****

**系统概要设计文档**

组长：梁聪聪

组员：梅意婕

白伟婷

韩家乐

目录

[**1 引言（梁聪聪） 1**](#_Toc37709926)

[1.1 编写目的 1](#_Toc37709927)

[1.2 背景 1](#_Toc37709928)

[1.3 文档概述 1](#_Toc37709929)

[1.4设计原则与设计要求 2](#_Toc37709930)

[**2 总体设计（梁聪聪） 2**](#_Toc37709931)

[2.1 需求规定 2](#_Toc37709932)

[2.2 运行环境 3](#_Toc37709933)

[2.3 系统模块设计 3](#_Toc37709934)

[2.4 处理流程设计 4](#_Toc37709935)

[2.5 功能需求与程序的关系 7](#_Toc37709936)

[**3 接口设计（韩家乐） 7**](#_Toc37709937)

[3.1 用户接口 7](#_Toc37709938)

[3.2 外部接口 8](#_Toc37709939)

[3.3 内部接口 9](#_Toc37709940)

[**4 运行设计（白伟婷） 9**](#_Toc37709941)

[4.1 运行模块组合 9](#_Toc37709942)

[4.2 运行控制 10](#_Toc37709943)

[4.3 运行时间 10](#_Toc37709944)

[**5 系统数据结构设计（梅意婕） 10**](#_Toc37709945)

[5.1 逻辑结构设计要点 10](#_Toc37709946)

[5.2 物理结构设计要点 12](#_Toc37709947)

[5.3 数据结构与程序的关系 14](#_Toc37709948)

[**6 系统出错处理设计（梅意婕） 15**](#_Toc37709949)

[6.1 出错信息 15](#_Toc37709950)

[6.2 补救措施 16](#_Toc37709951)

[6.3 系统维护设计 17](#_Toc37709952)

## 1 引言（梁聪聪）

### 1.1 编写目的

需求分析，得出了系统的基本需求，要实现整个系统，需要对用户的需求进行设计，概要设计主要是利用比较抽象的语言对整个系统进行概括，确定对系统的物理配置，确定整个系统的处理流程和系统的数据结构，接口设计，实现对系统的初步设计。目的在于推动软件工程的规范化，使设计人员遵循统一的概要 设计书写规范，节省制作文档的时间，降低系统实现的风险，做到系统设计资料的规范性与全面性，以利于系统的实现、测试、维护、版本升级等。

预期的读者包括上级领导，相关开发人员以及管理人员。

### 1.2 背景

本系统的名称是AI宿舍安防系统，主要是人脸识别技术与图像文字获取技术相结合的应用。大学宿舍作为一个相对开放的环境，导致经常有非本校学生进入宿舍分发广告、上门推销等等。特别是那些刚刚进入大学的新生们，非常容易被上门推销的人员所迷惑，一时冲动上当报名了很多不必要的课程甚至被骗取钱财。而“AI宿舍安防系统”则是针对这些情况所开发的，针对外来人员进入学生宿舍的行为，进行自动化管理与审核。

系统主要实现人脸注册，登录，人脸识别进出宿舍，文字识别扫描证件自动生成电子信息表，记录查询和信息排查等功能。

### 1.3 文档概述

本文档为AI宿舍安防系统的概要设计，主要阐述了系统的整体结构和设计思想。

### 1.4设计原则与设计要求

本系统进行概要设计的原则:

命名规则;

* 模块独立性原则:
* 数据库设计规则;
* 系统易操作性要求;
* 系统可维护性要求;

## 2 总体设计（梁聪聪）

### 2.1 需求规定

表2.1 系统功能需求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 功能需求标识 | 优先级 | 简要描述 |
| 1 | 注册 | A | 高 | 普通用户分为学生、教职工，录入人脸，提交注册信息 |
| 2 | 登录 | B | 高 | 用户包括系统管理员，学生，教职工 |
| 3 | 人脸识别 | C | 高 | 物联网对到访人员的脸部信息进行匹配 |
| 4 | 扫描证件 | D | 高 | 物联网设备自动识别外来人员信息生成电子化表格 |
| 5 | 查询记录 | E | 较高 | 用户登录账户，查询信息 |
| 6 | 信息排查 | F | 较高 | 管理员进行信息排查，查看学生不正常进出数据 |

### 2.2 运行环境

表2.2 硬件环境

|  |  |  |
| --- | --- | --- |
| 设备 | 类型 | 功能权限 |
| 客户端设备 | 手机或个人电脑 | 均需要授予开启摄像头的权限和授予访问相册权限 |
| 物联网设备 | 物联网摄像头 | 拍照 |

**软件环境**：

1. 操作系统：安卓系统和Windows操作系统
2. 数据库：MySQL（或SQL Sever）
3. 开发工具：Eclipse、IDEA、VS Code
4. 开发语言：Java

### 2.3 系统模块设计

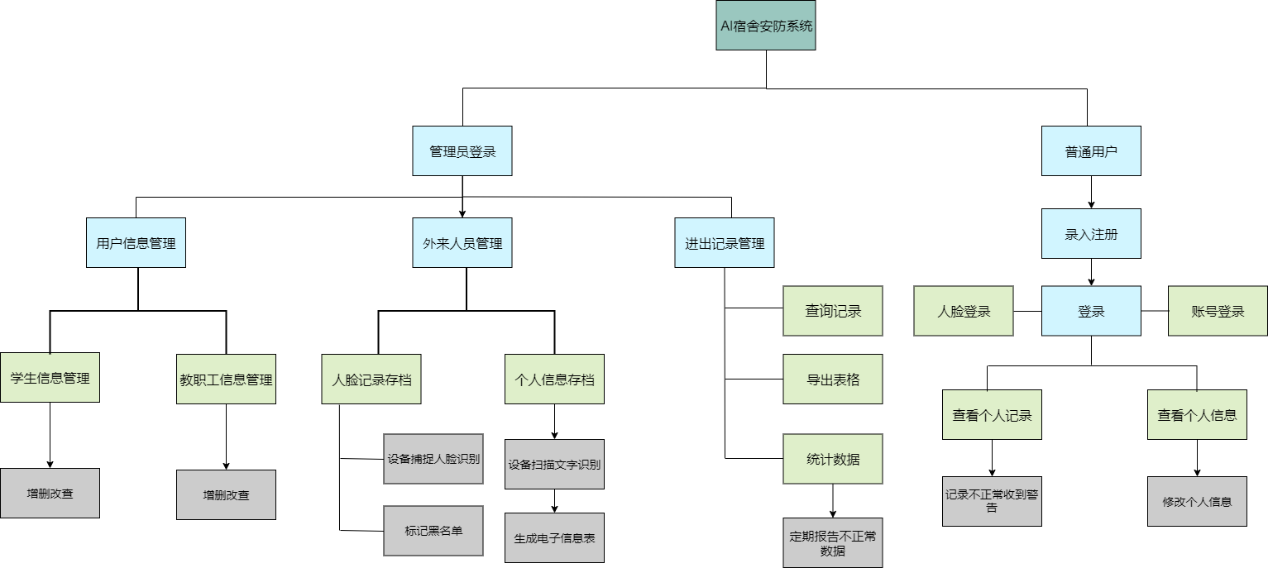


图2-1 系统模块设计

### 2.4 处理流程设计

#### 2.4.1 注册

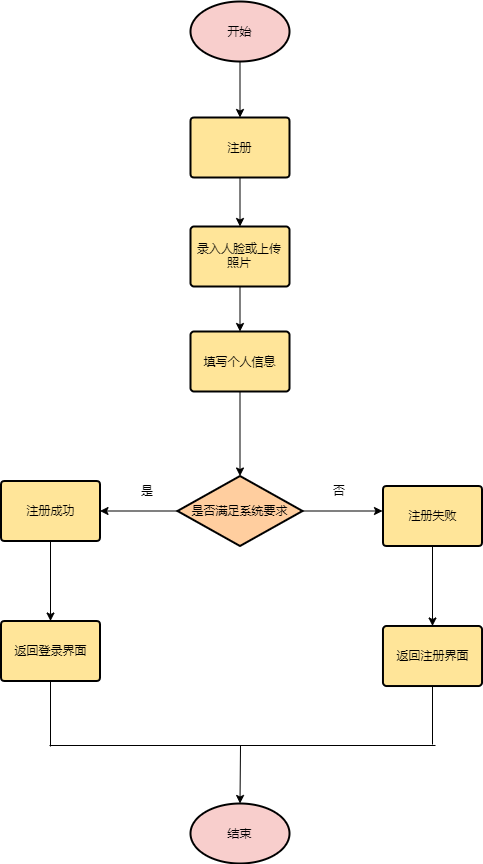


图2-2 注册流程设计

#### 2.4.2 登录

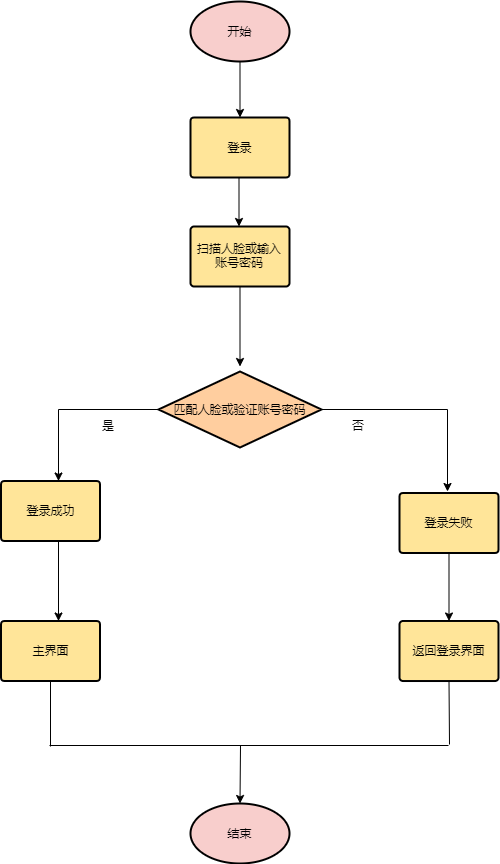


图2-3 登录流程设计

#### 2.4.3 管理员

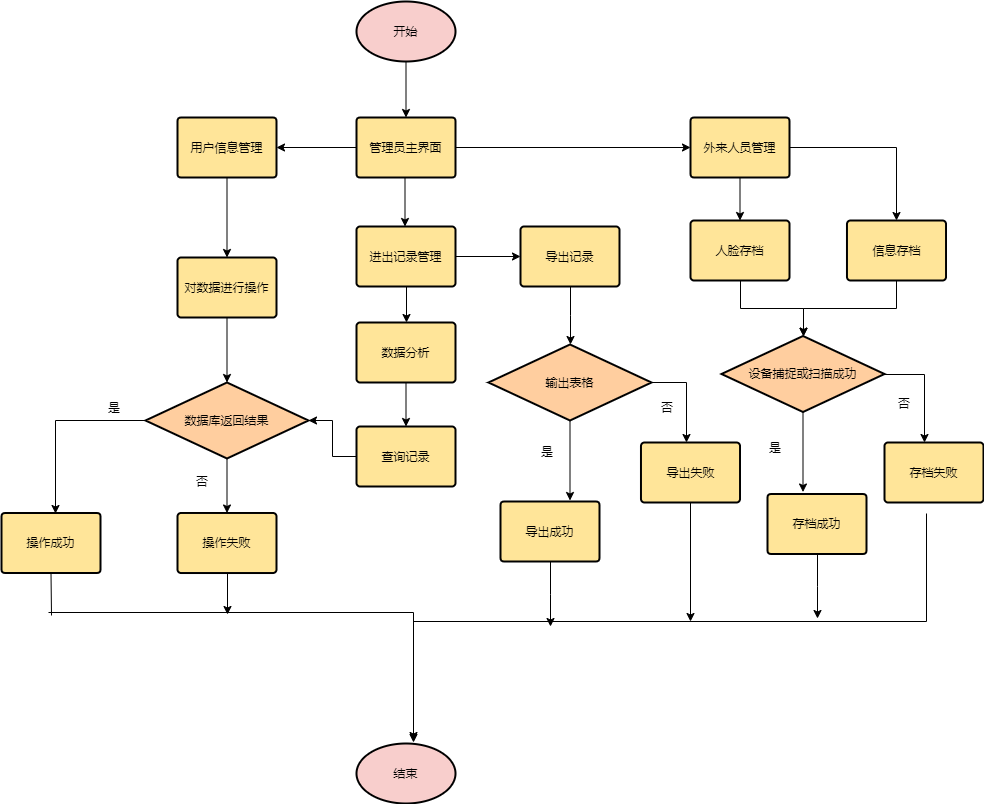


图2-4 管理员流程设计

#### 2.4.4 普通用户

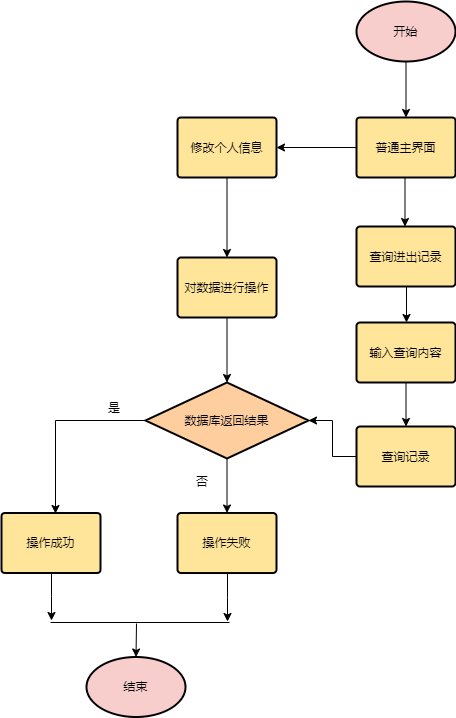


图2-4 普通用户流程设计

### 2.5 功能需求与程序的关系

表2.3 功能程序关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 注册程序 | 登录程序 | 人脸识别程序 | 文字识别程序 | 数据分析程序 |
| 注册 | √ |  |  |  |  |
| 登录 | √ | √ |  |  |  |
| 人脸识别 | √ | √ | √ |  | √ |
| 扫描证件 |  |  |  | √ | √ |
| 查询记录 | √ | √ | √ |  | √ |
| 信息排查 | √ | √ | √ |  | √ |

## 3 接口设计（韩家乐）

### 3.1 用户接口

#### 3.1.1 用户类别

* 提供普通用户（即学生或教职工）查阅的普通用户界面
* 提供管理员操作的管理员界面

#### 3.1.2 普通用户界面

1. 状态
   1. 注册命令：用户向服务器传递账号密码以及人脸信息，服务器连接数据库，匹配并修改用户信息。
   2. 登录命令：用户向服务器传递账号密码（或人脸信息），服务器返回匹配结果。
   3. 注销命令：用户向服务器传递注销命令，服务器返回响应。
2. 数据查询
   1. 查询个人记录：用户向服务器传递查询请求，服务器连接数据库，根据用户信息筛选数据并返回。
   2. 查看个人信息：用户向服务器传递查看请求，服务器返回用户信息。

#### 3.1.2 管理员用户界面

1. 状态
   1. 登录命令：用户向服务器传递账号密码（或人脸信息），服务器返回匹配结果。
   2. 注销命令：用户向服务器传递注销命令，服务器返回响应。
2. 数据输入
   1. 人脸识别存档：用户向服务器请求调用相关接口并获取数据，服务器链接数据库并添加所获数据，最后返回响应。
   2. 个人信息存档：用户向服务器请求调用相关接口并获取数据，服务器链接数据库并添加所获数据，最后返回响应。
3. 数据操作
   1. 查询记录：用户向服务器传递查询请求，服务器连接数据库，根据用户提供的条件筛选数据并返回。
   2. 用户个人信息的增删查改：用户向服务器传递相应请求，服务器连接数据库，根据相应请求对数据执行操作并返回响应。
4. 数据输出
   1. 导出记录：用户向服务器传递导出请求与导出数据，服务器根据所获数据制作文件并返回。
   2. 统计报告：用户向服务器传递统计请求（或系统定期传递请求）服务器根据已有数据进行相关筛选匹配，将结果制作成统计报告返回。

### 3.2 外部接口

#### 3.2.1 硬件接口

系统将在下表所列的硬件环境（参考）中运行：

* CPU：英特尔® 至强® E5405 四核处理器 2GHz
* 内存：4G（4\*1G） DDR2
* 网络摄像机：TL-IPC5220E-DC

#### 3.2.2 软件接口

系统需要的软件接口列表如下：

* Apache Tomcat服务器
* Chrome 浏览器
* 人脸识别API
* 图像识别API

### 3.3 内部接口

系统内部元素相互调用的接口如下：

* 普通用户查询个人记录时调用用户个人信息的接口
* 管理员用户人脸识别存档时调用选中用户个人信息的接口
* 管理员用户导出记录时调用页面筛选结果信息的接口

## 4 运行设计（白伟婷）

### 4.1 运行模块组合

当安卓端或者ios端进行数据输入时，物联网扫描设备接收信息，启动接收数据模块，将接收到的数据进行格式化，物联网设备接收到充分数据并且手机端输入数据充足时，将调用网络传输模块，将数据通过网络传送至后台接收器，此时服务器启动数据监测模块，将接收到的数据进行自动检测，若数据无误，则等待接收服务器返回的信息。接收到返回的信息后，随即调用数据输出模块，对信息进行可视化处理，反馈到电脑端控制界面。

当没有数据输入而对已储存的信息进行调用查看时，信息接收模块接收到相关指令，并将其格式化，信息传送模块将所接收到的指令向后台服务器进行传送，服务器根据指令调用信息储存模块提取数据，随即调用数据输出模块，对数据进行可视化处理，反馈到电脑端控制界面。

整个系统的运行在手机端支持ios系统及安卓系统，在电脑端支持Windows系统及MAC系统。

### 4.2 运行控制

运行控制将严格按照各模块间函数调用关系来实现。

在数据传输方面，当用户登录到系统时，用户输入的数据通过浏览器传输到服务器，服务器对数据进行检测，在对数据处理、访问数据库后，将返回信息送回浏览器，浏览器接收服务器返回的用户信息，给不同的用户展示不同的界面，用户在浏览器上对自己的个人信息进行操作，提交数据给服务器后，服务器校验数据，服务器返回提交结果给浏览器并更新可视化界面。

在数据查看方面，当管理员登录系统并且输入查询指令时，浏览器传输信号，服务器接收信号，服务器将所接收到的指令进行确认，随即对数据库进行访问及调用相关数据并对数据进行检测，检测无误后，服务器发送确认信号至浏览器，并将数据可视化反馈至管理员界面。

### 4.3 运行时间

在95％的情况下，各种运行模块组合一般时段响应时间皆不超过1.5秒，网络负载量大时不超过4秒；在推荐配置环境下每种运行模块组合占用各种资源时间最多不超过3秒。

## 5 系统数据结构设计（梅意婕）

### 5.1 逻辑结构设计要点

#### 5.1.1 用户信息表

**用户**（用户编号，用户账号，密码，账号绑定信息，学号）

p.s. 用户通过学号作为外键，匹配学生信息库中的学生详细信息。账号可以通过电话号码，微信，QQ等方式进行绑定。

#### 5.1.2 人脸信息表

**人脸信息**（用户编号，人脸细节信息）

p.s.通过用户编码匹配用户的人脸信息。

#### 5.1.3 管理员信息表

**管理员**（管理员编号，管理员账号，密码，工号）

p.s.管理员通过工号作为外键，匹配教职工信息库中的详细信息。

#### 5.1.4 进出宿舍记录表

**进出信息**（记录编号，进入时间，出去时间，进出楼栋号，用户编号）

p.s.此表格有两种形式，分别是管理员查询的进出宿舍记录表以及用户自己的信息记录表。上面表格逻辑结构为管理员，下面表格逻辑结构为用户。

**个人进出信息**（用户编号，记录编号，进入时间，出去时间，进出楼栋号）

#### 5.1.5 外来人员进出记录表

**外来人员进出信息**（记录编号，进入时间，出去时间，进出楼栋号，外来人员编号）

p.s.外来人员通过扫描身份相关证件进出宿舍，同时通过外来人员编号匹配详细信息，同时会获取被保存的外来人员信息与状态。

#### 5.1.6 外来人员信息表

**外来人员信息**（外来人员编码，姓名，性别，电话，住址，状态）

p.s.对于状态标识不对的外来人员（黑名单状态）给予警告。同时我们是默认外来人员不算做用户的。

#### 5.1.7 其他表格

**学生详细信息**（学号，姓名，性别，家庭住址，电话，邮箱，就读时间，专业信息，班级信息，楼栋号）

p.s.学生（用户）的详细信息通过连接学校学生信息库获取。

**教职工详细信息**（工号，姓名，性别，家庭住址，电话，邮箱，年龄，学院信息，楼栋号）

p.s.教职工（管理员）信息同样通过链接学校的教职工数据信息库获取。

**楼栋信息**（楼栋号，楼栋名，宿舍数量，所在校区）

p.s楼栋信息同样通过链接学校的楼栋数据信息库获取。

### 5.2 物理结构设计要点

表5.1 用户信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | userId | 用户编号 | int | 9 |  |  | √ | 自动递增 |
| 2 | userAccount | 用户账号 | varchar | 32 |  |  |  |  |
| 3 | password | 密码 | varchar | 20 |  |  |  |  |
| 4 | information | 账号绑定信息 | varchar | - |  | √ |  |  |
| 5 | studentId | 学号 | int | 7 |  |  |  |  |

表5.2 人脸信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | userId | 用户编号 | int | 9 |  |  | √ | 匹配用户信息表用户编号 |
| 2 | data | 人脸细节信息 | - | - |  |  |  |  |

表5.3 管理员信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | managerId | 管理员编号 | int | 9 |  |  | √ | 自主递增 |
| 2 | managerAccount | 管理员账号 | varchar | 32 |  |  |  |  |
| 3 | password | 密码 | varchar | 20 |  |  |  |  |
| 4 | managerNumber | 工号 | int | 7 |  |  |  |  |

表5.4 进出宿舍记录表A（进出信息表）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | documentId | 记录编号 | int | 9 |  |  | √ | 自主递增 |
| 2 | inTime | 进入时间 | Date | - |  |  |  |  |
| 3 | outTime | 出去时间 | Date |  |  |  |  |  |
| 4 | dormitoryId | 楼栋号 | varchar | 10 |  |  |  | 外键 |
| 5 | userId | 用户编号 | int | 9 |  |  |  | 匹配用户信息表中的用户编号 |

表5.5 进出宿舍记录表B（个人进出信息表）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | userId | 用户编号 | int | 9 |  |  | √ | 匹配用户信息表中的用户编号 |
| 2 | documentId | 记录编号 | int | 9 |  |  |  | 匹配进出信息表中的记录编号 |
| 3 | inTime | 进入时间 | Date | - |  |  |  |  |
| 4 | outTime | 出去时间 | Date |  |  |  |  |  |
| 5 | dormitoryId | 楼栋号 | varchar | 10 |  |  |  | 外键 |

表5.6 外来人员进出记录表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | notesId | 记录编号 | id | 9 |  |  | √ | 自主增加 |
| 2 | inTime | 进入时间 | Date | - |  |  |  |  |
| 3 | outTime | 出去时间 | Date |  |  |  |  |  |
| 4 | dormitoryId | 楼栋号 | varchar | 10 |  |  |  | 外键 |
| 4 | otherId | 外来人员编号 | int | 9 |  |  |  |  |

表5.7 外来人员信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | otherId | 外来人员编号 | int | 9 |  |  | √ | 匹配外来人员进出记录表中的外来人员编号 |
| 2 | name | 姓名 | varchar | 15 |  |  |  |  |
| 3 | sex | 性别 | int | 1 | 1 |  |  |  |
| 4 | phone | 电话 | int | 11 |  |  |  |  |
| 5 | address | 地址 | varchar | 50 |  |  |  |  |
| 6 | status | 状态 | int | 1 | 0 |  | √ | 默认不被标记 |

表5.8 学生信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | studentId | 学号 | int | 9 |  |  | √ |  |
| 2 | name | 姓名 | varchar | 15 |  |  |  |  |
| 3 | sex | 性别 | int | 1 | 1 |  |  |  |
| 4 | phone | 电话 | int | 11 |  |  |  |  |
| 5 | address | 地址 | varchar | 50 |  |  |  |  |
| 6 | subject | 专业 | varchar | 15 |  |  |  |  |
| 7 | email | 邮箱 | varchar | 50 |  |  |  |  |
| 8 | class | 班级 | varchar | 10 |  |  |  |  |
| 9 | dormitoryId | 楼栋号 | varchar | 10 |  |  |  | 外键 |
| 10 | startYear | 入学时间 | Date |  |  |  |  |  |

表5.9 教职工信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | studentId | 学号 | int | 9 |  |  | √ |  |
| 2 | name | 姓名 | varchar | 15 |  |  |  |  |
| 3 | sex | 性别 | int | 1 | 1 |  |  |  |
| 4 | phone | 电话 | int | 11 |  |  |  |  |
| 5 | address | 地址 | varchar | 50 |  |  |  |  |
| 6 | academy | 学院 | varchar | 15 |  |  |  |  |
| 7 | email | 邮箱 | varchar | 50 |  |  |  |  |
| 8 | age | 年龄 | varchar | 10 |  |  |  |  |
| 9 | dormitoryId | 楼栋号 | varchar | 10 |  |  |  | 外键 |

表5.9 楼栋信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 默认值 | 允许为空 | 主键 | 其他说明 |
| 1 | dormitoryId | 楼栋号 | varchar | 10 |  |  | √ |  |
| 2 | dormitoryName | 楼栋名 | varchar | 15 |  |  |  |  |
| 3 | number | 宿舍数量 | int |  |  |  |  |  |
| 4 | campus | 所在校区 | varchar | 10 |  |  |  |  |

### 5.3 数据结构与程序的关系

数据结构为关系型数据库，可以使用标准化的SQL语句与数据结构进行交互。

## 6 系统出错处理设计（梅意婕）

### 6.1 出错信息

用一览表的方式说明每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。

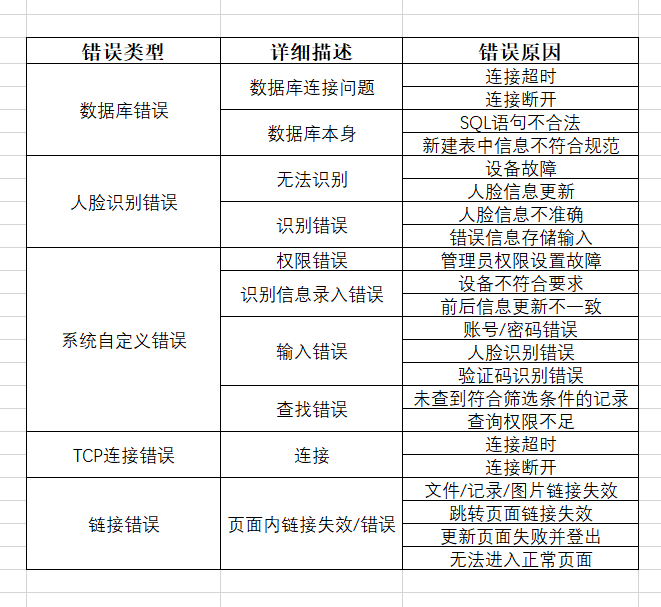


图6-1 一览表

### 6.2 补救措施

为了降低故障出现的概率，可以采取以下措施：

1. 使用备份技术，定期备份数据库信息，当原始数据丢失后，启用备份的副本的数据，尽量降低影响程度。
2. 外用设备（摄像头、拦截器等）的定期维护，周期性检查维护实体设备。保障设备的正常运行。

假如出现了故障情况，可以采用以下变通措施：

1. 对于软错误：在任何的修改、添加、删除操作中，加入确认机制，需要进行二次确认，重要的记录需要至少两人以上（管理员）的确认。同时分析错误的类型，并给出相应的错误提示机制，以及相关解决手段或联系方式。
2. 对于硬错误：硬错误的方向可寻，可以在可能出现错误的地方进行测试，或者重置系统到最新更新之前，重新输入指令。

### 6.3 系统维护设计

1. **代码维护**：后续测试，对代码进行完善（在各个模块升级时进行一定程度的增加、删除、修改等操作）。模块化代码，随着系统应用范围的改变（增大or减小），模块化管理维护，使得维护操作更加清晰、简便。
2. **数据维护**：定期更新数据、备份数据，周期性对数据结构进行的优化和更新。
3. **其他方面**：程序的每次更新、修改需要备份之前的版本，并对新版本的更新加入新的注释，对程序运行期间产生的错误进行收集、分析，避免相关故障的重复出现，对错误提示的优化（提示信息详细化）等。