1. 功能一： 关闭提醒
   1. **功能介绍**

**功能概述：**对于特定便签可以选择关闭提醒，使得对应时间的时候自动显示的闹钟图标被关闭，不再提醒用户当前时间的事项。

**操作步骤：**提醒时间到后，操作系统弹出对话框显示便签内容并响铃，闹钟变为“已过期”。用户点击关闭提醒后，闹钟不再响起。

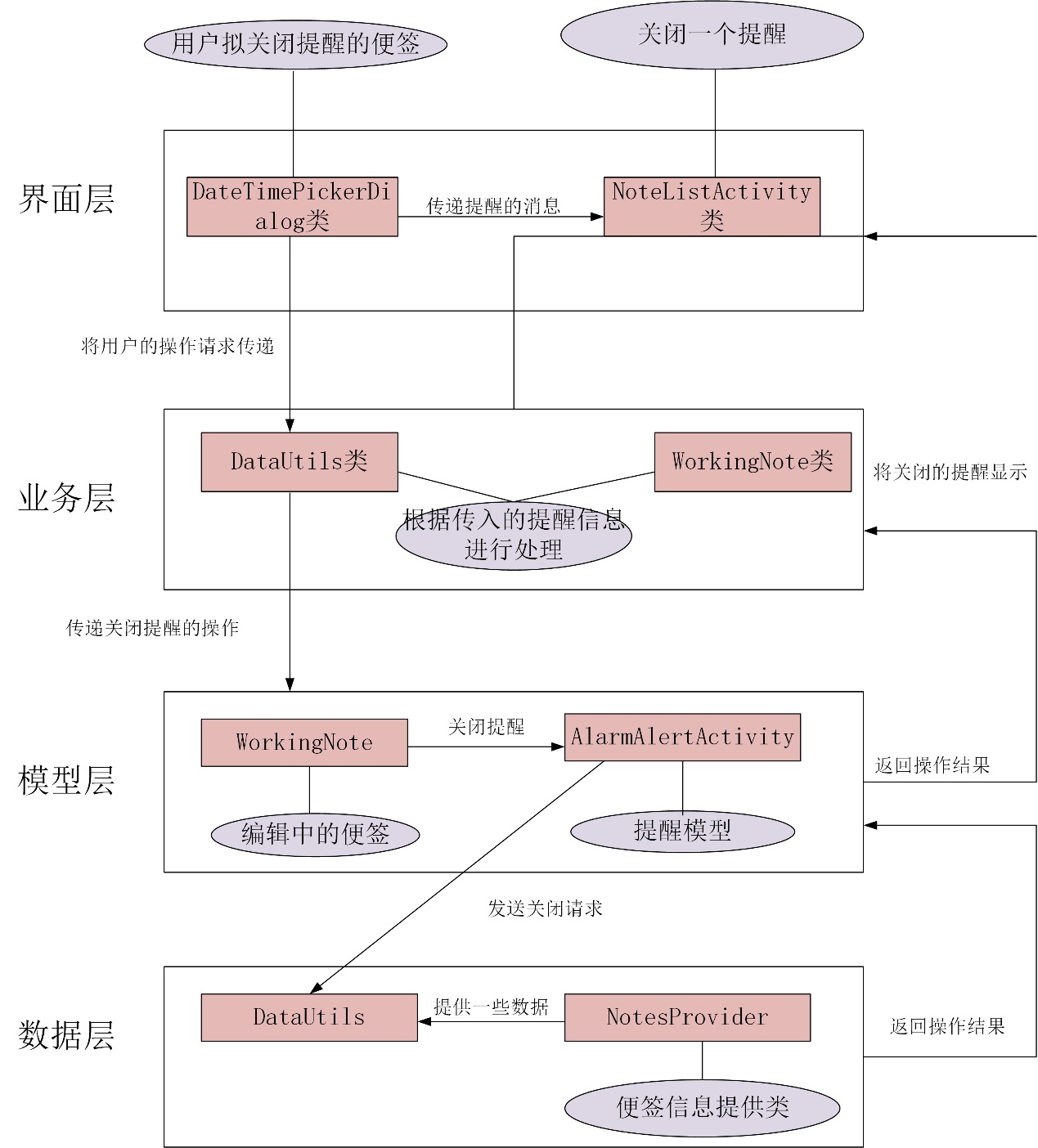
**业务请求与响应过程**

关闭提醒功能（下简称“功能”）的界面层主要由包ui中AlarmAlertActivity类通过调用showActionDialog()方法实现。showActionDialog()中使用AlertDialog.Builder构建了一个对话框，显示一个包含文件夹名称输入框和确认取消按钮的对话框。

功能的业务层由WorkingNote类中的setAlertDate()方法实现。首先，通过比较传入的日期参数(date)与当前的警报日期(mAlertDate)，判断是否需要进行更新。接着，将更新后的 mAlertDate 的值以字符串形式存储到相应的注释字段。如果存在 mNoteSettingStatusListener 对象（即监听器对象），则调用 onClockAlertChanged() 方法，将新的日期和设置状态作为参数传递给监听器。这样做是为了通知外部组件或模块，在警报日期发生改变时可以执行相关操作。

功能的模型层主要由ui包中的AlarmAlertActivity类来实现，主要包括onClick()和onDismiss()方法等。

功能的数据层主要由DataUtils类的对象来实现。包括了警报的信息，注释的内容、标题等等。



* 1. **类的作用**
     1. **界面层相关的类**

1. **类名：****AlarmAlertActivity**

该类是小米便签的提醒界面，在该界面可以进行便签提醒的相关操作，用于显示警报提醒并处理用户的操作。

1. **类的精读：**

**AlarmAlertActivity**类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteId | 警报的注释ID |
| mSnippet | 警报的摘要信息 |
| SNIPPET\_PREW\_MAX\_LEN | 摘要预览的最大长度 |
| mPlayer | MediaPlayer对象，用于播放警报声音 |

**AlarmAlertActivity**类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| isScreenOn() | / | 检查屏幕是否亮起 | 返回值：布尔值，表示屏幕是否亮起 |
| playAlarmSound() | / | 播放警报声音 | 首先获取默认警报铃声的URI。然后获取静音模式下受铃声模式影响的流。接着设置MediaPlayer的音频流类型。最后设置MediaPlayer的数据源并进行准备、循环播放和启动。 |
| showActionDialog() | / | 显示警报操作对话框。 | 创建一个AlertDialog.Builder对象。  设置对话框的标题和消息内容为警报的摘要信息。  设置对话框的确定按钮以及可能的取消按钮。  显示对话框并设置对话框消失时的监听器。 |
| onClick(DialogInterface dialog, int which) | DialogInterface dialog：被点击的对话框。  int which：被点击的按钮的标识。 | 处理对话框按钮的点击事件。 | 根据点击的按钮不同执行相应的操作，例如点击取消按钮跳转到NoteEditActivity页面。 |
| onDismiss(DialogInterface dialog) | DialogInterface dialog：被解雇的对话框。 | 对话框消失时调用，停止警报声音并关闭活动。 | 停止播放警报声音，释放MediaPlayer资源，并结束当前活动。 |
| stopAlarmSound() | / | 停止警报声音的播放。 | 停止MediaPlayer的播放，释放MediaPlayer资源。 |

* + 1. **业务层相关的类**

**类名：WorkingNote**

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 标记当前便签状态 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |
| NoteSettingChangedListener | 定义便签设置改变监听器的接口 |

表2 类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setBgClorId(int id) | Id:用来修改当前颜色的值 | 修改当前便签背景颜色的ID值 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getTitleBgResId() | 无 | 返回当前的标题颜色的ID值 | 通过背景颜色ID值来调用NoteBgResource内的方法来匹配返回标题的ID值 |
| WorkingNote(Context context, long folderId) | context:Android应用程序的上下文。  folderId:笔记所属文件夹的ID。 | 创建一个新的笔记对象 | 实例化对象 |
| getBgcolorId() | 无 | 返回当前的背景颜色的ID值 | Return直接返回 |
| getBgColorResId(): | 无 | 获取笔记背景颜色资源ID | Return直接返回 |
| getCheckListMode() | 无 | 获取笔记模式 | Return直接返回 |
| getModifiedDate() | 无 | 获取笔记修改日期 | Return直接返回 |
| getAlertDate() | 无 | 获取提醒日期 | Return直接返回 |
| getContent() | 无 | 获取笔记内容 | Return直接返回 |
| getNoteId() | 无 | 获取笔记ID | Return直接返回 |
| getWidgetType() | 无 | 获取笔记对应的小部件类型 | Return直接返回 |
| getTitleBgResId() | 无 | 获取笔记标题背景资源ID | Return直接返回 |
| createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId) | context: Android 应用程序的上下文。  folderId: 笔记所属文件夹的ID。  widgetId: 笔记对应的小部件的ID。  widgetType: 笔记对应的小部件类型。  defaultBgColorId: 笔记的默认背景颜色ID。 | 创建一个空白笔记 | 通过调用一系列set方法，设置笔记的初始情况 |
| getWidgetType() | 无 | 获取笔记对应的小部件类型 | Return直接返回 |
| isWorthSaving() | 无 | 判断当前注释是否值得保存 | 如果注释已被标记为删除(mIsDeleted)，或者注释既不在数据库中也没有内容且为空(TextUtils.isEmpty(mContent))，或者注释存在于数据库中但未进行本地修改(!mNote.isLocalModified())，则返回false；否则，返回true。 |
| **setAlertDate(long date, boolean set)** | **date：警报日期。**  **set：指示是否要设置警报日期。** | **设置警报日期并更新相关字段的值。** | **如果传入的日期与当前警报日期不相等，则更新警报日期(mAlertDate)，并将警报日期存储到注释字段中。**  **如果存在警报设置状态更改监听器(mNoteSettingStatusListener)，则通知外部组件警报日期的更改。** |
| markDeleted(boolean mark) | mark：指示是否标记为删除 | 将注释标记为已删除或未删除 | 根据传入的标记值(mark)更新mIsDeleted成员变量，表示是否标记为删除。  如果存在小部件ID和小部件类型，并且警报设置状态更改监听器不为空，则通知外部组件小部件的更改 |

* + 1. **模型层相关的类**

**类名：AlarmAlertActivity**

该类是小米便签的提醒界面，在该界面可以进行便签提醒的相关操作，用于显示警报提醒并处理用户的操作。

1. **类的精读：**

**AlarmAlertActivity**类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteId | 警报的注释ID |
| mSnippet | 警报的摘要信息 |
| SNIPPET\_PREW\_MAX\_LEN | 摘要预览的最大长度 |
| mPlayer | MediaPlayer对象，用于播放警报声音 |

**AlarmAlertActivity**类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| isScreenOn() | / | 检查屏幕是否亮起 | 返回值：布尔值，表示屏幕是否亮起 |
| playAlarmSound() | / | 播放警报声音 | 首先获取默认警报铃声的URI。然后获取静音模式下受铃声模式影响的流。接着设置MediaPlayer的音频流类型。最后设置MediaPlayer的数据源并进行准备、循环播放和启动。 |
| showActionDialog() | / | 显示警报操作对话框。 | 创建一个AlertDialog.Builder对象。  设置对话框的标题和消息内容为警报的摘要信息。  设置对话框的确定按钮以及可能的取消按钮。  显示对话框并设置对话框消失时的监听器。 |
| onClick(DialogInterface dialog, int which) | DialogInterface dialog：被点击的对话框。  int which：被点击的按钮的标识。 | 处理对话框按钮的点击事件。 | 根据点击的按钮不同执行相应的操作，例如点击取消按钮跳转到NoteEditActivity页面。 |
| onDismiss(DialogInterface dialog) | DialogInterface dialog：被解雇的对话框。 | 对话框消失时调用，停止警报声音并关闭活动。 | 停止播放警报声音，释放MediaPlayer资源，并结束当前活动。 |
| stopAlarmSound() | / | 停止警报声音的播放。 | 停止MediaPlayer的播放，释放MediaPlayer资源。 |

* + 1. **数据层相关的类**

**1、类名：**DataUtils

1. **类的作用**

一个数据处理的工具类，能够包含了一系列用于处理笔记数据的静态方法。这些方法涵盖了笔记的批量删除、移动到文件夹、获取文件夹的笔记部件信息、检查文件夹名是否存在等功能。

1. **类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

表2 类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个 对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |
| moveNoteToFoler | resolver：用于与应用程序的数据提供者进行通信。  id：要移动的便签的唯一标识符。  srcFolderId：便签当前所在的文件夹的唯一标识符。  desFolderId：要移动到的目标文件夹的唯一标识符。 | 将指定的便签移动到不同的文件夹中 | 创建一个对象，用于存储要更新的便签的新值。之后更新便签的属性值。  将更新后的 values 对象应用于指定的便签。指定要更新的便签的位置。 |
| batchMoveToFolder | resolver：用于交互。  ids：包含要移动的便签的 ID。  folderId：要移动到的目标文件夹的 ID。 | 将一组便签移动到指定文件夹 | 对于 ids 中的每个便签 ID，创建一个Operation对象，并设置更新的 URI 和要修改的列的值。这里的列包括父文件夹ID和本地修改标志。  将创建的Operation对象添加到 operationList 中。  使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。  根据返回的结果进行判断：如果 results 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 false。  如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。 |