  
λογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

---

### Χαρακτηριστικά κλάσης:

- yAxis: Λίστα επιλογής της μεταβλητής που θα αναλυθεί στο διάγραμμα.
- xAxis: Λίστα επιλογής της μεταβλητής ως προς την οποία θα γίνει η ανάλυση.
- DiagramType: Λίστα επιλογής του τύπου του διαγράμματος.
- Cancel: Κουμπί επιστροφής στην αρχική σελίδα.
- Diagram: Κουμπί δημιουργίας του διαγράμματος.

### Συναρτήσεις κλάσης:

- CreateDiagramPrGUI(): Συνάρτηση δόμησης της κλάσης.
- cancelClick(): Η μέθοδος αυτή καλεί τη συνάρτηση displayHomepage() του controller LoginLogoutControl.
- diagramClick(): Η μέθοδος αυτή καλεί την συνάρτηση checkEnteredData() του DiagramHandlerController.
- display(): Η μέθοδος αυτή προβάλλει την εν λόγω διεπαφή.



---

**DiagramGUI:**

«Boundary» <b>DiagramGUI</b>
-Diagram: Image -Cancel: Button -Share: Button
+DiagramGUI(diagram: Image): void +cancelClick(): void +shareClick(): void +display(): void

Η κλάση αυτή αποτελεί την διεπαφή για την προβολή του διαγράμματος.

Χαρακτηριστικά κλάσης:

- Diagram: Εικόνα με το προκύπτον διάγραμμα.
- Cancel: Κουμπί επιστροφής στην αρχική σελίδα.
- Share: Κουμπί αποστολής του διαγράμματος στο σύστημα instasis-back-end.

Συναρτήσεις κλάσης:

- DiagramGUI(diagram: Image): Συνάρτηση δόμησης της κλάσης.
- cancelClick(): Η μέθοδος αυτή καλεί τη συνάρτηση displayHomepage() του controller LoginLogoutControl.
- shareClick(): Η μέθοδος αυτή καλεί την συνάρτηση shareDiagram(diagram: Image) του DiagramHandlerController.
- display(): Η μέθοδος αυτή προβάλλει την εν λόγω διεπαφή.

**UploadDiagramGUI:**

«Boundary» <b>UploadDiagramGUI</b>
+Location: FileDialog +Cancel: Button +Upload: Button
+UploadDiagramGUI(): void +cancelClick(): void +uploadClick(): File +display(): void

Η κλάση αυτή αποτελεί την διεπαφή για το «ανέβασμα» (upload) ενός διαγράμματος από τον προσωπικό υπολογιστή του χρήστη.

Χαρακτηριστικά κλάσης:

- Location: Παράθυρο επιλογής διαγράμματος από τον προσωπικό υπολογιστή του χρήστη.



- Cancel: Κουμπί επιστροφής στην αρχική σελίδα.
- Upload: Κουμπί αποστολής του διαγράμματος στο σύστημα instasis-backend.

Συναρτήσεις κλάσης:

- UploadDiagramGUI(): Συνάρτηση δόμησης της κλάσης.
- cancelClick(): Η μέθοδος αυτή καλεί τη συνάρτηση displayHomepage() του controller LoginLogoutControl.
- uploadClick(): Η μέθοδος αυτή καλεί την συνάρτηση checkCompatibility(File) του DiagramHandlerController.
- display(): Η μέθοδος αυτή προβάλλει την εν λόγω διεπαφή.

### ErrorCreateDiagramGUI:

«Boundary» ErrorCreateDiagramGUI
-Error: String -Flag: Integer -Back: Button
+ErrorCreateDiagramGUI(error: String): void +backClick(): void +display(error: String): void

Η κλάση αυτή αποτελεί τη διεπαφή για εμφάνιση μηνύματος σφάλματος σε περίπτωση που δεν έχει συμπληρωθεί κάποιο πεδίο κατά την δημιουργία διαγράμματος.

Χαρακτηριστικά κλάσης:

- Error: Μήνυμα σφάλματος για ελλιπή συμπλήρωση πεδίων φόρμας ή για μη ύπαρξη των δεδομένων που ζήτησε ο χρήστης.
- Flag: Αναγνωριστικό της φόρμας στην οποία δεν συμπληρώθηκαν όλα τα πεδία.
- Back: Κουμπί για κλείσιμο της εν λόγω διεπαφής.

Συναρτήσεις κλάσης:

- ErrorCreateDiagramGUI(error: String): Συνάρτηση δόμησης της κλάσης.
- backClick(): Η μέθοδος αυτή κλείνει την εν λόγω διεπαφή και επιστρέφει τον έλεγχο στον DiagramHandlerController
- display(): Η μέθοδος αυτή προβάλλει την εν λόγω διεπαφή.



---

**ErrorUploadDiagramGUI:**

«Boundary» ErrorUploadDiagramGUI
-Error: String -Flag: Integer -TryAgain: Button
+ErrorUploadDiagramGUI(error: String): void +tryAgainClick(): void +display(): void

Η κλάση αυτή αποτελεί τη διεπαφή για εμφάνιση μηνύματος σφάλματος σε περίπτωση που κατά το «ανέβασμα» ενός διαγράμματος, το αρχείο που επιλέγεται δεν είναι αρχείο jpeg.

Χαρακτηριστικά κλάσης:

- Error: Μήνυμα σφάλματος επιλογής εικόνας που δεν είναι σε μορφή jpeg.
- Flag: Αναγνωριστικό του σφάλματος.
- Back: Κουμπί για κλείσιμο της εν λόγω διεπαφής.

Συναρτήσεις κλάσης:

- ErrorUploadDiagramGUI(error: String): Συνάρτηση δόμησης της κλάσης.
- tryAgainClick(): Η μέθοδος αυτή κλείνει την εν λόγω διεπαφή και επιστρέφει τον έλεγχο στον DiagramHandlerController.
- display(): Η μέθοδος αυτή προβάλλει την εν λόγω διεπαφή.

**GetDataProxy:**

«Boundary» GetDataProxy
+GetDataProxy(): void +getData(diagram: DiagramPrEntity): DiagramEntity

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με το σύστημα instasis-backend, για τη λήξη των ζητούμενων πληροφοριών.

Συναρτήσεις κλάσης:

- GetDataProxy(): Συνάρτηση δόμησης της κλάσης.
- getData(diagram: DiagramPrEntity): Η συνάρτηση αυτή επικοινωνεί με το σύστημα instasis-backend και επιστρέφει ένα αντικείμενο τύπου DiagramEntity.



---

**CreateDiagramProxy:**

«Boundary» <b>CreateDiagramProxy</b>
+CreateDiagramProxy(): void +create(diagram: DiagramEntity): DiagramEntity

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με το σύστημα matlab, για τη δημιουργία του εκάστοτε διαγράμματος.

Συναρτήσεις κλάσης:

- CreateDiagramProxy(): Συνάρτηση δόμησης της κλάσης.
- create(diagram: DiagramEntity): Η συνάρτηση αυτή επικοινωνεί με το εξωτερικό σύστημα matlab και δημιουργεί το διάγραμμα.

**ShareDiagramProxy:**

«Boundary» <b>ShareDiagramProxy</b>
+ShareDiagramProxy(): void +shareDiagram(diagram: Image): void

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με το σύστημα instasis-backend, για την αποστολή ενός διαγράμματος.

Συναρτήσεις κλάσης:

- ShareDiagramProxy(): Συνάρτηση δόμησης της κλάσης.
- shareDiagram(diagram: Image): Η συνάρτηση αυτή στέλνει το διάγραμμα στο σύστημα instasis-backend.



---

**DiagramPrEntity:**

«Entity» DiagramPrEntity
-StartDate: String -EndDate: String -ApplicationType: String -yAxis: String -xAxis: String -DiagramType: String
+DiagramPrEntity(start, end, applType, yaxis, xaxis, diagType): void +setStartDate(date: String): void +setEndDate(date: String): void +setApplicationType(type: String): void +setYaxis(y: String): void +setXaxis(x: String): void +setDiagramType(type: String): void +getStartDate(): String +getEndDate(): String +getApplicationType(): String +getYaxis(): String +getXaxis(): String +getDiagramType(): String

Η κλάση αυτή είναι κλάση οντότητας και περιέχει τις παραμέτρους του διαγράμματος.

Χαρακτηριστικά κλάσης:

- **StartDate:** Ημερομηνία από την οποία θα αρχίσει η ανάλυση δεδομένων για το διάγραμμα.
- **EndDate:** Ημερομηνία μέχρι την οποία θα γίνει η ανάλυση δεδομένων για το διάγραμμα.
- **ApplicationType:** Τύπος δεδομένων για τα οποία θα γίνει ανάλυση.
- **yAxis:** Η μεταβλητή για την οποία θα γίνει η ανάλυση δεδομένων.
- **xAxis:** Η μεταβλητή ως προς την οποία θα γίνει η ανάλυση δεδομένων.
- **DiagramType:** Ο τύπος του διαγράμματος.

Συναρτήσεις κλάσης:

- **DiagramPrEntity(start,end,applType,yaxis,xaxisdiagType):** Συνάρτηση δόμησης της κλάσης.
- **setStartDate(date: String):** Η μέθοδος αυτή ορίζει την μεταβλητή StartDate.
- **setEndDate(date: String):** Η μέθοδος αυτή ορίζει την μεταβλητή EndDate.
- **setApplicationType(type: String):** Η μέθοδος αυτή ορίζει την μεταβλητή ApplicationType.
- **setYaxis(y: String):** Η μέθοδος αυτή ορίζει την μεταβλητή yAxis.
- **setXaxis(x: String):** Η μέθοδος αυτή ορίζει την μεταβλητή xAxis.
- **setDiagramType(type: String):** Η μέθοδος αυτή ορίζει την μεταβλητή DiagramType.
- **getStartDate():** Η μέθοδος αυτή επιστρέφει την μεταβλητή StartDate.
- **getEndDate():** Η μέθοδος αυτή επιστρέφει την μεταβλητή EndDate.
- **getApplicationType():** Η μέθοδος αυτή επιστρέφει την μεταβλητή ApplicationType.





- getYaxis(): Η μέθοδος αυτή επιστρέφει την μεταβλητή yAxis.
- getXaxis(): Η μέθοδος αυτή επιστρέφει την μεταβλητή xAxis.
- getDiagramType(): Η μέθοδος αυτή επιστρέφει την μεταβλητή Diagram-Type.

**DiagramEntity:**

«Entity» DiagramEntity
+Yvariable[]: Float +Xvariable[]: Float +Diagram: Image +Flag: boolean
+DiagramEntity(Yvar[], Xvar[], flag): void +setYvariable(Yvar[]): void +setXvariable(Xvar[]): void +setDiagram(diagram: Image): void +setFlag(flag): void +getYvariable(): Float +getXvariable(): Float +getDiagram(): Image +getFlag(): boolean

Η κλάση αυτή αποτελεί κλάση οντότητας και περιέχει το προκύπτον διάγραμμα. Χαρακτηριστικά κλάσης:

- Yvariable[]: Πίνακας τιμών της μεταβλητής για την οποί γίνεται ανάλυση δεδομένων.
- Xvariable[]: Πίνακας τιμών της μεταβλητή ως προς την οποία γίνεται ανάλυση δεδομένων.
- Diagram: Εικόνα με το διάγραμμα.
- Flag: Σημαία αναγνώρισης της ύπαρξης των δεδομένων που ζήτησε ο χρήστης.

Συναρτήσεις κλάσης:

- DiagramEntity(Yvar[],Xvar[],flag): Συνάρτηση δόμησης της κλάσης.
- setYvariable(Yvar[]): Η μέθοδος αυτή ορίζει τον πίνακα Yvariable[].
- setXvariable(Xvar[]): Η μέθοδος αυτή ορίζει τον πίνακα Xvariable[].
- setDiagram(diagram: Image): Η μέθοδος αυτή ορίζει την μεταβλητή Diagram.
- setFlag(flag): Η μέθοδος αυτή ορίζει την μεταβλητή Flag.
- getYvariable(): Η μέθοδος αυτή επιστρέφει τον πίνακα Yvariable[].
- getXvariable(): Η μέθοδος αυτή επιστρέφει τον πίνακα Xvariable[].
- getDiagram(): Η μέθοδος αυτή επιστρέφει την μεταβλητή Diagram.
- getFlag(): Η μέθοδος αυτή επιστρέφει την μεταβλητή Flag.





## **Τεχνολογία Λογισμικού**

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπο-  
λογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017



---

**DiagramHandlerController:**

«Controller» DiagramHandlerController
+DiagramHandleController(): void +checkEnteredData(): boolean +isThereData(diagram: DiagramPrEntity): boolean +createDiagram(): void +shareDiagram(diagram: Image): void +checkCompatibility(File): boolean +uploadDiagram(): void

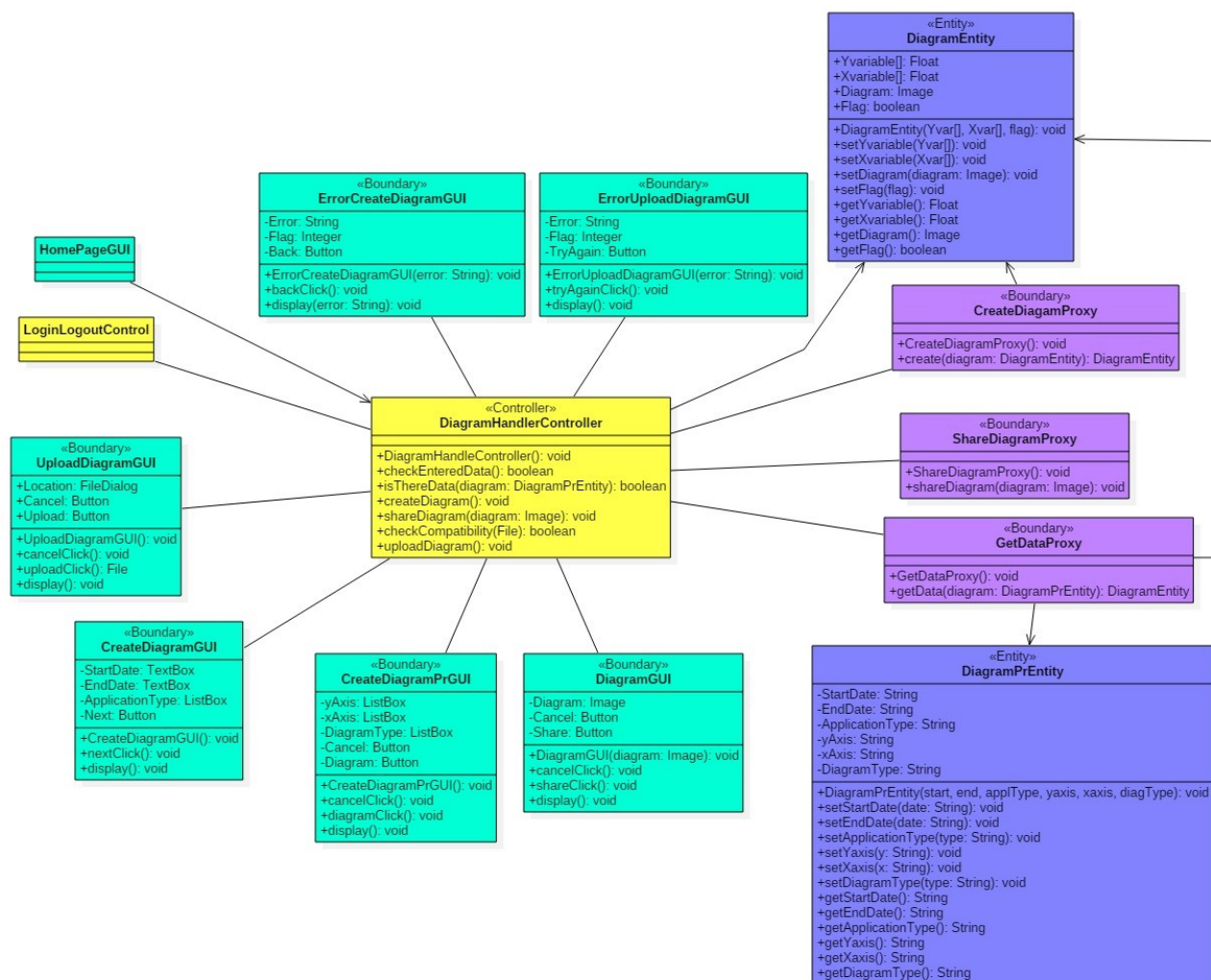
Η κλάση αυτή αποτελεί τον ελεγκτή του πακέτου και περιέχει όλες τις κατάλληλες συναρτήσεις για τον έλεγχο των λειτουργιών.

Συναρτήσεις κλάσης:

- DiagramHandlerController(): Συνάρτηση δόμησης της κλάσης.
- checkEnteredData(): Η μέθοδος αυτή ελέγχει εάν έχουν συμπληρωθεί όλα τα πεδία μιας φόρμας. Σε περίπτωση που έχουν συμπληρωθεί όλα επιστρέφει τη λογική τιμή TRUE, διαφορετικά επιστρέφει FALSE.
- isThereData(diagram: DiagramPrEntity): Η μέθοδος αυτή ελέγχει εάν υπάρχουν δεδομένα για την χρονική περίοδο που ζήτησε ο χρήστης. Σε περίπτωση που υπάρχουν επιστρέφει την λογική τιμή TRUE, διαφορετικά επιστρέφει FALSE.
- createDiagram(): Η μέθοδος αυτή καλεί την συνάρτηση display() της κλάσης CreateDiagramGUI για την δημιουργία διαγράμματος.
- shareDiagram(diagram: Image): Η μέθοδος αυτή αποστέλλει το διάγραμμα στο σύστημα instasis-backend μέσω του ShareDiagramProxy.
- checkCompatibility(File): Η μέθοδος αυτή ελέγχει εάν το αρχείο που επιλέγει ο χρήστης από τον προσωπικό του υπολογιστή, είναι αρχείο jpeg. Σε περίπτωση που αποτελεί αρχείο jpeg επιστρέφει την λογική τιμή TRUE, διαφορετικά επιστρέφει FALSE.
- uploadDiagram(): Η μέθοδος αυτή καλεί την συνάρτηση display() της κλάσης UploadDiagramGUI για το «ανέβασμα» ενός διαγράμματος από τον προσωπικό υπολογιστή του χρήστη.



### Διάγραμμα κλάσεων:



Εικόνα 3: Διάγραμμα κλάσεων πακέτου DiagramHandle

#### 1.2.1 Πακέτο QuestionnaireHandle

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που σχετίζονται με τη διαχείριση των ερωτηματολογίων, συμπεριλαμβανομένου της δημιουργίας τους, της τροποποίησής τους, της διαγραφής τους και της προβολής των αποτελεσμάτων τους.



## Τεχνολογία Λογισμικού

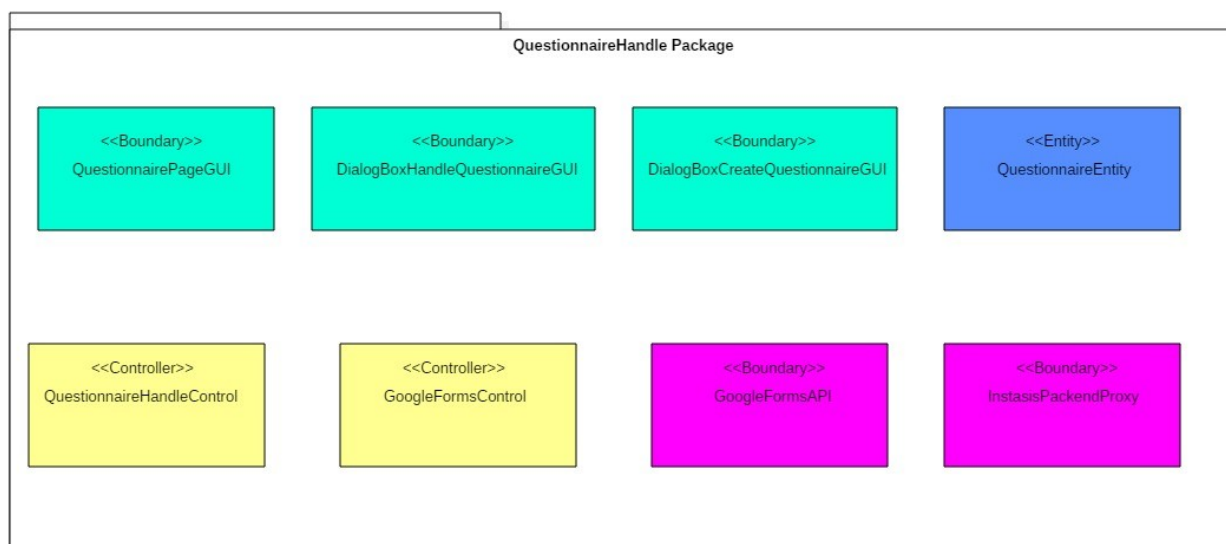
Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

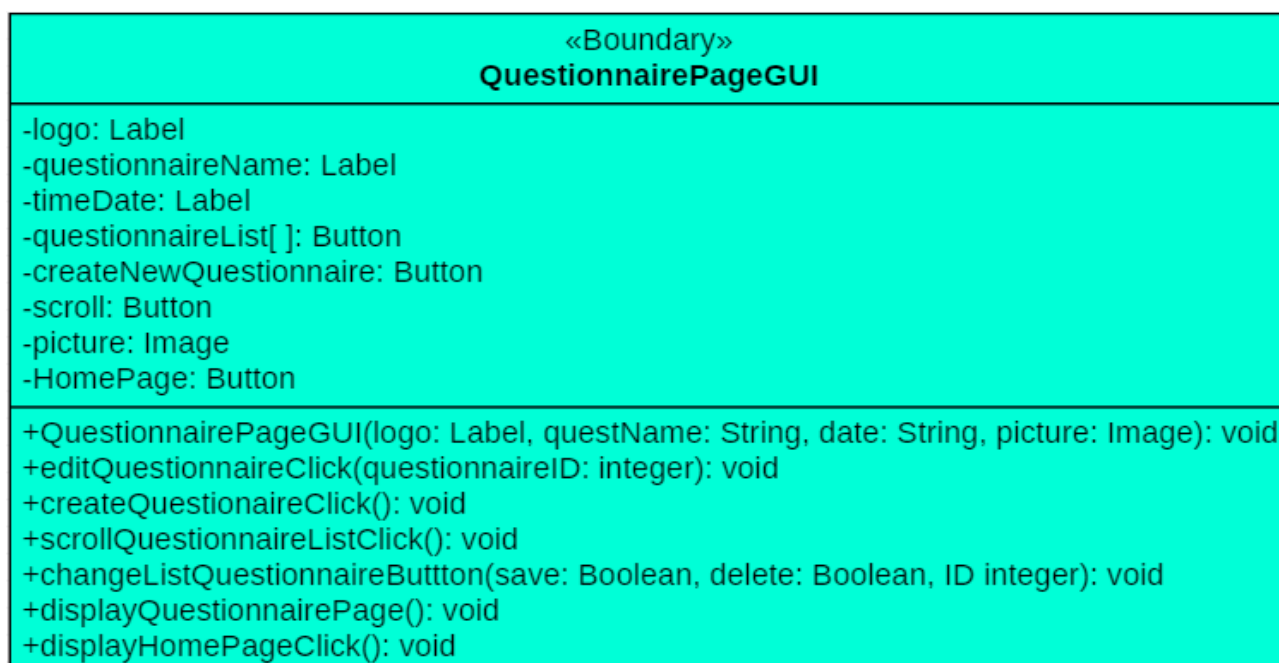
Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017



### QuestionnairePageGUI



Η κλάση αυτή αναπαριστά τη σελίδα ερωτηματολογίων δηλαδή την κλάση που λειτουργεί ως διεπαφή του χρήστη με το σύστημα για την προβολή της λίστας των ερωτηματολογίων καθώς και της δημιουργίας και επεξεργασίας ερωτηματολογίων.

Χαρακτηριστικά της κλάσης:

- logo: Λογότυπο της εφαρμογής instasis.



- `questionnaireName`: Επιγραφή πάνω απο την λίστα των ερωτηματολογίων.
- `timeDate`: Επιγραφή πάνω απο την λίστα των ερωτηματολογίων.
- `questionnaireList[]`: Λίστα κουμπίων που το κάθε ένα απο αυτά αντιστοιχεί σε ένα απο τα υπάρχοντα ερωτηματολόγια.
- `createNewQuestionnaire`: Κουμπί για την δημιουργία νέου ερωτηματολογίου.
- `scroll`: Κουμπί για την ολίσθηση της λίστας των ερωτηματολογίων.
- `picture`: Εικόνα στο φόντο(background) της σελίδας.
- `HomePage`: Κουμπί για να επιστρέψουμε στην αρχική σελίδα της εφαρμογής.

### Μέθοδοι της κλάσης:

- `QuestionnairePageGUI(logo:label,questName:String,date:String,picture:Image)`: Η μέθοδος αυτή είναι η συνάρτηση δόμησης της διεπαφής.
- `editQuestionnaireClick(questionnaireID:integer)`: Η μέθοδος αυτή ενεργοποιείται όταν επιλέξουμε κάποιο ερωτηματολόγιο από την λίστα ερωτηματολογίων για να το επεξεργαστούμε. Παίρνει σαν όρισμα το αναγνωριστικό κωδικό `questionnaireID` του ερωτηματολογίου και καλεί την μέθοδο `searchQuestionnaire(questionnaireID:integer)` του ελεγκτή `QuestionnaireHandleControl` που είναι υπεύθυνος για την επεξεργασία των ερωτηματολογίων.
- `createQuestionnaireClick()`: Η μέθοδος ενεργοποιείται όταν πατήσουμε το κουμπί `createNewQuestionnaire` για να δημιουργήσουμε νέο ερωτηματολόγιο και καλεί τον ελεγκτή `GoogleFormsControl` που είναι υπεύθυνος για την διαχείριση της API εφαρμογής.
- `scrollQuestionnaireListClick()`: Η μέθοδος αυτή πραγματοποιεί ολίσθηση στη λίστα ερωτηματολογίων.
- `changeListQuestionnaireButtton(save:Boolean,delete:Boolean,ID integer)`: Η μέθοδος χρησιμοποιείται για την ανανέωση των κουμπιών της λίστας των ερωτηματολογίων μετά από την δημιουργία ενός νέου ερωτηματολογίου ή μετά απο την διαγραφή κάποιου ήδη υπάρχοντος ερωτηματολογίου. Η επολογή για προσθήκη ή αφαίρεση πλήκτου γίνεται με βάση τα ορίσματα της μεθόδου.
- `displayQuestionnairePage()`: Η μέθοδος αυτή καλείται για να προβάλλει την εν λόγω σελίδα.
- `displayHomePageClick()`: Η συνάρτηση αυτή καλεί την μέθοδο `displayHomepage()` του ελεγκτή `LoginLogoutControl` του πακέτου `UserHandle` για να επιστρέψουμε στην αρχική σελίδα της εφαρμογής.



## **DialogBoxHandleQuestionnaireGUI**

«Boundary» <b>DialogBoxHandleQuestionnaireGUI</b>
-responses: Button -edit: Button -delete: Button -questObject: QuestionnaireEntity
+DialogBoxHandleQuestionnaireGUI(questObject: QuestionnaireEntity): void +viewResponsesClick(): void +editQuestionnaireClick(): void +deleteQuestionnaireClick(): void +displayDialog(): void

Η κλάση αποτελεί το αναδυόμενο παράθυρο που προκύπτει μετά από την επιλογή κάποιου ερωτηματολογίου και προβάλλει στον χρήστη τις επιλογές που έχει για επεξεργασία, διαγραφή ερωτηματολογίου και προβολή των αποτελεσμάτων του ερωτηματολογίου για να επιλέξει μία από αυτές.

Χαρακτηριστικά της κλάσης:

- responses: Το κουμπί για την επιλογή προβολής των αποτελεσμάτων ενός ερωτηματολογίου.
- edit: Το κουμπί για επιλογή επεξεργασία του ερωτηματολογίου.
- delete: Κουμπί για διαγραφή του ερωτηματολογίου.
- questObject: Αντικείμενο ερωτηματολογίου της κλάσης QuestionnaireEntity.

Μέθοδοι της κλάσης:

- DialogBoxHandleQuestionnaireGUI(in questObject: QuestionnaireEntity): Η μέθοδος δόμησης του αντικειμένου.
- viewResponsesClick(): Η μέθοδος χρησιμοποιείται για την προβολή των απαντήσεων των ερωτηματολογίων, καλείται όταν επιλέξουμε το responses κουμπί και καλεί τον GoogleFormsControl ελεκτή.
- editQuestionnaireClick(): Η μέθοδος καλείται όταν ο χρήστης επιλέξει το πλήκτρο edit και ο σκοπός της είναι η δυνατότητα για την επεξεργασία του ερωτηματολογίου.
- deleteQuestionnaireClick(): Η μέθοδος καλείται όταν ο χρήστης επιλέξει να διαγράψει το ερωτηματολόγιο.
- displayDialog(): Η μέθοδος χρησιμοποιείται για να προβάλλει την εν λόγω διεπαφή στην οθόνη.





## **DialogBoxCreateQuestionnaireGUI**

«Boundary» <b>DialogBoxCreateQuestionnaireGUI</b>	
-save: Button -continueEdit: Button -reject: Button -link: URL	
+DialogBoxCreateQuestionnaireGUI(link: URL) +saveQuestionnaireClick(): void +continueEditQuestionnaireClick(): void +rejectQuestionnaireClick(): void +displayDialog(): void	

Η κλάση αυτή αναπαριστά το αναδυόμενο παράθυρο που προκύπτει μετά την σύνταξη ενός ερωτηματολογίου και εμφανίζει στο χρήστη της επιλογές για αποθήκευση του ερωτηματολογίου που μόλις έφτιαξε, για συνέχιση της επεξεργασία του και για την απόρριψη του.

Χαρακτηριστικά της κλάσης:

- save:Κουμπί για την αποθήκευση του ερωτηματολογίου.
- continueEdit:Κουμπί για να συνέχιση της επεξεργασία του ερωτηματολογίου.
- reject:Κουμπί για την απόρριψη του ερωτηματολογίου.
- link: Ο ηλεκτρονικός σύνδεσμος του ερωτηματολογίου.

Μέθοδοι της κλάσης:

- DialogBoxCreateQuestionnaireGUI(link:URL):Η μέθοδος αυτή είναι η συνάρτηση δόμησης της διεπαφής.
- saveQuestionnaireClick(): Η συνάρτηση καλείται στην περίπτωση που επιλέξουμε την αποθήκευση του ερωτηματολογίου που μόλις συντάξαμε δηλαδή το κουμπί save.
- continueEditQuestionnaireClick():Η συνάρτηση ενεργοποιείται με το πλήκτρο continueEdit και δίνει στον χρήστη την δυνατότητα να συνεχίσει την σύνταξη του ερωτηματολογίου,η μέθοδος καλεί την κλάση
- editQuestionnaire(link:URL) του ελεγκτή GoogleFormsControl.
- rejectQuestionnaireClick(): Η μέθοδος απορρίπτει το ερωτηματολόγιο που μόλις δημιουργήσαμε,καλείται με το πλήκτρο reject.
- displayDialog(): Η μέθοδος χρησιμοποιείται για να προβάλει την εν λόγω διεπαφή στην οθόνη.



## QuestionnaireEntity

«Entity» QuestionnaireEntity
-questionnaireID: integer -name: String -date: String -link: URL -responsesLink: URL
+QuestionnaireEntity(questionnaireID: integer, name: String, date: string, link: URL, responsesLink: URL): void +setID(id: int): void +getID(): integer +setName(name: String): void +getName(): String +setDate(date: String): void +getDate(): String +setLink(link: URL): void +getLink(): URL +setResponsesLink(link: URL): void +getResponsesLink(): URL

Η κλάση αναπαριστά τις οντότητες των ερωτηματολογίων, την χρησιμοποιούμε για να δημιουργήσουμε αντικείμενα ερωτηματολογίων.

Χαρακτηριστικά της κλάσης:

- questionnaireID: Είναι ένας αριθμός που αντιστοιχεί σε ένα μοναδικό αναγνωριστικό κωδικό για κάθε ερωτηματολόγιο του συστήματος μας. Τον χρησιμοποιούμε ως ταυτότητα του ερωτηματολογίου μας.
- name: Ο τίτλος/όνομα του ερωτηματολογίου.
- date: Η ημερομηνία δημιουργίας του.
- link: Ο ηλεκτρονικός σύνδεσμος του ερωτηματολογίου.
- ResponsesLink: Ο ηλεκτρονικός σύνδεσμος των απαντήσεων του ερωτηματολογίου.

Μέθοδοι της κλάσης:

- QuestionnaireEntity(in questionnaireID: integer, in name: String, in date: string, in link: URL, in responsesLink: URL): Η συνάρτηση δόμησης της κλάσης.
- setID(id: int): Η μέθοδος θέτει τιμή στην μεταβλητή questionnaireID ίση με το όρισμα id της συνάρτησης.
- getID(): Η μέθοδος επιστρέφει την τιμή questionnaireID.
- setName(name: String): Η μέθοδος θέτει το όνομα του ερωτηματολογίου.
- getName(): Η μέθοδος επιστρέφει το όνομα του ερωτηματολογίου.
- setDate(date: String): Η μέθοδος θέτει την ημερομηνία δημιουργίας του ερωτηματολογίου.





- getDate(): Η μέθοδος επιστρέφει την ημερομηνία δημιουργίας του ερωτηματολογίου.
- setLink(link:URL): Η μέθοδος θέτει τον ηλεκτρονικό σύνδεσμο του ερωτηματολογίου.
- getLink(): Η μέθοδος επιστρέφει τον ηλεκτρονικό σύνδεσμο του ερωτηματολογίου.
- setResponsesLink(link:URL): Η μέθοδος θέτει τον ηλεκτρονικό σύνδεσμο των απαντήσεων του ερωτηματολογίου.
- getResponsesLink(): Η μέθοδος επιστρέφει τον ηλεκτρονικό σύνδεσμο των απαντήσεων του ερωτηματολογίου.

### QuestionnaireHandleControl

«Controller» QuestionnaireHandleControl
+QuestionnaireHandleControl(): void +saveQuestionnaire(link: URL): void +searchQuestionnaire(questionnaireID: integer): QuestionnaireEntity +deleteQuestionnaire(questionnaireID: integer): void +displayQuestionnairePageGUI(): void

Η κλάση αυτή διαχειρίζεται τα ερωτηματολόγια που υπάρχουν στο σύστημα, περιέχει τις κατέλληλες μεθόδους για δημιουργία, επεξεργασία και διαγραφή των ερωτηματολογίων καθώς και την προβολή των απαντήσεων των ερωτηματολογίων.

Μέθοδοι της κλάσης:

- QuestionnaireHandleControl(): Η μέθοδος δόμησης της συνάρτησης
- saveQuestionnaire(link:URL): Η μέθοδος αυτή δημιουργεί ένα αντικείμενο τύπου ερωτηματολογίου και καλεί την αντίστοιχη μέθοδο saveQuestionnaireIntoDB(in questObject:QuestionnaireEntity) της κλάσης InstasisPackendProxy για την αποθήκευση του ερωτηματολογίου στο instasisBackend. Η μέθοδος καλείται από την μέθοδο saveQuestionnaireClick(in link:URL) της διεπαφής DialogBoxCreateQuestionnaireGUI η οποία της παρέχει τον ηλεκτρονικό σύνδεσμο του ερωτηματολογίου που δημιουργήθηκε.
- searchQuestionnaire(questionnaireID:integer): Η μέθοδος καλείται από την συνάρτηση editQuestionnaireClick(in questionnaireID:integer) του αντικειμένου QuestionnairePageGUI και είναι υπεύθυνη για την αναζήτηση ενός ερωτηματολογίου στο



- σύστημα με βάση τον αποκλειστικό κωδικό του questionnaireID.
- deleteQuestionnaire(questionnaireID:integer): Η μέθοδος καλείται από τον ελεγκτή GoogleFormsControl μέσω της μεθόδου deleteQuestionnaire(in link:URL) και η χρήση της είναι να διαγράψει το ερωτηματολόγιο που βρίσκεται στο instasisBackend χρησιμοποιώντας φυσικά την κλάση ορίου InstasisPackendProxy.
- displayQuestionnairePageGUI(): Ο στόχος της μεθόδου είναι να καλέσει την displayQuestionnairePage() της κλάσης QuestionnairePageGUI και να προβάλει την σελίδα ερωτηματολογίων.

### GoogleFormsControl

«Controller» GoogleFormsControl
+GoogleFormsControl(): void +createQuestionnaire(): void +editQuestionnaire(link: URL): void +deleteQuestionnaire(link: URL): void +viewResponses(responsesLink: URL): void

Ο στόχος δημιουργίας της κλάσης αυτής είναι η διαχείριση των ενεργειών που έχουν να κάνουν με το εξωτερικό σύστημα API. Τέτοιες λειτουργίες είναι η διαγραφή, δημιουργία, επεξεργασία ερωτηματολογίων και προβολή αποτελεσμάτων. Επομένως ότι λειτουργίες σχετίζονται με αυτές τις ενέργειες και το εξωτερικό API της διαχειρίζεται αυτός ο ελεγκτής.

Μέθοδοι της κλάσης:

- GoogleFormsControl(): Η μέθοδος αυτή είναι η συνάρτηση δόμησης της κλάσης.
- createQuestionnaire(): Η συνάρτηση αυτή διαχειρίζεται την δημιουργία των ερωτηματολογίων. Καλείται από την μέθοδο createQuestionnaireClick() της κλάσης QuestionnairePageGUI και ύστερα καλεί την
- μέθοδο createNewQuestionnaire() της οριακής κλάσης GoogleFormsAPI για να δημιουργηθεί νέο ερωτηματολόγιο. Στην συνέχεια καλεί την διεπαφή DialogBoxCreateQuestionnaireGUI για να επιλέξει ο χρήστης μία από τις επιλογές για



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

---

αποθήκευση, απόρριψη ή συνέχεια στην σύνταξη του ερωτηματολογίου.

- `editQuestionnaire(link:URL)`: Η συνάρτηση καλείται για την επεξεργασία κάποιου υπάρχοντος ερωτηματολογίου καλώντας την μέθοδο `editQuestionnaire(in link:URL)` της κλάσης `GoogleFormsAPI`.
- `deleteQuestionnaire(link:URL)`: Καλεί την κλάση `GoogleFormsAPI` για να διαγράψει ένα ερωτηματολόγιο.
- `viewResponses(responsesLink:URL)`: Η μέθοδος καλεί την `GoogleFormsAPI` για προβολή των αποτελεσμάτων του ερωτηματολογίου.



---

### GoogleFormsAPI

«Boundary» GoogleFormsAPI
+editQuestionnaire(link: URL): void +deleteQuestionnaire(link: URL): void +createNewQuestionnaire(): URL +viewResponses(responseLink: URL): void

Η κλάση αυτή είναι η διεπαφή του συστήματος μας με το εξωτερικό σύστημα δημιουργίας ερωτηματολογίων Google Forms. Η κλάση περιέχει όλες εκείνες τις μεθόδους για την διαχείριση των ερωτηματολογίων σε συνεργασία με το εξωτερικό σύστημα.

Μέθοδοι της κλάσης:

- editQuestionnaire(link:URL): Η μέθοδος επικοινωνεί με το εξωτερικό σύστημα Google Forms για την επεξεργασία ενός ερωτηματολογίου.
- deleteQuestionnaire(link:URL): Η μέθοδος καλεί το εξωτερικό σύστημα Google Forms για να διαγραφεί η ηλεκτρονικός σύνδεσμος του ερωτηματολογίου από την εξωτερική εφαρμογή.
- createNewQuestionnaire(): Η μέθοδος δημιουργεί ένα νέο ερωτηματολόγιο στην εξωτερική εφαρμογή και επιστρέφει τον ηλεκτρονικό σύνδεσμο του.
- viewResponses(responseLink:URL): Η μέθοδος αυτή εμφανίζει με την βοήθεια του εξωτερικού συστήματος τις απαντήσεις των ερωτηματολογίων.

### InstasisPackendProxy

«Boundary» InstasisPackendProxy
+saveQuestionnaireIntoDB(questObject: QuestionnaireEntity): integer +searchQuestionnaireIntoDB(questionnaireID: integer): QuestionnaireEntity +deleteQuestionnaireIntoDB(questionnaireID: integer): void

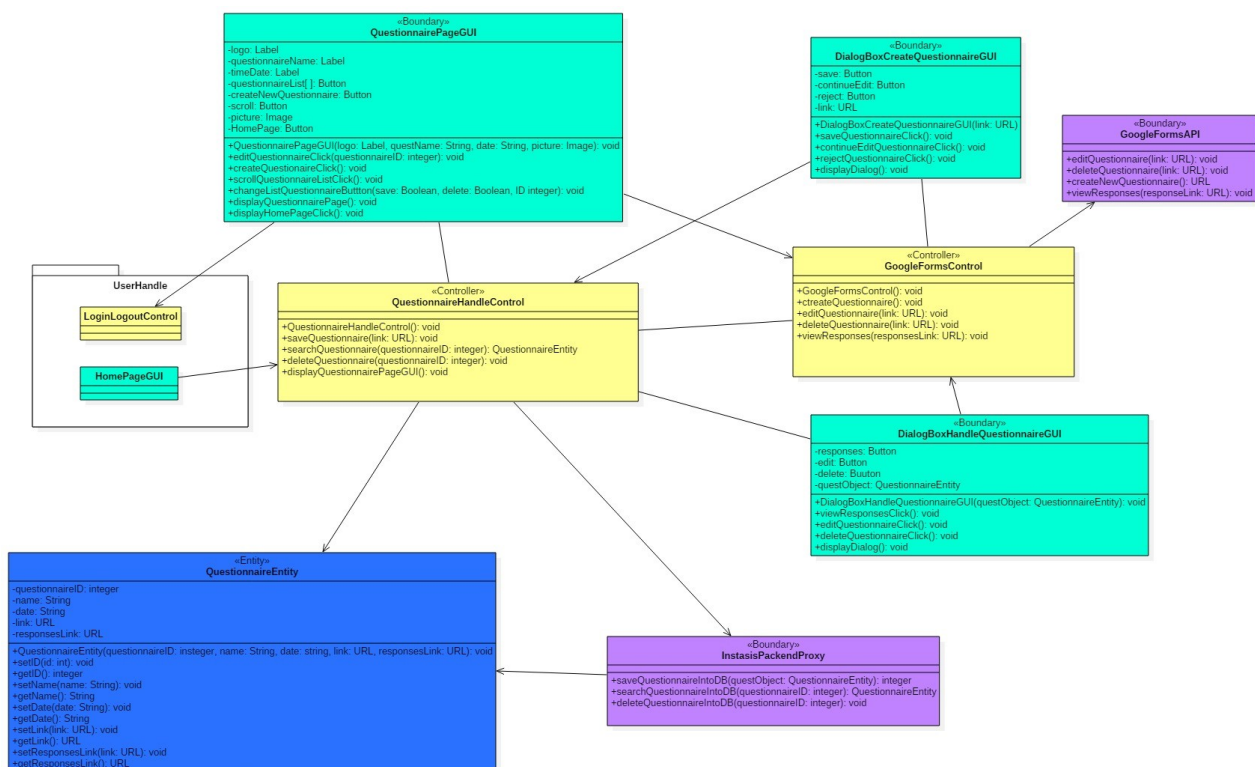
Η κλάση αυτή είναι η διεπαφή του συστήματός μας με το σύστημα InstasisBackend και περιέχει τις μεθόδους για την αποθήκευση, διαγραφή και αναζήτηση κάποιου ερωτηματολογίου στο σύστημα InstasisBackend.



### Μέθοδοι της κλάσης:

- `saveQuestionnaireIntoDB(questObject:QuestionnaireEntity): H` μέθοδος αυτή καλείται απο τον ελεγκτή `QuestionnaireHandleControl` και αποθηκεύει στο σύστημα ένα αντικείμενο ερωτηματολογίου που παίρνει σαν όρισμα. Η επιστρεφόμενη τιμή της είναι ο μοναδικός αναγνωριστικός κωδικός του ερωτηματολογίου `questionnaireID`.
- `searchQuestionnaireIntoDB(questionnaireID :integer): H` μέθοδος ψάχνει στο σύστημα `instasisBackend` και επιστρέφει ένα αντικείμενο ερωτηματολογίου με βάση τον μοναδικό κωδικό του `questionnaireID`.
- `deleteQuestionnaireIntoDB(questionnaireID:integer):H` μέθοδος παίρνει σαν όρισμα το μοναδικό κωδικό ενός αντικειμένου και το διαγράφει απο το σύστημα `instasisBacked`.

### Διάγραμμα Κλάσεων:



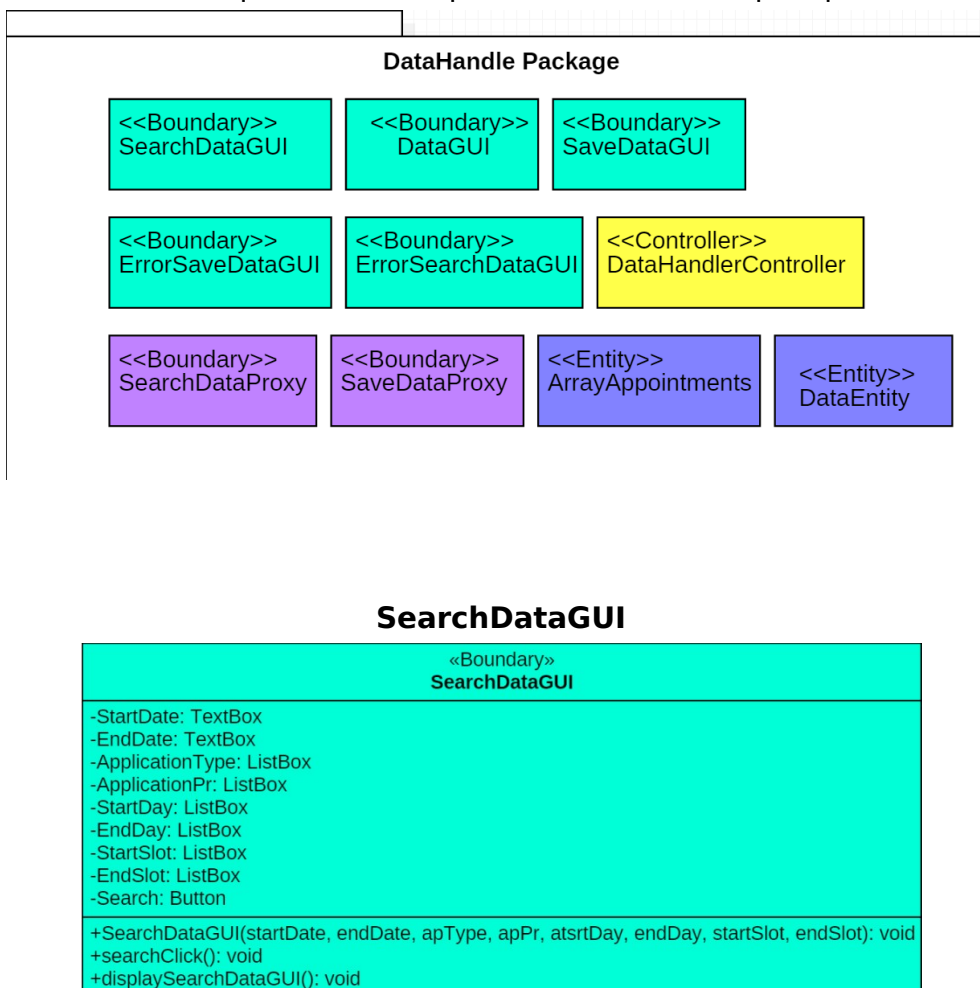
Εικόνα 4: Διάγραμμα κλάσεων του πακέτου QuestionnaireHandle



## 1.3 Πακέτο λεξιλογίου σεναρίων χαμηλής προτεραιότητας

### 1.3.1 Πακέτο DataHandle

Το πακέτο αυτό αναφέρεται στις λειτουργίες αναζήτησης δεδομένων και αποθήκευσης των αποτελεσμάτων στον προσωπικό υπολογιστή του χρήστη. Ο χρήστης μπορεί να αναζητήσει συγκεκριμένα δεδομένα, συμπληρώνοντας την αντίστοιχη φόρμα, και να τα αποθηκεύσει στον προσωπικό του υπολογιστή.



Η κλάση αυτή αποτελεί την διεπαφή για την συμπλήρωση από τον χρήστη φόρμας με τα κριτήρια της αναζήτησης.

Χαρακτηριστικά κλάσης:

- **StartDate**: Πεδίο συμπλήρωσης της ημερομηνίας από την οποία θα ξεκινήσει η αναζήτηση δεδομένων.
- **EndaDate**: Πεδίο συμπλήρωσης της ημερομηνίας μέχρι την οποία θα γίνει η αναζήτηση δεδομένων.
- **ApplicationType**: Λίστα επιλογής του τύπου των αιτήσεων που ενδιαφέρουν τον χρήστη.





## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

- ApplicationPr: Λίστα επιλογής της προτεραιότητας των αιτήσεων που ενδιαφέρουν τον χρήστη.
- StartDay: Λίστα επιλογής της αρχικής μέρας για την οποία θα γίνει η αναζήτηση δεδομένων.
- EndDay: Λίστα επιλογής της τελικής μέρας για την οποία θα γίνει αναζήτηση δεδομένων.
- StartSlot: Λίστα επιλογής του αρχικού slot για το οποίο θα γίνει αναζήτηση δεδομένων.
- EndSlot: Λίστα επιλογής του τελικού slot για το οποίο θα γίνει αναζήτηση δεδομένων.
- Search: Κουμπί για την αναζήτηση δεδομένων.

Συναρτήσεις κλάσης:

- SearchDataGUI(startDate, endDate, apType, apPr, atsrtday, endDay, startSlot, endSlot): Συνάρτηση δόμησης της κλάσης.
- searchClick(): Η μέθοδος αυτή δημιουργεί ένα αντικείμενο τύπου DataEntity με τα κριτήρια αναζήτησης και το αποστέλλει στον DataHandlerController.
- displaySearchDataGUI(): Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση της εν λόγω διεπαφής.

### DataGUI:

«Boundary» DataGUI
-StartDate: Label -EndDate: Label -ApplicationType: Label -ApplicationPr: Label -StartDay: Label -EndDay: Label -StartSlot: Label -EndSlot: Label -SaveData: Button -Cancel: Button -Scroll: Button -Appointments[]: AppointmentEntity
+DataGUI(startDate, endDate, apType, apPr, atsrtday, endDay, startSlot, endSlot, array): void +backClick(): void +saveDataClick(): void +scrollClick(): void +displayDataGUI(): void

Η κλάση αυτή αποτελεί την διεπαφή για την προβολή των αποτελεσμάτων της αναζήτησης.

Χαρακτηριστικά κλάσης:

- StartDate: Ετικέτα της ημερομηνίας από την οποία θα ξεκινήσει η αναζήτηση δεδομένων.





- **EndDate:** Ετικέτα της ημερομηνίας μέχρι την οποία θα γίνει η αναζήτηση δεδομένων.
- **ApplicationType:** Ετικέτα του τύπου των αιτήσεων που ενδιαφέρουν τον χρήστη.
- **ApplicationPr:** Ετικέτα της προτεραιότητας των αιτήσεων που ενδιαφέρουν τον χρήστη.
- **StartDay:** Ετικέτα της αρχικής μέρας για την οποία θα γίνει η αναζήτηση δεδομένων.
- **EndDay:** Ετικέτα της τελικής μέρας για την οποία θα γίνει αναζήτηση δεδομένων.
- **StarSlot:** Ετικέτα του αρχικού slot για το οποίο θα γίνει αναζήτηση δεδομένων.
- **EndSlot:** Ετικέτα του τελικού slot για το οποίο θα γίνει αναζήτηση δεδομένων.
- **SaveData:** Κουμπί για την αποθήκευση των αποτελεσμάτων της αναζήτησης.
- **Cancel:** Κουμπί για ακύρωση και επιστροφή στην αρχική σελίδα.
- **Scroll:** Κουμπί ολίσθησης για την εμφάνιση των αποτελεσμάτων που υπάρχουν στη λίστα.
- **Appointments[]:** πίνακας με αντικείμενα τύπου **AppointmentEntity**, που περιέχουν πληροφορίες για το εκάστοτε ραντεβού.

Συναρτήσεις κλάσης:

- **DataGUI(startDate, endDate, apType, apPr, atsrtDay, endDay, startSlot, endSlot, array):** Συνάρτηση δόμησης της κλάσης.
- **backClick():** καλεί την συνάρτηση **displayHomepage()** του **LoginLogoutControl**.
- **saveDataClick():** καλεί την συνάρτηση **saveData(array: ArrayAppointments)** του **DataHandlerController**.
- **scrollClick():** Η συνάρτηση αυτή θέτει σε λειτουργία τον ολισθαίνοντα σύνδεσμο προβολής της λίστας των αποτελεσμάτων της αναζήτησης.
- **displayDataGUI():** Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση της εν λόγω διεπαφής.

### **SaveDataGUI:**



«Boundary» SaveDataGUI
-Location: FileDialog -Name: TextBox -Save: Button
+SaveDataGUI(): void +saveClick(): void +displaySaveDataGUI(): void

Η κλάση αυτή αποτελεί την διεπαφή για την αποθήκευση στον προσωπικό υπολογιστή του χρήστη των αποτελεσμάτων που προέκυψαν στην αναζήτηση.

Χαρακτηριστικά κλάσης:

- Location: Παράθυρο επιλογής του φακέλου αποθήκευσης των αποτελεσμάτων της αναζήτησης.
- Name: Πεδίο συμπλήρωσης του ονόματος του αρχείου των αποτελεσμάτων της αναζήτησης.
- Save: Κουμπί αποθήκευσης των αποτελεσμάτων της αναζήτησης.

Συναρτήσεις κλάσης:

- SaveDataGUI(): Συνάρτηση δόμησης της κλάσης.
- saveClick(): Η μέθοδος αυτή καλεί την συνάρτηση checkEnteredData() του DataHandlerController.
- displaySaveDataGUI(): Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση της εν λόγω διεπαφής.

### ErrorSearchDataGUI:

«Boundary» ErrorSearchDataGUI
-Error: String -Back: Button
+ErrorSearchDataGUI(): void +display(error: string): void +backClick(): void

Η κλάση αυτή αποτελεί την διεπαφή για εμφάνιση μηνύματος σφάλματος σε περίπτωση που δεν βρεθούν τα δεδομένα που ζήτησε ο χρήστης.

Χαρακτηριστικά κλάσης:

- Error: Μήνυμα σφάλματος εύρεσης των δεδομένων που ζήτησε ο χρήστης.
- Back: Κουμπί για επιστροφή στην αρχική σελίδα.



### Συναρτήσεις κλάσης:

- `ErrorSearchDataGUI(error: String)`: Συνάρτηση δόμησης της κλάσης.
- `backClick()`: Η μέθοδος αυτή καλεί τη συνάρτηση `displayHomepage()` του `LoginLogoutControl`.
- `display()`: Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση της εν λόγω διεπαφής.

### ErrorSaveDataGUI:

«Boundary» <b>ErrorSaveDataGUI</b>
-Error: String -Back: Button
+ErrorSaveDataGui(): void +display(error: String): void +backClick(): void

Η κλάση αυτή αποτελεί την διεπαφή για την εμφάνιση μηνύματος σφάλματος σε περίπτωση που ο χρήστης δεν συμπληρώσει τα πεδία της φόρμας για την αποθήκευση των αποτελεσμάτων στον προσωπικό του υπολογιστή.

### Χαρακτηριστικά κλάσης:

- `Error`: Μήνυμα σφάλματος μη συμπλήρωσης των πεδίων της φόρμας αποθήκευσης των αποτελεσμάτων της αναζήτησης.
- `Back`: Κουμπί για κλείσιμο της εν λόγω διεπαφής.

### Συναρτήσεις κλάσης:

- `ErrorSaveDataGUI(error: String)`: Συνάρτηση δόμησης της κλάσης.
- `backClick()`: Η μέθοδος αυτή κλείνει την εν λόγω διεπαφή και επιστρέφει τον έλεγχο στον `DataHandlerController`.
- `display()`: Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση της εν λόγω διεπαφής.

### SearchDataProxy:

«Boundary» <b>SaveDataProxy</b>
+SaveDataProxy(): void +save(array, location, name): void



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

Η κλάση αυτή χρησιμοποιείται για την επικοινωνία με το σύστημα instasis-backend, για την λήξη των πληροφοριών που ζητήθηκαν από τον χρήστη κατά την αναζήτηση.

Συναρτήσεις κλάσης:

- SearchDataProxy(): Συνάρτηση δόμησης της κλάσης.
- search(data: DataEntity): Η μέθοδος αυτή επικοινωνεί με το σύστημα instasis-backend και επιστρέφει ένα αντικείμενο τύπου ArrayAppointments.

### SaveDataProxy:

«Boundary» SearchDataProxy
+SearchDataProxy(): void +search(data: DataEntity): ArrayAppointments

Η κλάση αυτή επικοινωνεί με τον προσωπικό υπολογιστή του χρήστη για την αποθήκευση των αποτελεσμάτων που προέκυψαν κατά την αναζήτηση.

Συναρτήσεις κλάσης:

- SaveDataProxy(): Συνάρτηση δόμησης της κλάσης.
- save(array, location, name): η μέθοδος αυτή επικοινωνεί με το λειτουργικό σύστημα του προσωπικού υπολογιστή του χρήστη και αποθηκεύει τις πληροφορίες.

### DataEntity:



«Entity» DataEntity
-StartDate: String -EndDate: String -ApplicationType: String -ApplicationPr: String -StartDay: String -EndDay: String -StartSlot: String -EndSlot: String
+DataEntity(startDate, endDate, apType, apPr, startDay, endDay, startSlot, endSlot): void +setStartDate(startDate): void +setEndDate(endDate): void +setApplicationType(apType): void +setApplicationPr(apPr): void +setStartDay(startDay): void +setEndDay(endDay): void +setStartSlot(startSlot): void +setEndSlot(endSlot): void +getStartDate(): String +getEndDate(): String +getApplicationType(): String +getApplicationPr(): String +getStartDay(): String +getEndDay(): String +getStartSlot(): String +getEndSlot(): String

Η κλάση αυτή είναι κλάση οντότητα και περιέχει στοιχεία με τα κριτήρια αναζήτησης.

Χαρακτηριστικά κλάσης:

- **StartDate:** Ημερομηνία από την οποία θα ξεκινήσει η αναζήτηση δεδομένων.
- **EndaDate:** Ημερομηνία μέχρι την οποία θα γίνει η αναζήτηση δεδομένων.
- **ApplicationType:** Τύπος των αιτήσεων που ενδιαφέρουν τον χρήστη.
- **ApplicationPr:** Προτεραιότητα των αιτήσεων που ενδιαφέρουν τον χρήστη.
- **StartDay:** Αρχική μέρα για την οποία θα γίνει η αναζήτηση δεδομένων.
- **EndDay:** Τελική μέρα για την οποία θα γίνει αναζήτηση δεδομένων.
- **StarSlot:** Αρχικό slot για το οποίο θα γίνει αναζήτηση δεδομένων.
- **EndSlot:** Τελικό slot για το οποίο θα γίνει αναζήτηση δεδομένων.

Συναρτήσεις κλάσης:

- **DataEntity(startDate, endDate, apType, apPr, startDay, endDay, startSlot, endSlot):** Συνάρτηση δόμησης της κλάσης.
- **setStartDate(startDate: String):** Η μέθοδος αυτή ορίζει την ημέρα από την οποία θα ξεκινήσει η αναζήτηση δεδομένων.
- **setEndDate(endDate: String):** Η μέθοδος αυτή ορίζει την ημέρα μέχρι την οποία θα γίνει η αναζήτηση δεδομένων.



- `setApplicationType(apType: String)`: Η μέθοδος αυτή ορίζει τον τύπο των αιτήσεων που ενδιαφέρουν τον χρήστη.
- `setApplicationPr(apPr: String)`: Η μέθοδος αυτή ορίζει την προτεραιότητα των αιτήσεων που ενδιαφέρουν τον χρήστη.
- `setStartDay(startDay: String)`: Η μέθοδος αυτή ορίζει την αρχική μέρα για την οποία θα γίνει αναζήτηση.
- `setEndDay(endDay: String)`: Η μέθοδος αυτή ορίζει την τελική μέρα για την οποία θα γίνει αναζήτηση.
- `setStartSlot(startSlot: String)`: Η μέθοδος αυτή ορίζει τον αρχικό slot για το οποίο θα γίνει αναζήτηση.
- `setEndSlot(endSlot: String)`: Η μέθοδος αυτή ορίζει το τελικό slot για το οποίο θα γίνει αναζήτηση.
- `getStartDate()`: Η μέθοδος αυτή επιστρέφει την ημέρα από την οποία θα ξεκινήσει η αναζήτηση δεδομένων.
- `getEndDate()`: Η μέθοδος αυτή επιστρέφει την ημέρα μέχρι την οποία θα γίνει η αναζήτηση δεδομένων.
- `getApplicationType()`: Η μέθοδος αυτή επιστρέφει τον τύπο των αιτήσεων που ενδιαφέρουν τον χρήστη.
- `getApplicationPr()`: Η μέθοδος αυτή επιστρέφει την προτεραιότητα των αιτήσεων που ενδιαφέρουν τον χρήστη.
- `getStartDay()`: Η μέθοδος αυτή επιστρέφει την αρχική μέρα για την οποία θα γίνει αναζήτηση.
- `getEndDay()`: Η μέθοδος αυτή επιστρέφει την τελική μέρα για την οποία θα γίνει αναζήτηση.
- `getStartSlot()`: Η μέθοδος αυτή επιστρέφει το αρχικό slot για το οποίο θα γίνει αναζήτηση.
- `getEndSlot()`: Η μέθοδος αυτή επιστρέφει το τελικό slot για το οποίο θα γίνει αναζήτηση.

### ArrayAppointments:

«Entity» ArrayAppointments
-Array[]: AppointmentEntity -DataExistance: boolean
+ArrayAppointments(apArray[], temp): void +setAppointments(apArray[]): void +setDataExistance(temp): boolean +getAppointments(): AppointmentEntity +getDataExistance(): boolean

Η κλάση αυτή είναι κλάση οντότητας και περιέχει τα αποτελέσματα της αναζή-



τησης.

Χαρακτηριστικά κλάσης:

- `Array[]`: Πίνακας που περιέχει αντικείμενα τύπου `AppointmentEntity` με τις πληροφορίες κάθε ραντεβού.
- `DataExistence`: Μεταβλητή αναγνώρισης της ύπαρξης των δεδομένων που ζήτησε ο χρήστης. Όταν είναι 1 υπάρχουν τα δεδομένα, διαφορετικά, όταν είναι μηδέν, δεν υπάρχουν.

Συναρτήσεις κλάσης:

- `ArrayAppointments(apArray[], temp)`: Συνάρτηση δόμησης της κλάσης.
- `setAppointments(apArray[]: AppointmentEntity)`: Η μέθοδος αυτή ορίζει τον πίνακα με τις πληροφορίες των ραντεβού.
- `setDataExistence(temp: boolean)`: Η μέθοδος αυτή ορίζει την μεταβλητή ύπαρξης των δεδομένων που ζήτησε ο χρήστης.
- `getAppointments()`: Η μέθοδος αυτή επιστρέφει τον πίνακα με τις πληροφορίες των ραντεβού.
- `getDataExistence()`: Η μέθοδος αυτή επιστρέφει την μεταβλητή ύπαρξης των δεδομένων που ζήτησε ο χρήστης.

### DataHandlerController:

«Controller» DataHandlerController
+DataHandler(): void +searchData(): void +saveData(array: ArrayAppointments): void +isThereData(data: DataEntity): boolean +checkEnteredData(): boolean

Η κλάση αυτή είναι ο ελεγκτής του πακέτου και περιέχει τις κατάλληλες συναρτήσεις για των έλεγχο των λειτουργιών.

Συναρτήσεις κλάσης:

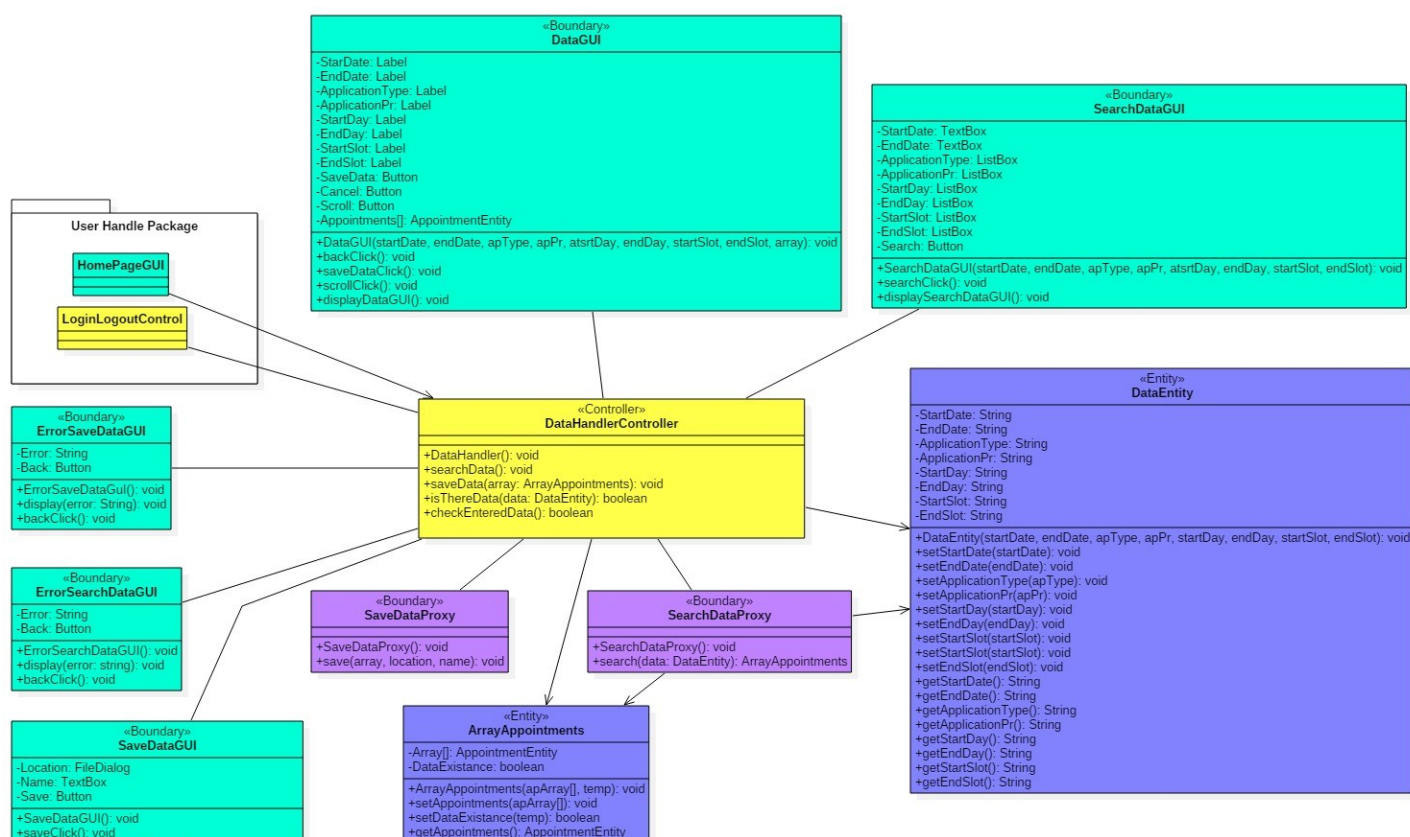
- `DataHandler()`: Συνάρτηση δόμησης της κλάσης.
- `searchData(data: DataEntity)`: Η μέθοδος αυτή καλεί την συνάρτηση `displaySearchDataGUI()` της κλάσης `SearchDataGUI`.
- `isThereData(array: ArrayAppointments)`: Η συνάρτηση αυτή ελέγχει εάν υπάρχουν τα δεδομένα που ζήτησε ο χρήστης. Σε περίπτωση που υπάρχουν επιστρέφει τη λογική τιμή `TRUE`, διαφορετικά επιστρέφει `FALSE`.
- `checkEnteredData()`: Η συνάρτηση αυτή ελέγχει εάν ο χρήστης συμπλήρωσε τον φάκελο αποθήκευσης και το όνομα με το οποίο θα αποθηκευτούν τα δεδομένα στον προσωπικό του υπολογιστή. Σε περίπτωση που ο χρήστης συμπλήρωσε τα πεδία επιστρέφει `TRUE`, διαφορετικά επιστρέφει `FALSE`.





- saveData(array: ArrayAppointments): Η μέθοδος αυτή καλεί την συνάρτηση displaySaveDataGUI() της κλάσης SaveDataGUI.

### Διάγραμμα κλάσεων:



Εικόνα 5: Διάγραμμα κλάσεων του πακέτου DataHandle



## **2 Δυναμική μοντελοποίηση του συστήματος**

### **2.1 Πακέτο Ανάλυσης Δεδομένων**

Το πακέτο αυτό περιλαμβάνει την όλη ανάλυση δεδομένων και δημιουργία προβλέψεων. Ο Χρήστης μπορεί να δημιουργεί και να διαγράφει ρυθμίσεις πρόβλεψης απασχόλησης. Επίσης, το σύστημα αυτόνομα μπορεί να ανταποκρίνεται σε αιτήματα του εξωτερικού συστήματος InstasisBackend για να δημιουργεί και να του στέλνει προβλέψεις πάνω σε αιτήματα φοιτητών για ραντεβού ή διαγράμματα προβλεπόμενης απασχόλησης της γραμματείας.

Οι λειτουργικές απαιτήσεις που σχετίζονται με αυτό το πακέτο χρήσης:

- <ΛΑ- 9> Το σύστημα θα πρέπει να μπορεί να κάνει προβλέψεις όταν του το ζητήσει το instasis-backend.
- <ΛΑ- 10> Το σύστημα θα πρέπει να μπορεί να αποστέλλει τις προβλέψεις στο instasis-backend.
- <ΛΑ- 11> Το σύστημα θα πρέπει να μπορεί να εκδίδει διαγράμματα με την πρόβλεψη απασχόλησης της γραμματείας.
- <ΛΑ- 12> Ο χρήστης θα πρέπει να μπορεί να σταματάει την έκδοση διαγραμμάτων με την πρόβλεψη απασχόλησης της γραμματείας.

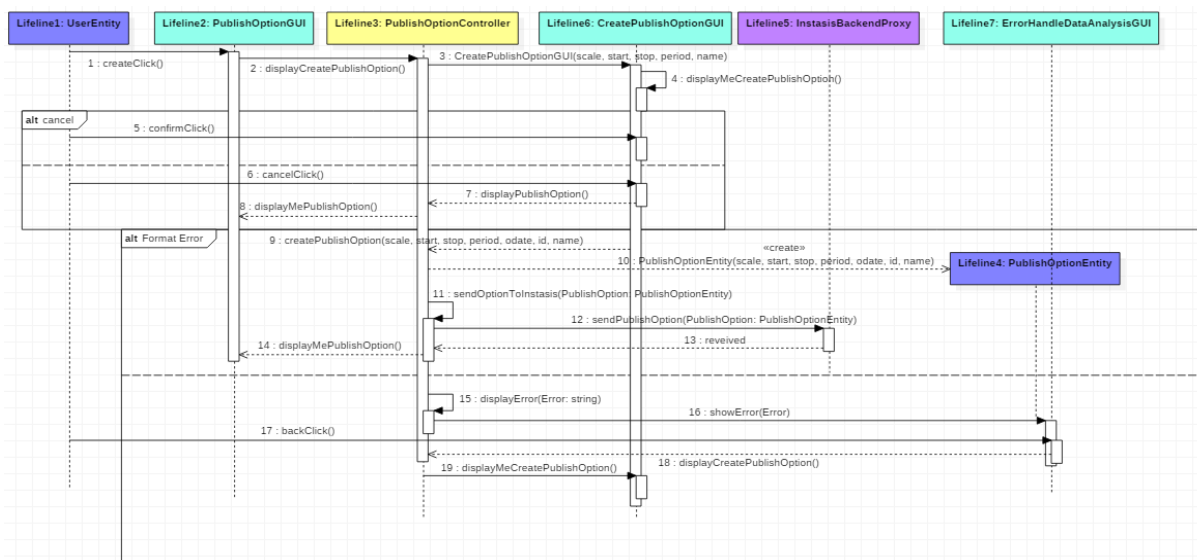
Επειδή, ωστόσο είναι αδύνατη η παρουσίαση όλων των αφηγήσεων θα παρουσιαστούν δύο, οι θεωρούμενες ως πιο βασικές αυτού του πακέτου. Η πρώτη έχει να κάνει με την δημιουργία ρυθμίσεων για την έκδοση διαγραμμάτων πρόβλεψης απασχόλησης της γραμματείας και την αποστολή τους στο InstasisBackend. Η δεύτερη έχει να κάνει με την διαγραφή ρυθμίσεων για την έκδοση διαγραμμάτων πρόβλεψης απασχόλησης της γραμματείας από το InstasisBackend όπου και είναι αποθηκευμένες.

#### **2.1.1 Αφήγηση σεναρίου δημιουργίας ρυθμίσεων**

Ο χρήστης έχει ήδη συνδεθεί στην εφαρμογή με λογαριασμό αναλυτή και έχει επιλέξει από την “Αρχική Σελίδα” τον σύνδεσμο “Ρυθμίσεις Προβλέψεων”. Εμφανίζεται λοιπόν στον χρήστη η διεπαφή της σελίδας αυτής PublishOptionGUI. Εκεί ο χρήστης επιλέγει τον σύνδεσμο Νέα “Ρύθμιση Πρόβλεψης”. Καλείται η συνάρτηση displayCreatePublishOption() του ελεγκτή PublishOptionController που με την σειρά της καλεί την συνάρτηση δόμησης της κλάσης CreatePublishOptionGUI για να φορτωθεί η διεπαφή δημιουργίας νέας ρύθμισης. Ανοίγει, λοιπόν, η σελίδα “Νέα Ρύθμιση Πρόβλεψης” με την κλήση της displayMeCreatePublishOption() που εμφανίζει την διεπαφή στο χρήστη. Ο χρήστης συμπληρώνει τα πεδία και επιλέγει έναν από τους συνδέσμους



“Επιβεβαίωση” “Άκυρο”. Αν επιλέξει “Άκυρο” καλείται η συνάρτηση `displayPublishOption()` του ελεγκτή `PublishOptionController` για να επιστρέψει στην οθόνη την διεπαφή `PublishOptionGUI` μέσω κλήσης στη συνάρτηση `displayMePublishOption()`. Ανοίγει λοιπόν και πάλι στον χρήστη η σελίδα “Ρυθμίσεις Προβλέψεων”. Αν ο χρήστης, όμως, επιλέξει “Επιβεβαίωση” παραπάνω, από την διεπαφή νωρίτερα καλείται η συνάρτηση `createPublishOption(in scale, in start, in stop, in period, in odate, in id, in name)` του ελεγκτή `PublishOptionController`. Αυτή λαμβάνει τις επιλογές του χρήστη και εξετάζει εάν είναι αποδεκτές. Αν βρει κάποιο σφάλμα καλεί την `displayError(in Error:string)` που με τη σειρά της καλεί την `showError(in Error)` της διεπαφής `ErrorHandleDataAnalysisGUI` η οποία εμφανίζει μήνυμα σφάλματος στον χρήστη, που εξηγεί ποια ήταν η λάθος επιλογή. Εκεί ο χρήστης έχει μόνο μια επιλογή, να επιλέξει “Πίσω”, οπότε θα κληθεί η `displayCreatePublishOption()` που όπως νωρίτερα θα καταλήξει να φέρει τον χρήστη στη διεπαφή `CreatePublishOptionGUI` και άρα στη σελίδα “Νέα Ρύθμιση Πρόβλεψης” για να ξανά συμπληρώσει σωστά τα πεδία. Εάν νωρίτερα όμως η `createPublishOption` συνάρτηση δεν βρει κανένα σφάλμα στις επιλογές του χρήστη, καλεί την συνάρτηση δόμησης της κλάσης οντότητας `PublishOptionEntity` για να δημιουργηθεί το αντικείμενο Ρύθμισης Διαγραμμάτων Πρόβλεψης απασχόλησης. Η `createPublishOption`, επίσης καλεί την `sendOptionToInstasis(PublishOption:PublishOptionEntity)` από τον ίδιο ελεγκτή με σκοπό να σταλεί το αντικείμενο που δημιουργήθηκε μόλις ως όρισμα. Για να σταλεί το αντικείμενο η `sendOptionToInstasis` καλεί την `sendPublishOption(PublishOption:PublishOptionEntity)` της οριακής κλάσης `InstasisBackendProxy`. Η συνάρτηση αυτή λαμβάνει το αντικείμενο της Ρύθμισης Πρόβλεψης και το στέλνει στο εξωτερικό σύστημα `InstasisBackend` που το αποθηκεύει στη βάση δεδομένων του μέσω δικών του διαδικασιών. Τέλος, καλείται από τον ελεγκτή η συνάρτηση `displayMePublishOption()` της διεπαφής `PublishOptionGUI` που εμφανίζει στον χρήστη την σελίδα “Ρυθμίσεις Προβλέψεων”.



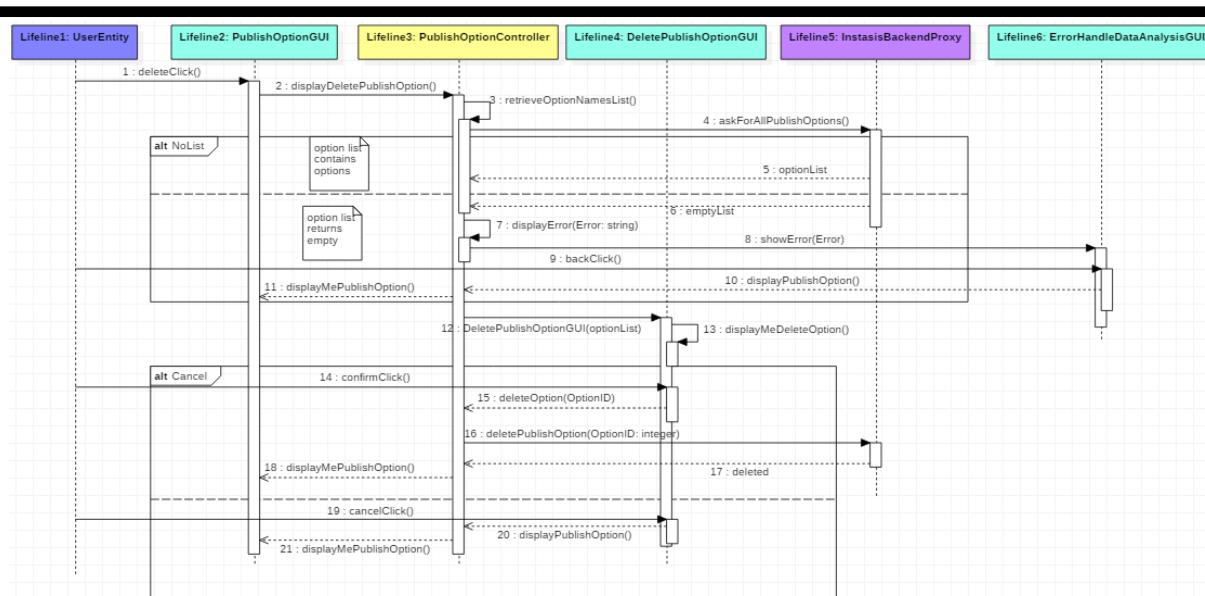
Εικόνα 6: Διάγραμμα ακολουθιών σεναρίου δημιουργίας ρυθμίσεων



---

**2.1.2 Αφήγηση σεναρίου διαγραφής ρυθμίσεων**

Ο χρήστης έχει ήδη συνδεθεί στην εφαρμογή με λογαριασμό αναλυτή και έχει επιλέξει από την “Αρχική Σελίδα” τον σύνδεσμο “Ρυθμίσεις Προβλέψεων”. Εμφανίζεται λοιπόν στον χρήστη η διεπαφή της σελίδας αυτής PublishOptionGUI. Εκεί ο χρήστης επιλέγει τον σύνδεσμο “Διαγραφή Ρύθμισης Πρόβλεψης”. Καλείται η συνάρτηση displayDeletePublishOption() του ελεγκτή PublishOptionController που με την σειρά της καλεί την καλεί τη συνάρτηση retrieveOptionNamesList(). Αυτή η συνάρτηση του ελεγκτή καλεί την askForAllPublishOptions() της οριακής κλάσης InstasisBackendProxy, η οποία επικοινωνεί με το InstasisBackend και επιστρέφει την λίστα με τις αποθηκευμένες “Ρυθμίσεις Πρόβλεψης”. Αυτή η λίστα έχοντας επιστρέψει στον ελεγκτή, του δίνει το δικαίωμα να καλέσει τις κατάλληλες συναρτήσεις για να προχωρήσει η διαδικασία. Αν η λίστα επιστρέψει κενή, ενώ εκτελείται ακόμα η retrieveOptionNamesList() του ελεγκτή, καλείται μια άλλη συνάρτηση του ελεγκτή. Αυτή είναι η displayError(Error:string) που είναι υπεύθυνη να καλέσει την συνάρτηση showError(Error) της διεπαφής ErrorHandlerDataAnalysisGUI η οποία εμφανίζει μήνυμα σφάλματος στον χρήστη, που εξηγεί πως δεν υπάρχει καμία ρύθμιση. Εκεί ο χρήστης έχει μόνο μια επιλογή, να επιλέξει “Πίσω”, οπότε θα κληθεί η displayPublishOption() που όπως νωρίτερα θα καταλήξει να φέρει τον χρήστη στη διεπαφή PublishOptionGUI και άρα στη σελίδα “Ρυθμίσεις Προβλέψεων”. Αν ωστόσο νωρίτερα η λίστα με τις ρυθμίσεις έχει επιστρέψει με τιμές από το InstasisBackend ο ελεγκτής PublishOptionController καλεί την συνάρτηση δόμησης της κλάσης DeletePublishOptionGUI για να φορτωθεί η διεπαφή διαγραφής μιας ρύθμισης, περνώντας σαν όρισμα την λίστα με τις διαθέσιμες προβλέψεις για να τις εμφανίσει στο χρήστη. Ανοίγει, λοιπόν, η σελίδα “Διαγραφή Ρύθμισης Πρόβλεψης” με την κλήση της displayMeDeletePublishOption() που εμφανίζει την διεπαφή στο χρήστη. Ο χρήστης τώρα επιλέγει μια Ρύθμιση από τη λίστα και έχει 2 επιλογές, “Άκυρο” και “Επιβεβαίωση”. Αν επιλέξει “Άκυρο” καλείται η συνάρτηση displayPublishOption() του ελεγκτή PublishOptionController για να επιστρέψει στην οθόνη την διεπαφή PublishOptionGUI μέσω κλήσης στη συνάρτηση displayMePublishOption(). Ανοίγει λοιπόν και πάλι στον χρήστη η σελίδα “Ρυθμίσεις Προβλέψεων”. Αν ο χρήστης, όμως, επιλέξει “Επιβεβαίωση” καλείται η deleteOption(OptionID) του ελεγκτή PublishOptionController για να διαγραφεί η ρύθμιση που επιλέχθηκε. Ο καθορισμός της ρύθμισης γίνεται μέσω του μοναδικού κωδικού κάθε μίας ρύθμισης που έχει περάσει σαν όρισμα νωρίτερα. Για να διαγραφεί η ρύθμιση καλείται η deletePublishOption(OptionID) της οριακής κλάσης InstasisBackendProxy. Τέλος, η διαδικασία ολοκληρώνεται με την κλήση της συνάρτησης displayMePublishOption() της διεπαφής PublishOptionGUI η οποία ανοίγει τη σελίδα “Ρυθμίσεις Προβλέψεων”.



Εικόνα 7: Διάγραμμα ακολουθιών σεναρίου διαγραφής ρυθμίσεων

## 2.2 Πακέτο δημιουργίας διαγράμματος στατιστικών

Το πακέτο αυτό περιλαμβάνει την αρχική σελίδα που βλέπει ο χρήστης, καθώς και τις σελίδες για τη δημιουργία και προβολή ενός διαγράμματος στατιστικών. Ο χρήστης επιλέγει να δημιουργήσει ένα διάγραμμα συμπληρώνοντας τα απαραίτητα στοιχεία.

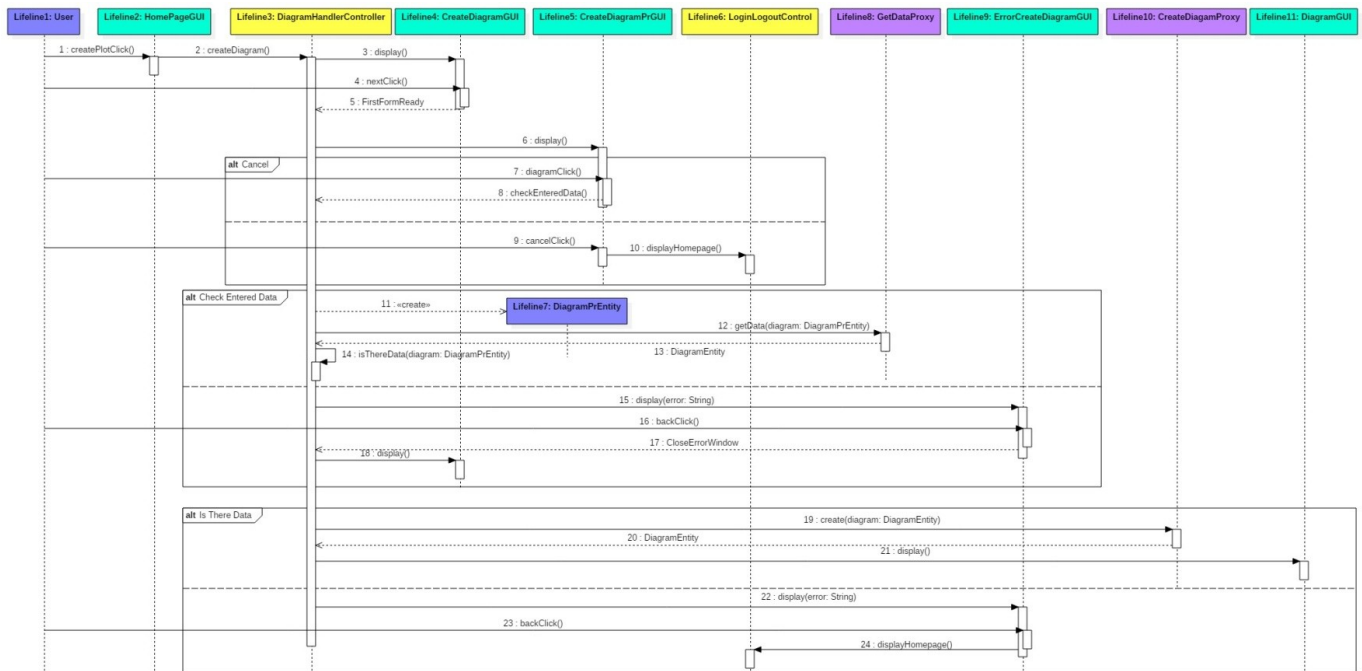
Οι λειτουργικές απαιτήσεις που σχετίζονται με αυτό το πακέτο είναι:

- <ΛΑ2> Ο χρήστης θα πρέπει να μπορεί να δημιουργεί διαγράμματα στατιστικών.

### 2.2.1 Αφήγηση σεναρίου Δημιουργίας Διαγράμματος

Ο χρήστης θα πρέπει να έχει συνδεθεί στην εφαρμογή και να βρίσκεται στην αρχική σελίδα. Από εκεί επιλέγει τον σύνδεσμο «Δημιουργία Διαγράμματος». Τότε καλείται η συνάρτηση `createDiagram()` του `DiagramHandlerController`, και εμφανίζεται η πρώτη φόρμα συμπλήρωσης των παραμέτρων του διαγράμματος. Ο χρήστης συμπληρώνει τα πεδία και επιλέγει τον σύνδεσμο «Επόμενο». Τότε εμφανίζεται η δεύτερη φόρμα συμπλήρωσης των παραμέτρων του διαγράμματος. Αφού ο χρήστης συμπληρώσει τα πεδία της δεύτερης φόρμας έχει δύο επιλογές, «Άκυρο» και «Διάγραμμα». Στην πρώτη περίπτωση καλείται η συνάρτηση `checkEnteredData()` του `DiagramHandlerController`, προκειμένου να ελέγξει εάν έχουν συμπληρωθεί όλα τα στοιχεία. Στη δεύτερη περίπτωση, δηλαδή αν ο χρήστης επιλέξει τον σύνδεσμο «Άκυρο», τότε προβάλλεται η αρχική σελίδα του συστήματος. Εάν ο χρήστης έχει συμπληρώσει όλα τα πεδία για τη δημιουργία του διαγράμματος και πατήσει τον σύνδεσμο «Διάγραμμα», τότε δημιουργείται ένα αντικείμενο `DiagramPrEntity` που περιέχει τις παραμέτρους του διαγράμματος





και μέσω του GetDataProxy λαμβάνεται ένα αντικείμενο DiagramEntity και καλείται η συνάρτηση isThereData() για να ελέγξει εάν υπάρχουν τα δεδομένα που ζήτησε ο χρήστης για την δημιουργία του διαγράμματος. Σε περίπτωση που ο χρήστης δεν έχει συμπληρώσει όλα τα πεδία για την δημιουργία του διαγράμματος τότε προβάλλεται μήνυμα σφάλματος και έπειτα, μετά από πάτημα του πλήκτρου «Πίσω» από το χρήστη ξαναπροβάλλεται η πρώτη φόρμα συμπλήρωσης των παραμέτρων του διαγράμματος. Εάν υπάρχουν τα απαραίτητα δεδομένα για την δημιουργία του διαγράμματος, τότε μέσω του CreateDiagramProxy δημιουργείται το ζητούμενο διάγραμμα. Εάν δεν υπάρχουν τα δεδομένα που ζήτησε ο χρήστης τότε εμφανίζεται αντίστοιχο μήνυμα σφάλματος και μετά από πάτημα του πλήκτρου «Πίσω» από τον χρήστη προβάλλεται η αρχική σελίδα του συστήματος.

Εικόνα 8: Διάγραμμα ακολουθιών σεναρίου δημιουργίας διαγράμματος

## 2.3 Πακέτο αναζήτησης δεδομένων και αποθήκευσης δεδομένων

Το πακέτο αυτό περιλαμβάνει την αρχική σελίδα και τις σελίδες αναζήτησης δεδομένων, προβολής των αποτελεσμάτων, και τη σελίδα αποθήκευσης αυτών στον προσωπικό υπολογιστή του χρήστη. Ο χρήστης μπορεί να επιλέξει αναζήτηση δεδομένων συμπληρώνοντας τα κριτήρια αναζήτησης. Επίσης έχει την επιλογή να κάνει αποθήκευση των δεδομένων, για τα οποία έκανε αναζήτηση, στον



προσωπικό του υπολογιστή.

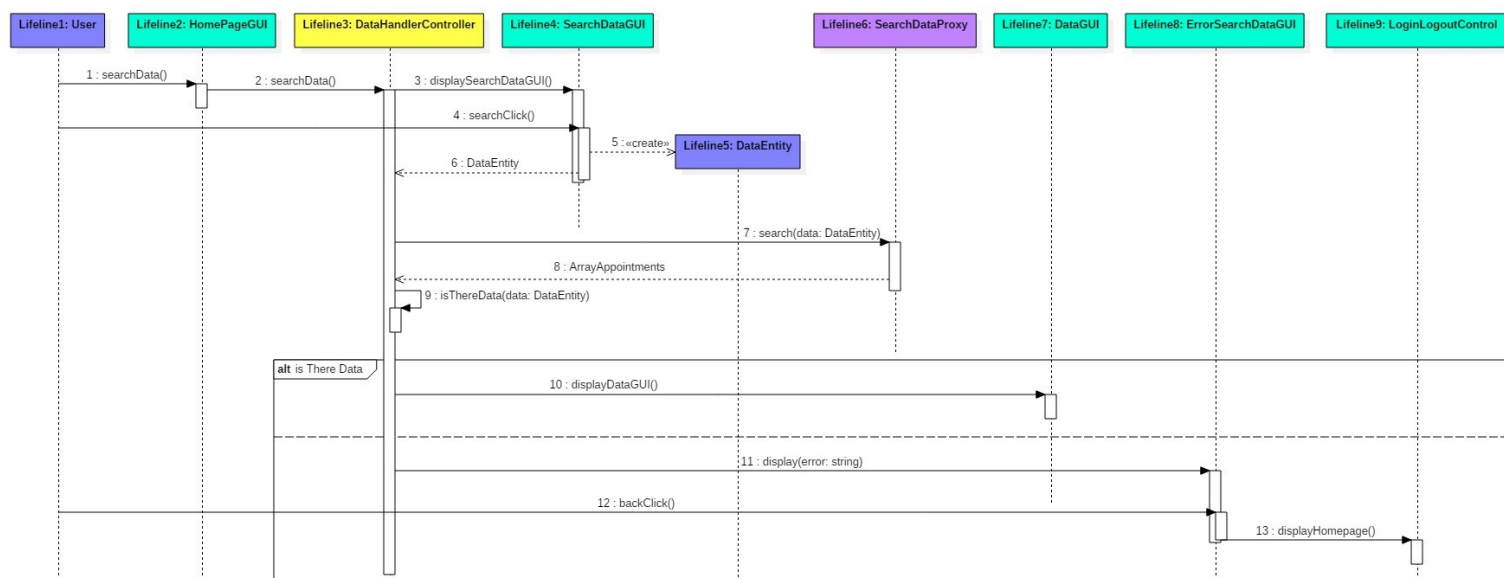
Οι λειτουργικές απαιτήσεις που σχετίζονται με αυτό το πακέτο χρήσης είναι:

- <ΛΑ- 4> Ο χρήστης θα πρέπει να μπορεί να κάνει αναζήτηση δεδομένων.
- <ΛΑ- 3> Ο χρήστης θα πρέπει να μπορεί να αποθηκεύει δεδομένα.

### 2.3.1 Αφήγηση σεναρίου Αναζήτησης Δεδομένων

Ο χρήστης θα πρέπει να έχει συνδεθεί στο σύστημα και να βρίσκεται στην αρχική σελίδα. από εκεί επιλέγοντας το σύνδεσμο «Αναζήτηση δεδομένων», καλείται η συνάρτηση `searchData()` του `DataHandlerController`, και προβάλλεται η φόρμα συμπλήρωσης των πεδίων με τα κριτήρια αναζήτησης. Εκεί ο χρήστης έχει τη δυνατότητα να αφήσει τις default επιλογές ή να τις αλλάξει και στη συνέχεια να πατήσει τον σύνδεσμο «Αναζήτηση». Αφού το κάνει αυτό δημιουργείται ένα αντικείμενο τύπου `DataEntity` και μέσω του `SearchDataProxy` γίνεται η αναζήτηση δεδομένων. Στη συνέχεια καλείται η συνάρτηση `isThereData()` προκειμένου να ελέγξει αν υπάρχουν δεδομένα σύμφωνα με τα κριτήρια που ζήτησε ο χρήστης. Σε περίπτωση που υπάρχουν προβάλλεται η σελίδα με τα δεδομένα. Σε περίπτωση που δεν υπάρχουν εμφανίζεται μήνυμα σφάλματος και μετά από πάτημα του πλήκτρου «Πίσω» από τον χρήστη,

Εικόνα 9: Διάγραμμα ακολουθιών σεναρίου αναζήτησης δεδομένων



προβάλλεται η αρχική σελίδα του συστήματος.

### 2.3.2 Αφήγηση σεναρίου Αποθήκευσης Δεδομένων

Ο χρήστης θα πρέπει να έχει κάνει αναζήτηση δεδομένων και να βρίσκεται στη σελίδα προβολής αυτών. Εκεί έχει σαν επιλογή είτε να πατήσει τον σύνδεσμο «Αποθήκευση Δεδομένων», είτε να πατήσει τον σύνδεσμο «Άκυρο». Στην πρώτη περίπτωση καλείται η συνάρτηση `saveData()` του `DataHandlerController` και





## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

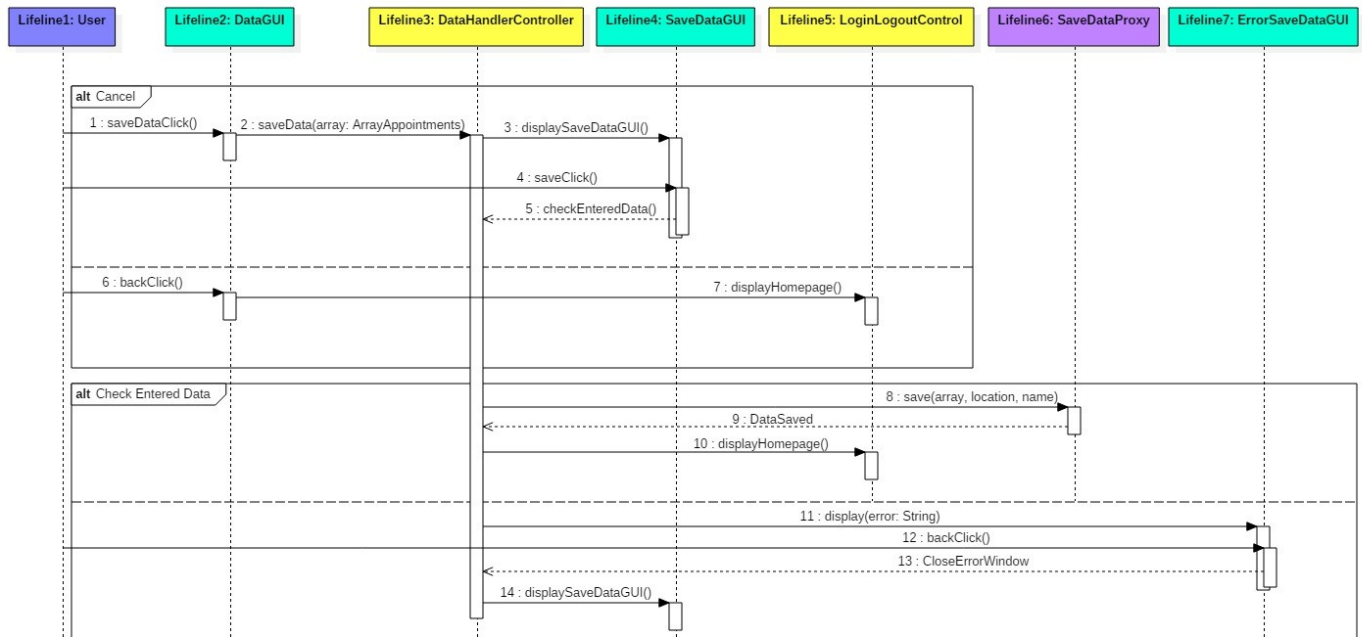
Άνοιξη  
2017

---

προβάλλεται η σελίδα συμπλήρωσης του φακέλου αποθήκευσης και του ονόματος με το οποίο θα αποθηκευτούν τα δεδομένα. Μόλις ο χρήστης πατήσει τον σύνδεσμο «Αποθήκευση», καλείται η συνάρτηση `checkEnteredData()` προκειμένου να ελέγξει εάν έχουν συμπληρωθεί όλα τα πεδία της φόρμας. Στην δεύτερη περίπτωση, όπου ο χρήστης επιλέγει τον σύνδεσμο «Άκυρο» προβάλλεται η αρχική σελίδα του συστήματος. Εάν ο χρήστης έχει συμπληρώσει όλα τα πεδία της φόρμας αποθήκευσης δεδομένων, τότε αυτά αποθηκεύονται μέσω του `SaveDataProxy` και έπειτα προβάλλεται η αρχική σελίδα του συστήματος. Εάν ο χρήστης δεν έχει συμπληρώσει όλα τα πεδία της φόρμας αποθήκευσης δεδομένων τότε προβάλλεται μήνυμα σφάλματος, και μετά από πάτημα του συνδέσμου «Πίσω» από τον χρήστη προβάλλεται εκ νέου η σελίδα αποθήκευσης δεδομένων.



Εικόνα 10: Διάγραμμα ακολουθιών σεναρίου αποθήκευσης δεδομένων



## 2.4 Πακέτο Δημιουργίας και επεξεργασίας ερωτηματολογίων από το χρήστη

Το πακέτο αυτό περιλαμβάνει την σελίδα δημιουργίας και τροποποίησης ερωτηματολογίων που βλέπει ο χρήστης όταν επιλέγει τον αντίστοιχο σύνδεσμο. Ο χρήστης μπορεί να επιλέξει να δημιουργήσει ένα ερωτηματολόγιο, να διαγράψει ένα ερωτηματολόγιο από το σύστημα, να το επεξεργαστεί και μετά την δημιουργία του καθώς και να μπορεί να δεί τα αποτελέσματα του.

Οι λειτουργικές απαιτήσεις που σχετίζονται με αυτό το πακέτο χρήσης:

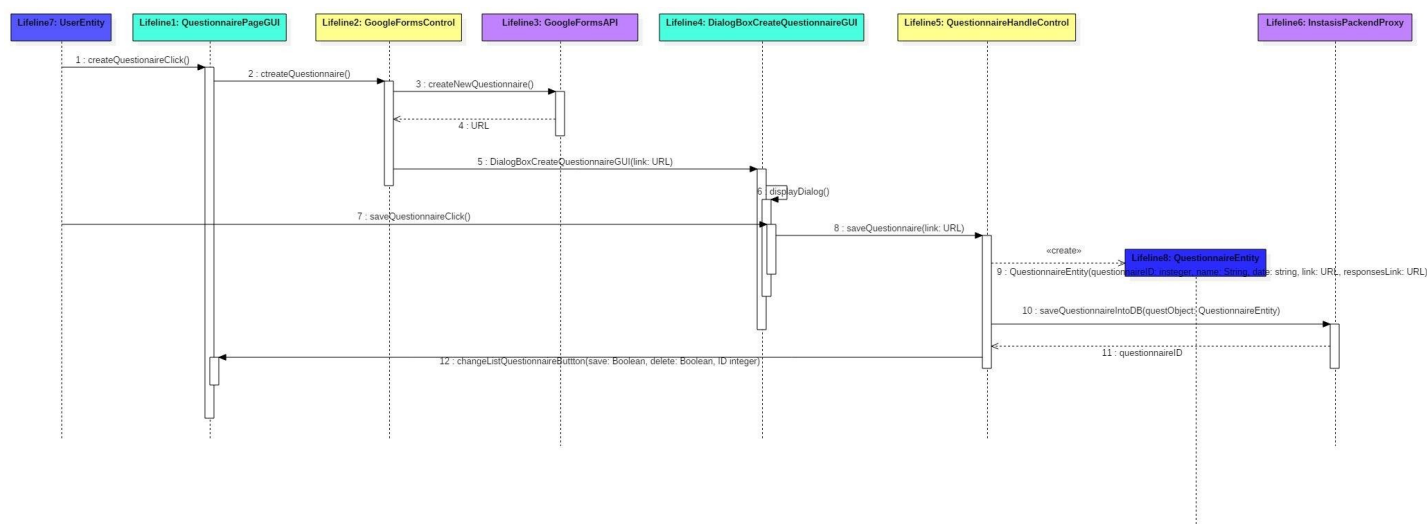
- <ΛΑ- 6> Ο χρήστης θα πρέπει να μπορεί να συντάσσει ερωτηματολόγιο.
- <ΛΑ- 7> Ο χρήστης θα πρέπει να μπορεί να διαχειρίζεται τα ερωτηματολόγια.

### 2.4.1 Αφήγηση σεναρίου Δημιουργίας ερωτηματολογίου

Ο χρήστης έχει συνδεθεί στο σύστημα και βρίσκεται στην σελίδα ερωτηματολογίων. Από την σελίδα ερωτηματολογίων πατάει το κουμπί “δημιουργία νέου” και ενεργοποιείται η μέθοδος `createQuestionnaireClick()` της σελίδας ερωτηματολογίων δηλαδή της κλάσης `QuestionnairePageGUI`. Στην συνέχεια η παραπάνω μέθοδος καλεί την μέθοδο `createQuestionnaire()` του ελεγκτή και αυτή με την σειρά της μέσω της συνάρτησης `createNewQuestionnaire()`



οδηγεί τον χρήστη στην εφαρμογή Google Forms για να συντάξει ένα ερωτηματολόγιο. Όταν ο χρήστης τελειώσει την σύνταξη η προηγούμενη συνάρτηση επιστρέφει τον ηλεκτρονικό σύνδεσμο που οδηγεί στο ερωτηματολόγιο της εφαρμογής Google Forms στο ελεγκτή `GoogleFormsControl` ο οποίος δημιουργεί μία διεπαφή επιβεβαίωσης της αποθήκευσης στο τύπο της κλάσης `DialogBoxCreateQuestionnaireGUI` μέσα από την οποία ο χρήστης επιλέγει να αποθηκεύσει το ερωτηματολόγιο καλώντας έτσι την συνάρτηση `saveQuestionnaireClick()`. Στην συνέχεια καλείται ο ελεγκτής `QuestionnaireHandleControl` και αφού πάρει τον ηλεκτρονικό σύνδεσμο μέσω της μεθόδου δημιουργεί ένα ερωτηματολόγιο στο τύπο της κλάσης `QuestionnaireEntity` και αποθηκεύει σε αυτό τον ηλεκτρονικό σύνδεσμο. Στην συνέχεια καλεί την συνάρτηση `saveQuestionnaireIntoDB(questObject:QuestionnaireEntity)` της οριακής κλάσης `IstasisPackendProxy` για να αποθηκεύσει στο σύστημα `IstasisBackend` το αντικείμενο ερωτηματολογίου και μετά η συνάρτηση επιστρέφει στον ελεγκτή τον μοναδικό κωδικό `questionnaireID` του ερωτηματολογίου. Ο ελεγκτής χρησιμοποιεί τον κωδικό για να καλέσει την μέθοδο `changeListQuestionnaireButton(save:Boolean, delete:Boolean, ID integer)` η οποία θα ανανεώσει την λίστα ερωτηματολογίων που προβάλλεται εκείνη την στιγμή από την σελίδα ερωτηματολογίων.



Εικόνα 11: Διάγραμμα ακολουθιών σεναρίου δημιουργίας ερωτηματολογίου

**Σημείωση:** Στο παραπάνω σενάριο για λόγους απλότητας στα πλαίσια της εργασίας δεν έχουμε συμπεριλάβει τις περιπτώσεις όπου ο χρήστης όταν εμφανίζεται η διεπαφή για την επιβεβαίωση της αποθήκευσης του ερωτηματολογίου μπορεί να επιλέξει να ξανά επεξεργαστεί το ερωτηματολόγιο ή να το απορρίψει.



#### **2.4.2 Αφήγηση σεναρίου Διαγραφής ερωτηματολογίου**

Ο χρήστης βρίσκεται στην σελίδα ερωτηματολογίων στην οποία φαίνεται η λίστα με τα ερωτηματολόγια που υπάρχουν. Ο χρήστης πληκτρολογεί πάνω σε ένα από τα ερωτηματολόγια και ενεργοποιείται η συνάρτηση `editQuestionnaireClick(questionnaireID:integer)` η οποία με την σειρά της ενεργοποιεί τον ελεγκτή `QuestionnaireHandleControl` για να επικοινωνήσει με την κλάση ορίου `InstasisPackendProxy` για να ζητήσει από το σύστημα `InstasisBackend` να επιστρέψει πίσω στον ελεγκτή το αντικείμενο τύπου ερωτηματολογίου `QuestionnaireEntity`. Στην συνέχεια ο ελεγκτής ανοίγει μία διεπαφή τύπου `DialogBoxHandleQuestionnaireGUI` από την οποία ο χρήστης επιλέγει "Διαγραφή Ερωτηματολογίου" και ενεργοποιείται η μέθοδος `QuestionnaireDeleteClick()`. Η μέθοδος αυτή έχει δύο στόχους, αρχικά να διαγράψει το ερωτηματολόγιο από την εξωτερική εφαρμογή `Google Forms` και δεύτερον να διαγράψει την καταχώρηση του ηλεκτρονικού συνδέσμου του ερωτηματολογίου από το σύστημα `InstasisBackend`. Για τον πρώτο από τους δύο στόχους ενεργοποιεί την συνάρτηση `deleteQuestionnaire(link:URL)` του ελεγκτή `GoogleFormsControl` ο οποίος καλεί την κλάση `GoogleFormsAPI` για να διαγράψει το ερωτηματολόγιο από το εξωτερικό σύστημα. Για να διαγράψει την καταχώρηση του ερωτηματολογίου από το σύστημα `InstasisBackend` καλεί την συνάρτηση `deleteQuestionnaire(questionnaireID:integer)` του ελεγκτή `QuestionnaireHandleControl` η οποία μέσω της μεθόδου `deleteQuestionnaireIntoDB(questionnaireID:integer)` επικοινωνεί με την κλάση ορίου `InstasisPackendProxy` για να διαγράψει το αντικείμενο ερωτηματολογίου που βρίσκεται στο σύστημα `InstasisBackend`. Τέλος ο ελεγκτής `QuestionnaireHandleControl` καλεί την συνάρτηση `changeListQuestionnaireButton(save:Boolean,delete:Boolean,ID integer)` της σελίδας ερωτηματολογίων για ανανεώσει την λίστα των ερωτηματολογίων που φαίνονται εκείνη την στιγμή στην οθόνη του χρήστη.



## Τεχνολογία Λογισμικού

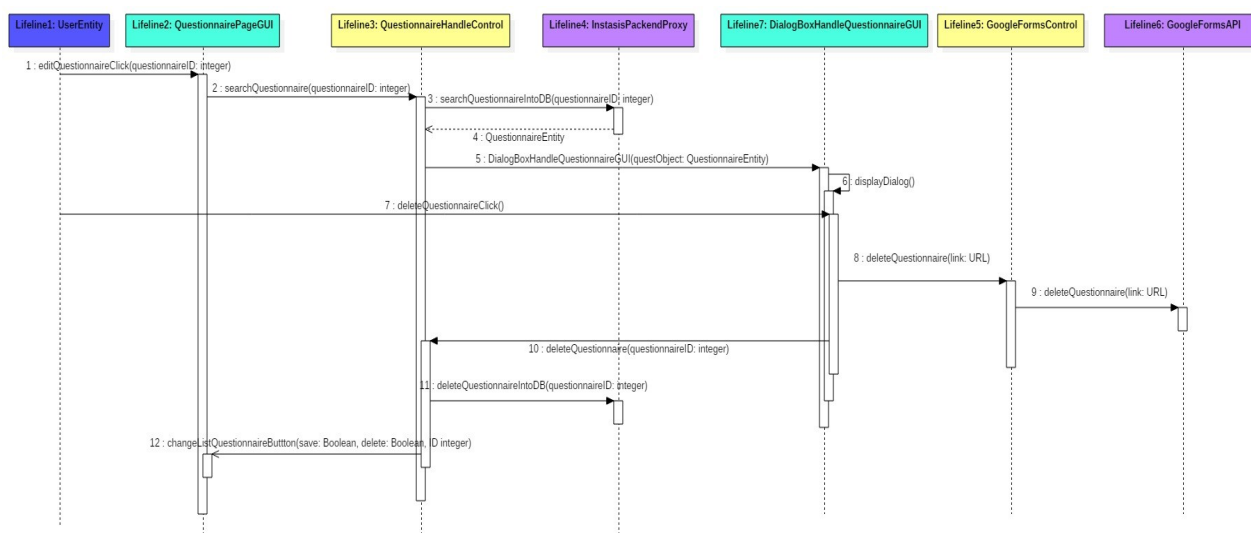
Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017



Εικόνα 12: Διάγραμμα ακολουθιών σεναρίου διαγραφής ερωτηματολογίου



## 3 Μη λειτουργικές απαιτήσεις

### 3.1 Απαιτήσεις Χρηστικότητας (Usability)

#### <ΜΛΑ- 1>

Το σύστημα θα πρέπει να υποστηρίζει την αγγλική και την ελληνική γλώσσα.

**Περιγραφή:** Όλες οι διαθέσιμες επιλογές και τα διαγράμματα θα πρέπει να υπάρχει η δυνατότητα εμφάνισης τους σε δύο διαφορετικές γλώσσες την ελληνική και την αγγλική.

**User Priority (5/5):** Είναι ιδιαίτερα σημαντικό ο χρήστης να μπορεί καθώς πλοηγείται στην εφαρμογή να κατανοεί το περιεχόμενό της.

**Technical Priority (5/5):** Είναι ανεξάρτητο από τις υπόλοιπες λειτουργίες και δε παίζει σημαντικό ρόλο στην ομαλή λειτουργία του συστήματος.

**Stability (5/5):** Η ανάγκη υποστήριξης και της αγγλικής γλώσσας θα παραμείνει και στο μέλλον καθώς τα αγγλικά είναι η διεθνής γλώσσα και λόγω της εξωστρέφειας του πανεπιστημίου.

### 3.2 Τεχνικές Απαιτήσεις περιβάλλοντος

#### <ΜΛΑ- 2>

Το σύστημα θα πρέπει να υποστηρίζει εικόνες jpeg.

**Περιγραφή:** Το σύστημα θα πρέπει να μπορεί να αποθηκεύσει τα παραγόμενα διαγράμματα σε μορφή jpeg. Επίσης, θα πρέπει να είναι δυνατό να ανοίξουμε jpeg για να προβάλουμε διαγράμματα και να στείλουμε εικόνες jpeg στο instasis-backend.

**User Priority (5/5):** Είναι πολύ σημαντική λειτουργία για τον χρήστη, καθώς με αυτόν το τρόπο θα μπορεί να δει διαγράμματα, να τα αποθηκεύει και να τα αποστέλλει.

**Technical Priority (4/5):** Είναι σημαντική λειτουργία καθώς αυτή εμπλέκεται και με το instasis-backend μέσω της αποστολής διαγραμμάτων.

**Stability (4/5):** Ίσως αυτή η απαίτηση να μεταβληθεί στο μέλλον με την ανάπτυξη νέων τεχνικών απεικόνισης των διαγραμμάτων.

#### <ΜΛΑ- 3>

Το σύστημα θα πρέπει να είναι συμβατό με matlab.

**Περιγραφή:** Το σύστημα θα πρέπει να μπορεί να επικοινωνεί με την εφαρμογή matlab, προκειμένου να χρησιμοποιεί συναρτήσεις για την δημιουργία διαγραμμάτων και την επεξεργασία στατιστικών δεδομένων.



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017

**User Priority (5/5):** Η εφαρμογή του matlab είναι αρκετά γνωστή και εύκολα διατηρήσιμη από έναν χρήστη. Ακόμη προσφέρει πολλές δυνατότητες για ανάλυση και επεξεργασία δεδομένων, ενώ μέσω του πανεπιστημίου δίνεται δυνατότητα για απομακρυσμένη χρήση της εφαρμογής, δηλαδή για χρήση της εφαρμογής μέσω διαδικτύου ακόμη και όταν δεν είναι εγκατεστημένη στον υπολογιστή που θα κάνει επεξεργασία δεδομένων

**Technical Priority (5/5):** Η συμβατότητα του συστήματος με το matlab είναι πολύ υψηλής προτεραιότητας για την λειτουργία του συστήματος, καθώς ολόκληρη η λειτουργία του βασίζεται στην επεξεργασία δεδομένων.

**Stability (4/5):** Ίσως αυτή η απαίτηση να μεταβληθεί στο μέλλον με την χρήση κάποιου άλλου προγράμματος επεξεργασίας των δεδομένων, αν και κάτι τέτοιο δε θεωρείται πιθανό.

### <ΜΛΑ- 4>

Το σύστημα θα πρέπει να είναι συμβατό με microsoft excel.

**Περιγραφή:** Η παρουσίαση των δεδομένων με τη μορφή πίνακα πρέπει να μπορεί να γίνεται με τη χρήση microsoft excel.

**User Priority (5/5):** Η απεικόνιση των δεδομένων με χρήση του προγράμματος excel είναι πολύ υψηλής προτεραιότητας για τον χρήστη καθώς με τον τρόπο αυτό μπορεί να ερμηνεύει με γρήγορο και απλό τρόπο τα αποτελέσματα.

**Technical Priority (5/5):** Η συμβατότητα του συστήματος με το πρόγραμμα microsoft excel είναι σημαντικής προτεραιότητας για το σύστημα καθώς αποτελεί το μοναδικό τρόπο απεικόνισης των δεδομένων.

**Stability (4/5):** Ίσως αυτή η απαίτηση να μεταβληθεί στο μέλλον με τη χρήση άλλου προγράμματος απεικόνισης των δεδομένων.

### <ΜΛΑ- 5>

Το σύστημα θα πρέπει να είναι συμβατό με την υπηρεσία Google forms.

**Περιγραφή:** Η υπηρεσία google docs παρέχει την εφαρμογή google forms η οποία μας δίνει την δυνατότητα να δημιουργούμε ηλεκτρονικά ερωτηματολόγια τα οποία είναι online στο διαδίκτυο. Η εφαρμογή αυτή θα δώσει στο σύστημα μας την δυνατότητα για την δημιουργία ερωτηματολογίων με απλό και εύχρηστο τρόπο.

**User Priority (4/5):** Για τον χρήστη η εφαρμογή αυτή είναι αρκετά απλή εύχρηστη και αποτελεσματική για τον βοηθήσει στην δημιουργία και την σύνταξη των ερωτηματολογίων του.

**Technical Priority (3/5):** Για το σύστημα η εφαρμογή δεν είναι πολύ σημαντική αφού δεν επηρεάζει τις λειτουργίες του παρα μόνο του δίνει επιπρόσθετες ιδιότητες.

**Stability (3/5):** Είναι πιθανό αυτή η απαίτηση να μεταβληθεί στο μέλλον καθώς η υπηρεσία Google forms είναι μια εφαρμογή της Google που συνεχώς εξελίσσεται και τροποποιείται οπότε δεν είναι σίγουρο ότι θα είναι πάντα το ίδιο αποτελεσματική για τη συγκεκριμένη εφαρμογή.





## **Τεχνολογία Λογισμικού**

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπο-  
λογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη  
2017



---

## Παράρτημα Ι - Πίνακας Ιχνηλασιμότητας

- Πακέτο **UserHandle**

Λειτουργικές απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-1>

Σενάρια χρήσης στα οποία απευθύνεται: <ΣΧ-1>, <ΣΧ-2>

- Πακέτο **DataAnalysis**

Λειτουργικές απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-9>, <ΛΑ-10>, <ΛΑ-11>, <ΛΑ-12>

Σενάρια χρήσης στα οποία απευθύνεται: <ΣΧ-5>, <ΣΧ-9>, <ΣΧ-10>, <ΣΧ-11>

- Πακέτο **DiagramHandle**

Λειτουργικές απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-2>, <ΛΑ-5>, <ΛΑ-8>

Σενάρια χρήσης στα οποία απευθύνεται: <ΣΧ-3>, <ΣΧ-8>

- Πακέτο **DataHandle**

Λειτουργικές απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-3>, <ΛΑ-4>

Σενάρια χρήσης στα οποία απευθύνεται: <ΣΧ-4>

- Πακέτο **QuestionnaireHandle**

Λειτουργικές απαιτήσεις στις οποίες απευθύνεται: <ΛΑ-6>, <ΛΑ-7>

Σενάρια χρήσης στα οποία απευθύνεται: <ΣΧ-6>, <ΣΧ-7>



## **Παράρτημα II - Ανοιχτά Θέματα**

- Σε κάποιες Οριακές κλάσεις που αποτελούν διεπαφές έχουν παραληφθεί κάποια γραφικά στοιχεία όπως το λογότυπο της εφαρμογής. Έχει δοθεί το βάρος σε στοιχεία που προσδίδουν λειτουργικότητα (κουμπιά, επιλογές κλπ...)
- Κατά τη φάση της υλοποίησης του συστήματος είναι πολύ πιθανόν κάποιες από τις κλάσεις να αλλάξουν όπως και κάποιες παραδοχές που έχουν γίνει. Αυτό είναι πιθανό καθώς το παρόν έγγραφο αποτελεί μια πρώιμη σχεδίαση των κλάσεων του συστήματος.
- Σε ορισμένες πολύ μεμονωμένες περιπτώσεις κατά την υλοποίηση του παρόντος εγγράφου δεν λήφθηκαν υπ' όψιν ακριβώς τα ενδεικτικά παράθυρα διεπαφής που ορίστηκαν σε προηγούμενο έγγραφο καθώς ο αρχικός σχεδιασμός δυσχεραίνει την παρούσα σχεδιαστική προσπάθεια. Είναι προτεραιότητα της ομάδας ανάπτυξης να μεταβάλλει το προηγούμενο έγγραφο σύμφωνα με τα νέα δεδομένα.
- Έχουμε θεωρήσει δεδομένο ότι το εξωτερικό σύστημα InstasisBackend λειτουργεί με έναν συγκεκριμένο τρόπο και παρέχει συγκεκριμένους τύπους δεδομένων σε κάποιες περιπτώσεις. Μένει να ενοποιηθεί η λογική λειτουργίας των 2 αυτών συστήματος πριν την ένωσή τους.