



华南师范大学

本科学生实验（实践）报告

院 系：计 算 机 学 院

实验课程：移动应用开发

实验项目：我的日记本

软件名称：每日心情 (DailyMood)

指导老师：曹阳

开课时间：2018 ~ 2019 年度第 1 学期

专 业：计算机科学与技术

班 级：2016 级 4 班

学 生：余梓权

学 号：20162180149

完成时间：2018 年 10 月 24 日

华南师范大学教务处

责任声明：感谢开源库 SwipeDelMenuLayout、floatingactionbutton、recyclerview、glide、eventbus、butterknife 的大神的功能支持，得以顺利完成该项目。

权利声明：该项目的所有内容，包括文档、源代码、资源文件，曹阳老师和本人都拥有对其进行修改、复制、引用、阅读的权利。

版本说明：截止该文档完成时间，该项目 App 为 V 1.2.0。

目录

一、软件内容简介	4
二、界面设计.....	4
(一) 主界面 (心情卡片列表界面) 以及一个国际化示例.....	4
(二) 心情卡片新建界面 (初始日期为当日, 用户可以保持默认不必输入)	7
(三) 心情卡片编辑界面	8
(四) 心情卡片详情界面	9
(五) 设置界面	10
(六) 关于界面	11
三、代码设计.....	12
(一) 项目分包, 模块解耦	12
(二) 系统的字符串常量的统一安排 (统一放置在 string.xml 文件中)	12
1、中文	13
2、英文	14
(三) CreateMoodActivity (同时充当心情卡片创建界面和编辑界面)	14
(四) APP 使用状态的存储和读取.....	16
(五) 背景音乐的简易控制 (BGMService.java)	17
(六) SQLite 数据库的应用	18
四、软件操作流程	18
五、难点和解决方案.....	19
(一) 获取图片的真实路径 (绝对路径)	19
(二) 提示语的显示和隐藏 (Activity 间的简易通讯实现方式)	19
(三) 优化卡片列表的加载, 使之滑动顺畅	19
(四) 日期重叠的处理.....	20
六、不足之处和今后的设想	20
七、参考文献.....	20

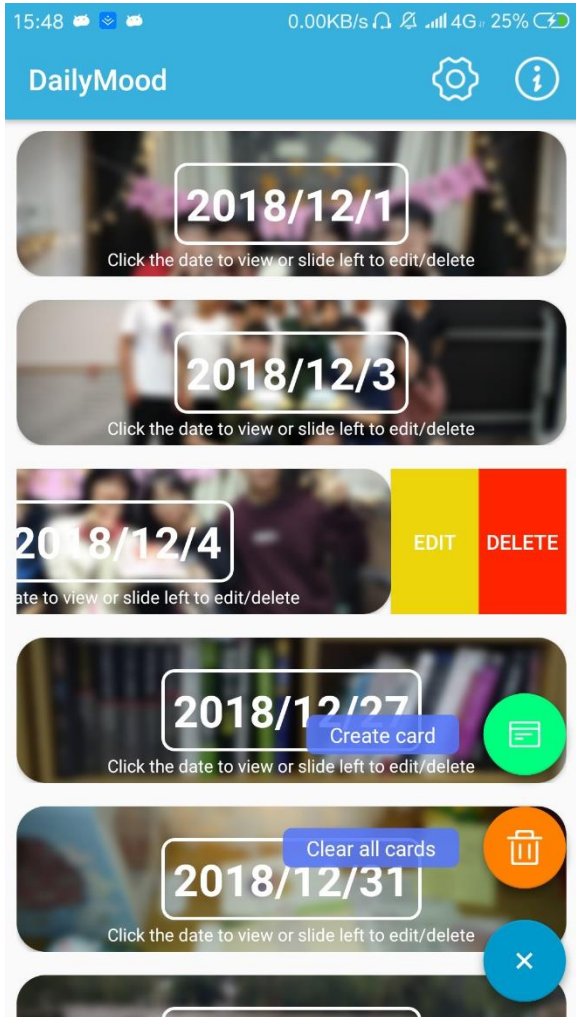
一、软件内容简介

这是一款旨在通过用图片和只言片语记录每日心情的 APP。目前支持心情卡片的创建、编辑、删除、清空；支持心情卡片的查看；支持简单背景音乐的播放；支持国际化。

二、界面设计

（一）主界面（心情卡片列表界面）以及一个国际化示例

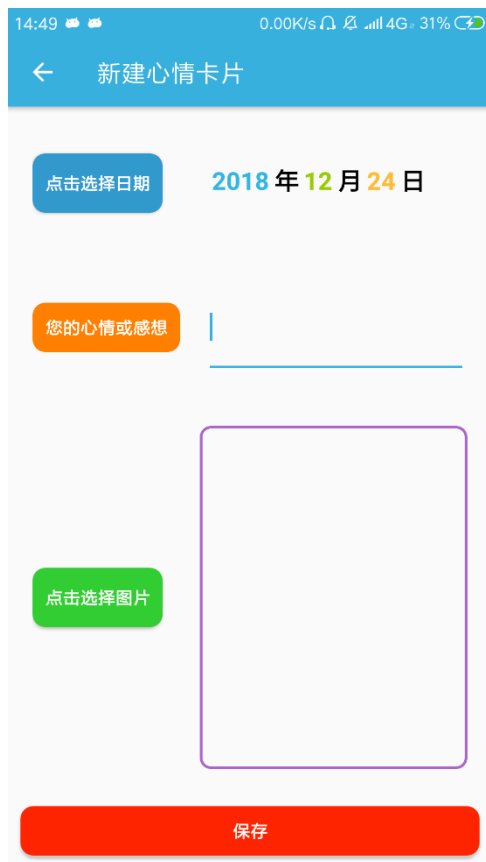




空空如也, 点击右下角添加心情卡片~~



(二) 心情卡片新建界面（初始日期为当日，用户可以保持默认不必输入）



（日历界面不是自己实现的，是用系统提供的 MD 风格的 DatePickerDialog 控件，甚是好
看）

（三）心情卡片编辑界面

11:26

0.44K/s 4G 55%

←

编辑心情卡片

点击选择日期

2018年12月13日

您的心情或感想

今天过生日，很开心~~~

点击选择图片



保存

(四) 心情卡片详情界面



（五）设置界面



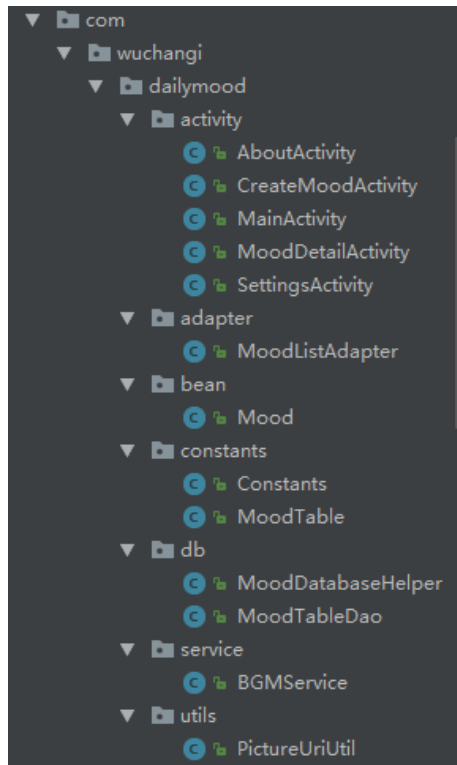
（六）关于界面



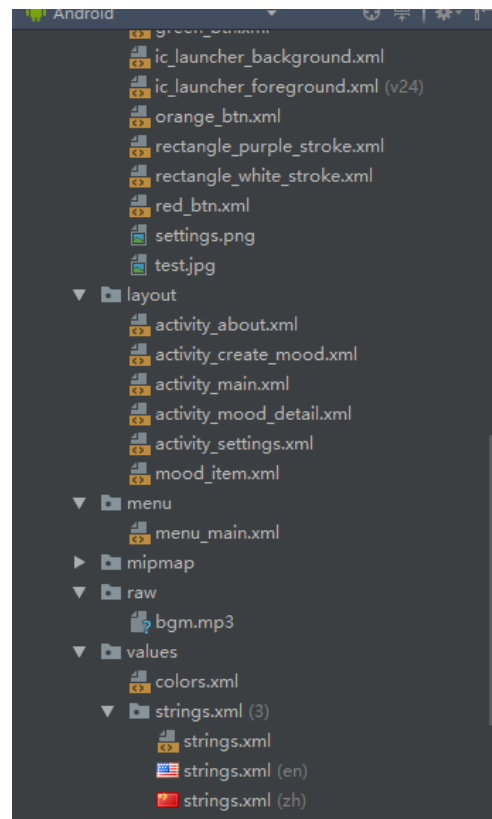
三、代码设计

（鉴于该项目源码有点多，下面仅提供主要部分的截图和代码说明。）

（一）项目分包，模块解耦



Java 源文件分包



xml 布局文件分包

（二）系统的字符串常量的统一安排（统一放置在 string.xml 文件中）

（备注：在老师的要求下，自己把 APP 的国际化也实现了）

1、中文

```
resources string
<resources>
    <string name="app_name">每日心情</string>
    <string name="select_date">点击选择日期</string>
    <string name="year">年</string>
    <string name="month">月</string>
    <string name="day">日</string>
    <string name="description">您的心情或感想</string>
    <string name="select_picture">点击选择图片</string>
    <string name="save">保存</string>
    <string name="please_select_picture">请选择图片</string>
    <string name="save_success">保存成功! </string>
    <string name="welcome">欢迎使用 每日心情 APP~~</string>
    <string name="hint">您需同意该权限才能正常使用该APP~~</string>
    <string name="create_card">新建心情卡片</string>
    <string name="clear_all_cards">清空所有卡片</string>
    <string name="clear_success">清空成功! </string>
    <string name="is_override">已有数据, 是否覆盖? </string>
    <string name="ok">确定</string>
    <string name="cancel">取消</string>
    <string name="cancel_success">取消成功! </string>
    <string name="your_mood">您当天的心情</string>
    <string name="settings">设置</string>
    <string name="about">关于该应用</string>
    <string name="is_clear_all">是否确定清空? </string>
    <string name="empty_hint">空空如也, 点击右下角添加心情卡片~~</string>
    <string name="app_about">一款用图片和只言片语记录每日心情的APP</string>
    <string name="version">当前版本: v 1.0</string>
    <string name="copyright">Copyright © 2018, WuchangI, All Rights Reserved.</string>
    <string name="github_address">项目地址: https://github.com/Yuziquan/DailyMood</string>
    <string name="delete">删除</string>
    <string name="edit">编辑</string>
    <string name="delete_success">删除成功! </string>
    <string name="edit_card">编辑心情卡片</string>
    <string name="modify_success">修改成功! </string>
```

2、英文

```
resources string
<resources>
    <string name="app_name">DailyMood</string>
    <string name="select_date">Select date</string>
    <string name="year"></string>
    <string name="month"></string>
    <string name="day"></string>
    <string name="description">Your mood</string>
    <string name="select_picture">Select picture</string>
    <string name="save">Save</string>
    <string name="please_select_picture">Please select picture</string>
    <string name="save_success">Save success!</string>
    <string name="welcome">Welcome to use DailyMood ~~</string>
    <string name="hint">You need to agree to this permission to use the APP normally~~</string>
    <string name="create_card">Create card</string>
    <string name="clear_all_cards">Clear all cards</string>
    <string name="clear_success">Clear success!</string>
    <string name="is_override">Existing data, whether to override?</string>
    <string name="ok">ok</string>
    <string name="cancel">cancel</string>
    <string name="cancel_success">Cancel success!</string>
    <string name="your_mood">Your mood that day</string>
    <string name="settings">Settings</string>
    <string name="about">About</string>
    <string name="is_clear_all">Whether to clear all?</string>
    <string name="empty_hint">Empty, click on the bottom right corner to add a mood card</string>
    <string name="app_about">An APP that uses pictures and snippets to record your daily</string>
    <string name="version">current version: v1.2</string>
    <string name="copyright">Copyright © 2018, WuchangI, All Rights Reserved.</string>
    <string name="github_address">Github: https://github.com/Yuziquan/DailyMood</string>
    <string name="delete">Delete</string>
    <string name="edit">Edit</string>
    <string name="delete_success">Delete success!</string>
    <string name="edit_card">Edit mood card</string>
    <string name="modify_success">Modify success!</string>
```

(三) CreateMoodActivity（同时充当心情卡片创建界面和编辑界面）

```
@OnClick({R.id.btn_select_date, R.id.btn_select_picture, R.id.btn_save})
public void handleAllClick(View v) {
    switch (v.getId()) {
        case R.id.btn_select_date:
            selectDate();
            break;

        case R.id.btn_select_picture:
            selectPicture();
            break;

        case R.id.btn_save:
            saveMoodCard();
            break;

        default:
            break;
    }
}
```

注册监听事件

```

private void selectDate() {
    DatePickerDialog datePickerDialog = new DatePickerDialog(context, this, (view, year, month, dayOfMonth) -> {
        mYear = year;
        mMonth = month;
        mDay = dayOfMonth;

        mTvYear.setText("" + mYear);
        mTvMonth.setText("" + (mMonth + 1));
        mTvDay.setText("" + mDay);
    }, mYear, mMonth, mDay);

    datePickerDialog.show();
}

```

日期选择

```

private void selectPicture() {
    if (ContextCompat.checkSelfPermission(context, this, Manifest.permission.WRITE_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(activity, this, new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, Constants.OPEN_ALBUM);
    } else {
        openAlbum();
    }
}

private void openAlbum() {
    Intent intent = new Intent(Intent.ACTION_PICK);
    intent.setType("image/*");
    startActivityForResult(intent, Constants.OPEN_ALBUM);
}

```

图片选择（1）

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case Constants.OPEN_ALBUM:
            if (data != null) {
                Uri uri = data.getData();
                mPicturePath = PictureUriUtil.getRealPathFromUri(context, uri);
                mIvPicture.setImageURI(uri);
                isSelectPicture = true;
            }
            break;

        default:
            break;
    }
}

```

图片选择（2）


```

private void saveMoodCard() {
    // 当前为心情卡片创建界面
    if (!isEditCard) {
        if (!isSelectPicture) {
            Toast.makeText( context: this, "请选择图片", Toast.LENGTH_SHORT).show();
            return;
        }

        mDate = mYear + "/" + (mMonth + 1) + "/" + mDay;
        mDescription = mEtDescription.getText().toString();

        final Mood mood = new Mood(mDate, mDescription, mPicturePath);

        // 日期 mDate 已有数据
        if (mMoodTableDao.findMoodByDate(mDate) != null) {

            AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
            builder.setMessage(mDate + " " + getResources().getString(R.string.is_override));
            builder.setPositiveButton("确定", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    mMoodTableDao.addMood(mood);
                    MainActivity.mMoodListAdapter.replaceMood(mood);
                    Toast.makeText( context: CreateMoodActivity.this, "保存成功!", Toast.LENGTH_SHORT).show();

                    finish();
                }
            });
        }
    }
}

```

卡片保存（1）

```

        finish();
    }
}
// 当前为心情卡片编辑界面
else {
    String date = mTvYear.getText() + "/" + mTvMonth.getText() + "/" + mTvDay.getText();
    String description = mEtDescription.getText().toString();
    Mood mood = new Mood(date, description, mPicturePath);

    // 日期 date 已有数据
    if (mMoodTableDao.findMoodByDate(date) != null && !date.equals(getIntent().getStringExtra( name: "date"))) {
        AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
        builder.setMessage(date + " " + "已有数据");
        builder.setPositiveButton("确定", listener: null);
        builder.show();
    } else {
        mMoodTableDao.updateMood(getIntent().getStringExtra( name: "date"), mood);
        MainActivity.mMoodListAdapter.updateMood(getIntent().getStringExtra( name: "date"), mood);
        Toast.makeText( context: CreateMoodActivity.this, getResources().getString(R.string.modify_success), Toast.LENGTH_SHORT).show();
    }

    finish();
}
}
}

```

卡片保存（2）

（四）APP 使用状态的存储和读取

```

mSPRadioButtonsState = getSharedPreferences( name: "radio_buttons_state", MODE_PRIVATE);
mEditor = mSPRadioButtonsState.edit();

// 第一次使用该应用
if(!mSPRadioButtonsState.contains("music_state")){
    // 默认有背景音乐为开启状态
    mEditor.putString( s: "music_state", s1: "on");
    mEditor.commit();
}

// 根据之前保存的状态决定是否播放背景音乐
if(mSPRadioButtonsState.getString( s: "music_state", s1: "on").equals("on")){
    Intent intent = new Intent( packageContext: this, BGMSERVICE.class);
    startService(intent);
}
}

```



```

mBGMSliderVolume.setProgress(mBGMSliderVolume);

// 设置界面中的单选按钮被选中的状态
mSPRadioButtonsState = getSharedPreferences( name: "radio_buttons_state", MODE_PRIVATE);
mEditor = mSPRadioButtonsState.edit();

if (mSPRadioButtonsState.getString( s: "music_state", s1: "on").equals("on")) {
    mRbBGMon.setChecked(true);
} else {
    mRbBGMOff.setChecked(true);
}

mRgBGMSwitch.setOnCheckedChangeListener((radioGroup, i) -> {
    Intent intent = new Intent( packageContext: SettingsActivity.this, BGMService.class);

    switch (i){
        case R.id.rb_bgm_on:
            startService(intent);
            mEditor.putString( s: "music_state", s1: "on");
            break;

        case R.id.rb_bgm_off:
            stopService(intent);
            mEditor.putString( s: "music_state", s1: "off");
            break;
    }

    mEditor.commit();
});

```

(五) 背景音乐的简易控制 (BGMService.java)

```

private void startBGM() {
    if(mBGMPPlayer == null){
        mBGMPPlayer = MediaPlayer.create( context: this, R.raw.bgm);
        mBGMPPlayer.start();

        mBGMPPlayer.setOnCompletionListener((mediaPlayer) -> {
            mBGMPPlayer.start();
            //设置循环播放
            mBGMPPlayer.setLooping(true);
        });
    }
}

private void stopBGM() {
    if(mBGMPPlayer != null && mBGMPPlayer.isPlaying()){
        mBGMPPlayer.stop();
        mBGMPPlayer.release();
        mBGMPPlayer = null;
    }
}

```

（六）SQLite 数据库的应用

```
public class MoodTable {  
  
    public static final String TABLE_NAME = "t_mood";  
  
    public static final String COL_DATE = "date";  
    public static final String COL_DESCRIPTION = "description";  
    public static final String COL_PICTURE_PATH = "picture_path";  
  
    public static final String CREATE_TABLE = "CREATE TABLE "  
        + TABLE_NAME + " ("  
        + COL_DATE + " TEXT PRIMARY KEY, "  
        + COL_DESCRIPTION + " TEXT, "  
        + COL_PICTURE_PATH + " TEXT);";  
}
```

```
public class MoodTableDao {  
  
    private MoodDatabaseHelper mDbHelper;  
  
    public MoodTableDao(Context context) { mDbHelper = new MoodDatabaseHelper(context); }  
  
    public List<Mood> listMoods(){  
        SQLiteDatabase db = mDbHelper.getReadableDatabase();  
  
        String sql = "SELECT * FROM " + MoodTable.TABLE_NAME + ";";  
        Cursor cursor = db.rawQuery(sql, null);  
  
        List<Mood> moodList = new ArrayList<>();  
        Mood mood = null;  
  
        while(cursor.moveToNext()){  
            mood = new Mood();  
            mood.setDate(cursor.getString(cursor.getColumnIndex(MoodTable.COL_DATE)));  
            mood.setDescription(cursor.getString(cursor.getColumnIndex(MoodTable.COL_DESCRIPTION)));  
            mood.setPicturePath(cursor.getString(cursor.getColumnIndex(MoodTable.COL_PICTURE_PATH)));  
  
            moodList.add(mood);  
        }  
  
        cursor.close();  
  
        return moodList;  
    }  
  
    public void addMood(Mood mood) {  
        SQLiteDatabase db = mDbHelper.getWritableDatabase();  
  
        ContentValues values = new ContentValues();  
        values.put(MoodTable.COL_DATE, mood.getDate());  
        values.put(MoodTable.COL_DESCRIPTION, mood.getDescription());  
        values.put(MoodTable.COL_PICTURE_PATH, mood.getPicturePath());  
        db.insert(MoodTable.TABLE_NAME, null, values);  
    }  
}
```

四、软件操作流程

（一）用户可点击主界面右下角的蓝色悬浮按钮，在弹出的菜单项中执行心情卡片的新建和清空操作；

（二）用户可点击主界面上方的设置按钮进入设置界面，对系统背景音乐进行相应的设置；

（三）用户在主界面的卡片列表中，用手指点击卡片日期可以查看卡片的详情页；

（四）用户在主界面的卡片列表中，用手指左滑选中卡片，可以对相应卡片执行编辑和删除操作。

五、难点和解决方案

（一）获取图片的真实路径（绝对路径）

选择系统相册中的图片时，返回的 Uri 不是绝对路径，需要转换为图片的绝对路径，这里使用的是网上某个大神的代码（主要是 `getRealPathFromUri(Context context, Uri uri)` 方法），我将其放置到 `PictureUriUtil` 工具类中以备调用。

（二）提示语的显示和隐藏（Activity 间的简易通讯实现方式）

当主界面为空时，主界面的提示语会显示；当主界面非空时，主界面的提示语会隐藏。这涉及到卡片新建界面和主界面之间的通讯，这里采用的实现方案是 `Eventbus` 第三方开源库。

（三）优化卡片列表的加载，使之滑动顺畅

这里的优化方案具体包括：

1、尽量减少卡片列表子项的布局嵌套（因为系统测量嵌套的 view 会花费很多时间，造成界面卡顿）

2、使用 `Glide` 第三方库加载相册图片进卡片列表封面，并实现毛玻璃效果

（若使用系统原生的方法来加载相册图片，因为相册图片大小较大，故会造成界面滑动卡顿，而 `Glide` 自带缓存方案，可优化加载速度）

原生方法：

```
holder.mlvCover.setImageBitmap(BitmapFactory.decodeFile(mood.getPicturePath()));
```

改进方法：

```
Glide.with(mContext).load(mood.getPicturePath()).into(holder.mlvCover);
```

3、使用 `RecyclerView`，并借助其自带的 `ViewHolder` 机制实现 view 的重用（若使用 `ListView`，则需自己管理 `ViewHolder`，况且谷歌首推 `RecyclerView` 来实现列表效果，当然用起来~~）

（四）日期重叠的处理

有时候用户新建/编辑的心情卡片的日期会与之前的卡片相同，这里使用 SQLite 的查操作判断当前数据库中是否有已存在的且日期相同的卡片，有则弹出对话框给予提示信息，让用户选择是否覆盖之前的卡片数据。

六、不足之处和今后的设想

该应用的 idea 纯属原创，诞生自本人冬至之前的某一天的想法，遂进行界面的手稿设计和编码的实现，主打简洁实用风。目前该 App 的功能还较为简单，后期有时间将在 Github 上继续迭代更新和维护。不得不说，Android 的界面设计占用了大部分的时间。

七、参考文献

1. 《Android 进阶之光》