

# Dezvoltarea Aplicațiilor pe Telefoane Mobile

## Contacts

*Loredana Cret*  
*loredana.cret@yahoo.com*

*Sl.dr.ing. Adriana STAN*

# Tema proiectului

Gestionarea unei liste de contacte. Aplicatia trebuie sa ofere lista tuturor contactelor introduse in aplicatie, cu posibilitate de cautare si sortare dupa diverse campuri.

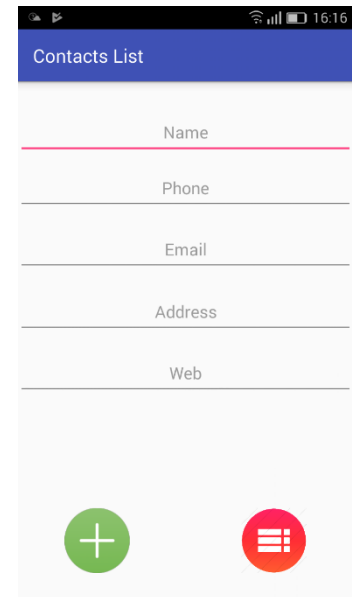
Din lista se poate alege un contact pentru vizualizare/editare. In modul vizualizare anumite campuri vor oferi integrare cu alte servicii.

De exemplu click pe campul adresa va deschide harta, click pe adresa de email va deschide aplicatia de email, click pe numarul de telefon va suna la numarul respectiv, click pe url va deschide un browser, etc.

## Moduri de utilizare ale aplicatiei

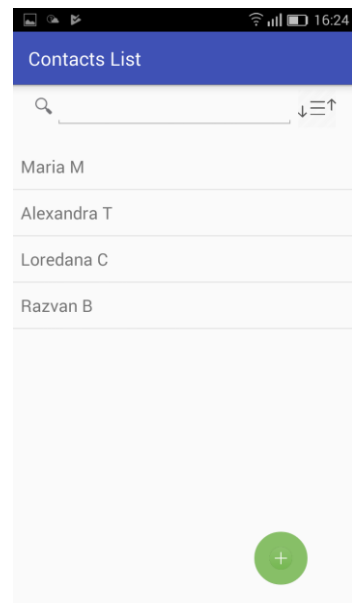
### Introducere de noi contacte

- Campuri obligatorii: Name, Phone. Se va afisa un mesaj informativ in cazul in care se incearca salvarea contactului fara aceste doua campuri completate.
- Validari:
  - Phone, trebuie sa contina doar cifre sau sa aibe formatul: +phone number
  - Email, trebuie sa respecte formatul: \*@\*.\*
- Butonul Plus adauga inregistrarea in baza de date.
- Cu ajutorul butonului Lista se realizeaza navigarea catre contactele deja existente.



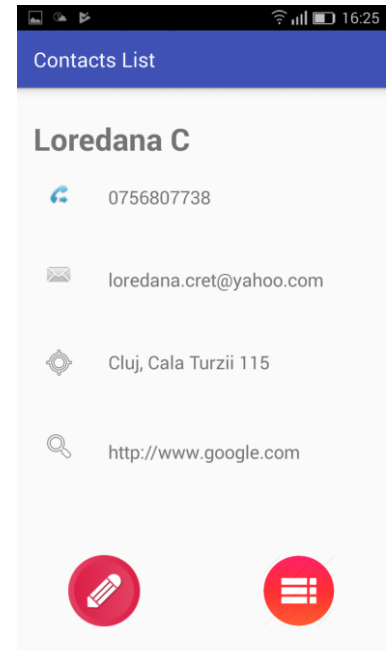
### Vizualizarea listei contactelor existente

- Lista cu toate contactele, nu exista nici o sortare prestabilita
- Bara de cautare in lista, aceasta cautare se face dupa campul nume
- Buton de sortare a listei, sortarea se realizeaza dupa campul nume
- vizualizarea detaliilor unui contact prin acesarea inregistrarii corespunzatoare acelui contact
- exista posibilitatea de introducere a unui nou contact prin apasare butonului Plus



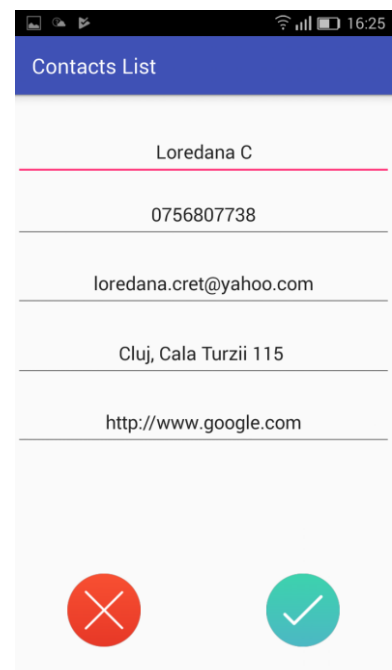
## Vizualizare contact

- in modul vizualizare sunt posibile urmatoarele actiuni:
  - apelare
  - trimitere email,
  - vizualizare pe harta a adresei
  - accesarea paginii web asociata cotactului
- aceste actiuni se realizeaza prin apasarea icoanei coraspunzatoare fiecarui camp
- se poate naviga inapoi la lista de contacte prin apasare butonului Lista
- se poate acesa modul editare al contactului curent, prin apasarea butonului Edit



## Editare contact

- in modul editare se pot aduce modificari tuturor campurilor
  - se aplica aceleasi reguli de validare ca si in cazul unui contact nou
  - se poate sterge contactul current, in acest caz un mesaj de dialog este afisat, prin care se cere acordul de a sterge contactul din baza de date
- dupa oricare din actiuni, stergere sau salvare, utilizatorul este redirectionat catre lista actualizata a contactelor si este afisat un mesaj de informare in urma actiunii realizate



# Descriere secvente cod

Creare lista:

```
//get the data and append to a list
Cursor data = mDatabaseHelper.getData(query);
ArrayList<String> listData = new ArrayList<>();
while(data.moveToNext()){
    //get the value from the database in column 1
    //then add it to the ArrayList
    listData.add(data.getString(1));
}
//create the list adapter and set the adapter
adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, listData);
mListView.setAdapter(adapter);
-----
```

Metoda ce primeste ca parametru un query string, folosita pentru crearea ninitiaa a listei, filtrare si sortare:

```
public Cursor getData(String q){
    SQLiteDatabase db = this.getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_NAME + q;
    Cursor data = db.rawQuery(query, null);
    return data;
}
```

Navigare spre pagina ce contine lista tuturor contactelor

```
btnViewData.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, ListDataActivity.class);
        startActivity(intent);
    }
});
```

Adaugare unui nu contact

```
public void AddData(String[] newEntry) {
    boolean insertData = mDatabaseHelper.addData(newEntry);

    if (insertData) {
        toastMessage("Data Successfully Inserted!");
    } else {
        toastMessage("Something went wrong");
    }
}
-----

public boolean addData(String[] contact) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
```

```

        contentValues.put(COL2, contact[1]);
        contentValues.put(COL3, contact[2]);
        contentValues.put(COL4, contact[3]);
        contentValues.put(COL5, contact[4]);
        contentValues.put(COL6, contact[5]);

        Log.d(TAG, "addData: Adding " + contact[1] + " to " + TABLE_NAME);

        long result = db.insert(TABLE_NAME, null, contentValues);

        //if date is inserted incorrectly it will return -1
        if (result == -1) {
            return false;
        } else {
            return true;
        }
    }
}

```

## Integrare cu alte aplicatii

```

public void phoneButton(View v) {
    View.OnClickListener listSet = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent callIntent = new Intent(Intent.ACTION_CALL);
            callIntent.setData(Uri.parse("tel:" + selectedPhone));
            if (ActivityCompat.checkSelfPermission(ViewDataActivity.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                toastMessage("You do not have permissions");
                return;
            }
            startActivity(callIntent);
        }
    };
    ImageButton phoneBtn = (ImageButton) findViewById(R.id.phoneImageButton);
    phoneBtn.setOnClickListener(listSet);
}

public void emailButton(View v) {
    View.OnClickListener listSet = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(Intent.ACTION_VIEW);
            Uri data = Uri.parse("mailto:"
                + selectedEmail
                + "?subject=" + "test" + "&body=" + "test");
            intent.setData(data);
            startActivity(intent);
        }
    };
    ImageButton emailBtn = (ImageButton) findViewById(R.id.emailImageButton);
    emailBtn.setOnClickListener(listSet);
}

public void addressButton(View v) {
    View.OnClickListener listSet = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Uri mapUri = Uri.parse("geo:0,0?q=" + Uri.encode(selectedAddress));
            Intent mapIntent = new Intent(Intent.ACTION_VIEW, mapUri);

```

```

        mapIntent.setPackage("com.google.android.apps.maps");
        startActivity(mapIntent);
    }
};
ImageButton addressBtn = (ImageButton) findViewById(R.id.addressImageButton);
addressBtn.setOnClickListener(listSet);
}

public void webButton(View v) {
    View.OnClickListener listSet = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent browserIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse(selectedWeb));
            startActivity(browserIntent);
        }
    };
    ImageButton webBtn = (ImageButton) findViewById(R.id.webImageButton);
    webBtn.setOnClickListener(listSet);
}
}

```

## Cautare in lista

```

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String s) {
        query = " WHERE name LIKE '%" + s + "%'";
        populateListView(query);
        return false;
    }

    @Override
    public boolean onQueryTextChange(String s) {
        if(s.equals("")) {
            query = "";
            populateListView(query);
        }
        return false;
    }
}
}

```

## Sortare

```

sort.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String q;
        asc = !asc;
        if(asc){
            q = query + " ORDER BY name ASC";
        }else{
            q = query + " ORDER BY name DESC";
        }
        populateListView(q);
    }
});
}

```