

# 小米便签应用代码泛读、标注和维护报告

黄泽楷 秦薪淇

2025 年 1 月 9 日

## 摘要

MiCode便签是MIUI团队发起的开源Android应用项目。该项目使用Java语言开发,代码结构完整,涵盖了Android应用开发的核心功能模块,包括用户界面、数据存储、系统服务等。项目总计约13000行代码,包含6个功能包和39个类,采用MIT开源协议。这个应用从Android开发者角度来看,虽然简单,但五脏俱全,非常适合作为初学者的参考。

本报告针对小米便签开源项目进行了系统的代码泛读、标注和维护工作。通过对项目源码的分析,我们完成了以下主要工作:

首先,对项目进行了全面的代码泛读,分析了软件的功能结构、系统架构以及各模块间的交互关系。通过泛读发现该项目采用了典型的MVC架构,包含界面层、业务层、模型层和数据层四层结构。

其次,对项目的6个包共39个类进行了详细的代码标注工作,标注代码行数达2350行。标注内容包括类的功能说明、关键方法实现逻辑、重要变量含义等,有助于理解代码结构与实现细节。

最后,在理解代码的基础上进行了功能维护与扩展。新增了包括登录密码保护、翻译功能、富文本编辑、朗读功能等10项新功能,显著提升了软件的实用性与用户体验。

通过本次实践,我们不仅掌握了面向大型项目的代码分析方法,也加深了对Java编程规范和软件工程开发流程的理解。

# 目录

# 1 小米便签的代码泛读

## 1.1 功能描述

根据对开源软件的使用以及对代码的阅读和理解，该软件的整体功能描述如下，其软件需求的用例模型如图1所示。

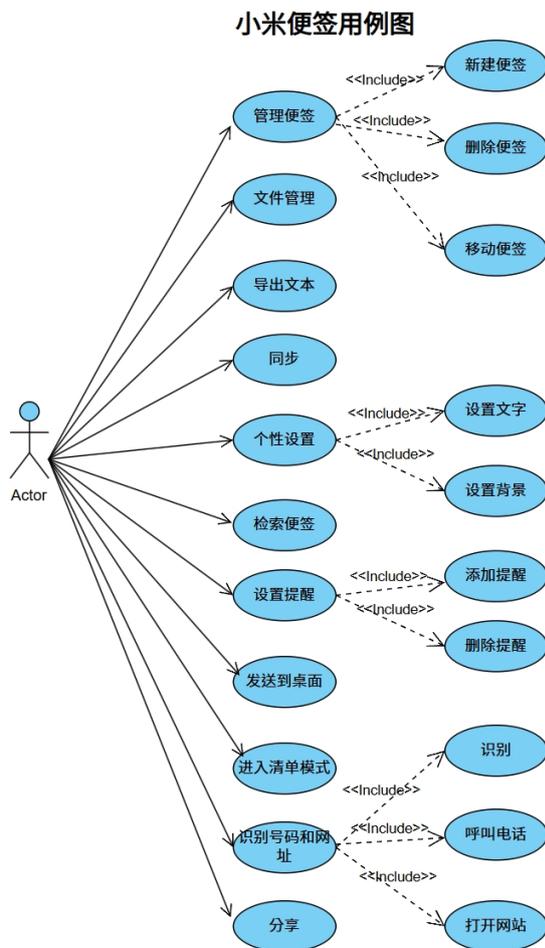


图 1: 小米便签的软件需求用例图

下面根据用例图详细阐述小米便签的功能:

- 功能1: 新建/删除/移动便签

1. 新建便签: 通过小米便签软件的主界面下方的”写便签”快捷键和文件夹、便签视图下的选项”新建便签”可以在当前目录创建一个便签并打开进入文本编辑。其中主界面下的快速创建方式可以迅速创建一个待编辑的便签,用于满足临时迅速记录的需求。

2. 删除便签：通过长按便签并选择删除选项可以删除已创建的便签，用于对便签集进行有效的管理。
3. 移动便签：通过长按便签并选择移动选项可以将已创建的便签移动到某个文件夹中，用于在快速创建便签后对便签集进行有效的管理。

- 功能2：文件管理

在当前目录下新建、查看、删除一个文件夹用于分类管理便签。用户在主界面的选项中选择“新建文件夹”创建一个文件夹，根据create参数值设置对话框标题（创建或是修改文件夹），设置对话框的正确按钮和取消按钮。

- 功能3：导出文本

在主界面的选项中选择”导出文件”，在Android手机提供SD卡支持的情况下，将小米便签中的便签内容逐个转化为.txt的文本文件。

- 功能4：同步

在主界面的选项中选择”同步”，与Google Task中的备忘录事项，将本地的事项上传到服务器，或将Google服务器上的表单下载到本地。

- 功能5：检索便签

在主界面的选项中选择”搜索”，通过关键词查找到包含该关键词的便签，显示在界面上。

- 功能6：修改字体大小

在便签编辑的界面，在选项中选择”字体大小”，可以将当前便签的所有字体进行放大和缩小，其中包括4中字体大小：小、正常、大、超大。

- 功能7：进入清单模式

在便签编辑的页面，可以选择进入清单模式的选项。选择后，在便签的每一行（段内部的自动换行除外）行首出现一个勾选框，用于当前便签下标记某事项的完成情况。如果该事项已完成，则用户在勾选框中轻触，此时勾选框中将出现一个对勾，框后的陈述文字被添加了中央删除线。（外侧文件夹并不能显示事项的完成状态，可优化）

- 功能8：发送到桌面

在Android操作系统的桌面创建小米便签的小部件后，在编写便签完成后，使用选项”发送至桌面”，便可在便签小部件上显示当前便签的内容。

- 功能9：添加/删除提醒

1. 添加提醒：在便签编辑界面可以选择”添加提醒”选项，然后弹出一个对话框用于选择提醒的时间（包括月、日、星期、时、分），之后会在便签上显示一

个闹钟的图标，标志提醒时间，到了提醒时间时，操作系统便会弹出一个对话框显示便签的内容并响铃，闹钟图标标志变为“已过期”。

2. 删除提醒：在便签编辑界面可以选择“删除提醒”选项。

- 功能10：分享

在便签编辑页面上可以选择“分享”选项，之后可以将便签内容分享给GTask、QQ、微信等应用程序，其过程以纯文本格式进行。

- 功能11：识别便签文本

1. 识别邮箱：小米便签可识别文本中邮箱，点击可复制相应的邮箱

2. 识别网站：小米便签可识别文本中网站，点击可直接打开网站

3. 识别电话：小米便签可识别文本中电话，可直接复制或者呼叫

## 1.2 小米便签的软件架构以及各个包和类的作用

### 1.2.1 小米便签的软件架构

小米便签整体上是一个层次性的体系结构，由界面层、业务层、模型层和数据层 4 层构成，每层包含若干个程序包。相邻层次的程序包之间存在交互。小米便签的体系结构图见图2。

- 界面层：根据表格（在前置动作部分）中出的主要功能，我们可以很轻松地将ui和 widget填入其中，而res包含了所有的xml文件和相对应图片，属于MVC架构里面 View部分，所以也应该填入界面部分。
- 业务层：根据MVC架构的逻辑，个人判断它应该担任是C（Control控制器）的角色：处理用户输入的信息。负责从视图读取数据，控制用户输入，并向模型发送数据，是应用程序中处理用户交互的部分。负责管理与用户交互交互控制。
- 模型层：该层负责对小米便签的单个便签项进行建模，提供了便签项的基本操作功能和对应业务，并与数据层进行交互，以支持便签的创建、访问和修改。该层主要通过 model 程序包中的 Note 类、WorkingNote 类等加以实现。
- 数据层：该层负责组织和存储小米便签的相关数据,提供数据访问、数据合法性检验、数据访问缺失异常处理等功能和服务。该层主要通过 data 和 gtask.data 程序包加以实现。

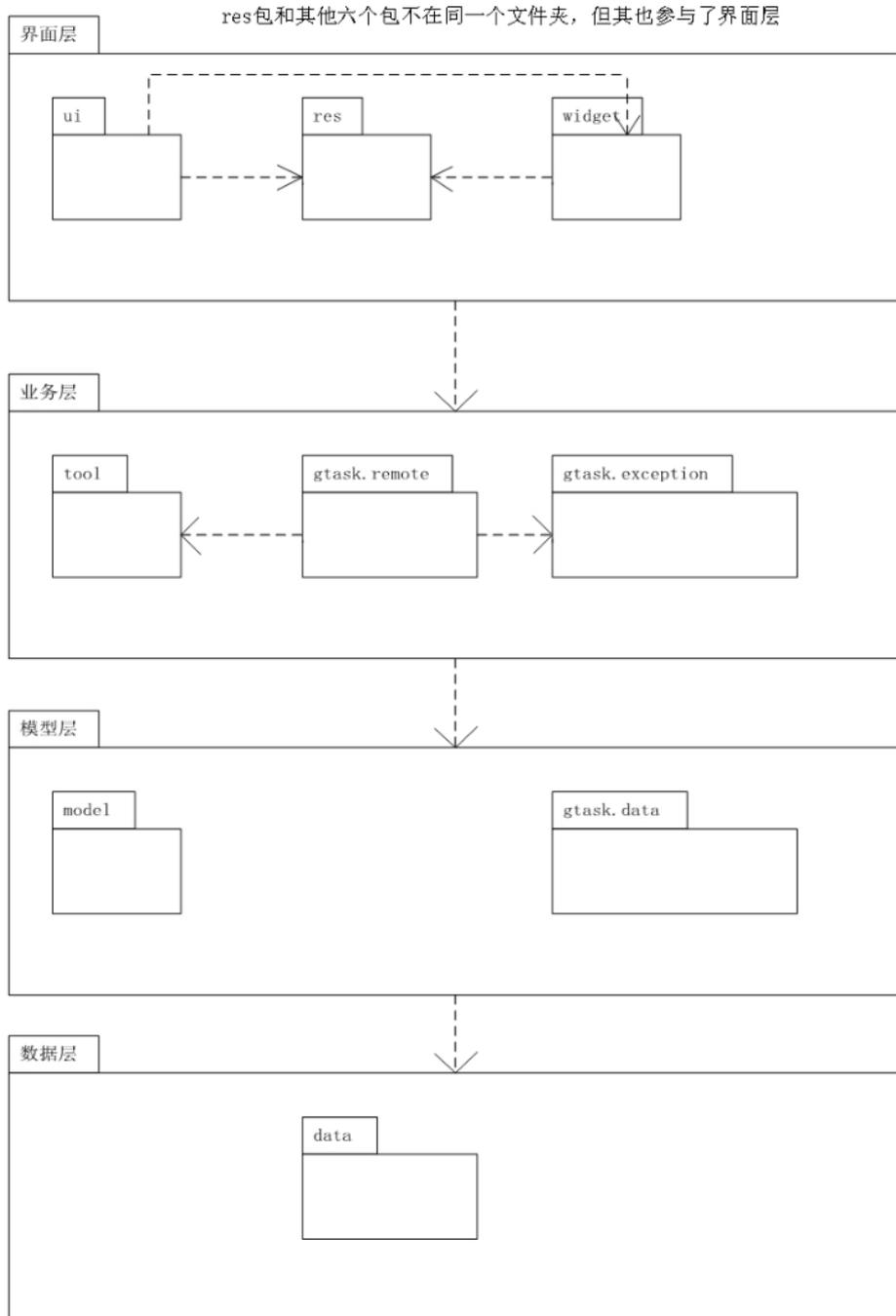


图 2: 小米便签的软件架构图

### 1.2.2 各个包内的关系图

该开源软件的类组织成7个子系统(6个源代码包，1个资源文件包)，这些子系统之间的关系如图3所示。

虚线表示包与包之间存在依赖关系。

Res包用于存放应用程序的资源文件。res是“resources”的缩写，这个包主要包含了

应用程序使用的各种资源，例如图像、布局文件、字符串、颜色等。这些资源文件在应用程序的开发和运行过程中起到重要的作用，可以提供界面展示、用户交互、美化等功能。通过使用这些资源文件，开发人员可以更方便地进行界面设计和开发。

Data包用于存储和处理备忘录应用程序的数据。通过将数据处理和管理相关的类放置在Data包中，可以使代码结构更加清晰，方便开发人员对数据进行操作和管理。

Model包用于存放数据模型相关的类，它主要负责定义应用程序中所需的数据结构、业务逻辑和数据操作的接口。通过将数据模型相关的类放置在model包中，可以使代码结构更加清晰，方便开发人员对数据进行管理和操作。

Widget包用于存放与应用程序桌面小部件（Widget）相关的类和方法。桌面小部件是一种能够在Android设备的主屏幕上显示内容的组件，它们可以提供快捷操作、实时信息展示等功能，为用户带来便利。

Tool包用于存放与工具类和辅助功能相关的类和方法。这个包通常用于处理一些通用的功能模块和工具方法，以提供开发人员更便捷的开发和实现。

Ui包用于存放与用户界面相关的类、布局和资源文件，比如闹钟提醒。通过将用户界面相关的类和资源文件放置在ui包中，可以使代码结构更加清晰，方便开发人员对界面进行管理和操作。

Gtask包用于存放与数据更新，同步，检测异常相关的类和方法。这个包通常用于实现应用程序中数据更新，网络同步功能。

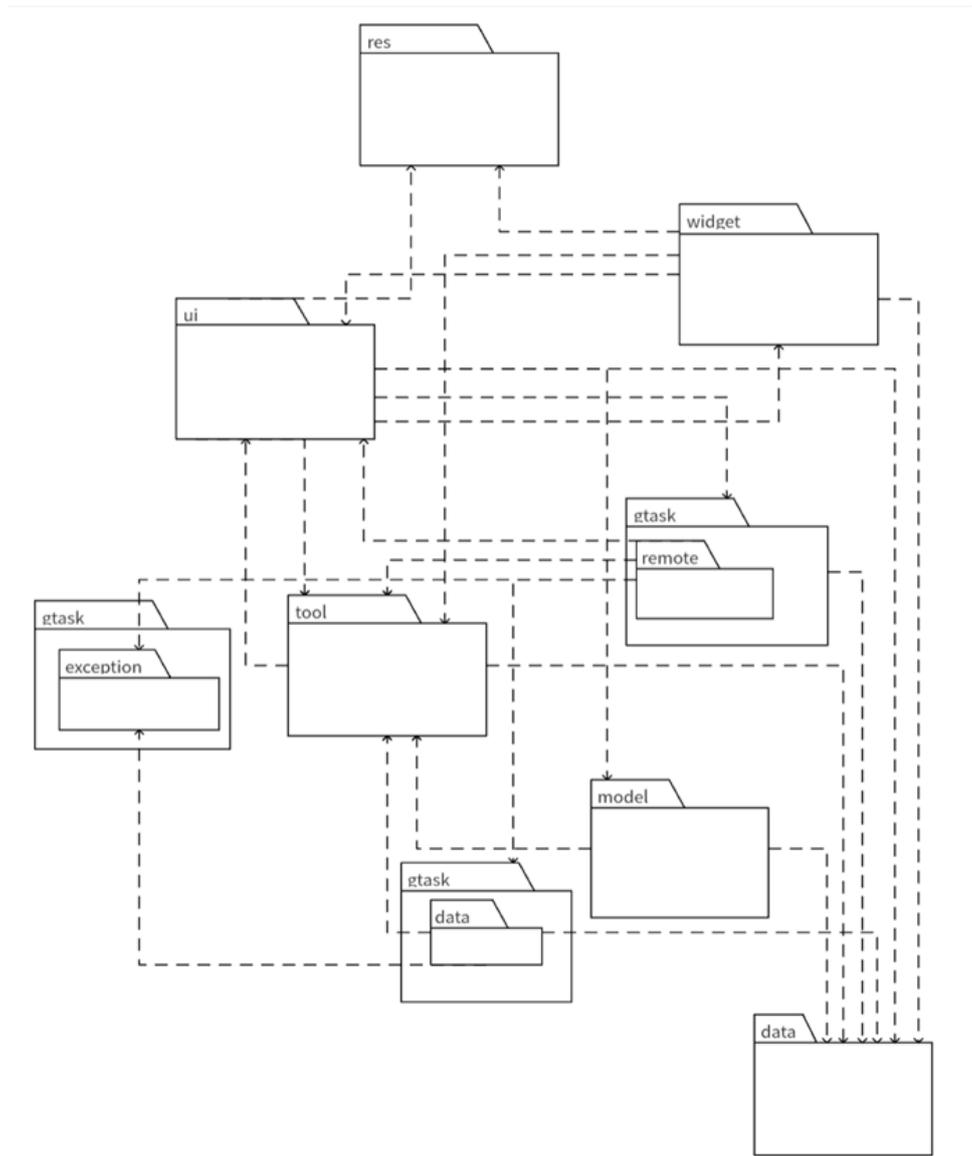


图 3: 小米便签的包间关系图

下面我将对每个类进行单独分析，以便更好地理解小米便签的代码结构。

- data包:

如图4所示，Data包当中:

Contact类独立，Notes类和NotesDatabaseHelper类关联，NotesDatabaseHelper类和NotesProvider类是聚合关系(后者为整体，前者为部分)。

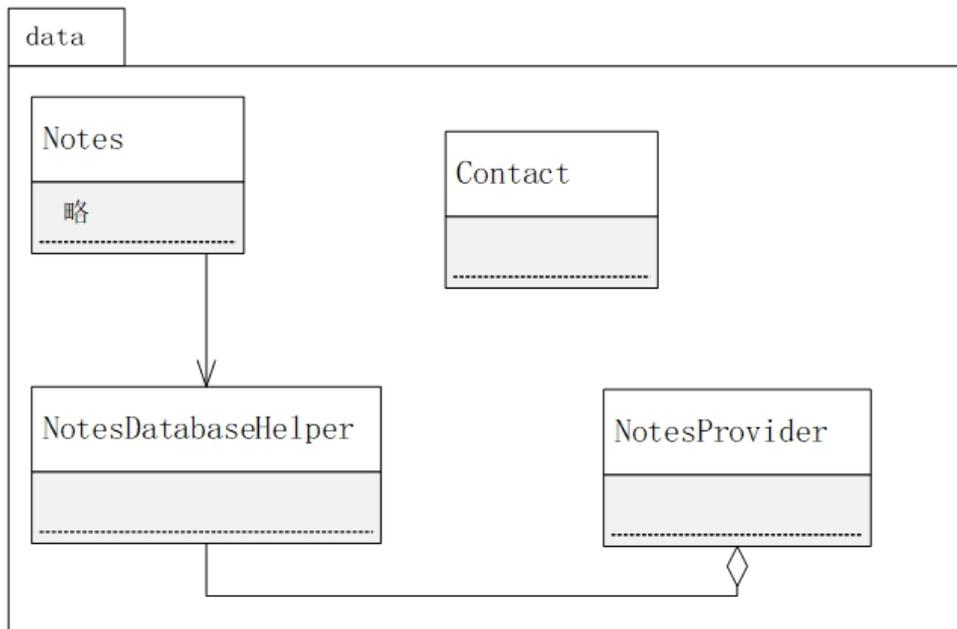


图 4: data 子系统 中的类图

- model 包:

如图5所示, Model包当中:

Note类和WorkingNote类是聚合关系(Note为部分, WorkingNote为整体)。

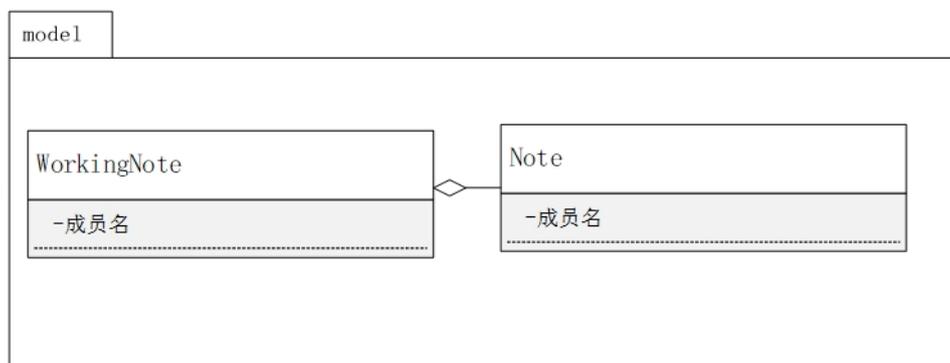


图 5: model 子系统 中的类图

- widget 包:

如图6所示, Widget包当中:

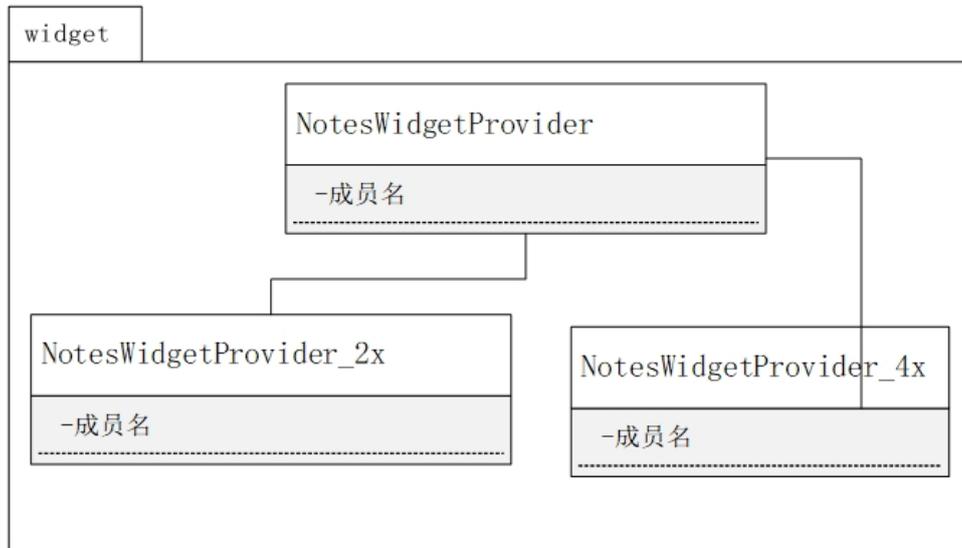


图 6: widget 子系统 中的类图

NotesWidgetProvider\_4x 类和 NotesWidgetProvider 类是泛化关系(前者是对后者的泛化或继承)。NotesWidgetProvider\_2x 类和 NotesWidgetProvider 类是泛化关系(前者是对后者的泛化或继承)。

- tool 包:

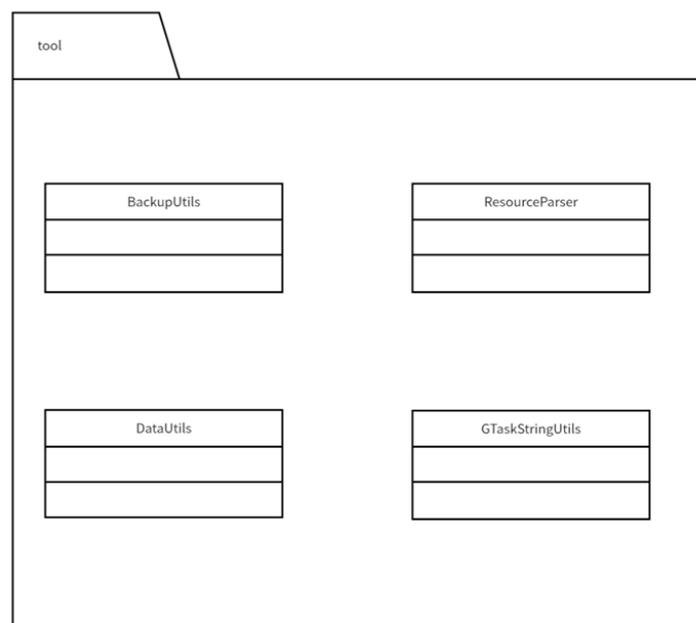


图 7: tool 子系统 中的类图

如图7所示，Tool包中：四个类独立并列，彼此之间没有明显的相关关系。

- ui包:

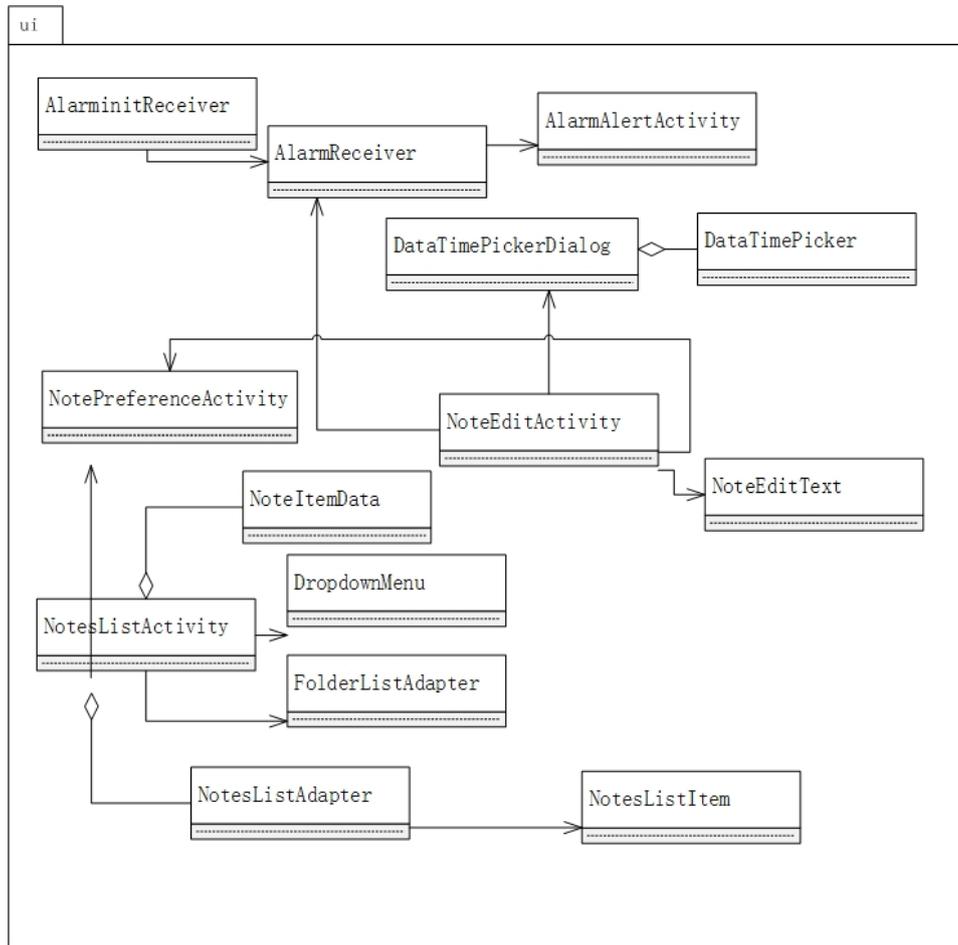


图 8: ui子系统 中的类图

如图8所示，Ui包中：

1. DateTimePicker类和DateTimePickerDialog类是聚合关系(前者为部分，后者为整体)；NoteItemData类和NotesListActivity类是聚合关系(前者为部分，后者为整体)；NotesListAdapter类和NotesListActivity类是聚合关系(前者为部分，后者为整体)。
2. AlarmInitReceiver类关联AlarmReceiver类；
3. AlarmReceiver类关联AlarmAlertActivity类；
4. NoteEditActivity类关联AlarmReceiver类，DateTimePickerDialog类NotesPreferenceActivity和NoteEditText类；
5. NoteEditActivity类和NotesListActivity类双向相互关联。
6. NoteItemData类关联NoteEditActivity类；



3. MetaData类关联SqlNote类；SqlNote类关联SqlData类；GTaskSyncService类关联GTaskASyncTask类；GTaskASyncTask类关联GTaskManager类；GTaskManager类关联GTaskClient类；
4. SqlNote类依赖ActionFailureException类；
5. SqlData类依赖ActionFailureException类；
6. TaskList类依赖ActionFailureException类；
7. Task类依赖ActionFailureException类；
8. GTaskManager类依赖MetaData类，SqlNote类，Task类，Node类和TaskList类；
9. GTaskClient类依赖ActionFailureException类，NetworkFailureException类和Node类；

### 1.3 软件功能与类间的对应关系

软件功能与类的实现对应关系见表1

表 1: 软件功能与类的实现对应关系

序号	功能名称	实现模块	主要实现方法
1	新建便签	NoteEditActivity	createNewNote()
2	删除便签	NoteEditActivity	onOptionsItemSelected()
3	移动便签	DataUtils	moveNoteFolder() batchMoveToFolder()
4	新建文件夹	NoteListActivity	showCreateOrModifyFolderDialog()
5	查看文件夹	NoteListActivity	openFolder()
6	删除文件夹	NoteListActivity	deleteFolder()
7	修改文件夹名称	NoteListActivity	showCreateOrModifyFolderDialog()
8	导出文本	NoteListActivity	exporNoteToText()
9	同步	GTaskSyncService	startSync()
10	搜索	NoteListActivity	onSearchRequest()
11	修改字体大小	NoteEditActivity	onOptionsItemSelected()
12	修改便签背景颜色	WorkingNote	setBgColorId()

下页继续

续表 1：软件功能与类的实现对应关系

序号	功能名称	实现模块	实现方法
13	进入清单模式	WorkingNote	setCheckListMode()
14	发送到桌面	NoteEditActivity	sendToDestop()
15	添加提醒	NoteEditActivity	setAlertData()
16	删除提醒	WorkingNote	setAlertData()
17	识别电话号码	Contact	getContact()
18	分享	NoteEditActivity	getWorkingText(),sendTo()
19	保存便签	WorkingNote	saveNote()

这部分的工作主要是通过增加断点进行代码调试工作才确认类对应的功能。主要涉及到的功能界面有两个，对应的类为NotesListActivity和NoteEditActivity。

像某些功能对应两种实现方式的需要特别注意。例如：新建便签和删除便签，在主界面和便签菜单栏都可以实现。因此调试的方法才更有必要，只有通过断点调试才能确定其中的正确的对应关系。

另外还有一些功能，在AS虚拟机上并不能实现。例如：导出文本功能，该功能需要手机装备SD卡；又例如同步功能和分享功能，该功能需要进行联网并备有Google等平台的账号，用来进一步的操作。

有一些功能无法进行断点调试。例如：识别电话号码、邮箱功能，在便签内输入类似电话、邮箱的文字后，会在文字产生下划线，并附有超链接，可以进行点击操作，但在调试过程中并没有出现中断现象。

## 2 小米便签的代码标注

小组对6个包中共39个类的代码进行了标注，标注的代码行数共有2350行。具体的，这些代码注释的分部如表2所示。

表 2: 代码标注分布情况

序号	包名称	类名称	标注的代码行数
1	data	Contact	20行。在该类中分布在类的总体介绍(2,3); 引入库文件的说明(8,9); 类的静态变量说明(23,32-36); 方法中关键字(39,44,49)
2	data	Notes	76行。在该类中分布在静态变量说明(23到49为int类型,之后为final string类型)
3	data	NotesDatabaseHelper	72行。在该类中分布在静态成员变量(18,21,25...)和对各种具体格式(SQL语句和uri接口)中
4	data	NotesProvider	47行。对各种变量、数据的sql语句进行了标注
5	gtask.data	MetaData	25行。在该类中分布在对元数据的定义、获取、传输、同步等方法中
6	gtask.data	Node	31行。对节点同步动作的常量、JSON动作、设置节点的相关内容方法进行了标注
7	gtask.data	SqlData	38行。对数据库管理类sqldata进行了说明,包括Cursor操作,属性说明,标签定义,以及重要方法sqlData构造函数的标注
8	gtask.data	SqlNote	88行。对静态变量进行了标注,对重要方法如获取、更新、记录变迁属性的功能进行了标注
9	gtask.data	Task	59行。对任务类task的成员变量tag等进行了标注,对JSON等方法进行了标注
10	gtask.data	TaskList	83行。任务列表类,继承自Node类。用于管理一组任务(Task)对象。对task任务列表的静态常量如标签、索引列表等静态常量做了标注。定义了类的构造函数,更新列表的重要方法的标注
11	gtask.exception	ActionFailureException	12行。该异常类是运行时异常的子类,用于表示操作失败的异常情况。它可以包含一个错误消息和导致异常的Throwable对象。解释了抛出异常的具体信息
12	gtask.exception	NetworkFailureException	6行。主要对网络操作失败异常类作用进行了注释,对构造函数进行了注释
13	gtask.remote	GTaskASyncTask	27行。对静态常量如同步通知栏ID等进行了标注,对重要方法如构造函数、发布、显示更新进度的方法等进行了标注

下页继续

续表 2: 代码标注分布情况

序号	包名称	类名称	标注的代码行数
14	gtask.remote	GTaskClient	172行。接下来的注释说明了GTaskClient类的作用,它用于与Google任务服务进行远程交互,并提供了登录、获取任务列表、添加任务等操作的方法。注释还解释了类的各种属性,如日志标签、基础URL、请求URL、单例模式实例、HTTP客户端、版本号、登录状态和操作ID等。对于GTaskClient的私有构造方法,注释说明了其用途是初始化各种属性。提供了getGTaskClient方法来获取GTaskClient的单例实例,并确保只创建一个实例
15	gtask.remote	GTaskManager	221行。说明了GTaskManager类的用途,包括日志标签、任务状态常量(如成功、网络错误、内部错误、同步进行中和同步已取消)。注释描述了类的各种属性,例如单例实例、关联的Activity对象、上下文对象、内容解析器、同步状态标记、任务列表和任务的HashMap、元数据的HashMap等
16	gtask.remote	GTaskSyncService	62行。注释描述了类中定义的同步操作类型常量,加后台同步、取消同步和无效操作。提供了服务广播的名称、标志以及广播的同步进度消息
17	model	Note	56行。描述了为数据库中添加新笔记生成一个新的笔记ID,创建一个新的笔记等方法

下页继续

续表 2: 代码标注分布情况

序号	包名称	类名称	标注的代码行数
18	model	WorkingNote	230行。描述了类中的各种属性,如笔记对象、笔记的唯一标识符、笔记的内容、笔记的模式、提醒日期的时间戳、笔记最后修改日期的时间戳、笔记背景颜色的资源ID、小部件的ID和小部件的类型等。描述了构造函数、设置提醒日期、标记笔记为已删除、设置笔记背景颜色ID、等方法
19	tool	BackupUtils	120行。主要对备份工具类和内部类TextExport等进行了说明
20	tool	DataUtils	134行。描述了批量删除笔记的方法。描述了将笔记移动到指定文件夹的方法。描述了批量将笔记移动到指定文件夹的方法
21	tool	GtaskStringUtils	34行。描述了批量删除笔记的方法、将笔记移动到指定文件夹的方法、批量将笔记移动到指定文件夹的方法。获取删除系统文件夹外的所有用户文件夹数量的方法。描述了检查指定类型的笔记在数据库中是否可见的方法等等方法
22	tool	ResourceParser	72行。描述了定义笔记背景颜色的常量,获取默认笔记背景id、笔记背景资源头、提供获取不同背景资源的方法
23	ui	AlarmAlertActivity	40行。提醒界面
24	ui	AlarmInitReceiver	21行。后台消息接收器
25	ui	AlarmReceiver	14行。间接提醒接收器
26	ui	DateTimePicker	130行。设置提醒时间的部件
27	ui	DateTimePickerDialog	32行。设置提醒时间的对话提示界面
28	ui	DropDownMenu	26行。下拉菜单界面

下页继续

续表 2: 代码标注分布情况

序号	包名称	类名称	标注的代码行数
29	ui	FoldersListAdapter	32行。文件夹列表链接器(链接数据库)
30	ui	NoteEditActivity	296行。便签编辑活动
31	ui	NoteEditText	64行。便签的文本编辑界面
32	ui	NoteItemData	32行。便签项数据
33	ui	NotesListActivity	主界面, 实现处理文件夹列表的活动
34	ui	NotesListAdapter	便签列表链接器(链接数据库)
35	ui	NotesListItem	便签列表项
36	ui	NotesPreferenceActivity	便签同步的设置界面
37	widget	NoteWidgetProvider	桌面挂件
38	widget	NoteWidgetProvider_2x	12行。NoteWidgetProvider, 负责处理2x大小的小部件更新和其他操作
39	widget	NoteWidgetProvider_4x	12, 4倍大小挂件
合计	6个包	39个类	大约3924行左右代码标注

### 3 小米便签的代码维护

#### 3.1 维护的内容

对小米便签进行了如表3所示的维护, 包括增加新的功能等等。

表 3: 维护内容列表

序号	维护类别	名称	描述
1	新增功能	设置界面背景	修改小米便签初始背景, 并可以进行切换操作。
2	新增功能	欢迎界面	打开小米便签时, 会有短暂的欢迎界面, 2s 后界面自动消失, 此时进入小米便签。
3	新增功能	登录密码	登录界面, 在进入主界面时需要输入密码, 设置了一个开放的密码锁保护隐私。可以进行密码的增添、删除和替换操作。
4	新增功能	翻译功能	在学习生活中, 我们会经常使用英汉互译功能, 将英汉互译功能添加到小米便签里面, 可以方便我们在特定条件下的使用。

下页继续

续表 3: 维护内容列表

序号	维护类别	名称	描述
5	新增功能	插入图片	现在主流便签都有插入图片功能, 这方便用户记录图片信息。
6	新增功能	统计字符个数	在主流编辑软件里面, 可以统计带空格的字符数, 这方便用户知道自己写作的篇幅。
7	新增功能	富文本功能	在主流的便签类软件都有富文本功能, 可以方便用户对文本中的内容进行加粗、斜体、删除线以及高亮等。
8	新增功能	朗读功能	在学习生活中, 我们会经常使用英汉互译功能, 将英汉互译功能添加到小米便签里面, 可以方便我们在特定条件下的使用。
9	新增功能	私密模式	在便签的初始界面会显示便签的第一行内容, 这样一定程度上容易泄露个人的隐私, 进入私密模式后可以设定第一行内容, 这样可以起到一定的保密作用。
10	新增功能	笔记编辑内搜索	在笔记编辑界面, 可以进行搜索操作, 如果没有搜索到内容会显示未找到相关内容, 如果搜索到内容会高亮所有搜索到的内容。
11	新增功能	模板便签	每个用户都有特定的笔记编辑习惯, 提前输入好模板, 可以便捷用户记录便签。
12	新增功能	语音听写	在学习生活中, 语音输入可以方便用户快速输入内容。
13	新增功能	语音合成	基于讯飞星火AI的语音合成功能, 支持多语种、多声色和智能调控音调、节奏等丰富功能。
14	新增功能	对话式大模型	如今市面上主流的软件都内置了大模型, 可以方便用户基于便签文本的上下文进行对话, 获取帮助。
15	新增功能	撤回功能	用户在对便签进行编辑时, 往往可能会误触删除一部分文本, 撤回功能可以撤销误删除的操作。
16	新增功能	获取地理位置	一些软件需要获取地理位置权限才可以正常运行, 为小米便签植入获取地理位置的功能, 可以帮助用户及时定位自己的位置。

## 3.2 开源软件维护后所产生的设计

### 3.2.1 架构设计

维护后的小米便签有如下架构设计：

### 3.2.2 界面设计

维护后的小米便签有如下界面设计：

### 3.2.3 详细设计

## 3.3 维护代码数量以及质量情况

### 3.3.1 NotesListActivity类

新增了set\_login\_password()方法跳转到设置密码登陆界面，change\_password()方法跳转到修改密码登陆界面，delete\_password()方法跳转到删除密码登陆界面。在 onOptionsItemSelected()方法中添加进入私密模式和退出私密模式的触发。

### 3.3.2 NoteEditActivity类

在 onOptionsItemSelected()方法中新增了撤回、设置字体的触发。新增了count()方法、operateText()方法、cutOfimage()方法实现了实时统计字符数功能，新增了texttoSpeech()方法、speak ()方法实现了朗读功能，新增了 convertToImage()方法、onActivityResult()方法、getPath()方法、getDataColumnn()方法、isMediaDocument()方法实现了插入图片功能,新增了showSingleAlertDiglog()方法实现了设置字体功能，新增了doTranslate()方法、translate\_u2z()、translate\_z2u()方法实现了翻译功能，新增了 doRevoke()方法实现了撤销功能,新增了local\_sel()、getLocation()方法实现了获取地理位置功能。

### 3.3.3 维护后的小米便签主要增加的包和类

1. 新增 translate\_demo 包用于实现翻译功能,新增speech包用于规范数据格式和存储、组装语音和文本数据。
2. 在 ui 包中：
  - 新增 ChangePassword 类：实现了修改登陆密码功能。
  - 新增 DeletePassword 类：实现了删除登陆密码功能。
  - 新增 LoginActivity 类：实现了使用登陆密码登陆功能。
  - 新增 CreatePassword 类：实现了设置登陆密码功能。
  - 新增 NoteEditor 类：实现笔记编辑页面中搜索功能。
  - 新增 SetPassword 类：实现了设置登录密码的功能。
  - 新增 SplashActivity 类：实现了登陆欢迎界面功能。
  - 新增 IATActivity 类：实现了转向语音合成与听写界面的功能。
  - 新增 IatDemo 类：实现了语音转写功能。
  - 新增 TtsDemo 类：实现了语音合成功能。
  - 新增 IflytexActivity 类：实现了与大模型交互的功能。

3. 在translate\_demo包中:

- 新增 BaiduTranslateService 类: 实现百度翻译交互。
- 新增 MD5Utils 类: 实现百度翻译发送 JSON 文件时的 MD5 加密。
- 新增 RespondBean 类: 实现百度翻译主要功能。

4. 在speech包中:

- 新增 settings 包: 规定Iat与Tts等功能输入数据和输出数据的格式。
- 新增 util 包: 包含Json结果解析类, 功能性函数扩展类等工具类, 用于传递语音交互时的数据。

由图10, 我们可以发现: 和之前的包图进行对比, 可以发现最大的改动是ui包中新增translate\_demo子包, 用来实现翻译功能;新增了speech包, 用来实现语音合成与转写功能。

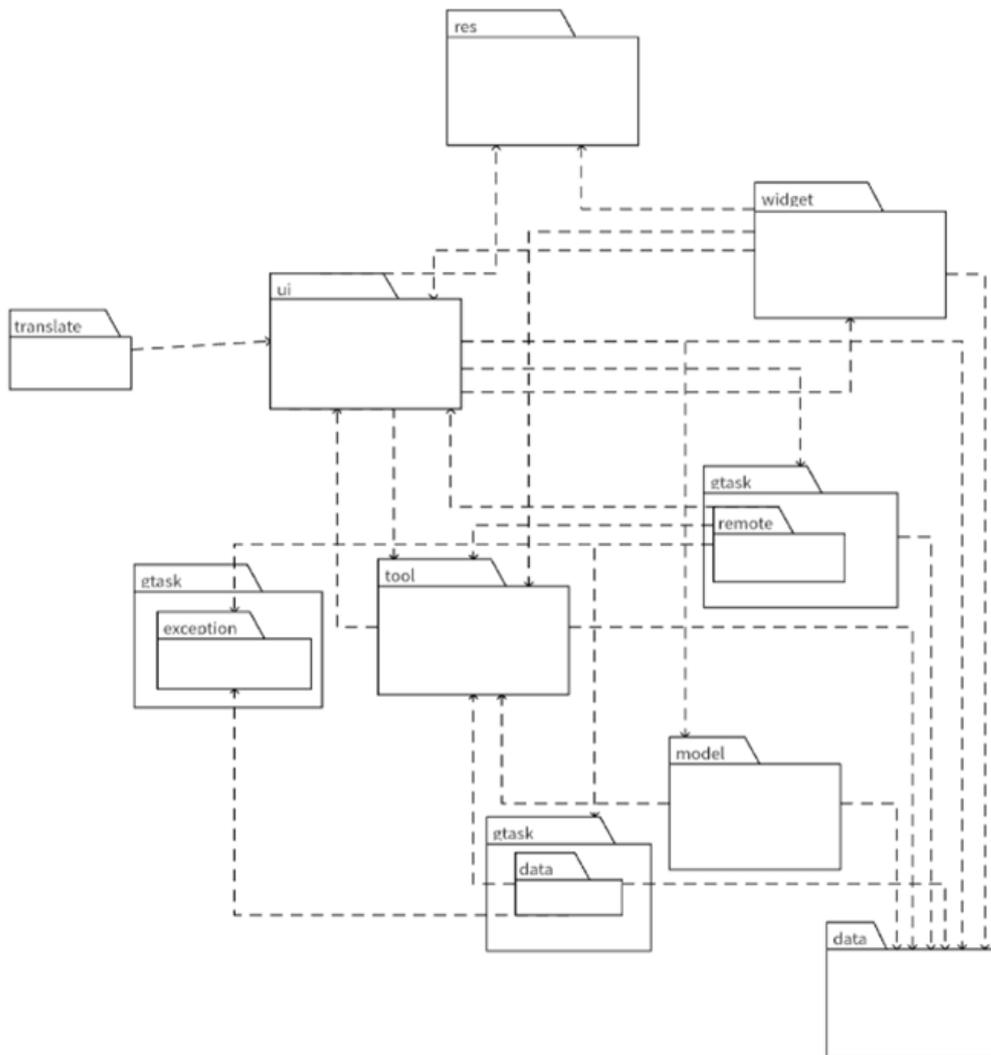


图 10: 维护后的小米便签包图

由图11，我们可以发现：和维护之前的ui包内的类图进行对比，维护后主要在新增了新增 ChangePassword类、DeletePassword 类和CreatePassword 类来实现修改、删除和新增密码的功能；新增LoginActivity 类、SplashActivity 类来实现了登陆欢迎界面功能。新增NoteEditor类，实现了笔记编辑页面中搜索功能。新增SetPassword类，实现了设置登录密码的功能。新增DeletePassword类，实现了删除登陆密码功能

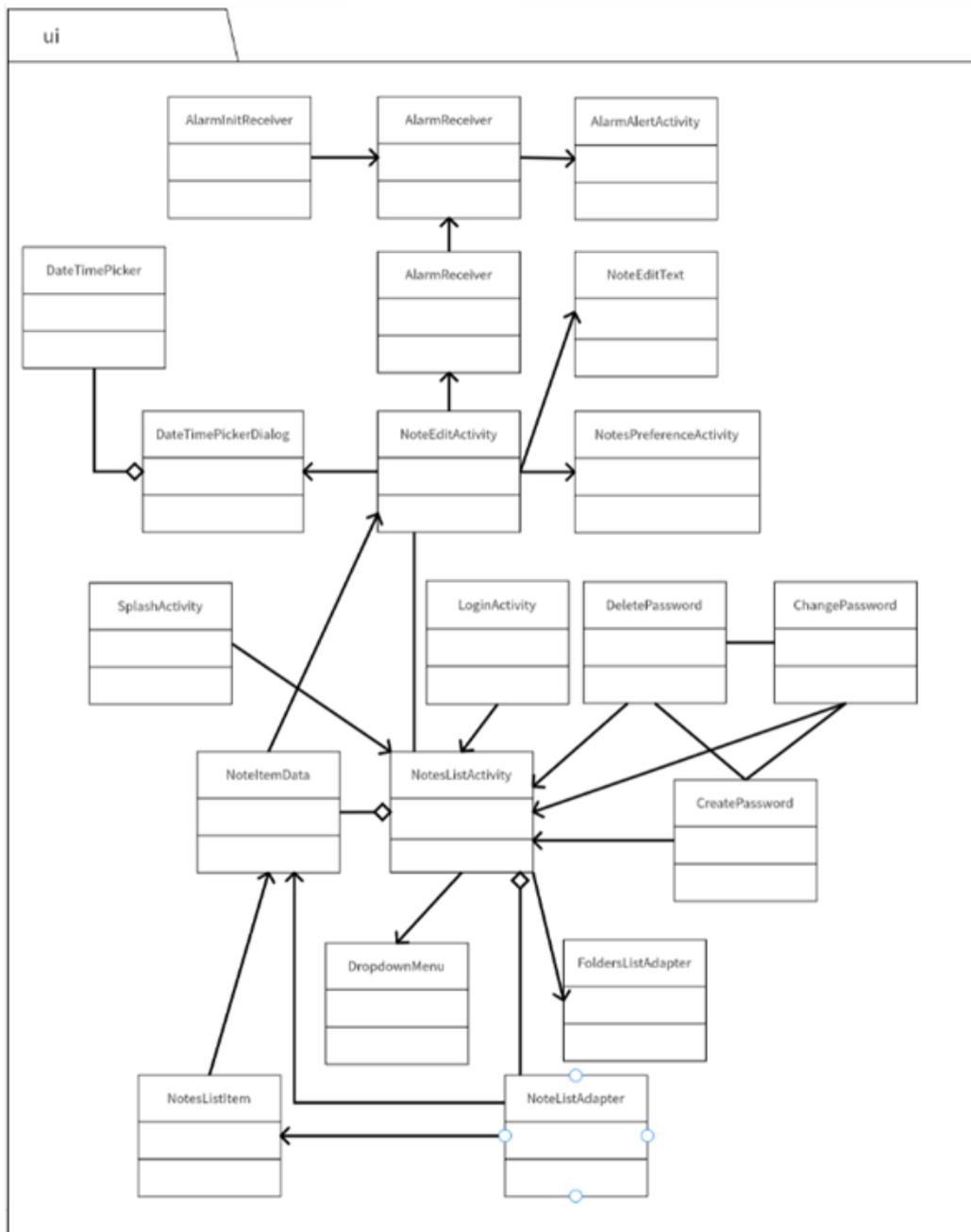


图 11: 维护后ui包内的类图

### 3.3.4 维护后的质量分析

在代码维护之后，我们小组再次对代码进行了质量分析，使用CodeArts中的代码审查功能对代码的质量进行分析。分析结果见图44



图 12: 代码审查结果

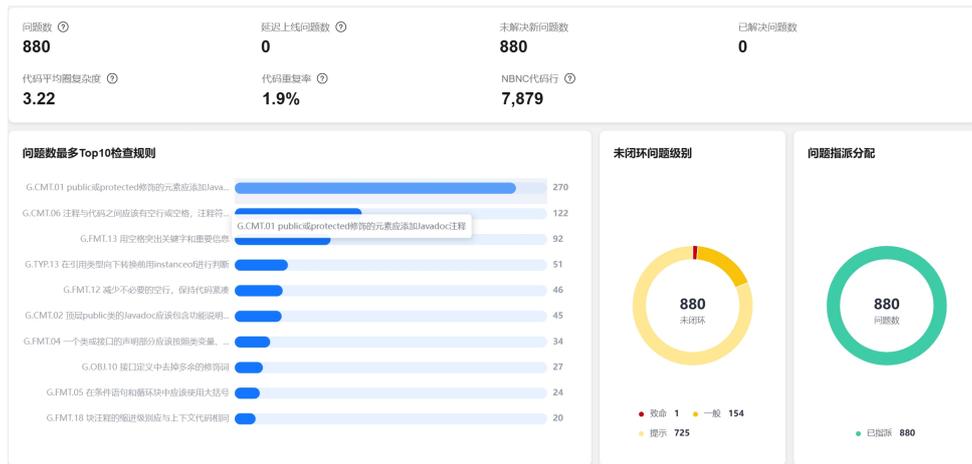


图 13: 代码审查结果

- 总体情况

1. 未解决问题数: 880
2. 已解决问题数: 0
3. 代码行数: 7,879
4. 代码重复率: 1.9%
5. 代码平均复杂度: 3.22

- 问题分析

1. 问题数量: 880个未解决的问题是一个相对较高的数字, 表明代码中存在较多的潜在问题, 可能影响代码的可维护性和可读性。
2. 问题类型:
  - G.CMT.01 (270个): 涉及public或protected修饰符的元素未添加JavaDoc注释。这表明代码文档化不足, 可能导致后续维护和理解困难。
  - G.CMT.06 (122个): 注释与代码之间存在不一致性, 可能导致误解。
3. 代码复杂度: 平均复杂度为3.22, 虽然不算高, 但仍需关注复杂度较高的部分, 可能需要重构以提高可读性和可维护性。
4. 代码重复率: 1.9%的重复率相对较低, 表明代码的重复使用情况良好, 但仍需定期检查以避免潜在的重复代码问题。

- 详细的分析结果汇总见表45, 如下所示:

表 4: 代码质量分析详细结果

规则编号	违规数量	规则说明
G.CMT.02	45	顶层public类的Javadoc应该包含功能说明和创建日期/版本信息
G.FMT.05	24	在条件语句和循环块中应该使用大括号
G.FMT.18	20	快注释的缩进级别应与上下文代码相同
G.EXP.04	7	表达式的比较, 应该遵循左侧倾向于变化、右侧倾向于不变的原则-使用equals方法进行字符串比较
G.FMT.12	46	减少不必要的空行, 保持代码紧凑
G.OBJ.10	27	接口定义中去掉多余的修饰词
G.FMT.04	34	一个类或接口的声明部分应该按照类变量、静态初始化块、实例变量、构造器、方法的顺序出现, 且用空行分隔
G.CMT.03	3	方法的Javadoc中应该包含功能说明, 根据实际需要按顺序使用@param、@return、@throws标签对参数、返回值、异常进行注释
G.EXP.04	4	表达式的比较, 应该遵循左侧倾向于变化、右侧倾向于不变的原则-表达式比较左变右不变
G.OBJ.08	1	正确实现单例模式
G.FMT.20	3	数字字面量应该设置合适的后缀, long类型应该使用L作为后缀
G.DCL.03	5	禁止C风格的数组声明
G.FMT.06	11	对于非空块结构, 左大括号应该放在行尾, 右大括号应该另起一行
G.OBJ.07	13	子类覆写父类方法或实现接口时必须加上@Override注解
G.ERR.03	6	不要直接捕获可通过预检查进行处理的RuntimeException, 如NullPointerException、IndexOutOfBoundsException等
G.FMT.08	6	使用空格进行缩进, 每次缩进4个空格
G.CTL.01	14	不要在控制性条件表达式中执行赋值操作或执行复杂的条件判断
G.OTH.01	1	安全场景下必须使用密码学意义上的安全随机数
G.NAM.04	5	方法名应采用小驼峰命名
G.CMT.06	122	注释与代码之间应该有空行或空格, 注释符与注释内容之间应该有空格
G.CMT.01	270	public或protected修饰的元素应添加Javadoc注释
G.TYP.13	51	在引用类型向下转换前用instanceof进行判断
G.OTH.03	15	不用的代码段包括import, 直接删除, 不要注释掉-不用的代码段, 直接删除, 不要注释掉
G.FMT.10	8	行宽不超过120个字符
G.NAM.03	2	类、枚举和接口名应采用大驼峰命名
G.FMT.07	1	应该避免空块, 必须使用空块时, 应采用统一的大括号换行风格

下页继续

续表 4: 代码质量分析详细结果

规则编号	违规数量	规则说明
G.CMT.03	2	方法的Javadoc中应该包含功能说明, 根据实际需要按顺序使用@param、@return、@throws标签对参数、返回值、异常进行注释-功能描述和标签之间要有一个空行
G.MET.01	15	方法要简短-方法的代码块深度不应超过4层
G.OTH.03	9	不用的代码段包括import, 直接删除, 不要注释掉-不用的import语句, 直接删除, 不要注释掉
G.MET.01	13	方法要简短-方法行数不应超过50行
G.MET.01	1	方法要简短-方法的参数不应超过5个
G.NAM.05	1	常量名采用全大写单词, 单词间以下划线分隔
G.CON.12	3	避免不加控制地创建新线程, 应该使用线程池来管控资源
G.FMT.13	92	用空格突出关键字和重要信息
总计	880	

### 3.4 维护后的软件原型以及功能展示

#### 3.4.1 设置界面背景

修改了小米便签的初始背景，更具有美观性，让用户有良好的使用体验。同时，可以对背景进行切换操作，总共设计了2款不同背景。（见图12）

设置界面背景的时序图如图16所示。



图 14: 设置背景1

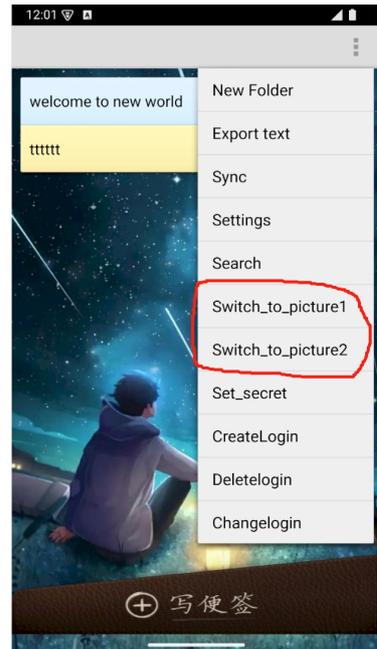


图 15: 设置背景按钮



图 16: 设置背景2

### 3.4.2 欢迎界面

实现进入便签前两秒钟的欢迎界面，让用户有良好的使用体验。（见图13）

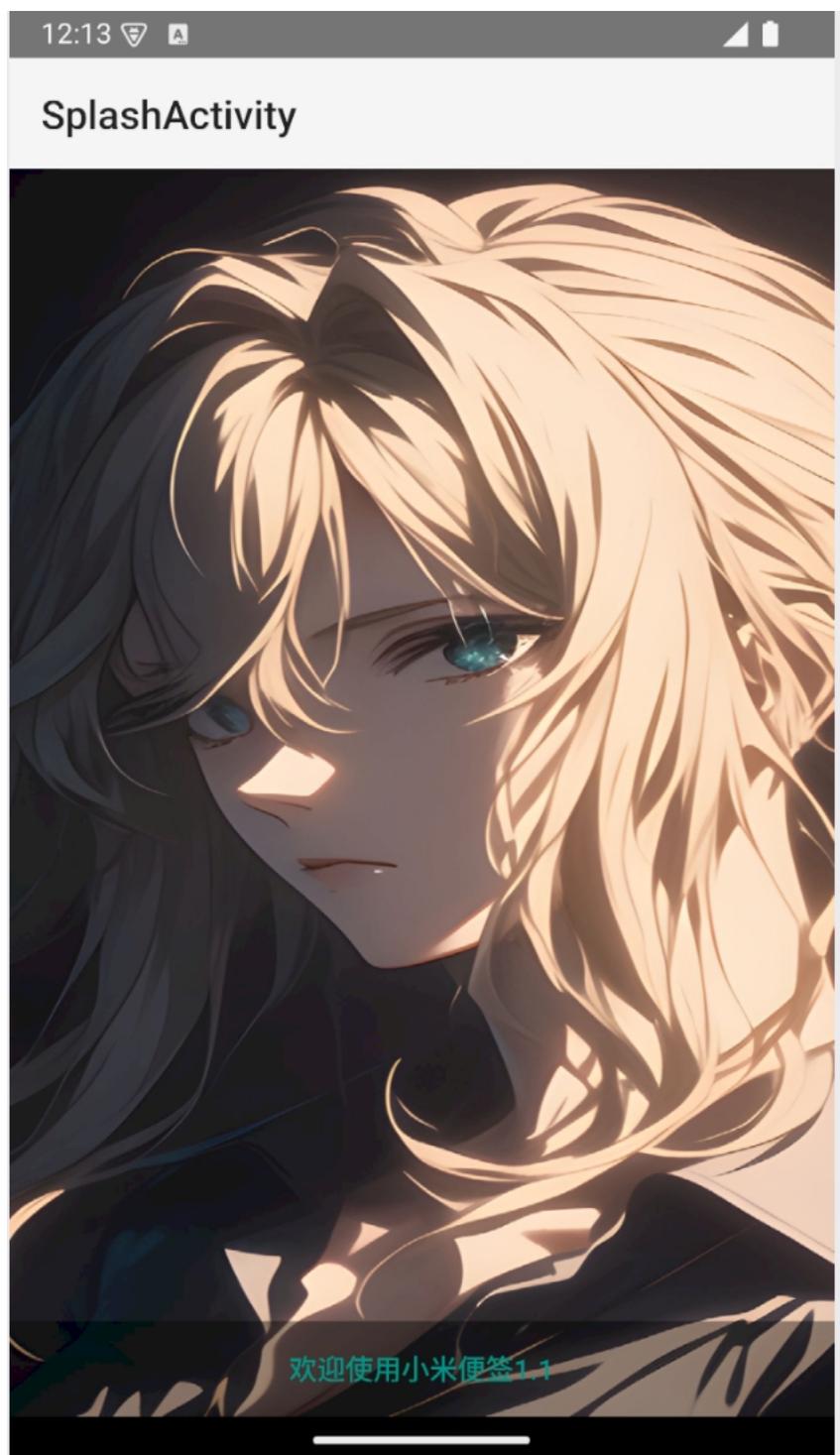


图 17: 欢迎界面

### 3.4.3 登录密码

在主界面菜单中，有新建登陆密码按钮、删除登陆密码按钮和修改登录密码按钮。点击新建登陆密码会出来新建登录密码界面，里面有输入密码和确认密码，如果输入密码和确认密码不一致则要求用户重新输入。（见图14）

这一功能的时序图如图16所示。

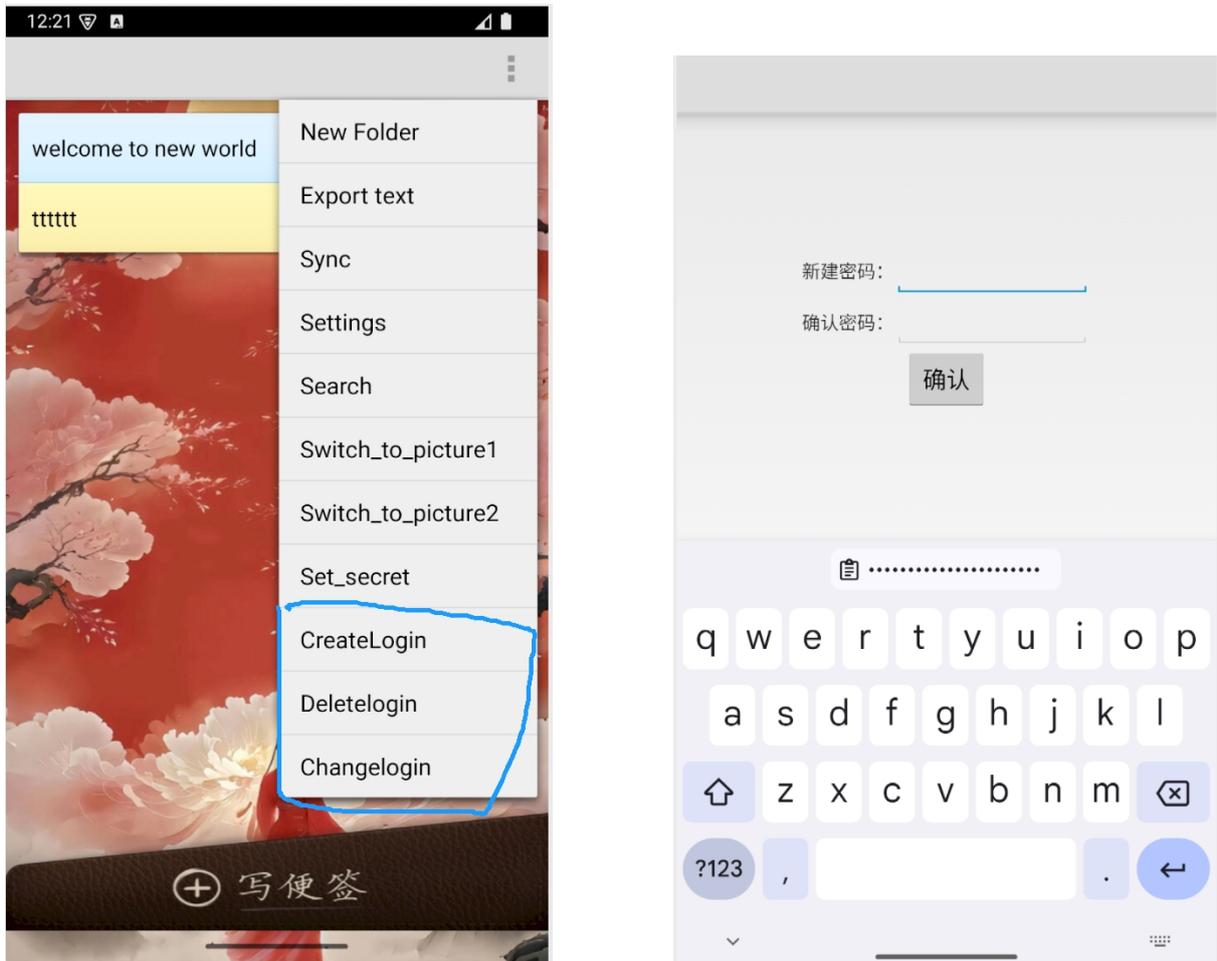


图 18: 新增设置密码的操作

我们新建密码为“111”，再次进行登录操作可以发现，必须要输入密码才能进入应用。输入密码之后可以成功进入。（见图15）



图 19: 新增密码后的登录界面，要求输入密码

如图16所示，我们重新设置密码的时候，需要我们输入现有的密码，然后才能设置新的密码，输入新的密码“123”，确认密码，密码进行更换。



图 20: 重新设置密码

再次登录的时候，原本的密码“111”已经不再适用，输入现有密码“123”才能够成功进入。（见图17）

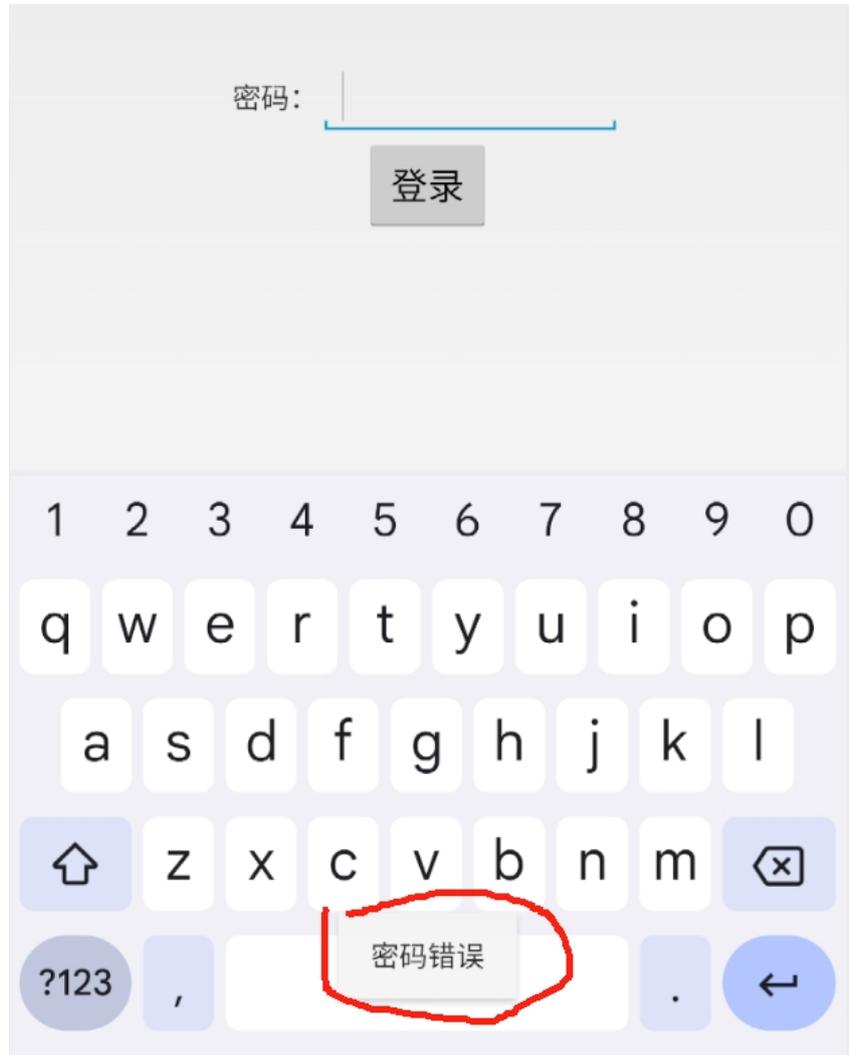
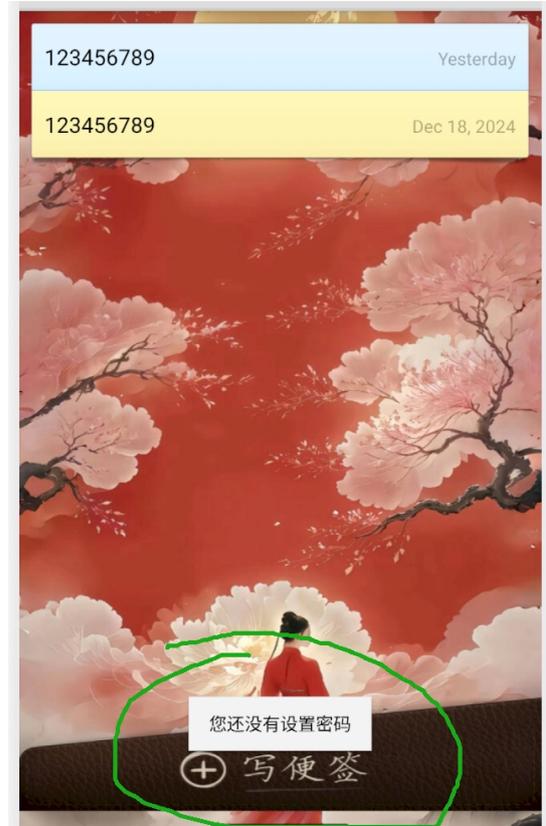


图 21: 使用原有的密码登录失败

同样的，要想删除设置的密码，也要先输入原有的密码。（如图18）删除之后，再次点击删除密码的按钮会提醒“没有设置密码”。



(a) 删除密码



(b) 删除密码之后显示没有设置密码

图 22: 删除设置的密码

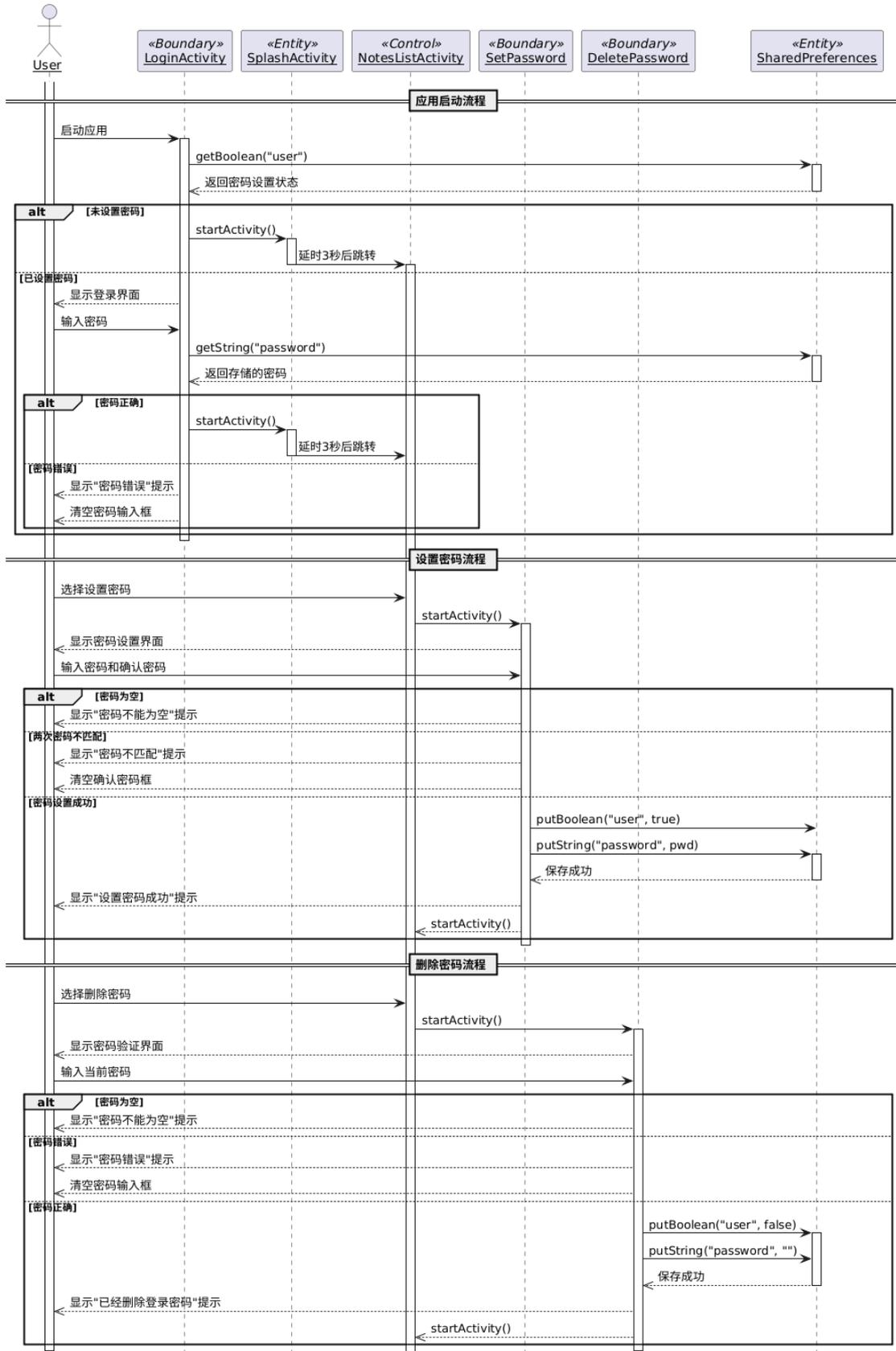


图 23: 登录密码的时序图

### 3.4.4 翻译功能

在便签编辑界面的上方，有个翻译按钮，点开之后有三个按钮可以选择：中文翻译为英文、英文翻译为中文、还原，其中还原可以将文本还原为翻译之前的内容。这一功能的时序图如图19所示。

如图19所示，可以将便签内的内容由英文转化为中文。例：“Welcome to the new world.”可以翻译为“欢迎来到新世界。”

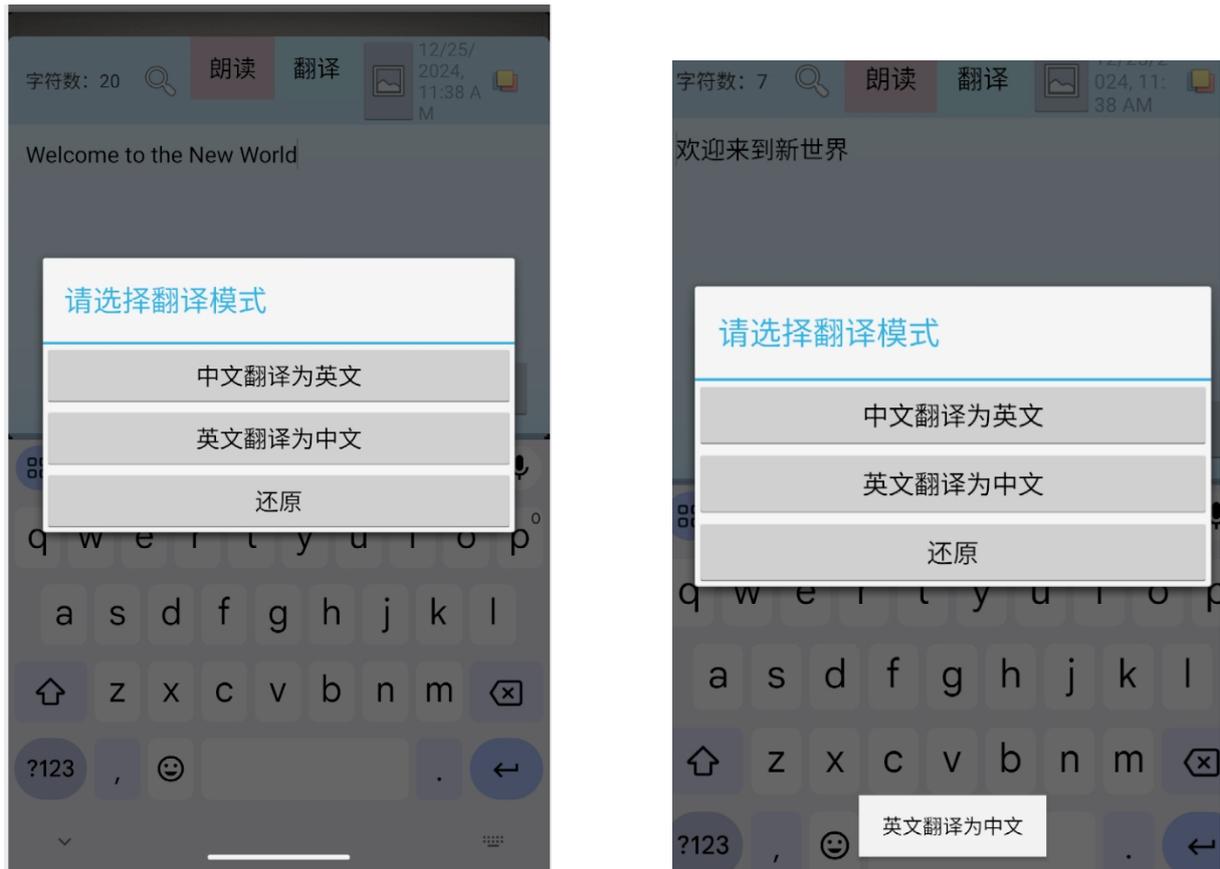


图 24: 翻译功能中的将英文翻译为中文

与此同时，我们也可以将便签内的内容由中文转化为英文。（见图20）例：“欢迎来到新世界。”可以翻译为“Welcome to the new world.”



图 25: 翻译功能中将中文转化为英文

另外，我们在翻译功能中还实现了，将翻译的内容进行还原，可以将翻译之后的内容再还原回去。（见图21）



图 26: 翻译功能中的还原操作

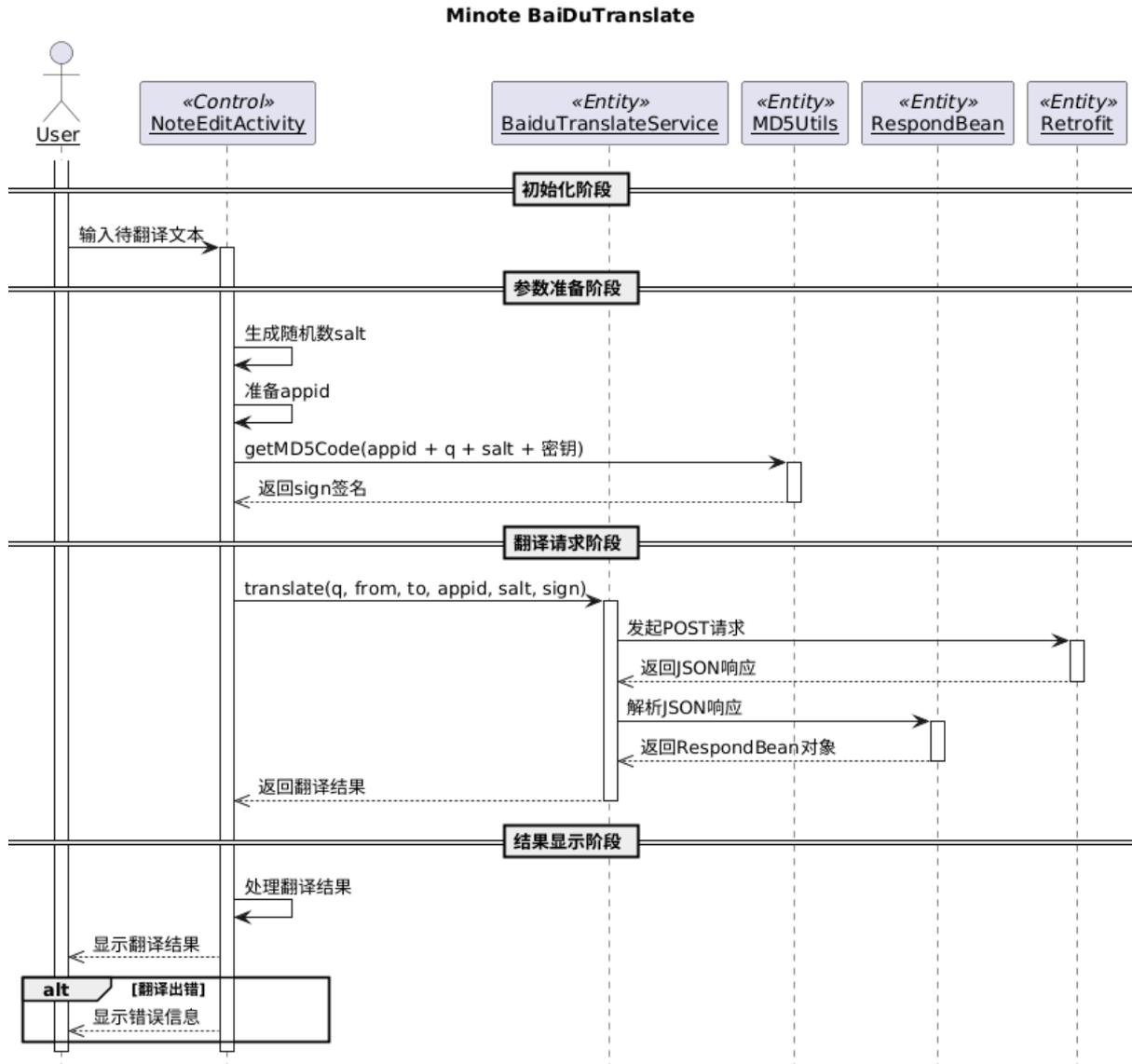


图 27: 翻译功能的时序图

### 3.4.5 插入图片

在便签编辑界面增加插入图片按钮，点击后跳转到相册，供用户选择插入图片，用户选好图片后将图片的路径插入到文本中，在将文本中的路径显示成图片。（见图22）

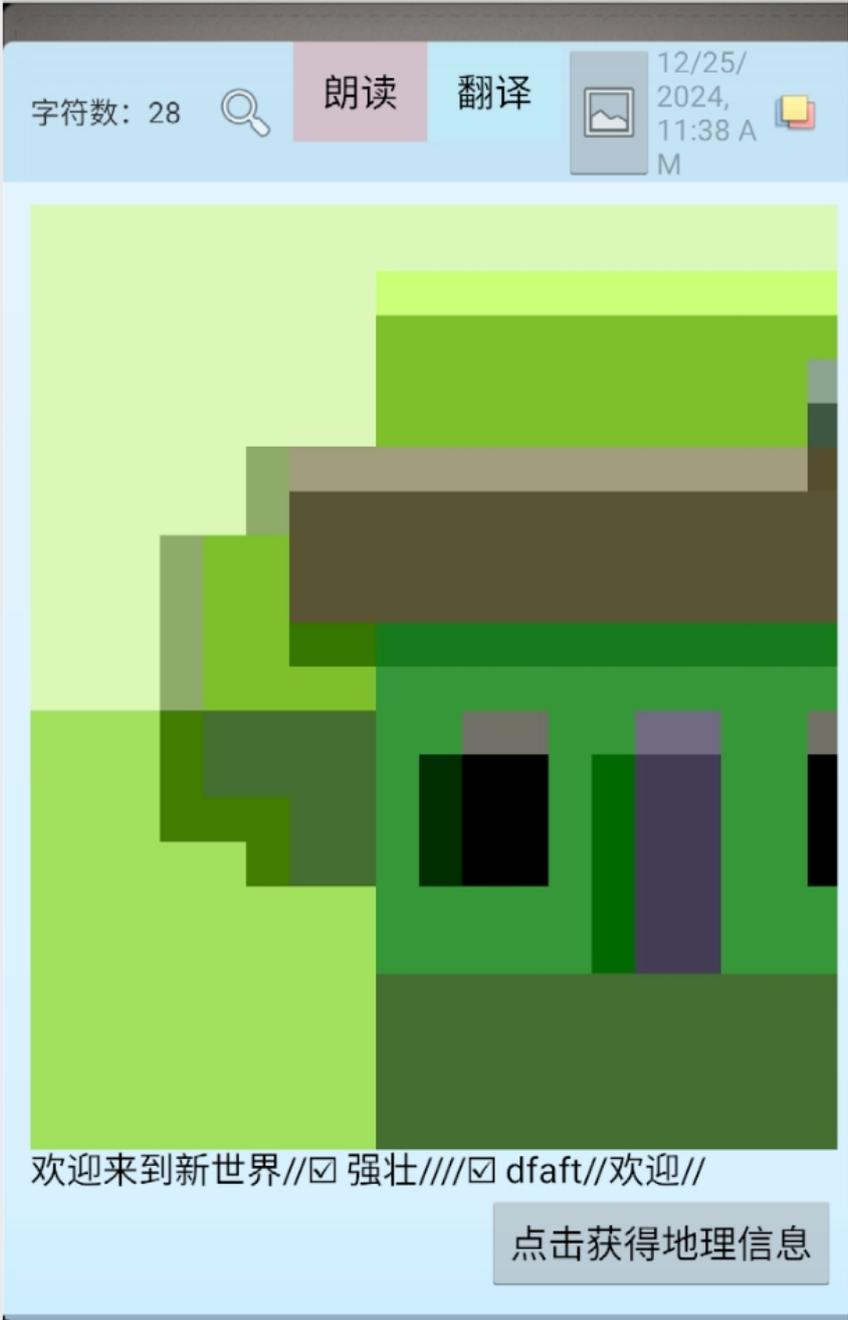


图 28: 将本地相册中的图片导入便签编辑界面中

### 3.4.6 统计字符个数

在便签编辑页面的左上角菜单里面，添加了统计总字符数的按钮，可以实时反映便签中的字符个数。如图23所示。



图 29: 统计字符个数

### 3.4.7 富文本功能

在便签编辑界面，当选中文本后，选中文本下面会显示加粗、斜体、删除线以及高亮按钮，可以对文本进行加粗、斜体、删除线以及高亮等操作。(见图24)

其中加粗功能见图25，斜体功能见图26，删除线功能见图27，高亮功能见图28。

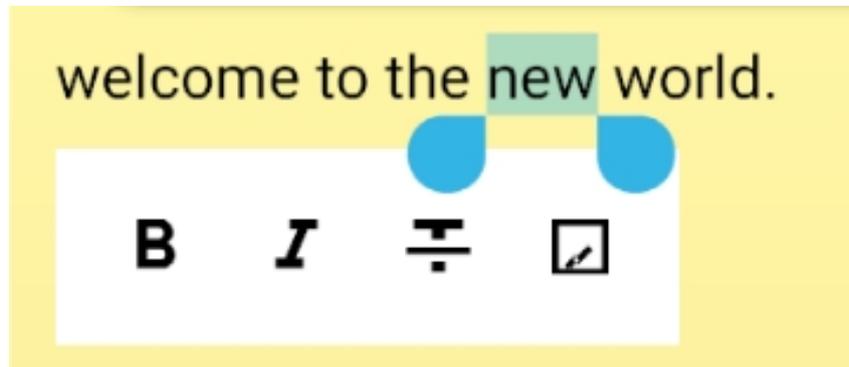


图 30: 富文本功能

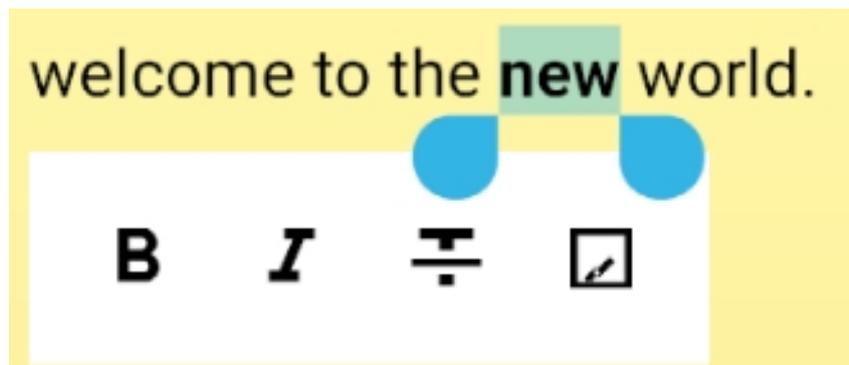


图 31: 加粗功能

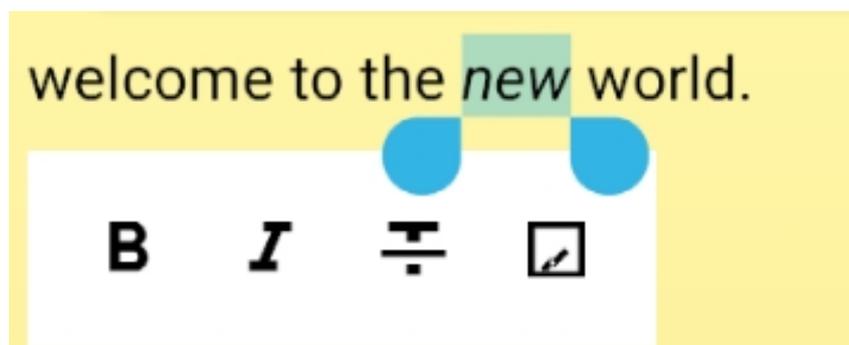


图 32: 斜体功能

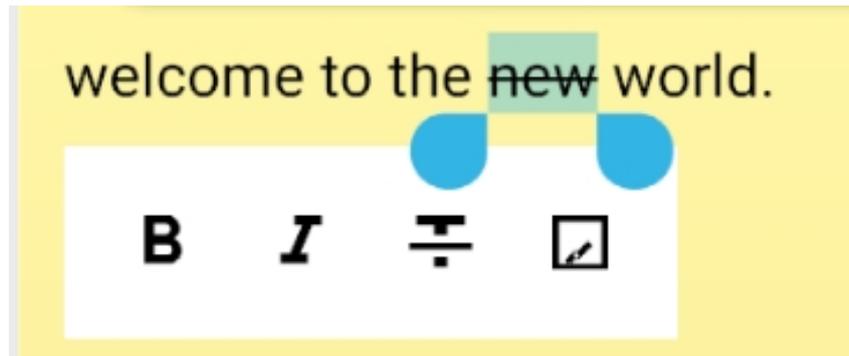


图 33: 删除线功能

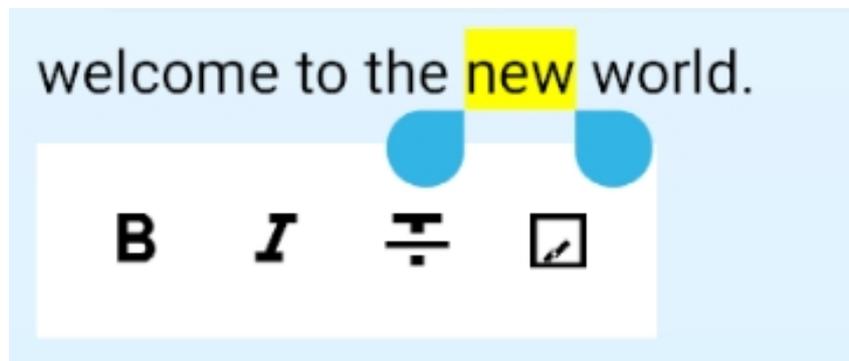


图 34: 高亮功能

### 3.4.8 朗读功能

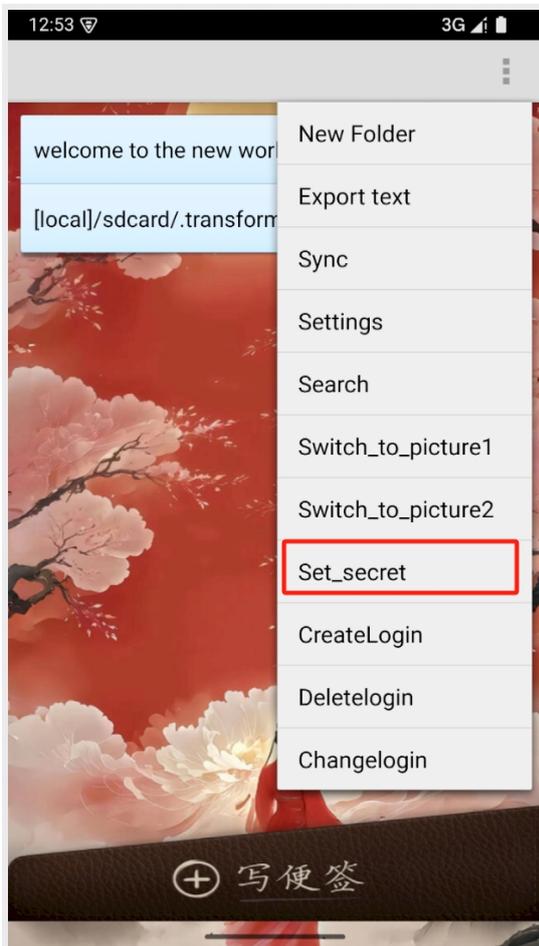
在修改便签界面上侧有朗读按钮，点击朗读按钮就可以朗读便签文本的内容（除了空格、换行符、图片路径），功能界面见图29。



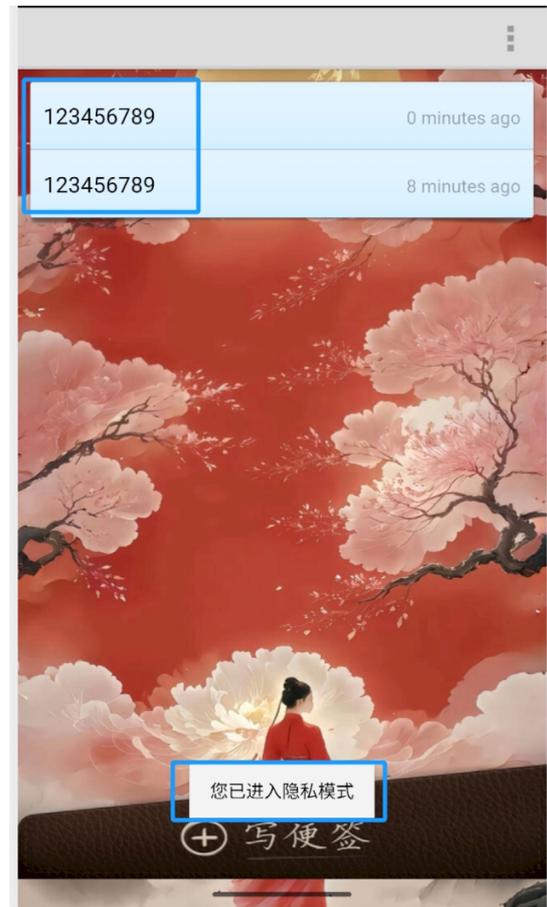
图 35: 朗读功能

### 3.4.9 私密模式

在写便签前的界面，右上角的菜单中，添加两个按钮：私密模式和退出私密模式。点击私密模式后，所有便签的在首页的显示都是“123456789”，（见图30）防止其他人通过首页了解便签内部信息。点击退出私密模式即可退出私密模式。



(a) 私密模式按钮



(b) 进入私密模式

图 36: 私密模式

当我们退出私密模式时，首行内容重新显示出来。如图31所示。

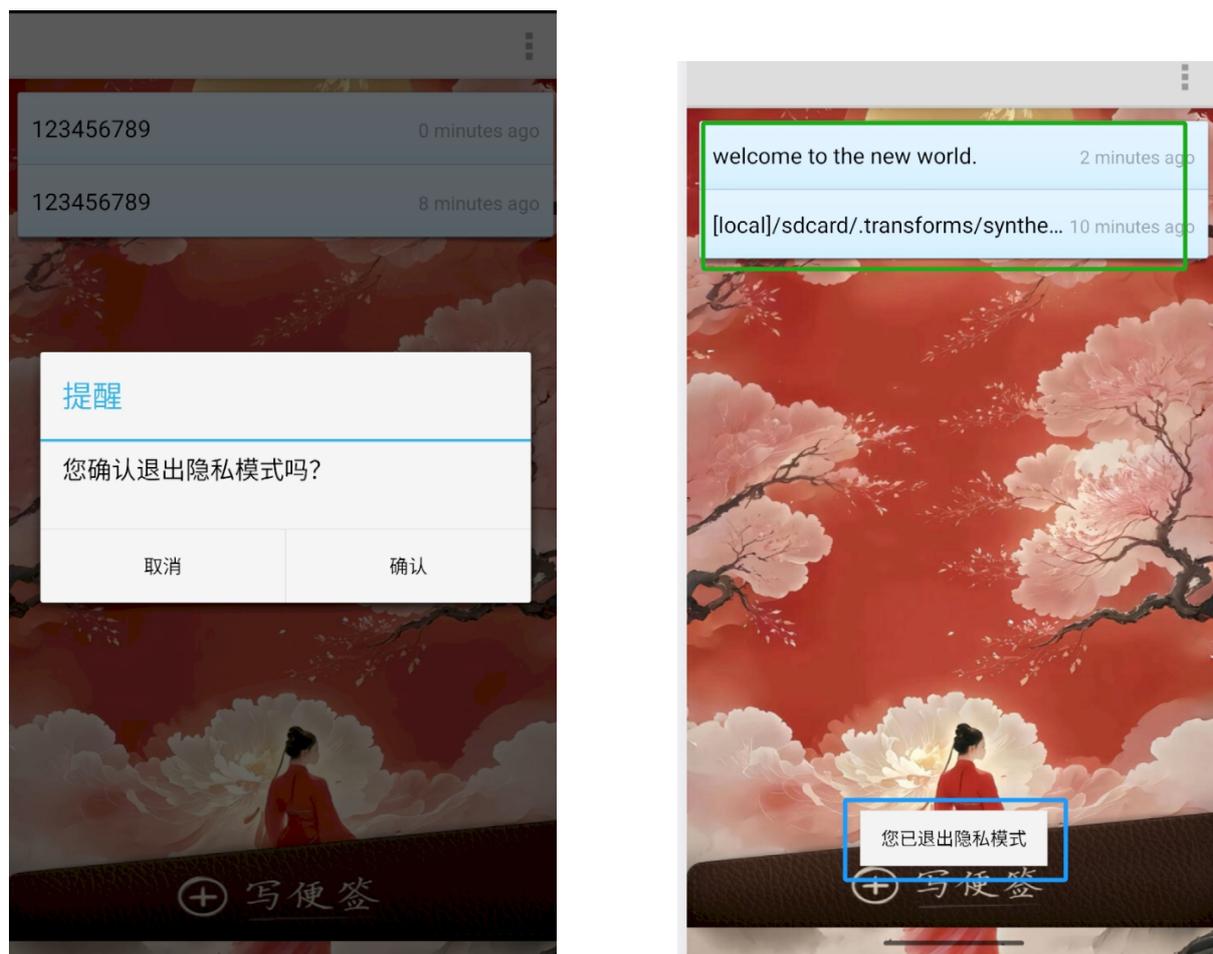


图 37: 退出私密模式

### 3.4.10 笔记编辑内搜索

- 在编辑笔记界面，点击搜索按钮，可以对笔记内容进行搜索。(见图32(a))
- 点击搜索按钮后，会弹出搜索框，输入要搜索的内容，点击确定按钮，可以搜索到所有包含该内容的内容。(见图32(b))
- 当搜索到内容时，会高亮所有搜索到的内容。(见图33(a))
- 如果搜索不到内容，会显示“未找到相关内容”。(见图33(b))
- 点击取消按钮，可以取消搜索，高亮的文本会恢复到原来的颜色。



(a) 搜索按钮



(b) 点击搜索按钮，弹出搜索框

图 38: 笔记内搜索



(a) 搜索到内容，高亮显示



(b) 搜索不到内容，显示“未找到相关内容”

图 39: 笔记内搜索

### 3.4.11 模板便签

在便签编辑界面，点击模板按钮，可以对便签内容进行模板化。（见图34）

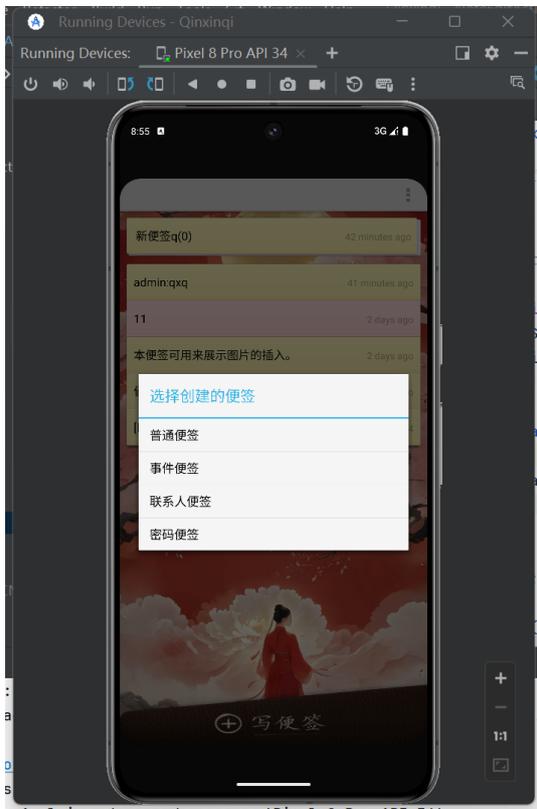


图 40: 模板便签(a)



图 41: 模板便签(b)

模板便签功能允许用户快速选择预设的便签格式，以提高便签创建的效率。用户可以在编辑界面中点击模板按钮，系统将展示可用的模板列表。选择一个模板后，便签内容将自动填充为该模板的格式，用户只需进行必要的修改即可。此功能特别适合需要频繁使用相似格式的用户，能够有效节省时间并保持便签的一致性。

### 3.4.12 语音听写

在便签编辑界面，点击语音听写按钮，可以对便签内容进行语音听写。（见图35）  
时序图如图35所示。



图 42: 语音听写(a)



图 43: 语音听写(b)

通过语音听写功能,用户可以直接通过语音输入内容,无需手动输入文字。具体使用步骤如下:

1. 在便签编辑界面点击语音听写按钮,进入语音听写模式
2. 系统会调用手机麦克风,用户可以开始说话
3. 语音识别引擎会实时将用户的语音转换为文字
4. 转换后的文字会自动插入到当前光标位置
5. 用户可以随时暂停或结束语音输入

该功能支持中文普通话识别,可以较为准确地识别日常用语和专业术语。语音识别过程在本地完成,保护用户隐私。同时支持实时预览识别结果,方便用户及时纠错。这大大提高了便签记录的效率,特别适合需要快速记录想法或无法使用手写输入的场景。

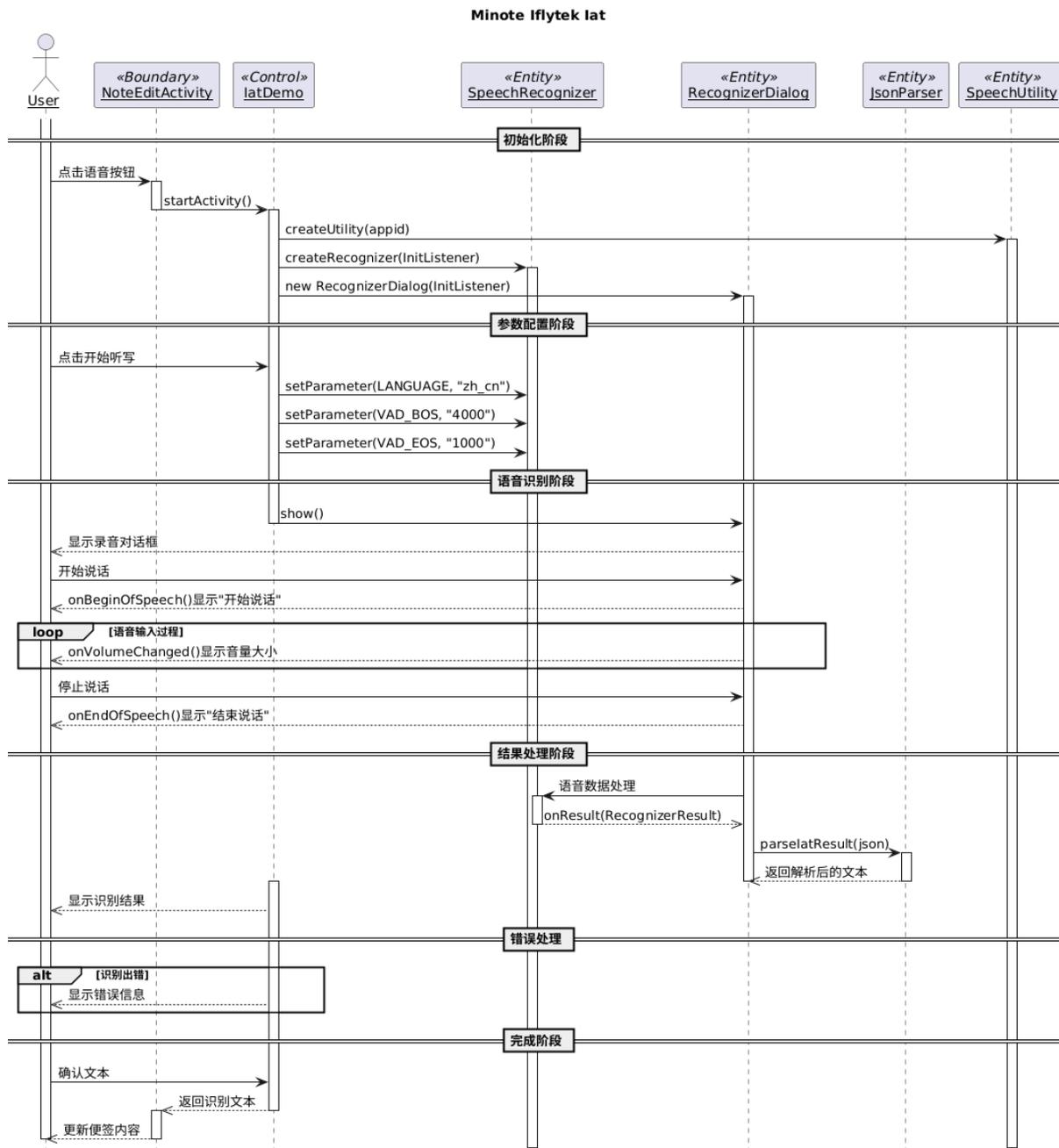


图 44: 语音听写的时序图

### 3.4.13 语音合成

在便签编辑界面，点击语音合成按钮，可以对便签内容进行语音合成。（见图36）点击语音合成按钮后，会弹出语音合成界面，输入要合成的内容，点击确定按钮，可以合成语音。（见图37）

时序图如图38所示。



图 45: 语音合成

该语音合成功能，还支持语音类型和音频类型，见图38



图 46: 语音类型



图 47: 音频类型



图 48: 语音合成的时序图

### 3.4.14 对话式大模型

在便签编辑界面和接入了对话式大模型，可以对便签内容进行对话式大模型交互。（见图39）

对话式大模型的时序图如图40所示。



图 49: 对话式大模型

picture/chatLLM.png

图 50: 对话式大模型的时序图

### 3.4.15 撤回功能

在进行便签编辑时，有时候会不小心错误的操作，这时候就需要撤回功能来恢复之前的状态。（见图40）点击菜单栏中的撤回按钮，可以恢复到上一个状态。（见图41）

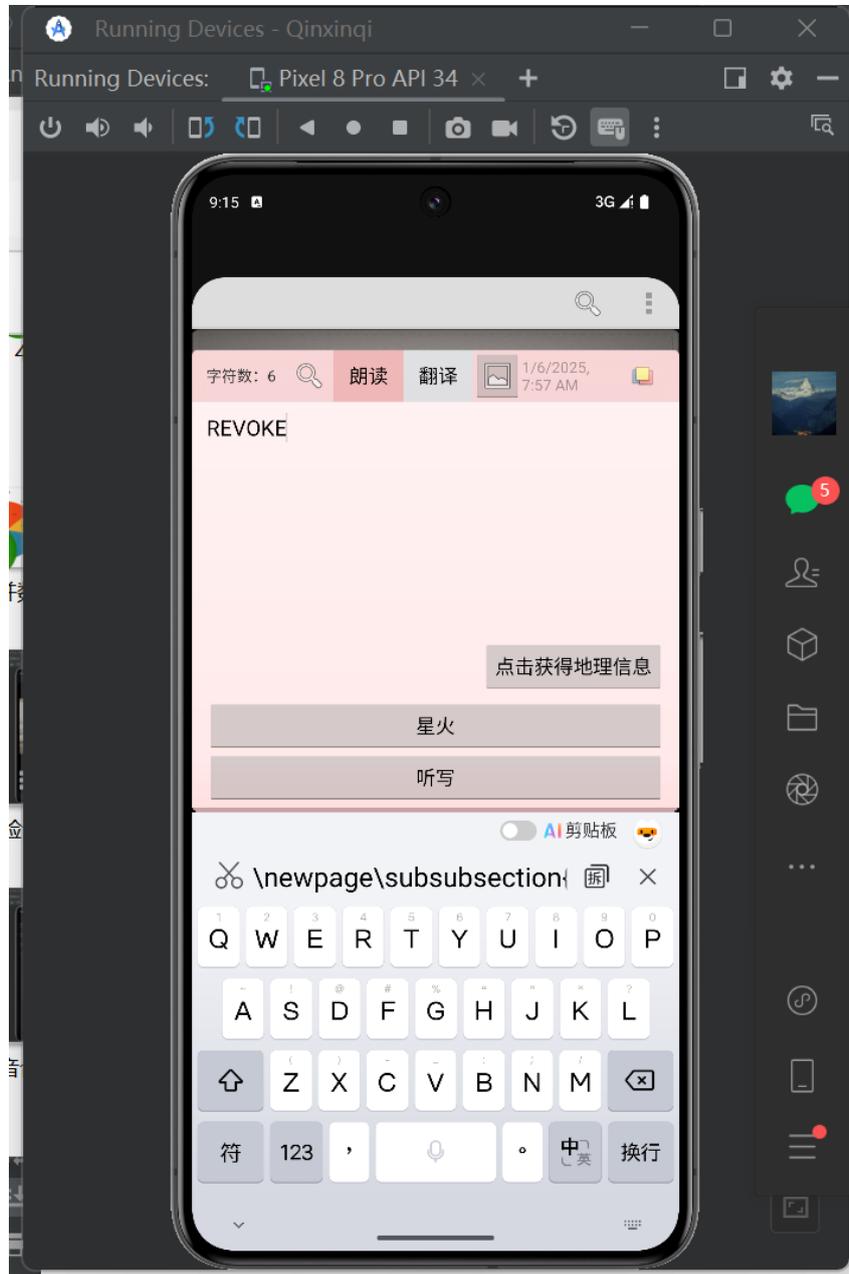


图 51: 撤回功能

### 3.4.16 获取地理位置

在编辑编辑的下方，有一个获取地理位置按钮，点击该按钮，弹出功能界面，有两个选项获取地理位置和清楚地理位置。

点击获取地理位置按钮，可以获取当前的地理位置。（见图42）

点击清楚地理位置按钮，可以清除当前的地理位置。（见图43）



图 52: 获取地理位置



图 53: 清除地理位置

## 4 实践收获和体会

### 4.1 收获

- 学会了面对特定应用的大规模代码时，如何进行逐步的分析。从略读代码分析应用的用例，各种包和各种类之间的关系，对应方法实现对应的功能等，到精读代码时，学会对代码进行标注等，再到最后对代码的质量分析以及代码的维护。
- 学学会了Java语言的使用规范，能够对Java代码进行精读并进行标注，能够使用Java编写一定行数的高质量代码，对软件功能进行维护。

- 学会了代码质量分析的方法，如人工代码质量分析和使用软件质量分析工具进行分析。特别是使用Sonarqube软件对代码的质量进行分析。
- 学会了面对大规模工程时，进行小组分工合作，将大规模的工作详细的划分成各个小工作，分给小组的成员，保证项目进度的顺利推进。同时也提高了小组合作能力以及领导分工能力。

## 4.2 问题

- 代码质量过大，包和类之间逻辑关系过于复杂时，会导致分析错误。
- 小组分工不够明确，导致工作进度缓慢。

## 5 参考文献