通讯录的设计与实验

软件名称：通讯录

完成人：周浩瀚

学号：20182131092

完成时间：2020 年 12 月 19 日

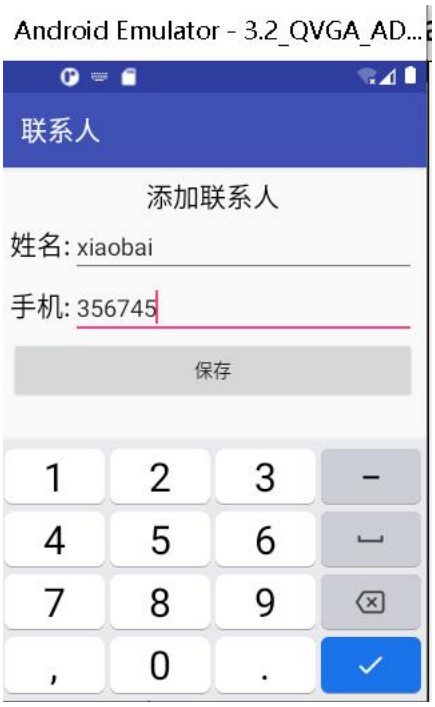
一．软件内容简介：

这是一个基于 kotlin 的简单通讯录。该应用程序能够实现联系人查询。通讯录生成列表，并选择电话号码实现拨打电话等功能。风格简洁，使用简单。

额外设计和实现了通讯录增删查改的功能。

该软件基于 ContentProvider 的方法实现。

二．界面设计：



Android Emulator - 3.2_QVGA_AD...

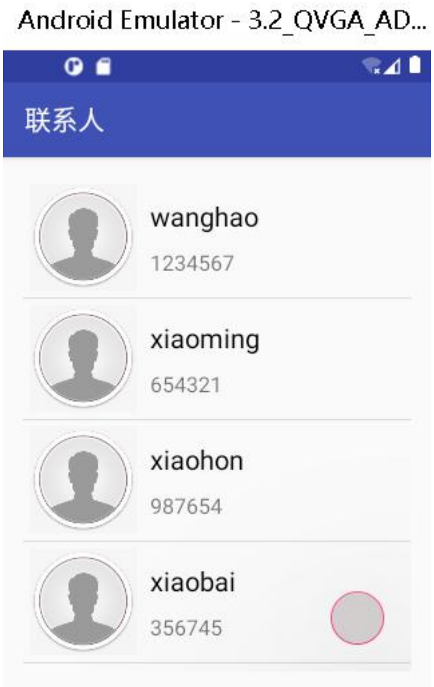
联系人

添加联系人

姓名: xiaobai

手机: 356745

保存



Android Emulator - 3.2_QVGA_AD...

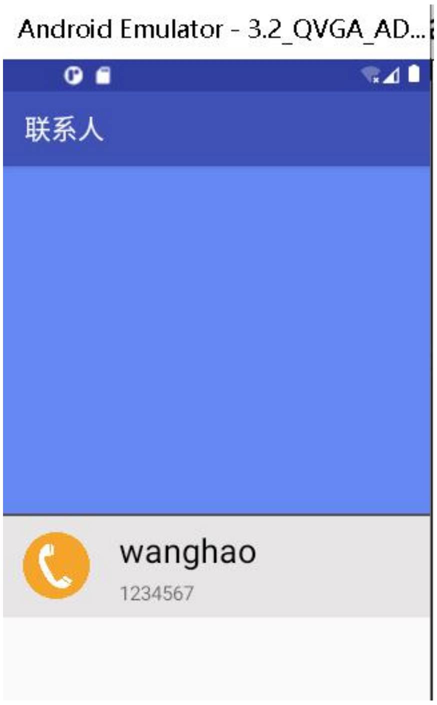
联系人

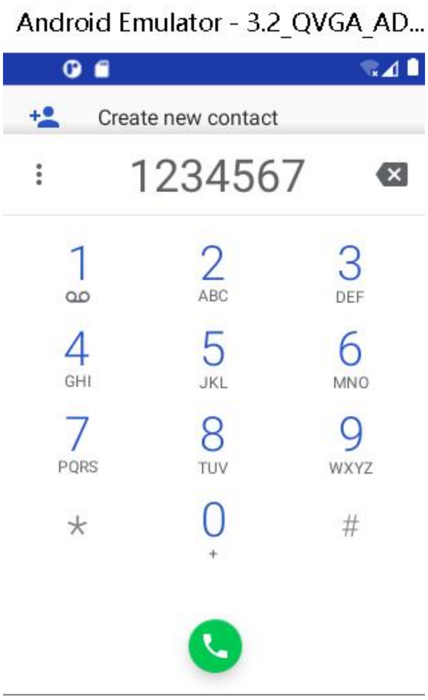
wanghao
1234567

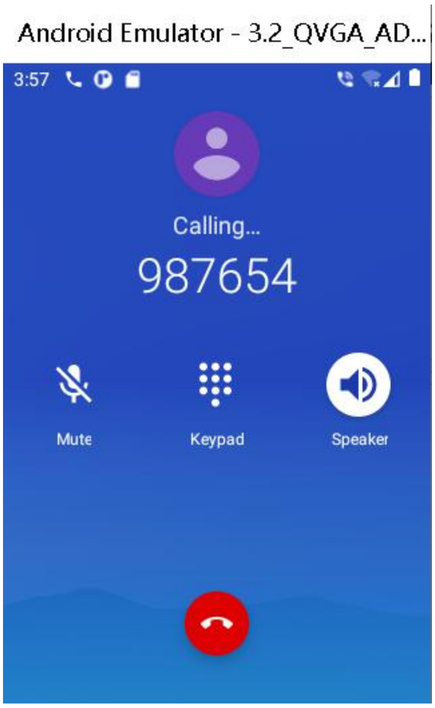
xiaoming
654321

xiaohon
987654

xiaobai
356745







三．主要代码设计：

布局代码：

Activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
        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:orientation="vertical"
       android:padding="15dp"
       tools:context="com.zpx.contact.MainActivity">
    <android.support.design.widget.FloatingActionButton
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:id="@+id/fab"
           android:layout_margin="20dp"
           android:layout_gravity="bottom|end"
           android:background="@color/colorPrimaryDark"
           android:backgroundTint="#d2cece"
           android:src="@drawable/img_add"
           app:elevation="150dp"/>

    <ListView
           android:id="@+id/lv_list"
           android:layout_width="match_parent"
           android:layout_height="match_parent"
           android:textSize="20sp"/>
</FrameLayout>
```

Add_contact:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
             android:id="@+id/activity_main"
             android:layout_height="match_parent"
             android:layout_width="match_parent"
             android:orientation="vertical"
             android:padding="8dp">

    <TextView
           android:layout_width="match_parent"
           android:layout_height="wrap_content"
           android:text="添加联系人"
           android:gravity="center"
           android:textColor="#000000"
           android:textSize="20sp"/>

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

```
        android:orientation="horizontal">

    <TextView
            android:id="@+id/tva_name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="姓名:"
            android:textColor="#000000"
            android:textSize="20sp"/>

    <EditText
            android:id="@+id/eta_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="请输入姓名"/>
</LinearLayout>

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

    <TextView
            android:id="@+id/tva_phoneNum"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="手机:"
            android:textColor="#000000"
            android:textSize="20sp"/>

    <EditText
            android:id="@+id/eta_phoneNum"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="请输入手机号"
            android:inputType="number"/>
</LinearLayout>

<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
    <Button
        android:id="@+id/btn_add"
```

```
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="保存"/>
    </LinearLayout>
</LinearLayout>
```

Contact_list:

```
<?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="wrap_content"
              android:gravity="center_vertical"
              android:padding="5dp">

    <ImageView
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:src="@drawable/img_head"/>

    <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="vertical">

        <TextView
                android:id="@+id/tvl_name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginBottom="10dp"
                android:textColor="#000000"
                android:textSize="20sp"/>
        <TextView
                android:id="@+id/tvl_phoneNum"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="16sp"/>
    </LinearLayout>
</LinearLayout>
```

Detail_info:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
          android:layout_height="match_parent"
          android:layout_width="match_parent"
          android:orientation="vertical">

<ImageView
        android:layout_width="match_parent"
        android:layout_height="260dp"
        android:background="#6588f5"/>

<!-- 下划线 -->
<ImageView
        android:layout_width="match_parent"
        android:layout_height="2dp"
        android:background="#545454"/>

<LinearLayout
        android:id="@+id/call"
        android:layout_width="match_parent"
        android:orientation="horizontal"
        android:background="#e7e5e5"
        android:layout_height="wrap_content">

    <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_weight="2">

        <ImageView
                android:layout_width="120dp"
                android:layout_height="50dp"
                android:src="@drawable/img_call"
                android:layout_marginTop="12dp"
                android:layout_marginLeft="-20dp"/>
    </LinearLayout>

    <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:orientation="vertical"
            android:layout_weight="5">

        <TextView
```

```
                    android:id="@+id/tvi_name"
                    android:layout_width="120dp"
                    android:layout_marginTop="8dp"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="5dp"
                    android:textColor="#000000"
                    android:layout_marginLeft="10dp"
                    android:textSize="25sp"/>

            <TextView
                    android:id="@+id/tvi_phoneNum"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="8dp"
                    android:layout_marginLeft="10dp"
                    android:textSize="15sp"/>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

Redact_contact:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:id="@+id/activity_main"
              android:layout_height="match_parent"
              android:layout_width="match_parent"
              android:orientation="vertical"
              android:padding="8dp">
    <EditText
            android:id="@+id/etu_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="联系人姓名"/>

    <EditText
            android:id="@+id/etu_phoneNum"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="联系方式"
            android:inputType="number"/>

    <Button
            android:id="@+id/btn_save"
            android:layout_width="0dp"
```

```
           android:layout_height="wrap_content"
           android:layout_weight="1"
           android:text="保存"/>
</LinearLayout>
```

逻辑代码：

Mainactivity:

```
package com.zpx.contact
import android.Manifest
import android.app.AlertDialog
import android.content.ContentValues
import android.content.Intent
import android.content.pm.PackageManager
import android.database.Cursor
import android.os.Bundle
import android.support.design.widget.FloatingActionButton
import android.support.v4.app.ActivityCompat
import android.support.v4.content.ContextCompat
import android.support.v7.app.AppCompatActivity
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.*
import android.widget.AdapterView.OnItemClickListener
import android.widget.AdapterView.OnItemLongClickListener
import java.util.*

class MainActivity : AppCompatActivity(), View.OnClickListener {
    private val tvlName: TextView? = null
    private val tvlPhoneNum: TextView? = null
    private var etuName: EditText? = null
    private var etuPhoneNum: EditText? = null
    private val etQueryByName: EditText? = null
    private var btnSave: Button? = null
    private val btnQuery: Button? = null
    private val tvList: TextView? = null
    private var listView: ListView? = null

    //数据源
    private val datas: MutableList<Contact> = ArrayList<Contact>()

    //自定义适配器对象
    var adapter = MyAdapter()
    //Data Access Object:对 contact 表进行 CRUD 操作的类
    private var contactDao: ContactDao? = null
    private var contact: Contact? = null
    private var cursor: Cursor? = null
    private var name: String? = null
    private var phoneNum: String? = null
    protected fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        listView = findViewById(R.id.lv_list)
        contactDao = ContactDao(this)
        //动态权限申请
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_CONTACTS)
                !== PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.READ_CONTACTS), 1)
        } else {
            readContacts()
        }
        //给 ListView 添加点击事件监听器
        listView!!.onItemClickListener = OnItemClickListener { parent, view,
position, id ->

            //联系人详情页面
            val contact: Contact = adapter.getItem(position) as Contact
            name = contact.getName()
            phoneNum = contact.getNumber()
            val intent = Intent(this@MainActivity, PeopleActivity::class.java)
            intent.putExtra("name", name)
            intent.putExtra("phoneNum", phoneNum)
            startActivity(intent)
        }

        //给 ListView 添加长按事件
        listView!!.onItemLongClickListener = OnItemLongClickListener { parent,
view, position, id ->
            val updateByName: String = datas[position].getName()
            val edit1 = AlertDialog.Builder(this@MainActivity)
            edit1.setTitle(datas[position].getName())
            edit1.setItems(arrayOf("编辑", "删除")) { dialog, which ->
                when (which) {
                    0 -> {
                        val addcontact = AlertDialog.Builder(this@MainActivity)
                        addcontact.setTitle("编辑联系人")
                        val redactView: View =
getLayoutInflater().inflate(R.layout.redact_contact, null)
                        etuName = redactView.findViewById(R.id.etu_name)
                        etuPhoneNum = redactView.findViewById(R.id.etu_phoneNum)
                        btnSave = redactView.findViewById(R.id.btn_save)
                        btnSave.setVisibility(View.GONE)
                        etuName.setText(datas[position].getName())
                        etuPhoneNum.setText(datas[position].getNumber())
                        addcontact.setView(redactView)
                        addcontact.setPositiveButton("确定") { dialog, which ->
                            val values = ContentValues()
                            contact.setName(etuName.getText().toString().trim { it
<= ' ' })

contact.setNumber(etuPhoneNum.getText().toString().trim { it <= ' ' })
                            values.put("name", contact.getName())
                            values.put("phonenumber", contact.getNumber())
                            val update: Int = contactDao.update(values, updateByName)
                            if (update > 0) {
                                Toast.makeText(this@MainActivity, "修改成功",
Toast.LENGTH_SHORT).show()
                            } else {
                                Toast.makeText(this@MainActivity, "修改失败",
Toast.LENGTH_SHORT).show()
                            }
                            readContacts()
                            adapter.notifyDataSetChanged()
                        }
                        addcontact.setNegativeButton("取消", null)
                        addcontact.create().show()
                    }
                    1 -> {
                        val edit3 = AlertDialog.Builder(this@MainActivity)
                        edit3.setTitle("删除联系人")
                        edit3.setTitle("确定删除  " +
datas[position].getName().toString() + "  吗")
                        edit3.setPositiveButton("确定") { dialog, which ->
                            val redactView: View =
getLayoutInflater().inflate(R.layout.redact_contact, null)
                            val delete: Int =
contactDao.deleteByName(datas[position])
                            if (delete > 0) {
                                Toast.makeText(this@MainActivity, "删除成功",
Toast.LENGTH_SHORT).show()
                            } else {
                                Toast.makeText(this@MainActivity, "删除失败",
Toast.LENGTH_SHORT).show()
                            }
                            readContacts()
                            adapter.notifyDataSetChanged()
                        }
                        edit3.setNeutralButton("取消", null)
                        edit3.create().show()
                    }
                    else -> {
                    }
                }
            }
            edit1.create().show()
            true
        }

        //找到控件
        val fab: FloatingActionButton = findViewById(R.id.fab)
        //设置监听
        fab.setOnClickListener(View.OnClickListener
{ //Toast.makeText(MainActivity.this,"FAB clicked",Toast.LENGTH_SHORT).show();
            val intent = Intent(this@MainActivity, AddContactActivity::class.java)
            startActivity(intent)
        })
    }

    //查询所有联系人信息
    private fun readContacts() {
        datas.clear()
        try {
            cursor = contactDao.selectAll()
            if (cursor != null && cursor!!.count > 0) {
                while (cursor!!.moveToNext()) {
                    //创建一个联系人对象
                    contact = Contact()
                    //获取联系人姓名
                    val name = cursor!!.getString(1)
                    //获取联系人手机号码
                    val number = cursor!!.getString(2)
                    //获取联系人的备注
                    //String remark = cursor.getString(2);
                    //设置联系人属性
                    //contact.setHeadImgId(headIds[i++]);
                    contact.setName(name)
                    contact.setNumber(number)
                    // contact.setRemark(remark);
                    //将联系人添加到集合中
                    datas.add(contact)
                }
                //给 ListView 设置适配器
                adapter = MyAdapter()
                listView!!.adapter = adapter
            }
        } catch (e: Exception) {
            e.printStackTrace()
        } finally {
            if (cursor != null) {
                cursor!!.close()
            }
        }
    }

    override fun onClick(v: View) {}
    fun onRequestPermissionsResult(requestCode: Int, @NonNull permisssions:
Array<String?>?,
                                @NonNull grantResults: IntArray) {
        when (requestCode) {
            1 -> if (grantResults.size > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                readContacts()
            } else {
                Toast.makeText(this, "你拒绝了权限", Toast.LENGTH_SHORT).show()
            }
        }
    }

    internal inner class MyAdapter : BaseAdapter() {
        //得到 item 的总数
        override fun getCount(): Int {
            return datas.size
        }

        //得到每个 position 位置上的 item 代表的对象
        override fun getItem(position: Int): Any {
            return datas[position]
        }

        //得到item 的id
        override fun getItemId(position: Int): Long {
            return position.toLong()
        }

        //得到item 的view 视图
        override fun getView(position: Int, convertView: View, parent: ViewGroup):
View {
            var convertView = convertView
            val holder: ViewHolder
            if (!datas.isEmpty()) {
                if (convertView == null) {
                    //利用布局填充器填充自定义的ListView item 的布局
                    convertView =
LayoutInflater.from(this@MainActivity).inflate(R.layout.contacts_list, null)
                    holder = ViewHolder()
                    //利用得到的view 来进行findViewById()
                    //holder.ivHead =convertView.findViewById(R.id.iv_head);
                    holder.tvName = convertView.findViewById(R.id.tvl_name)
                    holder.tvNumber = convertView.findViewById(R.id.tvl_phoneNum)
                    //将holder 存入view 的tag 中
                    convertView.tag = holder
                } else {
                    //在convertview 不为空的情况下，重复使用convertView
                    //并去除存储的tag
                    holder = convertView.tag as ViewHolder
                }
                //设置holder 中每个控件的内容

//holder.ivHead.setImageResource(datas.get(position).getHeadImgId());
                holder.tvName.setText(datas[position].getName())
                holder.tvNumber.setText(datas[position].getNumber())
                //最后不要忘了返回convertView
                return convertView
            }
            return null
        }

        //为了减少findViewById 的次数
        internal inner class ViewHolder {
            var ivHead: ImageView? = null
            var tvName: TextView? = null
            var tvNumber: TextView? = null
        }
    }
}
```

Contact

```
package com.zpx.contact
class Contact {
    //private int headImgId;
    //private String remark;
    var name: String? = null
    var number: String? = null

}
```

AddContactActivity:

```
package com.zpx.contact

import android.content.ContentValues
import android.content.Intent
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.text.TextUtils
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast

class AddContactActivity : AppCompatActivity(), View.OnClickListener {
    private var tvaName: TextView? = null
    private var tvaPhoneNum: TextView? = null
    private var etaName: EditText? = null
    private var etaPhoneNum: EditText? = null
    private var btnSave: Button? = null

    //Data Access Object:对contact表进行CRUD操作的类
    private var contactDao: ContactDao? = null
    private var name: String? = null
    private var phoneNum: String? = null
    protected fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.add_contact)
        btnSave = findViewById(R.id.btn_save)
        tvaName = findViewById(R.id.tva_name)
        tvaPhoneNum = findViewById(R.id.tva_phoneNum)
        contactDao = ContactDao(this)
        //初始化组件
        initView()
    }

    private fun initView() {
        tvaName = findViewById(R.id.tva_name)
        tvaPhoneNum = findViewById(R.id.tva_phoneNum)
        etaName = findViewById(R.id.eta_name)
        etaPhoneNum = findViewById(R.id.eta_phoneNum)
        btnSave = findViewById(R.id.btn_add)
        //为按钮注册监听
        btnSave!!.setOnClickListener(this)
    }

    override fun onClick(v: View) {
        name = etaName!!.text.toString().trim { it <= ' ' }
        phoneNum = etaPhoneNum!!.text.toString().trim { it <= ' ' }
        try {
            if (!TextUtils.isEmpty(name) && !TextUtils.isEmpty(phoneNum)) {
                val values = ContentValues()
                values.put("name", name)
                values.put("phonenumber", phoneNum)
                if (values != null) {
                    val number: Long = contactDao.insert(values)
                    if (number > 0) {
                        Toast.makeText(this, "添加成功", Toast.LENGTH_SHORT).show()
                        val intent = Intent(this@AddContactActivity,
MainActivity::class.java)
                        startActivity(intent)
                    }
                }
            }
        } catch (e: Exception) {
            e.printStackTrace()
        }
    }
}
```

DBHelper

```
package com.zpx.contact

import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
```

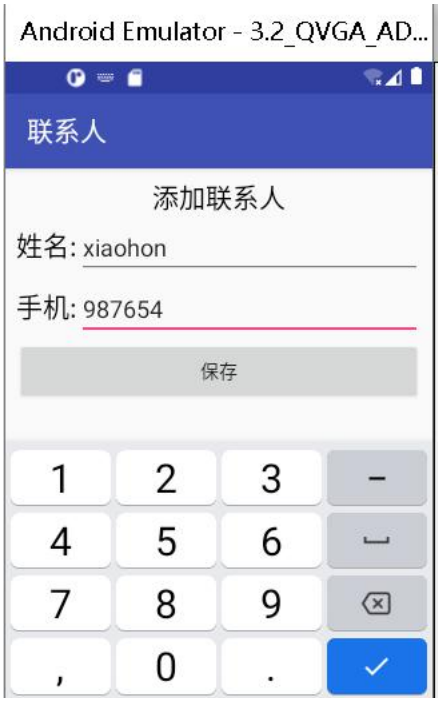
```
class DBHelper  //创建数据库
(context: Context?) : SQLiteOpenHelper(context, DB_NAME, null, DB_VERSION) {

    override fun onCreate(db: SQLiteDatabase) {
        val sql = "CREATE TABLE contact (_id INTEGER PRIMARY KEY AUTOINCREMENT, name
VARCHAR(10), phonenumber VARCHAR(12))"
        db.execSQL(sql)
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int)
{}

    companion object {
        private const val DB_NAME = "contacts.db"
        private const val DB_VERSION = 1
    }
}
```

ContactDao

```
package com.zpx.contact

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase

class ContactDao(private val mContext: Context) {
    private val helper: DBHelper

    //添加联系人
    fun insert(values: ContentValues?): Long {
        val db: SQLiteDatabase = helper.getWritableDatabase()
        val number = db.insert(TABLE_NAME, null, values)
        db.close()
        return number
    }

    //删除全部数据
    fun deleteAll(): Int {
        val db: SQLiteDatabase = helper.getWritableDatabase()
        val number = db.delete(TABLE_NAME, null, null)
        db.close()
        return number
    }
    //根据联系人姓名删除数据
    fun deleteByName(contact: Contact): Int {
        val db: SQLiteDatabase = helper.getWritableDatabase()
        val number = db.delete(TABLE_NAME, "name=?", arrayOf(contact.getName()))
        db.close()
        return number
    }

    //根据联系人姓名修改数据
    fun update(values: ContentValues?, name: String): Int {
        val db: SQLiteDatabase = helper.getWritableDatabase()
        val number = db.update(TABLE_NAME, values, "name=?", arrayOf(name))
        db.close()
        return number
    }

    //查询所有记录
    fun selectAll(): Cursor {
        val db: SQLiteDatabase = helper.getWritableDatabase()
        return db.query(TABLE_NAME, null, null, null, null, null, null)
    }

    companion object {
        private const val TABLE_NAME = "contact"
    }

    //生成DBHelper 类
    init {
        helper = DBHelper(mContext)
    }
}
```

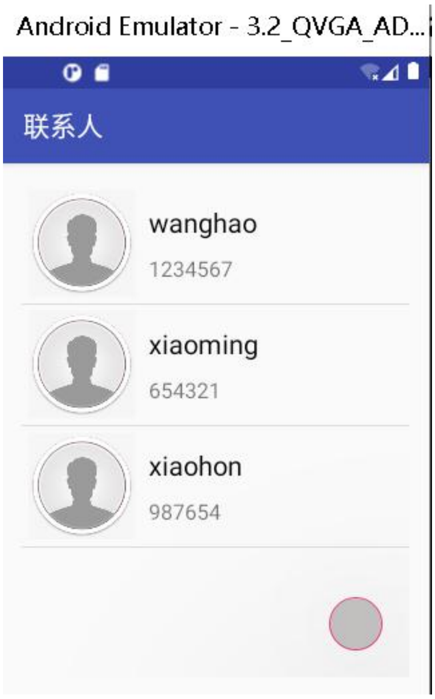
四．软件操作流程：

首先进入首页主界面 MainActivity，上面显示有所有的通讯录对象（通过 listview 实现），然后点击右下角的添加按钮可以进行添加联系人记录的操作，而添加完成后可以查看通讯录列表，并且可以选择对其进行修改或则删除操作，点击上面的联系人可以进行拨通电话的操作。

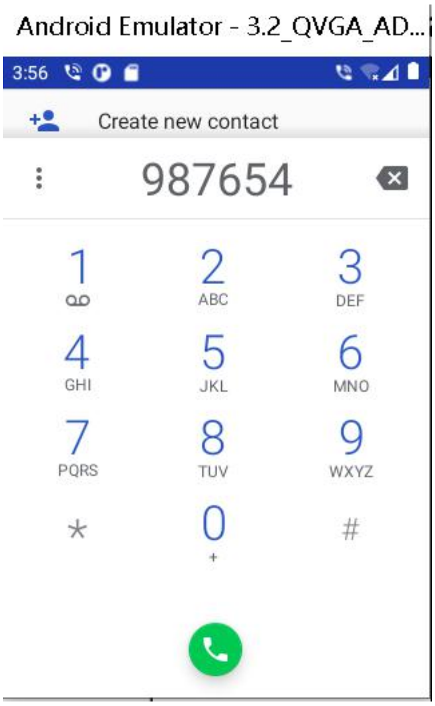


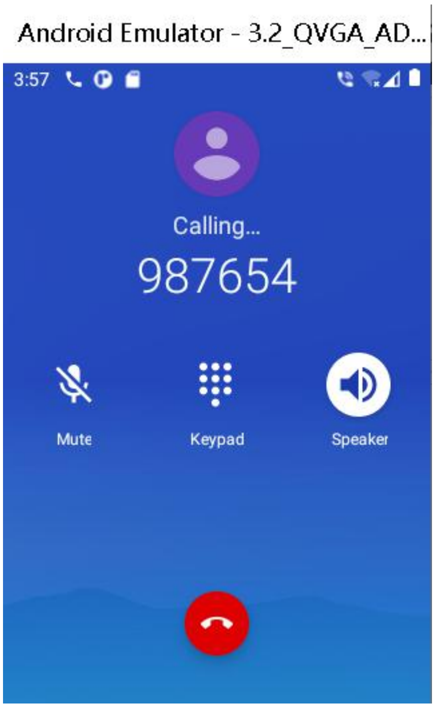
姓名: xiaohon

手机: 987654



wanghao	1234567
xiaoming	654321
xiaohon	987654





五．不足之处：

页面设计简单，不够美观。没有实现发短信的功能。

六．今后的设想：

今后可以继续完善该通讯录软件的功能，改进该软件的逻辑功能和布局，配色等。

因为时间不足的关系，没有完成发短信的功能，在今后会增加这一功能。

通过这一次的实验，让我巩固了所学的知识，掌握了 ContentProvider 方法，

实现了通讯录的功能。同时这一次的安卓实验也让我认识到了自己学习上的不足，在以后的学习我会更努力地学习安卓的相关知识。