



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΜΣ: «Μηχανική Λογισμικού για Διαδικτυακές & Φορητές
Εφαρμογές»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη android εφαρμογής για
καλύτερη οργάνωση και διαχείριση
εικόνων και βίντεο και αξιολόγηση ήδη
υπαρχόντων εφαρμογών ίδιου σκοπού

Μπάτσικας Νικόλαος

A.M. 7419013

nikobats2@uth.gr

Επιβλέπων Καθηγητής: Κακαρόντζας Γεώργιος

ΛΑΡΙΣΑ 2021

ΠΕΡΙΕΧΟΜΕΝΑ

Περιεχόμενα	2
Περίληψη	3
Abstract	4
1. Εισαγωγή.....	5
2. Android	7
2.1 Βασικά χαρακτηριστικά και λειτουργίες.....	7
2.2 Η αρχιτεκτονική του Android.....	8
2.3 Πλεονεκτήματα Ανάπτυξης Εφαρμογών Σε Android.....	9
3. Ανάπτυξη Android Εφαρμογής Για Οργάνωση Και Διαχείριση Φωτογραφιών Και Βίντεο	11
3.1 Περιγραφή Συστήματος	11
3.1.1 Περιπτώσεις χρήσεις εφαρμογής	11
3.1.2 Σχεδιαστικές και τεχνικές λεπτομέρειες.....	14
3.1.3 Ανάλυση βασικών κλάσεων και κώδικα.....	22
3.1.4 Βάση δεδομένων.....	38
3.2 Παρουσίαση/Επίδειξη Εκτέλεσης Εφαρμογής	43
3.2.1 Αρχική οθόνη	43
3.2.2 Προβολή φωτογραφίας.....	44
3.2.3 Εισαγωγή ετικετών.....	45
3.2.4 Προβολή βίντεο.....	46
3.2.5 Εισαγωγή τοποθεσίας	47
3.2.6 Προβολή άλμπουμ	48
3.2.7 Μαζική εισαγωγή ημερομηνίας σε πολλαπλά αρχεία.....	49
3.2.8 Αναζήτηση αρχείων.....	50
4. Αξιολόγηση Παρόμοιων εφαρμογών	53
4.1 Google Photos.....	53
4.2 Flickr	54
4.3 Mylio.....	56
4.4 F-Stop Gallery	58
4.5 Gallery Go	59
5. Συμπεράσματα	62
6. Βιβλιογραφία - Αναφορές.....	64

ΠΕΡΙΛΗΨΗ

Οι φορητές συσκευές όπως κινητά τηλέφωνα (smartphones) και tablets, τα τελευταία χρόνια έχουν επικρατήσει ως η πιο δημοφιλής προσωπική συσκευή. Συγκεκριμένα οι φορητές συσκευές οι οποίες χρησιμοποιούν ως λειτουργικό σύστημα το Android εκτιμάται ότι ξεπερνούν το ένα δισεκατομμύριο. Η συνεχής εξάπλωση των Android φορητών συσκευών έχει ανοίξει το δρόμο για την ανάπτυξη εκατοντάδων χιλιάδων εφαρμογών σε αυτές. Μία δημοφιλής κατηγορία εφαρμογών σχετίζεται με τις φωτογραφίες και τα βίντεο. Συγκεκριμένα, υπάρχουν εφαρμογές που παρέχουν λειτουργίες για αποτελεσματικότερη οργάνωση, διαχείριση και επεξεργασία αρχείων φωτογραφιών και βίντεο που αποθηκεύονται στις συσκευές των χρηστών.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η σχεδίαση και η ανάπτυξη μιας Android εφαρμογής η οποία θα προσφέρει εύκολη και αποδοτικότερη οργάνωση, ταξινόμηση και αναζήτηση των φωτογραφιών και των βίντεο που βρίσκονται στον αποθηκευτικό χώρο της συσκευής. Σε θεωρητικό επίπεδο, η συγκεκριμένη εργασία διακρίνεται σε πέντε βασικές ενότητες. Αρχικά, γίνεται μια εισαγωγή στις mobile εφαρμογές και στο λειτουργικό σύστημα Android. Ακολουθεί μια αναφορά στα βασικά χαρακτηριστικά και στην αρχιτεκτονική του, καθώς και στα πλεονεκτήματα που προσφέρει ως λειτουργικό σύστημα στην ανάπτυξη εφαρμογών. Στη συνέχεια, παρουσιάζεται η σχεδίαση και η υλοποίηση της εφαρμογής που αναπτύχθηκε για τη διπλωματική εργασία. Αναλύονται οι περιπτώσεις χρήσεις της εφαρμογής και ορισμένες τεχνικές λεπτομέρειες όπως βιβλιοθήκες, εργαλεία και APIs που χρησιμοποιήθηκαν. Επίσης γίνεται αναφορά σε αξιοσημείωτες κλάσεις και σημεία του κώδικα που καθόρισαν τη μορφή και τη λειτουργικότητα της εφαρμογής. Στη συνέχεια γίνεται μια επίδειξη εκτέλεσης της εφαρμογής μέσω εικόνων και επεξηγήσεων, στην οποία θα παρουσιαστούν οι δυνατότητες και τα χαρακτηριστικά της. Στο επόμενο κεφάλαιο, γίνεται μια ανασκόπηση παρόμοιων Android εφαρμογών οργάνωσης και διαχείρισης φωτογραφιών και βίντεο που είναι σήμερα διαθέσιμες, ώστε να αξιολογηθούν και να συγκριθούν οι δυνατότητες και οι λειτουργίες τους. Στο τελευταίο κεφάλαιο, γίνεται μια σύντομη αναφορά σε ορισμένες λειτουργίες και χαρακτηριστικά που θα μπορούσαν να προστεθούν μελλοντικά στην εφαρμογή που υλοποιήθηκε για την παρούσα εργασία και αναλύονται τα συμπεράσματα που αποκομίστηκαν τόσο κατά την ανάπτυξή της, όσο και από την μελέτη παρόμοιων εφαρμογών ίδιου σκοπού.

ABSTRACT

Mobile devices such as mobile phones (smartphones) and tablets, have prevailed as the most popular personal device in the past few years. It is estimated, that there are more than one billion mobile devices that use Android as an operating system. The constant growth of Android mobile devices has paved the way for the development of hundreds of thousands mobile applications. One of the most popular categories of mobile applications is related with photography. These applications, usually provide features and tools for organizing, managing and editing photo and video files efficiently.

The purpose of this paper is the design and development of an Android application which will provide effective organization, sorting and searching of photos and videos stored in a mobile device. The present paper is divided into five main sections. The introduction refers briefly to mobile devices and Android operating system. Following, is a reference to its basic features and architecture, as well as its advantages in mobile application development. Next, follows an analysis about the design and the implementation of the application which was developed for the present paper, including use cases and some technical details such as libraries, tools and APIs used. Furthermore, reference is made to important classes and parts of the code that defined the final form and the functionality of the application. Then, follows a presentation of the application through images and clarifications, in order to demonstrate its features and capabilities. The next section, presents a review of similar Android applications available today about photo and video organizing, in order to compare and evaluate their features and functionalities. In the last section, a brief reference is made to certain features that could be added later to the application of the present paper. Last but not least, the conclusions reached from the developing process of the application and the study of similar-same-purpose applications are analysed.

1. ΕΙΣΑΓΩΓΗ

Οι φορητές συσκευές παρουσιάζουν μία τεράστια εξάπλωση και σταθερή άνοδο την τελευταία δεκαετία. Ιδιαίτερα τα smartphones, αποτελούν τη δημοφιλέστερη προσωπική συσκευή καθώς πλέον χρησιμοποιούνται περισσότερο και από υπολογιστές. Η εξέλιξη της τεχνολογίας τόσο σε επίπεδο λογισμικού (software) όσο και υλικού (hardware), έχει βοηθήσει στην μετάβαση των κινητών τηλεφώνων από απλές συσκευές επικοινωνίας, σε “έξυπνες” φορητές συσκευές με δυνατότητες προσωπικού υπολογιστή. Πλέον, τα κινητά τηλέφωνα φέρουν στο υλικό τους χαρακτηριστικά τα οποία συναντούσαμε στο πρόσφατο παρελθόν σε προσωπικούς υπολογιστές. Ανάλογη είναι και η ανάπτυξη του λογισμικού με το οποίο λειτουργούν οι φορητές συσκευές σήμερα. Το πιο ευρέως διαδεδομένο λειτουργικό σύστημα για τις φορητές συσκευές καθώς και για την ανάπτυξη εφαρμογών σε αυτές είναι το Android.

Ένας από τους σημαντικότερους λόγους της ραγδαίας εξάπλωσης του Android είναι το μεγάλο πλήθος εφαρμογών που προσφέρει. Μία φορητή εφαρμογή (mobile app) είναι μια εφαρμογή λογισμικού σχεδιασμένη να τρέχει σε κάποια φορητή συσκευή όπως smartphones ή tablets. Οι φορητές εφαρμογές στο ξεκίνημα τους, στόχευαν στην εξυπηρέτηση βασικών λειτουργιών των κινητών τηλεφώνων και αναγκών του κοινού, καθώς και στην ανάκτηση πληροφοριών, συμπεριλαμβανομένων εφαρμογών για email, κατάλογο για διαχείριση των επαφών, ημερολόγιο, πληροφορίες για τον καιρό κτλ. Ωστόσο στην συνέχεια, οι αυξανόμενες απαιτήσεις και ανάγκες των χρηστών καθώς και η διαθεσιμότητα των εργαλείων ανάπτυξης εφαρμογών Android οδήγησε με γρήγορους ρυθμούς την εξέλιξη των φορητών εφαρμογών και σε άλλες κατηγορίες.

Πλέον υπάρχουν εφαρμογές σχεδόν για κάθε λειτουργία και χρήση που μπορεί να σκεφτεί κάποιος στα πλαίσια των φορητών συσκευών. Από επιστημονικές και ιατρικές εφαρμογές, μέχρι εφαρμογές για αθλητικά και ψυχαγωγία, προσφέρεται στους χρήστες μια γιγαντιαία ποικιλία εφαρμογών για να διαλέξουν ανάλογα με τις ανάγκες τους. Το Google Play Store, η επίσημη πλατφόρμα για λήψη και εγκατάσταση Android εφαρμογών εκτιμάται ότι έχει πάνω 3 εκατομμύρια διαθέσιμες εφαρμογές από την κυκλοφορία του στα τέλη του 2008 μέχρι και σήμερα. Οι κυριότερες και δημοφιλέστερες κατηγορίες φορητών εφαρμογών σήμερα, σχετίζονται με παιχνίδια και ψυχαγωγία, υπηρεσίες τοποθεσίας (location-based) και GPS, e-banking και χρηματιστηριακές υπηρεσίες, αγορές εισιτηρίων, κρατήσεις καταλυμάτων, κοινωνικά δίκτυα και υπηρεσίες ανταλλαγής μηνυμάτων, e-shops.

Μια σημαντική κατηγορία εφαρμογών σχετίζεται με την επεξεργασία και οργάνωση φωτογραφιών και βίντεο. Οι φορητές συσκευές και ιδιαίτερα τα smartphones έχουν κάνει τη δυνατότητα λήψης φωτογραφιών και βίντεο, απλούστερη και ταχύτερη όσο ποτέ, με αποτέλεσμα οι χρήστες να αποθηκεύουν εκατοντάδες, ακόμα και χιλιάδες φωτογραφίες στις συσκευές τους. Αυτό έχει συντελέσει στην ανάπτυξη πλήθους εφαρμογών και προγραμμάτων για διαχείριση και οργάνωση φωτογραφιών, καθώς και εφαρμογών που προσφέρουν δυνατότητες επεξεργασίας φωτογραφιών και βίντεο μέσω φίλτρων και άλλων εργαλείων, δυνατότητες για την αποθήκευσή τους σε cloud υπηρεσίες και επιλογές για κοινοποίηση σε κοινωνικά δίκτυα και άλλα μέσα.

Οι περισσότερες φορητές συσκευές διακρίνονται από κάμερα υψηλής ανάλυσης και διαθέτουν αρκετές δεκάδες ή και εκατοντάδες gigabytes αποθηκευτικού χώρου. Η απλή και γρήγορη διαδικασία λήψης φωτογραφιών με την χρήση μιας φορητής συσκευής, κυρίως των smartphones, έχει ως αποτέλεσμα, οι χρήστες συχνά να καταλήγουν με μία πληθώρα αποθηκευμένων φωτογραφιών στην συσκευή τους. Σε αυτό έχει συντελέσει και η δημοτικότητα των κοινωνικών δικτύων όπως Instagram και Facebook, των οποίων η βασική χρήση σχετίζεται με τη λήψη και το ανέβασμα φωτογραφιών.

Όσο ευχάριστη και διασκεδαστική και αν είναι η λήψη φωτογραφιών, η οργάνωση, η διαχείριση, και η αναζήτηση τους μπορεί να αποτελέσει μία περίπλοκη και κουραστική διαδικασία. Ιδιαίτερα όταν υπάρχει μεγάλος αριθμός φωτογραφιών και βίντεο στον αποθηκευτικό χώρο της συσκευής η ταξινόμηση και η οργάνωση τους μπορεί να διαρκέσει ακόμα και ώρες. Οπότε είναι σημαντικό, οι φωτογραφίες να αποθηκεύονται και να διατηρούνται οργανωμένες ανεξαρτήτως του πλήθους τους.

Για την αντιμετώπιση της συσσώρευσης μεγάλου όγκου φωτογραφιών και βίντεο στις φορητές συσκευές, έχουν αναπτυχθεί πολλές εφαρμογές και υπηρεσίες, οι οποίες προσφέρουν ευκολότερη διαχείριση και οργάνωση των αρχείων, ώστε να μπορούν οι χρήστες να αναζητούν και να βρίσκουν ευκολότερα τις φωτογραφίες και τα βίντεο τους. Μία εφαρμογή οργάνωσης αρχείων φωτογραφιών και βίντεο βοηθά στην αποθήκευση όλων των ψηφιακών αρχείων και συλλογών σε ένα μέρος, προσφέροντας παράλληλα ένα πλήθος χρήσιμων χαρακτηριστικών όπως την προσθήκη ετικετών στις φωτογραφίες και τα βίντεο, την αξιολόγησή τους και τη δημιουργία αντιγράφων ασφαλείας σε κάποια cloud υπηρεσία.

Το βασικότερο ζητούμενο και πρόβλημα που έχουν να αντιμετωπίσουν οι συγκεκριμένες εφαρμογές είναι να μειώσουν όσο το δυνατόν περισσότερο το χρόνο οργάνωσης και διαχείρισης των αρχείων και παράλληλα να είναι απλές και κατανοητές για ένα μέσο χρήστη. Για να επιτευχθεί αυτό, σε πολλές εφαρμογές οργάνωσης φωτογραφιών και βίντεο χρησιμοποιούνται ανεπτυγμένες τεχνολογίες που παρέχουν έξυπνη αναγνώριση εικόνων αυτοματοποιώντας τη διαδικασία προσθήκης ετικετών και την διαγραφή διπλότυπων. Επίσης, υπάρχει η δυνατότητα δημιουργίας συλλογών και άλμπουμ για αποτελεσματικότερη ταξινόμηση. Επιπλέον, ορισμένες εφαρμογές οργάνωσης φωτογραφιών συμπεριλαμβάνουν μεταξύ άλλων και δυνατότητες επεξεργασίας εικόνων, εκτύπωσης και κοινοποίησης σε διάφορα μέσα.

2. ANDROID

Το android είναι ένα λειτουργικό σύστημα το οποίο δημιουργήθηκε από τους Andy Rubin, Rich Miner, Nick Sears και Chris White το 2003 και εξαγοράστηκε από την Google το 2005 στην οποία ανήκει και υποστηρίζεται μέχρι και σήμερα. Ο πυρήνας του είναι βασισμένος στο Linux και οι συσκευές που το υποστηρίζουν βασίζονται στην αρχιτεκτονική ARM. Ο πηγαίος του κώδικας έχει χρησιμοποιηθεί σε πλήθος ηλεκτρονικών συσκευών όπως κινητά τηλέφωνα, tablets, κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, φορητές συσκευές αναπαραγωγής πολυμέσων, υπολογιστές, τηλεοράσεις και άλλα. Εκτιμάται ότι σήμερα υπάρχουν περισσότερες από ένα δισεκατομμύριο συσκευές οι οποίες τρέχουν με λειτουργικό σύστημα Android στο λογισμικό τους, καθιστώντας το ως τη μεγαλύτερη βάση εγκαταστημένου λειτουργικού συστήματος παγκοσμίως. Μερικές από τις δημοφιλέστερες εταιρίες που χρησιμοποιούν το λειτουργικό Android για τις φορητές συσκευές τους είναι η Samsung, Xiaomi, Huawei, LG, OnePlus και Sony.

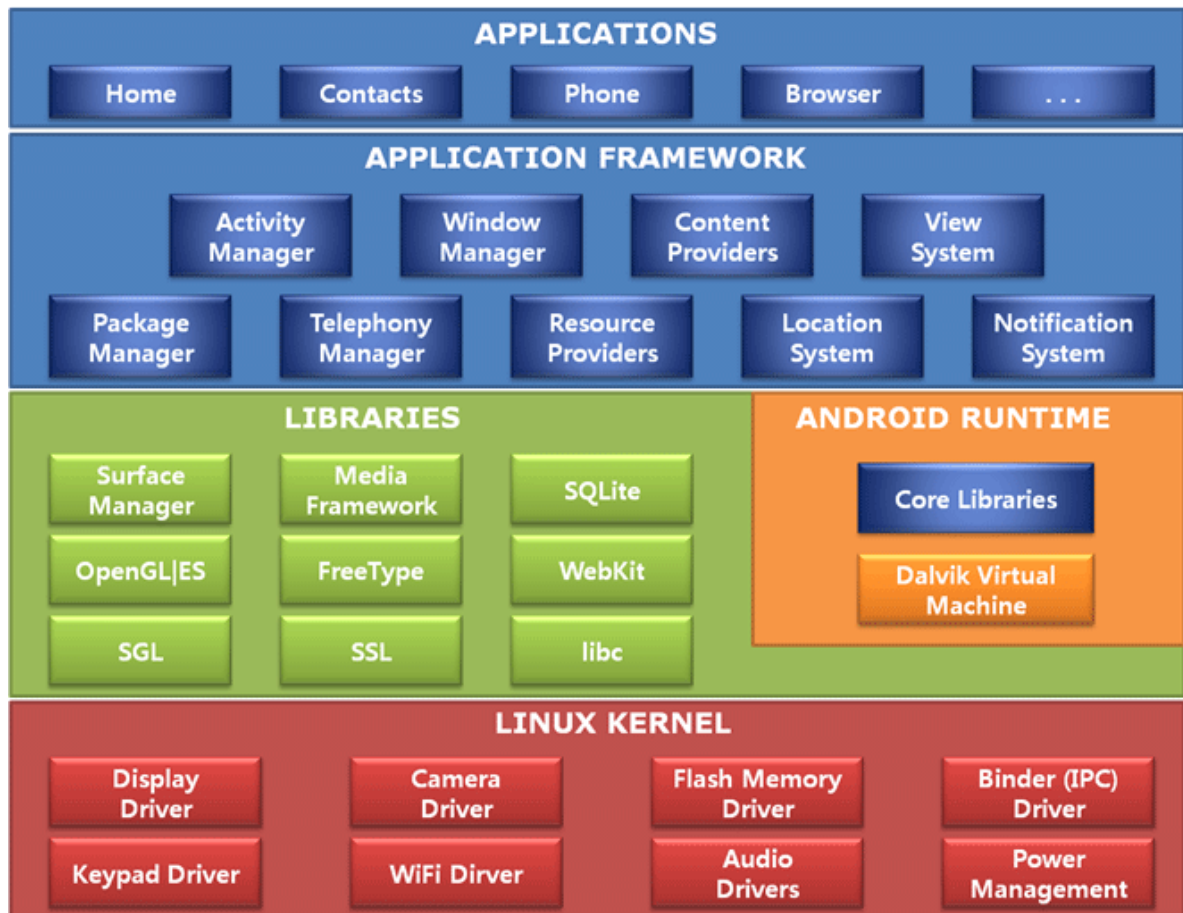
2.1 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ

- **Interface (Διεπαφή):** Η βασική διεπαφή του Android βασίζεται κυρίως στην άμεση αλληλεπίδραση του χρήστη με μια οθόνη αφής η οποία ανταποκρίνεται σε συμβάντα αφής για τον χειρισμό αντικειμένων της οθόνης και με ένα εικονικό πληκτρολόγιο. Η απόκριση στην είσοδο του χρήστη έχει σχεδιαστεί για να είναι όσο το δυνατόν περισσότερο άμεση, χρησιμοποιώντας συχνά τις δυνατότητες δόνησης της συσκευής για να παρέχει απτική ανάδραση στον χρήστη. Εσωτερικό υλικό των συσκευών όπως επιταχυνσιόμετρα, γυροσκόπια και άλλοι αισθητήρες αξιοποιούνται από το λογισμικό του Android για να ανταποκρίνονται σε πραγματικές ενέργειες των χρηστών όπως η προσαρμογή της οθόνης από κατακόρυφο σε οριζόντιο προσανατολισμό.
- **Μηνύματα:** Τα SMS και τα MMS αποτελούν τις διαθέσιμες μορφές μηνυμάτων. Επίσης τηλέφωνα με λειτουργικό σύστημα Android έχουν τη δυνατότητα αποστολής και λήψης RCS μέσω της εφαρμογής των μηνυμάτων, εάν υποστηρίζεται από τον πάροχο κινητής τηλεφωνίας.
- **Φωνητικές λειτουργίες:** Από την αρχική κυκλοφορία του Android υποστηρίζεται φωνητική αναζήτηση μέσω της μηχανής αναζήτησης της Google. Οι φωνητικές ενέργειες για κλήσεις, μηνύματα, πλοήγηση κ.λπ. υποστηρίζονται από την έκδοση 2.2 και μετά. Από την έκδοση του Android 4.1, η Google έχει επεκτείνει τις φωνητικές λειτουργίες με τη δυνατότητα συνομιλίας και ανάγνωσης απαντήσεων από το γράφημα γνώσεων της Google (Google's Knowledge Graph).
- **Multitasking:** Το Android παρέχει τη δυνατότητα εκτέλεσης πολλών διαφορετικών εφαρμογών ταυτόχρονα, χάρις στον ειδικό χειρισμό κατανομής μνήμης που υποστηρίζεται.
- **Προσβασιμότητα:** Για άτομα με προβλήματα όρασης παρέχεται η λειτουργία *text-to-speech* η οποία μετατρέπει το γραπτό κείμενο σε προφορικό. Επιπλέον υποστηρίζονται λειτουργίες για άτομα με προβλήματα ακοής.
- **Συνδεσιμότητα:** Το Android παρέχει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένων GSM/EDGE, Bluetooth, LTE, CDMA, 3G, 4G, UMTS, NFC, IDEN και Wi-Fi. Επίσης οι συσκευές Android μπορούν να χρησιμοποιηθούν και ως ασύρματο ή ενσύρματο Wi-Fi hotspot.
- **Πολυμέσα:** Το λειτουργικό Android υποστηρίζει τις ακόλουθες μορφές ήχου, στατικής και κινούμενης εικόνας: H.263, H.264 (σε 3GP ή MP4 container), MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, WAV, JPEG, PNG, GIF, BMP.

- **Αποθήκευση δεδομένων:** Η SQLite, μια σχεσιακή βάση δεδομένων, χρησιμοποιείται για την αποθήκευση δεδομένων στο λειτουργικό σύστημα Android.

2.2 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ANDROID

Το λειτουργικό σύστημα του Android είναι χωρισμένο σε πέντε τμήματα και τέσσερα βασικά επίπεδα τα οποία συγκροτούν την αρχιτεκτονική του.



Εικόνα 1: Η αρχιτεκτονική του Android [Source: <https://www.dev2qa.com/android-architecture-components-introduction/>]

- **Linux Kernel:** Πρόκειται για τον πυρήνα του λογισμικού και βρίσκεται στο χαμηλότερο επίπεδο. Ο πυρήνας είναι υπεύθυνος για την επικοινωνία του υλικού της συσκευής με το λογισμικό καθώς παρέχει όλους τους απαραίτητους drivers για την λειτουργία του συστήματος, όπως της οθόνης, της κάμερας, των ηχείων και άλλων βασικών εξαρτημάτων της συσκευής. Υποστηρίζει λειτουργίες δικτύωσης και διαχειρίζεται βασικούς πόρους που αφορούν τη μνήμη και την ισχύ της συσκευής.
- **Libraries:** Στο επόμενο επίπεδο βρίσκεται ένα σύνολο C/C++ και Java βιβλιοθηκών που παρέχουν υποστήριξη για ανάπτυξη λογισμικού σε Android και είναι διαθέσιμες για τους προγραμματιστές. Μερικές από τις βασικότερες βιβλιοθήκες του συνόλου είναι:
 - **Media Library:** Βιβλιοθήκη πολυμέσων για αναπαραγωγή και εγγραφή διάφορων μορφών ήχου και βίντεο.
 - **Surface Manager:** Βιβλιοθήκη για τη διαχείριση στοιχείων και γραφικών της οθόνης.
 - **SQLite:** Ελαφριά βάση δεδομένων για την αποθήκευση δεδομένων από εφαρμογές.

- **WebKit:** Βιβλιοθήκη για την υποστήριξη περιηγητή Web (web browser).
 - **SSL:** Βιβλιοθήκες υπεύθυνες για την ασφάλεια στο διαδίκτυο.
- **Android Runtime:** Το Android Runtime αποτελεί το τρίτο τμήμα της αρχιτεκτονικής του Android και βρίσκεται στο ίδιο επίπεδο με τις βιβλιοθήκες. Περιέχει και αυτό ένα σύνολο από βασικές βιβλιοθήκες και μια εικονική μηχανή (Dalvik Virtual Machine). Η εικονική μηχανή Dalvik επιτρέπει σε κάθε εφαρμογή να τρέξει την δικιά της διεργασία μαζί με τη δικιά της ξεχωριστή Dalvik virtual machine. Μετατρέπει τον κώδικα της εφαρμογής σε εντολές μηχανής οι οποίες εκτελούνται από την εικονική μηχανή στη συσκευή του χρήστη. Είναι ειδικά σχεδιασμένη ώστε να εξασφαλίσει ότι κάθε συσκευή μπορεί να εκτελεί πολλαπλά *instances* της εικονικής μηχανής ταυτόχρονα. Τέλος, οι βιβλιοθήκες του Android Runtime παρέχουν υποστήριξη για ανάπτυξη Android εφαρμογών με τη χρήση της γλώσσας προγραμματισμού Java.
 - **Application Framework:** Το τρίτο επίπεδο της αρχιτεκτονικής του Android είναι το Application Framework. Παρέχει υπηρεσίες υψηλότερου επιπέδου με τη μορφή κλάσεων οι οποίες μπορούν να χρησιμοποιηθούν από τους προγραμματιστές κατά την ανάπτυξη των εφαρμογών τους. Παρέχει επίσης ένα επίπεδο αφαίρεσης στην αλληλεπίδραση μεταξύ υλικού και εφαρμογών και διαχειρίζεται τους πόρους αυτών των εφαρμογών. Το Application Framework περιλαμβάνει τις ακόλουθες βασικές υπηρεσίες:
 - **Activity Manager** – Διαχειρίζεται τον κύκλο ζωής (life cycle) των εφαρμογών και των δραστηριοτήτων τους.
 - **Content Providers** – Επιτρέπει στις εφαρμογές να έχουν πρόσβαση σε δεδομένα άλλων εφαρμογών αλλά και να διαμοιράζονται τα δικά τους.
 - **Resource Manager** – Παρέχει πρόσβαση σε πόρους όπως συμβολοσειρές, ρυθμίσεις χρωμάτων και γραφικών στοιχείων που αφορούν τη διάταξη της διεπαφής.
 - **Notifications Manager** – Επιτρέπει στις εφαρμογές την εμφάνιση ειδοποιήσεων στον χρήστη.
 - **View System** – Παρέχει μια συλλογή από Views (λίστες, κουμπιά, πλαίσια εικόνας και κειμένου, πίνακες κ.α.) τα οποία μπορούν να χρησιμοποιηθούν για τη δημιουργία των διατάξεων της διεπαφής μιας εφαρμογής.
 - **Telephony Manager, Location Manager** – Υπηρεσίες τηλεφωνίας και τοποθεσίας.

Applications: Οι εφαρμογές βρίσκονται στο τελευταίο επίπεδο της αρχιτεκτονικής του Android. Περιλαμβάνονται τόσο οι προεγκατεστημένες εφαρμογές της συσκευής όπως αυτή των επαφών, της κάμερας, των μηνυμάτων και του ημερολογίου, όσο και οι εφαρμογές που κάνει εγκατάσταση ο χρήστης σύμφωνα με τις ανάγκες του. Το συγκεκριμένο επίπεδο τρέχει τις εφαρμογές μέσω του Android Runtime επιπέδου καθώς και με τη βοήθεια των κλάσεων και των υπηρεσιών του Application Framework επιπέδου.

2.3 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΣΕ ANDROID

Το Android SDK (Software Development Kit) αποτελεί ένα σύνολο βασικών εργαλείων για ανάπτυξη Android εφαρμογών. Περιλαμβάνει τις βιβλιοθήκες Java του Android που είναι απαραίτητες για την δημιουργία μια εφαρμογής, ένα προσομοιωτή (emulator) φορητών συσκευών για δοκιμαστική εκτέλεση της εφαρμογής, έναν ανιχνευτή σφαλμάτων (debugger) και έναν compiler για την παραγωγή του κώδικα σε γλώσσα μηχανής. Επιπλέον προσφέρει υποστήριξη μέσω δειγμάτων κώδικα, μαθημάτων (tutorials) και την απαραίτητη τεκμηρίωση (documentation) όλων των εργαλείων και υπηρεσιών που παρέχει.

Επιχειρήσεις μικρού και μεγάλου βεληνεκού, επιλέγουν να αναπτύξουν δικές τους Android εφαρμογές για να προσφέρουν τις υπηρεσίες τους με μικρό οικονομικό και κυρίως χρονικό κόστος. Τα εργαλεία και οι βιβλιοθήκες που παρέχονται στους προγραμματιστές μέσω του Android SDK απλοποιούν σημαντικά τη διαδικασία ανάπτυξης εφαρμογών και προσφέρουν τα εξής οφέλη και πλεονεκτήματα.

➤ **Ευελιξία και λειτουργικότητα**

Το Android επιτρέπει την ανάπτυξη εφαρμογών λογισμικού οι οποίες έχουν τη δυνατότητα να εκμεταλλεύονται πλήρως τις βασικές λειτουργίες και δυνατότητες της συσκευής, όπως η αποστολή μηνυμάτων, η χρήση της κάμερας, υπηρεσίες τοποθεσίας κ.α. Αυτό διευκολύνει τη δημιουργία περίπλοκων και λειτουργικά πλουσιότερων εφαρμογών. Επιπλέον, το Android υποστηρίζει την ταυτόχρονη εκτέλεση πολλαπλών εφαρμογών χωρίς να επηρεάζεται η απόδοση τους καθώς βασίζεται στο multitasking. Επίσης αποτελεί μια πλατφόρμα ανοιχτού κώδικα με αποτέλεσμα να μπορεί να επεκταθεί και να τροποποιηθεί ώστε να συμβαδίζει με τις νέες τεχνολογίες. Τέλος, η ανοιχτού κώδικα φιλοσοφία του Android συνδράμει στην αλληλεπίδραση των μελών της προγραμματιστικής κοινότητας για τις ανερχόμενες εξελίξεις και στην υποστήριξη και επίλυση προβλημάτων πάνω στην ανάπτυξη λογισμικού.

➤ **Ταχύτερη υλοποίηση και χαμηλό κόστος ανάπτυξης**

Η μεγάλη ποικιλία εργαλείων και κατάλληλων βιβλιοθηκών που παρέχει το Android για την ανάπτυξη λογισμικού, βοηθά τους προγραμματιστές να υλοποιήσουν ταχύτερα και ευκολότερα εφαρμογές. Συγκριτικά με άλλες πλατφόρμες το Android έχει χαμηλότερο κόστος, άρα και χαμηλότερο ρίσκο στην ανάπτυξη λογισμικού. Το Android SDK παρέχει στην κοινότητα των προγραμματιστών τα απαραίτητα εργαλεία και υπηρεσίες για ανάπτυξη εφαρμογών δωρεάν, ελαχιστοποιώντας με αυτό τον τρόπο τα κόστη ανάπτυξης.

➤ **Μεγαλύτερη αγορά**

Η ανάπτυξη κάθε εφαρμογής είτε είναι γενικού σκοπού, είτε επιστημονική, είτε gaming πάντα γίνεται με σκοπό να μπορέσει να καταλάβει όσο το δυνατόν μεγαλύτερο μερίδιο στην αγορά και να επικρατήσει στον ανταγωνισμό. Για να επιτευχθεί αυτό πρέπει να είναι συμβατή με τη πλατφόρμα που έχει τον μεγαλύτερο αριθμό χρηστών σε φορητές συσκευές την τρέχουσα στιγμή. Η παγκόσμια κυριαρχία στην αγορά είναι ένα από τα μεγαλύτερα πλεονεκτήματα του Android καθώς το λειτουργικό του σύστημα φιλοξενείται από έναν μεγάλο αριθμό κατασκευαστών φορητών συσκευών, οι οποίες τείνουν να είναι περισσότερο οικονομικά προσιτές από τον ανταγωνισμό. Αυτό έχει ως αποτέλεσμα το Android να διατηρεί το 70-80% του μεριδίου της παγκόσμιας αγοράς τα τελευταία χρόνια. Επομένως μια Android εφαρμογή σήμερα, απευθύνεται αυτομάτως σε έναν δραματικά μεγαλύτερο αριθμό χρηστών συγκριτικά με οποιαδήποτε άλλη πλατφόρμα.

➤ **Ασφάλεια**

Είναι πολύ σημαντικό για μια εφαρμογή φορητής συσκευής να είναι επαρκώς προστατευμένη, ώστε να μην υπάρχει καμία πιθανότητα παραβίασης στις πληροφορίες και στα προσωπικά δεδομένα που πιθανών να αποθηκεύει. Το λογισμικό του Android είναι βασισμένο στο Linux, το οποίο αποτελεί εγγύηση για την ασφάλεια και την απρόσκοπτη λειτουργία των εφαρμογών του. Καθόλη τη διάρκεια της ιστορίας του Android δεν έχει υπάρξει καταγεγραμμένο περιστατικό σχετικά με διαρροή πληροφοριών ή παραβίαση της ασφάλειας κάποιας εφαρμογής.

3. ΑΝΑΠΤΥΞΗ ANDROID ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΟΡΓΑΝΩΣΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΦΩΤΟΓΡΑΦΙΩΝ ΚΑΙ ΒΙΝΤΕΟ

Η συντριπτική πλειοψηφία κατόχων μιας φορητής συσκευής όπως smartphone, τη χρησιμοποιούν ως το βασικό τους μέσο για τη λήψη και αποθήκευση φωτογραφιών. Η συσσώρευση μεγάλου όγκου ψηφιακών αρχείων, ιδιαίτερα φωτογραφιών και βίντεο στις φορητές συσκευές είναι ένα συχνό φαινόμενο σήμερα. Αυτό έχει ως αποτέλεσμα η διαχείριση και η αναζήτηση τους να αποτελεί μια περίπλοκη και συχνά χρονοβόρα διαδικασία. Είναι σημαντικό κάθε συλλογή ψηφιακών φωτογραφιών και βίντεο να διατηρείται ταξινομημένη και οργανωμένη με τέτοιο τρόπο ώστε η αναζήτηση τους να γίνεται εύκολα και γρήγορα.

Σκοπός της παρούσας εργασίας είναι η σχεδίαση και ανάπτυξη μιας εφαρμογής για Android φορητές συσκευές η οποία θα παρέχει δυνατότητες οργάνωσης, ταξινόμησης και αναζήτησης των φωτογραφιών και βίντεο που βρίσκονται στον αποθηκευτικό χώρο της συσκευής, με τη χρήση ορισμένων λειτουργιών κατηγοριοποίησης και φιλτραρίσματος των αρχείων.

3.1 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ

3.1.1 Περιπτώσεις χρήσεις εφαρμογής

- ΠΧ1. Προβολή λίστας φωτογραφιών:** Ο χρήστης επιλέγει από το αρχικό μενού την επιλογή "Photos". Στην οθόνη εμφανίζεται μια λίστα σε μορφή πλέγματος η οποία περιέχει όλες τις φωτογραφίες που είναι αποθηκευμένες στην συσκευή του χρήστη. Οι φωτογραφίες εμφανίζονται ταξινομημένες σε φθίνουσα χρονολογική σειρά, με τις πιο πρόσφατες να βρίσκονται στην κορυφή της λίστας.
- ΠΧ2. Προβολή λίστας βίντεο:** Ο χρήστης επιλέγει από το αρχικό μενού την επιλογή "Videos". Στην οθόνη εμφανίζεται μια λίστα σε μορφή πλέγματος η οποία περιέχει όλα τα βίντεο που είναι αποθηκευμένα στην συσκευή του χρήστη. Τα βίντεο εμφανίζονται ταξινομημένα σε φθίνουσα χρονολογική σειρά, με τα πιο πρόσφατα να βρίσκονται στην κορυφή της λίστας.
- ΠΧ3. Προβολή άλμπουμ:** Ο χρήστης επιλέγει από το αρχικό μενού την επιλογή "Albums". Στην οθόνη εμφανίζονται όλοι οι φάκελοι/άλμπουμ της συσκευής που περιέχουν κάποιο αρχείο εικόνας ή βίντεο. Ο χρήστης επιλέγει ένα από τα διαθέσιμα άλμπουμ της συλλογής του. Στη συνέχεια το σύστημα εμφανίζει στην οθόνη του χρήστη, μια λίστα σε μορφή πλέγματος με όλα τα αρχεία (φωτογραφίες και βίντεο) του επιλεγμένου άλμπουμ.
- ΠΧ4. Προβολή φωτογραφίας/βίντεο:** Ο χρήστης πλοηγείται στη λίστα φωτογραφιών ή βίντεο ή σε κάποιο άλμπουμ της συσκευής του ακολουθώντας μια από τις προαναφερθέντες περιπτώσεις χρήσης. Στη συνέχεια επιλέγει ένα από τα διαθέσιμα αρχεία που προβάλλονται. Εάν η επιλογή του χρήστη ήταν αρχείο φωτογραφίας, τότε προβάλλεται σε πλήρες μέγεθος ανάλογο με τις διαστάσεις της οθόνης. Εάν η επιλογή του χρήστη ήταν αρχείο βίντεο, τότε ο χρήστης έχει τη δυνατότητα να πατήσει το πλήκτρο της αναπαραγωγής για την προβολή του βίντεο σε μια νέα διάταξη οθόνης.
- ΠΧ5. Εισαγωγή/διαγραφή ετικέτας σε φωτογραφία ή βίντεο:** Ο χρήστης επιλέγει προβολή μιας φωτογραφίας ή βίντεο ακολουθώντας την σχετική περίπτωση χρήσης. Στη συνέχεια επιλέγει "Tags" στη μπάρα πλοήγησης που βρίσκεται στο κάτω μέρος της οθόνης. Στο παράθυρο που εμφανίζεται ο χρήστης έχει της εξής δυνατότητες:

- Ο χρήστης επιλέγει μια ή περισσότερες από τις προκαθορισμένες ετικέτες της εφαρμογής.
- Ο χρήστης εισάγει μία ή περισσότερες ετικέτες της επιλογής του από το πληκτρολόγιο.
- Ο χρήστης αφαιρεί μία από τις υπάρχουσες ετικέτες του συγκεκριμένου αρχείου.
- Ο χρήστης εκτελεί οποιονδήποτε συνδυασμό των παραπάνω.

Τέλος επιλέγει "Save Tags" για την αποθήκευση των αλλαγών.

ΠΧ6. Εισαγωγή/Επεξεργασία ημερομηνίας σε φωτογραφία ή βίντεο: Ο χρήστης επιλέγει προβολή μιας φωτογραφίας ή βίντεο ακολουθώντας την σχετική περίπτωση χρήσης. Στη συνέχεια επιλέγει "Date" στη μπάρα πλοήγησης που βρίσκεται στο κάτω μέρος της οθόνης. Στην οθόνη εμφανίζεται ένα παράθυρο διαλόγου το οποίο περιέχει ένα ημερολόγιο στο οποίο ο χρήστης μπορεί να ορίσει μια νέα ημερομηνία ή να τροποποιήσει την ήδη υπάρχουσα εάν έχει ήδη οριστεί για το συγκεκριμένο αρχείο.

ΠΧ7. Εισαγωγή/Επεξεργασία/Διαγραφή τοποθεσίας σε φωτογραφία ή βίντεο: Ο χρήστης επιλέγει προβολή μιας φωτογραφίας ή βίντεο ακολουθώντας την σχετική περίπτωση χρήσης. Στη συνέχεια επιλέγει "Location" στη μπάρα πλοήγησης που βρίσκεται στο κάτω μέρος της οθόνης. Στο παράθυρο που εμφανίζεται ο χρήστης ξεκινά να πληκτρολογεί την τοποθεσία της επιλογής του. Εμφανίζεται στην οθόνη μία λίστα από προτάσεις τοποθεσιών, η οποία ανανεώνεται αυτόματα σύμφωνα με την είσοδο του χρήστη από το πληκτρολόγιο. Ο χρήστης επιλέγει μία από τις διαθέσιμες προτάσεις της λίστας η οποία αποθηκεύεται αυτομάτως ως τοποθεσία για το συγκεκριμένο αρχείο. Από το ίδιο παράθυρο διαλόγου ο χρήστης έχει τη δυνατότητα να αλλάξει ή να διαγράψει την τρέχουσα τοποθεσία του αρχείου εάν υπάρχει.

ΠΧ8. Μαζική εισαγωγή ετικετών/ημερομηνίας/τοποθεσίας σε πολλαπλά αρχεία: Ο χρήστης πλοηγείται στη λίστα φωτογραφιών ή βίντεο ή σε κάποιο άλμπουμ της συσκευής του ακολουθώντας μια από τις σχετικές περιπτώσεις χρήσης. Στη συνέχεια κάνει κλικ παρατεταμένα σε κάποιο αρχείο της λίστας. Η λίστα μπαίνει σε "selection mode", και ο χρήστης επιλέγει πολλαπλά αρχεία για μαζική επεξεργασία. Ο χρήστης επιλέγει από το μενού που είναι διαθέσιμο στο πάνω μέρος της οθόνης, την εισαγωγή ετικετών/ημερομηνίας/τοποθεσίας για τα αρχεία που επέλεξε. Εκτελεί τις αντίστοιχες ενέργειες σύμφωνα με την επιλογή του και οι αλλαγές αποθηκεύονται στα επιλεγμένα αρχεία.

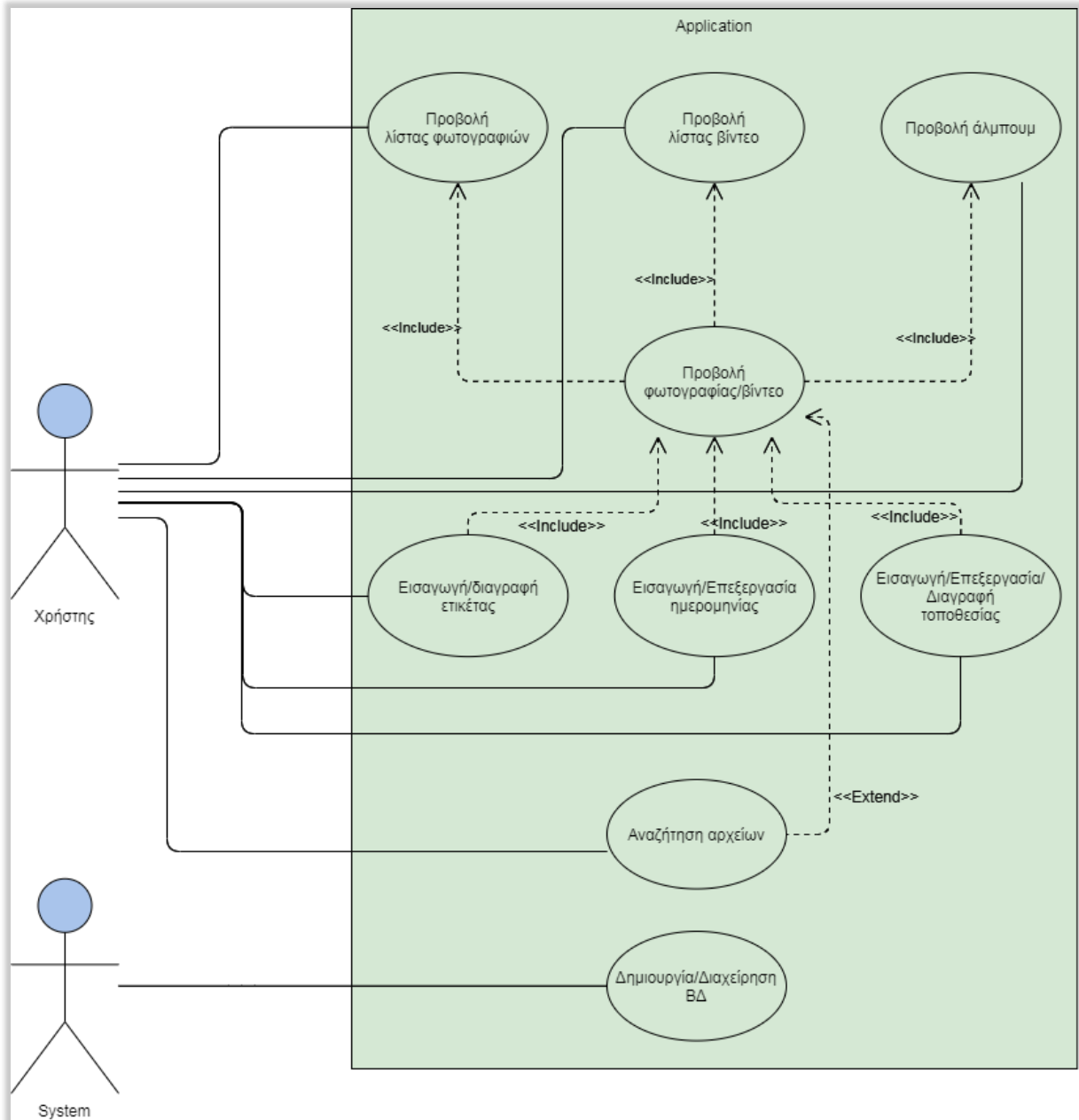
ΠΧ9. Αναζήτηση αρχείων: Ο χρήστης επιλέγει από το αρχικό μενού την επιλογή "Search". Στην οθόνη που εμφανίζεται ο χρήστης μπορεί να κάνει αναζήτηση φωτογραφιών και βίντεο σύμφωνα με τα ακόλουθα κριτήρια:

- Ο χρήστης εισάγει στην μπάρα αναζήτησης ένα τμήμα λέξεων από τις ετικέτες ή τις τοποθεσίες των αρχείων που θέλει να αναζητήσει.
- Ο χρήστης πατώντας στην επιλογή "Date Range" ορίζει ένα εύρος ημερομηνιών. Τα αρχεία που επιστρέφονται ως αποτελέσματα βρίσκονται εντός του εύρους ημερομηνίας που όρισε ο χρήστης.
- Ο χρήστης αναζητά αρχεία με συνδυασμό των παραπάνω κριτηρίων.

Εμφανίζονται στην οθόνη τα αποτελέσματα της αναζήτησης που ικανοποιούν τα κριτήρια της αναζήτησης. Ο χρήστης μπορεί να επιλέξει οποιαδήποτε φωτογραφία ή βίντεο από τα αποτελέσματα για προβολή ή για επεξεργασία των μεταδεδομένων τους.

ΠΧ10. Δημιουργία/Διαχείριση Βάσης Δεδομένων: Σε κάθε εκκίνηση της εφαρμογής, γίνεται έλεγχος εάν υπάρχει η βάση δεδομένων της εφαρμογής και αν δεν υπάρχει θα

δημιουργείται από την εφαρμογή. Κάθε φορά που ο χρήστης βρίσκεται στην αρχική οθόνη της εφαρμογής γίνεται έλεγχος των φωτογραφιών και των βίντεο της συσκευής. Εάν κάποιο αρχείο έχει διαγραφεί από τον αποθηκευτικό χώρο της συσκευής, τότε διαγράφεται και από τη βάση δεδομένων της εφαρμογής. Η συγκεκριμένη περίπτωση χρήσης δεν είναι ορατή από τους χρήστες, αλλά ενεργοποιείται κατά την εκκίνηση της εφαρμογής από το σύστημα.



Εικόνα 2: Use Case Diagram

3.1.2 Σχεδιαστικές και τεχνικές λεπτομέρειες

Κατά την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν ορισμένες βιβλιοθήκες, τεχνολογίες και APIs τα οποία διευκόλυναν την υλοποίηση βασικών λειτουργιών της εφαρμογής. Η χρήση βιβλιοθηκών και APIs εξασφαλίζει ένα επίπεδο αφαίρεσης μεταξύ του προγραμματιστή και των υπηρεσιών που παρέχουν. Με άλλα λόγια, ο προγραμματιστής μπορεί να χρησιμοποιήσει τις υπηρεσίες μια βιβλιοθήκης γνωρίζοντας μόνο ορισμένες έτοιμες βασικές συναρτήσεις που περιέχει, και αγνοώντας τον τρόπο με τον οποίο έχουν υλοποιηθεί και την εσωτερική λειτουργικότητα τους. Αυτό συντελεί στην εξοικονόμηση χρόνου και στην ταχύτερη ανάπτυξη της εφαρμογής. Η εισαγωγή βιβλιοθηκών και εξωτερικών πακέτων γίνεται συνήθως μέσω του αρχείου **build.gradle** στο μπλοκ των εξαρτήσεων. Στο συγκεκριμένο μπλοκ (dependencies) γίνεται η δήλωση των βιβλιοθηκών που θέλουμε να συμπεριληφθούν στο project. Στη συνέχεια, παρουσιάζονται συνοπτικά ορισμένα γενικά στοιχεία των βιβλιοθηκών που χρησιμοποιήθηκαν καθώς και ο τρόπος με τον οποίο χρησιμοποιήθηκαν στην εφαρμογή.

- **RecyclerView**

Η βιβλιοθήκη RecyclerView παρέχει ένα αποτελεσματικό μηχανισμό για την απεικόνιση ενός συνόλου δεδομένων σε μορφή λίστας. Αποτελεί ουσιαστικά την εξέλιξη του στοιχείου γραφικής διασύνδεσης ListView, το οποίο πλέον θεωρείται απαρχαιωμένο και προτείνεται από το Android αντί αυτού η χρήση του RecyclerView. Ο προγραμματιστής έχει τη δυνατότητα να ορίσει τον τύπο των στοιχείων και των δεδομένων που θα εισάγει στο RecyclerView καθώς και την μορφή με την οποία θα εμφανιστούν στην οθόνη.

Όπως υποδηλώνει και το όνομά του, το RecyclerView έχει τη δυνατότητα να "ανακυκλώνει" τα στοιχεία που περιέχει. Καθώς πρόκειται για μία δυναμική λίστα, όταν ένα αντικείμενο της δεν είναι ορατό στην οθόνη, τότε η όψη του δεν καταστρέφεται από το σύστημα, αλλά επαναχρησιμοποιείται για τα νέα αντικείμενα που έχουν μετακινηθεί στη θέση του.

Για παράδειγμα, εάν μια λίστα περιέχει εκατό αντικείμενα θα ήταν μεγάλη σπατάλη σε πόρους να δημιουργηθεί μία διαφορετική όψη και θέση για το κάθε ένα στην οθόνη, από τη στιγμή που δεν πρόκειται να είναι ορατά και τα εκατό αντικείμενα ταυτόχρονα. Μια αποδοτικότερη προσέγγιση θα ήταν να δημιουργηθούν όσες όψεις είναι ορατές κάθε χρονική στιγμή στην οθόνη, ως υποθέσουμε δέκα από τις εκατό. Με αυτό τον τρόπο θα υπάρχει δεκαπλάσιο κέρδος σε ταχύτητα και σε οικονομία χρήσης της μνήμης. Με τη μέθοδο της ανακύκλωσης, όταν ο χρήστης θα αρχίσει να μετακινείται προς τα πάνω ή προς τα κάτω στη λίστα και νέα στοιχεία θα εμφανίζονται στην οθόνη, αντί να δημιουργείται μια νέα όψη από την αρχή για κάθε στοιχείο που εμφανίζεται, θα ανακυκλώνεται και θα επαναχρησιμοποιείται η όψη των στοιχείων που πλέον δεν είναι ορατά στην οθόνη με αποτέλεσμα τα δεδομένα των μη ορατών στοιχείων να αντικαθίστανται από τα δεδομένα των ορατών στις ίδιες θέσεις.

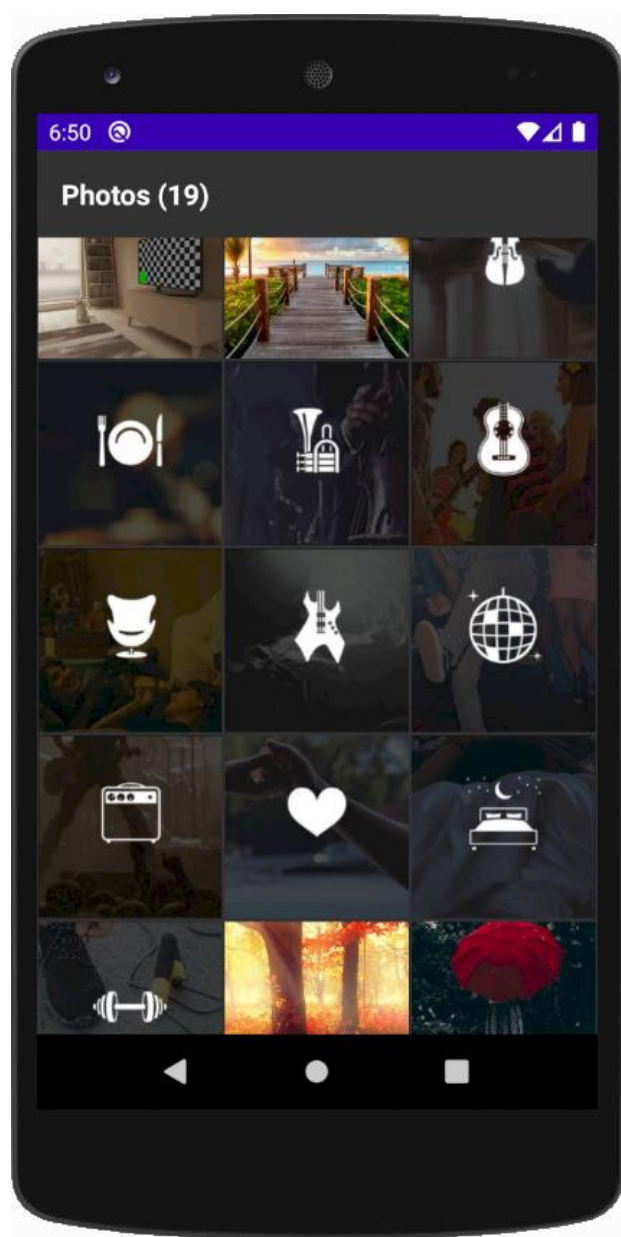
Αυτή η μέθοδος βελτιώνει σε μεγάλο βαθμό την απόδοση της εφαρμογής και μειώνει σημαντικά την κατανάλωση πόρων και ενέργειας της συσκευής. Προφανώς η συγκεκριμένη βιβλιοθήκη ενδείκνυται για την αποτελεσματική απεικόνιση κυρίως μεγάλων αλλά και μικρών συνόλων δεδομένων και στοιχείων.

Κατά την ανάπτυξη της εφαρμογής της παρούσας εργασίας χρησιμοποιήθηκε η βιβλιοθήκη RecyclerView για την αποδοτικότερη υλοποίηση βασικών λειτουργιών της εφαρμογής. Η συγκεκριμένη βιβλιοθήκη χρησιμοποιήθηκε ως το μέσο με το οποίο προβάλλονται στην οθόνη φωτογραφίες και βίντεο σε μορφή λίστας. Επίσης χρησιμοποιήθηκε και στην αρχική οθόνη της

εφαρμογής για να προβάλλει μια λίστα από κουμπιά, τα οποία οδηγούν στις βασικές δραστηριότητες της εφαρμογής.

Προκειμένου να δουλέψει σωστά ένα RecyclerView πρέπει να οριστεί ένας layoutManager σε αυτό. Ο layoutManager καθορίζει τη διάταξη που θα έχουν τα αντικείμενα πάνω στο RecyclerView. Είναι υπεύθυνος να ενημερώνει τη λίστα τότε πρέπει να ανακυκλώσει ένα στοιχείο της που έχει βγει εκτός ορίων της οθόνης. Σε γενικές γραμμές, ο βασικός ρόλος του layoutManager είναι η ύπαρξη ενός διαχειριστή για την δυναμική αλλαγή της διάταξης του συνόλου των δεδομένων.

Για την προβολή της λίστας των φωτογραφιών και βίντεο, ορίστηκε ως layoutManager ο GridLayoutManager. Πρόκειται για έναν διαχειριστή διατάξεων που εμφανίζει τα αντικείμενα της λίστας σε μορφή πλέγματος. Στην εφαρμογή, οι στήλες του πλέγματος ορίστηκαν να παρατάσσονται σε τριάδες.



Εικόνα 3: Προβολή λίστας φωτογραφιών σε μορφή πλέγματος με τη χρήση της βιβλιοθήκης RecyclerView

Η έκδοση της βιβλιοθήκης RecyclerView που χρησιμοποιήθηκε κατά την υλοποίηση της εφαρμογής είναι η 1.1.0 η οποία αποτελεί την πιο πρόσφατη σταθερή έκδοση της συγκεκριμένης βιβλιοθήκης έως και σήμερα. Είναι διαθέσιμη και η έκδοση 1.2.0 η οποία βρίσκεται σε δοκιμαστικό στάδιο καθώς είναι ακόμα υπό ανάπτυξη.

- **Glide**

Το Glide είναι μια απλή και εύχρηστη βιβλιοθήκη, η οποία προτείνεται και από τη Google για λειτουργίες σχετικές με φόρτωση εικόνων από τον αποθηκευτικό χώρο της συσκευής ή από το Internet σε στοιχεία γραφικής διασύνδεσης στο Android. Συγκεκριμένα, η βιβλιοθήκη υποστηρίζει μεταξύ άλλων τη λήψη, την αποκωδικοποίηση, και την προβολή φωτογραφιών, στατικών και κινούμενων εικόνων (GIFs) καθώς και video thumbnails.

Συχνά, κατά το φόρτωμα μιας εικόνας μεγάλου μεγέθους σε μια εφαρμογή, παρατηρούνται προβλήματα στην απόδοση τα οποία επηρεάζουν την ομαλή λειτουργία της εφαρμογής και κάνουν δυσάρεστη την εμπειρία χρήσης της. Ωστόσο, το Glide επιτρέπει γρήγορη και ομαλή κύλιση (scrolling) σε οποιουδήποτε είδους λίστα με εικόνες, ανεξαρτήτως του πλήθους φωτογραφιών που περιέχει. Επίσης αποτελεί ένα χρήσιμο εργαλείο για λειτουργίες που σχετίζονται με την επεξεργασία εικόνας όπως περικοπή, αλλαγή μεγέθους κ.λπ.

Σχετικά με την απόδοση κατά τη φόρτωση των εικόνων, το Glide λαμβάνει υπόψιν δύο βασικούς παράγοντες:

- Την ταχύτητα με την οποία οι εικόνες μπορούν να αποκωδικοποιηθούν.
- Τον όγκο περιττών δεδομένων και πληροφορίας (junk) που προέκυψε κατά την αποκωδικοποίηση της εικόνας.

Για να έχουν οι χρήστες μια ευχάριστη εμπειρία χρησιμοποιώντας μια εφαρμογή, οι εικόνες που φορτώνονται σε αυτήν, θα πρέπει όχι μόνο να εμφανίζονται γρήγορα αλλά και χωρίς να δημιουργούν μεγάλο όγκο περιττής πληροφορίας καταναλώνοντας διαθέσιμους πόρους από την επεξεργαστική ισχύ και τη μνήμη της συσκευής. Η βιβλιοθήκη Glide προκειμένου να διασφαλίσει την ομαλότερη και ταχύτερη δυνατή φόρτωση των εικόνων χρησιμοποιεί τεχνικές για την ελαχιστοποίηση του χρόνου αποθήκευσης και αποκωδικοποίησης των εικόνων. Επίσης επαναχρησιμοποιεί πόρους όπως συστοιχίες από byte και Bitmaps για να διαχειριστεί μεγάλους σωρούς περιττής πληροφορίας που πιθανών να δημιουργηθούν κατά την διαδικασία της φόρτωσης.

Η φόρτωση μιας εικόνας με τη χρήση της βιβλιοθήκης Glide είναι μια πολύ απλή διαδικασία καθώς απαιτεί μόλις τρεις παραμέτρους:

1. Το αντικείμενο περιβάλλοντος (Context) – Το Glide απαιτεί μια μεταβλητή περιβάλλοντος τύπου Context για την διεκπεραίωση των εσωτερικών λειτουργιών που απαιτούνται για την φόρτωση της εικόνας. Η συγκεκριμένη μεταβλητή θα περαστεί ως παράμετρος της **with()** συνάρτησης.
2. Την εικόνα – Πρόκειται για την εικόνα που πρόκειται να εμφανισθεί. Μπορεί να είναι ένα URL από το Internet, ένα URI ή η διεύθυνση ενός αρχείου από τον αποθηκευτικό χώρο της συσκευής. Αποτελεί την παράμετρο της **load()** συνάρτησης.
3. Το στοιχείο στο οποίο θα εμφανισθεί – Πρόκειται για το στοιχείο γραφικής διασύνδεσης στο οποίο θα εμφανισθεί η εικόνα, συνήθως κάποιο ImageView. Αποτελεί την παράμετρο της **into()** συνάρτησης.


```
Glide.with(context)
    .load(mData.get(position).getImage())
    .into(holder.imageViewButton);
```

Πίνακας 1: Φόρτωση εικόνας με τη χρήση της βιβλιοθήκης Glide

Τα σημαντικότερα χαρακτηριστικά και πλεονεκτήματα της βιβλιοθήκης Glide είναι:

- **Προσωρινή αποθήκευση δεδομένων(Disk caching):** Κατά τη φόρτωση μιας εικόνας σε κάποιο στοιχείο γραφικής διασύνδεσης (π.χ. ImageView), το Glide αποθηκεύει την εικόνα στην προσωρινή μνήμη του δίσκου (disk cache). Εάν η ίδια εικόνα φορτωθεί σε δυο διαφορετικά στοιχεία γραφικής διασύνδεσης τότε θα αποθηκευτούν δυο διαφορετικά αντίγραφα της εικόνας στην προσωρινή μνήμη. Αυτό, από τη μια μειώνει τον διαθέσιμο αποθηκευτικό χώρο της μνήμης αλλά από την άλλη αυξάνει την ταχύτητα επεξεργασίας και την απόδοση. Η επιλογή αυτή μπορεί να απενεργοποιηθεί από τον προγραμματιστή ανάλογα με τις ανάγκες της εφαρμογής.
- **Απλό και εύχρηστο:** Το Glide παρέχει μία πολύ απλοϊκή στη χρήση διεπαφή. Χρειάζονται ελάχιστες γραμμές κώδικα για το φόρτωμα μιας εικόνας καθώς το μόνο που απαιτεί είναι να οριστεί η διεύθυνση στη μνήμη ή το URL της εικόνας που θέλουμε να προβάλλουμε και το στοιχείο γραφικής διασύνδεσης στο οποίο θα προβληθεί.
- **Υποστήριξη GIF:** Η βιβλιοθήκη Glide υποστηρίζει τη χρήση κινούμενων εικόνων (GIFs), οι οποίες τείνουν να είναι ιδιαίτερα δημοφιλείς τα τελευταία χρόνια.
- **Δυνατότητες προσαρμογής:** Το Glide παρέχει ένα μεγάλο εύρος σε ρυθμίσεις προσαρμογών και παραμετροποίησης ώστε να εξυπηρετεί αποτελεσματικά τις ανάγκες της κάθε εφαρμογής.
- **Αξιοπιστία και υποστήριξη:** Η συγκεκριμένη βιβλιοθήκη προτείνεται από τη Google καθώς έχει χρησιμοποιηθεί και σε αρκετά δικά της project ανοικτού κώδικα όπως την εφαρμογή Google I/O. Το Glide δημιουργήθηκε από την εταιρία Bump Technologies και υποστηρίζεται μέχρι και σήμερα με νέες εκδόσεις και πλήρη τεκμηρίωση.

Η προβολή εικόνων αποτελεί ένα από τα κυριότερα χαρακτηριστικά της εφαρμογής της παρούσης εργασίας. Η διαχείριση τους είναι αρκετά απαιτητική καθώς προϋποθέτει γρήγορη φόρτωση και εμφάνιση των εικόνων, χειρισμό της μνήμης και των διαθέσιμων πόρων της συσκευής, ρύθμιση της εικόνας στο κατάλληλο για προβολή μέγεθος και διαχείριση και αντιμετώπιση των εξαιρέσεων που μπορεί να προκύψουν.

Σε μια εφαρμογή διαχείρισης και οργάνωσης φωτογραφιών και βίντεο όπως η συγκεκριμένη, είναι αναγκαίο να γίνεται γρήγορα και χωρίς κόστος σε πόρους η εμφάνιση των εικόνων. Η βιβλιοθήκη Glide χρησιμοποιήθηκε ευρέως κατά την ανάπτυξη της εφαρμογής για την φόρτωση φωτογραφιών, GIFs, video thumbnails και την αντιμετώπιση όλων των απαιτήσεων που έχει η συγκεκριμένη διαδικασία. Η χρήση του Glide παρείχε πολλαπλά οφέλη τόσο στη δημιουργία μιας διεπαφής η οποία θα προσφέρει μια ευχάριστη εμπειρία χρήσης, όσο και προγραμματιστικά, καθώς η εύκολη ενσωμάτωση της βιβλιοθήκης στις ανάγκες της εφαρμογής επιτάχυνε σημαντικά την ανάπτυξη και την υλοποίηση της.

Η έκδοση της βιβλιοθήκης Glide που χρησιμοποιήθηκε κατά την υλοποίηση της εφαρμογής είναι η 4.11.0 η οποία αποτελεί την πιο πρόσφατη σταθερή έκδοση της συγκεκριμένης βιβλιοθήκης έως και σήμερα. Για την ενσωμάτωση και χρήση του Glide σε μια Android εφαρμογή απαιτείται SDK έκδοση 14 ή πιο πρόσφατη.

- **BottomNavigationView**

Η πλοήγηση σε διάφορα τμήματα μιας εφαρμογής μπορεί να γίνει με τη χρήση ενός *fragment transaction*, δηλαδή μιας προγραμματιστικής διαδικασίας η οποία αναλαμβάνει να αντικαταστήσει την τρέχουσα διάταξη της οθόνης με μια άλλη, ώστε να μεταφερθεί ο χρήστης σε κάποιο άλλο τμήμα της εφαρμογής. Ωστόσο, σε εφαρμογές μεγάλης κλίμακας με πολλά διαφορετικά τμήματα (fragments) και διατάξεις, η εγγραφή πολλαπλών *fragment transactions* σε διαφορετικά σημεία του κώδικα αποτελεί μια πολύπλοκη διαδικασία και μια μη αποδοτική πρακτική, καθώς καθυστερεί τη συνολική υλοποίηση της εφαρμογής και καθιστά τη συντήρηση του κώδικα εξαιρετικά δύσκολη λόγω της πολυπλοκότητας της.

Οι μπάρες πλοήγησης βοηθούν στην ανάπτυξη περισσότερο φιλικών προς τον χρήστη αλλά και ως προς την ανάπτυξη εφαρμογών. Το *BottomNavigationView* είναι ένα στοιχείο γραφικής διασύνδεσης του Android, το οποίο προσφέρει δυνατότητες πλοήγησης σε διαφορετικές διατάξεις μιας εφαρμογής. Συγκεκριμένα πρόκειται για μια μπάρα η οποία βρίσκεται αγκιστρωμένη μόνιμα στο κάτω μέρος της οθόνης της φορητής συσκευής και περιέχει ορισμένους προορισμούς-διατάξεις της εφαρμογής, επιτρέποντας στους χρήστες να εξερευνούν και να πλοηγούνται εύκολα μεταξύ των συγκεκριμένων προορισμών. Κάθε προορισμός αναπαρίσταται από ένα εικονίδιο και προαιρετικά από κάποιο κείμενο. Ο χρήστης πατώντας σε ένα εικονίδιο μεταφέρεται άμεσα στον αντίστοιχο προορισμό.

Το *BottomNavigationView* προτείνεται να χρησιμοποιείται για τις εξής περιπτώσεις:

- Προορισμούς/διατάξεις υψηλής προτεραιότητας και χρήσης κατά τη λειτουργία της εφαρμογής, οι οποίοι πρέπει να είναι προσβάσιμοι από κάθε τμήμα της εφαρμογής.
- Τρεις έως πέντε βασικούς προορισμούς/διατάξεις.

Πλεονεκτήματα:

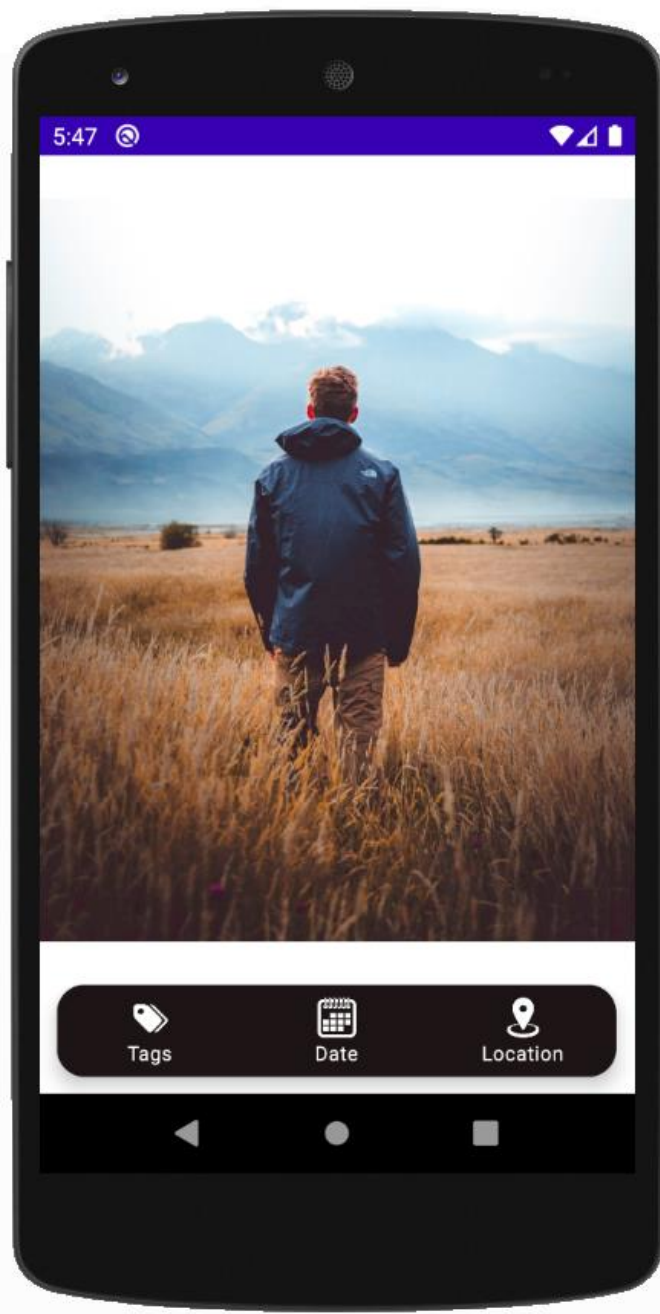
- Οι εφαρμογές γίνονται περισσότερο φιλικές ως προς τη χρήση
- Διευκολύνει την διαδικασία ανάπτυξης και υλοποίησης των εφαρμογών
- Συμβάλλει στην αποδοτικότερη διαχείριση και συντήρηση του κώδικα μειώνοντας το πλήθος των *fragment transactions*
- Προσφέρει πλουσιότερα και αισθητικά ομορφότερα *user interfaces* καθώς παρέχει δυνατότητες κινούμενων εφέ κατά τις μεταβάσεις στους προορισμούς-διατάξεις

Μειονεκτήματα:

- Μπορεί να χρησιμοποιηθεί για τρεις έως πέντε βασικούς προορισμούς-διατάξεις. Για περισσότερους ή λιγότερους αποτελεί μη αποδοτική πρακτική και προτείνονται διαφορετικά εργαλεία
- Η χρήση του περιορίζεται μόνο στις φορητές συσκευές όπως κινητά τηλέφωνα και tablets

Κατά την ανάπτυξη της εφαρμογής της παρούσας εργασίας χρησιμοποιήθηκε το στοιχείο γραφικής διασύνδεσης BottomNavigationView για τη δημιουργία μιας φιλικής προς το χρήστη διεπαφής. Συγκεκριμένα χρησιμοποιήθηκε για την εύκολη πλοήγηση μεταξύ ορισμένων βασικών λειτουργιών της εφαρμογής.

Η μπάρα πλοήγησης BottomNavigationView εμφανίζεται στο κάτω μέρος της οθόνης, κάθε φορά που ο χρήστης επιλέγει για προβολή μια φωτογραφία ή ένα βίντεο. Το BottomNavigationView περιέχει ένα μενού πλοήγησης με τρεις βασικούς προορισμούς. Οι προορισμοί αυτοί, αφορούν τις τρεις βασικές λειτουργίες οργάνωσης και ταξινόμησης των φωτογραφιών και βίντεο που προσφέρει η εφαρμογή. Πρόκειται για την διαχείριση των ετικετών, της ημερομηνίας και της τοποθεσίας των αρχείων.



Εικόνα 4: Μπάρα πλοήγησης με τη χρήση του στοιχείου γραφικής διασύνδεσης BottomNavigationView

Η χρήση του έχει ως αποτέλεσμα τη δημιουργία μιας εύχρηστης διεπαφής καθώς οι χρήστες μπορούν να αντιληφθούν εύκολα, με το άνοιγμα ενός αρχείου φωτογραφίας ή βίντεο, τις βασικές λειτουργίες που τους προσφέρει η εφαρμογή και την εύκολη πλοήγηση μεταξύ αυτών.

Η υλοποίηση μιας μπάρας πλοήγησης βοήθησε επίσης στη συγγραφή καθαρότερου κώδικα και στην εφαρμογή καλών πρακτικών καθώς οι τρεις προορισμοί που περιλαμβάνει αποτελούν βασικές λειτουργίες υψηλής προτεραιότητας και χρήσης της εφαρμογής. Επομένως οποιαδήποτε άλλη προσέγγιση θα οδηγούσε σε περισσότερες γραμμές κώδικα αυξάνοντας την πολυπλοκότητα του project και δυσκολεύοντας την διαδικασία ανάπτυξης της εφαρμογής.

Για να χρησιμοποιηθεί το συγκεκριμένο εργαλείο πρέπει να γίνει δήλωση και εισαγωγή της βιβλιοθήκης Material Components για Android, στο αρχείο build.gradle στο μπλοκ των εξαρτήσεων. Η βιβλιοθήκη Material Components έχει αναπτυχθεί και υποστηρίζεται από τη Google. Προσφέρει ένα πλήθος εργαλείων και στοιχείων γραφικής διασύνδεσης, μεταξύ αυτών και το BottomNavigationView, τα οποία διευκολύνουν την ανάπτυξη και την υλοποίηση αισθητικά όμορφων και λειτουργικών εφαρμογών στο Android.

Μερικές δημοφιλείς εφαρμογές φορητών συσκευών που χρησιμοποιούν το συγκεκριμένο εργαλείο πλοήγησης είναι το Facebook, το YouTube, το Instagram και το Twitter.

- **Place Autocomplete**

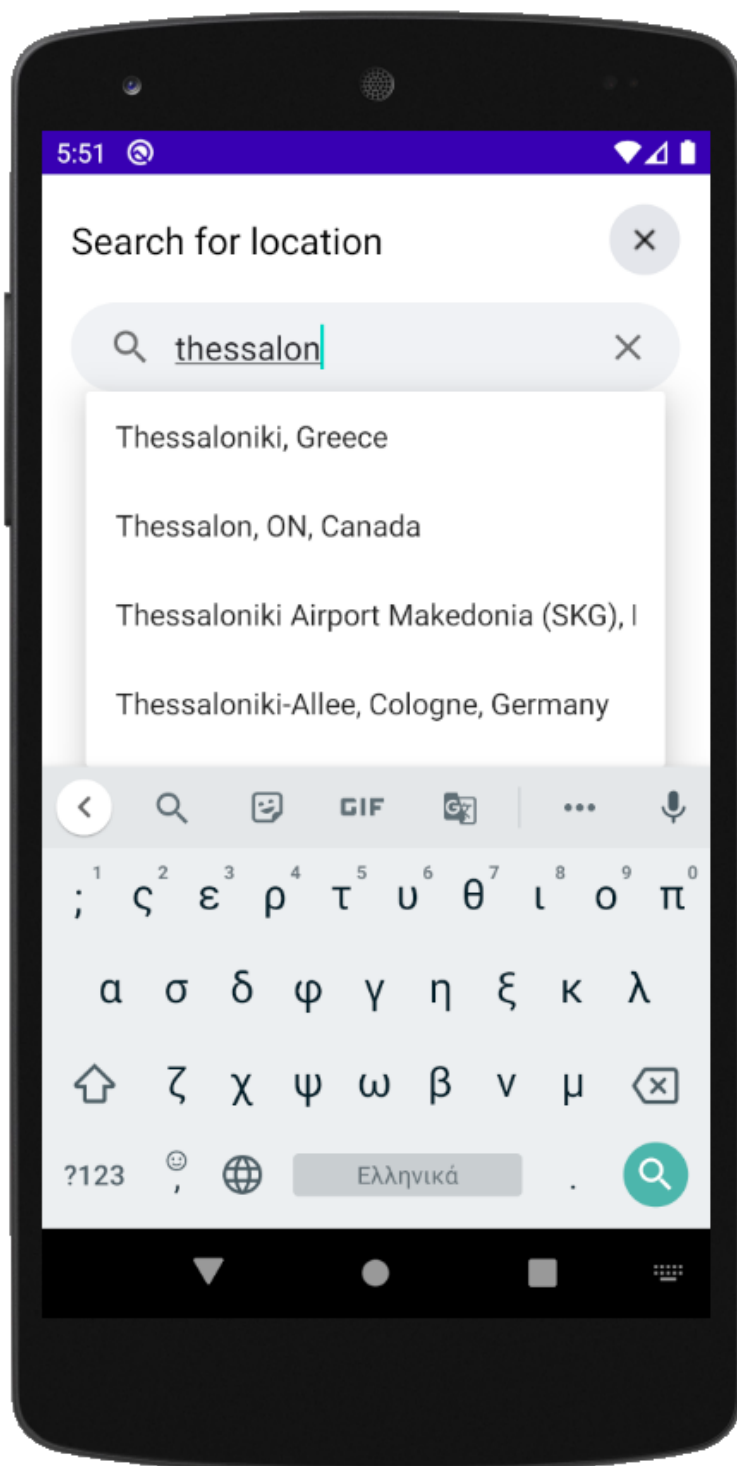
Το Place Autocomplete είναι μια web υπηρεσία η οποία δέχεται HTTP αιτήματα και επιστρέφει προβλέψεις τοποθεσιών ως απάντηση στα αιτήματα αυτά. Το αίτημα περιγράφεται από ένα κείμενο αναζήτησης και προαιρετικά γεωγραφικά όρια. Η υπηρεσία Place Autocomplete μπορεί να χρησιμοποιηθεί για την παροχή λειτουργιών αυτόματης συμπλήρωσης γεωγραφικών αναζητήσεων βάσει κειμένου. Οι χρήστες στέλνουν αιτήματα πληκτρολογώντας στην αναζήτηση κειμένου και η υπηρεσία επιστρέφει μεταξύ άλλων μέρη, όπως χώρες, πόλεις, διευθύνσεις, επιχειρήσεις, σημεία ενδιαφέροντος και γεωγραφικές τοποθεσίες.

Η υπηρεσία Place Autocomplete έχει τη δυνατότητα να ταιριάζει ολόκληρες λέξεις αλλά και μέρη αυτών. Επομένως οι εφαρμογές μπορούν να στέλνουν αιτήματα στην υπηρεσία κάθε φορά που ο χρήστης πληκτρολογεί και να λαμβάνουν τις αντίστοιχες προβλέψεις από αυτή. Οι επιστρεφόμενες προβλέψεις επιδιώκουν να βοηθήσουν το χρήστη στη επιλογή του επιθυμητού μέρους που αναζητά. Επιπλέον υπάρχει η δυνατότητα να περιοριστούν τα αποτελέσματα σε μια καθορισμένη χώρα ή και περιοχή, ορίζοντας τις κατάλληλες παραμέτρους στο HTTP αίτημα που αποστέλλεται στην υπηρεσία. Τα αποτελέσματα επιστρέφονται είτε σε μορφή JSON είτε σε XML μορφή.

Η συγκεκριμένη υπηρεσία έχει δημιουργηθεί από την Google και αποτελεί μέρος της πλατφόρμας Places API των χαρτών της Google (Google Maps). Η χρήση της απαιτεί τη δημιουργία ενός κλειδιού API (API key) για την πλατφόρμα Google Maps. Ένα κλειδί API αποτελεί ένα μοναδικό χαρακτηριστικό που επικυρώνει τα αιτήματα που σχετίζονται με την εκάστοτε εφαρμογή, project ή υπηρεσία που το χρησιμοποιεί για λόγους χρέωσης και χρήσης. Η χρήση της υπηρεσίας Place Autocomplete δεν απαιτεί την ύπαρξη χαρτών της Google στην εφαρμογή που τη χρησιμοποιεί προκειμένου να λειτουργήσει.

Στην εφαρμογή της παρούσας εργασίας γίνεται χρήση της υπηρεσίας Place Autocomplete για την αυτόματη συμπλήρωση γεωγραφικών τοποθεσιών. Συγκεκριμένα, η υπηρεσία χρησιμοποιείται όταν ο χρήστης επιλέξει να ορίσει μια τοποθεσία για κάποιο αρχείο φωτογραφίας ή βίντεο. Στην

οθόνη εμφανίζεται ένα παράθυρο διαλόγου με μία μπάρα αναζήτησης. Καθώς ο χρήστης πληκτρολογεί στο πεδίο αναζήτησης, στέλνονται HTTP αιτήματα στην υπηρεσία Place Autocomplete, και επιστρέφεται μια λίστα από προβλέψεις σχετικές με την είσοδο του χρήστη. Η λίστα ανανεώνεται κάθε φορά που ο χρήστης πληκτρολογεί ή διαγράφει έναν χαρακτήρα στο πεδίο αναζήτησης, καθώς για κάθε αλλαγή του κειμένου αναζήτησης ένα νέο αίτημα αποστέλλεται στην υπηρεσία. Οι προβλέψεις που προβάλλονται στη λίστα συμπεριλαμβάνουν μεταξύ άλλων χώρες, πόλεις, διευθύνσεις επιχειρήσεις και σημεία ενδιαφέροντος.



Εικόνα 5: Αυτόματη συμπλήρωση τοποθεσίας με τη χρήση της υπηρεσίας Place Autocomplete

Οι υπηρεσίες αυτόματης συμπλήρωσης τόσο για την εύρεση τοποθεσιών όσο και για αναζητήσεις κάθε είδους, χρησιμοποιούνται όλο και περισσότερο τα τελευταία χρόνια και εξελίσσονται συνεχώς. Αξίζει να σημειωθεί ότι πολλές δημοφιλείς εφαρμογές και ιστοσελίδες, όπως Facebook, Instagram, Twitter, YouTube και άλλες χρησιμοποιούν υπηρεσίες αυτόματης συμπλήρωσης για λειτουργίες εύρεσης τοποθεσίας, χρηστών, βίντεο κ.λπ.

Η χρήση της υπηρεσίας προσφέρει μια πιο φιλική προς τον χρήστη και μοντέρνα διεπαφή η οποία συμβαδίζει με τις τάσεις και τις δυνατότητες σύγχρονων εφαρμογών. Ο χρήστης μπορεί εύκολα και γρήγορα να αναζητήσει τοποθεσίες από όλο τον κόσμο και στη συνέχεια να επιλέξει μια από τις διαθέσιμες προβλέψεις ως τοποθεσία για το συγκεκριμένο αρχείο. Για τη λειτουργία της υπηρεσίας, προφανώς απαιτείται η σύνδεση της φορητής συσκευής στο Internet καθώς η χρήση της περιλαμβάνει την αποστολή HTTP αιτημάτων και τη λήψη δεδομένων σε μορφή JSON. Επίσης, βασική προϋπόθεση για την προσθήκη της υπηρεσίας στην εφαρμογή ήταν η δημιουργία ενός κλειδιού API για την χρήση της πλατφόρμας Google Maps η οποία τροφοδοτεί με τα δεδομένα της, την υπηρεσία Place Autocomplete.

3.1.3 Ανάλυση βασικών κλάσεων και κώδικα

Στη συγκεκριμένη ενότητα γίνεται αναφορά σε ορισμένες αξιοσημείωτες κλάσεις και σημεία του κώδικα, τα οποία καθόρισαν την ανάπτυξη και την τελική μορφή της εφαρμογής και έπαιξαν σημαντικό ρόλο στην υλοποίηση των βασικών λειτουργιών και δυνατοτήτων της.

- **File**

Η κλάση File αποτελεί μια βοηθητική κλάση η οποία αντιπροσωπεύει και μοντελοποιεί προγραμματιστικά τα αρχεία φωτογραφιών και βίντεο του χρήστη. Ουσιαστικά πρόκειται για μια κλάση του εννοιολογικού μοντέλου της εφαρμογής (domain model) καθώς κάθε αντικείμενο της, αποτελεί μια οντότητα για ένα αρχείο φωτογραφίας ή βίντεο της συσκευής. Κάθε λειτουργία της εφαρμογής που περιλαμβάνει τη χρήση κάποιας φωτογραφίας ή βίντεο, σχετίζεται με αντικείμενα της συγκεκριμένης κλάσης.

Τα αντικείμενα της κλάσης File χαρακτηρίζονται από δυο ιδιότητες τις οποίες δέχεται και ως παραμέτρους στον constructor της:

1. Μια μεταβλητή *String* στην οποία αποθηκεύεται η διεύθυνση (path) που βρίσκεται το συγκεκριμένο αρχείο στον αποθηκευτικό χώρο της συσκευής του χρήστη.
2. Μια μεταβλητή *Type* η οποία χαρακτηρίζει τον τύπο του συγκεκριμένου αρχείου, δηλαδή αν πρόκειται για αρχείο φωτογραφίας ή βίντεο. Η κλάση *Type* είναι μια κλάση τύπου enum, δηλαδή μια κλάση που αντιπροσωπεύει ένα σύνολο σταθερών μεταβλητών. Η enum *Type* περιέχει δυο σταθερές τιμές, *IMAGE* και *VIDEO*.

Η κλάση File διαθέτει getter μεθόδους για την ανάκτηση των δυο ιδιοτήτων της, καθώς και για μια μεταβλητή τύπου *boolean*. Η συγκεκριμένη μεταβλητή χρησιμοποιείται κατά την μαζική επιλογή αρχείων για ταυτόχρονη επεξεργασία και δηλώνει εάν ένα αρχείο είναι επιλεγμένο ή όχι από το χρήστη. Επιπλέον υλοποιεί τη διασύνδεση *Serializable*, μέσω της οποίας έχει τη δυνατότητα να περάσει αντικείμενα File από μια δραστηριότητα σε μια άλλη με τη χρήση προθέσεων (Intents).

```

public class File implements Serializable {

    private String path;
    private Type type;
    private boolean selected;

    public File(String path, Type type) {
        this.path = path;
        this.type = type;
    }

    public String getPath() {
        return path;
    }

    public Type getType() {
        return type;
    }

    public boolean isSelected() {
        return selected;
    }

    public void setSelected(boolean selected) {
        this.selected = selected;
    }
}

```

Πίνακας 2: Η κλάση File

- **MediaGallery**

Η κλάση MediaGallery αποτελεί άλλη μια βοηθητική κλάση της εφαρμογής. Είναι μια stateless κλάση καθώς δεν έχει ιδιότητες ούτε έχει οριστεί κάποιος constructor για τη δημιουργία αντικειμένων της. Αποτελείται μόνο από static μεθόδους, οι οποίες μπορούν να κληθούν από οποιαδήποτε σημείο της εφαρμογής χωρίς τη βοήθεια ενός αντικειμένου της. Η συγκεκριμένη κλάση παρέχει μια μορφή διασύνδεσης της εφαρμογής με τον αποθηκευτικό χώρο της συσκευής, συγκεκριμένα με τα αρχεία φωτογραφιών και βίντεο που είναι αποθηκευμένα σε αυτόν.

Η κλάση MediaGallery συγκροτείται από 5 static μεθόδους οι οποίες είναι υπεύθυνες για την ανάκτηση των δεδομένων που απαιτούνται για την διαχείριση των φωτογραφιών και των βίντεο της συσκευής. Τα δεδομένα αυτά αφορούν κυρίως διευθύνσεις και ημερομηνίες καθώς και τους γονικούς φακέλους (άλμπουμ) των συγκεκριμένων αρχείων. Η ανάκτηση αυτών των δεδομένων γίνεται από τον αποθηκευτικό χώρο της συσκευής και όχι από την βάση δεδομένων της εφαρμογής η οποία θα αναλυθεί σε παρακάτω ενότητα. Συγκεκριμένα, για την ανάκτηση των δεδομένων χρησιμοποιείται η κλάση MediaStore του Android.

Ουσιαστικά, πρόκειται για ένα API το οποίο παρέχει πρόσβαση στις συλλογές πολυμέσων της συσκευής, δηλαδή αρχεία εικόνας, βίντεο και ήχου. Επιπλέον, προσφέρει όλες τις βασικές λειτουργίες διαχείρισης αυτών των πολυμέσων, όπως δημιουργία νέου αρχείου, άνοιγμα, επεξεργασία ή διαγραφή ενός υπάρχοντος. Η κλάση MediaStore παρέχει πρόσβαση τόσο στον τοπικό χώρο αποθήκευσης της συσκευής (internal storage), όσο και στον εξωτερικό χώρο αποθήκευσης (external storage), δηλαδή σε μια κάρτα μνήμης που πιθανώς να έχει συνδεθεί με την συσκευή. Στην κλάση MediaGallery της εφαρμογής, γίνεται χρήση του MediaStore και από τις

πέντε μεθόδους της κλάσης, για την ανάκτηση των απαραίτητων δεδομένων από τις φωτογραφίες και τα βίντεο του αποθηκευτικού χώρου.

Οι πέντε μέθοδοι της κλάσης MediaGallery είναι οι εξής:

- **listOfImages():** Η συνάρτηση επιστρέφει μια λίστα με αντικείμενα της κλάσης File που αναλύθηκε παραπάνω. Τα αντικείμενα της επιστρεφόμενης λίστας αντιπροσωπεύουν όλα τα αρχεία φωτογραφιών που βρίσκονται στον αποθηκευτικό χώρο της συσκευής. Τα αρχεία ανακτώνται από τη μνήμη της συσκευής με τη χρήση της κλάσης MediaStore και προστίθενται στη λίστα ως αντικείμενα File, ταξινομημένα σε φθίνουσα χρονολογική σειρά.

```
public static ArrayList<File> listOfImages(Context context){  
  
    Uri uri;  
    Cursor cursor;  
    int column_index_data;  
    ArrayList<File> listOfAllImages = new ArrayList<>();  
    String absolutePathOfImage;  
    uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;  
  
    String [] projection = {MediaStore.MediaColumns.DATA,  
        MediaStore.Images.Media.BUCKET_DISPLAY_NAME,  
        MediaStore.Images.Media.DATE_TAKEN};  
  
    String orderBy = MediaStore.Images.Media.DATE_TAKEN;  
    cursor = context.getContentResolver().query(uri, projection,  
        null,  
        null,  
        orderBy+" DESC");  
    column_index_data = cursor.  
        getColumnIndexOrThrow(MediaStore.MediaColumns.DATA);  
    while (cursor.moveToNext()){  
        File file = new  
            File(cursor.getString(column_index_data),Type.IMAGE);  
        listOfAllImages.add(file);  
    }  
    cursor.close();  
    return listOfAllImages;  
}
```

Πίνακας 3: Η μέθοδος listOfImages

- **listOfVideos():** Η συνάρτηση επιστρέφει μια λίστα με αντικείμενα της κλάσης File. Τα αντικείμενα της λίστας αντιπροσωπεύουν όλα τα βίντεο που βρίσκονται στον αποθηκευτικό χώρο της συσκευής. Τα αρχεία ανακτώνται από τη μνήμη της συσκευής με τη χρήση της κλάσης MediaStore και μέσω της συνάρτησης query(). Στη συνάρτηση query() θα περαστούν ως παράμετροι ένα σύνολο αλφαριθμητικών σταθερών της κλάσης MediaStore οι οποίες αφορούν τα δεδομένα που θα επιστραφούν. Στη συνέχεια τα δεδομένα επιστρέφονται σε ένα αντικείμενο τύπου Cursor. Τέλος γίνεται η ανάκτηση των δεδομένων από τη μεταβλητή cursor, και προστίθενται στην επιστρεφόμενη λίστα ως File αντικείμενα ταξινομημένα σε φθίνουσα χρονολογική σειρά.


```

public static ArrayList<File> listOfVideos(Context context){
    Uri uri;
    Cursor cursor;
    int column_index_data;
    ArrayList<File> listOfAllVideos = new ArrayList<>();
    String absolutePathOfVideo;

    uri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;
    String[] projection = { MediaStore.Video.VideoColumns.DATA ,
        MediaStore.Video.Media.DISPLAY_NAME,
        MediaStore.Video.Media.BUCKET_DISPLAY_NAME,
        MediaStore.Video.Media.BUCKET_ID,
        MediaStore.Video.Media.DATE_TAKEN};

    String orderBy = MediaStore.Video.Media.DATE_TAKEN;
    cursor = context.getContentResolver().query(uri, projection,
        null,
        null,
        orderBy + " DESC");

    column_index_data = cursor.
        getColumnIndexOrThrow(MediaStore.Video.Media.DATA);
    while (cursor.moveToNext()){
        File file = new
            File(cursor.getString(column_index_data),Type.VIDEO);
        listOfAllVideos.add(file);
    }
    cursor.close();
    return listOfAllVideos;
}

```

Πίνακας 4: Η μέθοδος listOfVideos

- **getListOfAlbums():** Η συνάρτηση επιστρέφει μια συλλογή HashMap η οποία περιέχει ζεύγη κλειδιού-τιμής. Ως κλειδιά στη συγκεκριμένη συλλογή αποθηκεύονται όλα τα ονόματα των άλμπουμ και των συλλογών που έχουν φωτογραφίες ή βίντεο στον αποθηκευτικό χώρο της συσκευής, και ως τιμή του κάθε κλειδιού, η διεύθυνση μνήμης (path) της πιο πρόσφατης φωτογραφίας που έχει προστεθεί στο συγκεκριμένο άλμπουμ-κλειδί. Η συγκεκριμένη συνάρτηση εξυπηρετεί στην ανάκτηση των απαραίτητων δεδομένων για την αναπαράσταση και προβολή των άλμπουμ της συσκευής στην εφαρμογή.

```

public static HashMap<String, String> getListOfAlbums(Context context){

    String[] projection = new String[] {MediaStore.MediaColumns.DATA,
        MediaStore.Images.ImageColumns.BUCKET_DISPLAY_NAME,
        MediaStore.Images.Media.DATE_TAKEN,
        MediaStore.Images.Media.DATE_ADDED,
        MediaStore.Video.Media.DATE_TAKEN,
        MediaStore.Video.Media.DATE_ADDED,
        MediaStore.Video.Media.BUCKET_DISPLAY_NAME};

    String orderByImg = MediaStore.Images.Media.DATE_ADDED;
    String orderByVid = MediaStore.Video.Media.DATE_ADDED;

    Cursor imageCursor = context.getContentResolver().
        query(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, projection,

```

```

        null,
        null,
        orderByImg + " DESC");

    Cursor videoCursor = context.getContentResolver().
        query(MediaStore.Video.Media.EXTERNAL_CONTENT_URI, projection,
            null,
            null,
            orderByVid + " DESC");

    int imagePathColumn =
    imageCursor.getColumnIndexOrThrow(MediaStore.MediaColumns.DATA);
    int videoPathColumn =
    videoCursor.getColumnIndexOrThrow(MediaStore.Video.Media.DATA);

    HashMap<String, String> albumHashMap = new HashMap<>();

    HashMap<String, String> namesAndDates = new HashMap<>();
    while (imageCursor.moveToNext()) {

        if(!albumHashMap.containsKey(imageCursor.
            getString((imageCursor.getColumnIndex(MediaStore.Images.ImageColumns.BUCKET_
            _DISPLAY_NAME))) + "\n")){

            albumHashMap.put(imageCursor.getString((imageCursor.getColumnIndex(MediaSto
            re.Images.ImageColumns.BUCKET_DISPLAY_NAME))) + "\n",
                imageCursor.getString(imagePathColumn));

            namesAndDates.put(imageCursor.getString((imageCursor.getColumnIndex(MediaSt
            ore.Images.ImageColumns.BUCKET_DISPLAY_NAME))) + "\n",
                imageCursor.getString((imageCursor.getColumnIndex(MediaStore.Images.ImageCo
            lumns.DATE_ADDED))));
        }
    }

    long videoDate = 0;
    while (videoCursor.moveToNext()) {

        try{
            videoDate =
            Long.parseLong(videoCursor.getString(videoCursor.getColumnIndex(MediaStore.
            Video.Media.DATE_ADDED)));
        }catch (Exception e){
            e.printStackTrace();
        }

        if(albumHashMap.containsKey(videoCursor.
            getString((videoCursor.getColumnIndex(MediaStore.Video.VideoColumns.BUCKET_
            DISPLAY_NAME))) + "\n")) {

            try {
                if (videoDate >
                Long.parseLong(namesAndDates.get(videoCursor.
                getString((videoCursor.getColumnIndex(MediaStore.Video.VideoColumns.BUCKET_
                DISPLAY_NAME))) + "\n"))){

```

```

albumHashMap.put(videoCursor.getString((videoCursor.getColumnIndex(MediaStore.Video.VideoColumns.BUCKET_DISPLAY_NAME))) + "\n",
                videoCursor.getString(videoPathColumn));

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

    }else {

albumHashMap.put(videoCursor.getString((videoCursor.getColumnIndex(MediaStore.Video.VideoColumns.BUCKET_DISPLAY_NAME))) + "\n",
                videoCursor.getString(videoPathColumn));

        }

    }
    imageCursor.close();
    videoCursor.close();

    return albumHashMap;
}

```

Πίνακας 5: Η μέθοδος getListOfAlbums

- getSizeOfAlbums():** Η συνάρτηση επιστρέφει μια συλλογή HashMap η οποία περιέχει ζεύγη κλειδιού-τιμής. Ως κλειδιά στη συγκεκριμένη συλλογή αποθηκεύονται όλα τα ονόματα των άλμπουμ και των συλλογών που έχουν φωτογραφίες ή βίντεο στον αποθηκευτικό χώρο της συσκευής, και ως τιμή του κάθε κλειδιού, το μέγεθος του συγκεκριμένου άλμπουμ-κλειδιού, δηλαδή το πλήθος των αρχείων φωτογραφιών και βίντεο που περιέχει. Η συγκεκριμένη συνάρτηση εξυπηρετεί στην ανάκτηση των απαραίτητων δεδομένων για την αναπαράσταση και προβολή των άλμπουμ της συσκευής στην εφαρμογή.

```

public static HashMap<String, Integer> getSizeOfAlbums(Context context) {
    String[] projection = new String[]
    {MediaStore.Images.ImageColumns.BUCKET_DISPLAY_NAME,
     MediaStore.Video.Media.BUCKET_DISPLAY_NAME};

    Cursor imageCursor = context.getContentResolver().
        query(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, projection,
            null,
            null,
            null);

    Cursor videoCursor =
    context.getContentResolver().query(MediaStore.Video.Media.EXTERNAL_CONTENT_
    URI,
        projection,
        null,
        null,
        null);

    HashMap<String, Integer> albumSizeHashMap = new HashMap<>();

    while (imageCursor.moveToNext()) {

        if(!albumSizeHashMap.containsKey(imageCursor.

```

```

getString((imageCursor.getColumnIndex(MediaStore.Images.ImageColumns.BUCKET_
_DISPLAY_NAME))) + "\n"){
albumSizeHashMap.put(imageCursor.getString((imageCursor.getColumnIndex(Medi
aStore.Images.ImageColumns.BUCKET_DISPLAY_NAME))) + "\n", 1);

    }else {
albumSizeHashMap.put(imageCursor.getString((imageCursor.getColumnIndex(Medi
aStore.Images.ImageColumns.BUCKET_DISPLAY_NAME))) + "\n",

albumSizeHashMap.get(imageCursor.getString((imageCursor.getColumnIndex(Medi
aStore.Images.ImageColumns.BUCKET_DISPLAY_NAME))) + "\n")+1);
    }
}

while (videoCursor.moveToNext()) {

    if(!albumSizeHashMap.containsKey(videoCursor.

getString((videoCursor.getColumnIndex(MediaStore.Video.VideoColumns.BUCKET_
_DISPLAY_NAME))) + "\n")){

albumSizeHashMap.put(videoCursor.getString((videoCursor.getColumnIndex(Medi
aStore.Video.VideoColumns.BUCKET_DISPLAY_NAME))) + "\n", 1);
    }else {
albumSizeHashMap.put(videoCursor.getString((videoCursor.getColumnIndex(Medi
aStore.Video.VideoColumns.BUCKET_DISPLAY_NAME))) + "\n",

albumSizeHashMap.get(videoCursor.getString((videoCursor.getColumnIndex(Medi
aStore.Video.VideoColumns.BUCKET_DISPLAY_NAME))) + "\n")+1);
    }
}

imageCursor.close();
videoCursor.close();

return albumSizeHashMap;
}

```

Πίνακας 6: Η μέθοδος `getSizeOfAlbums`

- **getAlbumItems():** Η συνάρτηση δέχεται ως παράμετρο το όνομα ενός άλμπουμ από τον αποθηκευτικό χώρο της συσκευής και επιστρέφει μια λίστα με αντικείμενα της κλάσης `File`. Τα αντικείμενα της επιστρεφόμενης λίστας αντιπροσωπεύουν όλα τα αρχεία φωτογραφιών και βίντεο που περιέχει το συγκεκριμένο άλμπουμ.

```

public static ArrayList<File> getAlbumItems(Context context, String albumName){

    String[] projection = new String[] {MediaStore.MediaColumns.DATA,
        MediaStore.Images.ImageColumns.BUCKET_DISPLAY_NAME,
        MediaStore.Images.Media.DATE_TAKEN,
        MediaStore.Video.Media.DATE_TAKEN,
        MediaStore.Video.Media.BUCKET_DISPLAY_NAME};

    String orderByImg = MediaStore.Images.Media.DATE_TAKEN;
    String orderByVid = MediaStore.Video.Media.DATE_TAKEN;
    Cursor imageCursor = context.getContentResolver().
        query(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, projection,
            null,
            null,
            orderByImg + " DESC");

    Cursor videoCursor = context.getContentResolver().
        query(MediaStore.Video.Media.EXTERNAL_CONTENT_URI, projection,
            null,
            null,
            orderByVid + " DESC");

    int imagePathColumn =
imageCursor.getColumnIndexOrThrow(MediaStore.MediaColumns.DATA);
    int videoPathColumn =
videoCursor.getColumnIndexOrThrow(MediaStore.Video.Media.DATA);

    ArrayList<File> files = new ArrayList<>();

    while (imageCursor.moveToNext()) {
        if(imageCursor.getString(imageCursor.getColumnIndex(MediaStore.Images.
ImageColumns.BUCKET_DISPLAY_NAME)).equals(albumName)){
            File file = new File(imageCursor.
                getString(imagePathColumn),Type.IMAGE);
            files.add(file);
        }
    }
    while (videoCursor.moveToNext()) {
        if(videoCursor.getString(videoCursor.getColumnIndex(MediaStore.Video.
VideoColumns.BUCKET_DISPLAY_NAME)).equals(albumName)){
            File file = new File(videoCursor.
                getString(videoPathColumn),Type.VIDEO);
            files.add(file);
        }
    }

    imageCursor.close();
    videoCursor.close();
    return files;
}

```

Πίνακας 7: Η μέθοδος getAlbumItems

Οι πέντε παραπάνω συναρτήσεις αποτελούν τον κώδικα της κλάσης MediaGallery. Η συγκεκριμένη κλάση μέσω αυτών των συναρτήσεων παρέχει τα απαραίτητα δεδομένα στην εφαρμογή, για την απεικόνιση των πολυμέσων που υπάρχουν στον αποθηκευτικό χώρο της συσκευής, δηλαδή είναι υπεύθυνη για την ανάκτηση των φωτογραφιών, των βίντεο και των άλμπουμ-συλλογών στα οποία βρίσκονται.

- **GalleryAdapter**

Σε προηγούμενη ενότητα έγινε αναφορά στην βιβλιοθήκη RecyclerView, η οποία χρησιμοποιείται στην εφαρμογή ως το μέσο για την απεικόνιση των φωτογραφιών και βίντεο της συσκευής σε μορφή λίστας. Εκτός από τον ορισμό ενός LayoutManager, απαιτείται η δημιουργία και ο ορισμός ενός Adapter προκειμένου να δουλέψει σωστά ένα RecyclerView. Ο Adapter αποτελεί το σημαντικότερο κομμάτι της αρχιτεκτονικής του RecyclerView.

Στο Android, οι Adapters λειτουργούν ως γέφυρα μεταξύ στοιχείων γραφικής διασύνδεσης όπως RecyclerView, ListView, GridView, ViewPager, Spinner και των δεδομένων που θέλουμε να προβάλλουμε σε αυτά τα στοιχεία. Ο κύριος ρόλος τους είναι να παρέχουν πρόσβαση στα στοιχεία των δεδομένων και να δημιουργήσουν μια όψη (View) για κάθε ένα από αυτά. Ουσιαστικά οι Adapters είναι υπεύθυνοι για την μετατροπή των δεδομένων στην κατάλληλη μορφή και την προβολή τους σε στοιχεία γραφικής διασύνδεσης.

Ένας RecyclerView Adapter έχει και αυτός την αρμοδιότητα να διαχειρίζεται το σύνολο των δεδομένων που πρόκειται να τοποθετήσει στο RecyclerView, ώστε να εμφανισθούν στην οθόνη με την κατάλληλη μορφή. Σε αντίθεση όμως με τους ListView και ViewPager Adapters, οι οποίοι δημιουργούν νέες όψεις σύμφωνα με το πλήθος των στοιχείων του συνόλου των δεδομένων και τοποθετούν σε αυτές τα στοιχεία του συνόλου, ένας RecyclerView Adapter ακολουθεί μια διαφορετική προσέγγιση για να διαχειρίζεται και να προβάλλει τα δεδομένα.

Κάθε στοιχείο της λίστας ενός RecyclerView καθορίζεται από ένα αντικείμενο της κλάσης ViewHolder. Ένα αντικείμενο ViewHolder περιγράφει μέσω κάποιον μεταδεδομένων την μορφή που θα έχει ένα στοιχείο του συνόλου σχετικά με τη θέση του στη λίστα. Αρχικά όταν δημιουργείται ένα αντικείμενο ViewHolder δεν έχει στοιχεία συσχετισμένα με αυτό. Υπεύθυνος για την συσχέτιση των δεδομένων με τα αντικείμενα ViewHolder είναι ο RecyclerView Adapter. Επομένως ο συγκεκριμένος Adapter δημιουργεί ViewHolder αντικείμενα τα οποία ορίζουν την μορφή κάθε στοιχείου του συνόλου. Τα αντικείμενα ViewHolder αποθηκεύονται στην προσωρινή μνήμη (cache) ώστε να μπορούν να επαναχρησιμοποιηθούν εάν χρειαστεί.

Η υλοποίηση της κλάσης ViewHolder στην παρούσα εφαρμογή έχει γίνει στο αρχείο GalleryAdapter.java και αποτελεί μια εμφωλευμένη κλάση εντός της κλάσης GalleryAdapter η οποία βρίσκεται στο ίδιο αρχείο και θα εξηγηθεί παρακάτω. Η κλάση ViewHolder επεκτείνει την κλάση RecyclerView.ViewHolder και υλοποιεί τις διασυνδέσεις View.OnClickListener και View.OnLongClickListener. Στον Constructor της αρχικοποιούνται όλα τα απαραίτητα στοιχεία γραφικής διασύνδεσης και τα μεταδεδομένα της διάταξης που θα καθορίσουν την μορφή-όψη του συγκεκριμένου ViewHolder. Επιπλέον η κλάση ViewHolder εφόσον υλοποιεί τις διασυνδέσεις View.OnClickListener και View.OnLongClickListener πρέπει να υπερβεί τις μεθόδους onClick() και onLongClick(). Η μέθοδος onClick() χειρίζεται έναν ακροατή συμβάντων ο οποίος είναι υπεύθυνος για τις ενέργειες που θα λάβουν χώρα όταν ο χρήστης επιλέγει μια φωτογραφία ή ένα βίντεο από τη λίστα, ενώ η μέθοδος onLongClick χειρίζεται έναν ακροατή συμβάντων ο οποίος καθορίζει τις ενέργειες που θα λάβουν χώρα όταν ο χρήστης έχει κάνει παρατεταμένο κλικ σε ένα αρχείο της λίστας.

```

public class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener,View.OnLongClickListener{

    ImageView file;
    ImageView playArrow;
    ProgressBar progressBar;
    LinearLayout linearLayout;
    CheckBox checkBox;

    public ViewHolder(@NonNull View itemView) {
        super(itemView);

        file = itemView.findViewById(R.id.file_item);
        playArrow = itemView.findViewById(R.id.video_play);
        progressBar = itemView.findViewById(R.id.progBar);
        linearLayout = itemView.findViewById(R.id.LinearCheckBox);
        checkBox = itemView.findViewById(R.id.checkboxItem);
        itemView.setOnClickListener(this);
        itemView.setOnLongClickListener(this);
    }

    @Override
    public void onClick(View v) {
        clickListener.onClick(v,getAdapterPosition());
    }

    @Override
    public boolean onLongClick(View v) {
        clickListener.onItemLongClick(v,getAdapterPosition());
        return true;
    }
}

```

Πίνακας 8: Η κλάση ViewHolder

Η κλάση GalleryAdapter υλοποιεί τη λειτουργικότητα ενός RecyclerView Adapter. Επεκτείνει την κλάση RecyclerView.Adapter<GalleryAdapter.ViewHolder>. Ο constructor της δέχεται τέσσερις παραμέτρους, μια μεταβλητή περιβάλλοντος τύπου *Context*, τη λίστα με τις διευθύνσεις των αρχείων που θα εμφανιστούν στην οθόνη μέσω του RecyclerView, μια μεταβλητή *MediaListener* η οποία λειτουργεί ως ακροατής συμβάντων και μια μεταβλητή *SelectionManager* η οποία διαχειρίζεται τη λειτουργικότητα της επιλογής πολλαπλών αρχείων για μαζική επεξεργασία. Η κλάση GalleryAdapter πρέπει να υπερβεί υποχρεωτικά τρεις abstract μεθόδους της κλάσης RecyclerView.Adapter την οποία επεκτείνει. Οι τρεις μέθοδοι είναι οι εξής:

1. **getItemCount():** Η μέθοδος επιστρέφει το πλήθος των στοιχείων του συνόλου δεδομένων που θα χρησιμοποιηθεί στον συγκεκριμένο Adapter. Στην συγκεκριμένη περίπτωση πρόκειται για το μέγεθος της λίστας με τις διευθύνσεις των αρχείων.
2. **onCreateViewHolder():** Πρόκειται για τη μέθοδο που διαχειρίζεται τη δημιουργία και την επαναχρησιμοποίηση των ViewHolder αντικειμένων. Καλείται αυτόματα από το RecyclerView κάθε φορά που πρέπει να δημιουργηθεί και να αρχικοποιηθεί ένα ViewHolder με την καθορισμένη του μορφή-όψη(View). Ουσιαστικά, ορίζεται η διάταξη

που θα έχει το ViewHolder που πρόκειται να δημιουργηθεί ώστε να τοποθετηθούν σε αυτό τα δεδομένα του συνόλου. Στην συγκεκριμένη περίπτωση, η διάταξη των ViewHolder αντικειμένων της κλάσης GalleryAdapter, ορίζεται μέσω ενός αρχείου διάταξης XML (layout file), το οποίο περιέχει τα στοιχεία γραφικής διασύνδεσης στα οποία θα τοποθετηθούν οι φωτογραφίες και τα βίντεο της λίστας.

- 3. onBindViewHolder():** Το RecyclerView καλεί τη συγκεκριμένη μέθοδο για να φορτώσει και να προβάλλει δεδομένα σε μια συγκεκριμένη θέση(ViewHolder). Η μέθοδος onBindViewHolder() συνδέει τα στοιχεία του συνόλου με τα αντικείμενα ViewHolder που δημιούργησε η συνάρτηση onCreateViewHolder(). Στην συγκεκριμένη περίπτωση, ανακτά τις διευθύνσεις των φωτογραφιών και των βίντεο από τη λίστα και τροφοδοτεί με αυτές τη διάταξη του αντίστοιχο ViewHolder. Η κάθε διάταξη περιέχει ένα ImageView στοιχείο γραφικής διασύνδεσης, το οποίο θα εμφανίσει την κατάλληλη φωτογραφία ή βίντεο στην οθόνη. Αυτή η συνάρτηση αποτελεί το πιο "ακριβό" τμήμα κώδικα ολόκληρου του Adapter καθώς καλείται κάθε φορά που προβάλλονται δεδομένα στην οθόνη. Το RecyclerView χρησιμοποιείται κυρίως για μεγάλα σύνολα δεδομένων, επομένως χρειάζεται προσοχή κατά τον σχεδιασμό και την υλοποίηση της συγκεκριμένης μεθόδου ώστε να μην υπάρχουν προβλήματα στην απόδοση.

Οι παραπάνω τρεις μέθοδοι υλοποιούν τη βασική λειτουργικότητα ενός RecyclerView Adapter στην κλάση GalleryAdapter. Επιπλέον η κλάση διαθέτει μεθόδους getters/setters για την ανάκτηση και την επεξεργασία των βασικών ιδιοτήτων της. Η συγκεκριμένη κλάση αποτελεί μια από τις βασικότερες κλάσεις της εφαρμογής καθώς χρησιμοποιείται για την επίτευξη των κυριότερων λειτουργιών και δυνατοτήτων της εφαρμογής. Είναι υπεύθυνη για όλο τον μηχανισμό προβολής αρχείων φωτογραφίας και βίντεο στην οθόνη του χρήστη. Επίσης προσφέρει αφαίρεση, απλότητα και μείωση της πολυπλοκότητας του κώδικα καθώς με μια κλάση εξασφαλίζεται ένα μέρος της λειτουργικότητας όλων των βασικών δραστηριοτήτων της εφαρμογής, εφόσον όλες σχετίζονται με την προβολή φωτογραφιών και βίντεο για την επίτευξη των λειτουργιών τους.

```
public class GalleryAdapter extends RecyclerView.Adapter<GalleryAdapter.ViewHolder>{

    private Context context;
    private ArrayList<File> files;
    MediaListener clickListener;
    private SelectionManager selectionManager;

    public GalleryAdapter(Context context, ArrayList<File> files, MediaListener
clickListener, SelectionManager selectionManager) {
        this.context = context;
        this.files = files;
        this.clickListener = clickListener;
        this.selectionManager = selectionManager;
    }

    public ArrayList<File> getFiles() {
        return files;
    }

    public void setFiles(ArrayList<File> files) {
        this.files = files;
    }

    public MediaListener getClickListener() {
        return clickListener;
    }
}
```



```

    public void setClickListener(MediaListener clickListener) {
        this.clickListener = clickListener;
    }

    @NonNull
    @Override
    public GalleryAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        return new GalleryAdapter.ViewHolder(
LayoutInflater.from(context).inflate(R.layout.gallery_item,parent,false)
        );
    }

    @Override
    public void onBindViewHolder(@NonNull GalleryAdapter.ViewHolder holder, final int
position) {
        final String filePath = files.get(position).getPath();
        final Type fileType = files.get(position).getType();
        final ProgressBar progressBar = holder.progressBar;
        final ImageView playArrow = holder.playArrow;
        final File file = files.get(position);
        Glide.with(context).load(filePath)
            .listener(new RequestListener<Drawable>() {
                @Override
                public boolean onLoadFailed(@Nullable GlideException e, Object
model, Target<Drawable> target, boolean isFirstResource) {
                    return false;
                }

                @Override
                public boolean onResourceReady(Drawable resource, Object model,
Target<Drawable> target, DataSource dataSource, boolean isFirstResource) {
                    progressBar.setVisibility(View.GONE);
                    if(fileType == Type.VIDEO){
                        playArrow.setVisibility(View.VISIBLE);
                    }
                    if(fileType == Type.IMAGE &&
playArrow.getVisibility()==View.VISIBLE){
                        playArrow.setVisibility(View.GONE);
                    }

                    return false;
                }
            })
            .into(holder.file);

        holder.checkBox.setOnCheckedChangeListener(null);

        holder.checkBox.setChecked(file.isSelected());

        if(selectionManager.position == position){
            holder.checkBox.setChecked(true);
            selectionManager.position = -1;
        }
        if (selectionManager.isActionMode){
            holder.linearLayout.setVisibility(View.VISIBLE);
        }else {
            holder.linearLayout.setVisibility(View.GONE);
            holder.checkBox.setChecked(false);
        }

        holder.checkBox.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

```

```

        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            file.setSelected(isChecked);
            selectionManager.check(buttonView, position);
        }
    });
}

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

public class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener, View.OnLongClickListener{

    ImageView file;
    ImageView playArrow;
    ProgressBar progressBar;
    LinearLayout linearLayout;
    CheckBox checkBox;

    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        file = itemView.findViewById(R.id.file_item);
        playArrow = itemView.findViewById(R.id.video_play);
        progressBar = itemView.findViewById(R.id.progBar);
        linearLayout = itemView.findViewById(R.id.LinearCheckBox);
        checkBox = itemView.findViewById(R.id.checkboxItem);
        itemView.setOnClickListener(this);
        itemView.setOnLongClickListener(this);
    }

    @Override
    public void onClick(View v) {
        clickListener.onClick(v, getAdapterPosition());
    }

    @Override
    public boolean onLongClick(View v) {
        clickListener.onItemLongClick(v, getAdapterPosition());
        return true;
    }
}
}

```

Πίνακας 9: Η κλάση GalleryAdapter

- **FullSizeAdapter**

Οι χρήστες της εφαρμογής έχουν την δυνατότητα να εξερευνήσουν τις εικόνες και τα βίντεο που βρίσκονται στον αποθηκευτικό χώρο της συσκευής. Όταν ένας χρήστης επιλέξει για προβολή ένα αρχείο εικόνας ή βίντεο από τη συλλογή, το συγκεκριμένο αρχείο προβάλλεται στην οθόνη του χρήστη σε πλήρες μέγεθος.

Η κλάση FullSizeAdapter είναι υπεύθυνη για τη φόρτωση και την προβολή των εικόνων και βίντεο σε πλήρες μέγεθος, καθώς και για την πλοήγηση μεταξύ αυτών μέσω κύλισης (slide). Επιπλέον,

υλοποιεί μια μορφή διασύνδεσης μεταξύ των χρηστών και των λειτουργιών της εφαρμογής, καθώς υποστηρίζει τη μετάβαση σε ορισμένες από τις βασικότερες δυνατότητες της, όπως την εισαγωγή και επεξεργασία ετικετών, ημερομηνίας, τοποθεσίας και την αναπαραγωγή αρχείων βίντεο.

Αρχικά, για την υλοποίηση της κύλισης μεταξύ σελίδων στην οθόνη απαιτείται η χρήση ενός συγκεκριμένου διαχειριστή διατάξεων (Layout Manager), του ViewPager. Για να λειτουργήσει σωστά το συγκεκριμένο εργαλείο πρέπει να οριστεί ένας Adapter, ο οποίος θα τροφοδοτεί τον ViewPager με τις σελίδες που πρόκειται να προβάλλει. Το ρόλο αυτό, έχει αναλάβει η κλάση FullSizeAdapter, της οποίας τα αντικείμενα ορίζονται ως Adapters για το διαχειριστή διατάξεων ViewPager.

Σε επίπεδο κώδικα, ο constructor της κλάσης FullSizeAdapter, απαιτεί δύο παραμέτρους για την δημιουργία αντικειμένων της, μια μεταβλητή περιβάλλοντος τύπου Context και τη λίστα με τα αρχεία που πρόκειται να εμφανιστούν στην οθόνη σε πλήρες μέγεθος ως σελίδες του ViewPager. Η κλάση FullSizeAdapter επεκτείνει την κλάση PagerAdapter, η οποία παρέχει τις απαραίτητες μεθόδους για την τροφοδοσία και τη διαχείριση των σελίδων που φορτώνονται στον ViewPager. Επομένως η κλάση FullSizeAdapter για να παρέχει τη λειτουργικότητα ενός ViewPager Adapter πρέπει να υπερβεί υποχρεωτικά τις εξής τέσσερις μεθόδους:

1. **getCount():** Επιστρέφει το πλήθος των σελίδων που είναι διαθέσιμες για προβολή από τον συγκεκριμένο ViewPager. Στην συγκεκριμένη περίπτωση πρόκειται για το μέγεθος της λίστας που δέχεται ως παράμετρο ο constructor της κλάσης FullSizeAdapter, δηλαδή της λίστας με τα αρχεία που τροφοδοτεί ο FullSizeAdapter τον ViewPager.
2. **instantiateItem(ViewGroup container, int position):** Πρόκειται για τη συνάρτηση που θα δημιουργήσει και θα επιστρέψει τη σελίδα για τη θέση που της έχει δοθεί ως παράμετρος. Επιπλέον δέχεται ως παράμετρο άλλη μια μεταβλητή, container, η οποία αντιπροσωπεύει τη συλλογή στην οποία θα προστεθεί και θα προβληθεί η σελίδα. Στην συγκεκριμένη περίπτωση, η μεταβλητή container αναφέρεται στον ViewPager και η θέση αναφέρεται σε ένα από τα αντικείμενα της λίστας με τα αρχεία. Η συνάρτηση φορτώνει τη διάταξη που έχει οριστεί για τη σελίδα, η οποία περιέχει την εικόνα ή το βίντεο που επέλεξε για προβολή ο χρήστης καθώς και μια μπάρα πλοήγησης για τη διαχείριση των ετικετών, της ημερομηνίας και της τοποθεσίας του συγκεκριμένου αρχείου. Επιπλέον προσθέτει το συγκεκριμένο αρχείο στη βάση δεδομένων της εφαρμογής εάν δεν υπάρχει ήδη. Στη συνέχεια προσθέτει τη σελίδα-διάταξη που δημιούργησε στην συλλογή του ViewPager και τέλος την επιστρέφει. Η συγκεκριμένη συνάρτηση πιθανώς να καλεστεί αυτόματα περισσότερες από μια φορές, καθώς εκτός από τη δημιουργία της σελίδας που πρόκειται να προβληθεί στην οθόνη του χρήστη, το σύστημα πρέπει να γνωρίζει και τη σελίδα που ακολουθεί πριν και μετά από αυτή. Επομένως, σε περίπτωση που ο χρήστης κάνει κύλιση προς τα αριστερά ή προς τα δεξιά, το σύστημα πρέπει να είναι σε θέση να προβάλλει την κατάλληλη σελίδα.
3. **isViewFromObject(View view, Object object):** Ο ViewPager συσχετίζει κάθε σελίδα (View) με ένα αντικείμενο-κλειδί (object). Το συγκεκριμένο κλειδί αποτελεί ένα μοναδικό αναγνωριστικό για την κάθε σελίδα, το οποίο χαρακτηρίζει και διακρίνει τις σελίδες ανεξαρτήτως της θέσης που έχουν στον Adapter. Η μέθοδος isViewFromObject() καλείται από τον Adapter καθώς ο χρήστης πλοηγείται μεταξύ των σελίδων του ViewPager. Ουσιαστικά, ελέγχει εάν η σελίδα η οποία περάστηκε ως παράμετρος (view) κατά την κλήση της, σχετίζεται με το αντικείμενο-κλειδί (object) το οποίο έχει επιστραφεί από τη

συνάρτηση `instantiateltem()`. Η σελίδα (`view`) που σχετίζεται με το αντικείμενο-κλειδί `object` θα είναι εκείνη που θα προβληθεί στην οθόνη του χρήστη.

4. **`destroyItem(ViewGroup container, int position, Object object)`:** Ο `ViewPager` στην συλλογή του έχει το πολύ τρεις σελίδες, αυτή που προβάλλει, την προηγούμενη αυτής και την επόμενη αυτής εάν υπάρχει. Καθώς ο χρήστης μεταβαίνει από τη μια σελίδα στην άλλη, ο `Adapter` πρέπει να ενημερώνει τη συλλογή αντικαθιστώντας τις σελίδες κατάλληλα. Η συνάρτηση `destroyItem()` καλείται αυτόματα από τον `Adapter` του `ViewPager` όταν μια σελίδα πρέπει να αφαιρεθεί από τη συλλογή του `ViewPager` για τη θέση που έχει δοθεί ως παράμετρος (`position`).

Εκτός από τις τέσσερις παραπάνω μεθόδους που υλοποιούν τη λειτουργικότητα του `ViewPager Adapter`, η κλάση `FullSizeAdapter` περιέχει άλλες δυο μεθόδους:

- Τη μέθοδο `onNavigationItemClicked()`, η οποία λειτουργεί ως ακροατής συμβάντων στην μπάρα πλοήγησης που βρίσκεται στο κάτω μέρος της οθόνης σε κάθε σελίδα που προβάλλει ο `ViewPager`. Καλείται κάθε φορά που ο χρήστης επιλέγει έναν προορισμό από την μπάρα πλοήγησης και εκτελεί τις κατάλληλες ενέργειες ανάλογα με την επιλογή του χρήστη. Όπως έχει προαναφερθεί οι τρεις προορισμοί της μπάρας πλοήγησης σχετίζονται με τη διαχείριση ετικετών, ημερομηνίας και τοποθεσίας των αρχείων.
- Τη μέθοδο `playVideo()` η οποία καλείται όταν ο χρήστης επιλέξει το πλήκτρο αναπαραγωγής για ένα αρχείο βίντεο. Η μέθοδος, στέλνει μια υπονοούμενη πρόθεση σε όλες τις εφαρμογές της συσκευής που υποστηρίζουν την αναπαραγωγή `video`. Στη συνέχεια εμφανίζονται στην οθόνη οι εφαρμογές που πληρούν αυτό το κριτήριο. Ο χρήστης μπορεί να επιλέξει μία από τις διαθέσιμες εφαρμογές για την αναπαραγωγή του συγκεκριμένου αρχείου βίντεο.

● **SearchActivity**

Η κλάση `SearchActivity` υλοποιεί μία από τις βασικότερες δραστηριότητες της εφαρμογής. Πρόκειται για τη δραστηριότητα που σχετίζεται με την λειτουργία της αναζήτησης των φωτογραφιών και των βίντεο της εφαρμογής. Ο χρήστης μπορεί να αναζητήσει αρχεία μέσω των ετικετών, των ημερομηνιών και των τοποθεσιών που έχει προσθέσει. Η διάταξη της δραστηριότητας περιέχει μία μπάρα αναζήτησης και ένα κουμπί που οδηγεί σε ένα παράθυρο διαλόγου στο οποίο ο χρήστης μπορεί να ορίσει ένα εύρος ημερομηνιών για τα επιστρεφόμενα αποτελέσματα. Το υπόλοιπο μέρος της οθόνης καταλαμβάνεται από τη λίστα με τα αποτελέσματα της αναζήτησης, εάν υπάρχουν.

Καθώς αποτελεί μια δραστηριότητα της εφαρμογής, η κλάση `SearchActivity` επεκτείνει την κλάση `Activity`, η οποία παρέχει ορισμένα βασικά εργαλεία για την υλοποίηση δραστηριοτήτων στο `Android`. Η λειτουργικότητα της δραστηριότητας της αναζήτησης, περιγράφεται μέσα από τις μεθόδους της κλάσης `SearchActivity`:

- **`onCreate()`:** Πρόκειται για τη μέθοδο που εκτελείται με την εκκίνηση της δραστηριότητας. Ανακτά τα στοιχεία γραφικής διασύνδεσης που υπάρχουν στη διάταξη της δραστηριότητας και αρχικοποιεί κάποιες λίστες οι οποίες χρησιμοποιούνται για το φιλτράρισμα των επιστρεφόμενων αποτελεσμάτων. Επιπλέον ορίζει έναν ακροατή συμβάντων για τη μπάρα αναζήτησης, ο οποίος θα εκτελεί τις κατάλληλες ενέργειες κάθε φορά που αλλάζει το κείμενο της αναζήτησης.
- **`onQueryTextChange(String queryText)`:** Πρόκειται για τη μέθοδο που καλείται κάθε φορά που συμβαίνει μια αλλαγή στο κείμενο της αναζήτησης. Δέχεται ως παράμετρο το

κείμενο της αναζήτησης μετά την αλλαγή, και με βάση αυτό ανακτά από τη βάση δεδομένων τα κατάλληλα αρχεία. Εάν το κείμενο της αναζήτησης είναι κενό και ο χρήστης δεν έχει ορίσει κάποιο εύρος ημερομηνίας στα επιστρεφόμενα αρχεία τότε η συνάρτηση τερματίζει χωρίς να επιστρέψει αποτελέσματα. Σε διαφορετική περίπτωση το κείμενο της αναζήτησης μπορεί να αποτελείται από πολλαπλές λέξεις ή τμήματα λέξεων. Η μέθοδος χωρίζει το κείμενο σε λέξεις με βάση του κενούς χαρακτήρες που περιέχει. Στη συνέχεια, για κάθε λέξη που προκύπτει από το χώρισμα, αναζητά και ανακτά τα αρχεία που την περιέχουν στις ετικέτες τους ή στην τοποθεσία τους. Εάν ο χρήστης έχει ορίσει κάποιο εύρος ημερομηνιών στα επιστρεφόμενα αποτελέσματα, τότε η μέθοδος ανακτά μόνο τα αρχεία που βρίσκονται εντός του εύρους ημερομηνίας που έχει οριστεί. Στη συνέχεια γίνεται έλεγχος ώστε να μην υπάρχουν διπλότυπα αρχεία στα αποτελέσματα. Τέλος, τα αρχεία που έχουν επιστραφεί από την αναζήτηση συγκεντρώνονται σε μια λίστα ως αντικείμενα Files και καλείται η συνάρτηση που θα τα προβάλει στην οθόνη.

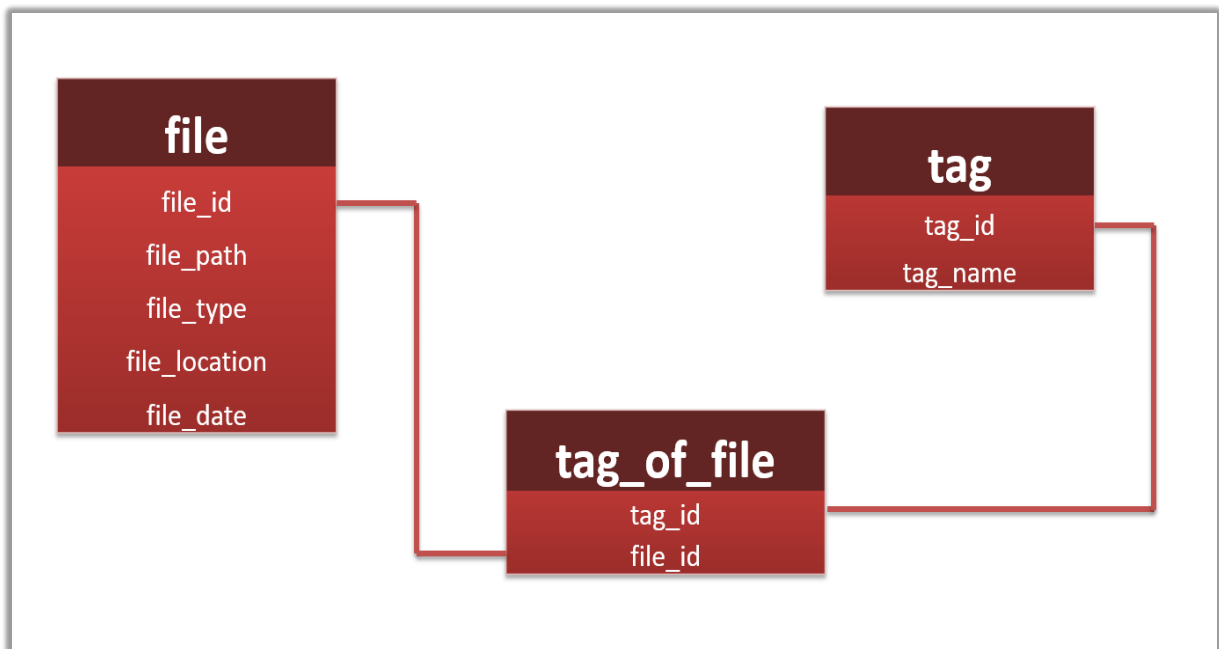
- **displayResults(ArrayList<File> fileListResults):** Η συγκεκριμένη μέθοδος εμφανίζει στην οθόνη τα αποτελέσματα της αναζήτησης, εάν υπάρχουν, κάνοντας τις απαραίτητες αλλαγές στη διάταξη της δραστηριότητας. Εάν δεν υπάρχουν αποτελέσματα που να ικανοποιούν τα κριτήρια της αναζήτησης εμφανίζεται στην οθόνη σχετικό μήνυμα. Δέχεται ως όρισμα τη λίστα με τα αρχεία φωτογραφιών και βίντεο που ανακτήθηκαν από την αναζήτηση. Τα αρχεία προβάλλονται σε μορφή πλέγματος σε ένα RecyclerView, το οποίο αρχικοποιεί η συγκεκριμένη μέθοδος, ορίζοντας τις κατάλληλες ρυθμίσεις για αυτό. Ο χρήστης έχει τη δυνατότητα να επιλέξει για προβολή οποιοδήποτε αρχείο από τα αποτελέσματα της αναζήτησης, όπως επίσης και να επιλέξει πολλαπλά αρχεία για μαζική επεξεργασία. Η μέθοδος είναι υπεύθυνη για την δυναμική αλλαγή της διάταξης της δραστηριότητας, εμφανίζοντας και αφαιρώντας στοιχεία γραφικής διασύνδεσης ανάλογα με το αν υπάρχουν ή όχι αρχεία προς προβολή από την αναζήτηση.
- **setTabLayout():** Τα αποτελέσματα της αναζήτησης ταξινομούνται σε 3 κατηγορίες, μόνο αρχεία φωτογραφιών, μόνο αρχεία βίντεο και όλα τα αρχεία μαζί. Η κατηγοριοποίηση γίνεται με τη χρήση ενός TabLayout, δηλαδή ενός στοιχείου γραφικής διασύνδεσης που χωρίζει τη διάταξη σε καρτέλες. Την διαχείριση αυτού του εργαλείου έχει αναλάβει η μέθοδος setTabLayout(). Είναι υπεύθυνη για την αρχικοποίηση του TabLayout και την ταξινόμηση των αρχείων στις κατάλληλες καρτέλες. Αρχικά ορίζει την καρτέλα με όλα τα αρχεία να προβάλλεται στον χρήστη. Ρυθμίζει τις καρτέλες που εμφανίζουν μόνο εικόνες και μόνο βίντεο να είναι διαθέσιμες μόνο όταν έχουν αρχεία να προβάλλουν, σε διαφορετική περίπτωση δεν μπορούν να επιλεγθούν από τον χρήστη. Επίσης, παρέχει ακροατές συμβάντων για όλες τις καρτέλες ώστε να προβάλλει τα κατάλληλα αρχεία όταν ο χρήστης αλλάζει καρτέλα.
- **editDateRange():** Η μέθοδος καλείται όταν ο χρήστης επιλέξει να ορίσει ένα εύρος ημερομηνιών για τα επιστρεφόμενα αποτελέσματα, πατώντας το σχετικό κουμπί στην οθόνη. Εμφανίζει στην οθόνη ένα νέο παράθυρο διαλόγου στο οποίο ο χρήστης μπορεί να ορίσει μια αρχική και μια τελική ημερομηνία ώστε να φιλτράρει τα αποτελέσματα της αναζήτησης σε αυτό το χρονολογικό εύρος.
- **dialogOnClose():** Η μέθοδος καλείται όταν κλείσει το παράθυρο διαλόγου για το εύρος ημερομηνίας των αρχείων. Εάν ο χρήστης έχει ορίσει ένα έγκυρο εύρος ημερομηνίας για την αναζήτηση, τότε η συνάρτηση ανακτά από τη βάση δεδομένων τα αρχεία που ικανοποιούν το συγκεκριμένο ημερολογιακό κριτήριο και καλεί την μέθοδο displayResults() ώστε να τα εμφανίσει στα αποτελέσματα της αναζήτησης. Εάν υπάρχουν ήδη αρχεία στα αποτελέσματα από κάποια αναζήτηση που πληκτρολόγησε ο χρήστης,

τότε καλεί τη μέθοδο `onQueryTextChanged()` για να εφαρμόσει στα συγκεκριμένα αποτελέσματα το εύρος ημερομηνίας που ορίστηκε, ώστε να ενημερωθεί η λίστα των αποτελεσμάτων κατάλληλα.

- **resetOnClick():** Η μέθοδος καλείται όταν ο χρήστης ακυρώσει το εύρος ημερομηνίας που έχει ορίσει για τα αποτελέσματα, πατώντας το κουμπί "Reset". Το συγκεκριμένο κουμπί είναι ορατό μόνο εάν ο χρήστης έχει ορίσει νωρίτερα κάποιο έγκυρο εύρος ημερομηνίας. Η μέθοδος εκτελεί τις απαραίτητες ενέργειες για να ακυρώσει το φίλτρο ημερομηνίας που έχει οριστεί και καλεί τη μέθοδο `onQueryTextChanged()` ώστε να ενημερώσει κατάλληλα τη λίστα με τα αποτελέσματα.
- **checkFileType():** Η μέθοδος δέχεται ως όρισμα ένα αντικείμενο της κλάσης `File` και ελέγχει εάν το συγκεκριμένο αρχείο είναι φωτογραφία ή βίντεο. Στη συνέχεια το τοποθετεί στην κατάλληλη λίστα. Πρόκειται για μια βοηθητική μέθοδο, η οποία χρησιμοποιείται για την κατηγοριοποίηση των αρχείων με βάση τον τύπο τους.

3.1.4 Βάση δεδομένων

Για την αποθήκευση ορισμένων βασικών δεδομένων της εφαρμογής, δημιουργήθηκε μία SQLite βάση δεδομένων. Το σχήμα της βάσης δεδομένων της εφαρμογής έχει την εξής δομή:



Εικόνα 6: Η δομή της βάσης δεδομένων

Η βάση δεδομένων της εφαρμογής αποτελείται από τρεις πίνακες:

- **file:** Αφορά τον πίνακα που αποθηκεύει τα αρχεία φωτογραφιών και βίντεο της συσκευής. Ένα αρχείο εισάγεται στον πίνακα `file` όταν ο χρήστης το ανοίξει, δηλαδή το επιλέξει για προβολή. Οι στήλες του αντιπροσωπεύουν τις βασικές πληροφορίες που απαιτούνται για τις λειτουργίες και τις δυνατότητες διαχείρισης των αρχείων τόσο από την εφαρμογή όσο και από τον χρήστη. Συγκεκριμένα αφορούν τη φυσική διεύθυνση του αρχείου στη μνήμη της συσκευής (`file_path`), τον τύπο του αρχείου (`file_type`), δηλαδή αν πρόκειται για αρχείο φωτογραφίας ή βίντεο, και την ημερομηνία (`file_date`) και την τοποθεσία (`file_location`) του αρχείου. Κατά την εισαγωγή ενός νέου αρχείου στη βάση δεδομένων, οι στήλες της

ημερομηνίας και της τοποθεσίας δεν έχουν κάποια τιμή (*null*). Η πρώτη στήλη του πίνακα είναι το ID του αρχείου το οποίο είναι πρωτεύον κλειδί στον πίνακα `file` και αποτελεί μοναδικό αναγνωριστικό της κάθε γραμμής. Ουσιαστικά πρόκειται για ένα ακέραιο αριθμό που αυξάνεται αυτόματα με την προσθήκη μιας νέας καταχώρησης στον πίνακα. Οι χρήστες της εφαρμογής δεν εμπλέκονται στην απόδοση τιμής του ID.

- **tag:** Ο συγκεκριμένος πίνακας αποθηκεύει τις ετικέτες (`tags`) των αρχείων. Αποτελείται από δυο στήλες, η μία είναι ένα μοναδικό ID (`tag_id`) το οποίο αποτελεί το πρωτεύον κλειδί του πίνακα και αυξάνεται αυτόματα από τη βάση με κάθε νέα προσθήκη, και η δεύτερη αφορά το όνομα της ετικέτας (`tag_name`) που ουσιαστικά αποθηκεύει σε μορφή κειμένου το περιεχόμενο της.
- **tag_of_file:** Πρόκειται για τον πίνακα που κρατάει πληροφορίες σχετικά με τις ετικέτες που έχουν προστεθεί σε ένα αρχείο φωτογραφίας ή βίντεο. Ουσιαστικά, αποτελεί ένα `intersection table` για τη σχέση μεταξύ ετικετών και αρχείων η οποία είναι `many-to-many` (πολλά προς πολλά). Οι στήλες του αποτελούνται από δυο ξένα κλειδιά. Το ένα αναφέρεται στο πρωτεύον κλειδί του πίνακα `tag` (`tag_id`) και το άλλο στο πρωτεύον κλειδί του πίνακα `file` (`file_id`). Με αυτό τον τρόπο, αποθηκεύονται δεδομένα σχετικά με τις ετικέτες που υπάρχουν σε κάποιο αρχείο. Μέσω κατάλληλων ελέγχων, ο συνδυασμός αυτών των δυο ξένων κλειδίων αποτελεί μοναδικό χαρακτηριστικό της κάθε γραμμής. Δηλαδή δεν μπορεί η ίδια ετικέτα να προστεθεί δεύτερη φορά στο ίδιο αρχείο.

Για την υλοποίηση της βάσης δεδομένων προγραμματιστικά, δημιουργήθηκε μία κλάση `DatabaseHelper` η οποία επεκτείνει την κλάση `SQLiteOpenHelper`, μια βοηθητική κλάση για τη διαχείριση `SQLite` βάσεων δεδομένων. Στην κλάση `DatabaseHelper` περιέχεται ο πλήρης κώδικας για την υλοποίηση και τη διαχείριση της βάσης δεδομένων της εφαρμογής.

Ο `constructor` της κλάσης `DatabaseHelper`, δέχεται ως παράμετρο, μια μεταβλητή περιβάλλοντος τύπου `Context` και στη συνέχεια καλεί τον `constructor` της υπερκλάσης (`super`) ο οποίος δέχεται 4 ορίσματα:

- την μεταβλητή `context`
- το όνομα της βάσης δεδομένων
- ένα αντικείμενο `CursorFactory`(`null`)
- τον αριθμό έκδοσης της βάσης

```
private DatabaseHelper(Context ctx) {  
    super(ctx, DATABASE_NAME, null, 1);  
}
```

Πίνακας 10: Ο constructor της κλάσης DatabaseHelper

Η κλάση `DatabaseHelper` πρέπει υποχρεωτικά να υπερβεί τις μεθόδους, `onCreate()` και `onUpgrade()`, εφόσον επεκτείνει την κλάση `SQLiteOpenHelper`.

Η μέθοδος onCreate() είναι υπεύθυνη για την δημιουργία της βάσης δεδομένων. Καλείται όταν η εφαρμογή εγκαθίσταται για πρώτη φορά σε μια συσκευή. Περιέχει κώδικα για την δημιουργία των πινάκων, καθώς και για την εισαγωγή ορισμένων default δεδομένων στον πίνακα των ετικετών.

```
@Override
public void onCreate(SQLiteDatabase db) {

    db.execSQL("CREATE TABLE " + FILE_TABLE +
        " (" + FILE_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
        + FILE_PATH + " TEXT, "
        + FILE_TYPE + " TEXT, "
        + FILE_LOCATION + " TEXT, "
        + FILE_DATE + " DATE)");

    db.execSQL("CREATE TABLE " + TAG_TABLE +
        " (" + TAG_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + TAG_NAME +
        TEXT)");

    db.execSQL("CREATE TABLE " + TAG_OF_FILE_TABLE + " (" + TAG_OF_FILE_TAG_ID +
        INTEGER, " +
        TAG_OF_FILE_FILE_ID + " INTEGER, FOREIGN KEY ( "+TAG_OF_FILE_TAG_ID+" )
        REFERENCES "+TAG_TABLE+"( "+TAG_ID+"), " +
        FOREIGN KEY ( "+TAG_OF_FILE_FILE_ID+" ) REFERENCES "+FILE_TABLE+"(
        "+FILE_ID+" ) ON DELETE CASCADE)");

    insertSampleTag(db, "Animals");
    insertSampleTag(db, "Art");
    insertSampleTag(db, "Family");
    insertSampleTag(db, "Food");
    insertSampleTag(db, "Friends");
    insertSampleTag(db, "Fun");
    insertSampleTag(db, "Holiday");
    insertSampleTag(db, "Music");
    insertSampleTag(db, "Nature");
    insertSampleTag(db, "People");
    insertSampleTag(db, "Places");
    insertSampleTag(db, "Sports");
    insertSampleTag(db, "Travel");
}
```

Πίνακας 11: Η μέθοδος onCreate()

Η μέθοδος onUpgrade() καλείται όταν πρόκειται να γίνει μια αναβάθμιση στην ήδη υπάρχουσα βάση δεδομένων. Η συγκεκριμένη μέθοδος χρησιμοποιείται κυρίως για τη διαγραφή και την εισαγωγή νέων πινάκων και γενικότερα για λειτουργίες που σχετίζονται με την αναβάθμιση της δομής της βάσης δεδομένων στη νέα έκδοση. Αυτό επιτυγχάνεται μέσω των παραμέτρων oldVersion και newVersion που δέχεται. Ουσιαστικά γίνεται έλεγχος εάν ο αριθμός έκδοσης της βάσης (newVersion) είναι μεγαλύτερος από τον αριθμό έκδοσης της βάσης που υπάρχει στη συσκευή (oldVersion). Εάν αυτό ισχύει τότε εκτελείται ο κώδικας της onUpgrade() ώστε να γίνουν οι κατάλληλες αναβαθμίσεις. Στην εφαρμογή, η onUpgrade() χρησιμοποιείται για να ξαναχτίσει την βάση δεδομένων από την αρχή, καθώς διαγράφει ότι δεδομένα έχει αποθηκευμένα και καλεί την μέθοδο onCreate() για να δημιουργήσει μια νέα, καθαρή βάση δεδομένων.


```

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TAG_TABLE);
    db.execSQL("DROP TABLE IF EXISTS " + FILE_TABLE);
    db.execSQL("DROP TABLE IF EXISTS " + TAG_OF_FILE_TABLE);
    onCreate(db);
}

```

Πίνακας 12: Η μέθοδος onUpgrade()

Η κλάση DatabaseHelper υπερβαίνει επίσης τη μέθοδο onOpen() της SQLiteOpenHelper για τις ανάγκες της εφαρμογής. Η συγκεκριμένη μέθοδος εκτελείται κάθε φορά κατά το άνοιγμα της βάσης δεδομένων, αφού έχει γίνει η δημιουργία του σχήματός της. Ο λόγος που υλοποιήθηκε είναι για να διαχειρίζεται κατάλληλα τη διαγραφή των ξένων κλειδιών, ώστε εάν κάποιο αρχείο διαγραφεί από τη βάση δεδομένων της εφαρμογής τότε να διαγράφονται και οι εγγραφές που πιθανώς να έχει στον πίνακα tag_of_file (**cascade delete**). Η ρύθμιση αυτή πρέπει να γίνεται κάθε φορά που ανοίγει η βάση δεδομένων της εφαρμογής καθώς η SQLite έχει προκαθορισμένη ρύθμιση να απενεργοποιεί τις λειτουργίες που αφορούν ξένα κλειδιά.

```

@Override
public void onOpen(SQLiteDatabase db) {
    super.onOpen(db);
    db.execSQL("PRAGMA foreign_keys=ON");
}

```

Πίνακας 13: Η μέθοδος onOpen()

Επιπλέον, στη κλάση DatabaseHelper έχει υλοποιηθεί η στατική μέθοδος getInstance(). Η συγκεκριμένη μέθοδος διασφαλίζει ότι θα υπάρχει μόνο ένα αντικείμενο DatabaseHelper σε όλη τη διάρκεια εκτέλεσης της εφαρμογής. Ουσιαστικά υλοποιεί ένα συνδυασμό των πρότυπων δημιουργίας *Abstract Factory* και *Singleton* χρησιμοποιώντας μια στατική μεταβλητή *mInstance* στην οποία θα βρίσκεται το μοναδικό αντικείμενο που θα παράγει η κλάση DatabaseHelper καθ' όλη τη διάρκεια ζωής της. Εάν δεν έχει αρχικοποιηθεί ένα αντικείμενο, δηλαδή η μεταβλητή *mInstance* είναι null τότε η μέθοδος θα το δημιουργεί, αλλιώς απλά θα επιστρέφεται το ήδη υπάρχων αντικείμενο. Με αυτό τον τρόπο αποτρέπονται πιθανές διαρροές στη μνήμη (memory leaks) καθ' όλη τη διάρκεια σύνδεσης με τη βάση δεδομένων και βελτιώνεται η συνολική απόδοση του συστήματος. Η μέθοδος getInstance() καλείται από όλες τις δραστηριότητες της εφαρμογής που χρησιμοποιούν τη βάση δεδομένων της.

```

public static DatabaseHelper getInstance(Context ctx) {
    if (mInstance == null) {
        mInstance = new DatabaseHelper(ctx.getApplicationContext());
    }
    return mInstance;
}

```

Πίνακας 14: Η μέθοδος getInstance()

Ο υπόλοιπος κώδικας της κλάσης DatabaseHelper αποτελείται από μεθόδους που σχετίζονται με τη διαχείριση των δεδομένων της βάσης. Συνοπτικά, έχουν υλοποιηθεί μέθοδοι για:

- Εισαγωγή/Ανάκτηση/Διαγραφή αρχείων
- Εισαγωγή/Ανάκτηση ετικετών
- Εισαγωγή ετικετών/ημερομηνίας/τοποθεσίας σε αρχείο
- Ανάκτηση ημερομηνίας/τοποθεσίας από αρχείο
- Διαγραφή ετικετών/ημερομηνίας από αρχείο
- Εισαγωγή default δεδομένων στον πίνακα tag

Τέλος, κάποιες σχεδιαστικές αποφάσεις άξιες αναφοράς, σχετικά με τη δομή και τη λειτουργία της βάσης δεδομένων είναι οι εξής:

Ετικέτες (tags): Οι ετικέτες αποτελούν μια ξεχωριστή οντότητα στη βάση δεδομένων της εφαρμογής καθώς αποθηκεύονται στον δικό τους πίνακα (*tag*), σε αντίθεση με την ημερομηνία και την τοποθεσία που αποτελούν ιδιότητες των αρχείων και έχουν υλοποιηθεί ως στήλες στον πίνακα *file*. Όλα αποτελούν μέσα ταξινόμησης και οργάνωσης των αρχείων, αλλά ο λόγος που οι ετικέτες είναι μια ξεχωριστή οντότητα στη βάση δεδομένων είναι η σχέση πληθικότητας (*cardinality*) που έχουν με τα αρχεία. Κάθε αρχείο έχει οριστεί να έχει το πολύ μια ημερομηνία και τοποθεσία, σε αντίθεση με τις ετικέτες που μπορεί λάβει απεριόριστες. Επομένως, οι ετικέτες έπρεπε να οριστούν ως ξεχωριστή οντότητα στη βάση δεδομένων και σε συνδυασμό με τον πίνακα *tag_of_file* που περιγράφεται παραπάνω να απεικονισθεί προγραμματιστικά η σχέση ετικετών και αρχείων ως many-to-many relationship.

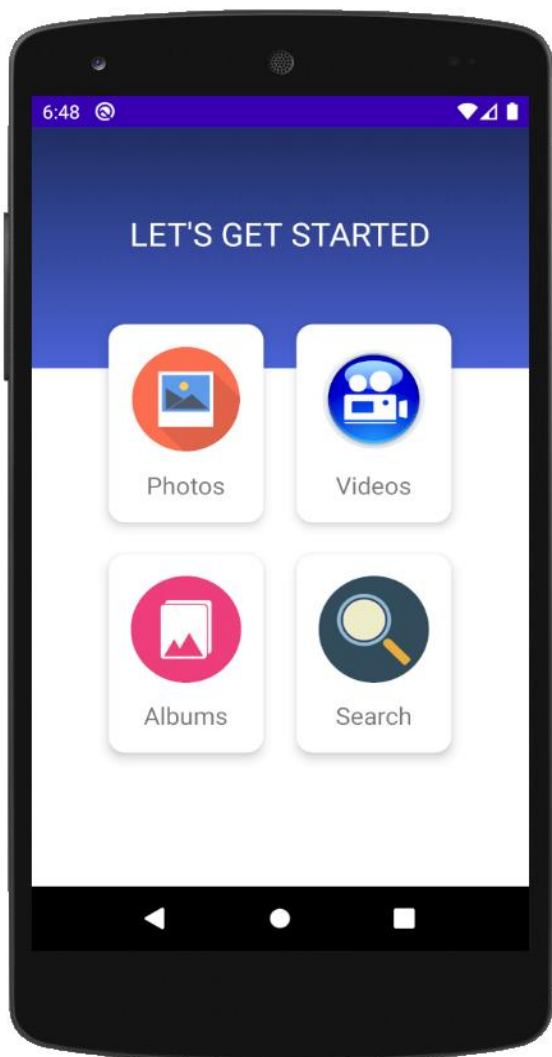
Συγχρονισμός συσκευής – βάσης δεδομένων: Κάθε φορά που ο χρήστης βρίσκεται στην αρχική οθόνη της εφαρμογής γίνεται έλεγχος για αλλαγές στα αρχεία της συσκευής. Εάν κάποιο αρχείο που είναι αποθηκευμένο στη βάση δεδομένων της εφαρμογής έχει διαγραφεί από τη μνήμη της συσκευής, τότε διαγράφεται η εγγραφή του και από τη βάση δεδομένων. Είναι σημαντικό να γίνεται αυτός ο έλεγχος στην αρχική οθόνη της εφαρμογής καθώς η βάση δεδομένων της θα είναι πάντα ενημερωμένη σύμφωνα με τον αποθηκευτικό χώρο της συσκευής. Με αυτό τον τρόπο αποτρέπονται προβλήματα κατά την αναζήτηση των αρχείων μέσω της εφαρμογής.

Εισαγωγή αρχείων: Η εφαρμογή φορτώνει και προβάλλει όλα τα άλμπουμ, τις φωτογραφίες και τα βίντεο από τον αποθηκευτικό χώρο της συσκευής μέσω της κλάσης MediaGallery που αναλύθηκε παραπάνω. Ωστόσο, δεν είναι εξαρχής όλα αυτά τα αρχεία αποθηκευμένα στη βάση δεδομένων της εφαρμογής. Ένα αρχείο εισάγεται στον πίνακα *file* όταν ο χρήστης το ανοίξει, δηλαδή το επιλέξει για προβολή μέσω της εφαρμογής. Ο λόγος αυτής της σχεδιαστικής απόφασης είναι η απόδοση του συστήματος. Κατά το άνοιγμα ενός αρχείου, η βάση δεδομένων της εφαρμογής αποθηκεύει το συγκεκριμένο αρχείο αφού ελέγξει εάν δεν υπάρχει ήδη στη βάση. Αυτή η διαδικασία κοστίζει αρκετά σε απόδοση, οπότε είναι προτιμότερο να γίνεται έλεγχος για εισαγωγή ενός αρχείου κάθε φορά, του αρχείου δηλαδή που έχει ανοίξει ο χρήστης εκείνη τη χρονική στιγμή. Εάν όλα τα αρχεία των φωτογραφιών και βίντεο ή όλα τα αρχεία ενός άλμπουμ αποθηκεύονταν μαζικά στη βάση δεδομένων της εφαρμογής κατά το άνοιγμα της συγκεκριμένης λίστας/άλμπουμ, θα έπρεπε κάθε φορά που ο χρήστης άνοιγε το συγκεκριμένο κατάλογο να γίνεται έλεγχος για όλα τα αρχεία που περιέχει, ώστε να εντοπιστούν και να αποθηκευτούν στη βάση δεδομένων νέα αρχεία που πιθανώς έχουν προστεθεί στη συσκευή. Κάτι τέτοιο θα είχε σημαντικές επιπτώσεις στην απόδοση του συστήματος και θα επηρέαζε αρνητικά την εμπειρία χρήσης της εφαρμογής, ειδικά σε συσκευές με μεγάλο πλήθος αρχείων.

3.2 ΠΑΡΟΥΣΙΑΣΗ/ΕΠΙΔΕΙΞΗ ΕΚΤΕΛΕΣΗΣ ΕΦΑΡΜΟΓΗΣ

Ακολουθεί η παρουσίαση της εφαρμογής στην τελική της μορφή. Θα παρουσιαστούν μέσω εικόνων και επεξηγήσεων, οι κυριότερες λειτουργίες και δυνατότητές της, από τη πλευρά ενός χρήστη. Για την εκτέλεση της εφαρμογής, χρησιμοποιείται ο προσομοιωτής (emulator) φορητών συσκευών που παρέχεται από το Android SDK.

3.2.1 Αρχική οθόνη



Κατά το άνοιγμα της εφαρμογής εμφανίζεται η αρχική της οθόνη. Περιλαμβάνει το αρχικό μενού, το οποίο αποτελείται από τέσσερα κουμπιά που οδηγούν σε τέσσερις βασικές δραστηριότητες της εφαρμογής. Πρόκειται για τις δραστηριότητες που προβάλλουν όλες τις φωτογραφίες, τα βίντεο ή τα άλμπουμ της συσκευής, καθώς και την δραστηριότητα της αναζήτησης. Όπως αναφέρθηκε και παραπάνω, στην αρχική οθόνη γίνεται ο έλεγχος για αλλαγές στα αρχεία της συσκευής, ώστε να ενημερωθεί κατάλληλα η βάση δεδομένων της εφαρμογής για αρχεία που έχουν διαγραφεί. Ο χρήστης δεν εμπλέκεται σε αυτή τη διαδικασία, καθώς συμβαίνει αυτόματα κάθε φορά που πλοηγείται στην αρχική οθόνη.

Εικόνα 7: Αρχική οθόνη

3.2.2 Προβολή φωτογραφίας

Ο χρήστης επιλέγει το κουμπί *Photos*. Εμφανίζονται στην οθόνη όλες οι φωτογραφίες της συσκευής ως μια λίστα σε μορφή πλέγματος.



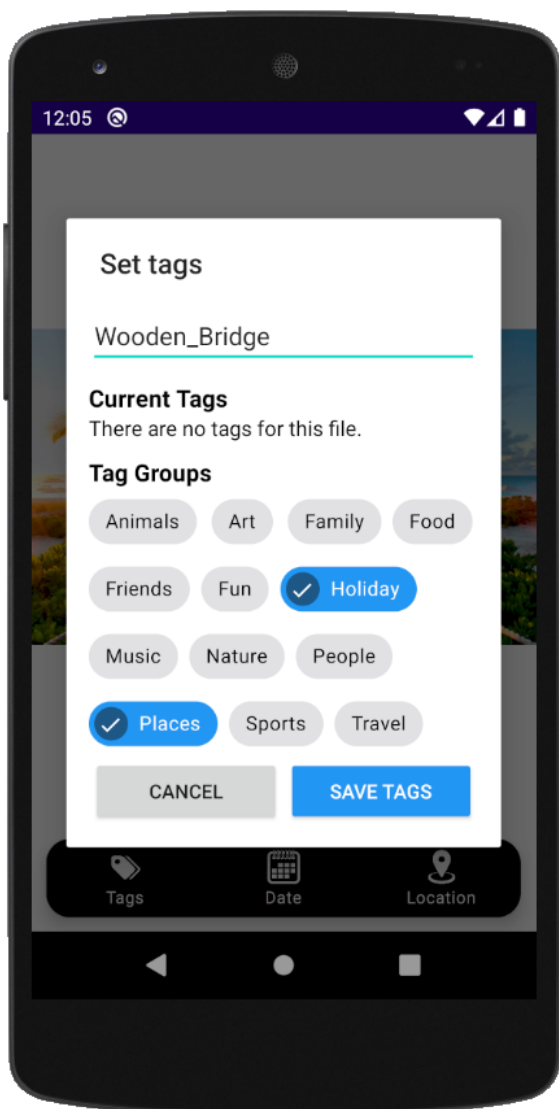
Εικόνα 8: Λίστα φωτογραφιών



Εικόνα 9: Προβολή αρχείου εικόνας

Στη συνέχεια επιλέγει μια φωτογραφία για άνοιγμα και προβολή. Η φωτογραφία που επέλεξε εμφανίζεται στην οθόνη με μια μπάρα πλοήγησης στο κάτω μέρος της.

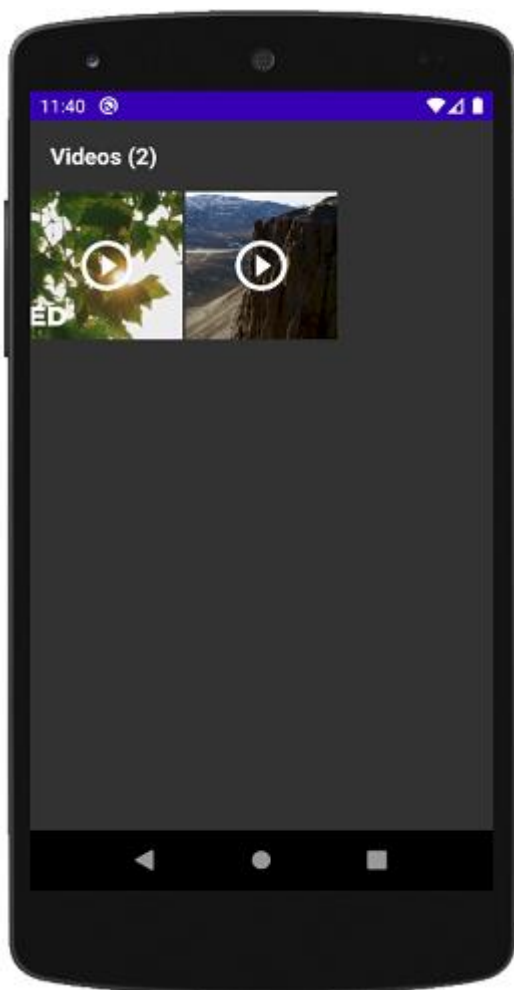
3.2.3 Εισαγωγή ετικετών



Από το μενού της μπάρας πλοήγησης επιλέγει "Tags" για να εισάγει ετικέτες στη φωτογραφία. Στο συγκεκριμένο αρχείο δεν έχουν προστεθεί ακόμα ετικέτες. Ανοίγει ένα παράθυρο διαλόγου στο οποίο ο χρήστης επιλέγει για εισαγωγή κάποιες από τις προκαθορισμένες ετικέτες της εφαρμογής και εισάγει από το πληκτρολόγιο μια ετικέτα της επιλογής του. Στη συνέχεια επιλέγει το κουμπί Save Tags και οι ετικέτες αποθηκεύονται στο αρχείο.

Εικόνα 10: Εισαγωγή ετικετών σε αρχείο

Εάν ο χρήστης ανοίξει ξανά το παράθυρο των ετικετών, στο πεδίο "Current Tags" θα φαίνονται οι ετικέτες που έχει ορίσει στο συγκεκριμένο αρχείο. Επίσης, έχει τη δυνατότητα να προσθέσει καινούριες ή να αφαιρέσει τις ήδη υπάρχουσες.



Εικόνα 11: Λίστα αρχείων βίντεο

Το βίντεο εμφανίζεται στην οθόνη με μια μπάρα πλοήγησης στο κάτω μέρος της. Ο χρήστης έχει τη δυνατότητα να πατήσει το κουμπί της αναπαραγωγής, για την προβολή του βίντεο μέσω των διαθέσιμων εφαρμογών που υποστηρίζουν αναπαραγωγή πολυμέσων.

3.2.4 Προβολή βίντεο

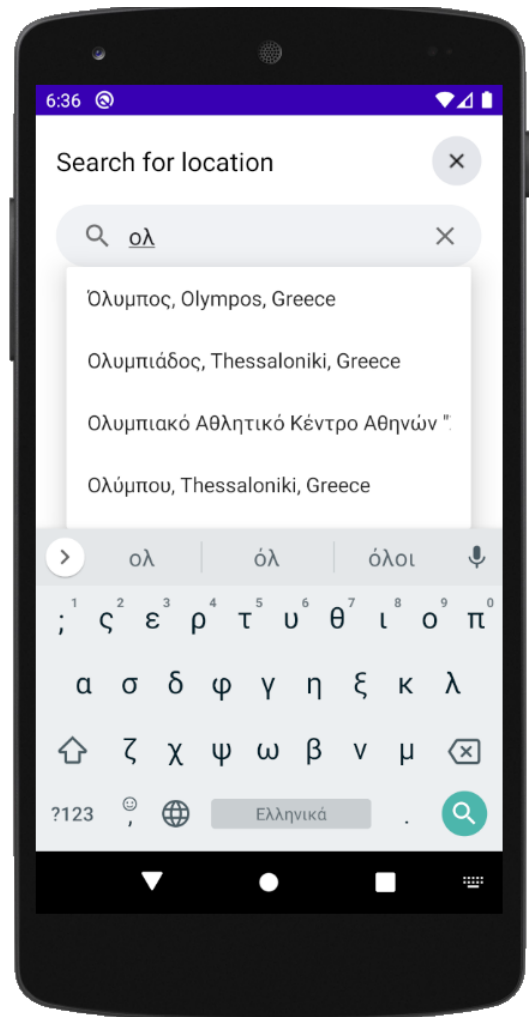
Ο χρήστης επιστρέφει στην αρχική οθόνη και επιλέγει το κουμπί *Videos*. Εμφανίζονται στην οθόνη όλα τα αρχεία βίντεο της συσκευής ως μια λίστα σε μορφή πλέγματος. Στη συνέχεια επιλέγει ένα από τα βίντεο της λίστας για άνοιγμα.



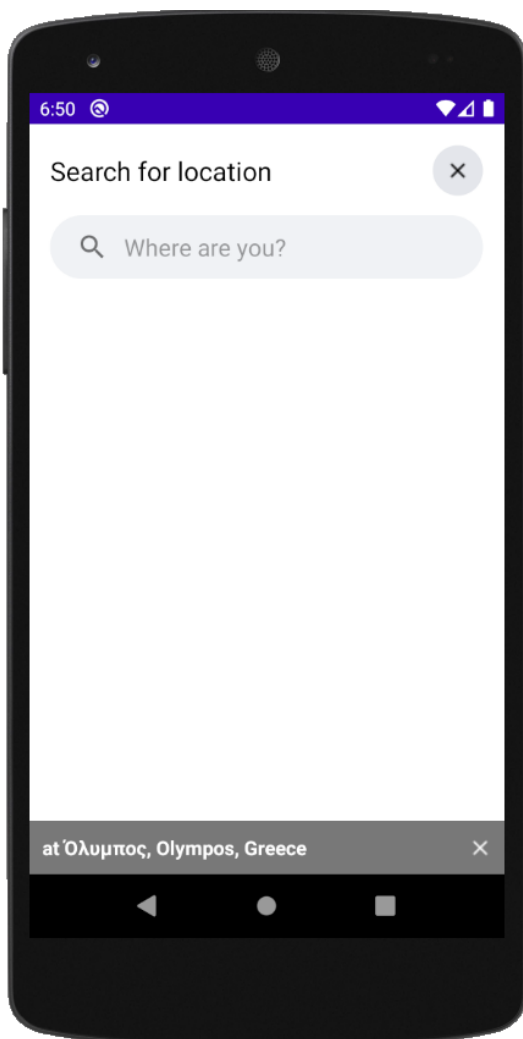
Εικόνα 12: Προβολή αρχείου βίντεο

3.2.5 Εισαγωγή τοποθεσίας

Από το μενού της μπάρας πλοήγησης επιλέγει "Location" για να εισάγει μία τοποθεσία στο βίντεο. Ανοίγει ένα παράθυρο διαλόγου στο οποίο ο χρήστης ξεκινά να πληκτρολογεί στο πεδίο αναζήτησης την τοποθεσία που επιθυμεί να ορίσει. Καθώς ο χρήστης πληκτρολογεί, εμφανίζονται προτάσεις αυτόματης συμπλήρωσης σχετικές με το κείμενο του χρήστη, ώστε να τον βοηθήσουν να επιλέξει ευκολότερα την τοποθεσία που επιθυμεί. Ο χρήστης επιλέγει μία από τις διαθέσιμες προτάσεις της λίστας η οποία αποθηκεύεται αυτομάτως ως τοποθεσία για το συγκεκριμένο αρχείο.

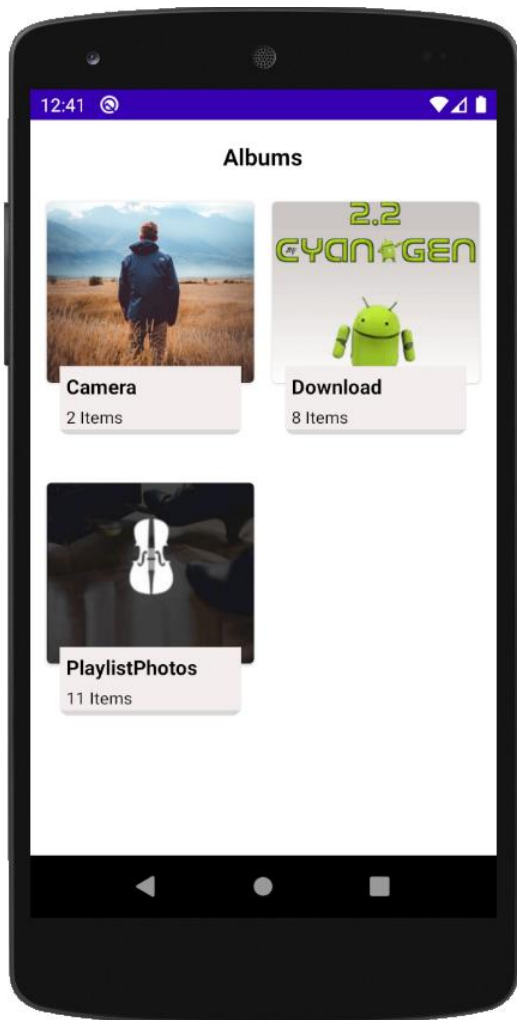


Εικόνα 13: Εισαγωγή τοποθεσίας σε αρχείο



Εικόνα 14: Παράθυρο διαλόγου τοποθεσίας

Ο χρήστης ανοίγοντας ξανά το παράθυρο της τοποθεσίας, βλέπει την τρέχουσα τοποθεσία που έχει ορίσει και έχει τη δυνατότητα να την διαγράψει ή να εισάγει μια καινούρια.

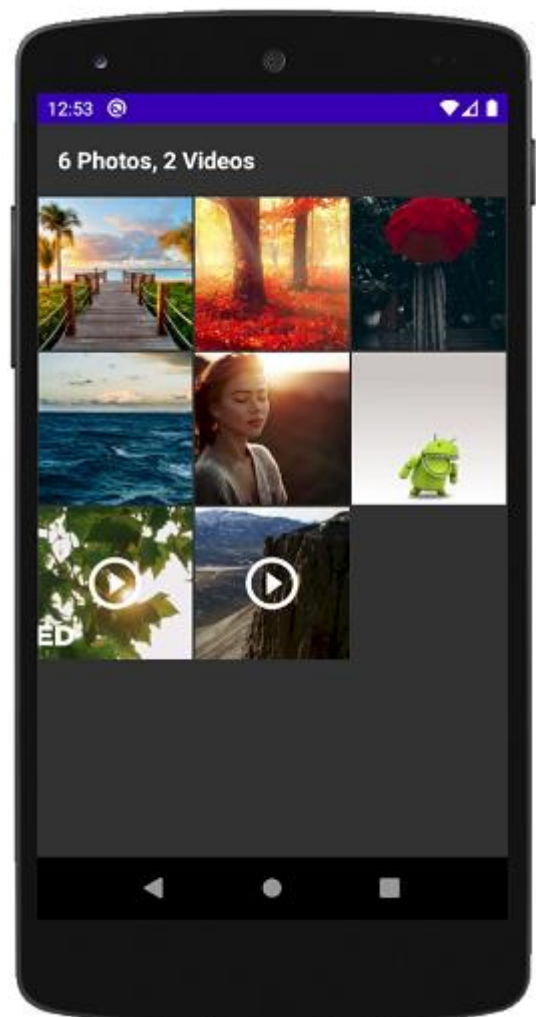


Εικόνα 15: Συλλογή άλμπουμ

3.2.6 Προβολή άλμπουμ

Ο χρήστης επιστρέφει στην αρχική οθόνη και επιλέγει το κουμπί *Albums*. Εμφανίζονται στην οθόνη όλοι οι κατάλογοι/άλμπουμ του αποθηκευτικού χώρου της συσκευής που περιέχουν κάποιο αρχείο εικόνας ή βίντεο. Στη συνέχεια επιλέγει το άλμπουμ *Download* από τη συλλογή του.

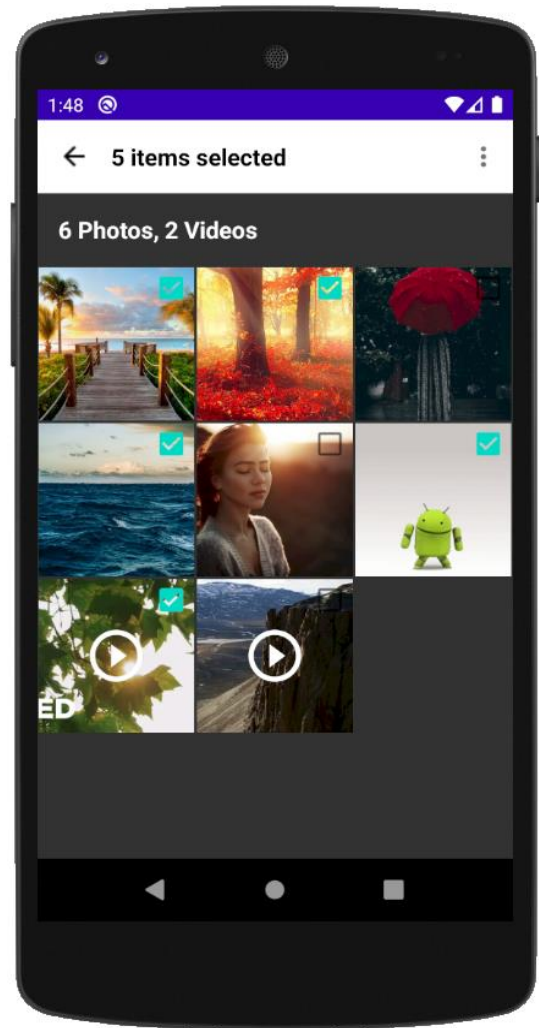
Ο χρήστης κατευθύνεται σε μία νέα διάταξη στην οποία εμφανίζονται στην οθόνη όλα τα αρχεία φωτογραφιών και βίντεο που περιέχει το συγκεκριμένο άλμπουμ/κατάλογος. Ο χρήστης μπορεί να ανοίξει οποιοδήποτε από τα αρχεία της λίστας για προβολή και να επεξεργαστεί τα δεδομένα τους (ετικέτες, τοποθεσία, ημερομηνία).



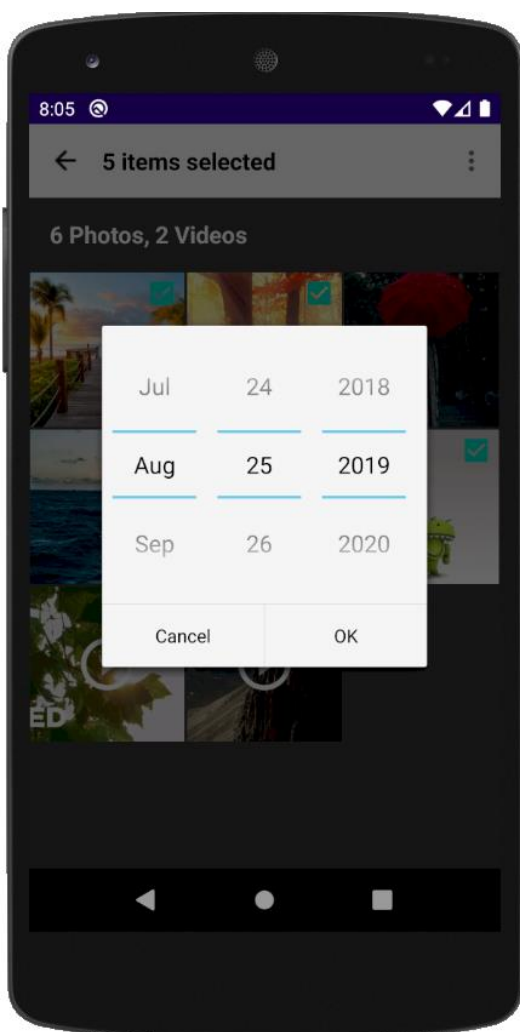
Εικόνα 16: Λίστα αρχείων άλμπουμ

3.2.7 Μαζική εισαγωγή ημερομηνίας σε πολλαπλά αρχεία

Ο χρήστης αγγίζει παρατεταμένα ένα από τα αρχεία της λίστας. Η λίστα μπαίνει σε "selection mode", δίνοντας τη δυνατότητα στο χρήστη να επιλέξει πολλαπλά αρχεία για μαζική επεξεργασία. Ο χρήστης διαλέγει από τη λίστα τα αρχεία που επιθυμεί επιλέγοντας τα σχετικά checkboxes. Αφού ο χρήστης έχει ολοκληρώσει την επιλογή των αρχείων που επιθυμεί, ανοίγει το μενού που βρίσκεται στο πάνω δεξιά μέρος της οθόνης και επιλέγει "Date".



Εικόνα 17: Επιλογή πολλαπλών αρχείων

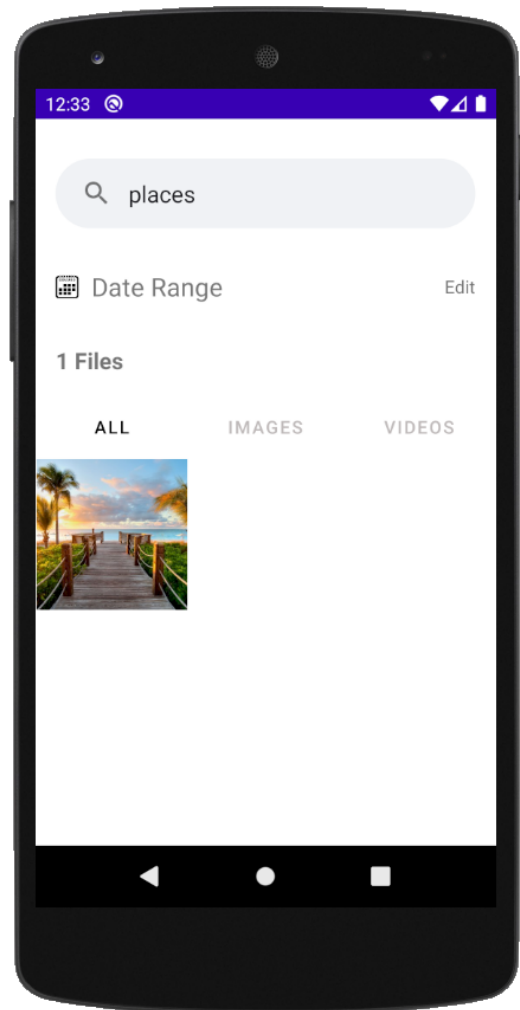


Εικόνα 18: Εισαγωγή ημερομηνίας

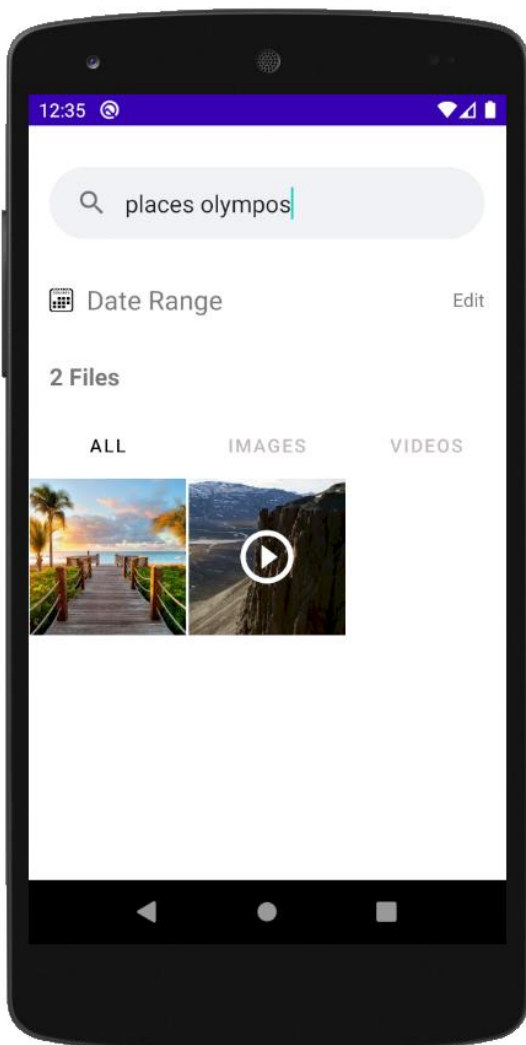
Στην οθόνη εμφανίζεται ένα παράθυρο διαλόγου που περιέχει ένα ημερολόγιο στο οποίο ο χρήστης μπορεί να ορίσει μια νέα ημερομηνία για τα αρχεία που έχει επιλέξει. Ο χρήστης εισάγει την ημερομηνία που επιθυμεί και στη συνέχεια πατάει το κουμπί OK. Αν η νέα ημερομηνία οριστεί επιτυχώς στα επιλεγμένα αρχεία, τότε το παράθυρο διαλόγου κλείνει και εμφανίζεται ένα σχετικό μήνυμα στην οθόνη για την επιτυχημένη αποθήκευση της ημερομηνίας στα συγκεκριμένα αρχεία.

3.2.8 Αναζήτηση αρχείων

Ο χρήστης επιστρέφει στην αρχική οθόνη και επιλέγει το κουμπί *Search*. Στη συνέχεια, στο πεδίο αναζήτησης που εμφανίζεται στη διάταξη της οθόνης, πληκτρολογεί μία από τις ετικέτες που πρόσθεσε νωρίτερα σε ένα αρχείο φωτογραφίας, συγκεκριμένα την ετικέτα "Places". Στην οθόνη εμφανίζεται η φωτογραφία που περιέχει τη συγκεκριμένη ετικέτα.



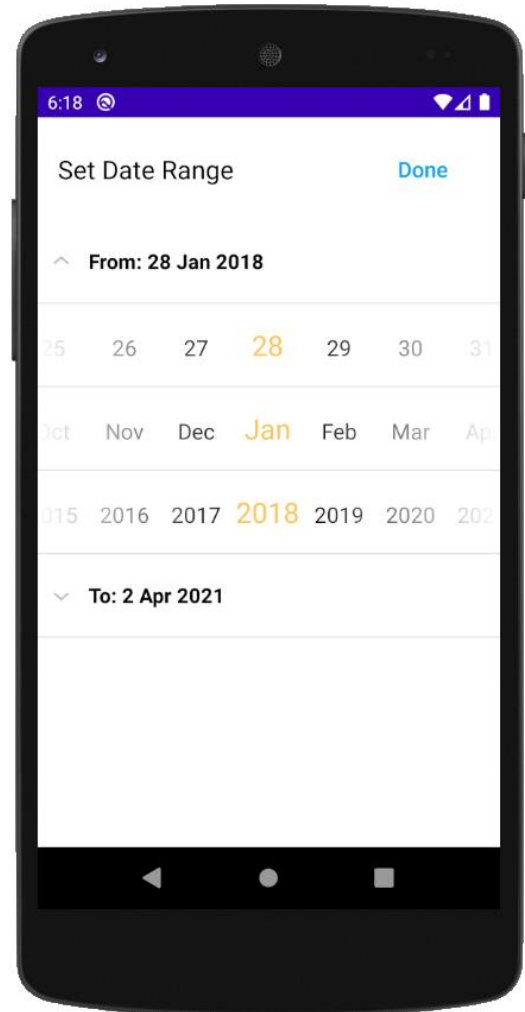
Εικόνα 19: Αναζήτηση αρχείων (1)



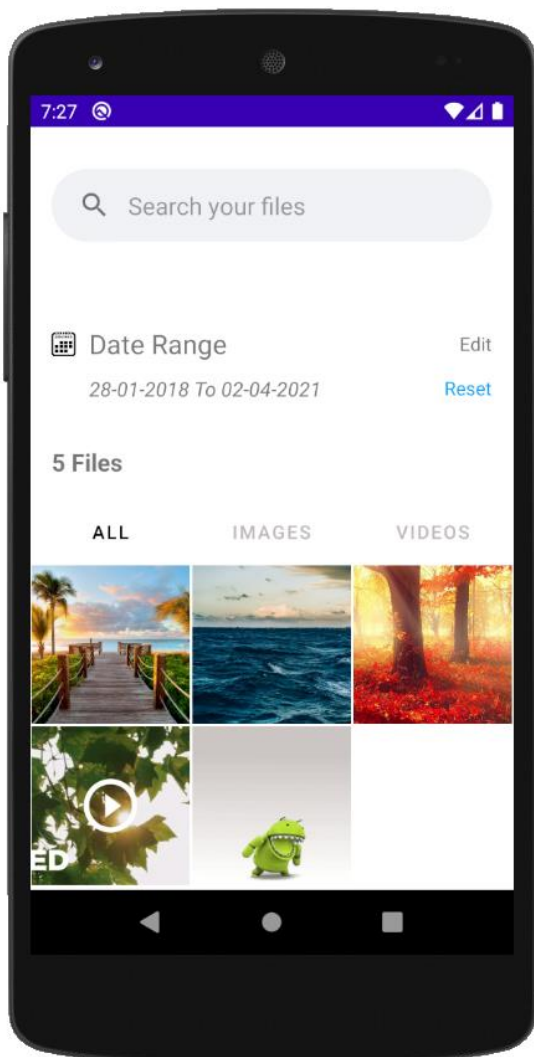
Εικόνα 20: Αναζήτηση αρχείων (2)

Στη συνέχεια ο χρήστης πληκτρολογεί μία τοποθεσία που όρισε νωρίτερα σε ένα αρχείο βίντεο. Στα αποτελέσματα της αναζήτησης εμφανίζεται και το αρχείο που περιέχει την συγκεκριμένη τοποθεσία. Ο χρήστης έχει τη δυνατότητα να αναζητήσει αρχεία εισάγοντας μια ή περισσότερες λέξεις, χωρισμένες με κάποιον κενό χαρακτήρα. Στη συγκεκριμένη περίπτωση ο χρήστης αναζήτησε αρχεία μέσω των ετικετών και της τοποθεσίας τους, εισάγοντας δυο λέξεις στο πεδίο αναζήτησης. Τα αποτελέσματα εμφανίζονται και ανανεώνονται αυτόματα σύμφωνα με την είσοδο του χρήστη στο πεδίο αναζήτησης καθώς πληκτρολογεί.

Ο χρήστης επιλέγει το κουμπί *Edit* προκειμένου να ορίσει ένα εύρος ημερομηνίας για τα επιστρεφόμενα αποτελέσματα. Στην οθόνη εμφανίζεται ένα νέο παράθυρο διαλόγου στο οποίο ο χρήστης ορίζει μια αρχική και μια τελική ημερομηνία, η οποία θα εφαρμοστεί ως φίλτρο στα επιστρεφόμενα αποτελέσματα. Τέλος, επιλέγει το κουμπί *Done* και η εφαρμογή επιστρέφει στην οθόνη με τη διάταξη της αναζήτησης.

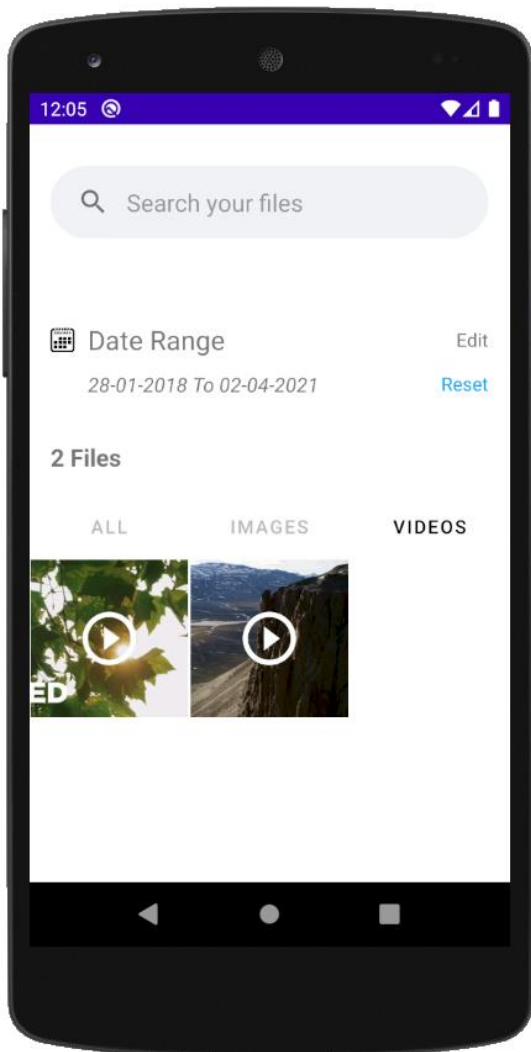


Εικόνα 21: Ορισμός φίλτρου ημερομηνίας



Εικόνα 22: Αναζήτηση αρχείων (3)

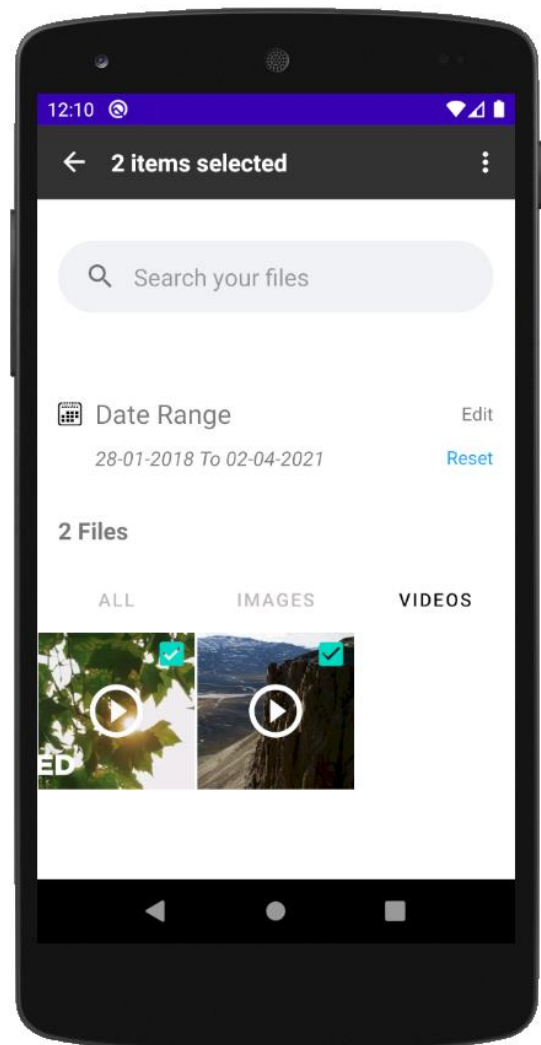
Εάν ο χρήστης έχει ορίσει ένα έγκυρο εύρος ημερομηνιών, τότε στα αποτελέσματα της αναζήτησης εμφανίζονται τα αρχεία που ικανοποιούν το συγκεκριμένο ημερολογιακό κριτήριο. Όσο είναι ενεργό το φίλτρο ημερομηνίας, οι αναζητήσεις που θα κάνει ο χρήστης πληκτρολογώντας στο πεδίο αναζήτησης, θα επιστρέφουν ως αποτελέσματα, αρχεία που περιέχουν το κείμενο αναζήτησης του χρήστη στις ετικέτες ή στην τοποθεσία τους και παράλληλα ικανοποιούν το εύρος ημερομηνίας που έχει οριστεί. Ο χρήστης μπορεί να ακυρώσει το εύρος ημερομηνίας που έχει ορίσει για τα αποτελέσματα, πατώντας το κουμπί "Reset" που εμφανίζεται πάνω από τη λίστα των αποτελεσμάτων της αναζήτησης.



Εικόνα 23: Αναζήτηση αρχείων (4)

Τέλος, ο χρήστης μπορεί να επιλέξει οποιαδήποτε φωτογραφία ή βίντεο από τα αποτελέσματα για προβολή καθώς και να επιλέξει ένα ή περισσότερα αρχεία από τη λίστα των αποτελεσμάτων για επεξεργασία (εισαγωγή ετικετών, ημερομηνίας, τοποθεσίας).

Τα αποτελέσματα της αναζήτησης ταξινομούνται σε τρεις κατηγορίες, "All", "Images" και "Videos". Αρχικά το σύστημα προβάλλει στο χρήστη την καρτέλα "All" και στη συνέχεια εκείνος έχει τη δυνατότητα να περιηγηθεί στις καρτέλες "Images" και "Videos" εάν θέλει να φιλτράρει τα αποτελέσματα της αναζήτησης με βάση τον τύπο των αρχείων.



Εικόνα 24: Αναζήτηση αρχείων (5)

4. ΑΞΙΟΛΟΓΗΣΗ ΠΑΡΟΜΟΙΩΝ ΕΦΑΡΜΟΓΩΝ

Η λήψη φωτογραφιών και βίντεο αποτελεί μία από τις βασικότερες ενασχολήσεις των χρηστών μίας φορητής συσκευής με κάμερα. Σε αυτό έχουν συντελέσει σε σημαντικό βαθμό οι κάμερες υψηλής ανάλυσης και οι μεγάλοι αποθηκευτικοί χώροι που διακρίνουν τις φορητές συσκευές σήμερα, καθώς επίσης και η ραγδαία εξάπλωση των κοινωνικών δικτύων όπως Instagram, Facebook, Twitter στα οποία καθημερινά οι χρήστες δημοσιεύουν εκατομμύρια φωτογραφίες. Αυτό έχει ως αποτέλεσμα τη δημιουργία και την ανάπτυξη εφαρμογών που σχετίζονται με την διαχείριση, την ταξινόμηση αλλά και την επεξεργασία φωτογραφιών και βίντεο.

Σε αυτή την ενότητα θα παρουσιαστούν ορισμένες από τις δημοφιλέστερες Android εφαρμογές οργάνωσης και διαχείρισης φωτογραφιών που είναι διαθέσιμες για χρήση. Θα μελετηθούν οι λειτουργίες και οι υπηρεσίες που προσφέρουν και θα αξιολογηθούν συγκριτικά με την αντίστοιχη εφαρμογή που αναπτύχθηκε στην παρούσα εργασία.

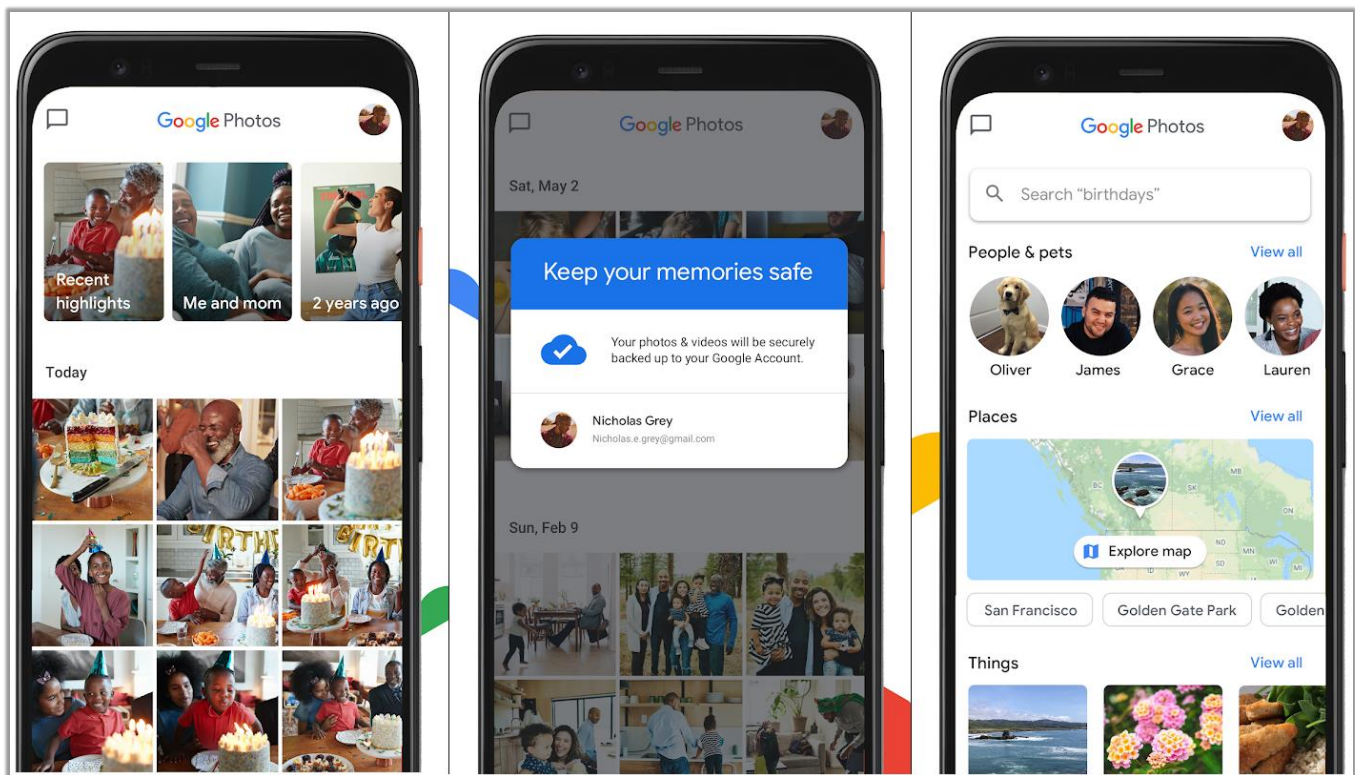
4.1 GOOGLE PHOTOS

Το Google Photos είναι η δημοφιλέστερη επιλογή στην συγκεκριμένη κατηγορία με περισσότερες από 1 δισεκατομμύριο λήψεις σε Android συσκευές και εξαιρετικές κριτικές. Διαθέτει πολλές λειτουργίες που το κάνουν να ξεχωρίζει όπως απεριόριστο αποθηκευτικό χώρο σε cloud για φωτογραφίες μέχρι 16 Megapixel και βίντεο μέχρι 1080p, δυνατότητα αυτόματης δημιουργίας αντιγράφων, πολλαπλή επιλογή φωτογραφιών για επεξεργασία και ανίχνευση διπλότυπων φωτογραφιών.

Αποτελεί ένα αξιόλογο εργαλείο οργάνωσης των φωτογραφιών και των βίντεο καθώς χρησιμοποιεί έναν πλήρως λειτουργικό αλγόριθμο αναγνώρισης εικόνων, ώστε να ορίζει κατάλληλες ετικέτες σε κάθε αρχείο κάνοντας την αναζήτηση τους ευκολότερη. Ο αλγόριθμος έχει τη δυνατότητα να εντοπίζει όχι μόνο πρόσωπα αλλά και αντικείμενα και μέρη. Αυτό επιτρέπει στους χρήστες να βρίσκουν άμεσα τα αρχεία που αναζητούν χωρίς να χρειάζεται να ψάχνουν σε φακέλους που μπορεί να περιέχουν μεγάλο πλήθος φωτογραφιών και βίντεο.

Επίσης παρέχει λειτουργίες κοινοποίησης των φωτογραφιών σε κοινωνικά δίκτυα και άλλες πλατφόρμες. Προσφέρει εύκολο συγχρονισμό των αρχείων μεταξύ διαφορετικών συσκευών που χρησιμοποιούν την εφαρμογή και έχουν κάνει χρήση της cloud υπηρεσίας που διαθέτει. Επιπλέον ταξινομεί τις φωτογραφίες χρονολογικά και παρέχει προτάσεις σχετικά με αρχεία που θα μπορούσαν να αρχειοθετηθούν στο cloud για μεγαλύτερη εξοικονόμηση χώρου της συσκευής.

Η εφαρμογή Google Photos δίνει τη δυνατότητα στο χρήστη να δημιουργήσει δικά του άλμπουμ και συλλογές καθώς και μικρά slideshows που αποτελούνται από κολλάζ φωτογραφιών και μικρά αποσπάσματα από βίντεο. Τέλος προσφέρει λειτουργίες επεξεργασίας των φωτογραφιών όπως ρύθμιση φωτεινότητας, περικοπή εικόνας και προσθήκη εφέ. Το σημαντικότερο είναι πως η συγκεκριμένη εφαρμογή παρέχεται δωρεάν για χρήση σε όλες τις φορητές συσκευές Android.



Εικόνα 25: Google Photos [Source:

<https://play.google.com/store/apps/details?id=com.google.android.apps.photos&hl=en&gl=US>]

Αδιαμφισβήτητα το δυνατό σημείο του Google Photos είναι ο αποθηκευτικός χώρος που παρέχει ως backup, μέσω της cloud υπηρεσίας του εξοικονομώντας χώρο από τη συσκευή του χρήστη. Σε συνδυασμό με τις δυνατότητες κοινής χρήσης και συγχρονισμού των αρχείων για ευκολότερη πρόσβαση μεταξύ διαφορετικών συσκευών κάνουν τη συγκεκριμένη εφαρμογή να ξεχωρίζει από τον ανταγωνισμό και αυτό δικαιολογείται και από τον αριθμό λήψεων και καλών κριτικών που έχει λάβει.

Συγκριτικά με την εφαρμογή της παρούσας εργασίας, το σημείο στο οποίο υστερεί ελαφρώς το Google Photos είναι στην επεξεργασία των μεταδεδομένων των αρχείων (ετικέτες, τοποθεσία κ.α.). Σχετικά με τις ετικέτες δε υπάρχει η δυνατότητα στο χρήστη να ορίσει τις δικές του, καθώς η εφαρμογή κατηγοριοποιεί και ορίζει tags αυτόματα μέσω ενός αλγορίθμου τεχνητής νοημοσύνης που διαθέτει, τα οποία πολλές φορές πιθανώς να μην είναι ιδιαίτερα χρήσιμα στην οργάνωση και ταξινόμηση των αρχείων. Όσον αφορά την τοποθεσία, δεν υποστηρίζεται κάποια λειτουργία στην Android έκδοση της εφαρμογής, που να επιτρέπει στο χρήστη να ορίσει μόνος του μέρη και τοποθεσίες σε αρχεία. Ο μόνος τρόπος είναι να ενεργοποιήσει τις υπηρεσίες GPS της συσκευής του, ώστε να ορίζεται αυτόματα μια τοποθεσία κάθε φορά που βγάζει μια νέα φωτογραφία ή βίντεο με την κάμερα της συσκευής.

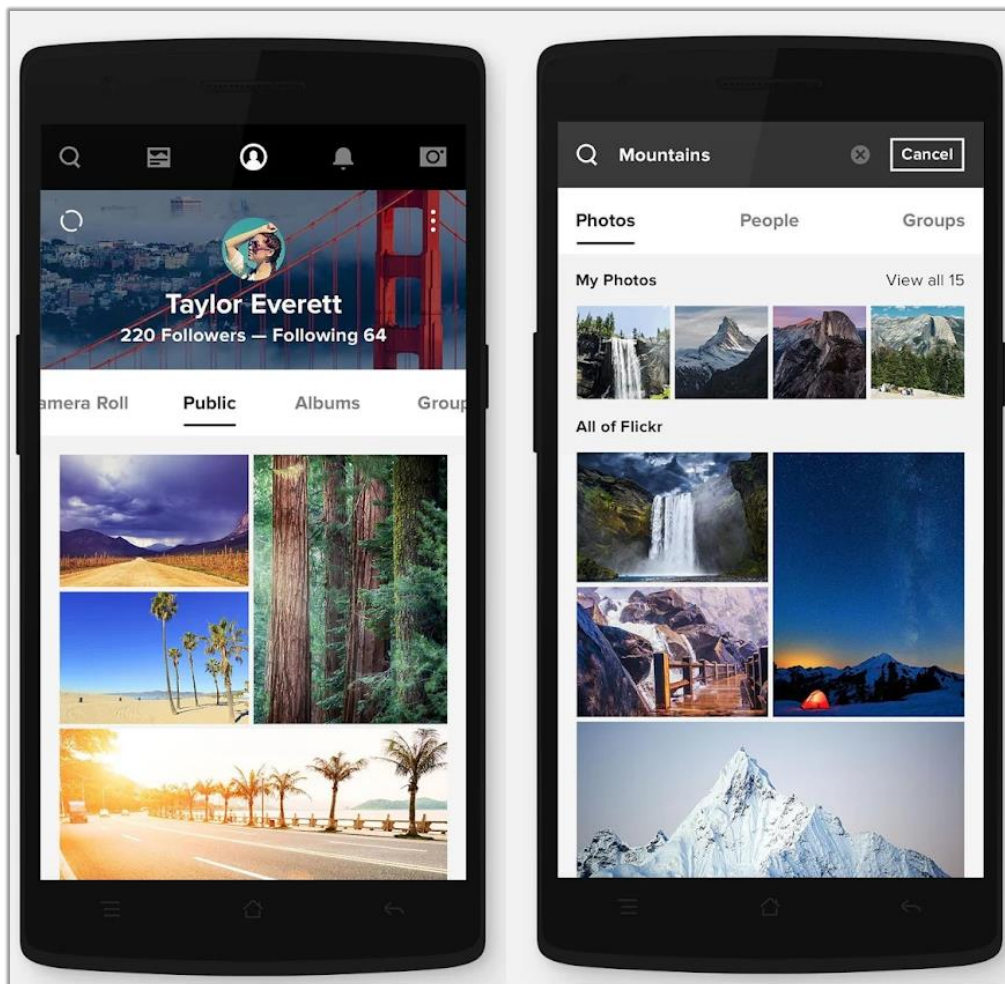
4.2 FLICKR

Το Flickr είναι μία αρκετά δημοφιλής υπηρεσία για την αποθήκευση και διαχείριση φωτογραφιών και βίντεο. Ουσιαστικά αποτελεί μια εφαρμογή οργάνωσης και αποθήκευσης αρχείων εικόνας και βίντεο με χαρακτηριστικά κοινωνικού δικτύου, καθώς οι χρήστες έχουν τη δυνατότητα να κοινοποιήσουν τις φωτογραφίες ή τα βίντεο τους δημόσια εάν το επιθυμούν ώστε να είναι ορατά και σε άλλους χρήστες της εφαρμογής. Η εφαρμογή προσφέρει ένα 1TB online αποθηκευτικού χώρου και απαιτεί έναν λογαριασμό Yahoo για την χρήση της. Προσφέρεται δωρεάν για χρήση, αλλά διαθέτει και PRO έκδοση χωρίς διαφημίσεις, η οποία είναι διαθέσιμη με μηνιαία συνδρομή και

προσφέρει κάποιες επιπλέον λειτουργίες καθώς και απεριόριστο αποθηκευτικό χώρο. Η εφαρμογή παρέχεται για χρήση σε Android φορητές συσκευές αλλά και σε υπολογιστές μέσω της ιστοσελίδας της.

Οι φωτογραφίες και τα βίντεο των χρηστών ταξινομούνται σε χρονολογική σειρά βασισμένη στην ημερομηνία αποθήκευσης τους στην εφαρμογή. Για την αναζήτηση των αρχείων, το Flickr διαθέτει μηχανισμούς που προβάλλουν τις φωτογραφίες και τα βίντεο με βάση το θέμα. Επιπλέον, στην έκδοση της εφαρμογής για υπολογιστές, η αναζήτηση των αρχείων προσφέρει και μια πληθώρα επιλογών από φίλτρα για αποδοτικότερη κατηγοριοποίηση των αποτελεσμάτων, όπως αναζήτηση με βάση το χρώμα, το μέγεθος, τον προσανατολισμό (οριζόντιο ή κατακόρυφο) και τον τύπο των αρχείων (φωτογραφίες/βίντεο). Στην αποτελέσματα της αναζήτησης εμφανίζονται και δημόσιες φωτογραφίες από άλλους χρήστες σχετικές με το κείμενο αναζήτησης.

Το Flickr όπως αναφέρθηκε ήδη, διαθέτει κάποια χαρακτηριστικά κοινωνικών δικτύων. Οι χρήστες μπορούν να αλληλοεπιδρούν μεταξύ τους, μοιράζοντας δημόσια τις συλλογές φωτογραφιών και βίντεο που διαθέτουν, καθώς και να δουν στατιστικά στοιχεία για αυτές (προβολές, κοινοποιήσεις, σχόλια κ.α.). Επίσης ο χρήστης μπορεί να επεξεργαστεί τον λογαριασμό του, ορίζοντας μια φωτογραφία προφίλ και εξωφύλλου.



Εικόνα 26: Flickr [Source: <https://play.google.com/store/apps/details?id=com.flickr.android>]

Για την αποδοτικότερη οργάνωση των φωτογραφιών και βίντεο υπάρχει η δυνατότητα δημιουργίας άλμπουμ και συλλογών στα οποία ορίζεται όνομα και περιγραφή. Επιπλέον οι χρήστες μπορούν να

εισάγουν ετικέτες και σε μεμονωμένα αρχεία φωτογραφιών ή βίντεο ώστε να είναι ευκολότερο να τα αναζητήσουν στο μέλλον. Επίσης το κάθε αρχείο διαθέτει μεταδεδομένα τοποθεσίας τα οποία εισάγονται αυτόματα εάν είναι ενεργοποιημένη η υπηρεσία GPS στη συσκευή του χρήστη, διαφορετικά μπορεί να ορίσει ο ίδιος μια τοποθεσία χειροκίνητα. Επίσης, το Flickr υποστηρίζει την επιλογή αυτόματης δημιουργίας αντιγράφων ασφαλείας των φωτογραφιών και βίντεο που αποθηκεύονται στις συσκευές των χρηστών. Τέλος διαθέτει και ορισμένες λειτουργίες επεξεργασίας των εικόνων μέσω φίλτρων και άλλων εργαλείων.

Συνοψίζοντας το Flickr είναι μια εφαρμογή αποθήκευσης και οργάνωσης φωτογραφιών με αρκετές δυνατότητες και λειτουργίες. Συγκριτικά με την εφαρμογή της παρούσας εργασίας, το Flickr εστιάζει αρκετά σε χαρακτηριστικά και λειτουργίες κοινωνικών δικτύων σε αντίθεση με την εφαρμογή της εργασίας, που στοχεύει στην ταξινόμηση και στην οργάνωση των αρχείων. Η Android εφαρμογή του Flickr έχει σχετικά περιορισμένες δυνατότητες στο κομμάτι της αναζήτησης των αρχείων, καθώς οι προχωρημένες επιλογές φίλτρων αναζήτησης διατίθενται μόνο στην Desktop version της εφαρμογής. Επιπλέον το Flickr εμφανίζει κατά την αναζήτηση αρχείων, φωτογραφίες και βίντεο άλλων χρηστών δημιουργώντας συχνά μεγάλο όγκο αποτελεσμάτων που είναι δύσκολο να διαχειριστούν και να ταξινομηθούν από το χρήστη.

Επίσης, απαιτεί το ανέβασμα των αρχείων από τη συσκευή στον online αποθηκευτικό του χώρο προκειμένου αυτά να είναι διαθέσιμα για χρήση, σε αντίθεση με την εφαρμογή της παρούσας εργασίας που φορτώνει και προβάλλει τα αρχεία κατευθείαν από τον αποθηκευτικό χώρο της συσκευής. Ένα ακόμα μειονέκτημα του Flickr είναι πως ενώ υποστηρίζει το ανέβασμα αρχείων βίντεο, η αναπαραγωγή τους περιορίζεται μόνο στα τρία πρώτα λεπτά του βίντεο για τα δωρεάν μέλη και στα δέκα λεπτά για την PRO έκδοση. Τα παραπάνω μειονεκτήματα σε συνδυασμό με τις διαφημίσεις που περιέχει η δωρεάν έκδοση του Flickr, έχουν ως αποτέλεσμα μια ελαφρώς κουραστική εμπειρία χρήσης συγκριτικά με την εφαρμογή της παρούσας εργασίας.

4.3 Μυλιο

Το Mylio είναι μια εφαρμογή η οποία χρησιμοποιείται για την οργάνωση φωτογραφιών, βίντεο και άλλων αρχείων. Η συγκεκριμένη εφαρμογή υποστηρίζει τη χρήση σε όλες τις φορητές συσκευές (Android, iOS), αλλά και σε λειτουργικά συστήματα υπολογιστών (Windows, Mac). Διατηρεί τις φωτογραφίες διαφορετικών συσκευών σε μία βιβλιοθήκη ώστε να είναι όλες προσβάσιμες ανά πάσα στιγμή ακόμα και χωρίς τη χρήση ίντερνετ. Επίσης επιτρέπει την διατήρηση της δομής των αρχείων της συσκευής, δηλαδή σε περίπτωση που ο χρήστης έχει ήδη ταξινομήσει σε κάποιους υποφακέλους ή συλλογές τα αρχεία του, η δομή παραμένει ίδια και στην εφαρμογή. Τυχόν αλλαγές που γίνονται στα αρχεία κατά τη χρήση της εφαρμογής συγχρονίζονται σε όλες τις συσκευές εξοικονομώντας χρόνο. Προσφέρει μια καλοσχεδιασμένη και εύχρηστη διεπαφή με αρκετά εργαλεία και χαρακτηριστικά για την αποδοτικότερη οργάνωση εικόνων και βίντεο.

Ένα χρήσιμο εργαλείο οργάνωσης και ταξινόμησης του Mylio είναι το Life Calendar. Η συγκεκριμένη λειτουργία οργανώνει τα αρχεία με χρονολογική σειρά χρησιμοποιώντας την εφαρμογή του ημερολογίου και κατηγοριοποιώντας τα με βάση την ημερομηνία και τα γεγονότα που λαμβάνουν χώρα τη συγκεκριμένη μέρα, καθώς συγχρονίζεται με το ημερολόγιο. Επίσης το Mylio δίνει τη δυνατότητα στους χρήστες να ταξινομήσουν τα αρχεία τους με βάση την τοποθεσία. Αυτό γίνεται μέσω ενός χάρτη με τον οποίο οι χρήστες μπορούν εύκολα να ορίσουν μια τοποθεσία σε ένα ή περισσότερα αρχεία. Το Mylio ορίζει αυτόματα την τοποθεσία στο χάρτη για αρχεία που διαθέτουν μεταδεδομένα τοποθεσίας. Ο χάρτης μπορεί να χρησιμοποιηθεί και για αναζήτηση αρχείων, καθώς οι φωτογραφίες και τα βίντεο ομαδοποιούνται ανά τοποθεσία με μια καρφίτσα που δείχνει πόσες φωτογραφίες/βίντεο τραβήχτηκαν σε κάθε μέρος.

Ένα από τα αξιοσημείωτα χαρακτηριστικά του Mylio για την αποτελεσματικότερη οργάνωση των φωτογραφιών και βίντεο είναι η τεχνολογία αναγνώρισης προσώπων που διαθέτει. Με τη χρήση της συγκεκριμένης τεχνολογίας οι χρήστες έχουν τη δυνατότητα προσθήκης ετικετών στα άτομα που βρίσκονται στις φωτογραφίες τους. Μόλις ο χρήστης ορίσει ένα όνομα-ετικέτα για κάποιον, το σύστημα θα δημιουργήσει αυτόματα μια συλλογή με όλα τα αρχεία που θα εντοπίσει με το ίδιο άτομο. Πρόκειται για έναν αλγόριθμο τεχνητής νοημοσύνης, ο οποίος βελτιώνεται όσο οι χρήστες τον τροφοδοτούν με δεδομένα, δηλαδή ορίζοντας ετικέτες σε πρόσωπα ώστε να τα αναγνωρίζει ευκολότερα. Επιπλέον, εκτός από την προσθήκη ετικετών σε πρόσωπα, οι χρήστες μπορούν να βαθμολογήσουν με αστέρια (rating) τα αρχεία τους, να ορίσουν λέξεις-κλειδιά (keywords) και σημαίες (flags), ακόμα και να τα ταξινομήσουν με βάση τα χρώματα.

Η αναζήτηση αρχείων στο Mylio αποτελεί μια εύκολη διαδικασία με τη χρήση όλων των παραπάνω δυνατοτήτων οργάνωσης και ταξινόμησης. Οι χρήστες της εφαρμογής μπορούν να χρησιμοποιήσουν τη μπάρα αναζήτησης εισάγοντας λέξεις ή φράσεις κλειδιά σύμφωνα με τις ετικέτες, την τοποθεσία και τα άτομα που έχουν οριστεί στα αρχεία. Το Mylio διαθέτει και ένα πλήθος φίλτρων που μπορούν να εφαρμοσθούν στα αποτελέσματα της αναζήτησης ώστε να βρίσκουν οι χρήστες ευκολότερα τα αρχεία που αναζητούν. Τα συγκεκριμένα φίλτρα σχετίζονται με την ημερομηνία, την βαθμολογία, τα χρώματα και τον τύπο των αρχείων(φωτογραφίες, βίντεο, έγγραφα).



Εικόνα 27: Mylio [Source:

<https://play.google.com/store/apps/details?id=com.myliollc.mylio&hl=en>]

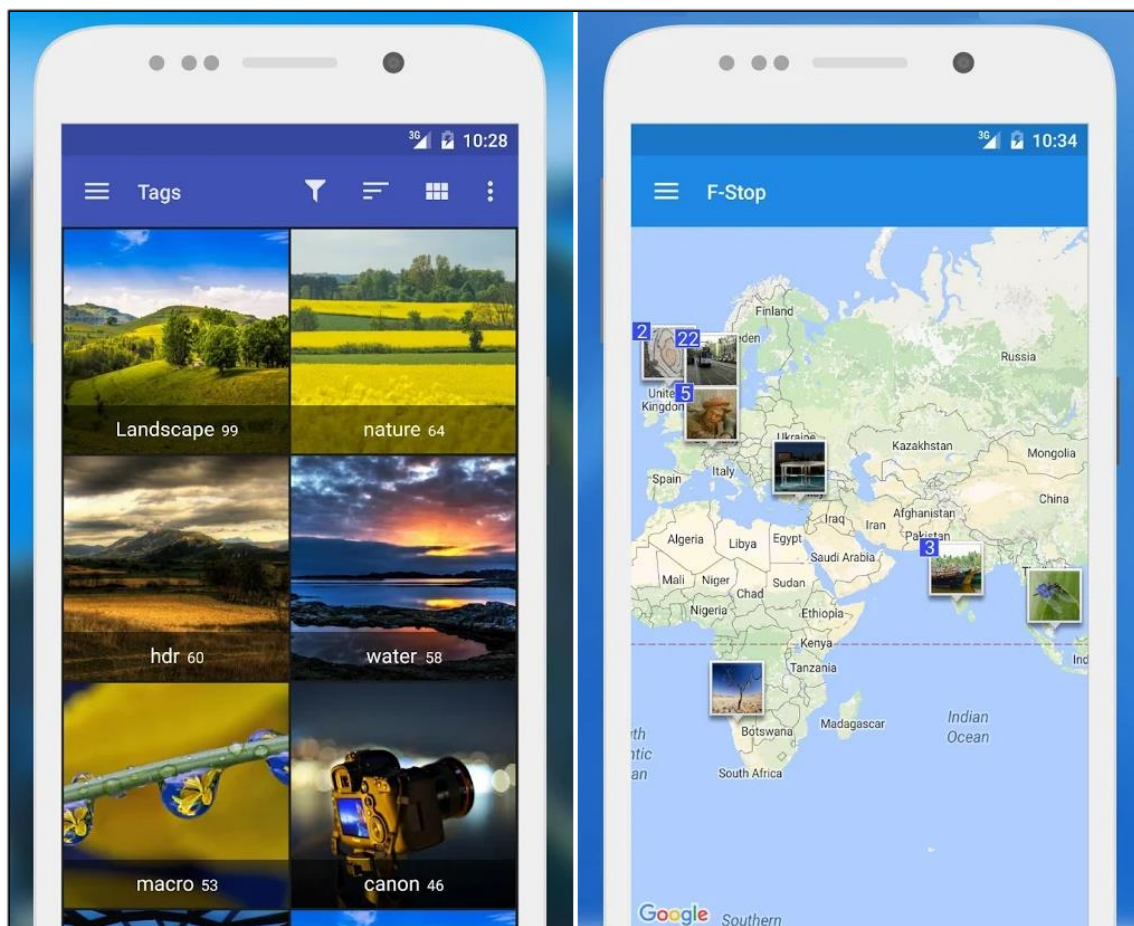
Το Mylio υποστηρίζει την κοινοποίηση των αρχείων και σε άλλες πλατφόρμες ή κοινωνικά δίκτυα όπως Facebook, email και Flickr. Η χρήση κάποιου cloud για online αποθήκευση των φωτογραφιών, δεν είναι υποχρεωτική καθώς το Mylio χρησιμοποιεί τον αποθηκευτικό χώρο της συσκευής που το χρησιμοποιεί για την αποθήκευση των αρχείων. Εάν κάποιος επιλέξει να χρησιμοποιήσει κάποια cloud υπηρεσία ως αποθηκευτικό μέσο, τότε τα αρχεία του κρυπτογραφούνται ώστε να παραμείνουν ασφαλή. Τέλος, το Mylio έχει κάποιες βασικές λειτουργίες επεξεργασίας εικόνων όπως η προσθήκη εφέ, φίλτρων, περικοπή και περιστροφή εικόνας κ.α. Η Android έκδοση της εφαρμογής διατίθεται δωρεάν για χρήση υποστηρίζοντας μέχρι τρεις διαφορετικές συσκευές και 25.000 αρχεία. Ωστόσο, υπάρχει και η δυνατότητα αναβάθμισης μέσω πληρωμής για απεριόριστες συσκευές και αρχεία.

4.4 F-STOP GALLERY

Άλλη μια Android εφαρμογή σχετικά με την οργάνωση και διαχείριση φωτογραφιών και βίντεο είναι το F-Stop Gallery. Πρόκειται για μια ελαφριά, απλή και ταυτόχρονα χρήσιμη εφαρμογή με πλήθος θετικών αξιολογήσεων. Απευθύνεται τόσο σε αρχάριους, όσο και σε επαγγελματίες χρήστες σχετικά με τη λήψη φωτογραφιών. Χρησιμοποιεί στοιχεία και πληροφορίες που υπάρχουν στα αρχεία για την αποδοτικότερη οργάνωση και ταξινόμησή τους. Η διεπαφή της εφαρμογής είναι εύχρηστη καθώς παρέχει πολλές επιλογές ρυθμίσεων σχετικά με την προβολή και την ταξινόμηση φακέλων και αρχείων.

Οι χρήστες μπορούν να οργανώσουν τα βίντεο και τις φωτογραφίες τους σε άλμπουμ και συλλογές, να προσθέσουν ετικέτες και να δημιουργήσουν υποφακέλους. Τα άλμπουμ και οι συλλογές μπορούν να ταξινομηθούν με βάση το όνομα, την ημερομηνία τροποποίησης και το πλήθος αρχείων που περιέχουν. Η εφαρμογή F-Stop Gallery προσφέρει στους χρήστες όλες τις βασικές λειτουργίες διαχείρισης αρχείων, όπως αντιγραφή, μετακίνηση και διαγραφή από φακέλους και άλμπουμ. Επίσης υπάρχει η δυνατότητα προσθήκης ετικετών και βαθμολογίας στα αρχεία καθώς και προσθήκης στα "Αγαπημένα" για αποδοτικότερη οργάνωση. Οι χρήστες μπορούν ακόμα να ορίσουν σελιδοδείκτες στους φακέλους και στα άλμπουμ τους για ευκολότερη πρόσβαση, ωστόσο η δωρεάν έκδοση της εφαρμογής υποστηρίζει μέχρι δυο σελιδοδείκτες. Η αναζήτηση αρχείων γίνεται με βάση το όνομα των αρχείων και των μεταδεδομένων που πιθανώς να περιέχουν, όπως ετικέτες, αξιολόγηση, μοντέλο κάμερας κτλ. Επίσης υποστηρίζεται και η χρήση ενός χάρτη για την ευκολότερη αναζήτηση αρχείων που διαθέτουν μεταδεδομένα τοποθεσίας.

Μία από τις σημαντικότερες λειτουργίες της συγκεκριμένης εφαρμογής είναι η δυνατότητα δημιουργίας "έξυπνων άλμπουμ" (Smart Albums). Στα συγκεκριμένα άλμπουμ θα προστίθεται αυτόματα ένα νέα αρχείο εάν πληροί κάποιες προϋποθέσεις τις οποίες έχει ορίσει ο χρήστης για το άλμπουμ, για παράδειγμα μια συγκεκριμένη ετικέτα, μια τοποθεσία ή μια συγκεκριμένη βαθμολογία (rating). Επίσης, οι συλλογές και οι φάκελοι μπορούν να προστατευθούν με τη χρήση κωδικού, ώστε να μην είναι ορατές από τρίτες εφαρμογές. Το F-Stop Gallery υποστηρίζει και την σύνδεση με cloud υπηρεσίες, συγκεκριμένα του Dropbox και του Google Drive, για την προβολή των εικόνων και των βίντεο που είναι αποθηκευμένα σε αυτές. Η εφαρμογή κυκλοφορεί σε δωρεάν έκδοση η οποία περιέχει διαφημίσεις, αλλά υπάρχει και η έκδοση PRO, η οποία είναι επί πληρωμή και προσφέρει ορισμένες επιπλέον λειτουργίες.



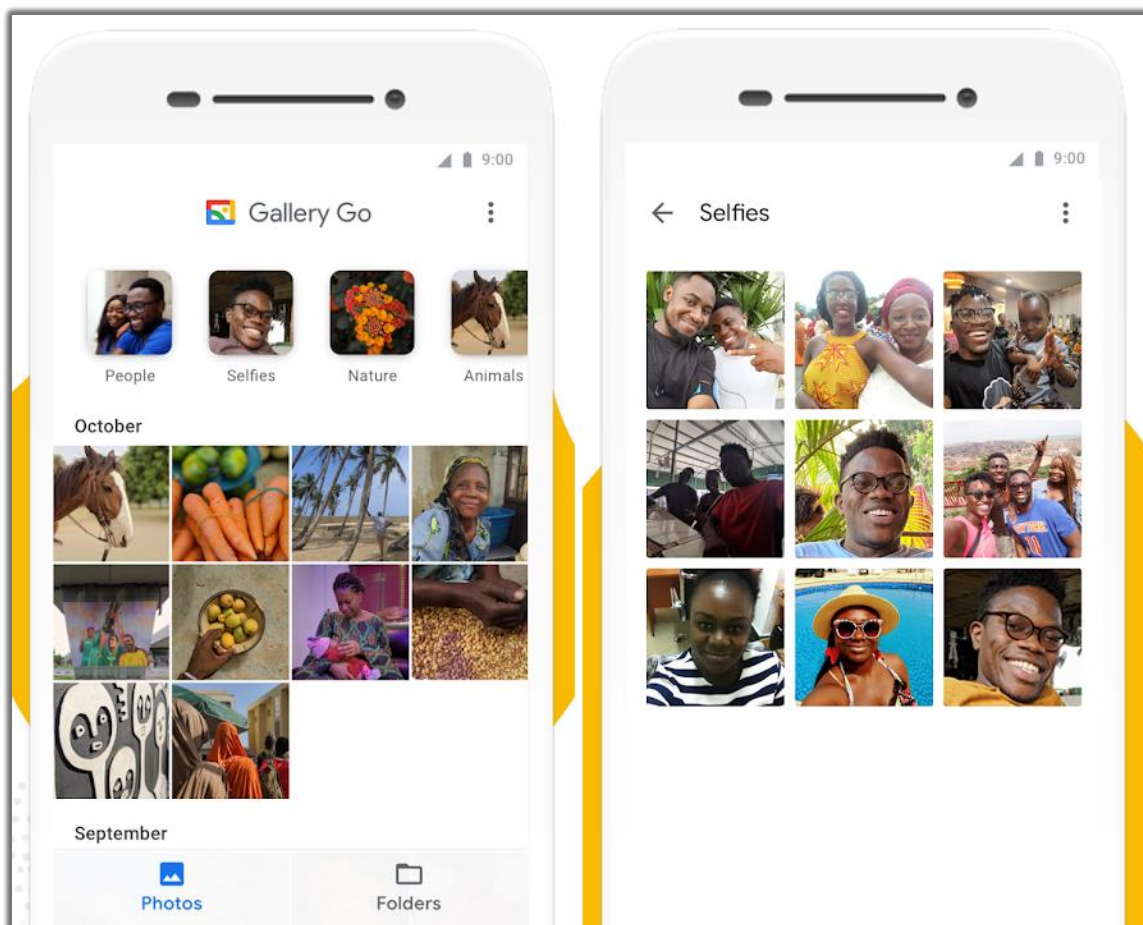
Εικόνα 28: F-Stop Gallery [Source: <https://play.google.com/store/apps/details?id=com.fstop.photo>]

Η εφαρμογή F-Stop Gallery συγκριτικά με την εφαρμογή της παρούσης εργασίας υστερεί σε ορισμένα χαρακτηριστικά και υπερτερεί σε άλλα. Αναμφίβολα τα "έξυπνα άλμπουμ" αποτελούν ένα μοναδικό χαρακτηριστικό για την αυτόματη ταξινόμηση και οργάνωση αρχείων βάση συγκεκριμένων παραμέτρων που ορίζονται από το χρήστη. Ωστόσο κάποιες δυνατότητες που παρέχει η εφαρμογή της παρούσας εργασίας δεν υποστηρίζονται από την εφαρμογή F-Stop Gallery, ή είναι διαθέσιμες μόνο στην PRO έκδοση. Για παράδειγμα, οι χρήστες της εφαρμογής F-Stop Gallery δεν μπορούν να ορίσουν τοποθεσία στις φωτογραφίες και στα βίντεο. Η συγκεκριμένη εφαρμογή μπορεί μόνο να αναγνωρίσει εάν κάποιο αρχείο έχει ήδη μεταδεδομένα τοποθεσίας, τα οποία μπορούν να αξιοποιηθούν μόνο στην PRO έκδοση της εφαρμογής με τη χρήση του χάρτη. Επίσης η αναζήτηση των αρχείων επιστρέφει μέχρι δέκα αποτελέσματα κάθε φορά, περιορίζοντας σημαντικά την λειτουργικότητα της εφαρμογής σχετικά με τη διαχείριση και οργάνωση των αρχείων.

4.5 GALLERY GO

Το Gallery Go είναι μια γρήγορη και απλή android εφαρμογή για την οργάνωση φωτογραφιών και βίντεο στις φορητές συσκευές. Η εφαρμογή έχει αναπτυχθεί από την Google και αποτελεί μία lightweight έκδοση της εφαρμογής Google Photos. Είναι σχεδιασμένη κυρίως για συσκευές μεσαίας κατηγορίας και χαμηλότερα όσον αφορά το υλικό, καθώς μπορεί να χρησιμοποιηθεί σε συσκευές με ελάχιστο αποθηκευτικό χώρο διότι απαιτεί μόνο 10MB διαθέσιμης μνήμης για την εγκατάσταση της. Παρόλα αυτά, δεν υστερεί ως προς τη λειτουργικότητα που προσφέρει.

Το Gallery Go οργανώνει αυτόματα τις φωτογραφίες και τα βίντεο σε ένα πλήθος κατηγοριών όπως, selfies, τοπία, ζώα, βίντεο ακόμα και ταινίες. Επιπλέον υποστηρίζει τεχνολογία αναγνώρισης προσώπων για αποτελεσματικότερη ταξινόμηση και οργάνωση των φωτογραφιών. Ωστόσο, το συγκεκριμένο χαρακτηριστικό είναι διαθέσιμο μόνο σε επιλεγμένα μέρη και χώρες του κόσμου. Επίσης ταξινομεί τις φωτογραφίες με χρονολογική σειρά και τις συλλογές και τα άλμπουμ με βάση το όνομα, το μέγεθος των αρχείων που περιέχουν ή την ημερομηνία τροποποίησης. Η εφαρμογή Gallery Go οργανώνει αυτόματα τα αρχεία σε φακέλους με βάση αυτά που έχουν τραβηχτεί από την κάμερα της συσκευής, αυτά που είναι στιγμιότυπα οθόνης (screenshots) και αυτά που είναι αποθηκευμένα στη μνήμη της συσκευής ή σε κάποια SD κάρτα. Οι χρήστες μπορούν εύκολα να δημιουργήσουν νέους φακέλους και να προσθέσουν ή να μετακινήσουν σε αυτούς τα αρχεία που επιθυμούν.



Εικόνα 29: Gallery Go [Source:

<https://play.google.com/store/apps/details?id=com.google.android.apps.photosgo>]

Παράλληλα παρέχει βασικές λειτουργίες επεξεργασίας εικόνων σχετικά με τη ρύθμιση της φωτεινότητας και της αντίθεσης, την περικοπή και μετακίνηση τμημάτων της εικόνας καθώς και μια λειτουργία αυτόματης επεξεργασίας, που επιλέγει τις καλύτερες ρυθμίσεις για την κάθε φωτογραφία. Οι χρήστες της εφαρμογής μπορούν να χρησιμοποιήσουν κανονικά όλες τις δυνατότητες και τα χαρακτηριστικά της, ακόμα και χωρίς σύνδεση στο ίντερνετ καθώς όλες οι λειτουργίες της αλληλοεπιδρούν κατευθείαν με τα αρχεία της συσκευής οπότε δεν είναι απαραίτητη η χρήση δεδομένων για συγχρονισμό των αρχείων σε κάποιο cloud. Η εφαρμογή παρέχεται δωρεάν για όλες τις Android συσκευές με έκδοση λειτουργικού 8.0 ή πιο πρόσφατη και ήδη έχει ξεπεράσει τις 50 εκατομμύρια λήψεις.

Σε σύγκριση με την εφαρμογή της παρούσας εργασίας, η εφαρμογή Gallery Go δεν διαθέτει λειτουργία αναζήτησης και προσφέρει περιορισμένες δυνατότητες επεξεργασίας μεταδεδομένων για την οργάνωση των αρχείων. Συγκεκριμένα, δεν υποστηρίζεται η δυνατότητα εισαγωγής ετικετών ή τοποθεσίας στις φωτογραφίες και στα βίντεο, καθώς η εφαρμογή Gallery Go χρησιμοποιεί έναν αλγόριθμο τεχνητής νοημοσύνης, ο οποίος ταξινομεί αυτόματα τα αρχεία σε κατηγορίες της επιλογής του, χωρίς ο χρήστης να μπορεί να επέμβει. Από τη μια μεριά αυτό μπορεί να εξοικονομήσει χρόνο από την οργάνωση των αρχείων, χωρίς όμως να σημαίνει πως το αποτέλεσμα θα είναι αποδοτικό και κατάλληλο για τις ανάγκες του χρήστη. Επίσης η τεχνολογία αναγνώρισης προσώπων που παρέχει η εφαρμογή Gallery Go, παρόλο που είναι ένα χρήσιμο εργαλείο οργάνωσης και ταξινόμησης των αρχείων, δεν είναι διαθέσιμη για όλους τους χρήστες. Αντίθετα, η εφαρμογή της παρούσας εργασίας δίνει τον έλεγχο στον χρήστη να οργανώσει και να ταξινομήσει τα αρχεία του σύμφωνα με τις δικές του επιλογές, ορίζοντας στα αρχεία ετικέτες, ημερομηνίες και τοποθεσίες της επιλογής του. Επιπλέον, η εφαρμογή Gallery Go ταξινομεί τις φωτογραφίες και τα βίντεο σύμφωνα με ορισμένες παραμέτρους, αλλά δεν διαθέτει καμιά λειτουργία αναζήτησης. Αυτό έχει ως αποτέλεσμα, οι χρήστες με μεγάλο πλήθος φωτογραφιών και βίντεο να μην μπορούν να βρουν και να διαχειριστούν εύκολα και γρήγορα τα αρχεία τους.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι Android φορητές συσκευές όπως smartphones και tablets αποτελούν τις δημοφιλέστερες προσωπικές συσκευές σήμερα. Μία από τις κυριότερες λειτουργίες τους είναι η λήψη φωτογραφιών και βίντεο. Σε αυτό οφείλεται η ραγδαία εξάπλωση των φορητών συσκευών αλλά και των κοινωνικών δικτύων τα οποία σχετίζονται άμεσα με την δημοσίευση και προβολή εικόνων και βίντεο. Ένα αρκετά διαδεδομένο και συχνό πρόβλημα που δημιουργείται από το παραπάνω φαινόμενο είναι η συσσώρευση μεγάλου όγκου ανοργάνωτων και μη ταξινομημένων φωτογραφιών και βίντεο στις φορητές συσκευές.

Με τη βοήθεια της τεχνολογίας που αναπτύσσεται καθημερινά, έχουν δημιουργηθεί και συνεχώς εξελίσσονται δεκάδες εφαρμογές σχετικά με την οργάνωση και διαχείριση φωτογραφιών και βίντεο. Βασικός στόχος των εφαρμογών αυτών είναι να βοηθήσουν τους χρήστες να διαχειρίζονται ευκολότερα και αποδοτικότερα το μεγάλο πλήθος εικόνων και βίντεο που συχνά συσσωρεύεται στον αποθηκευτικό χώρο των συσκευών τους. Οι συγκεκριμένες εφαρμογές μπορούν να ωφελήσουν και να αξιοποιηθούν από απλούς χρήστες που βγάζουν πολλές φωτογραφίες καθημερινά, από άτομα που ασχολούνται με τη φωτογραφία ερασιτεχνικά ή ως χόμπι, μέχρι και από επαγγελματίες φωτογράφους. Η αποτελεσματική οργάνωση των φωτογραφιών και των βίντεο παρέχει πολλά οφέλη, όπως ταχύτερη αναζήτηση και εύρεση συγκεκριμένων αρχείων, εξοικονόμηση χρόνου στη διαχείρισή τους, ακόμα και αποδοτικότερο έλεγχο προόδου και δουλειάς εάν πρόκειται για επαγγελματία φωτογράφο.

Οι περισσότερες εφαρμογές οργάνωσης φωτογραφιών και βίντεο διακρίνονται από ορισμένα κοινά χαρακτηριστικά όπως προσθήκη ετικετών, αξιολογήσεων, αυτόματη ανίχνευση και διαγραφή διπλότυπων και δημιουργία άλμπουμ και θεματικών συλλογών για αποτελεσματικότερη αρχειοθέτηση. Ακόμα, έχουν αναπτυχθεί εφαρμογές με τεχνολογίες αναγνώρισης προσώπου και θεματικών ενοτήτων ώστε να ταξινομούν αυτόματα τα αρχεία γλιτώνοντας τους χρήστες από μία αρκετά χρονοβόρα διαδικασία. Ορισμένες εφαρμογές συνδυάζουν τα παραπάνω χαρακτηριστικά και επιπλέον παρέχουν δυνατότητες επεξεργασίας φωτογραφιών και βίντεο καθώς και online υπηρεσίες για την αποθήκευση των αρχείων σε κάποιο cloud με σκοπό την εξοικονόμηση αποθηκευτικού χώρου από τις συσκευές των χρηστών. Σχετικά με το κόστος, οι περισσότερες εφαρμογές διαθέτουν μία δωρεάν έκδοση και μία επί πληρωμή, συνήθως μέσω μηνιαίας συνδρομής. Οι δωρεάν εκδόσεις των συγκεκριμένων εφαρμογών είναι σχετικά περιορισμένες και συνήθως παρέχουν τις βασικές λειτουργίες, ενώ οι συνδρομητικές, οι οποίες αποτελούν αναβάθμιση των δωρεάν, προσφέρουν περισσότερες και πιο εξειδικευμένες δυνατότητες και χαρακτηριστικά.

Η Android εφαρμογή που σχεδιάστηκε και αναπτύχθηκε στα πλαίσια της διπλωματικής εργασίας στοχεύει στην αποτελεσματικότερη οργάνωση και διαχείριση αρχείων φωτογραφίας και βίντεο. Η ανάπτυξη μιας εφαρμογής σε περιβάλλον Android προσφέρει πολλαπλά οφέλη και πλεονεκτήματα στην κοινότητα των προγραμματιστών. Η ευρεία διάδοση του συγκεκριμένου λειτουργικού έχει συμβάλει στην ανάπτυξη εργαλείων, βιβλιοθηκών και υπηρεσιών που βοήθησαν σημαντικά στην υλοποίηση πολλών από τις λειτουργίες της εφαρμογής.

Η συγκεκριμένη εφαρμογή αντιμετωπίζει επιτυχώς το πρόβλημα της συσσώρευσης μεγάλου πλήθους αρχείων στις συλλογές και στα άλμπουμ των χρηστών μέσω των λειτουργιών και των δυνατοτήτων που προσφέρει. Συνδυάζει αρκετά από τα χαρακτηριστικά αντίστοιχων εφαρμογών μαζί με μία απλοϊκή και εύχρηστη διεπαφή. Πρόκειται για μια εφαρμογή-gallery η οποία προβάλλει όλα τα αρχεία εικόνας και βίντεο της συσκευής και υποστηρίζει την εισαγωγή και επεξεργασία ετικετών, ημερομηνιών και τοποθεσιών στις φωτογραφίες και στα βίντεο για την αποδοτικότερη ταξινόμηση

και οργάνωση τους. Επίσης διαθέτει λειτουργία αναζήτησης αρχείων η οποία προσφέρει εύκολη και γρήγορη πρόσβαση στα αρχεία της συσκευής.

Η εφαρμογή έχει προοπτικές για ορισμένες αναβαθμίσεις μελλοντικά, με την προσθήκη ορισμένων νέων λειτουργιών και χαρακτηριστικών. Μία χρήσιμη λειτουργία που θα μπορούσε να προστεθεί στην εφαρμογή, είναι η δυνατότητα διαχείρισης της δομής των αρχείων και των φακέλων τους. Αυτό συμπεριλαμβάνει λειτουργίες όπως μετακίνηση, αντιγραφή και διαγραφή αρχείων, καθώς και τη δημιουργία νέων άλμπουμ και συλλογών για τις φωτογραφίες και τα βίντεο. Επίσης θα ήταν χρήσιμο να υλοποιηθεί η δυνατότητα διαμοιρασμού των φωτογραφιών και των βίντεο μέσω υπηρεσιών και εφαρμογών ανταλλαγής αρχείων και μηνυμάτων όπως mail, bluetooth, messenger και viber, αλλά και η δυνατότητα κοινοποίησης τους σε κοινωνικά δίκτυα όπως Facebook, Twitter, Instagram μέσω της εφαρμογής. Τέλος, θα ωφελούσε η εφαρμογή να έχει τη δυνατότητα να προτείνει στους χρήστες αρχεία μεγάλου μεγέθους, τα οποία θα μπορούσαν να αποθηκευτούν σε κάποια cloud υπηρεσία, ώστε η εφαρμογή να διατηρεί μια χαμηλότερης ανάλυσης έκδοση του αρχείου, εξοικονομώντας αποθηκευτικό χώρο στη συσκευή του χρήστη.

Συμπεραίνοντας, είναι προφανές πως το πρόβλημα της οργάνωσης μεγάλου αριθμού φωτογραφιών και βίντεο που συσσωρεύονται στις φορητές συσκευές, οδήγησε στην ανάπτυξη και εξέλιξη εφαρμογών για την αποτελεσματικότερη οργάνωση και ταξινόμησή τους. Η κάθε εφαρμογή στοχεύει στη συνεχή βελτίωση και εξέλιξη των λειτουργιών και των δυνατοτήτων της και στη δημιουργία μοναδικών χαρακτηριστικών με σκοπό να ξεχωρίσει από τον ανταγωνισμό. Ωστόσο, η τελική επιλογή εξαρτάται κυρίως από τις απαιτήσεις του κάθε χρήστη και τη λειτουργικότητα που ο ίδιος επιθυμεί να έχει.

6. ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ

- [1] WIKIPEDIA. 2021. *Android (operating system)*. Wikipedia. [Online]. Available: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)). [January 25, 2021].
- [2] WIKIPEDIA. 2021. *Android software development*. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Android_software_development. [January 3, 2021].
- [3] Android Developers. 2020. *Create dynamic lists with RecyclerView*. Android Developers. [Online]. Available: <https://developer.android.com/guide/topics/ui/layout/recyclerview>. [November 11, 2020].
- [4] Android Developers. 2020. *PagerAdapter*. Android Developers. [Online]. Available: <https://developer.android.com/reference/androidx/viewpager/widget/PagerAdapter>. [September 30, 2020].
- [5] Google Maps Platform. 2021. *Place Autocomplete*. Google Developers. [Online]. Available: <https://developers.google.com/places/web-service/autocomplete>. [February 2, 2021].
- [6] Singh, N. 2017. *Understanding RecyclerView Part 1: The Basics*. Medium. [Online]. Available: <https://medium.com/@nileshsingh/understanding-recyclerview-part-1-the-basics-a7bd07cfae93>. [December 19, 2017].
- [7] Charfaoui, Y. 2020. *Using Glide to Efficiently Load Images in Android*. Heartbeat. [Online]. Available: <https://heartbeat.fritz.ai/using-glide-to-efficiently-load-images-in-android-aec0ba9639ea>. [June 2, 2020].
- [8] Nzivu, B. 2020. *Bottom Navigation Bar in Android Applications*. Section. [Online]. Available: <https://www.section.io/engineering-education/bottom-navigation-bar-in-android/>. [October 28, 2020].
- [9] Tutorialspoint. *Android – Architecture*. [Online]. Available: https://www.tutorialspoint.com/android/android_architecture.htm.
- [10] Kordić, L. 2020. *Taming File Storage on Android - Part 2*. Medium. [Online]. Available: <https://medium.cobeisfresh.com/taming-file-storage-on-android-part-2-4afe9dfd9b6e>. [April 20, 2020].
- [11] Zibtek. 2020. *Advantages Of The Android Application Development*. Zibtek. [Online]. Available: <https://www.zibtek.com/Advantages-of-native-android-application-development>. [December 9, 2020].
- [12] AndroidAppsForMe. *13 Best Photo Organizer Apps for Android*. [Online]. Available: <https://androidappsforme.com/photo-organizer-apps/>.
- [13] Caioni, S. 2020. *Top 29 Best Photo Organizer Apps in 2020*. Pixinfocus. [Online]. Available: <https://www.pixinfocus.com/photo-organizer/>. [July 10, 2020].
- [14] Eapen, B. 2019. *Top 25 Photo Organizing Software and Apps of 2020*. Pixpa. [Online]. Available: <https://www.pixpa.com/blog/photo-organiser>. [November 6, 2019].
- [15] Clement, J. 2020. *Number of available applications in the Google Play Store from December 2009 to September 2020*. Statista. [Online]. Available: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>. [November 24, 2020].
- [16] Muchmore, M. 2016. *Flickr Review*. PCMag. [Online]. Available: <https://www.pcmag.com/reviews/flickr>. [March 10, 2016].
- [17] Rae, M. 2018. *The Best Apps for Organizing Photos*. Techlicious. [Online]. Available: <https://www.techlicious.com/tip/best-apps-photo-organization/>. [March 12, 2018].
- [18] MYLIO. 2021. *Mylio | Free Photo Organizer*. [Online]. Available: <https://mylio.com/>.
- [19] Ελπασίδης, Μ. 2013. *Σχεδίαση και ανάπτυξη εφαρμογής Android για καταγραφή επιδόσεων τζοβόλων*. [Πτυχιακή εργασία]. Τεχνολογικό Εκπαιδευτικό Ίδρυμα Καβάλας. 82pp. Available from <http://digilib.teiemt.gr/jspui/bitstream/123456789/1521/1/012013002.pdf>
- [20] Χαλιμούρδας, Π. 2016. *Οι Καλύτερες Εφαρμογές για Διαχείριση Φωτογραφιών*. PC-Steps. [Online]. Available: <https://www.pcsteps.gr/119238-οργάνωση-διαχείριση>. [August 31, 2016].