

项目编号：2005C56

密 级：

笔记&备忘概要设计

(版本号 NR01 V1.01)

笔记&备忘项目组

2016 年 4 月 26 日

审批记录

文档名称：笔记&备忘概要设计	现行版本号：NR01 V1.01
拟制人：刘慧琦、夏昊	日期：2016-4-26
审核人：	日期：2016-4-30
项目负责人审批意见： 签字： 日期：	
管理专家审批意见： 签字： 日期：	
用户代表审批意见： 签字： 日期：	

变更记录

变更 编号	版本号	日期	章节/段落/行 或图/表号	变更 状态	变更简单描述	审核人	批准人
1	V1.00	2016- 4-26		A	概要设计初稿完成		

变更状态：A – 增加 M – 修改 D – 删除

目录

项目编号：2005C56.....	0
1 引言.....	6
1.1 编写目的.....	6
1.2 项目背景.....	6
1.3 术语.....	6
1.4 参考资料.....	6
2 任务概述.....	6
2.1 目标.....	6
2.2 开发与运行环境.....	7
2.2.1 开发环境的配置.....	7
2.2.2 运行环境的配置.....	7
2.2.3 测试环境的配置.....	7
2.3 需求概述.....	8
2.4 条件与限制.....	8
3 总体设计.....	8
3.1 软件描述.....	8
3.2 处理流程.....	9
3.2.1 CHULI.001 备忘处理流程.....	11
3.2.2 CHULI.002 创建笔记处理流程.....	11
3.2.3 CHULI.003 添加图片处理流程.....	11
3.2.4 CHULI.004 添加画图处理流程.....	11
3.2.5 CHULI.005 添加标签处理流程.....	11
3.2.6 CHULI.006 添加时间处理流程.....	12
3.2.7 CHULI.007 删除笔记处理流程.....	12
3.2.8 CHULI.008 查询笔记处理流程.....	12
3.2.9 CHULI.009 登录处理流程.....	12
3.2.10 CHULI.010 导出处理流程.....	12
3.2.11 CHULI.011 筛选处理流程.....	12
3.3 功能结构设计.....	13
3.3.1 MODULE.001 主界面功能结构.....	16
3.3.2 MODULE.002 编辑笔记功能结构.....	16
3.3.3 MODULE.003 添加图片功能结构.....	16
3.3.4 MODULE.004 添加画图功能结构.....	16
3.3.5 MODULE.005 添加标签功能结构.....	17
3.3.6 MODULE.006 删除功能结构.....	17
3.3.7 MODULE.007 查询功能结构.....	17
3.3.8 MODULE.008 分类功能结构.....	17
3.3.9 MODULE.009 查询展示功能结构.....	17
3.3.10 MODULE.010 备忘属性功能结构.....	17
3.3.11 MODULE.011 提醒时间功能结构.....	17
3.3.12 MODULE.012 计时功能结构.....	17
3.3.13 MODULE.013 已完成界面功能结构.....	17

3.3.14 MODULE.014 登录功能结构.....	18
3.3.15 MODULE.015 导出功能结构.....	18
3.3.16 MODULE.016 本地笔记保存功能结构.....	18
3.3.17 MODULE.017 笔记上传功能结构.....	18
3.3.18 MODULE.018 笔记下载功能结构.....	18
3.3.19 MODULE.019 登录请求功能结构.....	18
3.3.20 MODULE.020 笔记同步功能结构.....	18
3.3.21 MODULE.021 笔记导出功能结构.....	18
3.3.22 MODULE.022 数据库更新功能结构.....	18
3.3.23 MODULE.023 解包功能结构.....	18
3.4 功能需求与程序代码的关系.....	19
5 接口设计.....	20
5.1 外部接口.....	20
5.1.1 MODULE.003 添加图片功能结构 \leftrightarrow 文件系统.....	20
5.1.2 MODULE.012 计时功能结构 \leftrightarrow 客户端系统.....	20
5.1.3 MODULE.017 笔记上传功能结构 MODULE.018 笔记下载功能结构 \leftrightarrow 客户端系统.....	21
5.2 内部接口.....	21
5.2.1 MODULE.001 主界面功能结构 \leftrightarrow 其它功能结构.....	21
5.2.2 MODULE.002 编辑笔记功能结构 \leftrightarrow 其它功能结构.....	22
5.2.3 MODULE.007 查询功能结构 \leftrightarrow 其它功能结构.....	22
5.2.4 MODULE.010 备忘属性功能结构 \leftrightarrow 其它功能结构.....	23
5.2.5 MODULE.023 解包功能结构 \leftrightarrow 其它功能结构.....	23
6 数据结构设计.....	24
6.1 逻辑结构设计.....	24
6.1.1 主控中心主要数据结构.....	24
6.2 物理结构设计.....	26
6.3 数据结构与程序代码的关系.....	26
7 数据库设计.....	27
7.1 数据库环境说明.....	27
7.2 数据库的命名规则.....	27
7.3 逻辑设计.....	28
7.3.1 逻辑设计的 E-R 图.....	28
7.3.2 数据库逻辑结构设计.....	28
7.4 物理设计.....	29
7.5 安全性设计.....	29
7.6 数据库管理与维护说明.....	31
8 界面设计.....	32
8.1 安卓版界面.....	32
8.1.1 登陆界面.....	32
8.1.2 主界面.....	33
8.1.3 笔记编辑界面.....	34
8.2 ios 版界面.....	35
8.2.1 登陆界面.....	35

8.2.2 主界面.....	36
8.2.3 笔记编辑界面.....	37
8.3 web 版界面.....	38
8.3.1 登陆界面.....	38
8.3.2 主界面.....	39
8.3.3 笔记编辑界面.....	40
9 出错处理设计.....	40
9.1 出错输出信息.....	40
9.1.1 用户登录.....	40
9.1.2 笔记同步.....	40
9.1.3 笔记编辑.....	41
9.1.4 查找.....	41
9.1.5 设置备忘.....	41
9.2 出错处理对策.....	41
9.2.1 用户登录.....	41
9.2.2 笔记同步.....	41
9.2.3 笔记编辑.....	42
9.2.4 查找.....	42
9.2.5 设置备忘.....	42
10 安全保密设计.....	42
11 维护设计.....	42

1 引言

1.1 编写目的

在本阶段的前一阶段，也就是需求分析阶段中，已经对本软件系统的功能需求做了详细的阐述。本阶段将会在需求分析阶段的基础上对笔记&备忘软件系统做进一步的概要设计，主要包括系统的操作处理流程设计、系统接口设计、出错处理设计和数据库结构的设计等。以上系统模块的设计将结合需求分析阶段的功能需求，把各模块间的关系给建立起来，从而完成整个系统的概要设计需求。

另外，在下一阶段的详细设计中，本阶段的概要设计将作为参考，以方便完成整个系统的设计工作。

1.2 项目背景

为了能高效地工作、学习，市场上出现了各种笔记备忘类软件，这些软件替代了传统的纸质笔记本，以其方便、便携、共享性受到用户青睐。一些常用系统，比如 iOS、MIUI、Mac OS 也将笔记和自己的日历、短信功能结合起来，大大提升了使用体验和工作效率。

市场上现有不少笔记备忘软件：Evernote、Wunderlist、MIUI 的笔记和便签、iOS 的备忘录提醒事项，这些软件的功能、体验很丰富，同时笔记类应用正变的越来越“重”：功能上，越来越复杂、交互上越来越难，用户体验则显得过于繁琐。

本软件致力于最精致地实现笔记和备忘的功能，专注于专一、纯粹。

1.3 术语

【列出本文档中所用到的专门术语的定义和外文缩写的原词组。】

缩写、术语	解 释

1.4 参考资料

【1】

2 任务概述

2.1 目标

设计目标：完成对笔记&备忘软件在 Android、iOS、Web Browser 平台的概要设计。

2.2 开发与运行环境

2.2.1 开发环境的配置

类别	标准配置	最低配置
计算机硬件	基于 Intel×86 结构的 CPU， 主频≥2.4GHz 内存≥4G 硬盘至少有 500G	基于 Intel×86 结构的 CPU， 主频≥2GHz 内存≥1G 硬盘至少有 100G
计算机软件	Windows 10 OS X v10.10 Yosemite Android Studio 2.0 Android SDK tools r24.4.1 Xcode 4 WAMP Server Eclipse Mars	Windows 7 OS X v10.10 Yosemite Android Studio 2.0 Android SDK tools r24.4.1 Xcode 4 WAMP Server Eclipse Mars
服务器	Debian 8: 8.4 released LAMP Server MySQL 5.7.12	Debian 8: 8.4 released LAMP Server MySQL 5.7.12
网络通信	至少要有一块可用网卡，能 运行 IP 协议栈即可。	至少要有一块可用网卡，能 运行 IP 协议栈即可。
数据库	采用 Microsoft SQL 2000 数 据库系统	采用 Microsoft SQL 2000 数 据库系统

2.2.2 运行环境的配置

类别	标准配置	最低配置
服务器	Debian 8: 8.4 released	Debian 8: 8.4 released
数据库	MySQL 5.7.12	MySQL 5.7.12
Android 设备	Android – 5.0 Lollipop 及以 上设备	Android – 5.0 Lollipop 及以 上设备
iOS 设备	iPhone iOS 9.0 及以上设备	iPhone iOS 9.0 及以上设备
浏览器	Chrome	Chrome

2.2.3 测试环境的配置

类别	标准配置	最低配置
服务器	Debian 8: 8.4 released	Debian 8: 8.4 released
数据库	MySQL 5.7.12	MySQL 5.7.12
Android 设备	Samsung Galaxy S6	Samsung Galaxy S6
iOS 设备	iPhone 6	iPhone 6
浏览器	Chrome	Chrome

2.3 需求概述

为了能高效地工作、学习，市场上出现了各种笔记备忘类软件，Evernote、Wunderlist、MIUI 的笔记和便签、iOS 的备忘录提醒事项等，大大提升了使用体验和工作效率。但这些应用变得越来越“重”：功能上，越来越复杂、交互上越来越难，用户体验则显得过于繁琐。本软件致力于最精致地实现笔记和备忘的功能，专注于专一、纯粹。

2.4 条件与限制

技术条件：技术开发人员不多。

资金状况：资金有限，需要考虑着重点。

开发环境：Android 设备更新迭代很快，产品的层次也参差不齐，需要考虑的版本问题很繁杂；各个浏览器对于 HTML 的标准支持各不相同，多设备支持比较困难。

时间限制：开发时间紧张。

3 总体设计

3.1 软件描述

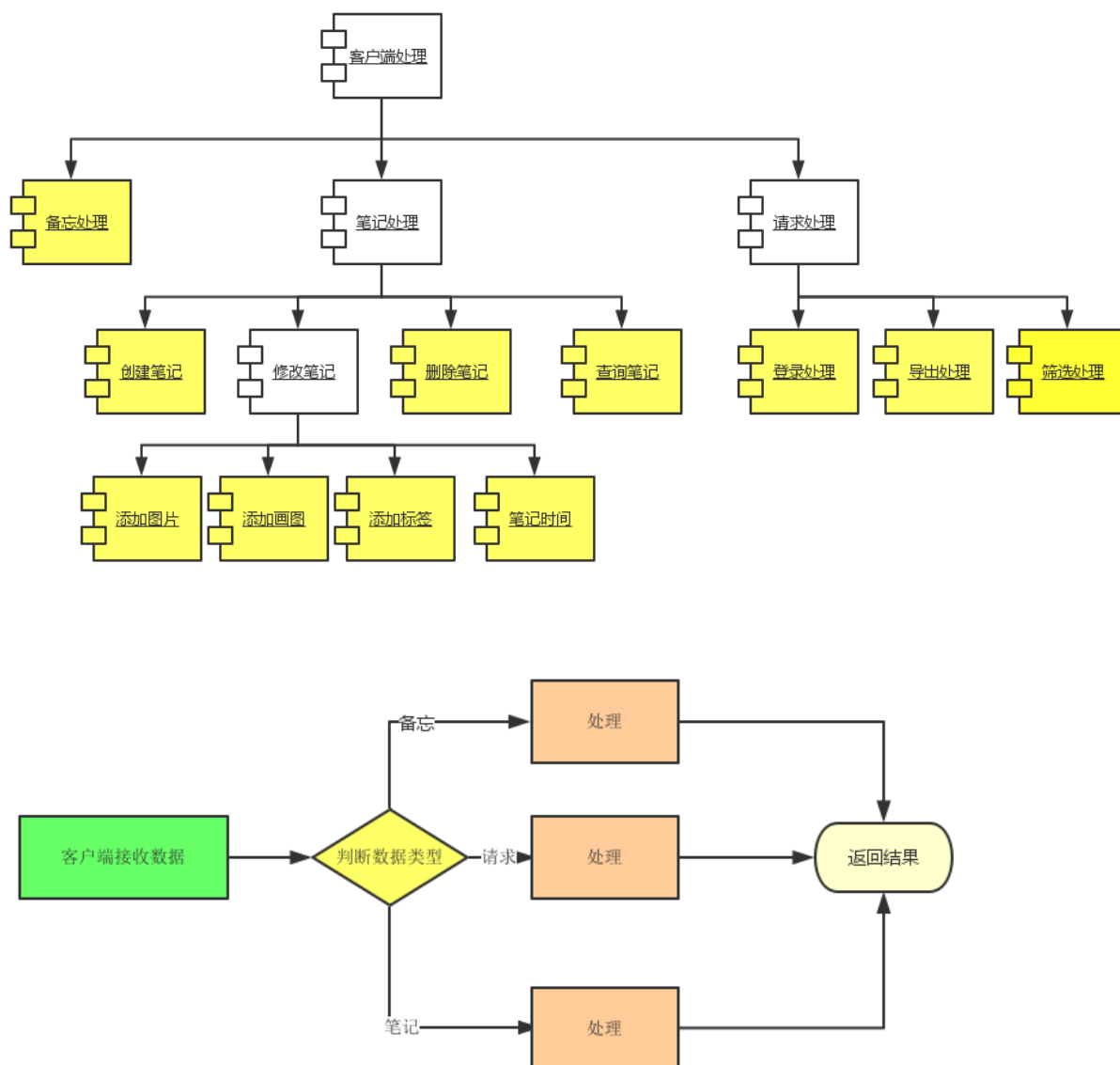
笔记&备忘软件的主要功能如下：

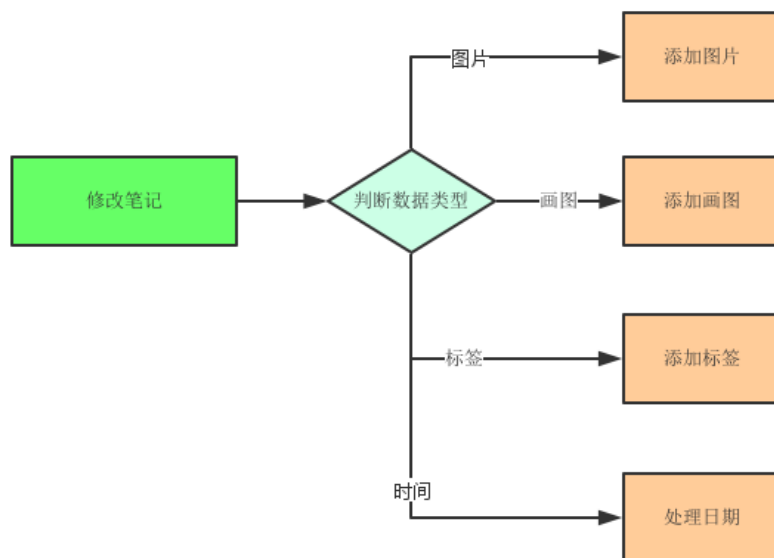
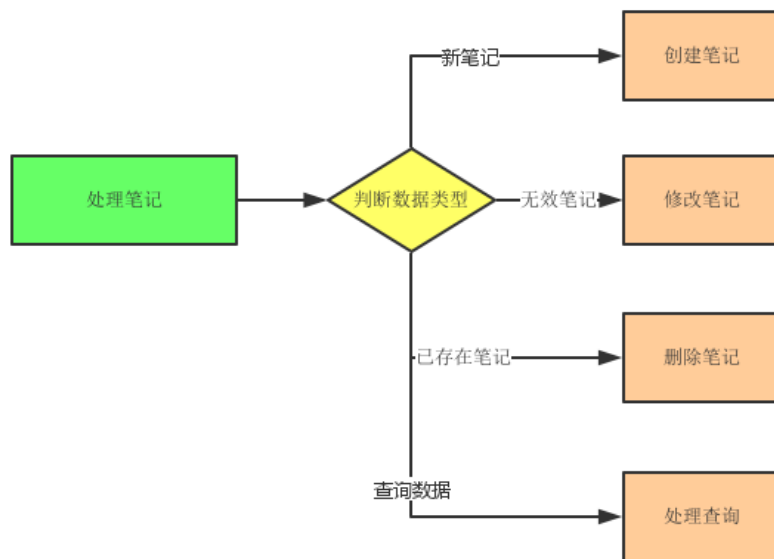
编号	功能	简介
XUQIU.001	用户注册	用户在 Android、iOS、Web Browser 初次使用本软件时注册账户。用户输入注册邮箱，输入密码，点击 Sign in 按钮完成注册。
XUQIU.002	用户登录和多平台同步	用户在 Android、iOS、Web Browser 输入用户名和密码登录对应账户。提示用户注册。登录成功后自动与服务器同步用户数据。
XUQIU.003	文本编辑	最精致地完成文本编辑的功能。方便地完成笔记记录功能。用户选择新笔记选项或者点击已经存在的笔记，进入编辑界面进行编辑，编辑结束后按完成按钮退出。
XUQIU.004	添加图片、画图功能	用户在编辑阶段，点击添加画图(由鼠标或手指画图)或者添加图片。
XUQIU.005	同步功能	在联网的情况下，每次返回主界面进行同步。
XUQIU.006	搜索功能	对笔记、备忘进行检索，可以根据创建时间、修改时间、标签、文本进行检索。
XUQIU.007	删除功能	删除某一笔记或备忘。
XUQIU.008	导出功能	对用户的笔记和备忘进行导出。

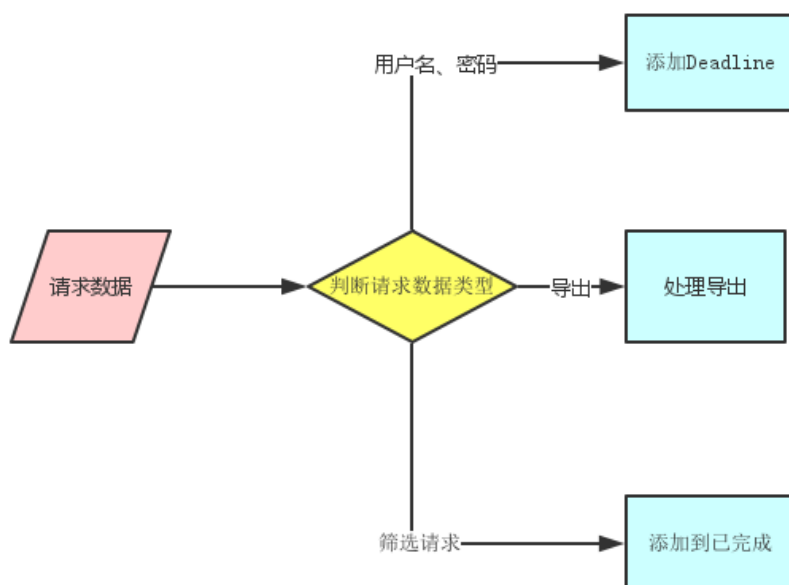
XUQIU.009	备忘功能	备忘功能就是设置时间在相应时间点提醒用户事件，注意由于要精致地完成本功能，就单纯地在本应用内完成提醒功能。
-----------	------	---

3.2 处理流程

（以 Android 为实例）







3.2.1 CHULI.001 备忘处理流程

对笔记添加备忘属性，客户端在本地保存此属性，编辑结束后上传此属性。

当提醒日期到达时刻（网页端没有此部分流程），客户端显示提醒，根据客户选择更改提醒时间或者添加该备忘到已完成列表，更改时间则更新此属性到提醒时刻，添加到已完成列表则保存该属性到本地，并将该备忘移动至已完成列表，结束后上传该属性。

3.2.2 CHULI.002 创建笔记处理流程

客户端创建新的笔记文件，编辑结束后上传该文件信息，文件内容经服务器处理后以数据库单元的形式存储在数据库中。

3.2.3 CHULI.003 添加图片处理流程

客户端在本地保存该图片的压缩文件，并在笔记文件中引用该图片。编辑结束后将压缩图片上传，并将图片和笔记的联系关系（存储在文件中）上传由服务器处理存储在数据库中。

3.2.4 CHULI.004 添加画图处理流程

客户端提供简单的画图界面供用户画图，画图结束后将画图保存为压缩图片格式，之后的处理流程调用处理图片的处理流程。

3.2.5 CHULI.005 添加标签处理流程

客户端编辑结束后，将笔记的标签信息存储在该笔记文件中，同时在全局的标签文件内存储相应的对应关系。上传该笔记文件后，在数据库的笔记信息和全局标签信息中都添加相应的内容。

3.2.6 CHULI.006 添加时间处理流程

创建笔记的时候，客户端自动标记创建时间到该笔记文件。每次编辑笔记的时候，客户端自动标记修改时间到该笔记文件。编辑结束后上传笔记文件信息，将创建时间和修改时间添加到数据库中。

3.2.7 CHULI.007 删除笔记处理流程

客户端删除笔记后，客户端删除笔记文件，同时在全局标签文件中删除笔记信息，搜索笔记对应的图片并删除。编辑结束后上传删除信息，服务器进行相同的处理。

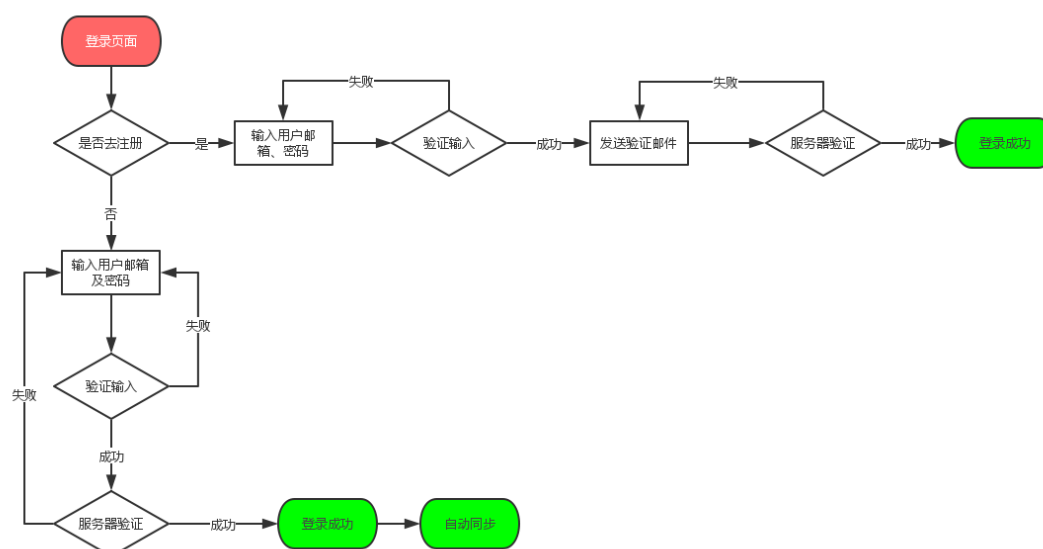
3.2.8 CHULI.008 查询笔记处理流程

客户端在本地进行查询处理流程，客户端遍历笔记，按照时间、标签、内容扫描文件并记录结果，搜索结果按照内容、标签、时间的顺序分类展示。

其中结果按照时间排序。

3.2.9 CHULI.009 登录处理流程

客户端发出登录请求，服务器按照下图响应。登录成功后下传同步数据。



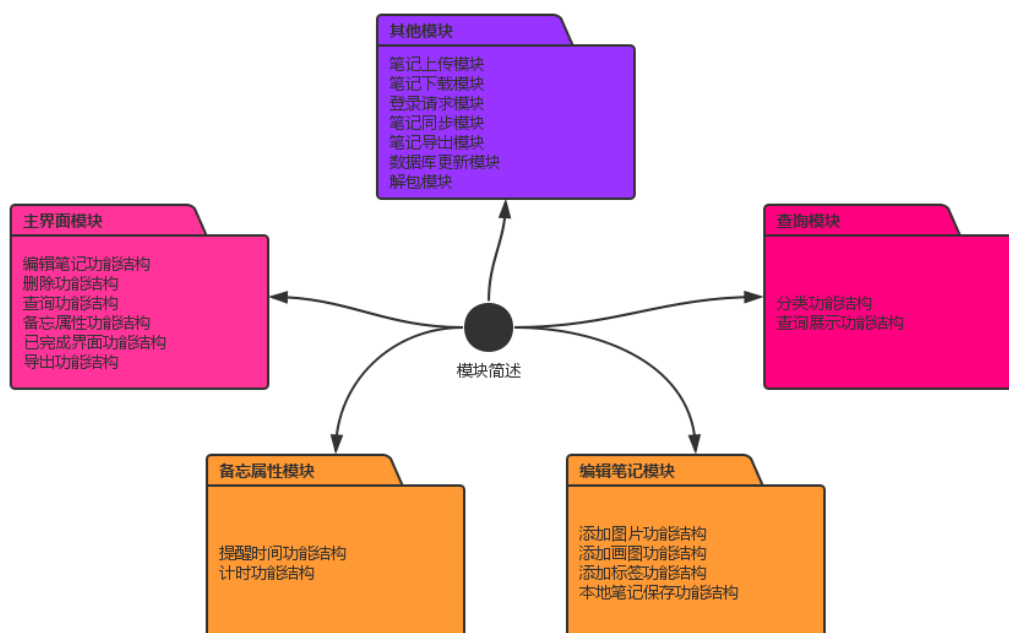
3.2.10 CHULI.010 导出处理流程

客户端发出导出的请求后，服务器将该客户当前的数据、图片打包发送到客户邮箱，反馈导出成功或者失败信息给客户。

3.2.11 CHULI.011 筛选处理流程

客户端点击已完成列表，然后显示已完成列表。

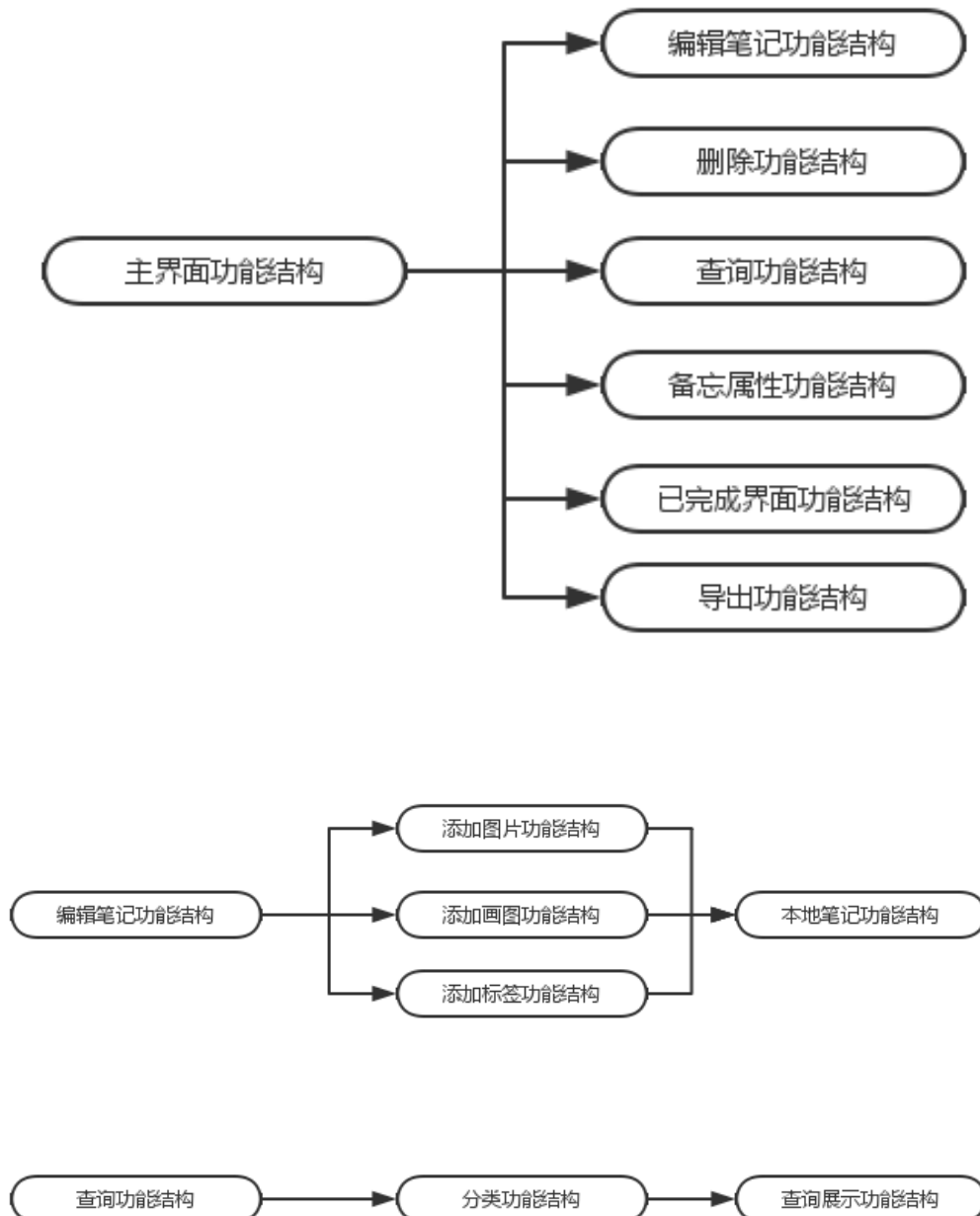
3.3 功能结构设计

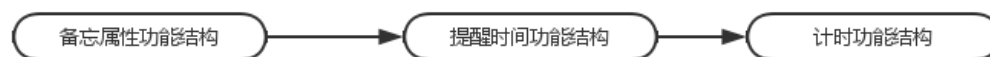


(以 Android 为实例)

模块编号	模块名称	子功能结构
M.MODULE 001	主界面模块	主界面功能结构 编辑笔记功能结构 删除功能结构 查询功能结构 备忘属性功能结构 已完成界面功能结构 导出功能结构
M.MODULE 002	备忘属性模块	备忘属性功能结构 提醒时间功能结构 计时功能结构
M.MODULE 003	编辑笔记模块	编辑笔记功能结构 添加图片功能结构 添加画图功能结构 添加标签功能结构 本地笔记保存功能结构
M.MODULE 004	查询模块	查询功能结构 分类功能结构 查询展示功能结构
M.MODULE 005	笔记上传模块	笔记上传功能结构

M.MODULE 006	笔记下载模块	笔记下载功能结构
M.MODULE 007	登录请求模块	登录请求功能结构
M.MODULE 008	笔记同步模块	笔记同步功能结构
M.MODULE 009	笔记导出模块	笔记导出功能结构
M.MODULE 010	数据库更新模块	数据库更新功能结构
M.MODULE 011	解包模块	解包功能结构





功能结构编号	功能结构名称	功能简述
客户端功能结构（以 Android 为实例）		
MODULE.001	主界面功能结构	完成客户端主界面的布局，实现和子功能结构的接口。
MODULE.002	编辑笔记功能结构	实现客户端笔记编辑的界面布局，实现基本的文字输入功能，实现和子功能结构的接口，记录笔记创建和更新时间。
MODULE.003	添加图片功能结构	实现从系统添加图片到笔记的功能。
MODULE.004	添加画图功能结构	完成画图界面的逻辑实现和保存图片功能。
MODULE.005	添加标签功能结构	实现本地全局标签的更新、同步和自动分类功能。
MODULE.006	删除功能结构	完成删除笔记或备忘的功能
MODULE.007	查询功能结构	实现查询关键字分词，识别功能。
MODULE.008	分类功能结构	实现查询，并对查询结果进行分类功能。
MODULE.009	查询展示功能结构	实现对查询结果的显示界面的布局。
MODULE.010	备忘属性功能结构	设置对笔记添加备忘功能。
MODULE.011	提醒时间功能结构	设置对笔记提醒的时间。
MODULE.012	计时功能结构	后台对备忘计时并提醒。
MODULE.013	已完成界面功能结构	对已完成笔记的界面布局和逻辑实现。
MODULE.014	登录功能结构	实现用户注册、登录
MODULE.015	导出功能结构	触发导出操作。
MODULE.016	本地笔记保存功能结构	实现对本地笔记的保存。
MODULE.017	笔记上传功能结构	实现对笔记的上传。
MODULE.018	笔记下载功能结构	实现对笔记的下载同步。
服务器端功能结构		
MODULE.019	登录请求功能结构	处理用户注册、登录。并响应。
MODULE.020	笔记同步功能结构	处理用户多平台同步、笔记更新同步逻辑实现。
MODULE.021	笔记导出功能结构	处理导出请求

MODULE.022	数据库更新功能结构	更新数据库。
MODULE.023	解包功能结构	处理客户端发送的 socket 包。

3.3.1 MODULE.001 主界面功能结构

布局由 main.xml 布局文件完成。实现主界面的布局。

主界面的触发事件也在布局文件中说明，事件的回调函数在 java 文件中实现。

本功能结构实现对编辑按钮、查询按钮、导出按钮、删除选项、备忘选项、已完成选项的布局和接口实现。

子功能结构：

MODULE.002 编辑笔记功能结构

MODULE.006 删除功能结构

MODULE.007 查询功能结构

MODULE.010 备忘属性功能结构

MODULE.013 已完成界面功能结构

MODULE.015 导出功能结构

3.3.2 MODULE.002 编辑笔记功能结构

布局在 note.xml 布局文件中实现。完成编辑界面的布局。编辑界面作为一个新的 Android Activity 实现。

编辑界面完成文本编辑功能，并为添加图片、添加画图、添加书签的实现准备好接口。

实现基本的时间记录功能，编辑结束后，笔记内容以 xml 的格式保存在本地 APP 文件下，每次笔记打开时读取 xml 文件并解析。

子功能结构：

MODULE.003 添加图片功能结构

MODULE.004 添加画图功能结构

MODULE.005 添加标签功能结构

MODULE.016 本地笔记保存功能结构

3.3.3 MODULE.003 添加图片功能结构

添加图片调用系统的照片库，图片插入显示在文本中依靠 MODULE.002 提供的支持。

3.3.4 MODULE.004 添加画图功能结构

添加画图界面作为一个 Android Fragment 的形式，只有基本的手写模式。

操作完成后保存为图片在该 APP 目录下，显示在文本中靠 MODULE.002 的支

持。

3.3.5 MODULE.005 添加标签功能结构

标签功能依托在编辑笔记功能结构中，本功能结构在笔记编辑结束后在全局标签记录中更新该笔记的记录。

3.3.6 MODULE.006 删除功能结构

删除功能由一个 Android Fragment 实现，会在主界面删除该笔记，并且删除本地笔记文件。

3.3.7 MODULE.007 查询功能结构

查询功能由触发查询按钮的回调函数实现，本功能结构完成对查询关键字的分词、分析和转化，子功能结构完成查询和分类展示的部分。

子功能结构：

MODULE.008 分类功能结构

MODULE.009 查询展示功能结构

3.3.8 MODULE.008 分类功能结构

本功能结构实现遍历笔记文件中标签、时间记录、标题和正文，搜索并记录结果，并且按照正文、标签、时间分类存储在三个结构体中。

3.3.9 MODULE.009 查询展示功能结构

本功能结构由一个 Android Fragment 实现结果的分类展示。

3.3.10 MODULE.010 备忘属性功能结构

备忘属性添加和备忘已完成、笔记删除在同一个 Android Fragment 中，触发该事件会调出另外一个 Android Fragment 用于设置提醒时间。

子功能结构：

MODULE.011 提醒时间功能结构

MODULE.012 计时功能结构

3.3.11 MODULE.011 提醒时间功能结构

实现一个 Android Fragment 用于时间的设置并触发一个 Thread 用于该提醒的计时。

3.3.12 MODULE.012 计时功能结构

在 MODULE.011 之后启动一个 Thread 用于提醒事项的计时，该 Thread 在提醒之前始终运行在该 APP 后台。

3.3.13 MODULE.013 已完成界面功能结构

界面由共享 main.xml 布局文件实现布局，布局结构和按钮安排基本和

MODULE.001 一样，实现一个新的 Fragment 来显示结果。已完成界面显示已完成备忘。

3.3.14 MODULE.014 登录功能结构

登录界面由一个 Android Activity 实现，并用 socket 的方式连接服务器确认登录情况。

3.3.15 MODULE.015 导出功能结构

导出按钮触发的事件完成一个 socket 的发送，请求服务器备份该用户的笔记。

3.3.16 MODULE.016 本地笔记保存功能结构

笔记在编辑结束后，按钮保存会触发一个事件。将该笔记记录在 APP 的变量数组中，并且将笔记的内容按照 xml 的格式存储在本地。

3.3.17 MODULE.017 笔记上传功能结构

3.3.16 MODULE.016 结束后自动调用该功能结构，实现笔记的上传功能。

3.3.18 MODULE.018 笔记下载功能结构

3.3.14 MODULE.014 登录功能结构成功后自动调用该功能结构，用于下载该用户的笔记文件和记录。

3.3.19 MODULE.019 登录请求功能结构

服务器查询数据库核对用户信息，并发回 socket 包确认登录成功与否。

3.3.20 MODULE.020 笔记同步功能结构

服务器调用数据库该用户的数据资料，整合成 xml 的格式以 socket 的方式不断发送给用户；同时接受用户的上传请求，将用户的 xml 包解析保存在数据库中。

3.3.21 MODULE.021 笔记导出功能结构

服务器响应用户的请求，将数据库中用户的笔记打包发送给用户邮箱，并以 socket 的形式反馈用户导出成功与否。

3.3.22 MODULE.022 数据库更新功能结构

服务器对数据库的请求均通过该功能结构。

3.3.23 MODULE.023 解包功能结构

用户发送给服务器的数据包均通过该功能结构解析，分类发送给下一个功能结构。

3.4 功能需求与程序代码的关系

功能需求编号	功能需求	功能功能结构	代码结构 (package)
XUQIU.001	用户注册	MODULE.019 登录请求功能结构	Package: register
XUQIU.002	用户登录和多平台同步	MODULE.014 登录功能结构 MODULE.018 笔记下载功能结构 MODULE.020 笔记同步功能结构	Package: login_and_synchronize
XUQIU.003	文本编辑	MODULE.001 主界面功能结构 MODULE.002 编辑笔记功能结构 MODULE.003 添加图片功能结构 MODULE.016 本地笔记保存功能结构 MODULE.004 添加画图功能结构 MODULE.005 添加标签功能结构	Package: editor
XUQIU.004	添加图片、画图功能	MODULE.002 编辑笔记功能结构 MODULE.003 添加图片功能结构 MODULE.004 添加画图功能结构	Package: pictures
XUQIU.005	同步功能	MODULE.017 笔记上传功能结构	Package: sync
XUQIU.006	搜索功能	MODULE.001 主界面功能结构 MODULE.007 查询功能结构 MODULE.008 分类功能结构 MODULE.009 查询展示功能结构	Package: search
XUQIU.007	删除功能	MODULE.001 主界面功能结构	Package: delete

		MODULE.006 删除功能结构	
XUQIU.008	导出功能	MODULE.001 主界面功能结构 MODULE.015 导出功能结构 MODULE.021 笔记导出功能结构	Package: backup
XUQIU.009	备忘功能	MODULE.001 主界面功能结构 MODULE.010 备忘属性功能结构 MODULE.011 提醒时间功能结构 MODULE.012 计时功能结构	Package:

5 接口设计

5.1 外部接口

5.1.1 MODULE.003 添加图片功能结构 \longleftrightarrow 文件系统

调用接口为 `openPicture(String dir).setOnClickListener(this)`

参数: `String dir` 表示图片的系统位置

用来打开客户端系统的文件夹, 选择图片。需要调用客户端系统的文件系统接口。

5.1.2 MODULE.012 计时功能结构 \longleftrightarrow 客户端系统

调用接口为 `pollingTime(remindNote t)`

参数: `remindNote t`

类型定义: `struct remindNote{`
`String time;`
`String note;`
`}`

该功能结构调用系统提醒功能结构, 并轮询检查时间是否到达备忘提醒的时刻。

5.1.3 MODULE.017 笔记上传功能结构 MODULE.018 笔记下载功能

结构←→客户端系统

调用接口为 `sendSocket(note t)`

`receveSocket(note t)`

参数: `Note t`

类型定义: `struct note{`
`String userid;`
`Byte data;`
`}`

该功能结构调用系统的 `socket` 接口发送和接收笔记记录, 用于同步笔记。

5.2 内部接口

5.2.1 MODULE.001 主界面功能结构←→其它功能结构

1、MODULE.002 编辑笔记功能结构

调用接口为: `addNote()`

参数: 空参数

该功能结构调用 `button` 事件创建新的 `Activity` 并且开始记录新的笔记。

2、MODULE.006 删除功能结构

调用接口为: `delete(noteName name)`

参数: `noteName name`

类型定义: `struct noteName{`
`String name;`
`String createDate;`
`String modifyDate;`
`Type type;`
`}`
`enum Type{`
`note=1, remind=2`
`}`

该功能结构实现对指定笔记的删除, 接口参数由主功能结构传递给该功能结构。

3、MODULE.007 查询功能结构

调用接口为: `search(keywords keyword)`

参数: `keywords keyword`

类型定义: `struct keywords{`
`String keyword;`
`}`

该功能结构实现主功能结构和查询功能结构的交互, 主功能结构将查询关键字传递给查询功能结构进行查询操作。

4、MODULE.010 备忘属性功能结构

调用接口为: `addRemind(noteName name)`

参数: noteName name

实现主功能结构与备忘功能结构的接口,主功能结构的点击事件驱动该功能结构事件的发生,对某一笔记添加备忘属性。

5、MODULE.013 已完成界面功能结构

调用接口为: finishBoard()

参数: 无参数

该功能结构实现一个新的 Fragment 来实现显示已完成的备忘,该事件的发生由主功能结构的点击事件驱动。

6、MODULE.015 导出功能结构

调用接口为: backup(backUp msg)

参数: backUp msg

类型定义: struct backUp{
String userid;
}

该功能结构实现导出功能,功能结构的导出信息由主功能结构传入,由主功能结构的点击事件触发,该功能结构处理信息并通过接口将信息打包给 socket。

5.2.2 MODULE.002 编辑笔记功能结构↔其它功能结构

1、MODULE.003 添加图片功能结构

调用接口为: addPic()

参数: 无参数

该功能结构由主功能结构的点击事件触发,该功能结构调用系统接口实现插入图片的操作。

2、MODULE.004 添加画图功能结构

调用接口为: addDraw()

参数: 无参数

该功能结构由主功能结构的点击事件触发,该功能结构调用该功能结构实现的画板来实现用户做图的功能,并保存为图片。

3、MODULE.005 添加标签功能结构

调用接口为: addTags(noteName name)

参数: noteName name

该功能结构由主功能结构的保存事件触发,实现在全局文件里保存该笔记的标签记录。

5.2.3 MODULE.007 查询功能结构↔ 其它功能结构

1、MODULE.008 分类功能结构

调用接口为: classify(keywords [] words)

参数: keywords [] words

分类功能结构通过该接口接收查询功能结构传递来的关键字数组,并将分类好的结果传回。

2、MODULE.009 查询展示功能结构

调用接口为: `showResult(keywordsResult [] results)`

参数: `keywordsResult [] results`

类型定义: `struct keywordsResult{`
`tagsResult [] tags;`
`dateResult [] dates;`
`textResult [] tests;`
`}`
`Struct tagsResult{`
`String tag;`
`noteName [] note;`
`}`
`Struct dateResult{`
`String date;`
`noteName note;`
`}`
`Struct textResult{`
`String keyword;`
`noteName note;`
`}`

展示功能结构由查询功能结构结束后调用该接口实现, 展示的结果由该功能结构绘制一个 Fragment 来实现。

5.2.4 MODULE.010 备忘属性功能结构 \longleftrightarrow 其它功能结构

1、MODULE.011 提醒时间功能结构

调用接口为: `setRemindTime(noteName note)`

参数: `noteName note`

该功能结构接收主功能结构传递给的事件触发, 绘制一个设置提醒时间的界面并设置提醒时间。

2、MODULE.012 计时功能结构

调用接口为: `openThread(remindNote t)`

参数: `remindNote t`

该功能结构接收一个由主功能结构传递的笔记名称和时间的结构体, 开启一个线程设置提醒, 并利用系统接口实现提醒功能。

5.2.5 MODULE.023 解包功能结构 \longleftrightarrow 其它功能结构

解包功能结构分析包的头部将包分登录请求、同步请求、导出请求。分别调用响应的功能结构操纵数据库实现服务器端的服务操作。

调用接口为: `commandHost(command d)`

参数: `command d`

类型定义: struct command{
 Byte b;
 }

1、MODULE.019 登录请求功能结构

```
struct loginCommand{
    commandID id;
    String userid;
    String password;
}
```

2、MODULE.020 笔记同步功能结构

```
struct updateCommand{
    commandID id;
    String userid;
    Byte s
}
```

3、MODULE.021 笔记导出功能功能结构

```
struct backupCommand{
    commandID id;
    String userid;
}
```

6 数据结构设计

6.1 逻辑结构设计

6.1.1 主控中心主要数据结构

6.1.1.1 服务器端

1、接收到的客户端请求

```
struct command{
    Byte b;
}
```

说明: 请求以 socket 的形式以字节流的格式传输到服务器端口。

2、登录请求的数据

```
struct loginCommand{
    commandID id;
    String userid;
    String password;
}
```

```
enum commandID{
```

```

        login=1, backup=2, update=3;
    }

```

说明：解析结果为登录请求，请求字段为用户邮箱和用密钥加密的密码。请求 ID 分别为登录请求、同步请求、导出请求。

4、同步请求

```

struct updateCommand{
    commandID id;
    String userid;
    Byte s
}

```

说明：解析结果为同步请求，请求字段为用户名，之后的为字节流，内容是笔记内容和名称，解析工作交给笔记同步功能结构完成。

5、导出请求

```

struct backupCommand{
    commandID id;
    String userid;
}

```

说明：解析结果为导出请求，请求字段为用户名。

6.1.1.2 客户端

1、笔记类型

```

struct noteName{
    String name;
    String createDate;
    String modifyDate;
    Type type;
}
enum Type{
    note=1, remind=2
}

```

说明：本地的笔记管理只需管理笔记名称和笔记类型、创建时间、修改时间，所以本地的笔记数据凡是记录笔记的都以该结构记录。

笔记类型分为笔记和备忘，默认笔记类型为笔记。

新建笔记时修改时间等于创建时间。

2、导出请求

```

struct backUp{
    String userid;
}

```

说明：导出请求记录一个笔记拥有者的邮箱就可以了。

3、查询结果

```

struct keywordsResult{
    tagsResult [] tags;
}

```

```

    dateResult [] dates;
    textResult [] tests;
}
Struct tagsResult{
    String tag;
    noteName [] note;
}
Struct dateResult{
    String date;
    noteName [] note;
}
Struct textResult{
    String keyword;
    noteName [] note;
}

```

说明： 查询结果是由多个结构体的数组来完成。分别为标签查询结果、日期查询结果、关键字查询结果，每个结果由一个该结果对应的关键字和笔记数组结构体。

4、 主界面功能结构

```

Class mainModule{
Private: noteName [] notes;
        Int notesNum;
        String userid;
        Remind [] remind;
}
struct Remind{
    noteName note;
    String time;
}

```

说明： 主界面的主功能结构，保存笔记的数目、笔记拥有者、笔记的数组结构和备忘的数组结构。

备忘是由笔记、提醒时间组成。

6.2 物理结构设计

各数据结构无特殊物理结构要求。

6.3 数据结构与程序代码的关系

功能编号	功能	代码结构（package）	数据结构
XUQIU.001	用户注册	Package: register	struct command

			struct loginCommand enum commandID
XUQIU.002	用户登录和多平台同步	Package: login_and_ synchronize	struct command struct loginCommand enum commandID
XUQIU.003	文本编辑	Package: editor	struct noteName
XUQIU.004	添加图片、画图功能	Package: pictures	struct noteName
XUQIU.005	同步功能	Package: sync	struct command struct updateCommand enum commandID
XUQIU.006	搜索功能	Package: search	struct noteName sturct keywordsResult
XUQIU.007	删除功能	Package: delete	struct noteName
XUQIU.008	导出功能	Package: backup	struct command struct backupCommand enum commandID
XUQIU.009	备忘功能	Package:	struct noteName struct Remind

7 数据库设计

7.1 数据库环境说明

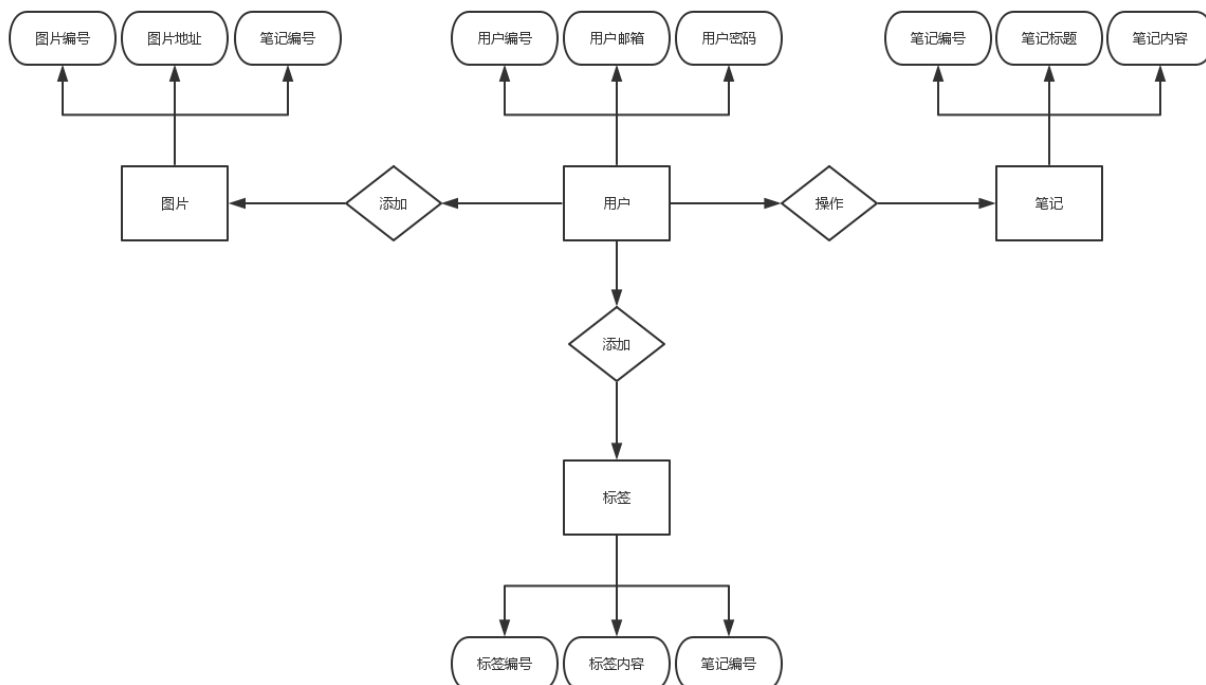
系统采用 Mysql 数据库。在服务器端采用 python 的编程语言和 PYMysql 数据库接口。数据库名字为 NoteDatabase

7.2 数据库的命名规则

数据表的命名规则为 XXtable，其中 xx 为数据表存储的内容的小写英文单词。表中每一个字段以字段代表含义的小写英文单词命名。

7.3 逻辑设计

7.3.1 逻辑设计的 E-R 图



7.3.2 数据库逻辑结构设计

用户表: usertable

列名	数据类型	可否为空	说明
UserID	Int	Not NULL	Key
Useremail	Varchar	Not NULL	用户邮箱
Userpassword	Char	Not NULL	用户密码

笔记表: notetable

列名	数据类型	可否为空	说明
NoteID	Int	Not NULL	Key
Notename	Nvarchar	NULL	笔记标题
Notecont	Ntext	NULL	笔记文本内容
Settime	Smalldatetime	Not NULL	笔记创建时间
Updatetime	Smalldatetime	Not NULL	笔记最后更新时间
Warntime	Datetime	NULL	备忘提醒时间
Finished	Int	Not NULL	是否完成

图片表: picturetable

列名	数据类型	可否为空	说明
----	------	------	----

PicID	Int	Not NULL	Key
Piccont	Varchar	Not NULL	图片地址
NoteID	Int	Not NULL	对应笔记的 ID

标签表: labeltable

列名	数据类型	可否为空	说明
LabelID	Int	Not NULL	Key
Labelname	Nvarchar	Not NULL	标签名
NoteID	Int	Not NULL	对应笔记的 ID

7.4 物理设计

数据库环境

数据库采用 MySQL。

数据存储可采用 RAID5+RAID1 磁盘阵列的方式。

内存配置在 8G，来减少磁盘读取时间。

数据库参数设计

数据库类型

由于数据库目标为企业级数据仓库，数据库类型选择 data warehouse 类型。

连接方式：同时连接类型选择专用方式连接，来满足数据装载时的大量批处理服务。

内存配置：根据服务器实际物理内存的大小，选择 70%-80% 的内存作为数据库内存大小。

字符集：为了使数据库能够正确支持多国语言，需要将数据库字符集配置为 UTF-8 字符集。

其他参数：聚合内存使用，连接数最大为 10、数据块大小 2M、缓冲区大小为 100M。

数据库存储设计

控制文件：控制文件中包含数据库重要信息，需要将控制文件存放在多个磁盘中，来保证数据库可恢复性。控制文件中参数设置，最大的数据文件数量不能小于数据库参数 db_files。

日志文件：数据仓库通常为批处理装载，在装载时会产生大量日志。可选择关闭某些事实表日志，对通常的维表及高频率装载的数据表，可以选择打开日志功能。日志文件的大小由数据库事务处理量决定，在设计过程中，确保每 20 分钟切换一个日志文件。对于数据仓库系统，日志文件大小通常为几百兆到几千兆。为了确保日志能够镜像作用，每日志组的成员为 2 个，日志文件组为 5-10 组。

回滚段配置：Undo space = UR * UPS * db_block_size + 冗余量。UR：表示在 undo 中保持的最长时间数（秒），由数据库参数 UNDO_RETENTION 值决定。UPS：表示在 undo 中，每秒产生的数据库块数量。

临时段表空间配置：数据库临时段表空间根据实际生产环境情况调整其大小，表空间属性为自动扩展。

系统表空间配置：系统表空间大小 1G 左右，除了存放数据库数据字典的数据外，其他数据不得存储在系统表空间。

表空间大小定义：当表空间大小小于操作系统对最大文件限制时，表空间由一个文件组成。如果表空间大小大于操作系统对最大文件限制时，该表空间由多个数据文件组成，表空间的总大小为估算为：Tablespace + sum (数据段+索引段)*150%。

表空间扩展性设计原则：表空间数据文件采用自动扩展的方式，扩展容量快大小按 2 的整数倍（1M、2M、4M、8M、16M、32M、64M）进行扩展，创建表空间时尽量采用 nologging 选项。表空间的最大限制一般采用 unlimited，除非确切知道表空间数据文件的最大使用范围。

(64 位系统文件最大 128G),最大为 2G。表空间采用 local 管理方式。

特殊表设计

分区表:对于数据量比较大的表,根据表数据的属性进行分区,以得到较好的性能。如果表按某些字段进行增长,则采用按字段值范围进行范围分区;如果表按某个字段的几个关键值进行分布,则采用列表分区;对于静态表,则采用 hash 分区或列表分区;在范围分区中,如果数据按某关键字段均衡分布,则采用子分区的复合分区方法。

聚簇表:如果某几个静态表关系比较密切,则可以采用聚簇表的方法。

完整性设计

主键约束:关联表的父表要求有主键,主键字段或组合字段必须满足非空属性和唯一性要求。对于数据量比较大的父表,要求指定索引段。

外键关联:对于关联两个表的字段,一般应该分别建立主键、外键。实际是否建立外键,根据对数据完整性的要求决定。为了提高性能,对于数据量比较大的表要求对外键建立索引。

索引设计

对于查询中需要作为查询条件的字段,可以考虑建立索引。最终根据性能的需要决定是否建立索引。对于复合索引,索引字段顺序比较关键,把查询频率比较高的字段排在索引组合的最前面。在分区表中,尽量采用 local 分区索引以方便分区维护。

视图设计

视图是虚拟的数据库表,在使用时要遵循以下原则:从一个或多个库表中查询部分数据项;为简化查询,将复杂的检索或字查询通过视图实现;提高数据的安全性,只将需要查看的数据信息显示给权限有限的人员;

视图中如果嵌套使用视图,级数不得超过 3 级;

由于视图中只能固定条件或没有条件,所以对于数据量较大或随时间的推移逐渐增多的库表,不宜使用视图;可以采用实体化视图代替。

视图中尽量避免出现数据排序的 SQL 语句。

包设计

存储过程、函数、外部游标必须在指定的数据包对象

PACKAGE 中实现。存储过程、函数的建立如同其它语言形式的编程过程,适合采用模块化设计方法;当具体算法改变时,只需要修改需要存储过程即可,不需要修改其它语言的源程序。当和数据库频繁交换数据是通过存储过程可以提高运行速度,由于只有被授权的用户才能执行存储过程,所以存储过程有利于提高系统的安全性。

存储过程、函数必须检索数据库表记录或数据库其他对象,甚至修改(执行 Insert、Delete、Update、Drop、Create 等操作)数据库信息。如果某项功能不需要和数据库打交道,则不得通过数据库存储过程或函数的方式实现。在函数中避免采用 DML 或 DDL 语句。

在数据包采用存储过程、函数重载的方法,简化数据包设计,提高代码效率。存储过程、函数必须有相应的出错处理功能。

安全性设计

管理默认用户:在生产环境中,必须严格管理 sys 和 system 用户,必须修改其默认密码,禁止用该用户建立数据库应用对象。删除或锁定数据库测试用户。

数据库级用户权限设计:必须按照应用需求,设计不同的用户访问权限。包括应用系统管理用户,普通用户等,按照业务需求建立不同的应用角色。用户访问另外的用户对象时,应该通过创建同义词对象 synonym 进行访问。

角色与权限:确定每个角色对数据库表的操作权限,如创建、检索、更新、删除等。每个角色拥有刚好能够完成任务的权限,不多也不少。在应用时再为用户分配角色,则每个用户的权限等于他所兼角色的权限之和。

应用级用户设计:应用级的用户帐号密码不能与数据库相同,防止用户直接操作数据库。用户只能用帐号登陆到应用软件,通过应用软件访问数据库,而没有其它途径操作数据库。
用户密码管理:用户帐号的密码必须进行加密处理,确保在任何地方的查询都不会出现密码的明文。

备份恢复设计原则

数据库热备份恢复:数据库通常提供了数据快速的热备份和热恢复手段,提供了数据库级、用户级和表级的数据备份恢复方式。这种方法一般作为数据库辅助备份手段。

数据库冷备份原则:数据库冷备份必须符合以下原则:数据库容量比较小。数据库允许关闭的情况。

数据库级备份原则:在数据库的数据量比较小,或数据库初始建立的情况下采用。不适合7*24的在线生产环境数据库备份。

用户级备份原则:在用户对象表数据容量比较小、或则用户对象初始建立的情况下使用。

表级备份原则:主要在以下场合采用的备份方式:参数表备份、静态表备份、分区表的分区备份。

7.5 安全性设计

由于系统本身的特点,不需要考虑数据库的安全性,因此不需要措施防止用户直接操作数据库。

7.6 数据库管理与维护说明

数据的管理和维护基于 Mysql 系统自身的功能进行,使用该管理器的导入数据,导出数据,备份数据库和还原数据库等功能对数据库进行管理和维护。

8 界面设计

8.1 安卓版界面

8.1.1 登陆界面



登陆界面说明：

注册邮箱：注册邮箱即用户注册时所使用的邮箱。

密码：密码即用户注册时所设置的密码。

自动登陆：勾选自动登陆选项后每次打开应用就会自动登陆，无需输入注册邮箱和密码，登陆成功直接进入主界面。

注册：没有注册的用户需要点击注册选项进行注册。

8.1.2 主界面



主界面说明：

左上角的 **Backup**: 该选项用于备份，按下该按键则会向服务器端备份笔记。

右上角的 “+”：该选项用于创建新的笔记，点击该选项会进入编辑笔记界面。

右上角的放大镜图标：该选项用于搜索。

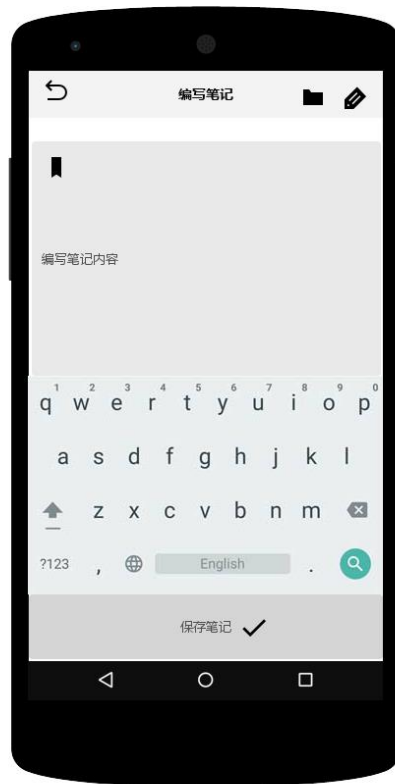
中间的标签：每个比较都有一个标签，标签会显示笔记标题和第一行的内容，右端显示最后编辑的时间。向左滑动标签会有动画效果，同时出现三个选项。第一个是闹钟图标，用于设置备忘；第二个是对勾图标，用于确认是否完成，确认完成的任务将会从笔记备忘栏中移除并加入到已完成栏中；第三个是垃圾桶图标，用于删除笔记。

左下角的 “笔记备忘”：笔记备忘处于高亮状态说明这时显示的是笔记备忘的标签。

下方中间的 “***条笔记”：***表示的是数字，该选项用于表示栏目中的笔记条数。

右下角的 “已完成”：高亮时表示显示的是已完成栏目。

8.1.3 笔记编辑界面



笔记编辑界面说明：

左上角的返回按钮：点击后返回主界面，不会修改原笔记。

右上角的“文件图标”：点击该图标可以向笔记内添加图片。

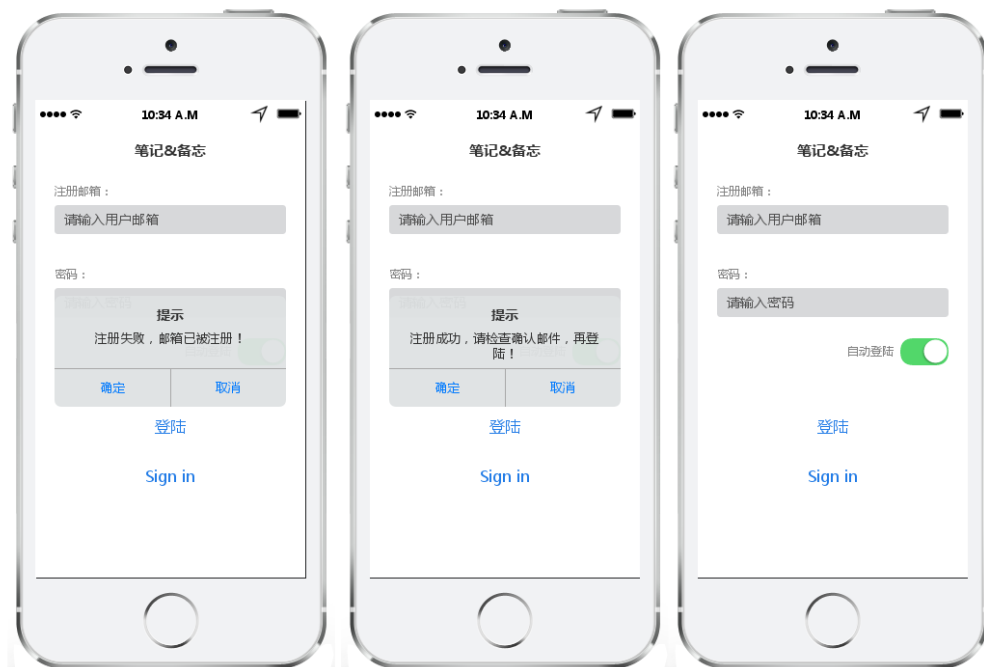
右上角的“画笔图标”：点击该图标后可以向笔记内画图。

中间空白：空白处会有光标，通过键盘可以向光标处输入字符。

下方“保存笔记”按钮：点击后会保存编辑后的笔记并返回主界面。

8.2 ios 版界面

8.2.1 登陆界面



登陆界面说明:

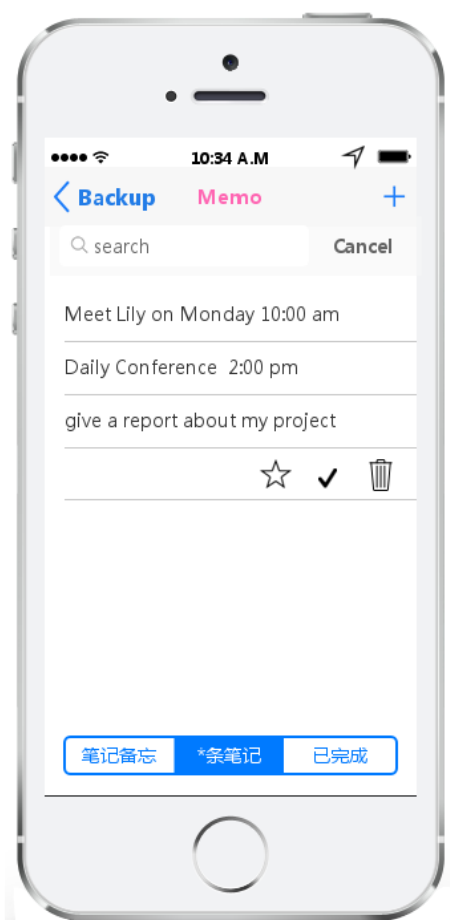
注册邮箱: 注册邮箱即用户注册时所使用的邮箱。

密码: 密码即用户注册时所设置的密码。

自动登陆: 勾选自动登陆选项后每次打开应用就会自动登陆, 无需输入注册邮箱和密码, 登陆成功直接进入主界面。

注册: 没有注册的用户需要点击注册选项进行注册。

8.2.2 主界面



主界面说明：

左上角的 **Backup**: 该选项用于备份，按下该按键则会向服务器端备份笔记。

右上角的 **“+”**：该选项用于创建新的笔记，点击该选项会进入编辑笔记界面。

中间偏上的搜索项：该选项用于搜索。

中间的标签：每个比较都有一个标签，标签会显示笔记标题和第一行的内容，右端显示最后编辑的时间。向左滑动标签会有动画效果，同时出现三个选项。第一个是闹钟图标，用于设置备忘；第二个是对勾图标，用于确认是否完成，确认完成的任务将会从笔记备忘栏中移除并加入到已完成栏中；第三个是垃圾桶图标，用于删除笔记。

左下角的 **“笔记备忘”**：笔记备忘处于高亮状态说明这时显示的是笔记备忘的标签。

下方中间的 **“***条笔记”**：***表示的是数字，该选项用于表示栏目中的笔记条数。

右下角的 **“已完成”**：高亮时表示显示的是已完成栏目。

8.2.3 笔记编辑界面



笔记编辑界面说明：

左上角的返回按钮：点击后返回主界面，不会修改原笔记。

右上角的“+”：点击该图标可以向笔记内添加图片。

左上角的“+”：点击该图标后可以选择向笔记内画图。

中间空白：空白处会有光标，通过键盘可以向光标处输入字符。

下方“完成”按钮：点击后会保存编辑后的笔记并返回主界面。

8.3 web 版界面

8.3.1 登陆界面



笔记&备忘

用户邮箱：

密码：

☐ 记住密码

提交 注册

登陆界面说明：

用户邮箱：用户邮箱即用户注册时所使用的邮箱。

密码：密码即用户注册时所设置的密码。

记住密码：勾选自动登陆选项后每次打开应用就会自动登陆，无需输入注册邮箱和密码，登陆成功直接进入主界面。

注册：没有注册的用户需要点击注册选项进行注册。

8.3.2 主界面



主界面说明：

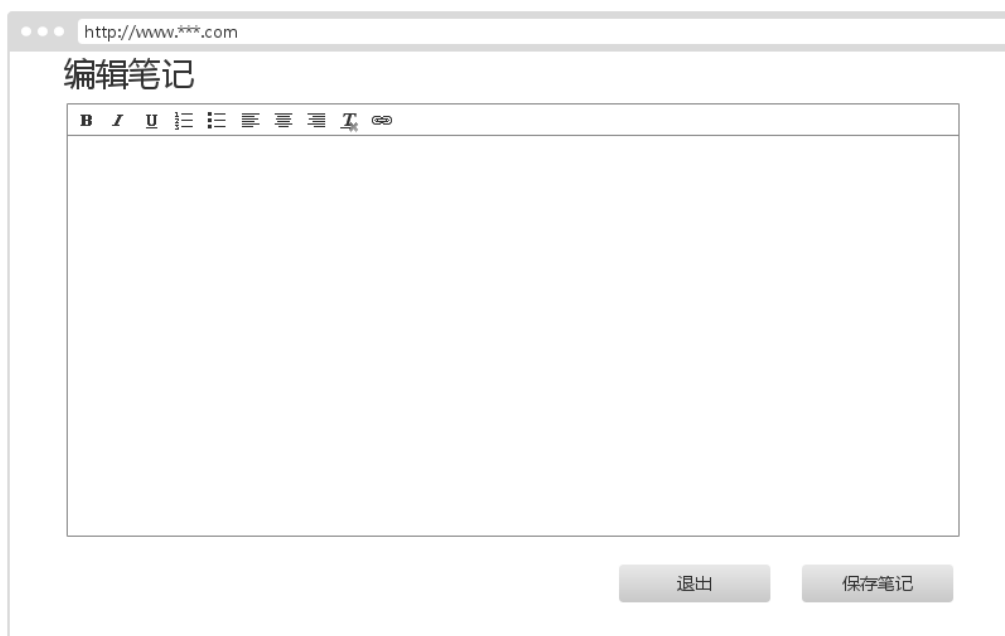
“新建笔记”按钮：点击新建笔记，进入笔记编辑界面。

“搜索”按钮：点击搜索，在用户的所有笔记（包括已完成和未完成）中搜索搜索框中的内容。

“删除”按钮：点击删除该笔记。

已完成选项：勾选则将笔记加入已完成队列。

8.3.3 笔记编辑界面



笔记编辑界面说明：

编辑框：在编辑框中可以编辑笔记。

退出：点击退出笔记编辑界面，而且不保存已修改的内容。

保存笔记：点击保存笔记。

9 出错处理设计

9.1 出错输出信息

9.1.1 用户登录

9.1.1.1 未在数据库中找到相同的用户邮箱

输出：用户未注册！

9.1.1.2 用户邮箱与密码不匹配

输出：邮箱与密码不匹配！

9.1.1.3 登陆超时

输出：登陆超时，请检查网络设置！

9.1.2 笔记同步

9.1.2.1 同步时失败/超时

输出：同步失败/超时，请检查网络设置！

9.1.2.3 同步时本地存储空间不足

输出：本地存储空间不足，同步失败！

9.1.3 笔记编辑

9.1.3.1 本地存储空间不足

输出：本地存储空间不足！

9.1.3.2 输入非法字符

输出：输入非法字符！

9.1.3.3 输入超过字符数量限制

输出：超过字符数量限制！

9.1.3.4 插入图片失败

输出：图像插入失败！

9.1.3.5 图像未找到

输出：图像未找到！

9.1.4 查找

9.1.4.1 无网络连接

输出：无法连接到服务器，请检查网络设置！

9.1.4.2 网络连接超时

输出：网络连接超时！

9.1.5 设置备忘

9.1.5.1 无网络连接

输出：无法连接到服务器，请检查网络设置！

9.1.5.2 网络连接超时

输出：网络连接超时！

9.2 出错处理对策

9.2.1 用户登录

9.2.1.1 未在数据库中找到相同的用户邮箱

处理：返回错误信息，登陆失败。

9.2.1.2 用户邮箱与密码不匹配

处理：返回错误信息，登陆失败。

9.2.1.3 登陆超时

处理：返回错误信息，登陆失败。

9.2.2 笔记同步

9.2.2.1 同步时失败/超时

处理：返回错误信息，保持已完整保存的笔记，丢弃未完整保存的笔记。

9.2.2.2 同步时本地存储空间不足

处理：返回错误信息，取消同步。

9.2.3 笔记编辑

9.2.3.1 本地存储空间不足

处理：返回错误信息，尝试将该笔记上传至服务器。

9.2.3.2 输入非法字符

处理：返回错误信息，删除/禁止输入该字符。

9.2.3.3 输入超过字符数量限制

处理：返回错误信息，禁止输入。

9.2.3.4 插入图片失败

处理：返回错误信息，取消插入操作。

9.2.3.5 图像未找到

处理：返回错误信息，取消插入操作。

9.2.4 查找

9.2.4.1 无网络连接

处理：返回错误信息，取消之前的操作。

9.2.4.2 网络连接超时

处理：返回错误信息，取消之前的操作。

9.2.5 设置备忘

9.2.5.1 无网络连接

处理：返回错误信息，取消之前的操作。

9.2.5.2 网络连接超时

处理：返回错误信息，取消之前的操作。

10 安全保密设计

由于系统本身的特点，不需要考虑数据库的安全性，因此不需要措施防止用户直接操作数据库。

11 维护设计

本系统不需要专用的维护程序。