

小米便签开源软件的维护方案及成果

撰写人：谢昌熔、樊译文、李天鸿、花涛

1. 开源软件简介

本产品是由小米团队开发的小米便签软件，在 github 上开源，编程语言为 java 语言，应用环境为 Android 操作系统，代码规模约为 7404 行。

功能：智能识别电话网址邮箱、正文清单模式、桌面 Widget 可选择颜色、自定义字体大小、支持中文简体及外文、可将便签文件备份至 SD 卡或邮箱并可导入还原、设置密码、批量输出文本至 SD 卡或邮箱、创建文件夹将便签分类管理、创建桌面工具、用不同颜色区分便签优先级、将便签设置闹钟提醒并可用分享给好友、在便签内加入照片、语音输入、字数统计、登录注册和云同步、改变便签视图、背景和排序规则等。

2. 软件维护需求描述

2.1 开源软件分析

小米便签作为一个已经投入市场并广泛应用的 APP，功能完善，底层逻辑周密，使用体验总体良好，但仍存在一些不足与尚可改进之处。UI 方面，便签排版、界面切换动画仍显生硬、粗糙，影响用户体验；便签云端功能有所欠缺，便签不能独立同步于多态设备；不注重便签的隐私性，并且无区别化对待，无法实现数据的差异化；便签的生命周期处理不完善，无法形成闭环，造成一系列小问题，影响结果导向。

2.2 开源软件维护需求

1. 新增功能描述

改变文件夹视图：优化 UI，改善外观，增强用户交互。

便签按时间排序：针对小米便签的排序单一做出升级，提高用户浏览、查找便签的效率。

添加密码：可为某个便签单独设立二级密码，保护隐私。

语音输入：采用讯飞开源语音输入。解决语音输入只能依靠输入法的问题。

字数统计：针对当前便签的已输入文本进行字数统计，在上方便签栏自动显示。没有多余的需要用户进行的操作，功能效果快捷直观，提高用户效率。

登录注册、同步功能：为小米便签单独设立账号，数据上传至云端。

切换背景：用户可自定义背景图片，美化软件界面

回收站：增加回收站并完善，使其兼容其它已有功能（如便签加密）

2. 代码缺陷描述

data/Task.java

A "NullPointerException" could be thrown; "js" is nullable here. 为何是问题? 12天前 L185 评论 cert, cwe

data/TaskList.java

A "NullPointerException" could be thrown; "js" is nullable here. 为何是问题? 12天前 L138 评论 cert, cwe

Remove this conditional structure or edit its code blocks so that they're not all the same. 为何是问题? 12天前 L203 评论 无标签

remote/GTaskClient.java

Cast one of the operands of this multiplication operation to a "long". 为何是问题? 12天前 L115 评论 cert, cwe, overflow, sans-top25-risky

Use try-with-resources or close this "BufferedReader" in a "finally" clause. 为何是问题? 12天前 L311 评论 cert, cwe, denial-of-service, leak

remote/GTaskManager.java

"NullPointerException" will be thrown when invoking method "doContentSync()". 为何是问题? 12天前 L334 评论 cert, cwe

"NullPointerException" will be thrown when invoking method "doContentSync()". 为何是问题? 12天前 L471 评论 cert, cwe

remote/GTaskManager.java

Use try-with-resources or close this "FileOutputStream" in a "finally" clause. 为何是问题? 38分钟前 L299 评论 cert, cwe, denial-of-service, leak

Do something with the "boolean" value returned by "createNewFile". 为何是问题? 38分钟前 L331 评论 cert, cwe, error-handling

1 "NullPointerException" could be thrown; "js" is nullable here.

```

174 ... }
175 }
176 }
177
178 public void setContentByLocalJSON(JSONObject js) {
179     if (js == null || !js.has(GTaskStringUtils.META_HEAD_NOTE)
180         || !js.has(GTaskStringUtils.META_HEAD_DATA)) {
181         Log.w(TAG, "setContentByLocalJSON: nothing is available");
182     }
183
184     try {
185         JSONObject note = js.getJSONObject(GTaskStringUtils.META_HEAD_NOTE);
186
187         JSONArray dataArray = js.getJSONArray(GTaskStringUtils.META_HEAD_DATA);
188
189         if (note.getInt(NoteColumns.TYPE) != Notes.TYPE_NOTE) {
190             Log.e(TAG, "invalid type");
191             return;
192         }
193
194         for (int i = 0; i < dataArray.length(); i++) {
195             JSONObject data = dataArray.getJSONObject(i);

```

A "NullPointerException" could be thrown; "js" is nullable here. 为何是问题? 12天前 L185 评论 cert, cwe

原因：图中虽然对 js 进行了 null 判断，但是在新建 note 这个对象的时候，js.getJSONObject（）返回对象可能失败，应当对此种情况进行考虑并做预判处理。

解决方法：增加 js 为空的判断，但是如果只是增加 js 为空的判断，还会引入 JSONException 的错误，因此使用 try catch 方法。

```

try {
    if(js != null) {
        JSONObject note = js.getJSONObject(GTaskStringUtils.META_HEAD_NOTE);
        JSONArray dataArray = js.getJSONArray(GTaskStringUtils.META_HEAD_DATA);

        if (note.getInt(NoteColumns.TYPE) != Notes.TYPE_NOTE) {
            Log.e(TAG, msg: "invalid type");
            return;
        }
    }
}

```

对 null 的引用不应该被引用/访问。这样做将导致抛出 `NullPointerException`。在最好的情况下，这样的异常将导致程序突然终止。在最坏的情况下，它可能会暴露对攻击者有用的调试信息，或者允许攻击者绕过安全措施。出现这种情况时，可以利用 JSR-305 中定义的 `@CheckForNull` 和 `@NonNull` 注释来理解哪些值可以为空，哪些值不能为空，除非在参数 `toEquals` 上使用 `@NonNull`，根据约定，这应该始终与 `null` 一起工作。前两个 bug 均为此问题。

2 Remove this conditional structure or edit its code blocks so that they're not all the same.

```

// apply remote to local
return SYNC_ACTION_UPDATE_LOCAL;
}
} else {
    // validate gtask id
    if (!c.getString(SqlNote.GTASK_ID_COLUMN).equals(getGid())) {
        Log.e(TAG, "gtask id doesn't match");
        return SYNC_ACTION_ERROR;
    }
    if (c.getLong(SqlNote.SYNC_ID_COLUMN) == getLastModified()) {

```

Remove this conditional structure or edit its code blocks so that they're not all the same. 12天前 L203

为何是问题?

Bug 主要 打开 未分配 15min 工作 评论 无标签

```

// local modification only
return SYNC_ACTION_UPDATE_REMOTE;
} else {
    // for folder conflicts, just apply local modification
    return SYNC_ACTION_UPDATE_REMOTE;
}
}
} catch (Exception e) {

```

原因：一个 if else 判断语句，但里面执行的语句确实一样的，属于逻辑错误

解决方法：将第一个 if 的返回值改为：SYNC_ACTION_UPDATE_LOCAL

```

if (c.getLong(SqlNote.SYNC_ID_COLUMN) == getLastModified()) {
    // local modification only
    return SYNC_ACTION_UPDATE_REMOTE;
} else {
    // for folder conflicts, just apply local modification
    return SYNC_ACTION_UPDATE_LOCAL;
}
}

```

3 Cast one of the operands of this multiplication operation to a "long".

```

public static synchronized GtaskClient getInstance() {
    if (mInstance == null) {
        mInstance = new GtaskClient();
    }
    return mInstance;
}

public boolean login(Activity activity) {
    // we suppose that the cookie would expire after 5 minutes
    // then we need to re-Login
    final long interval = 1000 * 60 * 5;
}

```

Cast one of the operands of this multiplication operation to a "long". 为何是问题? 12天前 L115

Bug 次要 打开 未分配 5min 工作 评论 cert, cwe, overflow, sans-top25-risky

```

if (mLastLoginTime + interval < System.currentTimeMillis()) {
    mLoggedIn = false;
}

// need to re-Login after account switch
if (mLoggedIn
    && !TextUtils.equals(getSyncAccount().name, NotesPreferenceActivity
        .getSyncAccountName(activity))) {
}

```

原因：如果 int 除法的结果被赋值给浮点变量，那么在赋值之前就会丢失精度。同样，如果乘法的结果被赋值为 long，那么它可能在赋值之前就已经溢出了。在这两种情况下，结果都不符合预期。上图便属于第二种情况。

解决方法：但这里 300000 还没有超过整型范围，因此这个错误可以不考虑。

4 Use try-with-resources or close this "BufferedReader" in a "finally" clause.

```

InputStream input = entity.getContent();
if (contentEncoding != null && contentEncoding.equalsIgnoreCase("gzip")) {
    input = new GZIPInputStream(entity.getContent());
} else if (contentEncoding != null && contentEncoding.equalsIgnoreCase("deflate")) {
    Inflater inflater = new Inflater(true);
    input = new InflaterInputStream(entity.getContent(), inflater);
}

try {
    InputStreamReader isr = new InputStreamReader(input);
    BufferedReader br = new BufferedReader(isr);
}

```

Use try-with-resources or close this "BufferedReader" in a "finally" clause. 为何是问题? 12天前 L311

Bug 阻断 打开 未分配 5min 工作 评论 cert, cwe, denial-of-service, leak

```

StringBuilder sb = new StringBuilder();

while (true) {
    String buff = br.readLine();
    if (buff == null) {
        return sb.toString();
    }
}

```

原因：上图中开启了两个文件流，但是在最后关闭时执行的 finally 块里面，只有“finally{input.close();}”关闭了输入流，还有一个输出流没有关闭

解决方法：参照相关规范，可以运用“if(fin!=null) fin.close(); if(fout!=null) fout.close()”这样的类似方法，在 finally 块里面关闭流。当然也可以将两个文件放在不同的 try 来进行处理。

5 "NullPointerException" will be thrown when invoking method "doContentSync():"

```

// go through remaining items
Iterator<Map.Entry<String, Node>> iter = mGTaskHashMap.entrySet().iterator();
while (iter.hasNext()) {
    Map.Entry<String, Node> entry = iter.next();
    node = entry.getValue();
    doContentSync(Node.SYNC_ACTION_ADD_LOCAL, node, null);
}

```

"NullPointerException" will be thrown when invoking method "doContentSync()". 为何是问题? 12天前 L334

Bug 主要 打开 未分配 10min 工作 评论 cert, cwe

```

}

```

```

// mCancelled can be set by another thread, so we need to check one by
// one
// clear local delete table
if (!mCancelled) {
    if (!DataUtils.batchDeleteNotes(mContentResolver, mLocalDeleteIdMap)) {
        throw new ActionFailureException("failed to batch-delete local deleted notes");
    }
}

```

原因：调用方法“doContentSync()”时将抛出空指针

6 Do something with the "boolean" value returned by "createNewFile".

```

try {
    if (!filedir.exists()) {
        filedir.mkdir();
    }
    if (!file.exists()) {
        file.createNewFile();
    }
}

```

Do something with the "boolean" value returned by "createNewFile". 为何是问题? 36分钟前 L331

Bug 次要 打开 未分配 15min 工作 评论 cert, cwe, error-handling

```

}
return file;
} catch (SecurityException e) {
    e.printStackTrace();
} catch (IOException e) {
}

```

原因：createNewFile 有一个返回值，没有可以赋值的对象

方法：加个变量赋值。

3. 维护设计和实现方案

3.1 开源软件体系结构的设计及调整

我们进行功能的添加主要通过对已有的类添加新方法的方式来实现，主要在 NotesListActivity、NotesEditActivity 等类上做修改，在 NotesListActivity 类中加入了 showListDialog、ul_upSync

、unlogin、downSync、change_nav_item、upSync、register、login、displayIaage、getImagePath、handeImage、openAlbum、

bg_from_photo、bg_from_camera、orderByTime、restoreDeletedNodes 等方法。在 NotesEditActivity 中主要添加了 showTip、startSpeechDialog、onRequestPermissionsResult、initPermission、initSpeech、hideSoftInput、showSoftInput、delete_secret、add_secret、showToast 等方法与 mRecoListener、MyInitListener、MyRecognizerDialogListener 这几个成员类，此外引入了 DBConnection 与 JsonParser 两个类。其主要的关系如下：

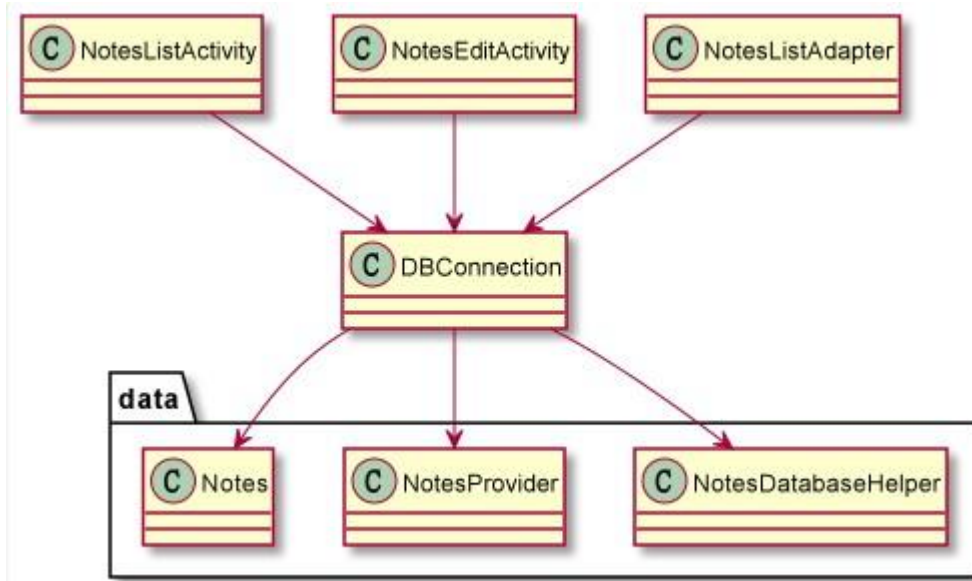


图 1 DBConnection 类关系

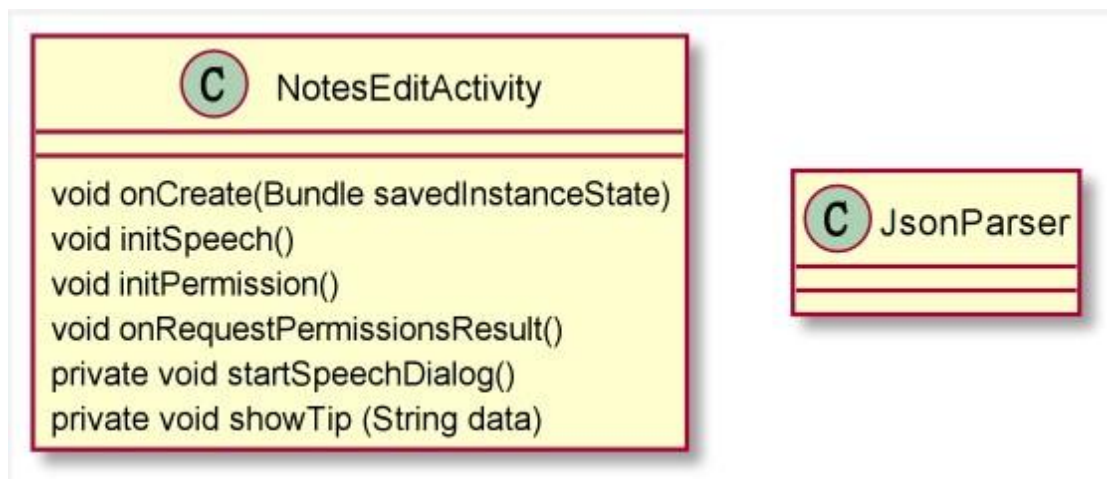


图 2 JsonParser 类的关系

新添加的类中部分功能如下：

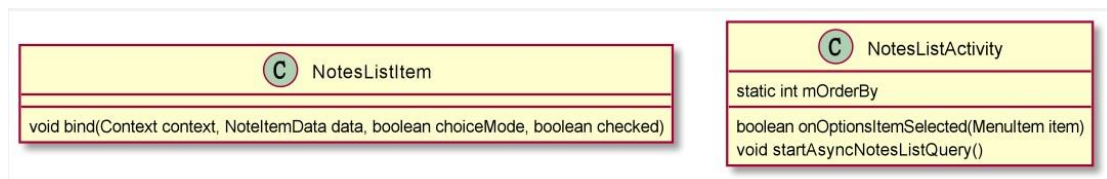


图 3 时间排序功能中的新方法

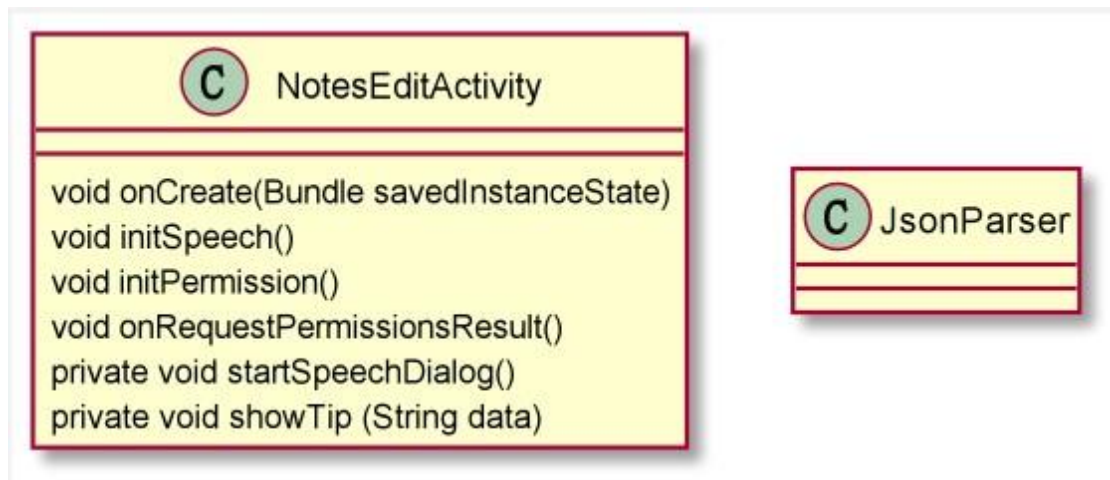


图 4 语音输入功能中的新方法

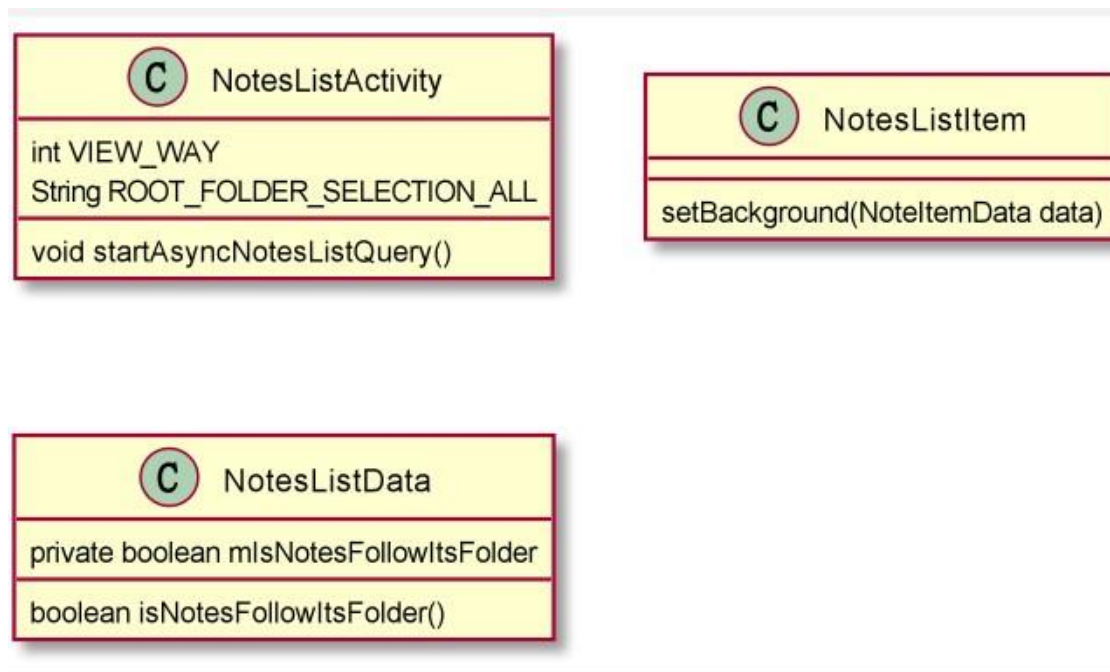


图 5 切换文件夹视图功能中的新方法

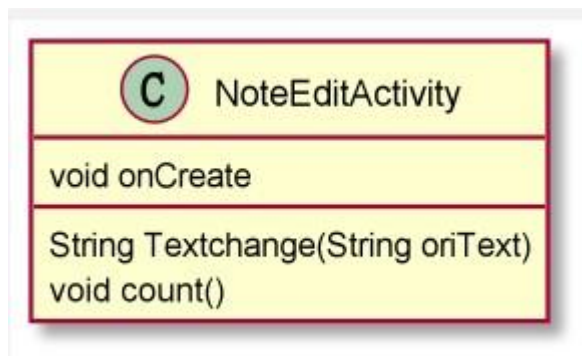


图 6 字数统计中的新方法

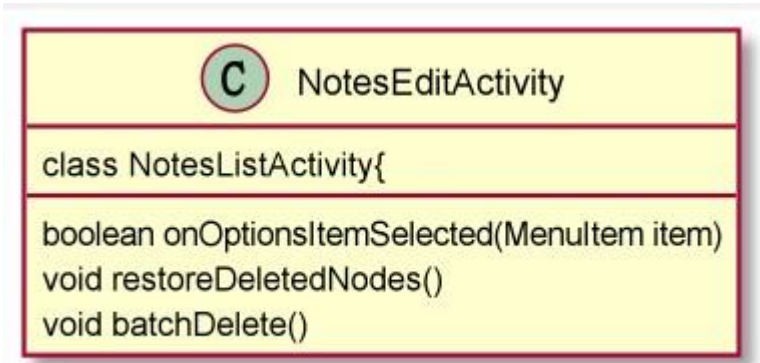


图 7 回收站功能中的新方法

3.2 维护需求的实施方案

1、登陆注册

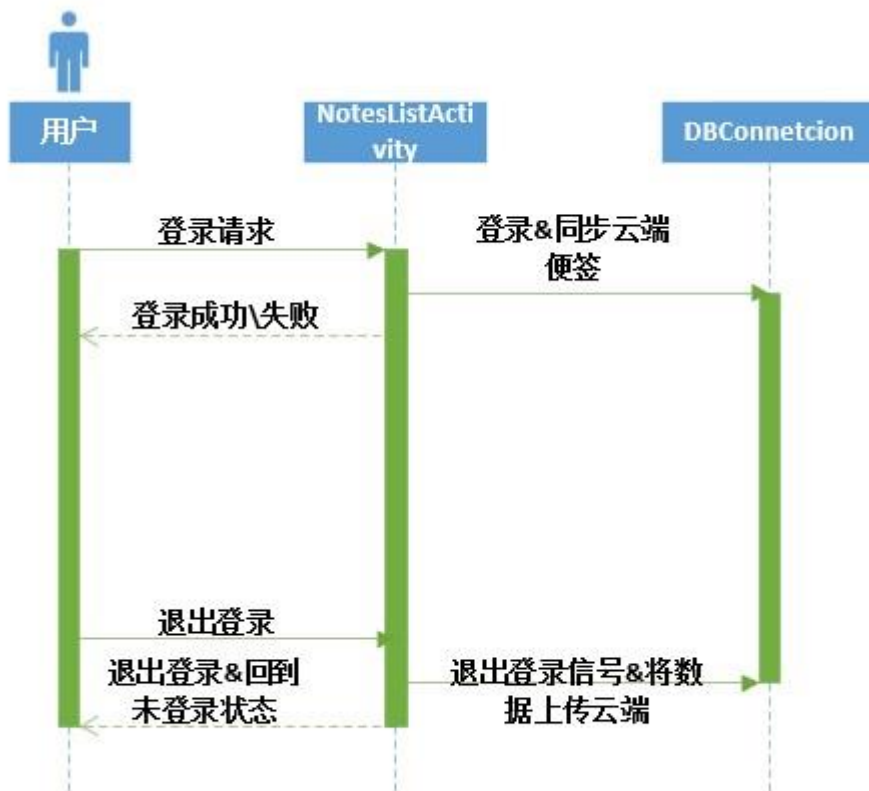


图 8 登录注册顺序图

用户通过 NotesListActivity 类发出登录请求到 DBConnection，并验证密码是否正确，如果正确则登录成功并将云端的便签数据库下拉。当退出登陆时，发出退出登录的请求，DBConnectino 将便签数据上传到云端并回到未登录状态。

2、便签加密

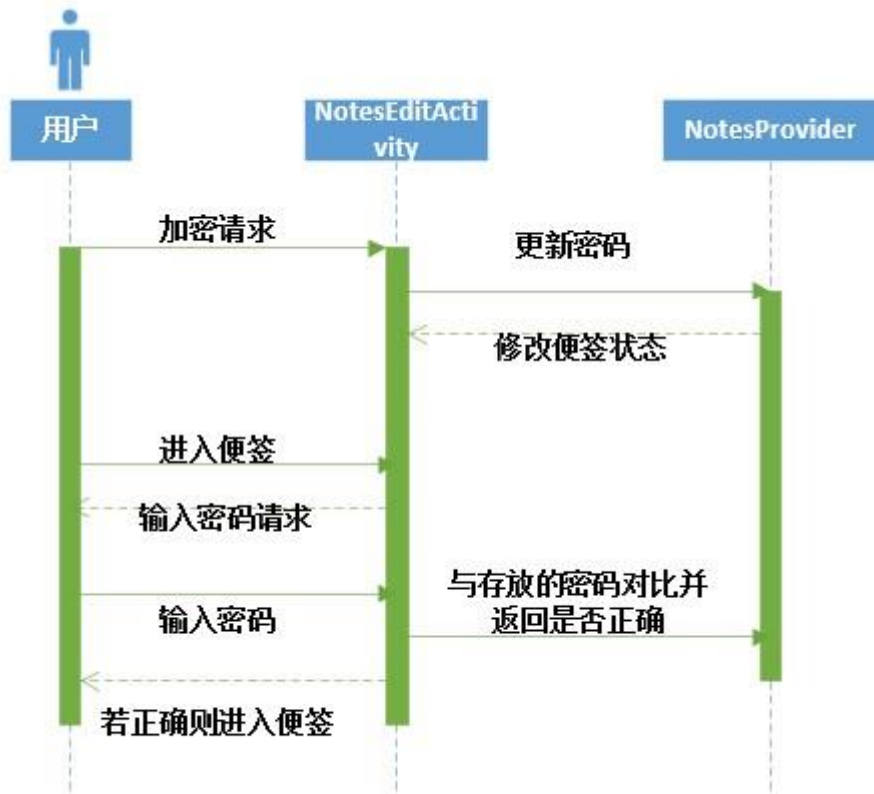


图 9 便签加密顺序图

用户通过在 NotesEditActivity 类中发起加密请求，随后 NotesEditActivity 类将用户输入的密码通过 NotesProvider 类添加到数据库中。

当用户再次进入便签时，需要重输入密码，此时 NotesEditActivity 类将输入的密码与数据库中存放的密码做比较，如果正确则进入便签，如果错误则让用户重新输入。

3、语音输入

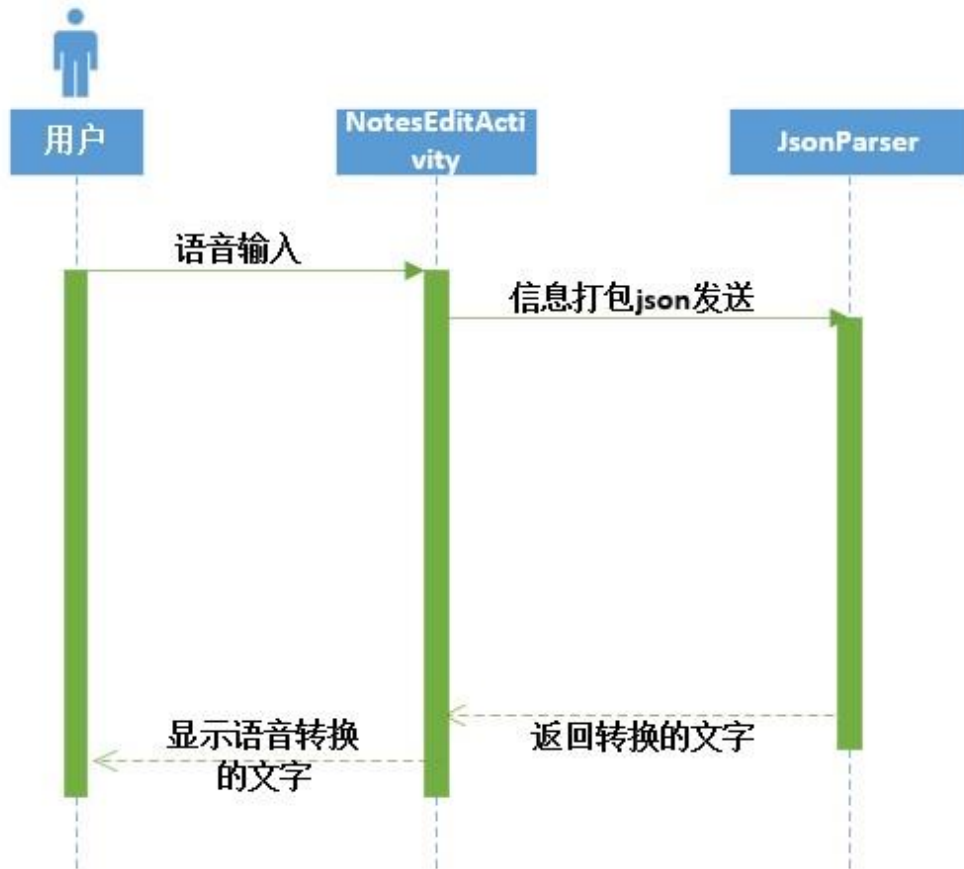


图 10 语音输入顺序图

用户通过 NotesEditActivity 发起语音输入请求，调用麦克风记录用户的音频输入后发送到 JsonParser 类中，并打包成 json 文件使用科大讯飞的语音转文字的包来转化为文字，随后将转化而得的文字返回给 NotesEditActivity 类，再由 NotesEditActivity 类将其回显在便签界面中。

4、替换背景

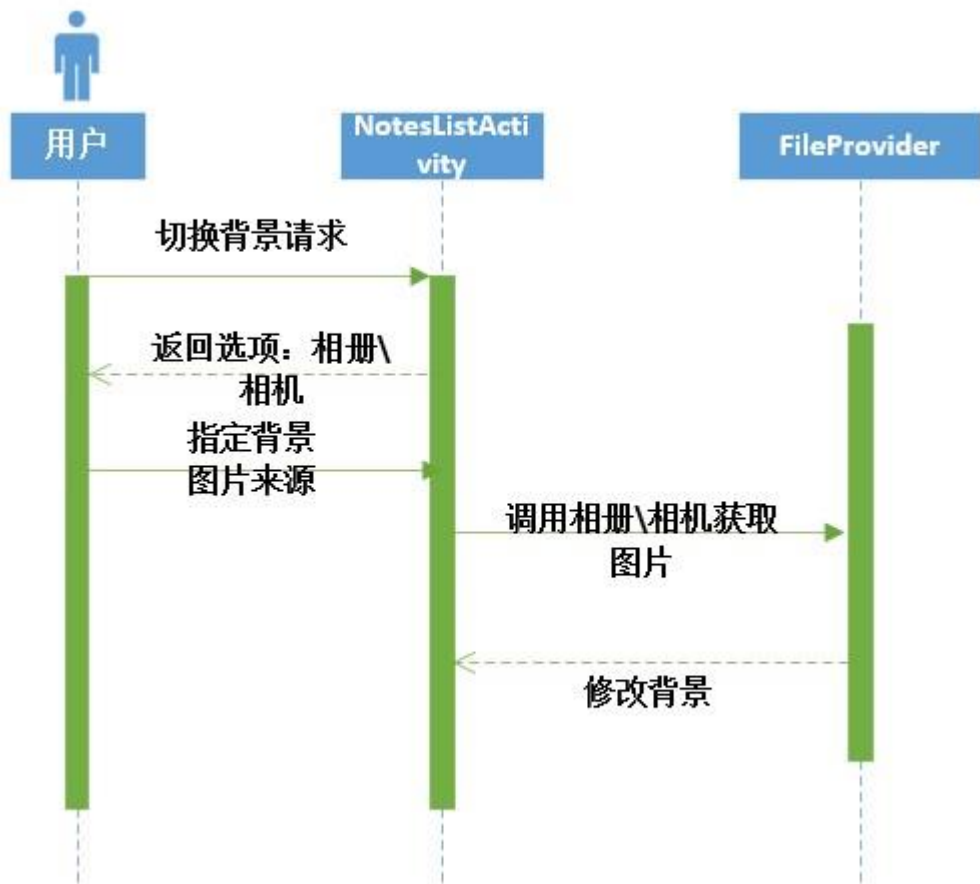


图 11 修改背景顺序图

用户通过 NotesListActivity 类发出切换背景请求，并输入图片来源：相册或者相机。当选定图片来源后，FileProvider 调用相机或相册获取图片并返回给 NotesListActivity 类，由 NotesListActivity 类进行修改背景的操作。

3.3 维护需求的类设计

1、DBConnetion 类：

- link_register：用于用户最开始的账号注册
- link_login：用户登录时运行的函数
- link_upSync：用于将本地的数据同步到远端
- link_downSync：用于将远端的数据下拉到本地

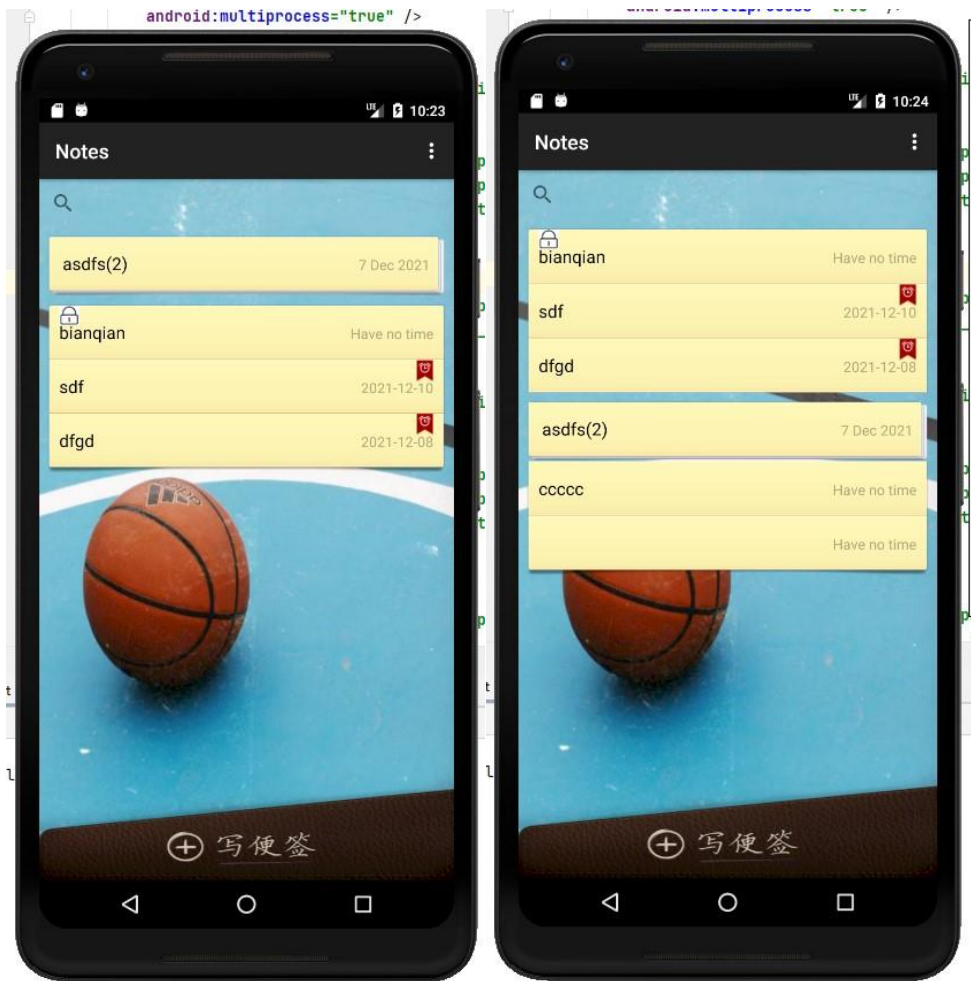
2、JsonParser 类：

- parseGrammarResult：用于将用户输入的语音转换为指定的格式并打包为 json
- parseLocalGrammarResult：用户将用户输入的语音在本地转换为文字

4. 维护后的开源软件

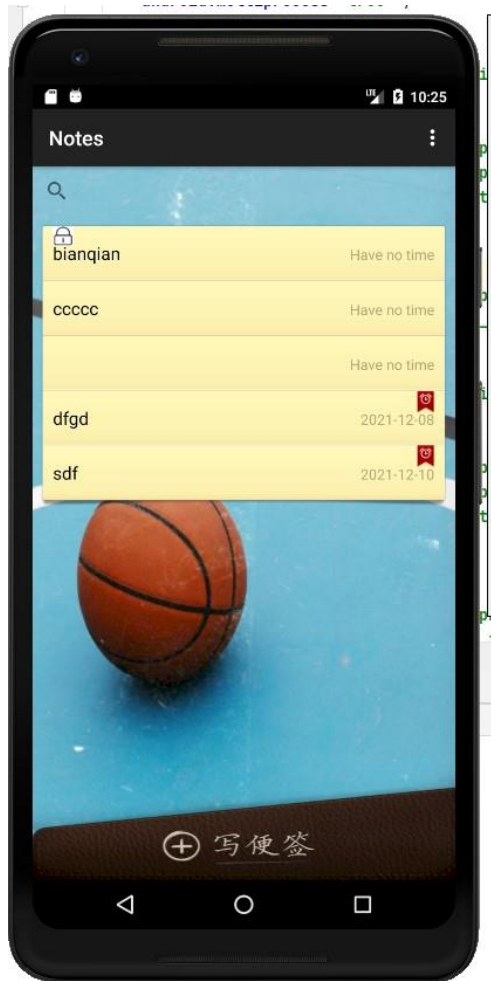
4.1 维护后的开源软件功能及演示

- 1、改变文件夹视图



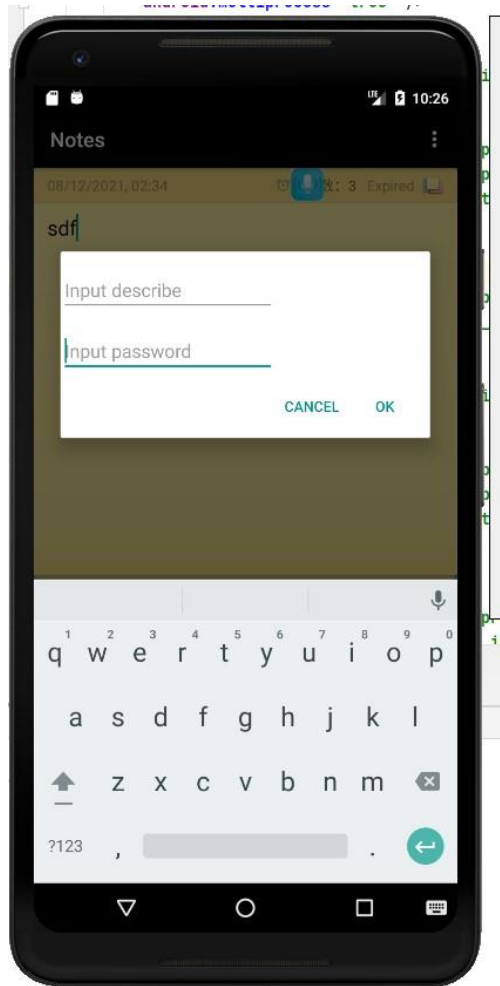
说明：将所有文件夹中的便签展示出来，方便用户查找。左位初始界面。

2、便签按事件时间排序



说明：将便签按创建时间进行排序，更有条理性。

3、添加密码



说明:对便签进行加密,保护用户的隐私,可以输入便签的描述便于用户识别加密便签,然后输入密码加密完成。

4、语音输入



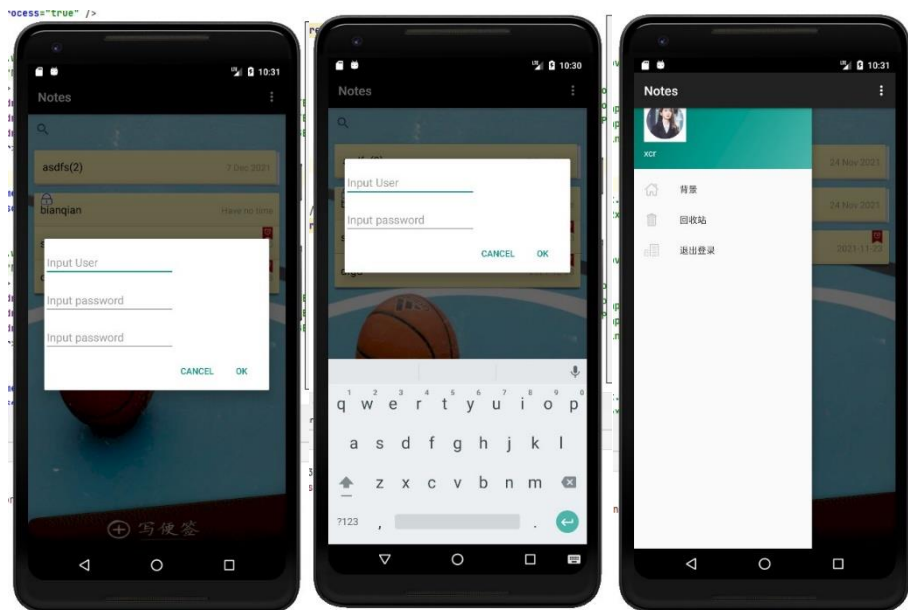
说明：打开便签后点击上面的话筒图标，就可以进行语音收集识别，然后将用户的语言编辑到便签中。

5、字数统计



说明：便签右上角有个字符数，当输入字符后会自动进行字数的统计。

6、登录注册



说明：左滑唤出侧边栏，点击注册，进入注册界面，输入用户名和密码，然后确认密码注册完毕。点击登录，然后输入用户名和密码登陆账号，图三显示成功登陆账号。

7、更换背景



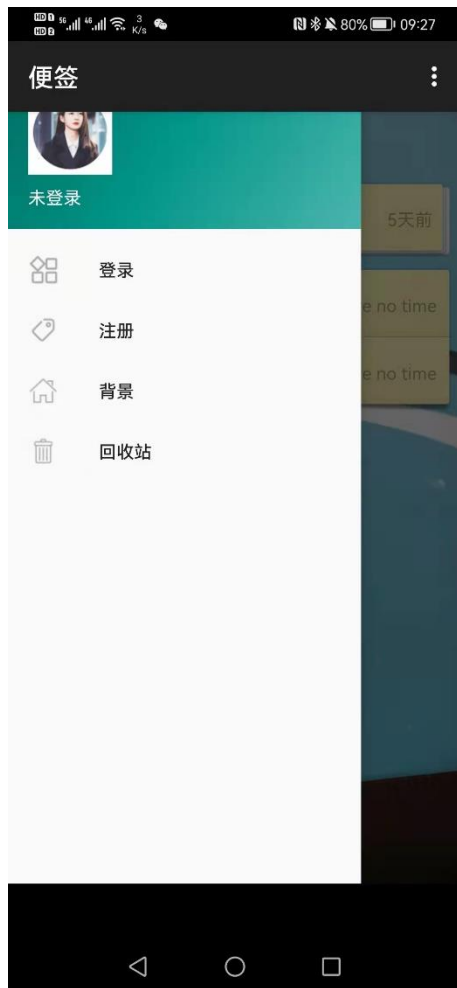
说明：左滑唤出侧边栏，点击背景，可以选择从相册选取或者使用相机拍摄照片更换背景。

8、回收站



说明：删除的便签会暂存在回收站，方便用户将误删的便签恢复。

9、侧边栏



4.2 对新增或修改的代码质量进行分析

类别	定性和定量描述
开源软件原先代码量	7404 行
标注的代码行数量	912 行
新增的代码量	1662 行
新增代码中重用的代码量	345 行
新增代码中独立编写的代码量	1317 行

Sonarqube 对软件维护前后分析情况

代码行数	7,404	代码行数	9,066
行数	10,051	行数	11,948
语句	3,587	语句	4,428
方法	471	方法	527
类	65	类	69
文件	39	文件	41
注释 (%)	5.2%	注释 (%)	4.6%
注释行	403	注释行	438

图 12 维护前后代码量



图 13 维护前 sonarqube 分析结果



图 14 维护后 sonarqube 分析结果