



Α.Τ.Ε.Ι. ΗΡΑΚΛΕΙΟΥ

ΣΧΟΛΗ ΤΕΧΝΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ

-- ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ--

**" Διαδραστική απεικόνιση βάσης δεδομένων,
με αντικειμενοστρεφή προγραμματισμό "**



 **NetBeans**


Java


MySQL

ΣΠΟΥΔΑΣΤΕΣ: ΑΝΑΠΛΙΩΤΗΣ ΧΑΡΑΛΑΜΠΟΣ

ΓΕΡΝΑ ΠΗΝΕΛΟΠΗ

ΕΙΣΗΓΗΤΗΣ: Δ. ΑΚΟΥΜΙΑΝΑΚΗΣ

ΗΡΑΚΛΕΙΟ, ΑΠΡΙΛΙΟΣ 2009

Ευχαριστίες

Στο σημείο αυτό θα θέλαμε να ευχαριστήσουμε όλους τους ανθρώπους αυτούς, που μας βοήθησαν και συντέλεσαν στη διεκπεραίωση αυτής της πτυχιακή εργασίας.

Θα θέλαμε να ευχαριστήσουμε αρχικά τον καθηγητή κ. Δημοσθένη Ακουμιανάκη διδάσκοντα του Α.Τ.Ε.Ι Κρήτης και επιβλέποντα μας, για την βοήθεια και καθοδήγηση του καθ' όλη τη διάρκεια αυτής της εργασίας.

Και τέλος, ένα μεγάλο ευχαριστώ στους δικούς μας ανθρώπους, οικογένεια και φίλους που στάθηκαν δίπλα μας σε όλη τη διάρκεια της πορείας μας μέχρι εδώ και σε όλους εκείνους που βοήθησαν ο καθένας με τον δικό του τρόπο.

Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ	3
ΠΕΡΙΕΧΟΜΕΝΑ	4
ΚΕΦΑΛΑΙΟ 1	
Εισαγωγή.....	7
ΚΕΦΑΛΑΙΟ 2	
Υπόβαθρο και τεχνολογικές προσεγγίσεις.....	8
2.1 Προβλήματα κλασικών βάσεων δεδομένων.....	8
2.2 Εργασίες και συστήματα που αναφέρονται σε Database Visualization.....	9
2.2.1 Εφαρμογές σε χάρτες σκέψεων.....	9
2.2.2 Εφαρμογές στην παρουσίαση ειδήσεων.....	10
2.2.3 Εφαρμογές στην παρουσίαση δεδομένων.....	11
2.2.4 Εφαρμογές παρουσίασης συνδέσεων.....	15
2.2.5 Εφαρμογές παρουσίασης ιστοσελίδων.....	18
2.2.6 Εργαλεία για απεικόνιση δεδομένων.....	19
2.3.6.1 Prefuse.....	19
2.3.6.2 Swivel.....	20
2.3.6.3 JGraph.....	20
2.3 Στόχος και προσέγγιση παρούσας εργασίας.....	20
ΚΕΦΑΛΑΙΟ 3	
Αρχιτεκτονική Συστήματος.....	21
3.1 Περιγραφή χρησιμοποιηθέντων τεχνολογιών.....	21
3.1.1 Αντικειμενοστραφής προγραμματισμός.....	21
3.1.1.1 Objects.....	24
3.1.1.2 Classes (Κλάσεις).....	24
3.1.1.3 Properties (Ιδιότητες).....	25

3.1.1.4 Methods (Μέθοδοι).....	25
3.1.1.5 Events (Γεγονότα).....	26
3.1.1.6 Inheritance (Κληρονομικότητα).....	26
3.1.1.7 Encapsulation.....	26
3.1.1.8 Polymorphism (Πολυμορφισμός).....	27
3.1.2 Java.....	27
3.1.3 NetBeans.....	31
3.1.4 Sql.....	32
3.1.5 MySql.....	32
3.2 Αναπτυχθέντα Τμήματα.....	34
 ΚΕΦΑΛΑΙΟ 4	
Σενάρια Χρήσης.....	35
4.1 Δυνατότητα Αναζήτησης Πακέτου στη Βάση Δεδομένων.....	35
4.2 Δυνατότητα Προσθήκης Πακέτου στη Βάση Δεδομένων.....	35
4.3 Δυνατότητα Προβολής των Σχετικών με την Εφαρμογή.....	36
4.4 Δυνατότητα Εξόδου από την Εφαρμογή.....	36
 ΚΕΦΑΛΑΙΟ 5	
Συμπεράσματα.....	37
 ΚΕΦΑΛΑΙΟ 6	
Βιβλιογραφία.....	38
 ΚΕΦΑΛΑΙΟ 7	
Παράρτημα.....	39
7.1 Κώδικας Εφαρμογής.....	39
7.1.1 Η σύνδεση με τη βάση δεδομένων (DBConnection).....	39
7.1.2 MainFrame.....	40
7.1.3 AddPanel.....	44
7.1.4 SearchPanel.....	49
7.1.5 PackagePanel.....	62

7.1.6 PackageViewer.....	71
7.1.7 DeployDayViewer.....	75
7.1.8 DetailViewer.....	77
7.1.9 DayPanel.....	79
7.1.10 NoIconTreeCellrenderer.....	81
7.1.11 DeployDayData.....	82
7.1.12 PackageData.....	83
7.1.13 PackageDetailsData.....	85
7.1.14 CategoryNode.....	87
7.1.15 Settings.....	89
7.2 Πρακτικό Παράδειγμα Της Εφαρμογής.....	90

-- ΚΕΦΑΛΑΙΟ 1 --

ΕΙΣΑΓΩΓΗ

Το αντικείμενο της πτυχιακής εργασίας έχει να κάνει με τον σχεδιασμό και υλοποίηση μιας αντικειμενοστρεφούς εφαρμογής, η οποία μας δίνει την δυνατότητα της διαδραστικής αναζήτησης, προβολής και επεξεργασίας ταξιδιωτικών πακέτων της ηλεκτρονικής κοινότητας (e-village) eKoNES.

Η εφαρμογή είναι ευέλικτη, αφού υπάρχει και η δυνατότητα εύκολης τροποποίησης του περιεχομένου του μια και αυτό αντλείται από μια βάση δεδομένων.

Σκοπός είναι η κατασκευή ενός περιβάλλοντος, το οποίο να αποκρίνεται σε όλες τις κινήσεις του χρήστη, αλλά και παράλληλα να ανταποκρίνεται σε όλες τις απαιτήσεις της εφαρμογής.

Έτσι χρησιμοποιώντας την βάση δεδομένων που υποστηρίζει την ηλεκτρονική κοινότητα (<http://www.e-kones.teiher.gr/web/Village/Home>) του χωριού eKoNES, έχουμε την δυνατότητα της αυτοματοποιημένης χρονολογικής ταξινόμησης και προβολής κάθε πακέτου σε μορφή ετήσιου ημερολογίου.

Επίσης, η εφαρμογή μας δίνει την δυνατότητα να προσθέσουμε τέτοια πακέτα στην βάση δεδομένων μας χρησιμοποιώντας δυναμικά ερωτήματα της γλώσσας MySQL και την ευελιξία της αντικειμενοστραφούς γλώσσας προγραμματισμού Java.

Τέλος, για την ανάπτυξη της εφαρμογής και κυρίως για το γραφικό κομμάτι της διεπαφής μας στηριχθήκαμε στο προγραμματιστικό περιβάλλον ανάπτυξης Java εφαρμογών "NetBeans 6.5".

-- ΚΕΦΑΛΑΙΟ 2 --

Υπόβαθρο και τεχνολογικές προσεγγίσεις

Γενικά μία **βάση δεδομένων** είναι μία αποθήκη πληροφοριών . Είναι κατά τον ορισμό του ν. 2121/1993 περί [Πνευματικής Ιδιοκτησίας](#) "η συλλογή έργων, δεδομένων ή άλλων ανεξάρτητων στοιχείων, διευθετημένων κατά συστηματικό ή μεθοδικό τρόπο και ατομικώς προσιτών με ηλεκτρονικά μέσα ή με άλλο τρόπο".

Οι βάσεις δεδομένων έχουν ιδιαίτερη σημασία καθώς είναι ο μόνος τρόπος για να αποθηκεύσουμε και ταξινομήσουμε πληθώρα πληροφοριών, τις οποίες χωρίς ταξινόμηση θα ήταν αδύνατο να χρησιμοποιήσει ή να επεξεργαστεί κανείς. Εάν θέλαμε ένα πρόγραμμα το οποίο θα το χρησιμοποιούσαμε μόνο μία φορά ή σχεδιάζαμε κάτι προσωρινό που δεν θα το χρησιμοποιούσαμε ξανά τότε δεν θα ήταν αναγκαία η ύπαρξη της βάσης. Για το λόγω όμως ότι θα θέλαμε στο μέλλον να ασχοληθούμε ξανά με τα δεδομένα μας, η ύπαρξη της βάσης κρίνεται αναγκαία.

Με την Access μπορούμε :

- Να αποθηκεύουμε δεδομένα σε πίνακες (tables) οι οποίοι αποτελούνται από διάφορες στήλες και γραμμές
- Να ανακτούμε δεδομένα μέσα από τους πίνακες της βάσης
- Να συνδέουμε τους πίνακες μεταξύ τους για την ανάκτηση δεδομένων που υπάρχουν σε άλλους πίνακες

2.1 Προβλήματα κλασικών βάσεων δεδομένων

Ένα από τα βασικότερα προβλήματα σε εφαρμογές ομότιμων βάσεων δεδομένων είναι η γρήγορη και ακριβής αντιστοίχιση δεδομένων που είναι εκφρασμένα σε εν γένει διαφορετικές μορφές, ο τρόπος δηλαδή που μπορεί να ξεπεραστεί το πρόβλημα της ετερογένειας δεδομένων. Στις περισσότερες περιπτώσεις, τα συστήματα ομότιμων Βάσεων Δεδομένων χρησιμοποιούν δηλωτικά ερωτήματα – όψεις, που φανερώνουν τις αντιστοιχίες πινάκων των δύο σχημάτων σε conjunctive μορφή. Παρόλα αυτά κρίνεται απαραίτητη η ανάπτυξη μιας διαδικασίας (ημι)αυτόματης παραγωγής των αντιστοιχιών αυτών, ώστε η αντιστοίχιση δύο σχημάτων να γίνεται με τη μικρότερη δυνατή παρέμβαση του χρήστη.

2.2 Εργασίες και συστήματα που αναφέρονται σε Database Visualization

Με τον όρο **Data Visualization** αναφερόμαστε στην παρουσίαση δεδομένων και συγκεντρωτικών πληροφοριών (summary information) χρησιμοποιώντας γραφήματα, κίνηση, τρισδιάστατες απεικονίσεις και άλλα εργαλεία πολυμέσων.

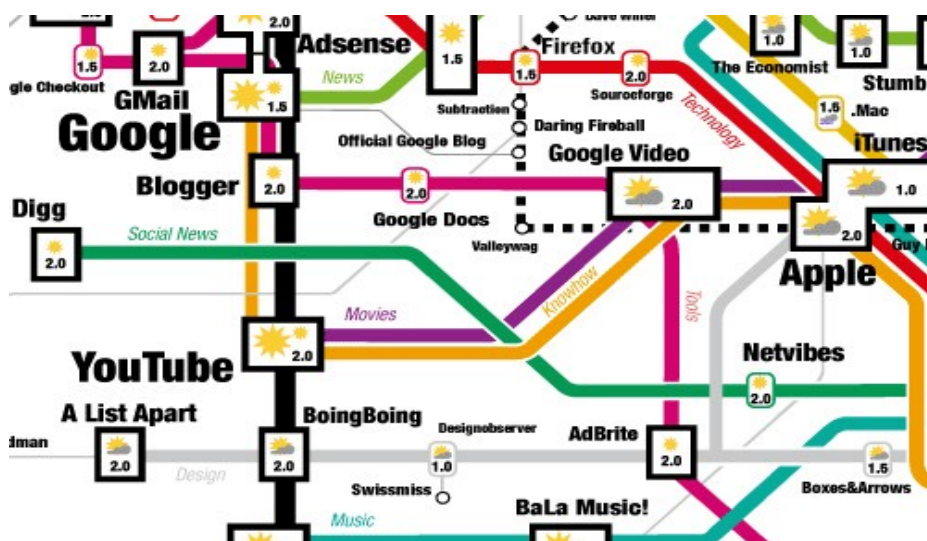
Η *παρουσίαση των δεδομένων* (data visualization) ανήκει στον κλάδο Αλληλεπίδρασης Ανθρώπου-Υπολογιστή της επιστήμης των υπολογιστών. Ο κλάδος αυτός ασχολείται με τις προκλήσεις που παρουσιάζονται στο να γίνουν οι υπολογιστές και οι υπολογισμοί χρήσιμοι, εύκολοι στη χρήση και ευρέως προσπελάσιμοι προς τους χρήστες.

Η υπολογιστική ισχύ των υπολογιστών έχει αυξηθεί πολύ στις μέρες μας, συνεπώς και οι δυνατότητες αποθήκευσης και επεξεργασίας μεγάλου όγκου δεδομένων. Οι δυνατότητες αυτές γίνονται πιο αποτελεσματικές όταν συνδυάζονται με κατάλληλη παρουσίαση της πληροφορίας προς τον χρήστη.

Τα **γραφικά** είναι ένας παράγοντας που έχει βοηθήσει αρκετά στην παρουσίαση των δεδομένων. Καθημερινά παρουσιάζονται καινούργιοι και πιο σύγχρονοι τρόποι-τεχνικές απεικόνισης των δεδομένων. Οι επιλογές σε τεχνικές απεικόνισης σήμερα είναι πολλές και πραγματικά καταφέρνουν να παρουσιάσουν τα δεδομένα εντυπωσιακά, συναρπαστικά και ιδιαίτερα περιγραφικά. Ας δούμε μερικά παραδείγματα από την εφαρμογή των τεχνικών αυτών στο διαδίκτυο.

2.2.1 Εφαρμογές σε χάρτες σκέψων

Η παρακάτω απεικόνιση δεδομένων (Εικ.1) παρουσιάζει τις 200 δημοφιλέστερες ιστοσελίδες στο διαδίκτυο, δομημένες ανά κατηγορία, σχέσης, επιτυχίας, δημοσιότητας και προοπτικής ως ένα χάρτη συγκοινωνιών (μετρό). Όσο πιο μεγάλος-κεντρικός σταθμός τόσο πιο δημοφιλή είναι η ιστοσελίδα.



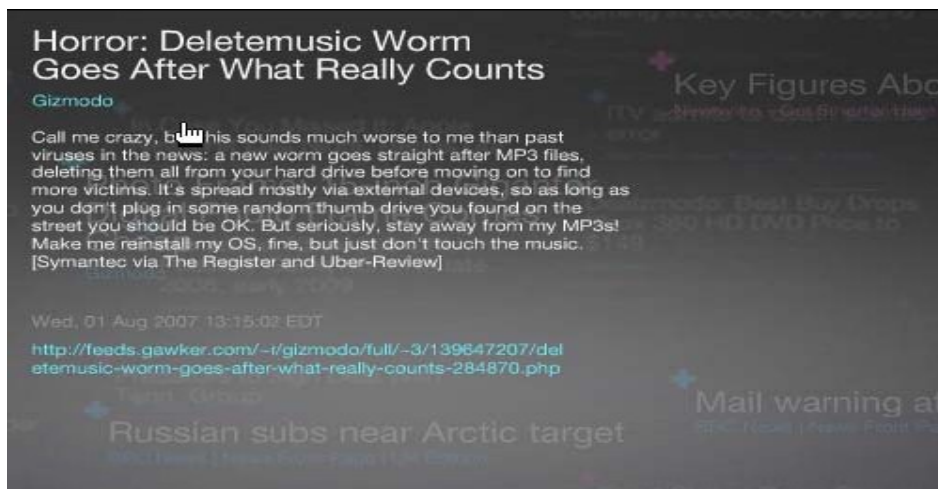
2.2.2 Εφαρμογές στην παρουσίαση ειδήσεων

Το Newsmap (Εικ.2) είναι μια εφαρμογή που αντιδράει αλλάζοντας οπτικά το χάρτη των ειδήσεων του Google. Τα διάφορα χρώματα απεικονίζουν τα διάφορα θέματα ειδήσεων(π.χ. πορτοκαλί ίσον τοπικά, κόκκινο ίσον εξωτερικά κτλ), όσο πιο ανοιχτό είναι το χρώμα τόσο πιο φρέσκια είναι η είδηση, ενώ το μέγεθος της γραμματοσειράς δείχνει πόσο δημοφιλές είναι το κάθε θέμα την συγκεκριμένη στιγμή.



Εικ.2 : Εφαρμογή απεικόνισης δεδομένων Newsmap.

Το Voyage (Εικ.3) είναι ένα RSS - feeder το οποίο παρουσιάζει τα πιο πρόσφατα νέα υπό μορφή διαφάνειας. Κάνοντας zoom ο χρήστης εισέρχεται σε άλλες διαφάνειες με παλαιότερα θέματα. Η πλοήγηση μπορεί να γίνει και με timeline, μια μπάρα που φέρει επάνω της όλες τις διαφάνειες με τα θέματα των ειδήσεων ανά μέρα.



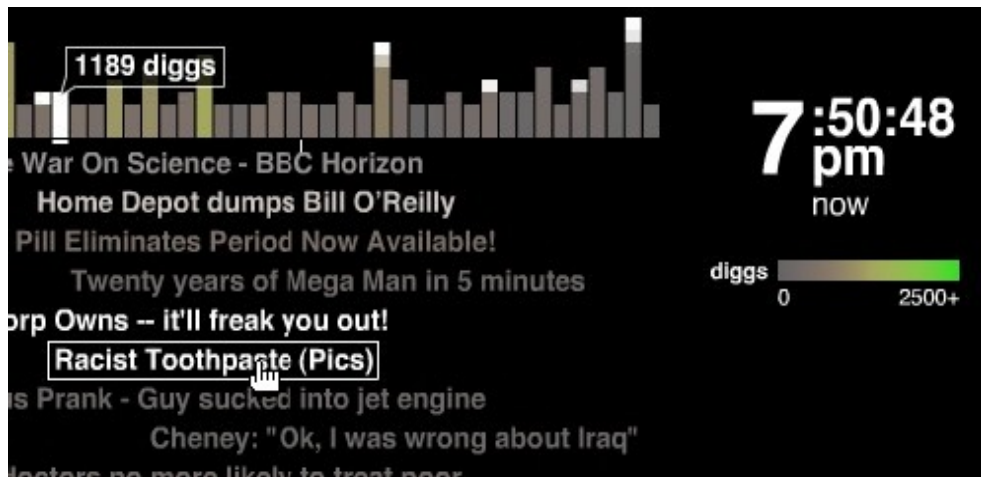
Εικ.3 : Εφαρμογή απεικόνισης δεδομένων Voya.

Το Digg BigSpy (Εικ.4) τακτοποιεί διάφορες ιστορίες από το διαδίκτυο ανάλογα με το πόσο ο κόσμος της αναζητεί. Όσο πιο μεγάλες τόσο πιο πολλές αναζητήσεις, παράλληλα γράφετε και ο αριθμός των αναζητήσεων.



Εικ.4 : Εφαρμογή απεικόνισης δεδομένων Digg BigSpy.

Το Digg Stack (Εικ.5) είναι παρόμοιο με το Digg BigSpy. Εδώ παρουσιάζονται οι αναζητήσεις θεμάτων σε στοίβες. Όσο πιο πολλές αναζητήσεις τόσο μεγαλύτερες γίνονται οι στοίβες.



Εικ.5 : Εφαρμογή απεικόνισης δεδομένων Digg Stack.

2.2.3 Εφαρμογές στην παρουσίαση δεδομένων

Το Amaztype (Εικ.6) εκτελεί μια τυπική αναζήτηση βιβλίων απεικονίζοντας τα αποτελέσματα όπως την λέξη αναζήτησης που του δίνουμε, αν θέλουμε να πάρουμε παραπάνω πληροφορία για ένα βιβλίο απλά το επιλέγουμε με το ποντίκι.



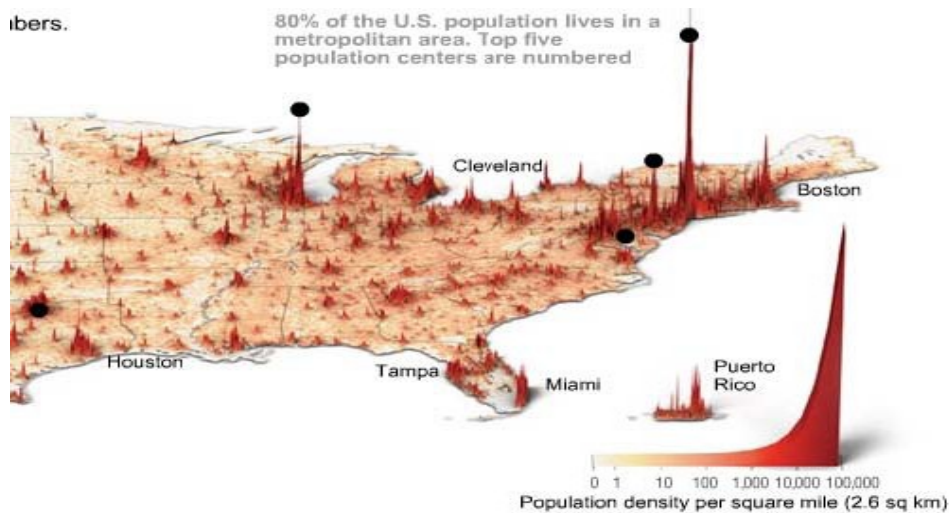
Εικ.6 : Εφαρμογή απεικόνισης δεδομένων Amaztype.

Παρόμοια ιδέα χρησιμοποιεί το flickertime (Εικ.7) όπου τα δεδομένα είναι φωτογραφίες που ανεβάζουν οι επισκέπτες της σελίδας και η παρουσίαση είναι σε μορφή της ώρας εκείνης την στιγμή. Για επιπλέον πληροφορία για μια φωτογραφία απλά την επιλέγουμε.



Εικ.7 : Εφαρμογή απεικόνισης δεδομένων flickrtime.

Το Time Magazine (Εικ.8) κάνει χρήση οπτικών κορυφών σε ένα χάρτη για την αναπαράσταση δεδομένων, τα οποία είναι ο πληθυσμός των πόλεων μιας χώρας (εδώ της Αμερικής).



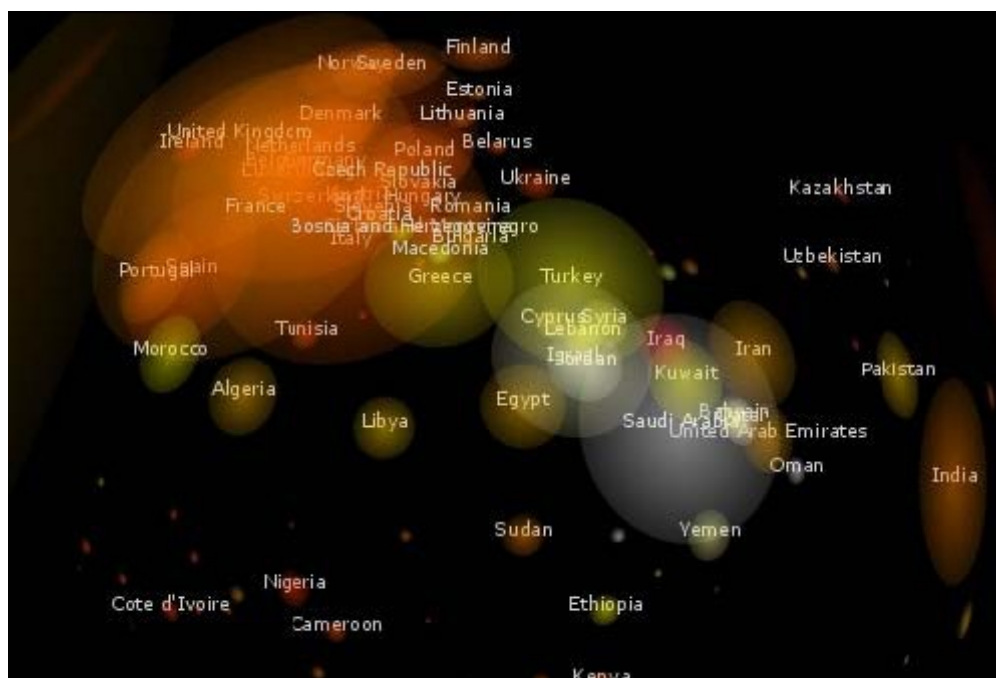
Εικ.8 : Εφαρμογή απεικόνισης δεδομένων Time Magazine.

Το crazyEgg (Εικ.9) μπορεί να χρησιμοποιηθεί σε μια σελίδα με σκοπό να καταγράψει την συμπεριφορά των επισκεπτών. Όσο πιο πολλά τα κλικ των επισκεπτών τόσο πιο κόκκινη γίνεται η περιοχή γύρω από το link (σύνδεση), κάτι σαν μετρητή του πιο καυτού link της σελίδας. Η εφαρμογή αυτή διαθέτει και άλλους τρόπους αναπαράστασης της πληροφορίας αυτής όπως σε λίστες ή σε μικρά παράθυρα.



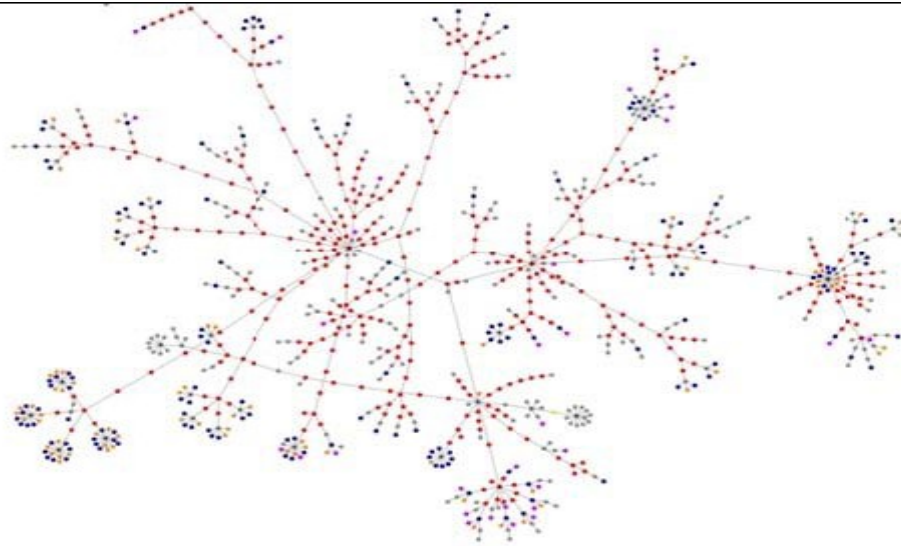
Εικ.9 : Εφαρμογή απεικόνισης δεδομένων crazyEgg.

Το Three Views (Εικ.10) παρουσιάζει δεδομένα σε χρωματιστούς κύκλους, τα δεδομένα είναι τα χρήματα που ξοδεύει η εκάστοτε χώρα για την αγορά στρατιωτικού εξοπλισμού(μέγεθος του κύκλου) και οι απολαβές από την χρήση του(χρώμα κύκλου).



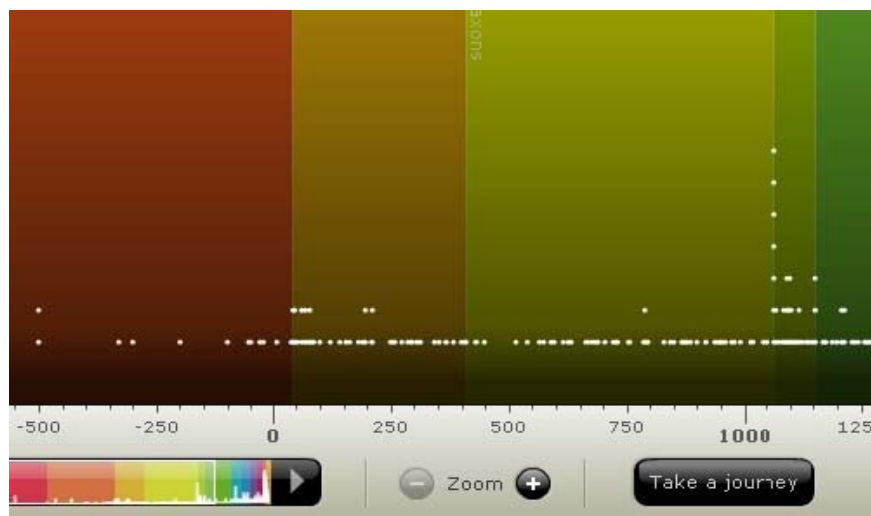
Εικ.10: Εφαρμογή απεικόνισης δεδομένων Three Views.

Στο Websites as graphs (Εικ.11), η εφαρμογή παρουσιάζει μια ιστοσελίδα σε μορφή γράφου. Η ιεραρχία του γράφου έχει κατανομηθεί βάση των ιδιοτήτων μιας σελίδας όπως συνδέσεις, πίνακες, ετικέτες, φωτογραφίες, φόρμες.



Εικ.11 : Εφαρμογή απεικόνισης δεδομένων Websites as graphs.

Το Interactive History Timeline (Εικ.12) είναι μια διαδραστική μπάρα χρόνου. Αυτή που εμφανίζεται στο σχήμα παρουσιάζει την ιστορία της Αγγλίας. Τα διάφορα χρώματα δείχνουν μια συγκεκριμένη περίοδο εποχής της χώρας(π.χ. πόλεμος).



Εικ. 12 : Εφαρμογή απεικόνισης δεδομένων Interactive History Timeline.

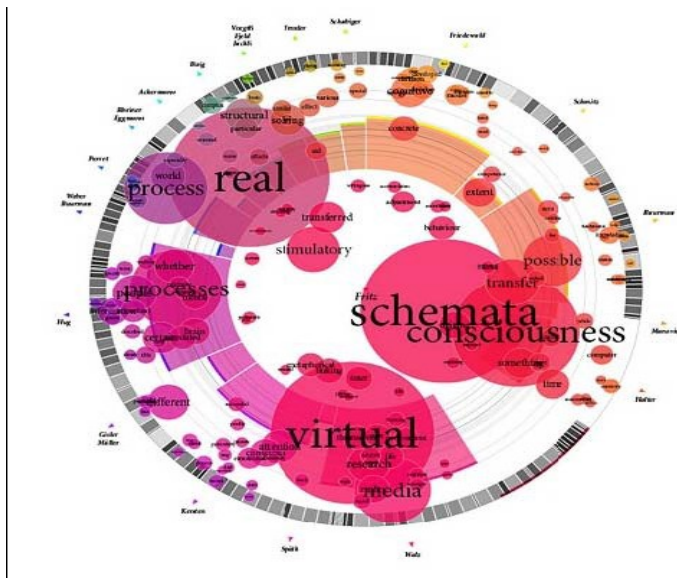
Το Elastic Lists (Εικ.13) παρουσιάζει ένα είδος ελαστικών λιστών με δεδομένα. Τα δεδομένα είναι όλοι οι νομπελίστες του κόσμου. Στην πρώτη λίστα οι νομπελίστες είναι χωρισμένοι ανά χώρα καταγωγής. Στη δεύτερη ο διαχωρισμός γίνεται ανά δεκαετίες. Και στην τελευταία λίστα γίνεται ανά έτος. Σε όλες τις περιπτώσεις το τελικό αποτέλεσμα εξαρτάται από την επιλογή που κάνουμε στην προηγούμενη λίστα και που έχει ως αποτέλεσμα να αλλάζει η επόμενη λίστα. Επίσης, υποστηρίζονται διάφοροι τρόποι φιλτραρίσματος (π.χ. φύλλο).

COUNTRY	all	filter	lock	DECADE	all	filter	lock	YEAR	all	filter	lock
United States of America				2000s				2004			
	281				56				12		
United Kingdom				1990s				2003			
	105				96				9		
France				1980s				2002			
	49				97				10		
Germany				1970s				2001			
	44				103				15		
Federal Republic of Germany				1960s				2000			
	31				79				10		
Switzerland				1950s				1999			
	30				72				7		
Sweden				1940s				1998			
	30				43				12		
the Netherlands				1930s				1997			
	15				56				12		
USSR				1920s				1996			
	14				54				13		
Italy				1910s				1995			
	14				40				10		
Denmark				1900s				1994			
	13				57				10		
Austria								1993			
	12								9		
Belgium								1992			
	11								7		
								1991			
									7		
								1990			
									11		

Εικ.13 : Εφαρμογή απεικόνισης δεδομένων Elastic Lists

2.2.4 Εφαρμογές παρουσίασης συνδέσεων

Το Munterbund (Εικ.14) παρουσιάζει τα αποτελέσματα έρευνας γραφικών απεικονίσεων από ομοιότητες κείμενων σε δοκίμια βιβλίων. Η πρόκληση είναι να βρεις φόρμες γραφικών και τυπογραφικών παρουσιάσεων των ομοιοτήτων αυτών που να είναι όμορφες και ενημερωτικές. Μια προσπάθεια να γίνει ένα τέτοιο σύστημα αυτόματης δημιουργίας γραφικών προκαθορισμένων κανόνων είναι και αυτή η εφαρμογή.



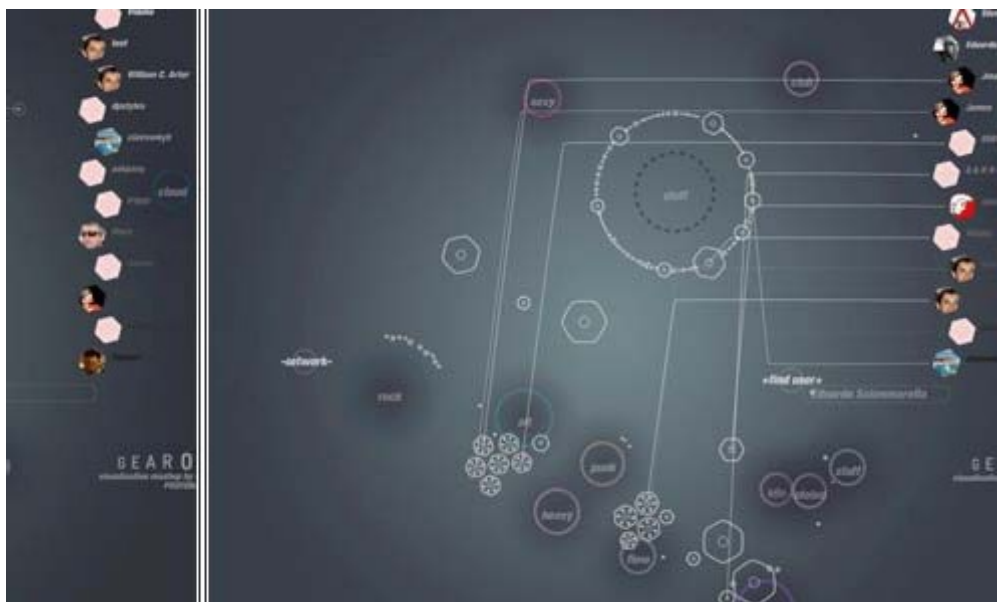
Εικ.14 : Εφαρμογή απεικόνισης δεδομένων Munterbund.

Το Burst Labs (Εικ.15) είναι μια εφαρμογή που προτείνει στους χρήστες όμοια ή συνδεδεμένα στοιχεία με την λέξη κλειδί αναζήτησης (αγαπημένοι τραγουδιστές, εκπομπές, ταινίες κτλ) μέσα σε μια φούσκα.



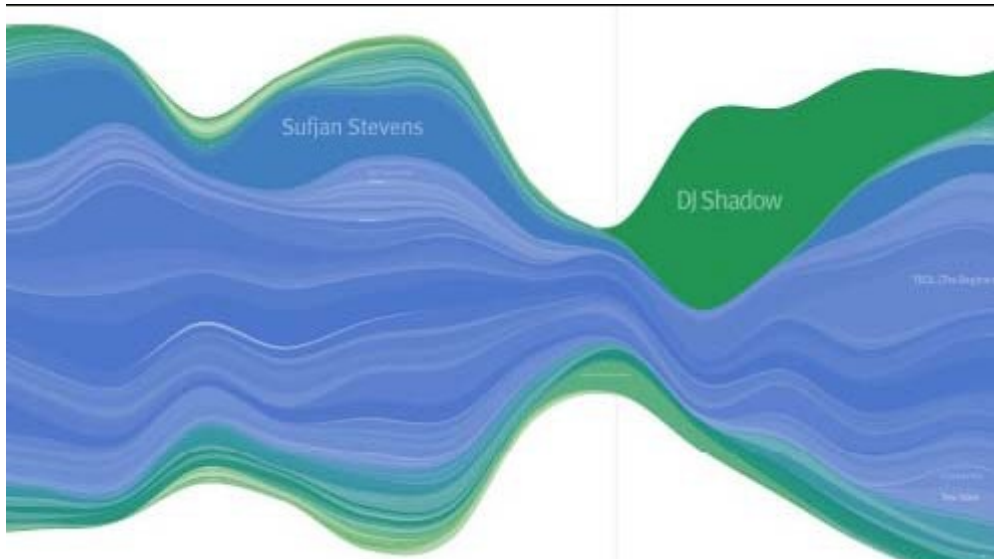
Εικ.15 : Εφαρμογή απεικόνισης δεδομένων Burst Labs.

Το Figd't Visualizer (Εικ.16) προσφέρει τη δυνατότητα οπτικής παρουσίασης ενός δικτύου. Η πληροφορία που παρουσιάζει ιδιαίτερα είναι οι τάσεις-προτιμήσεις μέσα στο διαδίκτυο των χρηστών. Όπως για παράδειγμα τι είδος μουσική ακούγεται περισσότερο (π.χ., ροκ ή κλασική).



Εικ.16 : Εφαρμογή απεικόνισης δεδομένων Figd't Visualizer.

Στην Εικ.17 παρουσιάζεται ένα ιστόγραμμα το οποίο προσπαθεί να καταγράψει κατά προσέγγιση την περιγραφή του τι είδος μουσική ακούει αυτός που το δημιούργησε (Lee Byron).

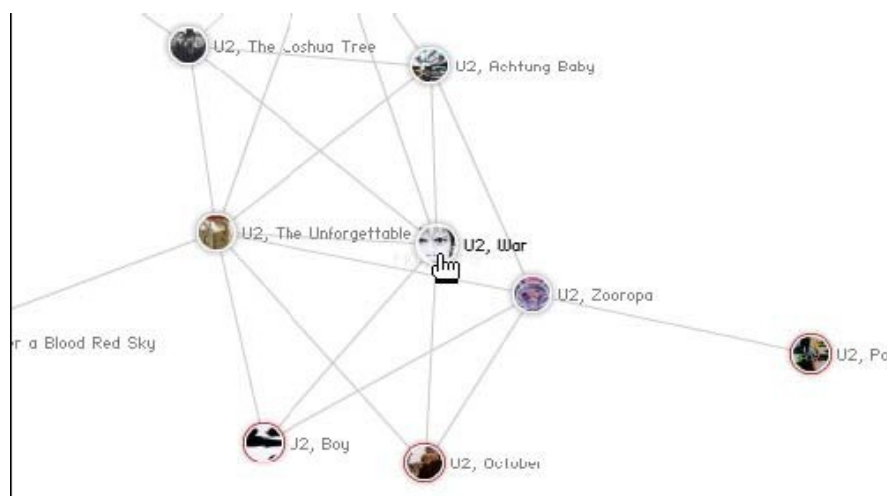


Εικ.17 : Εφαρμογή απεικόνισης δεδομένων του Lee Byron.

Shape of Song (σχήμα τραγουδιού) (Εικ.18), όπως λέει και το όνομα του παρουσιάζει ένα τραγούδι σε σχήμα. Τέλος, το Musicmap (Εικ.18), παρουσιάζει τις συνδέσεις ως συνδεδεμένες γραμμές, δημιουργώντας ένα δίκτυο.



Εικ.18 : Εφαρμογή απεικόνισης δεδομένων The Shape of Song.



Εικ.19 : Εφαρμογή απεικόνισης δεδομένων Musicmap.

2.2.5 Εφαρμογές παρουσίασης ιστοσελίδων

Η διαδραστική οπτική αναπαράσταση δεδομένων έχει χρησιμοποιηθεί ευρύτατα και σε εφαρμογές του Παγκόσμιου Ιστού. Ενδεικτικά το Spacetime (Εικ.20), παρουσιάζει το Google, Yahoo, Flickr, eBay και φωτογραφίες σε 3D απεικόνιση (3D πρόγραμμα πλοήγησης). Έτσι για παράδειγμα σε μια αναζήτηση του Google τα αποτελέσματα που θα βγουν θα είναι σε όμορφα 3D παράθυρα. Το UBrowser (Εικ.21), είναι μια εφαρμογή ανοιχτού κώδικα που προσφέρει ένα 3D πρόγραμμα πλοήγησης στο διαδίκτυο σε διάφορα σχήματα (π.χ. κύβος, κύκλος).



Εικ.20 : Εφαρμογή απεικόνισης δεδομένων Spacetime.



Εικ.21: Εφαρμογή απεικόνισης δεδομένων UBrowser.

2.2.6 Εργαλεία για απεικόνιση δεδομένων

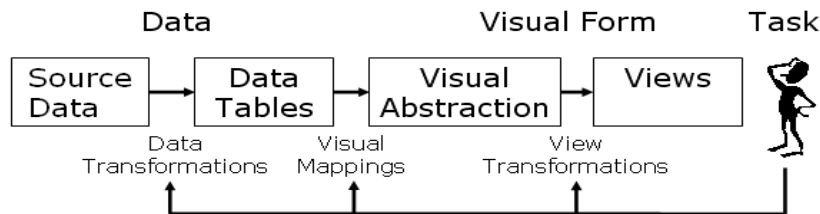
1 2.3.6.1 Prefuse

Ένα από τα σημαντικότερα εργαλεία (λογισμικό) για υλοποίηση πολλών από των παραπάνω διαδραστικών απεικονίσεων είναι το *perfuse*, χρησιμοποιεί την γλώσσα προγραμματισμού Java και είναι ανοιχτού κώδικα (*open source*). Διαθέτει επίσης και το *perfuse flare* εργαλείο που παρέχει αναπαραστάσεις και απεικονίσεις για *ActionScript* και το *Adobe Flash Player*. Το *perfuse* μπορεί να χρησιμοποιηθεί για να δημιουργήσει ανεξάρτητες εφαρμογές, απεικονίσεις ενσωματωμένες σε μεγαλύτερες εφαρμογές και *web applets*. Μερικά από τα κύρια χαρακτηριστικά του *perfuse* αναφέρονται παρακάτω.

1

- ✓ Πίνακες, γραφικές παραστάσεις, και δομές δεδομένων όπως δέντρα που υποστηρίζουν τις αυθαίρετες ιδιότητες στοιχείων, στοιχεία που συντάσσουν ευρετήριο και ερωτήσεις επιλογής, όλες με έναν πολύ αποδοτικό τρόπο διαχείρισης της μνήμης.
- ✓ Συστατικά για το σχεδιάγραμμα, το χρώμα, το μέγεθος, και την κωδικοποίηση μορφής, τις τεχνικές διαστρεβλώσεων, *animation*, και πολλά άλλα.
- ✓ Μια βιβλιοθήκη των ελέγχων αλληλεπίδρασης καθώς και διαδικασίες άμεσου-χειρισμού.
- ✓ Υποστήριξη *animation* μέσω μιας γενικής δραστηριότητας που σχεδιάζει το μηχανισμό.
- ✓ Δυνατότητα μετασχηματισμών που υποστηρίζουν *rapping* και *zooming* συμπεριλαμβανομένης της γεωμετρικής και σημασιολογικής μεγέθυνσης.
- ✓ Δυναμικά ερωτήματα (*query*) για το φιλτράρισμα των στοιχείων.
- ✓ Ενσωματωμένη αναζήτηση κειμένου που χρησιμοποιεί διάφορες διαθέσιμες μηχανές.
- ✓ Μια μηχανή προσομοίωσης για την δυναμική απεικόνιση και το *animation*.
- ✓ Ευελιξία για τις πολλαπλές εμφανίσεις, περιλαμβανομένου "*overview+detail*" και "*small multiples*" απεικονίσεις.
- ✓ Ενσωματωμένη γλώσσα σαν την *Sql* για την υποβολή ερωτημάτων στις δομές δεδομένων του *Prefuse* και για τη δημιουργία πεδίων με αντλημένα δεδομένα (*derived data fields*).
- ✓ Η υποστήριξη για την ισχύ των ερωτημάτων στις βάσεις δεδομένων *Sql* και τη χαρτογράφηση αυτών των αποτελεσμάτων σε δομές δεδομένων *Prefuse*.
- ✓ Απλά, φιλικά *APIs* για τη δημιουργία επεξεργασίας συγκεκριμένων διαδικασιών, αλληλεπιδράσεων, και *rendering* συστατικών.

Ο σχεδιασμός του εργαλείου *prefuse* είναι βασισμένος πάνω στο μοντέλο πληροφορία-απεικόνιση-αναφορά, μια αρχιτεκτονική λογισμικού που σπάει τη διαδικασία της απεικόνισης σε συγκεκριμένα βήματα, από την απόκτηση δεδομένων και μοντελοποίηση σε οπτική κωδικοποίηση δεδομένων στην παρουσίαση διαδραστικών προβολών. Αυτή η διαδικασία περιγράφεται στην παρακάτω εικόνα (Εικ.22) .



Εικ.28 : Διάγραμμα απεικόνισης της πληροφορίας του μοντέλου αναφοράς του *prefuse*.

2.3.6.2 Swivel

Αυτό το εργαλείο δημιουργεί εύκολα διαγράμματα πίτας (pie charts), απλά διαγράμματα (diagrams) και ιστογράμματα (histograms). Επίσης μέσω *Swivel API* διαθέτει την δυνατότητα προσθήκης είδη υπαρχόντων μεθόδων απεικόνισης.

2.3.6.3 JGraph

Το *JGraph* είναι ένα επιπλέον εργαλείο το οποίο μπορεί να απεικονίσει οπτικά δεδομένα.

2.3 Στόχος και προσέγγιση παρούσας εργασίας

Στόχος της παρούσας εργασίας είναι να σχεδιάσουμε μια εφαρμογή, η οποία μας δίνει την δυνατότητα της διαδραστικής αναζήτησης, προβολής και επεξεργασίας ταξιωτικών πακέτων της ηλεκτρονικής κοινότητας (e-village) *eKoNES*.

Μπορούμε δηλαδή, κάνοντας χρήση της βάσεως δεδομένων που υποστηρίζει την ηλεκτρονική κοινότητα του χωριού *eKoNES* (<http://www.e-kones.teiher.gr/web/Village/Home>), να προβάσουμε συγκεκριμένα πακέτα σύμφωνα με τα εκάστοτε κριτήρια που θέτουμε, χρονολογικά ταξινομημένα σε μορφή ετήσιου ημερολογίου.

Επίσης η χρήση δυναμικών ερωτημάτων της γλώσσας *MySQL* καθώς και η ευελιξία της αντικειμενοστραφούς γλώσσας προγραμματισμού *Java* μας επιτρέπει να δημιουργήσουμε ανά πάσα στιγμή κάποιο νέο πακέτο.

-- ΚΕΦΑΛΑΙΟ 3 --

Αρχιτεκτονική Συστήματος

Για την ανάπτυξη της εφαρμογής είναι απαραίτητη η ύπαρξη μιας Βάσης Δεδομένων στην οποία να είναι καταχωρημένα όλα τα στοιχεία που χρειάζονται για την σωστή και καλύτερη λειτουργία της υπηρεσίας.

Η διαχείριση της Βάσης μας θα γίνεται μέσω του γραφικού περιβάλλοντος της εφαρμογής μας. Έτσι θα πρέπει να αναπτύξουμε μια διεπαφή μέσω της οποίας θα μπορούμε να προσπελάσουμε την βάση, να κάνουμε αναζήτηση διαφόρων στοιχείων καθώς επίσης και να της προσθέσουμε νέα δεδομένα.

3.1 Περιγραφή χρησιμοποιηθέντων τεχνολογιών

Η γλώσσα που χρησιμοποιήθηκε για την ανάπτυξη του πρόγραμμα είναι **JAVA** η οποία αποτελεί μια υψηλού επιπέδου αντικειμενοστραφή γλώσσα προγραμματισμού.

Ενώ με την βοήθεια του προγραμματιστικού περιβάλλοντος ανάπτυξης Java εφαρμογών, "NetBeans 6.5" επιτεύχθηκε η περάτωση της εφαρμογής, και κυρίως για το γραφικό κομμάτι της διεπαφής μας.

3.1.1. ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Ο αντικειμενοστραφής προγραμματισμός (oop, object-oriented programming) είναι ένα μοντέλο γλώσσας προγραμματισμού οργανωμένο περισσότερο γύρω από τα δεδομένα παρά γύρω από τις ενέργειες και τη λογική του προγραμματισμού. εμφανίστηκε το 1960 στην επιστήμη των υπολογιστών, καθιερώθηκε το 1990 και αντικατέστησε σε μεγάλο βαθμό το δομημένο προγραμματισμό. Είναι ένα πρακτικό αποτέλεσμα της προσπάθειας να προσομοιωθούν πολύπλοκες δραστηριότητες του ανθρώπινου μυαλού, αντιμετωπίζοντας τα δεδομένα ως αντικείμενα όπως συμβαίνει δηλαδή και στον πραγματικό κόσμο.

Παραδοσιακά, ένα πρόγραμμα αντιμετωπίζεται ως μια λογική διαδικασία που λαμβάνει δεδομένα εισόδου, τα επεξεργάζεται και παράγει δεδομένα εξόδου. Η πρόκληση του προγραμματισμού ήταν το πώς θα αποτυπωθεί η λογική και όχι το πώς θα οριστούν τα δεδομένα. Αντίθετα ο αντικειμενοστραφής προγραμματισμός εστιάζει στα αντικείμενα που χρειάζεται να διαχειριστεί και όχι στη λογική της διαχείρισής τους.

Στην αντικειμενοστραφή τεχνολογία η πολυπλοκότητα των δεδομένων καμουφλάρεται μέσα στα αντικείμενα και τα δεδομένα προσπελάζονται από μια απλή διεπαφή. Επειδή λοιπόν τα αντικείμενα μπορούν να μοντελοποιήσουν πολύπλοκα δεδομένα με απλό τρόπο, είναι κατάλληλα για πολύπλοκες εφαρμογές.

Για τον ίδιο ακριβώς λόγο η αντικειμενοστραφής πρόσβαση στη βάση δεδομένων είναι η καλύτερη επιλογή για εισαγωγή δεδομένων και ενημέρωση της βάσης.

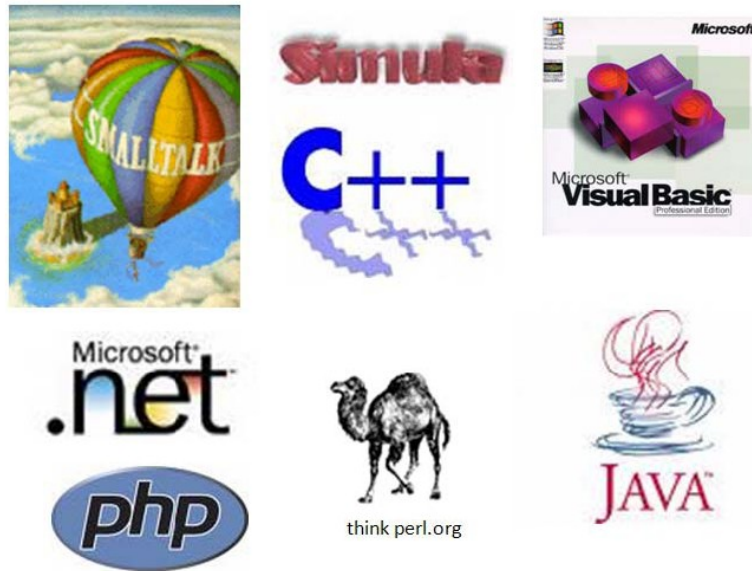
Στο προγραμματιστικό αυτό υπόδειγμα, ο χειρισμός των σχετιζόμενων δεδομένων και των διαδικασιών που επενεργούν σε αυτά γίνεται από κοινού, μέσω μιας δομής δεδομένων που τα περιβάλλει ως αυτόνομη οντότητα με ταυτότητα και δικά της χαρακτηριστικά. Αυτή η δομή λέγεται *κλάση* και έχει δύο συνιστώσες, τα *πεδία* και τις *μεθόδους*. Τα πεδία περιγράφουν τι είναι η κλάση και οι μέθοδοι περιγράφουν τι κάνει η κλάση. Με άλλα λόγια τα πεδία είναι οι μεταβλητές που κάνει χρήση η κλάση αυτή. Οι μέθοδοι είναι οι λειτουργίες της κλάσης, αν φανταστούμε ότι μια κλάση περιγράφει έναν άνθρωπο τότε μια μέθοδος θα ήταν το περπάτημα, τρέξιμο, άλμα κ.α. , ενώ ένα πεδίο του θα ήταν το φύλο, χρώμα δέρματος, η καταγωγή του κ.α. . Αντικείμενα ονομάζονται όλα τα στιγμιότυπα μιας κλάσης, έτσι αντικείμενα της κλάσης άνθρωπος θα ήταν ένας λευκός (πεδίο) άντρας (πεδίο) από την Ελλάδα (πεδίο) με μεθόδους, τραγουδάω (μέθοδος) χορεύω συρτάκι (μέθοδος)

Επομένως, κλάση είναι μια φόρμα ή ένα καλούπι που μέσω αυτού φτιάχνουμε τα αντίτυπα, εδώ αντικείμενα-στιγμιότυπα. Η χρήση των αντικειμένων του Αντικειμενοστραφή Προγραμματισμού συμβάλει σε πιο σωστό προγραμματισμό με το να παρέχει κώδικα σύντομο και πιο αποδοτικό αποφεύγοντας την επαναχρησιμοποίηση κώδικα ιδιαίτερα σε μεγάλα και πολύπλοκα προγράμματα.

Η πρώτη γλώσσα προγραμματισμού που περιείχε αντικειμενοστραφής έννοιες ήταν η γλώσσα προγραμματισμού Simula 67, από την οποία επηρεάστηκε τη δεκαετία του '70 η ανάπτυξη της γλώσσας προγραμματισμού Smalltalk που εισήγαγε τον όρο αντικειμενοστραφής προγραμματισμός. Η Smalltalk αναπτύχθηκε από τον Άλαν Κέι της εταιρείας Xerox στο πλαίσιο μίας εργασίας με στόχο τη δημιουργία ενός χρήσιμου, αλλά και εύχρηστου, προσωπικού υπολογιστή. Το 1980 η τελική έκδοση της Smalltalk έγινε διαθέσιμη σηματοδοτώντας ότι η έρευνα για την αντικατάσταση του δομημένου προγραμματισμού με ένα πιο σύγχρονο υπόδειγμα απέδιδε αποτελέσματα. Στη γλώσσα αυτή καταργήθηκαν οι παραδοσιακές δομές δεδομένων από τις κλάσεις και έτσι όλα ήταν αντικείμενα.

Περίπου την ίδια εποχή και επίσης με επιρροές από την Simula, ολοκληρωνόταν η ανάπτυξη της C++ που ήταν η επέκταση της γλώσσας προγραμματισμού C με αντικειμενοστραφή χαρακτηριστικά. Με την επίδραση της C++ κατά την δεκαετία του '80 έβγαλαν πολλές γλώσσες της εποχής εκείνης αντικειμενοστραφής εκδόσεις. Το προγραμματιστικό υπόδειγμα του αντικειμενοστραφή προγραμματισμού επικράτησε το πρώτο ήμισυ της δεκαετίας του 1990 επειδή ήταν καταλληλότερο στην ανάπτυξη της τότε καθιέρωσης των γραφικών διασυνδέσεων χρήστη στους μικροϋπολογιστές. Το 1995 εμφανίζεται η Java ως μια πλήρως αντικειμενοστραφή γλώσσα που έμοιαζε στο συντακτικό της με την C++ , προσέφερε πρωτοποριακές δυνατότητες για την εποχή και έδωσε νέα ώθηση στον αντικειμενοστραφή προγραμματισμό. Ταυτόχρονα εμφανίστηκαν ποικίλες άτυπες βελτιώσεις στο βασικό προγραμματιστικό υπόδειγμα, όπως οι αντικειμενοστραφείς γλώσσες μοντελοποίησης

λογισμικού και τα σχεδιαστικά πρότυπα. Το 2001 η Microsoft εστίασε την προσοχή της στην πλατφόρμα .NET, μία ανταγωνιστική της Java πλατφόρμα ανάπτυξης και εκτέλεσης λογισμικού η οποία ήταν εξολοκλήρου προσανατολισμένη στην αντικειμενοστρέφεια. Παρακάτω βλέπουμε μερικές από τις πιο γνωστές αντικειμενοστραφείς γλώσσες που αναφέραμε.



Ε.29 : Μερικές γλώσσες βασισμένες σε αντικειμενοστραφής υπόδειγμα.

Αυτή τη στιγμή η C++ και η Java είναι οι πιο δημοφιλείς αντικειμενοστραφείς γλώσσες. Η Java συγκεκριμένα έχει σχεδιαστεί ειδικά για χρήση σε καταμεμημένες εφαρμογές μέσω δικτύου και διαδικτύου.

Το **Object Model** που χρησιμοποιείται στη C# είναι παρόμοιο με άλλες αντικειμενοστραφείς γλώσσες (Object Oriented Languages). Η αντικειμενοστρέφεια είναι νεώτερη από το Δομημένο Προγραμματισμό, αλλά είναι πλέον πολύ σημαντική για τους προγραμματιστές σήμερα. Ο Object Oriented προγραμματισμός παρέχει μια τυποποιημένη γλώσσα και ένα πλαίσιο που είναι σημαντικό για να κάνει την εργασία τμημάτων λογισμικού (objects) ενωμένα μαζί σε ένα σύστημα. Οι Object Oriented έννοιες που μαθαίνονται για τη Visual Basic ισχύουν για τις περισσότερες γλώσσες που ένας προγραμματιστής είναι πιθανό να αντιμετωπίσει σήμερα.

Μια γλώσσα βασισμένη σε αντικείμενα έχει τις περισσότερες από τις ιδιότητες μιας αντικειμενοστραφούς γλώσσας, αλλά μπορεί να στερείται μερικών. Παραδείγματος χάριν η Visual Basic δεν έχει κληρονομικότητα, ενώ μια γλώσσα βασισμένη σε πρωτότυπα στηρίζεται σε αυτά αντί στις κλάσεις για να δημιουργήσει τα αντικείμενα.

Η αντικειμενοστρέφεια (Object Orientation) έχει αλλάξει τον τρόπο που οι βιομηχανίες λογισμικού προσεγγίζουν την ανάπτυξη λογισμικού,

και ιδιαίτερα, πώς οι τεχνικοί προσεγγίζουν την ανάλυση, το σχεδιασμό, την επαναχρησιμοποίηση κώδικα, και την υλοποίηση.

Ο Object Oriented προγραμματισμός είναι μια προσέγγιση (όχι ένα συγκεκριμένο εργαλείο) που οργανώνεται γύρω από τα αντικείμενα παρά από τις εφαρμογές, τα δεδομένα ή τη λογική.

3.1.1.1 Objects

Ως αντικειμενοστραφής γλώσσα η C# , οργανώνεται γύρω από τα αντικείμενα. Τα αντικείμενα είναι προγραμματισμένοι κατασκευαστές οι οποίοι έχουν δεδομένα, ταυτότητα και συμπεριφορά. Τα δεδομένα των αντικειμένων περιλαμβάνονται στα πεδία τους, στις ιδιότητες και στα γεγονότα (events), ενώ οι συμπεριφορές των αντικειμένων ορίζονται από τις μεθόδους και το interface τους. Τα αντικείμενα έχουν ταυτότητα δηλαδή δύο αντικείμενα με το ίδιο set δεδομένων δεν είναι απαραίτητα το ίδιο αντικείμενο.

Τα αντικείμενα έχουν τις εξής ιδιότητες:

- Οτιδήποτε και αν χρησιμοποιείται στην C# είναι αντικείμενο, ακόμα και οι φόρμες των Windows και τα controls.
- Τα αντικείμενα δημιουργούνται από τα templates που ορίζονται από κλάσεις και structs.
- Τα αντικείμενα χρησιμοποιούν τις Ιδιότητές τους για να λειτουργήσουν και να αλλάξουν τις πληροφορίες τις οποίες εμπεριέχουν.
- Συχνά τα αντικείμενα έχουν μεθόδους και γεγονότα τα οποία τους επιτρέπουν να κάνουν δραστηριότητες (actions).
- Όλα τα αντικείμενα στην C# ιεραρχοποιούνται από το Αντικείμενο.

Βλέποντας την ανάλυση και το σχεδιασμό του λογισμικού ως αντικείμενα είναι πολύ σημαντικό επειδή και οι άνθρωποι σκέφτονται σε σχέση με τα αντικείμενα. Όλα γύρω μας είναι αντικείμενα.

3.1.1.2 Classes (Κλάσεις)

Μια κλάση είναι ένα σύνολο παρόμοιων ενεργειών και ιδιοτήτων, και ένα αντικείμενο είναι ένα instance (μια περίπτωση) μιας κλάσης. Μπορούμε να σκεφτούμε μια κλάση ως σχεδιασμό ενός αντικειμένου που θα δημιουργηθεί στον κώδικά μας. Και οι κλάσεις και τα αντικείμενα έχουν ταυτότητα. Μια κλάση καθορίζει τις ιδιότητες και τις μεθόδους, ενώ ένα αντικείμενο τις χρησιμοποιεί.

Μια κλάση καθορίζει τα στοιχεία των δεδομένων και τη μορφή των δεδομένων που συνδυάζονται για να απεικονίσουν το τι ένα αντικείμενο «γνωρίζει». Παραδείγματος χάριν, μπορούμε να δημιουργήσουμε ένα αντικείμενο αποκαλούμενο 'Oscar' από μια κλάση που ονομάζεται Dog. Η κλάση Dog καθορίζει τι πρόκειται να είναι το αντικείμενο 'Oscar', και όλα τα «dog-related» μηνύματα που ένα αντικείμενο Oscar μπορεί να ενεργήσει επάνω τους. Όλες οι αντικειμενοστραφείς γλώσσες έχουν κάποιες έννοιες. Καλούν μερικές φορές ένα «εργοστάσιο», να κατασκευάσει τα instances ενός

αντικειμένου από τον καθορισμό μιας κλάσης. Μπορούμε να κάνουμε περισσότερα από ένα αντικείμενα αυτής της κλάσης, και να τα καλέσουμε Spot, Fido, Rover, κ.λπ. Η κλάση Dog καθορίζει τα μηνύματα που τα αντικείμενα Dog καταλαβαίνουν, όπως «bark», «fetch», και «roll-over».

Οι κλάσεις και τα αντικείμενα έχουν χαρακτηριστικά. Παραδείγματα χαρακτηριστικών τους για ένα άτομο είναι η ηλικία, το ύψος, το βάρος, και δείκτης νοημοσύνης. Αυτά τα χαρακτηριστικά αναφέρονται ως χαρακτηριστικά κλάσης και ιδιότητες αντικειμένου. Τα χαρακτηριστικά της κλάσης και οι ιδιότητες του αντικειμένου είναι επίσης γνωστές ως member fields. Επειδή ένα αντικείμενο έχει τη δυνατότητα να θέσει τις ιδιότητες, τα αντικείμενα λέγονται επίσης ότι έχουν την κυριότητα. Οι τρέχουσες τιμές των χαρακτηριστικών ενός αντικειμένου είναι η επικρατούσα κατάσταση τους.

3.1.1.3 Properties (Ιδιότητες)

Τα **Properties** (ιδιότητες) λένε για το αντικείμενο, πληροφορίες όπως είναι το όνομά του, η θέση του στον υπολογιστή, εάν είναι ορατό, ενεργό, ή το χρώμα του. Οι ιδιότητες είναι όπως τα επίθετα που περιγράφουν τα αντικείμενα (ουσιαστικά). Οι ιδιότητες στην .NET δηλώνονται με τη χρησιμοποίηση του ονόματος αντικειμένου, μιας περιόδου (.), και την επιθυμητή ιδιότητα. Για παράδειγμα αν θέλουμε να δηλώσουμε ότι ένα αντικείμενο είναι ορατό δηλαδή Visible, τότε θα χρησιμοποιήσουμε `public namespace Visible { get; set; }` και οι τιμές που παίρνει είναι true ή false. Η σημαντικότερη ιδιότητα κάθε αντικειμένου είναι γενικά το όνομά της. Το όνομα είναι αυτό που δίνει το αντικείμενο σε ένα script και το πώς ένα αντικείμενο δηλώνεται από οποιοδήποτε άλλο αντικείμενο. Η κωδικοποίηση της κλάσης πρέπει να λαμβάνεται υπόψιν κατά την ονομασία των αντικειμένων.

3.1.1.4 Methods (Μέθοδοι)

Οι κλάσεις και τα αντικείμενα έχουν ενέργειες. Αυτές οι ενέργειες αναφέρονται ως λειτουργίες κλάσης και **μέθοδοι** αντικειμένου. Τις αναφέρουμε γενικά ως μεθόδους (**methods**). Μέθοδος είναι το πώς ο κώδικας μπορεί να χρησιμοποιήσει ένα αντικείμενο κάποιας κλάσης. Οι μέθοδοι μπορούν να διαιρεθούν σε queries (ερωτήσεις) επιστρέφοντας την επικρατούσα κατάσταση και σε commands (εντολές) που την αλλάζουν (υπορουτίνα). Μερικές φορές η πρόσβαση στα δεδομένα ενός αντικειμένου είναι περιορισμένη στις μεθόδους της κλάσης του. Μια μέθοδος είναι ουσιαστικά η εφαρμογή μιας υπηρεσίας αντικειμένου η οποία είναι απλά η δράση που ένα μήνυμα μεταφέρει. Είναι ο κώδικας, ο οποίος εκτελείται όταν το μήνυμα στέλνεται σε ένα συγκεκριμένο αντικείμενο. Ορίσματα παρέχονται συχνά ως τμήμα ενός μηνύματος. Στον Object Oriented προγραμματισμό στέλνουμε ένα μήνυμα από ένα αντικείμενο σε ένα άλλο. Οι μέθοδοι και οι ιδιότητες είναι κλήσεις μηνυμάτων. Οι παράμετροι μιας μεθόδου είναι το περιεχόμενο των μηνυμάτων.

3.1.1.5 Events (Γεγονότα)

Τα γεγονότα (**Events**) είναι το πως τα αντικείμενα πρέπει να ανταποκρίνονται. Μπορούν να είναι ερεθίσματα για μεθόδους αντικειμένων και αναφέρονται ως `Object.Events`. Για παράδειγμα όταν σε μία εφαρμογή ο χρήστης θα πρέπει να πατήσει ένα κουμπί τότε ενεργοποιείτε το γεγονός (event) **`OnClick()`**; Άλλα γεγονότα είναι παραδείγματος χάριν:

event EventHandler DoubleClick

event EventHandler MouseEnter

event MouseEventHandler MouseMove

και φυσικά πολλά πολλά άλλα.

3.1.1.6 Inheritance (Κληρονομικότητα)

Η αρχή μιας κλάσης καθιστά πιθανό να καθοριστούν οι υποκλάσεις που μοιράζουν μερικά ή όλα τα κύρια χαρακτηριστικά της κλάσης. Αυτό καλείται κληρονομικότητα (**inheritance**). Η κληρονομικότητα μας επιτρέπει επίσης να επαναχρησιμοποιήσουμε τον κώδικα αποτελεσματικότερα. Ένας μηχανισμός για τις υποκλάσεις και παρέχει έναν τρόπο να καθοριστεί μια υποκλάση ως εξειδίκευση ή υποκατηγορία ή επέκταση μιας γενικότερης κλάσης. Μια υποκλάση κληρονομεί όλα τα μέλη (members) των superclass (υπερκλάσεων) της, αλλά μπορεί να επεκτείνει τη «συμπεριφορά» τους και να προσθέσει τα νέα μέλη.

3.1.1.7 Encapsulation

Το **encapsulation** (ενθυλάκωση) είναι το κρύψιμο των στοιχείων και του κώδικα και καλείται συχνά ως «black box» προσέγγιση, δεδομένου ότι οι χρήστες μιας κλάσης δεν μπορούν να δουν μέσα στην κλάση (μπορούν να δουν μόνο το public interface της). Εξασφαλίζει επίσης ότι ο κώδικας εξωτερικά από μια κλάση βλέπει μόνο τις λειτουργικές λεπτομέρειες εκείνης της κλάσης, αλλά όχι τις λεπτομέρειες της υλοποίησης. Το Encapsulation επιτυγχάνεται διευκρινίζοντας ποιες κλάσεις μπορούν να χρησιμοποιήσουν τα μέλη ενός αντικειμένου. Το αποτέλεσμα είναι ότι κάθε αντικείμενο εκθέτει σε οποιαδήποτε κλάση ένα συγκεκριμένο interface. Τα μέλη καθορίζονται συχνά ως public, protected και private. Αυτό καθορίζεται από το εάν είναι διαθέσιμα σε όλες τις κλάσεις, τις υποκλάσεις ή μόνο την καθορισμένη κλάση. Μερικές γλώσσες επεκτείνονται περαιτέρω: Η Java χρησιμοποιεί protected λέξη κλειδί για να περιορίσει την πρόσβαση. Η C# και η VB.NET υποκαθιστούν μερικά μέλη σε κλάσεις χρησιμοποιώντας τις λέξεις κλειδιά ως internal (C#) ή ως Friend (VB.NET). Τα αντικείμενα αλληλεπιδρούν το ένα με το άλλο μέσω μηνυμάτων. Το μόνο πράγμα που ένα αντικείμενο γνωρίζει για ένα άλλο είναι το interface του αντικειμένου. Τα δεδομένα και η λογική κάθε αντικειμένου είναι κρυμμένα από άλλα αντικείμενα. Αυτό επιτρέπει στον αναλυτή να χωρίσει την υλοποίηση ενός αντικειμένου από το interface του. Εφ' όσον το interface

παραμένει ίδιο, οποιεσδήποτε αλλαγές στην εσωτερική υλοποίηση είναι προφανείς στο χρήστη.

3.1.1.8 Polymorphism (Πολυμορφισμός)

Ένα άλλο όφελος που προκύπτει από το διαχωρισμό της υλοποίησης από τη συμπεριφορά είναι ο πολυμορφισμός (**polymorphism**). Ο πολυμορφισμός είναι συμπεριφορά που ποικίλλει ανάλογα με την κλάση στην οποία η συμπεριφορά επιδρά, δηλαδή δύο ή περισσότερες κλάσεις μπορούν να αντιδράσουν διαφορετικά στο ίδιο μήνυμα. Ο πολυμορφισμός επιτρέπει σε δύο ή περισσότερα αντικείμενα για να αποκριθούν στο ίδιο μήνυμα. Με άλλα λόγια, ο πολυμορφισμός επιτρέπει σε οποιαδήποτε απόγονο κλάση για να επαναπροσδιορίσει οποιαδήποτε μέθοδο που κληρονομείται από την κλάση γονέων του. Η επίδραση είναι ότι ο πολυμορφισμός επιτρέπει σε ένα σταλμένο αντικείμενο να επικοινωνήσει με τα διαφορετικά αντικείμενα κατά σύμφωνο τρόπο χωρίς ανησυχία για το πόσες διαφορετικές υλοποιήσεις ενός μηνύματος υπάρχουν.

3.1.2. JAVA

Η Java είναι μια νέα γλώσσα προγραμματισμού, η οποία σε αντίθεση με άλλες γλώσσες γνωστές ήδη από δεκαετίες, εμπεριέχει όλα τα χαρακτηριστικά της εξέλιξης της επιστήμης των υπολογιστών και είναι ιδιαίτερα γνωστή όχι μόνο από τους υπολογιστές αλλά και από τα κινητά και πολλές άλλες συσκευές πολυμέσων καθώς και από το διαδίκτυο.

Πρόκειται για μία αντικειμενοστραφής γλώσσα, που επιτρέπει σε οποιονδήποτε υπολογιστή όπου κι αν βρίσκεται αυτός, να έχει πρόσβαση και να χρησιμοποιεί μια εφαρμογή εγκατεστημένη σε κάποιο δίκτυο. Χρησιμοποιείται τόσο σε εφαρμογές διαδικτύου όσο και σε αυτόνομες εφαρμογές.

Αν τη συγκρίνουμε από πλευράς δομής με τις άλλες γλώσσες προγραμματισμού, η Java μοιάζει περισσότερο με τη C. Της μοιάζει αρκετά στον τρόπο σύνταξης και τη φιλοσοφία αλλά δεν είναι C.

Η δημιουργία της Java ξεκίνησε το 1990 στην εταιρεία Sun Microsystems από μια ομάδα ανθρώπων με επικεφαλή τον James Gosling. Ο σκοπός της ομάδας αυτής ήταν να δημιουργήσουν προγράμματα τα οποία θα έκαναν πιο "επαναστατική" τη χρήση και τη λειτουργία των οικιακών συσκευών (βίντεο, τηλεόραση, στερεοφωνικά κτλ). Στην αρχή χρησιμοποίησαν την C++ λόγω ταχύτητας αλλά ο Gosling, σύντομα διαπίστωσε ότι η C++ δεν ήταν η κατάλληλη γλώσσα προγραμματισμού για τα προγράμματα που ήθελαν να φτιάξουν. Οπότε με τη βοήθεια ενός άλλου προγραμματιστή, του Patrick Naughton τον Αύγουστο του 1991 δημιούργησε μια καινούργια γλώσσα την οποία ονόμασαν OAK (βελανιδιά, από μια βελανιδιά που υπήρχε στον κήπο του κτηρίου που ήταν το γραφείο τους). Κατάφεραν τελικά να φτιάξουν έναν υπολογιστή χειρός που έμοιαζε με τηλεχειριστήριο το οποίο όμως δεν παρουσίασε την αναμενόμενη εμπορική επιτυχία παρόλο που κατασκευαστικά πέτυχε. Το 1993 η ομάδα προσπάθησε να προωθήσει την OAK στον τομέα Video On Demand (VOD) κάνοντας σχετική

πρόταση στην εταιρεία Time Warner, η οποία ενδιαφερόταν για τέτοια έργα την εποχή εκείνη .Επειδή ο τομέας αυτός όμως κρίθηκε εμπορικά μη αξιοποιήσιμος η πρόταση απορρίφθηκε. Την εποχή όπου το internet ξεφεύγει από την κατάσταση απλού κειμένου και αρχίζουν να εμφανίζονται τα γραφικά , ένας από τους ιδύνοντες της Sun εκείνη την περίοδο, διέβλεψε ότι η OAK θα μπορούσε να χρησιμοποιηθεί στον παγκόσμιο ιστό. Έτσι η ομάδα το αντιλαμβάνεται και αλλάζει κατεύθυνση του έργου για τρίτη φορά, η φιλοσοφία της εταιρείας ήταν να διαθέτει τη γλώσσα δωρεάν ώστε να τυποποιηθεί. Κατόπιν μετονομάζεται σε Java τον Ιανουάριο του 1995, και λίγο αργότερα παρουσιάζεται επίσημα από την Sun ως εργαλείο ανάπτυξης εφαρμογών για το Internet. Η διάδοση της αρχίζει από τα προγράμματα πλοήγησης όπως Netscape Navigator (2.0), Internet Explorer(3.0) με υποστήριξη Java.

Όπως αναφέραμε και παραπάνω η Java από πλευράς δομής μοιάζει περισσότερο με την C/C++. Μοιάζει αρκετά στον τρόπο σύνταξης και χρησιμοποιεί πολλά στοιχεία της, δεν είναι όμως C/C++. Αποφεύγει πολλά χαρακτηριστικά της C/C++ για λόγους απλότητας και ασφάλειας, ενώ σε άλλα επεκτείνει τις λειτουργίες τους για να γίνει πιο ευέλικτη και πιο σύγχρονη γλώσσα. Έχει πάρει επίσης και στοιχεία από την γλώσσα προγραμματισμού Smalltalk όπως τον πηγαίο κώδικα (Byte code) και το Garbage Collection(αυτόματος καθαρισμός της μνήμης).

Μια από τις ιδιότητες που κάνουν την Java να ξεχωρίζει από άλλες γλώσσες προγραμματισμού, είναι τα applets (μικροεφαρμογές), ειδικά προγράμματα που το σημαντικό τους χαρακτηριστικό είναι η ασφάλεια που διαθέτουν από ιούς και προγράμματα με κακό σκοπό. Αυτό συμβαίνει διότι τα applets δεν μπορούν να γράψουν στο σκληρό δίσκο αν δεν πάρουν άδεια. Για να τρέξει ένα applet χρειάζεται ένα πρόγραμμα πλοήγησης (browser), όπως ο Internet Explorer, το Netscape Communicator κτλ., που να έχει σύνδεση στο διαδίκτυο ώστε να μπορούμε να το κατεβάσουμε. Όταν κάποιος κατεβάζει αρχεία από το διαδίκτυο, αντιμετωπίζει άμεσα τον κίνδυνο να "εισπράξει" κάποιον ιό ή άλλο πρόγραμμα, το οποίο πιθανόν να προκαλέσει ζημίες στον σκληρό δίσκο του υπολογιστή του. Η Java λύνει αυτό το πρόβλημα περιορίζοντας αυστηρά το τι μπορεί να κάνει ένα applet. Ένα applet λ.χ. δε μπορεί να γράψει στο σκληρό μας δίσκο αν δε μας ζητήσει πρώτα την άδεια, ούτε και να μπλοκάρει τον υπολογιστή μας.

Η Java εκτός από τα applet μπορεί να χρησιμοποιηθεί και για την δημιουργία παντός φύσεως προγραμμάτων τα οποία καλούνται εφαρμογές (applications).

Σήμερα αποτελεί μια ιδιαίτερα δημοφιλή γλώσσα που αξιοποιείται με ποικίλους τρόπους και για διάφορα πεδία εφαρμογών. Αυτό οφείλεται σε ορισμένα ιδιαίτερα χαρακτηριστικά που καθιστούν τη γλώσσα δημοφιλή, όπως τα ακόλουθα:

- **Είναι (σχετικά) απλή.**

Οι δημιουργοί της Java άφησαν έξω από τη γλώσσα "δύσκολα" χαρακτηριστικά που συναντάμε σε άλλες γλώσσες, όπως λ.χ. οι δείκτες (pointers) ή οι δομές και οι ενώσεις (structures and unions) της C.

- **Είναι αντικειμενοστραφής (object oriented).**

Αντικειμενοστρέφεια είναι μία προσέγγιση στην ανάπτυξη λογισμικού που οργανώνει τόσο το πρόβλημα όσο και τη λύση του ως μία συλλογή από διακριτά αντικείμενα. Η Java όπως και η C, χρησιμοποιεί τις κλάσεις για να οργανώσει τον κώδικα σε λογικές ενότητες. Κατά το χρόνο εκτέλεσης, το πρόγραμμα δημιουργεί από τις κλάσεις αντικείμενα. Τα αντικείμενα αυτά έχουν δύο συνιστώσες: τα πεδία και τις μεθόδους. Τα πεδία περιγράφουν τι είναι το αντικείμενο· οι μέθοδοι περιγράφουν τι κάνει το αντικείμενο. Οι κλάσεις μπορούν να κληρονομήσουν ιδιότητες από άλλες κλάσεις. Εκείνο που δεν επιτρέπεται είναι η πολλαπλή κληρονομικότητα (multiple inheritance), όπου μια κλάση έχει τη δυνατότητα να κληρονομήσει πεδία και μεθόδους από περισσότερες από μια άλλες κλάσεις.

- **Επιτρέπει την επαναχρησιμοποίηση κώδικα.**

Η επαναχρησιμοποίηση κώδικα επιτυγχάνεται με τη χρησιμοποίηση-εισαγωγή ήδη υπαρχόντων προγραμμάτων των λεγόμενων κλάσεων οι οποίες επιτρέπουν στον προγραμματιστή να φτάσει γρήγορα στο επιθυμητό τελικό αποτέλεσμα.

- **Είναι μεταγλωττιζόμενη** αλλά μπορεί να χαρακτηριστεί και ως **διερμηνευόμενη.**

Η μεταγλώττιση του πηγαίου προγράμματος έχει σαν αποτέλεσμα την παραγωγή ενός ειδικού κώδικα, ο οποίος ονομάζεται *κώδικας byte (bytecode)*. Ο κώδικας αυτός, ενώ μοιάζει με τον κώδικα σε γλώσσα μηχανής μπορεί να εκτελεσθεί από οποιοδήποτε λειτουργικό σύστημα (Windows, UNIX, Mac OS κτλ) που διαθέτει διερμηνευτή της C. Εδώ έχουμε τη λύση σχετικά με την τοποθέτηση προγραμμάτων σε σελίδες του παγκόσμιου ιστού του Internet. Τα προγράμματα των υπολογιστών είναι κατά κύριο λόγο προσαρμοσμένα στο *υλικό (hardware)* του υπολογιστή και στο λειτουργικό του σύστημα. Ένα πρόγραμμα γραμμένο για τα Windows δε μπορεί να τρέξει στον Macintosh, ούτε μια εφαρμογή για Macintosh μπορεί να εκτελεστεί σε ένα σταθμό εργασίας UNIX. Η Java λύνει αυτό το πρόβλημα δημιουργώντας εφαρμογές με *ανεξαρτησία πλατφόρμας (platform independence)*. Αυτό γίνεται διότι ο μεταγλωττιστής της Java δε δημιουργεί *γηγενή κώδικα (native code)* προσαρμοσμένο στο συγκεκριμένο τύπο υπολογιστή(όπως κάνουν άλλες γλώσσες), αλλά δημιουργεί κώδικα byte.

- **Είναι ασφαλής.**

Η Java έχει σχεδιαστεί από την αρχή με τέτοιο τρόπο, ώστε να παρέχει ασφάλεια εκτέλεσης του κώδικα σε δίκτυο. Αυτό είχε ως αποτέλεσμα τον περιορισμό αρκετών από τα χαρακτηριστικά που έχουν η C και η C++. Έτσι, δεν υπάρχουν δείκτες (pointers), ούτε μπορεί να γίνει αυθαίρετη προσπέλαση διευθύνσεων της μνήμης. Τα μέτρα αυτά παρέχουν προστασία από τη δράση των ιών. Η Java χρησιμοποιεί επίσης έναν ισχυρό μηχανισμό για τον έλεγχο αναμενόμενων και μη αναμενόμενων σφαλμάτων (handling).

- **Υποστηρίζει πολυνημάτωση (multithreading).**

Ένα πρόγραμμα Java μπορεί να περιλαμβάνει πολλές ξεχωριστές διαδικασίες, οι οποίες να εκτελούνται συνεχώς και ανεξάρτητα η μία από την άλλη. Μπορεί λόγω χάρη από το πρόγραμμα να μεταδίδεται μια εικόνα, και συγχρόνως ο χρήστης να εισαγάγει στοιχεία από το πληκτρολόγιο.

- Κάνει **συλλογή ακρήστων (garbage collection)**.

Τα προγράμματα της Java κάνουν από μόνα τους συλλογή ακρήστων. Αυτό σημαίνει ότι ο προγραμματιστής δε χρειάζεται να φροντίζει πλέον ο ίδιος για τη διαγραφή ακρήστων δεδομένων από τη μνήμη.

- Μπορεί να δημιουργήσει **γρήγορο κώδικα**.

Αυτό οφείλεται στην χρήση γρήγορων μεταγλωττιστών των λεγόμενων "just in time compilers" (JIT).

- Παρέχει βιβλιοθήκες κώδικα για διάφορες χρήσεις, όπως δημιουργία γραφικών, χειρισμό αλφαριθμητικών, μαθηματικές πράξεις, προσπέλαση αρχείων στο Διαδύκτιο, χειρισμό σχεσιακών βάσεων δεδομένων, δημιουργία εφαρμογών πελάτη-εξυπηρετητή, κλήση απομακρυσμένων αντικειμένων κ.τ.λ
- Με την Java μπορούμε να δημιουργήσουμε **δυναμικές ιστοσελίδες** (dynamic web pages), οι οποίες χρησιμοποιώντας το πρωτόκολλο μεταφοράς HTTP και την γλώσσα υπερκειμένου HTML προσφέρουν στους χρήστες του διαδικτύου πλείστες δυνατότητες. Μεταξύ άλλων, σε μια ιστοσελίδα μπορούμε:
 - Να "παίξουμε" ήχο ή βίντεο.
 - Να χρησιμοποιήσουμε διανυσματικά γραφικά αντί ψηφιογραφικών (bitmap) εικόνων.
 - Να δημιουργήσουμε κίνηση σε πραγματικό χρόνο.
 - Να δημιουργήσουμε φόρμες εισαγωγής στοιχείων από τους χρήστες.
 - Να φτιάξουμε παιχνίδια και εφαρμογές πραγματικού χρόνου, στα οποία να συμμετέχουν ταυτόχρονα πολλοί παίκτες.

Η εταιρία Sun προσφέρει την Java σε τρεις μορφές, κάθε μια από τις οποίες στοχεύει σε διαφορετικό τομέα τεχνολογίας. Οι μορφές αυτές είναι:

- **J2SE (Java 2 platform Standard Edition)**. Είναι το στάνταρτ πακέτο της Java το οποίο χρησιμοποιείται τόσο για εφαρμογές του Παγκόσμιου Ιστού όσο και για κανονικές προγραμματιστικές εφαρμογές.
- **J2EE (Java 2 platform Enterprise Edition)**. Το πακέτο περιλαμβάνει μια σειρά από τεχνολογίες κάτω από ενιαία αρχιτεκτονική και στοχεύει στη δημιουργία επιχειρηματικών εφαρμογών ηλεκτρονικού εμπορίου βασισμένων στον Παγκόσμιο Ιστό.
- **J2ME (Java 2 platform Micro Edition)**. Το πακέτο αποτελεί μια "ελαφρή" έκδοση για αποκλειστική χρήση σε εμπορικές συσκευές και εργαλεία, όπως έξυπνες πιστωτικές κάρτες, κινητά τηλέφωνα, εικονοτηλέφωνα, συστήματα πλοήγησης αυτοκινήτων, χειριστήρια ασυρμάτου ελέγχου συσκευών κ.τ.λ.

3.1.2.1 Μετατροπή των δεδομένων μιας Βάσης Δεδομένων σε αντικείμενα JAVA

Η εφαρμογή τα δεδομένα που παίρνει από την ΒΔ μπορεί και τα μετατρέπει σε αντικείμενα της Java. Όταν λέμε δεδομένα εννοούμε τις εγγραφές μιας Βάσης Δεδομένων. Κάθε τέτοια έγγραφη διαθέτει διάφορα χαρακτηριστικά (πίνακα στον οποίο ανήκει, πεδίο, τύπος δεδομένου κάθε πεδίου κ.α.). Πολλά από τα χαρακτηριστικά και συγκεκριμένα αυτά που χρειαζόμαστε για τις ανάγκες της εργασίας μας τα μεταβιβάζουμε και στα δικά μας αντικείμενα που δημιουργούμε. Τα χαρακτηριστικά αυτά υπάρχουν και μπορούν να προσπελαστούν μέσω των μεταβλητών ή των μεθόδων που διαθέτει το κάθε αντικείμενο.

3.1.3 NETBEANS

Το NetBeans IDE είναι ένα περιβαλλοντικό ανάπτυγμα IDE - ένα εργαλείο στη διάθεση των προγραμματιστών για να γράψουν, να κάνουν compile, debug και να αναπτύξουν προγράμματα. Χρησιμοποιείται κυρίως για σχεδίαση Java εφαρμογών και είναι γραμμένο σε Java, αλλά μπορεί να υποστηρίξει ένα μεγάλο εύρος γλωσσών προγραμματισμού, όπως Java, C/C++ και άλλες. Υπάρχει επίσης ένας μεγάλος αριθμός υπομονάδων (modules) που βοηθάνε στην επέκταση της λειτουργικότητας του NetBeans IDE.

Πλεονεκτήματα

- GNU license
- Υποστηρίζει πολλές γλώσσες προγραμματισμού
- Μεταφερσιμότητα λόγω του ότι είναι γραμμένο σε Java
- Debugger
- Ευκολία συγγραφής κώδικα
- Ευκολία δημιουργίας GUI
- Code completion

Μειονεκτήματα

- Αργός , επειδή τρέχει σε εικονική μηχανή

Παράλληλα, η έκδοση 6.5 έχει σημαντικές καινοτομίες σε σχέση με την προηγούμενη και η πιο σημαντική βρίσκεται στο γραφικό περιβάλλον το οποίο είναι πλέον πολύ εύκολο στην χρήση. Ταυτόχρονα, βοηθάει με την αυτόματη δημιουργία κώδικα έτσι ώστε να γλιτώνουμε πολύ χρόνο στην υλοποίηση της εφαρμογής. Επίσης, μέσω του γραφικού περιβάλλοντος μπορούμε και εντοπίζουμε τα λάθη και τις ανωμαλίες στον κώδικα που γράφουμε πολύ γρήγορα. Στην εφαρμογή έχει χρησιμοποιηθεί η έκδοση 6.5.

3.1.4 SQL (Δομημένη Γλώσσα Ερωτημάτων)

Η γλώσσα SQL είναι ένα εργαλείο για την οργάνωση, διαχείριση και ανάκτηση δεδομένων που είναι αποθηκευμένα σε έναν υπολογιστή. Γεννήθηκε μαζί με την τεχνολογία των σχεσιακών βάσεων δεδομένων και έγινε στερεότυπη για την διαδραστικότητα αυτών. Χρησιμοποιείται κατά κόρων σε σύγχρονες εφαρμογές για να θέτει ερωτήματα σε τέτοιες βάσεις. Επιτρέπει σε άλλες συμβατές με αυτή γλώσσες να εκτελούν ερωτήματα σε ένα μεγάλο εύρος δεδομένων και λειτουργεί αντιμετωπίζοντας τα δεδομένα με μια απλή και τυποποιημένη διάταξη δυοδιάστατων πινάκων δεδομένων, με γραμμές και στήλες.

Το απλό αυτό μοντέλο όμως, που επιτρέπει τη σύνταξη κομψών ερωτημάτων σε μια βάση δεδομένων έχει και το τμήμα του. Η πολυπλοκότητα των σχέσεων μεταξύ των δεδομένων του πραγματικού κόσμου δεν μπορεί να αποτυπωθεί εύκολα σε γραμμές και στήλες, αναγκάζοντας έτσι τα δεδομένα να κατακερματιστούν σε πολλαπλούς πίνακες, για να εκτελεστούν έστω και απλές εργασίες. Το γεγονός αυτό γεννάει δυο προβλήματα α) τα ερωτήματα μπορεί να γίνουν ιδιαίτερα πολύπλοκα προκειμένου να συγκεντρώσουν δεδομένα από πολλούς πίνακες και β) το κόστος επεξεργασίας για την πρόσβαση των δεδομένων με τέτοια ερωτήματα σε σχεσιακές βάσεις, μπορεί να αποβεί κολοσσιαίο.

Παρόλο που η SQL προέκυψε από τις σχεσιακές βάσεις δεδομένων δεν χρειάζεται να περιοριστεί σε αυτές. Η βάση δεδομένων Caché που έχουμε χρησιμοποιήσει υποστηρίζει την στερεότυπη SQL χρησιμοποιώντας όμως μια δυναμικότερη πολυδιάστατη τεχνολογία βάσης δεδομένων η οποία έχει επεκταθεί ώστε να μπορεί να περιλάβει αντικειμενοστραφή χαρακτηριστικά.

3.1.5 Η βάση δεδομένων MySql

MySQL είναι ένα πολύ γρήγορο και δυνατό περιβάλλον διαχείρισης (manager) σχεσιακών βάσεων δεδομένων. Μια βάση δεδομένων μας επιτρέπει να προσθέτουμε, να αναζητάμε, να ανακτάμε και να διαχειριζόμαστε πληροφορίες που είναι αποθηκευμένες εκεί. Η σχεσιακή MySQL σημαίνει ότι μια πληροφορία αποθηκεύεται σε χωριστούς πίνακες και όχι σε έναν μεγάλο πίνακα. Μπορούν να καθιερωθούν σχέσεις μεταξύ πινάκων και να ανακτείτε πληροφορίες χρησιμοποιώντας δομημένη γλώσσα διατύπωσης ερωτήσεων (SQL).

Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα σας , για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα , για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια

Η βάση δεδομένων MySQL έχει γίνει η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα διακομιστή, λόγω της αυξημένης απόδοσης, υψηλής αξιοπιστίας και ευκολίας στην διαχείριση της. Καθίσταται η απόλυτη επιλογή μιας νέας γενιάς εφαρμογών που υλοποιούνται σε

περιβάλλον LAMP (Linux, Apache, MySQL, PHP / Perl / Python). Η MySQL λειτουργεί σε περισσότερες από 20 πλατφόρμες συμπεριλαμβανομένων των Linux, Windows, OS/X, HP-UX, AIX, Netware και χρησιμοποιείται σε περισσότερες από 6 εκατομμύρια εγκαταστάσεις κλιμακούμενες από μεγάλες εταιρίες, όπως η NASA, το Yahoo, το Associated Press (AP), Suzuki, και Sabre Holdings, μέχρι εξειδικευμένες εφαρμογές με ενσωματωμένες βάσεις δεδομένων σε όλο τον κόσμο, ακόμα και στην Ανταρκτική.

Η QUALISYS μεταπωλεί την MySQL και παρέχει υπηρεσίες εγκατάστασης και υποστήριξης της βάσης δεδομένων.

Δέκα λόγοι για να επιλέξουμε την βάση δεδομένων MySQL!

- Scalability and Flexibility (Ευελιξία Και Επεκτασιμότητα)
- Αξιοπιστία και απόδοση
- High Availability (Υψηλή Διαθεσιμότητα)
- Robust Transactional Support
- Web and Data Warehouse Strengths (Αποθηκευτική Δύναμη)
- Strong Data Protection (Υψηλή Προστασία Δεδομένων)
- Comprehensive Application Development
- Ευκολία χρήσης και εγκατάστασης. Η MySQL αρχιτεκτονική καθιστά εξαιρετικά γρήγορα και απλά να προσαρμόσετε. Η μοναδική πολλαπλής αποθήκευσης κινητήρα αρχιτεκτονική της MySQL δίνει εταιρικούς πελάτες την ευελιξία που χρειάζονται με ένα DBMS προηγούμενο σε σταθερότητα, ταχύτητα, συμπαγές, και η ευκολία εγκατάστασης.
- Ελευθερία από πλατφόρμα Lock-in. Με την παροχή άμεσης πρόσβασης στον πηγαίο κώδικα, MySQL AB προσέγγιση διασφαλίζει την ελεύθερη, εμποδίζοντας έτσι το κλείδωμα σε ένα και μόνο προμηθευτή ή πλατφόρμα
- Cross-πλατφόρμα υποστήριξης. MySQL είναι διαθέσιμα σε περισσότερες από είκοσι διαφορετικές πλατφόρμες, συμπεριλαμβανομένων όλων των μεγάλων διανομές Linux, Unix, Microsoft Windows και Mac OS X.
- Εκατομμύρια των εκπαιδευμένων και πιστοποιημένων Developers (Open Source Freedom and 24 x 7 Support). Η MySQL είναι η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα. Αυτό διευκολύνει την εύρεση και έμπειρο knowledgable DBA και προγραμματιστές.
- Lowest Total Cost of Ownership (Χαμηλό Κόστος)
- Μεταφερσιμότητα
- Κώδικας Προέλευσης

3.2 Αναπτυχθέντα Τμήματα

Εφόσον η υπηρεσία θα έχει ως στόχο την αποτελεσματική λειτουργία μιας εφαρμογής που θα διαχειρίζεται τα ταξιδιωτικά μας πακέτα με όλες τις πλήρης πληροφορίες του καθενός, τα περιεχόμενα θα είναι τέτοια ώστε να εξυπηρετούν αυτό τον σκοπό. Τα βασικά στοιχεία μιας τέτοιας βάσης αναφέρονται παρακάτω:

Έτσι θα περιλαμβάνει:

- ✓ Το όνομα του πακέτου
- ✓ τη γενική περιγραφή του
- ✓ την ημερομηνία έναρξης του ταξιδιού
- ✓ την ημερομηνία λήξης του ταξιδιού
- ✓ την συνολική του διάρκεια
- ✓ την περιγραφή της κάθε μέρας του πακέτου
- ✓ αν είναι διαθέσιμο ή όχι
- ✓ καθώς και την ικόνα του πακέτου

-- ΚΕΦΑΛΑΙΟ 4 --

Σενάρια χρήσης

4.1 Δυνατότητα Αναζήτησης Πακέτου στην Βάση Δεδομένων

1. Εισαγωγή στο Κεντρικό Παράθυρο της εφαρμογής
2. Επιλογή "Edit"
3. Επιλογή "Search"
4. Επιλογή Δραστηριοτήτων
5. Επιλογή Διάρκειας (σε ημέρες)
6. Επιλογή Περιόδου (από...έως..)
7. Επιλογή "Search"
8. Προβολή ευρεθέντων πακέτων

⇒ Για την εκτέλεση του βήματος 8 πρέπει να έχει βρεθεί έστω και ένα πακέτο που να πληρεί τα κριτήρια που θέσαμε, αλλιώς δεν θα προβληθεί κανένα πακέτο.

4.2 Δυνατότητα Προσθήκης Πακέτου στην Βάση Δεδομένων

1. Εισαγωγή στο Κεντρικό Παράθυρο της εφαρμογής
2. Επιλογή "Edit"
3. Επιλογή "Add"
4. Δίνω το όνομα του πακέτου
5. Δίνω την περιγραφή του
6. Δίνω το εμπορικό όνομα του πακέτου
7. Δίνω το την εμπορική περιγραφή του
8. Δίνω τους όρους του
9. Επιλέγω το ημερολόγιο για να διαλέξω την ημερομηνία έναρξης του πακέτου
10. Επιλέγω την ημερομηνία έναρξης
11. Επιλέγω το ημερολόγιο για να διαλέξω την ημερομηνία λήξης του πακέτου.

12. Επιλέγω την ημερομηνία λήξης.
13. Τσεκάρω αν είναι διαθέσιμο ή όχι.
14. Επιλέγω "..." για να κάνω Browse την εικόνα του πακέτου
15. Στο Dialog που ανοίγει επιλέγω την εικόνα μου
16. Επιλέγω "OK".
17. Δίνω τον τίτλο της ημέρας του πακέτου.
18. Δίνω την περιγραφή της ημέρας
19. Επιλέγω "Save".

- ⇒ Για την εκτέλεση του βήματος 17 και 18 πρέπει να έχω επιλέξει της ημερομηνίες έναρξης και λήξης του πακέτου, ώστε να υπολογιστεί η διάρκεια του και να δημιουργηθούν αυτόματα οι καρτέλες για την περιγραφή της κάθε μέρας.
- ⇒ Επίσης τα βήματα 17 και 18, επαναλαμβάνονται τόσες φορές, όσες είναι και οι καρτέλες που έχουν δημιουργηθεί, δηλαδή όσες και η διάρκεια του πακέτου μου σε ημέρες (μια καρτέλα ανα ημέρα).

4.3 Δυνατότητα Προβολής των Σχετικών με την Εφαρμογή

1. Εισαγωγή στο Κεντρικό Παράθυρο της εφαρμογής.
2. Επιλογή "Help"
3. Επιλογή "About"

4.3 Δυνατότητα Εξόδου από την Εφαρμογή

1. Εισαγωγή στο Κεντρικό Παράθυρο της εφαρμογής.
2. Επιλογή "File"
3. Επιλογή "Exit"

-- ΚΕΦΑΛΑΙΟ 5 --

Συμπεράσματα

Μετά την σχεδίαση, ανάπτυξη και υλοποίηση της εφαρμογής μας, διαπιστώνουμε πόσο πολύτιμη ήταν η χρήση κάποιων τεχνολογιών, όπως η χρήση αντικειμενοστραφούς προγραμματισμού και Βάσεων Δεδομένων στην εργασία μας αυτή.

Συγκεκριμένα, με την βοήθεια μιας βάσης δεδομένων μπορέσαμε να αποθηκεύσουμε και ταξινομήσουμε τις πληροφορίες μας, τις οποίες χωρίς ταξινόμηση θα ήταν αδύνατο να χρησιμοποιήσουμε ή να επεξεργαστούμε. Έτσι θα μπορούμε οποιαδήποτε στιγμή θέλουμε να ασχοληθούμε ξανά με τα δεδομένα μας.

Επίσης διαπιστώνουμε πόσο χρήσιμη αποδिकνύεται η χρήση της αντικειμενοστραφής γλώσσας προγραμματισμού JAVA, καθώς η πολυπλοκότητα των δεδομένων καμουφλάρεται μέσα στα αντικείμενα και τα δεδομένα προσπελάσσονται πλέον από μια απλή διεπαφή πολύ πιο εύχρηστης στο χρήστη. Και στην δική μας περίπτωση τα δεδομένα αυτά είναι οι εγγραφές μας στην Βάσης Δεδομένων.

-- ΚΕΦΑΛΑΙΟ 6 --

ΒΙΒΛΙΟΓΡΑΦΙΑ

<http://nefeli.lib.teicrete.gr/>

<http://artemis.cslab.ntua.gr/Dienst/UI/1.0/Display/artemis.ntua.ece/DT2008-0241>

(Προβλήματα κλασικών βάσεων δεδομένων)

<http://www.astrolavos.tuc.gr/contents/glossaryDSSgr.htm>

(Εργασίες και συστήματα που αναφέρονται σε Database Visualization)

<http://www.ip.gr>

<http://www.qualisys.gr>

<http://el.tech-faq.com/free-database-software.shtml>

(Η βάση δεδομένων MySQL)

-- ΚΕΦΑΛΑΙΟ 7 --

ΠΑΡΑΡΤΗΜΑ

7.1 Κώδικας Εφαρμογής

7.1.1 Η σύνδεση με τη βάση δεδομένων (DBConnection)

Παρακάτω παρουσιάζουμε ένα απλό μέρος κώδικα που κάνει τη σύνδεση με τη βάση δεδομένων και εκτελεί ένα ερώτημα SQL σε ένα πίνακα της βάσης.

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package project1.util;
6
7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.SQLException;
10
11 /**
12  * One Singleton class.
13  * Usage:
14  * DBConnection dbConnection = DBConnection.getDBConnection();
15  * try{
16  *     Connection connection = dbConnection.createConnection();
17  *     Statement st = connection.createStatement();
18  *     ...
19  * @author Harry
20  */
21 public class DBConnection {
22
23     static {
24         try {
25             Class.forName("com.mysql.jdbc.Driver");
26         } catch (ClassNotFoundException ex) {
27             System.err.println("No database driver! Please check
the Classpath");
28             System.exit(1);
29         }
30     }
31 // only one instance from this class held here
32     private static DBConnection instance;
33
34     /**
35     * Get the instance of this class initialized
36     * @return
37     */
38     public static DBConnection getDBConnection() {
39         if (instance == null) {
40             instance = new DBConnection();
41         }
42         return instance;

```

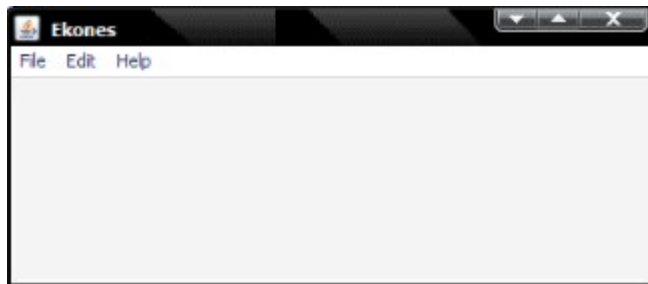
```

43     }
44     private String url;
45     private String user;
46     private String password;
47
48     private DBConnection() {
49         url = "jdbc:mysql://localhost/ekonesdb";
50         user = "harry";
51         password = "anaplas";
52     }
53
54     public Connection createConnection() throws SQLException {
55         return DriverManager.getConnection(url, user, password);
56     }
57 }

```

7.1.2 MainFrame

Εδώ παρουσιάζουμε το μέρος κώδικα όπου υλοποιούμε το κύριο – αρχικό παράθυρο της εφαρμογής μας.



```

1  /*
2  * MainFrame.java
3  *
4  * Created on 14 Ιανουάριος 2008, 6:12 μμ
5  */
6  package project1;
7
8  import java.awt.*;
9  //import javax.swing.ImageIcon;
10 import javax.swing.*;
11
12 /**
13  *
14  * @author Harry
15  */
16 public class MainFrame extends javax.swing.JFrame {
17
18     /** Creates new form MainFrame */
19     public MainFrame() {
20         initComponents();
21         // click on the search menu item:
22
23         pnAdd.add(new AddPanel());
24         pnSearch.add(new SearchPanel());
25
26         miSearch.doClick();
27     }
28
29     /** This method is called from within the constructor to

```



```

30     * initialize the form.
31     * WARNING: Do NOT modify this code. The content of this
method is
32     * always regenerated by the Form Editor.
33     */
34     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
35     private void initComponents () {
36
37         pnSearch = new javax.swing.JPanel ();
38         pnAdd = new AddPanel ();
39         pnContent = new javax.swing.JPanel ();
40         menuBar = new javax.swing.JMenuBar ();
41         mFile = new javax.swing.JMenu ();
42         miExit = new javax.swing.JMenuItem ();
43         mEdit = new javax.swing.JMenu ();
44         miSearch = new javax.swing.JMenuItem ();
45         miAdd = new javax.swing.JMenuItem ();
46         mHelp = new javax.swing.JMenu ();
47         miAbout = new javax.swing.JMenuItem ();
48
49
pnSearch.setBorder (javax.swing.BorderFactory.createTitledBorder ("Sear
ch"));
50         pnSearch.setLayout (new java.awt.BorderLayout ());
51
52
pnAdd.setBorder (javax.swing.BorderFactory.createTitledBorder ("Add"));
53         pnAdd.setLayout (new java.awt.BorderLayout ());
54
55
setDefaultCloseOperation (javax.swing.WindowConstants.DISPOSE_ON_CLOSE
);
56         setTitle ("Ekones");
57
58         pnContent.setLayout (new java.awt.BorderLayout ());
59         getContentPane ().add (pnContent,
java.awt.BorderLayout.CENTER);
60
61         mFile.setText ("File");
62
63
miExit.setAccelerator (javax.swing.KeyStroke.getKeyStroke (java.awt.eve
nt.KeyEvent.VK_Q, java.awt.event.InputEvent.CTRL_MASK));
64         miExit.setText ("Exit");
65         miExit.addActionListener (new
java.awt.event.ActionListener () {
66             public void
actionPerformed (java.awt.event.ActionEvent evt) {
67                 miExitActionPerformed (evt);
68             }
69         });
70         mFile.add (miExit);
71
72         menuBar.add (mFile);
73
74         mEdit.setText ("Edit");
75
76
miSearch.setAccelerator (javax.swing.KeyStroke.getKeyStroke (java.awt.e
vent.KeyEvent.VK_F, java.awt.event.InputEvent.CTRL_MASK));

```

```

77         miSearch.setText("Search");
78         miSearch.addActionListener(new
java.awt.event.ActionListener() {
79             public void
actionPerformed(java.awt.event.ActionEvent evt) {
80                 miSearchActionPerformed(evt);
81             }
82         });
83         mEdit.add(miSearch);
84
85
miAdd.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.
t.KeyEvent.VK_ADD, java.awt.event.InputEvent.CTRL_MASK));
86         miAdd.setText("Add");
87         miAdd.addActionListener(new
java.awt.event.ActionListener() {
88             public void
actionPerformed(java.awt.event.ActionEvent evt) {
89                 miAddActionPerformed(evt);
90             }
91         });
92         mEdit.add(miAdd);
93
94         menuBar.add(mEdit);
95
96         mHelp.setText("Help");
97
98
miAbout.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.ev
ent.KeyEvent.VK_ENTER, java.awt.event.InputEvent.ALT_MASK));
99         miAbout.setText("About");
100        miAbout.addActionListener(new
java.awt.event.ActionListener() {
101            public void
actionPerformed(java.awt.event.ActionEvent evt) {
102                miAboutActionPerformed(evt);
103            }
104        });
105        mHelp.add(miAbout);
106
107        menuBar.add(mHelp);
108
109        setJMenuBar(menuBar);
110
111        java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
112        setBounds((screenSize.width-629)/2, (screenSize.height-
676)/2, 629, 676);
113    } // </editor-fold>
114
115    private void miExitActionPerformed(java.awt.event.ActionEvent
evt) {
116        int option = JOptionPane.showConfirmDialog(this, "Are you
sure?", "Exit", JOptionPane.YES_NO_OPTION);
117        if (option == JOptionPane.YES_OPTION) {
118            System.exit(0);
119        }
120    }
121
122    private void
miAboutActionPerformed(java.awt.event.ActionEvent evt) {

```

```

123         // display a dialog
124         JLabel lb = new JLabel("<html><body>Created by: <b>Harry
& Popi</b></body></html>");
125         Font font = lb.getFont();
126         lb.setFont(font.deriveFont(36f));
127
128         lb.setPreferredSize(new Dimension(400, 150));
129         JDialog dlg = new JDialog(this, "About " + getTitle(),
true);
130         dlg.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
131         lb.setHorizontalAlignment(JLabel.CENTER);
132
133         dlg.getContentPane().add(lb);
134
135         dlg.pack();
136         dlg.setLocationRelativeTo(null);
137         dlg.setVisible(true);
138     }
139
140     private void miAddActionPerformed(java.awt.event.ActionEvent
evt) {
141         pnContent.removeAll();
142
143
144         pnAdd.removeAll();
145         pnAdd.add(new AddPanel());
146
147         pnContent.add(pnAdd);
148
149         pnContent.revalidate();
150         pnContent.repaint();
151     }
152
153     private void
miSearchActionPerformed(java.awt.event.ActionEvent evt) {
154         pnContent.removeAll();
155
156         pnContent.add(pnSearch);
157         // refresh:
158         pnContent.revalidate();
159         pnContent.repaint();
160     }
161
162     /**
163     *
164     * @param args
165     */
166     public static void main(String args[]) {
167         // set the native skin:
168         try {
169             UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
170
171         } catch (Exception ex) {
172         }
173         //start the app:
174         java.awt.EventQueue.invokeLater(new Runnable() {
175
176             public void run() {
177                 new MainFrame().setVisible(true);
178             }
179         });

```

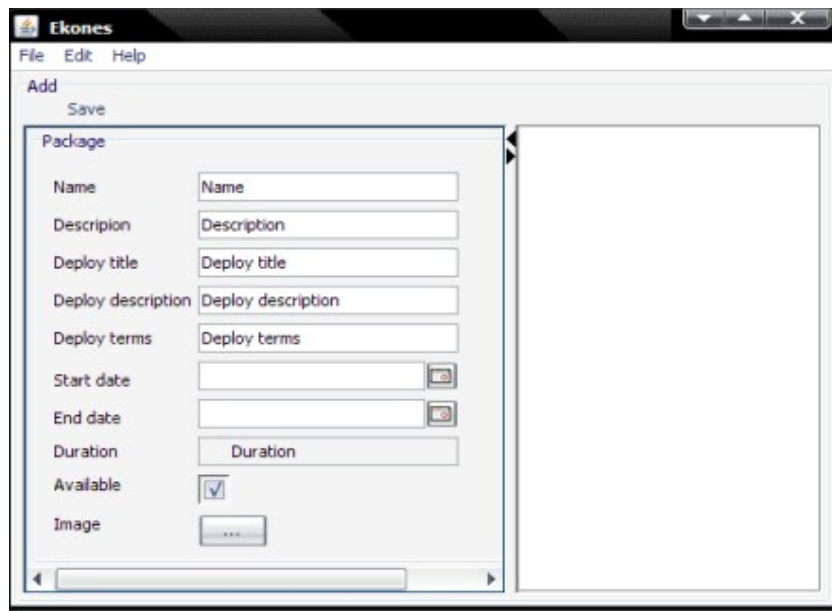
```

179     });
180 }
181 // Variables declaration - do not modify
182 private javax.swing.JMenu mEdit;
183 private javax.swing.JMenu mFile;
184 private javax.swing.JMenu mHelp;
185 private javax.swing.JMenuBar menuBar;
186 private javax.swing.JMenuItem miAbout;
187 private javax.swing.JMenuItem miAdd;
188 private javax.swing.JMenuItem miExit;
189 private javax.swing.JMenuItem miSearch;
190 private javax.swing.JPanel pnAdd;
191 private javax.swing.JPanel pnContent;
192 private javax.swing.JPanel pnSearch;
193 // End of variables declaration
194 }
195

```

7.1.3 AddPanel

Εδώ υλοποιείται το παράθυρο όπου προσθέτουμε νέα πακέτα.



```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  /*
7  * AddPanel.java
8  *
9  * Created on 2009.01.21., 18:41:20
10 */
11 package project1;
12
13 import java.awt.image.BufferedImage;
14 import java.io.IOException;
15 import project1.gui.PackagesPanel;
16 import project1.gui.DayPanel;
17 import java.beans.PropertyChangeEvent;

```

```

18 import java.beans.PropertyChangeListener;
19 import java.io.ByteArrayInputStream;
20 import java.io.ByteArrayOutputStream;
21 import java.sql.PreparedStatement;
22 import java.sql.ResultSet;
23 import java.sql.SQLException;
24 import java.sql.Statement;
25 import javax.imageio.ImageIO;
26 import javax.swing.JOptionPane;
27 import project1.util.DBConnection;
28
29 /**
30  *
31  * @author MatheSzabolcs
32  */
33 public class AddPanel extends javax.swing.JPanel {
34
35     PackagesPanel packagesPanel = new PackagesPanel();
36
37     /** Creates new form AddPanel */
38     public AddPanel() {
39         initComponents();
40         pnPackage.add(packagesPanel);
41         packagesPanel.addPropertyChangeListener(new
DurationPropertyChangeListener());
42     }
43
44     /** This method is called from within the constructor to
45      * initialize the form.
46      * WARNING: Do NOT modify this code. The content of this
method is
47      * always regenerated by the Form Editor.
48      */
49     @SuppressWarnings("unchecked")
50     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
51     private void initComponents() {
52
53         splitMain = new javax.swing.JSplitPane();
54         scLeft = new javax.swing.JScrollPane();
55         pnPackage = new javax.swing.JPanel();
56         tbDays = new javax.swing.JTabbedPane();
57         toolbar = new javax.swing.JToolBar();
58         btSave = new javax.swing.JButton();
59
60         FormListener formListener = new FormListener();
61
62         setLayout(new java.awt.BorderLayout());
63
64         splitMain.setDividerLocation(550);
65         splitMain.setDividerSize(8);
66         splitMain.setContinuousLayout(true);
67         splitMain.setName("splitMain"); // NOI18N
68         splitMain.setOneTouchExpandable(true);
69
70         scLeft.setName("scLeft"); // NOI18N
71
72
pnPackage.setBorder(javax.swing.BorderFactory.createTitledBorder("Pac
kage"));
73         pnPackage.setName("pnPackage"); // NOI18N

```

```

74     pnPackage.setLayout(new java.awt.BorderLayout());
75     scLeft.setViewportView(pnPackage);
76
77     splitMain.setLeftComponent(scLeft);
78
79     tbDays.setName("tbDays"); // NOI18N
80     splitMain.setRightComponent(tbDays);
81
82     add(splitMain, java.awt.BorderLayout.CENTER);
83
84     toolbar.setRollover(true);
85     toolbar.setName("toolbar"); // NOI18N
86
87     btSave.setText("Save");
88     btSave.setFocusable(false);
89
btSave.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
90     btSave.setMaximumSize(new java.awt.Dimension(80, 30));
91     btSave.setMinimumSize(new java.awt.Dimension(80, 30));
92     btSave.setName("btSave"); // NOI18N
93     btSave.setPreferredSize(new java.awt.Dimension(80, 30));
94
btSave.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
95     btSave.addActionListener(formListener);
96     toolbar.add(btSave);
97
98     add(toolbar, java.awt.BorderLayout.PAGE_START);
99 }
100
101 // Code for dispatching events from components to event
handlers.
102
103 private class FormListener implements
java.awt.event.ActionListener {
104     FormListener() {}
105     public void actionPerformed(java.awt.event.ActionEvent
evt) {
106         if (evt.getSource() == btSave) {
107             AddPanel.this.btSaveActionPerformed(evt);
108         }
109     }
110 } // </editor-fold>
111
112 private void btSaveActionPerformed(java.awt.event.ActionEvent
evt) {
113     if (!packagesPanel.isDataCompleted()) {
114         JOptionPane.showMessageDialog(this, "Warning",
"Please complete the package data!", JOptionPane.WARNING_MESSAGE);
115         return;
116     }
117     // collect packages data:
118     String packName = packagesPanel.getPackName();
119     String description = packagesPanel.getDescription();
120     String deployTitle = packagesPanel.getDeployTitle();
121     String deployDesc = packagesPanel.getDeployDescription();
122     String deployTerms = packagesPanel.getDeployTerms();
123     java.sql.Timestamp startDate = new
java.sql.Timestamp(packagesPanel.getStartDate().getTime());
124     java.sql.Timestamp endDate = new
java.sql.Timestamp(packagesPanel.getEndDate().getTime());
125     int duration = packagesPanel.getDuration();

```

```

126         int available = packagesPanel.getAvailable() ? 1 : 0;
127         BufferedImage img = packagesPanel.getBufferedImage();
128
129         // collect the days data
130
131         String[][] daysData = new String[duration][2];
132         for (int i = 0; i < daysData.length; i++) {
133             //tbDays.getTabCount()
134             DayPanel dayPanel = (DayPanel)
tbDays.getComponentAt(i);
135             daysData[i][0] = dayPanel.getTitle();
136             daysData[i][1] = dayPanel.getDescription();
137         }
138
139
140         java.sql.Connection myConnection = null;
141         try {
142             myConnection =
DBConnection.getDBConnection().createConnection();
143
144         } catch (SQLException e) {
145             System.out.println("Failed to get connection");
146             e.printStackTrace();
147             JOptionPane.showMessageDialog(this, "Error", "Could
not connect to database", JOptionPane.ERROR_MESSAGE);
148             return;
149         }
150
151         assert (myConnection != null);
152
153         Integer imgID = null; // failed
154
155         // convert the image to an input stream:
156         ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream();
157         try {
158             ImageIO.write(img, "PNG", byteArrayOutputStream);
159         } catch (IOException ex) { // not to file written...
160             }
161         byte[] bytes = byteArrayOutputStream.toByteArray();
162         ByteArrayInputStream bais = new
ByteArrayInputStream(bytes);
163
164         if (bytes.length > 65535) {
165             // can be a problem because maybe extends the BLOB
(default) maximum size, depends on server settings
166         }
167
168         try { // insert the image: - no problem if fail!
169             PreparedStatement pstmt =
myConnection.prepareStatement("INSERT INTO images (Ima_data) VALUES
(?)");
170             pstmt.setBinaryStream(1, bais, bytes.length);
171             pstmt.execute();
172             ResultSet rs = pstmt.getGeneratedKeys();
173             if (rs.next()) {
174                 imgID = rs.getInt(1);
175             }
176             rs.close();
177             pstmt.close();
178         } catch (SQLException ex) {

```

```

179         ex.printStackTrace();
180         JOptionPane.showMessageDialog(this, "Error",
"Couldn't save to database the image.\n" + ex.getMessage(),
JOptionPane.ERROR_MESSAGE);
181         // no problm, lets continue.
182     }
183
184     Integer packID = null;
185     // insert the package:
186     try {
187         Statement st = myConnection.createStatement();
188         String sqlInsertPackages = "INSERT INTO packages " +
189             "(Pac_name, Pac_descr, Pac_start_date,
Pac_end_date, " +
190             "Pac_duration, Pac_available,
Pac_deploy_title, Pac_deploy_descr, Pac_deploy_terms, Ima_id) " +
191             "VALUES (" +
192             "\"" + packName + "\", " +
193             "\"" + description + "\", " +
194             "\"" + startDate + "\", " +
195             "\"" + endDate + "\", " +
196             "\"" + duration + "\", " +
197             "\"" + available + "\", " +
198             "\"" + deployTitle + "\", " +
199             "\"" + deployDesc + "\", " +
200             "\"" + deployTerms + "\", " +
201             "\"" + imgID + "\" " +
202             ") ";
203         st.executeUpdate(sqlInsertPackages);
204         st.getGeneratedKeys();
205
206         ResultSet rs = st.getGeneratedKeys();
207         if (rs.next()) {
208             packID = rs.getInt(1);
209         }
210         rs.close();
211         st.close();
212
213     } catch (SQLException ex) {
214         ex.printStackTrace();
215         JOptionPane.showMessageDialog(this, "Error",
"Couldn't save to database the package.\n" + ex.getMessage(),
JOptionPane.ERROR_MESSAGE);
216         return;
217     }
218
219     // save the days:
220
221     for (int i = 0; i < daysData.length; i++) {
222
223         try {
224             Statement st = myConnection.createStatement();
225             String sqlInsert = "INSERT INTO deploy_days
(Pac_id, Dd_day, Dd_title, Dd_descr) VALUES (" +
226             "\"" + packID + "\", " +
227             "\"" + (i + 1) + "\", " +
228             "\"" + daysData[i][0] + "\", " +
229             "\"" + daysData[i][1] + "\" " +
230             ") ";
231             st.executeUpdate(sqlInsert);
232             st.close();

```



```

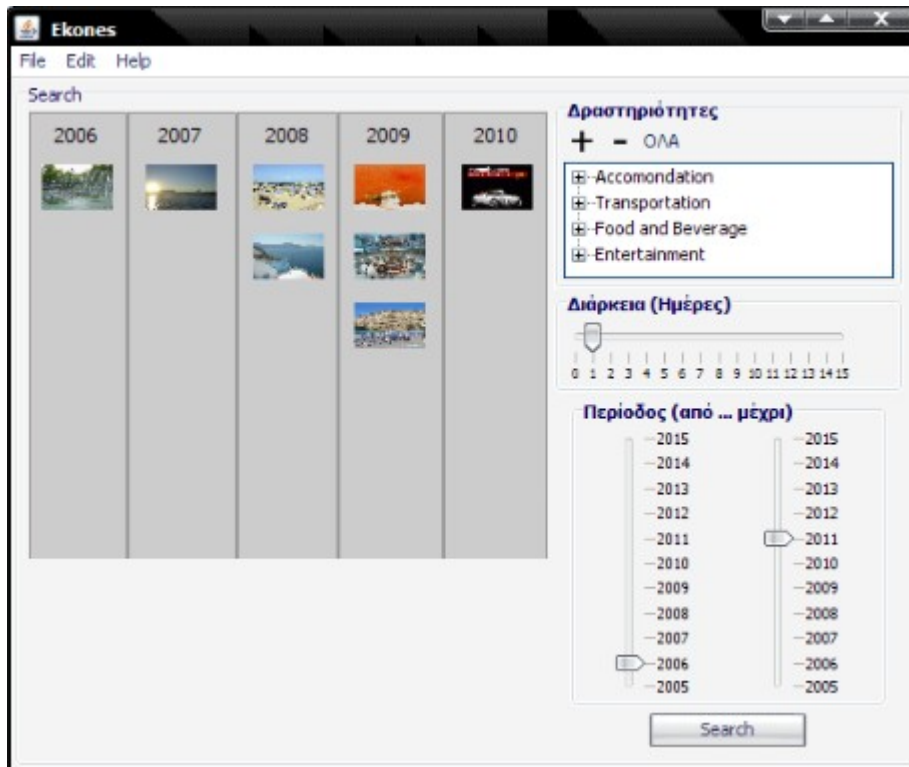
233         } catch (SQLException ex) {
234             ex.printStackTrace();
235             JOptionPane.showMessageDialog(this, "Error",
"Couldn't save to database the day " + (i + 1) + ".\n" +
ex.getMessage(), JOptionPane.ERROR_MESSAGE);
236         }
237     }
238
239     JOptionPane.showMessageDialog(this, "Info", "Data are
saved to databse successful", JOptionPane.INFORMATION_MESSAGE);
240
241     //clear :
242     packagesPanel.clear();
243
244     tbDays.removeAll();
245
246
247 }
248
249 // Variables declaration - do not modify
250 protected javax.swing.JButton btSave;
251 protected javax.swing.JPanel pnPackage;
252 protected javax.swing.JScrollPane scLeft;
253 protected javax.swing.JSplitPane splitMain;
254 protected javax.swing.JTabbedPane tbDays;
255 protected javax.swing.JToolBar toolbar;
256 // End of variables declaration
257
258 private class DurationPropertyChangeListener implements
PropertyChangeListener {
259
260     @Override
261     public void propertyChange(PropertyChangeEvent evt) {
262         String propName = evt.getPropertyName();
263         if (PackagesPanel.DURATION_PROPERTY.equals(propName))
{
264             // duration Changed: set the number of days
265             int duration = (Integer) evt.getNewValue();
266
267             // remove the days if needed:
268             while (tbDays.getTabCount() > duration) {
269                 tbDays.remove(tbDays.getTabCount() - 1);
270             }
271
272             // add days if needed:
273             while (tbDays.getTabCount() < duration) {
274                 tbDays.addTab("Day " + (tbDays.getTabCount()
+ 1), new DayPanel());
275             }
276         }
277     }
278 }
279 }
280

```

7.1.4 SearchPanel

Σε αυτό το μέρος του κώδικα αναπτύσουμε το παραθυρικό περιβάλλον όπου πραγματοποιείται η αναζήτηση των πακέτων που

Θέλουμε να προβάλλουμε σύμφωνα με τα κριτήρια που θέτουμε κάθε φορά.



```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 /*
7  * SearchPanel.java
8  *
9  * Created on 2009.01.23., 13:12:41
10 */
11 package project1;
12
13 import java.sql.Statement;
14 import java.awt.Color;
15 import java.awt.Dimension;
16 import java.awt.Font;
17 import java.awt.event.ActionEvent;
18 import java.awt.event.ActionListener;
19 import java.awt.event.KeyEvent;
20 import java.awt.image.BufferedImage;
21 import java.io.IOException;
22 import java.io.InputStream;
23 import java.sql.ResultSet;
24 import java.sql.SQLException;
25 import java.util.ArrayList;
26 import java.util.Calendar;
27 import java.util.Hashtable;
28 import java.util.Iterator;
29 import java.util.LinkedHashMap;
30 import java.util.Vector;
31 import javax.imageio.ImageIO;
```

```

32 import javax.swing.ImageIcon;
33 import javax.swing.JButton;
34 import javax.swing.JDialog;
35 import javax.swing.JFrame;
36 import javax.swing.JLabel;
37 import javax.swing.JOptionPane;
38 import javax.swing.JPanel;
39 import javax.swing.JScrollPane;
40 import javax.swing.border.LineBorder;
41 import javax.swing.tree.DefaultTreeModel;
42 import javax.swing.tree.TreePath;
43 import project1.data.CategoryNode;
44 import project1.data.PackageDetailsData;
45 import project1.gui.DetailViewer;
46 import project1.gui.NoIconTreeCellRenderer;
47 import project1.util.DBConnection;
48
49 /**
50  *
51  * @author Harry
52  */
53 public class SearchPanel extends javax.swing.JPanel {
54
55     private static Calendar cal = Calendar.getInstance(); // whole
the calendar variable, to get the YEAR
56     private Color colorButtonsBackground = Color.white;
57     private Font fontButtons = new java.awt.Font("Arial", 0, 10);
58     private ActionListener buttonActionListener = new
ButtonActionListener();
59     private Vector<JButton> vButtons = new Vector<JButton>(20); //
the action comands contains the package ID
60     private NoIconTreeCellRenderer noIconTreeCellRenderer = new
NoIconTreeCellRenderer();
61
62     /** Creates new form SearchPanel */
63     public SearchPanel() {
64         initComponents();
65
66         buildTreeData();
67     }
68
69     /**
70      * get data from database (categories)
71      */
72     private void buildTreeData() {
73         java.sql.Connection myConnection = null;
74         try {
75             myConnection =
DBConnection.getDBConnection().createConnection();
76
77             } catch (SQLException e) {
78                 System.out.println("Failed to get connection");
79                 e.printStackTrace();
80                 JOptionPane.showMessageDialog(this, "Error", "Could
not connect to database", JOptionPane.ERROR_MESSAGE);
81                 return;
82             }
83
84             try {
85                 Statement stmt = myConnection.createStatement();

```

```

86         String sql = "SELECT Cat_id, Cat_name, Cat_parent_id
" +
87             "FROM categories ORDER BY Cat_parent_id ";
88         ResultSet rs = stmt.executeQuery(sql);
89         // the id->CategoryNode
90         ArrayList<CategoryNode> alNodes = new
ArrayList<CategoryNode>();
91         // the parent ID->childs array
92         Hashtable<Integer, ArrayList<CategoryNode>>
htparentIDNodes = new Hashtable<Integer, ArrayList<CategoryNode>>();
93
94
95         while (rs.next()) {
96             CategoryNode node = new CategoryNode();
97             node.setId(rs.getInt("Cat_id"));
98             node.setName(rs.getString("Cat_name"));
99             node.setParentID(rs.getInt("Cat_parent_id"));
100
101             alNodes.add(node);
102
103             ArrayList<CategoryNode> al =
htparentIDNodes.get(node.getParentID());
104             if (al == null) {
105                 al = new ArrayList<CategoryNode>();
106             }
107             al.add(node);
108             htparentIDNodes.put(node.getParentID(), al);
109
110         }
111
112         ArrayList<CategoryNode> queue = new
ArrayList<CategoryNode>();
113         CategoryNode root = alNodes.remove(0);
114         queue.add(root);
115
116         while (queue.size() > 0) {
117             CategoryNode curNode = queue.remove(0);
118             ArrayList<CategoryNode> childNodes =
htparentIDNodes.remove(curNode.getId());
119             if (childNodes != null) {
120                 for (CategoryNode child : childNodes) {
121                     queue.add(child);
122                     curNode.add(child);
123                 }
124             }
125         }
126
127         DefaultTreeModel treeModel = new
DefaultTreeModel(root);
128         tree.setModel(treeModel);
129         tree.setCellRenderer(noIconTreeCellRenderer);
130
131
132     } catch (SQLException ex) {
133         ex.printStackTrace();
134     }
135 }
136
137 /** This method is called from within the constructor to
138     * initialize the form.

```

```

139     * WARNING: Do NOT modify this code. The content of this
method is
140     * always regenerated by the Form Editor.
141     */
142     @SuppressWarnings("unchecked")
143     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
144     private void initComponents() {
145
146         calendar = new javax.swing.JPanel();
147         criteria = new javax.swing.JPanel();
148         pnCategories = new javax.swing.JPanel();
149         scTree = new javax.swing.JScrollPane();
150         tree = new javax.swing.JTree();
151         tbCategories = new javax.swing.JToolBar();
152         btExpandAll = new javax.swing.JButton();
153         btCollapseAll = new javax.swing.JButton();
154         selectall = new javax.swing.JButton();
155         pnDuration = new javax.swing.JPanel();
156         sliderDuration = new javax.swing.JSlider();
157         pnPackageFromTo = new javax.swing.JPanel();
158         sliderStartDate = new javax.swing.JSlider();
159         sliderEndDate = new javax.swing.JSlider();
160         pnSearchButton = new javax.swing.JPanel();
161         buttonSearch = new javax.swing.JButton();
162
163         FormListener formListener = new FormListener();
164
165         setLayout(new javax.swing.BoxLayout(this,
javax.swing.BoxLayout.X_AXIS));
166
167         calendar.setMinimumSize(new java.awt.Dimension(500,
491));
168         calendar.setName("calendar"); // NOI18N
169         calendar.setPreferredSize(new java.awt.Dimension(136,
100));
170         calendar.setLayout(new java.awt.GridBagLayout());
171         add(calendar);
172
173         criteria.setMaximumSize(new java.awt.Dimension(220,
32767));
174         criteria.setMinimumSize(new java.awt.Dimension(220, 0));
175         criteria.setName("criteria"); // NOI18N
176         criteria.setPreferredSize(new java.awt.Dimension(220,
100));
177
178         pnCategories.setBorder(javax.swing.BorderFactory.createTitledBorder(n
ull, "Αρροστηριότητες",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 1, 11)));
179         pnCategories.setName("pnCategories"); // NOI18N
180         pnCategories.setPreferredSize(new java.awt.Dimension(220,
200));
181         pnCategories.setRequestFocusEnabled(false);
182         pnCategories.setLayout(new java.awt.BorderLayout());
183
184         scTree.setName("scTree"); // NOI18N
185
186         tree.setName("tree"); // NOI18N

```

```

187         tree.setRootVisible(false);
188         tree.setShowsRootHandles(true);
189         tree.addKeyListener(formListener);
190         scTree.setViewportView(tree);
191
192         pnCategories.add(scTree, java.awt.BorderLayout.CENTER);
193
194         tbCategories.setFloatable(false);
195         tbCategories.setRollover(true);
196         tbCategories.setName("tbCategories"); // NOI18N
197
198         btExpandAll.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/project1/images/plus.p
ng"))); // NOI18N
199         btExpandAll.setToolTipText("Expand all");
200         btExpandAll.setFocusable(false);
201
btExpandAll.setHorizontalTextPosition(javax.swing.SwingConstants.CENT
ER);
202         btExpandAll.setName("btExpandAll"); // NOI18N
203
btExpandAll.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM
);
204         btExpandAll.addActionListener(formListener);
205         tbCategories.add(btExpandAll);
206
207         btCollapseAll.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/project1/images/minus.
png"))); // NOI18N
208         btCollapseAll.setToolTipText("Colapse all");
209         btCollapseAll.setFocusable(false);
210
btCollapseAll.setHorizontalTextPosition(javax.swing.SwingConstants.CEN
TER);
211         btCollapseAll.setName("btCollapseAll"); // NOI18N
212
btCollapseAll.setVerticalTextPosition(javax.swing.SwingConstants.BOTTO
M);
213         btCollapseAll.addActionListener(formListener);
214         tbCategories.add(btCollapseAll);
215
216         selectall.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/project1/images/minus.
png"))); // NOI18N
217         selectall.setToolTipText("Colapse all");
218         selectall.setFocusable(false);
219
selectall.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER
);
220         selectall.setName("selectall"); // NOI18N
221
selectall.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
222         selectall.addActionListener(formListener);
223         tbCategories.add(selectall);
224
225         pnCategories.add(tbCategories,
java.awt.BorderLayout.PAGE_START);
226
227         criteria.add(pnCategories);
228

```

```

229
pnDuration.setBorder(javax.swing.BorderFactory.createTitledBorder(nul
1, "Διάρκεια (Ημέρες)",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 1, 11));
230     pnDuration.setName("pnDuration"); // NOI18N
231     pnDuration.setPreferredSize(new java.awt.Dimension(220,
80));
232     pnDuration.setRequestFocusEnabled(false);
233     pnDuration.setLayout(new java.awt.BorderLayout());
234
235     sliderDuration.setFont(new java.awt.Font("Tahoma", 0,
8));
236     sliderDuration.setMajorTickSpacing(1);
237     sliderDuration.setMaximum(15);
238     sliderDuration.setMinorTickSpacing(1);
239     sliderDuration.setPaintLabels(true);
240     sliderDuration.setPaintTicks(true);
241     sliderDuration.setSnapToTicks(true);
242     sliderDuration.setToolTipText("okkkk");
243     sliderDuration.setValue(1);
244     sliderDuration.setName("sliderDuration"); // NOI18N
245     sliderDuration.setPreferredSize(new
java.awt.Dimension(180, 43));
246     pnDuration.add(sliderDuration,
java.awt.BorderLayout.WEST);
247
248     criteria.add(pnDuration);
249
250
pnPackageFromTo.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Περίοδος (από ... μέχρι)",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 1, 11));
251     pnPackageFromTo.setName("pnPackageFromTo"); // NOI18N
252     pnPackageFromTo.setPreferredSize(new
java.awt.Dimension(200, 200));
253     pnPackageFromTo.setRequestFocusEnabled(false);
254     pnPackageFromTo.setLayout(new java.awt.GridLayout(1, 0));
255
256     sliderStartDate.setFont(new java.awt.Font("Tahoma", 0,
10));
257     sliderStartDate.setMajorTickSpacing(1);
258     sliderStartDate.setMaximum(2015);
259     sliderStartDate.setMinimum(2005);
260     sliderStartDate.setMinorTickSpacing(1);
261
sliderStartDate.setOrientation(javax.swing.JSlider.VERTICAL);
262     sliderStartDate.setPaintLabels(true);
263     sliderStartDate.setPaintTicks(true);
264     sliderStartDate.setSnapToTicks(true);
265     sliderStartDate.setValue(2006);
266     sliderStartDate.setName("sliderStartDate"); // NOI18N
267     sliderStartDate.setRequestFocusEnabled(false);
268     pnPackageFromTo.add(sliderStartDate);
269
270     sliderEndDate.setFont(new java.awt.Font("Tahoma", 0,
10));
271     sliderEndDate.setMajorTickSpacing(1);

```

```

272         sliderEndDate.setMaximum(2015);
273         sliderEndDate.setMinimum(2005);
274         sliderEndDate.setMinorTickSpacing(1);
275
sliderEndDate.setOrientation(javax.swing.JSlider.VERTICAL);
276         sliderEndDate.setPaintLabels(true);
277         sliderEndDate.setPaintTicks(true);
278         sliderEndDate.setSnapToTicks(true);
279         sliderEndDate.setValue(2011);
280         sliderEndDate.setName("sliderEndDate"); // NOI18N
281         pnPackageFromTo.add(sliderEndDate);
282
283         criteria.add(pnPackageFromTo);
284
285         pnSearchButton.setName("pnSearchButton"); // NOI18N
286         pnSearchButton.setPreferredSize(new
java.awt.Dimension(200, 200));
287         pnSearchButton.setRequestFocusEnabled(false);
288
289         buttonSearch.setText("Search");
290         buttonSearch.setName("buttonSearch"); // NOI18N
291         buttonSearch.addActionListener(formListener);
292
293         javax.swing.GroupLayout pnSearchButtonLayout = new
javax.swing.GroupLayout(pnSearchButton);
294         pnSearchButton.setLayout(pnSearchButtonLayout);
295         pnSearchButtonLayout.setHorizontalGroup(
296
pnSearchButtonLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
297
.addGroup(pnSearchButtonLayout.createSequentialGroup())
298             .addGap(50, 50, 50)
299             .addComponent(buttonSearch,
javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
300             .addContainerGap(51, Short.MAX_VALUE)
301         );
302         pnSearchButtonLayout.setVerticalGroup(
303
pnSearchButtonLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
304
.addGroup(pnSearchButtonLayout.createSequentialGroup())
305             .addComponent(buttonSearch)
306             .addContainerGap(177, Short.MAX_VALUE)
307         );
308
309         criteria.add(pnSearchButton);
310
311         add(criteria);
312     }
313
314     // Code for dispatching events from components to event
handlers.
315
316     private class FormListener implements
java.awt.event.ActionListener, java.awt.event.KeyListener {
317         FormListener() {}
318         public void actionPerformed(java.awt.event.ActionEvent
evt) {

```



```

319         if (evt.getSource() == btExpandAll) {
320             SearchPanel.this.btExpandAllActionPerformed(evt);
321         }
322         else if (evt.getSource() == btColapseAll) {
323             SearchPanel.this.btColapseAllActionPerformed(evt);
324         }
325         else if (evt.getSource() == buttonSearch) {
326             SearchPanel.this.buttonSearchActionPerformed(evt);
327         }
328         else if (evt.getSource() == selectall) {
329             SearchPanel.this.selectallActionPerformed(evt);
330         }
331     }
332
333     public void keyPressed(java.awt.event.KeyEvent evt) {
334     }
335
336     public void keyReleased(java.awt.event.KeyEvent evt) {
337         if (evt.getSource() == tree) {
338             SearchPanel.this.treeKeyReleased(evt);
339         }
340     }
341
342     public void keyTyped(java.awt.event.KeyEvent evt) {
343     }
344 } // </editor-fold>
345
346     private void
buttonSearchActionPerformed(java.awt.event.ActionEvent evt) {
347         // TODO add your handling code here:
348
349         //get the tree selectionID -s
350         TreePath[] selPaths = tree.getSelectionPaths();
351         ArrayList<Integer> alSelIDs = null;
352
353         if (selPaths != null) {
354             //no selection, do not filter with this?
355             alSelIDs = new ArrayList<Integer>();
356             for (TreePath selPath : selPaths) {
357                 CategoryNode selNode = (CategoryNode)
selPath.getLastPathComponent();
358                 alSelIDs.add(selNode.getId());
359             }
360         }
361
362         String selIDs = null;
363         if (selPaths != null) {
364             selIDs = "";
365             for (int i = 0; i < alSelIDs.size(); i++) {
366                 selIDs += " " + alSelIDs.get(i) + ",";
367             }
368             selIDs = selIDs.substring(0, selIDs.length() - 1);
369         }
370
371         java.sql.Connection myConnection = null;
372         try {
373             myConnection =
DBConnection.getConnection().createConnection();
374

```

```

375     } catch (SQLException e) {
376         System.out.println("Failed to get connection");
377         e.printStackTrace();
378         JOptionPane.showMessageDialog(this, "Error", "Could
not connect to database", JOptionPane.ERROR_MESSAGE);
379         return;
380     }
381
382
383     try {
384         Statement stmt = myConnection.createStatement();
385         //rs = stmt.executeQuery("select Act_id,Pac_id,
Act_descr, Loc_name from activities, locations where Loc_id_to =
locations.Loc_id");
386
387         String sql = "SELECT Pac_id, Pac_name,
Pac_start_date, Ima_data FROM packages " +//
388             "JOIN images ON images.Ima_id =
packages.Ima_id " +//
389             "WHERE Pac_duration > " +
sliderDuration.getValue() + " " +//
390             "AND YEAR(Pac_start_date) >= " +
sliderStartDate.getValue() + " " +//
391             "AND YEAR(Pac_end_date) <= " +
sliderEndDate.getValue() + " " +//
392             "ORDER BY Pac_start_date ASC";
393         //System.out.println("sql = " + sql);
394
395         if (selIDs != null) { // we have selection
396             sql = "SELECT DISTINCT packages.Pac_id, Pac_name,
Pac_start_date, Ima_data FROM package_categories_vw " +//
397                 "JOIN packages ON packages.Pac_id =
package_categories_vw.Pac_id " +//
398                 "JOIN images ON images.Ima_id =
packages.Ima_id " +//
399                 "WHERE Cat_id IN (" + selIDs + ") " +//
400                 "AND Pac_duration > " +
sliderDuration.getValue() + " " +//
401                 "AND YEAR(Pac_start_date) >= " +
sliderStartDate.getValue() + " " +//
402                 "AND YEAR(Pac_end_date) <= " +
sliderEndDate.getValue() + " " +//
403                 "ORDER BY Pac_start_date ASC";
404         }
405         // System.out.println("sql = " + sql);
406
407         // the view creation code:
408         /* select distinct
409         `activity_categories`.`Cat_id` AS `Cat_id`,
410         `activities`.`Pac_id` AS `Pac_id`
411         from (`activities` join `activity_categories`)
412         where (`activities`.`Act_id` =
`activity_categories`.`Act_id`)
413         */
414         ResultSet rs = stmt.executeQuery(sql);
415
416
417         refreshPanels();
418
419         // group by start date

```

```

420         LinkedHashMap<String, ArrayList<JButton>> hm = new
LinkedHashMap<String, ArrayList<JButton>>();
421
422
423
424         int nr = 1;
425         while (rs.next()) {
426
427             int id = rs.getInt("Pac_id");
428             String pacName = rs.getString("Pac_name");
429             java.sql.Timestamp startDate =
rs.getTimestamp("Pac_start_date");
430
431             InputStream is = rs.getBinaryStream("Ima_data");
432
433             BufferedImage bi = ImageIO.read(is);
434
435             JButton button = createButton(bi, pacName, "" +
id);
436             button.addActionListener(buttonActionListener);
437             vButtons.add(button);
438
439             cal.setTimeInMillis(startDate.getTime());
440             int year = cal.get(Calendar.YEAR);
441
442             ArrayList<JButton> values = hm.get("" + year);
443
444             if (values == null) {
445                 values = new ArrayList<JButton>();
446             }
447             values.add(button);
448
449             hm.put("" + year, values);
450
451             nr++;
452         } // each result
453
454
455         rs.close();
456         stmt.close();
457         myConnection.close();
458
459         // display the buttons in the group, max 7 group
allowed
460
461         Iterator<String> keys = hm.keySet().iterator();
462         for (int i = 0; ((i < 7) && (keys.hasNext())); i++) {
463             String key = keys.next(); // this is the column
name
464             //System.out.println("key = " + key);
465             ArrayList<JButton> buttons = hm.get(key); // al
buttons at this group
466
467
468             // create a cal panel with lael and buttons:
469
470             // copy from the GUI designer:
471             JPanel pnColumn = new JPanel();
472             pnColumn.setBackground(new Color(204, 204, 204));
473             pnColumn.setBorder(new LineBorder(new Color(153,
153, 153), 1, true));

```

```

474         pnColumn.setMaximumSize(new Dimension(100, 700));
475         pnColumn.setMinimumSize(new Dimension(80, 700));
476         pnColumn.setPreferredSize(new Dimension(82,
477 700));
478         pnColumn.add(new JLabel(key));
479         for (JButton button : buttons) {
480             pnColumn.add(button);
481         }
482
483         calendar.add(pnColumn);
484     }
485     // set for the
486
487
488     calendar.revalidate();
489     calendar.repaint();
490
491     } catch (Exception e) {
492         ///System.out.println("failed executing query");
493         e.printStackTrace();
494     }
495
496 }
497
498 private void treeKeyReleased(java.awt.event.KeyEvent evt) {
499     if (KeyEvent.VK_ESCAPE == evt.getKeyCode()) {
500         tree.clearSelection();
501     }
502 }
503
504 private void
btExpandAllActionPerformed(java.awt.event.ActionEvent evt) {
505     int row = 0;
506     while (row < tree.getRowCount()) {
507         tree.expandRow(row);
508         row++;
509     }
510
511 }
512
513 private void
btCollapseAllActionPerformed(java.awt.event.ActionEvent evt) {
514     int row = tree.getRowCount() - 1;
515     while (row >= 0) {
516         tree.collapseRow(row);
517         row--;
518     }
519
520 }
521
522 private void
selectallActionPerformed(java.awt.event.ActionEvent evt) {
523     tree.clearSelection();
524 }
525
526 // Variables declaration - do not modify
527 protected javax.swing.JButton btCollapseAll;
528 protected javax.swing.JButton btExpandAll;
529 protected javax.swing.JButton buttonSearch;
530 protected javax.swing.JPanel calendar;

```

```

531     protected javax.swing.JPanel criteria;
532     protected javax.swing.JPanel pnCategories;
533     protected javax.swing.JPanel pnDuration;
534     protected javax.swing.JPanel pnPackageFromTo;
535     protected javax.swing.JPanel pnSearchButton;
536     protected javax.swing.JScrollPane scTree;
537     protected javax.swing.JButton selectall;
538     protected javax.swing.JSlider sliderDuration;
539     protected javax.swing.JSlider sliderEndDate;
540     protected javax.swing.JSlider sliderStartDate;
541     protected javax.swing.JToolBar tbCategories;
542     protected javax.swing.JTree tree;
543     // End of variables declaration
544
545     private class ButtonActionListener implements ActionListener
546     {
547         @Override
548         public void actionPerformed(ActionEvent e) {
549             // Shows the dialog
550             String ac = e.getActionCommand(); // the package ID
551             //JOptionPane.showMessageDialog(MainFrame.this,
552             "Action Command is:" + ac);
553             try {
554                 // the package ID
555                 //JOptionPane.showMessageDialog(MainFrame.this,
556                 "Action Command is:" + ac);
557                 PackageDetailsData detailsData = new
558                 PackageDetailsData(ac);
559                 JDialog dlg = new JDialog((JFrame) null,
560                 "Details", true);
561                 dlg.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
562                 DetailViewer detailViewer = new DetailViewer();
563                 detailViewer.setData(detailsData);
564                 dlg.getContentPane().add(new
565                 JScrollPane(detailViewer));
566                 dlg.setPreferredSize(new Dimension(1000, 650));
567                 dlg.pack();
568                 dlg.setLocationRelativeTo(null); // center on
569                 screen
570                 dlg.setVisible(true);
571             } catch (Exception ex) {
572                 ex.printStackTrace();
573             }
574         }
575     }
576
577     private void refreshPanels() {
578         while (vButtons.size() > 0) {
579             // remove from cache:
580             JButton button = vButtons.remove(0);
581             // remove from action listener to avoid memory leak:
582             button.removeActionListener(buttonActionListener);
583             // remove from the parent:
584             button.getParent().remove(button);
585         }
586     }

```

```

584     calendar.removeAll();
585     // set the column header values to empty String:
586
587 }
588
589 /**
590  * Create a custom JButton, can be a class...
591  * @param title
592  * @param tooltip
593  * @param actionCommand
594  * @return
595  */
596 private JButton createButton(BufferedImage img, String
tooltip, String actionCommand) {
597     ImageIcon icon = null;
598     if(img != null){
599         icon = new ImageIcon(img.getScaledInstance(39, 25,
BufferedImage.SCALE_DEFAULT));
600     }
601     //System.out.println("nr = " + nr);
602     JButton result = new JButton(icon);
603     result.setToolTipText(tooltip);
604     result.setContentAreaFilled(false);
605     result.setActionCommand(actionCommand);
606     result.setBackground(colorButtonsBackground);
607     result.setFont(fontButtons);
608     // get the button nr digits, load them from resources,
merge digit images, set icon to the button
609
610
611     return result;
612 }
613 }
614

```

7.1.5 PackagePanel

Εδώ φτιάχνουμε το panel όπου θα συμπληρώνουμε τα στοιχεία του πακέτου που θέλουμε να προσθέσουμε στην βάση μας.

The screenshot shows a dialog box titled "Package" with the following fields and controls:

- Name:
- Description:
- Deploy title:
- Deploy description:
- Deploy terms:
- Start date:
- End date:
- Duration:
- Available:
- Image:

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  /*
7  * PackagesPanel.java
8  *
9  * Created on 2009.01.21., 20:44:25
10 */
11
12 package project1.gui;
13
14 import java.awt.Dimension;
15 import java.awt.Image;
16 import java.awt.image.BufferedImage;
17 import java.beans.PropertyChangeEvent;
18 import java.beans.PropertyChangeListener;
19 import java.io.File;
20 import java.io.IOException;
21 import java.util.Calendar;
22 import java.util.Date;
23 import javax.imageio.ImageIO;
24 import javax.swing.ImageIcon;
25 import javax.swing.JFileChooser;
26 import project1.util.Settings;
27
28 /**
29 *
30 * @author Harry
31 */
32 public class PackagesPanel extends javax.swing.JPanel {
33
34     public static final String DURATION_PROPERTY = "duation";
35
36     private static Calendar cal = Calendar.getInstance();
37     private BufferedImage img = null;
38     private Integer duration = null;
39
40     /** Creates new form PackagesPanel */
41     public PackagesPanel() {
42         initComponents();
43
44         DatePropertyChangeListener dateChangeListener = new
DatePropertyChangeListener();
45         dcStartDate.getDateEditor().addPropertyChangeListener(dateChangeListe
ner);
46         dcEndDate.getDateEditor().addPropertyChangeListener(dateChangeListene
r);
47     }
48
49
50     public boolean isDataCompleted() {
51         return ((img != null) && (dcStartDate.getDate() != null)
&& (dcEndDate != null) && (duration != null) );
52     }
53
54     public String getPackName(){
55         return tfName.getText();

```

```

56     }
57
58     public String getDescription() {
59         return tfDescription.getText();
60     }
61
62     public String getDeployTile() {
63         return tfDeployTile.getText();
64     }
65
66     public String getDeployDescription() {
67         return tfDeployDescription.getText();
68     }
69
70     public String getDeployTerms() {
71         return tfDeployTerms.getText();
72     }
73
74     public Date getStartDate() {
75         return dcStartDate.getDate();
76     }
77
78     public Date getEndDate() {
79         return dcEndDate.getDate();
80     }
81
82     public int getDuration() {
83         return duration;
84     }
85
86     public boolean getAvailable() {
87         return chbAvailable.isSelected();
88     }
89
90     public BufferedImage getBufferedImage() {
91         return img;
92     }
93
94     public void clear() {
95         tfName.setText("");
96         tfDescription.setText("");
97         tfDeployTile.setText("");
98         tfDeployDescription.setText("");
99         tfDeployTerms.setText("");
100        dcStartDate.setDate(null);
101        dcEndDate.setDate(null);
102        duration = null;
103        chbAvailable.setSelected(true);
104        img = null;
105    }
106
107
108    /** This method is called from within the constructor to
109     * initialize the form.
110     * WARNING: Do NOT modify this code. The content of this
method is
111     * always regenerated by the Form Editor.
112     */
113    @SuppressWarnings("unchecked")
114    // <editor-fold defaultstate="collapsed" desc="Generated
Code">

```



```

115     private void initComponents () {
116
117         lbName = new javax.swing.JLabel ();
118         tfName = new javax.swing.JTextField ();
119         lbDescription = new javax.swing.JLabel ();
120         tfDescription = new javax.swing.JTextField ();
121         lbDeployTitle = new javax.swing.JLabel ();
122         tfDeployTitle = new javax.swing.JTextField ();
123         lbDeployDescription = new javax.swing.JLabel ();
124         tfDeployDescription = new javax.swing.JTextField ();
125         lbDeployTerms = new javax.swing.JLabel ();
126         tfDeployTerms = new javax.swing.JTextField ();
127         lbStartDate = new javax.swing.JLabel ();
128         dcStartDate = new com.toedter.calendar.JDateChooser ();
129         lbEndDate = new javax.swing.JLabel ();
130         dcEndDate = new com.toedter.calendar.JDateChooser ();
131         lbDuration = new javax.swing.JLabel ();
132         tfDuration = new javax.swing.JTextField ();
133         lbAvailable = new javax.swing.JLabel ();
134         chbAvailable = new javax.swing.JCheckBox ();
135         lbImage = new javax.swing.JLabel ();
136         lbImageScaled = new javax.swing.JLabel ();
137         btChooseImage = new javax.swing.JButton ();
138
139         FormListener formListener = new FormListener ();
140
141         lbName.setText ("Name");
142         lbName.setName ("lbName"); // NOI18N
143
144         tfName.setText ("Name");
145         tfName.setName ("tfName"); // NOI18N
146
147         lbDescription.setText ("Description");
148         lbDescription.setName ("lbDescription"); // NOI18N
149
150         tfDescription.setText ("Description");
151         tfDescription.setName ("tfDescription"); // NOI18N
152
153         lbDeployTitle.setText ("Deploy title");
154         lbDeployTitle.setName ("lbDeployTitle"); // NOI18N
155
156         tfDeployTitle.setText ("Deploy title");
157         tfDeployTitle.setName ("tfDeployTitle"); // NOI18N
158
159         lbDeployDescription.setText ("Deploy description");
160         lbDeployDescription.setName ("lbDeployDescription"); //
NOI18N
161
162         tfDeployDescription.setText ("Deploy description");
163         tfDeployDescription.setName ("tfDeployDescription"); //
NOI18N
164
165         lbDeployTerms.setText ("Deploy terms");
166         lbDeployTerms.setName ("lbDeployTerms"); // NOI18N
167
168         tfDeployTerms.setText ("Deploy terms");
169         tfDeployTerms.setName ("tfDeployTerms"); // NOI18N
170
171         lbStartDate.setText ("Start date");
172         lbStartDate.setName ("lbStartDate"); // NOI18N
173

```

```

174         dcStartDate.setToolTipText ("dd.MM.yyyy");
175         dcStartDate.setDateFormatString ("dd.MM.yyyy");
176         dcStartDate.setName ("dcStartDate"); // NOI18N
177
178         lbEndDate.setText ("End date");
179         lbEndDate.setName ("lbEndDate"); // NOI18N
180
181         dcEndDate.setToolTipText ("dd.MM.yyyy");
182         dcEndDate.setDateFormatString ("dd.MM.yyyy");
183         dcEndDate.setName ("dcEndDate"); // NOI18N
184
185         lbDuration.setText ("Duration");
186         lbDuration.setName ("lbDuration"); // NOI18N
187
188         tfDuration.setEditable (false);
189
190         tfDuration.setHorizontalAlignment (javax.swing.JTextField.LEFT);
191         tfDuration.setText ("Duration");
192         tfDuration.setMargin (new java.awt.Insets (2, 21, 2, 21));
193         tfDuration.setName ("tfDuration"); // NOI18N
194
195         lbAvailable.setText ("Available");
196         lbAvailable.setName ("lbAvailable"); // NOI18N
197
198         chbAvailable.setSelected (true);
199         chbAvailable.setBorderPainted (true);
200         chbAvailable.setName ("chbAvailable"); // NOI18N
201
202         lbImage.setText ("Image");
203         lbImage.setName ("lbImage"); // NOI18N
204
205         lbImageScaled.setMaximumSize (new java.awt.Dimension (150,
150));
206         lbImageScaled.setMinimumSize (new java.awt.Dimension (150,
150));
207         lbImageScaled.setName ("lbImageScaled"); // NOI18N
208         lbImageScaled.setPreferredSize (new
java.awt.Dimension (150, 150));
209
210         btChooseImage.setText ("...");
211         btChooseImage.setName ("btChooseImage"); // NOI18N
212         btChooseImage.addActionListener (formListener);
213
214         javax.swing.GroupLayout layout = new
javax.swing.GroupLayout (this);
215         this.setLayout (layout);
216         layout.setHorizontalGroup (
layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
217             .addGroup (layout.createSequentialGroup ()
218                 .add (lbImageScaled)
219                 .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
t.LEADING)
220                     .addGroup (layout.createSequentialGroup ()
221                         .add (lbImageScaled,
javax.swing.GroupLayout.PREFERRED_SIZE, 196,
javax.swing.GroupLayout.PREFERRED_SIZE)
222                         .add (btChooseImage))
223                     .addGroup (layout.createSequentialGroup ()

```

```

224
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
t.LEADING)
225
        .addComponent(lbName)
226
        .addComponent(lbDescription)
227
        .addComponent(lbDeployTitle)
228
        .addComponent(lbDeployDescription)
229
        .addComponent(lbDeployTerms)
230
        .addComponent(lbStartDate)
231
        .addComponent(lbDuration)
232
        .addComponent(lbAvailable)
233
        .addComponent(lbImage)
234
        .addComponent(lbEndDate))
235
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
236
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
t.LEADING)
237
        .addComponent(btChooseImage)
238
        .addComponent(chbAvailable)
239
        .addComponent(tfDuration,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
240
        .addComponent(dcEndDate,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
241
        .addComponent(dcStartDate,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
242
        .addComponent(tfDeployTerms,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
243
        .addComponent(tfDeployDescription,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
244
        .addComponent(tfDeployTile,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
245
        .addComponent(tfDescription,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
246
        .addComponent(tfName,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE))
247
        .addContainerGap(109, Short.MAX_VALUE)))
248
    );
249
    layout.setVerticalGroup(
250
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
251
        .addGroup(layout.createSequentialGroup())
252
        .addContainerGap())
253
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
t.BASELINE)
254
        .addComponent(lbName)
255
        .addComponent(tfName))
256
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
257
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
t.BASELINE)
258
        .addComponent(tfDescription)
259
        .addComponent(lbDescription))
260
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
261
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
t.BASELINE)
262
        .addComponent(tfDeployTile)

```

```

263             .addComponent (lbDeployTitle) )
264
265 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
266
267 .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
268 t.BASELINE)
269             .addComponent (lbDeployDescription)
270             .addComponent (tfDeployDescription) )
271
272 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
273
274 .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
275 t.BASELINE)
276             .addComponent (tfDeployTerms)
277             .addComponent (lbDeployTerms) )
278
279 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
280
281 .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
282 t.TRAILING)
283             .addComponent (lbStartDate)
284             .addComponent (dcStartDate,
285 javax.swing.GroupLayout.PREFERRED_SIZE,
286 javax.swing.GroupLayout.DEFAULT_SIZE,
287 javax.swing.GroupLayout.PREFERRED_SIZE) )
288
289 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
290
291 .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
292 t.TRAILING)
293             .addComponent (dcEndDate,
294 javax.swing.GroupLayout.PREFERRED_SIZE,
295 javax.swing.GroupLayout.DEFAULT_SIZE,
296 javax.swing.GroupLayout.PREFERRED_SIZE)
297             .addComponent (lbEndDate) )
298
299 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
300
301 .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
302 t.BASELINE)
303             .addComponent (tfDuration)
304             .addComponent (lbDuration) )
305
306 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
307
308 .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
309 t.BASELINE)
310             .addComponent (lbAvailable)
311             .addComponent (chbAvailable) )
312
313 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
314
315 .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
316 t.LEADING)
317             .addComponent (lbImage)
318             .addComponent (btChooseImage) )
319
320 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
321
322 .addComponent (lbImageScaled,
323 javax.swing.GroupLayout.PREFERRED_SIZE, 161,
324 javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

294         .addContainerGap()
295     );
296 }
297
298 // Code for dispatching events from components to event
handlers.
299
300 private class FormListener implements
java.awt.event.ActionListener {
301     FormListener() {}
302     public void actionPerformed(java.awt.event.ActionEvent
evt) {
303         if (evt.getSource() == btChooseImage) {
304             PackagesPanel.this.btChooseImageActionPerformed(evt);
305         }
306     }
307 } // </editor-fold>
308
309 private void
btChooseImageActionPerformed(java.awt.event.ActionEvent evt) {
310
311     JFileChooser fc = new JFileChooser();
312     fc.setFileSelectionMode(JFileChooser.FILES_ONLY);
313     fc.setDialogTitle("Select image");
314     int returnVal = fc.showOpenDialog(this);
315
316     if (returnVal != JFileChooser.APPROVE_OPTION) {
317         return;
318     }
319     File file = fc.getSelectedFile();
320     if (!file.isFile()) {
321         return;
322     }
323
324     img = null;
325     // load the image
326     try {
327         img = ImageIO.read(file);
328     } catch (IOException e) {
329     }
330
331     if (img != null) { // display it
332         Dimension imgSize = Settings.getDetailImageSize();
333         Image scaledImg =
img.getScaledInstance(imgSize.width, imgSize.height,
Image.SCALE_DEFAULT);
334         ImageIcon icon = new ImageIcon(scaledImg);
335         lbImageScaled.setIcon(icon);
336     }
337 }
338
339
340 // Variables declaration - do not modify
341 protected javax.swing.JButton btChooseImage;
342 protected javax.swing.JCheckBox chbAvailable;
343 protected com.toedter.calendar.JDateChooser dcEndDate;
344 protected com.toedter.calendar.JDateChooser dcStartDate;
345 protected javax.swing.JLabel lbAvailable;
346 protected javax.swing.JLabel lbDeployDescription;
347 protected javax.swing.JLabel lbDeployTerms;

```

```

348     protected javax.swing.JLabel lbDeployTitle;
349     protected javax.swing.JLabel lbDescription;
350     protected javax.swing.JLabel lbDuration;
351     protected javax.swing.JLabel lbEndDate;
352     protected javax.swing.JLabel lbImage;
353     protected javax.swing.JLabel lbImageScaled;
354     protected javax.swing.JLabel lbName;
355     protected javax.swing.JLabel lbStartDate;
356     protected javax.swing.JTextField tfDeployDescription;
357     protected javax.swing.JTextField tfDeployTerms;
358     protected javax.swing.JTextField tfDeployTitle;
359     protected javax.swing.JTextField tfDescription;
360     protected javax.swing.JTextField tfDuration;
361     protected javax.swing.JTextField tfName;
362     // End of variables declaration
363
364     private class DatePropertyChangeListener implements
PropertyChangeListener {
365
366         @Override
367         public void propertyChange(PropertyChangeEvent evt) {
368             String propName = evt.getPropertyName();
369             if ("date".equals(propName)) {
370
371                 Integer newDuration = refreshDuration();
372
373                 if(newDuration != null ) {
374                     if(newDuration !=
PackagesPanel.this.duration){
375                         // notify the listeners about duration
change
376
377                         Integer oldDuration =
PackagesPanel.this.duration;
378                         PackagesPanel.this.duration =
newDuration;
379
380                         PropertyChangeListener[] listeners =
PackagesPanel.this.getPropertyChangeListeners();
381                         if(listeners != null){
382                             for(PropertyChangeListener listener :
listeners){
383                                 PropertyChangeEvent propEvt = new
PropertyChangeEvent(PackagesPanel.this, DURATION_PROPERTY,
oldDuration, newDuration);
384                                 listener.propertyChange(propEvt);
385                             }
386                         }
387                     }
388                 }
389             }
390         }
391     }
392
393     private Integer refreshDuration() {
394         Date start = dcStartDate.getDate();
395         Date end = dcEndDate.getDate();
396
397         if (start == null) {
398             tfDuration.setText("Please set the start date");
399             return null;

```

```

400     }
401     if (end == null) {
402         tfDuration.setText("Please set the end date");
403         return null;
404     }
405
406     cal.setTime(start);
407     int startDay = cal.get(Calendar.DAY_OF_YEAR);
408
409     cal.setTime(end);
410     int endDay = cal.get(Calendar.DAY_OF_YEAR);
411
412     if(((startDay + 1) - endDay) > 0 ){
413         tfDuration.setText("Please set the end date after the
start date");
414         return null;
415     }
416
417     Integer duration = endDay - startDay;
418
419     tfDuration.setText(duration + " days" );
420
421     return duration;
422 }
423 }
424 }
425 }
426

```

7.1.6 PackageViewer

Εδώ αναπτύσσετε το τμήμα όπου θα παρουσιάζεται η εικόνα του πακέτου με την ημερομηνία έναρξης και λήξης του του ταξιδιού, την διάρκεια που θα έχει, καθώς και την γενική περιγραφή του.

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  /*
7  * PackageDetailViewer.java
8  *
9  * Created on 2009.01.20., 12:19:34
10 */
11 package project1.gui;
12

```

```

13 import project1.data.PackageData;
14 import java.awt.Dimension;
15 import java.awt.Image;
16 import java.text.DateFormat;
17 import java.text.SimpleDateFormat;
18 import java.util.Calendar;
19 import javax.swing.ImageIcon;
20 import project1.util.Settings;
21
22 /**
23  *
24  * @author Harry
25  */
26 public class PackageViewer extends javax.swing.JPanel {
27
28
29     private static DateFormat df = new
SimpleDateFormat("dd/MM/yyyy");
30     /** Creates new form PackageDetailViewer */
31     public PackageViewer() {
32         initComponents();
33     }
34
35     public void setData(PackageData data) {
36         assert (data != null) : "The data can not be null";
37         lbTitle.setText(data.getTitle());
38         // sould scaled the imege or put the same size in
database!!!
39
40         if (data.getImage() != null) {
41             Dimension imgSize = Settings.getDetailImageSize();
42             Image image =
data.getImage().getScaledInstance(imgSize.width, imgSize.height,
Image.SCALE_DEFAULT);
43             ImageIcon icon = new ImageIcon(image);
44             lbIMG.setIcon(icon);
45         }
46
47
48         lbIMG.setText(null);
49         tpDesc.setText(data.getDesc());
50
51         tfStartDate.setText(df.format(data.getStartDate()));
52         tfEndDate.setText(df.format(data.getEndDate()));
53         tfDuration.setText(""+data.getDuration());
54
55
56     }
57
58     /** This method is called from within the constructor to
59     * initialize the form.
60     * WARNING: Do NOT modify this code. The content of this
method is
61     * always regenerated by the Form Editor.
62     */
63     @SuppressWarnings("unchecked")
64     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
65     private void initComponents() {
66
67         lbTitle = new javax.swing.JLabel();

```



```

68         lbIMG = new javax.swing.JLabel ();
69         lbStartDate = new javax.swing.JLabel ();
70         tfStartDate = new javax.swing.JTextField ();
71         lbEndDate = new javax.swing.JLabel ();
72         tfEndDate = new javax.swing.JTextField ();
73         lbDuration = new javax.swing.JLabel ();
74         tfDuration = new javax.swing.JTextField ();
75         scDesc = new javax.swing.JScrollPane ();
76         tpDesc = new javax.swing.JTextPane ();
77
78         lbTitle.setFont (new java.awt.Font ("Tahoma", 0, 18));
79
lbTitle.setHorizontalAlignment (javax.swing.SwingConstants.CENTER);
80         lbTitle.setText ("Title Here");
81         lbTitle.setName ("lbTitle"); // NOI18N
82
83
lbIMG.setHorizontalAlignment (javax.swing.SwingConstants.CENTER);
84         lbIMG.setText ("Image HERE");
85
lbIMG.setVerticalAlignment (javax.swing.SwingConstants.TOP);
86         lbIMG.setName ("lbIMG"); // NOI18N
87
88         lbStartDate.setText ("StartDate:");
89         lbStartDate.setName ("lbStartDate"); // NOI18N
90
91         tfStartDate.setEditable (false);
92         tfStartDate.setText ("start date");
93         tfStartDate.setName ("tfStartDate"); // NOI18N
94
95         lbEndDate.setText ("End date:");
96         lbEndDate.setName ("lbEndDate"); // NOI18N
97
98         tfEndDate.setEditable (false);
99         tfEndDate.setText ("end date");
100        tfEndDate.setName ("tfEndDate"); // NOI18N
101
102        lbDuration.setText ("Duration:");
103        lbDuration.setName ("lbDuration"); // NOI18N
104
105        tfDuration.setEditable (false);
106        tfDuration.setText ("duration");
107        tfDuration.setName ("tfDuration"); // NOI18N
108
109        scDesc.setName ("scDesc"); // NOI18N
110
111        tpDesc.setEditable (false);
112        tpDesc.setText ("Description Here");
113        tpDesc.setName ("tpDesc"); // NOI18N
114        scDesc.setViewportView (tpDesc);
115
116        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout (this);
117        this.setLayout (layout);
118        layout.setHorizontalGroup (
119
layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
120            .addGroup (layout.createSequentialGroup ()
121                .addContainerGap ()

```

```

122
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
123
        .addGroup(layout.createSequentialGroup()
124
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
125
.addGroup(layout.createSequentialGroup()
126
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
127
                .addComponent(lbDuration)
128
                .addComponent(lbEndDate))
129
                .addGap(8, 8, 8)
130
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
131
                .addComponent(tfDuration,
javax.swing.GroupLayout.PREFERRED_SIZE, 63,
javax.swing.GroupLayout.PREFERRED_SIZE)
132
                .addComponent(tfEndDate,
javax.swing.GroupLayout.PREFERRED_SIZE, 97,
javax.swing.GroupLayout.PREFERRED_SIZE))
133
                .addComponent(lbIMG,
javax.swing.GroupLayout.PREFERRED_SIZE, 140,
javax.swing.GroupLayout.PREFERRED_SIZE)
134
.addGroup(layout.createSequentialGroup()
135
                .addComponent(lbStartDate)
136
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
137
                .addComponent(tfStartDate,
javax.swing.GroupLayout.DEFAULT_SIZE, 97, Short.MAX_VALUE))
138
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
139
                .addComponent(scDesc,
javax.swing.GroupLayout.DEFAULT_SIZE, 282, Short.MAX_VALUE))
140
                .addComponent(lbTitle,
javax.swing.GroupLayout.DEFAULT_SIZE, 440, Short.MAX_VALUE))
141
                .addContainerGap()
142
        );
143
144
        layout.linkSize(javax.swing.SwingConstants.HORIZONTAL,
new java.awt.Component[] {tfEndDate, tfStartDate});
145
146
        layout.setVerticalGroup(
147
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
148
                .addGroup(layout.createSequentialGroup()
149
                .addComponent(lbTitle)
150
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
151
                .addGroup(layout.createSequentialGroup()
152
                .addGap(13, 13, 13)
153
                .addComponent(lbIMG,
javax.swing.GroupLayout.PREFERRED_SIZE, 183,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

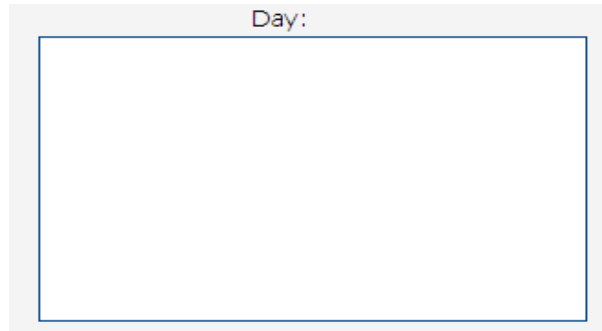
```

154 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
155 65, Short.MAX_VALUE)
156 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
157 t.BASELINE)
158 .addComponent(lbStartDate)
159 .addComponent(tfStartDate,
160 javax.swing.GroupLayout.PREFERRED_SIZE,
161 javax.swing.GroupLayout.DEFAULT_SIZE,
162 javax.swing.GroupLayout.PREFERRED_SIZE))
163 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
164 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
165 t.BASELINE)
166 .addComponent(lbEndDate)
167 .addComponent(tfEndDate,
168 javax.swing.GroupLayout.PREFERRED_SIZE,
169 javax.swing.GroupLayout.DEFAULT_SIZE,
170 javax.swing.GroupLayout.PREFERRED_SIZE))
171 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
172 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
173 t.BASELINE)
174 .addComponent(lbDuration)
175 .addComponent(tfDuration,
176 javax.swing.GroupLayout.PREFERRED_SIZE,
177 javax.swing.GroupLayout.DEFAULT_SIZE,
178 javax.swing.GroupLayout.PREFERRED_SIZE))
179 .addGroup(layout.createSequentialGroup())
180 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
181 .addComponent(scDesc,
182 javax.swing.GroupLayout.DEFAULT_SIZE, 327, Short.MAX_VALUE))
183 .addContainerGap()
184 );
185 }
186 }

```

7.1.7 DeployDayViewer

Εδώ θα υπάρχει η περιγραφή της κάθε μέρας του πακέτου.



```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 /*
7  * DeployDayViewer.java
8  *
9  * Created on 2009.01.20., 12:32:55
10 */
11
12 package project1.gui;
13
14 import project1.data.DeployDayData;
15
16 /**
17  *
18  * @author Harry
19  */
20 public class DeployDayViewer extends javax.swing.JPanel {
21
22     /** Creates new form DeployDayViewer */
23     public DeployDayViewer() {
24         initComponents();
25     }
26
27     public void setData(DeployDayData data){
28         assert (data != null): "The data can't be null";
29
30         lbTitle.setText("Day: "+data.getDay()+"
31 "+data.getTitle());
32         epDesc.setText(data.getDesc());
33     }
34
35     /** This method is called from within the constructor to
36      * initialize the form.
37      * WARNING: Do NOT modify this code. The content of this
38      * method is
39      * always regenerated by the Form Editor.
40      */
41     @SuppressWarnings("unchecked")
42     // <editor-fold defaultstate="collapsed" desc="Generated
43     Code">
44     private void initComponents() {
45
46         lbTitle = new javax.swing.JLabel();
47         scDesc = new javax.swing.JScrollPane();
48         epDesc = new javax.swing.JEditorPane();
```

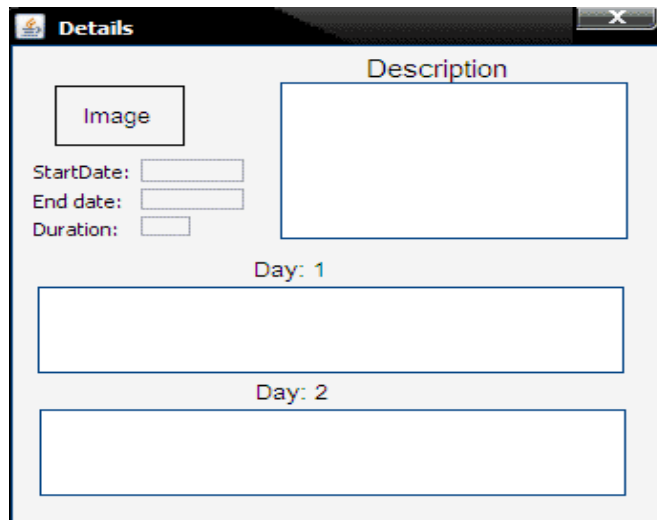
```

47
48         lbTitle.setFont(new java.awt.Font("Tahoma", 0, 14));
49
lbTitle.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
50         lbTitle.setText("Title HERE");
51         lbTitle.setName("lbTitle"); // NOI18N
52
53         scDesc.setName("scDesc"); // NOI18N
54
55         epDesc.setEditable(false);
56         epDesc.setName("epDesc"); // NOI18N
57         scDesc.setViewportViewView(epDesc);
58
59         javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
60         this.setLayout(layout);
61         layout.setHorizontalGroup(
62
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
63             .addGroup(layout.createSequentialGroup()
64                 .addGap(10, 10, 10)
65                 .addComponent(lbTitle,
javax.swing.GroupLayout.DEFAULT_SIZE, 437, Short.MAX_VALUE)
66                 .addGap(10, 10, 10)
67                 .addComponent(scDesc,
javax.swing.GroupLayout.DEFAULT_SIZE, 457, Short.MAX_VALUE)
68             );
69         layout.setVerticalGroup(
70
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
71             .addGroup(layout.createSequentialGroup()
72                 .addComponent(lbTitle)
73
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
74                 .addComponent(scDesc,
javax.swing.GroupLayout.DEFAULT_SIZE, 302, Short.MAX_VALUE)
75             );
76     } // </editor-fold>
77
78
79     // Variables declaration - do not modify
80     protected javax.swing.JEditorPane epDesc;
81     protected javax.swing.JLabel lbTitle;
82     protected javax.swing.JScrollPane scDesc;
83     // End of variables declaration
84
85 }
86

```

7.1.8 DetailViewer

Εδώ αναπτύσσουμε το γραφικό περιβάλλον της διεπαφής όπου θα φαίνεται η συνολική περιγραφή του πακέτου. Δηλαδή, η εικόνα του με την ημερομηνία έναρξης και λήξης του του ταξιδιού, την διάρκεια που θα έχει, την γενική περιγραφή του καθώς και όλες οι ημέρες του πακέτου, η κάθε μια με την περιγραφή της.



```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  /*
7  * DetailViewer.java
8  *
9  * Created on 2009.01.20., 12:15:10
10 */
11 package project1.gui;
12
13 import project1.data.DeployDayData;
14 import project1.data.PackageDetailsData;
15
16
17 /**
18  *
19  * @author Harry
20  */
21 public class DetailViewer extends javax.swing.JPanel {
22
23     /** Creates new form DetailViewer */
24     public DetailViewer() {
25         initComponents();
26     }
27
28     public void setData(PackageDetailsData data) {
29         assert (data != null) : "The data can't be null";
30
31         removeAll();
32
33
34         PackageViewer packageViewer = new PackageViewer();
35         packageViewer.setData(data.getPackageData());
36
37         this.add(packageViewer);
38
39         for (DeployDayData day: data.getVDays()) {
40             DeployDayViewer dayViewer = new DeployDayViewer();
41             dayViewer.setData(day);
42
43             this.add(dayViewer);

```

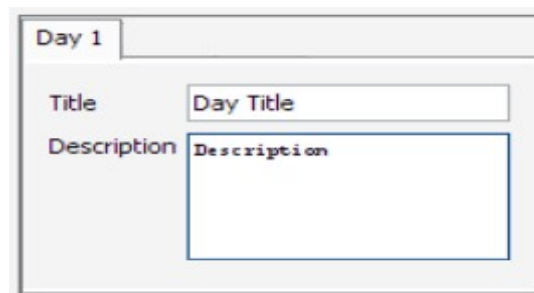
```

44
45     }
46 }
47
48 /** This method is called from within the constructor to
49  * initialize the form.
50  * WARNING: Do NOT modify this code. The content of this
method is
51  * always regenerated by the Form Editor.
52  */
53 @SuppressWarnings("unchecked")
54 // <editor-fold defaultstate="collapsed" desc="Generated
Code">
55 private void initComponents() {
56
57     setLayout(new javax.swing.BoxLayout(this,
javax.swing.BoxLayout.Y_AXIS));
58 }// </editor-fold>
59 // Variables declaration - do not modify
60 // End of variables declaration
61 }
62

```

7.1.9 DayPanel

Εδώ αναπτύσσουμε το Panel του κάθε Tab, το οποίο περιέχει την περιγραφή της κάθε μέρας στο παράθυρο του AddPanel.



```

1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 /*
7  * DayPanel.java
8  *
9  * Created on 2009.01.21., 20:36:57
10 */
11
12 package project1.gui;
13
14
15 /**
16  *
17  * @author Harry
18  */
19 public class DayPanel extends javax.swing.JPanel {
20
21     /** Creates new form DayPanel */

```

```

22     public DayPanel() {
23         initComponents();
24     }
25
26     public String getDescription() {
27         return taDescription.getText();
28     }
29
30     public void setDescription(String description) {
31         taDescription.setText(description);
32     }
33
34     public String getTitle() {
35         return tfTitle.getText();
36     }
37
38     public void setTitle(String title) {
39         tfTitle.setText(title);
40     }
41
42     public void clear(){
43         taDescription.setText("");
44         tfTitle.setText("");
45     }
46
47
48     /** This method is called from within the constructor to
49     * initialize the form.
50     * WARNING: Do NOT modify this code. The content of this
method is
51     * always regenerated by the Form Editor.
52     */
53     @SuppressWarnings("unchecked")
54     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
55     private void initComponents() {
56
57         lbTitle = new javax.swing.JLabel();
58         tfTitle = new javax.swing.JTextField();
59         lbDescription = new javax.swing.JLabel();
60         scDescription = new javax.swing.JScrollPane();
61         taDescription = new javax.swing.JTextArea();
62
63         lbTitle.setText("Title");
64         lbTitle.setName("lbTitle"); // NOI18N
65
66         tfTitle.setText("Day Title");
67         tfTitle.setName("tfTitle"); // NOI18N
68
69         lbDescription.setText("Description");
70         lbDescription.setName("lbDescription"); // NOI18N
71
72         scDescription.setName("scDescription"); // NOI18N
73
74         taDescription.setColumns(20);
75         taDescription.setRows(5);
76         taDescription.setText("Description");
77         taDescription.setName("taDescription"); // NOI18N
78         scDescription.setViewportViewView(taDescription);
79

```



```

80         javax.swing.GroupLayout layout = new
javax.swing.GroupLayout (this);
81         this.setLayout (layout);
82         layout.setHorizontalGroup (
83
layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
84             .addGroup (layout.createSequentialGroup ())
85                 .addContainerGap ()
86
.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
t.LEADING)
87                 .addComponent (lbDescription)
88                 .addComponent (lbTitle))
89
.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
90
.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
t.LEADING)
91                 .addComponent (tfTitle,
javax.swing.GroupLayout.DEFAULT_SIZE, 376, Short.MAX_VALUE)
92                 .addComponent (scDescription,
javax.swing.GroupLayout.DEFAULT_SIZE, 376, Short.MAX_VALUE))
93                 .addContainerGap ())
94         );
95         layout.setVerticalGroup (
96
layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
97             .addGroup (layout.createSequentialGroup ())
98                 .addContainerGap ()
99
.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
t.BASELINE)
100                 .addComponent (lbTitle)
101                 .addComponent (tfTitle,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
102
.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
103
.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignmen
t.LEADING)
104                 .addComponent (lbDescription)
105                 .addComponent (scDescription,
javax.swing.GroupLayout.DEFAULT_SIZE, 365, Short.MAX_VALUE))
106                 .addContainerGap ())
107         );
108     } // </editor-fold>
109
110
111     // Variables declaration - do not modify
112     protected javax.swing.JLabel lbDescription;
113     protected javax.swing.JLabel lbTitle;
114     protected javax.swing.JScrollPane scDescription;
115     protected javax.swing.JTextArea taDescription;
116     protected javax.swing.JTextField tfTitle;
117     // End of variables declaration
118
119 }
120

```

7.1.10 NoIconTreeCellRenderer

Σβήνουμε το εικονίδιο «φακελάκι» από την δενδρική δομή που χρησιμοποιούμε στις «Δραστηριότητες».

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package project1.gui;
6
7 import java.awt.Component;
8 import javax.swing.JLabel;
9 import javax.swing.JTree;
10 import javax.swing.tree.DefaultTreeCellRenderer;
11
12 /**
13  *
14  * @author Harry
15  */
16 public class NoIconTreeCellRenderer extends
DefaultTreeCellRenderer {
17
18     @Override
19     public Component getTreeCellRendererComponent(JTree tree,
Object value,
20         boolean sel,
21         boolean expanded,
22         boolean leaf, int row,
23         boolean hasFocus) {
24         JLabel lb = (JLabel)
super.getTreeCellRendererComponent(tree, value, sel, expanded, leaf,
row, hasFocus);
25         lb.setIcon(null);
26         return lb;
27     }
28 }
29
```

7.1.11 DeployDayData

Καταχωρούνται τα στοιχεία της κάθε ημέρας του πακέτου (ID, τίτλος, περιγραφή).

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package project1.data;
6
7 /**
8  * Deploy date data collection
9  * @author Harry
10  */
11 public class DeployDayData {
12
13     private int id;
14     private int packID;
15     private int day;
```

```

16     private String title;
17     private String desc;
18
19     public int getDay() {
20         return day;
21     }
22
23     public void setDay(int day) {
24         this.day = day;
25     }
26
27     public String getDesc() {
28         return desc;
29     }
30
31     public void setDesc(String desc) {
32         this.desc = desc;
33     }
34
35     public int getId() {
36         return id;
37     }
38
39     public void setId(int id) {
40         this.id = id;
41     }
42
43     public int getPackID() {
44         return packID;
45     }
46
47     public void setPackID(int packID) {
48         this.packID = packID;
49     }
50
51     public String getTitle() {
52         return title;
53     }
54
55     public void setTitle(String title) {
56         this.title = title;
57     }
58 }
59

```

7.1.12 PackageData

Καταχωρούνται τα γενικά στοιχεία του πακέτου (ID πακέτου, ID εικόνας, ημ/νια έναρξης, ημ/νια λήξης, τίτλος, περιγραφή, διάρκεια).

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package project1.data;
6
7  import java.awt.image.BufferedImage;
8  import java.sql.Timestamp;
9
10 /**

```

```

11  *
12  * @author Harry
13  */
14  public class PackageData {
15
16      private int ID;
17      private int imgID;
18      private String title;
19      private String desc;
20      private java.sql.Timestamp startDate;
21      private java.sql.Timestamp endDate;
22      private int duration;
23      private BufferedImage image = null;
24
25      public int getID() {
26          return ID;
27      }
28
29      public void setID(int ID) {
30          this.ID = ID;
31      }
32
33      public String getDesc() {
34          return desc;
35      }
36
37      public void setDesc(String desc) {
38          this.desc = desc;
39      }
40
41      public int getDuration() {
42          return duration;
43      }
44
45      public void setDuration(int duration) {
46          this.duration = duration;
47      }
48
49      public Timestamp getEndDate() {
50          return endDate;
51      }
52
53      public void setEndDate(Timestamp endDate) {
54          this.endDate = endDate;
55      }
56
57      public BufferedImage getImage() {
58          return image;
59      }
60
61      public void setImage(BufferedImage image) {
62          this.image = image;
63      }
64
65      public int getImgID() {
66          return imgID;
67      }
68
69      public void setImgID(int imgID) {
70          this.imgID = imgID;
71      }

```

```

72
73     public Timestamp getStartDate() {
74         return startDate;
75     }
76
77     public void setStartDate(Timestamp startDate) {
78         this.startDate = startDate;
79     }
80
81     public String getTitle() {
82         return title;
83     }
84
85     public void setTitle(String title) {
86         this.title = title;
87     }
88 }
89

```

7.1.13 PackageDetailsData

Καταχωρούνται συνολικά όλα τα στοιχεία του πακέτου, δηλαδή τα γενικά στοιχεία του μαζί με τα στοιχεία της κάθε μέρας ξεχωριστά.

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package project1.data;
6
7  import project1.util.DBConnection;
8  import java.io.File;
9  import java.io.FileOutputStream;
10 import java.io.IOException;
11 import java.io.InputStream;
12 import java.sql.Connection;
13 import java.sql.ResultSet;
14 import java.sql.SQLException;
15 import java.sql.Statement;
16 import java.util.Vector;
17 import javax.imageio.ImageIO;
18
19 /**
20 * Class to hold data from one package
21 * @author Harry
22 */
23 public class PackageDetailsData {
24
25     private PackageData packageData;
26     private Vector<DeployDayData> vDays;
27
28     public PackageDetailsData(final String packageID) throws
SQLException, IOException {
29         // load data from database:
30
31         PackageData packageData = new PackageData();
32
33         Connection con =
DBConnection.getDBConnection().createConnection();
34         Statement stPackage = con.createStatement();

```

```

35
36     // select the package data:
37     String sqlPackage = "SELECT " +//
38         "Ima_id, Pac_deploy_title, Pac_deploy_descr,
Pac_start_date, Pac_end_date, Pac_duration " +//
39         "FROM packages " +//
40         "WHERE Pac_id = '" + packageID + "' LIMIT 1";
41     ResultSet rsPackage = stPackage.executeQuery(sqlPackage);
42
43     packageData.setID(Integer.parseInt(packageID));//
NumberFormatException..
44
45     if (rsPackage.next()) {// we have found the package basic
data
46         packageData.setImgID(rsPackage.getInt("Ima_id"));
47
packageData.setTitle(rsPackage.getString("Pac_deploy_title"));
48
packageData.setDesc(rsPackage.getString("Pac_deploy_descr"));
49
packageData.setStartDate(rsPackage.getTimestamp("Pac_start_date"));
50
packageData.setEndDate(rsPackage.getTimestamp("Pac_end_date"));
51
packageData.setDuration(rsPackage.getInt("Pac_duration"));
52
53
54
55         rsPackage.close();
56         stPackage.close();
57         // load the image:
58         Statement stImg = con.createStatement();
59         String sqlImg = "SELECT Ima_data FROM images WHERE
Ima_id = '" + packageData.getImgID() + "' LIMIT 1";
60         ResultSet rsIMG = stImg.executeQuery(sqlImg);
61         if (rsIMG.next()) {// I have found the image data
62
63             // create a temp file: IOException
64             File fileImg = File.createTempFile("Image_" +
packageData.getImgID(), null);
65
66             //save blob as an image:
67             FileOutputStream fos = new
FileOutputStream(fileImg);
68
69             InputStream is = rsIMG.getBinaryStream(1);// the
first param of the result set as steam
70             byte[] buf = new byte[3000];
71             int read = 0;
72             while ((read = is.read(buf)) > 0) {
73                 fos.write(buf, 0, read);
74             }
75             fos.close();
76             is.close();
77             // image on the temp file...
78
79             packageData.setImage(ImageIO.read(fileImg));
80
81         }// after image read
82         rsIMG.close();
83         stImg.close();

```

```

84
85     }// after package data reads
86     rsPackage.close();
87     stPackage.close();
88     //data held in packageData, get the days data:
89
90     Statement stDays = con.createStatement();
91     String sqlDays = "SELECT " +
92         "Dd_id, Dd_day, Dd_title, Dd_descr " +
93         "FROM deploy_days " +
94         "WHERE Pac_id = '" + packageData.getID() + "'
ORDER BY Dd_day ASC";
95     ResultSet rsDays = stDays.executeQuery(sqlDays);
96
97     Vector<DeployDayData> vDays = new
Vector<DeployDayData>();
98
99     while (rsDays.next()) {
100         DeployDayData day = new DeployDayData();
101
102         // set the data:
103         day.setId(rsDays.getInt("Dd_id"));
104         day.setPackID(packageData.getID());
105         day.setDay(rsDays.getInt("Dd_day"));
106         day.setTitle(rsDays.getString("Dd_title"));
107         day.setDesc(rsDays.getString("Dd_descr"));
108
109         // add to the result collector:
110         vDays.add(day);
111     }
112     rsDays.close();
113     stDays.close();
114
115 //close the db connection.
116     con.close();
117
118     // No Exception happent: store the collected data at this
reference:
119     this.packageData = packageData;
120     this.vDays = vDays;
121
122 }
123
124 public PackageData getPackageData() {
125     return packageData;
126 }
127
128 public Vector<DeployDayData> getVDays() {
129     return vDays;
130 }
131 }
132

```

7.1.14 CategoryNode

Καταχωρούνται τα στοιχεία του κάθε κόμβου. (Σχετίζεται με τον σχεδιασμό της δενδρικής μας δομής που χρησιμοποιούμε στις «Δραστηριότητες».)

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package project1.data;
6
7  import javax.swing.tree.DefaultMutableTreeNode;
8
9  /**
10 *
11 * @author Harry
12 */
13 public class CategoryNode
14     extends DefaultMutableTreeNode {
15
16     private int id;
17     private String name;
18     private int parentID;
19
20     public CategoryNode(){
21         this(-1, "", -1);
22     }
23
24     public CategoryNode(int id, String name, int parentID) {
25         super(name);
26         this.id = id;
27         this.name = name;
28         this.parentID = parentID;
29     }
30
31     public int getId() {
32         return id;
33     }
34
35     public void setId(int id) {
36         this.id = id;
37     }
38
39     public String getName() {
40         return name;
41     }
42
43     public void setName(String name) {
44         this.name = name;
45     }
46
47     public int getParentID() {
48         return parentID;
49     }
50
51     public void setParentID(int parentID) {
52         this.parentID = parentID;
53     }
54
55     public String toString(){
56         return name;
57     }
58 }
59

```


7.1.15 Settings

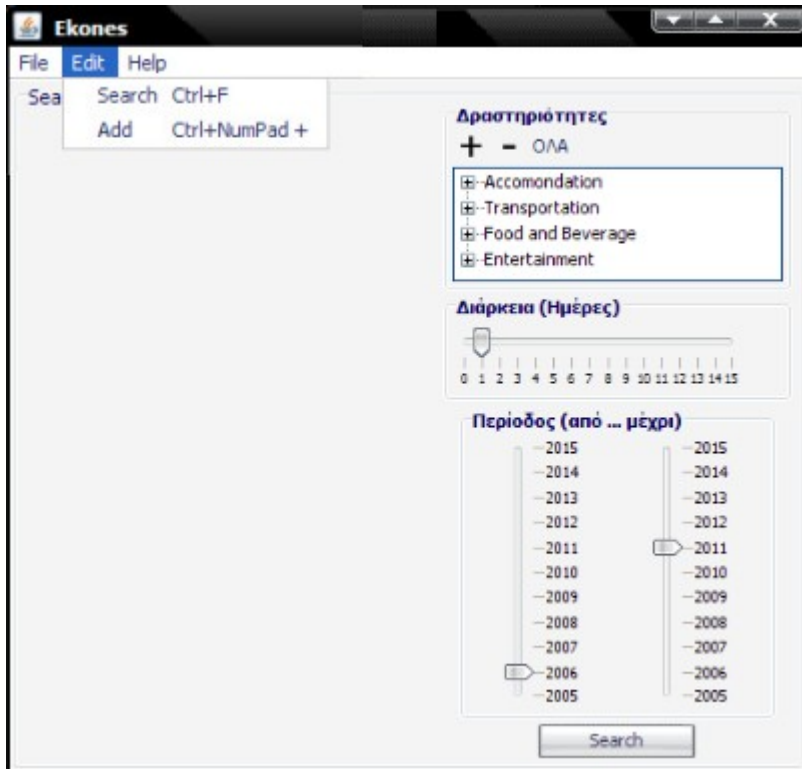
Αλλάζω τις διαστάσεις της εικόνας που κάνω Browse κάθε φορά, ώστε να έχει ένα συγκεκριμένο μέγεθος στην εφαρμογή μου.

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 package project1.util;
7
8 import java.awt.Dimension;
9
10 /**
11  *
12  * Singleton class to hold application settings
13  * @author Harry
14  */
15 public class Settings {
16
17
18     private Settings(){
19     }
20
21
22     //-- Singleton ---
23
24     private static Dimension detailImageSize = new
Dimension(150,150);
25
26
27
28     public static Dimension getDetailImageSize() {
29         return detailImageSize;
30     }
31
32     public static void setDetailImageSize(Dimension
detailImageSize) {
33         Settings.detailImageSize = detailImageSize;
34     }
35 }
36
```

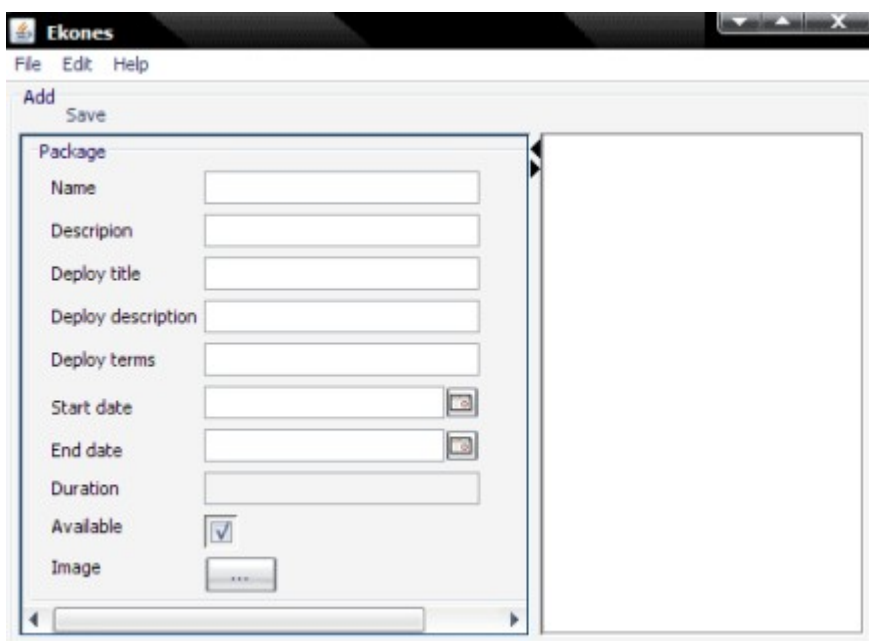
7.2 Πρακτικό Παράδειγμα Της Εφαρμογής

Παρακάτω παραθέτουμε μέσω Screenshots την πρόσθήκη ενός πακέτου στην βάση μας.

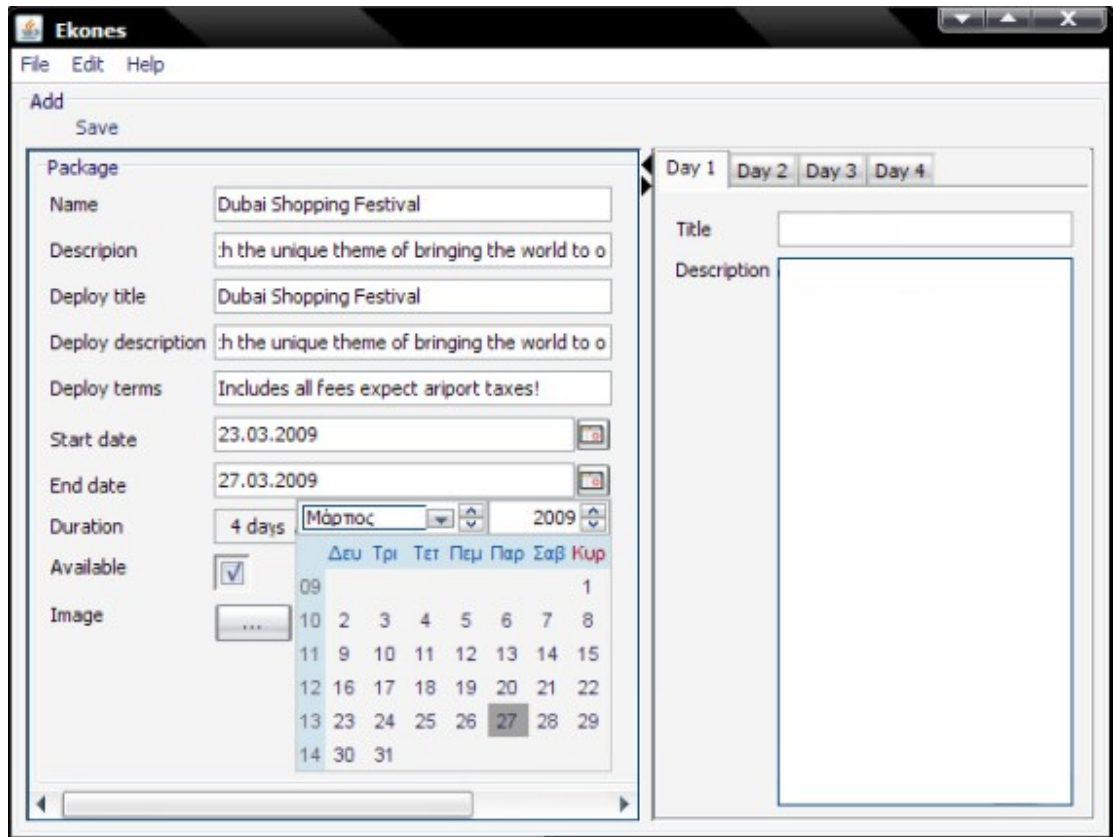
Εδώ έχουμε το κύριο παράθυρο της εφαρμογής μας.



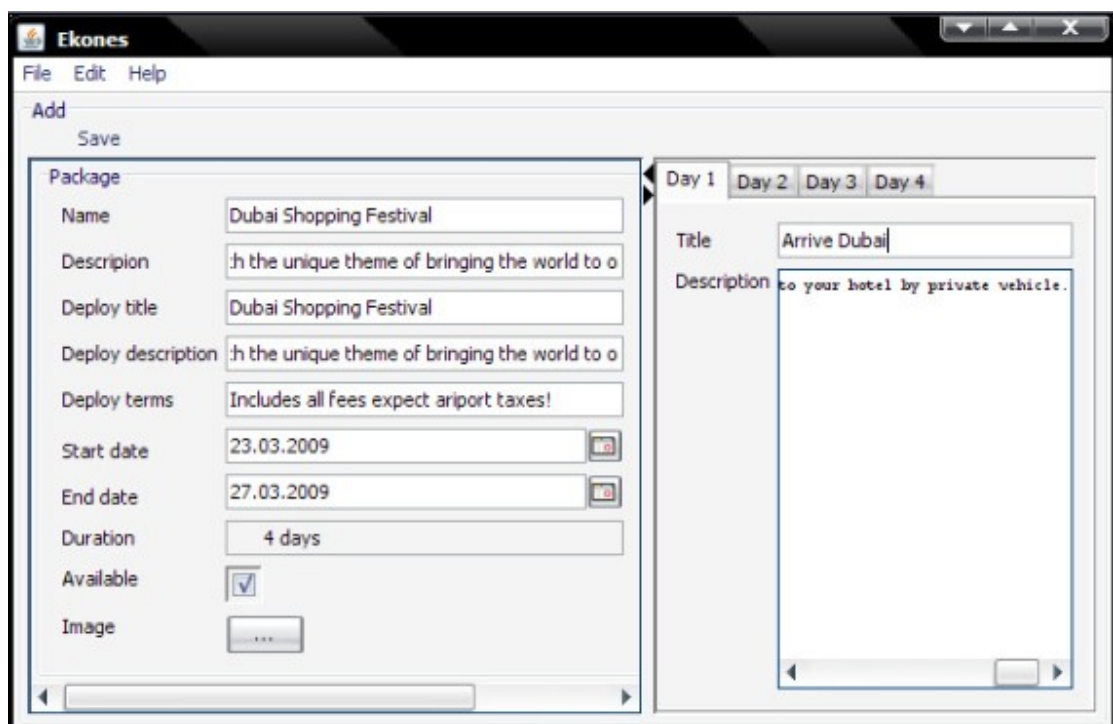
Στο Μενού επιλέγουμε Edit → Add και ανοίγει το παρακάτω παράθυρο όπου θα δώσουμε τα στοιχεία του πακέτου.



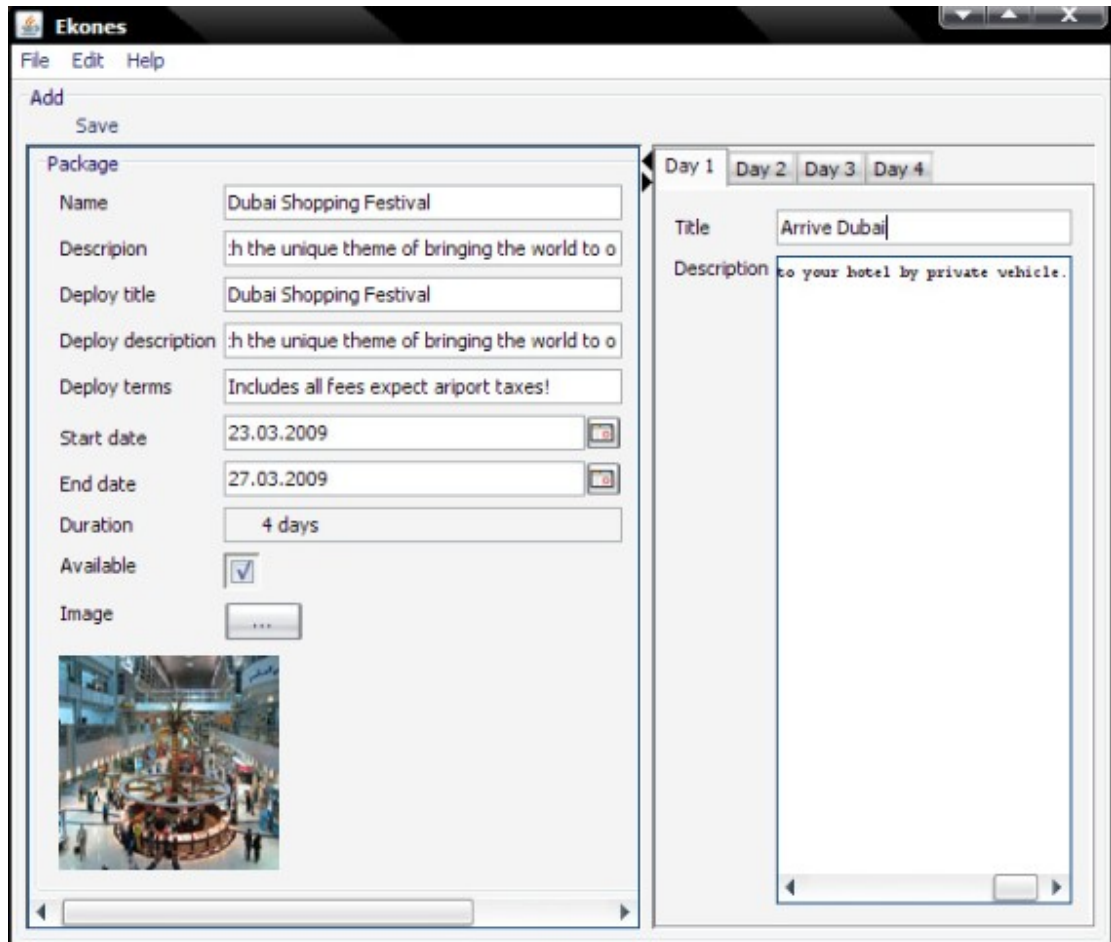
Μόλις επιλέξω και τις ημερομηνίες έναρξης και λήξης, υπολογίζεται η διάρκεια του πακέτου και δημιουργούνται αυτόματα στο δεξιό μέρος του παραθύρου καρτέλες για την περιγραφή της κάθε ημέρας.



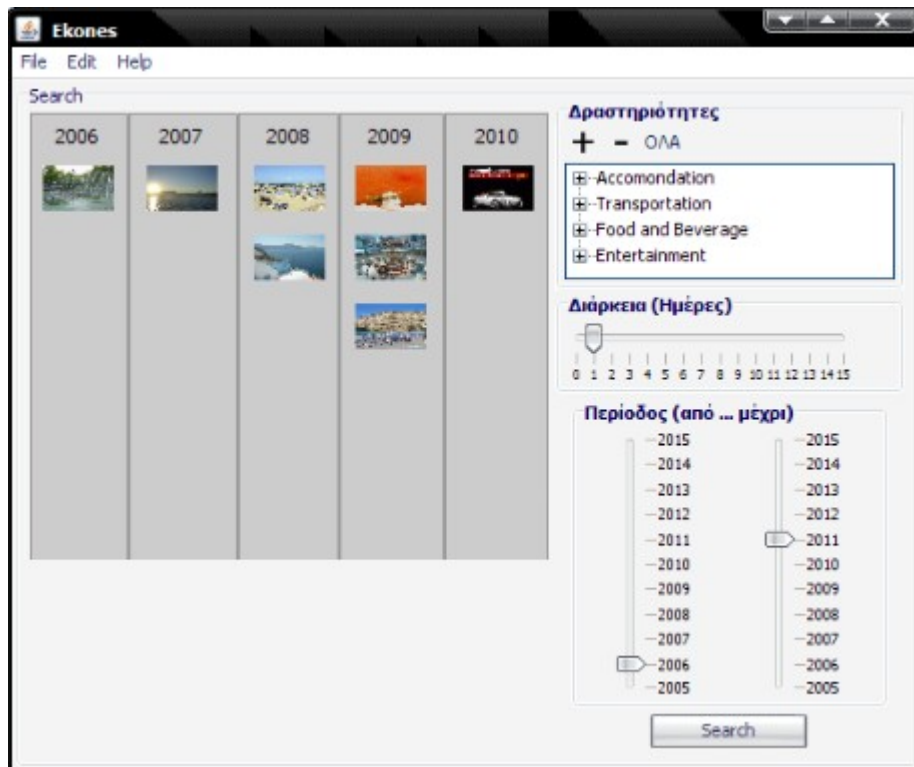
Συμπληρώνω τον τίτλο και την περιγραφή της πρώτης ημέρας στην πρώτη καρτέλα και ομοίως συνεχίζω και για τις υπόλοιπες καρτέλες – ημέρες.



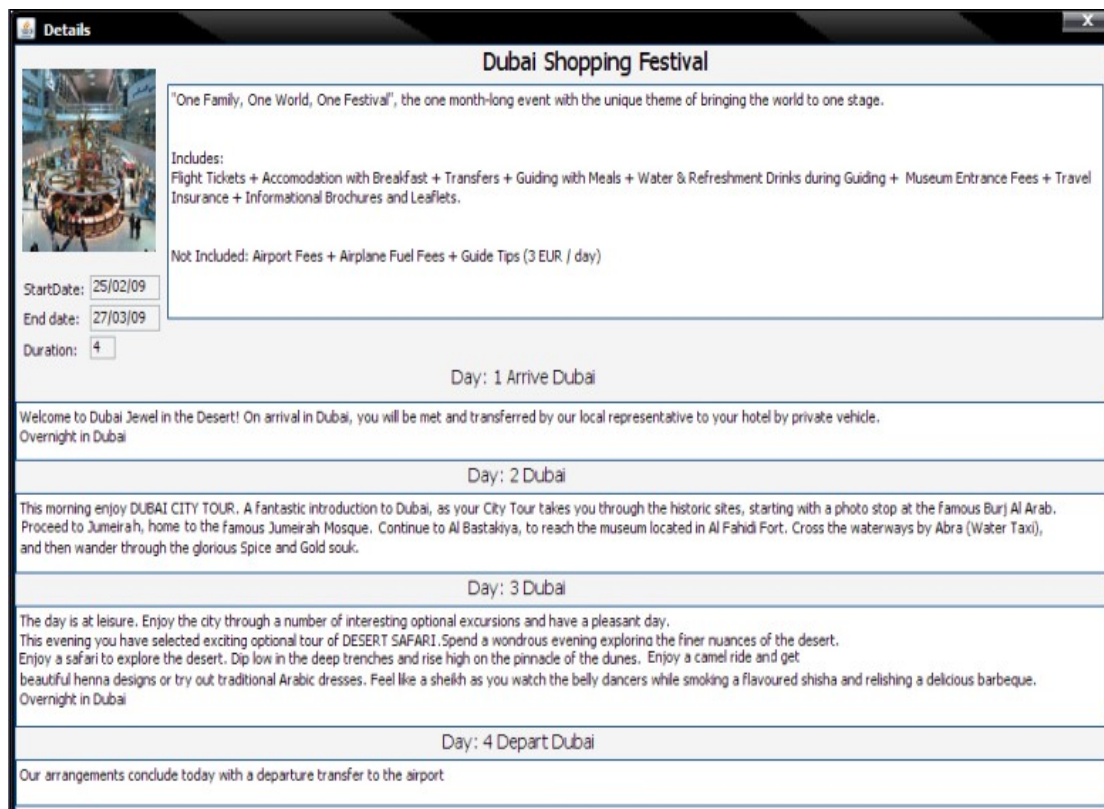
Τέλος, επιλέγω μια εικόνα για το πακέτο μου κάνοντάς την upload με την βοήθεια του Button "..." δίπλα από το "Image".



Επιλέγοντας τώρα στο Μενού, Edit → Search, μπορούμε να κάνουμε αναζήτηση των πακέτων βάση των κριτηρίων που θα θέσουμε. Τα πακέτα της αναζήτησης προβάλλονται στο δεξί μέρος του παραθύρου, χρονολογικά ταξινομημένα και σε μορφή ετήσιου ημερολογίου.



Επιλέγω το πακέτο που με ενδιαφέρει και «κλικάροντας» πάνω σε αυτό, μπορώ να δω αναλυτικά όλες τις πληροφορίες που το αφορούν.



Τέλος, επιλέγοντας στο Μενού, Help → About, εμφανίζεται και το παρακάτω παραθυρο.

