
Δυναμική Δημιουργία και Διαχείριση Γραφημάτων στο Διαδίκτυο

ΡΟΣΙΟΥ ΔΗΜΗΤΡΑ



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Ηλεκτρονικών Υπολογιστών

Εξεταστική Επιτροπή

Καθηγητής Σταύρος Χριστοδουλάκης (Επιβλέπων)

Επ. Καθηγήτρια Αικατερίνη Μανιά

Επ. Καθηγητής Σαμολαδάς Βασίλης

Χανιά, 2008

Περίληψη

Τα γραφήματα είναι ένας αποδοτικός τρόπος αναπαράστασης δεδομένων. Χρησιμοποιούνται κυρίως για να καταστήσουν ευκολότερη την κατανόηση μεγάλου όγκου δεδομένων και τη σχέση μεταξύ διαφορετικών τμημάτων δεδομένων. Τα γραφήματα μπορούν να διαβαστούν πιο γρήγορα απ' ό,τι τα ίδια τα ακατέργαστα δεδομένα που απεικονίζουν, και ο λόγος που συμβαίνει αυτό είναι ότι μια εικόνα γίνεται πιο εύκολα κατανοητή από τον ανθρώπινο εγκέφαλο σε σχέση με το απλό κείμενο.

Τα γραφήματα χρησιμοποιούνται από μια πληθώρα εφαρμογών σε όλους τους τομείς της καθημερινότητας. Μερικές από αυτές τις εφαρμογές μπορεί να είναι: η οικονομική ανάπτυξη μιας εταιρείας μέσα σε μια χρονική περίοδο ή η πορεία των πωλήσεων της, τα αποτελέσματα μια έρευνας με βάση κάποια συγκεκριμένα κριτήρια, τα αποτελέσματα μιας δημοσκόπησης με βάση κάποια συγκεκριμένα κριτήρια, η διακύμανση των μετεωρολογικών μεταβλητών μέσα σε ένα χρονικό διάστημα, τα ποσοστιαία αποτελέσματα των εκλογών κ.ά. Γραφήματα είναι δυνατόν να δημιουργηθούν με το χέρι (σε ένα ειδικό χαρτί γραφικών αναπαραστάσεων) ή από τον υπολογιστή χρησιμοποιώντας ειδικού τύπου εφαρμογές.

Η χρήση των γραφημάτων βρίσκει εφαρμογή σε πολλές δραστηριότητες της καθημερινότητας και οι λόγοι που οδηγούν σε αυτό είναι ότι τα γραφήματα:

- είναι ένας εύχρηστος και εύληπτος τρόπος για να μεταφέρουν πληροφορία,
- είναι ένας γρήγορος τρόπος για να απεικονίσει κανείς αυτά που θέλει να πει
- είναι ένα ισχυρό εργαλείο για να δοθεί έμφαση σε συγκεκριμένα σημεία ή μηνύματα τα οποία πρέπει να σημειωθούν
- δίνουν μια πειστική ερμηνεία ως προς το αποτέλεσμα
- είναι πιο ενδιαφέροντα από το απλό κείμενο και γίνονται πιο γρήγορα κατανοητά από τον ανθρώπινο εγκέφαλο. [21]

Παρόλα όμως τα πλεονεκτήματα από τη χρήση των γραφημάτων, υπάρχουν σημαντικές δυσκολίες όσον αφορά τη δημιουργία τους. Ο σχεδιασμός ενός γραφήματος απαιτεί γνώση πολλών παραμέτρων. Μια από τις πιο σημαντικές αποφάσεις στη δημιουργία ενός γραφήματος, είναι η επιλογή του κατάλληλου τύπου γραφήματος ανάλογα με τη φύση των δεδομένων. Για παράδειγμα, δεδομένα που

αντιπροσωπεύουν ποσοστά κάποιων κατηγοριών, όπως «ευχαριστημένοι», «δυσανεστημένοι», «αδιάφοροι», αναπαριστώνται συνήθως σε ένα γράφημα πίτας. Όμως, τα αποτελέσματα γίνονται πιο εύκολα αντιληπτά όταν αναπαριστώνται σε ένα οριζόντιο ραβδόγραμμα. Μια σημαντική παράμετρος στην επιλογή του κατάλληλου τύπου γραφήματος είναι η συνέχεια των αξόνων. Για παράδειγμα για την αναπαράσταση του μέσου όρου ηλικίας των κατοίκων ανάμεσα στους νομούς της Ελλάδας, θα μπορούσε να χρησιμοποιηθεί μια γραμμή ως γράφημα. Σε αυτή την περίπτωση όμως θα υπήρχε ασυνέπεια των αξόνων από τον ένα νομό στον επόμενο διότι δεν θα μπορούσε να δοθεί μια ερμηνεία σχετικά με το τι απεικονίζεται σε αυτό το σημείο (ανάμεσα σε δύο συνεχόμενους νομούς). Αντίστοιχα, θα μπορούσε να χρησιμοποιηθεί ένα ραβδόγραμμα, το οποίο θα απεικονίζει επακριβώς τον μέσο όρο ηλικίας κάθε νομού.

Η δεύτερη σημαντική απόφαση στην δημιουργία ενός γραφήματος είναι η μορφοποίηση του. Τις περισσότερες φορές που οι άνθρωποι συναντούν ένα γράφημα σε μια εφημερίδα ή σε ένα βιβλίο, το πρώτο πράγμα που σκέφτονται είναι ότι πρόκειται για ένα ψέμα. Για να είναι πειστικό και σωστά παρουσιασμένο ένα γράφημα, πρέπει να ακολουθούνται κάποιες τεχνικές σχεδιασμού και κανόνες που αφορούν την μορφοποίηση του. Για παράδειγμα, ένα σωστά σχεδιασμένο γράφημα είναι εκείνο που : έχει τον τίτλο πάνω από τα δεδομένα και όχι σε κάποια άλλη θέση, έχει λεζάντα, υπάρχουν labels στους άξονες, έχει χρησιμοποιηθεί σωστός συνδυασμός χρωμάτων, αφιερώνει το περισσότερο μελάνι για τα δεδομένα κ.ά. Όλες οι αποφάσεις που πρέπει να ληφθούν σχετικά με την τελική παρουσίαση του γραφήματος είναι σημαντικές δεδομένου ότι επηρεάζουν την καλή ερμηνεία του γραφήματος, την έμφαση, την ακρίβεια, καθώς και την αισθητική του [21].

Όσον αφορά τη δημιουργία γραφημάτων με χρήση ηλεκτρονικού υπολογιστή υπάρχουν σήμερα αρκετά εργαλεία που μπορούν να χρησιμοποιηθούν. Διάφορα πακέτα εφαρμογών γραφείου (MS Office, Open Office, κλπ) δίνουν πολύ καλές δυνατότητες τόσο για τη δημιουργία όσο και για τη διαμόρφωση γραφημάτων. Τέτοιου είδους πακέτα όμως στοχεύουν στη δημιουργία γραφημάτων τα οποία και θα παρουσιάζονται μέσω του δικού τους περιβάλλοντος ή μπορούν να αποθηκεύονται (στην καλύτερη των περιπτώσεων) σε αρχεία.

Από την άλλη πλευρά, υπάρχουν πολλές εφαρμογές οι οποίες έχουν ανάγκη δημιουργίας και παρουσίασης γραφημάτων ως αναπόσπαστο μέρος αυτών. Μια τέτοια περίπτωση, θα μπορούσε να είναι οποιαδήποτε ιστοσελίδα, στη οποία χρειάζεται να παρουσιαστεί ένα γράφημα. Για τέτοιου είδους εφαρμογές απαιτείται κάποιο υποσύστημα το οποίο θα μπορεί να ενσωματωθεί σε αυτές και να παρέχει τη σχετική λειτουργικότητα. Για το σκοπό αυτό υπάρχουν πολλές βιβλιοθήκες γραφημάτων, που επιτρέπουν την δημιουργία γραφημάτων με τη βοήθεια μια γλώσσας προγραμματισμού. Κάτι τέτοιο όμως σημαίνει ότι απαιτούνται πολλές γραμμές κώδικα για τη δημιουργία ενός γραφήματος και για το λόγο η λύση αυτή δεν είναι εύχρηστη.

Η παρούσα διπλωματική εργασία έχει τρεις βασικούς στόχους: πρώτον, τη παροχή κατάλληλης εφαρμογής για τη δυναμική δημιουργία και μορφοποίηση πρότυπων γραφημάτων με γραφικό τρόπο σύμφωνα με τις προτιμήσεις των χρηστών. Τα πρότυπα αυτά γραφήματα θα είναι επαναχρησιμοποιήσιμα, δηλαδή θα μπορεί ο χρήστης να χρησιμοποιήσει τις παραμέτρους που επέλεξε για την παρουσίαση γραφημάτων σε διαφορετικές εφαρμογές, με διαφορετικές παραμέτρους και διαφορετικά δεδομένα. Δεύτερον, τη παροχή μιας υπηρεσίας διαδικτύου απεικόνισης δεδομένων σε πρότυπα γραφήματα, σε πραγματικό χρόνο από εφαρμογές-πελάτες οι οποίες επιθυμούν τη απεικόνιση πραγματικών δεδομένων και παρουσίαση αυτών πάνω σε ένα πρότυπο γράφημα. Τρίτον, την παροχή ενός πλήρους οδηγού δημιουργίας σωστών γραφημάτων από πλευράς παρουσίασης, μέσω σαφών κανόνων και βοήθειας δημιουργίας και μορφοποίησης. Ο οδηγός αυτός συμπληρώνεται και από μια πρότυπη βιβλιοθήκη έτοιμων προτύπων (templates) τα οποία έχουν παραχθεί με χρήση αυτών των κανόνων.

Για την επίτευξη των παραπάνω στόχων σχεδιάσαμε πρώτον, την εφαρμογή διαδικτύου Chart Template Builder, με την υποστήριξη χρηστών, που επιτρέπει την δημιουργία πρότυπων γραφημάτων με γραφικό τρόπο, μέσα από μια ακολουθία βημάτων. Μέσω αυτής της εφαρμογής, είναι δυνατή η δημιουργία μιας πληθώρας πρότυπων γραφημάτων καθώς και η επεξεργασία αυτών. Ο εκάστοτε χρήστης έχει τη δυνατότητα να βλέπει άμεσα τις αλλαγές που εφαρμόζει σε ένα πρότυπο γράφημα και να τις επεξεργάζεται περαιτέρω. Το αποτέλεσμα είναι για κάθε χρήστη να

αποθηκεύεται μια πληθώρα πρότυπων γραφημάτων, καθώς και όλες οι παράμετροι που το χαρακτηρίζουν.

Δεύτερον υλοποιήσαμε μια υπηρεσία ιστού (Web Service) [13], που εξυπηρετεί εφαρμογές-πελάτες για την απόδοση πραγματικών δεδομένων πάνω σε ένα πρότυπο γράφημα, σε πραγματικό χρόνο. Η βασική ιδέα της υλοποίησης της υπηρεσίας ιστού έγκειται στο γεγονός ότι σκοπός της παρούσας διπλωματικής ήταν η δημιουργία ενός είδους «εξυπηρετητή» που θα δέχεται τα δεδομένα προς απεικόνιση και θα ανακατασκευάζει πλήρως το γράφημα. Η εισαγωγή δεδομένων θα μπορούσε να γίνει μέσω του http request, στέλνοντας τα δεδομένα με κάποιο τρόπο. Εξαιτίας όμως της πολυπλοκότητας δόμησης του τρόπου εισαγωγής δεδομένων από την βιβλιοθήκη JFreeChart, προέκυψε η ανάγκη οργάνωσης των δεδομένων σε μια συγκεκριμένη δομή ανάλογα με τον τύπο γραφήματος. Για τον λόγο αυτό, σχεδιάσαμε ένα web service που δέχεται ως είσοδο ένα xml αρχείο με τα δεδομένα οργανωμένα με κατάλληλο τρόπο. Το web service χρησιμοποιείται από οποιαδήποτε εφαρμογή-πελάτη για την απεικόνιση πραγματικών δεδομένων πάνω σε ένα πρότυπο γράφημα. Το αποτέλεσμα είναι να επιστρέφει το τελικό γράφημα ως εικόνα.

Τέλος, βασισμένοι στο βιβλίο του Edward R. Tufte «The Visual Display of Quantitative Information» [21], υλοποιήσαμε έναν οδηγό δημιουργίας σωστών γραφημάτων μέσω κατάλληλων τεχνικών σχεδιασμού και κανόνων μορφοποίησης. Επιπλέον για την καλύτερη κατανόηση του οδηγού αυτού, υλοποιήσαμε και μια πρότυπη βιβλιοθήκη έτοιμων γραφημάτων, τα οποία έχουν παραχθεί ακολουθώντας αυτούς τους κανόνες και μπορούν να χρησιμοποιηθούν άμεσα ως πρότυπα γραφήματα από τον εκάστοτε χρήστη.

Αφιέρωση

Αφιέρωση

Στην οικογένεια μου, τους συγγενείς και όλους τους φίλους.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω, τον καθηγητή κ. Σταύρο Χριστοδουλάκη για την επίβλεψη και την καθοδήγηση του κατά την διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας. Θα ήθελα να τον ευχαριστήσω επίσης για τις πολύτιμες εμπειρίες που μου προσέφερε στο Εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών (MUSIC).

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον Νεκτάριο Γιολδάση για την επίβλεψη και την πολύτιμη βοήθεια, που ήταν πάντα πρόθυμος να προσφέρει σε όλη τη διάρκεια εκπόνησης της εργασίας.

Να ευχαριστήσω προκαταβολικά τον κ. Βασίλη Σαμολαδά και την κ. Αικατερίνη Μανιά για την ανάγνωση του κειμένου της παρούσας διπλωματικής.

Τέλος, ένα μεγάλο ευχαριστώ την οικογένεια μου και τους φίλους που με στήριξαν σε όλη τη διάρκεια των σπουδών και συνεχίζουν ακόμα!

Περιεχόμενα

ΔΥΝΑΜΙΚΗ ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΓΡΑΦΗΜΑΤΩΝ ΣΤΟ ΔΙΑΔΙΚΤΥΟ.....	1
ΡΟΣΙΟΥ ΔΗΜΗΤΡΑ.....	1
<u>ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ.....</u>	<u>1</u>
<u>ΠΕΡΙΛΗΨΗ.....</u>	<u>2</u>
<u>ΑΦΙΕΡΩΣΗ.....</u>	<u>1</u>
<u>ΕΥΧΑΡΙΣΤΙΕΣ.....</u>	<u>1</u>
<u>ΚΕΦΆΛΑΙΟ 1.....</u>	<u>1</u>
<u>ΕΙΣΑΓΩΓΗ.....</u>	<u>1</u>
<u>1.1 ΓΕΝΙΚΑ.....</u>	<u>1</u>
<u>1.2 ΣΥΝΕΙΣΦΟΡΑ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ.....</u>	<u>4</u>
<u>1.3 ΓΕΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ.....</u>	<u>4</u>
<u>1.4 ΔΟΜΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....</u>	<u>6</u>
<u>ΚΕΦΆΛΑΙΟ 2.....</u>	<u>9</u>
<u>ΑΝΆΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΥΜΠΕΡΙΦΟΡΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΕ ΤΗ ΧΡΗΣΗ ΠΕΡΙΠΤΩΣΕΩΝ</u>	
<u>ΧΡΗΣΗΣ </u>	<u>9</u>
<u>2.1 ΕΙΣΑΓΩΓΗ.....</u>	<u>9</u>
<u>2.2 ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ </u>	<u>9</u>
<u>2.3 ΠΕΡΙΓΡΑΦΗ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ.....</u>	<u>11</u>
<u>2.3.1 Περιγραφή Περιπτώσεων Χρήσης Εφαρμογής Δημιουργίας Πρότυπων</u>	
<u>Γραφημάτων.....</u>	<u>15</u>
<u>2.3.2 Περιπτώσεις Χρήσης Υπηρεσίας Ιστού.....</u>	<u>24</u>
<u>2.4 ΠΕΡΙΛΗΨΗ.....</u>	<u>25</u>
<u>ΚΕΦΆΛΑΙΟ 3.....</u>	<u>29</u>
<u>ΣΧΕΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ.....</u>	<u>29</u>
<u>3.1 ΕΙΣΑΓΩΓΗ.....</u>	<u>29</u>
<u>3.2 GOOGLE CHART CREATOR.....</u>	<u>29</u>
<u>3.3 SMART DRAW.....</u>	<u>29</u>
<u>3.4 ADVGRAPHGENERATOR.....</u>	<u>30</u>
<u>3.5 CEWOLF.....</u>	<u>30</u>
<u>3.6 JWEB CHART.....</u>	<u>30</u>
<u>3.7 MS EXCEL, OPEN OFFICE.....</u>	<u>31</u>
<u>3.8 JFREECHART.....</u>	<u>31</u>
<u>3.9 ΠΕΡΙΛΗΨΗ.....</u>	<u>31</u>

ΚΕΦΆΛΑΙΟ 4.....	33
ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΕΠΙΛΟΓΩΝ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗΣ ΕΝΟΣ ΓΡΑΦΗΜΑΤΟΣ.....	33
4.1 ΕΙΣΑΓΩΓΗ.....	33
4.2 ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ JFreeChart.....	34
4.2.1 Τύποι Γραφημάτων.....	34
4.2.2 Κατηγορίες Δεδομένων.....	35
4.2.3 Επιλογές παραμετροποίησης των χαρακτηριστικών ενός γραφήματος.....	36
4.3 ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΕΠΙΛΟΓΩΝ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗΣ ΓΙΑ ΚΆΘΕ ΤΥΠΟ ΓΡΑΦΗΜΑΤΟΣ.....	36
4.3.1 Επιλογές των γενικών παραμέτρων ενός γραφήματος.....	37
4.3.2 Επεξεργασία των παραμέτρων του τίτλου ενός γραφήματος.....	39
4.3.3 Προσθήκη και μορφοποίηση υπότιτλου στο γράφημα	43
4.3.4 Μορφοποίηση της λεζάντας του γραφήματος.....	46
4.3.5 Προσθήκη labels στη περιοχή απεικόνισης δεδομένων.....	51
4.3.6 Μορφοποίηση της περιοχής απεικόνισης δεδομένων.....	57
4.3.7 Μορφοποίηση των αξόνων του γραφήματος.....	68
4.3.8 Προσθήκη και μορφοποίηση annotation στο γράφημα.....	75
4.3.9 Προσθήκη και μορφοποίηση marker στο γράφημα.....	78
4.4 ΠΕΡΙΛΗΨΗ.....	81
ΚΕΦΆΛΑΙΟ 5.....	82
ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΓΡΑΦΙΚΗΣ ΔΙΕΠΑΦΗΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	82
5.1 ΕΙΣΑΓΩΓΗ.....	82
5.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΓΡΑΦΙΚΗΣ ΔΙΕΠΑΦΗΣ.....	82
5.3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΓΡΑΦΙΚΗΣ ΔΙΕΠΑΦΗΣ ΓΙΑ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	95
.....	97
5.4 ΠΕΡΙΛΗΨΗ.....	98
ΚΕΦΆΛΑΙΟ 6.....	99
ΤΕΧΝΟΛΟΓΪΕΣ ΥΛΟΠΟΙΗΣΗΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	99
6.1 ΕΙΣΑΓΩΓΗ.....	99
6.2 ΤΕΧΝΟΛΟΓΪΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΣΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	99
6.3 ΠΕΡΙΛΗΨΗ.....	113
ΚΕΦΆΛΑΙΟ 7.....	114
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	114
7.1 ΕΙΣΑΓΩΓΗ.....	114
7.2 ΓΕΝΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....	114

Περιεχόμενα

7.2.1 Το πρότυπο MVC.....	114
7.2.2 Αρχιτεκτονική της εφαρμογής δημιουργίας πρότυπων γραφημάτων.....	116
7.2.3 Αρχιτεκτονική της υπηρεσίας ιστού (Web Service).....	121
7.3 ΠΕΡΙΛΗΨΗ.....	126
ΚΕΦΑΛΑΙΟ 8.....	127
ΑΡΧΕΣ ΚΑΙ ΟΔΗΓΙΕΣ ΣΩΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΓΡΑΦΗΜΑΤΩΝ – ΨΗΦΙΑΚΗ ΒΙΒΛΙΟΘΗΚΗ.....	127
8.1 ΕΙΣΑΓΩΓΗ.....	127
8.2 ΑΡΧΕΣ ΣΧΕΔΙΑΣΜΟΥ ΓΡΑΦΗΜΑΤΩΝ.....	128
8.2.1 Η Τελειότητα στο γράφημα.....	128
8.2.2 Ακεραιότητα.....	129
8.3 ΘΕΩΡΙΑ ΤΩΝ ΓΡΑΦΗΜΑΤΩΝ ΔΕΔΟΜΕΝΩΝ.....	130
8.3.1 Πυκνότητα εμφάνισης των δεδομένων.....	130
8.3.2 Στοιχεία του γραφήματος με πολλές λειτουργίες.....	130
8.3.3 Περισσότερο μελάνι για τα δεδομένα – Λιγότερο για όλα τα υπόλοιπα	131
8.3.4 Περισσότερα από ένα μικρά γραφήματα στο ίδιο γράφημα.....	132
8.3.5 Επιλογή χρωμάτων	132
8.4 ΕΠΙΛΟΓΗ ΤΥΠΟΥ ΓΡΑΦΗΜΑΤΟΣ.....	133
8.5 ΨΗΦΙΑΚΗ ΒΙΒΛΙΟΘΗΚΗ ΓΡΑΦΗΜΑΤΩΝ.....	135
ΚΕΦΑΛΑΙΟ 9.....	136
ΑΝΑΚΕΦΑΛΑΙΩΣΗ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	136
9.1 ΑΝΑΚΕΦΑΛΑΙΩΣΗ.....	136
9.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	138
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	139
ΠΑΡΑΡΤΗΜΑ Α.....	141

Περιεχόμενα Πινάκων

ΠΙΝΑΚΑΣ 1 - ΔΗΜΙΟΥΡΓΙΑ ΛΟΓΑΡΙΑΣΜΟΥ.....	16
ΠΙΝΑΚΑΣ 2 - ΈΛΕΓΧΟΣ ΣΤΟΙΧΕΙΩΝ ΦΌΡΜΑΣ.....	16
ΠΙΝΑΚΑΣ 3 - ΣΥΝΔΕΣΗ ΣΤΟ ΣΥΣΤΗΜΑ.....	17
ΠΙΝΑΚΑΣ 4 - ΑΠΟΣΥΝΔΕΣΗ ΑΠΌ ΤΟ ΣΥΣΤΗΜΑ.....	18
ΠΙΝΑΚΑΣ 5 - ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΛΟΓΑΡΙΑΣΜΟΥ.....	18
ΠΙΝΑΚΑΣ 6 - ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ ΠΡΌΤΥΠΟΥ ΓΡΑΦΉΜΑΤΟΣ	19
ΠΙΝΑΚΑΣ 7 - ΔΙΑΧΕΪΡΙΣΗ ΤΩΝ ΠΡΌΤΥΠΩΝ ΓΡΑΦΗΜΆΤΩΝ.....	20
ΠΙΝΑΚΑΣ 8 - ΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ ΠΑΡΑΜΈΤΡΩΝ ΤΟΥ ΠΡΌΤΥΠΟΥ ΓΡΑΦΉΜΑΤΟΣ.....	21
ΠΙΝΑΚΑΣ 9 - ΕΠΙΛΟΓΉ ΓΡΑΦΉΜΑΤΟΣ ΑΠΌ ΤΗ ΒΙΒΛΙΟΘΉΚΗ ΓΡΑΦΗΜΆΤΩΝ.....	21
ΠΙΝΑΚΑΣ 10 - ΔΙΑΧΕΪΡΙΣΗ ΑΙΤΗΜΆΤΩΝ ΓΙΑ ΝΕΟ ΛΟΓΑΡΙΑΣΜΌ.....	22
ΠΙΝΑΚΑΣ 11 - ΈΛΕΓΧΟΣ ΓΙΑ ΝΈΑ ΑΙΤΉΜΑΤΑ.....	23
ΠΙΝΑΚΑΣ 12 - ΠΡΟΣΘΉΚΗ ΠΡΌΤΥΠΟΥ ΓΡΑΦΉΜΑΤΟΣ ΣΤΗΝ ΨΗΦΙΑΚΉ ΒΙΒΛΙΟΘΉΚΗ.....	23
ΠΙΝΑΚΑΣ 13 - ΕΠΙΛΟΓΉ ΓΡΑΦΉΜΑΤΟΣ ΓΙΑ ΑΠΕΙΚΟΝΙΣΗ ΔΕΔΟΜΈΝΩΝ.....	24
ΠΙΝΑΚΑΣ 14 - ΥΠΟΣΤΗΡΙΖΟΜΕΝΟΙ ΤΎΠΟΙ ΓΡΑΦΗΜΆΤΩΝ.....	34
ΠΙΝΑΚΑΣ 15 - ΚΑΤΗΓΟΡΙΕΣ ΔΕΔΟΜΈΝΩΝ.....	36
ΠΙΝΑΚΑΣ 16 - ΥΠΟΣΤΗΡΙΖΟΜΕΝΑ ΧΡΏΜΑΤΑ	37
ΠΙΝΑΚΑΣ 17 - ΕΠΙΛΟΓΈΣ ΓΡΑΜΜΑΤΟΣΕΙΡΆΣ.....	42
ΠΙΝΑΚΑΣ 18 - ΥΠΟΣΤΗΡΙΖΟΜΕΝΟΙ ΤΎΠΟΙ PLOT.....	58

Περιεχόμενα Εικόνων

ΕΙΚΟΝΑ 1 - ΓΕΝΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	5
ΕΙΚΟΝΑ 2 - ΔΙΆΓΡΑΜΜΑ ΑΠΑΙΤΗΣΕΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΤΩΝ USE CASES.....	13
ΕΙΚΟΝΑ 3 - ΙΕΡΑΡΧΙΑ ΚΛΆΣΕΩΝ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ JFREECHART.....	34
ΕΙΚΟΝΑ 4 - ΙΕΡΑΡΧΙΑ ΤΩΝ ΔΕΔΟΜΈΝΩΝ.....	35
ΕΙΚΟΝΑ 5 - ΠΡΟΣΘΗΚΗ BORDER ΣΕ ΧΡΩΜΑ ΓΚΡΙ ΚΑΙ STROKE.....	38
ΕΙΚΟΝΑ 6 - ΑΛΛΑΓΗ ΤΟΥ BACKGROUND ΧΡΩΜΑΤΟΣ ΣΕ DARK CYAN.....	39
ΕΙΚΟΝΑ 7 - ΑΛΛΑΓΗ ΤΗΣ ΘΈΣΗΣ ΤΟΥ ΤΊΤΛΟΥ.....	40
ΕΙΚΟΝΑ 8 - ΑΛΛΑΓΗ ΤΟΥ ΧΡΩΜΑΤΟΣ ΤΟΥ ΤΊΤΛΟΥ.....	41
ΕΙΚΟΝΑ 9 - ΑΛΛΑΓΗ ΤΟΥ BACKGROUND ΧΡΩΜΑΤΟΣ ΤΟΥ ΤΊΤΛΟΥ.....	42
ΕΙΚΟΝΑ 10 - ΑΛΛΑΓΗ ΤΗΣ ΓΡΑΜΜΑΤΟΣΕΊΡΑΣ ΤΟΥ ΤΊΤΛΟΥ.....	43
ΕΙΚΟΝΑ 11 - ΑΛΛΑΓΗ ΤΗΣ ΘΈΣΗΣ ΤΟΥ ΥΠΌΤΙΤΛΟΥ.....	45
ΕΙΚΟΝΑ 12 - ΠΡΟΣΘΗΚΗ BORDER ΓΥΡΩ ΑΠΌ ΤΟΝ ΥΠΌΤΙΤΛΟ.....	46
ΕΙΚΟΝΑ 13 - ΑΛΛΑΓΗ ΤΗΣ ΘΈΣΗΣ ΤΗΣ ΛΕΖΆΝΤΑΣ (ΔΕΞΊΑ).....	47
ΕΙΚΟΝΑ 14 - ΕΠΕΞΕΡΓΑΣΊΑ ΤΩΝ ΧΡΩΜΆΤΩΝ ΤΗΣ ΛΕΖΆΝΤΑΣ.....	48
ΕΙΚΟΝΑ 15 - ΠΡΟΣΘΗΚΗ BORDER ΓΥΡΩ ΑΠΌ ΤΗ ΛΕΖΆΝΤΑ.....	49
ΕΙΚΟΝΑ 16 - ΑΛΛΑΓΗ ΤΗΣ ΘΈΣΗΣ ΤΩΝ ΣΥΜΒΟΛΩΝ ΤΗΣ ΛΕΖΆΝΤΑΣ.....	50
ΕΙΚΟΝΑ 17 - ΑΛΛΑΓΗ ΤΗΣ ΓΡΑΜΜΑΤΟΣΕΊΡΑΣ ΤΗΣ ΛΕΖΆΝΤΑΣ.....	51
ΕΙΚΟΝΑ 18 - ΠΡΟΣΘΗΚΗ LABEL ΣΤΟ ΓΡΆΦΗΜΑ.....	52
ΕΙΚΟΝΑ 19 - ΠΡΟΣΘΗΚΗ LABEL ΣΕ ΈΝΑ ΜΌΝΟ SERIES.....	53
ΕΙΚΟΝΑ 20 - ΑΛΛΑΓΗ ΧΡΩΜΑΤΟΣ ΤΩΝ LABELS.....	54
ΕΙΚΟΝΑ 21 - ΑΛΛΑΓΗ ΓΡΑΜΜΑΤΟΣΕΊΡΑΣ ΣΤΑ LABELS.....	55
ΕΙΚΟΝΑ 22 - ΤΡΌΠΟΣ ΕΜΦΆΝΙΣΗΣ ΤΩΝ LABELS.....	56
ΕΙΚΟΝΑ 23 - ΔΊΆΦΟΡΟΙ ΤΡΌΠΟΙ ΕΜΦΆΝΙΣΗΣ ΤΩΝ LABELS.....	57
ΕΙΚΟΝΑ 24 - ΔΙΑΧΩΡΙΣΜΌΣ ΤΩΝ ΠΕΡΙΟΧΩΝ ΕΝΌΣ ΓΡΆΦΗΜΑΤΟΣ.....	58
ΕΙΚΟΝΑ 25 - ΟΡΙΖΌΝΤΙΑ ΚΑΤΈΥΘΥΝΣΗ ΤΟΥ PLOT.....	59
ΕΙΚΟΝΑ 26 - ΠΡΟΣΘΗΚΗ ΓΡΑΜΜΏΝ ΠΛΈΓΜΑΤΟΣ ΣΤΟ ΓΡΆΦΗΜΑ.....	60
ΕΙΚΟΝΑ 27 - ΠΡΟΣΘΗΚΗ OFFSET ΣΤΟ PLOT.....	61
ΕΙΚΟΝΑ 28 - ΠΡΟΣΘΗΚΗ ΤΕΤΑΡΤΗΜΟΊΟΥ ΣΤΟ PLOT.....	62

Περιεχόμενα Εικόνων

ΕΙΚΟΝΑ 29 - ΚΑΘΟΡΙΣΜΕΝΗ ΚΑΤΕΥΘΥΝΣΗ ΠΙΤΑΣ.....	63
ΕΙΚΟΝΑ 30 - ΑΛΛΑΓΗ ΚΑΤΕΥΘΥΝΣΗ ΠΙΤΑΣ.....	63
ΕΙΚΟΝΑ 31 - ΠΡΟΣΘΗΚΗ ΣΚΙΑΣ ΓΥΡΩ ΑΠΟ ΤΗ ΠΙΤΑ	64
ΕΙΚΟΝΑ 32 - POLAR ΓΡΑΦΗΜΑ.....	65
ΕΙΚΟΝΑ 33 - ΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ GRID LINES.....	66
ΕΙΚΟΝΑ 34 - MULTIPLE ΡΙΕ ΓΡΑΦΗΜΑ.....	67
ΕΙΚΟΝΑ 35 - ΕΠΕΞΕΡΓΑΣΙΑ ΤΟΥ PLOT.....	68
ΕΙΚΟΝΑ 36 - ΕΠΕΞΕΡΓΑΣΙΑ ΤΟΥ ΆΞΟΝΑ Χ.....	69
ΕΙΚΟΝΑ 37 - ΑΛΛΑΓΗ ΤΟΥ ΧΡΩΜΑΤΟΣ ΤΗΣ ΕΤΙΚΕΤΑΣ ΤΟΥ ΆΞΟΝΑ Χ.....	70
ΕΙΚΟΝΑ 38 - ΑΛΛΑΓΗ ΤΗΣ ΘΕΣΗΣ ΤΟΥ ΆΞΟΝΑ Χ.....	71
ΕΙΚΟΝΑ 39 - ΑΛΛΑΓΗ ΤΗΣ ΘΕΣΗΣ ΤΩΝ LABELS ΤΟΥ ΆΞΟΝΑ Χ.....	72
ΕΙΚΟΝΑ 40 - ΠΡΟΣΘΗΚΗ ARROW ΣΤΟΝ ΆΞΟΝΑ Υ.....	73
ΕΙΚΟΝΑ 41 - ΑΛΛΑΓΗ ΤΟΥ ΕΥΡΟΥΣ ΤΙΜΩΝ ΤΟΥ ΆΞΟΝΑ Υ.....	74
ΕΙΚΟΝΑ 42 - ΑΝΤΙΣΤΡΟΦΗ ΦΟΡΑ ΕΜΦΑΝΙΣΗΣ ΤΙΜΩΝ ΤΟΥ ΆΞΟΝΑ Υ.....	75
ΕΙΚΟΝΑ 43 - LINE ANNOTATION.....	76
ΕΙΚΟΝΑ 44 - POINTER ANNOTATION.....	77
ΕΙΚΟΝΑ 45 - TEXT ANNOTATION.....	78
ΕΙΚΟΝΑ 46 - RANGE MARKER.....	79
ΕΙΚΟΝΑ 47 – DOMAIN LINE MARKER	80
ΕΙΚΟΝΑ 48 - DOMAIN MARKER.....	81
ΕΙΚΟΝΑ 49 – ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ.....	83
ΕΙΚΟΝΑ 50 - ΠΡΟΣΩΠΙΚΗ ΣΕΛΙΔΑ ΧΡΗΣΤΗ.....	84
ΕΙΚΟΝΑ 51 – ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ ΣΤΗΝ ΠΡΟΣΩΠΙΚΗ ΣΕΛΙΔΑ ΧΡΗΣΤΗ.....	85
ΕΙΚΟΝΑ 52 – ΤΑ ΠΡΟΤΥΠΑ ΓΡΑΦΗΜΑΤΑ ΕΝΟΣ ΧΡΗΣΤΗ ΚΑΙ ΟΙ ΕΠΙΛΟΓΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΑΥΤΩΝ.....	86
ΕΙΚΟΝΑ 53 - ΒΑΣΙΚΟ ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ.....	86
ΕΙΚΟΝΑ 54 - ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ ΓΡΑΦΗΜΑΤΟΣ.....	87
ΕΙΚΟΝΑ 55 – ΚΑΘΟΡΙΣΜΟΣ ΒΑΣΙΚΩΝ ΠΑΡΑΜΕΤΡΩΝ ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΓΡΑΦΗΜΑΤΟΣ.....	89

Περιεχόμενα Εικόνων

ΕΙΚΟΝΑ 56 - PREVIEW TEMPLATE.....	89
ΕΙΚΟΝΑ 57 – ΒΑΣΙΚΟ ΜΕΝΟΥ ΠΛΩΗΓΗΣΗΣ ΓΙΑ ΤΗΝ ΕΠΕΞΕΡΓΑΣΙΑ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ ΓΡΑΦΗΜΑΤΟΣ.....	90
ΕΙΚΟΝΑ 58 - ΕΠΙΛΟΓΕΣ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗΣ ΤΟΥ ΤΙΤΛΟΥ ΤΟΥ ΓΡΑΦΗΜΑΤΟΣ.....	91
ΕΙΚΟΝΑ 59 - ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΨΗΦΙΑΚΗΣ ΒΙΒΛΙΟΘΗΚΗΣ.....	92
ΕΙΚΟΝΑ 60 – ΠΛΗΡΟΦΟΡΙΕΣ ΚΑΘΕ ΓΡΑΦΗΜΑΤΟΣ ΚΑΙ ΔΙΑΘΕΣΙΜΑ ΑΡΧΕΙΑ.....	93
ΕΙΚΟΝΑ 61 - ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΛΟΓΑΡΙΑΣΜΟΥ.....	94
ΕΙΚΟΝΑ 62 - ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ ΛΟΓΑΡΙΑΣΜΟΥ.....	95
ΕΙΚΟΝΑ 63 - ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΑΙΤΗΜΑΤΩΝ ΓΙΑ ΝΕΟ ΛΟΓΑΡΙΑΣΜΟ ΑΠΟ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ.....	96
ΕΙΚΟΝΑ 64 - ΒΑΣΙΚΟ ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ ΠΛΩΗΓΗΣΗΣ ΓΙΑ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ.....	97
ΕΙΚΟΝΑ 65 - ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΠΡΟΤΥΠΩΝ ΓΡΑΦΗΜΑΤΩΝ ΓΙΑ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ.....	97
ΕΙΚΟΝΑ 66 - ΤΟ ΔΙΑΓΡΑΜΜΑ ΑΝΤΙΠΡΟΣΩΠΕΥΕΙ ΤΙΣ ΣΧΕΣΕΙΣ ΜΕΤΑΞΥ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΤΟΥ DAO.....	108
ΕΙΚΟΝΑ 67 - ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ ΥΠΗΡΕΣΙΩΝ ΙΣΤΟΥ [12].....	109
ΕΙΚΟΝΑ 68 - ΔΟΜΗ WSDL ΈΓΓΡΑΦΟΥ [12].....	111
ΕΙΚΟΝΑ 69 - ΔΟΜΗ SOAP ΜΗΝΥΜΑΤΟΣ [12].....	112
ΕΙΚΟΝΑ 70 - ΠΡΟΤΥΠΟ MVC.....	115
ΕΙΚΟΝΑ 71 - ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΠΡΟΤΥΠΟ MVC.....	117
ΕΙΚΟΝΑ 72 – ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	120
ΕΙΚΟΝΑ 73 - DAO PATTERN ΓΙΑ ΤΗ ΣΧΕΣΙΑΚΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	121
ΕΙΚΟΝΑ 74 - ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΥΠΗΡΕΣΙΑ ΙΣΤΟΥ.....	122
ΕΙΚΟΝΑ 75 - WSDL ΈΓΓΡΑΦΟ ΤΟΥ WEB SERVICE ΠΟΥ ΣΧΕΔΙΑΣΑΜΕ.....	124
ΕΙΚΟΝΑ 76 - SOAP ΜΗΝΥΜΑ ΚΑΤΑ ΤΗΝ ΚΛΗΣΗ ΤΟΥ WEB SERVICE ΜΕ ΕΙΣΟΔΟ ΈΝΑ XML ΑΡΧΕΙΟ.....	125
ΕΙΚΟΝΑ 77 - SOAP ΜΗΝΥΜΑ ΚΑΤΑ ΤΗΝ ΈΞΟΔΟ ΤΟΥ WEB SERVICE.....	126

Κεφάλαιο 1

Εισαγωγή

1.1 Γενικά

Τα γραφήματα αποτελούν την οπτική αναπαράσταση δεδομένων, τα οποία αναπαριστώνται από σύμβολα όπως ράβδους σε ένα ραβδόγραμμα ή ιστόγραμμα, γραμμές σε μια γραμμή, κομμάτια πίτας κ.ά. Ένα γράφημα μπορεί να αναπαραστήσει συνοπτικά αριθμητικά δεδομένα ή συναρτήσεις.

Τα γραφήματα χρησιμοποιούνται συχνά για να καταστήσουν ευκολότερη την κατανόηση μεγάλου όγκου δεδομένων και τη σχέση μεταξύ διαφορετικών τμημάτων δεδομένων. Η χρήση τους βρίσκει εφαρμογή σε πολλούς τομείς της ανθρώπινης δραστηριότητας και οι λόγοι που συμβαίνει αυτό είναι ότι τα γραφήματα :

- είναι ένας εύχρηστος και εύληπτος τρόπος για να μεταφέρουν πληροφορία,
- είναι ένας γρήγορος τρόπος για να απεικονίσει κανείς αυτά που θέλει να πει
- είναι ένα ισχυρό εργαλείο για να δοθεί έμφαση σε συγκεκριμένα σημεία ή μηνύματα τα οποία πρέπει να σημειωθούν
- δίνουν μια πειστική ερμηνεία ως προς το αποτέλεσμα
- είναι πιο ενδιαφέροντα από το απλό κείμενο και γίνονται πιο γρήγορα κατανοητά από τον ανθρώπινο εγκέφαλο. [21]

Για όλους αυτούς τους λόγους έχουν αναπτυχθεί ειδικές εφαρμογές που επιτρέπουν την ανάπτυξη γραφημάτων κυρίως με τη χρήση ηλεκτρονικού υπολογιστή. Ωστόσο, όσο εύκολη και αν φαίνεται η ανάπτυξη γραφημάτων μέσω ειδικών εφαρμογών, υπάρχουν κάποια σημαντικά προβλήματα στη χρήση τους.

Πιο συγκεκριμένα, υπάρχουν σήμερα αρκετά εργαλεία που μπορούν να χρησιμοποιηθούν για την ανάπτυξη γραφημάτων, όπως είναι το MS Excel, το Open Office κ.ά. Τέτοια εργαλεία όμως επιτρέπουν την δημιουργία γραφημάτων που θα παρουσιάζονται μόνο μέσω του δικού τους περιβάλλοντος. Από την άλλη πλευρά, υπάρχουν πολλές εφαρμογές που χρειάζεται να χρησιμοποιούν γραφήματα ως μέρος αυτών, όπως για παράδειγμα η παρουσίαση γραφήματος σε μια ιστοσελίδα. Πρακτικά, κάτι τέτοιο σημαίνει ότι υπάρχει η ανάγκη δημιουργίας ενός

υποσυστήματος που να επιτρέπει την ανάπτυξη γραφημάτων και παρουσίαση αυτών σε οποιοδήποτε περιβάλλον.

Επιπλέον, η δημιουργία ενός σωστά σχεδιασμένου γραφήματος απαιτεί τη γνώση πολλών παραμέτρων. Η πρώτη σημαντική παράμετρος για την ανάπτυξη ενός γραφήματος είναι η επιλογή του κατάλληλου τύπου γραφήματος ανάλογα με τη φύση των δεδομένων. Υπάρχουν πολλοί τύποι γραφημάτων καθένας από τους οποίους είναι κατάλληλος για συγκεκριμένες κατηγορίες εφαρμογών. Για παράδειγμα, για την αναπαράσταση των ποσοστών που έλαβαν οι απαντήσεις των πολιτών σε μια δημοσκόπηση, θα μπορούσε να χρησιμοποιηθεί μια πίτα. Με τη χρήση της πίτας ως τύπο γραφήματος θα απεικονίζεται η κατανομή των αποτελεσμάτων, όμως αν επιλεγθεί ένα ραβδόγραμμα για την αναπαράσταση των δεδομένων, τα δεδομένα θα γίνονται πιο εύκολα αντιληπτά από το ανθρώπινο μάτι και θα είναι πιο εύκολη η σύγκριση των αποτελεσμάτων. Ένα άλλο χαρακτηριστικό που καταδεικνύει την σημασία σωστής επιλογής τύπου γραφήματος, είναι η συνέχεια των αξόνων. Έστω ότι τα δεδομένα προς απεικόνιση είναι το ποσοστό επιτυχόντων φοιτητών σε όλα τα μαθήματα ενός τμήματος μιας σχολής. Σε αυτή την περίπτωση θα μπορούσε να χρησιμοποιηθεί μια γραμμή, που κάθε σημείο θα δείχνει το ποσοστό επιτυχόντων φοιτητών για το αντίστοιχο μάθημα και όλα τα σημεία θα ενώνονται από αυτή τη γραμμή. Όμως ανάμεσα σε δυο συνεχόμενα μαθήματα, δε θα μπορεί να δοθεί κάποια ερμηνεία καθώς δεν υπάρχει κάποια σχέση μεταξύ τους. Μια καλή επιλογή σε αυτή την περίπτωση είναι το ραβδόγραμμα, που θα απεικονίζει επακριβώς το ποσοστό επιτυχόντων φοιτητών για κάθε μάθημα.

Μια άλλη σημαντική παράμετρος που αφορά το σωστό σχεδιασμό ενός γραφήματος είναι η μορφοποίηση του. Ένα γράφημα μπορεί να έχει μια πληθώρα χαρακτηριστικών, εντούτοις υπάρχουν κοινά χαρακτηριστικά γνωρίσματα που παρέχουν σε ένα γράφημα τη δυνατότητα να αντιπροσωπεύει την ουσία των δεδομένων που αναπαριστά. Τυπικά, ένα γράφημα είναι μια εικόνα που έχει πολύ λίγο κείμενο, δεδομένου ότι το ανθρώπινο μάτι είναι ικανό να συμπεράνει την ουσία πιο γρήγορα και πιο εύκολα από την εικόνα παρά από το κείμενο.

Ένα σημαντικό παράδειγμα χρήσης κειμένου σε ένα γράφημα είναι ο τίτλος του. Ο τίτλος του βρίσκεται συνήθως πάνω από το γράφημα και συνήθως είναι μια σύντομη περιγραφή σχετικά με το τι είναι αυτό που αναπαριστά το συγκεκριμένο

γράφημα. Κάθε γράφημα αποτελείται από δύο άξονες καθένας από τους οποίους υποδεικνύει το εύρος τιμών των δεδομένων. Κάθε άξονας μπορεί να είναι μια κλίμακα αριθμητικών δεδομένων, που συνοδεύεται συνήθως από κάποια ετικέτα (label). Μέσα στην περιοχή απεικόνισης δεδομένων ενός γραφήματος, μπορούν να προστεθούν γραμμές πλέγματος. Το πλέγμα σε ένα γράφημα ενισχύει την κατανόηση των αποτελεσμάτων με περισσότερη ακρίβεια. Τα δεδομένα σε ένα γράφημα απεικονίζονται συνήθως με διάφορα σημεία ή σχήματα και σε διάφορα χρώματα για να είναι πιο έντονη η διάκριση μεταξύ τους. Για την αντιστοιχία μεταξύ των σχημάτων ή των χρωμάτων με τα δεδομένα που απεικονίζουν, προστίθεται στο γράφημα ένα υπόμνημα (legend) που δείχνει αυτή ακριβώς την αντιστοιχία [1]. Η προσθήκη και η μορφοποίηση των χαρακτηριστικών αυτών καθορίζουν σημαντικά την ακρίβεια, την σαφήνεια και την αισθητική του γραφήματος. Για αυτό και υπάρχουν συγκεκριμένες τεχνικές σχεδιασμού και κανόνες μορφοποίησης που πρέπει να ακολουθούνται κατά την ανάπτυξη ενός γραφήματος.

Από τα παραπάνω προκύπτει το συμπέρασμα ότι η ανάπτυξη ενός γραφήματος χρίζει ιδιαίτερης προσοχής και σημασίας καθώς υπάρχει μια πληθώρα παραμέτρων που πρέπει να ληφθούν υπόψη κατά το σχεδιασμό αυτών και κάποια προβλήματα που πρέπει να ξεπεραστούν. Επιπλέον, η ανάπτυξη ενός γραφήματος θα πρέπει να γίνεται με τέτοιο τρόπο ώστε να είναι εύχρηστος προς οποιονδήποτε χρήστη.

Η παρούσα διπλωματική εργασία έχει τρεις βασικούς στόχους: πρώτον, τη παροχή κατάλληλης εφαρμογής για τη δημιουργία και μορφοποίηση πρότυπων γραφημάτων με γραφικό τρόπο τα οποία θα είναι επαναχρησιμοποιήσιμα. Δεύτερον, τη παροχή μιας υπηρεσίας γραφικής απόδοσης γραφημάτων σε πραγματικό χρόνο από εφαρμογές-πελάτες οι οποίες επιθυμούν τη γραφική απόδοση και παρουσίαση αυτών για δικά τους δεδομένα. Τρίτον, την παροχή ενός πλήρους οδηγού δημιουργίας σωστών γραφημάτων μέσω σαφών κανόνων και βοήθειας δημιουργίας και μορφοποίησης. Ο οδηγός αυτός συμπληρώνεται και από μια πρότυπη βιβλιοθήκη έτοιμων προτύπων (templates) τα οποία έχουν παραχθεί με χρήση αυτών των κανόνων.

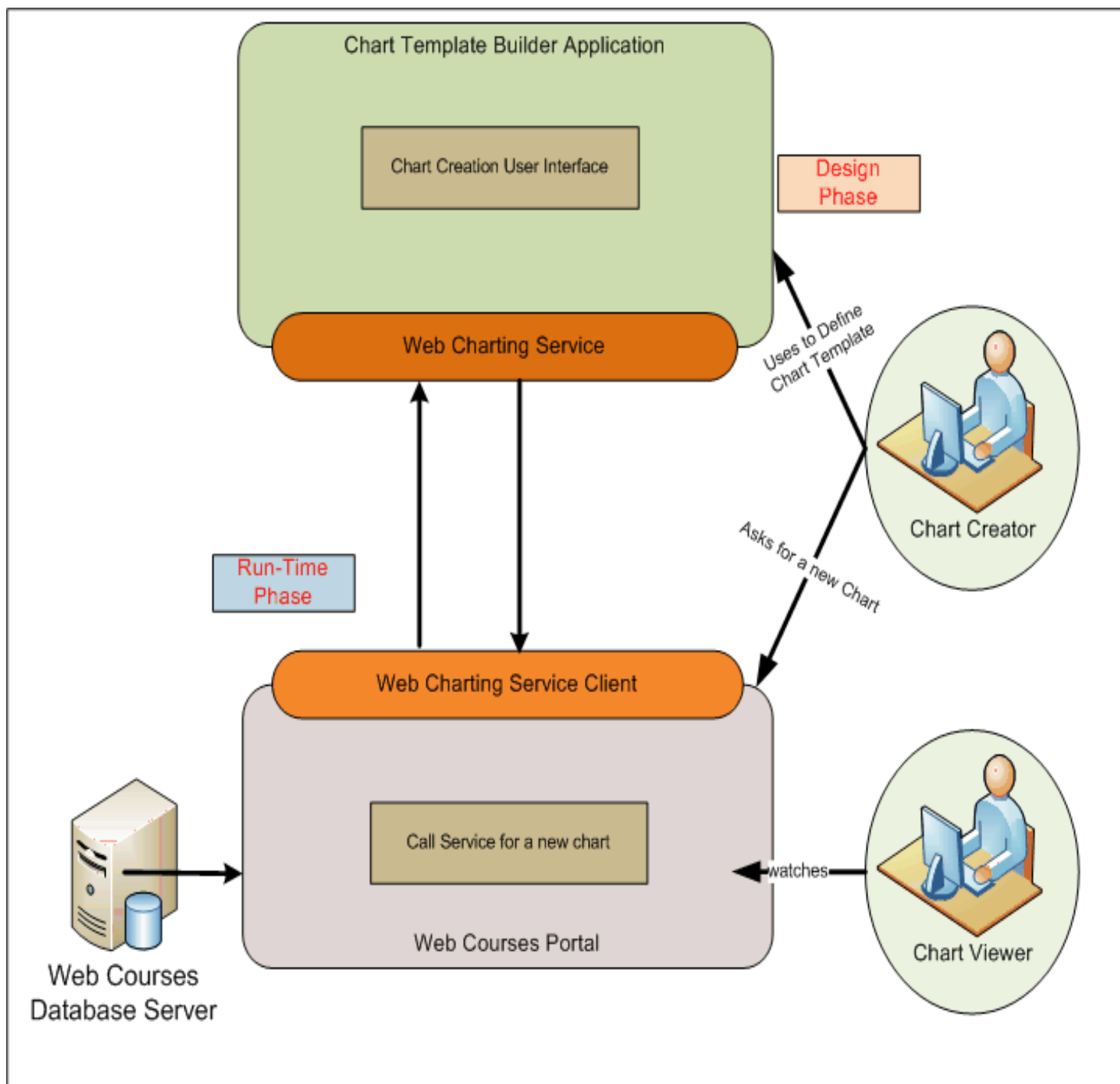
1.2 Συνεισφορά της διπλωματικής εργασίας

Η συνεισφορά της παρούσας διπλωματικής εργασίας είναι η εξής:

- Δημιουργία γραφημάτων με γραφικό τρόπο, χωρίς να χρειάζονται γραμμές κώδικα.
- Καταγραφή και δυνατότητα γραφικού ορισμού όλων των παραμέτρων για το σχεδιασμό ολοκληρωμένων γραφημάτων
- Δημιουργία επαναχρησιμοποιήσιμων πρότυπων γραφημάτων
- Ανάπτυξη ενός πλαισίου εργασίας αποτελούμενο από κανόνες δημιουργίας και διαμόρφωσης σωστών γραφημάτων, καθώς και έτοιμης βιβλιοθήκης με πρότυπα γραφήματα, ώστε να είναι δυνατή η επιλογή ενός έτοιμου γραφήματος για την απεικόνιση δεδομένων
- Ανάπτυξη μιας Υπηρεσίας Διαδικτύου για τη γραφική απόδοση γραφημάτων σε πραγματικό χρόνο για εφαρμογές – πελάτες που έχουν ανάγκη σχετικής λειτουργικότητας.
- Γραφική απόδοση στατιστικών γραφημάτων για τους δείκτες απόδοσης της εκπαιδευτικής διαδικασίας στο έργο ΕΠΕΑΕΚ II «Ενίσχυση Σπουδών Πληροφορικής στο τμήμα ΗΜΜΥ του Πολυτεχνείου Κρήτης»

1.3 Γενική Προσέγγιση

Πριν προχωρήσουμε στην περιγραφή της παρούσας διπλωματικής εργασίας, θα παρουσιάσουμε την γενική προσέγγιση που ακολουθήθηκε για την επίτευξη των στόχων της εργασίας. Όπως φαίνεται και στην εικόνα 1 παρακάτω, η ανάπτυξη ενός νέου γραφήματος περιλαμβάνει δύο φάσεις: τη φάση σχεδιασμού – υλοποίησης (Design Phase) και τη φάση εκτέλεσης (Run-Time Phase).



Εικόνα 1 - Γενική αρχιτεκτονική της εφαρμογής

Κατά τη φάση του σχεδιασμού ο χρήστης χρησιμοποιεί μια εξειδικευμένη εφαρμογή (Chart Template Builder) στην οποία και καθορίζει με γραφικό τρόπο τον τύπο του γραφήματος καθώς και όλες τις παραμέτρους που έχουν να κάνουν με τη διαμόρφωσή του. Το τελικό αποτέλεσμα είναι ένα πρότυπο γραφήματος (chart template) στο οποίο έχουν καταγραφεί όλες οι επιλογές του χρήστη σχετικά με τη

διαμόρφωσή του. Αυτό το πρότυπο γραφήματος θα είναι διαθέσιμο προς χρήση από εφαρμογές που επιθυμούν να χρησιμοποιήσουν γραφήματα που ακολουθούν αυτό το πρότυπο.

Αντίστοιχα, κατά τη φάση υλοποίησης οι εφαρμογές-πελάτες χρησιμοποιούν μια υπηρεσία ιστού που έχουμε υλοποιήσει (Chart Builder Service) για να απεικονίσουν τα δικά τους πραγματικά δεδομένα πάνω σε ένα πρότυπο γράφημα, που έχει αναπτυχθεί κατά την προηγούμενη φάση, σε πραγματικό χρόνο. Το αποτέλεσμα είναι το ολοκληρωμένο πλέον γράφημα, διαθέσιμο για χρήση σε οποιαδήποτε εφαρμογή.

Τέλος, στην παραπάνω εικόνα παρουσιάζονται και οι χρήστες που υποστηρίζει κάθε τμήμα της εφαρμογής. Αναλυτική περιγραφή των τμημάτων της εφαρμογής καθώς και των χρηστών που υποστηρίζει κάθε ένα από αυτά, ακολουθεί σε επόμενα κεφάλαια.

1.4 Δομή της εργασίας

Στο δεύτερο κεφάλαιο, γίνεται η ανάλυση απαιτήσεων της εφαρμογής με τη χρήση των Use Cases [2] . Στο κεφάλαιο αυτό, παρουσιάζονται αναλυτικά οι χρήστες που υποστηρίζει ολόκληρη η εφαρμογή που σχεδιάσαμε καθώς και οι επιλογές που έχει ο καθένας από αυτούς.

Ακολουθεί το τρίτο κεφάλαιο, στο οποίο γίνεται μια παρουσίαση των σχετικών εργασιών που υπάρχουν σήμερα όσον αφορά τη δημιουργία γραφημάτων. Για κάθε μια από τις εργασίες αυτές αναφέρονται τα πλεονεκτήματα και τα μειονεκτήματα τους, ώστε να είναι εμφανείς οι λόγοι που μας οδήγησαν να σχεδιάσουμε την παρούσα εργασία.

Στο τέταρτο κεφάλαιο γίνεται παρουσίαση του εργαλείου που χρησιμοποιήσαμε για τη δημιουργία γραφημάτων. Πρόκειται για την ανοιχτού κώδικα βιβλιοθήκη της Java, το JFreeChart [3]. Στο κεφάλαιο αυτό γίνεται εκτενή αναφορά σε όλα τα στοιχεία ενός γραφήματος που υποστηρίζει το σύστημα μας και επιτρέπει στον χρήστη να διαμορφώσει.

Στο πέμπτο κεφάλαιο παρουσιάζεται η γραφική διεπαφή της εφαρμογής δημιουργίας πρότυπων γραφημάτων. Γίνεται εκτενή αναφορά στον τρόπο

Κεφάλαιο 1 – Εισαγωγή

αλληλεπίδρασης του χρήστη με το σύστημα καθώς και παρουσιάζονται όλες οι δυνατότητες πλοήγησης του μέσα στην εφαρμογή.

Ακολουθεί το έκτο κεφάλαιο, στο οποίο γίνεται αναφορά για την υλοποίηση του συστήματος. Στο κεφάλαιο αυτό παρουσιάζονται όλα τα μοντέλα και τα σχήματα που υλοποιήσαμε για τις ανάγκες της παρούσας εργασίας καθώς και όλες οι τεχνολογίες που χρησιμοποιήσαμε για την ανάπτυξη του συστήματος μας.

Στο έβδομο κεφάλαιο παρουσιάζεται η αρχιτεκτονική της εφαρμογής. Δηλαδή, ο τρόπος με τον οποίο σχεδιάσαμε και είναι οργανωμένη η εφαρμογή μας καθώς και ο τρόπος με τον οποίο αλληλεπιδρούν τα διάφορα τμήματα μεταξύ τους.

Στο όγδοο κεφάλαιο γίνεται μια περιγραφή του οδηγού και των κανόνων σωστού σχεδιασμού γραφημάτων. Στο κεφάλαιο αυτό παρουσιάζονται οι κανόνες που πρέπει να ακολουθεί κανείς για τον σχεδιασμό σωστών γραφημάτων όσον αφορά την παρουσίαση του. Ο οδηγός αυτός έγινε βασισμένοι στο βιβλίο του Edward R. Tufte «The Visual Display of Quantitative Information» [21]. Τέλος γίνεται μια αναφορά στην ψηφιακή βιβλιοθήκη που σχεδιάσαμε που διαθέτει μια πληθώρα πρότυπων γραφημάτων, τα οποία έχουν παραχθεί με χρήση των κανόνων σωστού σχεδιασμού που αναφέραμε παραπάνω.

Τέλος, ολοκληρώνουμε με το ένατο κεφάλαιο στο οποίο γίνεται μια ανακεφαλαίωση της εργασίας και προτείνονται μελλοντικές επεκτάσεις.

Κεφάλαιο 2

Ανάλυση απαιτήσεων συμπεριφοράς της εφαρμογής με τη χρήση περιπτώσεων χρήσης

2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα περιγράψουμε τη λειτουργικότητα που υποστηρίζει η εφαρμογή, μέσω των περιπτώσεων χρήσης (Use Cases)[2]. Αρχικά, θα παρουσιάσουμε την τεχνολογία των Use Cases που χρησιμοποιήσαμε για την ανάλυση των απαιτήσεων συμπεριφοράς. Στη συνέχεια θα παρουσιάσουμε τις περιπτώσεις χρήσης για την εφαρμογή δημιουργίας προτύπων γραφημάτων (Chart Template Builder) καθώς και τις περιπτώσεις χρήσης της υπηρεσίας ιστού (Chart Builder Service).

2.2 Τεχνολογία των Περιπτώσεων Χρήσης

Οι περιπτώσεις χρήσης είναι μια τεχνική για την αποτύπωση των απαιτήσεων συμπεριφοράς ενός συστήματος. Περιγράφουν τις τυπικές αλληλεπιδράσεις μεταξύ συστήματος και των εμπλεκομένων (σύστημα, υποσύστημα, τρίτο υποσύστημα κ.ο.κ), παρέχοντας έτσι μια εξιστόρηση του τρόπου χρήσης του συστήματος.

Οι περιπτώσεις χρήσης, που περιγράφονται αναλυτικά παρακάτω, παρουσιάζονται με τη μορφή πίνακα ακολουθώντας τη μεθοδολογία καταγραφής που προτείνεται από τον Alistair Cockburn στο βιβλίο του «Writing Effective Use Cases» [2].

Οι περιπτώσεις χρήσης παρέχουν ένα τρόπο τυποποιημένης περιγραφής της απαιτούμενης λειτουργικότητας ενός συστήματος, καθιστώντας την παράλληλα κατανοητή προς τον άνθρωπο. Πιο συγκεκριμένα, μια περίπτωση χρήσης περιγράφει τη συμπεριφορά ενός συστήματος, κάτω από τις διάφορες συνθήκες οι οποίες μπορούν να εμφανιστούν, καθώς αυτό ανταποκρίνεται στην αίτηση ενός από τους χρήστες του.

Κεφάλαιο 2 – Ανάλυση απαιτήσεων

Η περιγραφή των περιπτώσεων και των σεναρίων χρήσης, ως βασική συνιστώσα της ανάλυσης απαιτήσεων, είναι το στάδιο το οποίο προηγείται της σημασιολογικής μοντελοποίησης. Ο βασικός στόχος, ο οποίος επιτυγχάνεται μέσω της περιγραφής των περιπτώσεων χρήσης, είναι η ανακάλυψη και ο καθορισμός των **απαιτήσεων συμπεριφοράς**. Βέβαια, μέσα από αυτή τη διαδικασία αναδύονται και οι βασικές έννοιες, οι οποίες θα αποτελέσουν το κεντρικό αντικείμενο της σημασιολογικής μοντελοποίησης.

Η περιγραφή των περιπτώσεων χρήσης γίνεται με τη μορφή κειμένου, επιτρέποντας έτσι την εύκολη κατανόησή τους ακόμα και από ανθρώπους οι οποίοι δεν έχουν κάποια ειδική εκπαίδευση. Οι τρεις βασικές έννοιες οι οποίες χρησιμοποιούνται για την περιγραφή μιας περίπτωσης χρήσης είναι:

1. **Το πεδίο (Scope)** : Τι πραγματικά είναι το υπό μελέτη σύστημα;
2. **Ο Πρωτεύων ενεργών (Primary Actor):** Ποιον ικανοποιεί η εκπλήρωση ενός στόχου;
3. **Το επίπεδο(level):** Πόσο υψηλού ή χαμηλού επιπέδου είναι ο στόχος αυτός;

Η φόρμα η οποία προτείνεται από τον Cockburn [2] για την περιγραφή των περιπτώσεων χρήσης φαίνεται στον παρακάτω πίνακα:

USE CASE #	«Το όνομα της περίπτωσης χρήσης»	
Goal in Context	«Περιγραφή του στόχου που καλείται να υπηρετήσει»	
Scope & Level	« Ποιο μέρος του συστήματος θεωρείται μαύρο κουτί κατά το σχεδιασμό. Ένα από: Summary, Primary Task, Sub function »	
Preconditions	«Οι προϋποθέσεις που χρειάζονται για την επίτευξη του στόχου»	
Success Condition	End	« Η κατάσταση του κόσμου έπειτα από μια επιτυχή εκτέλεση »
Failed Condition	End	« Η κατάσταση του κόσμου έπειτα από μια αποτυχημένη εκτέλεση »
Primary, Secondary Actors	« Το όνομα του πρωτεύοντος και άλλων πιθανών δευτερευόντων ενεργούντων »	
Trigger	« Η δράση πάνω στο σύστημα η οποία προκαλεί την εκκίνηση της περίπτωσης χρήσης »	
DESCRIPTION	Step	Action
	1	« Τα βήματα που ακολουθούνται από την εκκίνηση της περίπτωσης χρήσης έως την επιτυχή ολοκλήρωσή της »
	2	
	3	
	4	

EXTENSIONS	Step	Branching Action
	a	1 «Περιγραφή της συνθήκης που προκαλεί διακλάδωση : δράση ή όνομα της υποπερίπτωσης χρήσης »
SUB-VARIATIONS		Branching Action
	a	1 «Εναλλακτικά βήματα που μπορούν να ακολουθηθούν από τα αρχικά και να οδηγήσουν σε επιτυχή ολοκλήρωση»

Για να μπορέσει κανείς να κατανοήσει τον τρόπο περιγραφής των περιπτώσεων χρήσης ,σύμφωνα με τον Cockburn [2], είναι απαραίτητη η κατανόηση των ακόλουθων ορισμών:

1. **Ενεργών (Actor):** Οποιοσδήποτε ή οτιδήποτε έχει συμπεριφορά
2. **Συμμετέχων (Stakeholder):** Κάποιος ή κάτι το οποίο συμμετέχει στον καθορισμό της συμπεριφοράς του υπό μελέτη συστήματος
3. **Πρωτεύων Ενεργών (Primary Actor):** Ο συμμετέχων ο οποίος ξεκινάει την αλληλεπίδραση με το υπό μελέτη σύστημα, με σκοπό την εκπλήρωση κάποιου στόχου
4. **Δευτερεύων Ενεργών (Secondary Actor):** Ο συμμετέχων ο οποίος παίρνει μέρος στην περίπτωση χρήσης αλλά δεν έχει εκκινήσει ο ίδιος τη διαδικασία της αλληλεπίδρασης
5. **Περίπτωση χρήσης (Use Case):** Ένα συμβόλαιο σχετικά με τη συμπεριφορά του υπό μελέτη συστήματος
6. **Πεδίο (Scope):** Προσδιορίζει το σύστημα το οποίο μελετάται
7. **Προϋποθέσεις και Εγγυήσεις (Preconditions and Guarantees):** Τι πρέπει να είναι αληθές πριν και μετά την εκτέλεση μιας περίπτωσης χρήσης
8. **Πυροδοτητής (Trigger):** Μια κατάσταση ή ένα γεγονός το οποίο προηγείται και προκαλεί την εκκίνηση μιας περίπτωσης χρήσης
9. **Κύριο επιτυχές σενάριο (Main Success Scenario):** Η περίπτωση στην οποία τίποτα δεν πάει λάθος
10. **Επεκτάσεις (Extensions):** Οτιδήποτε διαφορετικό μπορεί να συμβεί κατά τη διάρκεια εκτέλεσης του επιτυχούς σεναρίου.

2.3 Περιγραφή Περιπτώσεων Χρήσης

Στην ενότητα αυτή θα περιγράψουμε τις περιπτώσεις χρήσης καθώς και τους τύπους χρηστών ολόκληρης της εφαρμογής. Η εφαρμογή δημιουργίας πρότυπων γραφημάτων που σχεδιάσαμε, επιτρέπει μέσω μιας κατάλληλης διεπαφής τη δημιουργία και τη διαχείριση γραφημάτων. Η εφαρμογή υποστηρίζει χρήστες, οι οποίοι έχουν το δικό τους λογαριασμό και μπορούν να χρησιμοποιήσουν όλη τη

Κεφάλαιο 2 – Ανάλυση απαιτήσεων

προσφερόμενη λειτουργικότητα. Άρα, απαραίτητη προϋπόθεση είναι η δημιουργία λογαριασμού. Για να ενεργοποιηθεί ένας λογαριασμός θα πρέπει να εγκριθεί από το διαχειριστή της εφαρμογής.

Κάθε χρήστης που έχει δημιουργήσει λογαριασμό, μπορεί να δημιουργήσει ένα πρότυπο γράφημα της επιλογής του, έτσι όπως επιθυμεί.

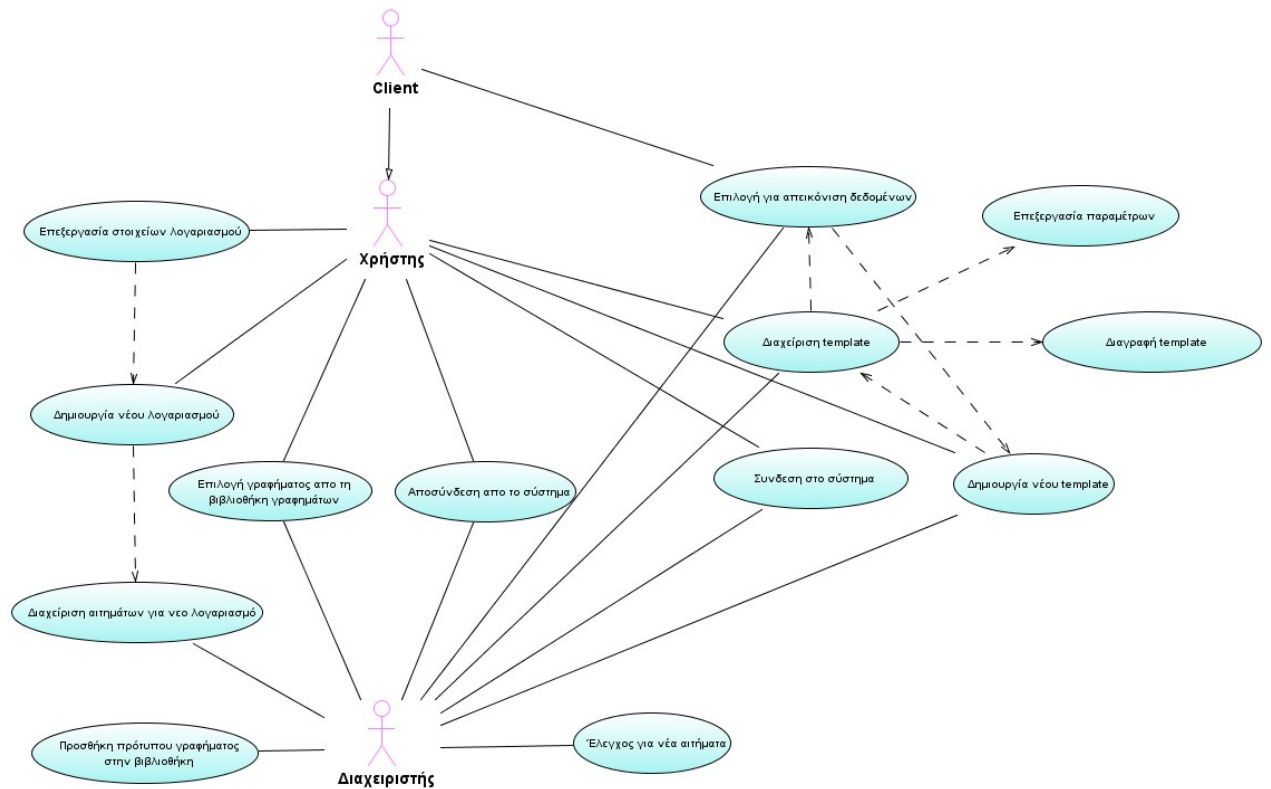
Κάθε χρήστης μπορεί να δημιουργήσει τα δικά του πρότυπα γραφήματα και να επεξεργάζεται τις παραμέτρους του όπως και όποτε επιθυμεί.

Τέλος, κάθε χρήστης μπορεί να επεξεργαστεί το λογαριασμό του και να αλλάξει τα προσωπικά του στοιχεία όπως επιθυμεί.

Για λόγους ασφάλειας, υπάρχει ένας διαχειριστής στην εφαρμογή που είναι υπεύθυνος για την έγκριση ή απόρριψη αιτημάτων για τη δημιουργία λογαριασμού.

Η υπηρεσία ιστού που σχεδιάσαμε (Web Service), εξυπηρετεί εφαρμογές-πελάτες, που επιθυμούν να απεικονίσουν τα δικά τους δεδομένα πάνω σε ένα πρότυπο γράφημα. Εφαρμογή-πελάτης, μπορεί να είναι είτε ο ίδιος ο δημιουργός του πρότυπου γραφήματος είτε κάποιος άλλος χρήστης που θέλει να απεικονίσει τα δεδομένα του σε ένα πρότυπο γράφημα ενός δημιουργού.

Στην παρακάτω εικόνα, ακολουθεί μια γενική περιγραφή του συστήματος που περιγράφει τους τύπους χρηστών που υποστηρίζει η εφαρμογή καθώς και τις επιλογές που έχει καθένας από αυτούς:



Εικόνα 2 - Διάγραμμα απαιτήσεων με τη χρήση των Use Cases

Όπως παρατηρούμε από την παραπάνω εικόνα, η εφαρμογή μας υποστηρίζει τους εξής τύπους χρηστών:

- **Χρήστης:** είναι αυτός που διαθέτει λογαριασμό στο σύστημα και δημιουργεί πρότυπα γραφήματα
- **Διαχειριστής:** διαθέτει ένα μοναδικό λογαριασμό διαχειριστή και έχει να κάνει με την απόρριψη ή την έγκριση ενός αιτήματος για τη δημιουργία λογαριασμού. Ο Διαχειριστής έχει όλες τις επιλογές που έχει και ένας Χρήστης εκτός από την δημιουργία και επεξεργασία του λογαριασμού του, δηλαδή μπορεί και αυτός να δημιουργήσει δικά του πρότυπα γραφήματα και να τα επεξεργαστεί. Επίσης έχει το επιπλέον δικαίωμα να προσθέτει πρότυπα γραφήματα στην ψηφιακή βιβλιοθήκη γραφημάτων.
- **Client:** είναι οποιαδήποτε εφαρμογή-πελάτης που καλεί το web service για την απεικόνιση δεδομένων σε ένα template γράφημα. Αυτός ο τύπος χρήστη

μπορεί να είναι ο ίδιος ο δημιουργός του template γραφήματος (Χρήστης) ή κάποιος άλλος που απλά χρησιμοποιεί το template γράφημα ενός δημιουργού

- **Παρατηρητής του τελικού γραφήματος:** είναι ο θεατής του τελικού γραφήματος. Κάθε Client που δημιουργεί ένα γράφημα μπορεί να το προσθέσει στην εφαρμογή που επιθυμεί (ιστοσελίδα ή οτιδήποτε άλλο). Εκείνος που βλέπει αυτό το τελικό γράφημα είναι παρατηρητής. Ο Παρατηρητής δεν έχει καμία ουσιαστική δυνατότητα στην εφαρμογή.

Ο Χρήστης είναι ο κύριος τύπος χρήστη της εφαρμογής και έχει την μεγαλύτερη λειτουργικότητα στην εφαρμογή. Οι επιλογές που έχει είναι:

Δημιουργία νέου λογαριασμού: συμπληρώνει τα απαραίτητα πεδία που χρειάζονται για τη δημιουργία ενός λογαριασμού

Επεξεργασία των στοιχείων του λογαριασμού του: ο Χρήστης μπορεί να αλλάξει κάποια στοιχεία στον προσωπικό του λογαριασμό όπως ο κωδικός του και το email του

Δημιουργία νέου πρότυπου γραφήματος: ο Χρήστης επιλέγει να δημιουργήσει ένα νέο γράφημα από όλους τους τύπους γραφήματος που υποστηρίζει η εφαρμογή.

Διαχείριση ενός πρότυπου γραφήματος: ο Χρήστης μπορεί να επεξεργάζεται περαιτέρω τις παραμέτρους ενός γραφήματος που έχει ήδη δημιουργήσει

Διαγραφή ενός πρότυπου γραφήματος: ο Χρήστης μπορεί να διαγράψει ένα template που έχει ήδη δημιουργήσει

Επιλογή πρότυπου γραφήματος για απεικόνιση δεδομένων: ο Χρήστης επιλέγει το template που τον ενδιαφέρει και συγκεντρώνει τις πληροφορίες που χρειάζονται για την απεικόνιση πραγματικών δεδομένων πάνω σε αυτό το template

Καθορισμός των παραμέτρων ενός πρότυπου γραφήματος: ο Χρήστης επιλέγει από μια λίστα με όλες τις παραμέτρους ενός γραφήματος εκείνες που θέλει να τροποποιήσει

Επιλογή γραφήματος από τη βιβλιοθήκη γραφημάτων: η εφαρμογή υποστηρίζει μια βιβλιοθήκη έτοιμων γραφημάτων για διάφορες εφαρμογές. Κάθε ένα

από τα γραφήματα είναι διαθέσιμα για αντιγραφή στα δικά του πρότυπα γραφήματα ώστε να μπορεί να τα χρησιμοποιήσει όπως επιθυμεί

Ο Διαχειριστής διαθέτει έναν μοναδικό λογαριασμό και για λόγους ασφαλείας δεν επιτρέπεται η δημιουργία λογαριασμού του μέσα από γραφικό περιβάλλον. Ο Διαχειριστής έχει τις ίδιες επιλογές με τον Χρήστη όσον αφορά την δημιουργία και επεξεργασία πρότυπων γραφημάτων που παρουσιάστηκαν παραπάνω. Οι επιπλέον επιλογές του Διαχειριστή είναι:

Διαχείριση των αιτημάτων χρηστών: ο Διαχειριστής βλέπει όλα τα νέα αιτήματα για τη δημιουργία ενός νέου λογαριασμού και αποφασίζει αν θα αποδεχτεί ή θα απορρίψει κάποιο από αυτά τα αιτήματα.

Προσθήκη πρότυπου γραφήματος στην ψηφιακή βιβλιοθήκη: Ο Διαχειριστής έχει την αρμοδιότητα να προσθέτει ένα από τα πρότυπα γραφήματα που έχει δημιουργήσει στην ψηφιακή βιβλιοθήκη, ώστε να είναι διαθέσιμο προς επιλογή για τους χρήστες της εφαρμογής.

Ο Client της εφαρμογής μπορεί να είναι οποιοσδήποτε χρήστης καλεί το web service για την απεικόνιση δεδομένων σε ένα πρότυπο γράφημα. Πιο συγκεκριμένα client μπορεί να είναι είτε ο ίδιος ο δημιουργός ενός πρότυπου γραφήματος (Χρήστης) είτε κάποιος άλλος που απλά χρησιμοποιεί τα στοιχεία ενός γραφήματος κάποιου δημιουργού (Χρήστη). Οι επιλογές ενός client είναι οι ακόλουθες:

Επιλογή πρότυπου γραφήματος για απεικόνιση δεδομένων: ο client επιλέγει το πρότυπο γράφημα που τον ενδιαφέρει και συγκεντρώνει τις πληροφορίες που χρειάζονται για την απεικόνιση πραγματικών δεδομένων πάνω σε αυτό το πρότυπο γράφημα.

Ο Παρατηρητής ενός γραφήματος μπορεί να είναι οποιοσδήποτε μπορεί να δει το τελικό γράφημα μέσα από κάποια εφαρμογή, εφόσον αυτό έχει αναπτυχθεί.

2.3.1 Περιγραφή Περιπτώσεων Χρήσης Εφαρμογής Δημιουργίας Πρότυπων Γραφημάτων

Θα χωρίσουμε την παρουσίαση των περιπτώσεων χρήσης σε δύο τμήματα. Πρώτα θα παρουσιάσουμε τις περιπτώσεις χρήσης για έναν Χρήστη και στη συνέχεια τις περιπτώσεις χρήσης για τον Διαχειριστή.

Κεφάλαιο 2 – Ανάλυση απαιτήσεων

Να σημειώσουμε εδώ ότι και ο Χρήστης και ο Διαχειριστής, πρέπει να κάνουν είσοδο στην εφαρμογή με το λογαριασμό τους για να μπορέσουν να χρησιμοποιήσουν όλη τη δυνατή λειτουργικότητα. Έτσι, αυτές οι περιπτώσεις χρήσης είναι κοινές και για τους δύο.

2.3.1.1 Περιπτώσεις Χρήσης ενός Χρήστη της εφαρμογής

Πίνακας 1 - Δημιουργία λογαριασμού

USE CASE 1	Δημιουργία λογαριασμού	
Goal in Scope	Αίτημα για τη δημιουργία νέου λογαριασμού μέσω του οποίου θα προσφέρεται η λειτουργικότητα της εφαρμογής	
Scope & Level	Σύστημα, Primary Task	
Preconditions	Καμία	
Success End Condition	Αποδοχή του αιτήματος για τη δημιουργία λογαριασμού	
Failed End Condition	Απόρριψη του αιτήματος για τη δημιουργία λογαριασμού	
Primary, Secondary Actors	υποψήφιος χρήστης Διαχειριστής	
Trigger	Ο υποψήφιος χρήστης θέλει να γίνει Χρήστης της εφαρμογής	
DESCRIPTION	Step	Action
	1	Ο υποψήφιος χρήστης επιλέγει από την κεντρική σελίδα της εφαρμογής την δημιουργία νέου λογαριασμού
	2	Ο υποψήφιος χρήστης συμπληρώνει τα απαραίτητα στοιχεία σε μια φόρμα για τη δημιουργία λογαριασμού
	3	<u>Έλεγχος στοιχείων φόρμας (2)</u>
	4	<u>Διαχείριση αιτήματος για τη δημιουργία λογαριασμού (9)</u>
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action

Πίνακας 2 - Έλεγχος Στοιχείων Φόρμας

USE CASE 2	Έλεγχος στοιχείων φόρμας	
Goal in Context	Έλεγχος των στοιχείων μιας φόρμας κάθε φορά που ο υποψήφιος χρήστης /Χρήστης χρειάζεται να συμπληρώσει κάποιες παραμέτρους	
Scope & Level	Σύστημα, Subfunction	
Preconditions	Υποβολή μιας φόρμας	
Success End	Υποβολή των στοιχείων της φόρμας	

Κεφάλαιο 2 – Ανάλυση απαιτήσεων

Condition		
Failed Condition	End	Ακύρωση των στοιχείων μιας φόρμας
Primary, Secondary Actors	Αιτών Σύστημα	
Trigger	Υποβολή μιας φόρμας	
DESCRIPTION	Step	Action
	1	Ο υποψήφιος χρήστης / Χρήστης επιλέγει να υποβληθεί η φόρμα με τις παραμέτρους που συμπλήρωσε
	2	Το Σύστημα ελέγχει την ορθότητα των παραμέτρων
	3	Το Σύστημα αποθηκεύει τις παραμέτρους που έθεσε ο υποψήφιος χρήστης /Χρήστης
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action
	3	Το Σύστημα ενημερώνει για τη μη ορθότητα των παραμέτρων της φόρμας: Το σενάριο συνεχίζει στο Βήμα 1

Πίνακας 3 - Σύνδεση στο σύστημα

USE CASE 3	Σύνδεση στο σύστημα (Login)	
Goal in Context	Ο Χρήστης εισάγει το username και το password και συνδέεται στο σύστημα.	
Scope & Level	Σύστημα, Subfunction	
Preconditions	<u>Δημιουργία λογαριασμού(1)</u>	
Success Condition	End	Επιτυχής σύνδεση: ο Χρήστης μπορεί να διαχειριστεί τα πρότυπα γραφήματα
Failed Condition	End	Αποτυχία σύνδεσης: ο Χρήστης δε μπορεί να διαχειριστεί τα πρότυπα γραφήματα
Primary, Secondary Actors	Χρήστης Σύστημα	
Trigger	Ο Χρήστης θέλει να διαχειριστεί τα πρότυπα γραφήματα	
DESCRIPTION	Step	Action
	1	Ο Χρήστης εισάγει το username και το password και επιλέγει login
	2	Το Σύστημα επικοινωνεί με τη βάση δεδομένων και ελέγχει τα στοιχεία του Χρήστη
	3	Ο Χρήστης συνδέεται επιτυχώς και μπορεί να διαχειριστεί τα πρότυπα γραφήματα
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action
	3	Το Σύστημα ενημερώνει για τη μη ορθότητα των στοιχείων που εισήγαγε ο Χρήστης:

		Το σενάριο συνεχίζει στο Βήμα 1
--	--	---------------------------------

Πίνακας 4 - Αποσύνδεση από το σύστημα

USE CASE 4	Αποσύνδεση από το σύστημα (Logout)	
Goal in Context	Ο Χρήστης επιλέγει να αποσυνδεθεί από το σύστημα	
Scope & Level	Σύστημα, Subfunction	
Preconditions	<u>Σύνδεση Χρήστη στο σύστημα(3)</u>	
Success Condition	End	Ο Χρήστης αποσυνδέεται επιτυχώς από το σύστημα
Failed Condition	End	Ο Χρήστης δεν αποσυνδέεται από το σύστημα
Primary, Secondary Actors	Χρήστης Σύστημα	
Trigger	Ο Χρήστης θέλει να αποσυνδεθεί από το σύστημα	
DESCRIPTION	Step	Action
	1	Ο Χρήστης επιλέγει από το βασικό μενού : Αποσύνδεση
	2	Το Σύστημα ελέγχει αν ο Χρήστης έχει αποθηκεύσει τυχόν αλλαγές που έχει κάνει
	3	Το Σύστημα αποσυνδέει τον Χρήστη από την εφαρμογή
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action
	2	Το Σύστημα ενημερώνει το Χρήστη ότι δεν έχουν αποθηκευτεί οι αλλαγές που έχει κάνει:
	3	Ο Χρήστης επιλέγει από το βασικό μενού: Αποθήκευση των αλλαγών και το σενάριο συνεχίζει από το Βήμα 1

Πίνακας 5 - Επεξεργασία στοιχείων λογαριασμού

USE CASE 5	Επεξεργασία στοιχείων λογαριασμού	
Goal in Context	Ο Χρήστης μπορεί να διαχειριστεί το λογαριασμό που έχει δημιουργήσει αλλάζοντας το password ή/και το email του	
Scope & Level	Σύστημα, Primary Task	
Preconditions	<u>Σύνδεση Χρήστη στο σύστημα(3)</u>	
Success Condition	End	Ο Χρήστης αλλάζει επιτυχώς το password ή/και το email του
Failed Condition	End	Ο Χρήστης δεν αλλάζει το password ή/και το email του
Primary, Secondary Actors	Χρήστης Σύστημα	
Trigger	Ο Χρήστης θέλει να διαχειριστεί το λογαριασμό του	

DESCRIPTION	Step	Action
	1	Ο Χρήστης επιλέγει από το βασικό μενού να δει το λογαριασμό του
	2	Ο Χρήστης εισάγει το καινούριο password ή/και email και επιλέγει αποθήκευση των νέων στοιχείων
	3	Το Σύστημα επικοινωνεί με τη βάση δεδομένων για να ενημερώσει το λογαριασμό του Χρήστη και αποθηκεύει τις αλλαγές
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action

Πίνακας 6 - Δημιουργία νέου πρότυπου γραφήματος

USE CASE 6	Δημιουργία νέου πρότυπου γραφήματος	
Goal in Context	Ο Χρήστης δημιουργεί ένα νέο πρότυπο γράφημα, επιλέγοντας τον τύπο γραφήματος που τον ενδιαφέρει και καθορίζει τις παραμέτρους του γραφήματος έτσι όπως επιθυμεί	
Scope & Level	Σύστημα, Primary Task	
Preconditions	<u>Σύνδεση Χρήστη στο σύστημα(3)</u>	
Success End Condition	Ο Χρήστης αποθηκεύει το πρότυπο γράφημα που δημιούργησε	
Failed End Condition	Ο Χρήστης δεν αποθηκεύει το πρότυπο γράφημα που δημιούργησε	
Primary, Secondary Actors	Χρήστης Σύστημα	
Trigger	Ο Χρήστης θέλει να δημιουργήσει ένα καινούριο πρότυπο γράφημα	
DESCRIPTION	Step	Action
	1	Ο Χρήστης επιλέγει από το βασικό μενού τη δημιουργία νέου πρότυπου γραφήματος
	2	Ο Χρήστης επιλέγει τον τύπο του γραφήματος που τον ενδιαφέρει να δημιουργήσει
	3	Ο Χρήστης συμπληρώνει τις απαραίτητες παραμέτρους που χρειάζονται για τη δημιουργία του πρότυπου γραφήματος, όπως ο τίτλος του, τα labels κ.ά.
	4	<u>Έλεγχος στοιχείων φόρμας (2)</u>
	5	Προβολή του πρότυπου γραφήματος που δημιούργησε ο Χρήστης
	6	Ο Χρήστης αποθηκεύει το πρότυπο γράφημα που δημιούργησε
EXTENSIONS	Step	Branching Action
SUB-		Branching Action

VARIATIONS		
	6	<u>Επεξεργασία των παραμέτρων του πρότυπου γραφήματος(8)</u>
	7	Ο Χρήστης αποθηκεύει το πρότυπο γράφημα που δημιούργησε

Πίνακας 7 - Διαχείριση των πρότυπων γραφημάτων

USE CASE 7	Διαχείριση των πρότυπων γραφημάτων του Χρήστη	
Goal in Context	Ο Χρήστης διαχειρίζεται τα πρότυπα γραφήματα που έχει ήδη δημιουργήσει. Μπορεί να επεξεργαστεί και να αλλάξει τις παραμέτρους ενός πρότυπου γραφήματος, να το διαγράψει, να το επιλέξει για απεικόνιση πραγματικών δεδομένων	
Scope & Level	Σύστημα, Primary Task	
Preconditions	<u>Σύνδεση Χρήστη στο σύστημα(3)</u>	
Success End Condition	Ο Χρήστης διαχειρίζεται επιτυχώς το πρότυπο γράφημα που τον ενδιαφέρει	
Failed End Condition	Ο Χρήστης δε διαχειρίζεται το πρότυπο γράφημα που τον ενδιαφέρει	
Primary, Secondary Actors	Χρήστης Σύστημα	
Trigger	Ο Χρήστης θέλει να διαχειριστεί ένα πρότυπο γράφημα που έχει δημιουργήσει	
DESCRIPTION	Step	Action
	1	Ο Χρήστης βλέπει μια λίστα με όλα τα πρότυπα γραφήματα που έχει δημιουργήσει
	2	Ο Χρήστης επιλέγει το πρότυπο γράφημα που τον ενδιαφέρει από τη λίστα
	3	Ο Χρήστης επιλέγει να επεξεργαστεί τις παραμέτρους του πρότυπου γραφήματος
	4	<u>Επεξεργασία των παραμέτρων του πρότυπου γραφήματος(8)</u>
	5	Ο Χρήστης αποθηκεύει το πρότυπο γράφημα που δημιούργησε
EXTENSIONS	Step	Branching Action
	3a	Ο Χρήστης επιλέγει να διαγράψει ένα συγκεκριμένο πρότυπο γράφημα
	4a	Το Σύστημα ρωτά τον Χρήστη αν είναι σίγουρος για τη διαγραφή
	5a	Ο Χρήστης επιβεβαιώνει τη διαγραφή
	6a	Το Σύστημα επικοινωνεί με τη βάση δεδομένων για τη διαγραφή του συγκεκριμένου πρότυπου <u>γραφήματος</u>
	7a	Το Σύστημα εμφανίζει στο Χρήστη την ενημερωμένη λίστα με τα υπόλοιπα πρότυπα γραφήματα
	3b	Ο Χρήστης επιλέγει το συγκεκριμένο πρότυπο γράφημα για απεικόνιση δεδομένων
	4b	Το Σύστημα εμφανίζει στο Χρήστη τις απαραίτητες

		πληροφορίες για το συγκεκριμένο πρότυπο γράφημα
SUB-VARIATIONS		Branching Action

Πίνακας 8 - Επεξεργασία των παραμέτρων του πρότυπου γραφήματος

USE CASE 8	Επεξεργασία των παραμέτρων του πρότυπου γραφήματος	
Goal in Context	Ο Χρήστης επεξεργάζεται διάφορες παραμέτρους ενός πρότυπου γραφήματος. Οι παράμετροι αυτοί είναι προαιρετικές και αφορούν την εμφάνιση και τη μορφοποίηση του γραφήματος. Αναφέρονται σε διάφορα στοιχεία ενός γραφήματος, όπως είναι ο τίτλος, υπότιτλος, legend, labels, plot, annotations, axes κ.ά.	
Scope & Level	Σύστημα, Primary Task	
Preconditions	<u>Δημιουργία νέου πρότυπου γραφήματος (6)</u>	
Success End Condition	Ο Χρήστης επεξεργάζεται επιτυχώς το πρότυπο γράφημα που τον ενδιαφέρει και το αποθηκεύει	
Failed End Condition	Ο Χρήστης δεν επεξεργάζεται το πρότυπο γράφημα που τον ενδιαφέρει και το αποθηκεύει	
Primary, Secondary Actors	Χρήστης Σύστημα	
Trigger	Ο Χρήστης θέλει να επεξεργαστεί ένα πρότυπο γράφημα που έχει δημιουργήσει	
DESCRIPTION	Step	Action
	1	Ο Χρήστης επιλέγει από το βασικό μενού το στοιχείο του πρότυπου γραφήματος που θέλει να επεξεργαστεί
	2	Το Σύστημα εμφανίζει όλες τις δυνατές επιλογές για το συγκεκριμένο στοιχείο στο Χρήστη
	3	Ο Χρήστης εφαρμόζει τις αλλαγές που επιθυμεί και τις υποβάλλει
	4	Το Σύστημα εμφανίζει το νέο πρότυπο γράφημα
	5	Ο Χρήστης αποθηκεύει το νέο πρότυπο γράφημα
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action

Πίνακας 9 - Επιλογή γραφήματος από τη βιβλιοθήκη γραφημάτων

USE CASE 9	Επιλογή έτοιμου γραφήματος από τη βιβλιοθήκη γραφημάτων	
Goal in Context	Ο Χρήστης επιλέγει ένα έτοιμο γράφημα από τη βιβλιοθήκη γραφημάτων να το αντιγράψει στα δικά του γραφήματα και να το επεξεργαστεί περαιτέρω	
Scope & Level	Σύστημα, Primary Task	
Preconditions	<u>Σύνδεση Χρήστη στο σύστημα(3)</u>	

Success Condition	End	Ο Χρήστης αντιγράφει επιτυχώς το γράφημα στα δικά του γραφήματα
Failed Condition	End	Ο Χρήστης δεν αντιγράφει επιτυχώς το γράφημα στα δικά του γραφήματα
Primary, Secondary Actors		Χρήστης Σύστημα
Trigger		Ο Χρήστης θέλει να χρησιμοποιήσει ένα έτοιμο γράφημα για την εφαρμογή του
DESCRIPTION	Step	Action
	1	Ο Χρήστης επιλέγει από το βασικό να δει τα γραφήματα από τη βιβλιοθήκη γραφημάτων
	2	Το Σύστημα εμφανίζει όλα τα γραφήματα από τη βιβλιοθήκη γραφημάτων στο Χρήστη
	3	Ο Χρήστης επιλέγει το γράφημα που τον ενδιαφέρει και το αντιγράφει στα δικά του γραφήματα
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action

2.3.1.2 Περιπτώσεις χρήσης Διαχειριστή

Use Case 6	Δημιουργία νέου πρότυπου γραφήματος
Use Case 7	Διαχείριση πρότυπων γραφημάτων
Use Case 8	Επεξεργασία παραμέτρων πρότυπου γραφήματος
Use Case 9	Επιλογή γραφήματος από την βιβλιοθήκη γραφημάτων

Πίνακας 10 - Διαχείριση αιτημάτων για νέο λογαριασμό

USE CASE 10	Διαχείριση των αιτημάτων για τη δημιουργία νέου λογαριασμού
Goal in Context	Ο Διαχειριστής κάθε φορά που συνδέεται στην εφαρμογή, βλέπει μια λίστα με τα προσωπικά στοιχεία όλων που επιθυμούν να δημιουργήσουν ένα νέο λογαριασμό και είτε αποδέχεται είτε απορρίπτει το αίτημα τους.
Scope & Level	Σύστημα, Primary Task
Preconditions	Ο Διαχειριστής να έχει κάνει είσοδο
Success Condition	End Ο Διαχειριστής διαχειρίζεται επιτυχώς ένα αίτημα για νέο λογαριασμό
Failed Condition	End Ο Διαχειριστής δε διαχειρίζεται ένα αίτημα για νέο λογαριασμό
Primary, Secondary Actors	Διαχειριστής Σύστημα
Trigger	Απόρριψη/Αποδοχή νέου Χρήστη

DESCRIPTION	Step	Action
	1	Το Σύστημα εμφανίζει στον Διαχειριστή μια λίστα με τα νέα αιτήματα
	2	Ο Διαχειριστής επιλέγει από το μενού αποδοχή του αιτήματος
	3	Το Σύστημα επικοινωνεί με τη βάση δεδομένων για την εισαγωγή νέου Χρήστη
	4	Το Σύστημα εμφανίζει στον Διαχειριστή την ανανεωμένη λίστα με τα αιτήματα
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action
	2	Ο Διαχειριστής επιλέγει από το μενού απόρριψη του αιτήματος
	3	Το Σύστημα ενημερώνει τη βάση δεδομένων
	4	Το Σύστημα εμφανίζει στον Διαχειριστή την ανανεωμένη λίστα με τα αιτήματα

Πίνακας 11 - Έλεγχος για νέα αιτήματα

USE CASE 11	Έλεγχος για νέα αιτήματα	
Goal in Context	Ο Διαχειριστής ενημερώνεται για τον αν υπάρχουν νέα αιτήματα για τη δημιουργία λογαριασμού	
Scope & Level	Σύστημα, Primary Task	
Preconditions	Ο Διαχειριστής να έχει κάνει είσοδο	
Success End Condition	Ο Διαχειριστής ενημερώνεται για τα νέα αιτήματα	
Failed End Condition	Ο Διαχειριστής δεν ενημερώνεται για τα νέα αιτήματα	
Primary, Secondary Actors	Διαχειριστής Σύστημα	
Trigger	Ο Διαχειριστής θέλει να ενημερωθεί για τα νέα αιτήματα	
DESCRIPTION	Step	Action
	1	Ο Διαχειριστής επιλέγει από το μενού ενημέρωση για νέα αιτήματα
	2	Το Σύστημα εμφανίζει την ανανεωμένη λίστα στον Διαχειριστή
EXTENSIONS	Step	Branching Action
	2a	Δεν υπάρχουν νέα αιτήματα: Το Σύστημα ενημερώνει τον Διαχειριστή και το σενάριο τερματίζει ανεπιτυχώς
SUB-VARIATIONS		Branching Action
	3	<u>Διαχείριση των αιτημάτων για τη δημιουργία νέου λογαριασμού(9)</u>

Πίνακας 12 - Προσθήκη πρότυπου γραφήματος στην ψηφιακή βιβλιοθήκη

USE CASE 12	Προσθήκη πρότυπου γραφήματος στην βιβλιοθήκη γραφημάτων	
Goal in Context	Ο Διαχειριστής προσθέτει ένα πρότυπο γράφημα που έχει δημιουργήσει στην ψηφιακή βιβλιοθήκη πρότυπων γραφημάτων ώστε να είναι διαθέσιμα προς επιλογή από τους χρήστες	
Scope & Level	Σύστημα, Primary Task	
Preconditions	<u>Δημιουργία πρότυπου γραφήματος(6)</u>	
Success End Condition	Ο Διαχειριστής προσθέτει το πρότυπο γράφημα στην βιβλιοθήκη	
Failed End Condition	Ο Διαχειριστής δεν προσθέτει το πρότυπο γράφημα στην βιβλιοθήκη	
Primary, Secondary Actors	Διαχειριστής Σύστημα	
Trigger	Ο Διαχειριστής θέλει να προσθέσει ένα νέο πρότυπο γράφημα στην βιβλιοθήκη	
DESCRIPTION	Step	Action
	1	Ο Διαχειριστής επιλέγει το πρότυπο γράφημα που τον ενδιαφέρει
	2	Ο Διαχειριστής επιλέγει από το βασικό μενού προσθήκη του γραφήματος στην βιβλιοθήκη
	3	Το Σύστημα προσθέτει το πρότυπο γράφημα στην βιβλιοθήκη γραφημάτων
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action

2.3.2 Περιπτώσεις Χρήσης Υπηρεσίας Ιστού

Πίνακας 13 - Επιλογή γραφήματος για απεικόνιση δεδομένων

USE CASE 13	Επιλογή πρότυπου γραφήματος για απεικόνιση δεδομένων	
Goal in Context	Ο client επιλέγει το πρότυπο γράφημα που τον ενδιαφέρει για να απεικονίσει τα δεδομένα του πάνω σε αυτό	
Scope & Level	Σύστημα, Primary Task	
Preconditions	<u>Δημιουργία πρότυπου γραφήματος(6)</u>	
Success End Condition	Ο client δημιουργεί το γράφημα με τα δικά του δεδομένα	
Failed End Condition	Ο client δε δημιουργεί το γράφημα με τα δικά του δεδομένα	
Primary, Secondary Actors	client Σύστημα	
Trigger	Ο client θέλει να δημιουργήσει ένα γράφημα με τα δικά του δεδομένα	
DESCRIPTION	Step	Action

Κεφάλαιο 2 – Ανάλυση απαιτήσεων

	1	Ο client επιλέγει το πρότυπο γράφημα που τον ενδιαφέρει
	2	Το Σύστημα εμφανίζει τα στοιχεία που χρειάζονται για το συγκεκριμένο πρότυπο γράφημα
	3	Ο client συλλέγει τα στοιχεία που χρειάζονται και καλεί το web service
	4	Το Σύστημα του επιστρέφει το τελικό γράφημα
EXTENSIONS	Step	Branching Action
SUB-VARIATIONS		Branching Action

2.4 Περίληψη

Στο κεφάλαιο αυτό παρουσιάσαμε τις απαιτήσεις συμπεριφοράς της εφαρμογής. Παρουσιάστηκαν αναλυτικά όλοι οι τύποι χρηστών που υποστηρίζονται από την εφαρμογή καθώς και οι επιλογές που έχει καθένας από αυτούς. Η περιγραφή των απαιτήσεων συμπεριφοράς έγινε με τη χρήση των Use Cases , όπως προτείνεται στο βιβλίο του Alistair Cockburn, «Writing Effective Use Cases» [2].

Κεφάλαιο 3

Σχετικές Εφαρμογές

3.1 Εισαγωγή

Τα γραφήματα είναι ένα πολύ χρήσιμο και σημαντικό εργαλείο σε όλες τις σημερινές εφαρμογές. Είναι ιδανικά για την απεικόνιση πληροφορίας και συναντώνται καθημερινά σε εφημερίδες , περιοδικά και επιχειρήσεις σε όλο το κόσμο. Για το λόγο αυτό έχουν αναπτυχθεί πολλές εφαρμογές μέσω διαδικτύου και όχι με σκοπό τη δημιουργία γραφημάτων. Στην ενότητα αυτή θα παρουσιάσουμε τις πιο σημαντικές από αυτές τις εφαρμογές, αναφέροντας τα μειονεκτήματα και τα πλεονεκτήματά τους.

3.2 Google Chart Creator

Το Google Chart Creator [] είναι μια εφαρμογή στο διαδίκτυο που επιτρέπει τη δημιουργία γραφημάτων. Ο εκάστοτε χρήστης επιλέγει έναν τύπο γραφήματος που τον ενδιαφέρει, εισάγει τα δεδομένα του και καθορίζει κάποιες παραμέτρους όπως ο τίτλος του γραφήματος, το background χρώμα, τις ετικέτες (labels) που θα εμφανίζονται, τις διαστάσεις του γραφήματος. Το γράφημα επιστρέφεται ως εικόνα και μαζί με αυτά ένα Link για την τοποθεσία της εικόνας και ένα html image tag για να μπορεί να ενσωματωθεί σε μια Html σελίδα.

Τα μειονεκτήματα της εφαρμογής αυτής είναι ότι υπάρχει περιορισμένος αριθμός στην επιλογή τύπου γραφημάτων και δεν υπάρχει μεγάλη δυνατότητα παραμετροποίησης από το χρήστη.[4]

3.3 Smart Draw

Το Smart Draw είναι μια Desktop εφαρμογή που επιτρέπει στον εκάστοτε χρήστη είτε να δημιουργεί ένα γράφημα από την αρχή είτε να επιλέγει, από ένα έτοιμο πρότυπο γράφημα, αυτό που τον ενδιαφέρει. Από εκεί και έπειτα ο χρήστης μπορεί να καθορίσει τις παραμέτρους του γραφήματος και να εισάγει τα δεδομένα του , επιλέγοντας απ' ευθείας την περιοχή πάνω στο γράφημα που τον ενδιαφέρει και

να εισάγει τις τιμές του. Η εφαρμογή αυτή υποστηρίζει πολλούς τύπους εξαγωγής του γραφήματος όπως εικόνα, html, pdf, svg κ.ά.

Το κύριο μειονέκτημα της εφαρμογής αυτής είναι ότι είναι ένα πληρωτέο πρόγραμμα. [5]

3.4 AdvGraphGenerator

Είναι μια desktop εφαρμογή που υποστηρίζει τους εξής τύπους γραφημάτων: Bar, Line, Pie, Stacked bar, Box-Whisker, 3D bar, 3D line, 3D pie, 3D stacked bar. Υποστηρίζει γραφήματα με μια ή περισσότερες σειρές δεδομένων και επιτρέπει την παραμετροποίηση ενός γραφήματος όσον αφορά τις ετικέτες (labels), το πλέγμα (grid), διαστάσεις, background χρώμα. Ο τύπος εξαγωγής του γραφήματος είναι ως εικόνα.

Το βασικό μειονέκτημα της εφαρμογής είναι ότι είναι πληρωτέο πρόγραμμα. [6]

3.5 CeWolf

Το cewolf είναι μια tag library που μπορεί να χρησιμοποιηθεί σε εφαρμογές διαδικτύου βασισμένες σε Servlets/JSP για να δημιουργήσει ένα γράφημα. Είναι μια δωρεάν βιβλιοθήκη που υποστηρίζει έναν μεγάλο αριθμό γραφημάτων και μεγάλη δυνατότητα παραμετροποίησης πάνω στο γράφημα.

Το μειονέκτημα της εφαρμογής είναι ότι δεν είναι μια αυτόματη εφαρμογή και πρέπει ο εκάστοτε χρήστης να γράψει κώδικα για να δημιουργήσει ένα γράφημα. [7]

3.6 J Web Chart

Το J Web Chart είναι μια εφαρμογή διαδικτύου ανοικτού κώδικα που παρουσιάζει κάποια παραδείγματα γραφημάτων. Δεν υποστηρίζει την εισαγωγή δεδομένων από τον ίδιο το χρήστη. Δεν υπάρχει η δυνατότητα παραμετροποίησης των στοιχείων του γραφήματος.[8]

3.7 MS Excel, Open Office

Είναι εργαλεία που επιτρέπουν την ανάπτυξη γραφημάτων. Το βασικό μειονέκτημα είναι ότι επιτρέπουν την παρουσίαση των γραφημάτων μόνο μέσα σε δικά τους περιβάλλοντα ή σε κάποιο άλλο αρχείο.

3.8 JFreeChart

Είναι μια δωρεάν βιβλιοθήκη της Java, που επιτρέπει την δημιουργία οποιουδήποτε τύπου γραφημάτων. Υποστηρίζει όλους σχεδόν τους τύπους γραφημάτων, από τους πιο διαδεδομένους μέχρι τους πιο εξειδικευμένους, καθώς και όλες τις παραμέτρους που σχετίζονται με αυτούς. Είναι ένα εύχρηστο εργαλείο με ένα πολύ καλά οργανωμένο API και μπορεί να χρησιμοποιηθεί σε οποιαδήποτε περιβάλλον ανάπτυξης.

3.9 Περίληψη

Στο κεφάλαιο αυτό παρουσιάσαμε σχετικές εφαρμογές που υπάρχουν όσον αφορά τη δημιουργία γραφημάτων. Για κάθε μια από αυτές παρουσιάσαμε τα πλεονεκτήματα και τα μειονεκτήματα τους. Το πιο σημαντικό μειονέκτημα από όλα είναι ότι οι περισσότερες από τις εφαρμογές που υπάρχουν δεν είναι ελεύθερα λογισμικά. Επίσης ένα άλλο μειονέκτημα είναι ότι δεν είναι εύχρηστα. Για τους λόγους αυτούς, επιλέξαμε τη βιβλιοθήκη JFreeChart, ως το βασικό πλαίσιο εργασίας για την υλοποίηση της εφαρμογής μας.

Κεφάλαιο 4

Παρουσίαση των επιλογών παραμετροποίησης ενός γραφήματος

4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε όλες τις επιλογές παραμετροποίησης για κάθε τύπο γραφήματος που έχει στη διάθεση του ο εκάστοτε χρήστης, κατά το σχεδιασμό ενός πρότυπου γραφήματος. Το βασικό εργαλείο που χρησιμοποιήσαμε για το σχεδιασμό όλων των παραμέτρων που απαρτίζουν ένα γράφημα, είναι η βιβλιοθήκη JFreeChart. Το JFreeChart είναι μια δωρεάν Java βιβλιοθήκη γραφημάτων, η οποία διευκολύνει στην ανάπτυξη ποιοτικών και επαγγελματικών γραφημάτων. Προσφέρει ένα καλά τεκμηριωμένο και οργανωμένο API που υποστηρίζει ένα ευρύ φάσμα τύπων γραφημάτων, έναν ευέλικτο σχεδιασμό που είναι εύκολο να επεκταθεί και ικανοποιεί εφαρμογές server-side και client-side, υποστηρίζει μια πληθώρα τύπων εξαγωγής ενός γραφήματος όπως Swing εφαρμογές, αρχεία εικόνων(JPEG,PNG), PDF, διανυσματικές μορφές αρχείων γραφικής αναπαράστασης(SVG) κ.ά.

Το JFreeChart είναι ένα λογισμικό ανοιχτού κώδικα και διανέμεται κάτω από τους όρους του GNU Lesser General Public License (LGPL) [2].

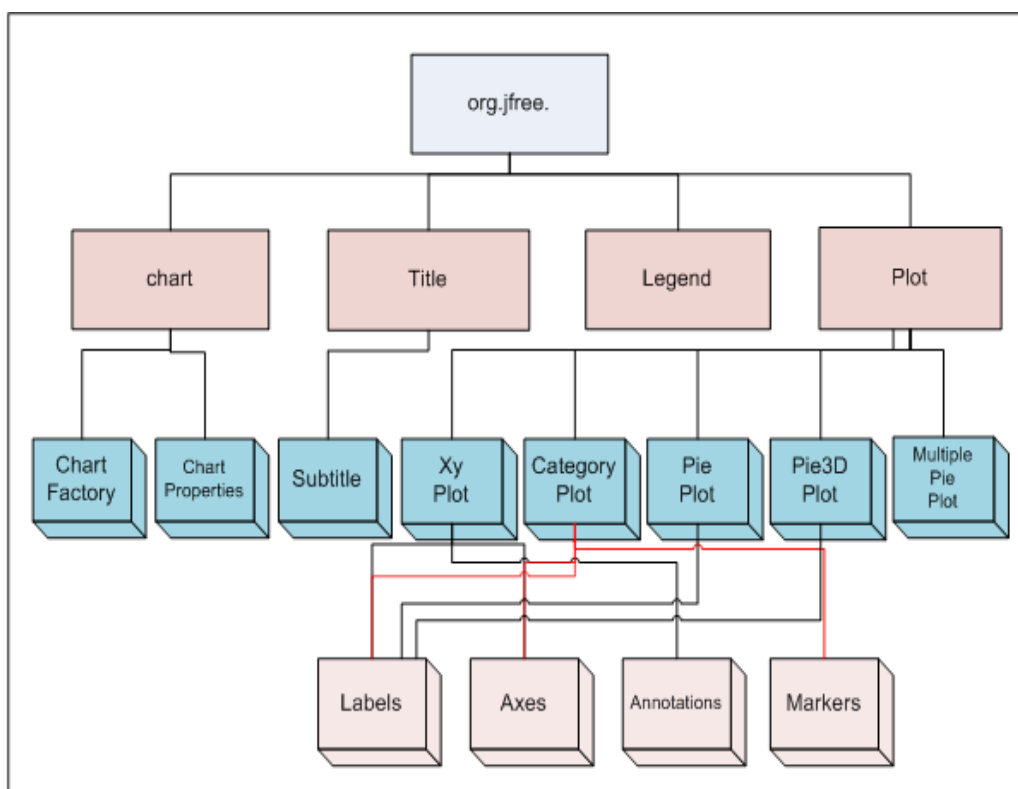
Το JFreeChart προσφέρει τη δυνατότητα δημιουργίας απλών γραφημάτων όπως Pie, Bar, Area, Line , αλλά και πιο περίπλοκων γραφημάτων όπως Histogram, Polar, Time Series, Gantt, Bubble, Step κ.ά.

4.2 Επισκόπηση της βιβλιοθήκης JFreeChart

Σε αυτό το σημείο θα παρουσιάσουμε συνοπτικά όλες τις δυνατότητες που προσφέρει η βιβλιοθήκη γραφημάτων JFreeChart, όσον αφορά την παραμετροποίηση των χαρακτηριστικών ενός γραφήματος. Αρχικά, θα παρουσιάσουμε τον τρόπο οργάνωσης των κλάσεων της βιβλιοθήκης JFreeChart και στη συνέχεια θα περιγράψουμε κάθε μια από τις κλάσεις αυτές. Στην εικόνα 3 που ακολουθεί, παρουσιάζεται ο τρόπος οργάνωσης της βιβλιοθήκης JFreeChart:

4.2.1 Τύποι Γραφημάτων

Το JFreeChart υποστηρίζει μια πληθώρα τύπων γραφημάτων που μπορούν να



Εικόνα 3 - Ιεραρχία κλάσεων της βιβλιοθήκης JFreeChart

χρησιμοποιηθούν για την απεικόνιση δεδομένων κάθε τύπου εφαρμογής. Οι τύποι γραφήματος περιέχονται στη κλάση Chart Factory που φαίνεται στην παραπάνω εικόνα (Εικόνα 3). Οι τύποι γραφημάτων που υποστηρίζει το JFreeChart, φαίνονται στον παρακάτω πίνακα:

Πίνακας 14 - Υποστηριζόμενοι τύποι γραφημάτων

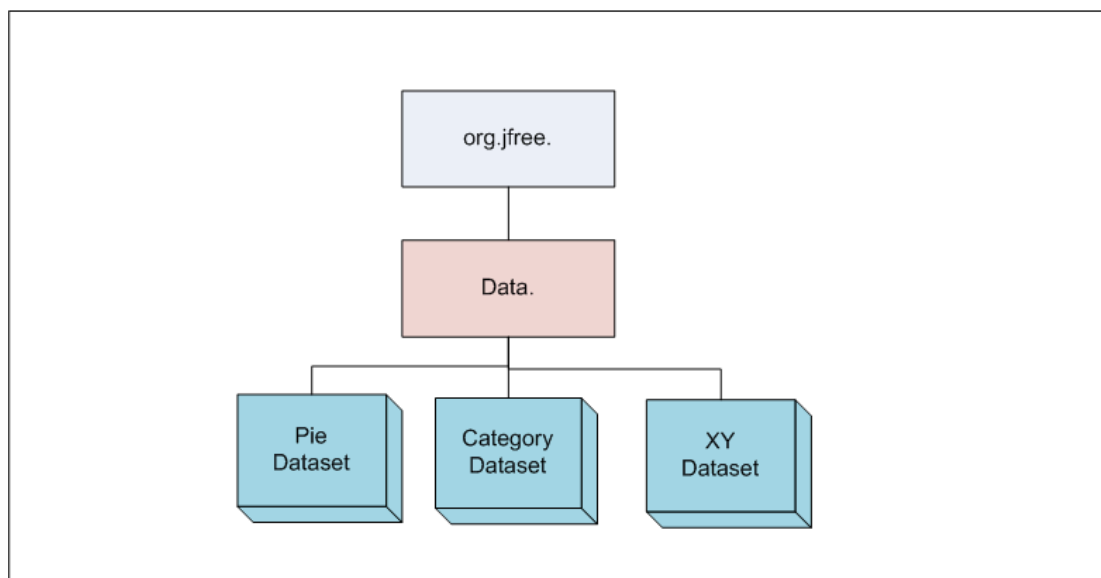
Area	Stacked Area	XY Area
Bar	Statistical Bar	Layered Bar

XY Bar	XY Step Area	Stacked Bar
Interval Bar	Pie	Pie3D
Ring	Multiple Pie	Multiple Pie 3D
Difference Pie	Line	Line3D
Multiple Line	Multiple Line 3D	XY Line
Scatter	Histogram	Polar
Time Series	Gantt	Bubble
XY Step	Box And Whisker	Combined Category Domain
Combined XY Range	Combined XY Domain	Combined Category Range

4.2.2 Κατηγορίες Δεδομένων

Η δημιουργία ενός dataset, είναι το πρώτο βήμα για την κατασκευή οποιουδήποτε γραφήματος. Είναι το βασικό κομμάτι για τη δημιουργία γραφήματος και είναι υποχρεωτικό να δημιουργηθεί. Το JFreeChart επιτρέπει την εισαγωγή δεδομένων με έναν ευέλικτο τρόπο, ανάλογα με τον τύπο γραφήματος που έχουμε επιλέξει, για να αναπαραστήσουμε τα δεδομένα.

Η ιεραρχία που ακολουθεί το JFreeChart όσον αφορά τα δεδομένα φαίνεται στην παρακάτω εικόνα:



Εικόνα 4 - Ιεραρχία των δεδομένων

Αν και ο τρόπος εισαγωγής δεδομένων ποικίλλει για κάθε κατηγορία γραφημάτων, στη γενική του μορφή απαρτίζεται από τα εξής στοιχεία :

1. Η κατηγορία των δεδομένων για απεικόνιση, που ουσιαστικά είναι μια ονομασία για το χαρακτηρισμό των δεδομένων
2. Η τιμή του x-άξονα
3. Η τιμή του y-άξονα

Οι κατηγορίες δεδομένων που υποστηρίζει το JFreeChart, παρουσιάζονται στον ακόλουθο πίνακα. Στον πίνακα αυτό, δίπλα από κάθε κατηγορία δεδομένων, φαίνονται και τα γραφήματα με τα οποία σχετίζεται κάθε κατηγορία.

Πίνακας 15 - Κατηγορίες δεδομένων

Κατηγορία δεδομένων	Τύποι γραφημάτων
Pie Dataset	Pie, Pie3D, Ring, Difference Pie
Category Dataset	Bar, Area, Stacked Area, Layered Bar, Interval Bar, Stacked Bar, Line, Line 3D, Multiple Pie, Multiple Pie3D, Gantt, Time Series, Box and Whisker, Statistical Bar
XY Dataset	Scatter, XY Line, XY Bar, Bubble, XY Step, XY Step Area, Polar, Histogram

4.2.3 Επιλογές παραμετροποίησης των χαρακτηριστικών ενός γραφήματος

Το JFreeChart προσφέρει μια πληθώρα επιλογών παραμετροποίησης για κάθε ένα από τα στοιχεία ενός γραφήματος. Με τον όρο στοιχεία γραφήματος αναφερόμαστε στον τίτλο του, υπότιτλο, στην λεζάντα (legend), στις ετικέτες δεδομένων(labels), στην περιοχή απεικόνισης δεδομένων (plot), και στους άξονες. Οι επιλογές παραμετροποίησης για κάθε ένα από τα παραπάνω στοιχεία, δεν είναι διαθέσιμες για όλους τους τύπους γραφημάτων αλλά ποικίλλουν για διάφορες κατηγορίες γραφημάτων. Παρακάτω ακολουθεί αναλυτική περιγραφή των επιλογών παραμετροποίησης από την εφαρμογή που σχεδιάσαμε.

4.3 Παρουσίαση των επιλογών παραμετροποίησης για κάθε τύπο γραφήματος

Η βιβλιοθήκη JFreeChart παρότι προσφέρει μια πληθώρα επιλογών παραμετροποίησης των χαρακτηριστικών ενός γραφήματος, δεν προσφέρει έναν εύχρηστο τρόπο καθορισμού των παραμέτρων αυτών από τον εκάστοτε χρήστη. Για το σκοπό αυτό υλοποιήσαμε ένα σύστημα που χρησιμοποιεί την βιβλιοθήκη JFreeChart ως rendering engine και επιτρέπει μέσω γραφικής διεπαφής, τον καθορισμό των παραμέτρων ενός γραφήματος. Στο σημείο αυτό θα παρουσιάσουμε αναλυτικά όλες τις διαγραμματικές παρουσιάσεις που μπορεί να κάνει αλληλεπιδραστικά ο εκάστοτε χρήστης του συστήματος μας.

4.3.1 Επιλογές των γενικών παραμέτρων ενός γραφήματος

Με τον όρο γενικές αναφερόμαστε σε παραμέτρους του γραφήματος σε ένα πιο υψηλό επίπεδο και έχουν να κάνουν με το border γύρω από το γράφημα, με το background χρώμα κ.ά. που θα δούμε στη συνέχεια. Οι γενικές παράμετροι βρίσκονται στη κλάση Chart Properties στην ιεραρχία του JFreeChart (Εικόνα 3).

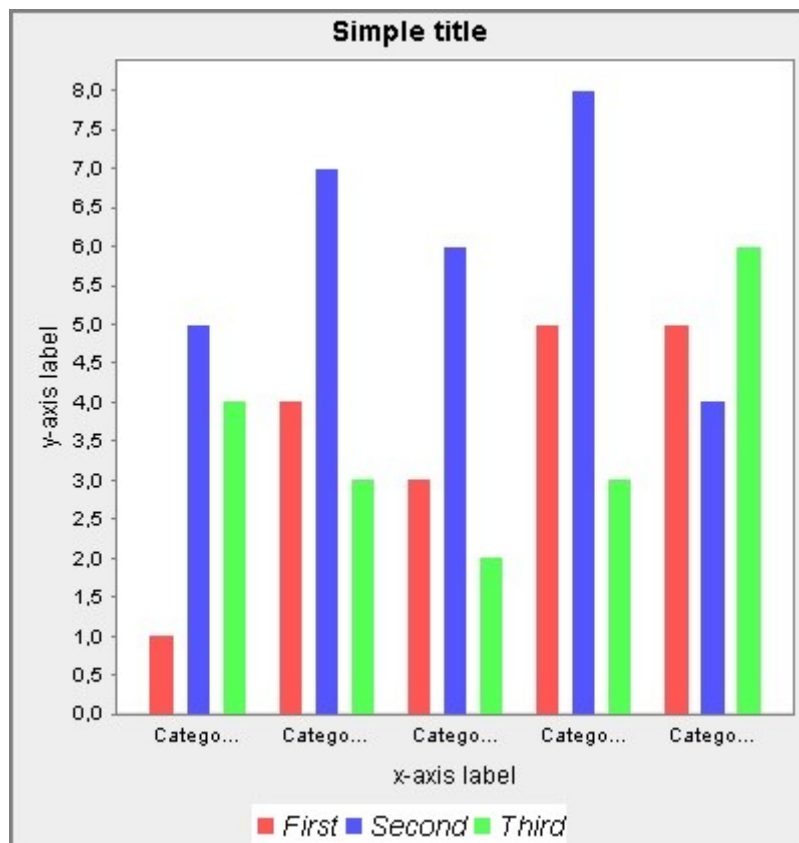
Σε αυτή την κατηγορία παραμέτρων, η εφαρμογή μας παρέχει τη δυνατότητα στον εκάστοτε χρήστη να προσθέσει border γύρω από το γράφημα, να καθορίσει το χρώμα του border και να επιλέξει πόσο έντονη θα είναι το border με τη προσθήκη ή όχι stroke στο border.

Έτσι αν ο χρήστης επιλέξει να προσθέσει border στο γράφημα, μπορεί να επεξεργαστεί και το χρώμα της γραμμής του border και να επιλέξει ένα από τα χρώματα που εμφανίζονται στον επόμενο πίνακα (Πίνακας 16) καθώς και να προσθέσει stroke για να κάνει πιο έντονη τη γραμμή του border:

Πίνακας 16 - Υποστηριζόμενα χρώματα

Blue	Cyan	Dark Gray	Gray	Light gray	Magenta
Pink	Yellow	Red	Dark Green	Orange	Dark Cyan
Light Cyan	Light Green	Dark Magenta	Light Magenta	Green	Dark Red
Light Red	Dark Blue	Dark Yellow	LightBlue	Light Yellow	White

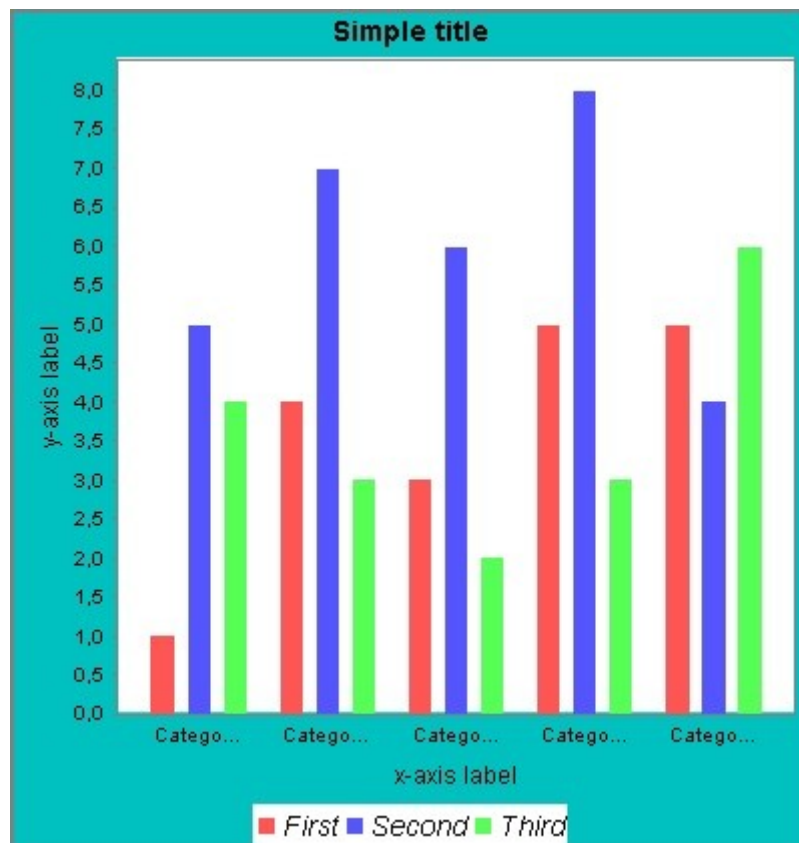
Στην επόμενη εικόνα, επιλέγουμε την προσθήκη border στο γράφημα, το χρώμα της γραμμής του border να είναι γκρι και προσθέτουμε stroke στο border:



Εικόνα 5 - Προσθήκη border σε χρώμα γκρι και stroke

Επιπλέον από το border γύρω από το γράφημα, το σύστημα μας επιτρέπει και την αλλαγή του καθορισμένου χρώματος που εμφανίζεται πίσω από το γράφημα. Η τιμή που μπορεί να πάρει είναι μια από αυτές που εμφανίζονται στον Πίνακα 16.

Στην επόμενη εικόνα αλλάζουμε το background χρώμα για το γράφημα σε dark cyan και έχουμε:



Εικόνα 6 - Αλλαγή του background χρώματος σε Dark Cyan

Τέλος, σε αυτό το επίπεδο το σύστημα μας παρέχει μια επιλογή σχετικά με το πόσο smoother θα είναι το γράφημα ή όχι. Η επιλογή αυτή ονομάζεται anti-alias και ο χρήστης μπορεί να επιλέξει αν θα είναι ενεργοποιημένη ή όχι.

4.3.2 Επεξεργασία των παραμέτρων του τίτλου ενός γραφήματος

Ο τίτλος ενός γραφήματος είναι ένα σύντομο κείμενο που περιγράφει τι απεικονίζει το συγκεκριμένο γράφημα και η καθορισμένη θέση του είναι στη κορυφή του γραφήματος, με οριζόντια ευθυγράμμιση στο κέντρο και το χρώμα του κειμένου είναι μαύρο. Η καθορισμένη γραμματοσειρά για τον τίτλο είναι Sans Serif, 12, bold. Οι παράμετροι για τον τίτλο βρίσκονται στην κλάση Title στην ιεραρχία του JFreeChart (Εικόνα 3).

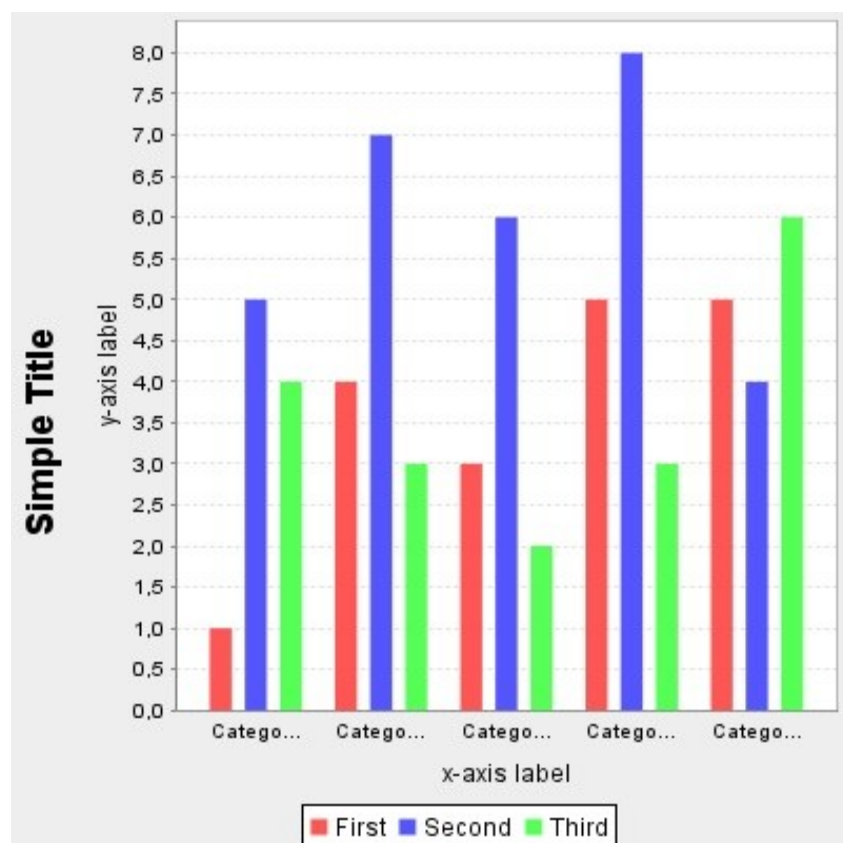
Ωστόσο η εφαρμογή που σχεδιάσαμε, επιτρέπει την αλλαγή των καθορισμένων παραμέτρων για τον τίτλο και οι αλλαγές που μπορούν να γίνουν παρουσιάζονται παρακάτω:

4.3.2.1 Επεξεργασία της θέσης του τίτλου

Όπως αναφέραμε παραπάνω η καθορισμένη θέση για τον τίτλο είναι στην κορυφή (top) του γραφήματος. Οι υπόλοιπες δυνατές θέσεις για τον τίτλο είναι δεξιά (right), αριστερά (left) και κάτω (bottom) από το γράφημα. Επιπλέον, για κάθε θέση του γραφήματος μπορεί να αλλάξει και η ευθυγράμμιση του. Οι δυνατές τιμές για την ευθυγράμμιση του τίτλου είναι: στο κέντρο (center), δεξιά (right) και αριστερά (left).

Στην παρακάτω εικόνα παρουσιάζεται το ίδιο γράφημα με παραπάνω αλλά η θέση του τίτλου είναι αριστερά και ευθυγράμμιση στο κέντρο :

4.3.2.2

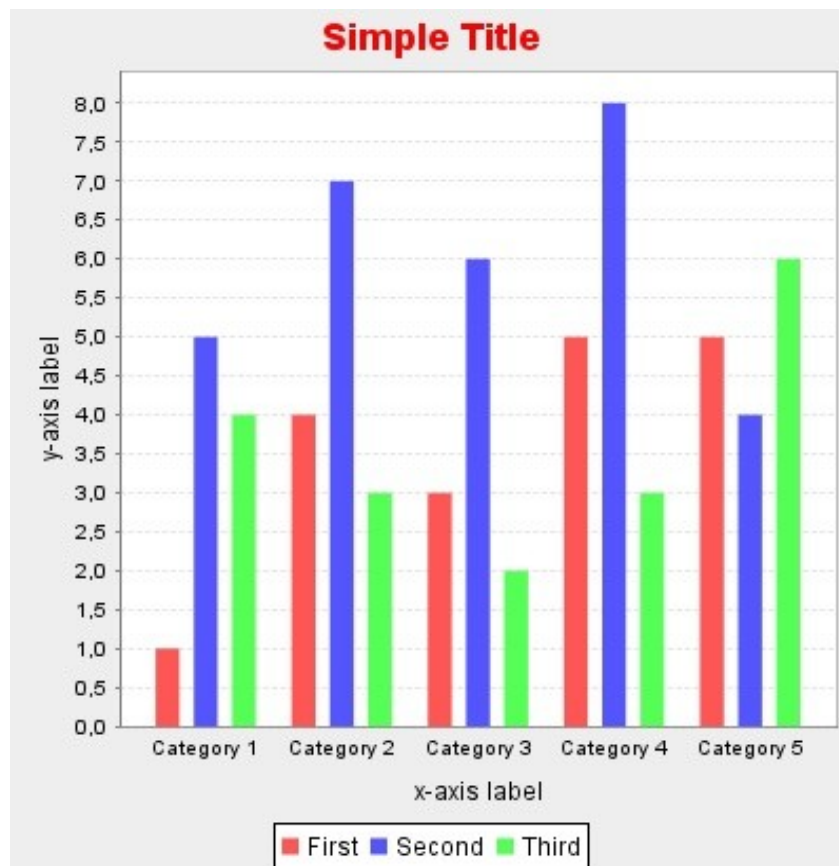


Εικόνα 7 - Αλλαγή της θέσης του τίτλου

Επεξεργασία των χρωμάτων

Το καθορισμένο χρώμα για τον τίτλο, είναι το μαύρο. Το χρώμα μπορεί να αλλάξει και να πάρει μια από τις δυνατές τιμές του Πίνακα 15.

Στην επόμενη εικόνα αλλάζουμε το χρώμα του κειμένου από μαύρο σε κόκκινο και έχουμε :



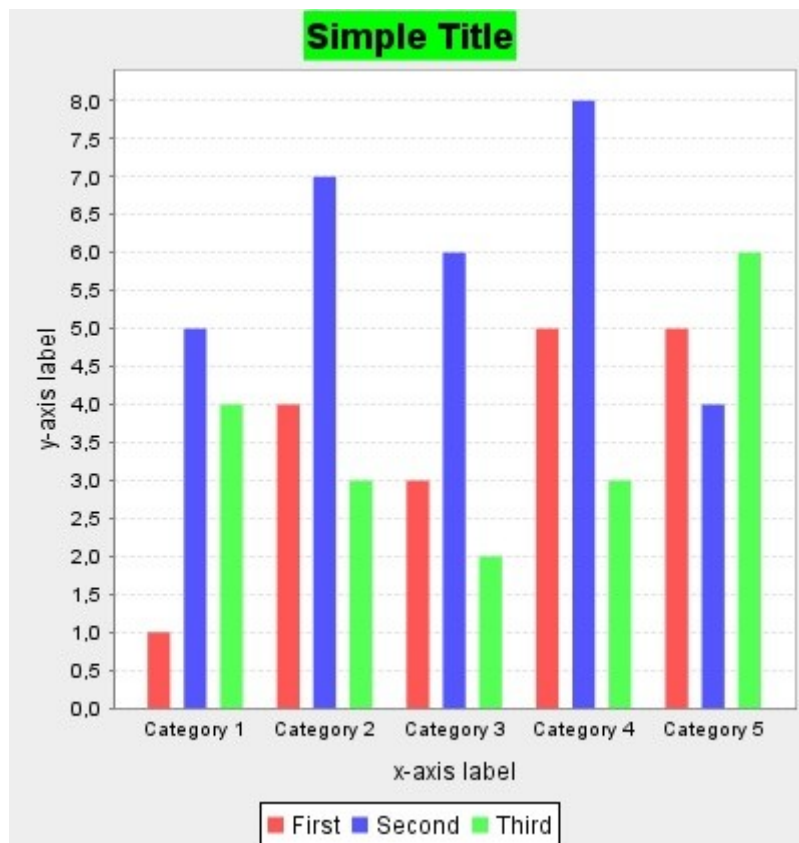
Εικόνα 8 - Αλλαγή του χρώματος του τίτλου

4.3.2.3 Μορφοποίηση της ευρύτερης περιοχής του τίτλου

Ο εκάστοτε χρήστης μπορεί να ορίσει τις εξής παραμέτρους που αφορούν την ευρύτερη θέση που βρίσκεται ο τίτλος:

1. Το χρώμα πίσω από τον τίτλο (background), που μπορεί να πάρει μια από τις δυνατές τιμές από τον Πίνακα 16.
2. Το ύψος και το μήκος της δεσμευμένης περιοχής για τον τίτλο
3. Να καθορίσει το εύρος για padding και margin
4. Να προσθέσει border γύρω από τον τίτλο καθορίζοντας το πάχος του.

Στην επόμενη εικόνα αλλάζουμε το background χρώμα του τίτλου, σε πράσινο, και καθορίζουμε το ύψος και το μήκος για τον τίτλο:



Εικόνα 9 - Αλλαγή του background χρώματος του τίτλου

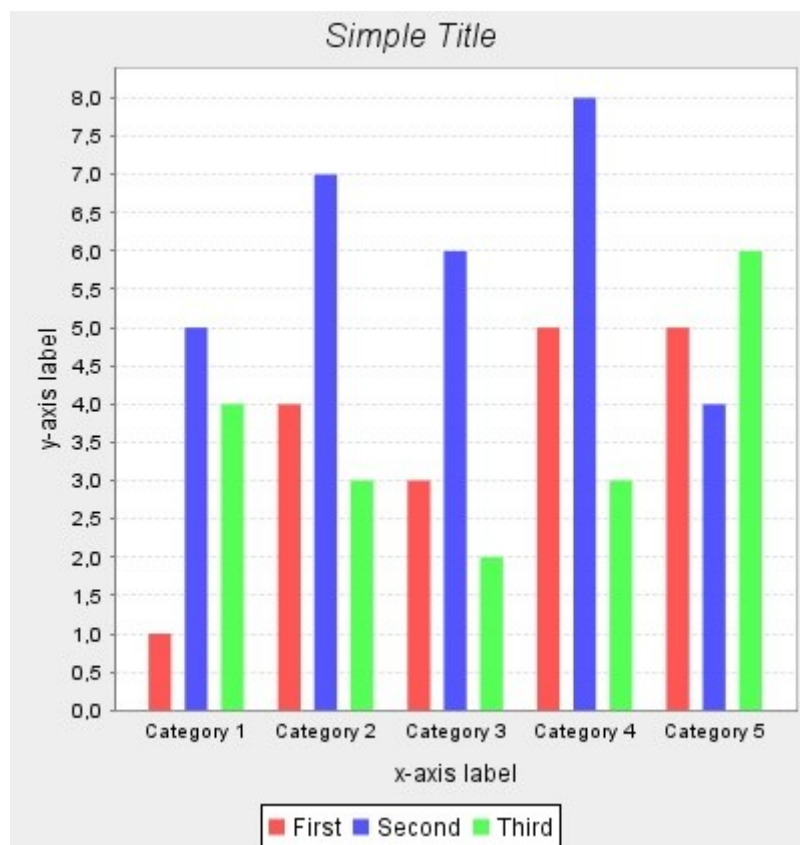
4.3.2.4 Επεξεργασία της γραμματοσειράς του τίτλου

Όπως αναφέραμε παραπάνω, η καθορισμένη γραμματοσειρά για τον τίτλο είναι Sans Serif, 12, bold. Οι δυνατές επιλογές για την γραμματοσειρά φαίνονται στον παρακάτω πίνακα και να σημειώσουμε εδώ ότι είναι δυνατός οποιοσδήποτε συνδυασμός από τις παρακάτω επιλογές :

Πίνακας 17 - Επιλογές γραμματοσειράς

Font	Font Size	Font Options
Times New Roman Arial Arial Black Calibri	10, 14, 16, 18, 20, 22, 24	Bold, italic, plain, Roman, True-Type-font, Hanging, Center, Type-1-Font, Left-To-Right, No-limit, Right-To-Left

Στην παρακάτω εικόνα αλλάζουμε την γραμματοσειρά για τον τίτλο σε Times New Roman, 16, italic και έχουμε:



Εικόνα 10 - Αλλαγή της γραμματοσειράς του τίτλου

Στο σημείο αυτό να αναφέρουμε ότι υπάρχει η δυνατότητα να διαφοροποιηθεί μία, δύο ή και όλες οι παράμετροι για τη γραμματοσειρά του τίτλου. Οποιαδήποτε παράμετρο δεν αλλάξει παραμένει στην καθορισμένη τιμή.

Τέλος, να σημειώσουμε ότι η μορφοποίηση της γραμματοσειράς του τίτλου είναι ένα ανεξάρτητο κομμάτι από την μορφοποίηση του τίτλου. Δηλαδή, ο τίτλος σε ένα γράφημα προστίθεται από τη στιγμή που δημιουργείται το γράφημα και έτσι είναι δυνατόν να γίνει μορφοποίηση της γραμματοσειράς του τίτλου χωρίς να έχει προηγηθεί κάποια επεξεργασία των παραμέτρων του τίτλου.

4.3.3 Προσθήκη και μορφοποίηση υπότιτλου στο γράφημα

Ο υπότιτλος είναι ένα από τα προαιρετικά στοιχεία που μπορούν να προστεθούν στο γράφημα μετά τη δημιουργία του. Ως υπότιτλος εμφανίζεται το κείμενο που καθορίζει ο εκάστοτε χρήστης και η καθορισμένη θέση για τον υπότιτλο είναι στην κορυφή του γραφήματος, κάτω από τον τίτλο, με οριζόντια ευθυγράμμιση στο κέντρο. Η καθορισμένη γραμματοσειρά για τον υπότιτλο είναι Sans Serif, 10,

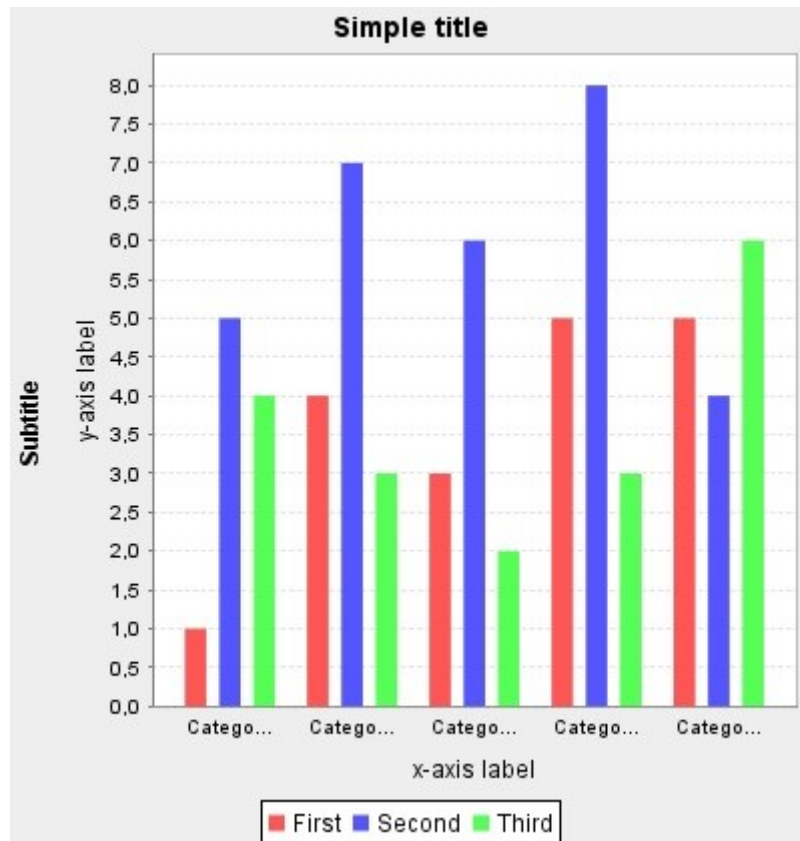
bold. Οι παράμετροι για τον υπότιτλο βρίσκονται στη κλάση Subtitle στην ιεραρχία του JFreeChart (Εικόνα 3).

Οι παράμετροι που καθορίζουν τον υπότιτλο και οι αλλαγές πάνω σε αυτές είναι παρόμοιες με αυτές που παρέχονται για τον τίτλο, με τη διαφορά ότι δεν υπάρχει η δυνατότητα να αλλάξει η καθορισμένη γραμματοσειρά για τον υπότιτλο. Παρακάτω ακολουθεί λεπτομερής ανάπτυξη και παραδείγματα για τις παραμέτρους του υπότιτλου.

4.3.3.1 Επεξεργασία της θέσης του υπότιτλου

Όπως αναφέραμε παραπάνω η καθορισμένη θέση για τον υπότιτλο είναι στην κορυφή (top) του γραφήματος κάτω από τον τίτλο. Οι υπόλοιπες δυνατές θέσεις για τον υπότιτλο είναι δεξιά (right), αριστερά (left) και κάτω (bottom) από το γράφημα. Επιπλέον, για κάθε θέση του γραφήματος μπορεί να αλλάξει και η ευθυγράμμιση του. Οι δυνατές τιμές για την ευθυγράμμιση του υπότιτλου είναι: στο κέντρο (center), δεξιά (right) και αριστερά (left) και κάτω (bottom).

Στην παρακάτω εικόνα η θέση του υπότιτλου είναι αριστερά και ευθυγράμμιση στο κέντρο :



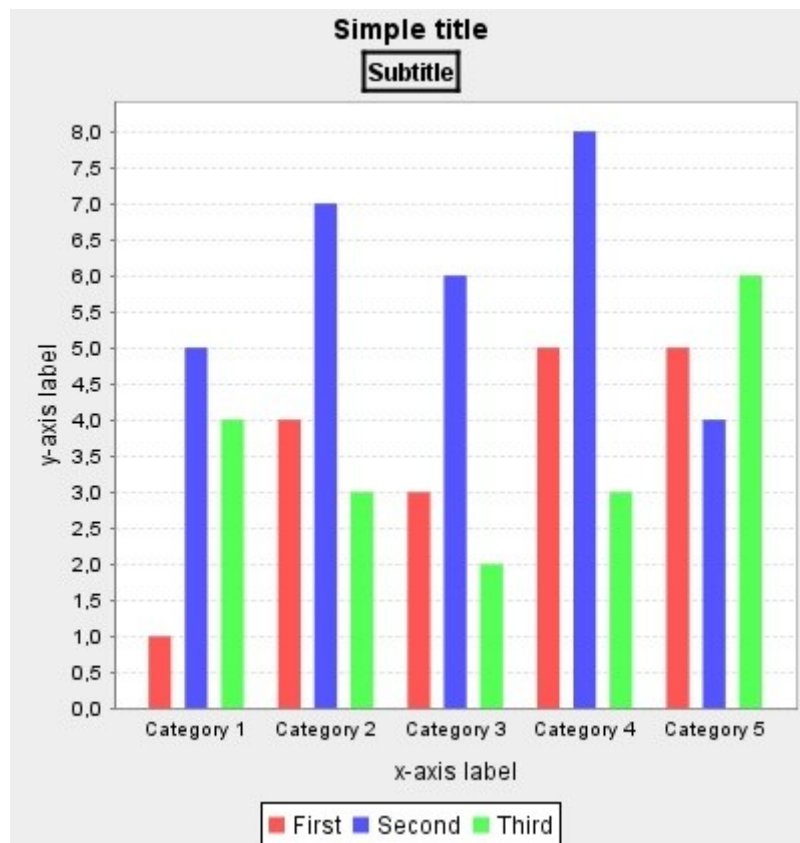
Εικόνα 11 - Αλλαγή της θέσης του υπότιτλου

4.3.3.2 Μορφοποίηση της ευρύτερης περιοχής του υπότιτλου

Ο εκάστοτε χρήστης μπορεί να ορίσει τις εξής παραμέτρους που αφορούν την ευρύτερη θέση που βρίσκεται ο υπότιτλος:

1. Το ύψος και το μήκος της δεσμευμένης περιοχής για τον υπότιτλο
2. Να καθορίσει το εύρος για padding και margin
3. Να προσθέσει border γύρω από τον υπότιτλο καθορίζοντας το πάχος του.

Στην επόμενη εικόνα προσθέτουμε border γύρω από τον υπότιτλο, θέτοντας τέσσερις τιμές (δεξιά, αριστερά, πάνω, κάτω) για το πάχος που θα έχει το border σε κάθε περίπτωση και έχουμε:



Εικόνα 12 - Προσθήκη border γύρω από τον υπότιτλο

4.3.4 Μορφοποίηση της λεζάντας του γραφήματος

Κατά τη δημιουργία του γραφήματος, ο εκάστοτε χρήστης επιλέγει εάν επιθυμεί να υπάρχει ή όχι λεζάντα (legend) στο γράφημα. Αν ο χρήστης επιλέξει να εμφανίζεται η λεζάντα, τότε μπορεί να επεξεργαστεί και τις σχετικές παραμέτρους. Η καθορισμένη θέση για τη λεζάντα είναι κάτω (bottom) από το γράφημα, με ευθυγράμμιση στο κέντρο και η καθορισμένη γραμματοσειρά είναι Sans Serif, 12.

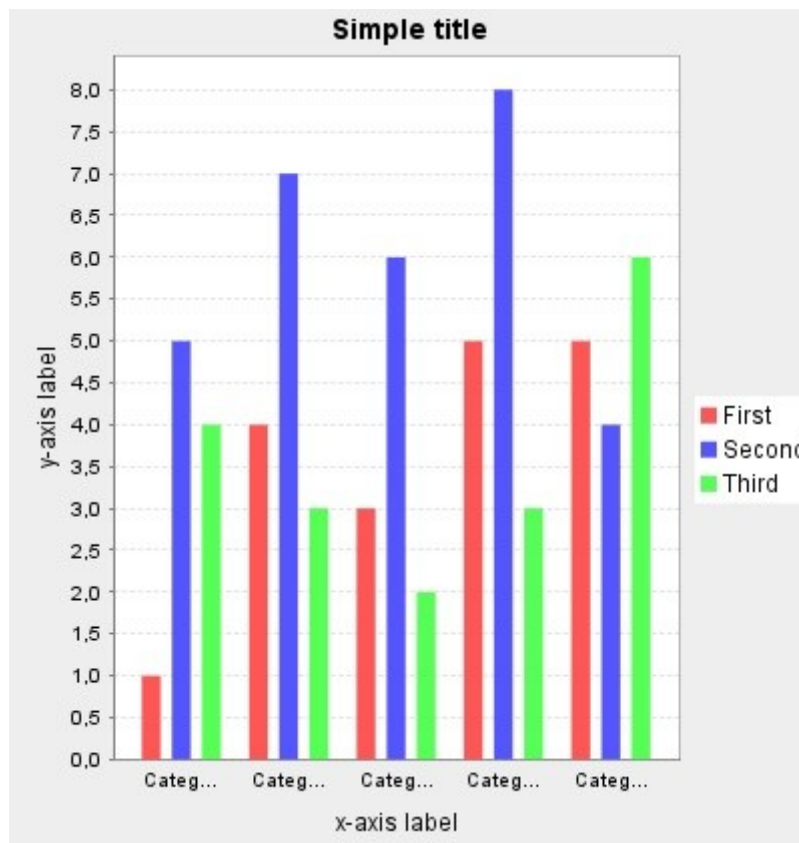
Οι παράμετροι που μπορούν να καθοριστούν από το χρήστη περιγράφονται αναλυτικά παρακάτω:

4.3.4.1 Επεξεργασία της θέσης της λεζάντας

Όπως αναφέραμε παραπάνω η καθορισμένη θέση για την λεζάντα είναι κάτω (bottom) από το γράφημα. Οι υπόλοιπες δυνατές θέσεις για την λεζάντα είναι δεξιά (right), αριστερά (left) και πάνω (top) από το γράφημα. Επιπλέον, για κάθε θέση του γραφήματος μπορεί να αλλάξει και η ευθυγράμμιση του. Οι δυνατές τιμές για την ευθυγράμμιση της λεζάντας είναι: στο κέντρο (center), δεξιά (right) και αριστερά

(left). Οι παράμετροι για τη λεζάντα βρίσκονται στη κλάση Legend στην ιεραρχία του JFreeChart (Εικόνα 3).

Στην παρακάτω εικόνα η θέση της λεζάντας είναι δεξιά και ευθυγράμμιση στο κέντρο :

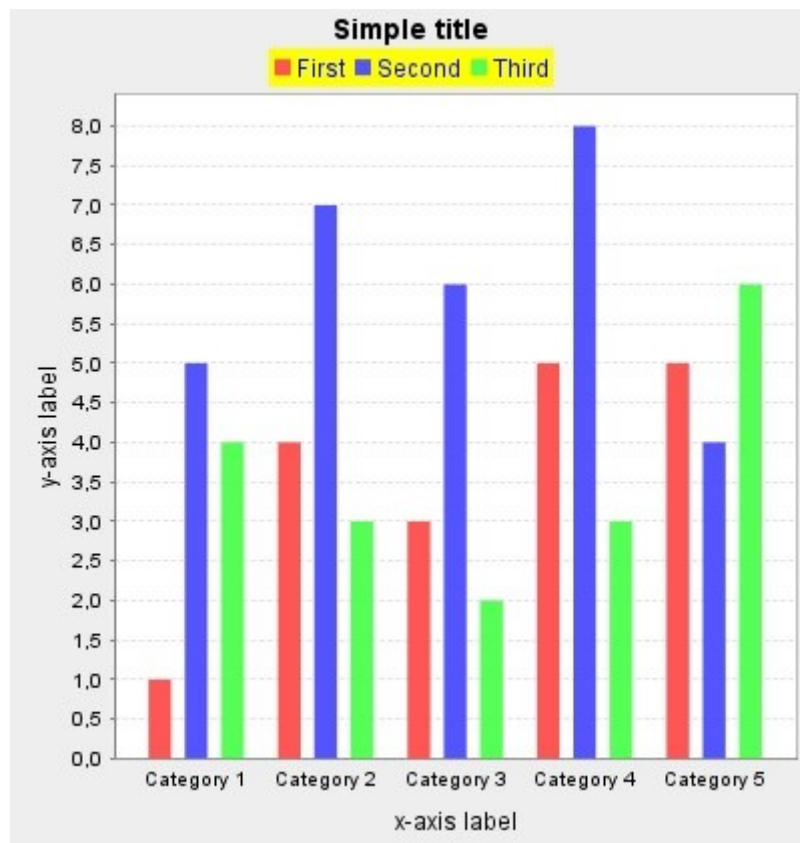


Εικόνα 13 - Αλλαγή της θέσης της λεζάντας (δεξιά)

4.3.4.2 Επεξεργασία των χρωμάτων της λεζάντας

Είναι δυνατόν να αλλάξει το χρώμα κειμένου της λεζάντας καθώς και το χρώμα στην περιοχή πίσω από το κείμενο (background). Και στις δυο περιπτώσεις οι δυνατές τιμές για το χρώμα είναι αυτές που παρουσιάσαμε στον Πίνακα 16.

Στην παρακάτω εικόνα έχουμε αλλάξει το χρώμα κειμένου της λεζάντας σε μπλε και το background χρώμα σε κίτρινο και έχουμε:

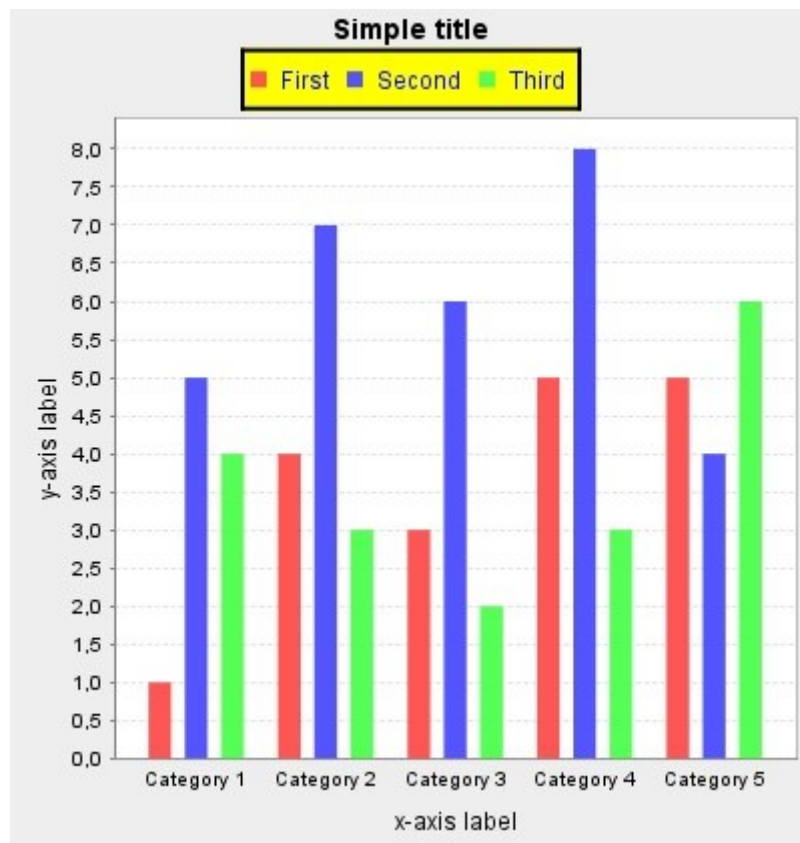


Εικόνα 14 - Επεξεργασία των χρωμάτων της λεζάντας

4.3.4.3 Μορφοποίηση της λεζάντας

Όπως και στην περίπτωση του τίτλου, έτσι και εδώ ο χρήστης μπορεί να καθορίσει τις τιμές για padding, margin και border γύρω από τη λεζάντα αλλά και για τα σύμβολα της λεζάντας. Δηλαδή, μπορούμε να ορίσουμε τιμές για το padding γύρω από τα σύμβολα (στη συγκεκριμένη περίπτωση, τετράγωνα) που εμφανίζονται στη λεζάντα (legend insets).

Αντίστοιχα, μπορούμε να προσθέσουμε και border γύρω από τη λεζάντα θέτοντας τέσσερις τιμές (δεξιά, αριστερά, πάνω, κάτω) για το πάχος που θα έχει το border σε κάθε περίπτωση και έχουμε:

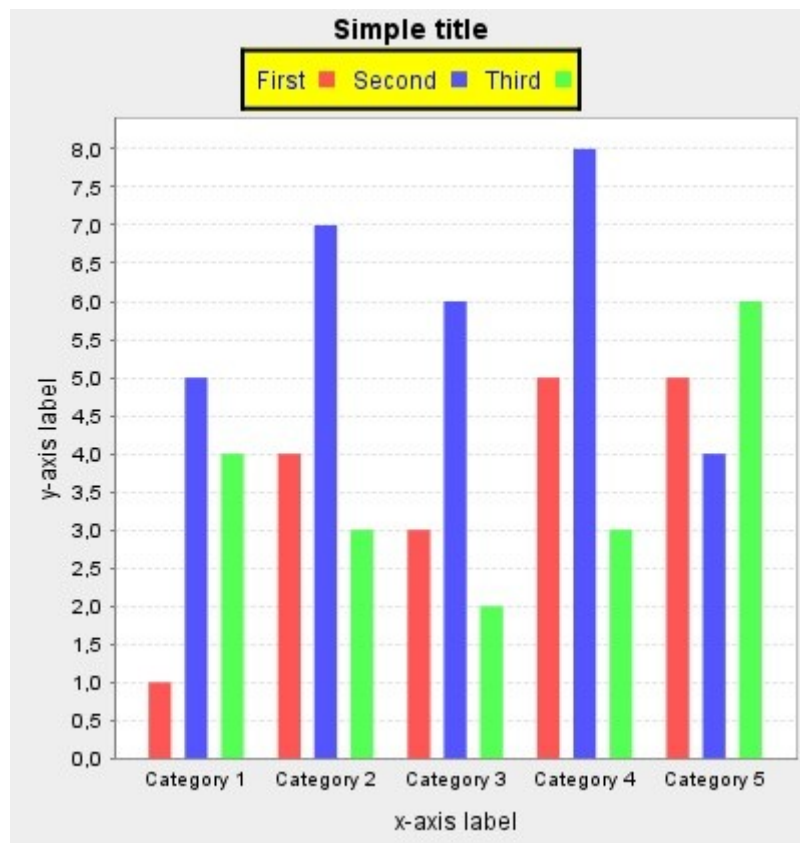


Εικόνα 15 - Προσθήκη border γύρω από τη λεζάντα

4.3.4.4 Επεξεργασία της θέσης των συμβόλων της λεζάντας

Εκτός από την θέση της λεζάντας είναι δυνατή και η αλλαγή της θέσης των συμβόλων που φαίνονται στη λεζάντα για απεικόνιση. Για παράδειγμα, στο παραπάνω γράφημα, μέσα στη λεζάντα εμφανίζονται αριστερά τα σύμβολα (τετράγωνα) και ακολουθεί το κείμενο. Αυτή η θέση είναι και η καθορισμένη. Οι υπόλοιπες δυνατές θέσεις είναι δεξιά, πάνω, και κάτω από το κείμενο.

Στην επόμενη εικόνα, αλλάζουμε τη θέση των συμβόλων της λεζάντας και τα τοποθετούμε δεξιά του κειμένου και έχουμε:

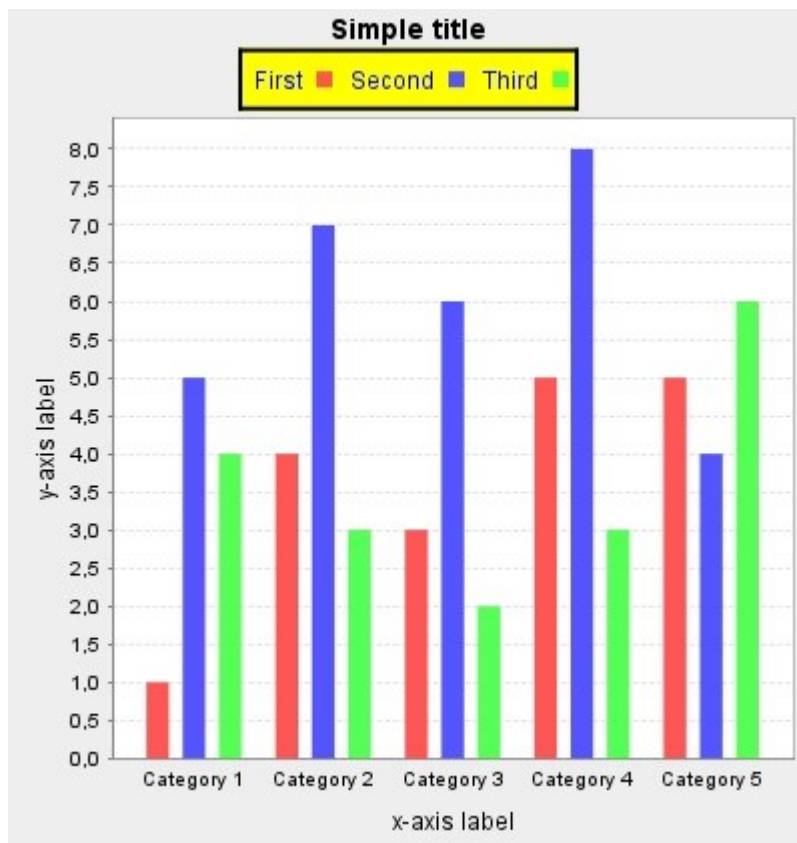


Εικόνα 16 - Αλλαγή της θέσης των συμβόλων της λεζάντας

4.3.4.5 Επεξεργασία γραμματοσειράς της λεζάντας

Όπως αναφέραμε παραπάνω, η καθορισμένη γραμματοσειρά για την λεζάντα είναι Sans Serif, 12. Οι δυνατές επιλογές για την γραμματοσειρά είναι οι ίδιες με αυτές που παρουσιάσαμε στον Πίνακα 17.

Στην παρακάτω εικόνα αλλάζουμε την γραμματοσειρά για την λεζάντα από την καθορισμένη επιλογή, Sans Serif , 12, σε Times New Roman, 14, italic και έχουμε:



Εικόνα 17 - Αλλαγή της γραμματοσειράς της λεζάντας

Όπως αναφέραμε και στην περίπτωση του τίτλου, είναι δυνατή η επεξεργασία οποιασδήποτε ή όλων των παραμέτρων για την γραμματοσειρά της λεζάντας και μπορεί να γίνει χωρίς να έχει προηγηθεί κάποια επεξεργασία των παραμέτρων της λεζάντας που αναφέραμε παραπάνω.

4.3.5 Προσθήκη labels στη περιοχή απεικόνισης δεδομένων

Όσον αφορά την προσθήκη labels πάνω στην περιοχή απεικόνισης δεδομένων, το σύστημα παρέχει στον χρήστη επιλογές που αφορούν τον τρόπο εμφάνισης των labels (χρώματα και γραμματοσειρά) και τον τρόπο απεικόνισης τους.

Πριν ξεκινήσουμε με την παρουσίαση των παραμέτρων για τα labels, να αναφέρουμε ότι πάνω σε ένα dataset με περισσότερες από μια κατηγορίες δεδομένων (data series), υπάρχουν δύο επιλογές στην προσθήκη label πάνω στα δεδομένα :

1. Να έχουν όλες οι κατηγορίες δεδομένων το ίδιο label, δηλαδή το label να έχει την ίδια εμφάνιση και τρόπο απεικόνισης για όλες τις κατηγορίες δεδομένων και

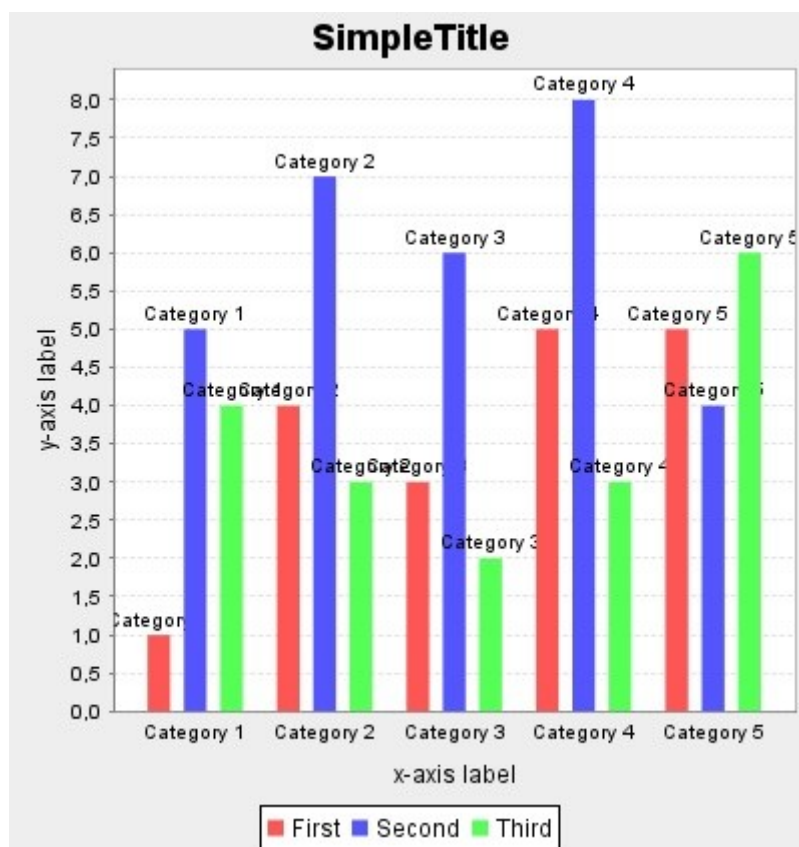
Κεφάλαιο 4 – Παρουσίαση των επιλογών παραμετροποίησης ενός γραφήματος

2. Κάθε κατηγορία δεδομένων να έχει το δικό της label, προσαρμοσμένο διαφορετικά από τις υπόλοιπες.

Για dataset που έχει μόνο μια κατηγορία δεδομένων, όπως είναι λογικό η επιλογή είναι μια.

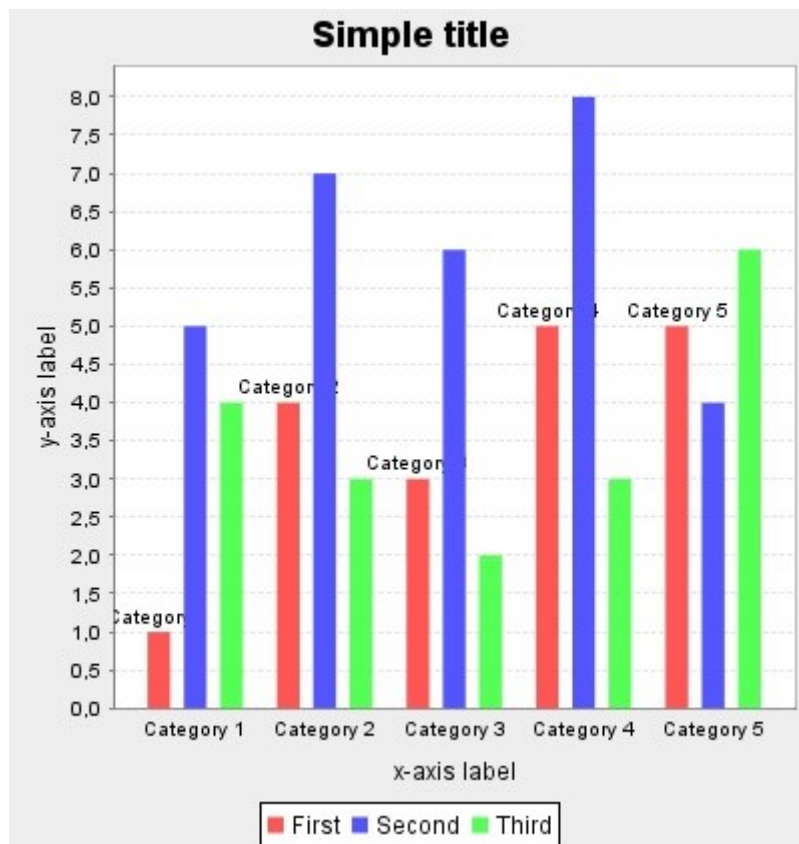
Οι τύποι γραφημάτων που υποστηρίζουν την προσθήκη label σε δεδομένα, είναι όλοι οι τύποι που αναφέραμε στον Πίνακα 13 εκτός από τους: Polar, XY Area, Multiple Pie, Multiple Pie 3D, XY Step Area, Stacked Area.

Ένα απλό παράδειγμα ενός γραφήματος, με dataset που αποτελείται από πολλές κατηγορίες δεδομένων είναι το γράφημα στο παράδειγμα μας παραπάνω. Σε αυτό το γράφημα υπάρχουν τρεις κατηγορίες δεδομένων: First, Second, Third, οι οποίες και απεικονίζονται με τα αντίστοιχα χρώματα. Αν τώρα επιλέξουμε να εμφανίζουμε το ίδιο label σε όλες τις κατηγορίες δεδομένων, τότε θα έχουμε:



Εικόνα 18 - Προσθήκη label στο γράφημα

Αντίστοιχα, αν θέλαμε να προσθέσουμε label μόνο στην κατηγορία δεδομένων με το όνομα First (κόκκινο χρώμα), θα είχαμε το εξής:



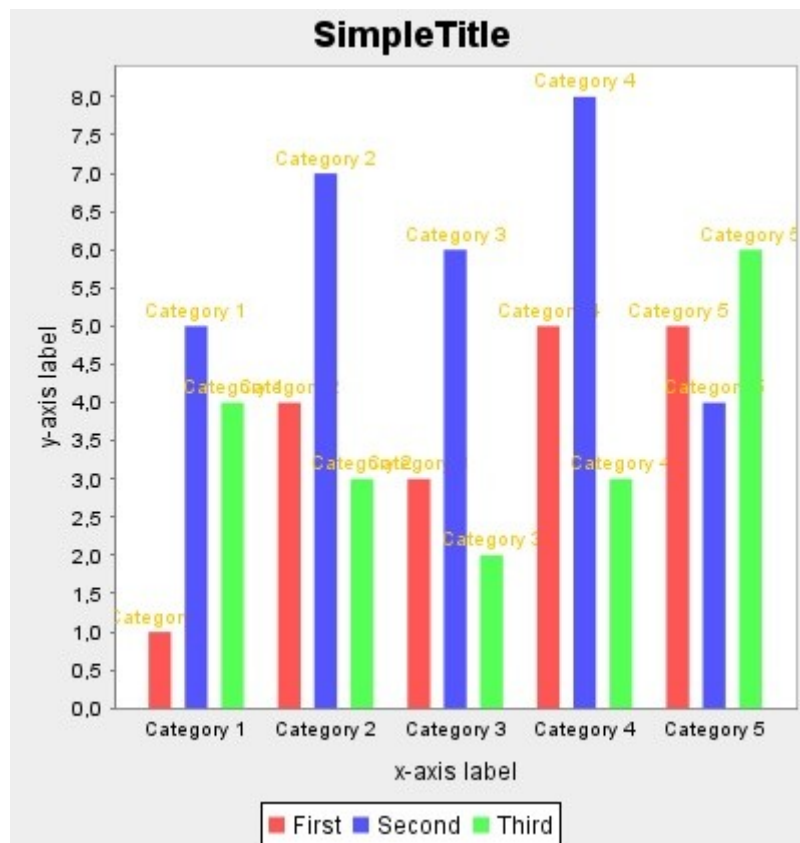
Εικόνα 19 - Προσθήκη label σε ένα μόνο series

Και στις δύο περιπτώσεις, οι παράμετροι για τα labels είναι ακριβώς ίδιες, απλά υπάρχει η επιλογή είτε να προστίθεται σε όλα τα δεδομένα το ίδιο label είτε μόνο σε συγκεκριμένες κατηγορίες.

4.3.5.1 Εμφάνιση των labels

- **Επεξεργασία χρωμάτων**

Το καθορισμένο χρώμα για το label είναι το μαύρο. Ωστόσο, αυτό μπορεί να αλλάξει και να γίνει ένα από τα χρώματα που φαίνονται στον Πίνακα 16. Στην παρακάτω εικόνα έχουμε αλλάξει το χρώμα σε πορτοκαλί και έχουμε:

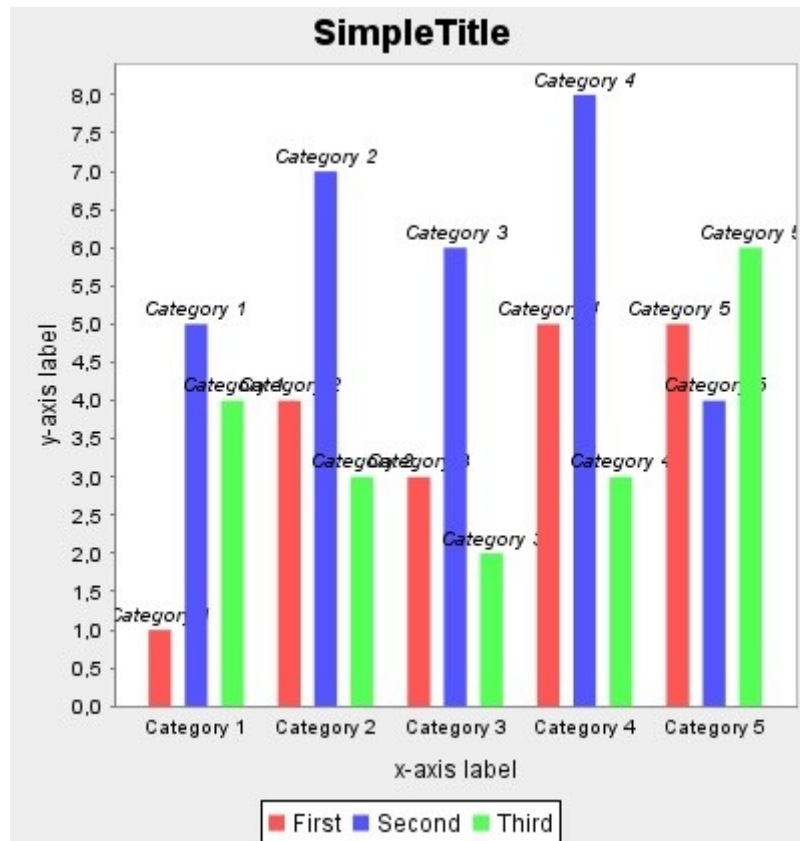


Εικόνα 20 - Αλλαγή χρώματος των labels

- **Γραμματοσειρά**

Η καθορισμένη γραμματοσειρά για τα labels είναι Sans Serif, 10, bold. Ωστόσο, οι επιλογές για τη γραμματοσειρά είναι αυτές που παρουσιάστηκαν στον Πίνακα 17 και ισχύουν όλα αυτά που αναφέραμε για τη περίπτωση του τίτλου και της λεζάντας.

Στην επόμενη εικόνα, αλλάζουμε την γραμματοσειρά για τα labels από την καθορισμένη σε Arial, 14, italic και έχουμε:



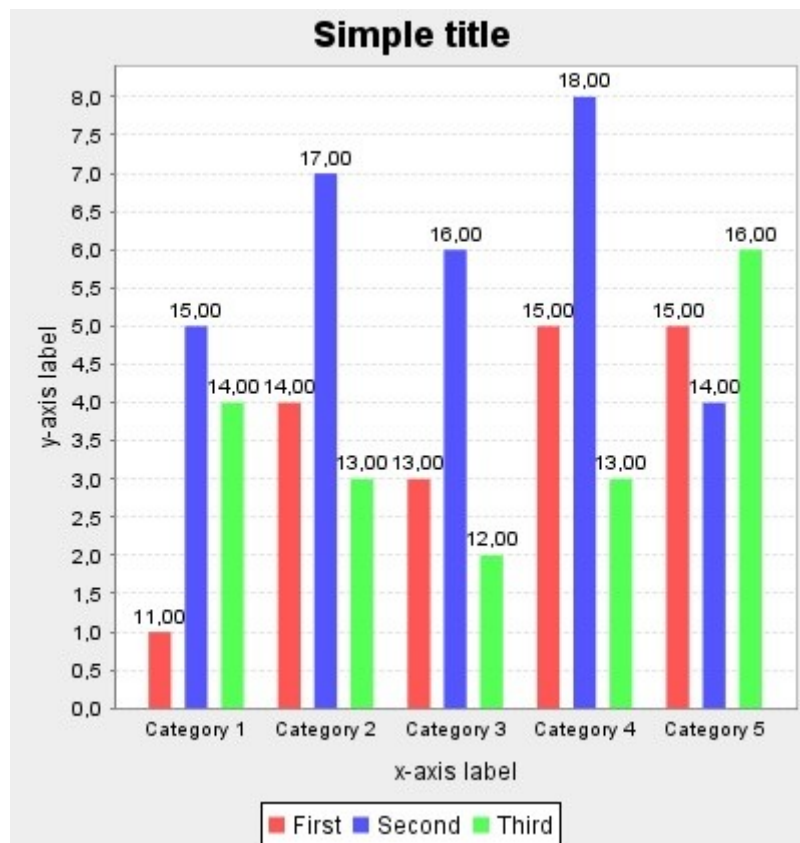
Εικόνα 21 - Αλλαγή γραμματοσειράς στα labels

4.3.5.2 Τρόπος απεικόνισης των labels

Όσον αφορά τον τρόπο απεικόνισης των labels, επειδή έχει άμεση σχέση με τα δεδομένα, υπάρχουν τρεις επιλογές:

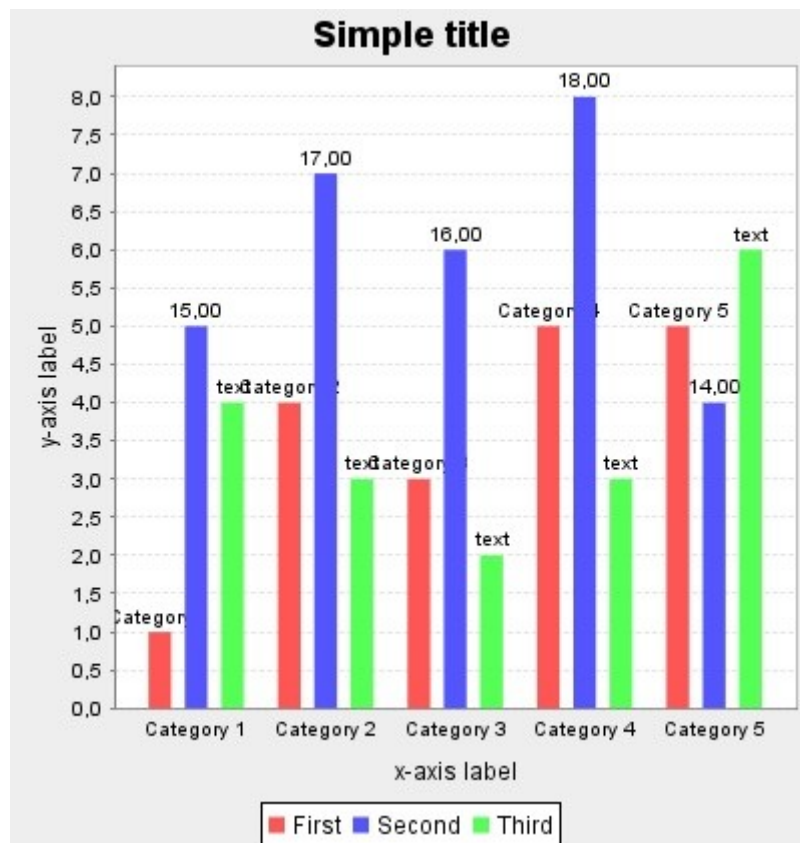
- Να εμφανίζονται ως labels οι τιμές του άξονα x
- Να εμφανίζονται ως labels οι τιμές του άξονα y
- Να εμφανίζεται ως label ένα κείμενο που καθορίζει ο χρήστης

Σε όλα τα προηγούμενα παραδείγματα για τα labels, επιλέξαμε ως τρόπο απεικόνισης τις τιμές του άξονα x . Στην παρακάτω εικόνα απεικονίζονται οι τιμές του άξονα y για το ίδιο γράφημα και έχουμε:



Εικόνα 22 - Τρόπος εμφάνισης των labels

Τέλος να αναφέρουμε ότι στην περίπτωση που επιλέξουμε να προσθέσουμε διαφορετικό Label για κάθε κατηγορία δεδομένων, είναι δυνατόν για κάθε κατηγορία να επιλέξουμε διαφορετικό τρόπο απεικόνισης των labels. Για παράδειγμα στην επόμενη εικόνα, επιλέγουμε για την πρώτη κατηγορία δεδομένων να απεικονίσουμε τις τιμές του άξονα – x, για τη δεύτερη κατηγορία τις τιμές του άξονα-y και για την τρίτη κατηγορία εισάγουμε ένα απλό κείμενο:



Εικόνα 23 - Διάφοροι τρόποι εμφάνισης των labels

4.3.6 Μορφοποίηση της περιοχής απεικόνισης δεδομένων

Η περιοχή απεικόνισης δεδομένων αναφέρεται με τον όρο plot (Εικόνα 3). Υπάρχουν έξι διαφορετικοί τύποι plot, καθένας από τους οποίους σχετίζεται με συγκεκριμένους τύπους γραφημάτων. Στον επόμενο πίνακα, παρουσιάζονται όλοι οι τύποι plot που υποστηρίζει η εφαρμογή και δίπλα οι τύποι γραφημάτων με τους οποίους σχετίζεται:

Πίνακας 18 - Υποστηριζόμενοι τύποι Plot

Τύποι Plot	Τύποι γραφημάτων
Category Plot	Area, Bar, Line, Line3D, Stacked Bar, Interval Bar, Layered Bar, Statistic Bar, Box and Whisker, Gantt, Multiple Line, MultipleLine3D
XY Plot	Histogram, Scatter, XY Bar, XY Line, XY Step Area, XY Step, Time Series, XY Area, Stacked XY Area
Polar Plot	Polar
Pie Plot	Pie, Pie 3D, Difference Pie
Ring Plot	Ring
Multiple Pie Plot	Multiple Pie, Multiple Pie 3D

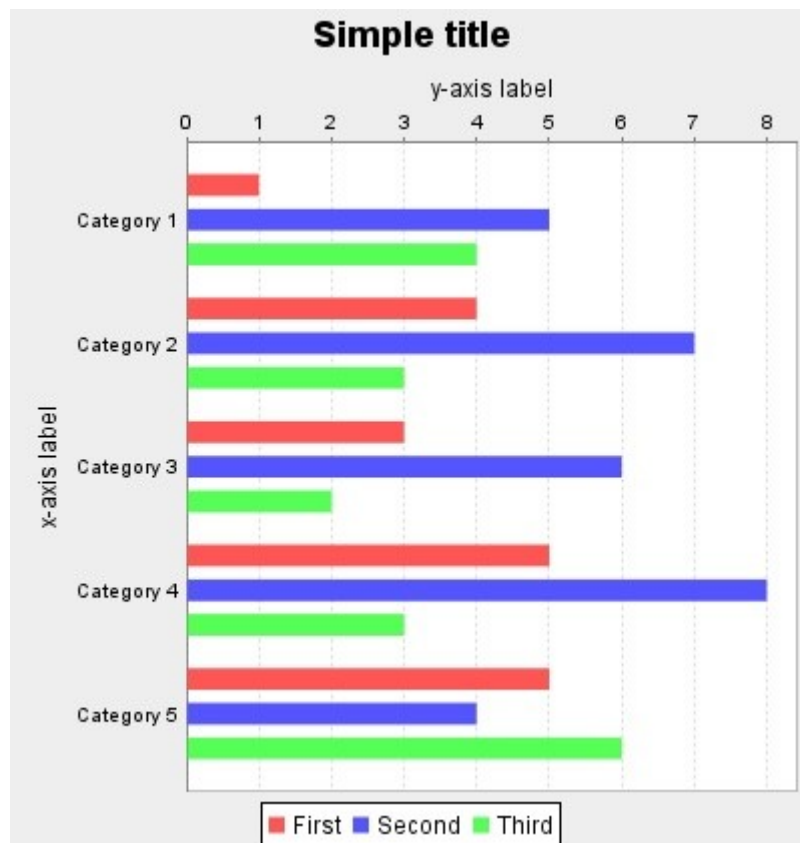
Στην επόμενη εικόνα φαίνεται πως διαχωρίζονται οι περιοχές ενός γραφήματος:



Εικόνα 24 - Διαχωρισμός των περιοχών ενός γραφήματος

4.3.6.1 Category Plot

Σε ένα category plot, το dataset μπορεί να έχει μια από τις δύο πιθανές κατευθύνσεις ως προς τους άξονες : οριζόντια ή κάθετη. Σε όλα τα παραδείγματα μέχρι τώρα η κατεύθυνση του dataset ήταν κάθετη. Αν αλλάξουμε την κατεύθυνση σε οριζόντια θα έχουμε το ακόλουθο γράφημα:



Εικόνα 25 - Οριζόντια κατεύθυνση του Plot

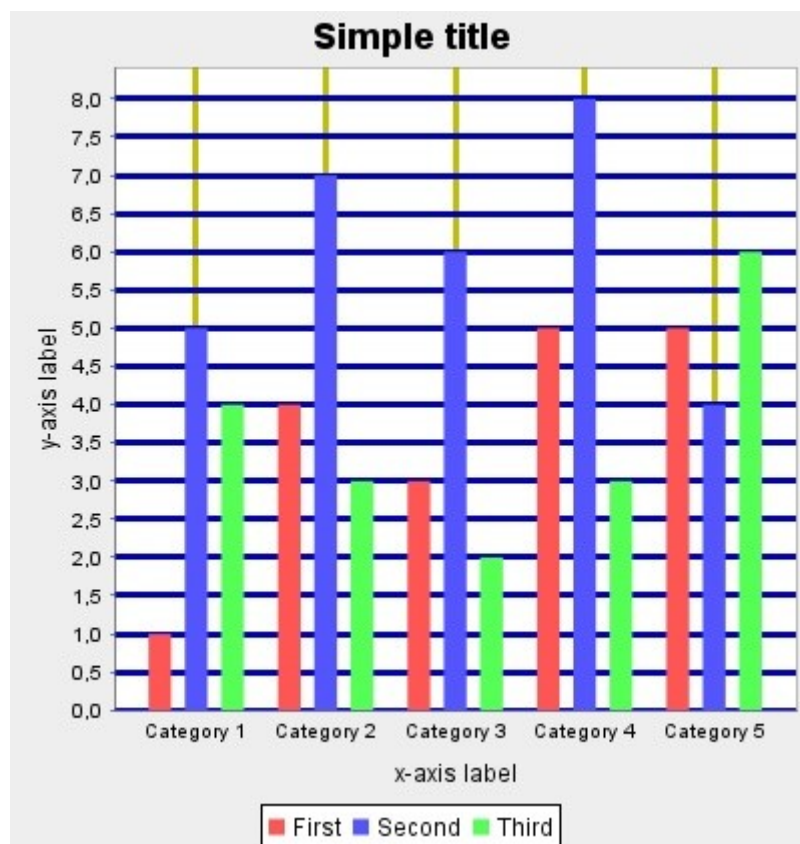
Σε ένα plot με περισσότερα από ένα datasets, είναι εφικτό να οριστεί η σειρά με την οποία θα εισάγονται τα datasets. Οι δύο πιθανές επιλογές είναι: «με τη σειρά» (forward), δηλαδή, το πρώτο dataset απεικονίζεται πρώτο, στη συνέχεια το δεύτερο κοκ. και «ανάποδα» (reverse), δηλαδή το τελευταίο dataset απεικονίζεται πρώτο κοκ. Το ίδιο ισχύει και για τη σειρά με την οποία θα απεικονίζονται στο γράφημα και οι τιμές του άξονα-x και του άξονα-y.

Σε ένα plot, όταν δημιουργείται ένα νέο γράφημα, είναι ορατές οι γραμμές πλέγματος (grid lines), ως προς τον άξονα-y και μόνο. Αυτό είναι εμφανές και στην προηγούμενη εικόνα. Ωστόσο, το σύστημα μας παρέχει την επιλογή στον χρήστη για το αν θα είναι εμφανείς ή όχι αυτές οι γραμμές πλέγματος και αν είναι, επιτρέπει την αλλαγή του χρώματος των γραμμών καθώς και την προσθήκη stroke πάνω στις γραμμές.

Επιπλέον, επιτρέπει την προσθήκη γραμμών πλέγματος και ως προς τον άξονα-x. Οι παράμετροι που μπορούν να καθοριστούν είναι ίδιοι με αυτές των γραμμών ως προς τον άξονα-y.

Κεφάλαιο 4 – Παρουσίαση των επιλογών παραμετροποίησης ενός γραφήματος

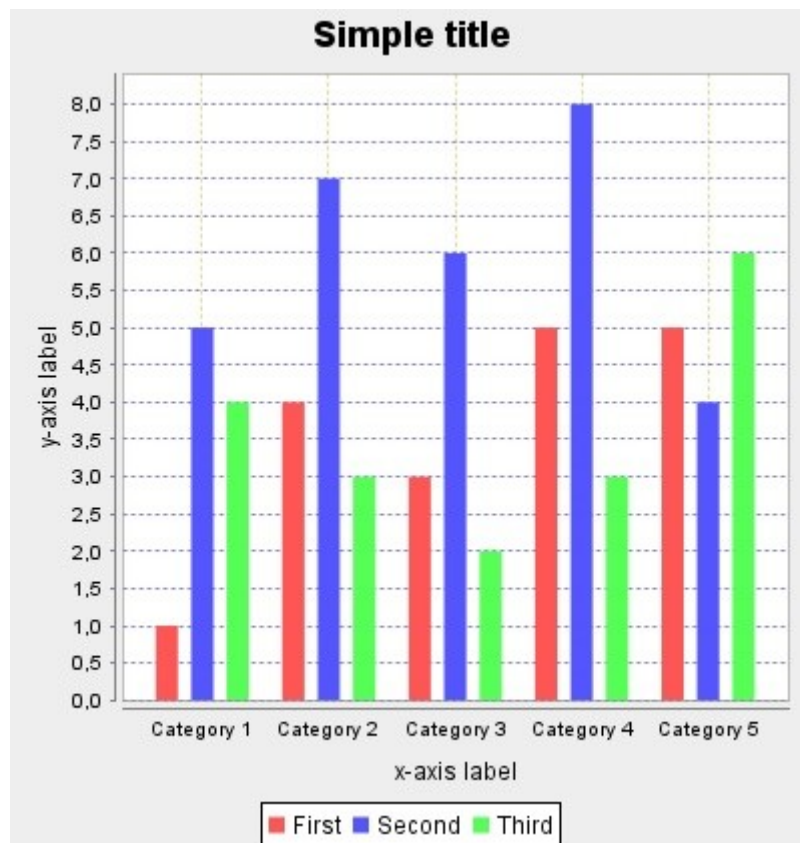
Στην επόμενη εικόνα, προσθέτουμε γραμμές πλέγματος και ως προς τον άξονα-y. Με μπλε χρώμα είναι οι γραμμές πλέγματος για τον άξονα-y και με κίτρινο για τον άξονα-x. έχουμε προσθέσει stroke στις γραμμές και έχουμε:



Εικόνα 26 - Προσθήκη γραμμών πλέγματος στο γράφημα

Το καθορισμένο χρώμα για το plot είναι το άσπρο. Ωστόσο, αυτό μπορεί να αλλάξει και να πάρει μια από τις τιμές του Πίνακα 16. Επίσης μπορούμε να προσθέσουμε και χρώμα στην περιοχή γύρω από το plot.

Τέλος, μπορούμε να ορίσουμε και το σχετικό offset, που θέλουμε να έχει το plot. Αν ορίσουμε offset ίσο με 4.0, θα έχουμε:

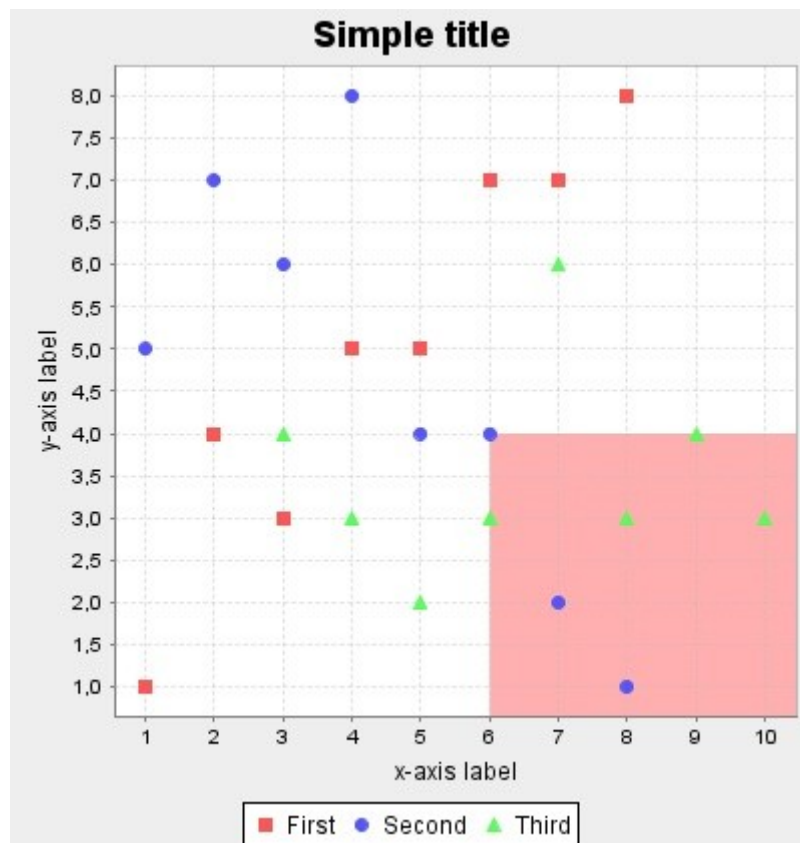


Εικόνα 27 - Προσθήκη offset στο plot

4.3.6.2 XY Plot

Το XY Plot είναι μια επέκταση του Category plot, που παρουσιάστηκε παραπάνω. Έχει τις ίδιες παραμέτρους με το Category plot, και η διαφορά του έγκειται στο ότι σχετίζεται με διαφορετικούς τύπους γραφημάτων, κυρίως με γραφήματα που απεικονίζουν τα δεδομένα ως σημεία πάνω στους άξονες. Επιπλέον από τις παραμέτρους που παρουσιάσαμε παραπάνω, σε ένα XY plot, μπορούμε να προσθέσουμε και τεταρτημόριο πάνω στους άξονες. Ολόκληρο το plot, χωρίζεται σε τέσσερα τεταρτημόρια. Έτσι, είναι δυνατόν να προσθέσουμε ένα από αυτά τα τεταρτημόρια, αν θέλουμε να δώσουμε έμφαση σε μια συγκεκριμένη περιοχή δεδομένων.

Στην επόμενη εικόνα, αλλάζουμε το γράφημα από bar που χρησιμοποιήσαμε στα παραδείγματα παραπάνω, σε scatter για να προσθέσουμε ένα τεταρτημόριο στο plot. Ορίζουμε, σε ποια από τις τέσσερις περιοχές θέλουμε να προσθέσουμε το τεταρτημόριο καθώς και το σημείο (x,y) πάνω στο plot που θέλουμε να το προσθέσουμε και έχουμε:

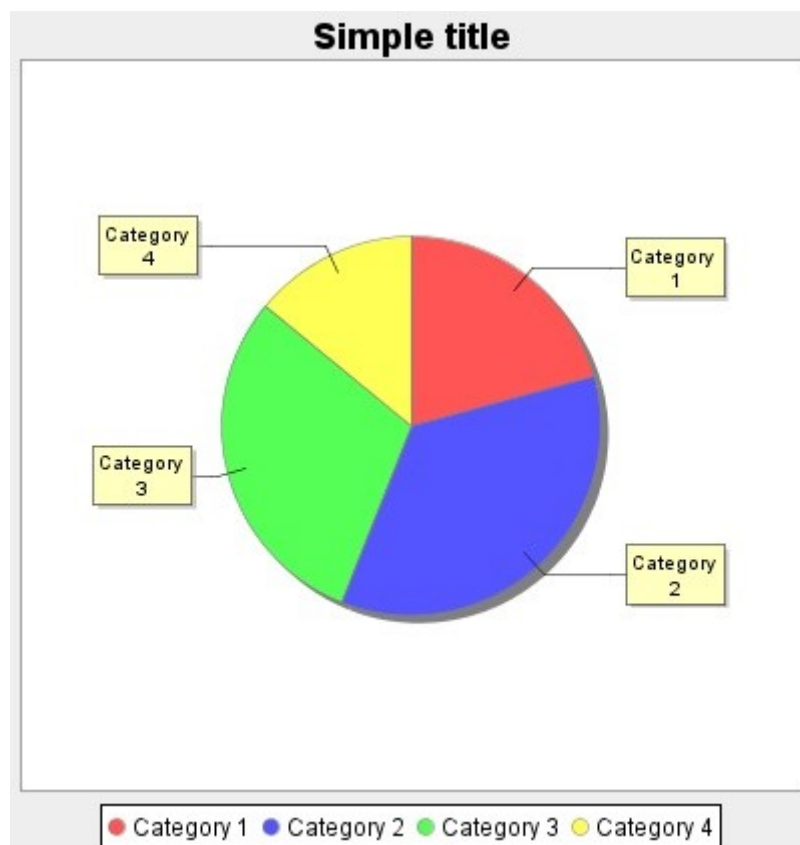


Εικόνα 28 - Προσθήκη τεταρτημορίου στο plot

Το καθορισμένο χρώμα για το τεταρτημόριο είναι το λευκό. Ωστόσο αυτό μπορεί να αλλάξει και να πάρει μια τιμή από τον Πίνακα 16. Στην προηγούμενη εικόνα το χρώμα είναι pink.

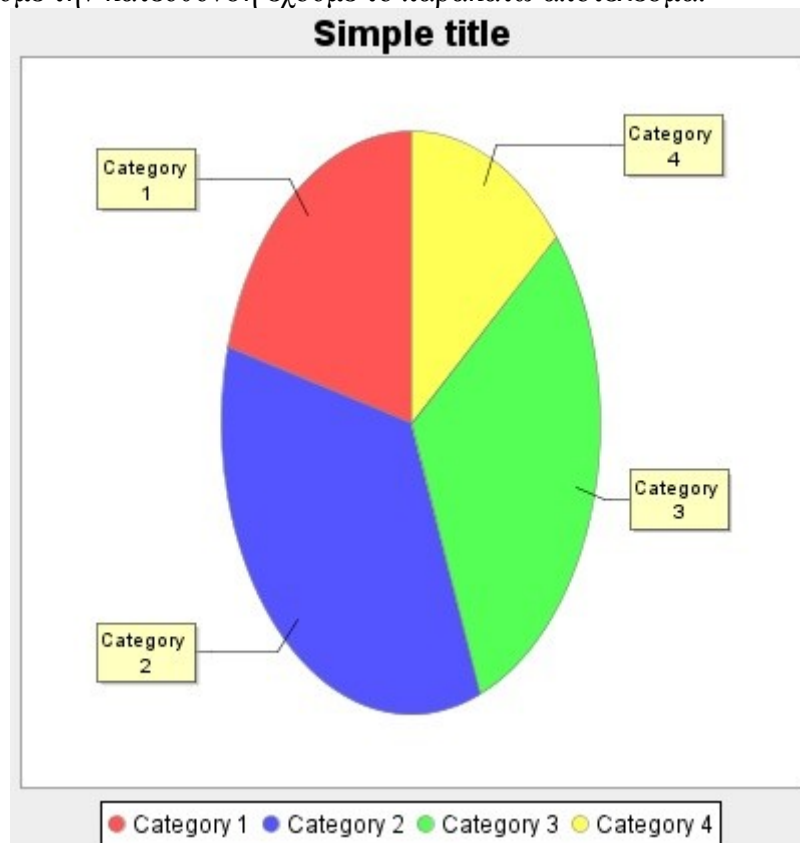
4.3.6.3 Pie Plot

Το Pie plot, αναφέρεται σε γραφήματα τύπου πίτας. Και σε αυτή την περίπτωση μπορούμε να αλλάξουμε το χρώμα του plot, όπως και προηγούμενα, το χρώμα γύρω από το plot και τη κατεύθυνση των δεδομένων. Η καθορισμένη κατεύθυνση είναι σύμφωνα με τη φορά των δεικτών του ρολογιού, όπως φαίνεται και στην επόμενη εικόνα:



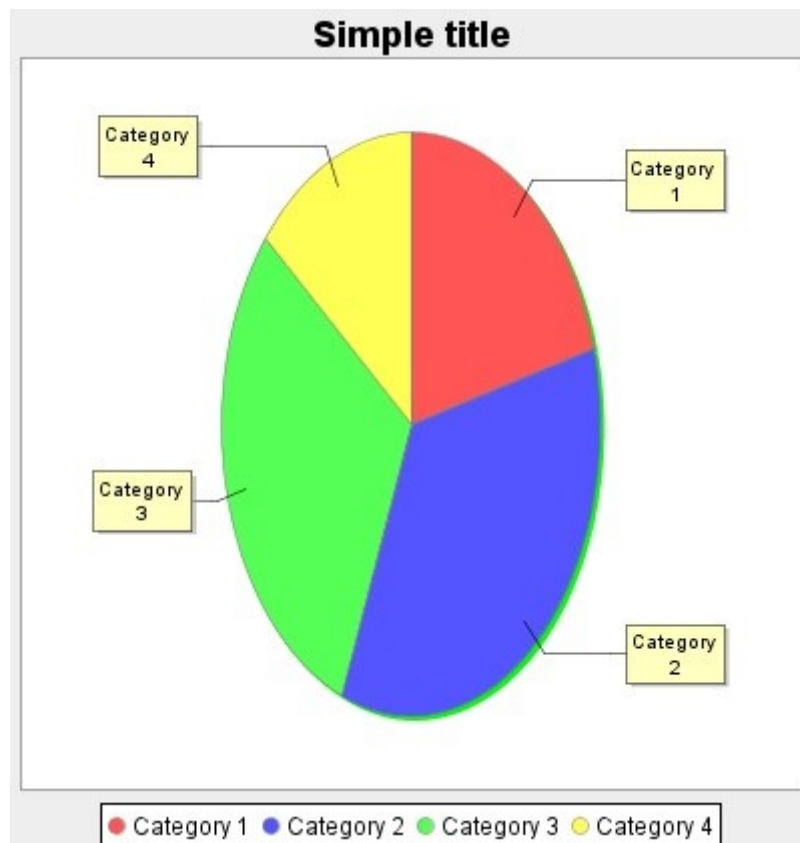
Εικόνα 29 - Καθορισμένη κατεύθυνση πίτας

Αν αλλάξουμε την κατεύθυνση έχουμε το παρακάτω αποτέλεσμα:



Εικόνα 30 - Αλλαγή κατεύθυνση πίτας

Σε αυτή την κατηγορία plot, ο χρήστης έχει την δυνατότητα να προσθέσει και σκιά (shadow) γύρω από ένα pie plot και να καθορίσουμε το offset που θα έχει από το plot. Στην επόμενη εικόνα προσθέτουμε shadow στο plot, σε χρώμα πράσινο και το offset από το plot είναι 2.0:



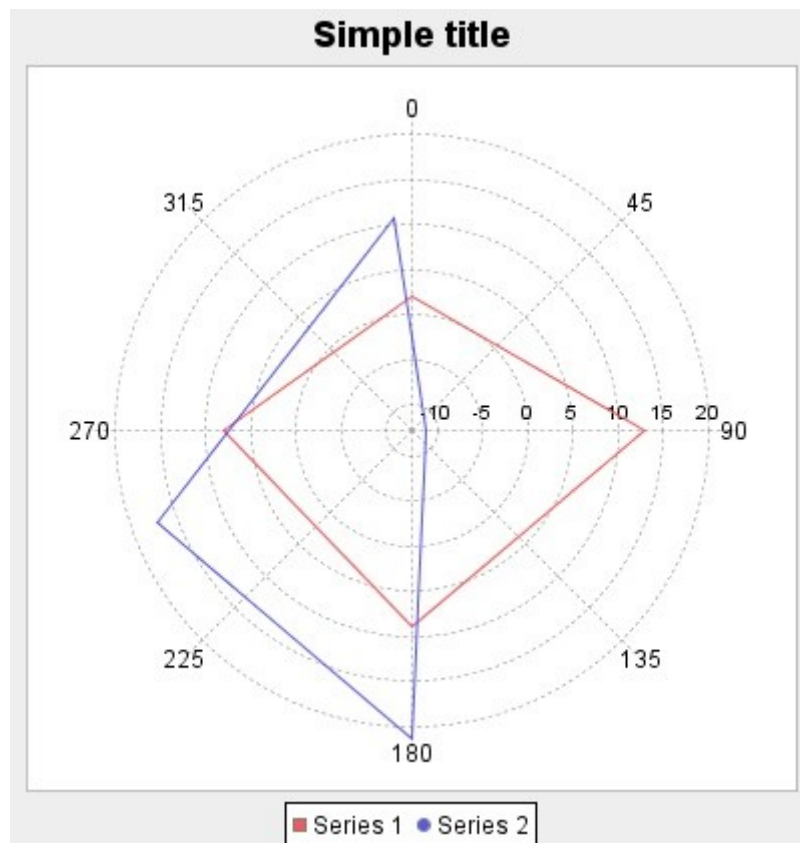
Εικόνα 31 - Προσθήκη σκιάς γύρω από τη πίτα

Τέλος, είναι δυνατόν να αλλάξουμε και το χρώμα από κάθε τομέα (κομμάτι) πάνω στην πίτα.

Τις ίδιες ιδιότητες με το pie plot έχει και το ring plot. Η διαφορά τους έγκειται στον τύπο γραφήματος που υποστηρίζουν. Ουσιαστικά ένα ring γράφημα είναι μια πίτα με μια «τρύπα» στη μέση.

4.3.6.4 Polar Plot

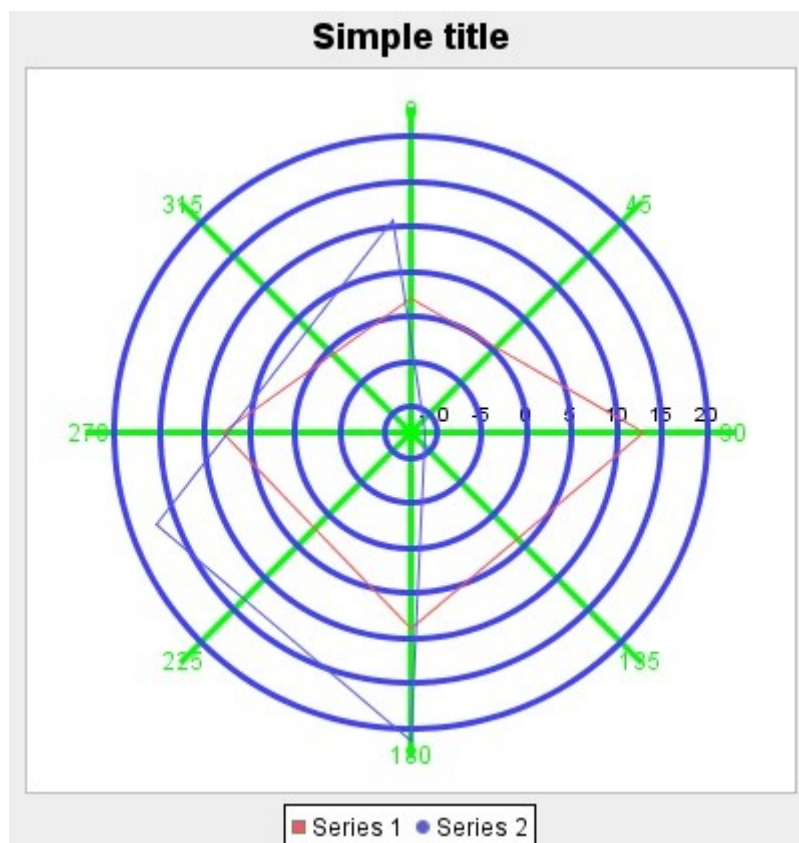
Ένα polar γράφημα, φαίνεται στην επόμενη εικόνα:



Εικόνα 32 - Polar γράφημα

Και σε αυτή την περίπτωση, όπως αναφέραμε παραπάνω, ένα polar γράφημα όταν δημιουργείται, έχει εμφανείς τις γραμμές πλέγματος(grid lines). Και εδώ, μπορούμε να επιλέξουμε αν θα εμφανίζονται ή όχι οι γραμμές και αν ναι, να αλλάξουμε το χρώμα των γραμμών και να προσθέσουμε stroke ή όχι.

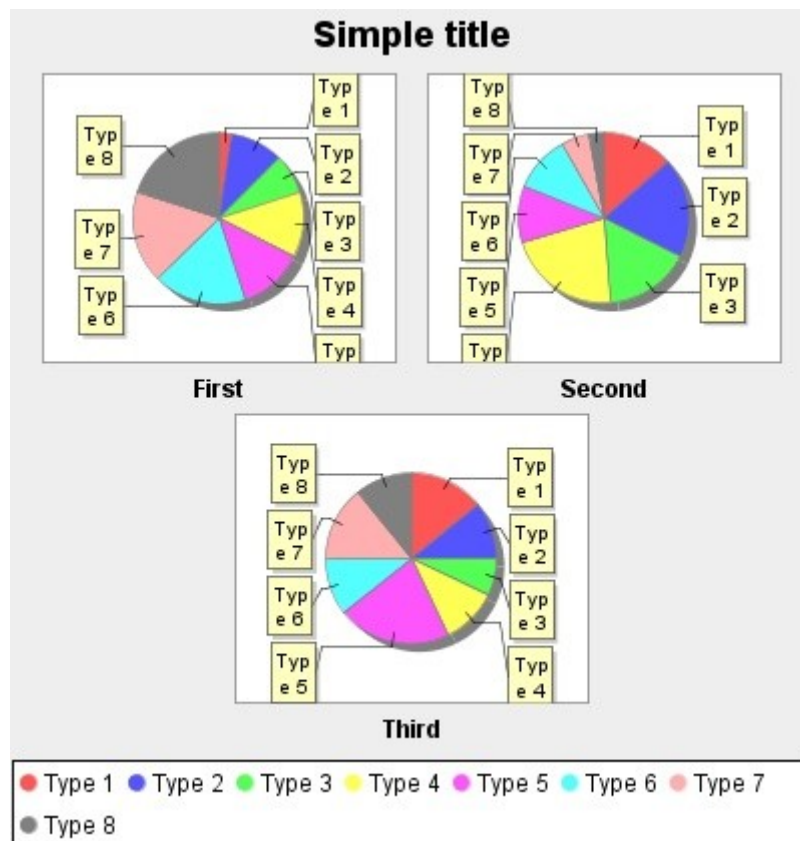
Στην παρακάτω εικόνα, αλλάζουμε το χρώμα των grid lines σε μπλε. Προσθέτουμε angle στο plot, σε χρώμα πράσινο και στο ίδιο χρώμα αλλάζουμε και τα labels:



Εικόνα 33 - Επεξεργασία των grid lines

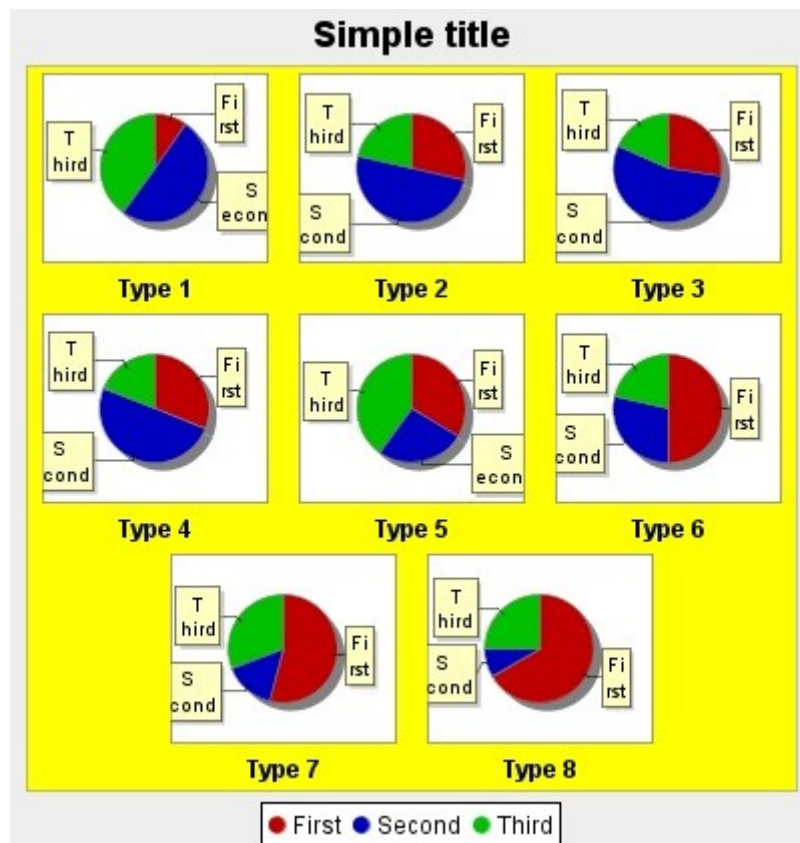
4.3.6.4 Multiple Pie Plot

Ένα απλό γράφημα multiple pie, φαίνεται στην παρακάτω εικόνα:



Εικόνα 34 - Multiple Pie γράφημα

Και σε αυτή την περίπτωση, μπορούμε να αλλάξουμε το χρώμα του plot, το χρώμα γύρω από το plot και να καθορίσουμε την σειρά εισαγωγής του dataset. Οι δυνατές επιλογές είναι είτε με βάση τις στήλες (by columns), είτε με βάση τις γραμμές (by rows). Στην προηγούμενη εικόνα, η σειρά εισαγωγής του dataset είναι by rows. Στην επόμενη εικόνα, αλλάζουμε το χρώμα του plot σε κίτρινο και την σειρά εισαγωγής του dataset, by columns:



Εικόνα 35 - Επεξεργασία του plot

4.3.7 Μορφοποίηση των αξόνων του γραφήματος

Η εφαρμογή που σχεδιάσαμε επιτρέπει την μορφοποίηση των αξόνων, μετά τη δημιουργία ενός γραφήματος. Η μορφοποίηση των αξόνων δεν υποστηρίζεται από όλους τους τύπους γραφημάτων και αυτό γιατί δεν απεικονίζονται όλα τα γραφήματα πάνω σε άξονες. Για παράδειγμα, ένα γράφημα τύπου Pie.

Υπάρχουν δύο κατηγορίες αξόνων: Category Axes και Value Axes

4.3.7.1 Category Axes

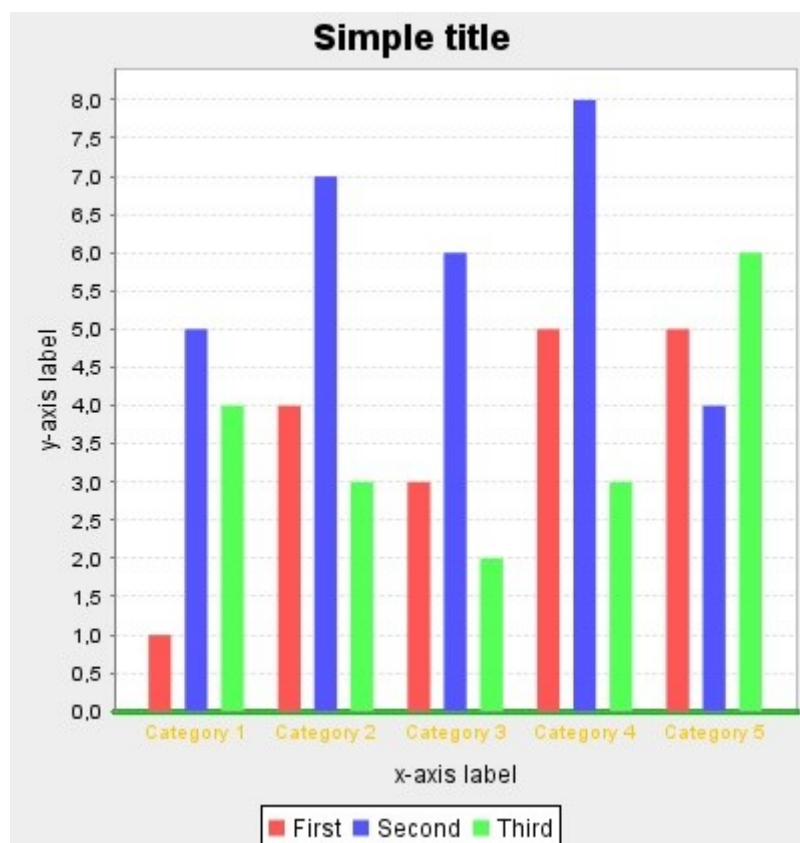
Με τον όρο Category αναφερόμαστε ουσιαστικά στον άξονα-x, και η δυνατότητα μορφοποίησης παρέχεται από το σύστημα για γραφήματα που απεικονίζουν τα δεδομένα σε μορφή «κατηγοριών». Υποστηρίζεται από ορισμένους τύπους γραφημάτων και αυτοί είναι:

Area, Bar, Line, Line3D, Stacked Bar, Interval Bar, Layered Bar, Statistic Bar, BoxAndWhisker, Gantt, Multiple Line, Multiple Line3D

4.3.7.1.1 Επεξεργασία των χρωμάτων

Οι αλλαγές που μπορούν να γίνουν σε αυτή την περίπτωση αφορούν στην γραμμή του άξονα και στις τιμές που εμφανίζονται πάνω στον άξονα.

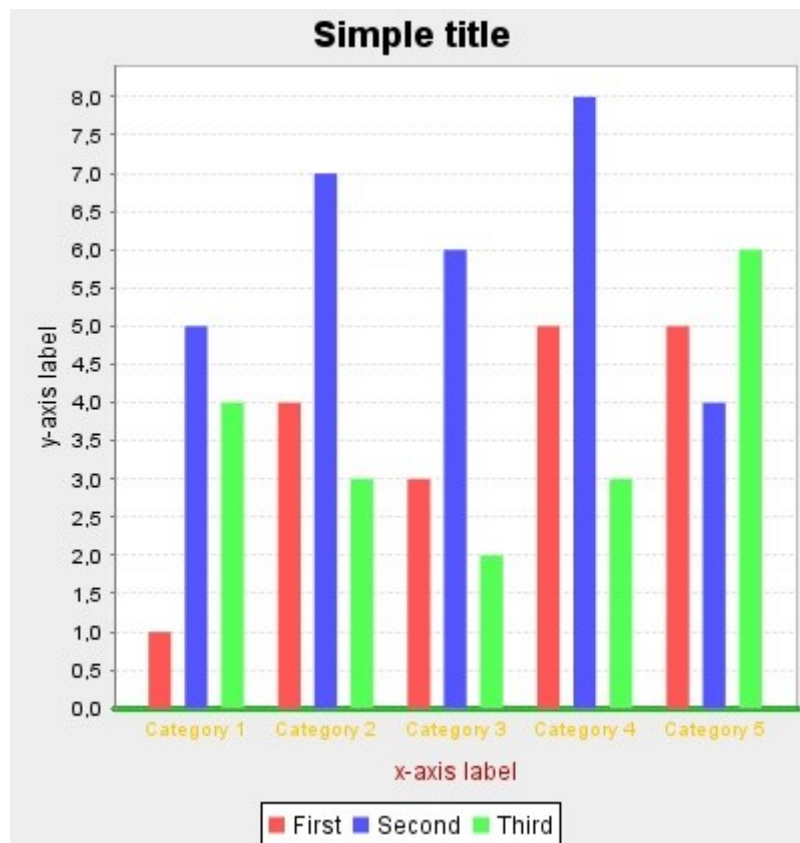
Στην επόμενη εικόνα αλλάζουμε το χρώμα της γραμμής του άξονα σε πράσινο και το χρώμα των τιμών που εμφανίζονται πάνω στον άξονα σε πορτοκαλί:



Εικόνα 36 - Επεξεργασία του άξονα x

Επίσης μπορούμε να προσθέσουμε και stroke πάνω στην γραμμή του άξονα για να είναι πιο έντονη.

Σε αυτή την κατηγορία αξόνων μπορούμε να αλλάξουμε και το χρώμα του label των αξόνων που προσθέσαμε κατά τη δημιουργία του γραφήματος. Στην επόμενη εικόνα αλλάζουμε το καθορισμένο χρώμα σε κόκκινο και έχουμε:

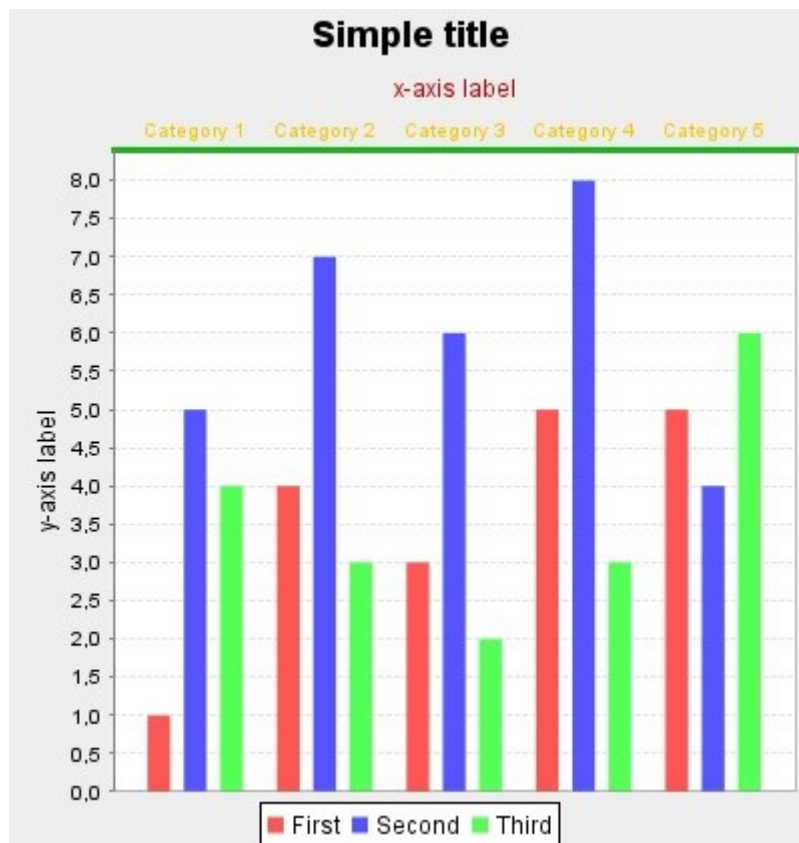


Εικόνα 37 - Αλλαγή του χρώματος της ετικέτας του άξονα x

4.2.8.1.2 Επεξεργασία της θέσης του άξονα

Η καθορισμένη θέση για τον άξονα είναι να βρίσκεται κάτω και αριστερά στο γράφημα. Ωστόσο, η θέση αυτή μπορεί να αλλάξει και οι υπόλοιπες δυνατές τιμές είναι: Bottom_Right, Top_Left, Top_Right.

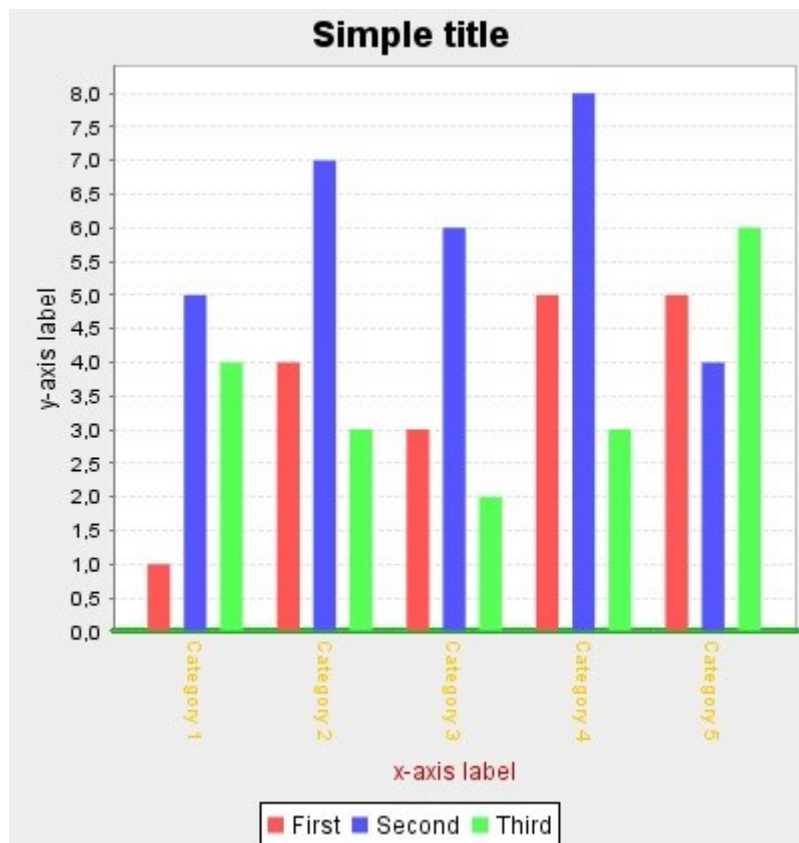
Στην επόμενη εικόνα αλλάζουμε τη θέση των αξόνων σε Top_Left και έχουμε:



Εικόνα 38 - Αλλαγή της θέσης του άξονα x

Εκτός από την θέση του άξονα, είναι δυνατή και η αλλαγή της θέσης των τιμών του άξονα. Η καθορισμένη θέση να εμφανίζονται είναι οριζόντια. Οι υπόλοιπες δυνατές θέσεις είναι: down_90, down_45, up_90, up_45.

Στην επόμενη εικόνα αλλάζουμε τη θέση των τιμών του άξονα σε down_90 και έχουμε:



Εικόνα 39 - Αλλαγή της θέσης των labels του άξονα x

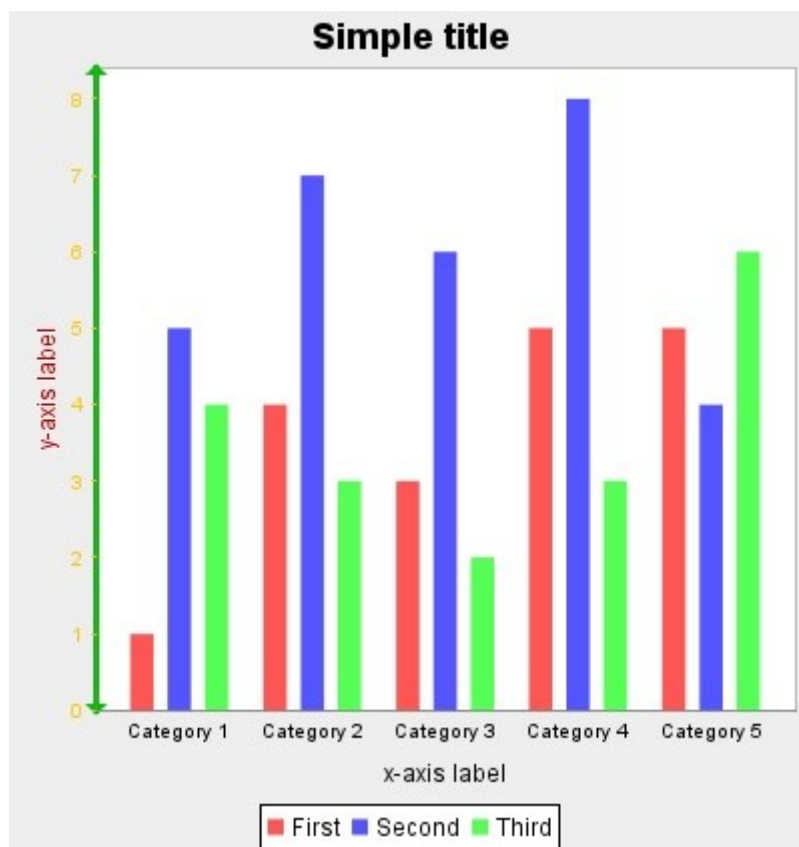
Τέλος, όπως αναφέραμε παραπάνω ο τύπος αυτός άξονα σχετίζεται με γραφήματα που απεικονίζουν τα δεδομένα τους σε «κατηγορίες». Το JFreeChart επιτρέπει στον εκάστοτε χρήστη να ορίσει ένα κενό ανάμεσα στις κατηγορίες που θα εμφανίζονται, καθώς και ένα ελάχιστο κενό ανάμεσα στην πρώτη κατηγορία και στους άξονες και το ίδιο για την τελευταία κατηγορία.

4.3.7.2 Value Axes

Αντίστοιχα με την προηγούμενη περίπτωση, με τον όρο value αναφερόμαστε στον άξονα που εμφανίζει το εύρος των τιμών, δηλαδή στον άξονα-y.

Σε γενικές γραμμές, οι παράμετροι που μπορούν να καθοριστούν και σε αυτή την περίπτωση είναι ίδιοι με παραπάνω, απλά υπάρχουν και κάποιες διαφοροποιήσεις που παρουσιάζονται στη συνέχεια.

Σε αυτή την περίπτωση του άξονα-y, μπορούμε να προσθέσουμε και βέλος προς μία ή/και στις δυο κατευθύνσεις του άξονα. Στην επόμενη εικόνα προσθέτουμε προς και τις δυο κατευθύνσεις και έχουμε:



Εικόνα 40 - Προσθήκη arrow στον άξονα y

4.3.7.2.1 Μορφοποίηση των τιμών του άξονα

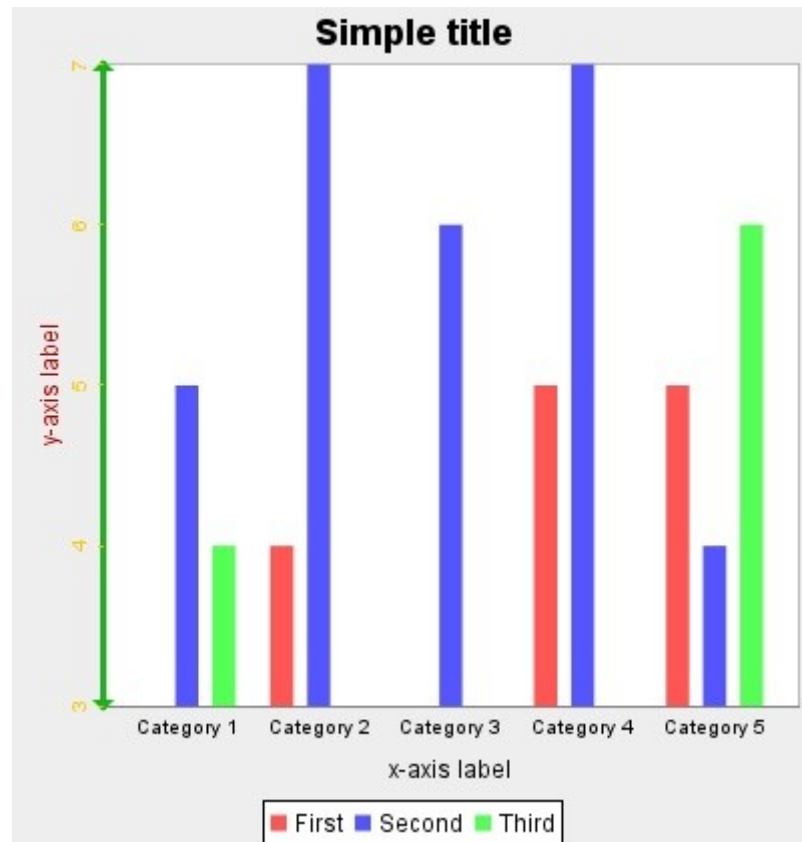
Επειδή αναφερόμαστε στον άξονα-y, υπάρχουν και κάποιες επιλογές που αφορούν τον τρόπο με τον οποίο θα εμφανίζονται οι τιμές των δεδομένων.

Τέτοιες επιλογές είναι αν για την απεικόνιση δεδομένων, ληφθούν υπ' όψιν και οι μηδενικές τιμές που ενδεχομένως να υπάρχουν, αν το εύρος των τιμών που θα εμφανίζεται στον άξονα θα ταιριάζει απόλυτα με τα δεδομένα. Αυτές οι επιλογές είτε είναι ενεργοποιημένες είτε όχι.

Επίσης υπάρχει η δυνατότητα να δημιουργήσουμε ουσιαστικά ένα νέο εύρος τιμών για τα δεδομένα. Για να το πετύχουμε αυτό, ορίζουμε τη νέα τιμή γύρω από την οποία θέλουμε να είναι συγκεντρωμένα τα δεδομένα και ορίζουμε μια νέα τιμή για το εύρος γύρω από αυτή τη τιμή.

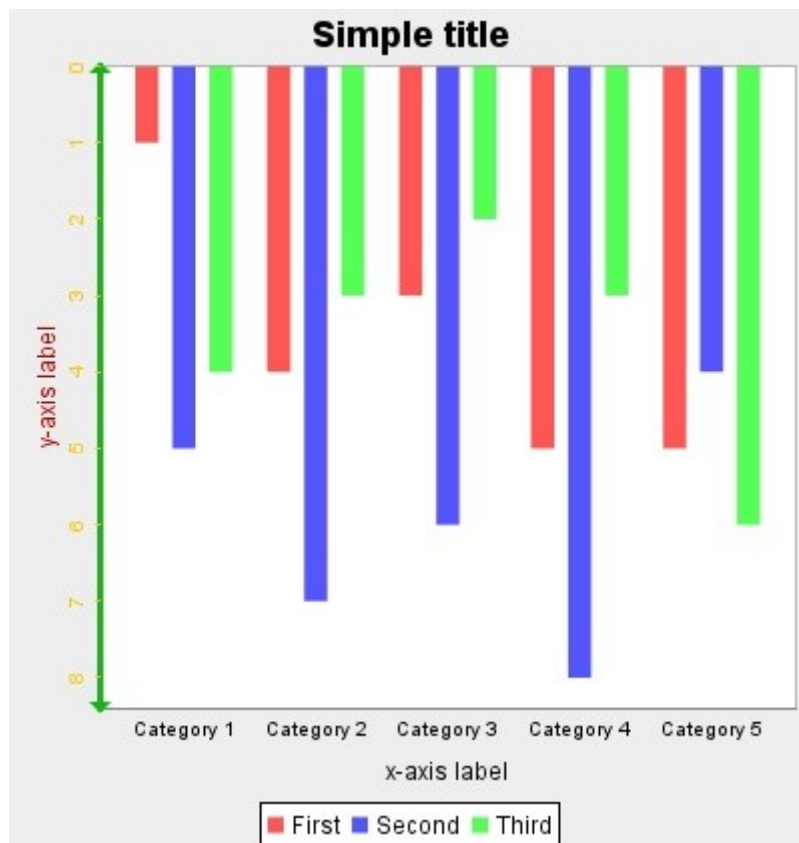
Κεφάλαιο 4 – Παρουσίαση των επιλογών παραμετροποίησης ενός γραφήματος

Στην επόμενη εικόνα, ορίζουμε τα δεδομένα να είναι συγκεντρωμένα γύρω από τη τιμή 5, και το εύρος το ορίζουμε στο 4:



Εικόνα 41 - Αλλαγή του εύρους τιμών του άξονα y

Τέλος, μπορούμε να αντιστρέψουμε και τη φορά με την οποία θα εμφανίζονται οι τιμές και θα έχουμε:



Εικόνα 42 - Αντίστροφη φορά εμφάνισης τιμών του άξονα y

4.3.8 Προσθήκη και μορφοποίηση annotation στο γράφημα

Τα annotations είναι από τα προαιρετικά στοιχεία που μπορούν να προστεθούν μετά τη δημιουργία ενός γραφήματος. Χρησιμοποιούνται κυρίως για το σχολιασμό κάποιων συγκεκριμένων τιμών δεδομένων και προστίθενται στην περιοχή απεικόνισης δεδομένων. Μπορούν να προστεθούν σε ορισμένους μόνο τύπους γραφημάτων και αυτοί είναι:

Area, Bar, Line, Line3D, Stacked Bar, Interval Bar, Layered Bar, Statistical Bar, BoxAndWhisker, Gantt, Multiple Line, MultipleLine3D, Histogram, Scatter, XY Bar, XY Line, XY Step Area, XY Step, Time Series, XY Area, Stacked XY Area

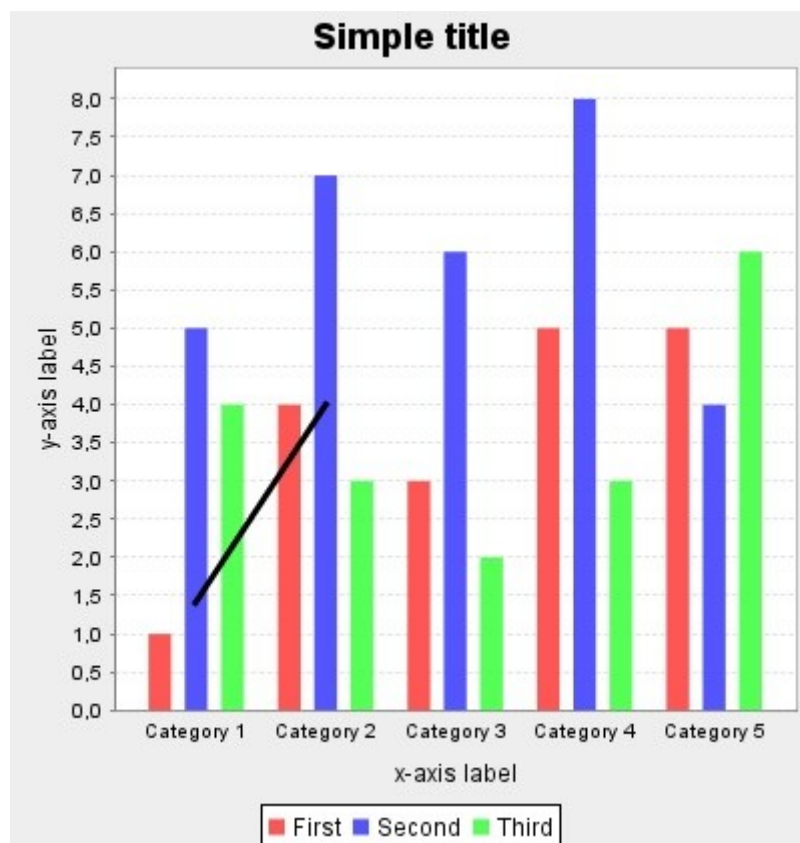
Υπάρχουν τέσσερις διαφορετικοί τύποι annotation που μπορούν να προστεθούν σε ένα γράφημα: line, text, pointer, box.

Να σημειώσουμε εδώ ότι στο ίδιο γράφημα μπορούμε να προσθέσουμε διαφόρους τύπους annotation αλλά από μια φορά το καθένα.

4.3.8.1 Line Annotation

Για να προσθέσουμε ένα annotation τύπου line, χρειάζεται να ορίσουμε το σημείο που θα αρχίζει η γραμμή και το σημείο που θα τελειώνει. Επίσης, το JFreeChart επιτρέπει την αλλαγή του καθορισμένου χρώματος για τη γραμμή.

Στην παρακάτω εικόνα προσθέτουμε ένα Line annotation που θα ξεκινάει από το Category 1, σε ύψος 1.4 και θα τερματίζει στο Category 2 σε ύψος 4.0.

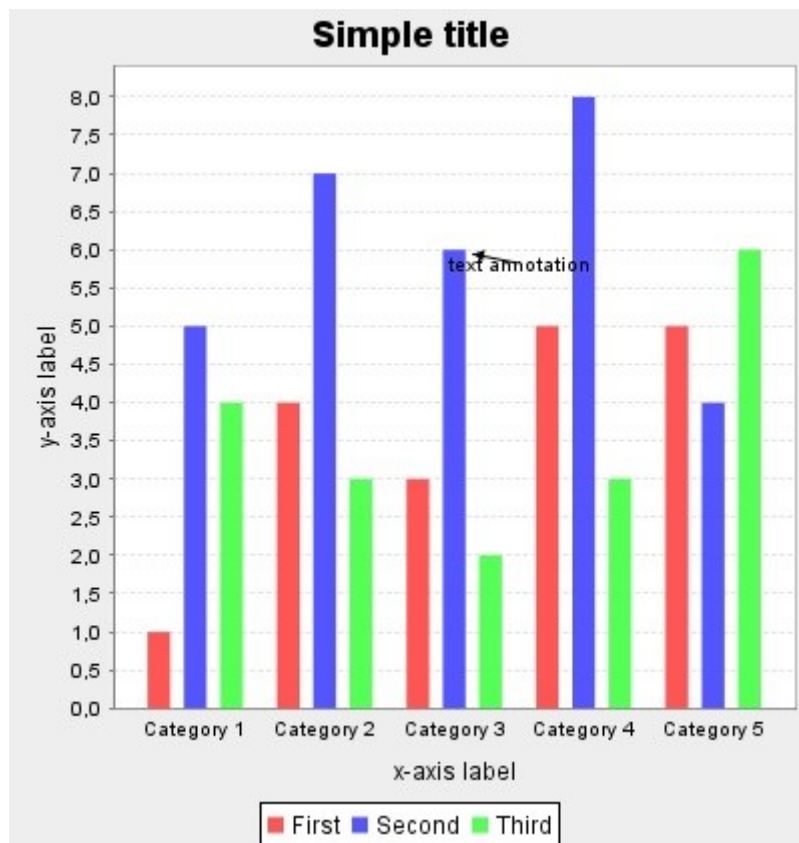


Εικόνα 43 - Line annotation

4.3.8.2 Pointer Annotation

Για να προσθέσουμε ένα annotation τύπου pointer, χρειάζεται να ορίσουμε το σημείο που θέλουμε να προστεθεί, δηλαδή την τιμή του άξονα-x και σε ποιο ύψος στον άξονα - y , το κείμενο που θέλουμε να εμφανιστεί και τέλος την κλίση που θα έχει το βέλος.

Στην επόμενη εικόνα προσθέτουμε ένα pointer annotation στην τιμή του άξονα-x, *Category 3*, και σε ύψος 6.0. Το κείμενο που θέλουμε να εμφανίζεται είναι το *text annotation* και η κλίση του βέλους είναι στο 0.1:

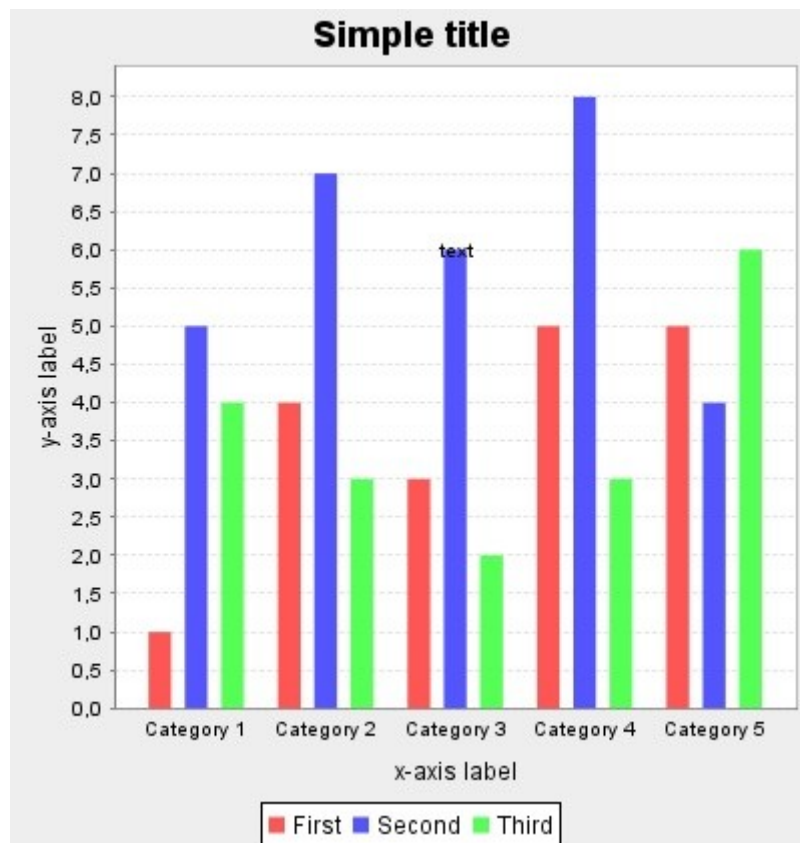


Εικόνα 44 - Pointer Annotation

4.3.8.3 Text annotation

Για να προσθέσουμε ένα text annotation ορίζουμε απλά τη τιμή του άξονα-x και το ύψος στον άξονα-y που θέλουμε να εμφανιστεί, και το κείμενο που θέλουμε να φαίνεται.

Στην επόμενη εικόνα προσθέτουμε στην τιμή του άξονα-x , *Category 3*, στο ύψος 6.0, το κείμενο *text* και έχουμε:



Εικόνα 45 - Text Annotation

4.3.9 Προσθήκη και μορφοποίηση marker στο γράφημα

Markers μπορούν να προστεθούν στην περιοχή απεικόνισης δεδομένων με σκοπό να δώσουν έμφαση σε συγκεκριμένες τιμές πάνω στα δεδομένα.

Η προσθήκη marker και κατά συνέπεια η επεξεργασία των παραμέτρων του , μπορεί να γίνει σε ορισμένους μόνο τύπους γραφημάτων και αυτοί είναι:

Area, Bar, Line, Line3D, Stacked Bar, Interval Bar, Layered Bar, Statistic Bar, BoxAndWhisker, Gantt, Multiple Line, MultipleLine3D

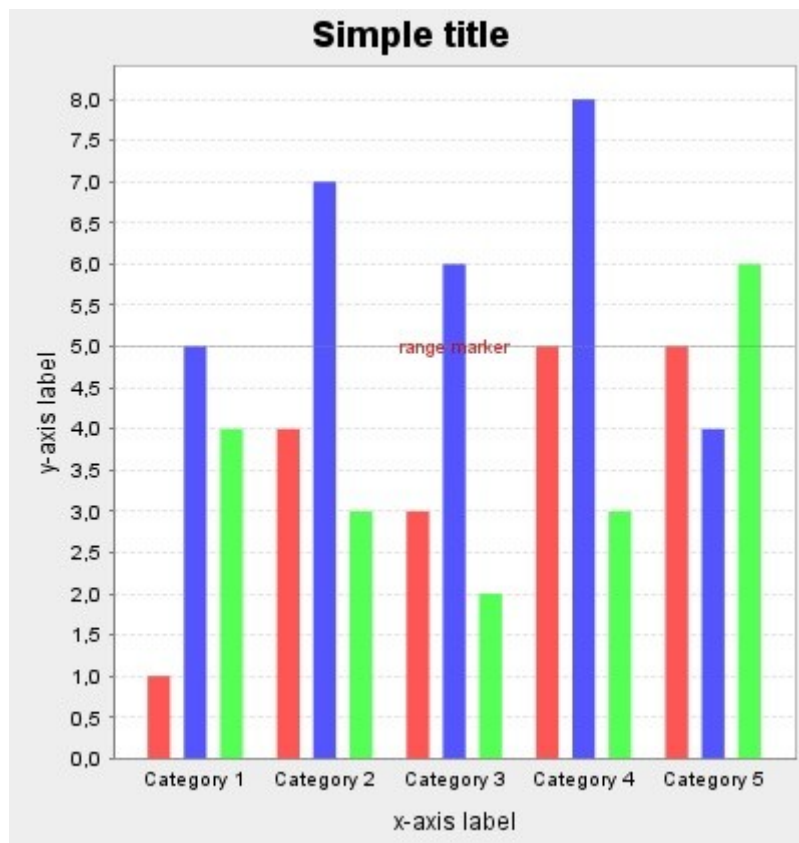
Υπάρχουν δυο κατηγορίες marker που μπορούν να προστεθούν σε ένα γράφημα και αυτές παρουσιάζονται αναλυτικά παρακάτω.

4.3.9.1 Range Marker

Marker τύπου range χρησιμοποιείται για να δώσει έμφαση σε κάποια τιμή του άξονα των τιμών (άξονας-y). Για να προσθέσουμε έναν range marker χρειάζεται απλά να ορίσουμε το σημείο του άξονα-y στο οποίο θέλουμε να εμφανίζεται. Επιπλέον η

εφαρμογή επιτρέπει καθορίσουμε το κείμενο που θα εμφανίζεται ως label του marker, το χρώμα κειμένου και τη θέση που θα εμφανίζεται.

Στην παρακάτω εικόνα, προσθέτουμε έναν range marker στη τιμή 5.0 του άξονα-y, ως label του marker εμφανίζεται το κείμενο range marker, το χρώμα του κειμένου είναι κόκκινο και η θέση του είναι στην κορυφή του marker:



Εικόνα 46 - Range Marker

Το χρώμα του κειμένου μπορεί να πάρει μια από τις τιμές που φαίνονται στον Πίνακα 16 και οι υπόλοιπες δυνατές τιμές για την θέση του είναι: Top-Left, Top-Right, Bottom, Bottom-Left, Bottom-Right, Left, Right, Center

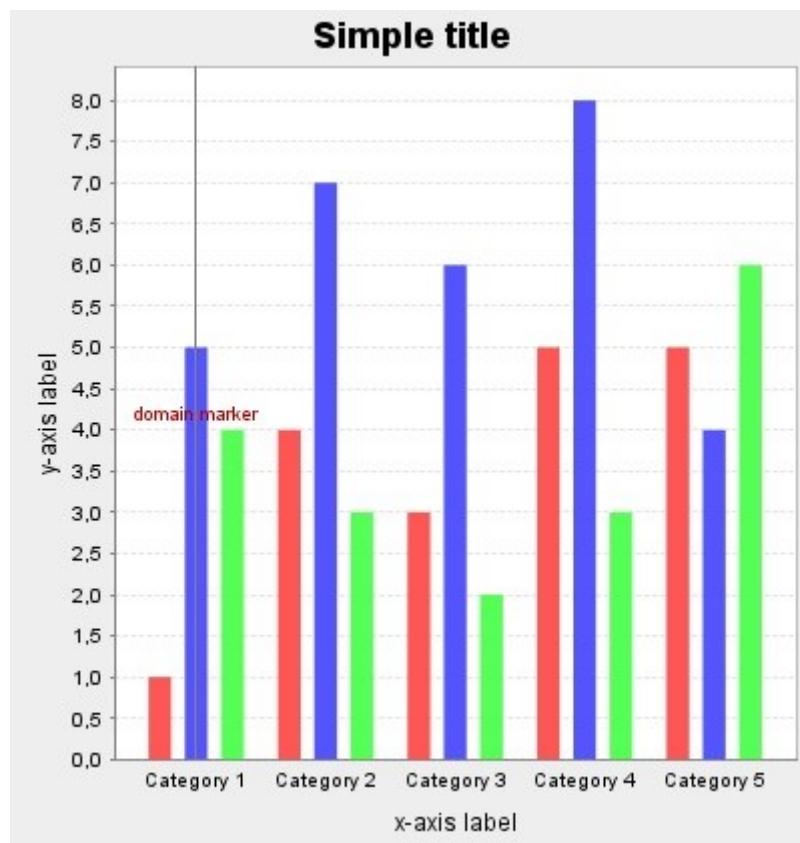
4.3.9.2 Domain Marker

Αντίστοιχα με προηγούμενα, ένας marker τύπου domain, χρησιμοποιείται για να δώσει έμφαση σε τιμές του άξονα-x. Για να προσθέσουμε έναν domain marker χρειάζεται απλά να ορίσουμε την τιμή του άξονα-x στην οποία θέλουμε να δώσουμε έμφαση. Επιπλέον η εφαρμογή επιτρέπει καθορίσουμε το κείμενο που θα εμφανίζεται ως label του marker, το χρώμα κειμένου και τη θέση που θα εμφανίζεται.

Κεφάλαιο 4 – Παρουσίαση των επιλογών παραμετροποίησης ενός γραφήματος

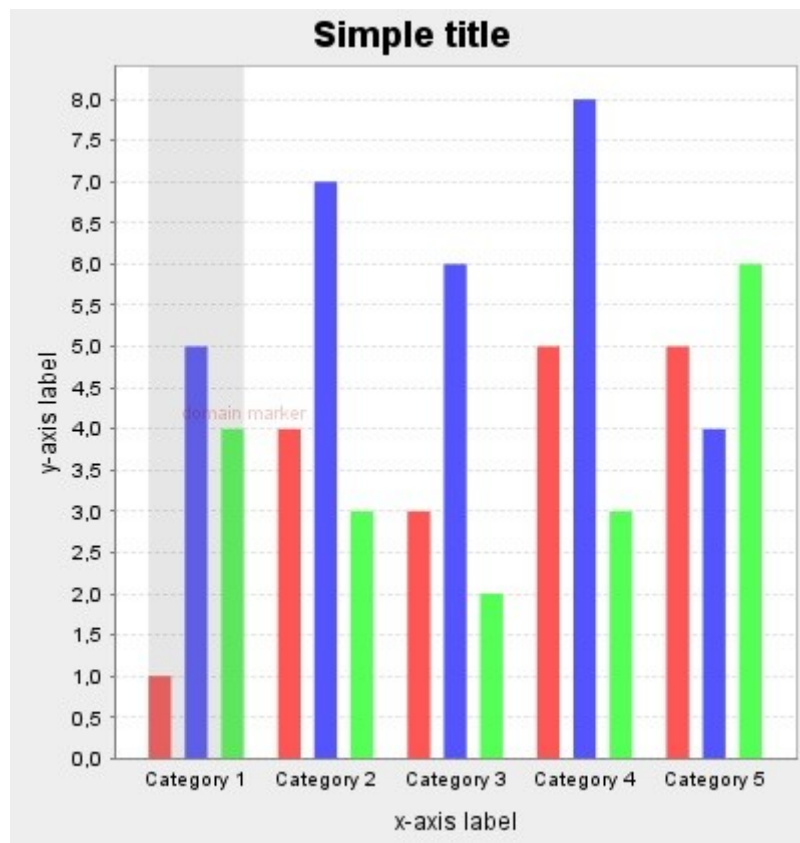
Σε αυτή την περίπτωση marker, ο χρήστης μπορεί να επιλέξει την προσθήκη marker ως μια γραμμή ή όχι. Παρακάτω ακολουθεί ένα αναλυτικό παράδειγμα .

Στην επόμενη εικόνα προσθέτουμε έναν domain marker στην τιμή του άξονα-x: *Category 1* , επιλέγουμε την προσθήκη marker ως μια γραμμή, ως label του marker εμφανίζεται το κείμενο domain marker, το χρώμα του κειμένου είναι κόκκινο και η θέση του είναι στο κέντρο του marker :



Εικόνα 47 – Domain line Marker

Αν τώρα στο ίδιο γράφημα με παραπάνω επιλέξουμε να μην προστεθεί ως μια γραμμή έχουμε:



Εικόνα 48 - Domain Marker

Το χρώμα του κειμένου μπορεί να πάρει μια από τις τιμές που φαίνονται στον Πίνακα 16 και οι υπόλοιπες δυνατές τιμές για την θέση του είναι: Top-Left, Top-Right, Bottom, Bottom-Left, Bottom-Right, Left, Right, Center.

4.4 Περίληψη

Στο κεφάλαιο αυτό παρουσιάσαμε αναλυτικά όλες τις επιλογές παραμετροποίησης ενός γραφήματος που παρέχονται από την εφαρμογή μας. Περιγράψαμε το βασικό εργαλείο που χρησιμοποιήσαμε, την βιβλιοθήκη JFreeChart και από εκεί και έπειτα αναδείξαμε με παραδείγματα όλες τις παραμέτρους που μπορεί να τροποποιήσει ο εκάστοτε χρήστης της εφαρμογής.

Κεφάλαιο 5

Παρουσίαση της γραφικής διεπαφής του συστήματος

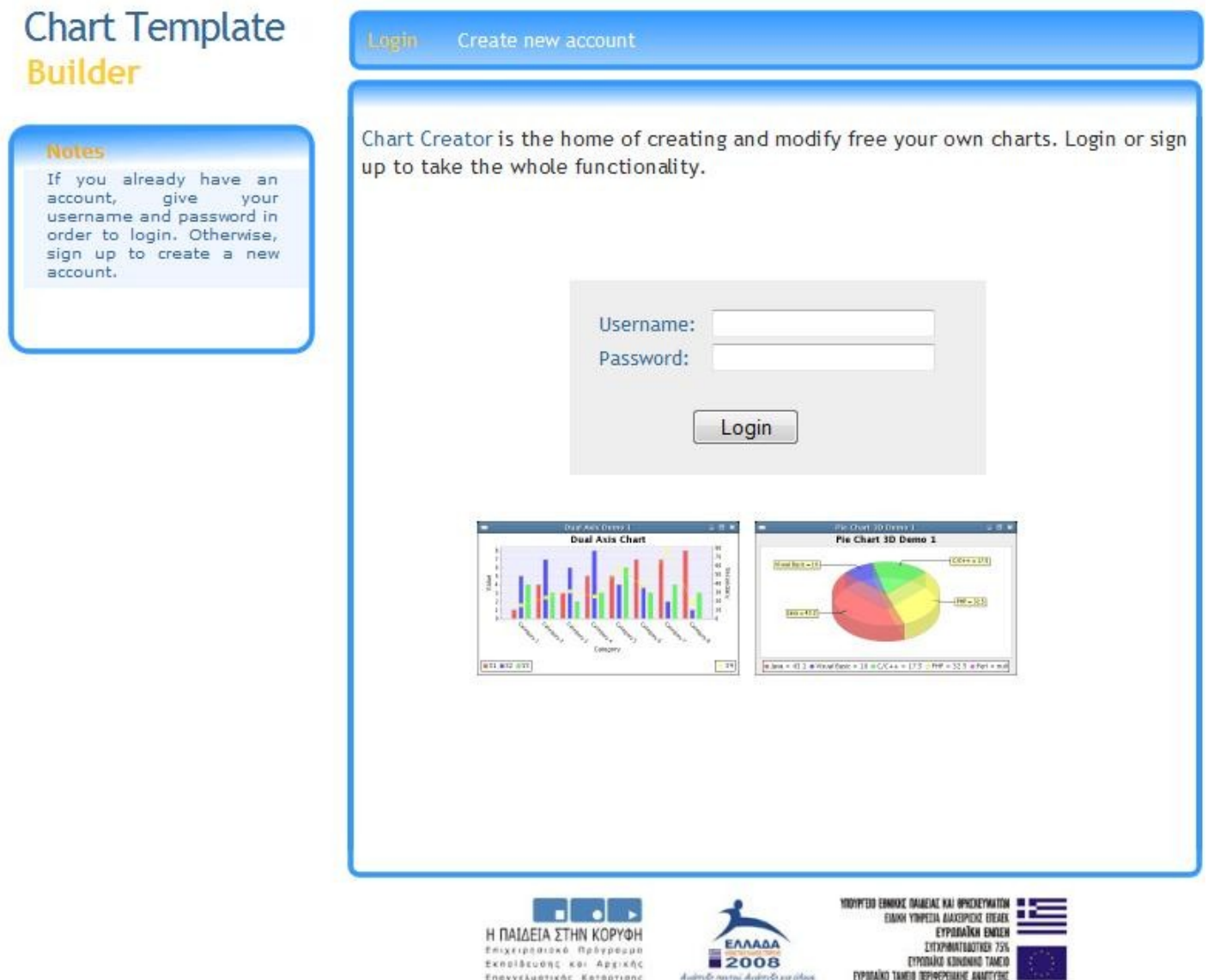
5.1 Εισαγωγή

Στην ενότητα αυτή θα παρουσιάσουμε τη γραφική διεπαφή που χρησιμοποιούν οι χρήστες για να δημιουργήσουν τα γραφήματα τους. Ο σχεδιασμός έγινε ακολουθώντας τις γενικές αρχές σχεδιασμού διεπαφών χρήστη (User Interface) που διέπουν την ευχρηστία της εφαρμογής απέναντι στον χρήστη, την άνετη πλοήγηση του στην ιστοσελίδα, την κατάλληλη επιλογή χρωμάτων κ.ά. που παρουσιάζονται αναλυτικά παρακάτω.

Στις επόμενες ενότητες παρουσιάζεται αναλυτικά η γραφική διεπαφή που σχεδιάσαμε, ξεκινώντας από την αρχική σελίδα και παρουσιάζοντας όλες τις επιλογές πλοήγησης που παρέχονται στον χρήστη του συστήματος.

5.2 Παρουσίαση της γραφικής διεπαφής

Η αρχική σελίδα της εφαρμογής, φαίνεται στην παρακάτω εικόνα. Ο χρήστης πρέπει να κάνει είσοδο στο σύστημα (login) για να έχει όλη τη λειτουργικότητα της εφαρμογής.



Εικόνα 49 – Είσοδος στο σύστημα

Τα χρώματα που χρησιμοποιήσαμε επιλέχθηκαν ώστε να μην είναι ενοχλητικά για το μάτι και χρησιμοποιούμε διαφορετικά χρώματα σε σημεία που θέλουμε να δώσουμε έμφαση (π.χ. περιπτώσεις λάθους). Όλες οι σελίδες ακολουθούν το ίδιο layout έτσι ώστε να υπάρχει συνέχεια και συνέπεια μεταξύ τους.

Κεφάλαιο 5 – Παρουσίαση του User Interface

Από τη στιγμή που ο χρήστης κάνει είσοδο στην εφαρμογή, εισέρχεται στην κεντρική σελίδα της εφαρμογής, που αποτελεί την προσωπική σελίδα κάθε χρήστη. Σε αυτή τη σελίδα κάθε χρήστης μπορεί να δει όλα τα πρότυπα γραφήματα που έχει δημιουργήσει, να τα διαχειριστεί καθώς και να δημιουργήσει καινούρια. Επιπλέον από το μενού μπορεί να πλοηγηθεί μέσα στην εφαρμογή όπως επιθυμεί. Αναλυτική περιγραφή ακολουθεί στην συνέχεια. Η κεντρική σελίδα φαίνεται στην παρακάτω εικόνα:

Η κάθε σελίδα διαχωρίζεται σε τέσσερις βασικές περιοχές, οι οποίες θα περιγραφούν ξεχωριστά στη συνέχεια.

Ξεκινώντας από αριστερά προς τα δεξιά, στην πάνω αριστερά γωνία, που

Chart Template Builder

Create new template Chart Library Guidelines Logout

Menu

- My Templates
- My Account

Notes

Here you can handle your account or the templates you have created. In order to handle your account select the option **My account** from the menu, where you can change your email and/or password. Otherwise you can handle the templates you have created.

My Templates: dimitra

Template title	Options
ko	Modify Delete Select
bar	Modify Delete Select
Bar3D	Modify Delete Select
Multiple Lines Chart 2	Modify Delete Select
Bar Chart 1	Modify Delete Select

Please select a chart template

Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

ΕΛΛΑΔΑ
2008
Αθήνα-πύλη Ανατολής

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΘΝΙΑ ΥΠΟΥΡΓΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΣΤΡΟΦΩΝ
ΕΡΕΥΝΑ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ
ΣΥΝΟΧΗ ΚΑΙ ΑΝΑΠΤΥΞΗ
ΕΡΕΥΝΑ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ
ΕΡΕΥΝΑ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ

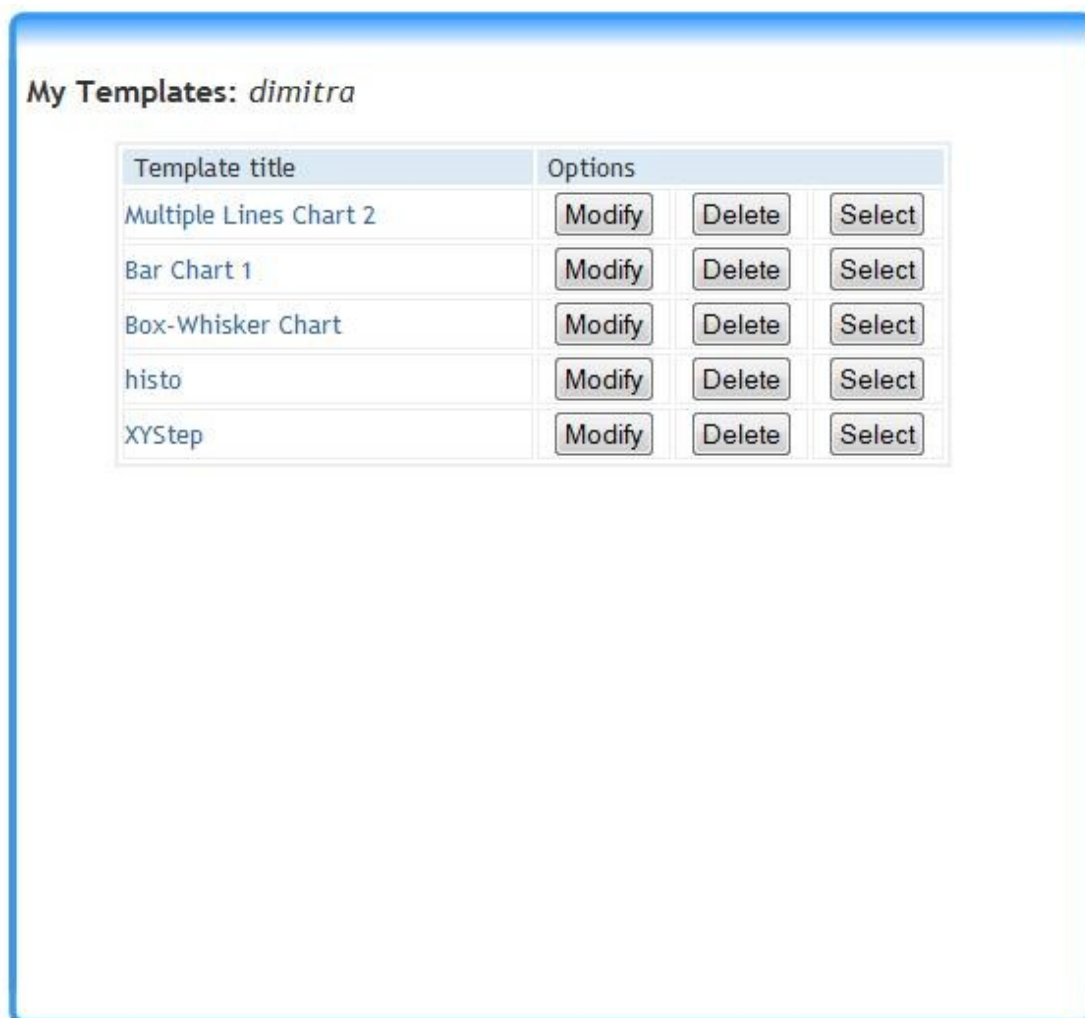
Εικόνα 50 - Προσωπική σελίδα χρήστη

είναι και η περιοχή που πέφτει πρώτα το μάτι, υπάρχει το βασικό μενού με τις επιλογές που έχει κάθε χρήστης ανάλογα με το σημείο που βρίσκεται. Οι επιλογές αυτές εξαρτώνται από το βήμα που βρίσκεται ο χρήστης. Από κάτω ακριβώς βρίσκεται η βοήθεια για το χρήστη, έτσι ώστε σε κάθε βήμα να έχει γνώση του τι χρειάζεται να κάνει.



Εικόνα 51 – Μενού επιλογών στην προσωπική σελίδα χρήστη

Στο κέντρο κάθε σελίδας υπάρχει το βασικό περιεχόμενο κάθε βήματος. Αυτό αλλάζει δυναμικά ανάλογα με το βήμα που βρίσκεται ο χρήστης. Για παράδειγμα, στην παραπάνω εικόνα που απεικονίζεται η προσωπική σελίδα κάθε χρήστη, το βασικό περιεχόμενο είναι όλα τα template γραφήματα που έχει δημιουργήσει. Έτσι εμφανίζεται ένας πίνακας με όλα τα γραφήματα του χρήστη και δίπλα οι επιλογές που έχει πάνω σε αυτά.



Εικόνα 52 – Τα πρότυπα γραφήματα ενός χρήστη και οι επιλογές διαχείρισης αυτών

Στο κέντρο και πάνω υπάρχει το βασικό μενού πλοήγησης μέσα στην εφαρμογή. Το μενού αυτό είναι σταθερό και παρουσιάζεται στην επόμενη εικόνα:



Εικόνα 53 - Βασικό μενού επιλογών

Οι επιλογές που έχει από εδώ ο κάθε χρήστης είναι:

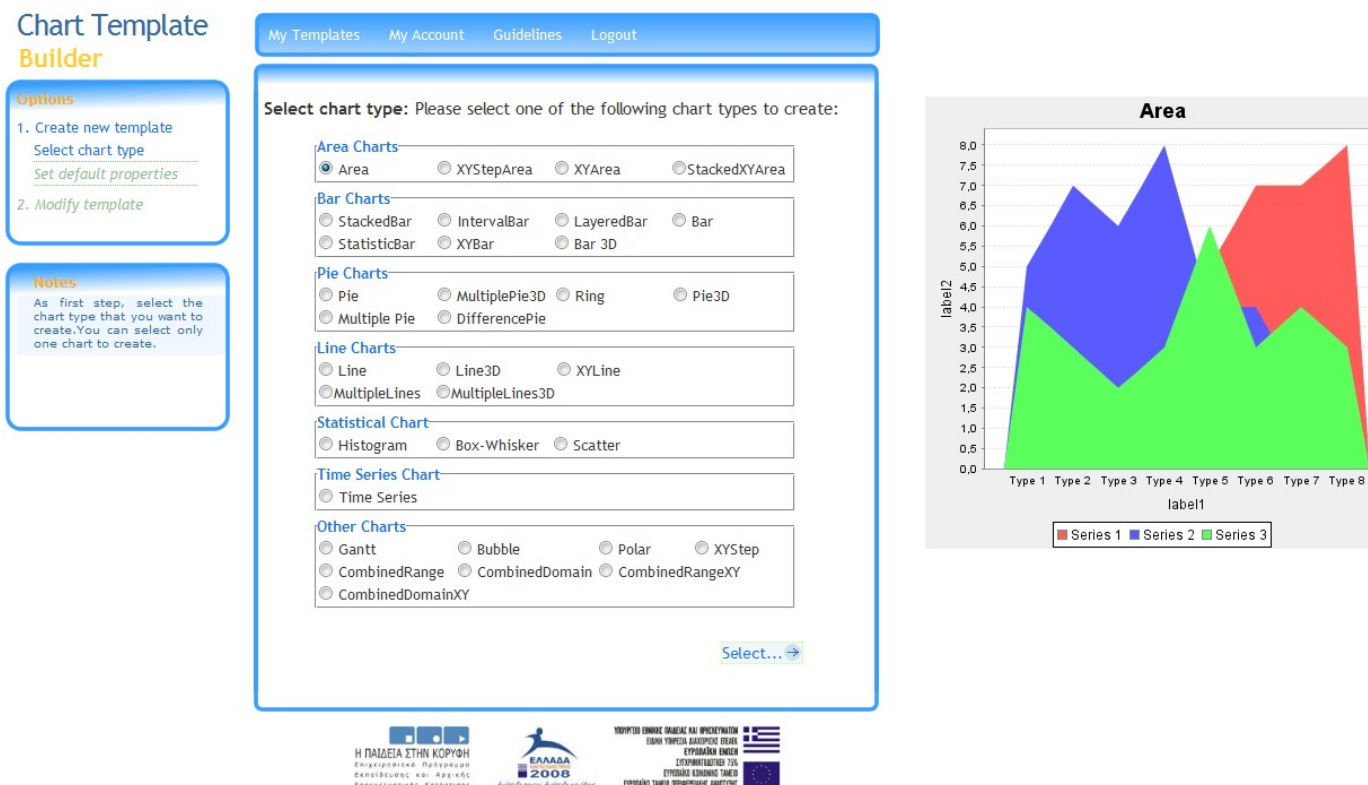
- **Create New Template:** για τη δημιουργία ενός νέου template γραφήματος

- **Chart Library:** για να δει όλα τα έτοιμα γραφήματα που υπάρχουν στη βιβλιοθήκη γραφημάτων και να επιλέξει κάποιο από αυτά αν επιθυμεί.
- **Guidelines:** είναι μια βοήθεια με οδηγίες για το σωστό σχεδιασμό γραφημάτων
- **Logout:** για να αποσυνδεθεί από το σύστημα.

Οι επιλογές αυτές παρουσιάζονται αναλυτικά παρακάτω.

Τέλος, η περιοχή δεξιά της σελίδας είναι δεσμευμένη ώστε να εμφανίζεται μια εικόνα με το γράφημα για να μπορεί ο χρήστης να βλέπει τις αλλαγές που θέλει να εφαρμόσει. Στη σελίδα του χρήστη η εικόνα αυτή αλλάζει δυναμικά ανάλογα με το template που θα επιλέξει να εμφανιστεί.

Από το βασικό μενού πλοήγησης που παρουσιάσαμε παραπάνω (Εικόνα 53), η επιλογή Create New Template, επιτρέπει στο χρήστη να ξεκινήσει την δημιουργία ενός νέου πρότυπου γραφήματος. Το πρώτο βήμα σε αυτή την περίπτωση είναι η επιλογή τύπου γραφήματος (Εικόνα 54):



Εικόνα 54 - Δημιουργία νέου γραφήματος

Όπως βλέπουμε από την παραπάνω εικόνα το βασικό περιεχόμενο αλλάζει και παρουσιάζει τις κατάλληλες επιλογές για το χρήστη. Εδώ παρουσιάζονται όλοι

δυνατοί τύποι γραφημάτων για δημιουργία. Κάθε φορά που επιλέγει έναν, δίπλα στην εικόνα εμφανίζεται και ένα αντίστοιχο δείγμα ώστε να γνωρίζει ο χρήστης ποιος είναι ο κάθε τύπος.

Επίσης εκείνο που αλλάζει εδώ αλλά και σε όλη τη διάρκεια δημιουργίας ενός γραφήματος είναι το μενού επιλογών στα αριστερά. Τώρα το μενού αυτό παρουσιάζει τις επιλογές που έχει για όλες τις παραμέτρους ενός γραφήματος, οργανωμένες σε κατηγορίες. Επιπλέον υπάρχει και διαχωρισμός με χρώματα για να γνωρίζει ο χρήστης σε ποιο βήμα βρίσκεται.

Για παράδειγμα έστω ότι ο χρήστης επιλέξει να δημιουργήσει μια πίτα από τις επιλογές που φαίνονται στην παραπάνω εικόνα (Εικόνα 54), το επόμενο βήμα φαίνεται παρακάτω:

Chart Template Builder

Options

- Create new template
 - Select chart type
 - Set default properties
- Modify template

Notes

Here you should give certain initial parameters for the chart. Below you might process more in detail these parameters. With **template name**, the template is saved in database. Its size should be less than 30 characters. The **chart title** is the main title for the chart. **Legend**, **tooltips**, and **urls** configure chart to generate them. If you do not add legend or tooltip then these parameters would not be presented afterwards.

My Templates My Account Logout

Default properties: Please fill the attributes below to create the chart:

Chart Parameters

Set the name to save the template:

Insert the main chart title:

Insert the image width:

Insert the image height:

Configure Chart

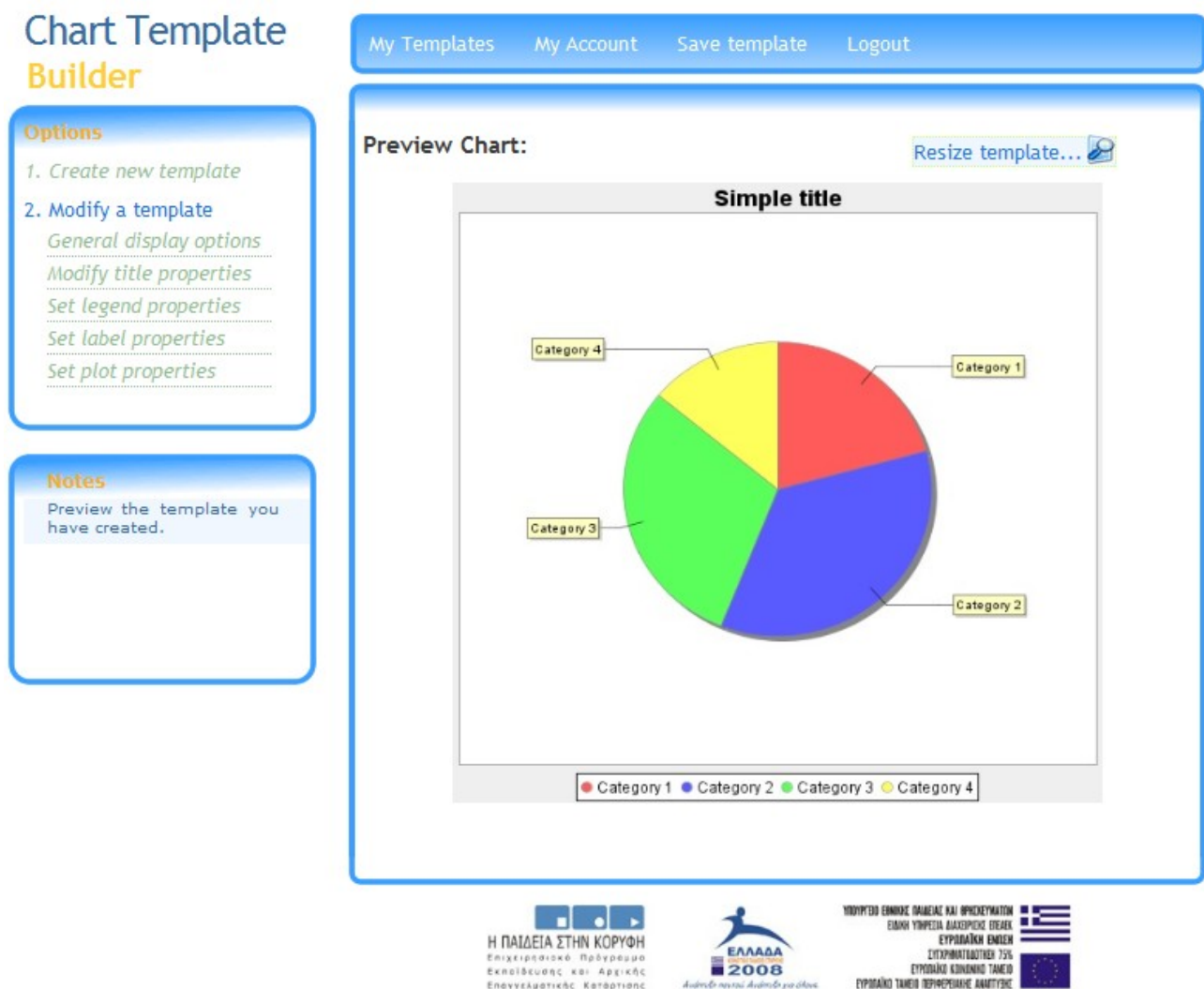
Add legend ☐ Add tooltips ☐ Add urls ☐

Create template

Εικόνα 55 – Καθορισμός βασικών παραμέτρων για τη δημιουργία ενός γραφήματος

Σε αυτό το βήμα ο χρήστης πρέπει να θέσει κάποιες βασικές παραμέτρους για τη δημιουργία του πρότυπου γραφήματος, όπως το όνομα με το οποίο θα αποθηκευτεί στη βάση, τις διαστάσεις του κ.ά. Οι παράμετροι αυτοί εξαρτώνται από τον τύπο γραφήματος.

Όταν ο χρήστης επιλέξει Create Template, τότε δημιουργείται το πρότυπο γράφημα και το επόμενο βήμα φαίνεται στην επόμενη εικόνα:



Εικόνα 56 - Preview template

Μέχρι αυτό το βήμα, έχει δημιουργηθεί το default template γράφημα πίτας, και από εδώ και έπειτα ο χρήστης είτε αποθηκεύει το template ως έχει είτε το επεξεργάζεται περαιτέρω.

Κεφάλαιο 5 – Παρουσίαση του User Interface

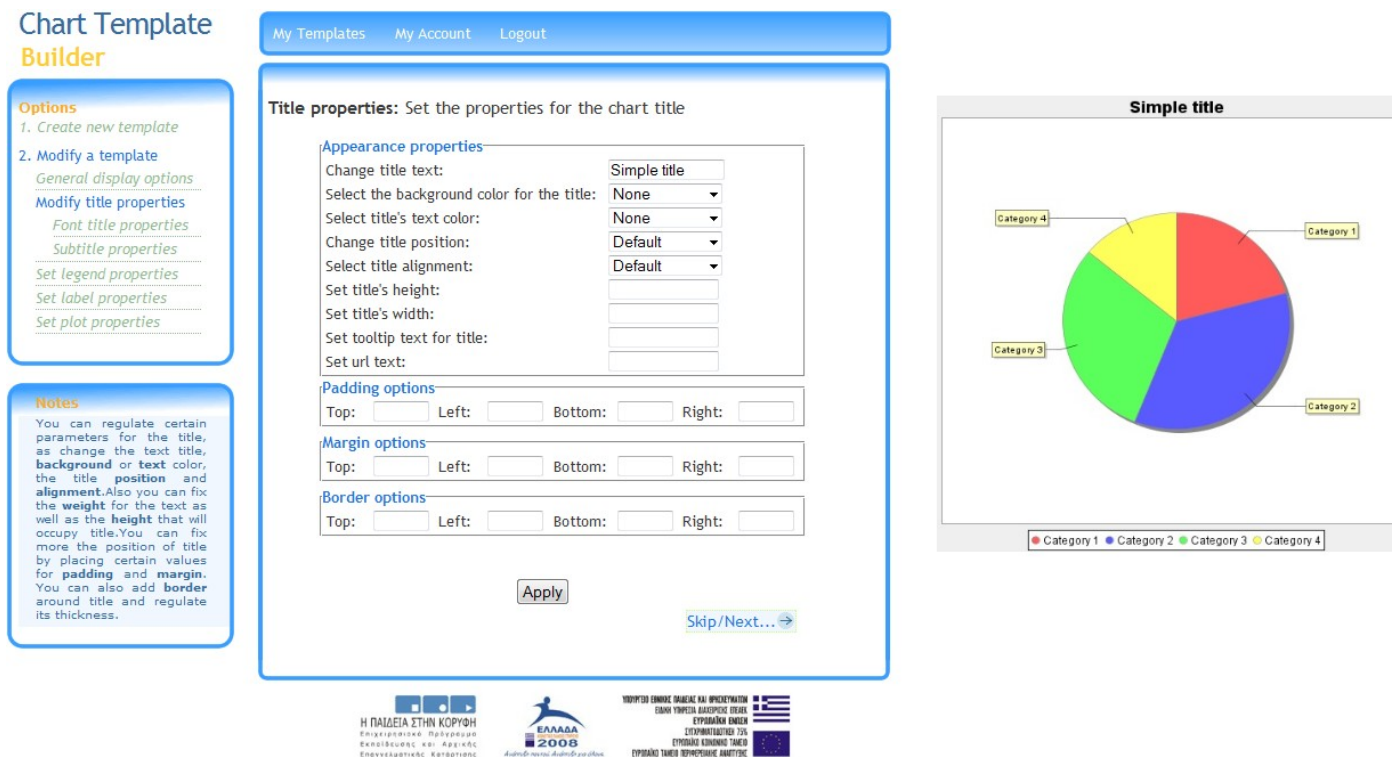
Να σημειώσουμε εδώ ότι, η εικόνα του γραφήματος εμφανίζεται στη σελίδα προσαρμοσμένη σε συγκεκριμένες διαστάσεις ώστε να μην υπάρχει πρόβλημα χώρου. Επιλέγοντας *Resize Image*, φαίνεται το πραγματικό μέγεθος της εικόνας.

Όπως φαίνεται στο μενού αριστερά, παρουσιάζονται όλες οι παράμετροι που μπορεί να επεξεργαστεί κάθε χρήστης οργανωμένες σε κατηγορίες (Εικόνα 57):



Εικόνα 57 – Βασικό μενού πλοήγησης για την επεξεργασία παραμέτρων του γραφήματος

Αν ο χρήστης αποφασίσει να επεξεργαστεί περεταίρω ένα γράφημα, μπορεί να πλοηγηθεί από το παραπάνω μενού και να επιλέξει μια από τις κατηγορίες παραπάνω για επεξεργασία. Για παράδειγμα αν επιλέξει να επεξεργαστεί τον τίτλο του γραφήματος, το επόμενο βήμα φαίνεται στην επόμενη εικόνα:



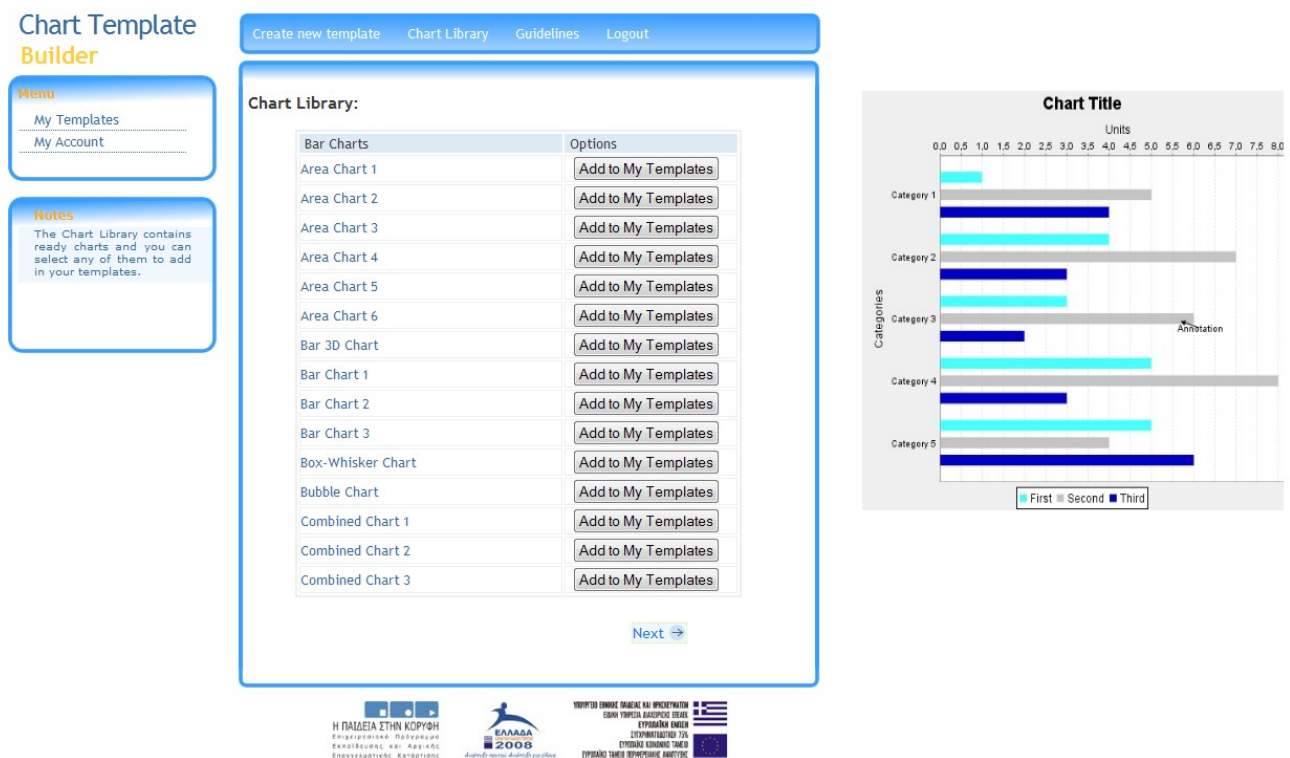
Εικόνα 58 - Επιλογές παραμετροποίησης του τίτλου του γραφήματος

Όπως βλέπουμε το βασικό περιεχόμενο της σελίδας παρουσιάζει τις παραμέτρους για τον τίτλο και το μενού αριστερά τις υποκατηγορίες στην κατηγορία του τίτλου. Ο χρήστης μπορεί να πλοηγηθεί παρακάτω είτε από την επιλογή κάτω δεξιά Skip/Next είτε από το μενού αριστερά.

Κάθε φορά που ολοκληρώνεται μια κατηγορία παραμέτρων, ο χρήστης επιστρέφει στην σελίδα στην εικόνα 56.

Όπως αναφέραμε παραπάνω, ο χρήστης μπορεί να επιλέξει ένα από τα γραφήματα της βιβλιοθήκης. Η σχετική σελίδα φαίνεται στην παρακάτω εικόνα:

Κεφάλαιο 5 – Παρουσίαση του User Interface



Εικόνα 59 - Παρουσίαση της ψηφιακής βιβλιοθήκης

Ο χρήστης μπορεί να δει όλα τα γραφήματα που υπάρχουν στην βιβλιοθήκη και αν κάποιον ενδιαφέρει μπορεί να επιλέξει να το προσθέσει στα δικά του templates, με την επιλογή Add to my Templates.

Στην εικόνα 60, στην σελίδα κάθε χρήστη φαίνονται και οι επιλογές που έχει για κάθε πρότυπο γράφημα, οι οποίες είναι:

1. **Modify:** για επεξεργασία των παραμέτρων. Η επόμενη σελίδα είναι αυτή στην εικόνα 78.
2. **Delete:** για διαγραφή του template
3. **Select:** για να συγκεντρώσει τις πληροφορίες που χρειάζεται για απεικόνιση δεδομένων όταν θα καλέσει το web service. Η σελίδα που εμφανίζεται φαίνεται στην επόμενη εικόνα:

Chart Template Builder

Menu

[My Templates](#)

[My account](#)

Notes

Here is the download center. In order to create your own chart, you need to download the files above.

Create new template Logout

Downloads!

In order to create your own chart you should download the following files.

Initially, the file `chartProperties.jar` will help you create the `.xml` file that needs in order to import your data and other optional choices for chart, as labels, annotations e.t.c with the help of `XmlBeans-2.3.0`.

In this `.xml` file it will be supposed you save also add the path that the chart is saved , as precisely it is given in the following table . There is also a file `(.java)` in which is materialised client in order to call web service. Finally, in order to work all right, make sure that the enviromental variable `CATALINA_HOME` is set to the Tomcat's base dir.

Template's Info	
Username:	<code>dimitra</code>
Configuration filename:	<code>BarTemplate1Properties</code>
Downloads	
Available libraries (.jar)	chartProperties
Test Client(.java)	Web Service Client
Manual(.doc)	Manual

Εικόνα 60 – Πληροφορίες κάθε γραφήματος και διαθέσιμα αρχεία

Όπως φαίνεται και στην εικόνα, ο χρήστης συγκεντρώνει τις πληροφορίες που χρειάζονται και είναι διαθέσιμα κάποια αρχεία όπως η απαραίτητη βιβλιοθήκη (`chartProperties.jar`) για τη δημιουργία `xml` αρχείου με δεδομένα, μια κλάση της Java (`TestClient.java`) που υλοποιεί ένα παράδειγμα `client service`, και αρχείο με οδηγίες για το πώς να δημιουργήσει ένα `xml` αρχείο με δεδομένα (`Manual.doc`).

Επίσης, στην σελίδα χρήστη είναι διαθέσιμη και η επιλογή `My Account`, μέσω της οποίας ο κάθε χρήστης μπορεί να επεξεργαστεί τον προσωπικό του λογαριασμό. Τα στοιχεία που μπορεί να αλλάξει είναι το `email` του και το `password` (Εικόνα 61).

The screenshot displays the 'Chart Template Builder' web application. On the left, there is a sidebar with a 'Menu' section containing 'My Templates' and 'My Account', and a 'Notes' section with the text 'Change your account details such as your password and/or your email.' The main content area is titled 'Modify Account: Set the new password and/or new email'. It features a form with two input fields: 'New password:' and 'New email:'. Below these fields is a 'Save' button. At the top of the main area, there are links for 'Create new template' and 'Logout'. The bottom of the page contains several logos, including the Ministry of Education and Religious Affairs, the Hellenic Republic, and the European Union.

Εικόνα 61 - Επεξεργασία στοιχείων λογαριασμού

Τέλος, σε περίπτωση που κάποιος δεν διαθέτει λογαριασμό, μπορεί να δημιουργήσει ένα νέο λογαριασμό, επιλέγοντας Create new account από την αρχική σελίδα. Τα στοιχεία που χρειάζεται να συμπληρώσει φαίνονται στην επόμενη εικόνα:

Chart Template Builder

Notes
Fill the required fields in order to create your account. If everything goes well, your account will be ready after 24 hours.

Login Create new account

User information

First name:

Last name:

Email:

Account details

Username:

Password:

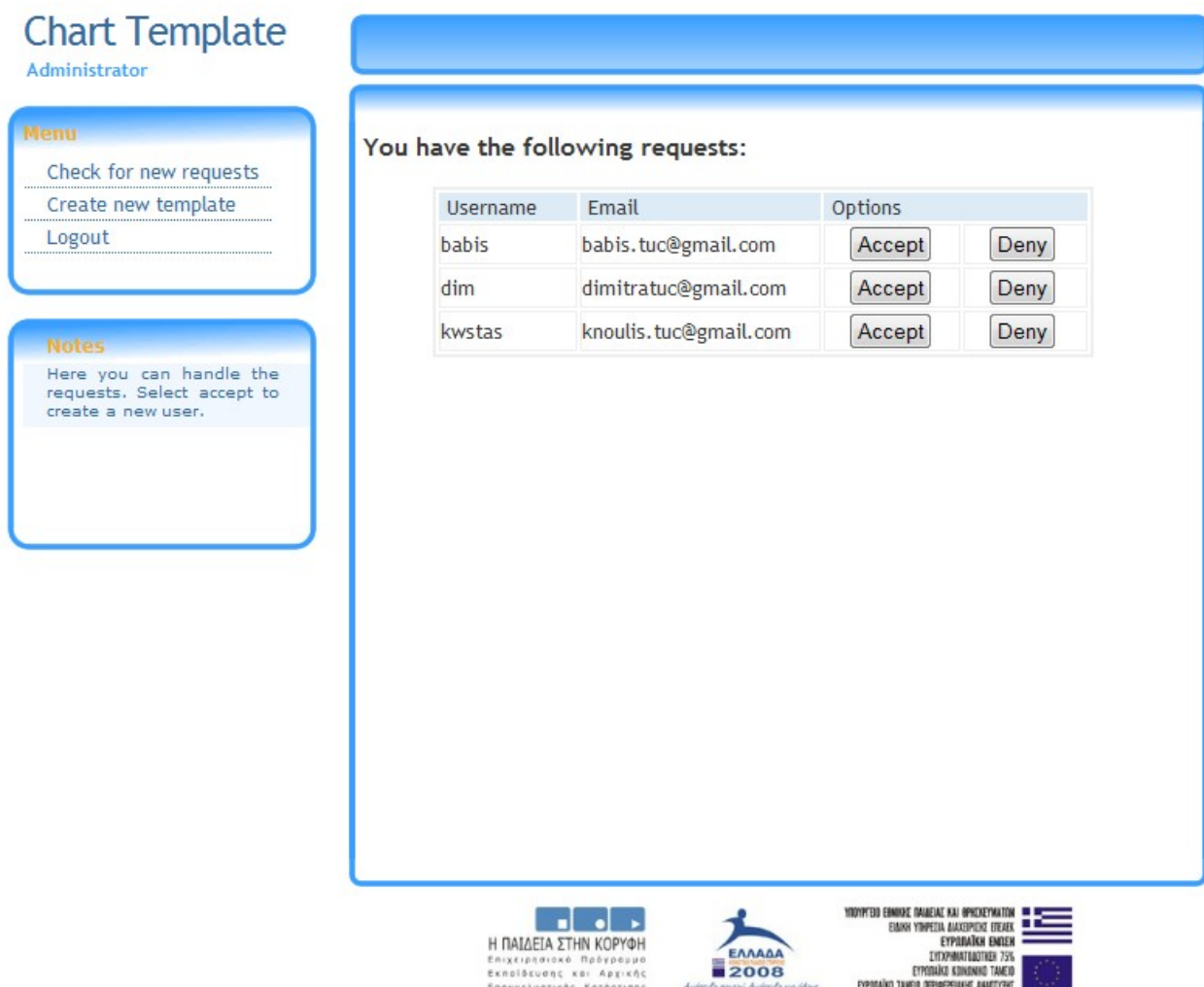
Confirm password:

Sign Up

Εικόνα 62 - Δημιουργία νέου λογαριασμού

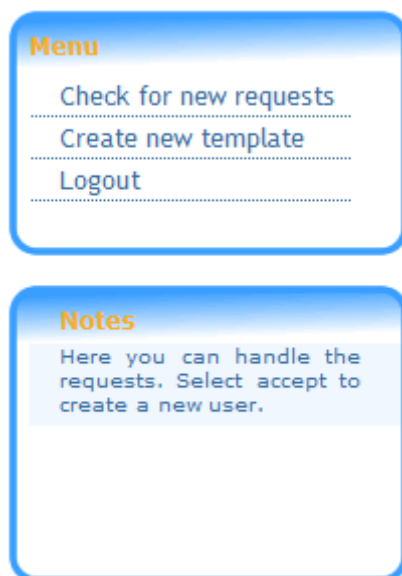
5.3 Παρουσίαση της γραφικής διεπαφής για τον διαχειριστή της εφαρμογής

Ο Διαχειριστής της εφαρμογής όπως έχουμε αναφέρει, διαχειρίζεται τα αιτήματα για τη δημιουργία νέων λογαριασμών στην εφαρμογή. Επιπλέον, έχει και την επιλογή να δημιουργήσει δικά του πρότυπα γραφήματα, να τα επεξεργαστεί και τη μοναδική αρμοδιότητα να τα προσθέσει στην βιβλιοθήκη πρότυπων γραφημάτων. Όταν ο Διαχειριστής κάνει είσοδο στο σύστημα, βρίσκεται στην κεντρική σελίδα που φαίνεται στην επόμενη εικόνα:



Εικόνα 63 - Διαχείριση των αιτημάτων για νέο λογαριασμό από τον Διαχειριστή

Όπως παρατηρούμε από την παραπάνω εικόνα, από τη σελίδα αυτή ο Διαχειριστής μπορεί και ενημερώνεται για όλα τα αιτήματα δημιουργίας νέων λογαριασμών. Για κάθε αίτημα ο Διαχειριστής μπορεί να το αποδεχτεί ή να το απορρίψει. Σε περίπτωση αποδοχής, ο λογαριασμός ενεργοποιείται αμέσως. Αντίθετα, σε περίπτωση απόρριψης ο λογαριασμός δεν δημιουργείται ποτέ. Στην πάνω αριστερή γωνία υπάρχει το βασικό μενού επιλογών για τον Διαχειριστή:



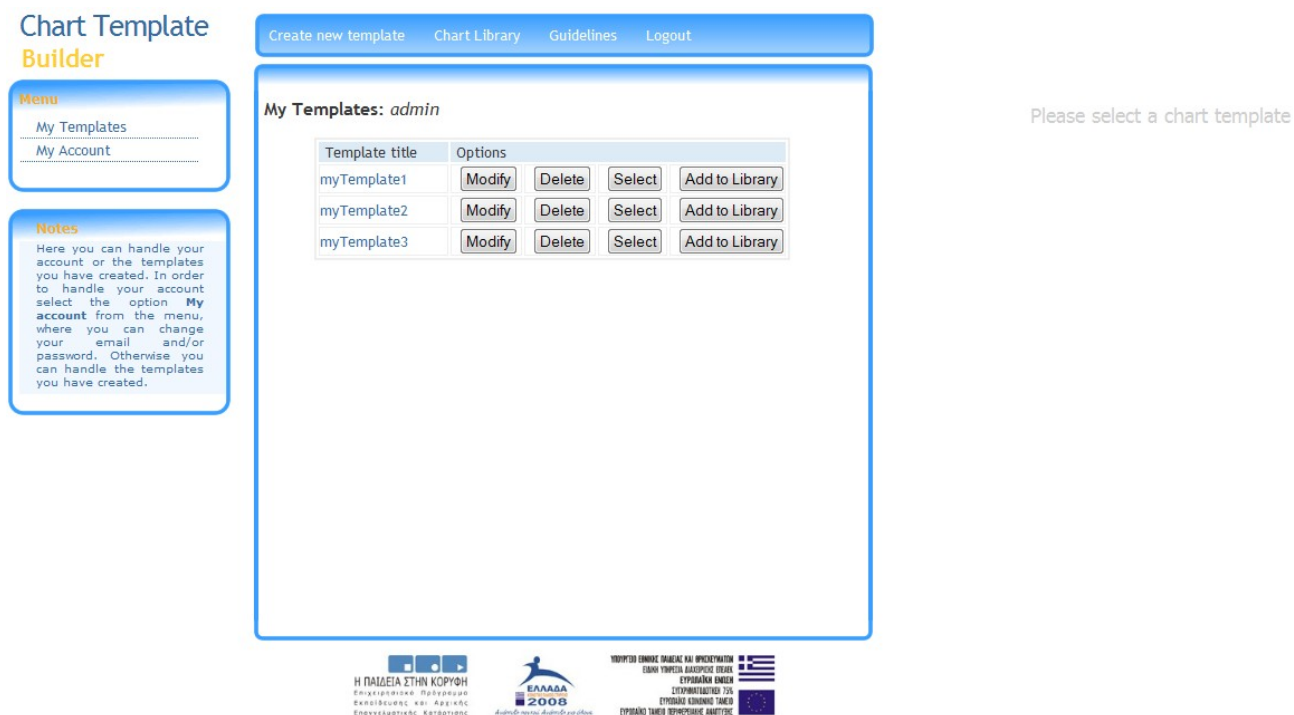
Εικόνα 64 - Βασικό μενού επιλογών πλοήγησης για τον Διαχειριστή

Όπως βλέπουμε οι επιλογές που έχει ο Διαχειριστής είναι:

Check for new requests: για να ελέγχει κάθε φορά αν υπάρχουν νέα αιτήματα

Create new template: για να δημιουργεί τα δικά του πρότυπα γραφήματα.

Η δεύτερη επιλογή μεταφέρει τον Διαχειριστή στην παρακάτω σελίδα:



Εικόνα 65 - Διαχείριση των πρότυπων γραφημάτων για τον Διαχειριστή

Ο Διαχειριστής μπορεί να δημιουργήσει και να διαχειριστεί τα δικά του πρότυπα γραφήματα με τον ίδιο τρόπο που παρουσιάσαμε παραπάνω για όλους τους χρήστες της εφαρμογής. Η επιπλέον επιλογή που έχει ο Διαχειριστής είναι να προσθέσει ένα πρότυπο γράφημα που έχει δημιουργήσει, στην ψηφιακή βιβλιοθήκη γραφημάτων επιλέγοντας Add to Library για ένα γράφημα.

5.4 Περίληψη

Στο κεφάλαιο αυτό παρουσιάσαμε τη γραφική διεπαφή της εφαρμογής μας. Η διεπαφή σχεδιάστηκε σύμφωνα με τις αρχές σχεδιασμού, ώστε να προσφέρει έναν εύχρηστο και άνετο τρόπο πλοήγησης. Επίσης, παρουσιάσαμε αναλυτικά τον τρόπο που ο εκάστοτε χρήστης χρησιμοποιεί τις επιλογές που έχει για να εκτελέσει τις απαραίτητες ενέργειες.

Κεφάλαιο 6

Τεχνολογίες Υλοποίησης του Συστήματος

6.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε λεπτομερώς όλες τις τεχνολογίες που χρησιμοποιήθηκαν κατά το σχεδιασμό της εφαρμογής. Επίσης, για τις ανάγκες της εφαρμογής αναπτύχθηκαν κάποια σχήματα και μοντέλα, όπως για παράδειγμα το xml αρχείο για την αποθήκευση των παραμέτρων ενός γραφήματος, η βάση δεδομένων κ.ά. Στο κεφάλαιο αυτό θα περιγράψουμε αυτά τα σχήματα όπως υλοποιήθηκαν.

6.2 Τεχνολογίες που χρησιμοποιήθηκαν στην υλοποίηση του συστήματος

JavaScript

Είναι μια scripting γλώσσα για το διαδίκτυο [18]. Χρησιμοποιείται σε σχεδόν όλες τις ιστοσελίδες για να προσθέσει λειτουργικότητα, να επικυρώσει ή όχι τα στοιχεία μιας φόρμας, να διακρίνει τον εκάστοτε browser και πολλά άλλα. Είναι μια απλή γλώσσα προγραμματισμού. Μπορεί να υπάρχει απευθείας μέσα σε ένα HTML έγγραφο ή να εισαχθεί από ένα ή περισσότερα εξωτερικά αρχεία. Στην εφαρμογή που σχεδιάσαμε, η γλώσσα Java Script, χρησιμοποιήθηκε κατά το σχεδιασμό των ιστοσελίδων για την επίτευξη συγκεκριμένων στόχων όπως: ο έλεγχος στοιχείων συμπλήρωσης μιας φόρμας, για την προσαρμογή των διαστάσεων των εικόνων σε συγκεκριμένες τιμές ανεξάρτητα από αυτές που ορίζει ο χρήστης, ώστε να αποφευχθεί τυχόν πρόβλημα με το χώρο και δυνατότητα εμφάνισης της εικόνας σε πραγματικό μέγεθος, για τον έλεγχο αποθήκευσης ενός γραφήματος σε περίπτωση log out από το σύστημα κ.ά.

Servlets

Τα servlets [19] είναι μια τεχνολογία που χρησιμοποιούνται για να δέχονται και να απαντούν σε αιτήματα (requests) από Web clients, συνήθως μέσω του πρωτοκόλλου HTTP. Είναι μικρά κομμάτια κώδικα που τρέχουν από τη πλευρά του

web server . Κάθε servlet έχει ένα συγκεκριμένο «κύκλο ζωής» (life-cycle) : κατασκευάζεται και αρχικοποιείται, στη συνέχεια όταν καλείται, διαχειρίζεται το αίτημα που δέχεται από τον client και στέλνει μια απάντηση ως επιστροφή, και τέλος καταστρέφεται όταν έχει τελειώσει. Η τεχνολογία των servlets χρησιμοποιήθηκε στην εφαρμογή από τη πλευρά του server, για να απαντά σε κάθε αίτημα που υποβάλλει κάθε χρήστης. Για παράδειγμα, κάθε φορά που ο χρήστης τροποποιεί τις παραμέτρους ενός γραφήματος, ένα servlet είναι υπεύθυνο να αποκριθεί σε αυτό το αίτημα, και να επιστρέψει στο χρήστη το ανανεωμένο γράφημα.

Java Server Pages (JSP)

Είναι μια τεχνολογία που χρησιμοποιείται για τη δημιουργία εφαρμογών διαδικτύου (web application) με δυναμικό περιεχόμενο [19]. Για το λόγο αυτό, χρησιμοποιήσαμε την τεχνολογία των JSPs για το σχεδιασμό των ιστοσελίδων της εφαρμογής. Ένα JSP μπορεί να αποτελείται από:

- Στατικά HTML/XML στοιχεία
- Ειδικά JSP tags

Τα JSP είναι μια τεχνολογία της Java που επιτρέπει τη δημιουργία HTML, XML ή άλλων εγγράφων δυναμικά ως απάντηση σε ένα αίτημα ενός web client.

Η σύνταξη JSP χρησιμοποιεί κάποια tags, πολύ κοντά στην XML, τα οποία αποκαλούνται JSP actions, και προσφέρουν συγκεκριμένη λειτουργία σε ένα JSP. Επιπλέον η τεχνολογία επιτρέπει τη δημιουργία των JSP tag libraries , οι οποίες ενεργούν ως επεκτάσεις των καθιερωμένων HTML ή XML tags.

Τα πλεονεκτήματα των JSP είναι:

1. υπάρχει διαχωρισμός ανάμεσα στο στατικό και στο δυναμικό περιεχόμενο μιας σελίδας,
2. Write Once Run Everywhere
3. Το δυναμικό περιεχόμενο μπορεί να διανεμηθεί να σε ποικίλα formats
4. Ισχυροποιεί πλήρως το Servlet API

eXtensible Markup Language (XML)

Η XML [10] είναι μια γλώσσα που σχεδιάστηκε για να μεταφέρει και να αποθηκεύει δεδομένα. Είναι μια markup γλώσσα όπως η HTML, ήταν σχεδιασμένη για να μεταφέρει δεδομένα και όχι να τα εμφανίζει. Η XML δεν έχει καθορισμένα tags που απλά να εμφανίζονται κατά τη δημιουργία ενός xml έγγραφου. Πρέπει ο χρήστης να ορίζει κάθε φορά τα δικά του tags, όπως επιθυμεί. Σχεδιάστηκε για να είναι μια εμπεριστατωμένη γλώσσα και είναι πλέον W3C Recommendation.

Η XML είναι επεκτάσιμη γλώσσα (eXtensible) γιατί αφήνει τον χρήστη να ορίσει τα δικά του στοιχεία (elements) και γνωρίσματα (attributes) έτσι ώστε να είναι σε θέση να περιγράψει δεδομένα οποιουδήποτε τύπου. Έχει την δυνατότητα να χρησιμοποιηθεί σε ένα τεράστιο σύνολο εφαρμογών αφού είναι συμβατή με την πλειονότητα των πρωτοκόλλων μετάδοσης δεδομένων που χρησιμοποιούνται σήμερα στο διαδίκτυο, είναι πολύ φιλική προς τους χρήστες και δεν εξαρτάται από κάποιο λειτουργικό σύστημα.

Τα XML έγγραφα είναι έγγραφα τα οποία περιέχουν δομημένο κείμενο. Το έγγραφο αποτελείται από στοιχεία (elements), γνωρίσματα (attributes) και δεδομένα (data).

Τα στοιχεία (elements) περικλείονται από < > και κάθε στοιχείο που ανοίγει πρέπει να έχει και ένα τέλος που σηματοδοτείται με </>. Κάθε στοιχείο είναι δυνατόν να έχει ένα ή περισσότερα γνωρίσματα. Σκοπός των γνωρισμάτων είναι να δώσουν περισσότερες λεπτομέρειες για να περιγράψουν τα δεδομένα.

Ένα πολύ απλό παράδειγμα XML εγγράφου είναι το ακόλουθο:

```
<person>
  <name>Dimitra</name>
  <age>24</age>
  <e-mail>drosiou@ced.tuc.gr</e-mail>
</person>
```

Στο παραπάνω έγγραφο XML βλέπουμε πώς μπορούμε να αναπαραστήσουμε με δομημένο τρόπο πληροφορία που αναφέρεται σε ένα πρόσωπο και συγκεκριμένα το όνομα, την ηλικία και την ηλεκτρονική του διεύθυνση.

Για να είναι σωστά δομημένο (well formed) ένα XML έγγραφο πρέπει να τηρούνται ορισμένοι κανόνες. Η XML έχει αυστηρή σύνταξη και αυτό βοηθάει στο

να είναι πιο απλό να υλοποιηθούν XML parsers. Πάντα στην αρχή του εγγράφου πρέπει να υπάρχει ένα στοιχείο ρίζα το οποίο περιέχει μέσα του όλα τα υπόλοιπα. Εκτός από αυτό δεν επιτρέπονται επικαλυπτόμενα στοιχεία.

Για παράδειγμα, το επόμενο έγγραφο XML δεν είναι σωστά δομημένο γιατί περιέχει δύο επικαλυπτόμενα στοιχεία:

```
<Στοιχείο 1>  
  <Στοιχείο 2>  
  </Στοιχείο 1>  
</Στοιχείο 2>
```

Η σωστή σύνταξη του παραπάνω παραδείγματος είναι:

```
<Στοιχείο 1>  
  <Στοιχείο 2>  
  </Στοιχείο 2>  
</Στοιχείο 1>
```

Επίσης σε ένα στοιχείο δεν επιτρέπεται να υπάρχουν γνωρίσματα που έχουν το ίδιο όνομα. Τέλος πρέπει για κάθε στοιχείο που ανοίγει να υπάρχει ένα ίδιο στοιχείο που το κλείνει (με τον τρόπο που αναφέραμε παραπάνω). Αν ισχύουν όλα τα παραπάνω το έγγραφο είναι σωστά δομημένο (well formed) δεν σημαίνει όμως ότι τα στοιχεία έχουν τα ονόματα που θέλει κάποιος ή την σειρά την οποία επιθυμούμε. Γι' αυτό το λόγο δημιουργήθηκε η XML Schema.

Η XML χρησιμοποιήθηκε στην παρούσα διπλωματική για τους εξής σκοπούς: πρώτον , για την δημιουργία ενός xml εγγράφου στο οποίο αποθηκεύονται με συγκεκριμένη δόμηση οι παράμετροι ενός γραφήματος. Δεύτερον, για την εισαγωγή δεδομένων στο Web Service. Κάθε φορά που ένας χρήστης θέλει να εισάγει πραγματικά δεδομένα σε ένα πρότυπο γράφημα, δημιουργεί ένα xml αρχείο με τα δεδομένα του και καλεί το Web Service, με είσοδο το αρχείο αυτό. Τέλος, για την επιστροφή του τελικού γραφήματος από το web service. Το web service επιστρέφει ένα xml έγγραφο, το οποίο περιέχει ως στοιχεία το όνομα της εικόνας και τα binary data της εικόνας. Η δομή που ακολουθεί κάθε ένα από τα αρχεία αυτά, είναι συγκεκριμένη και έχει οριστεί με τη βοήθεια της Xml-Schema που περιγράφεται παρακάτω.

XML-Schema

Η XML Schema [11] είναι βασισμένη στην XML και προσδιορίζει την ονοματολογία των στοιχείων (elements) αλλά και των γνωρισμάτων (attributes). Επίσης προσδιορίζει την σειρά με την οποία θα εμφανίζονται τα στοιχεία και ποια γνωρίσματα έχει το κάθε στοιχείο. Εκτός από τα παραπάνω υπάρχει η δυνατότητα να οριστούν πιο πολύπλοκοι κανόνες. Όταν ένα XML έγγραφο έχει δομηθεί με τέτοιο τρόπο ώστε να ακολουθεί τους κανόνες ενός XML Schema τότε λέμε ότι είναι έγκυρο (valid).

Η XML Schema χρησιμοποιείται ουσιαστικά για τον ορισμό κλάσεων XML εγγράφων, όπου για κάθε κλάση καθορίζονται τα χαρακτηριστικά, το περιεχόμενο και η δομή των στιγμιότυπων της. Αξίζει να επισημάνουμε εδώ ότι ένα XML έγγραφο μπορεί να περιέχει κομμάτια τα οποία υπακούν σε διαφορετικά XML Schemata. Για να γίνει εφικτό αναπτύχθηκε ο μηχανισμός των namespaces (ονοματολογίας).

Η γλώσσα XML Schema επιτρέπει τον ορισμό απλών και σύνθετων στοιχείων. Τα σύνθετα στοιχεία (complex elements) ανήκουν σε σύνθετους τύπους (complex types) και μπορεί να διαθέτουν γνωρίσματα και να περιέχουν άλλα στοιχεία. Τα απλά στοιχεία (simple elements) ανήκουν σε απλούς τύπους (simple types), οι οποίοι συνήθως ορίζονται ως περιορισμοί (restrictions) των βασικών τύπων που παρέχονται από την XML Schema, όπως οι συμβολοσειρές (strings), οι αριθμοί (ακέραιοι και πραγματικοί), τα σύμβολα (tokens) κ.α. Τα απλά στοιχεία δε διαθέτουν γνωρίσματα και δε μπορούν να περιέχουν άλλα στοιχεία. Η γλώσσα XML Schema υποστηρίζει κληρονομικότητα (inheritance) και περιορισμούς (constraints) για όλους τους τύπους, απλούς και σύνθετους. Επιπλέον, υποστηρίζονται επαναχρησιμοποιήσιμοι ορισμοί στοιχείων

Η γλώσσα XML Schema υποστηρίζει γνωρίσματα, που αναπαριστούν τα χαρακτηριστικά των τύπων, για σύνθετους τύπους μόνο. Γνωρίσματα τα οποία πρέπει να χρησιμοποιούνται ταυτόχρονα σχηματίζουν ομάδες γνωρισμάτων (attribute groups). τόσο για γνωρίσματα όσο και για απλού τύπου στοιχεία υπάρχει η δυνατότητα προσδιορισμού προκαθορισμένων (fixed) και εξ' ορισμού (default) τιμών.

Τα στοιχεία που περιέχονται σε κάποιο άλλο στοιχείο ή στον ορισμό κάποιου τύπου μπορεί να είναι εναλλάξιμα, οπότε απαρτίζουν επιλογές (choices) ή να έχουν προκαθορισμένη σειρά, οπότε απαρτίζουν ακολουθίες (sequences). Τόσο οι ακολουθίες όσο και οι επιλογές μπορεί να είναι εμφωλευμένες (nested). Ο ελάχιστος και ο μέγιστος επιτρεπτός αριθμός στοιχείων μέσα σε ακολουθίες και επιλογές ορίζονται από τα γνωρίσματα minOccurs και maxOccurs αντίστοιχα.

Επαναχρησιμοποιήσιμες σύνθετες δομές, που μπορεί να συνδυάζουν ακολουθίες και επιλογές, απαρτίζουν τα μοντέλα ομάδων (model groups). Τα μοντέλα ομάδων και οι ομάδες γνωρισμάτων ορίζονται σε πρώτο επίπεδο (toplevel) μόνο, ενώ τα γνωρίσματα και τα στοιχεία μπορούν να οριστούν τόσο σε πρώτο επίπεδο όσο και μέσα στους ορισμούς τύπων. Οι τύποι (απλοί και σύνθετοι) μπορούν να οριστούν τόσο σε πρώτο επίπεδο όσο και μέσα στους ορισμούς στοιχείων και γνωρισμάτων.

Οι τύποι (απλοί και σύνθετοι), τα στοιχεία, τα γνωρίσματα, οι ομάδες γνωρισμάτων και τα μοντέλα ομάδας πρώτου επιπέδου διαθέτουν μοναδικά ονόματα, που αναπαρίστανται από το χαρακτηριστικό “name”. Ονόματα διαθέτουν και τα εμφωλευμένα στοιχεία και γνωρίσματα, τα οποία πρέπει να είναι μοναδικά στα όρια των πιο κοντινών τύπων μέσα στους οποίους ορίζονται. Να σημειωθεί ότι οι εμφωλευμένοι τύποι (απλοί και σύνθετοι) είναι ανώνυμοι.

Επιπλέον, όλες οι XML Schema δομές (γνωρίσματα, στοιχεία, επιλογές, ακολουθίες, απλοί και σύνθετοι τύποι, μοντέλα ομάδες και ομάδες γνωρισμάτων) μπορεί να διαθέτουν (προαιρετικά) μοναδικά αναγνωριστικά (identifiers) που αναπαρίστανται από το γνώρισμα “id”.

Η Xml-Schema χρησιμοποιήθηκε για τη ονοματολογία των στοιχείων και των γνωρισμάτων των xml αρχείων που αναφέραμε παραπάνω, καθώς και για τον προσδιορισμό της σειράς των στοιχείων αυτών.

XML Beans

Ο όρος “XML data binding” αναφέρεται στην αναπαράσταση XML εγγράφων με αντικείμενα. Τα αντικείμενα αυτά χρησιμοποιούν ένα σχήμα (classes), σχεδιασμένο ειδικά για τα δεδομένα που υπάρχουν σε αυτά τα έγγραφα. Αυτό

επιτρέπει στις εφαρμογές να διαχειρίζονται δεδομένα τα οποία υπάρχουν σε XML μορφή. Ας θεωρήσουμε για παράδειγμα το εξής τμήμα ενός XML εγγράφου :

```
<Note>
  <To>Anna</To>
  <From>Dimitra</From>
  <Text>
    <Heading>Reminder</Heading>
    <Body>Don't forget to send the e-mail tomorrow</Body>
  </Text>
</Note>
```

Το τμήμα αυτό του XML εγγράφου, μπορεί να αναπαρασταθεί με τις κλάσεις “Note”, “From”, “To”, “Text”, “Heading” και “Body”, η ιεραρχία των οποίων είναι σύμφωνα με το XML Schema στο οποίο υπακούει το παραπάνω XML έγγραφο. Έτσι, κάθε φορά που μεταφέρονται δεδομένα από το XML έγγραφο, το αποτέλεσμα είναι ένα δένδρο από αντικείμενα.

Υπάρχουν αρκετά εργαλεία στη γλώσσα προγραμματισμού Java, που χρησιμοποιούνται για XML data binding. Αυτό που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής είναι το XMLBeans [20]. Το εργαλείο αυτό μας επιτρέπει να εκμεταλλευτούμε την αφθονία και τα χαρακτηριστικά γνωρίσματα της XML και του XML Schema και να τα χαρτογραφήσουμε, όσο το δυνατόν φυσικότερα, στα ισοδύναμα κατασκευάσματα της Java. Το XMLBeans χρησιμοποιεί το XML Schema, στο οποίο υπακούουν κάποια XML έγγραφα, για να δημιουργήσει τις κατάλληλες Java διεπαφές και κλάσεις, οι οποίες μπορούν να χρησιμοποιηθούν για την πρόσβαση και τροποποίηση των XML δεδομένων. Αν και υπάρχουν διάφορα Java εργαλεία για τη διαχείριση των XML εγγράφων, όπως το DOM, το JAXB και το SAX, επιλέξαμε να χρησιμοποιήσουμε το XMLBeans λόγω των εξής πλεονεκτημάτων :

- Το XMLBeans μπορεί να χειριστεί οποιοδήποτε έγγραφο, όσο μεγάλο και αν είναι, χωρίς να υπερφορτώνεται η μνήμη.
- Ένα από τα μεγαλύτερα πλεονεκτήματα του XMLBeans είναι ότι παρέχει πλήρη υποστήριξη στην XMLSchema
- Μας δίνει τη δυνατότητα πιστοποίησης της εγκυρότητας ενός XML εγγράφου, τη στιγμή που αυτό φορτώνεται στη μνήμη ως δενδρική δομή στιγμιότυπων των κλάσεων που έχουν παραχθεί από το αντίστοιχο XML Schema.

Η τεχνολογία XML Beans χρησιμοποιήθηκε τόσο στην εφαρμογή δημιουργίας πρότυπων γραφημάτων όσο και στην ανακατασκευή του γραφήματος από το web service, για την διαχείριση των αρχείων xml που χρησιμοποιούμε.

My SQL

Είναι μια από τις καθιερωμένες γλώσσες για την δημιουργία και διαχείριση βάσεων δεδομένων [17] . Εκτελεί queries πάνω σε μια βάση δεδομένων και επιστρέφει τα αποτελέσματα αν υπάρχουν. Επιτρέπει την εισαγωγή (insert) δεδομένων σε μια βάση, την ανάκτηση (select) δεδομένων, την διαγραφή(delete) και την ενημέρωση(update) πάνω σε συγκεκριμένες εγγραφές μια βάσης.

Για τις ανάγκες της εφαρμογής δημιουργήσαμε μια σχεσιακή βάση δεδομένων. Μια βάση δεδομένων αποτελείται από πίνακες που έχουμε δημιουργήσει, οι οποίοι χρησιμεύουν για την αποθήκευση δεδομένων. Κάθε πίνακας έχει ένα όνομα και αποτελείται από πολλά στοιχεία. Κάθε στοιχείο από αυτά έχει ένα όνομα και ένα τύπο δεδομένων που το χαρακτηρίζει. Ένα από τα στοιχεία αυτά σε κάθε πίνακα έχει μοναδική τιμή σε κάθε εγγραφή του πίνακα και χαρακτηρίζεται ως «Primary Key». Μεταξύ των πινάκων της βάσης, υπάρχει μια σύνδεση που δηλώνεται με την προσθήκη των λεγόμενων «Foreign Keys».

Τα queries που εκτελούνται πάνω στη βάση έχουν να κάνουν με την ανάκτηση(SELECT), την διαγραφή(DELETE) και την ενημέρωση (UPDATE) δεδομένων στη βάση.

Η σύνταξη που ακολουθείται για κάθε ένα από τα παραπάνω queries είναι η ακόλουθη:

Ανάκτηση:

- `SELECT * FROM <table_name>` : για την ανάκτηση όλων των τιμών των στοιχείων ενός πίνακα
- `SELECT attribute1, attribute2,... FROM <table_name>` : για την ανάκτηση συγκεκριμένων στοιχείων ενός πίνακα
- Κάθε μια από τις δυο παραπάνω εκφράσεις μπορεί να συνοδεύεται από την έκφραση `WHERE` , για την ανάκτηση δεδομένων που ικανοποιούν την συνθήκη.

Διαγραφή:

- DELETE FROM <table_name> : για την διαγραφή όλων των εγγραφών ενός πίνακα
- DELETE FROM <table_name> WHERE *condition*: για την διαγραφή μιας εγγραφής που ικανοποιεί την προϋπόθεση

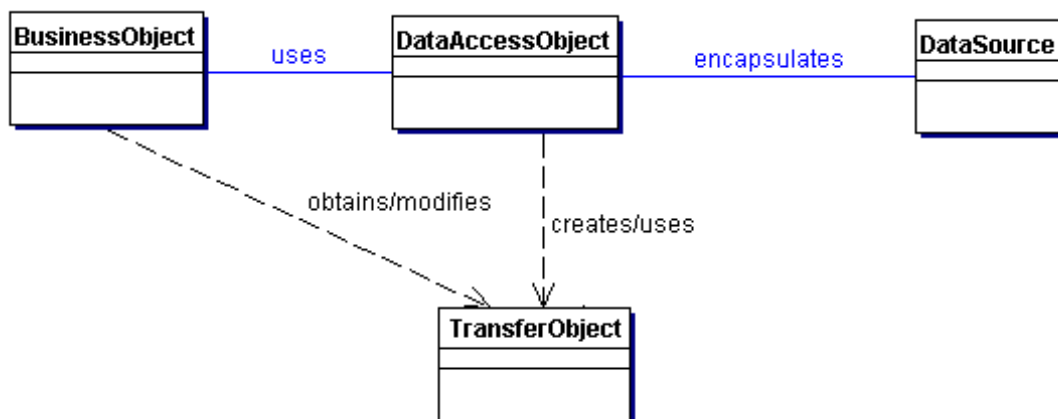
Ενημέρωση:

- UPDATE <table_name> SET <column_name1> = exp1, <column_name2> = exp2,...[WHERE where_condition] : για την ενημέρωση κάποιων στοιχείων ενός πίνακα που μπορεί να ικανοποιούν κάποια προϋπόθεση.

Η MySQL επιτρέπει τη χρήση λογικών και αριθμητικών τελεστών που μπορούν να χρησιμοποιηθούν μέσα σε εκφράσεις.

Data Access Object (DAO)

Το Data Access Object είναι μια τεχνολογία που διευκολύνει την πρόσβαση σε μια πηγή δεδομένων , όπως για παράδειγμα μια βάση δεδομένων, για την ανάκτηση και την αποθήκευση της πληροφορίας. Παρέχει όλους τους μηχανισμούς που χρειάζονται για την πρόσβαση με την πηγή των δεδομένων. Το business object που επικαλείται το DAO, χρησιμοποιεί την απλούστερη διεπαφή που προσφέρεται από το DAO για τους πελάτες της. Το DAO αποκρύπτει εντελώς τις λεπτομέρειες υλοποίησης της πηγής δεδομένων από τους πελάτες της. Ιδανικά, οι αλλαγές που γίνονται στην πηγή δεδομένων, απαιτούν αλλαγές μόνο στα data access objects χωρίς να επηρεάζουν τα business objects. Ουσιαστικά, το DAO λειτουργεί ως έναν διαμεσολαβητή μεταξύ του business object και της πηγής δεδομένων.



Εικόνα 66 - Το διάγραμμα αντιπροσωπεύει τις σχέσεις μεταξύ των αντικειμένων του DAO

Business Object: αντιπροσωπεύει τον client προς τα δεδομένα. Είναι το αντικείμενο που χρειάζεται να έχει πρόσβαση στην πηγή των δεδομένων για την ανάκτηση και ην αποθήκευση των δεδομένων.

Data Access Object: περιγράφει το επίπεδο υλοποίησης πρόσβασης στα δεδομένα έτσι ώστε να επιτρέπει στο business object να έχει πρόσβαση στην πηγή δεδομένων

Data Source: αντιπροσωπεύει την πηγή των δεδομένων. Μπορεί να είναι μια βάση δεδομένων όπως RDBMS, OODBMS, XML repository, flat file system.

Transfer Object: χρησιμοποιείται για την μεταφορά των δεδομένων. Ένα data access object χρησιμοποιεί ένα transfer object για να επιστρέψει τα δεδομένα στον client.

Στην εφαρμογή χρησιμοποιήσαμε το DAO για να εξασφαλίσουμε εύκολη πρόσβαση στη σχεσιακή βάση δεδομένων που υλοποιήσαμε.

Web Service

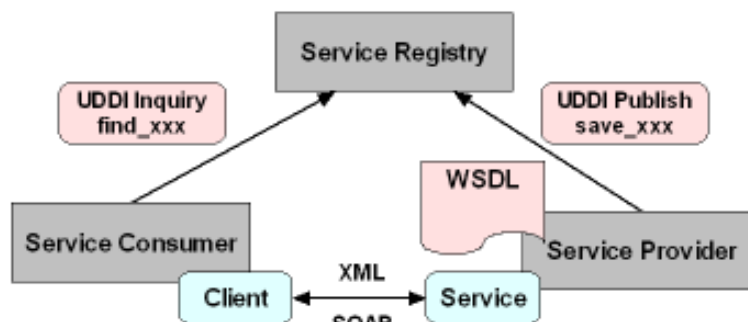
Οι υπηρεσίες παγκόσμιου ιστού (web services) [13] είναι μια τεχνολογία η οποία επιτρέπει σε εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας, γλώσσας προγραμματισμού και λειτουργικού συστήματος. Μια υπηρεσία παγκόσμιου ιστού είναι μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες που προσφέρονται στους χρήστες μέσω του διαδικτύου μέσω προτυποποιημένων μηνυμάτων XML. Χρησιμοποιούν πρότυπα βασισμένα σε XML (XML Schema) για να περιγράψουν μια λειτουργία αλλά και τα δεδομένα προς ανταλλαγή κατά την εκτέλεση της λειτουργίας.

Η αρχιτεκτονική των υπηρεσιών παγκόσμιου ιστού βασίζεται σε τρεις κύριους ρόλους:

1. Ο **Παροχέας Υπηρεσιών (Service Provider)** είναι υπεύθυνος για την ανάπτυξη μιας υπηρεσίας, τη δημιουργία της περιγραφής της και τη δημοσίευση της σε έναν ή περισσότερους καταλόγους (registry).

2. Ο **Καταναλωτής Υπηρεσιών (Service Consumer)** είναι υπεύθυνος για την εύρεση της περιγραφής μιας υπηρεσίας παγκόσμιου ιστού, η οποία έχει δημοσιοποιηθεί σε έναν κατάλογο.

3. Ο **Κατάλογος Υπηρεσιών (Service Registry)** επιτρέπει στους υποψήφιους καταναλωτές να ψάξουν στην συλλογή από περιγραφές υπηρεσιών την οποία διαθέτει και να ανακαλύψουν την υπηρεσία που ταιριάζει με τις απαιτήσεις τους.



Εικόνα 67 - Αρχιτεκτονική των υπηρεσιών ιστού [12]

Όπως φαίνεται στο παραπάνω σχήμα, στην αρχιτεκτονική των υπηρεσιών ιστού τον ρόλο του Καταλόγου Υπηρεσιών έχει η UDDI registry. Οι λειτουργίες για την εγγραφή και την εύρεση των υπηρεσιών υλοποιούνται με χρήση των APIs που διαθέτει η registry για εύρεση (UDDI Inquiry) και δημοσιοποίηση (UDDI Publish). Η περιγραφή της υπηρεσίας γίνεται μέσω ενός WSDL εγγράφου και χρησιμοποιείται για να γίνει η ένωση του πελάτη με την υπηρεσία.

Τέλος, όλες οι διαδικασίες που αφορούν στη μορφοποίηση και τη μεταφορά των μηνυμάτων μεταξύ των υπηρεσιών γίνονται μέσω του πρωτοκόλλου SOAP.

Όταν ο υπεύθυνος για την ανάπτυξη ενός λογισμικού έχει την ανάγκη χρήσης μιας νέας υπηρεσίας, αρχικά βρίσκει την επιθυμητή υπηρεσία, είτε δημιουργώντας και υποβάλλοντας μια ερώτηση προς την registry, είτε με λεπτομερή ψάξιμο και αναζήτηση αυτής.

Ο υπεύθυνος για την ανάπτυξη του λογισμικού ερμηνεύει έπειτα την περιγραφή της διεπαφής της υπηρεσίας (κυρίως μέσω κάποιων χαρακτηριστικών ετικετών, σχολίων, ή κάποιας πρόσθετης τεκμηρίωσης η οποία υπάρχει) και πραγματοποιεί μια κλήση προς την υπηρεσία μέσα στην εφαρμογή που αναπτύσσει. Αυτή η εφαρμογή έχει το ρόλο του Καταναλωτή της υπηρεσίας κατά τη διάρκεια της αλληλεπίδρασης. Σε αυτό το σημείο, ο Καταναλωτής της υπηρεσίας μπορεί αυτόματα να καλέσει την υπηρεσία (που παρέχεται από τον παροχέα υπηρεσιών) χρησιμοποιώντας το πρωτόκολλο επικοινωνίας υπηρεσιών ιστού SOAP .

Στις υποενότητες που ακολουθούν περιγράφουμε με μεγαλύτερη λεπτομέρεια τα πρότυπα UDDI, WSDL και SOAP.

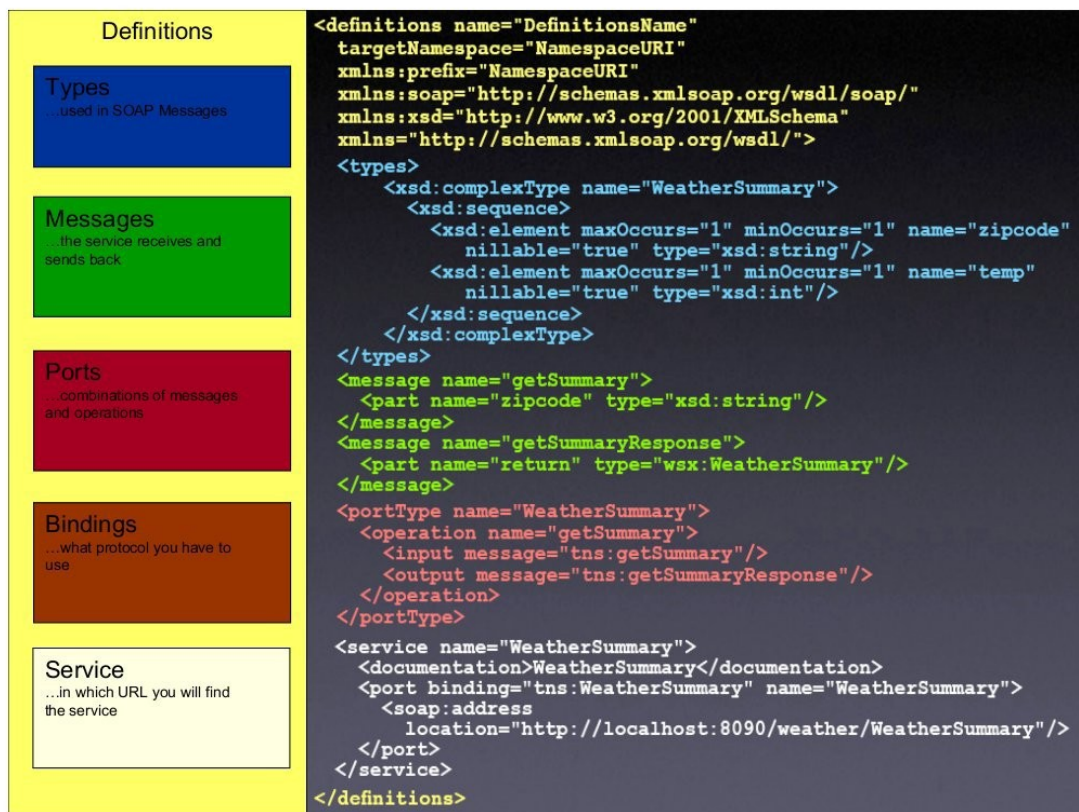
Πρότυπο WSDL

Η WSDL (Web Service Description Language) [14] είναι μια γλώσσα βασισμένη στην XML που προσδιορίζει την γραμματική για την περιγραφή μιας υπηρεσίας παγκόσμιου ιστού (web service). Περιγράφει όλα αυτά που χρειάζεται να ξέρει ένα πρόγραμμα για να “καλέσει” υπηρεσία.

Αυτές οι πληροφορίες αφορούν τα εξής:

1. Τι κάνει μια υπηρεσία
2. Πώς μπορεί να κάποιος να έχει πρόσβαση στην υπηρεσία
3. Που μπορεί κάποιος να βρει την υπηρεσία

Η δομή ενός WSDL εγγράφου περιγράφεται στο παρακάτω σχήμα μαζί με ένα παράδειγμα.



Εικόνα 68 - Δομή WSDL έγγραφου [12]

Όπως φαίνεται και στο παραπάνω σχήμα ένα WSDL έγγραφο περιέχει ένα ριζικό στοιχείο **<definitions>** το οποίο ορίζει το όνομα της υπηρεσίας, περιλαμβάνει πολλαπλά namespaces που θα χρησιμοποιηθούν και περιέχει τα στοιχεία της υπηρεσίας που θα περιγραφούν. Επιπλέον, το ριζικό αυτό στοιχείο περιλαμβάνει τα επόμενα τμήματα:

1. **Types:** Περιγράφει όλους τους τύπους δεδομένων μεταξύ πελάτη και εξυπηρετητή.
2. **Message:** Περιέχει τα μηνύματα που μια υπηρεσία λαμβάνει και στέλνει.
3. **PortType:** Ένας συνδυασμός πολλών μηνυμάτων που σχηματίζουν μια εργασία.
4. **Binding:** Προσδιορίζει το πρωτόκολλο επικοινωνίας που θα χρησιμοποιηθεί
5. **Service:** Περιέχει την διεύθυνση (url) που μπορεί να βρεθεί η υπηρεσία.

Το έγγραφο WSDL του Web Service που υλοποιήσαμε φαίνεται στην παρακάτω εικόνα:

Πρότυπο SOAP

Το SOAP [15] είναι ένα πρωτόκολλο, βασισμένο στην XML, το οποίο χρησιμοποιείται τόσο για μορφοποίηση μηνυμάτων όσο και για ανταλλαγή

δομημένων δεδομένων μέσω του διαδικτύου. Το SOAP (Simple Object Access Protocol) είναι σχεδιασμένο έτσι ώστε να λειτουργεί πάνω από διάφορα πρωτόκολλα επικοινωνίας όπως HTTP, FTP, SMTP και άλλα. Επίσης οι περισσότερες γλώσσες προγραμματισμού υποστηρίζουν την δημιουργία SOAP εξυπηρετητή με πολύ απλό τρόπο και το μόνο που έχει να κάνει ο χρήστης είναι να καθορίσει τις ενέργειες που θα εκτελεστούν όταν έρθει κάποιο μήνυμα.

Η δομή ενός SOAP μηνύματος φαίνεται παρακάτω:



Εικόνα 69 - Δομή SOAP μηνύματος [12]

Όπως φαίνεται από το σχήμα τα κυριότερα μέρη ενός SOAP μηνύματος είναι:

1. **SOAP message:** Είναι ένα XML έγγραφο
2. **Envelope:** Το πραγματικό μήνυμα που θα υποστεί επεξεργασία
3. **Headers:** Προαιρετικό στοιχείο που καθορίζει συγκεκριμένες απαιτήσεις της εφαρμογής

4. **Body:** Το περιεχόμενο του μηνύματος

6.3 Περίληψη

Στο κεφάλαιο αυτό αναφερθήκαμε στην υλοποίηση του συστήματος. Παρουσιάσαμε αναλυτικά όλες τις τεχνολογίες που χρησιμοποιήσαμε καθώς επίσης και τα σχήματα που αναπτύχθηκαν για τις ανάγκες της εφαρμογής.

Κεφάλαιο 7

Αρχιτεκτονική του Συστήματος

7.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε αναλυτικά την αρχιτεκτονική της εφαρμογής δημιουργίας πρότυπων γραφημάτων (Chart Template Builder) καθώς και την αρχιτεκτονική της υπηρεσίας ιστού (Chart Builder Service) που υλοποιήσαμε.

Κατά το σχεδιασμό της εφαρμογής δημιουργίας πρότυπων γραφημάτων, οι βασικοί στόχοι ήταν οι ακόλουθοι: το γραφικό περιβάλλον της εφαρμογής να παρέχει όλη τη βασική λειτουργικότητα του JFreeChart στο χρήστη και να είναι εύχρηστο, η αξιοποίηση των δυνατοτήτων του JFreeChart για τη διευκόλυνση των χρηστών όσον αφορά τις αλλαγές πάνω σε ένα γράφημα και τελευταίο και πιο σημαντικό ο σχεδιασμός να γίνει με τέτοιο τρόπο ώστε η εφαρμογή να είναι ανεξάρτητη από συγκεκριμένα περιβάλλοντα (Platform Independent).

Στις ακόλουθες παραγράφους θα παρουσιάσουμε πρώτα την αρχιτεκτονική της εφαρμογής δημιουργίας πρότυπων γραφημάτων, δηλαδή τον τρόπο με τον οποίο είναι οργανωμένο το σύστημα που αναπτύξαμε και τον τρόπο με τον οποίο αλληλεπιδρούν τα διάφορα τμήματα μεταξύ τους. Στη συνέχεια, θα παρουσιάσουμε την αρχιτεκτονική της υπηρεσίας ιστού, δηλαδή τι δέχεται ως είσοδο (request), τι επιστρέφει (response) και τον τρόπο ανακατασκευής του τελικού γραφήματος.

7.2 Γενική Αρχιτεκτονική

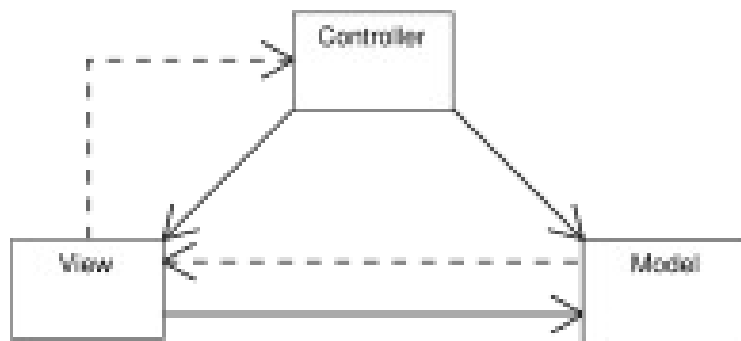
7.2.1 Το πρότυπο MVC

Η αρχιτεκτονική του συστήματος σχεδιάστηκε σύμφωνα με το μοντέλο Model-View-Controller (MVC) [9], το οποίο αποτελεί ένα αρχιτεκτονικό σχέδιο που χρησιμοποιείται από την τεχνολογία λογισμικού. Ο υπεύθυνος για τη σχεδίαση και την ανάπτυξη μιας εφαρμογής επιθυμεί συχνά να χωρίσει τα δεδομένα (μοντέλα) από τις γραφικές διεπαφές που φαίνονται στο χρήστη, έτσι ώστε οποιεσδήποτε αλλαγές γίνονται στο γραφικό περιβάλλον, να μην έχουν επιπτώσεις στο στάδιο της

επεξεργασίας των δεδομένων, αλλά και η αναδιοργάνωση των δεδομένων να μην επηρεάζει το γραφικό μέρος της εφαρμογής. Το MVC λύνει αυτό το πρόβλημα με την αποσύνδεση της πρόσβασης και των ενεργειών που γίνονται πάνω στα δεδομένα, από την παρουσίαση των στοιχείων και την αλληλεπίδραση με τους χρήστες, εισάγοντας ένα ενδιάμεσο επίπεδο (layer): τον ελεγκτή (controller). Έτσι λοιπόν, σύμφωνα με το MVC έχουμε:

- **Model:** Περιλαμβάνει την αναπαράσταση της πληροφορίας την οποία χρησιμοποιεί η εφαρμογή, όπως επίσης και τη λογική που χρησιμοποιείται για τη διαχείριση και επεξεργασία αυτών των δεδομένων. Τέλος, πολλά συστήματα χρησιμοποιούν μηχανισμούς αποθήκευσης της πληροφορίας (persistence layer), όπως βάσεις δεδομένων, ή αρχεία σε τοπικούς ή απομακρυσμένους δίσκους.
- **View:** Περιλαμβάνει τις γραφικές διεπαφές που χρησιμοποιούνται για την αλληλεπίδραση του χρήστη με την εφαρμογή.
- **Controller:** Επεξεργάζεται και ανταποκρίνεται στα γεγονότα, όπως οι ενέργειες των χρηστών, και μπορεί να προκαλέσει αλλαγές στις διάφορες πληροφορίες του συστήματος.

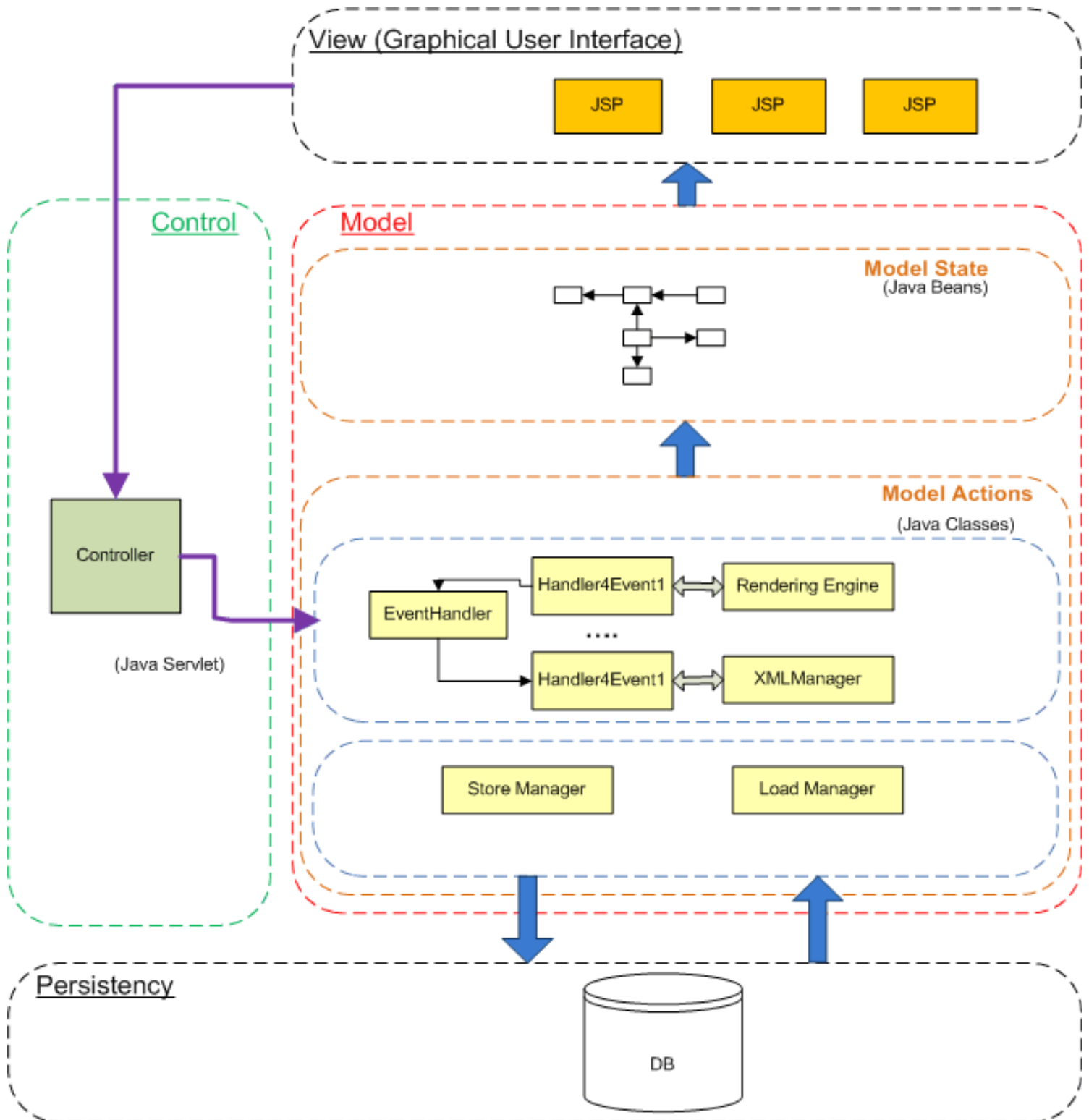
Στην επόμενη εικόνα, φαίνεται ένα απλό διάγραμμα που δείχνει τη σχέση μεταξύ των παραπάνω τμημάτων που αναφέραμε:



Εικόνα 70 - Πρότυπο MVC

7.2.2 Αρχιτεκτονική της εφαρμογής δημιουργίας πρότυπων γραφημάτων

Στην επόμενη εικόνα (75) παρουσιάζεται η γενική αρχιτεκτονική της εφαρμογής δημιουργίας πρότυπων γραφημάτων (Chart Template Builder), σύμφωνα με το πρότυπο MVC , έτσι όπως το σχεδιάσαμε:



Εικόνα 71 - Αρχιτεκτονική του συστήματος σύμφωνα με το πρότυπο MVC

7.2.2.1 Επίπεδο Graphical User Interface

Στο επίπεδο του GUI (Graphical User Interface), το οποίο αποτελεί το View σύμφωνα με το πρότυπο MVC, περιέχονται όλα τα JSPs που αποτελούν όλες τις γραφικές διεπαφές του συστήματος. Ο εκάστοτε χρήστης χρησιμοποιεί τις διεπαφές αυτές για να αλληλεπιδράσει με το σύστημα.

7.2.2.2 Controller

Ο Controller αποτελεί το ενδιάμεσο στοιχείο μεταξύ της γραφικής διεπαφής και των μοντέλων που περιλαμβάνει το σύστημα. Είναι υπεύθυνος για την επεξεργασία και εκτέλεση των εντολών που δίνονται από τον χρήστη. Για την τελική εκτέλεση των ενεργειών επικοινωνεί με τον Event Handler, ο οποίος ανάλογα με το συμβάν (event) που έχει ζητηθεί ενεργοποιεί τον κατάλληλο handler4Event.

7.2.2.3 Τα μοντέλα πληροφορίας του συστήματος

Το επίπεδο Model State αναπαριστά την τρέχουσα τιμή της πληροφορίας της εφαρμογής και περιλαμβάνει μεθόδους για την αλλαγή της τιμής αυτής. Για το λόγο αυτό, το επίπεδο Model State αποτελείται από τα Java Beans της εφαρμογής. Τα Java Beans δεν είναι τίποτα άλλο από Java κλάσεις που περιέχουν get() και set() συναρτήσεις. Τα Java Beans χρησιμοποιούνται από τα JSP είτε για να ορίσουμε είτε για να ανακτήσουμε την τιμή ενός αντικειμένου.

Το επίπεδο Model Action, αποτελείται από τον Event Handler ο οποίος επικοινωνεί με τον controller και αποκρίνεται σε κάθε νέο συμβάν. Όταν δημιουργηθεί ένα νέο συμβάν, ο Event Handler επικοινωνεί με τον κατάλληλο handler για να διαχειριστεί αυτό το συμβάν. Κάθε handler επικοινωνεί με το Rendering Engine και τον XML Manager. Το Rendering Engine είναι υπεύθυνο για την εφαρμογή των αλλαγών πάνω σε ένα γράφημα. Κάθε φορά που ο χρήστης υποβάλει κάποιες αλλαγές πάνω στο γράφημα, το Rendering Engine δέχεται αυτές τις αλλαγές ως είσοδο, τις εφαρμόζει πάνω στο γράφημα, χρησιμοποιώντας τις συναρτήσεις της βιβλιοθήκης JFreeChart, και επιστρέφει το νέο γράφημα. Η μονάδα XML Manager, είναι υπεύθυνη για την αποθήκευση των παραμέτρων ενός γραφήματος σε ένα xml αρχείο. Όλες οι παράμετροι ενός γραφήματος που τροποποιούνται ή προστίθενται από τον χρήστη, αποθηκεύονται από τον XML

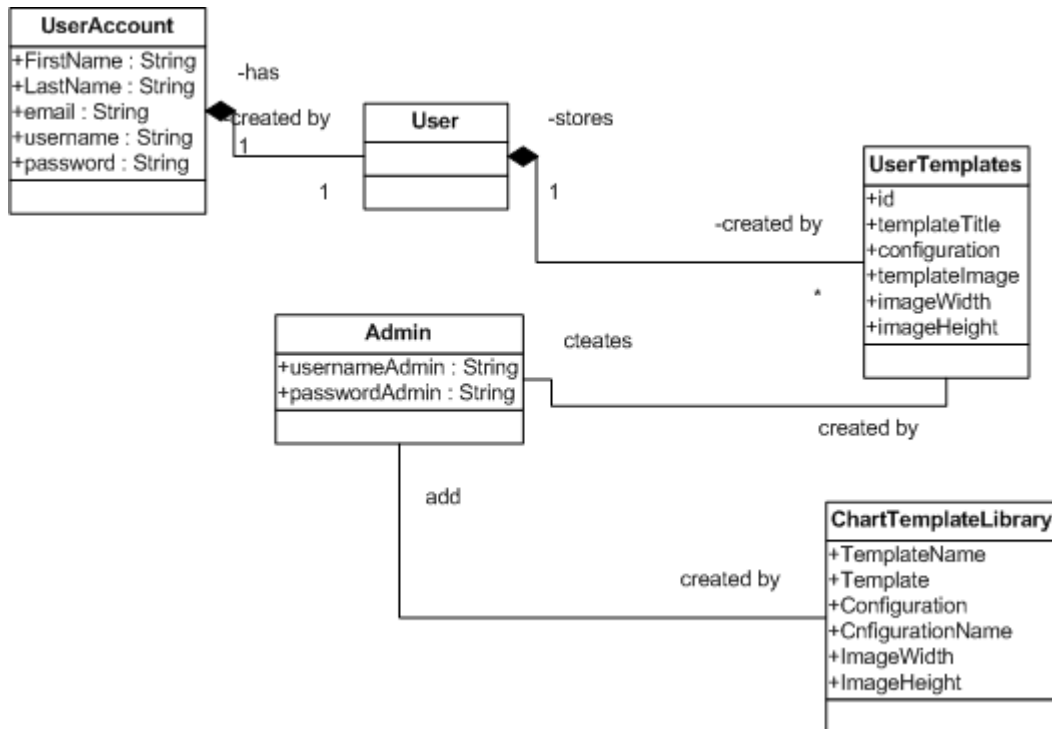
Manager με τη βοήθεια του XMLBeans σε ένα xml έγγραφο, ώστε να είναι δυνατή η ανάκτηση του γραφήματος ανά πάσα στιγμή.

Για παράδειγμα, έστω ότι ο χρήστης της εφαρμογής θέλει να επεξεργαστεί τις παραμέτρους του τίτλου ενός γραφήματος που έχει δημιουργήσει. Όταν ο χρήστης επιλέξει από το βασικό μενού επεξεργασία των παραμέτρων του τίτλου, ο Controller επικοινωνεί με τον Event Handler για να διαχειριστεί το συμβάν, που στην περίπτωση αυτή είναι ο τίτλος του γραφήματος. Στη συνέχεια, ο Event Handler επικοινωνεί με τον κατάλληλο handler που είναι υπεύθυνος να διαχειριστεί τον τίτλο του γραφήματος. Ο handler αυτός επικοινωνεί με το Rendering Engine για την εφαρμογή των αλλαγών πάνω στο γράφημα και με τον XML Manager για την αποθήκευση των παραμέτρων του τίτλου του γραφήματος στο xml έγγραφο. Σε περίπτωση που έχει επεξεργαστεί τις παραμέτρους του τίτλου για περισσότερες από μια φορές, τότε γίνεται απλά μια ενημέρωση των τιμών τους στο xml έγγραφο.

Όταν ο χρήστης τελειώσει με τις αλλαγές που θέλει να κάνει και επιλέξει αποθήκευση, τότε το xml έγγραφο καθώς και μια εικόνα με το τελικό γράφημα, αποθηκεύονται στη βάση δεδομένων. Επίσης, από το χρήστη μπορεί να ζητηθεί η ανάκτηση ενός γραφήματος για περαιτέρω επεξεργασία. Και στις δυο περιπτώσεις, ο αντίστοιχος handler επικοινωνεί με τον Load Manager/ Store Manager αντίστοιχα για να αποθηκεύσει ή να ανακτήσει τα δεδομένα από τη βάση δεδομένων.

7.2.2.4 Επίπεδο Persistency

Αποτελείται από τη βάση δεδομένων της εφαρμογής. Η βάση δεδομένων σχεδιάστηκε για την υποστήριξη χρηστών της εφαρμογής. Για όλους τους χρήστες που είναι εγγεγραμμένοι, αποθηκεύονται τα προσωπικά τους στοιχεία όπως όνομα, επίθετο, email καθώς και το επιθυμητό username και password. Για κάθε χρήστη αποθηκεύονται όλα τα πρότυπα γραφήματα που έχει δημιουργήσει. Για κάθε ένα από αυτά, αποθηκεύεται το όνομα του, οι διαστάσεις του, μια εικόνα με το γράφημα και ένα xml αρχείο με τις παραμέτρους του. Τέλος να σημειώσουμε ότι υπάρχει και ένας Διαχειριστής, ο οποίος είναι υπεύθυνος για την αποδοχή ή απόρριψη νέων λογαριασμών και μπορεί να αποθηκεύει πρότυπα γραφήματα στην βιβλιοθήκη γραφημάτων. Στην παρακάτω εικόνα φαίνεται το σχήμα της βάσης δεδομένων:



Εικόνα 72 – Διάγραμμα κλάσεων της βάσης δεδομένων

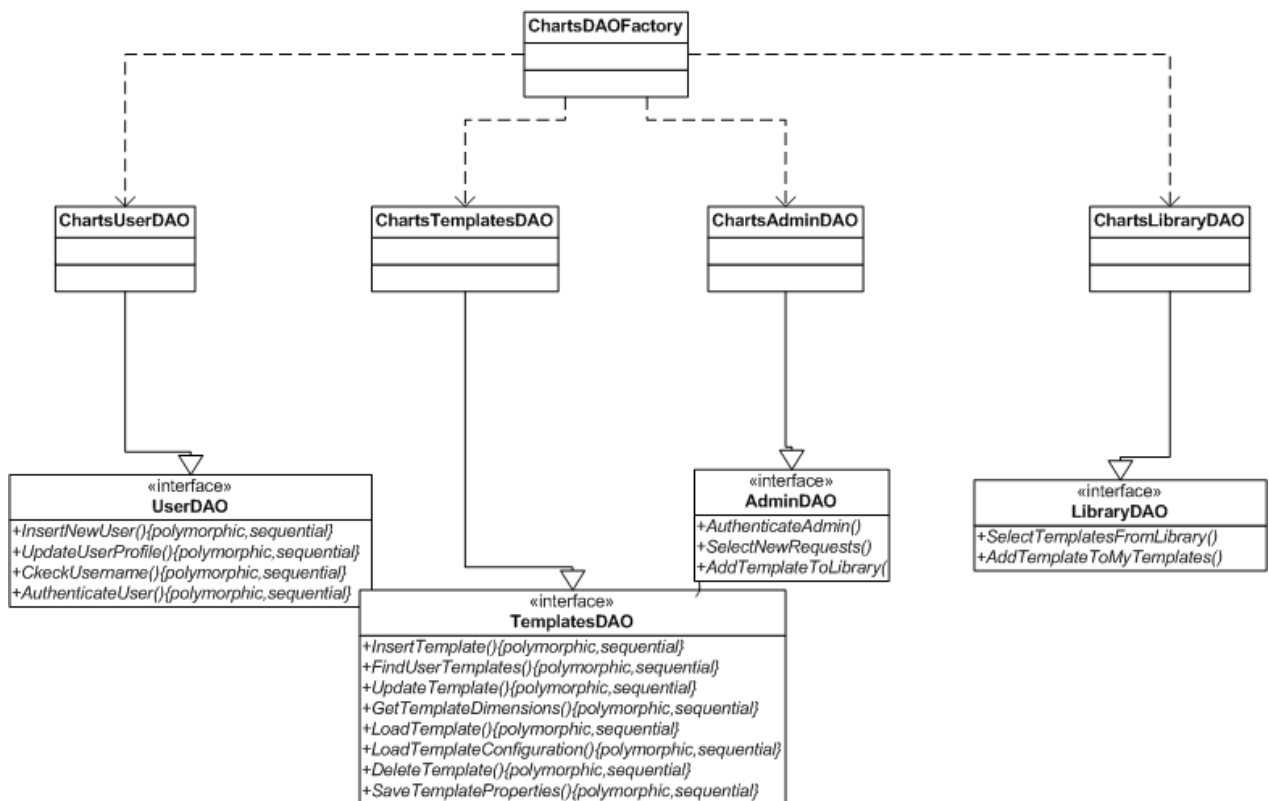
7.2.2.4.1 Λειτουργικότητα της σχεσιακής βάσης δεδομένων

Μπορούμε να συνοψίσουμε την λειτουργικότητα της σχεσιακής βάσης δεδομένων στα ακόλουθα:

- 1.Εισαγωγή νέου χρήστη
- 2.Ενημέρωση των προσωπικών στοιχείων του χρήστη(password, email)
- 3.Ανάκτηση των προσωπικών στοιχείων ενός χρήστη(username, password)
5. Έλεγχος των στοιχείων ενός χρήστη
- 6.Δημιουργία νέου πρότυπου γραφήματος για έναν χρήστη
- 7.Επεξεργασία ενός πρότυπου γραφήματος ενός χρήστη
- 8.Ανάκτηση όλων των πρότυπων γραφημάτων ενός χρήστη
- 9.Ανακτηση ενός συγκεκριμένου πρότυπου γραφήματος ενός χρήστη
- 10.Ανάκτηση των διαστάσεων ενός πρότυπου γραφήματος ενός χρήστη
- 11.Ανάκτηση σημαντικών πληροφοριών ενός πρότυπου γραφήματος
- 12.Ανάκτηση όλων των πρότυπων γραφημάτων από την βιβλιοθήκη γραφημάτων
- 13.Προσθήκη ενός πρότυπου γραφήματος από την βιβλιοθήκη στα πρότυπα γραφήματα ενός χρήστη
14. Έλεγχος των στοιχείων του Διαχειριστή

15. Προσθήκη ενός πρότυπου γραφήματος στην βιβλιοθήκη γραφημάτων από τον Διαχειριστή
16. Διαγραφή ενός πρότυπου γραφήματος ενός χρήστη
17. Αποθήκευση των παραμέτρων ενός πρότυπου γραφήματος

Για να επιτευχθεί η απαραίτητη λειτουργικότητα, χρειάζεται πρόσβαση στην βάση δεδομένων με σκοπό την ανάκτηση ή/και την αποθήκευση πληροφορίας. Ο τρόπος πρόσβασης στην βάση δεδομένων σχεδιάστηκε σύμφωνα με το σχεδιαστικό μοντέλο Data Access Object [22] και φαίνεται στην επόμενη εικόνα:



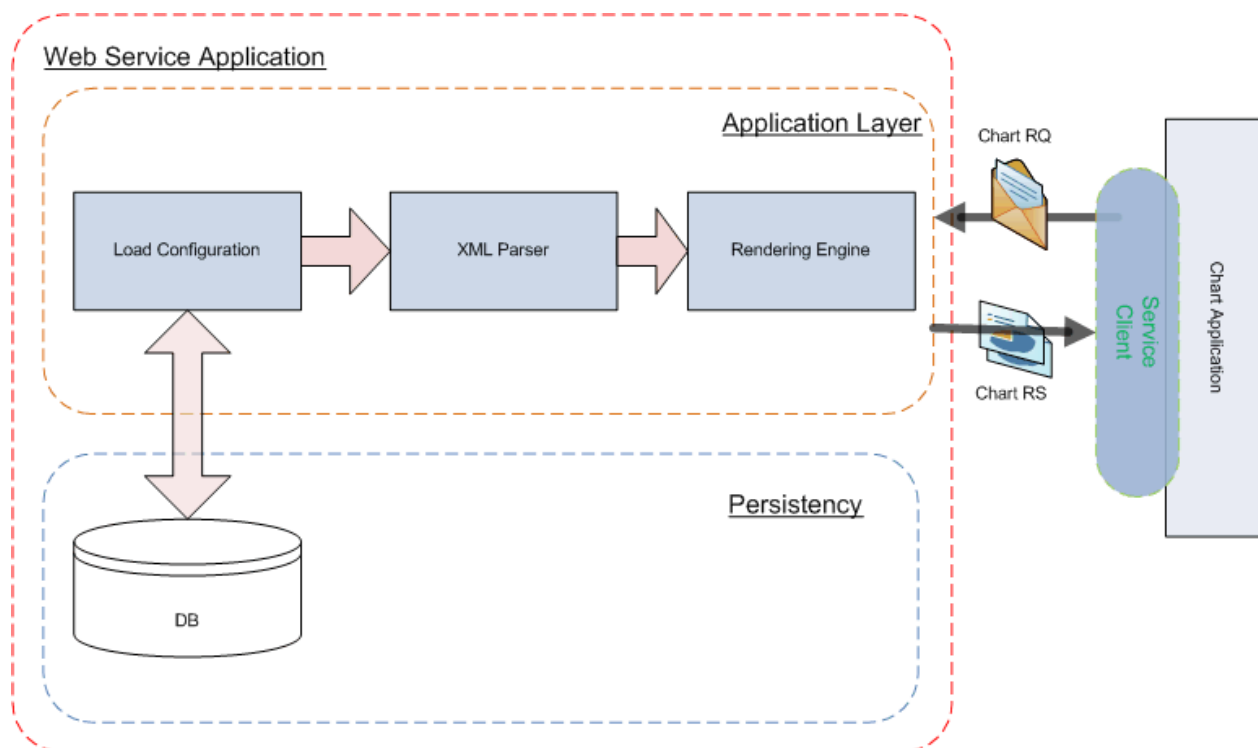
Εικόνα 73 - DAO pattern για τη σχεσιακή βάση δεδομένων

7.2.3 Αρχιτεκτονική της υπηρεσίας ιστού (Web Service)

Η υπηρεσία ιστού (Web Service) που σχεδιάσαμε, την οποία ονομάσαμε Chart Builder Service, μπορεί να χρησιμοποιηθεί από οποιαδήποτε εφαρμογή-πελάτη (Chart Application) για την απεικόνιση πραγματικών δεδομένων πάνω σε ένα πρότυπο γράφημα. Για να μπορέσει η εφαρμογή-πελάτης να καλέσει το web service, χρειάζεται τα εξής: ένα κατάλληλο run-time περιβάλλον ώστε να δημιουργήσει έναν Service Client που θα καλέσει το web service και το κατάλληλο API που χρειάζεται

για να δημιουργήσει το xml έγγραφο με τα δεδομένα. Το API αυτό είναι διαθέσιμο από την ιστοσελίδα της εφαρμογής.

Η αρχιτεκτονική του συστήματος που σχεδιάσαμε φαίνεται στην επόμενη εικόνα .



Εικόνα 74 - Αρχιτεκτονική συστήματος για την υπηρεσία ιστού

7.2.3.1 Chart Application, Service Client, Chart RQ, Chart RS

Με τον όρο Chart Application, αναφερόμαστε σε οποιαδήποτε εφαρμογή χρειάζεται να χρησιμοποιήσει το web service για αναπαράσταση πραγματικών δεδομένων πάνω σε ένα πρότυπο γράφημα. Για να μπορέσει να το κάνει αυτό, η εφαρμογή χρειάζεται να κάνει τα εξής δυο βήματα: πρώτον να χρησιμοποιήσει ένα κατάλληλο run-time περιβάλλον (service client), ώστε να μπορέσει να καλέσει το web service και δεύτερον να έχει στην διάθεση της το απαραίτητο service specific API , το οποίο είναι απαραίτητο για τη δημιουργία του αρχείου με τα δεδομένα(data binding).

Το Chart RQ(Request) απεικονίζει την είσοδο στο web service, που στην περίπτωση μας πρέπει να είναι το αρχείο xml με τα δεδομένα. Όπως

αναφέραμε και παραπάνω για να μπορέσει να δημιουργηθεί το xml αρχείο με τα δεδομένα, η εφαρμογή-πελάτης έχει στη διάθεση της το απαραίτητο API. Από εκεί και έπειτα χρειάζεται να προσθέσει το username του χρήστη που έχει δημιουργήσει το πρότυπο γράφημα, τον τίτλο του προτύπου γραφήματος και τα δεδομένα. Ο τρόπος εισαγωγής των δεδομένων ποικίλει σύμφωνα με τον τύπο του γραφήματος επιλογής.

Αντίστοιχα, το Chart RS(Response) απεικονίζει την έξοδο του Web Service που στην περίπτωσή μας είναι ένα xml αρχείο που περιέχει το όνομα της εικόνας και τα binary data αυτής. Για να μπορέσει να δει την τελική εικόνα η εφαρμογή, χρειάζεται να γράψει σε ένα αρχείο τα binary data της εικόνας που λαμβάνει , και να αποθηκεύσει με το όνομα της.

Η εφαρμογή μπορεί να χρησιμοποιήσει την εικόνα με το τελικό γράφημα σε οποιαδήποτε εφαρμογή θέλει, όπως για παράδειγμα να προσθέσει την εικόνα σε μια δική του εφαρμογή στο web. Επιπλέον, μπορεί να καλέσει το web service για απεικόνιση διαφορετικών δεδομένων πάνω στο ίδιο πρότυπο γράφημα.

7.2.3.2 Επίπεδο Application

Όπως έχουμε αναφέρει παραπάνω, το web service που σχεδιάσαμε δέχεται ως είσοδο ένα xml αρχείο, το οποίο περιλαμβάνει τα δεδομένα προς απεικόνιση , το username του χρήστη που έχει δημιουργήσει το πρότυπο γράφημα και το όνομα του γραφήματος , και επιστρέφει στην εφαρμογή – πελάτη μια εικόνα με το τελικό γράφημα. Για να γίνει όμως αυτό, έχουμε σχεδιάσει μια αρχιτεκτονική ανακατασκευής του γραφήματος από τις παραμέτρους του.

Το επίπεδο Application περιλαμβάνει τις βασικές μονάδες για την ανακατασκευή του γραφήματος. Κάθε φορά που καλείται το Web Service, η μονάδα Load Configuration είναι υπεύθυνη να επικοινωνήσει με τη βάση δεδομένων για να βρει το σχετικό πρότυπο γράφημα με βάση το username και το όνομα του γραφήματος από το xml αρχείο που δέχεται ως είσοδο το web Service. Από εκεί και έπειτα η μονάδα XML Parser είναι υπεύθυνη για δυο πράγματα: να ανακτήσει τις παραμέτρους από το xml αρχείο του προτύπου γραφήματος που είναι αποθηκευμένο στη βάση δεδομένων και να ανακτήσει τα δεδομένα από το xml αρχείο που δέχεται το service ως είσοδο. Τέλος, η μονάδα Rendering Engine δέχεται

ως είσοδο τα δεδομένα από τη μονάδα XML Parser και ανακατασκευάζει πλήρως το γράφημα. Το τελικό αποτέλεσμα είναι να επιστρέφεται μια εικόνα με το τελικό γράφημα.

Στην παρακάτω εικόνα φαίνεται το WSDL έγγραφο του Web Service που σχεδιάσαμε:

```
- <definitions name="ChartBuilder" targetNamespace="http://service.charttemplates.org">
- <types>
- <xsd:schema targetNamespace="http://service.charttemplates.org">
- <xsd:import namespace="http://serviceResponse.charttemplates.org" schemaLocation="ChartBuilderService?xsd=xsd0.xsd"/>
- <xsd:import namespace="http://properties.charttemplates.org" schemaLocation="ChartBuilderService?xsd=xsd2.xsd"/>
- </xsd:schema>
- </types>
- <message name="ChartBuilderOperationRequest">
- <part name="input" element="ns1:chartProperties"> </part>
- </message>
- <message name="ChartBuilderOperationResponse">
- <part name="output" element="ns0:AttachmentRequest"> </part>
- </message>
- <portType name="ChartBuilderPortType">
- <operation name="ChartBuilderOperation">
- <input name="input1" message="tns:ChartBuilderOperationRequest"> </input>
- <output name="output1" message="tns:ChartBuilderOperationResponse"> </output>
- </operation>
- </portType>
- <binding name="ChartBuilderBindingSOAP11" type="tns:ChartBuilderPortType">
- <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
- <operation name="ChartBuilderOperation">
- <soap:operation soapAction="attachment"/>
- <input name="input1">
- <soap:body use="literal"/>
- </input>
- <output name="output1">
- <soap:body use="literal"/>
- </output>
- </operation>
- </binding>
- <binding name="ChartBuilderBindingSOAP12" type="tns:ChartBuilderPortType">
- <soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
- <operation name="ChartBuilderOperation">
- <soap12:operation soapAction="attachment"/>
- <input name="input1">
- <soap12:body use="literal"/>
- </input>
- <output name="output1">
- <soap12:body use="literal"/>
- </output>
- </operation>
- </binding>
- <service name="ChartBuilderService">
- <port name="ChartBuilderPortSOAP12" binding="tns:ChartBuilderBindingSOAP12">
- <soap:address location="http://147.27.4.195:8084/axis2/services/ChartBuilderService"/>
- </port>
- <port name="ChartBuilderPortSOAP11" binding="tns:ChartBuilderBindingSOAP11">
- <soap:address location="http://147.27.4.195:8084/axis2/services/ChartBuilderService"/>
- </port>
- </service>
</definitions>
```

Εικόνα 75 - WSDL έγγραφο του Web Service που σχεδιάσαμε

Κεφάλαιο 7 – Αρχιτεκτονική συστήματος

Στις επόμενες εικόνες παρουσιάζονται τα SOAP μηνύματα που ανταλλάσσονται κατά την κλήση του Service:

```
<?xml version='1.0' encoding='utf-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body><prop:chartProperties xmlns:prop="http://properties.charttemplates.org">
    <prop:propertiesFilename>Bar3DProperties</prop:propertiesFilename>
    <prop:username>dimitra</prop:username>
    <prop:chartTitle>Chart Title</prop:chartTitle>
    <prop:xAxisLabel>Common Axis Label</prop:xAxisLabel>
    <prop:yAxisLabel>Y-axis Label</prop:yAxisLabel>
    <prop:dataset>
      <prop:BarDataset>
        <prop:seriesName>Series 5</prop:seriesName>
        <prop:columnName>First Column</prop:columnName>
        <prop:value>12.3</prop:value>
      </prop:BarDataset>
      <prop:BarDataset>
        <prop:seriesName>Series 6</prop:seriesName>
        <prop:columnName>Second Column</prop:columnName>
        <prop:value>26.3</prop:value>
      </prop:BarDataset>
      <prop:BarDataset>
        <prop:seriesName>Series 7</prop:seriesName>
        <prop:columnName>First Column</prop:columnName>
        <prop:value>16.4</prop:value>
      </prop:BarDataset>
      <prop:BarDataset>
        <prop:seriesName>Series 8</prop:seriesName>
        <prop:columnName>Second Column</prop:columnName>
        <prop:value>22.3</prop:value>
      </prop:BarDataset>
    </prop:dataset>
  </prop:chartProperties>
</soapenv:Body>
</soapenv:Envelope>
```

Εικόνα 76 - SOAP μήνυμα κατά την κλήση του Web Service με είσοδο ένα xml αρχείο

Κεφάλαιο 7 – Αρχιτεκτονική συστήματος

```
<?xml version='1.0' encoding='utf-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <AttachmentRequest xmlns="http://serviceResponse.charttemplates.org">
      <fileName>Bar3DService.jpg</fileName>
      <binaryData xmlns:axis2ns1="http://www.w3.org/2005/05/xmlmime" axis2ns1:contentType="image/jpeg">
/9j/4AAQSkZJRgABAgAAQABAAD/2wBDAAIBAQEBAQIBAQECAGICAgQDAgICAgUEBAMEBgUGBgYFBgYGBwIBGcJBWYGCAsICQoKC
goKBggLDAsKDAkKCgr/2wBDAAICAgICAgUDAwUKBwYHCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCg
oKCgoKCgr/wAARCAQQAQDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQ
AAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZ
WmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLTlNXW19jZ2uHi4+Tl5
ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBS
ExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN
0dXZ3eH16goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+
Pn6/9oADAMBAAIRAxEAPwD9uKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuE/4a1/Zj/wCF3f8ADM3/AA0
Z4D/4WR/0T7/hL7L+3P8Aj2+1/wDHj5vn/wDht+/+5/qvn+7zXhH/AAxQ/aN+J37KP/BJ74zfGz4N6r/Z/iS00G00zTdTSeaKaw/tL
ULXT2LqCSCSOSK5iju3lhkVvkmjjYhgpU/nnoH/AAbnf8Ev/iR/wRx+F3x28U/F7/hTXjPKPAfhvxt4i+N3izxGZrGOBUYYZpLS6tb
m5t7JbYteC3h8swTBo7YtLMfNWcA/ZHV/j58CvD/xe0r9n3X/AI0+ErHx7rtg19ongi88R2sWr6haqJmaeCzaQT8xgW9wS6oVagk5+
RsdBx4of8FKvBH7St3/AMHLX7Mfw3/Zu+NVhp/xDL/Z5bTbP4i+ONFhvGtLMXiik91V7S1jht571bb7RPFci4rZ7kRI6JCWUes/8ExP
+Ch3/BRr4haf+258Dvi/8c/hr478Xfs3+LTP3hH4ifEuwj8KaQsa3GqWtZd6odMi2IrKFNL+2CNU8zLyxPdKjJLAAfqtRX4of8E5f+C
uH7znjn/gSR8Nf2Rr39u2/wD2gfg98R/CWqXNP4x8R/AW18Fx30lvZahcG70pIglxNHFc6a9m002I2kW9jEOYop680/Zo/aG/4LMf8F
Rv+CPX7Q37Rvxt/wCCiWg6H4M8Da4tsNX0a3+EelXOp+MLEPw/FdXmm3EyRwRafbCF0SG4t0a4L310ZDiG3FAH7ufDP4p/DH41eCLL
4mfBv4jad4t8N6n5n5m+IPDOrw39jd+XK8Un1zwM0cm2N0baxwyMp5BFea+Kf+CkP/AATv8DeJ9S8E+Nv29fgvo+s6PfzW0r6Rqvx8
0i3urG6icxwYTR8XAeKRHV1ZGAZWUgEV8ff8Gr3gd43eHf+CwvhtX146/ab/wCEi8GeIvt3/CB+A/8AhFLa0/4RD7PrerR3v+mxeZd
Q+1TFZv3wHk7di2ZBr5M/4OqP+C237Ef7NF7Mdv+2P8E/gp/yvxi8f/HdP+Eu8R/8ACSalc/b/ALdZavfXX+jz3LwRb7mKOT93Gm3btX
apKkA/XS3/AOCkP/BO+78MXnja0/b1+C8ujaff21jf6vH8UtIa1trq4seS3gklFxsSSVLW52EJD0tvKVBEBY9K/wCFp/DH/hZ3/Ck/+
Fjad/wmf9g/25/wiP8Aa8P9p/2Z53kfbvsv7zfs3nfuv027N/y5zx43f8ABeL9jf8AZT/4I6/8E77Lxd+wF+z34S8OXfjfn40aDZ+J4
vGWixem7W6jttI154G28Ri/hgkQyygSxIkM2R1LFWIqh+zn8IP2pdZ/4Oz/AI+ad4U/bC/sa/0nQbPxB4m1L/hX1jc/294WMvhyDPD
</binaryData>
    </AttachmentRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Εικόνα 77 - SOAP μήνυμα κατά την έξοδο του Web Service

7.2.3.3 Επίπεδο Persistency

Αποτελείται από την ίδια σχεσιακή βάση δεδομένων που περιγράψαμε παραπάνω. Η λειτουργικότητα όμως αλλάζει και συνοψίζεται στα ακόλουθα:

1. Ανάκτηση των παραμέτρων ενός πρότυπου γραφήματος

7.3 Περίληψη

Στο κεφάλαιο αυτό παρουσιάσαμε την αρχιτεκτονική του συστήματος που σχεδιάσαμε. Αρχικά παρουσιάσαμε την αρχιτεκτονική της εφαρμογής Chart Template Builder (δημιουργίας template γραφημάτων) , την οποία αναπτύξαμε σύμφωνα με το πρότυπο MVC. Στη συνέχεια παρουσιάσαμε την αρχιτεκτονική του Web Service σύμφωνα με την οποία γίνεται η ανακατασκευή του τελικού γραφήματος.

Κεφάλαιο 8

Αρχές και οδηγίες σωστού σχεδιασμού γραφημάτων – Ψηφιακή βιβλιοθήκη

8.1 Εισαγωγή

Οι αρχές και οι πρακτικές για το σχεδιασμό γραφημάτων που θα παρουσιαστούν στηρίζονται στα ακόλουθα ζητήματα:

Πρόβλημα: το πρόβλημα είναι το να παρουσιαστούν μεγάλες ποσότητες πληροφορίας με έναν τρόπο που να είναι εύχρηστος, ακριβής, επαρκής για το συγκεκριμένο σκοπό και να γίνεται εύκολα κατανοητός. Πιο συγκεκριμένα, να φαίνεται στο σχεδιασμό η αιτία και το αποτέλεσμα, να διασφαλίζεται ότι οι απαραίτητες συγκρίσεις έχουν γίνει και να επιτυγχάνονται οι επιθυμητοί στόχοι.

Η σπουδαιότητα του: Η απεικόνιση πληροφορίας σε ένα γράφημα είναι τώρα η κινητήρια δύναμη πίσω από τις όλες ζωές μας. Δεν είναι πλέον περιορισμένο σε ειδικευμένους εργαζομένους σε επιλεγμένους τομείς αλλά προσκρούει σχεδόν σε όλους τους ανθρώπους μέσω της διαδεδομένης χρήσης των υπολογιστών και του Διαδικτύου. Οι γρήγορες και ακριβείς μεταφορές των πληροφοριών μπορούν να είναι ένα θέμα ζωτικής σημασίας για πολλούς ανθρώπους. Ο βαθμός στον οποίο τα σύμβολα και η γραφική παράσταση έχουν επιπτώσεις στις ζωές μας μπορεί να φανεί από τη δραματική αύξηση στα αποτελέσματα δείκτη νοημοσύνης (IQ) σε όλους τους πολιτισμούς που έχουν αποκτήσει την τεχνολογία των πληροφοριών: στις Ηνωμένες Πολιτείες έχει υπάρξει μια μέση ανάπτυξη 3 σημείων του IQ ανά δεκαετία τα τελευταία 60 έτη. Δεν υπάρχει καμία γνωστή επιστημονική εξήγηση για αυτήν την αύξηση και η πλέον πιθανή αιτία είναι διαδεδομένη έκθεση σε κείμενο, σύμβολα, και τη γραφική παράσταση που συνοδεύουν τη σύγχρονη ζωή.

Η εφαρμογή του: Ένα μέρος της πληροφορίας σχετίζεται με την απεικόνιση στατιστικών δεδομένων, αλλά και αρκετό από το υπόλοιπο σχετίζεται με οποιοδήποτε τύπο πληροφορίας, ακόμη και απλό κείμενο.

Η επίλυση: είναι να αναπτυχθεί μια συνεπής προσέγγιση στην γραφική απεικόνιση πληροφορίας , που θα τονίζει την ακρίβεια και την εύκολη κατανόηση της.

Τα γραφήματα είναι ένας αποδοτικός τρόπος για την απεικόνιση αριθμών και γενικά δεδομένων. Επιπλέον, από όλες τις μεθόδους ανάλυσης στατιστικής πληροφορίας , τα καλά σχεδιασμένα γραφήματα είναι συνήθως ο πιο απλός και αποδοτικός τρόπος απεικόνισης δεδομένων. Για το σκοπό αυτό, θα παρουσιάσουμε τις γενικές αρχές σχεδιασμού γραφημάτων που πρέπει να ακολουθεί ένα γράφημα.

Οι αρχές αυτές αφορούν κάποιες πρακτικές σχεδιασμού που πρέπει να ακολουθούνται καθώς και την θεωρία των γραφημάτων δεδομένων.

Οι πρακτικές σχεδιασμού αναφέρονται στην διάκριση και την ακεραιότητα που πρέπει να έχει ένα γράφημα και παρουσιάζονται αναλυτικά σε επόμενα κεφάλαια. Η θεωρία των γραφημάτων δεδομένων αφορά γενικές αρχές που πρέπει να ακολουθούνται στον τρόπο αναπαράστασης των δεδομένων, στο συνδυασμό χρωμάτων για την απεικόνιση δεδομένων κ.ά.

Οι αρχές σωστού σχεδιασμού που θα παρουσιαστούν στην συνέχεια, βασίζονται στο βιβλίο του Edward R. Tufte, *The Visual Display of Quantitative Information* [21].

8.2 Αρχές Σχεδιασμού Γραφημάτων

8.2.1 Η Τελειότητα στο γράφημα

Η τελειότητα σε στατιστικά γραφήματα έχει να κάνει με την σαφήνεια, την ακρίβεια και την αποδοτικότητα. Γι' αυτό και ένα γράφημα θα πρέπει:

- να εμφανίζει τα δεδομένα
- να πείθει το θεατή να προβληματίζεται περισσότερο για την ουσία παρά για τη μεθοδολογία που ακολουθήθηκε, το σχεδιασμό ή κάτι άλλο
- να αποφεύγει τη διαστρέβλωση των δεδομένων
- να απεικονίζει πολλούς αριθμούς σε μικρό χώρο
- να παρουσιάζει μεγάλο αριθμό δεδομένων σε μια λογική σύνδεση μεταξύ τους
- να παροτρύνει το «μάτι» να συγκρίνει διαφορετικά τμήματα δεδομένων

- να εμφανίζει τα δεδομένα σε διάφορα επίπεδα πληροφορίας, από μια γενική άποψη σε μια πολύ καλή δομή.
- να υπάρχει συνέπεια ανάμεσα στη στατιστική και στην λεκτική περιγραφή των δεδομένων

8.2.2 Ακεραιότητα

Τα γραφήματα μπορούν εύκολα να χρησιμοποιηθούν για να εξαπατήσουν με τα αποτελέσματα. Ο περισσότερος κόσμος μόλις δει ένα γράφημα το πρώτο πράγμα που σκέφτεται είναι αν ανταποκρίνεται όντως στην πραγματικότητα. Για παράδειγμα τέτοια «απατηλά» γραφήματα μπορεί να είναι εκείνα που:

- συγκρίνουν ολόκληρες χρονικές περιόδους με μικρότερες χρονικές περιόδους
- χρησιμοποιούν διαφορετικές αναπαραστάσεις από τις γραμμικές για να μεγαλοποιήσουν τις διαφορές
- αποτυγχάνουν να προσαρμόσουν την ανάπτυξη του πληθυσμού ή τον πληθωρισμό σε οικονομικά γραφήματα
- χρησιμοποιούν παραλλαγές στο σχεδιασμό για να αποκρύψουν ή να διαστρεβλώσουν την διακύμανση των δεδομένων
- διαστρεβλώνουν την κάθετη κλίμακα (άξονας - y)
- αποκαλύπτουν μόνο ένα τμήμα των δεδομένων, κάτι που δεν επιτρέπει ολοκληρωμένη σύγκριση

Οδηγίες που βοηθούν στο να ικανοποιείται η ακεραιότητα σε ένα γράφημα περιλαμβάνουν τα ακόλουθα:

- να αποφεύγεται η περιττή πληροφορία πάνω στο γράφημα
- να παρουσιάζονται τα δεδομένα με μια εύλογη ακρίβεια
- να μην μεγαλοποιείτε με οπτικές εντυπώσεις εκτός και αν χρειάζεται για να μεταφέρετε την πληροφορία. Για παράδειγμα είναι δύσκολο να απεικονίσει κανείς το μέγεθος και την τροχιά των πλανητών σε έναν άξονα στο ίδιο γράφημα.
- να αποφεύγετε την κακή απεικόνιση της πληροφορίας. Υπογραμμισμένο κείμενο μέσα σε έντονα περιστοιχισμένα «κουτάκια», είναι πολύ δύσκολο να διαβαστεί.
- πάντα πρέπει να απαντάτε τις ακόλουθες ερωτήσεις:
 - Απεικονίζεται η αλήθεια στο γράφημα ;
 - Είναι ακριβής η αναπαράσταση ;

- Είναι τεκμηριωμένα τα δεδομένα;
- Παρουσιάζονται οι κατάλληλες συγκρίσεις και διαφορές πάνω στο γράφημα;

8.3 Θεωρία των γραφημάτων δεδομένων

8.3.1 Πυκνότητα εμφάνισης των δεδομένων

Ένα γράφημα θεωρείται καλό όταν απεικονίζει ένα πολυπληθές σύνολο δεδομένων. Η πυκνότητα των δεδομένων μπορεί να υπολογιστεί από τον ακόλουθο τύπο:

Πυκνότητα δεδομένων =

αριθμός των εγγραφών των δεδομένων / την επιφάνεια του γραφήματος

Στο σημείο αυτό να σημειώσουμε ότι γραφήματα με χαμηλή πυκνότητα δεδομένων μας αναγκάζουν να παρατηρούμε το γράφημα σειριακά και όχι χωρικά, κάτι που κάνει αρκετά δύσκολη την κατανόηση του.

Ένα καλό γράφημα ποιοτικά θα πρέπει να είναι:

- Συγκριτικό
- Πολλών μεταβλητών
- Υψηλής πυκνότητας δεδομένων
- Να αποκαλύπτει τις συγκρίσεις και τις αντιθέσεις των δεδομένων
- Και σχεδόν όλο το μελάνι που χρησιμοποιείται να είναι για την απεικόνιση των δεδομένων.

8.3.2 Στοιχεία του γραφήματος με πολλές λειτουργίες

Σε ένα γράφημα είναι δυνατόν ένα στοιχείο του να μεταφέρει πληροφορία για τα δεδομένα και συγχρόνως να είναι μέρος της σχεδίασης. Τέτοια παραδείγματα είναι οι άξονες, το πλέγμα (grid), και οι ετικέτες (labels).

Κατά το σχεδιασμό των αξόνων του γραφήματος, προστίθενται σε αυτούς τα σχετικά μεγέθη για την απεικόνιση των δεδομένων. Μια καλή τεχνική στην προσθήκη μεγεθών στους άξονες είναι τα μεγέθη να ορίζονται πέρα από τα δεδομένα. Κάτι τέτοιο αυξάνει το ποσοτικό στοιχείο ενός γραφήματος και την αντίληψη της

διάστασης του. Ο ορισμός των μεγεθών είναι καθοριστικός για την απεικόνιση των δεδομένων. Μια καλή τεχνική πάνω σε αυτό είναι τα μεγέθη να ορίζονται έτσι ώστε να εξυπηρετούν την αναπαράσταση των δεδομένων αλλά και να μεταφέρουν πληροφορία ως προς μια άλλη μεταβλητή.

Επίσης η προσθήκη πλέγματος (grid) εξυπηρετεί αυτό το σκοπό. Δηλαδή, το πλέγμα πάνω σε ένα γράφημα μπορεί να αναφέρεται απευθείας στα δεδομένα. Επιπλέον, βοηθάει στην ακρίβεια του γραφήματος καθώς επιτρέπει τον θεατή να κατανοήσει με περισσότερη ευκρίνεια και ακρίβεια τα αποτελέσματα.

Τέλος, ένα άλλο στοιχείο με διπλή λειτουργία είναι οι ετικέτες των αξόνων. Πολλές φορές κατά το σχεδιασμό του γραφήματος, πέρα από τον καθορισμό των μεγεθών, προστίθενται και ετικέτες (labels) στους άξονες. Η προσθήκη ετικετών βοηθάει στην κατανόηση του γραφήματος, αλλά σε συνδυασμό με τα μεγέθη που θα παρουσιάζονται στους άξονες κουράζει το μάτι του θεατή. Γιατί και στις περισσότερες περιπτώσεις προτιμάται οι ετικέτες των αξόνων να ταυτίζονται με τα μεγέθη.

8.3.3 Περισσότερο μελάνι για τα δεδομένα – Λιγότερο για όλα τα υπόλοιπα

Στο σχεδιασμό του γραφήματος θα πρέπει να προσέξετε ώστε η μεγαλύτερη ποσότητα του συνολικού μελανιού να αφιερώνεται για τα δεδομένα και όχι για τα υπόλοιπα στοιχεία του γραφήματος. Για να το πετύχετε αυτό θα πρέπει:

- Να αποφεύγετε να υπερφορτωμένα πλέγματα (grids)
- Να αντικαταστήσετε την τετραγωνισμένη περιοχή απεικόνισης δεδομένων με διακεκομμένες γραμμές
- Να χρησιμοποιείτε ένα απλό μαρκάρισμα αν θέλετε να υποδείξετε την πραγματική θέση ενός σημείου
- Να αποφεύγετε την προσθήκη annotation, marker και labels στην περιοχή απεικόνισης δεδομένων, εκτός αν πρόκειται για σημαντική πληροφορία.
- Οι τιμές στους άξονες να ξεκινάνε από την αρχική τιμή των δεδομένων και τελειώνουν με την τελευταία

8.3.4 Περισσότερα από ένα μικρά γραφήματα στο ίδιο γράφημα

Είναι δυνατόν σε ένα γράφημα να παρουσιάζονται περισσότερα από ένα γραφήματα σε μικρό μέγεθος. Ένας τέτοιος τύπος γραφήματος είναι το Multiple Pie. Σε ένα τέτοιο τύπο γραφήματος καθένα από τα μικρά γραφήματα αναπαριστούν τα ίδια δεδομένα από διαφορετικές απόψεις.

Τέτοιου τύπου γραφήματα τονίζουν τις συγκρίσεις ανάμεσα στα δεδομένα, και τη διάσταση του γραφήματος και είναι κατάλληλα για αναπαραστάσεις δεδομένων πολλών μεταβλητών.

Εκείνο που χρειάζεται προσοχή σε αυτή την περίπτωση είναι να χρησιμοποιείται το ίδιο μέγεθος και η ίδια κλίμακα για όλα τα μικρά γραφήματα.

8.3.5 Επιλογή χρωμάτων

Η επιλογή χρωμάτων είναι πολύ σημαντική κατά το σχεδιασμό ενός γραφήματος. Τα χρώματα βοηθούν στην κατανόηση των αποτελεσμάτων ενός γραφήματος αλλά χρειάζεται προσοχή στη χρήση τους.

Ο διαχωρισμός των δεδομένων με χρώματα είναι αποδοτικός αλλά ο συνδυασμός τους πρέπει να είναι διακριτικός ώστε να μην κουράζουν το μάτι του παρατηρητή. Τα αμιγώς φωτεινά χρώματα θα πρέπει να χρησιμοποιούνται για να χαρακτηρίσουν μικρές περιοχές που χρειάζεται να δοθεί προσοχή και ποτέ ως background. Τα σκοτεινά χρώματα καλό είναι να αποφεύγονται. Ο συνδυασμός χρωμάτων ανάμεσα στις αποχρώσεις του γκρι και σε φωτεινά χρώματα είναι πάντα αποδοτικός. Διαχωριστικές γραμμές που πιθανόν να χρησιμοποιούνται μπορούν να αλλάζουν χρώμα σύμφωνα με το background χρώμα που υπάρχει.

Τα χρώματα μπορούν να χρησιμοποιηθούν σε ετικέτες (labels), markers, annotations, και γενικά για να δώσουν έμφαση σε συγκεκριμένες τιμές. Μια καλή τεχνική είναι να χρησιμοποιείται το χρώμα για αντίθεση. Μια καλή αντίθεση είναι η προσθήκη χρώματος σε σημεία με background χρώμα το γκρι.

Παρακάτω ακολουθεί ένας πίνακας που υποδεικνύει το σωστό συνδυασμό χρωμάτων:

	Black	Blue	Green	Cyan	Red	Magenta	Brown	White
Black				Y		Y		Y
Blue								Y
Green	Y	Y		Y				
Cyan	Y							
White	Y							
Bold green	Y							
Bold Cyan	Y	Y		Y	Y	Y		
Magenta	Y		Y					
Yellow	Y	Y		Y		Y		

8.4 Επιλογή τύπου γραφήματος

Οι πιο διαδεδομένοι τύποι γραφήματος είναι τα ιστογράμματα (bars), πίτες (pie), γραμμές (line). Κάθε γράφημα όμως είναι κατάλληλο για κάποια συγκεκριμένη κατηγορία εφαρμογών απεικόνισης δεδομένων. Σε αυτό το σημείο θα περιγράψουμε τον τύπο εφαρμογών για τον οποίο είναι κατάλληλα τα πιο δημοφιλή γραφήματα:

Ιστογράμματα (Bars)

Τα ιστογράμματα συγκρίνουν τιμές ανάμεσα σε διάφορες κατηγορίες και είναι ιδανικά για τη απεικόνιση και σύγκριση διαφορετικών παραμέτρων για ένα συγκεκριμένο αντικείμενο. Το πλεονέκτημα με τα ιστογράμματα είναι ότι χρησιμοποιούν την οριζόντια κατεύθυνση στην απεικόνιση δεδομένων, κάτι που βοηθάει τον παρατηρητή στην κατανόηση του αποτελέσματος. Ένα παράδειγμα εφαρμογής δεδομένων, που είναι κατάλληλα για απεικόνιση σε ένα ιστόγραμμα, είναι αν θέλετε να συγκρίνεται την απόδοση της Ferrari και της Lotus σε σχέση με την μέγιστη ταχύτητα, την κατανάλωση καυσίμων και την επιτάχυνση.

Πίτες (pie)

Τα διαγράμματα πίτων είναι ένας πολύ αποτελεσματικός τρόπος απεικόνισης πληροφορίας όταν θέλετε να παρουσιάσετε τα διαφορετικά μέρη ενός συνόλου, ή τα

ποσοστά ενός συνόλου. Τα γραφήματα πιτών επιδεικνύουν τη συνεισφορά κάθε τιμής. Γραφήματα πιτών μπορούν να αναπαραστήσουν μια σειρά δεδομένων.

Πολλά παραδείγματα μπορούν να βρεθούν για αυτόν τον τύπο γραφήματος : όταν θέλετε να επιδείξετε πώς το Κοινοβούλιο μιας χώρας διαιρείται σε διαφορετικά πολιτικά κόμματα (κάθε πολιτικό κόμμα θα διοριστεί είτε με το αριθμό θέσεων είτε με το ποσοστό του. Κάθε φέτα ή τομέας της πίτας θα παρουσιάσει αυτήν την αξία. Όλες οι τιμές για όλες τις φέτες, μαζί, θα κάνουν το σύνολο). Για παράδειγμα, μπορείτε να χρησιμοποιήσετε το γράφημα πιτών για να συγκρίνετε διαφορετικές χώρες προέλευσης μιας κατηγορίας σπουδαστών, ή η κατανομή του ετήσιου προϋπολογισμού στους διαφορετικούς τομείς της επιχείρησής σας.

Γραμμές (Lines)

Τα γραφήματα αυτά είναι πολύ χρήσιμα να παρουσιάσουν την εξέλιξη ενός στοιχείου με την πάροδο του χρόνου και χρησιμοποιούνται συχνά για να παρουσιάσουν τις μεταβολές περισσότερων από ενός στοιχείων με την πάροδο του χρόνου.

Φανταστείτε ότι θέλετε να συγκρίνετε τους βαθμούς σπουδαστών έπειτα από την αποφοίτη τους. Μπορείτε να θελήσετε να χρησιμοποιήσετε το γράφημα γραμμών δεδομένου ότι θα παρουσιάσει, με διαφορετικών χρωμάτων γραμμές, τους βαθμούς που έλαβε κάθε σπουδαστής κατά τη διάρκεια μιας χρονικής περιόδου. Σε αυτήν την περίπτωση, θα έχετε τα διαφορετικά σημεία στο σχέδιό σας που θα παρουσιάζουν τους βαθμούς που αντιστοιχούν στα διάφορα έτη μελέτης. Το διάγραμμα γραμμών θα συνδέσει ακριβώς για σας αυτά τα σημεία χρησιμοποιώντας τις ευθείες γραμμές.

Για την απεικόνιση πληροφορίας που σχετίζεται με μεγάλες χρονικές περιόδους, μπορούν να επιλεγούν και άλλοι τύποι γραφημάτων όπως είναι τα TimeSeries και Gantt. Για παράδειγμα, αν θέλετε να δείξετε την πρόοδο της επιχείρησής σας ανά κάποια έτη, ένας τέτοιος τύπος γραφήματος θα ήταν αποδοτικός.

Για την απεικόνιση στατιστικής πληροφορίας, ιδανικό είναι ένα γράφημα Scatter.

8.5 Ψηφιακή Βιβλιοθήκη Γραφημάτων

Η ψηφιακή βιβλιοθήκη γραφημάτων δημιουργήθηκε με σκοπό να συμπληρώσει τον οδηγό σωστού σχεδιασμού γραφημάτων και να βοηθήσει πρακτικά τους χρήστες της εφαρμογής. Για τον σκοπό αυτό έχει δημιουργηθεί μια βιβλιοθήκη που αποτελείται από πρότυπα γραφήματα, τα οποία έχουν σχεδιαστεί ακολουθώντας τις αρχές και τους κανόνες που περιγράψαμε παραπάνω.

Ο εκάστοτε χρήστης της εφαρμογής έχει τη δυνατότητα να πλοηγηθεί στην ψηφιακή αυτή βιβλιοθήκη, μέσω της εφαρμογής δημιουργίας πρότυπων γραφημάτων, να δει όλα τα γραφήματα που υπάρχουν εκεί και να επιλέξει να προσθέσει στα δικά του πρότυπα γραφήματα, οποιοδήποτε από αυτά τον ενδιαφέρει.

Κεφάλαιο 9

Ανακεφαλαίωση – Μελλοντικές Επεκτάσεις

9.1 Ανακεφαλαίωση

Τα γραφήματα είναι ένας αποδοτικός τρόπος αναπαράστασης δεδομένων. Χρησιμοποιούνται συχνά για να καταστήσουν ευκολότερη την κατανόηση μεγάλου όγκου δεδομένων και τη σχέση μεταξύ διαφορετικών τμημάτων δεδομένων. Τα γραφήματα μπορούν να διαβαστούν πιο γρήγορα απ' ό,τι τα ίδια τα ακατέργαστα δεδομένα που απεικονίζουν και αυτό διότι ο ανθρώπινος εγκέφαλος κατανοεί πιο εύκολα μια εικόνα από το απλό κείμενο. Χρησιμοποιούνται από μια πληθώρα εφαρμογών σε όλους τους τομείς της καθημερινότητας και είναι δυνατόν να δημιουργηθούν με το χέρι (σε ένα ειδικό χαρτί γραφικών αναπαραστάσεων) ή από τον υπολογιστή χρησιμοποιώντας μια ειδική εφαρμογή .

Η χρήση των γραφημάτων βρίσκει εφαρμογή σε πολλές δραστηριότητες της καθημερινότητας και οι λόγοι που οδηγούν σε αυτό είναι ότι τα γραφήματα:

1. είναι ένας εύχρηστος και εύληπτος τρόπος για να μεταφέρουν πληροφορία,
2. είναι ένας γρήγορος τρόπος για να απεικονίσει κανείς αυτά που θέλει να πει
3. είναι ένα ισχυρό εργαλείο για να δοθεί έμφαση σε συγκεκριμένα σημεία ή μηνύματα τα οποία πρέπει να σημειωθούν
4. δίνουν μια πειστική ερμηνεία ως προς το αποτέλεσμα
5. είναι πιο ενδιαφέροντα από το απλό κείμενο και γίνονται πιο γρήγορα κατανοητά από τον ανθρώπινο εγκέφαλο. [21]

Ωστόσο, παρόλα τα πλεονεκτήματα της χρήσης των γραφημάτων, υπάρχουν σημαντικά προβλήματα στην ανάπτυξη τους, κυρίως μέσω ειδικών εφαρμογών σε ηλεκτρονικό υπολογιστή. Και πιο συγκεκριμένα τα προβλήματα αυτά έχουν να κάνουν με:

1. Υπάρχει η ανάγκη υλοποίησης ενός υποσυστήματος, το οποίο θα επιτρέπει την ανάπτυξη γραφημάτων και παρουσίαση αυτών σε οποιοδήποτε περιβάλλον.

2. Η ανάπτυξη ενός γραφήματος να γίνεται με έναν εύχρηστο τρόπο.
3. Να παρέχεται η σχετική βοήθεια και οδηγίες που αφορούν την ανάπτυξη ενός σωστά σχεδιασμένου γραφήματος.

Η παρούσα διπλωματική εργασία εκπληρώνει τρεις βασικούς στόχους: πρώτον, τη παροχή κατάλληλης εφαρμογής για τη δημιουργία και μορφοποίηση πρότυπων γραφημάτων με γραφικό τρόπο τα οποία θα είναι επαναχρησιμοποιήσιμα. Δεύτερον, τη παροχή μιας υπηρεσίας γραφικής απόδοσης γραφημάτων σε πραγματικό χρόνο από εφαρμογές-πελάτες οι οποίες επιθυμούν τη γραφική απόδοση και παρουσίαση αυτών για δικά τους δεδομένα. Τρίτον, την παροχή ενός πλήρους οδηγού δημιουργίας σωστών γραφημάτων μέσω σαφών κανόνων και βοήθειας δημιουργίας και μορφοποίησης. Ο οδηγός αυτός συμπληρώνεται και από μια πρότυπη βιβλιοθήκη έτοιμων προτύπων (templates) τα οποία έχουν παραχθεί με χρήση αυτών των κανόνων.

Στα κεφάλαια που προηγήθηκαν, αρχικά περιγράψαμε τις απαιτήσεις συμπεριφοράς της εφαρμογής δημιουργίας πρότυπων γραφημάτων και της υπηρεσίας ιστού, παρουσιάζοντας έτσι τους τύπους χρηστών και τις δυνατότητες που έχει κάθε ένας από αυτούς. Στην συνέχεια παρουσιάσαμε σχετικές εφαρμογές που επιτρέπουν την ανάπτυξη γραφημάτων. Μέσα από εκεί είδαμε τα πλεονεκτήματα και κυρίως τα μειονεκτήματα κάθε εφαρμογής που μας οδήγησαν να σχεδιάσουμε την παρούσα εργασία. Ακολουθως, όλες τις επιλογές παραμετροποίησης ενός γραφήματος που προσφέρονται μέσα από την εφαρμογή. Συνεχίσαμε παρουσιάζοντας αναλυτικά την γραφική διεπαφή της εφαρμογής δημιουργία πρότυπων γραφημάτων για έναν χρήστη αλλά και για τον Διαχειριστή.

Στο κεφάλαιο 6 εξηγήσαμε όλες τις τεχνολογίες υλοποίησης που χρησιμοποιήσαμε για την ανάπτυξη ολόκληρης της εφαρμογής και σε αναφέραμε σε ποιο σημείο χρησιμοποιήθηκε κάθε μια από αυτές. Παρουσιάσαμε την αρχιτεκτονική της εφαρμογής δημιουργίας πρότυπων γραφημάτων σύμφωνα με το πρότυπο MVC και την αρχιτεκτονική της υπηρεσίας ιστού (Web Service) για την ανακατασκευή του τελικού γραφήματος. Τέλος, ενσωματώσαμε στο κεφάλαιο 8 τον οδηγό σχεδιασμού σωστών γραφημάτων παρουσιάζοντας τις πρακτικές και τους κανόνες που χρειάζεται να ακολουθηθούν κατά την δημιουργία γραφημάτων.

9.2 Μελλοντικές Επεκτάσεις

Στην παρούσα διπλωματική δόθηκε έμφαση στην υλοποίηση ενός συστήματος που επιτρέπει την ανάπτυξη γραφημάτων με έναν εύχρηστο τρόπο και παρέχει οδηγίες για το σωστό σχεδιασμό γραφημάτων. Μια επέκταση που θα μπορούσε να γίνει είναι η μοντελοποίηση και ο έλεγχος των κανόνων δημιουργίας σωστών γραφημάτων. Κάτι τέτοιο σημαίνει ότι θα γίνεται έλεγχος στις τιμές των παραμέτρων ενός γραφήματος σε πραγματικό χρόνο, κάθε φορά που ο εκάστοτε χρήστης αλλάζει την προκαθορισμένη τιμή τους.

Από την άλλη πλευρά, όσον αφορά την απεικόνιση πραγματικών δεδομένων πάνω σε ένα πρότυπο γράφημα, σχεδιάσαμε μια υπηρεσία ιστού, που δέχεται ως είσοδο ένα αρχείο με τα δεδομένα και ανακατασκευάζει το γράφημα. Μια επέκταση που θα μπορούσε να γίνει εδώ είναι να υπάρχει η δυνατότητα σύνδεσης με μια βάση δεδομένων – πελάτη, για την ανάκτηση των δεδομένων και παρουσίαση γραφημάτων μέσα από την εφαρμογή. Κάτι τέτοιο όμως σημαίνει ότι, θα πρέπει να είναι γνωστό εκ των προτέρων το σχήμα της βάσης δεδομένων.

Τέλος, όπως έχουμε αναφέρει η εφαρμογή που σχεδιάσαμε υποστηρίζει μια πληθώρα τύπων γραφημάτων. Μία επέκταση που θα μπορούσε να γίνει σε αυτό το επίπεδο, είναι η υποστήριξη και γεωγραφικών γραφημάτων, τα οποία θα αναπαριστούν δεδομένα με βάση κάποια γεωγραφικά κριτήρια.

BIBΛΙΟΓΡΑΦΙΑ

- [1] Wikipedia, <http://en.wikipedia.org/wiki/Charts>
- [2] Alistair Cockburn, “Writing Effective Use Cases”
- [3] JFreeChart 100% open source java library under the terms of the GNU [Lesser General Public License](http://www.gnu.org/licenses/lgpl.html) (LGPL), <http://www.jfree.org/jfreechart/>
- [4] <http://www.clabberhead.com/googlechartgenerator.html>
- [5] <http://www.smartdraw.com/specials/smartdraw.asp>
- [6] <http://aspfusion.net/advgraphgenerator-d.htm>
- [7] <http://cewolf.sourceforge.net/new/index.html>
- [8] <http://jwebchart.sourceforge.net/index.html>
- [9] Martin Fowler. “GUI Architectures”,
<http://www.martinfowler.com/eaDev/uiArchs.htm>
- [10] XML, <http://en.wikipedia.org/wiki/XML>
- [11] XML Schema, http://en.wikipedia.org/wiki/XML_Schema_%28W3C%29
- [12] Διαφάνειες μαθήματος “**Ενοποίηση Πληροφορίας και Υπηρεσιών στο Διαδίκτυο**”, Σ. Χριστοδουλάκης
- [13] Web Services, http://en.wikipedia.org/wiki/Web_services
- [14] WSDL, <http://www.w3.org/TR/wsdl>
- [15] SOAP, <http://www.w3.org/TR/soap>
- [16] UDDI, <http://www.oasis-open.org/committees/uddispec/doc/tcspecs.htm>
- [17] My SQL, <http://dev.mysql.com/doc/>
- [18] Java Script, <http://www.w3schools.com/js/default.asp>
- [19] Διαφάνειες Μαθήματος «**Ανάπτυξη εφαρμογών και πληροφοριακών Συστημάτων στο Διαδίκτυο**», Σ. Χριστοδουλάκης

Βιβλιογραφία

- [20] Apache Foundation, “XMLBeans XML Binding Framework”,
<http://xmlbeans.apache.org/2003>
- [21] Edward Tufte, «Visual Display of Quantitative Information»
- [22] Core J2EE Design Patterns, Data Access Object,
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>

Παράρτημα Α

Παρακάτω ακολουθεί το XML Schema για το αρχείο που επιστέφει το web service με τα δεδομένα της εικόνας

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://serviceResponse.charttemplates.org"
  xmlns:tns="http://serviceResponse.charttemplates.org"
  xmlns="http://serviceResponse.charttemplates.org"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
  elementFormDefault="qualified">

  <xsd:import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="http://www.w3.org/2005/05/xmlmime"/>

  <xsd:complexType name="AttachmentType">
    <xsd:sequence>
      <xsd:element name="fileName" type="xsd:string" />
      <xsd:element name="binaryData" type="xmime:base64Binary" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="AttachmentRequest" type="tns:AttachmentType" />

</xsd:schema>
```

Παρακάτω ακολουθεί το xml Schema για το αρχείο εισαγωγής δεδομένων στο Web Service:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://properties.charttemplates.org"
  xmlns:tns="http://properties.charttemplates.org"
  elementFormDefault="qualified">
  <xsd:complexType name="pointType">
    <xsd:sequence>
      <xsd:element name="x" type="xsd:double"/>
      <xsd:element name="y" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="categoryDatasetType">
    <xsd:sequence>
      <xsd:element name="seriesName" type="xsd:string" maxOccurs="1"/>
      <xsd:element name="columnName" type="xsd:string" maxOccurs="1"/>
      <xsd:element name="value" type="xsd:double" minOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="xyDatasetType">
    <xsd:sequence>
      <xsd:element name="seriesName" type="xsd:string" maxOccurs="1"/>
      <xsd:element name="xValue" type="xsd:double" minOccurs="1"/>
      <xsd:element name="yValue" type="xsd:double" minOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="pieDatasetType">
    <xsd:sequence>
```

Παράρτημα Α

```
<xsd:element name="seriesName" type="xsd:string" maxOccurs="1"/>
<xsd:element name="value" type="xsd:double" maxOccurs="1"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="histogramDatasetType">
  <xsd:sequence>
    <xsd:element name="key" type="xsd:string" maxOccurs="1"/>
    <xsd:element name="values" type="xsd:double" maxOccurs="unbounded"/>
    <xsd:element name="bin" type="xsd:int" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ganttDatasetType">
  <xsd:sequence>
    <xsd:element name="taskSeriesName" type="xsd:string"/>
    <xsd:element name="taskName" type="xsd:string" maxOccurs="unbounded"/>
>
    <xsd:element name="date1" type="xsd:string" maxOccurs="unbounded"/>
    <xsd:element name="date2" type="xsd:string" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="statisticDatasetType">
  <xsd:sequence>
    <xsd:element name="xVal" type="xsd:double"/>
    <xsd:element name="yVal" type="xsd:double"/>
    <xsd:element name="seriesName" type="xsd:string"/>
    <xsd:element name="key" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="timeSeriesDatasetType">
  <xsd:sequence>
    <xsd:element name="seriesName" type="xsd:string"/>
    <xsd:element name="date" type="xsd:string" maxOccurs="unbounded"/>
    <xsd:element name="value" type="xsd:double" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="bubbleDatasetType">
  <xsd:sequence>
    <xsd:element name="seriesName" type="xsd:string" maxOccurs="1"/>
    <xsd:element name="xValue" type="xsd:double" minOccurs="1"/>
    <xsd:element name="yValue" type="xsd:double" minOccurs="1"/>
    <xsd:element name="zValue" type="xsd:double" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="diffPieDatasetType">
  <xsd:sequence>
    <xsd:element name="newDataset" type="tns:pieDatasetType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="previousDataset" type="tns:pieDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="chartDatasetType">
  <xsd:sequence>
    <xsd:choice>
```

Παράρτημα Α

```
<xsd:element name="AreaDataset" type="tns:categoryDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="BarDataset" type="tns:categoryDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="LineDataset" type="tns:categoryDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="Line3DDataset" type="tns:categoryDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="MultiplePieDataset" type="tns:categoryDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="BoxAndWhiskerDataset"
type="tns:categoryDatasetType" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="StackedBarDataset" type="tns:categoryDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="IntervalBarDataset" type="tns:categoryDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="LayeredBarDataset" type="tns:categoryDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="MultipleLinesDataset"
type="tns:categoryDatasetType" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="MultipleLines3DDataset"
type="tns:categoryDatasetType" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="XYBarDataset" type="tns:xyDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="XYLineDataset" type="tns:xyDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="PolarDataset" type="tns:xyDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="XYStepDataset" type="tns:xyDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="XYStepAreaDataset" type="tns:xyDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="XYAreaDataset" type="tns:xyDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="ScatterDataset" type="tns:xyDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="StackedXYAreaDataset" type="tns:xyDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="PieDataset" type="tns:pieDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="Pie3DDataset" type="tns:pieDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="RingDataset" type="tns:pieDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="DifferencePieDataset" type="tns:diffPieDatasetType"
minOccurs="0"/>
<xsd:element name="HistogramDataset" type="tns:histogramDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="TimeSeriesDataset"
type="tns:timeSeriesDatasetType" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="StatisticalDataset" type="tns:statisticDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
<xsd:element name="GanttDataset" type="tns:ganttDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
```


Παράρτημα Α

```
<xsd:element name="BubbleDataset" type="tns:bubbleDatasetType"
minOccurs="0" maxOccurs="unbounded"/>
</xsd:choice>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="xyTextAnnotationType">
  <xsd:sequence>
    <xsd:element name="xytext" type="xsd:string"/>
    <xsd:element name="textPoint" type="tns:pointType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="xyLineAnnotationType">
  <xsd:sequence>
    <xsd:element name="pointLineOne" type="tns:pointType" maxOccurs="1"/>
    <xsd:element name="pointLineTwo" type="tns:pointType" maxOccurs="1"/>
    <xsd:element name="strokeLine" type="xsd:boolean"/>
    <xsd:element name="paintLine" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="xyBoxAnnotationType">
  <xsd:sequence>
    <xsd:element name="pointOne" type="tns:pointType" maxOccurs="1"/>
    <xsd:element name="pointTwo" type="tns:pointType" maxOccurs="1"/>
    <xsd:element name="stroke" type="xsd:boolean"/>
    <xsd:element name="outlinePaint" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="xyAnnotationType">
  <xsd:sequence>
    <xsd:element name="annotationVisible" type="xsd:boolean"/>
    <xsd:element name="xyBoxAnnotation" type="tns:xyBoxAnnotationType"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="xyLineAnnotation" type="tns:xyLineAnnotationType"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="xyTextAnnotation" type="tns:xyTextAnnotationType"
minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="categoryLineType">
  <xsd:sequence>
    <xsd:element name="key1" type="xsd:string"/>
    <xsd:element name="value1" type="xsd:double"/>
    <xsd:element name="key2" type="xsd:string"/>
    <xsd:element name="value2" type="xsd:double"/>
    <xsd:element name="linePaint" type="xsd:string"/>
    <xsd:element name="lineStroke" type="xsd:boolean"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="categoryTextType">
  <xsd:sequence>
    <xsd:element name="textAnnotation" type="xsd:string"/>
    <xsd:element name="keyAnnotation" type="xsd:string"/>
    <xsd:element name="valueAnnotation" type="xsd:double"/>
  </xsd:sequence>
</xsd:complexType>
```

Παράρτημα Α

```
<xsd:complexType name="pointerAnnotationType">
  <xsd:sequence>
    <xsd:element name="textAnnotation" type="xsd:string"/>
    <xsd:element name="keyAnnotation" type="xsd:string"/>
    <xsd:element name="valueAnnotation" type="xsd:double"/>
    <xsd:element name="anchor" type="xsd:double"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="categoryAnnotations">
  <xsd:sequence>
    <xsd:element name="annotationVisible" type="xsd:boolean"/>
    <xsd:element name="categoryLine" type="tns:categoryLineType"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="categoryText" type="tns:categoryTextType"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PointerAnnotation" type="tns:pointerAnnotationType"
minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="rangeMarkerType">
  <xsd:sequence>
    <xsd:element name="markerVisible" type="xsd:boolean"/>
    <xsd:element name="markerValue" type="xsd:double" minOccurs="0"/>
    <xsd:element name="labelMarker" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="domainMarkerType">
  <xsd:sequence>
    <xsd:element name="markerVisible" type="xsd:boolean"/>
    <xsd:element name="comparableKey" type="xsd:string" minOccurs="0"/>
    <xsd:element name="labelMarker" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="propertiesType">
  <xsd:sequence>
    <xsd:element name="propertiesFilename" type="xsd:string"/>
    <xsd:element name="username" type="xsd:string"/>
    <xsd:element name="chartTitle" type="xsd:string" maxOccurs="1"
nillable="false"/>
    <xsd:element name="chartSubtitle" type="xsd:string" minOccurs="0"
maxOccurs="1" />
    <xsd:element name="commonAxisLabel" type="xsd:string" maxOccurs="1"
nillable="false"/>
    <xsd:element name="xAxisLabel" type="xsd:string" maxOccurs="1"
nillable="false"/>
    <xsd:element name="yAxisLabel" type="xsd:string" maxOccurs="1"
nillable="false"/>
    <xsd:element name="dataset" type="tns:chartDatasetType" minOccurs="1"
maxOccurs="1"/>
    <xsd:element name="annotationXYChart" type="tns:xyAnnotationType"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="annotationCategoryChart"
type="tns:categoryAnnotations" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="rangeMarker" type="tns:rangeMarkerType"
minOccurs="0" maxOccurs="1"/>
```

Παράρτημα Α

```
<xsd:element name="domainMarker" type="tns:domainMarkerType"
minOccurs="0" maxOccurs="1"/>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="chartProperties" type="tns:propertiesType"/>
</xsd:schema>
```

Παρακάτω ακολουθεί το Xml Schema για το αρχείο αποθήκευσης των παραμέτρων ενός πρότυπου γραφήματος:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:complexType name="chartsType">
    <xs:sequence>
      <xs:element name="chartType">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Area"/>
            <xs:enumeration value="Bar"/>
            <xs:enumeration value="Bar3D"/>
            <xs:enumeration value="Line"/>
            <xs:enumeration value="Line3D"/>
            <xs:enumeration value="MultiplePie"/>
            <xs:enumeration value="MultiplePie3D"/>
            <xs:enumeration value="StackedArea"/>
            <xs:enumeration value="StackedBar"/>
            <xs:enumeration value="StackedBar3D"/>
            <xs:enumeration value="StackedXYBar"/>
            <xs:enumeration value="IntervalBar"/>
            <xs:enumeration value="StatisticalBar"/>
            <xs:enumeration value="DifferencePie"/>
            <xs:enumeration value="Pie"/>
            <xs:enumeration value="Pie3D"/>
            <xs:enumeration value="Ring"/>
            <xs:enumeration value="Histogram"/>
            <xs:enumeration value="XYBar"/>
            <xs:enumeration value="XYLine"/>
            <xs:enumeration value="Polar"/>
            <xs:enumeration value="XYStepArea"/>
            <xs:enumeration value="Scatter"/>
            <xs:enumeration value="XYStep"/>
            <xs:enumeration value="TimeSeries"/>
            <xs:enumeration value="XYArea"/>
            <xs:enumeration value="BoxAndWhisker"/>
            <xs:enumeration value="Bubble"/>
            <xs:enumeration value="Gantt"/>
            <xs:enumeration value="StackedXYArea"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Παράρτημα Α

```

value="MultipleLine3D"/>
value="CombinedRangeCat"/>
value="CombinedDomainCat"/>
value="CombinedRangeXY"/>
value="CombinedDomainXY"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="chart" type="chartCreationType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="optionalParameters"
type="optionalParametersType" minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="chartCreationType">
<xs:sequence>
<xs:element name="simpleTitle" type="xs:string"/>
<xs:element name="xAxisValue" type="xs:string"/>
<xs:element name="yAxisValue" type="xs:string"/>
<xs:element name="plotOrientation" type="xs:string"
minOccurs="0"/>
<xs:element name="tableOrder" type="xs:string"
minOccurs="0"/>
<xs:element name="legend" type="xs:boolean"/>
<xs:element name="tooltips" type="xs:boolean"/>
<xs:element name="urls" type="xs:boolean"/>
<xs:element name="percentDiff" type="xs:int" minOccurs="0"/>
<xs:element name="greenForIncrease" type="xs:boolean"
minOccurs="0"/>
<xs:element name="suttle" type="xs:boolean" minOccurs="0"/>
<xs:element name="showDiff" type="xs:boolean"
minOccurs="0"/>
<xs:element name="dateAxis" type="xs:boolean"
minOccurs="0"/>
<xs:element name="commonAxisLabel" type="xs:string"/>
<xs:element name="nameTemplate1" type="xs:string"/>
<xs:element name="nameTemplate2" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="XYType">
<xs:sequence>
<xs:element name="xValue" type="xs:double"/>
<xs:element name="yValue" type="xs:double"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="titleType">
<xs:sequence>

```

Παράρτημα Α

```

<xs:element name="titleChart" type="xs:string"/>
<xs:element name="backgroundPaint" type="xs:string"
minOccurs="0"/>
<xs:element name="border" type="borderType"
minOccurs="0"/>
<xs:element name="font" type="fontType" minOccurs="0"/>
<xs:element name="height" type="xs:double" minOccurs="0"/>
<xs:element name="weight" type="xs:double" minOccurs="0"/>
<xs:element name="alignment" type="xs:string"
minOccurs="0"/>
<xs:element name="margin" type="borderType"
minOccurs="0"/>
<xs:element name="padding" type="borderType"
minOccurs="0"/>
<xs:element name="position" type="xs:string" minOccurs="0"/>
<xs:element name="paintTitle" type="xs:string" minOccurs="0"/>
>
<xs:element name="tooltipText" type="xs:string"
minOccurs="0"/>
<xs:element name="urlText" type="xs:string" minOccurs="0"/>
<xs:element name="subTitle" type="subTitleType"
minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="subTitleType">
<xs:sequence>
<xs:element name="text" type="xs:string" minOccurs="0"/>
<xs:element name="border" type="borderType"
minOccurs="0"/>
<xs:element name="font" type="fontType" minOccurs="0"/>
<xs:element name="height" type="xs:double" minOccurs="0"/>
<xs:element name="weight" type="xs:double" minOccurs="0"/>
<xs:element name="halignment" type="xs:string"
minOccurs="0"/>
<xs:element name="valignment" type="xs:string"
minOccurs="0"/>
<xs:element name="margin" type="borderType"
minOccurs="0"/>
<xs:element name="padding" type="borderType"
minOccurs="0"/>
<xs:element name="position" type="xs:string" minOccurs="0"/>
<xs:element name="tooltipText" type="xs:string"
minOccurs="0"/>
<xs:element name="urlText" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="fontType">
<xs:sequence>
<xs:element name="fontName" type="xs:string"/>
<xs:element name="fontSize" type="xs:int"/>
<xs:element name="fontOptions" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="borderType">
<xs:sequence>

```

Παράρτημα Α

```

        <xs:element name="top" type="xs:double"/>
        <xs:element name="left" type="xs:double"/>
        <xs:element name="bottom" type="xs:double"/>
        <xs:element name="right" type="xs:double"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="optionalParametersType">
    <xs:sequence>
        <xs:element name="chartProperties"
type="chartPropertiesType"/>
        <xs:element name="categoryPlot" type="categoryPlotType"
maxOccurs="1"/>
        <xs:element name="piePlot" type="piePlotType"
maxOccurs="1"/>
        <xs:element name="piePlot3D" type="piePlot3DType"
maxOccurs="1"/>
        <xs:element name="ringPlot" type="ringPlotType"
maxOccurs="1"/>
        <xs:element name="xyPlot" type="xyPlotType"
maxOccurs="1"/>
        <xs:element name="polarPlot" type="polarPlotType"
maxOccurs="1"/>
        <xs:element name="multiplePie" type="multiplePieType"
maxOccurs="1"/>
        <xs:element name="titleProperties" type="titleType"
maxOccurs="1"/>
        <xs:element name="combinedPlot" type="combinedPlotType"
maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="combinedPlotType">
    <xs:sequence>
        <xs:element name="legend" type="legendType"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="multiplePieType">
    <xs:sequence>
        <xs:element name="backColor" type="xs:string"/>
        <xs:element name="outlineColor" type="xs:string" />
        <xs:element name="extractOrder" type="xs:string" />
        <xs:element name="limit" type="xs:double" />
        <xs:element name="stroke" type="xs:boolean" />
        <xs:element name="legend" type="legendType"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="polarPlotType">
    <xs:sequence>
        <xs:element name="angleGridLinePaint" type="xs:string"/>
        <xs:element name="angleGridLineStroke" type="xs:boolean"/>
        <xs:element name="angleGridLineVisible" type="xs:boolean"/>
        <xs:element name="angleLabelPaint" type="xs:string"/>
        <xs:element name="angleLabelVisible" type="xs:boolean"/>
        <xs:element name="radiusGridLinePaint" type="xs:string"/>
        <xs:element name="radiusGridLineStroke" type="xs:boolean"/

```

>

Παράρτημα Α

```

        <xs:element name="radiusGridLineVisible" type="xs:boolean"/
>
        <xs:element name="legend" type="legendType"/>
        <xs:element name="valAxis" type="valueAxisType"
minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="xyPlotType">
    <xs:sequence>
        <xs:element name="renderingOrderDataset" type="xs:string"/>
        <xs:element name="axisOffset" type="borderType"/>
        <xs:element name="quadrant" type="quadrantType"/>
        <xs:element name="domainGridLinesVisible"
type="xs:boolean"/>
        <xs:element name="domainGridLinePaint" type="xs:string"/>
        <xs:element name="domainGridLineStroke"
type="xs:boolean"/>
        <xs:element name="domainTickBandPaint" type="xs:string"/>
        <xs:element name="rangeGridLinesVisible" type="xs:boolean"/
>
        <xs:element name="rangeGridLinePaint" type="xs:string"/>
        <xs:element name="rangeTickBandPaint" type="xs:string"/>
        <xs:element name="rangeGridLineStroke" type="xs:boolean"/>
        <xs:element name="backgroundPaint" type="xs:string"/>
        <xs:element name="weight" type="xs:int"/>
        <xs:element name="backAlpha" type="xs:float"/>
        <xs:element name="outlinePaint" type="xs:string"/>
        <xs:element name="outlineStroke" type="xs:boolean"/>
        <xs:element name="outlineVisible" type="xs:boolean"/>
        <xs:element name="rangeZeroBaselinePaint" type="xs:string"/
>
        <xs:element name="rangeZeroBaselineStroke"
type="xs:boolean"/>
        <xs:element name="rangeZeroBaselineVisible"
type="xs:boolean"/>
        <xs:element name="seriesRenderingOrder" type="xs:string"/>
        <xs:element name="xyAnnotations" type="xyAnnotationType"/
>
        <xs:element name="labelVisible" type="xs:boolean"/>
        <xs:element name="label" type="labelType"/>
        <xs:element name="legend" type="legendType"/>
        <xs:element name="valAxis" type="valueAxisType"
minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="quadrantType">
    <xs:sequence>
        <xs:element name="qx" type="xs:double"/>
        <xs:element name="qy" type="xs:double"/>
        <xs:element name="paint" type="xs:string"/>
        <xs:element name="qlIdx" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="xyAnnotationType">
    <xs:sequence>

```

Παράρτημα Α

```

        <xs:element name="xyBoxAnnotation"
type="xyBoxAnnotationType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="xyLineAnnotation"
type="xyLineAnnotationType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="xyTextAnnotation"
type="xyTextAnnotationType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="xyTextAnnotationType">
    <xs:sequence>
        <xs:element name="xytext" type="xs:string"/>
        <xs:element name="textPoint" type="pointType"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="xyLineAnnotationType">
    <xs:sequence>
        <xs:element name="pointLine" type="pointType"
minOccurs="2" maxOccurs="2"/>
        <xs:element name="strokeLine" type="xs:boolean"/>
        <xs:element name="paintLine" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="xyBoxAnnotationType">
    <xs:sequence>
        <xs:element name="point" type="pointType" minOccurs="2"
maxOccurs="2"/>
        <xs:element name="stroke" type="xs:boolean"/>
        <xs:element name="outlinePaint" type="xs:string"/>
        <xs:element name="fillPaint" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="pointType">
    <xs:sequence>
        <xs:element name="x" type="xs:double"/>
        <xs:element name="y" type="xs:double"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ringPlotType">
    <xs:complexContent>
        <xs:extension base="piePlotType">
            <xs:sequence>
                <xs:element name="innerSeperatorExtension"
type="xs:double"/>
                <xs:element name="outerSeperatorExtension"
type="xs:double"/>
                <xs:element name="sectionDepth"
type="xs:double"/>
                <xs:element name="seperatorPaint"
type="xs:string"/>
                <xs:element name="seperatorStroke"
type="xs:boolean"/>
                <xs:element name="seperatorVisible"
type="xs:boolean"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```


Παράρτημα Α

```

        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="piePlot3DType">
        <xs:complexContent>
            <xs:extension base="piePlotType">
                <xs:sequence>
                    <xs:element name="darkerSides"
type="xs:boolean"/>
                    <xs:element name="fAlpha" type="xs:float"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="piePlotType">
        <xs:sequence>
            <xs:element name="baseSectionOutline" type="xs:string"/>
            <xs:element name="baseSectionOutlineStroke"
type="xs:boolean"/>
            <xs:element name="backColor" type="xs:string"/>
            <xs:element name="backImage" type="xs:string"/>
            <xs:element name="circular" type="xs:boolean"/>
            <xs:element name="direction" type="xs:string"/>
            <xs:element name="ignoreNullValues" type="xs:boolean"/>
            <xs:element name="ignoreZeroValues" type="xs:boolean"/>
            <xs:element name="interiorGap" type="xs:double"/>
            <xs:element name="sectionOutLinePaint"
type="secPaintType" maxOccurs="unbounded"/>
            <xs:element name="sectionOutlineVisible" type="xs:boolean"/>
        >
            <xs:element name="shadowPaint" type="xs:string"/>
            <xs:element name="shadowXOffset" type="xs:double"/>
            <xs:element name="shadowYOffset" type="xs:double"/>
            <xs:element name="startAngle" type="xs:double"/>
            <xs:element name="labelPie" type="labelPieType"/>
            <xs:element name="legendPie" type="legendType"/>
            <xs:element name="explode" type="explodeType"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="labelPieType">
        <xs:sequence>
            <xs:element name="displaykey" type="xs:boolean"/>
            <xs:element name="displayValue" type="xs:boolean"/>
            <xs:element name="displayPercent" type="xs:boolean"/>
            <xs:element name="numberFormat" type="xs:double"
minOccurs="0"/>
            <xs:element name="percentFormat" type="xs:double"
minOccurs="0"/>
            <xs:element name="backColor" type="xs:string"/>
            <xs:element name="outColor" type="xs:string"/>
            <xs:element name="labelColor" type="xs:string"/>
            <xs:element name="shadowColor" type="xs:string"/>
            <xs:element name="stroke" type="xs:boolean"/>
            <xs:element name="padding" type="borderType"/>
            <xs:element name="labelFont" type="fontType"/>
        </xs:sequence>
    </xs:complexType>

```

Παράρτημα Α

```

        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="secPaintType">
        <xs:sequence>
            <xs:element name="keySection" type="xs:int"/>
            <xs:element name="secPaint" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="explodeType">
        <xs:sequence>
            <xs:element name="keySection" type="xs:int"/>
            <xs:element name="percent" type="xs:double"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="categoryPlotType">
        <xs:sequence>
            <xs:element name="axisOffset" type="borderType"/>
            <xs:element name="renderingOrderDataset" type="xs:string"/>
            <xs:element name="renderingOrderColumns" type="xs:string"/>
            <xs:element name="renderingOrderRows" type="xs:string"/>
            <xs:element name="domainGridLinesVisible"
type="xs:boolean"/>
            <xs:element name="domainGridLinePaint" type="xs:string"/>
            <xs:element name="domainGridLineStroke"
type="xs:boolean"/>
            <xs:element name="rangeGridLinesVisible" type="xs:boolean"/>
            <xs:element name="rangeGridLinePaint" type="xs:string"/>
            <xs:element name="rangeGridLineStroke" type="xs:boolean"/>
            <xs:element name="backgroundPaint" type="xs:string"/>
            <xs:element name="weight" type="xs:int"/>
            <xs:element name="foregroundAlpha" type="xs:float"/>
            <xs:element name="anchorValue" type="xs:double"/>
            <xs:element name="outlinePaint" type="xs:string"/>
            <xs:element name="outlineStroke" type="xs:boolean"/>
            <xs:element name="outlineVisible" type="xs:boolean"/>
            <xs:element name="domainMarker" type="markerType"/>
            <xs:element name="rangeMarker" type="markerType"/>
            <xs:element name="labelVisible" type="xs:boolean"/>
            <xs:element name="label" type="labelType"/>
            <xs:element name="annotation" type="categoryAnnotations"/>
            <xs:element name="legendCat" type="legendType"/>
            <xs:element name="catAxis" type="categoryAxisType"
minOccurs="0" maxOccurs="1"/>
            <xs:element name="valAxis" type="valueAxisType"
minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="legendType">
        <xs:sequence>
            <xs:element name="backgroundPaint" type="xs:string"/>
            <xs:element name="border" type="borderType"/>
            <xs:element name="height" type="xs:double"/>
            <xs:element name="width" type="xs:double"/>

```

Παράρτημα Α

```
<xs:element name="align" type="xs:string"/>
<xs:element name="margin" type="borderType"/>
<xs:element name="padding" type="borderType"/>
<xs:element name="position" type="xs:string"/>
<xs:element name="font" type="fontType"/>
<xs:element name="paint" type="xs:string"/>
<xs:element name="itemLabelPadding" type="borderType"
minOccurs="0"/>
<xs:element name="itemGraphicAnchorPos" type="xs:string"
minOccurs="0"/>
<xs:element name="itemGraphicEdgePos" type="xs:string"
minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="categoryAnnotations">
<xs:sequence>
<xs:element name="categoryLine" type="categoryLineType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="categoryText" type="categoryTextType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="PointerAnnotation"
type="pointerAnnotationType" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="categoryLineType">
<xs:sequence>
<xs:element name="key1" type="xs:string"/>
<xs:element name="value1" type="xs:double"/>
<xs:element name="key2" type="xs:string"/>
<xs:element name="value2" type="xs:double"/>
<xs:element name="linePaint" type="xs:string"/>
<xs:element name="lineStroke" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="categoryTextType">
<xs:sequence>
<xs:element name="textAnnotation" type="xs:string"/>
<xs:element name="keyAnnotation" type="xs:string"/>
<xs:element name="valueAnnotation" type="xs:double"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="pointerAnnotationType">
<xs:sequence>
<xs:element name="textAnnotation" type="xs:string"/>
<xs:element name="keyAnnotation" type="xs:string"/>
<xs:element name="valueAnnotation" type="xs:double"/>
<xs:element name="anchor" type="xs:double"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="labelType">
<xs:sequence>
<xs:element name="labelFormat" type="labelFormatType"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="labelFormatType">
```

Παράρτημα Α

```
<xs:sequence>
  <xs:element name="displayRowKey" type="xs:boolean"/>
  <xs:element name="displayColKey" type="xs:boolean"/>
  <xs:element name="displayValue" type="xs:boolean"/>
  <xs:element name="displayPercent" type="xs:boolean"/>
  <xs:element name="displayLabelAll" type="xs:boolean"/>
  <xs:element name="labelToAll" type="labelAllType"
minOccurs="0"/>
  <xs:element name="labelToSeries" type="labelSeriesType"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="labelSeriesType">
  <xs:sequence>
    <xs:element name="series" type="xs:int"/>
    <xs:element name="displayRowKey" type="xs:boolean"/>
    <xs:element name="displayColKey" type="xs:boolean"/>
    <xs:element name="visible" type="xs:boolean"/>
    <xs:element name="labelText" type="xs:string"/>
    <xs:element name="labelPaint" type="xs:string" />
    <xs:element name="outlinePaint" type="xs:string"/>
    <xs:element name="outlineStroke" type="xs:boolean"/>
    <xs:element name="basePaint" type="xs:string" />
    <xs:element name="baseSeriesVisible" type="xs:boolean" />
    <xs:element name="baseSeriesVisibleInLegend"
type="xs:boolean"/>
    <xs:element name="baseStroke" type="xs:boolean" />
    <xs:element name="labelFont" type="fontType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="strokeSeriesType">
  <xs:sequence>
    <xs:element name="index" type="xs:int"/>
    <xs:element name="strokeSeries" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="fontSeriesType">
  <xs:sequence>
    <xs:element name="seriesIndex" type="xs:int"/>
    <xs:element name="seriesFont" type="fontType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="colorSeriesType">
  <xs:sequence>
    <xs:element name="seriesIndex" type="xs:int"/>
    <xs:element name="sseriesLabelPaint" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="booleanSeriesType">
  <xs:sequence>
    <xs:element name="seriesIndex" type="xs:int"/>
    <xs:element name="setVisible" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="labelAllType">
```

Παράρτημα Α

```

<xs:sequence>
  <xs:element name="labelText" type="xs:string"/>
  <xs:element name="labelFont" type="fontType"/>
  <xs:element name="labelPaint" type="xs:string"/>
  <xs:element name="outlinePaint" type="xs:string"/>
  <xs:element name="outlineStroke" type="xs:boolean"/>
  <xs:element name="basePaint" type="xs:string"/>
  <xs:element name="baseSeriesVisible" type="xs:boolean"/>
  <xs:element name="baseSeriesVisibleInLegend"
type="xs:boolean"/>
  <xs:element name="baseStroke" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="markerType">
  <xs:sequence>
    <xs:element name="comparableKey" type="xs:string"
minOccurs="0"/>
    <xs:element name="markerValue" type="xs:double"
minOccurs="0"/>
    <xs:element name="drawLine" type="xs:boolean"/>
    <xs:element name="alpha" type="xs:float"/>
    <xs:element name="labelMarker" type="xs:string"/>
    <xs:element name="labelAnchorPos" type="xs:string"/>
    <xs:element name="labelOffset" type="xs:double"/>
    <xs:element name="labelPaint" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="chartPropertiesType">
  <xs:sequence>
    <xs:element name="antiAlias" type="xs:boolean"/>
    <xs:element name="backGroundPaint" type="xs:string"/>
    <xs:element name="borderColor" type="xs:string"/>
    <xs:element name="borderVisible" type="xs:boolean"/>
    <xs:element name="borderStroke" type="xs:boolean"/>
    <xs:element name="padding" type="borderType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="abstractRenderer">
  <xs:sequence>
    <xs:element name="autoPopupateSeriesFillPaint"
type="xs:boolean"/>
    <xs:element name="autoPopupateSeriesOutlinePaint"
type="xs:boolean"/>
    <xs:element name="autoPopupateSeriesOutlineStroke"
type="xs:boolean"/>
    <xs:element name="autoPopupateSeriesPaint"
type="xs:boolean"/>
    <xs:element name="autoPopupateSeriesStroke"
type="xs:boolean"/>
    <xs:element name="baseCreateEntities" type="xs:boolean"/>
    <xs:element name="baseFillPaint" type="xs:string"/>
    <xs:element name="baseItemLabelFont" type="fontType"/>
    <xs:element name="baseItemLabelVisible" type="xs:boolean"/>
    <xs:element name="baseOutlinePaint" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

Παράρτημα Α

```

        <xs:element name="baseOutlineStroke" type="xs:boolean"/>
        <xs:element name="basePaint" type="xs:string"/>
        <xs:element name="baseSeriesVisible" type="xs:boolean"/>
        <xs:element name="baseSeriesVisibleInLegend"
type="xs:boolean"/>
        <xs:element name="baseStroke" type="xs:boolean"/>
        <xs:element name="seriesCreateEntities"
type="booleanSeriesType"/>
        <xs:element name="seriesFillPaint" type="colorSeriesType"/>
        <xs:element name="seriesItemLabelFont"
type="fontSeriesType"/>
        <xs:element name="seriesItemLabelPaint"
type="colorSeriesType"/>
        <xs:element name="seriesItemLabelVisible"
type="booleanSeriesType"/>
        <xs:element name="seriesOutlinePaint"
type="colorSeriesType"/>
        <xs:element name="seriesOutlineStroke"
type="strokeSeriesType"/>
        <xs:element name="seriesPaint" type="colorSeriesType"/>
        <xs:element name="seriesStroke" type="strokeSeriesType"/>
        <xs:element name="seriesVisible" type="booleanSeriesType"/
>
        <xs:element name="seriesVisibleInLegend"
type="booleanSeriesType"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="axisType">
    <xs:sequence>
        <xs:element name="axisLinePaint" type="xs:string"/>
        <xs:element name="axisLineStroke" type="xs:boolean"/>
        <xs:element name="labelPaint" type="xs:string"/>
        <xs:element name="tickLabelPaint" type="xs:string"/>
        <xs:element name="tickLabelVisible" type="xs:boolean"/>
        <xs:element name="tickMarkInsideLength" type="xs:float"/>
        <xs:element name="tickMarkOutsideLength" type="xs:float"/>
        <xs:element name="tickMarkPaint" type="xs:string"/>
        <xs:element name="tickMarkStroke" type="xs:boolean"/>
        <xs:element name="tickMarkVisible" type="xs:boolean"/>
        <xs:element name="axisVisible" type="xs:boolean"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="categoryAxisType">
    <xs:complexContent>
        <xs:extension base="axisType">
            <xs:sequence>
                <xs:element name="categoryMargin"
type="xs:double"/>
                <xs:element name="lowerMargin"
type="xs:double"/>
                <xs:element name="maxCatLabelLines"
type="xs:int"/>
                <xs:element name="maxCatLabelWidthRatio"
type="xs:float"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

Παράρτημα Α

```
type="xs:double"/>
type="xs:string"/>
<xs:element name="upperMargin"
<xs:element name="categoryLabelPosition"
<xs:element name="location" type="xs:string"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="valueAxisType">
  <xs:complexContent>
    <xs:extension base="axisType">
      <xs:sequence>
        <xs:element name="autoRange"
        <xs:element name="autoRangeMinSize"
        <xs:element name="fixedAutoRange"
        <xs:element name="inverted"
        <xs:element name="lowerBound"
        <xs:element name="lowerMargin"
        <xs:element name="negativeArrowVisible"
        <xs:element name="positiveArrowVisible"
        <xs:element name="rangeAboutValue"
        <xs:element name="upperBound"
        <xs:element name="upperMargin"
        <xs:element name="verticalTickLabels"
        <xs:element name="autoRangeIncludeZero"
        <xs:element name="autoRangeStickyZero"
        <xs:element name="location" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="charts" type="chartsType">
  <xs:annotation>
    <xs:documentation>This document holds the parameters in
order to create a chart</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:schema>
```