

分类号\_\_\_\_\_

密级\_\_\_\_\_

UDC \_\_\_\_\_

编号\_\_\_\_\_



湖北第二师范学院

# 本科毕业论文(设计)

题目\_\_\_\_\_基于钉钉的校园宿舍门禁系统\_\_\_\_\_

学 院 \_\_\_\_\_ 计算机学院 \_\_\_\_\_

专 业 名 称 \_\_\_\_\_ 物联网工程 \_\_\_\_\_

年 级 \_\_\_\_\_ 2015 级 \_\_\_\_\_

学 生 姓 名 \_\_\_\_\_ 李园庭 \_\_\_\_\_

学 号 \_\_\_\_\_ 1550360033 \_\_\_\_\_

指 导 教 师 \_\_\_\_\_ 孙延维 \_\_\_\_\_

二〇一八 年 四 月

## 湖北第二师范学院本科毕业论文（设计）原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

论文作者签名：

年 月 日

**摘要：**当前大部分校园宿舍管理制度仍以传统人工管理为主，不仅考勤等日常基本事务需要耗费大量的人力，其安全性与便捷性都有很大的局限性。本系统立足于校园宿舍管理现状，旨在利用所学的专业知识，希望在现有校园管理制度模式下结合移动互联网，进一步提升针对校园宿舍门禁这一方面的移动数字化建设。其中参考样本为本校宿舍管理现状，在对学校制度以及平台稳定性、发展性进行综合考量后，选择以钉钉软件平台作为开发基础，进行钉钉二次开发以满足功能需求。系统与钉钉平台进行数据对接，以钉钉企业内部应用的方式开放，采用更加智能化的方式有效改善当前学校宿舍管理存在的诸多不足之处。

**关键词：**门禁签到 宿舍 钉钉 二次开发

**Abstract:** Most of the current campus dormitory management system is still dominated by traditional manual management, which not only requires a lot of manpower for daily basic affairs such as attendance checking, but also has great limitations in security and convenience. This system is based on the current situation of campus dormitory management and aims to make use of the professional knowledge learned. It is hoped that under the current campus management system mode and combined with the mobile Internet, the mobile digital construction of campus dormitory access control can be further improved. The reference sample is the current situation of dormitory management of the university. After comprehensive consideration of the school system, platform stability and development, the nail-nailed software platform is selected as the development basis to conduct nail-nailed secondary development to meet functional requirements. The data connection between the system and the nailing platform is carried out, which is open in the form of internal application of the nailing enterprise. In addition, a more intelligent way is adopted to effectively improve the many deficiencies existing in the current management of school dormitories.

**Keywords:** Entrance guard sign in   dormitory   DingTalk   Secondary development

# 目 录

|  |    |
|--|----|
| 第 1 章 引言.....                            | 1  |
| 1.1 选题背景及意义.....                         | 1  |
| 1.2 国内外研究状况.....                         | 2  |
| 1.3 拟研究的主要内容和思路.....                     | 2  |
| 第 2 章 相关技术.....                          | 4  |
| 2.1 前端 (axml + acss + JavaScript) .....  | 4  |
| 2.2 后端 (NodeJS + Koa2 + Sequelize) ..... | 4  |
| 2.3 数据库 (MySQL + Navicat) .....          | 5  |
| 第 3 章 需求分析.....                          | 6  |
| 3.1 功能需求分析.....                          | 6  |
| 3.1.1 登录验证功能.....                        | 6  |
| 3.1.2 一键查寝功能.....                        | 7  |
| 3.1.3 来访登记功能.....                        | 7  |
| 3.1.4 详情查看功能.....                        | 7  |
| 3.2 可行性分析.....                           | 8  |
| 3.2.1 技术可行性分析.....                       | 8  |
| 3.2.2 应用可行性分析.....                       | 8  |
| 第 4 章 概要设计.....                          | 9  |
| 4.1 系统架构概要设计.....                        | 9  |
| 4.1.1 整体流程设计.....                        | 9  |
| 4.1.2 技术栈.....                           | 10 |
| 4.2 数据库结构设计.....                         | 11 |
| 4.3 系统功能设计.....                          | 13 |
| 4.3.1 登录验证功能.....                        | 13 |
| 4.3.2 一键查寝功能.....                        | 13 |

|                             |           |
|-----------------------------|-----------|
| 4.3.3 来访登记功能.....           | 13        |
| 4.3.4 详情查看功能.....           | 13        |
| 4.4 系统界面设计.....             | 14        |
| 4.4.1 应用首页.....             | 14        |
| 4.4.2 一键查寝页面.....           | 14        |
| 4.4.3 来访登记页面.....           | 14        |
| 4.4.4 详情信息页面.....           | 14        |
| <b>第 5 章 详细设计及编码.....</b>   | <b>15</b> |
| 5.1 前期准备.....               | 15        |
| 5.1.1 钉钉中与本系统业务相关的字段分析..... | 15        |
| 5.1.2 数据准备.....             | 16        |
| 5.2 前端页面实现.....             | 16        |
| 5.2.1 全局配置.....             | 16        |
| 5.2.2 首页.....               | 17        |
| 5.2.3 一键查寝页面.....           | 17        |
| 5.2.4 来访登记页面.....           | 18        |
| 5.2.5 用户详情页面.....           | 18        |
| 5.3 后端功能实现.....             | 19        |
| 5.3.1 实现后端 url 路由功能.....    | 19        |
| 5.3.2 应用内免登功能.....          | 19        |
| 5.3.3 扫码签到.....             | 20        |
| 5.3.4 查询用户基本信息.....         | 21        |
| <b>第 6 章 系统测试.....</b>      | <b>23</b> |
| <b>第 7 章 总结与展望.....</b>     | <b>25</b> |
| 7.1 总结.....                 | 25        |
| 7.2 展望.....                 | 25        |
| <b>参考文献.....</b>            | <b>26</b> |
| <b>致 谢.....</b>             | <b>27</b> |

## 第 1 章 引言

随着当今办公移动化的普及,越来越多的公司内部选择采用更为智能的软件化管理。而在教育部“教育信息化 2.0”的号召下,校园的教育数字化转型势在必行,智能化管理能够在一定程度上加速未来校园的建设,为校园管理人员及学生的工作和学习生活带来极大便利。传统的校园宿舍门禁多采用接触式 IC 卡,通过学生刷学生卡、访客刷身份证或通过手机短信的形式与被访教师取得联系,得到确认后方可进入的传统模式<sup>[1]</sup>。这种方式不仅需要携带卡片且易磨损丢失,在移动设备飞速发展的当下,非接触式门禁系统还有相当大的发展空间。而人工管理方式需要人工日常确认学生到寝状况,耗费大量人力。有不足就有发展空间,当前的宿舍管理模式显然可以进一步改进。

### 1.1 选题背景及意义

本选题立足于校园宿舍管理现状,旨在利用所学的专业知识,希望在现有校园管理制度模式下结合移动互联网,进一步提升针对校园宿舍门禁这一方面的移动数字化建设。其中参考样本为本校宿舍管理现状,在对学校制度以及平台稳定性、发展性进行综合考量后,选择以钉钉软件平台作为开发基础,进行钉钉二次开发以满足功能需求。

以本校为参考样本,当前校园宿舍管理现状及弊端简述如下:

1. 每日晚考勤繁琐且低效。在校期间每日晚考勤一般由班内同学到各寝室查寝后向辅导员报备,对于大学生管理工作人员来讲,部分机械的重复性工作挤占了大量工作时间,降低了工作效率<sup>[13]</sup>。
2. 手工签到统计不准确。每逢节假日,离校及返校需要手工签到,管理员无法一一提醒并督促,无法准确统计留校及返校人数,手工签到失去意义。
3. 无关人员出入宿舍控制不够严格。在大量人员出入或管理员疏忽情况下无关人员相对容易混入宿舍,对学生的人身财产安全造成一定威胁。

钉钉作为阿里旗下一款免费的移动办公化软件，有着相对健全的企业人员管理结构。根据 Quest mobile 第三方数据显示，钉钉在智能移动办公领域市场份额占比最高，成为企业智能移动办公首选。而当前学院也已引入了钉钉，基于钉钉上的组织架构及功能进行二次开发宿舍的门禁系统，将查寝签到功能并入具有一定的现实可行性，且能够实现使用钉钉一个管理软件集成打通校园组织架构管理和学生宿舍管理，便于实现统一管理。

## 1.2 国内外研究状况

当前门禁系统按照门禁系统的设计原理大致可分为两类：一是独立一体机：控制器自带读卡器，无需卡片密码，便宜易装，安全性差。二是控制器与读卡器分离：室内控制，需要有效卡片或密码，安全性较高，使用较普遍；按照进出识别方式分类有：密码识别、卡片识别、生物识别。

而从识别设备来看，非接触式 IC 卡是主流和发展方向，目前已经占据了约 9 成以上的市场。随着门禁系统的普及以及门禁应用领域的不断扩展，基于不同行业的应用需求，出现了与短息平台整合、与办公自动化整合、与 ERP 整合、与 IT 整合等多种新兴的门禁系统实例<sup>[4]</sup>。目前系统整合趋势有两大方向。一种整合了背景运算系统。用户能够使用应用多种功能，且整合后的系统能提供适当的信息和数据；另一种整合形式是在一张卡里同时整合多重技术，从而达到给予用户多种使用功能的目的，其系统独立于背景运算系统，用户需要分别注册每种应用。

未来，我国门禁系统市场发展将呈网络化、生物识别、集成化发展，提高门禁系统精度以及数据融合、信息集成方面将成为研究重点<sup>[2]</sup>。目前基于钉钉平台开发的校园宿舍门禁系统还尚未发展起来。

## 1.3 拟研究的主要内容和思路

内容概括：进行钉钉二次开发实现非接触式移动端扫码出入校园宿舍门禁并支持管理员应用内实时查寝功能。以钉钉平台为大背景，拟达成的功能：出入宿舍楼需要通过钉钉登录进行身份验证，通过后可出入门禁；学生通过门禁时宿舍



楼状态标志反转，从而支持实现管理员实时查寝功能；外来人员出入宿舍需通过管理员进行电子报备。

主要思路：利用钉钉平台，进行钉钉企业内部应用的二次开发，将企业组织架构按照宿舍楼及寝室进行划分，搭建与之匹配的后台服务器及数据库，用户通过钉钉软件实现相关操作。所有企业相关人员信息通过钉钉提供的相关 API 获取并导入数据库。

## 第 2 章 相关技术

本系统采用钉钉二次开发，需要前后台及数据库支持，前后台需要对接相应的钉钉开发 API，数据库需要对接钉钉用户数据。

### 2.1 前端 (axml + acss + JavaScript)

前端以钉钉 E 应用 IDE 为开发工具，使用标准化的基础框架以及组件进行业务开发。其中的 axml 和 acss 是框架中包装好的标签和样式，是对传统的 html 及 css 的扩充及修改。

### 2.2 后端 (NodeJS + Koa2 + Sequelize)

后端采用 VSCode 为开发工具，使用 NodeJS 进行后端开发。其中 NodeJS 是服务器端的 JavaScript 运行环境，具有无阻塞(non-blocking)和事件驱动(event-driven)等特色，被设计用来解决服务端阻塞问题，在 Node 上运行的 JavaScript 相比其他后端开发语言最大的优势是借助 JavaScript 天生的事件驱动机制加 V8 高性能引擎，使编写高性能 Web 服务轻而易举，可以通过它来搭建基于 JavaScript 的 Web App。

Koa 是基于 Node.js 平台的下一代 web 开发框架，依赖 node v7.6.0 或 ES2015 及更高版本和 async 方法支持。通过利用 async 函数，丢弃回调函数，并有力地增强错误处理。无任何中间件捆绑，而是提供了一套优雅的方法，帮助开发者快速编写服务端应用程序。

Sequelize 是一款基于 Nodejs 功能强大的异步 ORM (Object-Relational Mapping) 框架，能够把关系数据库的表结构映射到对象上，同时支持 PostgreSQL, MySQL, SQLite and MSSQL 多种数据库。

### 2.3 数据库 (MySQL + Navicat)

数据库采用 MySQL，使用 Navicat 作为可视化管理工具。其中 MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面是最好的 RDBMS (Relational Database Management System: 关系数据库管理系统) 应用软件之一。Navicat 是一套快速、可靠并且性价比较高的数据库管理工具。

## 第3章 需求分析

### 3.1 功能需求分析

本系统针对校园宿舍门禁，默认预设目标用户人群为宿舍管理员及住宿学生。

#### 3.1.1 登录验证功能

针对宿舍楼人员进出混乱的问题，该功能对所有用户开放。

用户在进出宿舍时需要验证身份，验证是本栋宿舍的相关人员(管理员、学生)后方可通过门禁。

用户通过身份验证后方可通过门禁，结合钉钉的接口后总结出几种方式如下：

- (1) 密码登录
- (2) 钉钉内免登
- (3) 钉钉扫码/刷脸登录

考虑到本项目打算基于钉钉开发企业内部应用，且需要满足方便快捷的需求，对几种方式的具体分析如下：

(1) 密码登录不仅需要更多操作，费时且体验更差，系统也需要额外存储用户的账号密码，这样系统设计时与独立开发一套系统相比更为复杂，从一定程度上来说失去了使用钉钉平台二次开发的意义，暂时排除。

(2) 钉钉内免登指企业员工在钉钉内使用该企业应用时，只要直接点击应用，便可免输入账户密码实现自动登录。这种方式需要加上定位功能以确定用户是否是在期望的范围内登陆系统，所以验证时需要用户打开定位功能，较为复杂，暂时保留。

(3) 钉钉扫码/刷脸登录则是希望用户以钉钉的账号登录时，可以通过钉钉扫码方式实现免密登录。这种方式与钉钉平台的粘合度较高，初步设想为在门禁处设置二维码即可，不需要额外做用户定位，且钉钉用户的验证交由钉钉提供

的 API 即可。须注意的是这种方式依然需要数据库保留用户的相关数据，且需要分析钉钉用户数据的配置方式。刷脸登录的设想是基于钉钉推出的刷脸签到功能，问题在于钉钉未开放人脸识别的 API，需要购买 API 的调用次数。

总结：优先考虑实现钉钉扫码登录方式验证。

### 3.1.2 一键查寝功能

针对查寝繁琐且耗时耗力的问题，该功能对管理员开放。

管理员可通过应用实时一键确认当前本宿舍楼学生到寝状况，省去人工查寝统计的繁琐步骤。

管理员在应用首页内点击“一键查寝”按钮，跳转至一键查寝页面，显示当前寝室楼栋、当前时间、未到寝人数及名单。

### 3.1.3 来访登记功能

针对外来人员需要进出宿舍楼的情况，该功能对管理员开放。

外来人员进出宿舍需由管理员批准，并由管理员上传详细信息至系统。到访记录存至后台数据库以便后期调档查看。

管理员在首页点击来访登记导航，跳转至来访登记页面，显示要填写的表单项。管理员填写后提交，前端通过 post 方式将信息提交至后台，后台写入数据库 Sign\_record 表中。

### 3.1.4 详情查看功能

针对管理员需要获知指定学生详细信息的情况，该功能对管理员开放。

管理员针对指定学生可查看其详细信息（如学生未到寝需获取其电话号码等其他详细信息时）。

管理员在一键查寝页面内点击显示的学生记录将跳转至详情页面，显示该生详细信息（如联系电话等）。

## 3.2 可行性分析

### 3.2.1 技术可行性分析

钉钉作为阿里旗下一款免费的移动办公化软件，有着相对健全的企业人员管理结构。经过近几年的急速发展，其规范性、扩展性都经过了市场检验，有配套的开发者平台及咨询服务。钉钉开放了丰富的服务端接口能力，开发者可以借助这些接口能力，实现企业系统与钉钉的集成打通。经查阅其开发文档，钉钉提供的开放接口能够满足本系统所需功能。故本系统的开发实现在技术上具有可行性。

### 3.2.2 应用可行性分析

本系统是基于钉钉的二次开发，故企业应当首先使用钉钉组织本企业组织架构。钉钉在智能移动办公领域市场份额占比最高，活跃用户数排名第一，成为企业智能移动办公首选。将系统作为钉钉企业内部应用并入便于学校进行统一管理，具有一定的现实可行性。

## 第4章 概要设计

### 4.1 系统架构概要设计

#### 4.1.1 整体流程设计

用户角色：管理员、学生

用户进出门禁流程，如图 4-1 所示：

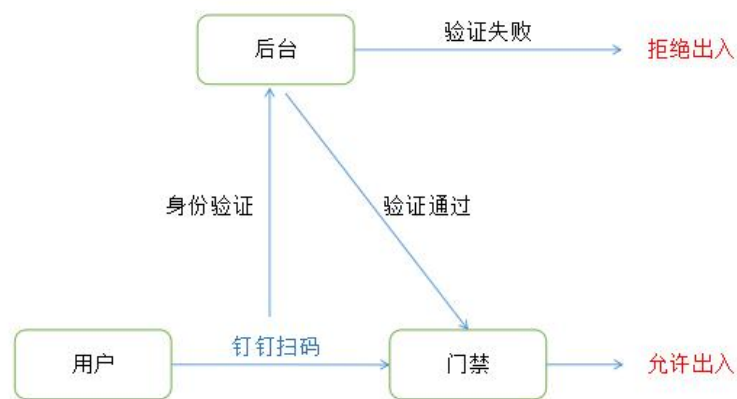


图 4-1 用户进出门禁流程图

学生进出门禁后台登记示意图，如图 4-2 所示：

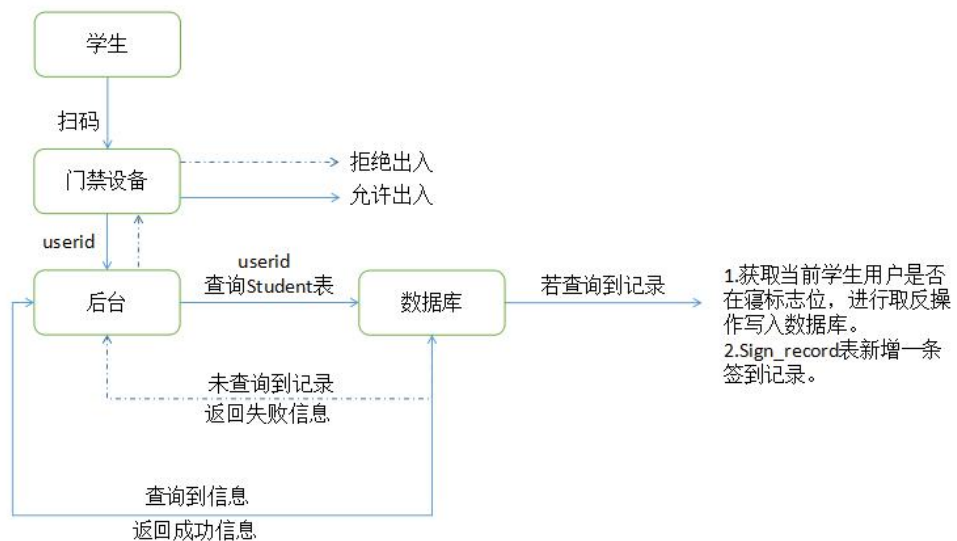


图 4-2 学生进出门禁后台登记示意图

管理员应用内登陆流程示意图，如图 4-3 所示：

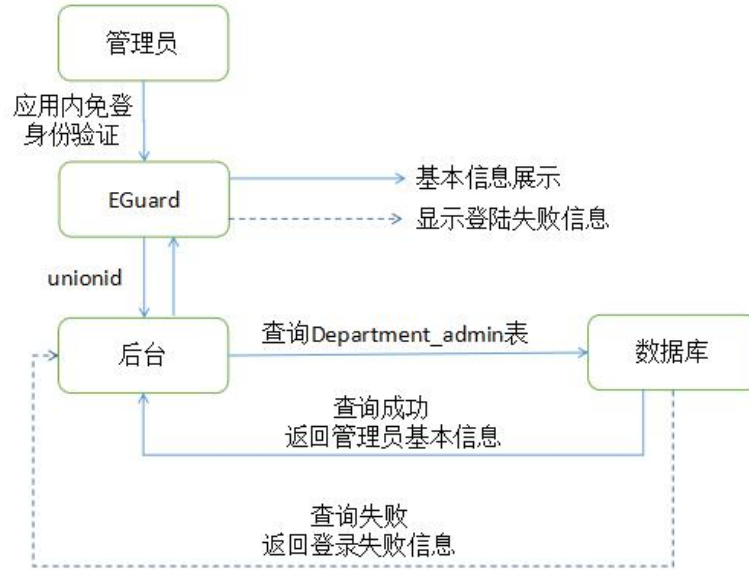


图 4-3 管理员应用内登陆流程示意图

管理员一键查寝流程示意图，如图 4-4 所示：

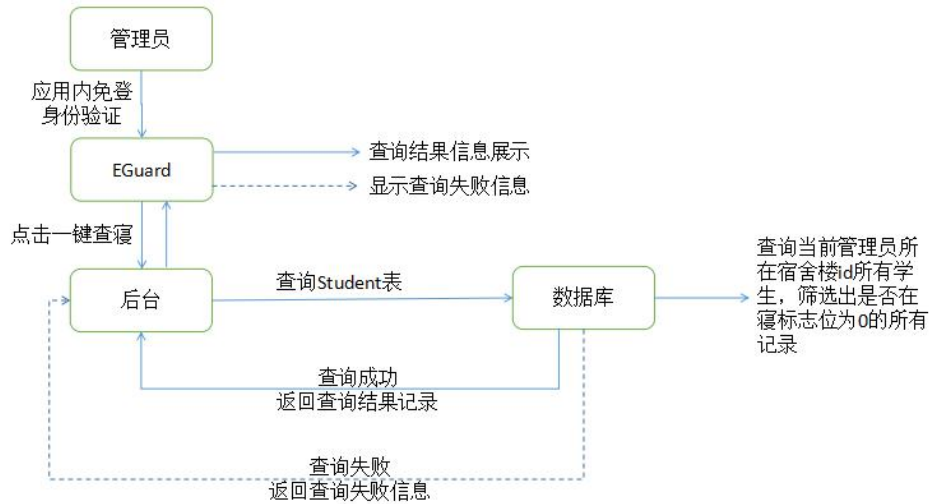


图 4-4 管理员一键查寝流程示意图

#### 4.1.2 技术栈

前端：钉钉 E 应用开发者 IDE (axml + acss + js)

后端：NodeJS (koa2 + Sequelize) + Visual Studio Code

数据库：MySQL8.0.14 + Navicat Premium 12



## 4.2 数据库结构设计

数据库表设计，如图 4-5 所示：

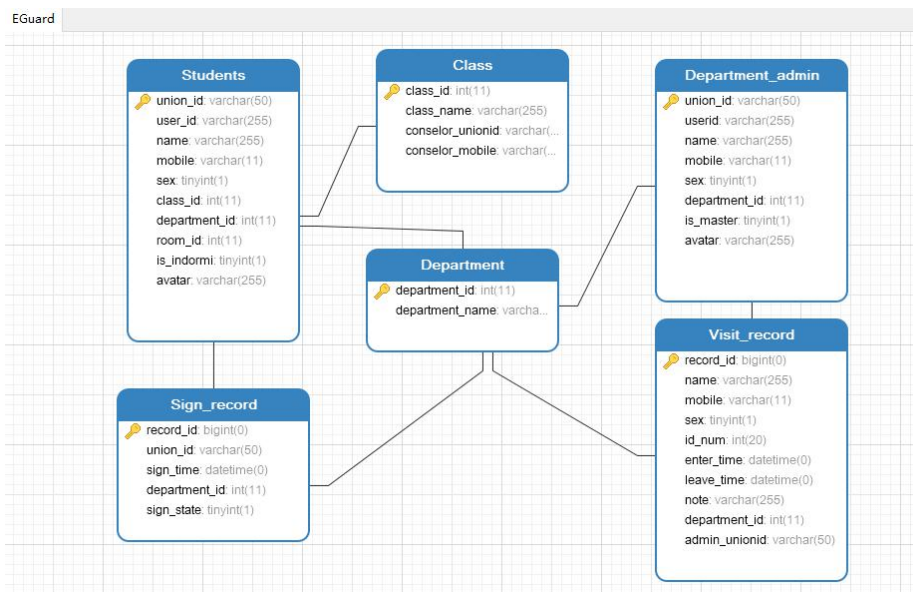


图 4-5 数据库表设计

Students 表设计，用于存储所有学生的基本信息，如图 4-6 所示：


|               |  |                               |    |      |    |         |        |     |                                     |  |                     |
|---------------|--|-------------------------------|----|------|----|---------|--------|-----|-------------------------------------|--|---------------------|
| 对象            |  | students @eguard (mysql) -... |    |      |    |         |        |     |                                     |  |                     |
| 保存            |  | 添加字段                          |    | 插入字段 |    | 删除字段    |        | 主键  |                                     | ↑ 上移 ↓ 下移  |                     |
| 字段            |  | 索引                            | 外键 | 触发器  | 选项 | 注释      | SQL 预览 |     |                                     |  |                     |
| 名             |  |                               |    |      |    | 类型      | 长度     | 小数点 | 不是 null                             | 键  | 注释                  |
| union_id      |  |                               |    |      |    | varchar | 50     | 0   | <input checked="" type="checkbox"/> |  1 | 钉钉用户唯一识别id          |
| user_id       |  |                               |    |      |    | varchar | 255    | 0   | <input type="checkbox"/>            |  | 钉钉userid/学号(企业可自定义) |
| name          |  |                               |    |      |    | varchar | 255    | 0   | <input type="checkbox"/>            |  | 学生姓名                |
| mobile        |  |                               |    |      |    | varchar | 11     | 0   | <input type="checkbox"/>            |  | 联系电话                |
| sex           |  |                               |    |      |    | tinyint | 1      | 0   | <input type="checkbox"/>            |  | 性别                  |
| class_id      |  |                               |    |      |    | int     | 11     | 0   | <input type="checkbox"/>            |  | 班级id                |
| department_id |  |                               |    |      |    | int     | 11     | 0   | <input type="checkbox"/>            |  | 部门id/宿舍楼id          |
| room_id       |  |                               |    |      |    | int     | 11     | 0   | <input type="checkbox"/>            |  | 寝室门牌号               |
| is_indormi    |  |                               |    |      |    | tinyint | 1      | 0   | <input type="checkbox"/>            |  | 是否在寝(0:不在寝, 1:在寝)   |
| avatar        |  |                               |    |      |    | varchar | 255    | 0   | <input type="checkbox"/>            |  | 用户头像url             |

图 4-6 Students 表设计

Class 表设计，用于存储各个班级基本信息，如图 4-7 所示：

对象

class @eguard (mysql) - 表

保存

添加字段

插入字段

删除字段

主键

上移

下移

字段

索引

外键

触发器

选项

注释

SQL 预览

| 名                | 类型      | 长度  | 小数点 | 不是 null                             | 键 | 注释         |
|------------------|---------|-----|-----|-------------------------------------|---|------------|
| class_id         | int     | 11  | 0   | <input checked="" type="checkbox"/> | 1 | 班级id       |
| class_name       | varchar | 255 | 0   | <input type="checkbox"/>            |   | 班级名称       |
| conselor_unionid | varchar | 50  | 0   | <input type="checkbox"/>            |   | 辅导员unionid |
| conselor_mobile  | varchar | 11  | 0   | <input type="checkbox"/>            |   | 辅导员联系电话    |
| conselor_name    | varchar | 50  | 0   | <input type="checkbox"/>            |   | 辅导员姓名      |

图 4-7 Class 表设计

Department 表设计，用于存储所有宿舍楼基本信息，如图 4-8 所示：

| 对象              | department @eguard (mys... |      |      |                                     |    |              |  |
|-----------------|----------------------------|------|------|-------------------------------------|----|--------------|--|
| 保存              | 添加字段                       | 插入字段 | 删除字段 | 主键                                  | 上移 | 下移           |  |
| 字段              | 索引                         | 外键   | 触发器  | 选项                                  | 注释 | SQL 预览       |  |
| 名               | 类型                         | 长度   | 小数点  | 不是 null                             | 键  | 注释           |  |
| department_id   | int                        | 11   | 0    | <input checked="" type="checkbox"/> | 1  | 钉钉部门id/宿舍楼id |  |
| department_name | varchar                    | 255  | 0    | <input type="checkbox"/>            |    | 钉钉部门名称/宿舍楼名称 |  |

图 4-8 Department 表设计

Department\_admin 表设计，用于存储所有宿舍管理员信息，如图 4-9 所示：

| 对象            | department_admin @eguar... |      |      |                                     |    |                          |  |
|---------------|----------------------------|------|------|-------------------------------------|----|--------------------------|--|
| 保存            | 添加字段                       | 插入字段 | 删除字段 | 主键                                  | 上移 | 下移                       |  |
| 字段            | 索引                         | 外键   | 触发器  | 选项                                  | 注释 | SQL 预览                   |  |
| 名             | 类型                         | 长度   | 小数点  | 不是 null                             | 键  | 注释                       |  |
| union_id      | varchar                    | 50   | 0    | <input checked="" type="checkbox"/> | 1  | 钉钉唯一识别id                 |  |
| userid        | varchar                    | 255  | 0    | <input type="checkbox"/>            |    | 钉钉userid/员工工号(企业可自定义)    |  |
| name          | varchar                    | 255  | 0    | <input type="checkbox"/>            |    | 管理员姓名                    |  |
| mobile        | varchar                    | 11   | 0    | <input type="checkbox"/>            |    | 管理员电话                    |  |
| sex           | tinyint                    | 1    | 0    | <input type="checkbox"/>            |    | 管理员性别(0: 男, 1: 女)        |  |
| department_id | int                        | 11   | 0    | <input type="checkbox"/>            |    | 部门id/宿舍楼id               |  |
| is_master     | tinyint                    | 1    | 0    | <input type="checkbox"/>            |    | 是否为主管理员(0:副管理员, 1:主管管理员) |  |
| avatar        | varchar                    | 255  | 0    | <input type="checkbox"/>            |    | 用户头像url                  |  |

图 4-9 Department\_admin

Sign\_record 表设计，用于存储学生进出门禁记录，如图 4-10 所示：

| 对象            | sign_record @eguard (mys... |      |      |                                     |    |                  |  |
|---------------|-----------------------------|------|------|-------------------------------------|----|------------------|--|
| 保存            | 添加字段                        | 插入字段 | 删除字段 | 主键                                  | 上移 | 下移               |  |
| 字段            | 索引                          | 外键   | 触发器  | 选项                                  | 注释 | SQL 预览           |  |
| 名             | 类型                          | 长度   | 小数点  | 不是 null                             | 键  | 注释               |  |
| record_id     | bigint                      | 20   | 0    | <input checked="" type="checkbox"/> | 1  | 记录id             |  |
| union_id      | varchar                     | 50   | 0    | <input type="checkbox"/>            |    | 钉钉唯一识别id         |  |
| sign_time     | datetime                    | 0    | 0    | <input type="checkbox"/>            |    | 签到时间时间戳          |  |
| department_id | int                         | 11   | 0    | <input type="checkbox"/>            |    | 部门id/宿舍楼id       |  |
| sign_state    | tinyint                     | 1    | 0    | <input type="checkbox"/>            |    | 签到状态(0:出寝, 1:入寝) |  |

图 4-10 Sign\_record 表设计

Visit\_record 表设计，用于存储来访登记记录，如图 4-11 所示：

| 对象            | visit_record @eguard (mysq... |      |      |                                     |    |            |  |
|---------------|-------------------------------|------|------|-------------------------------------|----|------------|--|
| 保存            | 添加字段                          | 插入字段 | 删除字段 | 主键                                  | 上移 | 下移         |  |
| 字段            | 索引                            | 外键   | 触发器  | 选项                                  | 注释 | SQL 预览     |  |
| 名             | 类型                            | 长度   | 小数点  | 不是 null                             | 键  | 注释         |  |
| record_id     | bigint                        | 20   | 0    | <input checked="" type="checkbox"/> | 1  | 记录id       |  |
| name          | varchar                       | 255  | 0    | <input type="checkbox"/>            |    | 访客姓名       |  |
| mobile        | varchar                       | 11   | 0    | <input type="checkbox"/>            |    | 联系电话       |  |
| sex           | tinyint                       | 1    | 0    | <input type="checkbox"/>            |    | 访客性别       |  |
| id_num        | int                           | 20   | 0    | <input type="checkbox"/>            |    | 来访人员身份证号   |  |
| enter_time    | datetime                      | 0    | 0    | <input type="checkbox"/>            |    | 来访时间时间戳    |  |
| leave_time    | datetime                      | 0    | 0    | <input type="checkbox"/>            |    | 来访结束时间时间戳  |  |
| note          | varchar                       | 255  | 0    | <input type="checkbox"/>            |    | 备注         |  |
| department_id | int                           | 11   | 0    | <input type="checkbox"/>            |    | 部门id/宿舍楼id |  |
| admin_unionid | varchar                       | 50   | 0    | <input type="checkbox"/>            |    | 管理员unionid |  |

图 4-11 Visit\_record 表设计

## 4.3 系统功能设计

### 4.3.1 登录验证功能

用户在进出宿舍时通过钉钉扫码，系统通过用户扫码获取 userid 进而查询数据库 student 表判断是否允许通过并返回，并同时对其 is\_indor(是否在宿舍楼标志位)取反以标注其当前到寝状态。

### 4.3.2 一键查寝功能

管理员发起查寝请求，应用前端向后端发送当前管理员所在宿舍楼 id（亦即部门 id），后台向数据库查询宿舍楼 id 匹配记录下所有学生的 is\_indor 标志位并筛选出该标志位为 0 的学生，将结果以数组发送给前端显示。

### 4.3.3 来访登记功能

外来人员进出宿舍得到管理员批准后由管理员在应用内相应页面填写登记表，前端页面提交表单数据至后台，写入数据库 Sign\_record 表便于后期调档查看。

### 4.3.4 详情查看功能

管理员发起查看详情的请求（例如点击未到人员名单中记录项时），前端将人员 id 发送至后台，后台在数据库查询对应用户详细信息返回给前端展示。

## 4.4 系统界面设计

### 4.4.1 应用首页

- (1) 轮播图展示图片
- (2) 个人资料卡，显示当前登录管理员基本信息（钉钉头像、系统编号、姓名、管理宿舍名称）
- (3) 一键查寝按钮
- (4) 通用底部 Tabbar 导航栏，分为首页、来访登记和报修登记

### 4.4.2 一键查寝页面

- (1) 当前查寝时间展示、当前查寝宿舍楼名称
- (2) 当前未到寝总人数
- (3) 未到寝人员列表（学号、姓名、寝室号、手机号），超过 n 条分页展示
- (4) 返回首页按钮

### 4.4.3 来访登记页面

- (1) 访客登记表单（姓名、性别、联系电话、身份证号、备注、确认提交按钮、重置表单按钮）
- (2) 通用底部 Tabbar 导航栏，分为首页、来访登记和报修登记

### 4.4.4 详情信息页面

- (1) 用户钉钉头像及基本信息（姓名、性别、寝室楼名称、寝室号）
- (2) 用户详细信息（学号、班级、联系电话、辅导员、辅导员电话、上次离寝时间）
- (3) 返回首页按钮

## 第 5 章 详细设计及编码

### 5.1 前期准备

#### 5.1.1 钉钉中与本系统业务相关的字段分析

(1) 钉钉用户相关字段:

|            |   |
|------------|---|
| union_id   | 员工在当前开发者企业账号范围内的唯一标识, 系统生成, 固定值, 不会改变, 选择其作为 EGuard 系统用户表主键 |
| userid     | 员工在当前企业内的唯一标识, 也称 staffId。可由企业在创建时指定, 并代表一定含义比如工号, 创建后不可修改  |
| name       | 员工名字  |
| mobile     | 手机号码  |
| isAdmin    | 是否为企业的管理员, true 表示是, false 表示不是                             |
| department | 成员所属部门 id 列表  |
| avatar     | 头像 url  |

(2) 钉钉部门相关字段:

|                       |  |
|-----------------------|--|
| department            | 部门列表数据, 以部门的 order 字段从小到大排列                              |
| id                    | 部门 id  |
| name                  | 部门名称   |
| parentid              | 父部门 id, 根部门为 1   |
| deptManagerUserIdList | 部门的主管列表, 取值为由主管的 userid 组成的字符串, 不同的 userid 使用 “ ” 符号进行分割 |
| sourceIdentifier      | 部门标识字段, 开发者可用该字段来唯一标识一个部门, 并与钉钉外部通讯录里的部门做映射              |

### 5.1.2 数据准备

由于系统基于钉钉二次开发，部分数据库用户基础数据需要通过钉钉相关 API 爬取本系统使用企业当前组织架构相关人员的信息。

数据爬取导入流程，如图 5-1 所示：

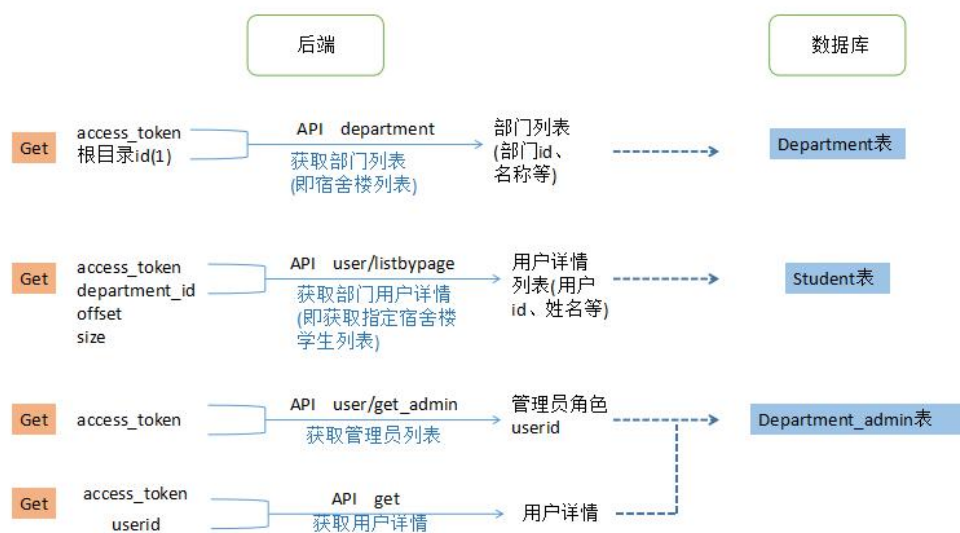


图 5-1 数据爬取导入流程

数据导入之前需要条件——学校使用钉钉并按照宿舍分配组织架构；

EGuard 使用之前需要导入数据才能正常使用，爬取数据并导入数据库作为单独部分执行。

## 5.2 前端页面实现

### 5.2.1 全局配置

App 代表顶层应用，管理所有页面和全局数据，以及提供生命周期方法。它也是一个构造方法，生成 App 实例。

全局配置文件如下：

app.js：全局应用逻辑

app.acss：全局应用样式

app.json: 全局应用配置, 决定页面文件的路径、窗口表现、设置网络超时时间、设置多 tab 等。

(1) 前端页面路由配置: pages 属性是一个数组, 每一项都是字符串, 用来指定 E 应用的页面。每一项代表对应页面的路径信息, 数组的第一项代表 E 应用的首页。E 应用中新增/减少页面, 都需要对 pages 数组进行修改。

(2) 默认通用页面头部配置: window 属性用于设置 E 应用通用的状态栏、导航条、标题、窗口背景色。

(3) 通用 Tabbar 底部导航栏, 即多 tab 应用 (客户端窗口的底部栏可以切换页面), 通过 tabBar 配置项指定 tab 栏的表现, 以及 tab 切换时显示的对应页面。

### 5.2.2 首页

(1) 首页轮播图部分: 利用滑块视图容器构造首页轮播效果, 多张图片利用对象数组传参。

(2) 管理员信息卡片部分: 对多个 view 嵌套布局, 构造各单元信息展示位置及样式。

(3) 查寝签到按钮: 设置 onTap 点击事件, 触发事件函数跳转至一键查寝页面。设置背景图片, 采用 fixed 水平居中定位。

### 5.2.3 一键查寝页面

跳转至页面时携带当前管理员所在宿舍楼 id 信息, 后台查询当前该宿舍楼所有学生是否在寝标志位, 筛选出所有值为 0 的记录并返回至前端。

(1) 前端页面展示当前查寝时间以及宿舍楼名称信息, 展示未到寝总人数信息以及具体未到寝人员名单, 结果以表单形式分页展示, 一次展示 n 条数据并按照学号进行排序: 由于这里不需要即时刷新数据, 所以在页面 onLoad 事件内一次性获取后台返回的结果信息及总人数并缓存, 默认当前页码为 1。

(2) 页面底部按钮分别对应相应的点击事件，对展示区数据即时切换，包括转至前一页数据、后一页数据、首尾页数据以及第 n 页数据。

(3) 下拉页面刷新当前时间及展示数据：在 Page 中通过自定义 onPullDownRefresh 函数，监听该页面用户的下拉刷新事件。在页面对应的 .json 配置文件中配置 "pullRefresh": true 选项，开启下拉刷新事件获取当前时间点服务器返回的数据信息。当处理完数据刷新后，调用 dd.stopPullDownRefresh 停止当前页面的下拉刷新。

(4) 本页面无通用 Tabbar 底部导航栏，用户通过点击左上角返回按钮返回至应用首页。

#### 5.2.4 来访登记页面

表单页，记录管理员输入的登记信息。

- (1) 表单项：访客姓名、访客性别、访客联系方式、访客身份证号码、来访时间、备注。
- (2) 底部设表单提交按钮，绑定 submitForm 提交函数。点击后表单提交向服务器发起 POST 请求，页面弹出提示框显示提交结果，确认后跳转回应用首页。
- (3) 底部设重置表单按钮，点击后表单页所填内容清空。
- (4) 本页面无左上角返回按钮，用户直接通过通用 Tabbar 底部导航栏进行页面切换跳转，且切换后原登记表单页面数据清空。

#### 5.2.5 用户详情页面

页面加载时携带所选择用户（学生）的信息（union\_id）向后端发起 GET 请求，接收后端返回结果后进行数据展示。

(1) 展示数据包括：用户头像、用户姓名、用户性别、用户学号、用户联系方式、用户寝室号、用户所在班级、用户所在班级辅导员及联系方式等。



(2) 本页面无通用 Tabbar 底部导航栏, 用户通过点击左上角返回按钮可返回至上一个页面。

## 5.3 后端功能实现

### 5.3.1 实现后端 url 路由功能

将 URL 处理函数集中到某几个 js 文件中, 让 app.js 自动导入所有处理 URL 的函数, 从而达到代码分离, 逻辑清晰的目的。

(1) 编写 .js 文件处理相应的 url 处理函数, 通过模块思想, 以如下代码形式把文件中的 URL 处理函数暴露出来。

```
module.exports = {  
    'GET /': fn_index,  
    'POST /signin': fn_signin  
};
```

(2) 修改 app.js 文件, 实现让其自动扫描 controllers 目录, 找到所有 js 文件, 导入, 然后注册每个 URL。

(3) 导入 fs 模块, 然后用 readdirSync 列出文件, 用 files.filter 过滤出所有的 .js 文件, 然后遍历处理每个 js 文件。最后把扫描 controllers 目录和创建 router 的代码从 app.js 中提取出来, 作为一个简单的 middleware 使用, 命名为 controller.js, 并在 app.js 中使用该 middleware。这样所有处理 URL 的函数按功能组存放在 controllers 目录, 今后只需要不断往这个目录下加东西就可以了, app.js 保持不变。

### 5.3.2 应用内免登功能

用户(管理员)通过移动设备登录钉钉后打开 EGuard 应用, 此过程用户不需要再次输入账号及密码登录, 应用免登并自动获取当前钉钉用户身份进行后台身份校验, 即 EGuard 系统账户密码与钉钉账户密码绑定。

免登实现流程，如图 5-2 所示：

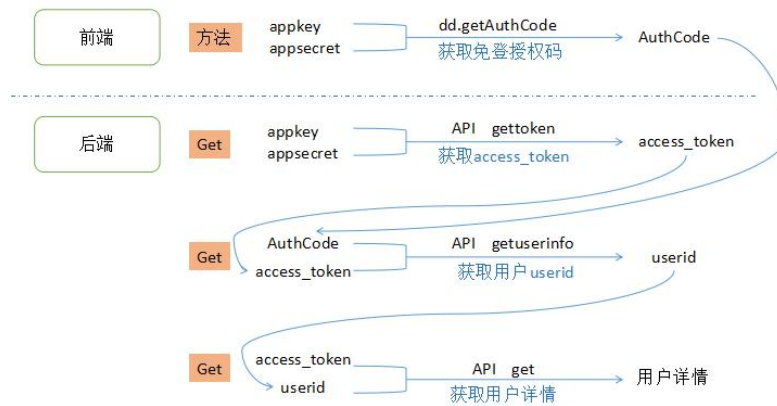


图 5-2 免登实现流程

使用 request 模块的 request 函数发起 GET 请求，通用请求函数如下：

```

request(url,function(error,response,body){
    if(!error && response.statusCode === 200){
        .....
    }
    else{
        .....
    }
}

```

### 5.3.3 扫码签到

使用钉钉客户端扫码并确认登录系统，在系统内获得正在访问用户的钉钉身份，而用户无需输入账户密码，将钉钉登录二维码内嵌到页面中。

(1) 在页面中引入指定 js 文件：

```
<script src="//g.alicdn.com/dingding/dinglogin/0.0.5/ddLogin.js"></script>
```

(2) 实例相关 JS 对象：

```
var url = encodeURIComponent('http://localhost.me/index.php?test=1&aa=2');
```

```

var goto = encodeURIComponent
('https://oapi.dingtalk.com/connect/oauth2/sns_authorize?appid=appid&response_type=code&scope=snsapi_login&state=STATE&redirect_uri='+url)
var obj = DDLogin({
  id:"login_container",//这里需要在页面定义一个 HTML 标签并设置 id
  goto: "",
  style: "border:none;background-color:#FFFFFF;",
  width : "365",
  height: "400"
});

```

#### 5.3.4 查询用户基本信息

使用 ORM 框架 Sequelize<sup>[11]</sup>操作数据库，并使用 ES7 的 `await` 来调用一个 Promise 对象。

首先创建一个 `sequelize` 对象实例并定义模型，告诉 `Sequelize` 如何映射数据库表（以下代码片段仅以其中一个表为例，其余省略）：

```

// 创建一个 sequelize 对象实例
var sequelize =
new Sequelize(db_config.database, db_config.username, db_config.password, {
  host: db_config.host,
  dialect: 'mysql',
  pool: {
    max: 5,
    min: 0,
    idle: 30000
  }
});

```

```
// 定义模型，告诉 Sequelize 如何映射数据库表
var Class = sequelize.define('class', {
  class_id: {      type: Sequelize.INTEGER(11),
                   primaryKey: true              },
  class_name: Sequelize.STRING(255),
  conselor_unionid: Sequelize.STRING(50),
  conselor_mobile: Sequelize.INTEGER(11),
  conselor_name: Sequelize.STRING(50)
}, {
  timestamps: false,
  freezeTableName: true
});
... ..
```

koa2<sup>[10]</sup> 体积很小，但扩展性很强，koa2 中需要理解 async/await（异步/等待）<sup>[15]</sup>，koa2 完全使用 promise 并结合 async 来实现异步，可以免除重复出现并且极其繁琐的回调函数的嵌套，因此开发异步 Web 程序变得简单许多<sup>[14]</sup>。

由于 Node.js 是单线程的，所以需要根据条件异步查询数据库后返回查询结果<sup>[12]</sup>，其中异步使用 ES7 关键字 async 和 await。async 声明该函数为一个异步函数，await 则表示等待右侧表达式执行完返回结果<sup>[9]</sup>：

```
(async () => {
  var admin = await Department_admin.findOne({
    where: {
      union_id: '1eZ4iPca85HInlgxLdyV5NwiEiE'
    }
  });
})();
```

## 第 6 章 系统测试

应用首页，管理员从钉钉内进入页面，如图 6-1 所示：

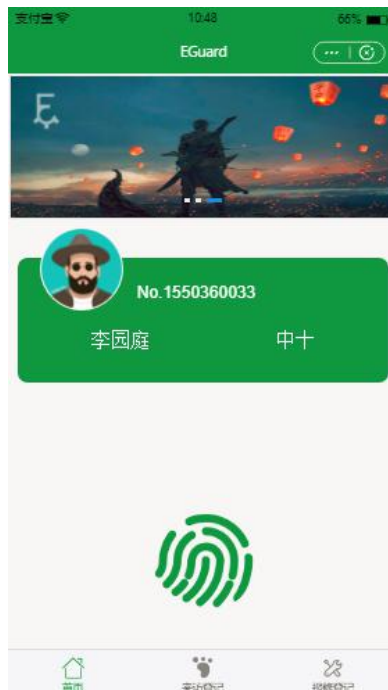


图 6-1 应用首页

一键查寝页面，首页中点击一键查寝按钮跳转至本页面，如图 6-2 所示：



图 6-2 一键查寝页面

来访登记页面，首页点击 Tabbar 导航栏来访登记跳转至本页面，如图 6-3 所示：



EGuard | 来访人员登记

访客姓名 李园庭

访客性别 ☐ 男 ☒ 女

联系电话 15172496781

访客身份证号 4210\*\*\*\*\*0026

备注：  
请在此输入来访事项等备注信息...  
0/140

确认提交

重置表单

图 6-3 来访登记页面

详情信息页面，管理员点击未到寝人员名单单项跳转至本页面，如图 6-4 所示：



EGuard | 用户详情

返回

李园庭  
西一 612

|        |                 |
|--------|-----------------|
| 学号     | 1550360033      |
| 班级     | 15级物联网工程1班      |
| 联系电话   | 15172496781     |
| 辅导员    | 熊军平             |
| 辅导员电话  | 15172496781     |
| 上次离寝时间 | 2019年3月2日 08:00 |

图 6-4 详情信息页面

## 第 7 章 总结与展望

### 7.1 总结

在技术日新月异的今天，技术应当成为人类前进的有力助手，而这也是大力发展技术的意义所在。一些传统低效的方式方法都将在可预见的未来逐渐被替代，发展更加高效、智能的人类社会，以进一步解放人类的双手。钉钉本身作为一款优秀的现象级企业管理平台本身提供了多样的功能，EGuard 在钉钉的基础上做二次开发是针对校园宿舍管理这一个特定方向做出的定制化延伸和发展，用同一套体系对各方面进行管理有利于企业管理中心化。与传统宿舍门禁管理系统需要单独部署相比，采用钉钉二次开发的 EGuard 扩展性相对更好。而受限于开发时间，目前的 EGuard 的功能十分有限，很多细节只做了简单处理。由于前后端采用的均为较新的技术，网上关于钉钉二次开发的问题也相对较少，在学习使用的过程中也有一些不甚理解之处，有的地方处理的方法可能不够理想。

综上所述，EGuard 目前的功能和细节都还不甚完善，还有待改进。在这个过程中我也学到了很多知识，提高了自己独立查阅资料、解决问题的能力。

### 7.2 展望

从应用功能的角度来看，EGuard 的初衷是希望改善宿舍门禁的现状，需要更加方便、灵活。目前的功能只是实现一些比较基础的功能，还有很多细节功能没有完善。立足于应用的使用情境，还有很多功能可以扩展，功能更加完善才能更好地服务于用户，使用户使用更加方便快捷应当是需要不断追求的目标。比如身份验证，在未来希望能增加刷脸验证方式，这样安全性更加有保障，验证过程也更加简便快捷。

从应用使用的角度来看，EGuard 还有很多需要进行优化的地方。页面加载时速度还有待提高。未来还要不断学习、不断优化代码部分。

总之，EGuard 的开发还远没有结束，未来还要不断为它添砖加瓦让它变得逐渐完善、更加方便易用。

## 参考文献

- [1] 金松.掌上校园门禁系统设计与实现[J].信息通信,2018(05):92-94.
- [2] 朱利龙. 基于非接触识别技术的校园门禁管理系统设计与实现[D].青岛理工大学,2014.
- [3] 刘廷龙.基于 Android 平台智能门禁管理系统设计与实现[J].信息通信,2016(01):282-284.
- [4] 2016 年中国门禁系统现状调研及市场前景走势分析报告[EB/OL].[http://www.cir.cn/R\\_JiXieDianZi/58/MenJinXiTongHangYeXianZhuangYuFaZhanQianJing.html](http://www.cir.cn/R_JiXieDianZi/58/MenJinXiTongHangYeXianZhuangYuFaZhanQianJing.html),2019 - 03 - 16.
- [5] 刘旭. 南开中学住宿楼门禁管理系统的设计与实现[D].天津大学,2016.
- [6] 钉钉开发文档[EB/OL].<https://open-doc.dingtalk.com>,2019 - 03 - 16.
- [7] Nicholas C.Zakas.Professional JavaScript for Web Developers,3Rrd Edition[M].北京:人民邮电出版社,2012.
- [8] David Flanagan,JavaScript:The Definitive Guide,Six Edition[M].北京:机械工业出版社,2012.
- [9] 廖雪峰.Javascript 教程[EB/OL].<https://www.liaoxuefeng.com>,2019 - 03 - 16.
- [10] koa2[EB/OL].<https://koa.bootcss.com>,2019 - 03 - 16.
- [11] Sequelize[EB/OL].<http://docs.sequelizejs.com>,2019 - 03 - 16.
- [12] 许会元,何利力.NodeJS 的异步非阻塞 I/O 研究[J].工业控制计算机,2015,28(03):127-129.
- [13]唐静.互联网时代大学生管理信息化研究——以钉钉软件为例[J].内蒙古科技与经济,2018(23):61-62.
- [14]王金龙,宋斌,丁锐.Node.js:一种新的 Web 应用构建技术[J].现代电子技术,2015,38(06):70-73.
- [15]杨明戊,戴经国,陈正铭.基于 Node.js+koa2 的全景图上传与展示技术研究[J].电脑编程技巧与维护,2018(10):149-152.



## 致 谢

毕业设计临近尾声，也标志着我的大学生涯即将结束。在这四年时光里，我收获了许多知识、友情和经验，许多老师和同学、朋友都给予了我莫大的帮助和鼓励，期间途径我人生旅途的人和事，都将成为我弥足珍贵的回忆。

在 EGuard 从构思设想到逐渐实现并完善的过程中碰到了不少困难，期间查看了很多的文档，也问了无数次度娘。有时候也会因为一个问题迟迟无法解决会感到沮丧，但当问题解决时，会感到非常开心，这种开心是玩乐所体会不到的开心。伴随着系统的逐步完善，我学到了很多新知识，学会了如何将平时课堂上学到的知识转化到具体应用情境中去，提高了查阅文献及资料并解决具体问题的能力。在这段时间里，孙延维老师从最初的定题、开题，到后来的实习、撰写论文、定稿、系统实现，都始终认真及时地督促我们的完成进度，并对我们的毕业设计给予积极耐心的指导和帮助。我的同学们也在我遇到困难时积极回应，帮助我想办法解决，让我的毕业设计得以一步步顺利进行。

在此非常感谢孙延维老师在毕业设计阶段对我给予的耐心指导和帮助，感谢我的同学们给予我的关怀与帮助。