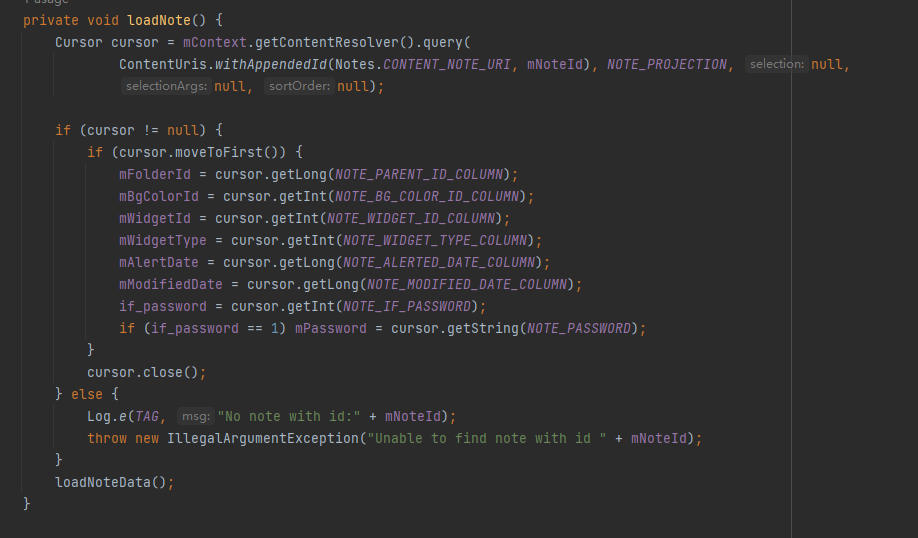
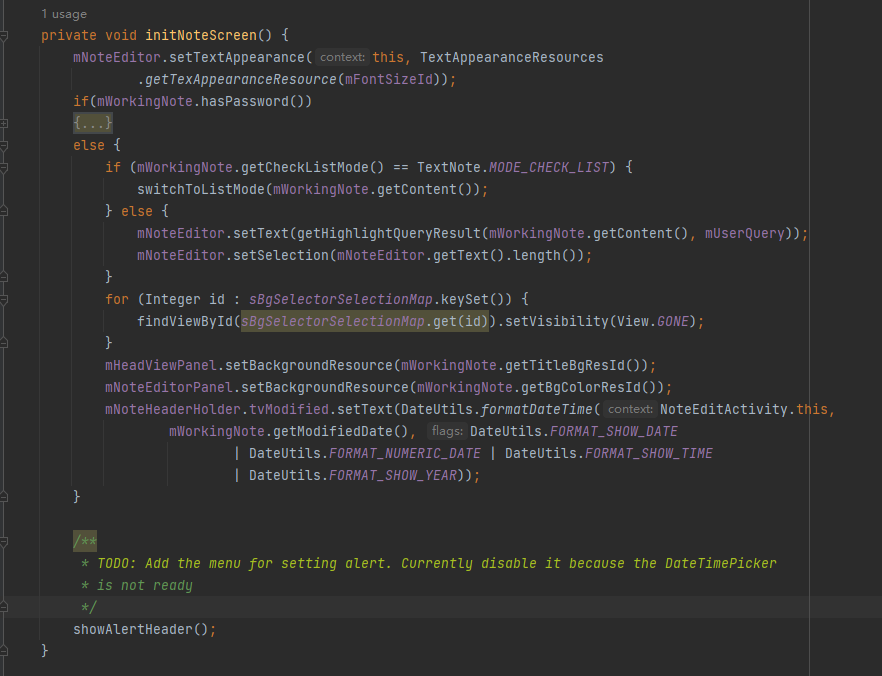
/model/WorkingNote.java

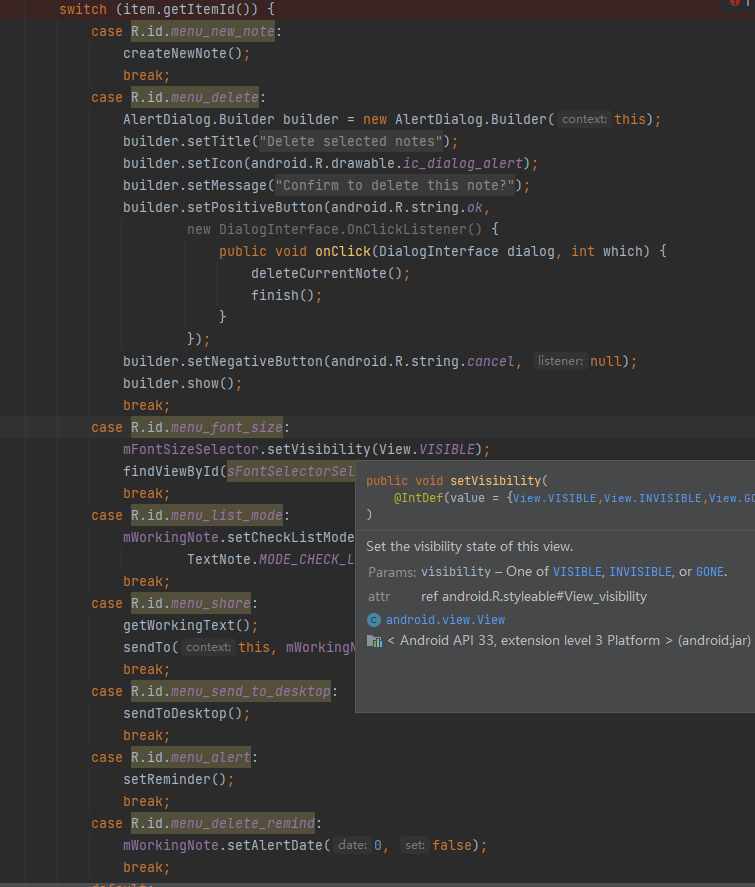


使用函数链进行主操作，可读性差，更好的方法是将所有函数在onCreate函数中调用，这样可读性好，便于观察Activity整体流程

/ui/NoteEditActivity.java

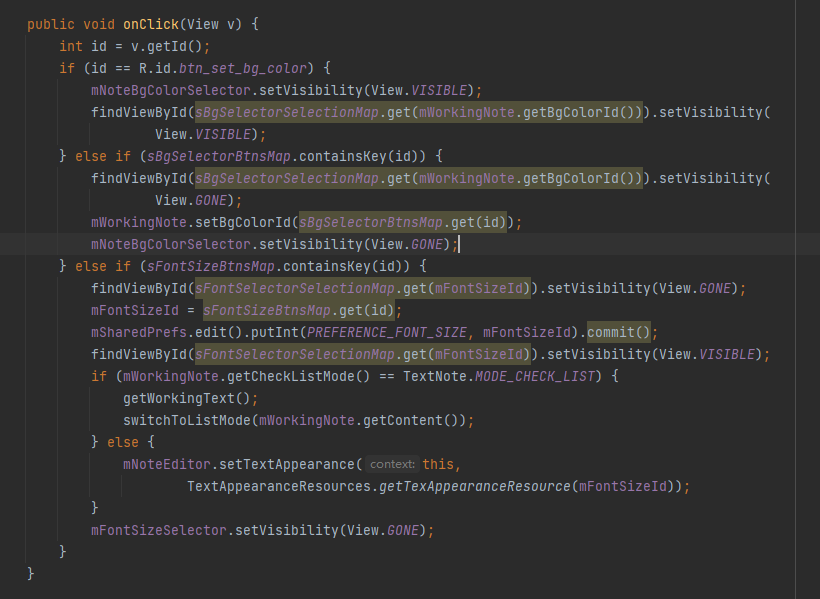


/ui/NoteEditActivity.java



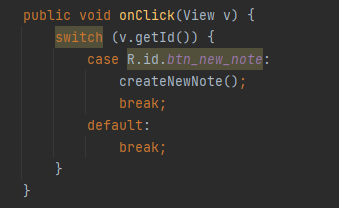
并未找到下拉菜单功能（推测是新版本的Android删除了某些固有组件），原计划将设置密码与解除密码操作置于下拉菜单中，由于未找到下拉菜单，所以在便签界面增加按钮作为替代。

/ui/NoteEditActivity.java



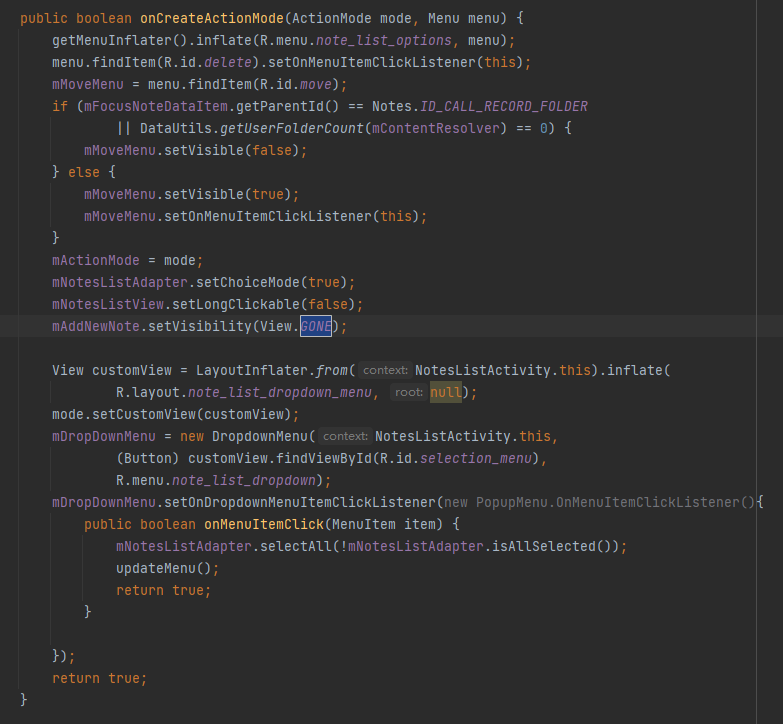
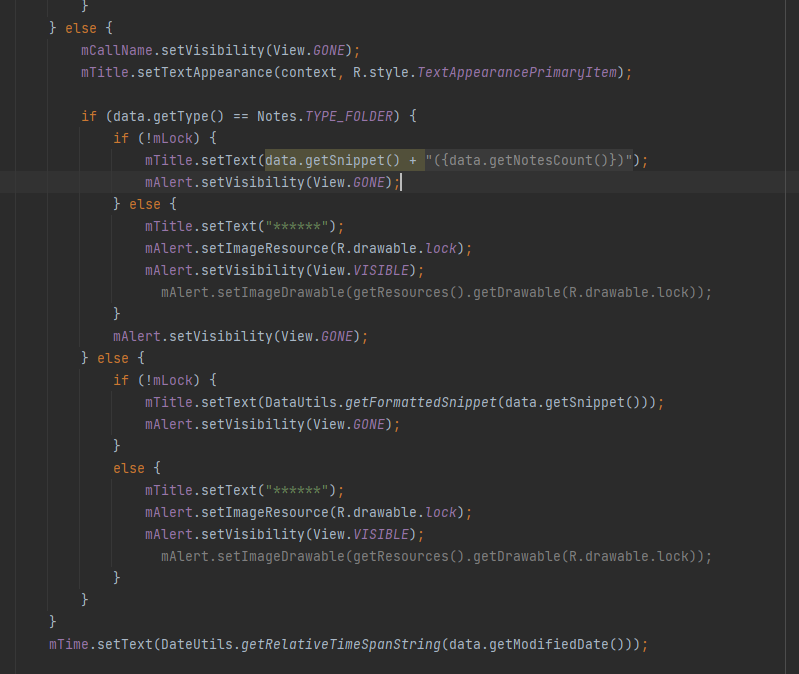
Onclick函数将所有组件集中在一起进行处理，其实对每个组件分别进行注册点击响应函数代码可能会更清晰简洁。

/ui/NotesListActivity.java



这里的switch只有一种可能，去掉switch会更好。

/ui/NotesListItem.java



对同一个视图反复的判断并设置可视化，逻辑混乱，代码维护较复杂。

可以将不同的图标设为不同的视图，分别进行判断。

5个严重问题：

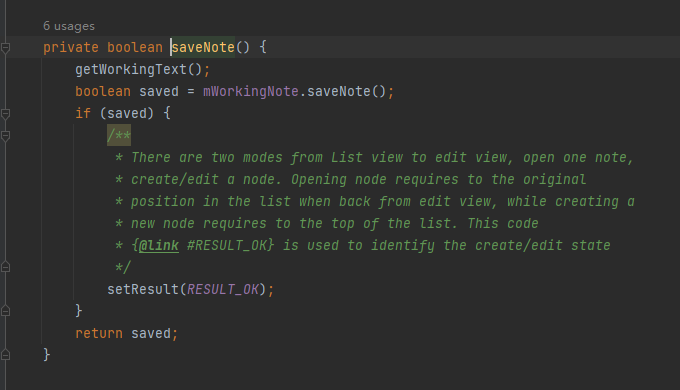
/ui/NoteEditActivity.java



便签编辑时，当用户按下返回键自动保存，而没有手动保存功能，如果程序中途崩溃，很可能会使内容丢失。

解决方案：在便签编辑ui添加手动保存按键，并实现手动保存功能，保存函数可以与此onBackPressed一致而去掉最后一步。

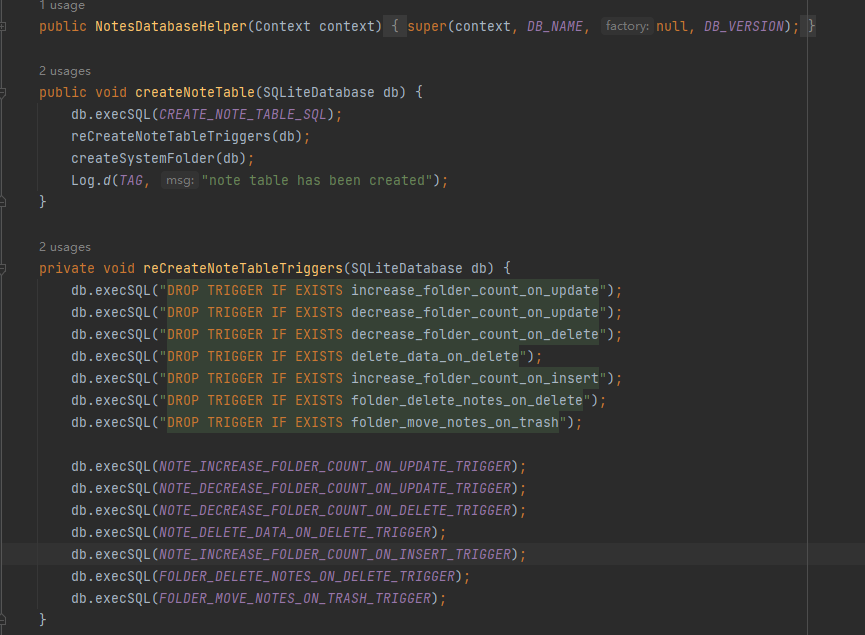
/ui/NoteEditActivity.java



saveNote函数通过获取便签界面文本框的内容进行写入保存，这样会加大实现完全加密便签的难度。即加密的情况下仍有短暂的时间会在文本框显示便签内容。

解决方案：可以修改WorkingNote，使其保存一个是否允许访问的变量，如果不允许访问则不对数据库进行修改，但经过评估该改进方法实现工作量太大，需要修改许多文件有关于WorkingNote的调用，故不予考虑。

/data/NotesDatabaseHelper.java

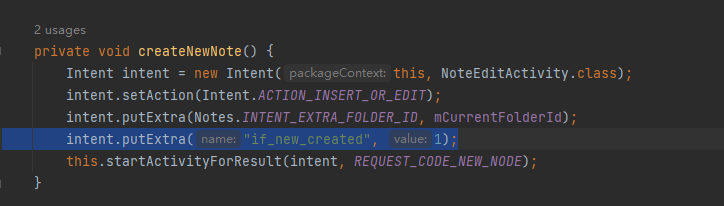


SQLite数据库并未加密（也可能是我没找到），数据库安全性不足。而且数据库丢失会造成很大程度的损失。

解决方案：1.使用SQLCipher对SQLite进行加密操作，

2.在云端创建数据库，并支持用户将便签内容上传至云端。

/ui/NotesListActivity.java



Activity传值不够完备，在NoteEditActivity中不能有效区分该便签是新建的还是已经有的，导致在添加功能的时候出现bug。需要添加“if\_new\_created字段进行传值”。

另外，startActivityForResult方法已经过时，更好地方法是使用registerForActivityResult。