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，项目以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MyBatis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 traditional access control systems, users often face the inconvenience of lost cards or leaked passwords, while administrators struggle with understaffing, lack of expertise, and data disarray.

Inspired by the "Internet+" concept, this project integrates QR code technology with mainstream development frameworks to develop a QR code-based access control management system. Without the need for security personnel, users can verify access permissions by scanning a QR code, and administrators can view access records and manage permissions in real time. Whether in schools, residential complexes, or other public places, various modern buildings can achieve intelligent and efficient access control through this project, providing a more convenient and secure lifestyle for users.

For the technical stack, the project employs the SSM (Spring, Spring MVC, MyBatis) architecture, utilizing Java and the Spring Boot framework for development. For data management, it uses Oracle's relational database MySQL, and the MyBatis persistence framework to handle data interactions between the database and the application. Communication between the client and the server is facilitated by Spring Boot's built-in Tomcat server.

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

目 录

[摘　要 1](#_Toc166594618)

[Abstract 2](#_Toc166594619)

[第一章 绪论 5](#_Toc166594620)

[1.1 课题研究背景及意义 5](#_Toc166594621)

[1.1.1 研究背景 5](#_Toc166594622)

[1.1.2 研究意义 5](#_Toc166594623)

[1.2 国内外研究综述 7](#_Toc166594624)

[1.3 研究目的与方法 8](#_Toc166594625)

[第二章 需求分析 9](#_Toc166594626)

[2.1 系统整体需求结构 9](#_Toc166594627)

[2.2 功能模块需求分析 10](#_Toc166594628)

[2.3 非功能需求分析 10](#_Toc166594629)

[第三章 系统总体设计 11](#_Toc166594630)

[3.1 开发技术概览 11](#_Toc166594631)

[3.1.1 Java语言 11](#_Toc166594632)

[3.1.2 Spring框架 12](#_Toc166594633)

[3.1.3 MySQL数据库 12](#_Toc166594634)

[3.1.4 Mybatis框架 12](#_Toc166594635)

[3.1.5 Vue框架与Element-UI 13](#_Toc166594636)

[3.2 系统架构设计 13](#_Toc166594637)

[3.3 系统功能结构 14](#_Toc166594638)

[3.4 数据库设计 14](#_Toc166594639)

[3.4.1 E-R图设计 15](#_Toc166594640)

[3.4.2 关系表设计 15](#_Toc166594641)

[第四章 关键功能实现 21](#_Toc166594642)

[4.1 登陆功能的实现 21](#_Toc166594643)

[4.2 访客出入功能的实现 21](#_Toc166594644)

[4.3 用户出入记录功能的实现 22](#_Toc166594645)

[4.4 门禁列表功能的实现 22](#_Toc166594646)

[第五章 总结与展望 25](#_Toc166594647)

[参考文献 26](#_Toc166594648)

[致 谢 27](#_Toc166594649)

# 绪论

## 1.1 课题研究背景及意义

### 1.1.1 研究背景

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。

### 1.1.2 研究意义

本文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：

1. **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。

1. **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。

1. **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。

1. **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。

1. **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 国内外研究综述

门禁系统是通过管制非特定人员进出某通道所使用的软硬件系统。在十九世纪70年代，密码门禁便开始兴起，频繁应用于金融、军事等安全要求严苛的领域。但随着技术的发展，即使拥有较好的安全防护，密码门禁也展现了其如密码泄漏、频繁更换带来的记忆困难等弊病。十九世纪80年代，卡片门禁成为潮流，通过电池感应实现卡片读卡识别，开门迅速、稳定，虚拟的身份标识也适合管控，同时也带来了携带不便、易复制的安全性缺陷与卡片消磁风险等问题。步入二十世纪，生物识别实现了颠覆性进步，刷脸、指纹等开门功能为人们生活带来了智能、无感的门禁方案，然而依然有生物识别的置信度存疑和因光线不足、污秽或指纹缺失等情况导致的频繁识别失败问题。

从二十世纪20年代发展至今，携带智能手机作为个人智能移动终端逐渐成为了人们生活的常态。在中国互联网络信息中心的《中国互联网络发展状况统计报告》中的截止至2023年6月的数据指出，中国的互联网普及率已达76.4%，形成了庞大的网民群体，其规模已达10.79亿人。

从二十世纪80年代开始，二维码技术在国内外陆续成为研究的热门方向，如今已成为一项成熟的技术。二维码是特殊的图片，由一些几何图形在二维方向上以黑白颜色进行规律排列分布所形成。图片中记录的数据符号信息可以解析成二进制中的“0”与“1”，因此可以以编码存储信息。通过光电设备或者图形输入等方式识别解读，就可以提取二维码中的信息进行数据处理。

关于国内外对于二维码技术与门禁系统的研究，徐国辉、陈婕娴[1]研究并介绍了关于二维码的起源、编码原理、技术标准等问题；杨军等人[2]对比了一维码与二维码的编码区别，分析了两者的优缺点并提出了各自的应用场景；Sutheebanjard P，Premchaiswadi W[3]设计了一套在web浏览器上对数据进行快速编码的二维码生成器；徐玲等人[4]设计并实现了一套实现手机二维码编码解码核心功能的系统；宋宝磊[5]针对门禁管理系统的构成、特点及应用情况进行了详细介绍；Sandhu R S和Samarati P[6]讨论了常见的访问控制策略并简要描述了访问控制管理方案；Kao Y W[7]等人实现了一套基于二维码技术存储信息的物理访问控制系统。

## 1.3 研究目的与方法

本文将结合门禁系统当前发展情况与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。

本文将按照软件工程开发流程进行开发，遵循软件开发流程，进行需求调研和分析，建立完整的分析模型，设计对应功能模块的程序流程、算法和数据结构，再根据统一、规范的程序编写规则进行编程实现，并进行相应的应用测试，确保程序的可用性、可靠性、实用性。

# 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：

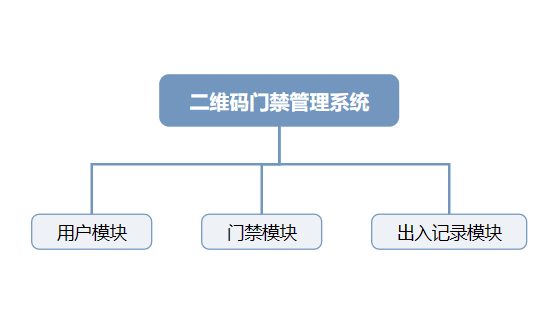


图2.1.1 系统整体需求结构图

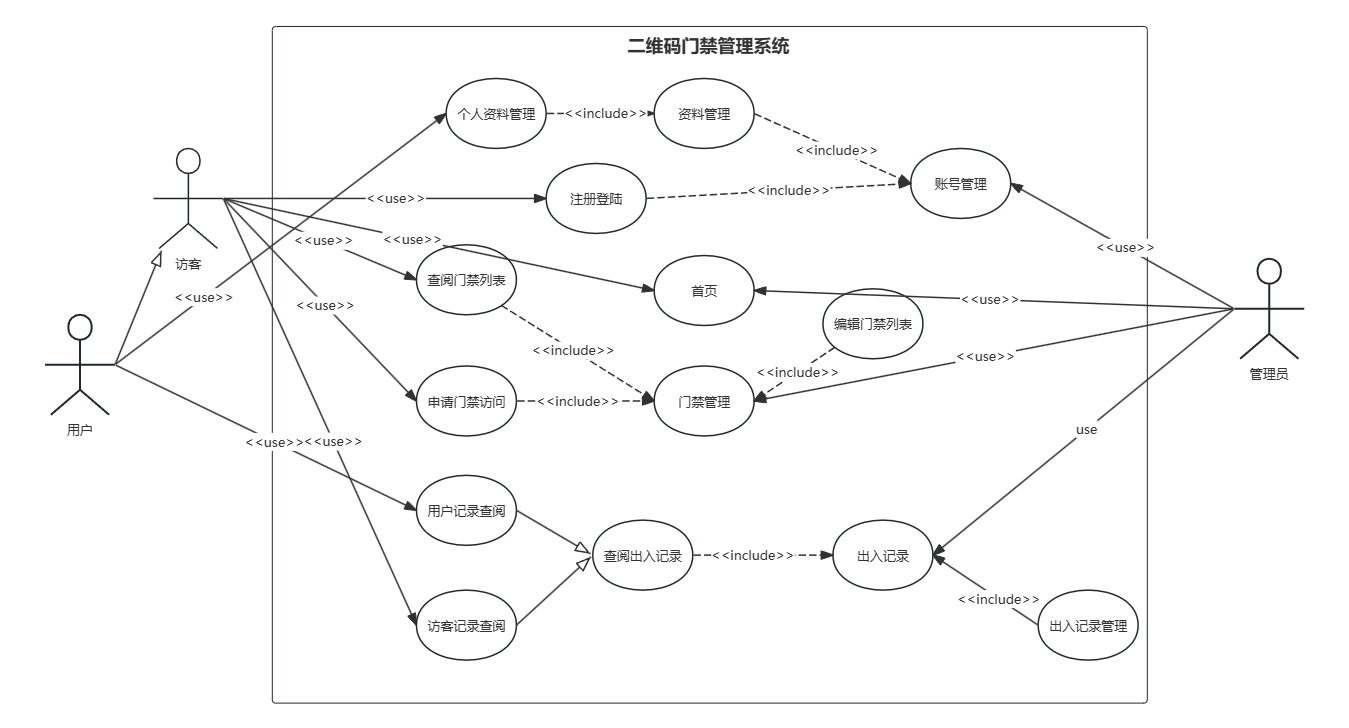
划分使用者，可分为用户、访客和管理员三大身份，用例如图2.1.2所示：

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”[9]。

### 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[10]。

### 3.1.3 MySQL数据库

MySQL 是当前流行的关系型数据库管理系统，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。

### 3.1.4 Mybatis框架

MyBatis 是一种流行的优秀的持久层框架，用于简化数据库中的JDBC操作，并将 SQL 语句与 Java 代码进行解耦。MyBatis 的核心思想是通过 XML 或注解配置 SQL 映射，将 SQL 查询、更新、删除等操作映射到 Java 接口或类的方法上，从而实现数据的持久化操作。

MyBatis-Plus 是 MyBatis 的增强工具包，提供了更多的便捷功能和增强功能，使得开发者能够更加轻松地进行数据库操作。

在本项目中，通过Spring集成Mybatis-Plus，可以简化CRUD操作，避免繁琐的实体类与Mapper创建操作，同时通过其提供的分页插件、IDEA插件等工具简化开发，提高开发效率。

### 3.1.5 Vue框架与Element-UI

Vue.js 是一种流行的开源 的JavaScript 前端框架，简单、轻量，开发者可以通过Vue框架高效且轻松地构建出交互式的 Web 用户界面。

Element-UI 是一套基于 Vue.js 的开源 UI 组件库，由饿了么前端团队开发和维护。Element-UI 提供了丰富的 UI 组件和样式和对主题定制和国际化支持等功能。Element-UI 的组件风格简洁明了，易于使用和定制，能够帮助开发者快速构建美观、功能丰富的 Web 应用界面。

Vue.js 和 Element-UI 的结合，使得前端开发变得更加高效和便捷。Vue.js 提供了强大的数据绑定和组件化开发能力，而 Element-UI 则提供了丰富的 UI 组件和样式，两者结合使用可以快速构建出符合用户期待的现代化 Web 应用界面。

## 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。

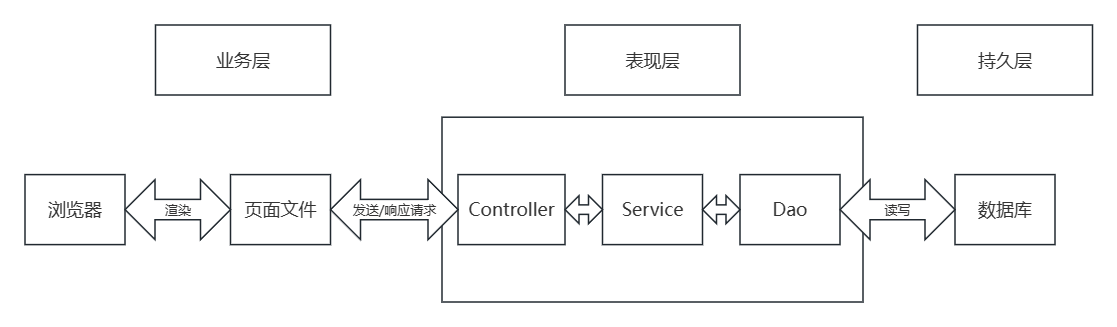
系统架构图如图3.2.1所示：

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：

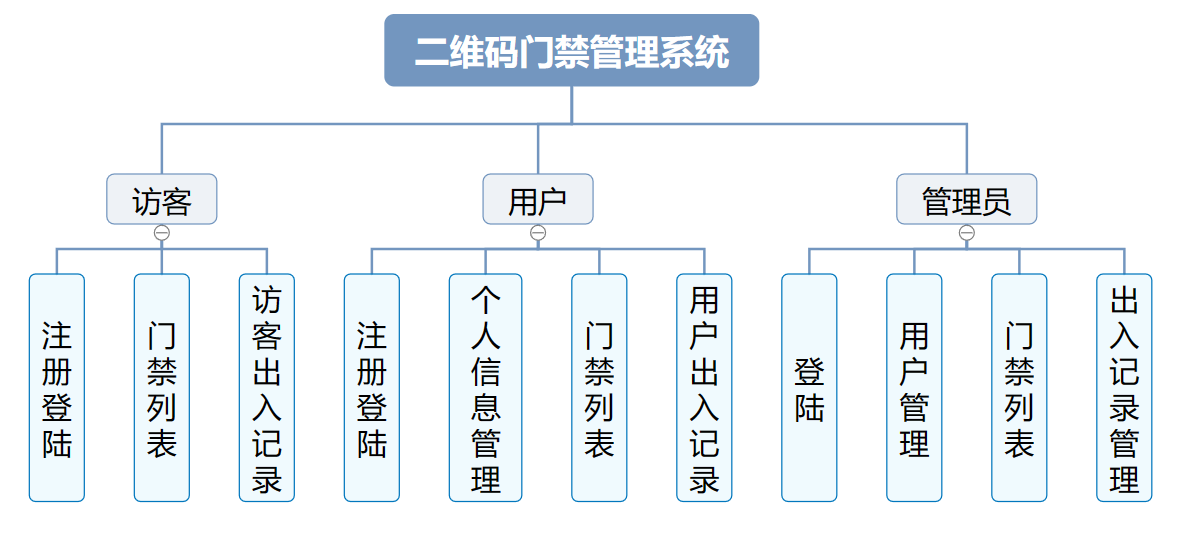


图3.3.1 系统功能结构图

## 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。

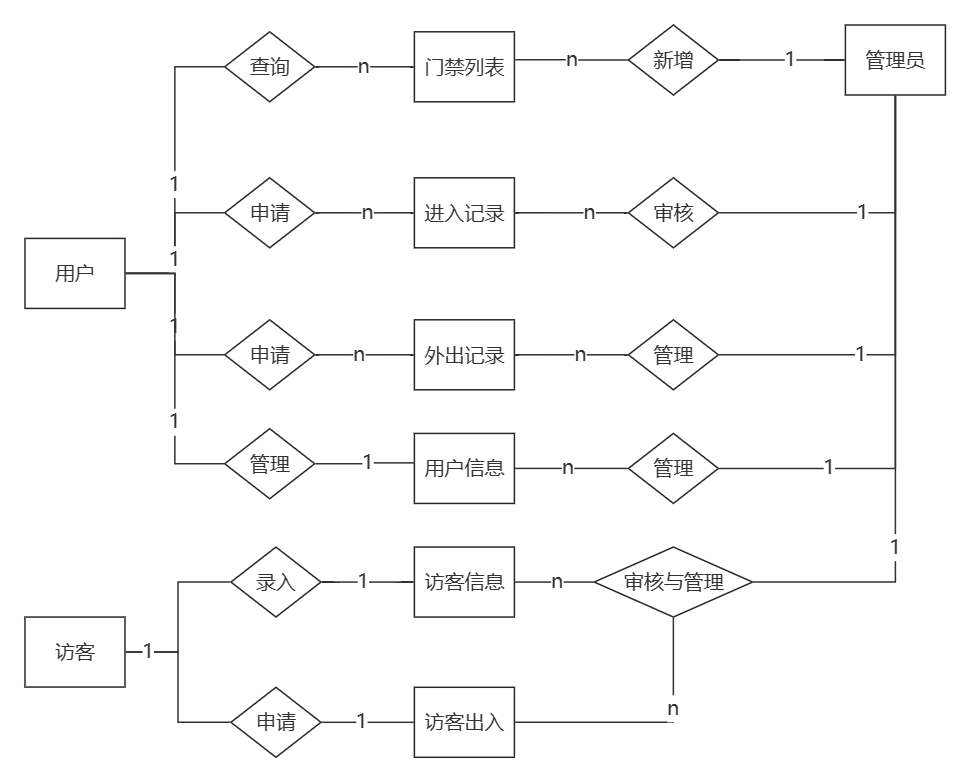
如图3.4.1.1所示，给出系统数据库设计E-R图：

图3.4.1.1 E-R图

### 3.4.2 关系表设计

经需求分析，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 |

表3.4.2.1 续

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 |

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： |

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 |

表3.4.2.4 续

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 |

# 第四章 关键功能实现

## 4.1 登陆功能的实现

系统首页为登录窗口，用户只有在登录成功后才可以访问并操作系统。

用户通过登陆窗口登陆，提交登陆信息表单，经过数据合法性校验和数据库中账号密码对比，判定成功后执行页面跳转，进入系统操作页面。

登录流程图如下所示：



图4.1.1登录流程图

注意，在登陆与注册过程中，为确保用户账户的安全性，注册时密码将不会以明文存储。密码将采用加盐方式，对密码明文插入随机字符串作为盐值后执行MD5加密，数据库中仅存储密码的加密字符串与盐值。登陆时，将对用户输入的密码执行与注册时相同的加盐与加密处理，再与数据库中的加密密码做对比，校验成功才允许登陆。

## 4.2 访客出入功能的实现

访客进出过程中，系统通过用户id查询数据库获得访客信息，查找到访客信息后填写并提交访客出入表单，后台将访客信息插入到访客进出表单中。

访客进出流程图如下所示：



图4.2.1访客进出流程图

## 4.3 用户出入记录功能的实现

用户出入记录功能可以根据用户id对相应的记录执行增加、删除、更改、查询操作。功能流程与访客出入功能相似，程序编码实现与下文中门禁列表功能实现相似。

## 4.4 门禁列表功能的实现

门禁列表功能主要为录入、更改、删除和查询列表操作。

在前后端交互过程中，用户通过浏览器交互填写信息，点击对应按钮后由前端组织并提交表单信息。后端Controller控制层接受表单信息，针对表单信息做数据处理后交给Service业务层，Service层进行业务逻辑处理，封装数据对象后交给DAO持久层执行对数据库的增删改查操作。DAO层获得执行结果后将数据作为方法返回传递回Service层，Service层执行业务逻辑后将数据作为方法返回传递回Controller层，Controller层组织数据并形成对前端请求的响应。前端通获得响应的数据后就可以处理页面文件和页面元素，最终交由浏览器渲染形成用户看到的反馈。

录入门禁列表信息时，管理员填写门禁信息，前端发起携带门禁信息表单数据的Post请求。后端通过查询数据库验证当前用户的权限后（其他操作执行前同样会进行权限校验），将门禁信息插入到数据库中。插入时，会根据当前时间在数据表create\_time字段中写入插入时间。



图4.4.1门禁列表流程图

修改门禁列表信息时，管理员选定具体的门禁列表时，会先根据选中的门禁信息id提交Get请求查询具体数据并渲染到用户界面。管理员根据现有信息进行修改，修改后的门禁信息作为表单信息提交Post请求，通过门禁列表id信息查询数据库，确认相应的门禁信息后（数据的修订即更新、删除时必须执行查询操作，避免操作过程中对其他记录执行错误操作），用新的门禁信息修改旧门禁信息字段，实现门禁列表的修改操作。修改时，会根据当前时间在数据库update\_time字段中写下修改时间。

删除门禁列表信息时，管理员确定删除后，会提交Post请求，数据库查找到具体数据后，对数据库中的记录执行删除操作。此操作为逻辑删除过程，将不会使用“Delete”语句，通过数据表中增添默认为0的“is\_delete”字段作为逻辑删除字段，删除时实际使用“Update”语句修改记录的“is\_delete”字段为1实现逻辑上的删除。当执行查询操作时，默认在“Where”条件后添加“is\_delete = 0”的判断。

# 第五章 总结与展望

本论文针对传统门禁管理系统存在的卡片遗失、密码泄露、管理效率低下等问题，提出通过使用二维码作为门禁凭证，取代传统的卡片或密码，实现更加便捷、安全和高效的门禁管理方式，并给出了二维码门禁管理系统的设计与实现方案。

系统采用了Spring Boot框架，结合MyBatis持久层框架，实现了数据的管理和交互，同时借助Vue和Element-UI等前端框架，实现了用户友好的界面设计。

通过对系统的设计与实现，作者验证了二维码门禁管理系统的可行性和有效性。在实际使用中，该系统能够提高门禁管理的效率，减少了卡片遗失和密码泄露的风险，为用户提供了更加便捷、安全的门禁体验。

然而，当前系统仍存在一些问题和不足之处，例如在大量用户访问的并发环境下数据一致性与安全性、系统的效率等问题。拟在之后的更新中加入Redis非关系型数据库作为数据缓存，加快运行效率，同时通过乐观锁等机制提高数据的并发安全性。

总之，在未来，作者将不断完善门禁管理系统的功能与性能，增强系统的安全性和稳定性，为用户提供更好的体验。

# 参考文献

1. 徐国辉, 陈婕娴.手机二维码技术原理及应用[J].信息与电脑: 理论版, 2013 (1):18-19.
2. 杨军，刘艳杜彦蕊.关于二维码的研究和应用[J].应用科技，2002, 11(1).
3. Sutheebanjard P, Premchaiswadi W. QR-code generator[C]//2010 Eighth International Conference on ICT and Knowledge Engineering. IEEE, 2010: 89-92.
4. 徐玲，蒋欣志，张杰.手机二维码识别系统的设计与实现[D].，2012.
5. 宋宝磊.门禁管理系统的研究与应用[J].中国安，2009(10):55-57.
6. Sandhu R S, Samarati P. Access control: principle and practice[J]. IEEE communications magazine, 1994, 32(9): 40-48.
7. Kao Y W, Luo G H, Lin H T, et al. Physical access control based on QR code[C]//2011 international conference on cyber-enabled distributed computing and knowledge discovery. IEEE, 2011: 285-288.
8. 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.
9. Curtin, Matt. "Write once, run anywhere: Why it matters." Technical Article. http://java. sun. com/features/1998/01/wo (1998).
10. 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.

# 致 谢

作者在论文设计期间都是在温智华老师全面、具体指导下完成进行的。温老师渊博的学识、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。

感谢我的学友和朋友对我的关心和帮助。