

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**



**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΑΙΤΗΜΑ ΣΥΝΑΝΤΗΣΗΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ
ANDROID**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΠΑΝΑΓΙΩΤΗΣ Δ. ΡΙΖΟΣ**

Επιβλέποντες Καθηγητές:

**Σταμούλης Γεώργιος
Καθηγητής Π.Θ.**

**Τσομπανοπούλου Παναγιώτα
Αναπληρώτρια Καθηγήτρια Π.Θ.**

Βόλος, 2016

**UNIVERSITY OF THESSALY
TECHNICAL UNIVERSITY**



**DEVELOPMENT OF MEETING REQUEST APPLICATION IN ANDROID
ENVIRONMENT**

**DIPLOMA THESIS
PANAGIOTIS D. RIZOS**

Supervisors:

**Stamoulis Georgios
Professor UTH**

**Tsompanopoulou Panagiota
Associate Professor UTH**

Volos, 2016

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

**Πολυτεχνική Σχολή Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών**



**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΑΙΤΗΜΑ ΣΥΝΑΝΤΗΣΗΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ
ANDROID**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΑΝΑΓΙΩΤΗΣ Δ. ΡΙΖΟΣ

(Υπογραφή)

.....

ΣΤΑΜΟΥΛΗΣ ΓΕΩΡΓΙΟΣ

ΚΑΘΗΓΗΤΗΣ Π.Θ.

(Υπογραφή)

.....

ΤΣΟΜΠΑΝΟΠΟΥΛΟΥ ΠΑΝΑΓΙΩΤΑ

ΑΝΑΠΛΗΡΩΤΡΙΑ ΚΑΘΗΓΗΤΡΙΑ Π.Θ.

(Υπογραφή)

.....

ΠΑΝΑΓΙΩΤΗΣ ΡΙΖΟΣ

© 2016 – ALL RIGHTS RESERVED

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών

COPYRIGHT © ΠΑΝΑΓΙΩΤΗΣ ΡΙΖΟΣ, 2016

ΜΕ ΕΠΙΦΥΛΑΞΗ ΠΑΝΤΟΣ ΔΙΚΑΙΩΜΑΤΟΣ. ALL RIGHTS RESERVED

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα

Η σελίδα αυτή είναι σκόπιμα λευκή

ΕΥΧΑΡΙΣΤΙΕΣ

Με την παρούσα εργασία περατώνονται οι σπουδές μου στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, τον κύριο Γεώργιο Σταμούλη για την βοήθεια, υποστήριξη και εμπιστοσύνη που μου έδειξε κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας, καθώς επίσης και την κυρία Παναγιώτα Τσομπανοπούλου συνεπιβλέπουσα της διπλωματικής. Με καθοδήγησαν με τρόπο καταλυτικό, αντιμετωπίζοντας τις όποιες δυσκολίες προέκυψαν.

Τέλος οφείλω ένα μεγάλο ευχαριστώ στην οικογένεια μου, για την πολύτιμη υπομονή και στήριξη τους προς το πρόσωπο μου όλα αυτά τα χρόνια.

Ρίζος Παναγιώτης
Βόλος, 2016

Η σελίδα αυτή είναι σκόπιμα λευκή

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία με τίτλο "ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΑΙΤΗΜΑ ΣΥΝΑΝΤΗΣΗΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ANDROID" έχει ως στόχο την υλοποίηση μιας εφαρμογής σε περιβάλλον Android συμβατό με τις περισσότερες πλέον κινητές συσκευές.

Η δημιουργία της εφαρμογής αυτής έγινε με την χρήση του λογισμικού Android Studio, ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην Android πλατφόρμα. Η εφαρμογή μπορεί να χρησιμοποιηθεί από όλους τους κατόχους και χρήστες κινητών τηλεφώνων με λειτουργικό Android. Ο σκοπός της εφαρμογής είναι η καλύτερη και ευκολότερη συνάντηση των χρηστών.

Συγκεκριμένα, ο χρήστης μπορεί να στείλει ένα αίτημα σε έναν άλλο χρήστη με σκοπό να συναντηθούνε, συμπληρώνοντας μια φόρμα με τα πεδία Contact, Title, Description, Location, Date & Time. Με βάση τα στοιχεία που συμπληρώνει ο χρήστης που στέλνει το αίτημα, ο χρήστης στον οποίο προορίζεται το αίτημα μπορεί να αρνηθεί και να δεχθεί το αίτημα συνάντησης. Τέλος, υπάρχει η δυνατότητα στον χρήστη αν θέλει να γίνει η συνάντηση αλλά δεν το βολεύουν κάποια από τα πεδία του αιτήματος (π.χ. Location, Date & Time) να τα τροποποιήσει και να στείλει πίσω στον αρχικό χρήστη ένα ανανεωμένο αίτημα συνάντησης.

ABSTRACT

This thesis **with the title** "DEVELOPMENT OF MEETING REQUEST APPLICATION IN ANDROID ENVIRONMENT" **has as a goal to implement an** application in Android environment which is compatible with the majority of smartphones.

The development of the application became with the usage of the Android Studio, an integrated development environment (IDE) for Android platform development. The application can be used from all the users who have Android smartphones. The goal of this app is the better and easier meeting of the users.

In particular, the user can send a request to another user in order to meet each other, by filling a form of fields like **"Contact, Title, Description, Location, Date & Time"**. **Based on the input of the user who send the** request, the user to whom this request is intended could Reject or Approve it. Last but not least, this app gives to the user the possibility to Update some of the fields (e.g Location, Date & Time) and send an updated request back to the initial user.

Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ.....	1
1.1	ΕΞΥΠΝΑ ΤΗΛΕΦΩΝΑ	1
1.2	ΤΙ ΕΙΝΑΙ ΤΟ ANDROID.....	3
1.3	ΕΚΔΟΣΕΙΣ ΤΟΥ ANDROID	4
1.3.1	Android 1.0 & 1.1	5
1.3.2	Android 1.5 Cupcake	6
1.3.3	Android 1.6 Donut.....	7
1.3.4	Android 2.0 & 2.1 Eclair	7
1.3.5	Android 2.2 Froyo.....	8
1.3.6	Android 2.3 Gingerbread	8
1.3.7	Android 3.0 Honeycomb.....	9
1.3.8	Android 4.0 Ice Cream Sandwich	10
1.3.9	Android 4.1 Jelly Bean	11
1.3.10	Android 4.4 Kit Kat	11
1.3.11	Android 5.0 Lollipop	12
1.3.12	Android 6.0 Marshmallow	13
1.4	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ANDROID.....	14
1.4.1	Λειτουργίες Οθόνης	14
1.4.2	Αποθήκευση Δεδομένων.....	14
1.4.3	Συνδεσιμότητα.....	14
1.4.4	Αποστολή Μηνυμάτων.....	14
1.4.5	Περιήγηση στον Ιστό	14
1.4.6	Υποστήριξη Java	14
1.4.7	Υποστήριξη Πολυμέσων	14
1.4.8	Επιπλέον Υποστήριξη Υλικού	15
1.4.9	Περιβάλλον Ανάπτυξης Λογισμικού	15
1.4.10	Αγορά και Εγκατάσταση Εφαρμογών	15
2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ANDROID.....	16
2.1	LINUX KERNEL.....	16
2.2	NATIVE LIBRARIES	17
2.3	ANDROID RUNTIME	18
2.4	APPLICATION FRAMEWORK	19

2.5	APPLICATIONS.....	19
3.0	ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID	20
3.1	ΕΡΓΑΛΕΙΑ	20
3.2	ΓΛΩΣΣΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ	21
3.3	ΚΥΚΛΟΣ ΖΩΗΣ ΕΦΑΡΜΟΓΗΣ ANDROID	22
4	ΔΟΜΗ ΕΦΑΡΜΟΓΗΣ ANDROID	24
4.1	ANDROID MANIFEST	24
4.2	ACTIVITY	25
4.3	INTENT RECEIVER.....	26
4.4	SERVICES.....	26
4.5	CONTENT PROVIDERS	26
4.6	RESOURCES.....	27
4.7	USER INTERFACE	27
4.7.1	Layout	27
4.7.2	Menu	28
4.7.3	Dialogs	28
4.7.4	Notifications.....	29
5	ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ BOOKI.....	30
5.1	VERIFICATION ΧΡΗΣΤΗ (Account Kit)	30
5.2	ΑΡΧΙΚΗ ΟΘΟΝΗ	31
5.3	MEETINGS.....	32
5.4	NEW MEETING	33
5.5	SETTINGS	39
6	ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	40
	ΒΙΒΛΙΟΓΡΑΦΕΙΑ.....	87

1 ΕΙΣΑΓΩΓΗ

1.1 ΕΞΥΠΝΑ ΤΗΛΕΦΩΝΑ

Η τεχνολογία ολοένα και εξελίσσεται καθώς περνάνε τα χρόνια και πιο συγκεκριμένα όσον αφορά το χώρο της κινητής τηλεφωνίας. Τα τελευταία χρόνια τα έξυπνα τηλέφωνα (smartphones) ενσωματώνουν δυνατότητες που μέχρι σήμερα δεν υπήρχαν, με αποτέλεσμα οι κινητές συσκευές να εισχωρούν όλο και περισσότερο στη ζωή μας.

Τα έξυπνα τηλέφωνα (smartphones) είναι κινητά τηλέφωνα βασισμένα σε ένα λειτουργικό σύστημα κινητής τηλεφωνίας με περισσότερη προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα απλό κινητό τηλέφωνο. Τα πρώτα smartphones συνδύαζαν τις λειτουργίες ενός προσωπικού ψηφιακού βοηθού (PDA) και ενός κινητού τηλεφώνου. Σε μεταγενέστερα μοντέλα προστέθηκαν λειτουργίες, όπως αναπαραγωγή μουσικών κομματιών, ψηφιακή φωτογραφική μηχανή και με δυνατότητα λήψης βίντεο καθώς και δυνατότητα πλοήγησης GPS, με αποτέλεσμα διαμορφωθεί μια πολυχρηστική συσκευή.

Πλέον τα σύγχρονα έξυπνα τηλέφωνα διαθέτουν όλα οθόνες αφής υψηλής ανάλυσης και διαδικτυακούς περιηγητές (web browsers) που εμφανίζουν τυποποιημένες ιστοσελίδες καθώς και βελτιστοποιημένες ιστοσελίδες για κινητά. Η πρόσβαση στο διαδίκτυο γίνεται μέσω Wi-Fi τεχνολογίας και κινητών ευρυζωνικών υπηρεσιών.



Εικόνα 1: Έξυπνα Τηλέφωνα (Smartphones)



Εικόνα 2: Έξυπνα Τηλέφωνα (Smartphones)

1.2 ΤΙ ΕΙΝΑΙ ΤΟ ANDROID

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, βασισμένο στο πυρήνα του λειτουργικού συστήματος Linux, για φορητές συσκευές όπως smartphones και tablets. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Handset Alliance|Open Handset Alliance και σκοπός είναι να επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Αν και κατά κύριο λόγο είναι σχεδιασμένο για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί και σε κονσόλες παιχνιδιών, σε ψηφιακές φωτογραφικές μηχανές, σε συνηθισμένους Η/Υ και σε άλλες ηλεκτρονικές συσκευές. Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με Android έχουν περισσότερες πωλήσεις από όλες τις συσκευές Windows, iOS και Mac OS X μαζί.



Εικόνα 3: Android Logo

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας πολλών τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του

κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Ιρίνα Μπλόκ.

Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας.

1.3 ΕΚΔΟΣΕΙΣ ΤΟΥ ANDROID

Η ιστορία εκδόσεων του Android ξεκίνησε με την κυκλοφορία του Android beta το Νοέμβριο του 2007. Το Android είναι υπό συνεχή ανάπτυξη από την Google και την Open Handset Alliance (OHA), και έχουν γίνει μια σειρά από ενημερώσεις στην λειτουργία του συστήματος από την αρχική κυκλοφορία του.

Από τον Απρίλη του 2009 μέχρι τον Νοέμβριο του 2015 έχουν κυκλοφορήσει 11 κύριες εκδόσεις, οι οποίες έχουν θέμα διαφόρων γλυκών επιδορπιών στην κωδική ονομασία τους, και κυκλοφόρησαν σε αλφαβητική σειρά, εξαιρουμένων των εκδόσεων 1.0 και 1.1, που δεν τέθηκαν υπό συγκεκριμένα κωδικά ονόματα.



Εικόνα 4: Android Versions

1.3.1 Android 1.0 & 1.1

Η πρώτη εμπορική έκδοση ήταν το Android 1.0 που κυκλοφόρησε το Σεπτέμβριο του 2008, μαζί με τη συσκευή, «Hero HTC». Τόσο το λειτουργικό σύστημα όσο και η συσκευή HTC έλαβαν θετικές κριτικές.



**ANDROID 1.0
APPLE PIE**

Εικόνα 5: Android 1.0

Η Έκδοση 1.1 κυκλοφόρησε το Φεβρουάριο του 2009.



Banana Bread

Εικόνα 6: Android 1.1

1.3.2 Android 1.5 Cupcake

Κυκλοφόρησε στις 30 Απριλίου του 2009, και σηματοδότησε την πρώτη σημαντική αναβάθμιση του Android. Μερικά από τα χαρακτηριστικά της έκδοσης ήταν η υποστήριξη εικονικών πληκτρολογίων με πρόβλεψη κειμένου, αντιγραφή και επικόλληση χαρακτηριστικών στο πρόγραμμα περιήγησης στο Web, ανέβασμα βίντεο στο YouTube και φωτογραφιών στο Picasa και στερεοφωνική υποστήριξη για Bluetooth.



Cupcake
Android 1.5

Ε.

Εικόνα 7: Android Cupcake

1.3.3 Android 1.6 Donut

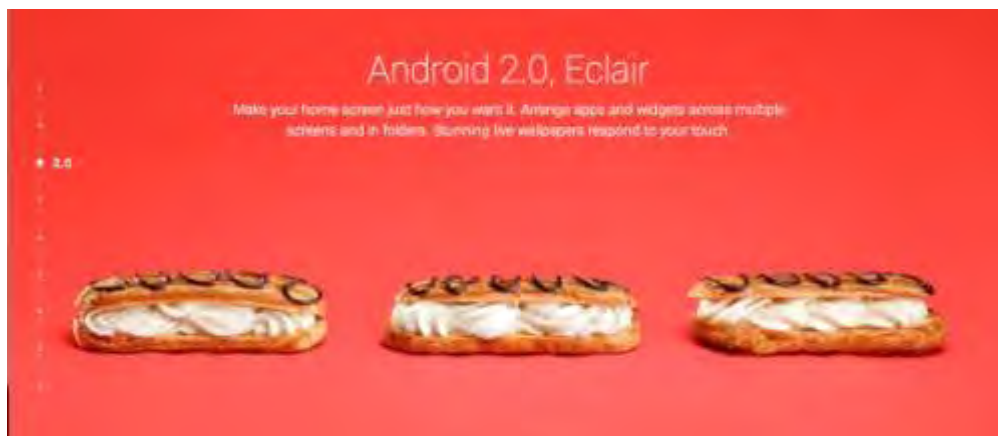
Κυκλοφόρησε στις 15 Σεπτεμβρίου του 2009 με σημαντικές αναβαθμίσεις στο βίντεο και στην φωτογραφική μηχανή, όπως για παράδειγμα δυνατότητα επιλογής πολλαπλών φωτογραφιών για διαγραφή. Το Google Maps ήταν η σημαντικότερη προσθήκη, ενώ η φωνή και το κείμενο αναζήτησης εισόδου ενισχύθηκαν και συμπεριέλαβαν σελιδοδείκτες, τις επαφές και το διαδίκτυο.



Εικόνα 8: Android Donut

1.3.4 Android 2.0 & 2.1 Eclair

Κυκλοφόρησε στις 26 Οκτωβρίου του 2009 και ακολουθήθηκε από την 2.1 τον Ιανουάριο του 2010. Κάποια χαρακτηριστικά της έκδοσης αυτής ήταν προσθήκη live wallpaper, υποστήριξη Bluetooth 2.1, multi-touch υποστήριξη και νέα χαρακτηριστικά κάμερας όπως φλας, ψηφιακό zoom, λειτουργία σκηνής, ισορροπία λευκού, εφέ χρώματος και macro εστίαση. Τέλος υπήρχε και επιλογή πολλαπλών λογαριασμών email σε μία συσκευή.



Εικόνα 9: Android Eclair

1.3.5 Android 2.2 Froyo

Κυκλοφόρησε στις 26 Μαΐου του 2010. Κάποια χαρακτηριστικά της έκδοσης αυτής ήταν Adobe Flash υποστήριξη για παρακολούθηση βίντεο μέσω browser, USB tethering και Wi-Fi hotspot λειτουργικότητα.



Εικόνα 10: Android Froyo

1.3.6 Android 2.3 Gingerbread

Κυκλοφόρησε στις 6 Δεκεμβρίου του 2010, ενώ τον Φεβρουάριο του 2011 επανεκδόθηκε σε Android 2.3.3. Μέσα στις αναβαθμίσεις ήταν υποστήριξη για Near Field Communication (NFC), που επιτρέπει στο χρήστη να διαβάσει μια ετικέτα NFC που είναι ενσωματωμένη σε μια αφίσα, αυτοκόλλητο, ή σε διαφήμιση, υποστήριξη για περισσότερους αισθητήρες και φυσικά υποστήριξη για πολλαπλές κάμερες στη συσκευή, συμπεριλαμβανομένων κάμερα στο μπροστινό μέρος.



Εικόνα 11: Android Gingerbread

1.3.7 Android 3.0 Honeycomb

Κυκλοφόρησε στις 9 Μαΐου του 2011, με την ιδιαιτερότητα ότι προοριζόταν αποκλειστικά για tablets. Ακολόθησαν γρήγορα οι εκδόσεις 3.1 και 3.2 κατά τη διάρκεια του ίδιου έτους. Υπάρχει ένα νέο, εντελώς διαφορετικό, User Interface και υποστηρίζονται διπύρρηνοι και τετραπύρρηνοι επεξεργαστές. Ακόμα, έχει απλοποιηθεί το multitasking έτσι ώστε ο χρήστης να μπορεί με τη χρήση ενός πλήκτρου (recent apps) να περνάει από μια εφαρμογή σε άλλη. Υπάρχει η δυνατότητα για Video Chat μέσω της εφαρμογής Google Talk καθώς η ανάγνωση βιβλίων μέσω του Google eBooks. Επιπλέον, μπορούν να κρυπτογραφηθούν όλα τα δεδομένα του χρήστη.



Εικόνα 12: Android Honeycomb

1.3.8 Android 4.0 Ice Cream Sandwich

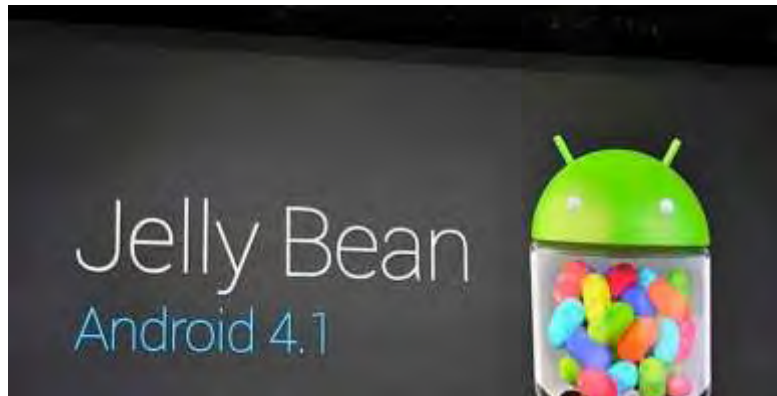
Κυκλοφόρησε στις 19 Οκτωβρίου του 2011. Μερικά κύρια καινούργια χαρακτηριστικά περιλάμβαναν ενσωματωμένο πρόγραμμα επεξεργασίας φωτογραφιών -1080p εγγραφή βίντεο, νέα καρτέλα στο Web με το Google Chrome , επιτρέποντας έως και 16 καρτέλες, Face Unlock, ένα χαρακτηριστικό που επιτρέπει στους χρήστες να ξεκλειδώσουν συσκευές που χρησιμοποιούν λογισμικό αναγνώρισης προσώπου και σύλληψη screenshot οθόνης.



Εικόνα 13: Android Ice Cream Sandwich

1.3.9 Android 4.1 Jelly Bean

Κυκλοφόρησε στις 9 Ιουλίου του 2012. Κάποια χαρακτηριστικά της έκδοσης αυτής ήταν ανανεωμένο σύστημα ειδοποιήσεων, δυνατότητα χρήσης εξωτερικής συσκευής USB ήχου, βελτιωμένη φωνητική αναζήτηση, δυνατότητα χρήσης της υπηρεσίας Google Wallet και OpenGL ES 3.0 υποστήριξη, που βελτιώνει τα γραφικά ενός παιχνιδιού.



Εικόνα 14: Android Jelly Bean

1.3.10 Android 4.4 Kit Kat

Κυκλοφόρησε στις 31 Οκτωβρίου του 2013. Κάποια χαρακτηριστικά της έκδοσης αυτής ήταν γρηγορότερο multitasking, υποστήριξη ασύρματης εκτύπωσης για τους εκτυπωτές που είναι συνδεδεμένοι στο Google Cloud Print και full screen artworks.



Εικόνα 15: Android Kit Kat

1.3.11 Android 5.0 Lollipop

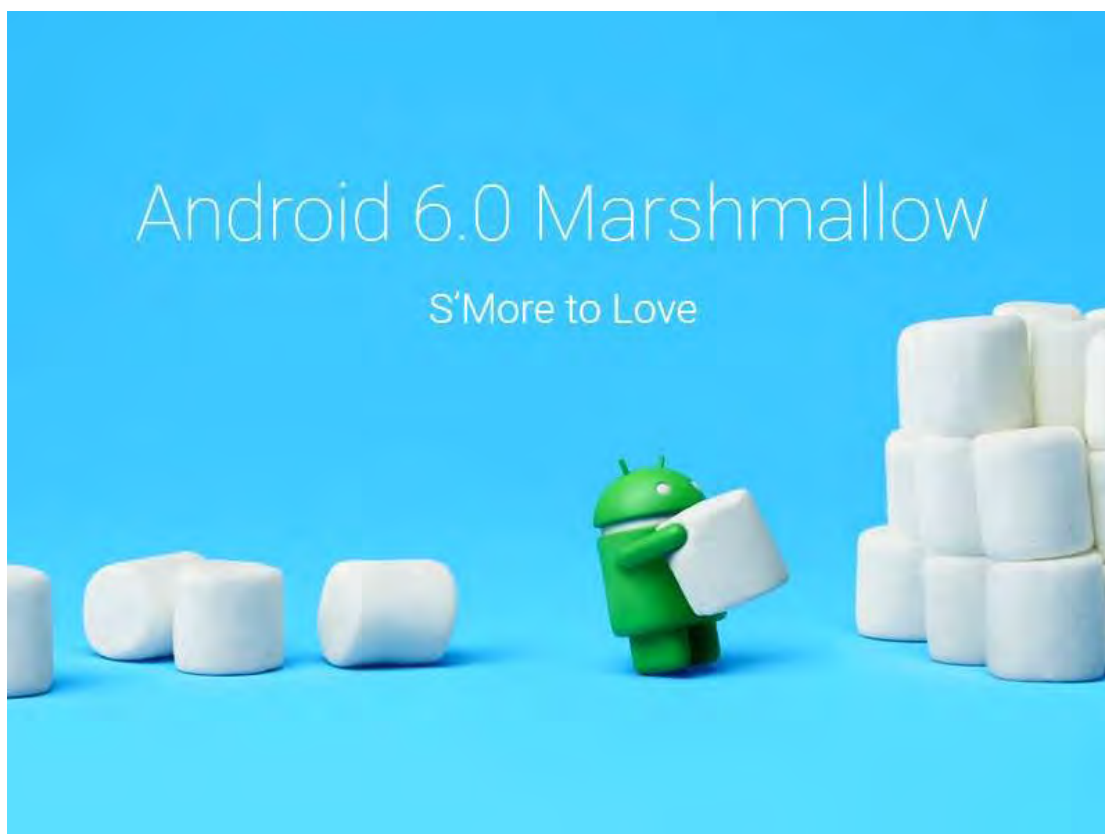
Κυκλοφόρησε τον Νοέμβριο του 2014. Το μόντο της νέα έκδοσης είναι "A sweet new take on Android" και μαζί του φέρνει καινούργιο design (Material Design) και πολλές και σημαντικές νέες λειτουργίες. Έγινε αρχικά διαθέσιμο από τα κινητά τηλέφωνα Nexus 6, Nexus 9 και Nexus Player.



Εικόνα 16: Android Lollipop

1.3.12 Android 6.0 Marshmallow

Κυκλοφόρησε τον Οκτώβριο του 2015 και αποτελεί την πιο πρόσφατη αναβάθμιση του Android. Η έκδοση αυτή επικεντρώνεται κυρίως στη βελτίωση της συνολικής εμπειρίας χρήστη της Lollipop, εισάγοντας μια νέα αρχιτεκτονική και ένα νέο σύστημα διαχείρισης ενέργειας που μειώνει την δραστηριότητα του φόντου όταν μια συσκευή δεν χρησιμοποιείται (physically).



Εικόνα 17: Android Marshmallow

1.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ANDROID

Τα βασικά χαρακτηριστικά και λειτουργίες του Android περιγράφονται παρακάτω:

1.4.1 Λειτουργίες Οθόνης

Η πλατφόρμα είναι προσαρμόσιμη σε πολλές ανάλυσεις οθόνης (από VGA μέχρι 4K), δισδιάστατες ψηφιακές γραφικές βιβλιοθήκες, τρισδιάστατα γραφικά βασισμένα στην OpenGL ES 3.0+ έκδοση χαρακτηριστικών, καθώς και παραδοσιακές απεικονίσεις οθόνης "έξυπνων" συσκευών κινητής τηλεφωνίας.

1.4.2 Αποθήκευση Δεδομένων

Χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης.

1.4.3 Συνδεσιμότητα

Το Android υποστηρίζει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένου GSM/EDGE, 3G, 4G, CDMA, EV-DO, UMTS, Bluetooth, NFC, και Wi-Fi.

1.4.4 Αποστολή Μηνυμάτων

SMS και MMS είναι οι διαθέσιμοι τρόποι ανταλλαγής μηνυμάτων.

1.4.5 Περιήγηση στον Ιστό

Για την περιήγηση στον ιστό το Android διαθέτει φυλλομετρητή βασισμένο στην ανοιχτή τεχνολογία WebKit. Και άλλοι φυλλομετρητές είναι διαθέσιμοι από το Google Play.

1.4.6 Υποστήριξη Java

Λογισμικό γραμμένο στην Java είναι δυνατόν να μεταγλωττιστεί και να εκτελεστεί στην εικονική μηχανή Dalvik, η οποία αποτελεί εξειδικευμένη υλοποίηση εικονικής μηχανής, σχεδιασμένης για χρήση σε φορητές συσκευές, παρόλο που δεν είναι πρότυπη εικονική μηχανή Java.

1.4.7 Υποστήριξη Πολυμέσων

Το λειτουργικό Android υποστηρίζει τις ακόλουθα 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, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP.

1.4.8 Επιπλέον Υποστήριξη Υλικού

Το λειτουργικό Android μπορεί να συνεργαστεί με κάμερες στατικής ή κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρα, δισδιάστατους καθώς και τρισδιάστατους επιταχυντές γραφικών.

1.4.9 Περιβάλλον Ανάπτυξης Λογισμικού

Περιλαμβάνει έναν προσομοιωτή συσκευής, εργαλεία για διόρθωση σφαλμάτων, μνήμη και εργαλεία ανάλυσης της απόδοσης του εκτελέσιμου λογισμικού καθώς και ένα επιπρόσθετο για το Eclipse IDE.

1.4.10 Αγορά και Εγκατάσταση Εφαρμογών

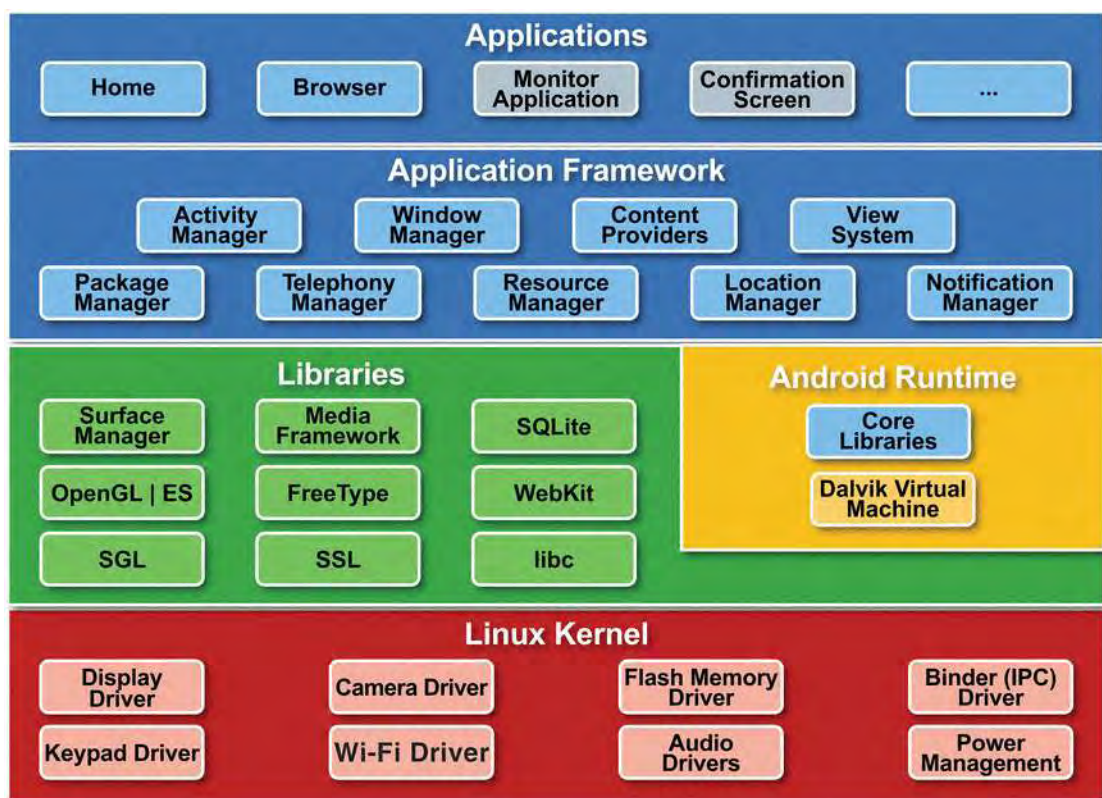
Παρόμοια με το App Store του iPhone OS, το Google Play είναι ένας κατάλογος εφαρμογών που μπορούν να μεταφορτωθούν και εγκατασταθούν στην συσκευή άμεσα μέσω ασύρματων καναλιών, χωρίς την χρήση υπολογιστή. Αρχικά μόνο δωρεάν εφαρμογές ήταν δυνατόν να εγκατασταθούν.



Εικόνα 18: Android Google Play

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ANDROID

Το Android αποτελείται από τον Kernel, ο οποίος βασίζεται στον αντίστοιχο kernel του Linux, το middleware, βιβλιοθήκες και APIs γραμμένα σε C και το Application Framework, το οποίο περιλαμβάνει βιβλιοθήκες συμβατές με Java. Το Android χρησιμοποιεί τη Dalvik virtual machine προκειμένου να τρέξει τα Dalvik-dex-code αρχεία (γνωστά και ως Dalvik Executables), τα οποία συνήθως παράγονται από τον δυαδικό κώδικα της Java. Η κύρια αρχιτεκτονική η οποία χρησιμοποιείται είναι η αρχιτεκτονική ARM, υπάρχει όμως και η υποστήριξη πλέον x86 και x64 όπως προαναφέραμε.



Εικόνα 19: Android Architecture

Πιο αναλυτικά, το Android αποτελείται από 4 επίπεδα και 5 ομάδες συνιστωσών, τα οποία παρουσιάζονται ακριβώς παρακάτω.

2.1 LINUX KERNEL

Το Android βασίζεται στον πυρήνα του Linux για βασικές λειτουργίες όπως η διαχείριση μνήμης, η διαχείριση διεργασιών, η διαχείριση των οδηγών της συσκευής και η διαχείριση της δικτύωσης, η οποία συνεπάγεται τη διαχείριση των Network Interfaces που έχει η κάθε συσκευή (GSM, Wi-Fi, Bluetooth, HSDPA, LTE, κτλ.).

Linux Kernel



- Android is built on the Linux kernel, but Android is not Linux
- No native windowing system
- No glibc support
- Does not include the full set of standard Linux utilities

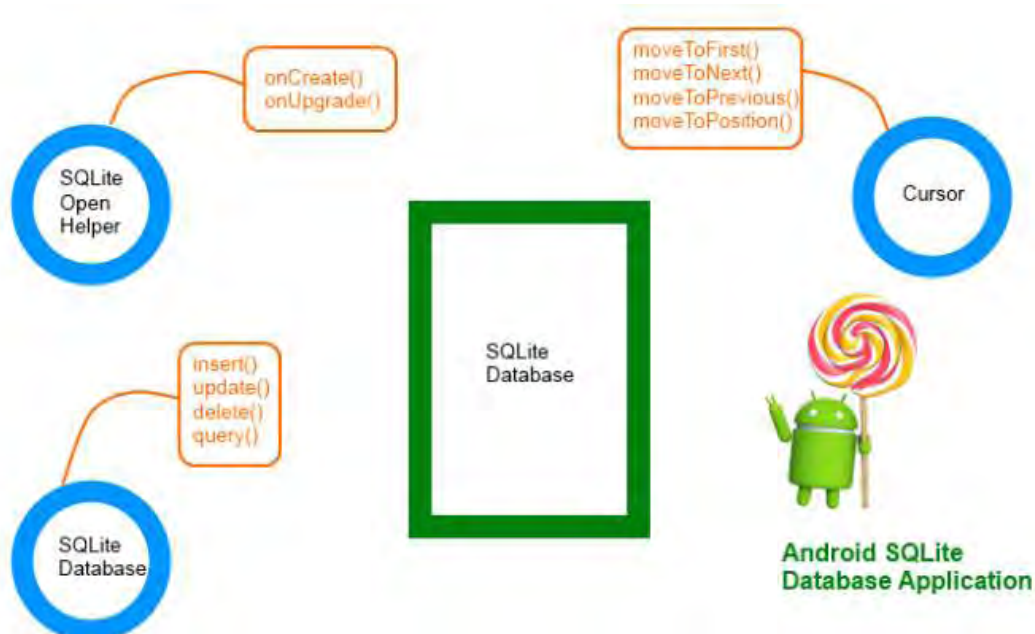


Εικόνα 20: Linux Kernel

2.2 NATIVE LIBRARIES

Οι βιβλιοθήκες του Android είναι γραμμένες σε C και C++ και είναι προσβάσιμες μέσω του κατάλληλου interface που παρέχεται από την Java. Κάποιες από τις κυριότερες είναι:

- **Surface Manager**, για τη δημιουργία παραθύρων και 2D ή 3D γραφικών.
- **Media Framework**, η οποία παρέχει αποκωδικοποιητές (codecs) για την αναπαραγωγή αρχείων πολυμέσων.
- **SQLite**, η οποία παρέχει τα κατάλληλα εργαλεία για την υποστήριξη της βάσης δεδομένων SQLite



Εικόνα 21: SQLite

- **Android WebKit**, η οποία περιέχει όλα τα κατάλληλα εργαλεία για την υποστήριξη των περιηγητών ιστού (Browsers)

2.3 ANDROID RUNTIME

Το Android Runtime αποτελείται από δύο βασικές συνιστώσες:

- **Βασικές βιβλιοθήκες** για την αλληλεπίδραση των εφαρμογών Java με το περιβάλλον της συσκευής στην οποία εκτελούνται
- **Dalvik virtual machine (DVK)**, η οποία είναι υπεύθυνη για τη δημιουργία των εκτελέσιμων αρχείων των εφαρμογών προκειμένου να μπορεί να εκκινήσει το λειτουργικό σύστημα.

Κάθε εφαρμογή του Android είναι γραμμένη σε γλώσσα Java, προγραμματισμού υψηλού επιπέδου και την οποία φυσικά ο υπολογιστής δεν είναι δυνατόν να αντιληφθεί. Για το λόγο αυτό η DMV αναλαμβάνει τη μετάφραση του κώδικα αυτού και τη δημιουργία των Dalvik executables (.dex), τα οποία μπορούν να εκτελεστούν από το λειτουργικό σύστημα. Κάθε τέτοιο εκτελέσιμο εκτελείται από τη δικιά του virtual machine ακόμα και όταν τα προγράμματα εκτελούνται παράλληλα, με συνέπεια το ένα πρόγραμμα να είναι ανεξάρτητο από το άλλο. Αυτό έχει ως αποτέλεσμα αν ένα πρόγραμμα παρουσιάσει κάποιο σφάλμα κατά την εκτέλεσή του, να μην προκαλέσει πρόβλημα σε κάποιο άλλο.

2.4 APPLICATION FRAMEWORK

Το Android προσφέρει μια ανοιχτή πλατφόρμα ανάπτυξης εφαρμογών, οι οποίες, για να μπορούν να είναι προχωρημένες και καινοτόμες, έχουν πρόσβαση στις βασικές βιβλιοθήκες του λειτουργικού συστήματος μέσω κατάλληλων διεπαφών. Παράλληλα, μέσα από το Application Framework μπορούν να παρέχουν με τη σειρά τους επιπλέον λειτουργίες και υπηρεσίες προς άλλες εφαρμογές, εφόσον δεν παραβιάζονται οι πολιτικές ασφαλείας του Framework.

Οι πιο βασικές οντότητες που περιλαμβάνονται σε αυτό είναι:

- **View System:** Επιτρέπει τη χρήση λιστών, πλαισίων, πεδίων κειμένου, κουμπιών, κλπ.
- **Content Providers:** Παρέχει στις εφαρμογές την πρόσβαση σε δεδομένα άλλων εφαρμογών ή τη δυνατότητα διαμοιρασμού των δικών τους δεδομένων.
- **Resource manager:** Επιτρέπει την πρόσβαση σε πόρους όπως τα γραφικά και σε αρχεία σχετικά με τη διάταξη των στοιχείων του γραφικού περιβάλλοντος.
- **Notification manager:** Διαχειρίζεται τα μηνύματα των εφαρμογών που εμφανίζονται στην Status bar, όπως τα εισερχόμενα μηνύματα, κλήσεις, ειδοποιήσεις συναντήσεων, κτλ.
- **Activity manager:** Είναι υπεύθυνο για τη διαχείριση του κύκλου ζωής των εφαρμογών και παρέχει τη δυνατότητα μετάβασης στις προγενέστερες καταστάσεις τους.

2.5 APPLICATIONS

Σε αυτή την ομάδα βρίσκονται οι εφαρμογές τις οποίες χρησιμοποιούν τελικά οι χρήστες με διαφανή τρόπο ως προς το τι συμβαίνει πίσω από αυτές ή το τι απαιτείται για την εκτέλεσή τους από το λειτουργικό σύστημα. Παραδείγματα τέτοιων εφαρμογών είναι οι εφαρμογές αποστολής και λήψης SMS, οι εφαρμογές e-mail, η προβολή χαρτών, οι RSS Readers, το ημερολόγιο, η πλοήγηση μέσω GPS, κτλ.

3.0 ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID

3.1 ΕΡΓΑΛΕΙΑ

Μέχρι πρότινος το βασικό εργαλείο για την ανάπτυξη μίας Android εφαρμογής ήταν το Eclipse, για το οποίο υπάρχουν επεκτάσεις όπως το Android Development Tools, το οποίο συνδέει το Eclipse με το Android SDK και όλες τις δυνατότητες του. Android SDK σημαίνει "Android Software Development Kit" και είναι επίσημο εργαλείο της Google για αυτούς που θέλουν να δημιουργήσουν στο Android.

Πλέον όμως τη σκυτάλη έχει πάρει το Android Studio, το οποίο και χρησιμοποιήθηκε για τη συγκεκριμένη εργασία. Αρχικά ανακοινώθηκε στις 16 Μαΐου του 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou και ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1. Τον Ιούνιο του 2014 ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8, ενώ η πρώτη σταθερή έκδοση (version 1.0) έγινε διαθέσιμη τον Δεκέμβρη του ίδιου χρόνου. Αυτή τη στιγμή η πιο πρόσφατη έκδοση είναι η 2.2.1.



Εικόνα 22: Android Studio Environment

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην Android πλατφόρμα. Είναι διαθέσιμο ελεύθερα μέσα από το επίσημο site του android, <https://developer.android.com/index.html>, με την άδεια Apache License

2.0. Βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για Android προγραμματισμό. Είναι διαθέσιμο για λειτουργικά συστήματα Windows, Mac OS X και Linux.

Το Android Studio περιλαμβάνει μια μεγάλη λίστα με εργαλεία ανάπτυξης για τη διευκόλυνση των προγραμματιστών. Μερικά ενδεικτικά εργαλεία είναι:

- Εργαλεία Debugging των εφαρμογών
- Βιβλιοθήκες
- Εξομοιωτής συσκευών (Android Virtual Machines)
- Documentation
- Δείγματα Κώδικα
- Tutorial

3.2 ΓΛΩΣΣΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή εφαρμογών Android είναι κυρίως η Java, μία από τις δημοφιλέστερες αντικειμενοστραφείς γλώσσες προγραμματισμού στον κόσμο. Τα σημαντικότερα αρχεία μίας εφαρμογής Android είναι τα .java, τα όποια όπως αποκαλύπτει και η κατάληξη τους είναι γραμμένα σε γλώσσα Java και είναι υπεύθυνα για το λειτουργικό κομμάτι της εφαρμογής.

Πολύ βασικό ρόλο έχουν και τα αρχεία .xml, τα οποία είναι γραμμένα σε γλώσσα XML (Extensible Markup Language). Αυτά τα αρχεία είναι υπεύθυνα για το γραφικό περιβάλλον που βλέπει ο χρήστης. Κάθε διαφορετική οθόνη που παρουσιάζεται στον χρήστη είναι και ένα διαφορετικό αρχείο .xml συνοδευόμενο από ένα αρχείο .java.

Για τη διαχείριση μίας βάσης δεδομένων, που χρειάστηκε στη συγκεκριμένη εφαρμογή, χρησιμοποιούνται εντολές SQL στον SQL Database Browser. Αυτά τα αρχεία έχουν την κατάλη .db.

Μέσω του εργαλείου NDK (Native Development Kit) επιτρέπεται η μεταγλώττιση κάποιων μερών μίας Android εφαρμογής χρησιμοποιώντας native γλώσσες προγραμματισμού όπως η C και η C++.

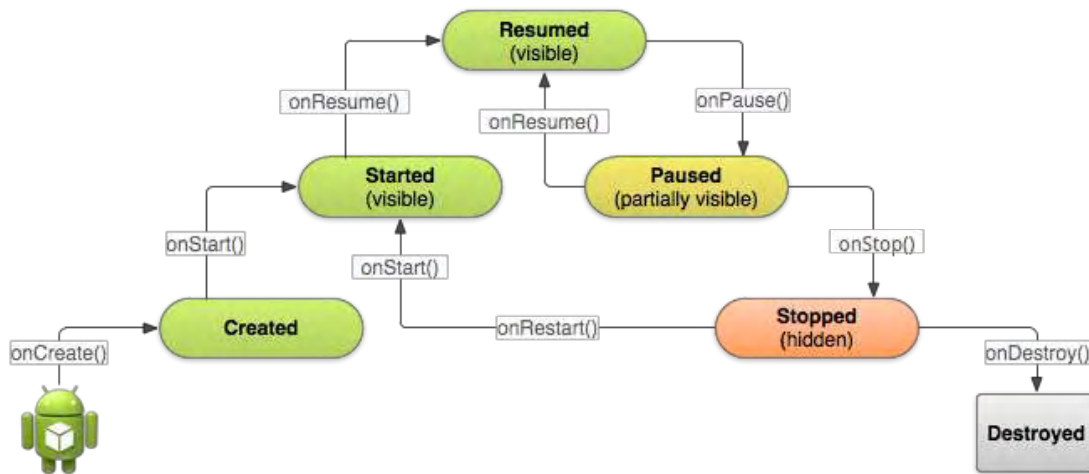
3.3 ΚΥΚΛΟΣ ΖΩΗΣ ΕΦΑΡΜΟΓΗΣ ANDROID

Σε αντίθεση με άλλα πρότυπα προγραμματισμού στα οποία οι εφαρμογές ξεκινούν με μία κλήση της μεθόδου `main()`, το σύστημα Android το σύστημα αρχίζει τον κώδικα σε μία δραστηριότητα (Activity) με την κλήση συγκεκριμένων μεθόδων επανάκλησης (callback), που αντιστοιχούν στα συγκεκριμένα στάδια της ζωής της. Υπάρχει μία σειρά από μεθόδους επανάκλησης που ξεκινούν μια δραστηριότητα και μια σειρά από μεθόδους επανάκλησης που καταστρέφουν μια δραστηριότητα.

Κατά τη διάρκεια της ζωής μίας δραστηριότητας, το σύστημα καλεί ένα σύνολο μεθόδων κύκλου ζωής σε μία ακολουθία παρόμοια με μία πυραμίδα βημάτων. Δηλαδή κάθε στάδιο του κύκλου ζωής της δραστηριότητας είναι ένα ξεχωριστό βήμα στην πυραμίδα. Δεδομένου ότι το σύστημα δημιουργεί μία νέα περίπτωση δραστηριότητας, κάθε μέθοδος επανάκλησης μετακινεί την κατάσταση της δραστηριότητας ένα βήμα προς την κορυφή. Η κορυφή της πυραμίδας είναι το σημείο στο οποίο η δραστηριότητα τρέχει στο πρώτο πλάνο και ο χρήστης μπορεί να αλληλεπιδράσει μαζί της.

Δεδομένου ότι ο χρήστης αρχίζει να αφήνει τη δραστηριότητα, το σύστημα καλεί άλλες μεθόδους που μετακινούν την κατάσταση της δραστηριότητας ένα βήμα προς τα κάτω στην πυραμίδα προκειμένου να αποσυναρμολογηθεί η δραστηριότητα. Σε ορισμένες περιπτώσεις η δραστηριότητα θα κινηθεί μερικώς κάτω από την πυραμίδα και θα περιμένει (όπως όταν ο χρήστης μεταπηδά σε μία άλλη εφαρμογή), από όπου η δραστηριότητα μπορεί να κινηθεί πίσω στην κορυφή (εάν ο χρήστης επιστρέψει στη δραστηριότητα) και να συνεχίσει από εκεί όπου ο χρήστης έφυγε.

Παρακάτω παρουσιάζονται οι μέθοδοι του κύκλου ζωής μίας εφαρμογής:



Εικόνα 23: Android Lifecycle Methods

- **OnCreate ():** Μόλις το σύστημα καλέσει την `onCreate()` η μέθοδος αυτή καλεί αμέσως την `onStart()`.
- **OnStart():** Μόλις το σύστημα καλέσει την `onStart()` η μέθοδος αυτή καλεί αμέσως την `onResume()`.
- **OnResume():** Σε αυτή τη κατάσταση η δραστηριότητα τρέχει στο πρώτο πλάνο και ο χρήστης μπορεί να αλληλεπιδράσει μαζί της.
- **OnPause ():** Σε αυτή τη κατάσταση η δραστηριότητα κρύβεται μερικώς από άλλη δραστηριότητα που στο πρώτο πλάνο είναι ημιδιάφανη ή δεν καλύπτει ολόκληρη την οθόνη. Η σταματημένη δραστηριότητα δεν λαμβάνει δεδομένα εισόδου και δεν μπορεί να εκτελέσει οποιοδήποτε κώδικα.
- **OnStop ():** Σε αυτή τη κατάσταση η δραστηριότητα είναι εντελώς κρυμμένη και μη ορατή στο χρήστη και θεωρείται ότι είναι στο υπόβαθρο. Όταν η δραστηριότητα είναι σταματημένη όλες οι πληροφορίες κατάστασης όπως οι μεταβλητές διατηρούνται αλλά δεν μπορεί να εκτελέσει τον οποιοδήποτε κώδικα.
- **OnDestroy ():** Το σύστημα καλεί αυτή τη μέθοδο για τη δραστηριότητά, ως το τελικό σήμα όταν η δραστηριότητα πρόκειται να αφαιρεθεί τελείως από τη μνήμη του συστήματος.

4 ΔΟΜΗ ΕΦΑΡΜΟΓΗΣ ANDROID

Η δομή μίας εφαρμογής Android βασίζεται σε τέσσερα διαφορετικά στοιχεία (δομικά blocks) τα οποία είναι:

- **Activity**
- **Intent Receiver**
- **Service**
- **Content Provider**

Τα συστατικά αυτά δεν είναι απαραίτητα ως σύνολο για την ανάπτυξη μίας εφαρμογής, αλλά ένας συνδυασμό αυτών και τουλάχιστον το στοιχείο Activity προκειμένου η εφαρμογή να περιέχει ένα γραφικό περιβάλλον χρήσης.

Οι εφαρμογές μπορούν να ξεκινούν άλλες εφαρμογές ή συγκεκριμένα στοιχεία άλλων εφαρμογών στέλνοντας μία προσθήκη (Intent). Αυτές περιέχουν μεταξύ άλλων και το όνομα της επιθυμητής δράσης που εκτελείται. Ο διαχειριστής προσθήκης (IntentManager) υπολογίζει τις εισερχόμενες προσθήκες και ξεκινάει τις κατάλληλες εφαρμογές ή τα κατάλληλα στοιχεία εφαρμογών. Τα στοιχεία Services και Broadcast Receivers επιτρέπουν στην εφαρμογή να εκτελεί διάφορες εργασίες στο παρασκήνιο της εφαρμογής, προσδίδοντας πρόσθετη μεταβλητότητα στα υπόλοιπα στοιχεία. Τα στοιχεία Broadcast Receivers μπορούν να ελεγχθούν από συγκεκριμένα γεγονότα και να λειτουργούν για ένα πολύ μικρό χρονικό διάστημα. Τέλος ο κώδικας από κάθε στοιχείο μιας εφαρμογής μαζί με πρόσθετες πηγές όπως βιβλιοθήκες, εικόνες και άλλα απαραίτητα δεδομένα, συνιστούν ένα απλό αρχείο .apk το οποίο αποτελεί τον τύπο αρχείου για μία εκτελέσιμη εφαρμογή Android.

4.1 ANDROID MANIFEST

Όλες οι εφαρμογές Android χρειάζονται να έχουν ένα αρχείο XML, στη διεύθυνση root της εφαρμογής, το οποίο ονομάζεται AndroidManifest.xml και δημιουργείται αυτόματα όταν ξεκινάμε καινούργιο project ενός android application. Στο αρχείο αυτό, δηλώνονται τα δομικά blocks που θα χρησιμοποιηθούν και ποιες δυνατότητες και προδιαγραφές θα εξυπηρετούν. Το αρχείο αυτό περιέχει βασικές πληροφορίες σχετικά με την εφαρμογή, τις οποίες το λειτουργικό σύστημα πρέπει να γνωρίζει προτού

τρέξει οποιοδήποτε άλλο κώδικα. Οι σημαντικότερες από αυτές τις πληροφορίες περιγράφονται παρακάτω:

- Η ονομασία του πακέτου της Java της εφαρμογής (Java package).
- Η έκδοση της εφαρμογής (π.χ. 2.2, 3.0, 4.2.2 κλπ).
- Η ελάχιστη έκδοση του λειτουργικού συστήματος Android που απαιτεί η εφαρμογή (min sdk version). Για παράδειγμα αν έχει δηλωθεί min sdk version ο αριθμός 7, που ισοδυναμεί με την έκδοση Android 2.1, τότε η εφαρμογή θα μπορεί εκτελεστεί σε συσκευές με έκδοση Android μεγαλύτερη ή ίση της 2.1.
- Το όνομα της εφαρμογής, καθώς και το εικονίδιό της.
- Οι άδειες που απαιτούνται για να εκτελεστούν ορισμένες λειτουργίες της εφαρμογής (δίκτυο, εξωτερική κάρτα μνήμης, GPS, κάμερα).

4.2 ACTIVITY

Μία Activity (δραστηριότητα) είναι ένα συστατικό της εφαρμογής το οποίο παρέχει τη διεπαφή με την οποία αλληλεπιδρά ο χρήστης ώστε να κάνει κάποιες ενέργειες. Είναι μία απλή οθόνη μιας εφαρμογής, όπως ένα παράθυρο περιαγωγής ή μία οθόνη ρυθμίσεων. Περιέχει τα οπτικά στοιχεία είτε αυτά παρουσιάζουν κάποια δεδομένα, όπως μία εικόνα, είτε επιτρέπουν την αλληλεπίδραση με τον χρήστη, όπως ένα πλήκτρο. Κάθε οθόνη που βλέπει και αλληλεπιδρά ο χρήστης είναι μια κλάση που κληρονομεί την κλάση activity από το android.app.Activity. Μια εφαρμογή συνήθως αποτελείται από πολλές activities που συνδέονται μεταξύ τους, η μεταφορά μεταξύ αυτών των δραστηριοτήτων γίνεται μέσω του intent. Το πιο κοινό σενάριο είναι να υπάρχει μια κεντρική activity, η οποία παρουσιάζεται στο χρήστη κατά την εκτέλεση της εφαρμογής. Κάθε activity μπορεί να ξεκινήσει μια άλλη ώστε να εκτελέσει διαφορετικές ενέργειες. Κάθε φορά που μια νέα activity ξεκινά, η προηγούμενη σταματάει, αλλά το σύστημα την βάζει σε μια στοίβα. Έτσι όταν ο χρήστης ολοκληρώσει τις ενέργειες του σε αυτή την activity και πατήσει το πλήκτρο επιστροφής, η προηγούμενη activity θα επαναφερθεί από τη στοίβα και θα συνεχίσει την εκτέλεσή της.

Κάθε Activity έχει ένα συγκεκριμένο κύκλο ζωής και κάποιες καταστάσεις, όπως για παράδειγμα Resumed, Paused και Stopped. Όταν μια Activity

περνάει σε μια κατάσταση αυτό αναγνωρίζεται από κάποιες callback methods.

4.3 INTENT RECEIVER

Ένας Intent Receiver χρειάζεται όταν ο προγραμματιστής της εφαρμογής θέλει να χρησιμοποιήσει κώδικα μέσα στην εφαρμογή του, που θα εκτελείται όταν συμβαίνει ένα εξωτερικό γεγονός, για παράδειγμα όταν χτυπά το τηλέφωνο ή όταν ένα ασύρματο δίκτυο γίνεται διαθέσιμο. Οι Intent Receivers δεν προβάλλουν κάποιο interface χρήστη, αλλά προβάλλουν Notifications για να ειδοποιήσουν τον χρήστη, εάν κάτι σημαντικό λαμβάνει χώρα. Οι Intent Receivers δηλώνονται και αυτοί στο AndroidManifest.xml.

4.4 SERVICES

Ένα Service είναι τμήμα κώδικα που εκτελείται χωρίς κάποιο interface χρήστη. Ένα καλό παράδειγμα service είναι ο media player που παίζει τραγούδια από μια λίστα. Σε μια εφαρμογή media player, είναι λογικό να υπάρχουν διάφορες οθόνες, άρα και πολλές Activities, όπου ο χρήστης θα μπορεί να επιλέξει τραγούδια και να τα ακούσει. Παρόλα αυτά το playback δε θα πρέπει να χειρίζεται από μια activity, γιατί ο χρήστης περιμένει να μπορεί να περιηγείται στη λίστα τραγουδιών του, χωρίς το τραγούδι που ακούει εκείνη τη στιγμή να σταματήσει. Σε αυτή την περίπτωση, η κύρια Activity του media player θα ξεκινήσει να εκτελεί ένα Service στο background, οπότε ο χρήστης θα μπορεί να κάνει pause, rewind κλπ.

4.5 CONTENT PROVIDERS

Οι εφαρμογές μπορούν να αποθηκεύσουν τα δεδομένα τους σε αρχεία, στη βάση δεδομένων SQLite, σε preferences ή χρησιμοποιώντας οποιονδήποτε άλλο μηχανισμό που τους παρέχει αυτή τη δυνατότητα. Ένας Content Provider επίσης είναι χρήσιμος εάν θέλουμε τα δεδομένα μιας εφαρμογής να γίνουν διαθέσιμα και σε άλλες εφαρμογές. Ένας content provider είναι μια κλάση που υλοποιεί ένα standard set από μεθόδους, οι οποίες επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να ανακτούν τον τύπο δεδομένων που χειρίζεται ο Content Provider.

4.6 RESOURCES

Στα Resources μιας εφαρμογής ορίζεται το layout των activities, οι διάφορες εικόνες και λεκτικά που χρησιμοποιούνται στα Activities. Σε κάθε Activity αντιστοιχεί ένα Layout αρχείο, το οποίο περιγράφει τη θέση των διάφορων αντικειμένων στην οθόνη. Το Layout αρχείο είναι ένα αρχείο XML. Στην πράξη το αρχείο αυτό διαμορφώνεται από κατάλληλους γραφικούς editors που προσφέρονται από ολοκληρωμένα περιβάλλοντα ανάπτυξης όπως το Android Studio.

4.7 USER INTERFACE

Η διεπαφή χρήστη έχει τεράστια σημασία για κάθε εφαρμογή. Αποτελεί την τελική εικόνα που βλέπει ο χρήστης, το γραφικό περιβάλλον στο οποίο θα περιηγηθεί, και ενεργοποιεί όλες τις λειτουργίες της εφαρμογής.

Το user interface και η λειτουργικότητα της εφαρμογής αποτελούν αλληλένδετα στοιχεία και χωρίς το ένα δε μπορεί να υπάρξει το άλλο. Πολλές φορές μάλιστα είναι δυσκολότερος ο σχεδιασμός ενός όμορφου και εύχρηστου περιβάλλοντος εργασίας, παρά η ίδια η λειτουργικότητα της εφαρμογής. Καθίσταται σαφές, λοιπόν, ότι απαιτεί μεγάλη προσπάθεια και προσοχή η δημιουργία ενός γραφικού περιβάλλοντος που θα προσελκύει τους χρήστες και θα τους ωθεί να χρησιμοποιούν μία συγκεκριμένη εφαρμογή έναντι μίας άλλης, με τις ίδιες λειτουργίες.

4.7.1 Layout

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

Υπάρχουν τέσσερα είδη layout, τα LinearLayout (γραμμική διάταξη), RelativeLayout (σχετική διάταξη), FrameLayout (διάταξη πλαισίου) και TableLayout (διάταξη πίνακα). (Υπάρχει και το AbsoluteLayout, το οποίο όμως έχει προταθεί να μην χρησιμοποιείται πλέον, διότι ορίζει τις απόλυτες θέσεις κάθε στοιχείου, οι οποίες όμως διαφέρουν ανάλογα με το κινητό τηλέφωνο και την οθόνη που χρησιμοποιείται η εφαρμογή).

Υπάρχουν δύο τρόποι για να δηλώσει κανείς τα Layouts (όπως και κάθε άλλο γραφικό στοιχείο) της εφαρμογής: α) μέσω αρχείων xml ή β) μέσα στις Activities της εφαρμογής. Τα αρχεία xml αποτελούν στατικό τρόπο δημιουργίας των γραφικών στοιχείων. Αποθηκεύονται σε συγκεκριμένο φάκελο του project και καλούνται για τη δημιουργία του γραφικού περιβάλλοντος μέσα από τις activities.

Πολλές φορές ωστόσο δε γνωρίζουμε εξ αρχής τη διάταξη που θα χρησιμοποιήσουμε, διότι ίσως να εξαρτάται από ορισμένες επιλογές του χρήστη. Στις περιπτώσεις αυτές δημιουργούμε δυναμικά τα Layouts μέσα στις Activities και ορίζουμε εκεί τις παραμέτρους αυτών. Ωστόσο ο πιο εύκολος και συνηθέστερος τρόπος είναι (αν έχουμε τη δυνατότητα) να δημιουργήσουμε για κάθε οθόνη ένα διαφορετικό xml αρχείο που θα περιλαμβάνει τη διάταξη όλων των γραφικών της στοιχείων, και να το καλέσουμε μέσα από την αντίστοιχη Activity.

4.7.2 Menu

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

4.7.3 Dialogs

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

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

4.7.4 Notifications

Σε πολλές περιπτώσεις θέλουμε να ενημερώσουμε το χρήστη για κάποιο γεγονός ή αποτέλεσμα σχετικό με την εφαρμογή μας. Ορισμένα από αυτά τα γεγονότα απαιτούν κάποια απάντηση από το χρήστη και άλλα όχι.

- **Toast Notification**

Το Toast Notification είναι ένα μήνυμα που εμφανίζεται για λίγα δευτερόλεπτα στο παράθυρο που βρίσκεται ο χρήστης, οποιασδήποτε εφαρμογής και αν είναι αυτό. Ο χώρος που καταλαμβάνει είναι ο ελάχιστος απαιτούμενος ώστε το μήνυμα να είναι εμφανές, ενώ ο χρήστης μπορεί όσο εμφανίζεται το μήνυμα, να αλληλεπιδρά με την activity στην οποία βρίσκεται. Χρησιμοποιείται συνήθως για μικρά μηνύματα που δεν απαιτούν κάποια ενέργεια από το χρήστη, όπως για παράδειγμα “Το αρχείο αποθηκεύτηκε επιτυχώς”, “Το ξυπνητήρι ορίστηκε στις ...” κλπ.

- **Status Bar Notification**

Το Status Bar Notification, όπως φανερώνει και το όνομά της, είναι μία ειδοποίηση η οποία εμφανίζεται στη Status Bar του κινητού τηλεφώνου μας, και την οποία μπορούμε να ανοίξουμε είτε βρισκόμαστε στο κεντρικό μενού του τηλεφώνου μας, είτε σε κάποια εφαρμογή. Αντίθετα με την Toast, η Status Bar Notification μπορεί να επιλεχθεί και να ξεκινήσει κάποια λειτουργία ανάλογα με τις ενέργειες που έχουμε ορίσει στον κώδικα της εφαρμογής. Για παράδειγμα, όταν κατεβάζουμε ένα αρχείο από το διαδίκτυο, όταν η λήψη ολοκληρωθεί, θα θέλαμε να επιλέξουμε την ειδοποίηση αυτή και με τον τρόπο αυτό είτε να ανοίξουμε το φάκελο που βρίσκεται το αρχείο, είτε να το τρέξουμε. Τις περισσότερες φορές οι Toast Notifications ενεργοποιούνται από Activities, ενώ οι Status Bar Notifications από Services.

5 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ BOOKI

Σε αυτό το κεφάλαιο θα γίνει μία παρουσίαση των λειτουργιών της εφαρμογής που υλοποιήθηκε, Booki, μέσα από κάποια screenshots.

5.1 VERIFICATION ΧΡΗΣΤΗ (Account Kit)

Το Account Kit είναι ένα εργαλείο που επιτρέπει στους χρήστες να εγγραφούν γρήγορα και να συνδεθούν στην εφαρμογή, χρησιμοποιώντας μόνο τον αριθμό τηλεφώνου τους ή τη διεύθυνση ηλεκτρονικού ταχυδρομείου - δεν απαιτείται κωδικός πρόσβασης. Είναι αξιόπιστο, εύκολο στη χρήση και τους δίνει μια επιλογή για το πώς μπορούν να εγγραφούν στην εφαρμογή. Το Account Kit ζητάει αρχικά από το χρήστη τον αριθμό τηλεφώνου του,



Εικόνα 24: Account Kit Overview

αφού ο χρήστης πληκτρολογήσει το αριθμό του και πατήσει το πλήκτρο “επόμενο” έρχεται στο κινητό του μέσω SMS (χωρίς χρέωση) ένας 6ψήφιος κωδικός, ώστε να τον πληκτρολογήσει ο χρήστης και μπει στην εφαρμογή.

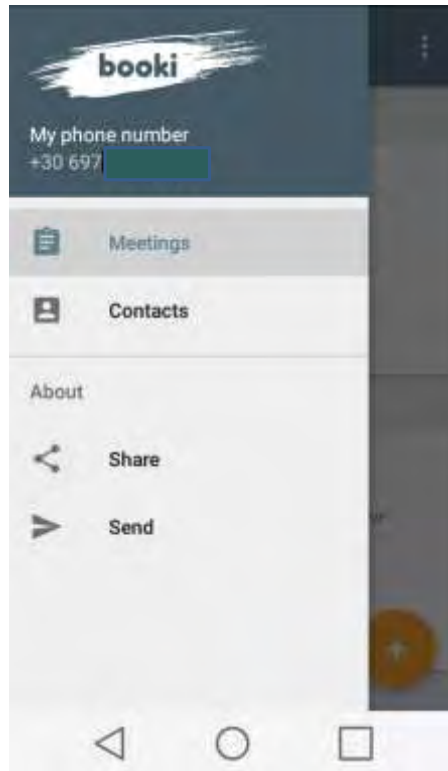


Εικόνα 25: Account Kit overview

Με τον εργαλείο αυτό εξασφαλίζουμε στον χρήστη ασφαλή περιήγηση στην εφαρμογή χωρίς τον φόβο να μπορεί κάποιος τρίτος να χρησιμοποιήσει τον λογαριασμό του εναγνοία του.

5.2 ΑΡΧΙΚΗ ΟΘΟΝΗ

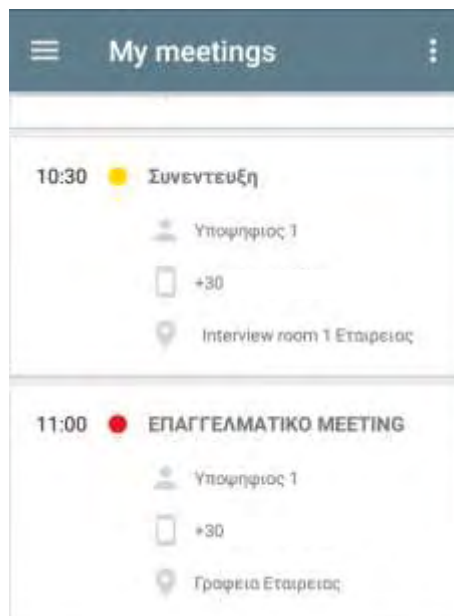
Στην αρχική οθόνη εμφανίζεται το μενού της εφαρμογής με τα διάφορα κουμπιά που προσφέρουν όλες τις δυνατές λειτουργίες της εφαρμογής.




Εικόνα 26: Αρχική Οθόνη

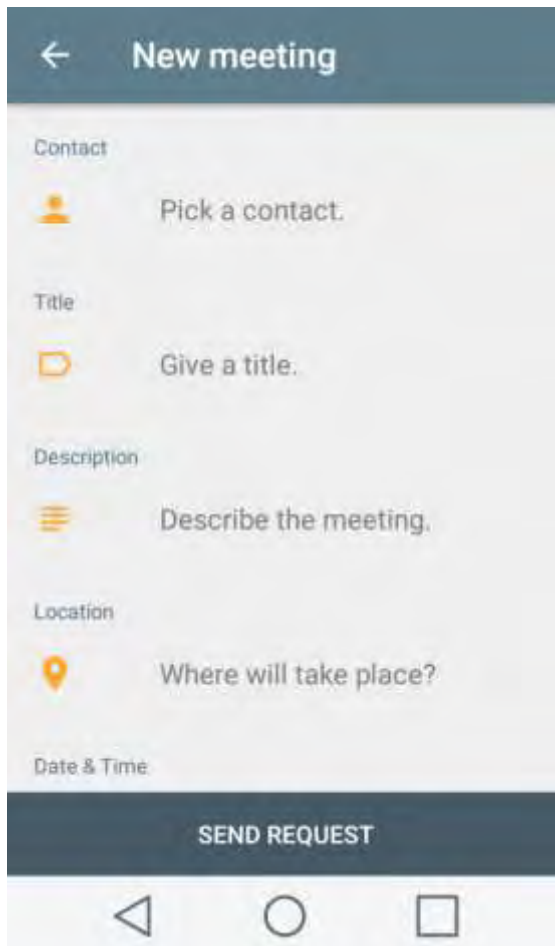
5.3 MEETINGS

Πατώντας το κουμπί “Meetings” ο χρήστης μπαίνει στην Οθόνη με το όλο το Ιστορικό, καθώς και όλα τα ανοιχτά ραντεβού/συναντήσεις του χρήστη.



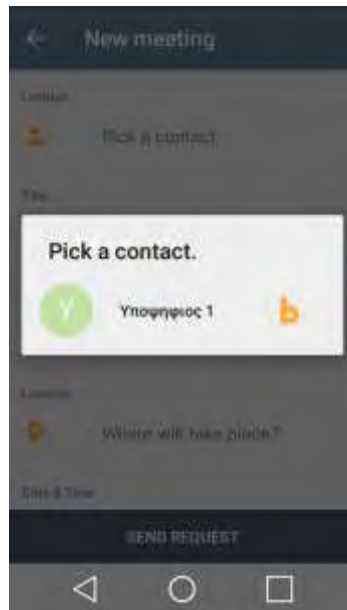
5.4 NEW MEETING

Πατώντας ο χρήστης το κουμπί  , μπαίνει στην οθόνη με τις ρυθμίσεις ώστε να διαμορφώσει και να στείλει ένα καινούργιο αίτημα συνάντησης σε κάποια από τις επαφές του.

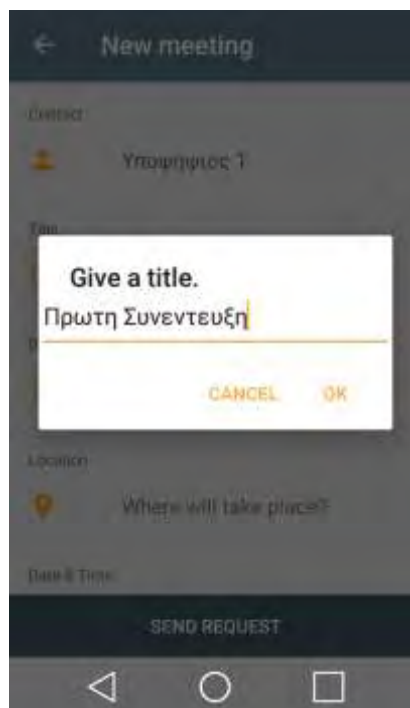


Η Οθόνη “New Meeting” αποτελείται από τα ακόλουθα πεδία που χρειάζεται ο χρήστης να συμπληρώσει και να στείλει το αίτημα σε μια από τις επαφές του.

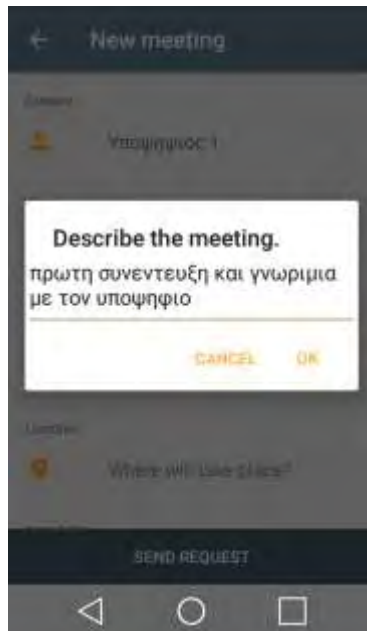
- **Contact:** Στο πεδίο αυτό ο χρήστης επιλέγει την επαφή που θέλει να στείλει το αίτημα



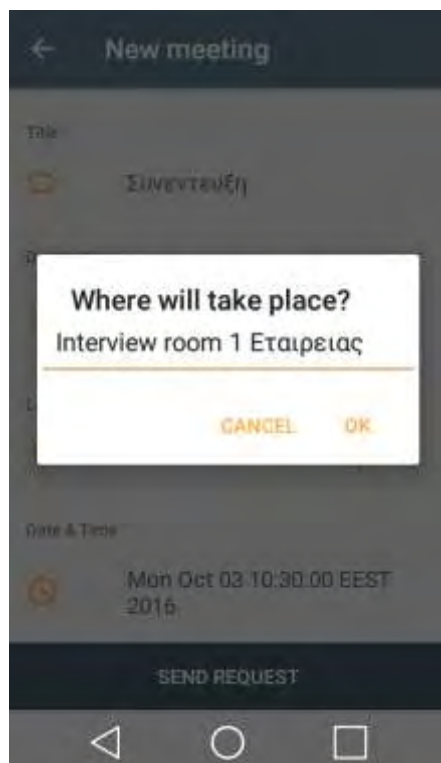
- **Title:** Στο πεδίο αυτό ο χρήστης δίνει τον Τίτλο που θα έχει η συνάντηση



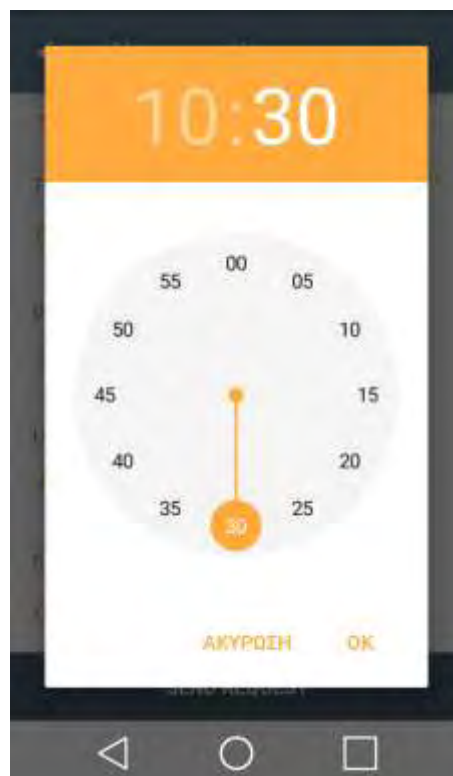
- **Description** Στο πεδίο αυτό ο χρήστης δίνει μια μικρή περιγραφή για ποιόν λόγο θέλει να κάνει την συνάντηση με το συγκεκριμένο άτομο



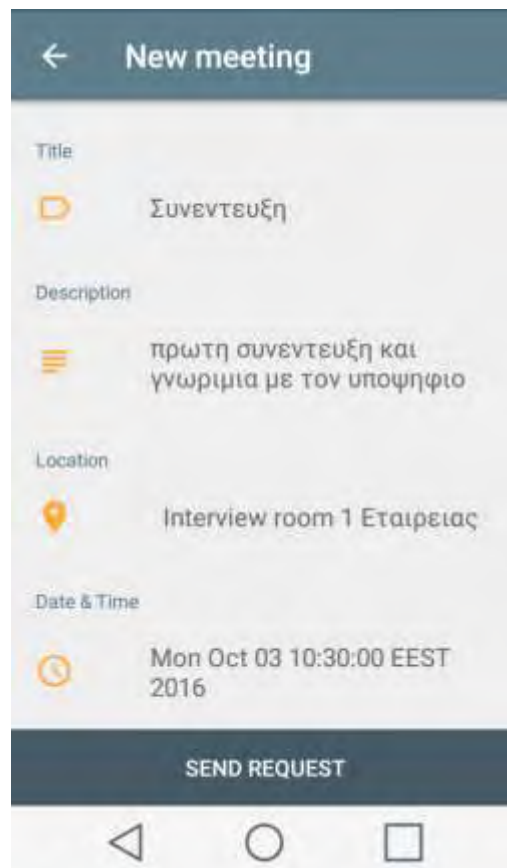
- **Location** Στο πεδίο αυτό ο χρήστης επιλέγει την τοποθεσία που θα γίνει η συνάντηση



- **Date & Time** Στο πεδίο αυτό ο χρήστης επιλέγει την ημέρα και ώρα που θα γίνει η συνάντηση



Αφού ο χρήστης συμπληρώσει τα παραπάνω πεδία είναι έτοιμος να στείλει το αίτημα

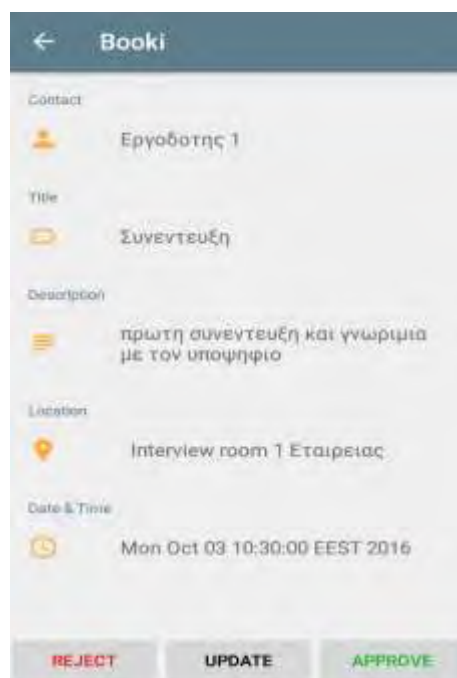


The screenshot shows a mobile application interface for creating a new meeting. The title bar at the top is dark blue with a white back arrow and the text "New meeting". Below the title bar, there are several form fields:

- Title:** Contains the text "Συνέντευξη" (Interview).
- Description:** Contains the text "πρωτη συνέντευξη και γνωριμια με τον υποψηφιο" (first interview and acquaintance with the candidate).
- Location:** Contains the text "Interview room 1 Εταιρειας" (Interview room 1 of the company).
- Date & Time:** Contains the text "Mon Oct 03 10:30:00 EEST 2016".

At the bottom of the form, there is a dark blue button with the text "SEND REQUEST". The Android navigation bar is visible at the very bottom.

Από την στιγμή που το αίτημα στάλθηκε, ο χρήστης στον οποίο το αίτημα προορίζεται, ειδοποιείται με ένα notification ώστε να δει το request.



The screenshot shows a mobile application interface for a notification. The title bar at the top is dark blue with a white back arrow and the text "Booki". Below the title bar, there are several form fields:

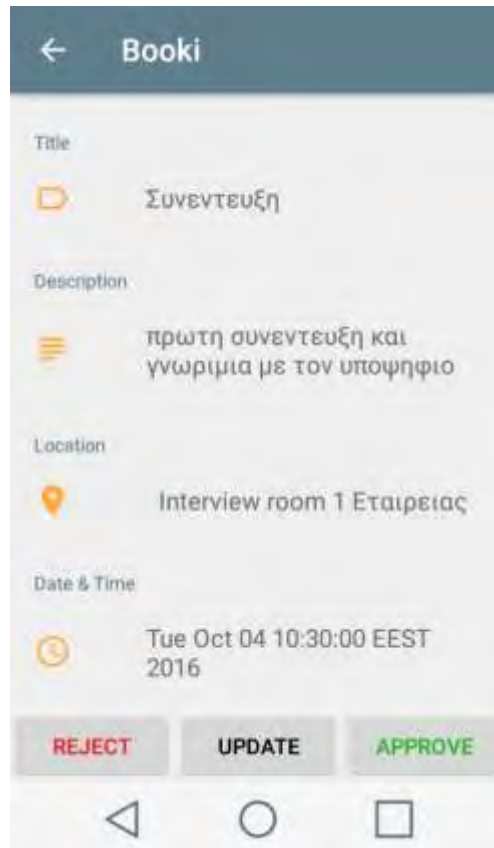
- Contact:** Contains the text "Εργοδοτης 1" (Employer 1).
- Title:** Contains the text "Συνέντευξη" (Interview).
- Description:** Contains the text "πρωτη συνέντευξη και γνωριμια με τον υποψηφιο" (first interview and acquaintance with the candidate).
- Location:** Contains the text "Interview room 1 Εταιρειας" (Interview room 1 of the company).
- Date & Time:** Contains the text "Mon Oct 03 10:30:00 EEST 2016".

At the bottom of the notification, there are three buttons: "REJECT" (red text), "UPDATE" (black text), and "APPROVE" (green text).

Η εφαρμογή αυτή δίνει την δυνατότητα στον χρήστη μέσω των τριών επιλογών (Reject, Update & Approve) "επέμβει" πάνω στο αίτημα. Με τις επιλογές Reject & Approve ο χρήστης απορρίπτει και αποδέχεται ανάλογα το αίτημα, και στέλνεται η αντίστοιχη ειδοποίηση στο αρχικό χρήστη. Με την επιλογή "Update" ο χρήστης μπορεί να τροποποιήσει κάποια από τα πεδία π.χ Date & Time και να στείλει ένα ανανεωμένο αίτημα συνάντησης στον αρχικό χρήστη.

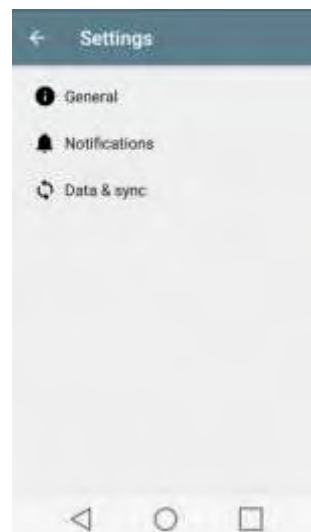


Η ίδια διαδικασία ακολουθείται πλέον από τον αρχικό χρήστη στον οποίο στάλθηκε το ανανεωμένο αίτημα.



5.5 SETTINGS

Στην οθόνη με τις ρυθμίσεις μπορεί ο χρήστης να αλλάξει τις υπάρχουσες ρυθμίσεις της εφαρμογής, π.χ. τον τύπο/ήχο των notifications με τα οποία ειδοποιείται.



6 ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

booki/booki/BookiApplication.java

```
package booki.booki;

import android.app.Application;

import com.facebook.accountkit.AccountKit;

public class BookiApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        AccountKit.initialize(getApplicationContext());
        Contacts.init(getBaseContext());
    }
}
```

booki/booki/Contact.java

```
package booki.booki;

import java.util.ArrayList;
import java.util.List;

public class Contact {

    public String name;

    public List<String> phones;

    public String dbNumber;

    protected boolean _member = false;

    public Contact(String name) {
        this.name = name;
        this.phones = new ArrayList();
    }

    public void addPhone(String number) {
        phones.add(number);
    }

    public void markAsMember() {
        _member = true;
    }

    public boolean isMember() {
        return _member;
    }
}
```

booki/booki/Contacts.java

```
package booki.booki;

import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.provider.ContactsContract;
import android.support.v4.util.LongSparseArray;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import org.apache.commons.collections4.CollectionUtils;
import org.apache.commons.collections4.Predicate;

import java.util.ArrayList;
import java.util.List;

public class Contacts {

    public static ContactsFragment.ContactAdapter adapter;

    private static DatabaseReference _database;

    private static LongSparseArray<String> _dbNumbers;

    protected static Context _context;

    protected static List<Contact> _contacts = new ArrayList();
    protected static List<Contact> _members = new ArrayList();

    public static void init(Context context) {
        _context = context;

        _database = FirebaseDatabase.getInstance().getReference();

        _dbNumbers = new LongSparseArray();

        _database.child("phones").addValueEventListener(
            new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    _dbNumbers.clear();
                    _contacts.clear();
                    _members.clear();

                    for (DataSnapshot postSnapshot:
dataSnapshot.getChildren()) {
                        String number = postSnapshot.getKey();
```

```

        long id = getIdByNumber(number);

        if (id != -1) {
            _dbNumbers.put(id, number);
        }
    }

    retrieveContacts();
    retrieveMembers();

    if (adapter != null) {
        adapter.notifyDataSetChanged();
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {

}
}
);
}

```

```

public static void retrieveContacts() {
    LongSparseArray<Contact> array = new LongSparseArray();

    Cursor cursor = getCursor();

    if (cursor != null) {
        while (cursor.moveToNext()) {
            long id = getId(cursor);

            Contact contact = array.get(id);

            if (contact == null) {
                contact = new Contact(getName(cursor));

                array.put(id, contact);
                _contacts.add(contact);
            }

            if (_dbNumbers.get(id) != null) {
                contact.markAsMember();
                contact.dbNumber = _dbNumbers.get(id);
            }

            contact.addPhone(getPhone(cursor));
        }

        cursor.close();
    }
}

```

```

public static void retrieveMembers() {
    _members.addAll(_contacts);

    CollectionUtils.filter(_members, new Predicate() {

        @Override
        public boolean evaluate(Object o) {
            return ((Contact) o).isMember();
        }

    });
}

public static String getNameByPhone(String phone) {
    String name = "This number is not in your contact list.";

    Uri uri =
Uri.withAppendedPath(ContactsContract.PhoneLookup.CONTENT_FILTER_
URI, Uri.encode(phone));
    String[] projection = {
ContactsContract.PhoneLookup.DISPLAY_NAME };
    Cursor cursor = _context.getContentResolver().query(uri,
projection, null, null, null);

    if (cursor != null) {
        if (cursor.moveToNext()) {
            name = getName(cursor);
        }

        cursor.close();
    }

    return name;
}

public static long getIdByNumber(String phone) {
    long id = -1;

    Uri uri =
Uri.withAppendedPath(ContactsContract.PhoneLookup.CONTENT_FILTER_
URI, Uri.encode(phone));
    String[] projection = { ContactsContract.PhoneLookup._ID };
    Cursor cursor = _context.getContentResolver().query(uri,
projection, null, null, null);

    if (cursor != null) {
        if (cursor.moveToNext()) {
            id =
cursor.getLong(cursor.getColumnIndex(ContactsContract.PhoneLookup._I
D));
        }
    }
}

```

```

        cursor.close();
    }

    return id;
}

protected static Cursor getCursor() {
    Uri uri =
ContactsContract.CommonDataKinds.Contactables.CONTENT_URI;

    String[] projection = {
        ContactsContract.Data.MIMETYPE,
        ContactsContract.Data.CONTACT_ID,
        ContactsContract.Contacts.DISPLAY_NAME,
        ContactsContract.CommonDataKinds.Contactables.DATA
    };

    String selection = ContactsContract.Data.MIMETYPE + " in (?, ?)";

    String[] selectionArgs = {
ContactsContract.CommonDataKinds.Phone.CONTENT_ITEM_TYPE
    };

    String sortOrder = ContactsContract.Contacts.DISPLAY_NAME + "
ASC";

    return _context.getContentResolver().query(uri, projection,
selection, selectionArgs, sortOrder);
}

protected static long getId(Cursor cursor) {
    return
cursor.getLong(cursor.getColumnIndex(ContactsContract.Data.CONTACT_
ID));
}

protected static String getName(Cursor cursor) {
    return
cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.DISPL
AY_NAME));
}

protected static String getPhone(Cursor cursor) {
    return
cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKi
nds.Contactables.DATA));
}
}

```


booki/booki/ContactsFragment.java

```
package booki.booki;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.TypedArray;
import android.graphics.Color;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.DialogFragment;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;

import com.github.pavlospt.roundedletterview.RoundedLetterView;

import java.util.List;

/**
 * A simple {@link Fragment} subclass.
 */
public class ContactsFragment extends DialogFragment {

    protected boolean _dialog;

    public ContactsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_contacts, container, false);
    }

    @Override
    public void onDismiss(DialogInterface dialog)
    {
        Activity activity = getActivity();
        if (activity instanceof NewActivity)
    }
}
```

```

        ((NewActivity)activity).onContactSelect();
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent
data) {
        if (requestCode == 999) {
            ContactsFragment fragment = new ContactsFragment();
            FragmentTransaction fragmentTransaction =
getFragmentManager().beginTransaction();
            fragmentTransaction.replace(R.id.fragment_container, fragment);
            fragmentTransaction.commit();
        }
    }

    @Override
    public void onViewCreated(View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        if (this.getArguments() != null) {
            _dialog = this.getArguments().getBoolean("dialog");
            getDialog().setTitle(this.getArguments().getString("title"));
        }

        List<Contact> contacts = _dialog
            ? Contacts._members
            : Contacts._contacts;

        Contacts.adapter = new ContactAdapter(getActivity(), contacts);
        ListView listView = (ListView) view.findViewById(R.id.contacts_list);
        listView.setAdapter(Contacts.adapter);

        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                Contact contact = Contacts.adapter.getItem(position);

                if (_dialog) {
                    Utils.selectedContact = contact;
                    getDialog().dismiss();
                } else {
                    if (contact.isMember()) {
                        Utils.selectedContact = contact;
                        startActivityForResult(new Intent(getActivity(),
NewActivity.class), 999);
                    } else {
                        new AlertDialog.Builder(getActivity())
                            .setTitle("Refer a friend")

```

```

        .setMessage("Do you want to inform your friend
about Booki?")
        .setCancelable(false)
        .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {
                // Whatever...
            }
        })
        .setNegativeButton("No", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {
                // Whatever...
            }
        }).create().show();
    }
}
});
}

public class ContactAdapter extends ArrayAdapter<Contact> {
    ContactAdapter(Context context, List<Contact> contacts) {
        super(context, R.layout.contact_list_item, R.id.name_text_view,
contacts);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        convertView = super.getView(position, convertView, parent);
        Contact contact = getItem(position);
        String name = contact.name;

        RoundedLetterView letter = (RoundedLetterView)
convertView.findViewById(R.id.letter_view);
        letter.setTitleText(getFirstLetter(name));
        letter.setBackgroundColor(getMaterialColor(name, position));

        TextView nameTextView = (TextView)
convertView.findViewById(R.id.name_text_view);
        nameTextView.setText(name);

        ImageView badge = (ImageView)
convertView.findViewById(R.id.badge_image_view);
        if (contact.isMember()) {
            badge.setVisibility(View.VISIBLE);
        } else {

```

```

        badge.setVisibility(View.GONE);
    }

    return convertView;
}

private int getMaterialColor(String name, int position)
{
    int returnColor = Color.BLACK;
    int arrayId = getResources().getIdentifier("material", "array",
getActivity().getApplicationContext().getPackageName());

    if (arrayId != 0)
    {
        TypedArray colors = getResources().obtainTypedArray(arrayId);
        int index = Math.abs(name.hashCode() + position*5) %
colors.length();
        returnColor = colors.getColor(index, Color.BLACK);
        colors.recycle();
    }

    return returnColor;
}

private String getFirstLetter(String name)
{
    String letter = name.substring(0, 1);

    if (letter.matches("[0-9]")) {
        return "#";
    }

    return letter.toUpperCase();
}
}

```

booki/booki/DetailsActivity.java

```

package booki.booki;

import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.InputType;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

```

```

import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.wdullaer.materialdatetimepicker.date.DatePickerDialog;
import com.wdullaer.materialdatetimepicker.time.RadialPickerLayout;
import com.wdullaer.materialdatetimepicker.time.TimePickerDialog;

import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

public class DetailsActivity extends AppCompatActivity {

    private Calendar _calendar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_details);

        _calendar = Calendar.getInstance();
        _calendar.setTimeInMillis(Utils.selectedMeeting.timestamp);

        TextView contactView = (TextView)
        findViewById(R.id.contact_text_view);

        contactView.setText(Contacts.getNameByPhone(Utils.selectedMeeting.get
        PartnerNumber()));

        TextView titleView = (TextView) findViewById(R.id.title_text_view);
        titleView.setText(Utils.selectedMeeting.getTitle());

        TextView descriptionView = (TextView)
        findViewById(R.id.description_text_view);
        descriptionView.setText(Utils.selectedMeeting.getDescription());

        final TextView locationView = (TextView)
        findViewById(R.id.location_text_view);
        locationView.setText(Utils.selectedMeeting.getLocation());

        TextView dateView = (TextView) findViewById(R.id.date_text_view);
        dateView.setText(_calendar.getTime().toString());

        Button approveButton = (Button)
        findViewById(R.id.approve_button);
        approveButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Utils.selectedMeeting.approve();
                sendToFirebase();
            }
        });
    }
}

```

```

    });

    Button updateButton = (Button) findViewById(R.id.update_button);
    updateButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Utils.selectedMeeting.update(locationView.getText().toString(),
            _calendar.getTimeInMillis());
            sendToFirebase();
        }
    });

    Button rejectButton = (Button) findViewById(R.id.reject_button);
    rejectButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Utils.selectedMeeting.reject();
            sendToFirebase();
        }
    });
}

private void sendToFirebase() {
    Map<String, Object> meeting = Utils.selectedMeeting.toMap();
    DatabaseReference database =
    FirebaseDatabase.getInstance().getReference("meetings");
    String key = Utils.selectedMeeting.key;

    Map<String, Object> updates = new HashMap<>();
    updates.put(Utils.selectedMeeting.getFrom() + "/" + key, meeting);
    updates.put(Utils.selectedMeeting.getTo() + "/" + key, meeting);

    database.updateChildren(updates, new
    DatabaseReference.CompletionListener() {
        @Override
        public void onComplete(DatabaseError databaseError,
    DatabaseReference databaseReference) {
            if (databaseError == null) {
                finish();
            } else {
                //
            }
        }
    });
}

public void openInputDialog(final View view) {
    final TextView textView = (TextView) view;

    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle(textView.getHint());

```

```

        final EditText input = new EditText(this);
        input.setText(textView.getText());
        input.setInputType(InputType.TYPE_CLASS_TEXT);
        builder.setView(input);

        builder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                textView.setText(input.getText());
            }
        });
        builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });

        builder.show();
    }

    public void openDatePicker(final View view) {
        DatePickerDialog.OnDateSetListener callBack = new
DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePickerDialog v, int year, int
monthOfYear, int dayOfMonth) {
                _calendar.set(year, monthOfYear, dayOfMonth);
                openTimePicker(view);
            }
        };

        DatePickerDialog dpd = DatePickerDialog.newInstance(
            callBack,
            _calendar.get(Calendar.YEAR),
            _calendar.get(Calendar.MONTH),
            _calendar.get(Calendar.DAY_OF_MONTH)
        );

        dpd.show(getFragmentManager(), "Datepickerdialog");
    }

    private void openTimePicker(final View view) {
        final TextView textView = (TextView) view;

        TimePickerDialog.OnTimeSetListener callBack = new
TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(RadialPickerLayout v, int hourOfDay, int
minute, int second) {

```

```

        _calendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
        _calendar.set(Calendar.MINUTE, minute);
        _calendar.set(Calendar.SECOND, second);

        textView.setText(_calendar.getTime().toString());
    }
};

TimePickerDialog dpd = TimePickerDialog.newInstance(
    callBack,
    _calendar.get(Calendar.HOUR),
    _calendar.get(Calendar.MINUTE),
    true
);

dpd.show(getFragmentManager(), "Timepickerdialog");
}
}

```

booki/booki/LoginActivity.java

```

package booki.booki;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;

import com.facebook.accountkit.Account;
import com.facebook.accountkit.AccountKit;
import com.facebook.accountkit.AccountKitCallback;
import com.facebook.accountkit.AccountKitError;
import com.facebook.accountkit.AccountKitLoginResult;
import com.facebook.accountkit.ui.AccountKitActivity;
import com.facebook.accountkit.ui.AccountKitConfiguration;
import com.facebook.accountkit.ui.LoginType;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

/**
 * A login screen that offers login via phone.
 */
public class LoginActivity extends AppCompatActivity {

    public static int APP_REQUEST_CODE = 7412;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```



```

        Intent intent = new Intent(this, AccountKitActivity.class);
        AccountKitConfiguration.AccountKitConfigurationBuilder
configurationBuilder =
            new AccountKitConfiguration.AccountKitConfigurationBuilder(
                LoginType.PHONE,
                AccountKitActivity.ResponseType.TOKEN); // or
.AccountKitActivity.ResponseType.TOKEN
        intent.putExtra(
            AccountKitActivity.ACCOUNT_KIT_ACTIVITY_CONFIGURATION,
            configurationBuilder.build());
        startActivityForResult(intent, APP_REQUEST_CODE);
    }

    @Override
    protected void onActivityResult(final int requestCode, final int
resultCode, final Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == APP_REQUEST_CODE) { // confirm that this
response matches your request
            AccountKitLoginResult loginResult =
data.getParcelableExtra(AccountKitLoginResult.RESULT_KEY);

            if (loginResult.getError() != null) {
                Toast.makeText(
                    this,
                    loginResult.getError().getErrorMessage(),
                    Toast.LENGTH_LONG)
                    .show();
            } else if (loginResult.wasCancelled()) {
                //
            } else {
                AccountKit.getCurrentAccount(new
AccountKitCallback<Account>() {
                    @Override
                    public void onSuccess(final Account account) {
                        String dbNumber =
account.getPhoneNumber().toString();
                        DatabaseReference database =
FirebaseDatabase.getInstance().getReference("phones").child(dbNumber)
;

                        database.setValue(true, new
DatabaseReference.CompletionListener() {
                            @Override
                            public void onComplete(DatabaseError databaseError,
DatabaseReference databaseReference) {
                                if (databaseError == null) {
                                    startActivity(new Intent(getApplicationContext(),
MainActivity.class));
                                } else {
                                    //
                                }
                            }
                        });
                    }
                });
            }
        }
    }

```



```

        _fButton = (FloatingActionButton) findViewById(R.id.fab);
        _fButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getApplicationContext(),
NewActivity.class));
            }
        });

        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.setDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);

        // Select the first item on drawer
        MenuItem item = navigationView.getMenu().getItem(0);
        onNavigationItemSelectedListener(item);
        item.setChecked(true);

        // Set the number on Drawer
        TextView mNavPhoneView = (TextView)
navigationView.getHeaderView(0).findViewById(R.id.side_nav_phone);

mNavPhoneView.setText(Utils.formatPhoneNumber(Utils.dbNumber));
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (AccountKit.getCurrentAccessToken() == null) {
            startLoginActivity();
            finish();
        } else {
            AccountKit.getCurrentAccount(new
AccountKitCallback<Account>() {
                @Override
                public void onSuccess(final Account account) {
                    Utils.dbNumber = account.getPhoneNumber().toString();

                    begin();
                }
            });

            @Override

```

```

        public void onError(final AccountKitError error) {
            // Handle Error
        }
    });
}
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout)
    findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_disconnect) {
        finish();

        AccountKit.logout();
        startLoginActivity();

        return true;
    }

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        startActivity(new Intent(getApplicationContext(),
        SettingsActivity.class));

        return true;
    }

    return super.onOptionsItemSelected(item);
}
}

```

```

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.nav_meetings) {
        setTitle("My meetings");
        _fButton.show();

        MeetingsFragment fragment = new MeetingsFragment();
        FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
        fragmentTransaction.replace(R.id.fragment_container, fragment);
        fragmentTransaction.commit();
    } else if (id == R.id.nav_contacts) {
        setTitle("My contacts");
        _fButton.hide();

        ContactsFragment fragment = new ContactsFragment();
        FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
        fragmentTransaction.replace(R.id.fragment_container, fragment);
        fragmentTransaction.commit();
    } else if (id == R.id.nav_share) {

    } else if (id == R.id.nav_send) {

    }

    DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}

private void startLoginActivity() {
    Intent i = new Intent(this, LoginActivity.class);

    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

    i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

    startActivity(i);
}
}

```

booki/booki/Meeting.java

```
package booki.booki;

import com.google.firebase.database.Exclude;

import java.util.Date;
import java.util.HashMap;
import java.util.Map;

public class Meeting {

    public String key;

    public String status;

    public String from;
    public String to;

    public String title;
    public String description;
    public String location;

    public String getDescription() {
        return description;
    }

    public long timestamp;

    protected Date _date;

    public Meeting() {

    }

    public String getFrom() {
        return from;
    }

    public String getTo() {
        return to;
    }

    public String getStatus() {
        return status;
    }

    public String getTitle() {
        return title;
    }

    public String getLocation() {
        return location;
    }
}
```

```

    }

    public Date getDate() {
        if (_date == null) {
            _date = new Date(timestamp);
        }

        return _date;
    }

    public String getPartnerNumber() {
        return from.equals(Utils.dbNumber)
            ? to
            : from;
    }

    public void create(String to, String title, String description, String
location, long timestamp) {
        status = "waiting_to";
        from = Utils.dbNumber;

        this.to = to;
        this.title = title;
        this.description = description;
        this.location = location;
        this.timestamp = timestamp;
    }

    public void reject() {
        status = "rejected";
    }

    public void approve() {
        status = "approved";
    }

    public void update(String location, long timestamp) {
        status = status.equals("waiting_from")
            ? "waiting_to"
            : "waiting_from";

        this.location = location;
        this.timestamp = timestamp;
    }

    @Exclude
    public Map<String, Object> toMap() {
        HashMap<String, Object> result = new HashMap<>();
        result.put("status", status);
        result.put("from", from);
        result.put("to", to);
        result.put("title", title);
    }

```

```

        result.put("description", description);
        result.put("location", location);
        result.put("timestamp", timestamp);

        return result;
    }
}

```

booki/booki/MeetingsFragment.java

```

package booki.booki;

import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v7.widget.CardView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.Adapter;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import org.apache.commons.lang3.time.DateUtils;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

/**
 * A simple {@link Fragment} subclass.
 */
public class MeetingsFragment extends Fragment {

    private DatabaseReference _database;

    private List<Meeting> _meetings;

```



```

private String _lastDate = "";

public MeetingsFragment() {
    // Required empty public constructor
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

    _database = FirebaseDatabase.getInstance().getReference();

    _meetings = new ArrayList();

    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_meetings, container, false);
}

@Override
public void onViewCreated(View view, @Nullable Bundle
savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    ListView listView = (ListView) view.findViewById(R.id.meetings_list);
    final MeetingsFragment.MeetingAdapter adapter = new
MeetingsFragment.MeetingAdapter(getActivity(), _meetings);
    listView.setAdapter(adapter);

    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            Utils.selectedMeeting = adapter.getItem(position);
            startActivity(new Intent(getActivity(), DetailsActivity.class));
        }
    });

    _database.child("meetings").child(Utils.dbNumber).orderByChild("timesta
mp").addValueEventListener(
        new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                _meetings.clear();

                for (DataSnapshot postSnapshot:
dataSnapshot.getChildren()) {

                    _meetings.add(postSnapshot.getValue(Meeting.class));
                    _meetings.get(_meetings.size()-1).key =
postSnapshot.getKey();

```

```

        }

        adapter.notifyDataSetChanged();
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
}

);
}

private class MeetingAdapter extends ArrayAdapter<Meeting> {
    MeetingAdapter(Context context, List<Meeting> meetings) {
        super(context, R.layout.meeting_list_item, R.id.title_text_view,
meetings);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        convertView = super.getView(position, convertView, parent);
        Meeting meeting = getItem(position);

        TextView date =
(TextView)convertView.findViewById(R.id.date_text_view);

        if (isSameDay(position)) {
            date.setVisibility(View.GONE);
        } else {
            date.setText(Utils.dateFormat.format(meeting.getDate()));
            date.setVisibility(View.VISIBLE);
        }

        TextView time =
(TextView)convertView.findViewById(R.id.time_text_view);
        time.setText(Utils.timeFormat.format(meeting.getDate()));

        setStatusColor(convertView, meeting);

        TextView title =
(TextView)convertView.findViewById(R.id.title_text_view);
        title.setText(meeting.getTitle());

        String number = meeting.getPartnerNumber();

        TextView name =
(TextView)convertView.findViewById(R.id.name_text_view);
        name.setText(Contacts.getNameByPhone(number));

```

```

        TextView phone =
        (TextView)convertView.findViewById(R.id.phone_text_view);
        phone.setText(Utils.formatPhoneNumber(number));

        TextView location =
        (TextView)convertView.findViewById(R.id.location_text_view);
        location.setText(meeting.getLocation());

        int color = R.color.mdtp_white;
        if (!meeting.status.equals("approved") &&
!meeting.status.equals("rejected")) {
            if (meeting.from.equals(Utils.dbNumber) &&
meeting.status.equals("waiting_from") ||
                meeting.to.equals(Utils.dbNumber) &&
meeting.status.equals("waiting_to")) {
                color = R.color.highlight_card;
            }
        }

        CardView card = (CardView)
convertView.findViewById(R.id.meeting_card);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            card.setBackgroundColor(getResources().getColor(color,
getResources().newTheme()));
        } else {
            card.setBackgroundColor(getResources().getColor(color));
        }

        return convertView;
    }
}

private boolean isSameDay(int position) {
    if (position == 0) {
        return false;
    }

    Date first = _meetings.get(position - 1).getDate();
    Date second = _meetings.get(position).getDate();

    return DateUtils.isSameDay(first, second);
}

private void setStatusColor(View convertView, Meeting meeting) {
    TextView status =
    (TextView)convertView.findViewById(R.id.status_text_view);

    int color;

    switch (meeting.getStatus()) {
        case "approved":

```

```

        color = R.color.status_approved;
        break;
    case "rejected":
        color = R.color.status_rejected;
        break;
    default:
        color = R.color.status_waiting;
        break;
    }

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        status.setTextColor(getResources().getColor(color,
getResources().newTheme()));
    } else {
        status.setTextColor(getResources().getColor(color));
    }
}
}
}
}

```

booki/booki/NewActivity.java

```

package booki.booki;

import android.app.Activity;
import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.InputType;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.wdullaer.materialdatetimepicker.date.DatePickerDialog;
import com.wdullaer.materialdatetimepicker.time.RadialPickerLayout;
import com.wdullaer.materialdatetimepicker.time.TimePickerDialog;

import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

public class NewActivity extends AppCompatActivity {

    private String _title;

```

```

private String _description;
private String _location;
private Calendar _calendar;

private TextView _contactView;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_new);

    _calendar = Calendar.getInstance();

    _contactView = (TextView) findViewById(R.id.contact_text_view);
    onContactSelect();

    Button sendButton = (Button) findViewById(R.id.send_button);
    sendButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (Utils.selectedContact != null) {
                sendToFirebase();
            }
        }
    });
}

@Override
protected void onDestroy() {
    super.onDestroy();

    Utils.selectedContact = null;

    setResult(Activity.RESULT_CANCELED, null);
}

private void sendToFirebase() {
    Map<String, Object> meeting = createMeeting();
    DatabaseReference database =
FirebaseDatabase.getInstance().getReference("meetings");
    String key = database.child(Utils.dbNumber).push().getKey();

    Map<String, Object> updates = new HashMap<>();
    updates.put(Utils.dbNumber + "/" + key, meeting);
    updates.put(Utils.selectedContact.dbNumber + "/" + key, meeting);

    database.updateChildren(updates, new
DatabaseReference.CompletionListener() {
        @Override
        public void onComplete(DatabaseError databaseError,
DatabaseReference databaseReference) {
            if (databaseError == null) {

```

```

        finish();
    } else {
        //
    }
}
});
}

private Map<String, Object> createMeeting() {
    Meeting meeting = new Meeting();
    meeting.create(Utils.selectedContact.dbNumber, _title, _description,
_location, _calendar.getTimeInMillis());

    return meeting.toMap();
}

public void onContactSelect() {
    if (Utils.selectedContact != null) {
        _contactView.setText(Utils.selectedContact.name);
    }
}

public void openContactPicker(final View view) {
    Bundle bundle = new Bundle();
    bundle.putBoolean("dialog", true);
    bundle.putString("title", _contactView.getHint().toString());

    ContactsFragment newFragment = new ContactsFragment();
    newFragment.setArguments(bundle);
    newFragment.show(getSupportFragmentManager(), "dialog");
}

public void openInputDialog(final View view) {
    final TextView textView = (TextView) view;
    final int id = textView.getId();

    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle(textView.getHint());

    final EditText input = new EditText(this);
    input.setText(textView.getText());
    if (id == R.id.description_text_view) {

input.setInputType(InputType.TYPE_TEXT_FLAG_IME_MULTI_LINE);
        input.setSingleLine(false);
        input.setImeOptions(EditorInfo.IME_FLAG_NO_ENTER_ACTION);
    } else {
        input.setInputType(InputType.TYPE_CLASS_TEXT);
    }
    builder.setView(input);
}

```

```

        builder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        String text = input.getText().toString();
        textView.setText(text);

        switch (id) {
            case R.id.title_text_view:
                _title = text;
                break;
            case R.id.description_text_view:
                _description = text;
                break;
            case R.id.location_text_view:
                _location = text;
                break;
        }
    }
});
        builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});

        builder.show();
    }

    public void openDatePicker(final View view) {
        DatePickerDialog.OnDateSetListener callBack = new
DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePickerDialog v, int year, int
monthOfYear, int dayOfMonth) {
        _calendar.set(year, monthOfYear, dayOfMonth);
        openTimePicker(view);
    }
};

        DatePickerDialog dpd = DatePickerDialog.newInstance(
            callBack,
            _calendar.get(Calendar.YEAR),
            _calendar.get(Calendar.MONTH),
            _calendar.get(Calendar.DAY_OF_MONTH)
        );

        dpd.show(getFragmentManager(), "Datepickerdialog");
    }

```

```

private void openTimePicker(final View view) {
    final TextView textView = (TextView) view;

    TimePickerDialog.OnTimeSetListener callBack = new
TimePickerDialog.OnTimeSetListener() {
    @Override
    public void onTimeSet(RadialPickerLayout v, int hourOfDay, int
minute, int second) {
        _calendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
        _calendar.set(Calendar.MINUTE, minute);
        _calendar.set(Calendar.SECOND, second);

        textView.setText(_calendar.getTime().toString());
    }
};

    TimePickerDialog dpd = TimePickerDialog.newInstance(
        callBack,
        _calendar.get(Calendar.HOUR),
        _calendar.get(Calendar.MINUTE),
        true
    );

    dpd.show(getFragmentManager(), "Timepickerdialog");
}
}

```

booki/booki/SettingsActivity.java

```

package booki.booki;

import android.annotation.TargetApi;
import android.content.Context;
import android.content.Intent;
import android.content.res.Configuration;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.preference.ListPreference;
import android.preference.Preference;
import android.preference.PreferenceActivity;
import android.support.v7.app.ActionBar;
import android.preference.PreferenceFragment;
import android.preference.PreferenceManager;
import android.preference.RingtonePreference;
import android.text.TextUtils;
import android.view.MenuItem;
import android.support.v4.app.NavUtils;

```



```

import java.util.List;

/**
 * A {@link PreferenceActivity} that presents a set of application settings.
 * On
 * handset devices, settings are presented as a single list. On tablets,
 * settings are split by category, with category headers shown to the left
 * of
 * the list of settings.
 * <p>
 * See <a
href="http://developer.android.com/design/patterns/settings.html" >
 * Android Design: Settings</a> for design guidelines and the <a
 *
href="http://developer.android.com/guide/topics/ui/settings.html" >Settin
 *
gs
 * API Guide</a> for more information on developing a Settings UI.
 */
public class SettingsActivity extends AppCompatActivity {
    /**
     * A preference value change listener that updates the preference's
     * summary
     * to reflect its new value.
     */
    private static Preference.OnPreferenceChangeListener
sBindPreferenceSummaryToValueListener = new
Preference.OnPreferenceChangeListener() {
        @Override
        public boolean onPreferenceChange(Preference preference, Object
value) {
            String stringValue = value.toString();

            if (preference instanceof ListPreference) {
                // For list preferences, look up the correct display value in
                // the preference's 'entries' list.
                ListPreference listPreference = (ListPreference) preference;
                int index = listPreference.findIndexOfValue(stringValue);

                // Set the summary to reflect the new value.
                preference.setSummary(
                    index >= 0
                    ? listPreference.getEntries()[index]
                    : null);
            } else if (preference instanceof RingtonePreference) {
                // For ringtone preferences, look up the correct display value
                // using RingtoneManager.
                if (TextUtils.isEmpty(stringValue)) {
                    // Empty values correspond to 'silent' (no ringtone).
                    preference.setSummary(R.string.pref_ringtone_silent);
                } else {

```

```

        Ringtone ringtone = RingtoneManager.getRingtone(
            preference.getContext(), Uri.parse(stringValue));

        if (ringtone == null) {
            // Clear the summary if there was a lookup error.
            preference.setSummary(null);
        } else {
            // Set the summary to reflect the new ringtone display
            // name.
            String name =
ringtone.getTitle(preference.getContext());
            preference.setSummary(name);
        }
    }

    } else {
        // For all other preferences, set the summary to the value's
        // simple string representation.
        preference.setSummary(stringValue);
    }
    return true;
}
};

/**
 * Helper method to determine if the device has an extra-large screen.
For
 * example, 10" tablets are extra-large.
 */
private static boolean isXLargeTablet(Context context) {
    return (context.getResources().getConfiguration().screenLayout
        & Configuration.SCREENLAYOUT_SIZE_MASK) >=
Configuration.SCREENLAYOUT_SIZE_XLARGE;
}

/**
 * Binds a preference's summary to its value. More specifically, when
the
 * preference's value is changed, its summary (line of text below the
 * preference title) is updated to reflect the value. The summary is also
 * immediately updated upon calling this method. The exact display
format is
 * dependent on the type of preference.
 *
 * @see #sBindPreferenceSummaryToValueListener
 */
private static void bindPreferenceSummaryToValue(Preference
preference) {
    // Set the listener to watch for value changes.

preference.setOnPreferenceChangeListener(sBindPreferenceSummaryToVa
lueListener);

```

```

        // Trigger the listener immediately with the preference's
        // current value.

sBindPreferenceSummaryToValueListener.onPreferenceChange(preference
,
        PreferenceManager
        .getDefaultSharedPreferences(preference.getContext())
        .getString(preference.getKey(), ""));
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setupActionBar();
    }

    /**
     * Set up the {@link android.app.ActionBar}, if the API is available.
     */
    private void setupActionBar() {
        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            // Show the Up button in the action bar.
            actionBar.setDisplayHomeAsUpEnabled(true);
        }
    }

    @Override
    public boolean onOptionsItemSelected(int featureId, MenuItem item) {
        int id = item.getItemId();
        if (id == android.R.id.home) {
            if (!super.onOptionsItemSelected(featureId, item)) {
                NavUtils.navigateUpFromSameTask(this);
            }
            return true;
        }
        return super.onOptionsItemSelected(featureId, item);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public boolean onIsMultiPane() {
        return isXLargeTablet(this);
    }

    /**
     * {@inheritDoc}
     */
    @Override

```

```

@TargetApi(Build.VERSION_CODES.HONEYCOMB)
public void onBuildHeaders(List<Header> target) {
    loadHeadersFromResource(R.xml.pref_headers, target);
}

/**
 * This method stops fragment injection in malicious applications.
 * Make sure to deny any unknown fragments here.
 */
protected boolean isValidFragment(String fragmentName) {
    return PreferenceFragment.class.getName().equals(fragmentName)
        ||
GeneralPreferenceFragment.class.getName().equals(fragmentName)
        ||
DataSyncPreferenceFragment.class.getName().equals(fragmentName)
        ||
NotificationPreferenceFragment.class.getName().equals(fragmentName);
}

/**
 * This fragment shows general preferences only. It is used when the
 * activity is showing a two-pane settings UI.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB)
public static class GeneralPreferenceFragment extends
PreferenceFragment {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.pref_general);
        setHasOptionsMenu(true);

        // Bind the summaries of EditText/List/Dialog/Ringtone
preferences
        // to their values. When their values change, their summaries are
        // updated to reflect the new value, per the Android Design
        // guidelines.

        bindPreferenceSummaryToValue(findPreference("example_text"));
        bindPreferenceSummaryToValue(findPreference("example_list"));
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == android.R.id.home) {
            startActivity(new Intent(getActivity(), SettingsActivity.class));
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
}

```

```

/**
 * This fragment shows notification preferences only. It is used when
the
 * activity is showing a two-pane settings UI.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB)
public static class NotificationPreferenceFragment extends
PreferenceFragment {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.pref_notification);
        setHasOptionsMenu(true);

        // Bind the summaries of EditText/List/Dialog/Ringtone
preferences
        // to their values. When their values change, their summaries are
        // updated to reflect the new value, per the Android Design
        // guidelines.

        bindPreferenceSummaryToValue(findPreference("notifications_new_messa
ge_ringtone"));
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == android.R.id.home) {
            startActivity(new Intent(getActivity(), SettingsActivity.class));
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}

/**
 * This fragment shows data and sync preferences only. It is used when
the
 * activity is showing a two-pane settings UI.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB)
public static class DataSyncPreferenceFragment extends
PreferenceFragment {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.pref_data_sync);
        setHasOptionsMenu(true);

        // Bind the summaries of EditText/List/Dialog/Ringtone
preferences

```

```

        // to their values. When their values change, their summaries are
        // updated to reflect the new value, per the Android Design
        // guidelines.

        bindPreferenceSummaryToValue(findPreference("sync_frequency"));
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == android.R.id.home) {
            startActivity(new Intent(getActivity(), SettingsActivity.class));
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
}
}
}

```

booki/booki/Utils.java

```

package booki.booki;

import android.os.Build;
import android.telephony.PhoneNumberUtils;

import java.text.DateFormat;
import java.text.SimpleDateFormat;

public class Utils {

    public static DateFormat dateFormat = new SimpleDateFormat("EEEE,
dd MMMM yyyy");
    public static DateFormat timeFormat = new
SimpleDateFormat("HH: mm");

    public static String dbNumber;

    public static Contact selectedContact;
    public static Meeting selectedMeeting;

    public static String formatPhoneNumber(String number) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            return PhoneNumberUtils.formatNumber(number, "GR");
        } else {
            return PhoneNumberUtils.formatNumber(number);
        }
    }
}
}

```

res/layout/activity_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context="booki.booki.NewActivity" >

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1" >

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical" >

<TextView
    android:text="Contact"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />

<TextView
    android:id="@+id/contact_text_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_person_accent"
    android:drawablePadding="48dp"
    android:textSize="16sp" />

<TextView
    android:text="Title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />

<TextView
```

```

        android:id="@+id/title_text_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:gravity="center_vertical"
        android:drawableStart="@drawable/ic_label"
        android:drawablePadding="48dp"
        android:textSize="16sp" />

<TextView
    android:text="Description"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />

<TextView
    android:id="@+id/description_text_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_subject"
    android:drawablePadding="48dp"
    android:textSize="16sp" />

<TextView
    android:text="Location"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />

<TextView
    android:id="@+id/location_text_view"
    android:clickable="true"
    android:onClick="openInputDialog"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_location_accent"
    android:drawablePadding="48dp"
    android:textSize="16sp" />

<TextView
    android:text="Date & Time"
    android:layout_width="match_parent"

```



```
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />
```

```
<TextView
    android:id="@+id/date_text_view"
    android:clickable="true"
    android:onClick="openDatePicker"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_access_time"
    android:drawablePadding="48dp"
    android:textSize="16sp" />
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
```

```
<Button
    android:id="@+id/reject_button"
    android:text="Reject"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/status_rejected"
    android:layout_weight="1" />
```

```
<Button
    android:id="@+id/update_button"
    android:text="Update"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@android:color/black"
    android:layout_weight="1" />
```

```
<Button
    android:id="@+id/approve_button"
    android:text="Approve"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/status_approved"
    android:layout_weight="1" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/drawer_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
android:background="#ECEFF1"
tools:openDrawer="start" >

<include
layout="@layout/app_bar_main"
android:layout_width="match_parent"
android:layout_height="match_parent" />

<android.support.design.widget.NavigationView
android:id="@+id/nav_view"
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:layout_gravity="start"
android:fitsSystemWindows="true"
app:headerLayout="@layout/nav_header_main"
app:menu="@menu/activity_main_drawer" />

</android.support.v4.widget.DrawerLayout>
```

res/layout/activity_new.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context="booki.booki.NewActivity" >

<ScrollView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1" >

<LinearLayout
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:orientation="vertical" >
```

```
<TextView
    android:text="Contact"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />
```

```
<TextView
    android:id="@+id/contact_text_view"
    android:hint="Pick a contact."
    android:clickable="true"
    android:onClick="openContactPicker"
    android:textColorHint="@android:color/tertiary_text_light"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_person_accent"
    android:drawablePadding="48dp"
    android:textSize="16sp" />
```

```
<TextView
    android:text="Title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />
```

```
<TextView
    android:id="@+id/title_text_view"
    android:hint="Give a title."
    android:clickable="true"
    android:onClick="openInputDialog"
    android:textColorHint="@android:color/tertiary_text_light"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_label"
    android:drawablePadding="48dp"
    android:textSize="16sp" />
```

```
<TextView
    android:text="Description"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />
```

```
<TextView
    android:id="@+id/description_text_view"
    android:hint="Describe the meeting."
    android:clickable="true"
    android:onClick="openInputDialog"
    android:textColorHint="@android:color/tertiary_text_light"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_subject"
    android:drawablePadding="48dp"
    android:textSize="16sp" />
```

```
<TextView
    android:text="Location"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical"
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />
```

```
<TextView
    android:id="@+id/location_text_view"
    android:hint="Where will take place?"
    android:clickable="true"
    android:onClick="openInputDialog"
    android:textColorHint="@android:color/tertiary_text_light"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_location_accent"
    android:drawablePadding="48dp"
    android:textSize="16sp" />
```

```
<TextView
    android:text="Date & Time"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical"
```

```
    android:textColor="@color/colorPrimary"
    android:textSize="12sp" />
```

```
<TextView
    android:id="@+id/date_text_view"
    android:hint="Pick a date and time."
    android:clickable="true"
    android:onClick="openDatePicker"
    android:textColorHint="@android:color/tertiary_text_light"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center_vertical"
    android:drawableStart="@drawable/ic_access_time"
    android:drawablePadding="48dp"
    android:textSize="16sp" />
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
<Button
    android:id="@+id/send_button"
    android:text="Send Request"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimaryDark"
    android:textColor="@android:color/white" />
```

```
</LinearLayout>
```

res/layout/app_bar_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="booki.booki.MainActivity" >

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay" >

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
```

```

        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay" />

</android.support.design.widget.AppBarLayout>

<FrameLayout
    android:id="@+id/fragment_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">

</FrameLayout>

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="@dimen/fab_margin"
    android:src="@android:drawable/ic_input_add"
    android:tint="#ffffff" />

</android.support.design.widget.CoordinatorLayout>

```

res/layout/contact_list_item.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp">

    <com.github.pavlospt.rounderletterview.RoundedLetterView
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:id="@+id/letter_view"
        android:layout_width="48dp"
        android:layout_height="48dp"
        app:rlv_titleColor="@android:color/white"
        app:rlv_titleSize="24dp" />

    <TextView
        android:id="@+id/name_text_view"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginStart="24dp"
        android:text="Full Name"

```

```

    android:textAppearance="@style/TextAppearance.AppCompat.Body2"
        android:gravity="center_vertical"
        android:maxLines="1"
        android:ellipsize="end"
        android:layout_weight="1" />

    <ImageView
        android:id="@+id/badge_image_view"
        android:layout_width="32dp"
        android:layout_height="match_parent"
        android:layout_marginStart="32dp"
        android:layout_marginEnd="16dp"
        app:srcCompat="@drawable/badge" />
</LinearLayout>

```

res/layout/fragment_contacts.xml

```

<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="booki.booki.ContactsFragment" >

    <ListView
        android:id="@+id/contacts_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

    </ListView>

</FrameLayout>

```

res/layout/fragment_meetings.xml

```

<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="booki.booki.MeetingsFragment" >

    <ListView
        android:id="@+id/meetings_list"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="4dp"
        android:divider="@android:color/transparent" >

    </ListView>

```

```
</FrameLayout>
```

res/layout/meeting_list_item.xml

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

```
<LinearLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#ECEFF1" >
```

```
    <TextView
```

```
        android:id="@+id/date_text_view"  
        android:text="TextView"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:paddingLeft="12dp"  
        android:paddingBottom="8dp"  
        android:paddingTop="8dp"  
        android:textAllCaps="true"  
        android:textColor="#B0BEC5"
```

```
        android:textAppearance="@style/TextAppearance.AppCompat.Caption"  
        android:textStyle="normal|bold" />
```

```
    <android.support.v7.widget.CardView
```

```
        android:id="@+id/meeting_card"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_margin="4dp"  
        android:background="@android:color/white"  
        android:foreground="?android:attr/selectableItemBackground"  
        android:clickable="false" >
```

```
        <LinearLayout
```

```
            android:orientation="horizontal"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:padding="16dp" >
```

```
            <TextView
```

```
                android:id="@+id/time_text_view"  
                android:text="06:00"  
                android:layout_height="wrap_content"  
                android:layout_weight="0"  
                android:layout_marginStart="2dp"
```

```
                android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
                android:layout_width="wrap_content" />
```

```
            <TextView
```



```
android:id="@+id/status_text_view"  
android:text="\u25CF"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_weight="0"  
android:layout_marginLeft="14dp"  
android:layout_marginRight="14dp" />
```

```
<LinearLayout
```

```
android:orientation="vertical"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="1" >
```

```
<TextView
```

```
android:id="@+id/title_text_view"  
android:text="The title of meeting"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:textStyle="bold"  
android:maxLines="1"  
android:ellipsize="end" />
```

```
<TextView
```

```
android:id="@+id/name_text_view"  
android:text="Contact Name"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="16dp"  
android:gravity="center_vertical"  
android:drawableStart="@drawable/ic_person"  
android:drawablePadding="8dp"  
android:textSize="12sp" />
```

```
<TextView
```

```
android:id="@+id/phone_text_view"  
android:text="Phone Number"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="12dp"  
android:gravity="center_vertical"  
android:drawableStart="@drawable/ic_smartphone"  
android:drawablePadding="8dp"  
android:textSize="12sp" />
```

```
<TextView
```

```
android:id="@+id/location_text_view"  
android:text="Location"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="12dp"  
android:gravity="center_vertical"
```

```

        android:drawableStart="@drawable/ic_location"
        android:drawablePadding="8dp"
        android:textSize="12sp" />

    </LinearLayout>
</LinearLayout>

</android.support.v7.widget.CardView>

</LinearLayout>

# res/layout/nav_header_main.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:background="@color/colorPrimaryDark"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark" >

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="64dp"
        android:scaleType="fitStart"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:src="@drawable/logo" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:text="My phone number"

        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

    <TextView
        android:id="@+id/side_nav_phone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>

```

ΒΙΒΛΙΟΓΡΑΦΕΙΑ

- [1] <https://el.wikipedia.org/wiki/Smartphone>
- [2] <https://el.wikipedia.org/wiki/Android>
- [3] https://en.wikipedia.org/wiki/Android_version_history
- [4] http://android.wikia.com/wiki/Android_Wiki
- [5] <http://developer.android.com>
- [6] https://en.wikipedia.org/wiki/Android_Studio
- [7] <http://www.javacodegeeks.com/page/2/?s=Android+Tutorial>
- [8] https://en.wikipedia.org/wiki/Android_%28operating_system%29
- [9] <http://developer.android.com/guide/components/activities.html>
- [10] <http://techspecs-hossain.blogspot.gr/2014/09/android-timeline-version.html>
- [11] <https://developers.facebook.com/docs/accountkit>
- [12] https://en.wikipedia.org/wiki/Firebase#Realtime_Database
- [13] <https://firebase.google.com/>