

1.软件名称：Diary

2.完成人：杨丹璐

3.学号：20152100131

4.完成时间：2017/12/8

5.软件内容简介：

(1)列表功能：列表显示日记标题；

(2)新建日记功能：点击列表界面的悬浮按钮或者菜单项，则进入新建日记界面；

(3)查询功能：新建日记界面，点击“查询”显示所有日记列表；

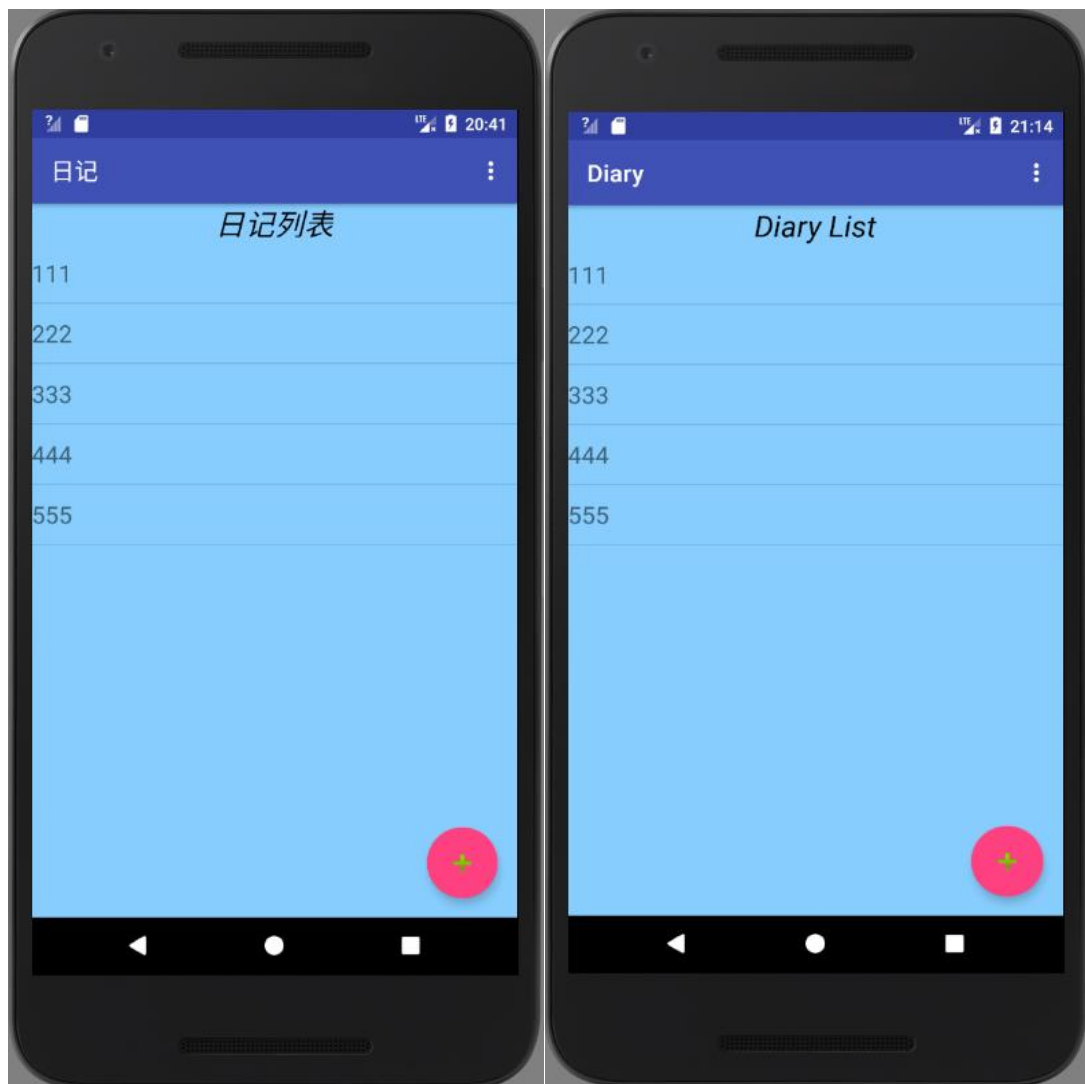
(4)查看详细信息功能：点击日记列表里的条目，显示对应标题的详细信息（包括标题、时间、内容）；

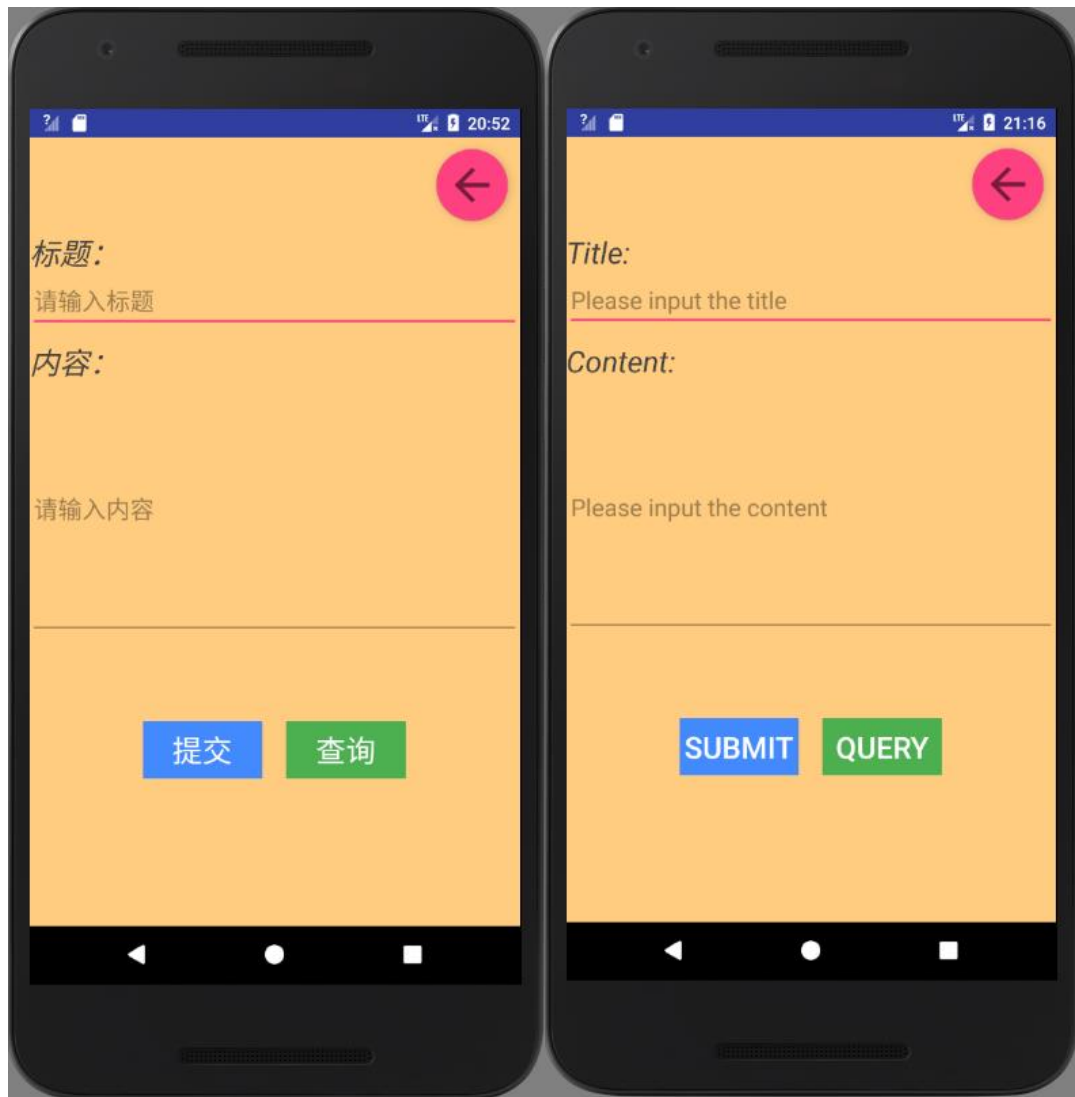
(5)修改功能：详细信息界面，直接在内容里修改，时间信息会自动更新；

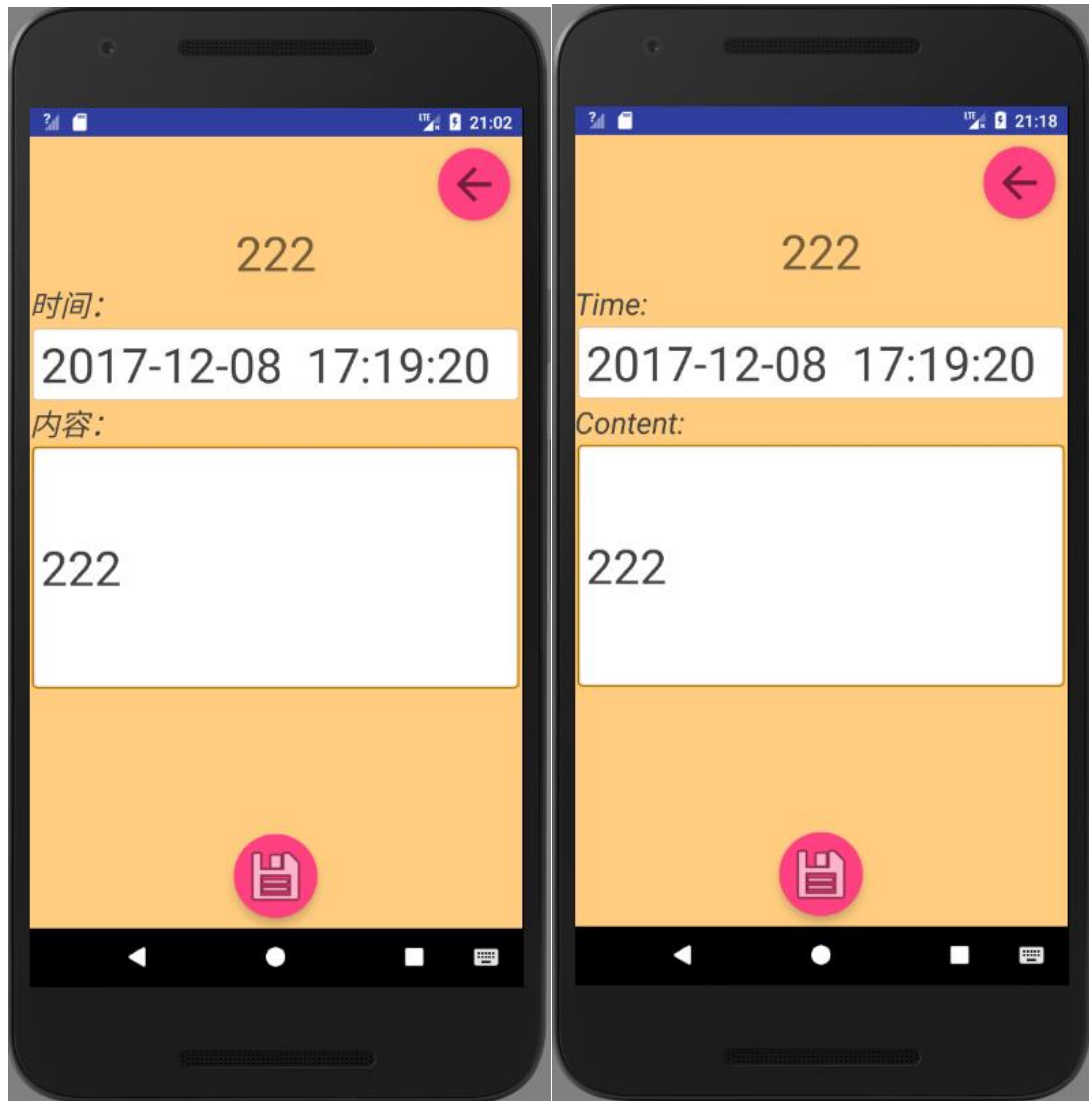
(6)长按删除功能：长按日记列表里的条目，会弹出提示对话框，选择“是”则删除对应的日记，同时更新数据库；

(7)国际化功能：手机设置为中文或英文时，打开 APP 都会对应设置的语言。

6.界面设计：







7.代码设计:

(1)主界面设计采用 `LinearLayout`，上方用一个 `TextView` 显示界面名称，下方 `ListView` 显示日记的标题，和一个悬浮按钮和一个菜单项用于跳转到新建日记界面。

界面文件如下： `//content_main.xml`

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="#87CEFF"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.echo.diary.MainActivity">
```

```

tools:showIn="@layout/activity_main">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="italic"
    android:textSize="25sp"
    android:textColor="#000000"
    android:text="@string/diary_list" />
<ListView
    android:id="@+id/lv_diary"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />
</LinearLayout>

```

(2)新建日记界面设计采用 **LinearLayout**，上方用两个 **EditText** 用于输入日记的标题和内容，右上角一个悬浮按钮用于跳转到日记列表界面，最下方两个 **Button**，一个用于保存新建的日记到数据库，另一个用于查询所有日记。

界面文件如下：//add_layout.xml

```

<?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"
    android:orientation="vertical"
    android:background="#FFCC80"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fbb1"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_margin="10dp"
        android:layout_gravity="top|end"
        android:clickable="true"
        app:fabSize="mini"
        app:srcCompat="?attr/actionModeCloseDrawable" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:textColor="#424242"
        android:textStyle="italic"

```

```
android:text="@string/title" />
```

```
<EditText
```

```
android:id="@+id/et_diary_title"
```

```
android:layout_width="fill_parent"
```

```
android:layout_height="wrap_content"
```

```
android:textColor="#424242"
```

```
android:textSize="20sp"
```

```
android:hint="@string/inputTitle"/>
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_marginTop="10dp"
```

```
android:textSize="25sp"
```

```
android:textColor="#424242"
```

```
android:textStyle="italic"
```

```
android:text="@string/content" />
```

```
<EditText
```

```
android:id="@+id/et_diary_content"
```

```
android:layout_width="fill_parent"
```

```
android:layout_height="wrap_content"
```

```
android:textColor="#424242"
```

```
android:textSize="20sp"
```

```
android:layout_marginBottom="10dp"
```

```
android:lines="8"
```

```
android:hint="@string/inputContent"/>
```

```
<LinearLayout
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_marginTop="60dp"
```

```
android:orientation="horizontal"
```

```
android:layout_gravity="center_horizontal">
```

```
<Button
```

```
android:id="@+id/bt_add"
```

```
android:layout_width="100dp"
```

```
android:layout_height="wrap_content"
```

```
android:textSize="25sp"
```

```
android:background="#448AFF"
```

```
android:textColor="FFFFFF"
```

```
android:textStyle=""
```

```
android:onClick="onClick"
```

```

        android:text="@string/submit" />
    <Button
        android:id="@+id/bt_search"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:textSize="25sp"
        android:background="#4CAF50"
        android:textColor="#FFFFFF"
        android:onClick="onClick"
        android:text="@string/query" />

</LinearLayout>

</LinearLayout>

```

(3) 详细信息界面设计采用 **LinearLayout** 和 **ScrollView**，上方用一个 **TextView** 显示日记标题，下方 **ListView** 显示日记建立的时间和内容，右上角一个悬浮按钮用于跳转到日记列表界面，最下方一个悬浮按钮用于保存更改的内容。

界面文件如下：//result.xml

```

<?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"
    android:orientation="vertical"
    android:background="#FFCC80"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fbb"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"
        android:layout_gravity="top|end"
        android:clickable="true"
        app:fabSize="mini"
        app:srcCompat="?attr/actionModeCloseDrawable" />

    <TextView
        android:id="@+id/result_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"

```

```

        android:textSize="40sp"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:textColor="#424242"
        android:textStyle="italic"
        android:text="@string/time"/>

    <TextView
        android:id="@+id/result_time"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="40sp"
        android:textColor="#424242"
        android:background="@android:drawable/editbox_background"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:textColor="#424242"
        android:textStyle="italic"
        android:text="@string/content"/>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="205dp">
        <EditText
            android:id="@+id/result_content"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:lines="4"
            android:textSize="40sp"
            android:textColor="#424242"
            android:background="@android:drawable/editbox_background"/>

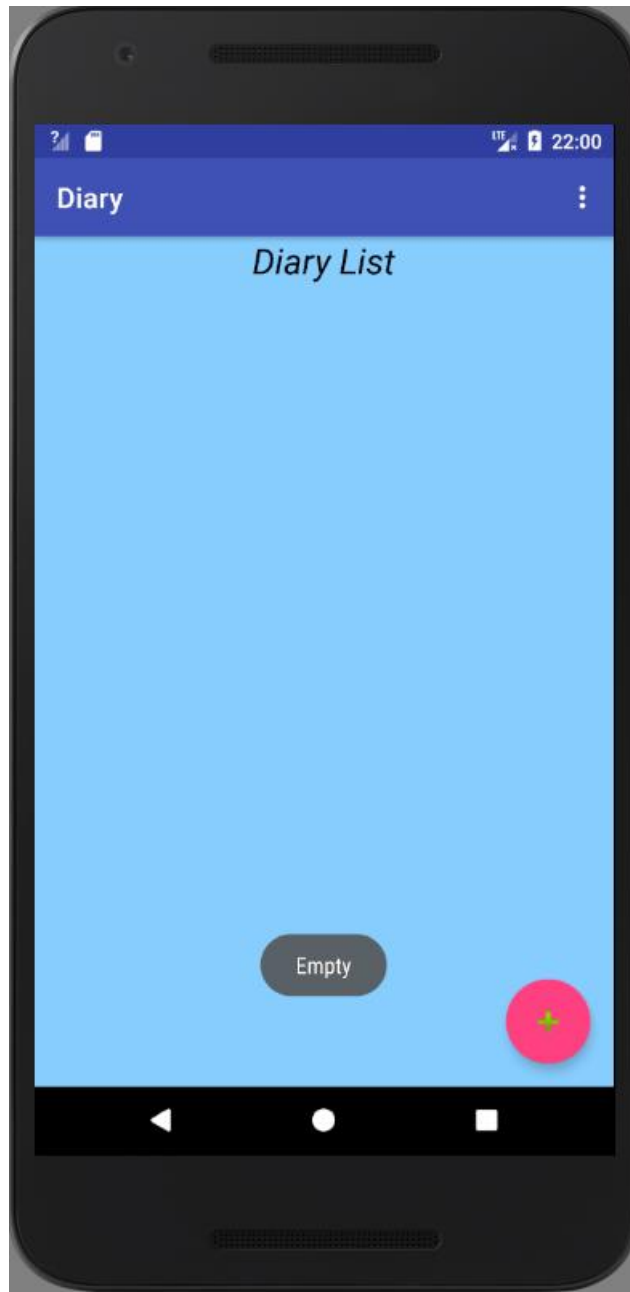
    </ScrollView>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fsb"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_marginTop="120dp"
        android:layout_gravity="center"

```

```
        android:clickable="true"
        app:fabSize="mini"
        app:srcCompat="@android:drawable/ic_menu_save" />
    </LinearLayout>
```

(4)当列表为空时，弹出 toast 提示列表为空。



//MainActivity.java 中对应代码及 listView 显示代码如下：

```
myhelper = new MyOpenHelper(getApplicationContext());

listView = (ListView) findViewById(R.id.lv_diary);
Cursor cursor = myhelper.query();
```



```

if(cursor.getCount() == 0){
    Toast.makeText(this, "Empty", Toast.LENGTH_SHORT).show();
}else{
    cursor.moveToFirst();
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("title", cursor.getString(1));
    map.put("time", cursor.getString(2));
    map.put("content", cursor.getString(3));
    listData.add(map);
}
while(cursor.moveToNext()){
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("title", cursor.getString(1));
    map.put("time", cursor.getString(2));
    map.put("content", cursor.getString(3));
    listData.add(map);
}
SimpleAdapter cursorAdapter = new SimpleAdapter(this, listData, R.layout.list_item, new
String[]{"title"}, new int[]{R.id.II});
listView.setAdapter(cursorAdapter);
cursor.close();

```

(5)点击菜单项 ADD 则跳转到新建日记界面，点击悬浮按钮同理。

//MainActivity.java 中代码如下：

```

if (id == 1) {
    Toast.makeText(MainActivity.this, "Click to add", Toast.LENGTH_SHORT).show();
    Intent intent = new Intent(MainActivity.this, AddActivity.class);
    startActivity(intent);
}

```

(6)点击 listView 里的条目则跳转到详细信息界面，数据通过 intent 传输到 SearchResultActivity。

//MainActivity.java 中代码如下：

```

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Map<String, Object> map = (HashMap<String, Object>)
parent.getItemAtPosition(position);
        Intent intent=new Intent(MainActivity.this, SearchResultActivity.class);
        intent.putExtra("title", (String) map.get("title"));
        intent.putExtra("time", (String) map.get("time"));
        intent.putExtra("content", (String) map.get("content"));
        startActivity(intent);
    }
}

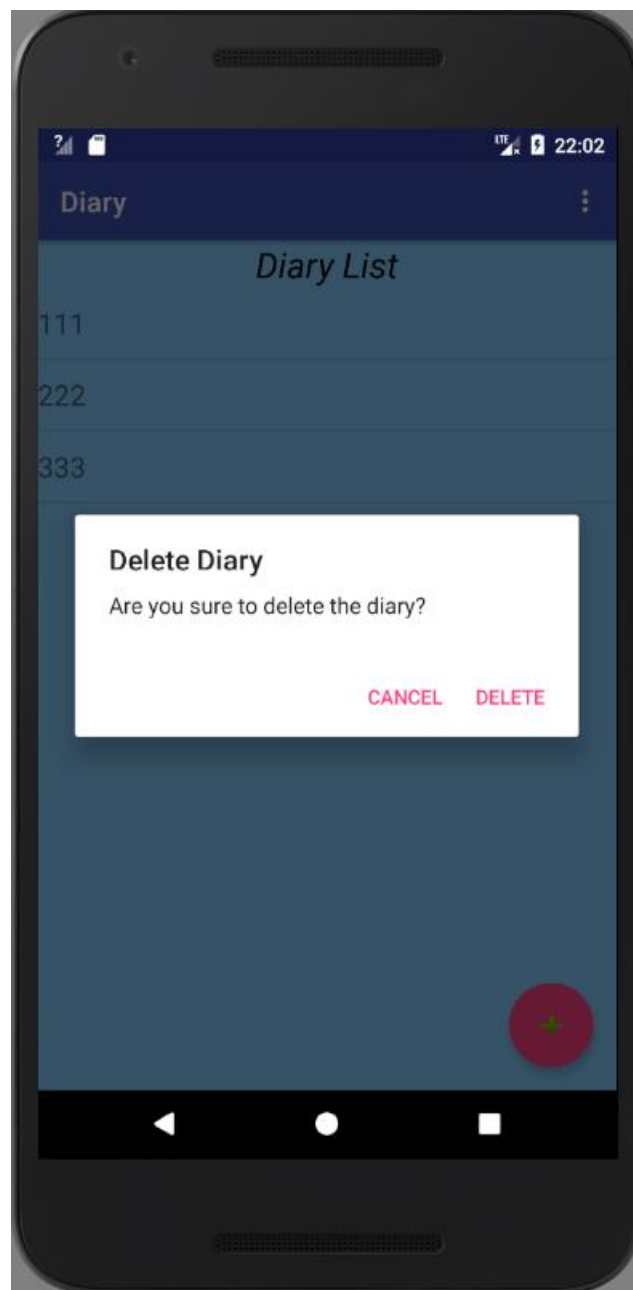
```

```

    }
});
//SearchResultActivity.java 内接受数据代码如下:
Intent intent = getIntent();
String title = intent.getStringExtra("title");
String time = intent.getStringExtra("time");
String content = intent.getStringExtra("content");

```

(7)长按 listView 里的条目则弹出对话框询问是否删除该日记记录。先通过 arg2 获取 listData 内所选条目的数据，再通过主键值来判断选中的 listView 中的 Item 所对应的数据库表中的_id，最后通过自定义的 deleteData 函数删除匹配的数据库里的记录，用 remove()函数删除 listView 里的条目后，刷新 listView。



//MainActivity.java 中代码如下:

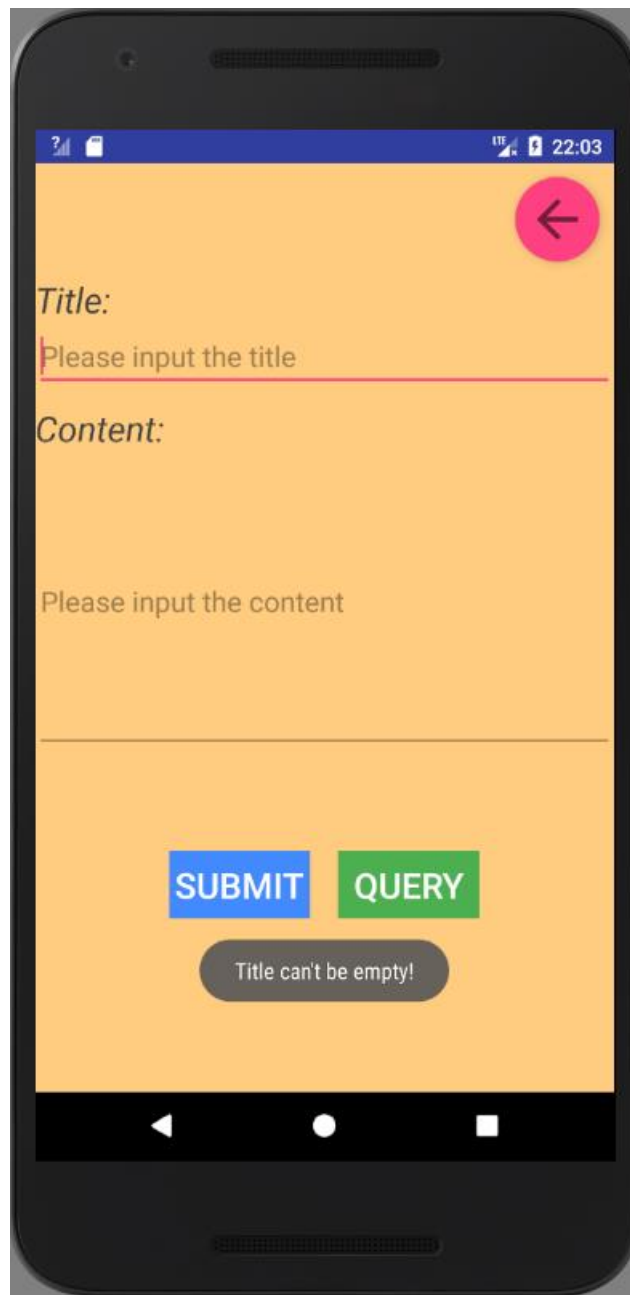
```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
  
    @Override  
    public boolean onItemClick(AdapterView<?> arg0, View arg1,  
                                int arg2, long arg3) {  
        // TODO Auto-generated method stub  
        final int pos = arg2;  
        Log.e("", "onItemClick");  
        AlertDialog.Builder builder = new AlertDialog.Builder(  
            MainActivity.this);  
        builder.setTitle("Delete Diary");  
        builder.setMessage("Are you sure to delete the diary?");  
        builder.setPositiveButton("Delete",  
            new DialogInterface.OnClickListener() {  
                public void onClick(DialogInterface dialog,  
                                    int which) {  
                    String deleteText = listData.get(pos).get("title").toString();  
                    SQLiteDatabase db = myhelper.getReadableDatabase();  
                    Cursor cursor = db  
                        .query("diary", null, null, null, null, null, null);  
                    while (cursor.moveToNext()) {  
                        String mtitle =  
cursor.getString(cursor.getColumnIndex("title"));  
  
                        if (deleteText.equals(mtitle)) {  
                            deleteData(cursor.getInt(0));  
                            break;  
                        }  
                    }  
                    listData.remove(pos);  
                    SimpleAdapter cursorAdapter = new  
SimpleAdapter(getApplicationContext(), listData, R.layout.list_item, new  
String[]{"title"}, new int[]{R.id.II});  
                    cursorAdapter.notifyDataSetChanged();  
                    listView.setAdapter(cursorAdapter);  
                    db.close();  
                }  
            });  
        builder.setNegativeButton("Cancel", null);  
        builder.create().show();  
        return true;  
    }  
});
```

```

public void deleteData(int index) {
    SQLiteDatabase db = myhelper.getReadableDatabase();
    // 从表中删除指定的一条数据
    db.execSQL("DELETE FROM " + "diary" + " WHERE _id="
        + Integer.toString(index));
    db.close();
}

```

(8)新建日记界面中，当 title 为空时，弹出 toast 提示框。



//AddActivity.java 中对应代码和存储代码如下：

```

if(title.equals(""))
    Toast.makeText(AddActivity.this, "Title can't be empty!", Toast.LENGTH_LONG).show();
else{
    ContentValues values = new ContentValues();
    values.put("title", title);
    values.put("time", time);
    values.put("content", content);
    MyOpenHelper helper = new MyOpenHelper(getApplicationContext());
    helper.insert(values);
    Intent intent = new Intent(AddActivity.this, MainActivity.class);
    startActivity(intent);
}

```

(9)新建日记界面中，还需获得存储数据时的时间，通过 SimpleDateFormat 设置时间格式并获取当前时间。

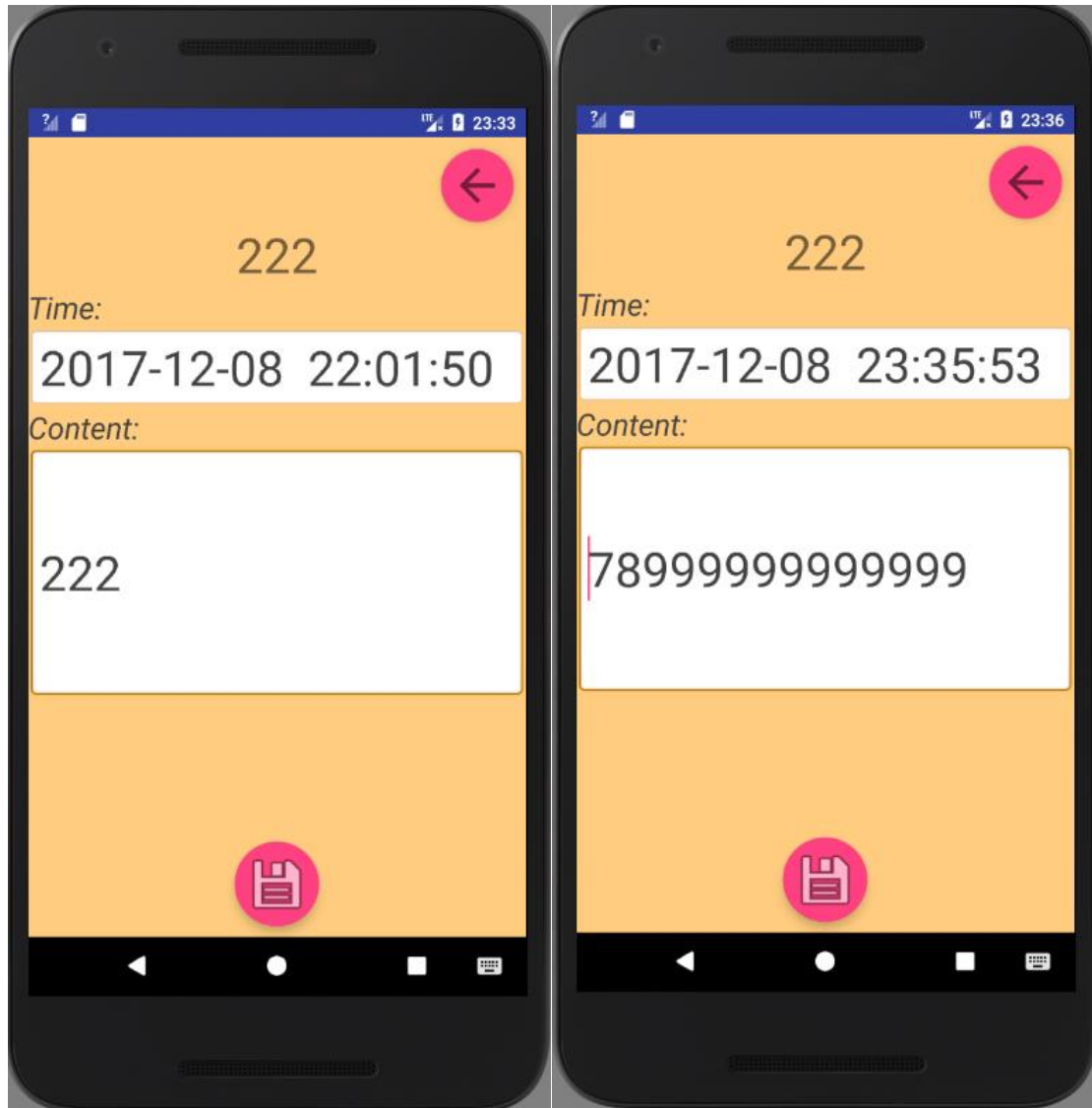
//AddActivity.java 中对应代码如下：

```

SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
Date curDate = new Date(System.currentTimeMillis());
String time = formatter.format(curDate);

```

(10)详细信息界面中，为实现编辑功能，当点击“保存”悬浮按钮时，获取 EditText 中的数据，再根据日记的 title 更新数据库内对应的数据。



//ResultActivity.java 中对应代码如下：

```
FloatingActionButton fsb = (FloatingActionButton) findViewById(R.id.fsb);
fsb.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String t = tv_title.getText().toString();
        String c = et_content.getText().toString();
        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        Date curDate = new Date(System.currentTimeMillis());
        String tim = formatter.format(curDate);
        MyOpenHelper helper = new MyOpenHelper(getApplicationContext());
        SQLiteDatabase db = helper.getWritableDatabase();
        db.execSQL("update diary set time=?,content=? where title=?",new
Object[]{tim,c,t});
        Toast.makeText(SearchResultActivity.this, "Save successfully",
Toast.LENGTH_SHORT).show();
    }
});
```

```

    }
});

```

(11)实现国际化，新建两个 strings.xml 分别实现中英两种语言的版本，改变手机语言设置为英语和中文显示相应的版本。

//zh-rCN\strings.xml 代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">日记</string>
    <string name="add_layout">新建日记</string>
    <string name="searchResult_layout">详细信息</string>
    <string name="action_settings">设置</string>
    <string name="action_add">添加</string>
    <string name="diary_list">日记列表</string>
    <string name="title">标题: </string>
    <string name="time">时间: </string>
    <string name="content">内容: </string>
    <string name="inputTitle">请输入标题</string>
    <string name="inputContent">请输入内容</string>
    <string name="submit">提交</string>
    <string name="query">查询</string>
</resources>

```

//en-rUS\strings.xml 代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Diary</string>
    <string name="add_layout">New Diary</string>
    <string name="searchResult_layout">Details</string>
    <string name="action_settings">Settings</string>
    <string name="action_add">Add</string>
    <string name="diary_list">Diary List</string>
    <string name="title">Title:</string>
    <string name="time">Time:</string>
    <string name="content">Content:</string>
    <string name="inputTitle">Please input the title</string>
    <string name="inputContent">Please input the content</string>
    <string name="submit">Submit</string>
    <string name="query">Query</string>
</resources>

```

8.软件操作流程:

用户使用 Diary 时，开始界面可显示日记列表，为空时弹出 toast 提示“Empty”，点击

特定条目可跳转到详细信息界面查看具体内容；通过长按条目可实现删除功能，弹出提示对话框；日记列表界面点击悬浮按钮和菜单项的 **ADD**，则跳转到新建日记界面；在新建日记界面，当标题框输入为空时弹出 **toast** 提示不能为空，点击 **SUBMIT** 按钮保存数据并跳转到日记列表界面查看插入结果，点击 **QUERY** 按钮直接跳转查询日记列表，点击返回悬浮按钮则返回主界面；详细信息界面，可编辑对应条目的内容，点击保存悬浮按钮自动更新时间并保存至数据库，点击返回悬浮按钮则返回主界面。

9.难点(或遇到的问题)和解决方案：

在实现 **Diary** 的过程中，有三个难点或问题：

- (1)获取时间数据时，**SimpleDateFormat** 报错，无法真机调试；
- (2)获取时间数据时，**Calendar** 函数获取的时间不正确；
- (3)最开始创建表时，只设置了 **title** 和 **content** 列，没有加上 **time**，后来加上之后，数据一直存不进表内；
- (4)设置 **listview** 条目的长按删除时，删除了所选条目的下一行。

解决方案：

- (1)获取时间数据时，**SimpleDateFormat** 报错，无法真机调试

SimpleDateFormat 要求手机 API 版本为 24 及以上，而 app 的 **gradle** 中代码如下所示：

```
defaultConfig {
    applicationId "com.example.echo.diary"
    minSdkVersion 21
    targetSdkVersion 25
    versionCode 1
    versionName "1.0"
    testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
}
```

可见，**minSdkVersion** 为 21，把代码改为如下所示（21 改为 24），并在 API 版本为 25 的虚拟机上运行，即可正常获取时间：

```
defaultConfig {
    applicationId "com.example.echo.diary"
    minSdkVersion 24
    targetSdkVersion 25
    versionCode 1
    versionName "1.0"
    testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
}
```

- (2)获取时间数据时，**Calendar** 函数获取的时间不正确

用 **Calendar** 类获取的时间不正确，月份数相比正确月份小 1，且获得的这些时间信息都是独立的，如果要一起显示的话，还要组装起来凑成一个字符串，稍显麻烦。不过如果只需要单个时间信息，如星期几，这种方法是比较方便的。并且可以根据 **Calendar.AM_PM** 属性判断当前是 AM 还是 PM（0 为 AM，1 为 PM），然后根据需要显示 12 小时或 24 小时的时间。或者直接获得 **Calendar.HOUR_OF_DAY** 显示 24 小时的时间。

与 **SimpleDateFormat** 类只需要三行代码相比较差，代码如下：


```
SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
Date curDate = new Date(System.currentTimeMillis());
String time = formatter.format(curDate);
```

(3)最开始创建表时，只设置了 title 和 content 列，没有加上 time，后来加上之后，数据一直存不进表内

MyOpenHelper 类内创建表时，加上 time 之后，对应的实体和类都做了相应更改，结果运行时报错如下：

[android.database.sqlite.SQLiteException: table diary has no column named content](#)

之后尝试更改表名也不能解决，思考之后觉得修改表后插入报错，可能 sqlite 没有更新该数据库下的表，只好修改数据库名称，新建一个，把数据库名从 Diary 改为 Diary2，果然没有问题了。代码如下：

```
private static final String DB_NAME="Diary2.db";
private static final String TBL_NAME="diary";
private static final String CREATE_TBL="create table diary(" +
    "_id integer primary key autoincrement, " +
    "title String, " +
    "time String, " +
    "content String);";
```

(4)设置 listView 条目的长按删除时，删除了所选条目的下一行

调试时，选择了第一行数据，但是删除了第二行数据，且刚删除后日记列表没有变化，从新建日记界面的 QUERY 按钮进入才更新。检查之后发现，在获取数据库数据之前对 listView 内 Item 进行了删除，把该行代码移到获取数据之后就解决了删除的与所选的 Item 不一致的问题，代码如下：

```
Cursor cursor = db
    .query("diary", null, null, null, null, null, null);
while (cursor.moveToNext()) {
    String mtitle = cursor.getString(cursor.getColumnIndex("title"));
    if (deleteText.equals(mtitle)) {
        deleteData(cursor.getInt(0));
        break;
    }
}
listData.remove(pos);
```

对于 listView 没变化的问题，刷新 listView 即可。代码如下：

```
SimpleAdapter cursorAdapter = new SimpleAdapter(getApplicationContext(), listData,
    R.layout.list_item, new String[]{"title"}, new int[]{R.id.I1});
cursorAdapter.notifyDataSetChanged();
listView.setAdapter(cursorAdapter);
```

10.不足之处和今后的设想：

本次用 **Android Studio** 开发的计算器 **Diary** 界面虽然还不够精美，但实现起来比较简洁，且使用了悬浮按钮和菜单栏，有相应的消息提示和对话框，用户交互性较好。没有实现播放音乐的功能。

希望以后可以使用酷炫的动效和素材实现更美观的界面，增加播放音乐等功能等。