# 详细设计

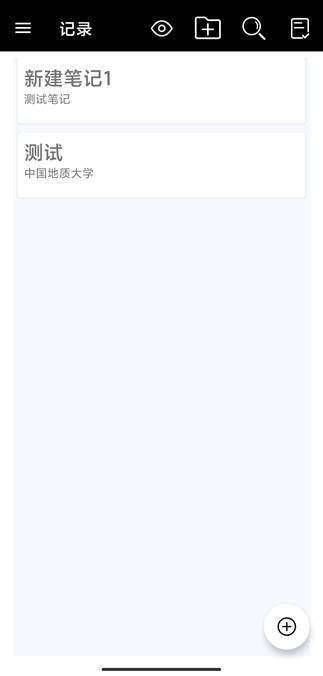
## 笔记列表

### 功能设计

笔记列表模块的**功能**是显示当前选中笔记本下的所有笔记及其笔记预览

### 界面设计

笔记列表模块的界面显示如下图



该fragment的xml布局如下,其组件树为



其组件结构如下

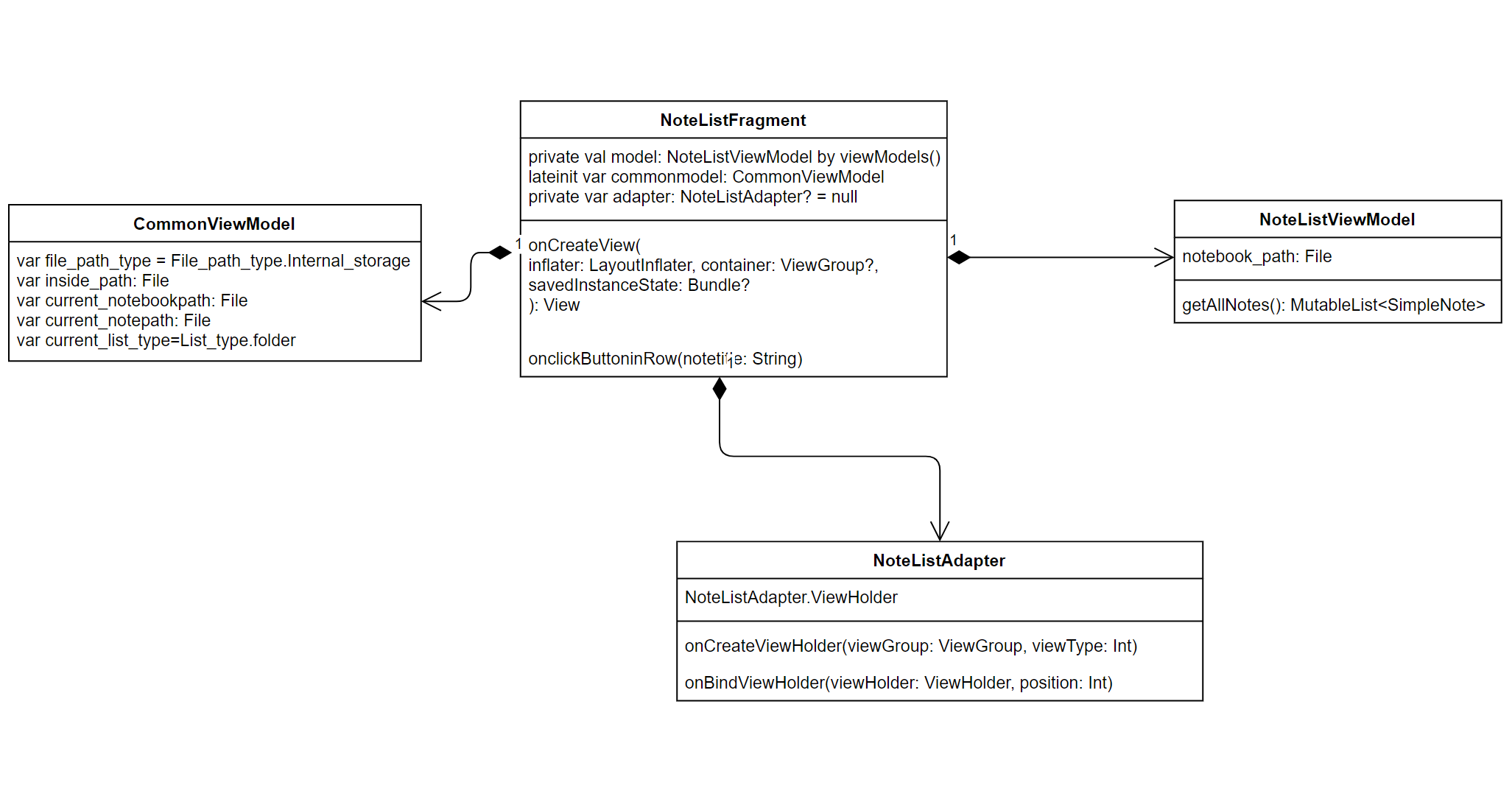
* androidx.constraintlayout.widget.ConstraintLayout
  + androidx.recyclerview.widget.RecyclerView

### 功能实现概述

该模块作为一个单独的fragment显示,通过NoteListFragment类实现,其核心实现类及其功能如下

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 功能 |
| NoteListFragment | 类 | fragment的实现类,负责界面绘制与用户交互 |
| NoteListAdapter | 类 | 适配器类，重写方法，其对象作为recyclerView的适配器 |
| ViewHolder | 类 | 关联数据对象与行模板，实现事件响应函数 |
| NoteListViewModel | 类 | 保存数据，进行Note列表的获取 |
| OnNoteClickListener | 接口 | 响应函数的接口 |

UML图如下



### 核心功能实现

在NoteListViewModel类中进行核心的Note列表的获取，利用NoteListAdapter更新NoteListFragment的界面，通过实现OnNoteClickListener接口实现鼠标响应，即点击进入编辑器

## Markdown编辑器

### 功能设计

本模块的核心功能是渲染打开的Markdown文件，并能够进行各种形式的编辑，如加粗，添加图片，添加录音等。

### 界面设计

功能界面由标题栏，内容栏，底部编辑功能栏，顶部功能栏组成。

各功能组件包含以下功能：

|  |  |
| --- | --- |
| 组件名称 | 功能 |
| 标题栏 | 输入与显示标题 |
| 内容栏 | 输入与显示经过Markdown渲染后的内容（包括文字，图片，音频） |
| 底部编辑功能栏 | 对内容进行操作，功能包括：加粗，斜体，高亮，一级标题，二级标题，添加图片，添加音频等 |
| 顶部功能栏 | 对笔记进行操作，功能包括：导出，分享，搜索，协同编辑，切换编辑/查看视图等 |



image-20220519202950676

其组件XML结构如下

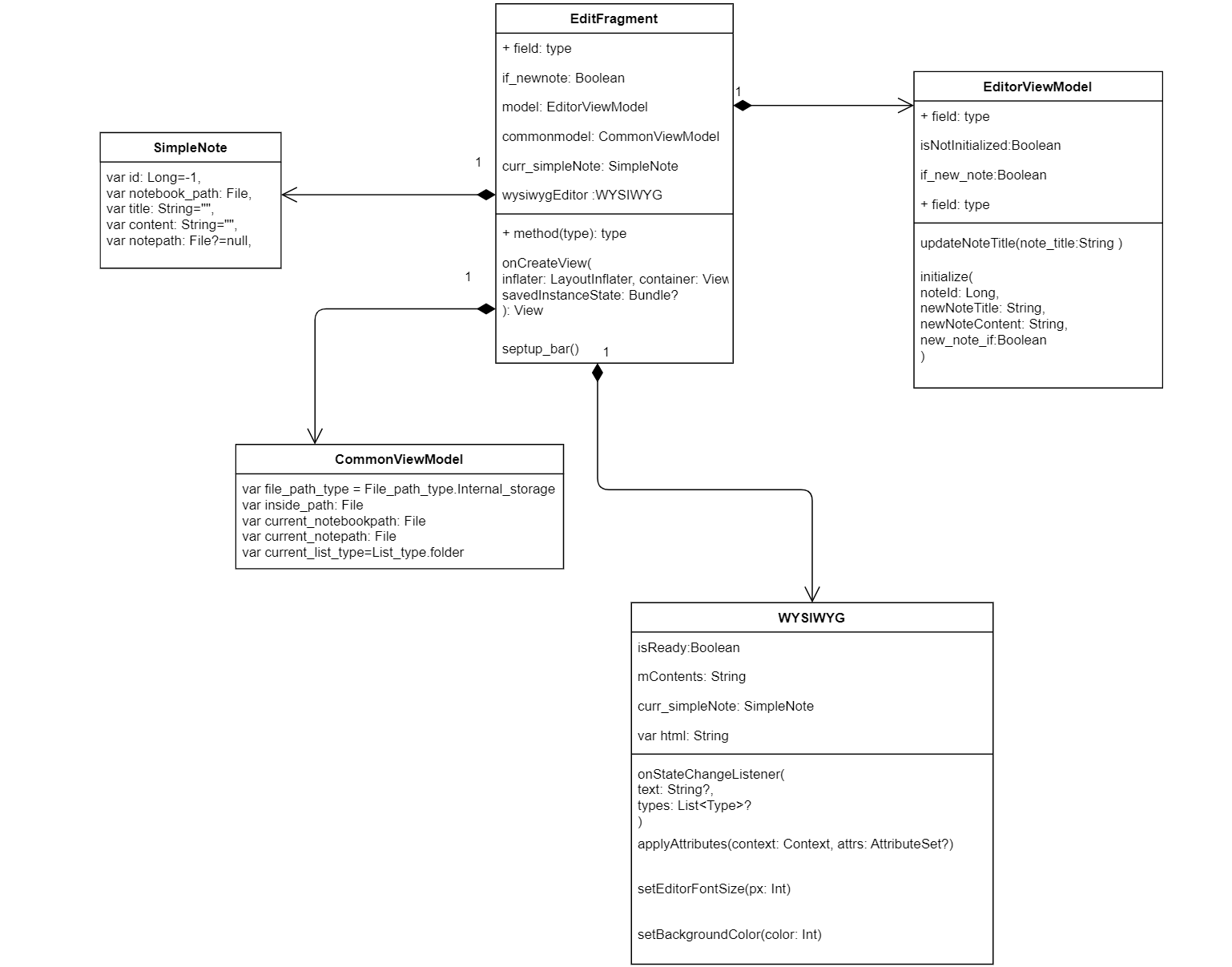
* layout
  + data
    - variable
  + androidx.constraintlayout.widget.ConstraintLayout
    - com.google.android.material.textfield.TextInputLayout
    - com.example.timetaker.ui.edit.WYSIWYG
    - androidx.appcompat.widget.Toolbar

### 功能实现概述

该模块作为一个单独的fragment显示,通过NoteListFragment类实现,其核心实现类及其功能如下

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 功能 |
| EditFragment | 类 | fragment的实现类,负责界面绘制与用户交互 |
| WYSIWYG | 类 | 继承自WebView的编辑器 |
| EditorViewModel | 类 | 保存数据，保存当前Note状态 |

UML图如下



### 核心功能实现

#### 数据绑定

利用Jetpack组件中的\*\* DataBinding 与 ViewBinding实现控件与数据的绑定。

在该fragment中将笔记显示控件的内容与SimpleNote类进行关联

<data>  
 <variable name="note" type="com.example.timetaker.data.model.SimpleNote" />  
</data>

#### webview编辑器

Markdown作为纯文本格式的标记语言,与部分html标签对应,适合进行浏览器渲染。

构建WYSIWYG组件进行笔记的渲染。

webview页面结构如下：

* editor.html
* normalize.css
* style.css
* w3.css
* wysiwyg.js

# 四、技术难点

## 设计思想的重难点

程序的设计思想如下：

1. **MVVM现代架构设计**：严格的MVVM架构，具有较强可扩展性
2. **纯Markdown格式笔记文件**：便于用户自定义与文件的批量转移。
3. **文件夹组织形式**：笔记通过文件夹进行组织,便于归类整理。
4. **数据库存储笔记路径**：数据库中存储笔记路径与配置、权限等信息
5. **WebView组件实现Markdown编辑器**：Markdown作为纯文本格式的标记语言,与部分html标签对应,适合进行浏览器渲染

## MVVM架构

MVVM是Model-View-ViewModel的简写，其中

**View:** 对应于Activity和XML，负责View的绘制以及与用户交互。

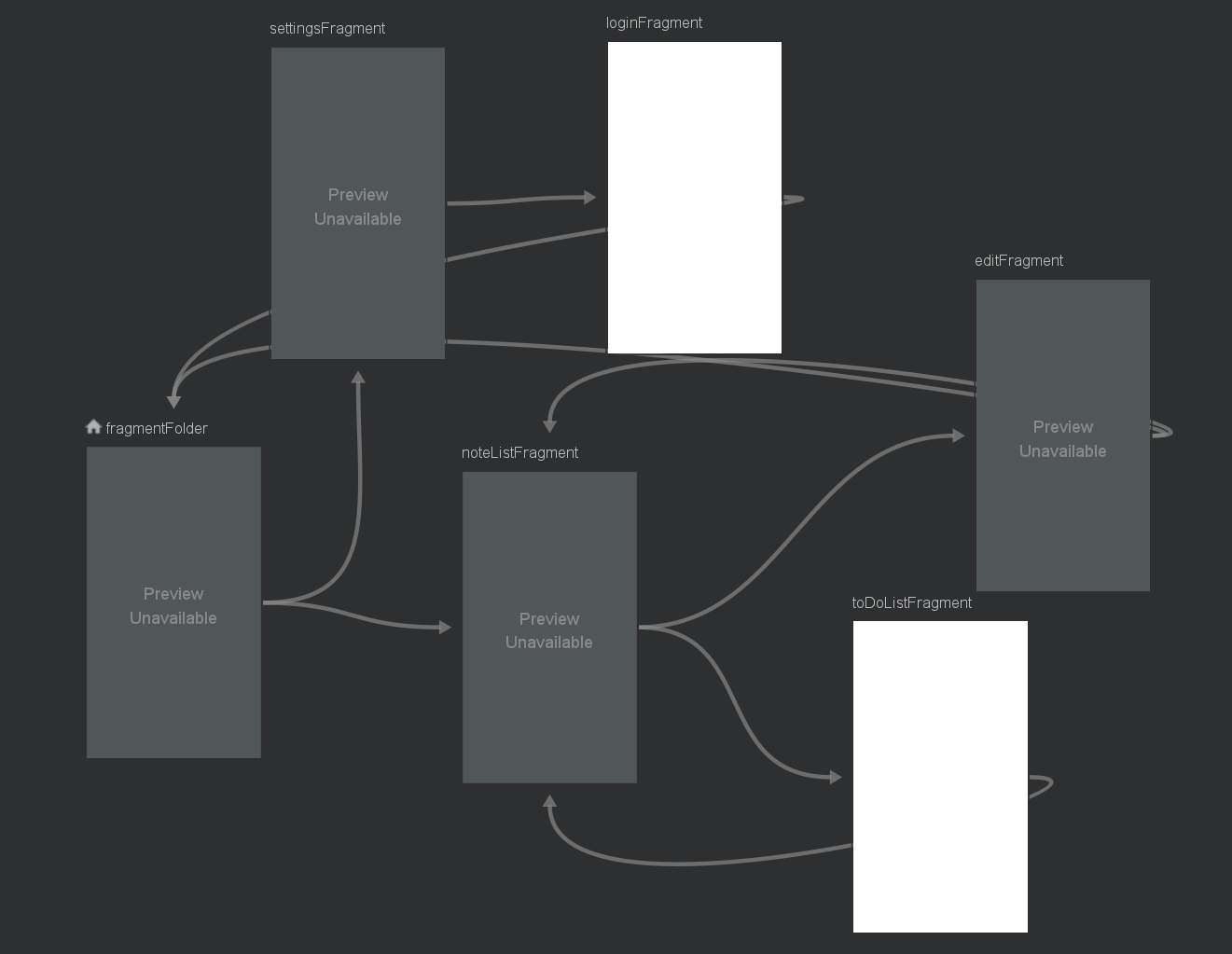
**Mode**l: 实体模型。

**ViewModel**: 负责完成View与Model间的交互，负责业务逻辑

### View层

采用**单activity多fragment**的页面组织结构，使用Jetpack组件中的**Navigation**组件对fragment进行管理与跳转，利用Jetpack组件中的DataBinding 与 ViewBinding实现控件与数据的绑定。

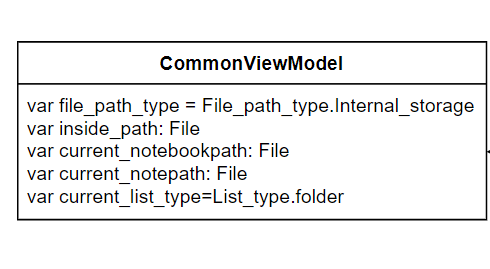
多fragment的导航图如下



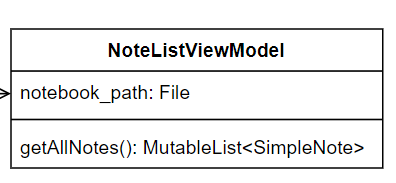
### ViewModel层

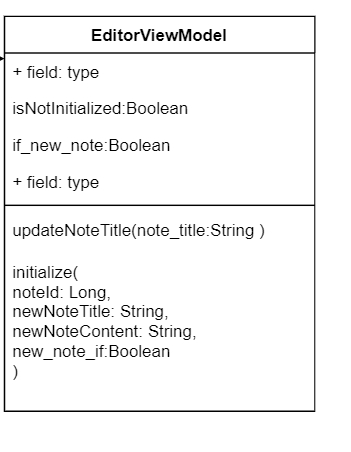
ViewModel只做和业务逻辑和业务数据相关的事,通过Jetpack组件中的ViewModel与LiveData实现

建立CommonViewModel进行各fragment间的数据交流,存储如当前笔记路径,当前笔记名称等,当前打开笔记本名称路路径等信息



各个fragment建立各自的ViewModel,由于其独立activity生命周期,避免数据受到屏幕旋转等生命周期变化的影响. 其只关注数据和业务逻辑,不涉及UI控件的引用与更新,实现View和ViewModel之间的松散耦合.





### **Mode**l层

Model层赋予了数据获取的职责，将数据的获取、存储、数据状态变化监听，获取网络数据接口，本地存储（增删改查）接口作为Model层的任务，提供数据获取接口供ViewModel调用。