

Android 移动应用开发期末项目报告

OneDay

组号：14 小组成

员：

张凯（组长） 杨金辉 杨志华 杨立坤

目录

1 概述	3
2 背景	3
3 项目说明	4
3.1 功能模块	4
3.2 逻辑流程	5
3.3 技术说明	5
3.4 代码架构	9
4 应用效果	9
5 开发过程中遇到的问题及解决办法	10
6 思考及感想	10
7 小组分工	11
8 参考资料	11

1 概述

本小组的期末项目的基本设计是实现一个简单而实用的日记软件，名为 OneDay。应用功能主要有对日记的增删改查，照片墙以及日历等。

2 背景

如今，我们生活节奏紧凑，想要回顾过去的记忆，却无暇坐下写篇日记；想记录当天内心想法隐私，却不想发布到朋友圈微博。这时，我们需要一个电子日记 APP。一小段文字，一张简单图片构成当天的日记，记录下每一天走过的足迹。

但反观当今的日记软件，有的简单到连密码保护的功能都没有以致于无法很好地保护隐私，有的复杂到可以把日记发到交友圈里以致于偏离日记太远而成为社交软件。我们小组设计日记的初衷比较贴近传统日记的特点：轻量级，实用性。

轻量级：OneDay 只允许用户在每一天的日记里添加一张照片，一段文字以及心情和天气。为什么只有一张照片呢？因为我们相信人在资源缺乏的时候会把资源用在最有用的地

方，所以我们相信人们会在精彩的一天结束后会把最有意义，最有价值，最能唤起当天回忆的一张照片保存下来。而不是像懒得去翻图库里成百上千的图片那样，OneDay 更能阻止生活碎片化。

实用性：OneDay 在提供了对日记的增删改查的基本功能和时间轴的查看日记的方式以外，还实现了从日历查看某天的日记以及照片墙方便用户浏览照片的功能，这样一来用户就很容易方便地操作和浏览日记。

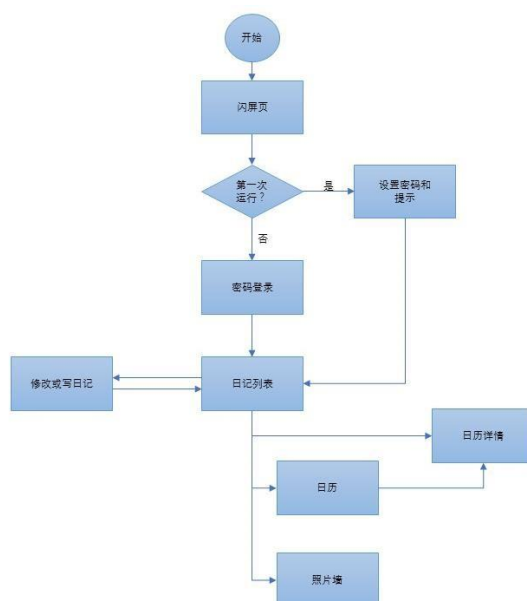
稳定性：照片墙是采用了 Android 提供的 LruCache 类来管理内存，该类采用最近最少使用算法，在图片内存总和大小达到阈值会自动释放图片的内存，解决了因为加载过多图片导致内存不足程序崩溃的问题。

3 项目说明

3.1 功能模块



3.2 逻辑流程



3.3 技术说明

【日历】

这里我使用了开源的一个日历控件——MaterialCalendarView

1. 首先是添加依赖：compile 'com.prolificinteractive:material-calendarview:1.4.0'
2. 然后将日历控件添加到布局文件中，就在相应的 Activity 中对其属性进行编辑。

```

<com.prolificinteractive.materialcalendarview.MaterialCalendarView
    app:layout_constraintTop_toBottomOf="@+id/Calendar_title"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:paddingLeft="5dp"
    android:paddingRight="5dp"
    android:id="@+id/calendar"
    android:layout_width="368dp"
    android:layout_height="wrap_content"
    app:mcv_showOtherDates="none"
    app:mcv_selectionColor="@color/colorPrimary"
    app:mcv_arrowColor="@color/colorPrimary">
  
```

3. 我首先用到的是设置日期的范围，使用了函数 `setMinimumDate()` 设置了我们的日记的最小时间范围，使用 `setFirstDayOfWeek()` 函数设置我们日历中一周开始的星期，以及最后日历显示的模式是 `setCalendarDisplayMode()` 是按月显示。

```

mcv.state().edit()
    .setFirstDayOfWeek(java.util.Calendar.MONDAY)
    .setMinimumDate(CalendarDay.from(2016, 4, 3))
    .setCalendarDisplayMode(CalendarMode.MONTHS)
    .commit();

```

4. 然后就是按要求，对我们已经有日记的天数进行特定的修饰，使得那些天与其他日子不一样，这里使用 `DayViewDecorator`。

```

public class TodayDecorator implements DayViewDecorator {

    private final ArrayList<CalendarDay> specaildate;
    private ForegroundColorSpan span; //文本颜色
    private Context context;
    public TodayDecorator(Context mContext, ArrayList<CalendarDay> dates) {
        specaildate = dates;
        span = new ForegroundColorSpan(mContext.getResources().getColor(R.color.colorWhite));
        context = mContext;
    }

    @Override
    public boolean shouldDecorate(CalendarDay day) {
        for (int i = 0; i < specaildate.size(); i++) {
            if (specaildate.get(i).equals(day)) return true;
        }
        return false;
    }

    @Override
    public void decorate(DayViewFacade view) {
        Drawable highlightDrawable = new ColorDrawable(Color.parseColor("#228BC34A"));
        Drawable h = context.getResources().getDrawable(R.drawable.background);
        view.setBackgroundDrawable(h);
        view.addSpan(span);
        /*view.addSpan(new StyleSpan(Typeface.BOLD));
        view.addSpan(new RelativeSizeSpan(1.6f));

        view.addSpan(new DotSpan2(8, span.getForegroundColor()));*/
    }
}

```

其中 `addSpan()` 是对字体进行修改，而 `setBackgroundDrawable()` 是对整个日期框就行修饰。最后就在 `Activity` 中把这个 `Decorator` 运用到我们的 `MaterialCalendarView` 即可：

```

mcv.addDecorators(new TodayDecorator(this, dates), new TodayDecorator2(this, CalendarDay.today()));

```

【照片墙】采用 `GridView` 来存放图片，由于照片墙的图片太多会导致内存不够程序崩溃，所以我们采用了 `LruCache` 类来限制加载的图片的总内存大小。

LRU 算法：LRU 是近期最少使用的算法，它的核心思想是当缓存满时，会优先淘汰那些近期最少使用的缓存对象。

LruCache 类介绍：泛型类，主要算法原理是把最近使用的对象用强引用（即我们平常使用的对象引用方式）存储在 LinkedHashMap 中。当缓存满时，把最近最少使用的对象从内存中移除，并提供了 get 和 put 方法来完成缓存的获取和添加操作。

【emoji】

采用 SpannableString 把 String 转换成预定义好的图片，实现心情和天气的图片显示。

【主界面】

1. 闪屏页面：

(1) setDuration 来设置闪屏页的停留时间

```
aa.setDuration(3000);
```

setAnimationListener 方法来实现闪屏页的渐变效果

```
aa.setAnimationListener(new Animation.AnimationListener()
{
    @Override
    public void onAnimationEnd(Animation arg0) {
        redirectTo();
    }
    @Override
    public void onAnimationRepeat(Animation animation) {}
    @Override
    public void onAnimationStart(Animation animation) {}
});
private void redirectTo(){
    Intent intent = new Intent(this, MainActivity.class);
    startActivity(intent);
    finish();
}
```

2. 登录界面：用 SharedPreferences 来保存密码和密码提示，初次登录时密码为空，进

入密码设置界面，否则进入登录界面

```

initial(),
SharedPreferences sharedPref = this.getSharedPreferences("MY_PREFERENCE", Context.MODE_PRIVATE);
final SharedPreferences.Editor editor = sharedPref.edit();
password = sharedPref.getString("password", "");
hintText = sharedPref.getString("hint", "");
if (password.equals("")) {

```

每一个数字为一个 button，每 click 一次 button 进行判断，如果还没输够 4 个数字就可继续输入，且设置对应位置的密码图片为填充，否则直接判断字符串是否对应密码

```

public void NumButClick(String i) {
    if (flag1.equals("")) {
        flag1 = i;
        code1.setBackgroundResource(R.drawable.but_back);
    } else if (flag2.equals("")) {
        flag2 = i;
        code2.setBackgroundResource(R.drawable.but_back);
    } else if (flag3.equals("")) {
        flag3 = i;
        code3.setBackgroundResource(R.drawable.but_back);
    } else if (flag4.equals("")) {
        flag4 = i;
        code4.setBackgroundResource(R.drawable.but_back);
    }
    if (!flag4.equals("")) {
        if (password.equals(flag1+flag2+flag3+flag4)) {
            Intent intent = new Intent(MainActivity.this, DiaryList.class);
            startActivity(intent);
        } else {
            textlog.setText("");
            textlog.setText("密码错误");
            initial();
        }
    }
}
}

```

3. 修改密码界面：

判断如果成功修改密码，则将原有生成文件删除，再生成新的 file

```

final File file= new File("/data/data/"+getPackageName().toString()+"/shared_prefs","Activity.xml");

if (file.exists()) {
    file.delete();
}

```

【写日记】

1.从相册选择照片。这里是使用了 Intent.ACTION_PICK 跳转相册 activity，然后在 onActivityResult 处理返回的照片数据，用 BitmapFactory 对返回的数据流解码传到一个调整照片宽高的函数，返回想要的宽高的 bitmap 图片。

3.按下确认键后。使用 SQLite 数据库把相关信息保存起来。



5 开发过程中遇到的问题及解决办法

问题 1：整合代码时显示 R 无法解决。

解决方法：在 manifest 添加 activity，例如

```
<activity android:name=".DiaryWrite" />
```

问题 2：在引用图片的时候闪屏。

解决方法：降低图片分辨率，降低图片大小。

6 思考及感想

这次项目让我们学到了团队合作的重要性，不像普通的作业一个人就可以完成，这次的期末项目明显就比较复杂。涉及知识点比较多，也有很多知识是在课堂上没学到的，所以我們也是边做边学习。当然，因为关于 Android 的技术博客和资料比较多，我们学起来也不难，主要是用心。

这次期末项目学到的新的知识主要有如何使用 GridView 来构造照片墙，如何使用 LruCache 来控制内存大小，如何使用 android 提供的日历接口，如何使用 ViewPager 来显示图片，如何用 SpannableString 显示 emoji 图片等。

最后感谢老师的教育和 TA 们的付出。

7 小组分工

组员	学号	负责内容	贡献百分比 (%)
杨立坤	15331361	照片墙和日记列表	25
杨金辉	15331357	主界面和展示	25
杨志华	15331370	日历和 ppt	25
张凯	15331393	写日记和日记详情 界面	25

8 参考资料

无