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

Ανάπτυξη διαδικτυακής εφαρμογής διαχείρισης πολυμεσικού περιεχομένου σε περιβάλλον Android

Διπλωματική Εργασία

Φραντζεσκάκης Στυλιανός

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π

Αθήνα, Ιούλιος 2011

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

Ανάπτυξη διαδικτυακής εφαρμογής διαχείρισης πολυμεσικού περιεχομένου σε περιβάλλον Android

Διπλωματική Εργασία

Φραντζεσκάκης Στυλιανός

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Ιουλίου 2011

.....	.....	.....
Στέφανος Κόλλιας	Ανδρέας Σταφυλοπάτης	Γιώργος Στάμου
Καθηγητής Ε.Μ.Π.	*Καθηγητής Ε.Μ.Π.*	*Καθηγητής Ε.Μ.Π.*

Αθήνα, Ιούλιος 2011

……

Στυλιανός Γ. Φραντζεσκάκης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η παρούσα διπλωματική εργασία έχει ως κύριο στόχο την ανάπτυξη μίας διαδικτυακής ηλεκτρονικής εφαρμογής σε περιβάλλον Android. Η εφαρμογή επιτρέπει στον κάτοχο κινητού τηλεφώνου με το λειτουργικό σύστημα Android να τραβήξει ψηφιακές φωτογραφίες με τη συσκευή του και εν συνεχεία να:

1. ανακτήσει παρόμοιες, με βάση το οπτικό περιεχόμενο, φωτογραφίες από το Διαδίκτυο,
2. να εντοπίσει το οπτικό περιεχόμενο της φωτογραφίας στον παγκόσμιο χάρτη (geo-location),
3. να προσθέσει χαρακτηρισμούς (tags) στη φωτογραφία του με βάση 2 κατηγορίες προτεινόμενων tags από άλλους χρήστες και από το σύστημα,
4. να αποθηκεύσει την φωτογραφία με τα καινούρια μεταδεδομένα της (tags & geo-tags) στην κάρτα μνήμης της συσκευής του,
5. να διατηρεί και να πλοηγείται στο ιστορικό χρήσης της εφαρμογής, το οποίο περιλαμβάνει παλαιότερες φωτογραφίες και αναζητήσεις του.

Αν θελήσουμε να συνοψίσουμε τα βασικά στοιχεία αυτής της διπλωματικής εργασίας, αυτά επικεντρώνονται σε δύο βασικούς άξονες. Ο πρώτος περιστρέφεται γύρω από τις λειτουργικές παραμέτρους και την αισθητική παρουσίαση της εφαρμογής, και οριοθετεί την ευχρηστία, την ευκολία πρόσβασης και την "ελαφρότητά" της, μέσω του interface ενός "έξυπνου κινητού τηλεφώνου" (smartphone). Ο δεύτερος αφορά την αξιοπιστία και τη διασύνδεση με το υπάρχον web-based σύστημα οπτικής ανάκτησης και γεωπροσδιορισμού φωτογραφιών (http://viral.image.ntua.gr/), με το οποίο καλείται να συνεργαστεί, και περιστρέφεται γύρω από τη χρήση κοινής βάσης δεδομένων για τα δεδομένα, την οργάνωση των προφίλ των χρηστών και των δεδομένων του ιστορικού χρήσης τους.

Συμπερασματικά, λοιπόν, πρόκειται για μία πλήρη διαδικτυακή εφαρμογή σε περιβάλλον Android, η οποία αναμένεται να τύχει ευρείας αποδοχής από τους χρήστες έξυπνων κινητών τηλεφώνων και η οποία είναι ιδιαίτερα εύπλαστη και σε θέση να ακολουθεί κάθε φορά τις εκάστοτε παρουσιαζόμενες ανάγκες διαχείρισης πολυμεσικού περιεχομένου από τους ίδιους τους χρήστες (user-generated content).

Λέξεις Κλειδιά: ψηφιακό πολυμεσικό περιεχόμενο, Android, χαρακτηρισμοί, γεω-πληροφορίες

Abstract

The main goal of this Diploma Thesis is the development of a web-based application within the Android environment. The developed application allows the owner of a mobile smart-phone, equipped with the Android operating system, to capture digital photographs using his device and then:

1. retrieve visually similar photographs from the Internet,
2. locate the visual content of the original photograph on the world map (i.e. its geo-location),
3. add tags to his photograph based on two distinct categories of suggested tags (i.e. by other users and by the system),
4. save the photograph together with its fresh metadata (tags & geo-tags) to the memory card of his device,
5. maintain and browse the application's usage history, which includes both photographs and searches.

In order to summarize the key elements of this Diploma Thesis, they focus on two main aspects. The first one has to do with the operational parameters and the aesthetic presentation of the application, and defines its usability, ease of access and lightness through the interface of a smart-phone. The second one concerns the reliability and linking up with the existing web-based visual image retrieval and localization system (http://viral.image.ntua.gr), with which the application strongly cooperates, and revolves around the use of common database for the data, the organization of the user profiles and the data of their usage history.

As a result, the herein desribed application forms a very interesting full-scale, standalone application developed within the Android environment and is expected to gain wide acceptance by smart-phones users in the near future in the field of user-generated content management.

Keywords: multimedia content, Android, tags, geo-tags

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1
Εισαγωγή
1.1 Αντικείμενο της διπλωματικής..... 12
1.2 Οργάνωση κειμένου..... 13

ΚΕΦΑΛΑΙΟ 2
Έννοιες προσωποποίησης διαδικτυακού πολυμεσικού περιεχομένου
2.1 Γενικά..... 15
2.2 Προφίλ χρήστη..... 16
2.3 Ιστορικό χρήσης..... 17
2.4 Συμπεράσματα..... 18

ΚΕΦΑΛΑΙΟ 3
Προγραμματισμός σε περιβάλλον Android
3.1 Εισαγωγή..... 19
3.2 Προγραμματιστικά εργαλεία..... 19
3.3 Συστατικά στοιχεία εφαρμογής..... 22
3.3.1 Activities..... 22
3.3.2 Services..... 23
3.3.3 Content Providers..... 24
3.3.4 Broadcast Receivers..... 24
3.4 User Interface..... 24
3.4.1 Layout..... 25
3.4.2 Μενού..... 27
3.4.2.1 Options Menu..... 27
3.4.2.2 Context Menu..... 28
3.4.2.3 Submenu..... 28
3.4.3 Dialogs..... 28
3.4.4 Ειδοποιώντας το χρήστη..... 30
3.4.4.1 Toast Notification..... 30
3.4.4.2 Status Bar Notification..... 31
3.5 Application Resources & Συμβατότητα..... 31
3.5.1 Προσανατολισμός..... 32
3.5.2 Μέγεθος Οθόνης..... 33
3.5.3 Γλώσσα..... 34
3.6 Android Manifest..... 36
3.7 Βέλτιστος Σχεδιασμός..... 37
3.7.1 Δυνατότητα Πρόσβασης..... 38
3.7.2 Επίδοση..... 39
3.7.3 Ανταπόκριση..... 40

ΚΕΦΑΛΑΙΟ 4
Αρχιτεκτονική της εφαρμογής
4.1 Εισαγωγή..... 43
4.2 Περιγραφή..... 43
4.3 Δομή..... 44

ΚΕΦΑΛΑΙΟ 5
Βασικές λειτουργίες και σενάριο χρήσης εφαρμογής

5.1 Εισαγωγή..... 49
5.2 Αρχική Οθόνη..... 49
5.3 Οθόνη Εισαγωγής Εικόνας..... 53
5.4 Οθόνη Αποστολής Εικόνας..... 57
5.5 Οθόνη Αποτελεσμάτων..... 62
5.5.1 Πληροφορίες Περιεχομένου..... 62
5.5.2 Χάρτης Τοποθεσίας..... 66
5.5.3 Παρόμοιες Εικόνες..... 68
5.5.4 Παραπλήσιες Εικόνες..... 69
5.6 Οθόνη Λογαριασμού..... 70
5.7 Οθόνη Ιστορικού..... 73
5.8 Οθόνη "Σχετικά"..... 76
5.9 Γλώσσα της Εφαρμογής..... 77

ΚΕΦΑΛΑΙΟ 6
Συμπεράσματα και μελλοντικές εξελίξεις

6.1 Σύνοψη και συμπεράσματα..... 79
6.2 Μελλοντικές εξελίξεις..... 80

ΠΑΡΑΡΤΗΜΑ..... 82

ΒΙΒΛΙΟΓΡΑΦΙΑ..... 160

1

Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Τα τελευταία χρόνια βιώνουμε μία επανάσταση των έξυπνων κινητών τηλεφώνων (smartphones), τα οποία ενσωματώνουν πλέον δυνατότητες, οι οποίες δεν υπάρχουν στα συμβατικά κινητά τηλέφωνα. Περιλαμβάνουν ολοκληρωμένο λειτουργικό σύστημα, ισχυρό επεξεργαστή ικανό για εκτέλεση πολύπλοκων και χρονοβόρων υπολογισμών, δυνατότητα σύνδεσης στο διαδίκτυο είτε μέσω ασύρματων δικτύων (Wi-Fi) είτε δικτύων τρίτης γενιάς (3G networks), κάμερα υψηλής ανάλυσης, συστήματα εντοπισμού θέσης (GPS) κ.α. Τα λειτουργικά συστήματα των συσκευών αυτών ποικίλουν, με το Android της Google να περιλαμβάνεται τη στιγμή αυτή στο μεγαλύτερο ποσοστό των smartphones της αγοράς. Μάλιστα επίσημα στοιχεία δείχνουν ότι η ιστοσελίδα της Google, Android Market, που περιέχει όλες τις εφαρμογές για Android περιλαμβάνει αυτή τη στιγμή περισσότερες από 250.000 εφαρμογές.

Στο πλαίσιο αυτό, σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μίας εφαρμογής που προορίζεται για τα παραπάνω κινητά τηλέφωνα. Φυσικά, η εφαρμογή μπορεί να εκτελεστεί όχι μόνο σε κινητά τηλέφωνα αλλά και σε οποιαδήποτε άλλη ηλεκτρονική συσκευή που περιλαμβάνει λειτουργικό σύστημα Android (όπως tablet pc). Μέσα από την εφαρμογή αυτή, ο χρήστης θα μπορεί να αντλεί πληροφορίες για το περιεχόμενο μίας φωτογραφίας που έχει τραβήξει με την κάμερα της συσκευής του, (ή κάποιας άλλης εικόνας), όπως τοποθεσία, ονομασία τοπωνυμίου, παρόμοιες εικόνες κ.α. Στα πλαίσια της προσωποποίησης και εξατομίκευσης της εφαρμογής, ο κάθε χρήστης θα μπορεί να αποθηκεύει το ιστορικό του στον προσωπικό του λογαριασμό, το οποίο θα είναι προσβάσιμο από οποιαδήποτε συσκευή, αφού θα αποθηκεύεται online.

1.2 Οργάνωση κειμένου

Η διπλωματική αυτή εργασία αναλύεται συνολικά σε 6 κεφάλαια. Συγκεκριμένα έχουμε:

Στο **1ο Κεφάλαιο** ασχολούμαστε με την εισαγωγή στο αντικείμενο που πραγματεύεται η διπλωματική εργασία. Εισάγουμε τον αναγνώστη στις νέες τεχνολογίες που χρησιμοποιούνται στις σημερινές κινητές συσκευές (mobile devices) και τον πληροφορούμε σχετικά με το σκοπό συγγραφής της διπλωματικής εργασίας και τον τρόπο που συνδέεται αυτή με τις παραπάνω τεχνολογίες.

Στο **2ο Κεφάλαιο** αναφέρονται οι ανάγκες που προκύπτουν, έπειτα από τον τεράστιο όγκο πληροφοριών που δέχεται καθημερινά ο χρήστης. Οι ανάγκες αυτές σηματοδοτούν τη μετάβαση σε μία νέα φιλοσοφία που υιοθετείται στην κατασκευή υπηρεσιών, ιστοσελίδων και εφαρμογών, και έχει ως βάση της τις μοναδικές ανάγκες του κάθε χρήστη. Εξηγείται, λοιπόν, η έννοια της προσωποποίησης ή εξατομίκευσης του ψηφιακού πολυμεσικού περιεχομένου, και προτείνονται τρόποι προσαρμογής αυτού, ώστε να ανταποκρίνεται στις προτιμήσεις και ανάγκες του κάθε χρήστη.

Το **3ο Κεφάλαιο** περιγράφει το λειτουργικό σύστημα Android και εισάγει τον αναγνώστη στον προγραμματισμό στο περιβάλλον αυτό. Εξηγούνται τα προγραμματιστικά εργαλεία που χρειάζονται και αναλύονται τα βασικά βήματα που απαιτούνται, για την ανάπτυξη μίας εφαρμογής που θα εκτελείται σε λειτουργικό σύστημα Android. Επιπλέον περιγράφονται ορισμένες προγραμματιστικές τεχνικές καθώς και αναλύονται τρόποι βελτιστοποίησης μίας εφαρμογής, ώστε να είναι προσβάσιμη σε ευρύ κοινό, να είναι γρήγορη, λειτουργική και αποδοτική. Το κεφάλαιο αυτό, λοιπόν, αποτελεί ουσιαστικά έναν εισαγωγικό οδηγό που πρέπει να ακολουθήσει ο προγραμματιστής ώστε να κατασκευάσει μία εφαρμογή Android.

Στο **4ο Κεφάλαιο** παρουσιάζεται αναλυτικά η αρχιτεκτονική της εφαρμογής που υλοποιήθηκε στο πλαίσιο της διπλωματικής αυτής εργασίας. Αρχικά αναφέρονται οι διάφορες λειτουργίες και δυνατότητες της εφαρμογής και έπειτα περιγράφεται η δομή της, το πώς δηλαδή οργανώνονται όλα τα αρχεία που περιέχονται σε αυτήν και ποια είναι η κύρια λειτουργία τους.

Στο **5ο Κεφάλαιο** παρουσιάζονται διεξοδικά όλες οι λειτουργίες της εφαρμογής και οι επιλογές που έχει ο χρήστης αλληλεπιδρώντας με αυτήν. Περιγράφεται κάθε διαφορετική οθόνη της εφαρμογής, οι επιλογές που έχει ο χρήστης στη συγκεκριμένη οθόνη, καθώς και ο τρόπος υλοποίησης των παραπάνω λειτουργιών στο προγραμματιστικό περιβάλλον. Όλα τα αποτελέσματα παρουσιάζονται σε εικόνες, οι οποίες ουσιαστικά εμφανίζουν ένα στιγμιότυπο από την αντίστοιχη οθόνη της συσκευής που εκτελεί την εφαρμογή, και βοηθάνε τον αναγνώστη στην κατανόηση των παραπάνω λειτουργιών.

Στο **6ο Κεφάλαιο** περιγράφονται τα συμπεράσματα που προκύπτουν από την εκπόνηση της διπλωματικής εργασίας, καθώς και οι μελλοντικές εξελίξεις της εφαρμογής. Αναφέρονται, ακόμα, συγκεκριμένες επεκτάσεις που σχεδιάζουμε να συμπεριλάβουμε στο προσεχές διάστημα στην εφαρμογή αυτή.

Στο **Παράρτημα** της εργασίας αυτής, παρατίθεται αναλυτικά ο κώδικας της εφαρμογής που αναπτύχθηκε, ταξινομημένος στις κλάσεις και έπειτα στα υπόλοιπα αρχεία που περιλαμβάνει το project της εφαρμογής. Φυσικά, ορισμένες ευαίσθητες πληροφορίες όπως διευθύνσεις IP, ονόματα χρήστη και κωδικοί που χρησιμοποιούνται για την επικοινωνία μεταξύ εφαρμογής και εξυπηρετητή, παραλείπονται για λόγους ασφαλείας.

Τέλος, παρουσιάζεται η **Βιβλιογραφία** που χρησιμοποιήθηκε για τη συγγραφή του παρόντος κειμένου, καθώς και όλες οι ιστοσελίδες που βοήθησαν στην ανάπτυξη της εφαρμογής.

2

Έννοιες προσωποποίησης διαδικτυακού πολυμεσικού περιεχομένου

2.1 Γενικά

Στη σημερινή εποχή, ο όγκος των ψηφιακών πληροφοριών τείνει ολοένα και αυξανόμενος. Το γεγονός αυτό προκαλεί στο μέσο χρήστη να έρχεται καθημερινά αντιμέτωπος με ένα "ψηφιακό χάος", είτε αυτό έχει σχέση με την εργασία του είτε την ψυχαγωγία του, το οποίο συνίσταται κυρίως σε διαδικτυακό πολυμεσικό περιεχόμενο όπως σελίδες υπερκειμένου, ψηφιακές φωτογραφίες, blogs, forum ή ακόμα και πολύπλοκα αλληλεπιδραστικά περιβάλλοντα. Ο χρήστης μπορεί να αλληλεπιδράσει με το ψηφιακό αυτό υλικό, μέσω ολοένα και περισσότερων ηλεκτρονικών συσκευών, όπως ηλεκτρονικούς υπολογιστές, smartphones (έξυπνα κινητά τηλέφωνα) ή smart televisions (έξυπνες τηλεοράσεις), οι οποίες αποτελούν πλέον μικρούς υπολογιστές και έχουν τη δυνατότητα να αναπαράγουν, να αποθηκεύουν και να επεξεργάζονται το πολυμεσικό αυτό περιεχόμενο. Καθίσταται, λοιπόν, σαφής η ανάγκη προσαρμογής των παραπάνω πληροφοριών ώστε να ταιριάζουν στις ανάγκες του κάθε χρήστη.

Η βοήθεια του τελικού χρήστη στο πλαίσιο των παραπάνω αναγκών, έρχεται μέσα από την προσωποποίηση του πολυμεσικού περιεχομένου. Με τον όρο personalization (προσωποποίηση ή εξατομίκευση) εννοούμε, δηλαδή, τους τρόπους με τους οποίους πληροφορίες και υπηρεσίες προσαρμόζονται, με σκοπό να ταιριάζουν στις μοναδικές και συγκεκριμένες ανάγκες ενός ατόμου ή μίας κοινότητας. Ρόλος της προσωποποίησης είναι να κάνει τη ζωή του χρήστη πιο εύκολη και πιο απλή, φροντίζοντας για την κάλυψη των πραγματικών αναγκών του,

προτιμήσεων και ενδιαφερόντων. Για παράδειγμα, όταν ο χρήστης αναζητά κάποια εικόνα ή βίντεο σε μία υπηρεσία, θα θέλαμε η αναζήτηση αυτή να λαμβάνει υπόψη της τις προτιμήσεις του συγκεκριμένου χρήστη, ώστε να αποφευχθούν περιπτώσεις επιστροφής μακροσκελών και άσχετων πληροφοριών. Επίσης, η επιλογή ειδήσεων για ανάγνωση στο διαδίκτυο, αποτελεί υποκειμενικό κριτήριο του εκάστου χρήστη, οπότε θα ήταν λανθασμένη προσέγγιση οι ειδήσεις να παρουσιάζονται με τον ίδιο ακριβώς τρόπο και την ίδια σειρά, σε όλους τους δυνατούς διαφορετικούς χρήστες. Ο απώτερος, λοιπόν, στόχος της προσωποποίησης είναι να καταστεί δυνατή η προσφορά πολυμεσικού περιεχομένου, στον τελικό χρήστη, το οποίο βρίσκεται πολύ κοντά στις επιθυμίες του.

2.2 Προφίλ χρήστη

Ορισμένες πληροφορίες σχετικά με το χρήστη μπορεί να είναι διαθέσιμες χωρίς να χρειαστεί κάποια επιπλέον ενέργεια από αυτόν. Για παράδειγμα η χώρα από την οποία προέρχεται ο χρήστης μπορεί να γίνει γνωστή στην υπηρεσία από τη διεύθυνση IP την οποία χρησιμοποιεί. Με τον ίδιο τρόπο γίνεται γνωστή και η γλώσσα που χρησιμοποιεί ο χρήστης. Επομένως, οι διάφορες εφαρμογές και υπηρεσίες μπορούν και συνιστάται να εκμεταλλεύονται τις παραπάνω πληροφορίες τόσο για όφελος του τελικού χρήστη όσο και της ίδια της υπηρεσίας. Για παράδειγμα αν ο χρήστης χρησιμοποιεί την ελληνική γλώσσα, τότε τις περισσότερες φορές θα προτιμήσει να παρουσιαστεί μία εφαρμογή, υπηρεσία ή ιστοσελίδα στα ελληνικά. Ομοίως μία ιστοσελίδα η οποία προβάλλει διαφημίσεις, προτιμά ανάλογα με τη χώρα από την οποία προέρχεται ο χρήστης, να εμφανίζει σε αυτόν και τις αντίστοιχες διαφημίσεις.

Για περισσότερο, βέβαια, εξειδικευμένες πληροφορίες δεν είναι εύκολο το σύστημα να βγάλει συμπεράσματα χωρίς να λαμβάνει υπόψη τις ενέργειες του χρήστη. Για το σκοπό αυτό, για την προσωποποίηση ουσιαστικά του πολυμεσικού περιεχομένου, είναι ανάγκη να δημιουργηθεί ένα προφίλ χρήστη. Με τον τρόπο αυτό, ο χρήστης θα μπορεί κάθε φορά να συνδέεται σε αυτό το προφίλ, και η υπηρεσία θα εξατομικεύει το περιεχόμενό της προς όφελός του. Εφαρμογές, ιστοσελίδες ή υπηρεσίες δημιουργούν, δηλαδή, ένα προφίλ για κάθε χρήστη το οποίο περιέχει τις προτιμήσεις, τις πληροφορίες και τις ρυθμίσεις που έχουν σχέση με αυτόν. Τις εξατομικευμένες αυτές πληροφορίες και υπηρεσίες θα μπορεί ο χρήστης να χρησιμοποιήσει και να επεξεργαστεί ανεξάρτητα από τη συσκευή την οποία χρησιμοποιεί, διότι οι πληροφορίες αυτές θα βρίσκονται αποθηκευμένες στον λογαριασμό του, τις οποίες θα ανακτήσει όταν συνδεθεί σε αυτόν από οποιαδήποτε συσκευή και τοποθεσία. Με τον τρόπο αυτό, ο χρήστης νιώθει την υπηρεσία ή εφαρμογή πιο φιλική σε αυτόν, πιο ευέλικτη και εύχρηστη, με αποτέλεσμα να αποκτάει ένα επιπλέον κίνητρο για να κάνει χρήση της υπηρεσίας ακόμα πιο συχνά.

Εκτός, όμως, από τις πληροφορίες και προτιμήσεις τις οποίες δηλώνει ο ίδιος ο χρήστης, η υπηρεσία θα πρέπει να αποκτά πληροφορίες για το χρήστη, τις οποίες δεν έχει δηλώσει ή αποθηκεύσει κάπου αυτός. Η εργασία αυτή υλοποιείται μέσω της συνεχούς παρακολούθησης των ενεργειών του χρήστη στη συγκεκριμένη υπηρεσία. Δηλαδή όλες οι κινήσεις του χρήστη στην υπηρεσία αποθηκεύονται, ώστε να προκύψουν συμπεράσματα για το προφίλ και τις προτιμήσεις του χρήστη, τα οποία χρησιμοποιεί η υπηρεσία για την προσωποποίηση.

Ο χρήστης καθημερινά εκτελεί έναν τεράστιο αριθμό ενεργειών σε διάφορες υπηρεσίες. Προκύπτει, λοιπόν, το ερώτημα πώς όλες αυτές οι ενέργειες θα μπορούν να στέλνονται στον αντίστοιχο εξυπηρετητή για επεξεργασία. Ένας τρόπος να

υλοποιηθεί το παραπάνω, είναι οι ενέργειες και πληροφορίες αυτές να αποθηκεύονται σε κάποιο αρχείο (σε δηλωτική γλώσσα όπως XML) στην τοπική συσκευή που χρησιμοποιεί ο χρήστης. Έπειτα ο εξυπηρετητής μπορεί να στέλνει συγκεκριμένες ερωτήσεις (queries) για τις τιμές των πεδίων του αρχείου αυτού, έτσι ώστε να αποκτά την πληροφορία για κάποια συγκεκριμένη ενέργεια. Οι πληροφορίες αυτές που συνήθως περιέχει το προφίλ του χρήστη, είναι μία λίστα με τα θέματα τα οποία τον ενδιαφέρουν, ο αριθμός επισκέψεων του χρήστη σε κάποιο συγκεκριμένο πόρο της υπηρεσίας, η διάρκεια χρήσης ενός συγκεκριμένου πόρου, καθώς και το ποσοστό ενδιαφέροντος του πόρου αυτού, το οποίο προκύπτει από τον αριθμό των επισκέψεων του χρήστη στον πόρο αυτό διαιρούμενο με το συνολικό αριθμό επισκέψεων του χρήστη στην υπηρεσία.

Πέρα από το προφίλ χρήστη, η υπηρεσία μπορεί επιπλέον να δημιουργεί ομάδες προφίλ χρηστών (user profile groups), χρησιμοποιώντας τις πληροφορίες για το προφίλ του κάθε χρήστη που λαμβάνει. Με τον τρόπο αυτό παρέχει προτάσεις στους χρήστες, με βάση τις προτιμήσεις άλλων χρηστών της ίδιας ομάδας, που πιθανότατα έχουν κοινά ενδιαφέροντα σε κάποιο συγκεκριμένο αντικείμενο. Για παράδειγμα, κάποια εφαρμογή μελετώντας τον αριθμό επισκέψεων του κάθε χρήστη σε συγκεκριμένες υπηρεσίες, θα μπορεί να τους ταξινομήσει σε ομάδες με κοινά χαρακτηριστικά. Έτσι όταν κάποιος χρήστης προσθέτει στα ενδιαφέροντά του κάποια άλλη υπηρεσία, αυτή μπορεί να προταθεί στους υπόλοιπους χρήστες της ομάδας. Με τον τρόπο αυτό, δηλαδή, χρησιμοποιούμε δύο ειδών πληροφορίες για προσωποποίηση του πολυμεσικού περιεχομένου: τις ενέργειες του συγκεκριμένου χρήστη και τις ενέργειες άλλων χρηστών με κοινά ενδιαφέροντα.

2.3 Ιστορικό χρήσης

Σημαντικό ρόλο στην προσωποποίηση περιεχομένου διαδραματίζει το ιστορικό χρήσης, το οποίο συνδέεται άμεσα με το προφίλ χρήστη και αποτελεί όλες τις υπηρεσίες τις οποίες έχει χρησιμοποιήσει ο χρήστης στο παρελθόν σε μία συγκεκριμένη εφαρμογή. Η αποθήκευση των πληροφοριών αυτών πραγματοποιείται αυτόματα, είτε τοπικά στη συσκευή του χρήστη, είτε online στον προσωπικό του λογαριασμό.

Ακολουθούν ενδεικτικά ορισμένα παραδείγματα χρήσης του ιστορικού και του προφίλ προς όφελος του τελικού χρήστη:

- Οι εφαρμογές φυλλομετρητών (browsers) αποθηκεύουν τοπικά όλες τις ιστοσελίδες τις οποίες ο χρήστης της αντίστοιχης συσκευής έχει επισκεφθεί στις προηγούμενες χρήσης της εφαρμογής. Έτσι σε περίπτωση που ο χρήστης θέλει να μεταβεί σε κάποια ιστοσελίδα που έχει επισκεφθεί στο παρελθόν και δε θυμάται την ονομασία της, μπορεί εύκολα να την εντοπίσει μέσα από το ιστορικό του.
- Σε ιστοσελίδα στην οποία ο χρήστης παρακολουθεί διάφορα βίντεο, αν κάθε φορά η υπηρεσία παρακολουθεί τα είδη των βίντεο τα οποία προτιμά ο χρήστης, τότε θα μπορεί να προτείνει και άλλα παρόμοια που πιθανότατα να τον ενδιαφέρουν.
- Σε ιστοσελίδα στην οποία ο χρήστης διαβάζει κριτικές για ταινίες, η υπηρεσία μπορεί να συμπεράνει τα είδη ταινιών τα οποία προτιμά ο χρήστης, τους αγαπημένους του ηθοποιούς κ.α., έτσι ώστε να προτείνει και άλλες ταινίες, οι οποίες πιθανόν να τον ενδιαφέρουν.

Σημειώνεται, βέβαια, ότι για λόγους ιδιωτικότητας των χρηστών, η κάθε υπηρεσία θα πρέπει να δηλώνει ρητά σε ποιες πληροφορίες και ενέργειες του χρήστη αποκτά πρόσβαση, και ο χρήστης να δηλώνει αν συμφωνεί ή διαφωνεί με τα παραπάνω.

2.4 Συμπεράσματα

Η μέθοδος της προσωποποίησης του διαδικτυακού πολυμεσικού περιεχομένου υιοθετείται τη σημερινή εποχή στις περισσότερες εφαρμογές και υπηρεσίες που χρησιμοποιεί καθημερινά ο χρήστης. Το γεγονός αυτό έχει ουσιαστικό αντίκτυπο στην εμπειρία που βιώνουν οι χρήστες από το σύστημα. Παρακάτω παρουσιάζονται τα σημαντικότερα οφέλη που προκύπτουν από την προσωποποίηση περιεχομένου τόσο για τις υπηρεσίες που την εφαρμόζουν όσο και για τους τελικούς χρήστες:

- Ο χρήστης νιώθει την υπηρεσία πιο εύχρηστη και φιλική προς αυτόν, καθώς την έχει εξατομικεύσει σύμφωνα με τα δικά του μέτρα. Με τον ίδιο τρόπο, η υπηρεσία προσφέρει στο χρήστη ένα επιπλέον κίνητρο για τη χρήση της.
- Όταν αναζητάει ο χρήστης ορισμένες πληροφορίες, είναι πιθανότερο να λάβει τα συγκεκριμένα αποτελέσματα τα οποία ζητάει, καθώς αυτά προσαρμόζονται στις προτιμήσεις του.
- Ο χρήστης μπορεί να ανακτήσει σημαντικές πληροφορίες τις οποίες δεν έχει αποθηκεύσει, μέσα από το ιστορικό χρήσης του.
- Ανεξάρτητα από τη συσκευή που χρησιμοποιεί, καθώς και τη φυσική τοποθεσία που βρίσκεται ο χρήστης, μπορεί να χρησιμοποιήσει την υπηρεσία με αποθηκευμένες όλες τις προσωπικές του προτιμήσεις.
- Εξοικονομείται χρόνος από την πλευρά του χρήστη, καθώς η υπηρεσία μπορεί να "θυμάται" ορισμένα στοιχεία τα οποία έχει εισάγει ο χρήστης στο παρελθόν.
- Εξοικονομούνται υπολογιστικοί πόροι της υπηρεσίας, καθώς δε χρειάζεται να παρουσιάζει έναν τεράστιο όγκο πληροφοριών στο χρήστη, παρά μόνο αυτές που πραγματικά χρειάζεται.

Παρά τα πολλά οφέλη που προσφέρει η προσωποποίηση περιεχομένου, χρειάζεται προσοχή στον τρόπο υλοποίησής της, καθώς αρκετοί χρήστες προτιμούν τη διαφύλαξη της ιδιωτικότητάς τους. Μία εφαρμογή, λοιπόν, θα πρέπει να παροτρύνει το χρήστη να δημιουργήσει λογαριασμό, ώστε αυτή να προσαρμόζεται στις ανάγκες του, και να δηλώνει ξεκάθαρα, σε ποια δεδομένα και πληροφορίες αποκτάει αυτή πρόσβαση. Σε περίπτωση όμως που ο χρήστης δε θελήσει να δημιουργήσει κάποιο λογαριασμό, η εφαρμογή θα πρέπει να σέβεται την επιλογή αυτή, και να συνεχίσει να προσφέρει τις υπηρεσίες της. Με λίγα λόγια μία εφαρμογή θα πρέπει να είναι λειτουργική απέναντι στο χρήστη που τη χρησιμοποιεί, είτε αυτός έχει δημιουργήσει λογαριασμό είτε όχι, όπου αυτό είναι εφικτό.

3

Προγραμματισμός σε περιβάλλον Android

3.1 Εισαγωγή

Το Android OS αποτελεί ένα λειτουργικό σύστημα το οποίο χρησιμοποιείται τα τελευταία χρόνια σε ολοένα και αυξανόμενο αριθμό κινητών τηλεφώνων (smartphones), καθώς και σε ηλεκτρονικές ταμπλέτες (tablet pc's). Όλες οι λειτουργίες του κινητού τηλεφώνου ελέγχονται από εφαρμογές (applications) τις οποίες οι χρήστες μπορούν να κατεβάσουν και να εγκαταστήσουν ελεύθερα.

Αξίζει να σημειωθεί ότι το Android OS αποτελεί ελεύθερο λογισμικό ανοιχτού κώδικα (free and open source software), πράγμα που σημαίνει ότι οι κατασκευαστές κινητών τηλεφώνων ή άλλων ηλεκτρονικών συσκευών έχουν τη δυνατότητα να το χρησιμοποιήσουν στα προϊόντα τους ελεύθερα και χωρίς κόστος. Το γεγονός αυτό αποτέλεσε τον καταλύτη στη διάδοση του λειτουργικού αυτού συστήματος. Παράλληλα τα εργαλεία για την κατασκευή εφαρμογών υπάρχουν ελεύθερα στο διαδίκτυο, έτσι ώστε οποιοσδήποτε προγραμματιστής να έχει τη δυνατότητα να κατασκευάσει και να δημοσιεύσει εφαρμογές.

Ακολουθεί μία σύντομη περιγραφή στα εργαλεία αυτά και μία ανάλυση των βασικών στοιχείων που αφορούν στην ανάπτυξη εφαρμογών σε λειτουργικό σύστημα Android.

3.2 Προγραμματιστικά εργαλεία

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή εφαρμογών Android είναι η Java, ενώ τους τελευταίους μήνες γίνονται προσπάθειες για να συμπεριληφθούν και άλλες γλώσσες όπως η C και C++. Οπότε βασική

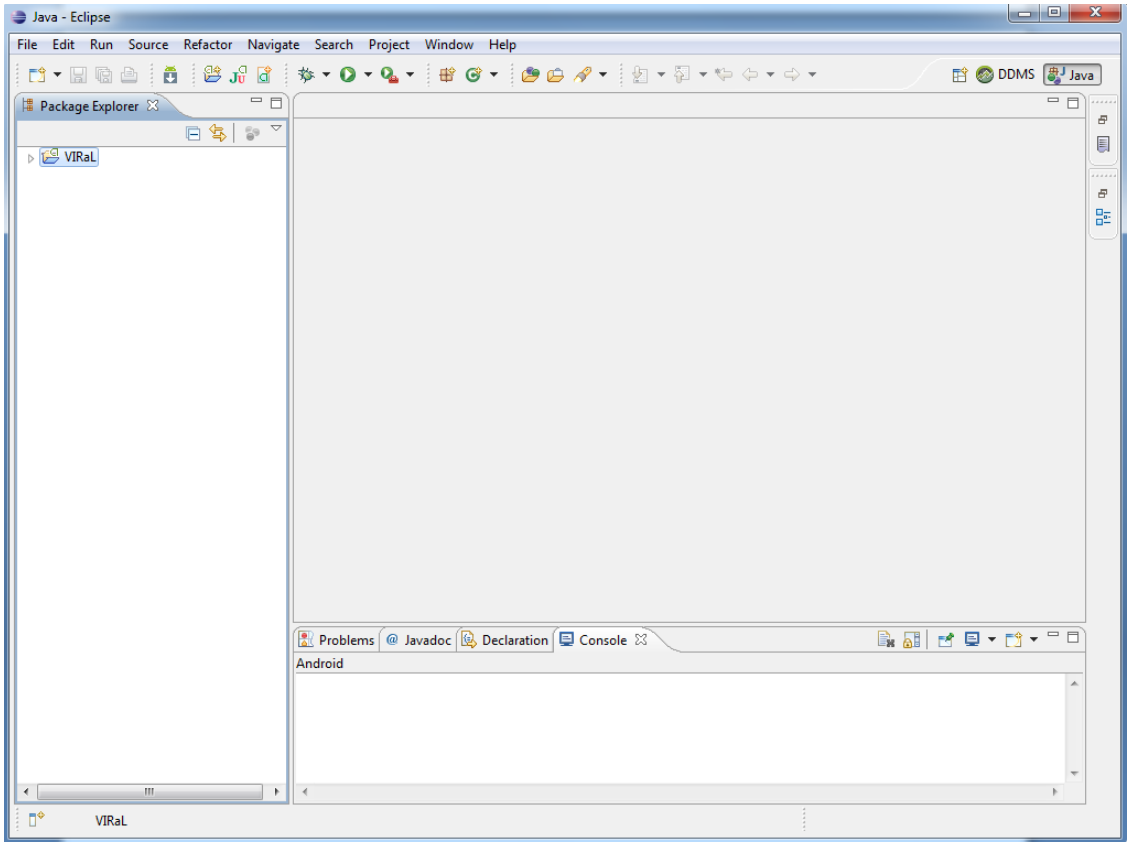
προϋπόθεση της κατασκευής είναι να διαθέτουμε τα αντίστοιχα εργαλεία της γλώσσας προγραμματισμού που θα χρησιμοποιήσουμε και συγκεκριμένα το Java Development Kit (JDK). Ακόμη, χρειαζόμαστε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment ή IDE) για να μεταγλωττίζουμε και να τρέχουμε τα προγράμματά μας, όπως το Eclipse (Εικόνα 3.1).

Το βασικότερο εργαλείο αποτελεί το Android SDK (software development kit) το οποίο ουσιαστικά μας παρέχει τα επιπλέον εργαλεία ώστε να μπορούμε να γράψουμε κώδικα συγκεκριμένα για την κατασκευή εφαρμογής Android. Η σύνδεση του Android SDK με το γραφικό μας περιβάλλον (Eclipse) γίνεται μέσω μίας επέκτασης (Android Development Tools ή ADT Plugin) που εγκαθιστάμε στο Eclipse, ώστε να μπορέσουμε να μεταγλωττίσουμε την εφαρμογή μας και έπειτα να την τρέξουμε.

Καθώς το λειτουργικό σύστημα Android κυκλοφορεί σε διάφορες εκδόσεις (με πιο γνωστές τις "Eclair", "Froyo", "Gingerbread" και "Honeycomb"), καθίσταται σαφές ότι η κάθε έκδοση θα χρησιμοποιεί και ορισμένα διαφορετικά προγραμματιστικά εργαλεία. Έτσι μέσα από την ADT Plugin μπορούμε να εγκαταστήσουμε τα εργαλεία για την υλοποίηση εφαρμογής σε οποιαδήποτε έκδοση. Για παράδειγμα αν θέλουμε η εφαρμογή μας να είναι συμβατή και σε παλαιότερες εκδόσεις του Android, τότε θα πρέπει να εγκαταστήσουμε τα αντίστοιχα εργαλεία και να δοκιμάσουμε την εφαρμογή μας στις εκδόσεις αυτές.

Τέλος, για να δοκιμάσουμε την εφαρμογή μας και να δούμε τα αποτελέσματά της θα χρειαστούμε κάποιον προσομοιωτή κινητού τηλεφώνου στον υπολογιστή μας. Τη λύση μας δίνει μία εικονική συσκευή Android (Android Virtual Device ή AVD) η οποία ουσιαστικά αποτελεί προσομοιωτή τόσο software όσο και hardware ενός κινητού τηλεφώνου με λειτουργικό σύστημα Android (Εικόνα 3.2). Τη συσκευή αυτή, την εγκαθιστάμε μέσα από την ADT Plugin, από όπου μπορούμε και να ρυθμίσουμε πολλές παραμέτρους της, όπως την έκδοση Android που θα χρησιμοποιεί, το μέγεθος της οθόνης, το μέγεθος της κάρτας SD και της cache, καθώς και άλλα χαρακτηριστικά όπως δυνατότητα λήψης φωτογραφιών (κάμερας), δέκτη GPS κλπ. Απαιτούνται βέβαια αρκετοί υπολογιστικοί πόροι για την εκτέλεση της AVD, πράγμα που καθιστά τις περισσότερες φορές σχετικά αργή την εκτέλεση της εφαρμογής μας στη συσκευή αυτή. Φυσικά, σε κάθε περίπτωση μπορούμε να εγκαθιστάμε και να εκτελούμε τις εφαρμογές μας σε φυσική συσκευή, όπως κινητό τηλέφωνο ή tablet pc, που χρησιμοποιεί το λειτουργικό σύστημα Android.

Στο σημείο αυτό, πρέπει να σημειώσουμε ότι όλα τα παραπάνω προγραμματιστικά εργαλεία διατίθενται ελεύθερα στο διαδίκτυο από τους κατασκευαστές τους, τα οποία μπορούμε να κατεβάσουμε και να χρησιμοποιήσουμε χωρίς κόστος.



Εικόνα 3.1: Το περιβάλλον ανάπτυξης λογισμικού Eclipse

Εικόνα 3.2: Εικονική συσκευή με λειτουργικό σύστημα Android (AVD)

3.3 Συστατικά στοιχεία εφαρμογής

Με τον όρο αυτό, εννοούμε τα απαραίτητα δομικά στοιχεία μίας εφαρμογής Android. Το κάθε στοιχείο από αυτά είναι ουσιαστικά ένας τρόπος πρόσβασης του λειτουργικού συστήματος στην εφαρμογή μας.

Μπορούμε να τα διακρίνουμε σε τέσσερα βασικά στοιχεία: τις δραστηριότητες (activities), τις υπηρεσίες (services), τους παρόχους περιεχομένου (content providers) και τους καθολικούς παραλήπτες μηνυμάτων (broadcast receivers).

3.3.1 Activities

Αποτελούν το βασικότερο στοιχείο κάθε εφαρμογής. Η κάθε activity αποτελεί μία διαφορετική οθόνη (παράθυρο) της εφαρμογής μας, με την οποία ο χρήστης αλληλεπιδρά. Σε αυτήν δηλαδή φορτώνεται το γραφικό περιβάλλον το οποίο τελικά βλέπει ο χρήστης.

Κάθε εφαρμογή αποτελείται συνήθως από αρκετές activities, οι οποίες συνδέονται μεταξύ τους, δηλαδή κάθε activity μπορεί να καλέσει κάποια άλλη. Η πρώτη οθόνη που θα δει ο χρήστης όταν τρέξει την εφαρμογή αποτελεί την κεντρική (main) activity, από την οποία προέρχονται όλες οι υπόλοιπες. Τη στιγμή που η εφαρμογή μας καλεί νέα activity, ο χρήστης βλέπει τη νέα οθόνη και με το πλήκτρο back μπορεί να επιστρέψει στην προηγούμενη, δηλαδή έχουμε μία μορφή στοίβας (LIFO). Αν δηλαδή η activity A καλέσει την activity B και αυτή την activity C (A->B->C) τότε στην activity C, με το πλήκτρο back επιστρέφουμε στην B κ.ο.κ. Φυσικά κάποιες activities μπορούμε να επιλέξουμε να μην τις αποθηκεύουμε στη στοίβα, δηλαδή αν στην περίπτωσή μας επιλέξουμε η B να μην αποθηκευτεί στη στοίβα, τότε με το πλήκτρο back από την C θα επιστρέψουμε στην A. Αν πατηθεί το πλήκτρο back στη main activity τότε βγαίνουμε από την εφαρμογή μας.

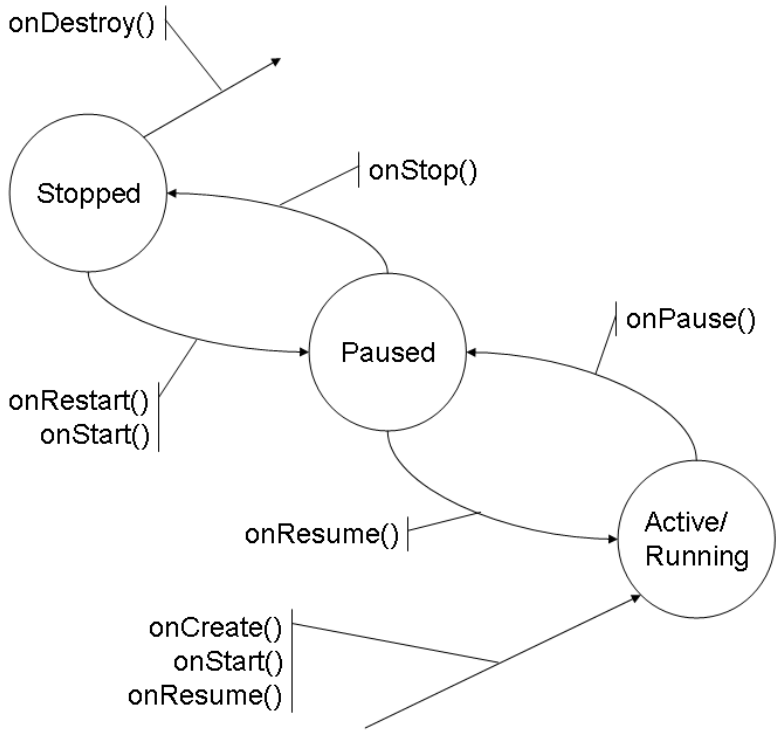
Θεωρούμε το παράδειγμα μιας απλής εφαρμογής ηλεκτρονικού ταχυδρομείου, η οποία μόλις ανοίγει μας εμφανίζει 3 επιλογές ή κουμπιά: Inbox, Compose, Sent Mail. Η συνηθέστερη υλοποίηση της εφαρμογής αυτής είναι να δημιουργήσουμε συνολικά 5 διαφορετικές οθόνες, δηλαδή 5 activities: 1. Η main activity που εμφανίζει τις επιλογές, 2. Inbox, δηλαδή η λίστα με τα εισερχόμενα μηνύματα, 3. ReadEmail, δηλαδή η οθόνη που διαβάζουμε ένα email, 4. Compose, δηλαδή η οθόνη που γράφουμε ένα email για να το στείλουμε και 5. SentMail, δηλαδή η λίστα με τα εξερχόμενα μηνύματά μας.

Κάθε activity πρέπει να υλοποιεί τη μέθοδο onCreate() η οποία καλείται τη στιγμή που ξεκινάει η activity. Η μέθοδος αυτή καλείται μόνο όταν τρέξουμε μία νέα activity είτε από κάποιο κουμπί είτε από κάποια επιλογή, δηλαδή δεν καλείται αν επιστρέψουμε με το πλήκτρο back σε κάποια προηγούμενη. Όταν επιστρέφουμε σε κάποια προηγούμενη activity βλέπουμε το στιγμιότυπο που αφήσαμε και η activity δεν ξαναδημιουργείται. Επίσης μέσα στη μέθοδο αυτή (συνήθως στην αρχή) πρέπει να δημιουργηθεί η διεπαφή χρήστη (User Interface ή UI), η οποία δηλώνεται με την εντολή setContentView(). (Το UI περιγράφεται σε επόμενη παράγραφο).

Για να ξεκινήσουμε μία activity B μέσα από μία activity A χρησιμοποιούμε τις παρακάτω εντολές: Intent intent = new Intent(this, MyActivityB.class); startActivity(intent);

Για να τερματίσουμε μία activity, αρκεί η εντολή finish(). Σε γενικές γραμμές πάντως, το Android διαχειρίζεται τον κύκλο ζωής της κάθε activity, οπότε δε χρειάζεται να τις τερματίζουμε εμείς, παρά μόνο όταν θέλουμε οπωσδήποτε να μη

μπορεί να επιστρέψει ο χρήστης σε αυτό το στιγμιότυπο της activity. Ο κύκλος ζωής και οι αντίστοιχες μέθοδοι των activities φαίνονται στην εικόνα 3.3.



Εικόνα 3.3: Ο κύκλος ζωής των activities και οι αντίστοιχες μέθοδοι

3.3.2 Services

Μία service αποτελεί ένα στοιχείο της εφαρμογής, το οποίο τρέχει στο παρασκήνιο (background) και μπορεί να εκτελέσει αρκετές και χρονοβόρες λειτουργίες. Ειδοποιός διαφορά με την activity είναι το ότι η service δεν παρέχει user interface. Επίσης, η service μπορεί να εκτελείται ακόμα και όταν τρέξουμε κάποια διαφορετική εφαρμογή από αυτήν που την ξεκίνησε.

Παράδειγμα υπηρεσίας αποτελεί η εφαρμογή για αναπαραγωγή μουσικής. Όταν ακούμε μουσική, είναι πιθανό να θέλουμε να συνεχίσουμε να την ακούμε ακόμα και αν τρέξουμε κάποια άλλη εφαρμογή και δίχως να απαιτούμε γραφικό περιβάλλον για την αναπαραγωγή. Επιπλέον όλη η επικοινωνία με το δίκτυο θέλουμε να συνεχίσει όταν ανοίξουμε κάποια εφαρμογή και να μην σταματήσει, άρα θα πρέπει να λειτουργεί σαν service.

Δύο βασικές λειτουργίες σχετίζονται με τις υπηρεσίες: η startService() και η bindService(). Η πρώτη καλείται από κάποια activity για να ξεκινήσει η service και θα συνεχίσει να εκτελείται ακόμα και αν η activity που την κάλεσε τερματιστεί. Για το λόγο αυτό η service δεν επιστρέφει κάποιο αποτέλεσμα στην activity που την κάλεσε, αλλά εκτελεί κάποια λειτουργία όπως το κατέβασμα κάποιου αρχείου από το διαδίκτυο. Η δεύτερη λειτουργία "δένει" τη service με κάποιο άλλο στοιχείο της εφαρμογής, πχ με μία activity, έτσι ώστε να υπάρχει αλληλεπίδραση μεταξύ των στοιχείων αυτών, όπως π.χ. αποστολή και λήψη αποτελεσμάτων. Στην περίπτωση

αυτή, αν τερματιστεί το στοιχείο που έχει "δεθεί" με τη service, τότε τερματίζεται και η service.

Προηγουμένως αναφέραμε ότι η service δεν παρέχει διεπαφή χρήστη, επομένως γεννιέται το ερώτημα, πώς θα ενημερώνεται ο χρήστης για κάποιο αποτέλεσμα της υπηρεσίας. Υπάρχουν δύο τρόποι: η ειδοποίηση Toast και η ειδοποίηση μέσω της Status Bar. Οι ειδοποιήσεις αυτές αναλύονται στην παράγραφο 3.4.4, καθώς αφορούν στο user interface.

Σε αντίθεση με τις activities, το λειτουργικό σύστημα δε διαχειρίζεται τον κύκλο ζωής των services (παρά μόνο όταν υπάρχει έλλειψη υπολογιστικών πόρων), επομένως η κάθε service θα πρέπει να δηλώνει το πότε θα σταματήσει με την εντολή stopSelf().

3.3.3 Content Providers

Οι content providers διαχειρίζονται αποθηκευτικούς χώρους για δεδομένα, οι οποίοι είναι προσπελάσιμοι από οποιαδήποτε εφαρμογή. Αποτελούν το μοναδικό τρόπο για αποθήκευση δεδομένων, τα οποία θέλουμε να είναι προσβάσιμα από αρκετές εφαρμογές. Για παράδειγμα μία εφαρμογή τηλεφωνικού καταλόγου, θα θέλαμε να αποθηκεύει σε συγκεκριμένο χώρο τις επαφές μας, ώστε αργότερα αν θελήσουμε να χρησιμοποιήσουμε διαφορετική εφαρμογή, να μπορεί αυτή να εντοπίσει και να επεξεργαστεί τα ήδη υπάρχοντα δεδομένα. Παρόμοια παραδείγματα αποτελούν τα αρχεία ήχου, βίντεο, οι εικόνες κλπ τα οποία θέλουμε να αποθηκεύονται σε κάποιους προεπιλεγμένους χώρους, ώστε να είναι προσβάσιμα από περισσότερες από μία εφαρμογές.

Στο πλαίσιο αυτό υπάρχει ενσωματωμένη βάση δεδομένων στο λειτουργικό σύστημα Android (SQLite database), στην οποία μπορούν να αποθηκεύουν ή να διαβάζουν δεδομένα οι content providers.

3.3.4 Broadcast Receivers

Οι broadcast receivers ενημερώνονται για κάποιο συγκεκριμένο γεγονός από το λειτουργικό σύστημα και τότε ενεργοποιούνται. Τέτοια γεγονότα μπορεί να είναι ότι η οθόνη έχει σβήσει, ότι η στάθμη της μπαταρίας είναι χαμηλή κλπ. Για παράδειγμα, αρκετές εφαρμογές λήψης φωτογραφιών απενεργοποιούν το flash σε περίπτωση χαμηλής μπαταρίας για λόγους εξοικονόμησης ενέργειας. Ενημερώνονται δηλαδή από τον broadcast receiver για το γεγονός αυτό και πράττουν ανάλογα. Οι broadcast receivers δεν παρέχουν user interface, αλλά μπορούν να ενημερώσουν το χρήστη για κάποιο γεγονός μέσω των status bar notifications.

3.4 User Interface

Η διεπαφή χρήστη έχει τεράστια σημασία για κάθε εφαρμογή. Αποτελεί την τελική εικόνα που βλέπει ο χρήστης, το γραφικό περιβάλλον στο οποίο θα περιηγείται, και ενεργοποιεί όλες τις λειτουργίες της εφαρμογής. Το user interface και η λειτουργικότητα της εφαρμογής αποτελούν αλληλένδετα στοιχεία και χωρίς το ένα δε μπορεί να υπάρξει το άλλο. Πολλές φορές μάλιστα είναι δυσκολότερος ο σχεδιασμός ενός όμορφου και εύχρηστου περιβάλλοντος εργασίας, παρά η ίδια η λειτουργικότητα της εφαρμογής. Καθίσταται σαφές, λοιπόν, ότι απαιτεί μεγάλη προσπάθεια και προσοχή η δημιουργία ενός γραφικού περιβάλλοντος που θα

προσελκύει τους χρήστες και θα τους ωθεί να χρησιμοποιούν μία συγκεκριμένη εφαρμογή έναντι μίας άλλης, με τις ίδιες λειτουργίες.

3.4.1 Layout

Σε κάθε οθόνη της εφαρμογής, πρωταρχικό στοιχείο του γραφικού περιβάλλοντος αποτελεί η διάταξη των γραφικών στοιχείων ή layout. Το layout περιλαμβάνει όλα τα γραφικά στοιχεία της οθόνης, τα οποία μπορεί να είναι διατεταγμένα σε επιμέρους layouts.

Υπάρχουν 4 είδη layout, τα LinearLayout (γραμμική διάταξη), RelativeLayout (σχετική διάταξη), FrameLayout (διάταξη πλαισίου) και TableLayout (διάταξη πίνακα). (Υπάρχει και το AbsoluteLayout, το οποίο όμως έχει προταθεί να μην χρησιμοποιείται πλέον, διότι ορίζει τις απόλυτες θέσεις κάθε στοιχείου, οι οποίες όμως διαφέρουν ανάλογα με το κινητό τηλέφωνο και την οθόνη που χρησιμοποιείται η εφαρμογή). Το LinearLayout αποτελεί διάταξη στοιχείων σε οριζόντια ή κατακόρυφη σειρά. Αν δηλαδή δηλώσουμε τρία στοιχεία A, B, C μέσα σε ένα οριζόντιο LinearLayout τότε τα στοιχεία αυτά θα εμφανίζονται στην οθόνη σε μία οριζόντια διάταξη με τη σειρά που τα δηλώσαμε το ένα δίπλα στο άλλο. Το RelativeLayout μας δίνει περισσότερη ελευθερία στη δήλωση των γραφικών στοιχείων, με την έννοια ότι κάθε στοιχείο μπορούμε να επιλέξουμε να το εμφανίσουμε σε συγκεκριμένο σημείο της οθόνης, όπως π.χ. στην αρχή, στο κέντρο, στο τέλος ή να επιλέξουμε τη θέση του σε σχέση με κάποιο άλλο στοιχείο. Το FrameLayout αποτελεί την απλούστερη διάταξη στοιχείων. Είναι απλά ένας κενός χώρος τον οποίο μπορούμε να γεμίσουμε με κάποιο αντικείμενο, π.χ. μία εικόνα. Για το λόγο αυτό συνήθως χρησιμοποιείται σαν ρίζα (root) στο δέντρο των γραφικών στοιχείων της οθόνης. Τέλος, το TableLayout όπως φανερώνει και η ονομασία του αποτελεί διάταξη πίνακα, δηλαδή μπορεί να διατάσσει τα παιδιά του (children) σε σειρές και στήλες. Σε όλα τα παραπάνω στοιχεία μπορούμε να ρυθμίσουμε αρκετές παραμέτρους όπως μέγεθος (πλάτος και ύψος), οριζόντια ή κατακόρυφη διάταξη, βαρύτητα (layout gravity) και άλλες.

Υπάρχουν δύο τρόποι για να δηλώσουμε τα layouts (όπως και κάθε άλλο γραφικό στοιχείο) της εφαρμογής μας: α) μέσω αρχείων xml ή β) μέσα στις activities της εφαρμογής μας. Τα αρχεία xml αποτελούν στατικό τρόπο δημιουργίας των γραφικών στοιχείων. Τα αποθηκεύουμε σε συγκεκριμένο φάκελο του project μας και τα καλούμε για δημιουργία του γραφικού περιβάλλοντος μέσα από τις activities. Παρουσιάζουν δενδρική δομή για εύκολη συγγραφή και κατανόηση του περιεχομένου τους. Πολλές φορές ωστόσο δε γνωρίζουμε εξαρχής τη διάταξη που θα χρησιμοποιήσουμε, διότι ίσως να εξαρτάται από ορισμένες επιλογές του χρήστη. Στις περιπτώσεις αυτές δημιουργούμε δυναμικά τα layouts μέσα στις activities και ορίζουμε εκεί τις παραμέτρους αυτών. Ωστόσο ο πιο εύκολος και συνηθέστερος τρόπος είναι (αν έχουμε τη δυνατότητα) να δημιουργήσουμε για κάθε οθόνη ένα διαφορετικό xml αρχείο που θα περιλαμβάνει τη διάταξη όλων των γραφικών της στοιχείων, και να το καλέσουμε μέσα από την αντίστοιχη activity.

Ακολουθούν δύο παραδείγματα διάταξης των γραφικών στοιχείων της οθόνης.

(Εικόνα 3.4)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
```

```
   xmlns:android="http://schemas.android.com/apk/res/android"
   >
   <TextView
      android:layout_width="105px"
      android:layout_height="wrap_content"
      android:text="Hello World,\nUIActivity"
      />
   <Button
      android:layout_width="100px"
      android:layout_height="wrap_content"
      android:text="Button"
      />
</LinearLayout>
```

Εικόνα 3.4: Παράδειγμα κατασκευής ενός LinearLayout που περιλαμβάνει ένα κείμενο και ένα κουμπί

(Εικόνα 3.5)

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_height="fill_parent"
   android:layout_width="fill_parent"
   android:background="#000044">
   <TableRow>
      <TextView
         android:text="User Name:"
         android:width ="120px"
         />
      <EditText
         android:id="@+id/txtUserName"
         android:width="200px" />
   </TableRow>
   <TableRow>
      <TextView
         android:text="Password:"
         />
      <EditText
         android:id="@+id/txtPassword"
         android:password="true"
```

```
        />
    </TableRow>
    <TableRow>
        <TextView />
        <CheckBox android:id="@+id/chkRememberPassword"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Remember Password"
            />
    </TableRow>
    <TableRow>
        <Button
            android:id="@+id/buttonSignIn"
            android:text="Log In" />
    </TableRow>
</TableLayout>
```

Εικόνα 3.5: Παράδειγμα κατασκευής ενός TableLayout το οποίο ζητάει όνομα χρήστη και κωδικό πρόσβασης

3.4.2 Μενού

Τα μενού αποτελούν ένα σημαντικό κομμάτι της διεπαφής χρήστη για κάθε οθόνη της εφαρμογής, διότι παρέχουν στο χρήστη ένα γνωστό και φιλικό τρόπο για να εισάγει τις επιλογές του. Στο λειτουργικό σύστημα Android, υπάρχουν τρία διαφορετικά είδη μενού, το μενού επιλογών (options menu), το μενού πλαισίου (context menu) και το υπομενού (submenu), τα οποία δηλώνονται και αυτά σε αρχεία xml.

3.4.2.1 Options Menu

Το options menu αποτελεί το βασικότερο μενού μίας εφαρμογής. Εμφανίζεται τη στιγμή που πατάμε το κουμπί menu του κινητού μας τηλεφώνου και περιέχει όλες τις βασικές επιλογές της εφαρμογής μας. Αποτελεί κυρίως τον τρόπο με τον οποίο περιηγούμαστε μεταξύ των διαφορετικών οθονών και activities της εφαρμογής μας. Στον κώδικα της εφαρμογής μας ορίζουμε κάθε επιλογή του μενού

σε ποια activity θα οδηγήσει το χρήστη. Παράδειγμα ενός options menu φαίνεται στην παρακάτω εικόνα:

Εικόνα 3.6: Το μενού επιλογών μίας εφαρμογής περιήγησης στο διαδίκτυο (browser)

3.4.2.2 Context Menu

Γνωρίζουμε όλοι τις επιλογές που μας δίνει το δεξί κλικ σε διάφορα προγράμματα ή εικονίδια στα λειτουργικά συστήματα που χρησιμοποιούνται στους υπολογιστές. Το ζήτημα είναι πώς θα εμφανίσουμε παρόμοιες επιλογές στις εφαρμογές Android, αφού διαθέτουμε οθόνη αφής. Τη λύση έρχεται να μας δώσει το context menu το οποίο περιλαμβάνει αυτές ακριβώς τις επιλογές σε οποιοδήποτε γραφικό στοιχείο το ορίσουμε (εικόνα, κείμενο κλπ) και ενεργοποιείται από τον χρήστη με παρατεταμένο πάτημα (press and hold ή long press) του στοιχείου αυτού. Για παράδειγμα στο context menu ενός κειμένου θα ορίζαμε επιλογές "αντιγραφή", "αποκοπή" κλπ, στο context menu μίας διεύθυνσης URL σε εφαρμογή web browser θα ορίζαμε επιλογές "άνοιγμα", "άνοιγμα σε νέα καρτέλα" κ.ο.κ.

3.4.2.3 Submenu

Το submenu μπορεί να προστεθεί σαν επιλογή στα δύο παραπάνω menu και όπως δείχνει και το όνομά του ανοίγει ένα επιπλέον μενού για περισσότερες επιλογές. Χρησιμοποιείται σε περιπτώσεις που η εφαρμογή μας εκτελεί αρκετές λειτουργίες και θέλουμε να τις οργανώσουμε σε μενού. (Αντίστοιχα με τις επιλογές "Αρχείο", "Επεξεργασία", "Προβολή" κλπ που διαθέτουν τα περισσότερα προγράμματα).

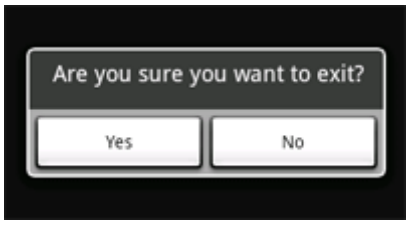
3.4.3 Dialogs

Ο διάλογος (dialog) είναι συνήθως ένα μικρό παράθυρο που εμφανίζεται στην οθόνη μπροστά από την activity που τον κάλεσε. Η activity αυτή χάνει την εστίαση που είχε (focus) και το παράθυρο του διαλόγου είναι το μοναδικό με το

οποίο μπορεί να αλληλεπιδράσει ο χρήστης. Χρησιμοποιείται είτε για ενημέρωση του χρήστη για κάποιο γεγονός είτε για να ορίσει ο χρήστης κάποια επιλογή του. Τα κυριότερα είδη διαλόγων είναι ο AlertDialog (διάλογος ειδοποίησης) και ο ProgressDialog (διάλογος προόδου).

Ο AlertDialog είναι ο πιο συνηθισμένος διάλογος. Αποτελείται από ένα τίτλο, ένα μήνυμα, ορισμένα κουμπιά ή μία λίστα από επιλογές. Για κάθε κουμπί του διαλόγου, ορίζουμε μέσα στην activity τις ενέργειες που θα ακολουθήσουν όταν το πατήσει ο χρήστης. Ακολουθεί ένα παράδειγμα κώδικα για τη δημιουργία ενός AlertDialog με δύο κουμπιά, καθώς και το αποτέλεσμα που παράγει.

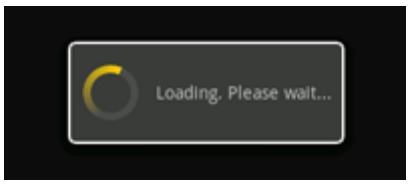
```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage("Are you sure you want to exit?")
       .setCancelable(false)
       .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
              MyActivity.this.finish();
          }
       })
       .setNegativeButton("No", new DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int id) {
              dialog.cancel();
          }
       });
AlertDialog alert = builder.create();
```



Εικόνα 3.7: Παράδειγμα AlertDialog

Ο ProgressDialog αποτελεί ουσιαστικά επέκταση του AlertDialog και χρησιμοποιείται όταν θέλουμε να εμφανίσουμε στο χρήστη την πρόοδο για κάποια ενέργεια. Για παράδειγμα όταν θέλουμε να κατεβάσουμε κάποιες εικόνες από το διαδίκτυο και να τις εμφανίσουμε στο χρήστη, θα χρειαστούμε κάποιο χρονικό διάστημα για να ολοκληρωθεί αυτή η ενέργεια. Οπότε για να μη βλέπει ο χρήστης μία κενή μαύρη οθόνη, εμφανίζουμε έναν ProgressDialog και στο background εκτελούμε τις χρονοβόρες διαδικασίες. Παρακάτω φαίνεται η απλούστερη μορφή ενός ProgressDialog.

```
ProgressDialog dialog = ProgressDialog.show(MyActivity.this, "",
                "Loading. Please wait...", true);
```



Εικόνα 3.8: Παράδειγμα ProgressDialog

Υπάρχουν δύο τρόποι για να ακυρώσουμε έναν διάλογο. Τις περισσότερες φορές θέλουμε να δώσουμε στο χρήστη την ελευθερία να ακυρώσει έναν διάλογο με τον πιο φυσικό τρόπο του Android, δηλαδή το back button. Το γεγονός αυτό το επιτυγχάνουμε ορίζοντας τον διάλογό μας cancellable (ακυρώσιμο). Ακόμη, στις ενέργειες που ακολουθούν όταν ο χρήστης πατήσει κάποιο κουμπί, τις περισσότερες φορές πρέπει να συμπεριλάβουμε και την εντολή myDialog.dismiss() (αν έχουμε ονομάσει τον διάλογό μας myDialog) ώστε ο διάλογός μας να εξαφανιστεί και έπειτα να συνεχίσει η ροή της εφαρμογής ανάλογα με την επιλογή του χρήστη.

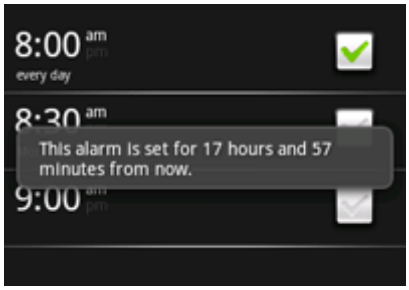
3.4.4 Ειδοποιώντας το χρήστη

Σε πολλές περιπτώσεις θέλουμε να ενημερώσουμε το χρήστη για κάποιο γεγονός ή αποτέλεσμα σχετικό με την εφαρμογή μας. Ορισμένα από αυτά τα γεγονότα απαιτούν κάποια απάντηση από το χρήστη και άλλα όχι. Για παράδειγμα όταν ο χρήστης αποθηκεύει ένα αρχείο, θα θέλαμε να δει κάποιο μήνυμα ότι το αρχείο αποθηκεύτηκε επιτυχώς. Ή όταν η εφαρμογή μας τρέχει στο background και θέλει να ενημερώσει το χρήστη για κάποιο γεγονός, θα πρέπει να στείλει κάποια ειδοποίηση την οποία ο χρήστης να μπορεί να "ανοίξει" όταν αυτός επιθυμεί. Στην πρώτη περίπτωση χρησιμοποιούμε toast notification, ενώ στη δεύτερη status bar notification.

3.4.4.1 Toast Notification

Η toast notification είναι ένα μήνυμα που εμφανίζεται για λίγα δευτερόλεπτα στο παράθυρο που βρίσκεται ο χρήστης, οποιασδήποτε εφαρμογής και αν είναι αυτό. Ο χώρος που καταλαμβάνει είναι ο ελάχιστος απαιτούμενος ώστε το μήνυμα να είναι εμφανές, ενώ ο χρήστης μπορεί όσο εμφανίζεται το μήνυμα να αλληλεπιδρά με την activity στην οποία βρίσκεται. Δεν υπάρχει κάποια επιλογή σε αυτή την ειδοποίηση, παρά μόνο ενημέρωση, δηλαδή ο χρήστης δε μπορεί να αλληλεπιδράσει με την ειδοποίηση. Χρησιμοποιείται συνήθως για μικρά μηνύματα που δεν απαιτούν κάποια ενέργεια από το χρήστη, όπως για παράδειγμα "Το αρχείο αποθηκεύτηκε επιτυχώς", "Το ξυπνητήρι ορίστηκε στις …" κλπ. Παρακάτω βλέπουμε ένα παράδειγμα μίας toast notification.

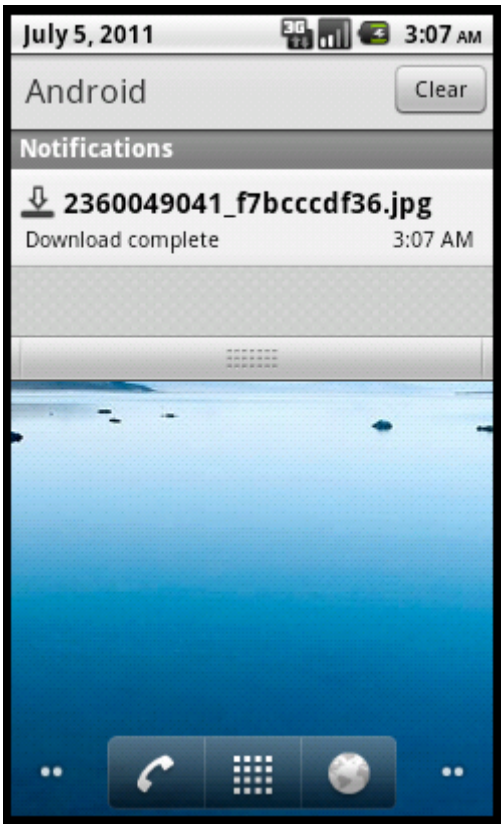
```
Context context = getApplicationContext();
CharSequence text = "This alarm is set for " + hours + " hours and " + minutes + " minutes from
now.";
int duration = Toast.LENGTH_SHORT;

Toast toast = Toast.makeText(context, text, duration);
toast.show();
```



Εικόνα 3.9: Παράδειγμα toast notification

3.4.4.2 Status Bar Notification

Η status bar notification, όπως φανερώνει και το όνομά της, είναι μία ειδοποίηση η οποία εμφανίζεται στη status bar του κινητού τηλεφώνου μας, και την οποία μπορούμε να ανοίξουμε είτε βρισκόμαστε στο κεντρικό μενού του τηλεφώνου μας, είτε σε κάποια εφαρμογή. Αντίθετα με την toast, η status bar notification μπορεί να επιλεχθεί και να ξεκινήσει κάποια λειτουργία ανάλογα με τις ενέργειες που έχουμε ορίσει στον κώδικα της εφαρμογής. Για παράδειγμα, όταν κατεβάζουμε ένα αρχείο από το διαδίκτυο, όταν η λήψη ολοκληρωθεί, θα θέλαμε να επιλέξουμε την ειδοποίηση αυτή και με τον τρόπο αυτό είτε να ανοίξουμε το φάκελο που βρίσκεται το αρχείο, είτε να το τρέξουμε (Εικόνα 3.10). Τις περισσότερες φορές οι toast notifications ενεργοποιούνται από activities, ενώ οι status bar notifications από services.



Εικόνα 3.10: Παράδειγμα status bar notification

3.5 Application Resources & Συμβατότητα

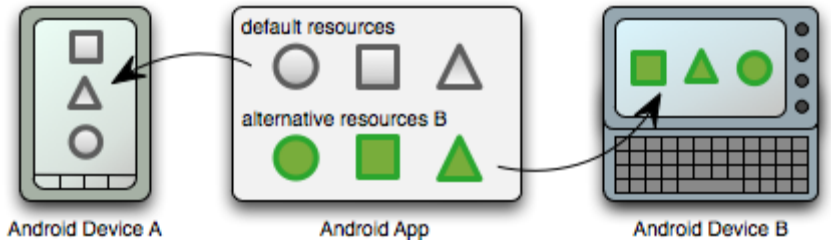
Με τον όρο application resources εννοούμε όλα τα στατικά στοιχεία τα οποία χρησιμοποιεί η εφαρμογή μας, όπως για παράδειγμα οι εικόνες και τα strings που χρησιμοποιούμε στον κώδικα. Όλα αυτά τα στοιχεία θα πρέπει να είναι ανεξάρτητα από τον κώδικα της εφαρμογής, δηλαδή να δηλώνονται σε διαφορετικά σημεία από τις κλάσεις μας, για λόγους συμβατότητας της εφαρμογής μας. Δηλαδή αν η εφαρμογή μας εμφανίζει σε μία οθόνη κάποιες εικόνες, θα θέλαμε ίσως να τις εμφανίσουμε με διαφορετικό τρόπο όταν βρισκόμαστε σε portrait mode και με διαφορετικό όταν βρισκόμαστε σε landscape. Επίσης, για να κάνουμε την εφαρμογή να υποστηρίζει αρκετές γλώσσες θα μπορούσαμε να κρατάμε σε διαφορετικό αρχείο τα strings για κάθε γλώσσα και να τα προσπελαύνουμε με τον ίδιο τρόπο από τον

κώδικα της εφαρμογής μας. Έτσι λοιπόν, για κάθε στοιχείο που θα χρησιμοποιήσουμε ορίζουμε κάθε φορά το προεπιλεγμένο (default) και έπειτα αν θέλουμε και άλλα εναλλακτικά (alternative), τα οποία μπορεί να εξαρτώνται από τον προσανατολισμό (orientation) της συσκευής, το μέγεθος της οθόνης, τη γλώσσα που χρησιμοποιεί ο χρήστης στη συσκευή κλπ.

3.5.1 Προσανατολισμός

Οι συσκευές με λειτουργικό σύστημα Android μπορούν να τοποθετηθούν είτε σε προσανατολισμό πορτρέτου (portrait mode) είτε σε προσανατολισμό τοπίου (landscape mode). Η πρώτη περίπτωση αφορά τον πιο συνηθισμένο τρόπο χρήσης του κινητού, αυτόν που χρησιμοποιούμε το κινητό μας τηλέφωνο σε όρθια θέση, δηλαδή η μεγαλύτερη πλευρά της οθόνης είναι κατακόρυφη. Η δεύτερη περίπτωση είναι η αντίστροφη, δηλαδή όταν η μεγαλύτερη πλευρά της οθόνης είναι οριζόντια. Αντιλαμβάνεται κανείς ότι τα γραφικά στοιχεία της οθόνης θα θέλαμε κάποιες φορές να εμφανίζονται με διαφορετικό τρόπο, ανάλογα τον προσανατολισμό της συσκευής. Το λειτουργικό σύστημα Android έχει τη δυνατότητα να επανασχεδιάζει τα γραφικά αυτά στοιχεία, όταν αλλάζουμε προσανατολισμό, και να τα εμφανίζει με τον τρόπο που έχουμε ορίσει εμείς. Ο προεπιλεγμένος τρόπος είναι να παρουσιάζεται το user interface με την ίδια διάταξη και στους δύο προσανατολισμούς. Έστω λοιπόν ότι θέλουμε να εμφανίσουμε τρεις εικόνες σε κάποια οθόνη της εφαρμογής μας, οι οποίες να καλύπτουν όλο το χώρο της οθόνης. Η καλύτερη λύση θα ήταν όταν ο χρήστης χρησιμοποιεί το κινητό του σε portrait mode, να εμφανίζονται η μία κάτω απ' την άλλη, ενώ όταν αλλάζει σε landscape mode, να εμφανίζονται η μία δίπλα στην άλλη. Στις παρακάτω εικόνες φαίνεται η διάταξη για το παραπάνω παράδειγμα σε portrait mode και σε landscape mode. Στην πρώτη εικόνα έχουμε ορίσει μόνο μία διάταξη, η οποία χρησιμοποιείται και για τους δύο προσανατολισμούς, ενώ στη δεύτερη εικόνα έχουμε ορίσει διαφορετική διάταξη για την περίπτωση του landscape mode, η οποία οπτικά είναι και καλύτερη.

Εικόνα 3.11: Κοινή διάταξη για portrait και landscape προσανατολισμό



Εικόνα 3.12: Διαφορετική διάταξη για portrait και landscape προσανατολισμό

Προηγουμένως στο user interface αναφέραμε ότι η διάταξη των γραφικών στοιχείων δηλώνεται σε αρχεία xml. Συγκεκριμένα στο project της εφαρμογής μας υπάρχει ο φάκελος res/layout/ στον οποίο τοποθετούμε όλα τα αρχεία xml που αφορούν στο user interface της εφαρμογής μας (το res προέρχεται από το resources). Έστω ότι η εφαρμογή μας περιέχει τρεις διαφορετικές οθόνες (δηλαδή τρεις διαφορετικές activities), με κάθε μία να χρησιμοποιεί το δικό της user interface. Τοποθετούμε λοιπόν τα αρχεία myfirstscreen.xml, mysecondscreen.xml και mythirdscreen.xml στον παραπάνω φάκελο. Όμως η τρίτη οθόνη της εφαρμογής μας εμφανίζει τρεις εικόνες όπως το παράδειγμα παραπάνω, άρα θέλουμε να χρησιμοποιήσουμε διαφορετική διάταξη για το landscape mode της τρίτης οθόνης. Όλα τα αρχεία xml που αφορούν στο user interface και συγκεκριμένα στην περίπτωση του landscape mode, τοποθετούνται στο φάκελο res/layout-land/ (δηλαδή layout-landscape). Άρα, λοιπόν, δημιουργούμε ένα ακόμα αρχείο xml με το ίδιο ακριβώς όνομα mythirdscreen.xml το οποίο όμως τοποθετούμε στο φάκελο res/layout-land/. Η εντολή που χρησιμοποιούμε για να δηλώσουμε το αρχείο xml που θα χρησιμοποιηθεί σε μία activity είναι η setContentView(). Επομένως σε κάθε μία από τις activities μας myFirstActivity.java, mySecondActivity.java και myThirdActivity.java γράφουμε την εντολή setContentView(R.layout.myfirstscreen), setContentView(R.layout.mysecondscreen) και setContentView(R.layout.mythirdscreen) αντίστοιχα. Όταν θα εκτελεστεί η εφαρμογή μας και συγκεκριμένα η εντολή setContentView() το λειτουργικό σύστημα Android θα ελέγξει τον προσανατολισμό της συσκευής μας και θα επιλέξει το κατάλληλο αρχείο xml. Αν βρισκόμαστε στην πρώτη οθόνη σε προσανατολισμό portrait και αλλάξουμε προσανατολισμό σε landscape, τότε το Android θα ελέγξει αν υπάρχει το αρχείο myfirstscreen.xml στον φάκελο res/layout-land/. Δε θα το βρει όμως διότι δεν το έχουμε ορίσει, άρα θα χρησιμοποιήσει το προεπιλεγμένο αρχείο το οποίο βρίσκεται στο φάκελο res/layout/ και τελικά η διάταξη των στοιχείων θα είναι κοινή και για τους δύο προσανατολισμούς. Τα ίδια ισχύουν και για τη δεύτερη οθόνη. Στην τρίτη όμως οθόνη, όταν αλλάξουμε προσανατολισμό από portrait σε landscape τότε θα βρεθεί το αρχείο mythirdscreen.xml στο φάκελο res/layout-land/ άρα τελικά θα χρησιμοποιηθεί αυτό για τη διάταξη των γραφικών στοιχείων. Φυσικά όταν επιστρέψουμε πάλι σε portrait mode θα αλλάξει και η διάταξη.

3.5.2 Μέγεθος Οθόνης

Παραπάνω μιλήσαμε για την περίπτωση αλλαγής του προσανατολισμού της συσκευής. Με τον ίδιο τρόπο μπορούμε να χρησιμοποιήσουμε διαφορετικά γραφικά στοιχεία ή διαφορετική διάταξη των στοιχείων αυτών ανάλογα με το μέγεθος της οθόνης. Για παράδειγμα αν θέλουμε να εμφανίσουμε κάποιες εικόνες, τότε στην οθόνη του κινητού τηλεφώνου (από 3 έως 4 ίντσες) θα χρησιμοποιήσουμε διαφορετική ανάλυση για την κάθε εικόνα, από αυτήν που θα χρησιμοποιήσουμε για ένα tablet pc (περίπου 10 ίντσες οθόνη). Ομοίως αν θέλουμε να εμφανίσουμε αρκετές εικόνες σε μία οθόνη, στο κινητό τηλέφωνο για να είναι ευδιάκριτες μπορούμε να εμφανίσουμε το πολύ 10-20 εικόνες, ενώ σε ένα tablet pc θα μπορούσαμε να εμφανίσουμε πολλές παραπάνω.

Ο τρόπος που υλοποιείται το παραπάνω σενάριο είναι παρόμοιος με τον τρόπο της διάταξης σε portrait και landscape προσανατολισμό. Όλες οι εικόνες αποθηκεύονται στο φάκελο res/drawable/. Για να χρησιμοποιήσουμε διαφορετικές εικόνες ανάλογα με το μέγεθος της οθόνης, τις αποθηκεύουμε σε διαφορετικούς φακέλους όπως res/drawable-small/, res/drawable-normal/ (είναι ουσιαστικά ίδιος με τον προεπιλεγμένο φάκελο res/drawable/), res/drawable-large/ και res/drawable-

xlarge/. Το λειτουργικό σύστημα θα εντοπίσει αυτόματα την οθόνη της συσκευής του χρήστη και θα επιλέξει τα κατάλληλα αρχεία. Στην περίπτωση της διαφορετικής διάταξης, π.χ. για πολύ μεγάλες οθόνες, θα πρέπει να αποθηκεύσουμε επιπλέον αρχεία xml που ορίζουν τη διάταξη, εκτός από τον φάκελο res/layout/, και στον φάκελο res/layout-xlarge/.

Στις activities, λοιπόν, θα χρησιμοποιούμε ενιαίες εντολές, χωρίς να δηλώνουμε κάτι σχετικό με το μέγεθος της οθόνης ή άλλη παράμετρο, και το λειτουργικό σύστημα θα επιλέγει αυτόματα τα κατάλληλα αρχεία από αυτά που έχουμε δηλώσει. Για παράδειγμα εμφανίζουμε μία εικόνα στην οθόνη μας με τις παρακάτω εντολές:

```
ImageView imageView = (ImageView) findViewById(R.id.myimageview);
imageView.setImageResource(R.drawable.myimage);
```

Αν έχουμε αποθηκεύσει το αρχείο myimage.jpg σε διαφορετικούς φακέλους για τα διάφορα μεγέθη της οθόνης, το λειτουργικό σύστημα θα εντοπίσει το κατάλληλο αυτόματα για να το εμφανίσει.

3.5.3 Γλώσσα

Οι εφαρμογές Android πρέπει να απευθύνονται σε ευρύ κοινό και όχι μόνο στη χώρα κατασκευής τους. Το σημαντικότερο ρόλο σε αυτό διαδραματίζει η φυσική γλώσσα που θα είναι γραμμένη η εφαρμογή. Αξίζει, λοιπόν, να προσπαθήσουμε να μεταφράσουμε την εφαρμογή μας σε πολλές διαφορετικές γλώσσες, ειδικά μάλιστα όταν το Android μας προσφέρει τέτοια δυνατότητα με αρκετά εύκολο τρόπο.

Το κείμενο που εμφανίζεται στην εφαρμογή είναι ουσιαστικά ένα σύνολο από strings. Είτε αυτό αφορά κείμενο σε κάποιο κουμπί, κείμενο κάτω από μία εικόνα, κείμενο σε μία ειδοποίηση ή ένα διάλογο. Έχουμε λοιπόν δύο τρόπους για να εμφανίσουμε ένα string σε κάποιο σημείο της εφαρμογής μας. Ο πρώτος τρόπος είναι ο συνηθισμένος που εφαρμόζεται στα περισσότερα προγράμματα. Δηλαδή ορίζουμε στον κώδικα μας ένα string με την τιμή του, για παράδειγμα String helloWorld = “Hello World”. Έπειτα εμφανίζουμε το string αυτό κάνοντας αναφορά στη μεταβλητή μας, δηλαδή την helloWorld. Φανταστείτε, όμως, σε μία εφαρμογή που έχουμε χιλιάδες γραμμές κώδικα, να θέλουμε να μεταφράσουμε όλα αυτά τα strings, ψάχνοντάς τα ένα ένα. Τη λύση έρχεται να μας δώσει το Android, η οποία αποτελεί και το δεύτερο (και σωστό) τρόπο εμφάνισης των strings. Στο φάκελο res/values/ του project μας υπάρχει το αρχείο strings.xml. Στο αρχείο αυτό δηλώνουμε όλα τα string που χρησιμοποιεί η εφαρμογή μας με τον τρόπο <string name=”helloWorld”>Hello World</string>, και αναφερόμαστε σε αυτά μέσα στις κλάσεις μας με την εντολή getString(R.string.helloWorld). Με τον τρόπο αυτό ανεξαρτητοποιούμε τη γλώσσα της εφαρμογής μας, με τον κώδικα που γράφουμε στις activities.

Έστω, λοιπόν, ότι θέλουμε προεπιλεγμένη γλώσσα της εφαρμογής μας να είναι η αγγλική, άρα στο αρχείο res/values/strings.xml δηλώνουμε τα strings με το όνομά τους και την τιμή τους στην αγγλική γλώσσα. Για λόγους συμβατότητας, θέλουμε οι Έλληνες χρήστες να χρησιμοποιούν την εφαρμογή μας στην ελληνική γλώσσα. Δημιουργούμε, λοιπόν, νέο φάκελο res/values-el/ και μέσα στο φάκελο αυτόν δημιουργούμε το αρχείο strings.xml. Χρησιμοποιούμε ακριβώς το ίδιο όνομα για κάθε string αντίστοιχα με το προηγούμενο αρχείο strings.xml, αλλά τώρα με διαφορετική τιμή, δηλαδή στην ελληνική γλώσσα. Όταν ο χρήστης εγκαταστήσει και εκτελέσει την εφαρμογή, το Android θα εντοπίσει αυτόματα τη γλώσσα που χρησιμοποιεί ο χρήστης στη συσκευή του. Αν, λοιπόν, ο χρήστης χρησιμοποιεί την

ελληνική γλώσσα τότε το Android για κάθε string που γίνεται αναφορά στον κώδικα, θα χρησιμοποιεί αυτό που βρίσκεται στο αρχείο strings.xml στο φάκελο res/values-el/. Με τον ίδιο τρόπο μπορούμε να δημιουργήσουμε φακέλους και για άλλες γλώσσες όπως res/values-de/, res/values-fr/ για γερμανική και γαλλική γλώσσα. Σε περίπτωση που δε βρεθεί φάκελος για τη γλώσσα που χρησιμοποιεί ο χρήστης, τότε χρησιμοποιούνται οι τιμές που έχουμε στον προεπιλεγμένο φάκελο res/values/.

Εκτός από τη γλώσσα του κειμένου της εφαρμογής μας, τα παραπάνω μπορούν να εφαρμοστούν και σε κάποιο άλλο φάκελο των resources μας, όπως π.χ. το φάκελο που δηλώνουμε τη διεπαφή χρήστη, δηλαδή τον res/layout/. Για παράδειγμα η γερμανική γλώσσα συνήθως χρησιμοποιεί μεγαλύτερες σε μήκος λέξεις από την αγγλική. Επομένως, ορισμένα στοιχεία της εφαρμογής μας όπως κουμπιά ή κείμενο, μπορεί να εμφανίζονται με διαφορετικό τρόπο από αυτόν που θέλουμε, διότι τα strings που χρησιμοποιούμε είναι τώρα υπερβολικά μεγάλα. Επομένως, μπορούμε να δημιουργήσουμε νέο φάκελο res/layout-de/ και τα αρχεία xml (δηλαδή η διάταξη) που θα ορίσουμε μέσα σε αυτόν, θα χρησιμοποιούνται μόνο σε περίπτωση γερμανικής γλώσσας της συσκευής του χρήστη.

Στο παρακάτω παράδειγμα ορίζουμε στο αρχείο res/values/strings.xml το string <string name=”hello”>Hello!</string> και στο αρχείο res/values-el/strings.xml το string <string name=”hello”>Γεια σου!</string> (ίδιο όνομα, διαφορετική τιμή). Όταν το εμφανίσουμε στην οθόνη το αποτέλεσμα θα είναι το παρακάτω:

Εικόνα 3.13: Η εφαρμογή όπως φαίνεται σε συσκευή που χρησιμοποιεί οποιαδήποτε γλώσσα εκτός της ελληνικής (χρησιμοποιείται το προεπιλεγμένο αρχείο res/values/strings.xml)

Εικόνα 3.14: Η ίδια εφαρμογή όπως φαίνεται σε συσκευή που χρησιμοποιεί ελληνική γλώσσα (χρησιμοποιείται το αρχείο res/values-el/strings.xml)

Εκτός από τα παραπάνω (προσανατολισμός, μέγεθος οθόνης, γλώσσα) υπάρχουν και άλλες παράμετροι που δηλώνονται με τον ίδιο τρόπο και το λειτουργικό σύστημα τις εντοπίζει αυτόματα. Μερικές από αυτές είναι η πυκνότητα της οθόνης, η νυχτερινή λειτουργία, αν υπάρχει οθόνη αφής, αν υπάρχει φυσικό πληκτρολόγιο και η έκδοση του λειτουργικού συστήματος. Δηλαδή ανάλογα με τις

παραπάνω παραμέτρους της συσκευής του χρήστη, μπορεί να αλλάζει αυτόματα η διάταξη των γραφικών στοιχείων, η γλώσσα του κειμένου, οι εικόνες που χρησιμοποιούμε κλπ.

Τα παραπάνω μπορούν να λειτουργήσουν και συνδυαστικά. Για παράδειγμα αν δηλώσουμε το user interface μίας οθόνης στο φάκελο res/layout-el-port/, τότε αυτό θα χρησιμοποιηθεί αν η συσκευή του χρήστη χρησιμοποιεί την ελληνική γλώσσα και βρίσκεται σε προσανατολισμό πορτρέτου.

Το συμπέρασμα που προκύπτει για το σωστό προγραμματισμό της εφαρμογής μας είναι πάντα να εξωτερικεύουμε τα application resources από το κομμάτι του κώδικα των κλάσεων. Υπάρχουν κατάλληλοι φάκελοι που αποθηκεύουμε ή δηλώνουμε τα παραπάνω στοιχεία και πρέπει να χρησιμοποιούνται. Έτσι η εφαρμογή μας καθίσταται κατανοητή στον προγραμματιστή και εύκολα επεξεργάσιμη, ενώ παράλληλα γίνεται συμβατή με όλες τις δυνατές παραμέτρους της συσκευής του κάθε χρήστη.

3.6 Android Manifest

Σε κάθε εφαρμογή πρέπει να υπάρχει το αρχείο AndroidManifest.xml, το οποίο δημιουργείται αυτόματα όταν ξεκινάμε καινούργιο project μίας android application. Το αρχείο αυτό περιέχει βασικές πληροφορίες σχετικά με την εφαρμογή, τις οποίες το λειτουργικό σύστημα πρέπει να γνωρίζει προτού τρέξει οποιοδήποτε άλλο κώδικα. Οι σημαντικότερες από αυτές τις πληροφορίες περιγράφονται παρακάτω:

- Η ονομασία του πακέτου της Java της εφαρμογής (Java package).
- Η έκδοση της εφαρμογής (π.χ. 1.0, 1.4.2, 2.7 κλπ).
- Η ελάχιστη έκδοση του λειτουργικού συστήματος Android που απαιτεί η εφαρμογή (min sdk version). Για παράδειγμα αν έχει δηλωθεί min sdk version ο αριθμός 7, που ισοδυναμεί με την έκδοση Android 2.1, τότε η εφαρμογή θα μπορεί εκτελεστεί σε συσκευές με έκδοση Android μεγαλύτερη ή ίση της 2.1.
- Το όνομα της εφαρμογής, καθώς και το εικονίδιό της.
- Οι άδειες που απαιτούνται για να εκτελεστούν ορισμένες λειτουργίες της εφαρμογής. Για παράδειγμα αν η εφαρμογή μας χρησιμοποιεί την κάμερα της συσκευής θα πρέπει να δηλώνεται και αντίστοιχη άδεια. Αν θέλουμε να έχουμε πρόσβαση στο διαδίκτυο από την εφαρμογή μας, θα πρέπει να δηλώνεται η αντίστοιχη άδεια. Όμοια ισχύουν για το αν θέλουμε να αποθηκεύσουμε αρχεία στην κάρτα SD, αν θέλουμε να στείλουμε μηνύματα SMS, αν θέλουμε να πραγματοποιήσουμε τηλεφωνικές κλήσεις κλπ. Αν δε δηλώσουμε τις άδειες που απαιτούνται για τις διάφορες λειτουργίες της εφαρμογής, τότε θα εμφανίζεται σφάλμα και η εφαρμογή δε θα λειτουργεί. Οι άδειες αυτές περιγράφονται στο χρήστη τη στιγμή που εγκαθιστά την εφαρμογή, και πρέπει να συμφωνήσει με αυτές για να ολοκληρωθεί η εγκατάσταση. Με τον τρόπο αυτό, ο χρήστης αισθάνεται ασφαλής απέναντι σε κακόβουλες εφαρμογές. Για παράδειγμα αν ο χρήστης επιχειρήσει να εγκαταστήσει μία εφαρμογή άσχετη με τηλεφωνικές κλήσεις ή μηνύματα, και δει πριν την εγκατάσταση ότι αυτή ζητάει άδεια για τηλεφωνικές κλήσεις ή αποστολή μηνυμάτων, τότε θα καταλάβει ότι η εφαρμογή αυτή είναι πιθανότατα κακόβουλη και δεν πρέπει να εγκατασταθεί. Αν, βέβαια, οι συγκεκριμένες αυτές άδειες δεν αναγράφονται, τότε ο χρήστης είναι σίγουρος ότι η εφαρμογή δε θα μπορέσει με κανένα τρόπο να στείλει κάποιο

γραπτό μήνυμα ή να πραγματοποιήσει κάποια τηλεφωνική κλήση. Ακόμα και αν επιχειρούσε να το κάνει, η εφαρμογή θα εμφάνιζε σφάλμα τη στιγμή που επιχειρούσαμε να την τρέξουμε και δε θα λειτουργούσε.

- Όλα τα συστατικά στοιχεία (activities, services, content providers, broadcast receivers) της εφαρμογής. Για παράδειγμα αν δημιουργήσουμε κάποια κλάση για activity της εφαρμογής, χωρίς να τη δηλώσουμε στο AndroidManifest.xml, δε θα μπορέσει να λειτουργήσει.
- Οι εξωτερικές βιβλιοθήκες που χρησιμοποιεί η εφαρμογή μας. Για παράδειγμα αν η εφαρμογή μας, σε κάποιο σημείο του κώδικα τρέχει την εφαρμογή google maps, τότε θα πρέπει να δηλωθεί η βιβλιοθήκη com.google.android.maps.

Ακολουθεί ένα παράδειγμα του AndroidManifest.xml μίας εφαρμογής:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.test.testapplication"
    android:versionCode="1"
    android:versionName="1.0">

  <uses-sdk android:minSdkVersion="7" />

  <uses-permission android:name="android.permission.CAMERA" />
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

  <uses-feature android:name="android.hardware.camera" />

  <application android:icon="@drawable/icon" android:label="@string/app_name"
android:theme="@style/myStyle">

    <uses-library android:name="com.google.android.maps" />

    <activity android:name=".myFirstActivity" android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".mySecondActivity"></activity>
    <activity android:name=".myThirdActivity"></activity>

  </application>
</manifest>
```

3.7 Βέλτιστος Σχεδιασμός

Μπορεί μία εφαρμογή λειτουργικά να έχει επιτύχει το βασικό σκοπό για τον οποίο σχεδιάστηκε, ωστόσο υπάρχουν σχεδόν πάντα περιθώρια βελτίωσης, ώστε να γίνει γρηγορότερη, να απαιτεί λιγότερη μνήμη, να απευθύνεται σε ευρύτερο κοινό, χωρίς να χάνει κομμάτι από τη λειτουργικότητά της. Για το λόγο αυτό είναι πολύ σημαντικό όταν σχεδιάζουμε την εφαρμογή μας, να λαμβάνουμε υπόψη διάφορους παράγοντες, ώστε ο κώδικας να είναι βελτιστοποιημένος και η εφαρμογή να αξιοποιεί με το βέλτιστο τρόπο τις δυνατότητές της.

3.7.1 Δυνατότητα Πρόσβασης

Πολλοί χρήστες συσκευών Android έχουν κάποιο είδος αναπηρίας, το οποίο τους αναγκάζει να χρησιμοποιούν τη συσκευή με διαφορετικό τρόπο. Άτομα με προβλήματα όρασης ή σωματική αναπηρία ίσως να μη μπορούν να δουν ή να χρησιμοποιήσουν πλήρως μία οθόνη αφής. Το Android περιλαμβάνει μεθόδους που βοηθάνε τους χρήστες αυτούς να πλοηγούνται στις συσκευές ευκολότερα, όπως υπηρεσίες κείμενο-σε-ομιλία (text-to-speech) ή ιχνόσφαιρα (trackball) για πλοήγηση αντί για τα εικονικά κουμπιά. Αρκούν, λοιπόν, ορισμένες εύκολες ενέργειες από το μέρος του προγραμματιστή ώστε η εφαρμογή να γίνει προσβάσιμη σε όλους τους δυνατούς χρήστες.

- Πολλές συσκευές Android έχουν επιπλέον, clickable trackball (με δυνατότητα κλικ) ή clickable d-pad (σταυρό κατεύθυνσης) πέρα από τα βασικά κουμπιά. Ο χρήστης, με τον τρόπο αυτό, έχει τη δυνατότητα να πλοηγείται πλήρως μέσα σε μία εφαρμογή, χωρίς να χρησιμοποιεί την οθόνη αφής. Για άτομα λοιπόν με σωματική αναπηρία, ο τρόπος αυτός είναι ο πρωταρχικός για τη χρήση της συσκευής και την πλοήγηση στις εφαρμογές. Συνήθως δεν απαιτούνται επιπλέον ενέργειες από τον προγραμματιστή, ώστε η πλήρης χρήση της εφαρμογής να γίνεται και με trackball εκτός από την οθόνη αφής, ωστόσο πρέπει να το επιβεβαιώσουμε αυτό δοκιμάζοντας την εφαρμογή σε εικονική συσκευή που υποστηρίζει trackball και ελέγχοντας πλήρως όλες τις λειτουργίες της.
- Άτομα με προβλήματα όρασης ίσως δεν έχουν τη δυνατότητα να δουν τα γραφικά στοιχεία της κάθε οθόνης της εφαρμογής ώστε να καταλάβουν τη χρήση του καθενός. Η παράμετρος android:contentDescription που μας παρέχει το Android και μπορούμε να χρησιμοποιήσουμε σε όλα τα γραφικά στοιχεία τα οποία δηλώνουμε στα xml αρχεία, λύνει το παραπάνω πρόβλημα. Αν ο χρήστης έχει ενεργοποιήσει τα εργαλεία βοηθητικής πρόσβασης μέσω των ρυθμίσεων της συσκευής, όταν εστιάσει (focus) σε κάποιο στοιχείο θα ακούσει τη φράση την οποία έχουμε δηλώσει. Για παράδειγμα το παρακάτω πλαίσιο εισαγωγής κειμένου:

  ```
  <EditText
     android:id=”@+id/username”
     android:contentDescription=”@string/insert_username”/>
  ```

  όταν γίνει focused θα ακουστεί στο χρήστη η φράση (το string) που έχει δηλωθεί στο αρχείο strings.xml με το όνομα insert_username. Αν δηλαδή το string αυτό έχει τιμή “Insert username”, τότε η φράση αυτή θα ακουστεί στο χρήστη. Επομένως μπορούμε να προσθέσουμε την παράμετρο αυτή σε όλα τα γραφικά μας στοιχεία, ώστε ο χρήστης να καταλαβαίνει κάθε φορά σε ποιο γραφικό στοιχείο βρίσκεται και ποια θα πρέπει να είναι η επόμενη του ενέργεια.
- Το TalkBack είναι μία Android εφαρμογή η οποία είναι προεγκατεστημένη σε πολλές συσκευές Android ή μπορούμε να την κατεβάσουμε δωρεάν από το επίσημο website Android εφαρμογών, Android Market. Με το TalkBack ενεργοποιημένο, ο χρήστης μπορεί να ακούει περιγραφή της οθόνης στην οποία βρίσκεται, κάθε φορά που αλλάζει οθόνη. Συγκεκριμένα μέσα στις εφαρμογές ακούει όλα τα είδη κειμένου που περιέχει η κάθε οθόνη, ώστε να ξέρει κάθε φορά που βρίσκεται και τι μπορεί να κάνει. Με ενεργοποιημένο, λοιπόν, το TalkBack και σε συνδυασμό με τις βελτιστοποιήσεις που περιγράψαμε παραπάνω, μπορούμε να δοκιμάσουμε την εφαρμογή μας, χωρίς να βλέπουμε την οθόνη και να προσπαθούμε να χρησιμοποιήσουμε

πλήρως τις λειτουργίες της μόνο με το trackball και με βάση αυτά που ακούμε. Με τον τρόπο αυτό, θα γνωρίζουμε κάθε φορά την προσβασιμότητα της εφαρμογής μας και τις ενέργειες που πρέπει να κάνουμε για να βελτιώσουμε την εμπειρία του χρήστη.

- Τέλος επισημαίνεται, ότι υπάρχουν εφαρμογές Android, οι οποίες όταν ενεργοποιηθούν προσθέτουν τη δυνατότητα στο χρήστη, σε οποιαδήποτε εφαρμογή και αν βρίσκεται, να εισάγει κείμενο στα πλαίσια εισαγωγής κειμένου μόνο με την ομιλία του (speech-to-text), χωρίς δηλαδή να χρησιμοποιεί το φυσικό ή εικονικό πληκτρολόγιο. Διαφορετικά, μπορούμε να προσθέσουμε εμείς ενσωματωμένη υπηρεσία speech-to-text στην εφαρμογή μας για κάθε πλαίσιο εισαγωγής κειμένου (Εικόνα 3.15).

Εικόνα 3.15: Δυνατότητα φωνητικής εισαγωγής κειμένου

3.7.2 Επίδοση

Μία εφαρμογή Android θα τρέχει σε μία κινητή συσκευή με περιορισμένη υπολογιστική ισχύ, διάρκεια ζωής μπαταρίας και περιορισμένο αποθηκευτικό χώρο. Για το λόγο αυτό, η εφαρμογή μας πρέπει να είναι αποδοτική. Ακόμα και όταν βλέπουμε την εφαρμογή μας να τρέχει αρκετά γρήγορα, ίσως υπάρχουν περιθώρια βελτίωσης όσον αφορά στη χρήση της μπαταρίας. Η διάρκεια ζωής της μπαταρίας είναι σημαντική στους χρήστες Android και μάλιστα έχουν τη δυνατότητα μέσω του λειτουργικού συστήματος, να ελέγξουν την κατανάλωση μπαταρίας που προκαλεί κάθε εφαρμογή. Πρέπει, λοιπόν, σε γενικές γραμμές να ακολουθάμε τους παρακάτω κανόνες:

- Να αποφεύγουμε τη δημιουργία περιττών αντικειμένων (objects). Ειδικότερα, όταν στον κώδικά μας δημιουργούμε ένα object μέσα σε ένα βρόχο (loop), θα πρέπει να γνωρίζουμε ότι έτσι θα έχουμε μία περιοδική συλλογή σκουπιδιών (garbage collection), η οποία θα προκαλεί ορισμένες καθυστερήσεις στην εκτέλεση της εφαρμογής. Με το ίδιο σκεπτικό, σχεδόν πάντα είναι αποδοτικότερο, να χρησιμοποιούμε δύο παράλληλους μονοδιάστατους πίνακες αντί για ένα δισδιάστατο. Γενικά, λοιπόν, πρέπει να

γνωρίζουμε ότι λιγότερα αντικείμενα σημαίνουν και λιγότερες φορές συλλογή σκουπιδιών, πράγμα το οποίο έχει αντίκτυπο στην εμπειρία των χρηστών.

- Αν δε χρειάζεται να έχουμε πρόσβαση στα πεδία ενός αντικειμένου, τότε προτιμάμε να δηλώνουμε τις μεθόδους μας static. Το γεγονός αυτό μας δίνει ταχύτερη απόκριση περίπου 15%-20%.
- Να δηλώνουμε τις σταθερές μας ως static final, όπου αυτό είναι εφικτό. Για παράδειγμα αν θέλουμε να δηλώσουμε τη σταθερά "Hello, world!", ο πιο αποδοτικός τρόπος θα ήταν: static final String strVal = "Hello, world!";
- Για κάθε βρόχο που χρησιμοποιούμε στον κώδικά μας, το αποδοτικότερο είναι να κάνουμε χρήση της τεχνικής του ενισχυμένου βρόχου (enhanced for loop) της Java. Το παράδειγμα που ακολουθεί επεξηγεί αυτήν την τεχνική:

```
static class Foo {
        int mSplat;
}
Foo[] mArray = ...

public void zero() {
int sum = 0;
for (int i = 0; i < mArray.length; ++i) {
        sum += mArray[i].mSplat;
}
}

public void one() {
int sum = 0;
Foo[] localArray = mArray;
int len = localArray.length;

for (int i = 0; i < len; ++i) {
        sum += localArray[i].mSplat;
}
}

public void two() {
int sum = 0;
for (Foo a : mArray) {
        sum += a.mSplat;
}
}
```

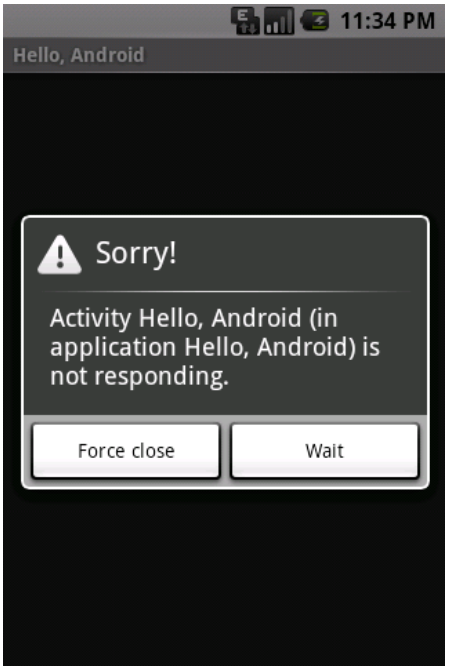
Η τεχνική zero() είναι η πιο αργή, διότι το μήκος του πίνακα υπολογίζεται σε κάθε επανάληψη.
Η τεχνική one() είναι γρηγορότερη, διότι πρώτον χρησιμοποιεί τοπικές μεταβλητές τις οποίες προσπελαύνει γρηγορότερα και δεύτερον το μήκος του πίνακα υπολογίζεται μόνο μία φορά.
Η τεχνική two() (enhanced for loop) είναι γρηγορότερη από την one() στις περισσότερες συσκευές. Στη χειρότερη περίπτωση όμως θα είναι σχεδόν ίδιου υπολογιστικού κόστους με την one(), οπότε τελικά προτιμάται αυτή.

3.7.3 Ανταπόκριση

Όσο βελτιστοποιημένος και αν είναι ο κώδικας της εφαρμογής, δε θα έχει σημασία αν τελικά αυτή δεν ανταποκρίνεται σε κάποιο σημείο. Όταν συμβεί κάτι

τέτοιο, η εφαρμογή μας εμφανίζει διάλογο ότι δεν ανταποκρίνεται και ο χρήστης έχει την επιλογή να τερματίσει την εφαρμογή ή να περιμένει (Εικόνα 3.16). Ο διάλογος αυτός σίγουρα δεν ευχαριστεί τους χρήστες, γι' αυτό το λόγο πρέπει προσέχουμε στο σχεδιασμό ώστε ποτέ να μη δημιουργείται τέτοιο πρόβλημα. Γενικά το σύστημα εμφανίζει τέτοιο διάλογο, όταν η εφαρμογή δεν ανταποκρίνεται σε ενέργειες του χρήστη, δηλαδή όταν έχει μπλοκάρει σε κάποια εντολή (π.χ. όταν η εντολή αυτή προκαλεί exception, η οποία δεν πιάνεται). Ομοίως, αν μία εφαρμογή σε κάποιο σημείο εκτελεί πολλούς και χρονοβόρους υπολογισμούς, τότε το σύστημα συμπεραίνει ότι η εφαρμογή έχει κολλήσει και τελικά εμφανίζεται αυτός ο διάλογος.
Για να μη δημιουργείται ποτέ σε μία εφαρμογή το παραπάνω πρόβλημα, θα πρέπει να ακολουθάμε στο σχεδιασμό τα παρακάτω:

- Αρχικά θα πρέπει να ελέγξουμε αν ο κώδικάς μας είναι σωστός, και όπου υπάρχει περίπτωση να προκληθεί exception, να τη χειριζόμαστε κατάλληλα.
- Οι εφαρμογές Android τρέχουν σε μία (την κύρια) διεργασία (main thread), η οποία είναι υπεύθυνη και για τη διεπαφή χρήστη. Αυτό σημαίνει ότι η κύρια διεργασία θα πρέπει να εκτελεί όσο γίνεται λιγότερες ενέργειες. Επομένως όλες οι χρονοβόρες διαδικασίες, όπως λειτουργίες που έχουν σχέση με το διαδίκτυο ή βάσεις δεδομένων ή κάποια άλλη χρονοβόρα λειτουργία, θα πρέπει να εκτελούνται σε διαφορετική διεργασία ή σε μία ειδική ασύγχρονη εργασία που περιλαμβάνει το Android και ονομάζεται AsyncTask. Σε αυτές τις περιπτώσεις, για να δείξουμε στο χρήστη ότι η εφαρμογή είναι απασχολημένη και εκτελεί κάποια λειτουργία, θα πρέπει στο χρονικό διάστημα που εκτελείται η λειτουργία αυτή, να εμφανίζουμε έναν ProgressDialog (π.χ. με κάποιο μήνυμα "Loading…").
- Αν η εφαρμογή μας χρειάζεται αρκετό χρόνο για τις αρχικές λειτουργίες, τη στιγμή που επιλέγουμε να την τρέξουμε, τότε θα ήταν καλύτερο να εμφανίζουμε στο χρήστη μία αρχική οθόνη (splash screen) για αυτό το χρονικό διάστημα.
- Δεν πρέπει να ξεχνάμε ότι το Android είναι λειτουργικό σύστημα για κινητές συσκευές. Αυτό σημαίνει ότι οποιαδήποτε στιγμή, μπορεί κάποια activity κάποιας άλλης εφαρμογής (όπως της εφαρμογής "Εισερχόμενη τηλεφωνική κλήση") να ενεργοποιηθεί και η εφαρμογή μας να περάσει στο background. Το γεγονός αυτό ενεργοποιεί τις μεθόδους onSaveInstanceState() και onPause() της activity που βρισκόταν ο χρήστης πριν την τηλεφωνική κλήση. Αν και τις περισσότερες φορές το γεγονός αυτό δε θα προκαλέσει κάποιο σφάλμα στην εφαρμογή μας, ωστόσο θα πρέπει να το δοκιμάσουμε και αν προκύψει κάποιο σφάλμα, να το εντοπίσουμε και να το λύσουμε κυρίως με την κατάλληλη υλοποίηση των παραπάνω μεθόδων.



Εικόνα 3.16: Ο διάλογος που εμφανίζεται στο χρήστη, όταν μία εφαρμογή δεν ανταποκρίνεται

4

Αρχιτεκτονική της εφαρμογής

4.1 Εισαγωγή

Στο κεφάλαιο που ακολουθεί αρχικά θα περιγράψουμε την εφαρμογή που αναπτύχθηκε για το λειτουργικό σύστημα Android, αναφέροντας όλες τις δυνατότητες που προσφέρει η εφαρμογή στο χρήστη. Έπειτα θα αναλύσουμε τη δομή της, τα αρχεία δηλαδή στα οποία οργανώνεται, καθώς και τη χρήση του καθενός.

Η ιδέα για την κατασκευή της συγκεκριμένης εφαρμογής προέρχεται από τη μηχανή αναζήτησης εικόνων, με βάση το περιεχόμενό τους, VIRaL 2.0, η οποία φιλοξενείται σε εξυπηρετητή του Εργαστηρίου Εικόνας, Βίντεο και Συστημάτων Πολυμέσων του Εθνικού Μετσόβιου Πολυτεχνείου. Η εφαρμογή αυτή ονομάστηκε "VIRaL" και χρησιμοποιεί στη λειτουργία της την παραπάνω υπηρεσία.

4.2 Περιγραφή

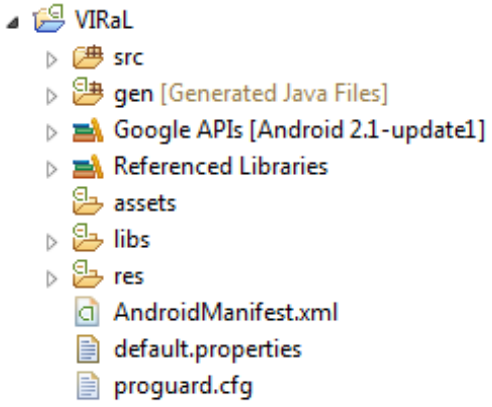
Κύριος σκοπός της εφαρμογής είναι η άντληση πληροφοριών σχετικά με το περιεχόμενο κάποιας εικόνας την οποία επιλέγει ο χρήστης. Η επιλογή της εικόνας αυτής μπορεί να γίνει με δύο διαφορετικούς τρόπους. Ο χρήστης μπορεί είτε να τραβήξει μία φωτογραφία μέσω της κάμερας του κινητού τηλεφώνου, είτε να επιλέξει μία εικόνα που βρίσκεται ήδη αποθηκευμένη στην συσκευή του σε κάποιο φάκελο. Η εικόνα αυτή, αποστέλλεται στον εξυπηρετητή που αναφέρθηκε παραπάνω, ο οποίος είναι υπεύθυνος για τη σύγκριση αυτής με άλλες ήδη γνωστές, ώστε να προκύψουν ορισμένες πληροφορίες σχετικά με το περιεχόμενο της εικόνας αυτής και να αποσταλούν πίσω στην εφαρμογή. Οι πληροφορίες αυτές περιέχουν τη φυσική τοποθεσία του περιεχομένου της εικόνας, την ονομασία του αν πρόκειται για κάποιο τοπωνύμιο ή μνημείο, καθώς και παρόμοιες εικόνες με τη συγκεκριμένη. Η

εφαρμογή επεξεργάζεται τις πληροφορίες αυτές και τελικά εμφανίζει στο χρήστη την πόλη και χώρα που βρίσκεται το περιεχόμενο της εικόνας, την ονομασία του, έναν δυναμικό χάρτη στο σημείο αυτό στον οποίο μπορεί να περιηγηθεί ο χρήστης, και παρόμοιες εικόνες βάσει περιεχομένου. Ο χρήστης μάλιστα μπορεί να αποθηκεύσει την εικόνα αυτή και να ενσωματώσει σε αυτή όλες τις παραπάνω πληροφορίες σαν μεταδεδομένα (metadata).

Παράλληλα ο χρήστης μπορεί να δημιουργήσει κάποιον λογαριασμό στον οποίο να αποθηκεύονται αυτόματα όλες οι εικόνες που στέλνει, μαζί με τα αποτελέσματά τους. Έτσι όταν ο χρήστης είναι συνδεδεμένος στο λογαριασμό του, οι εικόνες που έχει στείλει μπορούν να εμφανιστούν σαν ιστορικό στην αντίστοιχη οθόνη της εφαρμογής, είτε σε μία λίστα, είτε πάνω σε δυναμικό χάρτη, με την κάθε εικόνα να εμφανίζεται στο σημείο του χάρτη στο οποίο βρίσκεται το φυσικό περιεχόμενό της.

4.3 Δομή

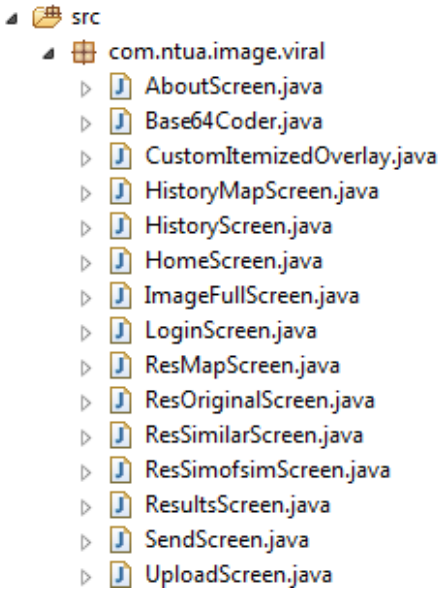
Πριν προχωρήσουμε στη λεπτομερή ανάλυση των διάφορων λειτουργιών της εφαρμογής που αναπτύχθηκε, θα ήταν χρήσιμο να παρουσιάσουμε πρώτα τη δομή της, πώς δηλαδή οργανώνονται όλα τα αρχεία που περιέχονται σε αυτήν και ποια είναι η κύρια λειτουργία τους. Στο προγραμματιστικό περιβάλλον Eclipse στο οποίο αναπτύχθηκε η εφαρμογή, το project εμφανίζεται με την παρακάτω δομή:



Εικόνα 4.1: Δομή της εφαρμογής

Ακολουθεί η επεξήγηση των παραπάνω φακέλων και αρχείων καθώς και του περιεχομένου τους:

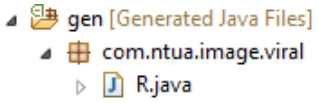
- **src/**



Εικόνα 4.2: Περιεχόμενα φακέλου src/

Ο φάκελος αυτός αρχικά περιέχει τα πακέτα τα οποία έχουμε δηλώσει στην εφαρμογή μας. Έχουμε δηλώσει ένα μόνο πακέτο, το "com.ntua.image.viral" οπότε εμφανίζεται αυτό. Έπειτα στο πακέτο αυτό περιέχονται όλες οι κλάσεις της εφαρμογής μας. Οι κλάσεις αυτές χωρίζονται στις activities, δηλαδή στις διαφορετικές οθόνες της εφαρμογής μας, στις οποίες έχουμε συμπεριλάβει στην ονομασία τους τη λέξη Screen για να ξεχωρίζουν, και στις υπόλοιπες οι οποίες εκτελούν κάποια εξειδικευμένη λειτουργία.

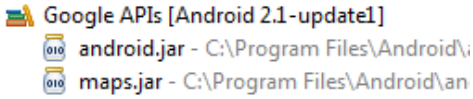
- **gen/**



Εικόνα 4.3: Περιεχόμενα φακέλου gen/

Ο φάκελος αυτός περιέχει για το πακέτο που χρησιμοποιούμε στην εφαρμογή μας, το αρχείο R.java το οποίο ουσιαστικά είναι μία κλάση η οποία κατασκευάζεται αυτόματα από το σύστημα. Η κλάση αυτή συνδέει όλα τα application resources με τον κώδικα της εφαρμογής μας. Έχουμε εξηγήσει ότι όλα αυτά τα resources δηλώνονται σε ξεχωριστά αρχεία και όχι μέσα στις κλάσεις της εφαρμογής μας. Η κλάση, λοιπόν, R.java δηλώνει κάθε στοιχείο από τις εικόνες, τα γραφικά στοιχεία, τα strings, τα styles ή themes, που χρησιμοποιούμε στην εφαρμογή μας, σαν μία ακέραια σταθερά (int) και με τιμή έναν μοναδικό δεκαεξαδικό αριθμό, ώστε να αναφερόμαστε σε αυτές μέσα από τις δικές μας κλάσεις σαν δημόσιες σταθερές (public constants).

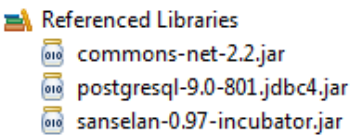
- **Google APIs**



Εικόνα 4.4: Βιβλιοθήκες του λειτουργικού συστήματος Android

Πρόκειται για τις βιβλιοθήκες που επιλέγονται αυτόματα από το σύστημα, όταν εμείς δηλώσουμε την ελάχιστη έκδοση Android με την οποία θα είναι συμβατή η εφαρμογή μας. Συγκεκριμένα, όταν ξεκινάμε ένα project για την κατασκευή Android application τότε ανάμεσα στις διάφορες επιλογές, πρέπει να επιλέξουμε και το Build Target της εφαρμογής μας, που σημαίνει την ελάχιστη έκδοση Android με την οποία θέλουμε να είναι συμβατή η εφαρμογή μας. Έτσι το σύστημα, τοποθετεί τις κατάλληλες βιβλιοθήκες για να λειτουργεί η εφαρμογή μας στις εκδόσεις αυτές (αρκεί να έχουμε εγκαταστήσει τις εκδόσεις αυτές μέσω της ADT Plugin). Προφανώς, το ιδανικό θα ήταν η εφαρμογή μας να είναι συμβατή με όλες τις εκδόσεις Android, όμως κάτι τέτοιο δεν είναι πάντα εφικτό, διότι κάθε νέα έκδοση Android έχει τις ίδιες και ορισμένες επιπλέον δυνατότητες από την αμέσως προηγούμενη. Επομένως, όταν στην εφαρμογή μας θέλουμε να χρησιμοποιήσουμε μία εξειδικευμένη λειτουργία του συστήματος, τότε πιθανόν η λειτουργία αυτή να μην υπάρχει για χαμηλές εκδόσεις Android, πράγμα το οποίο θα αποκλείσει την εφαρμογή μας από τις εκδόσεις αυτές. Διαισθητικά μπορούμε να σκεφτούμε ότι γενικά οι απλές εφαρμογές είναι συμβατές για όλες τις εκδόσεις Android, οι λίγο πιο εξειδικευμένες είναι συμβατές για εκδόσεις μεγαλύτερες ή ίσες της 2.1, ενώ άλλες ακόμα πιο εξειδικευμένες είναι συμβατές για εκδόσεις μεγαλύτερες ή ίσες της 2.2. Συγκεκριμένα η εφαρμογή μας είναι συμβατή για εκδόσεις Android μεγαλύτερες ή ίσες της 2.1, διότι χρησιμοποιεί ορισμένα χαρακτηριστικά του συστήματος, τα οποία δεν διατίθενται σε χαμηλότερες εκδόσεις. Βέβαια αξίζει να σημειώσουμε ότι τη στιγμή αυτή, το ποσοστό των χρηστών Android, οι οποίοι χρησιμοποιούν συσκευή με λειτουργικό σύστημα έκδοσης μικρότερης της 2.1 (δηλαδή 1.5 ή 1.6) είναι μόλις 3% και μάλιστα το ποσοστό αυτό συνεχώς μειώνεται, είτε γιατί οι συσκευές αυτές εγκαταλείπονται, είτε γιατί αναβαθμίζονται σε μεγαλύτερη έκδοση. Το αρχείο, λοιπόν, android.jar αποτελεί τη βασική βιβλιοθήκη του λειτουργικού συστήματος, ενώ το αρχείο maps.jar αποτελεί τη βιβλιοθήκη που χρειάζεται ώστε να λειτουργεί ο δυναμικός χάρτης της Google στην εφαρμογή μας. Τα δύο αυτά αρχεία προστέθηκαν αυτόματα όταν επιλέξαμε Build Target την έκδοση Android 2.1 που επιπλέον χρησιμοποιεί όμως τους χάρτες της Google.

- **Referenced Libraries**



Εικόνα 4.5: Επιπλέον βιβλιοθήκες που χρησιμοποιεί η εφαρμογή

Οι βιβλιοθήκες αυτές προστίθενται από τον προγραμματιστή για τη χρήση ορισμένων λειτουργιών, οι οποίες δεν είναι ενσωματωμένες στις βιβλιοθήκες ούτε του λειτουργικού συστήματος Android, ούτε της έκδοσης του JDK (Java Development Kit) που χρησιμοποιούμε. Η βιβλιοθήκη commons-net-2.2.jar μας παρέχει τρόπους για να γίνει σύνδεση ftp από την εφαρμογή μας στον εξυπηρετητή του VIRaL 2.0. Η βιβλιοθήκη postgresql-9.0-801.jdbc4.jar χρησιμοποιείται για να γίνει σύνδεση από την εφαρμογή μας στη βάση δεδομένων (PostgreSQL) στην οποία αποθηκεύεται το ιστορικό του χρήστη. Τέλος, η βιβλιοθήκη sanselan-0.97-incubator.jar χρησιμοποιείται για να ενσωματωθούν στη φωτογραφία που έχουμε τραβήξει πληροφορίες όπως πόλη, χώρα, γεωγραφικές συντεταγμένες και ονομασία τοπωνυμίου. Υπάρχει, ωστόσο, ενσωματωμένη βιβλιοθήκη στο Android, η οποία χρησιμοποιείται για παρόμοια λειτουργία, όμως έχει πολύ λιγότερες δυνατότητες από τη sanselan και για αυτό το λόγο δε χρησιμοποιήθηκε.

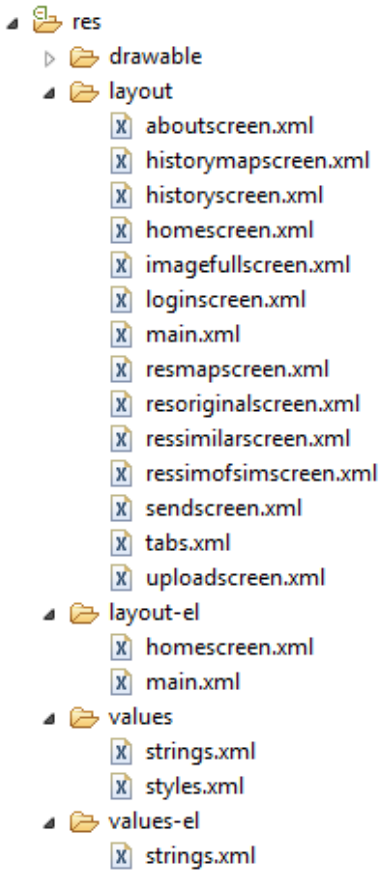
- **assets/**

  Στο φάκελο αυτό μπορούμε να αποθηκεύσουμε οποιοδήποτε αρχείο θέλουμε ή και να δημιουργήσουμε υποφακέλους. Ωστόσο όλα τα application resources αποθηκεύονται σε διαφορετικό φάκελο, πράγμα που καθιστά τον φάκελο αυτό όχι τόσο χρήσιμο όσο τους άλλους. Μπορεί, βέβαια, να αποθηκευτούν εκεί κάποια αρχεία τα οποία δεν χρησιμοποιούνται αυτή τη στιγμή στην εφαρμογή και θέλει για παράδειγμα ο προγραμματιστής να τα χρησιμοποιήσει στο μέλλον. Στις περισσότερες εφαρμογές, λοιπόν, ο φάκελος αυτός παραμένει κενός.

- **libs/**

  Ο φάκελος αυτός αποτελεί τη φυσική τοποθεσία των βιβλιοθηκών “Referenced Libraries” που εξηγήθηκαν παραπάνω. Δηλαδή ο προγραμματιστής τοποθετεί στο φάκελο αυτό τα αρχεία .jar τα οποία θέλει να χρησιμοποιήσει, και έπειτα τα συνδέει με το λειτουργικό σύστημα Android. Τα συνδεδεμένα αρχεία, τα οποία τελικά χρησιμοποιούνται, αποτελούν τις Referenced Libraries.

- **res/**



Εικόνα 4.6: Όλα τα resources της εφαρμογής

Στο φάκελο res/ αποθηκεύονται όλα τα application resources, τα οποία χρησιμοποιούμε. Συγκεκριμένα, στο φάκελο drawable αποθηκεύουμε όλα τα αρχεία εικόνας που χρησιμοποιούνται στην εφαρμογή. Στο φάκελο layout αποθηκεύονται τα αρχεία xml στα οποία δηλώνουμε το user interface της κάθε οθόνης. Έχουμε δημιουργήσει έναν επιπλέον φάκελο layout-el διότι όταν η εφαρμογή χρησιμοποιείται στην ελληνική γλώσσα, τα κουμπιά στο μενού θα πρέπει να είναι μεγαλύτερα, ώστε να χωρέσουν οι ελληνικές λέξεις, οι οποίες είναι μεγαλύτερες από τις αντίστοιχες αγγλικές. Τα κουμπιά αυτά τα δηλώνουμε στο main.xml καθώς και στο homescreen.xml. Ο φάκελος values περιλαμβάνει το αρχείο strings.xml με όλα τα strings της εφαρμογής δηλωμένα εκεί, και το αρχείο styles.xml, στο οποίο δηλώνονται τα styles τα οποία θέλουμε να χρησιμοποιήσουμε είτε σε όλη την εφαρμογή, είτε σε συγκεκριμένη οθόνη, είτε σε συγκεκριμένο γραφικό στοιχείο. Στο φάκελο values-el περιέχεται το αρχείο strings.xml, το οποίο περιλαμβάνει όλα τα strings δηλωμένα με το ίδιο όνομα, και τιμή ουσιαστικά την ελληνική τους μετάφραση.

Τέλος, παρατηρούμε τρία αρχεία τα οποία δημιουργούνται αυτόματα όταν ξεκινάμε καινούργιο project για μία εφαρμογή Android. Το αρχείο AndroidManifest.xml έχει αναλυθεί σε προηγούμενη ενότητα. Τα άλλα δύο αρχεία αφορούν το σύστημα και δε θα πρέπει να τα επεξεργαζόμαστε.

5

Βασικές λειτουργίες και σενάριο χρήσης εφαρμογής

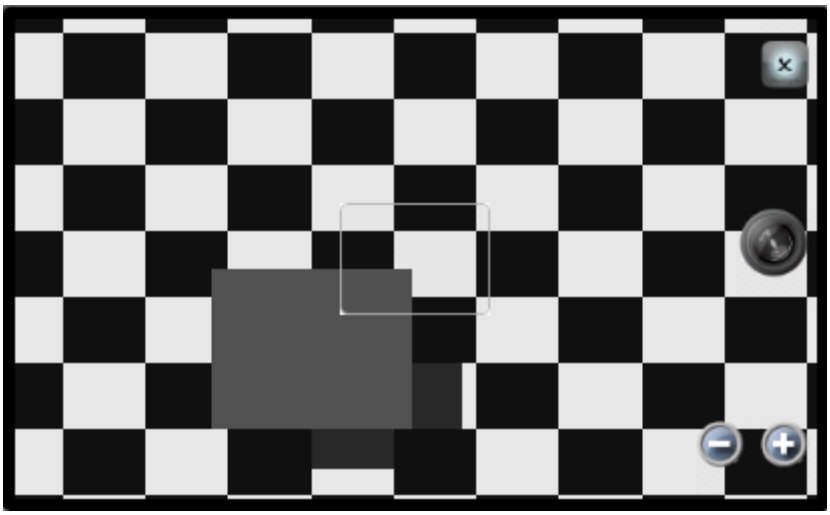
5.1 Εισαγωγή

Αφού παρουσιάστηκε η δομή της εφαρμογής, μπορούμε τώρα να παρουσιάσουμε αναλυτικά όλες τις λειτουργίες που περιλαμβάνει η εφαρμογή που αναπτύχθηκε, καθώς και τον τρόπο υλοποίησης αυτών στο προγραμματιστικό περιβάλλον. Στο κεφάλαιο αυτό, λοιπόν, θα περιγραφούν όλες οι διαφορετικές οθόνες της εφαρμογής, ο τρόπος σύνδεσης μεταξύ τους, οι κλάσεις στις οποίες υλοποιούνται, και όλες οι δυνατές επιλογές που έχει ο χρήστης όταν βρίσκεται σε συγκεκριμένη οθόνη. Τα διάφορα στιγμιότυπα της εφαρμογής παρουσιάζονται σε εικόνες και έπειτα εξηγούνται οι διάφορες λειτουργίες της, για να διευκολύνεται ο αναγνώστης στην κατανόηση των λειτουργιών αυτών.

Για διευκόλυνση της αναγνωσιμότητας του κειμένου, στο κεφάλαιο αυτό θα παρουσιαστούν κομμάτια κώδικα από τα σημαντικότερα αρχεία της εφαρμογής, ενώ ολοκληρωμένα τα αρχεία αυτά παρατίθενται στο παράρτημα της παρούσας εργασίας. Παρατίθενται, επομένως, τα κομμάτια κώδικα που αφορούν βασικές λειτουργίες της κάθε οθόνης, ενώ για πιο εξειδικευμένες λειτουργίες, προτείνεται στον αναγνώστη να μελετήσει τον ολοκληρωμένο κώδικα της αντίστοιχης κλάσης, που παρουσιάζεται στο παράρτημα.

5.2 Αρχική Οθόνη

Όταν η εφαρμογή εκτελεστεί, ενεργοποιείται η κάμερα της συσκευής, μέσω της οποίας μπορούμε να τραβήξουμε κάποια φωτογραφία. Η οθόνη αυτή φαίνεται στην παρακάτω εικόνα σε landscape mode:



Εικόνα 5.1: Αρχική οθόνη της εφαρμογής

Πρόκειται για την κλάση HomeScreen.java την οποία έχουμε ορίσει σαν main activity στο AndroidManifest.xml ως εξής:

```
<activity android:name=".HomeScreen" android:theme="@style/myHomeStyle"
android:screenOrientation="landscape" android:configChanges="keyboardHidden|orientation"
          android:label="@string/app_name">
        <intent-filter>
          <action android:name="android.intent.action.MAIN" />
          <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
</activity>
```

Από τις παραπάνω παραμέτρους παρατηρούμε ότι χρησιμοποιείται το myHomeStyle ως theme της activity αυτής, στο οποίο έχουμε απλά ορίσει την παράμετρο windowFullscreen ως true, ώστε να τρέχει η activity αυτή σε full screen. Αυτό γίνεται, διότι, αν δεν τρέξει σε full screen, τότε εμφανίζεται και η status bar της συσκευής με αποτέλεσμα να παραμορφώνεται η εικόνα που βλέπουμε στην κάμερα. Ακόμη, έχουμε ορίσει η οθόνη αυτή να τρέχει μόνο σε landscape mode, διότι όταν αλλάζουμε προσανατολισμό στη συσκευή, δε θέλουμε να αλλάζει η εικόνα που φαίνεται στην οθόνη.

Όταν ξεκινάει η activity δηλώνουμε πρώτα τη διεπαφή χρήστη της οθόνης αυτής με την εντολή setContentView(R.layout.homescreen); Στο αρχείο homescreen.xml έχουμε δηλώσει όλο το γραφικό περιβάλλον της οθόνης αυτής (εικόνα κάμερας, κουμπιά). Έχοντας δηλώσει τη μεταβλητή της κάμερας ως public Camera mCam; Πρέπει έπειτα να ορίσουμε τέσσερις βασικές μεθόδους για να υλοποιηθεί η λειτουργία της κάμερας:

```
public void surfaceCreated(SurfaceHolder holder) {
                // Surface created, now it is possible to set the preview
                try {
                        mCam = Camera.open();
                        mCam.setPreviewDisplay(holder);
                } catch (IOException e) {
                        e.printStackTrace();
                }
}
```

```
	public void surfaceDestroyed(SurfaceHolder holder) {
			mCam.stopPreview();
			mCam.release();
	}

	public void surfaceChanged(SurfaceHolder holder, int I, int J, int K) {
			Camera.Parameters parameters = mCam.getParameters();

			List<Size> previewSizes;
			if (Build.VERSION.SDK_INT > Build.VERSION_CODES.ECLAIR_MR1)
{
				//android 2.2 and over
				previewSizes = parameters.getSupportedPreviewSizes();
			} else {
				//android 2.1 and under
				String supportedSizesString = parameters.get("preview-size-
values");
				previewSizes = new ArrayList<Size>();
				if (supportedSizesString != null &&
supportedSizesString.length() > 0) {
				   StringSplitter ss = new TextUtils.SimpleStringSplitter(',');
				   ss.setString(supportedSizesString);
				   for (String supportedSize : ss) {
				      String[] dimensions = supportedSize.split("x");
				      if (dimensions.length > 1) {
				         Size size = mCam.new
Size(Integer.parseInt(dimensions[0]),
				            Integer.parseInt(dimensions[1]));
				         previewSizes.add(size);
				      }
				   }
				}
			}
			Size bestSize = null;
			int bestDiff = 0;
			int diff = 0;
			for (Size size : previewSizes) {
				diff = Math.abs(K - size.height) + Math.abs(J -
size.width);
				if (bestSize == null || diff < bestDiff) {
					bestSize = size;
					bestDiff = diff;
				}
				parameters.setPreviewSize(bestSize.width,
bestSize.height);
				mCam.setParameters(parameters);
			}

			//start preview of camera
			mCam.startPreview();
	}

	//Kaleitai otan paroume fwtografia
	PictureCallback jpegCallback = new PictureCallback() {
			public void onPictureTaken(byte[] data, Camera camera) {

				Intent i = new Intent (HomeScreen.this, SendScreen.class);
				i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
				i.putExtra("dataHome", data);
				startActivity (i);
```

```
                    overridePendingTransition(0,0);
                    isPressed=false;
                }
        };
```

Η surfaceCreated καλείται αρχικά για να ενεργοποιηθεί η κάμερα, η surfaceDestroyed καλείται όταν αλλάξουμε activity ή βγούμε από την εφαρμογή ώστε να απελευθερωθεί η κάμερα, ενώ η surfaceChanged ρυθμίζει παραμέτρους της κάμερας όπως την ανάλυση στην οποία θα λειτουργήσει η κάμερα. Μέσω της εντολής previewSizes = parameters.getSupportedPreviewSizes(); βρίσκουμε όλες τις δυνατές αναλύσεις στις οποίες μπορεί να λειτουργήσει η κάμερα και έπειτα επιλέγουμε την καλύτερη δυνατή. Η εντολή αυτή όμως υποστηρίζεται μόνο για έκδοση Android 2.2 και πάνω, γι' αυτό το λόγο αν η συσκευή του χρήστη χρησιμοποιεί έκδοση 2.1, οι δυνατές αναλύσεις βρίσκονται με άλλο τρόπο. Η μέθοδος jpegCallback καλείται όταν ο χρήστης τραβήξει φωτογραφία. Την ορίσαμε να ξεκινάει την activity SendScreen και να στέλνει σε αυτήν τον πίνακα byte[] data, ο οποίος αποτελεί τη φωτογραφία που τράβηξε ο χρήστης σε μορφή binary.

Ο χρήστης έχει τη δυνατότητα να τραβήξει φωτογραφία με τρεις διαφορετικούς τρόπους. Ο πρώτος τρόπος είναι το εικονικό κουμπί που βρίσκεται στη μέση και δεξιά, το οποίο ενεργοποιείται μόλις το αγγίξει ο χρήστης. Ο δεύτερος τρόπος είναι μέσω του trackball της συσκευής (φυσικό κουμπί), το οποίο όμως δεν υπάρχει σε όλες τις συσκευές Android, και για αυτό το λόγο πρέπει οπωσδήποτε να υπάρχει και το εικονικό. Ο τρίτος τρόπος είναι το κουμπί Search του κινητού τηλεφώνου. Αυτό έγινε, πρώτον διότι το κουμπί Search δε μας χρειάζεται τη στιγμή αυτή για κάποια λειτουργία, οπότε θα ήταν καλό να το εκμεταλλευτούμε και δεύτερον, η θέση που βρίσκεται το κουμπί αυτό στα περισσότερα κινητά, ίσως βολεύει το χρήστη περισσότερο από ότι το εικονικό κουμπί. Το κουμπί Search, να σημειώσουμε, ότι περιλαμβάνεται σε όλες τις συσκευές Android. Για παράδειγμα, όταν πατάμε το εικονικό κουμπί για λήψη φωτογραφίας καλείται η παρακάτω μέθοδος:

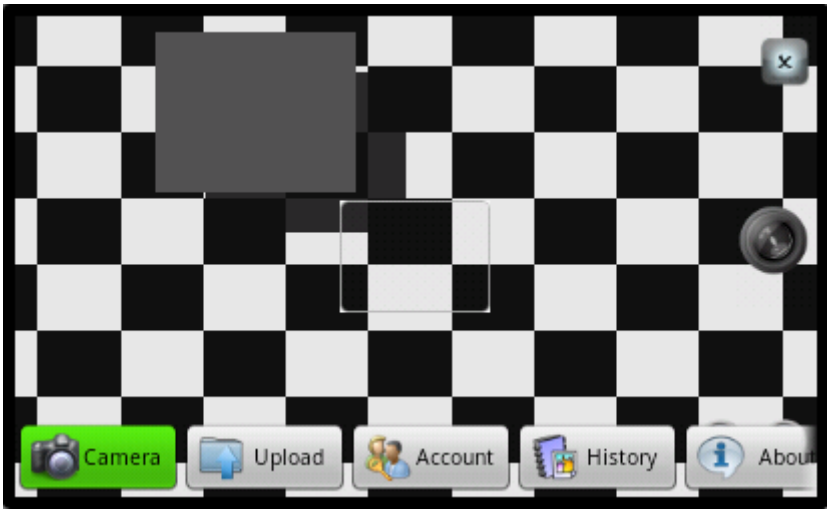
```
buttonClick = (Button) findViewById(R.id.buttonClick);
                buttonClick.setOnClickListener(new OnClickListener() {
                        public void onClick(View v) {
                                if (!isPressed) {
                                        isPressed=true;
                                        AutoFocusCallBackImpl autoFocusCallBack = new
AutoFocusCallBackImpl();
                                        mCam.autoFocus(autoFocusCallBack);
                                }
                        }
});
```

Το κουμπί αυτό, όπως βλέπουμε στον κώδικα, ενεργοποιεί το Auto Focus της κάμερας (αν η κάμερα υποστηρίζει Auto Focus), και έπειτα το Auto Focus καλεί τη μέθοδο για λήψη φωτογραφίας:

```
private class AutoFocusCallBackImpl implements Camera.AutoFocusCallback {
                public void onAutoFocus(boolean success, Camera camera) {
                //Called when the camera auto focus completes.
                 //If the camera does not support auto-focus and autoFocus is called,
                //onAutoFocus will be called immediately with a fake value of success set to true.
                   mCam.takePicture(null, null, jpegCallback);
                }
        }
```

Τα υπόλοιπα κουμπιά που βλέπουμε στην οθόνη είναι το flash της κάμερας, το zoom in και το zoom out. Αν η κάμερα δεν υποστηρίζει κάποια από αυτές τις λειτουργίες, τότε όταν πατηθεί το αντίστοιχο κουμπί δε γίνεται κάποια ενέργεια.

Στην οθόνη αυτή, όπως και σε κάθε άλλη οθόνη της εφαρμογής μας, με το πάτημα του κουμπιού menu της συσκευής Android, εμφανίζεται το μενού πλοήγησης, μέσω του οποίου μπορούμε να μεταβούμε σε κάποια άλλη οθόνη της εφαρμογής (Εικόνα 5.2). Όταν ο χρήστης πατήσει το κουμπί αυτό καλείται η μέθοδος public boolean onPrepareOptionsMenu(Menu menu) στην οποία έχουμε δηλώσει, να εμφανίζεται η διάταξη των κουμπιών του μενού, καθώς και τις ενέργειες που εκτελούνται όταν πατηθεί κάποιο από αυτά τα εικονικά κουμπιά. Ουσιαστικά όταν πατηθεί ένα κουμπί από αυτά, ξεκινάει η αντίστοιχη activity και μεταβαίνει ο χρήστης σε άλλη οθόνη. Σημειώνουμε ότι το μενού αυτό είναι scrollable, δηλαδή ο χρήστης μπορεί να το "σύρει" δεξιά ή αριστερά, ώστε να εμφανιστεί κάποια επιλογή που ίσως δε φαίνεται στη συσκευή του.



Εικόνα 5.2: Το μενού πλοήγησης της εφαρμογής, το οποίο ενεργοποιείται και απενεργοποιείται με το πάτημα του κουμπιού menu της συσκευής

5.3 Οθόνη Εισαγωγής Εικόνας

Στην οθόνη αυτή, ο χρήστης έχει τη δυνατότητα να εξερευνήσει τα αρχεία της συσκευής του, να εμφανίσει κάποια εικόνα την οποία έχει αποθηκευμένη και έπειτα να τη στείλει για αναζήτηση (Εικόνα 5.3). Ουσιαστικά πρόκειται για τον δεύτερο τρόπο αποστολής εικόνας στον εξυπηρετητή, αφού ο πρώτος είναι η απευθείας λήψη φωτογραφίας μέσω της κάμερας.

Εικόνα 5.3: Η οθόνη εισαγωγής και μεταφόρτωσης μίας εικόνας που βρίσκεται αποθηκευμένη στη συσκευή

Οι λειτουργίες της οθόνης αυτής ορίζονται στην activity UploadScreen, ενώ το user interface της ορίζεται αντίστοιχα στο αρχείο uploadscreen.xml. Ο χρήστης βλέπει μία εικόνα η οποία δηλώνει ότι δεν έχει επιλεχθεί κάποια εικόνα για αποστολή. Η εικόνα αυτή, όπως και κάθε στατική εικόνα που χρησιμοποιείται στην εφαρμογή μας, αποθηκεύεται στο φάκελο res/drawable/ και δηλώνεται στο αρχείο uploadscreen.xml ως εξής:

```
<ImageView android:id="@+id/imageView1" android:src="@drawable/noimage"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:scaleType="fitCenter" android:adjustViewBounds="true"></ImageView>
```

(την έχουμε ονομάσει noimage.jpg) Στο ίδιο αυτό αρχείο έχουν δηλωθεί και τα δύο κουμπιά που βλέπει ο χρήστης ως εξής:

```
<Button android:drawableLeft="@drawable/browse" android:text="@string/browsebutton"
android:id="@+id/buttonBrowse" android:layout_weight="1" android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
<Button android:drawableLeft="@drawable/upload5" android:text="@string/uploadimagebutton"
android:id="@+id/buttonUploadImage" android:layout_weight="1"
android:layout_width="wrap_content" android:layout_height="wrap_content"></Button>
```

Οι παραπάνω παράμετροι περιλαμβάνουν το όνομα (id) του κουμπιού, την εικόνα που θα προβάλλει, τη θέση της εικόνας και το string που θα εμφανίζει.

Όταν ο χρήστης πατήσει το κουμπί “Browse Image” τότε θα του εμφανιστεί η γκαλερί της συσκευής του ώστε να επιλέξει κάποια εικόνα, ή εάν έχει παραπάνω από μία εφαρμογές πλοήγησης των περιεχομένων της συσκευής, τότε θα του

εμφανιστεί μενού για να επιλέξει με ποιο τρόπο θέλει να εξερευνήσει τις εικόνες του. Η υλοποίηση του κουμπιού αυτού φαίνεται παρακάτω:

```
buttonBrowse = (Button) findViewById(R.id.buttonBrowse);
      buttonBrowse.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            Intent chooseFile = new Intent(Intent.ACTION_GET_CONTENT);
            chooseFile.setType("image/*");
            Intent c = Intent.createChooser(chooseFile, getString(R.string.choosefile));
            startActivityForResult(c, CHOOSE_FILE);
        }
});
```

Με το άγγιγμα λοιπόν του κουμπιού αυτού, ενεργοποιείται το intent chooseFile, μέσω του οποίου ο χρήστης εξερευνά τους φακέλους της κάρτας SD. Ουσιαστικά δηλαδή ξεκινάει μία νέα εφαρμογή η οποία θα κλείσει όταν επιστραφεί κάποιο αποτέλεσμα. Το φίλτρο "image/*" δηλώνει ότι θέλουμε μόνο αρχεία εικόνας, ώστε ο χρήστης να μη μπορεί να επιλέξει κάποιο άλλο. Η εντολή startActivityForResult σημαίνει ότι η εφαρμογή περιμένει κάποιο αποτέλεσμα από το χρήστη για να συνεχίσει τη λειτουργία της. Δηλαδή όσο ο χρήστης εξερευνά τα αρχεία του, δεν εκτελείται κάποια ενέργεια στην εφαρμογή μας, διότι ο χρήστης ουσιαστικά βρίσκεται προσωρινά σε διαφορετική εφαρμογή. Όταν ο χρήστης επιλέξει τελικά μία εικόνα, τότε καλείται η μέθοδος onActivityResult, η οποία είναι υπεύθυνη να εμφανίσει την εικόνα που επέλεξε ο χρήστης στην οθόνη (Εικόνα 5.4).

Εικόνα 5.4: Η εικόνα που επιλέξαμε εμφανίζεται στην οθόνη της εφαρμογής

Για να στείλουμε την εικόνα αυτή στον εξυπηρετητή αρκεί να πατήσουμε το κουμπί "Upload image", του οποίου η υλοποίηση παρουσιάζεται παρακάτω:

```
buttonUploadImage = (Button) findViewById(R.id.buttonUploadImage);
    buttonUploadImage.setOnClickListener(new OnClickListener() {
     public void onClick(View v) {
        if (bitmap==null) {
                //No image selected
                Context context = getApplicationContext();
                CharSequence text = getString(R.string.noImage);
                int duration = Toast.LENGTH_SHORT;
                Toast toast = Toast.makeText(context, text, duration);
                toast.setGravity(Gravity.CENTER, 0, 0);
                toast.show();
        }
        else {
                //Upload image
                //Create bitmap
                ByteArrayOutputStream stream = new ByteArrayOutputStream();
                bitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
                byte[] data = stream.toByteArray();
                //Start SendScreen activity
                Intent i = new Intent (UploadScreen.this, SendScreen.class);
                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                i.putExtra("dataUpload", data);
                startActivity (i);
            overridePendingTransition(0,0);
        }
    }
});
```

Αρχικά ελέγχουμε αν το bitmap είναι κενό ή όχι, δηλαδή αν ο χρήστης έχει επιλέξει κάποια εικόνα. Αν δεν έχει επιλέξει ο χρήστης κάποια εικόνα, τότε εμφανίζεται toast notification με το μήνυμα “No image selected”. Διαφορετικά το bitmap μετατρέπεται σε μορφή binary (byte array) και στέλνεται στην activity SendScreen, η οποία έπειτα ξεκινάει την εκτέλεσή της.

Παρατηρούμε ότι τόσο η HomeScreen όσο και η UploadScreen, στο τέλος στέλνουν την εικόνα στη SendScreen activity, η οποία είναι υπεύθυνη για την αποστολή της εικόνας στον εξυπηρετητή και τη λήψη του αποτελέσματος. Δηλαδή, προφανώς, χρειάζεται μία μόνο activity η οποία στέλνει την εικόνα στον εξυπηρετητή, την οποία καλούν οι δύο παραπάνω activities.

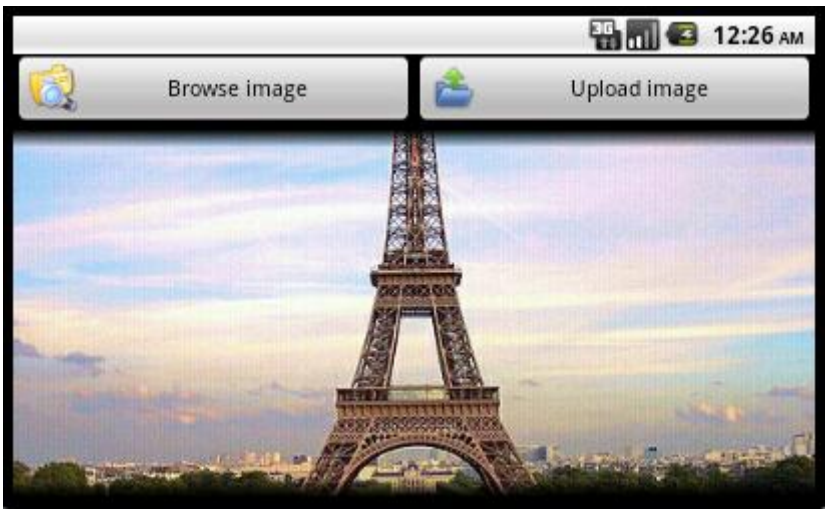
Στο σημείο αυτό, πρέπει να επισημάνουμε τη συμβατότητα της εφαρμογής μας με τους διαφορετικούς προσανατολισμούς που μπορεί να έχει η συσκευή του χρήστη. Η προεπιλεγμένη ενέργεια που εκτελείται όταν για παράδειγμα από portrait mode αλλάζουμε orientation σε landscape mode, είναι η αναδημιουργία (recreation) της activity. Δηλαδή η activity τρέχει ξανά όλο τον κώδικα που περιλαμβάνει. Το γεγονός αυτό όμως, τις περισσότερες φορές δεν είναι επιθυμητό για διάφορους λόγους. Πρώτον, αφού ήδη ο κώδικας έχει εκτελεστεί μία φορά, είναι περιττό να ξαναεκτελεστεί, ειδικά όταν αυτός περιλαμβάνει χρονοβόρες λειτουργίες. Δεύτερον, αν η activity αναδημιουργηθεί, αυτό σημαίνει ότι θα χάσουμε όποια πληροφορία είχε εισάγει ο χρήστης στην activity αυτή ή κάποιο αποτέλεσμα το οποίο είχε επιστραφεί. Για παράδειγμα στην UploadScreen, αν ο χρήστης επέλεγε κάποια εικόνα η οποία εμφανιζόταν στην οθόνη, τότε όταν άλλαζε προσανατολισμό η συσκευή, η εικόνα αυτή θα χανόταν, διότι η activity θα ξεκινούσε από την αρχή. Αρκούν, λοιπόν, ορισμένες εντολές που ξεπερνούν το παραπάνω πρόβλημα και δηλώνουν ότι η activity δε θα αναδημιουργηθεί σε περίπτωση αλλαγής προσανατολισμού, αλλά απλά θα αλλάξει η διάταξη των γραφικών στοιχείων. Πρώτον δηλώνουμε την παράμετρο android:configChanges της activity στο AndroidManifest.xml ως εξής:

```
            <activity android:name=".UploadScreen"
android:configChanges="orientation|keyboardHidden">
            </activity>
```

Έπειτα προσθέτουμε την παρακάτω μέθοδο στην activity μας:

```
@Override
   public void onConfigurationChanged(Configuration newConfig) {
      super.onConfigurationChanged(newConfig);
   }
```

η οποία δηλώνει ότι σε περίπτωση αλλαγής προσανατολισμού, η activity δε θα ξεκινήσει από την αρχή, αλλά απλά θα αλλάξουν προσανατολισμό τα γραφικά στοιχεία που περιλαμβάνει. Τις εντολές αυτές, τις έχουμε προσθέσει σε όλες τις activities της εφαρμογής μας για να διατηρήσουμε τη συμβατότητά της ως προς τον προσανατολισμό της συσκευής. Έτσι, λοιπόν, όταν έχουμε επιλέξει κάποια εικόνα στην οθόνη αυτή όπως στην εικόνα 5.4 και αλλάξουμε προσανατολισμό σε landscape, το αποτέλεσμα είναι το παρακάτω:



Εικόνα 5.5: Η activity UploadScreen όταν αλλάξουμε προσανατολισμό από portrait σε landscape (η φωτογραφία σε αυτή την περίπτωση είναι scrollable)

5.4 Οθόνη Αποστολής Εικόνας

Η οθόνη αυτή περιλαμβάνει την αποστολή της εικόνας από την εφαρμογή στον εξυπηρετητή, καθώς και τη λήψη του αντίστοιχου αποτελέσματος. Όσο η εικόνα αποστέλλεται ο χρήστης βλέπει ένα διάλογο “Uploading Image…”, ενώ στο background βρίσκεται η εικόνα που στέλνει (Εικόνα 5.6).

Εικόνα 5.6: Οθόνη αποστολής της εικόνας στον εξυπηρετητή

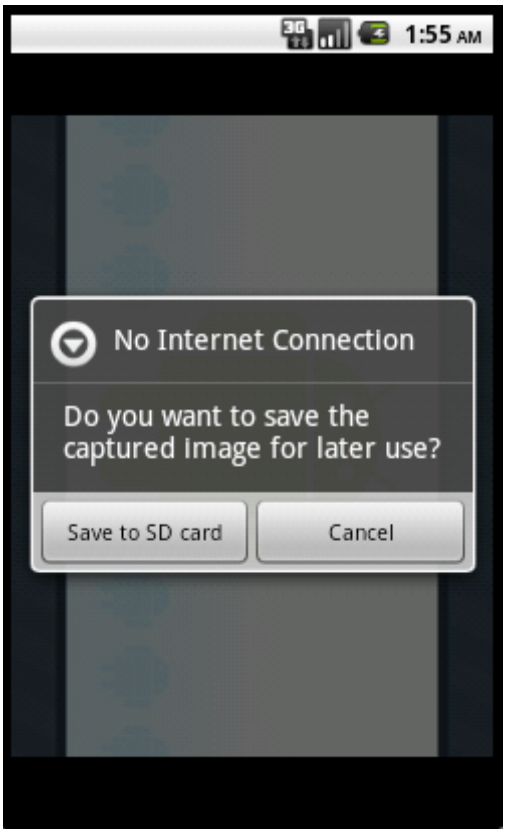
Αν και ο χρήστης στην οθόνη αυτή δεν έχει πολλές επιλογές, αντιθέτως εκτελούνται αρκετές λειτουργίες στην activity αυτή. Πρόκειται για τη SendScreen activity στην οποία εισέρχεται ο χρήστης είτε από τη HomeScreen με τη λήψη φωτογραφίας, είτε από την UploadScreen με τη μεταφόρτωση κάποιας εικόνας.

Πρώτη εργασία της activity αυτής είναι η αποκωδικοποίηση του byte[] data το οποίο περιέχει την εικόνα μας, σε bitmap, ώστε να εμφανιστεί η εικόνα που αποστέλλουμε στο background. Το byte[] data ανακτάται και μετατρέπεται σε bitmap ως εξής:

```
Bundle bundle = getIntent().getExtras();
byte[] dataHome=bundle.getByteArray("dataHome");
byte[] data=dataHome;
if (data==null) {
    //shmainei oti erxomai apo thn UploadScreen
    data=bundle.getByteArray("dataUpload");
}
//Create bitmap from data byte[]
bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);
```

Δεύτερη εργασία της SendScreen αποτελεί ο έλεγχος για σύνδεση με το διαδίκτυο. Αν δεν υπάρχει σύνδεση τότε εμφανίζεται διάλογος στο χρήστη ότι δεν υπάρχει σύνδεση και ο χρήστης μπορεί είτε να αποθηκεύσει την εικόνα για να τη χρησιμοποιήσει όταν έχει δίκτυο η συσκευή, είτε να ακυρώσει το διάλογο χωρίς περαιτέρω ενέργεια. Ωστόσο, η αποθήκευση της εικόνας έχει νόημα μόνο όταν ο χρήστης προέρχεται από τη HomeScreen, δηλαδή όταν ο χρήστης έχει τραβήξει κάποια φωτογραφία και την οποία στέλνει. Αν ο χρήστης προέρχεται από την UploadScreen, προφανώς θα έχει ήδη αποθηκευμένη την εικόνα στη συσκευή του.

Επομένως η επιλογή να αποθηκεύσει ο χρήστης την εικόνα εμφανίζεται μόνο όταν ο χρήστης έχει τραβήξει φωτογραφία μέσω της κάμερας, την οποία και στέλνει (Εικόνα 5.7).



Εικόνα 5.7: Ο διάλογος που εμφανίζεται στο χρήστη, όταν πραγματοποιεί τη λήψη μίας φωτογραφίας, αλλά δεν υπάρχει σύνδεση με κάποιο δίκτυο

Ο έλεγχος της σύνδεσης με το δίκτυο πραγματοποιείται με την παρακάτω μέθοδο:

```
public boolean networkAvailable() {
        ConnectivityManager cn = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo ni = cn.getActiveNetworkInfo();
        if (ni==null)
           return false;
        return ni.isConnected();
}
```

Για να αποθηκευτεί η φωτογραφία θα πρέπει πρώτα να δημιουργήσουμε έναν φάκελο στην κάρτα SD με το όνομα "VIRaL Captures". Έπειτα θα πρέπει να δώσουμε μία ονομασία στο αρχείο εικόνας που θα αποθηκευτεί. Συγκεκριμένα στην εφαρμογή μας ονομάζουμε τις εικόνες που αποθηκεύουμε ως "ViralImage_0001.jpg", "ViralImage_0002.jpg" κ.ο.κ. Άρα για να δώσουμε μία ονομασία στην εικόνα μας, θα πρέπει να γνωρίζουμε αν υπάρχουν ήδη αρχεία στο φάκελο "VIRaL Captures", και αν υπάρχουν να υπολογίσουμε το μέγιστο αριθμό που περιλαμβάνεται στα αρχεία αυτά και να προσθέσουμε 1. Δηλαδή αν ο χρήστης έχει αφήσει στο φάκελο αυτό τα αρχεία "ViralImage_0002.jpg", "ViralImage_0005.jpg" και "ViralImage_0017.jpg", τότε το αρχείο που

αποθηκεύουμε θα πρέπει να ονομαστεί “ViralImage_0018.jpg”. Τη λειτουργία αυτή εκτελεί η μέθοδος findFileName() που έχουμε υλοποιήσει. Η εικόνα αποθηκεύεται με τον παρακάτω τρόπο:

```
//Save to SD
String filenameSd = findFileName();
File fSd = new File(Environment.getExternalStorageDirectory()
                    + "/VIRaL Captures" + File.separator + filenameSd);
FileOutputStream outSd;
try {
        outSd = new FileOutputStream(fSd);
        bitmap.compress(Bitmap.CompressFormat.JPEG, 100, outSd);
        outSd.close();
} catch (FileNotFoundException e) {
        e.printStackTrace();
} catch (IOException e) {
        e.printStackTrace();
}
```

Αν υπάρχει σύνδεση με το δίκτυο, τότε γίνεται η αποστολή της εικόνας στον εξυπηρετητή μέσω ftp, και ο εξυπηρετητής επιστρέφει το αποτέλεσμα της αναζήτησης της εικόνας, στην activity αυτή. Η λειτουργία αυτή επειδή είναι χρονοβόρα εκτελείται σε AsyncTask, δηλαδή σε διαφορετικό thread, διότι αν την εκτελούσαμε στο main thread η εφαρμογή θα μπλόκαρε και θα εμφάνιζε σφάλμα στο χρήστη. Οι μέθοδοι υλοποίησης του AsyncTask είναι τρεις: Η μέθοδος doInBackground() αποτελεί το κύριο σώμα του AsyncTask, δηλαδή όλες τις χρονοβόρες λειτουργίες τις οποίες θέλουμε να εκτελέσει. Συγκεκριμένα στην εφαρμογή μας γίνεται η αποστολή της εικόνας στον εξυπηρετητή, μέσω ftp, και έπειτα η λήψη του αποτελέσματος που περιέχει τις πληροφορίες σχετικά με την εικόνα που αποστείλαμε. Η μέθοδος onPreExecute() περιέχει τις εντολές που θέλουμε να εκτελεστούν πριν ξεκινήσει η χρονοβόρα λειτουργία του AsyncTask. Συνήθως σε αυτό το κομμάτι δηλώνεται κάποιος διάλογος (διάλογος “Uploading Image…”) ο οποίος θέλουμε να εμφανίζεται στο χρήστη όσο ο AsyncTask εκτελεί τις εντολές που βρίσκονται στη μέθοδο doInBackground(). Τέλος η μέθοδος onPostExecute(String result) εκτελείται τη στιγμή που τελειώνει η χρονοβόρα διαδικασία που ανέλαβε να εκτελέσει ο AsyncTask. Στη μέθοδο αυτή έχουμε ορίσει να αποστέλλεται το αποτέλεσμα που πήραμε από τον εξυπηρετητή, από την activity αυτή στην activity ResultsScreen, και έπειτα να εκτελείται η ResultsScreen για την επεξεργασία του αποτελέσματος.

Στην περιγραφή της εφαρμογής, αναφέραμε ότι ο χρήστης έχει τη δυνατότητα να συνδεθεί στο λογαριασμό του ώστε να αποθηκεύει και να προβάλλει το ιστορικό του. Επομένως, η SendScreen εκτός των άλλων, θα πρέπει να ελέγχει αν ο χρήστης είναι συνδεδεμένος, ώστε σε περίπτωση που είναι, να αποθηκεύσει σε μία βάση δεδομένων, το username του χρήστη, την εικόνα που έστειλε στον εξυπηρετητή καθώς και το αποτέλεσμα που πήρε από αυτόν. Για το σκοπό αυτό, έχουμε δημιουργήσει μία βάση δεδομένων, η οποία φιλοξενείται σε εξυπηρετητή του Εργαστηρίου Εικόνας, Βίντεο και Συστημάτων Πολυμέσων του Εθνικού Μετσόβιου Πολυτεχνείου, και αποτελείται από δύο πίνακες. Ο πρώτος πίνακας έχει πεδία τα username και password των χρηστών και χρησιμοποιείται στη LoginScreen. Ο δεύτερος πίνακας περιέχει όλες τις εικόνες που έχει ανεβάσει ο κάθε χρήστης, καθώς και τα αποτελέσματα της αναζήτησής τους. Δηλαδή, έχει πεδία, το username, την εικόνα κωδικοποιημένη σε πίνακα byte[], και το αποτέλεσμα της αναζήτησης (string). Η SendScreen, λοιπόν, τη στιγμή που δέχεται το αποτέλεσμα από τον server

και είναι έτοιμη να προχωρήσει στη ResultsScreen, ελέγχει πρώτα αν ο χρήστης έχει συνδεθεί στο λογαριασμό του. Η σύνδεση αυτή γίνεται στη LoginScreen, στην οποία αποθηκεύεται το username του χρήστη σε ένα cookie στην cache του κινητού. Στη SendScreen, επομένως, ελέγχουμε αν υπάρχει αυτό το cookie, και εάν υπάρχει το διαβάζουμε για να βρούμε το username του χρήστη. Η αναζήτηση του cookie υλοποιείται στη μέθοδο checkCookie(), και η ανάγνωση του cookie για να βρεθεί το username, υλοποιείται στη μέθοδο findUsername(), όπως φαίνεται παρακάτω:

```
public boolean checkCookie() {
        File f = new File(getCacheDir()+"Cookie.viral");
        if (f.exists()) {
                return true;
        } else {
                return false;
        }
}

public String findUsername() {
        String user="";
        BufferedReader in;
        try {
                in = new BufferedReader(new FileReader(getCacheDir()+"Cookie.viral"));
                user=in.readLine();
        } catch (FileNotFoundException e) {
                e.printStackTrace();
        } catch (IOException e) {
                e.printStackTrace();
        }
        return user;
}
```

Εάν βρεθεί το cookie, τότε συνδεόμαστε στη βάση δεδομένων και εισάγουμε τα στοιχεία που αναφέραμε παραπάνω, ως εξής:

```
//syndesou sth vash
Class.forName("org.postgresql.Driver");
String url = "jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/xxxxx";
String user = "xxxxx";
String pass = "xxxxx";
Connection conn = DriverManager.getConnection(url, user, pass);
//vres to max id
int max=0;
String query = "SELECT id FROM android_history ORDER BY id DESC LIMIT 1;";
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery(query);
while (rs.next()) {
        max=rs.getInt("id");
}
rs.close();
st.close();
max++;  //apothikeuse ta stoixeia
String encodedResult = Base64Coder.encodeString(result);
String update = "INSERT INTO android_history (id, result, \"user\", query) VALUES ("+max+",
"'+encodedResult+"', "'+username+"', ?)";
PreparedStatement preSt = conn.prepareStatement(update);  //to user to vazw se eisagwgika dioti einai
preSt.setBytes(1, dataBitmap);                            //reserved word sthn postgreSql
preSt.executeUpdate();
preSt.close();
conn.close();  //close connection
```

Παρατηρούμε στον παραπάνω κώδικα, ότι προτού εισάγουμε το string result στη βάση, το κωδικοποιούμε πρώτα σε μορφή Base64 μέσω της κλάσης Base64Coder που έχουμε υλοποιήσει. Η κωδικοποίηση αυτή γίνεται, διότι το string result συνήθως περιέχει ειδικούς χαρακτήρες που δε μπορούν να αποθηκευτούν σωστά στη βάση. Με τον ίδιο σκεπτικό, στη HistoryScreen, όταν το ανακτάμε από τη βάση, πρώτα το αποκωδικοποιούμε μέσω της κλάσης Base64Coder, και έπειτα το χρησιμοποιούμε.

5.5 Οθόνη Αποτελεσμάτων

Στην οθόνη αυτή εισέρχεται ο χρήστης όταν έχει αποστείλει την εικόνα στον εξυπηρετητή και έχει λάβει το αποτέλεσμα της αναζήτησης. Η activity που αφορά στην οθόνη αυτή ονομάζεται ResultsScreen. Αρχικά, λοιπόν, διαβάζεται το string result, το οποίο επέστρεψε ο εξυπηρετητής. Υπάρχουν δύο βασικές περιπτώσεις για τα αποτελέσματα που λάβαμε από το server:

α) Ο server δε βρήκε αποτελέσματα σχετικά με την εικόνα την οποία έστειλε ο χρήστης. Στην περίπτωση αυτή το string result έχει συγκεκριμένη μορφή. Οπότε ελέγχουμε αν το string έχει αυτή τη μορφή με την εντολή: “if (result.equals("-1") || (result.contains("{\"images\":[]")))” και εάν η παραπάνω πρόταση είναι αληθής (δηλαδή δε βρέθηκαν αποτελέσματα), τότε εμφανίζουμε στο χρήστη μία toast notification ότι δε βρέθηκαν παρόμοιες εικόνες. Ο χρήστης στην περίπτωση αυτή μπορεί είτε να μεταβεί σε κάποια άλλη οθόνη μέσω του menu, είτε να αποθηκεύσει την εικόνα την οποία έστειλε.

β) Βρέθηκαν αποτελέσματα, δηλαδή η παραπάνω λογική πρόταση είναι ψευδής. Στην περίπτωση αυτή πρέπει να επεξεργαστούμε και να εμφανίσουμε τα αποτελέσματα στο χρήστη. Τα αποτελέσματα αυτά όμως περιέχουν αρκετές πληροφορίες οπότε είναι σχεδιαστικά λάθος να προσπαθήσουμε να τα εμφανίσουμε όλα σε μία οθόνη. Για το σκοπό αυτό, δημιουργούνται από την activity αυτή, 4 καρτέλες (tabs) στις οποίες μπορεί να περιηγηθεί ο χρήστης και να δει τα αντίστοιχα αποτελέσματα. Κάθε μία καρτέλα από αυτές αποτελεί ουσιαστικά μία subactivity της ResultsScreen. Η πρώτη καρτέλα, στην οποία μεταβαίνει αυτόματα ο χρήστης, αποτελεί τις βασικές πληροφορίες της εικόνας που έστειλε και ορίζεται στην activity ResOriginalScreen. Η δεύτερη καρτέλα αποτελεί το δυναμικό χάρτη, στο σημείο το οποίο βρίσκεται το περιεχόμενο της εικόνας και ορίζεται στην activity ResMapScreen. Στην τρίτη καρτέλα εμφανίζονται οι παρόμοιες εικόνες που βρέθηκαν και ορίζονται στην ResSimilarScreen, ενώ στην τέταρτη καρτέλα εμφανίζονται οι εικόνες που είναι παρόμοιες με αυτές που βρέθηκαν προηγουμένως, και ορίζονται στην ResSimofsimScreen. Με λίγα λόγια μπορούμε να πούμε ότι στην τρίτη καρτέλα εμφανίζονται εικόνες σχεδόν ίδιες με αυτή που έστειλε ο χρήστης, ενώ στην τέταρτη εμφανίζονται εικόνες στον ίδιο χώρο με αυτόν που βρίσκεται το περιεχόμενο της εικόνας, αλλά με λιγότερες ομοιότητες.

5.5.1 Πληροφορίες Περιεχομένου

Στην πρώτη καρτέλα εμφανίζονται οι βασικές πληροφορίες που προκύπτουν από το αποτέλεσμα της αναζήτησης της εικόνας, μαζί με την ίδια την εικόνα την οποία έστειλε ο χρήστης. Αυτές είναι η πόλη και χώρα στις οποίες βρίσκεται το περιεχόμενο της εικόνας, η ονομασία του αν πρόκειται για κάποιο τοπωνύμιο ή μνημείο, και οι ετικέτες του περιεχομένου αυτού, τις οποίες έχουν ορίσει χρήστες. Ακολουθούν δύο ενδεικτικές εικόνες που περιέχουν τα παραπάνω αποτελέσματα:

Εικόνα 5.8: Πρώτη καρτέλα αποτελεσμάτων (Ακρόπολη)

Εικόνα 5.9: Πρώτη καρτέλα αποτελεσμάτων (Πύργος του Eiffel)

Εξηγούμε την ανάκτηση των παραπάνω πληροφοριών από το string result. Το string αυτό έχει συγκεκριμένη δομή. Πρόκειται για ένα json string το οποίο περιέχει κάποια json objects. Η γενική του μορφή είναι η παρακάτω:

```
{
   "images": [],
   "tags": [
      ""
   ],
   "landmarks": [],
   "sos": [],
   "location": {
      "latitude": 100,
      "longitude": 100
   },
   "city": "",
   "country": ""
}
```

Στον πίνακα (json array) “images” περιέχονται τα URLs των similar εικόνων, στον πίνακα “tags” περιέχονται οι ετικέτες χρηστών, στα “landmarks” βρίσκονται ονομασίες για το περιεχόμενο της εικόνας, στο “sos” περιέχονται τα URLs των similar of similar εικόνων, “latitude” και “longitude” αποτελούν τις συντεταγμένες για να προσδιορίσουμε το περιεχόμενο της εικόνας στο χάρτη, ενώ “city” και “country” περιέχουν την πόλη και χώρα του περιεχομένου της εικόνας. Για παράδειγμα για να βρούμε την πόλη και χώρα εργαζόμαστε ως εξής:

```
jObject = new JSONObject(result);
city = jObject.getString("city");
country = jObject.getString("country");
```

Με τον ίδιο τρόπο αντλούμε και τις υπόλοιπες πληροφορίες.

Παρατηρούμε ότι η πόλη, η χώρα και οι ονομασίες του περιεχομένου της εικόνας περιέχουν μία υπερσύνδεση (link) στη σελίδα της Wikipedia και συγκεκριμένα στο θέμα που κάνει κλικ (ή touch) ο χρήστης. Δηλαδή αν ο χρήστης αγγίξει τη λέξη “Parthenon” στην εικόνα 5.8 τότε θα μεταβεί μέσω του φυλλομετρητή (browser) της συσκευής του, στη σελίδα http://en.m.wikipedia.org/wiki/Parthenon (Εικόνα 5.10).

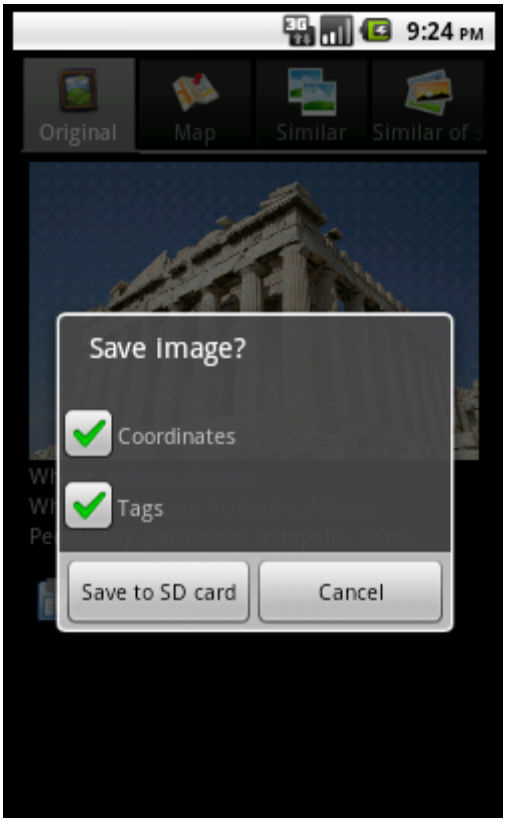
Εικόνα 5.10: Ο χρήστης όταν αγγίξει κάποια λέξη (πόλη, χώρα, ονομασία τοπωνυμίου) μεταβαίνει στην αντίστοιχη σελίδα της Wikipedia.

Για να δημιουργήσουμε μία υπερσύνδεση σε ένα κείμενο, αρκεί να γράψουμε το κείμενο με κώδικα html και έπειτα να ορίσουμε να το εμφανίσει η TextView που έχει δημιουργηθεί για το σκοπό αυτό. Για παράδειγμα έχουμε δηλώσει μία TextView στο αρχείο xml της παραπάνω κλάσης, για να εμφανίσει την πόλη και χώρα και την έχουμε ονομάσει TextViewLoc. Για να αναφερθούμε στο γραφικό στοιχείο αυτό μέσα από τον κώδικα της κλάσης, και να ορίσουμε το παραπάνω κείμενο, χρησιμοποιούμε τις παρακάτω εντολές:

```
TextView textViewLoc = (TextView)findViewById(R.id.TextViewLoc);
textViewLoc.setText(Html.fromHtml("<a
href=\"http://en.wikipedia.org/wiki/"+city+"\">"+city+"</a>"+", <a
href=\"http://en.wikipedia.org/wiki/"+country+"\">"+country+"</a>"));
textViewLoc.setMovementMethod(LinkMovementMethod.getInstance());
textViewLoc.setLinksClickable(true);
```

Ο χρήστης έχει τη δυνατότητα να αποθηκεύσει την εικόνα την οποία έχει στείλει στο server (και εμφανίζεται στην καρτέλα αυτή) με ή χωρίς τις παραπάνω πληροφορίες ενσωματωμένες σε αυτήν. Όταν πατήσει το εικονίδιο της δισκέτας που φαίνεται κάτω αριστερά (εικόνες 5.8, 5.9) τότε εμφανίζεται ένας διάλογος και ο χρήστης επιλέγει αν θέλει να αποθηκεύσει την εικόνα ή όχι (Εικόνα 5.11). Αν επιλέξει να ενσωματωθούν και οι παραπάνω πληροφορίες στην εικόνα, τότε θα μπορεί να τις ανακτήσει με κάποιον Exif/metadata viewer, ο οποίος διαβάζει τα μεταδεδομένα τα οποία βρίσκονται ενσωματωμένα στην εικόνα. Για παράδειγμα, έστω ότι ο χρήστης επιλέγει να αποθηκεύσει τη φωτογραφία της Ακρόπολης, την οποία έστειλε στο server και επιλέγει να ενσωματώσει σε αυτήν και τις πληροφορίες τις οποίες βλέπει. Τότε η τοποθεσία του περιεχομένου της εικόνας (πόλη και χώρα),

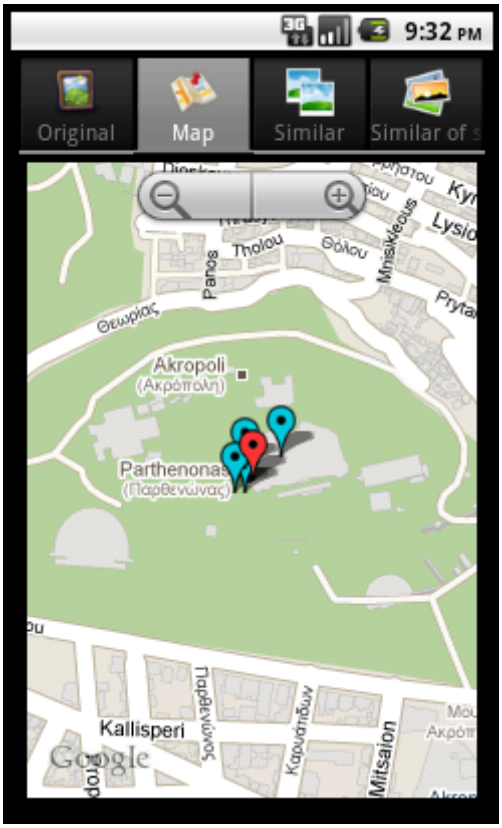
οι γεωγραφικές συντεταγμένες, καθώς και η ονομασία του τοπωνυμίου αυτού, θα αποθηκευτούν σαν μεταδεδομένα στην εικόνα, τα οποία μπορεί να διαβάσει ο χρήστης αν ανοίξει την εικόνα αυτή από οποιονδήποτε exif/metadata viewer, ο οποίος προφανώς δεν είναι ανάγκη να υπάρχει στη συσκευή του.



Εικόνα 5.11: Ο χρήστης έχει τη δυνατότητα να αποθηκεύσει την εικόνα στη συσκευή του, και αν θέλει να ενσωματώσει σε αυτή τις παραπάνω πληροφορίες

5.5.2 Χάρτης Τοποθεσίας

Η δεύτερη καρτέλα περιέχει έναν δυναμικό χάρτη στο σημείο που βρίσκεται το περιεχόμενο της εικόνας, την οποία έστειλε ο χρήστης (Εικόνα 5.12). Η κόκκινη καρφίτσα αντιπροσωπεύει την εκτιμώμενη τοποθεσία του περιεχομένου. Οι μπλε καρφίτσες αποτελούν τις τοποθεσίες των παρόμοιων εικόνων οι οποίες βρέθηκαν και εμφανίζονται στην τρίτη καρτέλα (similar images). Ο χρήστης αν αγγίξει κάποια από αυτές τις καρφίτσες, μπορεί να δει ακριβώς τις συντεταγμένες (latitude & longitude) της τοποθεσίας αυτής. Φυσικά ο χρήστης μπορεί να περιηγηθεί στο δυναμικό αυτό χάρτη ελεύθερα.



Εικόνα 5.12: Δεύτερη καρτέλα αποτελεσμάτων (χάρτης τοποθεσίας)

Για τη χρήση του παραπάνω χάρτη, θα πρέπει αρχικά να τον δηλώσουμε στο αρχείο xml, το οποίο αντιπροσωπεύει τη συγκεκριμένη activity. Επομένως στο αρχείο resmapscreen.xml, προσθέτουμε τις παρακάτω εντολές:

```
<com.google.android.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="xxxxx"/>
```

Έπειτα στην activity που βρισκόμαστε, ενεργοποιούμε το χάρτη με τις παρακάτω εντολές:

```
//Displaying map
mapView = (MapView) findViewById(R.id.mapView);
mc = mapView.getController();
//Coordinates of estimated location:
p1 = new GeoPoint(
        (int) (latitude * 1E6),
        (int) (longitude * 1E6));
mc.animateTo(p1);
mc.setZoom(17);
```

Για να εμφανιστεί η κόκκινη καρφίτσα στο χάρτη, θα πρέπει αρχικά να έχουμε την εικόνα αυτή αποθηκευμένη στα application resources και συγκεκριμένα στο φάκελο res/drawable/, και έπειτα να χρησιμοποιήσουμε την κλάση CustomItemizedOverlay για να εμφανιστεί η καρφίτσα στην τοποθεσία που θέλουμε, καθώς και να εμφανίσει

κάποιο μήνυμα τη στιγμή που την αγγίζει ο χρήστης. Η λειτουργία αυτή υλοποιείται με τις παρακάτω εντολές: (με τον ίδιο τρόπο προστίθενται και οι μπλε καρφίτσες)

```
//Add location markers
List<Overlay> mapOverlays = mapView.getOverlays();
Drawable drawable = this.getResources().getDrawable(R.drawable.redpin);
CustomItemizedOverlay itemizedOverlay= new CustomItemizedOverlay(drawable, this);
OverlayItem overlayitem= new OverlayItem(p1, getString(R.string.estimatedLoc),
Double.toString(latitude)+", "+Double.toString(longitude));      ;
//Red marker: estimated location
itemizedOverlay.addOverlay(overlayitem);
mapOverlays.add(itemizedOverlay);
mapView.invalidate();
```

5.5.3 Παρόμοιες Εικόνες

Στην καρτέλα αυτή, εμφανίζονται παρόμοιες εικόνες με αυτή που έστειλε ο χρήστης για αναγνώριση περιεχομένου. Οι εικόνες αυτές παρουσιάζονται σε τετράδες, ενώ για λόγους εξοικονόμησης χώρου και χρόνου, έχει οριστεί να εμφανίζονται το πολύ 16 εικόνες. Έστω, λοιπόν, ότι ο χρήστης επιλέγει την εικόνα του Πύργου του Eiffel για αναζήτηση παρομοίων. Τότε λαμβάνει τις παρακάτω εικόνες ως αποτέλεσμα:

Εικόνα 5.13: Τρίτη καρτέλα αποτελεσμάτων (παρόμοιες εικόνες)

Για την εμφάνιση των παραπάνω εικόνων, θα πρέπει αρχικά να έχουμε ορίσει τα κατάλληλα γραφικά στοιχεία στο ressimilarscreen.xml και συγκεκριμένα μία ImageView για κάθε εικόνα που θέλουμε να προβάλουμε. Έπειτα μέσα από την κλάση μας, ορίζουμε ένα URL (το οποίο αντλούμε από το json string result) για κάθε

ImageView, ώστε να προβληθεί η αντίστοιχη εικόνα. Επίσης, ορίζουμε ως clickable την κάθε ImageView, ώστε όταν ο χρήστης αγγίξει μία εικόνα, να ξεκινήσει η activity ImageFullScreen, η οποία προβάλλει την εικόνα την οποία επέλεξε ο χρήστης σε full screen (Εικόνα 5.14). (Σημειώνεται ότι οι παρόμοιες εικόνες που εμφανίζονται αποτελούν ιδιοκτησία της διαδικτυακής υπηρεσίας flickr® και για το λόγο αυτό, εμφανίζεται το λογότυπο της υπηρεσίας αυτής, το οποίο μάλιστα με touch μεταφέρει το χρήστη στην αντίστοιχη σελίδα.)

Εικόνα 5.14: Αν ο χρήστης επιλέξει μία εικόνα από τις εμφανιζόμενες, τότε η εικόνα αυτή προβάλλεται σε full screen (εδώ έχει επιλεχθεί η πρώτη εικόνα από τις παρόμοιες που εμφανίζονται στην εικόνα 5.13)

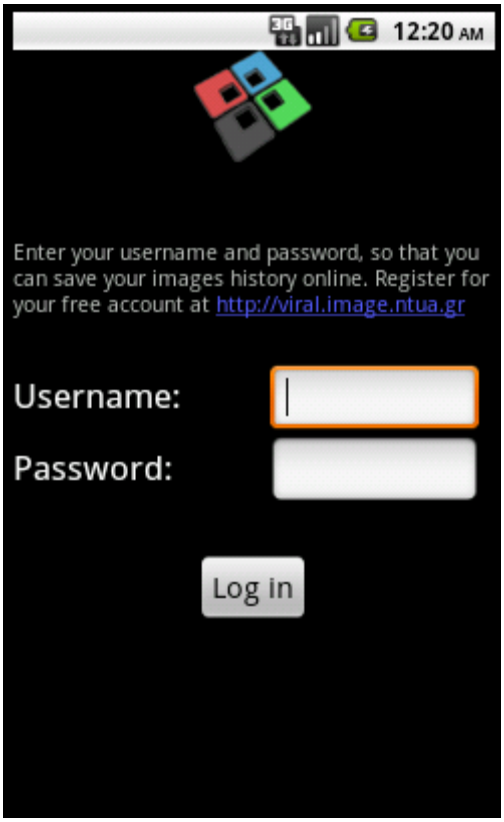
5.5.4 Παραπλήσιες Εικόνες

Πρόκειται για την τέταρτη και τελευταία καρτέλα των αποτελεσμάτων. Η υλοποίηση της καρτέλας αυτής είναι ακριβώς ίδια με την υλοποίηση της καρτέλας των παρόμοιων εικόνων. Η μοναδική διαφορά είναι ότι για να αντλήσουμε τα URLs των εικόνων αυτών από το json string result, χρησιμοποιούμε τον json array “sos” (sos = similar of similar) αντί για τον “images”, ο οποίος περιέχει τις εικόνες της τρίτης καρτέλας. Οι εικόνες αυτές είναι παρόμοιες με τις εικόνες που εμφανίζονται στην τρίτη καρτέλα, δηλαδή δεν αποτελούν άμεσα παρόμοια εικόνα με αυτή του χρήστη. (Από το γεγονός αυτό, προκύπτει και η ονομασία “similar of similar”.)

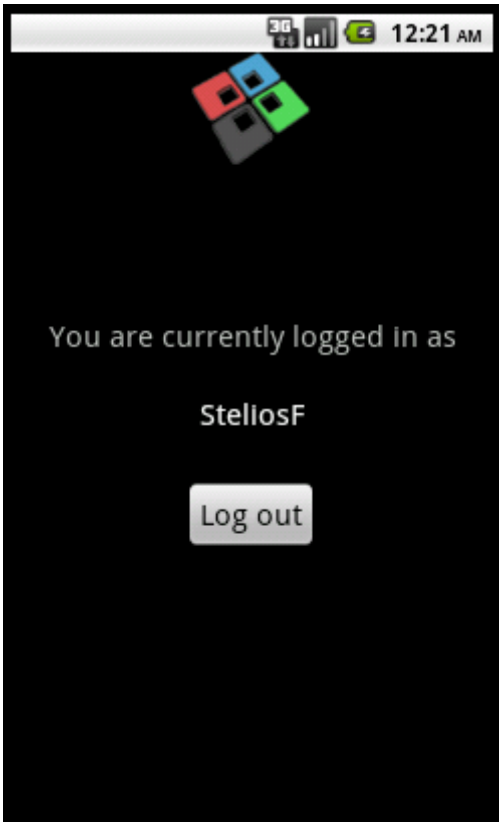
Εικόνα 5.15: Τέταρτη καρτέλα αποτελεσμάτων (παραπλήσιες εικόνες)

5.6 Οθόνη Λογαριασμού

Ο χρήστης εισέρχεται στην οθόνη αυτή μέσω της επιλογής “Account” του βασικού μενού πλοήγησης της εφαρμογής ή μέσω της επιλογής “History” εάν δεν έχει συνδεθεί στο λογαριασμό του. Πρόκειται για την activity LoginScreen, στην οποία ο χρήστης μπορεί να συνδεθεί στο λογαριασμό του, ή να αποσυνδεθεί αν είναι ήδη συνδεδεμένος (Εικόνες 5.16, 5.17).



Εικόνα 5.16: Οθόνη σύνδεσης του χρήστη στο λογαριασμό του



Εικόνα 5.17: Αν ο χρήστης έχει συνδεθεί, εμφανίζεται το παραπάνω μήνυμα

Όταν ο χρήστης εισέρχεται στην οθόνη αυτή, αρχικά ελέγχεται από την activity αν ο χρήστης είναι συνδεδεμένος ή όχι στο λογαριασμό του. Ο έλεγχος αυτός πραγματοποιείται με αναζήτηση του αντίστοιχου cookie στη μνήμη cache της συσκευής του χρήστη. Αν ο χρήστης δεν έχει συνδεθεί τότε εμφανίζονται τα πεδία username και password, για να τα συμπληρώσει. Αν ο χρήστης έχει συνδεθεί τότε εμφανίζεται το username του λογαριασμού του και ένα κουμπί για να αποσυνδεθεί. Η επιλογή "Log out" υλοποιείται απλά με τη διαγραφή του cookie από την cache της συσκευής του χρήστη.

Όταν ο χρήστης εισάγει τα στοιχεία του και πατήσει "Log in" τότε εκτελούνται διάφορες χρονοβόρες λειτουργίες σε AsyncTask. Αρχικά ελέγχεται αν ο χρήστης είναι συνδεδεμένος σε κάποιο δίκτυο, και εάν όχι, εμφανίζεται toast notification με το αντίστοιχο μήνυμα. Αν ο χρήστης είναι συνδεδεμένος με το διαδίκτυο, τότε η activity συνδέεται στη βάση δεδομένων, η οποία φιλοξενείται σε εξυπηρετητή του Εργαστηρίου Εικόνας, Βίντεο και Συστημάτων Πολυμέσων του Εθνικού Μετσόβιου Πολυτεχνείου, και περιέχει τα στοιχεία των χρηστών. Με τις παρακάτω εντολές, ελέγχεται αν το username και password του χρήστη συμφωνούν με τα αντίστοιχα πεδία του πίνακα "android_users" της βάσης δεδομένων:

```
Integer result=0;
Class.forName("org.postgresql.Driver");
String url = "jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/xxxxx";
String user = "xxxxx";
String pass = "xxxxx";
Connection conn = DriverManager.getConnection(url, user, pass);
String query = "SELECT * FROM android_users WHERE user_id='"+username+"' AND
password='"+password+"'";
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery(query);
while (rs.next()) {
        //vrethike o xrhsths
        result=1;
}
rs.close();
st.close();
conn.close();
```

Έπειτα ελέγχεται η τιμή της σημαίας (flag) result. Αν η τιμή αυτή είναι 0, τότε εμφανίζεται toast notification στο χρήστη πληροφορώντας τον ότι το username ή το password είναι λανθασμένο. Αν η τιμή του result είναι 1, σημαίνει ότι ο χρήστης βρέθηκε στη βάση και τα πεδία username, password συμφωνούν, επομένως ο χρήστης συνδέεται επιτυχώς στο λογαριασμό του. Δημιουργείται, λοιπόν, ένα αρχείο με το όνομα "Cookie.viral", στο οποίο γράφουμε το username του χρήστη και το αποθηκεύουμε στην cache ως εξής:

```
//vrethike o xrhsths ara dhmiourghse cookie
File f = new File(getCacheDir()+"Cookie.viral");
BufferedWriter output;
try {
        output = new BufferedWriter(new FileWriter(f));
        output.write(username);
        output.close();
} catch (IOException e) {
        e.printStackTrace();
}
```

Προφανώς, το username, πρέπει να αποθηκευτεί στο αρχείο αυτό, ώστε να γνωρίζει κάθε στιγμή η εφαρμογή μας, το username του χρήστη που έχει συνδεθεί.

5.7 Οθόνη Ιστορικού

Στην οθόνη αυτή εισέρχεται ο χρήστης μέσω της επιλογής "History" του βασικού μενού πλοήγησης της εφαρμογής. Πρόκειται για την activity HistoryScreen, η οποία είναι υπεύθυνη για την εμφάνιση όλων των εικόνων τις οποίες έχει ανεβάσει ο χρήστης, ανεξαρτήτως αποτελέσματος που έλαβε από τον εξυπηρετητή (δηλαδή εμφανίζονται όλες οι εικόνες, είτε βρέθηκαν παρόμοιες με αυτές είτε όχι). Έστω, λοιπόν, ότι ο χρήστης έχει αποστείλει στον εξυπηρετητή συνολικά 4 εικόνες. Όταν επιλέξει να δει το ιστορικό του, το αποτέλεσμα θα είναι το παρακάτω:

Εικόνα 5.18: Το ιστορικό του χρήστη

Η πρώτη εργασία της activity αυτής είναι να ελέγξει αν ο χρήστης είναι συνδεδεμένος ή όχι σε κάποιο λογαριασμό. Δηλαδή, ελέγχει την ύπαρξη του cookie στη μνήμη cache της συσκευής του χρήστη. Εάν το cookie υπάρχει, δηλαδή ο χρήστης έχει συνδεθεί, τότε εμφανίζει το ιστορικό του (αν υπάρχει σύνδεση με το διαδίκτυο), αλλιώς τον μεταφέρει στην activity LoginScreen, ώστε να εισάγει τα στοιχεία του για να συνδεθεί.

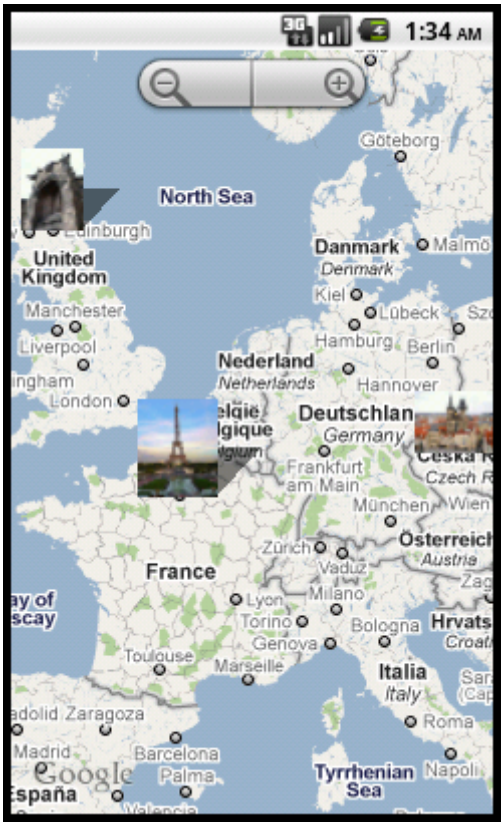
Αν το cookie υπάρχει, και η συσκευή του χρήστη είναι συνδεδεμένη με το διαδίκτυο, τότε η activity διαβάζει το username του χρήστη από το cookie και συνδέεται στη βάση δεδομένων (με τον ίδιο τρόπο που συνδέεται και στις προηγούμενες activities). Έπειτα εντοπίζει στον πίνακα της βάσης δεδομένων, "android_history", όσες εγγραφές έχουν ως πεδίο τους το username του χρήστη, δηλαδή όσες εικόνες έχει ανεβάσει ο συγκεκριμένος χρήστης. Από την κάθε εγγραφή, λαμβάνει το byte[] που περιέχει την εικόνα και το μετατρέπει σε bitmap

για να εμφανίσει την εικόνα, καθώς και το αποτέλεσμα (json string result), το οποίο είχε επιστρέψει ο εξυπηρετητής όταν ο χρήστης είχε ανεβάσει τη συγκεκριμένη εικόνα. Το result αποκωδικοποιείται (διότι έχει κωδικοποιηθεί στη μορφή Base64 όπως έχουμε αναφέρει) και αποθηκεύεται, ώστε όταν ο χρήστης επιλέξει κάποια εικόνα από το ιστορικό του, να μεταβεί κατευθείαν στην ResultsScreen, η οποία θα περιέχει τις καρτέλες με τα αποτελέσματα της συγκεκριμένης εικόνας. Δηλαδή ο χρήστης μπορεί να δει τα αποτελέσματα από τις εικόνες που περιέχονται στο ιστορικό του, χωρίς να περιμένει να ανέβει η κάθε εικόνα στο server και να λάβει τα αποτελέσματα.

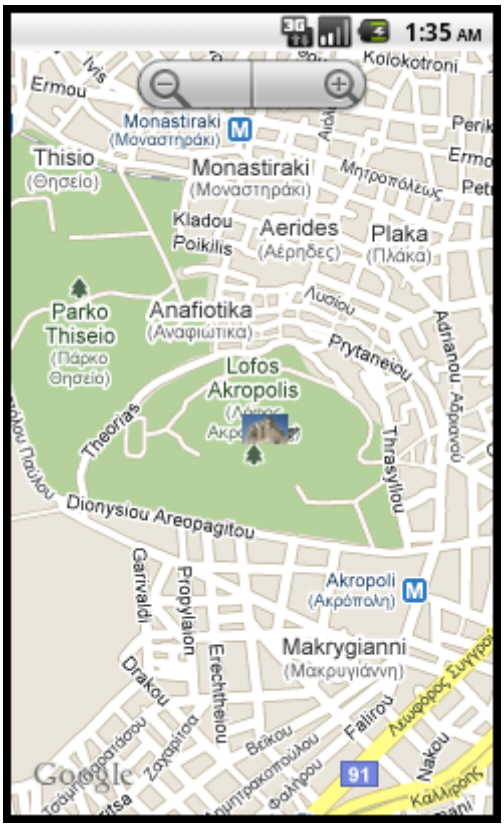
Η επιλογή "Clear history" καθαρίζει το ιστορικό του χρήστη, διαγράφοντας όλες τις εγγραφές που βρίσκονται στον αντίστοιχο πίνακα της βάσης δεδομένων και αφορούν το username του.

Τέλος, το κουμπί "History Map" εμφανίζει τις παραπάνω εικόνες σε δυναμικό χάρτη, με την κάθε εικόνα να τοποθετείται στο σημείο του χάρτη, στο οποίο βρίσκεται το περιεχόμενο της εικόνας. Η εργασία αυτή γίνεται στην activity HistoryMapScreen, και οι εικόνες τοποθετούνται στις αντίστοιχες γεωγραφικές συντεταγμένες, με τον ίδιο τρόπο που τοποθετούνται οι μπλε καρφίτσες στο χάρτη στην activity ResMapScreen. Στο παράδειγμα της εικόνας 5.18, ο χρήστης έχει ανεβάσει μία εικόνα από την Αθήνα, μία από το Παρίσι, μία από την Πράγα και μία από το Εδιμβούργο. Μόλις ο χρήστης πατήσει το κουμπί "History Map" θα δει τις εικόνες αυτές σε δυναμικό χάρτη. Οι παρακάτω εικόνες δείχνουν τα αποτελέσματα σε διαφορετικές μεγεθύνσεις του χάρτη:

Εικόνα 5.19: Χάρτης ιστορικού του χρήστη



Εικόνα 5.20: Χάρτης ιστορικού του χρήστη



Εικόνα 5.21: Χάρτης ιστορικού του χρήστη

Αν ο χρήστης πατήσει μία από τις παραπάνω εικόνες που βρίσκονται στο χάρτη, τότε θα εμφανιστεί διάλογος με πληροφορίες σχετικά με την πόλη, τη χώρα και την ονομασία του επιλεγμένου τοπωνυμίου (Εικόνα 5.22).

Εικόνα 5.22: Ο χρήστης αγγίζει την εικόνα της Ακρόπολης και εμφανίζονται οι αντίστοιχες πληροφορίες

5.8 Οθόνη "Σχετικά"

Στην οθόνη αυτή, η οποία ορίζεται στην activity AboutScreen, παρουσιάζονται πληροφορίες σχετικά με την εφαρμογή καθώς και την ομάδα που είναι υπεύθυνη για την κατασκευή της (Εικόνα 5.23).

Εικόνα 5.23: Η οθόνη "About" η οποία περιέχει πληροφορίες σχετικά με την εφαρμογή

5.9 Γλώσσα της Εφαρμογής

Σε προηγούμενη παράγραφο αναλύσαμε τον τρόπο με τον οποίο υλοποιούμε μία εφαρμογή ώστε να είναι συμβατή με περισσότερες από μία γλώσσες. Συγκεκριμένα η εφαρμογή μας χρησιμοποιεί ως προεπιλεγμένη γλώσσα την αγγλική, ενώ στις συσκευές που χρησιμοποιούν την ελληνική γλώσσα, όλα τα strings της εφαρμογής εμφανίζονται στα ελληνικά, με την αντίστοιχη τιμή που έχει οριστεί στο αρχείο res/values-el/strings.xml. Ενδεικτικά δείχνουμε ορισμένες εικόνες από την εφαρμογή όταν εκτελείται σε συσκευή που χρησιμοποιεί την ελληνική γλώσσα:

Εικόνα 5.24: Οθόνη εισαγωγής και μεταφόρτωσης εικόνας (με ενεργοποιημένο το κυρίως μενού)

Εικόνα 5.25: Οθόνη αποτελεσμάτων

6

Συμπεράσματα και μελλοντικές εξελίξεις

6.1 Σύνοψη και συμπεράσματα

Στην παρούσα διπλωματική εργασία ασχοληθήκαμε με την ανάπτυξη μίας εφαρμογής, η οποία θα εκτελείται σε κινητές συσκευές που χρησιμοποιούν το λειτουργικό σύστημα Android. Η επιλογή της ανάπτυξης της εφαρμογής για το συγκεκριμένο λειτουργικό σύστημα κρίνεται επιτυχής για αρκετούς λόγους:

- Το Android τη στιγμή αυτή χρησιμοποιείται στο μεγαλύτερο ποσοστό smartphones της αγοράς, ενώ περίπου 550.000 νέες συσκευές Android ενεργοποιούνται καθημερινά.
- Ο αριθμός αυτός αυξάνεται ανά εβδομάδα περίπου κατά 4,4% με πρόβλεψη για 1,5 εκατομμύριο καθημερινών ενεργοποιήσεων νέων συσκευών περί τα τέλη του 2011.
- Οι μεγαλύτερες εταιρίες κατασκευής κινητών τηλεφώνων, όπως Sony Ericsson, Samsung, LG, HTC, Motorola, υιοθετούν το λειτουργικό αυτό σύστημα σε όλο και περισσότερες συσκευές τους.
- Το Android Market τη στιγμή αυτή διαθέτει πάνω από 250.000 εφαρμογές, με περισσότερα από 6 δισεκατομμύρια downloads συνολικά.
- Σε γενικές γραμμές, κρίνεται εύκολη η εκμάθηση της φιλοσοφίας του λειτουργικού αυτού συστήματος, καθώς και ο τρόπος ανάπτυξης εφαρμογών.
- Δεν απαιτούνται εξειδικευμένες γνώσεις προγραμματισμού για την ανάπτυξη απλών εφαρμογών. Ο καθένας μπορεί να ασχοληθεί και να δημιουργήσει μία προσωπική εφαρμογή, αρκεί να έχει εξοικειωθεί με τη γλώσσα

προγραμματισμού Java. Ωστόσο, για απλές λειτουργίες, δεν απαιτούνται εξειδικευμένες γνώσεις ούτε της γλώσσας αυτής.

- Όλα τα προγραμματιστικά εργαλεία που χρειάζονται για την ανάπτυξη εφαρμογών κυκλοφορούν ελεύθερα στο διαδίκτυο και καθένας μπορεί να τα κατεβάσει και να τα χρησιμοποιήσει χωρίς κανένα κόστος.
- Η επίσημη ιστοσελίδα της Google σχετικά με την κατασκευή εφαρμογών σε Android, περιλαμβάνει αναλυτικούς οδηγούς που βοηθάνε τον προγραμματιστή στην εξοικείωση με τη νέα αυτή τεχνολογία.
- Υπάρχουν πάρα πολλές διαδικτυακές κοινότητες που ασχολούνται καθαρά με τον προγραμματισμό εφαρμογών για Android. Χαρακτηριστικό παράδειγμα αποτελεί το γεγονός ότι στις περισσότερες ιστοσελίδες ερωταπαντήσεων γενικού προγραμματιστικού ενδιαφέροντος, η ετικέτα "Android" βρίσκεται ανάμεσα στις δημοφιλέστερες. Αυτό σημαίνει ότι συνεχώς προγραμματιστές ανταλλάσουν ιδέες και πληροφορίες σχετικές με την κατασκευή εφαρμογών στο σύστημα αυτό.

Η έρευνα που πραγματοποιήθηκε για την εκπόνηση της διπλωματικής αυτής εργασίας κρίνεται άκρως ενδιαφέρουσα. Κατανοήθηκε η έννοια της προσωποποίησης περιεχομένου και εφαρμόστηκαν στην πράξη βασικές έννοιες σχετικά με την αλληλεπίδραση ανθρώπου-μηχανής και την ανάλυση περιεχομένου. Αποκτήθηκαν πολύ σημαντικές γνώσεις για τις τεχνολογίες των έξυπνων κινητών τηλεφώνων και τον τρόπο ανάπτυξης εφαρμογών σε αυτά. Προτείνεται ανεπιφύλακτα η ενασχόληση με τις τεχνολογίες αυτές, σε φοιτητές, προγραμματιστές ή μηχανικούς, καθώς αποτελεί έναν καινούριο και συνεχώς αναπτυσσόμενο κλάδο της επιστήμης υπολογιστών, ο οποίος σε καμία περίπτωση δεν είναι κορεσμένος, αλλά αντίθετα δημιουργεί συνεχώς νέες θέσεις εργασίας.

6.2 Μελλοντικές εξελίξεις

Το Android Market δίνει τη δυνατότητα στους προγραμματιστές να δημοσιεύσουν ελεύθερα και χωρίς κόστος τις εφαρμογές τις οποίες έχουν κατασκευάσει. Η διαδικασία είναι απλή και το μόνο αρχείο που χρειάζεται για την υπηρεσία αυτή είναι το εκτελέσιμο apk της εφαρμογής. Με τον τρόπο αυτό, η εφαρμογή γίνεται προσβάσιμη σε όλους τους χρήστες Android, οι οποίοι μπορούν να την κατεβάσουν ανά πάσα στιγμή από το Android Market, να τη βαθμολογήσουν ή να γράψουν κάποια κριτική.

Στο πλαίσιο αυτό, σχεδιάζεται η δημοσίευση της εφαρμογής μας στο Android Market το συντομότερο δυνατό, καθώς και η συνεχής ενημέρωση (update) αυτής για προσθήκη νέων δυνατοτήτων ή διόρθωση ορισμένων σφαλμάτων τα οποία αναφέρονται από χρήστες (feedback). Συγκεκριμένα προτείνονται οι παρακάτω επεκτάσεις και βελτιώσεις, προτού δημοσιευτεί η εφαρμογή στο Android Market, και οι οποίες είναι εύκολα υλοποιήσιμες:

- Βελτιωμένο user interface, ώστε να γίνει ακόμα πιο φιλικό προς το χρήστη.
- Προβολή των μεταδεδομένων, τα οποία είναι ενσωματωμένα στις εικόνες του χρήστη, μέσα από την εφαρμογή, χωρίς να χρειάζεται ο χρήστης να χρησιμοποιεί για τη λειτουργία αυτή διαφορετική υπηρεσία.
- Μετάφραση της γλώσσας της εφαρμογής στις πιο δημοφιλείς ξένες γλώσσες, με στόχο να αποκτήσουν οι αντίστοιχοι χρήστες επιπλέον κίνητρο να τη χρησιμοποιήσουν.

- Δημιουργία online προσωπικού χώρου για κάθε χρήστη, ως επέκταση του ιστορικού του, στον οποίο θα μπορεί να αποθηκεύει εικόνες, να τις ταξινομεί και να τις οργανώνει σε φακέλους ή άλμπουμ. Παράλληλα, θα μπορεί να εμφανίζεται αυτόματη λίστα στο χρήστη με τους προορισμούς στους οποίους έχει ταξιδέψει και προκύπτουν από τις πληροφορίες περιεχομένου της κάθε εικόνας. Επιπλέον, οι προορισμοί αυτοί μπορούν να ταξινομούνται κατά σειρά εκδήλωσης ενδιαφέροντος, το οποίο προκύπτει από τον αριθμό των εικόνων που έχει ανεβάσει ο χρήστης από κάθε προορισμό.

ПАРАРТНМА

Στη συνέχεια παρατίθεται αναλυτικά ο κώδικας από τα όλα τα επιμέρους αρχεία της εφαρμογής, τα οποία για διευκόλυνση της αναγνωσιμότητας του κειμένου δεν παρουσιάστηκαν ολοκληρωμένα στα παραπάνω κεφάλαια. Αρχικά παρουσιάζονται οι κλάσεις της εφαρμογής (σε αλφαβητική σειρά), έπειτα τα αρχεία xml τα οποία βρίσκονται στο φάκελο res/layout/ και ορίζουν τη διεπαφή χρήστη για κάθε activity, τα αρχεία strings.xml για αγγλική και ελληνική γλώσσα, και τέλος το AndroidManifest.xml.

AboutScreen.java

```
package com.ntua.image.viral;

import android.app.Activity;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Color;
import android.os.Bundle;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

public class AboutScreen extends Activity {

	Button buttonCamera;
	Button buttonUpload;
	Button buttonLogin;
	Button buttonHistory;
	Button buttonAbout;
	LinearLayout mainLay;

	@Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.aboutscreen);

      mainLay = (LinearLayout) findViewById(R.id.main);
      mainLay.setVisibility(View.GONE);

      TextView aboutText = (TextView)findViewById(R.id.aboutText);
      String link = "http://viral.image.ntua.gr";
      String htmlText = getString(R.string.aboutText) + " " + "<a
href=\""+link+"\">"+getString(R.string.viral20)+"</a>";
      aboutText.setText(Html.fromHtml(htmlText));
      aboutText.setMovementMethod(LinkMovementMethod.getInstance());
      aboutText.setLinksClickable(true);
   }

	@Override
   public void onConfigurationChanged(Configuration newConfig) {
      super.onConfigurationChanged(newConfig);
   }

   @Override
	public boolean onPrepareOptionsMenu(Menu menu) {
```

```
                    if (mainLay.getVisibility()==View.GONE) {
                            //set visible & enable buttons
                            mainLay.setVisibility(View.VISIBLE);

                            buttonCamera = (Button) findViewById(R.id.buttonCamera);
                        buttonCamera.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                            Intent i = new Intent (AboutScreen.this, HomeScreen.class);
                            i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                            startActivity (i);
                           overridePendingTransition(0,0);
                          }
                        });

                        buttonUpload = (Button) findViewById(R.id.buttonUpload);
                        buttonUpload.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                            Intent i = new Intent (AboutScreen.this, UploadScreen.class);
                            i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                            startActivity (i);
                           overridePendingTransition(0,0);
                          }
                        });

                        buttonLogin = (Button) findViewById(R.id.buttonLogin);
                        buttonLogin.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                            Intent i = new Intent (AboutScreen.this, LoginScreen.class);
                            i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                            startActivity (i);
                           overridePendingTransition(0,0);
                          }
                        });

                        buttonHistory = (Button) findViewById(R.id.buttonHistory);
                        buttonHistory.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                            Intent i = new Intent (AboutScreen.this, HistoryScreen.class);
                            i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                            startActivity (i);
                           overridePendingTransition(0,0);
                          }
                        });

                        buttonAbout = (Button) findViewById(R.id.buttonAbout);
                        buttonAbout.getBackground().setColorFilter(Color.rgb(95, 252, 17),
android.graphics.PorterDuff.Mode.MULTIPLY);

                    }
                    else {
                            mainLay.setVisibility(View.GONE);
                    }
                    return true;
            }

    @Override
    public void onBackPressed() {
            finish();
            overridePendingTransition(0,0);
    }
}
```

Base64Coder.java

```
package com.ntua.image.viral;

public class Base64Coder {

private static final String systemLineSeparator = System.getProperty("line.separator");

private static final char[] map1 = new char[64];
static {
   int i=0;
   for (char c='A'; c<='Z'; c++) map1[i++] = c;
   for (char c='a'; c<='z'; c++) map1[i++] = c;
   for (char c='0'; c<='9'; c++) map1[i++] = c;
   map1[i++] = '+'; map1[i++] = '/'; }

private static final byte[] map2 = new byte[128];
static {
   for (int i=0; i<map2.length; i++) map2[i] = -1;
   for (int i=0; i<64; i++) map2[map1[i]] = (byte)i; }

public static String encodeString (String s) {
return new String(encode(s.getBytes())); }

public static String encodeLines (byte[] in) {
return encodeLines(in, 0, in.length, 76, systemLineSeparator); }

public static String encodeLines (byte[] in, int iOff, int iLen, int lineLen, String lineSeparator) {
int blockLen = (lineLen*3) / 4;
if (blockLen <= 0) throw new IllegalArgumentException();
int lines = (iLen+blockLen-1) / blockLen;
int bufLen = ((iLen+2)/3)*4 + lines*lineSeparator.length();
StringBuilder buf = new StringBuilder(bufLen);
int ip = 0;
while (ip < iLen) {
   int l = Math.min(iLen-ip, blockLen);
   buf.append (encode(in, iOff+ip, l));
   buf.append (lineSeparator);
   ip += l; }
return buf.toString(); }

public static char[] encode (byte[] in) {
return encode(in, 0, in.length); }

public static char[] encode (byte[] in, int iLen) {
return encode(in, 0, iLen); }

public static char[] encode (byte[] in, int iOff, int iLen) {
int oDataLen = (iLen*4+2)/3;
int oLen = ((iLen+2)/3)*4;
char[] out = new char[oLen];
int ip = iOff;
int iEnd = iOff + iLen;
int op = 0;
while (ip < iEnd) {
   int i0 = in[ip++] & 0xff;
   int i1 = ip < iEnd ? in[ip++] & 0xff : 0;
   int i2 = ip < iEnd ? in[ip++] & 0xff : 0;
   int o0 = i0 >>> 2;
```

```
   int o1 = ((i0 &   3) << 4) | (i1 >>> 4);
   int o2 = ((i1 & 0xf) << 2) | (i2 >>> 6);
   int o3 = i2 & 0x3F;
   out[op++] = map1[o0];
   out[op++] = map1[o1];
   out[op] = op < oDataLen ? map1[o2] : '='; op++;
   out[op] = op < oDataLen ? map1[o3] : '='; op++; }
return out; }

public static String decodeString (String s) {
return new String(decode(s)); }

public static byte[] decodeLines (String s) {
char[] buf = new char[s.length()];
int p = 0;
for (int ip = 0; ip < s.length(); ip++) {
   char c = s.charAt(ip);
   if (c != ' ' && c != '\r' && c != '\n' && c != '\t')
      buf[p++] = c; }
return decode(buf, 0, p); }

public static byte[] decode (String s) {
return decode(s.toCharArray()); }

public static byte[] decode (char[] in) {
return decode(in, 0, in.length); }

public static byte[] decode (char[] in, int iOff, int iLen) {
if (iLen%4 != 0) throw new IllegalArgumentException ("Length of Base64 encoded input string is not a
multiple of 4.");
while (iLen > 0 && in[iOff+iLen-1] == '=') iLen--;
int oLen = (iLen*3) / 4;
byte[] out = new byte[oLen];
int ip = iOff;
int iEnd = iOff + iLen;
int op = 0;
while (ip < iEnd) {
   int i0 = in[ip++];
   int i1 = in[ip++];
   int i2 = ip < iEnd ? in[ip++] : 'A';
   int i3 = ip < iEnd ? in[ip++] : 'A';
   if (i0 > 127 || i1 > 127 || i2 > 127 || i3 > 127)
      throw new IllegalArgumentException ("Illegal character in Base64 encoded data.");
   int b0 = map2[i0];
   int b1 = map2[i1];
   int b2 = map2[i2];
   int b3 = map2[i3];
   if (b0 < 0 || b1 < 0 || b2 < 0 || b3 < 0)
      throw new IllegalArgumentException ("Illegal character in Base64 encoded data.");
   int o0 = ( b0       <<2) | (b1>>>4);
   int o1 = ((b1 & 0xf)<<4) | (b2>>>2);
   int o2 = ((b2 &   3)<<6) |  b3;
   out[op++] = (byte)o0;
   if (op<oLen) out[op++] = (byte)o1;
   if (op<oLen) out[op++] = (byte)o2; }
return out; }

private Base64Coder() {}

}
```

CustomItemizedOverlay.java

```
package com.ntua.image.viral;

import java.util.ArrayList;
import android.app.AlertDialog;
import android.content.Context;
import android.graphics.drawable.Drawable;

import com.google.android.maps.ItemizedOverlay;
import com.google.android.maps.OverlayItem;

public class CustomItemizedOverlay extends ItemizedOverlay<OverlayItem> {

  private ArrayList<OverlayItem> mapOverlays = new ArrayList<OverlayItem>();
  private Context context;

  public CustomItemizedOverlay(Drawable defaultMarker) {
      super(boundCenterBottom(defaultMarker));
  }

  public CustomItemizedOverlay(Drawable defaultMarker, Context context) {
      this(defaultMarker);
      this.context = context;
  }

  @Override
  protected OverlayItem createItem(int i) {
    return mapOverlays.get(i);
  }

  @Override
  public int size() {
    return mapOverlays.size();
  }

  @Override
  protected boolean onTap(int index) {
    OverlayItem item = mapOverlays.get(index);
    AlertDialog.Builder dialog = new AlertDialog.Builder(context);
    dialog.setTitle(item.getTitle());
    dialog.setMessage(item.getSnippet());
    dialog.show();
    return true;
  }

  public void addOverlay(OverlayItem overlay) {
    mapOverlays.add(overlay);
    this.populate();
  }

}
```

HistoryMapScreen.java

```
package com.ntua.image.viral;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
```

```
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.OverlayItem;

import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.ZoomControls;

public class HistoryMapScreen extends MapActivity {

        Button buttonCamera;
        Button buttonUpload;
        Button buttonLogin;
        Button buttonHistory;
        Button buttonAbout;
        LinearLayout mainLay;
        MapView mapView;
        ZoomControls zoomControls;
        MapController mc;
        ProgressDialog loadingDialog;
        LoadingTask loadTask=null;
        ArrayList<String> resultList;
        ArrayList<byte[]> imageList;
        JSONObject jObject, locationObj;
        JSONArray landmarksArr;
        double latitude,longitude;
        GeoPoint p;
        String city, country;

        @Override
   public void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
    setContentView(R.layout.historymapscreen);

    mainLay = (LinearLayout) findViewById(R.id.main);
    mainLay.setVisibility(View.GONE);

    loadingDialog = new ProgressDialog(this);
                loadingDialog.setMessage(getString(R.string.loadingMap));
                loadingDialog.setCancelable(true);
                loadingDialog.setOnCancelListener(new DialogInterface.OnCancelListener(){
      public void onCancel(DialogInterface dialog) {
        try {
                        loadTask.cancel(false);
                        finish();
                        overridePendingTransition(0,0);
        }
        catch (NullPointerException e) {
                        e.printStackTrace();
        }
      }
    });

                if (checkCookie()) {
                        loadTask = new LoadingTask();
                        loadTask.execute();
                }

  }

        private class LoadingTask extends AsyncTask<Void, Void, Integer> {

    protected void onPreExecute() {
        loadingDialog.show();
    }

    protected Integer doInBackground(Void... params) {
        if (!loadTask.isCancelled()) {
                resultList=new ArrayList<String>();
                imageList=new ArrayList<byte[]>();
                try {
                        Class.forName("org.postgresql.Driver");
                                String url = "jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/xxxxx";
                                String user = "xxxxx";
                                String pass = "xxxxx";
                                Connection conn = DriverManager.getConnection(url, user, pass);
                                String username = findUsername();
                                String query = "SELECT id, query, result FROM android_history
WHERE \"user\"='"+username+"' ORDER BY id DESC";
                                Statement st = conn.createStatement();
                                ResultSet rs = st.executeQuery(query);
                                while (rs.next()) {
                                        imageList.add(rs.getBytes("query"));

        resultList.add(Base64Coder.decodeString(rs.getString("result")));
                                }
                                rs.close();
                                st.close();
                                conn.close();
                } catch (ClassNotFoundException e) {
                                        e.printStackTrace();
                                        loadTask.cancel(false);
```

```
                                        loadingDialog.dismiss();
                                } catch (SQLException e) {
                                        e.printStackTrace();
                                        loadTask.cancel(false);
                                        loadingDialog.dismiss();

                                }
        }
                        return null;
}

protected void onPostExecute(Integer a) {
        if (!loadTask.isCancelled()) {
                int length = imageList.size();
                int i;
                Bitmap bitmap;

                //Displaying map
         mapView = (MapView) findViewById(R.id.historyMapView);
         mc = mapView.getController();
         //ZOOM buttons:
         zoomControls = (ZoomControls) findViewById(R.id.historyzoomcontrols);
         zoomControls.setOnZoomInClickListener(new View.OnClickListener() {
               public void onClick(View v) {
                     mc.zoomIn();
               }
         });
         zoomControls.setOnZoomOutClickListener(new View.OnClickListener() {
               public void onClick(View v) {
                     mc.zoomOut();
               }
         });

         mc.setZoom(1);

         //Add location markers (images)
         List<Overlay> mapOverlays = mapView.getOverlays();

         Drawable drawable;
         CustomItemizedOverlay itemizedOverlay;
         OverlayItem overlayitem;
         BitmapFactory.Options options = new BitmapFactory.Options();
                        options.inSampleSize=16;
                for (i=0; i<length; i++) {
                        final byte[] data = imageList.get(i);
                        final String result = resultList.get(i);
                        if (result.equals("-1") || (result.contains("{\"images\":[]"))) {
                //den exei vrei similar images edw ara mhn emfaniseis eikona sto xarth
                        } else {
                        bitmap = BitmapFactory.decodeByteArray(data, 0, data.length, options);
                        //Getting latitude & longitude
            try {
                                jObject = new JSONObject(result);
                                locationObj = jObject.getJSONObject("location");
                                latitude = locationObj.getDouble("latitude");
                                longitude = locationObj.getDouble("longitude");
                        } catch (JSONException e) {
                                e.printStackTrace();
                        }
                        //Getting city & country
            try {
```

```
                                        city = jObject.getString("city");
                                        country = jObject.getString("country");
                                } catch (JSONException e) {
                                        e.printStackTrace();
                                }
                                //Getting landmarks
                                String suggestedTags="";
                        String name="";
                                int j=0;
                                boolean found=false;
                                try {
                                        landmarksArr = jObject.getJSONArray("landmarks");
                                        int landmarksArrLength = landmarksArr.length();
                                        while (j<landmarksArrLength && j<4) {  //thelw ews 4 suggested
tags
                                                name =
landmarksArr.getJSONObject(j).getString("name");
                                                if (!name.equals(city)) {
                                                        //emfanizw landmark mono an einai diaforetiko
apo thn polh
                                                        suggestedTags = suggestedTags + name + ", ";
                                                        found=true;
                                                }
                                                j++;
                                        }
                                } catch (JSONException e) {
                                        e.printStackTrace();
                                }
                                if (!found) {
                                        //de vrhke landmarks, ara svhse to string Suggested tags
                                        suggestedTags="";
                                } else {
                                        //vrhke landmarks
                                        //afairw tous 2 teleutaious xarakthres ", " apo to string
                                        suggestedTags = suggestedTags.substring(0,
suggestedTags.length()-2);
                                }

                                p = new GeoPoint(
                                  (int) (latitude * 1E6),
                                  (int) (longitude * 1E6));
                                drawable =new BitmapDrawable(bitmap);
                        itemizedOverlay = new CustomItemizedOverlay(drawable,
HistoryMapScreen.this);

                                overlayitem = new OverlayItem(p, city+", "+country, suggestedTags);
                                itemizedOverlay.addOverlay(overlayitem);

                                mapOverlays.add(itemizedOverlay);
                                }

                        }
                        mapView.invalidate();
                        loadingDialog.dismiss();
                }
        }

    }

        @Override
   public void onConfigurationChanged(Configuration newConfig) {
```

```
    super.onConfigurationChanged(newConfig);
}

@Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                if (mainLay.getVisibility()==View.GONE) {
                        //set visible & enable buttons
                        mainLay.setVisibility(View.VISIBLE);

                        buttonCamera = (Button) findViewById(R.id.buttonCamera);
                    buttonCamera.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (HistoryMapScreen.this, HomeScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonUpload = (Button) findViewById(R.id.buttonUpload);
                    buttonUpload.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (HistoryMapScreen.this, UploadScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonLogin = (Button) findViewById(R.id.buttonLogin);
                    buttonLogin.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (HistoryMapScreen.this, LoginScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonHistory = (Button) findViewById(R.id.buttonHistory);
                    buttonHistory.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (HistoryMapScreen.this, HistoryScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonAbout = (Button) findViewById(R.id.buttonAbout);
                    buttonAbout.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (HistoryMapScreen.this, AboutScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                }
                else {
```

```
                        mainLay.setVisibility(View.GONE);
                    }
                    return true;
        }

    public boolean checkCookie() {
        File f = new File(getCacheDir()+"Cookie.viral");
        if (f.exists()) {
                return true;
        } else {
                return false;
        }
    }

    public String findUsername() {
        String user="";
                BufferedReader in;
                try {
                        in = new BufferedReader(new FileReader(getCacheDir()+"Cookie.viral"));
                        user=in.readLine();
                } catch (FileNotFoundException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                }
                return user;
    }

    @Override
    protected boolean isRouteDisplayed() {
       return false;
    }

    @Override
    public void onBackPressed() {
        finish();
        overridePendingTransition(0,0);
    }
}
```

HistoryScreen.java

```
package com.ntua.image.viral;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
```

```
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Gravity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Toast;

public class HistoryScreen extends Activity {

        Button buttonCamera;
        Button buttonUpload;
        Button buttonLogin;
        Button buttonHistory;
        Button buttonAbout;
        LinearLayout mainLay;
        ProgressDialog loadingDialog, clearingDialog;
        LoadingTask loadTask=null;
        ClearingTask clearTask=null;
        ArrayList<String> resultList;
        ArrayList<byte[]> imageList;
        float scale;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.historyscreen);

    mainLay = (LinearLayout) findViewById(R.id.main);
    mainLay.setVisibility(View.GONE);

    scale = getResources().getDisplayMetrics().density;

    if (!checkCookie()) {
        //den exei syndethei ara
        //emfanise toast oti den exei syndethei
        //kai phgaine sth LoginScreen
        Context context = getApplicationContext();
                        CharSequence text = getString(R.string.pleaseLogin);
                        int duration = Toast.LENGTH_LONG;
                        Toast toast = Toast.makeText(context, text, duration);
                        toast.setGravity(Gravity.CENTER, 0, 0);
                        toast.show();
                        Intent i = new Intent (HistoryScreen.this, LoginScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                        overridePendingTransition(0,0);
                        finish();
```

```
	} else {
		//exei syndethei
		//elegxw an exei internet gia na katevasei to history
		if (!networkAvailable()) {
			//den exei internet
			Context context = getApplicationContext();
				CharSequence text = getString(R.string.checkConnection);
				int duration = Toast.LENGTH_SHORT;
				Toast toast = Toast.makeText(context, text, duration);
				toast.setGravity(Gravity.CENTER, 0, 0);
				toast.show();
				finish();
				overridePendingTransition(0,0);
		} else {
			//exei internet ara emfanizw eikones
			loadingDialog = new ProgressDialog(this);
			loadingDialog.setMessage(getString(R.string.loadDialog));
			loadingDialog.setCancelable(true);
			loadingDialog.setOnCancelListener(new DialogInterface.OnCancelListener(){
		    public void onCancel(DialogInterface dialog) {
		      try {
				loadTask.cancel(false);
				finish();
				overridePendingTransition(0,0);
		      }
		      catch (NullPointerException e) {
				e.printStackTrace();
		      }
		    }
		  });

			loadTask = new LoadingTask();
			loadTask.execute();

			//button history map
			Button buttonHistoryMap = (Button) findViewById(R.id.buttonHistoryMap);
			buttonHistoryMap.setOnClickListener(new OnClickListener() {
				public void onClick(View v) {
					Intent i = new Intent (HistoryScreen.this, HistoryMapScreen.class);
					i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
					startActivity (i);
					overridePendingTransition(0,0);
				}
			});

			//button clear history
			Button buttonClearHistory = (Button) findViewById(R.id.buttonClearHistory);
			buttonClearHistory.setOnClickListener(new OnClickListener() {
				public void onClick(View v) {
				  //emfanizw alert dialog
				  AlertDialog.Builder builder = new AlertDialog.Builder(HistoryScreen.this);
				  builder.setMessage(getString(R.string.askClearHistory))
					.setCancelable(true)
					.setPositiveButton(getString(R.string.yes), new
DialogInterface.OnClickListener() {
						public void onClick(DialogInterface dialog, int id) {
							dialog.cancel();
							clearingDialog = new
ProgressDialog(HistoryScreen.this);

clearingDialog.setMessage(getString(R.string.clearingDialog));
```

```
                                                clearingDialog.setCancelable(true);
                                                clearingDialog.setOnCancelListener(new
DialogInterface.OnCancelListener(){
                                                    public void onCancel(DialogInterface dialog) {
                                                        try {
                                                                clearTask.cancel(false);
                                                        }
                                                        catch (NullPointerException e) {
                                                                e.printStackTrace();
                                                        }
                                                    }
                                                });
                                                clearTask = new ClearingTask();
                                                clearTask.execute();
                                            }
                                        })
                                        .setNegativeButton(getString(R.string.cancel), new
DialogInterface.OnClickListener() {
                                            public void onClick(DialogInterface dialog, int id) {
                                                dialog.cancel();
                                            }
                                        });
                                AlertDialog alert = builder.create();
                                alert.show();

                            }
                        });
            }
        }
    }

    public boolean checkCookie() {
            File f = new File(getCacheDir()+"Cookie.viral");
            if (f.exists()) {
                    return true;
            } else {
                    return false;
            }
    }

    public boolean networkAvailable() {
            ConnectivityManager cn = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
            NetworkInfo ni = cn.getActiveNetworkInfo();
            if (ni==null)
               return false;
            return ni.isConnected();
    }

    public String findUsername() {
            String user="";
                    BufferedReader in;
                    try {
                            in = new BufferedReader(new FileReader(getCacheDir()+"Cookie.viral"));
                            user=in.readLine();
                    } catch (FileNotFoundException e) {
                            e.printStackTrace();
                    } catch (IOException e) {
                            e.printStackTrace();
                    }
                    return user;
```

```
   }

   private class LoadingTask extends AsyncTask<Void, Void, Integer> {

      protected void onPreExecute() {
          loadingDialog.show();
      }

      protected Integer doInBackground(Void... params) {
          if (!loadTask.isCancelled()) {
                    resultList=new ArrayList<String>();
                    imageList=new ArrayList<byte[]>();
                    try {
                              Class.forName("org.postgresql.Driver");
                                        String url = "jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/xxxxx";
                                        String user = "xxxxx";
                                        String pass = "xxxxx";
                                        Connection conn = DriverManager.getConnection(url, user, pass);
                                        String username = findUsername();
                                        String query = "SELECT id, query, result FROM android_history
WHERE \"user\"='"+username+"' ORDER BY id DESC";
                                        Statement st = conn.createStatement();
                                        ResultSet rs = st.executeQuery(query);
                                        while (rs.next()) {
                                                  imageList.add(rs.getBytes("query"));

          resultList.add(Base64Coder.decodeString(rs.getString("result")));
                                        }
                                        rs.close();
                                        st.close();
                                        conn.close();
                    } catch (ClassNotFoundException e) {
                                                  e.printStackTrace();
                                                  loadTask.cancel(false);
                                                  loadingDialog.dismiss();
                                        } catch (SQLException e) {
                                                  e.printStackTrace();
                                                  loadTask.cancel(false);
                                                  loadingDialog.dismiss();

                                        }
          }
                              return null;
      }

      protected void onPostExecute(Integer a) {
          if (!loadTask.isCancelled()) {
                    int length = imageList.size();
                    int i;
                    LinearLayout history1 = (LinearLayout) findViewById(R.id.history1);
                    Bitmap bitmap;
                    BitmapFactory.Options options = new BitmapFactory.Options();
                    options.inSampleSize=4;
                    LinearLayout row = new LinearLayout(HistoryScreen.this);
                    row.setLayoutParams(new LayoutParams(
                      LayoutParams.FILL_PARENT,
                      LayoutParams.WRAP_CONTENT));
                    row.setWeightSum(4);
                    int px1 = (int) (75 * scale + 0.5f); //75 dp in pixels
          LinearLayout.LayoutParams Params = new LinearLayout.LayoutParams(px1, px1); //image
params
```

```
        int px2 = (int) (1 * scale + 0.5f); //1 dp in pixels
        LayoutParams rowParams = new LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT);
                for (i=0; i<length; i++) {
                        final byte[] data = imageList.get(i);
                        final String result = resultList.get(i);
                        bitmap = BitmapFactory.decodeByteArray(data, 0, data.length, options);
                        ImageView im = new ImageView (HistoryScreen.this);
                        im.setImageBitmap(bitmap);
            im.setLayoutParams(Params);
            im.setAdjustViewBounds(true);
            im.setPadding(px2, px2, px2, px2);

            if (i%4==0 && i!=0) {
                //apothikeuw thn palia seira
                history1.addView(row);
                //allazw seira
                LinearLayout row1 = new LinearLayout(HistoryScreen.this);
                row=row1;
                        row.setLayoutParams(rowParams);
                        row.setWeightSum(4);
            }
            row.addView(im);
            if (i==length-1) {
                //einai to teleutaio image ara prepei na prosthesoume th seira sto linear layout
                history1.addView(row);
            }
            im.setOnClickListener(new OnClickListener() {
                        public void onClick(View v) {
                            //apothikeush bitmap sthn cache
                            File f = new File(getCacheDir()+"viral_image.jpg");
                            FileOutputStream out;
                            try {
                                    out = new FileOutputStream(f);
                                    BitmapFactory.decodeByteArray(data, 0,
data.length).compress(Bitmap.CompressFormat.JPEG, 90, out);
                                    out.close();
                            } catch (FileNotFoundException e1) {
                                    e1.printStackTrace();
                            } catch (IOException e2) {
                                    e2.printStackTrace();
                            }
                            //anoikse ResultsScreen activity
                            Intent in = new Intent (HistoryScreen.this, ResultsScreen.class);
                            in.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                            in.putExtra("result", result);
                            startActivity (in);
                           overridePendingTransition(0,0);
                        }
            });
                }
                loadingDialog.dismiss();
        }
    }
  }

  private class ClearingTask extends AsyncTask<Void, Void, Integer> {

        protected void onPreExecute() {
        clearingDialog.show();
    }
```

```
        protected Integer doInBackground(Void... params) {
        if (!clearTask.isCancelled()) {
                try {
                                        Class.forName("org.postgresql.Driver");
                                        String url =
"jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/xxxxx";
                                        String user = "xxxxx";
                                        String pass = "xxxxx";
                                        Connection conn = DriverManager.getConnection(url, user,
pass);
                                        String update = "DELETE FROM ONLY android_history
WHERE \"user\"='"+findUsername()+"'";
                                        Statement st = conn.createStatement();
                                        st.executeUpdate(update);
                                        st.close();
                                        conn.close();
                                } catch (ClassNotFoundException e) {
                                        e.printStackTrace();
                                        clearTask.cancel(false);
                                        clearingDialog.dismiss();
                                } catch (SQLException e) {
                                        e.printStackTrace();
                                        clearTask.cancel(false);
                                        clearingDialog.dismiss();
                                }
        }
        return 1;
        }

        protected void onPostExecute(Integer a) {
        if (!clearTask.isCancelled()) {
                clearingDialog.dismiss();
                Intent i = new Intent (HistoryScreen.this, HistoryScreen.class);
                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity (i);
             overridePendingTransition(0,0);
        }
        }

   }

   @Override
   public void onConfigurationChanged(Configuration newConfig) {
      super.onConfigurationChanged(newConfig);
   }

   @Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                  if (mainLay.getVisibility()==View.GONE) {
                        //set visible & enable buttons
                        mainLay.setVisibility(View.VISIBLE);

                        buttonCamera = (Button) findViewById(R.id.buttonCamera);
                    buttonCamera.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (HistoryScreen.this, HomeScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
```

```
                        });

                        buttonUpload = (Button) findViewById(R.id.buttonUpload);
                        buttonUpload.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (HistoryScreen.this, UploadScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                              overridePendingTransition(0,0);
                          }
                        });

                        buttonLogin = (Button) findViewById(R.id.buttonLogin);
                        buttonLogin.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (HistoryScreen.this, LoginScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                              overridePendingTransition(0,0);
                          }
                        });

                        buttonHistory = (Button) findViewById(R.id.buttonHistory);
                        buttonHistory.getBackground().setColorFilter(Color.rgb(95, 252, 17),
android.graphics.PorterDuff.Mode.MULTIPLY);

                        buttonAbout = (Button) findViewById(R.id.buttonAbout);
                        buttonAbout.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (HistoryScreen.this, AboutScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                              overridePendingTransition(0,0);
                          }
                        });

                    }
                    else {
                            mainLay.setVisibility(View.GONE);
                    }
                    return true;
            }

    @Override
    public void onBackPressed() {
            finish();
            overridePendingTransition(0,0);
    }
}
```

HomeScreen.java

```
package com.ntua.image.viral;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.content.Intent;
```

```
import android.graphics.Color;
import android.hardware.Camera;
import android.hardware.Camera.Parameters;
import android.hardware.Camera.PictureCallback;
import android.hardware.Camera.Size;
import android.os.Build;
import android.os.Bundle;
import android.text.TextUtils;
import android.text.TextUtils.StringSplitter;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.view.Menu;
import android.view.MotionEvent;
import android.view.SurfaceHolder;
import android.view.SurfaceView;

public class HomeScreen extends Activity implements SurfaceHolder.Callback {

	Button buttonClick;
	ImageView buttonFlash;
	Button buttonZoomIn;
	Button buttonZoomOut;
	Button buttonCamera;
	Button buttonUpload;
	Button buttonLogin;
	Button buttonHistory;
	Button buttonAbout;
	LinearLayout mainLayout;
	public Camera mCam;
	boolean flashMode=false;
	SurfaceView cameraSurface;
	boolean isPressed=false;

	@Override
	public void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.homescreen);

		cameraSurface = (SurfaceView)findViewById(R.id.camerasurface);
		SurfaceHolder cameraHolder = cameraSurface.getHolder();
		cameraHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
		cameraHolder.addCallback(this);

		mainLayout = (LinearLayout) findViewById(R.id.mainLayout);

		buttonClick = (Button) findViewById(R.id.buttonClick);
		buttonClick.setOnClickListener(new OnClickListener() {
			public void onClick(View v) {
				if (!isPressed) {
					isPressed=true;
					AutoFocusCallBackImpl autoFocusCallBack = new
AutoFocusCallBackImpl();
					mCam.autoFocus(autoFocusCallBack);
				}
			}
		});

		buttonFlash = (ImageView) findViewById(R.id.buttonFlash);
```

```
                buttonFlash.setOnClickListener(new OnClickListener() {
                        public void onClick(View v) { // <5>
                                Parameters params = mCam.getParameters();
                                if (params.getFlashMode()!=null) {
                                        //flash is supported
                                        if (!flashMode) {

        params.setFlashMode(Camera.Parameters.FLASH_MODE_ON);
                                                flashMode=true;

        buttonFlash.setImageDrawable(getResources().getDrawable(R.drawable.flashon));
                                        }
                                        else {

        params.setFlashMode(Camera.Parameters.FLASH_MODE_OFF);
                                                flashMode=false;

        buttonFlash.setImageDrawable(getResources().getDrawable(R.drawable.flashoff));
                                        }
                                        mCam.setParameters(params);
                                }
                        }
                });

                buttonZoomIn = (Button) findViewById(R.id.buttonZoomIn);
                buttonZoomIn.setOnClickListener(new OnClickListener() {
                        public void onClick(View v) {
                                Parameters params = mCam.getParameters();
                                try {
                                        params.set("taking-picture-zoom", params.getInt("taking-
picture-zoom")+3);
                                        mCam.setParameters(params);
                                } catch (Exception e) {
                                        //zoom not supported or max zoom reached, do nothing
                                }
                        }
                });

                buttonZoomOut = (Button) findViewById(R.id.buttonZoomOut);
                buttonZoomOut.setOnClickListener(new OnClickListener() {
                        public void onClick(View v) {
                                Parameters params = mCam.getParameters();
                                try {
                                        params.set("taking-picture-zoom", params.getInt("taking-
picture-zoom")-3);
                                        mCam.setParameters(params);
                                } catch (Exception e) {
                                        //zoom not supported or min zoom reached, do nothing
                                }
                        }
                });

                mainLayout.setVisibility(View.GONE);

        }

        public void surfaceCreated(SurfaceHolder holder) {
                        // Surface created, now it is possible to set the preview
                        try {
                                mCam = Camera.open();
                                mCam.setPreviewDisplay(holder);
```

```
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
        }

        public void surfaceDestroyed(SurfaceHolder holder) {
                        mCam.stopPreview();
                        mCam.release();
        }

        public void surfaceChanged(SurfaceHolder holder, int I, int J, int K) {
                        Camera.Parameters parameters = mCam.getParameters();

                        List<Size> previewSizes;
                        if (Build.VERSION.SDK_INT > Build.VERSION_CODES.ECLAIR_MR1)
{
                                //android 2.2 and over
                                previewSizes = parameters.getSupportedPreviewSizes();
                        } else {
                                //android 2.1 and under
                                String supportedSizesString = parameters.get("preview-size-
values");
                                previewSizes = new ArrayList<Size>();
                                if (supportedSizesString != null && supportedSizesString.length() >
0) {
                                   StringSplitter ss = new TextUtils.SimpleStringSplitter(',');
                                   ss.setString(supportedSizesString);
                                   for (String supportedSize : ss) {
                                      String[] dimensions = supportedSize.split("x");
                                      if (dimensions.length > 1) {
                                         Size size = mCam.new Size(Integer.parseInt(dimensions[0]),
                                         Integer.parseInt(dimensions[1]));
                                         previewSizes.add(size);
                                      }
                                   }
                                }
                        }
                        Size bestSize = null;
                        int bestDiff = 0;
                        int diff = 0;
                        for (Size size : previewSizes) {
                                        diff = Math.abs(K - size.height) + Math.abs(J - size.width);
                                        if (bestSize == null || diff < bestDiff) {
                                                bestSize = size;
                                                bestDiff = diff;
                                        }
                                        parameters.setPreviewSize(bestSize.width,
bestSize.height);
                                        mCam.setParameters(parameters);
                        }

                        //start preview of camera
                        mCam.startPreview();
        }

        //Kaleitai otan paroume fwtografia
        PictureCallback jpegCallback = new PictureCallback() {
                        public void onPictureTaken(byte[] data, Camera camera) {

                                Intent i = new Intent (HomeScreen.this, SendScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
```

```
                                        i.putExtra("dataHome", data);
                                        startActivity (i);
                                        overridePendingTransition(0,0);
                                        isPressed=false;
                                }
        };

        private class AutoFocusCallBackImpl implements Camera.AutoFocusCallback {
                        public void onAutoFocus(boolean success, Camera camera) {
                           //Called when the camera auto focus completes.
                                //If the camera does not support auto-focus and autoFocus is called,
                                //onAutoFocus will be called immediately with a fake value of success set to
true.
                           mCam.takePicture(null, null, jpegCallback);
                        }
        }

        @Override
        public boolean onSearchRequested() {
                                if (!isPressed) {
                                        isPressed=true;
                                        AutoFocusCallBackImpl autoFocusCallBack = new
AutoFocusCallBackImpl();
                                        mCam.autoFocus(autoFocusCallBack);

                                }
                                return true;
        }

        @Override
        public boolean onTrackballEvent (MotionEvent event) {
                                if (event.getAction() == MotionEvent.ACTION_DOWN) {
                                        if (!isPressed) {
                                                isPressed=true;
                                                AutoFocusCallBackImpl autoFocusCallBack = new
AutoFocusCallBackImpl();
                                                mCam.autoFocus(autoFocusCallBack);
                                        }
                                }
                                return true;
        }


        @Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                        if (mainLayout.getVisibility()==View.GONE) {
                                //set visible & enable buttons
                                mainLayout.setVisibility(View.VISIBLE);

                                buttonCamera = (Button) findViewById(R.id.buttonCamera);
                                  buttonCamera.getBackground().setColorFilter(Color.rgb(95, 252, 17),
android.graphics.PorterDuff.Mode.MULTIPLY);

                                  buttonUpload = (Button) findViewById(R.id.buttonUpload);
                                  buttonUpload.setOnClickListener(new OnClickListener() {
                                    public void onClick(View v) {
                                        Intent i = new Intent (HomeScreen.this, UploadScreen.class);
                                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                        startActivity (i);
                                    overridePendingTransition(0,0);
                                    }
```

```
                        });

                        buttonLogin = (Button) findViewById(R.id.buttonLogin);
                        buttonLogin.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (HomeScreen.this, LoginScreen.class);
                             i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                             overridePendingTransition(0,0);
                          }
                        });

                        buttonHistory = (Button) findViewById(R.id.buttonHistory);
                        buttonHistory.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (HomeScreen.this, HistoryScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                             overridePendingTransition(0,0);
                          }
                        });

                        buttonAbout = (Button) findViewById(R.id.buttonAbout);
                        buttonAbout.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (HomeScreen.this, AboutScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                             overridePendingTransition(0,0);
                          }
                        });

                }
                else {
                        mainLayout.setVisibility(View.GONE);
                }
                return true;
        }
}
```

ImageFullScreen.java

```
package com.ntua.image.viral;

import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;

import android.app.Activity;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
```

```
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;

public class ImageFullScreen extends Activity {

        Button buttonCamera;
        Button buttonUpload;
        Button buttonLogin;
        Button buttonHistory;
        Button buttonAbout;
        LinearLayout mainLay;
        String imageUrl;
        LoadingTask loadTask=null;
        ImageView imageViewFullScreen;
        ProgressBar progressBarLarge;

        @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.imagefullscreen);

      mainLay = (LinearLayout) findViewById(R.id.main);
      mainLay.setVisibility(View.GONE);

      Bundle bundle = getIntent().getExtras();
      imageUrl = bundle.getString("imageUrl");

      imageUrl = imageUrl.replace("_s.jpg", ".jpg");

      progressBarLarge = (ProgressBar) findViewById(R.id.progressBarLarge);

      imageViewFullScreen = (ImageView) findViewById(R.id.imageViewFullScreen);

                  ImageView imageViewFlickr = (ImageView) findViewById(R.id.imageViewFlickr);
                  imageViewFlickr.setOnClickListener(new OnClickListener() {
               public void onClick(View v) {
                    Intent downloadIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(imageUrl));
                    startActivity(downloadIntent);
               }
          });

                  loadTask = new LoadingTask();
        loadTask.execute();
        }

        private class LoadingTask extends AsyncTask<Void, Void, Bitmap> {

      protected void onPreExecute() {
          //progress bar is displayed
      }

      protected Bitmap doInBackground(Void... params) {

         Bitmap fullScreenBitmap=null;
          if (!loadTask.isCancelled()) {
                  URL url=null;
                  try {
                          url = new URL(imageUrl);
                          fullScreenBitmap =
BitmapFactory.decodeStream(url.openConnection().getInputStream());
```

```
                } catch (MalformedURLException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }
                        return fullScreenBitmap;
    }

    protected void onPostExecute(Bitmap bm) {
        progressBarLarge.setVisibility(View.GONE);
        if (bm!=null) {
                imageViewFullScreen.setImageBitmap(bm);
        }
    }
}

        @Override
public void onConfigurationChanged(Configuration newConfig) {
   super.onConfigurationChanged(newConfig);
}

@Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                 if (mainLay.getVisibility()==View.GONE) {
                        //set visible & enable buttons
                        mainLay.setVisibility(View.VISIBLE);

                        buttonCamera = (Button) findViewById(R.id.buttonCamera);
                    buttonCamera.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (ImageFullScreen.this, HomeScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonUpload = (Button) findViewById(R.id.buttonUpload);
                    buttonUpload.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (ImageFullScreen.this, UploadScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonLogin = (Button) findViewById(R.id.buttonLogin);
                    buttonLogin.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (ImageFullScreen.this, LoginScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonHistory = (Button) findViewById(R.id.buttonHistory);
                    buttonHistory.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
```

```
                                Intent i = new Intent (ImageFullScreen.this, HistoryScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                               overridePendingTransition(0,0);
                             }
                            });

                            buttonAbout = (Button) findViewById(R.id.buttonAbout);
                            buttonAbout.setOnClickListener(new OnClickListener() {
                              public void onClick(View v) {
                                Intent i = new Intent (ImageFullScreen.this, AboutScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                               overridePendingTransition(0,0);
                             }
                            });

                        }
                        else {
                                mainLay.setVisibility(View.GONE);
                        }
                        return true;
        }

    @Override
    public void onBackPressed() {
            finish();
            overridePendingTransition(0,0);
    }
}
```

LoginScreen.java

```
package com.ntua.image.viral;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.util.Linkify;
```

```
import android.util.TypedValue;
import android.view.Gravity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

public class LoginScreen extends Activity {

        Button buttonCamera;
        Button buttonUpload;
        Button buttonLogin;
        Button buttonHistory;
        Button buttonAbout;
        Button buttonUserLogIn;
        LinearLayout mainLay;
        String username, password;
        ProgressDialog loadingDialog;
        LoginTask loginTask=null;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.loginscreen);

    mainLay = (LinearLayout) findViewById(R.id.main);
    mainLay.setVisibility(View.GONE);

    if (!checkCookie()) {
        //den exei syndethei
        TextView textViewAccountDesc = (TextView) findViewById(R.id.textViewAccountDesc);
        Linkify.addLinks(textViewAccountDesc, Linkify.WEB_URLS);
        final EditText usernameTextField = (EditText) findViewById(R.id.usernameTextField);
        final EditText passwordTextField = (EditText) findViewById(R.id.passwordTextField);


        buttonUserLogIn = (Button) findViewById(R.id.buttonUserLogIn);
            buttonUserLogIn.setOnClickListener(new OnClickListener() {
              public void onClick(View v) {
                  username = usernameTextField.getText().toString();
                  password = passwordTextField.getText().toString();

                  //Check for internet connection
                      if (!networkAvailable()) {
                          //den exei internet
                          Context context = getApplicationContext();
                          CharSequence text = getString(R.string.checkConnection);
                          int duration = Toast.LENGTH_SHORT;
                          Toast toast = Toast.makeText(context, text, duration);
                          toast.setGravity(Gravity.CENTER, 0, 0);
                          toast.show();
                      } else {
                          //exei internet
                          loadingDialog = new ProgressDialog(LoginScreen.this);
                          loadingDialog.setMessage(getString(R.string.loggingIn));
```

```
                                loadingDialog.setCancelable(true);
                                loadingDialog.setOnCancelListener(new
DialogInterface.OnCancelListener(){
                                 public void onCancel(DialogInterface dialog) {
                                    try {
                                          loginTask.cancel(false);
                                    }
                                    catch (NullPointerException e) {
                                          e.printStackTrace();
                                    }
                                 }
                               });

                                //Async task gia to an yparxei o xrhsths sth vash h oxi

                                loginTask = new LoginTask();
                                loginTask.execute();
                           }
                  }
                 });

      } else {
          //exw syndethei
          RelativeLayout accountLayout = (RelativeLayout) findViewById(R.id.accountLayout);
          accountLayout.removeAllViews();
          String user="";
          try {
                    BufferedReader in = new BufferedReader(new
FileReader(getCacheDir()+"Cookie.viral"));
                                user=in.readLine();
                           } catch (FileNotFoundException e) {
                                e.printStackTrace();
                           } catch (IOException e) {
                                e.printStackTrace();
                           }
                           //Layout
                           LinearLayout lay1 = new LinearLayout(this);
                           lay1.setLayoutParams(new LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT));
                           lay1.setOrientation(1); //vertical
                           lay1.setGravity(Gravity.CENTER);
                           //TextViews
                           TextView userTextView = new TextView(this);
                           userTextView.setLayoutParams(new
LayoutParams(LayoutParams.FILL_PARENT, LayoutParams.WRAP_CONTENT));
                           userTextView.setText(getString(R.string.currentUser));
                           userTextView.setTextSize(TypedValue.COMPLEX_UNIT_DIP, 20);
                           userTextView.setGravity(Gravity.CENTER);
                           lay1.addView(userTextView);
                           TextView userTextView2 = new TextView(this);
                           userTextView2.setLayoutParams(new
LayoutParams(LayoutParams.FILL_PARENT, LayoutParams.WRAP_CONTENT));
                           userTextView2.setText("\n"+user);
                           userTextView2.setTextSize(TypedValue.COMPLEX_UNIT_DIP, 20);
                           userTextView2.setTextColor(Color.WHITE);
                           userTextView2.setGravity(Gravity.CENTER);
                           lay1.addView(userTextView2);
          //Empty line
                           View emptyView = new View(this);
          int px = (int) (30 * getResources().getDisplayMetrics().density + 0.5f); //30 dp in pixels
          emptyView.setLayoutParams(new LayoutParams(LayoutParams.FILL_PARENT, px));
```

```
                        lay1.addView(emptyView);
        //Button Log out
                        Button buttonLogOut = new Button(this);
                        buttonLogOut.setLayoutParams(new
LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT));
                        buttonLogOut.setText(getString(R.string.logOut));
                        buttonLogOut.setTextSize(TypedValue.COMPLEX_UNIT_DIP, 20);
                        buttonLogOut.setGravity(Gravity.CENTER);
                        lay1.addView(buttonLogOut);
                        accountLayout.addView(lay1);
                        //Log out onClick
                        buttonLogOut.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        //delete cookie
                        File f = new File(getCacheDir()+"Cookie.viral");
                        f.delete();
                        Intent i = new Intent (LoginScreen.this, LoginScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                      overridePendingTransition(0,0);
                      }
                    });
    }
  }

  public boolean checkCookie() {
        File f = new File(getCacheDir()+"Cookie.viral");
        if (f.exists()) {
                return true;
        } else {
                return false;
        }
  }

  public boolean networkAvailable() {
        ConnectivityManager cn = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo ni = cn.getActiveNetworkInfo();
        if (ni==null)
           return false;
        return ni.isConnected();
  }

  private class LoginTask extends AsyncTask<Void, Void, Integer> {

    protected void onPreExecute() {
        loadingDialog.show();
    }

    protected Integer doInBackground(Void... params) {
        Integer result=0;
        if (!loginTask.isCancelled()) {
                try {
                                        Class.forName("org.postgresql.Driver");
                                        String url =
"jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/xxxxx";
                                        String user = "xxxxx";
                                        String pass = "xxxxx";
                                        Connection conn = DriverManager.getConnection(url, user,
pass);
```

```
                                        String query = "SELECT * FROM android_users WHERE
user_id="'+username+"' AND password="'+password+"'";
                                        Statement st = conn.createStatement();
                                        ResultSet rs = st.executeQuery(query);
                                        while (rs.next()) {
                                                //vrethike o xrhsths
                                                result=1;

                                        }
                                        rs.close();
                                        st.close();
                                        conn.close();

                                } catch (ClassNotFoundException e) {
                                        e.printStackTrace();
                                        loginTask.cancel(false);
                                        loadingDialog.dismiss();
                                } catch (SQLException e) {
                                        e.printStackTrace();
                                        loginTask.cancel(false);
                                        loadingDialog.dismiss();
                                }
        }
                        return result;
    }

    protected void onPostExecute(Integer result) {
        if (!loginTask.isCancelled()) {
                if (result==0) {
                        //de vrethike o xrhsths
                        Context context = getApplicationContext();
                        CharSequence text = getString(R.string.wrongUser);
                        int duration = Toast.LENGTH_SHORT;
                        Toast toast = Toast.makeText(context, text, duration);
                        toast.setGravity(Gravity.CENTER, 0, 0);
                        loadingDialog.dismiss();
                        toast.show();
                } else {
                        //vrethike o xrhsths ara dhmiourghse cookie
                        File f = new File(getCacheDir()+"Cookie.viral");
                        BufferedWriter output;
                                        try {
                                                output = new BufferedWriter(new FileWriter(f));
                                                output.write(username);
                                output.close();
                                        } catch (IOException e) {
                                                e.printStackTrace();
                                        }
                        loadingDialog.dismiss();
                        Intent i = new Intent (LoginScreen.this, LoginScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                }
        }
    }
  }

  @Override
  public void onConfigurationChanged(Configuration newConfig) {
     super.onConfigurationChanged(newConfig);
```

```
    }

    @Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                if (mainLay.getVisibility()==View.GONE) {
                        //set visible & enable buttons
                        mainLay.setVisibility(View.VISIBLE);

                        buttonCamera = (Button) findViewById(R.id.buttonCamera);
                    buttonCamera.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (LoginScreen.this, HomeScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonUpload = (Button) findViewById(R.id.buttonUpload);
                    buttonUpload.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (LoginScreen.this, UploadScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonLogin = (Button) findViewById(R.id.buttonLogin);
                    buttonLogin.getBackground().setColorFilter(Color.rgb(95, 252, 17),
android.graphics.PorterDuff.Mode.MULTIPLY);

                    buttonHistory = (Button) findViewById(R.id.buttonHistory);
                    buttonHistory.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (LoginScreen.this, HistoryScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonAbout = (Button) findViewById(R.id.buttonAbout);
                    buttonAbout.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (LoginScreen.this, AboutScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                }
                else {
                        mainLay.setVisibility(View.GONE);
                }
                return true;
        }

    @Override
    public void onBackPressed() {
```

```
            finish();
            overridePendingTransition(0,0);
    }
}
```

ResMapScreen.java

```
package com.ntua.image.viral;

import java.util.List;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.OverlayItem;

import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.ZoomControls;

public class ResMapScreen extends MapActivity {

            String result;
            double latitude,longitude;
            MapView mapView;
            ZoomControls zoomControls;
            JSONObject jObject, locationObj;
            JSONArray imagesArr;
            MapController mc;
    GeoPoint p, p1;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.resmapscreen);

        Bundle bundle = getIntent().getExtras();
        try {
            result = bundle.getString("result");
        }
        catch (NullPointerException e) {
            e.printStackTrace();
        }

        //Getting latitude & longitude
        try {
                                jObject = new JSONObject(result);
                                locationObj = jObject.getJSONObject("location");
                                latitude = locationObj.getDouble("latitude");
                                longitude = locationObj.getDouble("longitude");
                        } catch (JSONException e) {
                                e.printStackTrace();
                        }
```

```
    //Displaying map
    mapView = (MapView) findViewById(R.id.mapView);
    mc = mapView.getController();
    //ZOOM buttons:
    zoomControls = (ZoomControls) findViewById(R.id.zoomcontrols);
    zoomControls.setOnZoomInClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            mc.zoomIn();
        }
    });
    zoomControls.setOnZoomOutClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            mc.zoomOut();
        }
    });

    //Coordinates of estimated location:
    p1 = new GeoPoint(
        (int) (latitude * 1E6),
        (int) (longitude * 1E6));

    mc.animateTo(p1);
    mc.setZoom(17);

    //Add location markers
    List<Overlay> mapOverlays = mapView.getOverlays();

    Drawable drawable = this.getResources().getDrawable(R.drawable.bluepin);
    CustomItemizedOverlay itemizedOverlay;
    OverlayItem overlayitem;

    //Blue markers: similar images (to polu 16)
    try {
                        imagesArr = jObject.getJSONArray("images");
                        int i=0;
                        int imagesArrLength = imagesArr.length();
                        while (i<imagesArrLength && i<16) {

        locationObj=imagesArr.getJSONObject(i).getJSONObject("location");
                            latitude = locationObj.getDouble("latitude");
                            longitude = locationObj.getDouble("longitude");
                            p = new GeoPoint(
                        (int) (latitude * 1E6),
                        (int) (longitude * 1E6));

                    itemizedOverlay = new CustomItemizedOverlay(drawable, this);

                    overlayitem = new OverlayItem(p, getString(R.string.similarImage),
Double.toString(latitude)+", "+Double.toString(longitude));
                    itemizedOverlay.addOverlay(overlayitem);

                    mapOverlays.add(itemizedOverlay);

                            i++;
                        }
                } catch (JSONException e) {
                        e.printStackTrace();
                }

                //Red marker: estimated location
```

```
        drawable = this.getResources().getDrawable(R.drawable.redpin);
        itemizedOverlay = new CustomItemizedOverlay(drawable, this);

        overlayitem = new OverlayItem(p1, getString(R.string.estimatedLoc), Double.toString(latitude)+",
"+Double.toString(longitude));
        itemizedOverlay.addOverlay(overlayitem);

        mapOverlays.add(itemizedOverlay);

        mapView.invalidate();
    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }

    @Override
    public void onBackPressed() {
            finish();
            overridePendingTransition(0,0);
    }
}
```

ResOriginalScreen.java

```
package com.ntua.image.viral;

import java.io.BufferedOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.nio.channels.FileChannel;

import org.apache.sanselan.ImageReadException;
import org.apache.sanselan.ImageWriteException;
import org.apache.sanselan.Sanselan;
import org.apache.sanselan.common.IImageMetadata;
import org.apache.sanselan.formats.jpeg.JpegImageMetadata;
import org.apache.sanselan.formats.jpeg.exifRewrite.ExifRewriter;
import org.apache.sanselan.formats.tiff.TiffImageMetadata;
import org.apache.sanselan.formats.tiff.constants.TiffConstants;
import org.apache.sanselan.formats.tiff.constants.TiffFieldTypeConstants;
import org.apache.sanselan.formats.tiff.write.TiffOutputDirectory;
import org.apache.sanselan.formats.tiff.write.TiffOutputField;
import org.apache.sanselan.formats.tiff.write.TiffOutputSet;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.os.Environment;
```

```
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.view.Gravity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.CheckBox;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

public class ResOriginalScreen extends Activity {

	String result;
	Bitmap bitmap;
	ImageView imageView1, imageViewSave;
	String city="";
	String country="";
	String suggestedTags="";
	String suggestedTagsNoHtml="";
	String frequentTags="";
	JSONObject jObject;
	JSONArray landmarksArr, tagsArr;
	TextView textViewLoc, textViewStags, textViewFtags, textViewSuggestedTags,
textViewFrequentTags;
	double latitude, longitude;

   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.resoriginalscreen);

      Bundle bundle = getIntent().getExtras();
      try {
          result = bundle.getString("result");
      }
      catch (NullPointerException e) {
          e.printStackTrace();
      }

      //Getting original bitmap from cache
      imageView1 = (ImageView)findViewById(R.id.imageViewOriginal);
      try {
          bitmap=BitmapFactory.decodeFile(getCacheDir()+"viral_image.jpg");
          imageView1.setImageBitmap(bitmap);
      } catch(NullPointerException e) {
          e.printStackTrace();
      }

      //Getting city & country
      try {
                          jObject = new JSONObject(result);
                          city = jObject.getString("city");
                          country = jObject.getString("country");
                  } catch (JSONException e) {
                          e.printStackTrace();
                  }
                  textViewLoc = (TextView)findViewById(R.id.TextViewLoc);
                  textViewLoc.setText(Html.fromHtml("<a
href=\"http://en.wikipedia.org/wiki/"+city+"\">"+city+"</a>"+
```

```
                                        ", <a
href=\"http://en.wikipedia.org/wiki/"+country+"\">"+country+"</a>"));
                        textViewLoc.setMovementMethod(LinkMovementMethod.getInstance());
                        textViewLoc.setLinksClickable(true);

                        //Getting landmarks
                        textViewSuggestedTags = (TextView)findViewById(R.id.TextViewSuggestedTags);
                        textViewStags = (TextView)findViewById(R.id.TextViewStags);
        String name="", link="";
                        int i=0;
                        boolean found=false;
                        try {
                                landmarksArr = jObject.getJSONArray("landmarks");
                                int landmarksArrLength = landmarksArr.length();
                                while (i<landmarksArrLength && i<4) {  //thelw ews 4 suggested tags
                                        name = landmarksArr.getJSONObject(i).getString("name");
                                        if (!name.equals(city)) {
                                                //emfanizw landmark mono an einai diaforetiko apo thn
polh
                                                link = landmarksArr.getJSONObject(i).getString("link");
                                                suggestedTags = suggestedTags + "<a
href=\""+link+"\">"+name+"</a>"+", ";
                                                suggestedTagsNoHtml = suggestedTagsNoHtml + name +
", ";
                                                found=true;
                                        }
                                        i++;
                                }
                        } catch (JSONException e) {
                                e.printStackTrace();
                        }
                        if (!found) {
                                //de vrhke landmarks, ara svhse to string Suggested tags
                                textViewSuggestedTags.setText("");
                                suggestedTagsNoHtml="";
                        } else {
                                //vrhke landmarks
                                //afairw tous 2 teleutaious xarakthres ", " apo to string
                                suggestedTags = suggestedTags.substring(0, suggestedTags.length()-2);
                                suggestedTagsNoHtml = suggestedTagsNoHtml.substring(0,
suggestedTagsNoHtml.length()-2);
                                textViewStags.setText(Html.fromHtml(suggestedTags));
                                textViewStags.setMovementMethod(LinkMovementMethod.getInstance());
                                textViewStags.setLinksClickable(true);
                        }

                        //Getting frequent tags
                        textViewFrequentTags = (TextView)findViewById(R.id.TextViewFrequentTags);
                        textViewFtags = (TextView)findViewById(R.id.TextViewFtags);
                        i=0;
                        try {
                                tagsArr = jObject.getJSONArray("tags");
                                int tagsArrLength = tagsArr.length();
                                while (i<tagsArrLength && i<4) {  //thelw ews 4 frequent tags
                                        frequentTags = frequentTags + tagsArr.getString(i) + ", ";
                                        i++;
                                }
                        } catch (JSONException e) {
                                e.printStackTrace();
                        }
                        if (i==0 || frequentTags.equals(", ")) {
```

```
                        //de vrhke tags, ara svhse to string Frequent tags
                        textViewFrequentTags.setText("");
                        frequentTags="";
                } else {
                        //vrhke tags
                        //afairw tous 2 teleutaious xarakthres ", " apo to string
                        frequentTags = frequentTags.substring(0, frequentTags.length()-2);
                        textViewFtags.setText(frequentTags);
                }

                //Save image
                imageViewSave = (ImageView)findViewById(R.id.imageViewSaveImage);
                imageViewSave.setOnClickListener(new OnClickListener() {
              public void onClick(View v) {
                        //Rwtaw an thelei na apothikeusei thn eikona
                   final CheckBox checkBox1 = new CheckBox(ResOriginalScreen.this);
                   checkBox1.setChecked(true);
                   final CheckBox checkBox2 = new CheckBox(ResOriginalScreen.this);
                   checkBox2.setChecked(true);
                   TextView textView1 = new TextView(ResOriginalScreen.this);
                   textView1.setText(getString(R.string.includeCoordinates));
                   TextView textView2 = new TextView(ResOriginalScreen.this);
                   textView2.setText(getString(R.string.includeTags));
                   LinearLayout linearLayoutDialog = new LinearLayout(ResOriginalScreen.this);
                   linearLayoutDialog.setOrientation(1); //Vertical
                   LinearLayout linear1 = new LinearLayout(ResOriginalScreen.this);
                   LinearLayout linear2 = new LinearLayout(ResOriginalScreen.this);
                   linear1.addView(checkBox1);
                   linear1.addView(textView1);
                   linear2.addView(checkBox2);
                   linear2.addView(textView2);
                   linearLayoutDialog.addView(linear1);
                   linearLayoutDialog.addView(linear2);
                        AlertDialog.Builder builder = new
AlertDialog.Builder(ResOriginalScreen.this);
                        builder.setMessage(getString(R.string.saveImage))
                             .setView(linearLayoutDialog)
                             .setCancelable(true)
                             .setPositiveButton(getString(R.string.saveToSd), new
DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface dialog, int id) {
                                  //Save to SD:
                                  //Check if directory exists. If not create it.
                                  File dir = new File(Environment.getExternalStorageDirectory() +
"/VIRaL Captures");
                                  if(dir.exists() && dir.isDirectory()) {
                                    //do nothing
                                  } else {
                                            //create it
                                            dir.mkdir();
                                  }
                                  //Find filename
                                  String[] fileNames = dir.list();
                                  String number;
                                  if (fileNames==null) {
                                            //An o fakelos einai adeios
                                            number="0001";
                                  } else {
                                            int intNumber=0;
                                            int max=0;
                                            int i;
```

```
                                        //find max
                                        int fileNamesLength = fileNames.length;
                                        for (i=0; i<fileNamesLength; i++) {
                                                if (fileNames[i].contains("ViralImage")) {
                                                        number=fileNames[i].substring(11,
fileNames[i].length()-4);
                                                        intNumber=Integer.valueOf(number);
                                                        if (intNumber>max) max=intNumber;

                                                }
                                        }
                                        if (max>=9999) {
                                                intNumber=max+1;
                                                number=Integer.toString(intNumber);
                                        } else if (max==0) {
                                                number="0001";
                                        } else {
                                                //o max anhkei sto [1,998]
                                                intNumber=max+1;
                                                number=Integer.toString(intNumber);
                                                //elegxw to length, diladi ta pshfia
                                                int numLen=number.length();
                                                if (numLen==1) number="000"+number;
                                                else if (numLen==2) number="00"+number;
                                                else if (numLen==3) number="0"+number;
                                                //else tha exei 4 pshfia dld number=number
                                        }
                                }
                                String filenameSd="ViralImage_"+number+".jpg";
                                //Save to SD
                                File source = new File (getCacheDir()+"viral_image.jpg");
                                String path = Environment.getExternalStorageDirectory()
                                                                + "/VIRaL Captures" +
File.separator + filenameSd;
                                File dest = new File(path);
                                boolean gpsChecked=checkBox1.isChecked();
                                boolean tagsChecked=checkBox2.isChecked();

                                if (!gpsChecked && !tagsChecked) {
                                        //den exei epileksei metadata
                                        try {
                                                copyFile(source, dest);
                                        } catch (IOException e) {
                                                e.printStackTrace();

                                        }
                                } else {
                                        //exei epileksei metadata
                                        try {
                                                changeExifMetadata(source, dest, gpsChecked,
tagsChecked);
                                        } catch (ImageReadException e) {
                                                e.printStackTrace();
                                        } catch (ImageWriteException e) {
                                                e.printStackTrace();
                                        } catch (IOException e) {
                                                e.printStackTrace();
                                        }
                                }

                                dialog.cancel();
```

```
                                        Context context = getApplicationContext();
                                          CharSequence text = getString(R.string.imageSaved);
                                          int duration = Toast.LENGTH_SHORT;
                                          Toast toast = Toast.makeText(context, text, duration);
                                          toast.setGravity(Gravity.CENTER, 0, 0);
                                          toast.show();
                                    }
                                })
                                .setNegativeButton(getString(R.string.cancel), new
DialogInterface.OnClickListener() {
                                    public void onClick(DialogInterface dialog, int id) {
                                       dialog.cancel();
                                    }
                                });
                            AlertDialog alert = builder.create();
                            alert.show();
                }
               });
   }

   public void copyFile(File src, File dst) throws IOException {
      FileChannel inChannel = new FileInputStream(src).getChannel();
      FileChannel outChannel = new FileOutputStream(dst).getChannel();
      try {
         inChannel.transferTo(0, inChannel.size(), outChannel);
      } finally {
         if (inChannel != null)
            inChannel.close();
         if (outChannel != null)
            outChannel.close();
      }
   }

   public void changeExifMetadata(File jpegImageFile, File dst, boolean gpsChecked, boolean
tagsChecked)
                        throws IOException, ImageReadException, ImageWriteException {

                OutputStream os = null;
                try {
                        TiffOutputSet outputSet = null;

                    //note that metadata might be null if no metadata is found.
                    IImageMetadata metadata = Sanselan.getMetadata(jpegImageFile);
                    JpegImageMetadata jpegMetadata = (JpegImageMetadata) metadata;
                    if (null != jpegMetadata) {
                          // note that exif might be null if no Exif metadata is found.
                          TiffImageMetadata exif = jpegMetadata.getExif();

                          if (null != exif) {
                                outputSet = exif.getOutputSet();
                          }
                    }
                    // if file does not contain any exif metadata, we create an empty
                    // set of exif metadata. Otherwise, we keep all of the other
                    // existing tags.
                    if (outputSet == null) outputSet = new TiffOutputSet();

                    if (tagsChecked) {
                       // Note that you should first remove the field/tag if it already
                       // exists in this directory, or you may end up with duplicate
                       // tags.
```

```
                        TiffOutputField proc_method = new TiffOutputField(
                              TiffConstants.GPS_TAG_GPS_PROCESSING_METHOD,
                              TiffFieldTypeConstants.FIELD_TYPE_ASCII,
                              "VIRaL".length(),
                              "VIRaL".getBytes());

                        TiffOutputField location1 = new TiffOutputField(
                              TiffConstants.EXIF_TAG_SUBJECT_LOCATION_1,
                              TiffFieldTypeConstants.FIELD_TYPE_ASCII,
                              city.length(),
                              city.getBytes());

                        TiffOutputField location2 = new TiffOutputField(
                              TiffConstants.EXIF_TAG_SUBJECT_LOCATION_2,
                              TiffFieldTypeConstants.FIELD_TYPE_ASCII,
                              country.length(),
                              country.getBytes());


                        TiffOutputField desc = new TiffOutputField(
                              TiffConstants.EXIF_TAG_IMAGE_DESCRIPTION,
                              TiffFieldTypeConstants.FIELD_TYPE_ASCII,
                              suggestedTagsNoHtml.length(),
                              suggestedTagsNoHtml.getBytes());

                        TiffOutputField comment = new TiffOutputField(
                              TiffConstants.EXIF_TAG_USER_COMMENT,
                              TiffFieldTypeConstants.FIELD_TYPE_ASCII,
                              frequentTags.length(),
                              frequentTags.getBytes());

                        TiffOutputDirectory exifDirectory = outputSet.getOrCreateExifDirectory();
                        //make sure to remove old value if present (this method will
                        //not fail if the tag does not exist).

exifDirectory.removeField(TiffConstants.GPS_TAG_GPS_PROCESSING_METHOD);

exifDirectory.removeField(TiffConstants.EXIF_TAG_SUBJECT_LOCATION_1);

exifDirectory.removeField(TiffConstants.EXIF_TAG_SUBJECT_LOCATION_2);
                        exifDirectory.removeField(TiffConstants.EXIF_TAG_IMAGE_DESCRIPTION);
                        exifDirectory.removeField(TiffConstants.EXIF_TAG_USER_COMMENT);
                        exifDirectory.add(proc_method);
                        exifDirectory.add(location1);
                        exifDirectory.add(location2);
                        exifDirectory.add(desc);
                        exifDirectory.add(comment);
                     }

                     if (gpsChecked) {
                           JSONObject locationObj;
                                       try {
                                                   locationObj =
jObject.getJSONObject("location");
                                                   latitude = locationObj.getDouble("latitude");
                                                               longitude =
locationObj.getDouble("longitude");
                                                                           } catch
(JSONException e1) {

e1.printStackTrace();
```

```
                                                                    }

            outputSet.setGPSInDegrees(longitude, latitude);
                        }

                        os = new FileOutputStream(dst);
                        os = new BufferedOutputStream(os);

                        new ExifRewriter().updateExifMetadataLossless(jpegImageFile, os,
                                    outputSet);

                        os.close();
                        os = null;
                  } finally {
                        if (os != null)
                              try {
                                    os.close();
                              } catch (IOException e) {
                                          e.printStackTrace();
                              }
                  }
      }

      @Override
      public void onBackPressed() {
            finish();
            overridePendingTransition(0,0);
      }
}
```

ResSimilarScreen.java

```
package com.ntua.image.viral;

import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageView;
import android.widget.LinearLayout;

public class ResSimilarScreen extends Activity {

            JSONObject jObject;
            JSONArray imagesArr;
```

```
        String result;
        ArrayList<String> imageList;
        ProgressDialog loadingDialog;
        LoadingTask loadTask=null;
        LinearLayout similar1, similar2, similar3, similar4;
        float scale;
        int px1;
        int px2;
        LinearLayout.LayoutParams Params;

    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.ressimilarscreen);

        Bundle bundle = getIntent().getExtras();
        try {
            result = bundle.getString("result");
        }
        catch (NullPointerException e) {
            e.printStackTrace();
        }

        scale = getResources().getDisplayMetrics().density;
        px1 = (int) (75 * scale + 0.5f); //75 dp in pixels
        px2 = (int) (1 * scale + 0.5f);  //1 dp in pixels
        Params = new LinearLayout.LayoutParams(px1, px1, 1);

                    loadingDialog = new ProgressDialog(this);
                    loadingDialog.setMessage(getString(R.string.loadDialog));
                    loadingDialog.setCancelable(true);
                    loadingDialog.setOnCancelListener(new DialogInterface.OnCancelListener(){
          public void onCancel(DialogInterface dialog) {
            try {
                            loadTask.cancel(false);
            }
            catch (NullPointerException e) {
                            e.printStackTrace();
            }
            finish();
            overridePendingTransition(0,0);
          }
        });

                    similar1 = (LinearLayout) findViewById(R.id.similar1);
                    similar2 = (LinearLayout) findViewById(R.id.similar2);
                    similar3 = (LinearLayout) findViewById(R.id.similar3);
                    similar4 = (LinearLayout) findViewById(R.id.similar4);

                    loadTask = new LoadingTask();
          loadTask.execute();
    }

    public Bitmap getRemoteImage(String imageUrl) {
            URL url=null;
            Bitmap similarBitmap=null;
                    try {
                            url = new URL(imageUrl);
                            similarBitmap =
BitmapFactory.decodeStream(url.openConnection().getInputStream());
```

```
                } catch (MalformedURLException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                }
                return similarBitmap;
}

private class LoadingTask extends AsyncTask<Void, Integer, Integer> {

   protected void onPreExecute() {
      loadingDialog.show();
   }

   protected Integer doInBackground(Void... params) {
      if (!loadTask.isCancelled()) {
                //Create array with url's
       imageList=new ArrayList<String>();
       int i;
       try {
                        jObject = new JSONObject(result);
                        imagesArr = jObject.getJSONArray("images");
                        i=0;
                        String url;
                        int imagesArrLength = imagesArr.length();
                        while (i<imagesArrLength && i<16) {
                                url=imagesArr.getJSONObject(i).getString("uri");
                                imageList.add(url);
                                i++;
                        }
                } catch (JSONException e) {
                        e.printStackTrace();
                        loadTask.cancel(false);
                }

                if (!loadTask.isCancelled()) {
                        //Display images
                        int length=imageList.size();

                for (i=0; i<length; i++) {
                        publishProgress(i);
                }
                }
      }
                        return 1;
   }

   protected void onProgressUpdate(Integer... a) {
      //interaction with UI thread
      int i = a[0];
      ImageView im = new ImageView (ResSimilarScreen.this);
      final String imageUrl = imageList.get(i).toString();
      im.setImageBitmap(getRemoteImage(imageUrl));
    im.setLayoutParams(Params);
    im.setAdjustViewBounds(true);
    im.setPadding(px2, px2, px2, px2);
    if (i<4) similar1.addView(im);
    else if (i<8) similar2.addView(im);
    else if (i<12) similar3.addView(im);
    else similar4.addView(im);
    im.setOnClickListener(new OnClickListener() {
```

```
                            public void onClick(View v) {
                                Intent in = new Intent (ResSimilarScreen.this, ImageFullScreen.class);
                                in.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                in.putExtra("imageUrl", imageUrl);
                                startActivity (in);
                               overridePendingTransition(0,0);
                            }
                        });
        }

        protected void onPostExecute(Integer a) {
            if (!loadTask.isCancelled()) {
                        loadingDialog.dismiss();
            }
        }
    }

    @Override
    public void onBackPressed() {
            finish();
            overridePendingTransition(0,0);
    }
}
```

ResSimofsimScreen.java

```
package com.ntua.image.viral;

import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Toast;

public class ResSimofsimScreen extends Activity {

            JSONObject jObject;
            JSONArray imagesArr;
            String result;
            ArrayList<String> imageList;
            ProgressDialog loadingDialog;
```

```
        LoadingTask loadTask=null;
        LinearLayout sos1, sos2, sos3, sos4;
        float scale;
        int px1;
        int px2;
        LinearLayout.LayoutParams Params;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.ressimofsimscreen);

    Bundle bundle = getIntent().getExtras();
    try {
        result = bundle.getString("result");
    }
    catch (NullPointerException e) {
        e.printStackTrace();
    }

    scale = getResources().getDisplayMetrics().density;
    px1 = (int) (75 * scale + 0.5f); //75 dp in pixels
    px2 = (int) (1 * scale + 0.5f);  //1 dp in pixels
    Params = new LinearLayout.LayoutParams(px1, px1, 1);

                loadingDialog = new ProgressDialog(this);
                loadingDialog.setMessage(getString(R.string.loadDialog));
                loadingDialog.setCancelable(true);
                loadingDialog.setOnCancelListener(new DialogInterface.OnCancelListener(){
      public void onCancel(DialogInterface dialog) {
        try {
                        loadTask.cancel(false);
        }
        catch (NullPointerException e) {
                        e.printStackTrace();
        }
        finish();
        overridePendingTransition(0,0);
      }
    });

                sos1 = (LinearLayout) findViewById(R.id.sos1);
                sos2 = (LinearLayout) findViewById(R.id.sos2);
                sos3 = (LinearLayout) findViewById(R.id.sos3);
                sos4 = (LinearLayout) findViewById(R.id.sos4);

                loadTask = new LoadingTask();
        loadTask.execute();
  }

  public Bitmap getRemoteImage(String imageUrl) {
        URL url=null;
        Bitmap similarBitmap=null;
                try {
                        url = new URL(imageUrl);
                        similarBitmap =
BitmapFactory.decodeStream(url.openConnection().getInputStream());
                } catch (MalformedURLException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
```

```
                }
                return similarBitmap;
}

private class LoadingTask extends AsyncTask<Void, Integer, String> {

   protected void onPreExecute() {
       loadingDialog.show();
   }

   protected String doInBackground(Void... params) {
       String found="1";
       if (!loadTask.isCancelled()) {
                //Create array with url's
        imageList=new ArrayList<String>();
        int i=0;
        try {
                        jObject = new JSONObject(result);
                        imagesArr = jObject.getJSONArray("sos");
                        String url;
                        int imagesArrLength = imagesArr.length();
                        while (i<imagesArrLength && i<16) {
                                url=imagesArr.getJSONObject(i).getString("uri");
                                imageList.add(url);
                                i++;
                        }
                } catch (JSONException e) {
                        e.printStackTrace();
                        loadTask.cancel(false);
                }
                if (i==0) {
                        //de vrhke sos images
                        found="0";
                } else {
                        //vrhke sos images
                        if (!loadTask.isCancelled()) {
                        //Display images
                        int length=imageList.size();

                        for (i=0; i<length; i++) {
                                publishProgress(i);
                        }
                }
                }
       }
                        return found;
   }

   protected void onProgressUpdate(Integer... a) {
       //interaction with UI thread
       int i = a[0];
       ImageView im = new ImageView (ResSimofsimScreen.this);
       final String imageUrl = imageList.get(i).toString();
       im.setImageBitmap(getRemoteImage(imageUrl));
     im.setLayoutParams(Params);
     im.setAdjustViewBounds(true);
     im.setPadding(px2, px2, px2, px2);
     if (i<4) sos1.addView(im);
     else if (i<8) sos2.addView(im);
     else if (i<12) sos3.addView(im);
     else sos4.addView(im);
```

```
        im.setOnClickListener(new OnClickListener() {
                        public void onClick(View v) {
                            Intent in = new Intent (ResSimofsimScreen.this, ImageFullScreen.class);
                            in.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                            in.putExtra("imageUrl", imageUrl);
                            startActivity (in);
                           overridePendingTransition(0,0);
                        }
                    });
    }

    protected void onPostExecute(String found) {
        if (!loadTask.isCancelled()) {
                loadingDialog.dismiss();
                if (found.equals("0")) {
                //de vrhke sos images
                        Context context = getApplicationContext();
                   CharSequence text = getString(R.string.noSimofsimImages);
                   int duration = Toast.LENGTH_LONG;
                   Toast toast = Toast.makeText(context, text, duration);
                   toast.setGravity(Gravity.CENTER, 0, 0);
                   toast.show();
        }
        }
    }
  }

  @Override
  public void onBackPressed() {
        finish();
        overridePendingTransition(0,0);
  }
}
```

ResultsScreen.java

```
package com.ntua.image.viral;

import android.app.TabActivity;
import android.content.Context;
import android.content.Intent;
import android.content.res.Configuration;
import android.content.res.Resources;
import android.os.Bundle;
import android.view.Gravity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TabHost;
import android.widget.Toast;

public class ResultsScreen extends TabActivity {

        Button buttonCamera;
        Button buttonUpload;
        Button buttonLogin;
        Button buttonHistory;
        Button buttonAbout;
```

```
        String result;
        LinearLayout mainLay;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.tabs);

    mainLay = (LinearLayout) findViewById(R.id.main);
    mainLay.setVisibility(View.GONE);

    Bundle bundle = getIntent().getExtras();
    try {
        result = bundle.getString("result");
    }
    catch (NullPointerException e) {
        e.printStackTrace();
    }

    if (result.equals("-1") || (result.contains("{\"images\":[]"))) {
        //den exei vrei apotelesmata
        Context context = getApplicationContext();
                    CharSequence text = getString(R.string.noSimImages);
                    int duration = Toast.LENGTH_LONG;
                    Toast toast = Toast.makeText(context, text, duration);
                    toast.setGravity(Gravity.CENTER, 0, 0);
                    toast.show();
                    finish();
                    overridePendingTransition(0,0);
    }
    else {
        //vrhke apotelesmata
        //ara create tabs
        Resources res = getResources();
      TabHost tabHost = getTabHost();
      TabHost.TabSpec spec;
      Intent intent;

      //Create an Intent to launch an Activity for the tab (to be reused)
      //TAB 1
      intent = new Intent().setClass(this, ResOriginalScreen.class);
      intent.putExtra("result", result);

      //Initialize a TabSpec for each tab and add it to the TabHost
      spec = tabHost.newTabSpec("original").setIndicator(getString(R.string.restab1),
                  res.getDrawable(R.drawable.original3))
              .setContent(intent);
      tabHost.addTab(spec);

      //Do the same for the other tabs
      //TAB 2
      intent = new Intent().setClass(this, ResMapScreen.class);
      intent.putExtra("result", result);
      spec = tabHost.newTabSpec("map").setIndicator(getString(R.string.restab2),
                  res.getDrawable(R.drawable.map4))
              .setContent(intent);
      tabHost.addTab(spec);

      //TAB 3
      intent = new Intent().setClass(this, ResSimilarScreen.class);
      intent.putExtra("result", result);
```

```
        spec = tabHost.newTabSpec("similar").setIndicator(getString(R.string.restab3),
                    res.getDrawable(R.drawable.similar4))
                .setContent(intent);
        tabHost.addTab(spec);

        //TAB 4
        intent = new Intent().setClass(this, ResSimofsimScreen.class);
        intent.putExtra("result", result);
        spec = tabHost.newTabSpec("simofsim").setIndicator(getString(R.string.restab4),
                    res.getDrawable(R.drawable.simofsim))
                .setContent(intent);
        tabHost.addTab(spec);

        tabHost.setCurrentTabByTag("original");
    }
}

@Override
public void onConfigurationChanged(Configuration newConfig) {
   super.onConfigurationChanged(newConfig);
}

@Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                    if (mainLay.getVisibility()==View.GONE) {
                                //set visible & enable buttons
                                mainLay.setVisibility(View.VISIBLE);

                                buttonCamera = (Button) findViewById(R.id.buttonCamera);
                            buttonCamera.setOnClickListener(new OnClickListener() {
                              public void onClick(View v) {
                                Intent i = new Intent (ResultsScreen.this, HomeScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                               overridePendingTransition(0,0);
                              }
                            });

                            buttonUpload = (Button) findViewById(R.id.buttonUpload);
                            buttonUpload.setOnClickListener(new OnClickListener() {
                              public void onClick(View v) {
                                Intent i = new Intent (ResultsScreen.this, UploadScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                               overridePendingTransition(0,0);
                              }
                            });

                            buttonLogin = (Button) findViewById(R.id.buttonLogin);
                            buttonLogin.setOnClickListener(new OnClickListener() {
                              public void onClick(View v) {
                                Intent i = new Intent (ResultsScreen.this, LoginScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                               overridePendingTransition(0,0);
                              }
                            });

                            buttonHistory = (Button) findViewById(R.id.buttonHistory);
                            buttonHistory.setOnClickListener(new OnClickListener() {
                              public void onClick(View v) {
```

```
                                Intent i = new Intent (ResultsScreen.this, HistoryScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                               overridePendingTransition(0,0);
                             }
                            });

                            buttonAbout = (Button) findViewById(R.id.buttonAbout);
                            buttonAbout.setOnClickListener(new OnClickListener() {
                              public void onClick(View v) {
                                Intent i = new Intent (ResultsScreen.this, AboutScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                               overridePendingTransition(0,0);
                             }
                            });

                        }
                        else {
                                mainLay.setVisibility(View.GONE);
                        }
                        return true;
                }

    @Override
    public void onBackPressed() {
            finish();
            overridePendingTransition(0,0);
    }
}
```

SendScreen.java

```
package com.ntua.image.viral;

import java.io.BufferedReader;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.apache.commons.net.ftp.FTP;
import org.apache.commons.net.ftp.FTPClient;
import org.apache.commons.net.ftp.FTPConnectionClosedException;
import org.apache.commons.net.ftp.FTPReply;

import android.app.Activity;
import android.app.AlertDialog;
```

```
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.view.Gravity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Toast;

public class SendScreen extends Activity {

        Button buttonCamera;
        Button buttonUpload;
        Button buttonLogin;
        Button buttonHistory;
        Button buttonAbout;
        ProgressDialog uploadingDialog;
        String filename="";
        File f;
        UploadingTask myTask=null;
        Bitmap bitmap;
        LinearLayout mainLay;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.sendscreen);

    mainLay = (LinearLayout) findViewById(R.id.main);
    mainLay.setVisibility(View.GONE);

    Bundle bundle = getIntent().getExtras();
    byte[] dataHome=bundle.getByteArray("dataHome");
    byte[] data=dataHome;
    if (data==null) {
        //shmainei oti erxomai apo thn UploadScreen
        data=bundle.getByteArray("dataUpload");
    }

    //Check if directory exists. If not create it.
    File dir = new File(Environment.getExternalStorageDirectory() + "/VIRaL Captures");
    if(dir.exists() && dir.isDirectory()) {
       //do nothing
    } else {
        //create it
        dir.mkdir();
    }
```

```
                try {
            //Create bitmap from data byte[]
            bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);

            //Rotate image if
            //(erxomai apo th HomeScreen) AND (orientation==Portrait)
            if (dataHome!=null) {
                if
(getResources().getConfiguration().orientation==Configuration.ORIENTATION_PORTRAIT) {
                    Matrix matrix = new Matrix();
                    matrix.postRotate(90);
                    Bitmap bitmapRotated = Bitmap.createBitmap(bitmap, 0, 0,
bitmap.getWidth(), bitmap.getHeight(), matrix, true);
                    bitmap = bitmapRotated;
                }
            }

            //set background image & click listener (se periptwsh pou de vrei similar images tha
epistrepsei edw)
            ImageView imageSendscreen = (ImageView) findViewById(R.id.imageSendscreen);
            imageSendscreen.setImageBitmap(bitmap);
            imageSendscreen.setOnClickListener(new OnClickListener() {
                public void onClick(View v) {
                    AlertDialog.Builder builder = new AlertDialog.Builder(SendScreen.this);
                    builder.setMessage(getString(R.string.saveImage))
                            .setCancelable(true)
                            .setPositiveButton(getString(R.string.saveToSd), new
DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface dialog, int id) {
                                    //Save to SD
                                    String filenameSd = findFileName();
                                    File fSd = new
File(Environment.getExternalStorageDirectory()
                                            + "/VIRaL Captures" + File.separator + filenameSd);
                                    FileOutputStream outSd;
                                    try {
                                            outSd = new FileOutputStream(fSd);

bitmap.compress(Bitmap.CompressFormat.JPEG, 100, outSd);
                                            outSd.close();
                                                                    } catch
(FileNotFoundException e) {

e.printStackTrace();
                                                                    } catch (IOException
e) {

e.printStackTrace();
                                                                    }

                                    dialog.cancel();
                                    Context context = getApplicationContext();
                                      CharSequence text = getString(R.string.imageSaved);
                                      int duration = Toast.LENGTH_SHORT;
                                      Toast toast = Toast.makeText(context, text, duration);
                                      toast.setGravity(Gravity.CENTER, 0, 0);
                                      toast.show();
                                      finish();
                                      overridePendingTransition(0,0);
                                }
                            })
```

```
                                        .setNegativeButton(getString(R.string.cancel), new
DialogInterface.OnClickListener() {
                                            public void onClick(DialogInterface dialog, int id) {
                                               dialog.cancel();
                                               finish();
                                                 overridePendingTransition(0,0);
                                            }
                                        });
                                    AlertDialog alert = builder.create();
                                    alert.show();
                            }
                        });

                //Check for internet connection
                if (!networkAvailable()) {
                    //Den exei internet ara emfanizw alert dialog
                    if (dataHome!=null) {
                            //Erxomai apo th HomeScreen
                            //Rwtaw an thelei na apothikeusei thn eikona
                            AlertDialog.Builder builder = new AlertDialog.Builder(this);
                            builder.setTitle(getString(R.string.noInternet));
                            builder.setMessage(getString(R.string.askSaveImage))
                                .setCancelable(true)
                                .setOnCancelListener(new DialogInterface.OnCancelListener(){
                                    public void onCancel(DialogInterface dialog) {
                                            finish();
                                            overridePendingTransition(0,0);
                                    }
                                })
                                .setPositiveButton(getString(R.string.saveToSd), new
DialogInterface.OnClickListener() {
                                   public void onClick(DialogInterface dialog, int id) {
                                        //Save to SD
                                        String filenameSd = findFileName();
                                        File fSd = new File(Environment.getExternalStorageDirectory()
                                            + "/VIRaL Captures" + File.separator + filenameSd);
                                        FileOutputStream outSd;
                                        try {
                                                outSd = new FileOutputStream(fSd);
                                                bitmap.compress(Bitmap.CompressFormat.JPEG, 100,
outSd);
                                                outSd.close();
                                                                        } catch
(FileNotFoundException e) {
                                                                                e.printStackTrace();
                                                                        } catch (IOException e) {
                                                                                e.printStackTrace();
                                                                        }

                                        dialog.cancel();
                                        Context context = getApplicationContext();
                                          CharSequence text = getString(R.string.imageSaved);
                                          int duration = Toast.LENGTH_SHORT;
                                          Toast toast = Toast.makeText(context, text, duration);
                                          toast.setGravity(Gravity.CENTER, 0, 0);
                                          toast.show();
                                          finish();
                                          overridePendingTransition(0,0);
                                   }
                                })
```

```
                                .setNegativeButton(getString(R.string.cancel), new
DialogInterface.OnClickListener() {
                                    public void onClick(DialogInterface dialog, int id) {
                                        dialog.cancel();
                                        finish();
                                          overridePendingTransition(0,0);
                                    }
                                });
                            AlertDialog alert = builder.create();
                            alert.show();
                        } else {
                            //Erxomai apo thn UploadScreen
                            //Lew apla na elegksei th syndesh
                            //(profanws de tou zhtaw an thelei na apothikeusei thn eikona afou hdh thn
exei sthn SD)
                            AlertDialog.Builder builder = new AlertDialog.Builder(this);
                            builder.setTitle(getString(R.string.noInternet));
                            builder.setMessage(getString(R.string.checkConnection))
                                .setCancelable(true)
                                .setOnCancelListener(new DialogInterface.OnCancelListener(){
                                        public void onCancel(DialogInterface dialog) {
                                                finish();
                                                overridePendingTransition(0,0);
                                        }
                                })
                                .setPositiveButton(getString(R.string.ok), new
DialogInterface.OnClickListener() {
                                    public void onClick(DialogInterface dialog, int id) {

                                        dialog.cancel();
                                        finish();
                                          overridePendingTransition(0,0);
                                    }
                                });
                            AlertDialog alert = builder.create();
                            alert.show();
                        }

                    } else {
                        //Exei internet ara to stelnw
                        myTask = new UploadingTask();
                        myTask.execute();
                    }

                } finally {}
    }

    public boolean networkAvailable() {
            ConnectivityManager cn = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
            NetworkInfo ni = cn.getActiveNetworkInfo();
            if (ni==null)
               return false;
            return ni.isConnected();
    }

    public String findFileName() { //vriskei to onoma pou prepei na onomastei to arxeio
       File folder = new File(Environment.getExternalStorageDirectory() + "/VIRaL Captures");
              String[] fileNames = folder.list();
              String number;
              if (fileNames==null) {
```

```
                    //An o fakelos einai adeios
                    number="0001";
            } else {
                    int intNumber=0;
                    int max=0;
            int i;
            //find max
            int fileNamesLength = fileNames.length;
            for (i=0; i<fileNamesLength; i++) {
                    if (fileNames[i].contains("ViralImage")) {
                            number=fileNames[i].substring(11, fileNames[i].length()-4);
                            intNumber=Integer.valueOf(number);
                            if (intNumber>max) max=intNumber;

                    }
            }
            if (max>=9999) {
                    intNumber=max+1;
                    number=Integer.toString(intNumber);
            } else if (max==0) {
                    number="0001";
            } else {
                    //o max anhkei sto [1,998]
                    intNumber=max+1;
                    number=Integer.toString(intNumber);
                    //elegxw to length, diladi ta pshfia
                    int numLen=number.length();
                    if (numLen==1) number="000"+number;
                    else if (numLen==2) number="00"+number;
                    else if (numLen==3) number="0"+number;
                    //else tha exei 4 pshfia dld number=number
            }
            }
            String filenameSd="ViralImage_"+number+".jpg";
            return filenameSd;
}

private class UploadingTask extends AsyncTask<Void, Void,String> {

   protected void onPreExecute() {
         uploadingDialog = new ProgressDialog(SendScreen.this);
       uploadingDialog.setMessage(getString(R.string.upDialog));
       uploadingDialog.setCancelable(true);
       uploadingDialog.setOnCancelListener(new DialogInterface.OnCancelListener(){
          public void onCancel(DialogInterface dialog) {
             try {
                   myTask.cancel(false);
             }
             catch (NullPointerException e) {
                   e.printStackTrace();
             }
             finish();
             overridePendingTransition(0,0);
          }
       });
       uploadingDialog.show();
   }

   protected String doInBackground(Void... params) {
                           //Random number from 1000000 to 9999999
              double rand = 1000000 + Math.random()*8999999;
```

```
        int intrand = (int) Math.round(rand);
        filename = "and-"+intrand+".jpg";
        //Save image to cache
    String result = "";
        f = new File(getCacheDir()+"viral_image.jpg");
        FileOutputStream out;
                try {
                        out = new FileOutputStream(f);
            bitmap.compress(Bitmap.CompressFormat.JPEG, 90, out);
            out.close();
                } catch (FileNotFoundException e1) {
                        e1.printStackTrace();
                } catch (IOException e2) {
                        e2.printStackTrace();
                }

  //FTP connection
  FTPClient ftp = new FTPClient();
  String hostname="xxx.xxx.xxx.xxx";
        String username="xxxxx";
        String password="xxxxx";
        if (!myTask.isCancelled()) {
            try {
                        ftp.connect(hostname);
                int reply = ftp.getReplyCode();
                if (!FTPReply.isPositiveCompletion(reply)) {
                   ftp.disconnect();
                }
                else if (!ftp.login(username, password)) {
                   ftp.logout();
                }
                else {
                  //Set the properties for the transfer
                   ftp.setFileType(FTP.BINARY_FILE_TYPE);  //Binary transfer
                   ftp.enterLocalPassiveMode();  //Use passive mode
                   //Transfer the image and close the connection
                   FileInputStream input = new FileInputStream(f);
                   ftp.storeFile(filename, input);
                   input.close();
                }
             }
             catch (FTPConnectionClosedException e) {
          e.printStackTrace();
          if (!myTask.isCancelled()) {
                   //an den exei ginei cancel kai petaksei exception edw
                   //tote kanton cancel (shmainei oti den exei internet
                   //h exei pesei o server) alla de mporw na petaksw
                   //toast edw
                   myTask.cancel(false);
                   uploadingDialog.dismiss();
                   finish();
             overridePendingTransition(0,0);
          }
             }
             catch (IOException x) {
          x.printStackTrace();
          if (!myTask.isCancelled()) {
                   //an den exei ginei cancel kai petaksei exception edw
                   //tote kanton cancel
                   myTask.cancel(false);
                   uploadingDialog.dismiss();
```

```
                        finish();
                    overridePendingTransition(0,0);
                }
                    }
                    finally {
                try {
                    if (ftp.isConnected()) {
                        ftp.disconnect();
                    }
                }
                catch (IOException io) {
                    io.printStackTrace();
                    if (!myTask.isCancelled()) {
                                //an den exei ginei cancel kai petaksei exception edw
                                //tote kanton cancel
                                myTask.cancel(false);
                                uploadingDialog.dismiss();
                                finish();
                            overridePendingTransition(0,0);
                        }
                }
                    }
            }
            //FTP connection closed

            //SOCKET
            if (!myTask.isCancelled()) {
                try {
                        Socket sock = new Socket(hostname, xxxx);
                    PrintWriter out2 = new PrintWriter(sock.getOutputStream(), true);
                    BufferedReader in = new BufferedReader(new
InputStreamReader(sock.getInputStream()));

                    out2.println("AQ " + filename);
                    String line = "";
                    while ((line = in.readLine()) != null) {
                        result += line;
                    }
                    out2.close();
                    in.close();
                    sock.close();
                }
                    catch (Exception e) {
                e.printStackTrace();
                if (!myTask.isCancelled()) {
                        //an den exei ginei cancel kai petaksei exception edw
                        //tote kanton cancel
                        myTask.cancel(false);
                        uploadingDialog.dismiss();
                        finish();
                    overridePendingTransition(0,0);
                }
                    }
            }
            //SOCKET closed


            if (myTask.isCancelled()) {
                //an gia kapoio logo egine cancelled o task,
                //epestrepse keno string wste na mhn proxwrhsei
                //se allh othonh
```

```
                return ("");
            } else {
                //alliws epestrepse auto pou vrhkes
                    return result;
            }
    }

    protected void onPostExecute(String result) {

        if (result==null || result.equals("")) {
                //an gia kapoio logo, exei petaksei kapoio exception
                //o kwdikas, tote to result de tha parei timh
                //ara mhn proxwrhseis screen
        } else {
                //An exei syndethei o xrhsths tha prepei na apothikeuei ta
                //stoixeia sthn database
                if (checkCookie()) {
                        //vres to username tou xrhsth
                        String username = findUsername();
                        //ftiakse to byte[] apo to bitmap
                                //(den xrhsimopoiw to etoimo byte[] sthn arxh tou kwdika
giati ekei meta mporei na to kanw rotate)
                        ByteArrayOutputStream stream = new ByteArrayOutputStream();
                    bitmap.compress(Bitmap.CompressFormat.JPEG, 90, stream);
                    byte[] dataBitmap = stream.toByteArray();
                        try {
                                //syndesou sth vash
                                        Class.forName("org.postgresql.Driver");
                                        String url =
"jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/xxxxx";
                                        String user = "xxxxx";
                                        String pass = "xxxxx";
                                        Connection conn =
DriverManager.getConnection(url, user, pass);
                                        //vres to max id
                                        int max=0;
                                        String query = "SELECT id FROM
android_history ORDER BY id DESC LIMIT 1;";
                                        Statement st = conn.createStatement();
                                        ResultSet rs = st.executeQuery(query);
                                        while (rs.next()) {
                                                max=rs.getInt("id");

                                        }
                                        rs.close();
                                        st.close();
                                        //apothikeuse ta stoixeia

                                        max++;
                                        String encodedResult =
Base64Coder.encodeString(result);
                                        String update = "INSERT INTO android_history
(id, result, \"user\", query) VALUES ("+max+", '"+encodedResult+"', '"+username+"', ?)";
                                        PreparedStatement preSt =
conn.prepareStatement(update);  //to user to vazw se eisagwgika dioti einai reserved word sthn
postgreSql
                                        preSt.setBytes(1, dataBitmap);
                                        preSt.executeUpdate();
                                        preSt.close();
                                        //close connection
                                        conn.close();
```

```
                                        } catch (ClassNotFoundException e) {
                                                e.printStackTrace();
                                        } catch (SQLException e) {
                                                e.printStackTrace();
                                        }

                        }
                        uploadingDialog.dismiss();

        Intent i = new Intent (SendScreen.this, ResultsScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        i.putExtra("result", result);

                        startActivity (i);
                    overridePendingTransition(0,0);
        }
    }
}

public boolean checkCookie() {
        File f = new File(getCacheDir()+"Cookie.viral");
        if (f.exists()) {
                return true;
        } else {
                return false;
        }
}

public String findUsername() {
        String user="";
                BufferedReader in;
                try {
                        in = new BufferedReader(new FileReader(getCacheDir()+"Cookie.viral"));
                        user=in.readLine();
                } catch (FileNotFoundException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                }
                return user;
}

@Override
public void onConfigurationChanged(Configuration newConfig) {
   super.onConfigurationChanged(newConfig);
}

@Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                 if (mainLay.getVisibility()==View.GONE) {
                        //set visible & enable buttons
                        mainLay.setVisibility(View.VISIBLE);

                        buttonCamera = (Button) findViewById(R.id.buttonCamera);
                    buttonCamera.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (SendScreen.this, HomeScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
```

```
                        });

                        buttonUpload = (Button) findViewById(R.id.buttonUpload);
                        buttonUpload.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (SendScreen.this, UploadScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                              overridePendingTransition(0,0);
                          }
                        });

                        buttonLogin = (Button) findViewById(R.id.buttonLogin);
                        buttonLogin.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (SendScreen.this, LoginScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                              overridePendingTransition(0,0);
                          }
                        });

                        buttonHistory = (Button) findViewById(R.id.buttonHistory);
                        buttonHistory.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (SendScreen.this, HistoryScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                              overridePendingTransition(0,0);
                          }
                        });

                        buttonAbout = (Button) findViewById(R.id.buttonAbout);
                        buttonAbout.setOnClickListener(new OnClickListener() {
                          public void onClick(View v) {
                                Intent i = new Intent (SendScreen.this, AboutScreen.class);
                                i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                startActivity (i);
                              overridePendingTransition(0,0);
                          }
                        });

                    }
                    else {
                            mainLay.setVisibility(View.GONE);
                    }
                    return true;
            }

    @Override
    public void onBackPressed() {
            finish();
            overridePendingTransition(0,0);
    }
}
```

UploadScreen.java

```
package com.ntua.image.viral;
```

```
import java.io.ByteArrayOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;

import android.app.Activity;
import android.content.ContentResolver;
import android.content.Context;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.net.Uri;
import android.os.Bundle;
import android.view.Gravity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Toast;

public class UploadScreen extends Activity {

        Button buttonCamera;
        Button buttonUpload;
        Button buttonLogin;
        Button buttonHistory;
        Button buttonAbout;
        Button buttonBrowse;
        Button buttonUploadImage;
        Bitmap bitmap;
        ImageView imageView1;
        LinearLayout mainLay;
        static final int CHOOSE_FILE = 1;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.uploadscreen);

    mainLay = (LinearLayout) findViewById(R.id.main);
    mainLay.setVisibility(View.GONE);

    imageView1 = (ImageView)findViewById(R.id.imageView1);

    buttonBrowse = (Button) findViewById(R.id.buttonBrowse);
    buttonBrowse.setOnClickListener(new OnClickListener() {
      public void onClick(View v) {
          Intent chooseFile = new Intent(Intent.ACTION_GET_CONTENT);
          chooseFile.setType("image/*");
          Intent c = Intent.createChooser(chooseFile, getString(R.string.choosefile));
          startActivityForResult(c, CHOOSE_FILE);
      }
    });

    buttonUploadImage = (Button) findViewById(R.id.buttonUploadImage);
    buttonUploadImage.setOnClickListener(new OnClickListener() {
      public void onClick(View v) {
```

```
            if (bitmap==null) {
                        //No image selected
                        Context context = getApplicationContext();
                        CharSequence text = getString(R.string.noImage);
                        int duration = Toast.LENGTH_SHORT;
                        Toast toast = Toast.makeText(context, text, duration);
                        toast.setGravity(Gravity.CENTER, 0, 0);
                        toast.show();
            }
            else {
                        //Upload image
                        //Create bitmap
                        ByteArrayOutputStream stream = new ByteArrayOutputStream();
                        bitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
                        byte[] data = stream.toByteArray();
                        //Start SendScreen activity
                        Intent i = new Intent (UploadScreen.this, SendScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        i.putExtra("dataUpload", data);
                        startActivity (i);
                    overridePendingTransition(0,0);
            }
        }
       });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
            if (requestCode == CHOOSE_FILE) {
                        if(resultCode == RESULT_OK){
                                    Uri uri = data.getData();
                            ContentResolver cr = getContentResolver();
                            InputStream in1, in2;
                                                try {
                                                            in1 = cr.openInputStream(uri);
                                                            in2 = cr.openInputStream(uri);
                                                            BitmapFactory.Options o = new
BitmapFactory.Options();
                                                o.inJustDecodeBounds = true;
                                                BitmapFactory.decodeStream(in1, null, o);
                                                //vriskw width kai height tou original image
                                                int width = o.outWidth;
                                                int height = o.outHeight;
                                                int max=width;
                                                if (height>max) max=height;
                                                try {
                                                                        in1.close();
                                                            } catch (IOException e) {
                                                                        e.printStackTrace();
                                                            }
                                                BitmapFactory.Options options = new
BitmapFactory.Options();
                                                //kanw best scaling wste na exw eikona
                                                //me megaluterh diastash apo 512 ews 1024
                                                if (max<=1024) options.inSampleSize=1;
                                                else if (max<=2048) options.inSampleSize=2;
                                                else if (max<=4096) options.inSampleSize=4;
                                                else if (max<=8192) options.inSampleSize=8;
                                                else if (max<=16384) options.inSampleSize=16;
                                                else if (max<=32768) options.inSampleSize=32;
                                                else options.inSampleSize=64;
```

```
                                        //apothikeuw new bitmap
                                    bitmap = BitmapFactory.decodeStream(in2, null, options);

                                    try {
                                            in2.close();
                                        } catch (IOException e) {
                                            e.printStackTrace();
                                        }

                                        if (bitmap != null) {
                                  imageView1.setImageBitmap(bitmap);
                                }
                                    } catch (FileNotFoundException e) {
                                            e.printStackTrace();
                                    }
                            }
        }
  };

  @Override
  public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
  }

  @Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                if (mainLay.getVisibility()==View.GONE) {
                        //set visible & enable buttons
                        mainLay.setVisibility(View.VISIBLE);

                        buttonCamera = (Button) findViewById(R.id.buttonCamera);
                    buttonCamera.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (UploadScreen.this, HomeScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonUpload = (Button) findViewById(R.id.buttonUpload);
                    buttonUpload.getBackground().setColorFilter(Color.rgb(95, 252, 17),
android.graphics.PorterDuff.Mode.MULTIPLY);

                    buttonLogin = (Button) findViewById(R.id.buttonLogin);
                    buttonLogin.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (UploadScreen.this, LoginScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
                       overridePendingTransition(0,0);
                      }
                    });

                    buttonHistory = (Button) findViewById(R.id.buttonHistory);
                    buttonHistory.setOnClickListener(new OnClickListener() {
                      public void onClick(View v) {
                        Intent i = new Intent (UploadScreen.this, HistoryScreen.class);
                        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity (i);
```

```
                        overridePendingTransition(0,0);
                    }
                    });

                    buttonAbout = (Button) findViewById(R.id.buttonAbout);
                    buttonAbout.setOnClickListener(new OnClickListener() {
                     public void onClick(View v) {
                          Intent i = new Intent (UploadScreen.this, AboutScreen.class);
                          i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                          startActivity (i);
                        overridePendingTransition(0,0);
                     }
                    });

                }
                else {
                          mainLay.setVisibility(View.GONE);
                }
                return true;
        }

   @Override
   public void onBackPressed() {
        finish();
        overridePendingTransition(0,0);
   }
}
```

aboutscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">
  <ScrollView android:layout_width="fill_parent" android:layout_height="fill_parent"
android:fillViewport="true">
  <LinearLayout android:layout_width="fill_parent" android:layout_height="wrap_content"
android:orientation="vertical">

  <View android:layout_width="fill_parent" android:layout_height="30dp"/>

  <LinearLayout android:layout_width="fill_parent" android:layout_height="wrap_content"
android:gravity="center">
        <ImageView android:src="@drawable/aboutimage" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:scaleType="fitCenter"
android:adjustViewBounds="true"/>
  </LinearLayout>

  <View android:layout_width="fill_parent" android:layout_height="30dp"/>

  <TextView android:text="@string/aboutHeader"
android:textAppearance="?android:attr/textAppearanceLarge" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_gravity="center"/>

  <View android:layout_width="fill_parent" android:layout_height="30dp"/>

  <TextView android:id="@+id/aboutText" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_gravity="center"/>

  <View android:layout_width="fill_parent" android:layout_height="30dp"/>
```

```
  <TextView android:text="@string/aboutHeader2"
android:textAppearance="?android:attr/textAppearanceMedium"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:layout_gravity="center"/>

  <View android:layout_width="fill_parent" android:layout_height="10dp"/>

  <TextView android:text="@string/viralMobileTeam" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_gravity="center_horizontal"/>

  <View android:layout_width="fill_parent" android:layout_height="30dp"/>

  <TextView android:text="@string/aboutHeader3"
android:textAppearance="?android:attr/textAppearanceMedium"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:layout_gravity="center"/>

  <View android:layout_width="fill_parent" android:layout_height="10dp"/>

  <TextView android:text="@string/viralWebTeam" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_gravity="center_horizontal"/>

  </LinearLayout>
  </ScrollView>

  <include layout="@layout/main"/>

</FrameLayout>
```

historymapscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">

  <RelativeLayout android:layout_width="fill_parent" android:layout_height="fill_parent"
android:gravity="center_horizontal">

                <com.google.android.maps.MapView
        android:id="@+id/historyMapView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="xxxxx"
        />

        <RelativeLayout android:layout_height="wrap_content" android:layout_width="fill_parent"
android:gravity="center_horizontal">
        <ZoomControls android:id="@+id/historyzoomcontrols"
                        android:layout_height="wrap_content"
android:layout_width="wrap_content"/>
        </RelativeLayout>

  </RelativeLayout>

  <include layout="@layout/main"/>

</FrameLayout>
```

historyscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">

  <ScrollView android:layout_width="fill_parent" android:layout_height="fill_parent"
android:fillViewport="true">
  <LinearLayout android:id="@+id/history1" android:layout_width="fill_parent"
android:layout_height="fill_parent" android:orientation="vertical">

  </LinearLayout>
  </ScrollView>

  <RelativeLayout android:layout_width="fill_parent" android:layout_height="wrap_content"
android:layout_gravity="bottom" android:gravity="center">
  <LinearLayout android:layout_width="wrap_content" android:layout_height="wrap_content">
        <Button android:id="@+id/buttonHistoryMap" android:drawableLeft="@drawable/map2"
android:text="@string/historyMap" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_weight="1"/>
        <Button android:id="@+id/buttonClearHistory"
android:drawableLeft="@drawable/clearhistory" android:text="@string/clearHistory"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:layout_weight="1"/>
  </LinearLayout>
  </RelativeLayout>

  <include layout="@layout/main"/>

</FrameLayout>
```

homescreen.xml

```
<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">

<SurfaceView
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:id="@+id/camerasurface"
  />

<LinearLayout android:id="@+id/linearLayout2" android:layout_height="fill_parent"
android:layout_gravity="right" android:layout_width="wrap_content">
   <ImageView android:id="@+id/buttonFlash" android:layout_gravity="top"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:src="@drawable/flashoff" android:paddingRight="5dp"
android:paddingTop="15dp"></ImageView>
</LinearLayout>
<LinearLayout android:id="@+id/linearLayout1" android:layout_height="fill_parent"
android:layout_gravity="right" android:layout_width="wrap_content">
   <Button android:drawableTop="@drawable/snapshot" android:id="@+id/buttonClick"
android:layout_gravity="center" android:layout_width="wrap_content"
android:background="@android:color/transparent" android:layout_height="wrap_content"
android:paddingRight="5dp"></Button>
</LinearLayout>
<LinearLayout android:layout_height="fill_parent" android:layout_gravity="right"
android:layout_width="wrap_content">
```

```
   <Button android:drawableTop="@drawable/zoomout" android:id="@+id/buttonZoomOut"
android:layout_gravity="bottom" android:layout_width="wrap_content"
android:background="@android:color/transparent" android:layout_height="wrap_content"
android:paddingRight="10dp"></Button>
   <Button android:drawableTop="@drawable/zoomin" android:id="@+id/buttonZoomIn"
android:layout_gravity="bottom" android:layout_width="wrap_content"
android:background="@android:color/transparent" android:layout_height="wrap_content"
android:paddingRight="5dp"></Button>
</LinearLayout>

<HorizontalScrollView android:fillViewport="true"
xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:layout_gravity="bottom">
<LinearLayout android:weightSum="5" android:background="@android:color/transparent"
android:layout_alignParentTop="true" android:layout_width="fill_parent"
android:layout_gravity="bottom" android:id="@+id/mainLayout"
android:layout_height="wrap_content">
   <Button android:layout_weight="1" android:drawableLeft="@drawable/camera5"
android:text="@string/camerabutton" android:id="@+id/buttonCamera"
android:layout_width="110dp" android:layout_height="wrap_content"></Button>
   <Button android:layout_weight="1" android:drawableLeft="@drawable/upload4"
android:text="@string/uploadbutton" android:id="@+id/buttonUpload" android:layout_width="110dp"
android:layout_height="wrap_content"></Button>
   <Button android:layout_weight="1" android:drawableLeft="@drawable/login3"
android:text="@string/loginbutton" android:id="@+id/buttonLogin" android:layout_width="110dp"
android:layout_height="wrap_content"></Button>
   <Button android:layout_weight="1" android:drawableLeft="@drawable/history3"
android:text="@string/historybutton" android:id="@+id/buttonHistory" android:layout_width="110dp"
android:layout_height="wrap_content"></Button>
   <Button android:layout_weight="1" android:drawableLeft="@drawable/about1"
android:text="@string/aboutbutton" android:id="@+id/buttonAbout" android:layout_width="110dp"
android:layout_height="wrap_content"></Button>
</LinearLayout>
</HorizontalScrollView>

<ImageView android:src="@drawable/frame" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:scaleType="fitCenter"
android:adjustViewBounds="true" android:layout_gravity="center"/>

</FrameLayout>
```

imagefullscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">

  <LinearLayout android:layout_width="fill_parent" android:layout_height="fill_parent"
android:orientation="vertical">

  <RelativeLayout android:orientation="horizontal" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:gravity="center_horizontal">
        <ImageView android:id="@+id/imageViewFlickr" android:src="@drawable/flickr"
android:layout_weight="1" android:layout_gravity="center_horizontal"
android:layout_height="wrap_content" android:layout_width="wrap_content"></ImageView>
  </RelativeLayout>

  <ScrollView android:layout_width="fill_parent" android:layout_height="fill_parent"
android:fillViewport="true">
```

```
  <ImageView android:id="@+id/imageViewFullScreen" android:src="@drawable/blank"
android:layout_width="fill_parent" android:layout_height="fill_parent" android:scaleType="fitCenter"
android:adjustViewBounds="true">
  </ImageView>

  </ScrollView>

  </LinearLayout>

  <ProgressBar
                    android:id="@+id/progressBarLarge"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       style="@android:style/Widget.ProgressBar.Large"
       android:layout_gravity="center"/>

  <include layout="@layout/main"/>

</FrameLayout>
```

loginscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">

  <ImageView android:src="@drawable/ivml" android:layout_gravity="center_horizontal"
android:layout_height="wrap_content" android:layout_width="wrap_content">

  </ImageView>

  <RelativeLayout android:id="@+id/accountLayout" android:layout_width="fill_parent"
android:layout_height="fill_parent" android:gravity="center">

          <LinearLayout android:layout_width="fill_parent" android:layout_height="wrap_content"
android:orientation="vertical" android:weightSum="6">

                    <LinearLayout android:layout_width="fill_parent"
android:layout_height="wrap_content" android:orientation="horizontal" android:layout_weight="1">

                              <TextView android:text="@string/accountDesc"
android:id="@+id/textViewAccountDesc" android:layout_width="wrap_content"
android:layout_height="wrap_content"></TextView>

                    </LinearLayout>

                    <LinearLayout android:layout_width="fill_parent"
android:layout_height="wrap_content" android:orientation="horizontal" android:layout_weight="1">

                              <View
                                        android:id="@+id/emptyView"
                    android:layout_width="fill_parent"
                    android:layout_height="30dp">
                    </View>

                    </LinearLayout>

                    <LinearLayout android:layout_width="fill_parent"
android:layout_height="wrap_content" android:orientation="horizontal" android:layout_weight="1"
android:weightSum="3">
```

```
                    <TextView android:text="@string/username"
android:layout_height="wrap_content" android:textAppearance="?android:attr/textAppearanceLarge"
android:layout_width="170dp" android:id="@+id/textView1"></TextView>
                    <EditText android:id="@+id/usernameTextField"
android:layout_height="wrap_content" android:layout_weight="2.5" android:layout_width="90dp"
android:singleLine="true">
                        <requestFocus></requestFocus>
                    </EditText>

                </LinearLayout>

                <LinearLayout android:layout_width="fill_parent"
android:layout_height="wrap_content" android:orientation="horizontal" android:layout_weight="1"
android:weightSum="3">

                    <TextView android:text="@string/password"
android:layout_height="wrap_content" android:textAppearance="?android:attr/textAppearanceLarge"
android:layout_width="170dp" android:id="@+id/textView2"></TextView>
                    <EditText android:id="@+id/passwordTextField"
android:layout_height="wrap_content" android:layout_weight="2.5" android:layout_width="90dp"
android:inputType="textPassword" android:singleLine="true"></EditText>

                </LinearLayout>

                <LinearLayout android:layout_width="fill_parent"
android:layout_height="wrap_content" android:orientation="horizontal" android:layout_weight="1">

                    <View
                android:layout_width="fill_parent"
                android:layout_height="30dp">
                </View>

                </LinearLayout>

                <LinearLayout android:layout_width="fill_parent"
android:layout_height="wrap_content" android:orientation="horizontal" android:layout_weight="1">

                    <RelativeLayout android:layout_width="fill_parent"
android:layout_height="wrap_content" android:layout_gravity="center" android:gravity="center">
                    <Button android:layout_height="wrap_content"
android:text="@string/logIn" android:textSize="20dp" android:id="@+id/buttonUserLogIn"
android:layout_gravity="center" android:layout_width="wrap_content"></Button>
                    </RelativeLayout>

                </LinearLayout>

            </LinearLayout>

  </RelativeLayout>

  <include layout="@layout/main"/>

</FrameLayout>
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<HorizontalScrollView  xmlns:android="http://schemas.android.com/apk/res/android"
android:fillViewport="true" android:layout_width="fill_parent" android:layout_height="wrap_content"
android:layout_gravity="bottom">
<LinearLayout android:weightSum="5" android:background="#B5BCB4"
android:layout_alignParentTop="true" android:layout_width="fill_parent"
android:layout_gravity="bottom" android:id="@+id/main" android:layout_height="wrap_content">
   <Button android:layout_weight="1" android:drawableTop="@drawable/camera5"
android:text="@string/camerabutton" android:id="@+id/buttonCamera" android:layout_width="75dp"
android:layout_height="wrap_content"></Button>
   <Button android:layout_weight="1" android:drawableTop="@drawable/upload4"
android:text="@string/uploadbutton" android:id="@+id/buttonUpload" android:layout_width="75dp"
android:layout_height="wrap_content"></Button>
   <Button android:layout_weight="1" android:drawableTop="@drawable/login3"
android:text="@string/loginbutton" android:id="@+id/buttonLogin" android:layout_width="75dp"
android:layout_height="wrap_content"></Button>
   <Button android:layout_weight="1" android:drawableTop="@drawable/history3"
android:text="@string/historybutton" android:id="@+id/buttonHistory" android:layout_width="75dp"
android:layout_height="wrap_content"></Button>
   <Button android:layout_weight="1" android:drawableTop="@drawable/about1"
android:text="@string/aboutbutton" android:id="@+id/buttonAbout" android:layout_width="75dp"
android:layout_height="wrap_content"></Button>
</LinearLayout>
</HorizontalScrollView>
```

resmapscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">

        <RelativeLayout android:layout_width="fill_parent" android:layout_height="fill_parent"
android:gravity="center_horizontal">

                <com.google.android.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="xxxxx"
        />

        <RelativeLayout android:layout_height="wrap_content" android:layout_width="fill_parent"
android:gravity="center_horizontal">
        <ZoomControls android:id="@+id/zoomcontrols"
                        android:layout_height="wrap_content"
android:layout_width="wrap_content"/>
        </RelativeLayout>

   </RelativeLayout>

</FrameLayout>
```

resoriginalscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">
```

```
  <ScrollView android:layout_width="fill_parent" android:layout_height="fill_parent">
  <LinearLayout android:layout_width="fill_parent" android:layout_height="wrap_content"
android:orientation="vertical">

  <TableLayout android:id="@+id/tableLayout1" android:layout_width="fill_parent"
android:layout_height="wrap_content">
  <TableRow android:layout_width="fill_parent" android:gravity="center_horizontal">
  <ImageView android:id="@+id/imageViewOriginal" android:src="@drawable/blank"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:scaleType="fitCenter" android:adjustViewBounds="true"></ImageView>
  </TableRow>
  </TableLayout>

  <TableLayout android:layout_width="fill_parent" android:layout_height="wrap_content">

  <TableRow>
  <LinearLayout android:layout_width="wrap_content" android:layout_height="wrap_content"
android:orientation="horizontal">
  <TextView android:text="@string/location" android:id="@+id/TextViewLocation"
android:layout_width="wrap_content" android:layout_height="wrap_content"/>
  <TextView android:text=" " android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
  <TextView android:text="" android:id="@+id/TextViewLoc" android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
  </LinearLayout>
  </TableRow>

  <TableRow>
  <LinearLayout android:layout_width="wrap_content" android:layout_height="wrap_content"
android:orientation="horizontal">
  <TextView android:text="@string/sTags" android:id="@+id/TextViewSuggestedTags"
android:layout_width="wrap_content" android:layout_height="wrap_content"/>
  <TextView android:text=" " android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
  <TextView android:text="" android:id="@+id/TextViewStags" android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
  </LinearLayout>
  </TableRow>

  <TableRow>
  <LinearLayout android:layout_width="wrap_content" android:layout_height="wrap_content"
android:orientation="horizontal">
  <TextView android:text="@string/fTags" android:id="@+id/TextViewFrequentTags"
android:layout_width="wrap_content" android:layout_height="wrap_content"/>
  <TextView android:text=" " android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
  <TextView android:text="" android:id="@+id/TextViewFtags" android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
  </LinearLayout>
  </TableRow>

  </TableLayout>

  <RelativeLayout android:layout_width="wrap_content" android:layout_height="wrap_content"
android:layout_gravity="left" android:gravity="bottom">
     <ImageView android:id="@+id/imageViewSaveImage" android:src="@drawable/save"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:paddingLeft="5dp"></ImageView>
  </RelativeLayout>

  </LinearLayout>
```

```
  </ScrollView>

</FrameLayout>
```

ressimilarscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">
  <ScrollView android:layout_width="fill_parent" android:layout_height="fill_parent"
android:fillViewport="true">
  <LinearLayout android:layout_width="fill_parent" android:layout_height="fill_parent"
android:orientation="vertical">

        <LinearLayout android:id="@+id/similar1" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:weightSum="4" android:orientation="horizontal">
        </LinearLayout>
        <LinearLayout android:id="@+id/similar2" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:weightSum="4" android:orientation="horizontal">
        </LinearLayout>
        <LinearLayout android:id="@+id/similar3" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:weightSum="4" android:orientation="horizontal">
        </LinearLayout>
        <LinearLayout android:id="@+id/similar4" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:weightSum="4" android:orientation="horizontal">
        </LinearLayout>

  </LinearLayout>
  </ScrollView>
</FrameLayout>
```

ressimofsimscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">
  <ScrollView android:layout_width="fill_parent" android:layout_height="fill_parent"
android:fillViewport="true">
  <LinearLayout android:layout_width="fill_parent" android:layout_height="fill_parent"
android:orientation="vertical">

        <LinearLayout android:id="@+id/sos1" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:weightSum="4" android:orientation="horizontal">
        </LinearLayout>
        <LinearLayout android:id="@+id/sos2" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:weightSum="4" android:orientation="horizontal">
        </LinearLayout>
        <LinearLayout android:id="@+id/sos3" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:weightSum="4" android:orientation="horizontal">
        </LinearLayout>
        <LinearLayout android:id="@+id/sos4" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:weightSum="4" android:orientation="horizontal">
        </LinearLayout>

  </LinearLayout>
  </ScrollView>
</FrameLayout>
```

sendscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">

  <ImageView android:id="@+id/imageSendscreen" android:src="@drawable/blank"
android:layout_width="fill_parent" android:layout_height="fill_parent" android:scaleType="fitCenter"
android:adjustViewBounds="true">
  </ImageView>

  <include layout="@layout/main"/>

</FrameLayout>
```

tabs.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
   android:id="@android:id/tabhost"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
   <LinearLayout
      android:orientation="vertical"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:padding="5dp">
      <TabWidget
         android:id="@android:id/tabs"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content" />
      <FrameLayout
         android:id="@android:id/tabcontent"
         android:layout_width="fill_parent"
         android:layout_height="fill_parent"
         android:padding="5dp">
       </FrameLayout>
   </LinearLayout>
   <include layout="@layout/main"/>
</TabHost>
```

uploadscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent" android:layout_height="fill_parent">

  <TableLayout android:id="@+id/tableLayout1" android:layout_width="fill_parent"
android:layout_height="wrap_content">
     <LinearLayout android:orientation="horizontal" android:id="@+id/linearLayout1"
android:layout_height="wrap_content" android:weightSum="2" android:baselineAligned="false"
android:layout_width="fill_parent">
        <Button android:drawableLeft="@drawable/browse" android:text="@string/browsebutton"
android:id="@+id/buttonBrowse" android:layout_weight="1" android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
        <Button android:drawableLeft="@drawable/upload5"
android:text="@string/uploadimagebutton" android:id="@+id/buttonUploadImage"
android:layout_weight="1" android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
     </LinearLayout>
```

```
    <ScrollView>
    <ImageView android:id="@+id/imageView1" android:src="@drawable/noimage"
android:layout_width="fill_parent" android:layout_height="fill_parent" android:scaleType="fitCenter"
android:adjustViewBounds="true"></ImageView>
    </ScrollView>
  </TableLayout>

  <include layout="@layout/main"/>

</FrameLayout>
```

res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <string name="app_name">VIRaL</string>
   <string name="camerabutton">Camera</string>
   <string name="uploadbutton">Upload</string>
   <string name="loginbutton">Account</string>
   <string name="historybutton">History</string>
   <string name="aboutbutton">About</string>
   <string name="browsebutton">Browse image</string>
   <string name="uploadimagebutton">Upload image</string>
   <string name="choosefile">Choose image</string>
   <string name="noImage">No image selected</string>
   <string name="noInternet">No Internet Connection</string>
   <string name="askSaveImage">Do you want to save the captured image for later use?</string>
   <string name="saveToSd">Save to SD card</string>
   <string name="cancel">Cancel</string>
   <string name="imageSaved">Image successfully saved</string>
   <string name="checkConnection">Please check your Internet connection and try again</string>
   <string name="ok">OK</string>
   <string name="upDialog">Uploading Image...</string>
   <string name="noSimImages">No similar images found</string>
   <string name="restab1">Original</string>
   <string name="restab2">Map</string>
   <string name="restab3">Similar</string>
   <string name="restab4">Similar of similar</string>
   <string name="estimatedLoc">Estimated Location</string>
   <string name="similarImage">Similar Image</string>
   <string name="location">Where:</string>
   <string name="sTags">What:</string>
   <string name="fTags">People say:</string>
   <string name="loadDialog">Loading Images...</string>
   <string name="noSimofsimImages">No similar of similar images found</string>
   <string name="saveImage">Save image?</string>
   <string name="includeCoordinates">Coordinates</string>
   <string name="includeTags">Tags</string>
   <string name="accountDesc">Enter your username and password, so that you can save your images
history online. Register for your free account at http://viral.image.ntua.gr</string>
   <string name="username">Username:</string>
   <string name="password">Password:</string>
   <string name="logIn">Log in</string>
   <string name="loggingIn">Logging in...</string>
   <string name="wrongUser">Incorrect username or password</string>
   <string name="currentUser">You are currently logged in as</string>
   <string name="logOut">Log out</string>
   <string name="pleaseLogin">Please log in to view your images history</string>
   <string name="clearHistory">Clear history</string>
   <string name="askClearHistory">Are you sure you want to clear history?</string>
```

```
   <string name="clearingDialog">Clearing history...</string>
   <string name="yes">Yes</string>
   <string name="historyMap">History Map</string>
   <string name="loadingMap">Loading Map...</string>
   <string name="aboutHeader">VIRaL Mobile</string>
   <string name="aboutText">VIRaL Mobile is an Android application that uses the content-based
image search engine</string>
   <string name="viral20">VIRaL 2.0</string>
   <string name="aboutHeader2">VIRaL Mobile team:</string>
   <string name="viralMobileTeam">Stelios Frantzeskakis\nPhivos Mylonas</string>
   <string name="aboutHeader3">VIRaL 2.0 team:</string>
   <string name="viralWebTeam">Yannis Avrithis\nYannis Kalantidis\nPhivos Mylonas\nMarios
Phinikettos\nEvaggelos Spyrou\nGiorgos Tolias</string>
</resources>
```

res/values-el/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <string name="app_name">VIRaL</string>
   <string name="camerabutton">Κάμερα</string>
   <string name="uploadbutton">Μεταφόρτωση</string>
   <string name="loginbutton">Λογαριασμός</string>
   <string name="historybutton">Ιστορικό</string>
   <string name="aboutbutton">Σχετικά</string>
   <string name="browsebutton">Αναζήτηση εικόνας</string>
   <string name="uploadimagebutton">Μεταφόρτωση εικόνας</string>
   <string name="choosefile">Επιλέξτε εικόνα</string>
   <string name="noImage">Δεν έχετε επιλέξει εικόνα</string>
   <string name="noInternet">Δεν έχετε συνδεθεί στο διαδίκτυο</string>
   <string name="askSaveImage">Θέλετε να αποθηκεύσετε την εικόνα για μετέπειτα χρήση;</string>
   <string name="saveToSd">Αποθήκευση στην κάρτα SD</string>
   <string name="cancel">Άκυρο</string>
   <string name="imageSaved">Η εικόνα αποθηκεύτηκε επιτυχώς</string>
   <string name="checkConnection">Παρακαλούμε ελέγξτε τη σύνδεσή σας με το διαδίκτυο και
προσπαθήστε ξανά</string>
   <string name="ok">OK</string>
   <string name="upDialog">Μεταφόρτωση εικόνας...</string>
   <string name="noSimImages">Δε βρέθηκαν παρόμοιες εικόνες</string>
   <string name="restab1">Η εικόνα μου</string>
   <string name="restab2">Χάρτης</string>
   <string name="restab3">Παρόμοιες</string>
   <string name="restab4">Παραπλήσιες</string>
   <string name="estimatedLoc">Εκτιμώμενη τοποθεσία</string>
   <string name="similarImage">Παρόμοια εικόνα</string>
   <string name="location">Που είναι:</string>
   <string name="sTags">Τι είναι:</string>
   <string name="fTags">Ετικέτες χρηστών:</string>
   <string name="loadDialog">Φόρτωση εικόνων...</string>
   <string name="noSimofsimImages">Δε βρέθηκαν παραπλήσιες εικόνες</string>
   <string name="saveImage">Αποθήκευση εικόνας;</string>
   <string name="includeCoordinates">Συντεταγμένες</string>
   <string name="includeTags">Ετικέτες</string>
   <string name="accountDesc">Εισάγετε το όνομα χρήστη και τον κωδικό σας, ώστε να αποθηκεύετε
το ιστορικό σας online. Εγγραφείτε δωρεάν στην υπηρεσία στο http://viral.image.ntua.gr</string>
   <string name="username">Όνομα χρήστη:</string>
   <string name="password">Κωδικός:</string>
   <string name="logIn">Σύνδεση</string>
   <string name="loggingIn">Σύνδεση...</string>
   <string name="wrongUser">Λάθος όνομα χρήστη ή κωδικός πρόσβασης</string>
```

```
   <string name="currentUser">Έχετε συνδεθεί ως</string>
   <string name="logOut">Αποσύνδεση</string>
   <string name="pleaseLogin">Παρακαλούμε συνδεθείτε στο λογαριασμό σας για να δείτε το ιστορικό σας</string>
   <string name="clearHistory">Διαγραφή ιστορικού</string>
   <string name="askClearHistory">Είστε σίγουροι ότι θέλετε να διαγράψετε το ιστορικό σας;</string>
   <string name="clearingDialog">Διαγραφή ιστορικού...</string>
   <string name="yes">Ναι</string>
   <string name="historyMap">Χάρτης ιστορικού</string>
   <string name="loadingMap">Φόρτωση χάρτη...</string>
   <string name="aboutHeader">VIRaL Mobile</string>
   <string name="aboutText">Το VIRaL Mobile είναι μία εφαρμογή Android η οποία χρησιμοποιεί τη μηχανή αναζήτησης εικόνων, με βάση το περιεχόμενό τους,</string>
   <string name="viral20">VIRaL 2.0</string>
   <string name="aboutHeader2">Ομάδα VIRaL Mobile:</string>
   <string name="viralMobileTeam">Στέλιος Φραντζεσκάκης\nΦοίβος Μυλωνάς</string>
   <string name="aboutHeader3">Ομάδα VIRaL 2.0:</string>
   <string name="viralWebTeam">Γιάννης Αβρίθης\nΓιάννης Καλαντίδης\nΦοίβος Μυλωνάς\nΕυάγγελος Σπύρου\nΓιώργος Τόλιας\nΜάριος Φοινίκεττος</string>
</resources>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ntua.image.viral"
    android:versionCode="1"
    android:versionName="1.0">

   <uses-sdk android:minSdkVersion="7" />

   <uses-permission android:name="android.permission.CAMERA" />
   <uses-permission android:name="android.permission.INTERNET"/>
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

   <uses-feature android:name="android.hardware.camera" />

   <application android:icon="@drawable/icon" android:label="@string/app_name"
android:theme="@style/myStyle">

      <uses-library android:name="com.google.android.maps" />

      <activity android:name=".HomeScreen" android:theme="@style/myHomeStyle"
android:screenOrientation="landscape" android:configChanges="keyboardHidden|orientation"
            android:label="@string/app_name">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
      <activity android:name=".UploadScreen"
android:configChanges="orientation|keyboardHidden"></activity>
      <activity android:name=".LoginScreen"
android:configChanges="orientation|keyboardHidden"></activity>
      <activity android:name=".HistoryScreen"
android:configChanges="orientation|keyboardHidden"></activity>
      <activity android:name=".AboutScreen"
android:configChanges="orientation|keyboardHidden"></activity>
```

```
        <activity android:name=".SendScreen" android:noHistory="true"
android:configChanges="orientation|keyboardHidden"></activity>
        <activity android:name=".ResultsScreen"
android:configChanges="orientation|keyboardHidden"></activity>
        <activity android:name=".ResOriginalScreen"></activity>
        <activity android:name=".ResMapScreen"></activity>
        <activity android:name=".ResSimilarScreen"></activity>
        <activity android:name=".ResSimofsimScreen"></activity>
        <activity android:name=".ImageFullScreen"
android:configChanges="orientation|keyboardHidden"></activity>
        <activity android:name=".HistoryMapScreen"
android:configChanges="orientation|keyboardHidden"></activity>

    </application>
</manifest>
```

Βιβλιογραφία – Ιστοσελίδες

- Cingil I., Dogac A., Azgin A. “A Broader Approach to Personalization”. 2000
- Pluke M., Petersen F., Brown W. “Personalization and User Profile Management for Public Internet Access Points (PIAPs)”
- Gauch S., Speretta M., Chandramouli A., Micarelli A. “User Profiles for Personalized Information Access”. 2007
- Μυλωνάς Φ. “Προσαρμοζόμενη Πρόσβαση σε Πολυμεσικό Περιεχόμενο με βάση το Εννοιολογικό Πλαίσιο και το Προφίλ του Χρήστη”. 2008: http://www.image.ece.ntua.gr/php/savepaper.php?id=593
- Android Developers, The Developer’s Guide: http://developer.android.com/guide/index.html
- Google Groups, Android Developers: http://groups.google.com/group/android-developers
- Github Social Coding: https://github.com/
- Android Development Community: http://www.anddev.org/
- Android Competency Center, JSON Parsing in android. 2009: http://www.androidcompetencycenter.com/2009/10/json-parsing-in-android/
- Wong M. “Android & PostgreSQL”. 2011: http://www.pgcon.org/2011/schedule/attachments/194_pgcon2011-pgdroid.pdf