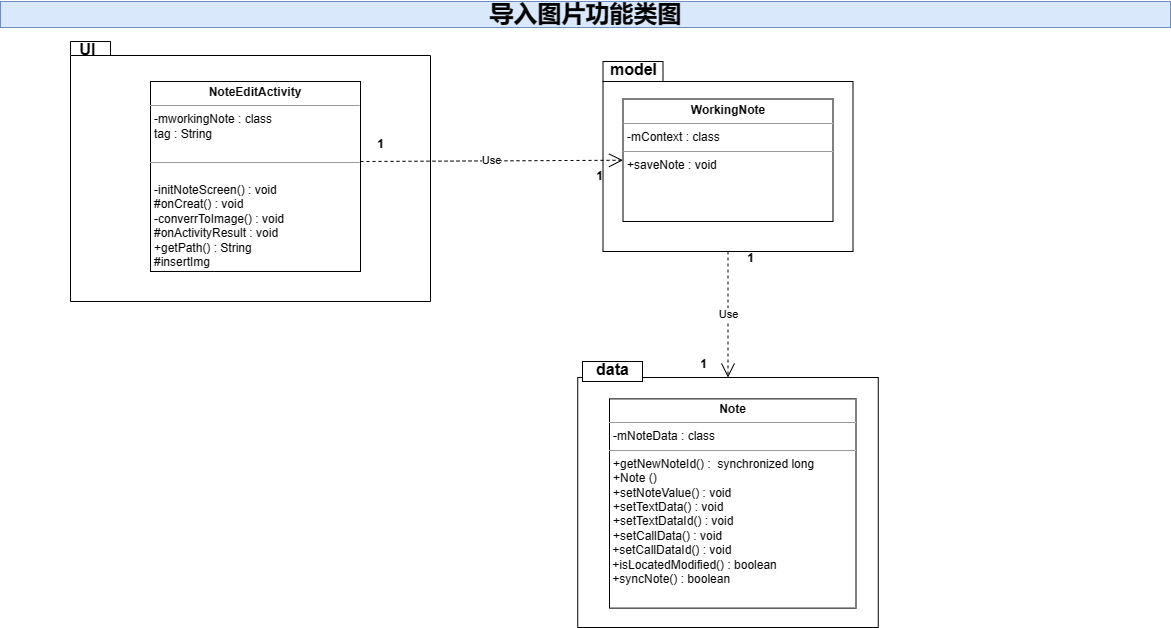
类图：

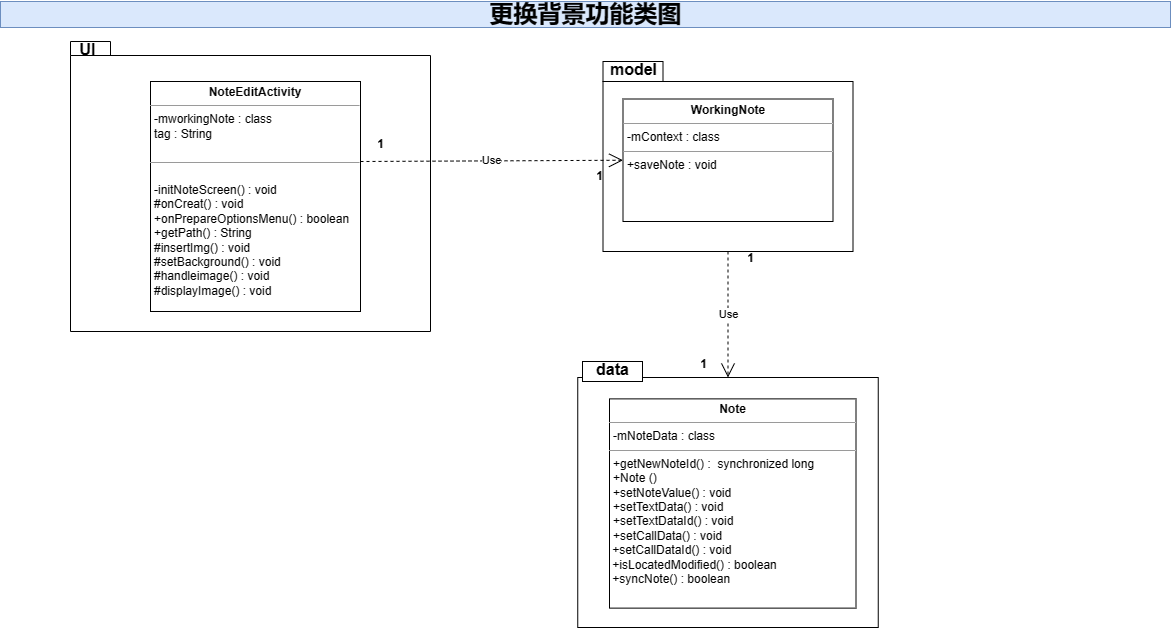
1. 导入图片



导入图片功能是先为插入功能定义一个按钮，在layout/note\_edit文件下新加入一个按钮用于表示导入图片，在NoteEditActivity类下的onClick（）中加入对按钮的点击对应的代码，新加入callGallery（）用于打开图库，调用inserImage（）用于将图片插入便签并显示。

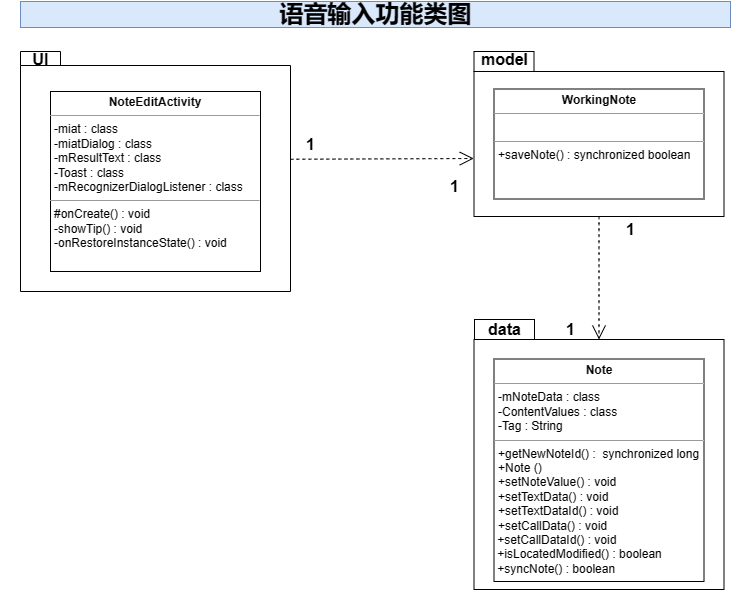
添加图片其实并不是添加真正的图片数据，而是图片的uri。通过图片uri获取图片的索引值。最后通过图片索引值获取图片的路径，并插入图片。当查看图片便签时，软件就访问对应路径，并将图片显示出来。

1. 更换背景



在layout/note\_edit文件下新加入湖水，极光，城市，水果等按钮，用来表示不同的背景图片，然后为按钮设置对应的图片路径，并事先在路径下保存有对应的图片，当用户点击按钮时可以显示按钮对应图片路径下的图片。由displayImage（）方法来显示对应的图片，在onCreate（）下显示按钮，由onPrepareOptionsMenu（）完成对按钮的隐藏功能。

1. 语音输入

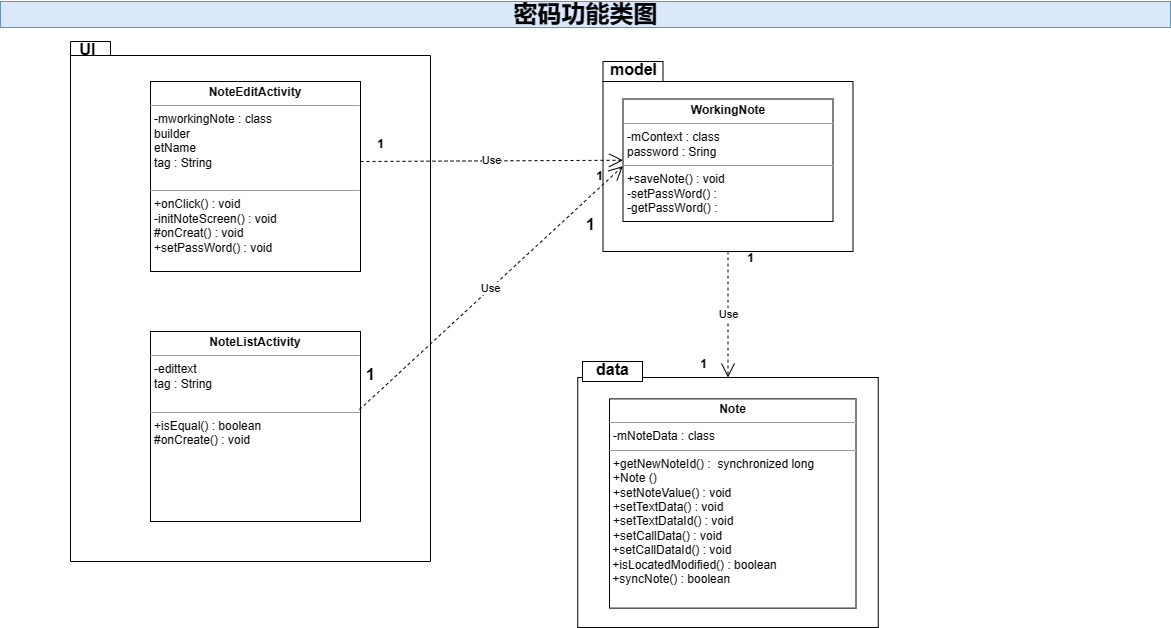


语音输入通过导入讯飞的语音识别的SDK实现，首先前往科大讯飞的开放云平台：[http://www.xfyun.cn](http://www.xfyun.cn/)，自行注册，并选择离线语音听写功能的SDK下载，并导入到Android Studio中。

在layout\note\_edit下构建新的按钮用于设置语音输入。

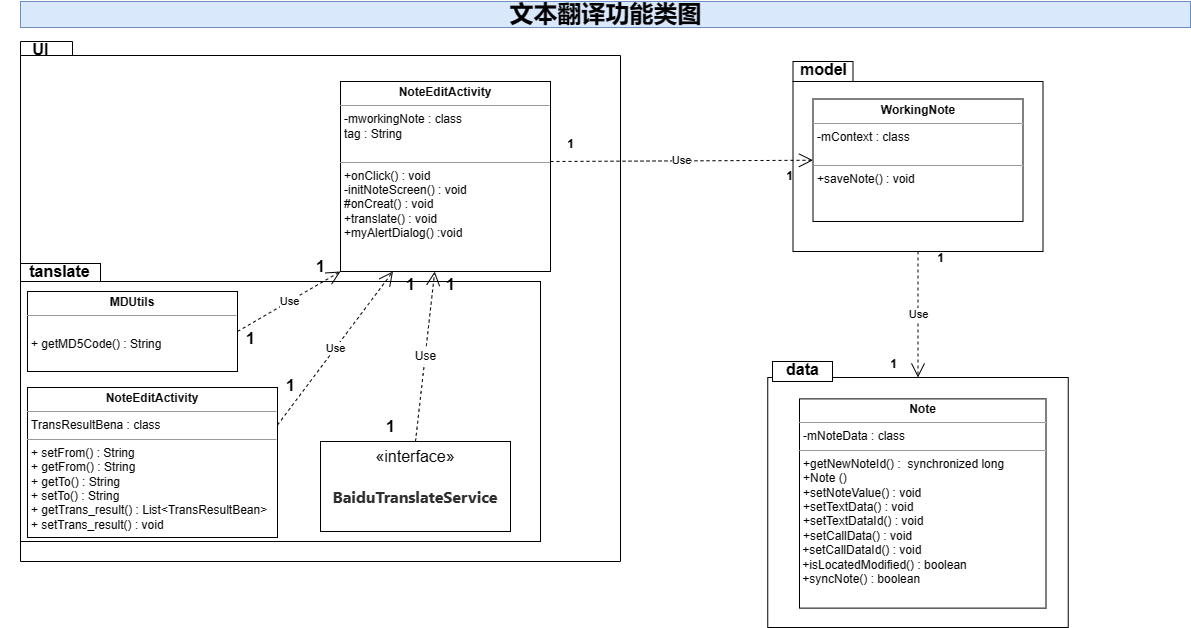
在NoteEditActivity调用SDK中的类，创建miat，miatdialog，mResultText,Toast,调用类下方法实现语音的输入功能和输出，在onCreate（）下实现按钮的功能。在前端调用离线语音听写包实现数据不回传，减少数据交互，提高速率，然后将语音听写到的内容向下传递保存文本。

1. 密码设置



在layout\note\_edit下构建新的按钮用于设置便签密码，在NoteEditActivity类下的onClick（）方法中，加入识别点击按钮下，对应的功能，先设置一个dialog响应，弹出dialog用于设置密码，在dialog中加入文本编辑框etName，在调用show方法之前给当前的dialog设置view，并用password来保存输入的字符串，用setpassword（）来保存密码，另外password的初始值设为空。

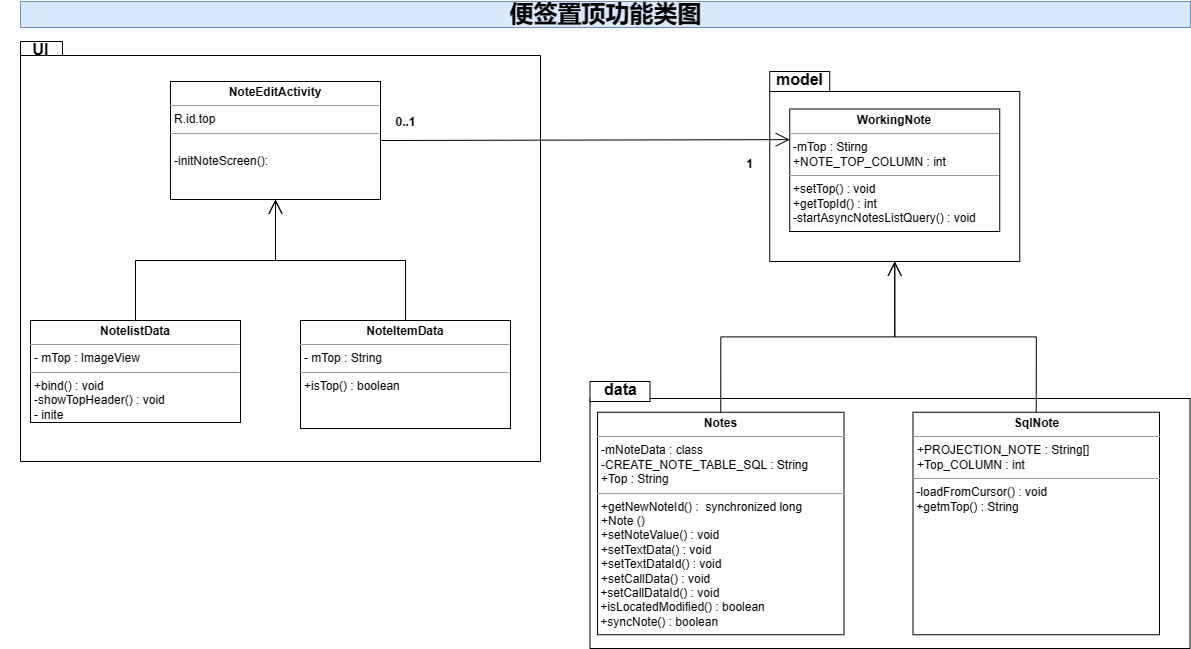
1. 文本翻译



翻译功能是用百度翻译平台的接口实现的，先加入BaiduTranslateService的接口，再加入MD5Utis类和ResponBean类。百度翻译的功能需要，登陆[百度翻译开放平台](https://fanyi-api.baidu.com/)，注册登陆百度账号，点击最上方"管理控制台",注册成为个人开发者，再点击获取通用翻译功能，实名认证获得APPID，用于翻译请标识。

从NoteEditActivity中调用出翻译的菜单，和翻译菜单的类型选项，再由NoteEditActivity获取文本内容，有NoteEditActivity负责交互myalertdialog（）用于显示菜单，onClick（）下调用Translate（）方法发起翻译请求，MD5Utis负责给文本内容加密，ResponBean负责包装文本内容，然后调用百度翻译的API接口，完成翻译最后取得文本内容，由NoteEditActivity类下的initNoteScree显示在界面上。

1. 便签置顶



先在note\_edit.xml修改布局文件并在string.xml文件里添加相应的string name用于按钮名字的定义。在NoteEditActivity的onCreate（）方法下显示按钮。

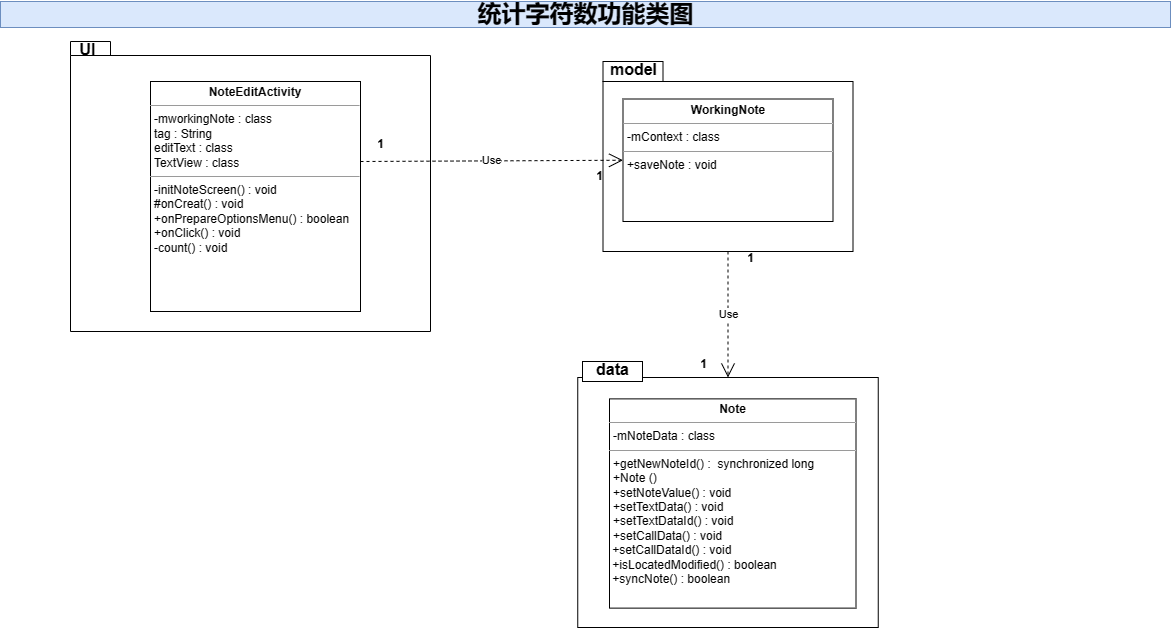
置顶功能是在点击便签后在便签里的菜单里添加的，而在便签里的功能所在的函数文件主要是在NoteEditActivity类下的onClick（）方法中，当用户点击按钮时实现对应功能。设置置顶则在workingNote类下的setTop（）getTopId（），

点击的便签在函数里是用mWorkingNote表示的

String类型参数mTop。用来表示是否为置顶便签。

点击了置顶功能的便签的top被修改为了1。但是，这仅仅是修改了数据库的数据，便签的置顶是通过读取数据库里的数据来进行排序的，通过读取数据库的数据的数据进行排序修改

1. 统计字符

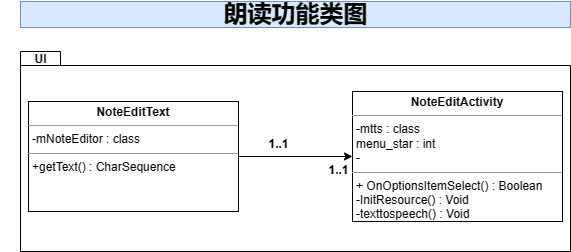


在title上显示字符数，在string.xml中添加键值对为**note\_length—Length**的字符串，同时可以添加其他语言的相应显示。这样就解决了多语言问题，系统会自动选择适配语言环境。

  采用的方法：监听**note\_edit\_view**这个**editText**控件，当有字符输入时，获取字符数，再将字符数通过**setText()**方法写到一个**textView**中，最终显示出来。

首先定义一个EditText和一个TextView， 再将note\_edit\_view绑定到editText上，将text\_num绑定到textView上，最终将上述方法封装成一个**private void count()**方法，添加到onCreate的初始化资源后面。这样每次启动这个类（即便签的编辑界面）时，就会自动实时统计字符。

1. 朗读功能



在layout\note\_edit下完成对朗读功能按钮的设计。朗读功能的实现主要利用Android自带的TextToSpeech库，oteExitActivity类中onOptionsItemSelected()方法中添加对朗读便签的响应，并且在initResourse()中初始化语音合成引擎，在编写texttospeech()方法，则可以完成语音的转换实现朗读功能。