

2013



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
“ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ”

ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ ΣΧΕΔΙΑΣΤΗ ΤΑΞΙΔΙΟΥ



ΟΙ ΕΠΙΣΚΕΠΤΕΣ ΤΗΣ ΧΩΡΑΣ ΜΑΣ ΘΑ ΦΥΓΟΥΝ
ΜΕ ΜΙΑ ΓΛΥΚΙΑ ΑΝΑΜΝΗΣΗ

ΑΝΤΩΝΙΟΣ ΒΙΤΩΡΑΤΟΣ
ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΕΠΙΒΛΕΠΩΝ: ΜΑΡΙΝΟΣ ΚΑΒΟΥΡΑΣ
ΣΥΝΕΡΓΑΤΗΣ: ΤΟΜΑΗ ΕΛΕΝΗ
31/01/2013

ΠΡΟΛΟΓΟΣ

Το μεταπτυχιακό πρόγραμμα ολοκληρώνεται με την εκπόνηση της μεταπτυχιακής εργασίας. Η μεταπτυχιακή εργασία αποτελεί μια μοναδική εμπειρία έρευνας και μελέτης που ο κάθε σπουδαστής οφείλει να την λαμβάνει πολύ σοβαρά. Όμως η γνώση ποτέ δεν ολοκληρώνεται μόνο με το πέρας μια μεταπτυχιακής σπουδής και ειδικά η γνώση στο τομέα της Γεωπληροφορικής που αποτελεί ένα ιδιαίτερο αντικείμενο με πολλούς τομείς. Πρέπει ο καθένας μόνος του να έχει τη θελήσει να προχωρήσει ακόμα πιο πέρα και να γίνει ο καλύτερος στο τομέα του. Το μεταπτυχιακό απλά αποτέλεσε την αρχή μιας μακράς πορείας με πολλές δυσκολίες αλλά με ευχάριστες στάσεις στην δίχως τέλος λεωφόρο της τεχνολογίας .

Θα ήθελα να ευχαριστήσω τον κ Μαρίνο Κάβουρα για την εμπιστοσύνη που μου έδειξε από τη πρώτη στιγμή και την συμπαράσταση του σε όλη τη διάρκεια του μεταπτυχιακού. Επίσης να ευχαριστήσω την κ. Ελένη Τομαή για όλη την βοήθεια της στην εκπόνηση της συγκεκριμένης εργασίας με την παροχή προσωπικών σημειώσεων και προσωπικής επίβλεψης καθώς επίσης και τον κ.Βεσκούκη για τις πολύτιμες συμβουλές του σε θέματα πληροφορικής.

Τέλος, ιδιαίτερα θέλω να ευχαριστήσω την σύζυγο μου, Χριστίνα Γουδίρα για την υπομονή που επιδεικνύει και την ψυχολογική συμπαράσταση που μου παρέχει για την πραγματοποίηση και ολοκλήρωση των σπουδών μου , που όμως δεν τελειώνουν με την εκπόνηση αυτής της μεταπτυχιακής εργασίας, ενώ παράλληλα με στηρίζει στην δύσκολη και απαιτητική επαγγελματική μου σταδιοδρομία.

ΣΥΝΟΨΗ

Η Εφαρμογή που θα παρουσιαστεί αναπτύχθηκε στο πλαίσιο εκπόνησης της μεταπτυχιακής εργασίας για το Μεταπτυχιακό πρόγραμμα σπουδών “Γεωπληροφορική” του Εθνικού Μετσόβιου Πολυτεχνείου.

Με την παρούσα εργασία επιδιώκεται η υλοποίηση ενός Σχεδιαστή Τουρισμού (Tourism Planner), με σκοπό την παροχή εύκολων και γρήγορων προτάσεων τουριστικών επισκέψεων σε επισκέπτες πόλεων, που θα διατίθενται από το σύστημα, αναλόγως του σκοπού επισκέψεως και των προσωπικών επιθυμιών των επισκεπτών. Το σύστημα που θα σχεδιαστεί είναι απλό με τη χρήση ενός οδηγού συμπλήρωσης στοιχείων, με το πέρας του οποίου και την επιβεβαίωση των στοιχείων, θα αναζητούνται στη βάση τα στοιχεία και θα εμφανίζονται οι προτάσεις τουρισμού με τις αντίστοιχες πληροφορίες.

Η εφαρμογή αυτή θα αναλυθεί και θα σχεδιαστεί για να μπορεί εύκολα να προσαρμοστεί και σε τεχνολογίες κινητών τηλεφώνων όπως “Android” και “iOS” πέρα από το web interface που θα παρέχεται στο χρήστη. Η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί είναι η Java Enterprise Edition με τη χρήση Java Server Faces εμπλουτισμένη με την τεχνολογία Prime Faces. Το λογισμικό στηρίζεται στη φιλοσοφία του ανοιχτού κώδικα και γι’ αυτό θα χρησιμοποιηθούν λογισμικά ανοιχτού κώδικα για τη σχεδίαση.

Επιπλέον, με την παρούσα εργασία, επιδιώκεται και η ενημέρωση για την χρησιμοποίηση νέων τεχνολογιών όπως η Java Server Faces, στο διαδικτυακό προγραμματισμό που δεν περιλαμβάνεται σε κανένα προπτυχιακό μάθημα και λίγοι έχουν ασχοληθεί με αυτή.

Abstract

The project that will be presented is a Thesis for Postgraduate Studies Program in Geoinformatics of the National Technical University of Athens.

With this project is intended the implementation of a Tourism Planner which will be able to supply tourists with easy and quick tourist sightseeing suggestions for the available with data towns according to their personal intensions and willing. The system that will be designed will be simple with just the use of a wizard which will guide the tourist to fill the necessary information. After finishing the confirmation of the data the appropriate suggestions will be presented to the tourist.

The software will be analyzed and designed so as to be easily converted for the use to android and Mac phones more than the common web interface that will be available to the user. The program language that is intended to be used is Java Enterprise Edition with the use of Java Server Faces enriched with Prime Faces. The software is based on open source software development and for that reason all the software that will be used in the project will be open source.

Moreover another objective of this thesis is to inform the students about the new technologies like Java Server Faces, for internet programming which is not includes in the study program of the universities.

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	σελ 9
1.1	Σχεδιαστής Τουρισμού	σελ 9
1.2	Αντικείμενο της εργασίας	σελ 10
1.3	Σημαντικότητα δημιουργίας τουριστικής υπηρεσίας	σελ 11
1.4	Τρόπος ανάπτυξης της υπηρεσίας	σελ 12
1.5	Η Ελληνική πραγματικότητα	σελ 13
1.6	Σύνοψη της υπηρεσίας	σελ 21
2.	ΑΝΑΛΟΓΕΣ ΕΦΑΡΜΟΓΕΣ	σελ 22
2.1	Γενικά	σελ 22
2.2	Εφαρμογές στην Ελλάδα	σελ 23
2.3	Εφαρμογές στο Εξωτερικό	σελ 24
2.4	Η εφαρμογή Duffel	σελ 25
2.5	Εφαρμογή NileGuide	σελ 26
2.6	Εφαρμογή Triporama	σελ 27
2.7	Εφαρμογή TripIt και WanderFly	σελ 28
2.8	Εφαρμογή Plnrr	σελ 30
2.9	Συμπεράσματα	σελ 31
3.	ΥΠΗΡΕΣΙΕΣ ΔΙΑΔΙΚΤΥΟΥ	σελ 37
3.1	Γενικά	σελ 37
3.2	Λειτουργία των GIS WEB SERVICES	σελ 41
3.3	Αρχιτεκτονική SOA	σελ 42
3.4	Πρωτόκολλα WEB SERVICES	σελ 44
3.5	Πρωτόκολλο REST	σελ 48

4	WEB AND JAVASERVERFACES TECHNOLOGY	σελ 50
4.1	Γενικά	σελ 50
4.1.1	XHTML	σελ 50
4.1.2	URLs	σελ 51
4.1.3	Διαδικασία αιτήματος απάντησης	σελ 51
4.1.4	HTTP Headers	σελ 51
4.1.5	POST AND GET Αιτήματα	σελ 52
4.1.6	Client side caching	σελ 52
4.1.7	MULTI TIER Εφαρμογές	σελ 53
4.2	JAVA EE application model	σελ 55
4.3	Λειτουργία εφαρμογής με JSF	σελ 56
4.4	Δομή σελίδας JSF	σελ 57
4.5	Classes και JSF	σελ 58
4.5.1	Java Bean	σελ 58
4.5.2	Η κλάση TourismWizard	σελ 58
4.5.3	ManagedBean Annotation	σελ 58
4.6	Αρχιτεκτονική εφαρμογών JSF	σελ 59
4.6.1	MODEL-VIEW-CONTROLLER	σελ 59
4.7	Java EE και Java Server Faces	σελ 60
4.8	Java EE Components	σελ 62
4.8.1	Γενικά	σελ 62
4.8.2	Java EE εφαρμογές	σελ 62
4.8.3	Java EE Clients	σελ 63
4.8.3.1	Web Clients	σελ 63
4.8.3.2	Application Clients	σελ 63
4.8.4	Web Components	σελ 63
4.8.5	Java EE Application Assembly and Deployment	σελ 65
4.9	Συμπεράσματα	σελ 66
5.	Η ΑΝΑΠΤΥΞΗ ΤΩΝ ΟΝΤΟΛΟΓΙΩΝ	σελ 67
5.1	Γενικά στοιχεία για τις οντολογίες	σελ 67
5.1.1	Τι είναι οι οντολογίες	σελ 67
5.1.2	Τα συστατικά των οντολογιών και τα είδη τους	σελ 68
5.1.3	Αρχές ανάπτυξης οντολογιών	σελ 69
5.2.	Περιβάλλον Ανάπτυξης	σελ 70
5.3	Η γλώσσα OWL	σελ 70

6.	ΑΠΑΙΤΗΣΕΙΣ ΤΗΣ ΥΠΗΡΕΣΙΑ	σελ 72
6.1	Γενικά	σελ 72
6.2	Προσανατολισμός Υπηρεσίας	σελ 72
6.3	Προτεινόμενο σύστημα για υλοποίηση	σελ 73
6.4	Παροχή υπηρεσίας	σελ 74
7.	ΛΟΓΙΣΜΙΚΑ ΚΑΙ ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ	σελ 76
7.1	Γενικά	σελ 76
7.2	Java	σελ 76
7.3	NetBeans IDE	σελ 77
7.4	Apache subversion	σελ 77
7.5	GlassFish Server	σελ 77
7.6	Google Maps Api	σελ 78
7.6.1	Χαρακτηριστικά	σελ 79
7.7	Google Geocoding API	σελ 80
7.8	Google Directions API.	σελ 80
7.9	Gmap for JSF	σελ 81
7.10	AJAX	σελ 83
7.11	Google Places	σελ 83
7.11.1	Λεπτομέρειες τοποθεσιών	σελ 84
7.11.2	Places autocomplete	σελ 85
7.11.3	Distance Matrix Service	σελ 85
7.12	Java Persistence	σελ 86
7.12.1	Entities	σελ 87
7.13	Βάση MySQL Workbench	σελ 88
7.13.2	PostgreSQL	σελ 89
7.14	Unified Modeling Language (UML)	σελ 91
7.14.1	Διαγράμματα Κλάσεων	σελ 92
8.	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΤΗΣ ΥΠΗΡΕΣΙΑΣ	σελ 93
8.1	Αρχιτεκτονική και σχεδίαση Tourism Planner	σελ 93
8.2	Events	σελ 98
8.3	The entities project	σελ 99
8.4	The resource project	σελ 102
8.5	To Google Service Project	σελ 102

8.6	To Booking Service Project	σελ 103
8.7	To Tourism Wizard Project	σελ 103
8.8	Υλοποίηση αλγορίθμου SEARCH	σελ 107
8.9	Απαιτούμενες Εκδόσεις Λογισμικών	σελ 110
8.10	Υλοποίηση	σελ 111
9.	ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ	σελ 112
9.1	Άξονες Μοντελοποίησης	σελ 112
9.2	Συλλογή απαιτήσεων	σελ 112
9.3	Μοντελοποίηση	σελ 113
9.4	UML	σελ 113
9.5	Διάγραμμα Χρήσης	σελ 114
9.6	Διάγραμμα Κλάσεων	σελ 120
9.7	Αλληλεπίδραση Συστήματος	σελ 127
9.8	Διάγραμμα Ακολουθίας	σελ 128
9.9	Διάγραμμα Συνεργασίας	σελ 130
9.10	Ροή Ελέγχου	σελ 132
9.11	Διάγραμμα Ανάπτυξης	σελ 137
9.12	Οντολογίες με χρήση protégé	σελ 139
10.	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ	σελ 142
10.1	Συμπεράσματα	σελ 142
10.2	Ανάπτυξη Αλγορίθμου	σελ 144
10.3	Μελλοντικές επεκτάσεις	σελ 145

1

ΕΙΣΑΓΩΓΗ

1.1 Σχεδιαστής Τουρισμού

Η ανάπτυξη στο τομέα των υπηρεσιών διαδικτύου και ειδικότερα στο τομέα της πληροφορικής τα τελευταία χρόνια είναι ραγδαία. Ο συνδυασμός τεχνολογιών εντοπισμού θέσης, εύρεσης πλησιέστερων σημείων ενδιαφέροντος, χρήσης χαρτών, πλοήγησης με τη χρήση γράφων, χωρικών βάσεων δεδομένων και τουριστικών πληροφοριών, έχουν δώσει μια νέα διάσταση στην δημιουργία εφαρμογών τουριστικού προσανατολισμού.

Οι εφαρμογές αυτές συνδυάζουν τα στοιχεία που εισάγονται από τον ενδιαφερόμενο και του δίνουν μια πλήρη εικόνα για τα αξιοθέατα της πόλης ενδιαφέροντος σε συνδυασμό με τον τρόπο μετακίνησης, τον χρόνο, το κόστος κτλ. Υπάρχουν αρκετές εφαρμογές που αφορούν μόνο την υλοποίηση ταξιδιού με χρήση των μέσων μαζικής μεταφοράς ή ιδιωτικά μέσα ή πεζή προκειμένου να υλοποιηθεί μια διαδρομή από ένα σημείο στο άλλο με τη χρήση διαφόρων αλγορίθμων όπως Dijkstra, A star κλπ.

Αξίζει να σημειωθεί ότι για να μπορέσουμε να μιλάμε για μια ολοκληρωμένη εφαρμογή θα πρέπει να επιδιώκουμε την ενσωμάτωση διαφόρων παρεχόμενων εφαρμογών και των δεδομένων τους σε μια κοινή και διαλειτουργική υπηρεσία όπου διάφορες βάσεις δεδομένων από διάφορα συστήματα και εφαρμογές θα χρησιμοποιούνται από ένα απλό web interface φιλικό προς το χρήστη σε μια υπηρεσία διαδικτύου εύκολης προσβάσεως και χρήσεως.

Το πιο σημαντικό στην σχεδίαση ενός προγράμματος είναι τα δεδομένα. Χωρίς την κατάλληλη υποδομή δεδομένων δεν θα είναι εύκολη η υλοποίηση καμιάς εφαρμογής ενώ η απαίτηση για ατομική συλλογή δεδομένων θα ήταν σίγουρα χρονοβόρα.

Δυστυχώς, στην Ελλάδα δεν έχει φτάσει ακόμα, στο επιθυμητό επίπεδο η συλλογή των δεδομένων που θα ήταν απαραίτητη για τη δημιουργία εφαρμογών τουρισμού με δυνατότητα παροχής οδηγιών και πληροφοριών για όλες τις πόλεις και όλα τα αξιοθέατα. Ευχάριστο είναι ότι γίνονται αρκετές προσπάθειες με τη δημιουργία εφαρμογών υψηλών δυνατοτήτων που άμεσα θα μπορέσουν να χρησιμοποιηθούν με την ύπαρξη των κατάλληλων δεδομένων και τη δημιουργία διαδικτυακής τουριστικής υπηρεσίας.

1.2 Αντικείμενο της εργασίας

Αντικείμενο της εργασίας είναι η ανάλυση και ο σχεδιασμός ενός σχεδιαστή τουρισμού με σκοπό την παροχή ταξιδιωτικών προτάσεων σε τουρίστες για τις περιοχές που θα περιλαμβάνονται στην εφαρμογή και για τις οποίες θα υπάρχουν διαθέσιμες πληροφορίες και χάρτες. Η εφαρμογή αυτή απευθύνεται σε τουρίστες και όχι μόνο, που εύκολα και γρήγορα θα μπορούν να εισάγουν την πόλη προορισμού, τις ημερομηνίες άφιξης και αναχώρησης και την επιθυμητή δραστηριότητα για υλοποίηση. Με την ολοκλήρωση συμπλήρωσης των στοιχείων αυτών, θα εμφανίζονται διάφορες ταξιδιωτικές προτάσεις για επισκέψεις σε μουσεία, αρχαιολογικούς χώρους και άλλα αξιοθέατα της περιοχής που επέλεξαν με την παροχή διαθέσιμων πληροφοριών και οδηγιών.

Η επικοινωνία με το χρήστη θα γίνεται με ένα φιλικό προς το χρήστη περιβάλλον είτε μέσω υπολογιστή ή κινητό, με τεχνολογία android σε πολλές γλώσσες, δεδομένου ότι η χώρα μας αποτελεί πόλο έλξης τουριστών διαφόρων κρατών. Η εφαρμογή θα είναι πολύ απλή και θα γίνει χρήση ενός απλού οδηγού συμπλήρωσης στοιχείων. Είναι βέβαιο ότι δεν χρησιμοποιούν όλοι σύγχρονες εφαρμογές στην καθημερινότητά τους, και δεν είναι εξοικειωμένοι όλοι με την τεχνολογία παρά του γεγονότος ότι διανύουμε έναν αιώνα όπου η τεχνολογία αναπτύσσεται ραγδαία και ολοένα και περισσότερες υπηρεσίες πραγματοποιούνται δικτυακά.

Γενικά, ο χρήστης θέλει γρήγορα να πάρει μια λίστα από προτάσεις για τουριστικές επισκέψεις, να πληροφορηθεί πως θα πάει εκεί, τις ώρες λειτουργίας, τις τιμές των εισιτηρίων και πιθανά άλλα κοντινά σημεία ενδιαφέροντος. Χρήσιμο θα ήταν μέσω της ίδιας εφαρμογής να μπορεί να κάνει κράτηση ή αγορά των εισιτηρίων για τις επισκέψεις του ενώ και η παροχή αντίστοιχων προτάσεων διαμονής και φαγητού θα διευκόλυναν ακόμη πιο πολύ την παραμονή του.

1.3 Σημαντικότητα δημιουργίας τουριστικής υπηρεσίας

Όπως αναφέρθηκε και προηγουμένως, η τεχνολογία εξελίσσεται ραγδαία και η χρήση του διαδικτύου αυξάνεται ολοένα και περισσότερο. Συνέχεια εμφανίζονται νέες εφαρμογές που δημιουργούνται για την διευκόλυνση του κοινού. Στόχος του διαδικτύου και των διαδικτυακών εφαρμογών και υπηρεσιών είναι η παροχή στο ευρύ κοινό πληροφοριών εύκολα και γρήγορα με φιλικά προς το χρήστη προγράμματα και χωρίς την ανάγκη εξειδικευμένης γνώσης.

Η χρήση των κινητών τηλεφώνων και ειδικότερα των έξυπνων τηλεφώνων και tablet αυξάνεται ραγδαία ενώ ολοένα και περισσότερες εφαρμογές και υπηρεσίες ενσωματώνονται σε αυτά. Δέκτες εντοπισμού θέσης, σύνδεση με διαδίκτυο, παιχνίδια ,εφαρμογές εργασίας και άλλες υπηρεσίες έχουν δώσει μια νέα μορφή στην τεχνολογία και οδήγησαν στην δημιουργία νέων γλωσσών προγραμματισμού και νέων τομέων έρευνας και ανάπτυξης.

Στην Ελλάδα, ο τουρισμός είναι ένα πεδίο το οποίο θα πρέπει να δώσουμε μεγαλύτερη και ιδιαίτερη έμφαση. Η δημιουργία εφαρμογών και υπηρεσιών με στόχο την διευκόλυνση των τουριστών και όχι μόνο, πρέπει να αποτελεί κύριο σημείο έρευνας και ανάπτυξης. Η ανάπτυξη εφαρμογής που θα έδινε την δυνατότητα εισαγωγής στοιχείων από το χρήστη, όπως το μέρος που επιθυμεί να επισκεφτεί, ο σκοπός της επίσκεψης, τις προσωπικές του επιθυμίες , όπως αν θέλει να δει μουσεία ή αν τον ενδιαφέρουν αθλητικές δραστηριότητες κλπ, με την κατάλληλη υποδομή δεδομένων και επιστροφή αντίστοιχων αποτελεσμάτων με τουριστικές προτάσεις θα αποτελούσε μεγάλη διαφήμιση και ανάπτυξη για τη Χώρα.

Επιπρόσθετα, η ενσωμάτωση υφιστάμενων σχεδιαστών ταξιδιών που δίνουν τη δυνατότητα στους τουρίστες να επιλέγουν σημεία που θέλουν να επισκεφθούν και με τη χρήση της εφαρμογής να τους δίνεται η δυνατότητα να το κάνουν ακολουθώντας τη συντομότερη διαδρομή, ενώ ταυτόχρονα μπορούν να επιλέξουν και τρόπο μετάβασης όπως Ιδιωτικής Χρήσης οχήματα, πεζοπορία ή με τα Μέσα Μαζικής Μεταφοράς και ταυτόχρονα να βλέπουν το χρόνο μετάβασης, το κόστος και την απόσταση για την επιλεγμένη διαδρομή, αυξάνει την αποτελεσματικότητα της υπηρεσίας και την επισκεψιμότητά της.

Τέλος για να μπορούμε να μιλάμε για έναν ολοκληρωμένο σχεδιαστή τουρισμού θα πρέπει να ενσωματωθούν και υπηρεσίες ταξιδιωτικών γραφείων όπως εύρεσης ξενοδοχείων, κρατήσεις εισιτηρίων κλπ

1.4 Τρόπος ανάπτυξης της υπηρεσίας

Για την εφαρμογή έπρεπε να ακολουθηθούν κάποια βήματα για την σωστή ανάπτυξη του λογισμικού. Όπως και για κάθε εφαρμογή έτσι και για αυτή, η διαδικασία είναι αυτής της τεχνολογίας λογισμικού και φαίνεται στο παρακάτω Σχήμα 1.



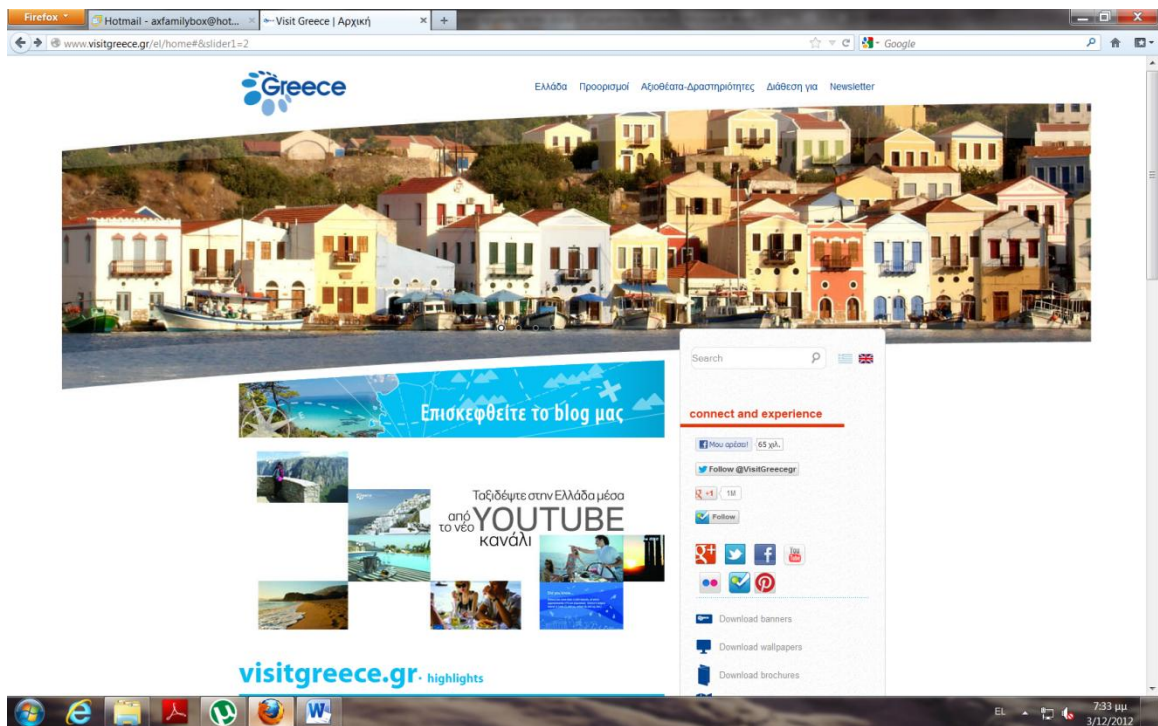
Σχήμα 1 Διαδικασία Ανάπτυξης

Αυτή η διαδικασία θα ακολουθηθεί εδώ η οποία και θα αναλυθεί παρακάτω.

1.5 Η Ελληνική Πραγματικότητα

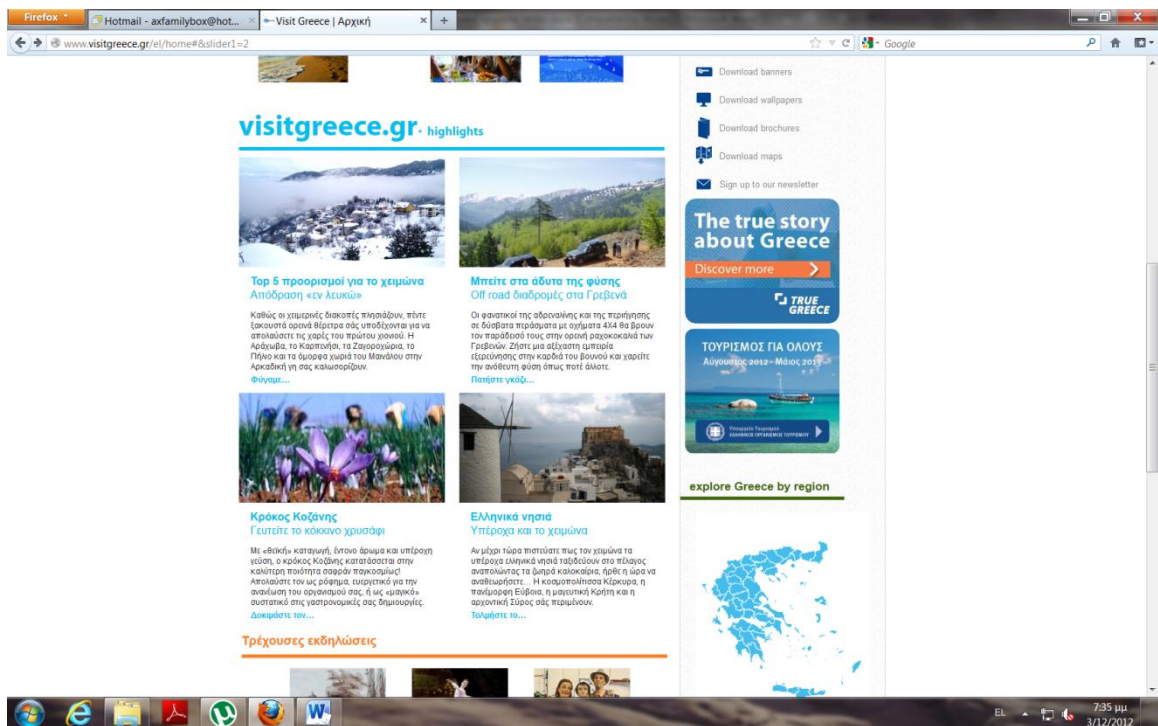
Όπως έχει ήδη αναφερθεί, η υπηρεσία που θα υλοποιηθεί είναι ένας σχεδιαστής ταξιδιού. Ο χρήστης, όταν εισέρχεται για πρώτη φορά, στην ιστοσελίδα δεν θα χρειάζεται κάποιο κωδικό για να εισέλθει στο σύστημα ούτε θα του ζητείται η δημιουργία κάποιου λογαριασμού εισόδου. Απλώς θα επιλέγει έναρξη και θα εισέρχεται στον οδηγό συμπλήρωσης των ταξιδιωτικών του προτιμήσεων. Θα του παρέχεται μια λίστα από διαθέσιμες πόλεις- περιοχές μια από τις οποίες θα επιλεγεί. Στο επόμενο στάδιο, θα του δίνεται ένα ημερολόγιο από το οποίο θα επιλέγει τις ημερομηνίες άφιξης και αναχώρησης. Στο επόμενο βήμα, θα γίνεται η επιλογή της δραστηριότητας που επιθυμεί. Αυτή η δραστηριότητα μπορεί να είναι επισκέψεις σε μουσεία, αρχαιολογικούς χώρους, πάρκα, αγορές και άλλα. Με το πέρας αυτού του σταδίου θα γίνεται η αναζήτηση των διαθέσιμων τουριστικών προτάσεων με τα στοιχεία που εισήγαγε ο χρήστης και θα του εμφανίζεται μια λίστα με τις προτάσεις και που βρίσκονται στο χάρτη καθώς και οι προτεινόμενες διαδρομές.

Η Ελλάδα στην υλοποίηση τέτοιων εφαρμογών έχει μείνει θα λέγαμε λίγο πίσω. Ο ΕΟΤ που αποτελεί και τον κυριότερο φορέα τουρισμού της χώρα μας έχει δημιουργήσει την ιστοσελίδα [visitgreece](http://visitgreece.gr) που σε καμία περίπτωση δεν αποτελεί την υλοποίηση της παραπάνω περιγραφής. Η ιστοσελίδα είναι απλής υλοποίησης χωρίς ιδιαίτερα χαρακτηριστικά όπως βλέπουμε και στην παρακάτω εικόνα



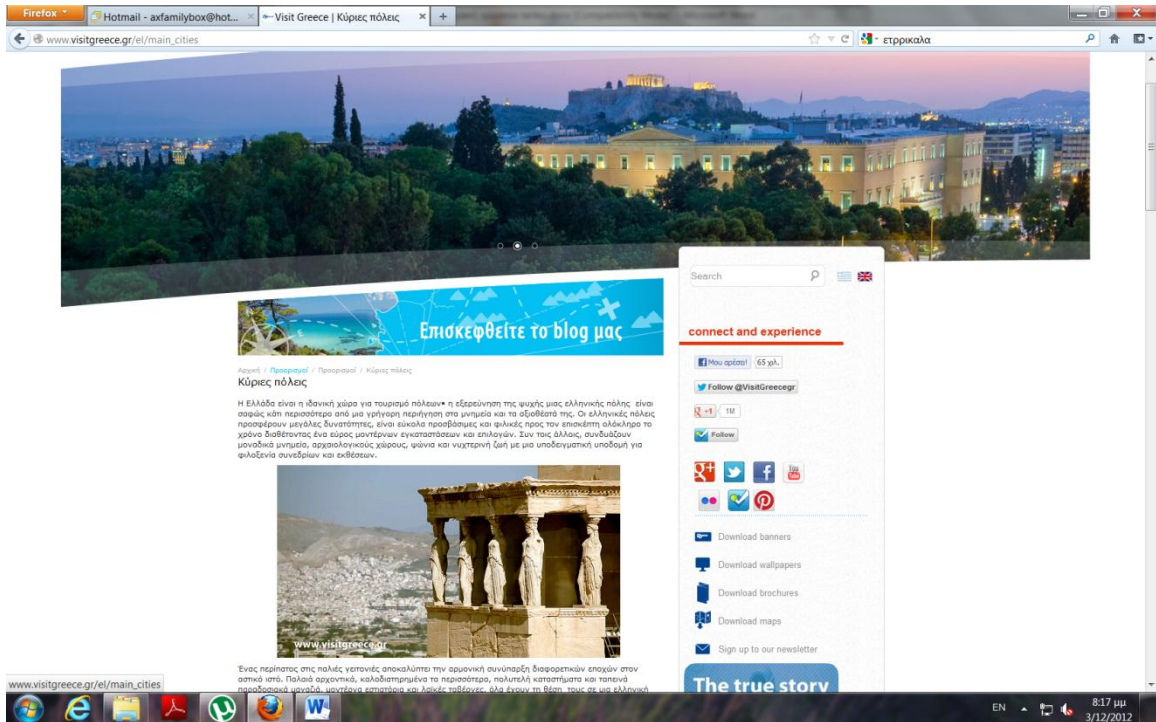
Εικόνα 1 Αποτελέσματα εφαρμογής [Πηγή:www.visitgreece.gr]

Οι επιλογές που δίνει είναι αρκετές όπως προορισμούς, αξιοθέατα και δραστηριότητες και ιστορικά στοιχεία. Δίνει και προτάσεις για κάποια μέρη που αξίζει κάποιος να επισκεφθεί και πληροφορίες για τρέχουσες εκδηλώσεις όμως αφορά όλη την Ελλάδα όπως βλέπουμε και στην αρχική οθόνη και όχι συγκεκριμένο προορισμό



Εικόνα 2 Αποτελέσματα εφαρμογής [Πηγή:www.visitgreece.gr]

Με την επιλογή προορισμού ερχόμαστε αντιμέτωποι με την διάθεση ιστορικών και άλλων γενικών πληροφοριών για τα νησιά, τις πόλεις και βλέπουμε κάποιες εικόνες. Διατίθενται γενικοί χάρτες και απλά όμορφες εικόνες.



Εικόνα 3 Αποτελέσματα εφαρμογής [Πηγή: www.visitgreece.gr]

Η ιστοσελίδα παρέχει γενικές πληροφορίες, μπορεί να αναφέρει επιγραμματικά τα αξιοθέατα και στοιχεία αυτών αλλά δεν αποτελεί σχεδιαστή ταξιδιού με προτάσεις που θα αφορούν ένα συγκεκριμένο μέρος, μια πόλη, με παροχή πληροφοριών που σχετίζονται με την επιθυμητή δραστηριότητα, τις ημερομηνίες άφιξης ώστε το σύστημα να προτείνει και ποια μνημεία είναι ανοιχτά τις συγκεκριμένες ημερομηνίες, χάρτη για εύρεση της τοποθεσίας και σχεδιασμό διαδρομής, ώστε ο επισκέπτης εύκολα να έχει ένα σημείο σαν αφετηρία και να ακολουθήσει μια διαδρομή με εκτιμώμενο χρόνο και μέρη που να επισκεφτεί με φωτογραφίες και πληροφορίες. Διαθέτει αρκετά link προς άλλες τοποθεσίες αλλά με μια πολύ γενική προσέγγιση.

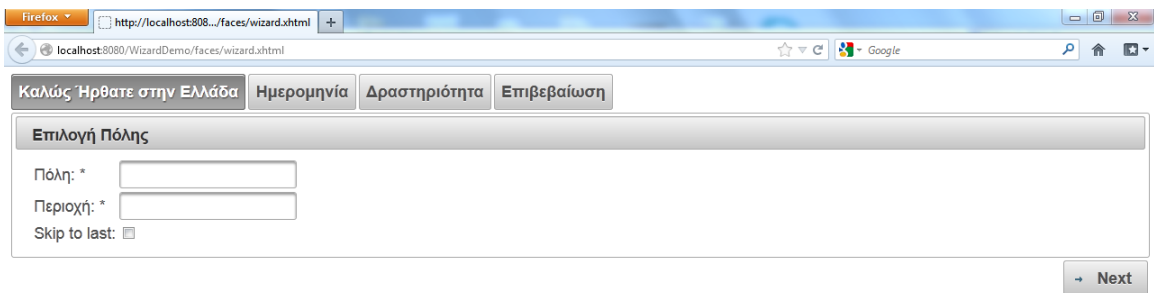
Παρόμοιες λειτουργίες ακολουθούν και τα υπόλοιπα διαθέσιμα sites τουρισμού που αποτελούν τους δικτυακούς τόπους περιοχών. Τέτοιες ιστοσελίδες δεν είναι εύκολες προς χρήση διότι ο χρήστης ούτε μπορεί να γνωρίζει για το ποιες περιοχές μιλάει ούτε θα μπει στη διαδικασία να αρχίζει να εξερευνεί όλο το site και να γράφει με στυλό και χαρτί που να πάει και τι να δει, ενώ για να πάρει πληροφορίες για αξιοθέατα θα πρέπει να χρησιμοποιεί άλλα link, με χρονοβόρες διαδικασίες αναζήτησης και χωρίς ουσιαστικό αποτέλεσμα. Ο χρήστης στην ουσία πρέπει να καταναλώσει αρκετό χρόνο να διαβάσει όλο το site, να δει ποια πόλη του αρέσει και να

αναζητήσει πάλι στο Google την πόλη και αν σταθεί τυχερός και υπάρχει ιστότοπος θα πάρει κάποιες πληροφορίες, αν όχι πάντα υπάρχει η βικιπαιδεία και τα forum.

Όμως επειδή αυτός ο τρόπος είναι αναζήτησης είναι πολύ μπακαλίστικος όπως θα τον χαρακτήριζε κάποιος και σίγουρα δεν αρμόζει σε μια σύγχρονη χώρα, εδώ έρχεται και ενσωματώνεται η δικιά μας υπηρεσία πλέον με σαφή οδηγό αναζήτησης που καθοδηγεί τον χρήστη τι πληροφορίες να εισάγει, να γίνεται αναζήτηση από μια εξολοκλήρου δικιά μας βάση δεδομένων πλήρως ενημερωμένη με στοιχεία, να εισάγει ημερομηνίες, πόλη ή και περιοχή ακόμα, την δραστηριότητα που θέλει, όπως για παράδειγμα μουσεία, διότι μπορεί μια μέρα να θέλει να δει μόνο τα μουσεία, ενώ την άλλη να επισκεφτεί αρχαιολογικούς χώρους, ενώ θα μπορεί να επιλέγει εύρεση όλων των προτάσεων ή των καλύτερων, που θα έχουν διαμορφωθεί ανάλογα των προτιμήσεων των χρηστών, με τελικό αποτέλεσμα την εμφάνιση όλων των προτάσεων πάνω στο χάρτη, με σχεδιασμένη την προτεινομένη διαδρομή και εικόνες των τοποθεσιών με τις αντίστοιχες πληροφορίες. Εύκολα, γρήγορα και πάνω απ'όλα συνολικά και συγκεντρωτικά, χωρίς την ανάγκη μετάβασης από ιστοσελίδα σε ιστοσελίδα.

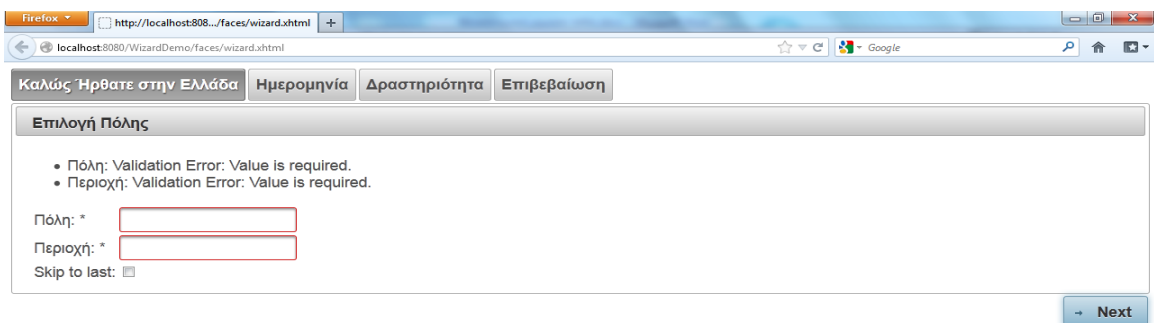
Ο οδηγός συμπλήρωσης μπορεί να υλοποιηθεί όπως παρακάτω, όπου τα στοιχεία εισάγονται συνολικά και στο τέλος αφού γίνει ανασκόπηση των επιλογών να δίνετε η εντολή για αναζήτηση των στοιχείων στη βάση, χωρίς να αλλάζουν σελίδες με αποτέλεσμα την πολύωρη αναμονή, αλλά ασύγχρονα, με βήματα. Σε κάθε βήμα ο χρήστης θα ρωτάτε και θα συμπληρώνει τα στοιχεία, στο τέλος θα παίρνει σαν απάντηση τον χάρτη με τις τοποθεσίες.

Στο πρώτο βήμα θα πραγματοποιείται η επιλογή της πόλης και της περιοχής



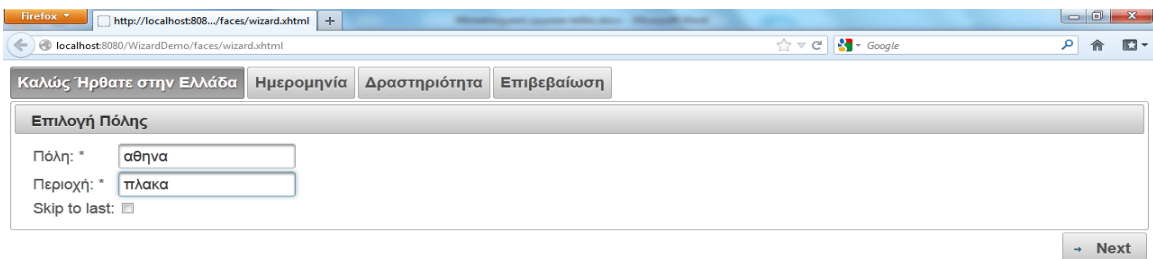
Εικόνα 4 Πρώτο βήμα οδηγού

Σε περίπτωση που ο χρήστης δεν τοποθετήσει στα κατάλληλα πεδία την πόλη και την περιοχή θα παίρνει το αντίστοιχο μήνυμα



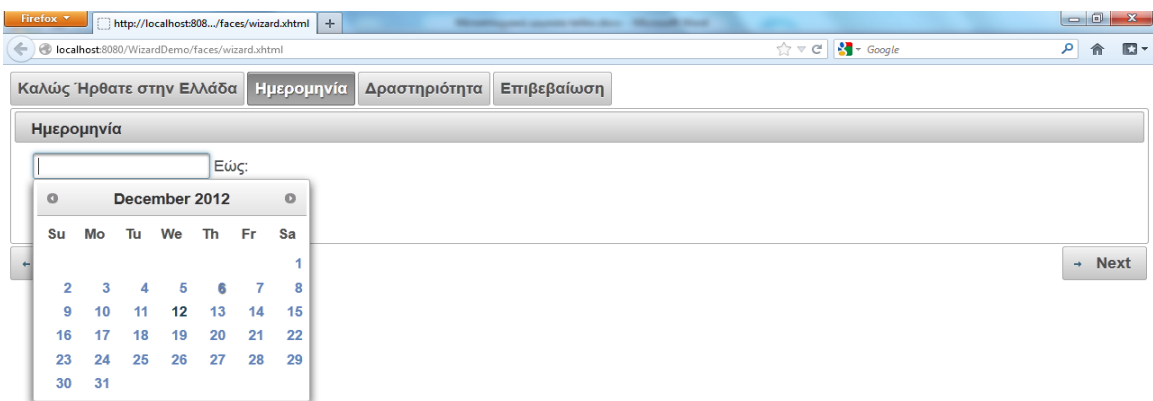
Εικόνα 5 Λανθασμένη εισαγωγή

Στη συνέχεια αφού συμπληρώσει σωστά τα στοιχεία θα πατάει next και θα πηγαίνει στο επόμενο βήμα

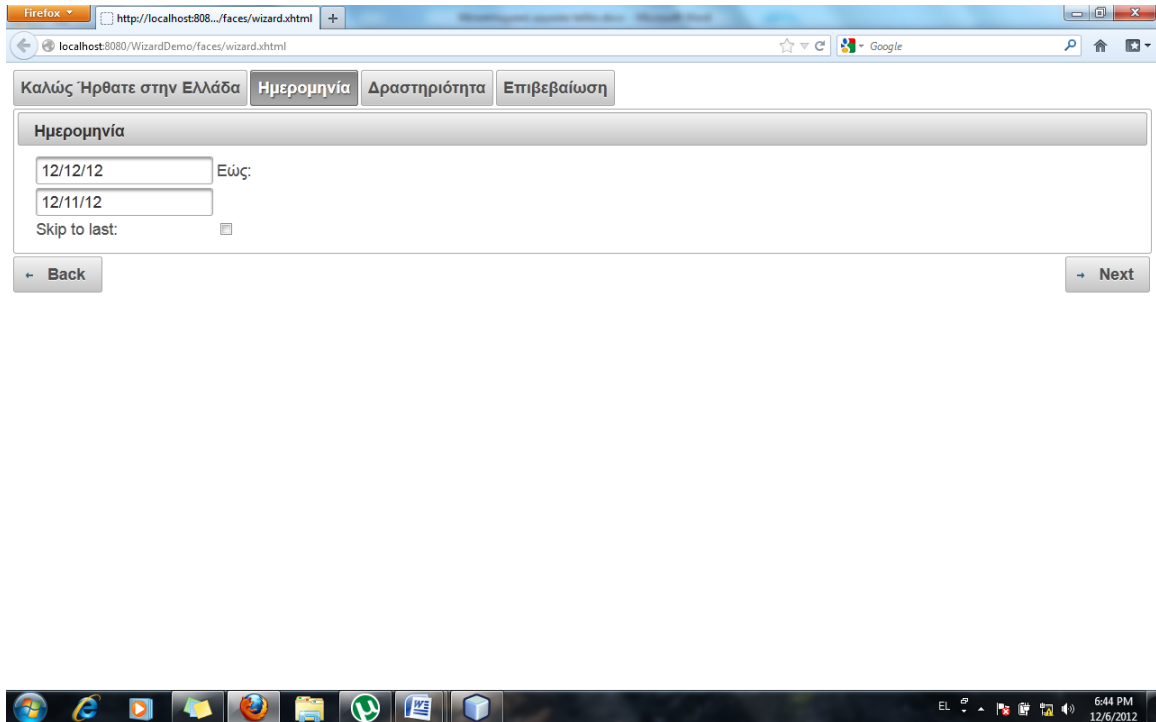


Εικόνα 6 Πρώτο βήμα οδηγού

Με το πάτημα πάνω στη περιοχή θα εμφανίζεται ένα ημερολόγιο προκειμένου να εισάγει τις ημερομηνίες άφιξης και αναχώρησης

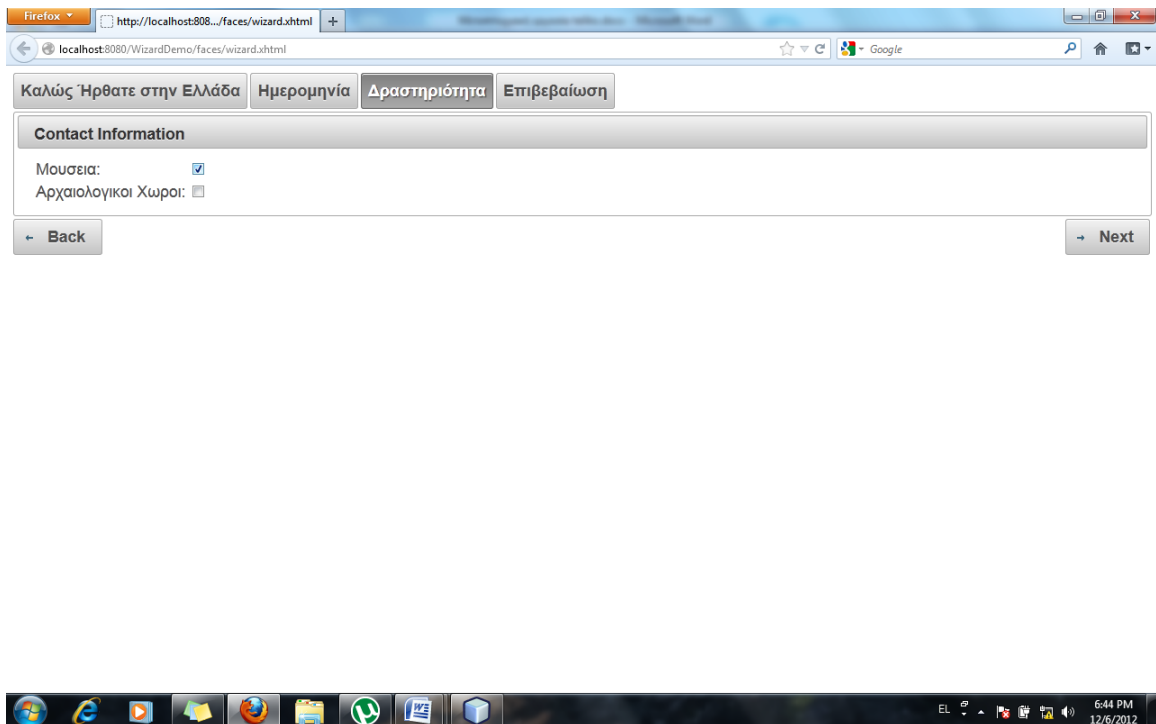


Εικόνα 7 Δεύτερο βήμα οδηγού



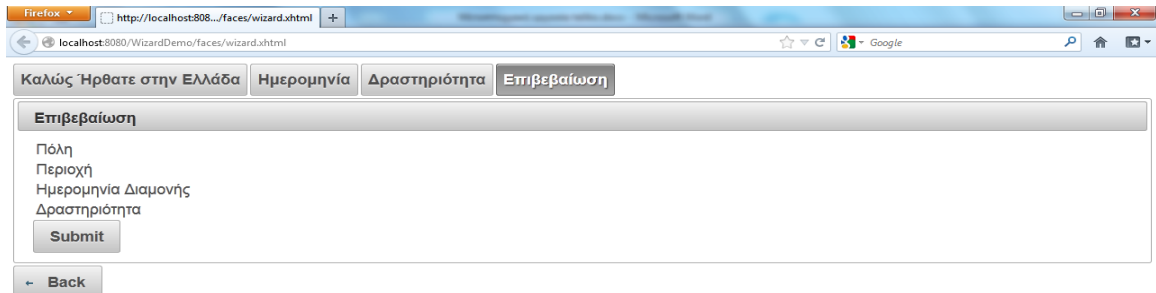
Εικόνα 8 Εισαγωγή ημερομηνιών

Θα επιλέγει την δραστηριότητα που επιθυμεί



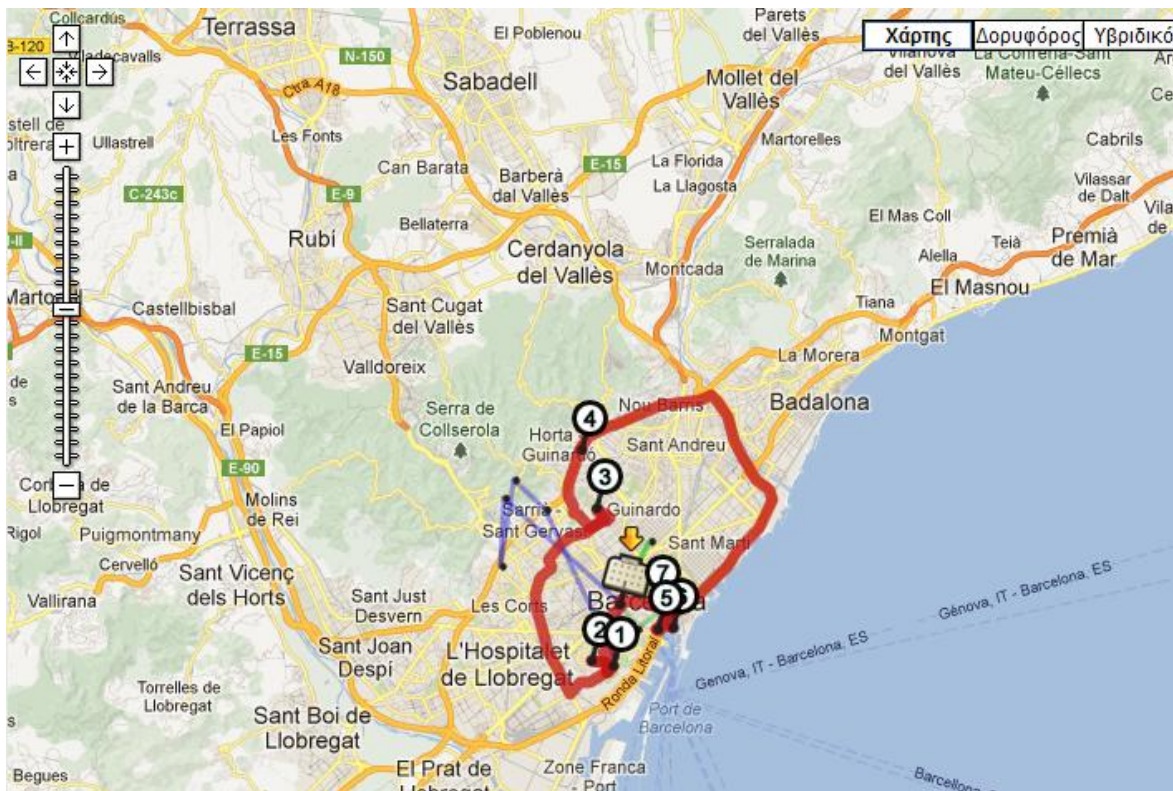
Εικόνα 9 Τρίτο βήμα οδηγού

Και τέλος θα γίνεται μια ανασκόπηση όλων των εισαγόμενων στοιχείων και θα πατάει το submit προκειμένου να πραγματοποιηθεί η αναζήτηση



Εικόνα 10 Τέταρτο βήμα οδηγού

Τα αποτελέσματα θα εμφανίζονται σε ένα χάρτη με markers όπως βλέπουμε και παρακάτω



Εικόνα 11 Εμφάνιση Αποτελεσμάτων

Ο οδηγός αυτός υλοποιήθηκε απλά για επίδειξη και σε καμία περίπτωση δεν είναι η τελική του μορφή. Λείπει η σύνδεση με τη βάση και την υπηρεσία της Google Map για τους χάρτες, ενώ εικόνες και χρώματα δεν έχουν ενσωματωθεί. Όμως ο κώδικας και η κλάση είναι έτοιμα και μπορούν να ενσωματωθούν σε οποιοδήποτε interface.

Η υπηρεσία θα πρέπει να δίνει την δυνατότητα στο χρήστη να μπορεί να συνδέεται με άλλες ιστοσελίδες που θα δίνουν την δυνατότητα κρατήσεων και αγοράς εισιτηρίων είτε κάποιας πολιτιστικής δραστηριότητας ή αεροπορικών, ακτοπλοϊκών και τρένων.

Επίσης θα πρέπει να υπάρχει και αντίστοιχος σύνδεσμος για κρατήσεις ξενοδοχείων ενώ η ενσωμάτωση και ενός σχεδιαστή ταξιδιού με τα μέσα μαζικής μεταφοράς θα ολοκλήρωνε την υπηρεσία.

1.6 Σύνοψη λογισμικού υλοποίησης της υπηρεσίας

Ο σχεδιαστής τουρισμού θα είναι μια διαδικτυακή υπηρεσία η οποία θα υλοποιηθεί με τεχνολογία Java Server Faces μαζί και με άλλα στοιχεία JavaEE6 όπως

- ❖ Java Server Faces με contexts and Dependency Injection για την χρήση της πλατφόρμας Java EE (CDI)
 - Ένα σύνολο από facelets σελίδες σε ένα πρότυπο το οποίο θα αποτελεί και το περιβάλλον του χρήστη της υπηρεσίας
 - CDI managed beans τα οποία σχετίζονται με τη λειτουργία κάθε σελίδας. Ο κώδικας δηλαδή για κάθε λειτουργία της σελίδας.
 - Ένα χάρτης με την ενσωμάτωση της υπηρεσίας της Google το Goggle Maps Api πάνω στον οποίο θα εμφανίζονται τα σημεία των προτάσεων και οι διαδρομές προς αυτά.
 - Οδηγό συμπλήρωσης με το δικό του bean και δυναμικά Tags με σύνδεση με τη βάση δεδομένων προκειμένου να εισάγονται τα στοιχεία και να γίνεται η αναζήτηση και εύρεση των προτάσεων με συνδυασμό των στοιχείων
 - Action listeners' οι οποίοι θα παίρνουν το id της επιλογής από κάθε στάδιο θα το αποθηκεύουν για το συγκεκριμένο session map ώστε να μπορεί να χρησιμοποιηθεί από το bean της επόμενης σελίδας.
- ❖ Enterprise beans:
- ❖ Το java Persistence API για την δημιουργία των entities από τη βάση δεδομένων

2

ΑΝΑΛΟΓΕΣ ΕΦΑΡΜΟΓΕΣ

2.1 Γενικά

Ο τουρισμός αποτελεί μια πολύ ισχυρή βιομηχανία για πολλές χώρες και ιδίως για εκείνες που έχουν καλό κλίμα και θάλασσα και πάνω από όλα ήλιο ,όπως έχει και χώρα μας. Η ανάπτυξη του διαδικτύου πρέπει να αξιοποιηθεί με τον καλύτερο δυνατό τρόπο ώστε να υπάρξει ανάπτυξη σε ένα οικονομικό πεδίο όπως αυτό του τουρισμού που μέχρι στιγμής δυστυχώς δεν έχει γίνει.

Σήμερα, ο τουρίστας είναι σε θέση να περιηγηθεί σε διάφορα μέρη του κόσμου και να έχει τη δυνατότητα να οργανώνει πλήρως το ταξίδι του από πριν βλέποντας τα μέρη ,τα ξενοδοχεία, τις υπηρεσίες μεταφοράς και άλλα πολλά πράγματα που θα κάνουν τις διακοπές του πιο ξεκούραστες και λιγότερο αγχωτικές. Αρκεί να σκεφτεί έναν προορισμό, να εισάγει την ονομασία του σε μια μηχανή αναζήτησης και θα βρεθεί μπροστά σε ένα πλήθος ιστότοπων που είτε προσφέρουν γενικές πληροφορίες για την κάθε περιοχή, είτε βοηθούν στην ανεύρεση καταλύματος, τρόπους μετάβασης και δραστηριοτήτων, είτε προσφέρουν εναλλακτικούς τρόπους τουρισμού, είτε όλα αυτά μαζί.

Το ζήτημα είναι, κατά πόσον όλο αυτό το πλήθος πληροφοριών μπορεί να αξιοποιηθεί ικανοποιητικά από έναν τουρίστα ο οποίος, συνήθως, δεν έχει προσωπική εμπειρία για τον εκάστοτε προορισμό ή δεν είναι και πολύ εξοικειωμένος με τη χρήση της τεχνολογίας . Σκοπός της τεχνολογίας είναι να διευκολύνει ουσιαστικά τον τουρίστα και να του παρέχει άμεσα χωρίς κόπο τις απαραίτητες πληροφορίες εύκολα , γρήγορα και πάνω από όλα σωστά. Στόχος είναι η εξοικονόμηση χρόνου και χρήματος με την παροχή σωστών και ασφαλών πληροφοριών, διότι πρέπει να έχουμε πάντα στο μυαλό μας ότι συνήθως, ο τουρίστας δεν γνωρίζει τη χώρα που επισκέπτεται.

Για αυτό το λόγο εξετάζονται εδώ τι είδους διαδικτυακές εφαρμογές έχουν δημιουργηθεί με στόχο την παροχή τέτοιων πληροφοριών. Η αναζήτηση των πληροφοριών γίνεται συνήθως με βάση την περιοχή γιατί και ο επισκέπτης θα κάνει την αναζήτησή του χρησιμοποιώντας το όνομα της περιοχής και μετά θα αρχίσει να ψάχνει για άλλες λεπτομέρειες.

2.2 Εφαρμογές στην Ελλάδα

Συνήθως, όταν πραγματοποιούμε αναζήτηση για κάποιο σχεδιαστή τουρισμού στην Ελλάδα αυτό που λαμβάνουμε ως απάντηση στο αίτημα μας είναι οι επίσημες διαδικτυακές πύλες, που είτε αποτελούν τμήμα της κεντρικής πύλης του εκάστοτε νομού ή δήμου, είτε αποτελούν ξεχωριστό ιστότοπο αφιερωμένο στον τουρισμό.

Υπάρχουν και διαδικτυακές πύλες που δεν λειτουργούν με ευθύνη της τοπικής αυτοδιοίκησης αλλά έχουν δημιουργηθεί από ιδιώτες που είχαν τη θέληση να δημιουργήσουν μια εφαρμογή με τη συλλογή προσωπικών πληροφοριών που άλλες φορές είναι αξιόπιστες και άλλες φορές όχι. Όμως οι εφαρμογές αυτές κυρίως εξυπηρετούν στην ανεύρεση καταλύματος ή εισιτηρίων όχι στην παρουσίαση προτάσεων επισκέψεων. Πολλές πόλεις στην Ελλάδα, έχουν ασχοληθεί μόνο με τη δημιουργία εφαρμογής για το δήμο τους όπου έχουν μια μίξη ιστορίας, καταλυμάτων, περιοχών ενδιαφέροντος και άλλες πληροφορίες γενικές, λίγο περίπλοκες για ένα απλό τουρίστα.

Οι περισσότερες διαδικτυακές εφαρμογές που υπάρχουν για περιοχές στην Ελλάδα δεν εμφανίζουν ιδιαίτερες διαφορές μεταξύ τους. Στις ιστοσελίδες αυτές παρουσιάζονται στοιχεία για τα βασικά θέματα που αναζητά κάποιος επισκέπτης χωρίς όμως με κάποιο τρόπο να δίνεται κάποιος προσανατολισμός στην ατομική εξυπηρέτηση και ικανοποίηση του επισκέπτη. Τα θέματα που προσφέρονται προς αναζήτηση είναι κατά κύριο λόγο η διαμονή, η μετάβαση στην περιοχή, το φαγητό, οι μετακινήσεις στην περιοχή, η διασκέδαση, οι δραστηριότητες, οι εκδηλώσεις χωρίς σαφή οργάνωση και ξεκάθαρη πρόταση στις προτιμήσεις του κάθε τουρίστα.

Ανάλογα, με τα χαρακτηριστικά κάθε τόπου, ο επισκέπτης πιθανόν, να έρθει αντιμέτωπος με πιο συγκεκριμένες επιλογές όπως μπορεί να είναι παραλίες, εναλλακτικός τουρισμός, έθιμα, παραδοσιακά προϊόντα, πανηγύρια. Ο επισκέπτης επίσης, συνήθως έχει τη δυνατότητα να δει μαγνητοσκοπημένα αποσπάσματα και φωτογραφίες της περιοχής και να ενημερωθεί για χρήσιμα τηλέφωνα και υπηρεσίες.

Παρόλα αυτά, διαπιστώνεται ότι ένα σύστημα με αυτή την δομή, δεν είναι λειτουργικό και όχι εύχρηστο. Πολλές πληροφορίες με διάφορες υπερσυνδέσεις και πολλά μενού ενώ υπάρχει και ένας χάρτης χωρίς διαδρομές και markers , χωρίς σαφής προτάσεις και πληροφορίες που ταιριάζουν στο προφίλ του χρήστη. Δυστυχώς στην Ελλάδα δεν υπάρχουν τέτοιες εφαρμογές αλλά ούτε και κατάλληλες υποδομές και βάσεις δεδομένων με στοιχεία όπως αρχαιολογικοί χώροι, πάρκα, αγορές και άλλες πληροφορίες.

2.3 Εφαρμογές στο Εξωτερικό

Η έρευνα για το εξωτερικό έγινε κυρίως σε εφαρμογές που αφορούν μεγάλες πόλεις. Παρατηρήθηκαν πολλές εφαρμογές με παρόμοια δομή και λειτουργίες με αυτών στην Ελλάδα.

Ο επισκέπτης στις περισσότερες περιπτώσεις κατακλύζεται από επιλογές των βασικών θεμάτων που ενδιαφέρουν τον ταξιδιώτη. Για να βρει τα σημεία ενδιαφέροντος που του ταιριάζουν πρέπει να περιηγηθεί σε πλήθος εγγραφών και να διαβάσει τις λεπτομέρειες που συνοδεύουν αυτές. Ουσιαστικά, δεν προσφέρονται ιδιαίτερες δυνατότητες για εξατομικευμένη αναζήτηση των χρηστών.

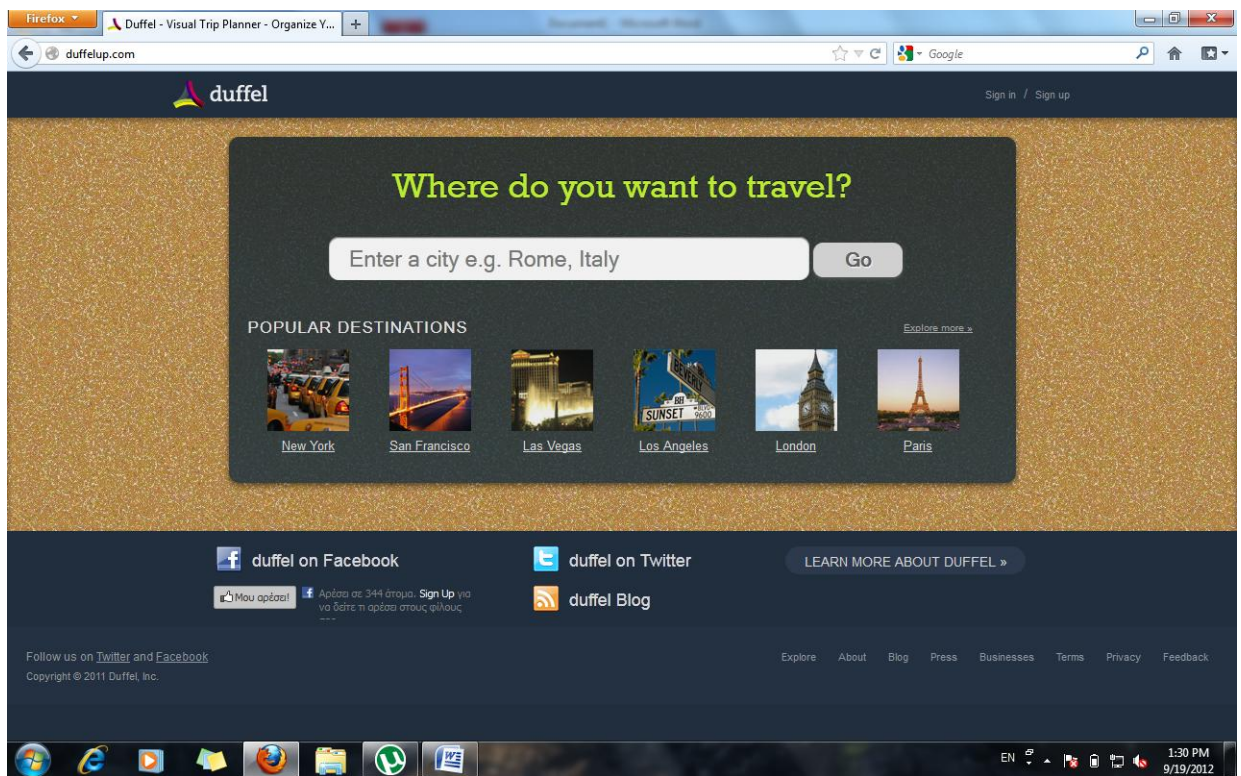
Πλην όμως, προσφάτως δημιουργήθηκαν και εφαρμογές που έχουν δημιουργηθεί είτε από φοιτητές ή από οργανισμούς που χρησιμοποιούν τη βάση δεδομένων που διαθέτουν σαν χώρα. Στην πρώτη περίπτωση οι φοιτητές από μόνοι τους αναζήτησαν και βρήκαν προτάσεις και στοιχεία για να ενημερώσουν τη βάση και να προσφέρουν αξιόλογα αποτελέσματα στους τουρίστες.

Βρήκαμε διάφορους τύπους από σχεδιαστές όπως θα δούμε και παρακάτω. Κάποιες εφαρμογές είναι εξαιρετικές και πολύ εύκολες στην χρήση. Το πιο σημαντικό κομμάτι είναι η ευκολία χρήσης .Ο τουρίστας θέλει κάτι γρήγορο εύκολο και πάνω από όλα οπτικό δηλαδή να βλέπει τα μέρη πως είναι και να νιώθει ότι είναι εκεί. Θα αναλυθούν κάποιες από αυτές και θα δοθεί ιδιαίτερη έμφαση σε αυτή που μου έκανε περισσότερη εντύπωση και θεωρώ πιο εύκολη για τον τουρίστα. Κάποιες εφαρμογές ακολουθούν έναν οδηγό και επιτρέπουν στον χρήστη να επιλέξει αυτό που θέλει και τότε θέλει.

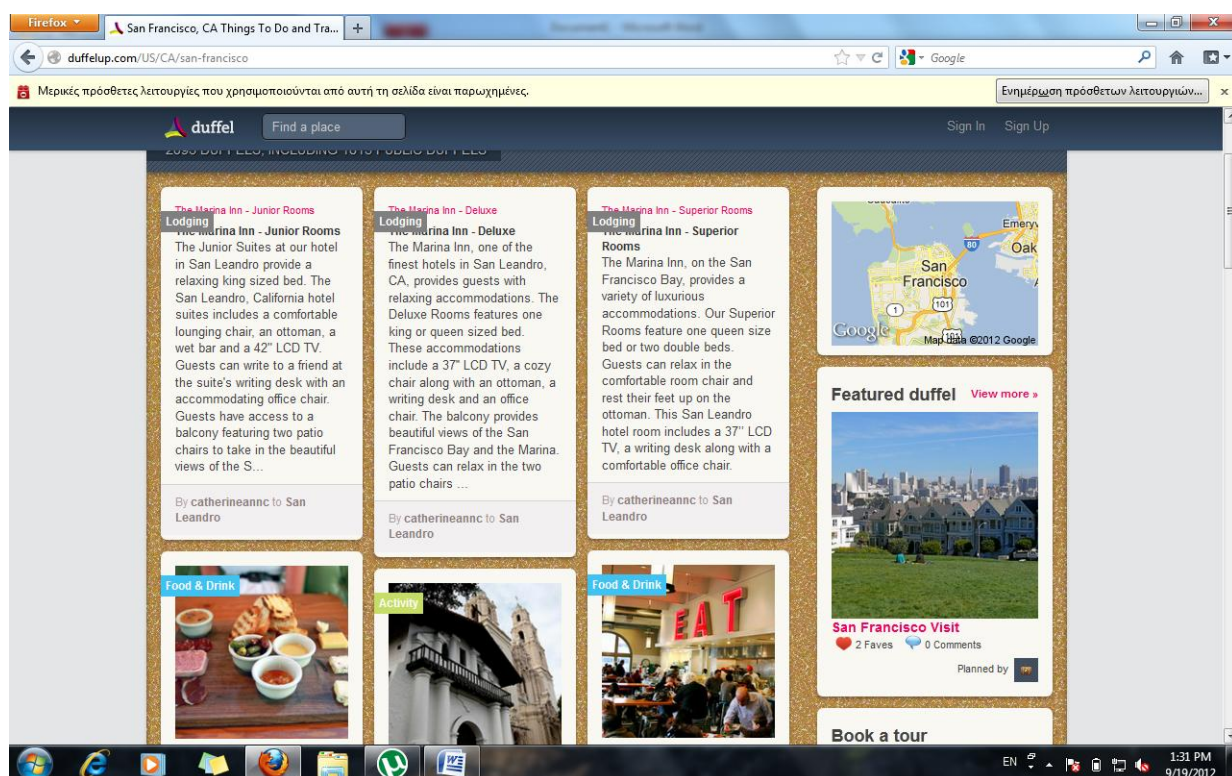
Κάποιες άλλες απλά βάζεις τον προορισμό και σου επιστρέφει έτοιμα αποτελέσματα με προτάσεις χωρισμένα σε κατηγορίες . Άλλες εφαρμογές είναι διαδραστικές με τους ντόπιους και να μεν επιλέγεις τον προορισμό αλλά στην συνέχεια επικοινωνείς και με ντόπιους όπου πραγματοποιείς διάφορες ερωτήσεις και εκείνοι σου απαντούν. Ας δούμε κάποιες αυτές για να δούμε και τι προσφέρουν.

2.4 Η εφαρμογή Duffel

Σε αυτή την εφαρμογή ο χρήστης έρχεται σε επαφή με ένα όμορφο περιβάλλον τύπου πίνακα ανακοινώσεων όπου εισάγει την πόλη προτίμησης του και στη συνέχεια «καρφίτσωνονται» στον πίνακα οι προτάσεις για τη συγκεκριμένη πόλη που μπορεί να περιλαμβάνουν επιλογές φαγητού, διασκέδασης, διαμονής, ξενοδοχείων, αγορών και άλλων πληροφοριών, αλλά με ένα μειονέκτημα, δεν υπάρχει διάκριση των επιλογών στις προτιμήσεις του χρήστη αλλά εμφανίζει όλες τις δραστηριότητες της πόλης επιλογής. Στην εικόνα 1 βλέπουμε στιγμιότυπο της εφαρμογής που ζητά την εισαγωγή της πόλης προτίμησης ενώ στην εικόνα 2 βλέπουμε τα αποτελέσματα.



Εικόνα 12 Περιβάλλον Εφαρμογής Duffel



Εικόνα 13 Αποτελέσματα εφαρμογή [Πηγή:www.duffel.com]

2.5 Εφαρμογή NileGuide

Η επόμενη εφαρμογή φαίνεται να έχει αρκετό ενδιαφέρον αφού βασίζεται στην ανεύρεση πληροφοριών όλων των ειδών από ντόπιους και όχι από κάποια βάση δεδομένων. Ο χρήστης γράφει την ερώτηση του και τι ακριβώς επιθυμεί να μάθει και κάποιος ντόπιος απαντά. Φυσικά έχει κάποια μειονεκτήματα γιατί σε καμία περίπτωση δεν μπορεί να αποτελεί μια εφαρμογή άμεσης αναζήτησης αλλά πρέπει να αποτελεί μια εφαρμογή μακροχρόνιου προγραμματισμού ενός ταξιδιού. Και αυτό είναι φυσιολογικό γιατί πρέπει να υπάρχει και ο κατάλληλος χρόνος ανταπόκρισης του ντόπιου στην ερώτηση, εκτός βέβαια αν η συγκεκριμένη εφαρμογή έχει από πίσω της ντόπιους εργαζόμενους online που εξυπηρετούν άμεσα το ερώτημα που τίθεται και δίνουν γρήγορες πληροφορίες. Στην εικόνα 3 βλέπουμε το περιβάλλον του χρήστη



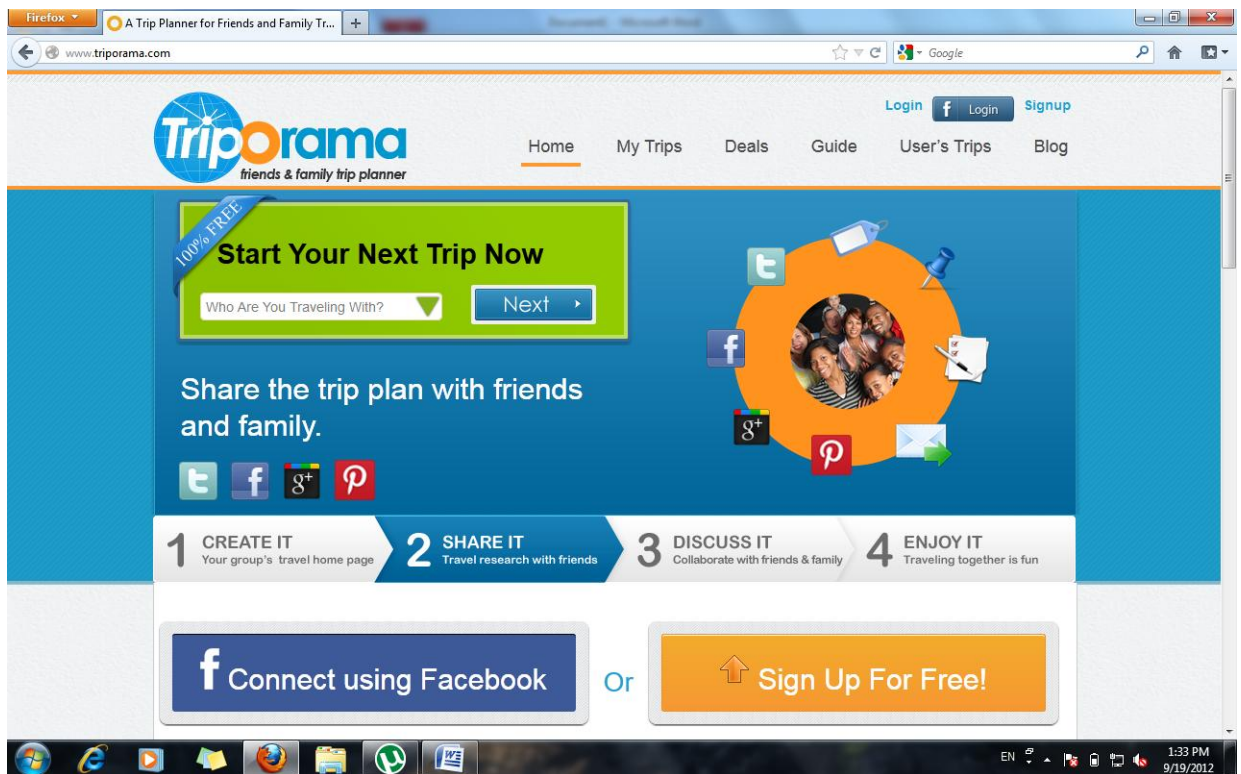
Εικόνα 14 Περιβάλλον χρήστη του Nileguide [Πηγή:www.nileguide.com]

2.6 Εφαρμογή Triporama

Σε αυτή την εφαρμογή απαντάται για πρώτη φορά αυτό που λέγαμε οδηγό συμπλήρωσης που αποτελεί το πιο εύκολο και πιο σαφή τρόπο για την εύρεση πληροφοριών για ένα ταξίδι. Η εφαρμογή είναι εύκολη στη χρήση, σε κάθε στάδιο «ρωτά» τον χρήστη για διάφορες πληροφορίες προκειμένου να δημιουργηθεί το κατάλληλο προφίλ του χρηστή και να αναζητηθούν πληροφορίες βάσει αυτού. Οι πληροφορίες που λαμβάνονται μέσω ερωτήσεων είναι του τύπου

- Ποια περιοχή θα επισκεφτείτε;
- Από ποιόν θα συνοδεύεστε;
- Τι είδους δραστηριότητες προτιμάται;
- Περίοδος επίσκεψης κ.α.

που βοηθούν στην ολοκλήρωση του προφίλ του τουρίστα. Επιπλέον η συγκεκριμένη εφαρμογή δίνει τη δυνατότητα του διαμοιρασμού των πληροφοριών που έλαβε ο χρήστης με άλλους φίλους του, της συζήτησης των προτάσεων αυτών και τέλος εκτύπωσης αυτών των πληροφοριών για χρησιμοποίησή τους. Η εικόνα 4 δίνει μια γεύση της εφαρμογής

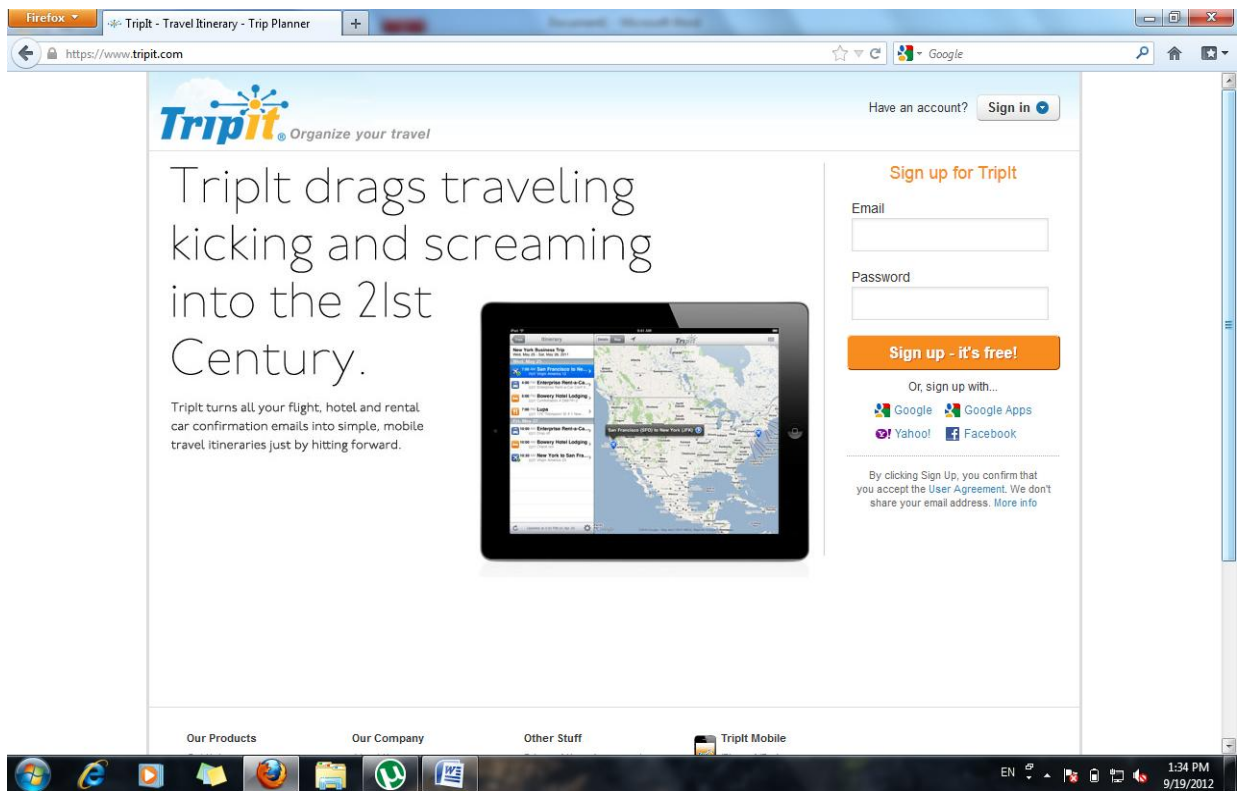


Εικόνα 15 Triporama [Πηγή:www.triporama.com]

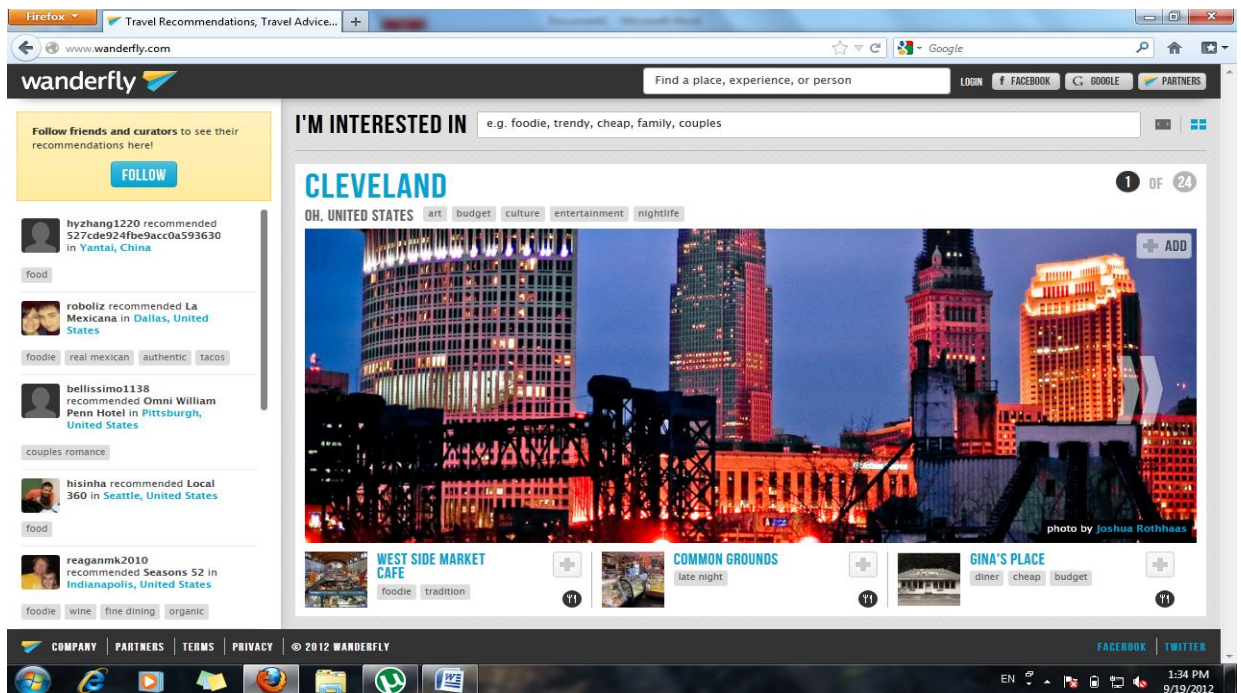
2.7 Εφαρμογή TripIt και WanderFly

Οι εφαρμογές αυτές είναι πιο απλές και χρειάζονται να δημιουργηθεί ένας λογαριασμός χρήστη. Συνήθως η δημιουργία λογαριασμού πρέπει να αποφεύγεται σε τέτοιου είδους εφαρμογές γιατί εκτός του ότι δυσκολεύουν τον προγραμματισμό και χρειάζονται διαδικασίες αυθεντικοποίησης, πιστοποίησης και ασφάλειας των δεδομένων του χρήστη, αφετέρου δυσχεραίνουν τον χρήστη στη χρήση του προγράμματος και κάποιους άλλους τους αποτρέπουν κιόλας.

Οι εφαρμογές αυτές είναι μαζικής πληροφορίας γιατί ενημερώνουν γενικά τον τουρίστα για το μέρος επιλογής του αλλά δίνουν πληροφορίες ολοκληρωμένες με εικόνες και οδηγίες για τις διαδρομές με χρήση GPS υπηρεσιών. Στις παρακάτω εικόνες βλέπουμε τις εφαρμογές αυτές.



Εικόνα 16 TripIt [Πηγή:www.tripit.com]



Εικόνα 17 WanderFly application [Πηγή:www.wanderfly.com]

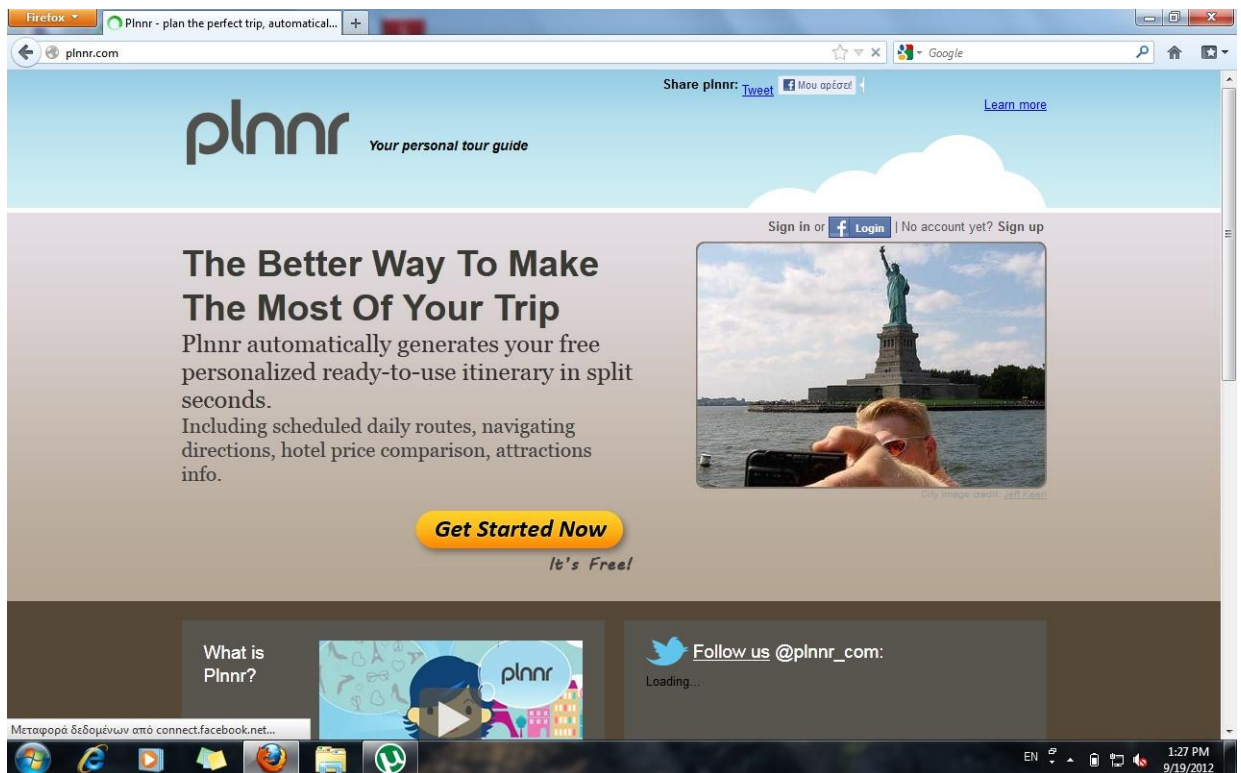
2.8 Εφαρμογή Plnr

Αυτή η υπηρεσία αποτελεί την πιο ολοκληρωμένη εφαρμογή που αφορά την υλοποίηση ενός σχεδιαστή τουρισμού. Διαθέτει απλώς ένα οδηγό συμπλήρωσης ενώ με την ταυτόχρονη χρήση διαφόρων υπηρεσιών της Google όπως το Google maps και το Google directions δίνει μια άλλη διάσταση στην παροχή πληροφοριών και οδηγιών. Περιέχει εικόνες από όλα τα μέρη που προτείνει ενώ ταυτόχρονα μπορεί να παρέχει και δυνατότητες σύγκρισης τιμών ξενοδοχείων και εισιτηρίων. Όλα τα στοιχεία είναι χωρισμένα σε κατηγορίες ενώ υπάρχουν και ενημερώσεις για το πότε είναι ανοιχτά τα διάφορα μέρη. Υπάρχει δυνατότητα επιλογής ξεχωριστά κάθε δραστηριότητας ή επιλογή των καλύτερων. Δίνεται η επιλογή και οικονομικής επιλογής ανάλογα με τα χρήματα που σκοπεύει να διαθέσει ο τουρίστας ενώ υπάρχει ακόμα και δυνατότητα επιλογής έντασης αναλόγως του τύπου του χρήστη. Στο τέλος του οδηγού εμφανίζονται όλα τα αποτελέσματα με εικόνες και οδηγίες ενώ υπάρχουν διαθέσιμοι και οι αντίστοιχοι χάρτες με τα προτεινόμενα σημεία ενδιαφέροντος.

Ας δούμε όμως λίγο πιο αναλυτικά το πρόγραμμα πως υλοποιείται.

Ο χρήστης έρχεται σε επαφή με ένα όμορφο περιβάλλον με απαλά χρώματα σε τόνους μπλε γκρι. Η επιλογή του χρώματος δεν είναι τυχαία γιατί όπως το κόκκινο δίνει εντολή στον εγκέφαλο για φαγητό και χρησιμοποιείται από τις περισσότερες αλυσίδες εστιατορίων έτσι και οι τόνοι του μπλε δίνουν την εντολή της ξεκούρασης και των διακοπών.

Η εφαρμογή δεν χρειάζεται δημιουργία λογαριασμού είναι έτοιμη με το πάτημα ενός κουμπιού να ξεκινήσει ο χρήστης όπως φαίνεται στην εικόνα.



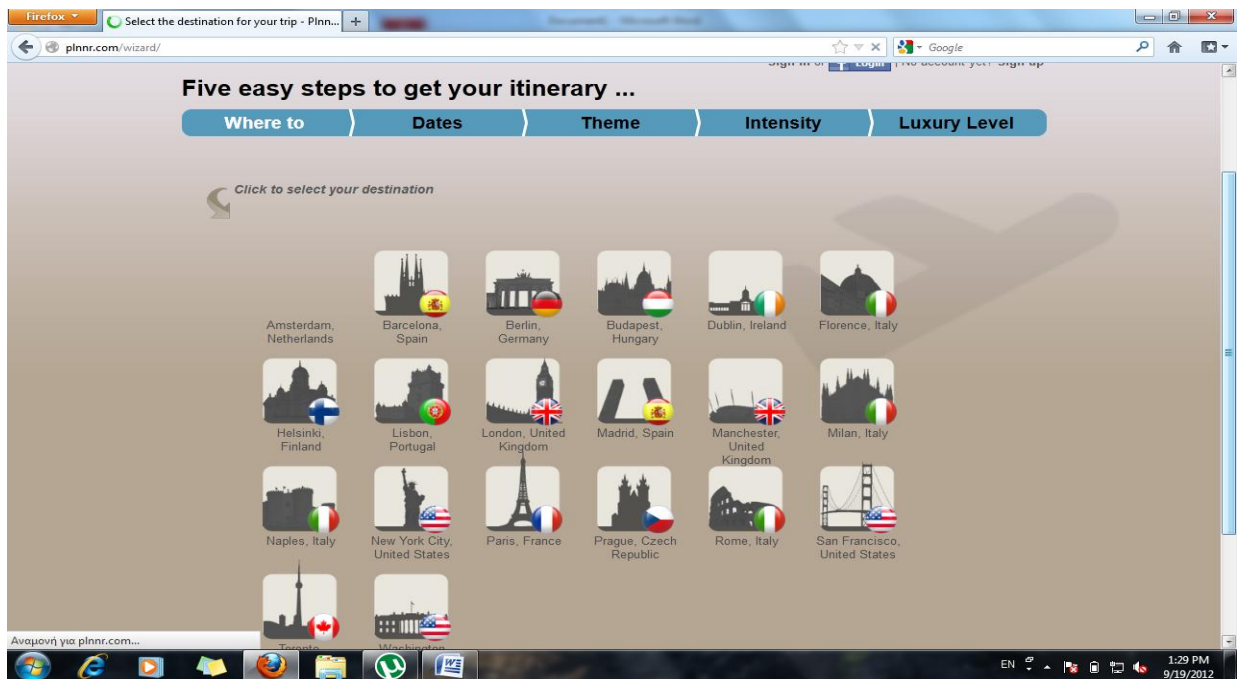
Εικόνα 18 Έναρξη εφαρμογής Plnnr [Πηγή: www.plnnr.com]

Στην εικόνα 8 βλέπουμε και κάποιες παροχές της εφαρμογής όπως αυτές εμφανίζονται κατά την ολοκλήρωση χρήσης της εφαρμογής. Οι παροχές αυτές είναι με σειρά η έναρξη της υπηρεσίας, στη συνέχεια η εμφάνιση των αποτελεσμάτων, ακολουθεί και η προσθήκη άλλων όπως η κράτηση ξενοδοχείου και μετά την επιλογή αυτών η σύγκριση με άλλα του ίδιου είδους. Τέλος είναι η εκτύπωση αυτών των επιλογών και προτάσεων.

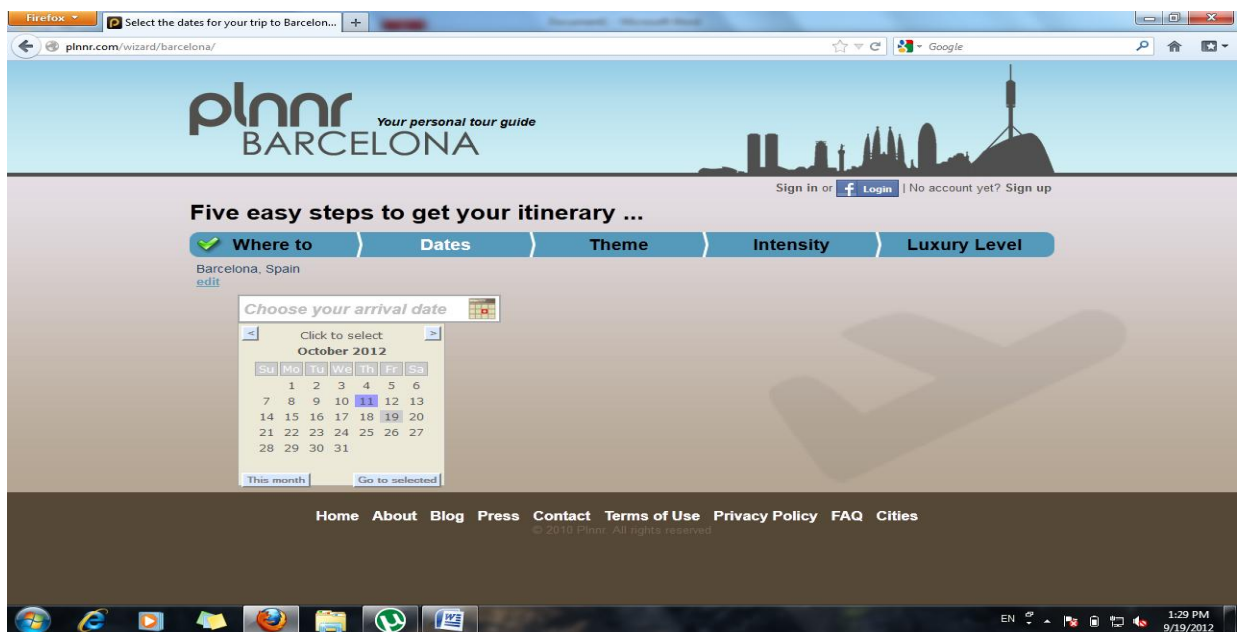


Εικόνα 19 Δυνατότητες εφαρμογής. [Πηγή: www.plnNr.com]

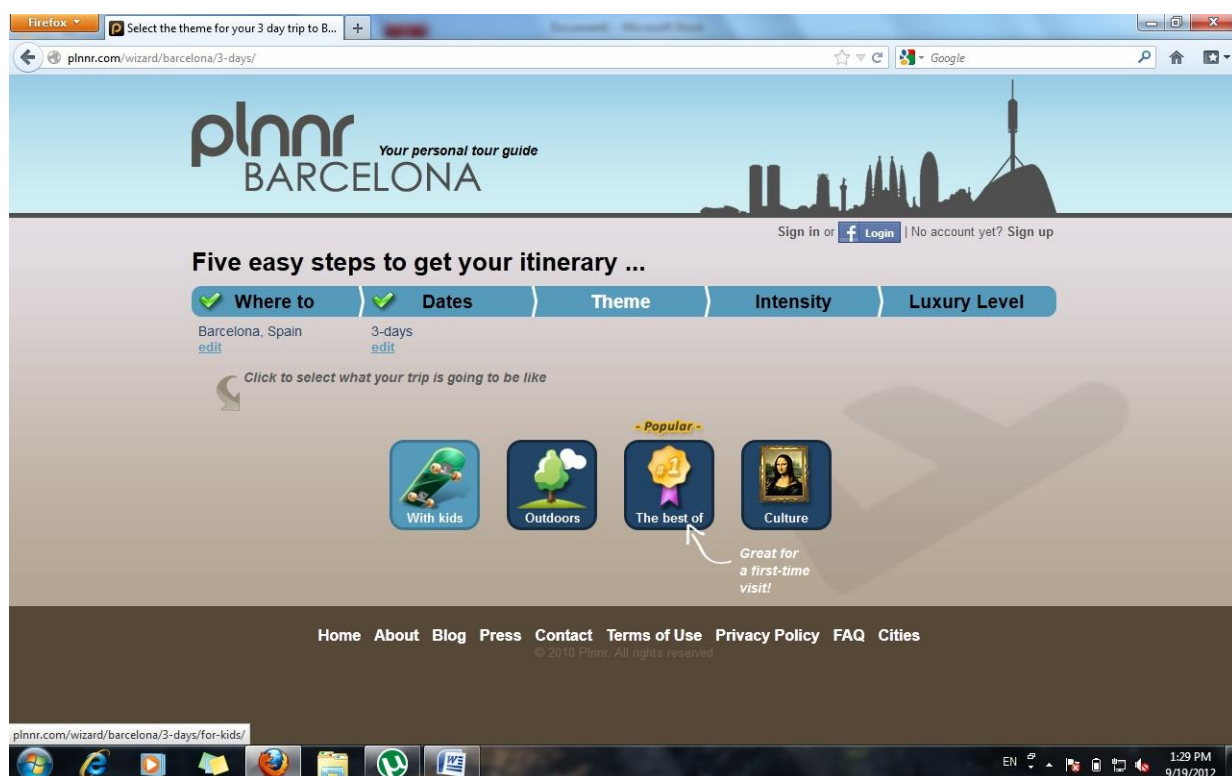
Αμέσως μετά την έναρξη της υπηρεσίας η εφαρμογή μας εμφανίζει τον οδηγό συμπλήρωσης των επιθυμητών στοιχείων. Ο οδηγός ολοκληρώνεται με την συμπλήρωση 5 απλών βημάτων μέσω των οποίων αποκτούμε μια πλήρη εικόνα του προφίλ του χρήστη και των επιθυμιών του. Στις εικόνες 9,10,11,12,13 έχουμε τα 5 βήματα ολοκλήρωσης του οδηγού.



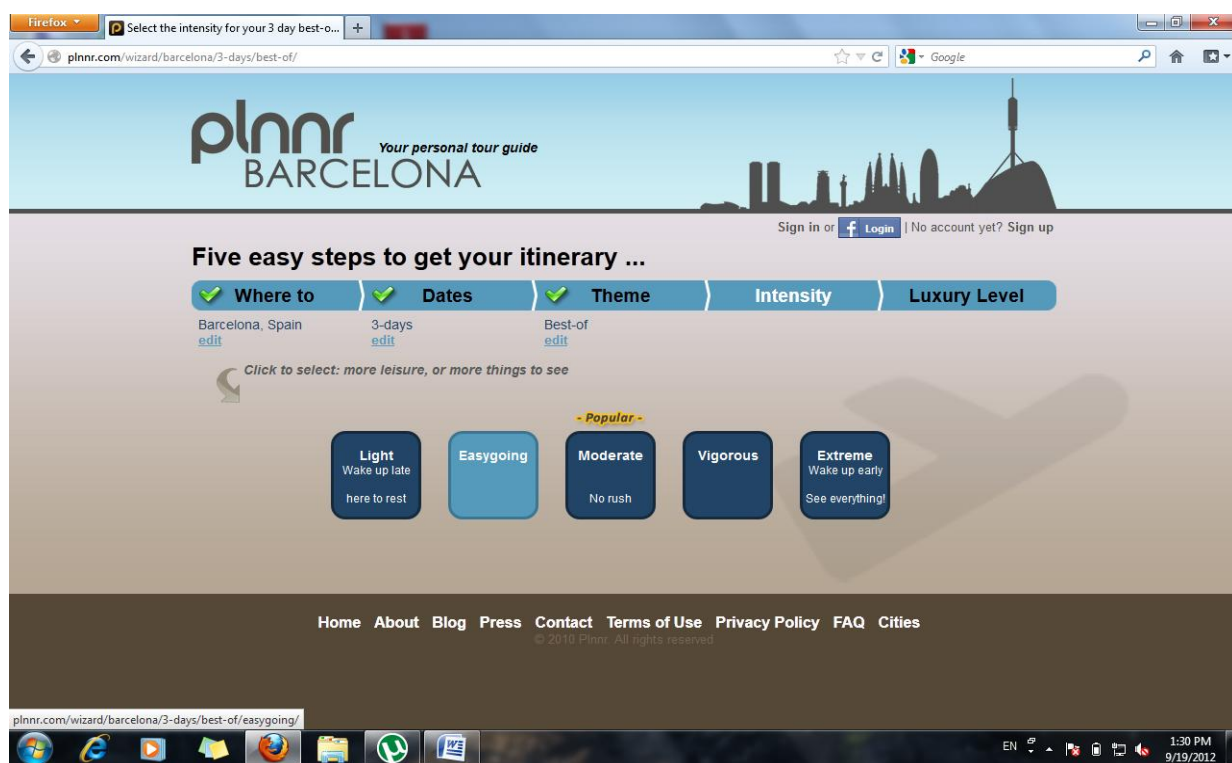
Εικόνα 20 Πρώτο βήμα:Επιλογή μέρους [Πηγή: www.plnr.com]



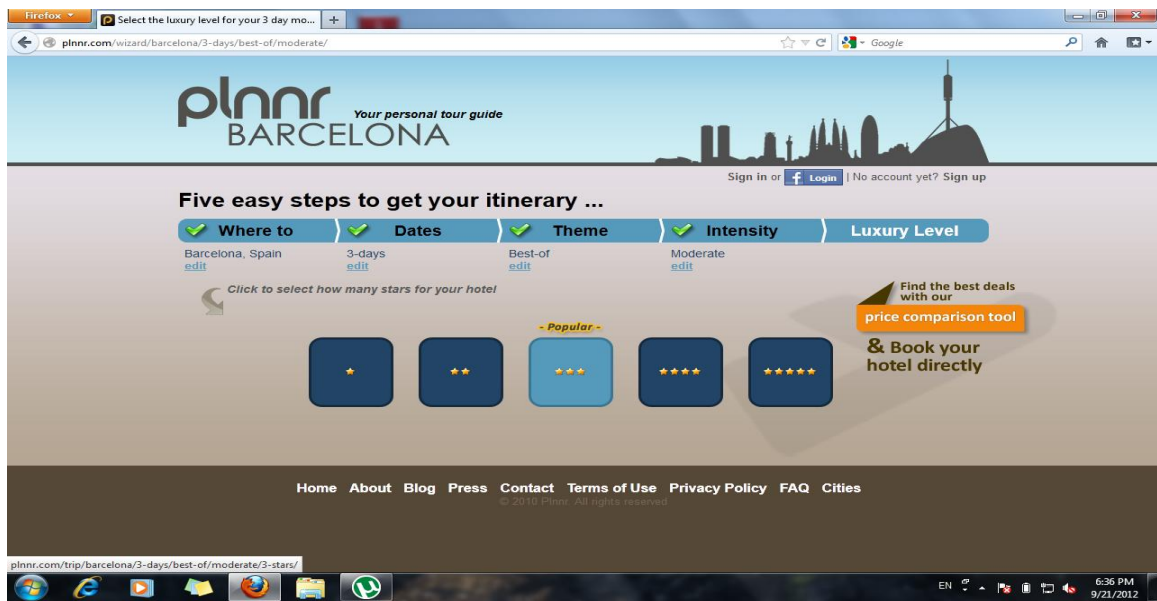
Εικόνα 21 Δεύτερο Βήμα :Επιλογή ημερομηνιών επίσκεψης[Πηγή: www.plnr.com]



Εικόνα 22 Τρίτο Βήμα: Επιλογή Δραστηριότητας [Πηγή: www.plnnr.com]

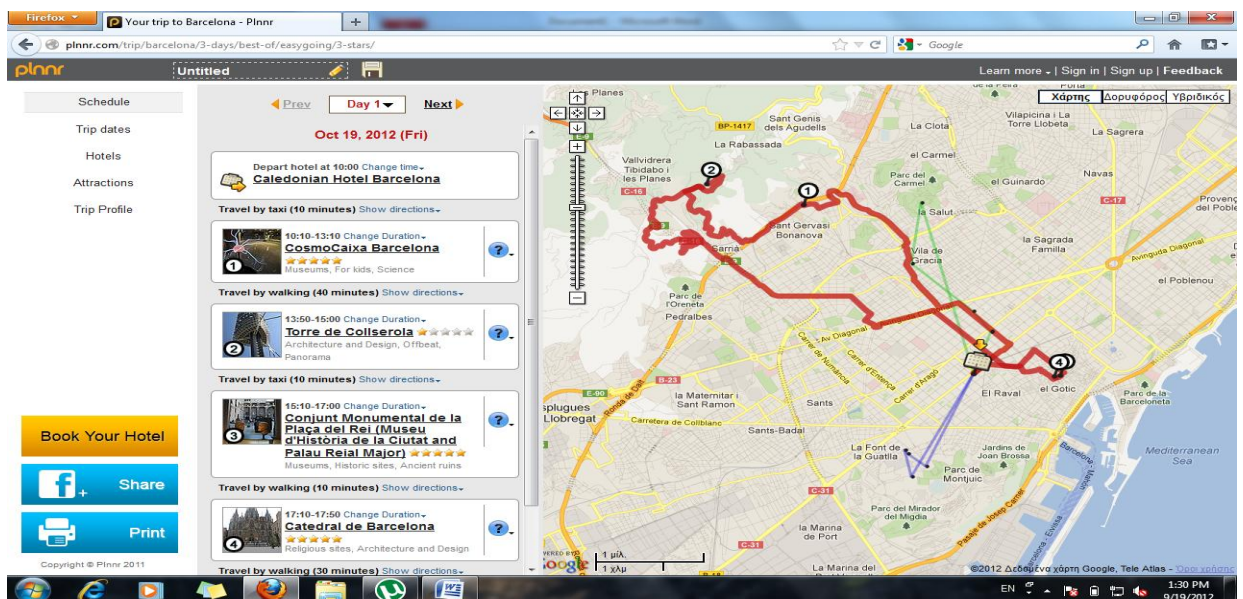


Εικόνα 23 Τέταρτο βήμα: Επιλογή έντασης δραστηριότητας [Πηγή: www.plnnr.com]



Εικόνα 24 Πέμπτο Βήμα: Επιλογή πολυτέλειας

Μετά την ολοκλήρωση του οδηγού πραγματοποιείται η αναζήτηση των στοιχείων και έχουμε την εμφάνιση των αποτελεσμάτων στην οθόνη μας, εικόνα 14. Βλέπουμε τις επιλογές και την οπτικοποίηση που παρέχει η εφαρμογή. Έχουμε τον χάρτη με τα μέρη και ένα προγραμματισμό επισκέψεων με τη διαδρομή που πρέπει να ακολουθηθεί και τη προτεινόμενη σειρά ανα ημέρα επίσκεψης μας. Επίσης έχουμε την δυνατότητα να δούμε και τον προγραμματισμό των άλλων ημερών, να ψάξουμε για ξενοδοχεία και να εκτυπώσουμε όλο το προτεινόμενο πλάνο.



Εικόνα 25 Εμφάνιση αποτελεσμάτων Pinnr [Πηγή: www.pinnr.com]

2.9 Συμπεράσματα

Όπως είδαμε υπάρχουν πολλές αξιόλογες εφαρμογές που έχουν δημιουργηθεί στο εξωτερικό και αποτελούν ολοκληρωμένες λύσεις σχεδιαστών τουρισμού με κορυφαία το r1nnp. Να σημειωθεί ότι το r1nnp δημιουργήθηκε από φοιτητές και όχι από κάποιο κρατικό φορέα, το οποίο μας δείχνει ότι υπάρχουν δυνατότητες δημιουργίας εφαρμογών αξιόπιστων και γρήγορων χωρίς την συμμετοχή κάποιου οργανισμού, αρκεί να υπάρχει θέληση. Επιπλέον πρέπει να αναφέρουμε και το γεγονός ότι σε καμία από αυτές τις εφαρμογές δεν υπάρχει σαν επιλογή η Αθήνα ή κάποιο άλλο ελληνικό μέρος. Αυτό μπορεί να λέει πολλά ή και τίποτα ανάλογα από τη σκοπιά που το βλέπει ο καθένας. Το γεγονός αυτό πρέπει να μας δώσει όμως την ώθηση να οδηγηθούμε στην ανάπτυξη τέτοιων και ακόμα καλύτερων εφαρμογών, διότι αν και δεν υπάρχει κατάλληλη υποδομή από το κράτος, υπάρχει η γνώση το μυαλό και η θέληση για ανάπτυξη.

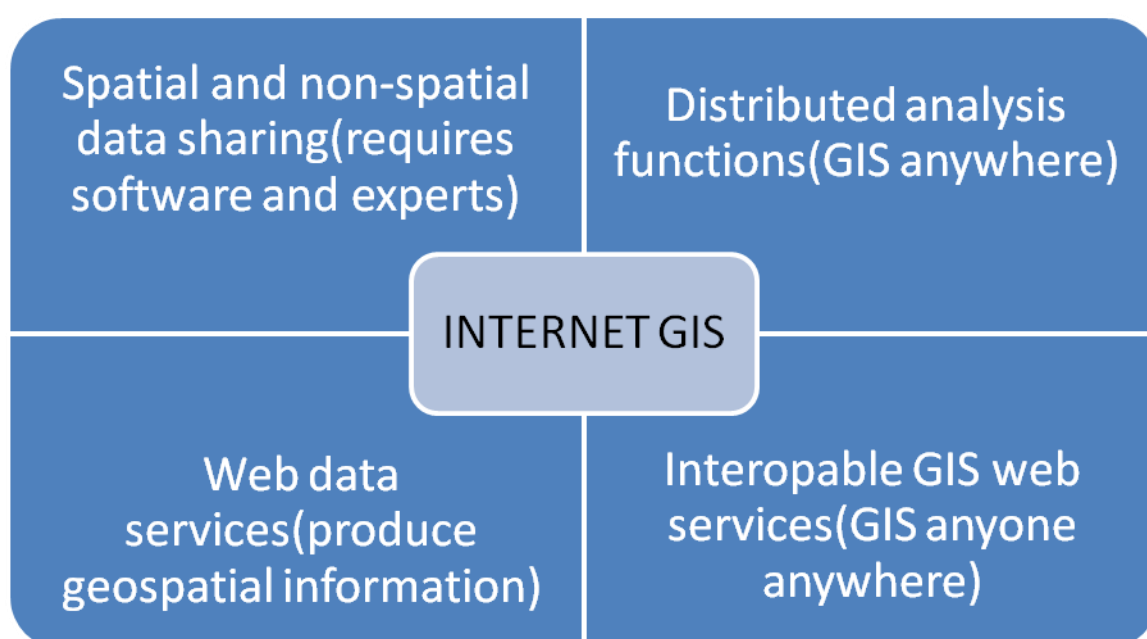
3

ΥΠΗΡΕΣΙΕΣ ΔΙΑΔΙΚΤΥΟΥ

3.1 Γενικά

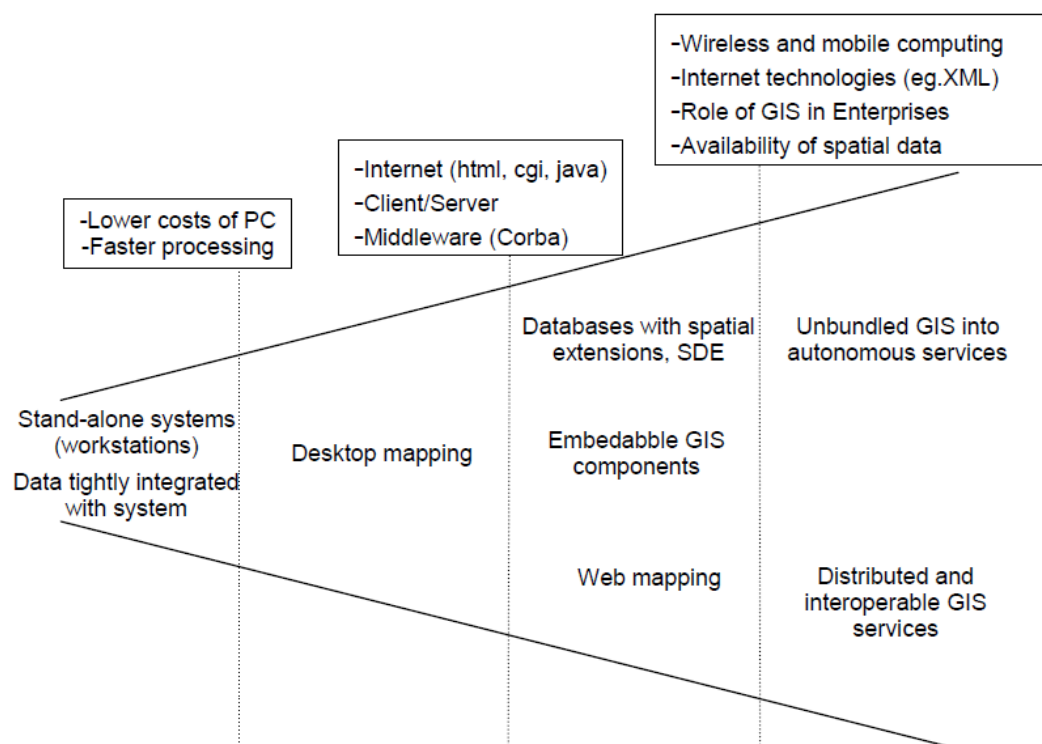
Τα γεωγραφικά πληροφοριακά συστήματα είναι συστήματα που χρησιμοποιούνται ευρέως για την επεξεργασία, ανάλυση και διαχείριση δεδομένων χωρικών και μη. Με τη χρήση γεωγραφικών συστημάτων επιτυγχάνεται η οπτικοποίηση και η ευκολότερη διαχείριση της πληροφορίας με αποτέλεσμα τη γρηγορότερη επίλυση προβλημάτων σε πολλά επίπεδα.

Παράλληλα όμως με την εξέλιξη του διαδικτύου ήρθε και η εξέλιξη των γεωγραφικών πληροφοριακών συστημάτων, το λεγόμενο “internet GIS” το οποίο έχει γίνει ευρέως αποδεκτό από διάφορους φορείς και χρησιμοποιείται για την υλοποίηση, από απλών εφαρμογών μέχρι υψηλών απαιτήσεων προγράμματα, από απλούς GIS χρήστες μέχρι GIS managers που ψάχνουν για ολοκληρωμένες λύσεις σε μία εφαρμογή. Στο σχήμα 2 βλέπουμε τους διάφορους τύπους που μπορούμε να συναντήσουμε.



Σχήμα 2. Διάφοροι τύποι INTERNET GIS

Όμως την ραγδαία αύξηση της τεχνολογίας του διαδικτύου ακολούθησαν και τα γεωγραφικά πληροφοριακά συστήματα τα οποία από εφαρμογές σε υπολογιστές desktop πέρασαν στο διαδίκτυο και έφτασαν στα λεγόμενα web services προσφέροντας διαλειτουργικότητα και σύνολο πληροφοριών και δεδομένων σε μια υπηρεσία προσβάσιμη και εύκολα χρησιμοποιήσιμη από όλους. Στην εικόνα 15 βλέπουμε μια πολύ καλή αναπαράσταση της εξέλιξης των γεωγραφικών πληροφοριακών συστημάτων με την αντίστοιχη εξέλιξη της χρησιμοποιούμενης τεχνολογίας.



Εικόνα 26. GIS εξέλιξη [Πηγή: Internet Gis. Wiley, 2003]

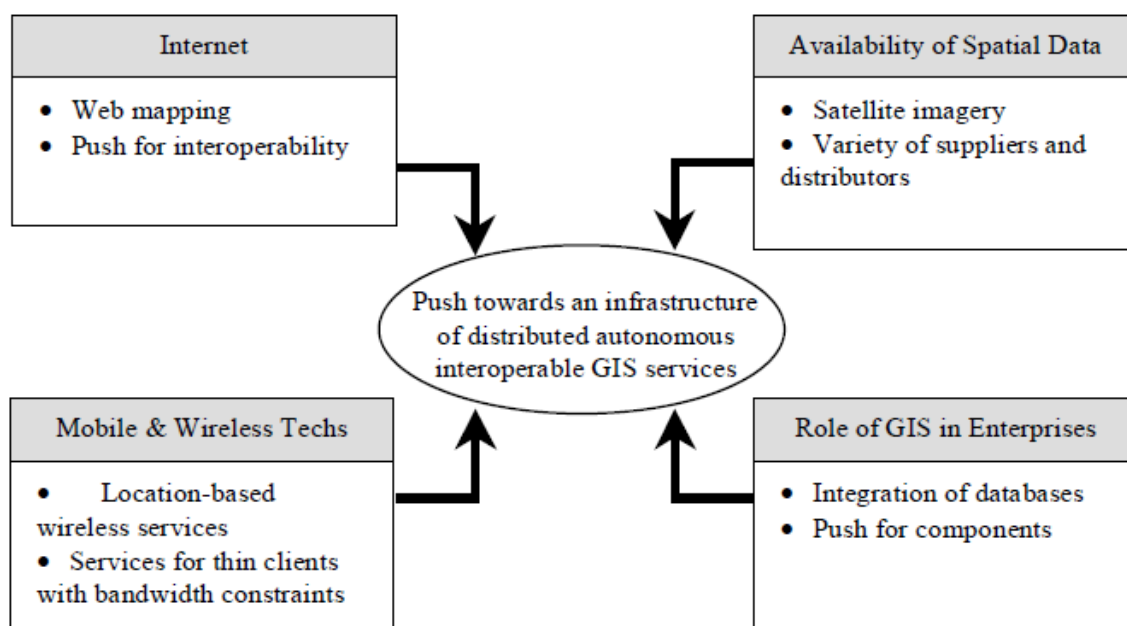
Η τελευταία στάση όπως βλέπουμε στην λεωφόρο της εξέλιξης της πληροφορίας είναι τα gis web services. Τα web services αποσκοπούν στην δημιουργία ενός νέου τρόπου σκέψης ανάπτυξης εφαρμογών και διαχείρισης χωρικών και μη δεδομένων μέσω του διαδικτύου. Τα web services έχουν πολλές δυνατότητες όπως:

- Εύρεση διεύθυνσης
- Επεξεργασία και ανάλυση χωρικών δεδομένων και χρήση ερωταποκρίσεων
- Πλοήγηση/εύρεση δρομολογίου
- Χαρτογράφηση
- Location based services
- Marketing
- Τουριστικές υπηρεσίες κ.α
- Εφαρμογές γεωδαισίας
- Κτηματολόγιο

Τα web services αυτά στην ουσία δημιουργήθηκαν για να μπορέσουν να απλοποιήσουν την χρήση των γεωγραφικών δεδομένων και να ενοποιήσουν τις διάφορες τεχνολογίες ώστε οποιοσδήποτε να μπορεί εύκολα να αναπτύξει εφαρμογές στο δικό του σύστημα χωρίς την ανάγκη απόκτησης επιπλέον υψηλού κόστους λογισμικού και συστήματος αφού του προσφέρεται μέσω του διαδικτύου ένα φιλικό και εύχρηστο περιβάλλον για την υλοποίηση και επεξεργασία χωρικών δεδομένων και ανάπτυξη εφαρμογών όπως συστημάτων πλοήγησης, χωροταξικό σχεδιασμό, ηλεκτρονικό εμπόριο, τουρισμό κ.α.

Τα GIS web services έδωσαν τη δυνατότητα στους χρήστες να ενσωματώσουν στις εφαρμογές τους δεδομένα χαρτών χωρίς την ανάγκη να εγκαταστήσουν άλλη εφαρμογή ή να αποθηκεύσουν στον υπολογιστή τους τεράστιο όγκο δεδομένων. Από τη στιγμή όμως που διαχειριζόμαστε τεράστιο όγκο δεδομένων θα πρέπει να εξασφαλίσουμε την ποιότητα του αποτελέσματος που μας δίνει ένα web service. Οπότε σημαντικό σε μια υπηρεσία είναι και η διασφάλιση της ποιότητας της. Με τη διασφάλιση της ποιότητας εξασφαλίζουμε τη ταχύτητα και την αξιοπιστία της υπηρεσίας και των δεδομένων που προσφέρονται.

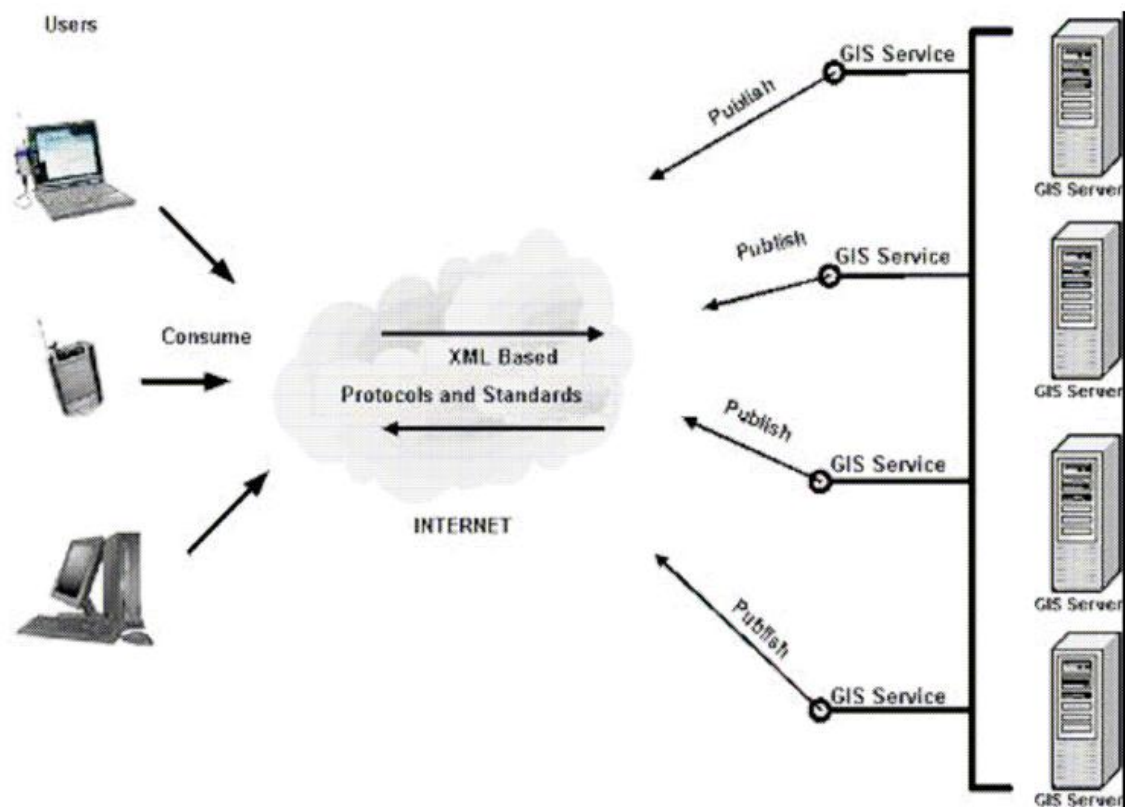
Τα web services χρησιμοποιούνται ευρέως από εταιρίες μέχρι ιδιώτες. Ανάλογα με την χρήση τους έχουμε web process services, web mapping services, web feature services κ.α. Χρησιμοποιούνται πρωτόκολλα και γλώσσες ευρέως γνωστά ώστε τελικά να εξασφαλίζεται η πολυπύθητη διαλειτουργικότητα και ο εύκολος διαμοιρασμός των πληροφοριών καθώς και η εύκολη χρήση και επέκταση των λειτουργιών αυτών. Στην εικόνα 16 βλέπουμε αυτή τη συγκέντρωση των διαφόρων τεχνολογιών σε μια GIS υπηρεσία.



Εικόνα 27. Autonomous distributed and interoperable GIS Service [Πηγή: Internet Gis. Wiley, 2003]

3.2 Λειτουργία των GIS WEB SERVICES

Πολύ σημαντικό για την κατανόηση των web services είναι να εξηγήσουμε τον τρόπο λειτουργίας τους. Πρέπει να δούμε πως λειτουργούν τα web services στα γεωγραφικά πληροφοριακά συστήματα, τι τεχνολογία χρησιμοποιούν και πως οι χρήστες τελικά αποκτούν πρόσβαση και χρησιμοποιούν αυτές τις υπηρεσίες. Στην εικόνα 17 βλέπουμε ένα παράδειγμα από χρήστες που επικοινωνούν με gis servers με τη χρήση κάποιου πρωτόκολλου και γλώσσας.



Εικόνα 28. Λειτουργία Web Service [Πηγή:Internet Gis. Wiley, 2003]

Τα web services έχουν τα ακόλουθα χαρακτηριστικά:

- Είναι εφαρμογές
- Επικοινωνούν με πρωτόκολλα δικτύου
- Ανιχνεύονται με χρήση UDDI
- Μπορούν να χρησιμοποιηθούν από και μέσα σε άλλες εφαρμογές
- Χρησιμοποιούν XML

Πλατφόρμες που χρησιμοποιούν

- Το application Programming interface(API) καθορίζει τον τρόπο πρόσβασης στην υπηρεσία
- Ένα API είναι ένα δημοσιευμένο χαρακτηριστικό το οποίο περιγράφει πως άλλα προγράμματα έχουν πρόσβαση στις λειτουργίες της υπηρεσίας
- Διαθέτει πηγαίο κώδικα
- Πραγματοποιούνται με αντικειμενοστραφή σχεδίαση
- Το GIS API χρησιμοποιείται για να ελέγχουν τη μεταφορά και παράδοση των δεδομένων στον πελάτη από τον WEB SERVER.
- Χρησιμοποιούνται απλές γλώσσες για προγραμματισμό όπως Java,Python,JavaScript ή V
-

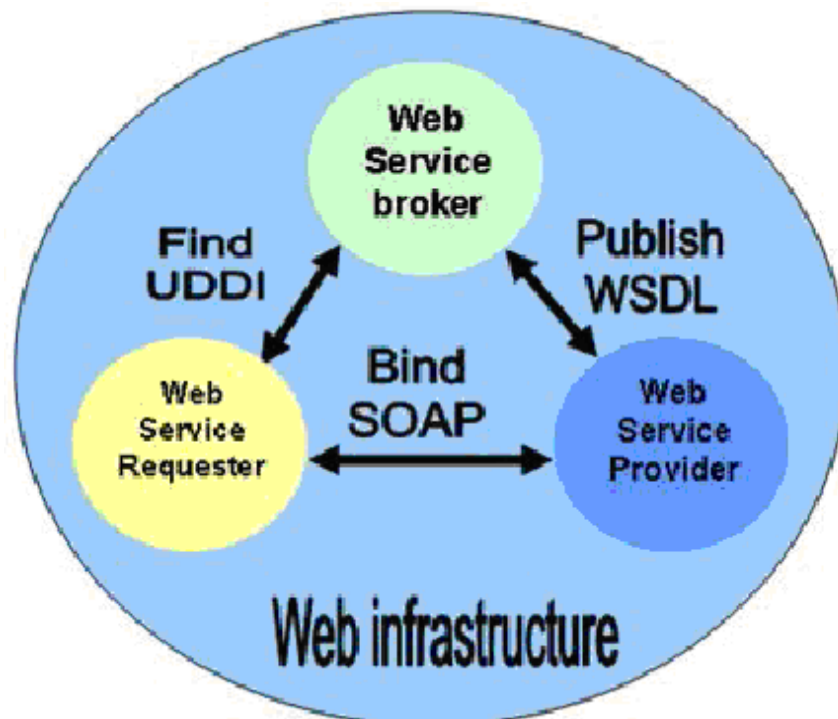
Τεχνολογίες που χρησιμοποιούν

- SOAP (simple object access protocol) είναι ένα πρωτόκολλο για ανταλλαγή XML μηνυμάτων στο διαδίκτυο με χρήση http.
- WSDL (web service description language) είναι ένα xml έγγραφο το οποίο περιγράφει ένα σύνολο soap μηνυμάτων και πως αυτά ανταλλάσσονται.

3.3 Αρχιτεκτονική SOA

Η ανάπτυξη της τεχνολογίας και του internet όπως αναφέρθηκε οδήγησε στο πέρασμα από τα κεντροποιημένα συστήματα στα κατανεμημένα συστήματα και στο λεγόμενο SOA (service oriented architecture).

Το SOA δημιουργεί μια δυναμική, ευέλικτη, κατανεμημένη και αναπρογραμματιζόμενη υπηρεσία στο ιντερνέτ που περιέχει όλα τα απαραίτητα δεδομένα και πληροφορίες που χρειάζεται ο κάθε χρήστης. Το σημαντικότερο χαρακτηριστικό του SOA είναι οι υπηρεσίες. Οι υπηρεσίες αυτές αποτελούν ένα καλά καθορισμένο σύνολο από ενέργειες ανεξάρτητες όμως από άλλες υπηρεσίες. Στην εικόνα 18 βλέπουμε τις βασικές διεργασίες του SOA.



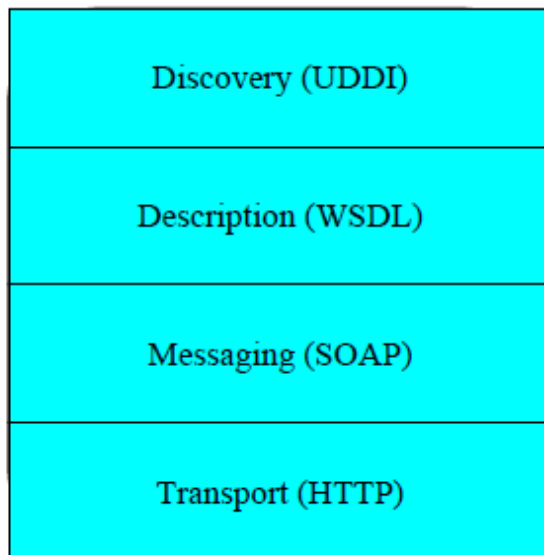
Εικόνα 29. SOA Architecture [Πηγή: Web Services in a nutshell. Topley, 2002]

Το SOA διαθέτει 3 οντότητες:

1. Service provider ο οποίος δημιουργεί και παρέχει αυτές τις υπηρεσίες στο ιντερνέτ
2. Service requester ο οποίος βρίσκει και χρησιμοποιεί την υπηρεσία που επιθυμεί
3. Service broker βοηθά στην επικοινωνία μεταξύ του requester και του provider

3.4 Πρωτόκολλα των WEB SERVICES

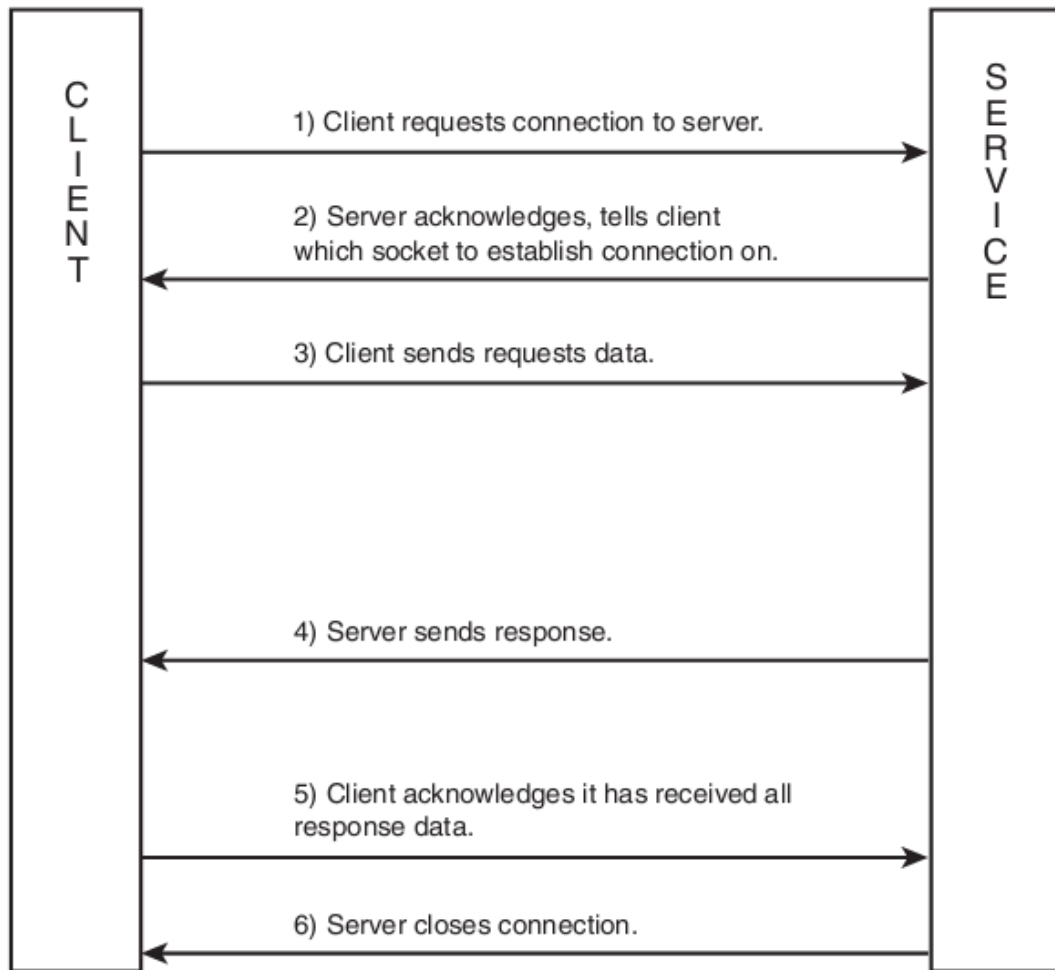
Τι είναι όμως αυτά τα πρότυπα; Τα πρότυπα αυτά είναι πρωτόκολλα δηλαδή κανόνες για το πώς θα επιτυγχάνεται η σύνδεση και η επικοινωνία μεταξύ των διαφόρων κόμβων (υπηρεσιών) στο διαδίκτυο με σκοπό φυσικά την αξιόπιστη ανταλλαγή δεδομένων και πληροφοριών. Στην εικόνα 19 βλέπουμε τη στοίβα αυτή που περιλαμβάνει αυτά τα απαραίτητα πρωτόκολλα και την αντίστοιχη λειτουργία τους για την υλοποίηση αυτών των web services



Εικόνα 30 . Πρωτόκολλα των Web Services [Πηγή: Web Services in a nutshell, Topley 2002]

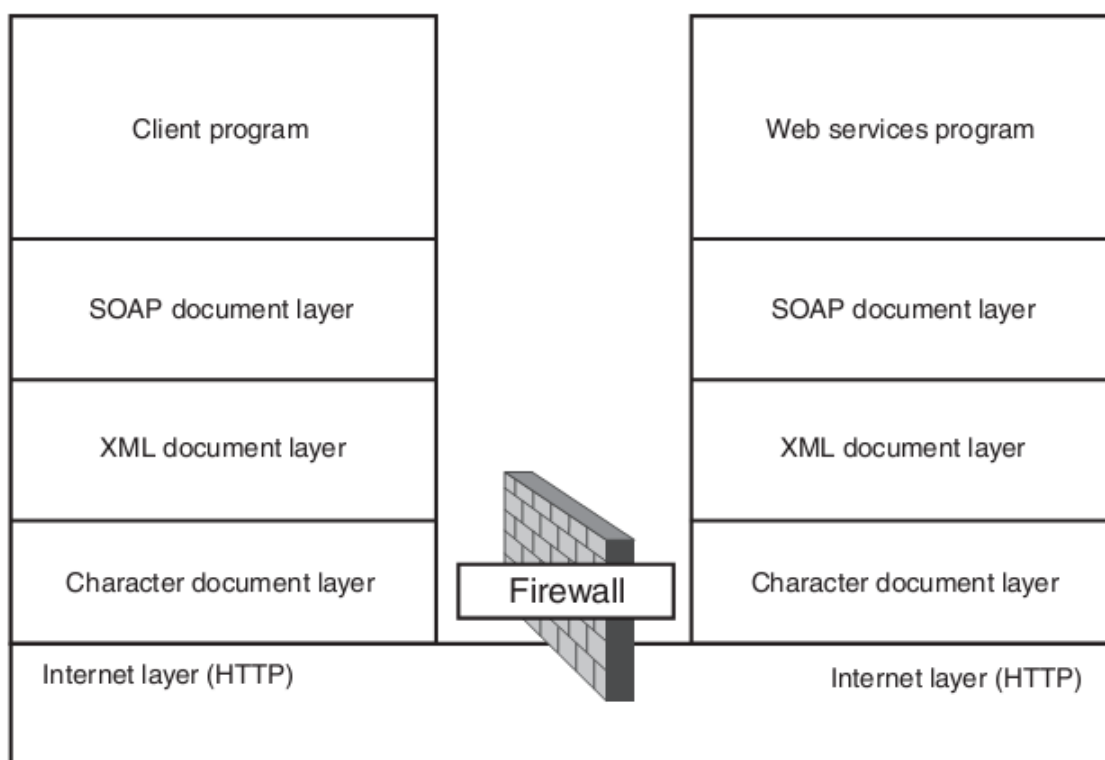
Ας δούμε λίγο πιο αναλυτικά τι είναι και πως λειτουργούν αυτά τα πρωτόκολλα.

Το πρώτο πρωτόκολλο είναι το transfer protocol το οποίο είναι υπεύθυνο για τη μεταφορά των μηνυμάτων μεταξύ των διαφόρων διαδικτυακών εφαρμογών δηλαδή από τον browser στον server. Το πρωτόκολλο που χρησιμοποιείται είναι το γνωστό HTTP το οποίο βρίσκεται στα χαμηλότερα επίπεδα μεταφοράς. Στην εικόνα 20 βλέπουμε πώς ένας πελάτης επικοινωνεί με τον εξυπηρετητή και επιτυγχάνεται μια σύνδεση.



Εικόνα 31. Επίτευξη σύνδεσης client-server [Πηγή Oreilly Web Services in a nutshell 2002]

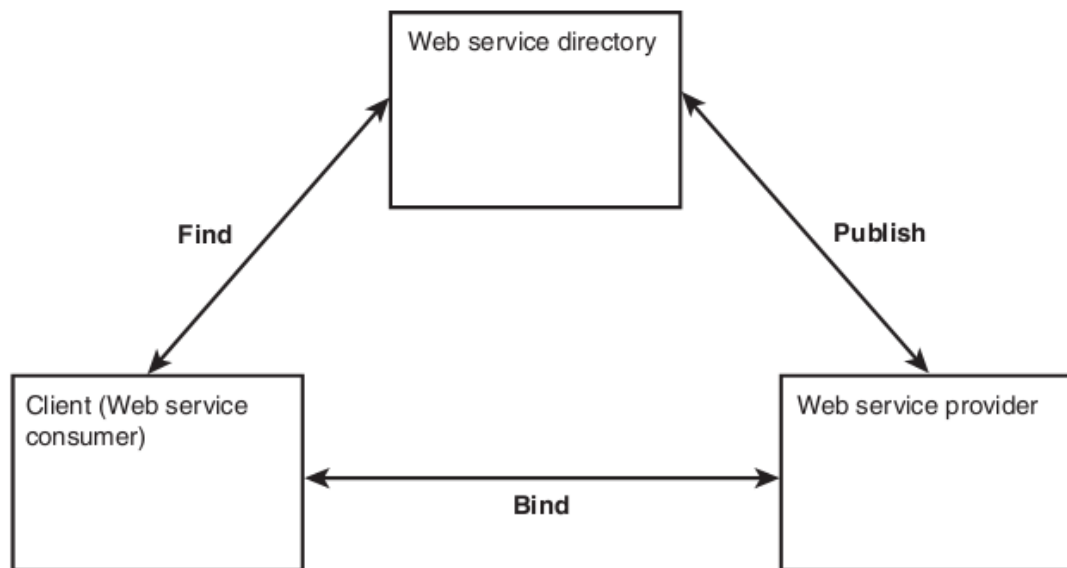
Το δεύτερο πρωτόκολλο είναι το λεγόμενο messaging protocol το οποίο είναι υπεύθυνο για την κωδικοποίηση του μηνύματος σε ένα XML αρχείο. Αυτό το αρχείο με αυτή την μορφή είναι εύκολα αναγνωρίσιμο από το διαδίκτυο και λέγεται SOAP. Στην εικόνα 21 βλέπουμε τα επίπεδα που περνάει με τη λειτουργία του SOAP.



Εικόνα 32. Λειτουργία SOAP [Πηγή Oreilly Web Services in a nutshell 2002]

Ο υπολογιστής 1-πελάτης στέλνει ένα xml document στον υπολογιστή 2 με http. Το firewall του υπολογιστή 2 επιτρέπει την είσοδο του αρχείου από το τείχος προστασίας του. Ο web server στον υπολογιστή 2 παίρνει το αρχείο και το μεταφέρει στον soap processor ο οποίος χρησιμοποιεί έναν xml parser για να μεταφράσει το αρχείο. Στη συνέχεια ο soap processor αφού επιβεβαιώσει το μήνυμα το στέλνει στο web server για επεξεργασία. Αφού επεξεργαστεί και δημιουργήσει το αρχείο με τα επεξεργασμένα δεδομένα μεταφέρονται πάλι με soap σε μορφή xml στο υπολογιστή 1-πελάτη. Αυτός παίρνει το αρχείο το διαβάζει και αφού επιβεβαιώσει ότι πήρε αυτό που ζήτησε στέλνει μια επιβεβαίωση στον υπολογιστή 2 (server) και τελειώνει η διαδικασία.

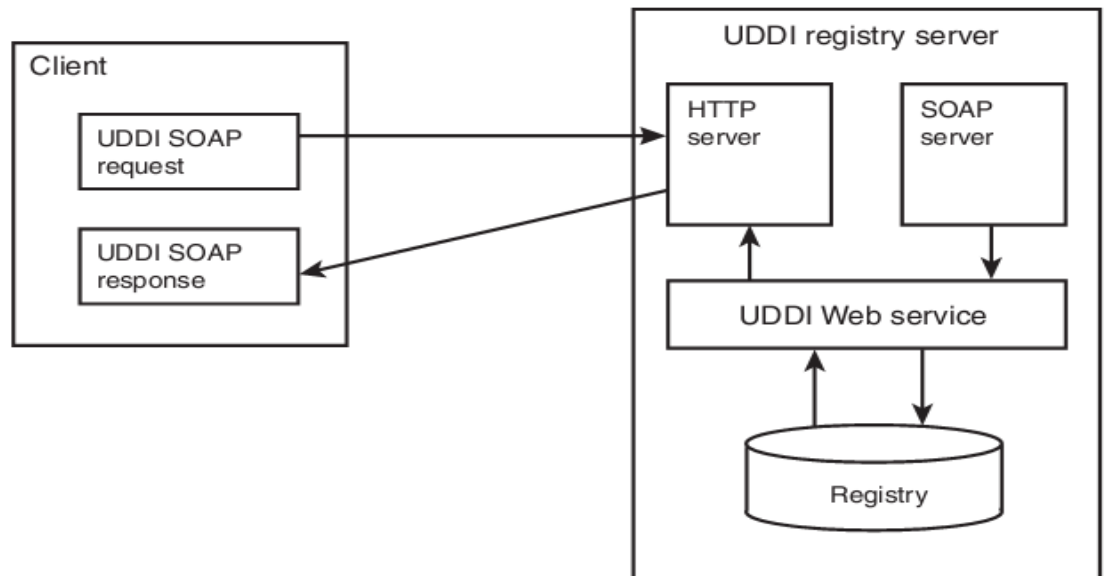
Το τρίτο πρωτόκολλο είναι το WSDL το οποίο χρησιμοποιείται για την περιγραφή του Interface της υπηρεσίας. Δηλαδή μας περιγράφει τι είδους μηνύματα δέχεται και επεξεργάζεται η υπηρεσία μας. Επίσης περιλαμβάνει και διάφορες λεπτομέρειες που αφορούν τον τρόπο σύνδεσης με μια υπηρεσία. Στην εικόνα 22 έχουμε μια πλήρη περιγραφή χρήσης αυτού του πρωτοκόλλου.



Εικόνα 33.Λειτουργία WSDL [Πηγή Oreilly Web Services in a nutshell 2002]

Τέλος, συναντάμε το UDDI το οποίο μετά από κατάλληλη δημοσίευση μιας υπηρεσίας από το παροχέα ο χρήστης έχει την ευκαιρία να ψάξει και να ανακαλύψει την υπηρεσία που έχει δημοσιευθεί (εικόνα 23). Στην ουσία περιγράφεται ως εξής:

- Είναι ο μηχανισμός περιγραφής μιας υπηρεσίας
- Περιγράφει τις άπλες μεθόδους για χρήση μιας υπηρεσίας
- Προσδιορίζει τη μέθοδο εγγραφής και εισόδου στην υπηρεσία



Εικόνα 34. Λειτουργία UDDI [Πηγή Oreilly Web Services in a nutshell 2002]

Συμπερασματικά, βλέπουμε ότι με τη χρήση των παραπάνω πρωτοκόλλων επιτυγχάνουμε την διαλειτουργικότητα και την εύκολη επικοινωνία των χρηστών για χρήση μιας υπηρεσίας σε διάφορες πλατφόρμες, με διαφορετικές γλώσσες προγραμματισμού και χρήστες διαφόρων επιπέδων. Επιτυγχάνουμε την επίλυση δύσκολων και απαιτητικών προβλημάτων εύκολα και χωρίς τη χρήση εξειδικευμένου προσωπικού, χωρίς τη συνεχή υψηλού κόστους αναβάθμιση λογισμικού και χωρίς επιβάρυνση του δικτύου.

3.5 Πρωτόκολλο REST

Το rest αναπτύχθηκε παράλληλα με το HTTP /1,1, με βάση το υπάρχον σχέδιο του HTTP/1.0. Η μεγαλύτερη εφαρμογή ενός συστήματος σύμφωνα με το REST είναι το World Wide Web. Το rest εξηγεί πώς η αρχιτεκτονική του Web αλληλεπιδρά με τα τέσσερα στοιχεία του Web, δηλαδή διακομιστές, πύλες, πληρεξούσια και πελάτες, χωρίς να επιβάλλει περιορισμούς για τους μεμονωμένους συμμετέχοντες. Ως εκ τούτου, το rest ουσιαστικά διέπει τη σωστή συμπεριφορά των συμμετεχόντων.

Οι υπηρεσίες αποτελούνται από τους πελάτες και τους διακομιστές. Οι πελάτες σηματοδοτούν την έναρξη αιτήσεων στους servers. Οι διακομιστές απαντούν στα αιτήματα εκτελώντας τις κατάλληλες διαδικασίες και επιστρέφουν τις κατάλληλες απαντήσεις. Οι αιτήσεις και οι απαντήσεις σχετίζονται με την εύρεση των κατάλληλων πόρων. Ένας πόρος μπορεί να είναι ουσιαστικά οποιαδήποτε συνεκτική και ουσιαστική έννοια που μπορεί να αντιμετωπιστεί. Μια

αναπαράσταση ενός πόρου είναι συνήθως ένα έγγραφο που αποτυπώνει την τρέχουσα ή προβλεπόμενη κατάσταση του πόρου.

Ο πελάτης ξεκινά την αποστολή των αιτήσεων προκειμένου να κάνουν τη μετάβαση σε μια νέα κατάσταση. Αν ένα ή περισσότερα αιτήματα είναι σε εκτέλεση, ο πελάτης θεωρείται ότι βρίσκονται σε μεταβατικό στάδιο. Η αναπαράσταση της κάθε φάσης της εφαρμογής περιέχει συνδέσμους που μπορούν να χρησιμοποιηθούν την επόμενη φορά που ο πελάτης επιλέγει να ξεκινήσει μια νέα μεταβατική κατάσταση.

Το REST διευκολύνει τη συναλλαγή μεταξύ των διακομιστών web, επιτρέποντας χαλαρή σύνδεση μεταξύ των διαφόρων υπηρεσιών. Το πρωτόκολλο rest υπερτερεί από το SOAP. Η γλώσσα του REST βασίζεται στη χρήση των ουσιαστικών και των ρημάτων, και δίνει έμφαση στην αναγνωσιμότητα. Σε αντίθεση με το SOAP, το REST δεν απαιτεί XML parsing και δεν απαιτεί μια κεφαλίδα μηνύματος από ένα φορέα παροχής υπηρεσιών.

4

WEB AND JAVASERVERFACES TECHNOLOGY

4.1 Γενικά

Η σύγχρονη τεχνολογία πλέον προσφέρει αμέτρητες δυνατότητες. Ολοένα και περισσότεροι προγραμματιστές κατευθύνονται προς την υλοποίηση κατακευκτών, εύκολα μεταφερόμενων και διαλειτουργικών εφαρμογών που θα προσφέρουν μεγαλύτερη ασφάλεια, ταχύτητα, ευκολία στη χρήση. Φυσικά μιλάμε για server sided enterprise εφαρμογές που παρέχουν το business logic. Το σημαντικό με τη χρήση αυτής της τεχνολογίας είναι η διαδραστικότητα με άλλες εφαρμογές.

Προκειμένου να αντιληφθούμε πως λειτουργεί μια Java Server Faces(JSF) πρέπει ο αναγνώστης να έχει μια βασική γνώση για το τι γίνεται στα παρασκήνια όταν ένας χρήστης κάνει μια αίτηση για μια ιστοσελίδα.

4.1.1 XHTML

Μια ιστοσελίδα στην ουσία δεν είναι τίποτα άλλο από ένα XHTML έγγραφο το οποίο περιγράφει το περιεχόμενο σε έναν web browser.Οι σελίδες αυτές οι XHTML συνήθως περιέχουν υπερσυνδέσεις για τη σύνδεση τους με άλλες ιστοσελίδες και άλλα μέρη της ίδιας ιστοσελίδας. Όταν ένας χρήστης κάνει κλικ πάνω στο σύνδεσμο πραγματοποιείται η φόρτωση της ιστοσελίδας.

4.1.2 URLs

Υπολογιστές που κάνουν χρήση λογισμικών βασισμένων σε τεχνολογία web server επιτρέπουν τη χρήση των πόρων όπως σελίδων, εικόνων, εγγράφων και αντικειμένων, για την πραγματοποίηση πολύπλοκων αναζητήσεων και συνδέσεων με τη βάση δεδομένων. Το πρωτόκολλο HTTP που χρησιμοποιείται επικοινωνεί με τους διάφορους web servers ώστε να γίνει η ανταλλαγή πληροφοριών με ένα κοινό τρόπο. Ο URL αναγνωρίζει την τοποθεσία στο διαδίκτυο από τους πόρους που αναφέραμε παραπάνω. Εάν γνωρίζεις το url μπορείς να έχεις πρόσβαση στο διαδικτυακό πόρο αυτό με χρήση του πρωτοκόλλου HTTP.

4.1.3 Διαδικασία αιτήματος απάντησης

Από τη στιγμή που δίνεται ένα url ο browser υλοποιεί μια σύνδεση με χρήση του πρωτοκόλλου HTTP προκειμένου να συνδεθεί με την ιστοσελίδα και να την προβάλει. Παρακάτω βλέπουμε ένα παράδειγμα

```
GET/professors/downloads.html HTTP/1.1
```

όταν όμως λέμε Get τι εννοούμε; Εννοούμε μια μέθοδο η οποία μας δείχνει πως ο πελάτης θέλει να αποκτήσει μια πηγή από ένα server. Το υπόλοιπο κομμάτι μας δίνει πληροφορίες για το path name της πηγής, το πρωτόκολλο και την έκδοση. Κάθε server που δέχεται το συγκεκριμένο πρωτόκολλο θα απαντήσει κατάλληλα στο αίτημα στέλνοντας τα απαραίτητα στοιχεία.

4.1.4 HTTP Headers

Ο server στέλνει μία ή και περισσότερες http headers οι οποίες στην ουσία παρέχουν επιπλέον πληροφορίες για τα δεδομένα τα οποία στάλθηκαν. Για παράδειγμα εάν ο server στείλει ένα HTML έγγραφο κειμένου η επικεφαλίδα θα είναι ως εξής:

```
Content-type:text/html
```

Οι πληροφορίες που περιλαμβάνονται στην επικεφαλίδα προσδιορίζουν το λεγόμενο MIME (Multipurpose Internet Mail Extensions) τύπο του περιεχομένου που μεταδίδει ο browser. MIME

είναι ένα πρότυπο του διαδικτύου το οποίο προσδιορίζει τα format των δεδομένων ώστε τα προγράμματα να μπορούν να διαχειρίζονται τα δεδομένα σωστά.

Η επικεφαλίδα ή το σύνολο των επικεφαλίδων ακολουθούνται από μια λευκή γραμμή η οποία στην ουσία ειδοποιεί στον browser του πελάτη ότι ο server σταμάτησε την αποστολή επικεφαλίδων. Στη συνέχεια, ο server στέλνει τα περιεχόμενα που ζητήθηκαν ο web browser εμφανίζει τα αποτελέσματα.

4.1.5 POST και GET αιτήματα

Τα πιο κοινά αιτήματα είναι το Get και το Post όμως πρέπει να διαχωρίσουμε τη λειτουργία τους. Με το get ζητάμε απλώς τους πόρους από ένα server όπως για παράδειγμα κάποιο έγγραφο ή μια εικόνα η τα αποτελέσματα από μία μηχανή αναζήτησης. Με το post το αίτημα είναι διαφορετικό, με αυτό το αίτημα μπορούμε να στείλουμε δεδομένα στο server. Μια πολύ συχνή χρήση λειτουργίας post είναι η αποστολή φορμών δεδομένων και εγγράφων στο server.

Όταν μια ιστοσελίδα περιέχει ένα HTML form όπου ο χρήστης μπορεί να εισάγει δεδομένα αυτό που συμβαίνει με αυτό το αίτημα είναι να κάνε post των δεδομένων σε server side form handler για επεξεργασία

Βέβαια και οι δύο μέθοδοι στέλνουν δεδομένα στον server απλά διαφέρει ο τρόπος αποστολής. Η μέθοδος get στέλνει πληροφορίες στο server στο url σε αντίθεση με τη μέθοδο post που στέλνει τα δεδομένα σαν μέρος ενός HTTP μηνύματος. Μια διαφορά των μεθόδων είναι τι η Post κρύβει τις υποσταλείσες πληροφορίες από το χρήστη.

4.1.6 Client side caching

Οι browsers συχνά αποθηκεύουν τις σελίδες για γρηγορότερη φόρτωση, το λεγόμενο cache. Εάν δεν υπάρχει διαφορά στην ήδη αποθηκευμένη και στην νέα, προς ζήτηση σελίδα, ο browser κάνει χρήση του αντιγράφου αυτής, για γρήγορη φόρτωση. Μια HTTP απόκριση μπορεί να υποδείξει το χρόνο που τα δεδομένα παραμένουν φρέσκα. Εάν αυτός ο χρόνος δεν έχει παρέλθει ο browser μπορεί να αποφύγει μια νέα αίτηση από τον server, αλλιώς ζητά το έγγραφο από τον server. Με αυτό το τρόπο ο Browser ελαχιστοποιεί το ποσοστό δεδομένων που πρέπει να φορτωθούν προκειμένου να εμφανιστεί μια ιστοσελίδα. Οι browsers τυπικά δεν αποθηκεύουν το αίτημα του server σε ένα post request γιατί υπάρχει περίπτωση το επόμενο post να μην επιστρέψει το ίδιο αποτέλεσμα.

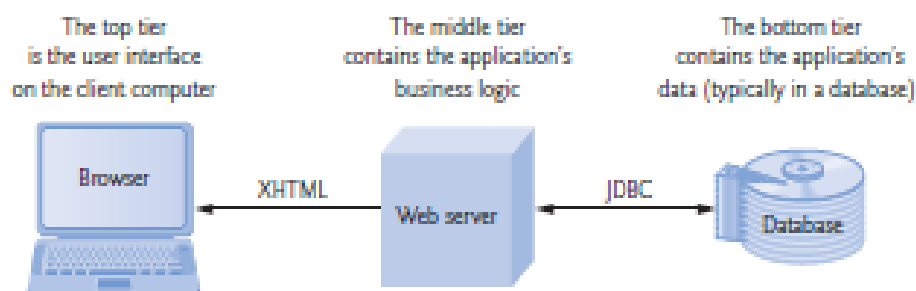
Όταν χρησιμοποιείται μια μηχανή αναζήτησης ο browser φυσιολογικά παρέχει τις πληροφορίες για που προσδιορίζονται σε μια φόρμα HTML σε ένα Get αίτημα. Αφού πραγματοποιηθεί η αναζήτηση και επιστρέφει τα αποτελέσματα σε μια ιστοσελίδα

Γιατί να χρησιμοποιήσουμε Asynchronous Javascript and XML (Ajax) στις εφαρμογές μας

Με το Ajax προστίθεται στις εφαρμογές ένα επιπλέον επίπεδο μεταξύ του client και του server προκειμένου να επιτευχθεί καλύτερη επικοινωνία μεταξύ τους. Όταν ο χρήστης χρησιμοποιεί μια ιστοσελίδα ο πελάτης δημιουργεί ένα αντικείμενο XML request το οποίον ελέγχει το αίτημα. Αυτό το αντικείμενο στέλνει το αίτημα στον server και περιμένει την απάντηση. Τα αιτήματα είναι ασύγχρονα έτσι ο χρήστης μπορεί να συνεχίσει να χρησιμοποιεί την εφαρμογή στην πλευρά του πελάτη ενώ ο server ασχολείται με την επεξεργασία του αιτήματος. Επιπλέον αιτήματα μπορούν να αποσταλούν στον server. Από τη στιγμή που ο server απαντήσει στο αρχικό μήνυμα το αντικείμενο XML που αναφέραμε καλεί τη λειτουργία του πελάτη που είναι υπεύθυνη να επεξεργάζεται τα δεδομένα που επέστρεψαν από τον server. Αυτή η λειτουργία που είναι γνωστή σαν callback function χρησιμοποιεί μερικώς αναβαθμισμένες σελίδες για να παρουσιάσει τα δεδομένα χωρίς να αναγκάζεται να φορτώνει όλη την σελίδα ξανά. Την ίδια στιγμή είναι εφικτό να εκτελείται και ένα δεύτερο αίτημα. Αποτέλεσμα είναι πιο λειτουργικές και γρήγορες εφαρμογές.

4.1.7 Multitier εφαρμογές

Οι διαδικτυακές υπηρεσίες είναι υπηρεσίες Multitier. Τι σημαίνει όμως αυτό; Multitier εφαρμογές σημαίνει ότι διαιρούν την λειτουργικότητα τους σε διάφορα tiers. Παρόλο που τα Tiers αυτά μπορούν να βρίσκονται όλα στον ίδιο υπολογιστή τα tiers των διαδικτυακών εφαρμογών μπορούν σε διάφορους υπολογιστές. Στο παρακάτω σχήμα μπορούμε να δούμε την αρχιτεκτονική αυτής της τεχνολογίας.



Σχήμα 3 Αρχιτεκτονική Multitier [Πηγή Java Programming Deitel 2011]

Παρακάτω αναλύονται τα τρία αυτά μέρη του συστήματος, που στην ουσία θα αποτελέσουν και τα μέρη του δικού μας συστήματος που θα υλοποιηθεί .

Το πρώτο μέρος είναι το information tier. Στην ουσία αποτελεί τα δεδομένα που πρόκειται να χρησιμοποιηθούν στην εφαρμογή. Αυτό το tier, είναι η βάση δεδομένων που θα χρησιμοποιηθεί για να έχει όλα τα στοιχεία των τουριστικών προτάσεων μας.

Το middle tier αποτελεί αυτό που λέμε το business logic, το controller logic και το presentation logic της εφαρμογής και στην ουσία αποτελούν τον ενδιάμεσο σταθμό και σύνδεσμο μεταξύ της εφαρμογής του χρήστη και των δεδομένων. Το controller logic επεξεργάζεται τα αιτήματα του χρήστη και τραβάει τα δεδομένα από τη βάση. Το presentation logic επεξεργάζεται αυτά τα δεδομένα που πήρε, ύστερα από το σχετικό αίτημα προς τη βάση και παρουσιάζει το περιεχόμενο στο χρήστη. Μια τυπική παρουσίαση που χρησιμοποιείται ευρέως είναι σε HTML document

Το business logic ενδυναμώνει τους επιχειρηματικούς κανόνες, διαβεβαιώνει τα δεδομένα πριν την εφαρμογή στο διακομιστή, αναβαθμίζει τη βάση και φορτώνει τα δεδομένα για παρουσίαση στο χρήστη. Αυτοί οι επιχειρηματικοί κανόνες κανονίζουν πως και αν οι πελάτες έχουν πρόσβαση στην υπηρεσία και πως θα γίνεται η επεξεργασία αυτών των δεδομένων.

Τέλος το client tier ή bottom tier που αποτελεί και το περιβάλλον χρήσης του χρήστη, το οποίο διαχειρίζεται τις εισαγωγές και παρουσιάζει τα αποτελέσματα. Οι χρήστες επικοινωνούν με την υπηρεσία κατευθείαν μέσω αυτού του περιβάλλοντος με το πληκτρολόγιο τους και το ποντίκι τους. Κάθε κίνηση τους έχει και το ανάλογο αντίκτυπο στο middle tier που είναι υπεύθυνο να ικανοποιήσει αυτό το αίτημα και στη συνέχεια να επικοινωνήσει με το information tier, τη βάση δηλαδή για να ανασύρει τα δεδομένα που θέλουμε. Το tier του πελάτη αναλαμβάνει να παρουσιάσει αυτά τα αποτελέσματα. Ποτέ δεν έρχεται σε επαφή ο πελάτης με τη βάση.

4.2 Java EE application model

Το μοντέλο της Java EE ξεκινά με το προγραμματισμό σε γλώσσα java και με τη χρήση virtual machine. Η φορητότητα, η ασφάλεια και η ανάπτυξη αποτελούν τα βασικά χαρακτηριστικά της Java EE. Η Java EE σχεδιάστηκε έτσι ώστε να υποστηρίζει εφαρμογές και υπηρεσίες οι οποίες είναι περίπλοκες και δίνουν τη δυνατότητα δημιουργίας κατανεμημένων συστημάτων με πρόσβαση στα δεδομένα από διάφορους υπολογιστές και πελάτες.

Ο καλύτερος έλεγχος και η διοίκηση επιτυγχάνεται σε αυτές τις εφαρμογές στο middle tier. Το middle tier τρέχει στον server και παρέχει δυνατότητες πλήρης πρόσβασης σε όλες τις υπηρεσίες της εφαρμογής.

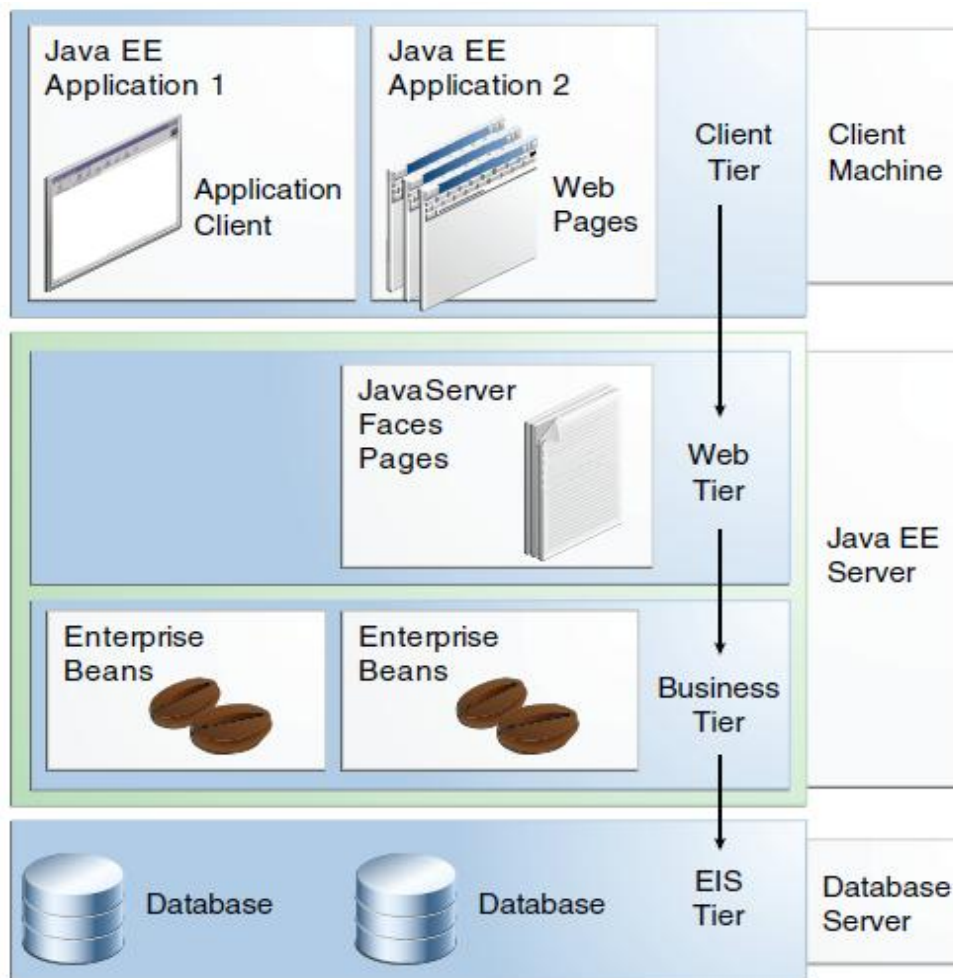
Αυτό το μοντέλο προσδιορίζει την αρχιτεκτονική για την δημιουργία τέτοιων εφαρμογών και υπηρεσιών τις λεγόμενες Multitier εφαρμογές που προσφέρουν πολύ καλή χρήση, πρόσβαση και δυνατότητες στις υπηρεσίες αυτές.

Το μοντέλο αυτό αποτελείται από δυο βασικά μέρη

- Το business και presentation logic που δημιουργείται από τον προγραμματιστή
- Οι στάνταρ υπηρεσίες του συστήματος που παρέχονται από την πλατφόρμα της Java EE

Όπως αναφέρθηκε οι εφαρμογές multitier αποτελούνται από τρία επίπεδα. Το application logic είναι χωρισμένο σε τμήματα αναλόγως τις λειτουργίες και τα στοιχεία της εφαρμογής και είναι εγκατεστημένα σε διάφορα μηχανήματα.

Τα tiers αυτά μπορεί να τρία ή τέσσερα. Γενικά θεωρούνται ως τρία ανάλογα και με τις τοποθεσίες που βρίσκονται. Οι τοποθεσίες αυτές μπορεί να είναι τα μηχανήματα του πελάτη, τα μηχανήματα του Java EE server και η βάση δεδομένων. Οι three tiered εφαρμογές που τρέχουν με αυτό τον τρόπο επεκτείνουν το two tiered client-server model τοποθετώντας ένα πολυνηματικό server εφαρμογών μεταξύ του πελάτη και της βάσης. Στην εικόνα 24 έχουμε την σχηματική αναπαράσταση των τριών επιπέδων που θα περιλαμβάνει και η εφαρμογή μας.



Εικόνα 35 Multitier model [Πηγή Oracle Java Tutorial 2011]

4.3 Λειτουργία εφαρμογής με τεχνολογία JSF

Ας δούμε λίγο τι θα περιλαμβάνει γενικά το πρόγραμμά μας για να εξηγήσουμε περιεκτικά τι σημαίνει το καθένα από αυτά.

Όταν ο χρήστης ζητήσει την εφαρμογή, ο browser θα αναζητήσει τη σελίδα jsf που περιλαμβάνει η εφαρμογή. Στην περίπτωση μας ονομάζεται `tourism_planner`. Στη συνέχεια, ο διακομιστής αφού πάρει το αίτημα το περνά στο λεγόμενο JSF web application framework για επεξεργασία. Αυτό το framework είναι διαθέσιμο στον Glassfish server που χρησιμοποιούμε για την εφαρμογή μας. Το framework αυτό περιλαμβάνει το faces servlet, το οποίο αποτελεί ένα λογισμικό, το οποίο τρέχει στον διακομιστή και επεξεργάζεται κάθε αίτημα για jsf pages, ώστε να επιστρέψει με το αποτέλεσμα από το αίτημα στο πελάτη.

Η εφαρμογή μας περιλαμβάνει σελίδες οι οποίες για κάθε μία υπάρχει το αντίστοιχο java source file και το αρχείο με το περιεχόμενο της σελίδας.

Όπως είπαμε η εφαρμογή μας στηρίζεται στα Facelets. Τα facelets είναι ένας συνδυασμός από XHTML markup και JSF markup. Η XHTML προσδιορίζει το περιεχόμενο της σελίδας, το οποίο βλέπουμε με τη χρήση του browser. Η XHTML διαχωρίζει τον τρόπο παρουσίασης ενός εγγράφου από τη δομή των δεδομένων του. Η παρουσίαση ενός εγγράφου μπορεί να προσδιορίζει τη θέση των στοιχείων της ιστοσελίδας, τη γραμματοσειρά, το χρώμα κ.α. Επιπρόσθετα, η παρουσίαση εμπλουτίζεται και διευκολύνεται με τη χρήση των CSS.

Με τη χρήση του NetBeans που όπως αναφέραμε πραγματοποιείται στην αρχή του προγράμματος το XML declaration που δείχνει την έκδοση XML που χρησιμοποιείται για την έκφραση των jsf αρχείων. Επίσης στην αρχή κάθε έγγραφου γίνεται και προσδιορισμός των XML namespaces που πρόκειται να χρησιμοποιηθούν στο έγγραφο και αυτό διακρίνεται από την ετικέτα

A small blue rectangular box with a white border containing the text "XMLNS:=h=" in red.

4.4 Δομή JSF Page

Γενικά ακολουθείται μια δομή που μοιάζει πολύ με χρήση απλής HTML. Τα δύο κυρίως στοιχεία που βρίσκουμε είναι το h:head και το h:body

Το h:head καθορίζει τη κεφαλή της σελίδας. Περιλαμβάνει κυρίως το τίτλο που πρόκειται να εμφανίζεται και στο παράθυρο του browser και είναι και ο τίτλος που θα χρησιμοποιούν οι μηχανές αναζήτησης. Επίσης μπορεί να περιέχει και meta στοιχείο που περιλαμβάνει και άλλα στοιχεία της ιστοσελίδας όπως για παράδειγμα το χρόνο ανανέωσης της σελίδας.

Το h:body περιλαμβάνει όλα τα δεδομένα της σελίδας. Μπορούμε εύκολα να χρησιμοποιήσουμε όλες τις γνωστές ετικέτες της HTML καθώς επίσης και άλλες τεχνολογίες όπως JavaScript, Ajax, CSS κ.α.

4.5 Classes JSF

Στην υλοποίηση της εφαρμογής μας θα χρησιμοποιηθούν διάφορα αντικείμενα για να μπορέσει να τρέξει η εφαρμογή μας. Ας δούμε λίγο συνοπτικά τι περιλαμβάνει.

4.5.1 Java Bean

Τα αντικείμενα Java Beans αποτελούν στιγμιότυπα των κλάσεων. Όπως θα δούμε και παρακάτω στο σχεδιασμό της εφαρμογής κάθε JavaBean περιλαμβάνει δεδομένα και μεθόδους. Το κάθε Java Bean προσφέρει τα δεδομένα του, θα λέγαμε, στο jsf document, σαν ιδιότητες που μπορεί να είναι τροποποιήσιμες ή όχι. Είναι τα λεγόμενα getters setters που συναντάμε σχεδόν σε κάθε εφαρμογή.

4.5.2 Η κλάση TourismWizard

Η κλάση αυτή στην ουσία αποτελεί τη ραχοκοκαλιά του προγράμματος γιατί καθορίζει τον οδηγό που θα συγκεντρώνει τα επιθυμητά στοιχεία και θα λαμβάνει από τον server τις προτάσεις ταξιδιού. Η κλάση αυτή θα περιλαμβάνει τον αλγόριθμο αναζήτησης και την ιδιότητα get Result για να παίρνει τα αποτελέσματα που θέλουμε.

4.5.3 ManagedBean Annotation

Θα συναντήσουμε το λεγόμενο @ManagedBean annotation το οποίο στην ουσία υποδεικνύει στο Jsf Framework ότι πρέπει να δημιουργήσει και να χειριστεί όλα τα αντικείμενα του Tourism Wizard.

4.6 Αρχιτεκτονική JSF

4.6.1 MODEL-VIEW-CONTROLLER

Οι εφαρμογές με τεχνολογία JSF ακολουθούν το μοντέλο MVC (Model View Controller) το οποίο διαχωρίζει τα δεδομένα της εφαρμογής από τα γραφικά της παρουσίασης και τη λογική επεξεργασίας όπως βλέπουμε στην εικόνα 25.



Εικόνα 36 MVC Model [Πηγή Java Programming Deitel 2011]

Στα JSF, ο controller είναι υπεύθυνος για την καθοδήγηση και την διαδραστικότητα μεταξύ της όψης και του μοντέλου. Το μοντέλο περιλαμβάνει τα δεδομένα της εφαρμογής, δηλαδή τη βάση μας και η όψη αφορά την παρουσίαση των δεδομένων, δηλαδή τις ιστοσελίδες μας. Όταν ένας χρήστης χρησιμοποιεί, απλά τη σελίδα, στη συνέχεια η σελίδα αυτή επικοινωνεί με το μοντέλο για αποθήκευση και εξόρυξη των δεδομένων. Όταν το μοντέλο αλλάζει η όψη αναβαθμίζεται με τα καινούργια δεδομένα

Στην εφαρμογή μας είναι σημαντικό να χρησιμοποιήσουμε τεχνολογίες session tracking. Όπως θα δείτε θα πραγματοποιηθεί χρήση του @SessionScoped Bean το οποίο θα βοηθήσει να διατηρηθούν οι επιλογές του χρήστη για τις προτιμήσεις του καθ όλη τη διάρκεια που χρησιμοποιεί την ιστοσελίδα. Αυτό το Bean δημιουργείται όταν ξεκινά η συνεδρία και υπάρχει για όλη τη διάρκεια της. Ένα SessionScoped Bean μπορεί να έχει πρόσβαση από όλες τις ιστοσελίδες και διατηρεί τη συνεδρία ο server.

4.7 Java EE και Java Server Faces

Τα java server faces όπως αναφέρθηκε, είναι μια τεχνολογία για ανάπτυξη του user interface διαδικτυακών εφαρμογών. Τα κυριότερα μέρη του είναι

- A GUI component framework.

Ένα ευέλικτο μοντέλο για την ενσωμάτωση διαφόρων τεχνολογιών και γλωσσών προγραμματισμού.

- Ένα RenderKit για HTML/4.01

Τα παρακάτω χαρακτηριστικά υποστηρίζουν τα στοιχεία του GUI.

- Input validation
- Event handling
- Data conversion between model objects and components
- Managed model object creation
- Page navigation configuration
- Expression Language (EL)

Στην Java EE 6 platform, τα νέα χαρακτηριστικά των JavaServer Faces περιλαμβάνουν τα παρακάτω:

- Τη δυνατότητα να χρησιμοποιούν annotations αντί ενός configuration αρχείου.
- Facelets, που αντικατέστησαν την τεχνολογία των JSP και χρησιμοποιούν την XHTML που αναφέρθηκε παραπάνω.
- Υποστήριξη Ajax
- Composite components
- Αποκλειστική καθοδήγηση

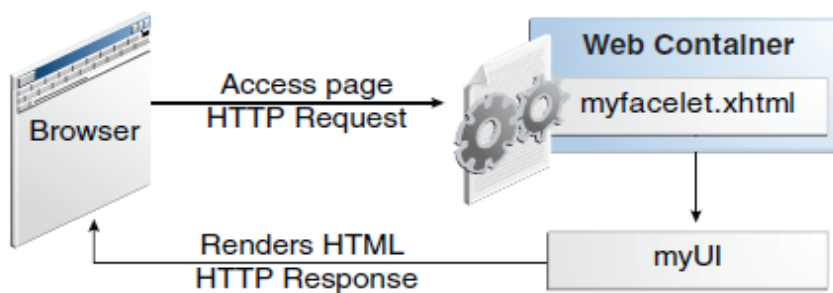
Η Java EE 6 platform χρειάζεται JavaServer Faces 2.0 και Expression Language 2.2.

Η λειτουργικότητα που παρέχουν τα java server faces είναι παρόμοια με τις άλλες εφαρμογές της Java. Μια τυπική εφαρμογή των Java Server Faces περιλαμβάνει τα κάτωθι:

- Ένα σύνολο από σελίδες
- Ένα σύνολο από ετικέτες για την ενσωμάτωση διάφορων χαρακτηριστικών
- Ένα σύνολο από managed beans τα οποία έχουν ελάχιστες απαιτήσεις και περιέχουν κάποιες βασικές υπηρεσίες του διαδικτύου.

- Ένα web deployment descriptor (web.xml file)
- Προαιρετικά ένα ή περισσότερα από configuration resource files όπως το faces-config.xml το οποίο μπορεί να χρησιμοποιηθεί για να καθορίσει τους κανόνες καθοδήγησης μεταξύ των διάφορων ιστοσελίδων και άλλων αντικειμένων
- Προαιρετικά, διάφορα άλλα αντικείμενα όπως μετατροπείς, listeners κ.α
- Διάφορες άλλες ετικέτες που δημιουργήθηκαν από τον προγραμματιστή.

Στην εικόνα 26 βλέπουμε πως αντιδρούν ο πελάτης και ένας διακομιστής με την τεχνολογία JavaServer Faces. Σε απάντηση δίνεται η ιστοσελίδα



Εικόνα 37 JavaServer Faces εφαρμογή[Πηγή Java EE Tutorial 2012]

4.8 Java EE Components

4.8.1 Γενικά

Με την πλατφόρμα Java Platform, Enterprise Edition (Java EE), η ανάπτυξη εφαρμογών σε Java έχει γίνει ευκολότερη και ταχύτερη. Σκοπός της είναι να παρέχει στους προγραμματιστές διάφορα χρήσιμα APIs για τη γρηγορότερη και λιγότερο πολύπλοκη ανάπτυξη εφαρμογών με βελτιωμένες αποδόσεις. Η πλατφόρμα Java EE αναπτύσσεται μέσω κοινοτικής διαδικασίας Java (η JCP), που είναι υπεύθυνη για όλες τις τεχνολογίες Java. Ομάδες εμπειρογνομώνων, αποτελούμενη, δημιούργησαν την Java Specification Requests (JSRs) για να ορίσουν τις διάφορες τεχνολογίες Java EE. Το έργο της Κοινότητας Java στο πλαίσιο του προγράμματος JCP βοηθά στην εξασφάλιση της σταθερότητας και της συμβατότητας της java στις διάφορες εφαρμογές. Η πλατφόρμα Java EE χρησιμοποιεί ένα απλουστευμένο μοντέλο προγραμματισμού. Περιγραφές ανάπτυξης της XML είναι προαιρετικές. Αντίθετα, ο προγραμματιστής απλώς μπορεί να εισάγει τις πληροφορίες ως ένα σχόλιο απευθείας σε ένα αρχείο προέλευσης δεδομένων της Java και ο διακομιστής της Java EE θα ρύθμιση τις παραμέτρους για την ανάπτυξη και την εφαρμογή.

Η Java EE platform περιλαμβάνει

- ❖ Java API for Restful Web Services
- ❖ Managed Beans
- ❖ Contexts and Dependency Injection for the Java

<http://www.oracle.com/technetwork/java/javaee/documentation/>

4.8.2 Java EE εφαρμογές

Οι εφαρμογές Java EE αποτελούνται από κάποια συστατικά όπως τα ονομάζουμε. Κάθε Java EE component θεωρείται αυτόνομο λογισμικό το οποίο ενσωματώνεται στην εφαρμογή και περιλαμβάνει τις κλάσεις του και τα αρχεία του, ώστε να επικοινωνεί και με τα άλλα components. Η Java EE περιλαμβάνει τα ακόλουθα Java EE components.

- Application clients και applets που τρέχουν στην πλευρά του πελάτη
- Java Servlet, JavaServer Faces και JavaServer Pages (JSP) που τρέχουν στον server.
- Enterprise JavaBeans (EJB) που τρέχουν και αυτά στην πλευρά του server.

4.8.3 Java EE Clients

Ένας Java EE πελάτης είναι είτε ένας web client ή ένας application client.

4.8.3.1 Web Clients

Ένας web client αποτελείται από δύο μέρη:

- Από δυναμικές ιστοσελίδες που περιλαμβάνουν διάφορες γλώσσες διαδικτυακού προγραμματισμού Dynamic (HTML, XML), οι οποίες τρέχουν στο web tier.
- Έναν browser, ο οποίος παρουσιάζει τις ιστοσελίδες που παίρνει από τον server.

Ένας web client καλείται *thin client*. Thin clients δεν ασχολούνται με τις βάσεις δεδομένων, ούτε με περίπλοκους κανόνες ή συνδέσεις. Σαν thin client, όλες οι δύσκολες διαδικασίες εκτελούνται στα enterprise beans στον Java EE server, ο οποίος και ασχολείται με θέματα ασφάλειας, ταχύτητας, και διαθεσιμότητας της υπηρεσίας.

4.8.3.2 Application Clients

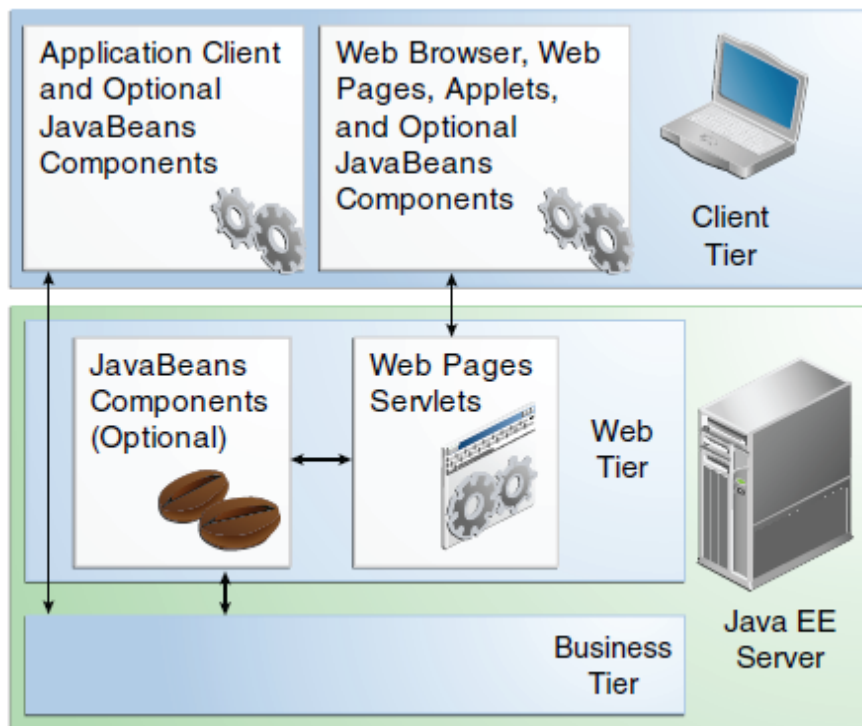
Μια εφαρμογή πελάτη τρέχει στο μηχάνημα του πελάτη και παρέχει ένα τρόπο στους χρήστες να έχουν ένα περιβάλλον χρήση πολύ πιο πλούσιο σε χαρακτηριστικά από ότι μπορούν να προσφέρουν οι απλές γλώσσες. Ένας πελάτης εφαρμογής έχει το λεγόμενο graphical user interface (GUI) το οποίο δημιουργείται με τα API των SWING και ABSTRACT WINDOW TOOLKIT.

Ένας πελάτης εφαρμογής έχει πρόσβαση κατευθείαν στα enterprise beans στο business tier. Παρόλα αυτά ο πελάτης μπορεί να ανοίξει και μια HTTP σύνδεση προκειμένου να επιτευχθεί μια επικοινωνία με ένα servlet στο web tier.

4.8.4 Web Components

Τα Web Components είναι τα servlets ή οι ιστοσελίδες που δημιουργήθηκαν με τη χρήση των Java Server Faces, τεχνολογία που θα χρησιμοποιηθεί και στην εφαρμογή μας. Πιο συγκεκριμένα, τα servlets είναι κλάσεις γραμμένα σε Java που επεξεργάζονται δυναμικά τα αιτήματα και παράγουν

απαντήσεις. Τα Java Server Faces είναι χτισμένα πάνω στα servlets και στα Java Server Pages και παρέχουν το περιβάλλον του χρήστη στις διάφορες εφαρμογές. Στατικές ιστοσελίδες και κλάσεις που τρέχουν στην πλευρά του server μπορούν να ενσωματωθούν στις διάφορες εφαρμογές αλλά δεν αποτελούν enterprise components.



Εικόνα 38 Web Components [Πηγή Java EE Tutorial 2012]

4.8.5 Java EE Application Assembly and Deployment

Αφού αναλύθηκε το τι περιλαμβάνει και πως είναι δομημένη μια εφαρμογή κρίνεται απαραίτητο να αναλυθεί και πως αυτή η εφαρμογή θα υλοποιηθεί και θα γίνει πράξη. Κάθε εφαρμογή είναι διαχωρισμένη σε ενότητες. Κάθε ενότητα περιλαμβάνει:

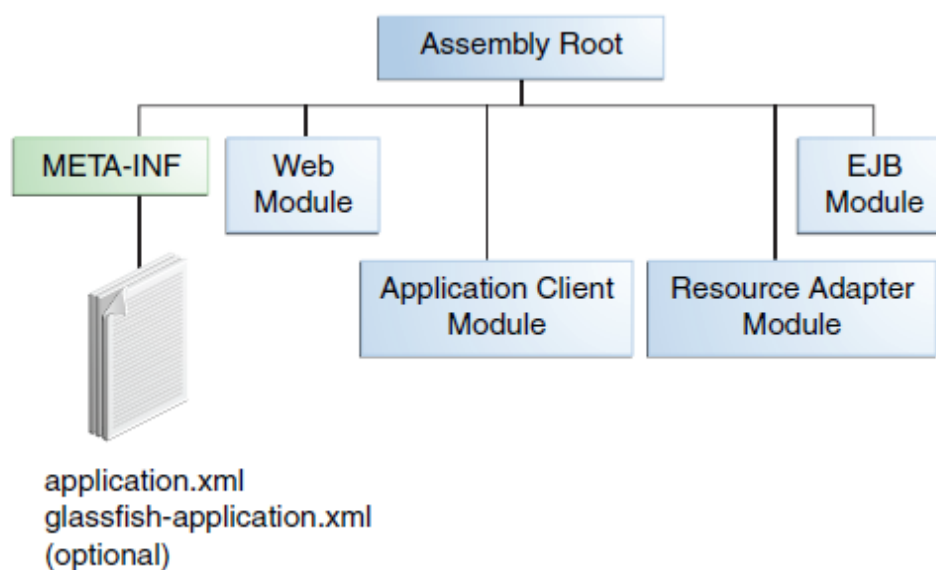
- Ένα λειτουργικό στοιχείο όπως ένα enterprise bean ή μια ιστοσελίδα ή ένα
- Έναν deployment descriptor που περιγράφει το περιεχόμενο.

Από τη στιγμή που μια μονάδα Java EE έχει παραχθεί, είναι έτοιμη για το λεγόμενο deployment. Το deployment περιλαμβάνει τη χρήση μιας πλατφόρμας σαν εργαλείο για να προσδιοριστούν πληροφορίες όπως τα ονόματα των χρηστών που έχουν δικαίωμα εισόδου καθώς και το όνομα της βάσης. Από τη στιγμή που ολοκληρωθεί η εφαρμογή είναι έτοιμη να τρέξει.

Κάθε εφαρμογή παραδίδεται σαν ένα αρχείο Java Archive (JAR) ή σαν Web Archive (WAR) ή σαν Enterprise Archive (EAR). Με τη χρήση τέτοιων αρχείων γίνεται εύκολη η ενσωμάτωση διαφόρων εφαρμογών προκειμένου να γίνεται εύκολη η χρήση ίδιων στοιχείων από άλλες εφαρμογές χωρίς την ανάγκη να γραφτεί επιπλέον κώδικας. Εδώ στηρίζεται και όλη η φιλοσοφία των web services.

Στις διαδικτυακές εφαρμογές θα συναντήσουμε διάφορα module τα οποία αποτελούν το σύνολο των κλάσεων και αρχείων μας. Τα modules αυτά είναι τα κάτωθι και θα αποτελούν μέρη και της δικιά μας εφαρμογής:

- EJB modules, τα οποία περιλαμβάνουν τις κλάσεις των enterprise beans και τον EJB deployment. Τα module αυτά θα είναι JAR αρχεία με προέκταση .jar.
- Web modules, τα οποία θα περιλαμβάνουν τις κλάσεις του servlet, τα αρχεία web, εικόνες, HTML αρχεία και έναν web application deployment descriptor. Τα αρχεία αυτά θα τα συναντήσουμε με προέκταση .war (web archive) σαν JAR αρχεία.
- Application client modules, τα οποία περιλαμβάνουν τις κλάσεις της εφαρμογής και έναν application client deployment descriptor. Τα αρχεία αυτά θα έχουν .jar προέκταση.
- Resource adapter modules, τα οποία περιλαμβάνουν όλες τα interface της Java που θα χρησιμοποιηθούν, τις βιβλιοθήκες, τις κλάσεις και άλλα έγγραφα.



Εικόνα 39 Assembly root [Πηγή Java EE Tutorial 2012]

4.9 Συμπεράσματα

Αναφερθήκαμε στην τεχνολογία του διαδικτύου και πως λειτουργεί μια υπηρεσία με τη χρήση νέων τεχνολογιών. Όπως διαπιστώσαμε και παραπάνω όλες οι διαδικτυακές εφαρμογές ακολουθούν νέα πρότυπα και νέες τεχνολογίες. Όμως προκειμένου να κατανοήσουμε την χρήση των νέων αυτών γλωσσών πρέπει να έρθουμε σε επαφή με τη τεχνολογία που βρίσκεται πίσω από αυτές. Στα πανεπιστήμια δεν συνηθίζεται να διδάσκεται η enterprise έκδοση αλλά πιο πολύ γίνεται αναφορά στην standard έκδοση, που όμως δεν χρησιμοποιείται στις διαδικτυακές εφαρμογές. Σίγουρα θα πρέπει να υπάρχει και η αντίστοιχη διδασκαλία στα πανεπιστήμια και να συμπεριληφθούν μαθήματα που θα αφορούν web services και Java EE δεδομένου ότι πλέον υπάρχει μεγάλη ανάπτυξη του διαδικτυακού προγραμματισμού και ολόένα και περισσότερες υπηρεσίες και εφαρμογές προσανατολίζονται στο παγκόσμιο ιστό. Οι σπουδαστές θα πρέπει να αναζητήσουν την γνώση και να την ολοκληρώσουν γύρω από το αντικείμενο τους προκειμένου να επιτύχουμε ανάπτυξη και στην Ελλάδα.

5

Η ΑΝΑΠΤΥΞΗ ΤΩΝ ΟΝΤΟΛΟΓΙΩΝ

5.1 Γενικά στοιχεία για τις οντολογίες

5.1.1 Τι είναι οι οντολογίες

Ο όρος οντολογία, προέρχεται από τη γενική του ουσιαστικού όν που σημαίνει ύπαρξη, πλάσμα. Είναι ευνόητο, επομένως, πως ο κλάδος της φιλοσοφίας που ασχολείται με τη μελέτη της ύπαρξης, θα φέρει αυτό το όνομα.

Η έννοια της οντολογίας υιοθετήθηκε από την τεχνολογία της τεχνητής νοημοσύνης. Η οντολογία, στα πλαίσια της τεχνητής νοημοσύνης και γενικότερα στην επιστήμη των υπολογιστών, έχει σαφώς διαφορετική διάσταση και χρησιμότητα από αυτή που έχει στη φιλοσοφία.

Ο πρώτος ορισμός που θα δοθεί αποδίδει την έννοια του όρου οντολογία χωρίς να τον προσεγγίζει από τη μεταφυσική σκοπιά, αλλά και χωρίς να τον συνδέει άμεσα με την επιστήμη των υπολογιστών.

«Οντολογία» μπορεί να οριστεί ως το σύνολο των ρητών προσδιορισμών εννοιών και των συσχετίσεών τους στο πλαίσιο ύπαρξής τους- «An ontology is a specification of a conceptualization» [Gruber,1993].

Ένας ορισμός που θα μπορούσε να δοθεί για την οντολογία και που δηλώνει το συσχετισμό το πεδίο της επιστήμης των υπολογιστών είναι ο ακόλουθος.

Οντολογία είναι η θεωρία που αφορά στα είδη των οντοτήτων και ειδικότερα των αφηρημένων εννοιών που γίνονται δεκτές σε μία γλώσσα υπολογιστικού συστήματος [Webster,2002].

Ένας άλλος ορισμός που περιγράφει την έννοια της οντολογίας στην επιστήμη των υπολογιστών και παράλληλα εξηγεί και τη χρησιμότητα των οντολογιών είναι ο ακόλουθος.

Χρήση μεθόδων δημιουργίας οντολογιών από την επιστήμη των πληροφοριών ως βάση για την οργάνωση της επιστημονικής γνώσης πεδίων που χρησιμοποιούν διαφορετικούς όρους, διαφορετικές έννοιες, διαφορετικό λεξιλόγιο [Κόκλα, 2009].

Επομένως, γίνεται σαφές ότι οι οντολογίες είναι απαραίτητες για την ανάπτυξη κάθε σύνθετης εφαρμογής, ώστε να γίνει πλήρης τεκμηρίωση και επεξήγηση των υπηρεσιών ταξιδιού μέσω οντολογιών

Επιπλέον, οι οντολογίες καθίστανται ιδιαίτερες χρήσιμες στα πλαίσια της διαλειτουργικότητας, καθώς με την πάροδο του χρόνου προκύπτει ολοένα και μεγαλύτερη ανάγκη για ανταλλαγή και επαναχρησιμοποίηση δεδομένων.

5.1.2 Τα συστατικά των οντολογιών και τα είδη τους

Οι οντολογίες μπορεί να απαρτίζονται από τα εξής συστατικά μέρη:

- Οντότητες ή έννοιες, όπου αποτελούν τα θεμελιώδη εκείνα στοιχεία η σημασία των οποίων πρέπει να περιγραφεί και να αποσαφηνιστεί
- Ιδιότητες, οι οποίες χαρακτηρίζουν τις οντότητες
- Σχέσεις, αναφέρονται στις συσχετίσεις μεταξύ των οντοτήτων και των χαρακτηριστικών τους
- Αξιώματα, χρησιμοποιούνται για τη διασφάλιση της πληρότητας των οντοτήτων (π.χ. περιορισμοί, κανόνες)
- Περιπτώσεις, παραδείγματα που αντιστοιχούν στις οντότητες

Οι οντολογίες μπορεί να περιλαμβάνουν όλα τα προαναφερθέντα μέρη ή κάποια από αυτά ανάλογα τη φύση τους και το σκοπό που εξυπηρετούν.

Όσο αφορά στα είδη των οντολογιών, υπάρχουν πάρα πολλά και διαφέρουν μεταξύ τους, στη παρούσα εργασία θα αναφερθούν και θα περιγραφούν συνοπτικά τα 4 κυριότερα είδη [Τομαή,2005].

- Οι υψηλού επιπέδου οντολογίες, που προσφέρουν γενικές έννοιες με τις οποίες συνδέονται όλοι οι όροι των οντολογιών
- Οι οντολογίες πεδίου, που δημιουργούν λεξιλόγιο για τις έννοιες, τις σχέσεις και ότι αφορά ένα συγκεκριμένο πεδίο γνώσης
- Οι οντολογίες έργου, που προσφέρουν λεξιλόγιο προκειμένου να επιλυθούν προβλήματα σε συγκεκριμένες υποθέσεις εργασίας.
- Οι οντολογίες εφαρμογής, που χρειάζονται για τη μοντελοποίηση μιας συγκεκριμένης εφαρμογής

Όσο αφορά την ιεραρχία αυτών των ειδών οντολογιών είναι σχεδόν προφανής, οι οντολογίες υψηλού επιπέδου βρίσκονται στο πρώτο επίπεδο καθώς είναι οι πιο γενικές, οι οντολογίες πεδίου και έργου βρίσκονται στο επόμενο επίπεδο και στο τρίτο επίπεδο βρίσκονται οι οντολογίες εφαρμογής καθώς είναι πολύ συγκεκριμένες.

5.1.3 Αρχές ανάπτυξης οντολογιών

Αφού καθορισθούν όλα τα προηγούμενα, κατά τη δόμηση των οντολογιών χρειάζεται να ακολουθούνται κάποιες γενικές αρχές [Τομαή,2005].

- Σαφήνεια και αντικειμενικότητα
- Πληρότητα
- Συνοχή
- Μέγιστη επεκτασιμότητα
- Ελάχιστες οντολογικές δεσμεύσεις
- Οντολογικός διαχωρισμός
- Διαφοροποίηση των ιεραρχιών
- Συναρμολογησιμότητα
- Ελαχιστοποίηση της σημασιολογικής απόστασης μεταξύ των συγγενών συλλογισμών
- Τυποποίηση των ονομάτων

5.2. Περιβάλλον Ανάπτυξης

Το Protégé αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης και επεξεργασίας οντολογιών βασισμένο σε Java, με σκοπό τη δημιουργία οντολογιών και τη μοντελοποίηση γνώσης. Η εφαρμογή αναπτύσσεται με ταχύτατους ρυθμούς και χρησιμοποιείται για την ανάπτυξη κάθε είδους οντολογίας. Οι χρήστες του Protégé προσφέρουν τις γνώσεις τους σε μια on-line κοινότητα, που έχει δημιουργηθεί για την εξάπλωση της γνώσης και την αντιμετώπιση προβλημάτων από τους χρήστες.

Η πλατφόρμα του Protégé υποστηρίζει δυο μεθόδους μοντελοποίησης οντολογιών:

- Ο Protégé Frames editor [Protégé,2010] δίνει τη δυνατότητα στους χρήστες να δημιουργούν και να αποθηκεύουν οντολογίες που βασίζονται στη μεθοδολογία των πλαισίων. Το μοντέλο γνώσης που υλοποιεί είναι συμβατό με το Open Knowledge Base Connectivity protocol (OKBC).
- Ο Protégé – OWL editor [Protégé,2010] είναι μια επέκταση του Protégé που υποστηρίζει τη Web Ontology Language (OWL). Η OWL είναι η πιο πρόσφατα αναπτυγμένη γλώσσα από τις γνωστές γλώσσες οντολογίας και στηρίζεται από το World Wide Web Consortium (W3C). Η OWL περιλαμβάνει κλάσεις, τις ιδιότητές τους και περιπτώσεις. Με τη σημασιολογία της γλώσσας αυτή καθορίζεται ο τρόπος εξαγωγής λογικών συμπερασμάτων, τα οποία δεν παρουσιάζονται άμεσα στην οντολογία αλλά συνεπάγονται από τη σημασιολογία.

5.3 Η γλώσσα OWL

Η γλώσσα OWL αναπτύχθηκε από το World Wide Web Consortium (W3C) ως μια γλώσσα για τον ορισμό και την υλοποίηση οντολογιών στο διαδίκτυο. Η OWL διευκολύνει την διαλειτουργικότητα του περιεχομένου του δια-δικτύου που μπορεί να υποστηριχθεί από την XML, το RDF και το RDF Schema (RDF-S) προσφέροντας περαιτέρω λεξιλόγιο μαζί με τυπική σημασιολογία. Η OWL έχει ως σκοπό να δώσει μια πλήρη γλώσσα που να μπορεί να χρησιμοποιηθεί για να περιγράψει κλάσεις και σχέσεις μεταξύ αυτών που υπάρχουν σε κείμενα και εφαρμογές τους Δια-δικτύου [Τομαή,2005].

Η γλώσσα OWL παρέχει 3 υπογλώσσες προκειμένου να εξυπηρετήσει διαφορετικές σχεδιαστικές ανάγκες, ανάλογα τους χρήστες και τους σχεδιαστές που τη χρησιμοποιούν [W3C,2004].

- Η OWL Lite υποστηρίζει αυτούς τους χρήστες που πρωτίστως χρειάζονται μια ιεραρχία ταξινόμησης και απλούς περιορισμούς. Η Owl Lite έχει τον μικρότερο βαθμό πολυπλοκότητας από τα άλλα δύο είδη της γλώσσας .
- Η OWL DL (Description Logics) υποστηρίζει αυτούς τους χρήστες που θέλουν τη μέγιστη εκφραστικότητα ενώ συγχρόνως την διατήρηση της υπολογιστικής πληρότητας και αποφασιστικότητας. Η OWL DL περιλαμβάνει όλες τις δομές της γλώσσας OWL αλλά μπορεί να χρησιμοποιηθεί μόνο κάτω από συγκεκριμένους περιορισμούς. Η OWL DL ονομάζεται έτσι λόγω της αντιστοιχίας της με την Description Logics, ένα πεδίο έρευνας από το οποίο πηγάζει η λογική που διαμόρφωσε την θεωρητική βάση της OWL.
- Η OWL Full είναι φτιαγμένη για τους χρήστες που θέλουν την μέγιστη εκφραστικότητα και την συντακτική ελευθέρια της RDF χωρίς όμως διαβεβαιώσεις για την υπολογιστική ικανότητα. Η OWL Full επιτρέπει σε μία οντολογία να αυξήσει το νόημα του ήδη ορισμένου λεξιλογίου σε RDF ή OWL.

6

ΑΠΑΙΤΗΣΕΙΣ ΤΗΣ ΥΠΗΡΕΣΙΑΣ

6.1 Γενικά

Όπως είπαμε σκοπός μας είναι ο σχεδιασμός μιας υπηρεσίας διαδικτύου με σκοπό την δημιουργία ενός σχεδιαστή τουρισμού. Η υπηρεσία αυτή αποσκοπεί στην ενσωμάτωση διαφόρων εφαρμογών και τεχνολογιών συμβάλλοντας έτσι στην διαλειτουργικότητα των συστημάτων και την επίτευξη της ενοποίησης των συστημάτων και εφαρμογών. Έχουμε αναφέρει και σε προηγούμενες εργασίες ότι πλέον η κατεύθυνση προγραμματιστών είναι προς την υλοποίηση εφαρμογών που δίνουν τη δυνατότητα μέσα από ένα φιλικό περιβάλλον ο χρήστης να αναζητεί δεδομένα από διάφορες εφαρμογές και διάφορες βάσεις δεδομένων. Οι υπηρεσίες αυτές μπορούν να επεξεργαστούν όλα τα δεδομένα απομακρυσμένα χωρίς την ανάγκη επιπλέον λογισμικού και γνώσης από το χρήστη. Σημαντικό στοιχείο στις διαδικτυακές υπηρεσίες είναι η επεκτασιμότητα και η προσαρμοστικότητα σε διάφορες τεχνολογίες.

6.2 Προσανατολισμός Υπηρεσίας

Η υπηρεσία απευθύνεται σε όσους επιθυμούν εύκολα και γρήγορα να έχουν μια συνολική εικόνα από τα αξιοθέατα (μουσεία, αρχαιολογικοί χώροι) που υπάρχουν στην πόλη επιλογής τους με δυνατότητα παροχής χρήσιμων πληροφοριών όπως ωράρια λειτουργίας, τιμές εισιτηρίων, διεύθυνση, κατευθυντήριες οδηγίες και παροχή χάρτη καθώς επίσης και πληροφορίες διασκέδασης όπως προτεινόμενα εστιατόρια, κέντρα διασκέδασης, ταβέρνες κ.α

Απώτερος σκοπός της υπηρεσίας αυτής είναι και η ενσωμάτωση άλλων υπηρεσιών όπως κρατήσεων εισιτηρίων είτε αυτό αφορά πολιτιστικές εκδηλώσεις ή αεροπορικά, τρένων κτλ, εύρεσης ξενοδοχείων αλλά και άλλων υπηρεσιών όπως σχεδιαστές ταξιδιών με MMM.

6.3 Προτεινόμενο σύστημα για υλοποίηση

Η υπηρεσία επειδή απευθύνεται σε ένα τουρίστα ο οποίος δεν θα είναι απαραίτητα αλλοδαπός αλλά ούτε και θα γνωρίζει απαραίτητα αγγλικά, η υπηρεσία προτείνεται να υλοποιηθεί στις βασικές ξένες γλώσσες όπως Αγγλικά, Γαλλικά, Γερμανικά και Ιταλικά. Εννοείται φυσικά ότι η κύρια γλώσσα υλοποίησης θα είναι η Ελληνική.

Ο τουρίστας που θα χρησιμοποιεί την υπηρεσία δεν θα χρειάζεται να γνωρίζει τίποτα ούτε να εγκαταστήσει κάποια εφαρμογή. Το μόνο που θα χρειάζεται να γνωρίζει είναι ποια πόλη και το τι θέλει να επισκεφτεί σ' αυτήν. Οι λειτουργίες της υπηρεσίας πρέπει να είναι απλές. Δεν χρειάζεται ο χρήστης να εμπλέκεται με περίεργες διεργασίες και μεγάλο όγκο δεδομένων ούτε με μακροχρόνιες διεργασίες εισαγωγής δεδομένων. Οι πληροφορίες πρέπει να είναι απλές γρήγορες και πολύ εύχρηστες.

Η τεχνολογία και η ανάπτυξη υπηρεσιών βασίζεται στα δεδομένα και στη δημιουργία μιας καλά σχεδιασμένης βάσης δεδομένων μετά από τη συλλογή των κατάλληλων δεδομένων είναι πολύ σημαντικό κομμάτι της ανάπτυξης και της σχεδίασης του όλου συστήματος. Ο μεγάλος όγκος δεδομένων και οι πολλοί χρήστες με δυνατότητα άμεσης και οργανωμένης παροχής πληροφορίας είναι το πιο σημαντικό κομμάτι για τον αναλυτή. Η γνώση που συλλέχτηκε και ενσωματώθηκε σε ένα σύστημα πρέπει να προσφέρεται οργανωμένα, φιλτραρισμένα και γρήγορα. Η συλλογή της γνώσης αποτελεί ένα μεγάλο στάδιο της εργασίας και γι' αυτό θα αφιερώσουμε ένα ξεχωριστό κεφάλαιο προκειμένου να δείξουμε πως δημιουργήθηκαν οι οντολογίες από την γνώση και πως αυτές οι οντολογίες θα χρησιμοποιούν σαν στιγμιότυπα για την δημιουργία της βάσης και των διαγραμμάτων κλάσεων του όλου συστήματος.

Το πώς θα υλοποιηθεί μια υπηρεσία και ποια θα είναι τα βήματα που θα ακολουθεί ο χρήστης αποτελεί μέρος της σχεδίασης και είναι θέμα του αναλυτή ο τρόπος που θα γίνεται. Σίγουρα όμως στόχος του αναλυτή είναι να καλύπτει όλα τα απαραίτητα στοιχεία για την παροχή ολοκληρωμένης γνώσης με εύκολες λειτουργίες και αναζητήσεις μικρής διάρκειας.

Τα εργαλεία που θα χρησιμοποιηθούν πρέπει να προσφέρουν ένα απλό και λειτουργικό περιβάλλον χρήσης. Ο τρόπος προβολής των δεδομένων και γενικά των πληροφοριών πρέπει να είναι οικείος, γι' αυτό το λόγο και στις περισσότερες εφαρμογές κάνουμε χρήση των υπηρεσιών που προσφέρει η Google ειδικότερα ότι αφορά τους χάρτες που χρησιμοποιούνται ευρέως από εκατομμύρια χρήστες ανά τον κόσμο. Επιπρόσθετα, η Google με την τεράστια βάση δεδομένων που προσφέρει και τις νέες υπηρεσίες όπως το Google places δίνει νέα ώθηση στην ανάπτυξη εφαρμογών στο διαδίκτυο και αποφορτίζει τους προγραμματιστές από το βάρος της διαχείρισης χωρικών δεδομένων και υλοποίηση γράφων και αλγορίθμων πολύπλοκων και χρονοβόρων.

6.4 Παροχή υπηρεσίας

Η υπηρεσία θα σχεδιαστεί σαν δικτυακή εφαρμογή προκειμένου να υλοποιείται μέσω του internet ώστε ο χρήστης να μπορεί εύκολα και γρήγορα χωρίς προαπαιτούμενα, όπως εγκατάσταση και δημιουργία λογαριασμού, να έχει πρόσβαση στην υπηρεσία. Η υπηρεσία θα υλοποιηθεί σε έναν απομακρυσμένο server ο οποίος και θα υποστηρίζει την υπηρεσία και ο χρήστης με την χρήση του domain name που θα του δοθεί θα καλεί την υπηρεσία. Στόχος είναι η εύκολη πρόσβαση χωρίς την ανάγκη χρήσης username και δημιουργίας λογαριασμών, προκειμένου να χρησιμοποιήσει την υπηρεσία. Έτσι αποφεύγουμε τις διαδικασίες αυθεντικότητας, πιστοποίησης και κρυπτασφάλισης και δίνουμε την ελευθερία σε όλους να χρησιμοποιούν την υπηρεσία χωρίς να δίνουν προσωπικά δεδομένα.

Επιπλέον, η εφαρμογή αυτή είναι ανεξάρτητη υπολογιστή. Η προτεινόμενη γλώσσα προγραμματισμού είναι η Java που είναι ανοιχτού κώδικα και παρέχει ένα μεγάλο πλήθος από βιβλιοθήκες, που μπορούν να χρησιμοποιηθούν, και πολύ καλή υποστήριξη. Πιο συγκεκριμένα, προτείνονται τα java server faces ώστε εύκολα να μπορεί να προσαρμοστεί και για άλλα λειτουργικά όπως android και Mac.

Η τεχνολογία σήμερα προχωρά με γρήγορα βήματα και εμείς πρέπει να την ακολουθούμε. Η χρήση των κινητών τηλεφώνων, των smartphone, των tablets και η χρήση γενικά του διαδικτύου αυξάνεται καθημερινά γι' αυτό και πρέπει η υπηρεσία μας να είναι διαθέσιμη σε όλες τις συσκευές. Η δυνατότητα επέκτασης και αναβάθμισης της υπηρεσίας πρέπει να αποτελεί κύριο συστατικό της σχεδίασης

Από οικονομικής πλευράς η υπηρεσία θα παρέχεται δωρεάν. Σε καμία περίπτωση δεν δύναται να έχει κερδοσκοπικό χαρακτήρα. Τα έξοδα για την υλοποίηση της εφαρμογής πρέπει να είναι ελάχιστα και όλα τα χρησιμοποιούμενα λογισμικά να είναι ελεύθερα. Όμως για να μπορεί να είναι βιώσιμη και επακτάσιμη μια υπηρεσία πρέπει να έχει κάποια έσοδα που εκτιμάται ότι θα είναι από διαφημίσεις ή με τη σύνδεση της με κάποια άλλη υπηρεσία με οικονομικό όφελος από την αύξηση επισκεψιμότητας της. Σίγουρα, μια πρόταση από τον Εθνικό Οργανισμό Τουρισμού για χρηματοδότηση της υπηρεσίας δεν θα ήταν καθόλου ευκαταφρόνητη.

7

ΛΟΓΙΣΜΙΚΑ ΚΑΙ ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ

7.1 Γενικά

Στο κεφάλαιο αυτό θα δούμε τα εργαλεία και τα λογισμικά που προτείνονται για τη σχεδίαση. Όλα τα προτεινόμενα εργαλεία είναι ανοιχτού κώδικα και ενδείκνυνται για την υλοποίηση διαδικτυακών εφαρμογών. Δεν θα δώσουμε αναλυτική περιγραφή των τεχνολογιών που θα χρησιμοποιηθούν απλώς κάποιες γενικές πληροφορίες για να φέρουμε πιο κοντά τον αναγνώστη στο να καταλάβει πως θα υλοποιηθεί η υπηρεσία. Για περισσότερες πληροφορίες για κάθε εργαλείο ο αναγνώστης θα μπορεί να ανατρέχει στα αντίστοιχα εγχειρίδια που θα αναφέρονται.

7.2 Java

Πρόκειται για μια γλώσσα προγραμματισμού ανοιχτού κώδικα που παρέχεται από την Oracle. Η γλώσσα αυτή χρησιμοποιείται πλέον ευρέως για την υλοποίηση των περισσότερων διαδικτυακών εφαρμογών στο μεγαλύτερο εύρος συσκευών. Υπάρχουν πάρα πολλές βιβλιοθήκες που μπορούν να χρησιμοποιηθούν ελεύθερα ενώ η παροχή βοήθειας με το javadoc κατά την υλοποίηση μιας εφαρμογής είναι ιδιαίτερα επαναστατική. Ο κώδικας εκτελείται μέσω του JVM (Java Virtual Machine). Ο compiler παρέχεται στο πακέτο JDK (Java Development Kit) και παράγει ένα ενδιάμεσο κώδικα, τον Java Bytecode ο οποίος μετασχηματίζεται σε εκτελέσιμη μορφή με τον compiler του JRE (Java Runtime Environment).

Η java αποτελεί μια αντικειμενοστραφή γλώσσα προγραμματισμού όπου όλες οι μεταβλητές της είναι τύπου object στα οποία και γίνονται αναφορές.

Για τη συγκεκριμένη εργασία θα εργαστούμε με την Java Enterprise Edition η οποία αποτελεί ένα υπερσύνολο της SE.

7.3 NetBeans IDE

Το NetBeans IDE είναι ένα περιβάλλον ανάπτυξης λογισμικού με τη χρήση διάφορων γλωσσών προγραμματισμού για Windows, Mac, Linux, and Solaris. Διαθέτει ένα ανοιχτού κώδικα ολοκληρωμένο σύστημα ανάπτυξης και μια πλατφόρμα εφαρμογών που επιτρέπει στους προγραμματιστές να δημιουργούν γρήγορα web, enterprise, desktop, and mobile εφαρμογές με τη χρήση Java platform Διαθέτει αρκετά plugins που βοηθούν στην ανάπτυξη εφαρμογών και υποστηρίζεται από μια κοινότητα που προσφέρει εκτενή βοήθεια με την παροχή κατάλληλου documentation και εκπαίδευσης. .

7.4 Apache subversion

Είναι το γνωστό svn και είναι ένα σύστημα διαχείρισης εκδόσεων λογισμικού. Μας παρέχει τη δυνατότητα να διατηρούνται διάφορες εκδόσεις αρχείων. Αποτελεί δηλαδή ένα λογισμικό κατηγορίας Software Configuration Management (SCM).

Τα repositories που χρησιμοποιεί μπορούν να είναι είτε βασισμένα στο σύστημα αρχείων είτε τοποθετείται σαν plugin και περισσότερες πληροφορίες μπορείτε να βρείτε στη διεύθυνση subversion.apache.org

7.5 GlassFish Server

Ο GlassFish είναι μια open-source server εφαρμογή η οποία ξεκίνησε από τη Sun Microsystems για την Java EE πλατφόρμα και τώρα υποστηρίζεται από την Oracle Corporation. Η υποστηριζόμενη έκδοση ονομάζεται Oracle GlassFish Server. Ο GlassFish είναι ελεύθερο λογισμικό, εκδίδεται με διπλή-άδεια σε δύο άδειες χρήσης ελεύθερου λογισμικού: το Common Development and Distribution License (CDDL) και το GNU General Public License (GPL), με την εξαίρεση classpath .

GlassFish είναι υλοποιεί την Java EE και ως εκ τούτου υποστηρίζει Enterprise JavaBeans, JavaServer Faces JMS, RMI, JavaServer Page, servlets, κλπ. Αυτό επιτρέπει στους

προγραμματιστές να δημιουργήσουν εφαρμογές των επιχειρήσεων που είναι φορητές και επεκτάσιμες και με ενσωμάτωση τεχνολογιών κληρονομιάς. Προαιρετικά συστατικά μπορούν επίσης να εγκατασταθούν για πρόσθετες υπηρεσίες.

Ο GlassFish βασίζεται σε πηγαίο κώδικα κυκλοφόρησε από τη Sun και Oracle Corporation 's . Χρησιμοποιεί ένα παράγωγο του Apache Tomcat ως servlet και ένα πρόσθετο στοιχείο το Grizzly για επεκτασιμότητα και την ταχύτητα.

7.6 Google Maps Api

Το Google Maps είναι μία υπηρεσία της Google, η οποία παρέχει χάρτες και δορυφορικές εικόνες ή ορθοφωτοχάρτες για το σύνολο του πλανήτη. Επιπλέον των χαρτών όμως, η υπηρεσία συμπεριλαμβάνει την τεχνολογία για την χαρτογραφική υποστήριξη διαφόρων εφαρμογών, μέσα από το Google Maps API.

Η υπηρεσία παρέχεται δωρεάν για μη εμπορική χρήση από κάποιο συγκεκριμένο domain, για το οποίο αφού δηλωθεί, παραχωρείται ένα αλφαριθμητικό κλειδί-ταυτότητα. Λόγω της μεγάλης πληρότητας και ακρίβειας των δεδομένων που περιέχουν, οι χάρτες της Google έχουν γίνει αυτή τη στιγμή οι ευρύτερα διαδεδομένοι διαδικτυακοί χάρτες. Σε αυτό έχουν βοηθήσει και παράγοντες βέβαιοι όπως η αξιοπιστία της υπηρεσίας, η παροχή των πληροφοριών σε πάρα πολλές γλώσσες, αλλά και η προσθήκη συνεχώς νέων θεμάτων, όπως είναι π.χ. κοντινά σημεία ενδιαφέροντος. Ιδιαίτερα έχει βοηθήσει η ύπαρξη του Google Maps API, το οποίο έχει επιτρέψει την δημιουργία πολυάριθμων εφαρμογών ανά τον κόσμο, με χρήση των Google Maps ως υπόβαθρο. Με τα εργαλεία που παρέχονται μπορεί κανείς να δημιουργήσει σχετικά εύκολα μία εφαρμογή, της οποίας ο χάρτης και οι λειτουργίες του να είναι ήδη οικεία στο μεγαλύτερο ποσοστό των χρηστών.

Το Google Maps API παρέχεται σε έκδοση για JavaScript. Το γεγονός αυτό μας εξυπηρετεί διότι η τεχνολογία Java Server Faces που έχουμε σχεδιάσει ότι θα χρησιμοποιήσουμε είναι ανοιχτή στην χρήση οποιασδήποτε τεχνολογίας και ιδιαίτερα της JavaScript

7.6.1 Χαρακτηριστικά

Το κύριο αντικείμενο του Google Maps API είναι το map, πάνω στο οποίο τοποθετούνται όλα τα υπόλοιπα αντικείμενα-παιδιά. Μια βελτιωμένη εκδοχή του map είναι το map3D, το οποίο παρέχει επιπλέον την δυνατότητα για χάρτες με τρισδιάστατη οπτική. Το αντικείμενο map έχει ιδιότητες που καθορίζουν το μέγεθος, το κεντράρισμα, την κλίμακα, την γλώσσα απεικόνισης, όπως επίσης τα είδη χάρτη και ποιά από αυτά είναι θα είναι επιλέξιμα από τον χρήστη (π.χ. χάρτης, δορυφορικός ορθοφωτοχάρτης, κ.α.). Το map δέχεται επάνω του διάφορα χειριστήρια, όπως για τον προσανατολισμό, την κλίμακα, ή την απεικόνιση ενός ευρύτερου χάρτη μικρής κλίμακας. Το αντικείμενο map μπορεί να αντιδρά σε διάφορα συμβάντα (π.χ. click του χρήστη σε κάποιο σημείο), κι ανάλογα να ενεργοποιούνται οι σχετικές ρουτίνες της εφαρμογής. Αντικείμενα-παιδιά του map μπορούν να είναι διάφορα επιθέματα, όπως Markers για σημεία, Polylines για γραμμικά στοιχεία, και Polygons για επιφανειακά στοιχεία, ή ακόμη και εικόνες (ground overlays). Κάθε ένα από αυτά έχει ένα πλήθος από ιδιότητες (σχήμα, μέγεθος, χρώμα, κτλ), και μπορεί να υπάρχει χειρισμός των συμβάντων, όπως π.χ. το click του χρήστη επάνω τους. Ένα ιδιαίτερα χρήσιμο στοιχείο, το οποίο μπορεί να ανήκει σε κάποιο από τα υπόλοιπα (Map, Markers, Polylines, Polygons) είναι το InfoWindow. Πρόκειται για ένα συννεφάκι στην πιο συνηθισμένη του μορφή, το οποίο μπορεί να περιέχει μέσα του περιεχόμενο από απλό κείμενο και HTML. [Τζιώτης 2010]

Περισσότερες και αναλυτικές πληροφορίες μπορεί κανείς να βρει στην διεύθυνση:
<http://code.google.com/intl/en/apis/maps/documentation>.

7.7 Google Geocoding API.

Το Google Geocoding είναι μία Web Υπηρεσία που παρέχεται από την Google και η οποία μπορεί να μετατρέπει διευθύνσεις σε γεωγραφικές συντεταγμένες, αλλά και το αντίστροφο. Το API δίνει την δυνατότητα πρόσβασης σε έναν geocoder μέσω ενός HTTP request. Η χρήση της Υπηρεσίας είναι δωρεάν κι ελεύθερη από διαφημίσεις μέχρι αυτή την στιγμή, αρκεί τα queries να μην ξεπεράσουν τα 2500/ημέρα για κάποιον χρήστη, ή να μην γίνονται σε πολύ γρήγορο ρυθμό. Απαραίτητη προϋπόθεση της άδειας χρήσης της υπηρεσίας είναι τα αποτελέσματα να σχετίζονται με την εμφάνισή τους σε έναν Google Map.

Εκτός από αυτήν την μορφή ερωτήματος και αποτελέσματος, το Google Geocoding παρέχει και λειτουργία μέσω SOAP (Simple Object Access Protocol)

7.8 Google Directions API.

Το Google Directions API είναι και αυτό μία υπηρεσία της Google η οποία βρίσκει διαδρομές μεταξύ τοποθεσιών, με ερωτήματα στην μορφή HTTP. Οι τοποθεσίες είναι δυνατόν να δίνονται είτε περιγραφικά, σαν κείμενο, οπότε γίνεται και geocoding, είτε σε μορφή γεωγραφικών συντεταγμένων.

Το ερώτημα προς την υπηρεσία είναι ένα HTTP URL στην ακόλουθη μορφή: <http://maps.googleapis.com/maps/api/directions/output?parameters> όπου:

Output: Είναι η αιτούμενη μορφοποίηση των αποτελεσμάτων, σε xml ή json μορφή. Και οι παράμετροι:

origin (απαιτείται): Η αφετηρία ως διεύθυνση σε αλφαριθμητική μορφή, ή ως ζεύγος γεωγραφικών συντεταγμένων.

destination (απαιτείται): Ο προορισμός ως διεύθυνση σε αλφαριθμητική μορφή, ή ως ζεύγος γεωγραφικών συντεταγμένων.

mode (προαιρετικά): Προσδιορίζει τον επιθυμητό τρόπο ταξιδιού. Μπορεί να είναι driving, walking, ή bicycling (μόνο για τις USA προς το παρόν).

waypoints (προαιρετικά): Καθορίζει μία λίστα από ενδιάμεσα σημεία διαδρομής (waypoints). Τα σημεία μπορούν να δίνονται όμοια με την αφετηρία ή τον προορισμό, ως διευθύνσεις ή ως ζεύγη γεωγραφικών συντεταγμένων. Εάν τεθεί true, τότε τα αποτελέσματα περιέχουν παραπάνω από μία προτάσεις διαδρομών, αυξάνοντας όμως τον χρόνο απόκρισης από τον server.

avoid (προαιρετικά): Καθορίζει ότι η προτεινόμενη διαδρομή θα πρέπει να αποφεύγει τα στοιχεία που δίνονται. Προς το παρόν υπολογίζονται τα στοιχεία διόδια (tolls) και αυτοκινητόδρομοι (highways).

units (προαιρετικά): Καθορίζει τις μονάδες μέτρησης στα αποτελέσματα. Δυνατές επιλογές είναι σε “metric” για το μετρικό, ή “imperial” για το αγγλοσαξονικό σύστημα (σε feet).

Η παραμετροποίηση επηρεάζει μόνο τις οδηγίες καθοδήγησης σε μορφή κειμένου που περιέχονται στην απάντηση, καθώς όλα τα υπόλοιπα στοιχεία είναι πάντα σε μέτρα. Επιπλέον, το ερώτημα δέχεται όπως και το Geocoding API τις παραμέτρους που τροποποιούν την προτεραιότητα των αποτελεσμάτων στο geocoding για ορθογώνιο πολύγωνο ή χώρα, όπως και για την γλώσσα και τον αισθητήρα θέσης. Σαν αποτελέσματα περιέχουν δύο στοιχεία: το status και το routes. Το status περιγράφεται την απόκριση του server στο ερώτημα, πρόκειται δηλαδή για το πεδίο που περιέχει την τιμή „OK” αν έχει βρεθεί απάντηση ή μία άλλη τιμή που αναφέρεται σε λάθος ή πρόβλημα. Αν το status είναι „OK”, τότε στο στοιχείο route περιέχονται τα αποτελέσματα για την διαδρομή (ή τις διαδρομές εάν έχει επιλεγεί το να παρουσιάζονται εναλλακτικές).

7.9 GMAP for JSF

Εφόσον μιλάμε για ανάπτυξη και νέες τεχνολογίες κρίνεται σκόπιμο να δούμε και την νέα τεχνολογία Gmap που ενσωματώνεται στα Java Server Faces και θα χρησιμοποιήσουμε και εμείς στην υπηρεσία μας. Αποτελεί ότι πιο σύγχρονο για την υλοποίηση διαδικτυακών GIS εφαρμογών και πρέπει να μάθουμε τη λειτουργία της.

Το GMaps4JSF αποσκοπεί στη χρήση του Google Maps API σε JSF εφαρμογές. Οι JSF χρήστες θα έχουν επίσης και τη δυνατότητα να κατασκευάσουν και περίπλοκα Streetview Panoramas και Χάρτες με λίγες μόνο γραμμές κώδικα(JSF tags).

GMaps4JSF αποτελεί μια JSF βιβλιοθήκη φτιαγμένη για το web 2.0 .

Η βιβλιοθήκη αυτή δίνει τη δυνατότητα να:

- Δημιουργούνται χάρτες με τη χρήση των συντεταγμένων ή των οδών.
- Προστίθενται marker(s) στο χάρτη.
- Προστίθενται πληροφορίες σαν κείμενο στο χάρτη.
- Προστίθενται χειριστήρια στο χάρτη.
- Δημιουργούνται event listener(s) στα αντικείμενα του χάρτη.
- Σχεδιάζονται πολύγωνα
- Σχεδιάζονται γραμμές.
- Σχεδιάζονται κύκλοι.
- Προστίθενται groundOverlay(s) στο χάρτη.
- Πραγματοποιούνται διάφορες διαδικασίες όπως zoom in και out, εναλλαγή χαρτών κ.α
- Δημιουργούνται Streetview Panoramas.

Χρήση της βιβλιοθήκης

- Απλά προσθέτουμε το παρακάτω στο pom.xml:

```
<repositories>
  ...
  <repository>
    <id>googlecode.com</id>

    <url>http://gmaps4jsf.googlecode.com/svn/trunk/gmaps4jsf-repo</url>
  </repository>
</repositories>

<dependencies>
  ...
  <dependency>
    <groupId>com.googlecode.gmaps4jsf</groupId>
    <artifactId>gmaps4jsf-core</artifactId>
    <version>1.1.1</version>
  </dependency>
</dependencies>
```

Για περισσότερες πληροφορίες και υλοποίηση παραδείγματος θα βρείτε στον παρακάτω σύνδεσμο.

Επιπλέον

- Το GMaps4JSF επιτρέπει και τη χρήση της Javascript.
- Το GMaps4JSF υποστηρίζει τα Facelets.
- Το GMaps4JSF υποστηρίζει τα Portlets.
- Όλα τα GMaps4JSF στοιχεία υπάρχουν στον παρακάτω σύνδεσμο σαν demo <http://www.mashups4jsf.com/gmaps4jsf-examples/home.jsf>.

7.10 Ajax

Ο όρος Ajax σημαίνει Asynchronous JavaScript and Xml και χρησιμοποιείται ευρέως για την δημιουργία δυναμικών εφαρμογών. Το Google Map που θα ενσωματώσουμε στην υπηρεσία μας αποτελεί εφαρμογή αυτού του είδους. Το Ajax διαχωρίζει τη χρήση που γίνεται από το χρήστη από τη χρήση που γίνεται στον server και τους δίνει την δυνατότητα να λειτουργούν παράλληλα. Έτσι, με αυτό τον τρόπο οι διαδικτυακές εφαρμογές είναι πιο γρήγορες.

7.11 Google Places

Μια νέα επαναστατική θα χαρακτηρίζαμε υπηρεσία της Google είναι το Google Places. Το Google places αποτελεί μια βιβλιοθήκη Javascript που επιτρέπει στην εφαρμογή να ψάξει για τοποθεσίες, όπως αυτές έχουν δημιουργηθεί στο Interface και μπορεί να είναι γεωγραφικές τοποθεσίες ή κοντινά σημεία ενδιαφέροντος, που υπάρχουν σε μια περιοχή στο χάρτη, γύρω από ένα σημείο.

Το places είναι μια αυτόνομη βιβλιοθήκη ξεχωριστή από το κυρίως MAPS API. Προκειμένου να πραγματοποιηθεί η χρήση του πρέπει πρώτα να φορτωθεί το script στο URL. Παρακάτω βλέπουμε τον κώδικα που αποτελεί το script που τρέχει την συγκεκριμένη υπηρεσία.

```
<script type="text/javascript"  
src="http://maps.googleapis.com/maps/api/js?libraries=places&sensor=true_or_false"></script>
```

Με την υπηρεσία δίνεται η δυνατότητα στον χρήστη

- Να ψάξει σημεία που βρίσκονται γύρω από τη δικιά σου τοποθεσία.
- Να πραγματοποιήσει αναζητήσεις με χρήση μιας λέξης κλειδί.
- Να εντοπίσει τοποθεσίες που περιλαμβάνουν αναλυτικές πληροφορίες για ένα συγκεκριμένο μέρος, συμπεριλαμβανομένου και τις γνώμες χρηστών

Οι πληροφορίες αυτές μπορούν να περιλαμβάνουν διάφορα όπως εστιατόρια, καταστήματα, γραφεία, υπηρεσίες, καθώς επίσης και αποτελέσματα geocoding όπου προσδιορίζεται η διεύθυνση, η περιοχή ή η πόλη και άλλα σημεία ενδιαφέροντος

Προκειμένου να πραγματοποιηθεί η αναζήτηση των στοιχείων που επιθυμούμε το Places παρέχει την μέθοδο search(). Με τη χρήση αυτής της μεθόδου η εφαρμογή επιστρέφει ένα πίνακα από τις τοποθεσίες που τηρούν τα κριτήρια αναζήτησης μας. Η αναζήτηση πραγματοποιείται παίρνοντας τις συντεταγμένες σαν αντικείμενα που αναπαριστούν ένα κεντρικό σημείο και ορίζοντας μια ακτίνα αναζήτησης.

.

7.11.1 Λεπτομέρειες τοποθεσιών

Πέρα από τις τοποθεσίες η υπηρεσία δίνει τη δυνατότητα και επιστροφής λεπτομερών πληροφοριών για ένα μέρος. Αυτές οι επιπλέον λεπτομέρειες μπορεί να είναι τηλεφωνικοί αριθμοί, ταχυδρομικοί κώδικες, κριτικές χρηστών κ.α

Η υπηρεσία πραγματοποιείται με τη κλήση της μεθόδου getDetails() method. Η μέθοδος αυτή επιστρέφει ένα αντικείμενο με τις παρακάτω ιδιότητες:

- Στοιχεία διευθύνσεων
- Τροποποιημένες διευθύνσεις με επιστροφή της πλήρης διεύθυνσης αναλόγως και της περιοχής που βρίσκεται.
- Τροποποιημένο αριθμού τηλεφώνου με την κατάλληλη εισαγωγή και του κωδικού περιοχής
- Γεωμετρικές-γεωχωρικές πληροφορίες όπως τις συντεταγμένες της περιοχής
- html_attributions: κείμενο με συσχετίσεις για το μέρος
- Icon: URL σε μια εικόνα που αναπαριστά το μέρος
- Id: Μοναδικό id, αναγνωριστικό του κάθε τόπου
- Name: Το όνομα του μέρους
- Opening_hours: με τις παρακάτω πληροφορίες.
 - Open_now: μια μεταβλητή Boolean δηλαδή ναι/όχι που υποδεικνύει αν το μέρος είναι ανοικτό την συγκεκριμένη ώρα αναζήτησης.
 - Periods []: είναι ένας πίνακας με τις περιόδους λειτουργίας του μέρους που θα καλύπτει και τις 7 ημέρες.
 - Κάθε περίοδος περιλαμβάνει

- Τη μεταβλητή open με αντικείμενα τις ώρες που το μέρος είναι ανοιχτό.
 - Day: έναν αριθμό από το 0-6 που αντιστοιχεί σε μία μέρα για παράδειγμα το 0 είναι η Κυριακή.
 - Time: περιέχει την ώρα
 - Close: περιλαμβάνει την ώρα που παραμένει κλειστή η τοποθεσία.
- Rating: Αποτελεί μια αξιολόγηση βάσει των βαθμολογιών που προκύπτουν από τις κριτικές χρηστών
 - Reviews: είναι ένας πίνακας που περιλαμβάνει μέχρι 5 απόψεις σχετικά με το μέρος

7.11.2 Places autocomplete

Το Places auto complete προσθέτει αυτόματα χαρακτηριστικά σε ένα text field καθώς ο χρήστης πληκτρολογεί την τοποθεσία. Η υπηρεσία στην συνέχεια επιστρέφει προτάσεις με προβλέψεις για τη λέξη και τις παρουσιάζει σε ένα drop down list. Όταν ο χρήστης επιλέγει το μέρος από τη λίστα όλες οι πληροφορίες που αφορούν το μέρος επιστρέφουν σαν ένα αντικείμενο και μπορεί να χρησιμοποιηθεί στην ολοκλήρωση της υπηρεσίας για την αναζήτηση.

7.11.3 Distance Matrix Service

Google's Distance Matrix είναι μια υπηρεσία που υπολογίζει την απόσταση που χρειάζεται κάποιος για να ταξιδέψει συμπεριλαμβανομένου και τη χρονική διάρκεια του ταξιδιού από το ένα μέρος στο άλλο και δύναται να περιλαμβάνει πολλαπλά

Η υπηρεσία αυτή δεν επιστρέφει λεπτομερείς πληροφορίες απλά περιλαμβάνει τα πολύγωνα με τα σημεία των τοποθεσιών και τις κατευθυντήριες οδηγίες.

Η χρήση της υπηρεσίας έχει κάποια όρια όπως φαίνονται και παρακάτω.

- 25 αρχικές τοποθεσίες
- 25 προορισμοί ανά αίτημα και
- μέχρι 100 στοιχεία ανά αίτημα

Αυτή η υπηρεσία μας γλιτώνει από την επίπονη διαδικασία της ενσωμάτωσης των αλγορίθμων των γράφων στην εφαρμογή μας. Είναι απλή στη χρήση και τα όρια που διαθέτει είναι ικανοποιητικά για την εφαρμογή μας.

Τα υποστηριζόμενα πεδία είναι τα ακόλουθα.

- Origin Addresses: αποτελεί ένα πίνακα που περιλαμβάνει όλες τις τοποθεσίες που θα περάσουν στη μήτρα υπολογισμού απόστασης και είναι οι αφετηρίες. Οι διευθύνσεις επιστρέφονται όπως αυτές ορίζονται από το geocoding
- Destination Addresses: είναι ένας πίνακας που περιέχει τις τοποθεσίες που αποτελούν τους προορισμούς μια διαδρομής.
- Rows: είναι ένας πίνακας που έχει τις σειρές της μήτρας που ανταποκρίνεται σε μια αφετηρία.
- Elements: είναι τα παιδιά των σειρών αυτών που ταιριάζουν τις αφετηρίες με τους προορισμούς και περιλαμβάνουν στοιχεία όπως κατάσταση, απόσταση, διάρκεια και πληροφορίες των προορισμών και των αφετηριών.

7.12 Java persistence

Το Java Persistence API δίνει την δυνατότητα στους προγραμματιστές να πραγματοποιήσουν την χαρτογράφηση της βάσης δημιουργώντας αντικείμενα που θα χρησιμοποιηθούν για την διαχείριση σχεσιακών βάσεων δεδομένων. Το interface αυτό αποτελείται από τις τέσσερις παρακάτω περιοχές

- Το Java Persistence API
- Την Γλώσσα εκτέλεσης των Queries
- Το Java Criteria API
- Μεταδεδομένα χαρτογράφησης αντικειμένων/σχέσεων

7.12.1 Entities

Τα entities αποτελούν στην ουσία τα αντικείμενα μιας βάσης. Με πιο απλά λόγια είναι ένας πίνακας της σχεσιακής βάσης δεδομένων και κάθε στιγμιότυπο αποτελεί μια σειρά στον πίνακα. Το persistence state αποτελεί τα πεδία της βάσης μας και τις ιδιότητες. Αυτά τα πεδία και οι ιδιότητες χρησιμοποιούν τα mapping annotations όπως λέγονται του χάρτη των entities και των σχέσεων της βάσης και το συσχετίζουν με τα αποθηκευμένα δεδομένα.

Τα πρωτεύοντα κλειδιά αποτελούν και τα μοναδικά αναγνωριστικά των κάθε entities. Το κάθε πρωτεύον κλειδί συσχετίζεται με ένα στιγμιότυπο της βάσης. Κάθε entity πρέπει να έχει κλειδί για να πραγματοποιηθεί η χαρτογράφηση και είναι μοναδικό.

Οι σχέσεις σε μια αντικειμενοστραφή βάση με τη χρήση του java persistence μπορεί να είναι οι παρακάτω:

- Ένα προς ένα: Κάθε στιγμιότυπο σχετίζεται με ένα μόνο άλλο στιγμιότυπο μια άλλης οντότητας. Για την υλοποίηση αυτής της σχέσης θα χρησιμοποιηθεί το `javax.persistence.OneToOne` annotation.
- Ένα προς πολλά: Ένα στιγμιότυπο μπορεί να σχετίζεται με πολλά άλλων οντοτήτων. Για την υλοποίηση αυτής της σχέσης θα χρησιμοποιηθεί το `javax.persistence.OneToMany` annotation.
- Πολλά προς ένα: Στην ουσία αποτελεί το αντίστροφο του ένα προς πολλά και δεν υπάρχουν διαφορές στην χρήση του καθώς όπως βλέπουμε χρησιμοποιούν και το ίδιο annotation, το `javax.persistence.OneToMany` annotation.
- Πολλά προς πολλά: Πολλά στιγμιότυπα μια οντότητας μπορούν να σχετίζονται με πολλά στιγμιότυπα μιας άλλης οντότητας. Στην περίπτωση αυτή χρησιμοποιείται το `javax.persistence.ManyToMany` annotation.

Τέλος, πρέπει να δούμε και πώς χειριζόμαστε τις οντότητες αυτές μέσα στο πρόγραμμα. Η διαχείριση είναι εύκολη και απλή, την όλη διαδικασία την χειρίζεται ο entity manager, ο οποίος χωρίζεται στους αντίστοιχους entity managers των στιγμιότυπων. Κάθε instance entity manager σχετίζεται με ένα περιεχόμενο που περιέχει συγκεκριμένα αποθηκευμένα δεδομένα της βάσης μας. Με τον entity manager πραγματοποιούνται όλες οι διαδικασίες χειρισμού των δεδομένων όπως διαγραφή, εισαγωγή κλπ.

7.13 Βάση Δεδομένων

Για την υλοποίηση της εφαρμογής θα χρησιμοποιηθεί βάση ανοιχτού κώδικα. Δεν χρειάζεται να καθορίσουμε εξ αρχής το εργαλείο ανάπτυξης της βάσης γιατί μπορεί να διαφέρει αναλόγως τις προτιμήσεις του χρήστη. Δύο πολύ γνωστά περιβάλλοντα ανάπτυξης είναι αυτά της mysql και η postgres. Παρακάτω θα περιγραφούν συνοπτικά τα χαρακτηριστικά και οι δυνατότητες των δυο αυτών σχεδιαστικών εργαλείων προκειμένου ο αναγνώστης να έχει μια ιδέα το τι προσφέρει κάθε εργαλείο.

7.13.1 MySQL Workbench

MySQL Workbench επιτρέπει σε ένα προγραμματιστή, να σχεδιάσει οπτικά και να δημιουργήσει βάσεις δεδομένων όλων των τύπων όπως Web, OLTP, και στοιχείων αποθήκης. Περιλαμβάνει μια μοντελοποίηση των δεδομένων και χρειάζεται για τη δημιουργία σύνθετων μοντέλων ER. Παρέχει επίσης βασικά χαρακτηριστικά για την εκτέλεση δύσκολων διαχειριστικών αλλαγών και τεκμηρίωσης που απαιτούν συνήθως πολύ χρόνο και προσπάθεια. Το MySQL Workbench είναι διαθέσιμο για Windows, Linux και Mac OS.

ΟΦΕΛΗ

- Απλοποιεί το σχεδιασμό και τη συντήρηση της βάσης δεδομένων.
- Αυτοματοποιεί χρονοβόρα και επιρρεπής σε λάθη εργασίες.
- Επιτρέπει μοντέλο με γνώμονα το σχεδιασμό βάσης δεδομένων με πιο αποτελεσματική μεθοδολογία για την δημιουργία έγκυρων και με καλές επιδόσεις βάσεις δεδομένων, παρέχοντας ταυτόχρονα την ευελιξία να ανταποκριθεί στις εξελισσόμενες απαιτήσεις των επιχειρήσεων
- Παρέχει δυνατότητες να διαβιβάσει φυσικά σχέδια της βάσης δεδομένων και αντίστροφα
- Επιτρέπει την εισαγωγή SQL scripts για τη δημιουργία μοντέλων και την εξαγωγή μοντέλων για DDL σενάρια που μπορούν να εκτελεστούν σε μεταγενέστερο χρόνο
- Δίνει τη δυνατότητα σύγκρισης δύο ζωντανών βάσεις δεδομένων ή ενός μοντέλου και μιας ζωντανής βάσης δεδομένων,
- Απλοποιεί την τεκμηρίωση των σχεδίων της βάσης δεδομένων, παρέχοντας ένα σημείο που παρέχει τεκμηρίωση σε μορφή HTML ή μορφή απλού κειμένου

7.13.2 PostgreSQL

Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων ανοικτού κώδικα με πολλές δυνατότητες. Η ανάπτυξη της διαρκεί ήδη πάνω από δύο δεκαετίες και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών της γύρω από την αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία.

Η PostgreSQL τρέχει σε όλα τα βασικά λειτουργικά συστήματα, στα οποία περιλαμβάνονται το Linux, το UNIX (AIX, BSD, HP-UX, SGI, IRIX, MAC OS X, Solaris, Tru64) και τα Windows. Είναι συμβατή με ACID, και συμπεριλαμβάνει τους περισσότερους SQL92 και SQL99 τύπους δεδομένων συμπεριλαμβανομένων INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL και TIMESTAMP. Επίσης υποστηρίζει αποθήκευση μεγάλων δυαδικών αντικειμένων (binary), όπως εικόνες, ήχοι ή βίντεο. Διαθέτει επίσης περιβάλλοντα προγραμματισμού για τις γλώσσες προγραμματισμού C, C++, Java, Perl, Python, Ruby, Tcl, και υποστήριξη για την πλατφόρμα .NET και το πρότυπο ODBC, ενώ περιλαμβάνει και εξαιρετικό εγχειρίδιο χρήσης.

Η PostgreSQL προέρχεται από το πακέτο POSTGRES, το οποίο γράφτηκε στο Πανεπιστήμιο του Μπέρκλεϋ στην Καλιφόρνια των Η.Π.Α.. Αναπτύσσεται πάνω από δύο δεκαετίες και αποτελεί πλέον την πιο προχωρημένη βάση δεδομένων ανοικτού κώδικα.

Σχέδιο POSTGRES

Το σχέδιο POSTGRES υπό την ηγεσία του καθηγητή Michael Stonebreaker ξεκίνησε να εφαρμόζεται το 1986. Η POSTGRES έχει αρκετά σημαντικές διανομές από τότε. Η πρώτη δοκιμαστική έκδοση του συστήματος παρουσιάστηκε το 1988 στο συνέδριο ACM-SIGMOD και κυκλοφόρησε σε μικρή ομάδα ατόμων τον Ιούνιο του 1989. Η POSTGRES χρησιμοποιήθηκε για την εφαρμογή ποικίλων εφαρμογών παραγωγής και έρευνας, όπως για παράδειγμα ένα οικονομικό σύστημα ανάλυσης δεδομένων, ένα πακέτο παρακολούθησης της απόδοσης ενός κινητήρα τζετ, βάση δεδομένων εντοπισμού αστεροειδή, μιας ιατρικής βάσης δεδομένων και μερικών συστημάτων γεωγραφικής πληροφορίας. Η POSTGRES χρησιμοποιήθηκε επίσης ως εργαλείο εκπαίδευσης σε ορισμένα πανεπιστήμια.

Χαρακτηριστικά

- ANSI SQL 89, 92, 93
- 100% συμβατή με ACID και πλήρη υποστήριξη commit και rollback.
- Online αντίγραφα ασφαλείας: υψηλότερη ασφάλεια και διαθεσιμότητα των δεδομένων.
- Τύποι δεδομένων: numeric, decimal, smallint, integer, bigint, real, double, serial, char, varchar, bit, text, date, time, timestamp, interval, boolean, network address, geometric types και πολλά άλλα.
- Δυνατότητα δημιουργίας νέων τύπων δεδομένων από τους χρήστες.
- Αποθήκευση BLOBS (binary large objects), συμπεριλαμβανομένων αρχείων κειμένου, ήχου, εικόνων ή βίντεο.
- Πλήρης υποστήριξη συναρτήσεων συγκεντρωτικών αποτελεσμάτων (GROUP_BY) όπως COUNT, SUM, AVG, MIN, MAX, STDDEV and VARIANCE. Δυνατότητα δημιουργίας νέων συγκεντρωτικών συναρτήσεων εφόσον χρειαστεί.
- Υποστήριξη όλων των τύπων ενώσεων (cross, inner, outer, left, right, full, natural).
- Συναρτήσεις ορισμένες από τον χρήστη, οι οποίες μπορούν να γραφούν σε πολλές γλώσσες προγραμματισμού όπως C, SQL, PL/SQL, TCL, Perl, Python and Ruby.
- Περιβάλλον ανάπτυξης γλωσσών προγραμματισμού όπως Perl, Python, Zope, PHP, TCL/TK, ODBC, JDBC, C/C++, Embedded SQL, Delphi/Kylix/Pascal, VB, ASP, Java.
- Βιβλιοθήκη συναρτήσεων και τελεστών με ορισμένες προεγκατεστημένες συναρτήσεις όπως math, date/time, string, geometric, formatting κ.α.
- Συναρτήσεις trigger μπορούν να οριστούν από οποιαδήποτε γλώσσα προγραμματισμού που υποστηρίζει server όπως C ή PL/pgQL.
- Προσωρινοί πίνακες οι οποίοι σβήνονται αυτόματα μετά το τέλος της συνόδου.
- Μοντέλο ασφαλείας ομάδας/χρήστη. Η πρόσβαση στο διακομιστή της βάσης δεδομένων μπορεί να περιορίζεται από το χρήστη, κεντρικό υπολογιστή ή μια βάση δεδομένων.
- Το μέγεθος του πίνακα και της βάσης δεδομένων είναι σχεδόν απεριόριστο. Απεριόριστες καταχωρήσεις και ευρετήρια ανά πίνακα.

7.14 Unified Modeling Language (UML)

Η Unified Modeling Language (UML, μη δόκιμη απόδοση στην Ελληνική γλώσσα: Ενοποιημένη Γλώσσα Μοντελοποίησης) πλέον είναι η πρότυπη γλώσσα μοντελοποίησης στη μηχανική λογισμικού. Χρησιμοποιείται για τη γραφική απεικόνιση, προσδιορισμό, κατασκευή και τεκμηρίωση των στοιχείων ενός συστήματος λογισμικού. Μπορεί να χρησιμοποιηθεί σε διάφορες φάσεις ανάπτυξης, από την ανάλυση απαιτήσεων ως τον έλεγχο ενός ολοκληρωμένου συστήματος. Αποτελείται από ένα σύνολο προσυμφωνημένων όρων, συμβόλων και διαγραμμάτων που επιτρέπουν:

- την εμφάνιση των ορίων ενός συστήματος και των βασικών λειτουργιών του, χρησιμοποιώντας «περιπτώσεις χρήσης» (use-cases) και «actors».
- την επεξήγηση της πραγματοποίησης των περιπτώσεων χρήσης με «διαγράμματα αλληλεπίδρασης».
- την αναπαράσταση μιας στατικής δομής ενός συστήματος χρησιμοποιώντας «διαγράμματα κλάσεων».
- τη μοντελοποίηση της συμπεριφοράς των αντικειμένων με «διαγράμματα καταστάσεων».
- την αποκάλυψη της υλοποίησης της αρχιτεκτονικής με «διαγράμματα συστατικών» και «ανάπτυξης».
- την επέκταση της λειτουργικότητας με «στερεότυπα».

7.14.1 Διαγράμματα Κλάσεων

Τα διαγράμματα κλάσεων της UML ορίζουν γεωμετρικά σχήματα ως συμβολισμούς για τα αντικείμενα, τις κλάσεις και τις διασυνδέσεις, ενώ διαφόρων τύπων γραμμές χρησιμοποιούνται για να συνδέουν αυτά τα σχήματα και να υποδηλώνουν έτσι τον τρόπο που κληρονομούν, συνεργάζονται ή εξαρτώνται μεταξύ τους. Τα αντικείμενα της ίδιας κλάσης αναπαριστώνται με ένα μόνο γεωμετρικό σχήμα. Όταν ένα αντικείμενο χρησιμοποιεί κώδικα κάποιας άλλης κλάσης (π.χ. καλώντας μία μέθοδό της), σύμφωνα με το πρότυπο της UML υπάρχει μία «εξάρτηση» (dependency) μεταξύ τους η οποία αναπαρίσταται με μία διακεκομμένη γραμμή. Αυτή η εξάρτηση μπορεί να είναι «συσχέτιση» (association), ένας τύπος εξάρτησης που υπονοεί πραγματική συνύπαρξη στη μνήμη στιγμιότυπων των συμμετεχόντων κλάσεων κατά τον χρόνο εκτέλεσης, «συνάθροιση» (aggregation), ένας τύπος συσχέτισης ο οποίος σημαίνει ότι το ένα αντικείμενο μπορεί να περιέχει στιγμιότυπα της άλλης κλάσης ως γνωρίσματα του, ή «σύνθεση» (composition), ένας πιο ισχυρός τύπος συνάθροισης που υπονοεί πως ο χρόνος ζωής των αντικειμένων είναι κοινός (δημιουργούνται και καταστρέφονται στη μνήμη ταυτόχρονα). Καθεμία από αυτές τις σχέσεις συμβολίζεται οπτικά με έναν διαφορετικό τύπο γραμμής μεταξύ των συμμετεχόντων κλάσεων, ενώ μπορεί να υπάρχουν και εξαρτήσεις οι οποίες δεν είναι καν συσχετίσεις (π.χ. όταν ένα αντικείμενο καλεί μία στατική μέθοδο κάποιας κλάσης).

8

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΥΠΗΡΕΣΙΑΣ

8.1 Αρχιτεκτονική και Σχεδίαση Tourism Planner

Αυτή η υπηρεσία τουρισμού που θα δημιουργήσουμε αποτελεί μια δικτυακή εφαρμογή η οποία σύμφωνα με το σχεδιασμό μας θα περιέχει αρκετά Java EE6 API και θα κάνει χρήση άλλων υπηρεσιών όπως θα δούμε παρακάτω

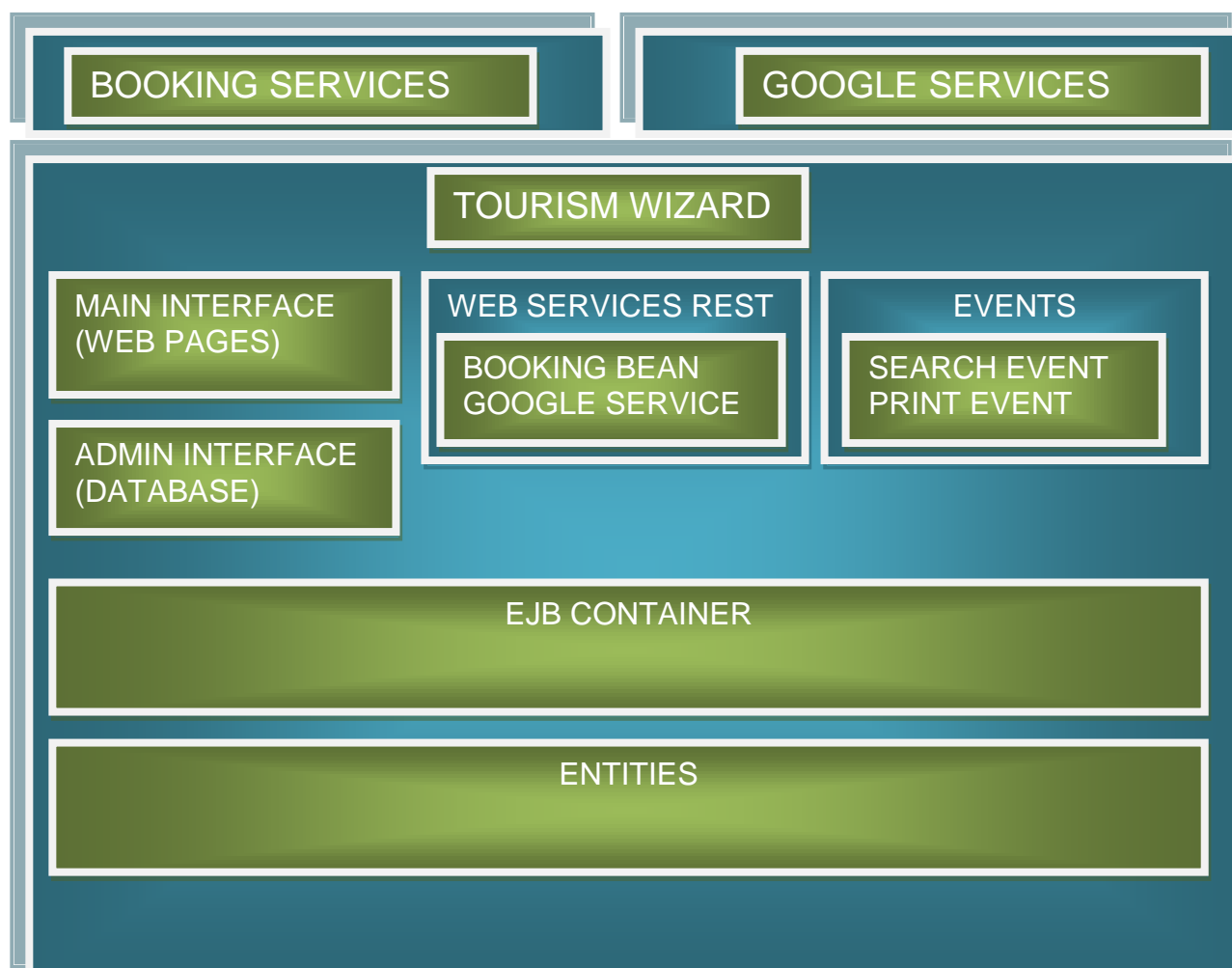
- Java Server Faces με Ajax
- Contexts and Dependency Injection για την πλατφόρμα Java
- Java API for XML Web Services
- Java API για Restful υπηρεσίες
- Java Persistence
- Java API για JavaBeans Validation
- Enterprise Beans
- PrimeFaces
- Google maps API
- Google directions API
- Google geocoding API
- Google places API
- GMAP

Η υπηρεσία θα περιέχει τις παρακάτω υλοποιήσεις:

- Tourism Wizard, αποτελεί την εφαρμογή του οδηγού που περιέχει τις επιλογές για τις πόλεις, τις ημερομηνίες επίσκεψης και τις δραστηριότητες προκειμένου να εξαχθούν τα αποτελέσματα των προτεινόμενων επισκέψεων.
- Booking services αποτελεί το σύνολο των διαθέσιμων υπηρεσιών για δυνατότητες κρατήσεων, εισιτηρίων, ξενοδοχείων κλπ.

- Google services είναι οι υπηρεσίες που παρέχονται από την Google και περιλαμβάνουν το Google maps API, το Google Places API, το geocoding API, το Google Directions API και το Gmap
- Tourism resources: Είναι ένα απλό Java project που περιέχει όλες τις πηγές που χρησιμοποιούνται για το πρόγραμμα όπως css styles, images, massages, locales για τις διάφορες γλώσσες, java server faces components, javascript components και Ajax components.
- Entities: Είναι το project που περιλαμβάνει όλες τις οντότητες που έχουν δημιουργηθεί από το persistence.
- Events: Αποτελεί ένα java archive project που περιλαμβάνει μια POJO κλάση που χρησιμοποιείται σαν CDI event.

Η υπηρεσία γενικά θα αποτελείται από τα τρία projects που αναφέραμε δηλαδή το tourism wizard, booking services και Google services. Στο παρακάτω διάγραμμα βλέπουμε πώς υλοποιείται η υπηρεσία μαζί με τα entities και τα events.



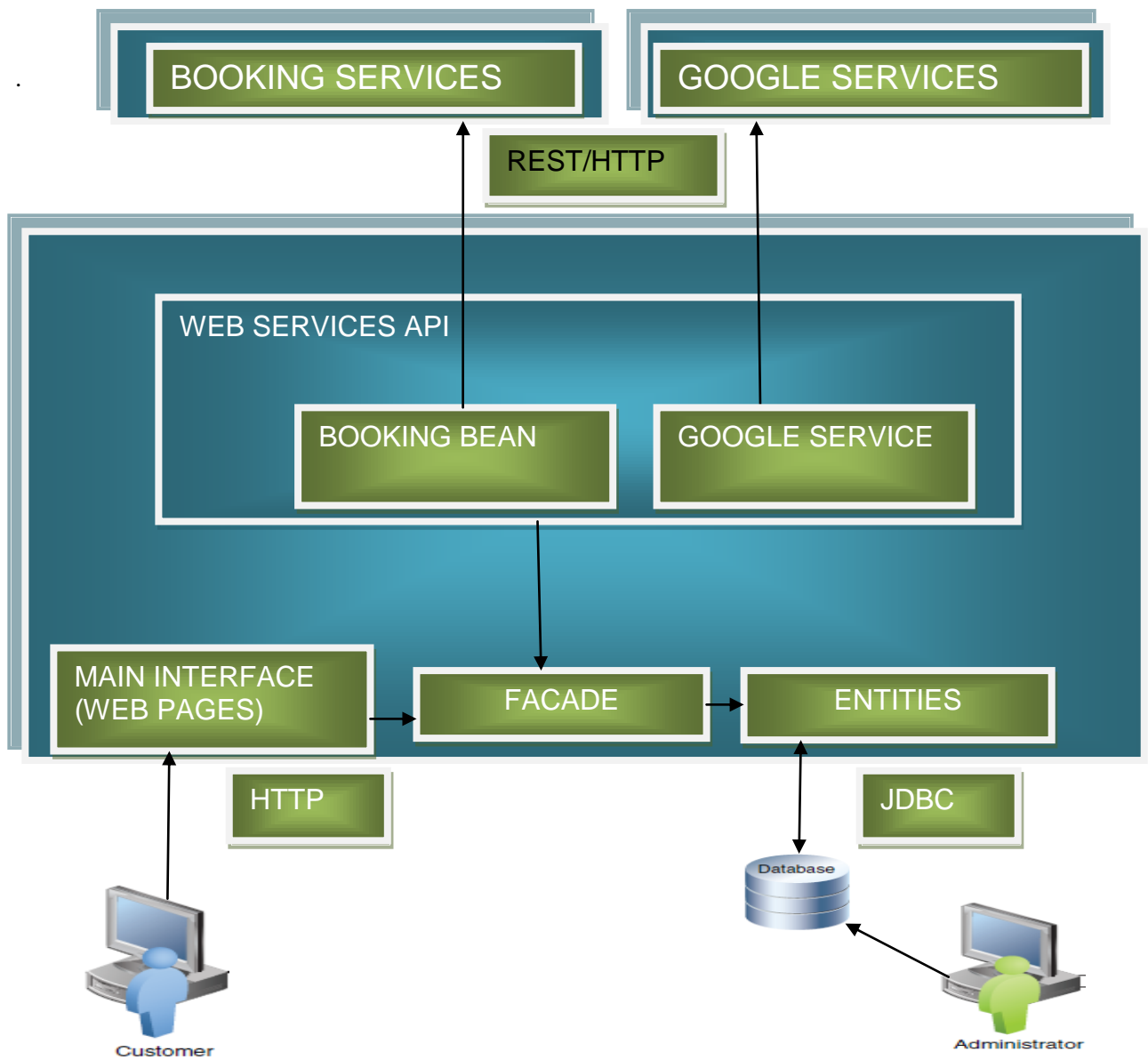
Σχήμα 4: Αναπαράσταση Υπηρεσίας

Παρακάτω θα δούμε τα χαρακτηριστικά της JavaEE6 που θα περιλαμβάνει

- Java Persistence API entities:
 - Bean Validation annotations on the entities for verifying data
 - XML annotations for Java API for XML Binding (JAXB) serialization
- Web services:
 - Google maps API
 - Google Geocoding API
 - Google directionAPI
 - Google Places API
 - GMAP
 - A JAX-RS η οποία είναι βαισμένη σε EJB
- Enterprise beans:
 - Local session beans
 - All enterprise beans packaged within theWAR
- Contexts and Dependency Injection (CDI):
 - CDI annotations for JavaServer Faces components
 - A CDI managed bean used as a wizard
 - Qualifiers
 - Events and event handlers
- Servlets:
 - A Servlet 3.0 file upload example
 - A servlet for dynamic image presentation
- JavaServer Faces technology, using Facelets for the web front end
 - Templating
 - Composite components
 - Resources packaged in a JAR file so they can be found in the classpath
- Security:
 - Java EE security constraints on the administrative interface business methods(enterprise beans)
 - Security constraints for customers and administrators (web components)

Το πρόγραμμα αυτό θα έχει ένα user interface για το χρήστη και ένα εικονικό interface το οποίο θα είναι διαθέσιμο στο διαχειριστή προκειμένου να εισάγει νέα δεδομένα στη βάση αλλά και να τροποποιεί την υπηρεσία σε περίπτωση εισχώρησης μια νέας υπηρεσίας άλλου φορέα.

Η εφαρμογή έχει ως εξής: Ο χρήστης μπαίνει στην ιστοσελίδα στο παρεχόμενο σε αυτόν user interface ενώ ο διαχειριστής κάνει χρήση το δικό του interface το εικονικό όπως αναφέραμε. Και τα δύο Interfaces έχουν πρόσβαση σε ένα facade το οποίο περιλαμβάνει όλα τα managed beans και τα stateless sessions τα οποία με τη σειρά τους συνεργάζονται με τα entities δηλαδή τη βάση δεδομένων μας. Το facade επίσης συνεργάζεται και με τα APIs της Google που αναφέραμε καθώς και με τυχόν άλλες υπηρεσίες κρατήσεων κλπ. Τέλος, είναι τα events και τα events handlers που αποτελούν τα πιο σημαντικά κομμάτια του προγράμματος τα οποία συνεργάζονται με τις πηγές του προγράμματος και τις διαδικασίες όπως το search event και το print event το οποίο θα περιλαμβάνει και τον τρόπο εύρεσης των επιθυμιών του χρήστη και θα ψάχνει αποτελέσματα με στοιχεία από τη βάση καθώς και την εκτύπωση αυτών των αποτελεσμάτων. Στο σχήμα βλέπουμε πως επικοινωνούν τα διάφορα μέρη του συστήματος μας



Σχήμα 5: Επικοινωνιακό σχήμα

8.2 Events

Τα events αποτελούν από τα πιο σημαντικά κομμάτια της εφαρμογής μας. Στην δική μας εφαρμογή το event που ξεχωρίζει είναι το search event που αποτελείται από τη κλάση με τον αλγόριθμο αναζήτησης των αποτελεσμάτων και επιπλέον είναι η υπεύθυνη κλάση για τα μηνύματα που αφορούν τα διάφορα αντικείμενα μας.

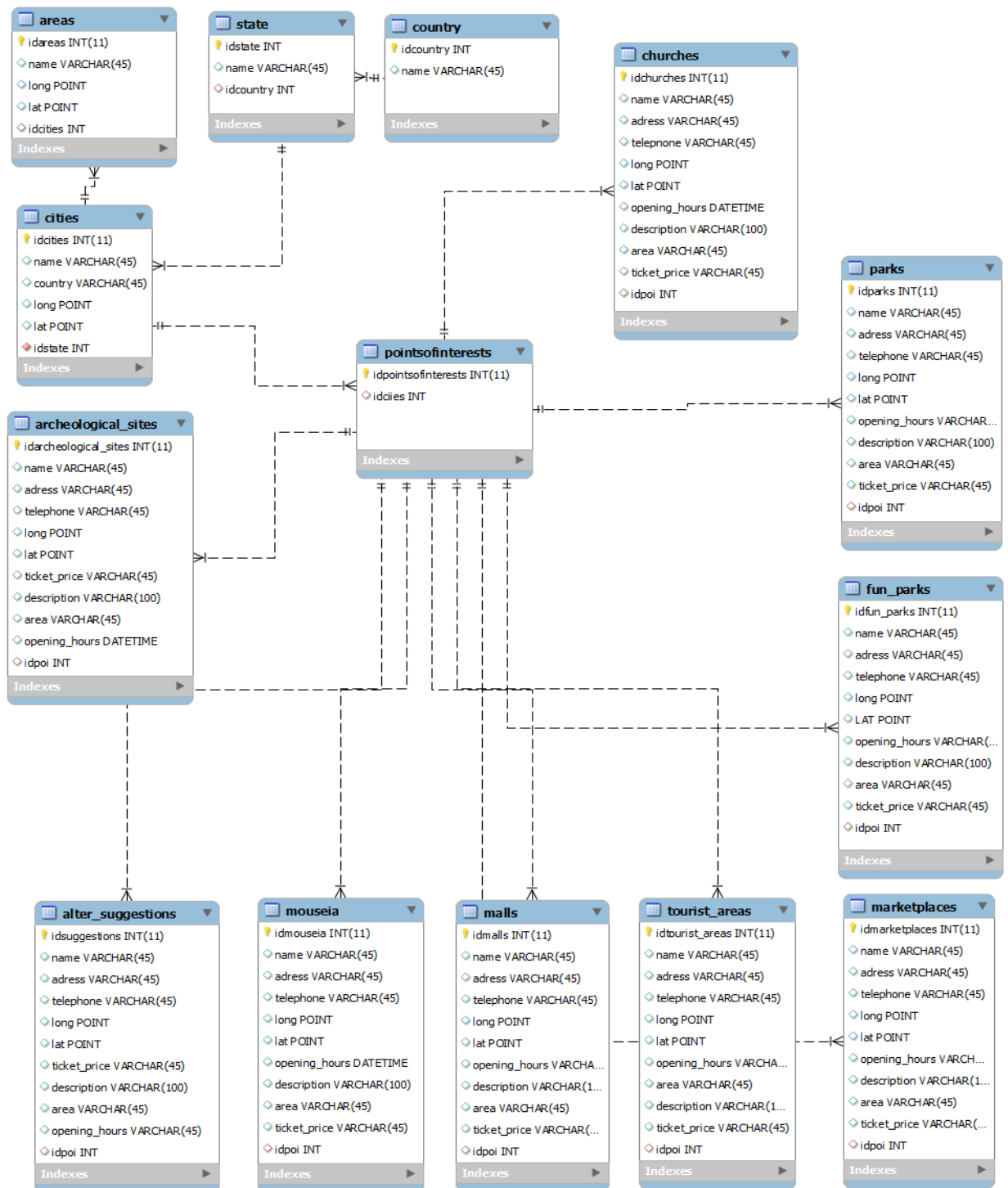
Η εφαρμογή μας θα στέλνει κάποια μηνύματα ανάλογα με τα διάφορα γεγονότα τα οποία συμβαίνουν στην εφαρμογή. Αυτά είναι όπως παρακάτω:

- @Trip planning start: Που σηματοδοτεί την έναρξη του σχεδιαστή.
- @Search: Όταν ολοκληρωθεί η συμπλήρωση του οδηγού για να αρχίσει η αναζήτηση
- @Completed: Όταν ολοκληρωθεί η διαδικασία
- @Printed: Όταν μια πρόταση ή διαδρομή επιλέχτηκε για εκτύπωση

Αντίστοιχα με τα events που αναφερθήκαν παραπάνω δηλαδή το search και το print δημιουργούνται και δύο event handlers που ελέγχουν και επεξεργάζονται όλες τις λειτουργίες που θα εκτελεστούν.

8.3 The entities project

Το project αυτό, στην ουσία, αποτελεί τη βάση δεδομένων που θα χρησιμοποιηθεί στο πρόγραμμα. Το σχεδιάγραμμα που παρουσιάζεται ακολούθως, αποτελεί το εκτεταμένο μοντέλο σε αντικειμενοστραφή παρουσίαση με χρήση της γλώσσας uml.



Σχήμα 6: Βάση δεδομένων με UML αναπαράσταση

Η Βάση περιλαμβάνει συνολικά 14 πίνακες που τηρούν μεταξύ τους τις αρχές κληρονομικότητας και βρίσκονται σε κανονική μορφή. Αναλυτικότερα οι πίνακες μας είναι εξής:

- Alter suggestion
- Museums
- Malls
- Tourist areas
- Market places
- Fun parks
- Archeological sites
- Parks
- Churches
- Points of interest
- Country
- State
- Cities
- Areas

Πιο αναλυτικά η βάση μας έχει ως εξής: Όλα τα μέρη μας ανήκουν σε κάποια χώρα. Κάθε χώρα έχει περιφέρειες που για την Ελλάδα υπάρχει η δυνατότητα να χωριστεί σε περιφέρειες σύμφωνα με τις τελευταίες αναπροσαρμογές (Καλλικράτης) ή να χρησιμοποιηθεί ο Νομός σαν ευρύτερη περιοχή. Η χώρα ενώνεται με σχέση ένα προς πολλά, γιατί δεν μπορεί να υπάρχει η ίδια περιοχή σε άλλη χώρα. Το πεδίο που αποτελεί το ξένο κλειδί και υλοποιεί τη σύνδεση στους δύο αυτούς πίνακες είναι το id της χώρας ενώ και οι δύο πίνακες διαθέτουν το πεδίο όνομα για αναγνώριση της χώρας.

Στη συνέχεια, κάθε περιοχή αποτελείται από πόλεις, πάλι με σχέση ένα προς πολλά γιατί κάθε περιφέρεια ή νομός αποτελείται από πολλές πόλεις. Η σύνδεση πόλης-περιφέρειας υλοποιείται με το ξένο κλειδί idstate ενώ οι πίνακες μας πέρα από τα πρωτεύοντα κλειδιά τους έχουν και τα πεδία name, country, long, lat. Το name χρησιμοποιείται για το όνομα της πόλης, το country για προσδιορισμό της χώρας που ανήκει και τα long, lat αποτελούν τις γεωγραφικές συντεταγμένες της πόλης.

Κάθε πόλη αποτελείται από περιοχές. Αυτό το πεδίο είναι σημαντικό γιατί θα βοηθήσει στην επεκτασιμότητα της υπηρεσίας που αντί να δίνονται γενικά πληροφορίες για την πόλη θα είναι δυνατή η ανεύρεση πληροφοριών βάσει περιοχής. Για παράδειγμα, η Αθήνα, που σαν πόλη διαθέτει αρκετές τουριστικές περιοχές με αρκετά σημεία ενδιαφέροντος η κάθε μία. Φανταστείτε πόσο δύσκολο είναι να γίνει επιλογή των τοποθεσιών που πρόκειται προταθούν στο χρήστη, πόσες τοποθεσίες θα αποκλειστούν από την επιλογή, ενώ μπορεί να αξίζουν για επίσκεψη, και πόσους

περιορισμούς θα έχουμε και από τις διαθέσιμες ημέρες που έχει ο χρήστης. Ενώ με το πεδίο περιοχή αυτόματα μικραίνουμε τον όγκο και έτσι προβάλλουμε ένα πιο ολοκληρωμένο σύστημα προτάσεων τοποθεσιών με πλήρεις πληροφορίες για όλα τα σημεία ενδιαφέροντος της περιοχής της πόλης χωρίς να αποκλείουμε μέρη.

Για λόγους κληρονομικότητας, ευκολίας στην χαρτογράφηση της βάσης και ενσωμάτωσης της στο πρόγραμμα όλοι οι παρακάτω πίνακες που θα αναλυθούν και στην ουσία αποτελούν τα σημεία ενδιαφέροντος της κάθε περιοχής θα έχουν τα παρακάτω πεδία με κοινά ονόματα. Τα πεδία αυτά των πινάκων είναι:

- Όνομα για την ονομασία της τοποθεσίας
- Διεύθυνση της τοποθεσίας
- Τηλέφωνο για επικοινωνία
- Συντεταγμένες της τοποθεσίας
- Τιμή εισιτηρίου
- Ώρες λειτουργίας
- Περιγραφή της τοποθεσίας με μια μικρή ίσως ιστορική αναδρομή και πληροφορίες για τους τουρίστες
- Περιοχή της τοποθεσίας και
- Το ξένο κλειδί idroi που πραγματοποιεί την σύνδεση με το κεντρικό πίνακα της βάσης μας, τον Points of interests

Η ουσία στην όλη σχεδίαση της βάσης είναι στο ότι η πόλη περιέχει πολλά σημεία ενδιαφέροντος. Σε αυτό τον πίνακα δεν χρειαζόμαστε τίποτα άλλο εκτός από το τα δύο κλειδιά, πρωτεύον και ξένο. Με αυτό τον τρόπο διευκολύνεται η αντικειμενοστραφής υλοποίηση στο netbeans και υλοποιείται εύκολα η πολλά προς πολλά σχέση που τυχόν μπορεί να προκύψει. Τα σημεία ενδιαφέροντος μπορούν να είναι:

- Αρχαιολογικοί χώροι
- Μουσεία
- Θεματικά πάρκα/Λούνα παρκ
- Πάρκα περιπάτων/ζωολογικοί κήποι
- Εκκλησίες/Μοναστήρια
- Αγορές
- Εμπορικά κέντρα
- Γενικά τουριστικές περιοχές και

- Εναλλακτικές προτάσεις

Οι εναλλακτικές προτάσεις αποτελούν και ένα ιδιαίτερο ξεχωριστό πεδίο διότι περιλαμβάνει προτάσεις ντόπιων και δεν αποτελεί γνωστή τουριστική περιοχή αλλά τοποθεσία που θα ήταν γνωστή πιο πολύ στους ντόπιους, προσφέροντας έτσι τουριστική ανάπτυξη και σε άλλες περιοχές ενισχύοντας οικονομικά και επιχειρήσεις που δεν είχαν την τύχη να βρίσκονται σε ευρέως γνωστές περιοχές ενώ προσφέρουν πολύ καλύτερες υπηρεσίες. Για παράδειγμα καλό το να φας σουβλάκι στην Πλάκα αλλά γιατί να μην δοκιμάσεις και το τεράστια ποικιλία κρεατικών στο Πειραιά πάνω σε μια μικρή πλατεία.

8.4 Το resources project

Αυτό το project περιλαμβάνει όλα τα αρχεία και τις πηγές που θα ενσωματωθούν στην υπηρεσία. Οι πηγές αυτές είναι.

- src/META-INF/resources directory:
- src/ValidationMessages.properties Bean Validation messages.
- src/META-INF/resources/css: Εδώ έχουμε τις πηγές που αφορούν το στυλ που θα έχει η εφαρμογή μας όπως χρώματα, γραμματοσειρές κ.α. Τα δύο αυτά στυλ είναι τα default.css και το jsfcrud.css.
- src/META-INF/resources/img που περιλαμβάνει τις εικόνες της εφαρμογής μας
- src/META-INF/resources/js που έχει τα javascript αρχεία που θα χρησιμοποιηθούν, util.js και
- src/META-INF/resources/util που περιλαμβάνει τα διάφορα στοιχεία/εργαλεία της υπηρεσίας.

8.5 Το Google service project

Αυτό το project περιλαμβάνει όλες τις υλοποιήσεις των διαφόρων υπηρεσιών της Google που θα χρησιμοποιηθούν από την υπηρεσία μας. Προκειμένου να πραγματοποιηθεί αυτό θα πρέπει, αν δεν έχουμε, να δημιουργηθεί ένας λογαριασμός στην Google μέσω του οποίου θα γίνει η εγγραφή στις διάφορες υπηρεσίες. Μετά την εγγραφή θα αποσταλεί από την Google το κλειδί, το οποίο θα χρησιμοποιήσουμε για να μπορέσουμε να κάνουμε χρήση των απαραίτητων βιβλιοθηκών. Για κάθε υπηρεσία υπάρχει ξεχωριστή διαδικασία ενσωμάτωσης της στην εφαρμογή μας. Στην παρούσα φάση δεν έχει γραφτεί ο κώδικας της εφαρμογής και γι' αυτό δεν έχουμε κάνει και εγγραφή στις υπηρεσίες αυτές ώστε να περιγράψουμε αναλυτικά τις διαδικασίες και τον κώδικα.

Το μόνο που είναι σίγουρο είναι ότι η ενσωμάτωση είναι πολύ απλή ακολουθώντας τις οδηγίες της Google και δεν χρειάζονται ιδιαίτερες γνώσεις για τη υλοποίηση. Για περισσότερες πληροφορίες μπορείτε να ανατρέξετε στον url της Google www.googlemaps/documentation.

8.6 Το booking service project

Σε αυτό το project δεν θα πραγματοποιηθεί τίποτα περισσότερο από μια σύνδεση με διάφορες υπηρεσίες κρατήσεων είτε αυτό αφορά ξενοδοχεία ή άλλες υπηρεσίες. Η υλοποίηση θα πραγματοποιηθεί με χρήση abstract μεθόδων για την ενσωμάτωση τους στη δική μας υπηρεσία και χρήση των υπερσυνδέσεων για μετάβαση σε αυτές τις ιστοσελίδες και παροχή επιπλέον πληροφοριών.

8.7 Το Tourism Wizard project

Το συγκεκριμένο Project αποτελεί και τον πυρήνα της όλης εφαρμογής. Εδώ θα υλοποιείται και το περιβάλλον του χρήστη και περιλαμβάνει όλες τις προαναφερθείσες λειτουργίες και υπηρεσίες.

Το κεντρικό περιβάλλον του χρήστη περιλαμβάνει συνοπτικά τις ακόλουθες λειτουργίες:



Σχήμα 7 Λειτουργίες χρήστη

Από τη πλευρά του διαχειριστή και την αλληλεπίδρασή του με τη βάση του δίνεται η δυνατότητα εκτέλεσης των ακόλουθων λειτουργιών:

- Δημιουργία
- Διαγραφή
- Αναβάθμιση
- Τροποποίηση

Οι παραπάνω λειτουργίες εκτελούνται σε όλες τις οντότητες που έχουν δημιουργηθεί στη βάση μας και αναφέρθηκαν παραπάνω.

Η υπηρεσία μας θα χρησιμοποιεί stateless session beans σαν facades για αλληλεπίδραση με τα entities του persistence και τα CDI managed beans σαν controllers για αλληλεπίδραση με τις σελίδες facelets. Έτσι πραγματοποιείται η υλοποίηση της αρχιτεκτονικής του Model-View-Controller που αναφέρθηκε στην αρχή και έχουμε εφαρμογή της ίδιας αρχιτεκτονικής σε όλες τις σελίδες μας. Πιο αναλυτικά έχουμε:

- AbstractFacade που είναι μια abstract class που παίρνει τον τύπο <T> και υλοποιεί το λεγόμενο CRUD δηλαδή create, edit, update, delete, όπου το <T> είναι μια jpa οντότητα.
- WizardBean είναι ένα stateless bean που κάνει extend το παραπάνω facade και υλοποιεί το Wizard σαν τύπο <T> και δημιουργείται το Persistence context για τον entity manager. Σε αυτό το bean υλοποιούνται οι δικές μας μέθοδοι και τα εκτελέσιμα query για τον search αλγόριθμο που θα υλοποιηθεί
- WizardController είναι ένα CDI managed bean που αλληλεπιδρά με τα κατάλληλα facelets και enterprise beans για τον έλεγχο των κατάλληλων δεδομένων και εμφάνιση αυτών από την εφαρμογή.

Τα Enterprise Beans που θα υλοποιηθούν στην υπηρεσία μας φαίνονται παρακάτω και παρέχουν το business logic της εφαρμογής μας. Όλα τα beans αυτά είναι stateless session.

- Alter_suggestionBean
- MouseiaBean
- MallsBean
- Tourist_areasBean
- Market_placesBean
- Fun_parksBean
- Archeological_sitesBean
- ParksBean
- ChurchesBean
- Points_of_interestBean
- CountryBean
- StateBean
- CitiesBean
- AreasBean
- WizardBean

Το wizard bean περιέχει όλες τις πληροφορίες που θα υλοποιηθούν σε όλα τα αιτήματα κατά την διάρκεια εκτέλεσης της εφαρμογής.

Τα Facelets είναι τα αρχεία που αποτελούν και τις σελίδες μας και θα είναι υπεύθυνα για αυτό που βλέπει ο χρήστης. Οι σελίδες αυτές θα είναι οργανωμένες σε καταλόγους για την γρήγορη εύρεση και εύκολη χρησιμοποίηση τους από όλη την εφαρμογή. Η εφαρμογή μας θα κάνει χρήση αρκετών facelets διότι θα περιλαμβάνει αρκετές ιστοσελίδες που θα μεταβαίνει ο χρήστης.

Τα Managed Beans που θα χρησιμοποιηθούν στην υπηρεσία μας είναι τα κάτωθι τα οποία ανταποκρίνονται πλήρως στα αντίστοιχα enterprise beans της εφαρμογής.

- Alter_suggestionController
- MouseiaController
- MallsController
- Tourist_areasController
- Market_placesController
- Fun_parksController
- Archeological_sitesController
- ParksController
- ChurchesController
- Points of interestController
- CountryController
- StateController
- CitiesController
- areasController
- WizardController

Υποβοηθητικές κλάσεις στην εφαρμογή μας είναι οι ακόλουθες.

- AbstractPaginationHelper που είναι μια abstract class με μεθόδους που χρησιμοποιούνται από τα managed beans.
- FileUploadServlet, ImageServlet Classes είναι οι κλάσεις που θα χρησιμοποιηθούν για την επεξεργασία των εικόνων της εφαρμογής μας. Ο FileUploadServlet φορτώνει μια εικόνα και την αποθηκεύει με το περιεχόμενο της στη βάση. Ο ImageServlet καλεί με τη σειρά του την εικόνα όταν χρειάζεται να την προβάλει.
- JsfUtil Class είναι η κλάση που χρησιμοποιείται για τις λειτουργίες των JavaServer Faces.

Επίσης υπάρχουν και τα Properties Files. Η εφαρμογή μας θα διαθέτει ένα αρχείο με τις ιδιότητες με την ονομασία src/java/Bundle.properties. Σε αυτό θα περιέχονται οι ιδιότητες του περιβάλλοντος του χρήστη με τις γλώσσες που πρόκειται να εισάγουμε για προβολή δηλαδή σαν κύρια την Ελληνική και με επιλογές την Αγγλική με μελλοντική επέκταση στη Γερμανική και Γαλλική.

Η όλη υλοποίηση και χρήση της εφαρμογή βρίσκεται στο κατάλογο web/WEB-INF directory και περιέχει:

- beans.xml που είναι ο deployment descriptor.
- faces-config.xml που είναι το απαραίτητο αρχείο για τα Jsf .
- glassfish-web.xml με τα απαραίτητα αρχεία για να τρέξει ο GlassFish Server.
- jaxws-catalog.xml που είναι ο deployment descriptor για JAX-WS web service client.
- web.xml που είναι το αρχείο για να τρέξει στο δίκτυο η εφαρμογή.

8.8 Αλγόριθμος Search

Η αναζήτηση ενεργοποιείται με ένα κάλεσμα της υπηρεσίας και πιο συγκεκριμένα της μεθόδου search() που δημιουργήσαμε. Πιο κάτω, βλέπουμε απόσπασμα του κώδικα της εφαρμογής μας για κλήση της υπηρεσίας.

```
service = new google.maps.places.PlacesService(map);  
service.search(request, callback);
```

Η μέθοδος μας περιλαμβάνει ένα αίτημα που περιέχει κάποια πεδία βάσει των οποίων θα πραγματοποιηθεί η αναζήτηση

```
var map;  
var service;  
var infowindow;
```

```
function initialize() {  
    var pyrmon = new google.maps.LatLng(LONG,LAT);//Εισαγωγή συντεταγμένων από τη βάση  
  
    map = new google.maps.Map(document.getElementById('map'), {  
        mapTypeId: google.maps.MapTypeId.ROADMAP,//τύπος χάρτη  
        center: pyrmon,  
        zoom: 15// επιλογή επιθυμητού zoom  
    });
```

```
var request = {
  location: pyrmont,
  radius: '500',//εισαγωγή ακτίνας που θα ψάξει για αποτελέσματα
  types: ['store']//εισαγωγή τύπου τοποθεσίας προς αναζήτηση
};

service = new google.maps.places.PlacesService(map);
service.search(request, callback);
}

function callback(results, status) {
  if (status == google.maps.places.PlacesServiceStatus.OK) {
    for (var i = 0; i < results.length; i++) {
      var place = results[i];
      createMarker(results[i]);
    }
  }
}
```

Στον παρακάτω αλγόριθμο θα γίνεται αναζήτηση βάσει των εισαγόμενων στοιχείων. Η μέθοδος μας θα ονομάζεται TextSearch().

```
service = new google.maps.places.PlacesService(map);
service.textSearch(request, callback);
```

Αυτή η μέθοδος χρειάζεται τα παρακάτω πεδία:

- query (required): Που περιλαμβάνει το είδος της τοποθεσίας που επιθυμούμε για παράδειγμα 'εστιατόριο»
- Συντεταγμένες: τις οποίες θα παίρνει από τη βάση ή από το geocoding βάσει της διεύθυνσης της τοποθεσίας
- Τοποθεσία και ακτίνα: παίρνοντας σαν αντικείμενα τις συντεταγμένες και την ακτίνα σαν ακέραιο

Ο κώδικας για την υλοποίηση της υπηρεσίας της Google φαίνεται παρακάτω:

```
var map;
var service;
var infowindow;

function initialize() {
  var pyrmon = new google.maps.LatLng(-33.8665433,151.1956316);//εισαγωγή δικών μας
  συντεταγμένων

  map = new google.maps.Map(document.getElementById('map'), {
    mapTypeId: google.maps.MapTypeId.ROADMAP,//τύπος χάρτη
    center: pyrmon,
    zoom: 15
  });

  var request = {
    location: pyrmon,
    radius: '500',//εισαγωγή ακτίνας
    query: 'restaurant'//εισαγωγή λέξης κλειδί
  };

  service = new google.maps.places.PlacesService(map);
  service.textSearch(request, callback);
}

function callback(results, status) {
  if (status == google.maps.places.PlacesServiceStatus.OK) {
    for (var i = 0; i < results.length; i++) {
      var place = results[i];
      createMarker(results[i]);//επιστροφή και τοποθέτηση marker στο χάρτη
    }
  }
}
```

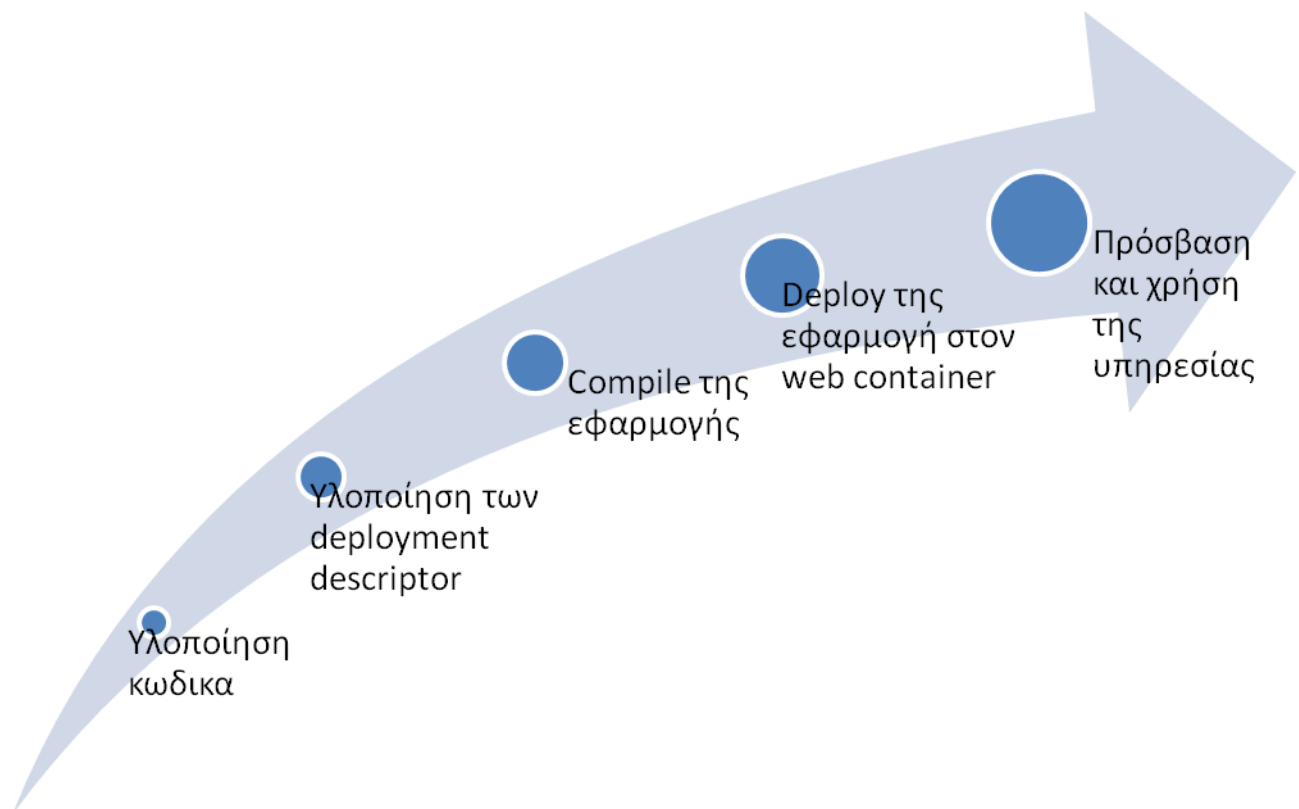
8.9 Απαιτούμενες εκδόσεις λογισμικών

Ήδη στο προηγούμενο κεφάλαιο αναλύθηκαν τα εργαλεία που θα χρησιμοποιηθούν επιπλέον όμως πρέπει να αναφερθούν και οι εκδόσεις της JAVA που πρέπει να είναι εγκατεστημένες στο σύστημα που θα πραγματοποιηθεί η ανάπτυξη της υπηρεσίας και πιο συγκεκριμένα αυτές είναι:

- Java platform standard edition και πιο συγκεκριμένα το java platform standard edition 6.0 developmentkit ή το 7.0 από το www.oracle.com/technetwork/java/javase/downloads/index.html
- JavaEE software development Kit από το www.oracle.com/technetwork/java/javaee/downloads/index.html
- JavaEE 6 Tutorial Component και
- NetBeans με glassfish server από το www.netbeans.org/downloads/inex.xhtml

8.10 Υλοποίηση

Η όλη διαδικασία ολοκλήρωσης της εφαρμογής μας μπορεί να συγκεντρωθεί στο παρακάτω διάγραμμα.

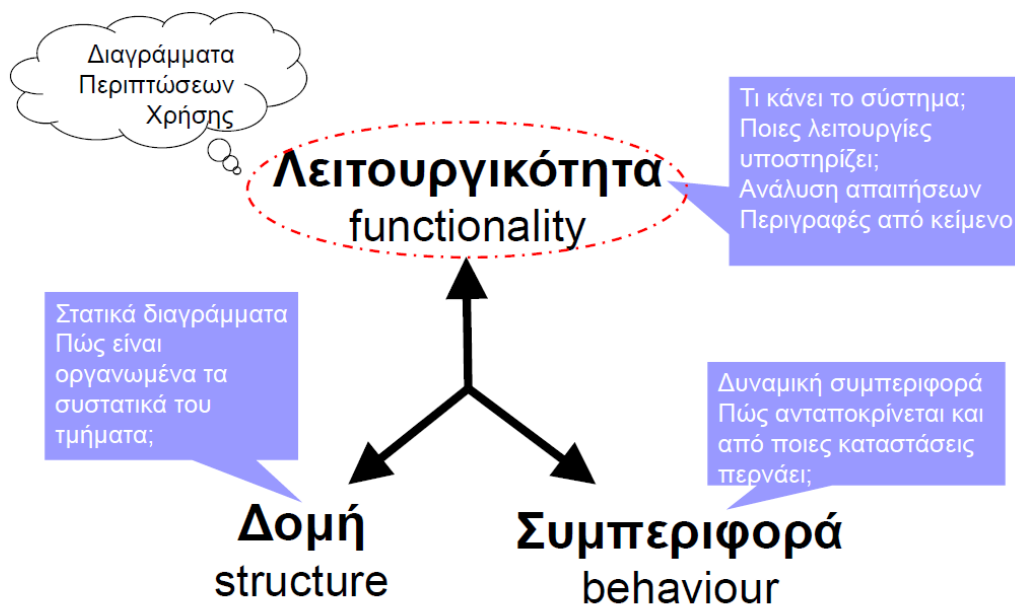


Σχήμα 8 Φάσεις ανάπτυξης εφαρμογής.

9

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗ ΑΝΑΛΥΣΗ
ΣΥΣΤΗΜΑΤΟΣ

9.1 Άξονες Μοντελοποίησης



Σχήμα 9 Άξονες μοντελοποίησης

9.2 Συλλογή απαιτήσεων

- Οι “απαιτήσεις” είναι ο «λόγος ύπαρξης» κάθε έργου ανάπτυξης λογισμικού
- Ορίζουν και οριοθετούν user-requirements
- Ορίζουν τη λειτουργικότητα που απαιτείται
- Προσδιορίζουν τους στόχους που πρέπει να επιτευχθούν
- ΠΡΕΠΕΙ να είναι κατανοητές πλήρως και με ακρίβεια από την ομάδα που παρέχει τη λύση
- Οι απαιτήσεις χρηστών (και τα μέλη των ομάδων) αλλάζουν διαρκώς; Επομένως οι απαιτήσεις πρέπει να είναι καλά τεκμηριωμένες (well-documented)

- Πλήρης και όχι διφορούμενη κατανόηση των απαιτήσεων είναι ζωτική έτσι ώστε όλοι να ξέρουν ΤΙ ΚΑΝΟΥΝ ΚΑΙ ΓΙΑΤΙ!
- Πρέπει να υπόκεινται κριτική ΞΑΝΑ ΚΑΙ ΞΑΝΑ ΚΑΙ ΞΑΝΑ πριν αρχίσει ο σχεδιασμός και η υλοποίηση!
- Εμπειρέχει τη συμμετοχή domain-experts για να επιβεβαιώνουν ότι οι απαιτήσεις έχουν γίνει σωστά αντιληπτές!

9.3 Μοντελοποίηση

Ένα μοντέλο είναι μία αναπαράσταση σε ένα συγκεκριμένο μέσο, κάποιας οντότητας στο ίδιο ή σε άλλο μέσο.

Το μοντέλο αναπαριστά τα σημαντικά (από μία ορισμένη οπτική γωνία), στοιχεία της οντότητας που μοντελοποιείται, απλοποιώντας ή παραλείποντας τα υπόλοιπα.

Τα πλεονεκτήματα της μοντελοποίησης είναι ο ακριβής καθορισμός των απαιτήσεων έτσι ώστε όλοι οι εμπλεκόμενοι να τις κατανοούν με κοινό τρόπο.

Μοντέλα ενός συστήματος λογισμικού κατασκευάζονται ώστε οι αναλυτές, προγραμματιστές, διαχειριστές έργων, πελάτες και τελικοί χρήστες να κατανοήσουν το σύστημα από την οπτική γωνία που τους ενδιαφέρει.

9.4 UML

Η UML αποτυπώνει:

1. τη στατική δομή, και
2. τη δυναμική συμπεριφορά ενός συστήματος.

Ένα αντικειμενοστραφές σύστημα μοντελοποιείται ως:

- συλλογή αντικειμένων που αλληλεπιδρούν για την εκτέλεση μίας λειτουργίας η οποία είναι τελικά αξιοποιήσιμη από τον χρήστη του συστήματος.

Η στατική δομή καθορίζει τα είδη των αντικειμένων που είναι σημαντικά για το σύστημα καθώς και τις συσχετίσεις μεταξύ τους.

Η δυναμική συμπεριφορά προσδιορίζει την εξέλιξη των αντικειμένων σε σχέση με τον χρόνο και την επικοινωνία μεταξύ τους.

9.5 Διάγραμμα Χρήσεων

Τα διαγράμματα περιπτώσεων χρήσης στη UML χρησιμοποιούνται για τη μοντελοποίηση της συμπεριφοράς ενός συστήματος, υποσυστήματος ή κλάσης, όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη. Τα διαγράμματα περιπτώσεων χρήσης διαμερίζουν τη λειτουργικότητα του συστήματος σε συναλλαγές που έχουν νόημα για τους χαρακτήρες (actors) - ιδανικούς χρήστες του συστήματος. Τα επιμέρους τμήματα της λειτουργικότητας ονομάζονται περιπτώσεις χρήσης.

Το σύνολο των περιπτώσεων χρήσης συνιστούν τη συμπεριφορά του συστήματος. Ο τυπικός ορισμός μιας περίπτωσης χρήσης είναι μία ακολουθία συναλλαγών που πραγματοποιείται από το σύστημα για την παραγωγή μετρήσιμων αποτελεσμάτων που έχουν νόημα για τον χρήστη.

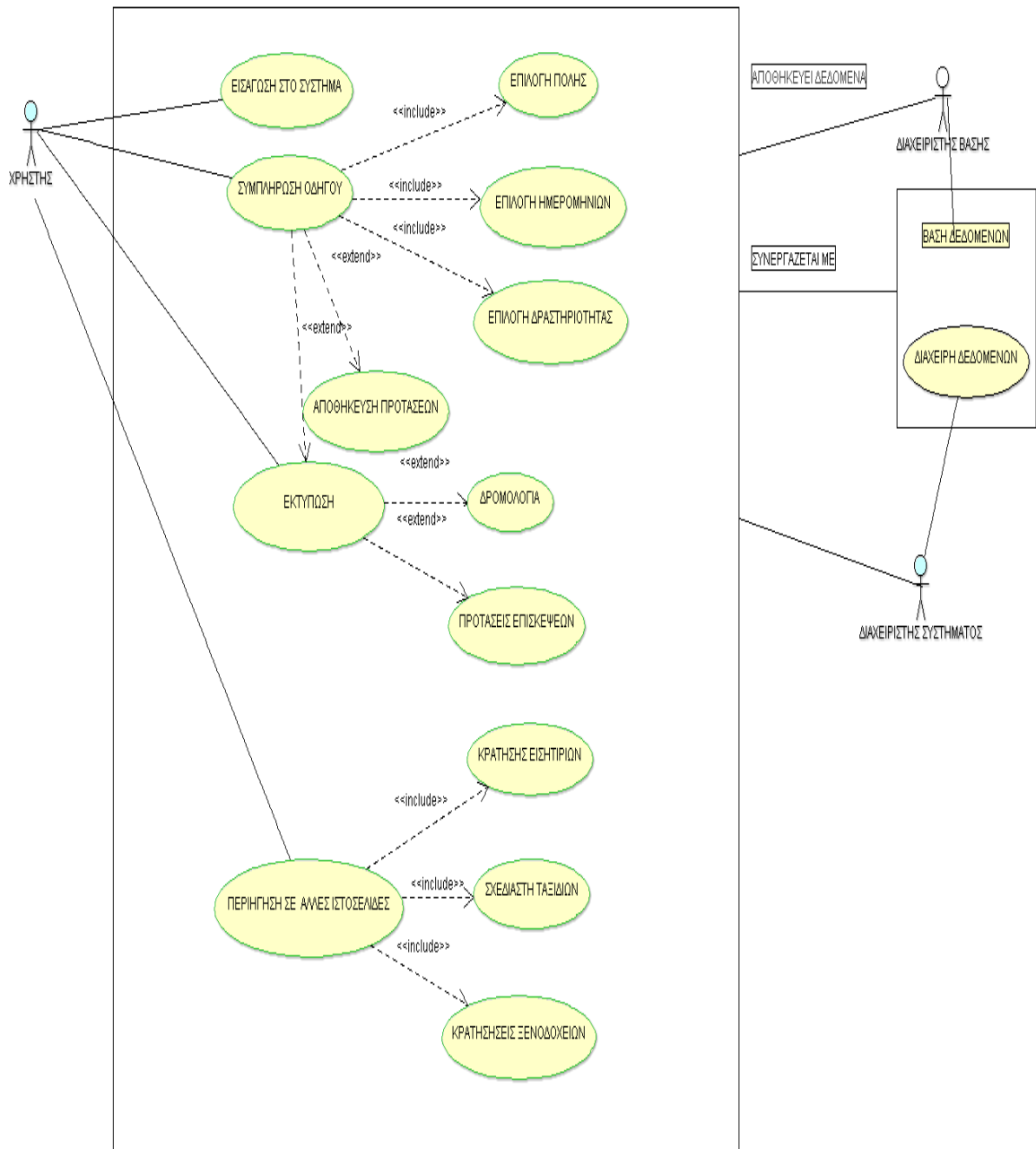
Σε κάθε διάγραμμα περίπτωσης χρήσης απεικονίζεται ένας χρήστης του συστήματος (άνθρωπος ή άλλο σύστημα) ως ένα σχηματικό ανθρωπάκι (stick person). Η ίδια η περίπτωση χρήσης, ως σύνολο λειτουργιών που έχουν κάποιο νόημα για το χρήστη, απεικονίζεται ως μία έλλειψη.

Ο χρήστης ενεργοποιεί μία περίπτωση χρήσης αναμένοντας την εκτέλεση κάποια συμπεριφοράς. Η συσχέτιση μεταξύ χρήστη και περίπτωσης χρήσης απεικονίζεται με μία ακμή μεταξύ τους ενώ η φορά της ενεργοποίησης με τη χρήση προσανατολισμένης ακμής.

Ο ορισμός μιας περίπτωσης χρήσης περιλαμβάνει όλες τις δυνατές συμπεριφορές που εμπερικλείει (κανονική και εναλλακτική συμπεριφορά). Η άποψη του συστήματος βάσει ενός μοντέλου περιπτώσεων χρήσης θεωρεί ότι η εκτέλεση κάθε περίπτωσης χρήσης είναι ανεξάρτητη από τις υπόλοιπες (ορθογωνικότητα), παρόλο που η υλοποίηση του συστήματος μπορεί να συνεπάγεται έμμεσες εξαρτήσεις μεταξύ τους λόγω κοινών αντικειμένων.

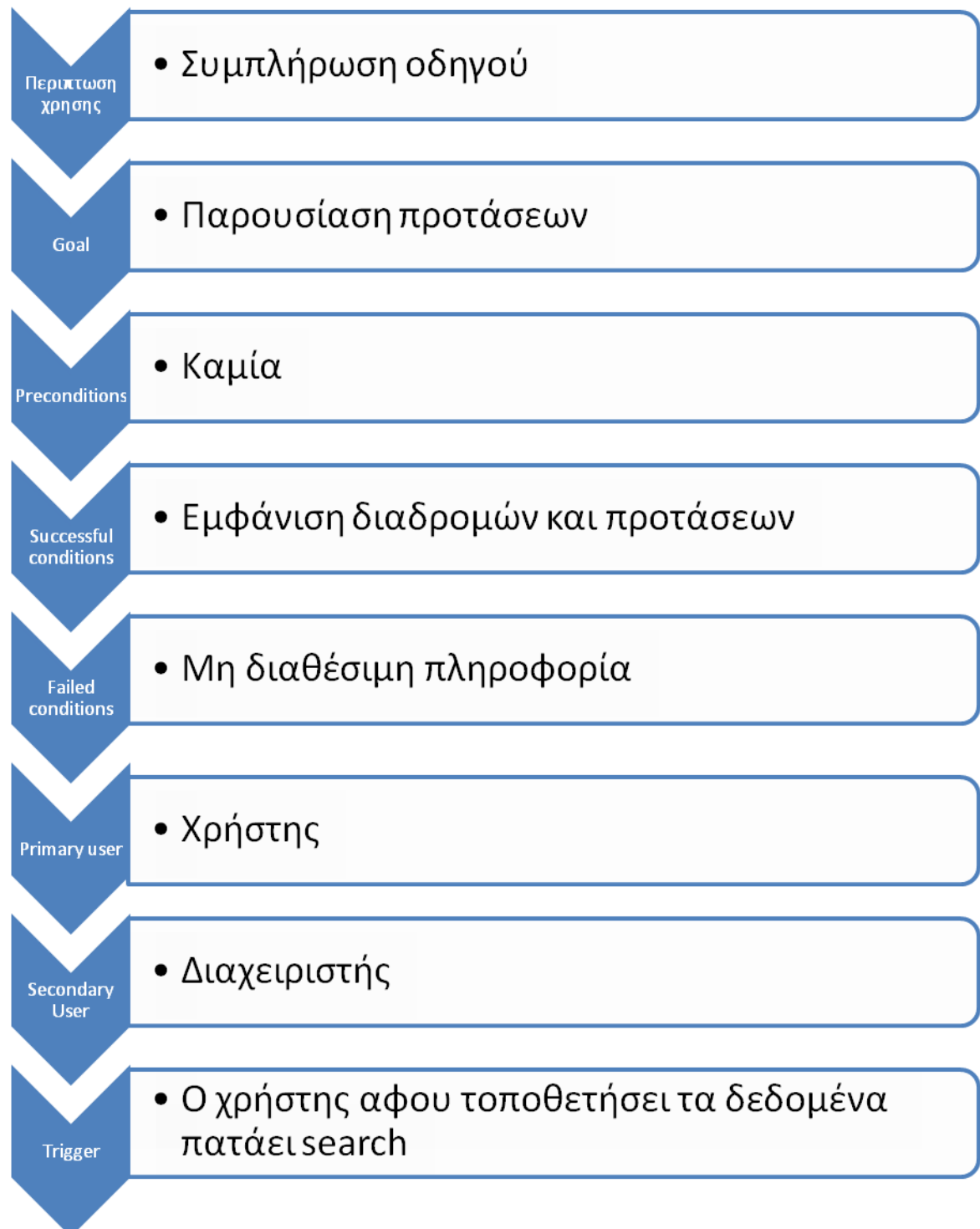
Η αναλυτική περιγραφή μιας περίπτωσης χρήσης προσδιορίζεται με άλλους τύπους διαγραμμάτων της UML όπως τα διαγράμματα καταστάσεων, διαγράμματα αλληλεπίδρασης (διαγράμματα ακολουθίας ή/και διαγράμματα συνεργασίας) ή και άτυπες περιγραφές κειμένου όπως θα δούμε και πιο κάτω.

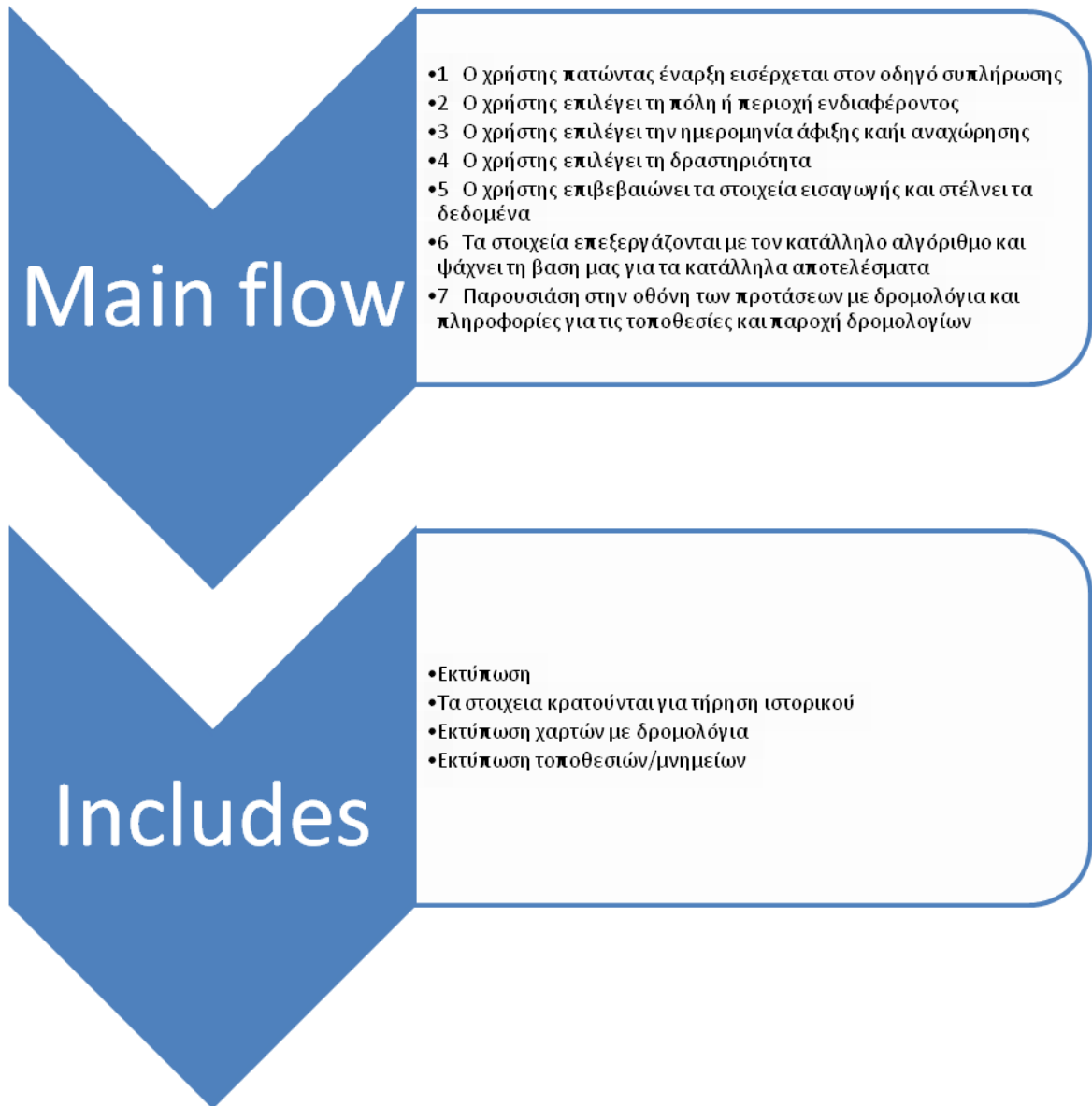
Ας δούμε πως υλοποιείται στη δική μας εφαρμογή το διάγραμμα χρήσης.



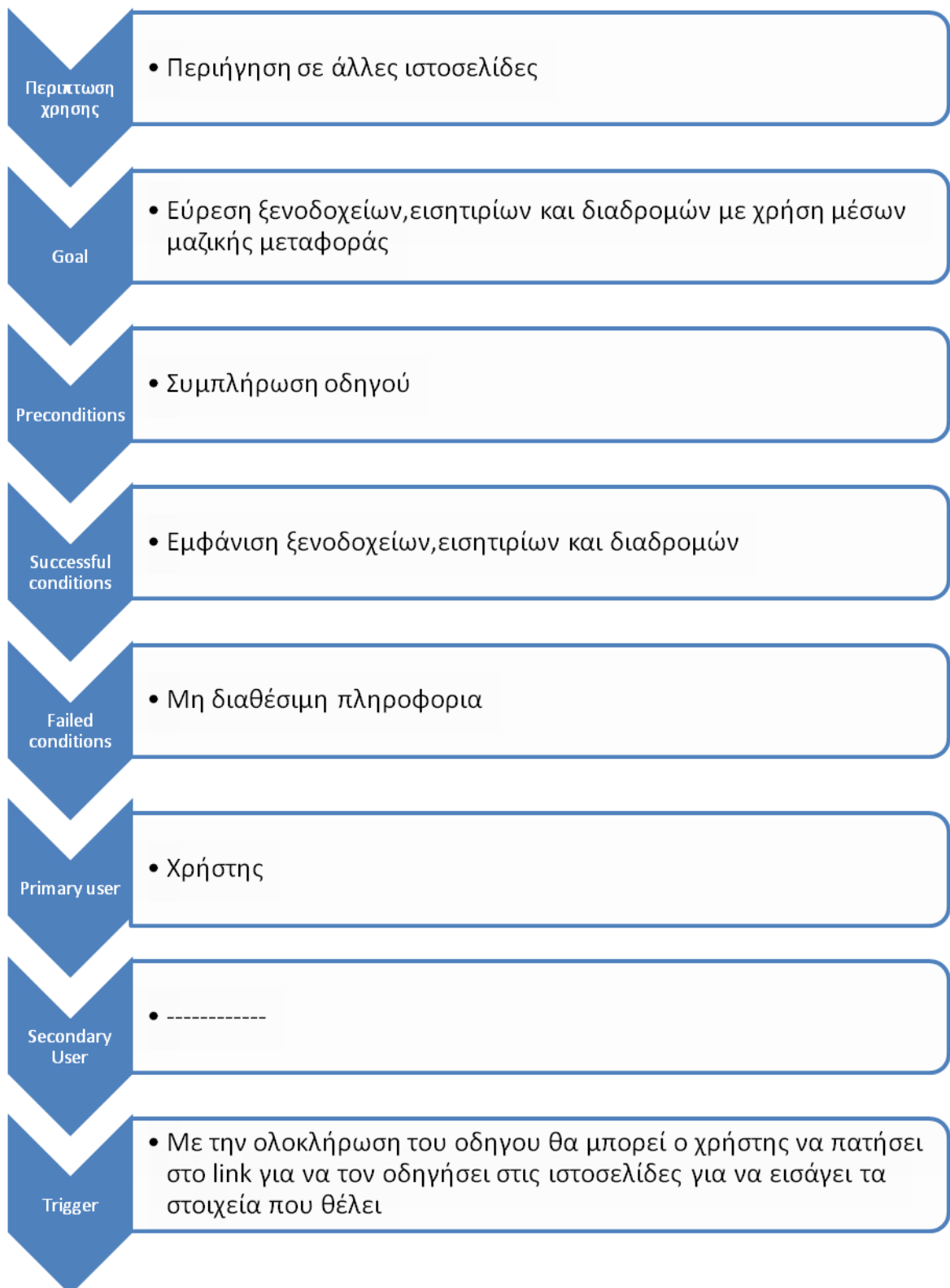
Σχήμα 10 Use Case Diagram

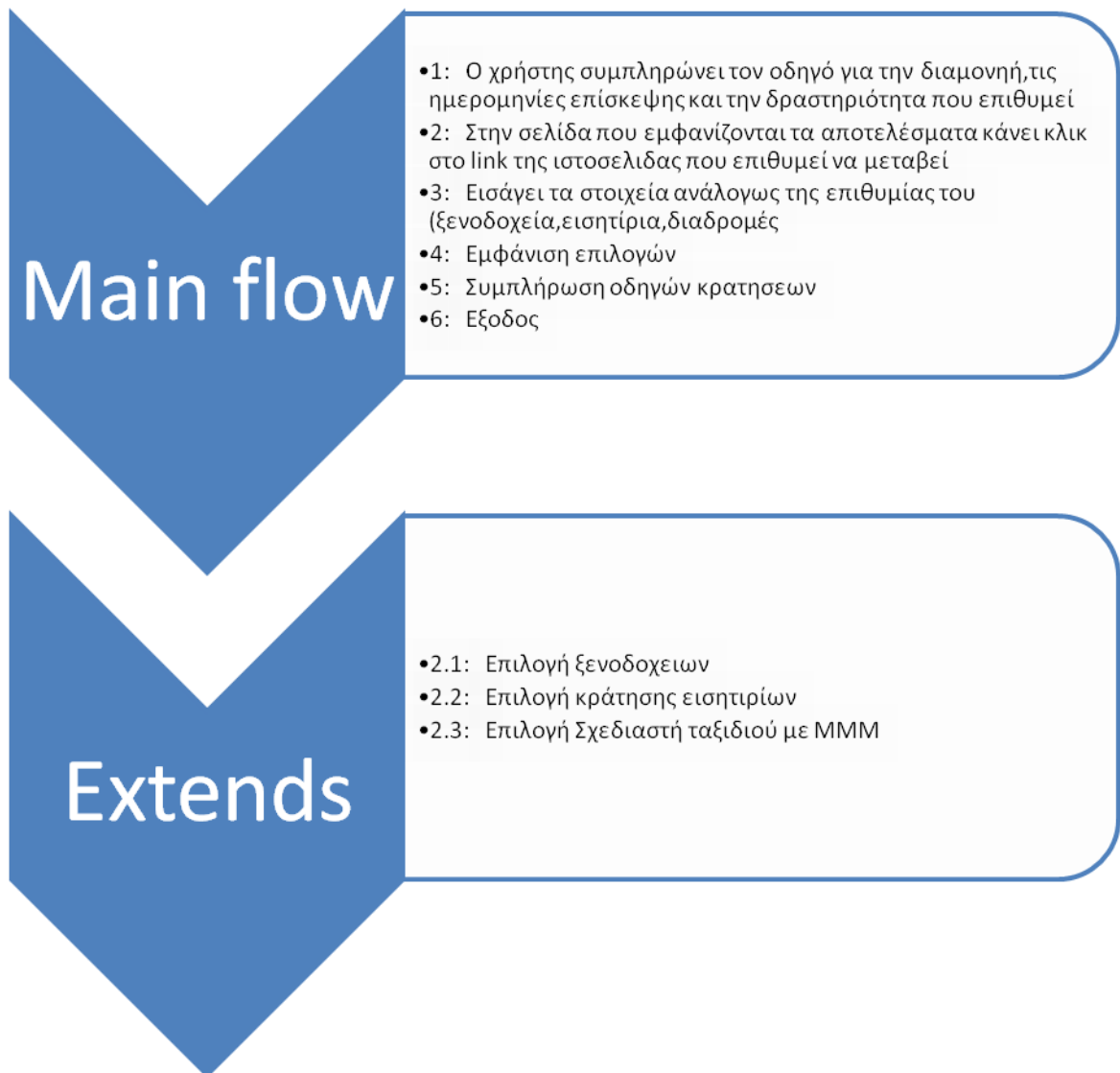
Ανάλυση περιπτώσεων

1^η Περίπτωση χρήσης



Σχήμα11 Πρώτη περίπτωση χρήσης

2^η Περίπτωση χρήσης

Σχήμα 12 2^η περίπτωση χρήσης

9.6 Διάγραμμα Κλάσεων

Η στατική άποψη ενός μοντέλου είναι θεμελιώδης στη UML καθώς αποτυπώνει την αρχιτεκτονική του συστήματος (μονάδες + μεταξύ τους σχέσεις)

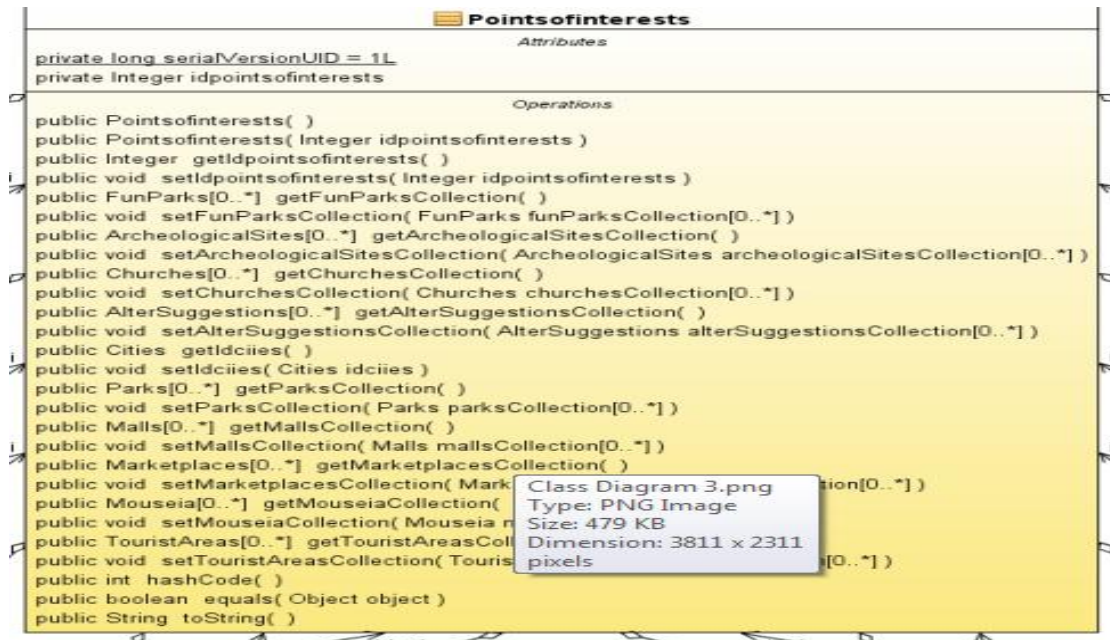
Σε ένα αντικειμενοστρεφές σύστημα τα δομικά του στοιχεία είναι οι κλάσεις και οι σχέσεις μεταξύ των κλάσεων επιτρέπουν τη συνεργασία των αντικειμένων τους

Τα διαγράμματα κλάσεων αποτυπώνουν τη στατική δομή

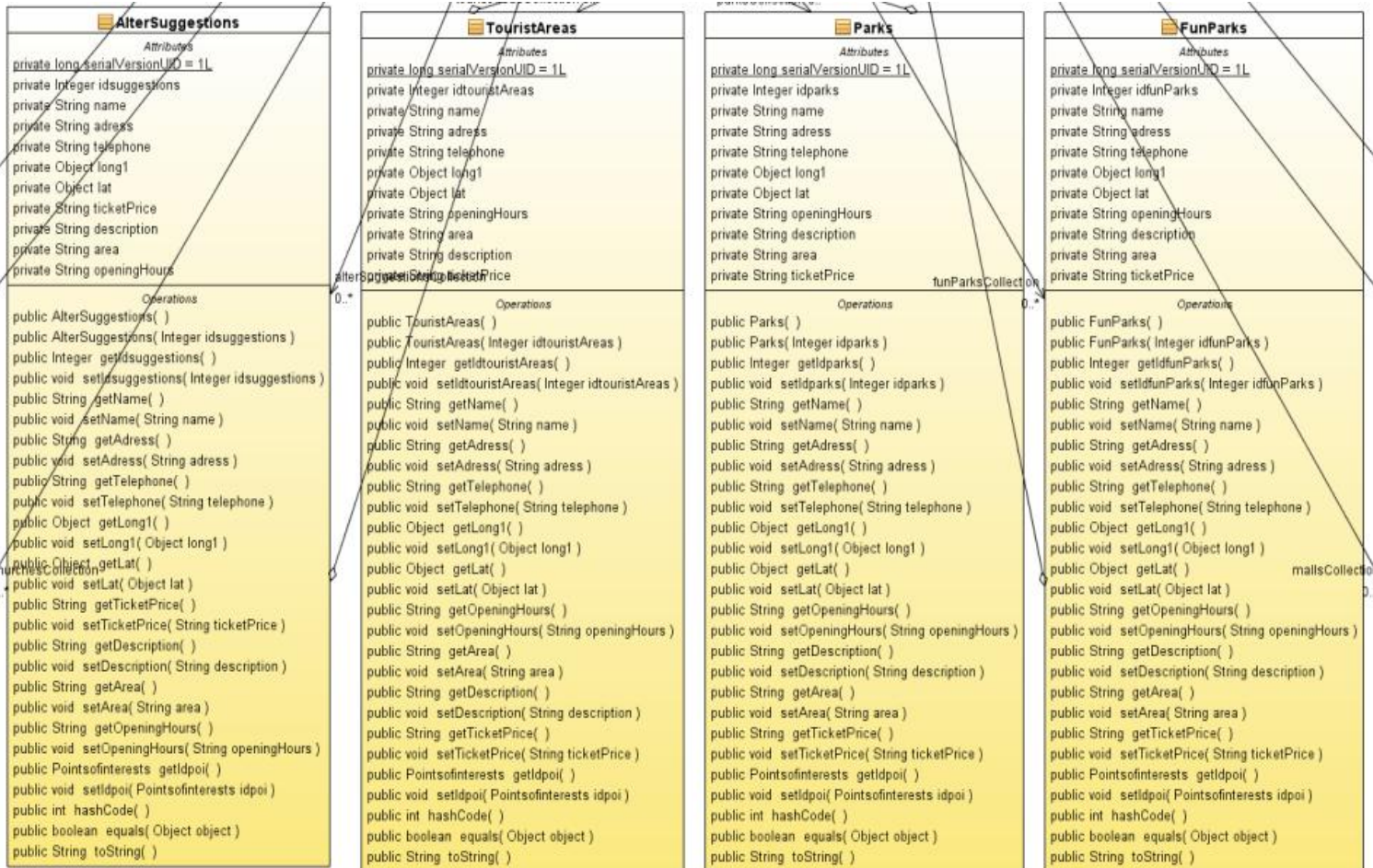
Πολύ συχνά, τα διαγράμματα κλάσεων είναι το μόνο είδος διαγραμμάτων που χρησιμοποιείται λόγω των πληροφοριών που παρέχει σχετικά με τον κώδικα

Πρώτα θα δούμε το διάγραμμα κλάσεων που δημιουργήθηκε από τις οντότητες της βάσης μας. Αυτό το διάγραμμα κλάσεων ενσωματώνει και τις μεθόδους και το κώδικα που απαιτείται για να γίνει σωστά μια χαρτογράφηση της βάσης και εισαγωγή της στο NetBeans. Χρησιμοποιώντας το persistence γλιτώσαμε ώρες εργασίας και πολλές σειρές κώδικα και κάνοντας εισαγωγή των παρακάτω κλάσεων έχουμε έτοιμη τη βάση σύμφωνα με την αρχιτεκτονική που αναφέραμε.

Λίγο πιο αναλυτικά

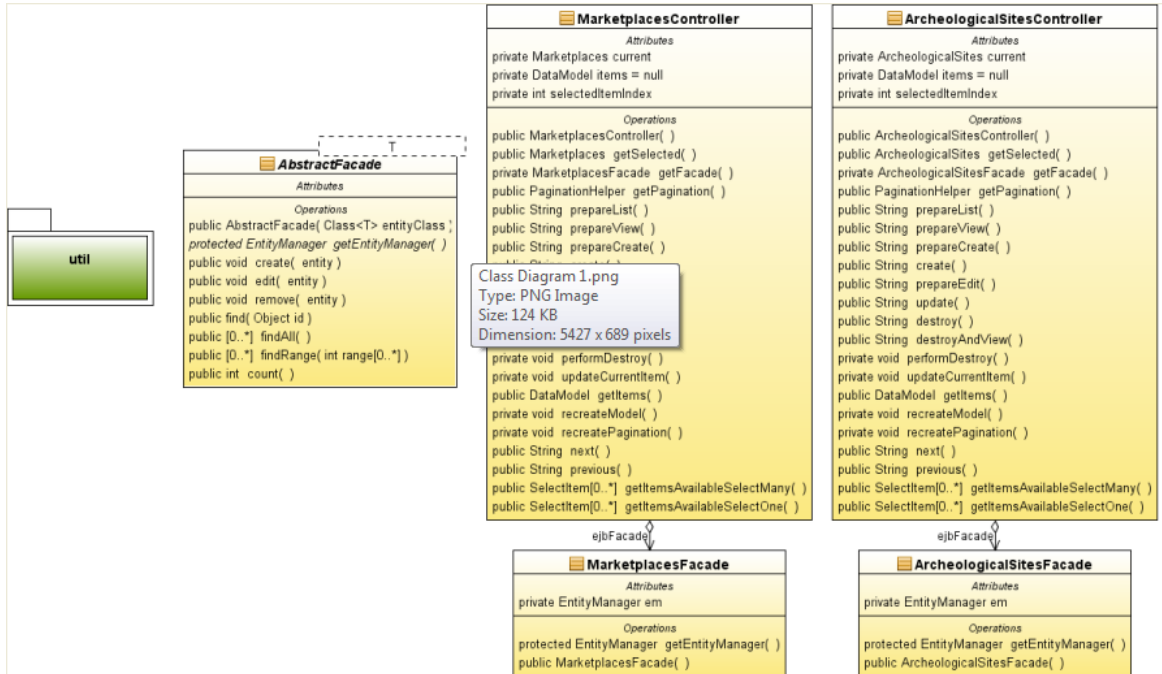


Class Diagram 3.png
Type: PNG Image
Size: 479 KB
Dimension: 3811 x 2311 pixels

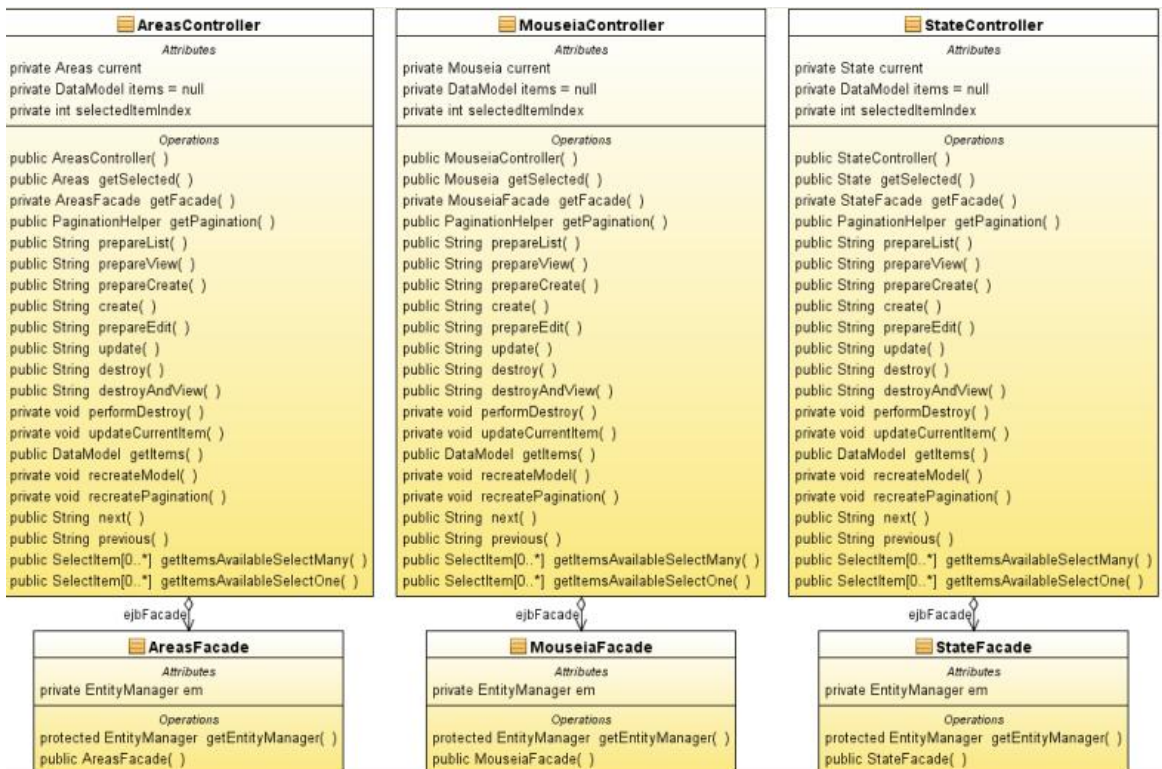


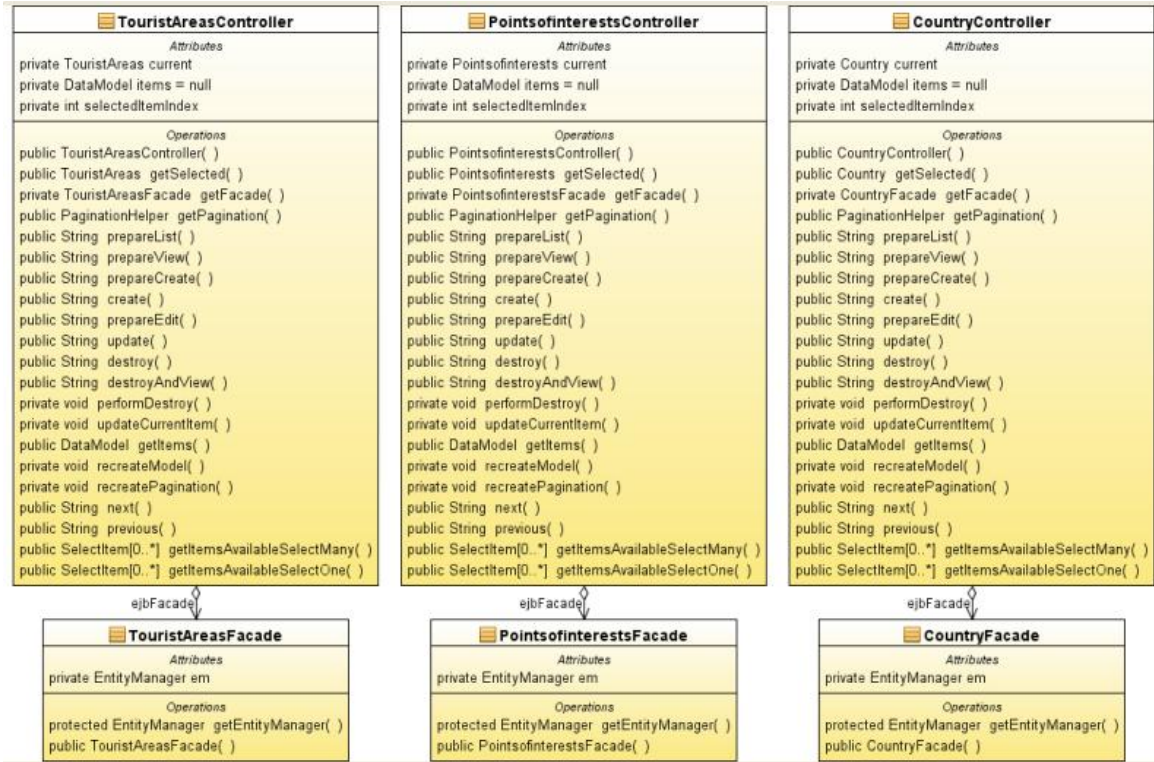
Σχήμα 14 Class diagram

Και το διάγραμμα με τη χρήση Jsf με τη χρήση των controller και των facades που ήδη έχουμε αναφέρει.

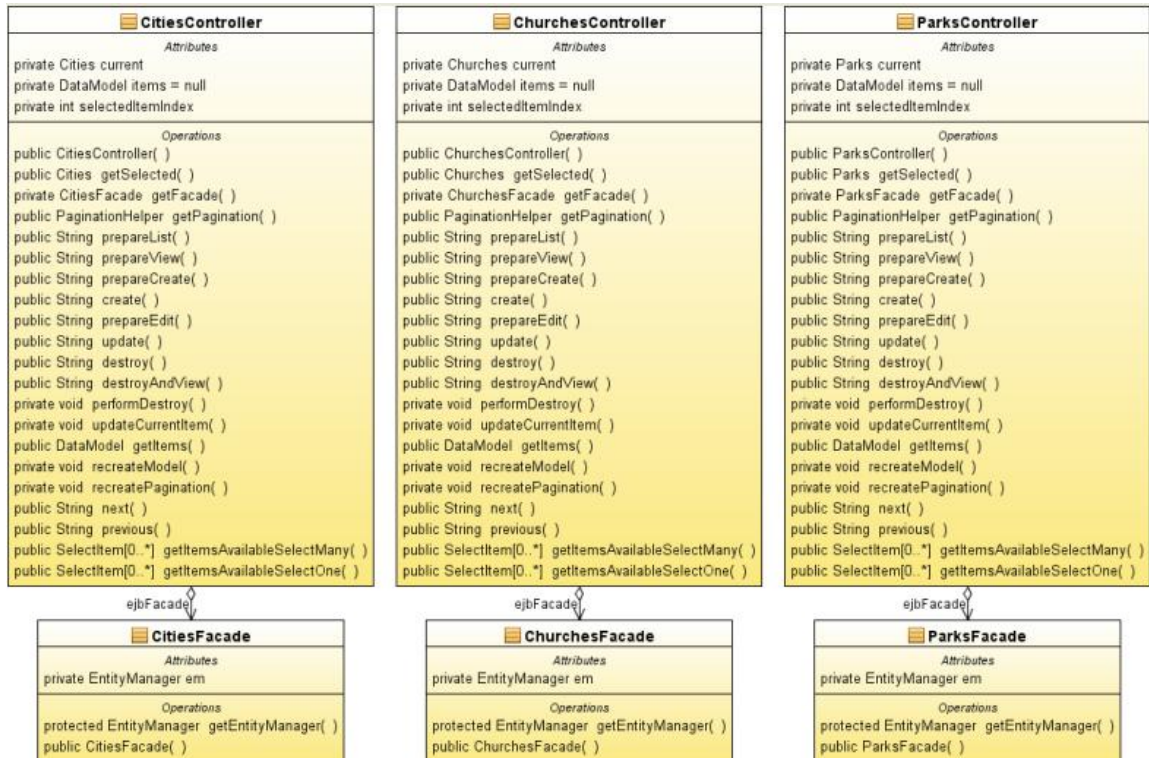


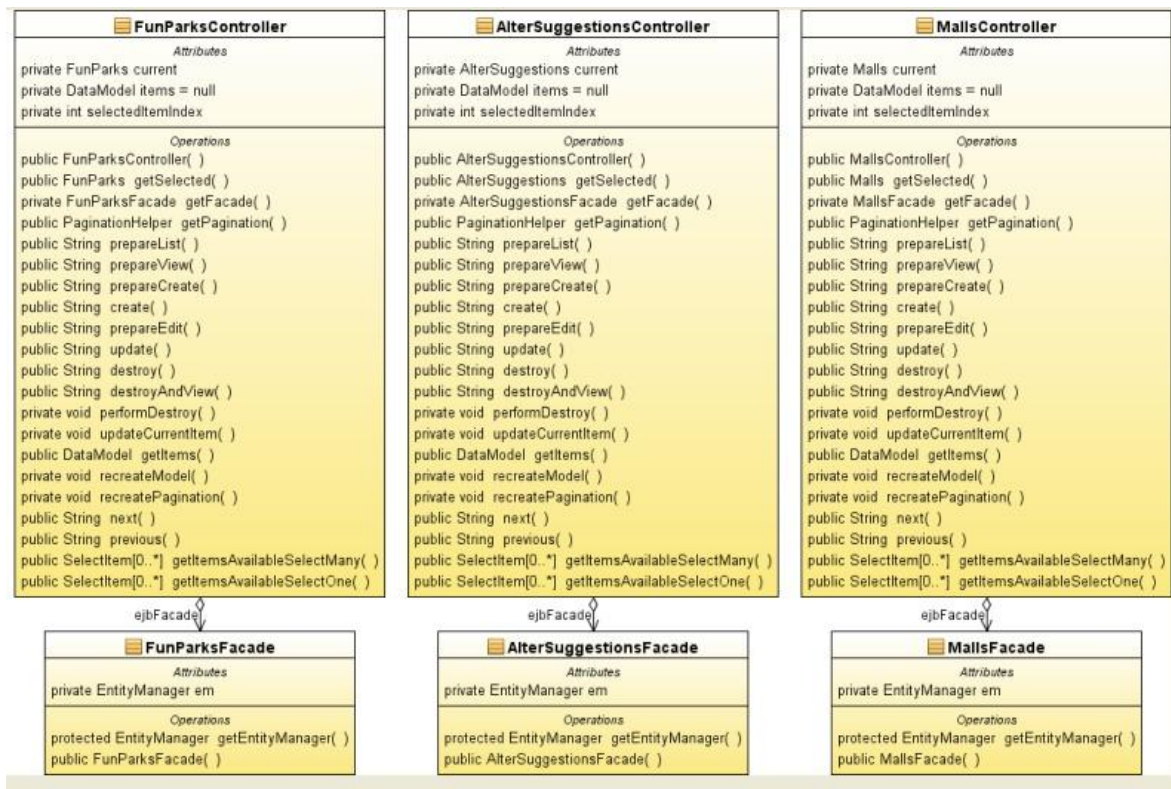
Σχήμα 15 JSF Class diagram





Σχήμα 16 JSF Class diagram

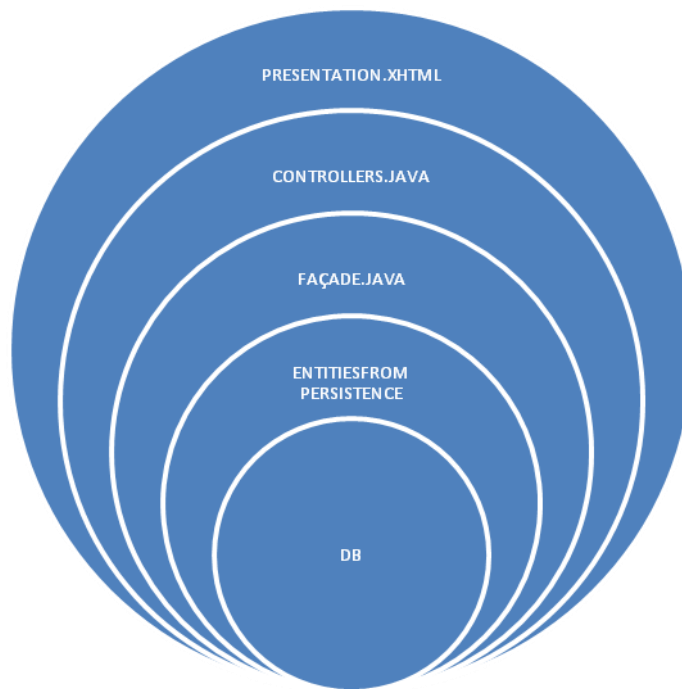




Σχήμα17 Υλοποίηση κλάσεων JSF

Στα παραπάνω διαγράμματα διαπιστώνουμε τις εξής κλάσεις, τους controller που αφορούν το presentation layer προγράμματος και περιέχουν όλες τις μεθόδους, το facade για το persistence και τις κλάσεις με τα αντίστοιχα beans. Για την αποφυγή κάποιων προβλημάτων θα δημιουργηθεί και το bug remover που αφαιρεί κάποια προβλήματα. Με τον ίδιο τρόπο δημιουργήθηκε και το διάγραμμα κλάσεων της εφαρμογής.

Η λειτουργία του όλου διαγράμματος συνεχίζεται στο παρακάτω σχήμα. Από τη βάση δημιουργούνται οι οντότητες με τη χρήση του persistence. Το facade περιέχει τον entity manager και υλοποιεί την επικοινωνία με τις αντίστοιχες κλάσεις του jsf που είναι οι controllers και στο τελευταίο επίπεδο έχουμε την παρουσίαση που είναι αυτό που βλέπει ο χρήστης.



Σχήμα 18 Λειτουργία και αρχιτεκτονική κλάσεων

9.7 Αλληλεπίδραση Συστήματος

Σε ένα αντικειμενοστραφές σύστημα, η συμπεριφορά υλοποιείται από την αλληλεπίδραση αντικειμένων. Αυτή η αλληλεπίδραση περιγράφεται στα πλαίσια ενός μοντέλου της UML με δύο είδη συμπληρωματικών διαγραμμάτων, τα διαγράμματα ακολουθίας και τα διαγράμματα συνεργασίας.

Τα διαγράμματα αυτά, και ιδιαίτερα τα διαγράμματα ακολουθίας δεν χρησιμοποιούνται μόνο για την τεκμηρίωση του συστήματος, αλλά και για τη διερεύνηση διαφόρων σεναρίων χρήσης του συστήματος με σκοπό τον εντοπισμό των λειτουργιών κάθε αντικειμένου

Τα διαγράμματα ακολουθίας (και συνεργασίας) είναι κατάλληλα για την απεικόνιση της στατικής συμπεριφοράς των αντικειμένων. Ως στατική συμπεριφορά ενός αντικειμένου νοείται το σύνολο των λειτουργιών που δεν εξαρτώνται από εσωτερικά ή εξωτερικά γεγονότα.

Η στατική άποψη περιγράφει τις εγγενείς ιδιότητες μιας κλάσης. Η περιγραφή συμπεριφοράς συνίσταται στο σύνολο των μηνυμάτων που ανταλλάσσονται από τα αντικείμενα που αντιστοιχούν στους ρόλους. Ένα τέτοιο σύνολο μηνυμάτων μιας συνεργασίας ονομάζεται αλληλεπίδραση (interaction).

Ένα μήνυμα είναι μία μονόδρομη επικοινωνία μεταξύ δύο αντικειμένων, μία ροή ελέγχου με πληροφορία από έναν αποστολέα προς έναν αποδέκτη. Ένα μήνυμα μπορεί να έχει παραμέτρους μεταφέροντας τιμές μεταξύ των αντικειμένων.

Η ακολουθία των μηνυμάτων παρουσιάζεται είτε με διαγράμματα ακολουθίας που εστιάζουν στην χρονική ακολουθία των μηνυμάτων είτε με διαγράμματα συνεργασίας που εστιάζουν στις σχέσεις μεταξύ των αντικειμένων που ανταλλάσσουν μηνύματα

9.8 Διάγραμμα Ακολουθίας

Ένα διάγραμμα ακολουθίας παρουσιάζει την αλληλεπίδραση μεταξύ αντικειμένων σε δύο διαστάσεις, όπου:

- η κάθετη διάσταση αντιστοιχεί στην κλίμακα του χρόνου
- η οριζόντια διάσταση στα ανεξάρτητα αντικείμενα.

Σε κάθε αντικείμενο αντιστοιχεί μία κάθετη γραμμή που ονομάζεται γραμμή ζωής (lifeline). Για όσο χρόνο ένα αντικείμενο υφίσταται, η γραμμή αυτή είναι διακεκομμένη ενώ για όσο χρόνο μία διαδικασία του εν λόγω αντικειμένου είναι ενεργή η γραμμή ζωής σχεδιάζεται ως μία διπλή γραμμή.

Ένα μήνυμα συμβολίζεται ως μία ακμή από τη γραμμή ζωής ενός αντικειμένου προς τη γραμμή ζωής ενός άλλου. Η θέση των ακμών αντιστοιχεί στην τοποθέτησή τους σε σχέση με τον χρόνο (μία ακμή χαμηλότερα από μία άλλη αντιστοιχεί σε μεταγενέστερο χρόνο).

Απαντήσεις σε μηνύματα υποδηλώνονται ως οριζόντιες διακεκομμένες ακμές

Τα διαγράμματα ακολουθίας είναι εξαιρετικά χρήσιμα για τον εντοπισμό λειτουργιών στις διάφορες κλάσεις. Αν ένα αντικείμενο λαμβάνει ένα μήνυμα *alpha*, συνεπάγεται ότι η κλάση στην οποία ανήκει το εν λόγω αντικείμενο θα πρέπει να έχει μία μέθοδο με το ίδιο όνομα για να εξυπηρετεί τον αποστολέα. Υπονοείται επίσης, ότι αφού δύο αντικείμενα επικοινωνούν μεταξύ τους μέσω μηνυμάτων, θα πρέπει να υπάρχει κάποιου είδους συσχέτιση μεταξύ τους που θα λειτουργεί ως διάυλος επικοινωνίας.

Για παράδειγμα, αν ένα αντικείμενο A στέλνει ένα μήνυμα ζητώντας κάποια λειτουργία από ένα αντικείμενο B, το A θα πρέπει με κάποιο τρόπο να γνωρίζει που να αποστείλει το μήνυμα. Αν αυτό συνεπάγεται την ύπαρξη μιας μονόδρομης συσχέτισης μεταξύ των αντίστοιχων κλάσεων, θα πρέπει κατά την υλοποίηση του συστήματος, στο συγκεκριμένο αντικείμενο A να γνωστοποιηθεί η θέση του αντικειμένου B

Κατά τη μελέτη σεναρίων (ανάπτυξη διαγραμμάτων ακολουθίας) είναι πιθανό να προκύψουν νέες κλάσεις, ιδιότητες και σχέσεις και συνεπώς να τροποποιηθεί το στατικό μοντέλο. Κάτι τέτοιο συμβαίνει συχνά και είναι επιθυμητό

Αν για παράδειγμα από τη μελέτη διαφόρων διαγραμμάτων ακολουθίας προκύπτει ότι δύο αντικείμενα αποκρίνονται με τον ίδιο τρόπο στα διάφορα δυνατά μηνύματα, τότε τα δύο αυτά αντικείμενα είτε θα ανήκουν στην ίδια κλάση είτε θα ανήκουν σε κλάσεις που συνδέονται με σχέση κληρονομικότητας.

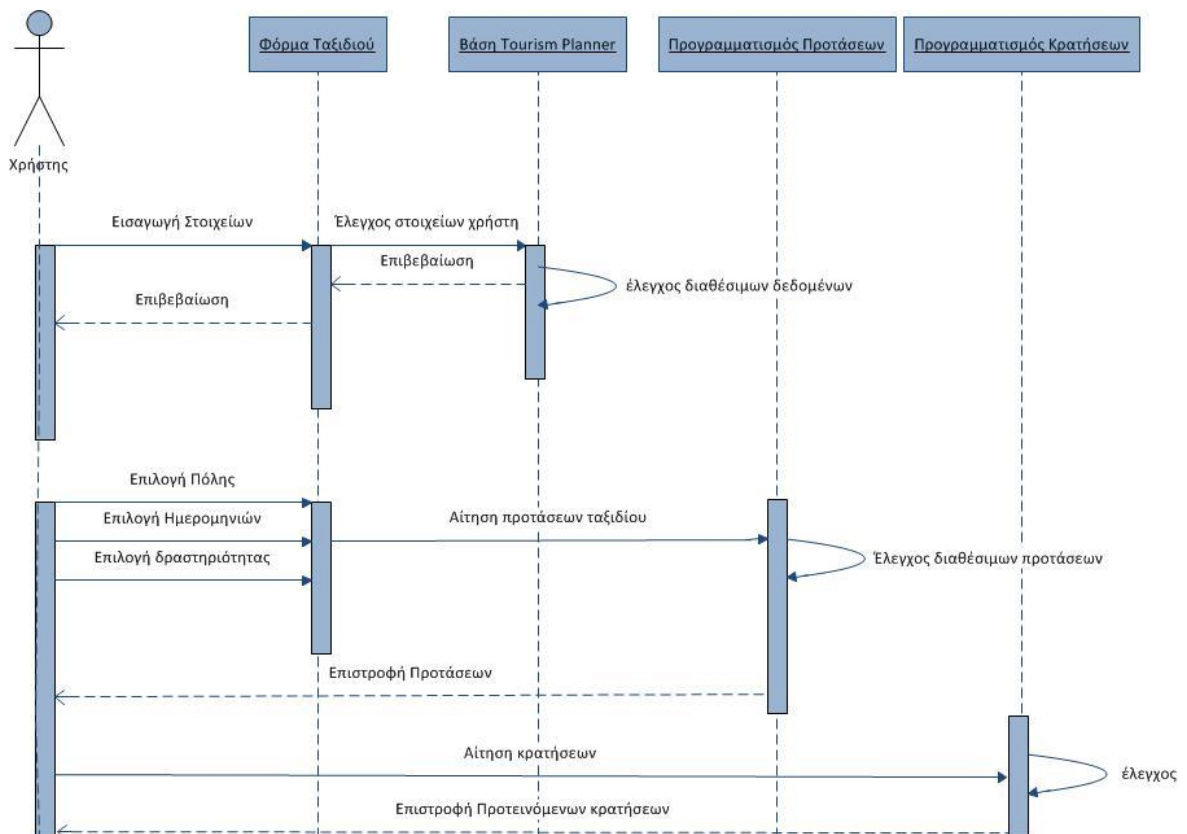
Υπό αυτή την έννοια, η επαναληπτική εξέταση διαγραμμάτων ακολουθίας και διαγραμμάτων κλάσεων οδηγεί στην ανάπτυξη του τελικού συστήματος, πολύ πριν αρχίσει η φάση της κωδικοποίησης. Θα πρέπει να γίνει κατανοητό, ότι αυτή η επαναληπτική διαδικασία στη φάση της ανάλυσης βρίσκεται στο επίκεντρο της αντικειμενοστραφούς μεθοδολογίας για την ανάπτυξη συστημάτων

Στο παρακάτω διάγραμμα μπορούμε να διαπιστώσουμε πως αλληλεπιδρούν τα διάφορα αντικείμενα και πως προκύπτει τελικά το επιθυμητό αποτέλεσμα με τις προτάσεις ταξιδιού στην δική μας εφαρμογή.

Ο χρήστης πραγματοποιεί την εισαγωγή στοιχείων και έχουμε την πρώτη αλληλεπίδραση με τη βάση όπου πραγματοποιείται ο έλεγχος των στοιχείων και γίνεται η επιβεβαίωση και στον οδηγό εφαρμογής και στο χρήστη.

Το άλλο σενάριο που ταυτίζετε με αυτό της εισαγωγής στοιχείων είναι αυτό της συμπλήρωσης των απαιτούμενων στοιχείων δηλαδή της πόλης, των ημερομηνιών και της δραστηριότητας όπου πραγματοποιείται το αίτημα γίνεται ο έλεγχος των διαθέσιμων προτάσεων και γίνεται επιστροφή και προβολή στο χρήστη.

Τέλος πραγματοποιείται και από το χρήστη η αίτηση για κάποιο είδος κράτησης όπως αεροπορικών εισιτηρίων, ξενοδοχείων, Παραστάσεων κτλ και επιστρέφει την επιβεβαίωση αφού ακολουθηθούν οι αναγκαίες συμπλήρωσης των απαιτούμενων στοιχείων από την εκάστοτε υπηρεσία.



Σχήμα 19 : Sequence Diagram

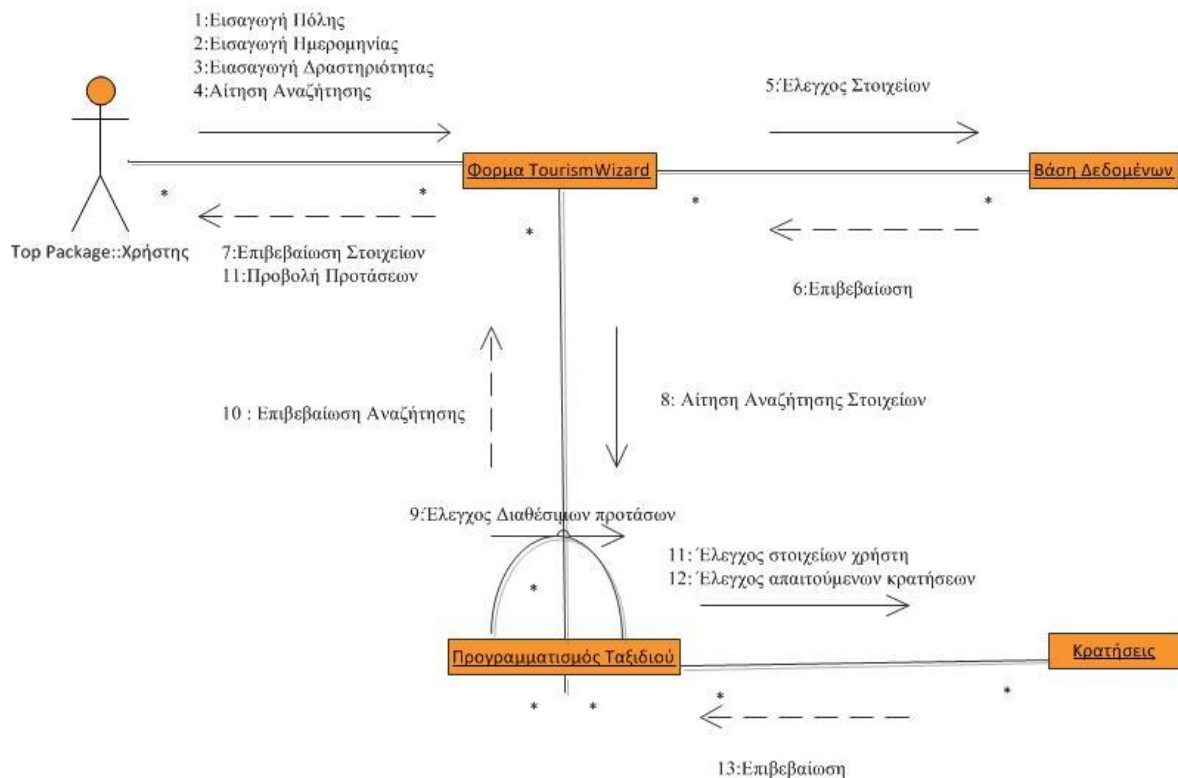
9.9 Διάγραμμα Συνεργασίας

Σε ένα διάγραμμα συνεργασίας τα αντικείμενα απεικονίζονται με τις γραμμές συσχετίσεων των κλάσεων τους να τα ενώνουν, δηλαδή απεικονίζονται οι στατικές συνδέσεις μεταξύ των αντικειμένων.

Ενώ τα διαγράμματα ακολουθίας απεικονίζουν κυρίως τη χρονική ροή των μηνυμάτων σε ένα σενάριο μιας περίπτωσης χρήσης, τα διαγράμματα συνεργασίας χρησιμοποιούνται για να παρουσιάσουν τις σχέσεις μεταξύ αντικειμένων.

Δεν υπάρχει συγκεκριμένη μορφή (τα αντικείμενα μπορούν να εμφανίζονται σε οποιοδήποτε σημείο του διαγράμματος) ενώ για να απεικονιστεί η ακολουθία των μηνυμάτων που ανταλλάσσονται χρησιμοποιείται αρίθμηση. Τα διαγράμματα ακολουθίας και συνεργασίας θεωρούνται συμπληρωματικά καθώς περιέχουν την ίδια πληροφορία αλλά κάθε ένα δίνει μια διαφορετική οπτική γωνία (σε πολλά εργαλεία το ένα είδος διαγράμματος παράγεται αυτόματα από το άλλο).

Το διάγραμμα συνεργασίας στη δική μας εφαρμογή είναι απλό και εύκολα τροποποιήσιμοι και επεκτάσιμο. Δίνονται αριθμητικά οι λειτουργίες που πραγματοποιεί ο χρήστης χωρίς αυτές να είναι δεσμευτικές αλλά που στο τέλος δίνουν τα επιθυμητά αποτελέσματα προτάσεων και κρατήσεων. Είναι λίγο πιο γενικό από το ακολουθίας και θα μπορούσαμε ίσως και να το παραβλέψουμε αλλά κρίθηκε σκόπιμο να δημιουργηθεί προκειμένου να επιτευχθεί και ο εκπαιδευτικός σκοπός αυτής της μεταπτυχιακής εργασίας και να δώσει στον αναγνώστη μια ιδέα ενός τέτοιου διαγράμματος.



Σχήμα 20 Collaboration Diagram

9.10 Ροή Ελέγχου

Τα διαγράμματα ακολουθίας/συνεργασίας δίνουν έμφαση στη ροή της πληροφορίας. Δεν είναι κατάλληλα για απεικόνιση της ροής του ελέγχου (control flow). Η ροή του ελέγχου παρουσιάζεται με τα διαγράμματα δραστηριότητας και καταστάσεων

Ένα διάγραμμα δραστηριότητας μοντελοποιεί τη ροή της εργασίας, αναπαριστώντας τις διάφορες καταστάσεις εκτέλεσης ενός υπολογισμού (Bohm & Jacorini). Παρουσιάζεται η ροή του ελέγχου μεταξύ δραστηριοτήτων του ίδιου αντικειμένου ή πολλών αντικειμένων

Μία κατάσταση δραστηριότητας συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες με περιγραφή της δραστηριότητας. Η μετάβαση κατά τη συμπλήρωση μιας δραστηριότητας συμβολίζεται ως ακμή. Οι διακλαδώσεις συμβολίζονται είτε με συνθήκες φρουρούς επί των μεταβάσεων είτε με κόμβους απόφασης (ρόμβους) με πολλαπλές εξερχόμενες ακμές.

Μία ένωση (join) συμβολίζει συνένωση πολλών εισερχομένων μεταβάσεων σε μία εξερχόμενη ενώ μία διχάλα (fork) την ανάλυση μιας εισερχόμενης σε πολλές παράλληλες ταυτόχρονες.

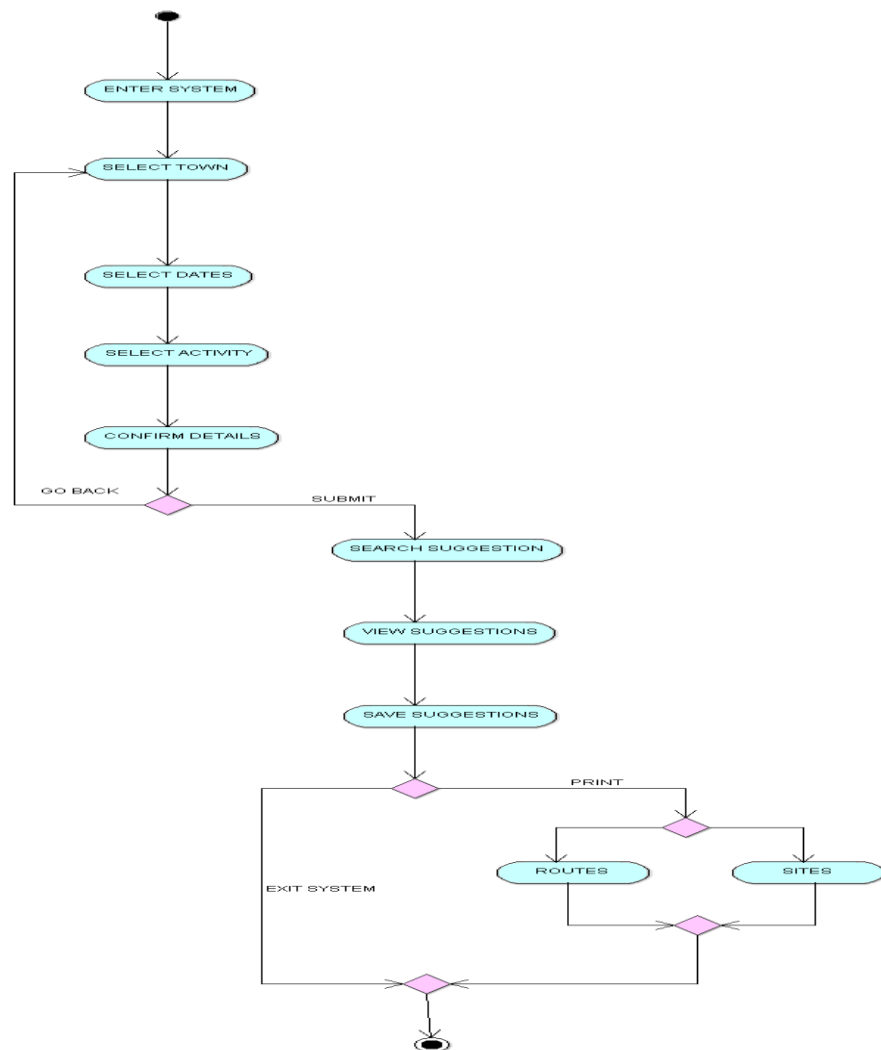
Στα παρακάτω σχήματα θα δούμε τη ροή εργασιών για την δική μας εφαρμογή. Στο σχήμα 9 θα δούμε την ροή εργασιών που περιγράψαμε και παραπάνω και αφορά την σειριακή επεξεργασία των εισόδων του χρήστη που θα πραγματοποιηθούν στο σύστημα. Στην με τα παρακάτω διαγράμματα έχουμε το ποιες δραστηριότητες θα εκτελεστούν σε κάθε σημείο. Στην εφαρμογή μας ξεχωρίζουμε δύο διαδοχικές δραστηριότητες, πρώτα την συμπλήρωση οδηγού και στην συνέχεια την κράτηση εισιτηρίων και εκτύπωση κρατήσεων και αποτελεσμάτων.

Στο πρώτο διάγραμμα ο χρήστης εισέρχεται στο σύστημα και ακολουθεί το ένα και μοναδικό μονοπάτι που του παρέχει η υπηρεσία και δεν είναι άλλο από την συμπλήρωση του οδηγού προκειμένου να εκτελεστούν τα κατάλληλα queries στη βάση. Στη συνέχεια ξεκινά τη συμπλήρωση επιλέγοντας τη πόλη, συνεχίζει τη διαδρομή του με επιλογή των ημερομηνιών και επιλέγει και την επιθυμητή δραστηριότητα.

Συναντάμε το πρώτο σταυροδρόμι που έχει την δυνατότητα να επιστρέψει πίσω στην αρχή του οδηγού ή να κάνει την επιβεβαίωση των στοιχείων και να προχωρήσει στο μονοπάτι. Στη συνέχεια της διαδρομής του θα συναντήσει την μπάρα αναμονής για την αναζήτηση των αποτελεσμάτων.

Με το πέρας αναζήτησης θα έχουμε την προβολή των αποτελεσμάτων και στο επόμενο στάδιο ο χρήστης θα μπορεί να σώσει τις επιλογές του για μελλοντική χρήση. Στην συνέχεια ο χρήστης έρχεται αντιμέτωπος με το επόμενο σταυροδρόμι που του παρέχονται οι επιλογές εξόδου από το σύστημα ή να συνεχίσει με τις εκτυπώσεις των προτάσεων.

Οι εκτυπώσεις για το χρήστη αποτελούν και το τελευταίο σταυροδρόμι και έχει τις επιλογές να εκτυπώσει τις διαδρομές ή τις τοποθεσίες. Τελειώνοντας τη διαδρομή οδηγείται πλέον οριστικά στην έξοδο από το σύστημα.

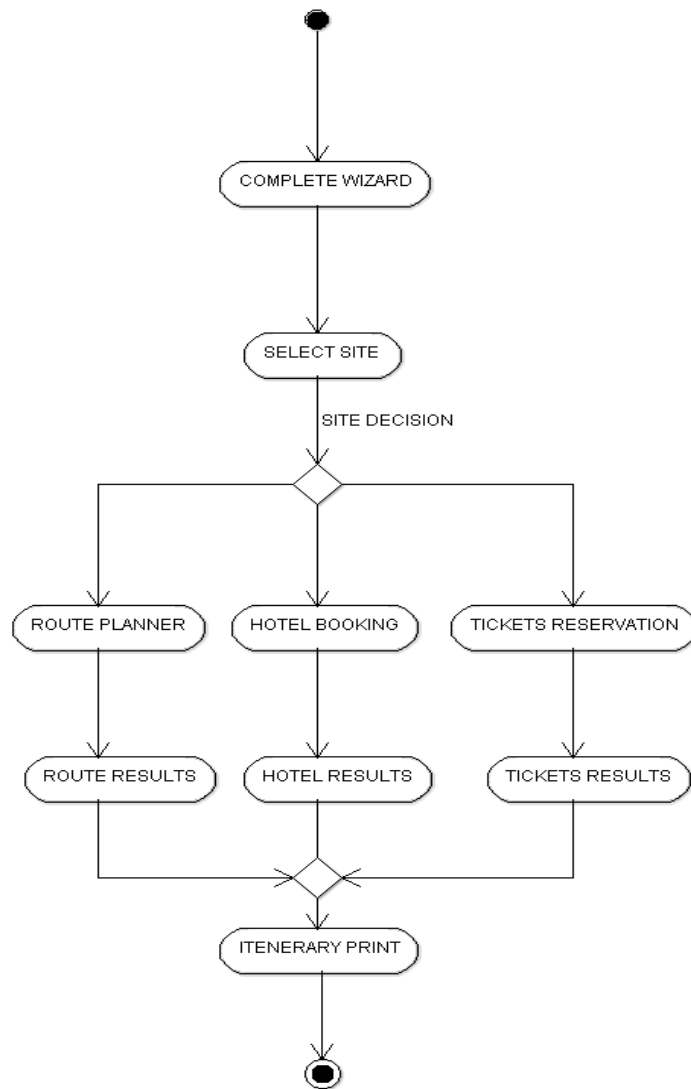


Σχήμα 21 Activity diagram για συμπλήρωση οδηγού TourismWizard

Στο επόμενο διάγραμμα εφόσον έχει ολοκληρωθεί η διαδικασία αναζήτησης και προβολής των διαθέσιμων προτάσεων ο χρήστης μπαίνει στην διαδρομή επιλογής της κατάλληλης ιστοσελίδας για τη πραγματοποίηση διαφόρων κρατήσεων.

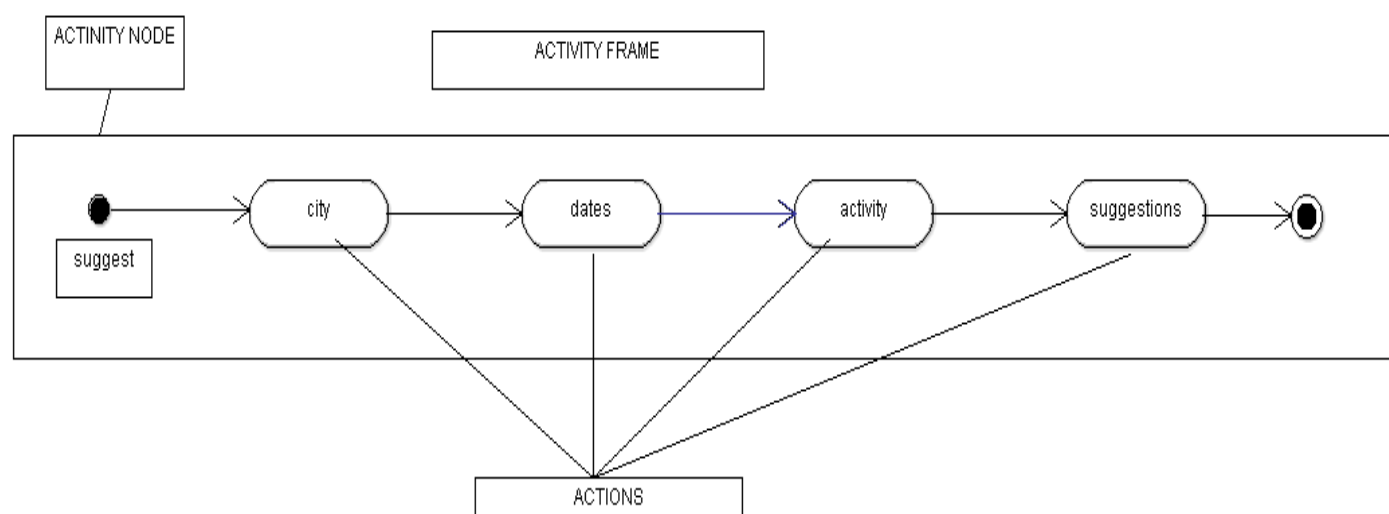
Ο κόμβος θα τον οδηγήσει είτε στην επιλογή κρατήσεων ξενοδοχείων ή στην επιλογή αεροπορικών κρατήσεων ή στην δυνατότητα επιλογής του σχεδιαστή ταξιδιού με μέσα μαζικής μεταφοράς που έχει πραγματοποιηθεί παλιότερα σαν μεταπτυχιακή εργασία και αποτελεί εξαιρετική υπηρεσία για ενσωμάτωση της στην δική μας εφαρμογή.

Τέλος αφού πραγματοποιήσει εκτύπωση του itinerary μπορεί να οδηγηθεί στην έξοδο της υπηρεσίας.



Σχήμα 22 Activity diagram για πραγματοποίηση κράτησης μετά την ολοκλήρωση του οδηγού

Εδώ βλέπουμε συνολικά τις δραστηριότητες του χρήστη



Σχήμα 23 Πλαίσιο δραστηριότητας

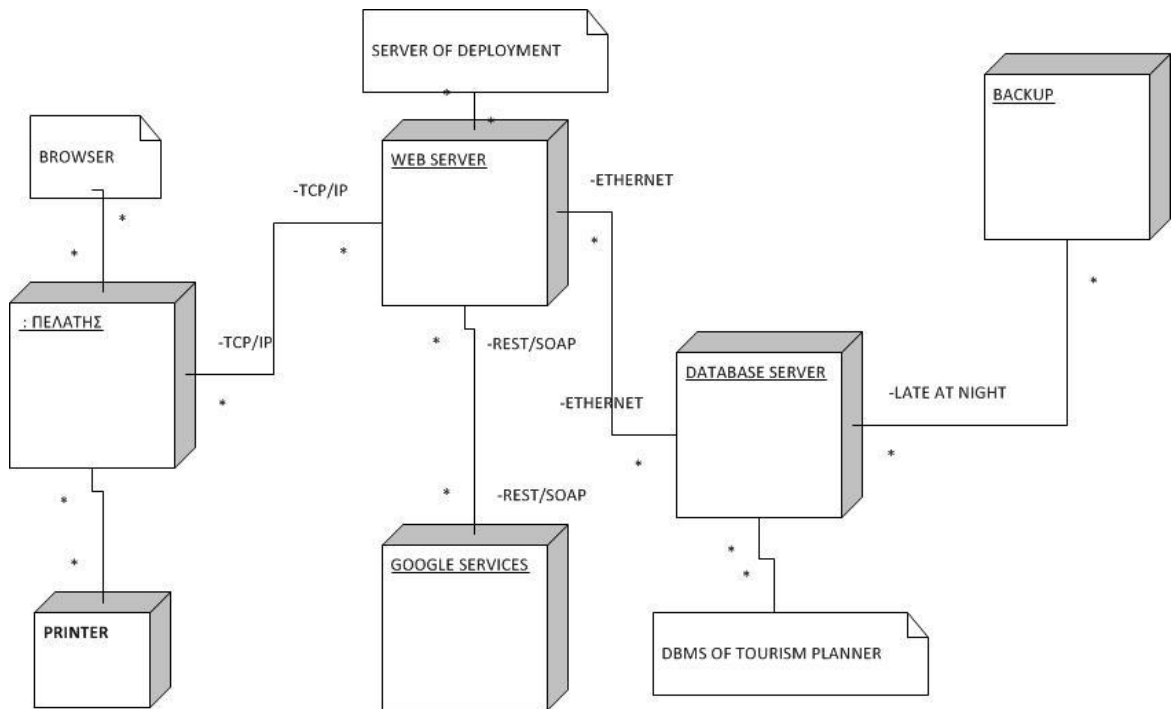
9.11 Διάγραμμα Ανάπτυξης

Τα διαγράμματα ανάπτυξης (deployment diagrams) αναπαριστούν την αντιστοίχιση του λογισμικού σε επεξεργαστικές μονάδες - κόμβους . Ένας κόμβους (node) είναι ένα φυσικό αντικείμενο που στη γενική περίπτωση έχει τουλάχιστον μνήμη και δυνατότητα επεξεργασίας.

Παρακάτω βλέπουμε το προς υλοποίηση πρόγραμμα μας και τα απαραίτητα στοιχεία του. Χρειαζόμαστε την πλευρά του πελάτη με ένα σύστημα με παροχή δικτύου για να μπορέσει να εισέλθει στην υπηρεσία και προαιρετικά έναν εκτυπωτή για τις κατάλληλες εκτυπώσεις /Μια παροχή σύνδεσης όπως αναφέρθηκε στο διαδίκτυο προκειμένου ο πελάτης με τη χρήση του πρωτοκόλλου TCP/IP να επικοινωνήσει με τον εξυπηρετητή που φιλοξενεί την υπηρεσία.

Ο εξυπηρετητής αυτός μέσω της εφαρμογής μας θα επικοινωνεί με τις υπηρεσίες τις Google προκειμένου να έχουμε τους χάρτες, τις διαδρομές και τις άλλες δυνατότητες που μας παρέχει η Google.Η σύνδεση θα επιτευχθεί με πρωτόκολλο REST ή SOAP όπως έχει αναφερθεί και παραπάνω.

Τέλος μέσω Ethernet θα γίνεται η επικοινωνία του εξυπηρετητή με τη βάση δεδομένων της εφαρμογής που θα έχει και αυτή αποσταλεί στο server που θα μας φιλοξενήσει. Αργά τη νύχτα θα πραγματοποιείται και το backup της βάσης.



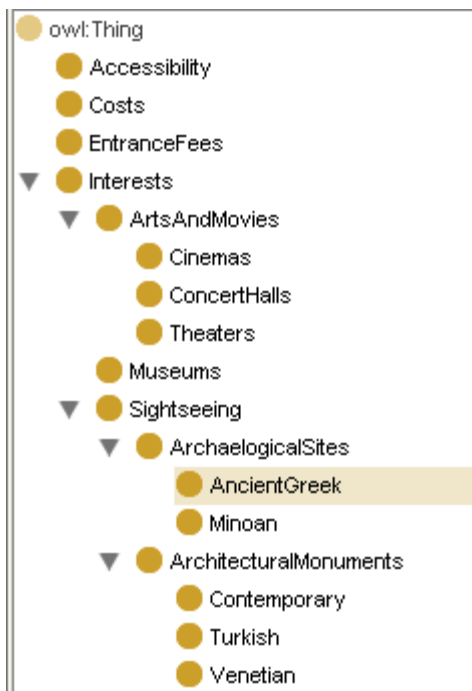
Σχήμα 24 Deployment diagram

9.12 Οντολογίες με χρήση protégé

Προκειμένου να έχουμε και μια πιο ολοκληρωμένη σχεδίαση δεν θα μπορούσε να λείπει και η δημιουργία οντολογιών με τη χρήση του προγράμματος protégé. Με την χρήση των οντολογιών δημιουργούμε κλάσεις που εύκολα μπορούν να χρησιμοποιηθούν στην εφαρμογή μας διευκολύνοντας την επέκταση του προγράμματος.

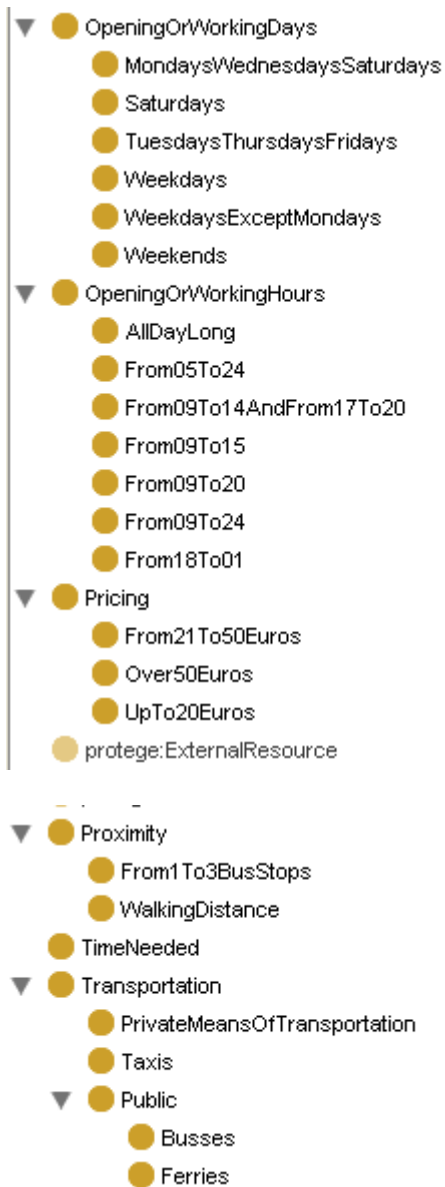
Οι οντολογίες αυτές περιλαμβάνουν μια ολοκληρωμένη εικόνα της εφαρμογής μας και ένα καλό στιγμιότυπο της βάσης μας. Περιλαμβάνουν πλήρη πληροφορίες για την κάθε δραστηριότητα και θα αποτελέσουν τη βάση για την μελλοντική αναβάθμιση του προγράμματος μας δεδομένου ότι στην παρούσα ανάλυση και σχεδίαση της βάσης δεν συμπεριλαμβάνονται δραστηριότητες φαγητού, εξόδου και αγορών λόγω του μεγάλου όγκου δεδομένων που πρέπει να περιέχουν.

Όμως η ανάλυση μας θεωρείται πλήρης με την ύπαρξη της παρούσας σχεδίασης, ενώ η κλάση σε γλώσσα java εύκολα δημιουργείται με τη χρήση του protégé. Παρακάτω αναλύονται οι κλάσεις και οι υποκλάσεις όπως σχεδιάστηκαν [Τομαη2005].



- ▼ ● Geomorphologies
 - Caves
 - Gorges
 - Plateaux
 - NaturalBeautyAreas
- ▼ ● ReligiousSites
 - Churches
 - Monasteries
- ▼ ● LeisureActivities
 - ▼ ● EatingOut
 - FastFoods
 - InternationalCuisines
 - LocalCuisineTaverns
 - Restaurants
 - ▼ ● Nightlife
 - Bars
 - Discos

- ▼ ● Shopping
 - Bookstores
 - FleaMarkets
 - GiftShops
 - MiscellaneousShops
- ▼ ● SportActivitiesAndFacilities
 - GolfCourts
 - MiscellaneousSportFacilities
- ▼ ● Swimming
 - Beaches
 - SwimmingPools
 - TennisCourts
- ▼ ● Location
 - GeocodingOrAddress
 - PointOnMap
 - TextDescription



Σχήμα 25 Ontologies diagram[Τομαη 2005]

10

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

10.1 Συμπεράσματα

Στην παρούσα εργασία ασχοληθήκαμε με την ανάλυση και σχεδίαση ενός λογισμικού, για τη δημιουργία μια εφαρμογής με τη νέα τεχνολογία των Java Server Faces και την χαρτογράφηση της βάσης με Java Persistence. Νέες τεχνολογίες, όπως αυτές των prime faces, θα ενσωματωθούν για να δημιουργήσουν ένα ελκυστικό περιβάλλον χρήσης, ενώ χρησιμοποιήθηκαν και οι οντολογίες προκειμένου να υπάρχει και μελλοντική εύκολη επέκταση της υπηρεσίας, διότι οι οντολογίες δεν ακολουθούν μια συγκεκριμένη τεχνολογία αλλά μπορούν εύκολα να χρησιμοποιηθούν και να ενσωματωθούν σε άλλες γλώσσες.

Ακολουθήθηκε η αντικειμενοστραφής σχεδίαση προκειμένου να μπορούν εύκολα να πραγματοποιούνται αλλαγές χωρίς να αναγκαζόμαστε να σχεδιάζουμε από την αρχή το σύστημα, ενώ παράλληλα να έχουμε μια ολοκληρωμένη εικόνα του συστήματος μας. Η αντικειμενοστραφής αυτή σχεδίαση είναι ανεξάρτητη γλώσσας προγραμματισμού και έτσι εύκολα μπορεί οποιοσδήποτε προγραμματιστής να την χρησιμοποιήσει για την ανάπτυξη παρόμοιου συστήματος. Δημιουργήθηκε μια αρχική βάση δεδομένων, που σε καμιά περίπτωση δεν είναι και η τελική της μορφή διότι κατά την πορεία ανάπτυξης και υλοποίησης του συστήματος θα ενσωματωθούν και άλλες οντότητες όπως προτάσεις για διασκέδαση, φαγητό και ότι άλλο κριθεί απαραίτητο για την ολοκλήρωση της υπηρεσίας.

Η όλη εφαρμογή θα υλοποιηθεί σαν υπηρεσία και όχι σαν απλή ιστοσελίδα για να μπορεί να χρησιμοποιηθεί ευρέως και από άλλες εφαρμογές, ενώ η επεκτασιμότητα και διαδραστικότητα θα αποτελούν τα κύρια χαρακτηριστικά της. Όπως θα διαπιστώθηκε και στην ανάλυση το όλο σύστημα έχει διατηρηθεί απλό και αυτό γιατί πρέπει να είναι εύκολη η υλοποίηση του και απλή η χρήση του. Ας μην ξεχνάμε ότι η υπηρεσία θα διατίθεται δωρεάν με αποτέλεσμα δεν θέλουμε ένα περίπλοκο σύστημα που θα χρειαστεί περισσότερα χρήματα από αυτά που θα αποδώσει.

Η τεχνολογία που χρησιμοποιείται είναι πολύ πρόσφατη και λόγω προηγούμενης εμπειρίας στην ανάπτυξη άλλων διαδικτυακών εφαρμογών με τη χρήση των JSF, είναι σίγουρο ότι θα αντιμετωπίσουμε αρκετά προβλήματα που με υπομονή και θέληση θα ξεπεραστούν. Ήδη έχει δημιουργηθεί μια μικρή κλάση ονομαζόμενη ως `bugremover` που αντιμετωπίζει κάποια ανεπιθύμητα μηνύματα κατά τη διάρκεια εκτέλεσης της εφαρμογής, αφαιρώντας τα τελείως. Αυτή η κλάση μπορεί να ενσωματωθεί σε οποιαδήποτε εφαρμογή εύκολα και χωρίς την ανάγκη εισαγωγής επιπλέον κώδικα.

Συμπερασματικά λοιπόν θα λέγαμε ότι με την δίκη μας εφαρμογή προσπαθούμε να καλύψουμε το κενό των υπάρχοντων εφαρμογών σε ότι αφορά την παροχή πληροφοριών. Πιο συγκεκριμένα ενώ όλες οι ελληνικές εφαρμογές απλά παρέχουν γενικές πληροφορίες για τις τοποθεσίες και συνολικά τα αξιοθέατα που υπάρχουν ενώ στην δικιά μας εφαρμογή δεν ασχολείται με την απλή παρουσίαση των αξιοθέατων κι των αρχαιολογικών χώρων αλλά με την οργάνωση ενός ταξιδιού και την οργάνωση των επισκέψεων με πρόταση που να πάει πως να πάει ώστε να κερδίζει χρόνο με την επίσκεψη κοντινών τοποθεσιών πρώτα σε αυτόν και φυσικά με γνώση του κόστους και του ωραρίου λειτουργίας.

Οποιοδήποτε ελληνικό site θα μπορεί να ενσωματώσει την υπηρεσία στην ιστοσελίδα του ενώ με τη βοήθεια του ελληνικού οργανισμού τουρισμού θα μπορούσαμε να δημιουργήσουμε μια ολοκληρωμένη βάση δεδομένων και κατ'επέκταση μια ολοκληρωμένη υπηρεσία με παροχή προτάσεων για όλα τα μέρη της Ελλάδος δίνοντας στον επισκέπτη τη δυνατότητα απλά και γρήγορα να βρίσκει και να επισκέπτεται όλα τα αξιοθέατα της κάθε περιοχής προάγοντας έτσι τον ελληνικό τουρισμό αλλά και την ελληνική τεχνολογία.

Έτσι όλοι οι τοπικοί φορείς θα μπορούσαν να έχουν μια ολοκληρωμένη υπηρεσία παροχής τουριστικών πληροφοριών ενώ ταυτόχρονα θα μπορούσαν εύκολα να διαφημιστούν και να προτείνονται ξενοδοχεία καταστήματα εμπορικά εστιατόρια και άλλα πολλά επιτυγχάνοντας ταυτόχρονα έτσι την εμπορική ανάπτυξη της περιοχής αυξάνοντας τα έσοδα των εκαστοτε περιοχών αφού ο τουρίστας θα κατευθύνεται μέσα από την εφαρμογή που να πάει.

Η υπηρεσία αυτή αποτελεί ανάπτυξη λογισμικού μεγάλης κλίμακας και για αυτό το λόγο δεν μπορεί να πραγματοποιηθεί η υλοποίηση της από ένα μόνο άτομο. Η δημιουργία της βάσης ο προγραμματισμός των κλάσεων η σχεδίαση της ιστοσελίδας η ανάλυση του συστήματος και το οπτικό αποτέλεσμα προκειμένου να ελκύει τον επισκέπτη αποτελεί έργο πολλών ατόμων και γι

αυτό το λόγο η βοήθεια οποιοδήποτε επιθυμεί να εμπλακεί στην υλοποίηση αυτή είναι ευπρόσδεκτη.

Στη φάση αυτή έχει ολοκληρωθεί η ανάλυση της υπηρεσίας και ο σχεδιασμός. Η τεχνολογία που θα χρησιμοποιηθεί έχει επιλεγεί και πάνω σε αυτή δημιουργήθηκε το demo του οδηγού συμπλήρωσης που ο κώδικας του βέβαια έχει υλοποιηθεί. Η βάση έχει αναπτυχτεί και υλοποιηθεί και αναμένεται συμπλήρωση και επέκταση της με λίγη βοήθεια και όρεξη σύντομα θα μπορέσουμε να έχουμε τα αποτελέσματα από μια πρώτη υλοποίηση της υπηρεσίας με αρχικά την παροχή πληροφοριών για την Αθήνα ενώ με την συμμετοχή και των αρμόδιων φορέων άλλων περιοχών θα μπορέσει η υπηρεσία να αναπτυχτεί ραγδαία και να επεκταθεί για όλη την Ελλάδα

10.2 Ανάπτυξη Αλγορίθμου

Οι αλγόριθμοι γενικότερα δεν είναι συνήθως κατασκευασμένοι ώστε να δίνουν την τέλεια λύση, αλλά μία ικανοποιητική λύση στις περισσότερες των περιπτώσεων, και μέσα σε σύντομο χρόνο, κάνοντας όσο το δυνατόν εξοικονόμηση των διαθέσιμων πόρων. Λειτουργούν δηλαδή πάνω στον συμβιβασμό της τελειότητας τού αποτελέσματος και τού περιορισμού των χρησιμοποιούμενων, προσπαθώντας να εκμεταλλευτούν τις ιδιαιτερότητες που παρουσιάζονται.

Η παραπάνω ανάλυση του συστήματος περιλαμβάνει όπως και οι περισσότερες εφαρμογές την εκτέλεση απλών υποερωτημάτων για την αναζήτηση αποτελεσμάτων, όπως αυτά διαμορφώνονται με την εισαγωγή σε πεδία κειμένου των λέξεων κλειδιών. Επιπλέον η χρήση και της πειραματικής εφαρμογής Google Places, όπου περιλαμβάνει κατάλληλους αλγόριθμους αναζήτησης με την εισαγωγή λέξεων κλειδιών από το χρήστη, για το είδος που επιθυμεί, όπως για παράδειγμα "εστιατόρια" και εμφανίζει τα εστιατόρια της περιοχής που έχει επιλέξει ο χρήστης σίγουρα βοηθά στην υλοποίηση μιας πιο ολοκληρωμένης εφαρμογής.

Ένας περιορισμός που υπάρχει είναι ότι η αναζήτηση βασίζεται στη βάση δεδομένων της Goggle, που μπορεί βέβαια να είναι αρκετά ενημερωμένη αλλά ίσως να μην περιέχει όλα τα δεδομένα που εμείς επιθυμούμε. Γι'αυτό η Google στην πειραματική αυτή εφαρμογή δίνει την δυνατότητα στους προγραμματιστές, της αναβάθμισης της βάσης τους, με εισαγωγή νέων δεδομένων προκειμένου να ικανοποιηθούν όλες οι απαιτήσεις των χρηστών αλλά και να είναι διαθέσιμα τα δεδομένα και για άλλους προγραμματιστές.

Όμως όπως είπαμε το καλύτερο είναι μια εφαρμογή να είναι αυτόνομη, χωρίς να χρειάζεται να πληρώνουμε για υπηρεσίες και για δεδομένα. Κρίνεται σκόπιμο λοιπόν η δημιουργία μιας ολοκληρωμένης βάσης με όλα τα απαραίτητα δεδομένα και η δημιουργία αλγορίθμου που να μπορεί να συνδυάσει την αναζήτηση δεδομένων από τη βάση μας, από ελεύθερες πηγές του διαδικτύου και τη βάση της Google με χρήση της υπηρεσίας της, όσο διατίθεται δωρεάν.

Ο αλγόριθμος αυτός προκειμένου να υλοποιηθεί χρειάζεται αρκετή γνώση της θεωρίας αλγορίθμων, ενώ απαιτεί αρκετό χρόνο για να μπορεί να αναπτυχθεί, να δοκιμαστεί και να ενσωματωθεί σε υπηρεσίες, δίνοντας σίγουρα αποτελέσματα. Η δημιουργία αυτή του αλγορίθμου θα μας απασχολήσει στο προσεχές μέλλον δεδομένου ότι η δημιουργία του περιβάλλοντος χρήσης δεν αποτελεί ιδιαίτερο πρόβλημα προγραμματιστικά.

10.3 Μελλοντικές επεκτάσεις

Η εφαρμογή αυτή όπως αναφέρθηκε στην αρχή θα περιλαμβάνει κυρίως τα αξιοθέατα. Στα πλαίσια όμως μια μελλοντικής επέκτασης του συστήματος και εφόσον υπάρχουν διαθέσιμες οι κατάλληλες πληροφορίες, θα είναι δυνατόν στην υπηρεσία να ενσωματωθούν τοποθεσίες όπως εστιατόρια, προτάσεις εξόδου, γεωμορφολογικές τοποθεσίες όπως σπηλιές, λίμνες που σαν χώρα διαθέτουμε αρκετές, ενώ δύναται να περιλαμβάνονται και δραστηριότητες όπως πισίνες, παραλίες και αθλητικές εγκαταστάσεις.

Επιπλέον για να ολοκληρωθεί η υπηρεσία πολύ σημαντικό θα ήταν και η ενσωμάτωση όπως αναφέρθηκε και στη σχεδίαση, υπηρεσιών κρατήσεων ξενοδοχείων, αεροπορικών εισιτηρίων, εισιτηρίων θεάτρου και κινηματογράφου και άλλων πολιτιστικών εκδηλώσεων ενώ θα μπορούσαν να ενσωματωθούν και υπηρεσίες χρυσού οδηγού και η πρόσφατη υπηρεσία taxi που δημιουργήθηκε.

Επιπρόσθετα ο σχεδιαστής ταξιδιού με μέσα μαζικής μεταφοράς, όπου υπάρχει η δυνατότητα σχεδίασης μιας διαδρομής με χρήση των μεταφορικών μέσων, κοστολόγηση της διαδρομής με τις υπάρχουσες τιμές των εισιτηρίων, διάρκεια διαδρομής και αναλυτική περιγραφή της διαδρομής με τις στάσεις και αλλαγές που πρέπει να πραγματοποιηθούν, θα έδινε την δυνατότητα δημιουργίας της πιο ολοκληρωμένης υπηρεσίας ταξιδιού και το πιο σημαντικό όλα σε μία εφαρμογή.

Τέλος θα μπορούσε να ενσωματωθεί και υπηρεσία σύγκρισης τιμών των επιλογών του χρήστη ώστε να του δίνεται και η δυνατότητα της επιλογής και εναλλακτικών προτάσεων αλλά και αναζήτησης άλλων παρόμοιων με αυτών της επιλογής του, οπότε στην ουσία μιλάμε για αναβάθμιση της υπηρεσίας μας σε ένα recommendation system.

ΑΝΑΦΟΡΕΣ

- [01] Wikipedia. Διάφορα Άρθρα, η πρόσβαση στα οποία μπορεί να γίνει με λέξεις αναζήτησης. <http://www.wikipedia.org>
- [02] Java Computer Programming Language. Η γλώσσα προγραμματισμού Java. <http://www.java.com>
- [03] NetBeans. Περιβάλλον ανάπτυξης εφαρμογών. <http://www.netbeans.org>
- [04] Subversion. Σύστημα διαχείρισης εκδόσεων λογισμικού. <http://subversion.apache.org>
- [05] Apache Maven. Εργαλείο Διαχείρισης Projects λογισμικού. <http://maven.apache.org>
- [06] Glassfish server. Server που θα χρησιμοποιηθεί από την υπηρεσία μας. <http://www.glassfish.com>
- [07] Duffel site. Προτεινομενη ιστοσελίδα προς μελέτη <http://www.duffel.com>
- [08] Nileguide Site. Προτεινομενη ιστοσελίδα προς μελέτη www.nileguide.com
- [09] Triporama Site. Προτεινομενη ιστοσελίδα προς μελέτη www.triporama.com
- [10] Tripit Site. Προτεινομενη ιστοσελίδα προς μελέτη www.tripit.com
- [11] WanderFly Site. Προτεινομενη ιστοσελίδα προς μελέτη www.wanderfly.com

- [12] Plnnr Site.Προτεινομενη ιστοσελίδα προς μελέτη
www.plnnr.com
- [13] Core Java Server Faces 3Rd Edition David Geary Prentige Hall 2011
- [14] Primefaces users guide.3.2 Çağatay Çivici 2011
- [15] Javaee tutorial 6 Oracle 2012
- [16] Java How to Program, 9th Edition – Deitel 2012
- [17] Web Services Essentials O'Reilly - 2002
- [18] Java Persistence Tutorial Oracle 2012
- [19] MySQL Workbrench user guide
- [20] UML 3rd Edition Oreily
- [21] SQL Καδης 2003
- [22] Advanced JPA2 for advanced Programming
- [23] TheDefinitive Guide of NetBeans Platform7. Apress Heiko Bock 2011
- [24] Oreilly.JavaScript.and.jQuery.The.Missing.Manual.2nd.Edition. David Sayer 2011
- [25] Μιχάλης Τζιώτης –Δημιουργία ενός σχεδιαστή ταξιδιού με MMM για τους επισκέπτες της Αθήνας 2010 Μεταπτυχιακή εργασία τμήμα Γεωπληροφορικής
- [26] Οργάνωση Εξατομικευμένων διαδικτυακών Υπηρεσιών Ταξιδιού μέσω Οντολογιών – Λειτουργικές προδιαγραφές σχεδιασμού συστήματος- Μεταπτυχιακή εργασία τμήμα Γεωπληροφορικής–Μαρίνα Ψυχογιού 2010

- [27] Protégé, 2010 Protégé Overview, Copyright © 2010 Stanford Center for Biomedical Informatics Research, <http://protege.stanford.edu/overview/protege-owl.html>
- [28] Τομαή, 2005 Τομαή Ελένη (2005), 'Ανάπτυξη Τυπικών Οντολογιών Στη Γεωπληροφορική Για Την Διαλειτουργικότητα Βάσεων Γεωγραφικής Γνώσης', Εθνικό Μετσόβιο Πολυτεχνεί
- [29] Tomai et al, 2006 Tomai Eleni, Michael Stavros and Prastacos Poulicos, 'An Ontology-based Web-Portal for Tourism', 2nd International Workshop on Web Portal-based Solutions for Tourism and other business areas, Münster, Germany, 20 September 2006
- [30] W3C, 2004 World Wide Web Consortium (W3C) Recommendation, OWL Web Ontology Language Overview, 10 February 2004, <http://www.w3.org/TR/owl-features>
- [31] Internet Gis. Wiley, 2003
- [32] Web Services in a nutshell. Topley, 2002
- [33] Google Documentation για τα διάφορα API που θα χρησιμοποιηθούν
- [34] Data structures in Java Michael Goodrich 5th edition 2011
- [35] O'Reilly - HTML and XHTML The Definitive Guide 6th Edition
- [36] Σημειώσεις καθηγητών από τη Σχολή Προγραμματιστών και Αναλυτών του Υπουργείου Εθνικής Άμυνας
- [37] Σημειώσεις Ανάπτυξη Διαδικτυακών Τόπων - Χατζηευθυμιάδης Στ.
- [38] OOAD_With_Applications Grady Booch Wisley 1998
- [39] W3C schools Ajax tutorial <http://www.w3schools.com/ajax/default.asp>
- [40] Αντικειμενοστραφής Σχεδίαση Αλ.Χατζηγεωργίου Κλειδάριθμος 2005

ΚΑΤΑΛΟΓΟΣ ΣΤΟΙΧΕΙΩΝ

Σχήμα 1	Διαδικασία Ανάπτυξης	σελ 12
Σχήμα 2.	Διάφοροι τύποι INTERNET GIS	σελ 37
Σχήμα 3	Αρχιτεκτονική Multitier [Πηγή Java Programming Deitel 2011]	σελ 53
Σχήμα 4:	Αναπαράσταση Υπηρεσίας	σελ 94
Σχήμα 5:	Επικοινωνιακό σχήμα	σελ 97
Σχήμα 6:	Βάση δεδομένων με UML αναπαράσταση	σελ 99
Σχήμα 7	Λειτουργίες χρήστη	σελ 103
Σχήμα 8	Φάσεις ανάπτυξης εφαρμογής	σελ 111
Σχήμα 9	Άξονες μοντελοποίησης	σελ 112
Σχήμα 10	Use Case Diagram	σελ 115
Σχήμα11	Πρώτη περίπτωση χρήσης	σελ 116
Σχήμα 12	2 ^η περίπτωση χρήσης	σελ 118
Σχήμα13	Entities Class diagram	σελ 121
Σχήμα 14	Class diagram	σελ 122
Σχήμα 15	JSF Class diagram	σελ 123
Σχήμα 16	JSF Class diagram	σελ 124
Σχήμα17	Υλοποίηση κλάσεων JSF	σελ 125
Σχήμα 18	Λειτουργία και αρχιτεκτονική κλάσεων	σελ 126
Σχήμα 19	Sequence Diagram	σελ 130
Σχήμα 20	Collaboration Diagram	σελ 131
Σχήμα 21	Activity diagram για συμπλήρωση οδηγού TourismWizard	σελ 134
Σχήμα 22	Activity diagram για πραγματοποίηση κράτησης	σελ 136
Σχήμα 23	Πλάισιο δραστηριότητας	σελ 137
Σχήμα 24	Deployment diagram	σελ 138
Σχήμα 25	Ontologies diagram[Τομαη 2005]	σελ 141

Εικόνα 1	Αποτελέσματα εφαρμογής [Πηγή:www.visitgreece.gr]	σελ 14
Εικόνα 2	Αποτελέσματα εφαρμογής [Πηγή:www.visitgreece.gr]	σελ 14
Εικόνα 3	Αποτελέσματα εφαρμογής [Πηγή:www.visitgreece.gr]	σελ 15
Εικόνα 4	Πρώτο βήμα οδηγού	σελ 17
Εικόνα 5	Λανθασμένη εισαγωγή	σελ 17
Εικόνα 6	Πρώτο βήμα οδηγού	σελ 18
Εικόνα 7	Δεύτερο βήμα οδηγού	σελ 18
Εικόνα 8	Εισαγωγή ημερομηνιών	σελ 19
Εικόνα 9	Τρίτο βήμα οδηγού	σελ 19
Εικόνα 10	Τέταρτο βήμα οδηγού	σελ 20
Εικόνα 11	Εμφάνιση Αποτελεσμάτων	σελ 20
Εικόνα 12	Περιβάλλον Εφαρμογής Duffel	σελ 25
Εικόνα 13	Αποτελέσματα εφαρμογής [Πηγή:www.duffel.com]	σελ 26
Εικόνα 14	Περιβάλλον χρήστη του Nileguide [Πηγή:www.nileguide.com]	σελ 27
Εικόνα 15	Triporama [Πηγή:www.triporama.com]	σελ 28
Εικόνα 16	TripIt [Πηγή:www.tripit.com]	σελ 29
Εικόνα 17	WanderFly application [Πηγή:www.wanderfly.com]	σελ 29
Εικόνα 18	Έναρξη εφαρμογής Plnnr [Πηγή: www.plnnr.com]	σελ 31
Εικόνα 19	Δυνατότητες εφαρμογής. [Πηγή: www.plnnr.com]	σελ 32
Εικόνα 20	Πρώτο βήμα:Επιλογή μέρους [Πηγή: www.plnnr.com]	σελ 33
Εικόνα 21	Δεύτερο Βήμα :Επιλογή ημερομηνιών [Πηγή: www.plnnr.com]	σελ 33
Εικόνα 22	Τρίτο Βήμα: Επιλογή Δραστηριότητας [Πηγή: www.plnnr.com]	σελ 34
Εικόνα 23	Τέταρτο βήμα: Επιλογή δραστηριότητας [Πηγή: www.plnnr.com]	σελ 35
Εικόνα 24	Πέμπτο Βήμα: Επιλογή πολυτέλειας	σελ 36
Εικόνα 25	Εμφάνιση αποτελεσμάτων Plnnr [Πηγή: www.plnnr.com]	σελ 36
Εικόνα 26	GIS εξέλιξη [Πηγή: Internet Gis. Wiley, 2003]	σελ 38
Εικόνα 27	Autonomous distributed and interoperable GIS Service 2003] [Πηγή: Internet Gis. Wiley,	σελ 40
Εικόνα 28	Λειτουργία Web Service [Πηγή:Internet Gis. Wiley, 2003]	σελ 41
Εικόνα 29	SOA Architecture [Πηγή: Web Services in a nutshell. Topley, 2002]	σελ 43
Εικόνα 30	Πρωτόκολλα Web Services [Web Services in nutshell, Topley 2002]	σελ 44
Εικόνα 31	Σύνδεσης client-server [Πηγή Oreilly Web Services in a nutshell 2002]	σελ 45
Εικόνα 32	Λειτουργία SOAP [Πηγή Oreilly Web Services in a nutshell 2002]	σελ 46
Εικόνα 33	Λειτουργία WSDL [Πηγή Oreilly Web Services in a nutshell 2002]	σελ 47
Εικόνα 34	Λειτουργία UDDI [Πηγή Oreilly Web Services in a nutshell 2002]	σελ 48
Εικόνα 35	Multithier model[Πηγή Oracle Java Tutorial 2011]	σελ 56

Εικόνα 36	MVC Model [Πηγή Java Programming Deitel 2011]	σελ 59
Εικόνα 37	JavaServer Faces εφαρμογή[Πηγή Java EE Tutorial 2012]	σελ 61
Εικόνα 38	Web Components[Πηγή Java EE Tutorial 2012]	σελ 64
Εικόνα 39	Assembly root[Πηγή Java EE Tutorial 2012]	σελ 66