

Android 开发第七次实验报告

13331251 王嘉威

一.实验目的

- 1.学会使用 SQLite 数据库保存数据。
- 2.学会对 SQLite 数据库里的数据进行增删改查等操作

二.实验内容

实现一个通讯录查看程序:

- A) 要求使用 SQLite 数据库保存通讯录,使得每次运行程序均能显示当前联系人列表。
- B) 主界面包含一个添加联系人按钮和一个联系人列表,每一项显示联系人学号,姓名,手机号码。
- C) 点击添加联系人按钮能添加新的联系人。
- D) 在次界面输入联系人信息之后点击确定按钮会返回主界面(如果输入的联系人信息有效的话,更新主界面的联系人列表)。

三.实验步骤

1)创建 MyDataBaseHelper 类,该类继承 SQLiteOpenHelper 类,重载 onCreate 方法,从中创建数据库:

```
@Override
public void onCreate(SQLiteDatabase db) {
    String CREATE_TABLE = "CREATE table " + TABLE_NAME + "(_id integer primary key autoincrement, " + "_no text not null, _name text not null, _pnumber text);";
    db.execSQL(CREATE_TABLE);
}
```

并实现其增删改查方法:

```

public long insert(Contact entity) {
    SQLiteDatabase db = getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("_no", entity.get_no());
    values.put("_name", entity.get_name());
    values.put("_pnumber", entity.get_pnumber());

    long id = db.insert(TABLE_NAME, null, values);
    db.close();
    return id;
}

public int update(Contact entity) {
    SQLiteDatabase db = getWritableDatabase();
    String whereClause = "_no = ?";
    String[] whereArgs = {entity.get_no()};

    ContentValues values = new ContentValues();
    values.put("_no", entity.get_no());
    values.put("_name", entity.get_name());
    values.put("_pnumber", entity.get_pnumber());

    int rows = db.update(TABLE_NAME, values, whereClause, whereArgs);
    return rows;
}

public int delete(Contact entity) {
    SQLiteDatabase db = getWritableDatabase();
    String whereClause = "_no = ?";
    String[] whereArgs = {entity.get_no()};

    int rows = db.delete(TABLE_NAME, whereClause, whereArgs);
    db.close();
    return rows;
}

public Cursor query() {
    SQLiteDatabase db = getWritableDatabase();
    return db.query(TABLE_NAME, null, null, null, null, null, null);
}

```

2)创建并注册 DetailActivity 类,编写对应的 detail.xml,此为联系人的创建活动页面:

```

<?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="match_parent"
    android:orientation="vertical">
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/edit_name"
        android:hint="name"/>
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/edit_pnumber"
        android:hint="Phone number"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/edit_contact"
        android:text="Edit"
        android:layout_gravity="center_horizontal"/>
</LinearLayout>

```

在

DetailActivity 类中实现创建通讯记录的事件绑定,通过将姓名,学号等信息传到 MainActivity 中,由 MainActivity 实现对数据库的创建操作:

```

submit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String no = ((EditText)findViewById(R.id.no_info)).getText().toString();
        String name = ((EditText)findViewById(R.id.name_info)).getText().toString();
        String pnumber = ((EditText)findViewById(R.id.pnumber_info)).getText().toString();
        if (no.length() != 8) {
            Toast.makeText(DetailActivity.this, "The length of no should be 8", Toast.LENGTH_SHORT).show();
            return;
        }
        if (name.length() < 4) {
            Toast.makeText(DetailActivity.this, "The length of name should be at least 4", Toast.LENGTH_SHORT).show();
            return;
        }
        if (pnumber.length() != 11) {
            Toast.makeText(DetailActivity.this, "The length of pnumber should be 11", Toast.LENGTH_SHORT).show();
            return;
        }
        Intent intent = new Intent(DetailActivity.this, MainActivity.class);
        Bundle mBundle = new Bundle();
        mBundle.putString("no", no);
        mBundle.putString("name", name);
        mBundle.putString("pnumber", pnumber);
        intent.putExtras(mBundle);
        startActivity(intent);
    }
}

```

3)创建 Contact 类,为通讯录数据库的行,编写对应的行内布局 xml 文件 contact.xml,内含_no,_name 和_pnumber 三个属性(元组):

```

public Contact(String no, String name, String pnumber) {
    this._no = no;
    this._name = name;
    this._pnumber = pnumber;
}

public String get_no() { return this._no; }
public String get_name() { return this._name; }
public String get_pnumber() { return this._pnumber; }
}

```

4)创建并注册 MainActivity 类,编写对应的 xml 函数,在类中实现 onCreate()方法,在其中

进行 ListView 的初始化(从数据库中读取数据并绑定到对定 Adapter)以及创建新联系人的 Button 控件事件的绑定.再 onStart()方法中,实现对 DetailActivity 传来的新联系人信息的处理,通过 MyDataBaseHelper 对象在数据库中创建新记录.

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

    DataBaseHelper = new MyDataBaseHelper(this);
    cursor = DataBaseHelper.query();

    dataList = new ArrayList<Map<String, String>>();
    setData(dataList);

    contactAdapter = new MySimpleAdapter(MainActivity.this, dataList, R.layout.contact, new String[] {"no", "name", "pnumber"}, new int[] {R.id.no, R.id.name, R.id.pnumber});

    contactList = new ListView(this);
    contactList.setAdapter(contactAdapter);
    contactList.setOnItemClickListener((adapterView, view, i, l) -> {
        deleteDialog = new DeleteDialog(MainActivity.this, i);
        deleteDialog.show();
    });
    contactList.setOnItemLongClickListener((parent, view, position, id) -> {
        editDialog = new EditDialog(MainActivity.this, position, ((TextView)view.findViewById(R.id.no)).getText().toString(),
        ((TextView)view.findViewById(R.id.name)).getText().toString(), ((TextView)view.findViewById(R.id.pnumber)).getText().toString());
        editDialog.show();
        return true;
    });

    Button addList = (Button)findViewById(R.id.add_contact);
    addList.setOnClickListener((v) -> {
        Intent intent = new Intent(MainActivity.this, DetailActivity.class);
        startActivity(intent);
    });

    LinearLayout main = (LinearLayout)findViewById(R.id.main);
    main.addView(contactList);
}

@Override
protected void onStart() {
    super.onStart();
    Bundle bundle = getIntent().getExtras();
    if (bundle != null) {
        String no = bundle.getString("no");
        String name = bundle.getString("name");
        String pnumber = bundle.getString("pnumber");
        Contact contact = new Contact(no, name, pnumber);
        DataBaseHelper.insert(contact);

        ArrayList<Map<String, String>> dataList = new ArrayList<Map<String, String>>();
        setData(dataList);
        contactAdapter = new MySimpleAdapter(MainActivity.this, dataList, R.layout.contact, new String[] {"no", "name", "pnumber"}, new int[] {R.id.no, R.id.name, R.id.pnumber});
        contactList.setAdapter(contactAdapter);
        contactAdapter.notifyDataSetChanged();

        Toast.makeText(MainActivity.this, "Add Success", Toast.LENGTH_SHORT).show();
    }
}
```

5)扩展项:为删除和编辑功能创建 AlertDialog 的扩展类,编写对应的 xml 文件 dialog_delete.xml 和 dialog_edit.xml,在其构造函数中绑定点击的列表项的位置及其项内的信息.最后在 onCreate()函数中绑定布局文件,并通过对其确认的按钮控件绑定点击事件来实现功能(对数据库的删除和修改操作,通过 MyDatabaseHelper 的方法实现).

```
private class EditDialog extends AlertDialog {
    private int id;
    private String no;
    private String name;
    private String pnumber;

    public EditDialog(Context context, int id, String no, String name, String pnumber) {
        super(context);
        this.id = id;
        this.no = no;
        this.name = name;
        this.pnumber = pnumber;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.dialog_edit);
        setTitle("Delete");
        Button delBtn = (Button) findViewById(R.id.edit_contact);
        ((EditText) findViewById(R.id.edit_name)).setText(name);
        ((EditText) findViewById(R.id.edit_pnumber)).setText(pnumber);
        delBtn.setOnClickListener((v) -> {
            LayoutInflater inflater = (LayoutInflater) (getApplicationContext()).getSystemService(LAYOUT_INFLATER_SERVICE);
            String name_info = ((EditText) findViewById(R.id.edit_name)).getText().toString();
            String pnumber_info = ((EditText) findViewById(R.id.edit_pnumber)).getText().toString();
            if (name_info.length() < 4) {
                Toast.makeText(MainActivity.this, "The length of name should be at least 4", Toast.LENGTH_SHORT).show();
                return;
            }
            if (pnumber_info.length() != 11) {
                Toast.makeText(MainActivity.this, "The length of name should be 11", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Please enter
better or L

四.实验结果

主界面:

学号	姓名	手机
----	----	----

创建:

Homework-7

学号: 13331251

姓名: william

手机: 18819473218

确定

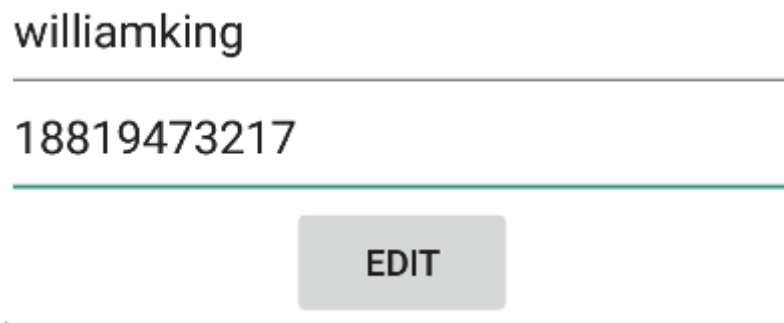
创建后:



The screenshot shows a mobile application interface. At the top, there is a blue header bar with the word "ADD" in white. Below the header, there is a table with three columns: "学号" (Student ID), "姓名" (Name), and "手机" (Phone). The first row of data contains the values "13331251", "william", and "18819473218". The status bar at the top right shows the time as 7:32.

学号	姓名	手机
13331251	william	18819473218

编辑:



The screenshot shows the edit screen of the mobile application. It features two text input fields. The first field contains the text "williamking" and has a purple underline. The second field contains the text "18819473217" and has a green underline. Below the second field is a grey button with the text "EDIT".

williamking

18819473217

EDIT

编辑后:



The screenshot shows the mobile application interface after editing. The blue header bar still has "ADD" in white. The table below now shows the updated contact information: "学号" is "13331251", "姓名" is "williamking", and "手机" is "18819473217". The status bar at the top right shows the time as 7:33.

学号	姓名	手机
13331251	williamking	18819473217

删除:



The screenshot shows a confirmation dialog box on a dark grey background. The dialog box is white and contains the text "是否删除此联系人" (Are you sure you want to delete this contact?). Below the text is a grey button with the text "确定" (Confirm).

是否删除此联系人

确定



五.实验总结

这次的实验学习了本地数据库 SQLite 的使用,还联系上了以前学过的 ListView 的运用,还有 Dialog 的使用.随着课程的进行,各次课学的东西渐渐联系到一起了.这次的实验对数据库的操作并不是很难,主要是通过建立一个 DatabaseHelper 类,重载其方法来实现对数据库的操作.反而是 ListView 的使用和 Dialog 的操作我忘了不少,又重新复习了一次,另外对 layoutinflater 类也加深了一定的了解.临近期末了,我也需要尽快去学习,争取做出一个好的期末作品.

本次实验查阅了以下资料:

1)<http://blog.csdn.net/cshichao/article/details/9333497>

ListView 数据动态刷新

2)http://blog.csdn.net/guolin_blog/article/details/12921889

Android LayoutInflater 原理分析

联系方式: 18819473218