

智记便签

软件需求规格说明书

Software Requirements Specification

版本：1.0

日期：2025 年 11 月 5 日

目录

1	引言	3
1.1	编写目的	3
1.2	项目背景	3
1.3	定义、首字母缩写词和缩略语	3
1.4	参考资料	4
2	项目概述	4
2.1	产品描述	4
2.2	产品功能	4
2.2.1	原有核心功能	4
2.2.2	新增功能	5
2.3	用户特点	5
2.4	运行环境	5
2.5	约束和假定	6
3	系统需求	6
3.1	功能需求	6
3.1.1	原有功能的需求概述	6
3.1.2	新功能 1: Markdown 富文本支持	6
3.1.3	新功能 2: 增强的导出分享功能	9
3.2	非功能需求	11
3.2.1	性能需求	11
3.2.2	可靠性需求	12
3.2.3	可维护性需求	13
3.2.4	安全性需求	13
3.2.5	可移植性需求	14
3.2.6	用户界面需求	14
4	数据模型	14
4.1	数据库设计	14
4.1.1	note 表	15
4.1.2	data 表	15
4.2	数据流	15

5	接口需求	16
5.1	用户界面	16
5.1.1	便签列表界面	16
5.1.2	便签编辑界面	16
5.1.3	格式选择对话框	17
5.2	软件接口	17
5.2.1	Android 系统接口	17
5.2.2	第三方库接口	17
5.3	硬件接口	17
6	质量属性	17
6.1	可用性	17
6.2	可靠性	18
6.3	性能	18
6.4	可维护性	18
7	项目约束	18
7.1	时间约束	18
7.2	成本约束	18
7.3	技术约束	19
8	附录	19
8.1	术语表	19
8.2	参考模型	19
8.3	更新记录	20

1 引言

1.1 编写目的

本文档详细说明了智记便签（Smart Note）软件的功能需求、性能需求、设计约束和质量属性。智记便签是在小米便签基础上进行功能增强和优化的移动端便签应用。本文档的预期读者包括：

- 项目开发人员：作为软件设计和实现的依据
- 项目管理人员：作为项目规划和进度控制的参考
- 测试人员：作为测试用例设计和验收测试的基础
- 用户：了解系统功能和使用方式

1.2 项目背景

项目名称：智记便签（Smart Note）

项目背景：随着移动互联网的发展，用户对移动端便签应用的需求日益增长。传统的纯文本便签已经无法满足用户对内容组织、格式化和数据交互的需求。智记便签在小米便签的基础上，引入 Markdown 富文本支持和增强的导出分享功能，为用户提供更加专业和高效的笔记体验。

用户：需要记录、整理和分享信息的移动设备用户，包括学生、职场人士、知识工作者等。

1.3 定义、首字母缩写词和缩略语

术语	定义
SRS	Software Requirements Specification, 软件需求规格说明
Markdown	一种轻量级标记语言, 允许使用纯文本格式编写文档并转换为格式化的内容
WYSIWYG	What You See Is What You Get, 所见即所得编辑器
MIME	Multipurpose Internet Mail Extensions, 多用途互联网邮件扩展类型
UI	User Interface, 用户界面
API	Application Programming Interface, 应用程序编程接口
SQLite	一种轻量级的关系型数据库管理系统

ContentProvider	Android 平台提供的数据共享机制
-----------------	---------------------

1.4 参考资料

1. IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications
2. GB/T 9385-2008 计算机软件需求规格说明规范
3. 小米便签 (MiCode Notes) 开源项目: <https://github.com/MiCode/Notes>
4. Android 开发者文档: <https://developer.android.com/>
5. Markdown 规范: <https://commonmark.org/>

2 项目概述

2.1 产品描述

智记便签是一款功能丰富的移动端便签应用,提供便签的创建、编辑、组织、搜索和同步功能。相比于原小米便签,智记便签新增 Markdown 富文本编辑功能和增强的数据导出分享能力,使用户能够创建格式化的内容并以多种格式分享给其他应用或用户。

2.2 产品功能

智记便签的主要功能包括:

2.2.1 原有核心功能

1. **便签管理**: 创建、编辑、删除便签,支持文本输入
2. **分类管理**: 通过文件夹组织便签
3. **提醒功能**: 为便签设置时间提醒
4. **云同步**: 与 Google Tasks 进行数据同步
5. **联系人关联**: 将便签与手机联系人关联
6. **桌面小部件**: 在主屏幕直接查看和编辑便签
7. **搜索功能**: 快速检索便签内容

2.2.2 新增功能

1. **Markdown 富文本支持**：在便签中使用 Markdown 语法进行格式化，支持预览/编辑模式切换
2. **增强的导出分享**：支持将便签导出为纯文本、Markdown 格式或 HTML 格式

2.3 用户特点

- **学生群体**：需要记录课堂笔记、整理学习资料，具有较强的学习能力和接受新事物的意愿
- **职场人士**：需要记录工作事项、会议纪要，注重效率和专业性
- **知识工作者**：需要整理思路、创作内容，对格式化和数据互操作有较高要求
- **普通用户**：使用便签记录日常事务、购物清单等简单信息

2.4 运行环境

硬件环境：

- 处理器：ARMv7 或更高
- 内存：至少 512MB RAM
- 存储空间：至少 50MB 可用空间
- 屏幕：分辨率至少 480×800 像素

软件环境：

- 操作系统：Android 4.0（API Level 14）或更高版本
- 数据库：SQLite 3.0 或更高版本
- 依赖库：Android Support Library、Markwon Markdown 渲染库

2.5 约束和假定

约束：

- 必须遵循 Android 平台的设计规范和开发标准
- 必须向后兼容现有的小米便签数据格式
- 云同步功能依赖 Google Play Services，在中国大陆可能无法使用
- 应用大小应控制在合理范围内（不超过 20MB）

假定：

- 用户具备基本的 Android 设备操作能力
- 使用 Markdown 功能的用户了解基本的 Markdown 语法
- 设备具有稳定的网络连接（用于云同步功能）

3 系统需求

3.1 功能需求

3.1.1 原有功能的需求概述

原有的七大核心功能（便签管理、分类管理、提醒功能、云同步、联系人关联、桌面小部件、搜索功能）的详细需求参见原小米便签的设计文档。以下用例图（图1）展示了原有功能的整体结构。

原有功能的类图结构（图2）展示了核心模块的关系：

3.1.2 新功能 1：Markdown 富文本支持

功能描述：

用户可以在便签中使用 Markdown 语法编写格式化内容，包括标题、列表、代码块、粗体、斜体等。系统提供编辑模式和预览模式的切换，在预览模式下将 Markdown 文本渲染为格式化的显示效果。

用例描述：

- **用例名称：**使用 Markdown 编写便签
- **参与者：**用户
- **前置条件：**用户已打开便签编辑界面

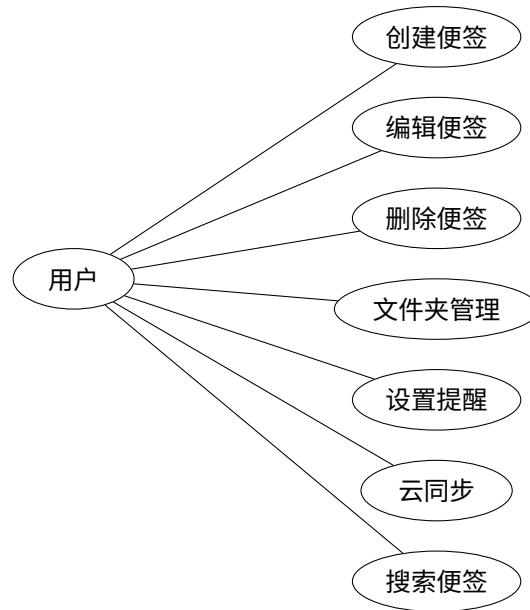


图 1: 原有功能用例图

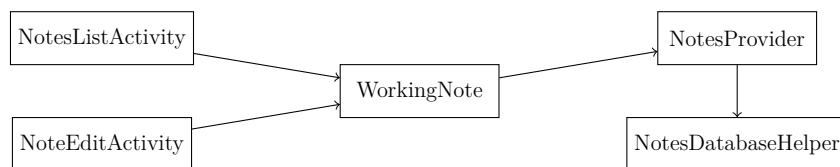


图 2: 原有功能类图（简化）

- **后置条件：**便签以 Markdown 格式保存

- **主要流程：**

1. 用户在编辑框中输入 Markdown 格式的文本
2. 用户点击”预览”按钮
3. 系统解析 Markdown 语法并渲染为格式化内容
4. 用户查看预览效果
5. 用户可点击”编辑”按钮返回编辑模式
6. 用户保存便签
7. 系统将 Markdown 原文保存到数据库

- **扩展流程：**

- 3a. 如果 Markdown 语法有误，系统仍按照最佳努力原则进行渲染

图3展示了 Markdown 功能的用例图：

图4展示了使用 Markdown 功能的活动图：

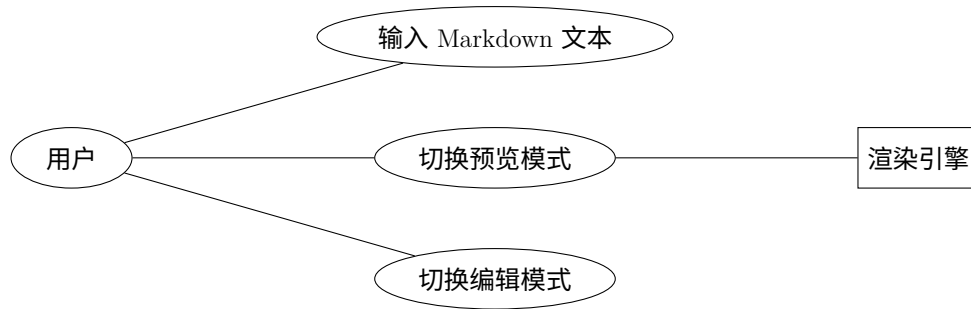


图 3: Markdown 功能用例图

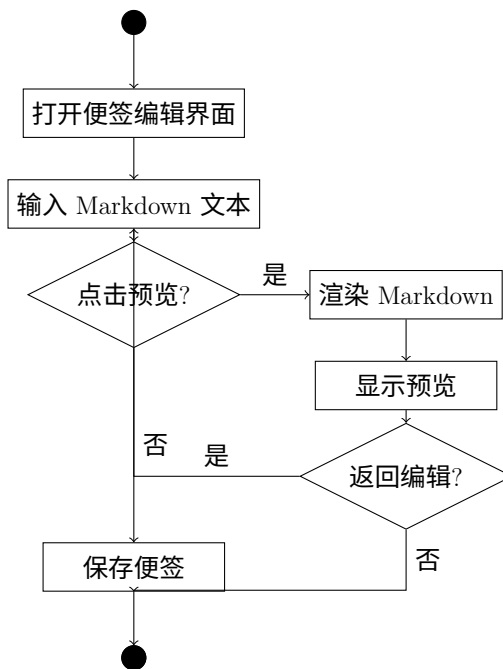


图 4: Markdown 功能活动图

图5展示了 Markdown 预览的序列图：

图6展示了编辑器的状态图：

图7展示了 Markdown 功能的类图：

性能需求：

- Markdown 渲染时间应小于 500ms（对于长度小于 10000 字符的文本）
- 编辑模式和预览模式切换应流畅，响应时间小于 200ms

数据需求：

- Markdown 原文存储在 data 表的 content 字段中
- MIME_TYPE 字段值保持为"text/note"，保证向后兼容
- 不需要单独存储渲染后的 HTML，每次预览时实时渲染

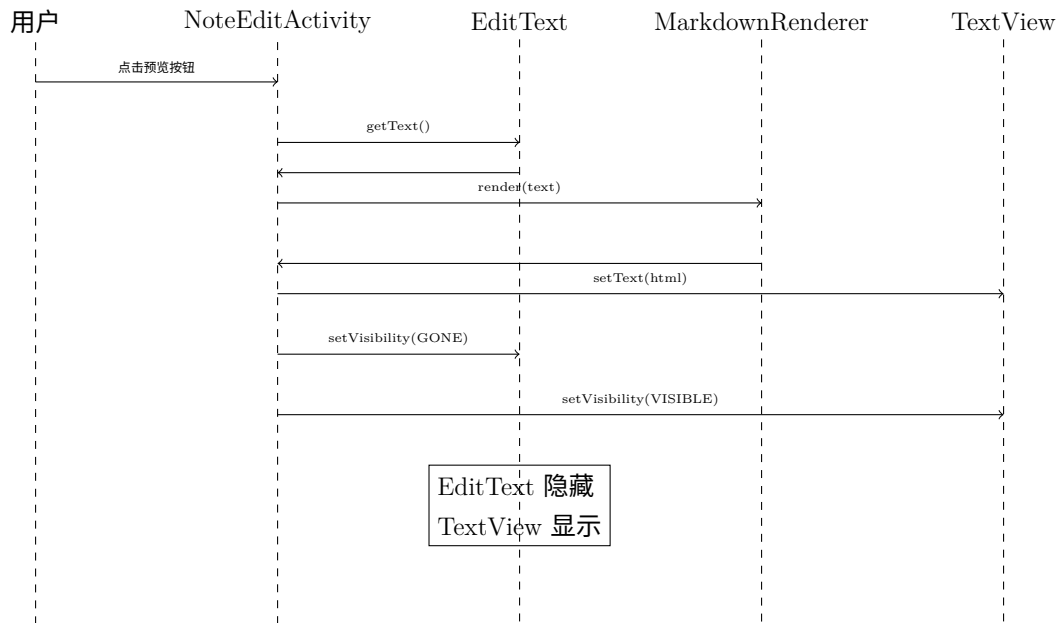


图 5: Markdown 预览序列图

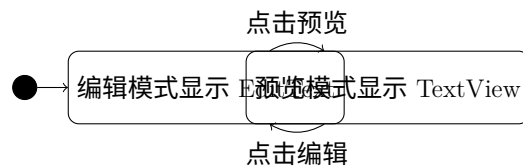


图 6: 编辑器状态图

3.1.3 新功能 2：增强的导出分享功能

功能描述：

用户可以将便签导出为多种格式（纯文本、Markdown、HTML），并通过 Android 系统的分享功能发送给其他应用或用户。这增强了智记便签与其他应用的互操作性。

用例描述：

- **用例名称：**分享便签为多种格式
- **参与者：**用户
- **前置条件：**用户已打开一条便签
- **后置条件：**便签以指定格式分享成功
- **主要流程：**
 1. 用户点击分享按钮
 2. 系统弹出格式选择对话框（纯文本/Markdown/HTML）

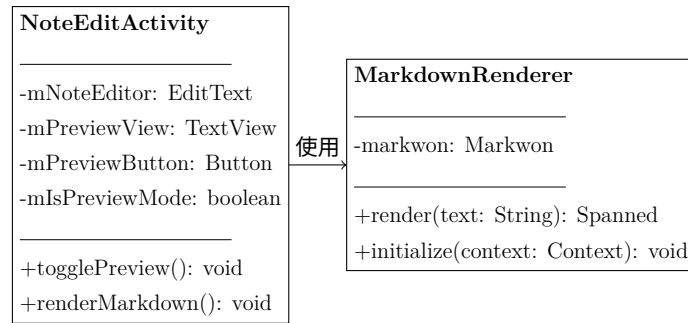


图 7: Markdown 功能类图

3. 用户选择导出格式
4. 系统根据选择的格式转换便签内容
5. 系统调用 Android 分享 Intent
6. 用户选择目标应用
7. 内容成功发送到目标应用

• 扩展流程：

- 4a. 如果选择 HTML 格式，系统将 Markdown 渲染为 HTML 并添加 CSS 样式
- 6a. 如果用户取消分享，返回便签编辑界面

图8展示了分享功能的用例图：

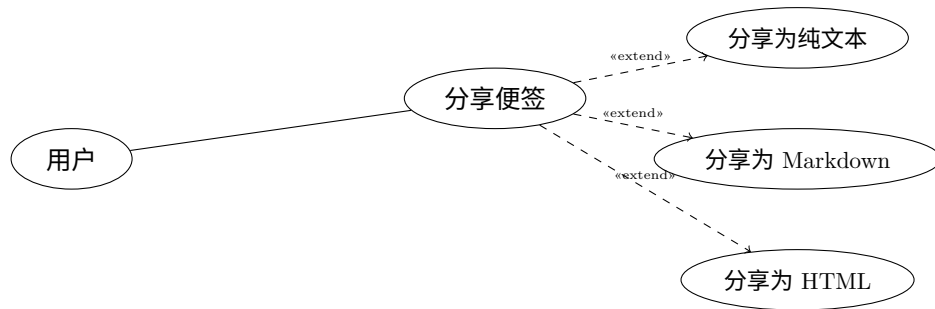


图 8: 增强分享功能用例图

图9展示了分享功能的活动图：

图10展示了分享为 HTML 格式的序列图：

图11展示了分享功能的类图：

性能需求：

- 格式转换时间应小于 1 秒

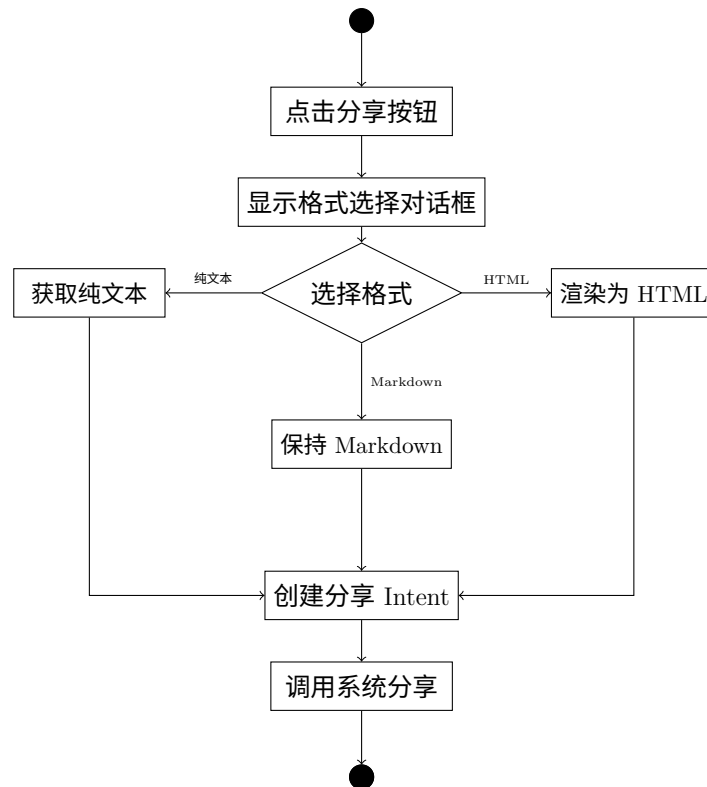


图 9: 增强分享功能活动图

- HTML 格式应包含合适的 CSS 样式，保证在目标应用中的可读性

兼容性需求：

- 纯文本格式应与原有分享功能保持一致
- 生成的 HTML 应符合 HTML5 标准
- Markdown 格式应遵循 CommonMark 规范

3.2 非功能需求

3.2.1 性能需求

- 响应时间：
 - 打开便签列表界面：小于 1 秒
 - 打开便签编辑界面：小于 1 秒
 - 保存便签：小于 500ms
 - Markdown 预览渲染：小于 500ms
 - 格式转换（分享功能）：小于 1 秒

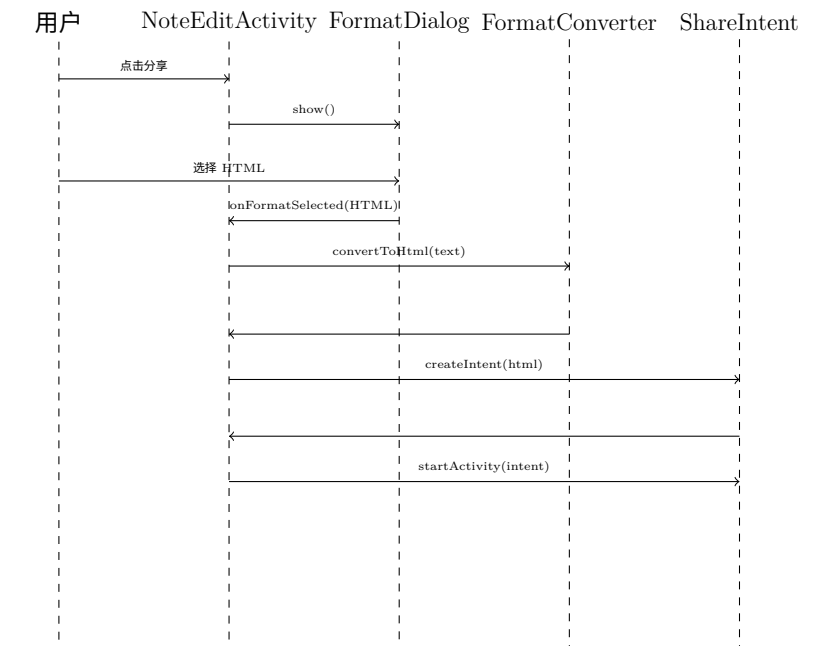


图 10: 分享为 HTML 格式序列图

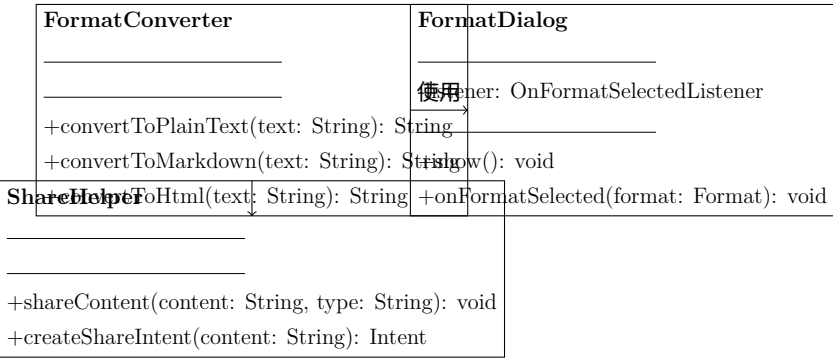


图 11: 增强分享功能类图

- 资源占用：
 - 应用内存占用：正常使用时小于 50MB
 - 数据库大小：1000 条便签应小于 10MB
 - APK 安装包大小：小于 20MB
- 并发性：支持多个便签同时打开（通过桌面小部件）
- 容量：支持至少 10000 条便签，单条便签最大支持 100000 字符

3.2.2 可靠性需求

- 可用性：系统可用性应达到 99.9%（不包括设备故障）
- 数据完整性：

- 所有数据操作必须通过 NotesProvider 进行，保证事务一致性
- 在应用异常退出时，已输入但未保存的内容应能够恢复
- 数据库操作失败时应有适当的错误处理和回滚机制

- **容错性：**

- Markdown 语法错误不应导致应用崩溃，应按最佳努力渲染
- 网络异常不应影响本地功能的使用
- 数据库升级失败时应保留原有数据

3.2.3 可维护性需求

- **代码质量：**

- 遵循 Android 开发最佳实践
- 保持代码注释完整，关键逻辑需有说明
- 类和方法遵循单一职责原则

- **可扩展性：**

- 使用 ContentProvider 机制，便于添加新的数据类型
- Markdown 渲染引擎应易于替换或升级
- 分享格式应易于扩展（如将来支持 PDF 格式）

- **可测试性：**

- 核心业务逻辑应与 UI 层分离，便于单元测试
- 数据库操作应可模拟，便于测试

3.2.4 安全性需求

- **数据安全：**

- 便签数据仅存储在应用私有目录，其他应用无法访问
- 云同步时使用 HTTPS 加密传输
- 用户认证信息使用 Android KeyStore 安全存储

- **权限控制：**

- 仅申请必要的系统权限
- 在使用敏感权限（如联系人）前向用户说明用途

3.2.5 可移植性需求

- 支持 Android 4.0 及以上版本（API Level 14+）
- 适配不同屏幕尺寸（手机、平板）
- 支持横屏和竖屏模式
- 遵循 Material Design 设计规范，保证在不同 Android 版本上的一致性

3.2.6 用户界面需求

- 易用性：
 - 界面设计简洁直观，新用户无需培训即可使用基本功能
 - 重要操作（如删除）应有确认提示
 - 提供必要的操作提示和帮助信息
- 一致性：
 - 遵循 Android 平台 UI 设计规范
 - 图标、颜色、字体使用保持一致
 - 交互方式符合用户习惯
- 响应性：
 - 界面操作应有及时的视觉反馈
 - 长时间操作应显示进度指示
 - 支持常见手势（如滑动删除）

4 数据模型

4.1 数据库设计

智记便签使用 SQLite 数据库存储数据，主要包含两个核心表：

4.1.1 note 表

存储便签的元数据：

字段名	类型	约束	说明
_id	INTEGER	PRIMARY KEY	便签唯一标识
type	INTEGER	NOT NULL	便签类型（普通/文件夹）
created_date	INTEGER	NOT NULL	创建时间（Unix 时间戳）
modified_date	INTEGER	NOT NULL	修改时间（Unix 时间戳）
sync_id	INTEGER		云同步 ID
parent_id	INTEGER		父文件夹 ID
alerted_date	INTEGER		提醒时间

4.1.2 data 表

存储便签的具体内容：

字段名	类型	约束	说明
_id	INTEGER	PRIMARY KEY	数据唯一标识
mime_type	TEXT	NOT NULL	内容类型（如 text/note）
note_id	INTEGER	NOT NULL	关联的便签 ID
content	TEXT	NOT NULL	实际内容（Markdown 原文）
data1	INTEGER		扩展字段 1
data2	INTEGER		扩展字段 2
data3	TEXT		扩展字段 3

4.2 数据流

图12展示了数据在系统中的流动：

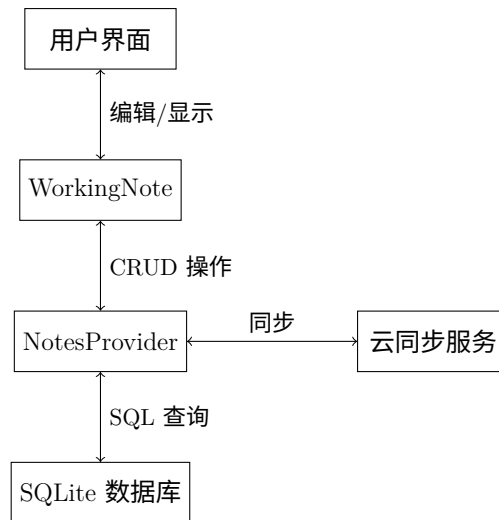


图 12: 数据流图

5 接口需求

5.1 用户界面

5.1.1 便签列表界面

- 显示便签列表，支持按文件夹分组
- 提供新建便签按钮
- 支持搜索功能
- 支持长按菜单（删除、移动等）

5.1.2 便签编辑界面

- 文本输入区域 (EditText)
- 预览区域 (TextView，默认隐藏)
- 工具栏包含：
 - 预览/编辑切换按钮
 - 分享按钮
 - 设置提醒按钮
 - 更多选项菜单
- Markdown 语法提示（可选）

5.1.3 格式选择对话框

- 三个选项：纯文本、Markdown、HTML
- 每个选项显示格式说明
- 确认和取消按钮

5.2 软件接口

5.2.1 Android 系统接口

- `AlarmManager`：用于提醒功能
- `ContentProvider`：提供数据访问接口
- `Intent`：用于分享功能和 `Activity` 间通信
- `AccountManager`：用于云同步认证

5.2.2 第三方库接口

- **Markwon 库**：用于 Markdown 渲染
 - `Markwon.create(context): Markwon`
 - `markwon.setMarkdown(textView, markdown): void`

5.3 硬件接口

- **网络接口**：用于云同步功能，支持 WiFi 和移动数据
- **存储**：访问设备内部存储，不需要外部存储权限
- **通知**：使用系统通知机制显示提醒

6 质量属性

6.1 可用性

- 新用户应能在 5 分钟内完成基本操作（创建、编辑、保存便签）
- Markdown 功能应提供语法帮助文档
- 重要操作应有撤销机制

6.2 可靠性

- 平均无故障时间 (MTBF): 大于 100 小时
- 数据丢失率: 小于 0.01%
- 云同步成功率: 大于 99% (在网络正常情况下)

6.3 性能

- 应用启动时间: 小于 2 秒 (首次启动)
- 内存占用: 峰值小于 100MB
- 电池消耗: 后台待机时每小时消耗小于 1%

6.4 可维护性

- 代码注释覆盖率: 大于 30%
- 模块耦合度: 低耦合, 各模块职责清晰
- 版本升级: 支持数据库平滑升级, 不丢失用户数据

7 项目约束

7.1 时间约束

- 设计阶段: 2 周
- 开发阶段: 6 周
- 测试阶段: 2 周
- 总周期: 10 周

7.2 成本约束

- 开发成本: 基于开源项目, 主要为人力成本
- 第三方库: 使用开源免费库, 无额外费用
- 云服务: 依赖 Google 服务, 无额外服务器成本

7.3 技术约束

- 必须使用 Java 语言开发
- 必须兼容 Android 4.0 及以上版本
- 必须保持与原小米便签数据格式的兼容性
- 应用大小不超过 20MB

8 附录

8.1 术语表

术语	定义
便签	用户创建的一条记录，包含文本内容和元数据
文件夹	用于组织便签的容器，本质上也是一种特殊类型的便签
工作便签	WorkingNote，内存中的便签对象，用于编辑和临时存储
渲染	将 Markdown 文本转换为格式化显示的过程
MIME 类型	标识数据类型的字符串，如"text/note"
云同步	将本地数据与云端服务（Google Tasks）同步的过程

8.2 参考模型

本系统采用经典的 MVC（Model-View-Controller）架构模式：

- **Model**：WorkingNote、Note 数据模型
- **View**：NotesListActivity、NoteEditActivity 等 UI 组件
- **Controller**：Activity 中的业务逻辑处理

同时采用 Android 平台推荐的数据访问模式：

- 使用 ContentProvider 作为数据访问的唯一入口
- 使用 AsyncQueryHandler 进行异步数据查询
- 使用 Cursor 管理数据库查询结果

8.3 更新记录

版本	日期	作者	更新内容
1.0	2025 年 11 月 5 日	开发团队	初始版本，包含所有功能需求