班　级　1703014
学　号　17039110002

西安电子科技大学

本科毕业设计论文

题　　目　基于的视频实时监控系统的设计与实现

学　　院　计算机科学与技术

专　　业　计算机科学与技术

学生姓名　陈鑫辉

导师姓名　鱼滨

毕业设计（论文）诚信声明书

本人声明：本人所提交的毕业论文《基于 springboot 的视频实时监控系统的设计与实现》是本人在指导教师指导下独立研究、写作成果，论文中所引用他人的无论以何种方式发布的文字、研究成果，均在论文中加以说明；有关教师、同学和其他人员对本文本的写作、修订提出过并为我在论文中加以采纳的意见、建议，均已在我的致谢辞中加以说明并深致谢意。

本文和资料若有不实之处，本人承担一切相关责任。

论文作者：_____（签字） 时间：　　年　　月　　日

指导教师已阅：_____（签字） 时间：　　年　　月　　日

摘　要

随着近年来各类智能设备数量的爆发性增长，视频监控系统正朝着数字化、智能化、网络化、人性化的方向发展。基于此，本文设计并实现了一个基于 Spring Boot 的视频实时监控系统。本文首先介绍说明了流媒体传输协议的定义，分析比较了 RTMP、HTTP-FLV 和 HLS 这三个常用的流媒体传输协议，对三者的原理进行了深入的分析，并讨论了它们在实际使用过程中各自的利弊。在这之后介绍了基于 Java 的 Spring Boot 开发框架的使用方法，并对其特点、设计理念和功能做了详细的阐述。此外，也介绍了对象拷贝技术 MapStruct、基于内存的数据库 Redis 和基于 RBAC 的权限系统。最后对实时监控视频系统的发展趋势作了分析。在此次毕业设计的任务中，运用到了 Spring Boot、Redis、MapStruct 和 Vue.js 等技术，逐步完成了视频实时监控系统的技术方案设计。最终实现了一个基于 Spring Boot 的视频实时监控系统，具备实时监控视频查看、监控视频保存和检索等功能。

关键词：流媒体传输技术　Spring Boot　HTTP-FLV　实时视频监控系统

Abstract

With the explosive growth in the number of various types of smart devices in recent years, video surveillance systems are developing in the direction of digitization, intelligence, networking, and humanization. Based on this, this paper designs and implements a real-time video surveillance system based on Spring Boot. This paper first introduces the definition of streaming media transmission protocol, analyzes and compares the three commonly used streaming media transmission protocols RTMP, HTTP-FLV and HLS, conducts an in-depth analysis of the principles of the three, and discusses their pros and cons in actual use. After that, the article introduces the use of Java-based Spring Boot development framework, and elaborates its characteristics, design concepts and functions in detail. In addition, the object copy technology MapStruct, the memory-based database Redis and the RBAC-based permission system are also introduced. Finally, the development trend of real-time surveillance video system is analyzed. In the task of this graduation project, technologies such as Spring Boot, Redis, MapStruct and Vue.js were used to gradually complete the technical scheme design of the real-time video surveillance system. Finally, a real-time video monitoring system based on Spring Boot is realized, with functions such as real-time monitoring video viewing, monitoring video storage and retrieval.

Key words: streaming media transmission protocol Spring Boot HTTP-FLV real-time video surveillance system

目　录

第一章　引言

视频监控技术的发展，给人们的生活和生产带来了极大的便利和充分的安全保障。视频监控系统是安防领域中的研究热点，随着近年来各类智能设备数量的爆发性增长，视频监控系统正朝着数字化、智能化、网络化、人性化的方向发展。随着人们生活水平的提高，人们更多的把目光投向了如何更好的保护自己和家人的人身安全以及经济财产，因此以前只出现在公共场所的监控摄像头，如今在人们的家中就能看到它们的身影，希望通过它们可以随时随地的对人或物进行安放监控。

视频实时监控系统是一个可以实时对远程监控摄像头采集的画面进行实时监测的平台。Web 端的视频实时监控系统可以依托手机、电脑等支持网页访问的设备进行查看，可以做到只要有网络，就可以随时随地查看监控视频。传统的视频监控系统需要在一个局域网内，在一台终端机器上才能实现监控视频的查看。本文提出了一种基于 Spring Boot 的视频实时监控系统的设计思路以及实现方法。

第二章　关键技术

本章将介绍系统开发设计过程中使用到的关键技术。

2.1　流媒体传输技术

实时的视频监控系统需要基于流媒体传输技术来实现，需要摄像头本身或者后置的图像处理服务器支持流媒体传输，常见的三种流媒体传输技术有：RTMP、HTTP-FLV 和 HLS。

2.1.1　流媒体以及流式传输

流媒体是指将一连串的媒体数据（包含音频、视频数据）压缩后，采用流式传输的方式发送数据，在网上即时传输影音以供观赏的一种技术与过程，此技术使得数据包得以像流水一样发送；如果不使用此技术，就必须在使用前下载整个媒体文件。流媒体在播放前不会下载整个文件，只将开始部分存入内存，同时也会在用户访问时对数据包进行缓存，让媒体数据正确地输出，流媒体数据流随时传送随时播放。

流式传输是指客户端通过链接流媒体服务器实时传输音频、视频数据，实现“边下载边播放”。流式传输是实现流媒体的关键技术。流式传输时，音频、视频等时基媒体由音视频服务器向用户计算机的连续、实时传送，用户不必等到整个文件全部下载完毕，而只需经过几秒或十数秒的启动延时即可进行观看。当音频、视频等时基媒体在客户机上播放时，文件的剩余部分将在后台从服务器内继续下载。流式不仅使启动延时成十倍、百倍地缩短，而且不需要太大的缓存容量。流式传输避免了用户必须等待整个文件全部从网络上下载才能观看的缺点。

2.1.2　基于 RTMP 的流媒体传输技术

实时消息传输协议（即 Real-Time Messaging Protocol，缩写 RTMP）最初是 Macromedia 公司为了满足在互联网上传输流媒体音视频而开发的一个私有协议。

RTMP（Real Time Messaging Protocol）是由 Adobe 公司基于 Flash Player 播放器对应的音视频 flv 封装格式提出的一种，基于 TCP 的数据传输协议。本身具有稳定、兼容性强、高穿透的特点。常被应用于流媒体直播、点播等场景。

2.1.3 基于 HTTP-FLV 的流媒体传输技术

2.1.4 基于 HLS 的流媒体传输技术

2.2 Spring Boot 开发框架

2.2.1 概述

Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。

Spring Boot 不仅继承了 Spring 框架原有的优秀特性，而且还通过简化配置来进一步简化了 Spring 应用的整个搭建和开发过程。在 Spring Boot 出现之前，使用 Spring 框架进行应用开发需要编写繁琐的配置文件，然而这些配置文件的内容大多都是类似的。

2.2.2 控制反转特性

Spring 框架具有控制反转（IOC）特性，IOC 旨在方便项目维护和测试，它提供了一种通过 Java 的反射机制对 Java 对象进行统一的配置和管理的方法。

2.2.3 面向切面编程

Spring 框架具有面向切面编程（AOP）框架，SpringAOP 框架基于代理模式，同时运行时可配置；AOP 框架主要针对模块之间的交叉关注点进行模块化。

2.3 对象拷贝技术 MapStruct

2.4 内存数据库 Redis

2.5 RBAC 权限系统

第三章　系统总体设计

本章将详细阐述系统的总体设计。对毕业设计题目分析调研过后，可以将设计需求拆分为一下五点。

1. 链接安防摄像头

2. 实时查看视频监控

3. 视频按时间顺序保存

4. 视频检索

5. 前端页面开发和后端服务开发

3.1　实体关系设计

本文的目标是设计并实现一个基于 Spring Boot 的视频实时监控系统。一个完整的系统必然设计到用户，因此就需要用户实体。由于是视频监控系统，那么有监控摄像头设备，因此需要设备实体。此外还需要对监控摄像头产生的录像文件进行保存并支持检索，所以需要视频文件实体。

综上所示，本视频实时监控系统包括三个实体：用户、设备和视频文件。本系统的实体关系图如下所示。

通过实体关系图，可以推导出数据库表结构和表关系，同样如下图所示。

3.2　交互接口设计

3.3　系统模块设计

系统设计分为五个模块，如下表所示

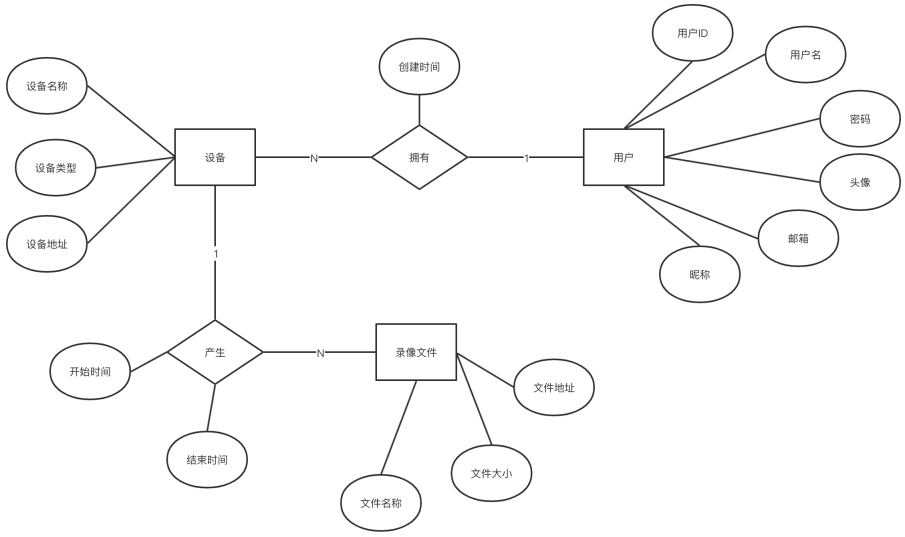

图 3.1　实体关系图

表 3.1　这是一个长表格

模块名称	模块功能
公共模块	提供公共工具的能力
数据存储模块	提供持久组件的读写的能力
业务逻辑模块	系统的主要业务逻辑
视频文件存取模块	离线视频文件的存取
前端接口交互模块	为前端提供接口交互

3.3.1　公共模块

公共模块（monitoring-system-common）提供公共工具的能力。模块内主要包括开发过程中需要用到的注解（annotation）、常量（constant）、枚举（enumerate）、异常（exception）以及公共的工具封装（utility）。

公共模块中封装的工具包括时间转换工具类、HTTP 请求工具类、Spring 工具类以及登陆验证码工具类。

时间转换工具类提供将 java.util.Date 对象转为具有一定格式的字符串（例如“yyyy-MM-dd HH:mm:ss”），同时也提供逆向转换的能力，即将字符串格式的时间转化为 java.util.Date 对象。

HTTP 请求工具类封装了 HTTP 请求中相关的 GET、POST 两种请求方法，借助第三方依赖 OkHttp 二次开发实现。支持带参数的 GET 请求、带请求体参数的 POST 请求。

Spring 工具类提供了对 Spring 上下文的 ApplicationContext 的静态访问，可

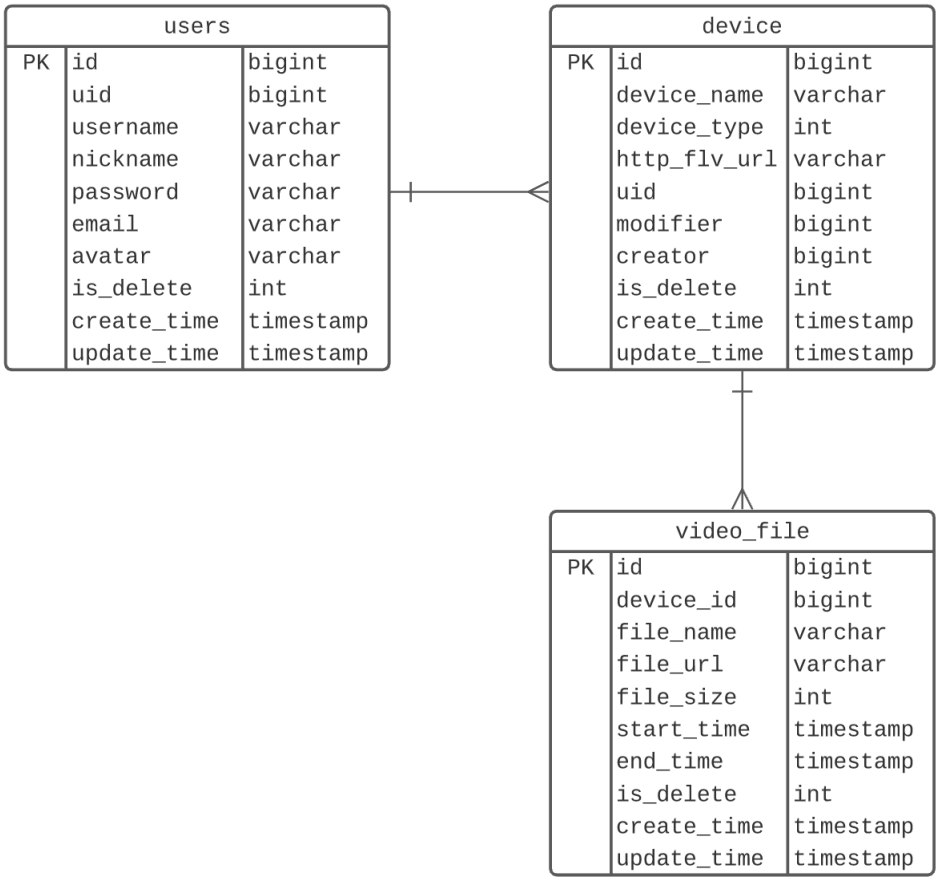

图 3.2　数据库表结构和表关系

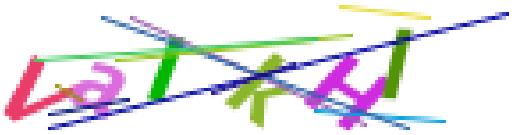

图 3.3　登陆验证码

系统设计使用 Mybatis 框架对数据库进行访问，因此该模块中的大部分代码有时有 Mybatis-Generator 自动生成，包括实体对象类、Mapper 接口和 Mapper 配置文件。除此之外，该模块还包括了各种实体的封装类，比如前端参数封装类（用于封装前端传递给后端的参数）、HTTP 响应类（封装后端接口返回对象）以及各个模块使用的实体类（VO、BO）。

该模块还负责对象之间的转换，即将 VO 转换成 BO 或者将 BO 转换成 BO 等。该功能是通过对象拷贝技术 MapStruct 生成的对象转换代码实现的。

3.3.3 业务逻辑模块

业务逻辑模块（monitoring-system-service）提供业务逻辑处理的功能。该模块是本系统最核心的一个模块，主要负责业务模块的逻辑应用设计，支持登陆鉴权、设备信息的增删改查、个人信息修改等。

业务逻辑模块的主要设计思路是面向接口编程。接口主要用于描述类具有什么功能，而并不给出每个功能的具体实现。一个类可以实现一个或多个接口，并在需要接口的地方，随时使用实现了相应接口的对象。

首先设计接口，再设计其对应的实现的类，接着利用 Spring 的依赖注入特性，就可以在上层模块声明接口类型的变量，而无需关心其具体实现。换句话说，业务逻辑模块的变动不会对上层模块的调用产生任何影响。

这就是软件六大设计原则之一的依赖倒置原则，即面向接口编程。上层模块不应该依赖底层模块，它们都应该依赖于抽象。抽象不应该依赖于细节，细节应该依赖于抽象。上层模块不应依赖底层模块，即上层的业务模块不应该依赖底层的实现模块。如：人出行使用交通工具，只需要知道是交通工具就可以，不用知道是哪种具体的交通工具。抽象不应该依赖于细节，细节应该依赖于抽象。在具体的 Java 编程中，抽象指代接口、抽象函数，统称为接口，而细节指代接口的具体实现。

3.3.4 视频文件存取模块

视频文件存取模块（monitoring-system-media）提供实时监控视频录制和离线查看检索功能。该模块提供了两个 HTTP 的接口，分别是监控回放文件的列表查询接口和监控回放文件下载接口。对于已经在本系统中绑定的监控摄像头设备，每隔一分钟视频文件存取模块会执行定时任务，对该设备进行录制，并且录制的时长为 80 秒，目的是做到前后视频文件能有重合部分，防止视频数据丢失。

对于视频文件的存放，可以在本地文件系统和云存储之间无缝切换。云存储采用阿里云的 OSS 对象存储服务或本地搭建的 Hadoop 分布式存储系统。对于视频文件的读取，该模块提供了一个 GET 方法的 HTTP 接口，入参为视频文件的名称，返回值为该文件内容。同时还提供搜索功能，支持按照监控视频的开始时间和结束时间进行范围查找。

3.3.5　前端接口交互模块

前端接口交互模块（monitoring-system-web）提供与前端交互的能力。该模块负责接受前端的请求，然后调用下层模块（如视频文件存取模块）提供的接口方法，实现业务逻辑，然后将下层模块的返回值进行一定程度的封装后返回给前端进行展示。

3.4　权限系统设计

第四章　系统实现

致　谢

虽为致谢环境，其实就是一个 Chapter，为啥这么费事？因为，致谢一章没有编号。直接使用 \chapter*{}的话，页眉又不符合工作手册要求，而且要往目录中添加该章节，还需要添加两行代码；为了简单快捷的设计出符合要求，又方便用户使用，只能借 \backmatter 模式和本模板自定义的 \comtinuematter 模式配合环境来做了。

附录 A　数据

这里是附录的数据部分，其实是瞎写；来看个公式编号对不对，如式(**??**)所示。

$$h_i = \frac{\max_{j=1}^{N}\{\frac{c_j}{f_j}\} - \frac{c_i}{f_i}}{\max_{j=1}^{N}\{\frac{c_j}{f_j}\} - \min_{j=1}^{N}\{\frac{c_j}{f_j}\}} \tag{A-1}$$

A.1　放松一下

A.1.1　惊鸿一面

A.1.2　无题

A.2　代码

代码 A.1　C++ code

```
/*
 Program: Hello world;
 Author: Stick Cui;
 Time: 2015/12/04.
*/
#include<stdio.h>

int main()
{
    printf("Hello world!");
    return 0;
}
```

Java code

```
/**
 * Program: Hello world;<br>
 * Time: 2015/12/04.
 * @author Stick Cui
 */
public class JavaTest{
    /**
     * The main methord.
     * @param args The parameter when the methord is called.
     */
    public static void main(String[] args){
        System.out.println("Hello world!");
    }
}
```

```
%floyd 算法通用程序，输入 a 为赋权邻接矩阵
% 输出为距离矩阵 D, 和最短路径矩阵 path
function [D,path]=floyd(a)
n=size(a,1);
D=a;
path=zeros(n,n);
for i=1:n
    for j=1:n
        if D(i,j)~=inf
            path(i,j)=j;
        end
    end
end
for k=1:n
    for i=1:n
```

```
        for  j=1:n
            if  D(i,k)+D(k,j)<D(i,j)
                D(i,j)=D(i,k)+D(k,j);
                path(i,j)=path(i,k);
            end
        end
    end
end
% 配合 floyd 算法的后续程序, s 为源点, t 为宿点
% L 为长度, R 为路由
function  [L,R]=router(D,path,s,t)
L=zeros(0,0);
R=s;
while 1
    if  s==t
        L=fliplr(L);
        L=[0,L];
        L=L(end);
        return
    end
    L=[L,D(s,t)];
    R=[R,path(s,t)];
    s=path(s,t);
    if  s==0
        return
    end
end
```

参考文献

[] 许嵩. 惊鸿一面 [A]. 海蝶音乐. 不如吃茶去 [C]. 北京：北京海蝶音乐有限公司, 2014, 8.