



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΗΠΕΙΡΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΓΕΩΓΡΑΦΙΚΟ ΠΛΗΡΟΦΟΡΙΑΚΟ
ΣΥΣΤΗΜΑΓΙΑ ΤΗΝ ΠΑΡΟΥΣΙΑΣΗ
ΤΟΥΡΙΣΤΙΚΩΝ ΣΗΜΕΙΩΝ
ΕΝΔΙΑΦΕΡΟΝΤΟΣ ΣΤΗΝ
ΠΕΡΙΦΕΡΕΙΑ ΤΗΣ ΗΠΕΙΡΟΥ

ΒΑΣΙΛΗΣ ΚΟΛΛΩΣΗΣ

29/1/2015

ΠΕΡΙΛΗΨΗ

Η πτυχιακή αυτή εργασία αναλύει την ανάπτυξη μίας διαδικτυακής εφαρμογής για την προβολή σημείων τουριστικού ενδιαφέροντος στην περιφέρεια της Ηπείρου.

Η ανάδειξη της Ηπείρου σε τουριστικό προορισμό αποτελεί σκοπό του Ελληνικού κράτους, με την υποστήριξη της Ευρωπαϊκής Ένωσης. Είναι μια περιοχή με πλούσιο φυσικό κάλος, ενδεδειγμένη για αγροτουρισμό και αθλητικό τουρισμό, τόπους αρχαιολογικού ενδιαφέροντος.

Η εν προκειμένω εφαρμογή για τον χειρισμό της απαιτεί την ύπαρξη διαδικτύου και εμφανίζει σημεία τουριστικού ενδιαφέροντος σε χάρτη της Google. Για κάθε σημείο μπορούν να υπάρχουν πληροφορίες, εικόνες, βίντεο από το Youtube της Google και σχόλια των χρηστών. Η περιήγηση στην εφαρμογή μπορεί να γίνει είτε από απλούς επισκέπτες είτε από συνδεδεμένο χρήστη.

Κατά την ανάλυση της εφαρμογής χρησιμοποιήθηκαν UML διαγράμματα. Στην συνέχεια για την δημιουργία της επιλέχθηκαν ως πλατφόρμα η Ruby on Rails και ως εξυπηρετητή ιστού ο Apache2. Τέλος χρησιμοποιήθηκαν ως μηχανή βάσης δεδομένων η MariaDB, ως λειτουργικό σύστημα το Ubuntu 14.04 LTS και για την φιλοξενία η υπηρεσία AWS της Amazon.

Πίνακας περιεχομένων

Εισαγωγή	7
ΤΟΥΡΙΣΜΟΣ.....	9
ΣΥΣΤΗΜΑ ΓΕΩΓΡΑΦΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ.....	11
Απαιτήσεις χρηστών.....	14
ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ	18
UML.....	18
Sequence Diagram	20
Activity Diagram	22
Υπάρχουσα κατάσταση. Σύγκριση πλατφόρμων με απαιτήσεις χρηστών. Επιλογή πλατφόρμας	24
ΛΕΙΤΟΥΡΓΙΚΟ	24
ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	26
ΠΛΑΤΦΟΡΜΕΣ - ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	27
Εξυπηρετητής Ιστού (Web Server).....	33
Περιγραφή χρησιμοποιούμενης μεθοδολογίας προκειμένου να δοθεί λύση.....	34
AWS (Amazon Web Services)	34
Ubuntu	35
RVM	36
MariaDB	37
Apache	37
PASSENGER.....	38
Παρουσίαση της υλοποίησης.	39
MVC	39
Controller.....	40
Model	43
Views	46
Παρατηρήσεις κατά τη διάρκεια της ανάπτυξης του συστήματος.....	60
Προβλήματα.....	60
Πιθανές μελλοντικές προσθήκες	60
Βιβλιογραφία	62

ΠΙΝΑΚΕΣ

Πίνακας 1 ΣΥΓΚΡΙΣΗ ΛΕΙΤΟΥΡΓΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	25
Πίνακας 2 ΣΥΓΚΡΙΣΗ ΣΧΕΣΙΑΚΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ	26
Πίνακας 3 ΣΥΓΚΡΙΣΗ ΠΛΑΤΦΟΡΜΩΝ	27
Πίνακας 4 ΣΥΓΚΡΙΣΗ ΕΞΥΠΗΡΕΤΗΤΩΝ ΙΣΤΟΥ.....	33

ΕΙΚΟΝΕΣ

Εικόνα 1 ΠΕΡΙΦΕΡΕΙΑ ΗΠΕΙΡΟΥ	7
Εικόνα 2 Σύστημα Γεωγραφικών Πληροφοριών (3).....	11
Εικόνα 3 ΑΠΕΙΚΟΝΙΣΗ ΣΗΜΕΙΩΝ ΣΕ ΧΑΡΤΗ (5).....	13
Εικόνα 4 ΤΟ ΜΟΝΤΕΛΟ MVC	40
Εικόνα 5 ΟΙ ΕΛΕΓΚΤΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΕΙ ΤΟ ΣΥΣΤΗΜΑ ΜΑΣ	42
Εικόνα 6 ΤΟ ΜΟΝΤΕΛΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΑΣ.....	45
Εικόνα 7 ΟΙ ΔΙΑΘΕΣΙΜΕΣ ΠΡΟΒΟΛΕΣ	46
Εικόνα 8 ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΕΦΑΡΜΟΓΗΣ.....	48
Εικόνα 9 ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΕΦΑΡΜΟΓΗΣ ΑΠΟ ΣΥΣΚΕΥΗ ΑΠΟ ANDROID	49
Εικόνα 10 Η ΣΕΛΙΔΑ ΣΧΕΤΙΚΑ	49
Εικόνα 11 Η ΣΕΛΙΔΑ ΕΠΙΚΟΙΝΩΝΙΑΣ	50
Εικόνα 12 Η ΣΕΛΙΔΑ ΜΕ ΤΟΝ ΧΑΡΤΗ.....	51
Εικόνα 13 Η ΣΕΛΙΔΑ ΜΕ ΤΟΝ ΧΑΡΤΗ ΜΕ ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ ΤΟ ΣΗΜΕΙΟ	52
Εικόνα 14 Η ΣΕΛΙΔΑ ΜΕ ΤΗΣ ΠΛΗΡΟΦΟΡΙΕΣ ΤΟΥ ΣΗΜΕΙΟΥ	53
Εικόνα 15 Η ΠΡΟΒΟΛΗ ΤΩΝ ΕΙΚΟΝΩΝ	54
Εικόνα 16 Η ΠΡΟΒΟΛΗ ΤΩΝ ΒΙΝΤΕΟ ΤΟΥ YOUTUBE	54
Εικόνα 17 ΤΑ ΣΧΟΛΙΑ ΤΟΥ ΣΗΜΕΙΟΥ.....	55
Εικόνα 18 Η ΣΕΛΙΔΑ ΕΝΗΜΕΡΩΣΗΣ ΣΗΜΕΙΟΥ	56
Εικόνα 19 ΤΟ ΜΕΝΟΥ ΕΠΕΒΕΒΑΙΩΣΗΣ ΔΙΑΓΡΑΦΗΣ	57
Εικόνα 20 Η ΣΕΛΙΔΑ ΤΗΣ ΠΡΟΣΘΗΚΗΣ ΣΗΜΕΙΟΥ	58
Εικόνα 21 Η ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΗΣ ΧΡΗΣΤΗ	58
Εικόνα 22 Η ΣΕΛΙΔΑ ΔΗΜΙΟΥΡΓΙΑΣ ΧΡΗΣΤΗ.....	59

ΣΧΗΜΑΤΑ

Σχήμα 1 UML USE CASE ΕΠΙΣΚΕΠΤΗ	14
Σχήμα 2 UML USE CASE ΣΥΝΔΕΔΕΜΕΝΟΥ ΧΡΗΣΤΗ 1.....	15
Σχήμα 3 UML USECASE ΣΥΝΔΕΔΕΜΕΝΟΥ ΧΡΗΣΤΗ 2.....	16
Σχήμα 4 UML USE CASE ΣΥΝΔΕΔΕΜΕΝΟΥ ΧΡΗΣΤΗ 3.....	17
Σχήμα 5 ΟΨΕΙΣ ΤΗΣ UML	19
Σχήμα 6 SEQUENCE DIAGRAM ΧΡΗΣΤΗ	20
Σχήμα 7 SEQUENCE DIAGRAM ΔΙΑΧΕΙΡΙΣΤΗ	21
Σχήμα 8 ACTIVITY DIAGRAM ΠΡΟΣΘΗΚΗ ΣΗΜΕΙΟΥ.....	22

Εισαγωγή

Στην εργασία που ακολουθεί θα παρατεθεί η διαδικασία δημιουργίας της πρωτότυπης εφαρμογής “Τουριστικές περιοχές της Ηπείρου”, για την προβολή σε χάρτη σημείων τουριστικού ενδιαφέροντος στην περιοχή της Ηπείρου.



Εικόνα 1 ΠΕΡΙΦΕΡΕΙΑ ΗΠΕΙΡΟΥ

Η διαδικτυακή αυτή εφαρμογή μπορεί κάλλιστα να χρησιμοποιηθεί τόσο μέσω ηλεκτρονικού υπολογιστή, όσο και μέσω tablet ή smartphone.

Βασικό κίνητρο στην προσπάθειά μας αυτή ήταν η μη ύπαρξη προηγούμενης ανάλογης εφαρμογής εξειδικευμένης για την Ήπειρο. Επιπλέον είναι αδιαμφισβήτητο γεγονός το ότι η περιοχή έχει να παρουσιάσει μεγάλη ποικιλία αξιόλογων φυσικών οικοσυστημάτων και πολιτιστικών μνημείων, ενώ ταυτόχρονα ο τουρισμός έχει αναδειχθεί σε κυρίαρχο κλάδο της τοπικής οικονομίας – ιδιαίτερα την τελευταία δεκαετία. Επομένως - τόσο για την προσέλκυση, όσο και για την εξυπηρέτηση τουριστών - είναι υπαρκτή η ανάγκη, να έχει γρήγορα και εύκολα στα χέρια του ο καθένας όλες τις πληροφορίες που αναζητεί. Η τουριστική εμπειρία αποκτά μια άλλη τεχνολογική διάσταση.

Άλλωστε “η πληροφορία αποτελεί ζωτικό και κρίσιμο παράγοντα της εύρυθμης λειτουργίας της τουριστικής βιομηχανίας (information is the lifeblood of tourism) και έτσι, εφαρμογές των τεχνολογιών επικοινωνίας και πληροφοριών συμβάλουν αποφασιστικά στην αποτελεσματικότερη και αποδοτικότερη λειτουργία της τουριστικής βιομηχανίας τόσο σε μακροοικονομικό όσο και σε μικροοικονομικό επίπεδο.” (1) Ταυτόχρονα οι τουρίστες, θα αναζητούν πάντα εκ των προτέρων δεδομένα για τον προορισμό τους, δηλαδή πριν την άφιξή τους στο εκάστοτε επιθυμητό σημείο.

Με άλλα λόγια, μεγάλο τμήμα της ζητούμενης πληροφόρησης θα πρέπει να γίνεται με εύκολο και συγκεντρωτικό τρόπο σε απομακρυσμένο σημείο – μέσω του διαδικτύου. Συνδυάζοντας ως στόχους την ευελιξία, με τον απλό χειρισμό και την πλούσια παράθεση λεπτομερειών, κινηθήκαμε κατά τη διάρκεια της προσπάθειάς μας.

Ας αναφερθούμε τώρα σε μια βασική παράμετρο χρήσης της εφαρμογής μας, δηλαδή στα πλεονεκτήματα που έχει αυτή να προσδώσει στους μελλοντικούς επισκέπτες των σημείων τουριστικού ενδιαφέροντος. Εν γένει η Ήπειρος όντας σχετικά δυσπρόσιτη περιοχή μέχρι τα τέλη του περασμένου αιώνα, δεν έχει αναδείξει πλήρως τον πλούτο της στα μάτια των επισκεπτών. Με την δυνατότητα προβολής οπτικοακουστικών πληροφοριών κάθε σημείου μέσω του διαδικτύου, τόσο οι παλαιότεροι επισκέπτες όσο και οι νέοι, θα μπορούν: Να ενημερώνονται για τους διαχρονικούς πόλους έλξης του τόπου μας, για σημαντικές δραστηριότητες και εκθέματα, αλλά και να πλοηγούνται σε σημεία πολλών και διαφορετικών επιλογών.

Ο εμπλουτισμός, ίσως σε εμπορικό επίπεδο, της εφαρμογής μας με τα αναρίθμητα σημεία που μπορούν να προστεθούν θα την καταστήσει πολύτιμο εργαλείο στην επικοινωνία τουριστών – πρακτορείων. Μάλιστα θα οδηγούσε σε σαφή μείωση του κόστους διαφήμισης των επιχειρήσεων παροχής τουριστικών υπηρεσιών.

ΤΟΥΡΙΣΜΟΣ

Η χρήση της τεχνολογίας στον τουρισμό προσφέρει τα εξής οφέλη για τους επισκέπτες:

- Ο επισκέπτης μπορεί να συλλέξει τις πληροφορίες που θέλει για τον τόπο που τον ενδιαφέρει, από το διαδίκτυο. Γεγονός που προσφέρει οικονομία αφού αποφεύγει τους τουριστικούς πράκτορες. Ενώ ταυτόχρονα μπορεί να κάνει την έρευνά του με ευκολία, με ταχύτητα και κυρίως όποτε θέλει.
- Η χρήση πολυμέσων διευκολύνει την επιλογή, μιας και δίνει κυριολεκτικά την εικόνα του κάθε σημείου.
- Μπορεί να καθορίσει το δικό του σύνολο πιθανών προορισμών, έχοντας κάνει αναζήτηση στο διαδίκτυο βάσει των ενδιαφερόντων του. Μάλιστα είναι ενδέχεται να εντοπίσει σημεία, τα οποία δεν θα μάθαινε με διαφορετικό τρόπο πως υπήρχαν.
- Ανάλογα με τις προσωπικές ανάγκες του, ο κάθε επισκέπτης, μπορεί να επικοινωνήσει προσωπικά με τους παρόχους και να διασφαλίσει σίγουρη και πλήρη εξυπηρέτηση.
- Με την κοινοποίηση σχολίων, η άποψη του πελάτη πλέον είναι εμφανής και στους υπόλοιπους. Έτσι μπορεί να ασκηθεί πίεση στις επιχειρήσεις, προς την κατεύθυνση βελτίωση των υπηρεσιών τους.

Από την σκοπιά των επιχειρήσεων, τα πλεονεκτήματα που μπορεί να προσφέρει η τεχνολογία είναι:

- Η προσωπική επαφή με τον πελάτη κάνει πιο πιθανή την επιλογή. Επίσης ο προορισμός γίνεται ακόμη πιο δελεαστικός καθώς ο χρόνος εξυπηρέτησης μειώνεται.
- Η προβολή της επιχείρησης σε παγκόσμια κλίμακα προσφέρει μεγαλύτερο πελατολόγιο. Ακόμη και ο πιο απομακρυσμένος πελάτης μπορεί να ενημερωθεί για τις υπηρεσίες που διατίθενται.
- Η εύρεση προμηθευτών για μια επιχείρηση είναι σημαντικός παράγοντας παροχής σωστών υπηρεσιών. Με το διαδίκτυο μπορούν και οι επιχειρήσεις να εκμεταλλευτούν στο βέλτιστο τις προσφορές παγκοσμίως.
- Για τις μικρότερες επιχειρήσεις που δεν θα μπορούσαν να κάνουν μια συμφωνία με κάποιο τουριστικό πρακτορείο: Είναι δυνατόν να αναπτύξουν καλοστημένα την προβολή τους στο διαδίκτυο. Γεγονός που τους παρέχει καλύτερη διαφημιστική προβολή, και μάλιστα με μικρότερο κόστος.

Συγκεκριμένα, για την περιοχή της Ηπείρου, το επιχειρησιακό πρόγραμμα της περιφέρειας Ηπείρου (Ε.Π. 2014 - 2020) (1) περιλαμβάνει τον τουρισμό στους βασικούς στόχους του. Κάποιες από τις προτάσεις είναι :

- Στον τομέα των μεταφορών προτεραιότητα αποτελεί η διασύνδεση των αστικών περιοχών με περιοχές ιδιαίτερου κάλους και τουριστικού ενδιαφέροντος. Επίσης η αναβάθμιση των λιμανιών για την καλύτερη προσπέλαση της ευρύτερης περιοχής.

- Υπάρχει σχεδιασμός για παρέμβαση υπέρ των επιχειρήσεων τουριστικού ενδιαφέροντος για την ανάπτυξη τους και την απασχόληση παραπάνω προσωπικού.
- Η ανάπτυξη ειδικών μορφών τουρισμού, όπως οικο-τουρισμού, ποδηλατικού τουρισμού, θρησκευτικού τουρισμού κ.α.

ΣΥΣΤΗΜΑ ΓΕΩΓΡΑΦΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ

Με τον όρο Σύστημα Γεωγραφικών Πληροφοριών (Σ.Γ.Π.), επίσης G.I.S (Geographic Information Systems), εννοούμε το πληροφοριακό σύστημα που σαν σκοπό έχει την προβολή γεωγραφικών πληροφοριών για την λήψη αποφάσεων. Ο ορισμός περιλαμβάνει και το ανθρώπινο δυναμικό που ασχολείται με το σύστημα, εκτός από το υλικό και το λογισμικό.



Εικόνα 2 Σύστημα Γεωγραφικών Πληροφοριών (3)

Το Σ.Γ.Π. μπορεί να λάβει, να αποθηκεύσει, να επεξεργαστεί και να προβάλει γεωγραφικά δεδομένα. Με την χρήση χάρτη θα προβάλλει διαδραστικές πληροφορίες (2). Σκοπός του είναι να προβάλει πληροφορίες που δεν έχουν γεωγραφικό προσανατολισμό αλλά σχέση με κάποιο γεωγραφικό χώρο.

Υπάρχουν δύο ειδών μοντέλα για την σύνδεση των διαδραστικών πληροφοριών, με εκείνες που συνδέονται με τον γεωγραφικό τόπο:

1. Σχεσιακό μοντέλο. Ξεχωριστές μεταξύ τους πληροφορίες που συνδέονται μέσω κοινών τιμών.
2. Αντικειμενοστραφές μοντέλο. Όπου και οι δύο πληροφορίες περιέχονται στο ίδιο αντικείμενο. Το οποίο είναι και το επικρατέστερο λόγω των περισσότερων δυνατοτήτων που έχει.

Ένα Σύστημα Γεωγραφικών Πληροφοριών αποτελείται (3) από τα παρακάτω στοιχεία:

- Λογισμικό
- Υλικό
- Ανθρώπινο δυναμικό
- Δεδομένα
- Διαδίκτυο

Η δημιουργία και η λειτουργία ενός τέτοιου συστήματος συνήθως ακολουθεί κάποια συγκεκριμένα βήματα. Αρχικά βρίσκουμε και φτιάχνουμε μια βάση δεδομένων για την αποθήκευση των δεδομένων μας. Στη συνέχεια βρίσκουμε τον τρόπο που θα προβάλλουμε τα δεδομένα αυτά. Έπειτα επιλέγουμε το περιβάλλον προβολής.

Σε αυτό το σημείο ακολουθεί το βήμα κατά το οποίο προσθέτουμε τις πληροφορίες στην βάση μας, συνδυασμένες με το γεωγραφικό χώρο όπου ανήκουν. Το επόμενο βήμα είναι η προβολή της πληροφορίας, η οποία θα συμβάλει στην διαμόρφωση γνώμης εκείνου που αναζητά τις πληροφορίες.



Εικόνα 3 ΑΠΕΙΚΟΝΙΣΗ ΣΗΜΕΙΩΝ ΣΕ ΧΑΡΤΗ (5)

Την πληροφορία μπορούμε να την προβάλλουμε μόνο με κάποιον από αυτούς του τρόπους:

- Σημείο
- Γραμμή
- Πολύγωνο

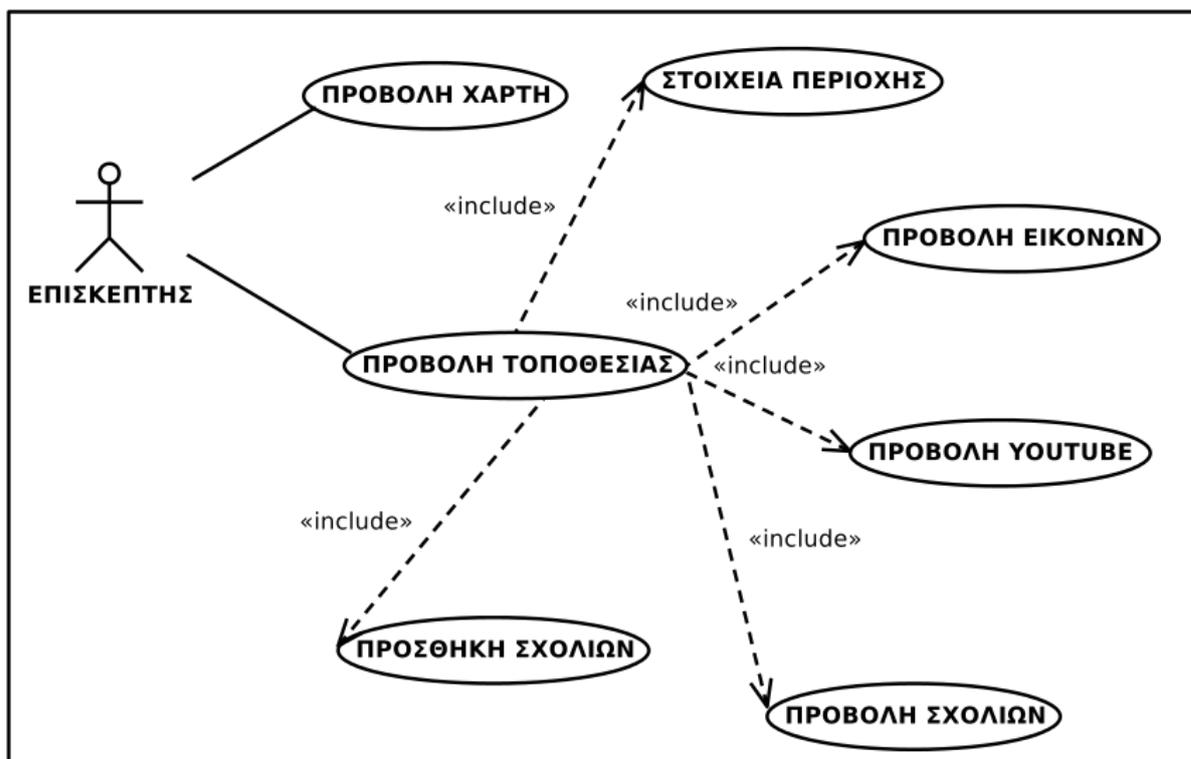
Εργαλεία που χρησιμοποιούνται σε τέτοια συστήματα είναι (4):

- Εργαλεία επιλογής.
- Εργαλεία σημείων και γραμμών.
- Στατιστικά εργαλεία.
- Τοπογραφικά εργαλεία.
- Εργαλεία χαρτών.

Απαιτήσεις χρηστών

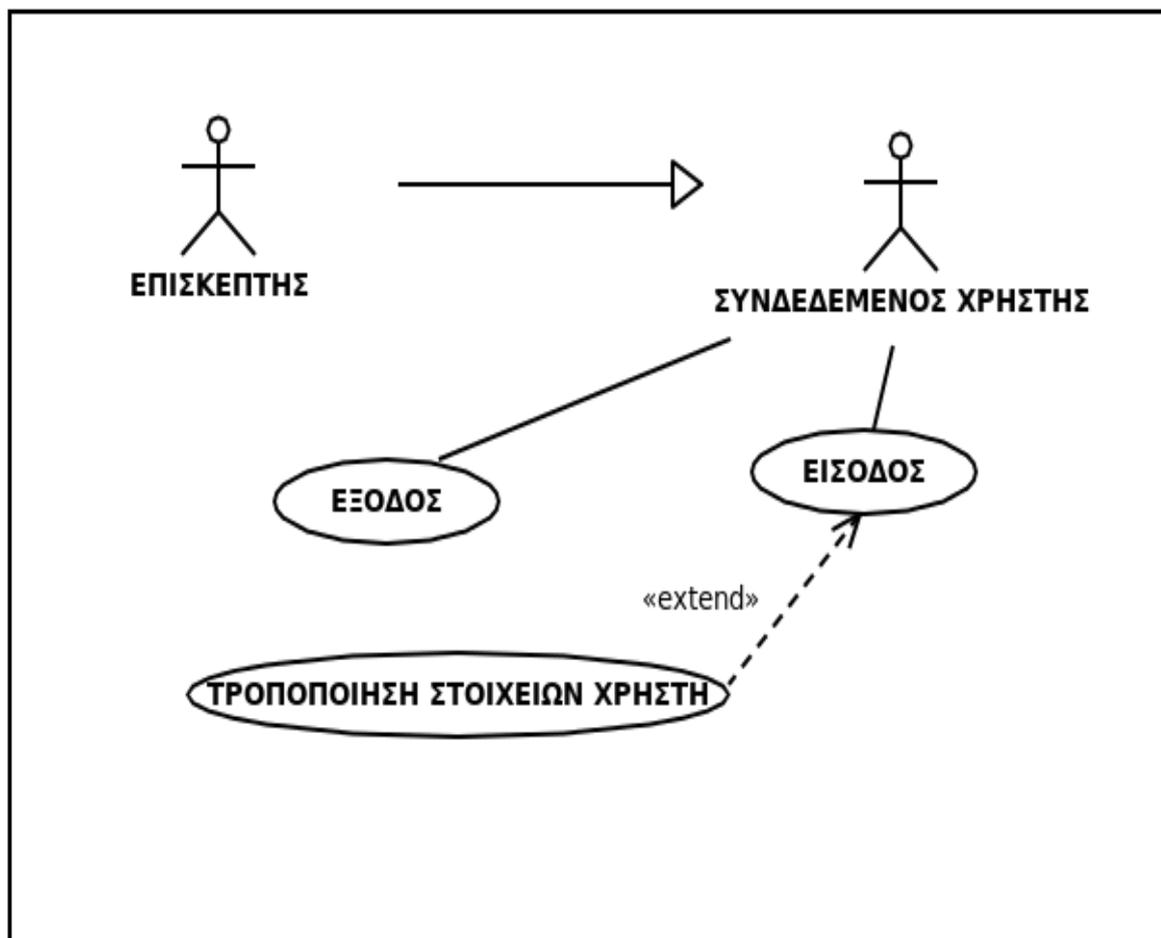
Η βασική λειτουργία του συστήματος προς εξέταση θα είναι, ως προς τις απαιτήσεις του: Να παρέχει πλήρεις δυνατότητες προβολής και αξιολόγησης του κάθε σημείου ενδιαφέροντος στους απλούς χρήστες/επισκέπτες και από την άλλη μόνο ο συνδεδεμένος χρήστης να μπορεί να επεξεργαστεί και να δημιουργεί νέες εγγραφές.

Όπως φαίνεται και στο Σχήμα 1 οι απλοί χρήστες/επισκέπτες θα έχουν την δυνατότητα να βλέπουν σε έναν χάρτη, που εστιάζει στην περιφέρεια της Ηπείρου, σημεία τουριστικού ενδιαφέροντος διαφόρων κατηγοριών. Επίσης θα μπορεί να βλέπει πληροφορίες για κάθε ένα από αυτά τα σημεία. Αυτές οι πληροφορίες θα αποτελούνται από: Στοιχεία της περιοχής, προβολή εικόνων του σημείου ενδιαφέροντος, προβολή βίντεο της υπηρεσίας Youtube της Google και η δυνατότητα χειρισμού σχολίων από τους χρήστες/επισκέπτες. Τα σχόλια θα μπορούν να προβάλλονται και να αναρτηθούν.



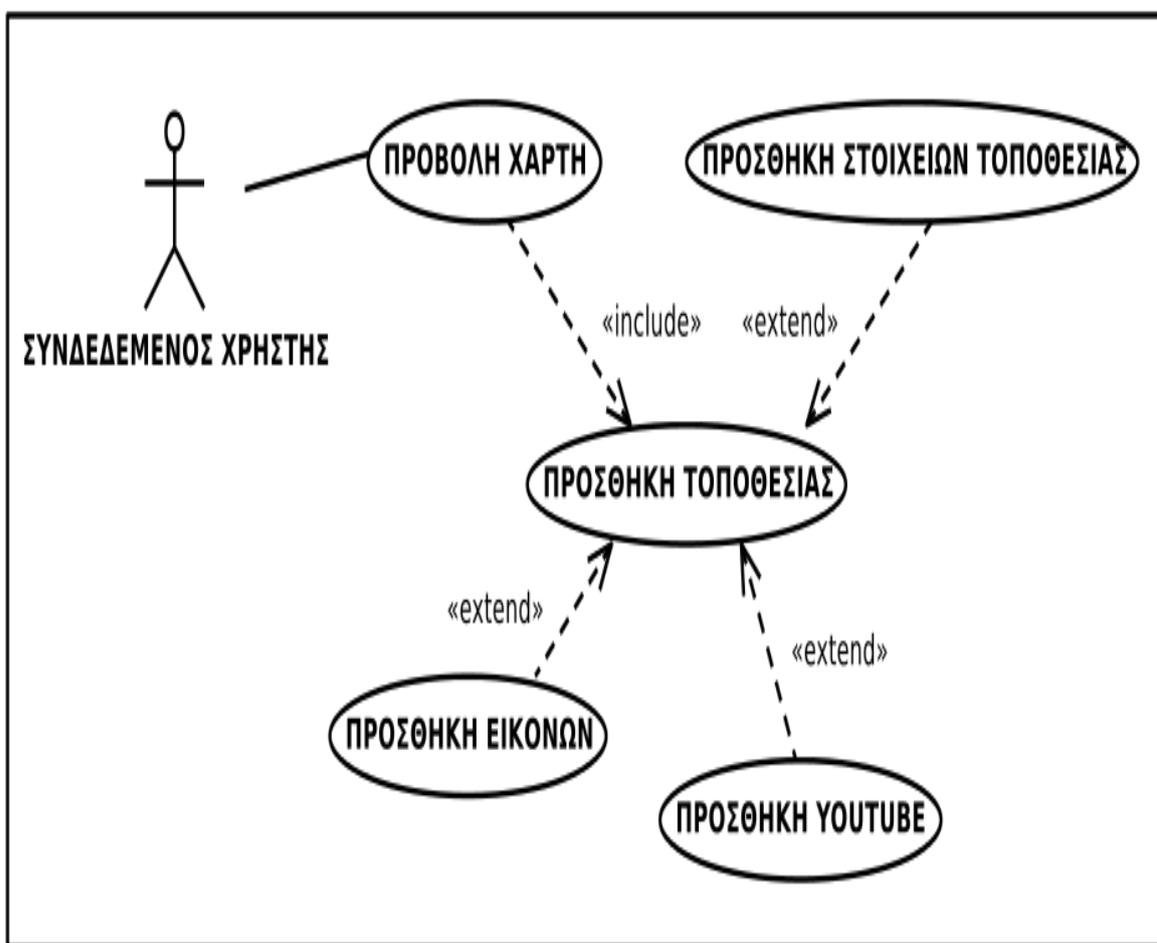
Σχήμα 1 UML USE CASE ΕΠΙΣΚΕΠΤΗ

Εκτός από τους απλούς χρήστες/επισκέπτες που όπως μόλις είδαμε μπορούν να βλέπουν τον χάρτη στην εφαρμογή κι εκτός από τα σχόλια όπου μπορούν και να αναρτούν, υπάρχει και ο συνδεδεμένος χρήστης. Ο τελευταίος θα αποτελεί τον διαχειριστή της εφαρμογής. Όπως φαίνεται και στο Σχήμα 2 ο συγκεκριμένος χρήστης κληρονομεί τις δυνατότητες του επισκέπτη της εφαρμογής και μπορεί επίσης να συνδέεται και να αποσυνδέεται από την εφαρμογή. Μετά την σύνδεση θα μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία.



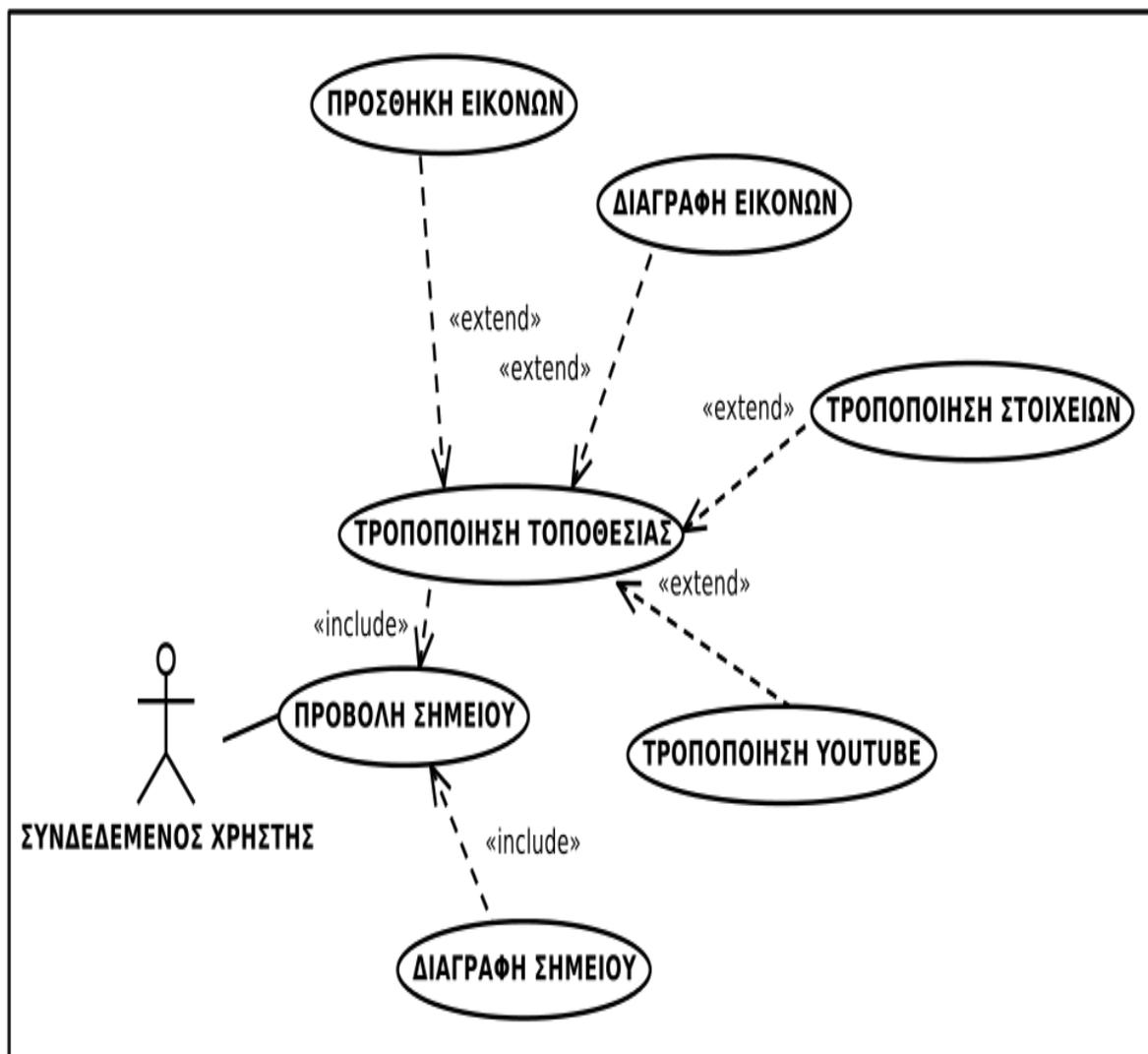
Σχήμα 2 UML USE CASE ΣΥΝΔΕΔΕΜΕΝΟΥ ΧΡΗΣΤΗ 1

Μια άλλη δυνατότητα του συνδεδεμένου χρήστη, όπως φαίνεται και στο Σχήμα 3, είναι η προσθήκη ενός νέου σημείου. Δυνατότητα η οποία περιλαμβάνεται (include) στην προβολή του χάρτη και οδηγεί σε νέο περιβάλλον. Σε αυτό θα μπορούμε να προσθέτουμε στις πληροφορίες για το εκάστοτε σημείο ενδιαφέροντος: Κείμενο, να ανεβάζουμε εικόνες από την τοποθεσία και να συμπληρώνουμε το βίντεο Youtube της Google.



Σχήμα 3 UML USECASE ΣΥΝΔΕΔΕΜΕΝΟΥ ΧΡΗΣΤΗ 2

Κάθε σημείο ενδιαφέροντος που έχουμε προσθέσει ίσως να χρειαστεί κάποιες μελλοντικές τροποποιήσεις ή απλά επιπρόσθετες πληροφορίες. Έτσι βασική δυνατότητα που προσφέρει το σύστημα στον συνδεδεμένο χρήστη, είναι η τροποποίηση των ήδη ενσωματωμένων σημείων ενδιαφέροντος. Σε αυτά, ανά πάσα στιγμή, θα μπορούμε να τροποποιούμε τα στοιχεία του, να διαγράψουμε εικόνες που είναι συνδεδεμένες με αυτό ή να προσθέσουμε νέες, όπως και να αλλάξουμε το βίντεο Youtube της Google που έχει.



Σχήμα 4 UML USE CASE ΣΥΝΔΕΔΕΜΕΝΟΥ ΧΡΗΣΤΗ 3

ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ

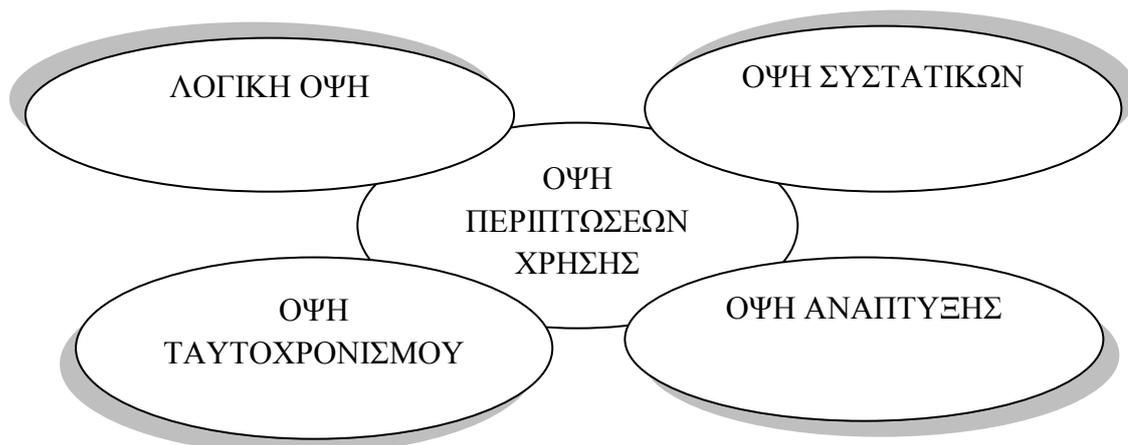
UML

Η UML (Unified Modeling Language = Ενοποιημένη Γλώσσα Περιβάλλοντος), σύμφωνα με τον οργανισμό διαχείρισής της (4), είναι «μια γλώσσα γραφικού τύπου που χρειάζεται για την ανάπτυξη και την τεκμηρίωση συστήματος λογισμικού». Κάθε αντικειμενοστραφής εφαρμογή για την μοντελοποίηση της χρησιμοποιεί την UML, καθώς αποτελεί στάνταρ στην εποχή μας.

Χρησιμοποιείται για την μοντελοποίηση όχι μόνο των εφαρμογών, αλλά και των συστημάτων (πληροφοριακά, επιχειρησιακά, τεχνολογικά κ.α.) που περιέχουν τις εφαρμογές. Η χρήση της αυτή αποτελεί την πιο δυνατή της εφαρμογή. Επιπροσθέτως, προσφέρεται και για την εκμάθηση της αντικειμενοστραφούς νοοτροπίας προγραμματισμού. (5)

Στην διαδικασία μοντελοποίησης της UML οι λογικές ενότητες ονομάζονται όψεις. Αυτές κατηγοριοποιούν τα διαγράμματα ανάλογα με την χρησιμότητα που έχουν αυτά. Πολλά διαγράμματα, μάλιστα, μπορεί να ανήκουν σε διάφορες όψεις.

- Όψη περιπτώσεων χρήσης (Use Case View). Περιγράφει το πώς λειτουργεί το σύστημα. Το πώς φαίνεται εξωτερικά. Πρακτικά αυτό πρέπει να είναι το τελικό αποτέλεσμα.
- Όψη συστατικών (Component View). Περιγράφει τα βασικά μέρη του συστήματος.
- Λογική όψη (Logical View). Περιγράφει την εσωτερική λειτουργία του συστήματος.
- Όψη ταυτοχρονισμού (Concurrency View). Περιγράφει τις ταυτόχρονες εργασίες στο σύστημα.
- Όψη ανάπτυξης (Deployment View). Περιγράφει το φυσικό επίπεδο του συστήματος.



Σχήμα 5 ΟΨΕΙΣ ΤΗΣ UML

Οι βασικές μορφές της UML είναι τα διαγράμματα. Τα κύρια διαγράμματα είναι:

- Use Case Diagram (Διάγραμμα Περιπτώσεων Χρήσης).
- Sequence Diagram (Διάγραμμα κλάσεων).
- Sequence Diagram (Διάγραμμα Ακολουθίας).
- Statechart Diagram (Διάγραμμα Καταστάσεων).
- Activity Diagram (Διάγραμμα Δραστηριότητας).
- Component Diagram (Διάγραμμα Συστατικών).
- Deployment Diagram (Διάγραμμα Ανάπτυξης).

Εμείς χρησιμοποιήσαμε όπως φάνηκε ήδη τα διαγράμματα περιπτώσεων χρήσης, στις απαιτήσεις χρηστών.

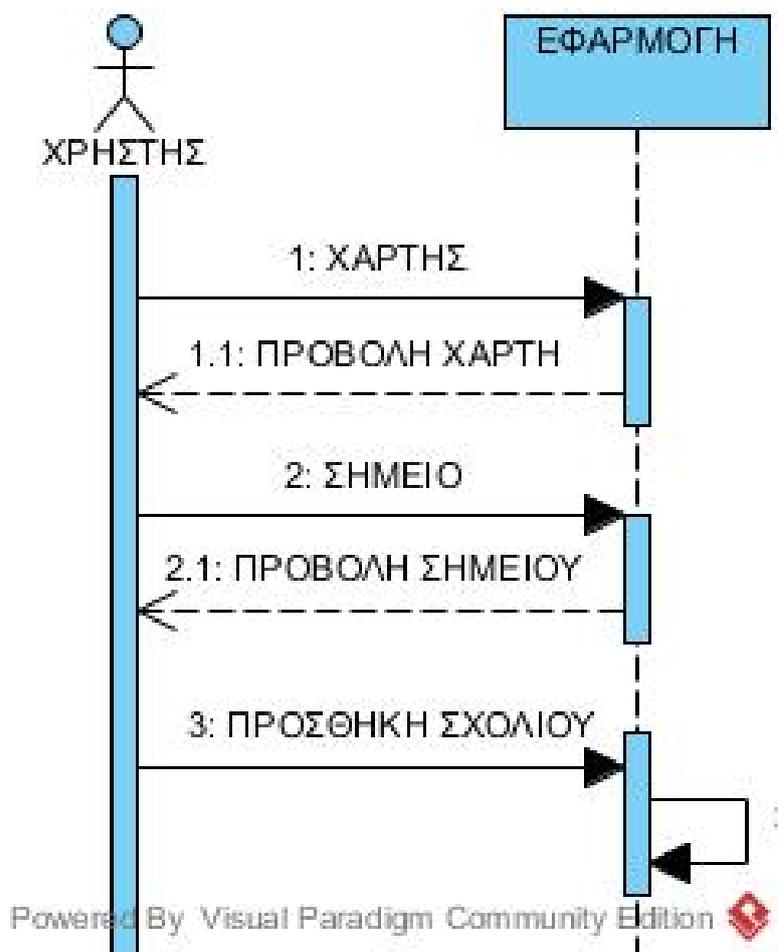
Παρακάτω θα δούμε την χρήση των διαγραμμάτων ακολουθιών και των διαγραμμάτων δραστηριότητας.

Sequence Diagram

Σε αυτό το σημείο της έρευνας μας έπρεπε να επιλέξουμε τον τρόπο με τον οποίο θα υλοποιήσουμε την εφαρμογή μας. Έτσι ακολουθώντας την συνήθη διαδικασία αρχίσαμε να σχεδιάζουμε το καινούριο σύστημα.

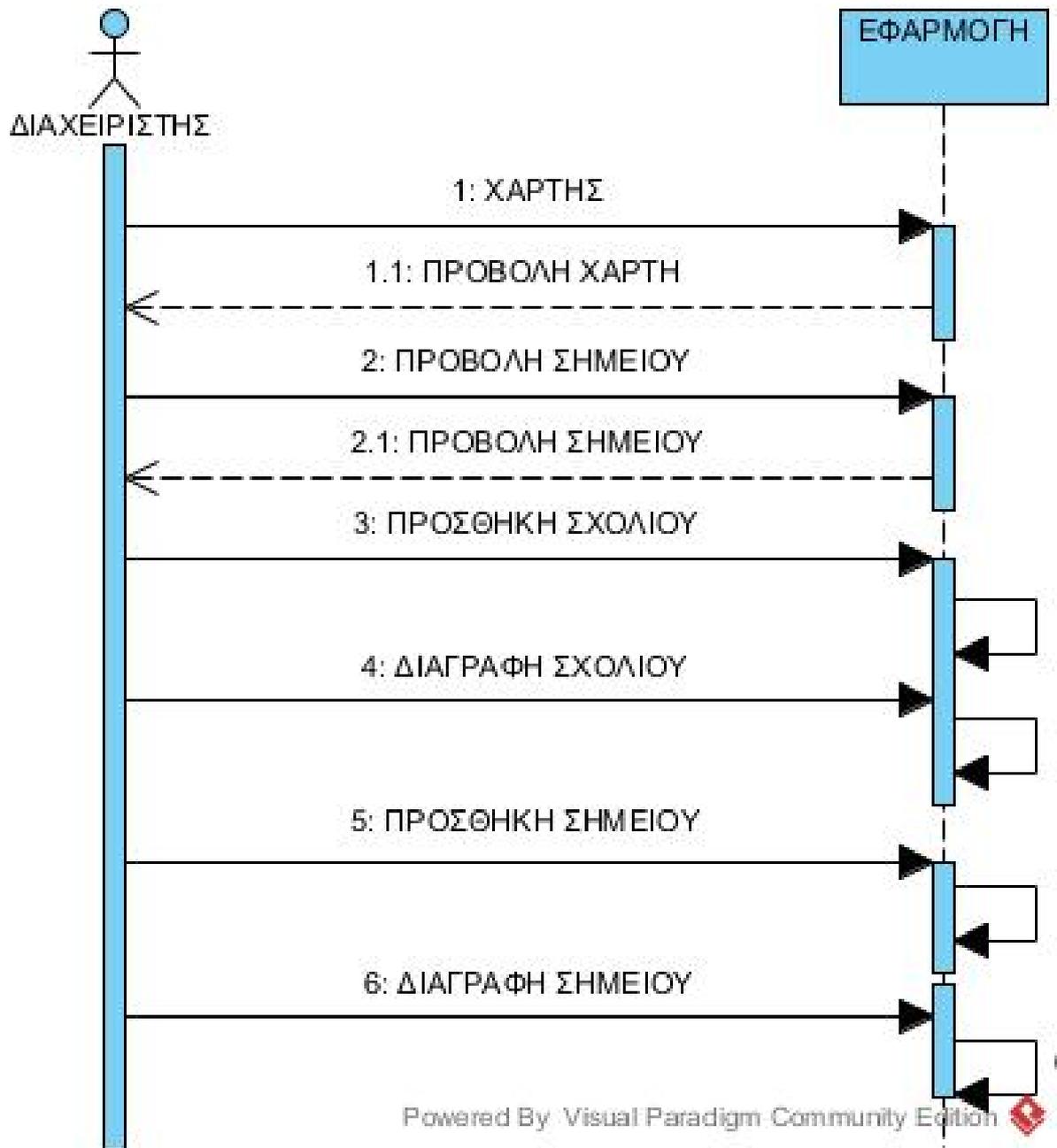
Αρχικά με την χρήση sequence διαγραμμάτων του UML μοντέλου, σχεδιάσαμε τις διαδρομές κάθε χρήστη. Όπως προαναφέραμε θα υπάρχουν δύο ειδών χρήστες: Οι χρήστες/επισκέπτες και οι διαχειριστές.

Ξεκινώντας την περιγραφή των δυνατοτήτων που έχουν, ο απλός χρήστης (που δεν χρειάζεται να συνδεθεί με το σύστημα) μπορεί να δει τον χάρτη. Στην συνέχεια να επιλέξει κάποιο σημείο ενδιαφέροντος και να προβάλει τις πληροφορίες αυτού. Ακολουθεί η προσθήκη σχολίων σχετικά με το σημείο.



Σχήμα 6 SEQUENCE DIAGRAM ΧΡΗΣΤΗ

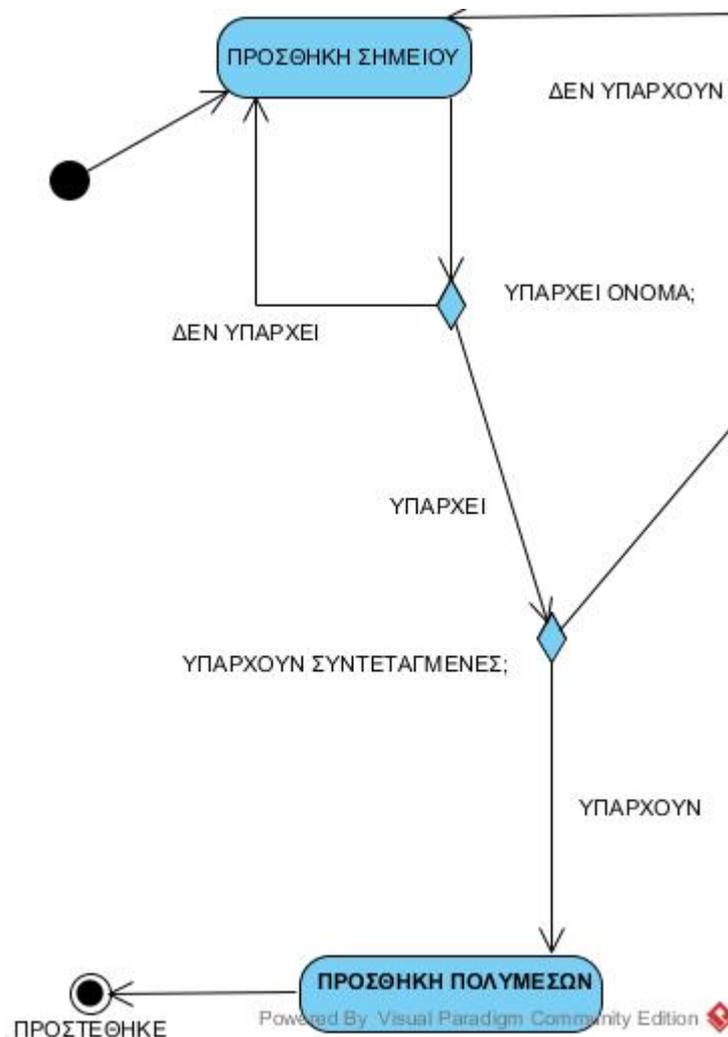
Ο διαχειριστής από την άλλη πλευρά, θα μπορεί να έχει πρόσβαση και σε επιπλέον δυνατότητες της εφαρμογής μας. Εκτός από τις βασικές, όπως η προβολή του χάρτη, των σημείων και την προσθήκη σχολίων, θα μπορεί να έχει την ελευθερία να διαγράφει σχόλια, να προσθέτει νέα σημεία και την επιλογή να διαγράφει κάποιο σημείο τουριστικού ενδιαφέροντος.



Σχήμα 7 SEQUENCE DIAGRAM ΔΙΑΧΕΙΡΙΣΤΗ

Activity Diagram

Μια από τις διαδικασίες που χρειάζεται και αναλύθηκε είναι η διαδικασία προσθήκης σημείου τουριστικού ενδιαφέροντος. Κατά την διαδικασία αυτή θα πρέπει να γίνει ο έλεγχος κάποιων δεδομένων που θα εισαχθούν. Τα βασικά δεδομένα που ελέγχονται είναι αν υπάρχει το όνομα του σημείου και το αν έχουν εισαχθεί οι συντεταγμένες του. Πρόκειται για τα υποχρεωτικά πεδία που θα αναζητούνται σε κάθε νέα εγγραφή. Στην συνέχεια υπάρχει η δυνατότητα εισαγωγής κάποιων πολυμέσων, δεν είναι όμως υποχρεωτική η προσθήκη τους.



Σχήμα 8 ACTIVITY DIAGRAM ΠΡΟΣΘΗΚΗ ΣΗΜΕΙΟΥ

Υπάρχουσα κατάσταση. Σύγκριση πλατφόρμων με απαιτήσεις χρηστών. Επιλογή πλατφόρμας .

Για την εργασία χρησιμοποιήσαμε συγκεκριμένα εργαλεία, τα οποία επιλέξαμε με σκοπό να υπάρχει μειωμένο κόστος (ως και μηδενικό ήτοι δυνατόν), ευκολία στην εκκίνηση της διαδικασίας δημιουργίας και την συνέχεια της ανάπτυξής της. Όπως επίσης ευκολία στην χρήση, χρήση των πιο σύγχρονων εργαλείων, δυνατότητα συνέχισης του έργου.

Χρησιμοποιήθηκαν εργαλεία ανοικτού κώδικα μιας και εξυπηρετούσαν τους σκοπούς μας. Το κόστος τους είναι μηδενικό και όσα χρησιμοποιήσαμε είχαν γνωσιακή βάση η οποία υπερκάλυπτε τις ανάγκες της εργασίας.

ΛΕΙΤΟΥΡΓΙΚΟ

Σαν λειτουργικό σύστημα επιλέχθηκε το ubuntu, μια διανομή του linux, που είναι η πιο διαδεδομένη. Προσφέρει μεγάλη ευκολία στην χρήση και είναι από τα πιο ασφαλή λειτουργικά συστήματα. Δίνει τη δυνατότητα εύκολης επίλυσης του όποιου προβλήματος θα μπορούσε να προκύψει κατά την διάρκεια της ανάπτυξής. Επιπλέον, η ευκολία με την οποία μπορούμε να τροποποιήσουμε το λειτουργικό είναι ένα από τα θετικά του, καθώς μπορούμε να επιλέξουμε τον τρόπο που λειτουργεί και με ποιό τρόπο θα το διαχειριστούμε.

Το λειτουργικό της εφαρμογής μας τρέχει σε ένα server της amazon στην Φρανκφούρτη της Γερμανίας μιας και ήταν εύκολο και χωρίς έξοδα να το δημιουργήσουμε εκεί. Το aws (amazon web services) αποτελεί μια συνήθη λύση για τους προγραμματιστές, μιας και δίνει την δυνατότητα να χρησιμοποιείς ένα ολόκληρο σύστημα, να το τροποποιείς και να το ρυθμίζεις κατά το δοκούν.

Δεν επιλέξαμε κάποιο άλλο λειτουργικό κυρίως για οικονομικούς λόγους. Και το Windows και το MacOS σε υποχρεώνουν να τα αγοράσεις. Βέβαια, πέραν του οικονομικού κόστους, αμφότερα θα μπορούσαν να υποστηρίξουν τα υπόλοιπα εργαλεία και πλατφόρμες που επιλέξαμε. Συγκεκριμένα:

- Το λειτουργικό Windows έχει πιο χαμηλό κόστος αλλά δεν εξυπηρετούσε από άποψη ασφαλείας. Έχοντας πολλά διαπιστωμένα κενά ασφαλείας θα έθετε σε κίνδυνο το σύστημα μας. Ακόμη δεν είναι εύκολα παραμετροποιήσιμο, μιας και δίνονται εξ' αρχής έτοιμα τα λειτουργικά εργαλεία για αυτή την δουλειά, χωρίς να μπορείς να τα τροποποιήσεις (δυνατότητα που προσφέρουν τα Linux και τα MacOS).
- Ενώ το λειτουργικό σύστημα MacOS αποτελεί μια καλή λύση από άποψη ασφαλείας, έχει πολύ υψηλό κόστος απόκτησης, καθώς εκτός από την αγορά του ίδιου χρειάζεται και το ανάλογο υλικό για να λειτουργήσει. Έπειτα, παρότι είναι από τα λειτουργικά που μπορείς να τροποποιήσεις αρκετά εύκολα, δεν θα κάλυπτε τις ανάγκες που θα είχαμε.

Στον

Πίνακας 1 ακολουθεί συνοπτική σύγκριση των ανωτέρω χαρακτηριστικών:

Πίνακας 1 ΣΥΓΚΡΙΣΗ ΛΕΙΤΟΥΡΓΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ιδιότητα:	<i>Κόστος</i>	<i>Ευελιξία</i>	<i>Ασφάλεια</i>
Λειτουργικό Σύστημα:			
<u>Linux</u>	Μηδενικό	Υψηλό Επίπεδο	Υψηλό Επίπεδο
<u>Windows</u>	Μεσαίο	Μεσαίο Επίπεδο	Χαμηλό Επίπεδο

<u>MacOS</u>	Πολύ Υψηλό	Υψηλό Επίπεδο	Μεσαίο Επίπεδο
---------------------	------------	---------------	----------------

ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Η χρήση μιας σχεσιακής βάσης είναι επιτακτική, καθώς είναι το μοντέλο το οποίο εξυπηρετεί το σύστημα μας. Ενδείκνυται για το μέγεθος που θα έχει η εφαρμογή μας και για τον τρόπο που θα επικοινωνεί με τα υπόλοιπα εργαλεία που επιλέξαμε.

Σαν μηχανή βάσης δεδομένων χρησιμοποιήσαμε την πλατφόρμα MariaDB. Αποτελεί μια τροποποίηση της MySQL για την αντικατάσταση αυτής. Σχεδιασμένη από προγραμματιστές της MySQL που διαφώνησαν με την εξαγορά αυτής από την εταιρεία Oracle. Είναι προγραμματιστές οι οποίοι ευελπιστούν ότι η υλοποίησή τους θα μπορέσει να επιτύχει καθώς απευθύνονται σε προγραμματιστές με ένα εργαλείο που δημιουργείται από προγραμματιστές. Η επιλογή μας αυτή λειτουργεί άριστα στο λειτουργικό σύστημα που επιλέξαμε.

Εναλλακτική και απολύτως συμβατή χωρίς ιδιαίτερες διαφορές θα ήταν η MySQL. Διότι στον τρόπο που γίνεται η χρήση της βάσης δεδομένων δεν έχει καμία διαφορά με την MariaDB. Η επιλογή μας στηρίχτηκε κυρίως στον χαρακτήρα της τελευταίας.

Μια άλλη επιλογή θα μπορούσε να είναι η MSSQL, η οποία δεν μπορεί να χρησιμοποιηθεί γιατί δεν υποστηρίζεται από το λειτουργικό linux που έχουμε. Αποκλειστικά λειτουργεί με το λειτουργικό Windows της ίδιας εταιρείας. Έχει κόστος αγοράς μαζί με το λειτουργικό, αλλά βέβαια θα αποτελούσε μία αξιόπιστη λύση για κάποιον που θέλει να την χρησιμοποιήσει (6).

Αναφέρουμε τώρα συγκεντρωτικά τα παραπάνω στον Πίνακα 2:

Πίνακας 2 ΣΥΓΚΡΙΣΗ ΣΧΕΣΙΑΚΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Ιδιότητα:	<i>Κόστος Χρήσης</i>	<i>Είδος Μοντέλου Βάσης</i>	<i>Linux</i>	<i>Windows</i>	<i>MacOS</i>
Βάση Δεδομένων:					
<u>MySQL</u>	Μηδενικό	Σχεσιακή	Ναι	Ναι	Ναι
<u>MS SQL</u>	Υψηλό	Σχεσιακή	Όχι	Ναι	Όχι

<u>MariaDB</u>	Μηδενικό	Σχεσιακή	Ναι	Ναι	Όχι
-----------------------	----------	----------	-----	-----	-----

ΠΛΑΤΦΟΡΜΕΣ - ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Το βασικό εργαλείο ανάπτυξης για να υλοποιήσουμε το σύστημα μας έπρεπε να επιλεγεί μέσα από μια μεγάλη γκάμα. Οι κυριότερες επιλογές μας ήταν η PHP με πληθώρα εργαλείων ανάπτυξης, η .NET με την χρήση του Visual Studio και η Ruby, αυτή που επιλέχθηκε, με εργαλείο το Ruby on Rails.

Και οι τρεις λύσεις χρησιμοποιούν το MVC μοντέλο ανάπτυξης (7). Μπορούν να επικοινωνούν και με τις τρεις μηχανές βάσης δεδομένων όπως φαίνεται και στον Πίνακα 3.

Πίνακας 3 ΣΥΓΚΡΙΣΗ ΠΛΑΤΦΟΡΜΩΝ

	<i>Κόστος</i>	<i>Αξιοπιστία</i>	<i>Linux</i>	<i>Windows</i>	<i>MacOS</i>	<i>MySQL</i>	<i>MSSQL</i>	<i>MariaDB</i>
<u>PHP</u>	Μηδενικό	Υψηλή	Ναι	Ναι	Ναι	Ναι	Ναι	Ναι
<u>.NET (VS)</u>	Υψηλό	Υψηλή	Όχι	Ναι	Όχι	Ναι	Ναι	Ναι
<u>Ruby (RoR)</u>	Μηδενικό	Υψηλή	Ναι	Ναι	Ναι	Ναι	Ναι	Ναι

Η πλατφόρμα της Microsoft (.NET) αποτελεί ίσως την καλύτερη λύση. Αλλά έχει κόστος απόκτησης, μιας και λειτουργεί αποκλειστικά σε λειτουργικό περιβάλλον της ίδιας της εταιρείας (Windows), που το ανεβάζει περισσότερο. Αυτή η πλατφόρμα έχει την μεγαλύτερη γνωσιακή βάση, μιας και απαιτεί την αγορά της, παρέχει πλήρη υποστήριξη. Το λειτουργικό και τα βοηθητικά εργαλεία είναι του ίδιου κατασκευαστή και λειτουργούν άριστα μεταξύ τους.

Η πλατφόρμες της PHP είναι για διαδικτυακές εφαρμογές μόνο, αλλά δεν έχουν την καλύτερη απόδοση. Η κοινότητα, όντας η μεγαλύτερη στον κόσμο σε διαδικτυακές λύσεις, παρέχει πάρα πολλά εργαλεία και μια μεγάλη γνωσιακή βάση, για την καλύτερη

απόδοση του προγραμματιστή. Όμως τα εργαλεία δεν είναι πλήρως συμβατά μεταξύ τους. Τα περισσότερα περιβάλλοντα ανάπτυξης είναι εξαρχής περισσότερο υπερφορτωμένα από όσο χρειάζεται.

Η Ruby είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού που είναι σχετικά νέα (1996 η έκδοση 1.0, προετοιμαζόταν από το 1993). Ξεκινώντας είχε μια ραγδαία εξάπλωση στην Ασία και κύρια στην Ιαπωνία που οι προγραμματιστές την αποδέχτηκαν πολύ γρήγορα. Επειδή σαν γλώσσα έχει επηρεαστεί από πολλές άλλες (όπως είναι οι C++, Python, LISP και η ADA) έχει ενσωματώσει τεχνολογίες αυτών, τα καλύτερα σημεία τους. Σε συνδυασμό με τις νέες επιπρόσθετες δυνατότητες αυτής, π.χ. τα πάντα είναι αντικείμενα, έχει αποκτήσει χαρακτηριστικά όπως η δύναμη και η ευελιξία.

Οι χρήστες μαθαίνοντας Ruby μαθαίνουν τα καλύτερα στοιχεία των άλλων γλωσσών. Χαρακτηριστικό παράδειγμα είναι ο τρόπος που δηλώνουμε μεταβλητές στην γλώσσα αυτή. Παίρνοντας στοιχεία των γλωσσών Perl και Python, αλλά τροποποιημένες, να μεν οι δηλώσεις των μεταβλητών και των κλάσεων γίνονται με λέξεις κλειδιά αλλά δεν είναι σαν την Perl που χρειάζεται ειδικό χαρακτήρα στην αρχή της λέξης αυτής.

Ακολουθεί την Open Source κουλτούρα έτσι τα περισσότερα εργαλεία ή τμήμα αυτών παρέχονται δωρεάν. Αυτό που εκτόξευσε του χρήστες της Ruby ήταν η δημιουργία του framework Ruby on Rails (2004), αυτό που επιλέξαμε και για την εργασία μας.

Όταν η RoR (Ruby on Rails) δόθηκε στον κόσμο σαν ένα ανοιχτού λογισμικού περιβάλλον δημιουργίας διαδικτυακών εφαρμογών, δεν περίμενε κανείς ότι θα φτάσει στο σημείο να αποτελεί σήμερα ένα καινοτόμο περιβάλλον που αναδεικνύει συνεχώς νέες μεθόδους επίλυσης των προβλημάτων. Αποτελεί παράδειγμα το πώς η κοινότητα του επιλύει τα προβλήματα στον κώδικα και πόσο συχνά εμφανίζουν λύσεις στο προσκήνιο που αργότερα ενσωματώνουν και οι άλλες πλατφόρμες.

Με την εντολή δημιουργίας νέας εργασία της Rails δημιουργείται ένας κατάλογος με τα βασικά περιεχόμενα χωρίς επιπλέον χαρακτηριστικά, τα οποία εμείς προσθέτουμε ανάλογα με τις ανάγκες μας. Το αρχικό περιβάλλον είναι κυρίως ως εξής:

```
app/  
app/assets/
```

```
app/controllers/  
app/helpers/  
app/models/  
app/views/  
bin/  
config/  
config/database.yml  
config/routes.rb  
config/environment/  
db/  
db/migrate/  
doc/  
lib/  
log/  
public/  
spec/  
tmp/  
vendor/  
gemfile
```

όπου:

app: Ο βασικότερος κατάλογος όλης της εφαρμογής καθώς εδώ προγραμματίζουμε κυρίως. Περιέχει τον κώδικα της εφαρμογής μας.

Στους υποφακέλους του είναι:

Ο assets που περιέχει τα css, javascript, εικόνες κ.α. που ενσωματώνουμε εμείς στην εφαρμογή.

Ο controllers περιέχει τους ελεγκτές του προτύπου MVC. Ελέγχουν τα αιτήματα των χρηστών και τους τα απαντάνε.

Ο helpers εμπεριέχει διάφορα βοηθητικά αρχεία που απλοποιούν την λειτουργία του προτύπου MVC.

Ο `models` είναι εδώ τα μοντέλα του προτύπου MVC. Επικοινωνούν με την βάση δεδομένων και εξασφαλίζουν την ορθή λειτουργία της.

Ο `views` περιέχει τις προβολές του προτύπου MVC. Ότι θα βλέπει ο χρήστης της εφαρμογής θα είναι από εδώ. Περιέχει και τον φάκελο `layouts` που έχει τα πρότυπα του περιβάλλοντος επικοινωνίας.

Στον φάκελο `config` περιέχονται δύο από τα βασικότερα αρχεία της εφαρμογής, το αρχείο `database.yml` και το `routes.rb`. Το πρώτο είναι υπεύθυνο για την επικοινωνία με την βάση δεδομένων μας, αποθηκεύονται εκεί ο τρόπος σύνδεσης με αυτή, το όνομα χρήστη και ο κωδικός πρόσβασης, η κωδικοποίηση των κειμένων σε αυτή και ποιες βάσεις χρησιμοποιούνται σε κάθε επίπεδο λειτουργίας της εφαρμογής. Ενώ το δεύτερο αρχείο αποτελεί τον χάρτη των προβολών, εδώ δίνουμε δυνατότητα πρόσβασης των χρηστών να βλέπουν την εφαρμογή μας. Στον υποφάκελο `environment` υπάρχουν οι ρυθμίσεις των επιπέδων λειτουργίας της εφαρμογής.

Στον υποφάκελο `migrate` του φακέλου `db` υπάρχουν οι τροποποιήσεις που έχουμε κάνει στην βάση δεδομένων μας.

Ο φάκελος `log` περιέχει τα αρχεία καταγραφής της εφαρμογής. Πολύ χρήσιμα και στο επίπεδο ανάπτυξης και δοκιμής αλλά και στο επίπεδο παραγωγής για την επίλυση των προβλημάτων που προκύπτουν.

Ο φάκελος `public` είναι αυτός που υποδεικνύουμε στον εξυπηρετητή ιστού για αρχικό της εφαρμογής.

Ενώ στον φάκελο `vendor` τοποθετούμε ότι έχουμε από τρίτους κατασκευαστές, κυρίως `css` και `javascript` αρχεία, για να ενισχύσουν την εφαρμογή μας.

Στο αρχείο `gemfile` δηλώνουμε τις βιβλιοθήκες που θα χρησιμοποιήσουμε ώστε αυτόματα να ενσωματωθούν στην εφαρμογή.

Σε αυτό το σημείο τονίζουμε πως η Ruby on Rails μας εξυπηρέτησε γιατί έχει τις εξής δυνατότητες:

1. Μοντέλο MVC. Τρεις μορφές στην εφαρμογή, ο ελεγκτής (`controller`), το μοντέλο (`model`) και η προβολή (`view`).

2. Τρία επίπεδα λειτουργίας της εφαρμογής:

- *Επίπεδο ανάπτυξης.* Όταν γράφουμε τον κώδικα μπορούμε και βλέπουμε το αποτέλεσμα άμεσα με την χρήση του webrick. Εργαλείο εξυπηρέτησης ιστού που άμεσα δέχεται τις αλλαγές και μπορούμε να απασφαλμάτουμε (debug) την εφαρμογή μας, καθώς περιέχεται το ανάλογο εργαλείο στον εξυπηρετητή. Έχουμε πρόσβαση από την θύρα 3000 του υπολογιστικού συστήματος μας.
- *Επίπεδο δοκιμής.* Μπορούμε να ενσωματώσουμε στην εφαρμογή διάφορες αυτοματοποιημένες δοκιμές του κώδικα και αν πάρουμε το σωστο αποτέλεσμα μπορούμε να είμαστε σίγουροι για την λειτουργία της εφαρμογής. Το κέλυφος (shell) irb που ακολουθεί κάθε Ruby on Rails εφαρμογή είναι άμεσο και μπορούμε να το χρησιμοποιούμε για να βλέπουμε το αποτέλεσμα κάθε εντολής.
- *Επίπεδο παραγωγής.* Εδώ είμαστε όταν η εφαρμογή μας τρέχει σε πραγματικό επίπεδο. Είμαστε σίγουροι για το τι έχουμε κάνει αφού μπορέσαμε να ελέγξουμε κάθε λειτουργία που προγραμματίσαμε στην εφαρμογή μας.

3. Το εργαλείο bundler όταν εκτελείται αναζητεί στο υπολογιστικό μας σύστημα την ύπαρξη των απαιτούμενων από την εφαρμογή βιβλιοθήκες (gem) και αν δεν τις βρει τις αναζητεί στο διαδίκτυο. Αυτή η αυτοματοποίηση στην χρήση βιβλιοθηκών από την εφαρμογή έχει λύσει τα χέρια των προγραμματιστών, αφού είναι ταχύτατη και οι βιβλιοθήκες λόγω της νοοτροπίας ανοιχτού λογισμικού που ακολουθείται, δεν έχουν κόστος.

4. Το ενσωματωμένο εργαλείο rake που τροποποιεί ανάλογα με τις επιθυμίες μας την διαμόρφωση της βάσης δεδομένων μας. Μπορούμε να την δημιουργήσουμε, να την αδειάσουμε, είτε να τις προσθέσουμε πεδία, είτε να την επιστρέψουμε σε προηγούμενο σημείο.

5. Η χρήση συμβάσεων αντί για ρυθμίσεις (Convention over Configuration) που παρέχει το περιβάλλον της Ruby on Rails μας επιτρέπει να μην χρειάζεται να πειράζουμε πολλά αρχεία για να ρυθμίσουμε την εφαρμογή μας. Έχει προεπιλεγμένες πολλές από τις ρυθμίσεις, συμβάσεις που έχουμε δεχθεί για την λειτουργία, σε σχέση με άλλες πλατφόρμες που χρειάζεται να επεξεργαστούμε μεγάλα αρχεία ρυθμίσεων πριν καν ξεκινήσουμε να προγραμματίζουμε. Το μόνο που χρειάζεται τις περισσότερες φορές να επηρεάσει ο προγραμματιστής είναι το αρχείο για την επικοινωνία με την βάση δεδομένων.

6. Η λογική της μη επανάληψης που χρησιμοποιείται στην πλατφόρμα, δηλαδή ο προγραμματισμός μιας λειτουργίας να μπαίνει κάπου κεντρικά και να καλείται κάθε φορά που χρειάζεται. Η συνεχής ρύθμιση για την σύνδεση στην βάση δεδομένων και ο κώδικας που επαναλαμβάνεται για την κάθε λήψη δεδομένων που χρησιμοποιείται από τις άλλες πλατφόρμες κατά κόρον είναι ένα παράδειγμα αυτής της λογικής καθώς εδώ κάθε φορά που το μοντέλο χρειάζεται δεδομένα δεν του γράφουμε κώδικα για την σύνδεση στην βάση.

7. Η χρήση διερμηνέα (interpreter) σε διαδικτυακές εφαρμογές ενδείκνυται σε σχέση με μεταγλωττιστή. Η εύρεση του σφάλματος (bug) την ώρα της εκτέλεσης είναι ευκολότερη σε on the fly συνθήκες.

8. Ο ευέλικτος τρόπος προγραμματισμού είναι ένα δυνατό χαρακτηριστικό, καθώς μπορείς να γράφεις κώδικα για το γραφικό περιβάλλον και τα διάφορα layout πρώτα και στην συνέχεια να ασχοληθείς με το τι θα περιέχει η εφαρμογή σαν δεδομένα.

Εξυπηρετητής Ιστού (Web Server)

Για την εργασία μας δίνεται έχουμε τη δυνατότητα να χρησιμοποιήσουμε για την εξυπηρέτηση ιστού τον Apache2 ή τον Nginx, ή ακόμα και τις δύο επιλογές. Είναι τα διασημότερα για το λειτουργικό σύστημα που χρησιμοποιούμε. Εμείς επιλέξαμε τον Apache2 (8). Εξηγούμε στη συνέχεια αυτή την επιλογή μας (Πίνακας 4):

Πίνακας 4 ΣΥΓΚΡΙΣΗ ΕΞΥΠΗΡΕΤΗΤΩΝ ΙΣΤΟΥ

Ιδιότητες:	Κόστος	Ταχύτητα εξυπηρέτησης σε μικρό φόρτο	Ταχύτητα εξυπηρέτησης σε μεγάλο φόρτο
Εξυπηρετητές Ιστού:			
<u>Apache2</u>	Μηδενικό	Υψηλή	Χαμηλή
<u>Nginx</u>	Μηδενικό	Χαμηλή	Υψηλή

Και οι δύο εξυπηρετητές χρησιμοποιούνται κατά κόρον σε λειτουργικό Linux και είναι οι πιο ασφαλείς που υπάρχουν. Χρησιμοποιούν διαφορετική τεχνική εξυπηρέτησης, η προσέγγιση του Apache2 είναι να δημιουργεί μία διεργασία για κάθε αίτημα, βέβαια στις πιο σύγχρονες εκδόσεις του μία διεργασία εξυπηρετεί περισσότερα αιτήματα, και στην συνέχεια την καταστρέφει. (9) Ο Nginx αντιθέτως δεν ακολουθεί αυτή την διαδικασία, η συνήθης λειτουργία του είναι να δημιουργεί ένα νήμα για κάθε επεξεργαστή και κάθε ένα από αυτά αναλαμβάνει να εξυπηρετεί αιτήματα.

Έχοντας εντοπίσει την βασική αυτή διαφορά των δύο εξυπηρετητών ιστού, ακολουθήσαμε τον εξής συλλογισμό: Ο Apache2 δυσκολεύεται στην γρήγορη εξυπηρέτηση με μεγάλο φόρτο, ενώ αντιθέτως εξυπηρετεί γρηγορότερα τις στατικές σελίδες και όταν δεν υπάρχει μεγάλος φόρτος εργασίας. Όμως εμείς δεν θα έχουμε μεγάλο φόρτο, αφού οι χρήστες θα βλέπουν αυτό που θέλουν και θα σταματούν την περιήγηση, κάναμε την επιλογή μας. (10)

Περιγραφή χρησιμοποιούμενης μεθοδολογίας προκειμένου να δοθεί λύση.

AWS (Amazon Web Services)

Όπως αναφέραμε ήδη το σύστημα μας φιλοξενείται σε έναν εξυπηρετητή της εταιρείας amazon, στο σύστημα της aws (11). Εκεί μας δόθηκε η δυνατότητα να επιλέξουμε ανάμεσα σε διάφορα λειτουργικά συστήματα και επιλέξαμε το Ubuntu Server 14.04 LTS (12).

Στην συνέχεια ερωτηθήκαμε για το υλικό πάνω στο οποίο θα λειτουργούμε τον εξυπηρετητή μας. Διαλέξαμε το χωρίς κόστος t2.micro το οποίο αποτελείται από 1 επεξεργαστή στα 2.5 GHz της Intel (της οικογενείας Xeon), με μνήμη RAM 1 GiB. Για αποθηκευτικό χώρο δημιουργήσαμε έναν δίσκο 15 GB που ικανοποιεί τις ανάγκες της ερευνάς μας. Τέλος επιλέξαμε ποιές πόρτες του πρωτοκόλλου IP θα επιτρέπουμε να είναι ανοιχτές στο διαδίκτυο.

Εμείς επιτρέπουμε τις πόρτες:

- 22 (SSH πρωτόκολλο για πρόσβαση στο περιβάλλον του εξυπηρετητή)
- 80 (Http πρωτόκολλο για προβολή της εφαρμογής μας)
- 3000 (Http πρωτόκολλο για έλεγχο της εφαρμογής κατά την διαδικασία της ανάπτυξης)

Ubuntu

1. Αφού ξεκίνησε το λειτουργικό να τρέχει, συνδεθήκαμε στον περιβάλλον του χρησιμοποιώντας την εντολή:

```
ssh -i ptychiakh.pem ubuntu@54.93.54.225
```

όπου ptychiakh.pem εννοείται το αρχείο που περιέχει το κρυφό κλειδί.

2. Αναβαθμίσαμε το σύστημα στις τελευταίες εκδόσεις των εφαρμογών, εκτελώντας τις παρακάτω εντολές:

```
sudo apt-get update  
sudo apt-get upgrade
```

3. Δημιουργήσαμε έναν νέο χρήστη με το όνομα bill. Του δώσαμε δικαιώματα υπερχρήστη και αντιγράψαμε τα κλειδιά του πρωτοκόλλου SSH στον κατάλληλο φάκελο, αφού τον δημιουργήσαμε, πληκτρολογώντας τα παρακάτω:

```
sudo adduser bill  
sudo visudo  
sudo mkdir /home/bill/.ssh  
sudo cp .ssh/authorized_keys  
    /home/bill/.ssh/authorized_keys  
sudo chown -R bill:bill /home/bill/.ssh
```

4. Προσθέσαμε γραφικό περιβάλλον για την καλύτερη διαχείριση του. Εγκαταστήσαμε το xfce4 μιας και είναι ένα από τα ελαφρύτερα και πιο σταθερά περιβάλλοντα. Έγινε με την εντολή:

```
sudo apt-get install xfce4 xfce4-goodies xfce4-artwork
xubuntu-icon-theme
```

5. Για τον απομακρυσμένο έλεγχο ένα από τα καλύτερα εργαλεία είναι το x2go (13). Το βάλαμε με τις εντολές:

```
sudo add-apt-repository ppa:x2go/stable
sudo apt-get install x2goserver x2goserver-xsession
```

RVM

Στην συνέχεια εγκαταστήσαμε το εργαλείο RVM (Ruby Version Manager) (14). Το οποίο χρειάζεται για την εγκατάσταση της νεώτερης έκδοσης της Ruby και της Rails, και την δυνατότητα ελέγχου των διάφορων gem της ruby. Ταυτόχρονα με την μία εγκαταστήσαμε και την ruby και το περιβάλλον rails.

```
gpg --keyserver hkp://keys.gnupg.net --recv-keys D39DC0E3
\curl -sSL https://get.rvm.io | bash -s stable -rails
```

Εγκαταστάθηκαν με επιτυχία η έκδοση 2.0.0p598 της ruby, η 2.4.5 των gem και η 4.1.8 της Rails.

MariaDB

Η εγκατάσταση της Βάσης Δεδομένων MariaDB (15) απαιτεί την εγκατάσταση του κλειδιού της και στην συνέχεια την εγκατάσταση της.

```
sudo apt-get install software-properties-common
sudo apt-key adv --recv-keys --keyserver
hkp://keyserver.ubuntu.com:80 0xcbc082a1bb943db
sudo add-apt-repository 'deb
http://ftp.hosteurope.de/mirror/mariadb.org/repo/5.5/ubuntu trusty
main'
sudo apt-get update
sudo apt-get install mariadb-server
```

Apache

Η διανομή ubuntu έχει τον εξυπηρετητή apache2 (16) προετοιμασμένο και το μόνο που χρειάζεται είναι η εγκατάσταση του από τα αποθετήρια της.

```
sudo apt-get install apache2 apache2-mpm-prefork apache2-
prefork-dev
```

Για να υποδείξουμε τον φάκελο που περιέχει την εφαρμογή μας δημιουργούμε ένα αρχείο ρυθμίσεων με το παρακάτω περιεχόμενο:

```
<VirtualHost *:80>
DocumentRoot "/home/bill/site-ptyxiakh/bill_ptyxiakh/public"
<Directory "/home/bill/site-ptyxiakh/bill_ptyxiakh/public">
Options -MultiViews
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

```
</VirtualHost>
```

PASSENGER

Για την προβολή της εφαρμογής μας, θα χρειαστούμε και το εργαλείο passenger της Phusion (17). Είναι το ενδιάμεσο μεταξύ του apache2 web server και της εφαρμογής μας. Θα τον προετοιμάσει, θα εγκαταστήσει τα σωστά module και όποια άλλα εργαλεία χρειαστούν, με στόχο να μπορεί να εξυπηρετήσει και τις εφαρμογές της πλατφόρμας Ruby on Rails. Κάναμε την εγκατάσταση του και στην συνέχεια το εκτελέσαμε

```
gem install passenger  
passenger-install-apache2-module
```

Παρουσίαση της υλοποίησης.

MVC

Η αρχιτεκτονική MVC που χρησιμοποιεί ruby on rails διαχωρίζει την εφαρμογή μας σε τρία μέρη:

1. Ελεγκτής (Controller). Υπεύθυνο για την λήψη του αιτήματος από τον χρήστη, στην συνέχεια συλλέγει τα δεδομένα που χρειάζεται από το μοντέλο, τα στέλνει στην προβολή όπου τα ενσωματώνει στο γραφικό περιβάλλον, στην περίπτωση μας για προβολή από φυλλομετρητή, και τελικά τα στέλνει πίσω στον χρήστη για προβολή.
2. Μοντέλο (Model). Αλληλεπιδρά με την βάση δεδομένων και είτε προσθέτει είτε συλλέγει δεδομένα από αυτήν. Είναι η διεπαφή με την βάση δεδομένων οπότε είναι και η ασφάλεια της, θέλει σωστή ρύθμιση για την μη απώλεια δεδομένων.
3. Προβολή (View). Η διαμόρφωση του γραφικού περιβάλλοντος είναι ευθύνη του. Ο ελεγκτής, αφού συλλέξει τα δεδομένα από το μοντέλο, τα φορτώνει στο γραφικό περιβάλλον και τα στέλνει στον φυλλομετρητή.

Εικόνα 4 TO MONTEAO MVC

Controller

Ο ελεγκτής είναι η καρδιά του συστήματος μας. Είναι ο οργανωτής των πάντων. Η εφαρμογή μας αποτελείται από τους παρακάτω ελεγκτές, όπως φαίνεται και στην Εικόνα 5:

- ApplicationController. Κεντρικός ελεγκτής της εφαρμογής, Έχει ευθύνη για κάποιες ασφάλειες συστήματος και είναι αυτός από όπου κληρονομούν ιδιότητες και οι υπόλοιποι ελεγκτές.
- StaticPagesController. Βοηθάει στην διαχείριση κάποιων σελίδων που δεν είναι δυναμικές. Αυτές είναι η αρχική, η σελίδα με τις πληροφορίες σχετικά με την εφαρμογή και τέλος η σελίδα της επικοινωνίας.

- UsersController. Αυτός ο ελεγκτής είναι υπεύθυνος για την λειτουργία των χρηστών. Από τις μεθόδους που έχει, η new χρειάζεται για να προβληθεί η δημιουργία νέου χρήστη και η create ασχολείται με την αποθήκευση του νέου χρήστη. Αναλόγως και η edit μέθοδος για την αλλαγή των στοιχείων του χρήστη, διαβάζει τα υπάρχοντα στοιχεία, και η update στέλνει τα νέα στοιχεία στο μοντέλο.
- SessionsController. Ασχολείται με την σύνδεση του χρήστη στην εφαρμογή. Με την προσπάθεια εισόδου εκτελείται η μέθοδος new. Δημιουργείται μια νέα σύνδεση με την create. Στην έξοδο εκτελείται η destroy.
- LocationsController. Ο ελεγκτής των τοποθεσιών έχει τις μεθόδους new και create για την προσθήκη νέας τοποθεσίας. Χρησιμοποιούνται οι μέθοδοι edit και update για την ενημέρωση των στοιχείων του σημείου. Με την destroy διαγράφουμε κάποιο που υπάρχει ήδη. Η show μας βοηθάει να βλέπουμε το κάθε σημείο τι πληροφορίες έχει και συνδέει τις εικόνες και τα σχόλια του σημείου. Τελικά με την μέθοδο list συμπληρώνουμε τον χάρτη με σημεία.
- CommentsController. Αυτός ο ελεγκτής έχει δύο μεθόδους. Την create για την δημιουργία του σχολίου και την destroy για την διαγραφή του.
- PicturesController. Και εδώ υπάρχουν οι μέθοδοι new και create για την εισαγωγή εικόνας στην βάση δεδομένων. Υπάρχει η destroy για την διαγραφή της. Και η μέθοδος show για την προβολή της, Την ζητάει από το μοντέλο και αυτό την επιστρέφει. Εδώ όμως στην βάση δεδομένων δεν αποθηκεύονται οι ίδιες οι εικόνες αλλά η διαδρομή για το που είναι αποθηκευμένες στο υπολογιστικό σύστημά μας. Καθώς με το upload της εικόνας η βιβλιοθήκη που χρησιμοποιούμε paperclip (18), και με χρήση του

εργαλείου `imagemagik` (19), αποθηκεύει στο σύστημα την εικόνα, όπως την ανεβάσαμε, δημιουργεί μια μικρογραφία της και μία ακόμα με μέγεθος τουλάχιστον 600pixel στο πλάτος, χωρίς να πειραχθεί η αναλογία της αρχικής εικόνας.

Model

Σε επίπεδο μοντέλου η εφαρμογή μας, λόγω του τρόπου που λειτουργεί η ίδια η ruby on rails, δεν χρειάζεται και πολλά για να ξεκινήσει να λειτουργεί. Πρωτίστως έπρεπε να εγκαταστήσουμε το ανάλογο gem (βιβλιοθήκη της ruby) για την MariaDB

```
gem install mysql2
```

και θα κάνουμε το αρχείο database.yml

```
development:
  adapter: mysql2
  encoding: utf8
  collation: utf8_general_ci
  database: bill_ptyxiakh_development
  pool: 5
  username: root
  password: ptyxiakh
  socket: /var/run/mysqld/mysqld.sock
```

```
# Warning: The database defined as "test" will be erased and
# re-generated from your development database when you run
"rake".
```

```
# Do not set this db to the same as development or production.
```

```
test:
  adapter: mysql2
  encoding: utf8
  collation: utf8_general_ci
  database: bill_ptyxiakh_test
  pool: 5
  username: root
```

```
password: ptyxiakh  
socket: /var/run/mysqld/mysqld.sock
```

```
production:  
  adapter: mysql2  
  encoding: utf8  
  collation: utf8_general_ci  
  database: bill_ptyxiakh_production  
  pool: 5  
  username: root  
  password: ptyxiakh  
  socket: /var/run/mysqld/mysqld.sock
```

Με αυτές τις ρυθμίσεις λέμε στο μοντέλο ότι για να συνδεθεί στην βάση μας θα χρειαστεί την συγκεκριμένη βιβλιοθήκη, την κωδικοποίηση, το όνομα της βάσης, το όνομα χρήστη για την σύνδεση και τον κωδικό για τον συγκεκριμένο χρήστη.

Το μοντέλο μας, όπως φαίνεται και στην Εικόνα 6, αποτελείται από τις εξής οντότητες:

- Users. Είναι η οντότητα που ασχολείται με τους χρήστες. Στην περίπτωση μας είναι ένας ο χρήστης. Έχει τα στοιχεία του, τον κωδικό του και τις τελευταίες αποθηκευμένες πληροφορίες στο cookie του χρήστη.
- Locations. Βρίσκουμε εδώ τις τοποθεσίες που βλέπουμε και στον χάρτη. Διαχειρίζεται το όνομα, τις συντεταγμένες, τις πληροφορίες και το Youtube ID.
- Comments. Με αυτή την οντότητα έχουμε τα σχόλια που αναρτώνται στις διάφορες τοποθεσίες. Τα στοιχεία της είναι το, όνομα του σχολιαστή, ο

τίτλος που βάζει και το κυρίως κείμενο του σχόλιου. Επικοινωνεί με το εξωτερικό κλειδί, `location_id`, με την οντότητα `location`.

- Pictures. Διαχειρίζεται τις εικόνες που έχουν οι διάφορες τοποθεσίες. Επικοινωνεί με το εξωτερικό κλειδί, `location_id`, με την οντότητα `location`.

Views

Η εφαρμογή μας σε επίπεδο προβολής περιέχει τις δυνατότητες που φαίνονται στην Εικόνα 7.

Παρακάτω θα αναλύσουμε κάθε περίπτωση ώστε κάθε χρήστης να μπορεί να ξέρει το τι μπορεί να κάνει. Βλέπουμε ότι δίνεται η δυνατότητα να πάμε και στις αυτόματα δημιουργημένες αλλά και στις δικές μας διαδρομές.

Το αρχείο ρυθμίσεων `routes.rb` περιέχει τις διαδρομές εντός της εφαρμογής:

```
./config/routes.rb
```

```
BillPtyxiakh::Application.routes.draw do
  root 'static_pages#home'

  match '/about', to: 'static_pages#about', via: 'get'
  match '/contact', to: 'static_pages#contact', via: 'get'
  match '/map', to: 'locations#list', via: 'get'
  resources :locations
  resources :users
  resources :comments, only: [:create, :destroy]
  resources :pictures
  resources :sessions, only: [:new, :create, :destroy]
  match '/signin', to: 'sessions#new', via: 'get'
  match '/signout', to: 'sessions#destroy', via: 'delete'
  match '/signup', to: 'users#new', via: 'get'

end
```

Όπως φαίνεται έχουμε αντιστοιχίσει κάποια ονόματα σε κάποιες διαδρομές για την ευκολία κατανόησης πχ το `/signin` με την `'sessions#new'` για την σύνδεση του χρήστη.

Για την προβολή των σημείων στον χάρτη χρησιμοποιήσαμε το gem gmap4rails (20).

Για την πιο όμορφη προβολή χρησιμοποιήσαμε το gem purecss-rails (21). Αυτό ενσωματώνει στην προβολή την CSS3 λύση purecss (22). Είναι πολύ ελαφριά και παίζει πολύ ωραία και σε mobile φυλλομετρητές, με το side-menu layout που ενσωματώσαμε, όπως φαίνεται και στην Εικόνα 8 σε σύγκριση με την Εικόνα 9.

Όλη η εφαρμογή μας έχει κάποια σταθερά στοιχεία. Αυτά είναι το τέλος της προβολής με δύο συνδέσμους, ο ένας για τις πληροφορίες σχετικά για την εφαρμογή που περιηγούμαστε και μία για την σελίδα επικοινωνίας. Άλλο σταθερό είναι η κεφαλή, όπου αποτελείται από συνδέσμους για της σχετικές πληροφορίες, για την σελίδα επικοινωνίας, την αρχική σελίδα, την σελίδα με τον χάρτη και τέλος την σελίδα εισόδου χρήστη, εάν δεν είσαι συνδεδεμένος. Αν είσαι συνδεδεμένος (βλ. αργότερα Εικόνα 13), εμφανίζει μια πτυσσόμενη επιλογή για διαχείριση του λογαριασμού μας και για έξοδο. Αυτά τα στοιχεία προβάλλονται σε κάθε σελίδα της εφαρμογής μας.

Η εφαρμογή μας ξεκινάει με την αρχική σελίδα που, όπως φαίνεται και στις Εικόνα 8 και Εικόνα 9, έχει τον τίτλο της και έναν σύνδεσμο, μέσω κουμπιού για τον χάρτη, που είναι η καρδιά του συστήματος.

Εικόνα 9 ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΕΦΑΡΜΟΓΗΣ ΑΠΟ ΣΥΣΚΕΥΗ ΑΠΟ ANDROID

Στην σελίδα που περιέχει πληροφορίες σχετικά με την εφαρμογή μας βλέπουμε στην Εικόνα 10 ότι περιέχει κάποιες βασικές πληροφορίες.

Εικόνα 10 Η ΣΕΛΙΔΑ ΣΧΕΤΙΚΑ

Η σελίδα της επικοινωνίας περιέχει συνδέσμους για την αποστολή ηλεκτρονικού μηνύματος στον φοιτητή και τον επιβλέπων καθηγητή.

Την σελίδα που περιέχει τον χάρτη βλέπουμε στην Εικόνα 12, όπου περιέχει έναν χάρτη που τροφοδοτείται από την υπηρεσία maps της google. Σημειωμένα στον χάρτη βλέπουμε σημεία ενδιαφέροντος στην περιφέρεια της ηπείρου. Επίσης μπορούμε να δούμε και το κουμπί για την δυνατότητα προσθήκης νέου σημείου, που εμφανίζεται μόνο σε συνδεδεμένο χρήστη.

Εικόνα 12 Η ΣΕΛΙΔΑ ΜΕ ΤΟΝ ΧΑΡΤΗ

Επιλέγοντας κάποιο από τα σημεία μας εμφανίζει πληροφορίες για το σημείο, όπως το όνομα του και διάφορα σχετικά στοιχεία. Επίσης υπάρχει σύνδεσμος για σελίδα που θα έχει περισσότερες πληροφορίες.

Εικόνα 13 Η ΣΕΛΙΔΑ ΜΕ ΤΟΝ ΧΑΡΤΗΣ ΜΕ ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ ΤΟ ΣΗΜΕΙΟ

Επιλέγοντας τον σύνδεσμο “περισσότερες πληροφορίες” μας πηγαίνει στην σελίδα του σημείου που όπως εμφανίζεται στην Εικόνα 14, περιέχει πληροφορίες για το σημείο ενδιαφέροντος, ένα χάρτη που δείχνει μόνο το σημείο αυτό, όπως επίσης και μία τύπου ακορντεόν λίστα με πληροφορίες. Οι πληροφορίες αυτές είναι εικόνες, βίντεο από την υπηρεσία Youtube της Google και η δυνατότητα ανάγνωσης και προσθήκης σχολίου. Στο τέλος της κύριας σελίδας βλέπουμε τρία κουμπιά, το πρώτο για επιστροφή στον χάρτη, το δεύτερο για να τροποποιήσουμε το σημείο και τέλος ένα κουμπί για την διαγραφή του σημείου. Τα δύο τελευταία κουμπιά εμφανίζονται μόνο σε συνδεδεμένου χρήστες.

Όταν επιλεγθεί η προβολή των εικόνων μπορούμε να δούμε (Εικόνα 15) μικρογραφία αυτών που έχουν προστεθεί. Κάνοντας κλικ σε κάποια από τις εικόνες εμφανίζεται σε μέγεθος μεγαλύτερο από 600pixels στο ύψος και μπορούμε να περιηγηθούμε και στις υπόλοιπες εικόνες, όλες σε παρόμοιο μέγεθος.

Εικόνα 15 Η ΠΡΟΒΟΛΗ ΤΩΝ ΕΙΚΟΝΩΝ

Με την επιλογή του συνδέσμου “Παρουσίαση Youtube” μας δίνεται η δυνατότητα προβολής του βίντεο (Εικόνα 16) από την υπηρεσία σε ενσωματωμένο περιβάλλον.

Εικόνα 16 Η ΠΡΟΒΟΛΗ ΤΩΝ ΒΙΝΤΕΟ ΤΟΥ YOUTUBE

Στην επιλογή της προβολής των σχολίων βλέπουμε (Εικόνα 17) ένα περιβάλλον όπου μπορούμε να διαβάσουμε αναρτημένα σχόλια ή να προσθέσουμε κάποιο σχόλιο. Υπάρχει ο ακριβής αριθμός των σχολίων και τα ήδη αναρτημένα σχόλια είναι σελιδοποιημένα, κάθε σελίδα έχει 5 σχόλια.

Πολλά από τα σημεία με τον χρόνο αλλάζουν έτσι κι εμείς μπορούμε να τροποποιήσουμε το κάθε σημείο. Εκτός από το ότι μπορούμε να αλλάξουμε το όνομα, μπορούμε να διορθώσουμε τις πληροφορίες, να αλλάξουμε το σημείο που προβάλλεται στον χάρτη, τον σύνδεσμο για την υπηρεσία Youtube και να προσθέσουμε εικόνες. Εδώ (Εικόνα 18) μπορούμε να σβήσουμε κάποια από τις υπάρχουσες εικόνες, με την διαφορά ότι ενώ τα υπόλοιπα τροποποιούνται μετά το πάτημα του κουμπιού “Ενημέρωση σημείου” οι εικόνες διαγράφονται άμεσα. Θέλει προσοχή το πότε πατάμε για διαγραφή της εικόνας, για ασφάλεια ρωτάει πριν την διαγραφή.

Με την επιλογή της διαγραφής του σημείου μας βγάζει ερώτημα “Είσαι σίγουρος;” (Εικόνα 19), με την επιβεβαίωση μας διαγράφει το σημείο.

Εικόνα 19 ΤΟ ΜΕΝΟΥ ΕΠΕΒΕΒΑΙΩΣΗΣ ΔΙΑΓΡΑΦΗΣ

Επειδή η ανάγκη προσθήκης νέου σημείου (Εικόνα 20) είναι σημαντική, η σελίδα όπου δημιουργούμε νέο σημείο περιμένει να συμπληρώσουμε το όνομα του σημείου, τις συντεταγμένες για προβολή στον χάρτη, πληροφορίες για το σημείο (τύπου κείμενο με μέγιστο μέγεθος 255 χαρακτήρες), το Youtube ID από βιντεάκι της υπηρεσίας σχετικά με το σημείο για την προβολή του από την εφαρμογή μας και έχουμε δυνατότητα επιλογής εικόνων από τοπικά αρχεία, έχουμε δυνατότητα πολλαπλής επιλογής στο μενού που θα ανοίξει. Με την επιλογή του κουμπιού “Προσθήκη σημείου”, ανάλογα με το μέγεθος των εικόνων, ίσως υπάρξει μια μικρή καθυστέρηση στην ανταπόκριση της εφαρμογής, καθώς αποστέλλει τις εικόνες.

Εικόνα 20 Η ΣΕΛΙΔΑ ΤΗΣ ΠΡΟΣΘΗΚΗΣ ΣΗΜΕΙΟΥ

Ο συνδεδεμένος χρήστης (Εικόνα 21) μπορεί να τροποποιήσει τις ρυθμίσεις του, όπως το όνομα, την διεύθυνση ηλεκτρονικού ταχυδρομείου του ή τον κωδικό του. Δεν έχει δικαίωμα ένας χρήστης να επηρεάσει τα στοιχεία άλλου.

Εικόνα 21 Η ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΗΣ ΧΡΗΣΤΗ

Τη φόρμα δημιουργίας νέου χρήστη (Εικόνα 22), μόνο συνδεδεμένος χρήστης μπορεί να την δει, είναι απλή αλλά περιέχει ότι χρειάζεται, δηλαδή ζητάει να εισαχθούν το όνομα, η διεύθυνση ηλεκτρονικού ταχυδρομείου, ο κωδικός και ο κωδικός ξανά για την επιβεβαίωση του, μάλιστα ο κωδικός θα πρέπει να είναι τουλάχιστον 6 χαρακτήρες.

Παρατηρήσεις κατά τη διάρκεια της ανάπτυξης του συστήματος.

Προβλήματα

Το κύριο πρόβλημα που υπήρξε ήταν το που θα φιλοξενούνταν ο εξυπηρετητής, καθώς η πρώτη λύση που είχε επιλεγθεί το σύστημα oceanos του gnet δεν ευδοκίμησε.

Ένα επόμενο πρόβλημα ήταν ότι συνήθως οι δωρεάν υπηρεσίες δεν είναι πλήρης και προσφέρουν μικρά συστήματα, σαν αυτό που επιλέξαμε. Ήταν όμως επαρκές για την εργασία μας, παρότι εμφανίστηκε δυσκολία επικοινωνίας με το περιβάλλον διαχείρισης.

Άλλο πρόβλημα είναι ότι οι περισσότεροι φυλλομετρητές, εκτός της Google, δεν εμφανίζουν τον χάρτη της υπηρεσίας Maps της Google για λόγους ασφαλείας, όπως λένε, και χρειάζεται να γίνει ανανέωση της σελίδας κατά την πρώτη επίσκεψη της.

Εν τέλει, επιλογή μας ως προς τα εργαλεία ανάπτυξης του περιβάλλοντος λειτουργίας αποδείχτηκε σωστή, καθώς το αποτέλεσμα τείνει να είναι αυτό που απαιτούνταν. Συνεπώς τα ανωτέρω προβλήματα μπόρεσαν να ξεπεραστούν. Η δυνατότητα μελλοντικής συνέχισης του συστήματος δεν εμπεριέχει πλέον καμία βασική δυσκολία.

Πιθανές μελλοντικές προσθήκες

Το σύστημα σχολίων είναι ανοικτό σε όλους και αυτό ίσως να δημιουργήσει πρόβλημα μη ελέγχου του τί αναρτάται. Δηλαδή θα έπρεπε είτε να υπάρχει έλεγχος γύρω από το τί γράφει ο κάθε ένας, με έγκριση πριν από την ανάρτηση. Είτε να δημιουργούνται πολλοί χρήστες με τη δυνατότητα ανάρτησης σχόλιου, ενώ ταυτόχρονα ο απλός επισκέπτης να μην μπορεί να προσθέσει κάποιο. Η τελευταία λύση απαιτεί και χρήστη διαχειριστή.

Επιπλέον βελτίωση θα αποτελούσε η πιο πλήρης περιγραφή των σημείων τουριστικού ενδιαφέροντος, με την ύπαρξη περισσότερων κατηγοριών στις πληροφορίες του σημείου, ανάλογες φυσικά με το είδος του σημείου. Τέτοιο παράδειγμα θα ήταν οι ώρες κοινού σε μουσεία, τα δρομολόγια λεωφορείων για πρόσβαση σε αρχαιολογικούς και φυσικού ενδιαφέροντος χώρους, κ.α.

Βιβλιογραφία

1. [Ηλεκτρονικό] http://www.espa.gr/el/Documents/Hpeirus_2014GR16M2OP004_1_3_el.pdf.
2. [Ηλεκτρονικό] http://el.wikipedia.org/wiki/%CE%A3%CF%8D%CF%83%CF%84%CE%B7%CE%BC%CE%B1_%CE%93%CE%B5%CF%89%CE%B3%CF%81%CE%B1%CF%86%CE%B9%CE%BA%CF%8E%CE%BD_%CE%A0%CE%BB%CE%B7%CF%81%CE%BF%CF%86%CE%BF%CF%81%CE%B9%CF%8E%CE%BD.
3. [Ηλεκτρονικό] <http://gic.geog.mcgill.ca/>.
4. **Β., Κόλλια – Κουσούρη.** *Γεωγραφικά Πληροφοριακά Συστήματα*. σ.λ. : Πανεπιστημιακές σημειώσεις , Γ.Π.Α, 2003.
5. [Ηλεκτρονικό] <http://www.kln.ac.lk/socialsciences/depts/geography/>.
6. [Ηλεκτρονικό] <http://academics.epu.ntua.gr/LinkClick.aspx?fileticket=WFYHpKkSy0U%3D&tabid=385&mid=2241> .
7. [Ηλεκτρονικό] <http://uml.org/>.
8. **Αρετάκη, Αικατερίνη.** [Ηλεκτρονικό] http://nemertes.lis.upatras.gr/jspui/bitstream/10889/2060/1/AretakiAikaterini_Thesis.pdf.
9. [Ηλεκτρονικό] <http://db-engines.com/en/system/MariaDB%3BMicrosoft+SQL+Server%3BMySQL>.
10. [Ηλεκτρονικό] <http://www.langpop.com/>.
11. [Ηλεκτρονικό] <http://www.rubyenterpriseedition.com/comparisons.html>.
12. [Ηλεκτρονικό] <https://anturis.com/blog/nginx-vs-apache/>.
13. [Ηλεκτρονικό] <http://www.itworld.com/article/2695421/consumerization/choosing-a-linux-web-server--nginx-vs--apache.html>.
14. [Ηλεκτρονικό] <https://aws.amazon.com/> .
15. [Ηλεκτρονικό] <http://www.ubuntu.com/>.
16. [Ηλεκτρονικό] <http://wiki.x2go.org/doku.php>.
17. [Ηλεκτρονικό] <https://rvm.io/>.
18. [Ηλεκτρονικό] <https://mariadb.org/en/>.
19. [Ηλεκτρονικό] <http://www.apache.org/>.
20. [Ηλεκτρονικό] <https://www.phusionpassenger.com/>.

21. [Ηλεκτρονικό] <https://github.com/thoughtbot/paperclip>.
22. [Ηλεκτρονικό] <http://www.imagemagick.org/>.
23. [Ηλεκτρονικό] <https://github.com/apneadiving/Google-Maps-for-Rails>.
24. [Ηλεκτρονικό] <https://github.com/brunohenrique/purecss-rails>.
25. [Ηλεκτρονικό] <http://purecss.io/>.