

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称：移动应用开发

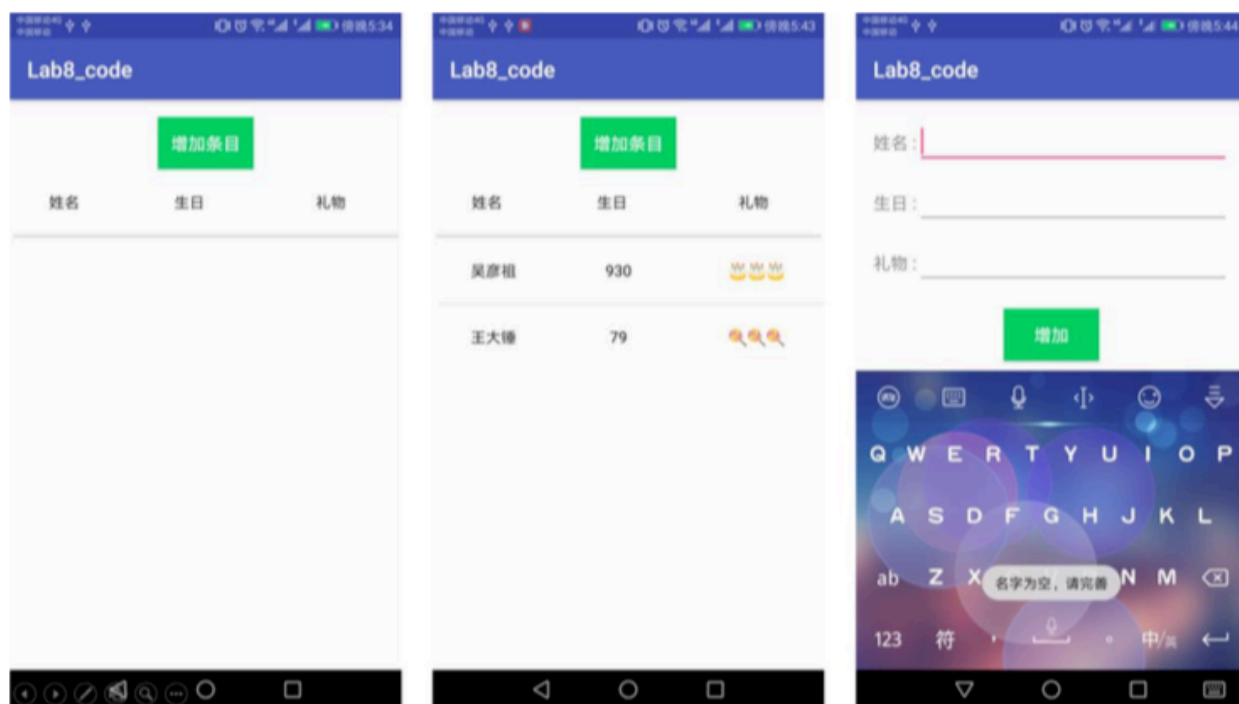
任课教师：郑贵锋

年级	2015 级	专业 (方向)	移动信息工程 (互联网方向)
学号	1535251	姓名	柯博
电话	13671412922	Email	kebo@mail2.sysu.edu.cn
开始日期		完成日期	

一、 实验题目

lab8 数据存储 (二)

二、 实现内容



从左至右：初始界面，添加一部分条目，名字不能为空；



从左至右：名字不能重复，点击条目显示信息（可修改），长按删除条目。

实现一个生日备忘录，要求实现：

使用 SQLite 数据库保存生日的相关信息，并使得每一次运行程序都可以显示出已经存储在数据库里的内容；

使用 ContentProvider 来获取手机通讯录中的电话号码。

功能要求：

- A. 主界面包含增加生日条目按钮和生日信息列表；
- B. 点击“增加条目”按钮，跳转到下一个 Activity 界面，界面中包含三个信息输入框(姓名、生日、礼物)和一个“增加”按钮，姓名字段不能为空且不能重复；
- C. 在跳转到的界面中，输入生日的相关信息后，点击“增加”按钮返回到主界面，此时，主界面中应更新列表，增加相应的生日信息；
- D. 主界面列表点击事件：
 - 点击条目：
 - 弹出对话框，对话框中显示该条目的信息，并允许修改；
 - 对话框下方显示该寿星电话号码(如果手机通讯录中有的话，如果没有就显示“无”)
 - 点击“保存修改”按钮，更新主界面生日信息列表。

长按条目：

- 弹出对话框显示是否删除条目；
- 点击“是”按钮，删除该条目，并更新主界面生日列表。

三、 课堂实验结果

（1） 实验截图

实验结果图如下，图片顺序与实验目的中的相同：





(2) 实验步骤以及关键代码

初始化数据库代码：

```
public class MyDatabase extends SQLiteOpenHelper {
    public static final String Database_name="Contacts.db";
    public static final String Table_name="Contacts";
    public static final int DB_VERSION=1;

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

    public MyDatabase(Context context, String name, int version) { this(context, name, factory: null, version); }
    public MyDatabase(Context context, String name) { this(context, name, factory: null, DB_VERSION); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_TABLE = "create table " + Table_name
            + " ("
            + "name text primary key, "
            + "birth text , "
            + "gift text);";
        db.execSQL(CREATE_TABLE);
    }
}
```

数据库操作，insert，update 和 delete 方法的接口实现：

```

public boolean insert(String name,String birth,String gift){ //传入参数，插入数据库
    SQLiteDatabase db=getWritableDatabase();
    ContentValues values=new ContentValues();
    values.put("name",name);
    values.put("birth",birth);
    values.put("gift",gift);
    Cursor cursor=db.rawQuery( sql: "select * from "+MyDatabase.Table_name, selectionArgs: null);
    if(cursor.moveToFirst()==false) {
        db.insert(Table_name, nullColumnHack: null,values);
        db.close();
        return true;
    }
    else {
        boolean flag=false;
        do{
            int name_index=cursor.getColumnIndex( columnName: "name");
            String name_tmp=cursor.getString(name_index);
            if(name_tmp.equals(name)){ //存在相同name
                flag=true;
                break;
            }
        }while (cursor.moveToNext());
        if(!flag){
            db.insert(Table_name, nullColumnHack: null,values);
            db.close();
            return true;
        }
    }
    return false;
}

public void update(String name,String birth,String gift){
    SQLiteDatabase db=getWritableDatabase();
    String whereClause= "name=?";
    String[] whereArgs={name};
    ContentValues values=new ContentValues();
    values.put("name",name);
    values.put("birth",birth);
    values.put("gift",gift);
    db.update(Table_name,values,whereClause,whereArgs);
}

public void delete(String name){
    SQLiteDatabase db=getWritableDatabase();
    String whereClause= "name=?";
    String[] whereArgs={name};
    db.delete(Table_name,whereClause,whereArgs);
}

```

增加数据库条目的 Activity，设置按钮监听函数，并判断输入合法性：

```

@Override
public void onClick(View v) {
    if(v.getId()==R.id.add){
        EditText editText1=(EditText)findViewById(R.id.name_edit);
        EditText editText2=(EditText)findViewById(R.id.birthday_edit);
        EditText editText3=(EditText)findViewById(R.id.gift_edit);
        String name=editText1.getText().toString();
        String birth=editText2.getText().toString();
        String gift=editText3.getText().toString();
        if(name.length()==0){
            Toast.makeText( context: AddInfoActivity.this, text: "姓名不可为空",Toast.LENGTH_SHORT).show();
        }
        else{
            boolean flag=MainActivity.myDatabase.insert(name,birth,gift);
            if(flag){
                AddInfoActivity.this.finish();
            }
            else Toast.makeText( context: AddInfoActivity.this, text: "姓名重复，插入失败，请重新检查",Toast.LENGTH_SHORT).show();
        }
    }
}

```

MainActivity 的 onCreate 函数。声明数据库 myDatabase，声明系统联系人的访问工具 contacts，由于在 MainActivity 中使用了 ListView 来存放数据库中的数据，因此需要设置 ListView 的 Adapter：

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    public static MyDatabase myDatabase;
    List<Map<String,String>> list=new ArrayList<>();
    SimpleAdapter simpleAdapter;
    Cursor contacts;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myDatabase=new MyDatabase( context: this,MyDatabase.Database_name);
        findViewById(R.id.add).setOnClickListener(this);
        contacts=getContentResolver().query(ContactsContract.Contacts.CONTENT_URI,
            projection: null, selection: null, selectionArgs: null, sortOrder: null);

        ListView listView=(ListView)findViewById(R.id.info);
        simpleAdapter=new SimpleAdapter( context: this,list,R.layout.list_item,new String[]{"name","birth","gift"},
            new int[]{R.id.item_name,R.id.item_birthday,R.id.item_gift});
        listView.setAdapter(simpleAdapter);
    }
}

```

设置 ListView 中每个 View 的点击事件，获取数据库中的信息，填充到对话框内对应的 View 中：

```

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    final String name=list.get(position).get("name");
    String birth=list.get(position).get("birth");
    String gift=list.get(position).get("gift");
    LayoutInflater factory=LayoutInflater.from(MainActivity.this);
    View entryView=factory.inflate(R.layout.dialoglayout, root: null);
    TextView name_view=(TextView) entryView.findViewById(R.id.name_edit);
    final EditText birth_view=(EditText) entryView.findViewById(R.id.birthday_edit);
    final EditText gift_view=(EditText) entryView.findViewById(R.id.gift_edit);
    TextView number=(TextView) entryView.findViewById(R.id.phone_number);
    name_view.setText(name);
    birth_view.setText(birth);
    gift_view.setText(gift);
}

```

通过 contacts 的遍历获取对应寿星的电话号码：

```

if(contacts.moveToFirst()==false){
    number.setText("电话: 无");
}
else{
    boolean flag=false;
    do{
        String contactId = contacts.getString(contacts.getColumnIndex(ContactsContract.Contacts._ID));
        Cursor phone = getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
            projection: null, selection: ContactsContract.CommonDataKinds.Phone.CONTACT_ID + "=" + contactId, sele
        String Number = "";
        while(phone.moveToNext()) {
            Number += phone.getString(phone.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER)) + " ";
        }
        String username = contacts.getString(contacts.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
        //Toast.makeText(MainActivity.this,username+" "+Number,Toast.LENGTH_SHORT).show();
        if(username.equals(name)){
            flag=true;
            number.setText("电话: "+Number);
        }
    }while (contacts.moveToNext());
    if(!flag) number.setText("电话: 无");
}
}

```

设置对话框，其中可以读取用户输入，调用数据库的 update 方法更新数据：

```

//设置对话框
AlertDialog.Builder dialog=new AlertDialog.Builder( context: MainActivity.this);
dialog.setTitle("(▽)~*");
dialog.setView(entryView);
dialog.setNegativeButton( text: "放弃修改", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
    }
});
dialog.setPositiveButton( text: "保存修改", (dialog, which) -> {
    String birth_new=birth_view.getText().toString();
    String gift_new=gift_view.getText().toString();
    myDatabase.update(name,birth_new,gift_new);
    UpdateListview();
});
dialog.show();

```

设置 ListView 中每个 View 的长按事件，长按弹出对话框确认是否删除信息，点击“是”则调用数据库 delete 方法删除数据：

```
listview.setOnItemLongClickListener((parent, view, position, id) -> {
    final String name=list.get(position).get("name");
    AlertDialog.Builder dialog=new AlertDialog.Builder(context: MainActivity.this);
    dialog.setTitle("是否删除");
    dialog.setNegativeButton(text: "否", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {

        }
    });
    dialog.setPositiveButton(text: "是", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            myDatabase.delete(name);
            UpdateListview();
        }
    });
    dialog.show();
    return true;
});
UpdateListview();
```

每次对 ListView 执行完操作都会调用 update 方法，重新读取一次数据库，刷新 ListView 中显示的内容：

```
public void UpdateListview(){
    SQLiteDatabase db=myDatabase.getWritableDatabase();
    Cursor cursor=db.rawQuery(sql: "select * from "+MyDatabase.Table_name, selectionArgs: null);
    if(cursor.moveToFirst()==false) return;
    else{
        list.clear();
        do{
            int name_index=cursor.getColumnIndex(columnName: "name");
            String name=cursor.getString(name_index);
            int birth_index=cursor.getColumnIndex(columnName: "birth");
            String birth=cursor.getString(birth_index);
            int gift_index=cursor.getColumnIndex(columnName: "gift");
            String gift=cursor.getString(gift_index);
            Map<String,String> tmp=new HashMap<>();
            tmp.put("name",name);
            tmp.put("birth",birth);
            tmp.put("gift",gift);
            list.add(tmp);
        }while (cursor.moveToNext());
        simpleAdapter.notifyDataSetChanged();
    }
}
```

(3) 实验遇到困难以及解决思路

遇到了闪退 bug，在 MainActivity 还没出现的时候就直接闪退了，也就是说在初始化的过程中出错了。检查 XML 文件和 Main.java 没有发现问题，最有可能是获取系统联系人时出问题了，在设置中对 app 赋予权限后问题解决。本来以为在程序运行的时候会弹出系统对话框确认权限，但转念一想我根本就还没写这个函数，自然是不存在了，如果把上次 MediaPlayer 的代码抄过来弄成动态权限的话就可以了。

四、 课后实验结果

五、 实验思考及感想

对于一个没有选数据库专选的人，WEB 和安卓都来一套 SQL 三连真的好吗？幸好这两次用的 Java（jsp 类 Java 语法）语言实现，有坑也早就排完了，数据库操作倒是顺风顺水的，反而是界面编程卡了挺久的，不喜欢设计，喜欢写实现，果然我比较适合写 **implement** 吗。。。

作业要求：

1. 命名要求: 学号_姓名_实验编号，例如 15330000_林 XX_lab1。
2. 实验报告提交格式为 pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。