

# **Ανοικτό Πανεπιστήμιο Κύπρου**

**Σχολή Θετικών και Εφαρμοσμένων Επιστημών**

## **Μεταπτυχιακή Διατριβή στα Πληροφοριακά Συστήματα**



**Ανάπτυξη Δυναμικού Συστήματος Παρουσίασης Πληροφοριών  
Σχετικά με Τουριστικές Υποδομές, με Χρήση PHP Framework.**

**Στεφανία Περιστέρη**

**Επιβλέπων Καθηγητής  
Μιχαήλ Βασιλακόπουλος**

**Αύγουστος 2012**

# **Ανοικτό Πανεπιστήμιο Κύπρου**

## **Σχολή Θετικών και Εφαρμοσμένων Επιστημών**

**Ανάπτυξη Δυναμικού Συστήματος Παρουσίασης Πληροφοριών  
Σχετικά με Τουριστικές Υποδομές, με Χρήση PHP Framework.**

**Στεφανία Περιστέρη**

**Επιβλέπων Καθηγητής  
Μιχαήλ Βασιλακόπουλος**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε  
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών  
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών  
του Ανοικτού Πανεπιστημίου Κύπρου

**Αύγουστος 2012**

## Περίληψη

Τα τελευταία έτη παρατηρείται μια συνεχώς αυξανόμενη τάση των προγραμματιστών ανάπτυξης διαδικτυακών εφαρμογών, να χρησιμοποιούν ως προγραμματιστικά εργαλεία ανάπτυξης κώδικα τα Πλαίσια Εφαρμογών (Application Frameworks). Τα Frameworks συνδράμουν σημαντικά στην εύκολη και γρήγορη ανάπτυξη εφαρμογών, αλλά δεν αποτελούν πάντα την καλύτερη επιλογή για τον προγραμματιστή καθώς θέτουν και περιορισμούς.

Στόχος της παρούσας Μεταπτυχιακής Διατριβής είναι να αξιολογηθούν τα PHP frameworks ως προγραμματιστικά εργαλεία ανάπτυξης δυναμικών διαδικτυακών εφαρμογών σε σχέση με την απλή συγγραφή php κώδικα και να διερευνηθούν οι τρόποι με τους οποίους μια δυναμική διαδικτυακή εφαρμογή μπορεί να παρέχει εξατομικευμένες υπηρεσίες στους χρήστες της.

Ειδικότερα, θα εντοπιστούν και θα σχολιαστούν τα πλεονεκτήματα και τα μειονεκτήματα της συγκεκριμένης μεθόδου και θα επιβεβαιωθεί κατά πόσο συντελεί στην εύκολη ανάπτυξη εφαρμογής με δυναμικά και αλληλοδραστικά χαρακτηριστικά, ή θα προσδιοριστούν οι περιορισμοί που θέτει στο τελικό αποτέλεσμα / σύστημα.

Για τους σκοπούς της έρευνας θα αναπτυχθεί ένα ολοκληρωμένο δυναμικό σύστημα παρουσίασης πληροφοριών σχετικά με τουριστικές υποδομές. Τις πληροφορίες αυτές θα τις καταχωρούν οι εξουσιοδοτημένοι ιδιοκτήτες ξενοδοχείων προς ενημέρωση των υποψηφίων πελατών τους, εξουσιοδοτημένων και μη. Επίσης, θα λαμβάνονται υπόψη τα χαρακτηριστικά των επισκεπτών-πελατών και το ιστορικό της περιήγησης τους με στόχο την εξατομικευμένη αναζήτηση ξενοδοχείων.

# Summary

Over the last years there is a gradually increasing hype in the area of web development regarding the use Web Application Frameworks as the de facto code programming toolset. Such frameworks contribute to a fast web application development process; however they do not always constitute the optimal solution for developers due to imposed constraints.

This MSc dissertation project aims to evaluate Application Frameworks based on the PHP programming language by comparing their use on web application development with framework-less development, investigating simplicity and productivity potential for a dynamic web application that is personalized to the user requirements.

More specifically, analysis of advantages and disadvantages of Application Frameworks will be presented leading to evaluation of whether the use of frameworks can lead to enhanced productivity in the development of dynamic web applications and the types of limitations such use can present on the end result.

For the purposes of this research project a dynamic web application in the domain of hotel booking will be developed. Registered users in the hotel owner role will provide end users with notifications regarding hotel bookings. Furthermore, user registration data and browsing history of visiting users will be taken into account in order to improve hotel search functionality and personalize it to the user.

## Ευχαριστίες

Θερμές ευχαριστίες εκφράζω στον επιβλέποντα καθηγητή μου κ. Μιχαήλ Βασιλακόπουλο για την επίβλεψη, καθοδήγηση και βοήθεια που μου παρείχε για την εκπόνηση της μεταπτυχιακής διατριβής.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου, γιατί είναι πάντα δίπλα μου και με στηρίζουν σε ό,τι κάνω.

# Περιεχόμενα

<b>1 Εισαγωγή.....</b>	<b>1</b>
1.1 Αξιοποίηση Frameworks για την Ανάπτυξη Δυναμικών Διαδικτυακών Εφαρμογών.....	2
1.2 Εξατομίκευση Διεπαφής Χρήστη.....	3
1.3 Θέμα Παρούσας Μεταπτυχιακής Διατριβής και Στόχοι.....	3
<b>2 PHP Frameworks .....</b>	<b>5</b>
2.1 Ορισμός Framework.....	6
2.2 MVC Frameworks.....	6
2.3 Χαρακτηριστικά Framework .....	7
2.4 Χρήση Framework.....	8
2.5 Επιλογή PHP Framework .....	9
<b>3 Εξατομίκευση της Διεπαφής Χρήστη .....</b>	<b>11</b>
3.1 Ορισμός Εξατομίκευσης .....	12
3.2 Οι λειτουργίες της εξατομίκευσης.....	14
3.3 Περιορισμοί Εξατομίκευσης .....	15
3.4 Μηχανές Συστάσεων .....	16
3.5 Κριτήρια για την επιλογή Μηχανής Συστάσεων .....	17
<b>4 Τεχνική Περιγραφή της Υλοποίησης.....</b>	<b>22</b>
4.1 Zend Framework.....	23
4.2 PHP .....	24
4.3 MySQL.....	25
4.4 Apache Web Server .....	26
<b>5 Σχεδίαση Συστήματος Εφαρμογής.....</b>	<b>27</b>
5.1 Παρουσίαση Χρηστών .....	28

5.2	Σχεδίαση Συστήματος με Βάση τη Μεθοδολογία ICONIX.....	29
5.2.1	Διάγραμμα Περιπτώσεων Χρήσης .....	30
5.2.2	Λεκτική Περιγραφή Περιπτώσεων Χρήσης.....	32
5.2.3	Ενδεικτικές Οθόνες του Συστήματος.....	41
5.2.4.	Διαγράμματα Ευρωστίας.....	44
5.2.5	Διαγράμματα Ακολουθίας .....	50
5.2.6	Διάγραμμα Κλάσεων.....	57
5.3	Σχεδίαση Βάσης Δεδομένων .....	58
<b>6 Παρουσίαση Εφαρμογής.....</b>		<b>63</b>
6.1	Αρχική Σελίδα της εφαρμογής .....	63
6.2	Είσοδος στην εφαρμογή για εγγεγραμμένους χρήστες - Login.....	64
6.3	Εγγραφή νέου μέλους – Γίνε μέλος.....	66
6.4	Σύνδεση Ξενοδόχου στην εφαρμογή .....	67
6.4	Σύνδεση Ιδιώτη στην εφαρμογή.....	70
6.5	Σύνδεση Διαχειριστή στην εφαρμογή .....	78
6.6	Αποσύνδεση εγγεγραμμένων χρηστών από την εφαρμογή .....	79
<b>7 Σύγκριση PHP και Zend Framework .....</b>		<b>81</b>
7.1	Μελέτη Περίπτωσης - Login .....	82
7.2	Σύγκριση PHP και Zend Framework.....	92
7.3	Χρήση Framework ως προγραμματιστική μέθοδος.....	94
<b>8 Επίλογος.....</b>		<b>97</b>
<b>Βιβλιογραφία.....</b>		<b>99</b>
<b>Παράρτημα Α .....</b>		<b>A1</b>
<b>Παρουσίαση Αρχείων Εφαρμογής.....</b>		<b>A1</b>

A.1	hotel\application\configs.....	A1
A.2	hotel\application\controllers.....	A1
A.3	hotel\application\layouts\scripts .....	A2
A.4	hotel\application\models .....	A2
A.5	hotel\application\models\DbTable.....	A4
A.6	hotel\application\views\scripts\admin .....	A4
A.7	hotel\application\views\scripts\error .....	A4
A.8	hotel\application\views\scripts\hotel .....	A4
A.9	hotel\application\views\scripts\index.....	A5
A.10	hotel\application\views\scripts\user.....	A5
A.11	hotel\application .....	A6



# Κεφάλαιο 1

## Εισαγωγή

Εκατομμύρια άνθρωποι ανά τον κόσμο χρησιμοποιούν καθημερινά το Διαδίκτυο για διάφορους λόγους (περιήγηση στον Παγκόσμιο Ιστό, μεταφορά αρχείων, ηλεκτρονικό ταχυδρομείο, συμμετοχή σε ομάδες συζήτησης, ζωντανή συνομιλία, απομακρυσμένη πρόσβαση υπολογιστή, τηλεδιασκέψεις, ηλεκτρονικό εμπόριο, παιχνίδια, κ.ά.). Το μέγεθος των δεδομένων και των πληροφοριών που είναι διαθέσιμα στο Διαδίκτυο, καθώς και οι ποικίλες δυνατότητες που αυτό μπορεί να προσφέρει στους χρήστες του, οδήγησαν στο να αποκτήσει η κίνηση στο Διαδίκτυο μεγάλη εμπορική αξία.

Επίσης, είναι γεγονός ότι το ποσοστό των επιχειρήσεων που έχουν δικό τους διαδικτυακό τόπο αυξάνεται διαρκώς. Ο βασικός λόγος είναι ότι οι ιδιοκτήτες μικρών επιχειρήσεων αναζητούν καλύτερους τρόπους επικοινωνίας με τους πελάτες και τους συνεργάτες τους και το Διαδίκτυο τους παρέχει πληθώρα εξελιγμένων επιλογών. Στόχος των επιχειρήσεων είναι η διαφήμιση και η προβολή τους, η παροχή πληροφοριών και υπηρεσιών, η πώληση προϊόντων και κατ' επέκταση η βιωσιμότητα, η μεγαλύτερη ανταγωνιστικότητα και η μεγιστοποίηση των κερδών τους.

Συνεπώς, για να εκμεταλλευτούν αυτή τη συνεχώς αναπτυσσόμενη αγορά, οι προγραμματιστές διαδικτυακών εφαρμογών προσπαθούν να αναπτύξουν ιστοσελίδες που να υποστηρίζουν την

άμεση αλληλεπίδραση με τους χρήστες και που να παρέχουν μια εύχρηστη, ευχάριστη, φιλική και αποτελεσματική διεπαφή χρήστη. Επίσης, πολλές εφαρμογές σήμερα παραμετροποιούνται σύμφωνα με τις ανάγκες των χρηστών και παρέχουν εξατομικευμένες υπηρεσίες και διεπαφές χρήστη, με στόχο την ικανοποίηση των χρηστών και την αύξηση της επισκεψιμότητάς τους.

## **1.1 Αξιοποίηση Frameworks για την Ανάπτυξη Δυναμικών Διαδικτυακών Εφαρμογών**

Οι δυναμικές διαδικτυακές σελίδες ορίζονται ως οι ιστοσελίδες που δημιουργούνται την ώρα που προσπελάζονται από το χρήστη ή που το περιεχόμενό τους (κείμενο, εικόνες, υπερσύνδεσμοι, κλπ.) αλλάζει ως αποτέλεσμα της αλληλεπίδρασης με τους χρήστες. [16]

Στην εποχή μας έχουν αναπτυχθεί αρκετά αξιόπιστα PHP Frameworks που βοηθούν στη μείωση του χρόνου ανάπτυξης μιας εφαρμογής [04, 06, 13, 15] και γι' αυτόν τον λόγο ελκύουν το ενδιαφέρον πολλών προγραμματιστών δυναμικών διαδικτυακών εφαρμογών. Αυτό συμβαίνει γιατί ένα Framework είναι σχεδιασμένο με τέτοιο τρόπο ώστε να παρέχει μια δομή για όλα τα κοινά στοιχεία (database interaction, presentation layer, application logic) και να δίνεται περισσότερο έμφαση στο σχεδιασμό της ίδιας τη εφαρμογής [04, 15]. Η επιλογή, βέβαια, ενός Framework που θα είναι κατάλληλο για την εκάστοτε εφαρμογή δεν είναι καθόλου απλή διαδικασία για τον προγραμματιστή.

Από την άλλη, πολλές φορές τα Frameworks δεν αποτελούν αναγκαιότητα για την ανάπτυξη μιας εφαρμογής [15]. Αυτό συμβαίνει κυρίως όταν δεν εξοικονομείται χρόνος ή προσπάθεια από την πλευρά του προγραμματιστή και γενικότερα όταν αυτή η προγραμματιστική μέθοδος δυσκολεύει την ανάπτυξη μιας εφαρμογής. Επίσης, παρατηρείται ότι ο προγραμματιστής πρέπει να επενδύσει αρκετό χρόνο στη μελέτη μιας μεθοδολογίας ενός συγκεκριμένου Framework [13] ώστε να γίνει πιο παραγωγικός και αποδοτικός στη συγγραφή κώδικα για διαδικτυακές εφαρμογές και αυτό αποτελεί σημαντικό περιοριστικό παράγοντα στην υιοθέτηση μιας τέτοιας μεθόδου ανάπτυξης συστημάτων.

## 1.2 Εξατομίκευση Διεπαφής Χρήστη

Μεγάλη πρόκληση σήμερα είναι η διασφάλιση της ικανοποίησης του χρήστη μετά την πλοήγηση σε ένα διαδικτυακό τόπο, καθώς αυτή θα κρίνει την εμπορική επιτυχία της διεπαφής χρήστη και θα προσελκύσει το ενδιαφέρον των χρηστών [05]. Η ικανοποίηση του χρήστη επιτυγχάνεται όταν το σύστημα του παρουσιάζει την πληροφορία που τον ενδιαφέρει στο σωστό χρόνο, χωρίς αυτός να χάνει τον στόχο του.

Η εξατομίκευση της διεπαφής χρήστη βρίσκει εφαρμογή σήμερα στο ηλεκτρονικό εμπόριο, στις πληροφοριακές πύλες, στα περιβάλλοντα ηλεκτρονικής μάθησης και στις μηχανές αναζήτησης. Τέλος, η εξατομίκευση κρίνεται απαραίτητη για τη συλλογή και στη συνέχεια διαμοίραση υψηλής ποιότητας δεδομένων στους διαδικτυακούς χρήστες, αλλά υπάρχουν πολλές προκλήσεις που πρέπει να αντιμετωπιστούν [01, 07].

Η ανάπτυξη εφαρμογών εξατομίκευσης, οι οποίες καλούνται να ανταποκριθούν σε συγκεκριμένες ανάγκες χρηστών, είναι μια προκλητική εργασία κυρίως διότι περιλαμβάνει την αντιμετώπιση πλήθους ζητημάτων όπως η εγγύηση της ασφάλειας και η διαφύλαξη της ανωνυμίας στα δεδομένα που συλλέγονται για το χρήστη, η ευχρηστία των εφαρμογών, η εξασφάλιση της δυναμικότητας των αποφάσεων που λαμβάνονται, η παροχή συστάσεων επισκόπησης ανάλογα με τα προφίλ των χρηστών ή ακόμα και ομάδων χρηστών, κτλ.

## 1.3 Θέμα Παρούσας Μεταπτυχιακής Διατριβής και Στόχοι

Ο τίτλος της παρούσας μεταπτυχιακής διατριβής είναι: «Ανάπτυξη Δυναμικού Συστήματος Παρούσας Πληροφοριών Σχετικά με Τουριστικές Υποδομές, με Χρήση PHP Framework». Γενικός σκοπός της μεταπτυχιακής διατριβής είναι η διερεύνηση, αξιοποίηση και ποιοτική αξιολόγηση σύγχρονων τάσεων στην ανάπτυξη ολοκληρωμένων διαδικτυακών εφαρμογών.

Πιο αναλυτικά, στα πλαίσια της μεταπτυχιακής διατριβής θα γίνει σχεδίαση και ανάπτυξη μιας ολοκληρωμένης διαδικτυακής εφαρμογής που θα παρουσιάζει πληροφορίες σχετικά με τουριστικές υποδομές στην Ελλάδα. Η εφαρμογή αυτή θα

υποστηρίζεται από κατάλληλα σχεδιασμένη και υλοποιημένη Βάση Δεδομένων που θα εξυπηρετεί τους σκοπούς του συστήματος.

Για την ανάπτυξη της εφαρμογής θα γίνει χρήση της PHP και ενός PHP framework, με στόχο να παρουσιαστούν και να συγκριθούν η λειτουργικότητα και τα αποτελέσματα των δύο αυτών μεθόδων ανάπτυξης του συστήματος. Κατά την εκπόνηση της μεταπτυχιακής διατριβής θα εντοπιστούν και θα σχολιαστούν τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης ενός PHP Framework ως μεθόδου ανάπτυξης και θα επιβεβαιωθεί κατά πόσο συντελεί στην εύκολη ανάπτυξη εφαρμογής με δυναμικά και αλληλοδραστικά χαρακτηριστικά, ή θα προσδιοριστούν οι περιορισμοί που θέτει στο τελικό αποτέλεσμα / σύστημα.

Επίσης, σε πολλά εμπορικά συστήματα γίνεται προσπάθεια αξιοποίησης του προφίλ του χρήστη για την εξατομίκευση των αποτελεσμάτων αναζήτησης, εστιάζοντας στην εύρεση των ενδιαφερόντων του χρήστη από την ανάλυση του ιστορικού περιήγησής του. Ομοίως, στην εφαρμογή θα ενσωματωθούν χαρακτηριστικά συνεχούς καταγραφής του προφίλ του χρήστη και αξιοποίησης του προφίλ αυτού κατά τη χρήση της εφαρμογής, με προσδοκώμενο αποτέλεσμα τη δημιουργία ενός συστήματος με εξατομίκευση των παρεχόμενων υπηρεσιών και της διεπαφής χρήστη.

# Κεφάλαιο 2

## PHP Frameworks

Η PHP είναι η πιο δημοφιλής scripting γλώσσα προγραμματισμού για την ευελιξία και ευκολία που προσφέρει στους χρήστες. Όμως, πολλές φορές η χρήση της PHP μετατρέπει την ανάπτυξη από μια δημιουργική σε μια μονότονη διαδικασία για τους προγραμματιστές. [14]

Τα PHP Frameworks παρέχουν μια βασική δομή για την ανάπτυξη web εφαρμογών. Με αυτό τον τρόπο προωθείται η ταχεία ανάπτυξη εφαρμογών, εξοικονομείται χρόνος, αναπτύσσονται πιο σταθερές εφαρμογές και μειώνεται το ύψος της συγγραφής επαναλαμβανόμενου κώδικα για τους προγραμματιστές. Συνεπώς, οι προγραμματιστές μπορούν να επενδύσουν περισσότερο χρόνο στην ανάπτυξη της πραγματικής web εφαρμογής.

Τα PHP Frameworks αποτελούν έναν πολύ καλό τρόπο για την ανάπτυξη διαδικτυακών εφαρμογών με στόχο την μείωση επαναλαμβανόμενης κωδικοποίησης, την επιτάχυνση της αναπτυξιακής διαδικασίας και την εξασφάλιση ορθής κωδικοποίησης. Η χρήση αυτών έχει ως συνέπεια την επιτάχυνση της ανάπτυξης εφαρμογών και την ενίσχυση της ασφάλειας.

Η γενική ιδέα πίσω από τη λειτουργία ενός PHP Framework αναφέρεται ως MVC- Model View Controller (Μοντέλο Προβολή Ελεγκτής), όπου Model αναφέρεται στο μοντέλο δεδομένων,

Viewer αναφέρεται στην παρουσίαση και Controller αναφέρεται στη λογική της εφαρμογής. Χρησιμοποιώντας ένα MVC Framework, η ανάπτυξη μιας web εφαρμογής μοιράζεται σε επιμέρους διαδικασίες που δεν επηρεάζονται μεταξύ τους. Ουσιαστικά, αυτό κάνει την κωδικοποίηση σε PHP ταχύτερη και λιγότερο πολύπλοκη.

## 2.1 Ορισμός Framework

Ένα Framework είναι ένας επαναχρησιμοποιήσιμος σχεδιασμός του συνόλου ή μέρους ενός συστήματος, που αντιπροσωπεύεται από ένα σύνολο αφηρημένων κλάσεων και από τον τρόπο που αλληλεπιδρούν τα στιγμιότυπά τους (Δομή του Framework). Ένα Framework είναι ο σκελετός μιας εφαρμογής που μπορεί να προσαρμοστεί από έναν προγραμματιστή εφαρμογών (Σκοπός του Framework) [04].

Η βασική ιδέα κάθε Framework είναι να προσφέρει μια σχεδίαση που να είναι εφικτό να χρησιμοποιηθεί και να εφαρμοστεί σε διάφορες εφαρμογές. Τα κοινά στοιχεία που εντοπίζονται στις εφαρμογές που αναπτύσσονται είναι η αλληλεπίδραση με τη Βάση Δεδομένων, στοιχεία που αφορούν την λογική που χρησιμοποιεί η εφαρμογή και τέλος η διεπαφή που προβάλλει τα δεδομένα στο χρήστη.

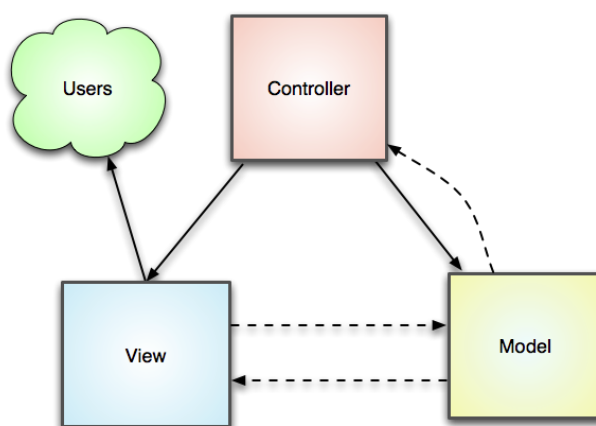
## 2.2 MVC Frameworks

Το MVC είναι μία αρχιτεκτονική προγραμματισμού που βασίζεται στο διαχωρισμό κώδικα λογικής από τον κώδικα εμφάνισης και αλληλεπίδρασης με το χρήστη [17]. Αυτή η αρχιτεκτονική αν υιοθετηθεί επιτρέπει καλύτερη οργάνωση του κώδικα, καθώς και εύκολη συντήρηση του κώδικα. Επίσης επιτρέπει σε πολλούς προγραμματιστές να δουλεύουν στο ίδιο σύστημα ταυτόχρονα, καθώς η ανάπτυξη και η αποσφαλμάτωση κάθε κομματιού κώδικα εκτελείται σαν ξεχωριστή διαδικασία.

Το MVC πρότυπο διαχωρίζει τη διεπαφή χρήστη σε ένα σύνολο από “View” αντικείμενα, που εστιάζουν στην παρουσίαση πληροφορίας και στην αλληλεπίδραση με το χρήστη μέσω των εντολών που στέλνει. Οι εντολές του χρήστη στέλνονται στα “Controller” αντικείμενα, τα οποία χειρίζονται τις εισόδους του χρήστη και τις απαντήσεις του συστήματος. Τα “Model” αντικείμενα αντιπροσωπεύουν την επιχειρησιακή λογική, δηλαδή την αναπαράσταση ή υλοποίηση της κεντρικής δομής της εφαρμογής και είναι

υπεύθυνα για την αποθήκευση, ανάκληση και τροποποίηση των δεδομένων. Μία MVC εφαρμογή είναι ένα σύνολο από Controllers, Models και Views που αλληλεπιδρούν μεταξύ τους [17].

Στην Εικόνα 2.1 φαίνεται η αλληλεπίδραση των Controller, Model και View. Ο Controller αλληλεπιδρά άμεσα με το Model και το View και δένει αυτά τα δύο μέρη μεταξύ τους, ενώ οι χρήστες του συστήματος αλληλεπιδρούν μόνο με τα Views, μέσω των οποίων δέχονται και στέλνουν πληροφορία.



**Εικόνα 2.1:** Πρότυπο Model View Controller [17]

Ανάλογα με το περιβάλλον ανάπτυξης στο οποίο το Framework λειτουργεί, τι συνιστά τα αντικείμενα που αλληλεπιδρούν εντός του προτύπου μπορεί να αλλάξει δραματικά. Για παράδειγμα, στο παράδειγμα μιας απλής ιστοσελίδας, το model θα είναι ο πηγαίος κώδικας, το view είναι η HTML και το CSS, που καθορίζει το πώς το μοντέλο πρέπει να εμφανίζεται, και ο web browser ενεργεί ως controller, που καθορίζει πως θα χειριστεί την είσοδο του χρήστη όταν επιλέγει έναν υπερσύνδεσμο. Σε μια διαδικτυακή εφαρμογή, όπου τα δεδομένα αλλάζουν δυναμικά από το χρήστη, το model αποτελείται από τις κλάσεις που χρησιμοποιούνται για αποθήκευση και χειρισμό δεδομένων σε μια βάση δεδομένων.

## 2.3 Χαρακτηριστικά Framework

Πολλοί προγραμματιστές καλούνται να επιλέξουν ένα Framework για την ανάπτυξη των εφαρμογών τους ή ακόμα και να δημιουργήσουν οι ίδιοι το δικό τους Framework σύμφωνα με

τις ανάγκες της εφαρμογής. Τα χαρακτηριστικά ενός αποδοτικού Framework που πρέπει να ληφθούν υπόψη και να αξιολογηθούν είναι [04] :

1. Επεκτασιμότητα : Ένα πλαίσιο εφαρμογών πρέπει να είναι αρκετά ευέλικτο ώστε να επιτρέπει στους προγραμματιστές να αναπτύσσουν λύσεις σε προβλήματα που η ενσωματωμένη λειτουργικότητα του πλαισίου δεν μπορεί να αντιμετωπίσει.
2. Ικανότητα ενσωμάτωσης : Τα πλαίσια πρέπει να παρέχουν επαρκή αποσύνδεση της παρουσίασης από τον έλεγχο, έτσι ώστε η εφαρμογή μπορεί να διασυνδεθεί εύκολα με άλλους χρήστες ή εφαρμογές.
3. Απόδοση - Λειτουργικότητα : Όλα τα πλαίσια εφαρμογών παρουσιάζουν ένα tradeoff μεταξύ της λειτουργικότητας και της απόδοσης.
4. Ωριμότητα : Τα Frameworks εξελίσσονται μέσω ενός αριθμού επαναληπτικών εκδόσεων, που οφείλεται στην ενσωμάτωση νέων απαιτήσεων που εμφανίζονται.
5. Καμπύλη μάθησης : Παρόλο που ο σκοπός ενός πλαισίου είναι να εξαλείψει την επαναληπτική και μονότονη κωδικοποίηση, οι προγραμματιστές πρέπει πρώτα να μάθουν τη δομή και τις ιδιαιτερότητες του πλαισίου, προκειμένου να το χρησιμοποιήσουν. Η καταλληλότητα, όμως, ενός πλαισίου για μια συγκεκριμένη εφαρμογή μπορεί να μην είναι εμφανής μέχρι η καμπύλη μάθησης να είναι ευθεία.
6. Τεκμηρίωση και Υποστήριξη : Τέλος, η τεκμηρίωση είναι ζωτικής σημασίας για την επιτυχία ενός πλαισίου. Για το λόγο αυτό, μια ενεργή κοινότητα προγραμματιστών είναι σημαντική για την επίκαιρη διατήρηση του πλαισίου και την ανάπτυξη με νέους τρόπους αντιμετώπισης των προβλημάτων που αντιμετωπίζουν οι χρήστες.

## 2.4 Χρήση Framework

Οι προγραμματιστές θα πρέπει να χρησιμοποιούν τα PHP Frameworks για διάφορους λόγους, αλλά ο κυριότερος λόγος είναι για την επιτάχυνση της αναπτυξιακής διαδικασίας. Επαναχρησιμοποίηση κώδικα σε παρόμοιες εφαρμογές θα συμβάλλει σημαντικά στην εξοικονόμηση χρόνου και προσπάθειας.



Η σταθερότητα και η απλότητα είναι άλλα δύο μεγάλα πλεονεκτήματα. Είναι αρκετά εύκολο για έναν προγραμματιστή, ιδιαίτερα για τους αρχάριους, να γράψει κακής ποιότητας κώδικα σε PHP και η εφαρμογή να εξακολουθεί να λειτουργεί. Όμως, έτσι, εν αγνοία τους μπορεί η εφαρμογή να μην είναι ασφαλής με κίνδυνο την απώλεια σημαντικών δεδομένων.

Τέλος, υπάρχουν πολλά διαθέσιμα PHP Frameworks, ώστε ο χρήστης να επιλέξει το κατάλληλο για την εφαρμογή του. Τα περισσότερα από αυτά διαθέτουν μεγάλες ομάδες υποστήριξης, που επιτρέπουν την αλληλεπίδραση μεταξύ των προγραμματιστών που χρησιμοποιούν το ίδιο Framework.

Για πολλούς αρχάριους, ένα Framework θα προσφέρει μεγαλύτερη απλότητα και σταθερότητα. Επίσης, θα βοηθήσει στη μείωση ή την εξάλειψη της κακής συγγραφής κώδικα και θα επιταχύνει τη διαδικασία ανάπτυξης. Στους έμπειρους προγραμματιστές, ένα Framework θα προσφέρει εξοικονόμηση χρόνου και θα συνεισφέρει στο να γίνει η εργασία τους λιγότερο μονότονη και κουραστική.

## 2.5 Επιλογή PHP Framework




















Υπάρχει αφθονία επιλογών διαθέσιμων PHP Frameworks. Το ποιο Framework είναι κατάλληλο για κάθε προγραμματιστή εξαρτάται κυρίως από την εφαρμογή που θέλει να αναπτύξει. Επίσης, ένα άλλο κριτήριο που πρέπει να ληφθεί υπόψη κατά την επιλογή είναι ότι εάν η ομάδα ανάπτυξης αποτελείται από πολλά άτομα τότε ενδεχομένως η χρήση ενός δημοφιλούς Framework είναι η καλύτερη επιλογή καθώς πολλοί προγραμματιστές μπορεί να είναι εξοικειωμένοι με αυτό.

Διάφοροι παράγοντες που πρέπει να συνυπολογιστούν κατά την επιλογή είναι: ευκολία χρήσης, ταχεία ανάπτυξη και απόδοση, δημοτικότητα στους προγραμματιστές, ισχυρά χαρακτηριστικά και υποστήριξη/φόρουμ. Η δοκιμή διαφόρων Framework θα βοηθήσει τον προγραμματιστή να βρει αυτό που ταιριάζει καλύτερα στις ανάγκες του. Κάθε Framework υπερτερεί ή υστερεί σε κάποιο χαρακτηριστικό σε σχέση με τα υπόλοιπα. Παραδείγματος χάρη, το Zend Framework προσφέρει πολλές δυνατότητες και ένα εκτενές σύστημα υποστήριξης στους προγραμματιστές που το επιλέγουν ως βασικό εργαλείο ανάπτυξης. Από την άλλη, το CakePHP προσφέρει ένα ασθενέστερο σύστημα υποστήριξης, αλλά είναι πολύ πιο φιλικό και εύκολο στη χρήση.

Επίσης, πρέπει να δοθεί ιδιαίτερη προσοχή κατά την επιλογή γιατί υπάρχουν πολλά Frameworks που δεν παρέχουν υποστήριξη ή έχουν δημιουργηθεί από άτομα με περιορισμένη γνώση της PHP. Η χρήση αυτών μπορεί να προκαλέσει τη μη σωστή λειτουργία των εφαρμογών ή σημαντικά προβλήματα ασφάλειας.

Τέλος, η συμβατότητα των Frameworks με τη Βάση Δεδομένων και τον web server απαιτεί να δοθεί έμφαση στις απαιτήσεις και στην τεκμηρίωση του Framework.

Στην Εικόνα 2.2 συγκρίνονται ως προς τα χαρακτηριστικά τους, μερικά από τα πιο δημοφιλή PHP Frameworks και από τη σύγκριση διαπιστώνουμε ότι καθένα από αυτά υπερτερεί ή υστερεί σε κάποιο χαρακτηριστικό. Τα Frameworks που συγκρίνονται είναι τα εξής: Akelos, ash.MVC, CakePHP, CodeIgniter, DIY, eZ Components, Fusebox, PHP on TRAX, PHPDevShell, PhpOpenbiz, Prado, QPHP, Seagull, Symfony, WACT, WASP, Yii, Zend και ZooP. Τα χαρακτηριστικά ως προς τα οποία γίνεται η σύγκριση είναι: PHP4, PHP5, MVC (Model View Controller υποστήριξη), Multiple DBs (υποστήριξη πολλαπλών Βάσεων Δεδομένων), ORM (Object-Record Mapper υποστήριξη), DB Objects (συμπερίληψη όλων των αντικειμένων των Βάσεων Δεδομένων), Templates (ενσωματωμένη μηχανή template), Caching (υποστήριξη caching), Validation (ενσωματωμένο validation), Ajax (υποστήριξη ajax), Auth Module (υποστήριξη πιστοποίησης χρηστών), Modules (ενσωμάτωση λειτουργιών όπως RSS feeds, PDF κλπ.), EDP (υποστήριξη Event Driven Programming).

PHP Framework	PHP4	PHP5	MVC	Multiple DB's	ORM	DB Objects	Templates	Caching	Validation	Ajax	Auth Module	Modules	EDP
<a href="#">Akelos</a> 	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
<a href="#">ash.MVC</a> 	-	✓	✓	-	-	✓	✓	-	✓	-	✓	✓	-
<a href="#">CakePHP</a> 	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-
<a href="#">CodeIgniter</a> 	✓	✓	✓	✓	-	✓	✓	✓	✓	-	-	-	-
<a href="#">DIY</a> 	-	✓	✓	-	✓	✓	✓	✓	-	✓	-	-	-
<a href="#">eZ Components</a> 	-	✓	-	✓	-	✓	✓	✓	✓	-	-	-	-
<a href="#">Fusebox</a> 	✓	✓	✓	✓	-	-	-	✓	-	✓	-	✓	-
<a href="#">PHP on TRAX</a> 	-	✓	✓	✓	✓	✓	-	-	✓	✓	-	✓	-
<a href="#">PHPDevShell</a> 	-	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-
<a href="#">PhpOpenbiz</a> 	-	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	-
<a href="#">Prado</a> 	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">QPHP</a> 	✓	✓	✓	✓	-	✓	✓	-	✓	✓	✓	✓	✓
<a href="#">Seagull</a> 	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
<a href="#">Symfony</a> 	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-
<a href="#">WACT</a> 	✓	✓	✓	✓	-	✓	✓	-	✓	-	-	✓	-
<a href="#">WASP</a> 	-	✓	✓	-	-	✓	✓	-	✓	✓	✓	✓	-
<a href="#">Yii</a> 	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">Zend</a> 	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
<a href="#">ZooP</a> 	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	-	-

Εικόνα 2.2: Σύγκριση δημοφιλών PHP Frameworks [15]

# Κεφάλαιο 3

## Εξατομίκευση της Διεπαφής Χρήστη

Η συνεχής αύξηση του μεγέθους και της χρήσης του Παγκοσμίου Ιστού επιβάλλει νέες μεθόδους σχεδιασμού και ανάπτυξης των υπηρεσιών παροχής πληροφοριών. Οι περισσότερες δομές Ιστού είναι σύνθετες και περίπλοκες και οι χρήστες χάνουν συχνά το στόχο της έρευνάς τους, ή λαμβάνουν διαφορετικά αποτελέσματα όταν προσπαθούν να πλοηγηθούν μέσω αυτών. Από την άλλη πλευρά, ο τομέας του ηλεκτρονικού εμπορίου εξελίσσεται γρήγορα και οι ανάγκες για ύπαρξη αγορών μέσω του Ιστού που θα περιλαμβάνουν τις ανάγκες των πελατών είναι περισσότερο εμφανέστερη από ποτέ άλλοτε.

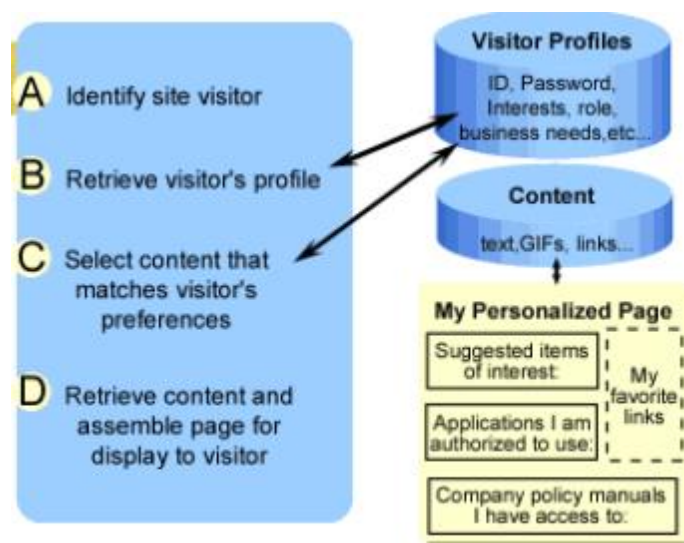
Η διασφάλιση της ικανοποίησης του χρήστη από τα αποτελέσματα της πλοήγησης απαιτεί την αξιοποίηση τεχνικών παραμετροποίησης και εξατομίκευσης στα δεδομένα και στις υπηρεσίες των ιστοτόπων. Ο σκοπός της εξατομίκευσης είναι να παρέχει στους χρήστες πληροφορίες που θέλουν ή χρειάζονται, χωρίς να αναμένει από αυτούς να τις ζητάνε ρητά. Οι δύο πιο κοινές προσεγγίσεις για την επίτευξη της εξατομίκευσης είναι η εξατομίκευση της δομής και η εξατομίκευση του περιεχομένου.

Η εξατομίκευση της δομής περιλαμβάνει τη μεταβολή της τοποθεσίας των διαθέσιμων υπερσυνδέσμων καθώς και της τοποθεσίας του περιεχομένου και στοχεύει στην παροχή περισσότερων και πιο σχετικών με τα ενδιαφέροντα του χρήστη συνδέσμων με την ταυτόχρονη προβολή τους σε σημείο της διεπαφής που είναι εύκολα αντιληπτό από το χρήστη.

Η εξατομίκευση του περιεχομένου αφορά στην αλλαγή στα δεδομένα του ιστότοπου ανάλογα με τα ενδιαφέροντα του χρήστη ως προς τη θεματολογία τους. Για παράδειγμα ο ιστότοπος myyahoo.com παρέχει τη δυνατότητα εύκολης αλλαγής του περιεχομένου σύμφωνα με τις προτιμήσεις των χρηστών με τρόπο τέτοιο ώστε κάποιος να μπορεί να επιλέγει και να διαγράφει σχετικά θέματα ή να εξειδικεύει σε υποκατηγορίες των γενικών θεμάτων, ακόμα και να προτείνει νέα θέματα ενδιαφέροντος για μελλοντική συμπερίληψή τους στις θεματικές κατηγορίες των ιστότοπων.

### 3.1 Ορισμός Εξατομίκευσης

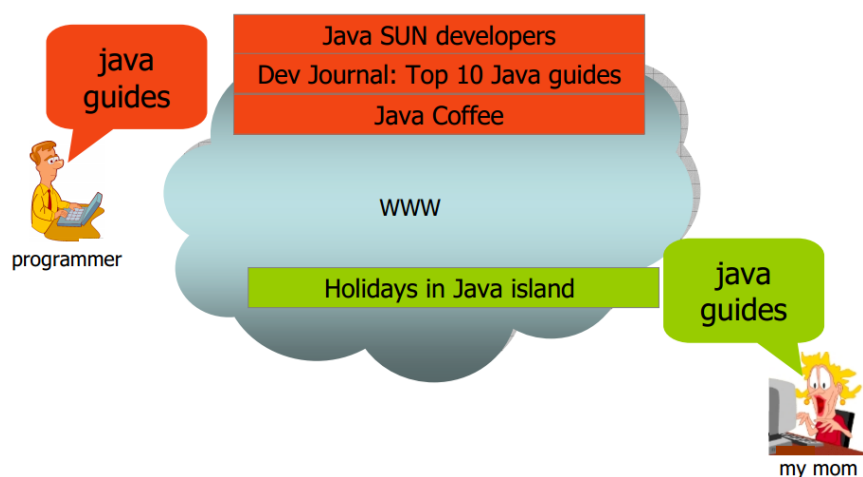
«Η διαδικασία της συγκέντρωσης και αποθήκευσης πληροφοριών αναφορικά με τους χρήστες ενός website, η ανάλυση των πληροφοριών αυτών και, με βάση την ανάλυση, η αποστολή σε κάθε χρήστη της σωστής πληροφορίας στο σωστό χρόνο.» [08]



Εικόνα 3.1: Διαδικασία εξατομίκευσης [08]

Στην Εικόνα 3.1 φαίνεται η διαδικασία της εξατομίκευσης. Το σύστημα αρχικά αναγνωρίζει τον επισκέπτη και ανακτά το προσωπικό του προφίλ, το οποίο χτίζει σταδιακά είτε από πληροφορίες που έχει δώσει ο ίδιος στο σύστημα, όπως π.χ. προσωπικός κωδικός σύνδεσης στο σύστημα που επιτρέπει την αναγνώρισή του, είτε από πληροφορίες που το σύστημα έχει εκμαιεύσει έμμεσα, όπως π.χ. ποια link επέλεξε ο χρήστης κατά την περιήγησή του. Έπειτα, το σύστημα βρίσκει εμφανίζει το περιεχόμενο ή/και τη δομή της ιστοσελίδας που ταιριάζει και ανταποκρίνεται σε μεγαλύτερο βαθμό στις ανάγκες και στα ενδιαφέροντα του κάθε χρήστη και πολλές φορές προτείνει στο χρήστη υπηρεσίες, προϊόντα ή ενέργειες.

«Οποιαδήποτε ενέργεια που προσαρμόζει την πληροφορία ή τις υπηρεσίες που παρέχονται από ένα website στη γνώση που κερδίζουμε από την πλοηγική συμπεριφορά των χρηστών και τα ατομικά τους ενδιαφέροντα, σε συνδυασμό με το περιεχόμενο και τη δομή του website.» [08]



**Εικόνα 3.2:** Αναγνώριση χρήστη και προσαρμογή περιεχομένου ή παρεχόμενων υπηρεσιών εξατομικευμένων ιστοσελίδων ή μηχανών αναζήτησης [08]

Στην Εικόνα 3.2 φαίνεται η διαδικασία της εξατομικευμένης αναζήτησης, όπου το σύστημα γνωρίζει πολύ καλά ποιοι είναι και τι θέλουν οι χρήστες του και τους το προσφέρει γρήγορα και άμεσα, χωρίς να κάνουν άσκοπες αναζητήσεις και να χάνουν το χρόνο τους.

Το πεδίο εφαρμογής της εξατομίκευσης είναι ευρύ, καθώς αυξάνονται συνεχώς οι απαιτήσεις των χρηστών. Μερικοί σημαντικοί τομείς όπου εφαρμόζεται εξατομίκευση είναι οι εξής:

1. Ηλεκτρονικό εμπόριο: προτάσεις για προϊόντα, εξατομικευμένη προώθηση, τιμολόγηση, διαμόρφωση της αρχικής σελίδας των ηλεκτρονικών καταστημάτων, κ.α.
2. Πληροφοριακές πύλες (portals): με τη μορφή π.χ. της παραμετροποίησης της αρχικής σελίδας όπως στο my.yahoo.com
3. Περιβάλλοντα ηλεκτρονικής μάθησης (e-learning): προτάσεις μαθημάτων, ειδικές διαμορφώσεις για μαθητές, εκπαιδευτές και διαχειριστές, προσαρμογές του εκπαιδευτικού περιεχομένου με βάση τις γνώσεις και δεξιότητες κάθε μαθητή, κτλ.
4. Μηχανές αναζήτησης (search engines): αποτελέσματα που φιλτράρονται ή/και ταξινομούνται σύμφωνα με το προφίλ κάθε χρήστη

## **3.2 Οι λειτουργίες της εξατομίκευσης**

Το εξατομικευμένο σύστημα απομνημονεύει το προφίλ του χρήστη, καταγράφοντας πληροφορίες τις οποίες χρησιμοποιεί για να ανακαλέσει τη συμπεριφορά του. Στην περιοχή Ιστού, η σκιαγράφηση χρηστών (δημιουργία προφίλ χρηστών) είναι η διαδικασία της συγκέντρωσης της πληροφορίας για κάθε επισκέπτη, είτε στατικά είτε δυναμικά. Οι παράμετροι χρήστη περιλαμβάνουν τις δημογραφικές πληροφορίες για τον χρήστη, τα ενδιαφέροντά του ακόμη και τη συμπεριφορά του κατά την περιήγηση σε έναν ιστότοπο. Οι πληροφορίες αυτές αποθηκεύονται συνήθως σε Βάσεις Δεδομένων και αξιοποιούνται προκειμένου να προσαρμόσουν το περιεχόμενο και τη δομή ενός ιστοχώρου στις ανάγκες του επισκέπτη. Παραδείγματα απομνημόνευσης των εξατομικευμένων διαδικτυακών εφαρμογών είναι ο χαιρετισμός του χρήστη, bookmarking και παροχή εξατομικευμένων δικαιωμάτων πρόσβασης.

Το εξατομικευμένο σύστημα καθοδηγεί τις ενέργειες του χρήστη. Βοηθά το χρήστη να εντοπίσει γρήγορα την πληροφορία που χρειάζεται και του παρέχει εναλλακτικές επιλογές πλοήγησης μέσω σύστασης σχετικών συνδέσμων. Η πληροφορία μ' αυτόν τον τρόπο προσφέρεται γρήγορα και εύκολα στο χρήστη, χωρίς αυτός να χάνει χρόνο σε αναζητήσεις. Έτσι, δίνεται η εντύπωση στο χρήστη ότι το σύστημα γνωρίζει τις ανάγκες και τις επιθυμίες του και προσπαθεί να τις ικανοποιήσει με τον καλύτερο δυνατό τρόπο.

Το εξατομικευμένο σύστημα παραμετροποιείται σύμφωνα με τις προτιμήσεις και τα ενδιαφέροντα του χρήστη. Η παραμετροποίηση μπορεί να εφαρμοστεί στο περιεχόμενο (π.χ. διαφοροποίηση προϊόντων), στη δομή (π.χ. εξατομικευμένο layout) και στην εμφάνιση της ιστοσελίδας (π.χ. επιλογή θέματος ή χρωμάτων).

Το εξατομικευμένο σύστημα υποστηρίζει διεκπεραίωση εργασιών για λογαριασμό του χρήστη. Παράδειγμα αυτής της λειτουργίας είναι οι εξατομικευμένες μηχανές αναζήτησης που συμπληρώνουν ή βελτιώνουν τις ερωτήσεις που υποβάλει ο χρήστης.

### **3.3 Περιορισμοί Εξατομίκευσης**

Υπάρχουν πολλοί περιοριστικοί παράγοντες κατά την εφαρμογή της εξατομίκευσης από ένα σύστημα [08]:

1. Διασφάλιση προσωπικού απορρήτου: Υπάρχει ανησυχία από την πλευρά των χρηστών για την εμπιστευτικότητα και τη διαφύλαξη των προσωπικών δεδομένων.
2. Χαμηλή ανεκτικότητα σε καθυστερήσεις: Οι χρόνοι απόκρισης των διεργασιών εξόρυξης δεδομένων πρέπει να είναι μικροί και γι' αυτό μέρος της διαδικασίας πρέπει να γίνεται off-line, αν αυτό είναι εφικτό.
3. Αποτίμηση της αποτελεσματικότητας της εξατομίκευσης: Αξίζει η επένδυση σε χώρο και υπολογιστική ισχύ που απαιτεί η παροχή υπηρεσιών εξατομίκευσης; Βοήθησαν τους χρήστες και ήταν ακριβείς;

4. Πρόβλημα απώλειας ελέγχου / Ενόχληση χρήστη: Η εξατομίκευση μπορεί σε ορισμένες περιπτώσεις να μειώσει την ευχρηστία ή να οδηγήσει σε λάθος παρουσίαση του προφίλ του χρήστη.

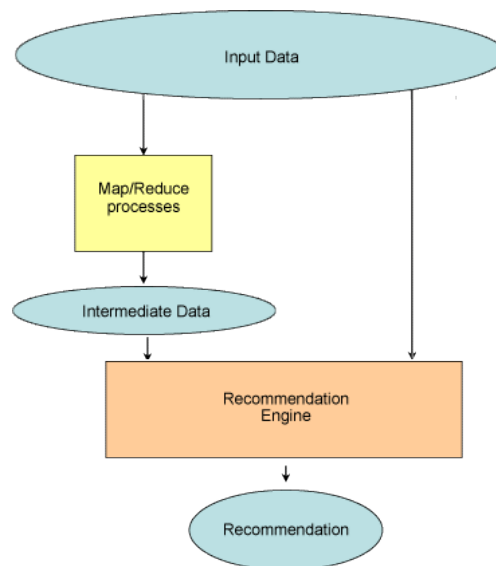
### 3.4 Μηχανές Συστάσεων

Οι Μηχανές Συστάσεων (Recommendation Engines) οδηγούν τους χρήστες στο να ανακαλύψουν περαιτέρω προτάσεις από ένα website ή από μια επιχείρηση. Αυτές οι μηχανές παρέχουν συστάσεις στους χρήστες και είναι χρήσιμες στην καθοδήγηση των χρηστών ώστε να εξετάσουν προτάσεις που ειδάλλως να μην γνώριζαν και βασίζονται στις συνήθειες αυτών.

Ο κύριος σκοπός μιας μηχανής συστάσεων είναι να αναλυθούν τα υπάρχοντα δεδομένα ώστε να παρουσιαστούν οι σχέσεις μεταξύ των αντικειμένων. Τα αντικείμενα μπορούν να είναι πολλά πράγματα, συμπεριλαμβανομένων των χρηστών, στοιχείων, προϊόντων, κ.ά. Οι σχέσεις παρέχουν έναν βαθμό ομοιότητας ή κτήσης μεταξύ των αντικειμένων. Παραδείγματος χάριν, οι σχέσεις μπορούν να αντιπροσωπεύσουν τις βαθμολογήσεις του κατά πόσο σε ένα χρήστη αρέσει ένα στοιχείο ή δείχνουν εάν ένας χρήστης ξεχώρισε και επισκέπτεται συχνά μια σελίδα.

Για να υποβάλουν μια σύσταση, οι μηχανές συστάσεων εκτελούν διάφορα βήματα για να εξαγάγουν τα δεδομένα. Στην Εικόνα 3.3 φαίνεται η διαδικασία εξαγωγής συστάσεων. Ο χρήστης εισάγει έμμεσα ή άμεσα δεδομένα στο σύστημα. Τα δεδομένα αυτά αποτελούν την είσοδο για τη μηχανή συστάσεων. Η έξοδος της μηχανής συστάσεων είναι η σύσταση που αφορά το συγκεκριμένο χρήστη.





**Εικόνα 3.3:** Διαδικασία εξαγωγής συστάσεων [07]

Στη συνέχεια αναλύουμε τη διαδικασία εξαγωγής συστάσεων μέσω ενός παραδείγματος. Έστω ότι οι χρήστες δίνουν βαθμολογίες στα στοιχεία. Χρησιμοποιώντας αυτό το δεδομένο εισόδου, μια μηχανή συστάσεων υπολογίζει την ομοιότητα μεταξύ των αντικειμένων. Ο υπολογισμός της ομοιότητας μεταξύ των αντικειμένων μπορεί να πάρει πολύ χρόνο ανάλογα με το μέγεθος των στοιχείων ή του αλγορίθμου που χρησιμοποιείται. Υπάρχουν διαφορετικοί τύποι αλγορίθμων για να υπολογίσουν τις ομοιότητες. Τέλος, χρησιμοποιώντας τις πληροφορίες ομοιότητας, η μηχανή συστάσεων μπορεί κάνει συστάσεις στους χρήστες.

Οι μηχανές συστάσεων είναι χρήσιμες, αλλά είναι δύσκολο να αναπτυχθούν. Η περιοχή των μηχανών συστάσεων είναι στην πραγματικότητα μεγάλη και μπορεί να είναι πολύ σύνθετη. Ευτυχώς, υπάρχουν εργαλεία διαθέσιμα που το καθιστούν εύκολο να παρουσιαστούν οι απαραίτητες έννοιες.

### 3.5 Κριτήρια για την επιλογή Μηχανής Συστάσεων

Υπάρχουν μερικά σημαντικά κριτήρια στην επιλογή και χρήση μιας μηχανής συστάσεων [07] :

1. Επιλογή αλγορίθμου: Η επιλογή του αλγορίθμου εξαρτάται από το τι θέλει ο χρήστης να διακρίνει και το τι τύπος συσχέτισης υπάρχει στα δεδομένα. Μερικές από τις κοινές προσεγγίσεις που χρησιμοποιούνται για τις συστάσεις είναι:

1.1. Συστήματα βασισμένα στο περιεχόμενο: Τα συστήματα βασισμένα στο περιεχόμενο απαιτούν την εξαγωγή των χαρακτηριστικών των αντικειμένων. Σε αυτά τα συστήματα η διαδικασία δημιουργίας του προφίλ του χρήστη αποτελείται από δύο στάδια. Αρχικά, το σύστημα πρέπει να προσδιορίσει το βαθμό ενδιαφέροντος του χρήστη απέναντι σε ένα σύνολο δεδομένων. Αυτή η διαδικασία μπορεί να πραγματοποιηθεί παρακολουθώντας τη συμπεριφορά του χρήστη, καθώς επίσης και με τη χρήση διαφόρων ευριστικών μεθόδων για τη ταξινόμηση των αντικειμένων σε δύο κατηγορίες. Η μία αφορά τα αντικείμενα που ενδιαφέρουν το χρήστη και η άλλη σε αυτά που του είναι αδιάφορα. Επιπλέον, αυτό μπορεί να γίνει με τη βαθμολόγηση των αντικειμένων από τους ίδιους τους χρήστες. Έτσι, το προφίλ του χρήστη που δημιουργείται μπορεί να χρησιμοποιηθεί για να παράγει αντικείμενα που θα του προταθούν. Ένα σημαντικό μειονέκτημα των συστημάτων που βασίζονται αποκλειστικά στο προφίλ του χρήστη είναι η έλλειψη της τυχαίας ανακάλυψης συστάσεων μιας και οι συστάσεις βασίζονται αποκλειστικά στα προηγούμενα ενδιαφέροντα του χρήστη. Επιπλέον, τα συστήματα βασίζονται στην ύπαρξη διαθεσιμότητας περιγραφών των αντικειμένων που έχουν προταθεί.

1.2. Συστήματα βασισμένα σε κανόνες: Στην περίπτωση των συστημάτων που βασίζονται σε κανόνες, το προφίλ του κάθε χρήστη μπορεί να αντιπροσωπευθεί ως ένα διάνυσμα από προσωπικά και δημογραφικά χαρακτηριστικά, που καλείται αποτύπωμα. Στο ηλεκτρονικό εμπόριο καθώς και σε εφαρμογές ανάλυσης δεδομένων, τα αποτυπώματα των χρηστών περιλαμβάνουν και κάποια υπολογίσιμα χαρακτηριστικά, όπως ο συνολικός χρόνος που ξόδεψαν, η συχνότητα μιας συναλλαγής ή επίσκεψης, καθώς και η πιο πρόσφατη πραγματοποίηση αυτών. Λίγα είναι τα συστήματα που χρησιμοποιούν δημογραφικά χαρακτηριστικά για την εξαγωγή των συστάσεων. Αυτό συμβαίνει διότι αυτά τα δεδομένα είναι πιο δύσκολο να συλλεχθούν, αλλά και όταν συλλέγονται τείνουν να έχουν χαμηλή ποιότητα. Επίσης, τα συστήματα που βασίζονται σε δημογραφικά χαρακτηριστικά έχει παρατηρηθεί ότι είναι λιγότερο ακριβή συγκριτικά με αυτά που κάνουν χρήση του περιεχομένου των αντικειμένων καθώς και του προφίλ των χρηστών.

1.3. Συνεργατικό φιλτράρισμα (Collaborative filtering): Αυτή η προσέγγιση στηρίζεται στην κοινωνική αλληλεπίδραση μεταξύ των χρηστών. Οι συστάσεις είναι βασισμένες στις αξιολογήσεις που παρέχονται από άλλους χρήστες. Στο συνεργατικό φιλτράρισμα, το σύστημα δεν κάνει χρήση μόνο του προφίλ του τρέχοντος χρήστη, αλλά επίσης διατηρεί μία βάση δεδομένων με τα προφίλ των άλλων χρηστών. Σε αντίθεση με τα συστήματα φιλτραρίσματος περιεχομένου που κάνουν σύγκριση μεταξύ των αντικειμένων, στα συνεργατικά συστήματα η σύγκριση πραγματοποιείται σε επίπεδο χρηστών. Τα προφίλ παρουσιάζονται ως ένα διάνυσμα ή ως ένα σύνολο εκτίμησης παρέχοντας με αυτό τον τρόπο τις προτιμήσεις των χρηστών απέναντι σε ένα σύνολο αντικειμένων. Το προφίλ του ενεργού χρήστη χρησιμοποιείται για να βρεθούν άλλοι χρήστες με παρόμοιες προτιμήσεις, δημιουργώντας με αυτόν τον τρόπο μια γειτονιά με παρόμοιους χρήστες.

Υπάρχουν δύο προσεγγίσεις του συνεργατικού φιλτραρίσματος:

#### 1.3.1. Σύσταση βασισμένη στο χρήστη

Αυτός ο τύπος σύστασης χτίζει τις ομοιότητες μεταξύ των χρηστών ελέγχοντας τις ομοιότητες των στοιχείων που βαθμολογούνται από κάθε χρήστη. Παραδείγματος χάριν, εάν τα στοιχεία είναι σειρές μαθημάτων, δύο χρήστες θα μπορούσαν να θεωρηθούν όμοιοι εάν και οι δύο πήραν τις ίδιες σειρές μαθημάτων. Αντιθέτως, η ομοιότητά τους θα ήταν χαμηλή εάν δεν επέλεξαν οποιαδήποτε παρόμοια σειρά μαθημάτων. Για να κάνουν συστάσεις, οι αλγόριθμοι στηρίζονται στις εκτιμήσεις που οι παρόμοιοι χρήστες έδωσαν σε εκείνες τις σειρές μαθημάτων που δεν λήφθηκαν από το χρήστη. Ο κύριος περιορισμός του αλγορίθμου είναι ότι προκειμένου να παραχθούν οι ομοιότητες, πρέπει να συγκρίνει κάθε χρήστη με κάθε άλλο χρήστη. Αυτό είναι αποδεκτό για μια εφαρμογή με έναν χαμηλό αριθμό χρηστών, αλλά εάν ο αριθμός χρηστών είναι μεγάλος, ο χρόνος για να εκτελεσθεί αυτή η αξιολόγηση αυξάνει εκθετικά.

#### 1.3.2. Σύσταση βασισμένη στα αντικείμενα

Αυτός ο τύπος σύστασης, αφ' ετέρου, αρχίζει με την εξέταση των στοιχείων που συνδέονται με το χρήστη. Για κάθε στοιχείο που συνδέεται με το χρήστη, ο αλγόριθμος υπολογίζει πόσο όμοιο είναι με τα άλλα στοιχεία στη συλλογή για να χτίσει τον κατάλογο συστάσεων. Προκειμένου ο αλγόριθμος να ορίσει πόσο πιθανό είναι στο χρήστη να αρέσει ένα συνιστώμενο στοιχείο, εξετάζει τις εκτιμήσεις που ο χρήστης έχει δώσει στο στοιχείο και δίνει μια σταθμισμένη εκτίμηση σε κάθε συνιστώμενο στοιχείο. Το κύριο ζήτημα του αλγορίθμου είναι ότι πρέπει να χτίσει έναν δείκτη ομοιότητας για κάθε διαθέσιμο στοιχείο. Οι αλλαγές στα στοιχεία, εντούτοις, είναι λιγότερο συχνές από τις αλλαγές στους χρήστες και, επομένως, είναι εφικτό με αυτόν τον τύπο σύστασης να προ-υπολογιστούν οι ομοιότητες offline και να ενημερωθούν στις συγκεκριμένες περιόδους.

1.4 Συσταδοποίηση (Clustering): Με αυτήν την προσέγγιση, η μηχανή συστάσεων προσπαθεί να χτίσει τις συστάσεις βασισμένες στις ομοιότητες μεταξύ είτε των χρηστών είτε των στοιχείων.

1.5 Κατηγοριοποίηση (Categorization): Αυτή η προσέγγιση κατηγοριοποιεί αυτόματα τα στοιχεία χρησιμοποιώντας τις κοινές τους ιδιότητες.

## 2 Πώς θα εκτελεστεί η διαδικασία στους χρήστες σας

Η πραγματική διαδικασία του υπολογισμού της ομοιότητας μεταξύ των χρηστών και των στοιχείων είναι μια εντατική λειτουργία. Ανάλογα με το μέγεθος του συνόλου στοιχείων, η λειτουργία θα μπορούσε να διαρκέσει από μερικά δευτερόλεπτα έως αρκετά. Στις διαδικτυακές εφαρμογές, ο χρόνος απόκρισης είναι σημαντικός εάν οι χρήστες πρέπει να περιμένουν μια εκτεταμένη χρονική περίοδο για να λάβουν μια σύσταση.

Η καλύτερη πρακτική είναι να εκτελεστούν όλοι οι απαιτούμενοι υπολογισμοί offline εάν τα δεδομένα είναι πολλά και σε περιπτώσεις όπου τα νέα στοιχεία προστίθενται σπάνια ή οι χρήστες βαθμολογούν στοιχεία μόνο περιστασιακά.

### 3 Αποθήκευση δεδομένων

Ανάλογα με τον όγκο ή τον τύπο των δεδομένων θα πρέπει να αποφασιστεί εάν τα δεδομένα θα αποθηκευτούν σε ένα κατανεμημένο σύστημα αρχείων ή σε ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων.

# Κεφάλαιο 4

## Τεχνική Περιγραφή της Υλοποίησης

Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής σχεδιάστηκε και υλοποιήθηκε μια εφαρμογή παρουσίασης πληροφοριών σχετικά με του τουριστικές υποδομές στην Ελλάδα. Ο κύριος στόχος της μεταπτυχιακής διατριβής ήταν να μελετηθούν τα πλεονεκτήματα και οι περιορισμοί που προκύπτουν από την αξιοποίηση ενός δημοφιλούς PHP Framework, του Zend Framework, ως προγραμματιστικής μεθόδου ανάπτυξης δυναμικών διαδικτυακών εφαρμογών που υποστηρίζουν την ύπαρξη διαφορετικών ρόλων χρηστών και την εξατομίκευση.

Ολόκληρο το σύστημα υλοποιήθηκε με τη χρήση των τεχνολογιών Zend Framework, HTML, CSS και JavaScript. Επίσης, ένα μέρος του συστήματος που αφορά τη σύνδεση του χρήστη στην εφαρμογή (Διαδικασία Login) υλοποιήθηκε αποκλειστικά με τη χρήση της PHP, ώστε να γίνει δυνατή η σύγκριση των μεθόδων από τεχνικής άποψης. Τα δεδομένα της εφαρμογής αποθηκεύονται σε βάση MySQL και ως HTTP εξυπηρετητής χρησιμοποιήθηκε ο Apache.

## 4.1 Zend Framework

Το Zend Framework έχει σχεδιαστεί και κατασκευαστεί για να βελτιώσει την παραγωγικότητά του προγραμματιστή εφαρμογών. Το πλαίσιο έχει γραφεί εξολοκλήρου σε PHP 5 και η τρέχουσα έκδοση έχει ελεγχθεί διεξοδικά [17].

Το Zend Framework έχει μια τεράστια και ενεργή κοινότητα χρηστών και εστιάζει στην ανάπτυξη web 2.0 εφαρμογών. Είναι ένα από τα πιο δημοφιλή Frameworks της PHP, διαθέτει ισχυρά χαρακτηριστικά που είναι σχεδιασμένα για ανάπτυξη εμπορικών εφαρμογών και απαιτεί εκτεταμένη γνώση της PHP.

Τα πλεονεκτήματα του Zend Framework συνοψίζονται στα παρακάτω [06, 12]:

1. **Επέκταση κλάσεων :** Το Zend Framework (ZF) είναι ένα πλήρως αντικειμενοστραφές πλαίσιο, που χρησιμοποιεί πολλές αντικειμενοστραφείς έννοιες, όπως την κληρονομικότητα και τις διεπαφές. Αυτό επιτρέπει την επέκταση ή την προσαρμογή των τμημάτων του ZF, ώστε να επιτευχθεί η επιθυμητή λειτουργικότητα της εφαρμογής. Λόγω της αντικειμενοστραφούς φύσης του πλαισίου, η επιτευχθείσα λειτουργικότητα μπορεί να επαναχρησιμοποιηθεί και σε άλλες μελλοντικές εφαρμογές.
2. **Αντικειμενοστρέφεια :** Στο Zend Framework όλα είναι αντικείμενα, γεγονός που περιπλέκει την ανάπτυξη κώδικα. Το κύριο πλεονέκτημά του, όμως, είναι η δυνατότητα να κατασταθεί ο κώδικας επαναχρησιμοποιήσιμος.
3. **Χρήση μόνο των απαιτούμενων παρεχόμενων δυνατοτήτων για την ανάπτυξη της εφαρμογής :** Το Zend Framework είναι απλά μια συλλογή κλάσεων και ο προγραμματιστής μπορεί να αξιοποιήσει μόνο τα συστατικά που χρειάζεται.
4. **Προσφέρει πολλές δυνατότητες :** Το Zend Framework πολλές λειτουργίες. Παραδείγματος χάρη, Zend\_Auth για ταυτοποίηση χρηστών, Zend\_Form για δημιουργία φορμών, Zend\_Feed για ανάγνωση RSS feed, κ.ά.
5. **Δεν υπάρχει υλοποίηση του Μοντέλου :** Αυτό σημαίνει ότι ο προγραμματιστής είναι ελεύθερος να χρησιμοποιήσει ό,τι μέσα έχει στη διάθεσή του ώστε να το υλοποιήσει ή απλά να ενσωματώσει τις υπάρχουσες υλοποιήσεις

6. Ενσωμάτωση : Μπορούν πολύ εύκολα να ενσωματωθούν άλλες βιβλιοθήκες στο Zend Framework, καθώς και το αντίστροφο.
7. Προτυποποίηση : Ο κώδικας που υποστηρίζει τη λειτουργία του Zend Framework ελέγχεται και ακολουθεί βασικές αρχές, ώστε να είναι ευανάγνωστος και υψηλής ποιότητας.
8. Κοινότητα και εγχειρίδια : Τα εγχειρίδια χρήσης (πλήρης οδηγός του ZF, παραδείγματα, κώδικας, θεωρίες χρήσης) και η συνεχώς αναπτυσσόμενη κοινότητα συμβάλλουν στην εκμάθηση του Zend Framework.

## 4.2 PHP

Η PHP είναι μια server-side scripting γλώσσα ειδικά σχεδιασμένη για το Διαδίκτυο. Μέσα σε μία HTML σελίδα μπορεί να ενσωματωθεί PHP κώδικας που θα εκτελείται κάθε φορά που κάποιος χρήστης θα επισκέπτεται τη συγκεκριμένη σελίδα. Ο PHP κώδικας ερμηνεύεται από έναν Web server, ο οποίος παράγει HTML ή άλλη έξοδο που παρουσιάζεται στο χρήστη ως ιστοσελίδα [14].

Η PHP είναι λογισμικό ανοιχτού κώδικα. Ο χρήστης μπορεί να προσπελάσει τον πηγαίο κώδικα. Μπορεί, δηλαδή, να τον χρησιμοποιήσει, να τον αλλάξει και να τον επαναμεταδώσει χωρίς καταβολή πληρωμής.

Κάποιοι από τους κύριους ανταγωνιστές της PHP είναι οι εξής: Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) και Allaire Cold Fusion.

Σε σύγκριση με αυτά τα λογισμικά, ο λόγος για τον οποίο επιλέχθηκε η PHP είναι ότι έχει πολλά θετικά [02, 03] :

1. Υψηλή απόδοση : χρησιμοποιώντας έναν φτηνό επεξεργαστή, μπορούν να εκτελεστούν εκατομμύρια hits ανά ημέρα
2. Διεπαφές σε πολλά διαφορετικά συστήματα βάσεων δεδομένων: απευθείας σύνδεση σε MySQL, PostgreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix, Interbase, Sybase βάσεις δεδομένων, κ.ά.



3. Βιβλιοθήκες : διαθέτει πολλές συναρτήσεις για εκτέλεση πολλών χρήσιμων εργασιών σε σχέση με το Διαδίκτυο. Ο χρήστης μπορεί να δημιουργήσει εικόνες GIF on-the-fly ή PDF αρχεία, να στείλει email, να χρησιμοποιήσει cookies και να συνδεθεί σε άλλες υπηρεσίες δικτύου μόνο με λίγες γραμμές κώδικα
4. Χαμηλό κόστος : η PHP διατίθεται δωρεάν στην ηλεκτρονική διεύθυνση <http://www.php.net>
5. Ευκολία εκμάθησης και χρήσης : το συντακτικό της PHP βασίζεται σε άλλες γλώσσες προγραμματισμού, κυρίως στις C και Perl. Αν κάποιος γνωρίζει ήδη C ή Perl, ή μία C-like γλώσσα όπως C++ ή Java, τότε θα είναι πολύ παραγωγικός στην χρήση της PHP σχεδόν αμέσως
6. Διαθέσιμη σε πολλά διαφορετικά λειτουργικά συστήματα : ο χρήστης μπορεί να γράψει PHP κώδικα σε ελεύθερα Unix-like λειτουργικά συστήματα όπως Linux και FreeBSD, σε επαγγελματικές εκδόσεις Unix όπως Solaris και IRIX ή διάφορες εκδόσεις των Microsoft Windows
7. Διαθεσιμότητα πηγαίου κώδικα : ο χρήστης έχει πρόσβαση στον πηγαίο κώδικα της PHP. Ανόμοια με εμπορικά, κλειστού κώδικα λογισμικά, αν ο χρήστης επιθυμεί να αλλάξει ή να προσθέσει κάτι στη γλώσσα, είναι ελεύθερος να το κάνει

## 4.3 MySQL

Η βάση δεδομένων με την οποία η PHP συνεργάζεται απόλυτα σε Διαδικτυακές εφαρμογές είναι η MySQL. Η MySQL είναι ένα πολύ γρήγορο, αξιόπιστο, σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (relational database management system-RDBMS) [02]. Μία βάση δεδομένων δίνει τη δυνατότητα στο χρήστη να αποθηκεύει, να κάνει αναζήτηση, να ταξινομεί και να ανακτά δεδομένα. Ο MySQL επεξεργαστής ελέγχει την πρόσβαση στα δεδομένα του χρήστη για να εγγυηθεί ότι πολλαπλοί χρήστες μπορούν να εργαστούν ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση στα δεδομένα, και για να εγγυηθεί ότι μόνο οι εξουσιοδοτημένοι χρήστες έχουν δικαίωμα πρόσβασης. Γι' αυτό το λόγο η MySQL είναι ένας επεξεργαστής πολλαπλών χρηστών και νημάτων (multi-user, multi-threaded server).

Χρησιμοποιεί SQL (Structured Query Language), την πρότυπη γλώσσα ερωτημάτων βάσεων δεδομένων παγκοσμίως και είναι λογισμικό ανοιχτού κώδικα που διανέμεται δωρεάν.

Η πλειοψηφία των χρηστών υποστηρίζει ότι η MySQL είναι η καλύτερη επιλογή για το μεγαλύτερο μέρος των εφαρμογών που αναπτύσσουν. Είναι ιδιαίτερα ικανοποιητική για εφαρμογές στο Διαδίκτυο για τους εξής σημαντικούς λόγους [02] :

1. Χαμηλό κόστος : αναμφίβολα οι Oracle, Sybase και Informix κατασκευάζουν υπέροχες βάσεις δεδομένων, αλλά το κόστος είναι απαγορευτικό για πολλούς χρήστες. Οποιοσδήποτε μπορεί να εγκαταστήσει και να χρησιμοποιεί τη MySQL δωρεάν
2. Είναι γρήγορη και ισχυρή : η MySQL διαθέτει όλη την απαιτούμενη δύναμη για διανομή περιεχομένου στο Διαδίκτυο και για κατασκευή εμπορικών ιστοσελίδων μετρίου μεγέθους. Επιπλέον, για μικρού έως μεσαίου μεγέθους βάσεις δεδομένων, η MySQL είναι υπερβολικά γρήγορη
3. Διαρκής βελτίωση : η ομάδα ανάπτυξης ανακοινώνει updates συχνά και προσθέτει εντυπωσιακά χαρακτηριστικά διαρκώς

## 4.4 Apache Web Server

Ο Apache HTTP Server είναι ο δημοφιλέστερος web server στο Διαδίκτυο από τον Απρίλιο του 1996, συγκεντρώνοντας σήμερα πάνω από το 60% του συνόλου. Ο Apache HTTP Server αποτελεί μία προσπάθεια ανάπτυξης και συντήρησης ενός ανοικτού-κώδικα HTTP server για σύγχρονα λειτουργικά συστήματα όπως, Unix και Windows NT. Ο σκοπός της προσπάθειας είναι να αναπτυχθεί ένας ασφαλής, αποδοτικός και επεκτάσιμος server ο οποίος να μπορεί να παρέχει σύγχρονες υπηρεσίες με βάση τα υπάρχοντα HTTP standards [02, 03].

# Κεφάλαιο 5

## Σχεδίαση Συστήματος Εφαρμογής

Η διαδικασία σχεδίασης είναι πολύ σημαντική γιατί μας επιτρέπει να φτάσουμε στη λύση που ταιριάζει καλύτερα στις απαιτήσεις και τους περιορισμούς του περιβάλλοντος. Επίσης, το στάδιο της σχεδίασης θα οδηγήσει στην ικανοποίηση των απαιτήσεων των χρηστών και θα προσφέρει πλήρεις και σαφείς οδηγίες προς τους προγραμματιστές.

Το σύστημα που σχεδιάστηκε και αναπτύχθηκε για τους σκοπούς της παρούσας μεταπτυχιακής διατριβής είναι ένας δυναμικός διαδικτυακός τύπος παρουσίασης πληροφοριών σχετικά με τουριστικές υποδομές. Για να επιτευχθεί η σωστή σχεδίαση και ανάπτυξη του συστήματος πρέπει να συλλεχθούν και να αναλυθούν οι απαιτήσεις της εφαρμογής.

Το σύστημα θα πρέπει να υποστηρίζει τις εξής διαφορετικές κατηγορίες χρηστών:

1. Εξουσιοδοτημένοι χρήστες: χρήστες που έχουν κάνει εγγραφή στο σύστημα και με την είσοδο τους σε αυτό τους παρέχονται επιπρόσθετα δικαιώματα. Σημαντικό πλεονέκτημα των εξουσιοδοτημένων χρηστών είναι ότι μπορούν να παραμετροποιήσουν την εμφάνιση της εφαρμογής επιλέγοντας τα χρώματα και δομή.
  - 1.1. Διαχειριστής της εφαρμογής
  - 1.2. Ξενοδόχοι των τουριστικών υποδομών
  - 1.3. Ιδιώτες (πελάτες και υποψήφιοι πελάτες) των τουριστικών υποδομών
2. Μη εξουσιοδοτημένοι χρήστες: χρήστες που δεν έχουν κάνει εγγραφή στο σύστημα και διαθέτουν περιορισμένα δικαιώματα.
  - 2.1. Κάθε τύπου επισκέπτες του διαδικτυακού τόπου

## 5.1 Παρουσίαση Χρηστών

Στη συνέχεια παρουσιάζονται οι διαφορετικοί ρόλοι χρηστών και τα δικαιώματα αυτών, όπως συναντώνται στο σύστημα που αναπτύχθηκε:

### 1. Διαχειριστής

Ο διαχειριστής του συστήματος διαχειρίζεται όλους τους υπόλοιπους χρήστες και τις ενέργειες αυτών. Αυτό βέβαια απαιτεί την είσοδο-σύνδεση του διαχειριστή στο σύστημα με τη χρήση του κατάλληλου κωδικού. Στη διαχείριση των χρηστών συμπεριλαμβάνεται η διαγραφή αυτών από την κατάλληλα σχεδιασμένη και αναπτυγμένη Βάση Δεδομένων, που θα επικοινωνεί με το σύστημα διαγράφοντας τα αντίστοιχα απαιτούμενα δεδομένα. Στη διαχείριση των ενεργειών των χρηστών συμπεριλαμβάνεται η διαγραφή και ενημέρωση των ξενοδοχείων

## 2. Ξενοδόχοι

Οι ξενοδόχοι ανήκουν στην κατηγορία των εξουσιοδοτημένων χρηστών που μέσω της εγγραφής τους στο σύστημα αποσκοπούν στη διαφήμιση των ξενοδοχειακών τους μονάδων και στην καλύτερη επικοινωνία με τους πελάτες τους. Αφού συνδεθούν στο σύστημα, μπορούν να εισάγουν πληροφορίες σχετικά με τα ξενοδοχεία τους (όνομα ξενοδοχείου, αστέρια, αριθμός κλινών, τύπος, τιμές, παροχές, περιγραφή, φωτογραφίες, στοιχεία επικοινωνίας, κ.ά.) και μετέπειτα να τις αλλάξουν ή ακόμα και να τις διαγράψουν.

## 3. Ιδιώτες

Οι ιδιώτες είναι πελάτες και υποψήφιοι πελάτες που ανήκουν στην κατηγορία των εξουσιοδοτημένων χρηστών και μέσω της εγγραφής τους στο σύστημα επιθυμούν την αποδοτικότερη πληροφόρηση και αναζήτηση ξενοδοχείων. Επίσης, στους πελάτες παρέχεται η δυνατότητα αξιολόγησης και βαθμολόγησης των παρεχόμενων υπηρεσιών των ξενοδοχείων που έχουν επισκεφτεί. Η βαθμολόγηση θα αξιοποιηθεί ως πληροφορία από το σύστημα την αναζήτηση. Πιο συγκεκριμένα, το σύστημα δεν θα επιστρέφει ως αποτέλεσμα αναζήτησης σε ένα συγκεκριμένο χρήστη ένα ξενοδοχείο που έχει αξιολογήσει ο ίδιος με χαμηλό βαθμό (βαθμό μικρότερο του 3 στα 5 αστέρια). Επίσης, σ' αυτήν την κατηγορία χρηστών το σύστημα θα εμφανίζει το αγαπημένο τους ξενοδοχείο, δηλαδή το ξενοδοχείο στο οποίο οι ίδιοι έχουν δώσει το μεγαλύτερο βαθμό.

## 4. Επισκέπτες

Οι επισκέπτες ανήκουν στην κατηγορία των μη εξουσιοδοτημένων χρηστών που επιθυμούν την απλή πληροφόρηση.

## **5.2 Σχεδίαση Συστήματος με Βάση τη Μεθοδολογία ICONIX**

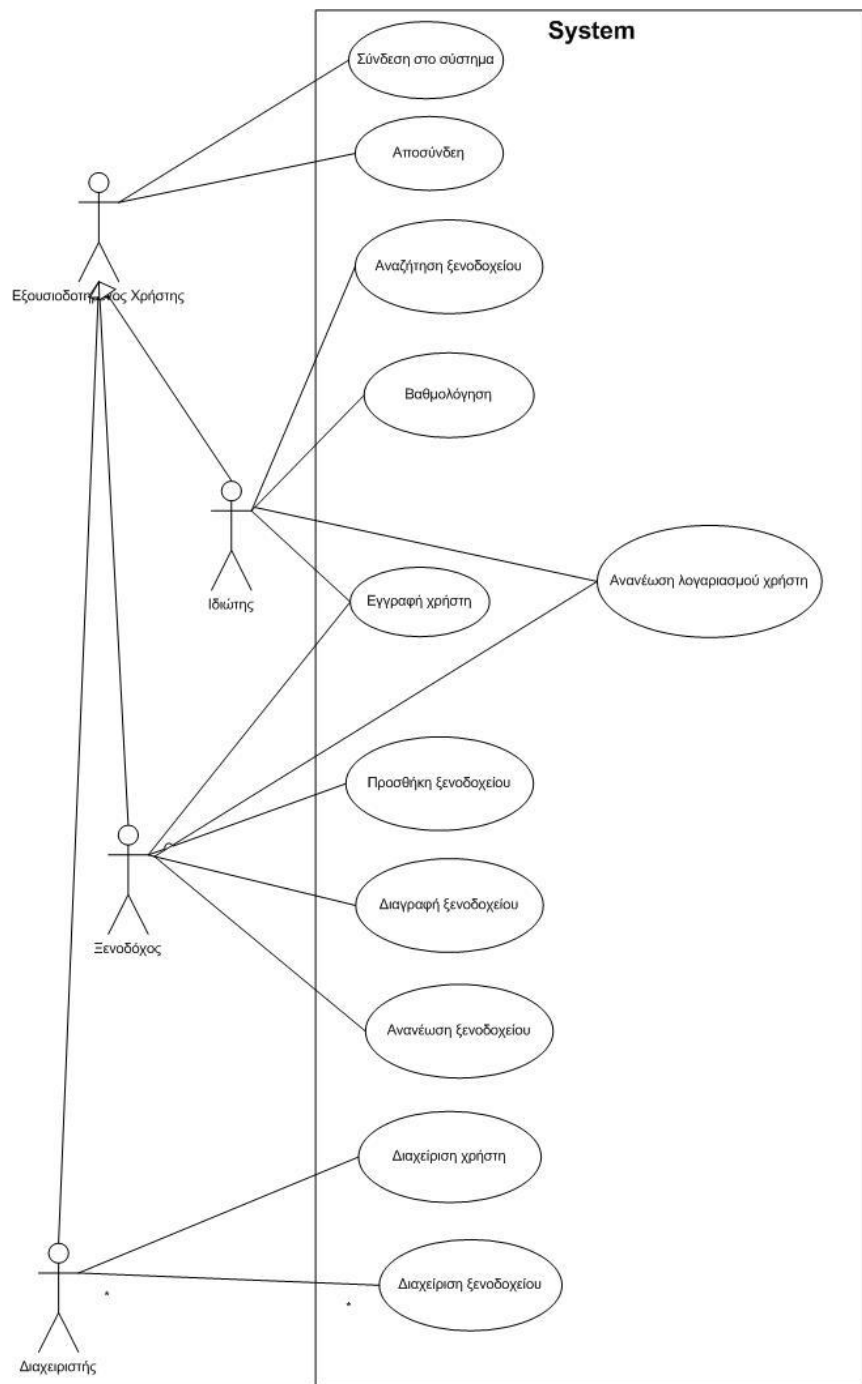
Η ICONIX είναι μια μεθοδολογία ανάπτυξης λογισμικού που επιτρέπει τη συστηματική μετάβαση από τις αρχικές απαιτήσεις όπως αυτές διατυπώνονται από τον πελάτη, στον

κώδικα που τελικά υλοποιεί τις απαιτήσεις αυτές. Το μείζον χαρακτηριστικό της μεθοδολογίας ICONIX είναι η επαναληπτικότητα. Αφενός, η διαδικασία είναι επαναληπτική διότι επιτρέπει την παραγωγή λειτουργικού κώδικα για κάθε μια περίπτωση χρήσης του συστήματος ξεχωριστά. Σε κάθε επανάληψη, εξετάζεται μια νέα περίπτωση χρήσης που καταλήγει στην προσθήκη λειτουργικότητας στο τελικό προϊόν. Αφετέρου, η διαδικασία είναι επαναληπτική διότι επιτρέπει (και υποβοηθά) την επιστροφή από ένα στάδιο της διαδικασίας ανάπτυξης (π.χ. το σχεδιασμό) σε προηγούμενα (π.χ. την ανάλυση απαιτήσεων) με σκοπό την αποσαφήνιση, βελτίωση και διόρθωση προηγούμενων ενεργειών. Η επαναληπτικότητα βρίσκεται στο κέντρο του αντικειμενοστρεφούς υποδείγματος προγραμματισμού καθώς αναγνωρίζει ότι ένα μεγάλο σύστημα λογισμικού δεν μπορεί να αναπτυχθεί με μιας και ότι οι αλλαγές σε προηγούμενες επιλογές είναι αναπόφευκτες [09].

### **5.2.1 Διάγραμμα Περιπτώσεων Χρήσης**

Το παρακάτω διάγραμμα δείχνει τους χρήστες του συστήματος, τις διάφορες εφαρμογές και τις μεταξύ τους σχέσεις.

Πιο αναλυτικά στο Σχήμα 5.1 παρατηρούμε ότι στο σύστημα υπάρχουν οι Εξουσιοδοτημένοι Χρήστες, οι οποίοι μπορούν να συνδεθούν και να αποσυνδεθούν από την εφαρμογή. Οι Ιδιώτες και οι Ξενοδόχοι, για να αποκτήσουν τα δικαιώματα των εξουσιοδοτημένων χρηστών πρέπει πρώτα να κάνουν εγγραφή στο σύστημα, ενώ ο λογαριασμός του Διαχειριστή προϋπάρχει και δεν απαιτεί εγγραφή. Τα στοιχεία λογαριασμού που έχουν δηλώσει οι χρήστες κατά την εγγραφή τους μπορούν στη συνέχεια να τα ενημερώνουν όποτε το επιθυμούν. Οι Ιδιώτες μπορούν να αναζητούν και να βαθμολογούν ξενοδοχεία. Οι Ξενοδόχοι μπορούν να δημιουργούν, να διαγράφουν ή να ανανεώνουν τα ξενοδοχεία τους. Τέλος, ο Διαχειριστής μπορεί να διαχειρίζεται χρήστες και ξενοδοχεία.



**Σχήμα 5.1:** Διάγραμμα Περιπτώσεων Χρήσης

## 5.2.2 Λεκτική Περιγραφή Περιπτώσεων Χρήσης

Ακολουθεί η λεκτική περιγραφή των περιπτώσεων χρήσης του συστήματος. Όπου αναφέρεται η φράση “ο χρήστης επιλέγει”, θεωρούμε ότι η επιλογή γίνεται μέσω δεικτικής συσκευής. Όπου αναφέρεται η φράση “ο χρήστης εισάγει”, θεωρούμε ότι ο χρήστης πληκτρολογεί σε κατάλληλο πεδίο της διεπαφής.

### Περίπτωση Χρήσης 1: ΣΥΝΔΕΣΗ ΣΤΟ ΣΥΣΤΗΜΑ

#### Βασική Ροή Γεγονότων:

1. Ο χρήστης επιλέγει το Πλήκτρο “Login” στην “Αρχική Οθόνη” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Login”.
3. Ο χρήστης εισάγει το όνομα χρήστη (username) και τον κωδικό (password) και επιλέγει το πλήκτρο Login.
4. Το σύστημα ελέγχει και διαπιστώνει ότι ο χρήστης ανήκει στον κατάλογο εγγεγραμμένων χρηστών.
5. Το σύστημα δίνει στο χρήστη τα κατάλληλα δικαιώματα χρήσης, εμφανίζει μήνυμα επιτυχούς σύνδεσης και εμφανίζει την αντίστοιχη “Οθόνη Επιλογών Χρήστη”.

#### Εναλλακτικές Ροές:

##### Εναλλακτική Ροή 1 – Μη εξουσιοδοτημένος χρήστης

4α) Το σύστημα ελέγχει και διαπιστώνει ότι ο χρήστης δεν ανήκει στον κατάλογο εγγεγραμμένων χρηστών και εμφανίζει μήνυμα ανεπιτυχούς σύνδεσης.

4β) Η ροή μεταφέρεται στο βήμα 2 της Βασικής Ροής.



## **Περίπτωση Χρήσης 2: ΑΠΟΣΥΝΔΕΣΗ**

### **Βασική Ροή Γεγονότων:**

1. Ο χρήστης επιλέγει το πλήκτρο “Logout” στην “ Οθόνη Επιλογών Χρήστη” του συστήματος.
2. Το σύστημα δίνει στο χρήστη τα κατάλληλα δικαιώματα χρήσης και εμφανίζει μήνυμα επιτυχούς αποσύνδεσης.
3. Το σύστημα εμφανίζει την “Αρχική Οθόνη” του συστήματος.

## **Περίπτωση Χρήσης 3: ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ**

### **Βασική Ροή Γεγονότων:**

1. Ο χρήστης επιλέγει το πλήκτρο “Εγγραφή Χρήστη” στην “Αρχική Οθόνη” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Εγγραφή Χρήστη”.
3. Ο χρήστης εισάγει τα στοιχεία του στα αντίστοιχα πεδία και επιλέγει το πλήκτρο “Καταχώριση”.
4. Το σύστημα ελέγχει και διαπιστώνει ότι ο χρήστης έχει καταχωρίσει σωστά όλα τα απαιτούμενα στοιχεία, δημιουργεί τον αντίστοιχο λογαριασμό χρήστη και εμφανίζει μήνυμα επιτυχούς εγγραφής.
5. Το σύστημα εμφανίζει την “Αρχική Οθόνη” του συστήματος.

### **Εναλλακτικές Ροές:**

#### **Εναλλακτική Ροή 1 – Λανθασμένη καταχώριση στοιχείων εγγραφής**

4α) Το σύστημα ελέγχει και διαπιστώνει ότι ο χρήστης δεν έχει καταχωρίσει σωστά όλα τα απαιτούμενα στοιχεία και εμφανίζει μήνυμα ανεπιτυχούς εγγραφής.

4β) Η ροή μεταφέρεται στο βήμα 2 της Βασικής Ροής.

### **Εναλλακτική Ροή 2 – Ακύρωση εγγραφής**

3α) Ο χρήστης επιλέγει το πλήκτρο “Ακύρωση εγγραφής”.

3β) Το σύστημα εμφανίζει την “Αρχική Οθόνη” του συστήματος.

## **Περίπτωση Χρήσης 4: ΔΙΑΓΡΑΦΗ ΧΡΗΣΤΗ**

### **Βασική Ροή Γεγονότων:**

1. Ο χρήστης επιλέγει το πλήκτρο “Διαγραφή Χρήστη” στην “Οθόνη Επιλογών Χρήστη” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Διαγραφή Χρήστη”.
3. Ο χρήστης επιλέγει το πλήκτρο “Διαγραφή”.
4. Το σύστημα ενημερώνει τον κατάλογο εγγεγραμμένων χρηστών και εμφανίζει μήνυμα επιτυχούς διαγραφής.
5. Το σύστημα εμφανίζει την “Αρχική Οθόνη” του συστήματος.

### **Εναλλακτικές Ροές:**

#### **Εναλλακτική Ροή 1 – Ακύρωση διαγραφής**

3α) Ο χρήστης επιλέγει το πλήκτρο “Ακύρωση διαγραφής”.

3β) Το σύστημα εμφανίζει την “Οθόνη Επιλογών Χρήστη” του συστήματος

## **Περίπτωση Χρήσης 5: ΑΛΛΑΓΗ ΣΤΟΙΧΕΙΩΝ ΛΟΓΑΡΙΑΣΜΟΥ ΧΡΗΣΤΗ**

### **Βασική Ροή Γεγονότων:**

1. Ο χρήστης επιλέγει το πλήκτρο “Αλλαγή Στοιχείων” στην “Οθόνη Επιλογών Χρήστη” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Αλλαγή Στοιχείων”.
3. Ο χρήστης εισάγει τα νέα στοιχεία του στα αντίστοιχα πεδία και επιλέγει το πλήκτρο “Αλλαγή”.
4. Το σύστημα ελέγχει και διαπιστώνει ότι ο χρήστης έχει καταχωρίσει σωστά όλα τα απαιτούμενα στοιχεία, ενημερώνει το λογαριασμό χρήστη και εμφανίζει μήνυμα επιτυχούς αλλαγής στοιχείων στην “Οθόνη Επιλογών Χρήστη”.

### **Εναλλακτικές Ροές:**

#### **Εναλλακτική Ροή 1 – Λανθασμένη καταχώριση στοιχείων**

4α) Το σύστημα ελέγχει και διαπιστώνει ότι ο χρήστης δεν έχει καταχωρίσει σωστά όλα τα απαιτούμενα στοιχεία και εμφανίζει μήνυμα ανεπιτυχούς καταχώρισης στοιχείων.

4β) Η ροή μεταφέρεται στο βήμα 2 της Βασικής Ροής.

#### **Εναλλακτική Ροή 2 – Ακύρωση αλλαγής**

3α) Ο χρήστης επιλέγει το πλήκτρο “Ακύρωση αλλαγής”.

3β) Το σύστημα εμφανίζει την “Οθόνη Επιλογών Χρήστη” του συστήματος

## **Περίπτωση Χρήσης 5: ΠΡΟΣΘΗΚΗ ΞΕΝΟΔΟΧΕΙΟΥ**

### **Βασική Ροή Γεγονότων:**

1. Ο ιδιοκτήτης επιλέγει το πλήκτρο “Προσθήκη Ξενοδοχείου” στην “Οθόνη Επιλογών Ιδιοκτήτη” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Προσθήκη Ξενοδοχείου”.
3. Ο ιδιοκτήτης εισάγει τα στοιχεία του ξενοδοχείου στα αντίστοιχα πεδία και επιλέγει το πλήκτρο “Καταχώριση”.
4. Το σύστημα ελέγχει και διαπιστώνει ότι ο ιδιοκτήτης έχει καταχωρίσει σωστά όλα τα απαιτούμενα στοιχεία, καταχωρεί το ξενοδοχείο και εμφανίζει μήνυμα επιτυχούς καταχώρισης ξενοδοχείου στην “Οθόνη Επιλογών Ιδιοκτήτη”.

### **Εναλλακτικές Ροές:**

#### **Εναλλακτική Ροή 1 – Ανεπιτυχής καταχώριση ξενοδοχείου**

4α) Το σύστημα ελέγχει και διαπιστώνει ότι ο ιδιοκτήτης δεν έχει καταχωρίσει σωστά όλα τα απαιτούμενα στοιχεία και εμφανίζει μήνυμα ανεπιτυχούς καταχώρισης ξενοδοχείου.

4β) Η ροή μεταφέρεται στο βήμα 2 της Βασικής Ροής.

#### **Εναλλακτική Ροή 2 – Ακύρωση προσθήκης ξενοδοχείου**

3α) Ο ιδιοκτήτης επιλέγει το πλήκτρο “Ακύρωση προσθήκης ξενοδοχείου”.

3β) Το σύστημα εμφανίζει την “Οθόνη Επιλογών Ιδιοκτήτη” του συστήματος.

## **Περίπτωση Χρήσης 7: ΑΝΑΝΕΩΣΗ ΞΕΝΟΔΟΧΕΙΟΥ**

### **Βασική Ροή Γεγονότων:**

1. Ο ιδιοκτήτης επιλέγει το πλήκτρο “Ανανέωση Ξενοδοχείου” στην “Οθόνη Επιλογών Ιδιοκτήτη” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Ανανέωση Ξενοδοχείου”.
3. Ο ιδιοκτήτης εισάγει τα νέα στοιχεία του ξενοδοχείου στα αντίστοιχα πεδία και επιλέγει το πλήκτρο “Ανανέωση”.
4. Το σύστημα ελέγχει και διαπιστώνει ότι ο ιδιοκτήτης έχει καταχωρίσει σωστά όλα τα απαιτούμενα στοιχεία, ανανεώνει τα στοιχεία ξενοδοχείου και εμφανίζει μήνυμα επιτυχούς αλλαγής στοιχείων.

### **Εναλλακτικές Ροές:**

#### **Εναλλακτική Ροή 1 – Ανεπιτυχής ανανέωση ξενοδοχείου**

4α) Το σύστημα ελέγχει και διαπιστώνει ότι ο ιδιοκτήτης δεν έχει καταχωρίσει σωστά όλα τα απαιτούμενα στοιχεία και εμφανίζει μήνυμα ανεπιτυχούς ανανέωσης στοιχείων.

4β) Η ροή μεταφέρεται στο βήμα 2 της Βασικής Ροής.

#### **Εναλλακτική Ροή 2 – Ακύρωση ανανέωσης ξενοδοχείου**

3α) Ο ιδιοκτήτης επιλέγει το πλήκτρο “Ακύρωση ανανέωσης ξενοδοχείου”.

3β) Το σύστημα εμφανίζει την “Οθόνη Επιλογών Ιδιοκτήτη” του συστήματος.

## **Περίπτωση Χρήσης 8: ΔΙΑΓΡΑΦΗ ΞΕΝΟΔΟΧΕΙΟΥ**

### **Βασική Ροή Γεγονότων:**

1. Ο ιδιοκτήτης επιλέγει το πλήκτρο “Διαγραφή Ξενοδοχείου” στην “Οθόνη Επιλογών Ιδιοκτήτη” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Διαγραφή Ξενοδοχείου”.
3. Ο ιδιοκτήτης επιλέγει το πλήκτρο “Διαγραφή”.
4. Το σύστημα διαγράφει το ξενοδοχείο από τον κατάλογο ξενοδοχείων και εμφανίζει μήνυμα επιτυχούς διαγραφής.

### **Εναλλακτικές Ροές:**

#### **Εναλλακτική Ροή 1 – Ακύρωση διαγραφής ξενοδοχείου**

- 3α) Ο ιδιοκτήτης επιλέγει το πλήκτρο “Ακύρωση διαγραφής ξενοδοχείου”.
- 3β) Το σύστημα εμφανίζει την “Οθόνη Επιλογών Ιδιοκτήτη” του συστήματος.

## **Περίπτωση Χρήσης 9: ΑΝΑΖΗΤΗΣΗ ΞΕΝΟΔΟΧΕΙΟΥ**

### **Βασική Ροή Γεγονότων:**

1. Ο επισκέπτης επιλέγει το πλήκτρο “Αναζήτηση Ξενοδοχείου” στην “Αρχική Οθόνη” του συστήματος.
2. Το σύστημα εμφανίζει την Οθόνη “Αναζήτηση Ξενοδοχείου”.
3. Ο επισκέπτης εισάγει τιμές στα αντίστοιχα πεδία και επιλέγει το πλήκτρο “Αναζήτηση”.

4. Το σύστημα αναζητά, ανακτά τα ξενοδοχεία από τον κατάλογο ξενοδοχείων και εμφανίζει τα αποτελέσματα της αναζήτησης.

### **Εναλλακτικές Ροές:**

#### **Εναλλακτική Ροή 1 – Ακύρωση αναζήτησης ξενοδοχείου**

3α) Ο επισκέπτης επιλέγει το πλήκτρο “Ακύρωση αναζήτησης ξενοδοχείου”.

3β) Το σύστημα εμφανίζει την “Αρχική Οθόνη” του συστήματος.

#### **Περίπτωση Χρήσης 10: ΑΞΙΟΛΟΓΗΣΗ ΞΕΝΟΔΟΧΕΙΟΥ**

### **Βασική Ροή Γεγονότων:**

1. Ο πελάτης επιλέγει το πλήκτρο “Αξιολόγηση Ξενοδοχείου” στην “Οθόνη Επιλογών Πελάτη” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “ Αξιολόγηση Ξενοδοχείου ”.
3. Ο πελάτης εισάγει αναφορά, επιλέγει βαθμολογία από μενού και την ένδειξη Αγαπημένο και επιλέγει το πλήκτρο “Καταχώριση”.
4. Το σύστημα ελέγχει και διαπιστώνει ότι ο πελάτης έχει καταχωρίσει τιμές στα αντίστοιχα πεδία, δημιουργεί την αξιολόγηση και εμφανίζει μήνυμα επιτυχούς καταχώρισης αξιολόγησης.

### **Εναλλακτικές Ροές:**

#### **Εναλλακτική Ροή 1 – Ανεπιτυχής αξιολόγηση**

4α) Το σύστημα ελέγχει και διαπιστώνει ότι ο πελάτης δεν έχει καταχωρίσει τιμές στα αντίστοιχα πεδία και εμφανίζει μήνυμα ανεπιτυχούς καταχώρισης αξιολόγησης.

4β) Η ροή μεταφέρεται στο βήμα 2 της Βασικής Ροής.

## **Εναλλακτική Ροή 2 – Ακύρωση αξιολόγησης**

3α) Ο πελάτης επιλέγει το πλήκτρο “Ακύρωση αξιολόγησης”.

3β) Το σύστημα εμφανίζει την “Οθόνη Επιλογών Πελάτη” του συστήματος.

## **Περίπτωση Χρήσης 11: ΔΙΑΧΕΙΡΙΣΗ ΧΡΗΣΤΩΝ**

### **Βασική Ροή Γεγονότων:**

1. Ο διαχειριστής επιλέγει το πλήκτρο “Διαχείριση Χρηστών” στην “Οθόνη Επιλογών Διαχειριστή” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Διαχείριση Χρηστών”.
3. Ο διαχειριστής επιλέγει χρήστη.
4. Το σύστημα εμφανίζει τα στοιχεία του λογαριασμού χρήστη και επιλογές διαχείρισης.

## **Περίπτωση Χρήσης 12: ΔΙΑΧΕΙΡΙΣΗ ΞΕΝΟΔΟΧΕΙΩΝ**

### **Βασική Ροή Γεγονότων:**

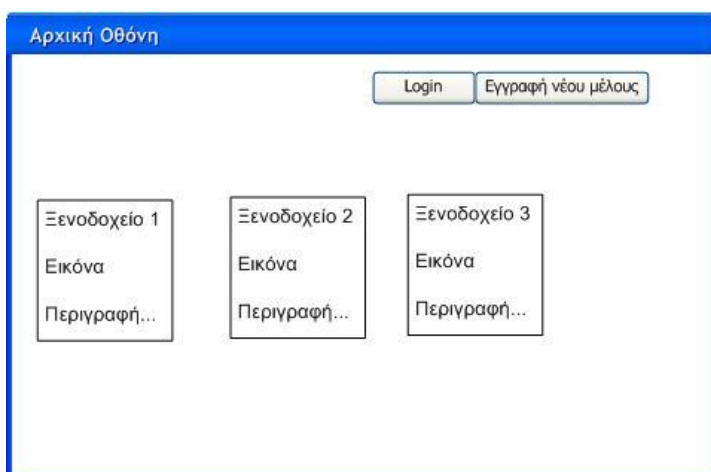
1. Ο διαχειριστής επιλέγει το πλήκτρο “Διαχείριση Ξενοδοχείων” στην “Οθόνη Επιλογών Διαχειριστή” του συστήματος.
2. Το σύστημα εμφανίζει την οθόνη “Διαχείριση Ξενοδοχείων”.
3. Ο διαχειριστής επιλέγει ξενοδοχείο.
4. Το σύστημα εμφανίζει τα στοιχεία του ξενοδοχείου και επιλογές διαχείρισης.



### 5.2.3 Ενδεικτικές Οθόνες του Συστήματος

Οι ενδεικτικές οθόνες που παρουσιάζονται στη φάση αυτή δεν είναι αναλυτικές (δηλαδή δεν περιλαμβάνονται όλα τα πλήκτρα, πεδία, χρώματα, μηνύματα κλπ), καθώς κάτι τέτοιο θα ήταν δεσμευτικό για την υλοποίηση, στοιχείο που δεν είναι επιθυμητό στο στάδιο της ανάλυσης των απαιτήσεων. Επίσης, οι οθόνες δεν καλύπτουν το σύνολο των διεπαφών μεταξύ χρήστη και συστήματος αλλά τις διεπαφές εκείνες που κρίνεται σκόπιμο να διερευνηθούν.

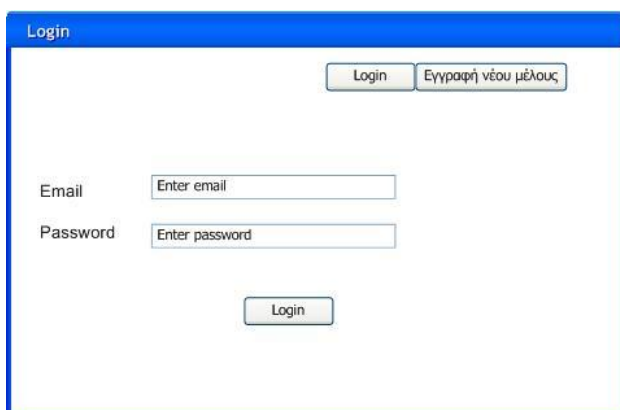
#### Αρχική Οθόνη



**Σχήμα 5.2:** Η Αρχική Οθόνη του Συστήματος

Στο Σχήμα 5.2 παρουσιάζεται η Αρχική Οθόνη της εφαρμογής. Στην Αρχική Οθόνη θα υπάρχουν οι σύνδεσμοι για σύνδεση στο σύστημα και για εγγραφή νέου μέλους. Επίσης στην Αρχική Οθόνη θα παρουσιάζεται ο κατάλογος όλων των καταχωρημένων ξενοδοχείων στην εφαρμογή, από όπου ο χρήστης θα μπορεί να επιλέξει ξενοδοχείο και να δει την πλήρη περιγραφή του. Στις πληροφορίες που αφορούν στα ξενοδοχεία θα έχουν πρόσβαση και οι μη εξουσιοδοτημένοι χρήστες.

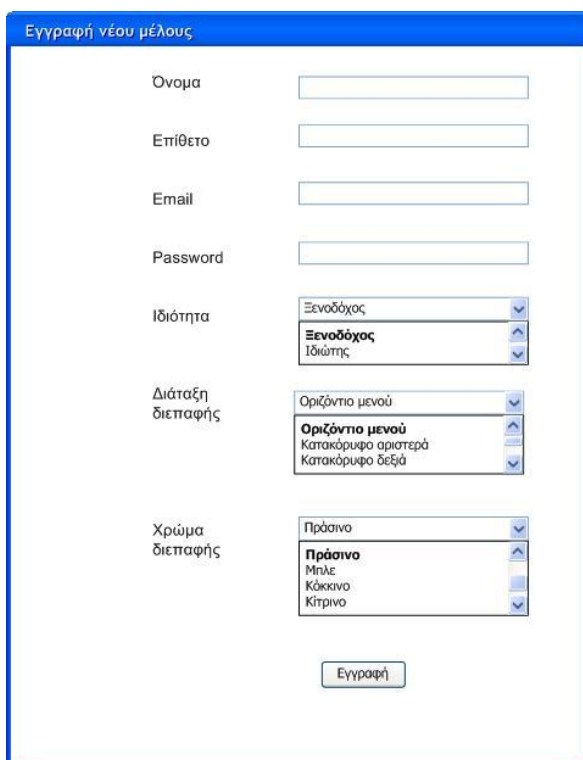
## Οθόνη Login



**Σχήμα 5.3:** Οθόνη Σύνδεσης εγγεγραμμένων χρηστών στο Σύστημα

Στο Σχήμα 5.3 παρουσιάζεται η Οθόνη Σύνδεσης στην εφαρμογή. Τα απαιτούμενα στοιχεία σύνδεσης είναι το email και ο κωδικός, που ο χρήστης έχει δώσει κατά την εγγραφή του στο σύστημα.

## Οθόνη Εγγραφής Χρήστη



**Σχήμα 5.4:** Οθόνη εγγραφής χρήστη στο Σύστημα

Στο Σχήμα 5.4 παρουσιάζεται η Οθόνη Εγγραφής χρήστη στο σύστημα. Τα απαιτούμενα στοιχεία εγγραφής είναι όνομα, επίθετο, email, κωδικός και ιδιότητα χρήστη (Ξενοδόχος ή Ιδιώτης) ώστε το σύστημα να διακρίνει το ρόλο και τα δικαιώματα του συγκεκριμένου χρήστη. Επίσης, κατά την εγγραφή του ο χρήστης μπορεί να επιλέξει διάταξη και χρώμα διεπαφής.

## Οθόνη Προσθήκης Ξενοδοχείου

Νέο Ξενοδοχείο

Τίτλος

Περιγραφή

Αστέρια

Τηλέφωνο

Fax

Τοποθεσία

Logo

Δωμάτια

Ανέσεις

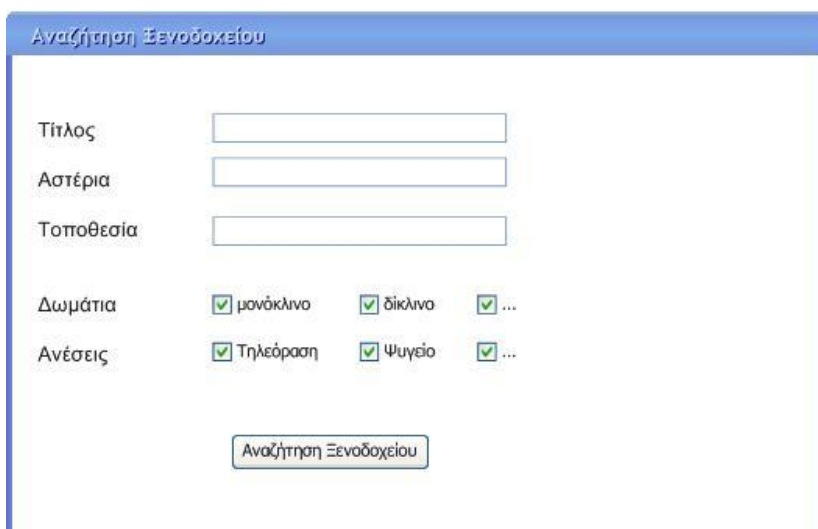
Εικόνες

Δημιουργία Ξενοδοχείου

**Σχήμα 5.5:** Οθόνη προσθήκης νέου ξενοδοχείου στο Σύστημα

Στο Σχήμα 5.5 παρουσιάζεται η Οθόνη προσθήκης νέου ξενοδοχείου σύστημα. Τα στοιχεία καταχώρησης ξενοδοχείου είναι τίτλος, περιγραφή, αστέρια, στοιχεία επικοινωνίας, τοποθεσία, επιλογή εικόνας προφίλ ξενοδοχείου, αριθμός δωματίων, επιλογή παρεχόμενων ανέσεων και ανέβασμα φωτογραφιών.

## Οθόνη Αναζήτησης Ξενοδοχείου



The screenshot shows a web form titled "Αναζήτηση Ξενοδοχείου". It contains the following fields and options:

- Τίτλος:
- Αστέρια:
- Τοποθεσία:
- Δωμάτια:  μονόκλινο  δίκλινο  ...
- Ανέσεις:  Τηλεόραση  Ψυγείο  ...

At the bottom, there is a button labeled "Αναζήτηση Ξενοδοχείου".

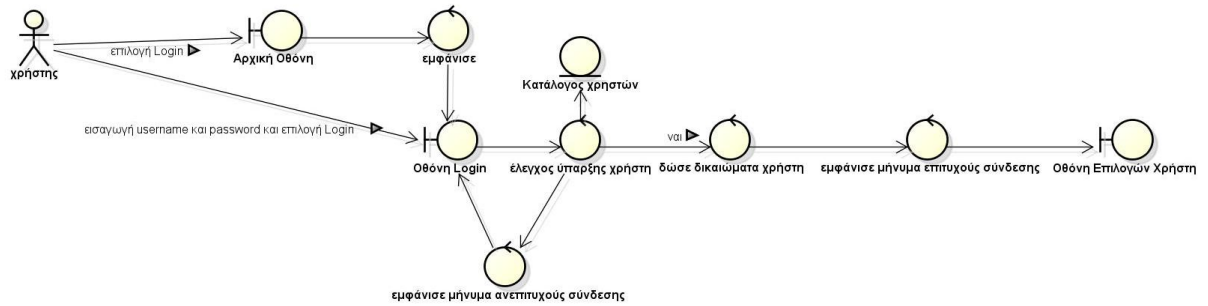
**Σχήμα 5.6:** Οθόνη αναζήτησης ξενοδοχείου

Στο Σχήμα 5.5 παρουσιάζεται η Οθόνη αναζήτησης ξενοδοχείου. Τα κριτήρια αναζήτησης μπορούν να είναι ο τίτλος ξενοδοχείου, τα αστέρια, η τοποθεσία, οι τύποι δωματίων και οι ανέσεις που προσφέρει στους επισκέπτες του.

### 5.2.4. Διαγράμματα Ευρωστίας

Η ανάλυση ευρωστίας αποτελεί μια τεχνική, ενταγμένη στη φάση της ανάλυσης απαιτήσεων, για τη μετάβαση από τις περιπτώσεις χρήσης σε ένα λεπτομερές σχέδιο.

## Διάγραμμα Ευρωστίας 1: ΣΥΝΔΕΣΗ ΣΤΟ ΣΥΣΤΗΜΑ



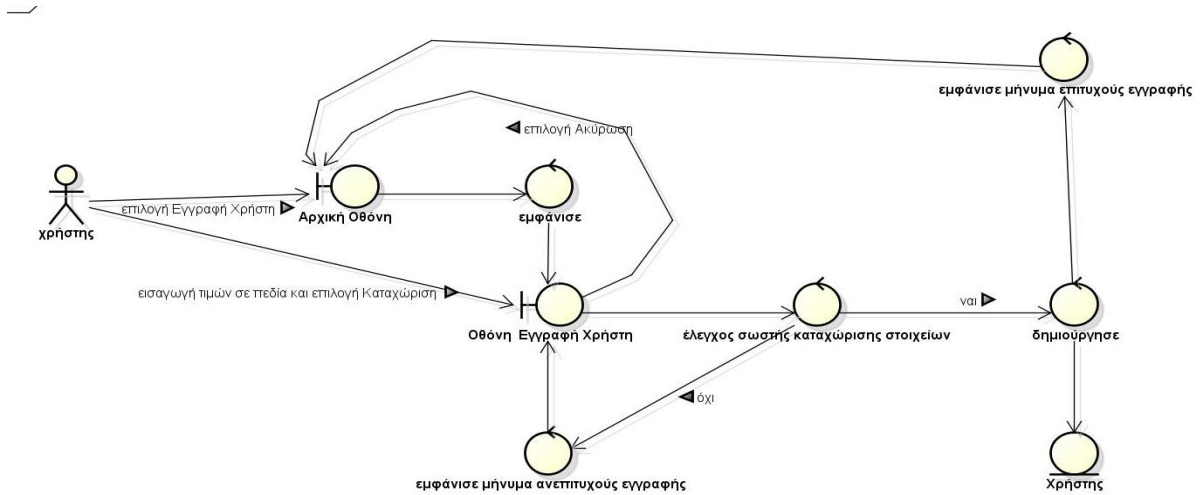
Σχήμα 5.7: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Σύνδεση στο Σύστημα'

## Διάγραμμα Ευρωστίας 2: ΑΠΟΣΥΝΔΕΣΗ



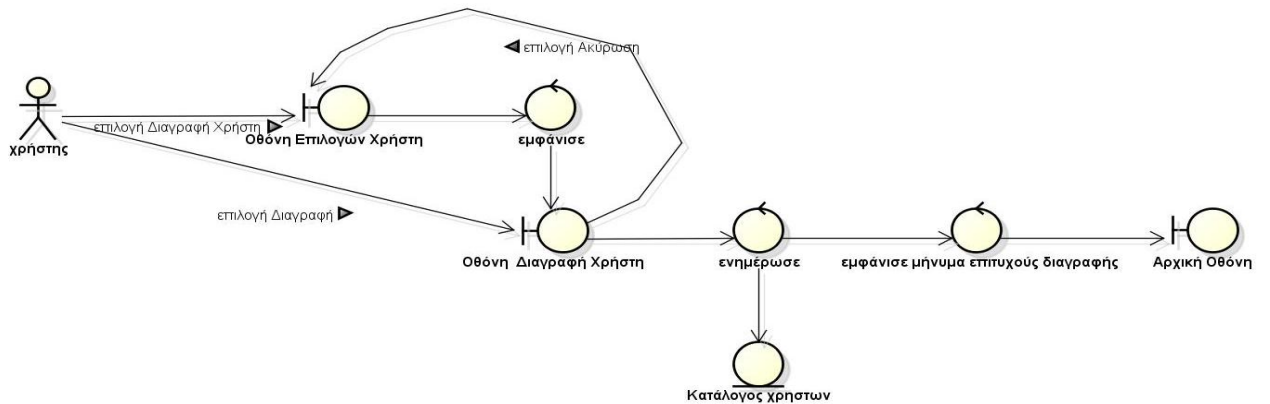
Σχήμα 5.8: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Αποσύνδεση'

### Διάγραμμα Ευρωστίας 3: ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ



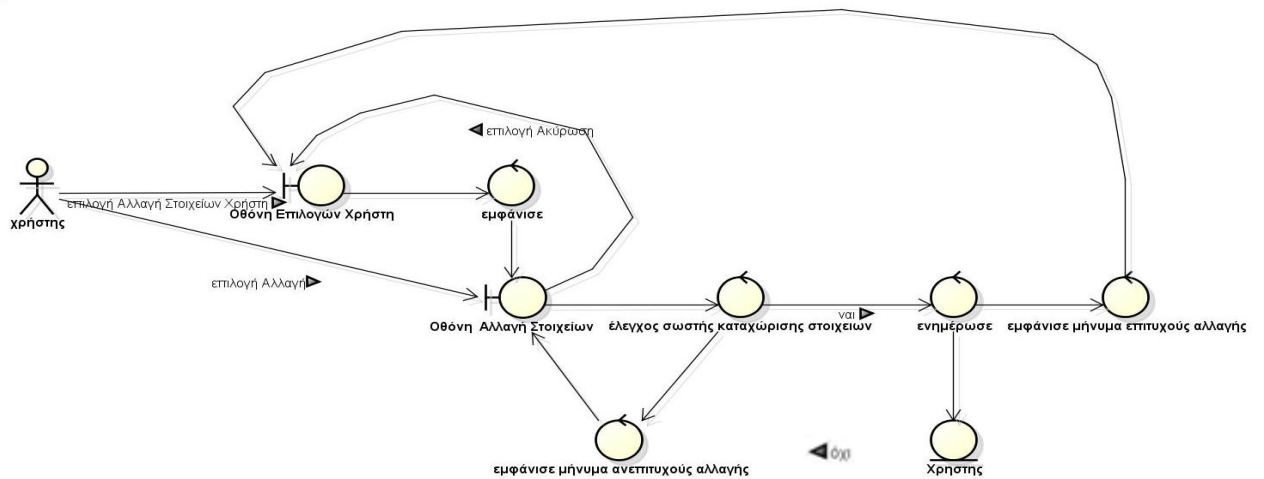
Σχήμα 5.9: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Εγγραφή Χρήστη'

### Διάγραμμα Ευρωστίας 4: ΔΙΑΓΡΑΦΗ ΧΡΗΣΤΗ



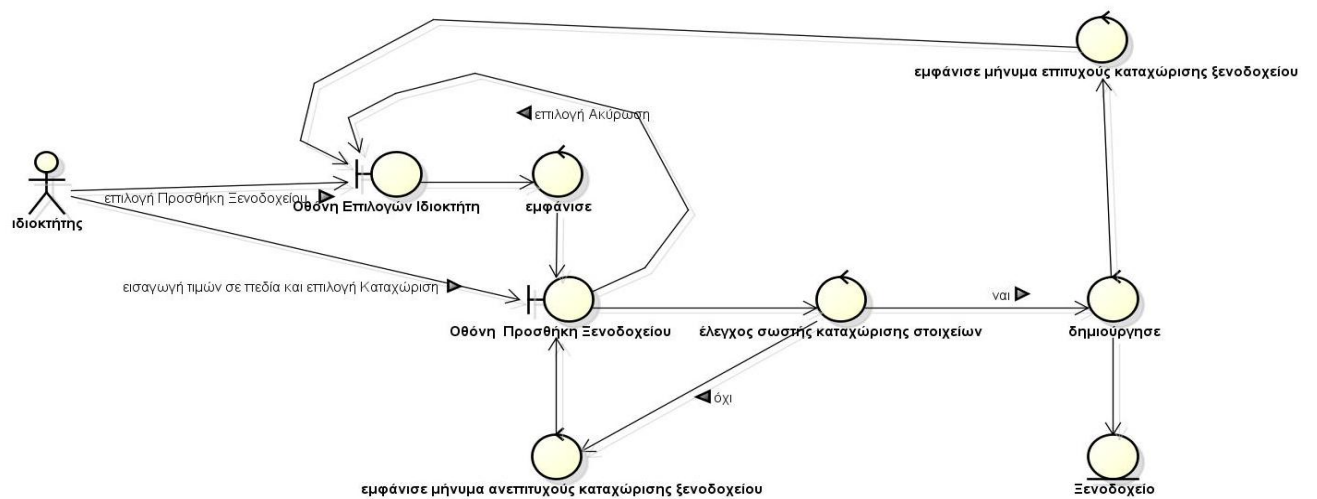
Σχήμα 5.10: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Διαγραφή Χρήστη'

## Διάγραμμα Ευρωστίας 5: ΑΛΛΑΓΗ ΣΤΟΙΧΕΙΩΝ ΛΟΓΑΡΙΑΣΜΟΥ ΧΡΗΣΤΗ



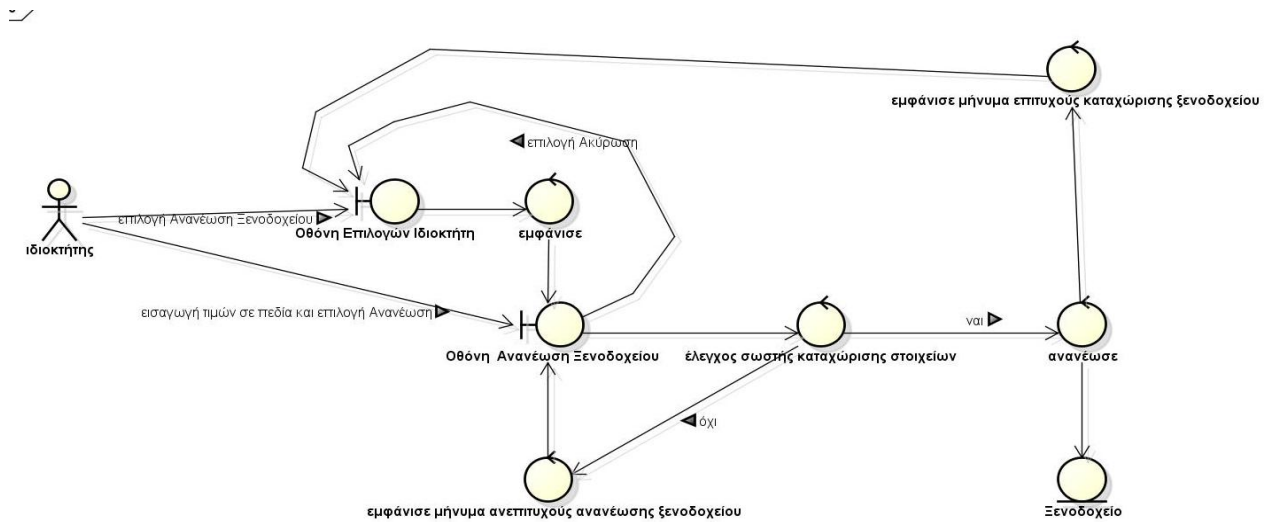
Σχήμα 5.11: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Αλλαγή Στοιχείων Λογαριασμού Χρήστη'.

## Διάγραμμα Ευρωστίας 6: ΠΡΟΣΘΗΚΗ ΞΕΝΟΔΟΧΕΙΟΥ



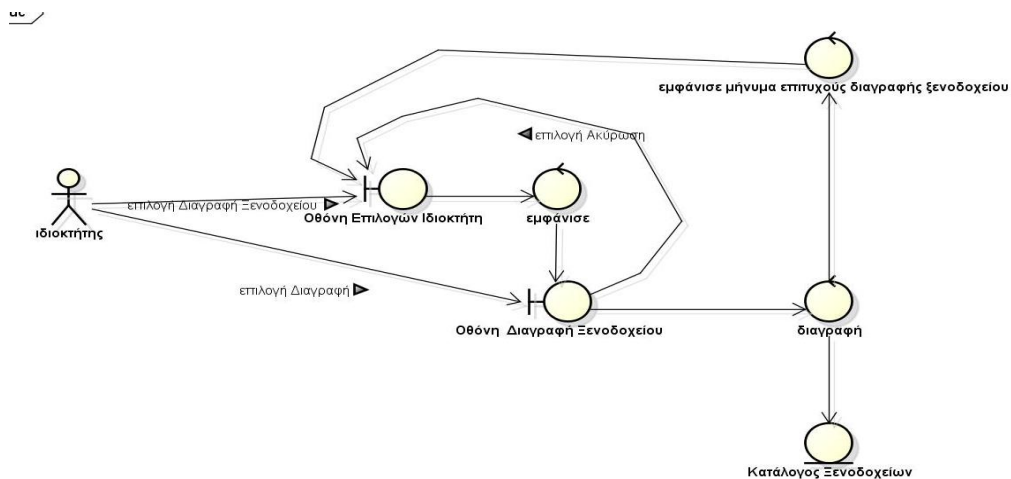
Σχήμα 5.12: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Προσθήκη Ξενοδοχείου'.

## Διάγραμμα Ευρωστίας 7: ΑΝΑΝΕΩΣΗ ΞΕΝΟΔΟΧΕΙΟΥ



Σχήμα 5.13: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Ανανέωση Ξενοδοχείου'.

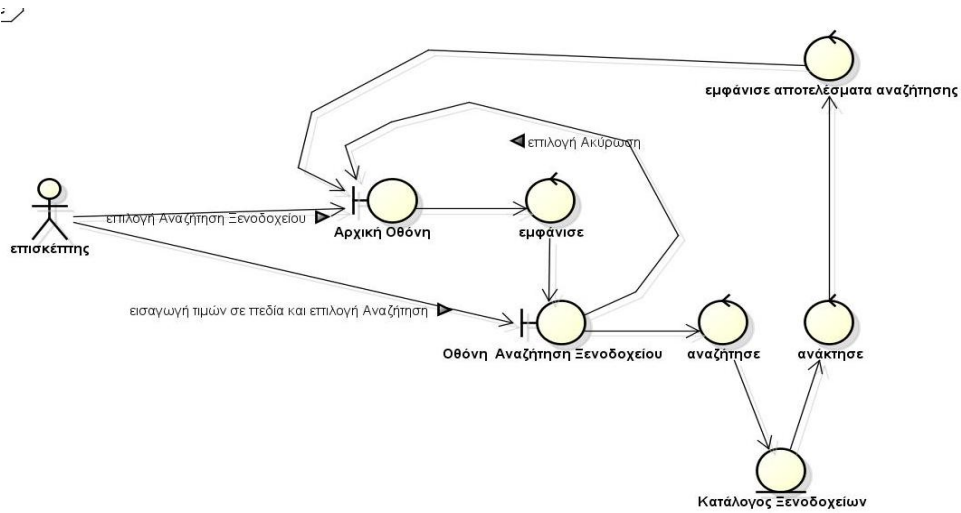
## Διάγραμμα Ευρωστίας 8: ΔΙΑΓΡΑΦΗ ΞΕΝΟΔΟΧΕΙΟΥ



Σχήμα 5.14: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Διαγραφή Ξενοδοχείου'.

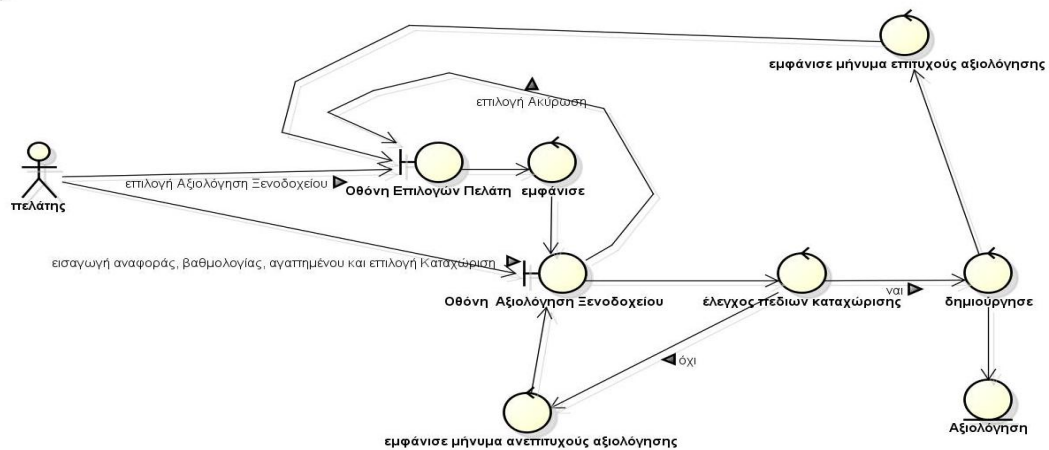


## Διάγραμμα Ευρωστίας 9: ΑΝΑΖΗΤΗΣΗ ΞΕΝΟΔΟΧΕΙΟΥ



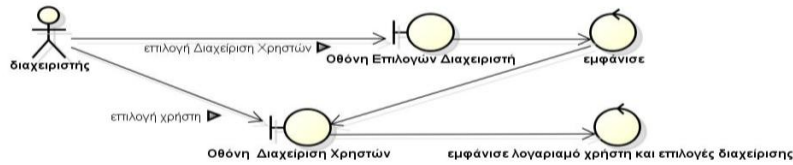
Σχήμα 5.15: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Αναζήτηση Ξενοδοχείου'.

## Διάγραμμα Ευρωστίας 10: ΑΞΙΟΛΟΓΗΣΗ ΞΕΝΟΔΟΧΕΙΟΥ



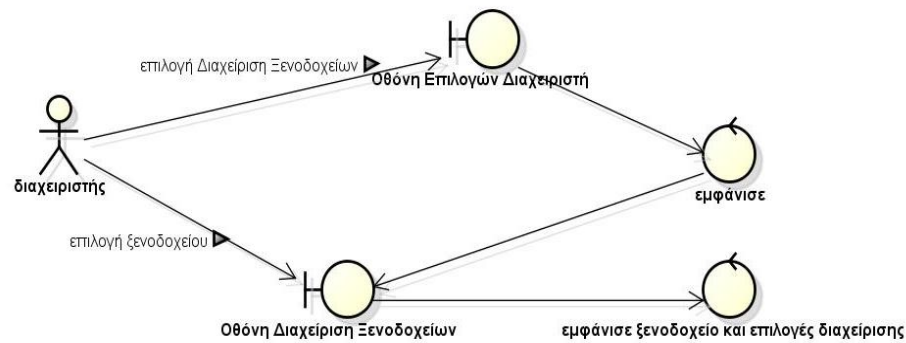
Σχήμα 5.16: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Αξιολόγηση Ξενοδοχείου'.

## Διάγραμμα Ευρωστίας 11: ΔΙΑΧΕΙΡΙΣΗ ΧΡΗΣΤΩΝ



Σχήμα 5.17: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Διαχείριση Χρηστών'.

## Διάγραμμα Ευρωστίας 12: ΔΙΑΧΕΙΡΙΣΗ ΞΕΝΟΔΟΧΕΙΩΝ

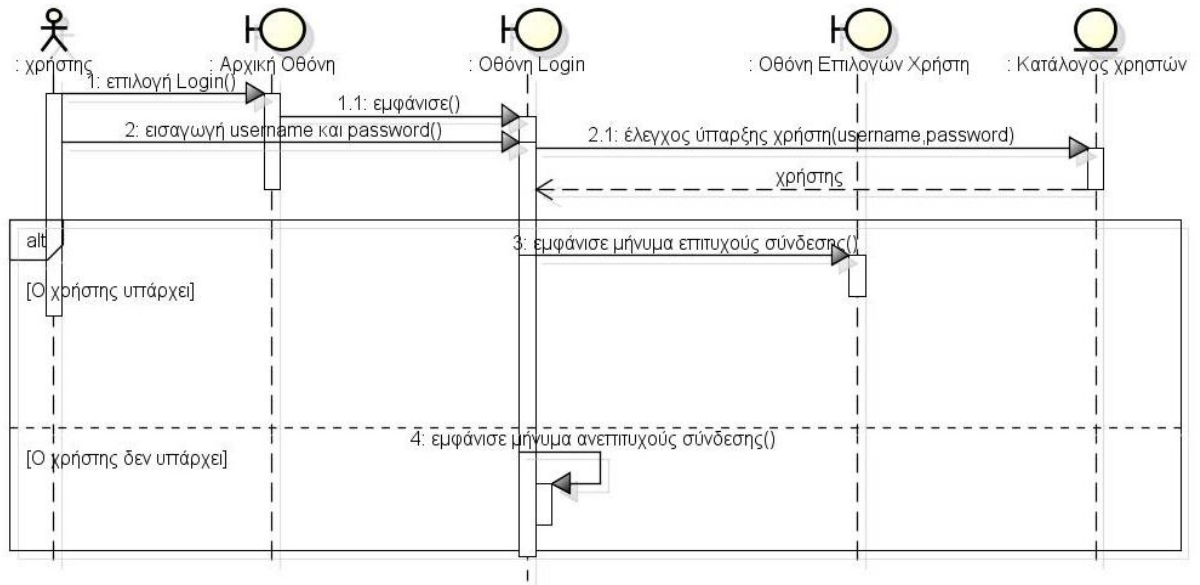


Σχήμα 5.18: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης 'Διαχείριση Ξενοδοχείων'.

### 5.2.5 Διαγράμματα Ακολουθίας

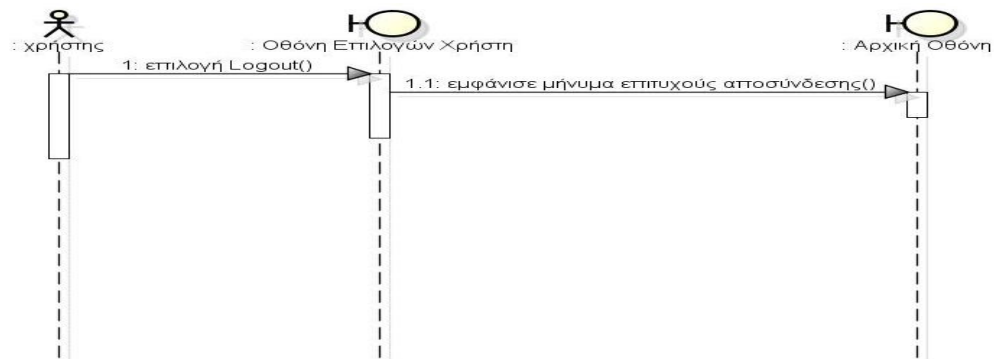
Ένα διάγραμμα ακολουθίας απεικονίζει τα μηνύματα που ανταλλάσσουν τα αντικείμενα του συστήματος μεταξύ τους για την ικανοποίηση της λειτουργικότητας ενός σεναρίου χρήσης.

## Διάγραμμα Ακολουθίας 1: ΣΥΝΔΕΣΗ ΣΤΟ ΣΥΣΤΗΜΑ



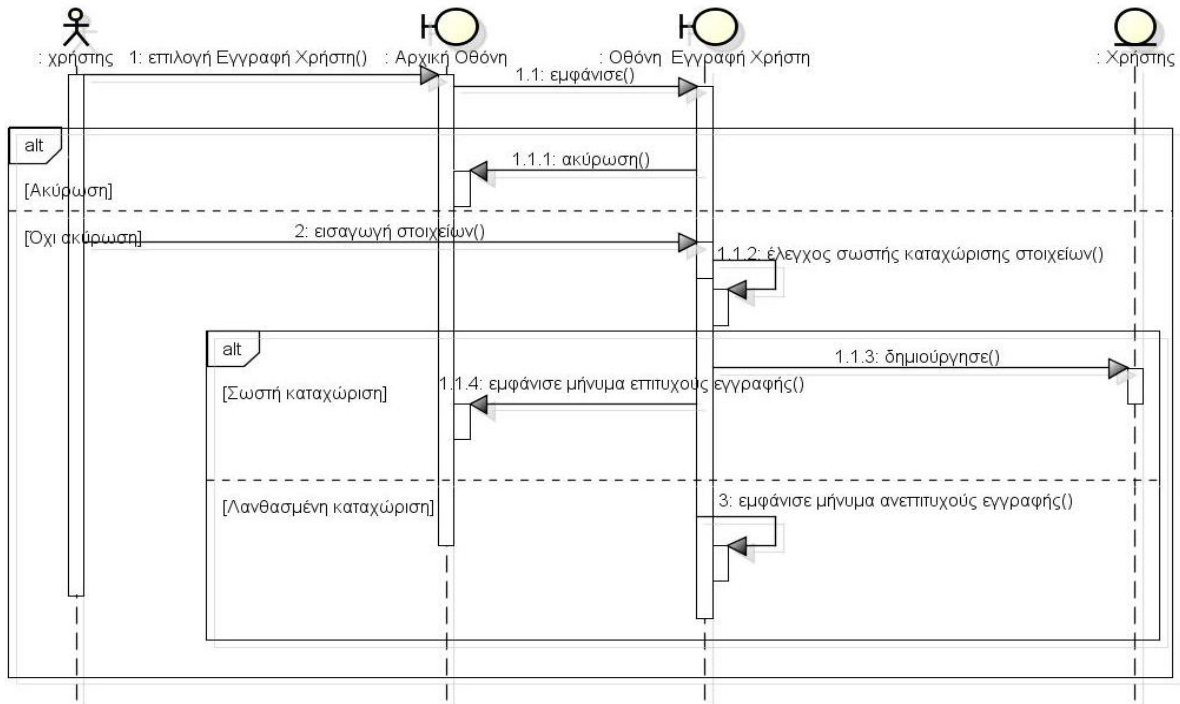
Σχήμα 5.19: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης 'Σύνδεση στο Σύστημα',

## Διάγραμμα Ακολουθίας 2: ΑΠΟΣΥΝΔΕΣΗ



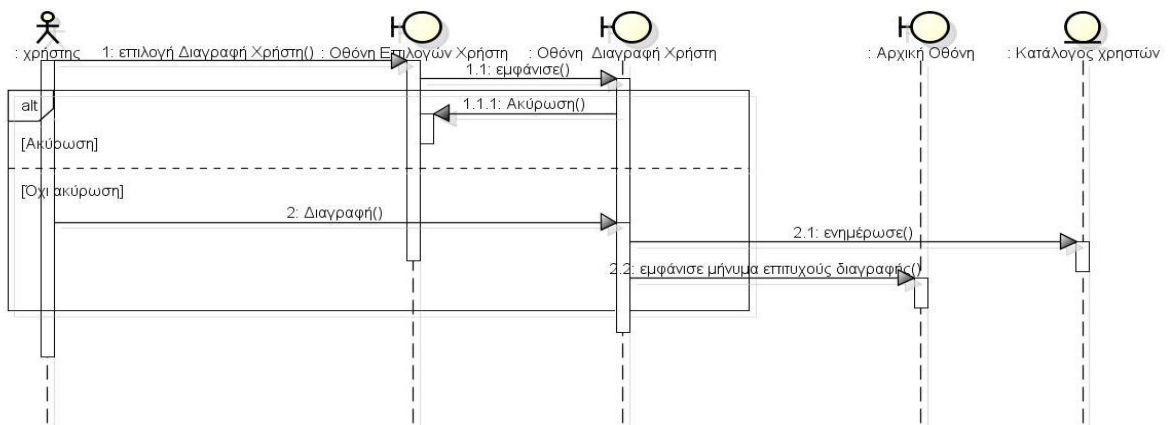
Σχήμα 5.20: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης 'Αποσύνδεση'

### Διάγραμμα Ακολουθίας 3: ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ



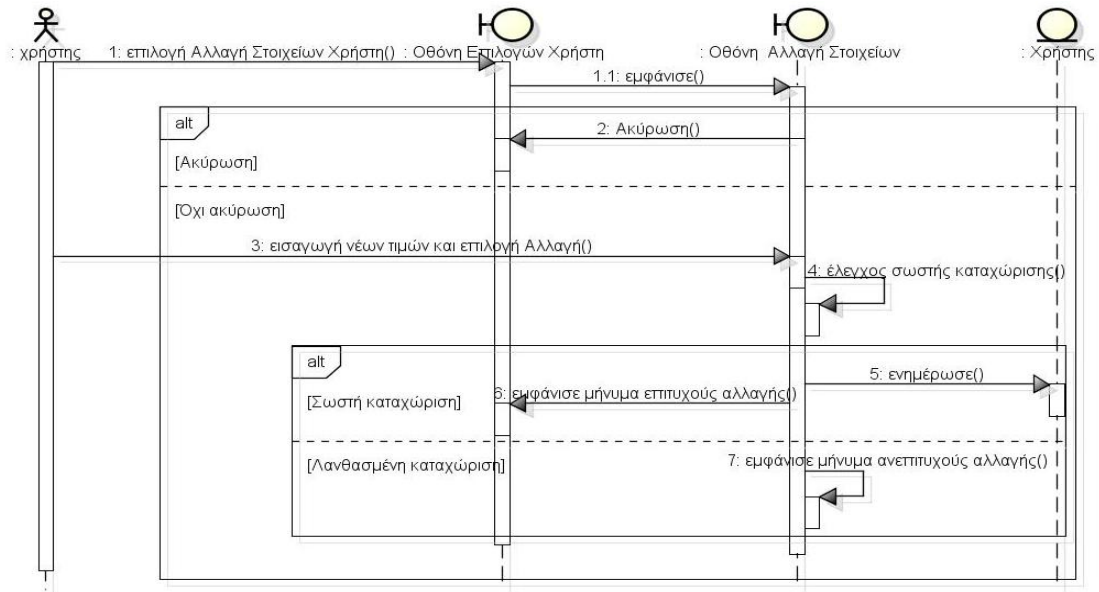
Σχήμα 5.21: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης ‘Εγγραφή Χρήστη’

### Διάγραμμα Ακολουθίας 4: ΔΙΑΓΡΑΦΗ ΧΡΗΣΤΗ



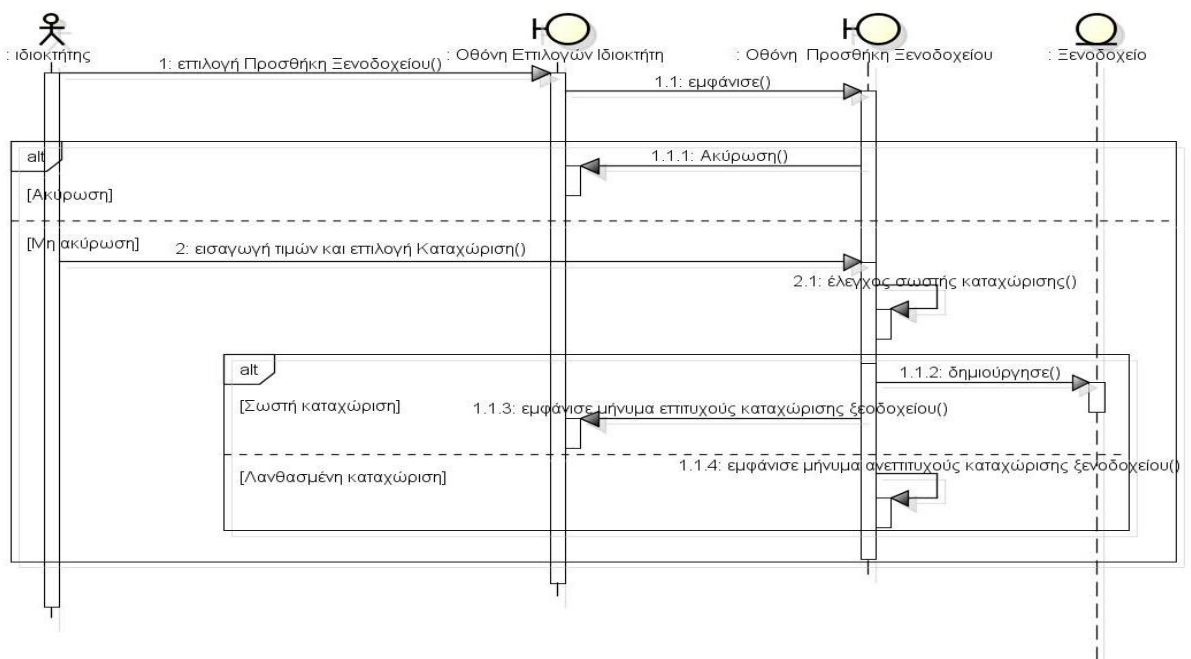
Σχήμα 5.22: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης ‘Διαγραφή Χρήστη’

## Διάγραμμα Ακολουθίας 5: ΑΛΛΑΓΗ ΣΤΟΙΧΕΙΩΝ ΛΟΓΑΡΙΑΣΜΟΥ ΧΡΗΣΤΗ



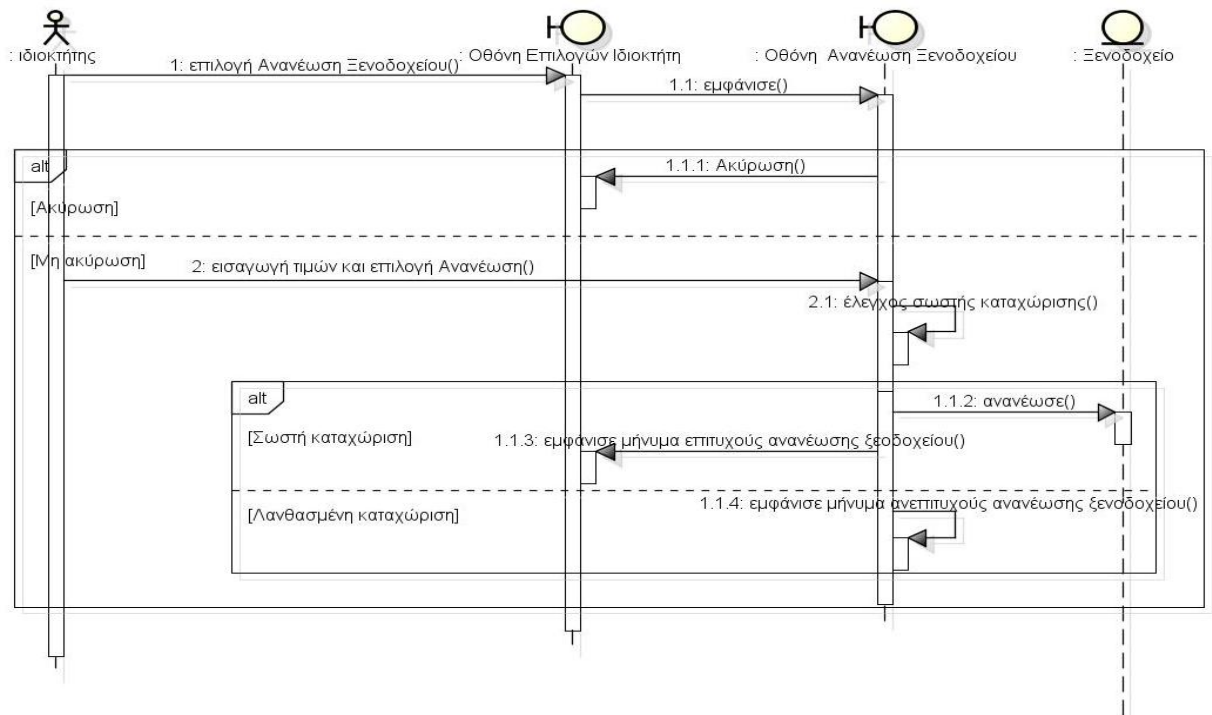
Σχήμα 5.23: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης ‘Αλλαγή Στοιχείων Λογαριασμού Χρήστη’.

## Διάγραμμα Ακολουθίας 6: ΠΡΟΣΘΗΚΗ ΞΕΝΟΔΟΧΕΙΟΥ



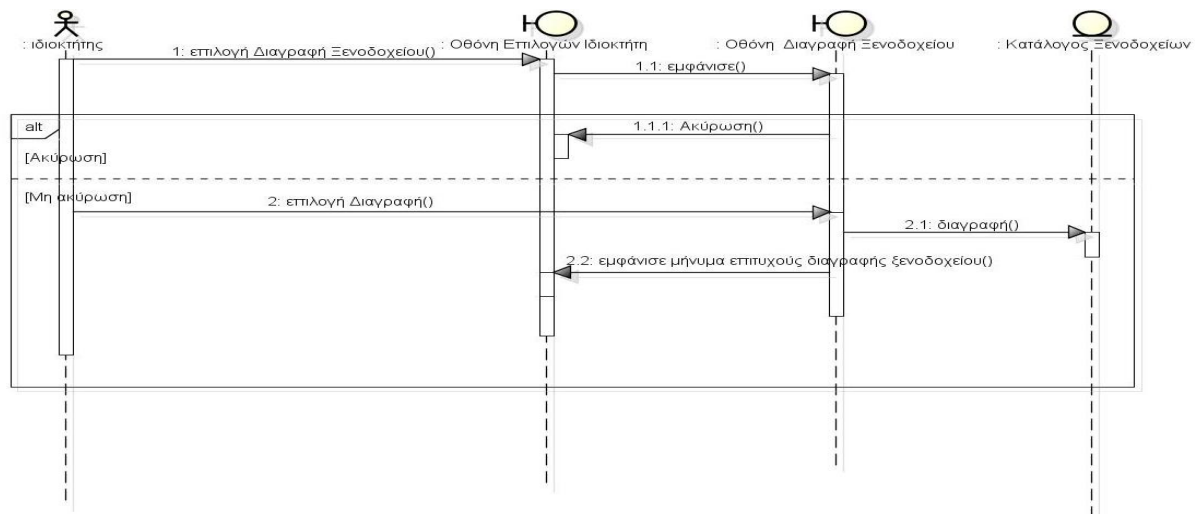
Σχήμα 5.24: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης ‘Προσθήκη Ξενοδοχείου’.

## Διάγραμμα Ακολουθίας 7: ΑΝΑΝΕΩΣΗ ΞΕΝΟΔΟΧΕΙΟΥ



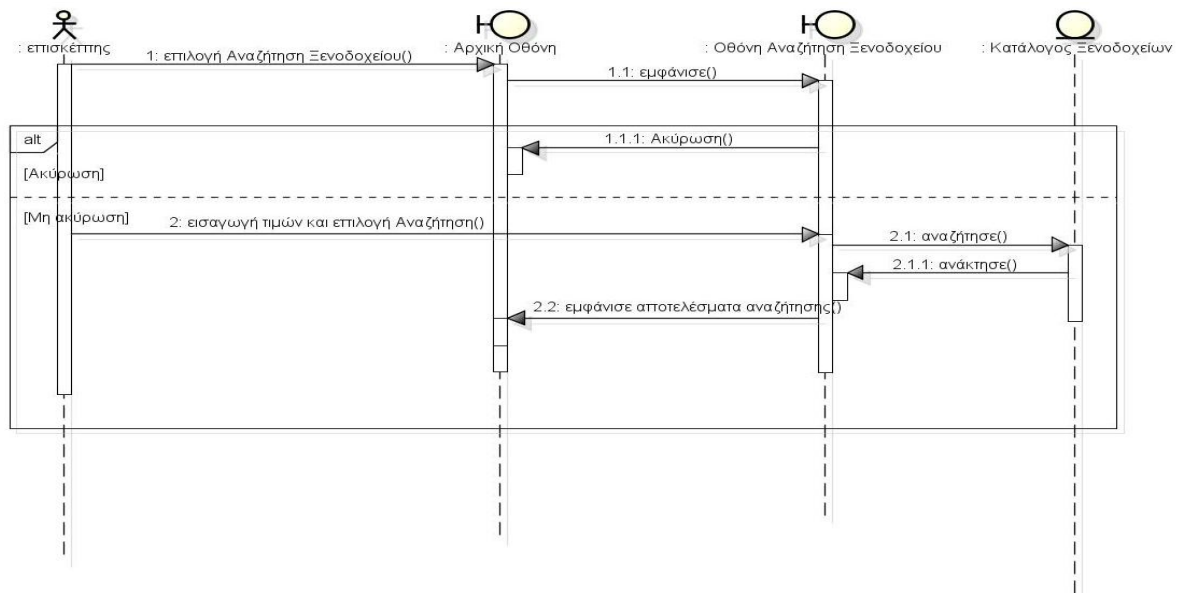
Σχήμα 5.25: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης 'Ανανέωση Ξενοδοχείου'.

## Διάγραμμα Ακολουθίας 8: ΔΙΑΓΡΑΦΗ ΞΕΝΟΔΟΧΕΙΟΥ



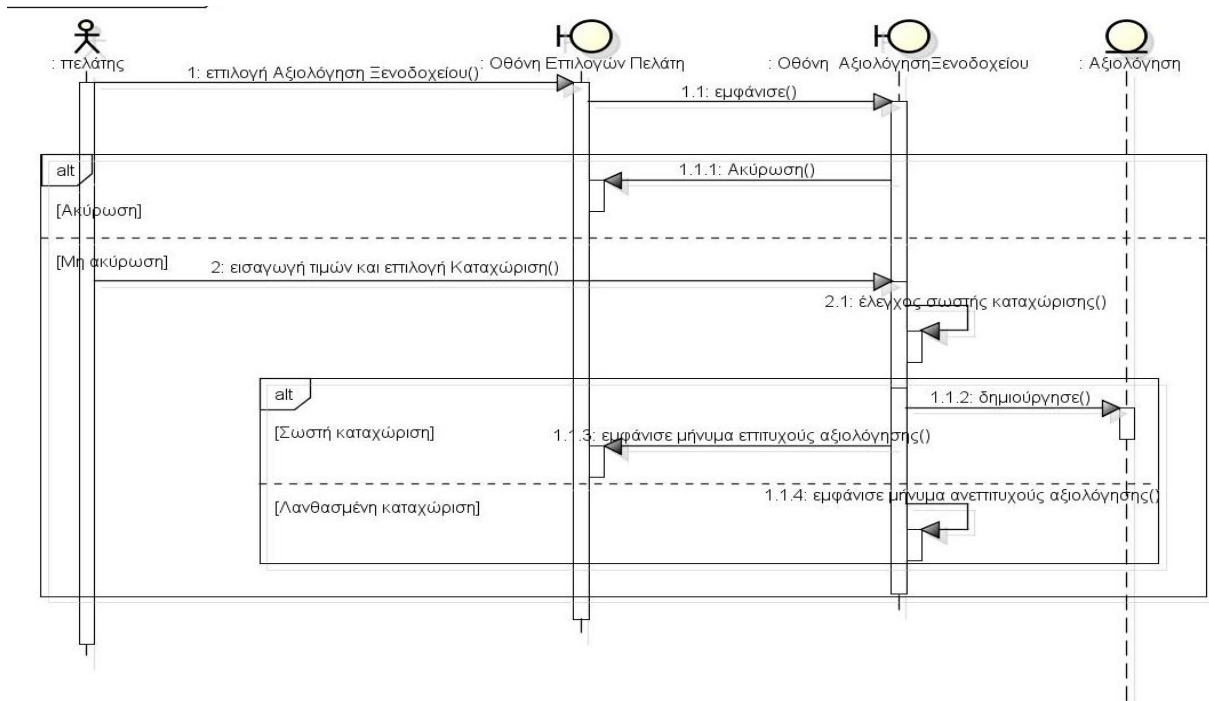
Σχήμα 5.26: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης 'Διαγραφή Ξενοδοχείου'.

## Διάγραμμα Ακολουθίας 9: ΑΝΑΖΗΤΗΣΗ ΞΕΝΟΔΟΧΕΙΟΥ



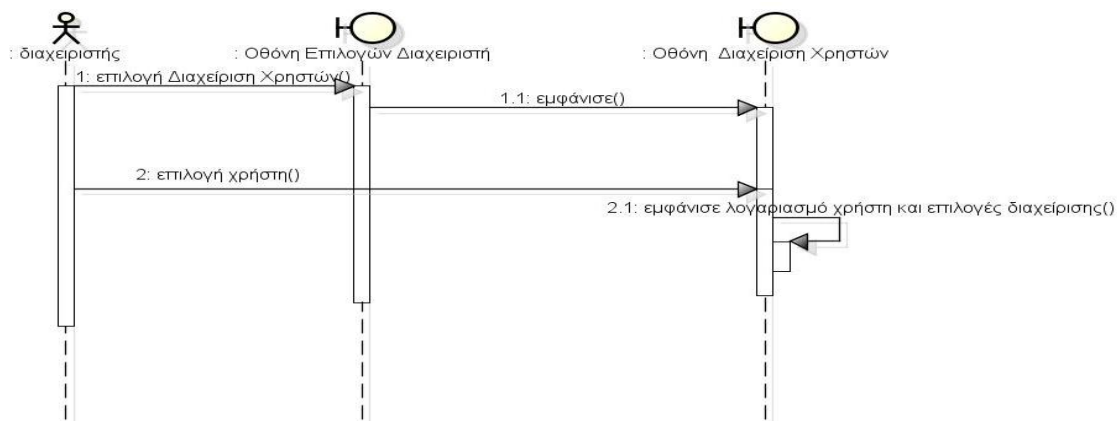
Σχήμα 5.27: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης ‘Αναζήτηση Ξενοδοχείου’.

## Διάγραμμα Ακολουθίας 10: ΑΞΙΟΛΟΓΗΣΗ ΞΕΝΟΔΟΧΕΙΟΥ



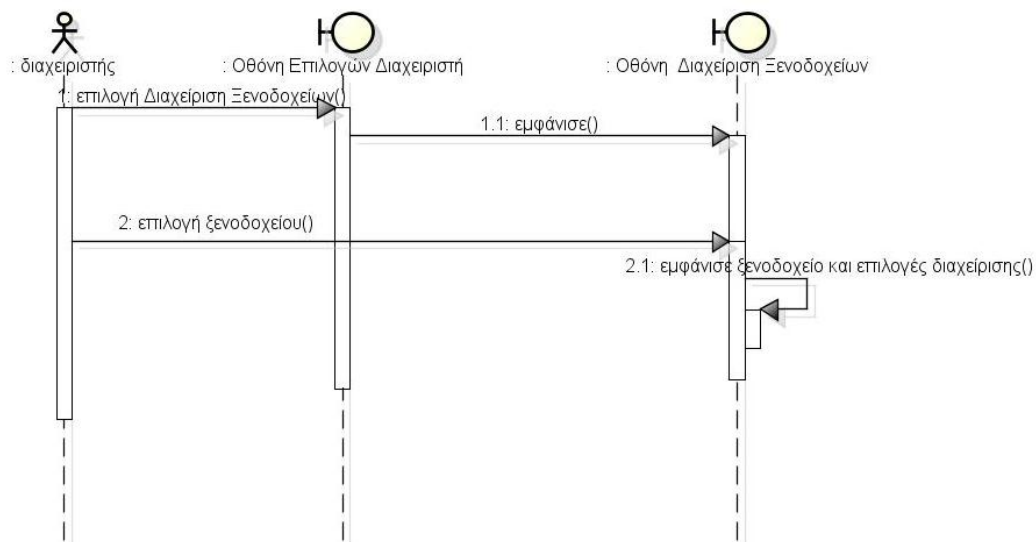
Σχήμα 5.28: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης ‘Αξιολόγηση Ξενοδοχείου’.

## Διάγραμμα Ακολουθίας 11: ΔΙΑΧΕΙΡΙΣΗ ΧΡΗΣΤΩΝ



Σχήμα 5.29: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης 'Διαχείριση Χρηστών'.

## Διάγραμμα Ακολουθίας 12: ΔΙΑΧΕΙΡΙΣΗ ΞΕΝΟΔΟΧΕΙΩΝ

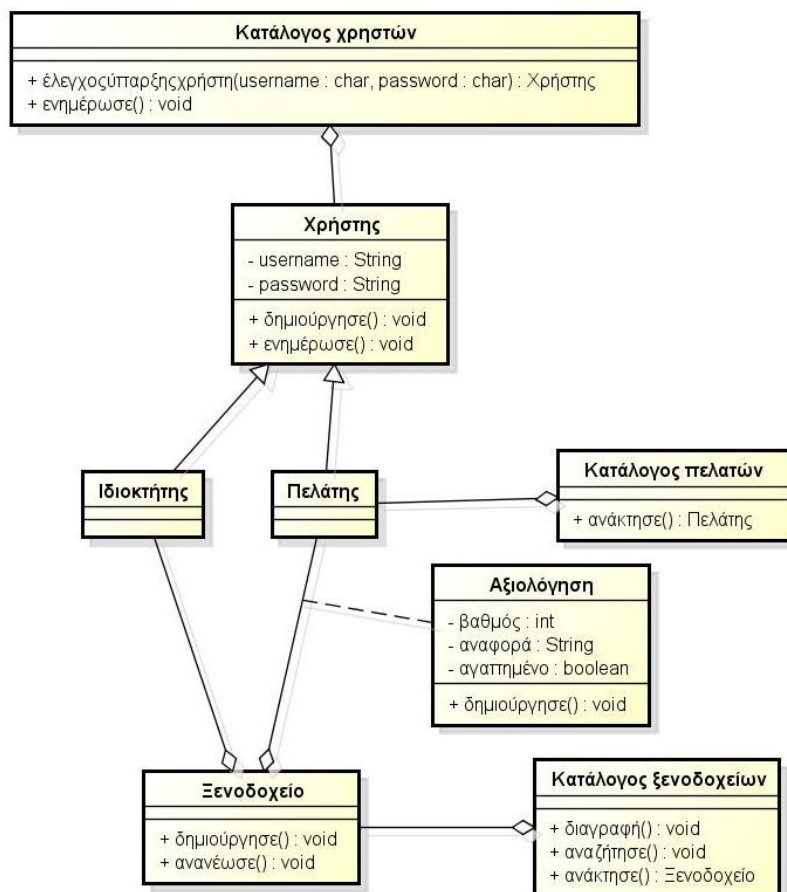


Σχήμα 5.30: Διάγραμμα Ακολουθίας για την Περίπτωση Χρήσης 'Διαχείριση Ξενοδοχείων'.



## 5.2.6 Διάγραμμα Κλάσεων

Σύμφωνα με την ανάλυση που έχει προηγηθεί, καταλήγουμε στο παρακάτω διάγραμμα κλάσεων:



Σχήμα 5.31: Διάγραμμα Κλάσεων.

Παρουσίαση κλάσεων του διαγράμματος:

1. Κατάλογος χρηστών : είναι ο κατάλογος των εγγεγραμμένων χρηστών στο σύστημα και περιλαμβάνει πολλούς χρήστες.
2. Χρήστης: είναι ο εγγεγραμμένος χρήστης που διαθέτει μοναδικό όνομα χρήστη και κωδικό σύνδεσης στο σύστημα.

3. Ιδιοκτήτης: είναι ένας εξουσιοδοτημένος χρήστης με τα αντίστοιχα δικαιώματα.
4. Πελάτης: είναι ένας εξουσιοδοτημένος χρήστης με τα αντίστοιχα δικαιώματα.
5. Κατάλογος πελατών : είναι ο κατάλογος των εγγεγραμμένων πελατών στο σύστημα και περιλαμβάνει πολλούς πελάτες.
6. Ξενοδοχείο: ένα ξενοδοχείο έχει ιδιοκτήτη και πελάτες
7. Αξιολόγηση: ο Πελάτης μπορεί έχει τη δυνατότητα αν θέλει να αξιολογήσει ένα ξενοδοχείο
8. Κατάλογος ξενοδοχείων: είναι ο κατάλογος των καταχωρημένων ξενοδοχείων στο σύστημα και περιλαμβάνει πολλά ξενοδοχεία.

### **5.3 Σχεδίαση Βάσης Δεδομένων**

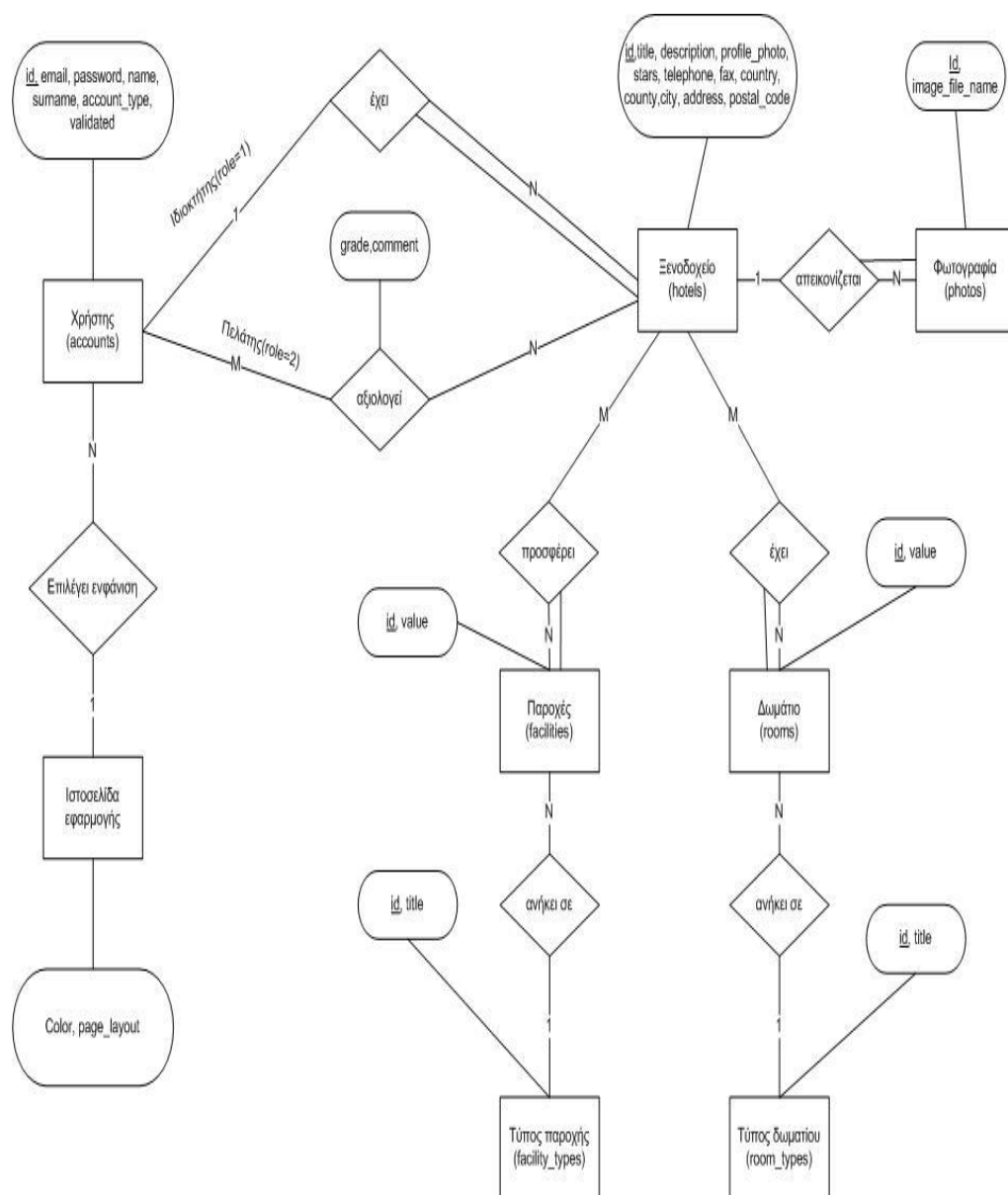
Το σύστημα που θα αναπτυχθεί είναι ένα δυναμικό διαδικτυακό σύστημα, που θα αλληλεπιδρά με μια σχεσιακή βάση δεδομένων. Συνεπώς το επόμενο βήμα είναι να σχεδιαστεί το Διάγραμμα Οντοτήτων Συσχετίσεων και το Σχεσιακό Μοντέλο.

Οι εξουσιοδοτημένοι χρήστες του συστήματος διαθέτουν στοιχεία λογαριασμού σύνδεσης που έχουν αποκτήσει κατόπιν της εγγραφής τους σε αυτό. Οι λογαριασμοί (accounts) μπορούν να ανήκουν είτε σε ξενοδόχους είτε σε ιδιώτες. Ο λογαριασμός του διαχειριστή και τα στοιχεία σύνδεσης υπάρχουν εξ αρχής καταχωρημένα στο σύστημα. Οι εξουσιοδοτημένοι χρήστες μπορούν να επιλέξουν τον τρόπο εμφάνισης της εφαρμογής (χρώμα και διάταξη διεπαφής). Κάθε χρήστης επιλέγει ένα χρώμα και μία διάταξη από τις επιλογές που του δίνονται, ενώ πολλοί χρήστες μπορεί να επιλέξουν τον ίδιο συνδυασμό χρωμάτων και διάταξης.

Ένας ξενοδόχος μπορεί να έχει πολλά ξενοδοχεία στην ιδιοκτησία του. Κάθε ξενοδοχείο μπορεί να έχει πολλά δωμάτια, που ανήκουν σε συγκεκριμένο τύπο δωματίου. Στην εφαρμογή ζητάμε από τον ξενοδόχο να δώσει τον αριθμό δωματίων του ξενοδοχείου του για κάθε τύπο (μονόκλινο, δίκλινο ή τρίκλινο). Επίσης, κάθε ξενοδοχείο μπορεί να προσφέρει πολλές παροχές στους πελάτες του, που ανήκουν που ανήκουν σε

συγκεκριμένο τύπο παροχής. Στην εφαρμογή ζητάμε από τον ξενοδόχο να ορίσει μέσω επιλογής checkbox αν υπάρχει ο κάθε τύπος παροχής (π.χ. πρωινό, τηλεόραση, κ.ά.). Τέλος, κάθε ξενοδοχείο απεικονίζεται σε πολλές φωτογραφίες που έχει ανεβάσει ο ξενοδόχος στο σύστημα. Μία από αυτές μπορεί να επιλεγεί ως φωτογραφία του προφίλ του ξενοδοχείου.

Κάθε ξενοδοχείο αξιολογείται από πολλούς πελάτες μέσω βαθμολογίας και σχολίων, ώστε να υπολογιστεί ο μέσος όρος της βαθμολογίας του και αριθμός σχολίων. Επίσης κάθε πελάτης μπορεί να βαθμολογήσει πολλά ξενοδοχεία ή ακόμα και το ίδιο ξενοδοχείο περισσότερες από μία φορές.



Σχήμα 5.32: Διάγραμμα Οντοτήτων Συσχετίσεων.

<b>Accounts</b>
<u>id</u> email password name surname account_type validated color page

<b>Hotels</b>
<u>id</u> title description <u>profile_photo_id</u> stars telephone fax country county city address postal_code <u>owner_id</u>

<b>Facility_Types</b>
<u>id</u> title

<b>Facilities</b>
<u>id</u> <u>facility_type_id</u> <u>hotel_id</u> value

<b>Room_Types</b>
<u>id</u> title

<b>Rooms</b>
<u>id</u> <u>room_type_id</u> <u>hotel_id</u> value

<b>Ratings</b>
<u>id</u> <u>account_id</u> <u>hotel_id</u> grade comment

<b>Photos</b>
<u>id</u> <u>hotel_id</u> image_file_name

**Σχήμα 5.33:** Σχεσιακό Μοντέλο.

Από το Διάγραμμα Οντοτήτων Συσχετίσεων, προκύπτει το Σχεσιακό Μοντέλο του Σχήματος 5.3.3. Για τις σχέσεις 1:N ορίζουμε ως ξένα κλειδιά τα πρωτεύοντα κλειδιά των οντοτήτων από την πλευρά του 1 στις οντότητες από την πλευρά του N. Για τις σχέσεις M:N δημιουργούμε νέο πίνακα. Στον πίνακα συμπεριλαμβάνουμε ως γνωρίσματα τα πρωτεύοντα κλειδιά των οντοτήτων που συμμετέχουν στη συσχέτιση. Επίσης, σε κάθε περίπτωση συμπεριλαμβάνουμε τυχόν απλά γνωρίσματα της σχέσης (π.χ. η σχέση αξιολογεί έχει τα γνωρίσματα grade και comment τα οποία συμπεριλαμβάνουμε στον πίνακα ratings).

Η Βάση Δεδομένων αποτελείται από τις εξής σχέσεις:

1. Accounts: Περιέχει στοιχεία που σχετίζονται με τους εξουσιοδοτημένους χρήστες του συστήματος. Τα στοιχεία αυτά είναι ο αύξων αριθμός χρήστη, το email, κωδικό σύνδεσης χρήστη, όνομα, επώνυμο, τύπος λογαριασμού, εξουσιοδότηση, χρώμα και layout σελίδας που έχει επιλέξει ο χρήστης. Αντιστοιχούμε στο διαχειριστή `account_type=3`, στους ιδιοκτήτες `account_type =2` και στους πελάτες `account_type =1`, για να διαχωρίσουμε αυτές τις κατηγορίες χρηστών. Όσον αφορά τις επιλογές διεπαφής τις καταχωρούμε σε αυτόν τον πίνακα, γιατί διαφορετικά θα έπρεπε να είχαμε έναν ξεχωριστό πίνακα με όλους τους συνδυασμούς χρώματος-διάταξης.
2. Hotels: Περιέχει στοιχεία που σχετίζονται με τα ξενοδοχεία του συστήματος. Τα στοιχεία αυτά είναι ο αύξων αριθμός ξενοδοχείου, όνομα, περιγραφή, φωτογραφία προφίλ ξενοδοχείου, αστέρια, τηλέφωνο, φαξ, χώρα, νομός, πόλη, διεύθυνση, T.K και κωδικό ιδιοκτήτη. Ο κωδικός ιδιοκτήτη είναι ξένο κλειδί στον πίνακα accounts και δείχνει τον ιδιοκτήτη του ξενοδοχείου. Ομοίως, η φωτογραφία προφίλ ξενοδοχείου είναι ξένο κλειδί στον πίνακα photos που περιέχει όλες τις φωτογραφίες του ξενοδοχείου.
3. Facilities: Περιέχει στοιχεία που σχετίζονται τις παροχές του ξενοδοχείου. Τα στοιχεία αυτά είναι οι παροχές που έχει το ξενοδοχείο. Ο κωδικός τύπου παροχής είναι ξένο κλειδί στον πίνακα Facility\_Types και ο κωδικός ξενοδοχείου είναι ξένο κλειδί στον πίνακα Hotels. Το στοιχείο value παίρνει τιμές 0 ή 1 και αντιστοίχως δείχνει αν δεν υπάρχει ή υπάρχει η συγκεκριμένη παροχή στο ξενοδοχείο.

4. Facility\_Types: Περιέχει στοιχεία που σχετίζονται τους τύπους παροχών. Τα στοιχεία αυτά είναι ο αύξων αριθμός παροχής και η περιγραφή παροχής. Ως παροχή ορίζουμε τηλεόραση, κλιματισμό, μπαρ, wifi, ψυγείο, κ.ά.
5. Rooms: Περιέχει στοιχεία που σχετίζονται τα δωμάτια του ξενοδοχείου. Τα στοιχεία αυτά είναι ο αριθμός δωματίων που υπάρχουν για κάθε αντίστοιχο τύπο δωματίου στο ξενοδοχείο. Ο κωδικός τύπου δωματίου είναι ξένο κλειδί στον πίνακα Room\_Types και ο κωδικός ξενοδοχείου είναι ξένο κλειδί στον πίνακα Hotels. Το στοιχείο value δείχνει τον αριθμό δωματίων του ξενοδοχείου.
6. Room\_Types: Περιέχει στοιχεία που σχετίζονται τους τύπους δωματίων. Τα στοιχεία αυτά είναι ο αύξων αριθμός τύπου δωματίου και ο τύπος. Ως τύπο ορίζουμε το μονόκλινο, δίκλινο, τρίκλινο, σουίτα, κ.ά.
7. Ratings: Περιέχει στοιχεία που σχετίζονται με τις αξιολογήσεις των ξενοδοχείων από τους εξουσιοδοτημένους πελάτες. Τα στοιχεία αυτά είναι οι κωδικοί ιδιοκτήτη και ξενοδοχείου, βαθμός και σχόλια.
8. Photos: Περιέχει στοιχεία που σχετίζονται με τις φωτογραφίες του ξενοδοχείου. Τα στοιχεία αυτά είναι ο κωδικός ξενοδοχείου και οι αντίστοιχες διευθύνσεις όπου αποθηκεύονται οι φωτογραφίες που ανεβάζουν στο σύστημα οι ιδιοκτήτες.

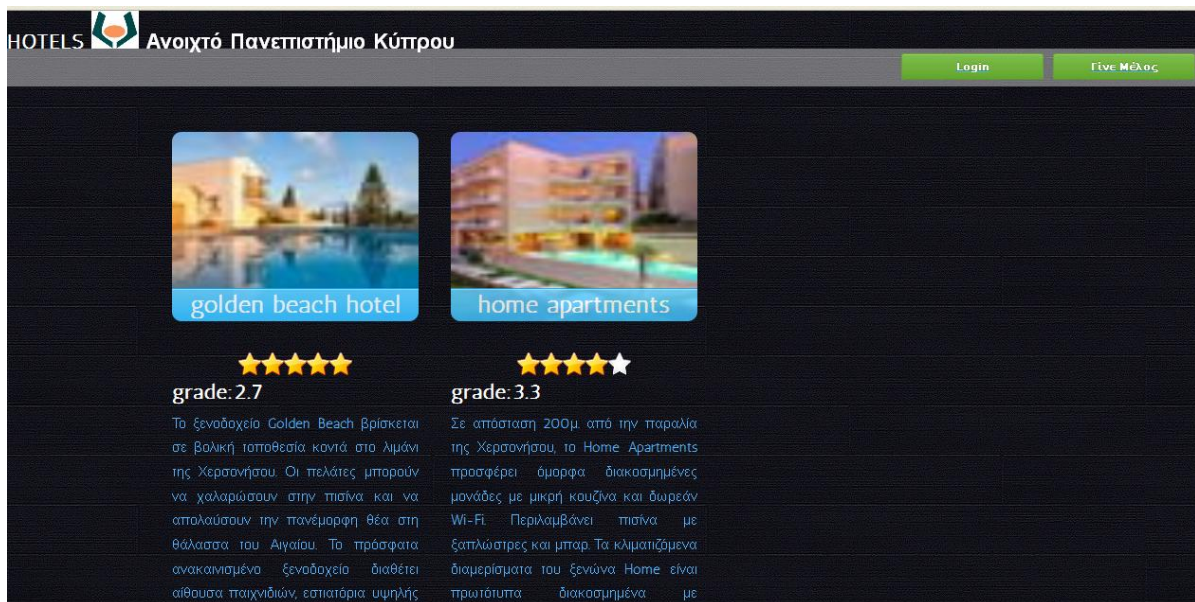
# Κεφάλαιο 6

## Παρουσίαση Εφαρμογής

Στο παρόν κεφάλαιο της μεταπτυχιακής διατριβής θα γίνει αναλυτική παρουσίαση μέσω οθονών του συστήματος που έχει αναπτυχθεί με χρήση του Zend Framework.

### 6.1 Αρχική Σελίδα της εφαρμογής

Στην αρχική σελίδα της εφαρμογής παρουσιάζονται τα καταχωρημένα ξενοδοχεία. Οι πληροφορίες που εμφανίζονται σε αυτή τη σελίδα είναι η εικόνα προφίλ και το όνομα ξενοδοχείου, τα αστέρια, ο μέσος όρος βαθμολογίας που προκύπτει από τις αξιολογήσεις των εγγεγραμμένων πελατών και η περιγραφή του.



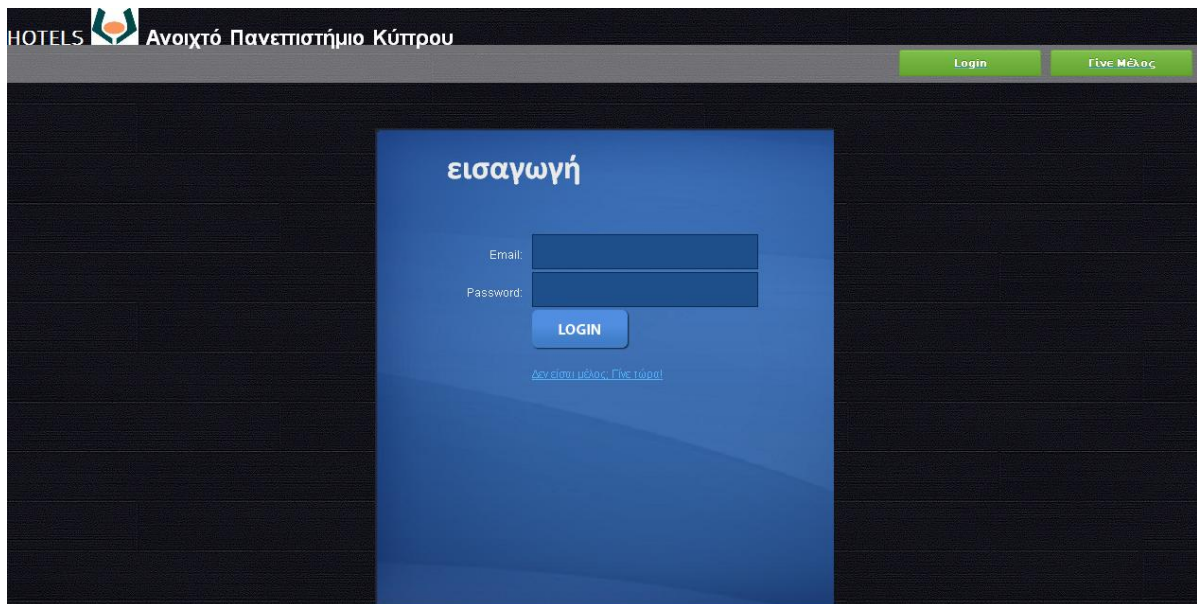
Εικόνα 6.1: Αρχική Οθόνη

## 6.2 Είσοδος στην εφαρμογή για εγγεγραμμένους χρήστες - Login

Το σύστημα υποστηρίζει τη δημιουργία λογαριασμού από τον χρήστη. Η διαδικασία εισόδου είναι η διαδικασία που επιτρέπει στον χρήστη να γίνει αναγνωρίσιμος από το σύστημα.

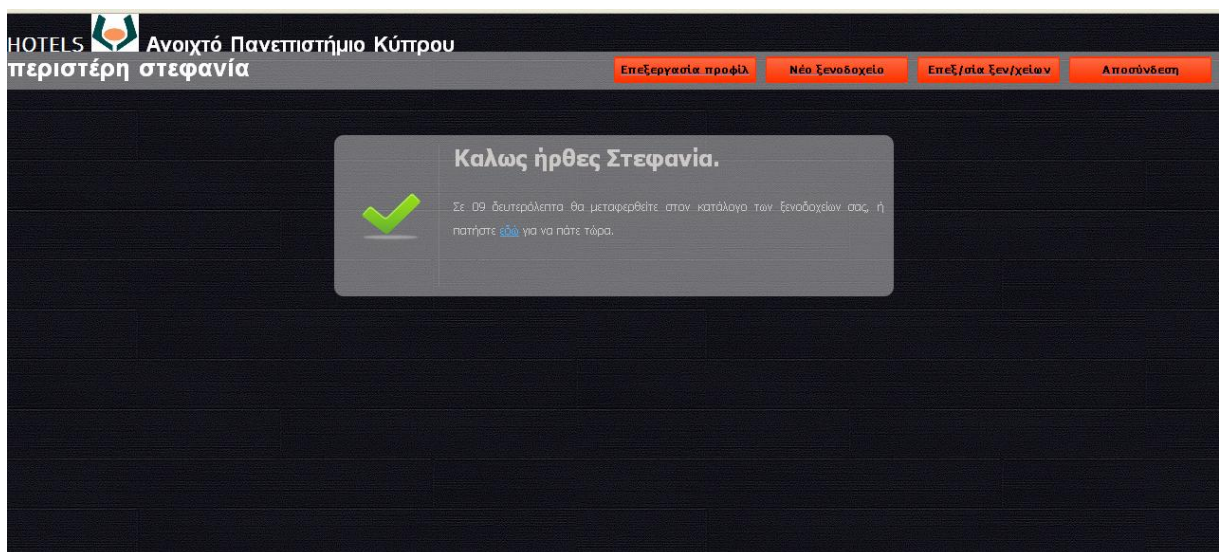
Οι εγγεγραμμένοι χρήστες (Διαχειριστής της εφαρμογής, Ιδιώτες και Ξενοδόχοι) μπορούν να συνδεθούν στο σύστημα επιλέγοντας το σύνδεσμο Login και συμπληρώνοντας τα πεδία Email και Password στην αντίστοιχη φόρμα.





**Εικόνα 6.2:** Οθόνη Login

Αφού ο χρήστης συνδεθεί στην εφαρμογή, το σύστημα τον καλωσορίζει και εμφανίζει το αντίστοιχο μενού επιλογών ανάλογα με το ρόλο του. Ο χαιρετισμός του χρήστη δίνει την εντύπωση στο χρήστη ότι το σύστημα τον αναγνωρίζει και αποτελεί στοιχείο εξατομίκευσης.

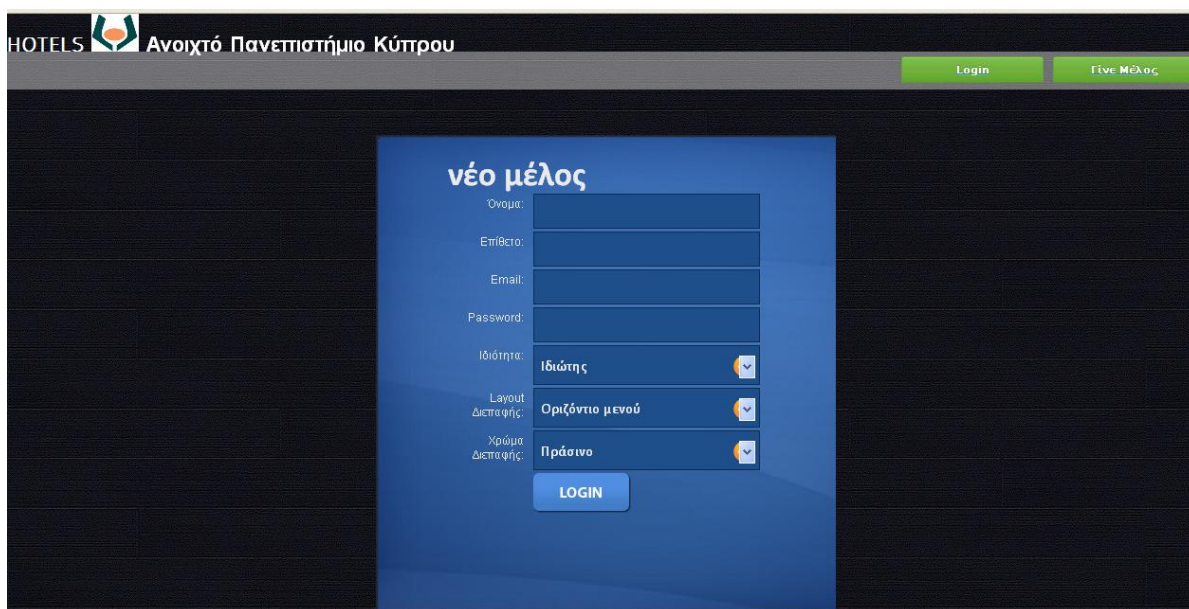


**Εικόνα 6.3:** Μήνυμα επιτυχούς σύνδεσης

## 6.3 Εγγραφή νέου μέλους – Γίνε μέλος

Στην εφαρμογή μπορούν να κάνουν εγγραφή ως νέα μέλη ιδιώτες ή ξενοδόχοι επιλέγοντας το σύνδεσμο Γίνε Μέλος και συμπληρώνοντας τα πεδία Όνομα, Επίθετο, Email, Password και Ιδιότητα (Ιδιώτης ή Ξενοδόχος).

Επίσης, κατά την εγγραφή ο χρήστης μπορεί να επιλέξει πως θέλει να εμφανίζεται η ιστοσελίδα του. Συγκεκριμένα, μπορεί να επιλέξει Layout και Χρώμα διεπαφής. Το layout αφορά στη θέση του μενού επιλογών και δίνονται οι επιλογές Οριζόντιο μενού, Κατακόρυφο αριστερά και Κατακόρυφο δεξιά. Το χρώμα αφορά το χρώμα κάποιων στοιχείων της διεπαφής όπως κουμπιά, κείμενο και φόντο και δίνονται οι επιλογές Μπλε, Πράσινο, Κόκκινο και Κίτρινο. Από τεχνικής άποψης, αυτό γίνεται εφικτό με διαφορετικά CSS για κάθε χρώμα.



The screenshot shows a web interface for a registration form titled "νέο μέλος". The form is set against a dark background. At the top left, the text "HOTELS" is followed by a logo and "Ανοιχτό Πανεπιστήμιο Κύπρου". At the top right, there are two buttons: "Login" and "Γίνε Μέλος". The registration form itself is a blue box with the following fields and options:

- Όνομα: [text input]
- Επίθετο: [text input]
- Email: [text input]
- Password: [text input]
- Ιδιότητα: [dropdown menu with "Ιδιώτης" selected]
- Layout Διεπαφής: [dropdown menu with "Οριζόντιο μενού" selected]
- Χρώμα Διεπαφής: [dropdown menu with "Πράσινο" selected]

At the bottom of the form is a blue "LOGIN" button.

Εικόνα 6.4: Εγγραφή νέου μέλους

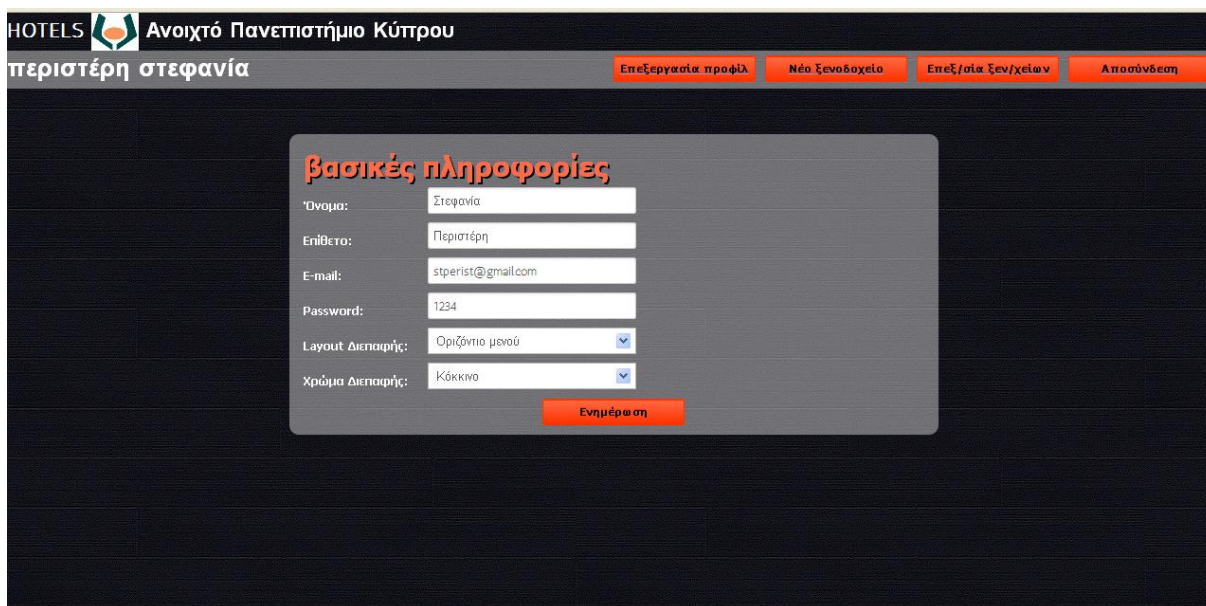
## 6.4 Σύνδεση Ξενοδόχου στην εφαρμογή

Στον συνδεδεμένο στην εφαρμογή Ξενοδόχο παρέχονται οι εξής επιλογές:

- Επεξεργασία προφίλ
- Νέο ξενοδοχείο
- Επεξεργασία ξενοδοχείων
- Αποσύνδεση

### Επεξεργασία προφίλ

Το σύστημα δίνει τη δυνατότητα σε κάθε εγγεγραμμένο χρήστη να αλλάξει τα στοιχεία του προφίλ του (Όνομα, Επίθετο, Email και Password, Layout διεπαφής και Χρώμα διεπαφής).



The screenshot shows a web interface for editing a user profile. At the top, there is a navigation bar with the text "HOTELS Ανοιχτό Πανεπιστήμιο Κύπρου" and a logo. Below this, the user's name "περιστέρη στεφανία" is displayed. To the right of the name are four buttons: "Επεξεργασία προφίλ", "Νέο ξενοδοχείο", "Επεξεργασία ξενοδοχείων", and "Αποσύνδεση". The main content area is a form titled "Βασικές πληροφορίες" (Basic Information). The form contains the following fields:

Όνομα:	Στεφανία
Επίθετο:	Περιστέρη
E-mail:	stperist@gmail.com
Password:	1234
Layout Διεπαφής:	Οριζόντιο μενού
Χρώμα Διεπαφής:	Κόκκινο

At the bottom of the form is a red button labeled "Ενημέρωση" (Update).


Εικόνα 6.5: Αλλαγή στοιχείων μέλους

## Νέο ξενοδοχείο

Οι εγγεγραμμένοι ξενοδόχοι μπορούν να δημιουργήσουν τη σελίδα του ξενοδοχείου τους καταχωρώντας βασικές πληροφορίες:

- Τίτλος
- Περιγραφή
- Αστέρια
- Τηλέφωνο
- Fax
- Τοποθεσία (Χώρα, Νομός, Πόλη, Διεύθυνση, Τ.Κ.)
- Εικόνα προφίλ, επιλέγοντας μία εικόνα από το σύνολο των εικόνων που έχουν επιλέξει για το ξενοδοχείο τους
- Δωμάτια (Αριθμός μονόκλινων, δίκλινων και τρίκλινων)
- Ανέσεις (πρωινό, μπάνιο, τηλεόραση, κλιματισμός, πιστολάκι, τζακούζι, internet, σάουνα, πισίνα, μεσημεριανό, είδη μπάνιου, κουζινάκι, ψυγείο)
- Εικόνες (Εισαγωγή εικόνων, Μεταφόρτωση, Ακύρωση Μεταφόρτωσης, Διαγραφή)



**HOTELS**  **Ανοιχτό Πανεπιστήμιο Κύπρου**

περισσότερη στεφάνια

**Βασικές πληροφορίες** [Επεξεργασία προφίλ](#) [Νέο Ξενοδοχείο](#) [Επέξ/σία Ξεν/χείων](#) [Αποσύνδεση](#)

τίτλος: Golden Beach Hotel

περιγραφή: Το ξενοδοχείο Golden Beach βρίσκεται σε βολική τοποθεσία κοντά στο λιμάνι της Χερσονήσου. Οι πελάτες μπορούν να χαλαρώσουν στην πισίνα και να απολαύσουν την πανέμορφη θέα στη θάλασσα του Αιγαίου.

αστέρια: 5


τηλέφωνο: 2112345678

fax: 2112345678

Χώρα: Ελλάδα Νομός: Λασιθίου

Τοποθεσία: Πόλη: Χερσονήσος Διεύθυνση: Ελ. Βελζέλου 191 Τ.Κ.: 71111

Εικόνα προφίλ:



**Εικόνα 6.6:** Εισαγωγή στοιχείων ξενοδοχείου

**δωμάτια**

μονόκλινα:

διόκλινα:

τρίκλινα:

**ανέσεις**

πρωινό

μπάνιο

τηλέφωνο

κλιματισμός

παρκιλίκι

τηλεοόθ

Internet

σάουνα

πάρκο

μασάζ/μαρινό

είδη μπάνιου

κουζίνα

φρυγιά

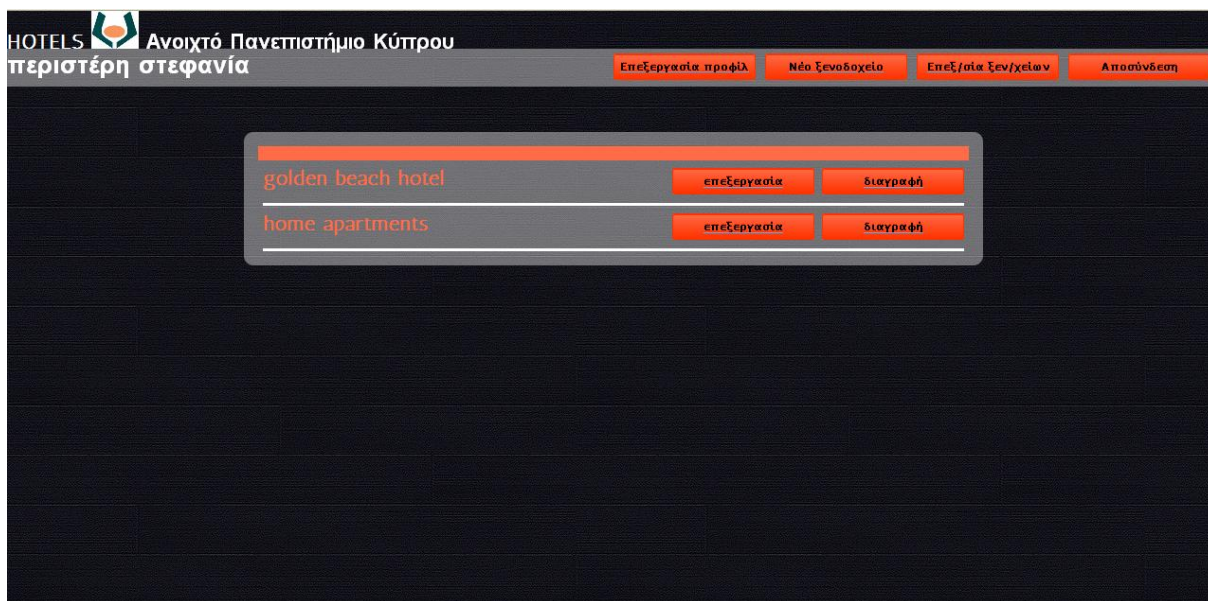
**ΕΙΚΟΝΕΣ**

+ Εισαγωγή εικόνας...

**Εικόνα 6.7:** Εισαγωγή στοιχείων ξενοδοχείου

## Επεξεργασία ξενοδοχείων

Οι εγγεγραμμένοι ξενοδόχοι μπορούν να ενημερώσουν τις πληροφορίες που αφορούν στο ξενοδοχείο τους ή να διαγράψουν τη σελίδα του ξενοδοχείου τους.



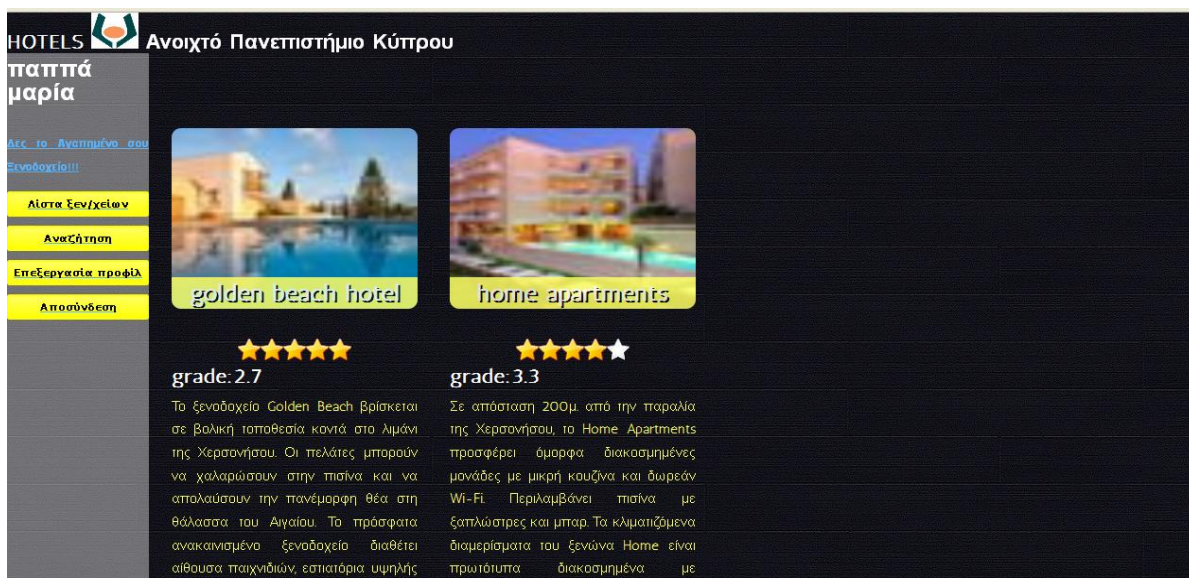
Εικόνα 6.8: Επεξεργασία ξενοδοχείων

## 6.4 Σύνδεση Ιδιώτη στην εφαρμογή

Στον συνδεδεμένο στην εφαρμογή Ιδιώτη παρέχονται οι εξής επιλογές:

- Λίστα ξενοδοχείων (Μετάβαση στην αρχική σελίδα της εφαρμογής)
- Αναζήτηση
- Επεξεργασία προφίλ
- Αποσύνδεση

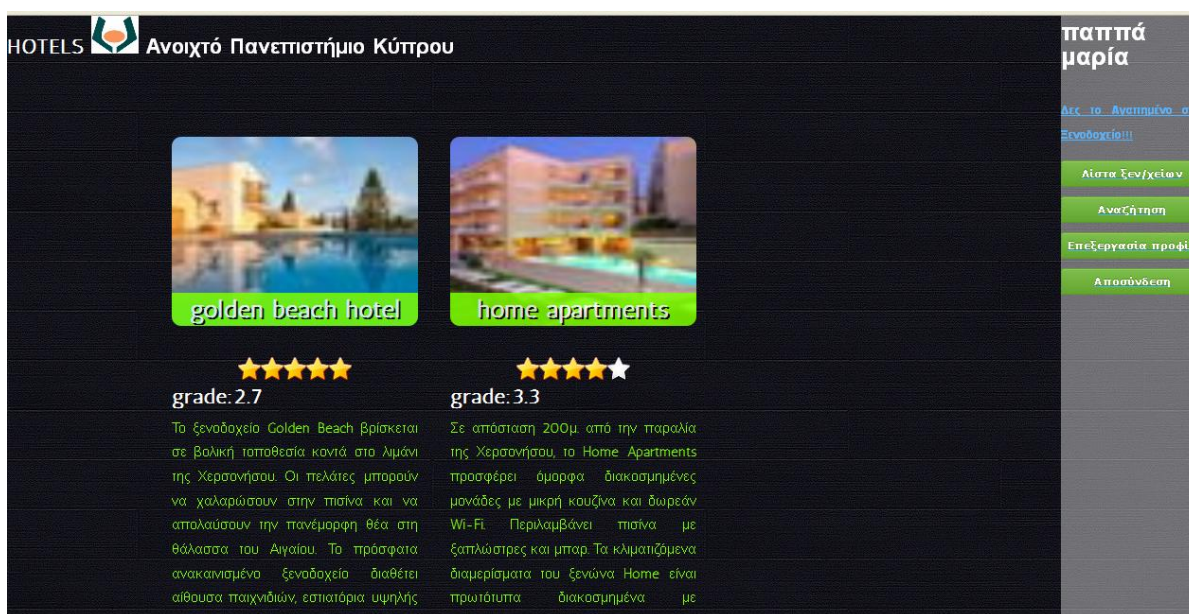




**Εικόνα 6.9:** Λίστα ξενοδοχείων

Μόλις ένας Ιδιώτης συνδεθεί στην εφαρμογή βλέπει τον κατάλογο των καταχωρημένων ξενοδοχείων και ένα μενού με τις επιλογές που του προσφέρονται.

Στην εικόνα που ακολουθεί, βλέπουμε την οθόνη λίστα ξενοδοχείων για τον ίδιο χρήστη με διαφορετικό χρώμα και διάταξη του μενού της διεπαφής. Επίσης, υπάρχει ο σύνδεσμος 'Δες το αγαπημένο σου ξενοδοχείο,.'. Το σύστημα βρίσκει το αγαπημένο ξενοδοχείο από τις βαθμολογίες που έχει βάλει ο συγκεκριμένος χρήστης στα ξενοδοχεία. Το αγαπημένο είναι αυτό που το έχει αξιολογήσει ο ίδιος με τη μεγαλύτερη βαθμολογία.



**Εικόνα 6.10:** Λίστα ξενοδοχείων και σύνδεσμος 'Δες το αγαπημένο σου ξενοδοχείο,'

Στη συνέχεια παρουσιάζουμε τον κώδικα που υπολογίζει το αγαπημένο ξενοδοχείο κάθε χρήστη. Η συνάρτηση `getFavHotForUser($accountId)` δέχεται ως όρισμα το αναγνωριστικό του ιδιώτη και επιστρέφει το ξενοδοχείο, το οποίο έχει βαθμολογηθεί με το μεγαλύτερο βαθμό από το συγκεκριμένο χρήστη. Πιο αναλυτικά, εκτελείται ένα ερώτημα επιλογής (SELECT) του μέγιστου βαθμού (MAX(grade)) και του αναγνωριστικού του ξενοδοχείου (hotel\_id) από τον πίνακα των βαθμολογιών (ratings) όπου το αναγνωριστικό του λογαριασμού είναι ίσο με το αναγνωριστικό του κάθε ιδιώτη είναι συνδεδεμένος στην εφαρμογή.

```
public function getFavHotForUser($accountId)
{
    $query = "SELECT MAX(grade),hotel_id FROM ratings WHERE account_id = ?";

    $result = $this->getDbTable()->getAdapter()->query($query, array($accountId));

    if (1 == count($result)) {
        $row = $result->fetch();
        return $row['hotel_id'];
    } else {
        return 0;
    }
}
```

Κάνοντας κλικ πάνω σε κάποιο ξενοδοχείο της λίστας, οι εγγεγραμμένοι πελάτες μπορούν να δουν όλες τις πληροφορίες που έχουν καταχωρήσει οι ξενοδόχοι ή που έχει υπολογίσει το σύστημα (μέσος όρος βαθμολογιών και αριθμός σχολίων), εκτελώντας τα αντίστοιχα ερωτήματα.

Στη συνέχεια παρουσιάζουμε τον κώδικα που υπολογίζει το μέσο όρο βαθμολογίας για κάθε ξενοδοχείο. Η συνάρτηση `getGradeForHotel($hotelId)` δέχεται ως όρισμα το αναγνωριστικό του ξενοδοχείου και επιστρέφει το μέσο όρο βαθμολογίας του, που προκύπτει από τις αξιολογήσεις των εξουσιοδοτημένων ιδιωτών. Πιο αναλυτικά, εκτελείται ένα ερώτημα επιλογής (SELECT) του μέσου όρου στρογγυλοποιημένου στο πρώτο δεκαδικό (`round(SUM(grade) / COUNT(*),1)`) από τον πίνακα των βαθμολογιών



(ratings) όπου το αναγνωριστικό του ξενοδοχείου είναι ίσο με το αναγνωριστικό του κάθε ξενοδοχείου που προβάλλεται στην εφαρμογή.

```
public function getGradeForHotel($hotelId)
{
    $query = "SELECT round(SUM(grade) / COUNT(*),1) as grade FROM ratings
WHERE hotel_id = ?";

    $result = $this->getDbTable()->getAdapter()->query($query, array($hotelId));

    if (1 == count($result)) {
        $row = $result->fetch();
        return $row['grade'];
    } else {
        return 0;
    }
}
```

Στη συνέχεια παρουσιάζουμε τον κώδικα που υπολογίζει τον αριθμό των αξιολογήσεων-σχολίων για κάθε ξενοδοχείο. Η συνάρτηση `getNumGradesForHotel($hotelId)` δέχεται ως όρισμα το αναγνωριστικό του ξενοδοχείου και επιστρέφει το πλήθος των σχολίων, που προκύπτει από τις αξιολογήσεις των εξουσιοδοτημένων ιδιωτών. Πιο αναλυτικά, εκτελείται ένα ερώτημα επιλογής (SELECT) του πλήθους όλων των βαθμολογιών (COUNT(\*)) από τον πίνακα των βαθμολογιών (ratings) όπου το αναγνωριστικό του ξενοδοχείου είναι ίσο με το αναγνωριστικό του κάθε ξενοδοχείου που προβάλλεται στην εφαρμογή.

```
public function getNumGradesForHotel($hotelId)
{
    $query = "SELECT COUNT(*) as numgrades FROM ratings WHERE hotel_id = ?";
```

```
$result = $this->getDbTable()->getAdapter()->query($query, array($hotelId));
```

```
if (1 == count($result)) {
    $row = $result->fetch();
    return $row['numgrades'];
} else {
    return 0;
}
```

**HOTELS** **Ανοιχτό Πανεπιστήμιο Κύπρου**  
**παππά μαρία**  
**home apartments** ★★★★★  
**βαθμολογία σχολίων : 3.3 (4 σχόλια)**

Σε απόσταση 200μ. από την παραλία της Χερσονήσου, το Home Apartments προσφέρει όμορφα διακοσμημένες μονάδες με μικρή κουζίνα και δωρεάν Wi-Fi. Περιλαμβάνει πισίνα με ξαπλώστρες και μπαρ. Τα κλιματιζόμενα διαμερίσματα του ξενοτύπου Home είναι πρωτίτως διακοσμημένα με χρωματιστές λεπτομέρειες και παραδοσιακά ξύλινα έπιπλα. Όλα διαθέτουν τραπεζαρία και καθιστικό. Όλα τα διαμερίσματα περιλαμβάνουν μπαλκόνι με θέα στην πισίνα και στον κήπο. Οι επισκέπτες μπορούν να απολαύσουν ποτά και κοκτέιλ στο μπαρ της πισίνας, ακούγοντας lounge μουσική. Το ευρωπαϊκό πρωινό με βιολογικά προϊόντα σερβίρεται καθημερινά στον κήπο ή στην αίθουσα πρωινού του ξενοτύπου. Το συγκρότημα διαθέτει επίσης εγκαταστάσεις μπάμπεκικιου και παραδοσιακό φούρνο με ξύλα. Το πολυσύχναστο κέντρο της Χερσονήσου απέχει μόλις 5 λεπτά με τα πόδια. Το γραφικό χωριό Κουτουλουφάρι βρίσκεται σε απόσταση 1,5χλμ. και το Διεθνές Αεροδρόμιο Νίκος Καζαντζάκης είναι 20χλμ. μακριά. Διατίθεται δωρεάν χώρος στάθμευσης κοντά στις εγκαταστάσεις.

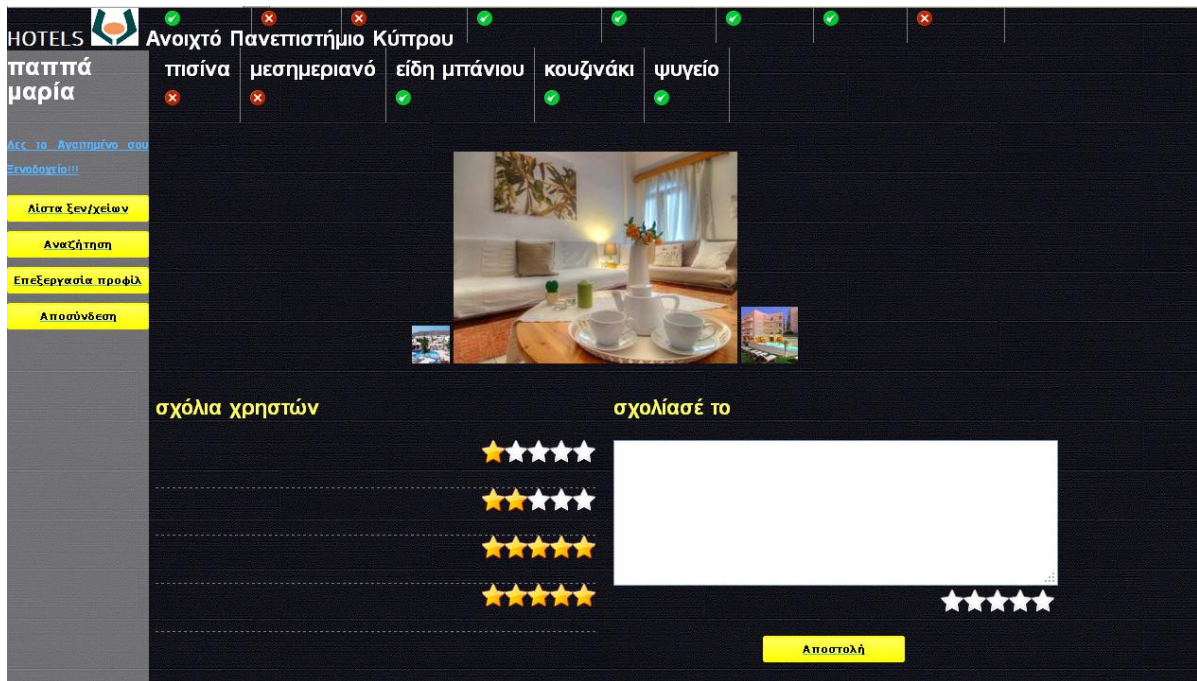
**τηλέφωνο: 11111**  
**fax: 11111**  
**Χερσόνησος Λασιθι**  
**Κυψέλης**  
**11111**  
**Ελλάδα**

μονόκλινο	δίκλινο	τρίκλινο
34	0	0

πρωινό	μπάνιο	τηλεόραση	κλιματισμός	πιστολάκι	τζακούζι	internet	σάουνα
✓	✗	✗	✓	✓	✓	✓	✗

**Εικόνα 6.11:** Αναλυτικές πληροφορίες ξενοδοχείου

Επίσης, οι συνδεδεμένοι ιδιώτες μπορούν γράψουν σχόλια για το ξενοδοχείο και να το βαθμολογήσουν. Από τις βαθμολογίες υπολογίζεται ο μέσος όρος του ξενοδοχείου και η εύρεση του αγαπημένου ξενοδοχείου για κάθε χρήστη.



**Εικόνα 6.11:** Αξιολόγηση ξενοδοχείου

Οι ιδιώτες μπορούν να αναζητούν ξενοδοχεία θέτοντας διάφορα κριτήρια. Το σύστημα, ως λειτουργία εξατομίκευσης, θα εμφανίσει μόνο τα ξενοδοχεία τα οποία πληρούν τα κριτήρια της αναζήτησης και τα οποία έχουν βαθμολογηθεί με βαθμό  $\geq 3$  από το συγκεκριμένο χρήστη που εκτελεί την αναζήτηση. Έτσι κάθε χρήστης βλέπει διαφορετικά αποτελέσματα, ανάλογα με τις αξιολογήσεις του.

Στη συνέχεια παρουσιάζουμε τον κώδικα που εκτελείται κατά την αναζήτηση ξενοδοχείου. Η συνάρτηση `search($title, $stars, $city, $county, $country, $rooms, $facilities, $userId)` δέχεται ως ορίσματα τον τίτλο, τα αστέρια, την πόλη, τον νομό, την χώρα, τους τύπους δωματίων και τις παροχές του ξενοδοχείου. Πιο αναλυτικά, εκτελείται ένα ερώτημα επιλογής που βρίσκει όλα τα ξενοδοχεία που έχουν τα κριτήρια που έχει δώσει ο χρήστης. Από το σύνολο αυτών των ξενοδοχείων εξαιρούνται όλα τα ξενοδοχεία (`h.id NOT IN`) τα οποία ο συγκεκριμένος χρήστης που κάνει την αναζήτηση έχει αξιολογήσει με βαθμό μικρότερο του 3 (`rat.grade<3`). Το σύστημα, δηλαδή για λογαριασμό του χρήστη εξαιρεί τα ξενοδοχεία που δεν του άρεσαν και έτσι δείχνει ότι γνωρίζει τους χρήστες του, τα ενδιαφέροντά τους και τις επιθυμίες τους.

```

public function search($title, $stars, $city, $county, $country, $rooms, $facilities,
$userId)
{
    $selectQuery = "SELECT DISTINCT h.id FROM hotels h";

    if (is_array($facilities) && !empty($facilities)) {
        $selectQuery .= " JOIN facilities f ON f.hotel_id = h.id";
    }

    if (is_array($rooms) && !empty($rooms)) {
        $selectQuery .= " JOIN rooms r ON r.hotel_id = h.id";
    }

    $selectQuery .= " WHERE 0=0 ";

    if (!is_null($title) && $title != "") {
        $selectQuery .= " AND h.title like '%" . $title . "%'";
    }

    if (!is_null($stars) && $stars != "") {
        $selectQuery .= " AND h.stars = ".$stars;
    }

    if (!is_null($city) && $city != "") {
        $selectQuery .= " AND h.city like '%" . $city . "%'";
    }

    if (!is_null($county) && $county != "") {
        $selectQuery .= " AND h.county like '%" . $county . "%'";
    }

    if (!is_null($country) && $country != "") {
        $selectQuery .= " AND h.country like '%" . $country . "%'";
    }

    if (is_array($facilities) && !empty($facilities)) {

```

```

$selectQuery .= " AND f.facility_type_id in (".implode(",", $facilities).)";
$selectQuery .= " AND f.value = '1'";
}

if (is_array($rooms) && !empty($rooms)) {
$selectQuery .= " AND r.room_type_id in (".implode(",", $rooms).)";
$selectQuery .= " AND r.value != '0' AND r.value != ''";
}

$selectQuery .= " AND h.id NOT IN ";

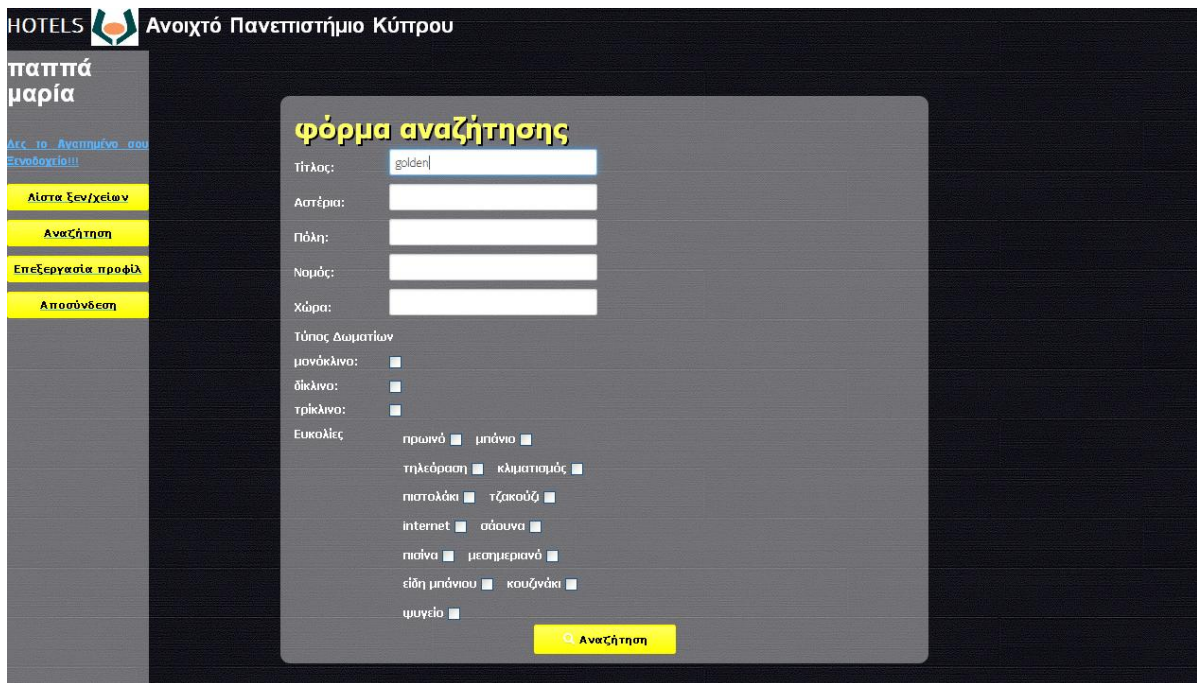
$selectQuery .= " (SELECT DISTINCT rat.hotel_id FROM ratings rat";
$selectQuery .= " WHERE 0=0 AND rat.account_id = ".$userId."";
$selectQuery .= " AND rat.grade<3 )";

$resultSet = $this->getDbTable()->getAdapter()->query($selectQuery);

$entries = array();
foreach ($resultSet as $row) {
$entry = new Application_Model_Hotel();
$this->find($row['id'], $entry);
$entries[] = $entry;
}
return $entries;
}

```

Τα ερωτήματα με στόχο την εξατομίκευση είναι πολύπλοκα και πολλές φορές δύσκολο να υλοποιηθούν. Συχνά απαιτούν συνδυασμό πολλών πινάκων και επιμέρους ερωτημάτων μέχρι να φτάσει στον χρήστη το τελικό αποτέλεσμα της εκτέλεσής τους.



Εικόνα 6.12: Αναζήτηση ξενοδοχείου

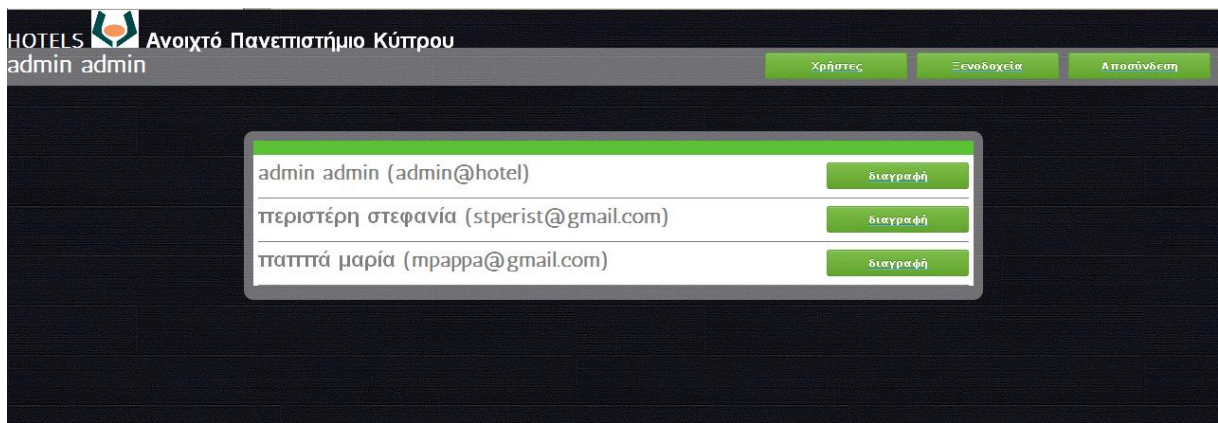
## 6.5 Σύνδεση Διαχειριστή στην εφαρμογή

Στον συνδεδεμένο στην εφαρμογή Διαχειριστή παρέχονται οι εξής επιλογές:

- Χρήστες (Διαχείριση χρηστών)
- Ξενοδοχεία (Διαχείριση ξενοδοχείων)
- Αποσύνδεση

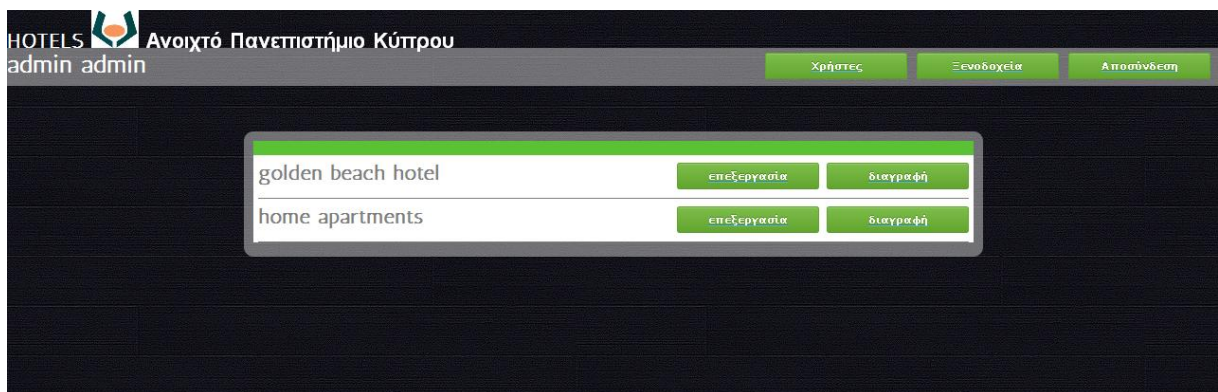
Στη σελίδα του διαχειριστή εμφανίζεται η λίστα των εγγεγραμμένων χρηστών, τους οποίους έχει το δικαίωμα να διαγράψει από την εφαρμογή.





**Εικόνα 6.13:** Διαχείριση χρηστών

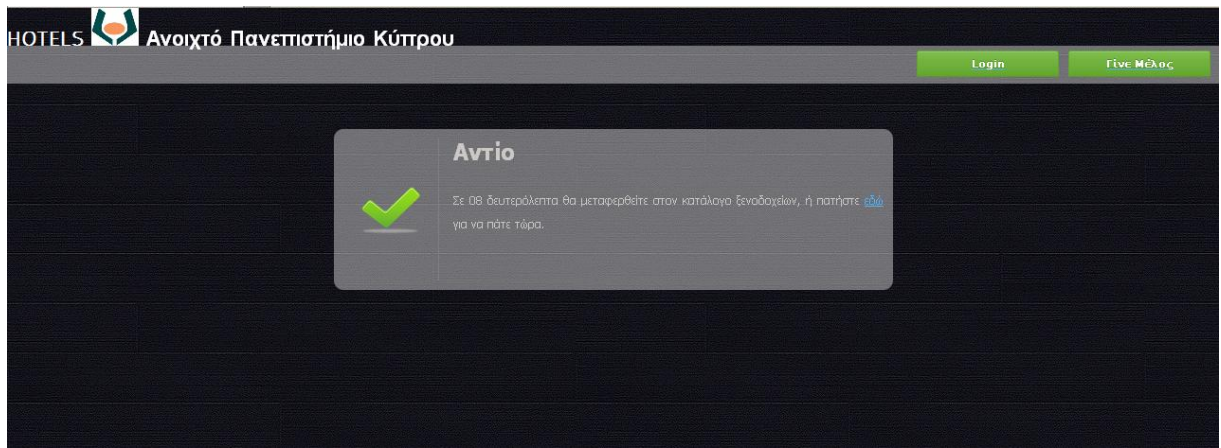
Στη σελίδα του διαχειριστή εμφανίζεται η λίστα των καταχωρημένων ξενοδοχείων, τα οποία έχει το δικαίωμα να ενημερώσει ή να διαγράψει από την εφαρμογή.



**Εικόνα 6.14:** Διαχείριση ξενοδοχείων

## 6.6 Αποσύνδεση εγγεγραμμένων χρηστών από την εφαρμογή

Οι εγγεγραμμένοι χρήστες όταν ολοκληρώσουν την περιήγησή τους στην εφαρμογή μπορούν να κάνουν αποσύνδεση και να μεταφερθούν στην αρχική σελίδα της εφαρμογής.



**Εικόνα 6.15:** Αποσύνδεση



# Κεφάλαιο 7

## Σύγκριση PHP και Zend Framework

Είναι δύσκολο να αναπτυχθεί μια ευχάριστη, αλληλοδραστική και επαναχρησιμοποιήσιμη ολοκληρωμένη διαδικτυακή εφαρμογή εκ του μηδενός με την PHP. Ωστόσο υπάρχουν αρκετά αξιόπιστα PHP Frameworks που στοχεύουν στη μείωση του χρόνου ανάπτυξης κάθε εφαρμογής. Αυτά τα Frameworks υποστηρίζονται από τεράστιες κοινότητες που βοηθούν τον προγραμματιστή στην αντιμετώπιση προβλημάτων. Αξίζει να εξεταστούν τα πλεονεκτήματα από τη χρήση ενός PHP Framework, όπως και οι περιορισμοί που θέτει στις δυνατότητες του τελικού συστήματος.

Στόχος της παρούσας μεταπτυχιακής διατριβής είναι να συγκριθούν ως προγραμματιστικές μέθοδοι η χρήση του Zend Framework και της απλής συγγραφής κώδικα σε PHP και να απαντηθούν τα εξής καίρια ερωτήματα:

1. Ένα PHP Framework βοηθάει στην ανάπτυξη μιας διαδικτυακής εφαρμογής ή περιπλέκει το έργο του προγραμματιστή;
2. Εξοικονομείται χρόνος και προσπάθεια;

3. Μπορεί να επαναχρησιμοποιηθεί μέρος του κώδικα της εφαρμογής;
4. Η εφαρμογή είναι πιο αποδοτική και σταθερή;
5. Είναι εύκολη η αλληλεπίδραση με τη βάση δεδομένων;
6. Υπάρχουν μειονεκτήματα από τη χρήση μιας τέτοιας προγραμματιστικής μεθόδου (π.χ. πόσο περιορίζει τις δυνατότητες του τελικού συστήματος);
7. Πώς μπορεί να γίνεται (με ενδεχόμενη αξιοποίηση του PHP Framework) συνεχής καταγραφή και αξιοποίηση του προφίλ του χρήστη για την εξατομίκευση των παρεχόμενων υπηρεσιών και της διεπαφής χρήστη;
8. Πόσο περιπλέκει την ανάπτυξη της εφαρμογής η ενσωμάτωση χαρακτηριστικών εξατομίκευσης;

## 7.1 Μελέτη Περίπτωσης - Login

Το σύστημα που αναπτύχθηκε με χρήση του Zend Framework είναι ένας δυναμικός διαδικτυακός τόπος παρουσίασης πληροφοριών που αφορούν σε τουριστικές μονάδες. Το σύστημα υποστηρίζει διαφορετικούς ρόλους χρηστών, τους οποίους και διακρίνει μέσα από τη διαδικασία εγγραφής και σύνδεσης με μοναδικό όνομα και κωδικό στην εφαρμογή. Η διαδικασία σύνδεσης στο σύστημα αναπτύχθηκε με χρήση Zend και με χρήση PHP ώστε να γίνει εφικτή η σύγκριση. Αξίζει να σημειωθεί ότι από πλευράς λειτουργικότητας και εμφάνισης της εφαρμογής προκύπτει όμοιο αποτέλεσμα. Η διάκριση των δύο μεθόδων παρατηρείται στη λογική ανάπτυξης κώδικα.

Στη συνέχεια, θα αντιπαραβάλουμε τον κώδικα ανάπτυξης της διαδικασίας Login:

1. Με χρήση της PHP :

Αρχείο login-form.php

```

2. <html>
3. <head>
4. <link href="css/login-box.css" rel="stylesheet" type="text/css" />
5. <link rel="stylesheet" type="text/css" href="css/style.css" />
6. <link rel="stylesheet" type="text/css" href="css/extras.css" />
7. <link rel="stylesheet" type="text/css" href="css/buttons.css" />
8. </head>
9. <body>

10. <?
11. if(@!$_POST['password']||!$_POST['email']){
12. ?>
13. <form action="" method="post" id="login-form">
14. <div style="width:485px; margin:0px auto;">
15. <div id="login-box">
16. <h2>Εισαγωγή</h2><br /><br />
17. <div id="login-box-name" style="margin-top:20px;">Email:</div>
18. <div id="login-box-field" style="margin-top:20px;">
19. <input name="email" class="form-login" value="" size="30" maxlength="2048" />
20. </div>
21. <div id="login-box-name">Password:</div>
22. <div id="login-box-field">
23. <input name="password" type="password" class="form-login" value="" size="30"
    maxlength="2048" />
24. </div>
25. <br style="clear:both;" />
26. <input type="image" src="images/login-btn.png" alt="Submit" width="103"
    height="42" style="margin-left:90px;" />
27. </div>
28. </div>
29. </form>

30. <?>
31. else
32. {
33. @$_password = $_POST[password];

```

```

34. @$email = $_POST["email"];

35. $db=mysql_connect("localhost", "root", "terry");
36. mysql_select_db("hotels", $db);

37. $query = sprintf("SELECT * FROM accounts WHERE password='%s' AND
    email='%s'",
38. mysql_real_escape_string($password), mysql_real_escape_string($email));
39. $result = mysql_query($query);
40. $num_rows = mysql_num_rows($result);

41. if (1 == $num_rows) {
42. $account = mysql_result($result,0,'account_type');
43. $name = mysql_result($result,0,'name');
44. switch ($account)
45. {
46. case 1:
47. $redirectUrl = "/hotel/index/";
48. $message = "<h1>Καλως ήρθες " . $name . " (Ιδιώτης).</h1>".
49. "<br /> Πατήστε <a href=\"\" . $redirectUrl . "\">εδώ</a> για να πάτε στον κατάλογο
    ξενοδοχείων.";
50. break;
51. case 2:
52. $redirectUrl = "/hotel/my-hotels/";
53. $message = "<h1>Καλως ήρθες " . $name . " (Ξενογόχος).</h1>".
54. "<br /> Πατήστε <a href=\"\" . $redirectUrl . "\">εδώ</a> για να πάτε στον κατάλογο
    των ξενοδοχείων σας.";
55. break;
56. case 3:
57. $redirectUrl = "/admin/index";
58. $message = "<h1>Καλως ήρθες " . $name . " (Διαχειριστής).</h1>".
59. "<br /> Πατήστε <a href=\"\" . $redirectUrl . "\">εδώ</a> για να πάτε στον κατάλογο
    ξενοδοχείων.";
60. break;
61. }

62. } else {

```

```
63. $redirectUrl = "login-form.php";
64. $messageClass = "error";
65. $message = "<h1>Λάθος στοιχεία λογαριασμού.</h1>".
66. "<br /> Πατήστε <a href=\"\" . $redirectUrl . "\">εδώ</a> για να πάτε στην φόρμα
    σύνδεσης.";
67. }

68. echo $message;
69. }?>
```

Με χρήση του Zend Framework:

Model: Αρχείο Account.php

```
1. <?php
2.
3. class Application_Model_Account
4. {
5.     protected $_name;
6.     protected $_surname;
7.     protected $_email;
8.     protected $_password;
9.     protected $_account_type;
10.    protected $_validated;
11.    protected $_id;
12.    protected $_color;
13.
14.    public function __construct(array $options = null)
15.    {
16.        if (is_array($options)) {
17.            $this->setOptions($options);
18.        }
19.    }
```

```
20.
21. public function __set($name, $value)
22. {
23.     $method = 'set' . $name;
24.     if (('mapper' == $name) || !method_exists($this, $method)) {
25.         throw new Exception('Invalid hotel property');
26.     }
27.     $this->$method($value);
28. }
29.
30. public function __get($name)
31. {
32.     $method = 'get' . $name;
33.     if (('mapper' == $name) || !method_exists($this, $method)) {
34.         throw new Exception('Invalid hotel property');
35.     }
36.     return $this->$method();
37. }
38.
39. public function getName(){return $this->_name;}
40.
41. public function setName($name)
42. {
43.     $this->_name = (string) $name;
44.     return $this;
45. }
46.
47. public function getSurname(){return $this->_surname;}
48.
49. public function setSurname($surname)
50. {
51.     $this->_surname = (string) $surname;
52.     return $this;
53. }
54.
55. public function getEmail(){return $this->_email;}
```

```
56.
57. public function setEmail($email)
58. {
59. $this->_email = (string) $email;
60. return $this;
61. }
62.
63. public function getPassword(){return $this->_password;}
64.
65. public function setPassword($password)
66. {
67. $this->_password = (string) $password;
68. return $this;
69. }
70.
71. public function getAccountType(){return $this->_account_type;}
72.
73. public function setAccountType($accountType)
74. {
75. $this->_account_type = (int) $accountType;
76. return $this;
77. }
78.
79. public function getValidated(){return $this->_validated;}
80.
81. public function isValidated()
82. {
83. return ($this->getValidated == 1)?true:false;
84. }
85.
86. public function setValidated($validated)
87. {
88. $this->_validated = (int) $validated;
89. return $this;
90. }
91.
```

```

92. public function setId($id)
93. {
94. $this->_id = (int) $id;
95. return $this;
96. }
97.
98. public function getId(){return $this->_id;}
99.
100. public function getColor(){return $this->_color;}
101.
102. public function setColor($color)
103. {
104. $this->_color = (string) $color;
105. return $this;
106. }
107. }

```

Model: Αρχείο AccountMapper.php

```

1. public function login($email, $password, Application_Model_Account $account)
2. {
3. $query = "SELECT id, name, surname, email, password, account_type, validated, color
   FROM accounts " .
4. "WHERE email = ? and password = ?";
5. $result = $this->getDbTable()->getAdapter()->query($query, array($email,
   $password));
6. if (1 == $result->rowCount()) {
7. $row = $result->fetch();
8. $account->setId($row['id']);
9. $account->setName($row['name']);
10. $account->setSurname($row['surname']);
11. $account->setEmail($row['email']);
12. $account->setPassword($row['password']);
13. $account->setAccountType($row['account_type']);
14. $account->setValidated($row['validated']);
15. $account->setColor($row['color']);

```



```
16. } else {
17. $account = NULL;
18. }
19. }
```

Viewer: Αρχείο login-form.phtml

```
1. <link href="/resources/css/login-box.css" rel="stylesheet" type="text/css" />
2. <form action="/user/login" method="post" id="login-form">
3. <div style="width:485px; margin:0px auto;">
4. <div id="login-box">
5. <h2>Εισαγωγή</h2>
6. <br /><br />
7. <div id="login-box-name" style="margin-top:20px;">Email:</div>
8. <div id="login-box-field" style="margin-top:20px;">
9. <input name="email" class="form-login" value="" size="30" maxlength="2048" />
10.</div>
11.<div id="login-box-name">Password:</div>
12.<div id="login-box-field">
13.<input name="password" type="password" class="form-login" value="" size="30"
    maxlength="2048" />
14.</div>
15.<br style="clear:both;" />
16.<a href="#" onclick="$('#login-form').submit()"></a>
17.<div style="margin-left:90px; padding-top:15px;"><a href="/index/register-
    form">Δεν είσαι μέλος; Γίνε τώρα!</a></div>
18.</div>
19.</div>
20.</form>
```

Controller: Αρχείο UserController.php

```
1. public function loginAction()
2. {
```

```

3. $email = $this->_request->getParam('email');
4. $password = $this->_request->getParam('password');
5.
6. $user = new Application_Model_Account();
7. $accountMapper = new Application_Model_AccountMapper();
8. $accountMapper->login($email, $password, &$user);
9. if ($user != null) {
10. $messageClass = "success";
11. switch ($user->getAccountType())
12. {
13. case 1:
14. $redirectUrl = "/hotel/index/";
15. $message = "<h1>Καλως ήρθες " . $user->getName() . ".</h1>";
16. "<br /> <span class=\"countdown\">Σε <span class=\"seconds\"></span>
    δευτερόλεπτα θα μεταφερθείτε στον κατάλογο ξενοδοχείων,</span> ";
17. "ή πατήστε <a href=\"\" . $redirectUrl . "\">εδώ</a> για να πάτε τώρα.";
18. break;
19.
20. case 2:
21. $redirectUrl = "/hotel/my-hotels/";
22. $message = "<h1>Καλως ήρθες " . $user->getName() . ".</h1>";
23. "<br /> <span class=\"countdown\">Σε <span class=\"seconds\"></span>
    δευτερόλεπτα θα μεταφερθείτε στον κατάλογο των ξενοδοχείων σας,</span> ";
24. "ή πατήστε <a href=\"\" . $redirectUrl . "\">εδώ</a> για να πάτε τώρα.";
25. break;
26.
27. case 3:
28. $redirectUrl = "/admin/index";
29. $message = "<h1>Καλως ήρθες " . $user->getName() . " (Διαχειριστής).</h1>";
30. "<br /> <span class=\"countdown\">Σε <span class=\"seconds\"></span>
    δευτερόλεπτα θα μεταφερθείτε στον κατάλογο ξενοδοχείων</span> ";
31. "ή πατήστε <a href=\"\" . $redirectUrl . "\">εδώ</a> για να πάτε τώρα.";
32. break;
33. }
34. $userNamespace = new Zend_Session_Namespace('user');
35. $userNamespace->user = $user;
36.

```

```
37. } else {
38. $redirectUrl = "/index/login-form/";
39. $messageClass = "error";
40. $message = "<h1>Λάθος στοιχεία λογαριασμού.</h1>".
41. "<br /> <span class=\"countdown\">Σε <span class=\"seconds\"></span>
    δευτερόλεπτα θα μεταφερθείτε στην φόρμα σύνδεσης</span> ".
42. "ή πατήστε <a href=\"\" . $redirectUrl . "\">εδώ</a> για να πάτε τώρα.";
43. }
44.
45. $this->_helper->layout->setLayout('index');
46. $this->view->messageClass = $messageClass;
47. $this->view->message = $message;
48. $this->view->redirectUrl = $redirectUrl;
49. }
```

Σύνδεση με τη Βάση Δεδομένων: Αρχείο application.ini

```
1. resources.db.adapter = "pdo_mysql"
2. resources.db.params.host = "localhost"
3. resources.db.params.username = "root"
4. resources.db.params.password = "terry"
5. resources.db.params.dbname = "hotels"
6. resources.db.params.charset = "utf8"
```

Οι παρατηρήσεις μας που προκύπτουν από τη χρήση των δύο παραπάνω διαφορετικών μεθόδων είναι οι εξής:

1. Με τη χρήση απλής PHP για τη μελέτη περίπτωσης Login, έχουμε ένα μόνο αρχείο (login-form.php) στο οποίο γίνεται η σύνδεση με τη Βάση Δεδομένων (34-35), παρουσιάζεται η φόρμα εισόδου στοιχείων στην εφαρμογή (12-28), εκτελείται το ερώτημα στη Βάση Δεδομένων (36-39), ελέγχεται και παρουσιάζεται η αντίστοιχη έξοδος μετά την εκτέλεση του ερωτήματος στο χρήστη (40-68).

2. Με τη χρήση του Zend Framework ο κώδικας οργανώνεται σε περισσότερα αρχεία και ακολουθεί το MVC πρότυπο. Συγκεκριμένα, για τη μελέτη περίπτωσης Login, στο Model παρατηρούνται τα αρχεία Account.php και AccountMapper.php. Στο αρχείο Account.php υπάρχει η αναπαράσταση του πίνακα account της Βάσης Δεδομένων και στο αρχείο AccountMapper.php υπάρχει η σύνταξη του κατάλληλου ερωτήματος στη Βάση Δεδομένων, που ελέγχει αν υπάρχει αποθηκευμένος ο αντίστοιχος λογαριασμός χρήστη. Στο View παρατηρείται το αρχείο login-form.phtml που έχει τον κώδικα για τη φόρμα σύνδεσης, την εμφάνιση της διεπαφής δηλαδή της εφαρμογής στο χρήστη. Στο Controller παρατηρείται το αρχείο UserController.php που έχει τον κώδικα για την κλήση του κατάλληλου ερωτήματος με τα αντίστοιχα ορίσματα και τον έλεγχο και την παρουσίαση της αντίστοιχης εξόδου μετά την εκτέλεση του ερωτήματος στο χρήστη. Τέλος, στο αρχείο application.ini γίνεται η σύνδεση με τη Βάση Δεδομένων.

## 7.2 Σύγκριση PHP και Zend Framework

Υπάρχουν πολλά πλεονεκτήματα που προκύπτουν από την αξιοποίηση του Zend Framework ως προγραμματιστικό εργαλείο στην ανάπτυξη δυναμικών διαδικτυακών εφαρμογών:

1. Μία εφαρμογή στο Zend Framework ακολουθεί συγκεκριμένη ιεραρχική δομή φακέλων για την οργάνωση των επιμέρους αρχείων, όπως φαίνεται στην εικόνα.



**Εικόνα 7.1:** Δομή φακέλων εφαρμογών αναπτυγμένων με χρήση του Zend Framework

Αυτό βοηθάει τον προγραμματιστή στην οργάνωση και επαναχρησιμοποίηση κώδικα, καθώς μπορεί πιο εύκολα να αναζητήσει τα επιμέρους κομμάτια κώδικα της εφαρμογής.

2. Ο κώδικας της εφαρμογής γίνεται πιο εύκολα αναγνώσιμος, συντηρήσιμος και επεκτάσιμος, καθώς οργανώνεται σε ξεχωριστά αρχεία που αφορούν τη σύνδεση με τη Βάση Δεδομένων, την αναπαράσταση των σχέσεων της Βάσης Δεδομένων, την αλληλεπίδραση του χρήστη με τη Βάση Δεδομένων (εισαγωγή, ενημέρωση, διαγραφή δεδομένων, εμφάνιση δεδομένων μετά από την εκτέλεση ερωτημάτων), την διεπαφή της εφαρμογής, τις λειτουργίες που εκτελούνται.
3. Η συγγραφή κώδικα είναι πιο εύκολη καθώς υπάρχουν πολλές έτοιμες κλάσεις, που διευκολύνουν και επιταχύνουν το έργο του προγραμματιστή. Με αυτόν τον τρόπο αποφεύγονται και λάθη, καθώς οι κλάσεις αυτές έχουν ελεγχθεί και υπάρχει αναλυτικό εγχειρίδιο με παραδείγματα για τη σωστή τους χρήση.

4. Η ανάπτυξη κώδικα ακολουθεί τις αρχές της αντικειμενοστρέφειας (classes, constructors, getters, setters, extends class, κ.λπ.).
5. Οργανωμένη κοινότητα υποστήριξης.

Υπάρχουν όμως και μειονεκτήματα τα οποία πρέπει να αναφερθούν:

1. Η χρήση του Zend Framework απαιτεί εξοικείωση με τον Object Oriented Προγραμματισμό και την απλή PHP.
2. Είναι δύσκολο και χρονοβόρο αρχικά για τον προγραμματιστή να κατανοήσει το MVC πρότυπο και απαιτείται αρκετός χρόνος μέχρι να γίνει παραγωγικός.
3. Αλλάζουν συνεχώς οι εκδόσεις του Zend Framework καθώς αυτό εξελίσσεται.
4. Διαφορετική φιλοσοφία ανάπτυξης κώδικα από την απλή PHP.
5. Προαπαιτείται η συμπερίληψη της Zend Βιβλιοθήκης.
6. Για μικρές εφαρμογές, περισσότερο περιπλέκει παρά απλουστεύει το έργο του προγραμματιστή.

### **7.3 Χρήση Framework ως προγραμματιστική μέθοδος**

Χρησιμοποιώντας ένα Framework ως προγραμματιστική μέθοδο παρατηρούμε ότι [10, 11]:

1. Διευκολύνεται η εργασία με δύσκολες τεχνολογίες.
2. Συνδέει ξεχωριστά αντικείμενα/συστατικά έτσι ώστε να προκύψει κάτι πιο χρήσιμο.
3. Αναγκάζει το άτομο ή την ομάδα που ασχολείται με τη συγγραφή κώδικα, να τον εφαρμόσει με τέτοιο τρόπο, έτσι ώστε να δημιουργήσει πιο συνεπή κώδικα, με λιγότερα λάθη και περισσότερο εύκαμπτες εφαρμογές.

4. Ο καθένας μπορεί εύκολα να ελέγξει και να διορθώσει τον κώδικα, ακόμη και αν δεν τον έχει συγγράψει ο ίδιος.
5. Αυξάνει την αξιοπιστία και μειώνει τον προγραμματιστικό χρόνο, καθώς γίνεται χρήση κώδικα ο οποίος έχει ήδη αναπτυχθεί, ελεγχτεί και χρησιμοποιηθεί από άλλους.

Όμως, πέρα από τα πλεονεκτήματα υπάρχουν και πολλοί περιορισμοί που προκύπτουν από τη χρήση τους:

- Πολλές φορές η απόδοση πέφτει καθώς χρησιμοποιείται ένας κοινός κώδικας. Αυτό συμβαίνει στις περιπτώσεις που ένα framework πρέπει να ελέγξει διάφορα σενάρια στα οποία χρησιμοποιείται για να καθορίσει μια πορεία δράσης. Η υποβάθμιση της απόδοσης, εν τούτοις, αντισταθμίζεται συχνά από τη βελτιωμένη ταχύτητα της ανάπτυξης και την ποιότητα της τελικής εφαρμογής.
- Πολλές φορές χρειάζεται ειδική γνώση για την αποτελεσματική και σωστή χρήση ενός framework. Επομένως τα συγκεκριμένα frameworks πολύ συχνά γίνονται πολυτιμότερα σε μεμονωμένους προγραμματιστές όταν χρησιμοποιούνται επανειλημμένα. Εάν σε κάθε εφαρμογή χρησιμοποιείται το ίδιο framework, τότε η εκμάθησή του αποτελεί ασήμαντο ζήτημα και η παραγωγικότητα αυξάνεται.
- Πολλές φορές η λειτουργία που πρέπει να προστεθεί σε ένα framework μπορεί να προκαλέσει περισσότερα προγραμματιστικά προβλήματα από το να αναπτυσσόταν όλη η εφαρμογή από την αρχή. Ορισμένα frameworks είναι τόσο άκαμπτα που η επιλογή τους για ένα ακατάλληλο πρόγραμμα μπορεί να είναι καταστρεπτική. Αυτό δεν είναι πρόβλημα των frameworks, αλλά ορισμένα frameworks είναι γενικότερης χρήστης και περισσότερο εύκαμπτα από άλλα. Έτσι, η επιλογή ενός framework πρέπει να γίνεται με ιδιαίτερη προσοχή.
- Λάθη και θέματα ασφάλειας σε ένα framework μπορούν να επηρεάσουν κάθε εφαρμογή που το χρησιμοποιεί. Γι' αυτό το λόγο πρέπει να ελέγχονται και να αναβαθμίζονται.





# Κεφάλαιο 8

## Επίλογος

Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής αναπτύχθηκε με χρήση του Zend Framework ένα ολοκληρωμένο δυναμικό διαδικτυακό σύστημα παρουσίασης πληροφοριών σχετικά με τουριστικές υποδομές και έγινε προσπάθεια να ενσωματωθούν στο σύστημα στοιχεία εξατομίκευσης τόσο στο περιεχόμενο όσο και στη δομή. Η ανάπτυξη της εφαρμογής απαιτούσε την εξοικείωση του προγραμματιστή τόσο με την γλώσσα προγραμματισμού PHP όσο και με την ανάπτυξη αντικειμενοστρεφών εφαρμογών.

Εντοπίστηκαν τα πλεονεκτήματα και οι περιορισμοί του Zend Framework μέσα από τη σύγκριση με την απλή συγγραφή κώδικα PHP και καταλήξαμε στο συμπέρασμα ότι ο προγραμματιστής είναι αυτός που θα κρίνει, ανάλογα με την εφαρμογή που θέλει να αναπτύξει, αν η υιοθέτηση ενός πλαισίου ως προγραμματιστική μέθοδος θα διευκολύνει ή θα δυσχεράνει το έργο του. Επίσης, η επιλογή του κατάλληλου πλαισίου για την κάθε εφαρμογή πρέπει να γίνεται αξιολογώντας τα χαρακτηριστικά που προσφέρει το κάθε πλαίσιο ή ακόμα και έπειτα από δοκιμή αυτών των εργαλείων.

Όσον αφορά την εξατομίκευση καταλήξαμε στο συμπέρασμα ότι είναι μια περίπλοκη διαδικασία, που απαιτεί τη συλλογή πολλών πληροφοριών σχετικά με τους χρήστες της εφαρμογής. Ακόμα πιο δύσκολο είναι να υλοποιηθεί η όλη διαδικασία σε πραγματικό χρόνο. Επίσης, από προγραμματιστικής άποψης είναι πολύπλοκο να υλοποιηθούν τα αντίστοιχα ερωτήματα στη βάση καθώς μπορεί να περιλαμβάνονται πολλοί πίνακες ή σύνθεση των αποτελεσμάτων που επιστρέφουν επιμέρους ερωτήματα.

Κάποιες προτάσεις για μελλοντικές επεκτάσεις του συστήματος είναι :

- Online κρατήσεις: Ο ιδιοκτήτης να ενημερώνεται για τις αιτήσεις κρατήσεων πελατών και οι πελάτες των ξενοδοχείων να μπορούν να υπολογίζουν αυτόματα το κόστος της διαμονής τους, να βλέπουν αν υπάρχουν διαθέσιμα δωμάτια για την περίοδο που τους ενδιαφέρει και να αποστέλλουν με το πάτημα ενός κουμπιού λεπτομερή αίτηση διαμονής ή/και online κράτηση στο ξενοδοχείο. Οι online κρατήσεις να γίνονται και με χρήση πιστωτικών καρτών.
- Προσθήκη χαρτών και γενικών πληροφοριών για την ευρύτερη περιοχή : Οι επισκέπτες της ιστοσελίδας να μπορούν να ενημερώνονται για την ευρύτερη περιοχή (αξιοθέατα, παραλίες, χιονοδρομικά κέντρα, κ.ά.) και η ιστοσελίδα να διαθέτει χάρτη της περιοχής.
- Σύσταση ξενοδοχείων ανάλογα με τα ενδιαφέροντα του χρήστη ή αξιολογώντας την περιήγησή του στην εφαρμογή ή κατόπιν σύγκρισης του προφίλ του με τα προφίλ των υπολοίπων χρηστών.

# Βιβλιογραφία

- [01] Chumki Basu, Haym Hirsh, William W. Cohen: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. AAAI/IAAI 1998: 714-720
- [02] Gilmore W. Jason, Beginning PHP and MySQL: From Novice to Professional, Fourth Edition, Apress, 2010
- [03] Rasmus Lerdorf, Kevin Tatroe, Programming PHP, O'Reilly Media, 2002
- [04] Ryan Somma, Application Frameworks Theory and Practice, CIS518
- [05] Murray Turoff, Personalization in user interface, CIS732, 2001
- [06] Vikram Vaswani, Zend Framework: A Beginner's Guide, McGrawHill, 2010
- [07] Emilio F Zegarra, Tony Efremenko, Using a recommendation engine to personalize your web application, 2011
- [08] Yannis E. Ioannidis, Georgia Koutrika: Personalized Systems: Models and Methods from an IR and DB Perspective. VLDB 2005: 1365
- [09] Αλέξανδρος Ν. Χατζηγεωργίου, Ανάπτυξη συστήματος λογισμικού βάσει της μεθοδολογίας ICONIX, 2008
- [10] DocForge, Framework, <http://docforge.com/wiki/Framework>, 2011
- [11] Marc Clifton, What Is A Framework, <http://www.codeproject.com/KB/architecture/WhatIsAFramework.aspx>, 2003

- [12] Nikko Bautista, 10 Compelling Reasons to Use Zend Framework, <http://net.tutsplus.com/tutorials/php/10-compelling-reasons-to-use-zend-framework/comment-page-1/#comments>, 2010
  
- [13] Noupe editorial team, Discussing PHP Frameworks: What, When, Why and Which?, <http://www.noupe.com/php/discussing-php-frameworks.html>, 2009
  
- [14] PHP, <http://php.net/>
  
- [15] PHP Frameworks, <http://www.phpframeworks.com>, 2007
  
- [16] Dynamic web page, [http://en.wikipedia.org/wiki/Dynamic\\_web\\_page](http://en.wikipedia.org/wiki/Dynamic_web_page), 2012
  
- [17] Zend, Zend Framework Tutorial, <http://framework.zend.com/>, 2006-2012

# Παράρτημα Α

## Παρουσίαση Αρχείων Εφαρμογής

Στη συνέχεια θα παρουσιαστούν συνοπτικά τα αρχεία της εφαρμογής..Επίσης στην εφαρμογή συμπεριλαμβάνονται αρχεία CSS, εικόνες και Javascript που συνθέτουν τη διεπαφή.

### **A.1 hotel\application\configs**

1. application.ini : ρυθμίσεις και σύνδεση με τη βάση δεδομένων

### **A.2 hotel\application\controllers**

1. AdminContoller.php : ενημέρωση ξενοδοχείων και χρηστών
2. ErrorContoller.php : διαχείριση errors εφαρμογής
3. HotelContoller.php : προβολή όλων των ξενοδοχείων, δημιουργία, αποθήκευση, αναζήτηση, ενημέρωση, προβολή ξενοδοχείων ανά ξενοδόχο, διαγραφή, αξιολόγηση.

4. IndexContoller.php : σύνδεση και εγγραφή μέλους
5. PhotoContoller.php : ανέβασμα φωτογραφιών, μετατροπή σε μικρογραφίες
6. UserContoller.php : σύνδεση χρήστη, εγγραφή, αποσύνδεση, ενημέρωση προφίλ, διαγραφή

### **A.3 hotel\application\layouts\scripts**

1. index.phtml : διάταξη σελίδων εφαρμογής

### **A.4 hotel\application\models**

1. Account.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται με τον λογαριασμό χρήστη
2. AccountMapper.php : αποθήκευση, εύρεση, ανάκτηση, διαγραφή λογαριασμού, υπολογισμός αγαπημένου ξενοδοχείου χρήστη
3. Facility.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται με τις παροχές
4. Facility Mapper.php : αποθήκευση, εύρεση, ανάκτηση, διαγραφή παροχών, εύρεση παροχών ανά ξενοδοχείο,
5. FacilityType.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται με τους τύπους παροχών
6. FacilityTypeMapper.php : αποθήκευση, εύρεση, ανάκτηση τύπων παροχών
7. Hotel.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται με το ξενοδοχείο
8. HotelMapper.php : αποθήκευση, εύρεση, ανάκτηση, διαγραφή ξενοδοχείου, υπολογισμός μέσου όρου βαθμολογίας, υπολογισμός αριθμού αξιολογήσεων, εύρεση φωτογραφιών

ξενοδοχείου, εύρεση παροχών, εύρεση αξιολογήσεων, εύρεση δωματίων, εισαγωγή φωτογραφίας, εισαγωγή αριθμού δωματίων, εύρεση όλων των ξενοδοχείων ανά ιδιοκτήτη, αναζήτηση ξενοδοχείου

9. HotelPhoto.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται με τις φωτογραφίες του ξενοδοχείου
10. HotelPhotoMapper.php : διαγραφή φωτογραφιών
11. Photo.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται τις φωτογραφίες όλων των ξενοδοχείων
12. Photo Mapper.php : αποθήκευση, εύρεση, ανάκτηση, διαγραφή φωτογραφιών
13. Rating.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται με την αξιολόγηση ξενοδοχείων
14. Rating Mapper.php : αποθήκευση, εύρεση, ανάκτηση, διαγραφή αξιολογήσεων, υπολογισμός μέσου όρου βαθμολογίας ανά ξενοδοχείο, υπολογισμός αριθμού αξιολογήσεων, εύρεση ξενοδοχείου με μέγιστο βαθμό ανά χρήστη, εύρεση αξιολογήσεων ανά ξενοδοχείο
15. Room.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται με τα δωμάτια
16. Room Mapper.php : αποθήκευση, εύρεση, ανάκτηση, διαγραφή δωματίων, εύρεση αριθμού δωματίων ξενοδοχείου ανά τύπο δωματίου
17. RoomType.php : μεταβλητές και getters και setters μεταβλητών που σχετίζονται με τους τύπους δωματίων
18. RoomTypeMapper.php : αποθήκευση, εύρεση, ανάκτηση, διαγραφή τύπων δωματίων

## **A.5 hotel\application\models\DbTable**

1. Account.php : σύνδεση με πίνακα accounts της Β.Δ.
2. Facility.php: σύνδεση με πίνακα facilities της Β.Δ.
3. FacilityTypes.php : σύνδεση με πίνακα facility\_types της Β.Δ.
4. Hotel.php: σύνδεση με πίνακα hotels της Β.Δ.
5. HotelPhoto.php: σύνδεση με πίνακα hotel\_photos της Β.Δ.
6. Photo.php: σύνδεση με πίνακα photos της Β.Δ.
7. Rating.php: σύνδεση με πίνακα ratings της Β.Δ.
8. Room.php: σύνδεση με πίνακα rooms της Β.Δ.
9. RoomTypes.php: σύνδεση με πίνακα room\_types της Β.Δ.

## **A.6 hotel\application\views\scripts\admin**

1. Edit-hotels.phtml : σελίδα διαχείρισης ξενοδοχείων
2. Edit-users.phtml: σελίδα διαχείρισης χρηστών

## **A.7 hotel\application\views\scripts\error**

1. Error.phtml : σελίδα παρουσίασης σφαλμάτων εφαρμογής

## **A.8 hotel\application\views\scripts\hotel**

1. Delete.phtml : μήνυμα διαγραφής ξενοδοχείου



2. Edit.phtml : σελίδα ενημέρωσης ξενοδοχείων
3. Index.phtml : σελίδα παρουσίασης λίστας όλων των ξενοδοχείων
4. My-hotels.phtml : σελίδα παρουσίασης λίστας όλων των ξενοδοχείων ανά ξενοδόχο
5. New.phtml : σελίδα καταχώρησης νέου ξενοδοχείου
6. Save.phtml : μήνυμα δημιουργίας ξενοδοχείου
7. Search.phtml : σελίδα αναζήτησης ξενοδοχείων
8. Update.phtml : μήνυμα ενημέρωσης ξενοδοχείου
9. View.phtml : προβολή σελίδας πληροφοριών ξενοδοχείου

## **A.9 hotel\application\views\scripts\index**

1. Login-form.phtml : σελίδα σύνδεσης χρήστη
2. Register-form.phtml : σελίδα εγγραφής νέου χρήστη

## **A.10 hotel\application\views\scripts\user**

1. Delete.phtml : μήνυμα διαγραφής χρήστη
2. Edit-form.phtml : σελίδα ενημέρωσης στοιχείων λογαριασμού χρήστη
3. Head.phtml : σελίδα που ορίζει το μενού επιλογών ανά ιδιότητα χρήστη
4. Login.phtml : μήνυμα σύνδεσης χρήστη
5. Logout.phtml : μήνυμα αποσύνδεσης χρήστη
6. Register.phtml : μήνυμα εγγραφής χρήστη

7. Update.phtml : μήνυμα ενημέρωσης χρήστη

## **A.11 hotel\application**

1. Bootstrap.php : αρχείο που περιλαμβάνει κώδικα που εκτελείται κατά την εκκίνηση της εφαρμογής