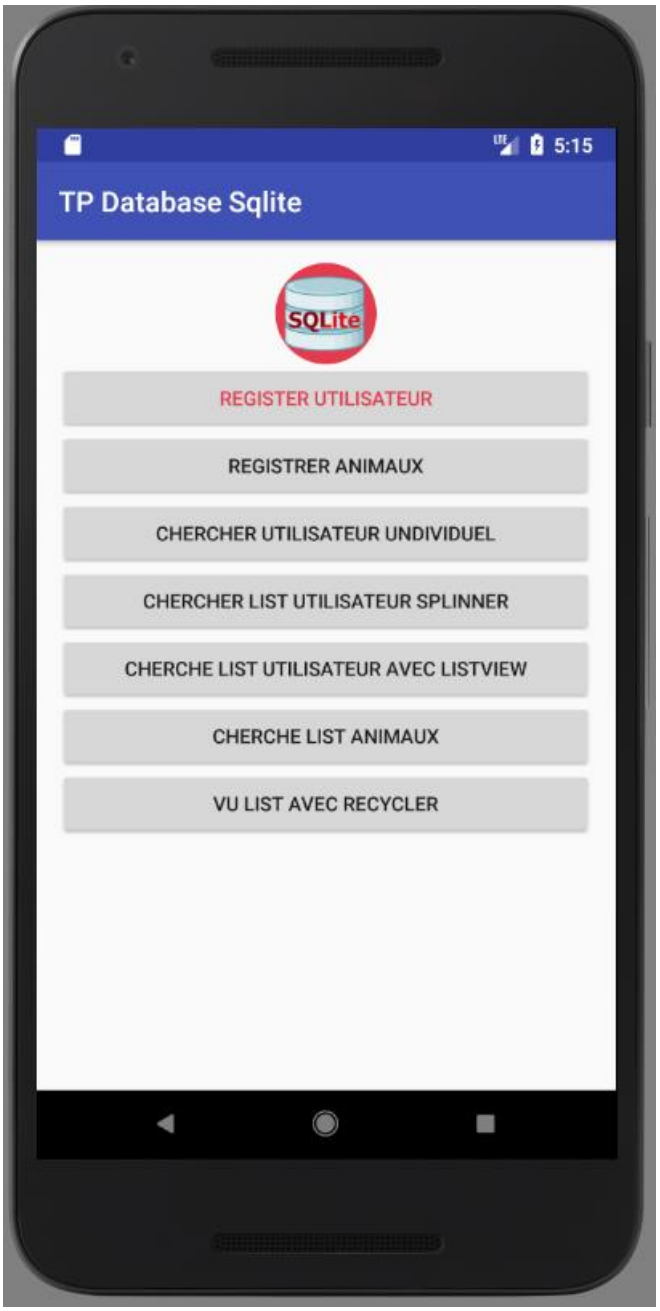
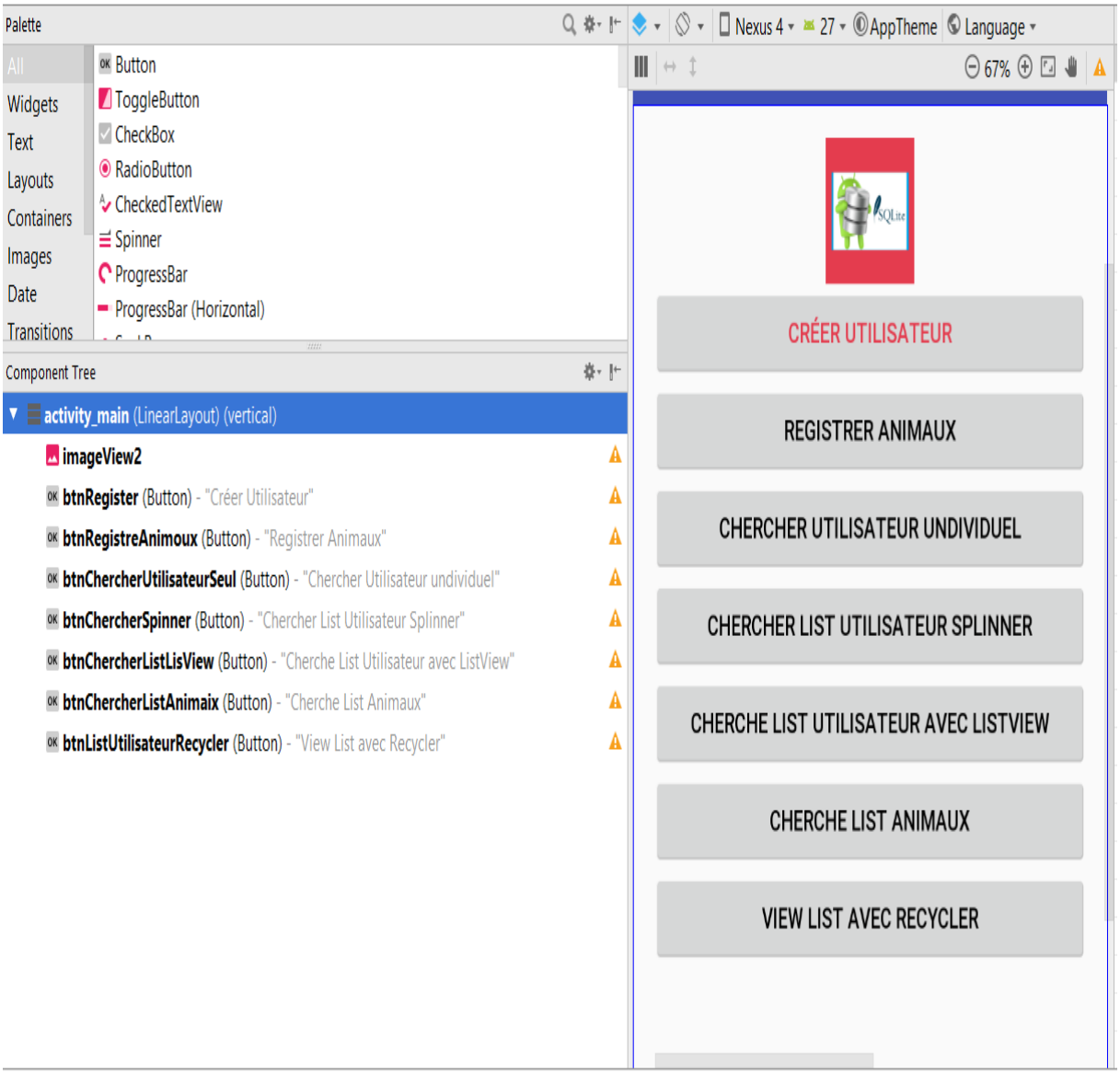
Cours Développement Mobil Android Studio 3
Séance :
Master EII 1ère année : Groupe 1 et 2.

Crée un projet Android studio
Choisir Activity : Empty Activity.

IHM Home app.



Structure sur le RES → Layout → MainActivity.xml



Img 2: Design Home Page

Code :

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    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"
    tools:context="fr.tp.tpdatabasesqlite.QueryUsersActivity">
```

```
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@mipmap/ic_launcher_round" />

    <Button
        android:id="@+id/btnRegister"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="Créer Utilisateur"
        android:textColor="@color/ic_launcher_background" />



</LinearLayout>
```

Info : correction des error dimen :
Cree un fichier sur :
res → values → dimen.xml

Ajointe :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <!-- Default screen margins, per the Android Design guidelines. -->
  <dimen name="activity_horizontal_margin">16dp</dimen>
  <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

Cree une Class : votre_nom_Pakage -> Dossier : JAVA -> User.java

//Variable à crée :

id, name, telephon.

Generate les Methodes GETTER & SETTER

Generate les Construtor

```
Users
1  package fr.tp.tpdatabasesqlite;
2
3  import java.io.Serializable;
4
5  /**
6   Created by wilian on 22/11/2017.
7   */
8
9  public class Users implements Serializable {
10
11     // Definition variables
12     private Integer id;
13     private String prenom;
14     private String telephon;
15     private String adresse;
16
17     //Constructor generate
18     public Users(Integer id, String prenom, String telephon, String adresse) {
19         this.id = id;
20         this.prenom = prenom;
21         this.telephon = telephon;
22         this.adresse = adresse;
23     }
24 // Auto-gerene Getter and setter
25     public Integer getId() { return id; }
28
29     public void setId(Integer id) { this.id = id; }
32
33     public String getPrenom() { return prenom; }
36
37     public void setPrenom(String prenom) { this.prenom = prenom; }
40
41     public String getTelephon() { return telephon; }
```

Img3 class User.java

Créer une class : ConexionSQLite.java :

```
package fr.tp.tp_2_database_sqlite;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import fr.tp.tp_2_database_sqlite.DatabaseSQLite;

import java.sql.DatabaseMetaData;

/**
 * Created by wilia on 24/11/2017.
```

```
 */

public class ConexionSQLite extends SQLiteOpenHelper{
   /**
    * Create a helper object to create, open, and/or manage a database.
    * This method always returns very quickly.  The database is not actually
    * created or opened until one of {@link #getWritableDatabase} or
    * {@link #getReadableDatabase} is called.
    *
    * @param context to use to open or create the database
    * @param name    of the database file, or null for an in-memory database
    * @param factory to use for creating cursor objects, or null for the default
    * @param version number of the database (starting at 1); if the database is older,
    *              {@link #onUpgrade} will be used to upgrade the database; if the database is
    *              newer, {@link #onDowngrade} will be used to downgrade the database
    */
   public ConexionSQLite(Context context, String name, SQLiteDatabase.CursorFactory factory, int
version) {
      super(context, name, factory, version);
   }

   /**
    * Called when the database is created for the first time. This is where the
    * creation of tables and the initial population of the tables should happen.
    *
    * @param db The database.
    */
   @Override
   public void onCreate(SQLiteDatabase db) {

      db.execSQL(DatabaseSQLite.TABLE_USER);
      // TP 2.2 db.execSQL(DatabaseMetaDataCREEATE_TABLE_Animaux);
   }

   /**
    * Called when the database needs to be upgraded. The implementation
    * should use this method to drop tables, add tables, or do anything else it
    * needs to upgrade to the new schema version.
    * <p>
    * <p>
    * The SQLite ALTER TABLE documentation can be found
    * <a href="http://sqlite.org/lang_altertable.html">here</a>. If you add new columns
    * you can use ALTER TABLE to insert them into a live table. If you rename or remove columns
    * you can use ALTER TABLE to rename the old table, then create the new table and then
    * populate the new table with the contents of the old table.
    * </p><p>
    * This method executes within a transaction.  If an exception is thrown, all changes
    * will automatically be rolled back.
    * </p>
    *
    * @param db        The database.
    * @param oldVersion The old database version.
```

```
     * @param newVersion The new database version.
     */
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        db.execSQL("DROP TABLE IF EXISTS " + DatabaseSQLite.TABLE_USER);
        //db.execSQL("DROP TABLE IF EXITS " + DatabaseMetaData.TABLE_ANIMAUX);
        onCreate(db);
    }
}
```

Créer une class : DatabaseSQLite :

```
package fr.tp.tp_2_database_sqlite;

/**
 * Created by wilia on 24/11/2017.
 */

public class DatabaseSQLite {
    // create database and col
    public static final String TABLE_USER = "userTable";
    public static final String CHAMP_id = "id";
    public static final String CHAMP_prenom = "prenon";
    public static final String CHAMP_telephon = "telephon";


}
```

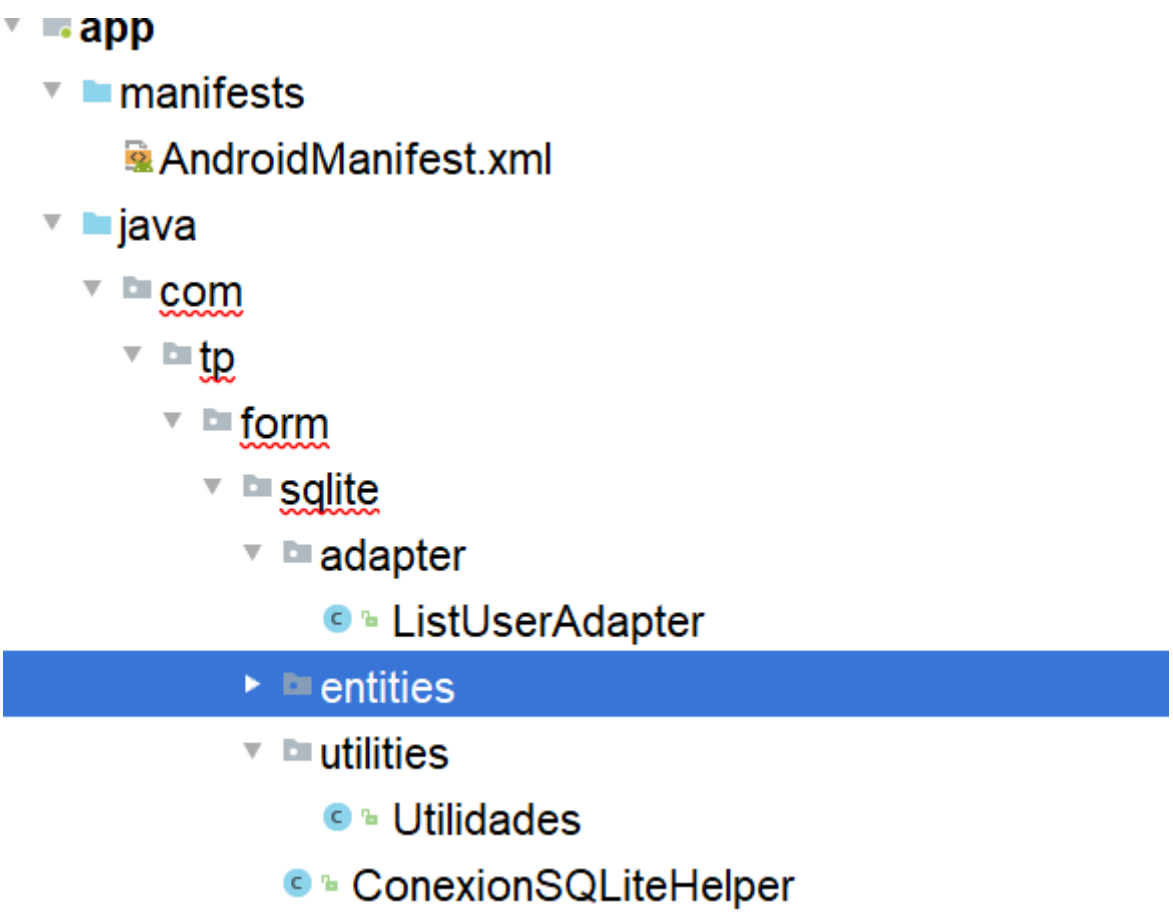
Modification sur le fichier AndroidManifest.xml

Ajointe :

```
<activity android:name="NOM_PAKAGE.NOM_TON_ACTIVITY" />
```

```
<activity android:name="fr.tp.tpdatabasesqlite.RegisterUsersActivity" />
```

Répertoire : TP 2



Class `ConexionSQLiteHelper.java`

```
public class ConexionSQLiteHelper extends SQLiteOpenHelper {



    public ConexionSQLiteHelper(Context context, String name,
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(Utilidades.CREAR_TABLA_USER);
        db.execSQL(Utilidades.CREAR_TABLA_PETS);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int versionAntigua, int
versionNueva) {


        db.execSQL("DROP TABLE IF EXISTS "+Utilidades.TABLA_USER);

      //  db.execSQL("DROP TABLE IF EXISTS "+Utilidades.TABLA_PETS);
        onCreate(db);
    }
}
```

Class utility.java

```
public class Utility {

    //Constant CHAMP table user
    public static final String TABLA_USUARIO="user";
    public static final String CAMPO_ID="id";
    public static final String CAMPO_NOMBRE="name";
    public static final String CAMPO_TELEFONO="telef";

    public static final String CREAR_TABLA_USUARIO="CREATE TABLE " +
            ""+TABLA_USER+" ("+CAMPO_ID+" " +
            "INTEGER, "+CAMPO_NOMBRE+" TEXT,"+CAMPO_TELEF+" TEXT)";

    //ConstanT CHAMP table PETS/animaux
    public static final String TABLA_PETS="pets";
    public static final String CHAMP _ID_PETS="id_pets";
    public static final String CHAMP _NAME_PETS="name_pets";
    public static final String CHAMP _RACE_PETS="race_pets";
    public static final String CHAMP _ID_PROPITIERO="id_propietier";

    public static final String CREAR_TABLA_PETS="CREATE TABLE " +
            ""+TABLA_PETS+" ("+CHAMP_ID_PETS+" INTEGER PRIMARY KEY
AUTOINCREMENT, "
            +CHAMP_NAME_PETS+" TEXT, "+CHAMP_RACE_PETS+"
TEXT,"+CHAMP_ID_PROPITIER+" INTEGER)";

}
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

        ConexionSQLiteHelper conn=new ConexionSQLiteHelper(this,"bd_users",null,1);
    }

    public void onClick(View view) {
        Intent miIntent=null;
        switch (view.getId()){
            case R.id.btnOptionRegistry:
                miIntent=new Intent(MainActivity.this,RegistryUsersActivity.class);
                break;
            case R.id.btnRegistryPets:
                miIntent=new Intent(MainActivity.this,RegistryPetsActivity.class);
                break;
/*
            case R.id.btnQuery1
                miIntent=new Intent(MainActivity.this,QueriesUsersActivity.class);
                break;
            case R.id.btnConsultaSpinner:
                miIntent=new Intent(MainActivity.this,QueriesComboActivity.class);
                break;
            case R.id.btnQueryList:
                miIntent=new Intent(MainActivity.this,QueriesListViewActivity.class);
                break;
            case R.id.btnQueryListPets:
                miIntent=new Intent(MainActivity.this,ListPetsActivity.class);
                break;
            case R.id.btnQueryListUserRecycler:
                miIntent=new Intent(MainActivity.this,ListUserRecycler.class);
*/
                break;
        }
        if (miIntent!=null){
            startActivity(miIntent);
        }

    }
}
```

Class pets.java / Class animaux.java

A définir :

Variable à déclare :
Name Pets / Nom Animaux
TP 4 → Type Pets (chien / chat)
TP 4 → Sex Pets
Owener info (primary Key) DataBase.

Prendre d'exemple la classe (User.java) dont il faut crée des variables pour après AUTO-GENERED les méthodes :

- GETTER
- SETTER

Note :

- Dans votre clase ConexionSQLiteHelper.java il faut crée la table pets

```
Exemple : //  db.execSQL(Utilidades.CREAR_TABLA_PETS);
```

- Classe Utility.java il faut définir la table pets

TABLE	PETS
CHAMP	ID
CHAMP	NAME
CHAMP	RACE
CHAMP	ID OWNER

```
//Constant champ table pets
public static final String TABLA……..
public static final String CHAMP……..
.
.
.
```