ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάλυση, Σχεδιασμός και Υλοποίηση Συμβουλευτικού Συστήματος στον Τουρισμό

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος Κοντογεώργος

Επιβλέπων : Ιωάννης Βασιλείου, Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάλυση, Σχεδιασμός και Υλοποίηση Συμβουλευτικού Συστήματος στον Τουρισμό

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος Κοντογεώργος

Επιβλέπων : Ιωάννης Βασιλείου, Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25η Σεπτεμβρίου 2014.

.....	.....	.....
Ιωάννης Βασιλείου	Γιώργος Στάμου	Κώστας Κοντογιάννης
Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2014

.....

Αλέξανδρος Κοντογεώργος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη Σκοπός της διπλωματικής είναι η δημιουργία ενός συστήματος που θα βελτιστοποιεί τις ταξιδιωτικές επιλογές των χρηστών του. Αυτή η βελτιστοποίηση προκύπτει τόσο από τα εργαλεία του συστήματος για αναζήτηση και σύγκριση όσο και από το Σύστημα Συστάσεων που υλοποιείται για αυτοματοποιημένη σύσταση ταξιδιωτικών πακέτων στους τελικούς χρήστες. Το διαδικτυκό μας σύστημα δίνει την δυνατότητα σε ταξιδιωτικά γραφεία να ανεβάζουν ταξιδιωτικά πακέτα, και τους χρήστες να αναζητούν, προβάλλουν, συγκρίνουν, αγοράζουν και να αξιολογούν ταξιδιωτικά πακέτα. Το Σύστημα Συστάσεων αναλύει τα δεδομένα αξιολόγησης των χρηστών και προτείνει πακέτα τα οποία δεν έχουν επιλέξει και είναι κοντά στις προτιμήσεις τους βάσει προηγούμενων επιλογών τους. Το διαδικτυακό σύστημα υλοποιήθηκε με PHP, MySQL, Apache HTTP Server, Apache Tomcat Server, Java, και Javascript.

Λέξεις-κλειδιά: Σύστημα Συστάσεων, διαδικτυακό σύστημα, αρχιτεκτονική τριών στρωμάτων, βιομηχανία τουρισμού, PHP, Java, Javascript, Apache, MySQL

Abstract The purpose of this thesis is the development of a system that optimizes the travel choices of its users. This optimization derives from the system tools that are used for search and comparison purposes as well as from the Recommender System that is developed for automated recommendation of travel packages towards its final users. Our web system enables travel agencies to upload travel packages and users to search, display, compare, buy and rate travel packages. Recommender System analyzes the user ratings data and recommends travel packages that are not yet bought from the users and are close to their interests according to their previous choices. The web system was developed with the use of PHP, MySQL, Apache HTTP Server, Apache Tomcat Server, Java and Javascript technologies.

Keywords Recommender System, web system, 3 – tier architecture, tourism industry, PHP, Java, Javascript, Apache, MySQL

Ευχαριστίες,

*στην οικογένειά μου για την υποστήριξή της στο έργο μου
για την ολοκλήρωση των ακαδημαϊκών μου στόχων,*

*στον Τερροβίτη Μανώλη για τις συμβουλές του και την καθοδήγησή
του στην διάρκεια της διπλωματικής μου εργασίας,*

*στον καθηγητή Βασιλείου Ιωάννη για την υποστήριξή του και την
καθοδήγησή του τόσο ως καθηγητής στο προπτυχιακό μου στάδιο όσο
και ως επιβλέπων στην διπλωματική μου εργασία.*

Περιεχόμενα

Κατάλογος Σχημάτων

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Στο κεφάλαιο αυτό γίνεται περιγραφή του αντικειμένου της διπλωματικής εργασίας και συνοπτική παρουσίαση της οργάνωσής της στα διάφορα κεφάλαια. Στην συνέχεια γίνεται η ανάλυση του προβλήματος και της προσέγγισης που επιλέξαμε, για την καλύτερη κατανόηση της δομής και της λειτουργικότητας του υλοποιημένου συστήματος.

1.1 Αντικείμενο της διπλωματικής

Αντικείμενο της διπλωματικής εργασίας είναι η δημιουργία ενός διαδικτυακού συστήματος που παρέχει εργαλεία για την βελτιστοποίηση των τελικών ταξιδιωτικών επιλογών ενός χρήστη. Πιο συγκεκριμένα, ο ταξιδιωτικός χώρος αφορά πολλές οντότητες σε ένα πραγματικό σενάριο. Απ'τις μεταφορές και την διαμονή ως τα δρομολόγια και τις δραστηριότητες σε έναν πιθανό προορισμό. Στην παρούσα εργασία θεωρούμε ως αντικείμενα χειρισμού απ'το σύστημα το ταξιδιωτικό πακέτο, τον ταξιδιώτη και το ταξιδιωτικό γραφείο. Τα διαδικτυακά εργαλεία που παρέχονται είναι ένα εργαλείο σύγκρισης ταξιδιωτικών πακέτων ως προς τα χαρακτηριστικά τους (τιμή, διάρκεια, κτλ) και ένα σύστημα προσωποποιημένων συστάσεων ως προς τους χρήστες. Η προσωποποιημένη αυτή σύσταση γίνεται βάσει ανάλυσης τόσο της συμπεριφοράς των -προς σύσταση- χρηστών όσο και της συμπεριφοράς των υπολοίπων χρηστών στα ταξιδιωτικά πακέτα. Οι χρήστες του συστήματος μπορούν να εξερευνήσουν τις διάφορες ταξιδιωτικές προτάσεις, να τις συγκρίνουν με άλλες πιθανές επιλογές τους, να αγοράσουν κάποιο ταξιδιωτικό πακέτο, να το αξιολογήσουν και τέλος να λάβουν νέες συστάσεις απ'το σύστημα. Τα ταξιδιωτικά γραφεία έχουν την δυνατότητα ενημέρωσης του συστήματος με νέες ταξιδιωτικές προτάσεις. Το σύστημα υλοποιήθηκε με την χρήση μόνο ελεύθερου λογισμικού ανοιχτού κώδικα. Πιο συγκεκριμένα χρησιμοποιήθηκαν οι τεχνολογίες PHP ως γλώσσα προγραμματισμού, Apache HTTP Server ως εξυπηρετητή ιστού, MySQL ως σύστημα διαχείρισης βάσης δεδομένων, Apache Tomcat Server ως Servlet Container, και την βιβλιοθήκη υλοποίησης αλγορίθμων μηχανικής μάθησης Apache Mahout η οποία είναι γραμμένη σε Java. Το σύστημα των συστάσεων αποτελεί κομμάτι του συνολικού διαδικτυακού συστήματος αλλά μπορεί να χρησιμοποιηθεί και ως Web Service.

1.2 Ανάλυση του προβλήματος και λειτουργικότητα

Κυρίαρχο ρόλο για την κατανόηση της δομής του συστήματος είναι η ανάλυση των οντοτήτων που συμμετέχουν σε αυτό, σύμφωνα με την δική μας προσέγγιση. Όπως αναφέρθηκε και παραπάνω, ο χώρος του τουρισμού στον πραγματικό κόσμο είναι εξαιρετικά πολύπλοκος, εμπλέκοντας πολλές οντότητες για την παροχή υπηρεσιών τόσο ως προς τους χρήστες όσο ως προς τις επιχειρήσεις που κινούνται στην τουριστική βιομηχανία. Γι'αυτό τον λόγο μπορεί κάποιος να διαπιστώσει την ύπαρξη διαδικτυακών συστημάτων που επιλέγουν

συγκεκριμένη προσέγγιση του χώρου ώστε να παρέχουν στους χρήστες τους ποιοτικές υπηρεσίες. Έτσι, θα μπορούσαμε να εντοπίσουμε ιστότοπους που παρέχουν δομημένη πληροφόρηση για διάφορους προορισμούς, συστήματα πώλησης ταξιδιωτικών πακέτων, συστήματα ενοικίασης χώρων διαμονής και πολλά άλλα. Η δική μας προσέγγιση έχει να κάνει με την επεξεργασία αγορών από χρήστες και με συστάσεις ταξιδιωτικών πακέτων. Η προσέγγιση αυτή αφορά τρείς οντότητες της τουριστικής βιομηχανίας στις οποίες και αποδίδουμε συγκεκριμένα χαρακτηριστικά, παρακάτω. Αυτές είναι:

1. Ταξιδιωτικά γραφεία
2. Ταξιδιωτικά πακέτα
3. Ταξιδιώτες / Χρήστες του συστήματος

1.2.1 Ταξιδιωτικά γραφεία

Τα ταξιδιωτικά γραφεία έχουν παθητικό ρόλο ως προς το σύστημα συστάσεων που πρόκειται να κατασκευαστεί. Ο παθητικός τους ρόλος ορίζεται απ'το γεγονός ότι στην ουσία γεμίζουν το σύστημα με ταξιδιωτικά πακέτα και αξιολογούνται από τους χρήστες ως προς τις υπηρεσίες που προσέφεραν χωρίς να συμμετέχουν ενεργά στην βελτιστοποίηση των συστάσεων που γίνονται στους τελικούς χρήστες του συστήματος. Πιο συγκεκριμένα, ένα πακέτο αξιολογείται για τους προορισμούς του και για το ταξιδιωτικό γραφείο που το προσέφερε. Η αξιολόγηση όμως του ταξιδιωτικού γραφείου απ'τους χρήστες δεν επιφέρει κάποια αλλαγή ή βελτιστοποίηση στις συστάσεις προς τους χρήστες, αλλά συμμετέχει στην τελική βαθμολογία του ταξιδιωτικού γραφείου για την μέχρι τώρα δραστηριότητά του στο σύστημα. Άρα, η οντότητα “ταξιδιωτικό γραφείο” χρησιμοποιείται για την ολοκληρωμένη μοντελοποίηση του ταξιδιωτικού χώρου σύμφωνα με την προσέγγισή μας.

Η οντότητα ταξιδιωτικό γραφείο συνδέεται με την οντότητα ταξιδιωτικά πακέτα αποδίδοντάς τους δύο βασικά χαρακτηριστικά. Αυτά είναι το diversity και το locality. Το χαρακτηριστικό “diversity” αναφέρεται στην πολυμορφία των ταξιδιωτικών πακέτων με την έννοια ότι ίδιοι προορισμοί προσφέρονται με διαφορετικό τρόπο από το σύνολο των ταξιδιωτικών γραφείων που υπάρχουν στο σύστημα. Είναι και χαρακτηριστικό του πραγματικού κόσμου όπου ένας προορισμός μπορεί να προσφερθεί με πολλούς διαφορετικούς τρόπους ως προς την τιμή του, την διαμονή που προσφέρει, την διάρκεια του κτλ. Επομένως, δεν αρκεί η αξιολόγηση μόνο ενός προορισμού αλλά και η αξιολόγηση της ταξιδιωτικής πρότασης που σχετίζεται με αυτόν προορισμό και άρα του ταξιδιωτικού γραφείου για αυτή την ταξιδιωτική του πρόταση. Το χαρακτηριστικό “locality” έχει να κάνει με την ιδιότητα των ταξιδιωτικών γραφείων να έχουν την ικανότητα να εξυπηρετούν τους χρήστες ως προς την τοποθεσία τους ή την περιοχή εξυπηρέτησής τους. Όπως και αν ορίζεται η περιοχή της δράσης τους, ένα ταξιδιωτικό γραφείο δεν μπορεί να εξυπηρετεί ταξιδιώτες απ'όλο τον κόσμο. Έτσι, ένας προορισμός μπορεί να εμπεριέχεται σε διάφορες ταξιδιωτικές προτάσεις οι οποίες να ενδιαφέρουν μερικώς ή και καθόλου ένα χρήστη. Με αυτή τη λογική, στο σύστημά μας έχει γίνει μέριμνα τόσο για την αυτοματοποιημένη σύσταση, όσο και για την εύρεση σε επίπεδο φίλτρων αναζήτησης έτσι ώστε τα πακέτα που εξετάζει ο χρήστης να έχουν νόημα για αυτόν.

Συνοψίζοντας, διαπιστώνουμε τόσο τον σημαντικό ρόλο των ταξιδιωτικών γραφείων στην μοντελοποίηση του ταξιδιωτικού χώρου σύμφωνα με την προσέγγισή μας όσο και τον παθητικό τους ρόλο ως προς την σύσταση των πακέτων στους χρήστες.

1.2.2 Ταξιδιωτικά πακέτα

Τα ταξιδιωτικά πακέτα αποτελούν την βασική οντότητα του συστήματος καθώς απ'την μια συγκεντρώνουν όλη την απαραίτητη πληροφορία για την σύνθεση ενός ταξιδιωτικού πακέτου που προσεγγίζει τον πραγματικό κόσμο (προορισμοί, τιμή, διάρκεια κτλ) και απ'την άλλη τα χαρακτηριστικά τους υπόκεινται σε αξιολόγηση έτσι ώστε να επιλεχθούν για σύσταση ή αγορά από τον χρήστη. Ένα ταξιδιωτικό πακέτο υπόκειται σε αξιολόγηση με δύο τρόπους : Είτε με την χρήση του εργαλείου σύγκρισης σε σχέση με άλλα πακέτα όπως προσφέρεται από την υλοποίηση του συστήματος είτε με την αυτοματοποιημένη ανάλυση του συστήματος συστάσεων που υλοποιήθηκε βάσει των προτιμήσεων των χρηστών.

Στην υλοποίηση του εργαλείου σύγκρισης δεν κάνουμε κάποια ιδιαίτερη προσέγγιση στην οντότητα αυτή. Γίνεται σύγκριση των στοιχείων κατά απόλυτη τιμή όπως θα συνέβαινε σε οποιοδήποτε πραγματική οντότητα ταξιδιωτικού πακέτου. Τιμή, απόσταση, διάρκεια και μέσος όρος βαθμολόγησης ταξιδιωτικού γραφείου που προτείνει το συγκρεκριμένο πακέτο είναι μερικά απ’τα συγκρίσιμα μεγέθη που χρησιμοποιούνται στην εκάστοτε σύγκριση. Αντιθέτως, στην αυτοματοποιημένη ανάλυση των ταξιδιωτικών πακέτων χρησιμοποιούμε διαφορετική προσέγγιση. Θεωρούμε ότι κάθε ταξιδιωτικό πακέτο, κατά την δημιουργία του απ'το ταξιδιωτικό γραφείο, αποτελεί διαφορετικό αντικείμενο κάθε φορά. Πιο συγκεκριμένα, με το πέρας της χρονικής περιόδου στην οποία αναφέρεται ένα οποιοδήποτε πακέτο, παύει να υφίσταται στο σύστημα. Επομένως, ακόμα και αν το ίδιο ταξιδιωτικό γραφείο εισάγει στο σύστημα ένα πακέτο με τα ίδια χαρακτηριστικά το οποίο όμως αναφέρεται σε άλλο χρονικό διάστημα, τότε αυτό το πακέτο θα αποτελεί εντελώς διαφορετικό αντικείμενο σε σχέση με το προηγούμενο. Άρα, η αξιολόγηση ενός ταξιδιωτικού πακέτου ως σύνολο δεν έχει κάποιο νόημα. Έτσι, στην προσέγγισή μας για την οντότητα του ταξιδιωτικού πακέτου, θεωρούμε ότι αυτό που αξιολογείται είναι οι προορισμοί οι οποίοι συνθέτουν αυτή την ταξιδιωτική πρόταση και όχι το ίδιο το αντικείμενο του πακέτου. Ένα νέο αντικείμενο, λοιπόν, αποκτά αμέσως νόημα ως προς την αξιολόγησή του ως ένα σύνολο προορισμών για το οποίο θα προταθεί ή όχι στον χρήστη, ανάλογα με την έξοδο του αλγορίθμου που επιλέξαμε και κατασκευάσαμε για αυτή την αυτοματοποιημένη διαδικασία.

1.2.3 Ταξιδιώτες / Χρήστες του συστήματος

Οι χρήστες του συστήματος, ή αλλιώς οι ταξιδιώτες που χρησιμοποιούν το σύστημα για την επιλογή των ταξιδιωτικών τους προορισμών, εκμεταλλεύονται τις λειτουργικές δυνατότητες του συστήματος όπως είναι η χρήση των φίλτρων αναζήτησης, η χρήση του εργαλείου σύγκρισης που αναφέρθηκε παραπάνω και η δυνατότητα να επιλέξουν, αγοράσουν και να αξιολογήσουν τους προορισμούς και το ταξιδιωτικό γραφείο που συνθέτουν το εκάστοτε πακέτο. Γι'αυτό τον λόγο αποτελούν και την ενεργή οντότητα του συστήματος. Με την αξιολόγηση των πακέτων οι χρήστες αλλάζουν την συμπεριφορά και την έξοδο του συστήματος συστάσεων και έτσι, οι συστάσεις που δέχονται γίνονται -ιδανικά- όλο και πιο ακριβείς ως προς τις πραγματικές μελλοντικές τους προτιμήσεις. Ακόμα και τα ταξιδιωτικά γραφεία τα οποία δέχονται τις αξιολογήσεις των χρηστών μπορούν να καθορίσουν την στρατηγική τους ως προς την σύνθεση πακέτων με συγκεκριμένα χαρακτηριστικά για τα οποία η ανταπόκριση των χρηστών είναι θετικότερη. Άρα γίνεται ξεκάθαρη η δυναμική σχέση του χρήστη/ταξιδιώτη με το υπόλοιπο σύστημα, η οποία καθορίζει τον χαρακτήρα της οντότητας του χρήστη ως ενεργού.

Ένα σημαντικό στοιχείο στην προσέγγιση που κάναμε για την υλοποίηση του

συστήματος είναι ότι δεν μας ενδιαφέρουν τα δημογραφικά χαρακτηριστικά των χρηστών ως προς την αυτοματοποιημένη σύσταση που υλοποιούμε. Θα δούμε και παρακάτω ότι η ανάλυση των δεδομένων του συστήματος αφορά μόνο το κομμάτι της αξιολόγησης των χρηστών ως προς τα αντικείμενα και αγνοεί τα δημογραφικά τους χαρακτηριστικά. Και άρα στην υλοποίησή μας η καταγραφή των δημογραφικών τους χαρακτηριστικών δεν είναι εξαντλητική. Παρ'όλα αυτά, τα αποτελέσματα των συστάσεων δεν επηρεάζονται σημαντικά και δίνουμε έμφαση στην βελτιστοποίησή τους βάσει αυτής μας της προσέγγισης.

1.3 Οργάνωση του τόμου

Η εργασία, έχει οργανωθεί στα εξής κεφάλαια:

Στο Κεφάλαιο 2, περιγράφουμε τις υπάρχουσες τεχνολογίες με τις οποίες ασχοληθήκαμε για την ανάπτυξη του συστήματός μας.

Στο Κεφάλαιο 3, περιγράφουμε το σύστημα και τις λειτουργίες του όπως αυτές εμφανίζονται στον χρήστη.

Στο Κεφάλαιο 4, περιγράφουμε την υλοποίηση του διαδικτυακού μας συστήματος τόσο σε επίπεδο αρχιτεκτονικής όσο και σε επίπεδο υλοποίησης των επί μέρους υποσυστημάτων του.

Στο Κεφάλαιο 5, αναλύουμε τα συμπεράσματα που προέκυψαν από την ενασχόληση με την συγκεκριμένη εργασία.

ΚΕΦΑΛΑΙΟ 2

Υπάρχουσες τεχνολογίες

Σε αυτό το κεφάλαιο γίνεται αναφορά στις έννοιες και τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος.

2.1 Διαδίκτυο (Internet)

Το Διαδίκτυο (Internet) είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών το οποίο χρησιμοποιεί πολυάριθμα τεχνολογικά πρωτόκολλα (τυποποιημένοι κανόνες επικοινωνίας) τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού. Η διασύνδεση αυτή των υπολογιστών γίνεται με τη χρήση των πρωτοκόλλων TCP/IP τα οποία έχουν πάρει το όνομα τους απ'τα δύο βασικότερα πρωτόκολλα που όρισαν το συγκεκριμένο μοντελο, το TCP και το IP. Φυσικά πέρα απ'το TCP και το IP, μπορούμε να αναφέρουμε μερικά απ'τα πιο γνωστά διαδικτύακα πρωτόκολλα τα οποία συμβάλλουν στην υλοποίηση και την λειτουργικά μερικών απ'των πιο γνωστών διαδικτυακών υπηρεσιών που χρησιμοποιούμε ευρέως μέχρι και σήμερα. Αυτά είναι το HTTP, UDP, DNS, IMAP, SMTP, SSH, Telnet και μπορούμε να τα συναντήσουμε στην χρήση των γνωστών -σε όλους τους χρήστες του διαδικτύου- διαδικτυακών υπηρεσιών όπως το ηλεκτρονικό ταχυδρομείο (email), οι ομάδες συζητήσεων (newsgroups), ο Παγκόσμιος Ιστος (World Wide Web), διαδικτυακή τηλεφωνία (VoIP), ανταλλαγή αρχείων (file sharing), κ.ά.

2.2 Υπερκείμενο (Hypertext)

Το υπερκείμενο είναι πληροφορία που παρουσιάζεται στην οθόνη του υπολογιστή μας -και σε άλλες ηλεκτρονικές συσκευές- σε μορφή κειμένου (text) το οποίο εμπεριέχει αναφορές σε άλλα κείμενα στα οποία ο χρήστης μπορεί να έχει πρόσβαση άμεσα ή μέσω σταδιακής παρουσίασης των λεπτομερειών του. Οι αναφορές αυτές σε ένα υπερκείμενο είναι γνωστές ως υπερσύνδεσμοι (hyperlinks). Φυσικά τα hypertexts μπορεί να έχουν διάφορες μορφές παρουσίασης της πληροφορίας όπως πίνακες, εικόνες, video κτλ. Τα υπερκείμενα αποτελούν την υποκείμενη δομή του Παγκόσμιου Ιστού, αποτελώντας ένα δίκτυο διασυνδεδεμένων κόμβων. Οι σελίδες των υπερκειμένων κατασκευάζονται με την HTML (Hypertext Markup Language).

2.3 Παγκόσμιος Ιστός (World Wide Web)

Ο Παγκόσμιος Ιστός (World Wide Web), είναι ένα σύστημα διασυνδεδεμένων εγγράφων υπερκειμένου τα οποία είναι προσβάσιμα μέσω του διαδικτύου. Αυτά τα έγγραφα είναι γνωστά ως ιστοσελίδες (web pages). Οι ιστοσελίδες είναι προσβάσιμες μέσω προγραμμάτων πλοήγησης (web browsers) και μπορεί να περιέχουν κείμενο, εικόνες, βίντεο και άλλα πολυμέσα. Η πλοήγηση μεταξύ των ιστοσελίδων γίνεται μέσω των υπερσυνδέσμων οι οποίοι είναι αναφορές σε δεδομένα είτε σε συγκεκριμένα σημεία μέσα στο ίδιο το έγγραφο υπερκειμένου είτε σε κάποιο νέο έγγραφο. Ένα σύνολο από ιστοσελίδες οι οποίες ανήκουν στον ίδιο χώρο του Παγκόσμιου Ιστού (domain), ονομάζεται δικτυακός τόπος (website). Αποσαφηνίζουμε την σχέση του του Παγκόσμιου Ιστού απ'το Διαδίκτυο διότι είναι σημαντικό να διαχωρίζουμε την υλική βάση απ'τις υπηρεσίες των συγκεκριμένων τεχνολογιών. Ακόμα και σήμερα, αρκετά χρόνια μετά την ίδρυση του Διαδικτύου ως τεχνολογία, η χρήση των όρων αυτών δεν είναι σαφώς διαχωρισμένη. Συνοψίζοντας λοιπόν, το Διαδίκτυο είναι τα διασυνδεδεμένα υπολογιστικά συστήματα στα οποία είναι αποθηκευμένα τα σύνολα των εγγράφων που αποτελούν τα υπερκείμενα ενώ ο Παγκόσμιος Ιστός αποτελεί τα διασυνδεδεμένα έγγραφα μέσω των υπερσυνδέσμων. Άρα γίνεται ξεκάθαρο και αυτό που αναφέραμε προηγουμένως ότι ο Παγκόσμιος Ιστός αποτελεί μια υπηρεσία του Διαδικτύου ανάμεσα σε πολλές άλλες, όπως είναι το ηλεκτρονικό ταχυδρομείο (email).

2.4 Ενιαίος Εντοπιστής Πόρων (URL)

Ενιαίος Εντοπιστής Πόρων (Uniform Resource Locator – URL) αποτελεί μια συμβολοσειρά που συνδέει μια αναφορά – είτε υπερσύνδεσμος είτε ρίζα μιας ιεραρχίας υπερσυνδέσμων- με τον Πόρο στο σύνολο του Παγκόσμιου Ιστού. Ένα παράδειγμα URL ή διεύθυνση μιας ιστοσελίδας είναι το http://www.example.com/Main_Page.html . Απ'την διεύθυνση, παίρνουμε πολλά στοιχεία για τον ιστοχώρο που επισκεπτόμαστε, όπως είναι το πρωτόκολλο επικοινωνίας που χρησιμοποιείται για πρόσβαση στο συγκεκριμένο μηχάνημα (http), το domain στο οποίο έχουμε πρόσβαση (example) και το υπερκείμενο στο οποίο πλοηγούμαστε (Main_Page.html).

2.5 Πρωτόκολλο Μεταφοράς Υπερκειμένου (HTTP)

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου – ΠΜΥ (Hypertext Transfer Protocol-HTTP) είναι ένα πρωτόκολλο επικοινωνίας δεδομένων του στρώματος εφαρμογής των δικτύων επικοινωνίας για κατανεμνημένα, συνεργατικά πληροφοριακά συστήματα υπερμέσων και αποτελεί την βάση της επικοινωνίας των δεδομένων στον Παγκόσμιο Ιστό. Το HTTP λειτουργεί ως πρωτόκολλο αιτήσεων – απαντήσεων (request – response) , στο υπολογιστικό μοντέλο πελάτη – εξυπηρετητή (client – server) . Ένα πρόγραμμα πλοήγησης για παράδειγμα είναι ο πελάτης και το πρόγραμμα που τρέχει στον υπολογιστή που φιλοξενεί την ιστοσελίδα είναι ο εξυπηρετητής. Ο πελάτης υποβάλει μια αίτηση σε μορφή μηνύματος στον εξυπηρετητή και ο εξυπηρετητής επιστρέφει μια απάντηση στον πελάτη παρέχοντας πόρους όπως αρχεία HTML και άλλου τύπου αρχεία ή εκτελώντας κάποιες ενέργειες στον εξυπηρετητή εκ' μέρους του πελάτη. Κάτι που θα είχε ιδιαίτερη αξία να αναφερθεί είναι ότι το HTTP είναι ένα ασυνεχές (stateless) πρωτόκολλο. Ένα ασυνεχές πρωτόκολλο είναι ένα πρωτόκολλο

επικοινωνίας που χειρίζεται κάθε αίτημα πελάτη σαν ένα ανεξάρτητο transaction το οποίο δεν σχετίζεται με κανένα προηγούμενο αίτημα έτσι ώστε η επικοινωνία να αποτελείται από ανεξάρτητα ζεύγη αιτημάτων – απαντήσεων (request – response pairs). Ένα ασυνεχές πρωτόκολλο δεν απαιτεί απ'τον εξυπηρετητή να κρατάει πληροφορίες για την συνεδρία (session) ή για την κατάσταση (status) για κάθε ζευγάρη πελάτη-εξυπηρετητή καθ'όλη την διάρκεια πολλαπλών αιτήσεων. Αντιθέτως, τα πρωτόκολλα αυτά ονομάζονται συνεχή (statefull protocol). Παρ'όλα αυτά μπορεί να επιτευχθεί χρήση αυτού του πρωτοκόλλου με χαρακτηρηστικά συνέχειας με σκοπό την αποδοτική υλοποίηση ενός διαδικτυακού συστήματος και την βελτίωση του user experience κατά την χρήση του συστήματος αυτού. Τέτοιες χρήσεις μπορούν να επιτευχθούν με την χρήση προσωρινών αρχείων κειμένου είτε στην πλευρά του πελάτη που ονομάζονται HTTP cookies είτε στην πλευρά του εξυπηρετητή που ονομάζονται HTTP Sessions και τα οποία διατηρούν πληροφορίες κατά τη διάρκεια διαδοχικών αιτημάτων του πελάτη. Στην υλοποίησή μας κάνουμε χρήση των HTTP Sessions και χρησιμοποιούμε την HTTP/1.1 όπως αυτή ορίζεται στο IETF RFC 2616.

2.6 Hypertext Markup Language (HTML)

Η HTML (Γλώσσα Σήμανσης Υπερκειμένων) είναι η βασική γλώσσα σήμανσης (Markup) για την δημιουργία ιστοσελίδων. Γράφεται στην μορφή στοιχείων της html που ονομάζονται ετικέτες που περικλείονται από αγκύλες με γωνία (tags) ορίζοντας μια ιεραρχική δομή δένδρου σε κάθε έγγραφο υπερκειμένου. Στόχος της HTML είναι να είναι επεξεργάσιμη απ'τα προγράμματα πλοήγησης τα οποία μεταφράζουν την γλώσσα που συνοδεύει το περιεχόμενο, τις αναφορές και τους πόρους κάθε εγγράφου υπερκειμένου σε ένα οπτικά οργανωμένο έγγραφο στην οθόνη του χρήστη αποκρύπτοντας οποιοδήποτε στοιχείο της html από αυτόν. Απ'την περιγραφή γίνεται αντιληπτός ο χαρκτήρας σήμανσης της συγκεκριμένης γλώσσας καθώς η HTML περιγράφει την δομή μιας ιστοσελίδας σημασιολογικά και γι'αυτό τον λόγο δεν θεωρείται γλώσσα προγραμματισμού.

2.7 Cascading Style Sheets (CSS)

Η CSS (Cascading Style Sheets - Διαδοχικά Φύλλα Στύλ) είναι μια γλώσσα που χρησιμοποιείται για την εμφάνιση και την μορφοποίηση ενός εγγράφου που είναι γραμμένο σε μια γλώσσα σήμανσης (όπως η HTML). Ο βασικός λόγος χρήσης της είναι ο διαχωρισμός του περιεχομένου ενός εγγράφου απ'τον τρόπο παρουσίασής τους βελτιώνοντας το user experience και υλοποιόντας ολοκληρωμένες διεπαφές χρήστη (user intefaces) για ιστοσελίδες και εφαρμογές διαδικτύου (web applications). Στην υλοποίησή μας χρησιμοποιούμε ένα αρχείο για την επίτευξη του στόχου αυτού χρησιμοποιόντας όλες τις δυνατότητες που μας παρέχει η συγκεκριμένη γλώσσα.

2.8 Javascript Language

Η γλώσσα προγραμματισμού Javascript είναι μια δυναμική γλώσσα προγραμματισμού.

Είναι συνήθως κομμάτι των περισσότερων προγραμμάτων πλοήγησης (web browsers), των οποίων η υλοποίηση επιτρέπει client-side scripts για αλληλεπίδραση με τον χρήστη, έλεγχο του browser, ασύγχρονη επικοινωνία και επεξεργασία δεδομένων, και τροποποίηση της δομής του υπερκειμένου και του περιεχομένου που προβάλλεται. Χρησιμοποιείται και για server-side προγραμματισμό με frameworks όπως το Node.js, για δημιουργία παιχνιδιών και για εφαρμογές υπολογιστών ή κινητών. Στην παρούσα υλοποίηση την χρησιμοποιούμε ως ένα χρήσιμο εργαλείο για την βελτίωση του user experience με όλες τις δυνατότητες που μας δίνει για αυτό τον σκοπό, περιορίζοντας την χρήση της στην πλευρά του χρήστη.

2.9 Java Language

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού ή οποία έχει ως βασικό χαρακτηριστικό σχεδίασης όσο το δυνατόν λιγότερους περιορισμούς εφαρμογής στις διάφορες πλατφόρμες. Πρόθεση, λοιπόν, της συγκεκριμένης γλώσσας είναι το να επιτρέψει στους application developers να φτιάχνουν εφαρμογές των οποίων ο κώδικας μόλις μεταγλωττιστεί ώστε να τρέχει σε μια πλατφόρμα δεν χρειάζεται να επαναμεταγλωττιστεί ώστε να τρέχει σε κάποια άλλη πλατφόρμα. Αυτό επιτυγχάνεται με την χρήση του JVM (Java Virtual Machine) το οποίο είναι ανεξάρτητο της αρχιτεκτονικής του υπολογιστή στην οποία τρέχει η εφαρμογή. Επομένως, με το να μεταγλωττίζονται τα Java applications σε bytecodes (το instruction set του JVM), η εφαρμογή τους σε υπολογιστικά συστήματα γίνεται αυτόματα καθολική χωρίς περαιτέρω περιορισμούς. Αυτό το χαρακτηριστικό της μαζί με άλλα που την διακρίνουν από άλλες αντικειμενοστρεφείς γλώσσες προγραμματισμού την έχουν κάνει την πιο διάσημη γλώσσα προγραμματισμού εν χρήση και ιδιαίτερα για client-server web applications (διαδικτυακές εφαρμογές πελάτη-εξυπηρετητή), το οποίο και μας ενδιαφέρει στην παρούσα υλοποίηση. Η δημοτικότητά της αυτή έχει οδηγήσει τις κοινότητες των προγραμματιστών να αναπτύξουν πολυπληθείς βιβλιοθήκες για διάφορους σκοπούς όπως για την υλοποίηση αλγορίθμων machine learning (αλγόριθμοι μηχανικής μάθησης).

Στην παρούσα υλοποίηση κάνουμε χρήση μιας τέτοιας βιβλιοθήκης, την οποία περιγράφουμε παρακάτω.

2.10 Μηχανική Μάθηση και βιβλιοθήκες Apache Mahout

Η μηχανική μάθηση (Machine Learning) είναι μια περιοχή της τεχνητής νοημοσύνης η οποία αφορά αλγορίθμους και μεθόδους που επιτρέπουν στους υπολογιστές να “μαθαίνουν”. Με την μηχανική μάθηση καθίσταται εφικτή η κατασκευή προσαρμόσιμων (adaptable) προγραμμάτων υπολογιστών τα οποία λειτουργούν βάσει αυτοματοποιημένης ανάλυσης σε σύνολα δεδομένων και όχι βάσει της διαίσθησης των μηχανικών που τα προγραμμάτισαν. Με χρήση αυτών των αλγορίθμων εξυπηρετούνται μια σειρά από υπολογιστικές διεργασίες (computing tasks) όπως το φιλτράρισμα ανεπιθύμητης αλληλογραφίας στις υπηρεσίες ηλεκτρονικού ταχυδρομείου (spam filtering), στις μηχανές αναζήτησης (search engines), στην οπτική αναγνώριση χαρακτήρων (optical character recognition), και στην τεχνητή όραση (computer vision).

2.10.1 Συστήματα συστάσεων (Recommender Systems)

Το πεδίο χρήσης των αλγορίθμων μηχανικής μάθησης που μας ενδιαφέρει στην υλοποίηση του συστήματός μας είναι η αυτοματοποιημένη σύσταση (recommendation). Τα συστήματα αυτοματοποιημένης σύστασης (Recommender Systems) είναι μια κατηγορία συστημάτων ανάλυσης δεδομένων που έχουν ως στόχο την πρόβλεψη της βαθμολογίας (rating) ή της προτίμησης (preference) που θα έδινε ένας χρήστης σε ένα αντικείμενο του συστήματος. Η προφανής εμπορική τους χρησιμότητα έχει κάνει τα συγκεκριμένα συστήματα πολύ δημοφιλή στο διαδίκτυο και στην ανάπτυξη διαδικτυακών συστημάτων και εφαρμογών. Οι δύο βασικές προσεγγίσεις των recommender systems είναι το συνεργατικό φιλτράρισμα (collaborative filtering) και το φιλτράρισμα βάσει περιεχομένου (content-based filtering). Οι μέθοδοι collaborative filtering κατηγοριοποιούνται περαιτέρω ως memory-based και model-based, και θα τις δούμε παρακάτω πιο αναλυτικά γιατί αποτελούν και την βάση της προσέγγισής μας στο παρούσα υλοποίηση. Οι μέθοδοι collaborative filtering στο σύνολό τους βασίζονται στην συλλογή και την ανάλυση μιας μεγάλης ποσότητας πληροφορίας της συμπεριφοράς, των βαθμολογήσεων και των δραστηριοτήτων των χρηστών του συστήματος, προβλέποντας ποια αντικείμενα θα αρέσουν στους χρήστες βάσει της ομοιότητάς τους με τους υπόλοιπους χρήστες του ίδιου συστήματος. Αντιθέτως οι μέθοδοι content-based filtering βασίζονται στην περιγραφή ενός αντικειμένου του συστήματος και στο προφίλ των προτιμήσεων ενός χρήστη. Με βάση αυτά τα στοιχεία η συγκεκριμένη μέθοδος προσπαθεί να συσχετίσει το περιεχόμενο ενός αντικειμένου προς σύσταση με κάποιο προφίλ προτιμήσεων ώστε να καταλήξει σε κάποια σύσταση ή όχι. Με άλλα λόγια, αυτοί οι αλγόριθμοι προσπαθούν να συστήσουν αντικείμενα τα οποία είναι όμοια με αυτά που άρεσαν στον χρήστη στο παρελθόν. Δεν θα αναλύσουμε περαιτέρω την κατηγορία των content-based αλγορίθμων μιας και όπως αναφέραμε παραπάνω δεν μας ενδιαφέρουν τα χαρακτηριστικά του ταξιδιωτικού πακέτου, ούτε τα χαρακτηριστικά του Χρήστη, παρά μόνο οι σχέσεις (ή αλλιώς προτιμήσεις) των Χρηστών με τους προορισμούς που συνθέτουν το κάθε ταξιδιωτικό πακέτο για λόγους που θα γίνουν ξεκάθαροι παρακάτω, στο κεφάλαιο που περιγράφουμε την υλοποίηση του συστήματος.

2.10.1.1 Memory – Based Αλγόριθμοι

Θα δούμε πιο αναλυτικά το κομμάτι του collaborative filtering που σχετίζεται με τους Memory-Based αλγόριθμους. Η γενική ιδέα των αλγορίθμων αυτής της κατηγορίας, βρίσκεται στα παρακάτω σημεία:

- Ο **Χρήστης Α** και ο **Χρήστης Β** έχουν την *ίδια άποψη* στο **ζήτημα Μ**.
- Ο **Χρήστης Α** *δεν έχει* εκφράσει ακόμα άποψη για το **ζήτημα Μ'** ενώ ο **Β** *έχει εκφράσει*.
- Είναι πολύ πιο πιθανό η **άποψη του Α** για το ζήτημα **Μ'** *να ταυτίζεται* με αυτή του **Β** παρά με μια άποψη ενός **τυχαία επιλεγμένου Χρήστη**.

Αυτή η γενική ιδέα είναι μια πολύ φυσική προσέγγιση του προβλήματος του recommendation, μιας και ακολουθεί την τάση των ανθρώπων να κατηγοριοποιούνται βάσει των προτιμήσεών τους. Μπορεί να μην υπάρχει κάποια καθολική ταύτιση μεταξύ των χαρακτηριστικών δύο

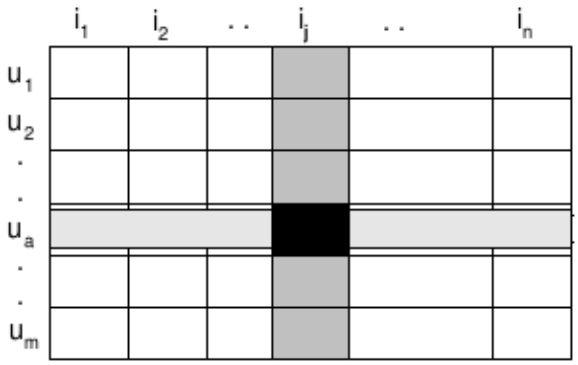
διαφορετικών ατομών αλλά είναι πολύ πιθανό οι προτιμήσεις τους ως προς ένα ένα συγκεκριμένο κομμάτι να ταυτίζονται πλήρως, όπως είναι για παράδειγμα τα μουσικά τους ακούσματα ή η προτιμήσεις τους ως προς τις αγορές τους σε ρούχα. Έτσι, το πρόβλημα της σύστασης αντικειμένων προς τους χρήστες ενός συστήματος ανάγεται σε πρόβλημα εντοπισμού συγγενικών προτιμήσεων. Αυτές οι συγγενικές προτιμήσεις μπορεί να αφορούν ομοιότητες χρηστών ή ομοιότητες αντικειμένων.

Για να γίνει πιο ξεκάθαρη η ομοιότητα των χρηστών σε σχέση με την ομοιότητα των αντικειμένων, αρκεί να σκεφτούμε ότι η συμπεριφορά ενός ανθρώπου ακολουθεί κάποιο μοτίβο (pattern), είτε αυτό συμβαίνει συνειδητά είτε όχι. Από την μία, οι άνθρωποι τείνουν να προτιμούν αντικείμενα τα οποία είναι όμοια με άλλα αντικείμενα που τους άρεσαν σε προηγούμενο χρονικό διάστημα. Έτσι, αν για έναν άνθρωπο το αγαπημένο του πιάτο ζυμαρικών συνοδεύεται από κοτόπουλο, ντομάτα, παρμεζάνα, τότε είναι πολύ πιθανό να του αρέσει και η αντίστιχη pizza με τα ίδια υλικά συν κάποιο ακόμα. Αντίστοιχα, οι άνθρωποι τείνουν να προτιμούν αντικείμενα τα οποία αρέσουν σε ανθρώπους με τους οποίους παρουσιάζουν μια ομοιότητα.

Το βασικό χαρακτηριστικό που πρέπει να αναφερθεί πριν μπούμε στην ανάλυση αυτών των αλγορίθμων, είναι αυτό που γίνεται ξεκάθαρο και από την παραπάνω περιγραφή, ότι η συγκεκριμένη προσέγγιση δεν απαιτεί γνώση των χαρακτηριστικών των αντικειμένων αλλά μόνο τις σχέσεις των προτιμήσεων που έχουν οι χρήστες ως προς τα αντικείμενα που εξετάζονται. Έτσι, δεν έχει σημασία αν προτείνουμε λουλούδια, βιβλία ή φαγητό γιατί κανένα απ'τα χαρακτηριστικά τους δεν μπαίνουν ως είσοδος στον αλγόριθμό μας. Μπορεί να φαίνεται ως μια κακή προσέγγιση γιατί με μια πρώτη ματιά απορρίπτουμε ένα μεγάλο κομμάτι πληροφορίας που αφορά τα χαρακτηριστικά των αντικειμένων. Παρ'όλα αυτά, θα δούμε ότι οι memory-based αλγόριθμοι λειτουργούν πολύ καλά μιας και η πληροφορία που χρειαζόμαστε δεν είναι πάντοτε διαθέσιμη, οπότε η εισαγωγή νέων παραμέτρων στον Recommder που στήνουμε κάθε φορά να λειτουργεί αρνητικά ως προς την ποιότητα των τελικών αποτελεσμάτων. Αυτό βέβαια, δεν αποτελεί σε καμία περίπτωση ανάλυση των Content-based αλγορίθμων που κάνουν ακριβώς χρήση των χαρακτηριστικών των αντικειμένων, και οι οποίοι δίνουν πολύ καλές λύσεις σε domain-specific προβλήματα. Αντιθέτως, σε αυτή την περίπτωση η πρόκληση είναι η επιλογή των χαρακτηριστικών των αντικειμένων που έχουν νόημα για χρήση τους από το σύστημα συστάσεων και σε τί βαθμό.

Σε ένα τυπικό collaborative filtering σενάριο έχουμε :

- ένα σύνολο από m χρήστες $U = \{u_1, u_2, \dots, u_m\}$
- ένα σύνολο από n αντικείμενα $I = \{i_1, i_2, \dots, i_n\}$
- αυτά σχηματίζουν έναν πίνακα προτιμήσεων (preference table) όπως φαίνεται παρακάτω.



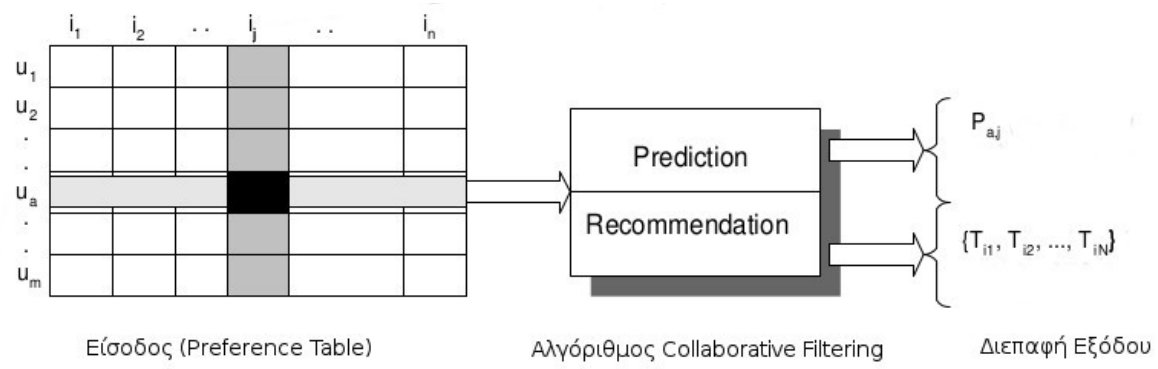
Σχήμα 2.1: Πίνακας προτιμήσεων (Preference Table)

Ο πίνακας αυτός εκφράζει την σχέση των χρηστών με τα αντικείμενα του συστήματος με την έννοια ότι για κάθε κελί (i,j) του πίνακα, είτε δεν υπάρχει τιμή που σημαίνει ότι ο χρήστης της i δεν έχει εκφράσει ακόμα άποψη για το αντικείμενο j, είτε υπάρχει κάποια τιμή που εκφράζει την προτίμηση του χρήστη i για το αντικείμενο j. Αυτή η τιμή μπορεί να είναι μια αριθμητική τιμή που κινείται σε ένα συγκεκριμένο εύρος τιμών ή να προκύπτει απ'τα αρχεία αγορών του χρήστη (αν αγόρασε ένα αντικείμενο ή όχι), απ'το ιστορικό πλοήγησής του κτλ. Υπάρχει, λοιπόν, ένας χρήστης u_a που ανήκει στο σύνολο των χρηστών που δίνεται στο σύστημα και ο οποίος ονομάζεται χρήστης – στόχος . Για τον χρήστη – στόχο ο αλγόριθμος του collaborative filtering καλείται να βρει την προτίμησή του ως προς τα αντικείμενα για τα οποία δεν έχει εκφράσει ακόμα άποψη. Η προτίμησή του αυτή (ή αλλιώς η πιθανότητα να του αρέσουν κάποια αντικείμενα) μπορεί να εκφραστεί με δύο μορφές:

1. **Πρόβλεψη (Prediction)** είναι μια αριθμητική τιμή $P_{a,j}$ που εκφράζει την προβλεπόμενη προτίμηση ενός αντικειμένου j για τον χρήστη-στόχο (α). Φυσικά, αυτή η προβλεπόμενη τιμή βρίσκεται στο ίδιο εύρος τιμών με αυτές που δίνονται απ'τον χρήστη στο σύστημα μέσω της διεπαφής που τους παρέχεται (π.χ. τιμές 1 – 10) .

2. **Σύσταση (Recommendation)** είναι μια λίστα από N αντικείμενα I_r τα οποία θα είναι το περισσότερο αρεστά στον χρήστη – στόχο .

Η περιγραφή που κάνουμε παραπάνω γίνεται πιο ξεκάθαρη στο παρακάτω σχήμα:



Σχήμα 2.2: Διαδικασία Collaborative Filtering

Οι memory-based αλγόριθμοι χρησιμοποιούν ολόκληρο το dataset χρηστών-αντικειμένων για να παράγουν μια πρόβλεψη. Αυτά τα συστήματα κάνουν χρήση στατιστικών τεχνικών για να βρουν ένα σύνολο αντικειμένων ή χρηστών που αποτελούν τους γείτονες του υπο εξέταση και υπό σύσταση αντικειμένου. Οι γείτονες αυτοί ορίζονται είτε ως τα αντικείμενα που μοιάζουν περισσότερο με κάποιο άλλο, είτε ως οι χρήστες οι οποίοι έχουν ιστορικό συμπεριφοράς που συμφωνεί περισσότερο με το αντίστοιχο του χρήστη-στόχου. Οι δύο αυτές προσεγγίσεις είναι γνωστές ως Item – based και User – based recommendation, αντίστοιχα, και τις εξετάζουμε ξεχωριστά:

User – based recommendation

Ποιοτικά, αυτή η κατηγορία των αλγορίθμων, όπως αναφέρθηκε προηγουμένως, αναζητά ένα σύνολο χρηστών οι οποίοι έχουν το ίδιο ιστορικό συμπεριφοράς με τον χρήστη-στόχο. Έτσι, σε ένα υποθετικό σενάριο όπου το ζητούμενο είναι η αγορά των καλύτερων μουσικών δίσκων για έναν άνθρωπο, το πρώτο βήμα που θα έκανε ένας user-based αλγόριθμος θα ήταν να βρει τους φίλους του που έχουν τις πιο κοντινές μουσικές προτιμήσεις. Θα κοιτούσε δηλαδή τον περίγυρο αυτού του ατόμου για να δει ποιανών φίλων του οι αγορές τους -ως προς τους μουσικούς δίσκους- ταυτίζονται περισσότερο με τις μέχρι τώρα αγορές του συγκεκριμένου ατόμου. Έτσι ορίζεται η “γειτονιά” του ατόμου αυτού με αλγοριθμικούς όρους. Στην συνέχεια, θα συνδύαζε τις προτιμήσεις των γειτόνων του ως προς τους μουσικούς δίσκους που ακόμα δεν έχει αγοράσει το άτομο αυτό ώστε να υπολογίσει μια εκτίμηση προτίμησης για αυτούς τους δίσκους.

Κατ'ανάλογο τρόπο, σε ένα data set οι user-based αλγόριθμοι χρησιμοποιούν στατιστικές τεχνικές ώστε να βρουν ένα σύνολο από γείτονες ως προς τον χρήστη-στόχο με την λογική ότι οι γείτονες έχουν την ιδιότητα το ιστορικό τους να ταιριάζει με το ιστορικό του χρήστη-στόχου. Αυτό μπορεί να σημαίνει ότι οι γείτονες και ο χρήστης-στόχος είτε έχουν βαθμολογήσει διαφορετικά αντικείμενα κατά τον ίδιο τρόπο είτε ότι απλά έχουν αγοράσει ίδια αντικείμενα. Μόλις εντοπιστεί η γειτονιά του χρήστη-στόχου, τότε εφαρμόζουν άλλους αλγόριθμους για να συνδυάσουν τα preferences των γειτόνων ώστε να παράγουν μια πρόβλεψη ή ένα σύνολο Ν-καλύτερων συστάσεων για τον χρήστη-στόχο.

Αν και αυτή η οικογένεια των συστημάτων ήταν πολύ δημοφιλής στην χρήση και την αποδοτικότητά τους έχουν εμφανίσει κάποιες αδυναμίες όπως:

1. **Αραιότητα των δεδομένων (data sparsity)** Σε πραγματικά εμπορικά συστήματα, τα recommender systems αναλύουν πολύ μεγάλα data sets (π.χ. amazon.com). Σε αυτά τα data sets, ακόμα και οι πιο ενεργοί χρήστες έχουν αγοράσει

λιγότερο από το 1% των αντικειμένων που σημαίνει ότι σε ένα σύνολο 2 εκατομμυρίων αντικειμένων, αυτό το ποσοστό μεταφράζεται σε 20.000 αντικείμενα, το οποίο και πάλι δεν ανταποκρίνεται στην πραγματικότητα. Επομένως, τα συστήματα που αναζητούν γείτονες σε τέτοιου είδους data sets -και πιο συγκεκριμένα item-sets – αδυνατούν να βρούν σύνολο χρηστών που να αποτελεί την γειτονιά του χρήστη-στόχου και επομένως να κάνουν κάποια σύσταση (ή ακόμα και αν την κάνουν αυτή να είναι όντως ποιοτική).

2. Κακή επεκτασιμότητα (scalability) Η διαδικασία εύρεσης γειτονιάς για έναν χρήστη-στόχο και εν συνεχεία η εξαγωγή των καλύτερων συστάσεων έχει μεγάλες υπολογιστικές απαιτήσεις . Οι απαιτήσεις αυξάνουν συναρτήσει και των χρηστών και των αντικειμένων, και επομένως σε data-sets του μεγέθους που αναφέρθηκε προηγουμένως, η εκτέλεση των συγκεκριμένων αλγορίθμων σε τυπικά web-based συστήματα θα προκαλέσει σοβαρά προβλήματα επεκτασιμότητας.

Item-based recommendation

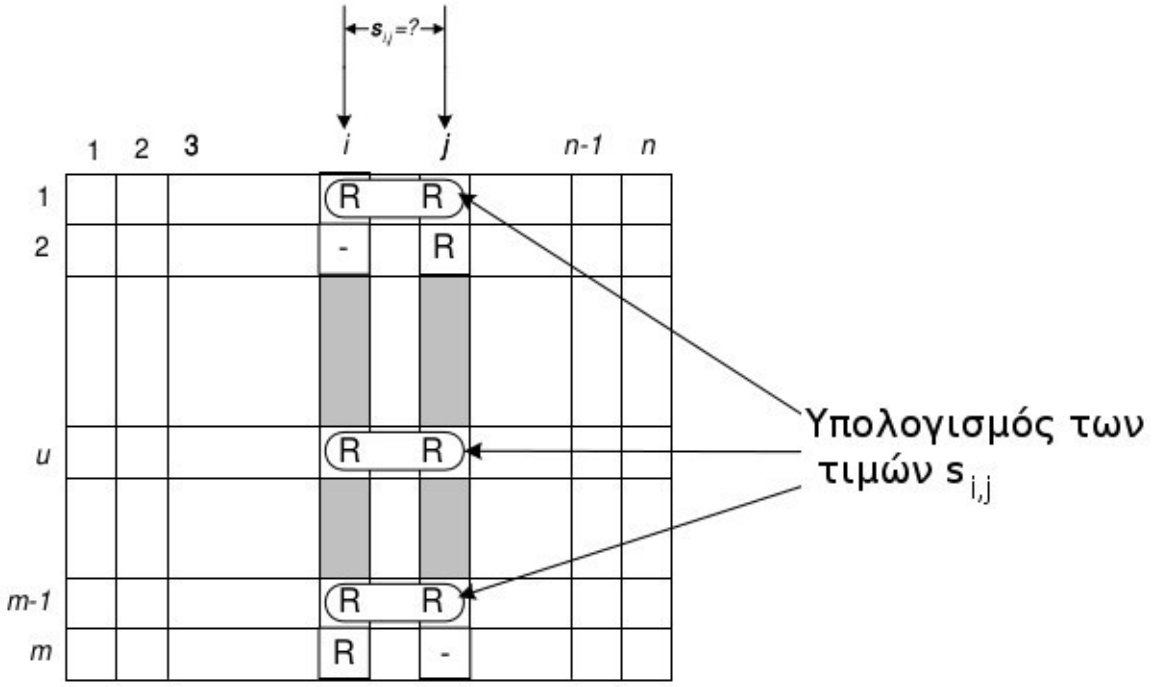
Κατ'ανάλογο τρόπο το item-based recommendation αναζητά ομοιότητα μεταξύ αντικειμένων και επιστρέφει τα καλύτερα αποτελέσματα για έναν χρήστη. Έτσι, στο παράδειγμά μας για την αγορά μουσικού δίσκου για ένα άτομο, ο item-based αλγόριθμος θα αγνοούσε τον περίγυρο του χρήστη-στόχου και θα επικεντρωνόταν στην εύρεση “γειτονιάς” ενός μουσικού δίσκου προς αγορά. Αν δηλαδή ο αλγόριθμος παρατηρούσε ότι το συγκεκριμένο άτομο δείχνει μεγάλη προτίμηση για το αντικείμενο “χ” και όποιος δείχνει προτίμηση για τον μουσικό δίσκο “χ” αγοράζει συνηθως και τα “ψ,ζ,ω”, τότε θα επέλεγε ένα απο αυτά τα αντικείμενα για σύσταση στο συγκεκριμένο άτομο.

Σε επίπεδο data-set, ο αλγόριθμος στο πρώτο του βήμα παίρνει το σύνολο των αντικειμένων για τα οποία ο χρήστης-στόχος έχει εκφράσει άποψη και υπολογίζει το πόσο όμοια είναι με το αντικείμενο-στόχο i, και τέλος επιλέγει τα k πιο όμοια αντικείμενα $\{i_1,i_2,...,i_k\}$. Ταυτόχρονα, υπολογίζονται οι ομοιότητες $\{s_{i1},s_{i2},...,s_{ik}\}$. Στο δεύτερο βήμα, μόλις βρεθούνε τα πιο όμοια αντικείμενα, η πρόβλεψη υπολογίζεται παίρνοντας τον σταθμισμένο μέσο όρο τον βαθμολογήσεων του χρήστη-στόχου, σε αυτά τα όμοια αντικείμενα. Αυτά τα δύο βήματα αποτελούν τα διακριτά βήματα υπολογισμού των συστάσεων τα οποία ονομάζονται υπολογισμός ομοιότητας, και παραγωγή προβλέψεων αντίστοιχα.

Το πρώτο σημαντικό βήμα για τον item-based recommender είναι το να υπολογίσει την ομοιότητα μεταξύ των αντικειμένων και μετά να επιλέξει τα πιο όμοια αντικείμενα. Για να προχωρήσει στον υπολογισμό δύο αντικειμένων (i,j), απομονώνει πρώτα τους χρήστες που έχουν βαθμολογήσει και τα δύο αντικείμενα. Αφού απομονώνει τους χρηστες αυτούς, εφαρμόζει ένα κριτήριο ομοιότητας όπως το Pearson correlation similarity, cosine-based similarity (για binary data – δηλαδή για δεδομένα που έχουν τιμές 0,1) , tanimoto similarity, και υπολογίζει την τιμή $s_{i,j}$. Η διαδικασία αυτή φαίνεται καθαρά στο σχήμα 2.3, όπου φαίνεται η απομόνωση των χρηστών που έχουν βαθμολογήσει τα i,j με την λογική ότι έχουν δώσει μια βαθμολόγηση R και στα δύο αντικείμενα. Και στην συγκεκριμένη εικόνα, αυτοί οι χρήστες που απομονώνονται είναι 1, u και m-1.

Αφού γίνει η απομόνωση των πιο όμοιων αντικειμένων βάση των παραπάνω τεχνικών, το επόμενο βήμα του συστήματος συστάσεων είναι να ελέγξει τις αξιολογήσεις/βαθμολογήσεις των χρηστών-στόχων και να χρησιμοποιησει κάποια τεχνική για να παράγει τις ζητούμενες προβλέψεις ως προς τα αντικείμενα για τα οποία δεν έχει εκφράσει κάποιο ενδιαφέρον μέχρι

τώρα.



Σχήμα 2.3: Υπολογισμός όμοιότητας αντικειμένων i,j

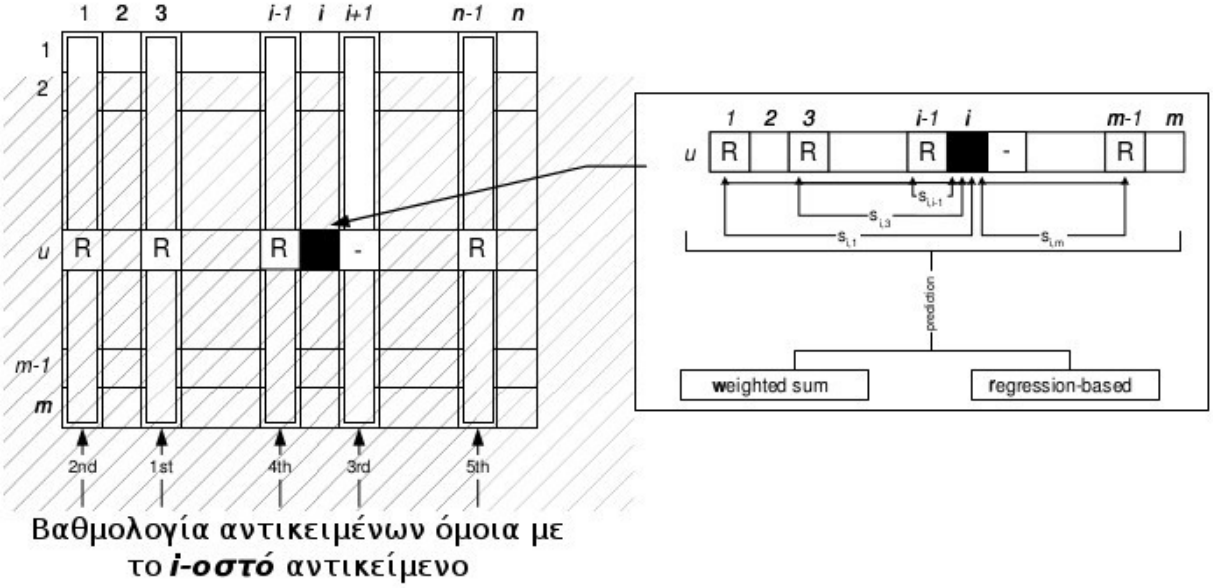
Η τεχνική που αναφέρουμε εδώ, που είναι και η πιο διαδεδομένη, είναι η τεχνική του σταθμισμένου αθροίσματος (weighted sum). Στόχος μας είναι η πρόβλεψη της τιμής ενός αντικειμένου i από έναν χρήστη-στόχο u, και όπως δηλώνει και το όνομα της συγκεκριμένης τεχνικής αυτό το κάνει υπολογίζοντας το άθροισμα των βαθμολογήσεων που έχει κάνει ο χρήστης u στα αντικείμενα που είναι όμοια του i. Κάθε βαθμολόγηση, όμως, σταθμίζεται από τον αντίστοιχο βαθμό ομοιότητας που έχουμε υπολογίσει προηγουμένως. Αυτός ο βαθμός ομοιότητας δεν είναι άλλος απ'την τιμή $s_{i,j}$ για κάθε ζεύγος αντικειμένων (i,j). Η σχέση που μας δίνει την πρόβλεψη της τιμής για τον χρήστη-στόχο u ως προς το αντικείμενο i, φαίνεται παρακάτω:

$$P_{u,i} = \frac{\sum_{\text{all similar items, N}} (s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, N}} (|s_{i,N}|)}$$

Όπου $R_{u,N}$ είναι οι βαθμολογήσεις του χρήστη-στόχου u προς όλα τα όμοια αντικείμενα N σε σχέση με το αντικείμενο i. Ουσιαστικά, αυτή η προσέγγιση προσπαθεί να αποτυπώσει τον τρόπο βαθμολόγησης του χρήστη-στόχου προς τα N όμοια αντικείμενα. Το σταθμισμένο άθροισμα του αριθμητή διαιρείται με το άθροισμα των όρων ομοιότητας ώστε να εξασφαλιστεί το ότι η τιμή της πρόβλεψης P θα είναι μέσα στο ορισμένο εύρος τιμών αξιολογήσεων. Αν για παράδειγμα οι χρήστες δίνουν αξιολογήσεις που κινούνται στις τιμές 1...5, τότε πρέπει και η τιμή πρόβλεψης να είναι στο αντίστοιχο εύρος τιμών.

Φυσικά, υπάρχουν και άλλες τεχνικές προβλέψεων. Μια από αυτές τις τεχνικές είναι η regression-based τεχνική η οποία δεν υπολογίζει τις δοσμένες τιμές αξιολόγησης των N όμοιων αντικειμένων όπως δίνονται από το data-set, αλλά τις τιμές προσέγγισης που

προκύπτουν βασει ενός linear regression model. Από κει και πέρα, το σταθμισμένο άθροισμα χρησιμοποιείται ως έχει. Χωρίς να μπαίνουμε σε εκτενή ανάλυση της συγκεκριμένης τεχνικής, αξίζει να την αναφέρουμε διότι έρχεται να αντιμετωπίσει ένα ουσιαστικό πρόβλημα που αντιμετωπίζεται συχνά απ'την απλή προσέγγιση του σταθμισμένου αθροίσματος. Πολλές φορές, οι βαθμοί ομοιότητας που υπολογίζονται απ'τις τεχνικές correlation similarity και cosine-based similarity, είναι αποπροσανατολιστικοί με την έννοια ότι δύο διανύσματα βαθμολόγησης μπορεί να έχουν μεγάλη απόσταση με την έννοια της Ευκλείδιας απόστασης, αλλά να παρουσιάζουν μεγάλη ομοιότητα βάσει των παραπάνω τεχνικών. Παρακάτω φαίνεται σχηματοποιημένη η διαδικασία που περιγράφηκε για το βήμα του υπολογισμού των προβλέψεων όπως υλοποιείται από τα συστήματα συστάσεων που χρησιμοποιούν item-based recommendation.

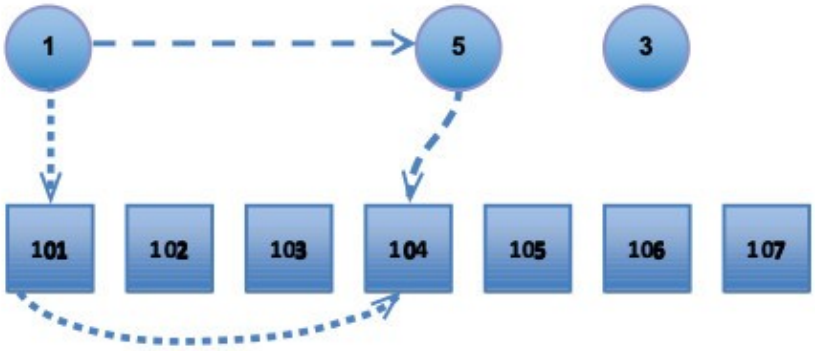


Σχήμα 2.4: Η διαδικασία παραγωγής προβλέψεων με την χρήση 5 όμοιων αντικειμένων

Όσον αφορά τα προβλήματα του item-based recommendation, ξεκινάμε με το πρόβλημα της ψυχρής εκκίνησης. Αυτό το πρόβλημα εμφανίζεται σε πολλές διαφορετικές αλγοριθμικές προσεγγίσεις όπως και στο user-based recommendation που αναφέραμε παραπάνω. Φυσικά κάτι τέτοιο θα μπορούσε να αντιμετωπιστεί εν μέρει από την καταγραφή διαφορετικού τύπου πληροφοριών που αποτυπώνουν έμμεσα τις προτιμήσεις των χρηστών, όπως είναι το ιστορικό πλοήγησης τους, ή ο χρόνος παραμονής τους σε μια ιστοσελίδα κτλ.

Επιπλέον, ένα πρόβλημα που αντιμετωπίσαμε και στο user-based recommendation είναι το πρόβλημα του data-sparsity. Προφανώς για μεγάλα data-set και μικρά ποσοστά συμμετοχής των χρηστών στο input των αλγορίθμων αυτών μέσω της αξιολόγησης των αντικειμένων, καθιστούν δύσκολη την εύρεση όμοιων αντικειμένων το οποίο είναι και το βασικό βήμα του σύστηματος πριν κάνει συστάσεις. Σε αυτό το σημείο όμως η item-based προσέγγιση υπερτερεί με την έννοια ότι η ανάλυση των δεδομένων ακολουθεί - με μια έννοια- μικρότερη διαδρομή για την εύρεση κατάλληλων προς σύσταση αντικειμένων προς τον χρήστη-στόχο. Αυτό προκύπτει από το γεγονός ότι στην user-based προσέγγιση το σύστημα πρέπει πρώτα να εντοπίσει χρήστες συγγενικούς ως προς τον χρήστη-στόχο και στην συνέχεια

να αναλύσει περαιτέρω τα δεδομένα του data-set ωστε να καταλήξει σε έναν αριθμό συστάσεων. Αντιθέτως, η item-based προσέγγιση κοιτάει κατευθείαν τις βαθμολογήσεις των αντικειμένων για να εξάγει μια σχέση μεταξύ τους, και από κει και πέρα η σύσταση στον χρήστη-στόχο γίνεται βάσει της προσωπικής του συμμετοχής με την έννοια ότι όσο περισσότερο συμμετέχει σε αξιολογήσεις, τόσο ανεβαίνει και η ποιότητα των συστάσεων που του γίνονται από το σύστημα. Απ'την, άλλη ακόμα και να είναι ελάχιστη η συμμετοχή του χρήστη-στόχου, η ποιότητα των συστάσεων μπορεί να βελτιώνεται απ'την συμμετοχή των υπολοίπων χρηστών καθώς συμβάλλουν στην καλύτερη αποτύπωση των σχέσεων μεταξύ των αντικειμένων για το σύνολο των χρηστών. Αυτή η διαφορά αποτυπώνεται καλύτερα και στο παρακάτω σχήμα. Φαίνεται καθαρά ότι η διαδρομή εύρεσης γειτονικών χρηστών ως προς τον χρήστη-στόχο αντιμετωπίζει μεγαλύτερο πρόβλημα στην αραιότητα δεδομένων μιας και η εύρεση γειτόνων ως προς τον χρήστη-στόχο και εν συνεχεία η εύρεση αντικειμένων που να συμφωνούν και με τους γείτονες και με αντικείμενα τα οποία δεν έχει αξιολογήσει ο χρήστης-στόχος, απαιτεί περισσότερη γνώση απ'το data-set η οποία δεν είναι διαθέσιμη.



Σχήμα 2.5: Στην user-based (χοντρές διακεκομμένες γραμμές) προσέγγιση, εντοπίζονται οι όμοιοι χρήστες και βλέπει ποιά αντικείμενα τους αρέσουν ενώ στην item-based (λεπτές διακεκομμένες) βλέπει τί αρέσει στον χρήστη και εντοπίζει τα όμοια αντικείμενα

Ένα πρόβλημα που επικεντρώνεται στα συστήματα που χρησιμοποιούν item-based recommendation είναι το πρόβλημα της απόδοσης βάσει δημοφιλών προτιμήσεων (popular taste problem). Αυτό σημαίνει ότι για αντικείμενα που έχουν επιλεχθεί πολλές φορές – από πολλούς διαφορετικούς χρήστες- υπαρχει μια καλή αποτύπωση των σχέσεών τους, μιας και υπάρχει διαθέσιμη μεγάλη πληροφορία τόσο για τα ίδια όσο και για τα αντικείμενα που πρόκειται να ορίσουν την γειτονιά τους. Έτσι, οι συστάσεις που αφορούν τα συγκεκριμένα δημοφιλή αντικείμενα, είναι επιτυχημένα. Τα αντικείμενα όμως για τα οποία δεν υπάρχει πληροφορία, είναι δύσκολο έως και αδύνατο να συσταθούν, μιας και το similarity τους που υπολογίζεται στο πρώτο αλγοριθμικό βήμα του συστήματος θα είναι πολύ μικρό έως μηδαμινό, αποκλείοντάς τα ακόμα και από χρήστες που ενδιαφέρονται γι'αυτά.

Απ'την άλλη πλευρά, στο κομμάτι των πλεονεκτημάτων, ξεκινάμε με το αντίστροφο φαινόμενο του popular taste problem. Και αυτό αφορά την αύξηση της ποιότητας των συστάσεων του συστήματός μας με το πέρασμα του χρόνου. Αυτό προκύπτει από το γεγονός ότι τα αντικείμενα αλλάζουν με μικρότερο ρυθμό από τους χρήστες και τις προτιμήσεις τους. Έτσι, ενώ οι χρήστες μπορεί χρονικά να δίνουν αξιολογήσεις οι οποίες κάνουν πιο ασαφή -ως προς το σύστημά μας- τον τρόπο με τον οποίο επιλέγουν τις αξιολογήσεις και τα προϊόντα που αγοράζουν, τα αντικείμενα παραμένουν ως έχουν και η ομοιότά τους συγκλίνει με την

πάροδο του χρόνου σε μια τιμή η οποία έχει επιβεβαιωθεί από τα δεδομένα πολλών διαφορετικών χρηστών, για τους οποίους μάλιστα δεν μας ενδιαφέρει να συμπεράνουμε να καταναλωτικά τους χαρακτηριστικά. Έτσι, ακόμα και ένας χρήστης που έχει κάνει μόνο δύο αγορές απ'το σύστημά μας (ανενεργός χρήστης) μας δίνει σημαντική πληροφορία για την λειτουργία του item-based recommender μας, την στιγμή που ο user-based θα αδυνατούσε να βγάλει ποιοτικά συμπεράσματα για τους γείτονές του.

Στην συνέχεια αναφερόμαστε στο πλεονέκτημα της συγκεκριμένης προσέγγισης που έχει να κάνει με το domain independence που την χαρακτηρίζει. Φυσικά αυτό το χαρακτηριστικό αφορά και την user-based προσέγγιση με την διαφορά ότι η item-based λογική μας δίνει κάποια προτερήματα για την οποία την επιλέγουμε στην υλοποίηση του συστήματος συστάσεων της παρούσας εργασίας, και γι'αυτό την αναφέρουμε εκτενώς εδώ. Domain independent σημαίνει αδιαφορία για τα χαρακτηριστικά των αντικειμένων, και άρα ευκολία στην προσέγγιση μεγαλύτερης κλάσης προβλημάτων σε ένα διαδικτυακό σύστημα. Μπορεί με μια πρώτη ματιά αυτό να φαίνεται σαν μια κακή προσέγγιση με την έννοια ότι αγνούμε πληροφορία η οποία θεωρητικά θα μας έδινε πάτημα για καλύτερη ανάλυση των καταναλωτικών προτιμήσεων των χρηστών του εκάστοτε συστήματος. Από την άλλη όμως, μπορεί να πρέπει να διαχειριστούμε αντικείμενα τα οποία παρουσιάζουν μεγάλη ανομοιογένεια ως προς τα χαρακτηριστικά τους, ενώ συνυπάρχουν στο σύστημά μας. Έτσι, στην ιδανική περίπτωση θα έπρεπε να διαχειριστούμε μόνο -για παράδειγμα- βιβλία, όπου η ομοιογένεια ορίζεται ως προς τα χαρακτηριστικά του συγγραφέα, του εκδοτικού οίκου και του περιεχομένου. Όμως κάτι τέτοιο δεν συμβαίνει όταν στο σύστημά μας έχουμε να διαχειριστούμε -σαν ένα πολύ πιθανό σενάριο- ηλεκτρονικές συσκευές στην γενικότητά τους, από οικιακά σκεύη μέχρι υπολογιστές. Σε μια τέτοια περίπτωση μια content-based προσέγγιση μπορεί να έδινε λύσεις ή να μην έδινε ποιοτικές λύσεις λόγω κακής επιλογής χαρακτηριστικών που θα αποτελούσαν είσοδο στον αλγόριθμό μας. Αποδεδειγμένα, το να αγνούμε πληροφορία -ως ένα επίπεδο- μπορεί να λειτουργεί ευεργετικά ως προς την ποιότητα των συστάσεων του συστήματός μας.

Παράλληλα, η item-based προσέγγιση μπορεί να διαποτιστεί με content-based χαρακτηριστικά ανάλογα με τις ανάγκες μας. Πιο συγκεκριμένα, θα μπορούσαμε στην έξοδο του item-based recommender, να ορίσουμε ποιά από τα αντικείμενα προς σύσταση έχουν περισσότερο νόημα για ένα συγκεκριμένο χρήστη, χωρίς να χαλάμε την γενικότητα του συστήματός μας. Άρα, στο παράδειγμα του διαδικτυακού συστήματος πώλησης ηλεκτρονικών ειδών, θα μπορούσαμε να ελέγχουμε το αν ένας χρήστης επιλέγει αυστηρά αντικείμενα που έχουν να κάνουν με τους ηλεκτρονικούς υπολογιστές, και να ενισχύουμε την βαθμολόγηση, που έχει προκύψει από τον recommender μας, των αντικειμένων που έχουν να κάνουν με υπολογιστές. Έτσι παρατηρούμε ότι η ευελίξία που μας δίνει ο item-based recommender μας βοηθά και στον να κρατάμε την απόδοσή του σε υψηλά επίπεδα τόσο σύνολο των αντικειμένων όσο και σε domain-specific προτάσεις. Ταυτόχρονα, οι μειωμένες υπολογιστικές απαιτήσεις της συγκεκριμένης προσέγγισης -σε σχέση με άλλες- διατηρούνται, μιας και οποιοδήποτε content-based injection επιχειρούμε να κάνουμε γίνεται σε μόλις γραμμικό χρόνο σε σχέση με την έξοδο του συστήματος, η οποία είναι σαφώς μικρότερη του συνόλου των αντικειμένων που υπόκεινται σε επεξεργασία, και άρα με εισαγωγή αμελητέας χρονικής πολυπλοκότητας.

Ένα ακόμα χαρακτηριστικό των item-based recommenders που τους κάνει να υπερτερούν των user-based recommeders, είναι ότι η χρονική τους πολυπλοκότητα αυξάνει μόνο συναρτήσει του αριθμού των αντικείμενων. Αυτό, και σε συνδυασμό με το γεγονός ότι έχουμε και σενάρια πολύ περιστασιακών χρηστών, όπως χρήστες που έχουν αγοράσει ένα μόνο αντικείμενο, ή πλοηγούνται απο γενικό ενδιαφέρον χωρίς προοπτική να αγοράσουν και να αξιολογήσουν κάποιο αντικείμενο, μας οδηγεί στο συμπέρασμα ότι οι item-based recommenders έχουν πολύ μεγαλύτερη δυνατότητα επεκτασιμότητας (scalability), την στιγμή

που οι user-based recommenders αναγκάζονται να αναλύουν όλες τις ασθενείς και μη, σχέσεις μεταξύ των χρηστών πριν καταλήξουν σε κάποιο βήμα πρόβλεψης.

Η προηγούμενη λογική που αφορά την σχέση των μεγεθών μεταξύ των συνόλων των χρηστών και των αντικειμένων, μας οδηγεί και στην σκέψη ότι στην γενική περίπτωση, ο αριθμός των χρηστών αυξάνει με μεγαλύτερο ρυθμό από τον αριθμό των αντικειμένων ενός συστήματος. Αυτό προκύπτει από το γεγονός ότι οι χρήστες ακολουθουν κατά βάση αυξητικό ρυθμό. Δηλαδή ακόμα και ένας ανενεργός χρήστης μπορεί να έχει έναν λογαριασμό σε ένα σύστημα για αρκετό χρόνικό διάστημα πριν απενεργοποιηθεί προσωρινά με το πέρας του χρονικού διαστήματος που έχει οριστεί απ'την υλοποίηση του συστήματος. Μάλιστα, αυτό το χρονικό διάστημα είναι μεγάλο γιατί δεν είναι επιθυμητό να απενεργοποιείται γρήγορα ο λογαριασμός ενός χρήστη ανά μικρά χρονικά διαστήματα μιας και κάτι τέτοιο εισάγει πολυπλοκότητα στην πλευρά του χρήστη ώστε να επαναφέρει τον λογαριασμό εν δράσει, και αυτό μπορεί να τον αποθαρρύνει να επισκεφτεί ξανά το συγκεκριμένο σύστημα. Ορισμένα συστήματα μπορεί να μην απενεργοποιούν και καθόλου λογαριασμούς, ενώ σε κάθε περίπτωση νέοι χρήστες εμφανίζονται συνεχώς, ιδιαίτερα αν το διαδικτυακό σύστημα έχει και εμπορική επιτυχία. Παράλληλα, η δημιουργία λογαριασμού είναι μια απλή και γρήγορη διαδικασία και επομένως ένας χρήστης το κάνει εύκολα ακόμα και για να έχει πρόσβαση σε όλα τα σημεία του συστήματος χωρίς να έχει πάντα την πρόθεση να αγοράσει και να αξιολογήσει. Απ'την άλλη μεριά, τα αντικείμενα τείνουν να αυξάνουν με χαμηλότερο ρυθμό, ή ακόμα και να διατηρείται ο αριθμός τους αν σκεφτεί κάποιος το σενάριο αντικειμένων που έχουν συγκεκριμένο χρόνο ζωής και αντικαθίστανται από νέες εκδόσεις τους με αποτέλεσμα να μην υπάρχει αύξηση του συνολικού αριθμού των αντικειμένων που διαχειρίζεται το σύστημα – για παράδειγμα στο δικό μας σύστημα στο οποίο ένα πακέτο παύει να έχει ισχύει με το πέρας της ημερομηνίας και αντικαθίσταται από ένα πακέτο ίδιου περιεχομένου αλλά με διαφορετικές ημερομηνίες από το εκάστοτε ταξιδιωτικό γραφείο.

Επομένως, η item-based προσέγγιση έχει σαφώς λιγότερες υπολογιστικές απαιτήσεις από την user-based προσέγγιση όπως την εξετάσαμε παραπάνω. Οι υπολογιστικές απαιτήσεις μεταφράζονται χρονικά σε γρηγορότερους ρυθμούς επιστροφής συστάσεων από τον recommender κάτι που είναι κρίσιμο για ένα διαδικτυακό σύστημα. Οι γρηγορότεροι υπολογισμοί συστάσεων δίνουν χώρο και για επιπλέον εισαγωγή πληροφορίας στον recommender μας προκειμένου να αυξήσουμε την ακρίβεια των συστάσεων μας. Έτσι, για παράδειγμα, μπορούμε να επιχειρήσουμε να εισάγουμε πληροφορία από το browsing history των χρηστών ώστε να συμπληρώσουν τα κενά των βαθμολογήσεων που παίζουν σημαντικό ρόλο στην διαδικασία υπολογισμού της ομοιότητας των αντικειμένων. Οι απαιτήσεις σε μνήμη και επεξεργαστική ικανότητα δεν αναλύονται μιας και οι ηλεκτρονικοί υπολογιστές που λειτουργούν ως server, έχουν πολύ καλά χαρακτηριστικά ως προς το υλικό τους ωστε να τους κάνουν ικανούν να μην αντιμετωπίζουν προβλήματα για μεγάλα dataset μέχρι και για δεδομένα της τάξης του ενός εκατομμυρίου. Φυσικά για μεγαλύτερα προβλήματα καταφεύγουμε σε λύσεις distributed computation – όπως πχ με την χρήση του Hadoop – σε συστάδες υπολογιστών.

Τέλος, παρατηρούμε ότι τα αντικείμενα ενός διαδικτυακού συστήματος είναι λιγότερο επιρρεπή σε αλλαγές σε σχέση με τους χρήστες. Έτσι, για παράδειγμα, ένας χρήστης μπορεί να αλλάζει προτίμηση σε επίπεδο ταινιών ανά τακτά χρονικά διαστήματα υποχρεώνοντας το σύστημα σε επιπλέον απαιτητικούς υπολογισμούς συσχέτισης του με τους υπόλοιπους χρήστες. Μια ταινία όμως δεν αλλάζει εύκολα την γειτονιά της, μιας και οι χρήστες αποτυπώνουν συσχετίσεις που τείνουν να συμφωνούν με το πέρασμα του χρόνου. Έτσι, μια επιπλέον δυνατότητα επεκτασιμότητας φαίνεται να εμφανίζεται, και αυτό πραγματοποιείται μέσω προεπεξεργασίας δεδομένων. Η προεπεξεργασία γίνεται με την λογική ότι μπορεί να υπολογίζεται ανά ορισμένα χρονικά διαστήματα ένα μοντέλο των αντικειμένων που προκύπτει από ένα σύνολο λιστών, μια για κάθε αντικείμενο, που περιέχει την γειτονιά του. Από κει και

πέρα, η διαδικασία υπολογισμού των προβλέψεων συνεχίζει όπως περιγράφηκε προηγουμένως, με το μοντέλο των αντικειμένων να ανανεώνεται ανα συγκεκριμένα χρονικά διαστήματα όπως ορίζεται και κρίνεται κατάλληλο απ'την εκάστοτε υλοποίηση.

Κάναμε εκτενή αναφορά στο κομμάτι της item-based προσέγγισης λόγω του ότι πληροί τις προϋποθέσεις για επιλογή στην υλοποίηση του συστήματός μας και η οποία αναφέρεται στο σχετικό κεφάλαιο.

2.10.1.2 Model – Based Αλγόριθμοι

Στο συγκεκριμένο υποκεφάλαιο θα κάνουμε και μια σύντομη αναφορά στην κατηγορία των model-based συστημάτων μιας και αφορούν μια επίσης δημοφιλή κατηγορία αλγοριθμικών προσεγγίσεων για το πρόβλημα των συστάσεων. Επιπλέον, με την μέχρι τώρα ανάλυσή μας και με τα model-based συστήματα, καλύπτεται ένα μεγάλο φάσμα των τεχνολογιών που καλύπτουν το κομμάτι των συστημάτων συστάσεων. Έτσι, γίνονται πιο ξεκάθαροι και οι λόγοι της τελικής επιλογής της προσέγγισης που επιλέξαμε για το κομμάτι της υλοποίησης του συστήματος.

Τα model-based συστήματα συστάσεων εμπλέκουν την κατασκευή ενός μοντέλου του τρόπου αξιολόγησης των χρηστών βασισμένου στο data-set των αξιολογήσεων. Πιο συγκεκριμένα, η προσέγγιση χρησιμοποιεί πληροφορίες από το data-set και το μετατρέπει σαν ένα μοντέλο για να πραγματοποιήσει συστάσεις χωρίς να χρειάζεται να χρησιμοποιεί ολόκληρο το data-set κάθε φορά που καλείται να υπολογίσει συστάσεις. Συνήθως, η κατασκευή των μοντέλων αυτών γίνεται με την χρήση Machine Learning αλγορίθμων – Bayesian network, Clustering, Association rule mining, κτλ.

Αν και η βασική ιδέα πίσω από τα model-based recommendation systems είναι η ίδια, υπάρχουν διαφορετικές προσεγγίσεις για την κατασκευή του μοντέλου και την χρήση του. Μερικές από τις πιό βασικές προσεγγίσεις είναι:

1. **Πιθανοτική προσέγγιση (Probability problem)** Σύμφωνα με αυτή την προσέγγιση, το πρόβλημα της πρόβλεψης μιας βαθμολόγησης για ένα ζεύγος χρήστη-στόχου και αντικειμένου, αντιμετωπίζεται ως πρόβλημα πρόβλεψης της πιθανότητας μια βαθμολόγησης να είναι μια συγκεκριμένη τιμή. Τα Bayesian networks, και οι Clustering αλγόριθμοι χρησιμοποιούν την συγκεκριμένη ιδέα. Τα Baysian networks, είναι κατευθυνόμενοι ακυκλικοί γράφοι των οποίων οι κόμβοι αντιπροσωπεύουν τυχαίες μεταβλητές και οι ακμές εξαρτήσεις υπό συνθήκη. Δύο κόμβοι που δεν συνδέονται αντιπροσωπεύουν δύο ανεξάρτητες μεταβλητές. Κάθε κόμβος, όμως, σχετίζεται με μια πιθανοτική συνάρτηση που δέχεται ως όρισμα ένα συγκεκριμένο σύνολο μεταβλητών που είναι οι μεταβλητές οι οποίες αντιστοιχούν στους κόμβους-πατέρα, και ως αποτέλεσμα δίνει την πιθανότητα της μεταβλητής που αντιπροσωπεύεται από τον συγκεκριμένο κόμβο. Απ'την άλλη το μοντέλο που κατασκευάζεται από τους clustering αλγόριθμους, χειρίζεται το πρόβλημα του συνεργατικού φιλτραρίσματος (collaborative filtering) ως πρόβλημα κατηγοριοποίησης (classification problem). Οι clustering αλγόριθμοι, συσταδοποιούν όμοιους χρήστες στην ίδια κατηγορία/κλάση υπολογίζοντας την πιθανότητα ένας χρήστης να ανήκει σε μια συγκεκριμένη κλάση X και στην συνέχεια υπολογίζει την πιθανότητα των βαθμολογήσεων των συγκεκριμένων χρηστών.

2. **Ενισχυμένοι memory-based αλγόριθμοι** Αναφέραμε αυτή την λογική στο κομμάτι των item-based αλγορίθμων, ως ένα κομμάτι προεπεξεργασίας των δεδομένων για αύξηση του scalability και της ταχύτητας των συγκεκριμένων συστημάτων. Ουσιαστικά, η αποθήκευση των τιμών ομοιότητας για επαναχρησιμοποίησή τους χωρίς την ανανέωσή τους κάθε φορά που απαιτείται κάποια σύσταση, συνιστά ένα μοντέλο των αντικειμένων. Πώς δηλαδή αυτά σχετίζονται μεταξύ τους. Στην model-based προσέγγιση, αναφέρεται χαρακτηριστικά η προτίμηση κατασκευής μοντέλου των αντικειμένων και όχι των χρηστών ακριβώς για λόγους υπολογιστικών απαιτήσεων. Αυτό επιβεβαιώνει την παραπάνω ανάλυσή μας και μάλιστα δίνεται πάτημα για επιπλέον scalability στο σύστημα με την τεχνική του trimming του μοντέλου. Με τον όρο αυτό, εννοείται ο περιορισμός του αριθμού των όμοιων αντικειμένων (ή και χρηστών αν θεωρηθεί σκόπιμο) που αποθηκεύονται για κάθε οντότητα. Οι ερευνητές έχουν διαπιστώσει ότι αυτός ο περιορισμός, συνήθως έχει μικρή επιρροή στην ακρίβεια των προβλέψεων.

3. **Πρόβλημα γραμμικής άλγεβρας** Το πρόβλημα της δημιουργίας συστάσεων αντιμετωπίζεται και πραγματοποιώντας πράξεις γραμμικής άλγεβρας στον πίνακα προτιμήσεων χρηστών-αντικειμένων (preference table).

Στα θετικά των model-based αλγορίθμων, τώρα:

i. **Scalability (επεκτασιμότητα)** Τα μοντέλα που κατασκευάζονται έχουν μέγεθος πολύ μικρότερο του data-set, έτσι ώστε και σε πολύ μεγάλα data-sets το μοντέλο καταλήγει να είναι αρκετά μικρό για να χρησιμοποιείται αποτελεσματικά. Αυτό το χαρακτηριστικό δίνει στο συνολικό σύστημα μεγάλη επεκτασιμότητα.

ii. **Prediction speed (ταχύτητα προβλέψεων)** Τα model-based συστήματα τείνουν να είναι και γρηγορότερα, τουλάχιστον ως προς τα memory-based συστήματα μιας και ο χρόνος που χρειάζεται για να ερωτηθεί το μοντέλο είναι πολύ μικρότερος από τον αντίστοιχο χρόνο που χρειάζεται να ερωτηθεί όλο το data-set.

iii. **Avoidance of overfitting (αποφυγή του overfitting)** Αν το data-set πάνω στο οποίο χτίζουμε το μοντέλο μας είναι αρκετά αντιπροσωπευτικό των δεδομένων του πραγματικού κόσμου, είναι ευκολότερο να αποφευχθεί το overfitting με την χρήση model-based συστημάτων.

Στα αρνητικά θα αναφέρουμε δύο κυρίως σημεία:

i. **Δυσκολία εισαγωγής νέων δεδομένων (Inflexibility)** Επειδή συνήθως η κατασκευή ενός μοντέλου χαρακτηρίζεται από μεγάλες χρονικές και και υπολογιστικές απαιτήσεις, είναι συνήθως πιο δύσκολο να εισάγουμε νέα δεδομένα στα model-based συστήματα σε σχέση με τις υπόλοιπες προσεγγίσεις. Έτσι, για παράδειγμα, στην δική μας υλοποίηση αυτό είναι βασικό πρόβλημα την στιγμή που η εισαγωγή νέων προορισμών είναι απαιτούμενο και δεν είναι επιθυμητός ο περιορισμός των ταξιδιωτικών γραφείων ως προς την επιλογή των προορισμών για τους οποίους πρόκειται να εκδόσουν ένα νέο ταξιδιωτικό πακέτο.

ii. **Ποιότητα των προβλέψεων (Quality of predictions)** Επειδή δεν χρησιμοποιούμε όλη την πληροφορία που είναι διαθέσιμη στο σύστημα (όλο το data-set) , είναι πιθανό να μην παράγονται προβλέψεις τόσο ακριβείς όσο στα memory-based συστήματα. Παρ'ολα αυτά, η ποιότητα των προβλέψεων εξαρτάται από τον τρόπο κατασκευής του μοντέλου και επομένως εξαρτάται από την προσεκτική μελέτη των δεδομένων και των επιθυμητών συστάσεων προκειμένου να επιτευχθούν ποιοτικά αποτελέσματα σε επίπεδο συστάσεων.

2.10.2 Βιβλιοθήκες Apache Mahout

Η υλοποίηση έγινε με χρήση των βιβλιοθηκών Apache Mahout, στην τελευταία τους έκδοση μέχρι την ημερομηνία συγγραφής της συγκεκριμένης διπλωματικής εργασίας που είναι η 0.9. Σε αυτή την έκδοση έχουν υλοποιηθεί ένας μεγάλος αριθμός αλγορίθμων μηχανικής μάθησης. Μερικές απ'τις κατηγορίες των αλγορίθμων αυτών είναι αυτές που σχετίζονται με Collaborative Filtering, Classification, Clustering κ.ά. Οι βιβλιοθήκες αυτές είναι βιβλιοθήκες ελεύθερου λογισμικού ανοιχτού κώδικα, είναι γραμμένες σε Java και εξυπηρετούν τις ανάγκες μας για την Collaborative Filtering προσέγγιση που επιλέξαμε για το σύστημά μας. Αξίζει να αναφέρουμε το γεγονός ότι οι συγκεκριμένες βιβλιοθήκες δεν προσφέρουν εργαλεία για content-based ανάλυση. Αυτό δεν συμβαίνει λόγω μη ενδιαφέροντος από την πλευρά των developers. Αντιθέτως, όπως αναφέραμε και παραπάνω, η content-based προσέγγιση προβλημάτων προσφέρει λύσεις και λειτουργεί ανά περιπτώσεις πολύ ικανονποιητικά. Όμως, είναι domain-specific, με αποτέλεσμα να είναι δύσκολη η ανάπτυξη ενός γενικού framework που να μπορεί να καλύπτει τις ανάγκες μεγάλης κλάσης προβλημάτων, και να είναι παράλληλα αποδοτικά υλοποιημένο.

2.11 Εξυπηρετητής Ιστού (Web Server)

Ο εξυπηρετητής Ιστού είναι ένα υπολογιστικό σύστημα του οποίου βασική λειτουργία είναι να επεξεργάζεται αιτήσεις του Πρωτόκολλου Μεταφοράς Υπερκειμένου (HTTP Requests). Γενικότερα, ένας εξυπηρετητής Ιστού μπορεί να φιλοξενεί σελίδες, διαδικτυακά παιχνίδια, αποθήκευση δεδομένων, να χειρίζεται αιτήσεις ηλεκτρονικού ταχυδρομείου κτλ. Ένα επιπλέον χαρακτηριστικό που μας ενδιαφέρει στους web servers είναι η δυνατότητα υποστήριξης γλωσσών σεναρίου εκτελούμενων στον εξυπηρετητή (server-side scripting), από γλώσσες προγραμματισμού σεναρίων όπως η ASP και η PHP. Αυτή η υποστήριξη δίνει την δυνατότητα παραγωγής ιστοσελίδων δυναμικά ανάλογα με τις επιλογές του χρήστη καθώς και να ενημερώνει την βάση δεδομένων. Αυτό σημαίνει ότι η συμπεριφορά των web servers μπορεί να προγραμματιστεί σε ξεχωριστά αρχεία σεναρίων χωρίς να αλλάζει το λογισμικό του εξυπηρετητή.

2.12 Java Εξυπηρετητής Ιστού (Servlet Container)

Ένας servlet container όπως είναι ο Apache Tomcat Server, είναι στην ουσία ένας εξυπηρετητής Ιστού με εκτεταμένες δυνατότητες. Πέρα απ'την δυνατότητα χειρισμού των HTTP Requests όπως και ένας τυπικός web server, υποστηρίζει τον χειρισμό JSP σελίδων (Java Server Pages) καθώς και την χρήση Java Servlets, απ'όπου προκύπτει και ο χαρακτηρισμός του εξυπηρετητή ως Servlet Container. Οι σελίδες JSP έχουν τον ίδιο χαρακτήρα και ρόλο με τις σελίδες που προγραμματίζονται σε PHP και ASP και το Servlet είναι μια κλάση της γλώσσας προγραμματισμού Java που μπορεί να υποστηρίζει τον χειρισμό διαφόρων τύπων αιτήσεων και κυρίως HTTP Requests, επεκτείνοντας τις εφαρμογές που φιλοξενούνται από τους Web Servers ώστε να θεωρούνται ως Java applets που εκτελούνται στον εξυπηρετητή και όχι στο λογισμικό πλοήγησης (web browser). Και αυτό διότι τα Java applets είναι προγράμματα γραμμένα σε γλώσσα Java το οποία είναι πρσβάσιμα στους χρήστες με την μορφή bytecode κατά την εκκίνησή τους από έναν web browser. Μετά την έναρξή τους απ'τον χρήστη εκτελούνται σε ένα JVM σε ξεχωριστή διεργασία από αυτή του browser. Γίνεται επομένως αντιληπτό ότι τα servlets δίνουν την δυνατότητα μιας server-side εκτέλεσης μιας εφαρμογής χωρίς επιπλέον επιβάρυνση των πόρων του υπολογιστικού συστήματος που χρησιμοποιεί ο χρήστης, ενώ παράλληλα είναι προσβάσιμα από διαφορετικά μηχανήματα χωρίς να απαιτείται κάποια επιπλέον εγκατάσταση λογισμικού στην πλευρά του χρήστη για την εξυπηρέτηση των αιτημάτων του.

2.13 Γλώσσες προγραμματισμού σεναρίων (Scripting Language)

Μια γλώσσα σεναρίων (Scripting Language, script language) είναι μια γλώσσα προγραμματισμού που επιτρέπει τον έλεγχο μιας ή περισσότερων εφαρμογών. Τα “σενάρια” (“scripts”) είναι διακριτά από τον βασικό κώδικα της εφαρμογής, καθώς γράφονται συνήθως σε διαφορετική γλώσσα και συχνά δημιουργούνται ή τροποποιούνται από τον τελικό χρήστη. Τα σενάρια συνήθως διερμηνεύονται από τον πηγαίο κώδικα ή τον bytecode ενώ η εφαρμογή συνήθως έχει ήδη πρώτα μεταγλωττιστεί σε κώδικα μηχανής. Η δημοτικότητά τους στις διαδικτυακές εφαρμογές γίνεται κατανοητή απ'το γεγονός ότι προσδίδουν δυναμικό χαρακτήρα στις Ιστοσελίδες χωρίς να επηρεάζουν το λογισμικό του εξυπηρετητή ιστού (web server) όπως αναφέραμε και παραπάνω. Στο σημείο αυτό αξίζει να σημειώσουμε ότι οι scripting languages διακρίνονται σε δύο βασικές κατηγορίες ανάλογα με το που εκτελούνται κατά την κλήση των αρχείων που τις περιέχουν. Αυτές είναι οι server-side scripting languages όπως για παράδειγμα η PHP και η ASP και οι client-side scripting languages όπως η Javascript.

2.14 Σύστημα Διαχείρισης Βάσης Δεδομένων

Ένα Σύστημα Διαχείρισης Βάσης Δεδομένων (Database Management System -DBMS), είναι ένα σύνολο μονάδων λογισμικού (προγραμμάτων) που παρέχει την δυνατότητα αποθήκευσης, επεξεργασίας, και ανάγνωσης πληροφοριών σε μία βάση δεδομένων. Ακόμα, ένα DBMS είναι υπεύθυνο για την ακεραιότητα, την ασφάλεια και την προσβασιμότητα (μέσω

λογαριασμών χρηστών) των δεδομένων, καθώς και για τις λειτουργίες επαναφοράς των δεδομένων σε προηγούμενη κατάσταση σε περίπτωση παραβίασης κάποιων απ'τις παραπάνω απαιτήσεις για ένα τέτοιο σύστημα. Ανάμεσα στα πιό διάσημα DBMS είναι η MySQL, PostgreSQL, Microsoft SQL Server, Oracle, SAP και το IBM DB2.

ΚΕΦΑΛΑΙΟ 3

Παρουσίαση Συστήματος

Σε αυτό το κεφάλαιο γίνεται μια παρουσίαση του συστήματος και των λειτουργιών που προσφέρει όπως αυτά φαίνονται στον χρήστη. Στο κεφάλαιο 1 έγινε μια περιγραφή της προσέγγισής μας στο πρόβλημα της ανάπτυξης ενός τετοιου συστήματος, αποσαφηνίζοντας τις οντότητες που υπάρχουν σε αυτό. Αυτή η ανάλυση θα μας χρησιμεύσει τόσο στην περιγραφή των λειτουργιών που προσφέρει το σύστημα όσο και στην περιγραφή του αυτοματοποιημένου συστήματος συστάσεων που έχουμε αναπτύξει. Ο ουσιαστικός σκοπός του συστήματος είναι η παροχή λειτουργιών για αγορά και αξιολόγηση ταξιδιωτικών πακέτων απ'την πλευρά των χρηστών καθώς και η λειτουργία αυτοματοποιημένης σύστασης ταξιδιωτικών πακέτων απ'την πλευρά του συστήματος με όσο το δυνατόν ποιοτικότερα αποτελέσματα σύστασης.

3.1 Συνοπτική Περιγραφή Συστήματος

Το σύστημα που αναπτύχθηκε στο πλαίσιο αυτής της εργασίας είναι ένα σύστημα πώλησης ταξιδιωτικών πακέτων. Σε αυτό το σύστημα, οι “Χρήστες” μπορούν να αναζητούν μελλοντικούς ταξιδιωτικούς προορισμούς σε μορφή ταξιδιωτικών πακέτων, να αγοράζουν και να αξιολογούν ταξιδιωτικά πακέτα. Τα “Ταξιδιωτικά Γραφεία” έχουν την δυνατότητα καταχώρησης και ενημέρωσης ταξιδιωτικών προορισμών. Και το “Συστημα Συστάσεων” έχει πρόσβαση στα δεδομένα του δικτυακού μας τόπου και εκτελεί συστάσεις στους Χρήστες. Το γενικό σενάριο που μας ενδιαφέρει στο σύστημά μας είναι το εξής: Ένας Χρήστης έχει το δικαίωμα να επιλέξει ένα από τα πολλά ταξιδιωτικά πακέτα που προσφέρονται απ'το διαδικτυακό σύστημα μέσω των Ταξιδιωτικών Γραφείων. Η επιλογή αυτή μπορεί να γίνει είτε με απλή αναζήτηση και προβολή των στοιχείων του πακέτου, είτε μέσω σύγκρισης με άλλα πακέτα με χρήση εργαλείου σύγκρισης που του προσφέρουμε. Μόλις ο Χρήστης καταλήξει σε μια επιλογή, μπορεί να το αγοράσει. Με την αγορά και αφού -θεωρητικά- ο Χρήστης έχει ταξιδέψει στους επιλεγμένους “Προορισμούς” του ταξιδιωτικού πακέτου, το σύστημα απαιτεί κάποια αξιολόγηση των Προορισμών και του Ταξιδιωτικού Γραφείου που συνέθεταν το συγκεκριμένο πακέτο. Το Συστημα Συστάσεων λαμβάνει αυτήν την πληροφορία υπ'όψη και αναλύει τα στοιχεία των Χρηστών καταλήγοντας σε κάποιες συστάσεις ταξιδιωτικών πακέτων που είναι πολύ πιθανό να ικανοποιούν τον εκάστοτε Χρήστη.

Οι λέξεις Χρήστης, Ταξιδιωτικό Γραφείο, Προορισμός και Σύστημα Συστάσεων έχουν συγκεκριμένο νόημα στα πλαίσια της περιγραφής του συστήματος. Με αυτό το νόημα χρησιμοποιούνται στο σύνολο της εργασίας και γι'αυτό τον λόγο αναγράφονται με κεφαλαίο αρχικό γράμμα.

3.2 Χρήστες του Συστήματος

Οι Χρήστες του συστήματος ή διαφορετικά οι επισκέπτες της ιστοσελίδας που υλοποιήσαμε χωρίζονται σε δύο κατηγορίες. Στους ταξιδιώτες και στους συνεργάτες. Οι ταξιδιώτες αποτελούν ουσιαστικά τους Χρήστες που είναι εγγεγραμμένοι στην βάση δεδομένων της σελίδας μας για αγορά κάποιου ταξιδιωτικού πακέτου ενώ οι συνεργάτες αποτελούν τους Χρήστες που σχετίζονται με κάποιο Ταξιδιωτικό Γραφείο και έχουν δικαιώματα χρήσης του λογαριασμού που είναι εγγεγραμμένος στο σύστημά μας.

Κατηγορία 1 – Ταξιδιώτες Οι συγκεκριμένοι Χρήστες αποτελούν και την βασική οντότητα Χρηστών του συστήματός μας. Όπως αναλύσαμε και στο Κεφάλαιο 1 της συγκεκριμένης εργασίας, αποτελούν την ενεργή οντότητα του συστήματος οπότε και σε αυτούς παρέχονται λειτουργίες οι οποίες σε πλήθος και σε ποιότητα προσεγγίζουν πραγματικά εμπορικά συστήματα. Θέλουμε να έχουν την δυνατότητα να βρίσκουν μη-αυτοματοποιημένα την βέλτιστη επιλογή τους σε επίπεδο ταξιδιωτικού πακέτου ανάλογα με τις απαιτήσεις τους έτσι ώστε η αξιολόγηση που θα επιστρέψουν στο σύστημα να είναι η ποιοτικότερη δυνατή. Έτσι, η αυτοματοποιημένη σύσταση αποκτά ουσιαστικότερο νόημα.

Για να ανήκει ένας Χρήστης σε αυτή την κατηγορία θα πρέπει να κάνει εγγραφή στο σύστημα. Με την επίσκεψή του στην ιστοσελίδα, μπορεί να κάνει τις παρακάτω λειτουργίες:

1. Είσοδο (log-in) στο σύστημα.
2. Προβολή και επεξεργασία/ενημέρωση των στοιχείων του προφίλ του.
3. Αλλαγή στοιχείων εισόδου στο σύστημα.
4. Αναζήτηση ταξιδιωτικών πακέτων βάσει της τοποθεσίας του Χρήστη.
5. Αναζήτηση ταξιδιωτικών πακέτων βάσει του προορισμού που επιθυμεί να επισκεφτεί.
6. Αναζήτηση ταξιδιωτικών πακέτων βάσει ημερομηνιών αναχώρισης και επιστροφής.
7. Κατάταξη των αποτελεσμάτων αναζήτησης κατά αύξουσα σειρά τιμής, ημερομηνίας αναχώρισης (μια ημερομηνία θεωρείται μικρότερη μιας άλλης όταν προηγείται αυτής χρονικά), χρονικής διάρκειας πακέτου και ημερομηνία έκδοσης του πακέτου.
8. Αναλυτική προβολή των στοιχείων του πακέτου.
9. Αποθήκευση ενός ταξιδιωτικού πακέτου σε μια λίστα που αποτελείται από προσωρινές προτιμήσεις του Χρήστη κατά την αναζήτηση πακέτων στο συνολικό περιεχόμενο του συστήματος.
10. Έναρξη διαδικασίας σύγκρισης των πακέτων μέσω της προσωρινής λίστας που επιθυμητών προορισμών του Χρήστη ή εκκαθάριση αυτής.
11. Σύγκριση πακέτων.
12. Αποθήκευση πακέτου σε καλάθι αγορών. Ενημέρωση καλαθιού ή άδειασμά του.

13. Αγορά πακέτου.

14. Προβολή των αξιολογήσεων/βαθμολογήσεων που μπορεί να κάνει.

15. Αξιολόγηση προορισμών και ταξιδιωτικών γραφείων που συνθέτουν το πακέτο.

16. Αποσύνδεση από το σύστημα (log-out).

Οι λειτουργίες 4 - 7, όπως αναφέρονται παραπάνω, λειτουργούν συνδυαστικά με την έννοια ότι ένας χρήστης μπορεί να συγκεκριμενοποιήσει τα κριτήρια αναζήτησής του επιλέγοντας πολλούς τρόπους αναζήτησης ή να την γενικεύσει, επιλέγοντας μόνο ένα, ανάλογα με την επιθυμία του.

Κατηγορία 2 – Συνεργάτες Η κατηγορία των συνεργατών αφορά τους Χρήστες οι οποίοι είναι εγγεγραμμένοι στο σύστημα ως ιδιοκτήτες ή εξουσιοδοτημένοι χειριστές ενός Ταξιδιωτικού Γραφείου. Επιλέγουμε το όνομα “συνεργάτες” για τον προφανή λόγο του ότι γεμίζουν το σύστημα με ταξιδιωτικά πακέτα και τηρούν τους όρους της ιστοσελίδας για πώληση των προϊόντων/υπηρεσιών τους προς τους καταναλωτές/ταξιδιώτες, αναφερόμενοι στο -πάντα- υποθετικό εμπορικό σενάριο με το οποίο ασχολείται η παρούσα εργασία.

Για να μπορεί ένας Χρήστης να γίνει συνεργάτης, αρκεί να επισκεφτεί τον σχετικό σύνδεσμο (Link) στην αρχική σελίδα του συστήματός μας και από εκεί έχει την επιλογή είτε να συνεχίσει την είσοδό του στο υπόλοιπο σύστημα είτε να δημιουργήσει ένα νέο λογαριασμό και να εγγραφεί στο σύστημα ωστε να έχει πρόσβαση στον χώρο των συνεργατών.

Οι λειτουργίες που μπορεί να κάνει επομένως ένας συνεργάτης είναι:

1. Εγγραφή στο σύστημα (sign-up).

2. Είσοδο στο σύστημα (log-in).

3. Προβολή όλων των ταξιδιωτικών πακέτων που έχει καταχωρήσει στην βάση δεδομένων του συστήματος.

4. Ενημέρωση των ταξιδιωτικών του πακέτων.

5. Διαγραφή ταξιδιωτικών πακέτων.

6. Έξοδο απ'το σύστημα.

Ο διαχωρισμός αυτός τον Χρηστών πέρα από επίπεδο ρόλων και λειτουργιών στο σύστημα, αποκτά νόημα και από μία ακόμα παράμετρο που έχει να κάνει με την εξουσιοδότηση ενός Χρήστη για πρόσβαση σε συγκεκριμένη περιοχή της Ιστοσελίδας. Αν και δεν ασχολούμαστε εξαντλητικά με το κομμάτι της ιεραρχίας των χρηστών στο σύστημα και την προσβασιμότητά τους στα διάφορα επίπεδα του συστήματος, δόθηκε ιδιαίτερη προσοχή στον έλεγχο της ταυτότητας του Χρήστη κατά την αίτησή του για μια σελίδα από τον Server. Κάτι τέτοιο επιτυγχάνεται με τον μην επιτρέπεται σε Χρήστη με τον ρόλο του ταξιδιώτη να έχει πρόσβαση σε χώρο που προορίζεται για Χρήστες με τον ρόλο του συνεργάτη.

Κατ'αυτόν τον τρόπο επιγχάνεται η σαφέστερη διαφοροποίηση των ρόλων των διάφορων Χρηστών, αποφεύγεται η πρόκληση τεχνικών σφαλμάτων κατά τις ενημερώσεις και τις αναγνώσεις της βάσης δεδομένων και κατ'επέκταση αυξάνει η ασφάλεια του συστήματός

μας.

3.3 Σύγκριση ταξιδιωτικών πακέτων

Τα ιδιαίτερα χαρακτηριστικά που αναπτύχθηκαν στα πλαίσια της εργασίας είναι το εργαλείο σύγκρισης των ταξιδιωτικών πακέτων και το Σύστημα Συστάσεων των πακέτων προς τους Χρήστες. Στο συγκεκριμένο υποκεφάλαιο, αναλύουμε το πρώτο.

3.3.1 Ανάλυση προβλήματος σύγκρισης

Το πρόβλημα που διαπιστώθηκε κατά την μελέτη κλασικών διαδικτυακών συστημάτων πώλησης ταξιδιωτικών πακέτων και προορισμών είναι ότι οι χρήστες του διαδικτύου δεν έχουν κάποιο εργαλείο που να τους βοηθάει να καταλήξουν στην βέλτιστη λύση μέσα από ένα σύνολο πιθανών επιλογών. Στην έλλειψη κάποιου τέτοιου εργαλείου συμβάλλει και η δομή των διαδικτυακών συστημάτων με την έννοια ότι καθώς ένας χρήστης αναζητά κάποιο προορισμό σε μια συγκεκριμένη σελίδα (έγγραφο υπερκειμένου – web page), η προβολή των αναλυτικών του χαρακτηριστικών γίνεται σε ξεχωριστή σελίδα. Επομένως η δομή των συστημάτων αυτών οδηγεί στην αναζήτηση και αξιολόγηση ενός αντικειμένου (ταξιδιωτικών πακέτων, προορισμών, τρόπων μετακίνσης) τη φορά. Έτσι, ο χρήστης αναγκάζεται να πλοηγείται σε προηγούμενες και επόμενες σελίδες, επανειλημμένα, διότι πρέπει να αλλάξει κάποιο στοιχείο στα φίλτρα αναζήτητησης ή μπορεί να έχει ξεχάσει κάποιο στοιχείο ενός αντικειμένου που εξέταζε κάποια προηγούμενη χρονική στιγμή ή και ακόμα να διαπίστωσε ότι πρέπει να ελέγξει κάποια παράμετρο που δεν έλαβε υπόψη του από ένα αντικείμενο που έχει ηδη εξετάσει αλλά δεν βρίσκεται στην σελίδα στην οποία πλοηγείται εκείνη την στιγμή.

Έτσι, οι χρήστες αναγκάζονται να καταφεύγουν σε έμμεσους τρόπους συστηματοποιημένης σύγκρισης. Ένας από αυτούς τους τρόπους είναι το άνοιγμα πολλών καρτελών του προγράμματος πλοήγησης που χρησιμοποιούν έτσι ώστε κάθε καρτέλα να έχει ως περιεχόμενο έναν αντικείμενο προς σύγκριση σε σχέση με το υπόλοιπο σύνολο. Ακόμα και με αυτόν τον τρόπο όμως η σύγκριση δεν είναι αποδοτική. Ένα αντικείμενο μπορεί να είναι καλύτερο από τα υπόλοιπα ως προς ένα χαρακτηριστικό (π.χ. τιμή) αλλά τα υπόλοιπα χαρακτηριστικά του να είναι κατώτερα των υπολοίπων (π.χ. ποιότητα ξενοδοχείου ή τόπου διαμονής) οπότε στο σύνολό του να αποτελεί μια κακή επιλογή την στιγμή που ο χρήστης μπορεί να το θεωρεί ως καλή επιλογή βάσει ενός μόνο σημαντικού χαρακτηριστικού.

Η δική μας προσέγγιση για την λύση του προβλήματος της σύγκρισης είναι αποδοτική κυρίως για τον λόγο του ότι προσφέρει εργαλείο για σύγκριση στον χρήστη με τρόπο γραφικό, κάνοντας ξεκάθαρες τις διαφορές μεταξύ των - προς σύγκριση – αντικειμένων.

3.3.2 Εργαλείο σύγκρισης

Στην λύση που πρσφέρουμε στην υλοποίησή μας, η σύγκριση γίνεται γρήγορα, εύκολα και αποδοτικά. Για να γίνουν κατανοητά τα θετικά της υλοποίησής του εργαλείου αυτού, περιγράφουμε την λειτουργικότητά του μέσω ενός σεναρίου χρήσης. Ξεκινάμε την υπόθεσή

μας θεωρώντας ότι ο Χρήστης μόλις έχει μπει στο σύστημα, βρίσκεται στην σελίδα όπου όλα τα πακέτα παρουσιάζονται σε μορφή λίστας, συνοδευόμενα από τα διάφορα φίλτρα αναζήτησης που αναφέραμε παραπάνω σχετικά με τους προορισμούς, τις ημερομηνίες αναχώρισης – άφιξης και των τιμών, δεν έχει βρει κάτι που τον ενδιαφέρει και μόλις ξεκινάει την αναζήτησή του.

Στην σελίδα αναζήτησης των ταξιδιωτικών πακέτων, υπάρχει μια λίστα στην δεξιά πλευρά της οθόνης του Χρήστη η οποία αρχικά είναι άδεια. Αυτή η λίστα περιέχει το σύνολο των ταξιδιωτικών πακέτων που επιθυμεί ο Χρήστης να συγκρίνει. Ο Χρήστης μόλις συναντήσει κάποιο ταξιδιωτικό πακέτο που να τον ενδιαφέρει, έχει την δυνατότητα μέσω ενός κουμπιού με την ετικέτα “+ add destination” να πρσθέσει αυτό το πακέτο στην λίστα και να συνεχίσει την αναζήτησή του στο υπόλοιπο περιεχόμενο του συστήματος. Φυσικά, ο Χρήστης μπορεί να προσθέσει αμέσως το πακέτο στην λίστα με το που το συναντήσει στην συνολική λίστα μιας και του παρέχονται βασικές πληροφορίες οι οποίες είναι αρκετές ώστε να του δώσουν μια πρώτη εικόνα για το πόσο μπορεί να τον ενδιαφέρει το συγκεκριμένο πακέτο. Απ'την άλλη όμως μπορεί να νιώσει την ανάγκη να δεί λεπτομερώς το περιεχόμενο του πακέτου επειδή μπορεί να τον ενδιαφέρει κάποια λεπτομέρεια η οποία δεν παρέχεται στην συγκεκριμένη σελίδα του συστήματος (π.χ. περιγραφή των διαδρομών). Μπορεί, λοιπόν, να επιλέξει την προβολή των αναλυτικών στοιχείων του ταξιδιωτικού πακέτου μέσω κατάλληλων συνδέσμων που του παρέχονται. Αφού ελέγξει το κατά πόσο μπορεί να τον ενδιαφέρει το πακέτο, μπορεί να το προσθέσει ή όχι στην τρέχουσα λίστα που έχει σχηματίσει μέχρι τώρα και να συνεχίσει την αναζήτησή του από εκεί που έχει μείνει. Το σύστημα διασφαλίζει ότι αυτή η λίστα θα διατηρείται στο πρόγραμμα πλοήγησης (browser) μέχρις ότου ο Χρήστης να επιλέξει την εκκαθάρισή της ή όχι. Παρατηρούμε, λοιπόν, ότι ο Χρήστης έχει την ελευθερία κίνησης στο σύστημα διατηρώντας τα αντικείμενα που τον ενδιαφέρουν χωρίς αυτό να επηρεάζει την ποιότητα των αναζητήσεών του μιας και εξασφαλίζεται ότι πακέτα που έχει προσθέσει στην λίστα δεν θα ξαναβρεθούν στο δρόμο των νέων αναζητήσεών του. Επίσης, η υλοποίηση γίνεται με client-side scripting εκμεταλλευόμενοι το local storage του browser, με αποτέλεσμα να μην επηρεάζεται η εμπειρία του χρήστη κατά την πλοήγησή του στο υπόλοιπο σύστημα, όσα πακέτα και αν έχει προσθέσει στην λίστα του.

Αφού καταλήξει σε μια συγκεκριμένη λίστα, ο Χρήστης με το πάτημα ενός κουμπιού στο κάτω μέρος της λίστας (με ετικέτα “checkIt”) μεταφέρεται στην σελίδα του συστήματος όπου γίνεται η σύγκριση. Εκεί, το σύστημα αρχικά διαλέγει ένα πακέτο ως βάση σύγκρισης για τα άλλα πακέτα. Οι τιμές οι οποίες συγκρίνονται είναι η τιμή του πακέτου, η διάρκειά του και η απόσταση των προορισμών από την παρούσα θέση του Χρήστη. Το πακέτο το οποίο αποτελεί την βάση σύγκρισης εμφανίζεται σε ένα μεγάλο πλαίσιο, για να διαχωρίζεται από τα υπόλοιπα πακέτα, και οι τιμές του έχουν δίπλα τους μια ένδειξη σε μαύρο χρώμα που δείχνει ξεκάθαρα στον χρήστη ότι οι τιμές του αποτελούν την βάση σύγκρισης με τις υπόλοιπες τιμές. Τα υπόλοιπα πακέτα έχουν στις τιμές τους ενδείξεις και σύμβολα χρωματισμένα έτσι ώστε να γίνεται ξεκάθαρο για το αν κάθε τιμή του είναι καλύτερη, χειρότερη ή ίδια με την αντίστοιχη του πακέτου με το οποίο συγκρίνονται. Έτσι, αν η τιμή του ταξιδιωτικού πακέτου είναι μεγαλύτερη από αυτή με την οποία συγκρίνεται, τότε δίπλα του εμφανίζεται ένα βέλος με ένδειξη προς τα πάνω και με κόκκινο χρώμα δηλώνοντας ακριβώς ότι αυτή η τιμή είναι πιο ακριβή από αυτή που συγκρίνει ο Χρήστης. Αν η τιμή είναι μικρότερη ή ίση τότε το χρώμα είναι πράσινο και το σύμβολο γίνεται βέλος προς τα κάτω ή η μαθηματική ισότητα,αντίστοιχα. Στην διάρκεια του πακέτου η λογική είναι αντίστροφη. Αν η διάρκεια ενός πακέτου είναι μεγαλύτερη από αυτή της βάσης σύγκρισης τότε το βέλος είναι πράσινο και η φορά του προς τα πάνω, ενώ αν είναι ίση, το αντίστοιχο σύμβολο εμφανίζεται πάλι με πράσινο χρώμα. Αν όμως είναι μικρότερη, τότε το βέλος έχει φορά προς τα κάτω και το χρώμα είναι κόκκινο. Αυτό προκύπτει φυσικά απ'το γεγονός ότι μεγαλύτερη διάρκεια είναι συμφέρουσα ως προς την ίδια τιμή μεταξύ δύο διαφορετικών πακέτων. Επομένως, μεγαλύτερη διάρκεια είναι μια θετική

διαφορά ως προς την βάση σύγκρισης και το σύστημα δηλώνει στον Χρήστη ότι είναι μια συμφέρουσα επιλογή μιας και θα διανύσει μεγαλύτερη απόσταση. Τέλος, η απόσταση ακολουθεί την λογική της διάρκειας ενός ταξιδιωτικού πακέτου. Μεγαλύτερη απόσταση σημαίνει καλύτερη εκμετάλλευση του κόστους της ταξιδιωτικής επιλογής ενώ μικρότερη απόσταση σημαίνει και μικρότερη ταξιδιωτική εμπειρία. Η ορθότητα των παραπάνω κριτηρίων προκύπτει άμεσα απ'το γεγονός ότι ένας Χρήστης συνήθως επιλέγει τους προορισμούς του βάσει συγκεκριμένου εύρους τιμών οπότε και οι υπόλοιπες παράμετροι ενός ταξιδιωτικού πακέτου σχετίζονται με το κατά πόσο μπορεί κάποιος Χρήστης να αυξήσει την ταξιδιωτική του εμπειρία με το ίδιο κόστος. Έτσι, αν κάποιος χρήστης μπορεί με το ίδιο κόστος να επιλέξει ένα ταξιδιωτικό πακέτο με το οποίο θα ταξιδέψει πιο μακρια και για μεγαλύτερη διάρκεια, σίγουρα αποτελεί μια ελκυστική πρόταση ανάμεσα στις πιθανές επιλογές του. Φυσικά, ακόμα και αν θέλει να συγκρίνει πακέτα με περίπου την ίδια διάρκεια, η λογική της σύγκρισης είναι ίδια. Δηλαδή, πόσο φθηνά και μακριά μπορεί να πάει για ένα συγκεκριμένο χρονικό διάστημα που του επιτρέπει το πρόγραμμά του.

Όλη αυτή η λογική της σύγκρισης γίνεται εύκολα αντιληπτή απ'την χρήση των χρωμάτων και των συμβόλων καθώς με μια ματιά ο χρηστης μπορεί εύκολα να απορρίψει επιλογές και να επικέντρώσει την προσοχή του για τελική επιλογή, αν όχι σε δύο πακέτα τότε σε ένα σαφέστατα μικρότερο σύνολο απ'το αρχικά επιλεγμένο. Και αυτό συμβαίνει διότι απλούστατα ένα πακέτο με περισσότερα κόκκινα στοιχεία από καποιο άλλο, σίγουρα θα αποτελεί χειρότερη επιλογή για τον Χρήστη.

Αν, τώρα, ο Χρήστης θέλει να αλλάξει τους συσχετισμούς των συγκρίσεων και να επιλέξει ένα διαφορετικό πακέτο ως βάση σύγκρισης σε σχέση με αυτό που χρησιμοποιείται ήδη, τότε με χρήση του κουμπιού με ετικέτα “compare this” που υπάρχει διαθέσιμο σε κάθε πακέτο, μπορεί εύκολα να διαπιστώσει αν όντως η χρωματική και αριθμητική διαίσθηση που χρησιμοποιείται, επιβεβαιώνει και τον χαρακτήρα του βέλτιστου ενός πακέτου. Φυσικά, απ'την συγκεκριμένη σελίδα ο Χρήστης μπορεί να δεί και το λεπτομερές περιεχόμενο του συγκεκριμένου πακέτου ή και να το αγοράσει με την χρήση κατάλληλων κουμπιών που παρέχονται (με τις αντίστοιχες ετικέτες “view” και “buy now” που παρέχονται. Σε περίπτωση που μετά την αναλυτική προβολή των στοιχείων του πακέτου ή την αγορά του, ο Χρήστης επιθυμεί να συνεχίσει την αναζήτηση ή την σύγκριση τότε μπορεί να το κάνει εύκολα χωρίς να επηρεάζεται για άλλη μια φορά ούτε η λίστα του ούτε η εμπειρία του κατα την πλοήγησή του στις διάφορες σελίδες.

3.4 Περιεχόμενο συστήματος

Όπως αναφέρθηκε και προηγουμένως, σκοπός του συστήματος είναι η διαχείριση ταξιδιωτικών πακέτων που αποτελούνται από τους Προορισμούς και λοιπά στοιχεία. Τα ταξιδιωτικά πακέτα δημιουργούνται από τα Ταξιδιωτικά Γραφεία, προβάλλονται και αγοράζονται από τον Χρήστη και προτείνονται από το Σύστημα Συστάσεων. Ξεκινάμε με την ανάλυση του πακέτου.

3.4.1 Χαρακτηριστικά ταξιδιωτικού πακέτου

Τα στοιχεία που χαρακτηρίζουν ένα ταξιδιωτικό πακέτο μπορούν να χωριστούν σε δύο κατηγορίες:

- **Βασικά** Είναι τα στοιχεία που χαρακτηρίζουν το ταξιδιωτικό πακέτο ως οντότητα και βάσει αυτών των στοιχείων το Σύστημα Συστάσεων αναλύει τα δεδομένα και επιστρέφει προτάσεις στον Χρήστη, και βάσει αυτών των στοιχείων ο Χρήστης αναζητά τις ταξιδιωτικές του επιλογές.

- **Δευτερεύοντα** Είναι τα στοιχεία που δεν έχουν κάποια ιδιαίτερη πληροφορία για το σύστημα στο σύνολό του αλλά έχουν νόημα για τον χρήστη και την ολοκληρωμένη εικόνα της συγκεκριμένης οντότητας.

Βασικά χαρακτηριστικά πακέτου

Προορισμοί Οι Προορισμοί οι οποίοι περιέχονται σε κάθε ταξιδιωτικό πακέτο. Οι Προορισμοί είναι διαθέσιμοι στα Ταξιδιωτικά Γραφεία και επιλέγονται από αυτά. Κάθε πακέτο μπορεί να έχει απεριόριστους προορισμούς σύμφωνα με την υλοποίηση του συστήματός μας. Παρ'όλα αυτά, στην σύνθεση των δεδομένων που κάναμε, επιτρέψαμε μέχρι τρεις Προορισμούς σε κάθε ταξιδιωτικό πακέτο με την έννοια ότι ο συγκεκριμένος αριθμός είναι ένα λογικό άνω όριο, καθώς και στον πραγματικό κόσμο της τουριστικής βιομηχανίας παρατηρήσαμε το πολύ τρεις προορισμούς στα μεγαλύτερης διάρκειας ταξιδιωτικά πακέτα που διατίθενται.

Ταξιδιωτικό Γραφείο Τα Ταξιδιωτικά Γραφεία παίζουν σημαντικό ρόλο στον χαρακτηρισμό του πακέτου. Ένα πακέτο συνδέεται με ένα Ταξιδιωτικό Γραφείο, κληρονομώντας την τοποθεσία του και την ποιότητα που χαρακτηρίζει το ίδιο το πακέτο. Η τοποθεσία κληρονομείται με την έννοια ότι κάθε Ταξιδιωτικό Γραφείο όπως θα δούμε στην συνέχεια, έχει μια συγκεκριμένη τοποθεσία η οποία αφορά και την περιοχή εξυπηρέτησής του και κατ'επέκταση την αφετηρία κάθε ταξιδιωτικού πακέτου. Έτσι, όταν ένας Χρήστης αναζητά ένα πακέτο, προσδιορίζει την τοποθεσία του και το σύστημα γνωρίζει ποια πακέτα είναι κατάλληλα για προβολή. Όσον αφορά την ποιότητα, αυτή κληρονομείται από την βαθμολογία που έχει λάβει το ταξιδιωτικό γραφείο για την μέχρι τώρα προσφορά υπηρεσιών του στους καταναλωτές και γίνεται διαθέσιμη στην σύγκριση των ταξιδιωτικών πακέτων με το εργαλείο που παρέχεται.

Ημερομηνία αναχώρησης Η ημερομηνία αναχώρησης αφορά την ημερομηνία έναρξης του ταξιδιωτικού πακέτου και είναι αναγκαία αν και όχι υποχρεωτική για την αναζήτηση ταξιδιωτικών πακέτων με το φίλτρο αναζήτησης που υλοποιήθηκε. Χρησιμοποιείται και για την διάταξη των πακέτων κατά την λειτουργία της αναζήτησης.

Ημερομηνία επιστροφής Η ημερομηνία επιστροφή αφορά την ημερομηνία λήξης του ταξιδιωτικού πακέτου και χρησιμοποιείται επίσης για την αναζήτησης ταξιδιωτικών πακέτων με το φίλτρο αναζήτησης που υλοποιήθηκε.

Τιμή Η τιμή χρησιμοποιείται για την διάταξη των ταξιδιωτικών πακέτων καθώς και ως είσοδος στον αλγόριθμό του Συστήματος Συστάσεων για βελτιστοποίηση των συστάσεων

που πραγματοποιούνται στον Χρήστη, όπως θα δούμε παρακάτω.

Δευτερεύοντα χαρακτηριστικά πακέτου

Περιγραφή Η περιγραφή αφορά το περιεχόμενο του ταξιδιωτικού πακέτου σε όρους προγράμματος και δραστηριοτήτων.

Επιπλέον πληροφορίες (Extras) Το συγκεκριμένο περιεχόμενο αφορά επιπλέον στοιχεία τα οποία είναι αναγκαία για την πλήρη περιγραφή ενός ταξιδιωτικού πακέτου όπως είναι το τί περιέχεται και τί όχι στην τιμή όπως τα εισητήρια σε αξιοθέατα, τα γεύματα που προσφέρονται από τους χώρους διαμονής ή όχι κτλ.

Ξενοδοχείο Αφορά τον ή τους χώρους διαμονής των ταξιδιωτών.

Αεροπορική γραμμή Είναι η πληροφία που αφορά την αεροπορική γραμμή η οποία εξυπηρετεί τις μετακινήσεις των ταξιδιωτών στους προορισμούς του ταξιδιωτικού πακέτου.

Φωτογραφία Μια φωτογραφία η οποία συνοδεύει ένα ταξιδιωτικό πακέτο και είναι χαρακτηριστική των προορισμών που περιέχονται.

3.4.2 Χαρακτηριστικά Ταξιδιωτικού Γραφείου

Το Ταξιδιωτικό Γραφείο έχει τον βασικό ρόλο του να γεμίζει το σύστημα με ταξιδιωτικά πακέτα. Παράλληλα, όπως αναφέραμε και παραπάνω αποτελεί σημείο αναφοράς για την σύγκριση των διάφορων ταξιδιωτικών πακέτων καθώς και για την αναζήτηση κατάλληλων ταξιδιωτικών πακέτων σύμφωνα με την τοποθεσία του εκάστοτε Χρήστη. Τα χαρακτηριστικά του Ταξιδιωτικού Γραφείου είναι:

Ονομασία Είναι η επωνυμία του Ταξιδιωτικού Γραφείου και το χαρακτηρίζει η μοναδικότητά του.

Url Αφορά τον εξωτερικό σύνδεσμο προς τον ιστότοπο του Ταξιδιωτικού Γραφείου, σε περίπτωση ανάγκης για περαιτέρω πληροφόρησης από την πλευρά του Χρήστη.

E-mail Η διεύθυνση ηλεκτρονικού ταχυδρομείου του Ταξιδιωτικού Γραφείου για απ'ευθείας επικοινωνία με αυτό.

Τηλέφωνο Το τηλέφωνο επικοινωνίας με το Ταξιδιωτικό Γραφείο.

Avatar Η φωτογραφία προφίλ του Ταξιδιωτικού Γραφείου, για τοποθέτηση του χαρακτηριστικού του λογότυπου.

Πόλη Η τοποθεσία του ταξιδιωτικού γραφείου, για την οποία θεωρούμε ότι αποτελεί και την περιοχή εξυπηρέτησής του. Επειδή, όπως ακριβώς αναλύσαμε και στο Κεφάλαιο 1, το Ταξιδιωτικό Γραφείο έχει παθητικό ρόλο ώς προς το Σύστημα Συστάσεων, περιορίζουμε το περιεχόμενο της πόλης ως χαρακτηριστικό απλής πληροφόρησης και ολοκληρωμένης λειτουργικότητας της ιστοσελίδας μας ως προς την διαχείριση των ταξιδιωτικών πακέτων. Έτσι, δεν υλοποιήσαμε εξειδικευμένο πεδίο που να αφορά το πεδίο δράσης του Ταξιδιωτικού Γραφείου αλλά περιορίσαμε την έννοια του πεδίου δράσης στην έννοια της έδρας του Ταξιδιωτικού Γραφείου και συσχετίζουμε κάθε Ταξιδιωτικό Γραφείο με μία μόνο πόλη. Έτσι, πληρείται η προϋπόθεση δύο πακέτα με ίδιους προορισμούς αλλά από Ταξιδιωτικά Γραφεία που εξυπηρετούν διαφορετικές περιοχές να μην εμφανίζονται ταυτόχρονα σε έναν Χρήστη ο οποίος ενδιαφέρεται για μία μόνο εκ των δύο περιοχών εξυπηρέτησης λόγω -για παράδειγμα- διαμονής του σε αυτήν.

3.5 Λειτουργίες

Εδώ αναφέρουμε πιο συγκεκριμένα τις λειτουργίες ενός Χρήστη του συστήματος. Όπως αναφέραμε παραπάνω, οι Χρήστες του διαδικτυακού μας συστήματος χωρίζονται σε ταξιδιώτες και συνεργάτες. Για τους ταξιδιώτες, είναι λογικό να υπάρχει αντιστοιχία 1 προς 1 αντιστοιχία με στιγμιότυπα της οντότητας Χρήστης. Παρ'όλα αυτά, σύμφωνα με τη εννοιολογική μας προσέγγιση, ένας συνεργάτης μπορεί να μην έχει την ίδια αντιστοίχιση με την οντότητα Ταξιδιωτικό Γραφείο. Έτσι, ένα Ταξιδιωτικό Γραφείο μπορεί να σχετίζεται με πολλούς συνεργάτες οι οποίοι είναι εξουσιοδοτημένοι να διαχειρίζονται τις λειτουργίες του συγκεκριμένου Ταξιδιωτικού Γραφείου. Παρ'όλα αυτά σε επίπεδο υλοποίησης, απλοποιούμε το σενάριο θεωρώντας ότι ένας συνεργάτης αντιστοιχεί σε ένα Ταξιδιωτικό Γραφείο και επομένως – σε επίπεδο λειτουργιών στο σύστημα – οι έννοιες αυτές ταυτίζονται. Επομένως, παρακάτω, περιγράφουμε τις λειτουργίες του Χρήστη αναφερόμενοι στους ταξιδιώτες, και τις λειτουργίες του Ταξιδιωτικού Γραφείου αναφερόμενοι στους συνεργάτες.

3.5.1 Λειτουργίες Χρήστη

Οι λειτουργίες που μπορεί να εκτελέσει ένας χρήστης στα πλαίσια του συστήματός μας είναι:

Εγγραφή (sign-up) Ένας χρήστης του διαδικτύου μπορεί να εγγραφεί στο σύστημα για να αποκτήσει τα δικαιώματα του Χρήστη. Η εγγραφή είναι απλή απαιτώντας μόνο μια διεύθυνση ηλεκτρονικού ταχυδρομείου (email) και έναν κωδικό πρόσβασης, χωρίς να χρειάζεται επιβεβαίωση στοιχείων. Το ηλεκτρονικό του ταχυδρομείο αποτελεί και το όνομα χρήστη στο σύστημα (username) επομένως τα υπόλοιπα στοιχεία όπως το πραγματικό του όνομα, η ηλικία του κτλ είναι αρχικά απροσδιόριστα στην πρώτη του εγγραφή και ο Χρήστης μπορεί να τα

αλλάξει όποτε θέλει.

Είσοδο (log-in) Ο Χρήστης παρέχοντας το username (το οποίο και είναι η διεύθυνση του ηλεκτρονικού ταχυδρομείου) και το password του μπορεί να εισέλθει στο σύστημα και να εκτελέσει τις λειτουργίες που του παρέχονται.

Αναζήτηση ταξιδιωτικών πακέτων Ο Χρήστης μπορεί να αναζητήσει κάποιο ταξιδιωτικό πακέτο βάσει διαφόρων κριτηρίων. Μπορεί να αναζητήσει βάσει της τοποθεσίας του, βάσει ενός από τους προορισμούς που επιθυμεί να επισκεφτεί και βάσει ημερομηνιών αναχώρισης και επιστροφής. Σύμφωνα με την υλοποίησή μας ο Χρήστης μπορεί να κάνει και συνδυαστική αναζήτηση με όλους τους παραπάνω τρόπους και πιθανούς συνδυασμούς, όπως για παράδειγμα να επιλέξει ώς κριτήρια αναζήτησης τον προορισμό του και την ημερομηνία αναχώρισης.

Κατάταξη των αποτελεσμάτων αναζήτησης Η κατάταξη των αποτελεσμάτων αναζήτησης γίνεται κατά αύξουσα σειρά τιμής, ημερομηνία αναχώρισης, χρονικής διάρκειας πακέτου και ημερομηνία έκδοσης του πακέτου. Μια ημερομηνία θεωρείται μικρότερη μιας άλλης όταν προηγείται αυτής χρονικά. Η κατάταξη των αποτελεσμάτων λειτουργεί επίσης συνδυαστικά με τα παραπάνω κριτήρια αναζήτησης που περιγράψαμε.

Προβολή του ταξιδιωτικού πακέτου Ένας Χρήστης μπορεί να προβάλει τα αναλυτικά στοιχεία ενός ταξιδιωτικού πακέτου. Στην σελίδα αναζήτησης των πακέτων, και ύστερα από οποιοδήποτε εφαρμογή κριτηρίου αναζήτησης τα ταξιδιωτικά πακέτα εμφανίζονται σε διάταξη λίστας, προβάλλοντας τα βασικά τους στοιχεία. Με κλικ πάνω στον τίτλο του ταξιδιωτικού πακέτου ή με χρήση του σχετικού συνδέσμου που παρέχεται ο Χρήστης μεταφέρεται σε μια νέα σελίδα που προβάλλονται αναλυτικά τα στοιχεία του ταξιδιωτικού πακέτου συνοδευόμενα από ένα χάρτη ο οποίος δείχνει την διαδρομή του πακέτου.

Σύγκριση ταξιδιωτικών πακέτων Όπως περιγράφεται αναλυτικά στο τμήμα 3.3.2, ένας Χρήστης μπορεί να χρησιμοποιήσει το εργαλείο σύγκρισης ταξιδιωτικών πακέτων που του παρέχεται ως εξής: Αρχικά μπορεί να αποθηκεύσει ένα ταξιδιωτικό πακέτο σε μια λίστα που αποτελείται από προσωρινές προτιμήσεις του Χρήστη κατά την αναζήτηση πακέτων στο συνολικό περιεχόμενο του συστήματος. Στην συνέχεια, μπορεί να κάνει έναρξη διαδικασίας σύγκρισης των πακέτων μέσω της προσωρινής λίστας που επιθυμητών προορισμών του Χρήστη ή εκκαθάριση αυτής.

Χρήση καλαθιού αγορών (shopping cart) Ο Χρήστης μπορεί να επιλέξει ένα πακέτο για μελλοντική αγορά τοποθετώντας το στο καλάθι αγορών του. Πιο συγκεκριμένα, στην σελίδα της προβολής των στοιχείων ενός ταξιδιωτικού πακέτου του παρέχεται σύνδεσμος (button) για την προσθήκη ενός πακέτου στο καλάθι αγορών του. Στην συνέχεια, μπορεί να συνεχίσει την αναζήτηση άλλων ταξιδιωτικών πακέτων. Όταν το επιθυμεί, μπορεί να επισκεφτεί το καλάθι των αγορών του και να επιλέξει την ποσότητα σε επίπεδο ατόμων που επιθυμεί να αγοράσει, και το καλάθι ενημερώνει τον Χρήστη για το συνολικό κόστος των αγορών του σύμφωνα με τις επιλογές του. Φυσικά, μπορεί να αφαιρέσει κάποιο πακέτο αν

τελικά δεν θέλει να το αγοράσει, καθώς και να αδειάσει τελείως το καλάθι αγορών.

Αγορά ταξιδιωτικών πακέτων Στην σελίδα του καλαθιού των αγορών του, ο Χρήστης έχει την δυνατότητα να αγοράσει ένα ταξιδιωτικό πακέτο με τον σύνδεσμου που του παρέχεται. Επιλέγοντας αγορά του πακέτου το σύστημα αντιλαμβάνεται την ενέργεια και ζητά στην συνέχεια αξιολόγηση των Προορισμών και του Ταξιδιωτικού Γραφείου που συνθέτουν το ταξιδιωτικό πακέτο.

Αξιολόγηση προορισμών Όπως περιγράφεται και παραπάνω, μόλις ένας Χρήστης αγοράσει ένα ταξιδιωτικό πακέτο, το σύστημα αντιλαμβάνεται ότι το αγόρασε και υποθετικά ταξίδεψε. Παραλείπουμε το σενάριο στο οποίο ο Χρήστης το αγόρασε αλλά δεν το αξιοποίησε γιατί αφορά την επέκταση του διαδικτυακού μας συστήματος σε εμπορικό σύστημα το οποίο πρέπει να διαχειρίζεται σενάρια του πραγματικού κόσμου, τα οποία κινούνται έξω από τα πλαίσια της συγκεκριμένης εργασίας.

Αποσύνδεση (log-out) Ο χρήστης μπορεί να αποσυνδεθεί από το σύστημα με τον σύνδεσμο που του παρέχεται.

3.5.2 Λειτουργίες Ταξιδιωτικού Γραφείου

Οι λειτουργίες που μπορεί να εκτελέσει ένα Ταξιδιωτικό Γραφείο στα πλαίσια της εργασίας μας είναι:

Εγγραφή (sign-up) Ένας χρήστης που θέλει να δημιουργήσει λογαριασμό εκ μέρους κάποιου Ταξιδιωτικού Γραφείου, μπορεί να το κάνει απ'την βασική σελίδα εγγραφής των συνεργατών του διαδικτυακού μας συστήματος. Θα πρέπει να επισκεφτεί πρώτα την βασική οθόνη του συστήματος (αρχική σελίδα), όπου υπάρχει σύνδεσμος για τον χώρο λειτουργιών που προορίζονται για τα Ταξιδιωτικά Γραφεία. Ο σύνδεσμος αυτός, μεταφέρει έναν χρήστη σε μια σελίδα που δίνει την δυνατότητα είτε σύνδεσης στο σύστημα ως Ταξιδιωτικό Γραφείο είτε δημιουργίας νέου λογαριασμού. Εφόσον ένας χρήστης επιλέξει την δημιουργία ενός νέου λογαριασμού, τότε μεταφέρεται σε μια νέα σελίδα όπου του ζητούνται η εισαγωγή ονόματος του Ταξιδιωτικού Γραφείου, το url, η διεύθυνση ηλεκτρονικού ταχυδρομείου, ένας κωδικός πρόσβασης, ένα τηλέφωνο και η τοποθεσία του. Δεν ζητείται η επιβεβαίωση των στοιχείων του καθώς ακολουθούμε το απλοποιημένο μοντέλο του διαδικτυακού συστήματος που δεν εξαντλεί τα σενάρια χρήσης του σε επίπεδο πραγματικού κόσμου.

Είσοδο (log-in) Με την διαδικασία που περιγράφηκε και στο κομμάτι της δημιουργίας του λογαριασμού παραπάνω, ένα Ταξιδιωτικό Γραφείου έχει την δυνατότητα πρόσβασης στο σύστημα παρέχοντας την διεύθυνση του ηλεκτρονικού του ταχυδρομείου που αντιστοιχεί στο username του, και τον κωδικό πρόσβασης.

Αλλαγή στοιχείων (log-in) Ο Χρήστης, έχει την δυνατότητα αλλαγής των στοιχείων σύνδεσής του στο σύστημα (διεύθυνση ηλεκτρονικού ταχυδρομείου, κωδικός πρόσβασης). Μετά την αλλαγή των στοιχείων αυτών, το σύστημα αποσυνδέει τον Χρήστη και του ζητάει τα νέα του στοιχεία για επνασύνδεση στο σύστημα.

Διαχείριση ταξιδιωτικού πακέτου Το Ταξιδιωτικό Γραφείο έχει την δυνατότητα δημιουργίας ή διαχείρισης ταξιδιωτικών πακέτων, αφού πλοηγηθεί στην σελίδα διαχείρισης των ταξιδιωτικών του πακέτων μέσω του κεντρικού μενού που του παρέχεται. Αφού επιλέξει την σελίδα αυτή, εμφανίζεται ένα υπομενού αριστερά στην οθόνη που αφορά τις ενέργειες που μπορεί να πραγματοποιήσει σε αυτά. Η πρώτη επιλογή αφορά την δημιουργία ενός νέου ταξιδιωτικού πακέτου. Αυτό γίνεται παρέχοντας τα απαραίτητα στοιχεία:

- Προορισμοί, οι οποίοι επιλέγονται από κατάλληλο input που παρέχεται για ευκολη αναζήτηση και χωρίς περιορισμό ως προς τον αριθμό που πρόκειται να καταχωρηθεί
- Ημερομηνία έναρξης του ταξιδιωτικού πακέτου (ημερομηνία αναχώρησης για τον ταξιδιώτη)
- Ημερομηνία λήξης (ημερομηνία επιστροφής για τον ταξιδιώτη)
- Κείμενο περιγραφής του ταξιδιωτικού πακέτου
- Επιπλέον πληροφορίες (extras)
- Αεροπορική εταιρεία
- Ξενοδοχείο
- Τιμή
- Εικόνα που συνοδεύει το πακέτο

Μια δεύτερη επιλογή είναι η επεξεργασία ενός ταξιδιωτικού πακέτου. Από τον σύνδεσμο που του παρέχεται για την συγκεκριμένη λειτουργία, εμφανίζεται μια λίστα των πακέτων που έχει καταχωρήσει στο σύστημα με τα βασικά τους στοιχεία ώστε να είναι εύκολο στο Ταξιδιωτικό Γραφείο να επιλέξει το κατάλληλο πακέτο προς επεξεργασία. Κάθε πακέτο έχει ένα κουμπί (button) το οποίο εμφανίζει τα πεδία εισαγωγής της νέας πληροφορίας από το ταξιδιωτικό γραφείο. Κάθε πεδίο έχει την τιμή που είχε προηγουμένως καταχωρηθεί για το συγκεκριμένο πακέτο κάνοντας εύκολη την διαδικασία της διόρθωσης/επεξεργασίας, ή ακόμα και της εξ'ολοκλήρου αλλαγής των στοιχείων του πακέτου. Το ίδιο συμβαίνει και για την φωτογραφία που συνοδεύει το πακέτο, το οποίο προβάλλεται ως μικρογραφία. Ακόμα, η επεξεργασία κάθε πεδίου έχει υλοποιηθεί με τέτοιο τρόπο ώστε να γίνεται ξεχωριστά από τα υπόλοιπα πεδία, παρέχοντας το κατάλληλο κουμπί αποθήκευσης, διευκολύνοντας την διαδικασία αυτή ακόμα περισσότερο.

Τέλος, παρέχεται και η δυνατότητα διαγραφής κάποιου πακέτου. Στην αντίστοιχη επιλογή που υπάρχει στο μενού, εμφανίζεται μια λίστα των πακέτων κατά τον ίδιο τρόπο που εμφανίζονται και κατά την επιλογή της επεξεργασίας τους. Δίπλα από κάθε πακέτο, υπάρχει κατάλληλο κουμπί που οδηγεί στην διαγραφή του πακέτου και την κατάλληλη ενημέρωση της βάσης δεδομένων.

Προβολή των ταξιδιωτικών πακέτων Το Ταξιδιωτικό Γραφείο έχει την δυνατότητα της

προβολής των καταχωρημένων του ταξιδιωτικών πακέτων, όπως εμφανίζονται στον χρήστη. Αυτό γίνεται με επιλογή της σελίδας διαχείρισης των πακέτων, πριν πλοηγηθεί στις εξειδικευμένες λειτουργίες τους όπως περιγράφηκαν. Η προβολή αυτή γίνεται και μετά την την εισαγωγή ενός νέου πακέτου στο σύστημα, αυτοματοποιημένα με κατάλληλη εμφάνιση μηνύματος για επιτυχημένη καταχώρηση του πακέτου.

Επεξεργασία των στοιχείων λογαριασμού Δίνεται η δυνατότητα επεξεργασίας των στοιχείων του, τα οποία παρείχε αρχικά κατά την εγγραφή του, καθώς το ανέβασμα (uploading) του λογότυπου του Ταξιδιωτικού Γραφείου, είτε προς πρώτη καταχώρηση είτε προς ανανέωση. Παρέχεται ακόμα, η δυνατότητα αλλαγής των στοιχείων σύνδεσης στο σύστημα (ηλεκτρονική διεύθυνση και κωδικός). Με την αλλαγή των στοιχείων αυτών, το σύστημα αποσυνδέει το Ταξιδιωτικό Γραφείο από το σύστημα ζητώντας του εκ νέου καταχώρηση των στοιχείων του για επανασύνδεση.

Έξοδο (log-out) Με επιλογή που είναι διαθέσιμη στο κεντρικό μενού των Ταξιδιωτικών Γραφείων, μπορεί να γίνει έξοδος από το σύστημα, διαγράφοντας τα στοιχεία της τελευταίας συνεδρίας (session).

3.6 Σύστημα Συστάσεων (Recommender)

Το Σύστημα Συστάσεων, και πιο συγκεκριμένα, οι συστάσεις αυτού είναι διαθέσιμα στον Χρήστη στην σελίδα αναζήτησης των ταξιδιωτικών πακέτων. Βρίσκονται στο πάνω μέρος της σελίδας, και είναι το πρώτο κομμάτι που αντικρίζει ο Χρήστης με το που επισκέπτεται αυτή την σελίδα. Η τοποθέτηση αυτή έγινε με την λογική ότι τα προτεινόμενα πακέτα είναι πιο πιθανό να αρέσουν στον Χρήστη και έτσι διευκολύνεται η διαδικασία αναζήτησής του για τον επόμενο προορισμό. Αυτά εμφανίζονται συνοδευόμενα από δύο σημαντικά χαρακτηριστικά, τις ονομασίες των Προορισμών και την τιμή τους, μιας και αυτά αρκούν για μια πρώτη εκτίμηση από τον Χρήστη. Φυσικά, παρέχεται και σχετικός σύνδεσμος στο κάθε πακέτο, έτσι ώστε αν κάποιο από αυτά τον ενδιαφέρει να έχει την δυνατότητα αναλυτικής προβολής των στοιχείων του. Τα προτεινόμενα από το Σύστημα Συστάσεων πακέτα είναι σε αριθμό πέντε, τα οποία κρίνονται ικανοποιητικά μιας και κάθε φορά που ο Χρήστης εκτελεί κάποια αναζήτηση ή λειτουργία στην συγκεκριμένη σελίδα, αυτά ανανεώνονται, είτε προτείνοντας νέα πακέτα είτε διατηρώντας μερικά ίδια λόγω της ιδιότητάς τους να κρίνονται ως τα πιο κατάλληλα για σύσταση σύμφωνα με την πρόβλεψη του Recommender.

ΚΕΦΑΛΑΙΟ 4

Υλοποίηση και ανάπτυξη

Στο συγκεκριμένο κεφάλαιο γίνεται αναφορά στις υπάρχουσες αρχιτεκτονικές συστημάτων οι οποίες ήταν υποψήφιες για επιλογή ως προς τον σχεδιασμό και την υλοποίηση του διαδικτυακού μας συστήματος. Στην συνέχεια, γίνεται μια σύντομη ανάλυση ως προς την αρχιτεκτονική η οποία επιλέχθηκε. Τέλος, θα περιγράψουμε την διαδικασία σχεδίασης, σύνθεσης, ανάπτυξης, και τελικης υλοποίησης του συστήματος.

4.1 Αρχιτεκτονική Συστήματος

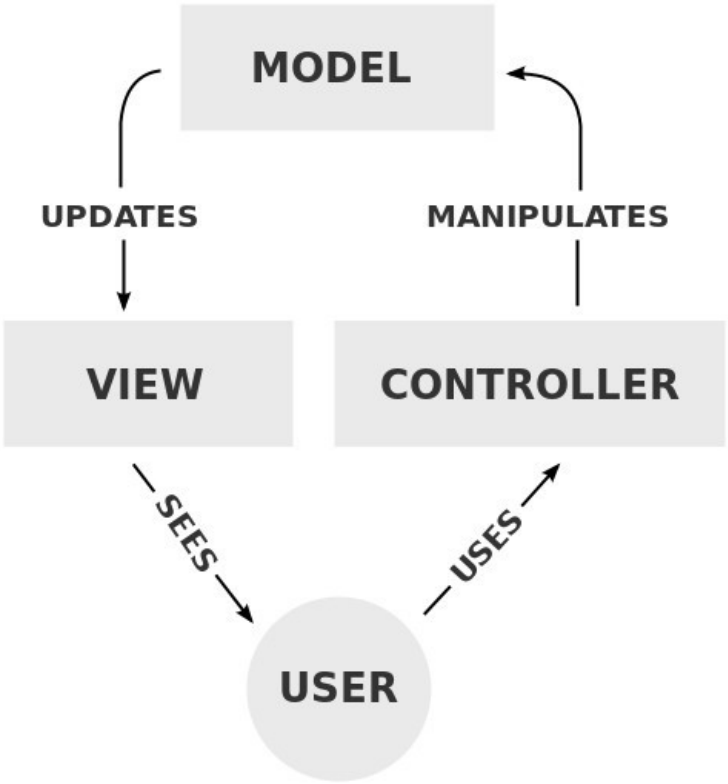
Αρχικά, γίνεται αναφορά στις 4 υποψήφιες αρχιτεκτονικές για την υλοποίηση του διαδικτυακού μας συστήματος και αυτές είναι το Μοντέλο Απεικόνισης – Ελεγκτή (Model – View Cotroller), η Αρχιτεκτονική βασισμένη σε εξαρτήματα (Component – based), η Αντικειμενοστρεφής τεχνική (Object Oriented Programming – based) και η Αρχιτεκτονική Τριών Στρωμάτων (Three – tier architecture). Στην πορεία αναλύεται η Αρχιτεκτονική Τριών Στρωμάτων μιας και αποτέλεσε την τελική μας επιλογή, και οι λόγοι της επιλογής της.

4.1.1 Μοντέλο Απεικόνισης – Ελεγκτή (MVC)

Το μοντέλο Απεικόνισης – Ελεγκτή (Model View Controller - MVC) είναι ένα μοντέλο αρχιτεκτονικής λογισμικού για την δημιουργία διεπαφών χρήστη (user-interfaces). Διαχωρίζει μια εφαρμογή σε τρία διασυνδεδμένα μέρη έτσι ώστε να διαχωρίζει την εσωτερική αναπαράσταση της πληροφορίας από τους τρόπους με τους οποίους η πληροφορία παρουσιάζεται στον χρήστη του συστήματος.

Το βασικό τμήμα της συγκεκριμένης αρχιτεκτονικής είναι το Μοντέλο (Model). Το τμήμα αυτό αναλαμβάνει την συμπεριφορά του συστήματος ως προς το πεδίο του προβλήματος το οποίο διαχειριζόμαστε, ανεξάρτητα από το user-interface. Διαχειρίζεται απευθείας τα δεδομένα, την λογική και τους κανόνες που τίθενται στην εφαρμογή. Το τμήμα View μπορεί να είναι οποιαδήποτε μορφή αναπαράστασης της πληροφορίας (ιστοσελίδα, διάγραμμα, κτλ) ως προς τον χρήστη. Πολλαπλά Views για την ίδια πληροφορία είναι πιθανά και μάλιστα αναμενόμενα για την συγκεκριμένη αρχιτεκτονική. Για παράδειγμα σε ένα σύστημα διαχείρισης περιεχομένου (Content Management System – CMS), για την ίδια πληροφορία έχουμε διαφορετικό View για τον εγγεγραμμένο χρήστη του συστήματος, διαφορετικο για τον επισκέπτη και διαφορετικό για τον διαχειριστή του συστήματος, πάντα

στο ίδιο κομμάτι της πληροφορίας. Το τρίτο κομμάτι, ο Controller, δέχεται την είσοδο του χρήστη και την μετατρέπει σε εντολές για το Model ή για το View. Υπάρχουν πολλές παραλλαγές υλοποίησης αυτής της αρχιτεκτονικής αλλά εμείς περιοριζόμαστε στον παραδοσιακό του ορισμό μιας και αρκεί για την ανάλυση της επιλογής μας ως προς τις υπάρχουσες αρχιτεκτονικές συστημάτων. Παρακάτω, βλέπουμε μια εικόνα η οποία παρουσιάζει τα διάφορα τμήματα του MVC, ορίζοντας τις μεταξύ τους αλληλεπιδράσεις οι οποίες περιγράφονται στην συνέχεια:



Σχήμα 4.1: Μια τυπική διασύνδεση – συνεργασία των τμημάτων της αρχιτεκτονικής Model – View Controller (MVC)

Ο **Controller** δέχεται το αποτέλεσμα της δράσης του χρήστη και είτε στέλνει εντολές στο Model για να αλλάξει την κατάστασή του (π.χ. κατά την αποθήκευση δεδομένων στην βάση δεδομένων ενός συστήματος) είτε στέλνει εντολές στο View για να αλλάξει την παρουσίαση του Model μέσω του View (π.χ. κατά το scrolling μιας ιστοσελίδας, ή το scrolling ενός εγγράφου κειμένου).

Το **Model** ενημερώνει τα διασυνδεδεμένα Views και τους Controllers όταν υπάρχει αλλαγή στην κατάστασή του. Η ενημέρωση αυτή προκαλεί τα Views να παρουσιάσουν updated output στον χρήστη, και τα Controllers να αλλαξουν το διαθέσιμο σύνολο των εντολών τους για επέμβαση στο Model απ'τον χρήστη.

Το **View** ζητάει δεδομένα απ'το Model για παρουσίασή τους στον χρήστη.

Η συγκεκριμένη αρχιτεκτονική αναπτύχθηκε αρχικά για desktop-computing, αλλά υιοθετήθηκε και από τις εφαρμογές του διαδικτύου οι οποίες υλοποιήθηκαν με δημοφιλείς γλώσσες προγραμματισμού. Η διάδοση αυτή προκάλεσε και την δημιουργία μεγάλων web application frameworks που δομήθηκαν πάνω σε αυτή την αρχιτεκτονική ακριβώς για τα χαρακτηριστικά που παρέχει ως προς την δυνατότητα δημιουργίας πολλαπλών interfaces πάνω σε ένα βασικό μοντέλο. Έτσι, ένα σύστημα διαχείρισης υλικού θα αξιοποιούσε αυτό το χαρακτηριστικό σε μεγάλο βαθμό σε σχέση με το δικό μας σύστημα το οποίο όπως θα δούμε έχει μικρό αριθμό Views.

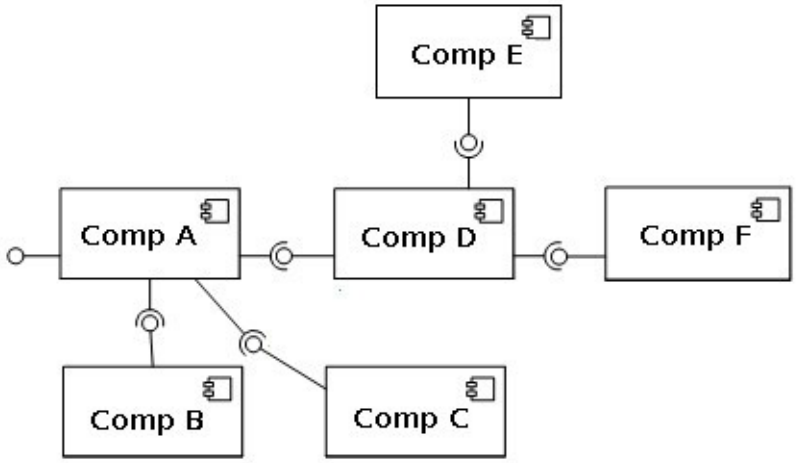
4.1.2 Αρχιτεκτονική βασισμένη σε εξαρτήματα (Component - based)

Η αρχιτεκτονική βασισμένη σε εξαρτήματα (Component – based architecture) είναι ένας κλάδος της μηχανικής λογισμικού (software engineering) που δίνει έμφαση στην αρχή σχεδίασης που είναι γνωστή στην επιστήμη των υπολογιστών ως separation of concerns (SoC), και αφορά την υλοποίηση μεγάλης έκτασης λειτουργικότητας σε ένα σύστημα. Σύμφωνα με αυτή την αρχή ένα πρόγραμμα για υπολογιστές διαχωρίζεται σε διακριτά τμήματα έτσι ώστε κάθε τμήμα διαχειρίζεται ένα διαφορετικό πρόβλημα (concern). Ένα concern είναι ένα σύνολο πληροφοριών που επηρεάζει τον κώδικα ενός προγράμματος. Η γενικότητα της έννοιας του concern μπορεί να είναι μεγάλη ή και πολύ μικρή και ο σαφής ορισμός του και αποτελεσματικότητά του ορίζεται από τους σχεδιαστές και προγραμματιστές κάθε προγράμματος ή και συστήματος. Η αξία του SoC είναι η απλοποίηση της ανάπτυξης και της συντήρησης των προγραμμάτων. Όταν τα concerns είναι σαφώς ορισμένα, κάθε τμήμα του προγράμματος μπορεί να χρησιμοποιηθεί ατομικά, και να παραχθεί ή να αναβαθμιστεί ανεξάρτητα από τα υπόλοιπα τμήματα. Από αυτό, προκύπτει η ακόμα μεγαλύτερης αξίας ιδιότητα αυτής της αρχής σχεδίασης λογισμικού η οποία είναι η δυνατότητα που δίνεται αργότερα στον προγραμματιστή να βελτιώσει ή να τροποποιήσει ένα κομμάτι του κώδικα χωρίς να χρειάζεται να γνωρίζει τις λεπτομέρειες των άλλων τμημάτων, και χωρίς να χρειάζεται να κάνει αλλαγές στα άμεσα συνδεόμενα τμήματα με το εκάστοτε προς τροποποίηση τμήμα.

Επιστρέφοντας στην έννοια του component, ένα software component είναι ένα software package, ένα web service, ένα web resource ή ένα module το οποίο ενθυλακώνει ένα set σχετικών λειτουργιών (ή και δεδομένων). Όλες οι διεργασίες του συστήματος είναι τοποθετημένες σε διαφορερικά components έτσι ώστε όλα τα δεδομένα και οι λειτουργίες μέσα σε κάθε component είναι σημασιολογικά συσχετισμένες. Για ευκολότερη κατανόηση της έννοιας, θα μπορούσαμε να πούμε ότι το τεχνολογικό ανάλογο της δομής των components είναι οι κλάσεις των γλωσσών αντικειμενοστρεφούς προγραμματισμού. Στην λογική της ευρείας λειτουργικότητας ενός συστήματος που αναφέραμε παραπάνω, τα components αυτά επικοινωνούν μεταξύ τους με διεπαφές (interfaces). Όταν ένα component προσφέρει τις υπηρεσίες του στο υπόλοιπο σύστημα, αποκτά μια διεπαφή προδιορίζει τις υπηρεσίες που τα άλλα components μπορούν να χρησιμοποιήσουν. Αυτού του είδους η διεπαφή στο παράδειγμα του σχήματος 4.2 συμβολίζεται με τον κύκλο. Όταν, ένα component χρειάζεται κάποιο άλλο component για να εκτελέσει τις λειτουργίες του τότε αποκτά μια διεπαφή ορίζει τις υπηρεσίες που χρειάζεται. Αυτή η διεπαφή συμβολίζεται με την ανοιχτή υποδοχή όπως φαίνεται παρακάτω.

Μια τέτοια αρχιτεκτονική αν και έχει πολύ ισχυρά χαρακτηριστικά δεν θεωρήθηκε κατάλληλη για την ανάπτυξη του συστήματος της συγκεκριμένης διπλωματικής εργασίας. Το

σύστημά μας δεν αφορά την ανάπτυξη ενός συστήματος με μεγάλο εύρος λειτουργιών που να απαιτούν τμηματοποίηση τους και ανάπτυξη ανα components. Αντιθέτως, οι λειτουργίες είναι περιορισμένες με την λογική ότι καλύπτουν ένα βασικό σύνολο εξυπηρέτησης των αναγκών ενός διαδικτυακού συστήματος και επικεντρωνόμαστε στο κομμάτι της ανάπτυξης λειτουργιών για την βελτιστοποίηση των τελικών καταναλωτικών επιλογών ενός χρήστη (εργαλείο σύγκρισης, σύστημα συστάσεων). Το μόνο στοιχείο που λειτουργεί ως component στο σύστημά μας είναι το σύστημα συστάσεων (recommender) μιας και υλοποιείται ως web service. Πέρα από αυτό, περαιτέρω ανάλυση των λειτουργιών του συστήματος σε επιμέρους components θα έκανε τον σχεδιασμό και την υλοποίηση του συστήματος πιο πολύπλοκη χωρίς αυτή η πολυπλοκότητα να οδηγούσε στην αξιοποίηση των θετικών χαρακτηριστικών αυτής της αρχιτεκτονικής.



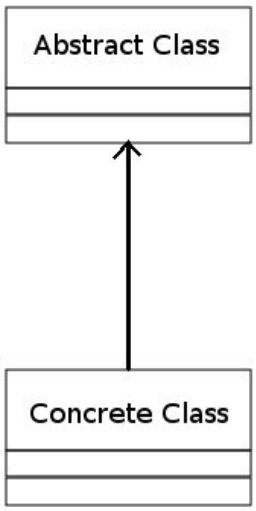
Σχήμα 4.2: Παράδειγμα component-based αρχιτεκτονικής με 6 components και συνολικά 11 διεπαφές

4.1.3 Χρήση αντικειμενοστρεφούς τεχνικής

Η χρήση αντικειμενοστρεφούς τεχνικής έχει να κάνει με την εκμετάλλευση των χαρακτηριστικών των αντικειμενοστρεφών γλωσσών προγραμματισμού για την ανάπτυξη του συστήματός μας με την χρήση κάποιου σχεδιαστικού προτύπου (design pattern). Τα design patterns αποτελούν μια επαναχρησιμοποιούμενη δοκιμασμένη λύση σε ένα συχνά εμφανιζόμενο πρόβλημα στο κομμάτι του software engineering. Δεν αποτελούν σε καμία περίπτωση ένα τελειωμένο σχέδιο που μπορεί μετασχηματιστεί σε κώδικα, αντιθέτως, αποτελούν περιγραφή για το πώς μπορεί να αντιμετωπιστεί ένα πρόβλημα σε πολλές διαφορετικές περιπτώσεις που έχουν στην βάση τους κοινά χαρακτηριστικά. Πιο συγκεκριμένα, τα αντικειμενοστρεφή design patterns τυπικά δείχνουν σχέσεις και αλληλεπιδράσεις μεταξύ κλάσεων και αντικειμένων χωρίς να προσδιορίζουν τον τελικό κώδικα της εφαρμογής. Τα αντικειμενοστρεφή design-patterns και γενικότερα τα πρότυπα που

εισάγουν μια μεταβαλλόμενη κατάσταση δεν είναι εφαρμόσιμα στις γλώσσες συναρτησιακού προγραμματισμού.

Μεταξύ των διαθέσιμων design-patterns (observer pattern, singleton pattern, iterator pattern, κτλ) ενδεικτικά αναφέρουμε το πιο σχετικό με τις ανάγκες της υλοποίησής μας pattern που αφορά το template method. Όπως δηλώνει και η ονομασία της μεθόδου, το template ορίζει μια συγκεκριμένη δομή η οποία χρησιμοποιείται ως βάση για την ανάπτυξη της υπόλοιπης εφαρμογής έτσι ώστε να μην γίνεται επανακατασκευή της δομής αυτής κάθε φορά που χρησιμοποιείται. Όπως και στην δημιουργία των GUI's σε ένα διαδικτυακό σύστημα, χρησιμοποιείται ένα template που ορίζεται -στην απλούστερη μορφή του- από μια κεφαλίδα, ένα περιεχόμενο και ένα υποσέλιδο, τα οποία αλλάζουν βάσει αναγκών και τοποθεσίας στον ιστότοπο έτσι ώστε να μην αναγκάζεται ο developer να επανακασκευάζει ίδια κομμάτια της σελίδας που επαναλαμβάνονται σε πολλά σημεία στον ιστότοπο. Έτσι και σε επίπεδο κώδικα λογισμικού η template method ορίζει έναν αλγόριθμο σε μια βασική κλάση χρησιμοποιώντας abstract λειτουργίες, οι οποίες γίνονται override υποκλάσεις με σκοπό να οριστεί μια συγκροτημένη δομή στο τελικό σύστημα. Στο σχήμα 4.3 φαίνεται η βασική δομή του template method, που περιγράφει ακριβώς την λογική ότι η abstract class εμπεριέχει την βάση δόμησης του κώδικα και η concrete class ορίζει τις επιμέρους διαφορετικές λειτουργίες με override στην abstract.

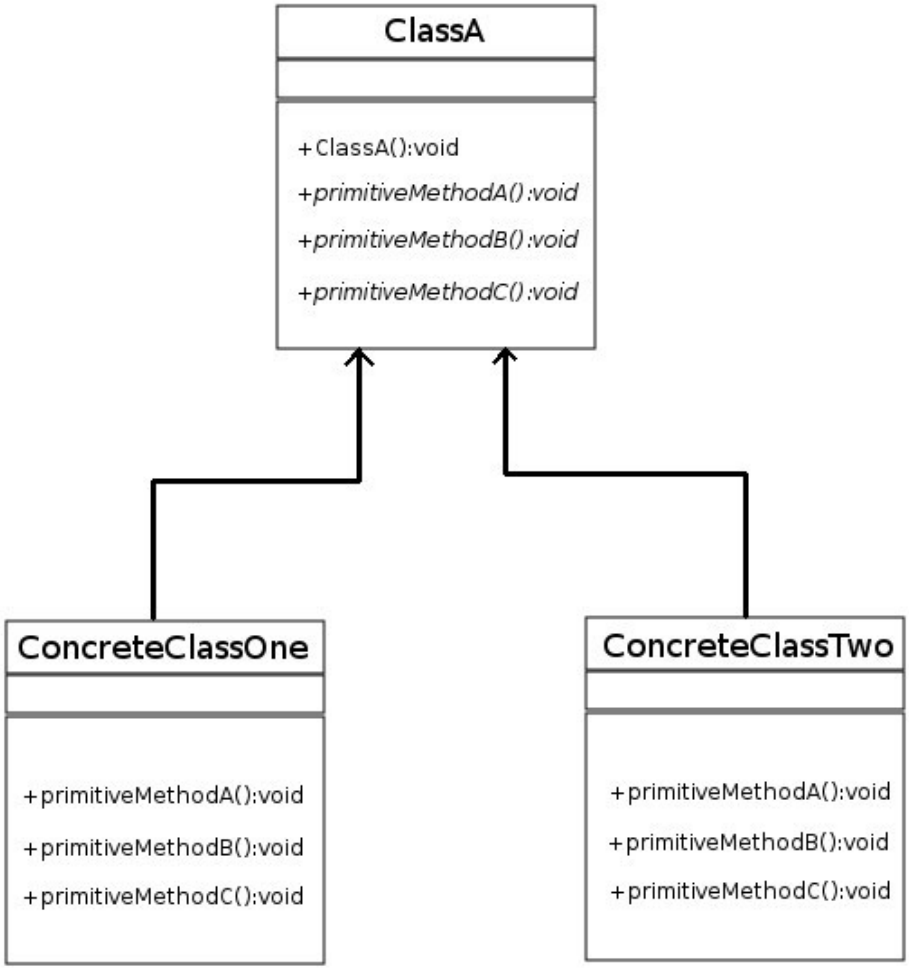


Σχήμα 4.3: UML διάγραμμα του σκελετού του template method

Η παραπάνω δομή εκφράζεται πιο αναλυτικά, από το παρακάτω σχήμα (4.4) όπου φαίνεται ο σκελετός που παρέχεται από την abstract class βάσει του οποίου, οι κλάσεις του συστήματος κάνουν override τις αρχικές λειτουργίες (primitiveOperations) της abstract class.

Παρά τα θετικά που παρέχει η συγκεκριμένη προσέγγιση όπως το reusability του κώδικα, η αποφυγή επαναληψιμότητας του κώδικα καθώς και η ανεξαρτησία των τμημάτων του συστήματος σε περίπτωση refactoring, δεν αποτελεί κατάλληλη προσέγγιση για την ανάπτυξη του συστήματος μας. Και αυτό διότι δεν παρατηρείται κάποια ομοιότητα στις λειτουργίες που παρέχονται από το σύστημα ώστε να εξάγουμε ένα σκελετό που θα αποτελέσει το template του κώδικά μας. Το μόνο που θα μπορούσε να ειπωθεί είναι ότι υπάρχει μια ενιαία λειτουργικότητα του συστήματος ως προς την ανάγκη του για συνεχή επικοινωνία με την βάση δεδομένων, τα ερωτήματα που πραγματοποιούνται και και τα updates που γίνονται σε αυτή. Πέρα από αυτό, το σύστημα παρουσιάζει μεγάλη ανομοιογένεια ως προς την ανάπτυξη κάθε

σελίδας. Ο μόνος διαχωρισμός που θα μπορούσε να γίνει στην λογική ανάπτυξης του συστήματός μας είναι σε επίπεδα επικοινωνίας από τον χρήστη (client) στον εξυπηρετητή (server). Αυτό το σκεπτικό μας οδηγεί στην προσέγγιση της αρχιτεκτονικής N – tier architecture, και πιο συγκεκριμένα στην 3- tier architecture την οποία αναλύουμε παρακάτω.

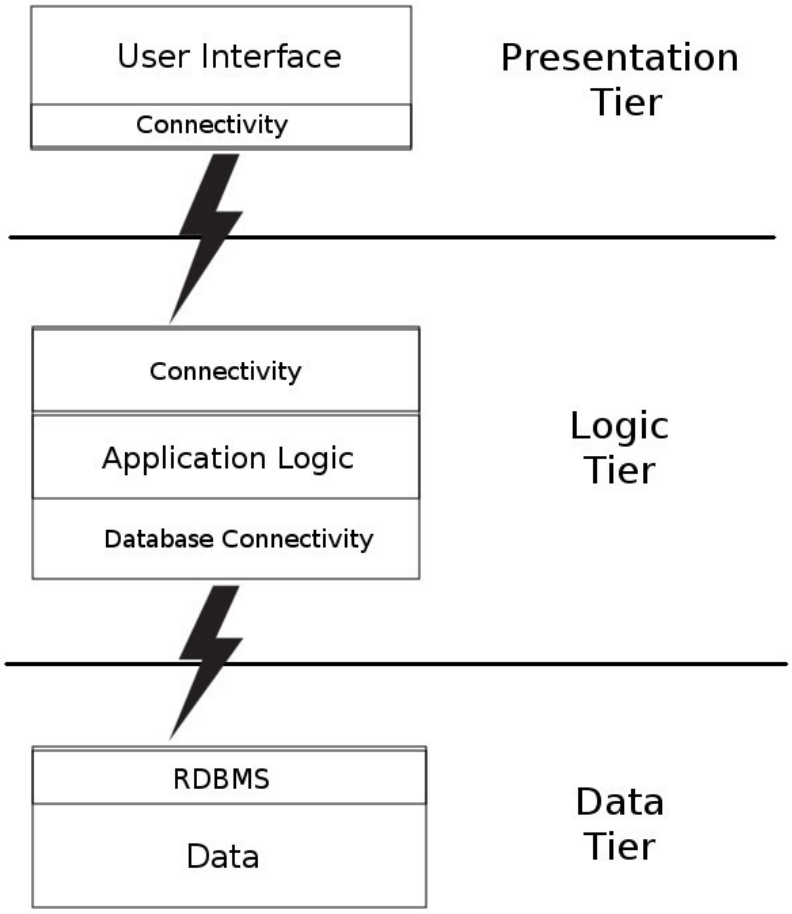


Σχήμα 4.4: Η ClassA είναι η βασική κλάση απ'την οποία γίνονται override οι primitive μέθοδοι από την εκάστοτε concrete κλάση.

4.1.4 Χρήση αρχιτεκτονικής τριών στρωμάτων

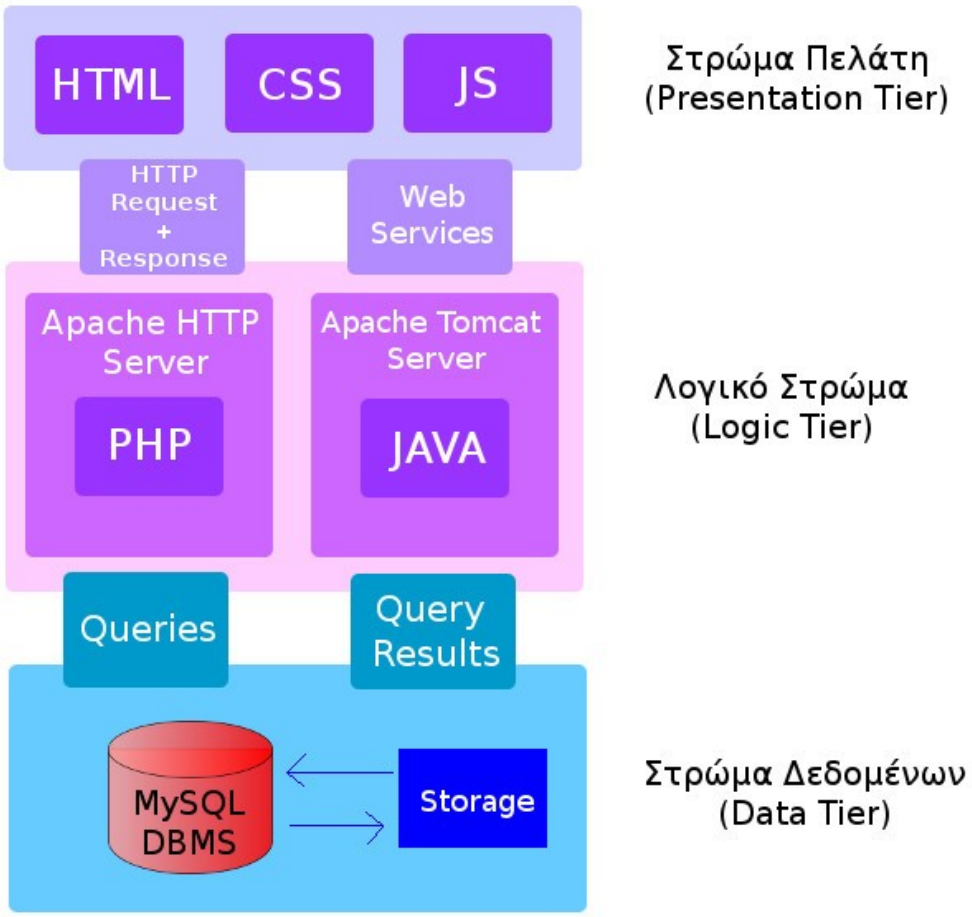
Η αρχιτεκτονική τριών στρωμάτων είναι μια αρχιτεκτονική πελάτη-εξυπηρετητή (client-server) στην οποία οι λειτουργίες της παρουσίασης, της επεξεργασίας και της

διαχείρισης των δεδομένων μιας διαδικτυακής εφαρμογής διαχωρίζονται σε τρία αντίστοιχα φυσικά επίπεδα. Τα επίπεδα αυτά είναι γνωστά ως στρώμα πελάτη (presentation tier), λογικό στρώμα (logic tier) και στρώμα δεδομένων (data tier). Πέρα από τα θετικά χαρακτηριστικά ενός τμηματοποιημένου λογισμικού με σαφώς ορισμένες διεπαφές, μερικά από τα οποία προκύπτουν λογικά από τις παραπάνω αναλύσεις μας, η αρχιτεκτονική τριών στρωμάτων δομείται ώστε να επιτρέπει την αναβάθμιση ή την αντικατάσταση οποιουδήποτε στρώματος ανεξάρτητα από τα υπόλοιπα ανάλογα με τις εκάστοτε ανάγκες λειτουργίας του συστήματος ή τις επιταγές της τεχνολογίας. Για παράδειγμα, μια αλλαγή λειτουργικού συστήματος σε επίπεδο πελάτη, θα επηρέαζε μόνο την διεπαφή χρήστη (user interface) και οποιαδήποτε αναγκαία αλλαγή στον κώδικα θα αφορούσε μόνο αυτό το κομμάτι. Σε ένα τυπικό σενάριο συστήματος με αυτή την αρχιτεκτονική είναι το εξής: Το στρώμα πελάτη εκτελείται σε έναν προσωπικό υπολογιστή, το στρώμα λογικής εκτελείται σε έναν εξυπηρετητή ιστού (το οποίο μάλιστα στρώμα μπορεί να αποτελείται από περισσότερα στρώματα σε περίπτωση χρήσης της N-tier architecture) και το στρώμα δεδομένων να εκτελείται σε έναν εξυπηρετητή βάσης δεδομένων με την χρήση ενός Συστήματος Διαχείρισης Βάσης Δεδομένων όπως περιγράψαμε στο κεφάλαιο 2. Η γενική ιδέα της αρχιτεκτονικής αυτής φαίνεται στο παρακάτω σχήμα (4.5).



Σχήμα 4.5: Η Δομή ενός συστήματος με την χρήση αρχιτεκτονικής τριών στρωμάτων και η γενικευμένη τμηματοποίηση των εμπλεκόμενων λειτουργιών σε κάθε στρώμα (tier)

Στην υλοποίηση της εφαρμογής μας χρησιμοποιούμε την συγκεκριμένη αρχιτεκτονική, με την διαστρωμάτωση των τεχνολογιών και την διασύνδεση των στρωμάτων όπως αυτή φαίνεται στο σχήμα 4.6 και η οποία θα αναλυθεί τμήμα – τμήμα.



Σχήμα 4.6: Η Διαστρωμάτωση της υλοποίησής μας όπως πραγματοποιήθηκε με την αρχιτεκτονική τριών στρωμάτων

Με διαθέσιμη την εικόνα του συνολικού συστήματος, εξετάζουμε ένα ένα τα στρώματα όπως τα υλοποιήσαμε, τον τρόπο με τον οποίο κάναμε αυτή την υλοποίηση καθώς και τους λόγους για τους οποίους επιλέξαμε αυτές τις τεχνολογίες.

4.1.4.1 Το Στρώμα Πελάτη (Presentation Tier)

Το Στρώμα Πελάτη αποτελεί το υψηλότερο στρώμα της αρχιτεκτονικής αυτής, και παρουσιάζει δεδομένα στον χρήστη συσχετισμένα με λειτουργίες όπως προβολή περιεχομένου μιας ιστοσελίδας, διαδικτυακές αγορές και περιεχόμενα καλαθιών αγοράς ανάμεσα σε πολλά άλλα. Επικοινωνεί με τα υπόλοιπα στρώματα υπό την έννοια ότι παρουσιάζει τα δεδομένα στον browser ή στο GUI μιας εφαρμογής και στέλνει δεδομένα στο λογικό στρώμα σύμφωνα

με τις ενέργειες των χρηστών.

Το Στρώμα Πελάτη πρέπει να υλοποιείται από ένα ελάχιστο σύνολο ενεργειών που διασφαλίζουν την ποιότητά του και την ασφάλεια του. Η υλοποίησή μας έγινε με γνώμονα αυτές τις λειτουργίες οι οποίες αναλύονται:

Caching Το caching είναι από τους βασικότερους μηχανισμούς που πρέπει να υλοποιεί το στρώμα δεδομένων για την βελτίωση της αποκρισιμότητας και γενικότερα των επιδόσεων της εφαρμογής. Με κατάλληλη μελέτη των δεδομένων και των λειτουργιών που χρησιμοποιούνται πιο συχνά από τους χρήστες, μπορεί να δημιουργηθεί ένας μηχανισμός προσωρινής αποθήκευσης αυτών, και να αποφεύγονται πολλαπλές αναζητήσεις και επεξεργασίες οι οποίες κοστίζουν σε χρόνο και σε υπολογιστικούς πόρους όταν η χρήση του συστήματος είναι μεγάλη. Το caching θα μπορούσε για παράδειγμα να υλοποιηθεί με την εκμετάλλευση του local storage ή των cookies που παρέχουν οι browsers στην περίπτωση μιας διαδικτυακής εφαρμογής, για την αποθήκευση στοιχείων του χρήστη, επιλογών του κατά την πλοήγηση στο σύστημα κτλ. Στο σύστημά μας έχει δωθεί βαρύτητα σε αυτό το κομμάτι καθώς η πλοήγηση σε μια μέση περίπτωση είναι συνεχής και εναλλασσόμενη μιας και ο χρήστης αναζητά, συγκρίνει και εξερευνά το περιεχόμενο του συστήματος σε αντίθεση με διαδικτυακές εφαρμογές όπου η κίνηση είναι περιορισμένη όπως για παράδειγμα σε μια ιστοσελίδα δημοσιογραφικού χαρακτήρα όπου ο χρήστης έχει μεγάλους χρόνους αναμονής σε μια σελίδα για ανάγνωση κάποιου άρθρου, η εισαγωγή στοιχείων είναι περιορισμένη και η αναζήτηση χρησιμοποιείται σε μικρό βαθμό.

Επικοινωνία με τα υπόλοιπα στρώματα η οποία αφορά την κλήση λειτουργιών του συστήματος από την πλευρά του χρήστη. Αυτές οι λειτουργίες μπορεί να αφορούν ερωτήματα της βάσης δεδομένων, εγγραφές σε αυτήν και κλήση πολλών δυναμικών σελίδων. Όσο πιο περιορισμένη και αποτελεσματική είναι η κλήση αυτών των λειτουργιών από τον χρήστη τόσο αποδοτικότερο είναι και το σύστημά μας. Έτσι, όταν για παράδειγμα απαιτούνται τα στοιχεία του χρήστη για πλοήγησή του σε όλο το σύστημα πρέπει να διασφαλίζουμε ότι η επικοινωνία με τα υπόλοιπα στρώματα του συστήματος θα γίνει μία φορά κατά την σύνδεσή του, κάνοντας χρήση του caching για αποφυγή επιπλέον περιττών συνδέσεων με την βάση δεδομένων.

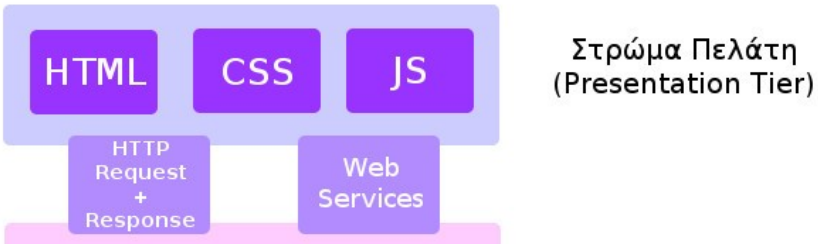
Διαχείριση εξαιρέσεων οι οποίες πρέπει να γίνονται με σαφώς ορισμένο μηχανισμό έτσι ώστε η διεπαφή χρήστη να παρέχει τις απαραίτητες πληροφορίες στον χρήστη για την πλοήγησή του στο σύστημα και να μη εκθέτουν εσωτερικές πληροφορίες και δομές του συστήματος που θα θέσουν την ασφάλειά του σε κίνδυνο.

Η Πλοήγηση στις σελίδες του συστήματος καθώς και η χρήση των λειτουργιών/υπηρεσιών του θα πρέπει να υλοποιείται με τρόπο αποδοτικό ως προς τον χρήστη. Πιο συγκεκριμένα, και για τις ανάγκες του συστήματος μας, η πλοήγηση με την χρήση κατάλληλων menus και submenus, και η εύκολη χρήση του πεδίου αναζήτησης για τα ταξιδιωτικά πακέτα – ιδιαίτερα όταν τα κριτήρια είναι πολλά – παίζουν μεγάλο ρόλο στην διευκόλυνση του χρήστη για πλοήγηση στο σύστημά μας. Διαφορετικά, υπάρχει ο κίνδυνος της δυσανασχέτισης του χρήστη και η επιλογή συστημάτων που προσφέρουν τις ίδιες λειτουργίες με πιο αποδοτικό τρόπο. Σε αυτό το κομμάτι συμπεριλαμβάνεται και ο σαφής διαχωρισμός της πληροφορίας και των στοιχείων παρουσίασης από τα στοιχεία πλοήγησης. Ο διαχωρισμός αυτός εξασφαλίζεται τόσο από την δομή του συστήματος όσο και από την γραφιστική διαμόρφωση των σελίδων όπου έχει δοθεί βαρύτητα με επιλογή κατάλληλης αντιστοιχίας χρωμάτων στα στοιχεία της παρουσίασης και στα στοιχεία της πλοήγησης (κουμπιά, μενού, πεδία εισαγωγής δεδομένων κτλ.)

Η Εμπειρία Χρήστη ορίζεται ώς η αντίληψη και η ανταπόκριση ενός χρήστη από την χρήση ενός συστήματος ή μιας υπηρεσίας. Η καλή εμπειρία χρήστη (user experience), λοιπόν, έχει ιδαίτερη βαρύτητα στα εμπορικά συστήματα, και μερικές φορές θα μπορούσαμε να πούμε ότι έχει μεγαλύτερη σημασία και από το ίδιο περιεχόμενο και τη λειτουργία του συστήματος με την έννοια ότι ένα σύστημα με πολύ καλό user experience και σχετικά λιγότερο υλικό από άλλα συστήματα μπορεί πάντα να προτιμάται από τους χρήστες, μιας και τους είναι ευχάριστο και εύκολο να το χρησιμοποιήσουν. Για να γίνει πιο κατανοητό, αν ο χρήστης κατά την χρήση μιας υπηρεσίας η οποία απαιτεί αρκετό χρόνο για να ολοκληρωθεί, ενημερώνεται για την εξέλιξή της τότε η εμπειρία του θα είναι θετική. Αν όμως υπάρχει καθυστέρηση σε φαινομενικά μη αναμενόμενο σημείο – για παράδειγμα κατά την εισαγωγή στοιχείων σε φίλτρα αναζήτησης – τότε η εμπειρία του θα είναι αρνητική και δύσκολα θα προτιμήσει τις υπηρεσίες που του προσφέρονται από το υπόλοιπο σύστημα, διατηρώντας το προηγούμενο συμβάν στο μυαλό του. Στην υλοποίηση του συστήματός μας δόθηκε βαρύτητα και σε αυτό το κομμάτι μέσω της καλής υλοποίησης των χαρακτηριστικών του Caching και της Πλοήγησης που αναφέραμε παραπάνω, καθώς και της γενικότερης γρήγορης ανταποκρισημότητας του συστήματος.

Η Διεπαφή Χρήστη (User Interface) είναι ένα επιπλέον σημαντικό χαρακτηριστικό του στρώματος πελάτη στο οποίο δώσαμε βάση. Ιδιαίτερα για ένα διαδικτυακό σύστημα σαν αυτό που υλοποιούμε για τις ανάγκες της εργασίας θα πρέπει να λαμβάνουμε υπόψη το μεγάλο πλήθος ανομοιογένειας ως προς το υλικό και το λογισμικό πρόσβασης στο user interface μας. Αυτό σημαίνει διαφορετικές αναλύσεις οθονών και μεγέθη και διαφορετικά προγράμματα πλοήγησης δημιουργούν προβλήματα τα οποία πρέπει το στρώμα πελάτη του συστήματός μας να είναι σε θέση να τα αντιμετωπίσει. Έτσι, η ανάπτυξη των σελίδων γίνεται με κώδικα που να είναι συμβατός με όλους τους browsers, η μορφοποίησή του με κανόνες που να μεταφράζονται ομοιόμορφα απ'όλους τους browsers, και μεγέθη τα οποία να είναι responsive στο μέγεθος της οθόνης στην οποία γίνεται προβολή του υλικού της εκάστοτε σελίδας.

Η υλοποίηση του Στρώματος Πελάτη γίνεται με την χρήση εγγράφων υπερκειμένου (HTML) . Οι σελίδες αυτές παράγονται δυναμικά από το Στρώμα λογικής από την PHP όπως θα δούμε



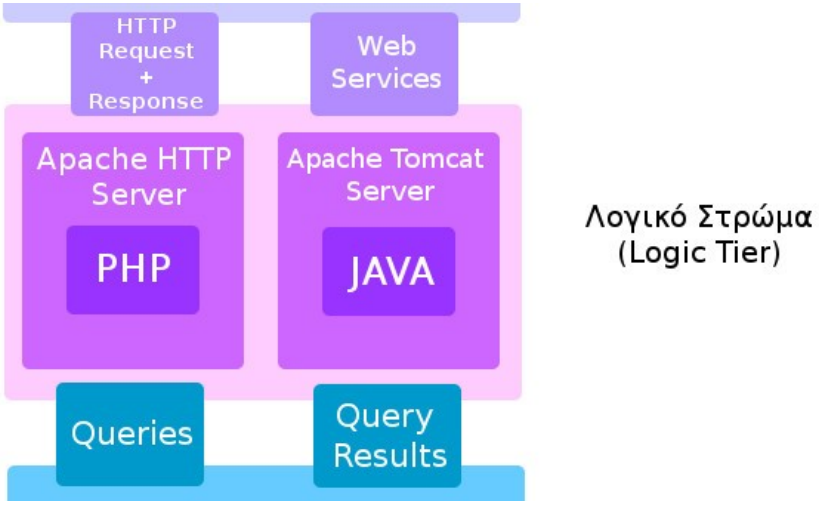
Σχήμα 4.7: Το Στρώμα Πελάτη και οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή του. Η επικοινωνία με το Στρώμα Λογικής γίνεται με χρήση HTTP και με την χρήση Web Services

παρακάτω. Οι σελίδες αυτές απεικονίζονται σε έναν περιηγητή ιστού (web browser) και έχει εξασφαλιστεί η συμβατότητα με όλους τους διάσημους περιηγητές ιστού όπως είναι ο Internet Explorer, ο Mozilla Firefox, ο Google Chrome, ο Opera και ο Safari. Η επικοινωνία με το Στρώμα Λογικής γίνεται μέσω του διαδικτύου καθώς τα δύο αυτά στρώματα γίνεται σε διαφορετικούς φυσικούς χώρους. Η επικοινωνία αυτή γίνεται μέσω client και ενός

εξυπηρετητή ιστού ή ενός Java εξυπηρετητή ιστού όπως φαίνεται και στο σχήμα 4.7. Η επιλογή των τεχνολογιών για την υλοποίηση αυτού του στρώματος, έγινε βάσει χαρακτηριστικών του συστήματός μας (διαδικτυακό) άρα και υποχρεωτικά HTML για την παρουσίαση των δεδομένων και CSS για την μορφοποίηση του περιεχομένου και των διαχωρισμό πληροφορίας – λειτουργιών. Ανάμεσα σε άλλες επιλογές για την μορφοποίηση του περιεχομένου, επιλέχθηκε η CSS ως η πλέον συμβατή με όλα τα μηχανήματα μέσω των οποίων είναι πιθανό να γίνει πρόσβαση στο σύστημα, ενώ αποτελεί λύση για χρήστες οι οποίοι δεν έχουν διαθέσιμη (ή και απενεργοποιημένη) την javascript. Η επιλογή της javascript ως γλώσσας προγραμματισμού για το κομμάτι του Στώματος Πελάτη ήταν άμεση και φυσική μιας και είναι συμβατή με όλους τους browsers και παρέχει όλα τα απαραίτητα εργαλεία για την απόδοση των χαρακτηριστικών της Πλοήγησης, του Caching και της καλής Εμπειρίας Χρήστη που περιγράψαμε παραπάνω. Ακόμα η javascript χρησιμοποιείται σε πολλά frameworks που προσφέρουν λειτουργίες από DOM manipulation μέχρι και σε server-side network programming, δημιουργία responsive themes κτλ. Στην υλοποίησή μας, δεν χρησιμοποιούμε κάποιο framework, ενώ κάνουμε αυστηρό διαχωρισμό των Στρωμάτων της αρχιτεκτονικής μας που σημαίνει ότι δεν επιτρέπουμε την πρόσβαση χρήστη από το Στρώμα Πελάτη στο Στρώμα Δεδομένων κατευθείαν, αλλά περιοριζόμαστε στο DOM manipulation και στην χρήση του Local Storage των browsers.

4.1.4.2 Το Στρώμα Λογικής (Logic Tier)

Το Στρώμα Λογικής είναι γνωστό και ως Στρώμα Επιχειρηματικής Λογικής, ή Μέσο Στρώμα. Σε αυτό το στρώμα υλοποιούμε όλες τις βασικές λειτουργίες του συστήματος. Ως βασικές λειτουργίες ορίζουμε την επικοινωνία με την βάση δεδομένων (ανάκτηση, εγγραφή), την διασύνδεση Στρώματος Πελάτη και Στρώματος Δεδομένων ως προς τις επιλογές του χρήστη καθώς και την επεξεργασία των δεδομένων. Η επεξεργασία των δεδομένων μπορεί να γίνεται είτε άμεσα για ανάγκες παρουσίασης όπως για παράδειγμα η επεξεργασία κειμένου για παρουσίαση περίληψή του στον χρήστη είτε με την εφαρμογή μαθηματικών και λογικών μοντέλων για την εκτέλεση άλλων υπηρεσιών του συστήματος όπως είναι για το σύστημα συστάσεων που υλοποιούμε. Στο Στρώμα Λογικής απαιτείται ιδιαίτερη προσοχή στο ζήτημα των κανόνων λογικής και της πολιτικής που έχουμε ορίσει κατά την σχεδίαση, διαφορετικά τίθεται σε κίνδυνο η ασφάλεια του συστήματός μας. Έτσι, η καταχώρηση μιας πληρωμής και η ενημέρωση του λογαριασμού του χρήστη για την κίνησή του αυτή κινείται στα πλαίσια των κανόνων λογικής που αναφέραμε, και χρειάζεται ιδιαίτερη προσοχή η έγκυρη λειτουργία τους αλλιως καταλήγουμε σε ασυνέπεια των δεδομένων μας και της λειτουργικότητας του συστήματός μας. Ακόμα, η περιοχές του ιστοτόπου στις οποίες έχει πρόσβαση ο χρήστης αποτελεί ένα παράδειγμα πολιτικής του συστήματος που ορίζεται και πάλι στο στρώμα αυτό, όπου κακή εφαρμογή της πολιτικής σημαίνει έκθεση του συστήματος σε ζητήματα ασφάλειας. Η εκτέλεση του Στρώματος Λογικής -κατά γενικό κανόνα και ιδιαίτερα σε διαδικτυακές εφαρμογές – εξυπηρετείται από έναν application server ο οποίος βρίσκεται σε απομακρυσμένο φυσικό χώρο από τον πελάτη (client) και λειτουργεί σε έναν εξυπηρετητή ιστού – και στην περίπτωση της εργασίας μας σε έναν Java εξυπηρετητή ιστού.



Σχήμα 4.8: Το Στρώμα Πελάτη, οι τεχνολογίες που χρησιμοποιήθηκαν και ο ρόλος του ώς διασύνδεση του Στρώματος Παρουσίασης με το Στρώμα Δεδομένων

Οι βασικές λειτουργίες του Στρώματος Δεδομένων όπως τις προσεγγίσαμε στην υλοποίηση μας είναι:

Πιστοποίηση και Εξουσιοδότηση χρηστών Εξουσιοδότηση κατά τέτοιο τρόπο ώστε να αποφεύγεται η πρόσβαση σε χώρους του συστήματός μας από μη εξουσιοδοτημένους χρήστες, διότι αυτό θέτει σε κίνδυνο το σύστημά μας και την ακεραιότητα των λειτουργιών που προσφέρουμε στους χρήστες. Έτσι, στο παράδειγμα του δικού μας συστήματος πρέπει και είναι σαφής ο ορισμός των ρόλων των χρηστών και των χώρων στους οποίους έχουν πρόσβαση. Οι ρόλοι αυτοί είναι όπως έχει αναφερθεί στο σχετικό κεφάλαιο, ο Χρήστης και το Ταξιδιωτικό Γραφείο. Θα μπορούσε να εισαχθεί και ο ρόλος του Διαχειριστή όπου η πρόκληση της υλοποίησης απαιτεί ακόμα πιο προσεκτικό σχεδιασμό αλλά αυτό δεν συνέβη λόγω της προσέγγισης που επιλέξαμε για την παρούσα εργασία. Στην ίδια λογική, δεν υπάρχει μια αυστηρή στρατηγική ελέγχου ταυτότητας και πιστοποίησης του χρήστη με την έννοια ότι αυτό κινείται έξω από τα πλαίσια της διπλωματικής, αλλά διατηρείται μια τυπική πιστοποίηση σε επίπεδο μή εμπορικής χρήσης του συστήματος.

Caching Το Στρώμα Λογικής αναλαμβάνει γενικά ένα μεγάλο κομμάτι του caching, με την έννοια ότι υλοποιεί την στρατηγική αποθήκευσης των κατάλληλων δεδομένων για αποφυγή περιττών κλήσεων σε μονάδες του συστήματος και στην επεξεργασία των δεδομένων. Έτσι, η μη επαναληψιμότητα των ερωτημάτων στην βάση δεδομένων μπορεί να βοηθήσει στα μέγιστα την απόδοση του συστήματός μας και αυτό μπορεί να γίνει με την χρήση session variables, είτε με την αποθήκευση στοιχείων που έχουν ήδη ερωτηθεί στην βάση δεδομένων και μας εξυπηρετεί η διατήρησή τους σε κάποια μεταβλητή, την στιγμή που αυτά τα στοιχεία θα χρειαστούν σε πολλά διαφορετικά σημεία και λειτουργίες του συστήματος. Να σημειωθεί ότι στην υλοποίηση της εφαρμογής μας το caching επιτεύχθηκε τόσο απ'το Στρώμα Πελάτη όσο και από το Στρώμα Λογικής χωρίς ιδιαίτερη βαρύτητα σε κάποιο από τα δύο στρώματα, λαμβάνοντας υπόψη κυρίως τις ανάγκες των λειτουργιών που θέλουμε να βελτιώσουμε. Έτσι, κατά την επιλογή πακέτων από το εργαλείο σύγκρισης χρησιμοποιούμε το Στρώμα Πελάτη για την διατήρηση των στοιχείων αυτών μιας και υλοποιείται ευκολότερα σε επίπεδο DOM manipulation, ενώ κατά την είσοδο (log-in) του χρήστη στο σύστημα, χρησιμοποιούμε το Στρώμα Λογικής για αποθήκευση των στοιχείων του κατά την πλοήγησή του στο σύστημα. Τέλος, αξίζει να σημειωθεί ότι -όπως και στα εμπορικά

συστήματα- δίνεται προσοχή στα ευαίσθητα δεδομένα, τα οποία δεν τα αποθηκεύουμε πουθενά για λόγους ασφαλείας.

Διαχείριση Εξαιρέσεων Ο σχεδιασμός ενός αποτελεσματικού συστήματος διαχείρισης εξαιρέσεων είναι πολύ σημαντικός για την ασφάλεια και την αξιοπιστία ενός συστήματος. Παρ'ολα αυτά στο σύστημά μας δεν έχει δοθεί κάποια ιδιαίτερη βαρύτητα στο κομμάτι αυτό μιας και ενδιαφερόμαστε περισσότερο για το αποτελεσματικό deployment των υπηρεσιών που υλοποιήσαμε. Το κομμάτι της ασφάλειας κινείται έξω απ'τα πλαίσια της συγκεκριμένης εργασίας αλλά η Διαχείριση Εξαιρέσεων αποτελεί γενικότερα πολύ σημαντικό κομμάτι του software engineering και αξίζει να σημειωθεί.

Καταγραφή (logging) Η υλοποίηση ενός αποτελεσματικού συστήματος καταγραφής και ελέγχου της λειτουργίας του συστήματος καθώς και της χρήσης του από τους χρήστες είναι επίσης σημαντική για την ασφάλεια και την αξιοπιστία του συστήματος. Έτσι, μπορούν να καταγράφονται κινήσεις των χρηστών για κινήσεις τους μέσα στο σύστημα οι οποίες μπορούν να βοηθήσουν τον προγραμματιστή ή τον διαχειριστή του συστήματος να επιδιορθώσει προβλήματα και bugs τα οποία μπορεί να προκύψουν από την απρόβλεπτη συμπεριφορά των χρηστών καθώς επίσης και να υπάρχει μια σαφής εικόνα της υγείας και της λειτουργικότητας του συστήματος.

Συνέχεια/Συνέπεια Κατάστασης Στην εφαρμογή που αναπτύχθηκε, το Στρώμα Λογικής έχει και τον πολύ σημαντικό ρόλο του να αποδίδει χαρακτηριστικά συνέχειας στις καταστάσεις λειτουργίας (statefull) του συστήματός μας. Όπως αναφέραμε και στο κεφάλαιο 2 το πρωτόκολλο HTTP είναι stateless πρωτόκολλο, οπότε χρησιμοποιούμε την PHP για την υλοποίηση συνεχών και συνεπών καταστάσεων λειτουργίας του συστήματος με την χρησιμοποίηση συνεδριών (sessions). Έτσι, όταν για παράδειγμα ένας χρήστης έχει αναγνωριστεί από το σύστημα ως εγγεγραμμένος χρήστης του συστήματος, παραμένει αναγνωρισμένος μέχρι να επιλέξει ο ίδιος την έξοδό του από το σύστημα χωρίς να του ξαναζητηθούν τα στοιχεία του (logged – in).

Έλεγχος δεδομένων κατά την εισαγωγή τους από έναν χρήστη σε ένα html form. Στην εφαρμογή μας, υπάρχουν πολλές περιοχές όπου η html χρησιμοποιείται ώς διεπαφή για την είσοδο δεδομένων στο σύστημα. Έτσι, πέρα από το φίλτρο αναζήτησης υπάρχει ανάγκη για input του χρήστη σε πεδία που έχουν σχέση με εισαγωγή δεδομένων στην βάση δεδομένων μας. Επομένως, πέρα από την αξιολόγηση για την έγκυρη είσοδο δεδομένων στα αντίστοιχα πεδία – όπως έλεγχο για τον αν ένα input είναι όντως αριθμός την στιγμή που ζητείται ένας αριθμός τηλεφώνου – απαιτείται προστασία αυτών των πεδίων από τις κακόβουλες επιθέσεις. Στην υλοποίηση της συγκεκριμένης εργασίας λάβαμε σοβαρά υπόψη αυτόν τον παράγοντα μιας και το σύστημά μας αφενός διαχειρίζεται σημαντικό αριθμό html forms, αφαιτέρου λειτουργεί σε live server, προσβάσιμο από οποιονδήποτε χρήστη του διαδικτύου.

Στην εφαρμογή μας, χρησιμοποιήσαμε τις τεχνολογίες που φαίνονται στην εικόνα 4.8 για την υλοποίηση του Στρώματος Λογικής. Όλη η εφαρμογή υλοποιείται σε δύο βασικά τμήματα. Ο κορμός του συστήματος υλοποιείται σε PHP από έναν εξυπηρετητή ιστού Apache HTTP Server. Το δεύτερο τμήμα αφορά την υλοποίηση του συστήματος συστάσεων και αυτό υλοποιήθηκε σε Java και εξυπηρετείται από ένα Java εξυπηρετητή ιστού, τον Apache Tomcat Server. Η επικοινωνία ανάμεσα στο Στρώμα Πελάτη και στο Στρώμα Λογικής γίνεται μέσω του διαδικτύου και επομένως αυτά τα δύο στρώματα βρίσκονται σε διαφορετικούς φυσικούς χώρους. Αντιθέτως, η επικοινωνία του Στρώματος Λογικής με το Στρώμα Δεδομένων γίνεται στον ίδιο φυσικό χώρο, από τον ίδιο Server.

Η γλώσσα προγραμματισμού PHP είναι μια γλώσσα προγραμματισμού σεναρίων (scripting language) με ό,τι χαρακτηριστικά συνεπάγεται αυτό, όπως αναλύσαμε στο Κεφάλαιο 2 στο κομμάτι των γλωσσών προγραμματισμού σεναρίων. Είναι γλώσσα ελεύθερου λογισμικού ανοιχτού κώδικα η οποία χρησιμοποείται για την ανάπτυξη εφαρμογών ιστού μέσω της δημιουργίας δυναμικών σελίδων. Είναι από τις πρώτες server-side γλώσσες προγραμματισμού, και ο διερμηνέας της ονομάζεται Zend Engine. Η PHP μπορεί να χρησιμοποιηθεί είτε συνθέτοντάς την με HTML είτε με την χρήση διαφόρων μηχανών template engines και web frameworks.

Τα βασικά μειονεκτήματα της PHP είναι η αδύναμη υποστήριξη αντικειμενοστρεφούς προγραμματισμού και ορισμένες αδυναμίες ως προς την ασφάλεια των εφαρμογών που αναπτύσσονται με την χρήση της γλώσσας αυτής. Παρ'ολα αυτά με την αξιοποίηση ασφαλών πρακτικών προγραμματισμού, το πρόβλημα της ασφάλειας αντιμετωπίζεται με επιτυχία, διατηρώντας την ως βασική επιλογή για την ανάπτυξη εφαρμογών διαδικτύου. Τα πλεονεκτήματα της PHP για την οποία την επιλέξαμε είναι ότι σε αντίθεση με άλλες γλώσσες (όπως είναι η ASP) αποτελεί λογισιμικό ελεύθερου λογισμικού ανοιχτού κώδικα όπως αναφέραμε παραπάνω, παρέχει ευκολία διασύνδεσης με άλλες οντότητες ενός διαδικτυακού συστήματος όπως είναι ένα Σύστημα Διαχείρισης Βάσης Δεδομένων και ένα Java Servlet και ευκολία εγκατάστασης και μεταφερσιμότητας σε σχεδόν οποιοδήποτε εξυπηρετητή Ιστού, με την χρήση οποιουδήποτε λειτουργικού συστήματος και πλατφόρμας.

Ο εξυπηρετητής Ιστού Apache (Apache HTTP Server) χρησιμοποιήθηκε γιατί πρώτον είναι open source (ελεύθερο λογισμικό ανοιχτού κώδικα) και είναι διαθέσιμος σε όλες σχεδόν τις σύγχρονες υπολογιστικές πλατφόρμες. Για τον λόγο αυτό είναι ο πιο δημοφιλης εξυπηρετητής Ιστού παγκοσμίως και έτσι ευνοεί την εγκατάσταση και την μεταφερσιμότητα του συστήματός μας ενώ παράλληλα διευκολύνει την διασύνδεση τόσο με το Στρώμα Πελάτη όσο και με το υλικό που θα χρησιμοποιηθεί για την εξυπηρέτηση όλου του συστήματος.

Για την επιλογή της γλώσσας Java, μπορούμε να πούμε ότι η επιλογή ήταν αναπόφευκτη. Στο κεφάλαιο 2 αναλύσαμε πλήρως τα χαρακτηριστικά της. Εδώ επαναλαμβάνουμε τα βασικά της χαρακτηριστικά για τα οποία είναι διάσημη και κατ'επέκταση αποτέλεσε επιλογή μας. Αρχικά, είναι open-source. Κατά δεύτερον, αποτελεί γλώσσα ή οποία μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα χωρίς καμία επέμβαση στον κώδικά της, λόγω της χρήσης του JVM για την μεταγλώττιση της, γι'αυτό και χαρακτηρίζεται στην κοινότητα των developers από την φιλοσοφία “write once, run anywhere”. Για τους λόγους αυτούς, λοιπόν, η συγκεκριμένη γλώσσα αντικειμενοστρεφούς προγραμματισμού είναι διάσημη στην ανάπτυξη μεγάλου πλήθους εφαρμογών, με κύριο χαρακτηριστικό την ανάπτυξη εφαρμογών πελάτη-εξυπηρετητή. Φυσικά σε μια τέτοια δημοφιλή γλώσσα προγραμματισμού, πάντα υπάρχουν και διαθέσιμες βιβλιοθήκες για την ανάπτυξη πολλών τύπων εφαρμογών, από mobile applications μέχρι και την εφαρμογή αλγορίθμων Μηχανικής Μάθησης. Έτσι, και την υλοποίηση του συστήματος συστάσεων την πραγματοποιήσαμε σε βιβλιοθήκες Java, οι οποίες είναι διαθέσιμες από το open-source project Apache Mahout, το οποίο έχει επίσης αναλυθεί στο κεφάλαιο 2. Το σύστημα συστάσεων για να μπορέσει να εξυπηρετηθεί, χρειάζόταν έναν servlet container. Για servlet container επιλέξαμε τον Apache Tomcat Server επειδή είναι open-source και αποτελεί την πιο δημοφιλή επιλογή για Java Web Servers. Επομένως, για άλλη μια φορά, η εφαρμογή του σε διάφορετικές πλατφόρμες είναι εύκολη και αποδοτική, καθώς επισής καθιστά εύκολη την μεταφερσιμότητά του.

4.1.4.3 Το Στρώμα Δεδομένων (Data Tier)

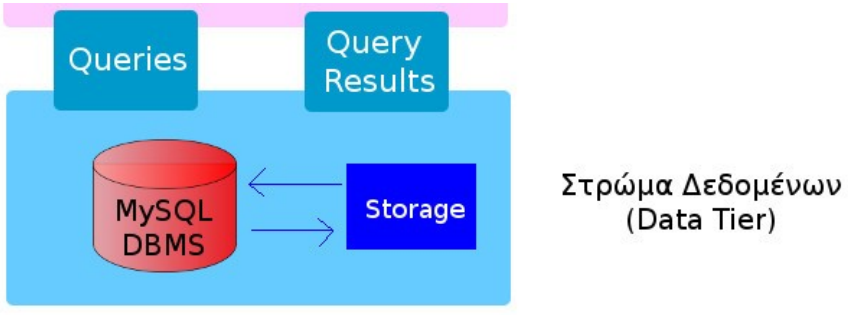
Είναι γνωστό και ως Στρώμα βάσης δεδομένων. Το συγκεκριμένο στρώμα είναι υπεύθυνο για την πρόσβαση στα δεδομένα του συστήματος από το Στρώμα Λογικής μέσω

σαφώς ορισμένων διεπαφών. Επίσης είναι υπεύθυνο για την διαχείριση των δεδομένων κατά τα ερωτήματα του Στρώματος Λογικής, την εγγραφή δεδομένων στο φυσικό μέσο αποθήκευσης (σκληρό δίσκο) καθώς και στην συνέπεια του συστήματος ώς προς την λειτουργία του. Αυτό, πραγματοποιείται από κάποιο Σύστημα Διαχείρισης Βάσεων Δεδομένων. Στο σχήμα 4.9, βλέπουμε τις τεχνολογίες που εμπλέκονται στο Στρώμα Δεδομένων του συστήματός μας και τον τρόπο επικοινωνίας με τα υπόλοιπα Στρώματα.

Οι κυριότερες λειτουργίες του Στρώματος Δεδομένων, περιγράφονται παρακάτω.

Ομαδοποίηση των ερωτημάτων ως προς την βάση δεδομένων. Συνήθως, οι λειτουργίες που αφορούν ερωτήματα και ενέργειες στον χώρο της βάσης δεδομένων, αποτελούν συνήθως το μεγαλύτερο κομμάτι κατανάλωσης πόρων από την εκάστοτε εφαρμογή. Μάλιστα, ιδιαίτερα όταν η επισκεψιμότητα στην εφαρμογή είναι μεγάλη, τα προβλήματα που αντιμετωπίζονται σε θέματα ανταποκρισημότητας και επεκτασιμότητας, αφορούν κυρίως την βάση δεδομένων. Επομένως, η ομαδοποίηση ερωτημάτων μπορεί να βελτιώσει κατά πολύ τις επιδόσεις του συστήματος. Στο σύστημά μας, δεν δόθηκε βαρύτητα στο κομμάτι της ομαδοποίησης διότι τα ερωτήματα στην βάση δεδομένων είναι περιορισμένου αριθμού και εμβέλειας. Παρ'όλα αυτά δόθηκε βάση στην υλοποίηση της διεπαφής με το Στρώμα Λογικής ώστε η πρόσβαση στο Στρώμα Δεδομένων, να γίνεται μόνο από αυτό.

Οι συνδέσεις με τις βάσεις δεδομένων απαιτούν επίσης προσοχή με την έννοια ότι αποτελούν παράγοντα παραβίασης της ασφάλειας του συστήματος μας. Έτσι, όσο το δυνατόν λιγότερες και μικρότερης διάρκειας συνδέσεις, τόσο καλύτερο για την ασφάλεια του συστήματός μας. Στο κομμάτι αυτό, η εφαρμογή μας έδωσε βαρύτητα, πραγματοποιώντας συνδέσεις που έχουν μόνο όταν αυτές έχουν νόημα, χωρίς να αφήνουμε ανοιχτές συνδέσεις καθ'όλη την διάρκεια της συνεδρίας ενός χρήστη με μία ή περισσότερες σελίδες.



Σχήμα 4.9: Το Στρώμα Δεδομένων και οι τεχνολογίες που εμπλέκονται. Ο μόνος τρόπος επικοινωνίας ενός χρήστη με αυτό ειναι μέσω του Στρώματος Λογικής

Ερωτήματα Τα ερωτήματα είναι ο βασικός μηχανισμός των δεδομένων του συστήματος γι'αυτό θα πρέπει να εκτελούνται όσο το δυνατόν πιο αποδοτικά. Κακή απόδοση των απαιτήσεων του συστήματος σε μορφή ερωτημάτων για την βάση δεδομένων μπορεί να προκαλέσει ανάγκες σε υπολογιστικούς όρους οι οποίες να μην είναι πραγματικές. Έτσι, ένα σύστημα που μπορεί να τρέξει σε ένα μηχάνημα, μπορεί να χρειαστεί να μεταφέρει τις

υπολογιστικές του εργασίες σε συστάδα (cluster) υπολογιστών λόγω κακής υλοποίησης των ερωτημάτων. Σε αυτό το κομμάτι έχει δοθεί ιδιαίτερη βαρύτητα και στην υλοποίηση της διπλωματικής μας εργασίας υπό την έννοια ότι χρησιμοποιείται φίλτρο αναζήτησης, το οποίο -ως υπηρεσία- αποτελεί μια από τις πιο απαιτητικές λειτουργίες σε ένα σύστημα τόσο ως προς την πολυπλοκότητα των ερωτημάτων όσο και προς την συχνότητα χρήσης του.

Μορφή Δεδομένων Η μορφή δεδομένων είναι σημαντική ως προς την αξιοποίηση ζητημάτων χώρου σε επίπεδο φυσικού μέσου αποθήκευσης, όσο και προς την βελτιστοποίηση της διαλειτουργικότητας με άλλα στρώματα και εφαρμογές.

Συναλλαγές Συναλλαγή είναι μια κινητή μονάδα εκτέλεσης ενός προγράμματος που προσπελαύνει και πιθανώς ενημερώνει διάφορα δεδομένα. Μια βάση δεδομενων πρεπει να εξασφαλίζει τη σωστή εκτέλεση των συναλλαγών, ανέξαρτητα από απρόσμενα προβλήματα που μπορεί να παρουσιαστούν κατά την διάρκεια της εκτέλεσης. Με άλλα λόγια, η συναλλαγή θα πρέπει να εκτελείτε ολόκληρη ή όχι. Για να διασφαλιστεί η ακεραιότητα των δεδομένων, απαιτείται η βάση δεδομένων να διατηρεί τις εξής ιδιότητες συναλλαγών: Ατομικότητα, Συνέπεια, Απομόνωση και Ανοχή. Αυτές οι ιδιότητες εξασφαλίζονται από το Σύστημα Διαχείρισης Βάσεων Δεδομένων χωρίς την ανάγκη επέμβασης του προγραμματιστή.

Foreign Key Constraints Η χρήση των ξένων κλειδιών στις σχεσιακές βάσεις δεδομένων βοηθά στην ορθή ανάπτυξη και λειτουργία της βάσης δεδομένων, διασφαλίζοντας τις σχέσεις μεταξύ των διαφόρων πινάκων της βάσης δεδομένων όπως έχουν οριστεί από τον αρχικό σχεδιασμό του συστήματος. Γι'αυτό τον λόγο και χρησιμοποιήσαμε την μηχανή InnoDB στα tables της βάσης δεδομένων μας, επειδή ακριβώς υποστηρίζει την χρήση των Foreign Key Constraints.

Στην εφαρμογή μας υλοποιούμε το Στρώμα Δεδομένων με το πιο διαδεδομένο Σύστημα Διαχείρισης Δεδομένων ανοιχτού κώδικα, την MySQL. Πέρα απ'το ότι η MySQL είναι ανοικτού κώδικα, την επιλέγουμε διότι είναι συμβατή με όλες σχεδόν τις υπολογιστικές πλατφόρμες είναι και εύκολη η δημιουργία εφαρμογών λόγω της πληθώρας των διεπαφών που παρέχει για όλες τις δημοφιλείς γλώσσες προγραμματισμού. Ένα ακόμα βασικό πλεονέκτημα της MySQL είναι η ιδιαίτερα μεγάλη της ταχύτητα για εφαρμογές που έχουν να κάνουν με λειτουργίες που αφορούν περισσότερο αναγνώσεις παρά εγγραφές όπως η δική μας. Έτσι, και λαμβάνοντας υπ'όψη την εύκολη διαλειτουργικότητα που παρέχει και με τις υπόλοιπες τεχνολογίες που επιλέξαμε μέχρι τώρα, δηλαδή την PHP, τον Apache Web Server, τον Apache Tomcat Server, και την Java.

4.2 Υλοποίηση Συστήματος

Έχοντας καθορίσει τις βασικές λειτουργίες του κάθε στρώματος όπως τις υλοποιήσαμε, σε αυτό το τμήμα του Κεφαλαίου γίνεται η ανάλυση της υλοποίησης του συστήματός μας. Ξεκινάμε με την περιγραφή του χαμηλότερου στρώματος της αρχιτεκτονικής μας, το Στρώμα Δεδομένων, και προχωράμε στα υψηλότερα στρώματα. Τέλος, περιγράφουμε την υλοποίηση του συστήματος συστάσεών μας.

4.2.1 Στρώμα Δεδομένων

Το Στρώμα Δεδομένων στα πλαίσια της υλοποίησής μας αποτελείται από μια βάση δεδομένων στο Σύστημα Διαχείρισης Βάσεων Δεδομένων MySQL, για τους λόγους τους οποίους αναφέραμε στο κεφάλαιο 4.1.4.3. Αρχικά, εξετάζουμε την τεχνική προσέγγιση της βάσης δεδομένων και στην συνέχεια αναλύουμε τους πίνακες και τα πεδία τα οποία υλοποιήσαμε στην βάση δεδομένων για τις ανάγκες του συστήματός μας.

Γενικά τεχνικά χαρακτηριστικά

Η κωδικοποίηση που επιλέχθηκε για την βαση δεδομένων είναι η latin_swedish_ci, η οποία αποτελεί και την default κωδικοποίηση της βάσης δεδομένων στην MySQL. Ο λόγος που αφήσαμε την κωδικοποίηση ώς έχει, είναι ότι το σύστημα είναι γραμμένο σε Αγγλικά και δεν υπάρχει πρόβλημα κωδικοποίησης όλων των χαρακτήρων στις σελίδες του συστήματος.

Μηχανή βάσης δεδομένων ονομάζεται το λογισμικό διαχείρισης μιας βάσης δεδομένων. Στην MySQL, οι δύο βασικότερες μηχανές που χρησιμοποιούνται λόγω της γενικής εφαρμογής των χαρακτηριστικών τους είναι η InnoDB και η MyIsam. Παρά τις ομοιότητες που φαίνεται να έχουν κατά την μελέτη τους, η InnoDB προσφέρει κάποια συγκεκριμένα χαρακτηριστικά στην υλοποίησή της που εξυπηρετούν τις ανάγκες του συστήματός μας. Πρώτον, θέλουμε η βάση μας να υποστηρίζει foreign key constraints για λόγους που αναλύσαμε παραπάνω. Στην συνέχεια, θέλουμε μέγιστη απόδοση για επεξεργασία μεγάλου μεγέθους δεδομένων. Όπως θα δούμε και στην υλοποίηση του συστήματος συστάσεων, διαχειριζόμαστε μεγάλα δεδομένα τα οποία αποτελούν σημείο αναφοράς για την επίδοση του συστήματός μας. Αν και η απόδοση του συστήματος συστάσεων εξαρτάται κυρίως απ'την αλγοριθμική υλοποίησή του, δεν παύει να ισχύει το γεγονός ότι έχουμε να δουλέψουμε με μεγάλα δεδομένα, των οποίων οι λειτουργίες -τουλάχιστον- της ανάγνωσης είναι συνεχείς δημιουργώντας απαιτήσεις για επιπλέον ταχύτητα στις λειτουργίες μας στην βάση δεδομένων. Τέλος, η InnoDB υποστηρίζει συναλλαγές με τα χαρακτηριστικά τα οποία περιγράψαμε στο κεφάλαιο 4.1. Αντιθέτως, η MyIsam μηχανή δεν υποστηρίζει αυτά τα σημεία – κλειδιά ενώ παράλληλα διακρίνονται θετικά στοιχεία που δεν αφορούν την δική μας υλοποίηση όπως για παράδειγμα η υποστήριξη αναζήτησης πλήρους κειμένου (fulltext search).

Σχεδίαση και υλοποίηση βάσης δεδομένων

Μετά την ανάλυση των βασικών οντοτήτων που συμμετέχουν στο σύστημά μας καθώς και την περιγραφή των τεχνικών διαγραφών, προχωράμε στην περιγραφή της σχεδίασης και της υλοποίησης της βάσης δεδομένων του συστήματός μας. Βάσει της ανάλυσης των οντοτήτων που έγινε στο πρώτο κεφάλαιο, οι βασικές έννοιες του συστήματός μας είναι:

- Χρήστης (User)
- Ταξιδιωτικό Γραφείο (Travel Agency)
- Ταξιδιωτικό Πακέτο (Travel Package)

Αυτές είναι και οι βασικές οντότητες των δεδομένων μας, οι οποίες συμπληρώνονται από τις εξής δευτερεύουσες οντότητες οι οποίες είναι αναγκαίες για την ολοκληρωμένη λειτουργικότητα του συστήματός μας:

- Προορισμοί (Destination – Cities)

Οι οντότητες αυτές είναι απλές ώστε να περιγραφούν σε ένα μόνο πίνακα η κάθε μια, η περιγραφή των οποίων γίνεται παρακάτω, όπου σε παρένθεση η ονομασία του αντίστοιχου πίνακα στο ΣΔΒΔ όπως τον υλοποιήσαμε ή του πεδίου ενός πίνακα:

Πίνακας Χρήστης (User) Ο πίνακας αυτός διατηρεί όλα τα απαραίτητα στοιχεία για έναν χρήστη. Αρχικά για πρωτεύον κλειδί αυτού του πίνακα είχε χρησιμοποιηθεί ένας μοναδικός στον πίνακα αριθμός ως αναγνωριστικό (id). Στην συνέχεια όμως, επειδή ο χρήστης για είσοδό του στο σύστημα χρησιμοποιεί ως αναγνωριστικό την διεύθυνση του ηλεκτρονικού του ταχυδρομείου, και καθώς κάθε τέτοια διεύθυνση (email) είναι μοναδική στο διαδίκτυο, οδήγησε στην χρησιμοποίησή του ως πρωτεύον κλειδί. Τα πεδία αυτού του πίνακα έχουν ως εξής:

1. Αριθμός Χρήστη (**userID**) Ο αριθμός χρήστη είναι ένας ακέραιος αριθμός ο οποίος προσδιορίζει μοναδικά τον κάθε χρήστη. Ο Τύπος του είναι bigint(20) καλύπτοντας τις ανάγκες μας για χρήση του συστήματος απο έναν μεγάλο αριθμό χρηστών. Χρησιμοποιείται ως ένα ακόμα πεδίο για τον προσδιορισμό των χρηστών στις σχέσεις του πίνακα με τις υπόλοιπες οντότητες παρέχοντας ευκολία ως προς την διεπαφή με τα υπόλοιπα στρώματα σε επίπεδο υλοποίησης, χωρίς να αναιρεί την χρήση του email ως πρωτευόντος κλειδιού.

2. Διεύθυνση ηλεκτρονικού ταχυδρομείου (**email**) Το email αποτελεί το πρωτεύον κλειδί του συγκεκριμένου πίνακα και είναι ένα string με το πολύ 100 χαρακτήρες για να είμαστε σίγουροι ότι οποιοδήποτε μήκος ηλεκτρονικής διεύθυνσης μπορεί να καταχωρηθεί στην βάση μας. Τύπος του στην βάση μας είναι Char().

3. Αρχικό Όνομα (**FirstName**) Το FirstName αποτελεί το πεδίο για την καταχώρηση του μικρού ονόματος ενός χρήστη και είναι ένα string που καταχωρείται ως VarChar(), με μέγιστο μήκος 45. Δεν είναι υποχρεωτικό πεδίο, ούτε κατά την εγγραφή επομένως μπορεί να είναι και Null.

4. Δευτερεύων Όνομα (**LastName**) Το LastName συμπληρώνει το FirstName ως προς τις ανάγκες καταχώρησης του ονόματος ενός χρήστη και έχει τα ίδια χαρακτηριστικά καταχώρησης στην βάση δεδομένων όπως το FirstName.

5. Ηλικία (**Age**) Το Age αφορά την ηλικία του χρήστη, που αποτελεί και αυτό μη υποχρεωτικό πεδίο (Null) ενώ καταχωρείται ως smallint(6) στην βάση μας, μιας και η συγκεκριμένη τιμή δεν μπορεί να πάρει μεγάλες τιμές.

6. Φύλο (**Gender**) Το Gender αφορά το φύλο του χρήστη. Είναι ένα string το οποίο υλοποιείται ως Char() στην βάση μας, και είναι επίσης μή υποχρεωτικό (Null).

7. Κωδικός (**password**) Το password είναι ένα string που αποτελείται από το

πολύ 128 χαρακτήρες με την λογική ότι το πεδίο αυτό κωδικοποιείται σε ένα String μεγάλου μήκους, για τις ανάγκες της ασφάλειας των προσωπικών δεδομένων ενός χρήστη και φυσικά δεν μπορεί να είναι κενό (Not Null).

Ακολουθούν μερικά παραδείγματα εγγραφών όπως αυτά είναι αποθηκευμένα στην βάση δεδομένων μας.

userID	email	FirstName	LastName	Age	Gender	password
100	user100@gmail.com	George	LastName100	33	male	1ZCkdz.fx$5MecCj/QdqSVvG7pIROoM.
101	user101@yahoo.com			51	male	1ZCkdz.fx$5MecCj/QdqSVvG7pIROoM.
102	user102@otenet.gr	FirstName102	LastName102	55	male	1ZCkdz.fx$5MecCj/QdqSVvG7pIROoM.
103	user103@yahoo.com	FirstName103	LastName103	28	male	1ZCkdz.fx$5MecCj/QdqSVvG7pIROoM.
104	user104@yahoo.com	FirstName104	LastName104	22	female	1ZCkdz.fx$5MecCj/QdqSVvG7pIROoM.
105	user105@otenet.gr	FirstName105	LastName105	57	male	1ZCkdz.fx$5MecCj/QdqSVvG7pIROoM.

Να σημειωθεί ότι επειδή τα δεδομένα είναι συνθετικά (μη πραγματικά) όπως θα αναλυθεί και παρακάτω, στο παραπάνω screenshot φαίνεται η χρήση του ίδιου κωδικού πρόσβασης απ'όλους τους χρήστες. Παρ'όλα αυτά, τηρούνται οι προϋποθέσεις κρυπτογράφησης του κωδικού των χρηστών που αποτελεί και την ουσία της υλοποίησής μας.

Πίνακας Ταξιδιωτικό Γραφείο (Agencies) Ο πίνακας αυτός εμπεριέχει όλα τα απαραίτητα στοιχεία για ένα ταξιδιωτικό γραφείο. Ως πρωτεύον κλειδί χρησιμοποιούμε το ένα μοναδικό αναγνωριστικό που είναι ένας ακέραιος αριθμός, ενώ τα πεδία του – σε αντίθεση με τον χρήστη – διακρίνονται σε υποχρεωτικά και μή, για την διασφάλιση της λειτουργικότητας του συστήματός μας. Αναλυτικά, τα πεδία αυτά είναι:

1. Αριθμός Ταξιδιωτικού Γραφείου (**AgencyID**) Το πεδίο αυτό είναι ένας ακέραιος αριθμός και αποθηκεύεται με τον τύπο int(11) της MySQL και αποτελεί το πρωτεύον κλειδί του συγκεκριμένου πίνακα. Ο λόγος για τον οποίο επιλέγουμε ένα μικρότερου μεγέθους πεδίο να απαριθμεί το σύνολο των μοναδικών αυτών αναγνωριστικών σε σχέση με το αντίστοιχο πεδίο των χρηστών, είναι ότι τα ταξιδιωτικά γραφεία σε αριθμό είναι πολύ μικρότερα των χρηστών. Προκύπτει ούτως ή άλλως από τα δεδομένα του πραγματικού κόσμου, όπου σε ένα ταξιδιωτικό γραφείο αντιστοιχούν πολλοί χρήστες.

2. Όνομα γραφείου (**AgencyName**) Το όνομα του ταξιδιωτικού γραφείου αντιστοιχεί στο εμπορικό όνομα της εκάστοτε επιχείρησης τουρισμού που διαχειρίζεται ταξιδιωτικά πακέτα. Είναι υποχρεωτικό πεδίο (Not Null) και αποθηκεύεται ως string με χρήση του τύπου varchar(), με μέγιστο αριθμό χαρακτήρων, 45. Το μέγεθος των χαρακτήρων του ονόματος δικαιολογείται από το γεγονός ότι θέλουμε να λάβουμε υπόψη οποιαδήποτε περίπτωση ονομασίας σε παγκόσμια κλίμακα, ενώ παράλληλα ο τύπος varchar() μας εξασφαλίζει ότι στο πεδίο της εγγραφής, δεν θα δεσμευτεί παραπάνω χώρος από αυτόν που πραγματικά χρειάζεται για την αποθήκευση του ονόματος.

3. Διεύθυνση ηλεκτρονικού ταχυδρομείου (**email**) Η διεύθυνση ηλεκτρονικού ταχυδρομείου είναι επίσης υποχρεωτική, αποτελεί στοιχείο εισόδου στο σύστημα από την πλευρά του ταξιδιωτικού γραφείου και αποθηκεύεται ως ένα string το πολύ 100 χαρακτήρων με με τον τύπο varchar της MySQL.

4. Ενιαίος Εντοπιστής Πόρων (**Url**) Το πεδίο αυτό αφορά την ηλεκτρονική διεύθυνση του ιστότοπου του εκάστοτε ταξιδιωτικού γραφείου. Υλοποιείται ως ένα string το πολύ 45 χαρακτήρων, δεδομένων των μεγεθών ενός πραγματικού url, και αποθηκεύεται με τον τύπο δεδομένων varchar(). Το πεδίο αυτό δεν είναι υποχρεωτικό καθώς απλά συνεισφέρει στην πλήρη περιγραφή του προφίλ ενός γραφείου.

5. Αριθμός τηλεφώνου (**phoneNo**) Το πεδίο αυτό μας δίνει τον αριθμό τηλεφώνου ενός γραφείου, εφόσον μας επιτρέψει να το έχουμε διαθέσιμο, μιας και δεν είναι υποχρεωτικό πεδίο και υλοποιείται ξανά ως ένα string με τον τύπο δεδομένων varchar().

6. Αξιολόγηση ταξιδιωτικού (**score**) Στο συγκεκριμένο πεδίο αποθηκεύεται η συνολική αξιολόγηση ενός ταξιδιωτικού γραφείου όπως έχει δοθεί από τους χρήστες μέχρι τώρα. Είναι ένας απλός μέσος όρος των αξιολογήσεων που του έχουν δοθεί και κινείται στην κλίμακα από 1 έως 10, παίρνοντας και δεκαδικές τιμές. Γι'αυτό τον λόγο αποθηκεύεται στην βάση μας ώς float.

7. Κωδικός (**password**) Όπως και στην περίπτωση του χρήστη, το συγκεκριμένο πεδίο έχει τον κωδικό πρόσβασης του ταξιδιωτικού γραφείου στο σύστημά μας, είναι υποχρεωτικό, αποτελεί ένα string 128 χαρακτήρων και το οποίο έχει κρυπτογραφηθεί για την ασφάλεια των προσωπικών δεδομένων του εκάστοτε γραφείου.

8. Φωτογραφία προφίλ (**avatar**) Η φωτογραφία προφίλ αφορά την φωτογραφία που επιλέγει το κάθε ταξιδιωτικό γραφείο να φαίνεται ως αναγνωριστικό της επιχείρησής του, γνωστό ως λογότυπο. Αποθηκεύεται στην βάση δεδομένων ως ένα directory στον χώρο του server στην προσπάθειά μας να μην επιβαρύνουμε την βάση δεδομένων με τον χειρισμό αρχείων μεγάλου όγκου, όπως συνηθίζουν να είναι οι φωτογραφίες. Γι'αυτό τον λόγο αποθηκεύεται ως string με τον τύπο varchar της MySQL όπου το string αυτό ακολουθεί την δομή:

   /public_html/php/pictures/logos/{logoName.fileExtenstion}

   όπου το logoName είναι η ονομασία του αρχείου που γίνεται upload απ'το ταξιδιωτικό γραφείο και το fileExtension μπορεί να είναι οτιδήποτε απ'τις διαθέσιμες κωδικοποιήσεις εικόνων. Το πεδίο δεν είναι υποχρεωτικό και για λόγους ομοιομορφίας στην παρουσίαση των πληροφοριών στον χρήστη, το σύστημα αναλαμβάνει την προσθήκη μιας default φωτογραφίας προφίλ σε περίπτωση που δεν έχει οριστεί από τον χρήστη.

9. Περιοχή Γραφείου (**CityID**) Το πεδίο αφορά την περιοχή στην οποία βρίσκεται η έδρα του γραφείου. Αποθηκεύεται ως ένας ακέραιος αριθμός με τον τύπο int που αποτελεί το ξένο κλειδί της οντότητας των Προορισμών που θα δούμε παρακάτω. Το πεδίο αυτό είναι υποχρεωτικό, για την διασφάλιση της λειτουργικότητας του συστήματός μας.

Ακολουθούν μερικά παραδείγματα εγγραφών ταξιδιωτικών γραφείων όπως αυτά είναι αποθηκευμένα στην βάση δεδομένων μας. Η παρατήρηση για τους κωδικούς πρόσβασης είναι ίδια με αυτή στο αντίστοιχο πεδίο των χρηστών, μιας και τα δεδομένα όλου του συστήματός μας είναι συνθετικά. Επίσης, οι εικόνες προφιλ φαίνεται να είναι ίδιες, αλλά κάτι τέτοιο δεν ισχύει καθώς οι εγγραφές είναι απλά διατεταγμένες ως προς αυτό το πεδίο, όπως φαίνεται

ξεκάθαρα και στο screenshot.

AgencyID	AgencyName	email	Url	phoneNo	score	password
190	AgencyNo190	AgencyNo190@ag190domain.eu	www.agencyNo190.eu	5736928	1.36056	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
216	AgencyNo216	AgencyNo216@ag216domain.eu	www.agencyNo216.eu	2110769	1.20833	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
116	AgencyNo116	AgencyNo116@ag116domain.eu	www.agencyNo116.eu	8493679	1.12654	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
118	AgencyNo118	AgencyNo118@ag118domain.eu	www.agencyNo118.eu	8439010	5.47291	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
125	AgencyNo125	AgencyNo125@ag125domain.eu	www.agencyNo125.eu	7060988	5.80627	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
79	AgencyNo79	AgencyNo79@ag79domain.eu	www.agencyNo79.eu	2662177	5.48103	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
272	AgencyNo272	AgencyNo272@ag272domain.eu	www.agencyNo272.eu	3044057	4.53022	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
183	AgencyNo183	AgencyNo183@ag183domain.eu	www.agencyNo183.eu	5105038	1.00461	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
56	AgencyNo56	AgencyNo56@ag56domain.eu	www.agencyNo56.eu	9347576	3.55977	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.
287	AgencyNo287	AgencyNo287@ag287domain.eu	www.agencyNo287.eu	3277239	6.78733	1ZCkdz.fx$5MecCj/QdqSVvG7plROoM.

avatar ▲ 1	CityID
pictures/logos/logoN1.jpg	1671
pictures/logos/logoN1.jpg	756
pictures/logos/logoN1.jpg	1587
pictures/logos/logoN1.jpg	1331
pictures/logos/logoN1.jpg	199
pictures/logos/logoN1.jpg	12
pictures/logos/logoN1.jpg	41
pictures/logos/logoN1.jpg	783
pictures/logos/logoN1.jpg	1152
pictures/logos/logoN1.jpg	1662

Πίνακας Ταξιδιωτικό Πακέτο (TravelPackages) Ο πίνακας αυτός εμπεριέχει όλα τα ταξιδιωτικά πακέτα, όπως τα ανεβάζουν τα ταξιδιωτικά γραφεία. Το πρωτεύον κλειδί είναι και εδώ ένας μοναδικός ακέραιος αριθμός, ενώ πολλά από τα πεδία είναι υποχρεωτικά διότι μέσω αυτών εξασφαλίζεται η λειτουργικότητα του συστήματός μας, όπως είναι η αναζήτηση των πακέτων. Αναλυτικά, τα πεδία του πίνακα είναι τα εξής:

1. Αναγνωριστικό Πακέτου (**PackageID**) Το πεδίο αυτό αφορά το πρωτεύον κλειδί αυτού του πίνακα, και είναι ένας integer, πιο αποθηκεύεται με τον τύπο δεδομένων int της MySQL.

2. Ημερομηνία Έναρξης (**StartDate**) Αφορά την ημερομηνία έναρξης του ταξιδιωτικού πακέτου. Αποθηκεύεται με την χρήση του τύπου date και είναι υποχρεωτικό. Έχει την μορφή YYYY-MM-DD, και αναλαμβάνουμε την υποχρέωση της μετατροπής του σε DD-MM-YYYY, στο Στρώμα Λογικής. Εννοιολογικά, η ημερομηνία έναρξης του ταξιδιωτικού πακέτου συμπίπτει με την ημερομηνία αναχώρησης για τον ταξιδιώτη, και το διευκρινίζουμε καθώς θα μπορούσε ένα πακέτο να έχει ισχύ σε ένα ευρύτερο χρονικό διάστημα, στη διάρκεια του οποίου να πραγματοποιούνταν πολλές εκδρομές.

3. Ημερομηνία Λήξης (**EndDate**) Αφορά την ημερομηνία λήξης του ταξιδιωτικού πακέτου και έχει ακριβώς τα ίδια χαρακτηριστικά με το πεδίο StartDate.

4. Αεροπορική Εταιρεία (**Airline Company**) Το πεδίο αυτό αφορά την ταξιδιωτική εταιρεία που αποτέλει το μέσο μεταφοράς στον εκάστοτε προορισμό. Το πεδίο αυτό δεν είναι υποχρεωτικό, και αποθηκεύεται ως ένα string με χρήση του τύπου varchar().

5. Τιμή πακέτου (**StartingPrice**) Το πεδίο αυτό αφορά την τιμή του

πακέτου και είναι υποχρεωτικό. Είναι ένας smallint καθώς μπορεί να αποθηκεύσει όλες τις πιθανές τιμές που πρόκειται να εισαχθούν, βάσει λογικής που προκύπτει απ'τα δεδομένα του πραγματικού κόσμου. Η ονομασία του πεδίου είναι Αρχική Τιμή γιατί στον αρχικό σχεδιασμό υπήρχε η πρόθεση δημιουργίας μιας υπηρεσίας για τα ταξιδιωτικά γραφεία, που θα αφορούσε την δημιουργία προσφορών. Κάτι τέτοιο δεν υλοποιήθηκε διότι επικεντρωθήκαμε στην ουσία του συστήματος που είναι δημιουργία συστάσεων και εργαλείων βελτιστοποίησης των ταξιδιωτικών επιλογών του χρήστη.

6. Περιγραφή πακέτου (**Route Description**) Το πεδίο αποθηκεύει την περιγραφή των διαδρομών και των δραστηριοτήτων που περιλαμβάνει το ταξιδιωτικό γραφείο. Το πεδίο είναι υποχρεωτικό μιας και δεν νοείται ταξιδιωτικό πακέτο χωρίς περιγραφή του περιεχομένου του. Για τις ανάγκες μεγάλων περιγραφών, το πεδίο αυτό υλοποιείται ως τύπος δεδομένων longtext.

7. Επιπλέον Πληροφορίες (**Extras**) Οι επιπλέον πληροφορίες αφορούν τις ανάγκες πληροφοριών στις οποίες ο ταξιδιώτης πρέπει να δώσει ιδιαίτερη σημασία, αν αυτές υπάρχουν. Οι πληροφορίες αυτές μπορεί να περιλαμβάνουν πολιτική γευμάτων, ανάγκη για επιπλέον πληρωμές στα αξιοθέατα που πρόκειται να επισκεφτούν ή όχι και άλλου τέτοιου είδους σημειώσεις. Γι'αυτό τον λόγο τα χειριζόμαστε ως διαφορετικό πεδίο, και χωρίς να είναι υποχρεωτικό.

8. Ξενοδοχείο (**Hotel**) Το πεδίο αυτό αφορά τον χώρο διαμονής των ταξιδιωτών, είναι υποχρεωτικό και υλοποιείται ως string στην βάση μας.

9. Φωτογραφία (**Photos**) Το πεδίο της φωτογραφίας, αφορά την φωτογραφία που συνοδεύει το κάθε ταξιδιωτικό πακέτο και είναι χαρακτηριστική των προορισμών που περιλαμβάνει το πακέτο αυτό. Είναι υποχρεωτικό για λόγους πληρότητας σε επίπεδο πληροφόρησης του χρήστη. Όπως και στις φωτογραφίες προφίλ των χρηστών, αυτές αποθηκεύονται ως directory στον εξυπηρετητή μας με το directory να έχει την δομή:

   /public_html/php/pictures/packages/{fileName.fileExtension}

10. Αναγνωριστικό ταξιδιωτικού γραφείου (AgencyID) Αυτό το πεδίο είναι το ξένο κλειδί, που δείχνει το ταξιδιωτικό γραφείο με το οποίο σχετίζεται το εκάστοτε πακέτο.

Παρακάτω, ακουλουθούν μερικά παραδείγματα εγγραφών από την βάση μας. Για άλλη μια φορά, τα AgencyIDs είναι ίδια λόγω ταξινόμησης των εγγραφών στην βάση δεδομένων.

PackageID	StartDate	EndDate	AirlineCompany	StartingPrice	Extras	RouteDescription
1	2015-01-03	2015-01-09	Aeronautical Academy of Europe	355	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...
2	2015-01-03	2015-01-09	AlbaStar	324	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...
3	2015-01-08	2015-01-16	Helios Airways	426	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...
4	2015-01-04	2015-01-17	Elitellina	1243	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...
5	2015-01-06	2015-01-10	Slovak Air Lines	392	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...
6	2015-01-01	2015-01-07	AirOne Atlantic	412	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...
7	2015-01-11	2015-01-14	Wondair on Demand Aviation	169	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...
8	2015-01-10	2015-01-17	Gendarmerie Belge	520	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...
9	2015-01-12	2015-01-21	Aegean Airlines	649	List of Extras Extra1 Extra2 Extra3 Ex...	Descriptive text for the destination! Here, someon...

Hotel	Photos	AgencyID
Whitbread Hotels	pictures/packages/GR8.jpg	1
Best Western	pictures/packages/GR1.jpg	1
Groupe du louvre	pictures/packages/GR1.jpg	1
Best Western	pictures/packages/GR7.jpg	1
Intercontinental	pictures/packages/GR7.jpg	1
Whitbread Hotels	pictures/packages/GR2.jpg	1
Groupe du louvre	pictures/packages/GR1.jpg	1
Hilton Worldwide	pictures/packages/GR5.jpg	1
Groupe du louvre	pictures/packages/GR4.jpg	1

Πίνακας Προορισμοί (Cities) Ο πίνακας αυτός περιλαμβάνει όλους τους πιθανούς Προορισμούς όπως έχουν προσδιοριστεί απο τα ταξιδιωτικά γραφεία. Το όνομα “cities” που έχει δοθεί στον συγκεκριμένο πίνακα προκύπτει απ'το γεγονός ότι όλοι οι Προορισμοί είναι πόλεις και αυτό προέκυψε απο την παραγωγή των δεδομένων μας, χωρίς βλάβη όμως της γενικότητας και του σεναρίου που προσεγγίζει το σύστημά μας. Τα βασικά στοιχεία αυτών είναι τα εξής:

1. Αναγνωριστικό Πόλης (**c_id**) Το αναγνωριστικό αυτό είναι ένας ακέραιος αριθμός που αποτελεί το πρωτεύον κλειδί του πίνακά μας. Αποθηκεύεται με τον τύπο int της MySQL.

2. Όνομα Πόλης (**Name**) Το όνομα της πόλης. Υλοποιείται με έναν

varchar(), και είναι η ονομασία της πόλης που αντιστοιχεί στον προορισμό και το πεδίο είναι υποχρεωτικό. Το πρόβλημα της ίδιας ονομασίας διαφορετικών πόλεων αντιμετωπίζεται από το πεδίο Country που περιγράφουμε παρακάτω.

3. Γεωγραφικό πλάτος (**latitude**) Το γεωγραφικό πλάτος της γεωγραφικής θέσης της συγκεκριμένης πόλης. Αποθηκεύεται με τον τύπο decimal(10,6) της MySQL λόγω της μορφής των δεδομένων όπως τα αποκτήσαμε από την βάση δεδομένων “Geonames.org” και το πεδίο είναι υποχρεωτικό.

4. Γεωγραφικό μήκος (**longitude**) Το γεωγραφικό μήκος της εκάστοτε πόλης, και αποθηκεύεται με ακριβώς την ίδια λογική που περιγράφεται στο πεδίο latitude.

5. Χώρα (**Country**) Η χώρα στην οποία αντιστοιχεί η κάθε πόλη και το πεδίο είναι υποχρεωτικό. Χρησιμοποιείται για την λύση του προβλήματος της διαφοροποίησης δύο πόλεων με την ίδια ονομασία. Αποθηκεύεται ως string με τον τύπο δεδομένων char της MySQL, μιας και το μέγεθος του πεδίου είναι fixed. Αυτό προκύπτει απ'το γεγονός ότι για την ονομασία των χωρών χρησιμοποιήθηκε το πρότυπο κωδικοποίησης ISO-3166.

Παράδειγμα εγγραφών στον πίνακα δίνεται παρακάτω:

c_id	Name	latitude	longitude ▲ 1	Country
1637	Ponta Delgada	37.733330	-25.666670	PT
398	Los Llanos de Aridane	28.658510	-17.918210	ES
315	Santa Cruz de la Palma	28.683510	-17.764210	ES
1579	Camara de Lobos	32.633330	-16.966670	PT

Σχέσεις μεταξύ πινάκων Μετά τον σχεδιασμό και την αποτύπωση των βασικών οντοτήτων σε πίνακες της βάσης δεδομένων μας, προχωράμε στην διερεύνηση και αποτύπωση των σχέσεων μεταξύ των πινάκων αυτών. Αυτές είναι:

Χρήστης – Βαθμολόγηση – Ταξιδιωτικό Γραφείο (Agency_Rating) Η σχέση αυτή, αποτυπώνει την δυνατότητα των χρηστών να βαθμολογούν ταξιδιωτικά γραφεία σε περίπτωση που έχουν κάνει χρήση κάποιου ταξιδιωτικού τους πακέτου. Επομένως, αφού ένας χρήστης μπορεί να αγοράσει περισσότερων του ενός ταξιδιωτικά πακέτα, και τα ταξιδιωτικά πακέτα μπορούν να αγοραστούν από περισσότερους από έναν χρήστες, προκύπτει άμεσα και η σχέση των χρηστών ως προς την βαθμολόγηση των ταξιδιωτικών γραφείων και αυτή είναι M:N. Από αυτήν ακριβώς την σχέση προκύπτει και η συνολική βαθμολόγηση ενός ταξιδιωτικού γραφείου. Πρέπει να σημειωθεί ότι υπάρχει ένας περιορισμός ο οποίος προκύπτει και απ'την περιγραφή που μόλις κάναμε: Ένας χρήστης μπορεί να βαθμολογήσει ένα ταξιδιωτικό γραφείο αν και μονό αν έχει κάνει χρήση ενός πακέτου του. Αυτό όμως δεν αποτυπώνεται σε επίπεδο βάσης δεδομένων αλλά στο Στρώμα Λογικής όπως θα δούμε παρακάτω. Τα πεδία της σχέσης αυτής είναι:

1. Αναγνωριστικό βαθμολογησης (**RatingID**) Το αναγνωριστικό κάθε εγγραφής που έχει να κάνει με την βαθμολόγηση ενός χρήστη προς ένα ταξιδιωτικό γραφείο ανά εγγραφή, και αποθηκεύεται ως integer.

2. Αναγνωριστικό γραφείου (**AgencyID**) Το ξένο κλειδί που δείχνει το αντικείμενο της οντότητας Ταξιδιωτικό Γραφείο που συμμετέχει στην σχέση αυτή για κάθε εγγραφή.

3. Αναγνωριστικό χρήστη (**UserID**) Το ξένο κλειδί που δείχνει το αντικείμενο της οντότητας Χρήστης που συμμετέχει στην σχέση αυτή για κάθε εγγραφή.

4. Βαθμολόγηση (**Score**) Η βαθμολόγηση που καταχωρεί ο κάθε χρήστης για κάθε ταξιδιωτικό γραφείο που αξιολογεί. Το πεδίο αυτό είναι ένας ακέραιος αριθμός που παίρνει τιμές από 1- 10, και αποθηκεύεται ως int() στην βάση δεδομένων μας.

Παράδειγμα εγγραφών στην βάση μας:

RatingID	AgencyID	UserID	Score
1	500	1	8
2	303	1	10
3	500	1	8
4	500	1	6
5	500	1	4

Χρήστης – Αγορά – Ταξιδιωτικό Πακέτο (Cart) Η σχέση αυτή, αποτυπώνει την σχέση των χρηστών ως προς τα ταξιδιωτικά πακέτα που πρόκειται να αγοράσουν. Έτσι, για άλλη μια φορά η σχέση αυτή είναι N:M μιας και ένας ταξιδιώτης μπορεί να επιλέξει την αγορά περισσότερων του ενός ταξιδιωτικών πακέτων και ένα ταξιδιωτικό πακέτο μπορεί να αγοραστεί από περισσότερους του ενός Χρήστες. Δεν τίθεται κάποιος περιορισμός, διότι δεν υπάρχει κάποιο κριτήριο διαθεσιμότητας των ταξιδιωτικών πακέτων. Επομένως, τα πεδία της σχέσης αυτής είναι:

1. Αναγνωριστικό Εγγραφής (**item_id**) Το αναγνωριστικό της εγγραφής έχει μικρή εννοιολογική σημασία και χρησιμοποιείται απλά σαν μοναδικό αναγνωριστικό στις εγγραφές αυτής της σχέσης. Αποθηκεύεται σαν integer.

2. Αναγνωριστικό Ταξιδιωτικού Πακέτου (**package_id**) Το ξένο κλειδί του ταξιδιωτικού πακέτου που συμμετέχει στην σχέση για την εκάστοτε εγγραφή.

3. Αναγνωριστικό Ταξιδιωτικού Γραφείου (**agency_id**) Το αναγνωριστικό του ταξιδιωτικού γραφείου στο οποίο ανήκει το προς αγορά ταξιδιωτικό πακέτο. Το συγκεκριμένο πεδίο δεν παίζει ρόλο στην σχέση, απλά καταγράφεται κατά την επιλογή ενός πακέτου για αγορά, για διευκόλυνση των υπηρεσιών αξιολόγησης στο Στρώμα Λογικής. Αυτό το πεδίο θα μπορούσε να αποτελεί ξένο κλειδί σε μια σχέση της μορφής Χρήστης- Αγορά -Ταξιδιωτικό Γραφείο, η έλλειψη της οποίας όμως, διαπιστώθηκε στην πορεία της ανάπτυξης. Επομένως, αποτελεί ένα βοηθητικό πεδίο για την προγραμματιστική υλοποίηση των αξιολογήσεων στο Στρώμα Λογικής χωρίς κάποια εννοιολογική σημασία.

4. Αναγνωριστικό Χρήστη (**user_id**) Το ξένο κλειδί του χρήστη που συμμετέχει σε αυτή την σχέση για την εκάστοτε εγγραφή που αφορά την τοποθέτηση ενός πακέτου στο καλάθι αγορών αυτού του χρήστη.

Ακολουθεί παράδειγμα εγγραφών στην βάση δεδομένων μας για την συγκεκριμένη σχέση:

item_id	package_id	agency_id	user_id
1	6000	500	user1@hotmail.com
2	5998	500	user1@hotmail.com

Ταξιδιωτικά Πακέτα – Περιέχουν – Προορισμούς (destinationsIndex) Στην σχέση αυτή συμμετέχουν τα ταξιδιωτικά πακέτα και οι προορισμοί. Ένα ταξιδιωτικό πακέτο εμπεριέχει πολλούς ταξιδιωτικούς προορισμούς ενώ και οι προορισμοί εμπεριέχονται σε πολλά ταξιδιωτικά πακέτα. Επομένως η σχέση μας είναι πάλι M:N. Τα πεδία αυτής της σχέσης είναι:

1. Αναγνωριστικό εγγραφής (**dest_id**) Το αναγνωριστικό της εγγραφής δεν έχει κάποια εννοιολογική σημασία και πάλι, αλλά αποτελεί το πρωτεύον κλειδί αυτής της σχέσης. Το αναγνωριστικό αυτό είναι ένας ακέραιος αριθμός.

2. Αναγνωριστικό Ταξιδιωτικού Πακέτου (**PackageID**) Το αναγνωριστικό αυτό είναι το ξένο κλειδί του Ταξιδιωτικού Πακέτου που συμμετέχει σε αυτή την σχέση για κάθε προορισμό με τον οποίο συνδέεται σε ξεχωριστές εγγραφές κάθε φορά.

3. Αναγνωριστικό Πόλης (**CityID**) Το αναγνωριστικό αυτό είναι το ξένο κλειδί των Προορισμών – Πόλεων που συμμετέχουν σε αυτή την σχέση για κάθε ταξιδιωτικό πακέτο με στο οποίο περιέχονται, ανά εγγραφή.

Παραδείγματα εγγραφών απ'την βάση μας:

dest_id ▲ 1	PackageID	CityID
14485	6001	706
14484	6000	334
14483	6000	1570
14482	6000	1544

Προορισμοί – Προτείνονται – Χρήστες (Recommendations) Η σχέση αυτή, περιλαμβάνει τα recommendations που αντιστοιχούν σε κάθε χρήστη. Έτσι, και σύμφωνα με την μέχρι τώρα ανάλυσή μας, κάθε φορά που ο χρήστης συνδέεται και πλοηγείται στην σελίδα της αναζήτησης, υπολογίζονται κάποιοι προορισμοί οι οποίοι προτείνονται στον χρήστη σε μορφή πακέτου. Η σχέση είναι M:N καθώς όλοι οι προορισμοί προτείνονται σε πολλούς χρήστες και στους χρήστες συστήνονται παραπάνω από ένας προορισμοί. Τα πεδία της σχέσης είναι τα εξής:

1. Αναγνωριστικό σύστασης (**RecID**) Αποτελεί το αναγνωριστικό της σύστασης στην συγκεκριμένη σχέση και άρα είναι το πρωτεύον κλειδί, είναι ένας μοναδικός ακέραιος και αποθηκεύεται με τον τύπο int της MySQL.

2. Αναγνωριστικό χρήστη (**UserID**) Αποτελεί το ξένο κλειδί του χρήστη που συμμετέχει με κάποιον προορισμό στην συγκεκριμένη σχέση.

3. Αναγνωριστικό προορισμοού (**CityID**) Αποτελεί το ξένο κλειδί του προορισμού που συμμετέχει με κάποιον χρήστη στην συγκεκριμένη σχέση.

Ακολουθεί παράδειγμα εγγραφών από την βάση δεδομένων μας:

RecID	UserID	CityID
1	1	792
2	1	782
3	1	602

Χρήστες – Αξιολογούν – Προορισμούς (UserRatings) Στην σχέση αυτή, αποτυπώνεται η διαδικασία της αξιολόγησης των προορισμών από τους χρήστες. Έτσι, όλοι οι χρήστες μπορούν να αξιολογήσουν όλους τους προορισμούς και όλοι οι προορισμοί μπορούν να αξιολογηθούν από όλους τους χρήστες, επομένως η σχέση μας είναι ξανά M:N. Εδώ υπάρχει ένας μόνο περιορισμός ο οποίος αφορά την προϋπόθεση για να μπορεί ένας χρήστης να βαθμολογήσει έναν προορισμό. Αυτή η προϋπόθεση δεν είναι άλλη από το ότι ο χρήστης πρέπει να έχει ταξιδέψει στον προορισμό που πρόκειται να αξιολογήσει. Αυτή η προϋπόθεση όμως δεν λαμβάνεται υπόψη στην βάση δεδομένων αλλά στο Στρώμα Λογικής όπου διασφαλίζεται ότι αφού αγοράσει ένας χρήστης ένα πακέτο, τότε και μόνο τότε του ζητείται αξιολόγηση των προορισμών που περιέχει το πακέτο αυτό. Τα πεδία της σχέσης αυτής είναι:

1. Αναγνωριστικό αξιολόγησης (**PrefID**) Το πρωτεύον κλειδί της σχέσης, που είναι ένας ακέραιος αριθμός, τύπου int().

2. Αναγνωριστικό χρήστη (**UserID**) Το ξένο κλειδί του χρήστη που συμμετέχει στην σχέση με κάποιον προορισμό ανά εγγραφή.

3. Αναγνωριστικό προορισμού (**CityID**) Το ξένο κλειδί του προορισμού που συμμετέχει στην σχέση με κάποιον χρήστη ανά εγγραφή.

4. Βαθμολόγηση (**Preference**) Το πεδίο που αφορά την βαθμολόγηση που έχει δοθεί από έναν χρήστη σε κάθε προορισμό. Η τιμή του πεδίου αυτού μπορεί να είναι μια από τις τιμές 1 – 5, και υλοποιείται με τον τύπο δεδομένων int() της MySQL. Αυτή η τιμή χρησιμοποιείται από το σύστημα συστάσεων για ανάλυση και τελική σύσταση πακέτων προς τον χρήστη.

Ακολουθεί παράδειγμα μερικών εγγραφών αυτής της σχέσης:

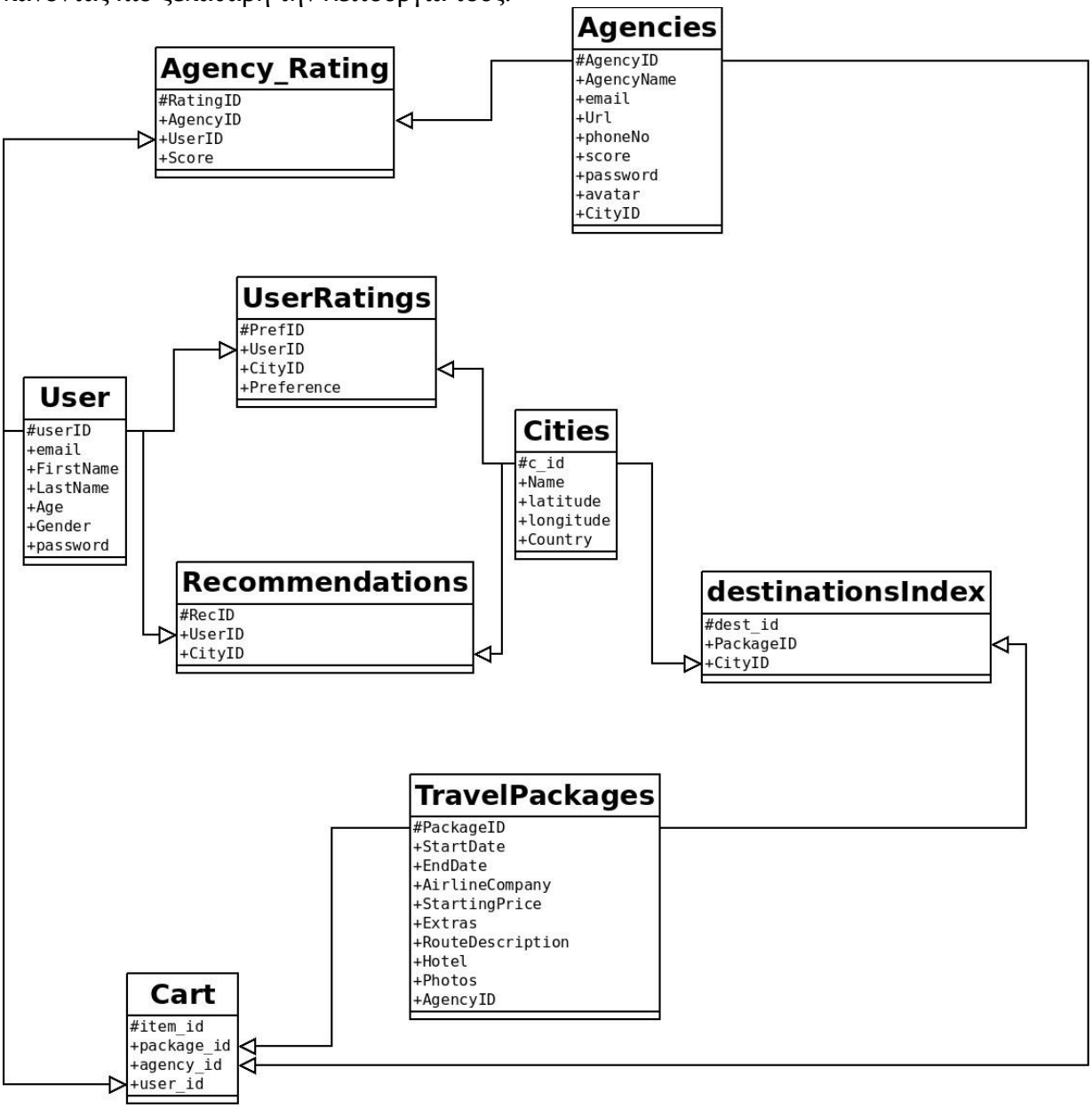
PrefID ▲ 1	UserID	CityID	Preference
1	1	1	5
2	1	2	3
3	1	3	4
4	1	4	3

Η παρουσίαση του Στρώματος Δεδομένων ολοκληρώνεται με την συνολική παρουσίαση των πινάκων στο σχήμα 4.10.

4.2.2 Στρώμα Λογικής

Το Στρώμα Λογικής αποτελεί την καρδιά του συστήματος καθώς αναλαμβάνει την υλοποίηση όλων των λειτουργιών του συστήματος. Αυτές οι λειτουργίες αφορούν την διασύνδεση των υπόλοιπων στρωμάτων με όλα τα χαρακτηριστικά που περιγράφηκαν στο κομμάτι ανάλυσης των στρωμάτων όλης της αρχιτεκτονικής μας (4.1.4.2). Αυτό που πρέπει να τονιστεί, είναι ότι όλα αυτά τα αρχεία δεν καλούνται από τον χρήστη άμεσα, αλλά καλούνται από τα αρχεία του Στρώματος Πελάτη κατά την χρήση των υπηρεσιών του συστήματός μας.

Ξεκινάμε με την περιγραφή των αρχείων αυτών που συνθέτουν το Στρώμα Λογικής και στην συνέχεια, στο Στρώμα Πελάτη, παρουσιάζουμε την σύνδεση όλων αυτών των κομματιών, κάνοντας πιο ξεκάθαρη την λειτουργία τους.



Σχήμα 4.10: Διάγραμμα οντοτήτων και σχέσεων όπως διαμορφώθηκαν σε πίνακες στην βάση δεδομένων μας

dbc.php Πραγματοποιεί την σύνδεση με την βάση δεδομένων (ΒΔ).

agencyCon.php πραγματοποιεί την σύνδεση (log-in) του χρήστη που σχετίζεται με κάποιον λογαριασμό ταξιδιωτικού γραφείου και γίνεται η κατάλληλη ανάθεση τιμών σε session variables για την χρήση τους σε μετέπειτα στάδιο.

agency.php Γίνονται τα απαραίτητα ερωτήματα στην ΒΔ για τα στοιχεία των ταξιδιωτικών γραφείων, η ανάθεση των τιμών αυτών σε μεταβλητές ενώ λαμβάνονται υπόψη και οι περιπτώσεις κενών πεδίων για προβολή αντίστοιχων μηνυμάτων στον χρήστη για μη διαθεσιμότητα αυτών.

Ακόμα, γίνεται ο χειρισμός των δεδομένων σε περίπτωση αιτήματων για κάποιο update στα στοιχεία του ταξιδιωτικού γραφείου, τόσο σε επίπεδο αποθήκευσης ή ενημέρωσης αυτών στην ΒΔ όσο και σε επίπεδο ελέγχου για εγκυρότητα των στοιχείων – τα αριθμητικά πεδία να είναι αριθμοί, τα urls να είναι έγκυρα, κτλ.

Τέλος, υπάρχουν δύο συναρτήσεις οι refreshURL() και η inputFilter() που χειρίζονται θέματα ανακατεύθυνσης (redirection) και φιλτραρίσματος των δεδομένων για την ασφάλεια του συστήματός μας, αντίστοιχα.

userCon.php Κατ'αναλογία, γίνεται η σύνδεση του χρήστη που είναι εγγεγραμμένος στο σύστημά μας, και η ανάθεση τιμών σε session variables.

user.php Όπως και στο αρχείο agency.php, γίνονται ακριβώς οι ίδιες λειτουργίες χειρισμού δεδομένων του προφίλ, αποθήκευσης αυτών σε μεταβλητές, ελέγχου εγκυρότητας δεδομένων εισαγωγής και χρήση συναρτήσεων για ανακατεύθυνση και φιλτράρισμα δεδομένων από html entities κτλ αλλά απ'την πλευρά του χρήστη.

insert.php Το αρχείο αυτό κάνει εισαγωγή των δεδομένων του χρήστη (ταξιδιώτη) στην βάση δεδομένων, όπως δίνονται στο σύστημα κατά την εγγραφή του (sign-up).

insertAgency.php Κατ'αναλογία με το insert.php, εισάγει τα δεδομένα του ταξιδιωτικού γραφείου στην ΒΔ κατά την εγγραφή του, με τους απαραίτητους ελέγχους των δεδομένων που εισάγονται.

insertPckgHandler.php Εισάγει τα στοιχεία ενός νέου πακέτου στην ΒΔ κάνοντας έλεγχο της εγκυρότητας τους εννοιολογικά όπως κατά την εγγραφή των χρηστών, και παράλληλα ενημερώνει τον πίνακα που αποτυπώνει την σχέση των προορισμών που συνθέτουν ένα ταξιδιωτικό πακέτο, με την αντίστοιχη οντότητα των πόλεων/προορισμών.

Ακόμα, περιέχει την συνάρτηση fileUploading() που χρησιμοποιείται για το ανέβασμα αρχείων εικόνων με τις προδιαγράφές εικόνων προφίλ ως προς τις διαστάσεις τους, το μέγεθός τους και τον τύπο αρχείο τους. Υποστηρίζονται όλες οι γνωστές κωδικοποιήσεις εικόνων.

updatePckg.php Καλείται από το σύστημα, σε περίπτωση που κάποιος χρήστης ταξιδιωτικού γραφείου επιθυμεί να αλλάξει τα στοιχεία ενός ταξιδιωτικού πακέτου που ανήκει στο γραφείο του. Κάνει προβολή των πακέτων σε μορφή λίστας, δίνοντας την διεπαφή στον χρήστη για περαιτέρω επεξεργασία απ'το αρχείο updatePckgHandler.php που βλέπουμε στην συνέχεια. Περιλαμβάνεται και η συνάρτηση tansformDate() που κάνει μετατροπή του format των ημερομηνιών από YYYY-MM-DD όπως είναι αποθηκευμένες στην ΒΔ σε format πιο

εύκολα αναγνώσιμο από τον χρήστη και συμβατό με τα εργαλεία της διεπαφής που είναι το DD-MM-YYYY.

updatePckgHandler.php Καλείται απ'το updatePckg.php για τον χειρισμό της επεξεργασίας των δεδομένων ενός ταξιδιωτικού πακέτου. Πραγματοποιεί όλες τις αλλαγές στα δεδομένα των ταξιδιωτικών πακέτων, ανάλογα με τις επιλογές του χρήστη. Κάνει προβολή των υπαρχόντων δεδομένων ενός πακέτου όπως αυτά είναι αποθηκευμένα στην ΒΔ και στην συνέχεια δίνεται η δυνατότητα επεξεργασίας των στοιχείων αυτών ξεχωριστά για το κάθε πεδίο. Και εδώ γίνεται η χρήση των βοηθητικών συναρτήσεων φιλτραρίσματος των δεδομένων όπως σε κάθε τμήμα του συστήματος που σχετίζεται με εισαγωγή των δεδομένω στην ΒΔ, ενώ παρέχεται εκ νέου η δυνατότητα ανεβάσματος αρχείου για αλλαγή φωτογραφίας προφίλ.

Book.php Το αρχείο αυτό καλείται όταν κάποιος χρήστης κάνει αγορά κάποιου πακέτου. Πραγματοποιεί μια τεχνητή καθυστέρηση για περίπου 5 δευτερόλεπτα και ενημερώνει το σύστημα για τις νέες συνθήκες. Ό,τι δηλαδή ένας χρήστης αγόρασε ένα πακέτου του οποίου τα στοιχεία προς αξιολόγηση πρέπει να σταλούν στον χρήστη. Έτσι, το σύστημα ενημερώνει τον χρήστη για την αξιολόγηση αυτή, και ανακατευθύνει τον χρηστη στην σελίδα αναζήτησης των πακέτων μετά την λήξη της τεχνητής καθυστέρησης.

cartHandler.php Αναλαμβάνει τον χειρισμό των αιτήσεων του χρήστη στην σελίδα του καλαθιού των αγορών. Χειρίζεται την εισαγώγή πακέτων στο καλάθι και την αφαίρεσή τους, ενώ είναι υπεύθυνο και για τους λογικούς ελέγχους που έχουν να κάνουν με επανεπιλογή πακέτων που ήδη υπάρχουν στο καλάθι. Γίνεται χρήση της μετατροπής του format των ημερομηνιών όπως και σε κάθε αρχείο που αναλαμβάνει την προβολή στοιχείων που έχουν να κάνουν με στοιχεία ταξιδιωτικού γραφείου.

displayPckg.php Αναλαμβάνει την προβολή των ταξιδιωτικών πακέτων για το εκάστοτε ταξιδιωτικό γραφείο. Αυτό συμβαίνει είτε όταν ο χρήστης του ταξιδιωτικού γραφείου επιλέξει την προβολή αυτών, είτε όταν εισάγει ένα νέο πακέτο και οπότε το σύστημα καλεί το displayPckg.php για προβολή της ανανεωμένης λίστας των πακέτων αυτών με προβολή κατάλληλου μηνύματος για την επιτυχή καταχώρηση του εκάστοτε πακέτου.

displayPckgToUser.php Καλείται όταν γίνεται προβολή όλων των ταξιδιωτικών πακέτων απ'ολα τα ταξιδιωτικά γραφεία κατά την πλοήγηση του χρήστη-ταξιδιώτη. Πέρα απ'την προβολή των πακέτων γίνεται και χειρισμός των φίλτρων αναζήτησης κατά προορισμό, τοποθεσία του χρήστη (και άρα περιοχής εξυπηρέτησης των ταξιδιωτικών γραφείων), ημερομηνιών αναχώρησης και επιστροφής και κατάταξη αυτών ανάλογα με το κριτήριο επιλογής.

paging.php Κάνει τον χειρισμό του paging της σελίδας προβολής των ταξιδιωτικών πακέτων προς τον χρήστη. Διατηρεί 10 πακέτα ανά σελίδα και χειρίζεται την πλοήγηση προς προηγούμενες ή επόμενες σελίδες που μπορεί να έχουν προκύψει κατά την αναζήτηση.

file_uploading.php Χειρίζεται το ανέβασμα φωτογραφιών από το ταξιδιωτικού γραφείο πέρα από την φωτογραφία προφίλ του. Δημιουργεί τα κατάλληλα directories στον εξυπηρετητή εφόσον χρειάζεται -αν δηλαδή δεν υπάρχουν ήδη- και στην συνέχεια αποθηκεύει τα directories στην ΒΔ.

queryAgencies.php Κάνει λήψη των στοιχείων του ταξιδιωτικού γραφείου όπως είναι στην βάση δεδομένων και τα αναθέτει σε μεταβλητές. Η διαφορά αυτού του αρχείου από το αρχείο

που κάνει περίπου τις ίδιες λειτουργίες με αυτό (agency.php) είναι ότι το queryAgencies.php χρησιμοποιείται στον χώρο πρόσβασης των χρηστών ταξιδιωτών και όχι στον χώρο πρόσβασης των ταξιδιωτικών γραφείων όπως συμβαίνει με το agency.php.

updatePrefs.php Καλείται από την σελίδα αξιολόγησης του κάθε χρήστη για ενημέρωση της ΒΔ σχετικά με τις αξιολογήσεις του ως προς το ταξιδιωτικό γραφείο και τον/τους προορισμό/σμούς.

header.php, headerAG.php, footer.php, footerAG.php Τα αρχεία αυτά καλούνται από τα αρχεία του Στρώματος Πελάτη για την πλήρη δημιουργία της διεπαφής και της λειτουργικότητάς τους. Η ανάλυσή τους είναι πιο ξεκάθαρη στην αρχή του κεφαλαίου 4.2.3 όπου περιγράφουμε το template που διαμορφώνεται από τα αρχεία του Στρώματος Πελάτη. Απλά αναφέρουμε εδώ ότι βασικός τους ρόλος ώς προς την λειτουργικότητα του συστήματος είναι το άνοιγμα και το κλείσιμο συνδέσεων προς την βάση δεδομένων καθώς και η δήλωση των κατάλληλων αρχείων για την εκάστοτε σελίδα – όπως για παράδειγμα το πιο αρχείο της javascript είναι κατάλληλο για την λειτουργικότητα μιας υπηρεσίας που προσφέρει η εκάστοτε ιστοσελίδα.

ratings.php Ελέγχει την βάση δεδομένων για πιθανή ανάγκη αξιολόγησης από τον χρήστη σε περίπτωση που έχει αγοράσει κάποιο πακέτο, στέλνοντας τα αντίστοιχα αιτήματα στο Στρώμα Πελάτη, διαφορετικά εμφανίζει τα κατάλληλα μηνύματα στην evaluate.php.

4.2.3 Στρώμα Πελάτη

Στο Στρώμα Πελάτη περιγράφουμε τα αρχεία που καλούνται άμεσα από τον χρήστη στην πλοήγησή του στο σύστημά μας. Βασική τους λειτουργία είναι η προβολή των σελίδων του συστήματος σε HTML και η κλήση των αναγκαίων αρχείων του Στρώματος Λογικής που περιγράψαμε παραπάνω για την λειτουργικότητα του συστήματός μας. Πριν προχωρήσουμε στην περιγραφή κάθε αρχείου ξεχωριστά, δείχνουμε τις βασικές περιοχές στις οποίες χωρίζεται το template των σελίδων μας. Το template, αν και δεν είχαμε να διαχειριστούμε μεγάλο όγκο πληροφορίας, κρίθηκε αναγκαίο για την απλότητα που προσφέρει στο development κάθε οθόνης του συστήματος. Έτσι, οι σελίδες με κοινές περιοχές βάσει του template δεν κατασκευάζονται από την αρχή, αλλά διαφοροποιούν το περιεχόμενο που χρειάζεται. Παρακάτω, φαίνεται ένα διάγραμμα του template σε μια τυχαία οθόνη που δείχνει ακριβώς αυτόν τον διαχωρισμό. (Σχήμα 4.11)

Σχήμα 4.11: Το template των σελίδων μας όπως φαίνονται στον χρήστη. Η σελίδα διαχωρίζεται σε 4 βασικές περιοχές, το Header, το MainContent, το Side Content και το Footer.

Έτσι, στην περιοχή header τοποθετείται το content που έχει να κάνει με το μενού πλοήγησης του χρήστη το οποίο επαναλαμβάνεται παντού. Σε επίπεδο κώδικα, στην ίδια περιοχή και η αρχικοποίηση όλων κάθε σελίδας όπως η σύνδεση με την βάση, η επιλογή κατάλληλων συνδέσμων προς τα διαθέσιμα javascript αρχεία κτλ. Το ίδιο ισχύει βέβαια και για το header των ταξιδιωτικών γραφείων, μιας και η παραπάνω σελίδα έχει ληφθεί τυχαία απ'το σύνολο οθονών του χρήστη-ταξιδιώτη.

Στην περιοχή side-content, γίνονται διαθέσιμες βοηθητικές πληροφορίες και υπηρεσίες για την λειτουργία κάθε σελίδας. Έτσι, τοποθετούνται υπομενού, φίλτρα αναζήτησης, πληροφορίες για τον χρήστη ή τον προορισμό, συμπληρώνοντας κάθε σελίδα ξεχωριστά ανάλογα με τις ανάγκες της.

Στην περιοχή main-content, τοποθετείται το κυρίως περιεχόμενο της κάθε σελίδας. Είναι αυτό που διαφοροποίειται πιο ουσιαστικά μεταξύ κάθε σελίδας και είναι ακριβώς το σημείο όπου γίνεται κλήση αρχείων του Στρώματος Λογικής για την λήψη δεδομένων απ'την ΒΔ, την καταχώρηση δεδομένων σε αυτή κτλ.

Στην περιοχή footer φαινομενικά δεν αποδίδεται κάποια σημαντική λειτουργικότητα. Από άποψη υλικού στην περίπτωση του συστήματός μας όντως δεν έχει προστεθεί κάτι γιατί δεν υπήρχε αυτή η ανάγκη, αλλά βοηθά στο κλείσιμο του template και του κώδικα κατά ενιαίο τρόπο και παράλληλα εκτελεί την πολύ σημαντική λειτουργία -όπως αναλύσαμε παραπάνω- του να τερματίζει την σύνδεση στις ΒΔ αφού έχει φορτωθεί όλο το απαραίτητο υλικό.

Το ίδιο template ισχύει και για τα ταξιδιωτικά γραφεία, για τα οποία οι περιοχές που περιγράψαμε αναπτύσσονται σε διαφορετικά αρχεία. Αυτό το κάνουμε διότι κατά την ανάπτυξη του συστήματος, λαμβάνουμε υπόψη τον περιορισμό ένας χρήστης-ταξιδιώτης να μην έχει πρόσβαση στην περιοχή του ιστοτόπου για την οποία δεν είναι εξουσιοδοτημένος όπως είναι για παράδειγμα ο χώρος των ταξιδιωτικών γραφείων. Φυσικά, ισχύει και ο αντίστροφος περιορισμός.

Όλος αυτός ο διαχωρισμός γίνεται κατανοητός στο σύστημα μέσω της υλοποίησης του

γραφικού περιβάλλοντος σε css. Με την css ορίσαμε περιοχές οι οποίες έχουν ως βασικό περιτύλιγμα τις περιοχές header, footer, main content και side content και από κει και πέρα η γραφιστική υλοποίηση των περιεχομένων ακολουθεί μια συγκεκριμένη λογική που να είναι ενιαία σε όλον τον ιστότοπο, όπως συμβαίνει σε ένα σαφώς ορισμένο και καλά δομημένο stylesheet και κατ'επέκταση template. Στην συνέχεια ακολουθεί η περιγραφή όλων των υπόλοιπων αρχείων του Στρώματος Πελάτη.

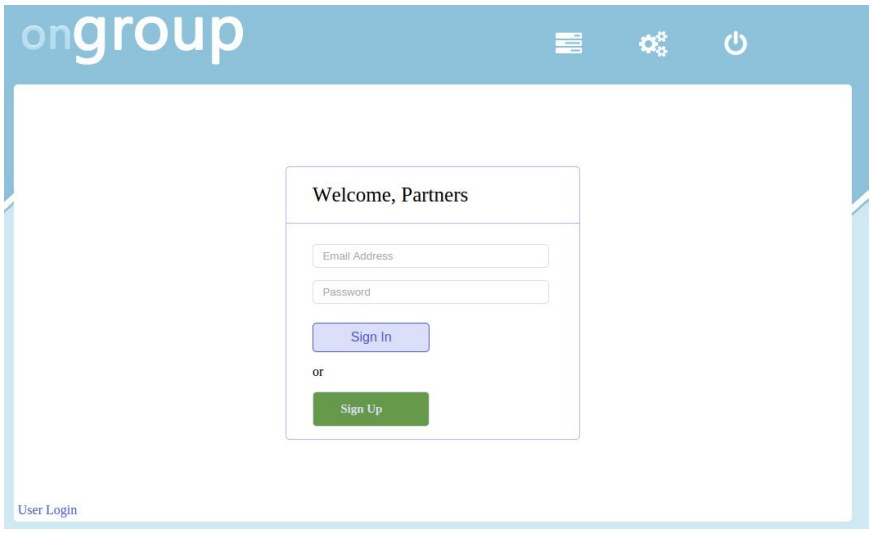
index.php Η αρχική σελίδα του συστήματος όπου ο χρήστης δίνει τα στοιχεία του και κάνει είσοδο. Χρησιμοποιεί το userCon.php για επικοινωνία με την βάση δεδομένων.

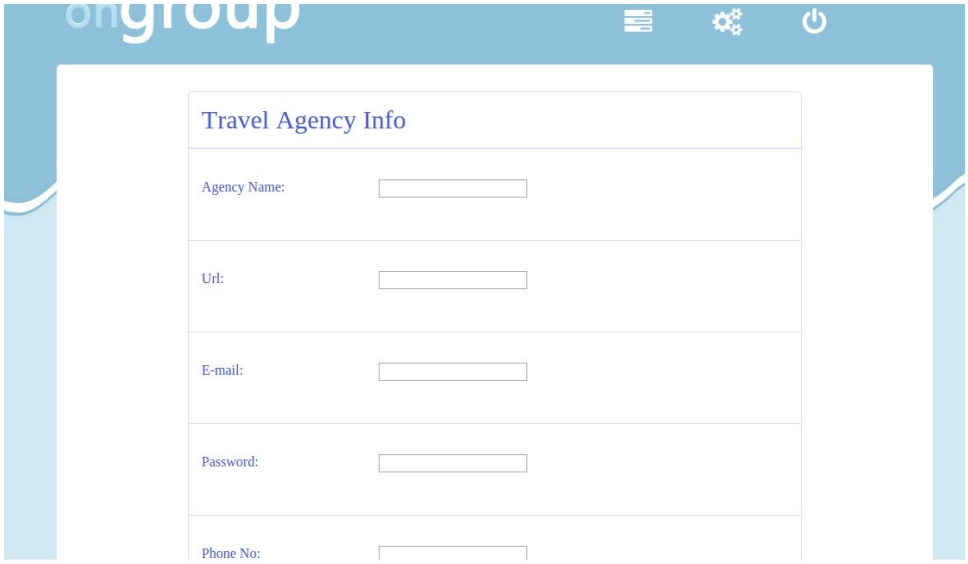
logout.php Ο χρήστης καλεί το αρχείο για έξοδο από το σύστημα. Ακολουθεί screenshot όπου φαίνεται η αρχική σελίδα και η διεπαφή για έξοδο από το σύστημα με το χαρακτηριστικό εικονίδιο on/off και το κατάλληλο μήνυμα επιβεβαίωσης εξόδου.

agencyLogin.php Η διεπαφή για την είσοδο στο σύστημα. Παρέχεται η φόρμα για εισαγωγή στοιχείων και σύνδεση ή η επιλογή για εγγραφή στο σύστημα.

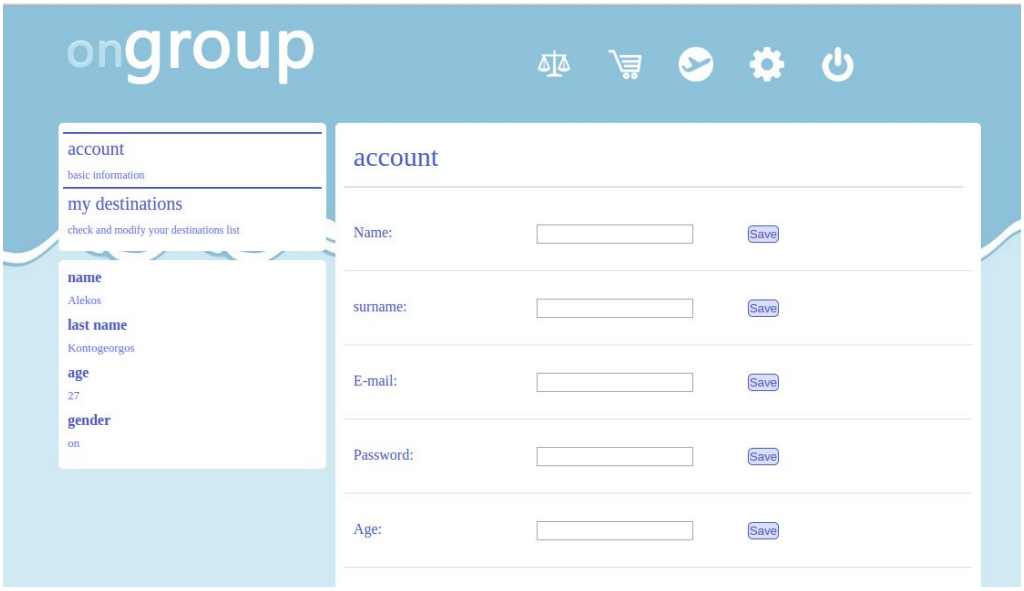
insertAgencyForm.php Η διεπαφή για εγγραφή στο σύστημα για τα ενδιαφερόμενα ταξιδιωτικά γραφεία.

logoutAG.php Η διεπαφή για έξοδο από το σύστημα όσον αφορά τα ταξιδιωτικά γραφεία. Όπως και στους χρήστες, παρέχεται και κατάλληλη πλοήγηση για είσοδος ώς χρήστης-ταξιδιώτης. Ακολουθούν τα screenshot της διεπαφής εισόδου και κομμάτι της διεπαφής για εγγραφή στο σύστημα.



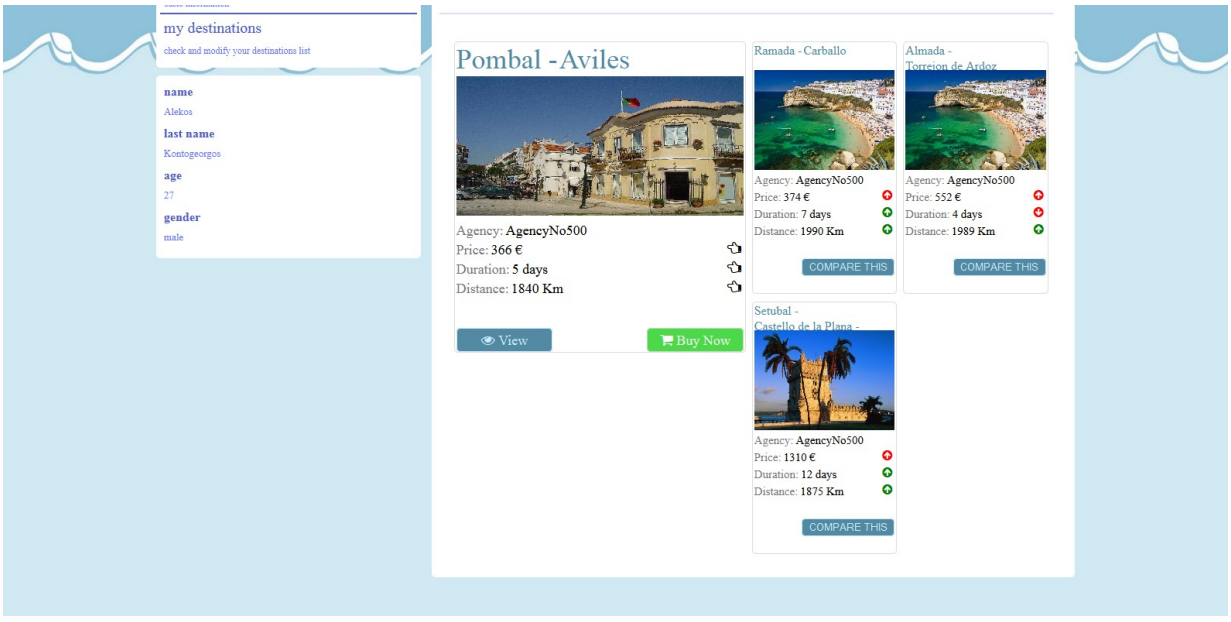


userHandler.php Δίνει την διεπαφή για την επεξεργασία των στοιχείων ενός χρήστη, όπως και την αλλαγή των στοιχείων εισόδου του στο σύστημα. Τμήμα της οθόνης φαίνεται παρακάτω.



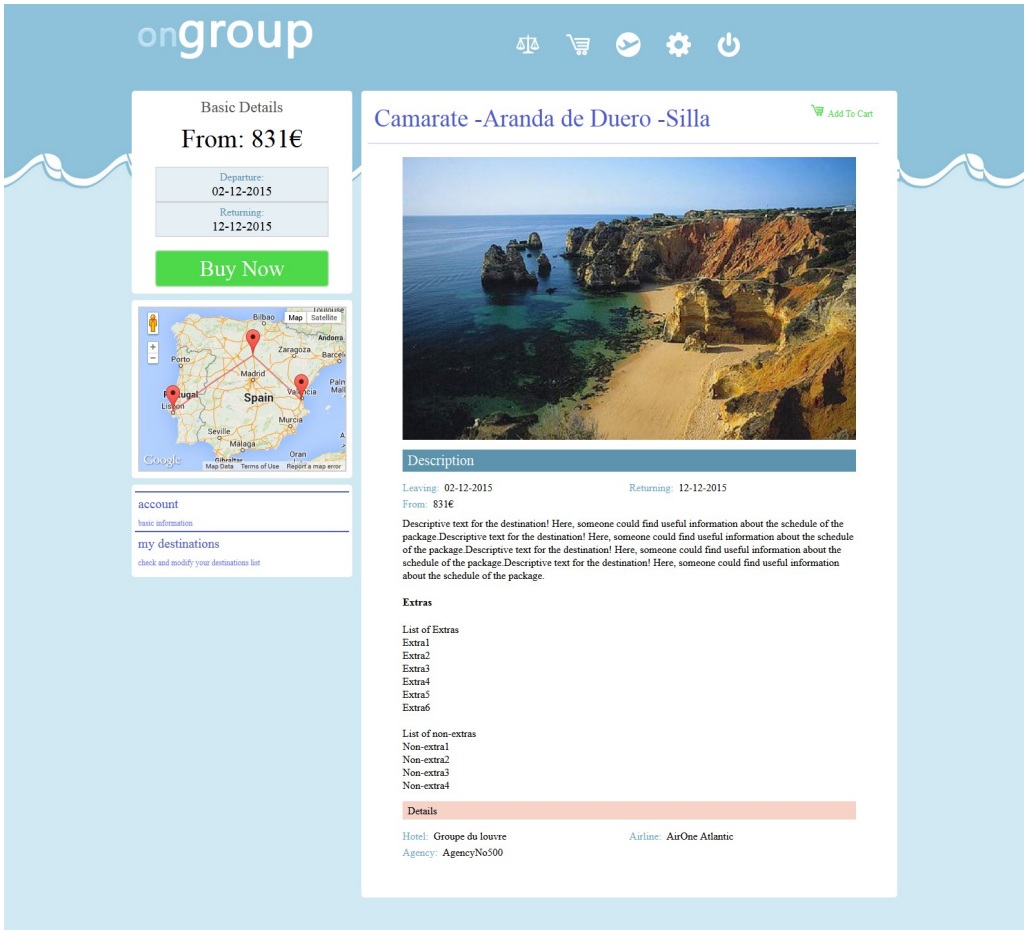
compare.php Παράγει το γραφικό περιβάλον για την σύγκριση των ταξιδιωτικών πακέτων.

compare.js Πραγματοποιεί την τελική σύγκριση των πακέτων, κάνοντας ανάκτηση απ'το local storage του browser και αλλάζοντας το γραφικό περιβάλλον και το πακέτο που αποτελεί την βάση σύγκριση ανάλογα με τις ενέργεις του χρήστη. Επίσης, για τις ανάγκες της σύγκρισης γίνεται χρήση του geolocation API, με το οποίο υπολογίζουμε την απόσταση που πρόκειται να ταξιδέψει ο χρήστης αν πραγματοποιήσει το συγκεκριμένο ταξίδι. Ως τέτοια απόσταση λαμβάνεται η απόσταση μεταξύ της θέσης του χρήστη που επισκέπτεται τον ιστοχώρο σε σχέση με το μέσο των αποστάσεων που προκύπτει απ'τις συντεταγμένες των προορισμών που συνθέτουν το πακέτο. Ακολουθεί screenshot της οθόνης:



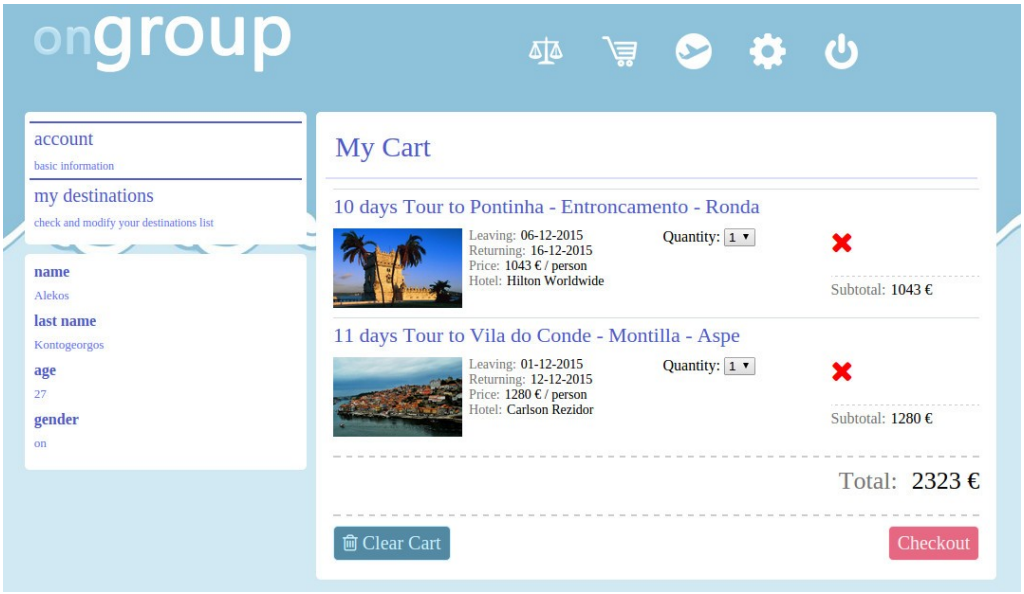
destination.php Κάνει αναλυτική προβολή των στοιχείων ενός πακέτου, ενώ με την βοήθεια του geoPckg.js προβάλει και την διαδρομή σε χάρτη με χρήση του google Maps API. Δίνει επίσης την δυνατότητα προσθήκης κάποιου πακέτου στο καλάθι αγορών.

geoPckg.js Στο συγκεκριμένο αρχείο κάνουμε λήψη των συντεταγμένων των προορισμών όπως μας παρέχονται από το Στρώμα Λογικής και παρουσιάζουμε τις διαδρομές του πακέτου με χρήση του google Maps API, με χρήση κόκκινης γραμμής. Παρακάτω φαίνεται ένα screenshot της οθόνης που συνθέτουν αυτά τα αρχεία.

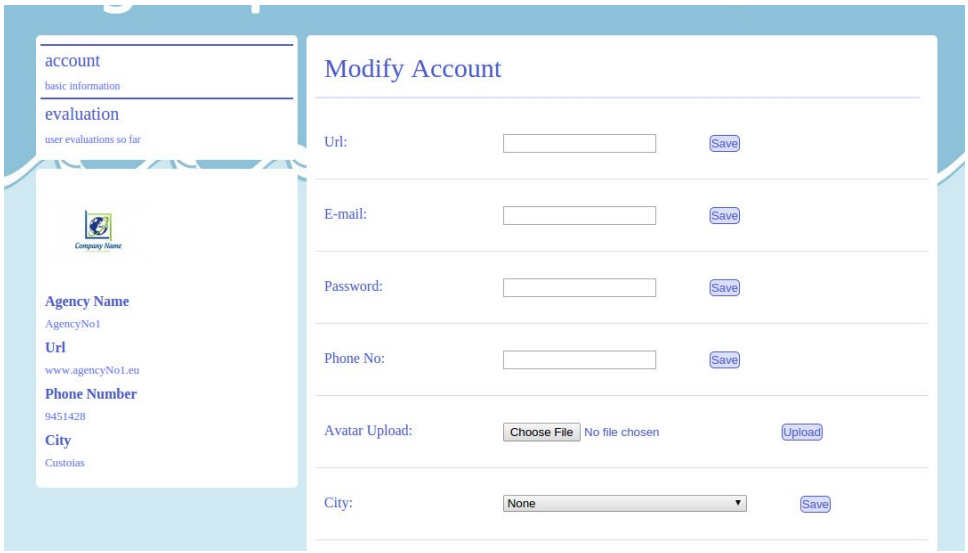


cart.php Το αρχείο αυτό δίνει την διεπαφή για το καλάθι αγορών. Χρησιμοποιεί το cartHandler.php για την λειτουργικότητά του σε επίπεδο εκκαθάρισης πακέτων από το καλάθι και την κλήση του συστήματος για αγορά καποιου πακέτου, καθώς και το cart.js για την διευκόλυνση των λειτουργιών του χρήστη στην συγκεκριμένη σελίδα.

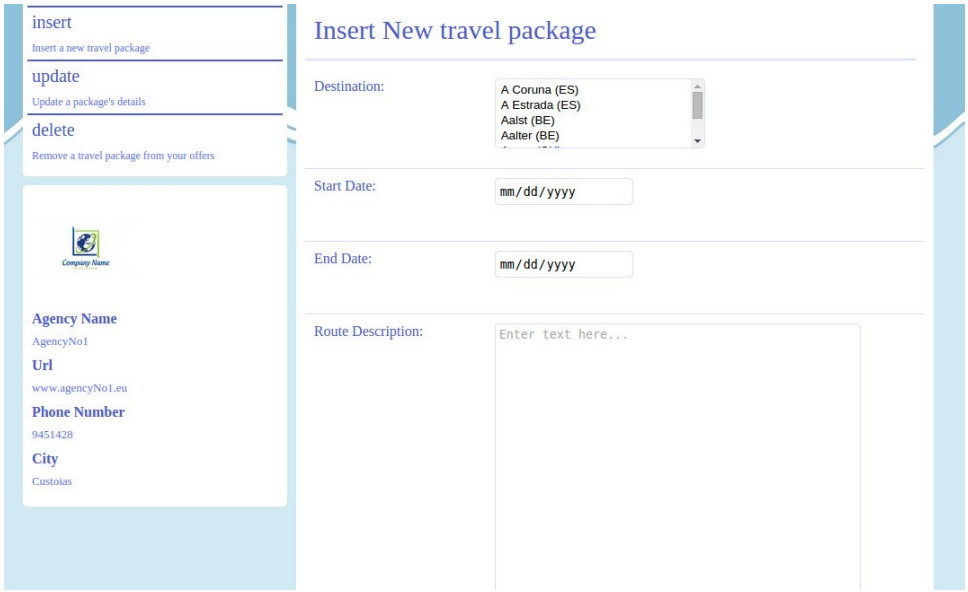
cart.js Αυτό το javascript αρχείο δίνει την δυνατότητα στον χρήστη να μπορεί να επιλέγει την ποσότητα των πακέτων στο καλάθι αγορών και να ενημερώνει κατάλληλα τις επιμέρους και την συνολική τιμή καθώς και την ενημέρωση του GUI σε περίπτωση αφαίρεσης πακέτου για να μην χρειάζεται συνεχής επικοινωνία με το Στρώμα Λογικής. Παρακάτω φαίνεται ένα screenshot της διεπαφής που δημιουργού αυτά τα αρχεία για δύο πακέτα, το ένα εκ των οποίων έχει επιλεγεί για δύο άτομα.



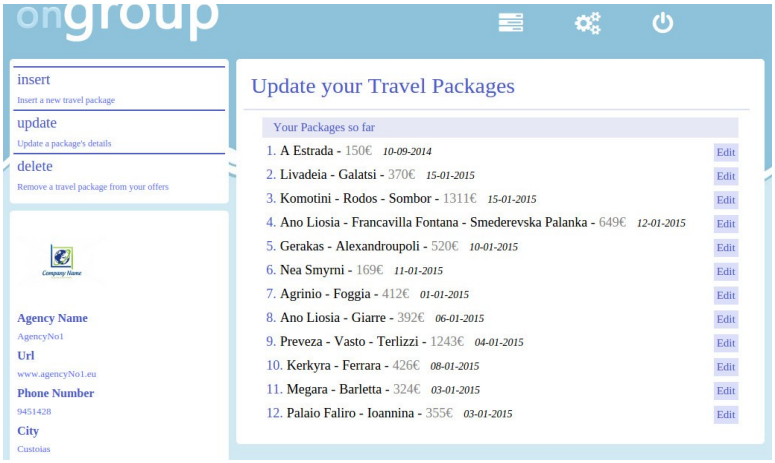
agencyAccount.php Η διεπαφή αυτή προσφέρει την δυνατότητα στο ταξιδιωτικό γραφείο να επεξεργαστεί τα στοιχεία του προφίλ του. Ακολουθεί screenshot.

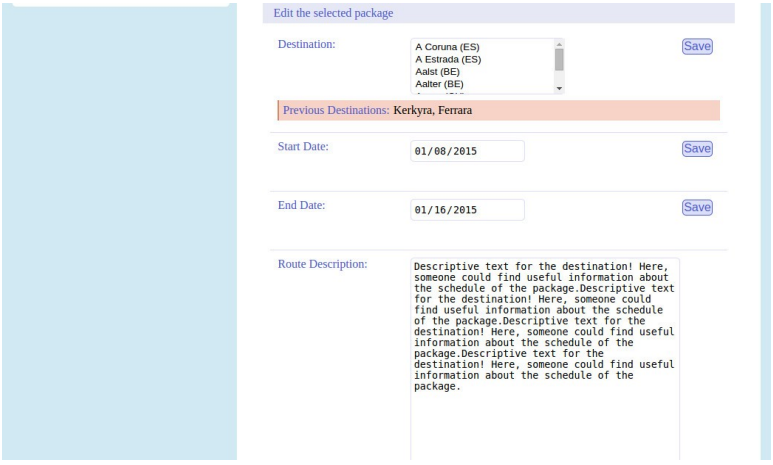


packageFunctions.php Είναι η διεπαφή μέσω της οποίας τα ταξιδιωτικά γραφεία έχουν πρόσβαση στις υπηρεσίες που τους προσφέρει το σύστημα ως προς τα ταξιδιωτικά πακέτα. Έτσι, μπορούν να εισάγουν, ενημερώσουν ή να διαγράψουν κάποιο πακέτο. Η διεπαφή χρειάζεται τις λειτουργίες των **insertPckg.php, updatePckg.php, deletePckg.php** για την ολοκλήρωση των λειτουργιών της, αντίστοιχα. Ακολουθεί αρχικά screenshot ενός τμήματος της διεπαφής εισαγωγής κάποιου πακέτου από ένα ταξιδιωτικό γραφείο.



Και στην συνέχεια, ακολουθούν screenshot απ'την διεπαφή για ενημέρωση των ταξιδιωτικών πακέτων. Το πρώτο, αφορά την λίστα που εμφανίζεται στο ταξιδιωτικό γραφείο για τα πακέτα που έχει ήδη εμφανίσει. Το δεύτερο, αποτυπώνει κομμάτι της φόρμας που εμφανίζεται μετά την επιλογή κάποιου πακέτου για επεξεργασία, όπου προβάλλονται τα στοιχεία του πακέτου όπως είχαν αποθηκευτεί την τελευταία φορά.

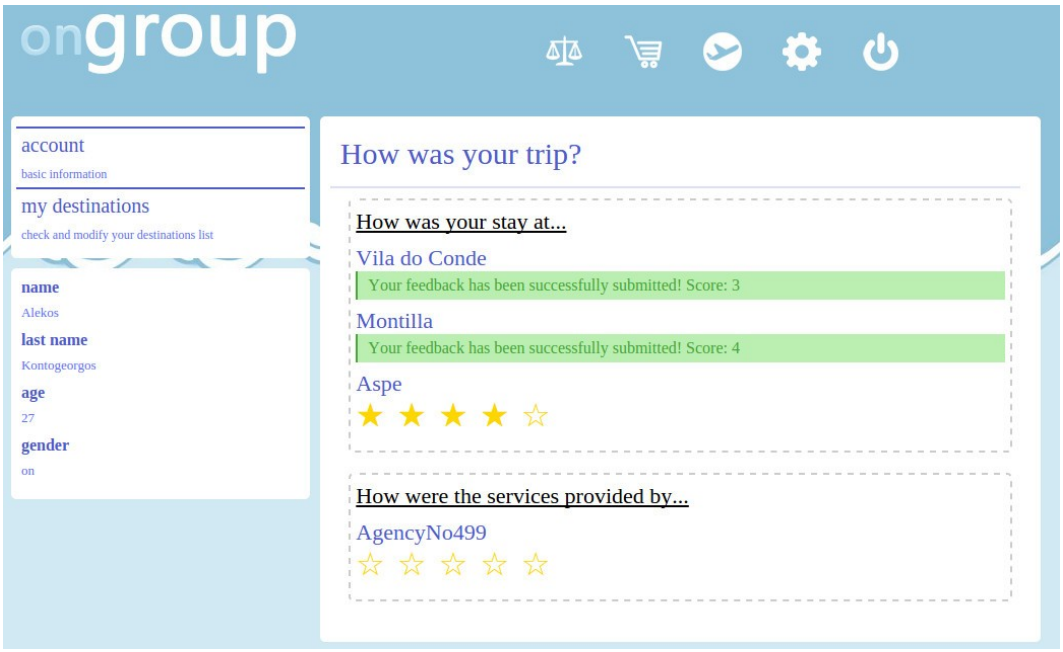




evaluate.php Είναι η διεπαφή για την αξιολόγηση των προορισμών και των ταξιδιωτικών γραφείων μετά την αγορά κάποιου πακέτου. Η συγκεκριμένη διεπαφή, για να δουλέψει πλήρως, χρησιμοποιεί το evaluate.js το οποίο περιγράφουμε στη συνέχεια.

evaluate.js Είναι το script με το οποίο δίνουμε λειτουργικότητα στο GUI της αξιολόγησης των προορισμών και του ταξιδιωτικού γραφείου που προσφέρει το πακέτο. Αυτό γίνεται μέσω του συστήματος βαθμολόγησης με έναν αριθμό αστεριών ο οποίος στο Στρώμα Λογικής μεταφράζεται σε ratings ανάλογα με το εύρος τιμών που έχει οριστεί για κάθε αξιολόγηση (π.χ. 1 αστέρι = 1 βαθμός αξιολόγησης για εύρος τιμών 1-5). Το αρχείο του Στρώματος Λογικής που αναλαμβάνει αυτήν την επεξεργασία αναλύθηκε παραπάνω και είναι το updatePrefs.php, ενώ το evaluate.js περιορίζεται σε θέματα GUI και διασύνδεσης με το χαμηλότερο στρώμα της αρχιτεκτονικής μας.

Ακολουθεί screenshot της διεπαφής αυτών των αρχείων.

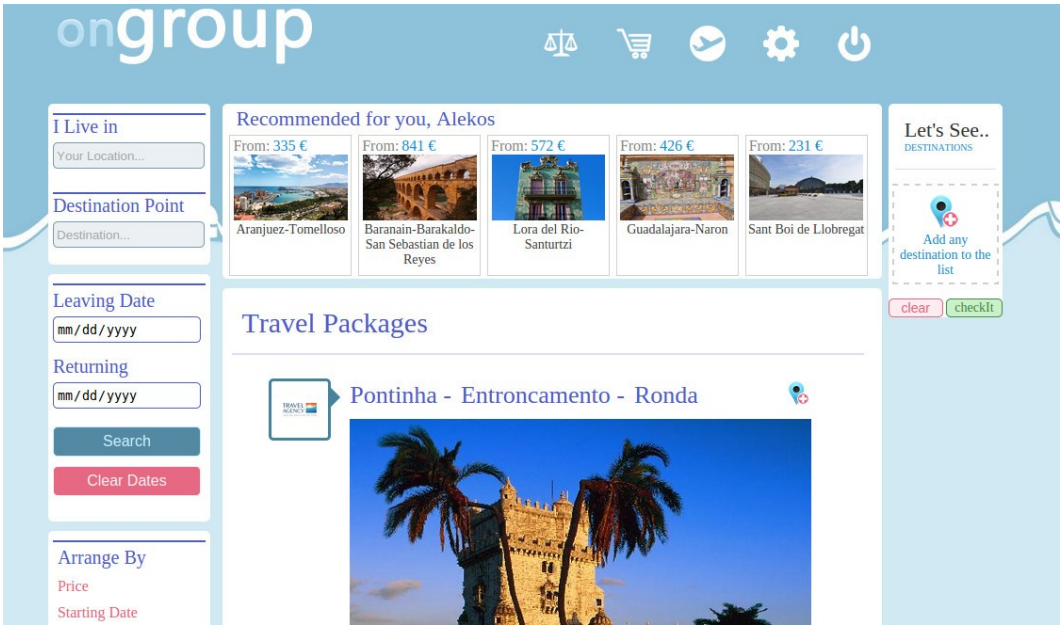


packages.php Είναι η διεπαφή μέσω της οποίας ο χρήστης κάνει όλες τις αναζητήσεις του για τα ταξιδιωτικά πακέτα, ενώ προσφέρει χώρο και για τα πακέτα που συστήνονται από το σύστημα συστάσεων. Παρέχει επομένως τα εργαλεία αναζήτησης και κατάταξης των πακέτων ανά κριτήρια που τίθενται απ'τον χρήστη καθώς και το εργαλείο προσωρινής αποθήκευσης των πακέτων που επιθυμεί να συγκρίνει ο χρήστης, το οποίο λειτουργεί με την βοήθεια του listHandler.js.

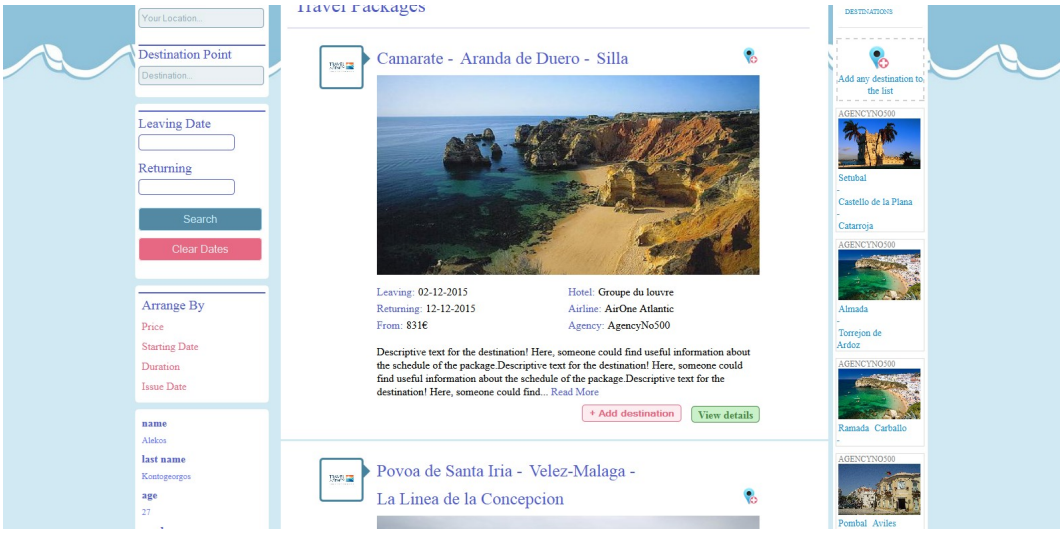
listHandler.js Είναι το script με το οποίο γίνεται η διαχείριση των πακέτων που επιθυμεί να αποθηκεύσει προσωρινά ο χρήστης κατά την πλοήγησή του στο σύστημά μας, με την λογική ότι αποτελούν τα πιο ενδιαφέροντα και πρόκειται να τα συγκρίνει στην συνέχεια όπως περιγράψαμε και στο σχετικό κεφάλαιο. Η λειτουργικότητα αυτή εξασφαλίζεται με την χρήση κατάλληλων κουμπιών σε κάθε πακέτο που προβάλλεται και με την αξιοποίηση του local storage του εκάστοτε browser. Φυσικά δίνεται και η δυνατότητα αδειάσματος της λίστας, όποτε επιθυμεί ο χρήστης.

Στην συνέχεια ακολουθούν δύο screenshots. Το πρώτο δείχνει την διεπαφή που δημιουργούν

τα δύο παραπάνω αρχεία, με τα πακέτα που συστήνονται να βρισκονται στην κορυφή του Main Content Area, τα εργαλεία αναζήτησης να βρισκονται στο Side Content Area, και την λίστα να είναι προσωρινά άδεια.

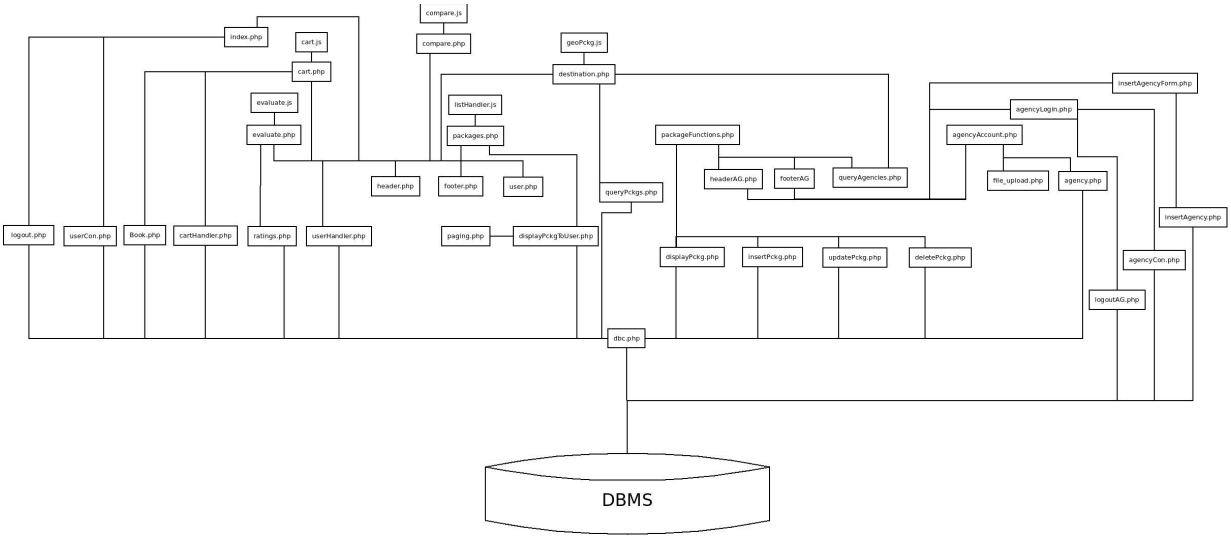


Το δεύτερο screenshot, δείχνει την ίδια διεπαφή, με την λίστα των προς σύγκριση πακέτων γεμάτη με μερικά πακέτα, λίγο πριν την σύγκριση με την χρήση του κουμπιού “checkIt”. Επίσης, φαίνεται ακόμα πιο καθαρά το Side Content όπως είναι διαμορφωμένο σε αυτήν την διεπαφή.



Η ανάλυση της υλοποίησης όλων των στρωμάτων σύμφωνα με την αρχιτεκτονική που

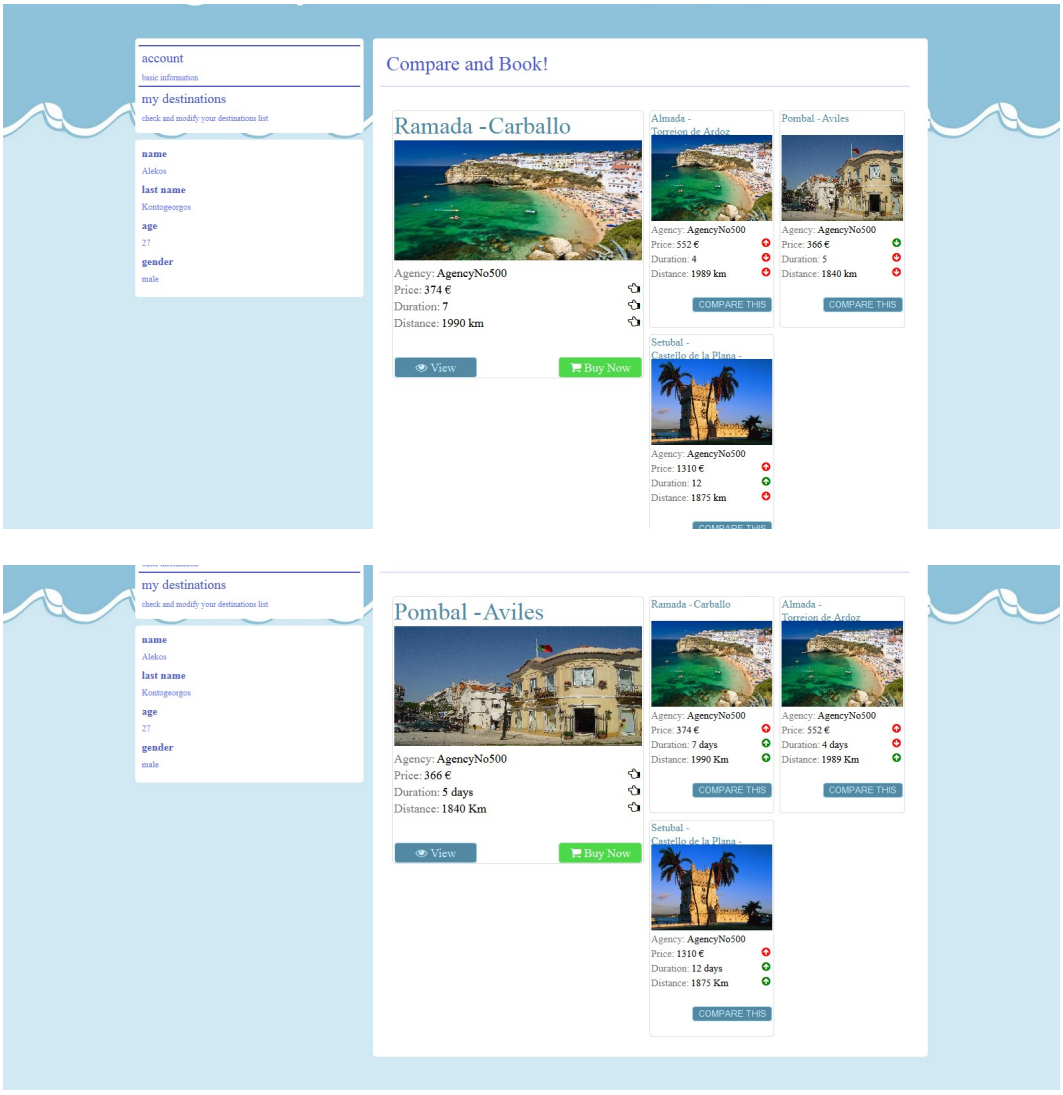
επιλέξαμε, τελειώνει με τη παρουσίαση της διασύνδεσης των στρωμάτων σε επίπεδο αρχείων. Η παρουσίαση αυτή γίνεται στο σχήμα 4.12 που ακολουθεί. Στην συνέχεια, παρουσιάζουμε την υλοποίηση του συστήματος συστάσεων, σύμφωνα με την προσέγγισή μας στο συγκεκριμένο πρόβλημα, όπως την αναλύσαμε στι Κεφάλαιο 1.



Σχήμα 4.12: Η διασύνδεση των αρχείων που συνθέτουν το σύστημά μας, κάνοντας διακριτά τα Στρώματα της Αρχιτεκτονικής μας.

4.2.4 Εργαλείο Σύγκρισης Πακέτων

Η ανάλυση του εργαλείου σύγκρισης που παρέχεται, γίνεται αναλυτικά στο Κεφάλαιο 3.3. Εδώ συμπληρώνουμε την ανάλυσή μας με κάποια τεχνικά χαρακτηριστικά και οθόνες απ'την χρήση του εργαλείου. Το εργαλείο αυτό υλοποιήθηκε εξ'ολοκλήρου με javascript χωρίς την χρήση κάποιου framework. Με την javascript, εκμεταλλευόμαστε την δυνατότητα των browsers για local storage, έτσι ώστε να αποφεύγεται η συνεχής επικοινωνία με το Στρώμα Λογικής και κατ'επέκταση με την βάση δεδομένων. Έτσι, όταν γίνεται η χρήση του εργαλείου, ανακτάται απ'το local storage η λίστα των πακέτων που πρόκειται να συγκριθούν. Εκεί είναι διαθέσιμη όλη η απαραίτητη πληροφορία εκτός του στοιχείου που αφορά την απόσταση που πρόκεται να διανύσει ένας χρήστης επιλέγοντας ένα πακέτο. Αυτό αντιμετωπίζεται με χρήση του Geolocation API , το οποίο λαμβάνει υπόψη την γεωγραφική θέση του χρήστη με την γεωγραφική θέση ενός προορισμού ή την γεωγραφική θέση ενός μέσου σημείου δύο ή περισσότερων προορισμών. Η λογική της σύγκρισης γίνεται βάσει πάντα ενός πακέτου το οποίο μπορεί να αλλάξει σύμφωνα με τις επιλογές του χρήστη. Δίνεται η δυνατότητα προβολής στοιχείων ενός πακέτου και η αγορά του, ενώ ένα σενάριο χρήσης του εργαλείου φαίνεται παρακάτω:



Απ'τα παραπάνω screenshot φαίνεται πιο καθαρά η αξία του εργαλείου σύγκρισης όπως αναλύθηκε στο σχετικό κεφάλαιο. Παρατηρούμε οτι στο πρώτο screenshot που αφορά το πακέτο Ramada – Carballo, τα υπόλοιπα πακέτα στην πλειοψηφία των στοιχείων του έχουν κατώτερα χαρακτηριστικά όπως τα κρίνει το σύστημά μας. Αντιθέτως, το δεύτερο screenshot που αφορά την σύγκριση του πακέτου Pombal – Aviles δείχνει ότι δύο πακέτα απ'τα υπόλοιπα με τα οποία συγκρίνεται, αφορούν -πιθανότατα- καλύτερες επιλογές. Η σύγκριση αυτή προέκυψε γρήγορα και εύκολα για τον χρήστη χωρίς λεπτομερή έλεγχο στα χαρακτηριστικά του κάθε πακέτου απ'την δική του πλευρά. Φυσικά η τελική επιλογή για το ποιά είναι η πιο συμφέρουσα επιλογή για τον χρήστη, αφήνεται στον ίδιο.

4.2.5 Σύστημα Συστάσεων (Recommender)

Τέλος, αναλύουμε την υλοποίηση του συστήματος συστάσεών του διδικτυακού μας συστήματος. Υπενθυμίζουμε ότι ο recommender μας λειτουργεί ως εξής:

- Καταγράφει τις κινήσεις των χρηστών ως προς τις αγορές τους (booking requerst).

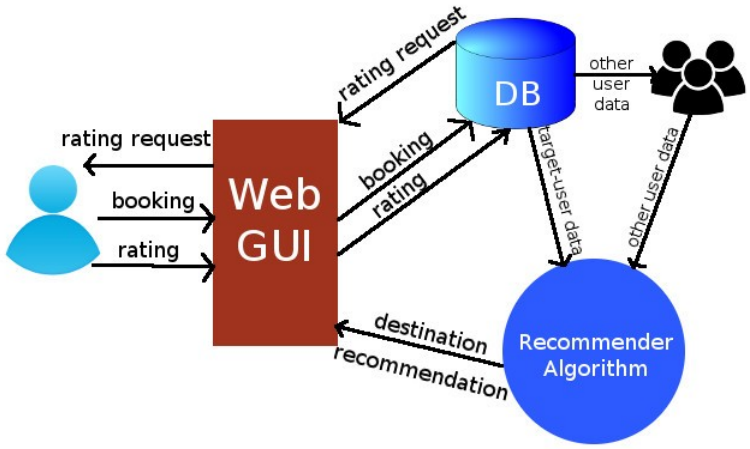
- Το σύστημα ενημερώνεται για το booking request απ'την πλευρά του χρήστη και στέλνει αίτημα αξιολόγησης του πακέτου ως προς τους προορισμούς και το ταξιδιωτικό γραφείο που συνθέτουν το πακέτο (rating request).

- Καταγράφει τις αξιολογήσεις αυτές του χρήστη (rating).

- Αναλύει τα δεδομένα του χρήστη-στόχου και των υπολοίπων χρηστών που έχουν ήδη καταχωρηθεί.

- Κάνει σύσταση προορισμών που δεν έχει επισκεφτεί ο χρήστης.

Στο σχήμα 4.13 φαίνεται η αναπαράσταση της λειτουργίας του ως προς το συνολικό σύστημα και στην συνέχεια περιγράφουμε τα επιμέρους χαρακτηριστικά του.



Σχήμα 4.13: Η γενική ιδέα του συστήματος συστάσεων, σε συνδυασμό με τις υπόλοιπες οντότητες του ιστότοπού μας

Το πρόβλημα που είχαμε να αντιμετωπίσουμε όπως το αναλύσαμε και στο σχετικό κεφάλαιο, ήταν το ότι κάθε πακέτο αφορούσε μια νέα οντότητα. Έτσι, ένα πακέτο του οποίου ο χρόνος ζωής περιορίζεται μέχρι την ημερομηνία έναρξής του, δεν μας δίνει την δυνατότητα ανάλυσης του ιστορικού του γιατί αυτό θα είναι πολύ μικρό, ενώ το ίδιο το πακέτο θα εμφανιστεί ξανά, αργότερα ως μια νέα οντότητα, αναγκάζοντας μας να το επεξεργαστούμε απ'την αρχή. Γι'αυτό το λόγο, αξιολογούνται προορισμοί και όχι πακέτα.

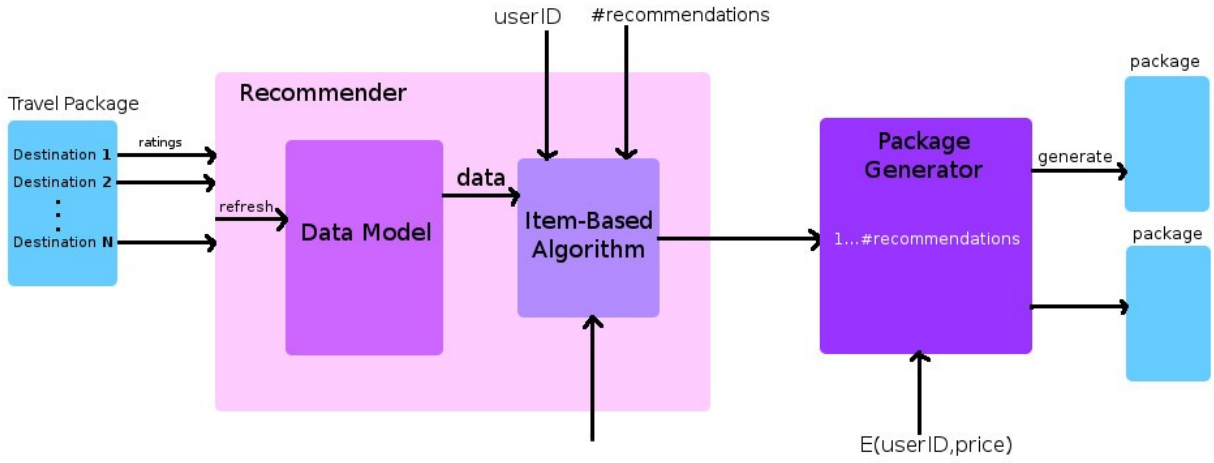
Υπήρξε και μια εναλλακτική λύση με εφαρμογή data mining αλγορίθμου. Σύμφωνα με αυτή την λύση, θα μπορούσαμε να κάνουμε clustering στα ταξιδιωτικά πακέτα κατατάσσόντάς σε συστάδες οι οποίες θα αποτελούσαν την βάση για την συνέχεια την ανάλυσή μας.

Χωρίς την χρήση κάποιας data mining τεχνικής, το σύστημά μας αφού δεχθεί ως είσοδο κάποια αξιολόγηση, κάνει refresh στο data model του για έγκυρες συστάσεις. Στην συνέχεια, του παρέχεται το id του χρήστη στόχου, αναλύει τα διαθέσιμα δεδομένα και επιστρέφει κάποιους προτεινόμενους προορισμούς για τον χρήστη. Αυτές οι συστάσεις αφορούν προορισμούς που δεν έχει επισκεφτεί. Φυσικά, αυτές οι συστάσεις δεν έχουν νόημα

ως έχουν. Έτσι, το σύστημά μας, δέχεται την έξοδο του recommender και εφαρμόζει έναν απλό αλγόριθμο, σύνθεσης αυτών των αποτελεσμάτων. Πιο συγκεκριμένα, αναζητά πακέτα τα οποία περιέχουν κάποιον από αυτούς τους προτεινόμενους προορισμούς – και όσο περισσότεροι προορισμοί σε ένα τέτοιο πακέτο τόσο το καλύτερο. Στην συνέχεια, επιχειρήθηκε και μια μικρή παρεμβολή content-based σύστασης ως προς τις συστάσεις. Η λογική μας είχε να κάνει σε σχέση με τον μέσο όρο τιμών πακέτων που προτιμά ο ταξιδιώτης. Έτσι, είναι λογικό να υποθέσουμε ότι ένας ταξιδιώτης προτιμά για τους ίδιους προορισμούς είτε φθηνά είτε ακριβά πακέτα, σύμφωνα με τις αγοραστικές του προτιμήσεις και τα χαρακτηριστικά των πακέτων που συνθέτουν μη πολυτελείς και πολυτελείς προτάσεις, αντίστοιχα. Αυτή η προσέγγιση όπως θα δούμε παρακάτω, είχε προβλήματα ως προς την αξιοπιστία των συστάσεων διότι τα δεδομένα μας ήταν συνθετικά με αποτέλεσμα να μην εξάγεται κάποια ουστιαστική πληροφορία ως προς το περιεχόμενο των πακέτων, και κατ'επέκταση ως προς την τιμή τους.

Ως προς την αλγοριθμική προσέγγιση των συστάσεων, επιλέξαμε item-based recommendation για τους λόγους που αναλύθηκαν εκτενώς στο Κεφάλαιο 2.9. Έτσι, πληρούνται οι προϋποθέσεις για βέλτιστες συστάσεις όπως μας εξασφαλίζει κατά γενικό κανόνα το item-based recommendation, το συντομότερο δυνατό. Ιδιαίτερα το ζήτημα της ταχύτητας, μας απασχολεί ως προς το ότι έχουμε να κάνουμε με ένα διαδικτυακό σύστημα όπου οι χρόνοι αναμονής είναι καθοριστικοί για την εξασφάλιση καλού user experience, ζητούμενο σε κάθε ιστότοπο.

Στο σχήμα 4.14 φαίνεται πιο καθαρά η λειτουργία του συστήματος συστάσεων μας, από την καταχώρηση των αξιολογήσεων μέχρι την εξαγωγή νέων συστάσεων.



Σχήμα 4.14: Η λειτουργία του reocmmender μας. Δέχεται τις αξιολογήσεις. Κάνει ανανέωση του data model του. Ο Item-based αλγόριθμος δέχεται από το σύστημα ως είσοδο το id του χρήστη-στόχου και των αριθμών των συστάσεων ως έξοδο. Η έξοδος των συστάσεων συντίθεται σε πακέτα στα οποία επιχειρήθηκε η content-based παρεμβολή, και αυτό παράγει τα πακέτα προς σύσταση.

Σε επίπεδο δεδομένων, δεν είχαμε διαθέσιμα δεδομένα οπότε έγινε χρήση συνθετικών δεδομένων, που κατασκευάσταν με δική μας αλγοριθμική διαδικασία. Πήραμε το MovieLens dataset διαθέσιμο από το grouplens.org. Απ'τα διαθέσιμα dataset πήραμε το dataset που περιέχει 100k ratings. Το συγκεκριμένο dataset έχει ids χρηστών που έχουν βαθμολογήσει ταινίες. Έτσι, τα 100k είναι στη μορφή εγγραφών ως εξής: Κάθε εγγραφή έχει ένα id χρήστη ένα id ταινίας και μια τιμή που αφορά το rating που έχει δώσει σε κάθε ταινία. Οι τιμές των

ratings, κινούνται στο εύρος τιμών 1-5, το οποίο βρίσκεται και σε συμφωνία με το σύστημα αξιολόγησης που έχουμε υλοποιήσει στο σύστημά μας. Σε επίπεδο πόλεων, χρησιμοποιήσαμε το dataset που παρέχεται από το geonames.org, επιλέγοντας πόλεις με πληθυσμό άνω των 15.000. Από αυτά τα δύο dataset, κάναμε την αντιστοίχιση των users και των ταινιών του MovieLens με τους users και τις πόλεις/προορισμούς του συστήματός μας, αντίστοιχα. Οι προορισμοί μας -επειδή ήταν περιορισμένοι σε αριθμό- αντιστοιχήθηκαν σε πόλεις 11 χωρών στην Ευρώπη, έτσι ώστε η -στην συνέχεια- σύνθεση των πακέτων να έχει περισσότερο νόημα, απ'τον να επιλέγαμε τυχαίες πόλεις.

Στο κομμάτι της δημιουργίας των πακέτων, ο αλγόριθμος σύνθεσης των δεδομένων, κινείται ως εξής:

Σε κάθε πακέτο,

- ➔ Αντιστοιχήσαμε μέχρι 3 προορισμούς ώστε τα πακέτα να προσεγγίζουν την πραγματικότητα, όπως αυτή παρατηρήθηκε από πραγματικά τουριστικά εμπορικά συστήματα.

- ➔ Το κόστος των πακέτων, τέθηκε ανάλογα με την διάρκεια της εκδρομής. Έτσι δημιουργήθηκαν εύρη τιμών ανάλογα με την διάρκειά τους. Για παράδειγμα, πακέτο με διάρκεια 1-3 μέρες μπορούσε να έχει τιμή από 100-250 ευρώ, για διάρκεια 3-5 μέρες να έχει τιμή 250-450 ευρώ κτλ.

- ➔ Χρονικά τα πακέτα είναι ομοιόμορφα κατανεμνημένα μέσα στη διάρκεια ενος έτος. Έτσι, έχουμε για παράδειγμα 100 πακέτα ανα μήνα, σε διάφορες χρονικές περιόδους μέσα στον μήνα αυτό.

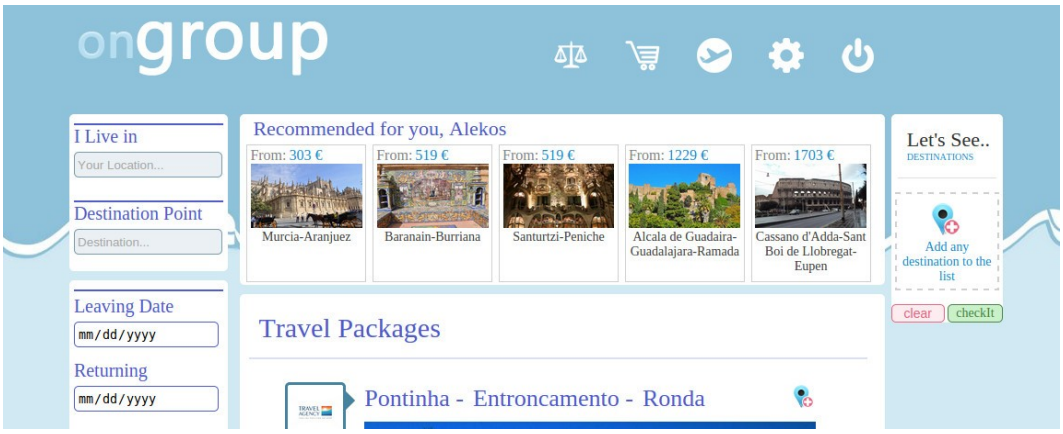
- ➔ Και για να έχει νόημα η σύσταση, τηρήθηκαν περιορισμοί όπως το να αντιστοιχούνται σε κάθε προορισμό τουλάχιστον 5 πακέτα και κάθε ταξιδιωτικό γραφείο να έχει τουλάχιστον 10 πακέτα.

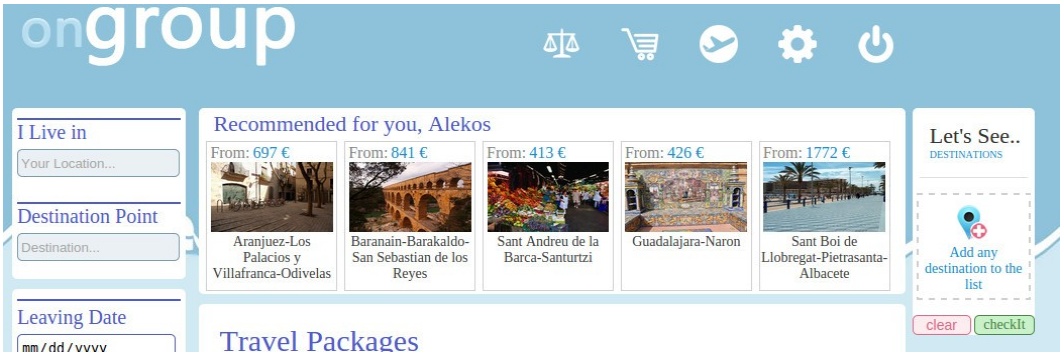
Τα υπόλοιπα δεδομένα, όπως για παράδειγμα οι ηλικίες των χρηστών, τα email επικοινωνίας, έγιναν με σύνθεση χωρίς κάποια συγκεκριμένη λογική μιας και δεν μας απασχολούν στην ανάλυσή μας. Κάθε βήμα σύνθεσης έγινε με την βοήθεια συναρτήσεων που παράγουν τυχαίους αριθμούς οι οποίοι είτε χρησιμοποιούνταν αυτούσιοι (πχ τιμές ταξιδιωτικών πακέτων) είτε χρησιμοποιούνταν για τον προσδιορισμό μιας πόλης/προορισμού (πχ ο αριθμος αντιστοιχεί στην θέση της πόλης που πρόκειται να χρησιμοποιηθεί βάσει της θέσης της στην βάση δεδομένων).

Το σύστημά μας λειτούργησε χωρίς κάποιο ιδιαίτερο πρόβλημα, και μάλιστα το σύστημα συστάσεων λειτούργησε στο μεγαλύτερό του ποσοστό. Το ποσοστό αυτό προσδιορίζεται από το γεγονός ότι ναι μεν στον χρήστη προτείνονται πακέτα που -όπως αναμενόταν- είναι πιο πιθανό να του αρέσουν βάσει και του αλγορίθμου σύνθεσης που υλοποιήσαμε, απ την άλλη όμως οποιοδήποτε άλλο εγχείρημα για content-based επέμβαση απέτυχε. Η αποτυχία αυτή είναι εύκολο να εξηγηθεί αν σκεφτεί κάποιος ότι αν για παράδειγμα πάρουμε τον μέσο όρο των τιμών των ταξιδιωτικών πακέτων που έχει επιλέξει ένας χρήστης, αυτός ο μέσος όρος δεν θα έχει καμία απολύτως αξία στην βάση της τυχαιότητάς του. Ένα ακόμα θετικό του συστήματός μας είναι ότι τα αποτελέσματα παράγονται γρήγορα, χωρίς να επιβάλει κάποια καθυστέρηση στην λειτουργία της διαδικτυακού μας συστήματος γενικά και μάλιστα στην σελίδα που διαχειρίζεται το περισσότερο υλικό.

Στην συνέχεια, ακολουθούν δύο screenshot που δείχνουν δύο διαδοχικές επισκέψεις

στην σελίδα που παράγονται οι συστάσεις:





Αυτό που αξίζει να παρατηρηθεί είναι ότι σε δύο διαδοχικές επισκέψεις, παράγονται διαφορετικά πακέτα (άρα και ποικιλία για συστάσεις στον χρήστη) τα οποία όμως έχουν ως κοινούς κάποιους προορισμούς. Αυτοί οι προορισμοί μπορεί να ανήκουν σε διαφορετικά πακέτα, δείχνουν όμως ότι το σύστημα συστάσεών μας λειτουργεί καλά διοτι τους περιλαμβάνει πάντα σε κάποιο πακέτο. Μάλιστα, αν υπάρχει πακέτο το οποίο να περιλαμβάνει περισσότερους από έναν προορισμούς που είναι υπο σύσταση σε κάποιο χρήστη, τότε αυτό το πακέτο εμφανίζεται πολυ συχνά στο τμήμα των συστηνόμενων πακέτων, κάτι που επίσης είναι επιθυμητό. Τέλος, αν κοιτάξει κάποιος τα δεδομένα των χρηστών και τις συστάσεις των πακέτων που παράγονται, θα παρατηρήσει ένα ακόμα ενδιαφέρον στοιχείο. Ο recommender, με τον τρόπο που τον υλοποιήσαμε, προτείνει και πακέτα τα οποία περιλαμβάνουν προορισμούς που έχει ήδη επισκεφθεί. Αυτό όμως δεν είναι αρνητικό. Διότι στο πακέτο περιλαμβανονται προορισμοί στους οποίους δεν έχει ταξιδέψει και μπορεί να παραμένει μια ελκυστική επιλογή για τον ίδιο. Έτσι, ακόμα και σε ένα πραγματικό σενάριο, ένας ταξιδιώτης μπορεί να έχει ταξιδέψει -για παράδειγμα- στην Θεσσαλονίκη, αλλά να θέλει να επισκεφθεί ξανά αυτή την πόλη σε συνδυασμό με την επίσκεψή του στον Βόλο. Επομένως, το πακέτο Βόλος- Θεσσαλονίκη, εφ'όσον ο Βόλος κρίνεται ως μια καλή σύσταση από τον recommender μας, αποτελεί μια ελκυστική πρόταση για τον ταξιδιώτη.

Κεφάλαιο 5

Συμπεράσματα

Στο κεφάλαιο αυτό, αναφερόμαστε στα συμπεράσματα στα οποία καταλήξαμε κατά την μελέτη, τον σχεδιασμό και την υλοποίηση του τελικού μας συστήματος. Αυτά τα συμπεράσματα αφορούν τις τεχνολογίες που χρησιμοποιήθηκαν για την συνολική ανάπτυξη του συστήματός μας καθώς και την λειτουργία του συστήματος συστάσεών μας.

3.1 Εργαλεία Ελεύθερου Λογισμικού Ανοιχτού Κώδικα

Για την ανάπτυξη του συστήματος χρησιμοποιήθηκαν εξ'ολοκλήρου εργαλεία ελεύθερου λογισμικού ανοιχτού κώδικα. Αυτό μας παρέχει ευκολίες είναι οποίες είναι τόσο καταγεγγραμένες στις βιβλιογραφίες που αφορούν το ελεύθερο λογισμικό όσο και στην ανάλυσή μας για τις τεχνολογίες που χρησιμοποιήθηκαν στο Κεφάλαιο 2. Αυτό που αξίζει να αναφέρουμε ως επιπλέον χαρακτηριστικό, είναι ότι λόγω του δημοφιλούς τους χαρακτήρα, υπάρχει πολύ μεγάλη υποστήριξη από τον κόσμο που εμπλέκεται με το ελεύθερο λογισμικό με αποτέλεσμα να έχουμε ένα μεγάλο εύρος λύσεων σε κάθε πρόβλημα που παρουσιαζόταν χωρίς να καθυστερείται η ανάπτυξη του συστήματος, η να αναγκαζόμαστε να τροποποιούμε τον αρχικό μας σχεδιασμό για την αντιμετώπιση αυτών των προβλημάτων.

3.2 Σύστημα συστάσεων

Όπως αναλύθηκε και στο κομμάτι της υλοποίησης, δεν υπήρχε η δυνατότητα χρήσης πραγματικών δεδομένων. Επομένως καταφύγαμε στην σύνθεση αυτών από πραγματικά dataset, που όμως θέτουν περιορισμούς ως προς την αλγοριθμική προσέγγιση του συστήματος συστάσεων, όπως διαπιστώσαμε και προηγουμένως.

Αρχικά, τα συνθετικά δεδομένα εμποδίζουν την ίδια την αξιολόγηση ενός αλγορίθμου. Έτσι, αν και το Mahout παρέχει εργαλεία για την αξιολόγηση των συστάσεων ενός αλγορίθμου, αυτά δεν έχουν αξία αν τα δεδομένα δεν είναι πραγματικά. Απλά αναφέρουμε ότι ένα σημαντικό τέτοιο εργαλείο είναι η χρήση κλασικών τεχνικών μετρικών ανάκτησης πληροφοριών που εφαρμόζονται σε μια κλάση αλογίθμων που έχουν να κάνουν π.χ. με μηχανές αναζήτησης, όπου το ζητούμενο είναι η επιστροφή αποτελεσμάτων βάσει ενός αιτήματος – για την μηχανή αναζήτησης ενός query – ανάμεσα σε ένα σύνολο πιθανών αποτελεσμάτων. Αυτό του είδους οι μετρικές εφαρμόζουν πολύ καλά και σε αλγόριθμους που σχετίζονται με συστάσεις. Και αυτό διότι όπως μια μηχανή αναζήτησης δεν θα έπρεπε να

επιστρέφει αποτελέσματα άσχετα προς το ερώτημα που της τέθηκε, έτσι και στο σύστημα συστάσεων στόχος είναι η επιστροφή όσο το δυνατόν πιο σχετικών αποτελεσμάτων γίνεται. Αυτές οι μετρικές είναι για παράδειγμα το precision και το recall. Το precision είναι το ποσοστό των κορυφαίων αποτελεσμάτων που είναι σχετικά με το ζητούμενό μας, για κάποιο ορισμό σχετικότητας που του έχουμε αποδώσει. Το recall είναι το ποσοστό όλων των σχετικών αποτελεσμάτων που περιλαμβάνονται στα κορυφαία αποτελέσματα. Παρ'όλα αυτά δεν υπάρχει νόημα να εξετάσουμε αυτές τις μετρικές όταν τα δεδομένα για τα οποία ρυθμίζουμε τον αλγόριθμό μας σχετίζονται με ταινίες και τα αποδίδουμε ως ταξιδιωτικές επιλογές.

Ο περιορισμός αυτός επεκτείνεται και ως προς το customization ενός αλγορίθμου. Όπως είδαμε και στο Κεφάλαιο 2, οι item-based αλγόριθμοι προσφέρουν ευελιξία ως προς το πεδίο του προβλήματος (domain independent). Σε αυτή την ευελιξία μπορούμε να στηριχτούμε ώστε να εισάγουμε στο σύστημά μας μια δόση content-based χαρακτήρα. Και το επιχειρήσαμε χωρίς όμως αποτελέσματα. Και αυτό διότι, όταν ο χρήστης βαθμολογεί θετικά δύο ταινίες με ids το x και το y, εννοείται ότι στο ζεύγος αυτών των αναγνωριστικών που εκπροσωπούν τις ταινίες, υπάρχει μια σχέση την οποία καλούμαστε να την ανακαλύψουμε αλγοριθμικά. Η σχέση αυτή μπορεί να είναι ότι αφορούν και οι δύο ταινίες δράσης ή οτι έχουν τον ίδιο σκηνοθέτη. Όταν όμως χρησιμοποιούμε τα συγκεκριμένα δεδομένα για την αναπαράσταση των δικών μας, τότε δημιουργείται ένα μεγάλο εννοιολογικό χάσμα για το ζεύγος αντικειμένων x,y το οποίο δεν μπορεί να ερμηνευτεί εκ νέου από τον αλγόριθμό μας μιας και αντιλαμβάνεται μόνο ratings. Έτσι, τα x και y μπορεί να αντιπροσωπεύουν πακέτα που είναι εντελώς άσχετα μεταξύ τους σε επίπεδο περιεχομένου, και να μην υπήρχε ταξιδιώτης που να επέλεγε την αγορά και των δύο πακέτων -πόσο μάλλον την παρόμοια βαθμολόγησή τους- και παρ'όλα αυτά ο recommender μας να τα παρουσιάζει ως σχετικά. Επομένως, αν και ο χώρος των ταξιδιωτικών πακέτων προσφέρεται για content-based recommendation, αυτό δεν είναι εφικτό στην περίπτωσή μας.

Το μόνο που θα μπορούσαμε να αναφέρουμε είναι ότι σε ένα πραγματικό σενάριο, όπου τα δεδομένα μας θα ήταν πραγματικά, υπάρχουν πολύ καλές προοπτικές για επιτυχημένο content-based recommendation βάσει τιμών των πακέτων καθώς επίσης και ως προς τον χαρακτήρα των προορισμών – πολυτελείς διαμονές ή διάρκεια ταξιδιού. Και αυτό είναι λογικό μιας και ένας ταξιδιώτης αλλάζει αργά προτιμήσεις. Έτσι, ένας χρήστης με χαμηλή αγοραστική δυνατότητα, θα προτιμά μικρές και φθηνές εκδρομές την στιγμή που κάποιος χρήστης που μεγάλη αγοραστική δυνατότητα προτιμά πολυτελείς επιλογές και ένας εργαζόμενος προτιμά πακέτα που συμβαδίζουν με το πρόγραμμα των αδείων του (π.χ. 10 ήμερα ταξίδια).

Φυσικά, η ανάπτυξη με item-based recommendation μας έδωσε και κάποια πολύ χρήσιμα συμπεράσματα. Ένα από αυτά είναι η ταχύτητα εκτέλεσής τους. Σε ένα τυπικό online σύστημα, ο ρυθμός αύξησης των χρηστών σε σχέση με τον ρυθμό αύξησης των αντικειμένων είναι μεγαλύτερος κατά γενικό κανόνα. Και αφού οι χρονικές και χωρικές αποδόσεις του item-based recommender σχετίζονται με το μέγεθος του συνόλου των αντικειμένων προς ανάλυση, αντιλαμβανόμαστε άμεσα έναν απ'τους λόγους της πολύ καλής τους απόδοσης. Άρα, η επίδοσή τους είναι ένα πολύ σημαντικό κριτήριο επιλογής για το σύστημά μας που αφορά ένα διαδικτυακό σύστημα όπου οι χρόνοι εκτέλεσης και επιστροφής αποτελεσμάτων είναι μεγάλης σημασίας. Οι χρόνοι εκτέλεσης δοκιμάστηκαν και σε server του οποίου ο φυσικός χώρος βρισκεται στην Αγγλία. Κατά την περίοδο της μελέτης των αλγορίθμων που θα χρησιμοποιήσουμε, οι item-based αλγόριθμοι για data-sets της τάξης του ενός εκατομμυρίου δεδομένων επέστρεφαν αποτελέσματα σε περίπου ένα δευτερόλεπτο, χρόνος πλήρως αποδεκτός για την ποιοτική λειτουργία ενός διαδικτυακού συστήματος. Το τελευταίο μάλιστα, είναι θετικό και ως προς την ευκολία της υλοποίησης ενός τέτοιου συστήματος, διότι εφόσον οι χρόνοι ανάλυσης τόσων μεγάλων data-sets μπορούν να διατηρούνται σε χαμηλό επίπεδο από έναν μόνο server, τότε ένα εμπορικό σύστημα μπορεί να υλοποιήσει εύκολα και

οικονομικά ένα σύστημα συστάσεων, με κύριο αρχικό στόχο την βελτιστοποίηση του αλγορίθμου που θα χρησιμοποιήσει χωρίς να ασχολείται με τις υπολογιστικές του ανάγκες. Στην συνέχεια, και αφού το σύστημα συνεχίσει την λειτουργία του με επιτυχία, θα χρειαστεί και χρήση υπολογιστικών τεχνικών που αφορούν για παράδειγμα τεχνικές κατανεμνημένων υπολογισμών σε συστάδες υπολογιστών.

Τέλος, ως προς το σύστημα που υλοποιήσαμε, παρατηρήσαμε μια δυσκολία στην εύρεση κατάλληλου data-set που να αποτυπώνει σε πραγματικό επίπεδο τα χαρακτηριστικά που θα είχε το data-set του συστήματός μας αν δεχόταν πραγματικά δεδομένα. Η διαφορά σε σχέση με το MovieLens data-set που χρησιμοποιήσαμε είναι ότι στο δικό μας data-set τα δεδομένα θα ήταν πολύ λιγότερα, μιας και ένας ταξιδιώτης ταξιδεύει λίγες φορές τον χρόνο σε σχέση με την συχνότητα αγοράς/ενοικίασης ταινιών. Παρ'όλα αυτά, υπάρχει και μια θετική όψη σε αυτή την παρατήρηση. Η μικρή διαθεσιμότητα δεδομένων σε ένα πραγματικό σενάριο, σε συνδυασμό με το γεγονός ότι οι χρήστες αλλάζουν αργά προτιμήσεις όπως αναλύσαμε παραπάνω, μας δίνει χώρο για προεπεξεργασία των δεδομένων μας και έτσι επίτευξη καλύτερων χρόνων επιστροφής αποτελεσμάτων. Βέβαια, αυτό το χαρακτηριστικό θα μπορούσαμε να το εκμεταλλευτούμε και διαφορετικά. Λίγα δεδομένα από τους χρήστες σημαίνει χώρος για επιπλέον δεδομένα από την συμπεριφορά τους στο σύστημά μας. Έτσι, θα μπορούσαμε να αναλύουμε παράλληλα με τις αξιολογήσεις και την παραμονή των χρηστών σε μία σελίδα που σχετίζεται με το περιεχόμενο ενός πακέτου. Άλλα δεδομένα προκύπτουν από τα κριτήρια αναζήτησης στα φίλτρα αναζήτησης που του παρέχουμε. Όλα αυτά μπορούν αυξήσουν την ποιότητα της πληροφορίας που έχουμε και κατ'επέκταση την πληροφορία των συστάσεων που παρέχουμε.

Βιβλιογραφία

[1] http://www.wikipedia.org

[2] http://www.php.net

[3] http://docs.oracle.com/javase/specs/

[4] Sean Owen, Robin Anil, Ted Dunning, Ellen Friedman, Mahout in Action, Manning Publications, 2012

[5] Item-based Collaborative Filtering Recommendation Algorithms, Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl, GroupLens Research Group/Army HPC Research Center Department of Computer Science and Engineering University of Minessota, Minneapolis, MN 55455

[6] Silberschhatz Abraham, Korth F. Herny, Sudarshan, Συστήματα Βάσεων Δεδομένων, Η Πλήρης Θεωρία των Βάσεων Δεδομένων, 4η έκδοση , Εκδόσεις Γκιούρδας

[7] Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3η έκδοση， Morgan Kaufmann Publications

[8] Toby Segaran, Programming Collective Intelligence, 1η έκδοση, O'Reilly publications