Android Lab Assignment #5

Simple Contact Book

JianyuShen

JIS77@pitt.edu

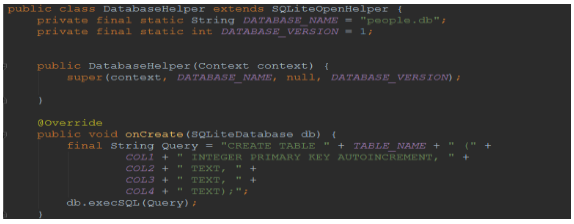
1.Introduction

SQL can be implemented within SQLite inside the Android system and many kinds of skills and experience from SQL development can be implement through Android SQLite, the most typical kinds of operation included insert, delete, display, update data.

2.Essential Section Design

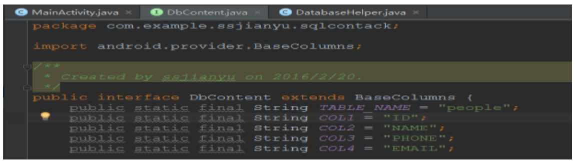
2.1 Create Database

Add a java class named "DatabaseHelper" and create database name and content:



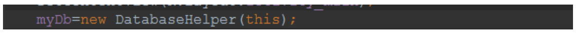
```
public class DatabaseHelper extends SQLiteOpenHelper {
    private final static String DATABASE_NAME = "people.db";
    private final static int DATABASE_VERSION = 1;


    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);

    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        final String Query = "CREATE TABLE " + TABLE_NAME + " (" +
                COL1 + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COL2 + " TEXT, " +
                COL3 + " TEXT, " +
                COL4 + " TEXT);";
        db.execSQL(Query);
    }
```

Then create a interface named "DbContent" to identify the content.



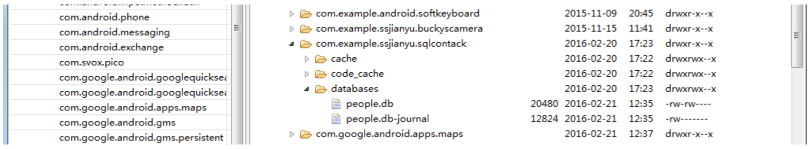
```
package com.example.ssjianyu.sqlcontack;

import android.provider.BaseColumns;

/**
 * Created by ssjianyu on 2016/2/20.
 */
public interface DbContent extends BaseColumns {
    public static final String TABLE_NAME = "people";
    public static final String COL1 = "ID";
    public static final String COL2 = "NAME";
    public static final String COL3 = "PHONE";
    public static final String COL4 = "EMAIL";
```

Finally, define a database named "myDb" to use the database.



```
myDb=new DatabaseHelper(this);
```

Check: Open the Android Device Monitor to check whether the database has been created:



```
com.android.phone
com.android.messaging
com.android.exchange
com.svox.pico
com.google.android.googlequicksea
com.google.android.googlequicksea
com.google.android.apps.maps
com.google.android.gms
com.google.android.gms.persistent
```

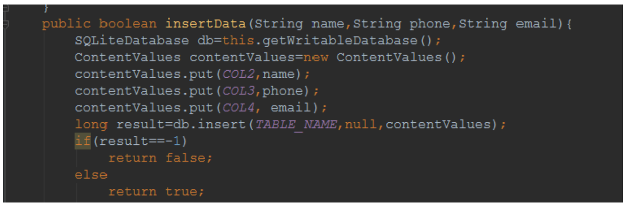
```
▷ com.example.android.softkeyboard                 2015-11-09  20:45  drwxr-x--x
▷ com.example.ssjianyu.buckyscamera                2015-11-15  11:41  drwxr-x--x
◢ com.example.ssjianyu.sqlcontack                  2016-02-20  17:23  drwxr-x--x
   ▷ cache                                         2016-02-20  17:22  drwxrwx--x
   ▷ code_cache                                    2016-02-20  17:22  drwxrwx--x
   ◢ databases                                     2016-02-20  17:23  drwxrwx--x
        people.db                           20480  2016-02-21  12:35  -rw-rw----
        people.db-journal                   12824  2016-02-21  12:35  -rw-------
▷ com.google.android.apps.maps                     2016-02-21  12:37  drwxr-x--x
```

2.2 Add "Insert " function

2.2.1 Add "Insert " Button on layout and set on ClickListener.

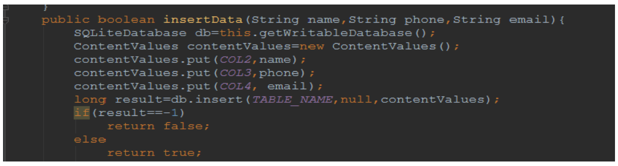
2.2.2 Java code for Insert funtion

Codeing the "Insert " function:



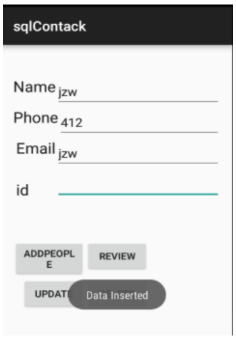
```java
    }
    public boolean insertData(String name,String phone,String email){
        SQLiteDatabase db=this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put(COL2,name);
        contentValues.put(COL3,phone);
        contentValues.put(COL4, email);
        long result=db.insert(TABLE_NAME,null,contentValues);
        if(result==-1)
            return false;
        else
            return true;
```

Declare the "Insert" function in Mainactivity:



```java
    }
    public boolean insertData(String name,String phone,String email){
        SQLiteDatabase db=this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put(COL2,name);
        contentValues.put(COL3,phone);
        contentValues.put(COL4, email);
        long result=db.insert(TABLE_NAME,null,contentValues);
        if(result==-1)
            return false;
        else
            return true;
```

2.2.3 "Insert" function test

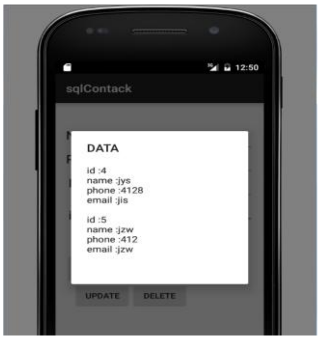


Name jzw

Phone 412

Email jzw

id



DATA

id :4
name :jys
phone :4128
email :jis

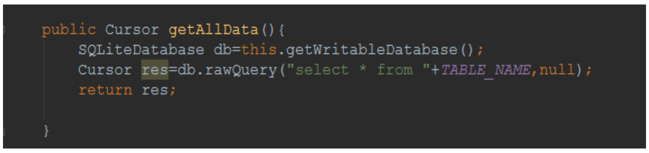
id :5
name :jzw
phone :412
email :jzw

2.3 Add "ViewAll" function

2.3.1 Add "ViewAll " Button on layout and set on ClickListener.

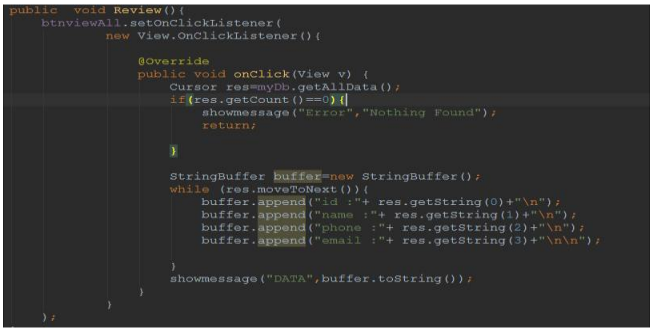
2.3.2 Java code for Insert funtion

Codeing the "ViewAll " function:



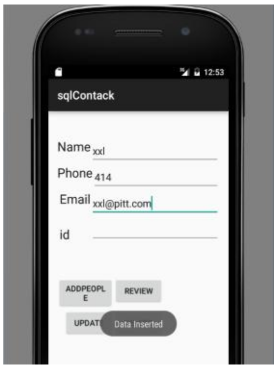
```java
    public Cursor getAllData(){
        SQLiteDatabase db=this.getWritableDatabase();
        Cursor res=db.rawQuery("select * from "+TABLE_NAME,null);
        return res;

    }
```

Declare the "ViewAll" function in Mainactivity:



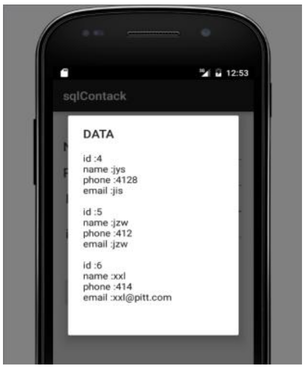
```java
public  void Review(){
    btnviewAll.setOnClickListener(
            new View.OnClickListener(){

                @Override
                public void onClick(View v) {
                    Cursor res=myDb.getAllData();
                    if(res.getCount()==0){
                        showmessage("Error","Nothing Found");
                        return;

                    }

                    StringBuffer buffer=new StringBuffer();
                    while (res.moveToNext()){
                        buffer.append("id :"+ res.getString(0)+"\n");
                        buffer.append("name :"+ res.getString(1)+"\n");
                        buffer.append("phone :"+ res.getString(2)+"\n");
                        buffer.append("email :"+ res.getString(3)+"\n\n");

                    }
                    showmessage("DATA",buffer.toString());
                }
            }
    );
```

2.3.3 "ViewAll" function test



Name xxl
Phone 414
Email xxl@pitt.com
id

Data Inserted



DATA

id :4
name :jys
phone :4128
email :jis

id :5
name :jzw
phone :412
email :jzw

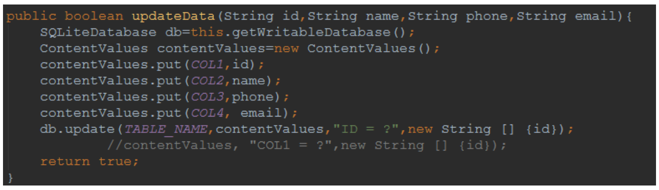
id :6
name :xxl
phone :414
email :xxl@pitt.com

2.4 Add "Update" function

2.4.1 Add "Update " Button on layout and set on ClickListener.

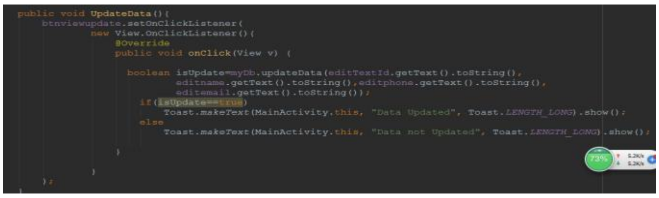
2.4.2 Java code for Insert funtion

Codeing the "Update " function:



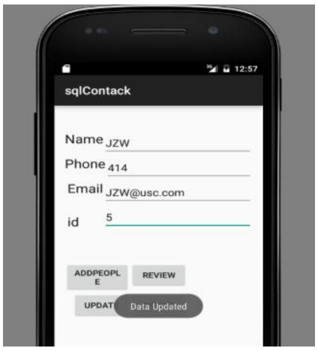
```
public boolean updateData(String id,String name,String phone,String email){
    SQLiteDatabase db=this.getWritableDatabase();
    ContentValues contentValues=new ContentValues();
    contentValues.put(COL1,id);
    contentValues.put(COL2,name);
    contentValues.put(COL3,phone);
    contentValues.put(COL4, email);
    db.update(TABLE_NAME,contentValues,"ID = ?",new String [] {id});
          //contentValues, "COL1 = ?",new String [] {id});
    return true;
}
```

Declare the "Update" function in Mainactivity:



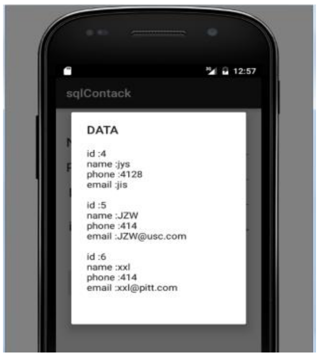
```
public void UpdateData(){
    btnviewupdate.setOnClickListener(
            new View.OnClickListener(){
                @Override
                public void onClick(View v) {

                    boolean isUpdate=myDb.updateData(editTextId.getText().toString(),
                            editname.getText().toString(),editphone.getText().toString(),
                            editemail.getText().toString());
                    if(isUpdate==true)
                        Toast.makeText(MainActivity.this, "Data Updated", Toast.LENGTH_LONG).show();
                    else
                        Toast.makeText(MainActivity.this, "Data not Updated", Toast.LENGTH_LONG).show();

                }

            }
    );
}
```

2.4.3 "Update" function test



Name JZW
Phone 414
Email JZW@usc.com
id 5

Data Updated



DATA

id :4
name :jys
phone :4128
email :jis

id :5
name :JZW
phone :414
email :JZW@usc.com

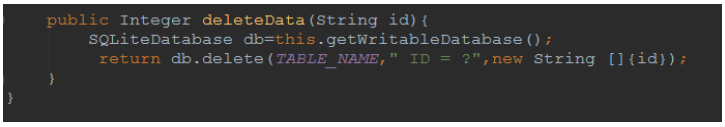
id :6
name :xxl
phone :414
email :xxl@pitt.com

2.5 Add "Delete" function

2.5.1 Add "Delete " Button on layout and set on ClickListener.

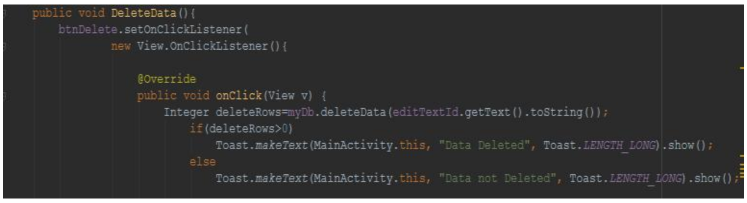
2.5.2 Java code for Insert funtion

Codeing the "Delete " function:



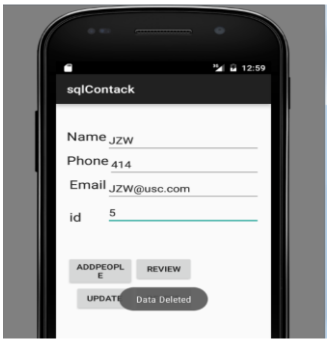
```
    public Integer deleteData(String id){
        SQLiteDatabase db=this.getWritableDatabase();
         return db.delete(TABLE_NAME," ID = ?",new String [] {id});
    }
}
```

Declare the "Delete" function in Mainactivity:



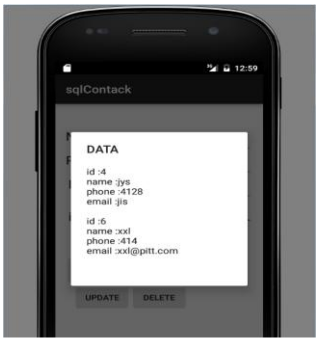
```
public void DeleteData(){
    btnDelete.setOnClickListener(
            new View.OnClickListener(){

                @Override
                public void onClick(View v) {
                    Integer deleteRows=myDb.deleteData(editTextId.getText().toString());
                        if(deleteRows>0)
                            Toast.makeText(MainActivity.this, "Data Deleted", Toast.LENGTH_LONG).show();
                        else
                            Toast.makeText(MainActivity.this, "Data not Deleted", Toast.LENGTH_LONG).show();
```

2.5.3 "Delete" function test



Name JZW
Phone 414
Email JZW@usc.com
id 5

Data Deleted



DATA

id :4
name :jys
phone :4128
email :jis

id :6
name :xxl
phone :414
email :xxl@pitt.com

3.Conclusion

Android SQLite is a excellent database to manage data, by adopting some basic concepts of the storage solutions in the Android system, some more complex application can be built in the Android system with storage content.

4.Reference

https://www.youtube.com/watch?v=cp2rL3sAFmI
https://www.youtube.com/watch?v=LZ8kJg4Pg4Y
https://www.youtube.com/watch?v=5_-knAoiVMI
https://www.youtube.com/watch?v=38DOncHIazs