基于SpringBoot的二维码

门禁管理系统

# 摘　要

传统模式下的门禁管理系统中，用户难免遭遇卡片遗失或密码泄露的尴尬，管理方也面临人手不足、专业性欠缺和数据混乱的窘况。

经由“互联网+”思维启发，本项目结合二维码技术，应用主流开发框架，开发了一款二维码门禁管理系统。无需保安或门卫，用户可以通过扫描二维码验证访问权限，管理方可以实时查看门禁记录并管理权限。无论学校、小区还是其他公共场所，各种现代化建筑都可以通过本项目，实现智能且高效的门禁管理，为用户提供更加便捷、更加安全的生活方式。

技术选型方面，本系统以SSM作为主体架构，采用Java语言结合SpringBoot框架开发。数据管理方面，采用Oracle的关系型数据库MySQL进行数据管理，通过My Batis持久层框架，实现数据库与应用程序的数据交互。客户端与服务端之间采用SpringBoot自带的Tomcat服务器进行通信。

**关键词：**二维码；门禁管理系统；SpringBoot；

# Abstract

In the access control management system under the traditional mode, users may inevitably encounter the embarrassment of card loss or password leakage, and the management will also face the dilemma of insufficient manpower, lack of professionalism and data confusion.

Inspired by the thinking of "Internet+", this project combines QR code technology with the main popular development frameworks to develop an access control management system based on QR code technology. Without the need for security guards or gatekeepers, users can verify access permissions by scanning QR codes while ensuring security. Administrators can view access records and manage permissions in real-time. Whether it's schools, residential communities, or other public places, modern buildings can achieve intelligent and efficient access control management through this project, providing a more convenient, smarter, and safer way of life.

In terms of technology selection, this system is built on the SSM architecture, developed using Java language combined with the Spring Boot framework. For data management, Oracle's relational database MySQL is used, and data interaction between the database and the application is achieved through the MyBatis persistence layer framework. Communication between the client and the server is facilitated by the Tomcat server built into Spring Boot.

**Key words:** Access control management system; Spring Boot; Mybatis

目 录

[摘　要 1](#_Toc164945954)

[Abstract 2](#_Toc164945955)

[第一章 绪论 5](#_Toc164945956)

[1.1 立项背景与行业痛点分析 5](#_Toc164945957)

[1.2 主要工作和贡献 7](#_Toc164945958)

[1.3 论文结构安排 7](#_Toc164945959)

[第二章 需求分析 8](#_Toc164945960)

[2.1 系统整体需求结构 8](#_Toc164945961)

[2.2 功能模块需求分析 8](#_Toc164945962)

[第三章 系统总体设计 9](#_Toc164945963)

[3.1 开发技术概览 9](#_Toc164945964)

[3.2 系统架构设计图 9](#_Toc164945965)

[3.2.1 系统的逻辑架构图 9](#_Toc164945966)

[3.2.2 系统的物理架构图 9](#_Toc164945967)

[3.3 系统功能图 9](#_Toc164945968)

[3.4 系统流程图 9](#_Toc164945969)

[3.5 数据库设计 9](#_Toc164945970)

[3.5.1 E-R图设计 9](#_Toc164945971)

[3.5.2 关系表设计 9](#_Toc164945972)

[第四章 关键技术实现 10](#_Toc164945973)

[4.1 xx模块的实现 10](#_Toc164945974)

[4.2 xx模块的实现 10](#_Toc164945975)

[4.3 xx模块的实现 10](#_Toc164945976)

[4.4 xx模块的实现 10](#_Toc164945977)

[第五章 总结与展望 11](#_Toc164945978)

[5.1 总结 11](#_Toc164945979)

[5.2 展望 11](#_Toc164945980)

[参考文献 12](#_Toc164945981)

[致 谢 13](#_Toc164945982)

# 绪论

## 1.1 立项背景与行业痛点分析

社会运转与发展过程中，人们对重要设施和通道的安全防范管理需求催生了门禁系统的出现。现如今，不仅仅是关键通道的出入口，许多智能化的建筑和场所也应用了门禁系统。门禁系统不仅帮助了管理者实现多方面需求，也方便了用户的通行。

但是，门禁系统的实现手段各异，不同管理者之间难以做到互通有无，用户在使用过程时常常会面临一些窘况。设想一下，当用户出门时，需要记忆密码或携带卡片，一两个固定场所或许还游刃有余，但如果遇到了规划外的行程，或是行程过多，出于安全考虑设置的复杂密码或者多张卡片就会造成负担。忘记密码、卡片遗失是常有的事，也难免遇到混淆对应门禁系统的情况。当用户来到目的地前却被门禁阻拦在冰冷的大门前，如热锅上的蚂蚁一般焦急但又求助无门，这是多么尴尬的情况呀！

好在移动互联网兴起，随身携带移动智能终端成为了现代人生活的常态。

诸如手机、智能手表等设备，基于其强大的信息承载能力和便携性，可以作为门禁系统权限凭证的载体。而在信息的存储选择中，二维码具有成本低、易生成识别、信息容量大等优点，特别适合用于门禁系统的信息存储。将之与现代化门禁系统相结合，可以有效解决传统门禁系统存在的诸多问题，有利于提升门禁管理的智能化水平，提高用户凭证的安全性和便携性。

本文将结合传统门禁方式面临的现实问题，将传统门禁管理方式与二维码门禁管理方式进行对比，分析定位两种方式的痛点，并对优先考虑的方向采用雷达图进行对比。痛点定位图如图1.1.1所示。

图1.1.1 传统门禁与二维码门禁的痛点定位图

在该痛点定位图中，主要考虑因素如下：

* **安全性：**门禁系统是否能根据用户权限准确放行和限制人员？

**分析：**

* 传统门禁系统面临密码泄露、卡片遗失情况，容易面临身份冒用的风险
* 二维码门禁系统通过二维码技术进行身份验证，安全性更高，但也存在账户泄露的风险。
* **便携性：**用户携带通行凭证是否足够便携，不会造成负担？

**分析：**

* 传统门禁系统中，用户需携带门禁卡片或记住密码，操作繁琐，便捷性差。
* 二维码门禁系统中，用户只需使用手机扫描二维码即可验证，操作简便，手机在生活中通常随身携带，便捷性更高。
* **成本：**门禁系统的实现和后续运转与维护所花费的成本如何？

**分析：**

* 传统门禁系统通常需要安装大量设备、雇佣看管人员或购买制作卡片，后续也要持续支付人力成本和硬件迭代成本。
* 二维码门禁系统依托二维码技术和互联网服务，减少了人力成本和通行凭证制作成本，后续迭代升级也可通过软件升级实现。
* **智能化：**门禁系统是否可以根据管理需要和用户需要提供智能化功能？

**分析：**

* 传统门禁系统缺乏智能化管理，无法实时监控和数据统计。
* 二维码门禁系统提供了权限管理、实时监控、数据统计等智能化功能，提升了门禁管理的智能化水平。
* **用户体验：**用户在与系统的交互过程中的感受如何，是否诚心如意？

**分析：**

* 在传统门禁系统中，用户需要额外携带卡片或记忆密码，容易造成用户不满。
* 在二维码门禁系统中，用户通过手机扫码即可进出，操作简便，提升了用户体验。

由此可见，传统门禁系统面临的诸多问题，在二维码门禁管理系统中可以得到有效改善。

## 1.2 主要工作和贡献

本文结合门禁系统当前发展情况，结合个人移动智能终端的普及性，旨在设计并开发一款二维码门禁系统的管理端。本文按照软件工程开发流程进行开发，涵盖需求分析、系统设计、分模块编码和测试等环节。

在需求分析阶段，本文根据门禁系统管理方和用户方的不同需求，将给出门禁管理方和用户方的用例图。在系统设计阶段，本文将分析设计数据表，并给出详细的E-R图。之后，本文将根据设计对各个模块进行代码实现，出示关键代码。最后，本文将展示开发成果，展示关键功能。

除了系统的开发和展示外，本文将对开发过程中应用的软件和框架进行简单介绍。

## 1.3 论文结构安排

第一章为绪论部分，介绍本文立项背景和痛点分析、主要工作和贡献以及论文结构安排。

第二章为需求分析部分，本文将给出门禁系统管理方和用户方的功能性与非功能性的需求分析。

第三章为系统总体设计，本文将提供系统架构的详细设计，并给出相应的数据表设计和数据库E-R图。

第四章为关键技术实现，此章节为二维码门禁管理系统的编程实现，将给出核心功能的设计思路和关键代码。

第五章为总结与展望，此章节将总结二维码门禁管理系统的开发过程并展望其中的待实现和优化功能。

# 需求分析

## 2.1 系统整体需求结构

根据针对系统功能详细分析总结，二维码门禁管理系统可以分为用户、设施门禁和出入记录三大模块。如图2.1.1为系统整体需求结构图：

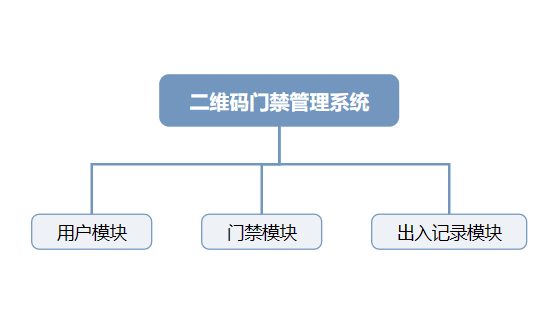


图2.1.1 系统整体需求结构图

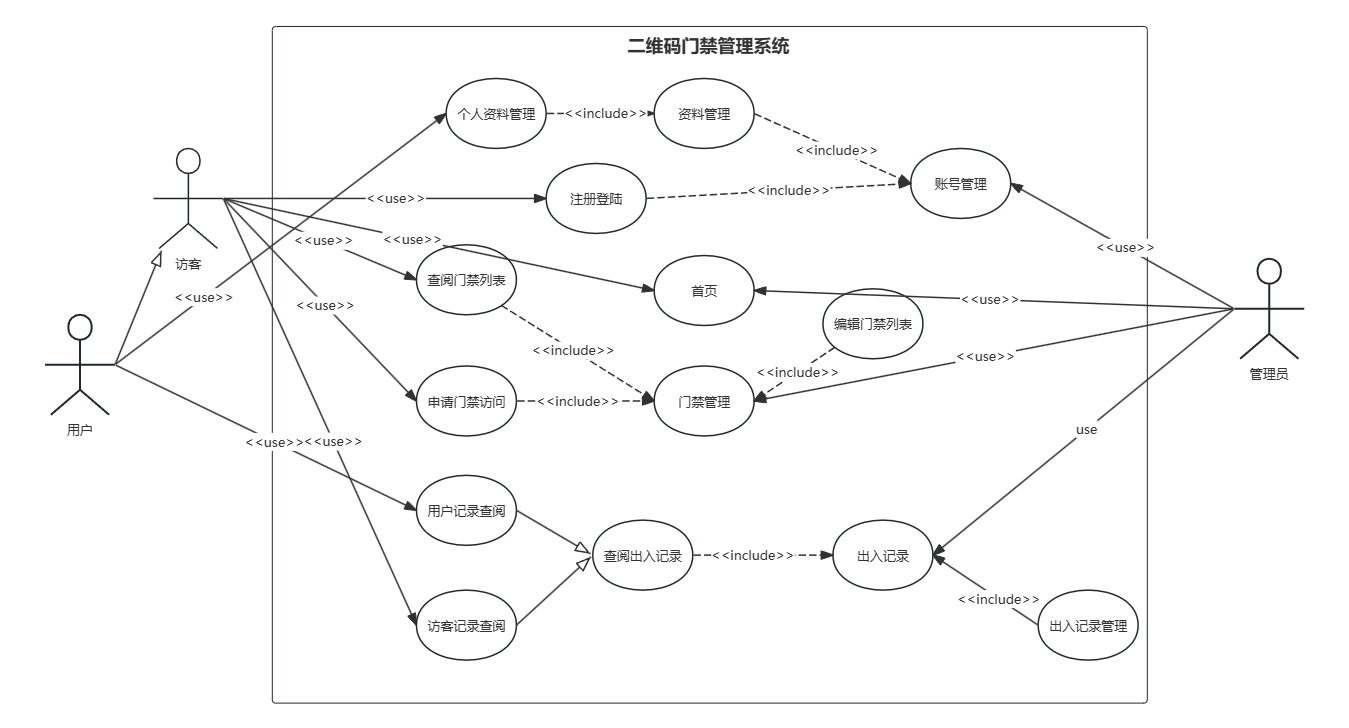
划分使用者，可分为用户、访客和管理员三大身份，用例如图2.1.2所示：

图2.1.2 用例图

根据用例图，简述参与者和用例之间的关系：

1. **访客：**访客可以以访客账户登陆系统，查看门禁列表和申请临时访问，行为将记录在访客出入记录表中。
2. **用户：**用户可以以用户账户登陆系统并编辑个人资料，查看门禁列表和申请门禁权限，行为将记录在用户出入记录表中。
3. **管理员：**管理员可以以管理员账户登陆系统，并对门禁列表、用户和访客列表以及用户和访客的出入记录进行查看和编辑。

## 2.2 功能模块需求分析

1. **用户模块:**

用户模块包含了对系统使用者的管理，包括使用者的登陆注册环节和对资料录入、查询和编辑环节。同时用户模块也区分了不同使用者的权限，访客、用户和管理员拥有不同的权限。用户拥有个人资料，管理员才能管理所有条目。

1. **门禁模块:**

门禁模块是该系统核心部分，包含两个子模块：门禁设施模块和门禁权限模块。门禁设施模块包含了对所有门禁设施的录入、查询和编辑功能，门禁权限包含了特定用户对特定门禁设施的访问权限的申请、审批和查询、编辑功能。

1. **出入记录模块:**

出入记录模块是对使用者行为的记录。访客或用户使用门禁设施时会记录相应的出入记录，模块功能包括对出入记录的记录、查询和编辑。

## 2.3 非功能需求分析

系统分析设计时，应当充分考虑系统的性能需求，包括可承载的最大用户数量、系统是否易用、稳定等。系统设计当以人为本，要模拟用户的使用流程，充分考虑用户的使用体验，确保系统简单易上手。

# 系统总体设计

## 3.1 开发技术概览

系统开发环境：

1. 后端开发工具 IntelliJ IDEA Ultimate 2024.1
2. 前端开发工具WebStorm 2024.1
3. 数据库管理工具DataGrip 2024.1.2
4. 版本管理工具 Git
5. 操作系统Win11家庭版23H2
6. CPU AMD R5 4600H
7. 内存 16G。

### 3.1.1 Java语言

Java语言是一门应用广泛的面向对象设计的计算机编程语言，其在Web应用程序开发中广受开发者和企业的喜爱，在Web应用开发时具有以下优点：

1. **简单且安全：**Java语言在综合C++语言设计优点的同时摒弃了复杂的指针和多重继承等思想。同时通过JVM的内存管理和代码校验，减少了非法的内存操作和针对操作系统的损害行为，使得其同时兼具简单与安全的特性。
2. **面向对象**：Java语言是面向对象的编程语言，通过一切皆对象的编程思想，可以提高在Web应用开发时代码的复用性和模块化，既降低了代码的耦合度，也有利于对代码的管理和组织。
3. **平台无关性与可移植性：**Java语言运行在强大的Java虚拟机（Java Virtual Machine, JVM）上，通过先编译后解释的方式，开发者编写的Java代码文件会先被编译成可被JVM解释的字节码文件，再通过JVM运行。针对不同操作系统开发对应的JVM，通过JVM将统一的字节码文件解释成不同平台对应的机器语言来运行，从而实现了强大的跨平台特性，即“Write Once, Run Anywhere”[2]。

### 3.1.2 Spring框架

Spring Framework是一个开源的轻量级的Java应用程序开发框架。基于其核心的具有控制反转（IOC）特性的容器和面向切面编程（AOP）设计，开发者可以通过Spring框架轻松地构建各种Java应用程序。

本项目中，通过Spring框架，可以简化在开发Java Web应用中遇到的通用的、样板的代码编写和配置环节， 让开发者专注于核心的业务逻辑，提高代码的可读性和可维护性。其轻量级和模块化设计，使得框架具有更高的灵活性和可扩展性，开发者可以根据需要集成相应的模块。同时，通过Spring框架对声明式事务的支持， 方便了开发者对事物的支持和管理。Spring框架同时还支持集成其他应用开发框架，提供更强大的开发能力。

SpringBoot框架则是在Spring的基础上开发的框架，主要用来简化Spring开发框架的开发、配置、调试、部署工作,同时在项目内集成了大量易于使用且实用的基础框架[3]。

### 3.1.3 MySQL数据库

MySQL 是当前流行的关系型数据库管理系统，其具有高性能、可靠性和可扩展性等特点。

在 Web 应用开发中，MySQL不仅具有稳定且高效的性能，还提供了简单易用的管理工具和丰富的开发接口，使得开发者可以轻松地进行数据库的配置、管理和操作。同时，MySQL 的高性能和可扩展性能够满足大规模 Web 应用的需求，其安全特性和可靠性也能够保障数据的安全和稳定性。

## 3.2 系统架构设计

本项目采用主流的B/S（Browser/Server）架构。B/S架构对数据的访问方式为：用户以浏览器作为客户端，通过浏览器访问进入系统，在操作系统过程中客户端向服务端发送相应请求，由后台服务端处理请求并响应数据回浏览器。过程中绝大部分业务逻辑在后台服务端完成，前端浏览器仅作少量逻辑处理。

本项目采用三层架构设计，即表现层、业务层和持久层。表现层用于操作客户端与服务端之间交互的数据，主要设计对界面元素的处理，如前端传递的表单信息，后端响应的数据的分页处理等；业务层承担了系统的主要功能，业务层承上启下，分析处理表现层的请求，从持久层中获取数据并加工整合，最终把数据响应回表现层，核心的业务逻辑均在业务层完成；持久层负责对数据的管理与读写。

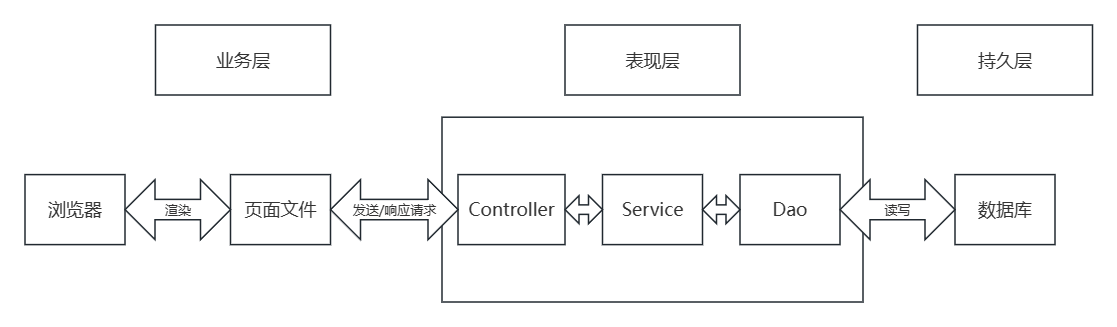
系统架构图如图3.2.1所示：

图3.2.1 系统架构图

## 3.3 系统功能结构

经过需求分析，系统功能模块可划分为：用户模块、门禁模块和出入记录模块。根据使用者的角色，可以区分成访客、用户和管理员三大部分。

系统功能结构图如下图所示：

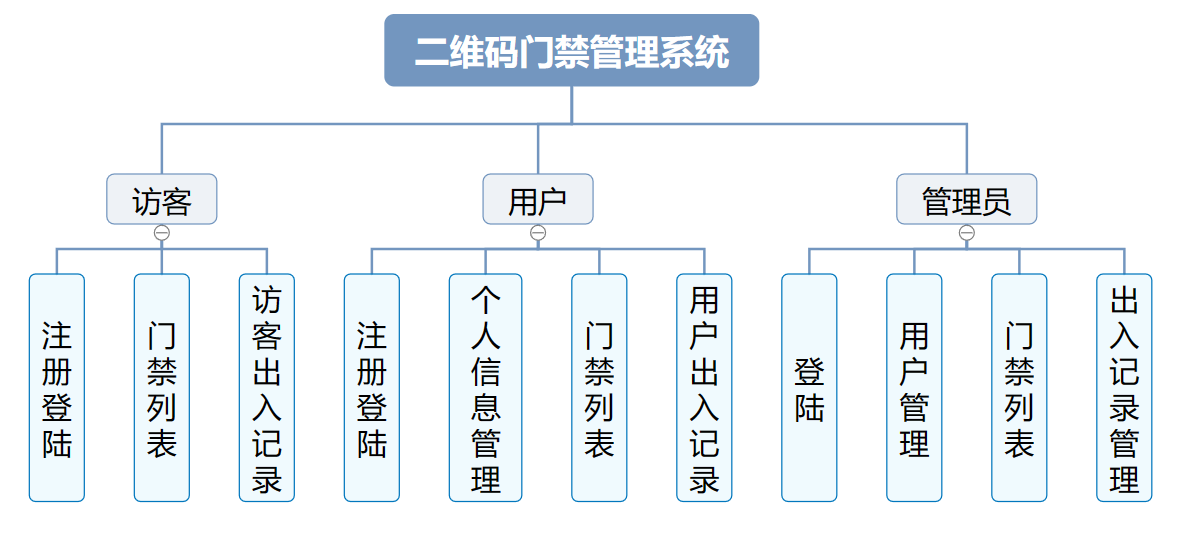


图3.3.1 系统功能结构图

## 3.4 数据库设计

本项目将根据应用程序的实际需求和业务逻辑，设计出合适的数据库的表和关系，并给出数据的存储方式、约束条件和索引等定义，从而实现对数据的有效管理和高效访问。

下文中将展示核心业务功能相关的数据库设计。

### 3.4.1 E-R图设计

经需求分析，在本项目核心功能中，有如下实体：访客、用户、管理员、用户信息、访客信息、用户出入记录、访客出入记录和门禁列表。

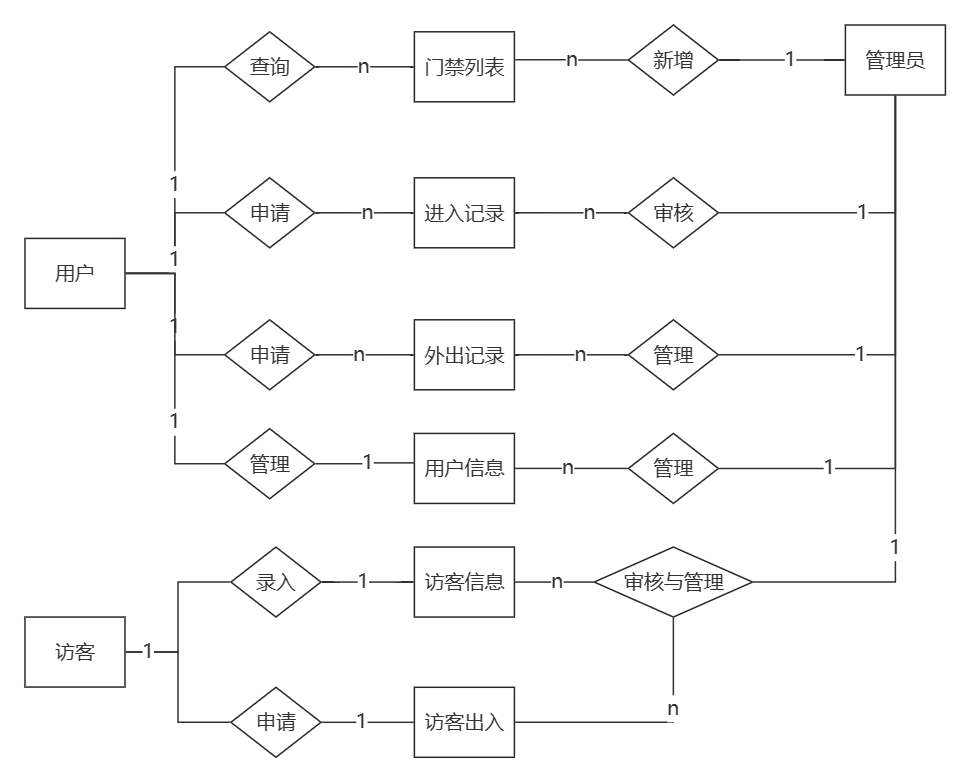
如图3.4.1.1所示，给出系统数据库设计E-R图：

图3.4.1.1 E-R图

### 3.4.2 关系表设计

经需求分析，在本项目中核心功能中，相关的数据表有：访客用户表、普通用户表、用户权限表、门禁列表表、用户进入表、用户外出表、访客出入表。

如下给出相关数据表的详细设计：

1. **访客用户表**

表3.4.2.1 guest\_user表（访客用户表）

| 编号 | 名称 | 数据类型 | 允许空值 | 主键 | 说明 |
| --- | --- | --- | --- | --- | --- |
| 1 | guest\_user\_id | int | N | Y | 访客用户ID |
| 2 | visitor\_name | varchar | Y | N | 访客姓名 |
| 3 | visitor\_gender | varchar | Y | N | 访客性别 |
| 4 | qr\_code | varchar | Y | N | 二维码 |
| 5 | qr\_code\_number | varchar | Y | N | 二维码编号 |
| 6 | examine\_state | varchar | N | N | 审核状态 |
| 7 | recommend | int | N | N | 智能推荐 |
| 8 | user\_id | int | N | N | 用户ID |
| 9 | create\_time | datetime | N | N | 创建时间 |
| 10 | update\_time | timestamp | N | N | 更新时间 |

1. **普通用户表**

表3.4.2.2 ordinary\_users表（普通用户表）

| 编号 | 名称 | 数据类型 | 允许空值 | 主键 | 说明 |
| --- | --- | --- | --- | --- | --- |
| 1 | ordinary\_users\_id | int | N | Y | 普通用户ID |
| 2 | user\_name | varchar | Y | N | 用户姓名 |
| 3 | user\_gender | varchar | Y | N | 用户性别 |
| 4 | building\_no | varchar | Y | N | 楼号 |
| 5 | unit\_no | varchar | Y | N | 单元号 |
| 6 | room\_number | varchar | Y | N | 房间号 |
| 7 | qr\_code | varchar | Y | N | 二维码 |
| 8 | qr\_code\_number | varchar | Y | N | 二维码编号 |
| 9 | examine\_state | varchar | N | N | 审核状态 |
| 10 | recommend | int | N | N | 智能推荐 |
| 11 | user\_id | int | N | N | 用户ID |
| 12 | create\_time | datetime | N | N | 创建时间 |
| 13 | update\_time | timestamp | N | N | 更新时间 |

表3.4.2.5 续

1. **用户权限表**

表3.4.2.3 auth表（用户权限表）

| 编号 | 名称 | 数据类型 | 允许空值 | 主键 | 说明 |
| --- | --- | --- | --- | --- | --- |
| 1 | auth\_id | int | N | Y | 授权ID： |
| 2 | user\_group | varchar | Y | N | 用户组： |
| 3 | mod\_name | varchar | Y | N | 模块名： |
| 4 | table\_name | varchar | Y | N | 表名： |
| 5 | page\_title | varchar | Y | N | 页面标题： |
| 6 | path | varchar | Y | N | 路由路径： |
| 7 | position | varchar | Y | N | 位置： |
| 8 | mode | varchar | N | N | 跳转方式： |
| 9 | add | tinyint | N | N | 是否可增加： |
| 10 | del | tinyint | N | N | 是否可删除： |
| 11 | set | tinyint | N | N | 是否可修改： |
| 12 | get | tinyint | N | N | 是否可查看： |
| 13 | field\_add | text | Y | N | 添加字段： |
| 14 | field\_set | text | Y | N | 修改字段： |
| 15 | field\_get | text | Y | N | 查询字段： |
| 16 | table\_nav\_name | varchar | Y | N | 跨表导航名称： |
| 17 | table\_nav | varchar | Y | N | 跨表导航： |
| 18 | option | text | Y | N | 配置： |
| 19 | create\_time | timestamp | N | N | 创建时间： |
| 20 | update\_time | timestamp | N | N | 更新时间： |

表3.4.2.3 续

1. **门禁列表表**

表3.4.2.4 access\_control\_list表（门禁列表表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 允许空值 | 主键 | 说明 |
| 1 | access\_control\_list\_id | int | N | Y | 门禁列表ID |
| 2 | name\_of\_access\_control | varchar | Y | N | 门禁名称 |
| 3 | access\_control\_number | varchar | Y | N | 门禁编号 |
| 4 | specific\_location | varchar | Y | N | 具体位置 |
| 5 | contracting\_unit | varchar | Y | N | 承包单位 |
| 6 | person\_in\_charge | varchar | Y | N | 负责人员 |
| 7 | repair\_personnel | varchar | Y | N | 维修人员 |
| 8 | recommend | int | N | N | 智能推荐 |
| 9 | create\_time | datetime | N | N | 创建时间 |
| 10 | update\_time | timestamp | N | N | 更新时间 |

1. **进入记录表**

图3.4.2.5 enter\_the\_record表（进入记录表）

| 编号 | 名称 | 数据类型 | 允许空值 | 主键 | 说明 |
| --- | --- | --- | --- | --- | --- |
| 1 | enter\_the\_record\_id | int | N | Y | 进入记录ID |
| 2 | name\_of\_access\_control | varchar | Y | N | 门禁名称 |
| 3 | access\_control\_number | varchar | Y | N | 门禁编号 |
| 4 | specific\_location | varchar | Y | N | 具体位置 |
| 5 | record\_time | datetime | Y | N | 记录时间 |
| 6 | record\_users | int | Y | N | 记录用户 |
| 7 | user\_name | varchar | Y | N | 用户姓名 |
| 8 | building\_no | varchar | Y | N | 楼号 |
| 9 | unit\_no | varchar | Y | N | 单元号 |
| 10 | room\_number | varchar | Y | N | 房间号 |
| 11 | qr\_code | varchar | Y | N | 二维码 |
| 12 | qr\_code\_number | varchar | Y | N | 二维码编号 |
| 13 | door\_opening\_status | varchar | Y | N | 开门状态 |
| 14 | recommend | int | N | N | 智能推荐 |
| 15 | create\_time | datetime | N | N | 创建时间 |
| 16 | update\_time | timestamp | N | N | 更新时间 |

表3.4.2.5 续

1. **外出记录表**

表3.4.2.6 out\_of\_office\_records表（外出记录表）

| 编号 | 名称 | 数据类型 | 允许空值 | 主键 | 说明 |
| --- | --- | --- | --- | --- | --- |
| 1 | out\_of\_office\_records\_id | int | N | Y | 外出记录ID |
| 2 | name\_of\_access\_control | varchar | Y | N | 门禁名称 |
| 3 | access\_control\_number | varchar | Y | N | 门禁编号 |
| 4 | specific\_location | varchar | Y | N | 具体位置 |
| 5 | record\_time | datetime | Y | N | 记录时间 |
| 6 | record\_users | int | Y | N | 记录用户 |
| 7 | user\_name | varchar | Y | N | 用户姓名 |
| 8 | building\_no | varchar | Y | N | 楼号 |
| 9 | unit\_no | varchar | Y | N | 单元号 |
| 10 | room\_number | varchar | Y | N | 房间号 |
| 11 | qr\_code | varchar | Y | N | 二维码 |
| 12 | qr\_code\_number | varchar | Y | N | 二维码编号 |
| 13 | door\_opening\_status | varchar | Y | N | 开门状态 |
| 14 | recommend | int | N | N | 智能推荐 |
| 15 | create\_time | datetime | N | N | 创建时间 |
| 16 | update\_time | timestamp | N | N | 更新时间 |

表3.4.2.6 续

1. **访客出入表**

表3.4.2.7 visitor\_access表（访客出入表）

| 编号 | 名称 | 数据类型 | 允许空值 | 主键 | 说明 |
| --- | --- | --- | --- | --- | --- |
| 1 | visitor\_access\_id | int | N | Y | 访客进出ID |
| 2 | name\_of\_access\_control | varchar | Y | N | 门禁名称 |
| 3 | access\_control\_number | varchar | Y | N | 门禁编号 |
| 4 | specific\_location | varchar | Y | N | 具体位置 |
| 5 | record\_time | datetime | Y | N | 记录时间 |
| 6 | record\_users | int | Y | N | 记录用户 |
| 7 | visitor\_name | varchar | Y | N | 访客姓名 |
| 8 | qr\_code | varchar | Y | N | 二维码 |
| 9 | qr\_code\_number | varchar | Y | N | 二维码编号 |
| 10 | door\_opening\_status | varchar | Y | N | 开门状态 |
| 11 | recommend | int | N | N | 智能推荐 |
| 12 | create\_time | datetime | N | N | 创建时间 |
| 13 | update\_time | timestamp | N | N | 更新时间 |

表3.4.2.7 续

# 第四章 关键技术实现

## 4.1 xx模块的实现

## 4.2 xx模块的实现

## 4.3 xx模块的实现

## 4.4 xx模块的实现

# 第五章 总结与展望

## 5.1 总结

## 5.2 展望

# 参考文献

1. 苏健, 翁文勇. Java程序设计的教学实践与思考[J]. 计算机教育, 2008, (24): 147-149+62.
2. Curtin, Matt. "Write once, run anywhere: Why it matters." Technical Article. http://java. sun. com/features/1998/01/wo (1998).
3. 王永和,张劲松,邓安明,等.Spring Boot研究和应用[J].信息通信,2016(10):91-94.

论文中引用的文献应尽可能引用近期发表的与论文工作直接有关的文献。

各专业可根据各自论文（设计）类型不同，规定参考文献的数量（其中

工程设计型外文文献不少于2篇。实验研究、理论研究、计算机软件等类型论文的外文文献不少于3篇。文献综述型论文外文文献有10篇以上）。

# 致 谢

作者在论文（设计）期间都是在×××教授全面、具体指导下完成进行的。×老师渊博的学识、敏锐的思维、民主而严谨的作风使学生受益非浅，并终生难忘。

感谢×××副教授等在毕业设计工作中给予的帮助。

感谢我的学友和朋友对我的关心和帮助。

不少于2篇。实验研究、理论研究、计算机软件等类型论文的外文文献不少于3篇。文献综述型论文外文文献有10篇以上）。