



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ  
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ

**Τήρηση ταξιδιωτικού ημερολογίου σε κινητή συσκευή**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

**ΧΡΗΣΤΟΥ ΤΣΕΛΙΑ**

**Επιβλέπων :** Ιωάννης Θεοδωρίδης  
Καθηγητής Πανεπιστημίου Πειραιώς

Αθήνα, Ιούνιος 2015

---

*(Υπογραφή)*

.....

**ΧΡΗΣΤΟΣ ΤΣΕΛΙΑΣ**

Πτυχιούχος Πληροφορικής Ο.Π.Α.

© 2015 – All rights reserved

## Περίληψη

Ο σκοπός της μεταπτυχιακής εργασίας ήταν η υλοποίηση ενός συστήματος, στο οποίο μέσω μιας εφαρμογής για κινητές συσκευές ο χρήστης θα μπορούσε να τηρεί ένα ηλεκτρονικό ταξιδιωτικό ημερολόγιο.

Συγκεκριμένα, η εφαρμογή υλοποιήθηκε για κινητές συσκευές που έχουν ενσωματωμένο λειτουργικό σύστημα android. Μέσα από την εφαρμογή ο χρήστης έχει τη δυνατότητα να καταγράφει ηλεκτρονικά τα ταξίδια του και να τα απεικονίζει πάνω σε χάρτες. Μπορεί να δημιουργεί διαδρομές σε κάθε ταξίδι του, να κάνει επισκέψεις, να κρατά σημειώσεις και να βγάζει φωτογραφίες από σημεία που τον ενδιαφέρουν. Το σύστημα εκτός από την εφαρμογή της κινητής συσκευής περιλαμβάνει και έναν κεντρικό database server, στον οποίο ο χρήστης μπορεί να κοινοποιεί δεδομένα από τα ταξίδια και να τα μοιράζεται με άλλους χρήστες της εφαρμογής. Η κοινοποίηση των δεδομένων γίνεται μέσω web services, τα οποία αποτελούν συστατικό του συστήματος που υλοποιήθηκε. Έτσι στην συγκεκριμένη εργασία παρουσιάζονται: συγγενικές εφαρμογές, η αρχιτεκτονική του συστήματος, τα λογισμικά και τα εργαλεία για την υλοποίηση του συστήματος και ένα ολοκληρωμένο σενάριο χρήσης της εφαρμογής.

**Λέξεις Κλειδιά:** Ταξιδιωτικό ημερολόγιο, πληθοπορισμός, χωρικός πληθοπορισμός, κινητή συσκευή, χάρτες, χωρικά δεδομένα, γεωγραφική θέση.



## **Abstract**

The scope of this thesis was to implement a system, in which the user could keep an online travel blog through a mobile application.

Specifically, a mobile application was implemented for devices that have built-in android operating system. The user has the ability to record his trip in the mobile application and see it in global maps. He can create routes on each trip, he can visit, take notes and take pictures from points of interest. In addition, the system includes a central database server, in which the user can transfer the data from his trip and share those data with other users of the application. The data of the trip are transferred through web services, which are another component of the system that was implemented. Thus, in this thesis are presented: related applications, the system architecture, software and tools that are used for system implementation and a user scenario of the application.

**Keywords:** Travel blog, crowdsourcing, spatial crowdsourcing, mobile device, maps, spatial data, geolocation



# Πίνακας περιεχομένων

<b>1 Εισαγωγή.....</b>	<b>1</b>
1.1 Αντικείμενο και σκοπός της εργασίας.....	1
1.2 Οργάνωση κειμένου .....	4
<b>2 Συγγενικές εφαρμογές και εργασίες.....</b>	<b>5</b>
2.1 Συγγενικές εφαρμογές .....	6
2.2 Σχετικές Εργασίες.....	13
2.2.1 Πληθοπορισμός.....	13
2.2.2 Χωρικός Πληθοπορισμός.....	15
2.2.3 Ποιότητα δεδομένων από χωρικό πληθοπορισμό.....	17
2.2.4 Εφαρμογές πληθοπορισμού και χωρικού πληθοπορισμού.....	20
2.2.4.1 SmartTrace.....	20
2.2.4.2 Crowdcast.....	21
2.2.4.3 SmartP2P.....	22
2.2.4.4 Πλατφόρμα πληθοπορισμού βασισμένη στην γεωγραφική θέση.....	23
2.2.4.5 gMission: Μια πλατφόρμα χωρικού πληθοπορισμού.....	24
2.2.4.6 Αξιοποίηση του Flickr API στη διερεύνηση τοπωνυμίων.....	25
<b>3 Αρχιτεκτονική συστήματος .....</b>	<b>27</b>
3.1 Αρχιτεκτονικό διάγραμμα .....	29
3.2 Τοπική βάση δεδομένων.....	31
3.3 Αισθητήρες συσκευής.....	32
3.4 Χάρτες.....	32
3.5 Database Server.....	33
3.6 Web Services.....	34
3.7 Διεπαφές χρήστη.....	34
<b>4 Υλοποίηση συστήματος.....</b>	<b>37</b>
4.1 Android λειτουργικό σύστημα συσκευής.....	39
4.2 Περιβάλλον ανάπτυξης Android Studio .....	39
4.3 SQLite με την επέκταση Spatialite .....	41
4.3.1 SQLite.....	41

4.3.2	<i>Spatialite</i> .....	42
4.4	PostgreSQL με την επέκταση PostGIS.....	43
4.4.1	<i>PostgreSQL</i> .....	43
4.4.2	<i>Πλήρης αναζήτηση κειμένων στην PostgreSQL</i> .....	43
4.4.3	<i>PostGIS</i> .....	46
4.5	Microsoft .NET Web Services.....	46
4.6	Χάρτες Google και OpenStreetMap.....	47
4.7	Δημιουργία τοπικής βάσης δεδομένων.....	49
4.8	Δημιουργία κεντρικής βάσης δεδομένων.....	52
4.9	Δημιουργία Web services.....	53
<b>5</b>	<b>Υπόδειγμα χρήσης της εφαρμογής</b> .....	<b>56</b>
<b>6</b>	<b>Επίλογος</b> .....	<b>68</b>
6.1	Συμπεράσματα.....	68
6.2	Μελλοντικές επεκτάσεις.....	70
<b>7</b>	<b>Βιβλιογραφία</b> .....	<b>71</b>
<b>8</b>	<b>Παραρτήματα</b> .....	<b>74</b>
8.1	Διαχείριση SQLite – Spatialite.....	74
8.2	Διαχείριση PostgreSQL.....	76
8.3	Web Services.....	79



# 1

## Εισαγωγή

### 1.1 Αντικείμενο και σκοπός της εργασίας

Ο στόχος της εργασίας είναι να δημιουργηθεί ένα σύστημα στο οποίο, μέσω μιας εφαρμογής για κινητές συσκευές ο χρήστης θα μπορεί να τηρεί ένα ηλεκτρονικό ημερολόγιο των ταξιδιών του. Τα προηγούμενα χρόνια, πριν η τεχνολογία φτάσει στο σημείο που είναι σήμερα, όσοι αγαπούσαν τα ταξίδια, κρατούσαν σημειώσεις από τα μέρη που επισκέπτονταν σε κάποιο ημερολόγιο, συνοδευόμενες πολλές φορές από κάποιες εκτυπωμένες φωτογραφίες. Άλλοι πάλι δεν κρατούσαν καν σημειώσεις, παρά μόνο κάποιες φωτογραφίες και έτσι οι αναμνήσεις χάνονταν με το πέρασμα των χρόνων. Πολλοί ταξιδιώτες επέστρεφαν σε μέρη που είχαν πάει πριν χρόνια και δεν θυμούνταν σχεδόν τίποτα, ούτε μπορούσαν να καταλάβουν τι έχει αλλάξει από την τελευταία τους επίσκεψη, καθώς τους ήταν δύσκολο να συγκρίνουν τις δυο περιόδους. Θα έπρεπε κάθε φορά να ανατρέχουν σε παλιές φωτογραφίες και σελίδες απ' το ημερολόγιό τους.

Η τεχνολογική εξέλιξη ήρθε να τα αλλάξει όλα αυτά. Αρχικά με το *Διαδίκτυο*, τα *κινητά τηλέφωνα* αλλά και τις *ψηφιακές κάμερες*. Οι φωτογραφίες άρχισαν να αποθηκεύονται ψηφιακά και να μην χρειάζεται να τις εκτυπώσει ο ταξιδιώτης. Έτσι ο ταξιδιώτης μπορούσε να τις έχει πάντα μαζί του σε κάποια ηλεκτρονική συσκευή, π.χ. αρχικά με τους φορητούς υπολογιστές και στη συνέχεια με τις υπόλοιπες κινητές συσκευές που βγήκαν αργότερα. Έτσι καθώς οι τεχνολογίες και οι τηλεπικοινωνίες εξελίσσονταν με γρήγορους ρυθμούς, άρχισαν να δημιουργούνται τα πρώτα ηλεκτρονικά ταξιδιωτικά ημερολόγια σε μορφή ιστοσελίδων και ιστολογίων. Σε αυτά τα ηλεκτρονικά ημερολόγια οι ταξιδιώτες μπορούσαν να γραφτούν και να γίνουν μέλη μιας κοινότητας. Έτσι, έχοντας από οπουδήποτε πρόσβαση στο Διαδίκτυο μπορούσαν να συνδεθούν στην ιστοσελίδα του ημερολογίου και να αποθηκεύσουν σχόλια και φωτογραφίες από τις επισκέψεις των ταξιδιών τους. Ακόμα και έτσι υπήρχε το πρόβλημα της

αποθήκευσης των εντυπώσεων από τα ταξίδια τους σε πραγματικό χρόνο. Αυτό γιατί ο ταξιδιώτης δεν είχε παντού πρόσβαση στο Διαδίκτυο από τα μέρη που επισκεπτόταν ή ακόμα και αν είχε δεν μπορούσε να έχει πάντα μαζί του το φορητό υπολογιστή για να αποθηκεύει τις επισκέψεις του από τα αγαπημένα του μέρη στην ιστοσελίδα του ηλεκτρονικού ημερολογίου. Έτσι έπρεπε πρώτα να κρατεί σημειώσεις και φωτογραφίες κάπου προσωρινά και στη συνέχεια όταν βρει διαθέσιμη ευρυζωνική σύνδεση να συνδεθεί στην ιστοσελίδα και να ανεβάσει τις φωτογραφίες και τα σχόλια από τις επισκέψεις του. Μια τέτοια χαρακτηριστική ιστοσελίδα είναι το trip Advisor, όπου καθένας μπορεί να γίνει μέλος και να μοιραστεί τις εμπειρίες του από τα ταξίδια του. Στο trip advisor κάθε μέλος μπορεί να ανεβάζει σχόλια και φωτογραφίες από τις επισκέψεις του από τα ταξίδια του, ώστε να τα βλέπουν και άλλοι χρήστες που πιθανόν θέλουν να επισκεφτούν το ίδιο μέρος.

Μια άλλη μεγάλη τεχνολογική εξέλιξη που βοήθησε να αναπτυχθούν όλα αυτά τα ηλεκτρονικά ταξιδιωτικά ημερολόγια ήταν οι *διαδικτυακοί χάρτες*. Οι χάρτες μέσω κυρίως της Google αλλά και άλλων εταιρειών εξελίχθηκαν και ενημερώθηκαν με όλες τις περιοχές της γης. Έτσι κάποιος μπορεί να εντοπίσει στο χάρτη τη θέση του ή το σημείο που θέλει να επισκεφτεί και να βάλει έναν σελιδοδείκτη πάνω στον χάρτη, ώστε να ξέρει πού ήταν ακριβώς το σημείο που επισκέφτηκε. Και αυτό όμως δεν μπορούσε να γίνει εύκολα σε πραγματικό χρόνο γιατί ο χρήστης θα πρέπει να έχει πρόσβαση στο Διαδίκτυο για να δει το χάρτη και να βρει τη θέση του ή το σημείο που τον ενδιαφέρει. Το trip Advisor αλλά και άλλα ταξιδιωτικά ημερολόγια δίνει αυτή τη δυνατότητα στα μέλη του, να εντοπίζουν τη θέση τους στον χάρτη που προσφέρεται από την Google και να βάζουν σελιδοδείκτη στην επίσκεψη τους πάνω στο χάρτη μαζί με σχόλια και φωτογραφίες..

Όλα αυτά τα προβλήματα όμως ήρθε για να λύσει η ευρεία διάδοση των «έξυπνων» κινητών τηλεφώνων. Αρχικά τα κινητά δεν κουβαλούσαν λειτουργικά συστήματα αρκετά προηγμένα ώστε να δίνουν τη δυνατότητα στους χρήστες να κάνουν ό,τι έκαναν με τους ηλεκτρονικούς υπολογιστές. Οι μνήμες που είχαν τα παλιά κινητά ήταν πολύ μικρές για να αποθηκεύει ο χρήστης οτιδήποτε, ακόμα και οι ενσωματωμένες κάμερες είχαν χαμηλή ποιότητα. Όμως, η τελευταία γενιά των κινητών το κάνει αυτό και με το παραπάνω. Επιπλέον οι αισθητήρες που έχουν ενσωματωμένους (όπως το GPS και η κάμερα τελευταίας γενιάς) συνέτειναν ώστε τα ταξιδιωτικά ημερολόγια και γενικότερα οι ταξιδιωτικές εφαρμογές να γίνουν δημοφιλείς. Πλέον η δημοτικότητα των έξυπνων κινητών είναι τόσο μεγάλη, που πολύ λίγοι είναι εκείνοι οι οποίοι δεν έχουν έξυπνο κινητό. Το γεγονός ότι μπορούν οι χρήστες να έχουν μια τέτοια ηλεκτρονική συσκευή πάντα μαζί τους, με την οποία θα μπαίνουν στο Διαδίκτυο και θα βγάζουν φωτογραφίες καλής ποιότητας ανεβάζουν τη δημοτικότητά τους. Επίσης η εξέλιξη αυτή στα κινητά και τα προηγμένα λειτουργικά συστήματα, έδωσαν τη δυνατότητα σε ελεύθερους προγραμματιστές και σε εταιρείες πληροφορικής να αναπτύξουν

τις δικές τους εφαρμογές και να τις ενσωματώσουν στο λειτουργικό σύστημα των κινητών συσκευών. Έτσι σε ένα τέτοιο ταξιδιωτικό ημερολόγιο όπως το trip Advisor, ο χρήστης πλέον μπορεί να δημιουργεί δεδομένα σε πραγματικό χρόνο. Μέσω του GPS μπορεί να εντοπίζει τη θέση του σε πραγματικό χρόνο και μέσω των ευρυζωνικών δικτύων που διαθέτουν, μπορούν σε πραγματικό χρόνο να συνδεθούν στο Διαδίκτυο στην ιστοσελίδα του ταξιδιωτικού ημερολογίου και να σημειώσουν τη θέση τους στο χάρτη ή οποιοδήποτε άλλο σημείο μαζί με φωτογραφίες και σχόλια από την επίσκεψή τους. Έτσι, οι χρήστες μπορούν να βλέπουν άμεσα τα δεδομένα που ανεβάζει κάποιος άλλος χρήστης και να βοηθηθούν στο ταξίδι τους.

Όλα τα παραπάνω αποτέλεσαν αφορμή για να δημιουργηθεί το σύστημα που περιγράφεται στη συγκεκριμένη εργασία. Λόγω των αρκετά μεγάλων αποθηκευτικών χώρων που διαθέτουν πλέον τα κινητά, ο χρήστης θα μπορεί θεωρητικά να αποθηκεύει στη συσκευή απεριόριστα ταξίδια. Σημαντικό θα ήταν ο χρήστης να μπορεί να χωρίζει το ταξίδι του σε διαδρομές. Σχεδόν κανένα ταξίδι δεν είναι μια ενιαία διαδρομή. Π.χ. άλλη διαδρομή μπορεί να κάνει ο χρήστης τη μια μέρα, άλλη την δεύτερη κ.ο.κ. Η κάθε διαδρομή αποτελείται από επισκέψεις του χρήστη σε μέρη που τον ενδιαφέρουν. Μέσω του GPS της συσκευής θα εντοπίζεται η θέση του χρήστη στο χάρτη, θα εμφανίζονται τα σημεία ενδιαφέροντος που υπάρχουν γύρω του και ο χρήστης θα επιλέγει κάποια από αυτά τα προτεινόμενα σημεία ενδιαφέροντος ή θα δημιουργεί κάποιο καινούριο και θα βάζει έναν σελιδοδείκτη στον χάρτη στο σημείο που τον ενδιαφέρει. Έτσι ο χρήστης θα μπορεί να δημιουργεί πολλές τέτοιες επισκέψεις σε κάθε διαδρομή πάνω στο χάρτη. Επιπλέον σε κάθε επίσκεψη θα ήταν χρήσιμο ο χρήστης να μπορεί να κρατά σημειώσεις για τα σημεία και να βγάζει όσες φωτογραφίες θέλει, προκειμένου να διατηρήσει λεπτομέρειες που έχουν σημασία για τον ίδιο. Έτσι θα ολοκληρώνεται και η καταγραφή του ταξιδιού του χρήστη μέσω της εφαρμογής.

Σημαντικό θα ήταν ο χρήστης να βρίσκει εύκολα το ταξίδι του που βρίσκεται σε εξέλιξη. Π. χ. αν τελειώσει μια διαδρομή τη μια μέρα και τη δεύτερη μέρα θέλει να συνεχίσει την καταγραφή, θα ήταν χρήσιμο μόλις ανοίξει ξανά την εφαρμογή με το πάτημα ενός κουμπιού να βρίσκει το εν εξέλιξη ταξίδι του. Επίσης, εφόσον ο χρήστης έχει αποθηκευμένα πολλά ταξίδια στην συσκευή του θα ήταν σημαντικό να ανατρέχει σε οποιοδήποτε ταξίδι του εύκολα, σε περίπτωση που θέλει να βρει κάποια ανάμνηση του ή και να επισκεφτεί ξανά το ίδιο μέρος. Έτσι, θα πρέπει να υπάρχει ένας εύκολος τρόπος αναζήτησης των παλιών ταξιδιών του, εντοπίζοντας τα πάνω στο χάρτη, ψάχνοντας την περίοδο που επιθυμεί ή και τα δυο μαζί.

Τέλος, για τους χρήστες θα ήταν σημαντικό να μπορούν να κοινοποιούν τα ταξίδια τους σε μια κοινότητα ανθρώπων που τους αρέσουν τα ταξίδια και ανταλλάσσουν εντυπώσεις γι' αυτά. Έτσι, θα μπορούν να βλέπουν τί έκαναν άλλοι χρήστες σε περιοχές που τους ενδιαφέρει ή που θέλουν να επισκεφτούν ή ακόμα και να βοηθήσουν άλλους χρήστες στο ταξίδι τους προσφέροντας τις δικές τους πληροφορίες για τον συγκεκριμένο προορισμό.

## **1.2 Οργάνωση κειμένου**

Στο κεφάλαιο 2 που ακολουθεί περιγράφονται σχετικές εφαρμογές σε κινητά που είναι διαθέσιμες για καταγραφή ταξιδιών. Επίσης, επειδή πρόκειται καθαρά για μια εφαρμογή πληθοπορισμού γίνεται επισκόπηση εφαρμογών πληθοπορισμού.

Στο κεφάλαιο 3 παρουσιάζεται η γενική αρχιτεκτονική του συστήματος που δημιουργείται στα πλαίσια αυτής εδώ της εργασίας και αναλύονται οι απαιτήσεις των κύριων συστατικών μερών του.

Στο κεφάλαιο 4 ακολουθεί η υλοποίηση του συστήματος. Αναλύονται οι συγκεκριμένες τεχνολογίες που χρησιμοποιήθηκαν για να δημιουργηθεί αλλά και το πώς μετατρέπεται πλέον η αρχιτεκτονική του.

Στο κεφάλαιο 5 παρουσιάζεται ένα ολοκληρωμένο σενάριο χρήστη με εικόνες για να γίνει αντιληπτό πώς λειτουργεί αυτή η εφαρμογή του συστήματος και ποιες δυνατότητες παρέχει στο χρήστη.

Τέλος, στο κεφάλαιο 6 παρουσιάζονται κάποια χρήσιμα συμπεράσματα από την δημιουργία ενός τέτοιου συστήματος καθώς και μελλοντικές επεκτάσεις του.

# 2

## *Συγγενικές εφαρμογές και εργασίες*

Υπάρχουν πολλές εφαρμογές για κινητά με τις οποίες μπορεί να διατηρήσει κανείς ένα ταξιδιωτικό ημερολόγιο, να δημοσιοποιήσει φωτογραφίες, να κοινοποιήσει αναμνήσεις του ταξιδιού με τους φίλους του, να κρατήσει στο λεύκωμα του σημαντικές επισκέψεις που έκανε σε μέρη που τον ενδιαφέρουν, να μοιραστεί αυτές τις επισκέψεις με φίλους του, να δει προτάσεις φίλων του από μέρη που τον ενδιαφέρουν και γενικά να μοιραστεί το πάθος του για ταξίδια με άλλα άτομα σαν και αυτόν. Πρόκειται για εφαρμογή πληθοπορισμού, στην οποία δίνεται στους χρήστες η δυνατότητα να συλλέξουν πληροφορίες, χωρίς να είναι εξειδικευμένο προσωπικό. Αυτές οι πληροφορίες μπορεί να είναι χωρικές και περιγραφικές. Χωρικές γιατί συλλέγουν τις συντεταγμένες μιας θέσης από τις επισκέψεις τους, αλλά και περιγραφικές γιατί μπορεί να συλλέξουν φωτογραφίες και σχόλια από αυτές τις τοποθεσίες για την κοινότητα των ταξιδιωτών. Έτσι, είναι φανερό ότι τέτοιες ταξιδιωτικές εφαρμογές για κινητά σχετίζονται με σύγχρονες τεχνικές και μεθοδολογίες διαχείρισης δεδομένων όπως:

- Πληθοπορισμός (crowdsourcing)
- Χωρικός πληθοπορισμός (spatial crowdsourcing)
- Συλλογή περιεχομένου από χρήστες (User Generated Content)

## 2.1 Συγγενικές Εφαρμογές

Αρχικά θα παρουσιαστούν κάποιες συγγενικές εφαρμογές με αυτήν που υλοποιήθηκε στα πλαίσια αυτής εργασίας:

- Travelog–City Guides & Blogs Asia By Travelog Pte Ltd.
- livemapp - photo blog travel pics By Inspirational Futures Ltd
- TripMayor - Travel Tips from Locals By pixels magic limited
- Everplaces
- TravelPod - Travel Blog By TravelPod

Πρόκειται για εφαρμογές με τις οποίες ο χρήστης μπορεί να διατηρεί ένα ταξιδιωτικό ημερολόγιο στην κινητή του συσκευή. Ο χρήστης μπορεί να γίνει μέλος μιας κοινότητας και να μοιραστεί τις ταξιδιωτικές του εμπειρίες. Μπορεί να δημιουργήσει περιεχόμενο όπως φωτογραφίες, κείμενο και γεωγραφικά σημεία στο χάρτη. Σημαντικό είναι ότι σε όλες αυτές τις εφαρμογές, αν ο χρήστης επιθυμεί μπορεί να μοιραστεί αυτό το περιεχόμενο με τα υπόλοιπα μέλη της κοινότητας στην οποία έχει γίνει μέλος, κοινοποιώντας αυτό το περιεχόμενο σε έναν κεντρικό Server. Έτσι μπορεί ανά πάσα στιγμή να πάρει πληροφορίες από άλλους χρήστες των εφαρμογών από σημεία που τον ενδιαφέρουν. Αυτό βοηθά το χρήστη να προγραμματίσει καλύτερα το ταξίδι του, ίσως να αποφύγει μέρη που δεν αξίζει να επισκεφτεί ή ακόμα και να ανακαλύψει μέρη που αρχικά δεν τα είχε συμπεριλάβει στα σχέδιά του. Κάποιες από αυτές τις εφαρμογές μπορεί να τρέχουν σε κινητά που υποστηρίζουν λειτουργικό iOS, κάποιες σε android και άλλες πάλι μπορεί να τρέχουν σε κινητά είτε υποστηρίζουν iOS, είτε android λειτουργικό σύστημα. Σε κάποιες από αυτές τις εφαρμογές ο χρήστης μπορεί να δημιουργήσει και να μοιραστεί ακόμη και βίντεο. Συνοπτικά στον παρακάτω πίνακα 1 φαίνονται τα βασικά χαρακτηριστικά αυτών των εφαρμογών και πως διαφοροποιούνται σε σχέση με την εφαρμογή **Travelogue** που υλοποιήθηκε στα πλαίσια αυτής της εργασίας.

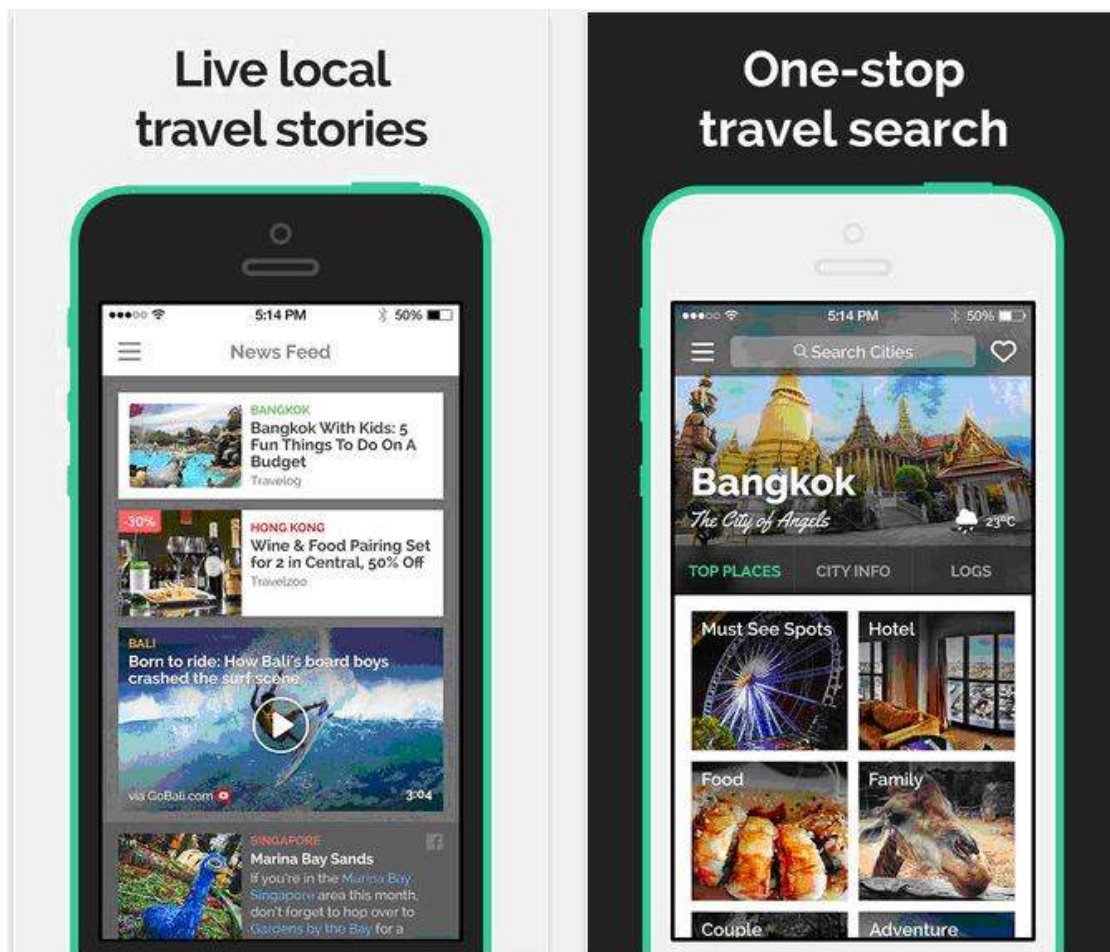
	Travelog	livemapp	TripMayor	Everplaces	TravelPod	Geospike	Travelogue
Βίντεο	✗	✗	✗	✗	✓	✗	✗
Φωτογραφίες	✓	✓	✓	✓	✓	✓	✓
Κείμενο	✓	✓	✓	✓	✓	✓	✓
Λειτουργικό	iOS	iOS	iOS/android	iOS	iOS	iOS/android	android
Κοινοποίηση σε κεντρικό Server	✓	✓	✓	✓	✓	✓	✓
Χάρτες	online	online	Online/offline	Online/offline	Online/offline	Online/offline	Online/offline

**Πίνακας 1: Βασικά χαρακτηριστικά των εφαρμογών**

**Travelog.** Ειδικά η εφαρμογή Travelog [30] έχει αναπτυχθεί για συσκευές που τρέχουν σε λειτουργικό σύστημα iOS. Η εφαρμογή παρέχει προτάσεις, σχόλια και αξιολογήσεις για ταξίδια από μεγάλες πόλεις σε όλη την Ασία. Επιπλέον περιλαμβάνονται οδηγοί πόλης με φωτογραφίες, σχόλια, blogs, προτάσεις, πληροφορίες, χάρτες, και άρθρα από ειδικούς. Οδηγίες και σημεία ενδιαφέροντος προσφέρονται από κάποιες μεγάλες πόλεις της Ασίας όπως το Πεκίνο, η Σαγκάη, η Σεούλ, η Σιγκαπούρη, η Μπανγκόκ, το Πουκέτ, η Κουάλα Λουμπόρ, το Χονγκ Κονγκ, το Τόκιο και το Μπαλί. Συνοπτικά ο χρήστης μπορεί:

- Να κάνει έρευνα και σχεδιασμό του ταξιδιού του σε ένα από τα παραπάνω μέρη.
- Να λάβει εκτιμήσεις και συστάσεις για καταλύματα, εστιατόρια, αξιοθέατα, όχι μόνο από τα υπόλοιπα μέλη της κοινότητας, αλλά και από ειδικούς.
- Να έχει πρόσβαση σε κριτικές μέσω της Google.
- Να δει δημοφιλή άρθρα από διάφορους προορισμούς.
- Να δει ενημερώσεις και προτάσεις από χρήστες που βρίσκονται κοντά.

Στην παρακάτω εικόνα 1 φαίνονται ενδεικτικά κάποια στιγμιότυπα της εφαρμογής:

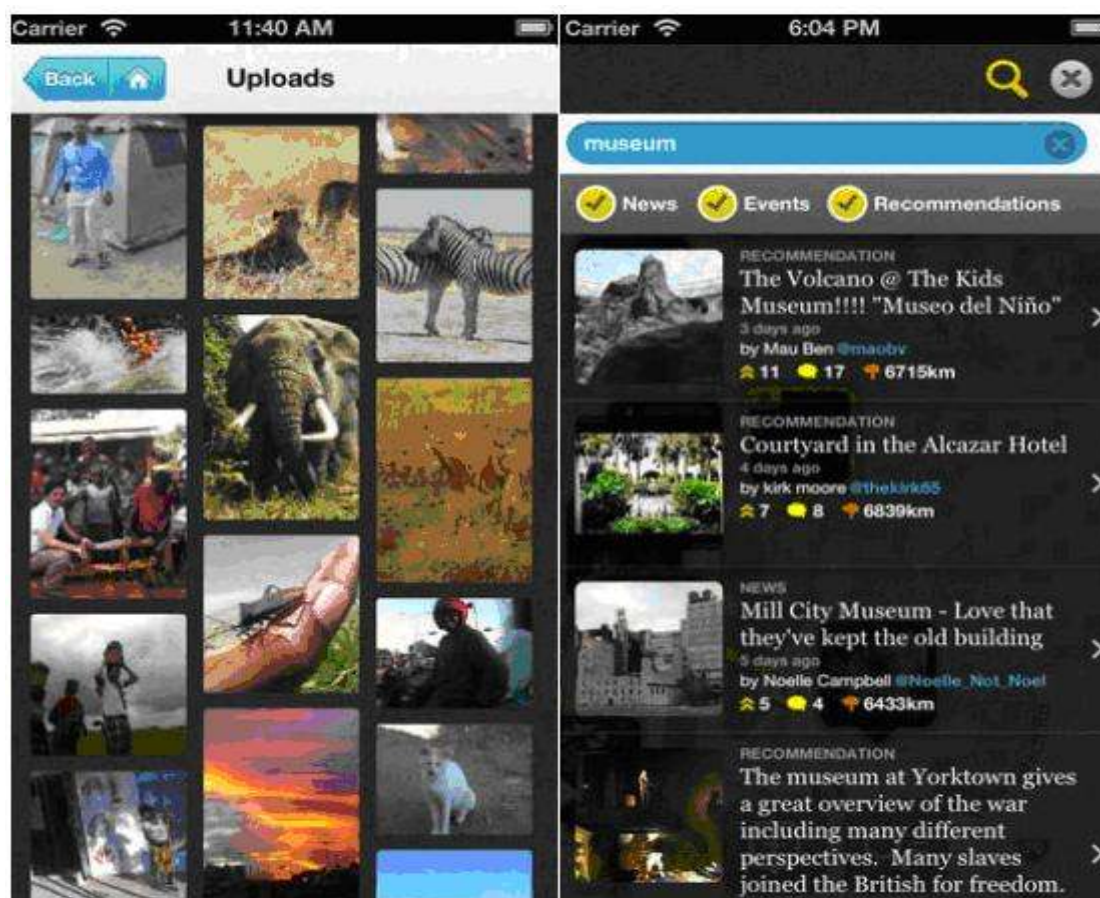


Εικόνα 1: Στιγμιότυπα της εφαρμογής [30]

**Livemapp.** Η εφαρμογή livemapp [14] έχει αναπτυχθεί για συσκευές που τρέχουν σε λειτουργικό σύστημα iOS. Προσφέρει ένα γρήγορο και διασκεδαστικό τρόπο για να μοιραστεί κάποιος χρήστης τις φωτογραφίες του όταν ταξιδεύει. Πέρα από τα βασικά χαρακτηριστικά, ο χρήστης μπορεί:

- Να ανεβάσει τις φωτογραφίες του ταξιδιού του σε ένα ζωντανό χάρτη.
- Να εξερευνήσει διαφορετικές χώρες και πολιτισμούς
- Να προγραμματίσει το ταξίδι του και να πάρει ιδέες από άλλους χρήστες που είναι κοντά του.
- Να δει μια εικόνα του τι συμβαίνει οπουδήποτε στον κόσμο σε κάποια χρονική στιγμή.
- Να κοινοποιήσει πληροφορίες στο Facebook και στο Twitter
- Να ανακτήσει φωτογραφίες με βάση τη θέση του.

Στην παρακάτω εικόνα 2 φαίνονται ενδεικτικά κάποια στιγμιότυπα της εφαρμογής:



Εικόνα 2: Στιγμιότυπα της εφαρμογής [14]



**TripMayor.** Η εφαρμογή TripMayor [33] έχει αναπτυχτεί για συσκευές που τρέχουν είτε σε iOS, είτε σε android λειτουργικό σύστημα. Με αυτήν την εφαρμογή ο χρήστης μπορεί να μοιραστεί αγαπημένες του ταξιδιωτικές εμπειρίες. Επιπλέον μπορεί να πάρει πληροφορίες ακόμη και όταν το ταξίδι του είναι εν εξελίξει από τοπικούς χρήστες ή άλλους χρήστες που βρίσκονται ανά πάσα στιγμή κοντά του. Συνοπτικά ο χρήστης:

- Μπορεί να συμβουλευτεί κάποιον τοπικό χρήστη.
- Μπορεί να ανακαλύψει μέρη άγνωστα στο ευρύ κοινό βάσει των ταξιδιωτικών οδηγιών που προσφέρονται από ειδικευμένο προσωπικό.
- Μπορεί να χρησιμοποιήσει την εφαρμογή και τους χάρτες της σε offline λειτουργία, ώστε να μην ξοδεύει MB από την σύνδεση του στο Διαδίκτυο.

Στην παρακάτω εικόνα 3 φαίνονται κάποια στιγμιότυπα της εφαρμογής:



Εικόνα 3: Στιγμιότυπα της εφαρμογής [33]

**Everplaces.** Η εφαρμογή Everplaces [7] έχει αναπτυχτεί για συσκευές που τρέχουν σε λειτουργικό σύστημα iOS. Την εφαρμογή αυτή μπορεί να τη χρησιμοποιήσει κάποιος ως εργαλείο μνήμης για τις επισκέψεις του. Βοηθάει το χρήστη να θυμηθεί παλιά του ταξίδια από μέρη που επισκέφτηκε, και να τα συστήσει σε άλλους χρήστες της κοινότητας. Αυτή η εφαρμογή μπορεί να χρησιμοποιηθεί για να βάλει ο χρήστης σελιδοδείκτες σε θέσεις που τον ενδιαφέρουν, να διαβάσει γι' αυτές τις θέσεις και να τις ανακτήσει εύκολα ανά πάσα στιγμή. Τα δεδομένα από τις διάφορες θέσεις και οι χάρτες μπορούν να διατεθούν τοπικά στη συσκευή, οπότε δεν χρειάζεται 3G για να ανακτήσει πληροφορίες για κάποια θέση. Επίσης, ο χρήστης μπορεί:

- Να δημιουργήσει τους δικούς του χάρτες και ταξιδιωτικούς οδηγούς.
- Να προσαρμόσει τους χάρτες με περιγραφές, ετικέτες και εικόνες.

Στην παρακάτω εικόνα 4 φαίνονται κάποια στιγμιότυπα της εφαρμογής:

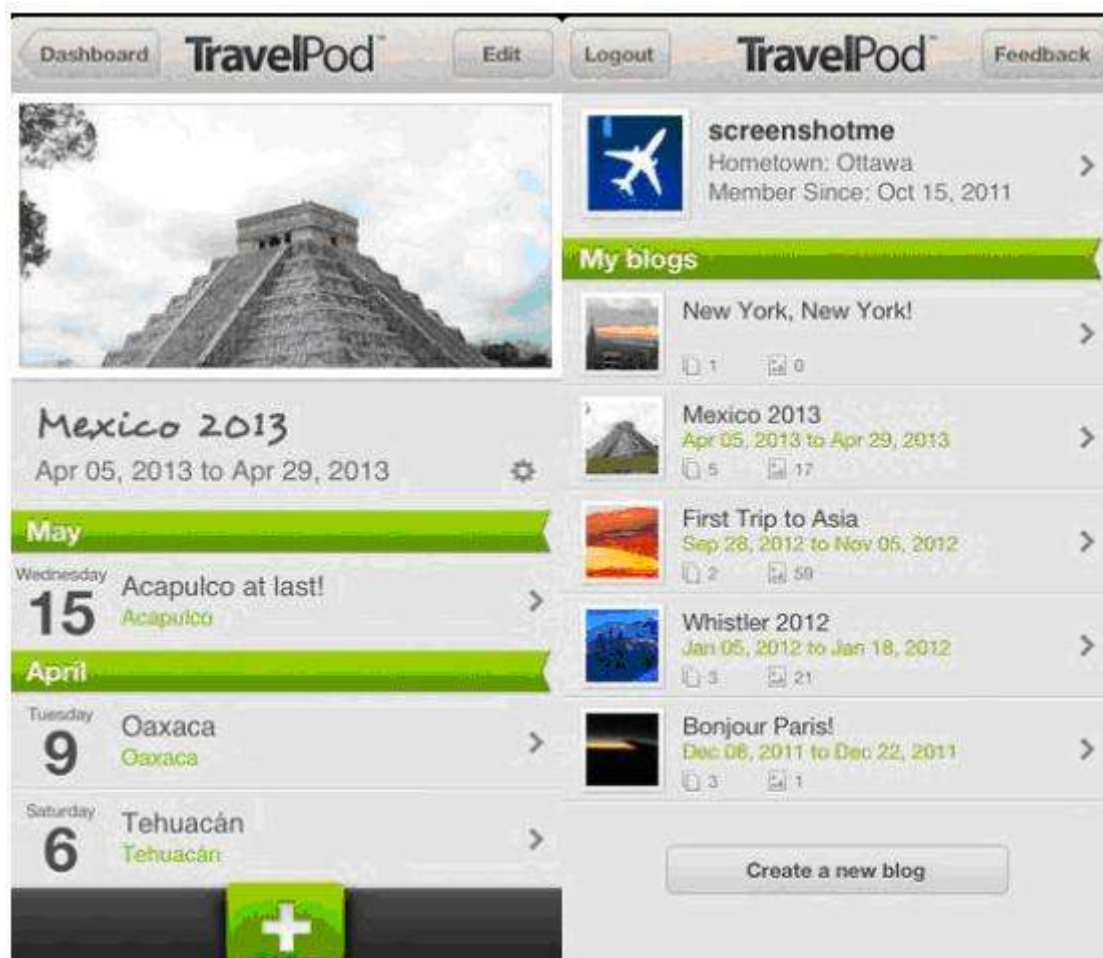


**Εικόνα 4: Στιγμιότυπα της εφαρμογής [7]**

**TravelPod.** Η εφαρμογή TravelPod [31] έχει αναπτυχθεί για συσκευές που τρέχουν σε λειτουργικό σύστημα iOS και λειτουργεί με ή χωρίς πρόσβαση στο Διαδίκτυο, ώστε να μπορεί ο χρήστης να προσθέσει και να επεξεργαστεί πληροφορίες του ταξιδιού του όπου κι αν βρίσκεται και να συγχρονίσει άψογα αυτές τις πληροφορίες όταν αποκτήσει ξανά πρόσβαση. Πέρα από φωτογραφίες ο χρήστης μπορεί να δημιουργήσει και να μοιραστεί και βίντεο. Έτσι ένας χρήστης του TravelPod μπορεί να:

- Επεξεργάζεται είτε από το site είτε από το κινητό και να συγχρονίζει εύκολα πληροφορίες που αποθηκεύονται στο κινητό και κοινοποιούνται στο site της εφαρμογής.
- Ανεβάζει βίντεο από τα ταξίδια του και να τα κοινοποιεί στο site της εφαρμογής.
- Χρησιμοποιεί την εφαρμογή είτε online είτε offline, έτσι ώστε να μην χάνει στιγμή από τα ταξίδια του.
- Να μοιραστεί τις πληροφορίες του είτε στο Facebook, είτε με e – mail.

Στην παρακάτω εικόνα 5 φαίνονται κάποια στιγμιότυπα της εφαρμογής:



Εικόνα 5: Στιγμιότυπα της εφαρμογής [31]

**Geospike.** Η εφαρμογή Geospike [9] έχει αναπτυχθεί για συσκευές που τρέχουν είτε σε iOS, είτε σε android λειτουργικό σύστημα. Το Geospike είναι ένας πολύ καλός τρόπος για να αποθηκεύσει και να μοιραστεί ο χρήστης στιγμές ενώ είναι εν κινήσει. Το καλύτερο κομμάτι αυτής της εφαρμογής είναι ότι ο χρήστης μπορεί να τα κάνει όλα αυτά χωρίς σύνδεση στο Διαδίκτυο, κάτι που είναι ιδανικό για τους ταξιδιώτες που ταξιδεύουν στο εξωτερικό, γιατί η μεταφορά δεδομένων στο εξωτερικό κοστίζει ακριβά. Στο Geospike δεν υπάρχουν προκαθορισμένα σημεία ενδιαφέροντος, παρά μόνο αυτά που δημιουργεί ο χρήστης. Έτσι ο χρήστης του Geospike μπορεί:

- Να χρησιμοποιεί την εφαρμογή είτε online, είτε offline.
- Να κοινοποιεί πληροφορίες από την θέση του στο χρονολόγιο του facebook, χωρίς να περιμένει να το κάνει ο χρήστης, αλλά το χειρίζεται η εφαρμογή στο παρασκήνιο.

Στην παρακάτω εικόνα 6 φαίνονται κάποια στιγμιότυπα της εφαρμογής:



Εικόνα 6: Στιγμιότυπα της εφαρμογής [9]

Τέλος, υπάρχουν πολλά θετικά για τις παραπάνω εφαρμογές, αλλά και πολλά πράγματα που θα μπορούσαν να διορθωθούν. Π.χ. για την εφαρμογή Travelog [30], η οποία δίνει πληροφορίες και χάρτες για πόλεις της Ασίας, θα μπορούσε να επεκταθεί και σε περισσότερες πόλεις και αγορές. Σημαντικό θα ήταν όλες οι εφαρμογές, να μπορούσαν να υποστηρίζουν χάρτες όχι μόνο σε online λειτουργία, αλλά και σε offline, καθώς είναι γνωστό ότι η 3G περιαγωγή δεδομένων κοστίζει ακριβά στο εξωτερικό. Επίσης, ένα άλλο σημαντικό στοιχείο θα ήταν η δημιουργία και κοινοποίηση περιεχομένου βίντεο. Μόνο η εφαρμογή TravelPod [31] το επιτρέπει αυτό. Πλέον, οι αποθηκευτικοί χώροι είναι μεγάλοι και στα κινητά, οπότε η παραγωγή βίντεο θα ήταν εφικτή και ο χρήστης θα μπορούσε να έχει καλύτερες αναμνήσεις από το ταξίδι του.

## 2.2 Σχετικές Εργασίες

Στη συνέχεια θα παρουσιαστούν κάποιες σχετικές εργασίες, σύμφωνα με τις εξής θεματικές ενότητες:

- Πληθοπορισμός (crowdsourcing).
- Χωρικός Πληθοπορισμός (Spatial Crowdsourcing).
- Συλλογή περιεχομένου από χρήστες (User Generated Content).

### 2.2.1 Πληθοπορισμός

Από την στιγμή που τα κινητά τηλέφωνα εξελίχθηκαν σε έξυπνα κινητά, το φαινόμενο του πληθοπορισμού αποκτά όλο και περισσότερο ενδιαφέρον. Οι δυνατότητες πλέον των έξυπνων κινητών είναι μεγάλες. Με τη χρήση δυνατοτήτων όπως η κάμερα και το GPS, ο καθένας μπορεί να παράγει και να μοιράζεται περιεχόμενο, καθώς επίσης να βοηθά άλλους χρηστές κινητών τηλεφώνων να λύσουν καθημερινά προβλήματα. Ένα πρόβλημα μπορεί εύκολα μέσω κάποιου blog η κάποιου web site η κάποιας έξυπνης εφαρμογής στα κινητά να κοινοποιηθεί άμεσα ή έμμεσα στο πλήθος. Έτσι, εκατομμύρια χρήστες κινητών μπορούν εύκολα εθελοντικά να αναλάβουν να λύσουν το πρόβλημα. Υπάρχουν πολλές γνωστές εφαρμογές [4] που χρησιμοποιούν τον πληθοπορισμό για να λύνουν προβλήματα μέσω κάποιας ιστοσελίδας. Στη συνέχεια επεκταθήκαν και σε εφαρμογή στα κινητά τηλέφωνα και πολλές περισσότερες που δημιουργήθηκαν από τότε που εξελιχθήκαν σε έξυπνα κινητά. Εκτός από τα εμφανή χαρακτηριστικά και τις δυνατότητες που έχουν τα κινητά [4], υπάρχουν και άλλα εγγενή χαρακτηριστικά:

- Κάθε εφαρμογή σε κινητό μπορεί να είναι επέκταση μιας web εφαρμογής η μπορεί να είναι μια καινούρια εφαρμογή.

- Η συνεισφορά του πλήθους στο παραγόμενο περιεχόμενο είναι συμμετοχική ή ευκαιριακή.
- Η συνεισφορά στο παραγόμενο περιεχόμενο μπορεί να είναι αποτέλεσμα ατομικής ή συλλογικής ενέργειας.
- Η ποιότητα της συνεισφοράς μπορεί να είναι ομοιογενής ή ετερογενής.
- Τα κίνητρα που δίνονται στο πλήθος για να συμμετέχουν στη συλλογή περιεχομένου, μπορεί να είναι: οικονομικά, ηθικά, ψυχαγωγικά, αλλά και η ανταλλαγή υπηρεσιών.
- Η ανθρώπινη δεξιότητα.

Συνοπτικά στον παρακάτω Πίνακα 2 φαίνεται μία ταξινόμηση γνωστών εφαρμογών σύμφωνα με τα παραπάνω χαρακτηριστικά:

Applications	Επέκταση web εφαρμογής	Συνεισφορά πλήθους	Συνεισφορά στα δεδομένα είναι αποτέλεσμα	Ποιότητα συνεισφοράς	Κίνητρα	Ανθρώπινη δεξιότητα	Αισθητήρες	Γεωγραφική Θέση
Gigwalk.com	✓	συμμετοχική	ατομική	ετερογενής	οικονομικά	Εργατικός	Κάμερα	✓
Jana.com	✓	συμμετοχική	ατομική	ετερογενής	οικονομικά	Οπτικός	*	✓
Crowd Translator	✓	συμμετοχική	συλλογική	ομοιογενής	ανταλλαγή υπηρεσιών	Οπτικός	Κάμερα	*
Waze.com	*	συμμετοχική και ευκαιριακή	συλλογική	ομοιογενής	Ηθικά/ ανταλλαγή υπηρεσιών	Οπτικός	*	✓
CityExplorer	*	ευκαιριακή	συλλογική	ομοιογενής	ψυχαγωγικά	Οπτικός	Κάμερα	✓
VTrack	*	ευκαιριακή	συλλογική	ομοιογενής	ηθικά/ανταλλαγή υπηρεσιών	*	*	✓
SignalGuru	*	ευκαιριακή	συλλογική	ομοιογενής	Ηθικά/ ανταλλαγή υπηρεσιών	*	Κάμερα	✓
Ear-Phone	*	ευκαιριακή	συλλογική	ομοιογενής	ηθικά	*	Ήχος	✓
NoiseTube	*	ευκαιριακή	συλλογική	ομοιογενής	ηθικά	*	Ήχος	✓
PotHole	*	ευκαιριακή	συλλογική	ομοιογενής	ηθικά	*	δόνηση	✓
AirPlace	*	ευκαιριακή	συλλογική	ομοιογενής	ανταλλαγή υπηρεσιών	*	*	✓
SmartTrace	*	ευκαιριακή	συλλογική	ομοιογενής	ανταλλαγή υπηρεσιών	*	*	✓
Crowdcast	*	ευκαιριακή	συλλογική	ομοιογενής	ανταλλαγή υπηρεσιών	*	*	✓
SmartP2P	*	ευκαιριακή	συλλογική	ομοιογενής	ανταλλαγή υπηρεσιών	*	*	✓

**Πίνακας 2: Εγγενή χαρακτηριστικά εφαρμογών που σχετίζονται με τη συλλογή περιεχομένου [4]**

### 2.2.2 Χωρικός Πληθοπορισμός

Τα τελευταία χρόνια, η μεγάλη ανάπτυξη που υπάρχει στα κινητά τείνει να κάνει τον χωρικό πληθοπορισμό εξαιρετικά δημοφιλή. Υπάρχει τεράστιο ενδιαφέρον για χρήση των έξυπνων κινητών αλλά και του Web ώστε να δημιουργηθούν, να συγκεντρωθούν και να διαδοθούν γεωγραφικές πληροφορίες που προσφέρονται εθελοντικά από απλούς χρήστες και όχι εξειδικευμένο προσωπικό. Αυτό που κάνει τον χωρικό πληθοπορισμό τόσο δημοφιλή στη σημερινή εποχή είναι η δημοτικότητα των κινητών τηλεφώνων και των αισθητήρων που περιέχουν, όπως το GPS, καθώς και η εξέλιξη της ευρυζωνικότητας στα κινητά δίκτυα. Πλέον υπάρχουν αρκετές εφαρμογές που ακολουθούν το μοντέλο του χωρικού πληθοπορισμού. Για να λειτουργήσει αυτό το μοντέλο πρέπει να επιτραπεί στους χρήστες να λύνουν προβλήματα ενώ ταξιδεύουν. Έτσι, οι δυναμικοί χρήστες αυτών των εφαρμογών είναι κυρίως όσοι ταξιδεύουν πολύ. Ένα τέτοιο παράδειγμα είναι το waze.com [34], μία εφαρμογή με την οποία ο χρήστης μέσω κινητού τηλεφώνου μπορεί να αναφέρει προβλήματα στο κυκλοφορία μιας περιοχής και να τα κοινοποιεί στα υπόλοιπα μέλη της εφαρμογής, ώστε να αποφύγουν τέτοια σημεία.

Υπάρχουν και άλλες ιστοσελίδες και εφαρμογές που ακολουθούν το μοντέλο του χωρικού πληθοπορισμού [12], τέτοιες είναι το Wikimapia [35] και το OpenStreetMap [19]. Αυτές οι πρωτοβουλίες επιτρέπουν στο πλήθος να συλλέξουν γεωγραφικές πληροφορίες και να ενισχύσουν την βάση δεδομένων που διατηρούν από γεωγραφικά δεδομένα σε παγκόσμιο επίπεδο. Επίσης, η Google μέσω των τεχνολογιών που έχει αναπτύξει, παροτρύνει προγραμματιστές να υλοποιήσουν εφαρμογές χρησιμοποιώντας τα δικά τους γεωγραφικά δεδομένα. Στη WikiMapia καθένας με μια σύνδεση στο Internet μπορεί να επιλέξει μια περιοχή στην επιφάνεια της Γης και να της ορίσει μια περιγραφή, καθώς επίσης να συμπεριλάβει συνδέσμους σε άλλες πηγές. Στη συνέχεια οποιοσδήποτε μπορεί να επεξεργαστεί αυτές τις εγγραφές. Εθελοντές αξιολογητές μπορούν να οπτικοποιήσουν τα αποτελέσματα και να κάνουν έλεγχο για την ακρίβεια και τη σπουδαιότητα των αποτελεσμάτων. Όπως περιγράφεται και στην εργασία [12], τα γεωγραφικά δεδομένα σε αυτές τις ιστοσελίδες / εφαρμογές δημιουργούνται βάσει εθελοντικής προσπάθειας των χρηστών. Μια άλλη παρόμοια ιστοσελίδα είναι το Flickr [8], το οποίο επιτρέπει στους χρήστες να ανεβάζουν και να εντοπίζουν φωτογραφίες στην επιφάνεια της Γης βάσει του γεωγραφικού πλάτους και μήκους. Επίσης το MissPronouncer [16], μια ιστοσελίδα που δημιουργήθηκε για να βοηθήσει τους ανθρώπους να μάθουν να προφέρουν μερικά από τα πιο χαρακτηριστικά τοπωνύμια.

Χαρακτηριστικές τεχνολογίες που χρησιμοποιούνται στην συλλογή γεωγραφικής πληροφορίας από εθελοντές είναι οι παρακάτω:

- Έξυπνα κινητά.
- Web 2.0 (Ιστοσελίδες, μπλογκς, wikis).
- Γεωναφορά (Georeferencing). Το περιεχόμενο που συλλέγεται από τους χρήστες πρέπει να συνοδεύεται από συντεταγμένες.
- Geotags. Ένα geotag είναι ένας τυποποιημένος κωδικός, ο οποίος μπορεί να εισαχθεί έτσι ώστε να γίνει γνωστή η θέση από την οποία προέρχεται η πληροφορία.
- GPS.
- Γραφικά για την οπτικοποίηση των γεωγραφικών πληροφοριών.
- Τηλεπικοινωνίες.

Το μοντέλο όμως αυτό του χωρικού πληθοπορισμού δημιουργεί [12] κάποια εύλογα ερωτήματα όπως:

- Τι οδηγεί τους ανθρώπους στο να το κάνουν αυτό;
- Πόσο ακριβή είναι τα αποτελέσματα;
- Απειλείται η ιδιωτικότητα;

Η αυτοπροβολή είναι ένα μεγάλο κίνητρο για να ασχοληθεί κάποιος με το web και κατ' επέκταση με την εθελοντική γεωγραφική πληροφορία. Η δημοτικότητα των έξυπνων κινητών δίνει από μόνη της κίνητρο στους χρήστες να συλλέξουν γεωγραφικές πληροφορίες. Επίσης πολλοί άνθρωποι βλέπουν το web σαν ένα εύκολο μέσο για να διαθέσουν (γεωγραφικές και μη) πληροφορίες σε φίλους τους, άσχετα αν αυτές γίνονται τελικά προσβάσιμες σε όλους. Το γεγονός αυτό αποτελεί τη βάση για τη δημοτικότητα ορισμένων ιστοσελίδων όπως το Picasa [20], όπου μπορεί κάποιος να επισημάνει φίλους του σε φωτογραφίες, αλλά δεν μπορεί να εξηγήσει τη δημοτικότητα του Flickr και του WikiMapia, όπου το περιεχόμενο είναι εντελώς ανώνυμο. Οι χρήστες του OpenStreetMap μπορούν π.χ. να ικανοποιηθούν βλέποντας αυτή τη τεράστια χωρική βάση να μεγαλώνει από δεδομένα που προσέφεραν οι ίδιοι, αλλά δεν μπορούν να νιώσουν κάποια ικανοποίηση αυτοπροβολής μιας και το έργο του OpenStreetMap βασίζεται στην ανωνυμία. Η ακρίβεια στα αποτελέσματα εξαρτάται αρκετά από την χρήση των διάφορων αισθητήρων, όπως τους δέκτες GPS και των σφαλμάτων που περιέχουν. Εξαρτάται επίσης από την αξιοπιστία που υπάρχει προς τον εθελοντή παραγωγό γεωγραφικού περιεχομένου. Οι δέκτες GPS, όσο εξελίσσεται η τεχνολογία, εξελίσσονται και αυτοί και βελτιώνονται συνεχώς. Επίσης μεγάλες ιστοσελίδες δημιουργούν διάφορους μηχανισμούς ώστε να ταυτοποιείται η θέση του χρήστη που παράγει το γεωγραφικό περιεχόμενο, έτσι ώστε να εξασφαλίζεται μεγάλο επίπεδο αξιοπιστίας ανάμεσα στον παραγωγό και στον καταναλωτή περιεχομένου. Τέλος, από τη στιγμή που τα γεωγραφικά δεδομένα κοινοποιούνται στο Διαδίκτυο, είναι λίγο δύσκολο να μην απειλείται η



ιδιωτικότητα. Πολλές από αυτές τις ιστοσελίδες που προαναφέρθηκαν στηρίζονται στην ανωνυμία αυτών που παράγουν περιεχόμενο. Άλλες πάλι επιτρέπουν στους χρήστες να παράγουν περιεχόμενο, εφόσον είναι μέλη μιας κοινότητας. Έτσι, είναι δύσκολο να μην παραβιαστεί η ιδιωτικότητα.

### **2.2.3 Ποιότητα δεδομένων από χωρικό πληθοπορισμό**

Η ποιότητα των δεδομένων έχει μεγάλη σημασία για τις εταιρείες που προσφέρουν ανοιχτά γεωγραφικά δεδομένα. Έτσι πολλές από αυτές αναπτύσσουν μηχανισμούς ώστε τα δεδομένα που παράγει το πλήθος να είναι αξιόπιστα και μάλιστα [15] υπάρχει τρόπος για ασφαλή εντοπισμό. Επίσης μπορούν να δημιουργηθούν υπηρεσίες που να επιτρέπουν στους χρήστες που παράγουν περιεχόμενο να επισημάνουν το περιεχόμενό τους χωροχρονικά, έτσι ώστε να δείχνει τη θέση τους ανά πάσα στιγμή, διατηρώντας την ιδιωτικότητα των χρηστών που παράγουν το περιεχόμενο. Για να χτισθεί ένα επίπεδο εμπιστοσύνης μεταξύ αυτών που χρησιμοποιούν το περιεχόμενο και αυτών που παράγουν το περιεχόμενο, βασικό θα ήταν να επαληθευτεί ως προς την εγκυρότητα του το περιεχόμενο. Μία χαρακτηριστική προσέγγιση είναι η χωροχρονική σήμανση του περιεχομένου, επαληθεύοντας τον χρόνο και τη θέση. Μια τέτοια προσέγγιση είναι το geotagging, που χρησιμοποιεί π.χ. το Flickr για να επιτρέπει στους χρήστες του να επισημαίνουν χωροχρονικά φωτογραφίες που δημιουργούν οι ίδιοι. Μια αξιόπιστη υπηρεσία geotagging, δεν θα επέτρεπε εύκολα σε ένα χρήστη να επισημάνει χωροχρονικά περιεχόμενο σε μέρος που δεν έχει πάει ποτέ. Έτσι, ο καταναλωτής μπορεί να εμπιστευτεί περισσότερο το περιεχόμενο που παράγουν οι χρήστες χωρίς να ξέρει από ποιον χρήστη προέρχεται. Από τη στιγμή που ο χρήστης πρέπει να διανύσει μια απόσταση για να επισημάνει περιεχόμενο σε μια επιθυμητή θέση, είναι δύσκολο και να δημιουργήσει περιεχόμενο που δεν είναι αληθινό. Επίσης, ένα σύστημα επαλήθευσης περιεχομένου που παράγεται από κινητές συσκευές θα μπορούσε να έχει τρεις οντότητες [15]:

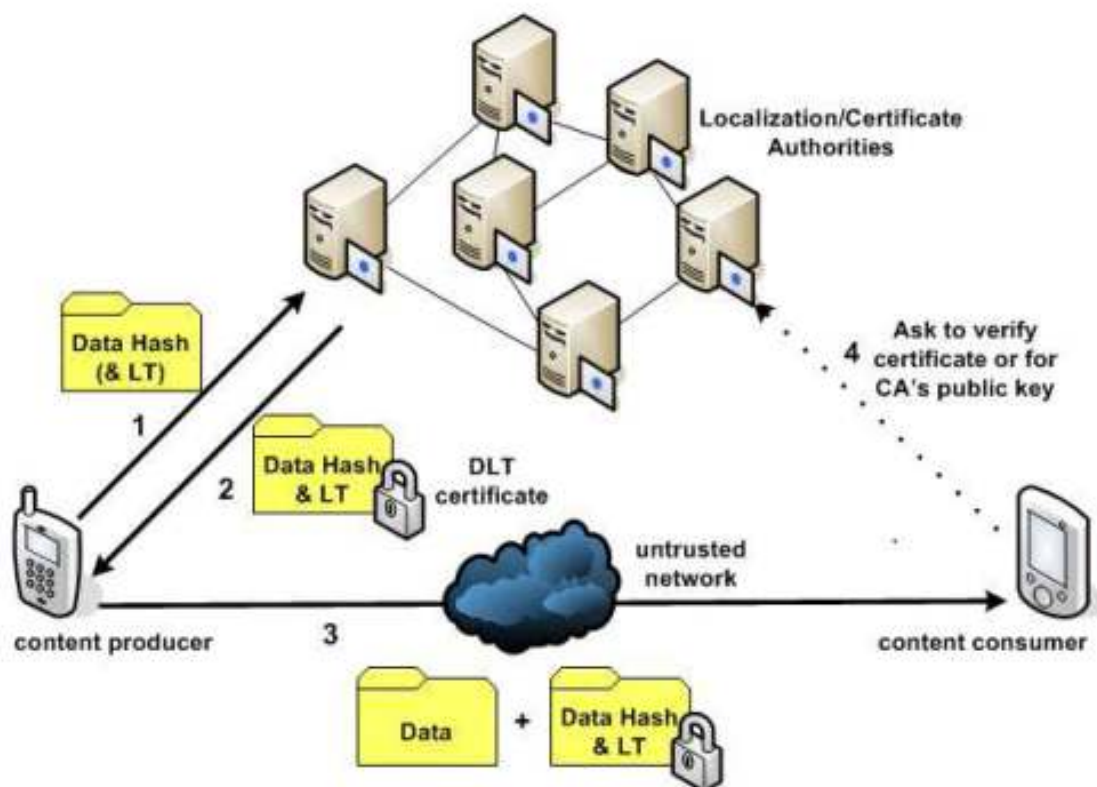
- Παραγωγούς περιεχομένου,
- Καταναλωτές περιεχομένου,
- Υπηρεσία επαλήθευσης τοποθεσίας και ώρας.

Όταν ένας παραγωγός περιεχομένου έχει κάποιο περιεχόμενο που θέλει να πιστοποιήσει γεωγραφικά, κάνει ένα αίτημα προς την αρχή έκδοσης πιστοποιητικών θέσης. Το αίτημα αυτό συνοδεύεται και από ένα αναγνωριστικό το οποίο παράγεται από τον αλγόριθμο SHA (Secure Hash Algorithm). Τότε η αρχή έκδοσης, καθορίζει τη θέση του παραγωγού με ασφαλή τρόπο και απαντά στον παραγωγό περιεχομένου με ένα πιστοποιητικό θέσης και ώρας που συνδέεται με τη θέση του την τρέχουσα ώρα και για το συγκεκριμένο

αναγνωριστικό περιεχομένου. Έτσι όταν ο καταναλωτής περιεχομένου αποκτήσει κάποιο περιεχόμενο μπορεί πλέον να επαληθεύσει τον χρόνο και τον χώρο του περιεχομένου. Η υλοποίηση ενός τέτοιου συστήματος για την επαλήθευση της θέσης θα μπορούσε να γίνει μέσω GPS ή κεραιών κινητής τηλεφωνίας. Για την έκδοση πιστοποιητικών υπάρχουν δυο προσεγγίσεις:

- *Συγκεντρωτική.* Ένας κεντρικός Server εξετάζεται όταν υπάρχει κάποιο αίτημα για πιστοποίηση περιεχομένου και αυτός είναι υπεύθυνος για την επαλήθευση της θέσης του περιεχομένου και την αντίστοιχη έκδοση του πιστοποιητικού.
- *Αποκεντρωμένη.* Σε αυτή την περίπτωση εξετάζονται πολλά μηχανήματα που μπορούν και εκδίδουν πιστοποιητικά, έτσι ώστε να αποφεύγεται και ο φόρτος που μπορεί να έχει ένας κεντρικός Server, όταν υπάρχουν πολλά αιτήματα.

Ολόκληρο το σύστημα φαίνεται στην παρακάτω εικόνα 7:



**Εικόνα 7: Επισκόπηση του συστήματος επαλήθευσης [15]**

Τύποι εφαρμογών που θα μπορούσαν να επωφεληθούν από ένα τέτοιο αξιόπιστο σύστημα ή υπηρεσία (χωροχρονικής σήμανσης) είναι οι παρακάτω:

- *Ειδησεογραφικές εφαρμογές.* Η δημιουργία του περιεχομένου που παράγεται από τέτοιες εφαρμογές, π.χ. βίντεο από συμβάντα, θα μπορούσε να ελεγχθεί με το χώρο και τον χρόνο των συμβάντων.
- *Κοινοποίηση Φωτογραφιών.* Με την πιστοποίηση της θέσης κατά τη στιγμή της δημιουργίας της φωτογραφίας, τέτοιες εφαρμογές μπορούν να φιλτράρουν τις αυθεντικές φωτογραφίες.
- *Φιλτράρισμα ενοχλητικών e-mail.*
- *Χαρτογράφηση δεδομένων.* Π.χ. η Google χρησιμοποιεί φωτογράφους εφοδιασμένους με GPS για να αποκτήσει την χαρτογράφηση των δεδομένων που χρειάζεται. Η εξακρίβωση της θέσης θα μπορούσε να αποδειχθεί πολύ χρήσιμη για την εμπιστοσύνη του περιεχομένου που οι χρήστες δημιουργούν.
- *Έλεγχος του κυκλοφοριακού.* Είναι κρίσιμο να καθοριστεί αν πράγματι ένα μήνυμα προέρχεται από μια συγκεκριμένη θέση, αφού έτσι αποτρέπει έναν χρήστη να ενημερώσει για κίνηση σε μέρος όπου δεν έχει φυσική παρουσία.

Στην εφαρμογή που αναπτύχθηκε στα πλαίσια αυτής της εργασίας, ο χρήστης μπορεί και δημιουργεί σημεία ενδιαφέροντος μέσω της κινητής συσκευής, τα οποία έχουν χωροχρονική σήμανση. Ο χρήστης δημιουργεί το σημείο το οποίο συνοδεύεται από συντεταγμένες και καταγράφεται επίσης στη βάση η ώρα δημιουργίας του σημείου. Στη συνέχεια τα κοινοποιεί στον Server όπου μπορούν άλλοι χρήστες (καταναλωτές) να δουν αυτά τα σημεία και εάν επιθυμούν να τα επισκεφτούν.

## 2.2.4 Εφαρμογές πληθοπορισμού και χωρικού πληθοπορισμού.

Στην συνέχεια θα παρουσιαστούν αναλυτικότερα κάποιες εφαρμογές και πλατφόρμες που χρησιμοποιούν το μοντέλο του πληθοπορισμού.

### 2.2.4.1 SmartTrace

Η SmartTrace [4], επιτρέπει την αναζήτηση της ομοιότητας των ίχνων μεταξύ των χρηστών των έξυπνων κινητών. Απαντά σε ερωτήματα της μορφής «Ανάφερε χρήστες με όμοια ίχνη». Βελτιστοποιεί τα ερωτήματα σε σχέση με το χρόνο απόκρισης και την κατανάλωση ενέργειας στα κινητά, χωρίς να κοινοποιούνται οι τροχιές των χρηστών. Στην εφαρμογή έχει δημιουργηθεί μια διεπαφή χρήστη, μέσω της οποίας καταγράφονται τα ίχνη των χρηστών σε μια τοπική βάση και στη συνέχεια κοινοποιούνται στο Server και εμφανίζονται στην οθόνη. Μέσω της διεπαφής ο χρήστης μπορεί να συνδεθεί στον Server της εφαρμογής και να κάνει ερωτήματα πάνω στις τροχιές που είναι αποθηκευμένες από άλλους χρήστες. Στην Εικόνα 8 φαίνεται η λειτουργία και οι διεπαφές αυτής της εφαρμογής:



Εικόνα 8: (α) Μοντέλο συστήματος και (β) Στιγμιότυπα της εφαρμογής SmartTrace από το κινητό

### 2.2.4.2 Crowdcast

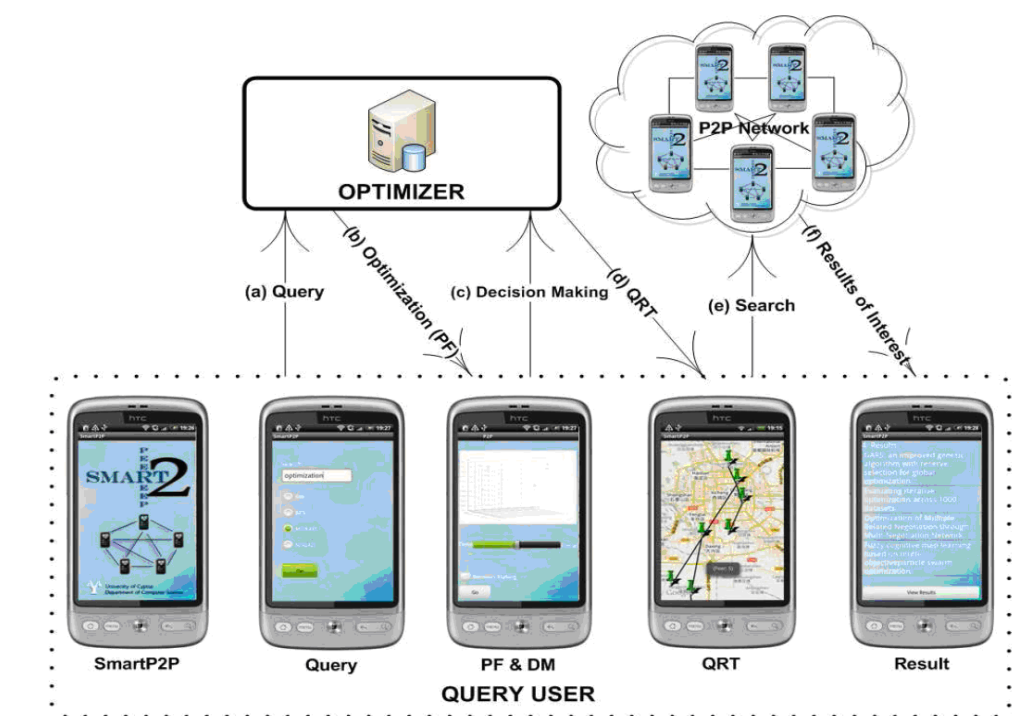
Η Crowdcast [4] ενεργοποιεί την αλληλεπίδραση μεταξύ των χρηστών η οποία είναι βασισμένη στη θέση, υπολογίζοντας αποτελεσματικά τους  $k$  εγγύτερους γείτονες κάθε χρήστη ανά πάσα στιγμή. Ο Server έχει την εικόνα με τα ίχνη των χρηστών και μπορεί να υπολογίσει τους  $K$  εγγύτερους γείτονες για κάθε χρήστη όπως φαίνεται στην Εικόνα 9 (β), ανά πάσα στιγμή. Χρησιμοποιεί ως χώρο αναζήτησης έναν κύκλο μιας συγκεκριμένης ακτίνας, έτσι κάθε χρήστης μέσα σε αυτόν τον κύκλο είναι υποψήφιος γείτονες για κάθε χρήστη μέσα στον κύκλο, όπως φαίνεται και στην παρακάτω Εικόνα 9 (γ). Τέλος στην Εικόνα 9 (α) φαίνονται κάποια στιγμιότυπα της εφαρμογής από το κινητό:



Εικόνα 9: Εφαρμογή Crowdcast [4]

### 2.2.4.3 SmartP2P

Η SmartP2P [4] χρησιμοποιεί δεδομένα θέσης που διατίθενται από το πλήθος για να βελτιστοποιήσει τη διαδικασία της αναζήτησης. Όταν ο χρήστης υποβάλει ένα ερώτημα του τύπου «δείξε μου φωτογραφίες από τους δρόμους του Χολαργού», τότε το σύστημα θα αναζητήσει φωτογραφίες από κάποιον που ζει κοντά στην περιοχή του Χολαργού. Αν είναι περισσότεροι από ένας, θα τις αναζητήσει πρώτα σε αυτόν που είναι πιο κοντά στον χρήστη που υποβάλει το ερώτημα. Με κάθε νέο ερώτημα, ο χρήστης αρχικά κατεβάζει ένα δένδρο δρομολόγησης του ερωτήματος από τον Server. Αυτό το δένδρο φτιάχνεται έτσι ώστε να βελτιστοποιεί πολλαπλούς σκοπούς ταυτόχρονα, κατά τη διάρκεια των αναζητήσεων σε ένα P2P δίκτυο για κινητά. Με το δένδρο αυτό ελαχιστοποιείται η κατανάλωση ενέργειας κατά τη διάρκεια της αναζήτησης, καθώς και ο χρόνος απόκρισης. Για να επιτευχθούν όλα αυτά, δεν υπάρχει μόνο ένα δένδρο που να βελτιστοποιεί όλους αυτούς τους σκοπούς της αναζήτησης. Έτσι, χρησιμοποιείται ένα σύστημα λήψης αποφάσεων, το οποίο επιλέγει το δένδρο, που απαντά καλύτερα στο ερώτημα του χρήστη. Τέλος το επιλεγμένο δένδρο χρησιμοποιείται στην αναζήτηση στο P2P δίκτυο των χρηστών των κινητών για να ανακτήσει ο χρήστης τις κατάλληλες απαντήσεις, όπως φαίνεται στην Εικόνα 10:



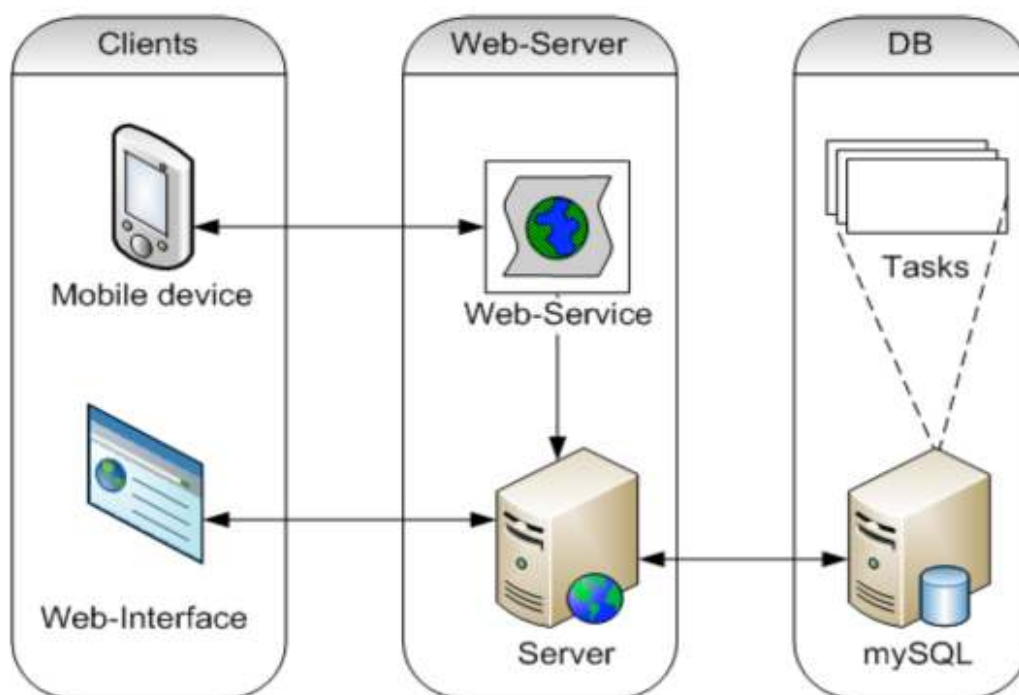
Εικόνα 10: Το διάγραμμα ροής εργασίας της εφαρμογής SmartP2P και οι διεπαφές χρήστη από συσκευή android [4]

#### 2.2.4.4 Πλατφόρμα πληθοπορισμού βασισμένη στην γεωγραφική θέση

Στην εργασία [2] δημιουργήθηκε μια πλατφόρμα που ενσωματώνει την τοποθεσία ως παράμετρο για τη διανομή εργασιών στο πλήθος. Η πλατφόρμα αυτή αποτελείται από τρία στοιχεία:

- Μια web διεπαφή, όπου αυτοί που ψάχνουν άτομα για εργασία, να δημιουργήσουν εργασίες που έχουν να κάνουν με την γεωγραφική πληροφορία.
- Έναν Server, συμπεριλαμβανομένης και της βάσης δεδομένων, ο οποίος θα είναι υπεύθυνος για τη διανομή των εργασιών στο πλήθος.
- Μια εφαρμογή στα κινητά, στην οποία θα μπορεί να βρει ο χρήστης διαθέσιμες εργασίες προς ανάληψη οι οποίες είναι κοντά στο μέρος που βρίσκεται.

Η αρχιτεκτονική του συστήματος φαίνεται στην παρακάτω Εικόνα 11:



Εικόνα 11: Η Αρχιτεκτονική του συστήματος [2]

Τα αποτελέσματα της συγκεκριμένης εργασίας έδειξαν ότι η ενσωμάτωση των εργασιών από τον ψηφιακό κόσμο στον πραγματικό είναι εφικτή και χρήσιμη. Παρατηρήθηκε ότι οι περισσότεροι χρήστες των κινητών προτιμούσαν να αναλαμβάνουν κάποια εργασία, παρά να αναθέτουν σε άλλους. Η συλλογή εικόνων ήταν η πιο ευχάριστη εργασία, ενώ οι χρήστες

προτιμούσαν να λύνουν προβλήματα και εργασίες που ήταν κοντά στο μέρος όπου βρίσκονταν και αυτοί. Μπορεί εύκολα κάποιος να παρατηρήσει ότι το αρχιτεκτονικό διάγραμμα μοιάζει πολύ με το διάγραμμα του συστήματος που δημιουργήθηκε στο πλαίσιο αυτής της εργασίας, άσχετα αν το περιεχόμενο μπορεί περιορίζεται σε μια συγκεκριμένη διεργασία, δηλαδή την συλλογή φωτογραφιών και σχολίων από επισκέψεις χρηστών σε διάφορες περιοχές.

#### **2.2.4.5 gMission: Μια πλατφόρμα χωρικού πληθοπορισμού**

Η πλατφόρμα gMission [3] διαθέτει μια συλλογή από νέες τεχνικές, όπως η γεωγραφική ανίχνευση, η ανίχνευση ανθρώπων που είναι υποψήφιοι να αναλάβουν μια εργασία και οι συστάσεις εργασιών προς του χρήστες. Το σύστημα μπορεί να:

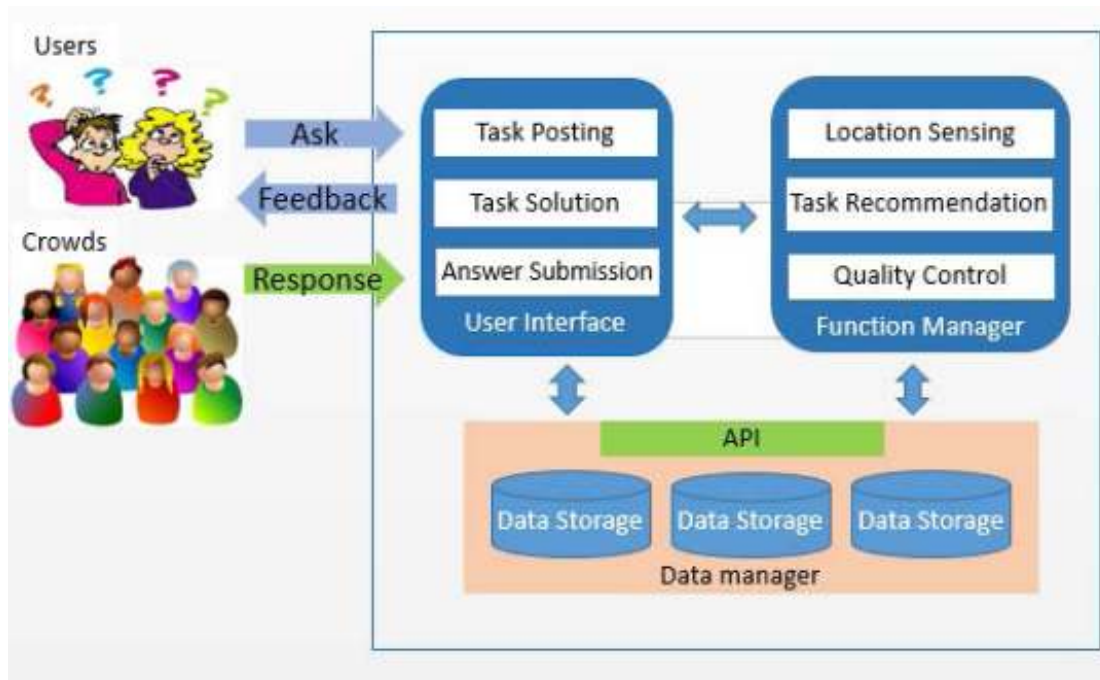
- Δίνει στο πλήθος μια πλατφόρμα μέσω της οποίας μπορεί ο χρήστης να ζητήσει αλλά και να συλλέξει ο ίδιος γεωγραφικές πληροφορίες.
- Είναι εξοπλισμένο με πολλαπλές λειτουργίες ανίχνευσης θέσης, έτσι ώστε να πραγματοποιείται η οπτικοποίηση του πλήθους που έχει αναλάβει κάποια εργασία αλλά και να γίνονται εισηγήσεις εργασιών από το πλήθος που να είναι βασισμένες στην θέση του χρήστη.

Η αρχιτεκτονική του συστήματος αποτελείται από τις παρακάτω ενότητες.

- *Την διεπαφή* μέσω της οποίας μπορεί ο χρήστης να βάλει μια εργασία προς ανάθεση, να λύσει μια διεργασία και να καταθέσει μια απάντηση σε κάποιο από τα ερωτήματα που έχουν ζητηθεί απαντήσεις.
- *Την ανίχνευση θέσης*. Η πλατφόρμα μπορεί να κρατάει τα ίχνη των χρηστών σε αυτήν την ενότητα.
- *Την σύσταση εργασιών* σύμφωνα με τη θέση των χρηστών.
- *Ο έλεγχος ποιότητας*, ο οποίος αποτελείται από τρεις ενότητες, αυτές του ελέγχου του κύκλου ζωής του συστήματος, της επαλήθευσης βάσει θέσης και τον wise – market ποιοτικό έλεγχο. Το wise – market είναι ένα μοντέλο πιθανοτήτων που επιτυγχάνει ισορροπία μεταξύ του κόστους και της πυκνότητας των απαντήσεων που έχουν υποβληθεί.

Το αρχιτεκτονικό σχέδιο φαίνεται παρακάτω στην Εικόνα 12:





Εικόνα 12: Η αρχιτεκτονική του συστήματος της πλατφόρμας gMission [3]

#### 2.2.4.6 Αξιοποίηση του Flickr API στη διερεύνηση τοπωνυμίων

Στην εργασία [13] διερευνάται η γλώσσα που χρησιμοποιείται για να περιγράψει και να ανακαλύψει κάποιος σημαντικές περιοχές μιας πόλης όπως το κέντρο. Αυτό βασίζεται σε γεωαναφερόμενα μεταδεδομένα και μεταδεδομένα με επισήμανση ετικέτας που σχετίζονται με εικόνες του Flickr και έτσι φαίνεται πώς ένας μεγάλος αριθμός ανθρώπων αποκαλούν κύριες περιοχές μιας πόλης. Το μεγαλύτερο μέρος των γεωαναφερόμενων εικόνων (περίπου το 70%) που υπάρχουν στο Flickr έχουν επισημανθεί με ετικέτες που περιγράφουν πολύ συγκεκριμένα το μέρος, χρησιμοποιώντας ειδικά τοπωνύμια. Έτσι, χρησιμοποιώντας τα μεταδεδομένα του Flickr, όχι μόνο μπορεί κάποιος να καταλάβει κάποιο μέρος από τον όρο που χρησιμοποιείται γι' αυτό, αλλά και να εξερευνήσει τα οριά του, σε επίπεδο πόλης. Ο κύριος στόχος είναι να διερευνηθεί πότε και πως μπορούν να χρησιμοποιηθούν οι επισημάνσεις με ετικέτα στις εικόνες του Flickr, έτσι ώστε να γίνει αντιληπτό πώς αποκαλούν διάφορες περιοχές οι άνθρωποι και να βοηθήσει στις προσπάθειες που γίνονται ώστε να μιμηθούν αυτή τη συμπεριφορά οι υπολογιστές.

Πιο συγκεκριμένα, χρησιμοποιήθηκε το API του Flickr για να συλλεγούν μεταδεδομένα για πάνω από 8 εκατομμύρια εικόνες. Επιλέχτηκαν έξι πόλεις με ποικίλα πολιτισμικά και γλωσσικά χαρακτηριστικά, ως παράδειγμα για την ποιοτική και ποσοτική αξιολόγηση της χρήσης των ετικετών στα διάφορα μέρη: η Ζυρίχη, το Λονδίνο, το Σέφιλντ,

το Σικάγο, το Σιάτλ και το Σύδνεϋ. Τα γεωαναφερμένα μεταδεδομένα συλλέχτηκαν από συγκεκριμένες περιοχές κάθε πόλης. Επίσης δημιουργήθηκε ένα δεύτερο σύνολο από δεδομένα που έχουν επισημανθεί με γεωγραφική ετικέτα, από οπουδήποτε στον πλανήτη και οι ετικέτες τους ταιριάζουν με τη λέξη «downtown», «central», «cbd», «innercity», «citycentre» για το κέντρο της πόλης. Τέλος, δημιουργήθηκε ένα τρίτο σύνολο δεδομένων με εικόνες που επιλέχτηκαν δειγματοληπτικά από τις περιοχές της μεγάλης Βρετανίας, της Αμερικής και της Αυστραλίας, κι είχαν σαν ετικέτα μια από τις παραπάνω πέντε λέξεις. Για να βγει ένα συμπέρασμα για την ποιότητα των δεδομένων που συλλέχτηκαν στο πρώτο σύνολο, εξετάστηκαν ως προς την ακρίβεια η θέση των δεδομένων με γεωγραφική ετικέτα. Η πλειονότητα των εικόνων φαίνεται να είχαν υψηλή ακρίβεια ως προς τη θέση. Δεδομένου ότι αυτή η προσέγγιση εστιάζει στην επιλογή σημείων για ανάλυση με βάση τις ετικέτες, παρουσιάζεται μια μέθοδος που διερευνά την επίδραση των παραγωγικών χρηστών στη δημιουργία του πρότυπου μεταδεδομένων των ετικετών. Η μέθοδος ελέγχει για στρεβλώσεις που δημιουργούνται από τη συλλογή δεδομένων τόσο από τους παραγωγικούς χρήστες, όσο και από τους μη παραγωγικούς.

# 3

## *Αρχιτεκτονική*

### *συστήματος*

Ο σκοπός της εργασίας αυτής ήταν να βρεθεί ένας τρόπος, ώστε να μπορεί κάποιος να τηρήσει ηλεκτρονικά ένα ημερολόγιο από τα ταξίδια του. Από τη στιγμή που η τεχνολογία εξελίσσεται ραγδαία, δεν θα ήταν ελκυστικό αυτό να γίνεται με τον παραδοσιακό τρόπο, δηλαδή την καταγραφή των αναμνήσεων σε χαρτί. Πλέον υπάρχουν πολλά ηλεκτρονικά ημερολόγια και πολλά από αυτά βρίσκονται online, έτσι ώστε να μπορεί κάποιος να καταγράψει οτιδήποτε είναι σημαντικό για τον ίδιο. Όμως απευθύνονται σε χρήστες που θέλουν να καταγράψουν γενικά εργασίες που θεωρούν σημαντικές στη ζωή τους και όχι αποκλειστικά τα ταξίδια τους. Γι' αυτό δημιουργήθηκαν πολλά ταξιδιωτικά ιστολόγια στο Διαδίκτυο, π.χ. όπως το Trip Advisor [32], στο οποίο μπορεί ο καθένας να κάνει εγγραφή, να καταγράψει τις αναμνήσεις από τα ταξίδια του και να τις μοιραστεί με φίλους του ή ακόμα και με ταξιδιώτες, οι οποίοι έχουν κοινά ενδιαφέροντα. Ακόμα μια εξέλιξη σε αυτόν τον τομέα ήρθε με την ραγδαία ανάπτυξη των κινητών. Η ιδέα ήταν κάποιος άνθρωπος να μπορούσε να διατηρεί τις αναμνήσεις του στο κινητό τηλέφωνο, έτσι ώστε να τις κουβαλά πάντα μαζί του. Πριν μερικά χρόνια αυτό ήταν δύσκολο να γίνει, επειδή η τεχνολογία που υπήρχε στα κινητά τηλέφωνα δεν ήταν τόσο προηγμένη. Από την στιγμή που τα κινητά εξελίχθηκαν σε έξυπνα πλέον αυτό είναι εφικτό. Μεγάλες εταιρείες που δημιουργούν λειτουργικά συστήματα για κινητά, επιτρέπουν σε προγραμματιστές να δημιουργούν εθελοντικά εφαρμογές με σχετική ευκολία, ώστε να κάνουν τη ζωή του καταναλωτή εύκολη.

Όπως αναφέρθηκε και στο κεφάλαιο 2, υπάρχουν αρκετές εφαρμογές για κινητά μέσω των οποίων μπορεί ο χρήστης να καταγράψει τις αναμνήσεις από τα ταξίδια του. Αυτό το σκοπό έχει και η εφαρμογή που υλοποιήθηκε στο πλαίσιο αυτής της εργασίας. Μέσω της εφαρμογής ο χρήστης μπορεί να καταγράψει τις αναμνήσεις του από τα ταξίδια του και να τις

έχει πάντα μαζί του, στο κινητό του. Είναι σημαντικό να συγκεντρωθεί όλη αυτή η πληροφορία από τα ταξίδια σε μια εφαρμογή. Ο χρήστης θα μπορεί να αποθανατίζει με φωτογραφίες τις στιγμές από σημαντικές επισκέψεις που έκανε σε χαρακτηριστικά σημεία ενδιαφέροντος. Θα μπορεί επίσης να κρατεί σημειώσεις από τις επισκέψεις του και να κάνει κριτικές, που αργότερα σε μια επιστροφή του στην ίδια περιοχή να του φανούν χρήσιμες ή που θα μπορούσαν να φανούν χρήσιμες σε κάποιον φίλο του που πρόκειται να επισκεφτεί το ίδιο μέρος. Σημαντικό θα ήταν επίσης ο χρήστης μια τέτοιας εφαρμογής να σημειώνει ακριβώς πάνω σε ένα χάρτη τις επισκέψεις του και να μπορεί να τις βλέπει όποτε θελήσει πάνω στο χάρτη, ώστε να μπορεί να έχει μια εικόνα από το ταξίδι του, για το πώς ακριβώς κινήθηκε. Ιδανικά, θα μπορούσαν να προταθούν σημεία ενδιαφέροντος από τα μέρη που επισκέπτεται, τα οποία να μην είχε υπόψη του. Ένα σημαντικό στοιχείο ακόμα για τον χρήστη θα ήταν να μπορούσε να δημιουργεί όσα ταξίδια θέλει και να μην υπάρχει περιορισμός στα ταξίδια που μπορεί να κρατήσει στη συσκευή του. Βέβαια αυτό έχει να κάνει και με τον αποθηκευτικό χώρο κάθε συσκευής, αλλά με τη σημερινή εξέλιξη στα κινητά, οι αποθηκευτικοί χώροι έχουν μεγαλώσει αρκετά, ώστε να μπορεί ο χρήστης να κρατεί στη συσκευή όσα περισσότερα δεδομένα μπορεί. Επιπλέον καλό θα ήταν ο χρήστης να μπορεί να δημιουργεί όσες διαδρομές θέλει ανά ταξίδι, λ.χ. να μπορεί να χωρίσει το ταξίδι του σε μέρες καταγράφοντας τις αναμνήσεις του και τις επισκέψεις του ανά ημέρα και όχι ως ενιαίο ταξίδι και έτσι θα είναι πιο οργανωμένη η πληροφορία του.

Επίσης, σημαντικό θα ήταν ο χρήστης να μπορεί να αναζητά με διάφορους τρόπους αυτά τα ταξίδια που έκανε στο παρελθόν. Εφόσον, ο χρήστης έχει κάνει ένα μεγάλο αριθμό ταξιδιών, θα πρέπει να μπορεί και να τα αναζητά με κάποια κριτήρια, έτσι ώστε να μην χάνεται μέσα στον κατάλογο. Τα κριτήρια καλό θα ήταν να ποικίλουν, ώστε να διευκολύνουν στην αναζήτηση. Θα μπορούσαν να είναι είτε *χρονικά*, είτε *χωρικά* μέσω χαρτών, είτε μέσω *λέξεων κλειδιών*, είτε τέλος συνδυασμός όλων αυτών, ώστε να περιορίζονται τα αποτελέσματα σε αυτό που ψάχνει ο χρήστης.

Τέλος, σημαντικό θα ήταν να μπορεί ο χρήστης να ανεβάζει αυτά τα ταξίδια του σε κάποιο κεντρικό Server, είτε γιατί μπορεί να έχει μικρό αποθηκευτικό χώρο στο κινητό του, είτε γιατί θέλει να τα μοιραστεί με άλλους ταξιδιώτες της κοινότητας. Έτσι θα μπορούσε να βοηθήσει στα ταξίδια τους άλλους χρήστες, αλλά και να βοηθηθεί από άλλους, βλέποντας σχόλια και φωτογραφίες από μέρη που πιθανόν να επισκεφτεί.

### 3.1 Αρχιτεκτονικό διάγραμμα

Για να μπορεί ο χρήστης να συγκεντρώσει όλες τις πληροφορίες από τα ταξίδια του στην συσκευή, θα πρέπει στην εφαρμογή να χρησιμοποιηθεί μια *τοπική βάση δεδομένων* στη συσκευή, η οποία θα μπορεί να διαχειρίζεται *χωρικές* (γεωμετρικά σημεία, τα οποία θα είναι η γεωγραφική θέση από τις επισκέψεις του χρήστη αλλά και γεωμετρικές γραμμές οι οποίες θα συνθέτονται από τα γεωγραφικά σημεία των επισκέψεων του χρήστη), αλλά και *περιγραφικές* πληροφορίες. Μέσω της χωρικής βάσης δεδομένων θα γίνονται όλες οι λειτουργίες ανάκτησης και αποθήκευσης των πληροφοριών ενός ταξιδιού.

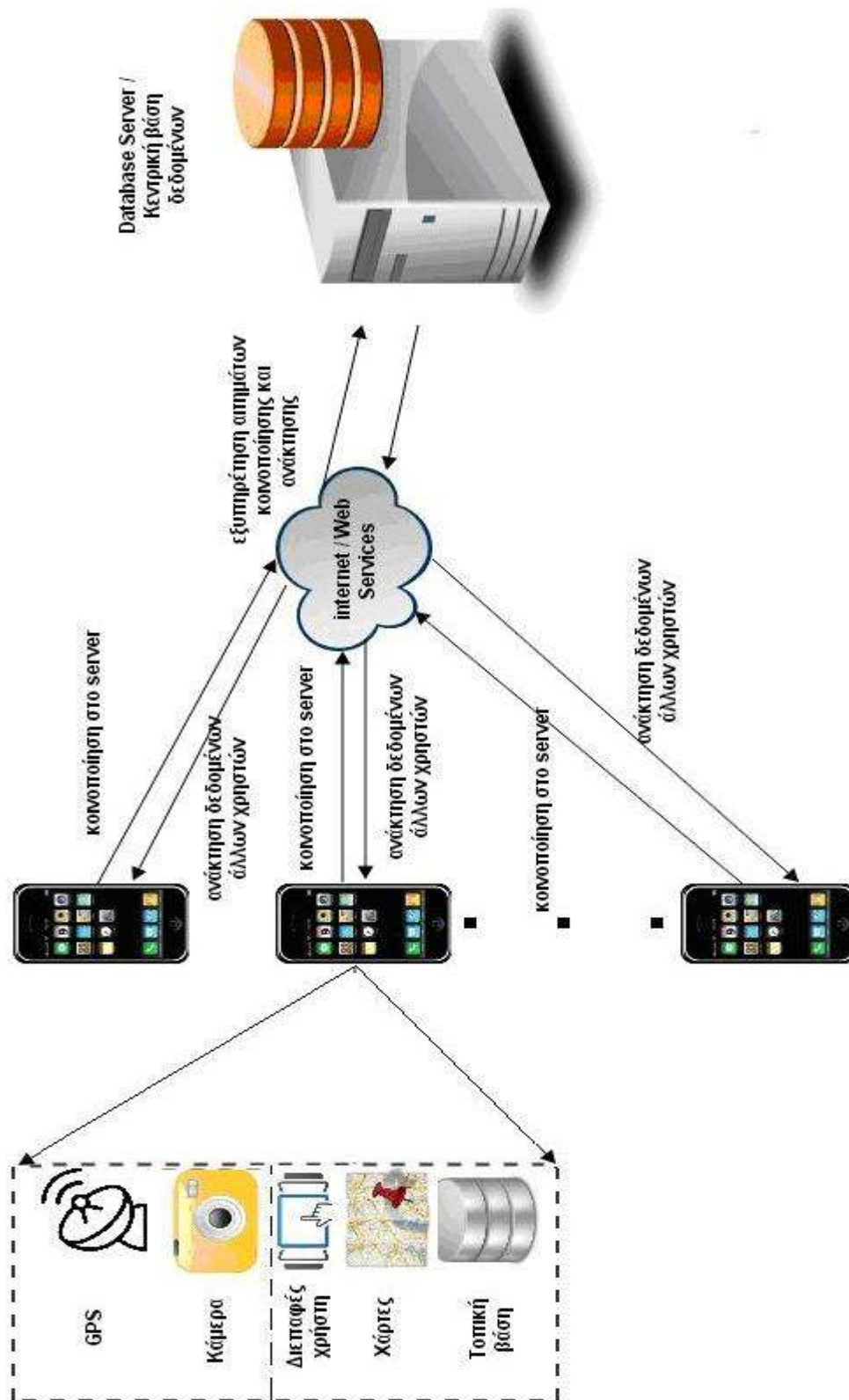
Οι *φωτογραφίες* θα αποθηκεύονται και θα ανακτώνται από τη μνήμη της συσκευής για να μην φορτώνεται η τοπική βάση δεδομένων. Έτσι το κινητό θα πρέπει να είναι εφοδιασμένο με μια κάμερα, για να μπορεί να συλλέγει φωτογραφίες από τις επισκέψεις του ο χρήστης και να τις αποθηκεύει στην συσκευή.

Για να μπορεί ο χρήστης να αποθηκεύει με ακρίβεια τη θέση του στη τοπική χωρική βάση δεδομένων θα πρέπει το κινητό να υποστηρίζει κάποιον παγκόσμιο *χάρτη*, πάνω στον οποίο μέσω του αισθητήρα *GPS* που είναι πλέον εφοδιασμένα τα περισσότερα έξυπνα κινητά θα εντοπίζεται η θέση του χρήστη, αλλά και θα μπορεί ο χρήστης να σημειώσει κάποια επίσκεψη του σε ένα μέρος που τον ενδιαφέρει.

Ένα σημαντικό ακόμα συστατικό αυτού του συστήματος είναι να χρησιμοποιηθεί ένας κεντρικός *Server* με μια χωρική βάση δεδομένων, στην οποία θα αποθηκεύονται οι πληροφορίες από τους χρήστες όλων των συσκευών που θέλουν να κοινοποιήσουν το ταξίδι τους. Η βάση αυτή θα πρέπει να έχει τα ίδια χαρακτηριστικά με την τοπική βάση, δηλαδή να δίνει τη δυνατότητα στο χρήστη να αποθηκεύει και να ανακτά τόσο περιγραφικές, όσο και χωρικές πληροφορίες. Για να κοινοποιεί ο χρήστης τα δεδομένα του ταξιδιού του στον κεντρικό *Server*, θα πρέπει να χρησιμοποιηθεί ένας *web Server*, στον οποίο θα πρέπει να υλοποιηθούν κάποια *web services* για να εξυπηρετούν τα αιτήματα κοινοποίησης και ανάκτησης δεδομένων από / προς τον *Database Server*. Για να γίνει η εξυπηρέτηση αυτών των αιτημάτων, θα πρέπει η συσκευή να μπορεί να χρησιμοποιεί κάποιο ευρυζωνικό δίκτυο, κάτι που πλέον είναι εφικτή δυνατότητα.

Για να είναι δυνατή όλη αυτή η διαχείριση πληροφοριών στη συσκευή και στη κεντρική βάση δεδομένων θα πρέπει να δημιουργηθούν στο κινητό οι κατάλληλες *διεπαφές* χρήστη. Οι *διεπαφές* χρήστη πρέπει να είναι φιλικές, ώστε να μπορεί ο χρήστης εύκολα να κάνει αυτό που θέλει στο ταξίδι του ανά πάσα στιγμή και να μην χάνεται. Μέσα από τις *διεπαφές* αυτές θα μπορεί να προσθέτει, να διαγράφει, να επεξεργάζεται και να ανακτά πληροφορίες ταξιδιών τόσο από την τοπική βάση δεδομένων, όσο και από την βάση

δεδομένων του Server. Στην παρακάτω εικόνα φαίνεται ένα γενικό αρχιτεκτονικό σχέδιο του συστήματος που υλοποιήθηκε σε αυτήν εδώ την εργασία.



Εικόνα 13: Η αρχιτεκτονική του συστήματος

### 3.2 Τοπική βάση δεδομένων

Η τοπική βάση δεδομένων διαχειρίζεται όλες τις χωρικές και περιγραφικές πληροφορίες που απαιτούνται από την εφαρμογή. Θα πρέπει λοιπόν να αποθηκεύονται στη βάση δεδομένων και πληροφορίες που περιέχουν γεωμετρία. Για τη δημιουργία ενός ταξιδιού, ο χρήστης θα έχει τη δυνατότητα, ενώ ταξιδεύει σε μία περιοχή, να δημιουργεί όσες διαδρομές θέλει και στην κάθε διαδρομή ο χρήστης να δημιουργεί και όσες επισκέψεις θέλει. Αυτή είναι και η έννοια του ταξιδιού για τον χρήστη της εφαρμογής. Κάθε διαδρομή αποτελείται από έναν αριθμό επισκέψεων σε σημεία πάνω στο χάρτη και η κάθε επίσκεψη είναι ένα σημείο πάνω στο χάρτη οπουδήποτε στην περιοχή που έχει επισκεφτεί. Έτσι, η τοπική βάση δεδομένων θα πρέπει να έχει τη δυνατότητα να αποθηκεύει τόσο περιγραφικές όσο και χωρικές πληροφορίες για τις επισκέψεις, τις διαδρομές και κατ' επέκταση τα ταξίδια του χρήστη.

Πιο συγκεκριμένα, για το ταξίδι θα είναι δυνατόν να αποθηκεύονται περιγραφικές πληροφορίες όπως ο τίτλος του, μια σύντομη περιγραφή, η ημερομηνία δημιουργίας, αλλά και η ημερομηνία που τελειώνει το ταξίδι για τον χρήστη.

Εφόσον ο χρήστης θέλει να χωρίσει το ταξίδι του σε ημέρες ή να κρατεί ξεχωριστές διαδρομές σε ένα ταξίδι και όχι να φαίνεται όλο του το ταξίδι σαν μια ενιαία διαδρομή, θα πρέπει να αποθηκεύονται τόσο περιγραφικές όσο και χωρικές πληροφορίες για τις διαδρομές. Τέτοιες πληροφορίες μπορεί να είναι ο τίτλος της διαδρομής, μια σύντομη περιγραφή, η ημερομηνία δημιουργίας, αλλά και η τροχιά από τις επισκέψεις που θα κάνει ο χρήστης σε κάθε διαδρομή, η οποία θα πρέπει να έχει γεωγραφική υπόσταση.

Επίσης, σε κάθε διαδρομή ο χρήστης θα πρέπει να έχει τη δυνατότητα να καταγράφει όσες επισκέψεις θέλει σε σημεία που τον ενδιαφέρουν. Στην τοπική βάση δεδομένων θα πρέπει να αποθηκεύονται χωρικές και περιγραφικές πληροφορίες της επίσκεψης όπως ο τίτλος της επίσκεψης, η κατηγορία της (πχ αν επισπεύτηκε κάποιο μουσείο, αξιοθέατο, εστιατόριο, ξενοδοχείο κ.τ.λ.), η ημερομηνία δημιουργίας της και τέλος η γεωγραφική θέση της επίσκεψης που πραγματοποιήσε.

Σε κάθε επίσκεψη του ο χρήστης καλό θα ήταν να αποθηκεύει στη βάση δεδομένων όσα σχόλια και να κρατεί όσες σημειώσεις θέλει για την επίσκεψή του. Μια τέτοια πληροφορία θα βοηθούσε το χρήστη αν επισκεπτόταν ξανά το ίδιο μέρος, να αποφύγει επισκέψεις που δεν του άρεσαν ή να δει τι έχει αλλάξει από το προηγούμενο ταξίδι του. Επίσης, εφόσον ο χρήστης κοινοποιήσει αυτές τις πληροφορίες, αυτές θα μπορούσαν να φανούν χρήσιμες σε άλλους χρήστες της εφαρμογής που έχουν σκοπό να επισκεφτούν το ίδιο μέρος.

Όπως και με τις σημειώσεις, σημαντικό θα ήταν ο χρήστης να μπορεί να βγάζει και να αποθηκεύει όσες περισσότερες φωτογραφίες θέλει από κάθε επίσκεψη του, έτσι ώστε να έχει

την καλύτερη δυνατή ανάμνηση από το σημείο που επισκέφτηκε. Για να μην επιβαρύνεται η βάση δεδομένων με εικόνες, ο χρήστης θα μπορεί να αποθηκεύει τις εικόνες στη μνήμη της συσκευής και στην βάση να κρατηθεί μια αναφορά στο σημείο της μνήμης όπου βρίσκεται η φωτογραφία.

Για να μπορεί ο χρήστης να βλέπει προτάσεις για επισκέψεις με *γνωστά σημεία ενδιαφέροντος*, θα πρέπει η βάση δεδομένων να έχει φορτωμένες πληροφορίες, όπως το όνομα του σημείου, η κατηγορία του σημείου (π.χ. μουσείο, αρχαιολογικό χώρο, εστιατόριο, ξενοδοχείο κ.τ.λ.) αλλά και η γεωγραφική θέση του σημείου ενδιαφέροντος, τα οποία διατίθενται ελεύθερα στο Διαδίκτυο από το OpenStreetMap [19].

Τέλος, όπου υπάρχει γεωγραφική πληροφορία όπως στις διαδρομές και στις επισκέψεις, θα πρέπει να χρησιμοποιηθούν *χωρικά ευρετήρια*, έτσι ώστε η αναζήτηση που θα είναι βασισμένη σε χωρικά κριτήρια και χωρικούς τελεστές να είναι όσο το δυνατόν πιο γρήγορη.

### **3.3 Αισθητήρες συσκευής**

Κάθε συσκευή θα πρέπει να είναι εφοδιασμένη με μια *κάμερα*, έτσι ώστε ο χρήστης να βγάζει όσες φωτογραφίες θέλει από τις επισκέψεις του. Δεν τίθενται συγκεκριμένες προδιαγραφές, αλλά εξαρτάται καθαρά από την τεχνολογία της κάθε συσκευής, αφού όλα σχεδόν τα έξυπνα κινητά είναι εφοδιασμένα με μια κάμερα. Ανάλογα με την τεχνολογία του κινητού εξαρτάται και η ποιότητα της κάμερας και ανάλογα με την ποιότητα της κάμερας, είναι και το μέγεθος κάθε φωτογραφίας. Όσο καλύτερη είναι η κάμερα, τόσο μεγαλύτερο μέγεθος έχει και η φωτογραφία που βγάζει ο χρήστης. Ένας άλλος σημαντικός αισθητήρας που είναι απαραίτητος για το σύστημα που σχεδιάστηκε σε αυτήν την εργασία είναι ο *αισθητήρας εντοπισμού θέσης*. Τα περισσότερα έξυπνα κινητά πλέον είναι εφοδιασμένα με κάποιον αισθητήρα εντοπισμού θέσης, έστω και αν είναι υποβοηθούμενος από το ευρυζωνικό δίκτυο που χρησιμοποιεί η συσκευή. Μέσω αυτού του αισθητήρα, η εφαρμογή θα δίνει τη δυνατότητα στο χρήστη να εντοπίζει τη θέση του πάνω σε έναν παγκόσμιο χάρτη, ώστε να δει τι υπάρχει γύρω του και να δημιουργήσει τις επισκέψεις που επιθυμεί στο ταξίδι του. Σημαντικό είναι επίσης η συσκευή να έχει τη δυνατότητα να χρησιμοποιεί κάποιο ευρυζωνικό δίκτυο, ώστε να μπορεί ο χρήστης να κοινοποιεί πληροφορίες από το ταξίδι του στον κεντρικό Server, χρησιμοποιώντας τα web services που δημιουργήθηκαν γι' αυτό το σύστημα.

### **3.4 Χάρτες**

Μέσω του αισθητήρα εντοπισμού θέσης δίνεται η δυνατότητα στο χρήστη να εντοπίζει τη θέση του πάνω σε ένα χάρτη. Ο χάρτης θα πρέπει να είναι παγκόσμιος και σε διάφορες



κλίμακες, έτσι ώστε ο χρήστης να μπορεί να επισκεφτεί οποιοδήποτε μέρος πάνω στη γη. Επίσης θα πρέπει να υποστηρίζει κάποιο γνωστό σύστημα γεωγραφικών συντεταγμένων, ώστε να μπορεί ο χρήστης να προσδιορίσει τη θέση του ή την επίσκεψη του πάνω στο χάρτη. Ο χρήστης θα πρέπει να έχει τη δυνατότητα να μεγεθύνει όσο θέλει στο χάρτη, ώστε να προσδιορίσει την θέση του με όση μεγαλύτερη ακρίβεια γίνεται και να καθορίζει παράθυρα πάνω στο χάρτη για να κάνει αναζητήσεις πληροφοριών που θα είναι βασισμένες στο χάρτη. Σημαντικό, επίσης θα ήταν να υπάρχει η δυνατότητα ο χάρτης να διατίθεται και σε online και σε offline λειτουργία στο χρήστη. Σε περιοχές όπου δεν διατίθεται κάποιο ευρυζωνικό δίκτυο στη συσκευή ή σε περιοχές όπου η χρήση κάποιου ευζωνικού δικτύου κοστίζει ακριβά, η offline λειτουργία θα ήταν πολύ σημαντική. Με κάποιο τρόπο γεωαναφερμένες εικόνες με τμήματα χαρτών από περιοχές ενδιαφέροντος θα πρέπει να μπορούν να αποθηκεύονται τοπικά στη συσκευή και να προσφέρονται σε διάφορες κλίμακες στο χρήστη, ώστε να χρησιμοποιεί τους χάρτες μέσα στην εφαρμογή και σε offline λειτουργία.

### **3.5 Database Server**

Ο Server θα φιλοξενεί τη κεντρική βάση δεδομένων, στην οποία θα κοινοποιούνται τα ταξίδια όσων χρηστών το επιθυμούν. Επίσης θα απαντά μέσω των web services στα αιτήματα ανάκτησης και αποθήκευσης πληροφοριών από την συσκευή στον Server. Η σημασία της δημιουργίας αυτής της κεντρικής βάσης δεδομένων έγκειται στην δυνατότητα που δίνει στο χρήστη να μπορεί να εντοπίζει πληροφορίες επισκέψεων άλλων χρηστών εύκολα και γρήγορα, οι οποίες μπορεί να του φανούν χρήσιμες στο δικό του ταξίδι. Π.χ. ο χρήστης θα έχει τη δυνατότητα να υποβάλει ένα ερώτημα στην κεντρική βάση δεδομένων όπως «Δείξε μου επισκέψεις χρηστών που επισκέφτηκαν το Παρίσι και έφαγαν κρουασάν», τότε η βάση να του απαντά γρήγορα εξετάζοντας χιλιάδες κείμενα με επισκέψεις χρηστών που είχαν σαν λέξεις κλειδιά το «Παρίσι» και το «κρουασάν». Σημαντικό είναι επίσης ότι με τη δημιουργία αυτής της κεντρικής βάσης δεδομένων, ο χρήστης θα μπορεί να ανεβάσει όλα τα ταξίδια του στον Server σε περίπτωση που ο αποθηκευτικός χώρος της συσκευής είναι μικρός ή έχει γεμίσει ή ακόμα να τον χρησιμοποιήσει ως αντίγραφο ασφαλείας για τα ταξίδια του. Η βάση δεδομένων θα πρέπει να έχει παρόμοια χαρακτηριστικά με την τοπική βάση δεδομένων στις κινητές συσκευές, οπότε θα πρέπει να έχει τη δυνατότητα αποθήκευσης τόσο χωρικών όσο και περιγραφικών πληροφοριών. Σε αυτή τη βάση δεδομένων, αποθηκεύονται πληροφορίες από όλες τις συσκευές των χρηστών, οπότε θα πρέπει να υπάρχει ένας τρόπος να ξεχωρίζονται τα ταξίδια που κοινοποιούνται ανά χρήστη. Ένας τρόπος για να γίνει αυτό, είναι μαζί με τις πληροφορίες του ταξιδιού που αποθηκεύονται από την συσκευή στον Server να αποθηκεύεται και ένα αναγνωριστικό της συσκευής, έτσι ώστε να διαχωρίζονται ανά συσκευή οι πληροφορίες που κοινοποιούνται.

Θα πρέπει επίσης να υπάρχει η δυνατότητα να κρατεί τις σημειώσεις και τις εικόνες των επισκέψεων του κάθε χρήστη. Εδώ σε αντίθεση με τη συσκευή, οι φωτογραφίες θα μπορούν να αποθηκευτούν στη βάση δεδομένων μιας και ο Server είναι πολύ πιο δυνατό μηχανήμα και με μεγαλύτερους αποθηκευτικούς χώρους από τη συσκευή.

Για κάθε γεωγραφική πληροφορία θα πρέπει η βάση δεδομένων του Server να δίνει τη δυνατότητα να δημιουργηθούν χωρικά ευρετήρια, έτσι ώστε να είναι όσο το δυνατόν καλύτερη η αναζήτηση. Ένα άλλο σημαντικό στοιχείο είναι να υποστηρίζει κάποιο *έξυπνο ευρετήριο* για τις σημειώσεις των επισκέψεων, έτσι ώστε να γίνεται εύκολη και γρήγορη αναζήτηση με *λέξεις κλειδιά* σε μεγάλο αριθμό πληροφοριών.

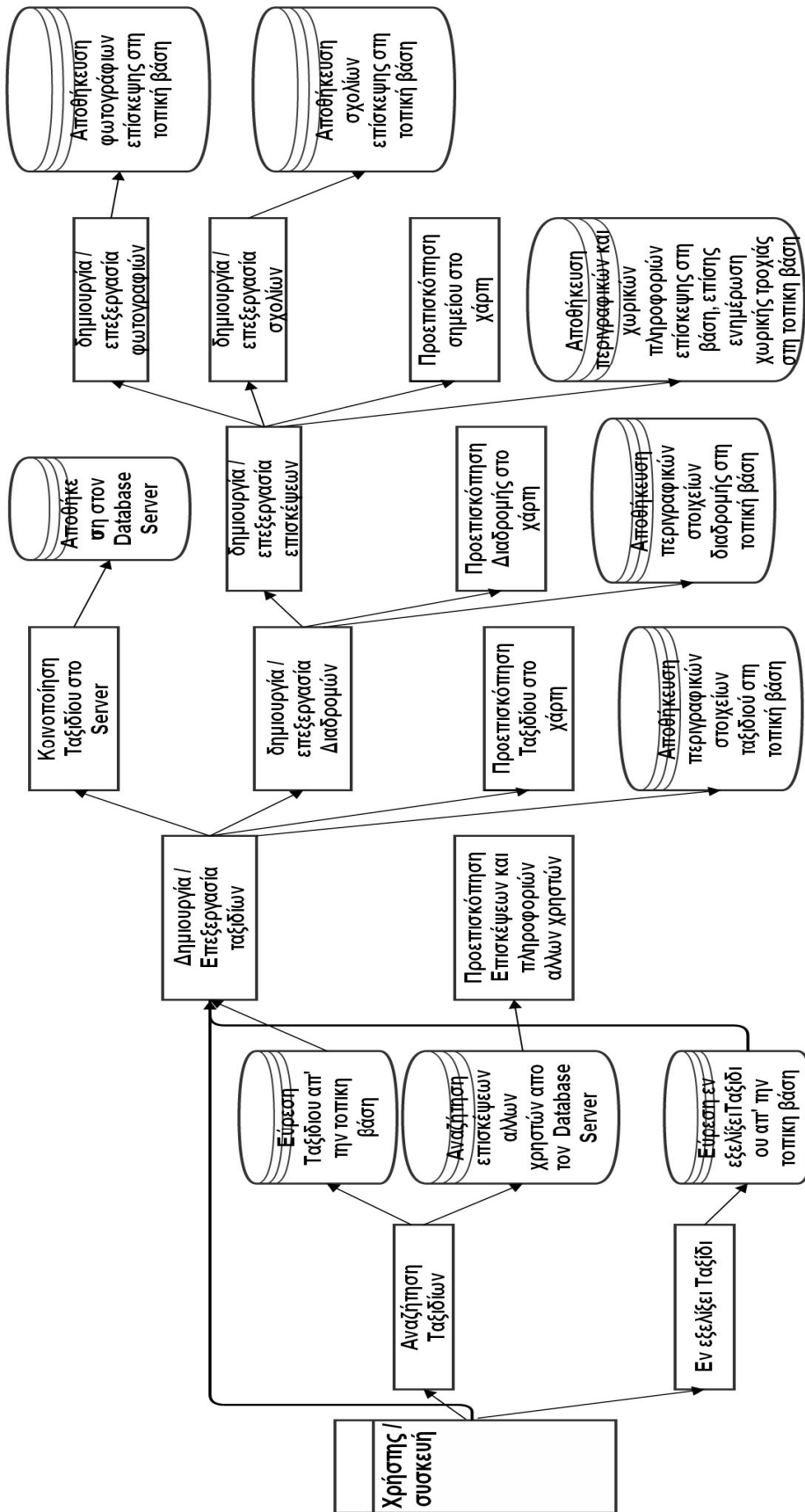
### **3.6 Web Services**

Τα web services, τα οποία θα φιλοξενοούνται σε κάποιον Server θα είναι υπεύθυνα για την εξυπηρέτηση των αιτημάτων ανάκτησης και αποθήκευσης δεδομένων από την συσκευή προς την βάση δεδομένων του Server και αντίστροφα. Για να γίνει αυτό θα πρέπει ο Server να είναι εφοδιασμένος με κάποιο web Server, έτσι ώστε να εξυπηρετεί αυτά τα αιτήματα μέσω του http πρωτόκολλου. Τα web services που θα δημιουργηθούν σε αυτό το σύστημα θα πρέπει να έχουν κάποιες ενσωματωμένες συναρτήσεις οι οποίες θα αποθηκεύουν, θα διαγράφουν και θα ανακτούν πληροφορίες από / προς την βάση δεδομένων του Server. Επίσης, τα web services θα πρέπει να επιστρέφουν τα αποτελέσματα του αιτήματος σε μορφή που να μπορούν να διαβαστούν από την εφαρμογή της συσκευής, όπως π.χ. σε xml ή json.

### **3.7 Διεπαφές χρήστη**

Για να είναι δυνατή όλη αυτή η διαχείριση των πληροφοριών του ταξιδιού προς την τοπική βάση δεδομένων, αλλά και προς την βάση δεδομένων του κεντρικού Server, θα πρέπει στο κινητό να δημιουργηθούν κάποιες διεπαφές χρήστη, που θα αναλάβουν αυτό το ρόλο. Για τους αντίστοιχους πίνακες της βάσης θα πρέπει να υπάρχουν και αντίστοιχες διεπαφές χρήστη. Π.χ. θα πρέπει να δημιουργηθεί μια διεπαφή στην οποία θα μπαίνει ο χρήστης και θα αποθηκεύει τις περιγραφικές πληροφορίες του ταξιδιού του στον πίνακα «ταξίδι», όπως ο τίτλος και η σύντομη περιγραφή. Αντίστοιχα θα πρέπει να δημιουργηθούν διεπαφές χρήστη για την διαχείριση των πληροφοριών «διαδρομές», «επισκέψεις», «σημειώσεις» και «φωτογραφίες». Στην διεπαφή για την διαχείριση των επισκέψεων, θα πρέπει να δίνεται η δυνατότητα στο χρήστη να χρησιμοποιεί κάποιον χάρτη για να μπορεί να βλέπει τη θέση του, να ελέγχει γνωστά σημεία ενδιαφέροντος και να σημειώνει την επίσκεψη που επιθυμεί απ' ευθείας πάνω στον χάρτη. Επίσης στην διεπαφή που θα διαχειρίζεται τις φωτογραφίες, θα πρέπει να δίνεται η δυνατότητα να χρησιμοποιεί την κάμερα του κινητού για να βγάζει

φωτογραφίες από τις επισκέψεις που τον ενδιαφέρουν. Μέσα από τις διεπαφές θα πρέπει να δίνεται η δυνατότητα για προσθήκη, διαγραφή, και επεξεργασία δεδομένων από την τοπική βάση δεδομένων, έτσι ώστε ο χρήστης να έχει τη μεγαλύτερη δυνατή διαχείριση της. Σημαντικό θα ήταν επίσης να δημιουργηθούν διεπαφές που να διευκολύνουν την αναζήτηση και ανάκτηση δεδομένων από την τοπική βάση δεδομένων αλλά και από την βάση δεδομένων του Server, καθώς επίσης να παρουσιάζονται τα αποτελέσματα αυτά σε λίστες που να διευκολύνουν στην εύρεση της κατάλληλης πληροφορίας που αιτήθηκε. Θα πρέπει επίσης μέσα από την διεπαφή που διαχειρίζεται τη συλλογή «ταξιδιών» να δίνεται η δυνατότητα στον χρήστη πατώντας κάποιο κουμπί να κοινοποιεί το ταξίδι του στη βάση δεδομένων του Server. Παρακάτω φαίνεται ένα διάγραμμα ροής δεδομένων μέσα από τις διεπαφές του χρήστη:



Εικόνα 14: Διάγραμμα ροής για την καταγραφή ταξιδιών

# 4

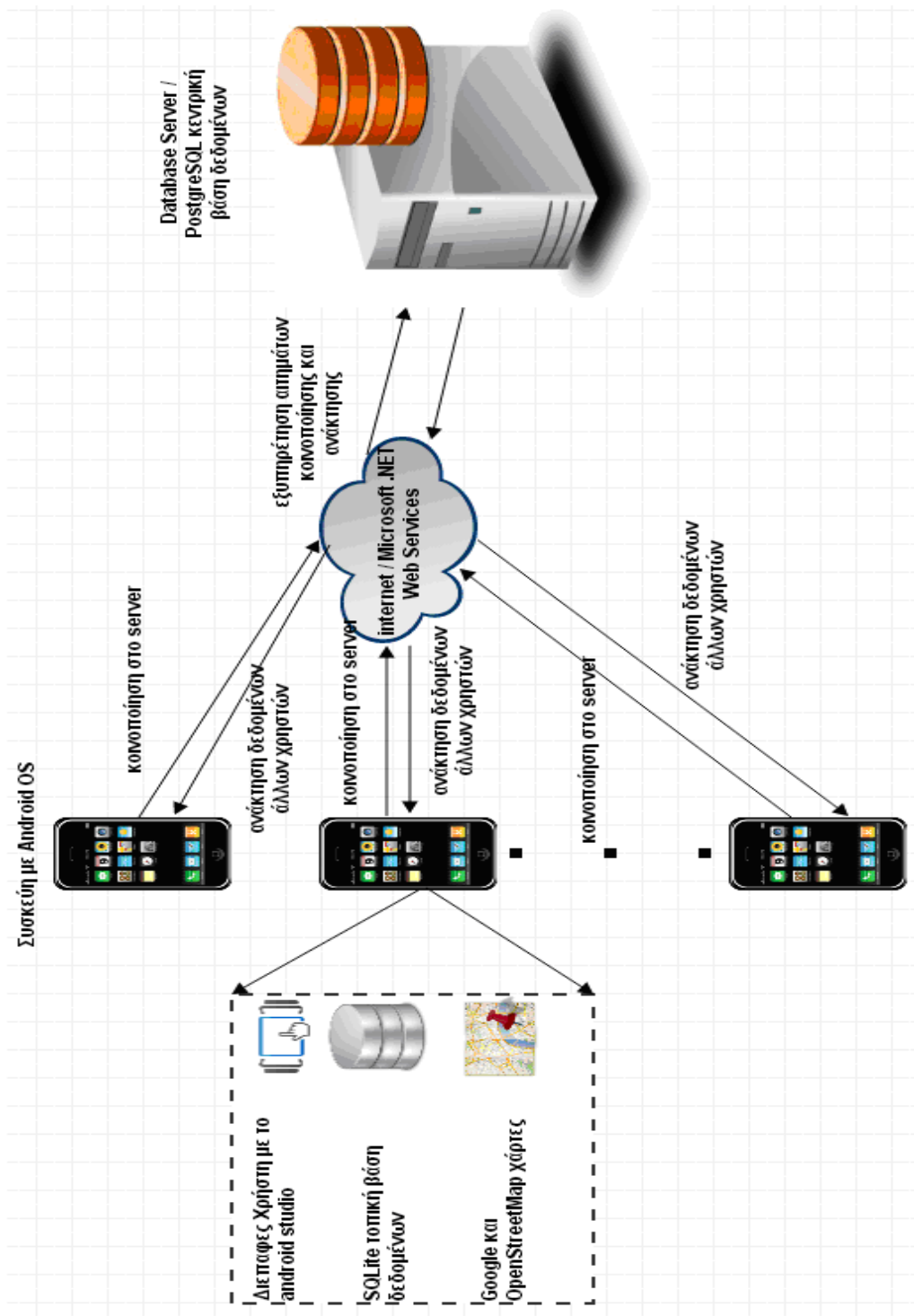
## Υλοποίηση συστήματος

Στο προηγούμενο κεφάλαιο αναπτύχθηκε η σχεδίαση του συστήματος. Παρουσιάστηκαν τα υποσυστήματα και τα διάφορα συστατικά του. Στη συνέχεια θα παρουσιαστούν τα συγκεκριμένα εργαλεία που χρησιμοποιήθηκαν ώστε να δημιουργηθεί το σύστημα:

- Η εφαρμογή του κινητού θα τρέχει σε συσκευές οι οποίες έχουν λειτουργικό σύστημα android 4.4 [1] ή και νεότερη έκδοση
- Για την υλοποίηση κλάσεων και διεπαφών έγινε χρήση του android studio 1.0.2 [5], το οποίο χρησιμοποιεί γλώσσα προγραμματισμού java.
- Για την τοπική βάση δεδομένων χρησιμοποιήθηκε η SQLite έκδοση 3.7.9 [27] με ενεργοποιημένη την επέκταση Spatialite έκδοση 3.0.1 [26] για την υποστήριξη των χωρικών οντοτήτων και λειτουργιών.
- Για κεντρική βάση δεδομένων χρησιμοποιήθηκε το σύστημα βάσεων δεδομένων PostgreSQL 9.3 [22] με ενεργοποιημένη την επέκταση PostGIS 2.1 [21] για την υποστήριξη των γεωμετριών στις χωρικές οντότητες. Επίσης, η PostgreSQL προσφέρει έναν μηχανισμό για πλήρη αναζήτηση κειμένων [28] στη βάση δεδομένων.
- Τα web services υλοποιήθηκαν με τεχνολογίες Microsoft .NET και γλώσσα προγραμματισμού C# [17]. Έτσι, ως web server χρησιμοποιήθηκε ο Internet Information Services της Microsoft, κάτω από τον οποίο φιλοξενούνται τα web services.
- Ως σύστημα γεω - εντοπισμού χρησιμοποιήθηκε το GPS, με το οποίο είναι εφοδιασμένες οι περισσότερες συσκευές android.

- Για την υποστήριξη των χαρτών χρησιμοποιήθηκαν για την online λειτουργία οι χάρτες της Google [11], ενώ για την offline λειτουργία χρησιμοποιήθηκαν τα tiles του OpenStreetMap [19], τα οποία διατίθενται δωρεάν.

Έτσι με την χρήση των συγκεκριμένων τεχνολογιών πλέον το αρχιτεκτονικό σχέδιο (εικόνα 13) του συστήματος παίρνει την μορφή που φαίνεται στην παρακάτω εικόνα:



Εικόνα 15: Η αρχιτεκτονική του συστήματος με τις τεχνολογίες που χρησιμοποιήθηκαν

## **4.1 Android λειτουργικό σύστημα συσκευής**

Η εφαρμογή του κινητού υλοποιήθηκε ώστε να τρέχει σε συσκευές που διαθέτουν λειτουργικό σύστημα android [1]. Το android είναι βασισμένο πάνω στο λειτουργικό σύστημα Linux και έχει αναπτυχτεί από την Google. Αρχικά το android σχεδιάστηκε για να τρέχει σε κινητές συσκευές με οθόνες αφής, όπως π.χ. τα έξυπνα κινητά και τα tablets, αλλά στη συνέχεια με ειδικές διεπαφές χρήστη χρησιμοποιήθηκε σε τηλεοράσεις, αυτοκίνητα, κονσόλες παιχνιδιών, ψηφιακές κάμερες ακόμα και ρολόγια. Το γεγονός ότι το android χρησιμοποιείται από τόσων ειδών συσκευές, αποτελεί ένα σημαντικό λόγο για τον οποίο επιλέχτηκε ώστε να αναπτυχτεί και η εφαρμογή του συστήματος συγκεκριμένης εργασίας.

Η Google, αποκάλυψε πως το 2014 είχε πάνω από 1 δισεκατομμύριο ενεργούς χρήστες το μηνά, ενώ τον προηγούμενο χρόνο είχε περίπου 538 εκατομμύρια. Αυτό δείχνει πως συνεχώς οι χρήστες που επιλέγουν κινητή συσκευή που διαθέτει android λειτουργικό σύστημα αυξάνονται, συνεπώς μεγαλώνει και ο αριθμός που χρησιμοποιούν εφαρμογές για android.

Το android είναι ένα ανοιχτού κώδικα λειτουργικό σύστημα, που μπορεί να επεκταθεί και να παραμετροποιηθεί ώστε να έρθει στα μέτρα του χρήστη. Συνήθως συνοδεύεται με αρκετές εφαρμογές από την Google που διευκολύνουν τον χρήστη στην καθημερινότητα και αυτό το κάνει ακόμα πιο δημοφιλές. Οι εταιρείες τεχνολογίας που απαιτούν ένα έτοιμο, χαμηλού κόστους και προσαρμόσιμο λειτουργικό σύστημα για συσκευές υψηλής τεχνολογίας γενικά επιλέγουν το android. Η ανοιχτή φύση του έχει ενθαρρύνει μια μεγάλη κοινότητα προγραμματιστών να χρησιμοποιήσουν τον open-source κώδικα ως θεμέλιο για την ανάπτυξη μεγάλων έργων - εφαρμογών, στα οποία προσθέτουν νέα χαρακτηριστικά για προχωρημένους χρήστες.

Τέλος, ένας ακόμα λόγος που συντέλεσε στην επιλογή του android είναι οι ευκολίες που δίνει στους προγραμματιστές. Το Standard Development Kit δίνει τη δυνατότητα σε κάθε προγραμματιστή να φτιάξει τις δίκες του εφαρμογές, να τις ανεβάσει στο market της android και να τις εκμεταλλευτεί οικονομικά. Το επίσημο περιβάλλον ανάπτυξης android studio, ή το eclipse της java δίνει πολλές ευκολίες στους προγραμματιστές, ώστε να αναπτύξουν τις δίκες τους εφαρμογές.

## **4.2 Περιβάλλον ανάπτυξης Android Studio**

Μέσω των καρτελών και των όψεων που προσφέρει το android studio [5], ο προγραμματιστής μπορεί να έχει κατάλληλα οργανωμένα τον σχεδιασμό και την υλοποίηση

της εφαρμογής. Κάθε ενότητα του έργου εμφανίζεται ως φάκελος στο ανώτερο επίπεδο της ιεραρχίας. Οι βασικές ενότητες του έργου είναι οι παρακάτω:

- java/ - για τα αρχεία του πηγαίου κώδικα.
- manifests/ - για τα αρχεία με δηλώσεις που αφορούν όλο το έργο.
- res/ - για τα αρχεία που έχουν να κάνουν με το μορφή των διεπαφών.
- Gradle Scripts/ - για τα αρχεία που έχουν να κάνουν με τη δημιουργία του έργου.

Επίσης, προσφέρει έναν εύκολο τρόπο δημιουργίας και τυποποίησης της εφαρμογής σε ένα αρχείο, το οποίο μπορεί μετά να εγκατασταθεί στην συσκευή. Ορίζοντας κάποιες παραμέτρους ο προγραμματιστής μπορεί με ένα κλικ να δημιουργήσει το πακέτο του έργου και στη συνέχεια να το εγκαταστήσει σε περιβάλλον android.

Ένα αλλά σημαντικό στοιχείο είναι ότι το android studio συνοδεύεται από μια μεγάλη γκάμα εξομοιωτών κινητών συσκευών. Έτσι, ο προγραμματιστής μπορεί να δοκιμάσει την εφαρμογή του σε όσους περισσότερους εξομοιωτές μπορεί και στη συνέχεια θα γνωρίζει ότι η εφαρμογή του μπορεί να τρέξει σε διάφορους τύπους συσκευών. Π.χ. θα μπορεί να δημιουργήσει τις διεπαφές με τέτοιο τρόπο, ώστε να παίζει σε όσο το δυνατόν περισσότερες συσκευές, από κινητά τηλεφώνια με διάφορες διαστάσεις, tablets, μέχρι και σε έξυπνες τηλεοράσεις που διαθέτουν android.

Το android studio διαθέτει επίσης ένα πολύ καλό σύστημα εκσφαλμάτωσης της εφαρμογής. Ένα τέτοιο σύστημα βοηθά τον προγραμματιστή να εντοπίζει εύκολα τα λάθη και να τα διορθώνει, ώστε να προχωρεί και πιο γρήγορα η ανάπτυξη.

Η εργαλειοθήκη διαθέτει πολλά εργαλεία, που βοηθούν τον προγραμματιστή να δημιουργήσει εύκολα τις δικές του διεπαφές, έτσι ώστε να είναι φιλικές προς τον τελικό χρήστη. Ο προγραμματιστής μπορεί πολύ εύκολα με ένα drag n drop των διαφόρων κουμπιών που διαθέτει η εργαλειοθήκη να δημιουργήσει εύχρηστες διεπαφές.

Τέλος η γλώσσα προγραμματισμού είναι η java, άρα υπάρχουν πλεονεκτήματα όπως κληρονομικότητα, ενθυλάκωση, πολυμορφισμός, εύκολη γραφή, έτοιμες βιβλιοθήκες που διευκολύνουν τον προγραμματιστή, το ότι μπορεί και τρέχει σε πολλά γνωστά λειτουργικά συστήματα και η αμέτρητη βοήθεια και υποστήριξη που υπάρχει στο Διαδίκτυο για τη γλώσσα java.



## 4.3 *SQLite με την επέκταση Spatialite*

### 4.3.1 *SQLite*

Για την διαχείριση των δεδομένων που θα δημιουργούνται από τις διεπαφές χρήστη στην κινητή συσκευή επιλέχθηκε η SQLite [27]. Η SQLite είναι ένα ολοκληρωμένο σύστημα διαχείρισης βάσεων δεδομένων. Είναι λογισμικό ανοιχτού κώδικα και έτσι οι προγραμματιστές μπορούν να το χρησιμοποιήσουν ώστε να φτιάξουν τις δικές τους επεκτάσεις. Η SQLite είναι δημοφιλής επιλογή για τη διαχείριση βάσεων δεδομένων σε κινητά τηλέφωνα, PDA, MP3 players, set-top boxes, και άλλες ηλεκτρονικές συσκευές, γιατί έχει ένα μικρό αποτύπωμα κώδικα και κάνει αποδοτική χρήση της μνήμης, του χώρου και του εύρους του δίσκου. Επίσης είναι πολύ αξιόπιστη, και δεν απαιτεί καμία συντήρηση από κάποιον διαχειριστή βάσης δεδομένων.

Ένας σημαντικός παράγοντας που έπαιξε ρόλο ώστε να προτιμηθεί η SQLite είναι ότι μπορεί να χρησιμοποιηθεί σε πολλά λειτουργικά συστήματα (Android, iOS, Linux, Mac, Solaris, VxWorks, και Windows). Συνεπώς, τα δεδομένα αν χρειαστεί μπορούν εύκολα να μεταφερθούν από το ένα περιβάλλον στο άλλο, χωρίς να είναι απαραίτητο να μεταφερθεί και η εφαρμογή.

Η SQLite δεν χρειάζεται εγκατάσταση και όλα τα δεδομένα μπορούν να μεταφερθούν μέσα σε ένα αρχείο. Η διαχείριση των βάσεων δεδομένων μπορεί να γίνει μέσα από τη γραμμή εντολών (command line). Αυτή η γραμμή εντολών μπορεί να χρησιμοποιηθεί και μέσα από το android studio.

Φυσικά διαθέτει επίσης πλήρη υποστήριξη της γλώσσας SQL, όποτε όλες οι ενημερώσεις των δεδομένων μπορούν να γίνουν μέσω δοσοληψιών (transaction), ενώ επίσης υποστηρίζει ευρετήρια (όπως B δένδρα).

Σημαντικό επίσης είναι ότι μπορεί να υποστηρίξει βάσεις δεδομένων μεγάλης χωρητικότητας (επιπέδου Terabytes). Μπορεί επίσης να υποστηρίξει Gigabytes από δυαδικά δεδομένα και συμβολοσειρές, καθώς επίσης χρονικά δεδομένα. Το android studio και η java διαθέτουν βιβλιοθήκες για την SQLite, οι οποίες βοηθούν τον προγραμματιστή να διαχειριστεί τα δεδομένα της εφαρμογής πολύ εύκολα μέσα από τις κλάσεις του.

Στην εφαρμογή του συστήματος της συγκεκριμένης εργασίας υπάρχουν και γεωμετρικά δεδομένα, όπως οι επισκέψεις του χρήστη σε κάποιο σημείο, αλλά και η καταγραφή της τροχιάς του χρήστη στο ταξίδι του. Η SQLite δεν διαθέτει την κατάλληλη υποδομή για να υποστηρίξει χωρικά δεδομένα. Έτσι επιλέχθηκε η επέκταση Spatialite [26], η οποία μπορεί να υποστηρίξει και χωρικά δεδομένα.

### 4.3.2 *Spatialite*

Η Spatialite [26] ενεργοποιεί χωρικά την SQLite ώστε να υποστηρίζει χωρικά δεδομένα συμβατά με τις προδιαγραφές του OGC [18] (Open Geospatial Consortium). Η Spatialite μπορεί να υποστηρίζει τα πρότυπα WKT (Well Known Text) και WKB (Well Known Binary) [36]. Επειδή η γεωμετρία είναι ένας πολύπλοκος τύπος δεδομένων η Spatialite μπορεί να καταλάβει τα χωρικά δεδομένα μέσω ενός κατάλληλα μορφοποιημένου κειμένου που ακολουθεί τα πρότυπα των WKT ή WKB.

Υποστηρίζει επίσης όλες τις γνωστές γεωμετρίες όπως σημεία, γραμμές, πολύγωνα, αλλά και σύνθετες γεωμετρίες. Μέσω της Spatialite ο προγραμματιστής μπορεί να υλοποιεί χωρικές συναρτήσεις όπως οι AsText(), GeomFromText(), Area(), PointN(), αλλά και τοπολογικούς τελεστές όπως Overlaps(), Touches(), Union(), Buffer(), Intersects() κ.τ.λ.

Η Spatialite μπορεί επίσης να υποστηρίζει πλήρως τα χωρικά μεταδεδομένα από τις προδιαγραφές του OGC καθώς και εναλλακτικά πρότυπα αναπαράστασης γεωμετρίας - EWKT, GML, KML, και GeoJSON. Σημαντικό είναι ότι μπορεί να υποστηρίξει εισαγωγή και εξαγωγή, από και προς shapefiles, καθώς είναι γνωστό ότι πολλά χωρικά δεδομένα που κυκλοφορούν στο Διαδίκτυο είναι σε μορφή αρχείων shapefile. Επιπλέον ο προγραμματιστής έχει τη δυνατότητα αλλαγής προβολικού συστήματος των χωρικών δεδομένων μέσω της βιβλιοθήκης PROJ.4 [23] και των κωδικών EPSG [6].

Τέλος πολύ σημαντικό είναι ότι μπορεί να υποστηρίζει χωρικά ευρετήρια βασισμένα στην επέκταση της SQLite με το RTree [24]. Το RTree είναι ένα ειδικό ευρετήριο, το οποίο σχεδιάστηκε για να μπορεί ο προγραμματιστής να εκτελεί χωρικά ερωτήματα στη χωρική βάση δεδομένων και να ανακτά γρήγορα τα αποτελέσματα. Κάθε είσοδος στο ερώτημα μπορεί να είναι ένα ορθογώνιο με ελάχιστες και μέγιστες συντεταγμένες. Έτσι, δίνοντας ως είσοδο στο ερώτημα ένα ορθογώνιο, με το R-Tree ο προγραμματιστής είναι σε θέση να βρει γρήγορα όλες τις καταχωρήσεις που περιέχονται στο εσωτερικό του ορθογωνίου ή το επικαλύπτουν. Το RTree υλοποιείται ως ένας εικονικός πίνακας με περιττό αριθμό στηλών μεταξύ 3 και 11. Η πρώτη στήλη είναι πάντα ένας 64-bit ακέραιος και θεωρείται πρωτεύον κλειδί. Οι άλλες στήλες είναι ζεύγη, ένα ζεύγος ανά διάσταση, που περιέχει τις ελάχιστες και μέγιστες τιμές συντεταγμένων για την εν λόγω διάσταση.

## 4.4 *PostgreSQL με την επέκταση PostGIS*

Σε συνδυασμό με την τοπική βάση δεδομένων θα πρέπει να χρησιμοποιηθεί ένα ολοκληρωμένο σύστημα διαχείρισης βάσεων δεδομένων για την κεντρική βάση δεδομένων, η οποία θα διαχειρίζεται τα δεδομένα όλων των χρηστών που κοινοποιούν τα ταξίδια τους στον

Server. Η PostgreSQL [22] με την επέκταση PostGIS [21] είναι μια εξαιρετική επιλογή μιας και μπορεί να διαχειριστεί τόσο περιγραφικά όσο και χωρικά δεδομένα.

#### **4.4.1 PostgreSQL**

Όπως και η SQLite, έτσι και η PostgreSQL [22] είναι ένα δωρεάν και ανοιχτού κώδικα λογισμικό, το οποίο μπορεί να παραμετροποιηθεί και να επεκταθεί. Λόγω της μακρόχρονης ύπαρξης της και της αποδεδειγμένα προηγμένης αρχιτεκτονικής της έχει ισχυρή φήμη όσον αφορά την αξιοπιστία, την ακεραιότητα και την ορθότητα των δεδομένων.

Ένας ακόμα σημαντικός λόγος για τον οποίο επιλέχτηκε η PostgreSQL είναι ότι μπορεί να τρέξει σε όλα τα κύρια λειτουργικά συστήματα (Linux, UNIX, AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64, τα Windows). Έτσι, σε περίπτωση που χρειάζεται να αλλάξει ο server και αυτός περιέχει διαφορετικό λειτουργικό σύστημα από τον προηγούμενο να μπορούν εύκολα να μεταφερθούν και τα δεδομένα των χρηστών.

Η PostgreSQL έχει επίσης πλήρη υποστήριξη των εκφράσεων της SQL, καθώς επίσης και των ξένων κλειδιών, των ενώσεων, των όψεων, των σκανδαλιστών (triggers), και των διαδικασιών (stored procedures). Περιλαμβάνει το μεγαλύτερο μέρος των SQL τύπων δεδομένων, όπως INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, και TIMESTAMP. Επίσης, υποστηρίζει αποθήκευση δυαδικών αντικειμένων, όπως εικόνες, ήχο και βίντεο. Όλα αυτά έπαιξαν σημαντικό ρόλο ώστε να επιλεγεί η PostgreSQL ως κεντρική βάση δεδομένων.

Για να αποθηκευτούν όλα τα δεδομένα όλων των χρηστών η κεντρική βάση θα έπρεπε να είχε μεγάλα όρια στην αποθήκευση δεδομένων. Πράγματι, η PostgreSQL επιτρέπει απεριόριστο μέγεθος για μια βάση δεδομένων: 32TB για το μέγεθος ενός πίνακα, 1.6 TB για το μέγιστο μέγεθος μιας γραμμής και απεριόριστο αριθμό γραμμών ανά πίνακα.

Η PostgreSQL υποστηρίζει λειτουργικά ευρετήρια που μπορούν να χρησιμοποιήσουν οποιαδήποτε από τις μεθόδους αποθήκευσης, όπως B-tree, R-tree, hash πίνακες, ή GIST δένδρο. Το ευρετήριο GIST [10] (Generalized Search Tree) αποτελεί ένα προηγμένο σύστημα, το οποίο συγκεντρώνει ένα ευρύ φάσμα διαφορετικών αλγόριθμων ταξινόμησης και αναζήτησης, συμπεριλαμβανομένου του B - tree, B+ - δένδρο, R-δένδρο, κ.α.

#### **4.4.2 Πλήρης αναζήτηση κειμένων στην PostgreSQL**

Επειδή στην εφαρμογή ο χρήστης θα πρέπει να έχει τη δυνατότητα να κάνει αναζήτηση στην κεντρική βάση δεδομένων σε χιλιάδες κείμενα άλλων χρηστών με λέξεις κλειδιά, η βάση θα πρέπει να υποστηρίζει κάποιον τρόπο, έτσι ώστε να γίνεται αυτή η αναζήτηση

εύκολα και κυρίως γρήγορα. Η PostgreSQL μέσω της πλήρους αναζήτησης κειμένων [28] παρέχει τη δυνατότητα να εντοπίζονται έγγραφα φυσικής γλώσσας που ικανοποιούν ένα ερώτημα, και προαιρετικά να ταξινομούνται κατά συνάφεια προς το ερώτημα. Ο πιο κοινός τύπος της αναζήτησης είναι να βρεθούν όλα τα έγγραφα που περιέχουν συγκεκριμένους όρους ομοιότητας με το ερώτημα. Η απλούστερη αναζήτηση θεωρεί το ερώτημα ως ένα σύνολο λέξεων και την ομοιότητα ως τη συχνότητα των λέξεων του ερωτήματος μέσα στο έγγραφο. Σαν έγγραφο στην PostgreSQL θεωρείται ένα πεδίο σε ένα πίνακα που είναι τύπου κείμενο (text). Έτσι η PostgreSQL χρησιμοποιεί ένα συγκεκριμένο τύπο δεδομένων για τέτοια κείμενα και σε συνδυασμό με τη δημιουργία ευρετηρίου η αναζήτηση γίνεται γρήγορη. Ένα τέτοιο ευρετήριο επιτρέπει τα κείμενα να είναι προεπεξεργασμένα στα εξής στάδια:

- *Ανάλυση των εγγράφων σε tokens.* Αναγνωρίζονται και χωρίζονται τα έγγραφα σε αριθμούς, λέξεις, πολύπλοκες λέξεις, e-mails.
- *Μετατροπή των tokens σε λεξήματα (lexemas).* Ένα λέξημα είναι ένα αλφαριθμητικό όπως και το token μόνο που έχει κανονικοποιηθεί ώστε διαφορετικές μορφές της ίδιας λέξης να μετρώνται σε μια κατηγορία. Για παράδειγμα, μια τέτοια κανονικοποίηση είναι ο μη διαχωρισμός πεζών και κεφαλαίων γραμμάτων ή ακόμα και η κατάληξη es του πληθυντικού στα αγγλικά να μην μετρείται σαν διαφορετική λέξη από αυτή του ενικού.
- *Αποθήκευση βελτιστοποιημένων προεπεξεργασμένων κειμένων για εύκολη αναζήτηση.* Κάθε έγγραφο μπορεί να αναπαρασταθεί ως ένας ταξινομημένος πίνακας από κοινωνικοποιημένα λεξήματα. Μαζί με το λέξημα αποθηκεύονται και οι θέσεις όπου εμφανίζεται το λέξημα, έτσι ώστε να αξιοποιηθεί αυτή η πληροφορία συχνότητας εμφανίσεων μιας λέξης σε ένα κείμενο και έτσι να ανέβει στην κατάταξη των αποτελεσμάτων του ερωτήματος.

Στη δημιουργία ενός τέτοιου ευρετηρίου ο προγραμματιστής μπορεί να ορίσει ποιες λέξεις θέλει να αποφεύγει στη καταμέτρηση της συχνότητας των λέξεων, όπως π.χ. άρθρα ή συνδέσμους. Ο τύπος δεδομένων που χρησιμοποιεί η PostgreSQL για την αποθήκευση τέτοιων επεξεργασμένων εγγράφων είναι ο tsvector και στην ουσία είναι ο πίνακας με τα λεξήματα και τις θέσεις όπου αυτά εμφανίζονται. Για να μετατραπεί ένα κείμενο σε δεδομένο τύπου tsvector, η PostgreSQL χρησιμοποιεί τη συνάρτηση `to_tsvector(columnname)`, όπου το `columnname` είναι το όνομα της στήλης που περιέχει τα κείμενα. Ο τύπος tsquery χρησιμοποιείται για να συγκρίνει τις λέξεις κλειδιά του ερωτήματος με τον πίνακα tsvector που περιέχει τα λεξήματα. Για να μετατραπούν οι λέξεις κλειδιά σε δεδομένα τύπου tsquery, η PostgreSQL χρησιμοποιεί τη συνάρτηση `to_tsquery`. Ο πιο σημαντικός τελεστής σύγκρισης των λέξεων κλειδίων με τον πίνακα tsvector είναι το «`@@`». Ένα tsquery περιέχει τους όρους αναζήτησης, οι οποίοι πρέπει να είναι ήδη ομαλοποιημένα λεξήματα, και μπορούν να

συνδυαστούν με τελεστές όπως AND, OR, και NOT. Έτσι πχ αν στη βάση δημιουργηθεί μια στήλη τύπου tsvector που περιέχει τα κείμενα μια άλλης στήλης χωρισμένα σε λεξήματα, με τη δημιουργία ενός ευρετηρίου στη συγκεκριμένη στήλη, η αναζήτηση θα μπορεί να γίνει πολύ γρήγορη. Έτσι π.χ. το παρακάτω ερώτημα:

```
SELECT title
FROM notes
WHERE textsearchable_index_col @@ to_tsquery('london &
museum')
ORDER BY last_mod_date DESC LIMIT 10;
```

Αν υποθέσουμε ότι η στήλη textsearchable\_index\_col είναι τύπου tsvector και περιέχει τα λεξήματα των σχολίων μιας στήλης του πίνακα notes και πάνω σε αυτή τη στήλη έχει δημιουργηθεί ένα ευρετήριο, θα μας επιστρέψει τους 10 πρώτους τίτλους των σχολίων του πίνακα notes που περιέχουν ταυτόχρονα τις λέξεις κλειδιά «london» και «museum» ταξινομημένα βάσει της ημερομηνίας επεξεργασίας σε φθίνουσα σειρά.

Επίσης στην PostgreSQL μπορούν να καταταγούν τα αποτελέσματα τέτοιων ερωτημάτων. Μια τέτοια κατάταξη προσπαθεί να μετρήσει πόσο σχετικά είναι τα έγγραφα που αναζητήθηκαν με το ερώτημα, έτσι ώστε όταν υπάρχουν πολλά σχετικά αποτελέσματα να έρχεται πρώτο το πιο σχετικό. Η PostgreSQL διαθέτει δυο προκαθορισμένες συναρτήσεις κατάταξης: ts\_rank και την ts\_rank\_cd. Αυτές οι συναρτήσεις λαμβάνουν υπόψη τη συχνότητα εμφάνισης μια λέξης, την εγγύτητα των λέξεων κλειδιών του ερωτήματος και πόσο σημαντικό είναι το μέρος του εγγραφου όπου εμφανίζονται οι λέξεις. Έτσι για παράδειγμα το παρακάτω ερώτημα:

```
SELECT Visits.name, ts_rank(Notes.SearchableText,
to_tsquery('english','london | museum')) AS rank
FROM Visits
INNER JOIN Notes ON Visits.Id = Notes.VisitsId
WHERE Notes.SearchableText @@ to_tsquery('english',' london
| museum ')
ORDER BY rank DESC
LIMIT 10;
```

Αν υποθεθεί ότι στον πίνακα Notes υπάρχει η στήλη SearchableText, η οποία είναι τύπου tsvector και έχουν εισαχθεί δεδομένα μιας άλλης στήλης τύπου κείμενο του πίνακα Notes αφού πρώτα τα έχουν μετατραπεί σε tsvector δεδομένα, χρησιμοποιώντας τη συνάρτηση to\_tsvector, τα αποτελέσματα που θα επιστρέψει είναι οι 10 πρώτες εγγραφές ταξινομημένες ανά κατάταξη σε φθίνουσα σειρά, οι οποίες θα περιέχουν τα ονόματα των επισκέψεων από τα

σχόλια που έχουν γίνει στις συγκεκριμένες επισκέψεις και περιέχουν τις λέξεις κλειδιά «London» ή «museum».

#### 4.4.3 *PostGIS*

Στην κεντρική βάση πρέπει επίσης να αποθηκεύονται χωρικά δεδομένα και αυτό μπορεί να γίνει μέσω της επέκτασης που διαθέτει η PostgreSQL, το PostGIS [21]. Το PostGIS προσθέτει υποστήριξη για γεωγραφικά αντικείμενα στην PostgreSQL, ώστε να μπορεί να χρησιμοποιηθεί ως χωρική βάση δεδομένων για τα γεωγραφικά συστήματα πληροφοριών (GIS). Έτσι μέσω του ευρετηρίου GIST, το οποίο μπορεί και αξιοποιεί το RTree, ο προγραμματιστής μπορεί να δημιουργήσει χωρικά ευρετήρια που διευκολύνουν την χωρική αναζήτηση στη βάση δεδομένων. Όπως και η Spatialite, έτσι και η PostGIS υποστηρίζει:

- τα πρότυπα WKT και WKB.
- τις γνωστές γεωμετρίες όπως σημεία, γραμμές, πολύγωνα, αλλά και σύνθετες γεωμετρίες όπως συλλογές από απλές γεωμετρίες.
- Συναρτήσεις και τελεστές όπως AsText(), GeomFromText(), Area(), PointN(), αλλά και οι Overlaps(), Touches(), Union(), Buffer(), Intersects() κ.τ.λ.
- Απόδοση και εισαγωγή διανυσματικών δεδομένων διαφόρων μορφών όπως KML, GML, GeoJSON, GeoHash και WKT χρησιμοποιώντας την SQL.
- Απόδοση Raster δεδομένων διαφόρων μορφών όπως GeoTIFF, PNG, JPG, NetCDF.
- Εισαγωγή και εξαγωγή, από και προς shapefiles.
- Πολλές επιλογές προβολικών και γεωγραφικών συστημάτων.

### 4.5 *Microsoft .NET Web Services*

Τα web services που χρειάστηκαν για την εξυπηρέτηση των αιτημάτων από τη συσκευή προς τον server και αντίστροφα υλοποιήθηκαν με τεχνολογίες Microsoft .NET [17]. Η Microsoft δίνει στον προγραμματιστή πολλά εργαλεία για να υλοποιήσει εύκολα και γρήγορα τις συναρτήσεις που θα εξυπηρετούν τα αιτήματα από και προς τον Server. Επίσης με την γλώσσα C# που χρησιμοποιήθηκε στην ανάπτυξη των web services, ο προγραμματιστής έχει όλες τις δυνατότητες μιας αντικειμενοστρεφούς γλώσσας προγραμματισμού, όποτε μπορεί να οργανώσει τον κώδικά του με τις κατάλληλες κλάσεις για την εξυπηρέτηση των αιτημάτων. Τα αιτήματα αυτά είναι συναρτήσεις που εκτελούν εκφράσεις SQL στην κεντρική βάση δεδομένων, όπως πχ εισαγωγή, επεξεργασία και διαγραφή ενός ταξιδιού. Επειδή τα web services της Microsoft φιλοξενούνται σε server ο

όποιος διαθέτει λειτουργικό σύστημα Windows είναι πολύ εύκολο για τον προγραμματιστή να τα εγκαταστήσει σε έναν τέτοιο server και να τα ενημερώνει όποτε υπάρχει κάποια σημαντική αλλαγή. Επειδή τα web services που υλοποιούνται με τεχνολογίες Microsoft χρησιμοποιούν το SOAP [25] ως πρωτόκολλο μεταφοράς δεδομένων, είναι εύκολο να χρησιμοποιηθούν από οποιαδήποτε εφαρμογή που μπορεί να τρέχει σε οποιοδήποτε περιβάλλον. Τέλος, αυτά τα web services έχουν τη δυνατότητα να επιστρέφουν τα αποτελέσματα σε τέτοια μορφή, έτσι ώστε να μπορούν να διαβαστούν από οποιοδήποτε μηχάνημα και στη συγκεκριμένη περίπτωση από κινητή συσκευή. Τέτοιες μορφές είναι αρχεία τύπου xml ή Json.

## **4.6 Χάρτες Google και OpenStreetMap**

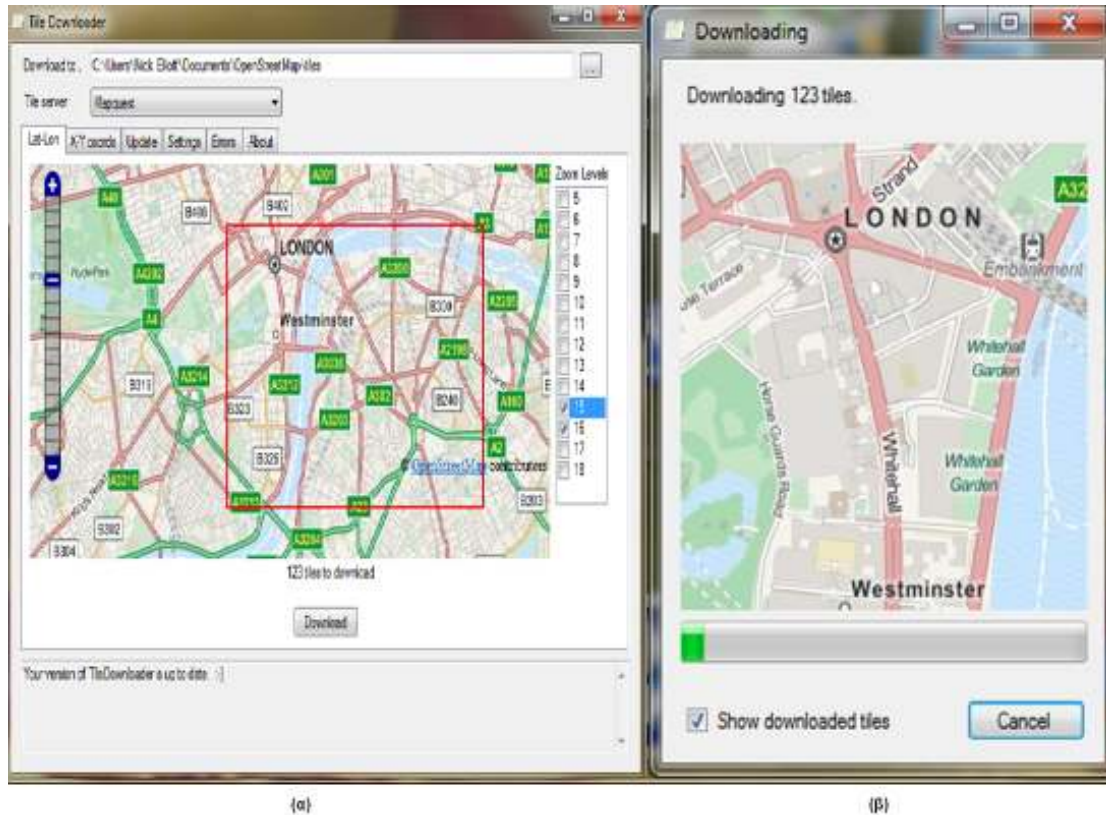
Η εφαρμογή θα έχει τη δυνατότητα να παρέχει τους χάρτες στον χρήστη είτε βρίσκεται σε online λειτουργία, δηλαδή συνδεδεμένος στο Διαδίκτυο, είτε βρίσκεται σε offline λειτουργία.

Για την online λειτουργία επιλέχθηκαν οι χάρτες της Google [11]. Η Google διαθέτει ενημερωμένους χάρτες για σχεδόν όλες τις περιοχές της γης, όποτε ο χρήστης βλέπει επικαιροποιημένα στοιχεία για μια περιοχή που θέλει να επισκεφτεί. Επίσης αυτοί οι χάρτες έχουν δημιουργηθεί από την Google, όπως και το android λειτουργικό σύστημα και αυτό σημαίνει ότι μπορούν να συνδυαστούν άψογα με εφαρμογές που τρέχουν σε λειτουργικό σύστημα android. Η Google έχει υλοποιήσει ένα API (Application Programming Interface) [11] με πάρα πολλές συναρτήσεις, τις οποίες μπορεί ο προγραμματιστής να χρησιμοποιήσει μέσα από android εφαρμογές και να διαχειριστεί τους χάρτες. Μπορεί να προβάλει χωρικά δεδομένα, όπως σημεία, γραμμές και πολύγωνα, που έχει αποθηκευμένα στην χωρική του βάση, να δημιουργήσει μέσω των χαρτών χωρικά δεδομένα και να τα αποθηκεύσει στη βάση του. Όλα αυτά μπορούν να γίνουν εύκολα μέσω του API που προσφέρεται στους προγραμματιστές. Επίσης προσφέρει αρκετούς τρόπους απεικόνισης των δεδομένων που θέλει να προβάλει στους χάρτες.

Για την offline λειτουργία επιλέχθηκαν τα Tiles του OpenStreetMap [19]. Τα tiles αυτά διατίθενται δωρεάν από το OpenStreetMap και κάθε προγραμματιστής μπορεί να τα κατεβάσει και να τα αξιοποιήσει στις εφαρμογές του. Υπάρχουν διάφορα εργαλεία που διατίθενται δωρεάν στο Διαδίκτυο και καθένας μπορεί να τα χρησιμοποιήσει για να κατεβάσει τα tiles του OpenStreetMap. Σε αυτήν την εργασία χρησιμοποιήθηκε το εργαλείο TileDownloader [29]. Όπως φαίνεται στην παρακάτω εικόνα 30, ο χρήστης μπορεί να

διαλέξει μια περιοχή από το χάρτη και να κατεβάσει τα σχετικά tiles στο δίσκο του. Η δομή των φακέλων στον οποίο έχουν κατέβει τα tiles είναι της μορφής:

- Mapquest/zoom/x/y.png



**Εικόνα 30: (α) Διεπαφή εργαλείου TileDownloader επιλογής περιοχής των tiles και (β) διεπαφή εξέλιξης αποθήκευσης των tiles στον δίσκο**

Αφού ο προγραμματιστής κατεβάσει τα tiles από μια περιοχή που τον ενδιαφέρει, μπορεί να τα αποθηκεύσει στη μνήμη της συσκευής με την ίδια μορφή, ώστε να μπορούν να διαβαστούν από την εφαρμογή του κινητού. Στη συνέχεια και πάλι το API της Google δίνει τη δυνατότητα στο προγραμματιστή να διαχειριστεί τα δικά του tiles, γράφοντας ελάχιστο κώδικα, επεκτείνοντας την ήδη έτοιμη βιβλιοθήκη που διαθέτει το android για διαχείριση των tiles. Το OpenStreetMap διαθέτει δωρεάν για κατέβασμα επικαιροποιημένους χάρτες από σχεδόν όλες τις περιοχές της γης, κάτι που ήταν πολύ σημαντικό για να επιλεγούν για την offline λειτουργία τα tiles του OpenStreetMap.



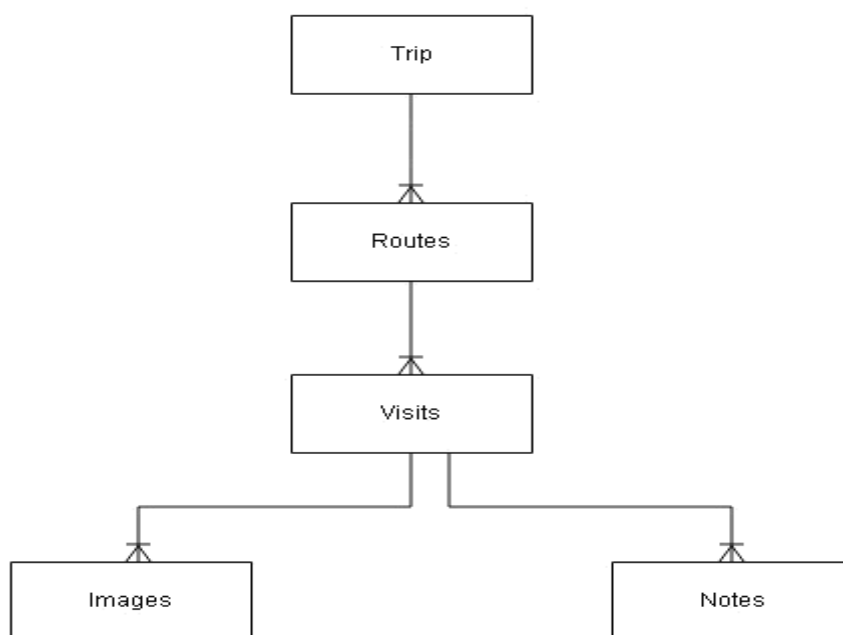
## 4.7 Δημιουργία τοπικής βάσης δεδομένων

Όπως αναφέρθηκε και στην εισαγωγή, σκοπός της εργασίας είναι η τήρηση ταξιδιωτικού ημερολογίου σε κινητή συσκευή. Ο χρήστης μιας τέτοιας συσκευής θα μπορεί στο ταξίδι του να δημιουργεί όσες διαδρομές θέλει. Θα μπορεί να χωρίσει το ταξίδι του σε διαδρομές πχ ανά ημέρα. Στην κάθε διαδρομή θα μπορεί να κάνει όσες επισκέψεις θέλει σε οποιοδήποτε σημείο της περιοχής που έχει επισκεφτεί. Επίσης σε κάθε σημείο που επισκέπτεται θα μπορεί να κάνει όσα σχόλια και να κρατεί όσες σημειώσεις θέλει. Τέλος σε κάθε σημείο που επισκέπτεται θα μπορεί να βγάζει και να αποθηκεύει όσες φωτογραφίες επιθυμεί.

Έτσι πρέπει να πειλαμβάνει τις παρακάτω οντότητες:

- Ταξίδι (Trip)
- Διαδρομές (Routes)
- Επισκέψεις (Visits)
- Σημειώσεις (Notes)
- Εικόνες (Images)

Μια πρώτη εικόνα των οντοτήτων αυτών και το πώς συσχετίζονται φαίνεται παρακάτω στην εικόνα 16:



**Εικόνα 16: Οι βασικές οντότητες με τις συσχείσεις τους.**

Πιο συγκεκριμένα, στην οντότητα «**Ταξίδι**» θα αποθηκεύονται περιγραφικές πληροφορίες για το ταξίδι, όπως:

- Τίτλος (τύπου κείμενο)
- Σύντομη περιγραφή (τύπου κείμενο)
- Ημερομηνία δημιουργίας (τύπου ημερομηνία)
- Ημερομηνία τερματισμού (τύπου ημερομηνία)

Για λόγους ευκολίας χρησιμοποιήθηκε αγγλική ορολογία, όπου με την παρακάτω εντολή δημιουργήθηκε ο πίνακας στην Spatialite, δημιουργώντας μαζί τα απαραίτητα πρωτεύον κλειδί:

```
CREATE TABLE Trip (  
  Id INTEGER PRIMARY KEY AUTOINCREMENT,  
  Title TEXT,  
  ShortDesc TEXT,  
  DateCreated DATETIME,  
  DateClosed DATETIME);
```

Στη συνέχεια, εφόσον ο χρήστης θέλει να χωρίσει το ταξίδι του σε ημέρες ή να κρατεί ξεχωριστές διαδρομές σε ένα ταξίδι και όχι να φαίνεται όλο του το ταξίδι σαν μια ενιαία διαδρομή, θα πρέπει να δημιουργηθεί ένας πίνακας «**διαδρομές**», στον οποίο θα αποθηκεύονται τόσο περιγραφικές όσο και χωρικές πληροφορίες. Ο πίνακας θα έχει τα παρακάτω πεδία:

- Τίτλος διαδρομής (τύπου κείμενο)
- Σύντομη περιγραφή (τύπου κείμενο)
- Ημερομηνία δημιουργίας (τύπου ημερομηνία)
- Τροχιά (τύπου γραμμή): εδώ θα αποθηκεύεται η γεωμετρία των επισκέψεων του χρήστη σε μια διαδρομή ως πολυγραμμή (polyline).

Εδώ δημιουργήθηκε και ένα χωρικό ευρετήριο για την εύκολη και γρήγορη αναζήτηση δεδομένων, που είναι βασισμένη σε χωρικά κριτήρια. Η εντολή δημιουργίας του χωρικού ευρετηρίου παρατίθεται στο παράρτημα 8.1.

Επίσης, σε κάθε διαδρομή ο χρήστης θα πρέπει να έχει τη δυνατότητα να κάνει όσες επισκέψεις θέλει σε σημεία που τον ενδιαφέρουν. Για να μπορέσει να αποθηκευτεί αυτή η

πληροφορία θα πρέπει να δημιουργηθεί ένας πίνακας «επισκέψεις», στον οποίο θα αποθηκεύονται χωρικές και περιγραφικές πληροφορίες. Τα πεδία του πίνακα αυτού θα είναι:

- Τίτλος επίσκεψης (τύπου κείμενο)
- Κατηγορία επίσκεψης (τύπου κείμενο): πρόκειται για μουσείο, αξιοθέατο, εστιατόρια, ξενοδοχείο κ.τ.λ.
- Ημερομηνία δημιουργίας (τύπου ημερομηνία)
- Γεωγραφική Θέση (τύπου σημείο): θα αποθηκεύεται η γεωμετρία του σημείου που επισκέφτηκε ο χρήστης.

Δημιουργήθηκε και εδώ ένα χωρικό ευρετήριο για την εύκολη και γρήγορη αναζήτηση δεδομένων, που είναι βασισμένη σε χωρικά κριτήρια.

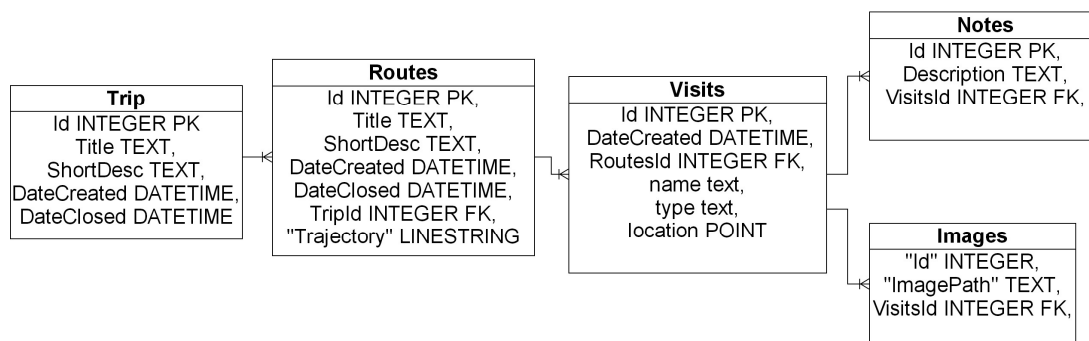
Σε κάθε επίσκεψή του ο χρήστης καλό θα ήταν να αποθηκεύει όσα σχόλια και να κρατεί όσες σημειώσεις. Για να αποθηκευτεί αυτή η πληροφορία, θα πρέπει να δημιουργηθεί ένας πίνακας «Σημειώσεις» και να έχει τα παρακάτω πεδία:

- Περιγραφή-σχόλιο (τύπου κείμενο).

Όπως και με τις σημειώσεις, σημαντικό θα ήταν ο χρήστης να μπορεί να βγάζει και να αποθηκεύει όσες περισσότερες φωτογραφίες θέλει από κάθε επίσκεψη του, έτσι ώστε να έχει την καλύτερη δυνατή ανάμνηση από το σημείο που επισκέφτηκε. Για να αποθηκευτεί μια τέτοια πληροφορία, θα πρέπει να δημιουργηθεί ένας πίνακας «Εικόνες» και θα έχει τα παρακάτω πεδία:

- Το μονοπάτι του αρχείου της εικόνας από την μνήμη του κινητού (τύπου κείμενο).  
Επιλέχτηκε ο τρόπος της αποθήκευσης της εικόνας στη μνήμη της συσκευής και η αποθήκευση της διαδρομής της εικόνας από τη μνήμη στη βάση, έτσι ώστε να μην επιβαρύνεται η βάση δεδομένων με εικόνες.

Έτσι μαζί με τα πεδία πλέον το τελικό σχήμα της βάσης φαίνεται στην παρακάτω εικόνα 17:



**Εικόνα 17: Τελικό σχήμα της βάσης με τα πεδία των πινάκων**

Ένας πρόσθετος πίνακας αφορά «**Σημεία ενδιαφέροντος**» που διατίθενται ελεύθερα στο Διαδίκτυο. Αυτός ο πίνακας θα έχει τα παρακάτω πεδία:

- Όνομα σημείου ενδιαφέροντος (τύπου κείμενο)
- Κατηγορία σημείου ενδιαφέροντος (τύπου κείμενο), πχ μουσείο αρχαιολογικό χώρος, εστιατόριο, ξενοδοχείο κ.τ.λ.
- Γεωγραφική Θέση (τύπου σημείο). Εδώ αποθηκεύεται η γεωμετρία του σημείου ενδιαφέροντος.

Ομοίως με την παραπάνω εντολή CREATE TABLE έχουν δημιουργηθεί και οι υπόλοιποι πίνακες στην Spatialite καθώς και τα απαραίτητα ευρετήρια. Για περισσότερες πληροφορίες σχετικά με τη δημιουργία της βάσης στην Spatialite ο αναγνώστης παραπέμπεται στο παράρτημα 8.1 διαχείριση sqlite – spatialite.

## **4.8 Δημιουργία κεντρικής βάσης δεδομένων**

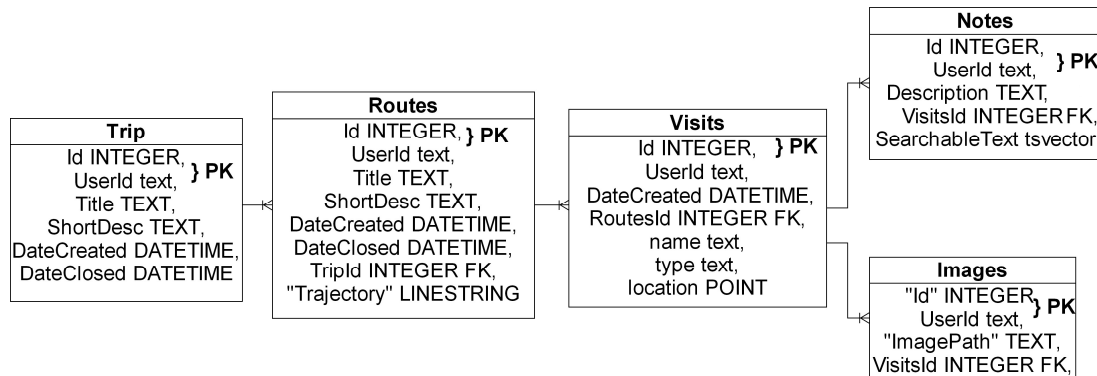
Για τη δημιουργία της κεντρικής βάσης δεδομένων επιλέχθηκε η PostgreSQL με την επέκταση της PostGIS. Η βάση δεδομένων θα πρέπει να έχει παρόμοια χαρακτηριστικά με την τοπική βάση δεδομένων στις κινητές συσκευές άρα και τη δυνατότητα αποθήκευσης χωρικών και περιγραφικών πληροφοριών. Σε αυτή τη βάση δεδομένων, αποθηκεύονται πληροφορίες από όλες τις συσκευές των χρηστών, οπότε θα πρέπει να υπάρχει ένας τρόπος να διακρίνονται τα ταξίδια που κοινοποιούνται ανά χρήστη. Ένας τρόπος για να γίνει αυτό, είναι μαζί με τις πληροφορίες του ταξιδιού που αποθηκεύονται από την συσκευή στον server να αποθηκεύεται και ένα αναγνωριστικό της συσκευής σε κάθε οντότητα, έτσι ώστε να γίνεται αντιληπτό πάντα από ποια συσκευή προέρχονται οι πληροφορίες που κοινοποιούνται.

Για λόγους ευκολίας χρησιμοποιήθηκε και πάλι αγγλική ορολογία, οπότε με την παρακάτω εντολή δημιουργήθηκε ο πίνακας «Trip» στην PostgreSQL:

```
CREATE TABLE Trip
(
  Id integer NOT NULL,
  UserId text NOT NULL,
  Title text,
  ShortDesc text,
  DateCreated timestamp without time zone NOT NULL,
  DateClosed timestamp without time zone,
  CONSTRAINT Trip_Unique_ID UNIQUE (Id, UserId)
```

) ;

Οι εντολές δημιουργίας ολόκληρης της βάσης δεδομένων στην PostgreSQL ο αναγνώστης μπορεί να τις βρει στο παράρτημα 8.2. Το σχήμα της βάσης δεδομένων τροποποιήθηκε ελαφρώς, ώστε να είναι δυνατόν η βάση να διαχειρίζεται ταξίδια πολλών χρηστών. Το σχήμα της βάσης φαίνεται στην παρακάτω εικόνα 18:



**Εικόνα 18: Σχήμα της κεντρικής βάσης δεδομένων στην PostgreSQL**

Επίσης όπως περιγράφηκε στην ενότητα 4.4 σε πίνακα «Notes» έχει προστεθεί μια ακόμα στήλη που είναι τύπου tsvector και περιέχει τα λεξήματα του πεδίου Description του πίνακα Notes, καθώς επίσης και τα λεξήματα των πεδίων name και type του πίνακα Visits για να γίνεται εύκολη και γρήγορη αναζήτηση με λέξεις κλειδιά. Για να εισαχθούν αυτά τα λεξήματα στην στήλη SearchableText του πίνακα Notes δημιουργήθηκε ένας trigger, ο οποίος έπειτα από κάθε εισαγωγή στον πίνακα Notes ενημερώνει την στήλη SearchableText με τα δεδομένα των πεδίων description, name και type σε μορφή tsvector. Την εντολή δημιουργίας του trigger ο αναγνώστης μπορεί να τη βρει στο παράρτημα 8.2.

## 4.9 Δημιουργία Web services

Τα web services ήταν απαραίτητα για την εξυπηρέτηση αιτημάτων από την συσκευή προς τον server. Τα αιτήματα αυτά δεν είναι άλλα από εντολές SQL στην κεντρική βάση δεδομένων. Οι εντολές αυτές για την εξυπηρέτηση αυτών των αιτημάτων περιλαμβάνουν τις βασικές εντολές

- ανάκτησης (SELECT)
- προσθήκης (INSERT)
- διαγραφής (DELETE)

Έχουν δημιουργηθεί συναρτήσεις δημιουργίας εγγράφων σε όλους τους πίνακες «Trip», «Routes», «Visits», «Notes», «Images», επίσης συναρτήσεις ανάκτησης δεδομένων από τους παραπάνω πίνακες, καθώς και συναρτήσεις διαγραφής δεδομένων από τους παραπάνω πίνακες, έτσι ώστε να εξυπηρετούνται όλα τα πιθανά αιτήματα προς τον server.

Έτσι π.χ., όταν ο χρήστης θέλει να κοινοποιήσει ένα ταξίδι του στον server, η συσκευή καλεί τις κατάλληλες συναρτήσεις του web service και κοινοποιεί το ταξίδι στον server. Το πρώτο βήμα είναι να καλέσει η συσκευή τις κατάλληλες συναρτήσεις για να διαγράψει τυχόν ταξίδι που υπάρχει στο server και έχει τα ίδια χαρακτηριστικά, δηλαδή έχει το ίδιο id ταξιδιού και το ίδιο userid της συσκευής που το καλεί, έτσι ώστε να ανεβάσει το επικαιροποιημένο πλέον ταξίδι. Όποτε πρέπει να διαγραφούν τα δεδομένα του συγκεκριμένου ταξιδιού από τους πίνακες «Trip», «Routes», «Visits», «Notes» και «Images» της PostgreSQL. Έτσι η συσκευή καλεί τις συναρτήσεις deleteTrip, deleteTripRoutes, deleteTripVisits, deleteTripNotes, deleteTripImages και σβήνει τα δεδομένα του συγκεκριμένου ταξιδιού του συγκεκριμένου χρήστη. Π.χ., για να διαγραφούν τα δεδομένα του πίνακα «Trip» καλείται η συνάρτηση deleteTrip του web service από την εφαρμογή του κινητού κάνοντας κλήση στο παρακάτω λινκ:

- <http://ServerIP/TravelogueWS/service.asmx/deleteTripNotes?Id=tripid&UserId=DeviceId>

Αν το ταξίδι με τα συγκεκριμένα χαρακτηριστικά δεν υπάρχει στον server, τότε οι συγκεκριμένες συναρτήσεις δεν θα κάνουν καμία ενέργεια και δεν θα σβήσουν κανένα δεδομένο από την PostgreSQL. Το επόμενο βήμα είναι να ανεβάσει το ταξίδι με τα χαρακτηριστικά που έχει επιλέξει από τη συσκευή. Όποτε πρέπει να ανεβάσει τα δεδομένα από τους πίνακες «Trip», «Routes», «Visits», «Notes» και «Images» που έχει επιλέξει από την συσκευή στους αντίστοιχους πίνακες της PostgreSQL. Έτσι, καλεί τις συναρτήσεις insertTrip, insertRoute, insertVisit, insertNote, insertImage του web service για να αποθηκεύσει τα δεδομένα του ταξιδιού στους πίνακες της PostgreSQL. Π.χ. για την συνάρτηση insertTrip γίνεται κλήση στο παρακάτω λινκ:

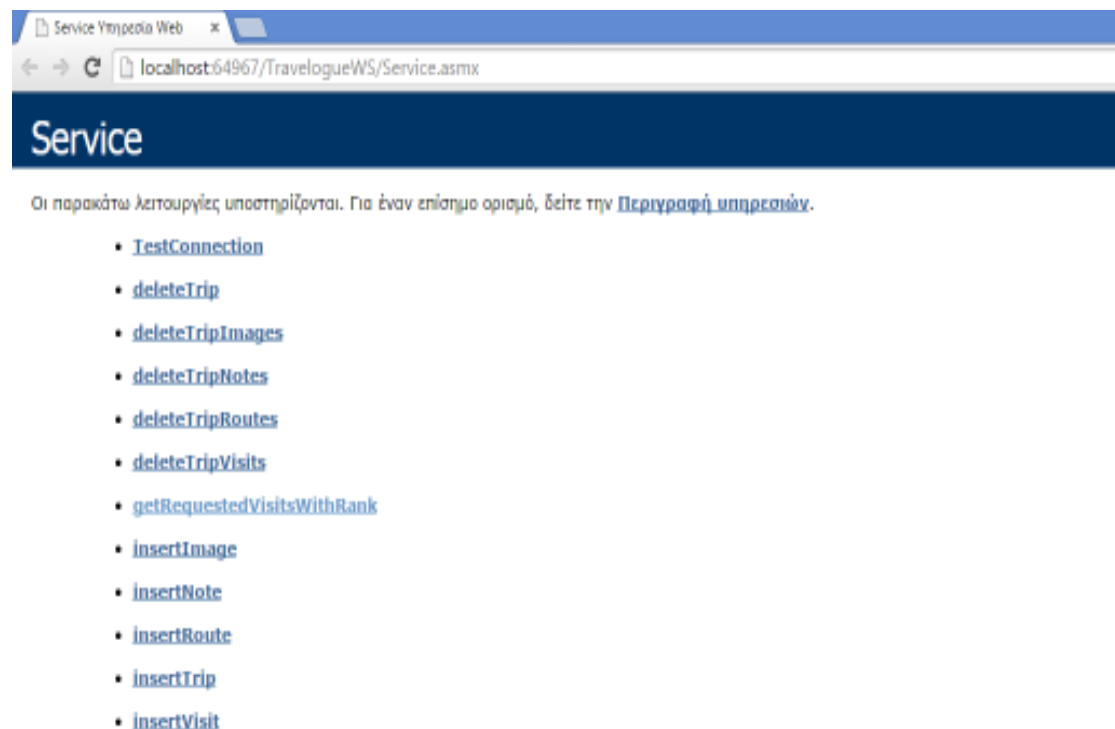
- <http://ServerIP/TravelogueWS/service.asmx/insertTrip?Id=tripid&UserId=DeviceId&Title=title&ShortDesc=shortDesc&DateCreated=dateCreated&DateClosed=dateClosed>

Επίσης, ο χρήστης της συσκευής έχει τη δυνατότητα να κάνει αναζητήσεις από την εφαρμογή της συσκευής στον server με λέξεις κλειδιά. Όταν κάνει μια τέτοια αναζήτηση στον server, η συσκευή καλεί την συνάρτηση getRequestedVisitsWithRank του web service, τότε εκτελείται ένα SELECT στην PostgreSQL και επιστρέφονται στον χρήστη όλα τα δεδομένα που σχετίζονται με τις λέξεις κλειδιά που ζήτησε ο χρήστης σε μορφή xml, ταξινομημένα σύμφωνα με τη συνάρτηση ts\_rank που περιγράφηκε στην υποενότητα 4.4.2.

Αν υποθεθεί πως ο χρήστης κάνει μια αναζήτηση με λέξεις κλειδιά «acropolis museum», το ερώτημα αυτό φαίνεται παρακάτω:

```
SELECT Visits.Id,  
Visits.name,  
Visits.type,  
ts_rank(Notes.SearchableText,  
to_tsquery('english','acropolis | museum')) AS rank ,  
ST_X(Visits.location) AS lon,  
ST_Y(Visits.location) AS lat  
FROM Visits  
INNER JOIN Notes ON Visits.Id = Notes.VisitsId  
WHERE Notes.SearchableText @@  
to_tsquery('english','acropolis | museum')  
ORDER BY rank DESC  
LIMIT 10;
```

Τα αποτελέσματα έχουν τη μορφή xml αρχείου και ο αναγνώστης μπορεί να το βρει στο παράρτημα 8.3. Επίσης μια λίστα με τις συναρτήσεις του web service φαίνεται στην παρακάτω εικόνα 27 και η υλοποίησή τους υπάρχει στο παράρτημα 8.3.



**Εικόνα 27: Τα υλοποιημένα Web Services που επιτρέπουν αλληλεπίδραση με την κεντρική βάση δεδομένων.**

# 5

## *Υπόδειγμα χρήσης της εφαρμογής*

Στη συνέχεια θα αναπτυχθεί ένα σενάριο για να γίνει αντιληπτό στον αναγνώστη πως συνδυάζονται όλες αυτές οι τεχνολογίες μαζί ώστε να δημιουργηθεί το τελικό σύστημα. Μέσω των διεπαφών του χρήστη και χρησιμοποιώντας τεχνολογίες των κινητών συσκευών, όπως κάμερα, Gps και χάρτες αποθηκεύονται και ανακτώνται χωρικά και περιγραφικά δεδομένα από και προς την τοπική βάση του κινητού. Επίσης το σύστημα αξιοποιεί την τεχνολογία των web services για να κοινοποιήσει αλλά και να ανακτήσει δεδομένα από την κεντρική βάση δεδομένων του Server. Όλες οι διεπαφές χρήστη σχεδιαστήκαν μέσω των εργαλείων του android studio.





**Εικόνα 19: Αρχικό μενού**

Με την έναρξη της εφαρμογής εμφανίζεται μια κεντρική οθόνη όπου υπάρχουν τέσσερις επιλογές που καθοδηγούν το χρήστη:

- *Δημιουργία νέου ταξιδιού.*
- *Εν εξελίξει ταξίδι.*
- *Αναζήτηση.* Εδώ ο χρήστης μπορεί να αναζητήσει με κριτήρια, ταξίδια που έχει αποθηκευμένα στην συσκευή του ή να αναζητήσει επισκέψεις άλλων χρηστών από τον Server από περιοχές που τον ενδιαφέρουν.
- *Ρυθμίσεις της εφαρμογής.*

Η οθόνη αυτή φαίνεται στην εικόνα 19.

## Δημιουργία ταξιδιού

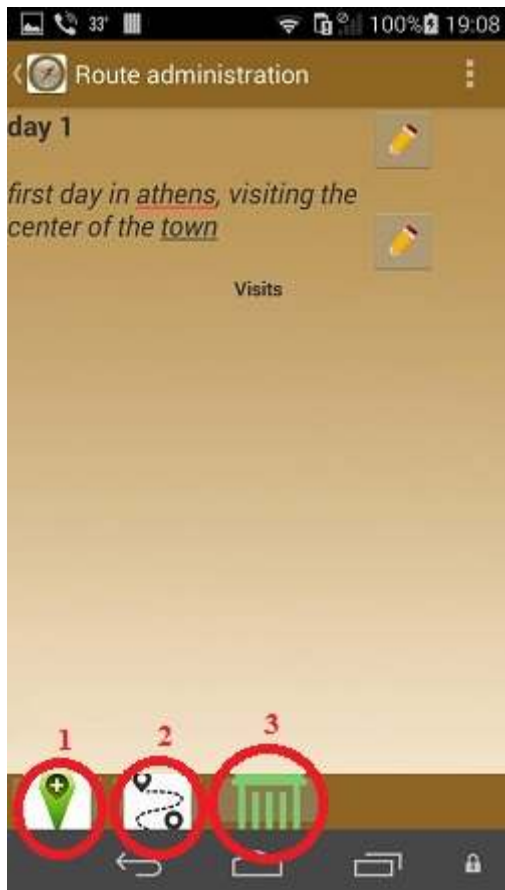


**Εικόνα 20:** Καρτέλα ταξιδιού

Πατώντας το κουμπί «New Trip» από την διεπαφή της εικόνας 19, ξεκινάει η διαδικασία για να καταγραφεί ένα νέο ταξίδι του χρήστη. Στην οθόνη μπορεί να συμπληρώσει τις περιγραφικές πληροφορίες του ταξιδιού που θα αποθηκευτούν στην τοπική βάση στον πίνακα «Trip». Έστω ότι ο χρήστης πραγματοποιεί ένα ταξίδι στην Αθήνα, οπότε η οθόνη αυτή θα είναι όπως στην εικόνα 20.

Η οθόνη αυτή εμφανίζει επίσης σε λίστα τις διαδρομές που έχει κάνει (αλλά σε αυτή τη φάση δεν έχουν ακόμη δημιουργηθεί). Επίσης στο κάτω μέρος της οθόνης υπάρχει ένα μενού με ενέργειες που μπορεί να κάνει ο χρήστης. Τα αριθμημένα κουμπιά αντιστοιχούν στις παρακάτω ενέργειες:

- 1) Δημιουργία νέας διαδρομής.
- 2) Προβολή όλων των διαδρομών του ταξιδιού στο χάρτη.
- 3) Τέλος ταξιδιού.
- 4) Διαγραφή ταξιδιού.
- 5) Κοινοποίηση ταξιδιού στον Server.



**Εικόνα 21: Καρτέλα διαδρομής**

Πατώντας το κουμπί 1 από την διεπαφή της εικόνας 20 εμφανίζεται η οθόνη για να δημιουργήσει την πρώτη του διαδρομή στο ταξίδι και να αποθηκεύσει τις περιγραφικές πληροφορίες στον πίνακα «Routes». Έτσι η οθόνη αυτή φαίνεται στην εικόνα 21.

Όπως φαίνεται στην εικόνα 21 ο χρήστης αποθήκευσε τις περιγραφικές πληροφορίες της πρώτης του διαδρομής, την έχει ονομάσει «day 1» και στην σύντομη περιγραφή έχει γράψει ότι επισκέπτεται το κέντρο της Αθήνας. Από αυτή την οθόνη όπως φαίνεται από τα αριθμημένα κουμπιά ο χρήστης μπορεί να κάνει τις παρακάτω ενέργειες:

- 1) *Δημιουργία επίσκεψης στη διαδρομή.*
- 2) *Προβολή της συγκεκριμένης διαδρομής στο χάρτη.*
- 3) *Διαγραφή της διαδρομής με όλες τις επισκέψεις του χρήστη.*

Σε αυτή την οθόνη ο χρήστης μπορεί να δει σε λίστα και τις επισκέψεις που έχει κάνει στη συγκεκριμένη διαδρομή (δεν έχουν εισαχθεί ακόμα επισκέψεις).



**Εικόνα 22: Καρτέλα επίσκεψης**

Οι ενέργειες που μπορεί να κάνει ο χρήστης από την οθόνη της εικόνας 22 είναι σύμφωνα με τα αριθμημένα κουμπιά οι παρακάτω:

- 1) *Να γράψει μια σημείωση για την επίσκεψη.*
- 2) *Να βγάλει μια φωτογραφία από την επίσκεψη στο συγκεκριμένο σημείο.*
- 3) *Να κάνει προβολή του σημείου στο χάρτη και*
- 4) *Να διαγράψει το σημείο από την διαδρομή του.*

Σε αυτήν την οθόνη ο χρήστης μπορεί να δει και τις σημειώσεις που έχει κρατήσει από την συγκεκριμένη επίσκεψη, καθώς επίσης και τις φωτογραφίες που έβγαλε (όχι ακόμη για την επίσκεψη στην Ακρόπολη).



**Εικόνα 23: Δημιουργία επίσκεψης στο χάρτη**

Έτσι στη συνέχεια ο χρήστης πατώντας το κουμπί 1 από την διεπαφή της εικόνας 22 του εμφανίζεται η οθόνη για να κάνει κάποιο σχόλιο ενώ με το κουμπί 2 από την διεπαφή της εικόνας 22 μπορεί να χρησιμοποιήσει την κάμερα του κινητού του για να βγάλει κάποια φωτογραφία από το σημείο που επισκέφτηκε. Οι δυο αυτές οθόνες φαίνονται στην παρακάτω εικόνα 24:

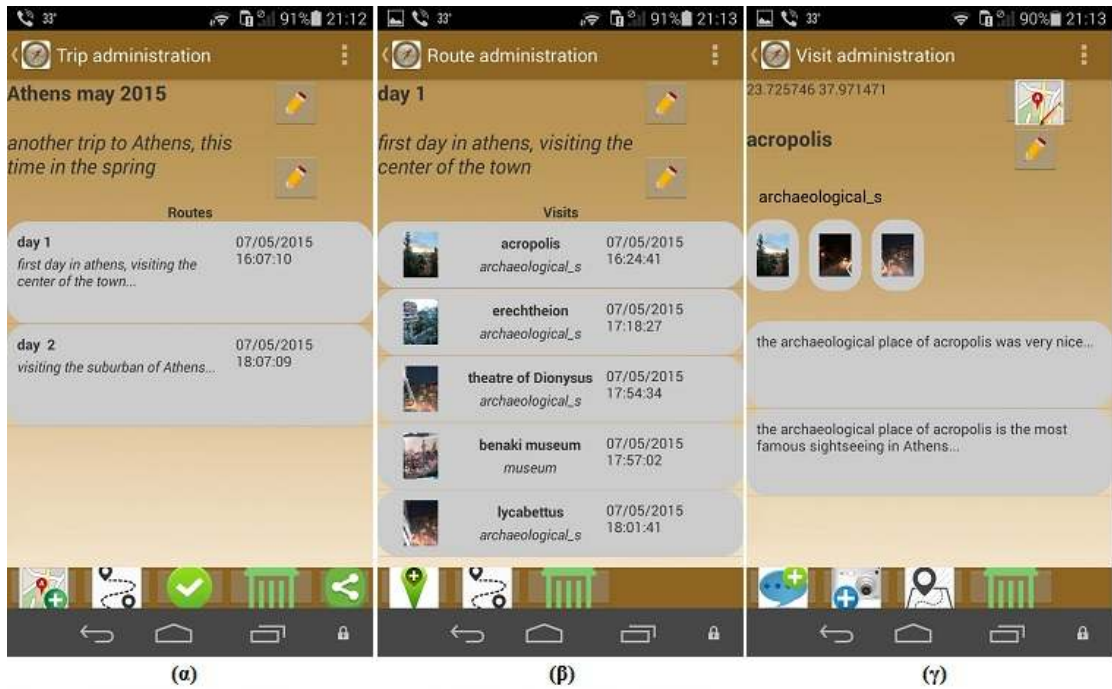
Την χωρική πληροφορία που υπάρχει στην εικόνα 22, την ανέκτησε πατώντας του κουμπί 5 της εικόνας 22 μετά το οποίο του εμφανίστηκε ο χάρτης με τα σημεία ενδιαφέροντος του πίνακα «POIS» της τοπικής βάσης, που υπάρχουν γύρω του και επέλεξε την Ακρόπολη, όπως φαίνεται στην εικόνα 23.

Στον κύκλο 2 φαίνεται η θέση του χρήστη, η οποία έχει εντοπιστεί από την εφαρμογή μέσω του GPS της συσκευής. Πατώντας το κουμπί 1 από την διεπαφή της εικόνας 23 ο χρήστης έχει επιλέξει τον αρχαιολογικό χώρο Ακρόπολη και επιστρέφει στην οθόνη της εικόνας 22.



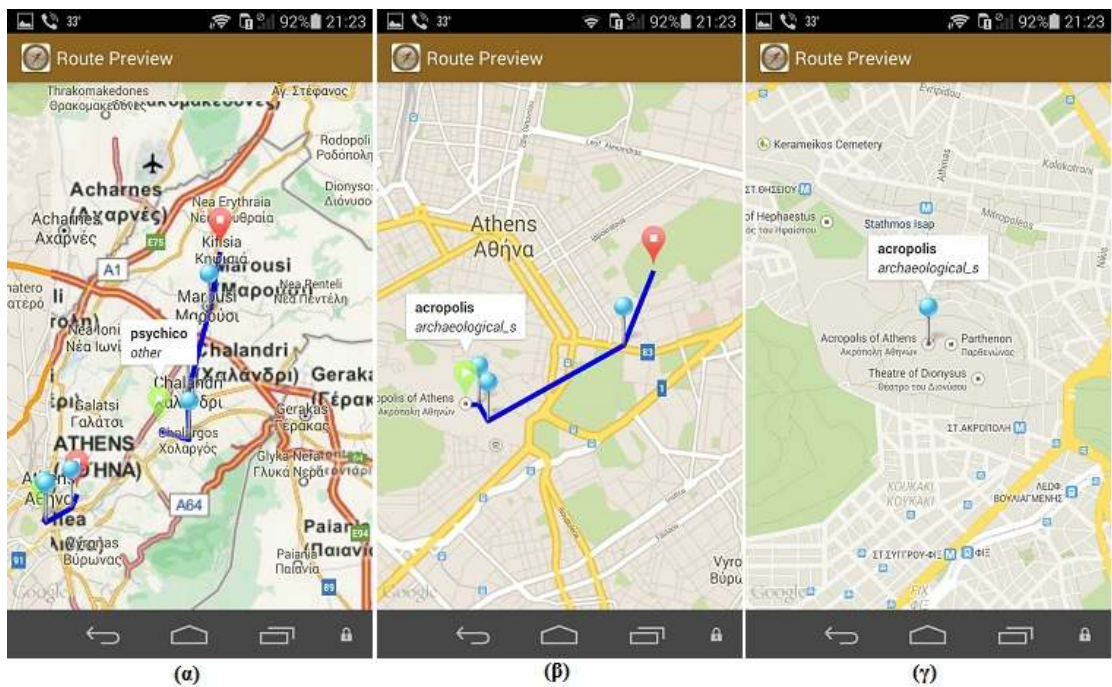
**Εικόνα 24: (α) Καρτέλα σημείωσης και (β) Καρτέλα φωτογραφίας**

Έτσι ο χρήστης έχει δημιουργήσει σε αυτό το ταξίδι του μια πρώτη διαδρομή κάνοντας μια πρώτη επίσκεψη στην Ακρόπολη, βγάζοντας μια φωτογραφία από την επίσκεψη του και κάνοντας ένα πρώτο σχόλιο. Με τον ίδιο τρόπο ο χρήστης μπορεί να κάνει όσα σχόλια θέλει στην επίσκεψη του, να βγάλει όσες φωτογραφίες θέλει από το σημείο που επισκέπτεται, να κάνει όσες επισκέψεις θέλει στην διαδρομή του και τέλος να δημιουργήσει όσες διαδρομές θέλει στο ταξίδι του. Έτσι σε ένα ολοκληρωμένο ταξίδι οι παραπάνω οθόνες συμπληρωμένες με όλες τις πληροφορίες φαίνονται στην εικόνα 25:



**Εικόνα 25: (α) Καρτέλα ταξιδιού, (β) Καρτέλα διαδρομής και (γ) Καρτέλα επίσκεψης**

Στις εικόνες 20,21 και 22 όπως αναφέρθηκε και πιο πάνω υπάρχει μια ενέργεια που προβάλλει όλες τις διαδρομές του ταξιδιού, μία διαδρομή ενός ταξιδιού και μία επίσκεψη μιας διαδρομής. Πρόκειται για την ίδια οθόνη, στην οποία κάθε φορά αποστέλλονται τα κατάλληλα δεδομένα, δηλαδή αν πρόκειται για πολλές τροχιές, για μία τροχιά ή ένα σημείο. Αυτή η οθόνη φαίνεται στην εικόνα 26 με όλες τις εκδοχές:



**Εικόνα 26: (α) Προβολή όλων των διαδρομών, (β) Προβολή της διαδρομής «day 1» και (γ) Προβολή της επίσκεψης στην Ακρόπολη από τη διαδρομή «day 1»**

Τέλος στην εικόνα 20 και στην καρτέλα του ταξιδιού, η επιλογή του κουμπιού με αριθμό 5 δίνει τη δυνατότητα στο χρήστη να ανεβάσει το ταξίδι του στην κεντρική βάση δεδομένων. Έτσι ο χρήστης, έχοντας πατήσει το συγκεκριμένο κουμπί, αφού έχει συμφωνήσει να κοινοποιήσει το ταξίδι του στον server ώστε να το δουν και οι υπόλοιποι χρήστες, καλεί τις κατάλληλες συναρτήσεις (deleteTrip, deleteTripRoutes, deleteTripVisits, deleteTripNotes, deleteTripImages). Με ορίσματα το id του ταξιδιού και το userid της συσκευής από το web service που υπάρχει στο Διαδίκτυο σε κάποιο web server, διαγράφει το ταξίδι που μπορεί να είναι ήδη ανεβασμένο στον server και στη συνέχεια καλεί τις συναρτήσεις insertTrip, insertRoute, insertVisit, insertNote, insertImage και κοινοποιεί το επικαιροποιημένο ταξίδι του στη κεντρική βάση δεδομένων.

### **Εν εξελίξει ταξίδι**

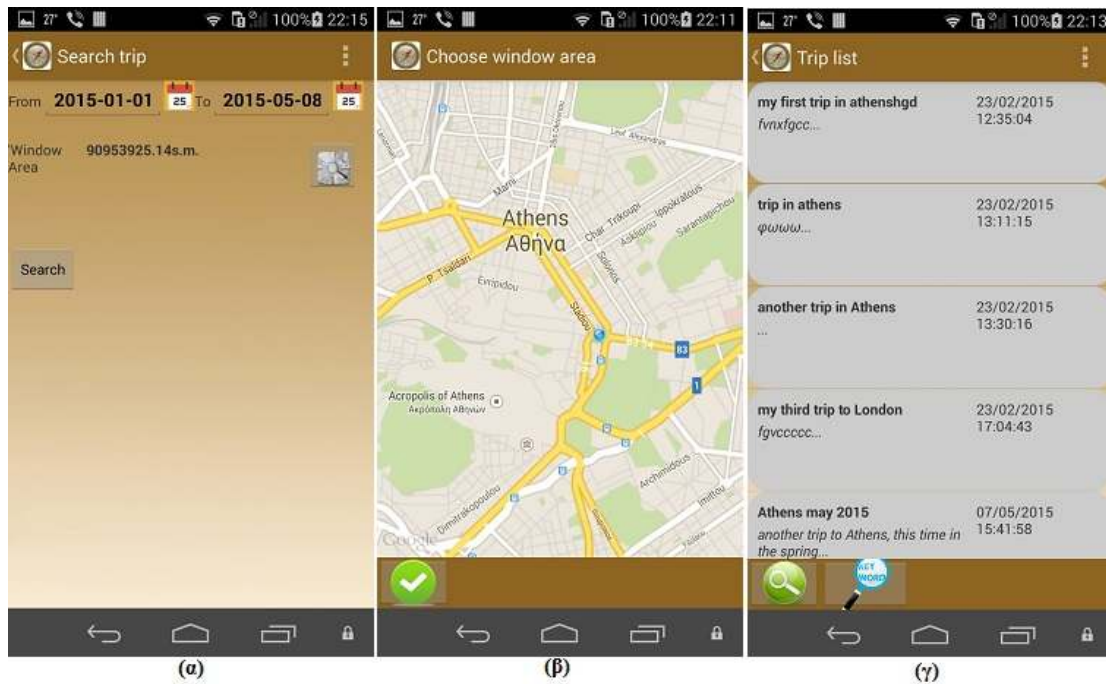
Ουσιαστικά στο εν εξελίξει ταξίδι υπάρχουν οι ίδιες ακριβώς οθόνες με τη δημιουργία ταξιδιού, απλά πατώντας το κουμπί «ongoing trip» του αρχικού μενού, στις διεπαφές φορτώνεται από την τοπική βάση δεδομένων το τελευταίο χρονικά ταξίδι που δεν το έχει κλείσει ο χρήστης. Αυτό δηλαδή που δεν έχει ορίσει ημερομηνία τερματισμού «DateClosed» στην τοπική βάση δεδομένων.

### **Αναζήτηση**

Ο χρήστης έχει τη δυνατότητα μέσω της εφαρμογής να κάνει *χωροχρονική* αναζήτηση των ταξιδιών του που έχει αποθηκευμένα στη συσκευή, αλλά και αναζήτηση με *λέξεις κλειδιά* στην κεντρική βάση δεδομένων.

Για να κάνει αναζήτηση με χωροχρονικά κριτήρια έχει φτιαχτεί μια διεπαφή στην οποία ο χρήστης μπορεί να επιλέξει, τις *ημερομηνίες* που πραγματοποίησε τα ταξίδια του αλλά και μια *περιοχή στο χάρτη* από την οποία πέρασε κατά τη διάρκεια των ταξιδιών του. Στη συνέχεια τα αποτελέσματα αυτά της αναζήτησης εμφανίζονται σε μια λίστα. Οι τρεις αυτές οθόνες φαίνονται στην παρακάτω εικόνα 28:





**Εικόνα 28: (α) Οθόνη χωροχρονικής αναζήτησης, (β) Επιλογή περιοχής αναζήτησης και (γ) Τα αποτελέσματα της αναζήτησης ταξιδιών σε λίστα**

Αφού πατήσει ο χρήστης το κουμπί search της οθόνης της εικόνας 28 (α), τότε εκτελείται ένα χωροχρονικό ερώτημα. Χρονικό με τις ημερομηνίες που έχει επιλέξει και χωρικό, χρησιμοποιώντας τον τελεστή intersect για να δει ποια ταξίδια τέμνουν την περιοχή που έχει επιλέξει. Τα αποτελέσματα όπως αναφέρθηκε και στην προηγούμενη παράγραφο φαίνονται στην εικόνα 28 (γ). Ανάμεσα στα αποτελέσματα είναι και το ταξίδι που δημιούργησε ο χρήστης σε αυτήν την ενότητα. Το ερώτημα που συντάσσεται αυτόματα βάσει των επιλογών που κάνει ο χρήστης από την οθόνη, είναι το εξής:

```
SELECT distinct trip.id,
trip.title,
substr(trip.shortDesc,1,75) || '...' as ShortDesc1,
trip.DateCreated,
trip.DateClosed
FROM trip
LEFT JOIN routes ON trip.id=routes.tripid
WHERE trip.Datecreated >=date('2015-01-01')
AND trip.Datecreated <=date('2015-05-08')
AND ST_INTERSECTS(routes.trajectory,
ST_GeomFromText('POLYGON( (23.683062717318535
```

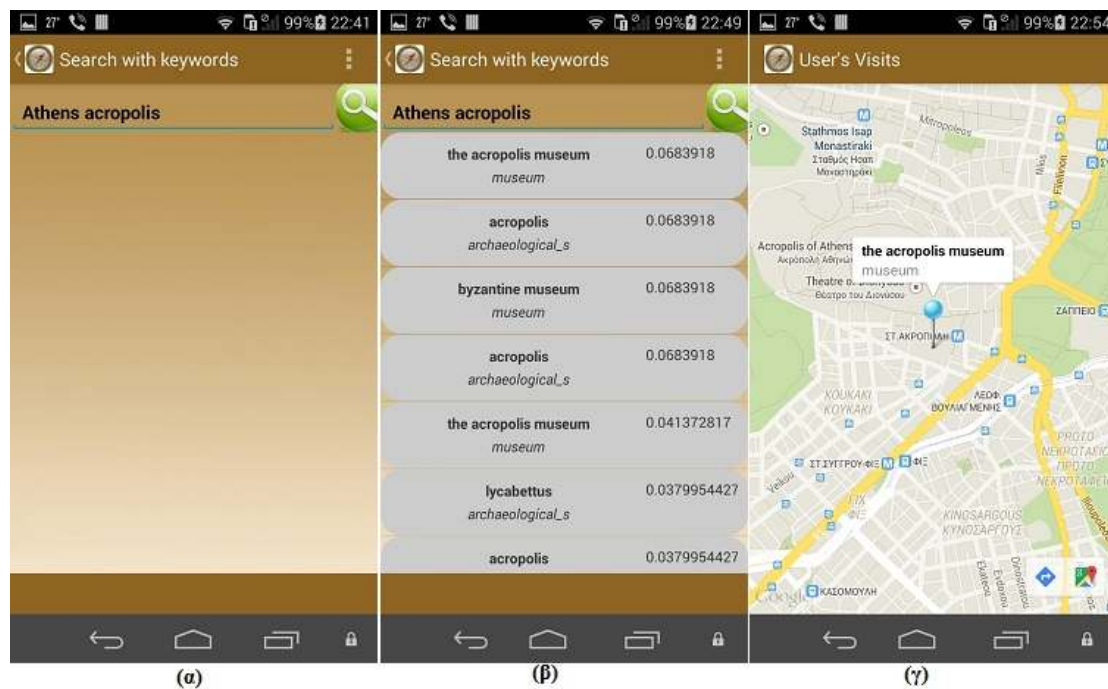
```

37.94362136630896,      23.76799788326025      37.94362136630896,
23.76799788326025      38.03101703651244,      23.683062717318535
38.03101703651244,      23.683062717318535      37.94362136630896
))',4326) );

```

```
ORDER BY trip.datecreated
```

Επιπλέον ο χρήστης έχει τη δυνατότητα να κάνει μια αναζήτηση στο server με λέξεις κλειδιά για να δει τι έχουν κάνει άλλοι χρήστες. Όπως περιγράφηκε και στην ενότητα 4.8 με τη δημιουργία του κατάλληλου ευρετηρίου για αναζήτηση λέξεων κλειδιών σε κείμενα άλλων χρηστών εκτελεί αυτό το ερώτημα και λαμβάνονται τα αποτελέσματα σε λίστα ταξινομημένη σύμφωνα με τη βαθμολογία που βγήκε εκτελώντας την συνάρτηση tsrank της PostgreSQL που αναπτύχτηκε στην ενότητα 4.4, η οποία δείχνει στο χρήστη πόσο ταιριάζουν οι λέξεις κλειδιά με τις σημειώσεις των επισκέψεων άλλων χρηστών. Στην συνέχεια πατώντας πάνω στην κάθε επίσκεψη της λίστας μπορεί να δει στο χάρτη την επίσκεψη και το σχόλιο του χρήστη. Οι οθόνες αυτές φαίνονται την παρακάτω εικόνα 29:



**Εικόνα 29: (α) Αναζήτηση με λέξεις κλειδιά, (β) Αποτελέσματα αναζήτησης και (γ) Προβολή στο χάρτη της πρώτης επίσκεψης κάποιου χρήστη από τη λίστα.**

Το ερώτημα που υπέβαλλε ο χρήστης όπως προκύπτει από τις παραπάνω οθόνες φαίνεται παρακάτω:

```
SELECT Visits.Id,
```

```
Visits.name,  
    Visits.type,  
    TS_RANK(Notes.SearchableText, TO_TSQUERY('english', 'Athens |  
acropolis)) AS rank ,  
    ST_X(Visits.location) AS lon,  
    ST_Y(Visits.location) AS lat  
FROM Visits  
INNER JOIN Notes ON Visits.Id = Notes.VisitsId  
WHERE Notes.SearchableText @@ TO_TSQUERY('english', 'Athens  
| acropolis ')  
ORDER BY rank DESC  
LIMIT 10;
```

# 6

## *Επίλογος*

Συνοψίζοντας, στα πλαίσια αυτής της μεταπτυχιακής αναπτύχθηκε ένα σύστημα με το οποίο κάθε ταξιδιώτης μπορεί να τηρεί ένα ηλεκτρονικό ημερολόγιο μέσω κινητής συσκευής. Δημιουργήθηκε μια εφαρμογή για κινητές συσκευές που τρέχουν σε λειτουργικό android, με την οποία ο κάθε ταξιδιώτης μπορεί να αποθηκεύει και να διαχειρίζεται τα ταξίδια του. Επιπλέον δημιουργήθηκε μια κεντρική βάση δεδομένων στην οποία κάθε χρήστης μπορεί (εφόσον το επιθυμεί) να ανεβάσει το ταξίδι του, με σκοπό να μπορούν οι χρήστες της εφαρμογής να βλέπει ο ένας τα ταξίδια του άλλου. Για να αναπτυχτεί αυτό το σύστημα χρησιμοποιήθηκαν διάφορες τεχνολογίες, όπως το android OS, το android studio, SQLite με την επέκταση Spatialite για την υποστήριξη χωρικών δεδομένων, η PostgreSQL με την επέκταση PostGIS, τα Microsoft.Net Web Services. Στο κεφάλαιο 2 παρουσιάστηκαν κάποιες σχετικές εφαρμογές που υπάρχουν και αναδείχθηκαν ομοιότητες και διαφορές με την παρούσα εφαρμογή. Επίσης παρουσιάστηκαν σχετικές εργασίες χωρικού πληθοπορισμού και συλλογής γεωγραφικού περιεχομένου από το πλήθος. Στο κεφάλαιο 3 αναπτύχθηκε η αρχιτεκτονική του συστήματος. Στο κεφάλαιο 4 τα εργαλεία και τα συστατικά που χρειάστηκαν για να δημιουργηθεί το σύστημα. Τέλος στο κεφάλαιο 5, παρουσιάστηκε ένα ολοκληρωμένο σενάριο χρήσης με τις λειτουργίες και τις διαδικασίες του συστήματος.

### **6.1 Συμπεράσματα**

Τα ηλεκτρονικά ταξιδιωτικά ημερολόγια και γενικότερα οι εφαρμογές ταξιδιωτικού περιεχομένου φαίνεται να γνωρίζουν μεγάλη ανάπτυξη τα τελευταία χρόνια και θα αναπτυχθούν ακόμα περισσότερο στο μέλλον. Αυτό οφείλεται σε μεγάλο βαθμό στην εξέλιξη της τεχνολογίας και των τηλεπικοινωνιών. Όσο οι κινητές συσκευές εξελίσσονται σε φορητούς υπολογιστές και η δημοτικότητά τους μεγαλώνει, τόσο πιο δημοφιλή θα γίνουν και τα ταξιδιωτικά ιστολόγια και ιστοσελίδες. Πλέον η χρήση κινητών συσκευών είναι

κομμάτι της καθημερινότητας του ανθρώπου, με την κατάλληλη εφαρμογή ο χρήστης μπορεί να κάνει πολλές λειτουργίες. Αυτό ισχύει και στις εφαρμογές ταξιδιωτικού περιεχομένου. Η δημιουργία αναμνήσεων από ταξίδια είναι μια ιδιαίτερα ευχάριστη ψυχαγωγία για τον άνθρωπο, έτσι η δημιουργία κατάλληλων εφαρμογών σε κινητές συσκευές είναι πολύ σημαντική. Η δημιουργία αναμνήσεων σε πραγματικό χρόνο είναι χρήσιμη για τους ταξιδιώτες, γιατί έτσι τους δίνεται η δυνατότητα να μην ξεχάσουν να κρατήσουν καμία από τις αναμνήσεις που τους ενδιαφέρουν.

Σημαντικό ρόλο στο να γίνουν αυτές οι εφαρμογές δημοφιλείς, έπαιξε και η χρήση χαρτών στο Διαδίκτυο και στις κινητές συσκευές. Πλέον όλες οι συσκευές έχουν ενσωματωμένους χάρτες από μεγάλες εταιρείες, όπως η Google, που δίνουν τη δυνατότητα στο χρήστη να περιηγηθεί στην περιοχή που θέλει. Έτσι η ενσωμάτωση αυτών των χαρτών σε εφαρμογές ταξιδιωτικού περιεχομένου έπαιξαν μεγάλο ρόλο στην εξέλιξη των εφαρμογών αλλά και να ανεβάσουν τη δημοτικότητα τους. Οι εφαρμογές δίνουν τη δυνατότητα στο χρήστη να δημιουργήσει τους δικούς του χάρτες και τις δίκες του διαδρομές, κάτι που είναι πολύ σημαντικό στην δημιουργία της καλύτερης δυνατής ανάμνησης από ταξίδια.

Επίσης, οι εφαρμογές κοινωνικών δικτύων είναι πολύ δημοφιλείς. Οι εφαρμογές ταξιδιωτικών ημερολογίων, όπως και η συγκεκριμένη, περιέχουν αυτό το στοιχείο, καθώς ο χρήστης μπορεί και μοιράζεται σημαντικά γεγονότα από τα ταξίδια του με άλλους. Μπορεί και μαθαίνει τί έκαναν άλλοι χρήστες σε μέρη που τον ενδιαφέρουν, αλλά μπορεί και να βοηθήσει άλλους χρήστες με τα δικά του σχόλια στα ταξίδια τους.

Τέλος, η εφαρμογή σχετίζεται με τον πληθοπορισμό και ειδικά με τον χωρικό πληθοπορισμό. Ο πληθοπορισμός είναι ένα πεδίο της επιστήμης στο οποίο έχει γίνει αρκετή ερευνά αφού δίνεται η δυνατότητα στο πλήθος να δημιουργήσει χωρικά δεδομένα. Η συλλογή σημείων ενδιαφέροντος θα μπορούσε να είναι ένα έργο που θα δινόταν σε μια εταιρεία να το υλοποιήσει. Με αυτήν την εφαρμογή όμως αυτό μπορεί να γίνει επώνυμα ή ανώνυμα από το κοινό. Έτσι, δημιουργείται μια βάση δεδομένων με περιγραφικές και χωρικές πληροφορίες που θα μπορούσε να αποδειχτεί αρκετά χρήσιμη και για την ανάπτυξη και άλλων έργων. Με αυτό το τρόπο συλλέγονται πληροφορίες με προτιμήσεις χρηστών και έτσι μπορεί να δημιουργηθεί ένα προφίλ ανά χρήστη, που σε κάποια άλλη εφαρμογή θα μπορούσε να αξιοποιηθεί. Π.χ. ανάλογα με το προφίλ του χρήστη θα μπορούσε να δημιουργηθεί μια εφαρμογή που να του προτείνει προϊόντα και υπηρεσίες που πιθανόν να τον ενδιαφέρουν.

## 6.2 Μελλοντικές επεκτάσεις

Μια σημαντική επέκταση που θα μπορούσε να γίνει στο υπάρχον σύστημα θα ήταν να δημιουργηθούν κάποιες διεπαφές χρήστη σε web μορφή, έτσι ώστε να μπορεί να διαχειρίζεται τις πληροφορίες που ανεβάζει στην κεντρική βάση δεδομένων, αλλά και να βλέπει και πληροφορίες άλλων χρηστών μέσα από το site.

Μια άλλη σημαντική επέκταση είναι να δημιουργηθεί ένα προφίλ χρήστη στην εφαρμογή της κινητής συσκευής. Έτσι ο κάθε χρήστης να μπορεί στις ρυθμίσεις της εφαρμογής να σημειώνει τις προτιμήσεις του και η εφαρμογή να τις λαμβάνει υπόψη, ώστε ο χρήστης να βλέπει σημεία ενδιαφέροντος και προτάσεις άλλων χρηστών που ταιριάζουν με το προφίλ του. Επίσης θα μπορούσε να δημιουργηθεί ένα social login, έτσι ώστε ο χρήστης να μπορεί να βλέπει τις προτιμήσεις άλλων φίλων του από τα κοινωνικά δίκτυα και να ανακτά πληροφορίες από φίλους του ή από άτομα που έχουν σχετικά κοινά ενδιαφέροντα και όχι από τυχαίους άλλους χρήστες.

Τέλος, μια σημαντική επέκταση θα ήταν και το επίπεδο ασφάλειας και κοινοποίησης των ταξιδιών του. Θα ήταν σημαντικό ο χρήστης να επέλεγε ποια ταξίδια θα κοινοποιεί δημόσια σε όλους, ποια σε φίλους και ποια θα τα κρατούσε μόνο για τον εαυτό του. Επίσης σημαντικό θα ήταν να μπορούσε να κοινοποιεί μέρος του ταξιδιού του, όπως π.χ. μια διαδρομή από όλες ή μερικές επισκέψεις από το σύνολο.

# 7

## Βιβλιογραφία

- [1] Android. <https://www.android.com/>
- [2] F. Alt, A. Sahami Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz. *Location-based Crowdsourcing: Extending Crowdsourcing to the Real World*. In Proceedings of the 6th Nordic Conference on Human-Computer Interaction (NordiCHI '10), pp. 13-22, Reykjavik, Iceland, October 2010.
- [3] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C. Chen, C. Yongxin, T. Chen, J. Zhang. *gMission: A General Spatial Crowdsourcing Platform*. In Proceedings of the 40th International Conference on Very Large Databases (PVLDB), 7(13):1629-1632, Hangzhou, China, 2014.
- [4] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, D. Zeinalipour-Yazti. *Crowdsourcing with Smartphones*. In Proceedings of the IEEE Internet Computing (IC '12), pp. 36-44, 2012.
- [5] Android Developer. <http://developer.android.com/tools/studio/index.html>
- [6] EPSG. <http://www.epsg.org/>
- [7] Everplaces. <http://everplaces.com>
- [8] Flickr. <https://www.flickr.com/>
- [9] Geospike. <http://geospike.com/>
- [10] GIST. J. M. Hellerstein, J. F. Naughton and A. Pfeffer. *Generalized Search Trees for Database Systems*. In Proceedings of the 21st Int'l Conf. on Very Large Data Bases, pp. 562–573, Zurich, Switzerland, September 1995.
- [11] Google Maps Android API.  
<https://developers.google.com/maps/documentation/android/>

- [12] M. F. Goodchild. *CITIZENS AS SENSORS: THE WORLD OF VOLUNTEERED GEOGRAPHY*, GeoJournal, Volume 69, Issue 4, pp 211-221, 2007.
- [13] L. Hollenstein and R. S. Purves. *Exploring place through user-generated content: Using Flickr tags to describe city cores*, Journal of spatial information science, Number 1, pp. 21–48, 2010.
- [14] Livemapp. <http://livemapp.com>
- [15] V. Lenders, E. Koukoumidis, P. Zhang and M. Martonosi. *Location-based Trust for Mobile User-generated Content: Applications, Challenges and Implementations*, In Proceedings of the 9th workshop on Mobile computing systems and applications (HotMobile '08), pp. 60-64, New York, USA, 2008
- [16] Miss Pronouncer. <http://www.misspronouncer.com/>
- [17] msdn.microsoft. <https://msdn.microsoft.com/en-us/library/ms950421.aspx>
- [18] Open Geospatial Consortium. <http://www.opengeospatial.org/>
- [19] OpenStreetMap. <https://www.openstreetmap.org>
- [20] Picasa. <https://picasa.google.com/>
- [21] PostGIS. <http://postgis.net/>
- [22] PostgreSQL. <http://www.postgresql.org/>
- [23] PROJ.4. <https://trac.osgeo.org/proj/>
- [24] Rtree. A. Guttman, *R-Trees: A Dynamic Index Structure for Spatial Searching*. In Proceedings of the 1984 ACM SIGMOD international conference on Management of data (SIGMOD '84), pp 47 – 57, New York, USA, June 1984.
- [25] Simple Object Access Protocol. <http://www.w3.org/TR/soap/>
- [26] Spatialite. <http://www.gaia-gis.it/gaia-sins/>
- [27] SQLite. <https://www.sqlite.org/>
- [28] PostgreSQL Full Text Search.  
<http://www.postgresql.org/docs/9.3/static/textsearch.html>
- [29] TileDownloader. <http://tiledownloader.sourceforge.net/>
- [30] Travelog. <http://www.travelog.me/>
- [31] TravelPod. <http://www.travelpod.com/>
- [32] TripAdvisor. <http://www.tripadvisor.com.gr/>



- [33] TripMayor. <http://tripmayor.com/>
- [34] Waze. <https://www.waze.com/>
- [35] Wikimapia. <http://wikimapia.org/>
- [36] Well Known Text - Well Known Binary.  
<http://www.gaia-gis.it/gaia-sins/spatialite-cookbook/html/wkt-wkb.html>

# 8

## *Παραρτήματα*

### ***8.1 Διαχείριση SQLite – Spatialite***

#### **Δημιουργία πίνακα «Trip»**

```
CREATE TABLE Trip (  
  Id INTEGER PRIMARY KEY AUTOINCREMENT,  
  Title TEXT,  
  ShortDesc TEXT,  
  DateCreated DATETIME,  
  DateClosed DATETIME)
```

#### **Δημιουργία πίνακα «Routes»**

```
CREATE TABLE Routes (  
  Id INTEGER PRIMARY KEY AUTOINCREMENT,  
  Title TEXT,  
  ShortDesc TEXT,  
  DateCreated DATETIME,  
  DateClosed DATETIME,  
  TripId INTEGER,  
  "Trajectory" LINESTRING,  
  FOREIGN KEY(TripId) REFERENCES Trip(Id) ON DELETE CASCADE)
```

### **Δημιουργία χωρικού ευρετηρίου στον πίνακα «Routes»**

```
CREATE VIRTUAL TABLE "idx_Routes_Trajectory" USING rtree(  
pkid, xmin, xmax, ymin, ymax)
```

### **Δημιουργία του πίνακα «Visits»**

```
CREATE TABLE "Visits" ("Id" INTEGER PRIMARY KEY AUTOINCREMENT  
, "DateCreated" DATETIME  
, "RoutesId" INTEGER  
, "name" text  
, "type" text  
, "location" POINT,  
FOREIGN KEY(RoutesId) REFERENCES Routes(Id) ON DELETE CASCADE)
```

### **Δημιουργία χωρικού ευρετηρίου στον πίνακα «Visits»**

```
CREATE VIRTUAL TABLE "idx_Visits_location" USING rtree(  
pkid, xmin, xmax, ymin, ymax)
```

### **Δημιουργία του πίνακα «Notes»**

```
CREATE TABLE "Notes" (  
"Id" INTEGER PRIMARY KEY AUTOINCREMENT,  
"Description" TEXT,  
"VisitsId" INTEGER,  
FOREIGN KEY(VisitsId) REFERENCES Visits(Id) ON DELETE CASCADE)
```

### **Δημιουργία του πίνακα «Images»**

```
CREATE TABLE "Images" (  

```

```
"Id" INTEGER PRIMARY KEY AUTOINCREMENT,  
"ImagePath" TEXT,  
VisitsId INTEGER,  
FOREIGN KEY(VisitsId) REFERENCES Visits(Id) ON DELETE CASCADE)
```

### **Δημιουργία του πίνακα «Pois»**

```
CREATE TABLE Pois (  
PK_UID INTEGER PRIMARY KEY AUTOINCREMENT,  
"osm_id" TEXT,  
"timestamp" TEXT,  
"name" TEXT,  
"type" TEXT, "Geometry" POINT)
```

## ***8.2 Διαχείριση PostgreSQL***

### **Δημιουργία του πίνακα «Trip»**

```
CREATE TABLE "Trip"  
(  
"Id" integer NOT NULL,  
"UserId" text NOT NULL,  
"Title" text,  
"ShortDesc" text,  
"DateCreated" timestamp without time zone NOT NULL,  
"DateClosed" timestamp without time zone,  
CONSTRAINT "Trip_Unique_ID" UNIQUE ("Id", "UserId")  
)
```

### **Δημιουργία του πίνακα «Routes»**

```

CREATE TABLE "Routes"
(
  "Id" integer NOT NULL,
  "UserId" text NOT NULL,
  "Title" text,
  "ShortDesc" text,
  "DateCreated" timestamp without time zone NOT NULL,
  "DateClosed" timestamp without time zone,
  "TripId" integer NOT NULL,
  "Trajectory" geometry(LineString,4326),
  CONSTRAINT "Routes_Unique_ID" UNIQUE ("Id", "UserId"),
  CONSTRAINT          routes_valid_check          CHECK
(st_isvalid("Trajectory"))
)

```

### **Δημιουργία του πίνακα «Visits»**

```

CREATE TABLE "Visits"
(
  "Id" integer NOT NULL,
  "UserId" text NOT NULL,
  "DateCreated" timestamp without time zone NOT NULL,
  "RoutesId" integer,
  name text,
  type text,
  location geometry(Point,4326),
  CONSTRAINT "Visits_Unique_ID" UNIQUE ("Id", "UserId"),
  CONSTRAINT visits_valid_check CHECK (st_isvalid(location))
)

```

### Δημιουργία του πίνακα «Notes»

```
CREATE TABLE "Notes"
(
  "Id" integer NOT NULL,
  "Description" text,
  "VisitsId" integer NOT NULL,
  "UserId" text,
  "SearchableText" tsvector,
  CONSTRAINT "Notes_Unique_ID" UNIQUE ("Id", "UserId")
)
```

### Δημιουργία του trigger «update notes row trigger»

```
CREATE OR REPLACE FUNCTION update_notes_row_trigger()
  RETURNS trigger AS
$BODY$
begin
  new."SearchableText" := to_tsvector('english', (select
coalesce("Visits"."name", '' ) || ' ' ||
coalesce("Visits"."type", '') || ' ' ||
coalesce(new."Description", '')
          from "Visits"
          inner join "Notes" on
"Visits"."Id"="Notes"."VisitsId"
          where "Visits"."Id"=new."VisitsId"));
  return new;
end
$BODY$
```

### Δημιουργία του πίνακα «Images»

```
CREATE TABLE "Images"
```

```
(
    "Id" integer NOT NULL,
    "ImagePath" text,
    "VisitsId" integer NOT NULL,
    "UserId" text,
    "ImageBytes" bytea,
    CONSTRAINT "Images_Unique_ID" UNIQUE ("Id", "UserId")
)
```

## 8.3 Web Services

### Υλοποίηση insertTrip

```
public bool insertTrip(int Id, string UserId, string Title,
string ShortDesc, DateTime DateCreated, DateTime DateClosed)
{
    bool Succeeded = false;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {
        NpgsqlCommand cmd = new NpgsqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "insert into public.\"Trip\" " +
            "\"(\"Id\", \"UserId\", \"+
            "\"Title\", \"+
            "\"ShortDesc\", \" +
            "\"DateCreated\", \"+
            "\"DateClosed\" )          values
(:Id, :UserId, :Title, :ShortDesc, :DateCreated, :DateClosed);";
        cmd.Parameters.AddWithValue("Id", Id);
        cmd.Parameters.AddWithValue("UserId", UserId);
```

```

        if (Title!=null)
            cmd.Parameters.AddWithValue("Title", Title);
        else
            cmd.Parameters.AddWithValue("Title",
System.DBNull.Value);
        if (ShortDesc!=null)
            cmd.Parameters.AddWithValue("ShortDesc",
ShortDesc);
        else
            cmd.Parameters.AddWithValue("ShortDesc",
System.DBNull.Value);
        cmd.Parameters.AddWithValue("DateCreated",
DateCreated);
        if      (DateClosed!=DateTime.Parse("1900-01-01
00:00:00"))
            cmd.Parameters.AddWithValue("DateClosed",
DateClosed);
        else
            cmd.Parameters.AddWithValue("DateClosed",
System.DBNull.Value);
        cmd.Connection = conn;
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
        Succeeded = true;
    }
    catch (Exception ex)
    {
        string mes = ex.Message;
        Succeeded = false;
    }
    return Succeeded;

```



```
}
```

### **Υλοποίηση deleteTrip**

```
public int deleteTrip(int Id, string UserId)
{
    int rowsAffected = 0;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {
        NpgsqlCommand cmd = new NpgsqlCommand("DELETE FROM
public.\"Trip\" where \"Id\"=:Id and \"UserId\"=:UserId ;",
conn);

        cmd.Parameters.AddWithValue("Id", Id);
        cmd.Parameters.AddWithValue("UserId", UserId);
        cmd.Connection = conn;
        conn.Open();
        rowsAffected = cmd.ExecuteNonQuery();
        conn.Close();
    }
    catch (Exception ex)
    {
        string mes = ex.Message;
    }
    return rowsAffected;
}
```

### **Υλοποίηση deleteTripVisits**

```
public int deleteTripVisits(int Id, string UserId)
{
    int rowsAffected = 0;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
```

```

        try
        {
            NpgsqlCommand cmd = new NpgsqlCommand("DELETE "+
                                                    "          FROM
\"Visits\" "+
                                                    "          where
\"RoutesId\" in (select \"Id\" from \"Routes\" where
\"TripId\"=:Id) and \"UserId\"=:UserId ;", conn);

            cmd.Parameters.AddWithValue("Id", Id);
            cmd.Parameters.AddWithValue("UserId", UserId);
            cmd.Connection = conn;
            conn.Open();
            rowsAffected = cmd.ExecuteNonQuery();
            conn.Close();
        }
        catch (Exception ex)
        {
            string mes = ex.Message;
        }
        return rowsAffected;
    }
}

```

### **Υλοποίηση deleteTripNotes**

```

public int deleteTripNotes(int Id, string UserId)
{
    int rowsAffected = 0;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {
        NpgsqlCommand cmd = new NpgsqlCommand("delete from
\"Notes\" where \"VisitsId\" in (select \"Id\" from \"Visits\"
where \"RoutesId\" in ( select \"Id\" from \"Routes\" where
\"TripId\" =:Id)) and \"UserId\" =:UserId;", conn);
    }
}

```

```

        cmd.Parameters.AddWithValue("Id", Id);
        cmd.Parameters.AddWithValue("UserId", UserId);
        cmd.Connection = conn;
        conn.Open();
        rowsAffected = cmd.ExecuteNonQuery();
        conn.Close();
    }
    catch (Exception ex)
    {
        string mes = ex.Message;
    }
    return rowsAffected;
}

```

### **Υλοποίηση deleteTripImages**

```

public int deleteTripImages(int Id, string UserId)
{
    int rowsAffected = 0;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {
        NpgsqlCommand cmd = new NpgsqlCommand("delete from
\"Images\" where \"VisitsId\" in (select \"Id\" from
\"Visits\" where \"RoutesId\" in ( select \"Id\" from
\"Routes\" where \"TripId\" =:Id)) and \"UserId\" =:UserId;",
conn);

        cmd.Parameters.AddWithValue("Id", Id);
        cmd.Parameters.AddWithValue("UserId", UserId);
        cmd.Connection = conn;
        conn.Open();
        rowsAffected = cmd.ExecuteNonQuery();
        conn.Close();
    }
}

```

```

    }
    catch (Exception ex)
    {
        string mes = ex.Message;
    }
    return rowsAffected;
}

```

### **Υλοποίηση insertRoute**

```

public bool insertRoute(int Id, string UserId, string Title,
string ShortDesc, DateTime DateCreated, DateTime DateClosed,
int TripId, string Trajectory)
{
    bool Succeeded = false;

    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {
        NpgsqlCommand cmd = new NpgsqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "INSERT INTO \"Routes\" (" +
            "\"Id\", \"UserId\", \"Title\", \"ShortDesc\",
\"DateCreated\", \"DateClosed\", "+
            "\"TripId\", \"Trajectory\") "+
            "VALUES
(:Id, :UserId, :Title, :ShortDesc, :DateCreated, :DateClosed,
:TripId, ST_GeomFromText('\" + Trajectory + \"', 4326));";
        cmd.Parameters.AddWithValue("Id", Id);
        cmd.Parameters.AddWithValue("UserId", UserId);
        if (Title != null)
            cmd.Parameters.AddWithValue("Title", Title);
        else

```

```

        cmd.Parameters.AddWithValue("Title",
System.DBNull.Value);
        if (ShortDesc != null)
            cmd.Parameters.AddWithValue("ShortDesc",
ShortDesc);
        else
            cmd.Parameters.AddWithValue("ShortDesc",
System.DBNull.Value);
        cmd.Parameters.AddWithValue("DateCreated",
DateCreated);
        if (DateClosed != DateTime.Parse("1900-01-01
00:00:00"))
            cmd.Parameters.AddWithValue("DateClosed",
DateClosed);
        else
            cmd.Parameters.AddWithValue("DateClosed",
System.DBNull.Value);
        cmd.Parameters.AddWithValue("TripId", TripId);
        //if (Trajectory != null&&Trajectory!="")
        //    cmd.Parameters.AddWithValue("Trajectory",
Trajectory);
        //else
        //    cmd.Parameters.AddWithValue("Trajectory",
System.DBNull.Value);
        cmd.Connection = conn;
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
        Succeeded = true;
    }
    catch (Exception ex)
    {
        string mes = ex.Message;

```

```

        Succeeded = false;
    }
    return Succeeded;
}

```

### **Υλοποίηση deleteTripRoutes**

```

public int deleteTripRoutes(int TripId, string UserId)
{
    int rowsAffected = 0;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {
        NpgsqlCommand cmd = new NpgsqlCommand("DELETE FROM
\"Routes\" where \"TripId\"=:TripId and \"UserId\"=:UserId ;",
conn);
        cmd.Parameters.AddWithValue("TripId", TripId);
        cmd.Parameters.AddWithValue("UserId", UserId);
        cmd.Connection = conn;
        conn.Open();
        rowsAffected = cmd.ExecuteNonQuery();
        conn.Close();
    }
    catch (Exception ex)
    {
        string mes = ex.Message;
    }
    return rowsAffected;
}

```

```
}
```

### **Υλοποίηση insertVisit**

```
public bool insertVisit(int Id, string UserId, DateTime
DateCreated, int RoutesId, string name, string type, string
location)
{
    bool Succeeded = false;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {
        NpgsqlCommand cmd = new NpgsqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "INSERT INTO \"Visits\"( "+
            "\"Id\",          \"UserId\",          \"DateCreated\",
\"RoutesId\", name, type, location) "+
            " VALUES (:Id, :UserId, :DateCreated, :RoutesId,
:name, :type, ST_GeomFromText('\" + location + '\",4326));";
        cmd.Parameters.AddWithValue("Id", Id);
        cmd.Parameters.AddWithValue("UserId", UserId);
        cmd.Parameters.AddWithValue("DateCreated",
DateCreated);
        cmd.Parameters.AddWithValue("RoutesId", RoutesId);
        if (name != null)
            cmd.Parameters.AddWithValue("name", name);
        else
            cmd.Parameters.AddWithValue("name",
System.DBNull.Value);
        if (type != null)
            cmd.Parameters.AddWithValue("type", type);
        else
```

```

        cmd.Parameters.AddWithValue("type",
System.DBNull.Value);
        //if (location != null && location != "")
        //      cmd.Parameters.AddWithValue("location",
location);
        //else
        //      cmd.Parameters.AddWithValue("location",
System.DBNull.Value);
        cmd.Connection = conn;
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
        Succeeded = true;
    }
    catch (Exception ex)
    {
        string mes = ex.Message;
        Succeeded = false;
    }
    return Succeeded;
}

```

### **Υλοποίηση insertNote**

```

public bool insertNote(int Id, string UserId, int VisitsId,
string Description)
{
    bool Succeeded = false;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {

```



```

        NpgsqlCommand cmd = new NpgsqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "INSERT INTO \"Notes\"( "+
            "\"Id\",      \"UserId\",      \"Description\",
\"VisitsId\") "+
            "VALUES      (:Id,      :UserId,      :Description,
:VisitsId);";

        cmd.Parameters.AddWithValue("Id", Id);
        cmd.Parameters.AddWithValue("UserId", UserId);

        cmd.Parameters.AddWithValue("VisitsId", VisitsId);
        if (Description != null)
            cmd.Parameters.AddWithValue("Description",
Description);
        else
            cmd.Parameters.AddWithValue("Description",
System.DBNull.Value);

        cmd.Connection = conn;
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
        Succeeded = true;
    }
    catch (Exception ex)
    {
        string mes = ex.Message;
        Succeeded = false;
    }
    return Succeeded;
}

```

## Υλοποίηση insertImage

```
public bool insertImage(int Id, string UserId, int
VisitsId, string ImagePath)
{
    StreamWriter w = new StreamWriter(@"c:\temp\log.txt");
    bool Succeeded = false;
    NpgsqlConnection conn = new
NpgsqlConnection(connstring);
    try
    {
        NpgsqlCommand cmd = new NpgsqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "INSERT INTO \"Images\"( " +
" \"Id\", \"UserId\", \"ImagePath\", \"VisitsId\")
" +
" VALUES (:Id, :UserId, :ImagePath, :VisitsId);";
        cmd.Parameters.AddWithValue("Id", Id);
        cmd.Parameters.AddWithValue("UserId", UserId);
        cmd.Parameters.AddWithValue("VisitsId", VisitsId);
        if (ImagePath != null)
            cmd.Parameters.AddWithValue("ImagePath",
ImagePath);
        else
            cmd.Parameters.AddWithValue("ImagePath",
System.DBNull.Value);
        // Convert the Base64 UUEncoded input into binary
output.
        //byte[] binaryData = null;
        //try
        //{
        //    w.WriteLine("βημα 1");
    }
```

```

        //          binaryData          =
System.Convert.FromBase64String(ImageBytes);
        //      SaveData(binaryData);
        //          //cmd.Parameters.AddWithValue("ImageBytes",
binaryData);
        //          //cmd.Parameters.AddWithValue("ImageBytes",
System.DBNull.Value);
        //}
        //catch (System.ArgumentNullException)
        //{
        //      w.WriteLine("βημσ 2");
        //      System.Console.WriteLine("Base 64 string is
null.");
        //          //cmd.Parameters.AddWithValue("ImageBytes",
System.DBNull.Value);
        //      //return;
        //}
        //catch (System.FormatException)
        //{
        //      w.WriteLine("βημσ 3");
        //          System.Console.WriteLine("Base 64 string
length is not " +
        //          "4 or is not an even multiple of 4.");
        //          //cmd.Parameters.AddWithValue("ImageBytes",
System.DBNull.Value);
        //      //return;
        //}

cmd.Connection = conn;
conn.Open();
cmd.ExecuteNonQuery();
w.WriteLine("βημσ 4");

```

```

        conn.Close();
        Succeeded = true;
    }
    catch (Exception ex)
    {
        w.WriteLine("βηµα 5");
        string mes = ex.Message;
        Succeeded = false;
    }
    w.Flush();
    w.Close();
    return Succeeded;
}

protected bool SaveData(byte[] Data)
{
    BinaryWriter Writer = null;
    string Name = @"C:\temp\MyImage.jpg";

    try
    {
        // Create a new stream to write to the file
        Writer = new BinaryWriter(File.OpenWrite(Name));

        // Writer raw data
        Writer.Write(Data);
        Writer.Flush();
        Writer.Close();
    }
    catch
    {
        //...
    }
}

```

```

        return false;
    }

    return true;
}

```

### **Υλοποίηση getRequestedVisitsWithRank**

```

public      DataTable      getRequestedVisitsWithRank(string
whereClause)
{
    string[] sKeywords = whereClause.Split(' ');
    int size = sKeywords.Length;
    whereClause = "";
    for (int i = 0; i < size; i++)
    {
        whereClause += " | " + sKeywords[i];
    }
    if (whereClause != "")
    {
        whereClause = whereClause.Substring(3);
    }

    NpgsqlConnection conn = new
NpgsqlConnection(connstring);

    DataSet ds = new DataSet("Visits");
    DataTable dt = new DataTable("Visits");
    try
    {
        string sqlQuery = string.Format(
            "select                                \"Visits\".\"Id\",
\"Visits\".\"name\",
\"Visits\".\"type\",ts_rank(\"Notes\".\"SearchableText\",
to_tsquery('english','{0}'))                AS                rank                ,

```

```

ST_X("\"Visits\".\"location\")          as          longt,
ST_Y("\"Visits\".\"location\") as lat" +
        " from \"Visits\" " +
        " inner join \"Notes\" on \"Visits\".\"Id\" =
\"Notes\".\"VisitsId\" " +
        " where \"Notes\".\"SearchableText\" @@
to_tsquery('english','{0}')" +
        " order by rank desc " +
        " limit 10;", whereClause);

NpgsqlDataAdapter          adapter          =          new
NpgsqlDataAdapter(sqlQuery, conn);

adapter.Fill(ds);

dt = ds.Tables[0];
}

catch (Exception ex)
{
    string mes = ex.Message;
}

return dt;
}

```

### **Αποτελέσματα σε xml μορφή στην αναζήτηση με λέξεις κλειδιά**

```

<?xml version="1.0" encoding="utf-8"?>
<DataTable xmlns="http://tempuri.org/">
  <xs:schema          id="Visits"          xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element          name="Visits"          msdata:IsDataSet="true"
msdata:MainDataTable="Table" msdata:UseCurrentLocale="true">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="Table">

```

```

        <xs:complexType>
            <xs:sequence>
                <xs:element          name="Id"          type="xs:int"
minOccurs="0" />
                <xs:element          name="name"        type="xs:string"
minOccurs="0" />
                <xs:element          name="type"        type="xs:string"
minOccurs="0" />
                <xs:element          name="rank"        type="xs:float"
minOccurs="0" />
                <xs:element          name="longt"       type="xs:double"
minOccurs="0" />
                <xs:element          name="lat"         type="xs:double"
minOccurs="0" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram          xmlns:msdata="urn:schemas-microsoft-
com:xml-msdata"          xmlns:diffgr="urn:schemas-microsoft-com:xml-
diffgram-v1">
    <Visits xmlns="">
        <Table diffgr:id="Table1" msdata:rowOrder="0">
            <Id>13</Id>
            <name>the acropolis museum</name>
            <type>museum</type>
            <rank>0.08732594</rank>
            <longt>23.728518</longt>
            <lat>37.96828</lat>
        </Table>

```

```
<Table diffgr:id="Table2" msdata:rowOrder="1">
  <Id>8</Id>
  <name>erechtheion</name>
  <type>archaeological_s</type>
  <rank>0.0759908855</rank>
  <longt>23.726978</longt>
  <lat>37.971451</lat>
</Table>
<Table diffgr:id="Table3" msdata:rowOrder="2">
  <Id>15</Id>
  <name>byzantine museum</name>
  <type>museum</type>
  <rank>0.0748847947</rank>
  <longt>23.744758</longt>
  <lat>37.974744</lat>
</Table>
<Table diffgr:id="Table4" msdata:rowOrder="3">
  <Id>17</Id>
  <name>the acropolis museum</name>
  <type>museum</type>
  <rank>0.07176917</rank>
  <longt>23.728576</longt>
  <lat>37.968324</lat>
</Table>
<Table diffgr:id="Table5" msdata:rowOrder="4">
  <Id>9</Id>
  <name>byzantine museum</name>
  <type>museum</type>
  <rank>0.0444884375</rank>
  <longt>23.745156</longt>
  <lat>37.975385</lat>
</Table>
```



```
<Table diffgr:id="Table6" msdata:rowOrder="5">
  <Id>33</Id>
  <name>British museum</name>
  <type>museum</type>
  <rank>0.04327259</rank>
  <longt>-0.12703</longt>
  <lat>51.519406</lat>
</Table>
<Table diffgr:id="Table7" msdata:rowOrder="6">
  <Id>36</Id>
  <name>British museum</name>
  <type>museum</type>
  <rank>0.04327259</rank>
  <longt>-0.126918</longt>
  <lat>51.519271</lat>
</Table>
<Table diffgr:id="Table8" msdata:rowOrder="7">
  <Id>10</Id>
  <name>national archaeological museum</name>
  <type>archaeological_s</type>
  <rank>0.041372817</rank>
  <longt>23.732522</longt>
  <lat>37.988879</lat>
</Table>
<Table diffgr:id="Table9" msdata:rowOrder="8">
  <Id>41</Id>
  <name>British museum</name>
  <type>museum</type>
  <rank>0.041372817</rank>
  <longt>-0.126973</longt>
  <lat>51.519321</lat>
</Table>
```

```
<Table diffgr:id="Table10" msdata:rowOrder="9">
  <Id>23</Id>
  <name>musee du louvre</name>
  <type>museum</type>
  <rank>0.041372817</rank>
  <longt>2.337541</longt>
  <lat>48.860533</lat>
</Table>
</Visits>
</diffgr:diffgram>
</DataTable>
```