**[麋鹿寻途]**

**《软件概要设计说明书》**

团队名称： 三带一队

指导教师： 代祖华

完成时间：2020年6月5日

目录

[1.引言 3](#_Toc42272054)

[1.1. 编写目的 3](#_Toc42272055)

[1.2. 背景 3](#_Toc42272056)

[1.3. 定义 3](#_Toc42272057)

[1.4. 参考资料 4](#_Toc42272058)

[2.总体设计 4](#_Toc42272059)

[2.1. 需求规定 4](#_Toc42272060)

[2.2. 运行环境 4](#_Toc42272061)

[2.3. 基本设计概念和处理流程 5](#_Toc42272062)

[2.4. 结构 5](#_Toc42272063)

[2.5. 功能器求与程序的关系 6](#_Toc42272064)

[2.6. 人工处理过程 7](#_Toc42272065)

[2.7. 尚未解决的问题 7](#_Toc42272066)

[3.接口设计 7](#_Toc42272067)

[3.1. 用户接口 7](#_Toc42272068)

[3.2. 外部接口 8](#_Toc42272069)

[3.3. 内部接口 8](#_Toc42272070)

[4.运行设计 8](#_Toc42272071)

[4.1. 运行模块组合 8](#_Toc42272072)

[4.2. 运行控制 8](#_Toc42272073)

[4.3. 运行时间 8](#_Toc42272074)

[5.系统数据结构设计 9](#_Toc42272075)

[5.1. 逻辑结构设计要点 9](#_Toc42272076)

[5.2. 物理结构设计要点 10](#_Toc42272077)

[5.3. 数据结构与程序的关系 10](#_Toc42272078)

[6.系统出错处理设计 10](#_Toc42272079)

[6.1. 出错信息 10](#_Toc42272080)

[6.2. 补救措施 11](#_Toc42272081)

[6.3. 系统维护设计 11](#_Toc42272082)

[7.软件重用方案 11](#_Toc42272083)

[7.1. 知识重用 11](#_Toc42272084)

[7.2. 方法和标准的重用 11](#_Toc42272085)

[7.3. 软件成分的重用 11](#_Toc42272086)

[8.关键类的设计 12](#_Toc42272087)

# 1.引言

## 1.1. 编写目的

编写本说明书的目的在于概要描述麋鹿寻途系统的软件设计模式与软件总体结构、系统数据库的逻辑结构、软件重用方案与关键类的重点服务设计等，并方便系统开发人员根据这些设计内容着手开发。

该设计说明书的预期读者：系统开发人员、系统评审人员。

## 1.2. 背景

待开发软件系统的名称：麋鹿寻途系统

项目任务的提出者：三带一队

项目的开发者：三带一队

项目的用户：老人及其亲属、民警

运行该软件的计算站（中心）：Tomcat服务器、阿里云服务器（暂定）

## 1.3. 定义

Tomcat：Tomcat是Apache软件基金会（Apache Software Foundation）的Jakarta项目中的一个核心项目，由Apache、Sun和其他一些公司及个人共同开发而成。Tomcat服务器是一个免费的开放源代码的Web应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP程序的首选。

B/S：B/S结构（Browser/Server，浏览器/服务器模式），是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器，如Netscape Navigator或Internet Explorer，服务器安装SQL Server、Oracle、MYSQL等数据库。浏览器通过Web Server同数据库进行数据交互。

MVC：MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

JDBC：JDBC全名为Java Database Connectivity，它是Java语言中用来规范客户端程序如何来访问数据库的应用程序接口，提供了诸如查询和更新数据库中数据的方法。

JDK：JDK(Java Development Kit) 是 Java 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的java应用程序。

## 1.4. 参考资料

[1]《软件概要设计说明书》 （计算机软件产品开发文件编制指南标准GB/T 8567-1988）

[2]《软件工程导论（第6版）》 （作者：张海藩、牟永敏 ISBN：9787302330981 出版日期：2013年8月 出版单位：清华大学出版社）

[3]《构建之法：现代软件工程（第三版）》 （作者：邹欣 ISBN：9787115460769 出版日期：2017年7月 出版单位：人民邮电出版社）

# 2.总体设计

## 2.1. 需求规定

本系统旨在为警察提供一个可以快速获取走失老年人家庭信息的平台，目的是帮助走失的老年人更快的寻找到家人，在保障了社会的和谐稳定的同时提高社会公民的幸福感与安全感。其中系统用户主要为民警、老人以及家属，老人以及家属将老人的面部信息与家庭的联系信息上传至系统，一旦出现老人走失情况，警察能够利用本系统识别老人的面部信息，从数据库中快速获取老人的家属联系方式或者其现住地址等信息，从而将老人安全送回。

系统采用了MVC设计模式构建基于B/S架构的系统平台，系统的主要输入项目为老人的面部信息与家庭联系信息，主要输出项目为老人的家庭联系信息。

系统主要分为警务模块、用户模块与管理员模块这三大模块，其中用户家庭模块负责老年人的面部信息与家庭信息的录入与添加、删除与修改老年人的相关信息；警务模块负责对老人进行人脸识别并在查找到老年人信息时显示老年人的家庭信息；管理员模块负责完成管理员对信息的查询与修改操作。

## 2.2. 运行环境

* + 1. 服务器端：

硬件环境：单核处理器 内存2G以上 硬盘20G以上

软件环境：Windows10、Linux IE7.0以上、Chrome内核浏览器 MySQL8.0 JDK1.8

* + 1. 客户端：

硬件环境：Intel第二代处理器（或同类性能处理器）以上 内存2G以上 硬盘20G以上

软件环境：WindowsXP+ IE7.0以上、Chrome内核浏览器

## 2.3. 基本设计概念和处理流程

老人以及家属首先进行注册登录用户家庭端，将老人的面部信息与家庭的联系信息提交至系统，若数据提交错误或家庭信息与老人信息有所更改，则在信息管理页面进行添加、删除或修改。在警察发现老人走失的情况下，警察登录系统的警务端，通过人脸识别从数据库中快速获取老人的家属联系方式或者其现住地址。管理员可以登录系统，对系统中数据进行查询与修改。

系统的处理流程如下图1所示：

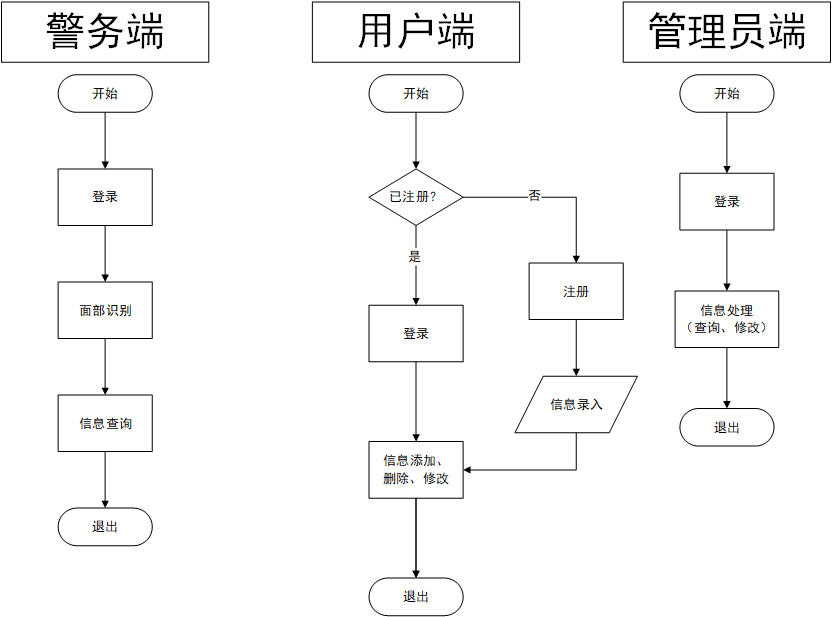


图1. 系统处理流程图

## 2.4. 结构

系统的层次结构图如下图2所示：

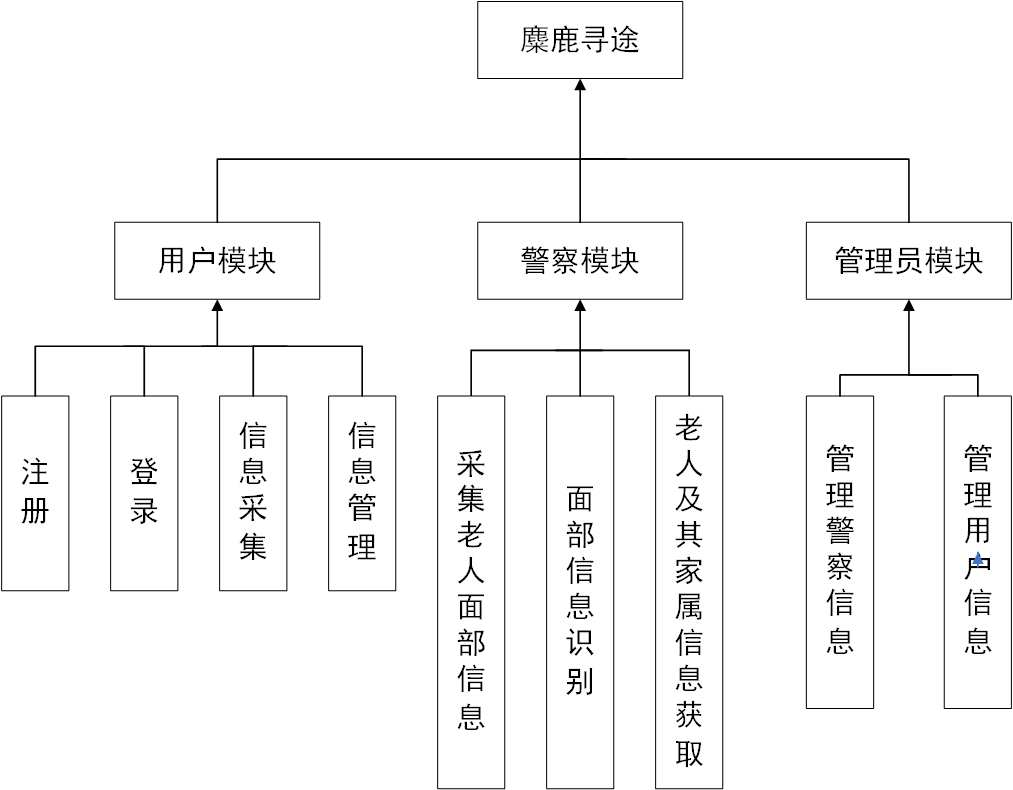


图2. 系统结构图

对于用户模块，其下的主要功能有用户的注册、登录、老人及家属信息的采集与相关信息的管理。

对于警察模块，其下的主要功能为老人面部信息的采集与识别与相关的信息获取。

对于管理员模块，其下的主要功能为用户信息与警察信息的管理。

## 2.5. 功能器求与程序的关系

|  |  |  |  |
| --- | --- | --- | --- |
|  | 用户模块 | 警察模块 | 管理员模块 |
| 用户的注册 | √ |  |  |
| 用户的登录 | √ |  |  |
| 用户家属信息采集 | √ |  |  |
| 用户信息管理 | √ |  |  |
| 老人面部信息采集 | √ | √ |  |
| 面部信息识别 |  | √ |  |
| 老人家属信息获取 |  | √ |  |
| 警察信息管理 |  |  | √ |
| 用户信息管理 |  |  | √ |

## 2.6. 人工处理过程

本软件系统的工作过程中的人工处理过程如下：

1. 用户端老人及家属信息的录入需要用户人工进行操作；
2. 用户端老人及家属信息的添加、删除与修改需要人工进行操作；
3. 警务端老人的面部信息采集需要警察进行人工操作；
4. 管理员端管理用户与警察的信息需要管理员人工操作。

## 2.7. 尚未解决的问题

1. 人脸识别功能的实现；
2. 在数据量大的情况下使用时可能会有信息的延时或者卡顿情况。

# 3.接口设计

## 3.1. 用户接口

系统界面设计主要通过墨刀原型设计工具实现，界面的设计应当简洁明了，各个模块的界限分明，且功能模块的入口清晰，对于基本的错误信息需要有出错处理。总的来说，系统的界面需遵循可靠性、简单性与易学性。

用户访问各个模块的状态如下图3：

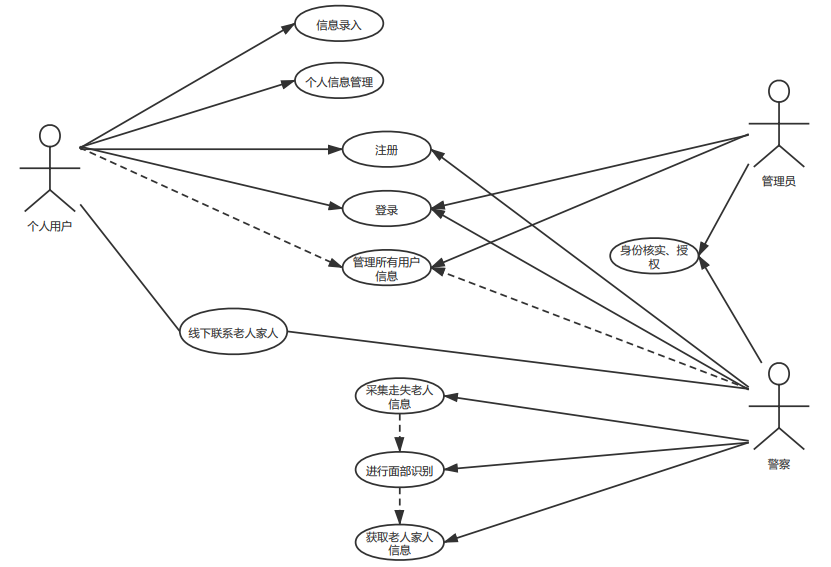


图3. 系统用例图

## 3.2. 外部接口

本系统建立在软件的基础上，且暂时没有相互支持的软件，因此没有软硬件之间与各支持软件之间的接口。

## 3.3. 内部接口

系统基于B/S架构的MVC设计模式，数据库与系统之间通过JDBC进行连接，各个模块之间相互独立，且各模块中各个类与方法独立性也较强。

# 4.运行设计

## 4.1. 运行模块组合

说明对系统施加不同的外界运行控制时所引起的各种不同的运行模块组合，说明每种运行所历经的内部模块和支持软件。

1. 用户的注册功能调用用户模块，完成用户的注册操作并跳转至用户家属信息采集服务；
2. 用户的登录功能调用用户模块，完成用户登录名与密码的确认，在确认正确后完成用户的登录；
3. 用户家属信息采集功能调用用户模块，实现用户家属信息与老人信息的采集；
4. 用户信息管理功能调用用户模块，实现用户对已有家属信息的添加、修改与删除操作；
5. 老人面部信息采集功能调用警务模块，实现对老人面部信息的采集并跳转至面部信息识别服务；
6. 面部信息识别功能调用警务模块，实现根据上一步中采集到的老人面部信息与现有数据库中的老人面部信息进行对比，最终匹配老人的家属信息；
7. 警察与用户信息管理功能调用管理员模块，实现对用户与警察信息的管理。

## 4.2. 运行控制

在本系统中的外界控制均通过用户对如按钮、链接等的实体点击来实现。

## 4.3. 运行时间

在系统中警务模块与用户模块将占用数据库资源的大部分时间，其中面部识别服务与面部信息上传功能占用的资源最多。

# 5.系统数据结构设计

## 5.1. 逻辑结构设计要点

数据库E-R图如下图4所示：

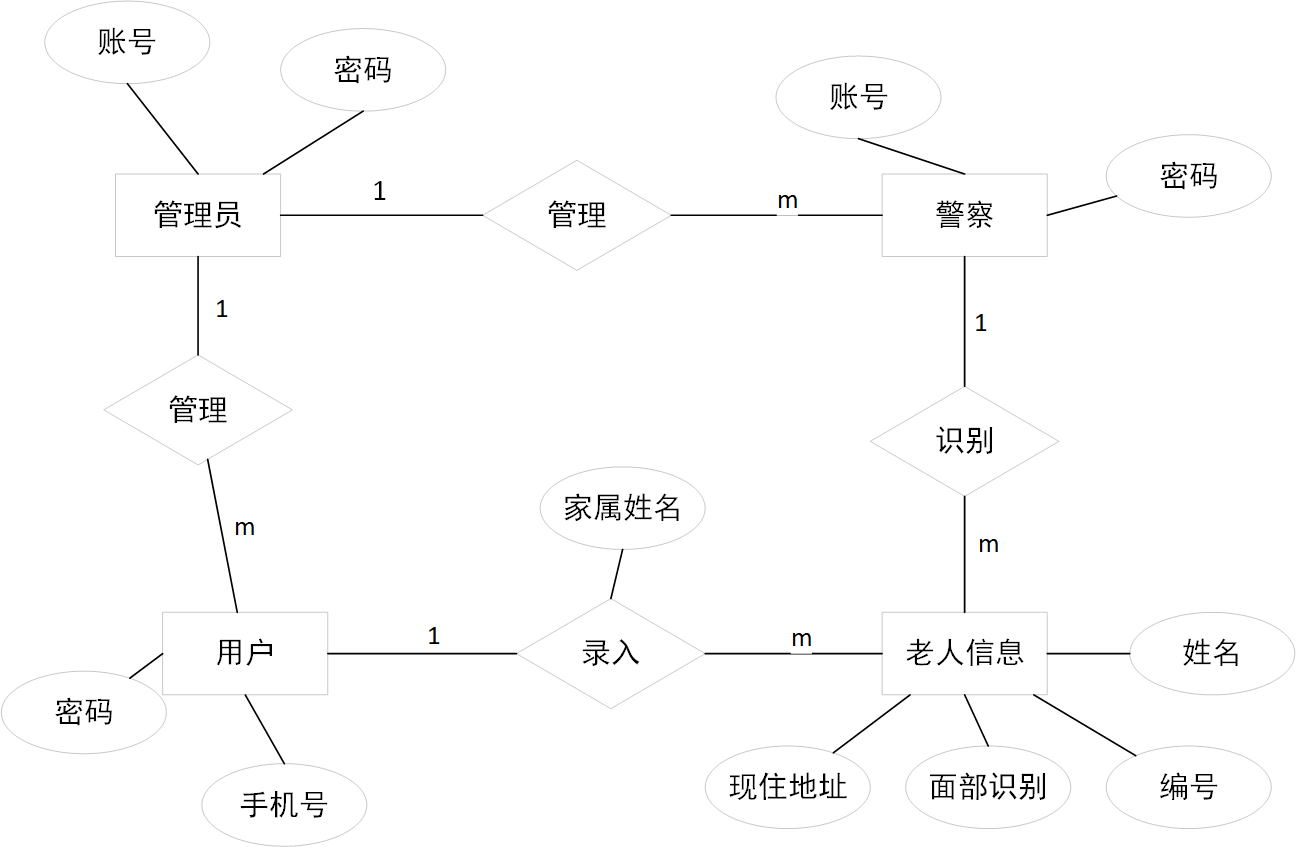


图4. E-R图

* + 1. 用户信息表User：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 含义 |
| Tel | Varchar | 11 | 电话号码（主键） |
| Password | Varchar | 24 | 密码 |

* + 1. 管理员信息表Admin：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 含义 |
| Account | Varchar | 20 | 账号（主键） |
| Password | Varchar | 24 | 密码 |

* + 1. 警察信息表Police：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 含义 |
| Account | Varchar | 20 | 账号（主键） |
| Password | Varchar | 24 | 密码 |

* + 1. 老年人信息表OM\_Info：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 含义 |
| ID | Varchar | 20 | 编号（主键） |
| Name | Varchar | 12 | 姓名 |
| Address | Varchar | 60 | 家庭地址 |
| Img\_Path | Varchar | 255 | 面部图像路径地址 |

* + 1. 家庭信息表FM\_Info：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 含义 |
| FM\_Tel | Varchar | 20 | 家属电话号码（主键） |
| OM\_ID | Varchar | 20 | 老人编号（主键） |
| FM\_Name | Varchar | 12 | 家属姓名 |

## 5.2. 物理结构设计要点

本系统中的数据均需要程序通过JDBC来进行访问，为了提高数据访问效率、提升用户使用体验，数据库使用的引擎与建立的索引结构需要调整，此外需要对一些敏感数据进行加密。

## 5.3. 数据结构与程序的关系

在进行用户的注册与登录时需要使用数据库的User表，在进行家属信息的录入时，需要使用数据库的OM\_Info表与FM\_Info表；在警察进行登录时需要使用数据库的Police表；在警察调用面部识别功能并获取老年人家属信息时需要使用数据库的OM\_Info表、FM\_Info表与User表；在管理员进行信息管理时需要使用数据库中的User表与Police表。

# 6.系统出错处理设计

## 6.1. 出错信息

|  |  |  |
| --- | --- | --- |
| 错误名称 | 系统输出信息 | 系统处理方法 |
| 用户无法登录系统 | 用户名或密码错误 | 停留在用户登录界面 |
| 用户未注册账号 | 该用户不存在 | 跳转至用户注册界面 |
| 警察遇到的老人的信息未被收录 | 未找到此人信息 | 输出走失老人问题解决的典型案例与方法 |
| 警察信息未被录入系统 | 该警察信息未被录入 | 通知管理员进行人员的确认 |
| 警察提交的老人照片不清晰 | 识别失败，用户信息未录入或图片清晰度低 | 提示警察重新进行图片提交 |

## 6.2. 补救措施

1. 定期对数据库进行备份，当数据库中的原始数据因为数据库崩溃等意外损坏或丢失，可以将最近的备份数据恢复至数据库，再根据日志文件将数据库恢复至数据出错前的状态；
2. 定期对数据库与相应的日志文件进行评估与检测，解决数据库使用期间所积累的问题，并对一些冗余数据进行清理。

## 6.3. 系统维护设计

对于数据库的维护，在本系统中可以通过数据库的备份与还原功能完成。而对于系统的维护，由于本系统采用模块化设计，系统的各个模块之间的独立性较强，便于以模块为单位进行系统的维护。

# 7.软件重用方案

## 7.1. 知识重用

在本软件的设计过程中使用的设计方法与设计思想均在相关的设计文档中有所体现，因此根据软件设计文档对相关的软件工程知识、软件设计方法知识与其他的相关知识均可以进行重用。

## 7.2. 方法和标准的重用

从方法上来说，在本软件中所使用的面向对象方法与模块化的设计方法与软件的设计模式与架构等均可进行重用；从标准上来说，本软件在设计中所采用的设计标准为计算机软件产品开发文件编制指南标准GB/T 8567-1988，这一软件文档设计标准可以在之后的软件开发之中进行重用，但是需注意，这一设计标准在2006年7月1日已被废弃，取而代之的是新的标准GB/T 8567-2006，在之后的软件开发中，应当更多选择的使用新的软件文档制定标准为宜。

## 7.3. 软件成分的重用

由于本软件的开发使用了面向对象方法与模块化的设计方法，各个类之间、各函数方法之间的界限较为明确，因此在软件中所使用的类与方法等软件成分均可以进行重用。

# 8.关键类的设计

在本软件中设计有以下关键类：

1. 普通用户类：定义用户的基本属性；
2. 警察类：定义警察的基本属性；
3. 老人信息类：定义与老年人相关的基本信息属性；
4. 管理员类：定义管理员的基本属性；
5. 用户服务类：定义与实现与用户相关的基本方法；
6. 警察服务类：定义与实现与警察相关的基本方法；
7. 管理员服务类：定义与实现与管理员相关的基本方法；
8. 用户控制类：控制用户的相关服务；
9. 警察控制类：控制警察的相关服务；
10. 管理员控制类：控制管理员的相关服务；