

指导书

第一次迭代：

在本次迭代中需要明确待开发软件系统的目标，从中引导出软件系统的概要需求，并注意要从创意性、可行性等方面持续构思、讨论、改进和完善软件需求，完成前景文档的撰写。

1、实践内容

- (1) 构思与创意项目，确定选题，调研项目背景与商机，分析问题及根源，明确产品定位；
- (2) 识别项目的干系人，并进行详细说明；
- (3) 识别项目的约束；
- (4) 获取常用术语；
- (5) 识别需求来源，收集需求；
- (6) 产品定位；
- (7) 分析产品特性(功能)与风险，划分特性优先级，确定迭代开发计划；
- (8) 定义质量范围；
- (9) 撰写前景文档。

2、提交前景文档，前景文档撰写提纲与规范请参考教材“4S系统的前景文档”案例，主要内容包括：

- (1) 简介，包括目的、范围、术语表、参考资料、概述等；
- (2) 定位，包括商机、问题说明、产品定位；
- (3) 项目干系人和用户说明；
- (4) 产品概述，包括总体效果、功能概要、成本与定价、许可与安装等；
- (5) 产品特性（重点描述），给出功能层次图；
- (6) 约束；
- (7) 质量范围，包括性能、可靠性、易用性、可维护性等；
- (8) 特性优先级，迭代安排与版本提交计划。

第二次迭代：

采用UML进行需求分析建模，形成以用例图、时序图、类图为核心内容的分析模型，并完成需求分析文档。

1、实践内容

- (1) 完善前景文档，形成带用例分析的需求分析文档；
- (2) 系统用例建模：对项目的干系人需求进行分析，建立用例模型，优化后的用例模图，用包含、扩展和泛化关系组织优化用例模型；
- (3) 建立概念模型：识别系统的关键概念类，画出类图；
- (4) 分析类识别：以用例图为基础，根据MVC设计模式识别出边界类、控制类、实体类；

(5) 构建交互模型：通过用例的事件流分析与分析类职责分配，构建用例实现的一组分析类对象交互的时序图。

(6) 构建类图：以时序图为基础，同时辅助通信图，完成类图的构建

2、提交需求分析文档，对前景文档进行精化，同时补充需求分析建模的相关内容，形成需求分析文档。主要包括：

(1) 用例图，给出核心用例的用例规约，其格式参考教材“提交购车单”用例规约表，包括事件流与可选字段；

(2) 概念模型描述，给出类图/静态结构图；

(3) 对具有复杂逻辑事件流用例给出活动图；

(4) 交互行为描述，给出时序图和通信图；

(5) 完成用例实现的分析类所构成的视图，即VOPC类图；

(6) 带用例分析的需求分析文档。

第三次迭代：

在第三次迭代中完成系统的软件设计，包括架构设计、界面设计、构件级设计以及相关测试用例的设计等。

1、实践内容

(1) 识别设计元素，确定架构风格，设计整体结构；

(2) 确定界面风格，完成人机界面设计和系统原型；

(3) 识别子系统，绘制包图；

(4) 确定软件物理架构，绘制部署图；

(5) 构件级设计，包括子系统的设计，完善类图；

(6) 类的设计，优化类图；

(7) 主要功能的测试用例设计；

(8) 形成软件设计文档。

2、需要提交软件设计文档，主要包括：

(1) 概述；

(2) 架构描述，给出包图；

(3) 部署图；

(4) 系统原型界面；

(5) 类图；

(6) 测试用例列表；

(7) 软件设计文档与原型演示视频。

模板

软件设计文档

1. 概述

本文档旨在详细描述系统的软件设计，包括架构设计、界面设计、构件级设计以及相关测试用例的设计等方面。

2. 架构描述

2.1 包图

在此部分，我们将展示系统的包图，明确系统中的主要模块和它们之间的关系。

3. 部署图

3.1 系统原型界面

展示系统的用户界面设计和原型，以便清晰地展示系统的外观和交互。

4. 类图

4.1 类图

展示系统的类图，包括各个类的属性、方法以及它们之间的关系。

5. 测试用例列表

5.1 主要功能的测试用例设计

在此部分，我们列出了针对系统主要功能的测试用例，包括输入、预期输出和执行步骤。

测试用例编号	测试目标	输入数据	期望输出	执行步骤
TC001	用户登录	用户名和密码	登录成功	1. 打开登录界面
TC002	创建新记录	相关数据	记录成功创建	1. 进入记录创建页面
TC003	数据查询	查询条件	返回符合条件的数据	1. 进入查询页面
...

6. 软件设计文档与原型演示视频

6.1 软件设计文档

在此部分，我们提供了软件设计文档的详细内容，包括架构描述、部署图、类图等。

[附录：软件设计文档链接]

6.2 原型演示视频

提供系统原型演示的视频链接，以便项目相关人员能够直观地了解系统的外观和基本操作。

[附录：原型演示视频链接]

附录：

[软件设计文档链接]

[原型演示视频链接]

版本历史

版本号	作者	日期	描述
1.0	[作者名]	YYYY-MM-DD	初稿软件设计文档
1.1	[作者名]	YYYY-MM-DD	修订根据反馈的修改
...

此模板可以根据具体项目的需要进行调整和扩展。确保文档详尽全面，能够清晰地传达系统的设计和实现细节。

~~

概述

本文档旨在详细描述小米便签的软件设计，包括架构设计、界面设计、构件级设计以及相关测试用例的设计等方面。在版本1.0的基础上，迭代至版本2.0，完成第九次迭代，添加导入照片功能，使用户可以在便签中插入照片。

架构描述，给出包图

架构设计

基于**分层架构风格**的架构设计：

1、表示层

表示层主要关注用户界面和用户交互。

用户界面与交互：在表示层中，ui 包和 widget 包负责处理用户界面和交互。`NotesListActivity` 充当应用的主界面，负责展示便签列表以及处理文件夹列表的相关逻辑。各个活动和界面组件负责直接与用户进行交互，处理用户输入并调用业务逻辑层的相应功能。

界面元素：widget 包下的类，如 `NotewidgetProvider` 和 `NotewidgetProvider_2x`，提供了桌面挂件的实现，用于在桌面上展示便签信息，增强用户体验。这些元素与用户的直观互动形成了应用程序的外观。

- **ui 包**
 - `AlarmAlertActivity`: 闹铃提醒界面

- `AlarmInitReceiver`: 闹铃提醒启动消息接收器
- `AlarmReceiver`: 闹铃提醒接收器
- `DateTimePicker`: 设置提醒时间的部件
- `DateTimePickerDialog`: 设置提醒时间的对话框界面
- `DropDownMenu`: 下拉菜单界面
- `FoldersListAdapter`: 文件夹列表链接器（链接数据库）
- `NoteEditActivity`: 便签编辑活动
- `NoteEditText`: 便签的文本编辑界面
- `NoteItemData`: 便签项数据
- `NotesListActivity`: 主界面，实现处理文件夹列表的活动
- `NotesListAdapter`: 便签列表链接器（链接数据库）
- `NotesListItem`: 便签列表项
- `NotesPreferenceActivity`: 便签同步的设置界面
- `InsertImage`: 插入图片的选项
- **widget 包**
 - `NoteWidgetProvider`: 桌面挂件
 - `NoteWidgetProvider_2x`: 2倍大小的桌面挂件
 - `NoteWidgetProvider_4x`: 4倍大小的桌面挂件

2、业务逻辑层

业务逻辑层包含了处理业务规则和逻辑的类。

任务同步与管理: 业务逻辑层主要包含 `gtask` 包，其中的类负责处理与任务同步和管理相关的业务逻辑。`Task` 类将创建、更新和同步动作封装成 JSON 对象，用于在本地和远程同步任务的结点内容。

`TaskList` 类将多个 `Task` 组织成同步任务列表进行有效管理。

异常处理: `exception` 包下的类，如 `ActionFailureException` 和 `NetworkFailureException`，处理业务逻辑层可能遇到的异常情况，确保系统在面对错误时能够进行适当的处理，提高系统的稳定性。

- **gtask 包**
 - `MetaData`: 关于同步任务的元数据
 - `Node`: 同步任务的管理结点，用于设置、保存同步动作的信息
 - `SqlData`: 数据库中基本数据，方法包括读取数据、获取数据库中数据、提交数据到数据库
 - `SqlNode`: 数据库中便签数据，方法包括读取便签内容、从数据库中获取便签数据、设置便签内容、提交便签到数据库
 - `Task`: 同步任务，将创建、更新和同步动作包装成 JSON 对象，用本地和远程的 JSON 对结点内容进行设置，获取同步信息，进行本地和远程的同步
 - `TaskList`: 同步任务列表，将 `Task` 组织成同步任务列表进行管理
- **exception 包**
 - `ActionFailureException`: 动作失败异常
 - `NetworkFailureException`: 网络失败异常

3、数据访问层

数据访问层主要负责与数据存储交互，包括数据库操作和远程数据访问。

数据存储与检索：数据访问层通过 data 包和 remote 包进行数据库操作和远程数据访问。data 包中的类负责联系人和便签数据库的存储和检索，而 remote 包下的类则处理与远程任务同步和数据访问相关的功能。

数据库辅助和提供：NotesDatabaseHelper 类是一个数据库帮助类，用于辅助创建、处理便签数据库的条目。NotesProvider 类是一个便签信息提供类，为上层业务逻辑提供对便签数据的访问接口。

- data 包

- contact: 联系人数据库
- Notes: 便签数据库，用于记录便签相关属性和数据
- NotesDatabaseHelper: 数据库帮助类，用于辅助创建、处理数据库的条目
- NotesProvider: 便签信息提供类
- Images: 图片提供类，用于面向图片管理软件的接口

- remote 包

- GTaskAsyncTask: GTask异步任务，方法包括任务同步和取消，显示同步任务的进程、通知和结果
- GTaskClient: GTask客户端，提供登录Google账户，创建任务和任务列表，添加和删除结点，提交、重置更新更新，获取任务列表等功能
- GTaskManager: GTask管理者，提供同步本地和远端的任务，初始化任务列表，同步内容、文件夹，添加、更新本地和远端结点，刷新本地同步任务ID等功能
- GTaskSyncService: GTask同步服务，用于提供同步服务（开始、取消同步），发送广播

4、其他层次

- model 包

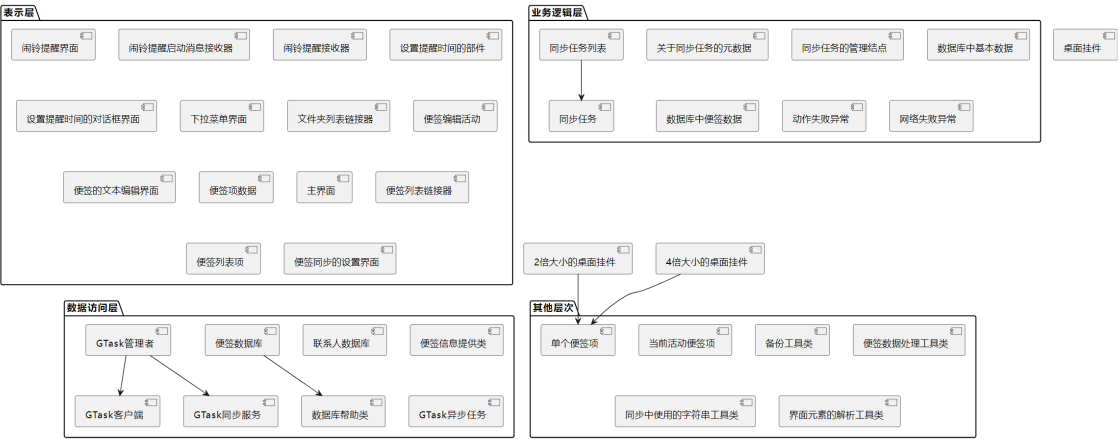
- Note: 单个便签项
- WorkingNote: 当前活动便签项

- tool 包

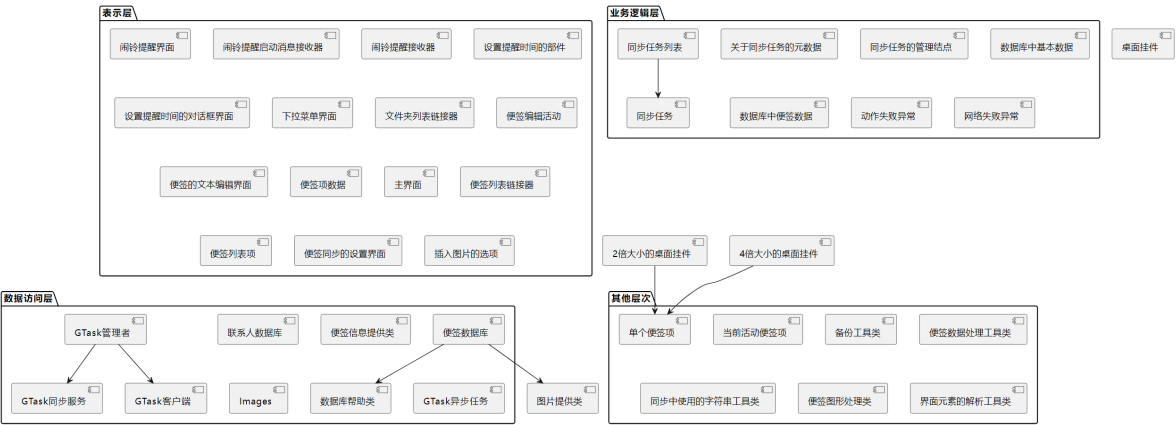
- BackupUtils: 备份工具类，用于数据备份读取、显示
- DataUtils: 便签数据处理工具类，封装如查找、移动、删除数据等操作
- GTaskStringUtils: 同步中使用的字符串工具类，为 jsonObject 提供 string 对象
- ImageUtils: 便签图形处理类，封装如图片链接、渲染、删除等操作
- ResourceParser: 界面元素的解析工具类，利用 R.java 这个类获取资源供程序调用

这种分层架构有助于代码的组织和维护，使各个层次的功能单一而清晰。

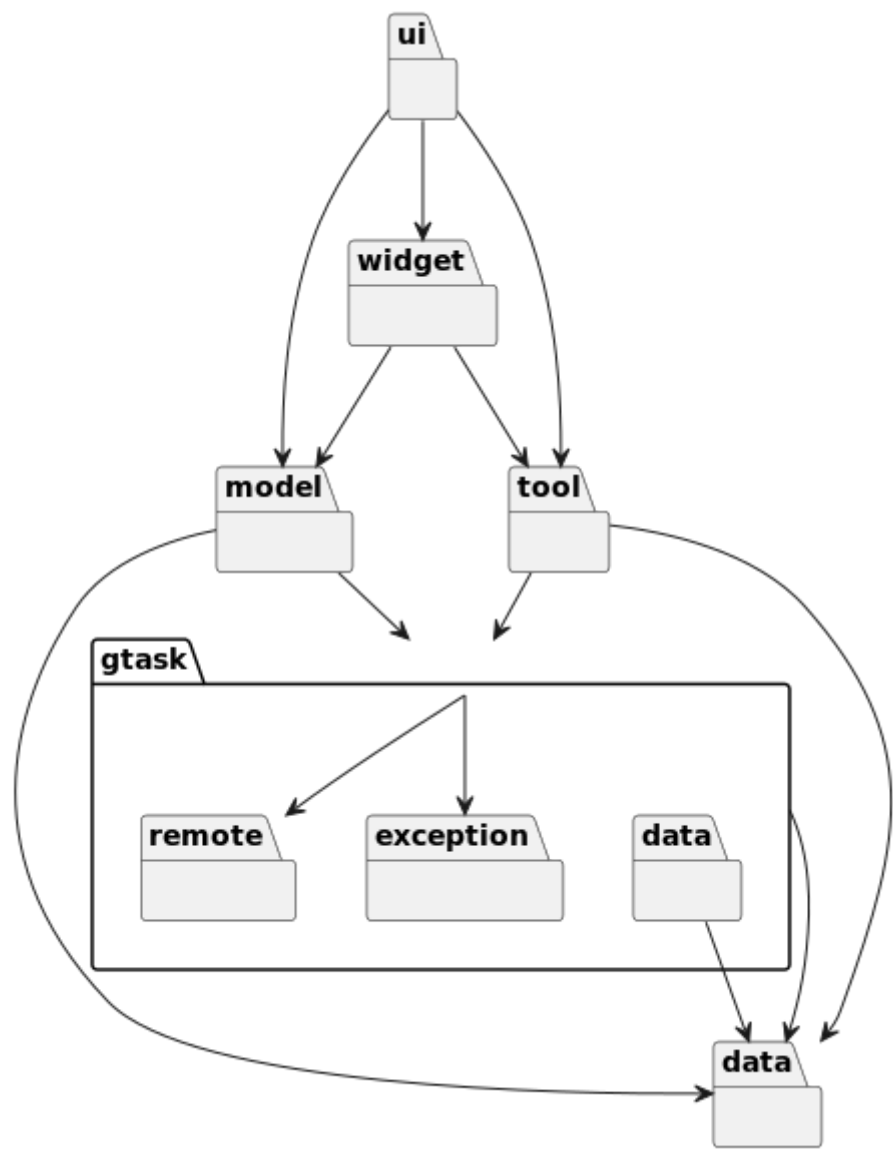
5-1、架构设计图（未加入导入照片功能）



5-2、架构设计图（加入导入照片功能）

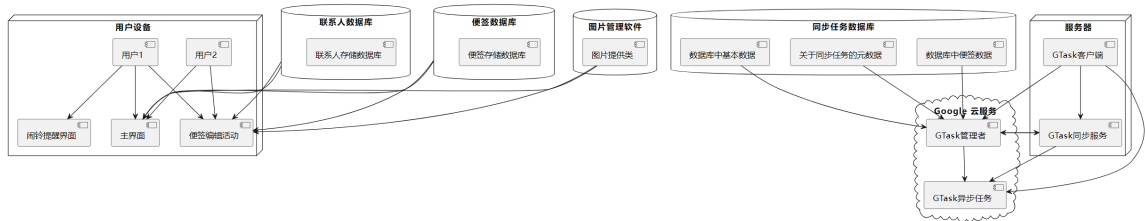


软件包图



部署图

多个用户设备通过界面与系统进行交互，同时与服务器和云服务进行数据同步。数据库节点存储了不同类型的数据，而服务器和云服务负责协调和同步数据。如图：



1. 用户设备节点：

- 包括用户在设备上使用的界面和应用。
- [用户1] 和 [用户2] 表示两个不同的用户。
- [闹钟提醒界面] 用于提供闹钟提醒功能。

- [主界面] 是用户与系统进行交互的主要入口。
- [便签编辑活动] 是用户编辑便签的界面。

2. 数据库节点：

- 包括联系人数据库、便签数据库、同步任务数据库和图片管理软件的数据。
- [联系人存储数据库] 用于存储联系人数据。
- [便签存储数据库] 用于存储便签相关的属性和数据。
- [关于同步任务的元数据]、[数据库中基本数据] 和 [数据库中便签数据] 分别用于存储同步任务的元数据、基本数据和便签数据。
- [图片提供类] 是图片管理软件的数据。

3. 服务器节点：

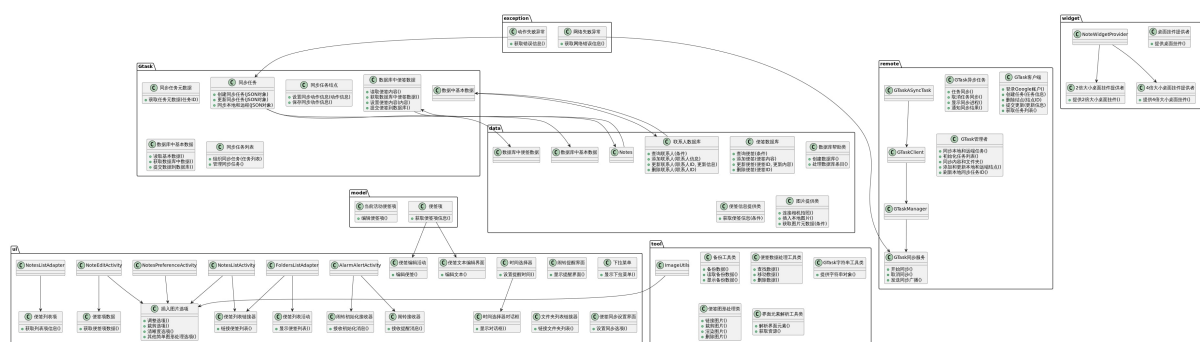
- 包括运行在服务器上的 GTask 客户端和 GTask 同步服务。
- [GTask客户端] 用于处理用户的 Google 账户登录、任务和任务列表的创建、结点的添加和删除等功能。
- [GTask同步服务] 提供同步服务，包括开始和取消同步，并通过广播通知相关事件。

4. Google 云服务节点：

- 代表与 Google 云服务相关的组件。
- [GTask异步任务] 用于处理异步任务，包括任务同步和取消等操作。
- [GTask管理者] 是管理同步任务的组件，负责初始化任务列表、同步内容和文件夹、添加、更新本地和远端结点等。
-

系统原型界面*

类图



1. data 包：

- **联系人数据库 (ContactDatabase)**：管理联系人信息的数据库，被便签编辑活动和主界面使用。
- **便签数据库 (NotesDatabase)**：管理便签信息的数据库，被便签编辑活动和主界面使用。
- **数据库帮助类 (NotesDatabaseHelper)**：提供创建和处理数据库条目的辅助方法。
- **便签信息提供类 (NotesProvider)**：提供获取便签信息的方法。
- **图片提供类 (Images)**：提供获取图片信息的接口，面向图片管理软件。

2. Gtask 包:

- **同步任务元数据 (MetaData)** : 提供获取同步任务元数据的方法。
- **同步任务结点 (Node)** : 用于设置和保存同步任务的管理结点信息。
- **数据库中基本数据 (SqlData)** : 提供读取、获取和提交基本数据到数据库的方法。
- **数据库中便签数据 (SqlNode)** : 提供读取、获取、设置和提交便签数据到数据库的方法。
- **同步任务 (Task)** : 将创建、更新和同步动作包装成 JSON 对象, 用于本地和远程同步。
- **同步任务列表 (TaskList)** : 组织和管理同步任务列表。

3. exception 包:

- **动作失败异常 (ActionFailureException)** : 表示动作执行失败的异常。
- **网络失败异常 (NetworkFailureException)** : 表示网络操作失败的异常。

4. remote 包:

- **GTask异步任务 (GTaskASyncTask)** : 异步执行同步任务, 显示同步进程、通知和结果。
- **GTask客户端 (GTaskClient)** : 提供与 Google 云服务交互的方法, 包括登录、创建任务等。
- **GTask管理者 (GTaskManager)** : 管理同步任务, 提供同步本地和远端任务、初始化任务列表等功能。
- **GTask同步服务 (GTaskSyncService)** : 提供同步服务, 包括开始和取消同步, 发送同步广播。

5. model 包:

- **便签项 (Note)** : 表示单个便签项, 提供获取便签项信息的方法。
- **当前活动便签项 (WorkingNote)** : 表示当前编辑的便签项, 提供编辑便签项的方法。

6. tool 包:

- **备份工具类 (BackupUtils)** : 提供数据备份、读取和显示的工具类。
- **便签数据处理工具类 (DataUtils)** : 封装便签数据的查找、移动、删除等操作。
- **GTask字符串工具类 (GTaskStringUtils)** : 提供同步中使用的字符串工具。
- **便签图形处理类 (ImageUtils)** : 处理便签图形操作, 包括链接、渲染、删除和插入本地图片等。
- **界面元素解析工具类 (ResourceParser)** : 解析界面元素, 获取资源供程序调用。

7. ui 包:

- 包含多个与用户界面相关的类, 如闹铃提醒界面、便签编辑活动、时间选择器等。
- 这些类用于显示和处理用户界面的各种操作。

8. widget 包:

- 包含桌面挂件提供者类, 用于提供不同大小的桌面挂件。

测试用例列表*

软件设计文档与原型演示视频*

\