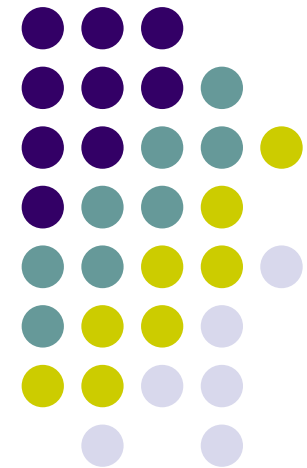


Τεχνολογία και Προγραμματισμός Κινητών Συσκευών

6b



Πάροχοι περιεχομένου
Ιωάννης Έλληνας



Content providers



- Σε πολλές περιπτώσεις είναι επιθυμητή η κοινή χρήση μιας βάσης δεδομένων, όπως για παράδειγμα η κοινή χρήση των επαφών μιας κινητής συσκευής.
- Οι πάροχοι περιεχομένου (Content Providers) προσφέρουν τη δυνατότητα κοινής χρήσης μιας βάσης δεδομένων.
- Ο χρήστης μπορεί να κάνει χρήση παρόχου περιεχομένου που διαθέτει το λειτουργικό σύστημα ή παρόχου περιεχομένου που έχει αναπτύξει ο ίδιος.
- Ο πάροχος περιεχομένου είναι ένας εξυπηρετητής των εφαρμογών της συσκευής οι οποίες ζητούν δεδομένα και συμπεριφέρονται σαν πελάτες του.
- Στον πάροχο περιεχομένου μπορεί να γίνει αναζήτηση δεδομένων, προσθήκη, διαγραφή ή ενημέρωση.
- Οι συσκευές Android διαθέτουν παρόχους περιεχομένου για διάφορους σκοπούς, όπως στους φυλλομετρητές (ιστορικό, σελιδοδείκτες), στον κατάλογο των επαφών (στοιχεία επαφής), στις κλήσεις (στοιχεία κλήσης, αναπάντητες), στα πολυμέσα (αρχεία ήχου, εικόνας, βίντεο).



- Ο χρήστης ή θα χρησιμοποιήσει τους παρόχους περιεχομένου που διαθέτει μια συσκευή ή θα δημιουργήσει το δικό του πάροχο περιεχομένου ο οποίος θα είναι εφοδιασμένος με τις κατάλληλες μεθόδους για να επιτρέψει την κοινή χρήση των δεδομένων της εφαρμογής από άλλες εφαρμογές.
- Οι μέθοδοι που απαιτούνται γ' αυτόν το σκοπό μοιάζουν με τις αντίστοιχες μεθόδους μιας βάσης δεδομένων και είναι οι εξής:

onCreate() - Καλείται κατά τη δημιουργία του παρόχου και πραγματοποιεί όλες τις αρχικοποιήσεις που απαιτούνται.

insert() - Καλείται για την εισαγωγή μιας σειράς στη βάση δεδομένων του παρόχου. Δέχεται σαν παράμετρο τον προορισμό της εισαγωγής και επιστρέφει το URI (Uniform Resource Identifier) της νέας σειράς. Το URI είναι μια συμβολοσειρά που προσδιορίζει την ονομασία ενός πόρου στο διαδίκτυο.

delete() - Καλείται για τη διαγραφή σειρών της βάσης δεδομένων και επιστρέφει τον αριθμό γραμμών που διαγράφηκαν.

update() - Καλείται για την ενημέρωση ήδη υπάρχουσών γραμμών και επιστρέφει τον αριθμό γραμμών που ενημερώθηκαν.

query() - Καλείται για την αναζήτηση δεδομένων στη βάση και επιστρέφει το αποτέλεσμα σαν αντικείμενο του δρομέα (Cursor object).

getType() - Επιστρέφει τον τύπο των δεδομένων του παρόχου, γνωστό σαν τύπο MIME (Multipurpose Internet Mail Extensions). Το MIME είναι ένα διαδικτυακό πρότυπο το οποίο επεκτείνει το σχηματικό ενός email πέραν του συνηθισμένου κώδικα ASCII προσδιορίζοντας τους μηχανισμούς αποστολής.

- Κάθε πάροχος έχει τη δυνατότητα να παρέχει πρόσβαση σε πολλούς πίνακες δεδομένων και συνεπώς τόσο ο εξυπηρετητής όσο και ο πελάτης πρέπει να ταυτοποιούν έναν πίνακα, δηλαδή πρέπει κάθε πίνακας να έχει μια μοναδική διεύθυνση. Αυτό πραγματοποιείται με το URI περιεχομένου (Content URI), το οποίο παρέχει τη διαδρομή ενός πίνακα δεδομένων σε έναν καθορισμένο πάροχο περιεχομένου.



`<standard_prefix>://<authority>/<data_path>/<id>`

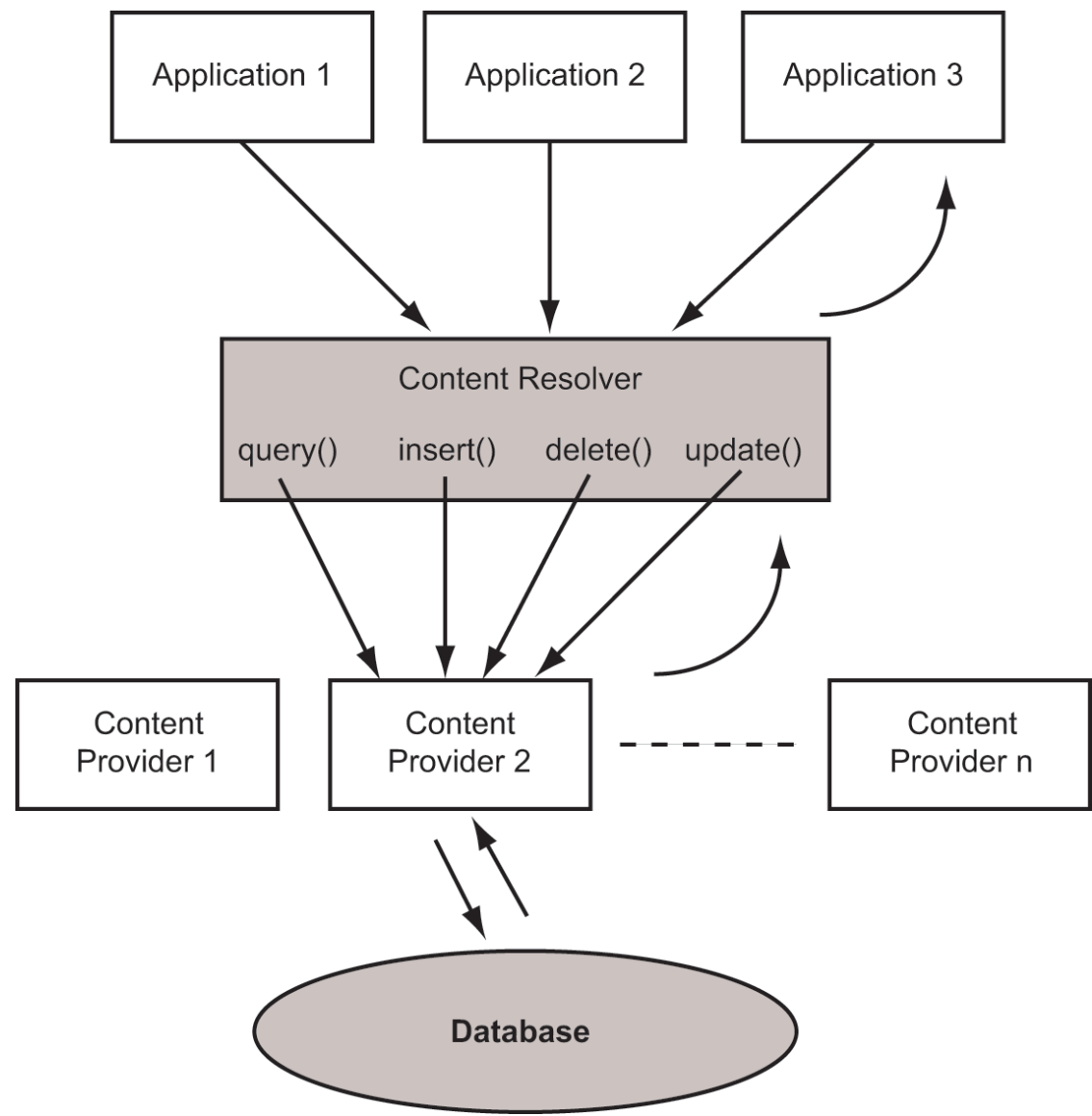
- Ο πρώτος όρος για τα URI περιεχομένου είναι “*content://*”.
- Ο δεύτερος όρος είναι συνήθως το όνομα πακέτου της εφαρμογής, για παράδειγμα “*gr.mybook.mycontentprovider*”.
- Ο τρίτος όρος καθορίζει τον τύπο δεδομένων που αναζητούμε από το συγκεκριμένο πίνακα (για παράδειγμα, σε έναν πίνακα με δεδομένα σπουδαστών “*students*”). Το URI μέχρι και τον τρίτο όρο πρέπει να είναι “*content://gr.mybook.mycontentprovider/students*”.
- Ο τέταρτος όρος καθορίζει την εγγραφή που αναζητεί ο χρήστης. Αν στον πίνακα των σπουδαστών αναζητείται η τρίτη καταχώρηση, τότε το URI θα είναι: “*content://gr.mybook.mycontentprovider/students/3*”.

- Η εφαρμογή που διαθέτει πάροχο περιεχομένου πρέπει να δημιουργήσει έναν αναλυτή περιεχομένου (*Content Resolver*) ο οποίος θα αναλύει την URI που αποστέλλεται από κάποια άλλη εφαρμογή και θα παρέχει τις μεθόδους που διαθέτει ένας πάροχος περιεχομένου.
- Η παρουσία ενός παρόχου περιεχομένου σε μια εφαρμογή δηλώνεται στο αρχείο μανιφέστου της εφαρμογής.

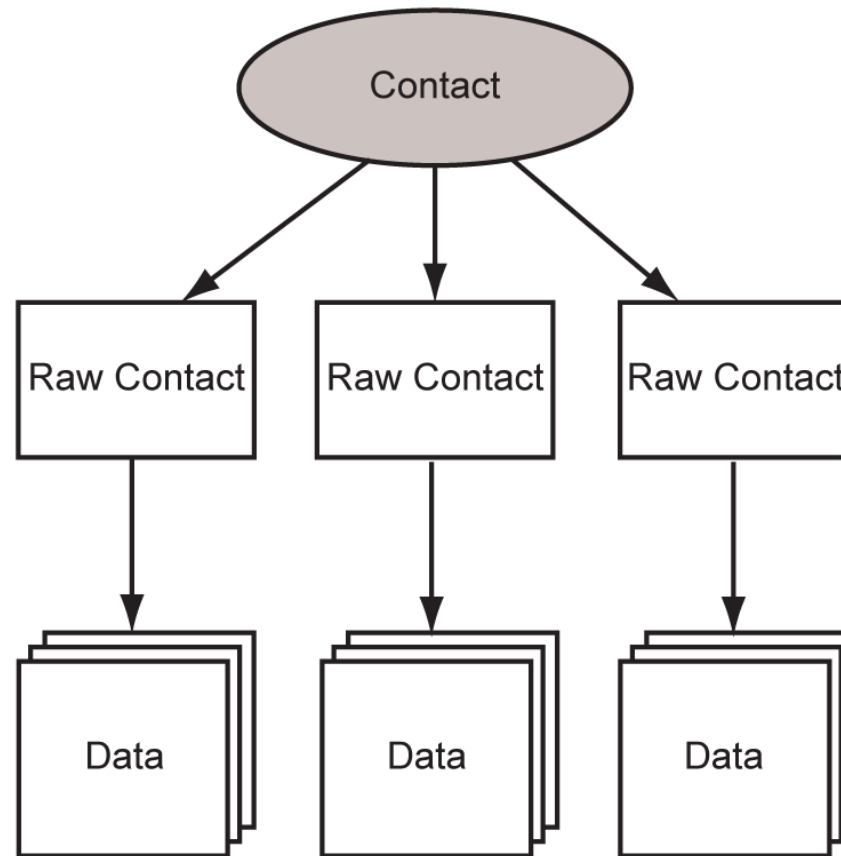


```
<provider android:name = "MyContentProvider"  
    android:authorities = "gr.mybook.mycontentprovider" />
```

- Στη δήλωση του παρόχου μπορεί να δίνονται άδειες για πρόσβαση στον πίνακα δεδομένων από άλλες εφαρμογές, όπως:
android:permission = "string"
android:readPermission = "string"
android:writePermission = "string"
- Η άδεια "*permission*" δίνει τη δυνατότητα σε μια εφαρμογή να διαβάσει ή να γράψει σε έναν πίνακα του παρόχου αν δηλώνει στο δικό της αρχείο μανιφέστου το ίδιο όνομα.
- Οι άδειες "*readPermission*" και "*writePermission*" δίνουν τη δυνατότητα μόνο ανάγνωσης ή μόνο εγγραφής και έχουν προτεραιότητα έναντι της προηγούμενης.
- Για να είναι η άδεια μοναδική για τον κάθε πάροχο, συνήθως χρησιμοποιείται η ονομασία του πακέτου, όπως για παράδειγμα "*gr.mybook.mycontentprovider.permission.READ_PROVIDER*".



Πίνακας Περιεχομένου Συσκευής (Contacts)





- Ο πίνακας *Data* αποθηκεύει όλα τα δεδομένα ενός *Raw Contact*, όπως όνομα, τηλέφωνο, email, φωτογραφία, κλπ. Η στήλη “MIME type” του πίνακα προσδιορίζει τον τύπο δεδομένων κάθε σειράς (π.χ. τηλέφωνο ή email). Οι στήλες δεδομένων του πίνακα περιέχουν δεδομένα γενικού τύπου τα οποία εξαρτώνται από τον τύπο MIME της κάθε σειράς.
- Μια σειρά του πίνακα *Raw Contact* αντιπροσωπεύει τα δεδομένα του πίνακα *Data* και άλλες πληροφορίες μιας επαφής που προέρχονται από μια συγκεκριμένη πηγή (π.χ. από το Google, το Facebook, κλπ).
- Μια σειρά του πίνακα *Contact* αποτελείται από τη συνάθροιση όλων των *Raw Contacts* μιας επαφής. Για παράδειγμα, ένας χρήστης έχει μια επαφή σαν “φίλο” και σαν “συνεργάτη” με κάποια κοινά στοιχεία (π.χ. διεύθυνση). Τα δύο *Raw Contacts* συναθροίζονται σε μια επαφή.

Ο πάροχος περιεχομένου επαφών (*android.provider.ContactsContract*) συναθροίζει τις όμοιες επαφές και τις παρουσιάζει στον χρήστη σαν μια επαφή. Το επίπεδο *Contact* είναι πρώτο στην ιεραρχία και συναθροίζει τα *Raw Contacts*. Κάθε *Raw Contact* αναφέρεται σε κάποιο μεμονωμένο δεδομένο το οποίο προέρχεται από μια συγκεκριμένη πηγή (π.χ. τη διεύθυνση gmail μιας επαφής). Τα δεδομένα των *Raw Contacts* αποθηκεύονται σε πίνακες *Data*, όπου κάθε σειρά έχει έναν τύπο MIME που δίνει πληροφορία για τον τύπο του δεδομένου που είναι αποθηκευμένο. Έτσι τα δεδομένα για έναν τηλεφωνικό αριθμό έχουν τύπο MIME “Phone.CONTENT_ITEM_TYPE”, ενώ για email ο τύπος MIME είναι “Email.CONTENT_ITEM_TYPE”.



Contacts.Contract.Contacts

_ID	Display_Name	Account_Type	-----
1			-----
2			-----
3			-----
4			-----

Contacts.Contract.RawContacts

_ID	Contact_ID	Account_Name	Account_Type	-----
1	1	this@gmail.com	com.google	-----
2	2	that@gmail.com	com.google	-----
3	3	this@yahoo.com	com.yahoo	-----
4	4	that@yahoo.com	com.yahoo	-----

Η επαφή 1 έχει ένα Raw Contact το οποίο αντιπροσωπεύει τρία δεδομένα, δύο email και έναν τηλεφωνικό αριθμό.

Contacts.Contract.Data

_ID	MIME Type	Raw_Contact_ID	Data1	Data15	Contact_ID
1	Email.CONTENT_ITEM_TYPE	1	xy@gmail.com		1
2	Phone.CONTENT_ITEM_TYPE	1	21053811000		1
3	Email.CONTENT_ITEM_TYPE	1	ab@gmail.com		1
4	Phone.CONTENT_ITEM_TYPE	2			



Μια εφαρμογή παίρνει δεδομένα από τη βάση δεδομένων ενός παρόχου περιεχομένου με τη χρήση του αντικειμένου *ContentResolver*, ο οποίος επιστρέφει το αντικείμενο *Cursor*. Ο κώδικας για την κλήση του *ContentResolver* είναι:

```
mCursor = getContentResolver().query(CONTENT_URI, mProjection,
                                     mSelectionClause, mselectionArgs, mSortOrder);
```

CONTENT_URI – Είναι το URI, η διεύθυνση, ενός πίνακα δεδομένων σε έναν πάροχο περιεχομένου, το οποίο περιλαμβάνει το όνομα του παρόχου περιεχομένου και το όνομα του πίνακα στον οποίο επιχειρείται η πρόσβαση. Στην περίπτωση του παρόχου επαφών το URI είναι *ContactsContract.Contacts.CONTENT_URI*.

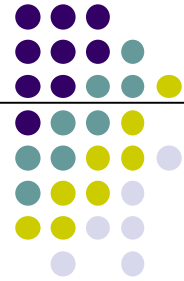
mProjection – Είναι η παράμετρος η οποία καθορίζει τις στήλες από τις οποίες θα αντληθούν δεδομένα.

mSelectionClause - Είναι η παράμετρος η οποία καθορίζει τα κριτήρια για τις σειρές από τις οποίες θα αντληθούν δεδομένα. Ο τελεστής “?” λειτουργεί σαν παράμετρος και καθορίζεται στο επόμενο βήμα.

mSelectionArgs – Είναι ένας πίνακας που περιέχει τις παραμέτρους του προηγούμενου βήματος.

mSortOrder - Είναι η παράμετρος η οποία καθορίζει τη σειρά με την οποία εμφανίζονται οι σειρές του πίνακα στο αντικείμενο *Cursor*.

Ανάκτηση τηλεφωνικού αριθμού πρώτης επαφής



```
// "Projection" defines the columns returned for each row
String[] mProjection = {
    // ContactsContract data table data1 column name
    ContactsContract.CommonDataKinds.Phone.NUMBER
};

// One-element String array to contain the selection argument
String[] mSelectionArgs = {""};
// Get data for 1st record
mSelectionArgs[0] = 1;

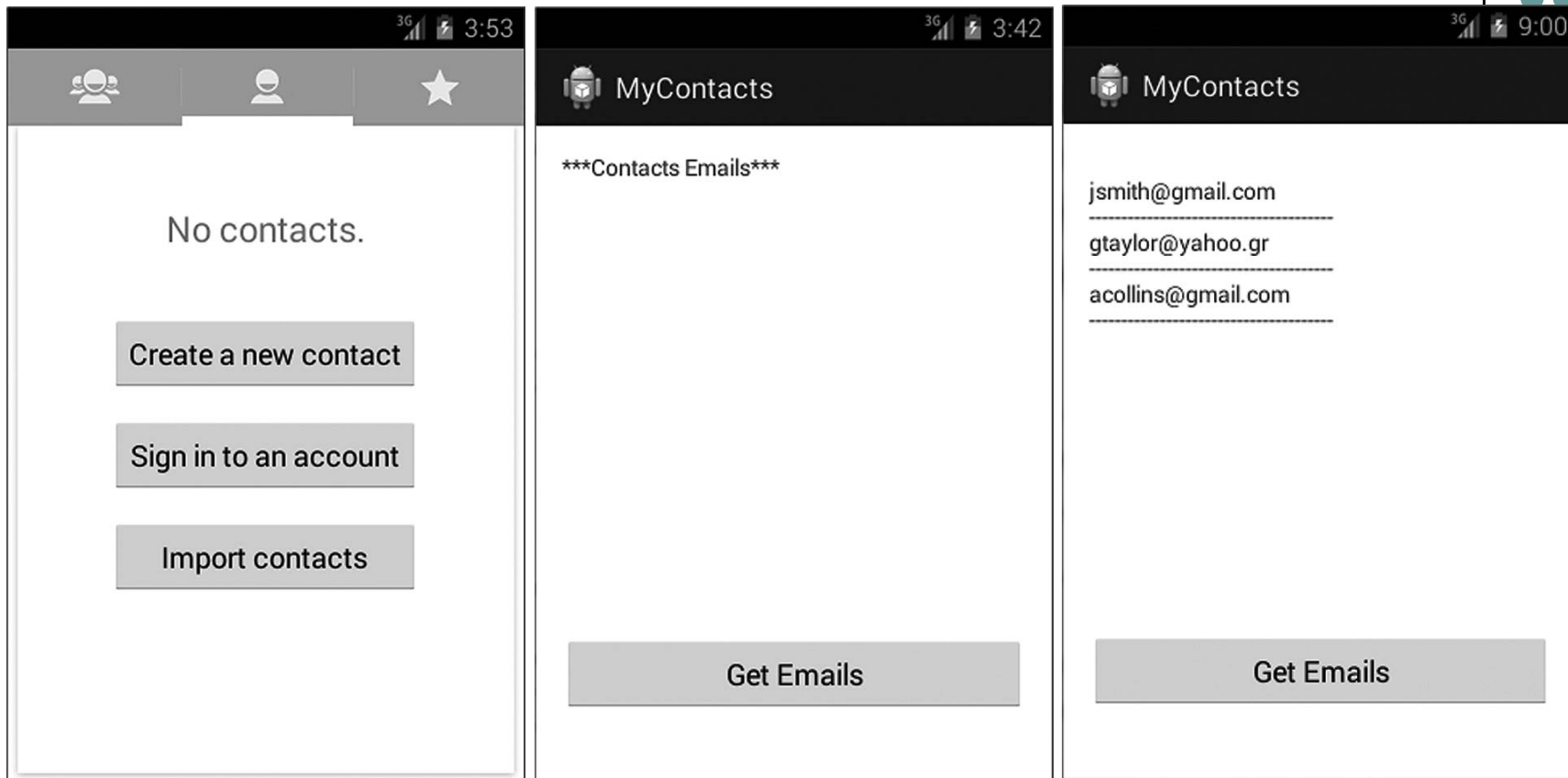
// Selection clause that matches contact_id from data table.
mSelectionClause = ContactsContract.CommonDataKinds.Phone.CONTACT_ID + " = ?";

// Query the table and returns a Cursor object
mCursor =
getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
mProjection,          // The columns to return for each row
mSelectionClause,     // Data for first contact
mSelectionArgs,      // First contact
mSortOrder);         // The sort order for the returned rows
```

```
// If the Cursor is empty, the provider found no matches
if (mCursor.getCount() > 0) {
    while (mCursor.moveToNext()) {
        // Gets the value from the column.
        phoneNumber = mCursor.getString(index);
        // Show phone number in Logcat
        Log.i("PhoneNumbers", "Numbers : " + phoneNumber);
    }
} else {
    Log.i("PhoneNumbers", "Phone Numbers not found");
}
```



Παράδειγμα “MyContacts”



Εισάγουμε επαφές με τα αντίστοιχα email τους στον εξομοιωτή του Android Studio και δημιουργούμε εφαρμογή η οποία θα διαβάζει αυτά τα δεδομένα από τον πάροχο περιεχομένου του συστήματος για τις επαφές και θα τα απεικονίζει στην οθόνη.

Εργασία



A-a@gmail.com-(210) 111-1111/(210) 999-9999/

B-b@gmail.com-(210) 222-2222

C-c@gmail.com-(210) 333-3333

GET CONTACTS

Να συμπληρωθεί η εφαρμογή ώστε να εμφανίζονται σε κάθε γραμμή:

Name – email - phone