北京理工大学珠海学院

Android 移动开发技术说明书

_2019_—_2020_学年第_2_学期

作品名称: 时序

学　　院: 计算机学院

专业班级: 18 软件工程 3 班

学　　号: 180021105120

学生姓名: 刘堉嘉

指导教师: 许婷

成　　绩:

时　　间: 2020-05-30

2020 年　06 月 01 日

1. 作品介绍

1.1 设计背景

生活中经常会有一些容易忘的事情，特别是网课，课程作业平台多，作业也随着增加，事情管理难度大，在家微小的放松，总会让人忘记一些事情，为了解决这样的情况，做了时序 APP，灵感也来自一个我经常用的《时间序》APP。

1.2 设计方案

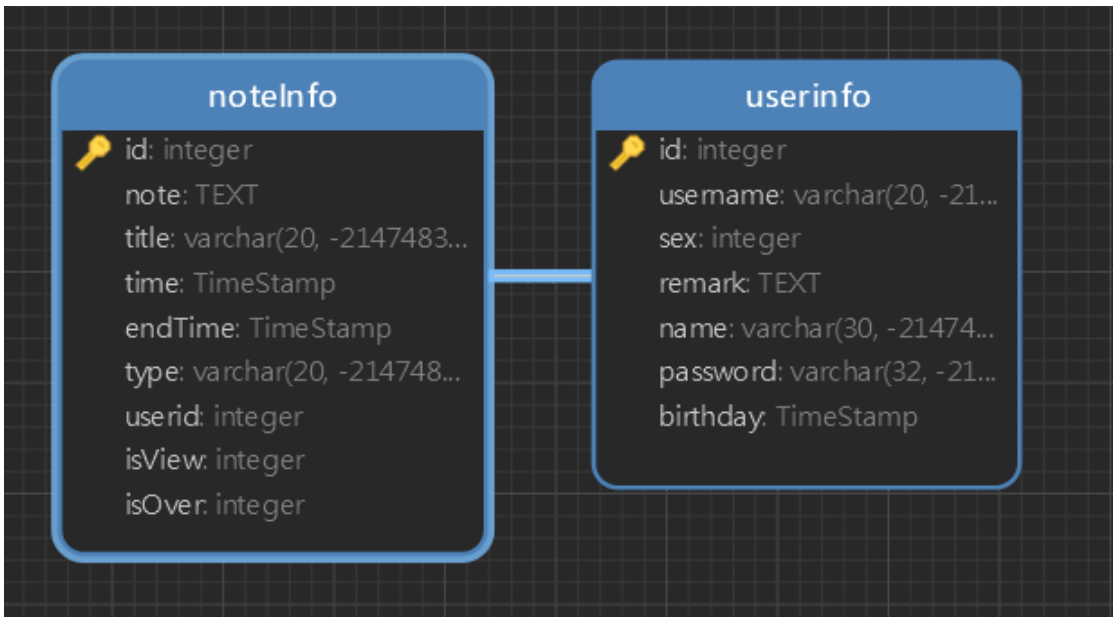


图 1 数据库设计图

用户表		笔记表	
id	唯一标识	id	唯一标识
username	用户名唯一	note	笔记
password	密码	title	标题
sex	性别	endTime	到期时间
remark	个性签名	type	类型
name	称谓	userid	标识用户
brithday	生日	isView	是否展示
		isOver	是否结束

图 2 表字段意义

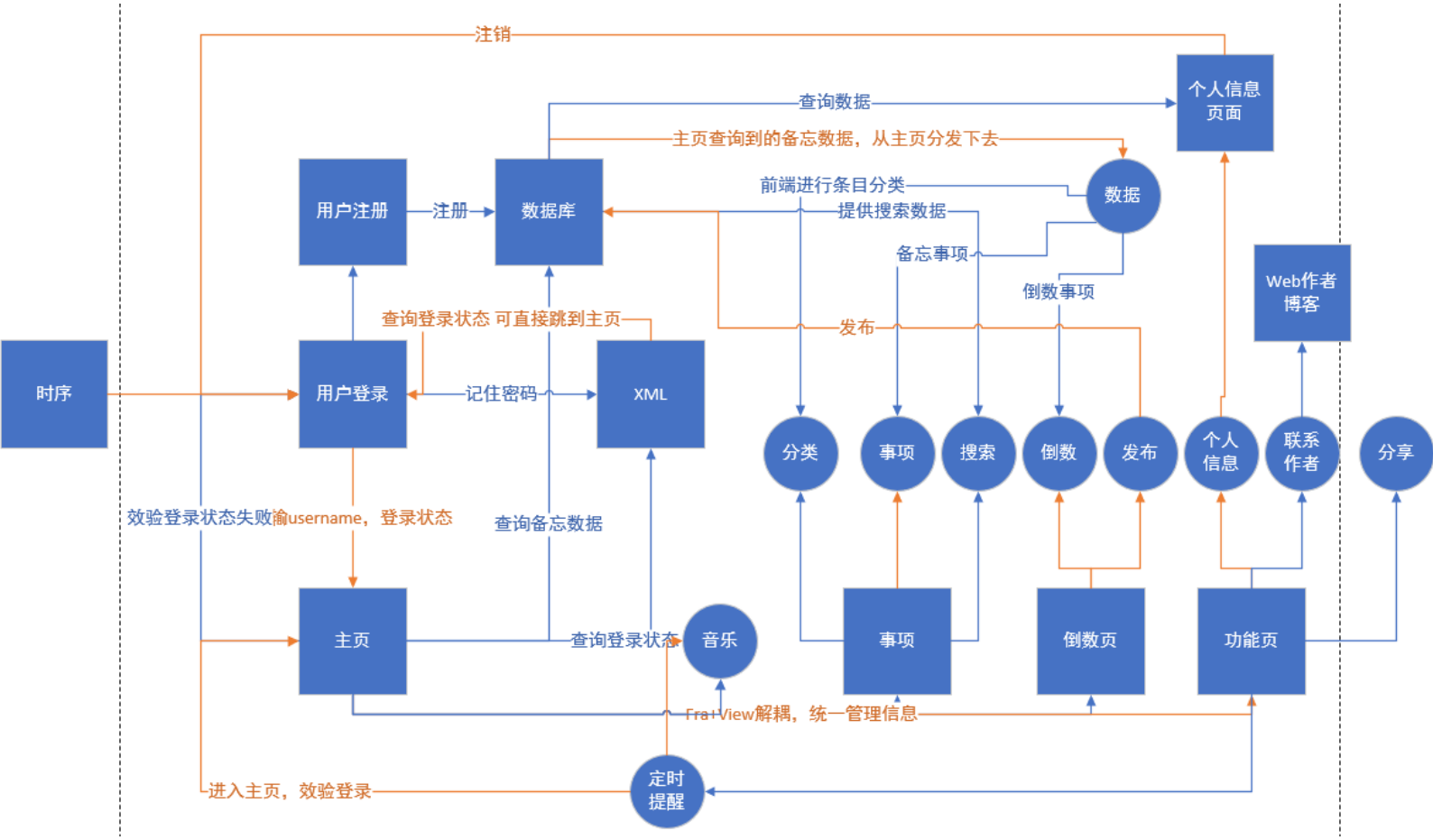


图 3 功能流程设计图

1. 数据库设计

因为是类似备忘录的小程序，数据库的建立比较简单如**图一、图二**所示。大概上分布了我所需模块功能上的选择，在个人信息上做多了一些预留处理，因为精力有限，尚未全部完成。可以待后面功能补全利用，如生日在他设定好之后其其添加到倒数计时模块，姓名也可以多样化使用等。

2. 模块设计

总体上 UI 设计简洁，体验上尽量设计增强用户体验，进行多次功能测试，BUG 上很少，发现的都已修正。

2.1 登录页

图 3 登陆

图 4 注册

图 5 个人信息

用户模块：

基础模块功能，用户注册、登陆和注销

在登陆上采用较为灵活的方式，注册为用户提供良好的信息答复体验，用户名密码长度效验。**注册后**自动跳转登陆页，**自动填写账号密码**。

登陆时，还可以选择**记住密码**进行下次**自动填写**。如果懒于注册，可以选择**离线模式**进行

默认账号：admin　密码： admin（只适用于离线模式，在线模式不能使用）

登陆后，下次进入 APP 会**自动登陆跳转到首页**，在个人页面注销后进行重新登陆。省去频繁的登陆步骤，极好优化体验。

密码进行 MD5 加密效验，安全性高。

注册成功后帮其插入欢迎来到时序，产生第一条事项，使首页不太空阔。

2.2 首页：

在设计上采用 tabor 栏目的模式，顶部音乐模块，然后可以点击顶部跳转也可以滑动屏幕切换，体验很好。

图 6 事项页

图 7 近期页

图 8 我的页

事项浏览： 图 6 所示，采取左边分类筛选，顶部搜索对事项进行筛选展示，重点让事项的跟容易看错总的内容，有星期几和标题等。左边的圆圈点击为完成事项。会给予去除。

每日计时：根据日期倒数的模式，清晰直观。并且在右下角设置添加文章功能。

每日计时和事项页都可以点击后查看文章详细，修改，如图 9 图 10 所示。

图 9.1 浏览修改

图 9.2 浏览修改

图 10 发布页面

图 5 个人信息

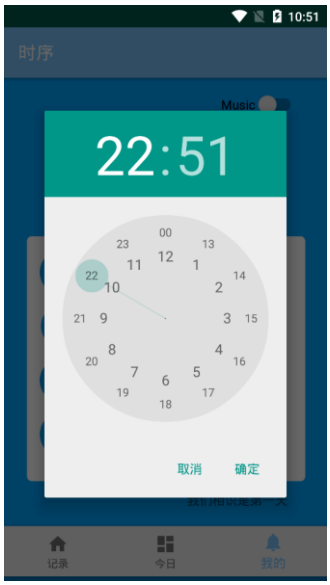


图 11 定时提醒

图 12 博客页

我的页面：

个人信息：通过更改性别可以切换男女头像，和其他修改信息和注销。

定时提醒： 定时提醒，设定时间后倒计时，会唤醒 APP，然后播放音乐，可在首页关闭

作者博客：可以联系作者，查看版本更新，进行软件下载，同时在分享功能，也主要是分享博客地址，进行页面下载。

3. 外观设计

logo 和 icon 由网上素材用 PS 制作。UI 设计参考 时间序。

2 技术方案

2.1 登录功能*（顶格、四号黑体）*

登录功能：

登　　录：采用 Sqlit 数据库查询效验账号密码

```
public class UserSqlInfo extends SQLite {
    //查询用户
    public static Map<String,String> queryUser(Context context){
        SqlHelper help = querySql(context, tableName: "userinfo",new String[]{"id","username","password"}, calName: null, values: null);
        Cursor cursor = help.cursor;
        Map<String,String> mp = new HashMap<String,String>();
        while(cursor.moveToNext()){
            mp.put(cursor.getString(cursor.getColumnIndex( s: "username") ),cursor.getString(cursor.getColumnIndex( s: "password") ));
        }
        help.db.close();
        return mp;
    }
```

图 13 数据库查询

记住密码：采用 SharedPreferences 进行本地保存。

```
public static boolean saveUserInfo(Context context, String username, String password) {
    SharedPreferences spf = context.getSharedPreferences( s: "userdata", Context.MODE_PRIVATE);
    SharedPreferences.Editor edit = spf.edit();
    edit.putString( s: "username", username);
    edit.putString( s: "password", password);
    edit.commit();
   return true;
}
public static HashMap<String,String> readUserInfo(Context context) {
    SharedPreferences spf = context.getSharedPreferences( s: "userdata", Context.MODE_PRIVATE);
    String username = spf.getString( s: "username", s1: null);
    String password = spf.getString( s: "password", s1: null);
    HashMap<String,String> h = new HashMap<>();
    h.put("username", username);
    h.put("password", password);
    return h;
}
```

图 14 XML 处理

自动登录：采用 SharedPreferences 保存登录状态，判定登录。

离线 在线： 测试账号采用在创建数据库的时候进行插入 admin 用户，由于密码进行 MD5 加密的密码，在这里我采用直接 admm 进行模拟密码录入，所以用户输入基本不可能在线登录，除非对 admin 进行逆向 md5 加密。

```
sqLiteDatabase.execSQL("CREATE TABLE noteInfo(" +
        "id integer primary key autoincrement ," +
        "note   TEXT DEFAULT \"\" ," +
        "title   varchar(20) NOT NULL," +
        "time TimeStamp NOT NULL DEFAULT (strftime('%s','now')*1000)," +
        "endTime TimeStamp DEFAULT  (strftime('%s','now')*1000) ," +
        "type varchar(20) DEFAULT \"日常\" ," +
        "userid integer NOT NULL ," +
        "isView integer DEFAULT 1 ," +
        "isOver integer DEFAULT 0 ," +
        "FOREIGN KEY(userid) REFERENCES userinfo(id)" +
        ");");
        sqLiteDatabase.execSQL("insert into userinfo(id,username,password) values(1,\"admin\",'admin')");
        sqLiteDatabase.execSQL("insert into noteInfo(id,userid,title,isView) values(1,1,'欢迎来到时序！',1)");
```

图 15 数据库创建初始化

2.2 注册功能

注册：简单 Sql 注册，简单有效性效验。对密码 MD5 加密存储。

```
//插入用户
public static long insertSql(Context context ,String tableName,String[] calName,String []values){
    long flag;
    SqlHelper help= setDataSql(context,calName,values);
    flag = help.db.insert(tableName, nullColumnHack: null,help.values);
    help.db.close();
    return flag;
}
```

图 16 插入

```
if( UserSqlInfo.insertUser( context: RegisterActivity.this,us,pw) ){
    //saveInfoXml.saveUserInfo(RegisterActivity.this,  us,pw);
    intent.putExtra( name: "username",us);
    intent.putExtra( name: "password",pw);
    System.out.println("注册成功");
    return 0;
}
```

```
register.setOnClickListener((view) -> {
        int stc = Register();
        if(stc==0){
            Toast.makeText( context: RegisterActivity.this, text: "注册成功！",Toast.LENGTH_SHORT).show();
            setResult( resultCode: 1,intent);
            RegisterActivity.this.finish();
        }else if(stc==-1){
            Toast.makeText( context: RegisterActivity.this, text: "再检查一下哦！",Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText( context: RegisterActivity.this, text: "已存在用户了！",Toast.LENGTH_SHORT).show();
        }
} );
```

图 17 数据效验存储

自动填写账号密码：采用请求回调

```
protected void onActivityResult(int requestCode ,int resultCoide,Intent data){

    if( requestCode==0 && resultCoide==1){
        username.setText(data.getStringExtra( name: "username"));
        password.setText(data.getStringExtra( name: "password"));
    }
}
```

图 18 处理回调

2.3 首页切换功能

首页 navigation 和分页

技术采用 BottomNavigationView+ ViewPager+ Fragment 的方式，

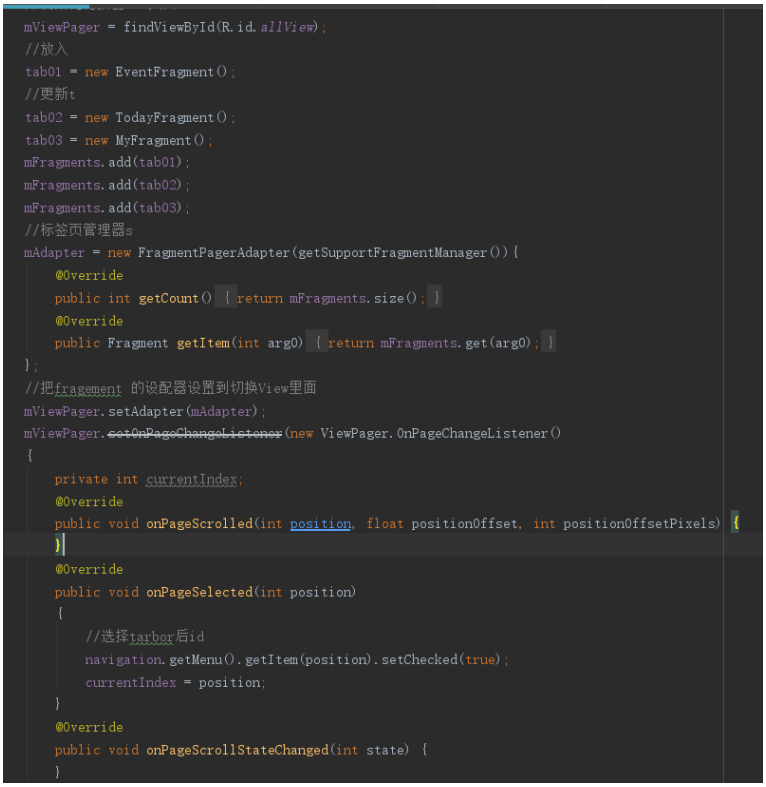
BottomNavigationView 底部栏点击动画和切换事件分发

```
/*
 * ViewPager
 */
private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener
        = (item) -> {
        switch (item.getItemId()) {
            case R.id.navigation_home:
                mViewPager.setCurrentItem(0);
                return true;
            case R.id.navigation_dashboard:
                mViewPager.setCurrentItem(1);
                return true;
            case R.id.navigation_notifications:
                mViewPager.setCurrentItem(2);
                return true;
        }
        return false;
};
```

图 19 页面切换

Fragment 对页面进行首页解耦，更灵活去设计功能。

Fragment+ViewPager 配合页面进行切换



```
mViewPager = findViewById(R.id.allView);
//放入
tab01 = new EventFragment();
//更新t
tab02 = new TodayFragment();
tab03 = new MyFragment();
mFragments.add(tab01);
mFragments.add(tab02);
mFragments.add(tab03);
//标签页管理器s
mAdapter = new FragmentPagerAdapter(getSupportFragmentManager()){
    @Override
    public int getCount() { return mFragments.size(); }
    @Override
    public Fragment getItem(int arg0) { return mFragments.get(arg0); }
};
//把fragement 的设配器设置到切换View里面
mViewPager.setAdapter(mAdapter);
mViewPager.setOnPageChangeListener(new ViewPager.OnPageChangeListener()
{
    private int currentIndex;
    @Override
    public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {
    }
    @Override
    public void onPageSelected(int position)
    {
        //选择tarbor后id
        navigation.getMenu().getItem(position).setChecked(true);
        currentIndex = position;
    }
    @Override
    public void onPageScrollStateChanged(int state) {
    }
```

图 20 滑动

音乐播放采用常规的 server 进行后台播放，开关监听开关。

2.4 事项页和倒计时数据共享功能

事项页和倒计时页采用 RecyclerView 进行列表展示，因为涉及搜索、分类、同步修改、实时更新在设计上略微麻烦。

初始化：

因为是数据基本相同，并且不想代码显得太多乱，所以没有写 2 个 Adapter 或者 2 个 Holder 而是采用 id 匹配选择性索引和布局的方式。

```
ViewHolder(View itemView) {
    super(itemView);
    //找到对应元素
    switch(rcvID){
        case R.id.mainRecy:
            image = itemView.findViewById(R.id.imageViewIcon);
            day = itemView.findViewById(R.id.textDay);
            break;
        case R.id.rightRecy:
            allView = itemView.findViewById(R.id.allView);
            radio = itemView.findViewById(R.id.radioButton2);
            break;
    }
```

```
if( rcvID ==R.id.mainRecy ){
    view = LayoutInflater.from(parent.getContext()).inflate(R.layout.main_page_item_right,parent, attachToRoot: false);
    //实例化得到Item布局文件的View对象
    //View v = View.inflate(context, R.layout.,null);
}else{
    view = LayoutInflater.from(parent.getContext()).inflate(R.layout.main_page_item_left,parent, attachToRoot: false);
}
```

图 21 选择合适的

在数据获取上：

数据是保存在主页的 activity 上面的，分发给各个 Fragment（实现了自己写的简单 FragmentData 接口），然后 Fragment 进行主动获取数据，而数据赛选排序和是否展示等处理在各个页面进行变换，从而实现数据共享功能。然后主页的 activity 也进行通过更新数据，下发到 Fragment，进行 notifyDataSetChanged 数据的更新。

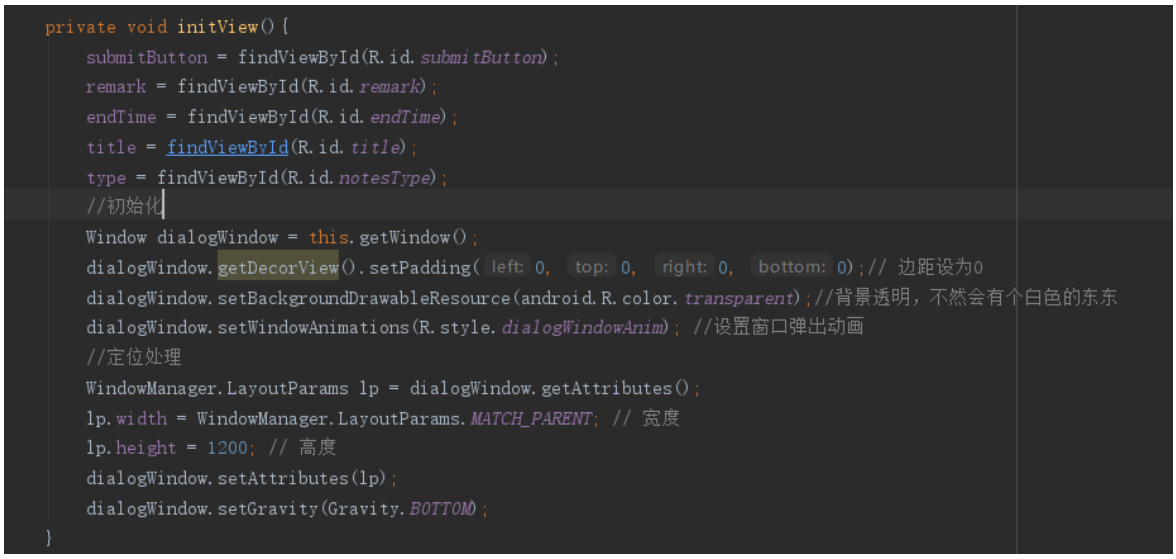
而期间遇到数据库查询的异步问题，可是对 android 异步处理不懂时间缘故采用了严格轮转解决。排序问题在前端解决。

```
public void updata() {
    //不安全
    while(MainPageActivity.getList()==null){
    }
    DataList.clear();
    List <DayInfo> l= MainPageActivity.getList();
    for(int i =0;i<l.size();i++){
        if(!l.get(i).isOver){
            DataList.add(l.get(i));
        }
    }
    rcv.getAdapter().notifyDataSetChanged();
}
```

图 22 数据更新

2.4 修改预览文章功能

通过重写 Dialog，进行出入动画修改等设置。修改具体数据，在数据的的定位修改，采用在数据位置再次重写 Dialog 的 cance 方法。



```
<!--show-->
<translate
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="300"
    android:fromXDelta="0"
    android:fromYDelta="100%"
    android:toXDelta="0"
    android:toYDelta="0">
</translate>
```

```
//自定义 窗口
final Dialog dialog = new ItemDialog(activity,R.style.Theme_AppCompat_Dialog,data.get(position)){
    @Override
    public void cancel() {
        super.cancel();
        //改变了
        if( isChange ){
            List<DayInfo> v =  MainPageActivity.getList();
            //InfoAdapter.this.notifyItemChanged(position);
            for( int i =0;i<v.size() ;i++){
                if(data.get(position).id==v.get(i).id){
                    v.set(i,data.get(position));
                }
            }
        }
        activity.updateFra();
    }
} ;
```

图 23 重写 Dialog

2.5 搜索功能和分类

分类采用写 RecyclerView，然后在，点击分类后触发后，采用前端筛选的方法进行过滤。然后搜索功能，通过查询数据库更新首页数据，下发更新。

```
holder.typeText.setOnClickListener((view) -> {
            List<DayInfo> soucs = MainPageActivity.getList();
            Vector<DayInfo> datas = new Vector<>() ;
            //Log.d("len:",""+datas.size());
            for( DayInfo d : soucs){
                //Log.d("data:",s+d.type);
                if( (s.equals(d.type) || "全部".equals(s))  && !d.isOver){
                    datas.add(d);
                }
            }
            right.getData().clear();
            right.getData().addAll(datas);
            right.notifyDataSetChanged();
    });
}
```

图 24 搜索查询监听

2.5 发布功能

简单插入数据库，通过回调处理数据更新，添加新发布的条目。

2.6 个人信息、注销、切换头像

简单逻辑切换，更新数据库操作，注销采用更新登录状态后，进行销毁其他页面进入登录。

2.7 安利一下和博客功能

安利采用 share_intent.setAction(Intent.ACTION_SEND); 进行分享，博客采用 WebView 组件然后进行 https 检查。

2.8 定时提醒

定时提醒采用 AlarmManager 定时提醒，回调到登陆页面，自动登录。播放音乐。

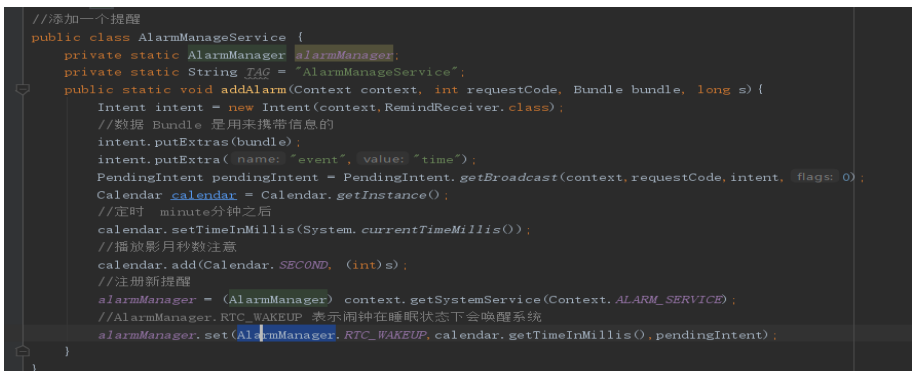


图 25 定时唤醒

3.总结说明

初学 Android 查找了许多资料，额外学到了很多东西，锻炼了自己的自学能力，也了解了一些，特别是在处理，同步 2 个列表，数据更新上，想了很多种方案，和方法去更好，优雅地改善一个功能方法的实现途径。当然还有意义很大的就是，对我 java 功底的一些锻炼，java 用得更熟练了。因为在写数据库的时候，也写了挺多工具类，包括数据库预处理数组封装，MD5 加密，时间处理的一些，对知识有了一定的积累。体会到了 UI 的重要性，UI 改了好几次，直到最后一版本看起来，感觉好像 APP 好了一个档次的样子。还有就是对 Android 的理解更加深刻了，从原理到它的机制等等。