应用商店系统
概要设计

	人员	日期
拟制	董恒 李楠	2019-5-9
评审人	董恒 李楠	2019-5-10
批准	董恒 李楠	2019-5-11
签发	董恒 李楠	2019-5-12

摘　要

本说明是软件工程-应用商店系统案例研究项目软件产品的总体设计和实现说明，记录了系统整体实现上技术层面上的考虑，并且以需求说明作为依据，同时该文档将作为产品实现、特性要求和控制的依据。

本阶段已在系统的需求分析研究的基础上,对应用商店系统做概要设计。该阶段正式进入了实际开发阶段，它的目的就是进一步细化软件设计阶段得出的软件总体概貌，把它加工成在程序细节上非常接近于源程序的软件表示。概要设计说明书主要解决了实现本系统需求的程序模块设计问题。

关键词：软件工程　应用商店系统　　概要设计

表 1　缩略词清单

缩略语	英文全名	中文解释
app	application	应用
dev	developer	开发者
inter	interface	接口
info	information	信息
passwd	password	密码

目 录

图目录

表目录

第 1 章　引言

1.1　编写目的

在本项目的前一阶段，也就是需求分析阶段，已经将系统用户对本系统的需求做了详细的阐述，这些用户需求已经在上一阶段中对不同用户所提出的不同功能，实现的各种效果做了调研工作，并在需求规格说明书中得到详尽得叙述及阐明。

本阶段已在系统的需求分析的基础上，对应用商店系统做概要设计。主要解决了实现应用商店系统的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明，在设计过程中起到了提纲挈领的作用。

在下一阶段的详细设计中，程序设计员可参考此概要设计报告，在概要设计应用商店系统所做的模块结构设计的基础上，对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书，以便于了解在概要设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误。

1.2　项目背景

应用商店的开发与时俱进，现在出现的 PC 端与手机端应用商店出现了严重的割裂，而且应用的质量得不到保证、用户的反馈得不到及时的回应、开发者的合理权益得不到保障、用户需要专门花时间去找应用等等问题层出不穷。

现在出现的应用商店无法解决这样问题，这也是本项目出现的主要原因，即构建一个面向未来的、可拓展的、友好的应用商店系统。以达到全面替代当前的各种应用商店的目的。

1.3　术语

表 1.1　术语表

缩写、术语	解释
用户	应用使用人员
开发者	应用开发人员
管理员	服务器与数据库维护人员

第 2 章　任务概述

本系统的目标是实现一个应用商店系统，包括客户端、服务器端两个部分。

在需求分析文档中，我们提到，不应严格区分普通用户和开发者，即普通用户和开发者共用一个客户端。

客户端面向普通用户和开发者，服务器端面向管理者。

2.1　目标

实现应用商店系统，实现需求规格说明书中所描述的开发者端应用管理功能、开发者端信息反馈功能、服务器端开发者管理系统功能、服务器端应用管理系统功能、服务器端用户管理系统功能、服务器端第三方管理系统功能、客户端应用管理功能、客户端信息反馈功能等等，并且保证系统的健壮性和数据安全。

2.2　开发与运行环境

2.2.1　开发环境的配置

开发环境配置见表2.1

2.2.2　测试环境的配置

测试环境见表2.2

2.2.3　运行环境的配置

运行环境配置见表2.3

2.3　需求概述

功能需求包括:

表 2.4　功能需求概述

需求编码	需求名称	解释

需求编码	需求名称	解释
SRS_Dev_App_P01	开发者端. 应用管理	实现开发者的应用上传、更新、删除等功能
SRS_Dev_Info_P01	开发者端. 信息反馈	实现开发者查询收入、收入提现或转账、评论查看或回复等功能
SRS_Server_Dev_P01	服务器端. 开发者管理系统	实现对开发者的活动进行管理和响应的功能
SRS_Server_App_P01	服务器端. 应用管理系统	实现对应用的管理，包括应用信息的维护与更新、应用的上传、下载、删除等等
SRS_Server_User_P01	服务器端. 用户管理系统	实现对用户的活动进行管理和响应的功能
SRS_Server_Other_P01	服务器端. 第三方管理系统	实现第三方的广告投放、资助等功能
SRS_User_App_P01	客户端. 应用管理	实现普通用户的应用查询、购买、安装、删除等功能
SRS_User_Info_P01	客户端. 信息反馈	实现普通用户对应用的评分、评价等功能

2.4 条件与限制

可参照需求分析文档中的总体设计约束部分，具体如下:

2.4.1 标准符合性

开发过程中的以下内容应遵从如下标准:

表 2.5 采用的标准和规范

内容	标准或规范
字符集	utf-8
网络协议	RFC

表 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 x86 结构的 CPU 主频 >=2.4GHz 内存 >=8G 硬盘 >=200G	基于 x86 结构的 CPU 主频 >=1.6GHz 内存 >=512M 硬盘 >=2G
计算机软件	Linux >=4.10) GNU gcc (version>=6.3.1)	Linux (kernel version>=3.10) GNU gcc (version>=5.4)
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
其他	采用 Oracle 数据库	采用 Oracle 数据库

注：对操作系统不做特殊要求但要求代码能够跨平台编译或者为不同的平台分别编写代码

表 2.2　测试环境的配置

类别	标准配置	最低配置
计算机硬件	基于 x86 结构的 CPU 主频 >=2.4GHz 内存 >=8G 硬盘 >=200G	基于 x86 结构的 CPU 主频 >=1.6GHz 内存 >=512M 硬盘 >=2G
计算机软件	应用商店所支持的操作系统平台 GNU gcc (version>=6.3.1)	应用商店所支持的操作系统平台 GNU gcc (version>=5.4)
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
其他	服务器端：采用 MySQL 数据库	服务器端：采用 MySQL 数据库

表 2.3　运行环境的配置

类别	标准配置	最低配置
计算机硬件	基于 x86 结构的 CPU 主频 >=2.4GHz 内存 >=8G 硬盘 >=200G	基于 x86 结构的 CPU 主频 >=1.6GHz 内存 >=512M 硬盘 >=2G
计算机软件	应用商店所支持的操作系统平台 GNU gcc (version>=6.3.1)	应用商店所支持的操作系统平台 GNU gcc (version>=5.4)
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
其他	服务器端：采用 MySQL 数据库	服务器端：采用 MySQL 数据库

2.4.2　硬件限制

该应用商店可在 Windows、Macintosh、Linux、Android、iOS 等多种平台上运行，在不同平台上的约束要求如下：

表 2.6　在不同平台运行的需求

平台	**需求**
Windows	内存占用不超过 100MB，存储空间占用不超过 100MB
Macintosh	内存占用不超过 100MB，存储空间占用不超过 100MB
Linux	内存占用不超过 100MB，存储空间占用不超过 100MB
Android	内存占用不超过 200MB，存储空间占用不超过 100MB
iOS	内存占用不超过 100MB，存储空间占用不超过 200MB

2.4.3　技术限制

本节描述了开发人员在开发该应用商店系统时使用的技术限制。

表 2.7　技术限制

技术	**限制**
数据库	Oracle 数据库
通讯协议	客户端、服务器、开发者端之间的通信均采用 TCP 协议
编程规范	采用 Google 代码规范，详见 https://zh.google

第 3 章　总体设计

3.1　软件描述

系统包括服务器端、用户端和开发者端三个部分。

服务器端主要功能是：维护应用、开发者、用户的数据，响应和处理开发者端、用户端的请求，提供管理人员管理接口，便于维护更新与第三方的接入。

用户端主要功能是：支持用户查询、下载、更新、安装、删除应用，并且能够对应用评价、评分。

开发者端端主要功能是：支持开发者上传、更新、删除应用，并且能够查询收入、将收入提现或转账、查询或回复应用的评价。

3.2　处理流程

3.2.1　总体流程

总体流程针对使用者的逻辑。见图3.1

无论是开发者、用户还是管理员，都需要都账户，作为账户以及提供权限。

其中管理员的注册需要额外的信息以及身份验证。

3.2.2　系统基本流程

系统基本流程简要表述不同的端与服务器的交互逻辑，见图3.2.

主要需要区分管理人员和普通用户（包括应用开发人员）

普通用户是通过开发者端或者用户段软件与服务器进行连接，然后请求通过网络交由服务器处理；而管理员需要使用另外的专门的接口，通过命令来操作。

3.2.3　客户端基本流程

我们之前提到过，不对开发者和普通用户用户作严格区分，两者共用一个客户端，开发者一定是普通用户，但反过来不成立。简单描述客户端处理用户请求的逻辑，见图3.3

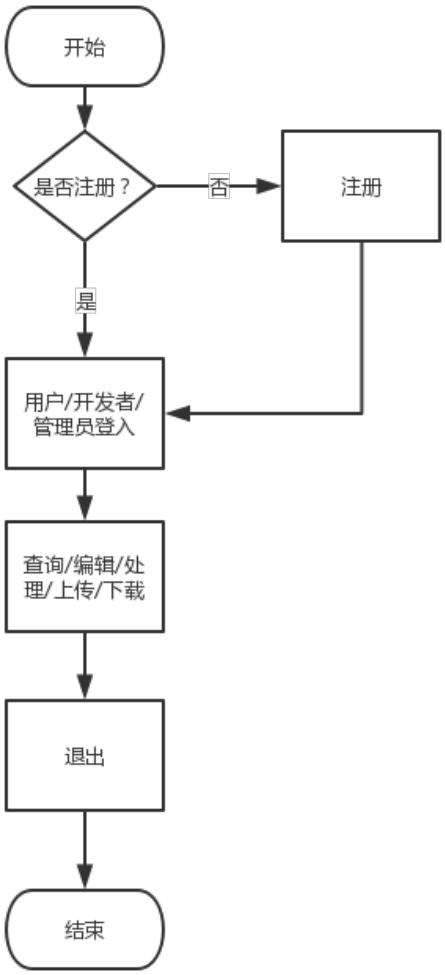

图 3.1　总体流程

3.2.4　客户端. 注册与登录请求处理功能

对于客户的登录与注册请求，见图3.4

3.2.5　客户端. 普通用户应用管理请求处理功能

对于客户的应用管理请求，见图3.5

3.2.6　客户端. 普通用户信息反馈请求处理功能

对于客户的信息反馈请求，见图3.6

3.2.7 客户端. 开发者应用管理请求处理功能

对于开发者的应用管理请求，见图3.7

3.2.8 客户端. 开发者信息反馈请求处理功能

对于开发者的信息反馈请求，见图3.8

3.2.9 服务器端基本流程

简单描述服务器端处理请求的逻辑，见图3.9 . 注意管理员与服务器的交互也是通过网络连接的。但是管理员没有相应的 UI 界面。

3.2.10 服务器. 注册与登录请求处理具体流程

对于 2 类人员的登录与注册处理，见图3.10

此处的注册不包括管理员的注册，因为管理员的注册需要更多的权限，需要通过其他管理员添加进注册人员列表。

3.2.11 服务器. 信息查询与修改请求处理具体流程

对于 3 类人员的信息查询与修改处理，见图3.11

修改需要额外的权限，比如非本 app 的开发人员不能修改该 app 的名称，简介等等。

3.2.12 服务器. 应用上传与下载请求处理具体流程

对于应用的上传与下载处理见图3.12.

上传有应用审核的要求，此处没有具体实现，同时搜索界面也有根据用户的习惯或者统计数据的自动推荐功能。

3.2.13 服务器. 账户充值与提现请求处理具体流程

对于账户的管理见图3.13

可以添加不同的账户，这一点需要在后面提供统一的对外接口。

3.2.14 服务器. 应用信息查询与更新请求处理具体流程

对应用的信息查询与评价、评分流程见3.14.

应用的信息主要针对某个 app 的评论与评分等等基本信息，与个人信息区分开。

3.2.15 服务器. 管理员操作请求处理具体流程

对于管理员级别的操作，见图3.15

管理员直接通过命令行与服务器建立连接，同时需要检查每项操作的权限。

3.2.16 应用开发系统

此处结合开发者端、服务器端、用户端，统一处理流程为图3.16

3.3 功能结构设计

3.3.1 整体结构

整体结构见图3.17

服务器端提供 3 种不同的对外接口，分别针对开发者、用户、管理员。

3.3.2 客户端结构

客户端的结构见图3.18。

注意到，客户端可分为 5 个模块，其中注册/登录模块只会在其他模块中被调用，其他的模块相互独立。

3.3.3 服务器端结构

服务器端结构见图3.19

服务器端的内部储存采用数据库，分布且冗余，避免物理失效。

为了安全起见，对外统一有数据库统一接口，接口的修改只能是最高权限的管理员。

普通管理员对外接口也是通过数据库统一接口操作的，同时兼任第三方管理系统的管理员。

3.4 功能需求与程序代码的关系

[此处指的是不同的需求分配到哪些模块去实现。可按不同的端拆分此表]

表 3.1　功能需求与程序代码的关系表

需求编码	需求名称	开发者端应用管理模块	开发者端信息反馈模块	客
SRS_Dev_App_P01	开发者端. 应用管理	Y	·	
SRS_Dev_Info_P01	开发者端. 信息反馈	.	Y	
SRS_User_App_P01	客户端. 应用管理	.	·	
SRS_User_Info_P01	客户端. 信息反馈	·	·	

注：各项功能需求的实现与各个程序模块的分配关系

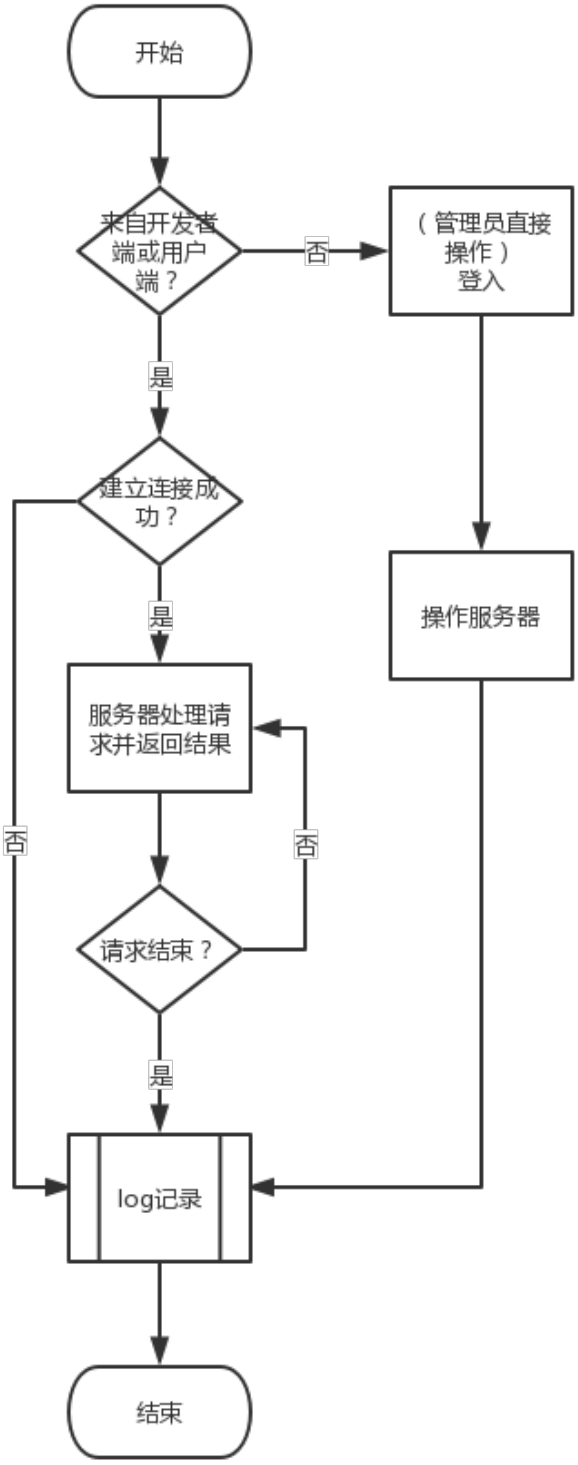
开始
来自开发者端或用户端？
否
（管理员直接操作）登入
是
建立连接成功？
操作服务器
是
服务器处理请求并返回结果
否
否
请求结束？
是
log记录
结束

图 3.2　系统基本流程

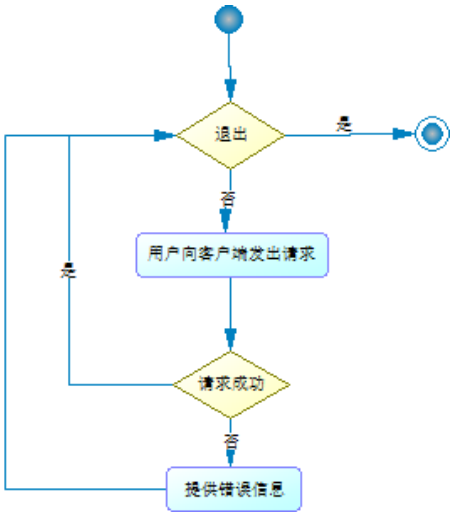

图 3.3　客户端基本流程

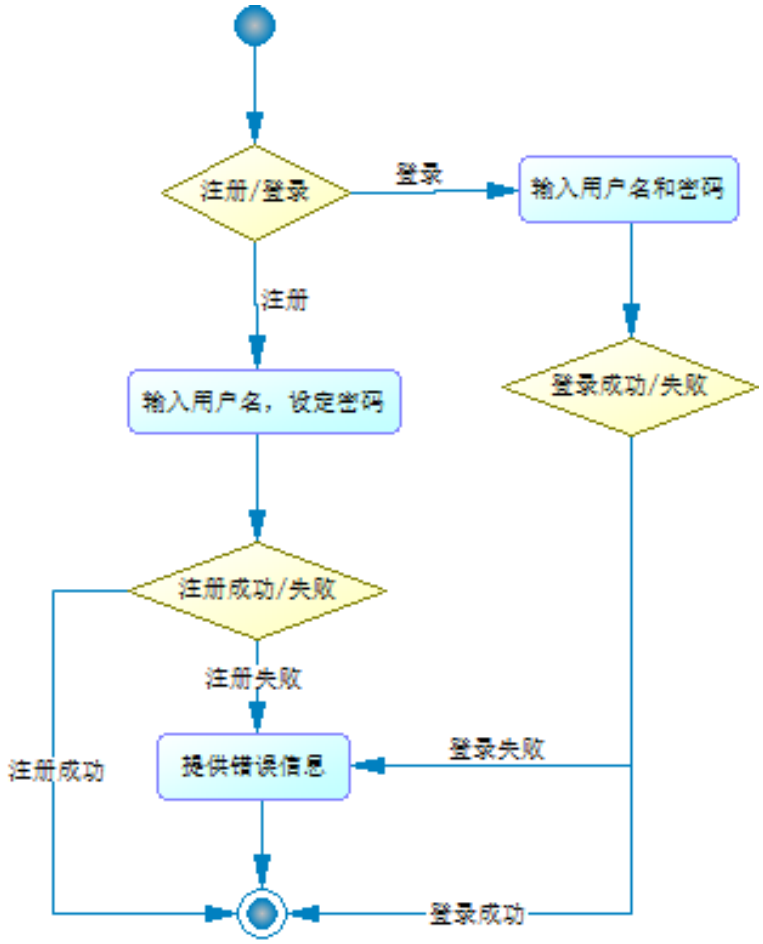

图 3.4　客户端. 注册与登录请求处理具体流程

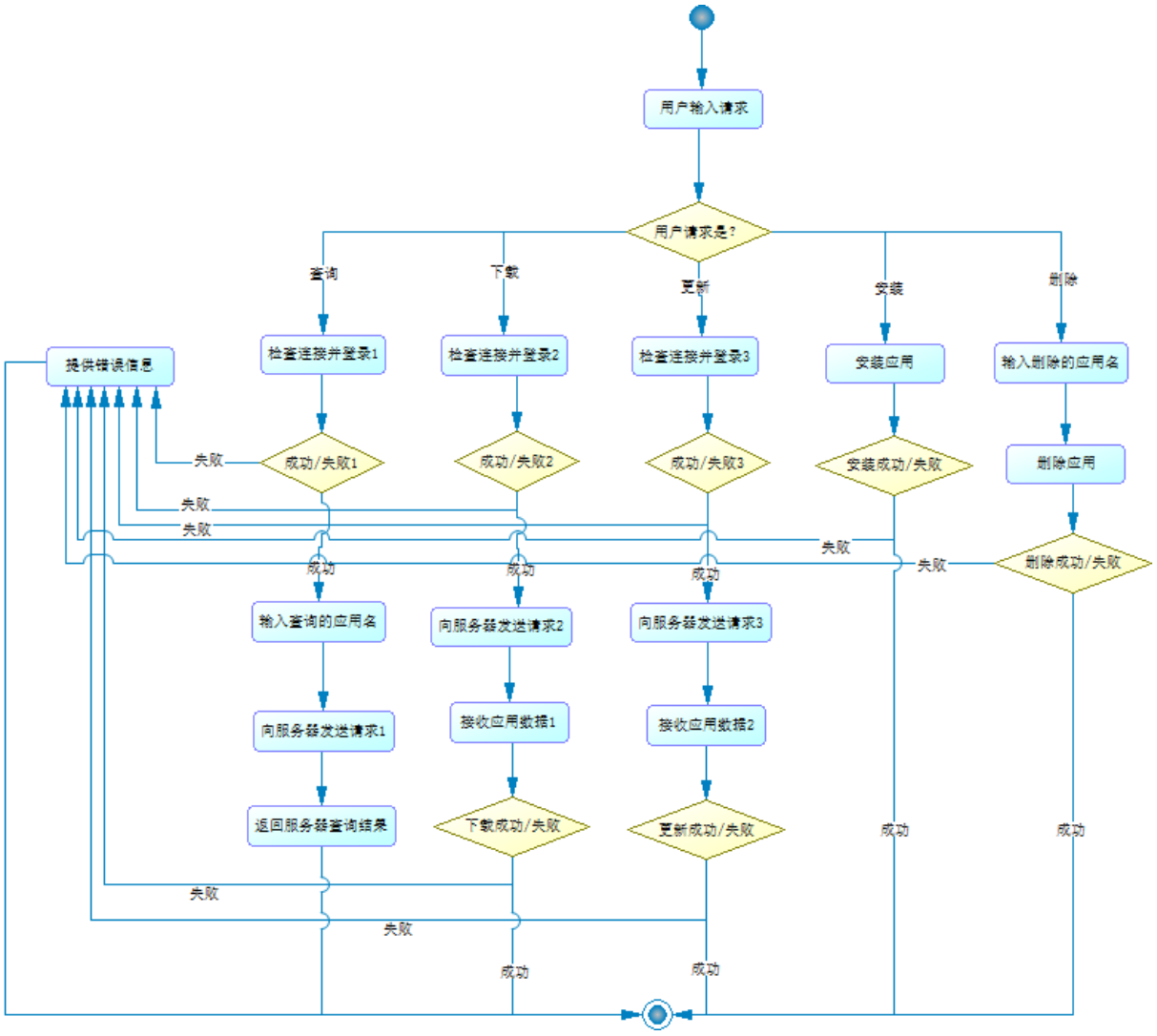

图 3.5　客户端. 普通用户应用管理请求处理具体流程

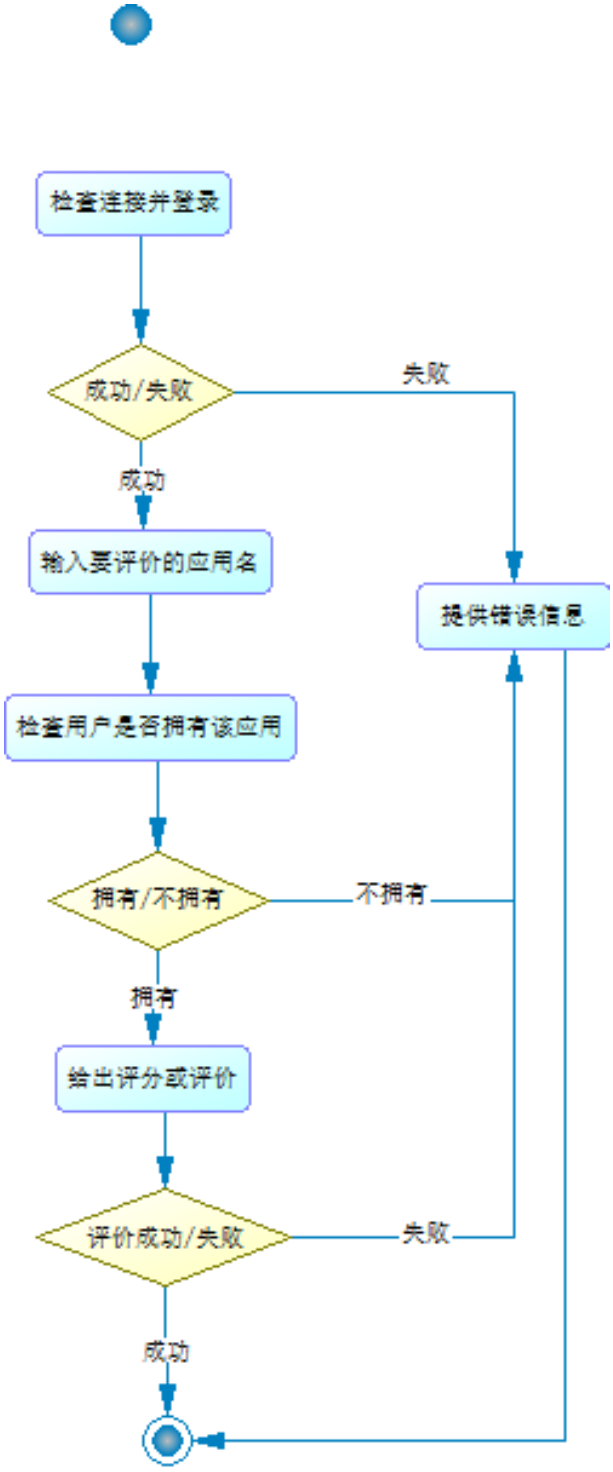

图 3.6　客户端. 普通用户信息反馈处理具体流程

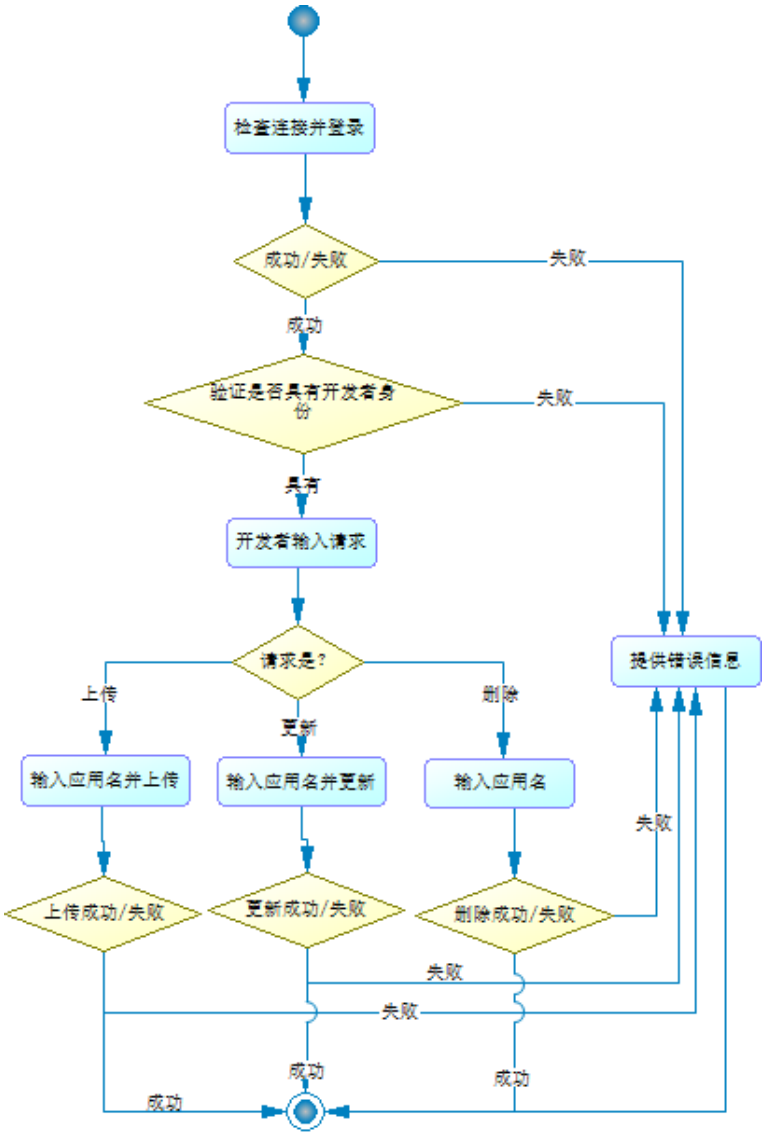

图 3.7　客户端. 开发者应用管理请求处理具体流程

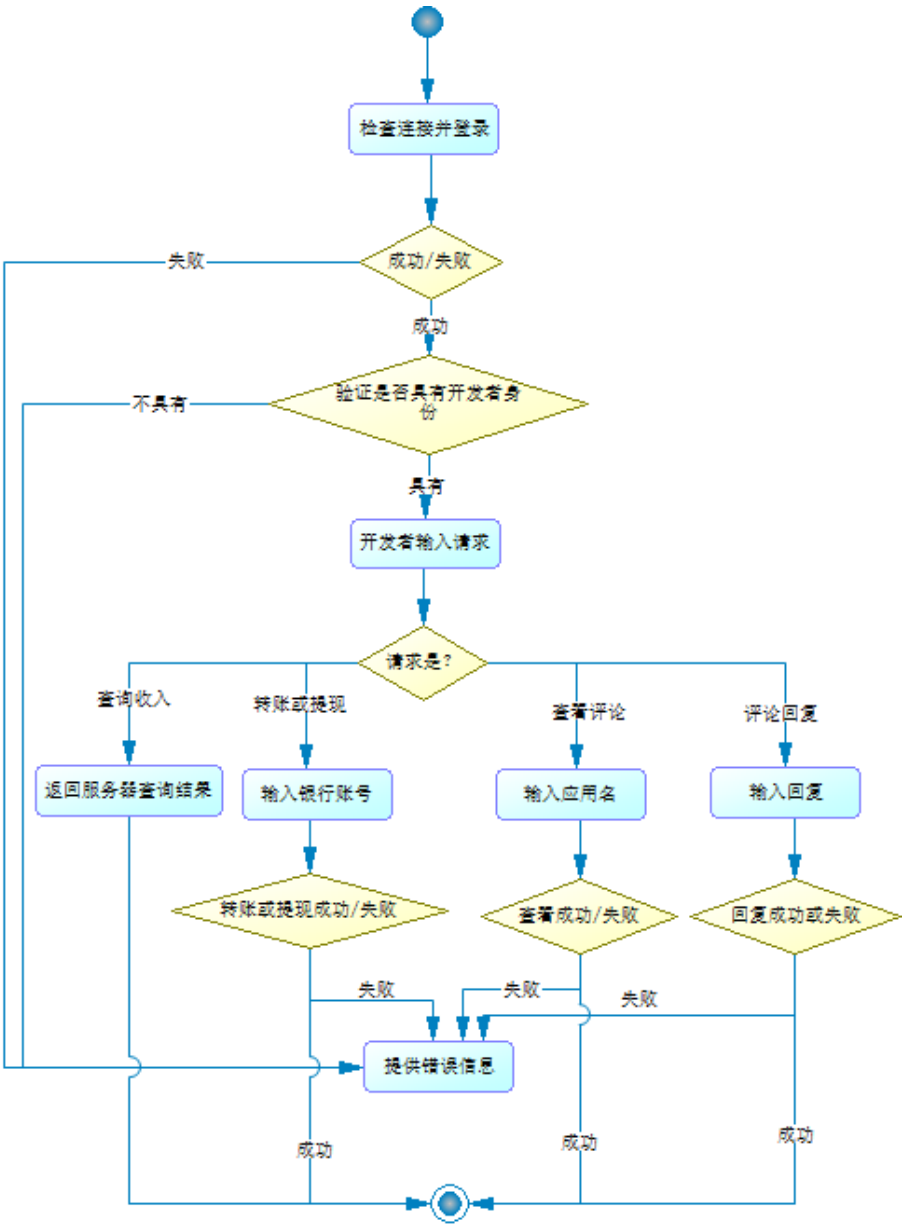

图 3.8　客户端. 开发者信息反馈请求处理具体流程

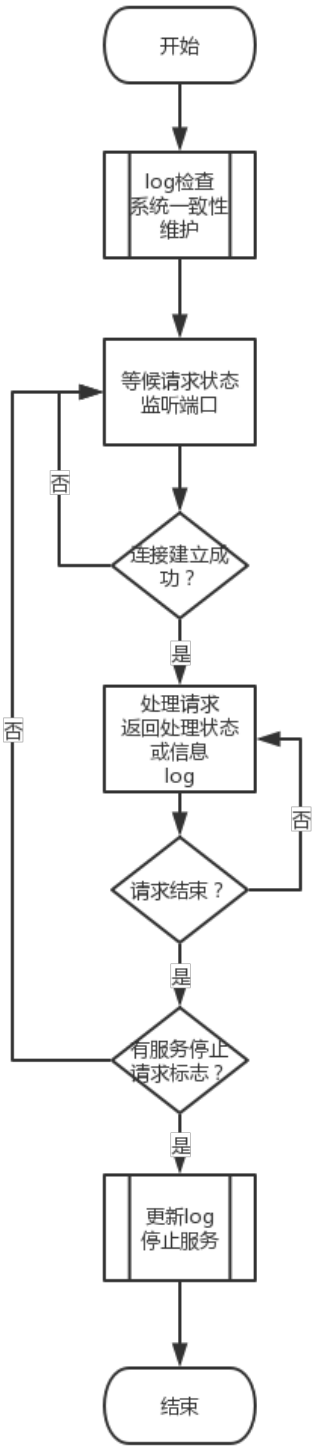

图 3.9　服务器端基本流程

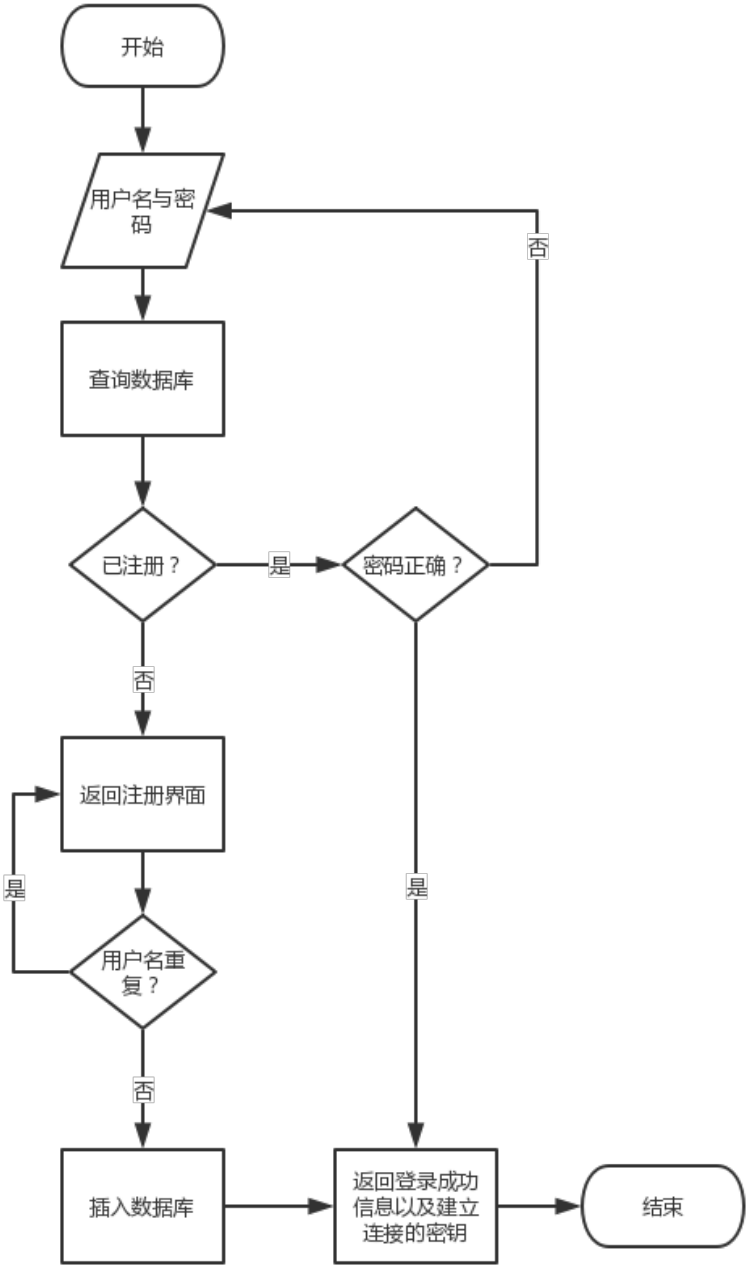

图 3.10　服务器. 注册与登录请求处理具体流程

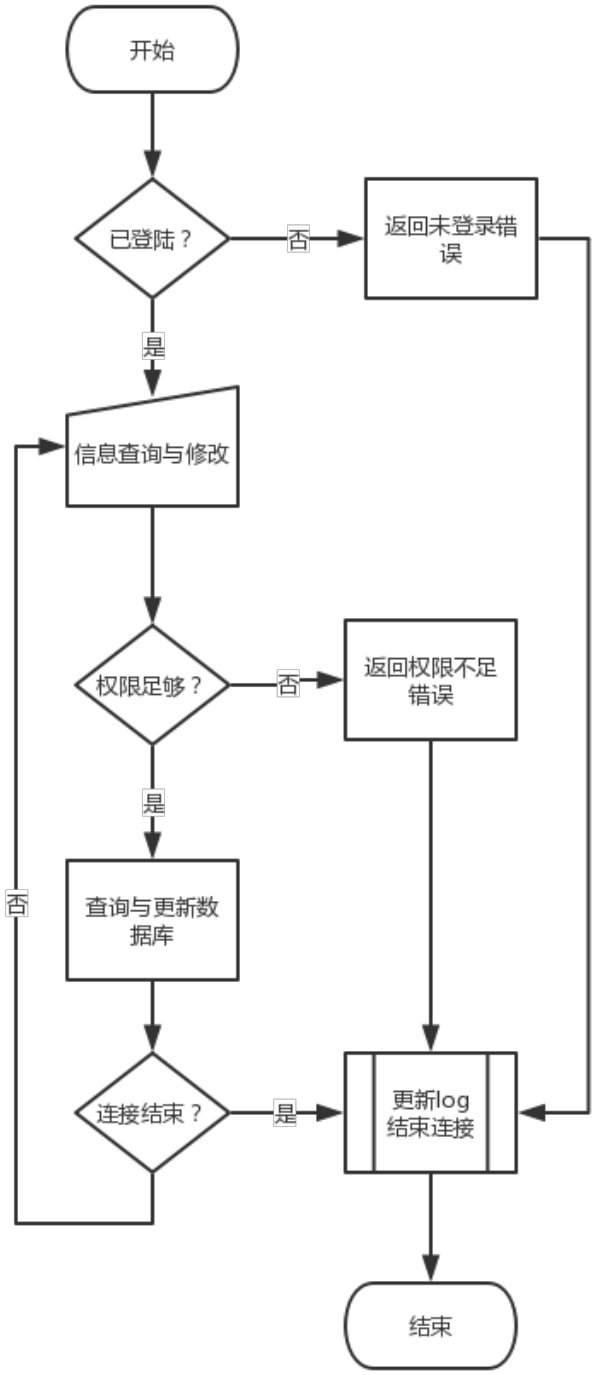

图 3.11　服务器. 信息查询与修改请求处理具体流程

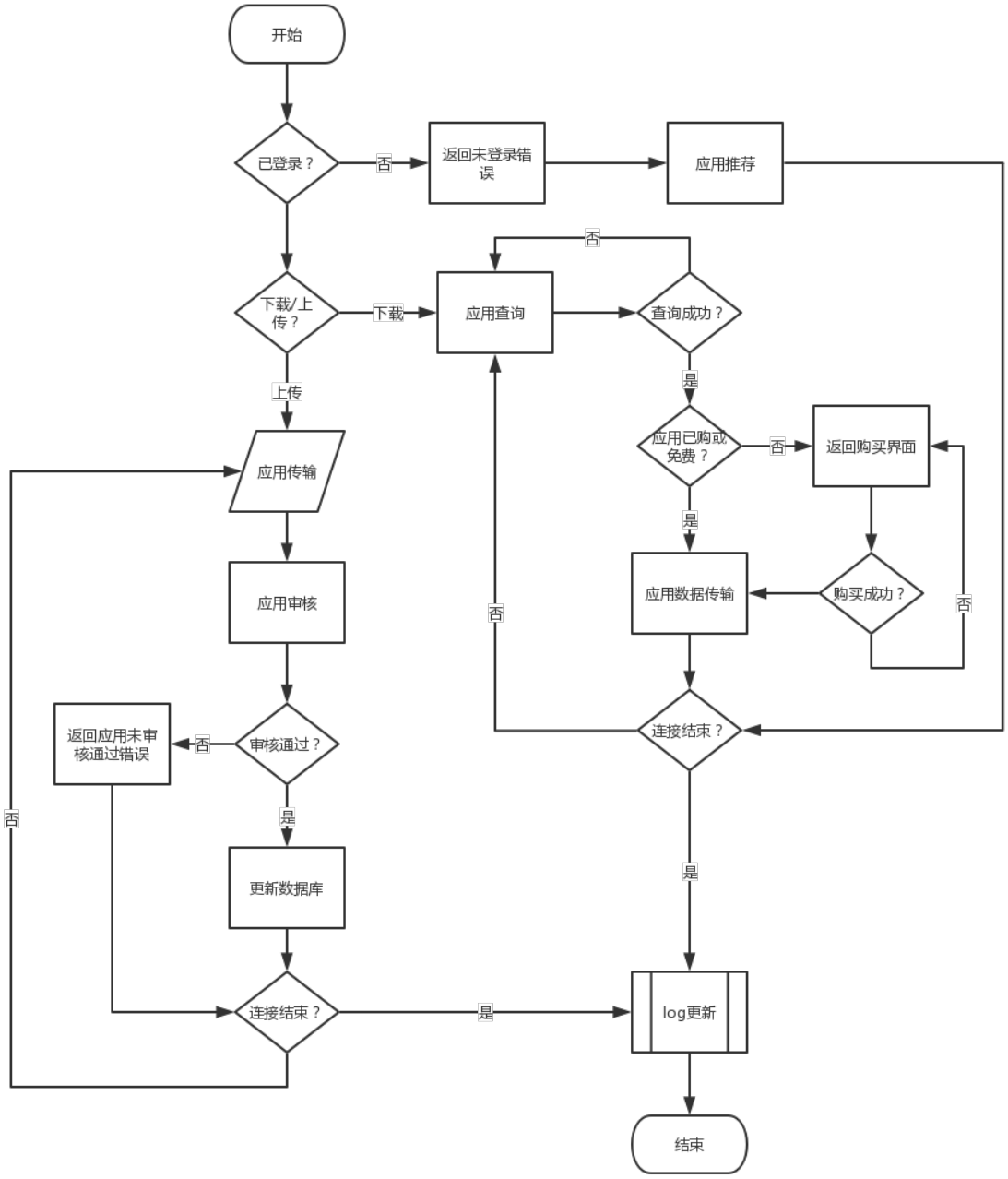

图 3.12　服务器. 应用上传与下载请求处理具体流程

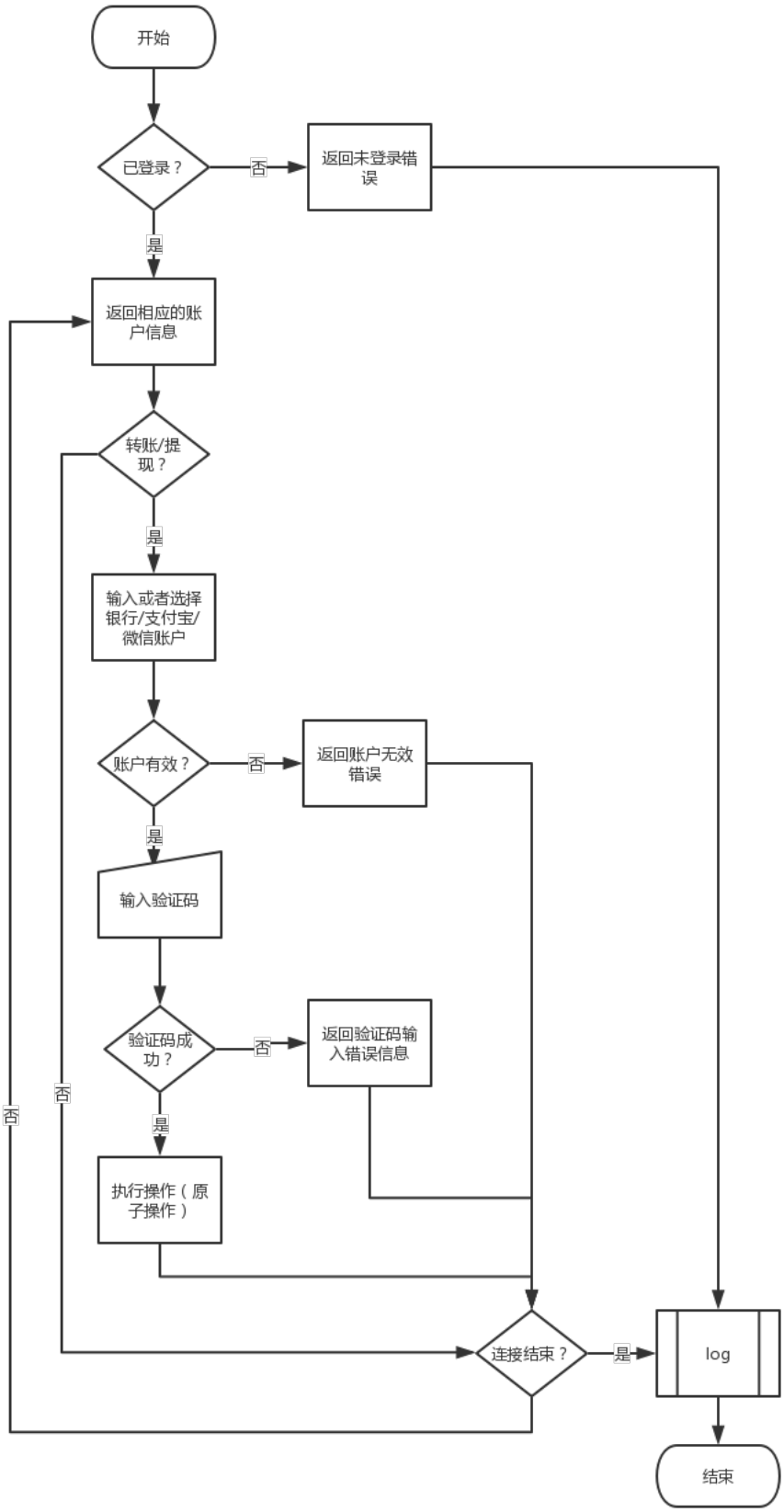

图 3.13　服务器. 账户充值与提现请求处理具体流程

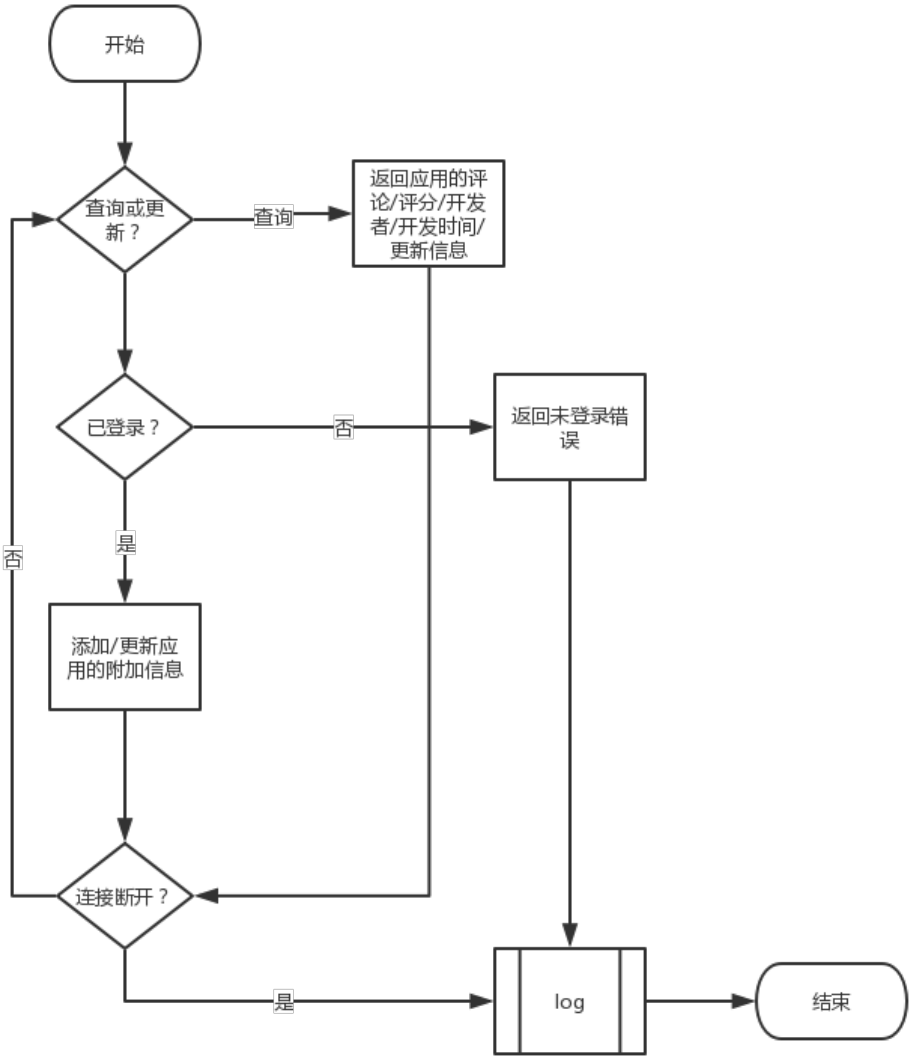

图 3.14　服务器. 应用信息查询与更新请求处理具体流程

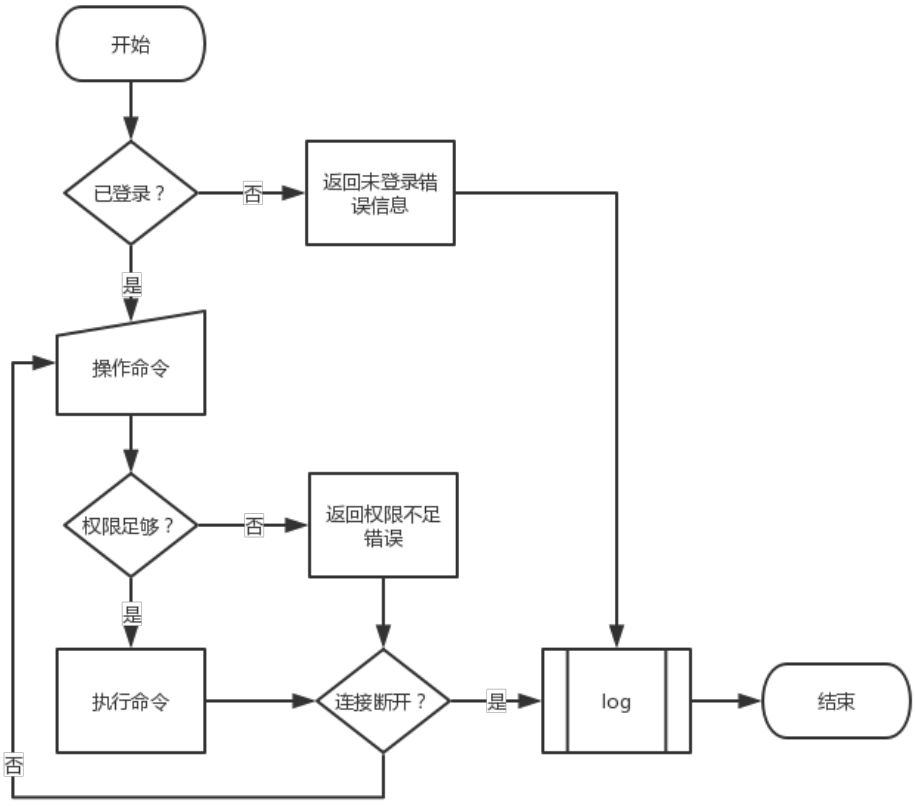

图 3.15　服务器. 管理员操作请求处理具体流程

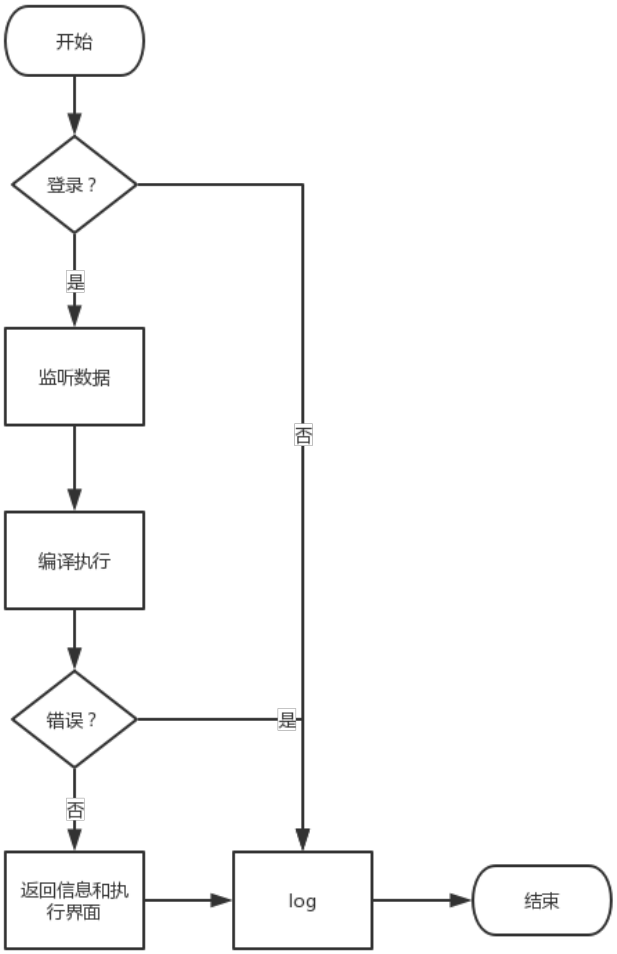

图 3.16　服务器. 管理员操作请求处理具体流程

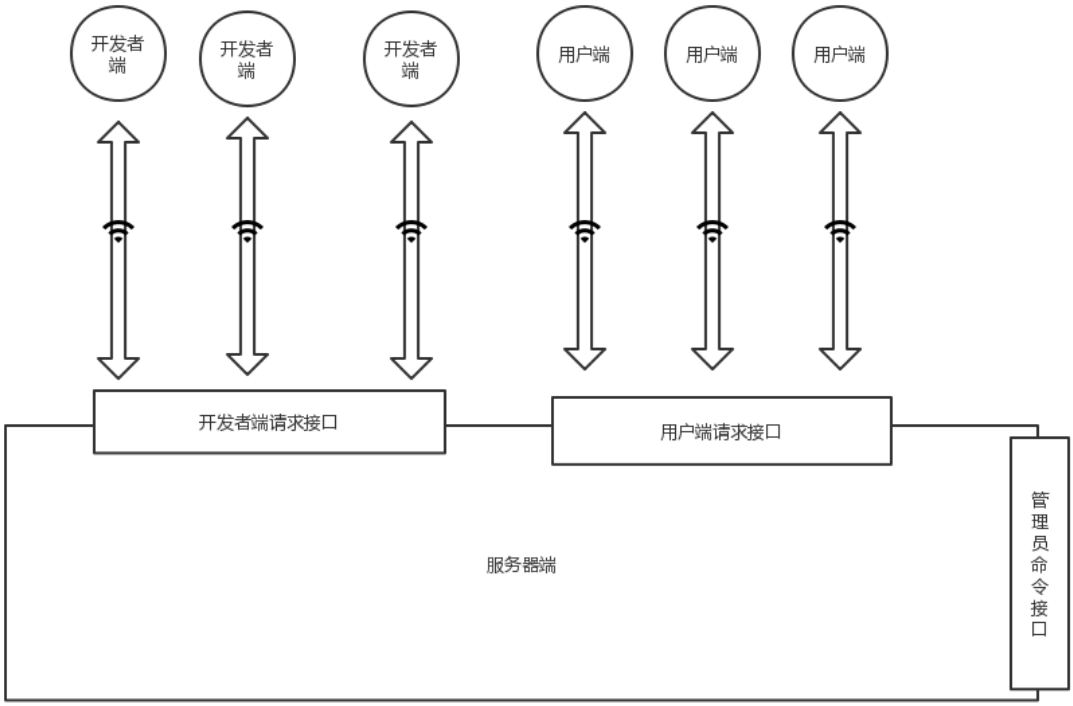

图 3.17　整体结构

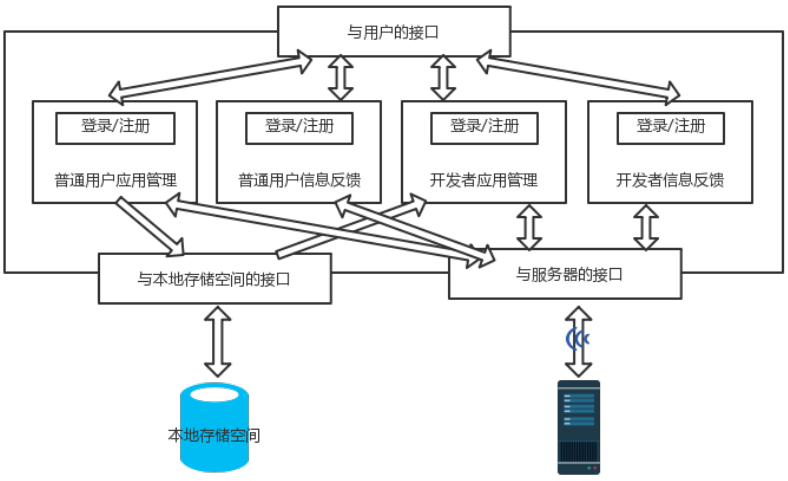

图 3.18　客户端结构

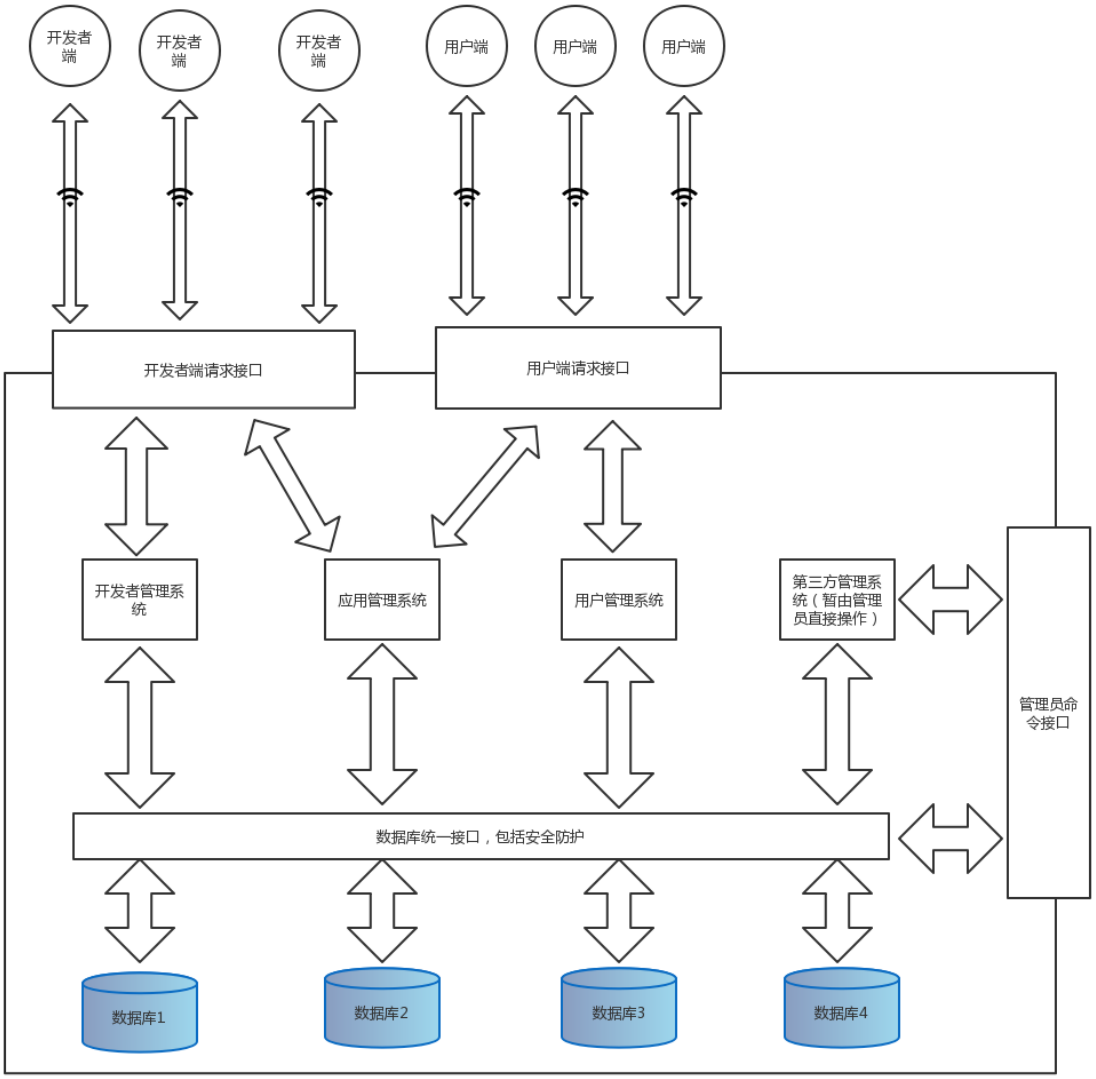
开发者端
开发者端
开发者端
用户端
用户端
用户端
开发者端请求接口
用户端请求接口
开发者管理系统
应用管理系统
用户管理系统
第三方管理系统（暂由管理员直接操作）
管理员命令接口
数据库统一接口，包括安全防护
数据库1
数据库2
数据库3
数据库4

图 3.19　服务器端结构

第 4 章　接口设计

4.1　外部接口

4.1.1　账户接口

账户的接口应该可以包括常见的电子账户，比如支付宝、微信、银行卡、Paypal 等等，在服务器内部形成统一的接口。需要实现的功能包括：电子账户添加、电子账户认证、电子账户余额查询、电子账户充值到商店虚拟账户、商店虚拟账户转账到电子账户。

提供给内部的接口应该针对开发者管理系统和用户管理系统。

接口的具体定义为

代码 4.1　账户接口

```
{
        string username //用户名
        string passwd //密码
        string account //账户
        vector<int> code //验证码
        float money //转账金额
        int error //错误码
}
```

4.1.2　安装/卸载接口

该接口用于客户端安装或卸载某个应用，只需提供该应用的安装/卸载程序的文件描述符或文件路径以及安装目录（如果是安装的话）即可。

接口的具体定义为

代码 4.2　安装/卸载接口

```
{
        string username //用户名
        vector<int> code //验证码
        string file //文件路径
}
```

4.1.3 用户接口

用户接口分为提供给普通用户和开发者的接口。

提供给普通用户的接口包括登录、注册、应用查询、应用购买、应用安装、应用删除、应用评分等等。

提供给开发者的接口包括登录、注册、应用上传、应用更新、应用删除、收入查询、转账提现、评论查看、评论回复等等。

分别参见表4.1和表4.2。

表 4.1 开发者与应用商店系统的接口

开发者需求	**输入**	**输出**
登录	账号、密码	登录成功与否
注册	账号、密码	注册成功与否
应用上传	本地应用	上传成功与否
应用更新	本地应用和待更新应用 id	更新成功与否
应用删除	待删除应用 id	删除成功与否
收入查询	无	收入
转账提现	银行账号	转账或提现成功与否
评论回复	回复内容	回复成功与否

表 4.2 普通用户与应用商店系统的接口

用户需求	**输入**	**输出**
登录	账号、密码	登录成功与否
注册	账号、密码	注册成功与否
应用查询	应用 id	相关的应用，以列表呈现
应用购买	待购买应用 id 和银行账户	购买成功与否
应用安装	待安装应用 id	安装成功与否
应用删除	待删除应用 id	删除成功与否
应用评分	待评价应用 id 和评分	评分成功与否
应用评价	待评价应用 id 和评论	评论成功与否

用户接口最终将以图形界面呈现。

接口的具体定义为

代码 4.3　安装/卸载接口

```
{
        string username //用户名
        string passwd //密码
        string command //指令
        string permission //权限等级
}
```

4.2　内部接口

4.2.1　客户端

如第三章的描述，客户端主要有这几个模块：注册或登录模块、普通用户应用管理模块、普通用户信息反馈模块、开发者应用管理模块、开发者信息反馈模块。

后面 4 个模块相对独立，相互之间没有交互。注册或登录模块会作为子模块被后面 4 个模块调用。

接口的具体定义为

代码 4.4　安装/卸载接口

```
{
        vector<int> number //终端编码
        vector<int> code //验证码
        int command //指令编号
        data_type data //数据
}
```

4.2.2　服务器端与数据库

服务器端内部提供数据库统一接口，提供给开发者管理系统、应用管理系统、用户管理系统、第三方管理系统使用，目的是提高安全性与简便。

提供高阶的增删改查接口，不需要直接使用 PL/SQL.

不同的功能需要有不同的权限，比如修改删除需要的权限高于查询，同时，只有管理员才有直接使用 PL/SQL 查询的权限。另外，应该提供方便的增加高阶功能的接口。

接口的具体定义为

代码 4.5　安装/卸载接口

```
{
        vector<int> number //服务器编码
        vector<int> code //验证码
        int permission //权限等级
        string command //指令
        data_type data //数据
}
```

第 5 章　数据结构设计

5.1　逻辑结构设计

5.1.1　客户端数据结构

客户端在运行过程中，需要维护的信息如下：
生命周期为整个程序运行过程的信息有：

- 用户名，如果尚未登录则为空
- 当前请求
- 请求结果

生命周期为某个请求处理过程的信息有：

1. 普通用户应用管理：数据结构 1

    - 目标应用名

2. 普通用户信息反馈：数据结构 2

    - 目标应用名
    - 评价或评分

3. 开发者应用管理：数据结构 3

    - 目标应用名

4. 开发者信息反馈：数据结构 4

    - 目标应用名

- 回复的评论
- 银行账号

根据具体的请求，以上的数据结构还可再进一步细化。

5.1.2 服务器端数据结构

数据部分使用 Oracle 数据库存储，并不需要特别的数据结构。

5.2 物理结构设计

各数据结构无特殊物理结构要求。

5.3 数据结构与程序模块的关系

[此处指的是不同的数据结构分配到哪些模块去实现。可按不同的端拆分此表]

表 5.1 数据结构与程序代码的关系表

·	普通用户应用管理	普通用户信息反馈	开发者应用管理	开发者信息反馈
数据结构 1	Y	·	·	·
数据结构 2	·	Y	·	·
数据结构 3	·	·	Y	·
数据结构 4	·	·	·	Y

注：各项数据结构的实现与各个程序模块的分配关系

第 6 章　数据库设计

6.1　数据库环境说明

6.1.1　检索速度 v.s. 稳定性

本应用并不强调检索的速度，因为应用的数量并不会过多，所以更多地从稳定性考虑。

从稳定性考虑，本系统的数据系统采用 Oracle 12 数据库系统；使用 centOS 作为操作系统。

6.1.2　数据一致性考虑

本系统同时也不强调一致性，应用的开发、应用的下载并不是冲突的操作。用户下载的应用仅仅是系统中的最新版本即可。

6.2　数据库的命名规则

允许单词缩写，缩写的要求见表1

表名为单数,当前阶段设计的表有 app,user,developer,people,dev_app,user_own_app,comment.

字段无需加上前缀

6.3　逻辑设计

基本的实体关系逻辑模型见图6.1

6.4　物理设计

6.4.1　数据库产品

使用 oracle 12c 数据库，要求是分布的且冗余。

6.4.2　实体属性、类型、精度

6.4.2.1　app 数据表设计

app 表设计见图6.1

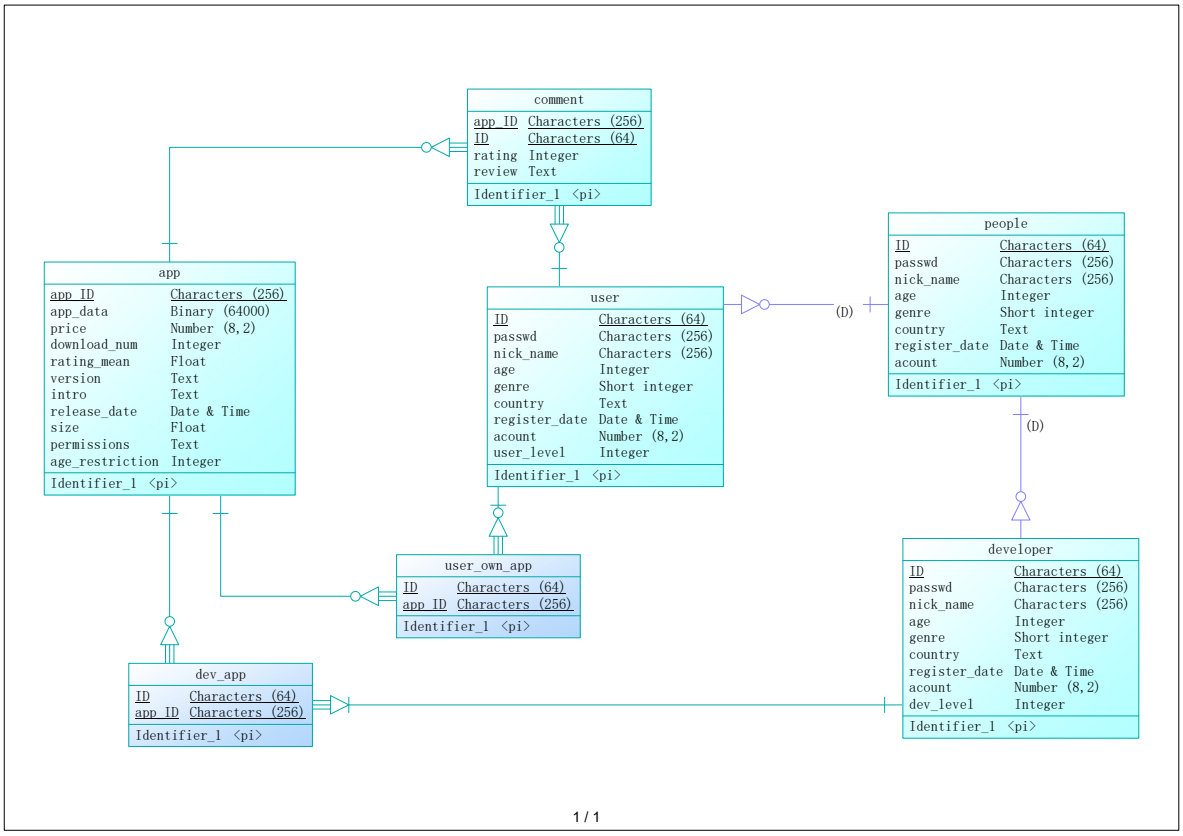

图 6.1　逻辑模型图

6.4.2.2　user 数据表设计

user 数据表设计见图6.2

6.4.2.3　developer 数据表设计

developer 数据表设计见表6.3

6.4.2.4　comment 数据表设计

评论、评分数据表设计见表6.4

6.4.2.5　dev_app 数据表设计

开发者开发的 app 表设计见表6.5

表 6.1　app 数据表 Users 设计

字段名	类型	大小	说明	备注
app_ID	char	256	app 的全局唯一标识符	主键
app_data	binary	·	app 数据	·
price	number	·	app 的价格	·
download_num	int	·	下载数量	·
rating_mean	float	·	平均评分	·
version	text	·	版本号	·
intro	text	·	简介	·
release_date	Date & Time	·	发行时间	·
size	float	·	app 数据大小	·
permissions	text	·	权限要求	·
age_restriction	int	·	用户年龄限制	·

表 6.2　user 数据表设计

字段名	类型	大小	说明	备注
ID	char	64	人员 ID	主键
passwd	char	256	加密后的密码	·
nick_name	char	256	昵称	·
age	int	·	年龄	考虑到年龄限制
genre	char	1	性别	·
country	text	·	国家	考虑到地域限制
register_date	Date& Time	·	注册时间	·
acount	number	·	账户金额	·
user_level	int	·	用户的等级	·

表 6.3　developer 数据表设计

字段名	类型	大小	说明	备注
ID	char	64	人员 ID	主键
passwd	char	256	加密后的密码	·
nick_name	char	256	昵称	·
age	int	·	年龄	考虑到年龄限制
genre	char	1	性别	·
country	text	·	国家	考虑到地域限制
register_date	Date& Time	·	注册时间	·
acount	number	·	账户金额	·
dev_level	int	·	开发者的等级	·

表 6.4　comment 数据表设计

字段名	类型	大小	说明	备注
app_ID	char	256	app 的标识符	外键，来自 app 表
ID	char	64	人员的 ID	外键，来自 user 表
rating	int	·	评分	0-5
review	text	·	评论	·

表 6.5　dev_app 数据表设计

字段名	类型	大小	说明	备注
app_ID	char	256	app 的标识符	外键，来自 app 表
ID	char	64	人员的 ID	外键，来自 user 表

表 6.6　user_own_app 数据表设计

字段名	类型	大小	说明	备注
app_ID	char	256	app 的标识符	外键，来自 app 表
ID	char	64	人员的 ID	外键，来自 user 表

6.4.2.6 user_own_app 数据表设计

用户拥有的（已购买或者已下载的）app 表设计见表6.6

6.5 安全性设计

备份和容灾设计。

6.6 数据库管理与维护说明

对于数据库的维护，随时对数据库中的信息加以调试和保存备份。同样需要个工作人员进行系统的分析和用户的反馈，对系统进行升级以及功能的完善。同时保证系统安全有序的运行。

第 7 章　界面设计

7.1　客户端界面

之前已经多次提到过，普通用户和开发者共用一个客户端。

界面示例见图7.1、7.2、7.3、7.4。

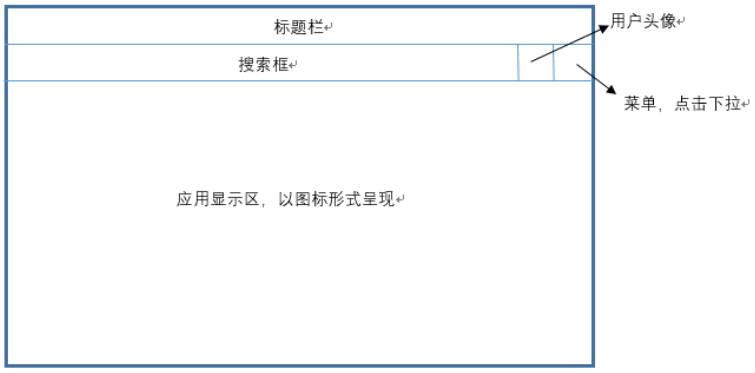

图 7.1　桌面端应用商店主界面

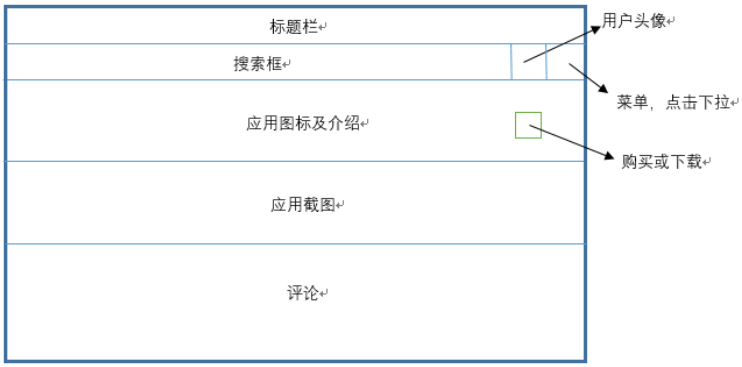

图 7.2　桌面端应用商店应用界面

以上只列出了部分用户界面。最终的用户界面可能与目前的示例有所偏差。

7.2　服务器端界面

服务器端不需要界面。管理员直接使用命令行交互。

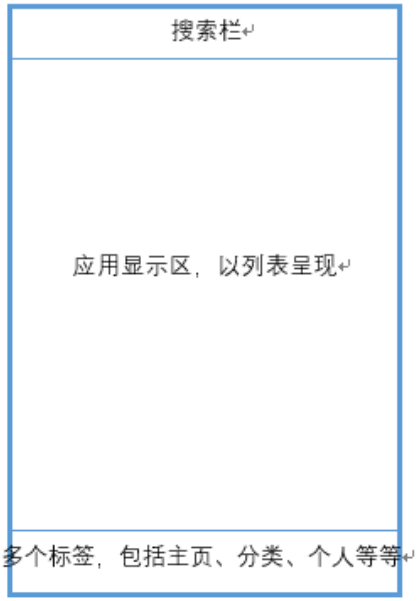

图 7.3　移动端应用商店主界面

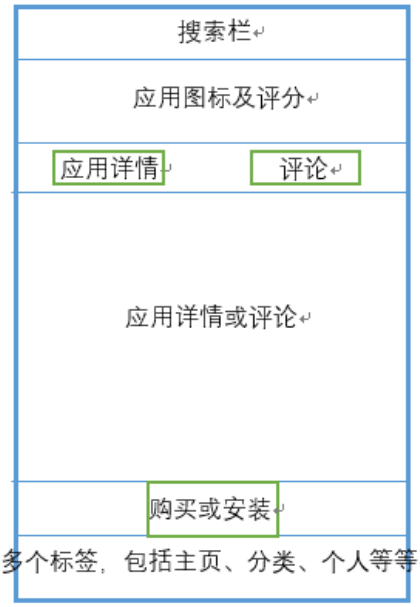

图 7.4　移动端应用商店应用界面

第 8 章　出错处理设计

8.1　客户端出错处理

客户端的错误可以分为两类：由于网络异常导致的和服务器通信中断、用户发出的请求未能得到满足。

第一类错误应提醒用户检查网络状态，第二类错误是因为用户不具备完成其请求的条件，如试图评价一个用户不拥有的应用，此类错误由服务器返回，提醒用户重新操作即可。

另外，客户端可能会由于未知的原因而崩溃，对于这类问题可以设置一个日志来存储客户端的行为，以便重启客户端后进行恢复以及向服务器发送崩溃记录，但也可不做处理，因为应用商店不是类似办公软件的程序，不会涉及到具有很大价值的数据。

8.2　服务器端出错处理

服务器有两种避免重大错误的方式，一是使用 log, 二是数据库采用分布式冗余设计。

紧急错误出现时，立即停止系统，并且上线备份系统。

通常的错误可以使用 log 自动或者人工修正，管理员可以添加自动修正规则。

非常重大的人为错误或者物理失效，仍然有分布式冗余的数据库作为保证。

第 9 章　安全保密设计

9.1　客户端安全设计

客户端涉及的重要数据只有用户名和密码以及银行账号。

可以对这些信息加密传输，具体方法如下：

1. 服务器端通过 RSA 公钥加密系统生成一对公钥和私钥，并将公钥派发给客户端。
2. 客户端在传输上述重要信息前，首先生成一个对称密钥。
3. 客户端使用服务器的公钥对生成的对称密钥加密，然后发送给服务器。
4. 服务器使用私钥对收到的信息解密，得到客户端的对称密钥。
5. 客户端和服务器可以通过对称密钥进行通信。

9.2　服务器端安全设计

最重要的数据在数据库，所以最关键的安全防护在于数据库统一接口，避免人为失误，同时也能避免网络攻击。

同时另一个是用户端接口和开发者端接口，即暴露给外面的接口有限，而且经过严格的设计与攻击检测。

第 10 章　维护设计

10.1　客户端维护

客户端需要记录的信息只有：已安装应用及其卸载程序路径、已下载但未安装的应用安装程序路径。

10.2　服务器端维护

所有不符合常规的错误都会形成 log 定期发送给管理员。

同时每隔固定的时间检测整个系统的稳定性。

参考文献