



Τεχνολογία Λογισμικού
Τομέας Ηλεκτρονικής και Υπολογιστών
Τμήμα ΗΜΜΥ
Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2017



Σχεδίαση Συστήματος

Del.3.1

Έκδοση 1.0

Μπεκιάρης Θεοφάνης(8200)
Πεγλιβάνης Νίκος (8228)
Κελεσάκης Δημήτριος (7898)
Μπούζος Ραφαήλ (7945)

theompek@auth.gr
nikopech@auth.gr
dkelesak@auth.gr
mprouzos@auth.gr

20/05/2017

Ιστορικό Αλλαγών

Όνομα	Ημ/νία	Περιγραφή Αλλαγής	Εκδ.
Α. Συμεωνίδης	29/05/2009	Δημιουργία Εγγράφου Προσαρμογή του ESA software engineering standards guidelines (1991) και του εγγράφου SDD document, από τους Bruegge και Dutoit (2004).	0.1
Reporter007	11/05/2017	Συγγραφή κεφαλαίων 1 και 2	0.2
Reporter007	14/05/2017	Συγγραφή κεφαλαίου 3	0.3
Reporter007	17/05/2017	Συγγραφή κεφαλαίου 4	0.4
Reporter007	19/05/2017	Διόρθωση κεφαλαίων 1,2,3 και προσθήκες στο κεφάλαιο 4	0.5
Reporter007	20/05/2017	Προσθήκη ανοικτών θεμάτων, και μορφοποίηση του εγγράφου	1.0

Μέλη Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Μπεκιάρης Θεοφάνης	Reporter007	theompek@auth.gr
Πεχλιβάνης Νίκος	Reporter007	nikopech@auth.gr
Κελεσάκης Δημήτριος	Reporter007	dkelesak@auth.gr
Μπρούζος Ραφαήλ	Reporter007	mprouzos@auth.gr

Πίνακας Περιεχομένων

1. Τρέχουσα Αρχιτεκτονική Λογισμικού	5
1.1 Αρχιτεκτονική Πελάτη-Εξυπηρετητή (Client-Server)	5
1.2 Αρχιτεκτονική τριών επιπέδων (Three tier)	6
2. Προτεινόμενη Αρχιτεκτονική Λογισμικού	9
2.1 Αποδόμηση Συστήματος	9
2.1.1 Υποσύστημα LoginHandler	9
2.1.2 Υποσύστημα LoginGUI	9
2.1.3 Υποσύστημα CreateUploadDiagramHandler	10
2.1.4 Υποσύστημα CreateUploadDiagramGUI	10
2.1.5 Υποσύστημα SearchSaveDataHandler	11
2.1.6 Υποσύστημα SearchSaveDataGUI	11
2.1.7 Υποσύστημα QuestionnaireHandler	12
2.1.8 Υποσύστημα QuestionnaireGUI	12
2.1.10 Υποσύστημα PredictionGUI	13
2.1.11 Υποσύστημα DataProvider	14
2.2 Απεικόνιση Υλικού/Λογισμικού	16
2.2.1 Client Device	17
2.2.2 System Server	17
2.2.3 Συνολικό διάγραμμα ανάπτυξης	18
2.3 Έλεγχος Πρόσβασης και Ασφάλεια	20
3. Προδιαγραφές Τμηματικού Σχεδιασμού (Component Design Specifications)	23
3.1 Πρότυπα Σχεδιασμού που υιοθετήθηκαν	23
3.1.1 Δομικά πρότυπα	23
4. Πίνακας ιχνηλασιμότητας εγγράφων Σχεδίασης και Απαιτήσεων Λογισμικού	28
5. Παράρτημα I – Ανοιχτά Θέματα	29

Λίστα Σχημάτων

Σχήμα 1: Αναπαράσταση αρχιτεκτονικής πελάτη-διακομιστή	5
Σχήμα 2: Αναπαράσταση αρχιτεκτονικής τριών επιπέδων	6
Σχήμα 3: Αναπαράσταση αρχιτεκτονικής τριών επιπέδων	8
Σχήμα 4: Υποσύστημα LoginHandler	9
Σχήμα 5: Υποσύστημα LoginGUI	9
Σχήμα 6: Υποσύστημα CreateUploadDiagramHandler	10
Σχήμα 7: Υποσύστημα CreateUploadDiagramGUI	10
Σχήμα 8: Υποσύστημα SearchSaveDataHandler	11
Σχήμα 9: Υποσύστημα SearchSaveDataGUI	11
Σχήμα 10: Υποσύστημα QuestionnaireHandler	12
Σχήμα 11: Υποσύστημα QuestionnaireGUI	13
Σχήμα 12: Υποσύστημα PredictionHandler	13
Σχήμα 13: Υποσύστημα PredictionGUI	14
Σχήμα 14: Υποσύστημα Databases	15
Σχήμα 15: Συνολικό διάγραμμα τμημάτων	16
Σχήμα 16: Client Device	17
Σχήμα 17: System Server	18
Σχήμα 18: Διάγραμμα ανάπτυξης	19
Σχήμα 20: Το πρότυπο Proxy	24
Σχήμα 19: Το πρότυπο Proxy όπως εφαρμόστηκε στο υποσύστημα DiagramHandle	24
Σχήμα 22: Το πρότυπο Bridge	26
Σχήμα 21: Το πρότυπο Bridge όπως εφαρμόστηκε στο υποσύστημα DataAnalysisHandle	26
Σχήμα 23: Το πρότυπο Bridge όπως εφαρμόστηκε στο υποσύστημα DiagramHandle	27

1. Τρέχουσα Αρχιτεκτονική Λογισμικού

Το σύστημα reporter007 βασίζεται στην συνεχή επικοινωνία του χρήστη καθώς και του συστήματος Instasis-backend με αυτήν. Συνεπώς, εξετάζοντας τα πλεονεκτήματα των διαθέσιμων αρχιτεκτονικών, βρέθηκε ότι καταλληλότερη αρχιτεκτονική για το σύστημα είναι αυτή του πελάτη-εξυπηρετητή 3 επιπέδων (Three-tier Client-Server).

1.1 Αρχιτεκτονική Πελάτη-Εξυπηρετητή (Client-Server)

Η αρχιτεκτονική πελάτη-εξυπηρετητή (ή πελάτη-διακομιστή) είναι μία ιδιαίτερα γνωστή μέθοδος ανάπτυξης λογισμικού, στην οποία ο πελάτης (που αποτελεί ένα τμήμα λογισμικού) ζητά κάτι, και ο εξυπηρετητής (ένα άλλο τμήμα λογισμικού) του το επιστρέφει. Με άλλα λόγια, το σύστημα (server) δέχεται αιτήσεις για τις υπηρεσίες του από μια οντότητα (client), εκτός συστήματος, και προωθεί τις απαντήσεις στην οντότητα αυτή.



Σχήμα 1: Αναπαράσταση αρχιτεκτονικής πελάτη-διακομιστή

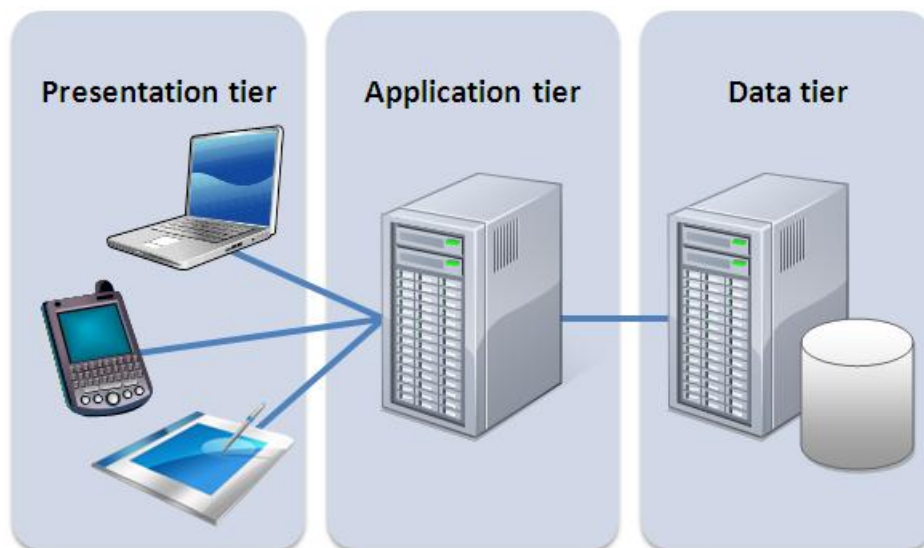
Πιο συγκεκριμένα, οι πελάτες και οι εξυπηρετητές επικοινωνούν μέσω ενός δικτύου υπολογιστών. Ένας εξυπηρετητής τρέχει ένα ή περισσότερα προγράμματα τα οποία

μοιράζονται τους πόρους με του πελάτες. Ως εκ τούτου οι πελάτες επικοινωνούν με τους εξυπηρετητές και αξιοποιώντας τους πόρους των δεύτερων, στέλνουν αιτήσεις και λαμβάνουν άμεσα συνήθως απαντήσεις σε αυτές.

Ένα σημαντικό χαρακτηριστικό της εν λόγω αρχιτεκτονικής είναι ότι ο εξυπηρετητής είναι σχεδιασμένος ώστε να ανταποκρίνεται στα αιτήματα πολλών πελατών. Αυτό δίνει την δυνατότητα γρήγορης και αποδοτικής εξυπηρέτησης. Εκτός αυτού ένα εξίσου σημαντικό στοιχείο της αρχιτεκτονικής πελάτη-εξυπηρετητή, είναι η εύκολη διαχείριση και μεταφερισιμότητά της, που είναι και οι κύριοι λόγοι που αυτή η αρχιτεκτονική είναι αρκετά διαδεδομένη. Τέλος η αρχιτεκτονική αυτή προσφέρει ευελιξία στην διεπαφή με τον χρήστη, υψηλή απόδοση του συστήματος, και αξιοπιστία.

1.2 Αρχιτεκτονική τριών επιπέδων (Three tier)

Η αρχιτεκτονική τριών επιπέδων είναι ένας τύπος αρχιτεκτονικής στην οποία το σύστημα χωρίζεται σε τρία επίπεδα. Κάθε επίπεδο αποτελεί ένα ξεχωριστό υποσύστημα και εξυπηρετεί διαφορετικό σκοπό. Αυτό έχει σαν συνέπεια την ανεξαρτησία μεταξύ των επιπέδων, πράγμα που σημαίνει ότι μπορεί ένα επίπεδο να μεταβληθεί εντελώς χωρίς να επηρεαστούν όμως τα υπόλοιπα. Ακόμη είναι δυνατή η προσθήκη ενός επιπλέον επιπέδου αφήνοντας πάλι ανεπηρέαστα τα υπόλοιπα.



Σχήμα 2: Αναπαράσταση αρχιτεκτονικής τριών επιπέδων

Ένα χαρακτηριστικό της αρχιτεκτονικής αυτής είναι ότι κάθε επίπεδο ζητά υπηρεσίες και πληροφορίες από το προηγούμενό του, και παρέχει υπηρεσίες και πληροφορίες στο επόμενο. Με αυτή τη λογική τα τρία επίπεδα που συναντώνται στην αρχιτεκτονική διαμορφώνονται ως εξής:

❖ **Πρώτο Επίπεδο - Database Server:**

Το πρώτο επίπεδο αποτελεί ουσιαστικά την βάση δεδομένων του συστήματος και είναι υπεύθυνο για την παροχή όλων των απαραίτητων λειτουργιών που εξυπηρετούν την αποθήκευση, ανάκτηση, ενημέρωση και συντήρηση των δεδομένων του συστήματος.

❖ **Δεύτερο Επίπεδο - Application Server:**

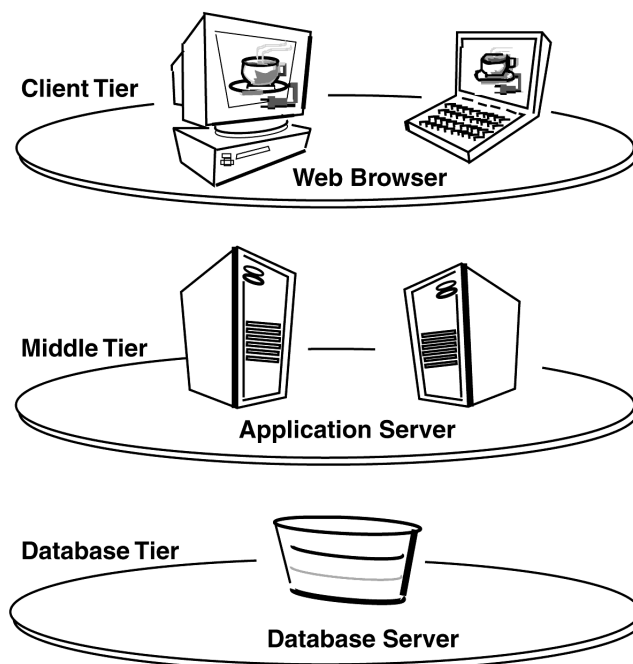
Το δεύτερο επίπεδο αποτελεί ουσιαστικά τον κορμό της εφαρμογής, καθώς είναι αυτό που ελέγχει τη σωστή λειτουργία του συστήματος. Σε αυτό το επίπεδο περιλαμβάνονται όλοι οι απαραίτητοι ελεγκτές για την παροχή των υπηρεσιών της εφαρμογής στους πελάτες. Κάθε Application Server ανταποκρίνεται σε συγκεκριμένο αριθμό αιτημάτων των πελατών επιτυγχάνοντας με αυτό τον τρόπο την αποσυμφόρηση του συνολικού φόρτου του συστήματος.

❖ **Τρίτο Επίπεδο – Client:**

Το τρίτο επίπεδο του λογισμικού αποτελεί τη επαφή του χρήστη με το σύστημα (User Interface). Στο επίπεδο αυτό, περιλαμβάνονται όλες οι απαραίτητες οθόνες επικοινωνίας της εφαρμογής με τον χρήστη, καθώς και υπηρεσίες που σχετίζονται με την μορφοποίηση των δεδομένων που προβάλλονται.

1.3 Αρχιτεκτονική του συστήματος Reporter007

Όπως έχει ήδη αναφερθεί και παραπάνω η αρχιτεκτονική που θα ακολουθηθεί στο σύστημα είναι αυτή του πελάτη-εξυπηρετητή τριών επιπέδων (Three tier client-server). Ο λόγος που επιλέχθηκε αυτή είναι οι αρχιτεκτονική είναι διότι τα πλεονεκτήματα των δύο επιμέρους αρχιτεκτονικών προσφέρουν υψηλή απόδοση και εύκολη μεταβλητότητα στο σύστημα.



Σχήμα 3: Αναπαράσταση αρχιτεκτονικής τριών επιπέδων

Αναλυτικότερα τα τρία επίπεδα που αναφέρθηκαν παραπάνω, στο σύστημα Reporter007 θα διαμορφωθούν ως εξής:

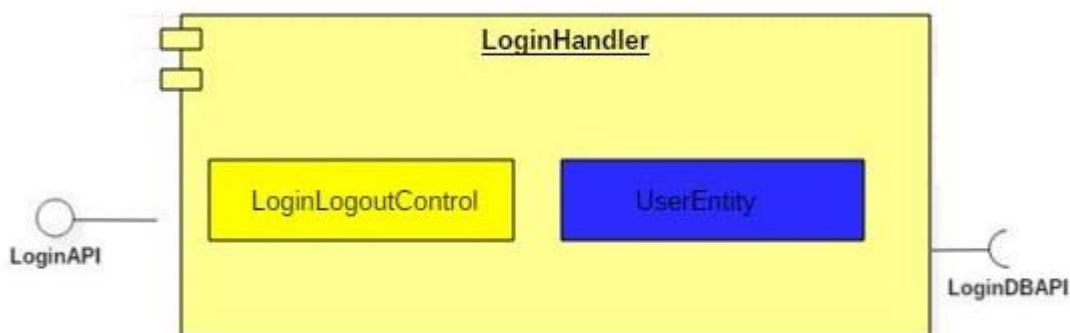
- ❖ **Πρώτο Επίπεδο (First Tier) - Database Server:**
Στο πρώτο επίπεδο περιλαμβάνονται όλα τα proxy για επικοινωνία του συστήματος με τις βάσεις δεδομένων του. Μέσω του επιπέδου επιτυγχάνεται η επικοινωνία του συστήματος με τις βάσεις δεδομένων του για παροχή και ανάγνωση δεδομένων.
- ❖ **Δεύτερο Επίπεδο (Second Tier) - Application Server:**
Στο δεύτερο επίπεδο περιλαμβάνονται όλοι οι ελεγκτές που ελέγχουν την λειτουργία της εφαρμογής. Το επίπεδο αυτό δέχεται αιτήματα που προέρχονται από το πρώτο επίπεδο και ανταποκρίνεται σε αυτά προβαίνοντας στις κατάλληλες λειτουργίες ώστε αυτά να ικανοποιηθούν.
- ❖ **Τρίτο Επίπεδο (Third Tier) – Client:**
Το τρίτο επίπεδο περιλαμβάνει όλες τις απαραίτητες διεπαφές για κάθε λειτουργία που εξυπηρετεί η εφαρμογή. Το επίπεδο αυτό αποτελεί ουσιαστικά την επικοινωνία του χρήστη με το σύστημα. Το υποσύστημα αυτό βασίζεται στη λειτουργία των δύο προηγούμενων υποσυστημάτων προκειμένου να εξυπηρετηθεί ο χρήστης. Αναλόγως της λειτουργίας που επιθυμεί ο χρήστης να εκτελεστεί εμφανίζεται η αντίστοιχη διεπαφή και μέσω των δύο υπολοίπων επιπέδων εξυπηρετείται το αίτημά του.

2. Προτεινόμενη Αρχιτεκτονική Λογισμικού

2.1 Αποδόμηση Συστήματος

2.1.1 Υποσύστημα LoginHandler

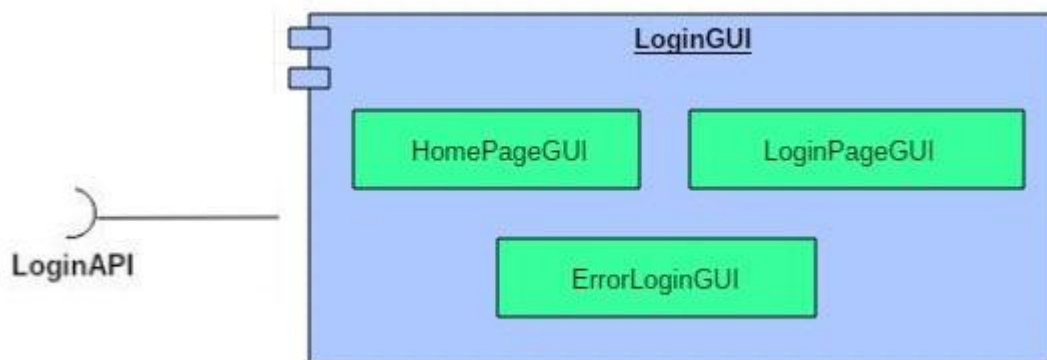
Το συγκεκριμένο υποσύστημα εμπεριέχει ως τμήματά του τον ελεγκτή που ελέγχει τη σύνδεση/ αποσύνδεση του χρήστη στο σύστημα, τον ελεγκτή που χειρίζεται την αρχική σελίδα της εφαρμογής, καθώς και την οντότητα με τις πληροφορίες που αφορούν το χρήστη. Συγκεκριμένα, οι κλάσεις που περιλαμβάνει το υποσύστημα είναι οι LoginLogoutControl και η UserEntity.



Σχήμα 4: Υποσύστημα LoginHandler

2.1.2 Υποσύστημα LoginGUI

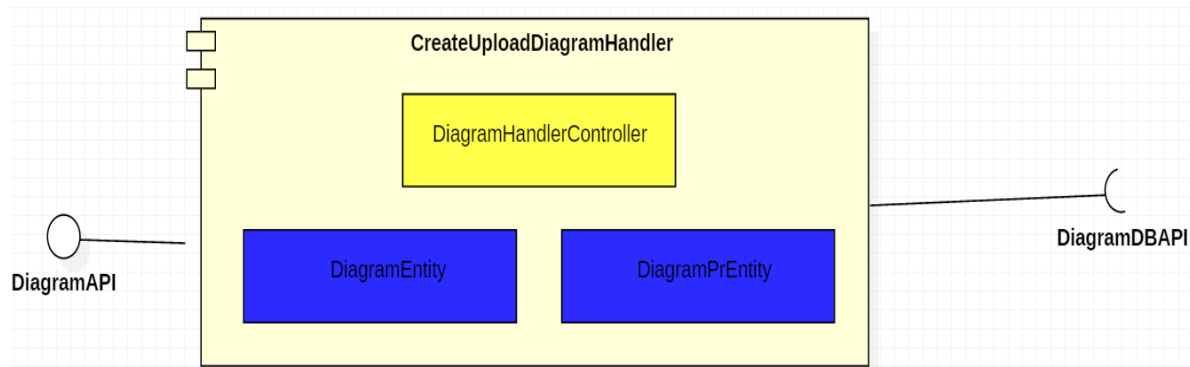
Το υποσύστημα αυτό αποτελείται από τις διεπαφές που συναντά ένας χρήστης πριν και αμέσως μετά την είσοδό του στο σύστημα. Αναλυτικότερα, οι κλάσεις που το απαρτίζουν είναι η LoginPageGUI, δηλαδή η σελίδα που εξυπηρετεί την είσοδο του χρήστη στο σύστημα, η ErrorLoginGUI, που αφορά το μήνυμα σφάλματος το οποίο εμφανίζεται σε περίπτωση εισαγωγής λανθασμένων στοιχείων στα πεδία της LoginPageGUI και η HomepageGUI, που αποτελεί την αρχική σελίδα της εφαρμογής.



Σχήμα 5: Υποσύστημα LoginGUI

2.1.3 Υποσύστημα CreateUploadDiagramHandler

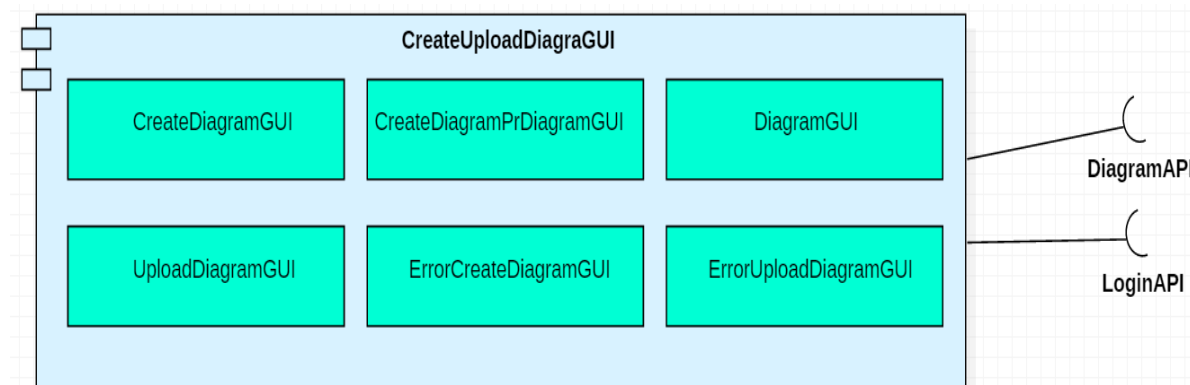
Το συγκεκριμένο υποσύστημα εμπεριέχει τον ελεγκτή που είναι υπεύθυνος για την διαχείριση της δημιουργίας ενός διαγράμματος στατιστικών, της μεταφόρτωσης (upload) ενός διαγράμματος από τον προσωπικό υπολογιστή του χρήστη, και της κοινοποίησής του στο σύστημα Instasis-backend. Ακόμη περιέχει την οντότητα που υλοποιεί το διάγραμμα καθώς και την οντότητα με τις πληροφορίες των χαρακτηριστικών του. Οι κλάσεις που περιλαμβάνονται είναι οι DiagramHandlerController, DiagramEntity και DiagramPrEntity.



Σχήμα 6: Υποσύστημα CreateUploadDiagramHandler

2.1.4 Υποσύστημα CreateUploadDiagramGUI

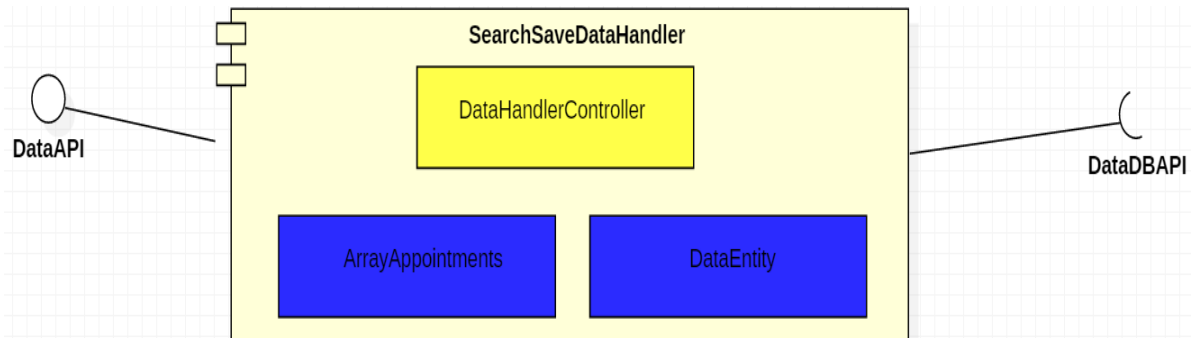
Το υποσύστημα αυτό περιλαμβάνει τις διεπαφές που σχετίζονται με την δημιουργία, την μεταφόρτωση και την αποστολή στο σύστημα instasis-backend, ενός διαγράμματος στατιστικών. Αναλυτικότερα περιλαμβάνει τις κλάσεις CreateDiagramGUI και CreateDiagramPrGUI, που αποτελούν τις διεπαφές για την εισαγωγή των χαρακτηριστικών του διαγράμματος, την DiagramGUI, που αποτελεί την διεπαφή για την προβολή του διαγράμματος, την UploadDiagramGUI, που είναι η διεπαφή για μεταφόρτωση ενός διαγράμματος, και τις ErrorCreateDiagramGUI και ErrorUploadDiagramGUI, που αποτελούν διεπαφές για προβολή μηνυμάτων σφάλματος δημιουργίας και μεταφόρτωσης ενός διαγράμματος αντίστοιχα.



Σχήμα 7: Υποσύστημα CreateUploadDiagramGUI

2.1.5 Υποσύστημα SearchSaveDataHandler

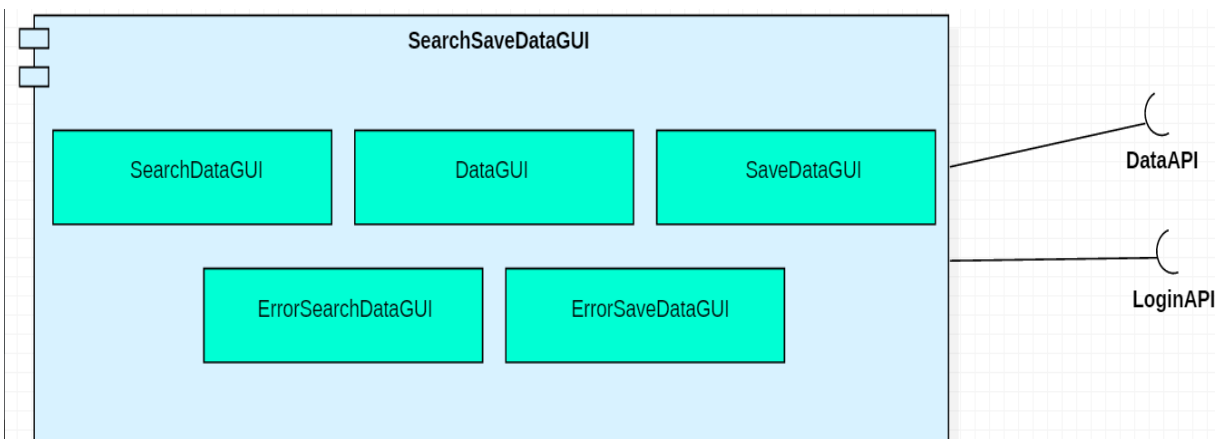
Το συγκεκριμένο υποσύστημα εμπεριέχει τον ελεγκτή που είναι υπεύθυνος για την αναζήτηση δεδομένων, την προβολή τους και την αποθήκευσή τους στον προσωπικό υπολογιστή του χρήστη. Ακόμη περιέχει την οντότητα που περιέχει τα δεδομένα που προκύπτουν από την αναζήτηση, καθώς και την οντότητα με τα κριτήρια της αναζήτησης. Οι κλάσεις που περιλαμβάνονται είναι οι DataHandlerController, ArrayAppointments και DataEntity.



Σχήμα 8: Υποσύστημα SearchSaveDataHandler

2.1.6 Υποσύστημα SearchSaveDataGUI

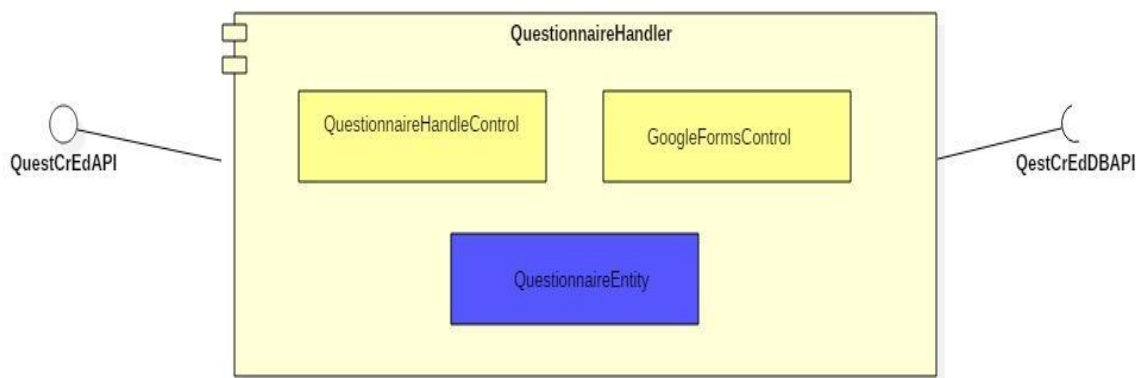
Το υποσύστημα αυτό περιλαμβάνει τις διεπαφές που σχετίζονται με την αναζήτηση δεδομένων, την προβολή τους και την αποθήκευσή τους στον προσωπικό υπολογιστή του χρήστη. Αναλυτικότερα περιλαμβάνει την κλάση SearchDataGUI, που αποτελεί την διεπαφή για την εισαγωγή από τον χρήστη των κριτηρίων αναζήτησης, την DataGUI, που αποτελεί την διεπαφή για την προβολή των δεδομένων που προέκυψαν από την αναζήτηση, την SaveDataGUI, που είναι η διεπαφή για την αποθήκευση των δεδομένων στον προσωπικό υπολογιστή του χρήστη, και τις ErrorSearchDataGUI και ErrorSaveDataGUI, που αποτελούν τις διεπαφές για την προβολή μηνυμάτων σφάλματος αναζήτησης και αποθήκευσης των δεδομένων.



Σχήμα 9: Υποσύστημα SearchSaveDataGUI

2.1.7 Υποσύστημα QuestionnaireHandler

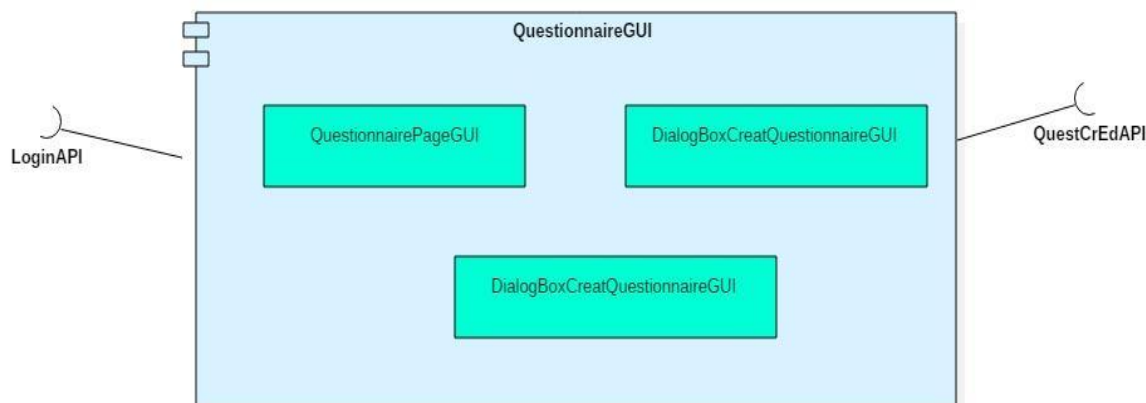
Το υποσύστημα QuestionnaireHandle περιέχει τις κλάσεις ελεγκτών και την κλάση Entity που σχετίζονται με την δημιουργία, διαχείριση και τροποποίηση των ερωτηματολογίων. Μέσα στο σύστημα περιέχεται η κλάση QuestionnaireHandleControl η οποία διαχειρίζεται τις ενέργειες που πρέπει να γίνουν σχετικά με τα ερωτηματολόγια και συνδέονται με το σύστημα InstasisBackend. Επιπλέον μέσα στο υποσύστημα βρίσκεται η κλάση GoogleFormsControl η οποία είναι υπεύθυνη για τις ενέργειες που πρέπει να γίνουν πάνω στα ερωτηματολόγια και σχετίζονται με την εξωτερική εφαρμογή ερωτηματολογίων την οποία χρησιμοποιούμε για να κατασκευάσουμε τα ερωτηματολόγια μας. Τέλος η κλάση QuestionnaireEntity που περιέχεται και αυτή στο υποσύστημα είναι η κλάση που αναπαριστά τα αντικείμενα ερωτηματολογίων που βρίσκονται στο σύστημα. Η διεπαφές που παρέχει το υποσύστημα είναι αυτές που χρειάζονται για την δημιουργία, τροποποίηση , και προβολής ερωτηματολογίων από τις αντίστοιχες κλάσεις όπως παρουσιάστηκαν παραπάνω.



Σχήμα 10: Υποσύστημα QuestionnaireHandler

2.1.8 Υποσύστημα QuestionnaireGUI

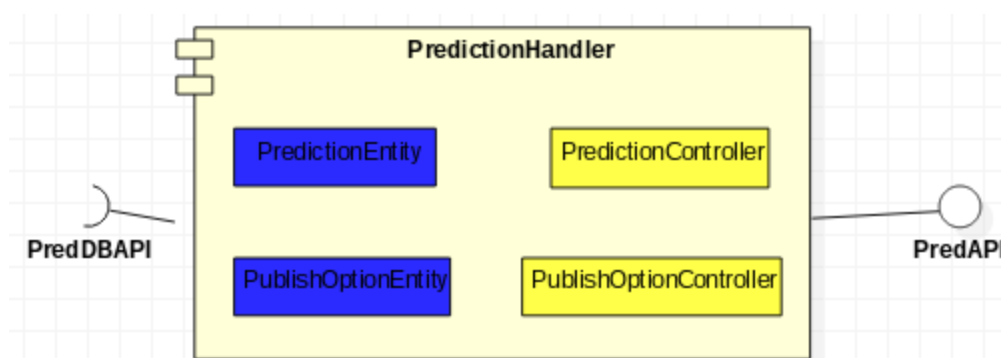
Το υποσύστημα QuestionnaireGUI εμπεριέχονται οι κλάσεις που δημιουργούν τις γραφικές διεπαφές ανάμεσα στο χρήστη και στην σελίδα ερωτηματολογίων και την λίστα ερωτηματολογίων, τα αναδυόμενα παράθυρα για την επιβεβαίωση μετά από την επιλογή του χρήστη να επεξεργαστεί ή να διαγράψει ένα ερωτηματολόγιο. Συγκεκριμένα οι κλάσεις DialogBoxCreatQuestionnaireGUI και DialogBoxCreatQuestionnaireGUI αποτελούν τις διεπαφές για την επιβεβαίωση των κινήσεων του χρήστη και η κλάση QuestionnairePageGUI αναπαριστά την αρχική σελίδα των ερωτηματολογίων



Σχήμα 11: Υποσύστημα QuestionnaireGUI

2.1.9 Υποσύστημα PredictionHandler

Το υποσύστημα αυτό έχει αναλάβει την διαχείριση των προβλέψεων του συστήματος. Περιέχει 2 ελεγκτές, έναν για την δημιουργία και αποστολή προβλέψεων πάνω σε κάποιο αίτημα φοιτητή για ραντεβού (PredictionController) και έναν για την δημιουργία και την ρύθμιση διαγραμμάτων πρόβλεψης της απασχόλησης της γραμματείας (PublishOptionController). Οι ελεγκτές αυτοί διαχειρίζονται τα 2 αυτά είδη προβλέψεων χάρη στις 2 κλάσεις οντότητας, που μοντελοποιούν μια πρόβλεψη σε αίτημα (PredictionEntity) ή μια ρύθμιση διαγραμμάτων πρόβλεψης (PublishOptionEntity).

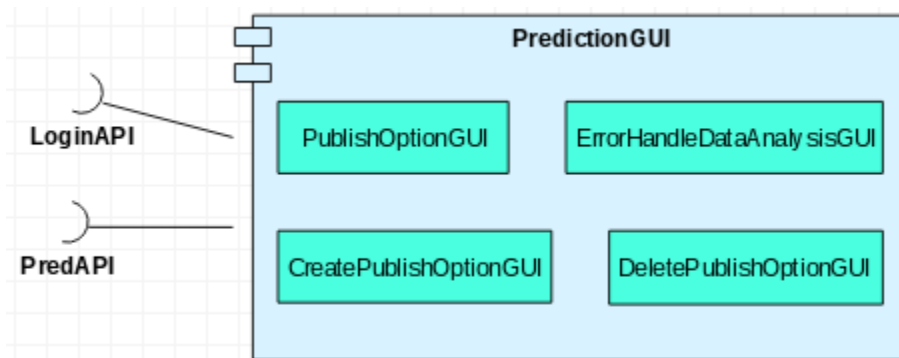


Σχήμα 12: Υποσύστημα PredictionHandler

2.1.10 Υποσύστημα PredictionGUI

Το υποσύστημα αυτό αποτελείται από τις διεπαφές που συναντά ο χρήστης (αναλυτής) όταν ξεκινήσει την διαδικασία της ρύθμισης διαγραμμάτων πρόβλεψης της απασχόλησης της γραμματείας. Αυτές οι διεπαφές είναι συνολικά 4. Η πρώτη, είναι η οθόνη κατά την εκκίνηση της ρύθμισης προβλέψεων που δίνει τις επιλογές της δημιουργίας και της διαγραφής κάποιας ρύθμισης (PublishOptionGUI). Η δεύτερη, είναι μια οθόνη σφάλματος που εμφανίζει κάθε φορά το μήνυμα του σφάλματος (ErrorHandleDataAnalysisGUI). Η τρίτη, είναι η διεπαφή στην οποία ο χρήστης επιλέγει τις παραμέτρους της ρύθμισης διαγραμμάτων απασχόλησης, που θέλει να δημιουργήσει (CreatePublishOptionGUI). Τέλος, η τέταρτη είναι η διεπαφή που

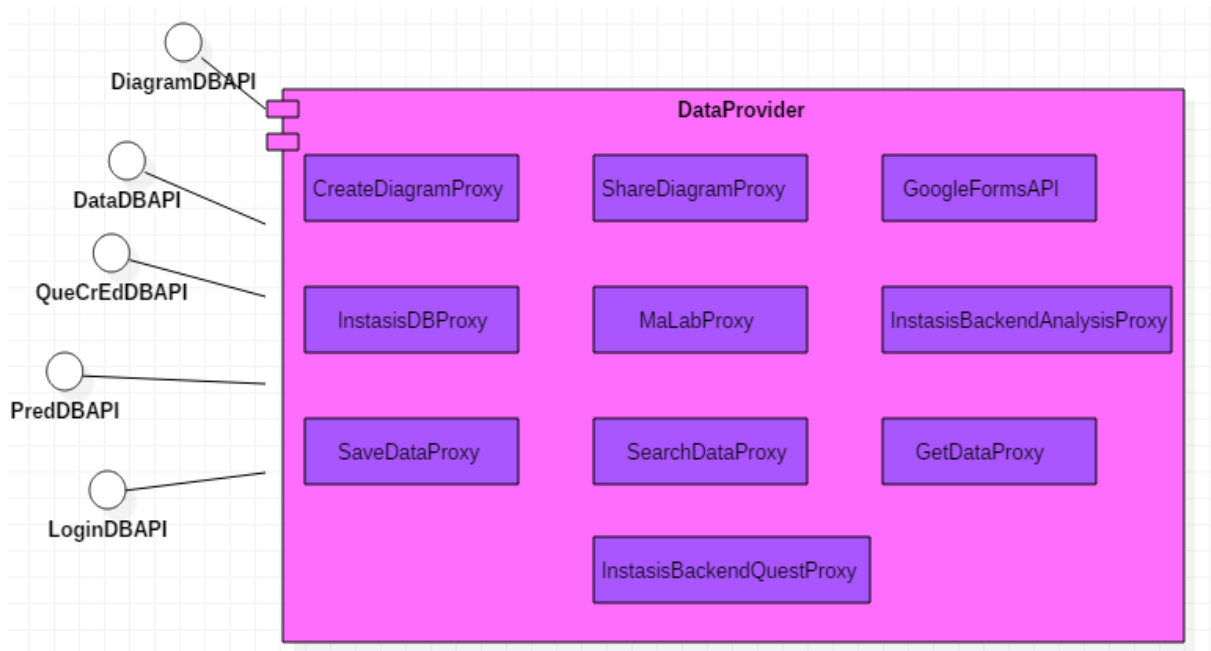
δίνει στον χρήστη την δυνατότητα να διαγράψει όποια ήδη υπάρχουσα ρύθμιση διαγραμμάτων πρόβλεψης επιθυμεί (DeletePublishOptionGUI).



Σχήμα 13: Υποσύστημα PredictionGUI

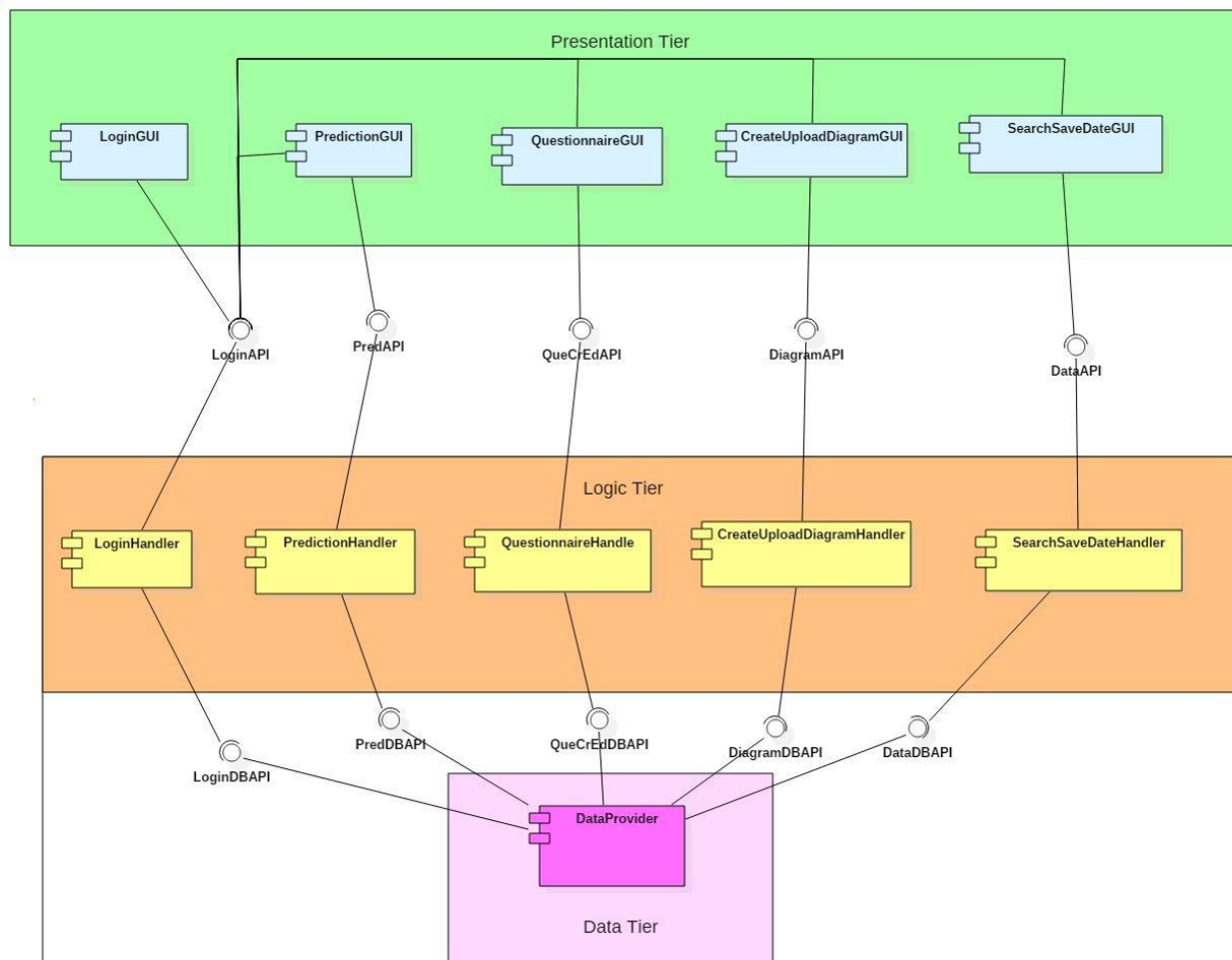
2.1.11 Υποσύστημα DataProvider

Το υποσύστημα αυτό είναι απαραίτητο για την λειτουργία όλων των παραπάνω. Ουσιαστικά δημιουργεί τις συνδέσεις με όλα τα εξωτερικά συστήματα και παρέχει όλα τα δεδομένα ώστε να τα επεξεργαστούν τα υπόλοιπα υποσυστήματα. Πιο συγκεκριμένα, το υποσύστημα αποτελείται από 6 συνδέσεις με το εξωτερικό σύστημα Instasis-Backend, 2 συνδέσεις με το MataLab, 1 σύνδεση με το GoogleForms και 1 με το λειτουργικό του χρήστη που χρησιμοποιεί την εφαρμογή μας. Για το Instasis-Backend έχουμε σύνδεση, αρχικά, για την πιστοποίηση του χρήστη (InstasisDBProxy), όπως και για τις ρυθμίσεις διαγραμμάτων πρόβλεψης και την αποστολή προβλέψεων κάποιου αιτήματος για ραντεβού (InstasisBackendAnalysisProxy). Επίσης, έχουμε σύνδεση για την αποστολή και λήψη πληροφοριών για τα ερωτηματολόγια (InstasisBackendQuestProxy) καθώς και για την αποστολή διαγραμμάτων που φτιάχτηκαν ανεξάρτητα από την εφαρμογή (ShareDiagramProxy). Οι τελευταίες συνδέσεις για το Instasis-Backend είναι μια για την λήψη δεδομένων αναζήτησης από τις καταχωρήσεις των ραντεβού (SearchDataProxy) και μια για την λήψη των πληροφοριών ενός διαγράμματος (GetDataProxy). Όσον αφορά το εξωτερικό σύστημα MataLab, υπάρχει μια σύνδεση για την αποστολή των απαραίτητων πληροφοριών και τη λήψη του αντίστοιχου διαγράμματος (ShareDiagramProxy) και μια για την αποστολή των δεδομένων και την λήψη προβλέψεων αιτήματος ή διαγράμματος πρόβλεψης της απασχόλησης της γραμματείας (MaLabProxy). Τέλος, υπάρχουν 2 συνδέσεις ακόμα, μια για την δημιουργία και την διαχείριση των ερωτηματολογίων μέσω της εξωτερικής εφαρμογής GoogleForms (GoogleFormsAPI) και μια για την αποθήκευση αρχείων (από τα δεδομένα αναζήτησης) στον σκληρό δίσκο του χρήστη, μέσω του λειτουργικού του (SaveDataProxy).



Σχήμα 14: Υποσύστημα Databases

Παρακάτω στο Σχήμα 15 μπορούμε να δούμε το διάγραμμα τμημάτων που παρουσιάζει τις εξαρτήσεις μεταξύ των επιμέρους υποσυστημάτων. Μέσω του διαγράμματος μπορεί να γίνει εύκολα αντιληπτή η συσχέτιση μεταξύ των υποσυστημάτων και επιπλέον παρατηρώντας το μπορούμε να συμπεράνουμε αν καλύπτεται η λειτουργικότητα του συστήματος. Τέλος το διάγραμμα μπορεί να χρησιμοποιηθεί ως γνώμονας κατά την διαδικασία επιμερισμού του συνολικού συστήματος σε υποσυστήματα και ανάθεσης αυτών των υποσυστημάτων σε ομάδες ανάπτυξης,



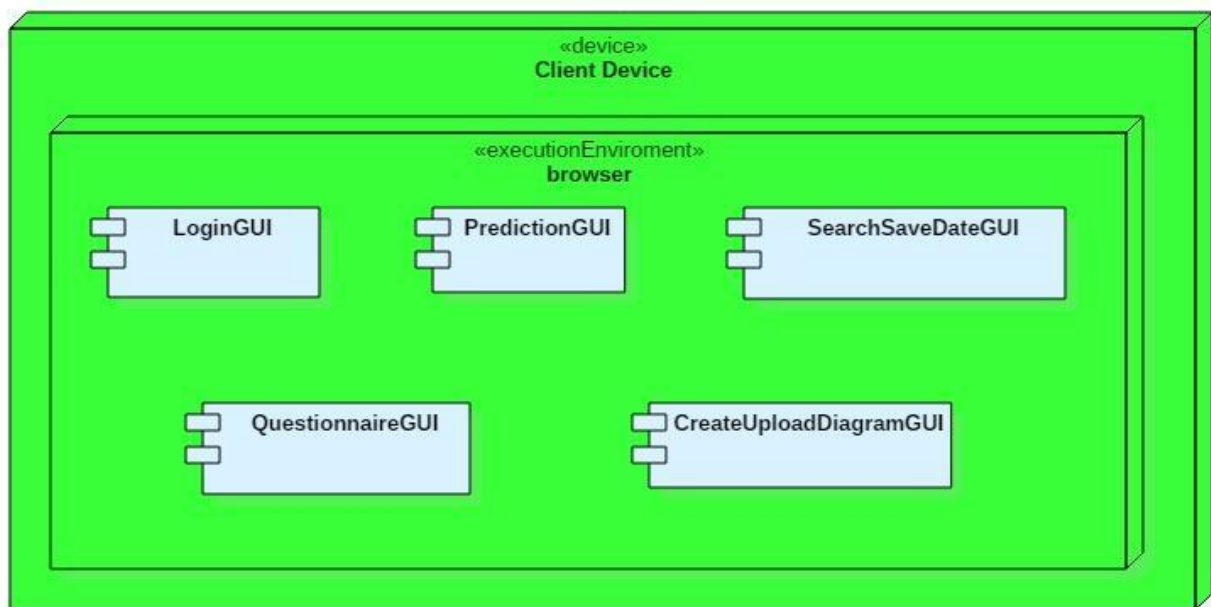
Σχήμα 15: Συνολικό διάγραμμα τμημάτων

2.2 Απεικόνιση Υλικού/Λογισμικού

Σκοπός αυτής της παραγράφου είναι η κατανόηση του τρόπου σύνδεσης των τμημάτων του συστήματος και ο διαχωρισμός του συστήματος σε φυσικούς κόμβους με βάση την λειτουργίες που θέλουμε να εξυπηρετεί το σύστημα και ταυτόχρονα να ικανοποιούνται οι μη λειτουργικές απαιτήσεις του. Με βάση την αρχιτεκτονική που χρησιμοποιήσαμε το υλικό του συστήματος καθώς και τα προγράμματα αλλά και τα εξωτερικά συστήματα που χρησιμοποιούνται οργανώνονται σε φυσικό επίπεδο σύμφωνα με τα όσα αναλύονται στην συνέχεια. Η παράγραφος αυτή στην ουσία θα μας δείξει την σχέση που υπάρχει ανάμεσα σε υλικό και λογισμικό.

2.2.1 Client Device

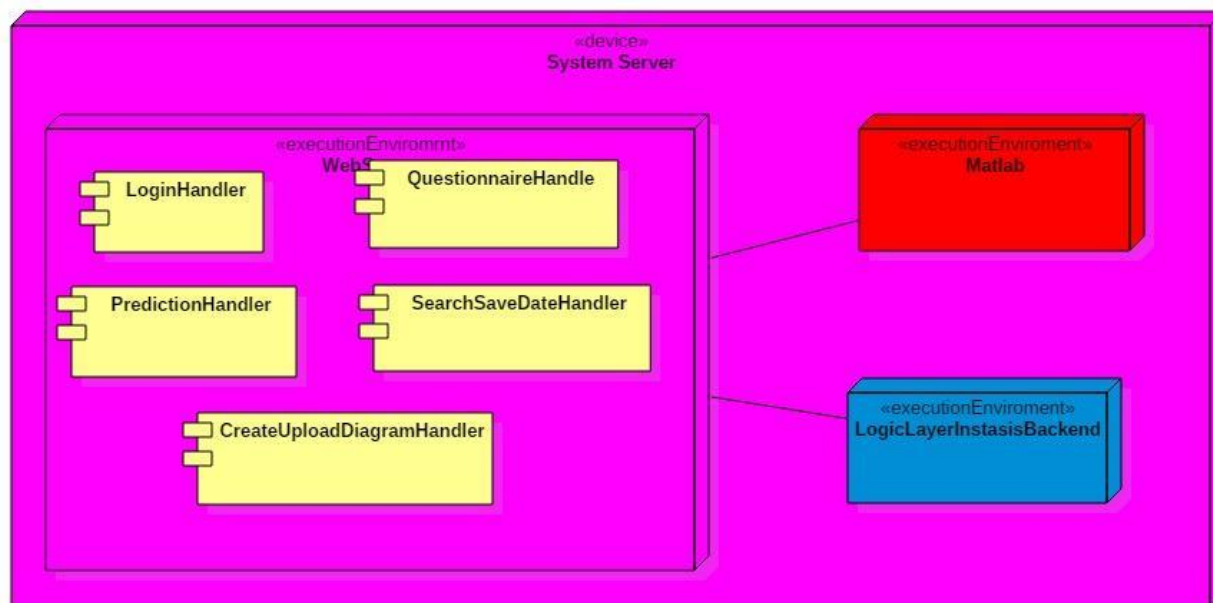
Ο συγκεκριμένος κόμβος αφορά τις λειτουργίες του χρήστη και αποτελεί τον κόμβο της συσκευής που χρησιμοποιεί ο χρήστης για να συνδεθεί μέσω διαδικτύου στο σύστημα του server της εφαρμογής. Ο κόμβος αυτός πρέπει να είναι ένα υπολογιστικό σύστημα με αυξημένα χαρακτηριστικά και ικανότητες σε σχέση με μία φορητή συσκευή χειρός, δηλαδή θα πρέπει να είναι ένας υπολογιστής. Επιπλέον για το σύστημα η σύνδεση στο διαδίκτυο είναι απαραίτητη για να γίνει η επικοινωνία με τον server και κατά επέκταση η ανταλλαγή πληροφοριών. Ο κόμβος με βάση την αρχιτεκτονική τριών επιπέδων συσχετίζεται αποκλειστικά με τον χρήστη και επομένως αποτελείται εξ ολοκλήρου από το γραφικό περιβάλλον και τις διεπαφές που θα ανοίξουν από κάποιον φυλλομετρητή (browser) του χρήστη και θα του δώσουν την δυνατότητα επικοινωνίας με τον server του συστήματός μας.



Σχήμα 16: Client Device

2.2.2 System Server

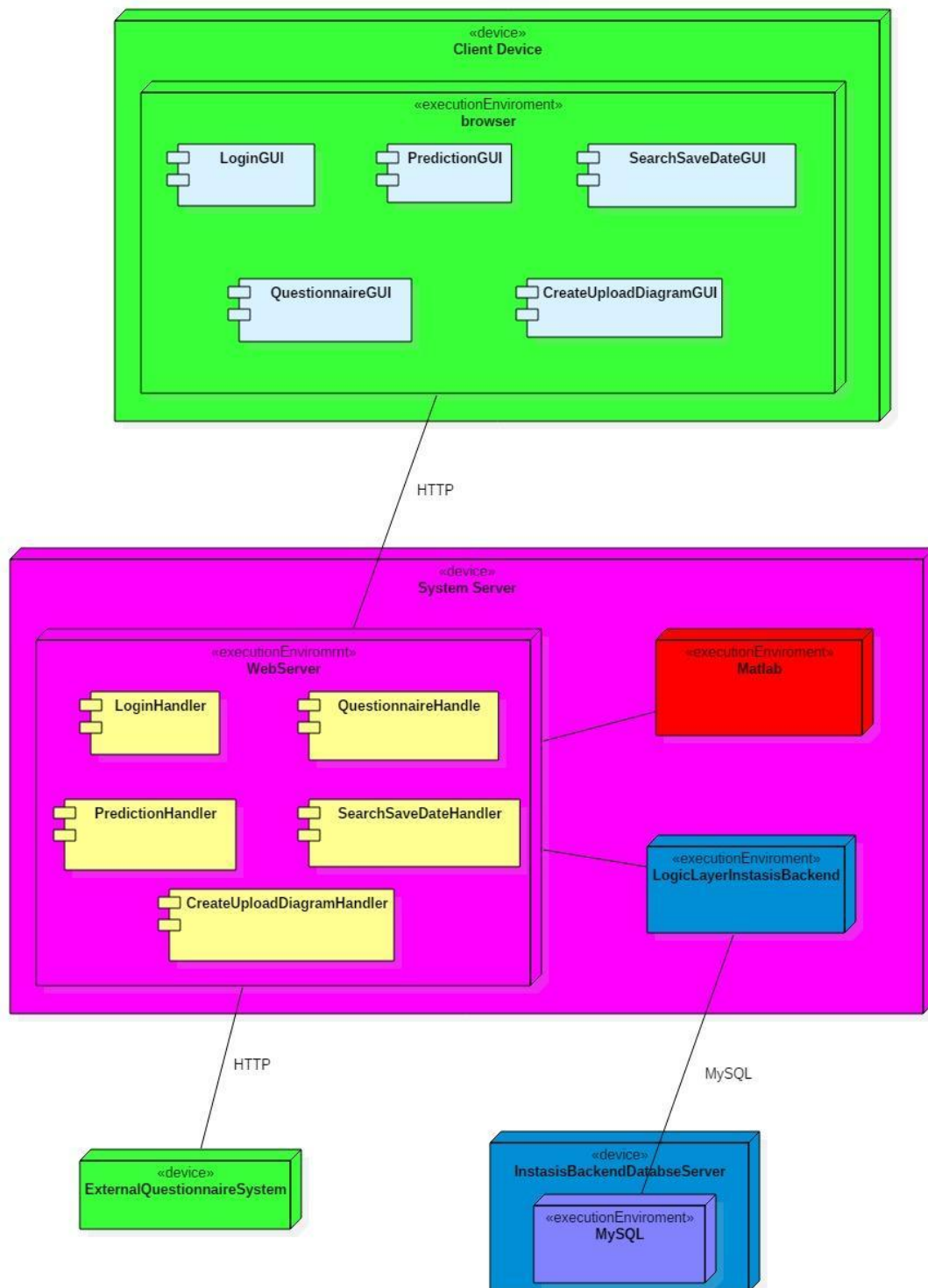
Ο συγκεκριμένος κόμβος αποτελείται από ένα εξυπηρετητή παγκόσμιου ιστού (WebServer) ώστε να είναι εφικτή η επικοινωνία με το σύστημα του χρήστη. Τέτοιου είδους προγράμματα μπορεί να είναι αίτημα προγράμματα ανοιχτού κώδικα όπως ο εξυπηρετητής Apache HTTP και Internet Information Services ή και ακόμα να γίνει ανάπτυξη κάποιου Webserver με την βοήθεια της πλατφόρμας πλατφόρμα .NET. Επιπρόσθετα στον κόμβο αυτό περιέχεται και το πρόγραμμα Matlab που θα μας δώσει την δυνατότητα για την δημιουργία των διαγραμμάτων και την επεξεργασία των δεδομένων που χρειαζόμαστε. Τέλος ο κόμβος περιέχει επιπλέον το λογικό κομμάτι του συστήματος InstasisBacked ακριβώς όπως επιβάλλει η αρχιτεκτονική τριών επιπέδων.



Σχήμα 17: System Server

2.2.3 Συνολικό διάγραμμα ανάπτυξης

Από το συνολικό διάγραμμα τμημάτων φαίνεται η σχέση ανάμεσα στους κόμβους που έχουν δημιουργηθεί. Αρχικά βλέπουμε ότι η επικοινωνία του συστήματος του εξυπηρετητή με το σύστημα του χρήστη θα γίνεται μέσω του πρωτοκόλλου HTTP. Το συγκεκριμένο πρωτόκολλο είναι το πιο διαδεδομένο πρωτόκολλο επικοινωνίας για τέτοιες περιπτώσεις και παρέχει υψηλής ποιότητας κρυπτογράφησης και ασφάλειας κατά την διαδικασία ανταλλαγής δεδομένων εντός διαδικτύου. Το σύστημα του server με την χρήση του ίδιου πρωτοκόλλου επικοινωνεί και με το εξωτερικό σύστημα ερωτηματολογίων. Η επικοινωνία μπορεί να γίνει μέσω των γλωσσών HTML, CSS και JSON. Στην συνέχεια παρατηρούμε ότι στο σύστημα το server υπάρχει το πρόγραμμα Matlab το οποίο χρησιμοποιείται από το σύστημα μας για την ανάπτυξη και επεξεργασία διαγραμμάτων καθώς και για την επεξεργασία των δεδομένων του συστήματος. Επιπλέον στο κόμβο του server συμπεριλαμβάνουμε και το λογικό κομμάτι του συστήματος InstasisBackend, που είναι υπεύθυνο για την επικοινωνία και την ανταλλαγή μηνυμάτων και πληροφοριών. Τέλος βλέπουμε ότι εξωτερικά από τον κόμβο που διαχειρίζεται το λογικό κομμάτι του συστήματος μας, δηλαδή το κόμβο του server, υπάρχει το κομμάτι του InstasisBackend που είναι υπεύθυνο για την αποθήκευση των δεδομένων και είναι ο κόμβος InstasisBackendDatabaseServer. Στον κόμβο αυτό μέσω του περιβάλλοντος MySQL και τον αντίστοιχων προτύπων επικοινωνίας το λογικό κομμάτι του InstasisBackend διαχειρίζεται να δεδομένα του συστήματος.



Σχήμα 18: Διάγραμμα ανάπτυξης

Στη συνέχεια παρατίθεται οι ελάχιστες και οι προτεινόμενες απαιτήσεις σε επίπεδο χρήστη και server:

	Minimum Requirements	Recommended Requirements
Client	CPU: Dual core 2.2 GHZ	CPU: Dual core 2.6 GHZ
	RAM: 2 GB	RAM: 4 GB
	Video Card: nVidia 9600GT	Video Card: nVidia Geforce GTX 660
Server	CPU: i3 2.7 GHZ	CPU: i7 4.4 GHZ
	RAM: 4 GB	RAM: 8 GB
	Video Card: nVidia Geforce GTX 970	Video Card: nVidia Geforce GTX Titan X

2.3 Έλεγχος Πρόσβασης και Ασφάλεια

Η πρόσβαση στο σύστημα γίνεται μόνο μετά από έλεγχο ενός ονόματος χρήστη (username) και ενός κωδικού (password). Αυτά εισάγονται στη σελίδα σύνδεσης από τον χρήστη. Για την δημιουργία του λογαριασμού και την παραχώρηση κωδικού και ονόματος στον χρήστη είναι υπεύθυνος ο διαχειριστής του όλου συστήματος (Instasis-Admin). Με διαδικασίες στο δικό του υποσύστημα ο διαχειριστής παρέχει στον αναλυτή, λοιπόν, τα 2 αυτά στοιχεία για την είσοδό του στο σύστημα. Τα δεδομένα των χρηστών είναι αποθηκευμένα στο εξωτερικό σύστημα InstasisBackend που τα έχει αποθηκεύσει και κρυπτογραφήσει με τις δικές του διαδικασίες σε βάση δεδομένων. Αυτό το σύστημα κάνει και τον έλεγχο για να επιβεβαιώσει ότι ο χρήστης έχει δικαίωμα πρόσβασης στο σύστημά μας.

Θεωρούμε 2 είδη χρηστών, τον επισκέπτη (guest user), που δεν έχει ακόμα συνδεθεί με όνομα χρήστη και κωδικό και τον εγγεγραμμένο χρήστη (registered user), ο οποίος έχει ήδη εισάγει όνομα χρήστη και κωδικό και έχει επιβεβαιωθεί από το σύστημα. Έτσι, παρουσιάζονται για τους 2 αυτούς χρήστες κάθε γραφικό περιβάλλον και τα δικαιώματα που έχει σε αυτό ο χρήστης.

Παρακάτω λοιπόν παρουσιάζεται ο πίνακας πρόσβασης (access matrix) αρχικά για τον επισκέπτη και στη συνέχεια για τον εγγεγραμμένο χρήστη. Τα ορίσματα των μεθόδων έχουν παραληφθεί στον πίνακα για λόγους απλότητας, από την στιγμή που δεν θα

Guest user	LoginPageGUI	ErrorLoginGUI
	LoginPageGUI()	display()
	loginClick()	backClick()

Registered user	HomepageGUI	PublishOptionGUI
	HomePageUserGUI()	PublishOptionGUI()
	logoutClick()	createClick()
	createPlotClick()	homeClick()
	searchDataClick()	deleteClick()
	addPlotsClick()	displayMePublishOption()
	questionnairesClick()	
	predictionSettingsClick()	
	CreatePublishOptionGUI	DeletePublishOptionGUI
	CreatePublishOptionGUI()	DeletePublishOptionGUI()
	cancelClick()	cancelClick()
	homeClick()	confirmClick()
	confirmClick()	displayMeDeletePublishOption()
	displayMeCreatePublish()	
	ErrorHandleDataAnalysisGUI	CreateDiagramGUI
	backClick()	CreateDiagramGUI()
	showError()	nextClick()
		display()
	CreateDiagramPrGUI	DiagramGUI
	CreateDiagramPrGUI()	DiagramGUI()
	cancelClick()	cancelClick()
	diagramClick()	shareClick()
	display()	display()
	UploadDiagramGUI	ErrorCreateDiagramGUI
	UploadDiagramGUI()	ErrorCreateDiagramGUI()
	cancelClick()	backClick()
	uploadClick()	display()
	display()	
	ErrorUploadDiagramGUI	QuestionnairePageGUI
	ErrorUploadDiagramGUI()	QuestionnairePageGUI()
	tryAgainClick()	editQuestionnaireClick()
	display()	createQuestionnaireClick()
		scrollQuestionnaireListClick()
		changeListQuestionnaireButtton()
		displayQuestionnairePage()
		displayHomePageClick()
	DialogBoxHandleQuestionnaireGUI	DialogBoxCreateQuestionnaireGUI
	DialogBoxHandleQuestionnaireGUI()	DialogBoxCreateQuestionnaireGUI()

	viewResponsesClick()	saveQuestionnaireClick()
	editQuestionnaireClick()	continueEditQuestionnaireClick()
	deleteQuestionnaireClick()	editQuestionnaire()
	displayDialog()	rejectQuestionnaireClick()
		displayDialog()
	SearchDataGUI	DataGUI
	SearchDataGUI()	DataGUI()
	searchClick()	backClick()
	displaySearchDataGUI()	saveDataClick()
		scrollClick()
		displayDataGUI()
	SaveDataGUI	ErrorSearchDataGUI
	SaveDataGUI()	ErrorSearchDataGUI()
	saveClick()	backClick()
	displaySaveDataGUI()	display()
	ErrorSaveDataGUI	
	ErrorSaveDataGUI()	
	backClick()	
	display()	

3. Προδιαγραφές Τμηματικού Σχεδιασμού (Component Design Specifications)

3.1 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

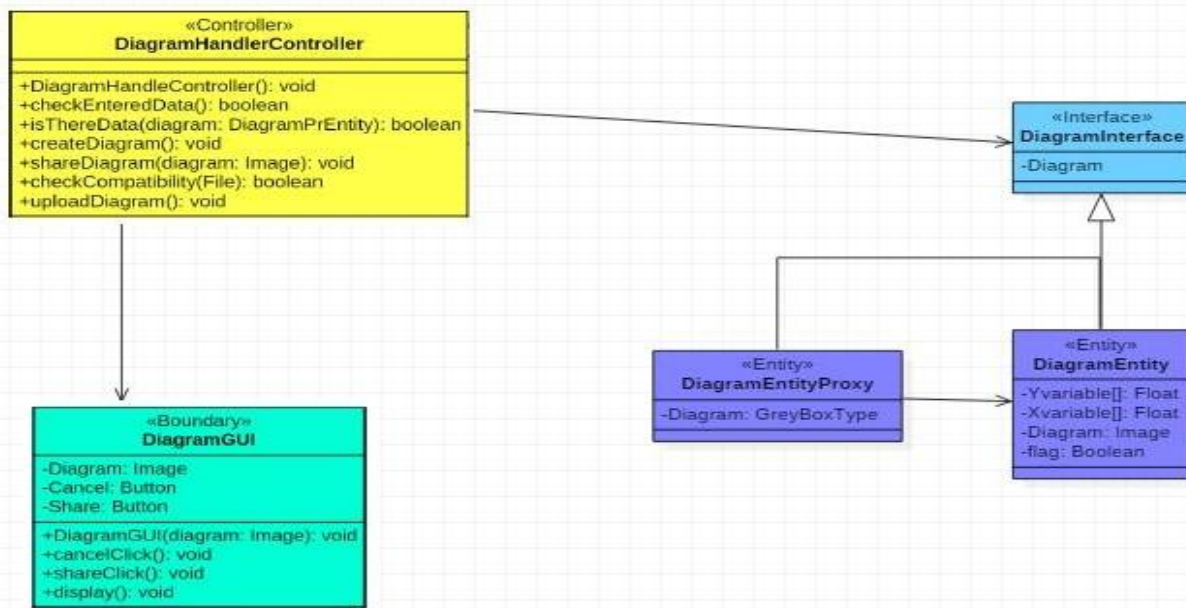
3.1.1 Δομικά πρότυπα

Τα δομικά πρότυπα εισάγουν abstract κλάσεις για μελλοντικές επεκτάσεις του συστήματος. Επίσης, μειώνουν την σύζευξη ανάμεσα σε κλάσεις και κάνουν πιο εύκολη την συντήρηση του συστήματος.

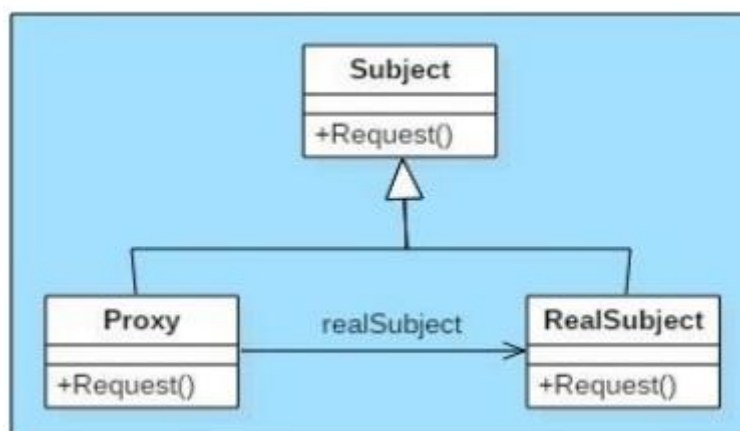
Proxy Design Pattern:

Το πρότυπο αυτό βοηθάει στην πιο γρήγορη φόρτωση σελίδων καθώς χρησιμοποιεί ένα άλλο αντικείμενο το Proxy, το οποίο λειτουργεί ως αντικαταστάτης του πραγματικού αντικειμένου. Το Proxy δηλαδή αντικαθιστά το πραγματικό αντικείμενο μέχρι εκείνο να είναι διαθέσιμο.

- Πρόβλημα που αντιμετωπίστηκε: Το πρότυπο αυτό χρησιμοποιήθηκε γιατί στην κλάση DiagramGUI πρέπει να εμφανιστεί σε μικρογραφία το διάγραμμα που δημιουργήθηκε. Το αντικείμενο όμως DiagramEntity είναι ακριβό και είναι πιθανόν να καθυστερήσει να δημιουργηθεί. Για να είναι, λοιπόν, σε θέση το σύστημα να ανοίγει την σελίδα ταχύτερα, μέσω του προτύπου αυτού οι εικόνες και αποθηκεύονται και φορτώνονται ξεχωριστά από το κείμενο και εάν μια εικόνα ή βίντεο δεν φορτώνεται, ο DiagramEntityProxy δείχνει ένα γκρι ορθογώνιο στη θέση της εικόνας.
- Υποσυστήματα που αναδιοργανώθηκαν: DiagramHandle



Σχήμα 19: Το πρότυπο Proxy όπως εφαρμόστηκε στο υποσύστημα DiagramHandle



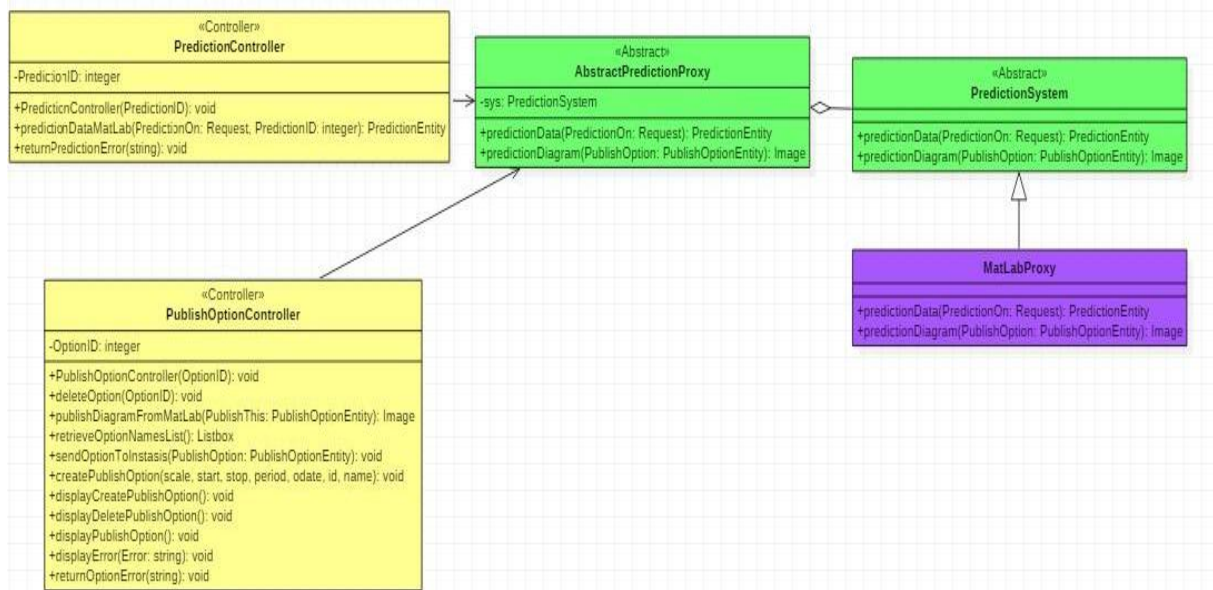
Σχήμα 20: Το πρότυπο Proxy

Bridge Design Pattern:

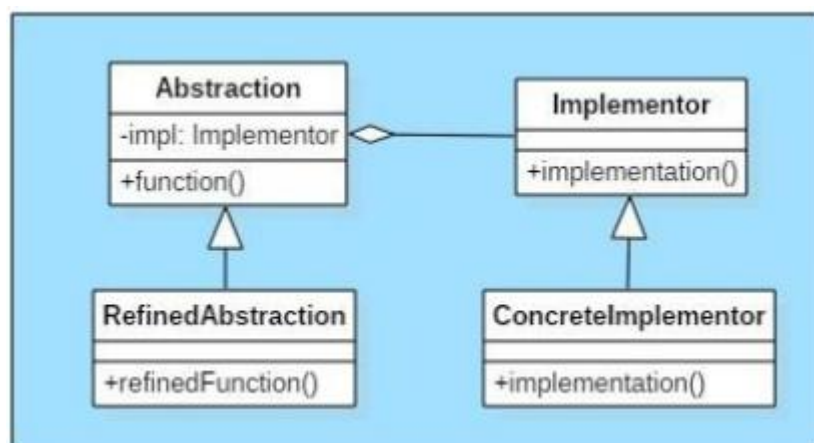
Το πρότυπο της γέφυρας έχει ως στόχο την αποσύνδεση μιας αφαίρεσης από την υλοποίησή της, ώστε να μπορούν να μεταβάλλονται ανεξάρτητα. Όταν μια αφαίρεση μπορεί να έχει περισσότερες από μία υλοποιήσεις, ο συνήθης τρόπος οργάνωσης είναι με τη χρήση κληρονομικότητας. Με τον όρο αφαίρεση εννοείται μια αφηρημένη κλάση που ορίζει μια διασύνδεση (ένα σύνολο υπογραφών), ενώ υλοποιήσεις είναι οι συγκεκριμένες παράγωγες κλάσεις οι οποίες υλοποιούν τις μεθόδους της αφηρημένης κλάσης. Με τον τρόπο αυτό είναι

δυνατόν πολλές υλοποιήσεις της ίδιας λειτουργίας να βρίσκονται κάτω από την ίδια διεπαφή, κάνοντας το σύστημα συμβατό με διάφορα εξωτερικά συστήματα.

- Πρόβλημα που αντιμετωπίστηκε: Το πρότυπο αυτό χρησιμοποιήθηκε για την διασύνδεση της εφαρμογής με το εξωτερικό σύστημα Matlab, το οποίο χρησιμοποιείται για την παραγωγή προβλέψεων και διαγραμμάτων πρόβλεψης. Ο λόγος που χρησιμοποιήθηκε το πρότυπο γέφυρας είναι για να επιτρέψει μελλοντικές επεκτάσεις, με άλλα συστήματα εκτός Matlab αλλά και διαφορετικές υλοποιήσεις μέσω Matlab. Έτσι, αν θεωρηθεί σκόπιμο μπορεί να προστεθεί σύνδεση και με άλλο εξωτερικό σύστημα προβλέψεων στο μέλλον ή και να προστεθούν επιπλέον δυνατότητες στο ήδη υπάρχων του Matlab.
- Υποσυστήματα που αναδιοργανώθηκαν: DataAnalysisHandle



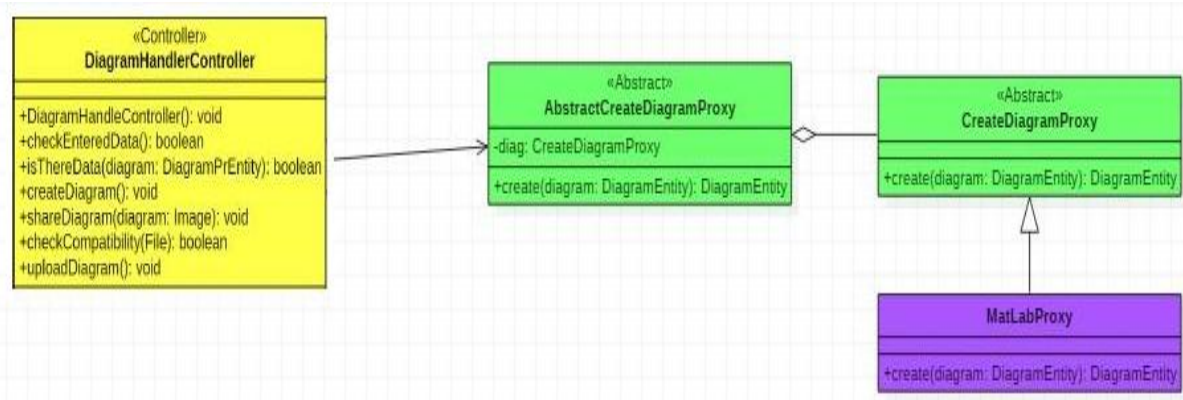
Σχήμα 21: Το πρότυπο Bridge όπως εφαρμόστηκε στο υποσύστημα DataAnalysisHandle



Σχήμα 22: Το πρότυπο Bridge

- Πρόβλημα που αντιμετωπίστηκε: Το πρότυπο αυτό χρησιμοποιήθηκε για την διασύνδεση της εφαρμογής με το εξωτερικό σύστημα Matlab, το οποίο χρησιμοποιείται για την δημιουργία ενός διαγράμματος στατιστικών. Ο λόγος που χρησιμοποιήθηκε το πρότυπο γέφυρας είναι η συνεργασία ενδεχομένως ασύμβατων διεπαφών μεταξύ της εφαρμογής και του συστήματος Matlab.

- Υποσύστημα που αναδιοργανώθηκε: DiagramHandle



Σχήμα 23: Το πρότυπο Bridge όπως εφαρμόστηκε στο υποσύστημα DiagramHandle

4. Πίνακας ιχνηλασιμότητας εγγράφων Σχεδίασης και Απαιτήσεων Λογισμικού

Η μετάβαση από το έγγραφο SRD στο έγγραφο SDD δεν εμπεριέχει κάποιες μεταβολές του αρχικού. Οι μόνες αλλαγές που πραγματοποιήθηκαν αφορούν τις αλλαγές λόγω προσθήκης προτύπων σχεδιασμού. Οι προσθήκες αυτές έχουν ήδη περιγραφεί αναλυτικά και για την ανάγνωσή του προτείνεται στον αναγνώστη του παρόντος εγγράφου η μετάβαση στη παρακάτω παράγραφο του εγγράφου:

Παράγραφος 3.1.1 (Σελίδες 25-26-27-28)

5. Παράρτημα I – Ανοιχτά Θέματα

- Οι ελάχιστες και προτεινόμενες απαιτήσεις σε επίπεδο server και client είναι ενδεικτικές και ενδέχεται να μεταβληθούν. Αν επέλθει κάποια μεταβολή τότε το έγγραφο αυτό πρέπει να συμπληρωθεί ξανά με τις νέες τιμές.
- Καλύτερος τεχνικός σχεδιασμός του διακομιστή της εφαρμογής από επαγγελματίες, με σκοπό την ελαχιστοποίηση των δαπανών κατασκευής του.
- Είναι σημαντικό να γίνει σωστή κοστολόγηση του υπολογιστή server, ανάλογα με τον πληθυσμό των χρηστών. Η κοστολόγηση των servers πρέπει να γίνει ανάλογα με τους αναμενόμενους χρήστες της εφαρμογής με δυνατότητα επέκτασης.
- Δυνατότητα χρήσης της εφαρμογής από διάφορα λειτουργικά συστήματα (Windows, Linux, iOS).
- Μερικά κομμάτια του προτεινόμενου συστήματος πρέπει να μπορούν να αντικατασταθούν από εναλλακτικές εφαρμογές εάν παραστεί ανάγκη.