

太原工业学院

毕 业 设 计

派出所办案流程标准化智能 管理平台的设计与实现

学生姓名： 吴海飞 学号： 132054401

系 部： 计算机工程系

专 业： 计算机科学与技术

指导教师： 杨慧炯（副教授）

二〇一七年六月

诚信申明

本人郑重声明：本设计（论文）及其研究工作是本人在指导教师的指导下独立完成的，在完成设计（论文）时所利用的一切资料均已在参考文献中列出。

本人签名：

年 月 日

毕业设计（论文）任务书

设计（论文）题目：____派出所办案流程标准化智能管理平台的设计与实现____

系部：____计算机工程系____专业：____计算机科学与技术____学号：____

132054401

学生：____吴海飞____指导教师（含职称）：____杨慧炯（副教授）____

1. 课题意义及目标

通过研发山西公安派出所办案流程标准化智能管理平台（简称智能化派出所系统），可以实现对警务作业全过程的标准化工作指导，并对进入办案区的犯罪嫌疑人进行符合国家法律的标准化提示提醒及各类权利义务告知。从而实现对各类执法作业的信息化记录、流程化管理、全方位监督。

2. 主要任务

- （1）用户信息管理。完成系统用户的注册、审核、及角色分配。
- （2）角色管理。完成系统角色的设置及权限分配。
- （3）采集设备管理。完成采集设备和 RIFD 手环的注册及开通、注销。
- （4）涉案人员信息管理。通过读取二代身份证，完成涉案人员信息的注册及采集、维护。
- （5）采集信息填报。在不同区域，按照相应功能和用户角色权限，完成采集数据的填报和上传保存。
- （6）系统日志管理。完成系统操作及通讯日志的保存。
- （7）数据库管理。实现系统数据库的备份和恢复。
- （8）涉案信息查询汇总。根据具体应用需求，完成相关数据的汇总查询。

3. 主要参考资料

- [1] 王占军;王伟; 赵志刚. 基于 FTP 协议的文件传输组件设计与实现 [J/OL]. 沈阳师范大学学报(自然科学版), 2012-02, 中国知网 <http://epub.cnki.net/>.
- [2] 刘继元. 基于树莓派的物联网应用 [J/OL]. 探索与观察 (2016 年 08 期), 2016-04, 中国知网 <http://epub.cnki.net/>.
- [3] 毕翼. 基于 Struts、XML 和 Ajax 整合框架的接处警信息管理系统设计及实现 [D/OL]. 东北大学, 2010-06 中国知网 <http://epub.cnki.net/>.

[4] 蔡沁舟. 政法信息共享平台公安局刑事案件协同系统设计与实现 [D/OL]. 电子科技大学, 2015-09, 中国知网 <http://epub.cnki.net/>.

4. 进度安排

	设计（论文）各阶段名称	起 止 日 期
1	通过查阅相关文献资料，熟悉当前管理平台的主要功能及管理模式，然后分析当前该系统的市场前景，确定系统整体业务逻辑及功能需求。	选题结束至第八学期开学
2	进一步了解项目实际需求基础及进行相关文献、技术研究基础上，撰写并提交开题报告	2017 年 3 月 2 日至 3 月 10 日
3	在进一步了解项目实际需求基础上，进行项目需求分析。确定系统的整体架构及解决方案。	2017 年 3 月 11 日至 3 月 29 日
4	基于概要模型设计的结果，分层（模块）依次实现系统功能，对各层（模块）进行测试并撰写过程文档。	2017 年 3 月 30 日至 5 月 3 日
5	按照软件工程技术方法与要求，和用户进行交流沟通，对系统进行测试，记录测试用例和测试结果。	2017 年 5 月 4 日至 5 月 17 日
6	按照软件工程系统开发的环节过程、校系对毕业设计（论文）的写作要求，完成论文写作。	2017 年 5 月 18 日至 5 月 31 日
7	论文定稿及答辩工作	2017 年 6 月 1 日至 6 月 22 日

审核人:_____ 年 月 日

派出所办案流程标准化智能管理平台的设计与实现

摘 要

根据山西瑞泽恒达公司提出的需求，通过研发山西公安派出所办案流程标准化智能管理平台（简称智能化派出所系统），实现了对特定警务作业全过程的标准化工作指导，实现对各类执法作业的信息化记录、流程化管理、全方位监督。

系统使用 B/S 与 C/S 相结合的模式，采用 MVC 模式的设计思想，满足其安全性与扩展性以及可维护性，运维成本低的要求，采用 JavaEE，Linux 等技术进行研发。

该系统作为核心业务系统，配合客户的其他系统软件，实现了办案流程的智能化，数据可视化以及数据的持久化管理。

系统在整体设计中，注重用户习惯，具有界面简洁大方，操作简单方便，管理自由快速，软件的可配置化，应用范围广等特点。

关键词：智能化派出所系统，警务作业标准化，JavaEE，Linux

The design and implementation of the standardized intelligent management platform of the police station

Abstract

According to the requirements of Shanxi Ruize Hengda Company, through the development of Shanxi police station handling process standardization intelligent management platform (referred to as intelligent police station system), to achieve the specific process of the specific police operations of the standardization of work guidance to achieve all kinds of law enforcement operations Of the information record, process management, all-round supervision.

The system uses the combination of B / S and C / S, and adopts the design idea of MVC model to meet the requirements of security and scalability and maintainability, operation and maintenance cost, and adopt JavaEE, Linux and so on.

The system as a core business system, with the customer's other system software, to achieve the handling of the process of intelligent, data visualization and data persistence management.

System in the overall design, pay attention to user habits, with a simple interface and generous, simple and convenient operation, free management fast, the software can be configured, the application of a wide range of features.

Keywords: Intelligent police station system, Standardization of police operations, JavaEE, Linux

目 录

1 前言.....	1
1.1 系统研究的目的及意义.....	1
1.2 国内外研究进展.....	1
1.3 系统的开发环境及工具.....	2
1.3 可行性分析.....	2
1.3.1 技术可行性分析.....	3
1.3.2 经济可行性分析.....	3
1.3.3 市场可行性分析.....	3
1.3.4 人员可行性分析.....	3
1.3.5 法律可行性分析.....	3
2 系统需求分析.....	4
2.1 系统需求分析.....	4
2.1.1 系统需求描述.....	4
2.1.2 系统功能分析建模.....	11
2.2 系统数据需求分析.....	16
2.2.1 系统数据信息描述.....	16
2.2.2 系统数据分析建模.....	16
3 系统总体设计.....	21
3.1 整体架构设计.....	22
3.1.1 系统体系结构.....	22
3.1.2 系统架构设计.....	22
3.1.3 智能化派出所系统软件架构.....	24
3.1.4 系统架构实现.....	26
3.1.5 系统功能设计.....	27
3.2 系统数据库设计.....	28
4 系统详细设计及编码实现.....	34
4.1 智能化派出所系统详细设计.....	34
4.1.1 实体类.....	34
4.1.2 业务逻辑层.....	34
4.1.3 数据访问层.....	35
4.1.4 表现层.....	36
4.2 智能化终端程序.....	40
4.2.1 RFID 串口读取程序.....	40
4.2.2 web 端树莓派访问接口程序.....	41
5 软件测试.....	42
5.1 测试的目的及技术.....	43
5.2 测试的步骤.....	43
5.3 测试的过程.....	43
5.3.1 系统架构测试.....	43

5.3.2 WEB 端树莓派接口测试.....	44
5.3.3 系统主要业务测试.....	44
5.3.4 树莓派读取串口数据测试.....	45
5.4 测试结论.....	45
5.5 项目部署.....	45
6 结论.....	47
6.1 完成的功能模块.....	47
6.2 待优化的功能.....	47
参考文献.....	48
致谢.....	49

1 前言

1.1 系统研究的目的及意义

随着互联网技术的快速发展，物联网异军突起成为继计算机硬件、互联网软件之后信息产业的第三次发展浪潮，因此物联网在未来公安系统的工作中应用前景十分广阔。

近年来，随着国家公共安全、生产安全等问题的日益严峻，各种需求十分迫切。特别是在公安系统的相关部门，随着执法规范化建设的不断深入，对公安系统办案功能区等方面的执法硬件设施提出了更高、更精、更细的要求。然而市场上并没有与之相契合的完整的软硬件解决方案，因此通过研发山西公安派出所办案流程标准化智能管理平台（简称智能化派出所系统），实现了对特定警务作业全过程的标准化工作指导，并对进入办案区的犯罪嫌疑人进行符合国家法律的标准化提示提醒及各类权利义务告知。从而实现对各类执法作业的信息化记录、流程化管理、全方位监督。从而大幅度提高公检法机关人员的办案效率。

本课题采用目前最流行的 B/S 架构，结合现代互联网，数据库，嵌入式等技术实现办案流程的信息化管理，另外本课题研究了在国内流行的设计模式，开发框架等，在经过分析斟酌后，选择其中最适合本系统的，将其整合，实现本系统的开发，并为系统的后期维护和扩展提供强有力的技术保障。

1.2 国内外研究进展

近年来公安工作信息化工作有了实质性进展。且随着金盾工程的进一步推广深入，各级公安机关纷纷建立起自己的网站，设立网上办事窗口，公安信息化工作有了实质性的进展，整体处于一个快速发展的阶段。^[6]

但是，我国仍然是一个发展中国家，在公安工作信息化建设方面的起步较晚，^[6]所以办案系统智能化方面就比较落后。公安系统办案功能区管理的智能化，可以提高公安干警的办案效率，为公安干警与相关业务使用者提供极大的便利，然而当前市场上并没与这种需求相契合的软硬件解决方案。

在上述情况下，本课题提出一套完整的智能化派出所系统软硬件的解决方案。本

课题采用树莓派做数据采集终端，采用 C/S 架构的 WEB 系统做整个解决方案的控制中心，视频录播系统做视频监控，从而实现了对各类干警执法作业的信息化记录、流程化管理以及全方位的监督。

1.3 系统的开发环境及工具

（1）系统开发的环境

硬件：RFID 读卡器、身份证阅读器、树莓派 3 代、Windows 服务器、录播系统服务器。

软件：Windows 10 、raspberrypi（Linux） 、Java8 、Eclipse 的 Mars 版本 、MySQL 5.6 、Tomcat 7.0。

（2）系统开发的工具

系统采用 Eclipse Mars、MySQL 5.6、Tomcat 7.0。

下面我们来对开发中所采用的工具进行说明：

① Eclipse Mars 简介

Eclipse 是开源的 IDE，它是基于 Java 开发的可扩展开发的平台。Eclipse 的设计思想是：除了核心功能外，其他附加功能都是插件。Eclipse 的软件核心很小，其它所有附加功能都是以插件的形式安装在 Eclipse 软件核心上。这款 IDE 能够很好的支持 MySQL、Tomcat 等软件，很适合本系统的开发使用。

② MySQL 数据库

MySQL 是一个开放源码的关系型数据库管理系统，由瑞典的一家公司主持开发的。目前 MySQL 属于 Oracle 旗下一个开源项目。因 MySQL 体积小、速度快、引擎可热插拔等特点，因此中小型网站都选择 MySQL 作为网站数据管理系统。

③ Tomcat 7.0

Tomcat 是 Apache 是开源组织维护的一个开源项目，由 Apache、Sun 和其开源公司以及一些开源爱好者共同开发而成。Tomcat 服务器是开放源代码的 Servlet 容器，属于轻量级 Web 应用服务器。因其开源、支持高并发等特点很适合于本系统的开发使用。

1.3 可行性分析

1.3.1 技术可行性分析

智能化派出所系统为 B/S 与 C/S 相结合的模式，整合山西瑞泽恒达公司在之前项目中已有的软硬件资源，包括物控子系统，第三方集成子系统等资源，系统采用目前主流的 Java 技术。本项目使用 SSH（Spring，Spring MVC，Hibernate）三大框架进行项目的开发，便于系统的扩展，维护，并且可以一键部署，符合主流系统设计。

1.3.2 经济可行性分析

因为本系统的开发所涉及到的硬件环境都是客户公司之前已有的环境，不需要另外采购新的硬件环境。系统开发时只需要在开发人员的计算机上安装 Eclipse，Tomcat，MySQL 等开发工具即可。系统开发工程中只需要少量的 RFID 读卡设备等硬件环境，没有其他大型设备的采购花销。

1.3.3 市场可行性分析

本系统是应客户公司在做其他项目过程中遇到的问题，结合自身已有的软硬件资源以及技术优势，提出项目需求，而进行开发的一个自研项目，符合市场运行规律。

1.3.4 人员可行性分析

本系统由移动互联实验室 3 人开发小组进行研发，开发人员基础扎实，项目分工明确，具备开发能力，因此开发可行。

1.3.5 法律可行性分析

根据山西瑞泽恒达公司提出的需求，与指导老师协商后，在法律允许的范围内，开发完成的。本系统需求及相关设计是在指导老师跟客户公司的共同协商，指导老师的指导下由本团队独立开发完成的，不会出现侵权问题。

2 系统需求分析

2.1 系统需求分析

2.1.1 系统需求描述

根据与山西瑞泽恒达公司的多次电话会议以及当面的会议讨论，在老师的指导下完成项目的需求的初步敲定。

(1) 系统构成

瑞泽智慧警务系统主体采用结构化设计，各子系统也要求采用 SOA 结构化设计，便于系统后期升级及功能扩展。

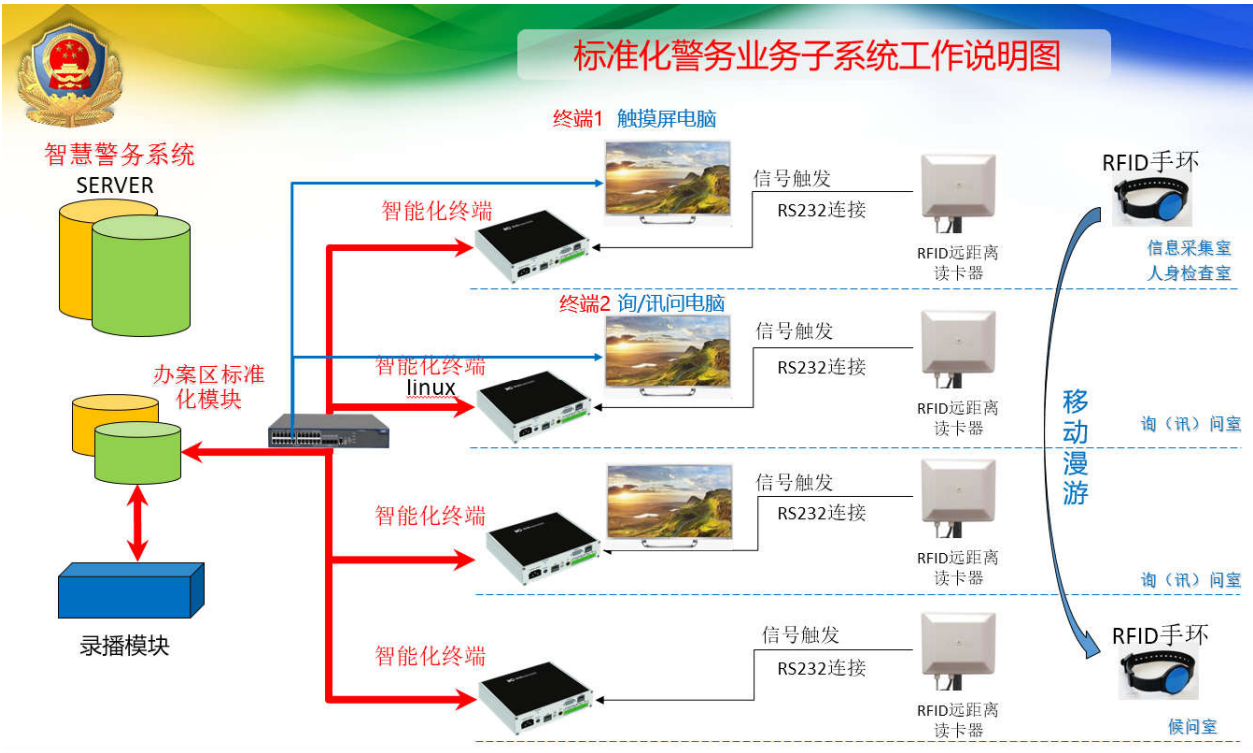


图 2.1 工作说明图

如图 2.1 所示，标准化警务业务子系统包括：SERVER 端、终端 1（触屏电脑）、终端 2（询/讯问电脑）、RFID 涉案人员登记管理设备及第三方录播模块这几个部分。系统各部采用星型网络连接，其中 RFID/ID 身份采集模块需要以 RS232 串口或 USB 接口的模式连接“智慧警务物联终端”实现数据中转。

系统工作时遵循涉案人员入区的标准要求，即涉案人员在入区时做手环（RFID / ID）登记，登记后即按照标准作业流程在相应的业务房间进行人身检查、信息采集，在此两项工作完成后，进入办案问询环节，直至达到出区条件。按照办案区管理规定办理出区和离区手续，出区状态意味着涉案人员还在羁押状态，档案不能闭环。离区状态作为涉案人员入区档案的闭环标记。

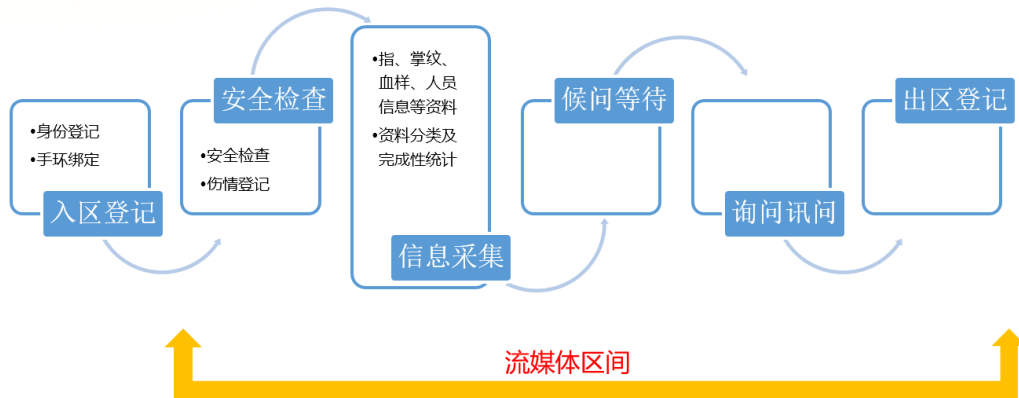


图 2.2 入区标准流程

（2）系统要求

① 支持对不同规模派出所的场景初始化管理，即对不同派出所的场所（功能房间，例如人身安全检查室、信息采集室）、场所内业务子系统的组件设备（RFID/ID 设备、触屏机等）进行定义和标识管理；

② 实现对派出所办案区涉案人员的入区档案信息化管理，要求支持派出所办案区实际作业需求，可同时对多人并发进行业务管理，支持完善的报表管理；

③ 实现对瑞泽智慧警务物控子系统（即物联网子系统）的数据对接功能，包括 RFID（或 ID）系统的报警信号对接；

④ 实现与第三方涉案人员录播子系统的数据对接功能，实现对涉案人员入出办案区的基于角色和时间的录像文档管理；

(3) 系统主要功能

基于上述系统结构及业务流程描述,系统主要分为基于 LINUX 的智能化终端(树莓派)端程序和智能化派出所系统程序两部。

① 智能化终端程序

需要对接已选型的 RFID 设备或 ID 卡,实现对 RFID 手环或 ID 卡的触发信号读取,并通过智能化终端做信号转发,由系统服务器集中对接第三方录播系统,实现监控录像的“启”、“停”控制及视频流文件管理。同时,需要考虑与平台物控子系统的对接需求,完成进入业务房间的语音提示等功能接口。

② 智能化派出所系统程序

系统基于 B/S 架构,对于涉案人员进入办案区各个功能房间的信息登记,需要侦查干警在该场所结束工作时做办案干警身份登记。除此之外,系统设备、信息汇总、用户管理等其他功能,则需用户登录进行身份认证后才能进行操作。下面是智能化派出所系统的主要功能。

- ✧ 涉案人员入区档案登记
- ✧ 办案区信息登记
- ✧ 采集设备管理
- ✧ 派出所及房间信息管理
- ✧ 用户信息管理
- ✧ 角色管理
- ✧ 系统日志管理
- ✧ 数据库管理
- ✧ 涉案信息查询汇总

(4) 模块构成

当前版本的标准化警务业务子系统主要围绕办案区设计功能,下面详细介绍各模块功能:

① 涉案人员信息采集单元

由二代身份证读取器, RFID / ID 制卡器、读卡器及 RFID 手环三个单元构成,完成身份识别和信息录入。

- ✧ 二代证读取器:用于入区登记管理,辨别二代证真伪,系统设计时需考虑涉案人员未携带身份证时的手工录入条件;

- ✧ 发卡单元：部署在“值班管理室”，用于对入区涉案人员进行身份登记及入区档案建立，同时将涉案人员身份和佩戴的 RFID 手环（ID 卡）进行绑定；
- ✧ 读卡器单元：部署在办案区多个功能房间门口，涉案人员进出对应功能房间时需要刷卡登记。值班室也需要配置读卡器，用于对涉案人员临时离区登记管理；
- ✧ RFID 手环：腕带式，涉案人员入区登记后，需佩戴个人专属的手环，办理离区登记后，手环与涉案人员身份解绑，可用于下一个入区涉案人员管理。

② 涉案人员信息显示输出单元

在办案区“人身安全检查室”及“信息采集室”内，壁挂安装有触摸屏一体机，要求系统实现以下功能：

- ✧ 当无涉案人员进入对应功能房间时，触屏一体机屏幕显示初始化的业务窗口，显示公安局的形象 LOGO；
- ✧ 涉案人员进入房间刷卡登记进入后，触屏一体机可自动显示涉案人员入区登记的身份信息及照片等资料；
- ✧ 除涉案人员身份信息外，触摸屏一体机还需显示对应功能房间（安全检查室、信息采集室）的信息登记表，以便干警做信息完整性登记采集；
- ✧ 涉案人员离开房间刷卡登记离开后，触屏一体机屏幕显示业务窗口恢复初始状态，不可继续显示涉案人员信息。

③ 智能化终端

RFID/ID 单元需要与智能化终端连接后，通过 HTTP 协议的方式连接本业务系统服务器。智能终端其负责接收读卡器的信号，将接收到的各房间 RFID 手环触发信号传输给后台服务器，同时还需实现和平台物控系统进行信号对接，按当前处理模式，将 RFID/ID 接收到的“登记进入信号”作为控制逻辑。执行当前文件夹下的 c.sh 文件，该文件为物控系统控制输出的对接文件。

④ 智能化派出所系统

当前版本智能化派出所系统主要完成办案区管理，系统设计需要注意以下几个原则：

- ✧ 涉案人员“入区~离区”的档案一致性及完整性：即信息化档案登记及录制音视频档案的一致性、完整性，因此从涉案人员入区开始到出区的行为过程，原则上都需要登记；
- ✧ 涉案人员入区数量存在不确定性：系统设计时需要考虑多涉案人员入区的管理问

题，尤其是多涉案人员入区时与音视频录播系统的对接关系问题；

- ✧ 系统处理的办案流程不能做硬编码设计：由于在实际办案中，标准状态是 1 人单独处理（因一个涉案人员必须配置 2 名干警看护），因此，在突发入区大批涉案人员的情况时，会先将大批人员简单安全检查后先关至候问室排队等候处理。
- ✧ 当前版本的系统设计的理念是疏导实现标准化，而不是强制必须标准化：在业务功能中，系统的总体目标是实现标准化警务登记管理，但在流程处置上，允许非标准化档案建立的情况。

(5) 系统业务流程

前文已经展示了标准流程，但在实际作业中，涉案人员的入区时间和人数并无绝对规律，因此，系统的业务流程需要能匹配实际的工作模式，即保证标准化的作业达成，又要满足多人并发情况下的科学可行，对现有业务时效不能产生负面影响。系统业务流程如图 2.3 所示：

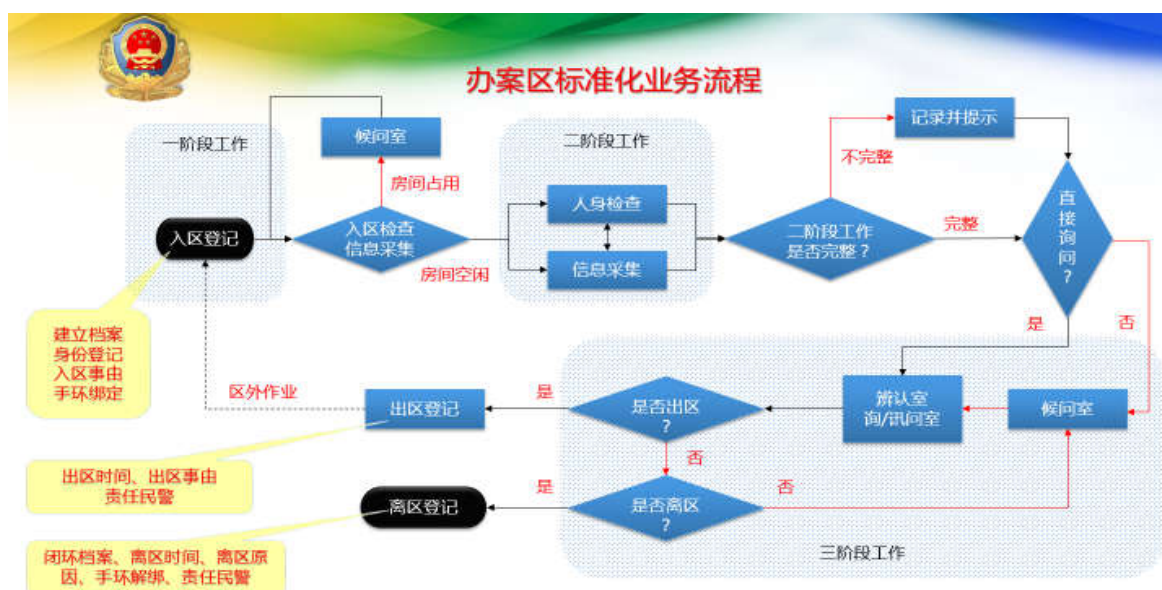


图 2.3 系统业务实际流程

按照图 2.3 实际作业模式，系统业务流程的各个环节描述如下：

① 涉案人员入区：首先在“入区登记”环节，建立入区涉案人员档案、根据当前时间自动生成该人员的档案编号、登记入区时间、入区事由、完成身份信息注册，并将 RFID 手环（ID 卡）和涉案人员身份进行绑定。身份信息注册可以通过二代身份证读卡设备完成，也可以通过手工录入完成；

② 安全检查及信息采集：入区登记完成后，涉案人员佩戴 RFID 手环进入办案区，需要首先进行第二阶段工作，根据业务房间空闲情况及涉案人员数量情况，干警会带领涉案人员分别进行“安全检查”、“信息采集”、“候问等待”等环节，上述各个环节过程即每个环节的“入门~出门”全程，要求对涉案人员进行全程录像；

③ 案件侦办：“安全检查”和“信息采集”两个环节完成后，涉案人员会进入案件侦办阶段，实际情况下，干警会根据办案需求，在“候问室”、“辨认室”、“询问、讯问室”之间多次带领涉案人员进行侦办审理。在这个环节中，涉案人员存在入区多次询问（讯问）的可能，因此对每一次进入“候问室”及询问（讯问），都需要确保进行录像。同“安全检查”和“信息采集”环节一样，系统设计为当涉案人员进入某个区域范围，采用 RFID 手环识读涉案人员“进入”和“离开”状态，达到对每次询问（讯问）的时间采集以及“启动”和“停止”该区域的监控录像。同时该区域的警务人员通过本系统将涉案人员在该区域中的必要信息进行采集并录入上报到系统数据库。

④ “临时出区登记”：在案件侦办过程中，有时需要带涉案人员出区作业，例如起赃、指认现场等。在系统中定义为“出区登记”，出区登记需要记录“出区时间、出区事由、返回时间、责任民警”等信息。

⑤ 离区登记：在涉案人员经历以上程序后，达到羁押释放条件的，会办理“离区登记”。在离区登记环节，同样需通过 RFID 手环触发监控录像停止。离区登记完成后，系统将给录播系统发送针对该涉案人员的视频打包结束标志，即可完成针对该涉案人员本次入区出区的视频文件生成。

(6) 对接系统说明

智慧警务业务系统需要与以下相关系统对接：

① 智慧警务物控系统

涉案人员进入办案区各功能房间时，需要做刷卡登记进入和离开，登记进入时，需要联动智慧警务物控系统的声音播报功能，对嫌疑人入区进行警示播报。

② 涉案财物保管柜系统

涉案人员入区后，进行人身安全检查时，需将随身财物及各类违禁物品保管存放至涉案财物保管柜，基于 ID 卡管理，系统后台支持数据库对接，需要采集其 ID 与嫌疑人及箱号的绑定关系、嫌疑人开闭箱的日志记录。

③ RFID/ID 系统

涉案人员进、出各功能房间的行为，靠 RFID/ID 刷卡登记实现标记管理。需要选型满足系统需求的系统，形成高效和稳定的管理目标；

④ FTP 服务器（存储）

录播系统在嫌疑人完成整个办案区的业务流程之后生成该嫌疑人的案件视频档案，需及时上传至 FTP 服务器，作为电子档案长期存储。重要案件及人员的视频档案将刻盘的方式另行保管。

⑤ 第三方录播系统

它是第三方提供的一个音视频文件的服务器，它提供录像的启动，停止，房间切换，在收到停止录像的指令后，将该嫌疑人在业务房间内所有视频文件按照一定格式生成该嫌疑人的案件视频档案进行存档。由于硬件条件的限制，录播系统当前最多支持 16 路录像，也就是最多可以同时进行 16 个涉案人员的业务过程。

⑥ 当前系统采用的硬件厂家型号等信息介绍

- ✧ 录播系统采用的是深圳捷视飞通科技有限公司定制的 MCV5000L 视频录播设备。
- ✧ 树莓派采用的是最新版的 Pi3，系统是树莓派的定制版 Linux 系统 raspberrypi。
- ✧ RFID 读卡器是一种能识别电子标签设备的识别设备，系统中采用的 RFID 是捷通公司提供的 JT-8280B 读写器，在系统中用于门禁刷卡识别嫌疑人信息。
- ✧ 本系统中采用的身份证阅读器是华视电子生产的 CVR-100U 居民身份证阅读器，将它与计算机通过 USB 口进行连接，可用于读取第二代居民身份证内相关信息的专用阅读机具。
- ✧ 本系统采用的网络摄像机是海康威视生产的 H265 300 万半球形网络摄像机。在内网配置 IP 用于业务房间内监控嫌疑人的网络设备。

2.1.2 系统功能分析建模

本系统采用 UML 时序图和用例图进行系统功能的建模。

(1) 业务流程时序图

① 时序图的介绍

时序图（Sequence Diagram），又名序列图，是一种 UML 建模的对象交互图。时序图通过描述各个对象之间发送消息的时间顺序来显示多个对象之间的动态协作与关系。

② 核心业务（安全检查、信息采集、候问、询问/讯问）时序图

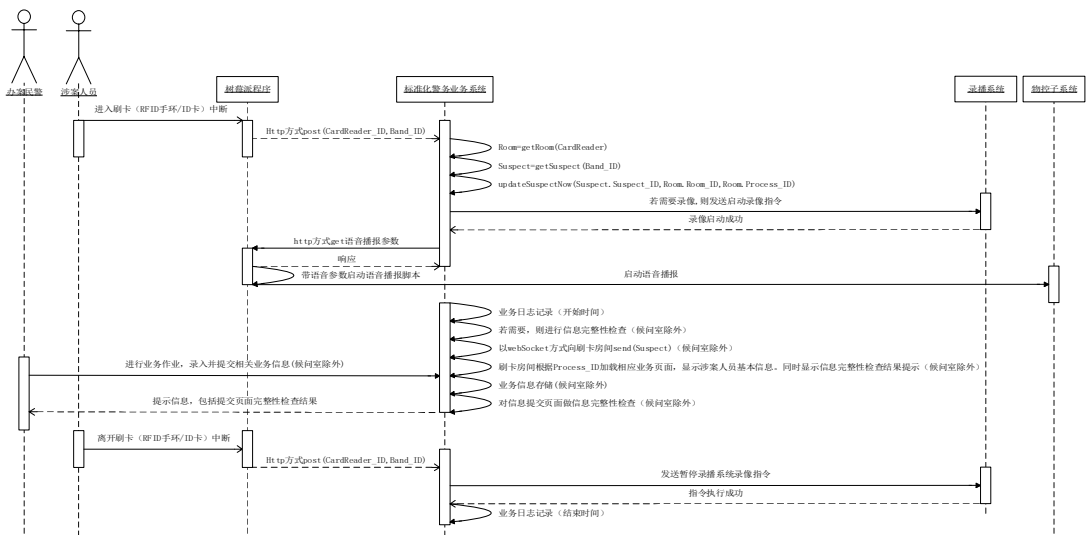


图 2.4 核心业务过程时序图

如图 2.4 所示，核心业务执行流程如下：

- ✧ 涉案人员佩戴手环进入某个房间后，通过刷卡进行身份验证。读卡器检测到涉案人员佩戴的手环后，以串口通信的方式将手环信息，通过树莓派程序以 http 方式将相关信息上传到服务端 web 应用。
- ✧ 服务端 web 应用程序接收到树莓派上传的手环编号和读卡器编号信息后，通过比对数据库的数据，获得当前的涉案人员信息。
- ✧ 对录播系统的启动控制，需根据涉案人员的录像状态标志，进行指令控制。
- ✧ 对物控子系统语音播报控制，树莓派程序根据 web 服务器返回的参数，并通过该参数调用语音播报脚本。

③ 入区登记时序图

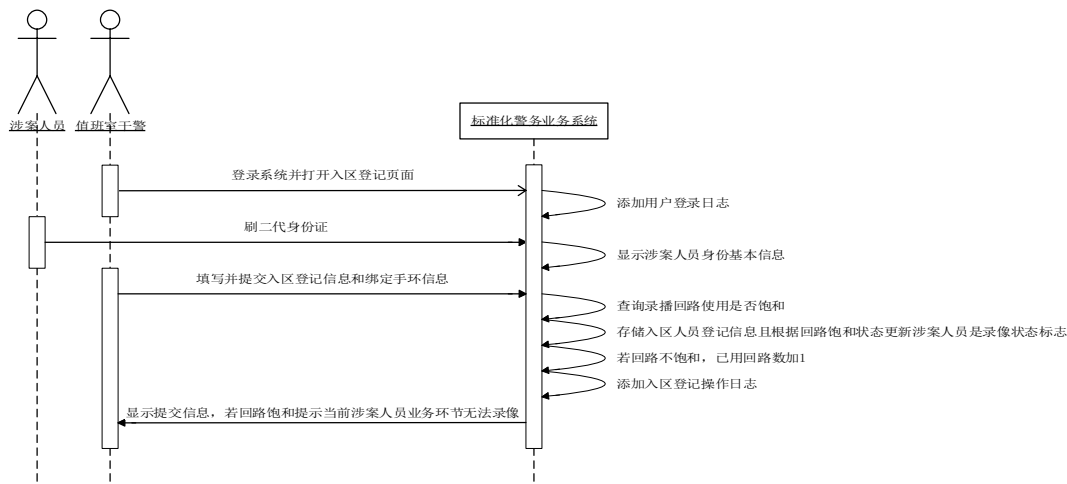


图 2.5 入区登记时序图

④ 离区时序图

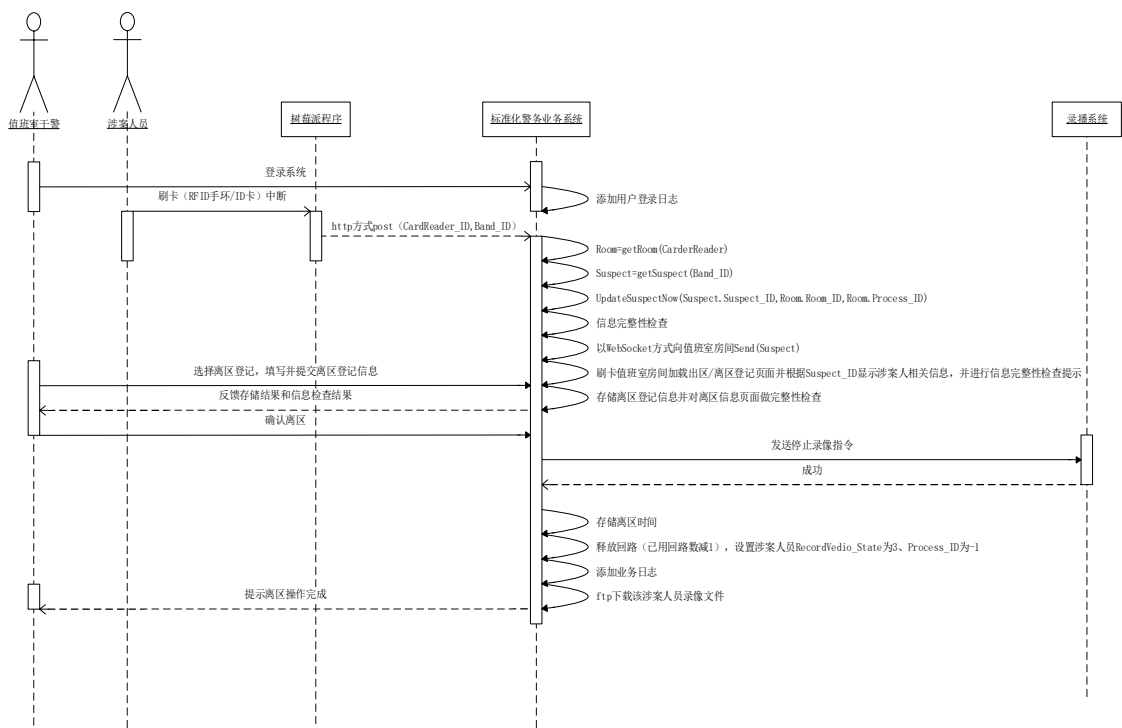


图 2.6 离区时序图

⑤ 出区时序图

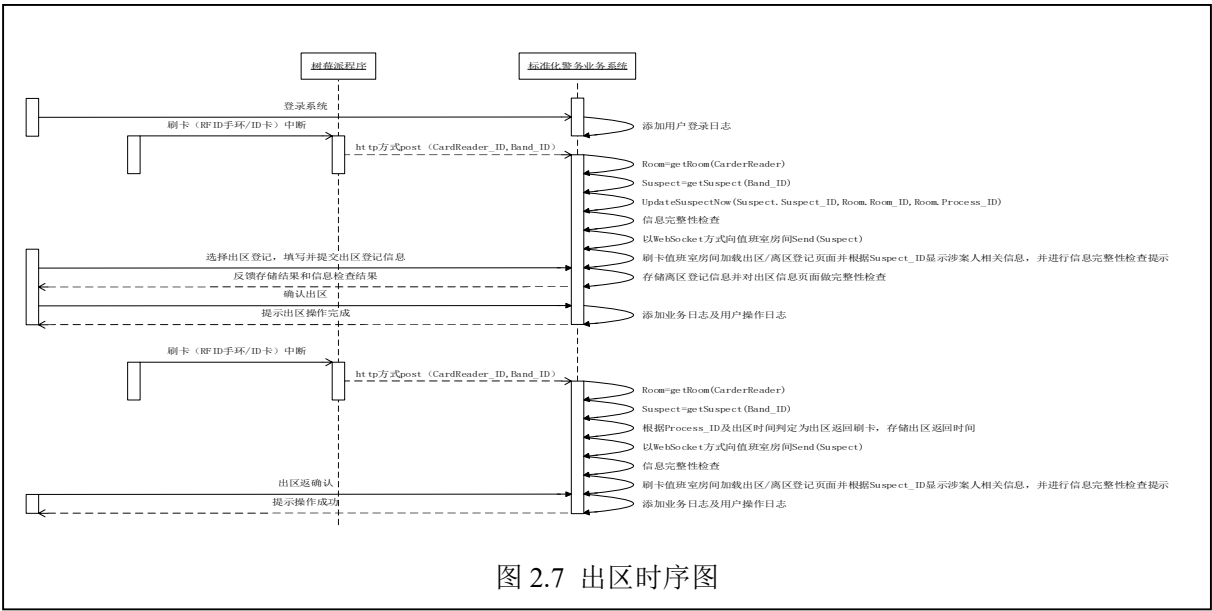


图 2.7 出区时序图

(2) 系统功能分析用例图

① 用例图的介绍

用例图即 Use Case 图, 是 UML 面向对象建模工具的一种常用方法。它用于定义系统的功能需求, 描述了系统的参与者与系统提供的用例之间的连接关系。这里的参与者可以是人, 也可以是另一个系统[2]。

② 智能化派出所系统整体功能模型

智能化派出所系统, 主要包含办案区业务管理、角色管理、嫌疑人信息管理等模块功能。如图 2.8 所示。

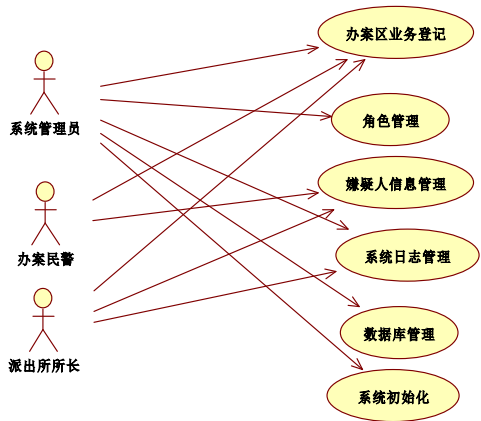


图 2.8 系统总体用例图

③ 智能化派出所系统功能模型

智能化派出所系统派出所干警用例模型具有登录, 办案区业务登记, 嫌疑人信息管理等功能, 派出所干警用例图, 如图 2.9 所示。

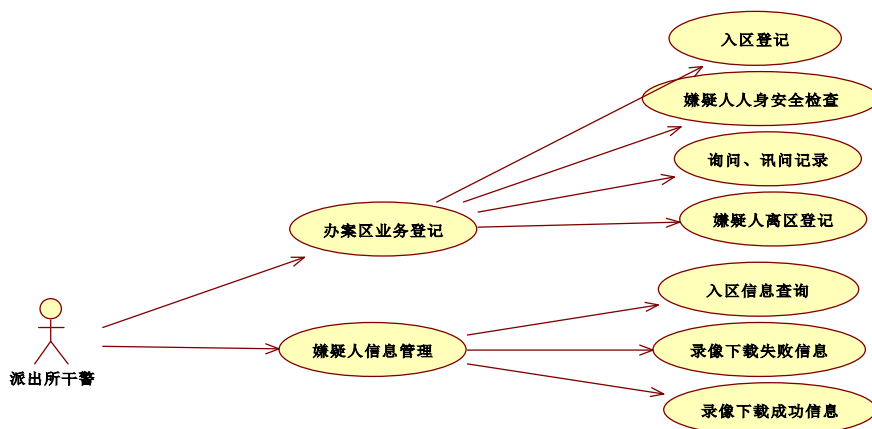


图 2.9 智能化派出所干警端用例图

- ✧ 智能化派出所系统派出所领导用例模型：具有登录，查看所有的嫌疑人档案，所有的统计信息等功能，派出所领导用例图，如图 2.10 所示。

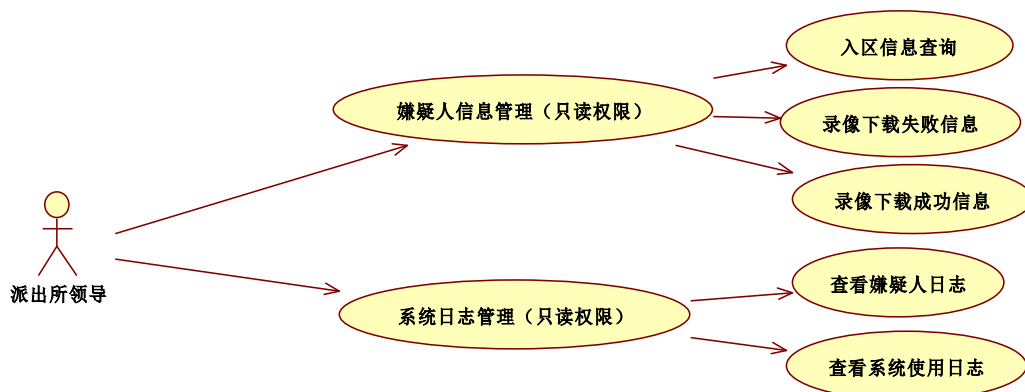


图 2.10 智能化派出所领导端用例图

- ✧ 智能化派出所系统超级管理员用例模型：它具有登录，角色管理，嫌疑人信息管理，系统日志信息管理等模块功能，如图 2.11。

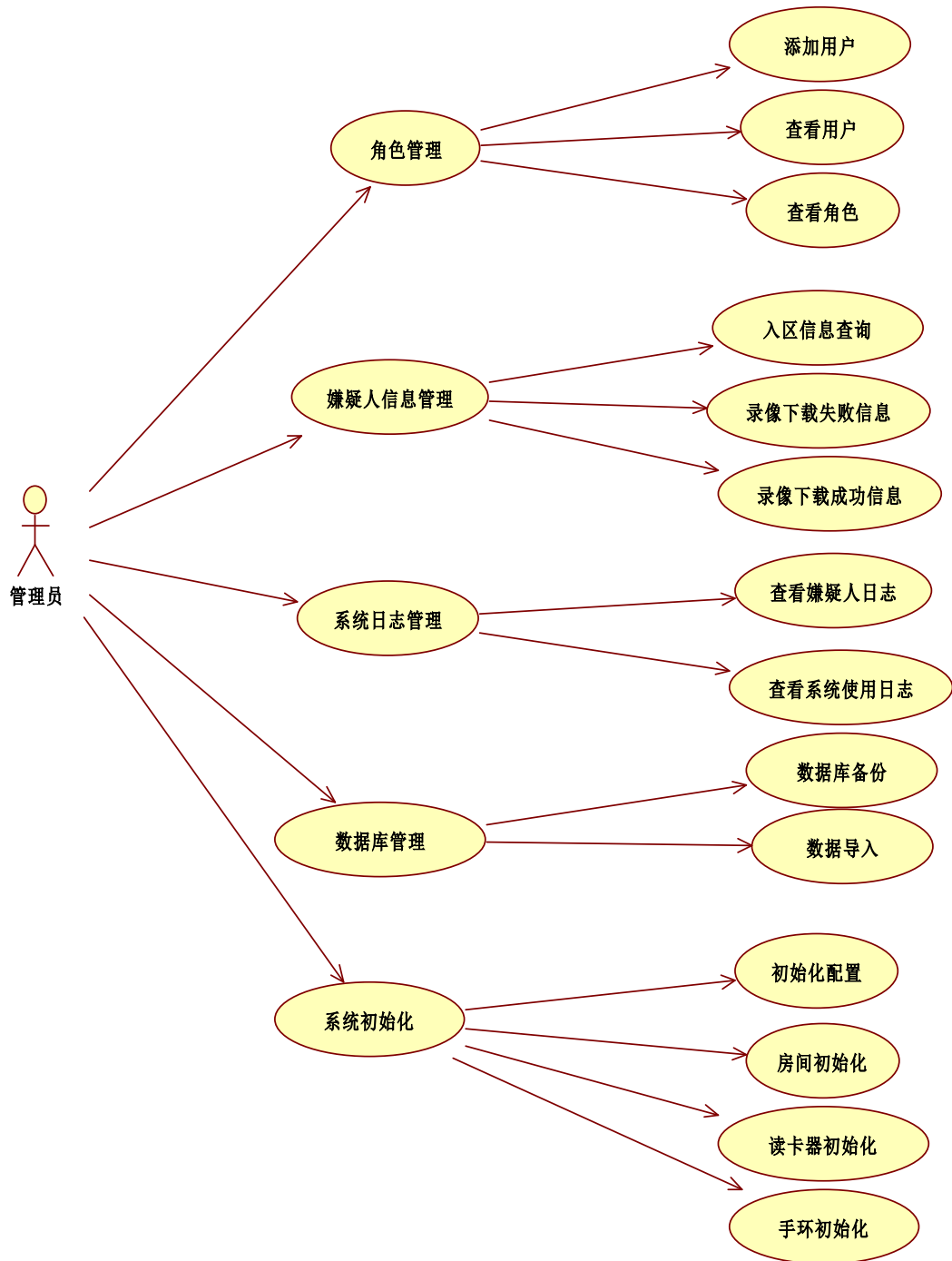


图 2.11 智能化派出所管理员端用例图

2.2 系统数据需求分析

2.2.1 系统数据信息描述

当前版本的标准化警务业务涉及以下标准化数据库及数据对象：

（1）当前版本的系统对象数据包括：各级干警信息、涉案人员信息、时间、功能房间、RFID/ID 身份标识、二代身份证读卡信息等；

（2）办案区警务标准化数据库包括：

- ① 涉案人员入区登记表
- ② 涉案人员人身安全检查表
- ③ 涉案人员财物保管登记
- ④ 涉案人员信息采集表
- ⑤ 涉案人员询问/讯问情况表
- ⑥ 涉案人员出区登记表
- ⑦ 涉案人员离区登记表
- ⑧ 涉案人员入区档案列表
- ⑨ 涉案人员入区档案报告
- ⑩ 统计报表（支持饼图、K 线图、堆积图、表格等常见报表格式）

2.2.2 系统数据分析建模

（1）数据结构模型（UML 类图）建立

在面向对象的处理中，类是其中最重要的构造块，类图处于核心地位。类图显示了一组类、接口、协作以及他们之间的关系。它提供了用于定义和使用对象的主要规则，同时，类图是正向工程（将模型转化为代码）的主要资源，是逆向工程（将代码转化为模型）的生成物。因此，类图是任何面向对象系统的核心，类图随之也成了最常用的 UML 图。^[2]

（2）系统中主要实体类的 UML

① 嫌疑人信息，业务流程关系，系统中涉及到嫌疑人的流程包括，入区登记，人身安全检查，嫌疑人询问/讯问记录，离区登记等流程，下面列出关于嫌疑人信

PHCSMP_Activity_Record
<ul style="list-style-type: none">serialVersionUID : longactivity_Record_ID : intsuspect_ID : Stringstart_Time : Stringend_Time : Stringroom_ID : intactivity_Record : Stringremark : Stringstaff_ID : Integertotal_record : intfill_record : int
<ul style="list-style-type: none">getActivity_Record_ID() : intsetActivity_Record_ID(in activity_Record_ID : int) : voidgetSuspect_ID() : StringsetSuspect_ID(in suspect_ID : String) : voidgetStart_Time() : StringsetStart_Time(in start_Time : String) : voidgetEnd_Time() : StringsetEnd_Time(in end_Time : String) : voidgetRoom_ID() : intsetRoom_ID(in room_ID : int) : voidgetActivity_Record() : StringsetActivity_Record(in activity_Record : String) : voidgetRemark() : StringsetRemark(in remark : String) : voidgetStaff_ID() : IntegersetStaff_ID(in staff_ID : Integer) : voidgetTotal_record() : intsetTotal_record(in total_record : int) : voidgetFill_record() : intsetFill_record(in fill_record : int) : voidtoString() : String

图 2.12 活动记录信息

PHCSMP_Suspect
<ul style="list-style-type: none">serialVersionUID : longid : intsuspect_ID : Stringband_ID : intsuspect_Name : Stringsex : Stringbirthday : Stringnation : Stringphone : Stringtype_ID : StringidentifyCard_Number : Stringsuspected_Cause : Stringenter_Time : Stringdetain_Time : Stringvedio_Number : Stringstaff_ID : Integeraddress : Stringnow_address : Stringtotal_record : intfill_record : intprocess_New : introom_Now : intidentityCard_Photo : Stringfrontal_Photo : StringsideWays_Photo : StringrecordVideo_State : intis_RecordVideo_Download : intcardReader_Switch : int

图 2.13 嫌疑人信息

PHCSMP_Line
<ul style="list-style-type: none">serialVersionUID : longline_Count : intline_Used : int
<ul style="list-style-type: none">getLine_Count() : intsetLine_Count(in line_Count : int) : voidgetLine_Used() : intsetLine_Used(in line_Used : int) : void

图 2.14 嫌疑人业务流程信息

息，以及嫌疑人涉及到的业务流程，如图 2.8 到 2.14 所示。

PHCSMP_BelongingS
<ul style="list-style-type: none"> serialVersionUID : long belongings_ID : int suspect_ID : String belonging_Number : String belonging_Name : String belonging_Character : String belonging_Count : int belonging_Unit : String keeping_ID : String cabinet_Number : String staff_ID : String staff_ID_Belonging : int room_ID : int total_record : int fill_record : int
<ul style="list-style-type: none"> getBelongings_ID() : int setBelongings_ID(in belongings_ID : int) : void getSuspect_ID() : String setSuspect_ID(in suspect_ID : String) : void getBelonging_Number() : String setBelonging_Number(in belonging_Number : String) : void getBelonging_Name() : String setBelonging_Name(in belonging_Name : String) : void getBelonging_Character() : String setBelonging_Character(in belonging_Character : String) : void getBelonging_Count() : int setBelonging_Count(in belonging_Count : int) : void getBelonging_Unit() : String setBelonging_Unit(in belonging_Unit : String) : void getKeeping_ID() : String setKeeping_ID(in keeping_ID : String) : void getCabinet_Number() : String setCabinet_Number(in cabinet_Number : String) : void getStaff_ID() : String setStaff_ID(in staff_ID : String) : void getStaff_ID_Belonging() : int setStaff_ID_Belonging(in staff_ID_Belonging : int) : void getRoom_ID() : int setRoom_ID(in room_ID : int) : void getTotal_record() : int setTotal_record(in total_record : int) : void getFill_record() : int setFill_record(in fill_record : int) : void toString() : String

图 2.15 随身物品检查信息

PHCSMP_Personal_Check
<ul style="list-style-type: none"> serialVersionUID : long check_ID : int suspect_ID : String self_ReportS : String checks_ReportS : String check_Situation : String is_Drink : String is_Diseases : String check_StartTime : String check_EndTime : String staff_ID : String room_ID : int total_record : int fill_record : int
<ul style="list-style-type: none"> getIs_Drink() : String setIs_Drink(in is_Drink : String) : void getIs_Diseases() : String setIs_Diseases(in is_Diseases : String) : void getCheck_StartTime() : String setCheck_StartTime(in check_StartTime : String) : void getCheck_EndTime() : String setCheck_EndTime(in check_EndTime : String) : void getCheck_ID() : int setCheck_ID(in check_ID : int) : void getSuspect_ID() : String setSuspect_ID(in suspect_ID : String) : void getSelf_ReportS() : String setSelf_ReportS(in self_ReportS : String) : void getCheck_ReportS() : String setCheck_ReportS(in check_ReportS : String) : void getCheck_Situation() : String setCheck_Situation(in check_Situation : String) : void getStaff_ID() : String setStaff_ID(in staff_ID : String) : void getRoom_ID() : int setRoom_ID(in room_ID : int) : void getTotal_record() : int setTotal_record(in total_record : int) : void getFill_record() : int setFill_record(in fill_record : int) : void toString() : String

图 2.16 人身检查信息

PHCSMP_Leave_Record
<ul style="list-style-type: none"> serialVersionUID : long leave_Record_ID : int suspect_ID : String leave_Time : String leave_Reason : String belongings_Treatment_Method : String belongings_Treatment_Record : String recipient_Person : String recipient_Person_Number : String treatment_Time : String staff_ID : Integer room_ID : int total_record : int fill_record : int
<ul style="list-style-type: none"> getLeave_Record_ID() : int setLeave_Record_ID(in leave_Record_ID : int) : void getSuspect_ID() : String setSuspect_ID(in suspect_ID : String) : void getLeave_Time() : String setLeave_Time(in leave_Time : String) : void getLeave_Reason() : String setLeave_Reason(in leave_Reason : String) : void getBelongings_Treatment_Method() : String setBelongings_Treatment_Method(in belongings_Treatment_Method : String) : void getBelongings_Treatment_Record() : String setBelongings_Treatment_Record(in belongings_Treatment_Record : String) : void getRecipient_Person() : String setRecipient_Person(in recipient_Person : String) : void getRecipient_Person_Number() : String setRecipient_Person_Number(in recipient_Person_Number : String) : void getTreatment_Time() : String setTreatment_Time(in treatment_Time : String) : void getStaff_ID() : Integer setStaff_ID(in staff_ID : Integer) : void getRoom_ID() : int setRoom_ID(in room_ID : int) : void getTotal_record() : int setTotal_record(in total_record : int) : void getFill_record() : int setFill_record(in fill_record : int) : void toString() : String

图 2.17 嫌疑人离区信息

Temporary_Leave
<ul style="list-style-type: none"> serialVersionUID : long temporary_Leave_Id : int suspect_ID : String templeave_Time : String templeave_Reason : String return_Time : String templeave_staff_ID : int return_staff_ID : int room_ID : int templeave_manager : int return_manager : int
<ul style="list-style-type: none"> <<create>> Temporary_Leave() getTemporary_Leave_Id() : int setTemporary_Leave_Id(in temporary_Leave_Id : int) : void getSuspect_ID() : String setSuspect_ID(in suspect_ID : String) : void getTempleave_Time() : String setTempleave_Time(in templeave_Time : String) : void getTempleave_Reason() : String setTempleave_Reason(in templeave_Reason : String) : void getReturn_Time() : String setReturn_Time(in return_Time : String) : void getRoom_ID() : int setRoom_ID(in room_ID : int) : void getTempleave_staff_ID() : int setTempleave_staff_ID(in templeave_staff_ID : int) : void getReturn_staff_ID() : int setReturn_staff_ID(in return_staff_ID : int) : void getTempleave_manager() : int setTempleave_manager(in templeave_manager : int) : void getReturn_manager() : int setReturn_manager(in return_manager : int) : void toString() : String

图 2.18 嫌疑人临时离区信息

② RFID 读卡设备信息关系实体的 UML。包括 RFID 手环，RFID 读卡器，如图 2.13,2.14 所示。

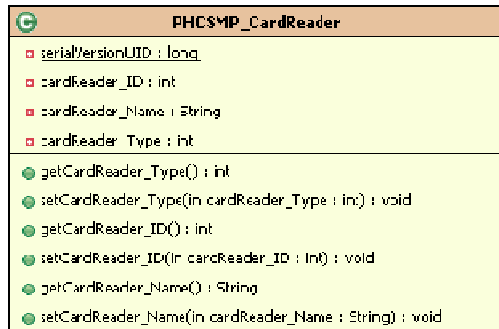


图 2.19 RFID 读卡器信息

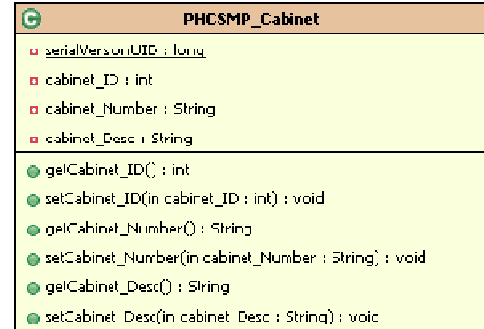


图 2.20 RFID 手环信息

③ 用户角色实体类信息实体的 UML。包括系统用户信息，角色信息，如图 2.15, 2.16 所示。

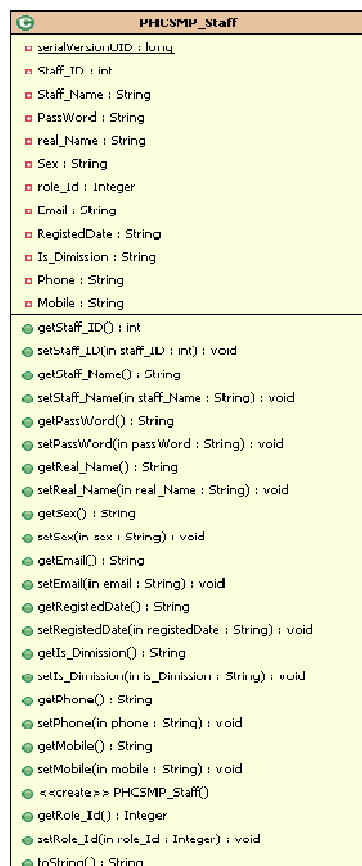


图 2.21 系统用户信息

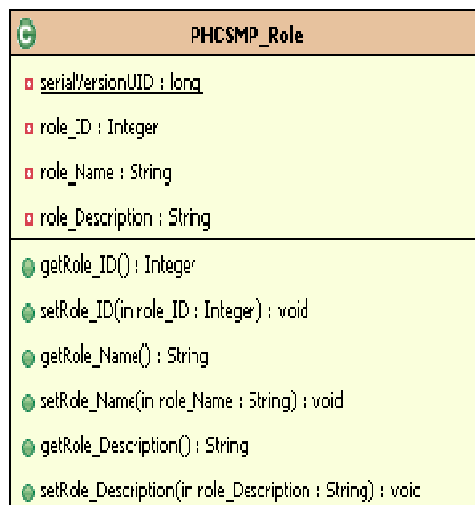


图 2.22 用户角色信息

④ 房间信息与系统日志信息实体的 UML，包括业务房间信息，系统操作日志信息，如图 2.19，2.20 所示。

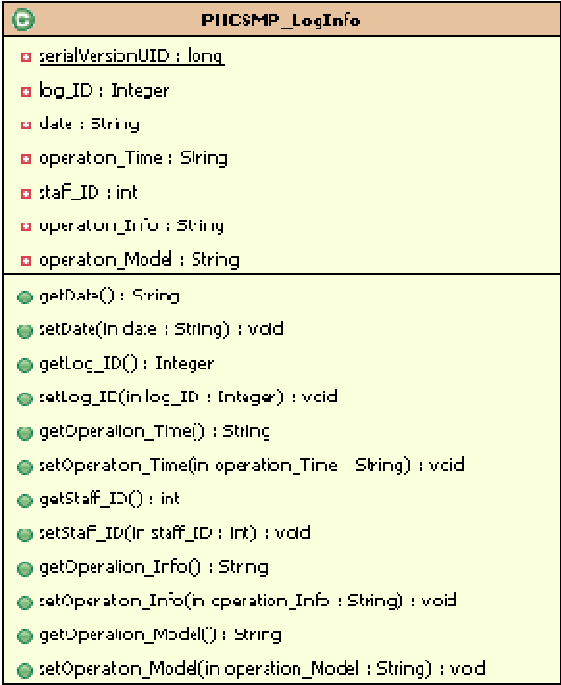


图 2.23 系统操作日志信息

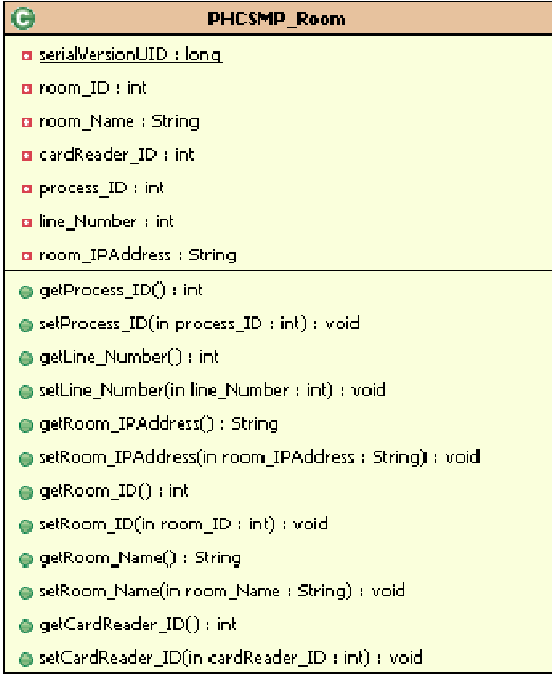


图 2.24 业务房间信息

3 系统总体设计

3.1 整体架构设计

3.1.1 系统体系结构

本系统采用 B/S 结构与 C/S 结构两种结构相结合的模式,来实现派出所干警办案流程智能化,标准化的需求。

通过开发山西公安派出所办案流程标准化智能管理平台(简称智能化派出所系统),可以实现对警务作业全过程的标准化工作指导,并对进入办案区的犯罪嫌疑人进行符合国家法律的标准化提示提醒及各类权利义务告知。

3.1.2 系统架构设计

智能化派出所系统分为智能化派出所系统 web 程序,数据采集端树莓派脚本两部分。下面来一一介绍。

(1) 系统总体架构

① 智能化派出所系统执行流程

本节介绍的是整个系统各个模块之间的数据交互方式与整个系统的执行流程,下面是整个系统的架构图,如图 3.1 所示

- ✧ 采集嫌疑人信息,绑定手环,建立 WebSocket 连接。通过二代身份证读卡器采集入区嫌疑人基本信息,并给嫌疑人绑定 RFID 手环,建立 WebSocket 连接,提交信息到数据库。
- ✧ 嫌疑人在相应的业务房间读取 RFID 手环,通过 HTTP POST 方式将手环信息发送到 web 服务器,触发 WebSocket 主动刷新页面,加载出手环对应的嫌疑人的信息,同时向录播服务器发送开始录像的指令,指令执行成功后反馈给树莓派 RFID 读取程序,触发树莓派上的 shell 脚本,开始播放音频信号。
- ✧ 嫌疑人在离区业务房间办理离区,解除手环绑定的嫌疑人,生成嫌疑人入区报告。
- ✧ 嫌疑人完成离区业务办理的时候,web 服务器通过 HTTP 协议向录播系统发送上传视频文件到 FTP 服务器的指令。
- ✧ 录播系统在收到 Web 服务器的上传文件的指令后,会将嫌疑人对应的业务房间的

视频文件拼接成一个视频文件，然后以 FTP 的方式将文件上传的 FTP 服务器。

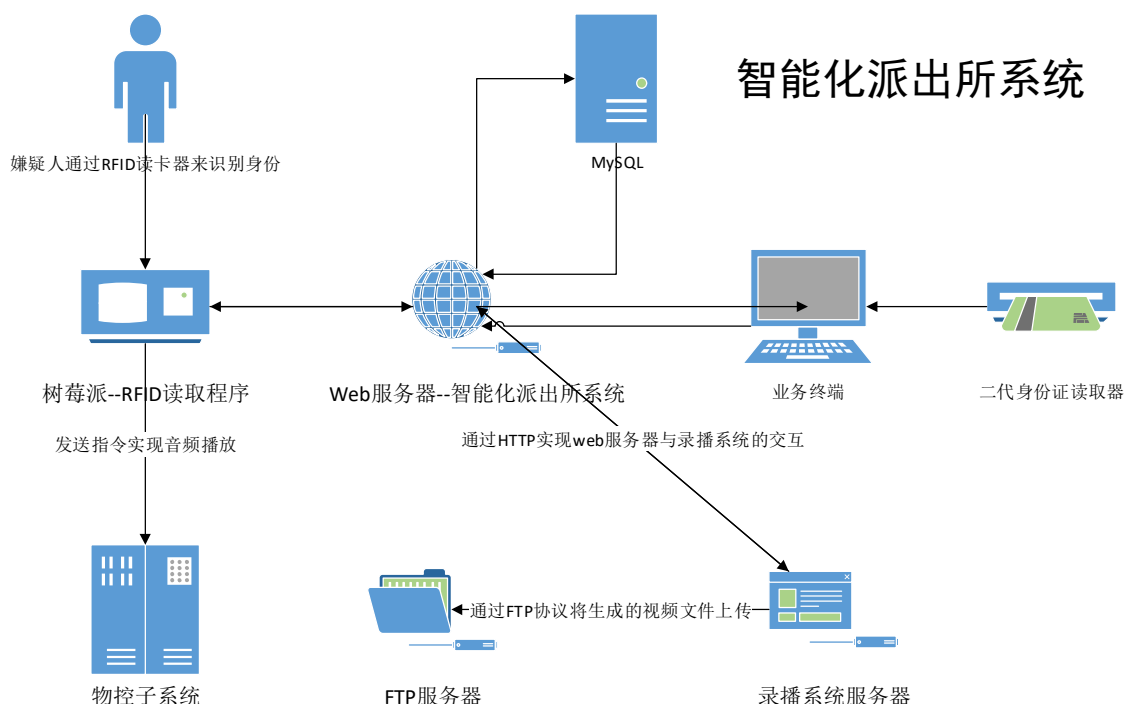


图 3.1 系统数据架构图

② 树莓派开发环境搭建

- ✧ 安装 jdk。树莓派的 jdk 是 arm 版本的，安装对应版本的 jdk 即可。
- ✧ 安装串口读取工具包 RXTX

- A. 进入 RXTX 解压包的目录执行 `./configure` 命令。
- B. 执行 `make` 命令。
- C. 执行 `make install` 命令，成功安装 RXTX 串口读取工具。

③ 发生的错误及解决方法

- ✧ 错误 1: `/rxtx-2.1-7r2/./src/I2CImp.c:135: error: 'UTS_RELEASE' undeclared (first use in this function)`。

解决办法:

- A. 使用命令 `uname -r` 获取当前系统的版本信息。
- B. 在系统根目录下，执行 `sudo nano /usr/include/linux/version.h`，添加 `#define UTS_RELEASE "当前系统的版本信息"`。

- ✧ 错误 2: `install: armv6l-unknown-linux-gnu/librxtxRS485.la' is not a directory`

解决办法: RXTX 解压包的 `configure` 文件中查找 `1.2*|1.3*|1.4*|1.5*`，修改为

1.2*|1.3*|1.4*|1.5*|1.6*|1.7*|1.8*。

④ 智能化派出所系统开发环境搭建

- ✧ 安装对应版本的 jdk。
- ✧ 安装 Eclipse。
- ✧ 安装 MySQL。
- ✧ 安装 maven。
- ✧ 安装 git。

3.1.3 智能化派出所系统软件架构

智能化派出所系统基本架构使用 MVC 和三层架构的设计模式。

(1) 三层架构

在软件工程系统架构设计中,为了实现软件架构的“高内聚低耦合”的思想,会将整个业务应用分为多层。三层架构一般指:表现层、业务逻辑层、数据访问层,如图 3.2 所示:

数据访问层: 主要功能是实现与数据库之间的数据交互。主要实现对数据的各种操作。

业务逻辑层: 位于表现层和数据访问层中间,起到一个承上启下的作用。实现系统主要的业务功能逻辑。

表现层: 主要是指与用户交互的具体页面。用于处理用户在页面输入的数据和在页面显示处理后用户需要的数据。

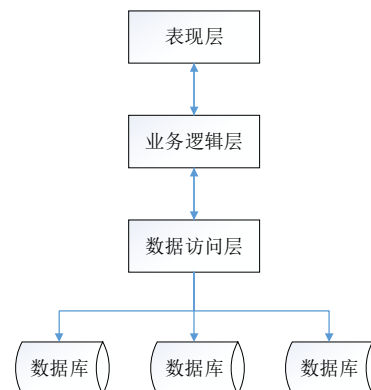


图 3.2 三层架构分层结构图

(2) 三层架构优点:

- ① 扩展性强。不同层之间不存在代码的耦合,负责各自的层面。
- ② 安全性高。用户端只能通过业务逻辑层来访问数据层,仅有唯一的入口点,把很多可能出现的危险的用户操作都屏蔽了。
- ③ 项目结构更清晰,开发人员职责更明确,有利于系统后期的维护和升级。

(2) MVC 模式

MVC 即,模型(model)、视图(view)、控制(controller)的简写。这种设计模式中,所谓模型,就是 MVC 需要提供的数据库,负责数据的访问和维护。所谓视图

图，就是用于显示模型中数据的用户界面。所谓控制器，就是用来处理用户的输入，负责改变模型的状态并选择适当的视图来显示模型的数据。[10]

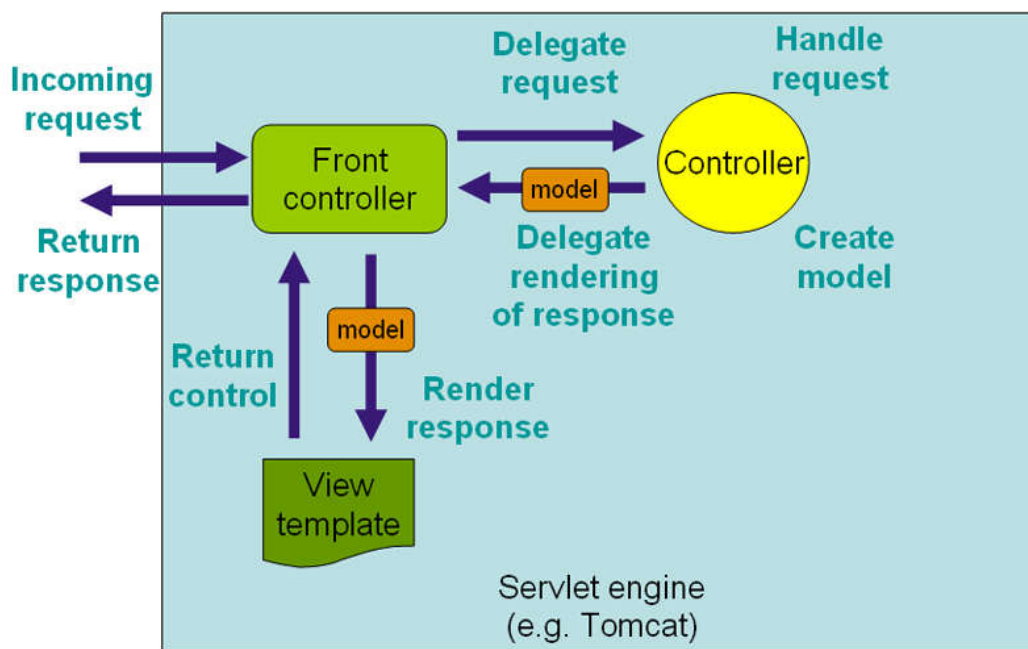


图 3.3 MVC 交互图

从图 3.3 可知，MVC 整个交互过程为：

- ① HTTP 请求：客户端请求提交，前置控制器 DispatcherServlet 拦截所有的 HTTP 请求。
- ② DispatcherServlet 寻找 Controller：由 DispatcherServlet 控制器查询一个或多个 HandlerMapping，找到处理请求所对应的 Controller。
- ③ 处理器 Controller 执行：DispatcherServlet 将请求提交到 Controller，Controller 调用业务逻辑处理后，返回具体的页面调用逻辑。
- ④ 处理视图映射并返回模型：DispatcherServlet 查询一个或多个视图解析器，找到指定的视图。
- ⑤ Http 响应：视图负责将结果渲染成 HTML 数据展现到客户端。

3.1.4 系统架构实现

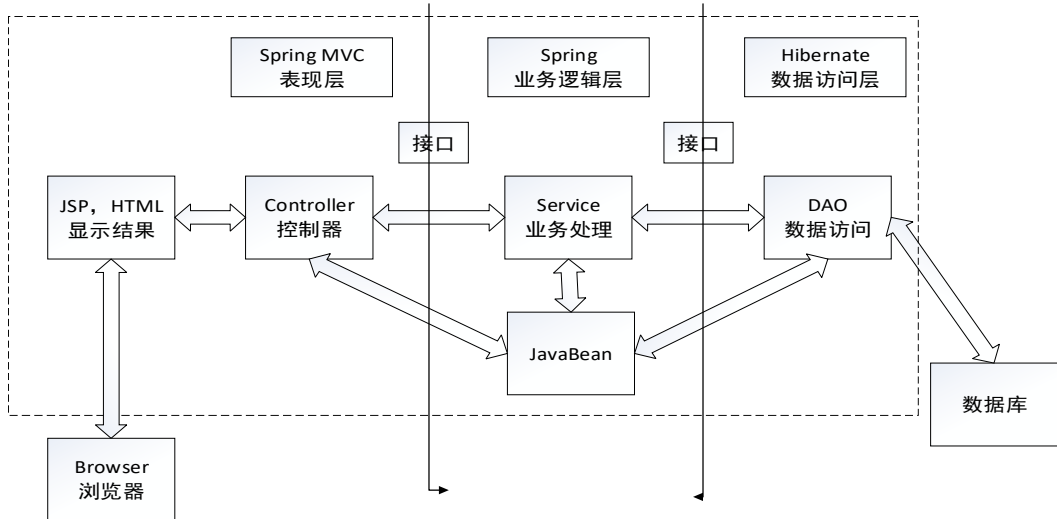


图 3.4 架构具体实现图

根据三层架构和 MVC 模式的软件架构设计思想，最终设计出项目整体的架构具体实现方案。其中表现层，业务逻辑层，数据访问层分别使用 Spring MVC，Spring，Hibernate 框架来具体实现；Model 即模型一般是指 JavaBean，Control 即控制指的是用于业务处理，控制页面跳转等的 Controller，而 View 代表视图用 jsp 和 html 实现，由上图的基本设计最终对项目进行了以下的架构设计，项目如图 3.4：



图 3.5 项目的程序包结构

com.haifeiWu.entity: JavaBean，也是 Java 实体类。

com.haifeiWu.action: 表现层的动作控制。

com.haifeiWu.base: 一些类的基础继承类。

com.haifeiWu.dao: Dao 接口，数据访问层的接口定义。

com.haifeiWu.daoImpl: Dao 实现，数据访问层的接口实现。

com.haifeiWu.service: Service 接口，业务逻辑层的接口定义。

com.haifeiWu.serviceImpl: Service 实现，业务逻辑层的接口实现。

com.haifeiWu.utils: 工具包。

3.1.5 系统功能设计

系统主要分为基于 LINUX 的智能化终端（树莓派）端程序和智能化派出所系统程序两部分。

（1）智能化终端程序

对接已选型的 RFID 设备，实现对 RFID 手环的触发信号读取，并通过智能化终端做信号转发，由智能化派出所系统集中对接第三方录播系统，实现监控录像的“启”、“停”控制及视频流文件管理。另外，完成与平台物控子系统的对接需求，完成嫌疑人进入业务房间的语音提示等功能。

（2）智能化派出所系统

系统基于 B/S 架构，对于涉案人员进入办案区各个功能房间的信息登记，需要侦办干警在该场所结束工作时做办案干警身份登记。除此之外，系统设备、信息汇总、用户管理等其他功能，则需用户登录进行身份认证后才能进行操作。

（3）智能化派出所系统的功能模块结构图：

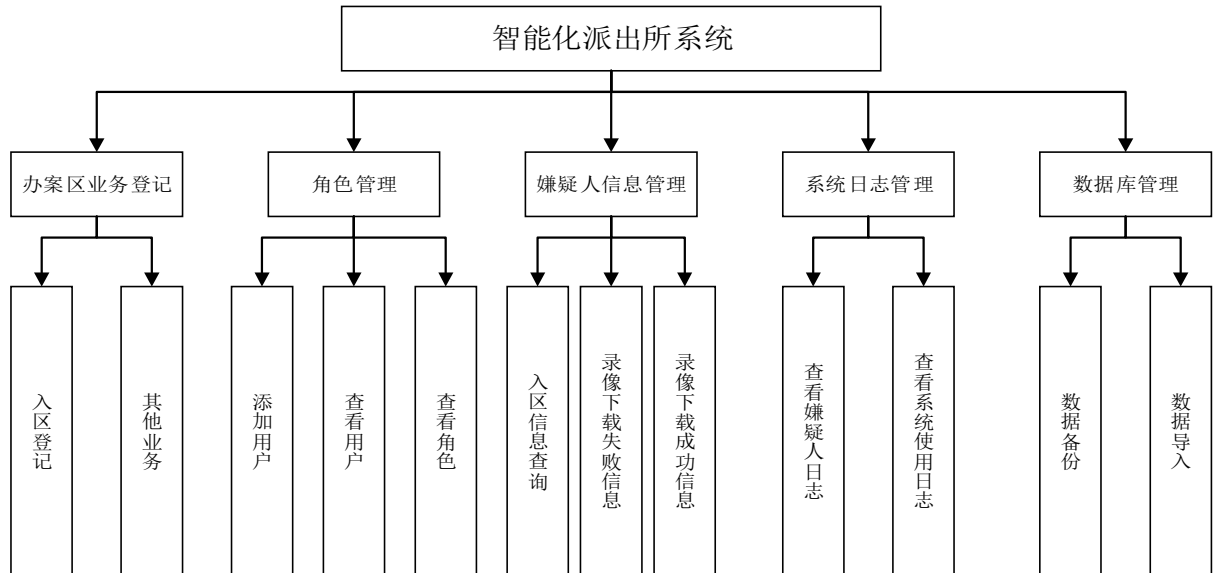


图 3.6 智能化派出所系统的功能模块结构图

3.2 系统数据库设计

数据库的逻辑设计是数据表结构和数据表关系的设计，下面列出智能化派出所系统数据库表的设计。

表 3.2.1 PHCSMP_Staff: 用户信息表

字段名	类型	长度	允许空	备注
Staff_ID	Int(自动加一)	4	No	用户 ID (主键)
Staff_Name	Varchar	20	No	用户名 (默认工号)
PassWord	varchar	20	Yes	密码
Real_Name	varchar	12	Yes	真实姓名
Sex	Bit	1	Yes	性别
Duties_ID	int	4	Yes	职务
Email	Varchar	100	Yes	邮箱
RegisteredDate	Datetime	8	Yes	注册日期
Is_Dimission	Bit	1	Yes	是否离职
Phone	Varchar	50	Yes	固定电话
Mobile	Varchar	50	Yes	移动电话
Is_FirstLogin	bit	1	No (默认 1)	是否第一次登录

表 3.2.2 PHCSMP_Duties: 职务信息表

字段名	类型	长度	允许空	备注
Duties_ID	Int(自动加一)	4	No	职务 ID (主键)
Duties_Name	varchar	50	No	职务名称
Duties_Desc	varchar	500	Yes	职务简介

表 3.2.3 PHCSMP_Role: 角色信息表 (管理员、办案民警等)

字段名	类型	长度	允许空	备注
Role_ID	Int(自动加一)	4	No	角色 ID (主键)
Role_Name	Varchar	20	No	角色名称名称
Role_Description	Varchar	200	Yes	角色说明
Is_Default_Role	bit	1	No	是否系统默认角色

表 3.2.4 PHCSMP_Staff_In_Role: 用户所属角色表

字段名	类型	长度	允许空	备注
Staff_ID	Int	4	No	用户 ID
Role_ID	Int	4	No	所属角色 ID

表 3.2.5 PHCSMP_Dic_Action_Cause: 案由字典表 (投案自首、治安传唤、刑事传唤等)

字段名	类型	长度	允许空	备注
-----	----	----	-----	----

Cause_ID	Int(自动加一)	4	No	案由 ID（主键）
Cause_Name	varchar	30	No	案由名称

表 3.2.6 PHCSMP_Dic_Inspection_Situation: 人身检查记录字典表（体表有伤痕、有饮酒）

字段名	类型	长度	允许空	备注
Situation_ID	Int(自动加一)	4	No	检查情况 ID（主键）
Situation_Name	Varchar	20	No	检查情况选项

表 3.2.7 PHCSMP_Dic_Keeping_Way: 随身物品保管措施字典表（扣押、暂存、代保管）

字段名	类型	长度	允许空	备注
Keeping_ID	Int（自动加一）	4	No	保管方式 ID（主键）
Keeping_Name	Varchar	20	No	保管方式选项

表 3.2.8 PHCSMP_Dic_Collection_Item: 采集项目字典表（身份信息、指纹、血液等）

字段名	类型	长度	允许空	备注
Item_ID	Int(自动加一)	4	No	采集项目 ID（主键）
Item_Name	Varchar	20	No	采集项目

表 3.2.9 PHCSMP_Dic_Leaving_Reason: 离开原因字典表（查证结束、刑拘、行政拘留等）

字段名	类型	长度	允许空	备注
Leaving_ID	Int(自动加一)	4	No	离开原因 ID（主键）
Leaving_Name	Varchar	20	No	离开原因

表 3.2.10 PHCSMP_Dic_Treatment_Method 随身物品处理方式字典表

字段名	类型	长度	允许空	备注
Leaving_ID	Int(自动加一)	4	No	离开原因 ID（主键）
Leaving_Name	Varchar	20	No	离开原因

表 3.2.11 PHCSMP_Dic_IdentifyCard_Type: 身份证件种类字典表（身份证、军官证等）

字段名	类型	长度	允许空	备注
Type_ID	Int(自动加一)	4	No	身份证件种类 ID（主键）
Type_Name	Varchar	20	No	身份证件种类

表 3.2.12 PHCSMP_Suspect: 进入人员基本信息表

字段名	类型	长度	允许空	备注
Suspect_ID	Varchar	10	No	档案编号（主键）
Band_ID	Int	4	Yes	手环 ID
Suspect_Name	Varchar	12	No	姓名
Sex	Bit	1	Yes	性别
Birthday	datetime	8	Yes	出生日期
Phone	Varchar	20	Yes	联系方式

Type_ID	int	4	Yes	身份证件种类
IdentifyCard_Number	varchar	50	Yes	身份证件号码
Suspected_Cause	varchar	20	Yes	进入办案区原因（案由字典表中 ID 用逗号分隔）
Enter_Time	datetime	8	Yes	进入办案区时间
Staff_ID	int	4	Yes	办案民警
Process_Now	Int	4	No	当前所处的流程状态，对应 PHCSMP_Dic_Process 表中的 Process_ID, 整个流程结束后置为-1,表示该涉案人员在办案区完整流程已经走完。
Room_Now	int	4	Yes	当前涉案人员所处的房间编号，对应 PHCSMP_Room 表中的 Room_ID
Total_record	Int	4	default	需要填写的总记录数
Fill_record	Int	4	yes	当前填写的记录数
IdentityCard_Photo	Blob		Yes	身份证照片 base64 编码，照片最大 65K
Frontal_Photo	Varchar	20	Yes	正面照图像文件地址
SideWays_Photo	Varhar	20	Yes	侧面照图像文件地址
RecordVedio_State	int	4	Yes	录像状态，0：不进行录像 1：录像还未启动 2：录像已经启动 3：录像结束
Is_RecoedVedio_DownLoad	bool	1	No	录像文件下载是否成功，默认 false

表 3.2.13 PHCSMP_Personal_Check: 进入人员人身检查信息表

字段名	类型	长度	允许空	备注
Check_ID	int	4	No	安全检查信息 ID（主键）
Suspect_ID	Varchar	10	No	档案编号
Self_ReportS	varchar	200	Yes	自述症状
Check_ReportS	Varchar	200	Yes	检查情况
Check_Situation	varchar	420	Yes	人身检查状态（人身检查记录字典表中 ID 逗号分隔）
Staff_ID	int	4	Yes	办案民警
Room_ID	Int	4	Yes	信息登记房间
Total_record	Int	4	default	需要填写的总记录数
Fill_record	Int	4	yes	当前填写的记录数

表 3.2.14 PHCSMP_BelongingS: 进入人员随身财物检查记录表

字段名	类型	长度	允许空	备注
Suspect_ID	Varchar	10	No	档案编号
Belonging_Number	varchar	10	No	物品编号
Belonging_Name	varchar	50	No	物品名称
Belonging_Character	varchar	50	Yes	物品特征
Belonging_Count	int	4	No	物品数量

Belonging_Unit	varchar	10	No	物品单位
Keeping_ID	Int	4	No	保管措施（随身物品保管措施字典表中 ID 逗号隔开）
Cabinet_Number	Varchar	10	No	保管柜编号
Staff_ID	int	4	Yes	办案民警
Staff_ID_Belonging	int	4	Yes	随身财物管理民警
Room_ID	Int	4	Yes	信息登记房间
Total_record	Int	4	default	需要填写的总记录数
Fill_record	Int	4	yes	当前填写的记录数

表 3.2.15 PHCSMP_Cabinet: 保管柜信息表

字段名	类型	长度	允许空	备注
Cabinet_Number	Varchar	10	No	保管柜编号（主键）
Cabinet_Desc	Varchar	20	No	保管柜描述

表 3.2.16 PHCSMP_Information_Collection: 进入人员信息采集表

字段名	类型	长度	允许空	备注
Suspect_ID	Varchar	10	No	档案编号
Is_Collected	bit	1	No	是否进行采集
Collected_Item	varchar	30	Yes	采集项目（采集项目字典表中 ID 逗号隔开，只有 Is_Collected 为真时有效）
Is_Stored	bit	1	No	是否入库
Is_Checked	bit	1	No	是否检查对比
Staff_ID	int	4	Yes	办案民警
Room_ID	Int	4	Yes	信息登记房间
Total_record	Int	4	default	需要填写的总记录数
Fill_record	Int	4	yes	当前填写的记录数

3.2.17 PHCSMP_Activity_Record: 询问、讯问等活动记录表

字段名	类型	长度	允许空	备注
Suspect_ID	Varchar	10	No	档案编号
Start_Time	datetime	8	No	开始时间
End_Time	datetime	8	No	结束时间
Room_ID	Int	4	No	活动房间
Activity_Record	varchar	200	No	活动内容
Vedio_Number	varchar	100	No	音频视频编码
Remark	Varchar	300	Yes	备注
StaffS	Varchar	20	Yes	办案民警 ID（多个民警 ID 间逗号间隔）
Room_ID	Int	4	Yes	信息登记房间
Total_record	Int	4	default	需要填写的总记录数
Fill_record	Int	4	yes	当前填写的记录数

表 3.2.18 PHCSMP_Dic_Process:办案区流程字典表（入区登记：0、安全检查：1、信息采集：2、讯问/询问：3、候问：4、出区/离区登记）

字段名	类型	长度	允许空	备注
Process_ID	Int	4	No	流程 ID（主键）
Process_Name	Varchar	20	Yes	流程名称

表 3.2.19 PHCSMP_Room: 房间信息表

字段名	类型	长度	允许空	备注
Room_ID	Int	4	No	房间 ID（主键）
Room_Name	Varchar	20	Yes	房间名称
CardReader_ID	Int	4	Yes	设备 ID
Process_ID	Int	4	Yes	房间功能
Line_Number	int	4	Yes	连接录播系统线路编号 (1-Line_Count)
Room_IPAddress	varchar	20	Yes	房间 IP 地址及通讯端口

表 3.2.20 PHCSMP_Leave_Record: 离开办案区登记表

字段名	类型	长度	允许空	备注
Suspect_ID	Varchar	10	No	档案编号
TempLeave_Time	datetime	8	Yes	最近一次临时离开时间
TempLeave_Reason	varchar	100	Yes	最近一次临时离开原因
Return_Time	datetime	8	Yes	最近一次返回时间
Leave_Time	datetime	8	Yes	最终离开时间
Leave_Reason	varchar	50	Yes	离开原因(离开原因字典表 ID 逗号隔开)
BelongingS_Treatment_Method	Varchar	20	Yes	随身物品处理方式字典表 ID 逗号隔开
BelongingS_Treatment_Record	varchar	500	Yes	未返还物品处理记录
Recipient_Person	varchar	10	Yes	领取人姓名
Recipient_Person_Number	varchar	20	Yes	领取人身份证号码
Treatment_Time	datetime	8	Yes	随身物品处理时间
Staff_ID	int	4	Yes	办案民警
Room_ID	Int	4	Yes	信息登记房间
Total_record	Int	4	default	需要填写的总记录数
Fill_record	Int	4	yes	当前填写的记录数

表 3.2.21 PHCSMP_CardReader: 设备信息表

字段名	类型	长度	允许空	备注
CardReader_ID	Int(自动加一)	4	No	设备 ID（主键）
CardReader_Name	Varchar	50	Yes	设备名称
CardReader_Type	int	4	No	设备类型（0: RFID、1: ID）

表 3.2.22 PHCSMP_Band: 手环信息表

字段名	类型	长度	允许空	备注
Band_ID	Int(自动加一)	4	No	手环 ID (主键)
Band_Type	int	4	No	手环类型 (0: RFID、1: ID)
Remark	Varchar	100	Yes	备注

表 3.2.23 PHCSMP_Process_Log: 警务业务作业日志表

字段名	类型	长度	允许空	备注
Log_ID	Int(自动加一)	4	No	业务日志 ID (主键)
Suspect_ID	Varchar	10	No	档案编号
Process_ID	Int	4	No	流程 ID
Start_Time	datetime	8	Yes	开始时间
End_Time	datetime	8	Yes	结束时间
Staff_ID	int	4	Yes	操作用户
IP_Address	varchar	20	No	操作 IP 地址

表 3.2.24 PHCSMP_BasicOperationLog: 系统基本操作日志表

字段名	类型	长度	允许空	备注
Log_ID	Int(自动加一)	4	No	业务日志 ID (主键)
Log_Information	Varchar	255	No	完成操作
Log_Time	datetime	8	No	操作时间
Staff_ID	int	4	No	操作用户
IP_Address	varchar	20	No	操作 IP 地址

表 3.2.25 PHCSMP_Line: 回路信息表

字段名	类型	长度	允许空	备注
Line_Count	int	4	No	回路总数 (默认 16)
Line_Used	int	4	No	已使用回路数 (默认 0); 涉案人员进入入区登记环节, 完成入区登记并成功提交数据后, 若录制回路不饱和可以进行录像, 则本字段加 1; 当离区登记成功且调用停止录像指令后本字段减 1。

4 系统详细设计及编码实现

系统包括智能化派出所系统和智能化终端程序两部分。智能化派出所系统有办案区业务登记模块、角色管理模块、嫌疑人信息管理模块、系统日志管理模块、数据库管理模块和系统初始化模块六大部分。

4.1 智能化派出所系统详细设计

4.1.1 实体类

实体类层主要是通过 Hibernate 框架建立对象与数据库的数据表的关系映射，通过对象的数据结构来对应操作数据库的数据表中的数据。以 PHCSMP_Activity_Record 类为例，部分代码如图 4.1 所示：

```
public class PHCSMP_Activity_Record {  
    /**  
    private int activity_Record_ID;  
    private String suspect_ID; // 档案编号  
    private String start_Time; // 开始时间  
    private String end_Time; // 结束时间  
    private int room_ID; // 房间ID  
    private String activity_Record; // 活动内容  
    private String remark; // 备注  
    private Integer staff_ID; // 办案民警ID（多个民警ID间逗号间隔）  
    private int total_record = 60; // 需要填写的总记录数  
    private int fill_record; // 当前填写的记录数  
  
    public int getActivity_Record_ID() {}  
  
    public void setActivity_Record_ID(int activity_Record_ID) {}  
  
    public String getSuspect_ID() {}  
  
    public void setSuspect_ID(String suspect_ID) {}  
  
    public String getStart_Time() {}  
}
```

图 4.1 PHCSMP_Activity_Record 的实体类代码

4.1.2 业务逻辑层

系统中业务逻辑层主要分为接口和实现，其中接口只负责定义，实现是对这个接口进行实现的。

(1) 系统中业务逻辑层接口的定义，下面以 ActivityRecordService 为例，代码如图 4.2 所示：

```
public interface ActivityRecordService {
    * 保存活动记录信息的方法
    void saveActivityRecordInfor(PHCSMP_Activity_Record model);

    * 保存活动记录信息list的方法
    void saveActivitysInfor(List<PHCSMP_Activity_Record> activitys);

    * 根据档案号查询嫌疑人档案信息
    List<PHCSMP_Activity_Record> findInforBySuspetcId(String suspectId);

    * 根据档案号登记办案区活动记录
    List<PHCSMP_Activity_Record> selectActivityRecordInfor(String suspectId);

    public void updatevedio_Number(String vedio_Number, int bandID, String identifyCard_Number);
}
```

图 4.2 ActivityRecordService 接口定义代码

(2) 业务逻辑层接口的实现，下面以 ActivityRecordServiceImpl 为例，代码如图 4.3 所示：

```
@Service("activityRecordService")
public class ActivityRecordServiceImpl extends DaoSupportImpl<PHCSMP_Activity_Record>
    implements ActivityRecordService {

    private ActivityRecordDao activityRecordDao;

    public void saveActivityRecordInfor(PHCSMP_Activity_Record model) {}

    public void saveActivitysInfor(List<PHCSMP_Activity_Record> activitys) {}

    public List<PHCSMP_Activity_Record> selectActivityRecordInfor(String suspectId) {}

    public List<PHCSMP_Activity_Record> findInforBySuspetcId(String suspectId) {}

    public void updatevedio_Number(String vedio_Number, int band_ID, String identifyCard_Number) {}
}
```

图 4.3 ActivityRecordServiceImpl 接口实现代码

4.1.3 数据访问层

数据访问层主要是实现与数据库进行数据增删改查等操作。数据访问层代码分为接口和实现，其中接口只负责数据访问层与数据库之间数据交互的接口的定义，实现则是对接口的具体实现，涉及到如何与数据库建立连接等操作。下面我们

ActivityRecordDao 和 ActivityRecordDaoImpl 为例，来阐述数据访问层数据库之间数据交互的代码实现。

(1) 数据访问层数据访问接口的定义，由于项目中使用 druid 数据库连接池，提高数据库的查询效率，并使用 Spring 来代理数据源，因此省去了大部分关于如何建立与数据库的连接等操作，在数据访问层的接口定义中，本系统只定义关于数据增删改查的操作。下面以 ActivityRecordDao 为例，代码如图 4.4 所示：

```
public interface ActivityRecordDao extends DaoSupport<PHCSMP_Activity_Record> {
    * 批量保存活动记录信息
    void saveActivitysInfor(List<PHCSMP_Activity_Record> activitys);
    * 根据档案号登记办案区活动记录
    List<PHCSMP_Activity_Record> selectActivityRecordInfor(String suspectId);
    public void updatevedio_Number(String vedio_Number, int band_ID,
}
```

图 4.4 ActivityRecordDao 接口定义代码

(2) 数据访问层数据访问接口的实现，下面以 ActivityRecordDaoImpl 为例，代码如图 4.5 所示：

```
@Repository("activityRecordDao")
public class ActivityRecordDaoImpl extends DaoSupportImpl<PHCSMP_Activity_Record>
    implements ActivityRecordDao {
    private Transaction tx = null;
    private Session session = null;
    private String hql = null;
    * 批量保存询问讯问等活动记录
    public void saveActivitysInfor(List<PHCSMP_Activity_Record> activitys) {}
    * 查询记录，此方法和DaoSupport有重复，没必要写
    public List<PHCSMP_Activity_Record> selectActivityRecordInfor(String suspectId) {}
    public void updatevedio_Number(String vedio_Number, int band_ID, String identifyCard_Number) {}
}
```

图 4.5 ActivityRecordDaoImpl 接口实现代码

4.1.4 表现层

表现层就是将后台处理好的数据用 HTML 页面展示出来，本节中主要介绍的是

系统中主要的几个业务逻辑的页面实现效果。以及嫌疑人在办案区完成操作后生成的信息报告。下面一一介绍：

（1）嫌疑人入区登记：主要功能是实现嫌疑人基本信息的录入，RFID 手环与嫌疑人的绑定。实现效果如图 4.6 所示：

Personal information registration
嫌疑人入区信息登记

档案编号：LB-HB-201703180

0%

25%

50%

75%

100%

入区登记进度

身份证信息

2017年03月18日

嫌疑人入区登记

身份证照

身份证号码

姓名：德古拉

性别：男

出生日期：1961

家庭住址：山西省晋石市灵石区灵城镇三海村委会名地村37号

民族：汉

140104196108123556

读卡

人员联系信息

证件类型：	<input type="checkbox"/> 居民身份证	<input type="checkbox"/> 临时身份证	<input type="checkbox"/> 军官证	<input type="checkbox"/> 武警警官证
	<input type="checkbox"/> 士兵证	<input type="checkbox"/> 军队学员证	<input type="checkbox"/> 军队文职干部证	<input type="checkbox"/> 军队离休退休干部证
	<input type="checkbox"/> 军队职工证	<input type="checkbox"/> 护照	<input type="checkbox"/> 港澳同胞回乡证	<input type="checkbox"/> 港澳居民来往内地通行证
	<input type="checkbox"/> 中华人民共和国来往港澳通行证	<input type="checkbox"/> 台湾居民来往大陆通行证	<input type="checkbox"/> 大陆居民往来台湾通行证	<input type="checkbox"/> 外国人居留证
	<input type="checkbox"/> 外国人出入境证	<input type="checkbox"/> 外交官证	<input type="checkbox"/> 领事馆证	<input type="checkbox"/> 难民证
证件号码：				
现住址：				
联系方式：				

进入办案区信息

入区时间：				RFID手环：请选择.....
进入办案区原因：	<input type="checkbox"/> 投案自首	<input type="checkbox"/> 治安传唤	<input type="checkbox"/> 刑事传唤	<input type="checkbox"/> 传唤	
	<input type="checkbox"/> 刑事传唤	<input type="checkbox"/> 取保候审	<input type="checkbox"/> 监视居住	<input type="checkbox"/> 逮捕	
	<input type="checkbox"/> 被害人	<input type="checkbox"/> 证人	<input type="checkbox"/> 辨认陪伴人员		

图 4.6 嫌疑人入区信息登记

（2）嫌疑人人身安全检查：主要功能是实现嫌疑人人身信息的采集，包括嫌疑人自述的身体情况，干警检查嫌疑人体表，是否饮酒，是否拍照，以及其他特殊情况

血液检查，尿检，其他项目等信息的检查，采集，入库。并进行历史信息的核查比对，看是否存在二进宫或者身负其他刑事案件的情况。如图 4.8 所示：

Personal Information Collection
嫌疑人信息采集

档案编号：LB-HB-201703175

0%25%50%75%100%

入区登记人身安全检查信息采集及项目登记信息入库检查对比

活动记录登记表

入区事由：投案自首

嫌疑人入区登记照片

身份证照

身份证号码

姓名：周鑫

性别：男民族：汉

出生日期：1994年08月21日

住址：山西省长治市郊区故县文明北街花园小区17栋3单元18户

140411199408210451

信息采集

信息采集：☒是☐否

采集项目：☐身份信息☐指纹☐血液☐尿液☐其他

信息入库：☒是☐否

检查对比：☒是☐否

图 4.8 嫌疑人信息采集

（4）询问讯问活动记录：该功能模块是实现了对嫌疑人在办案区内进行审问情况的记录。实现效果如图 4.9 所示：

人身检查信息:

自述症状正常

信息登记房间1

人身检查状态体表有伤痕,有饮酒

办案民警围绕

检查情况中

信息采集信息:

是否入库否

信息登记房间1

是否进行采集否

办案民警

采集项目指纹,血液,结婚后

活动记录登记

填写完整度0%

序号	开始时间	结束时间	活动内容	备注
0	<input type="text"/>	<input type="text"/>	---请选择---	
1	<div>+添加-删除</div>			

注：1、请办案民警注意对嫌疑人在办案区的活动做详细记录，确保嫌疑人在办案区内无时间盲区的登记遗漏。

图 4.9 询问讯问活动记录



图 4.10 嫌疑人入区信息报告

(5) 嫌疑人入区信息报告：该页面主要功能是实现嫌疑人从入区到离区的整个流程的信息，以报告的方式显示出来，并提供报告 pdf 格式的下载。实现效果如图 4.10 所示：

4.2 智能化终端程序

智能化终端程序包括树莓派的 RFID 串口读取程序和 web 端树莓派访问接口程序两部分。

4.2.1 RFID 串口读取程序

RFID 串口读取程序主要是通过 RXTX 串口读取工具包读取树莓派的串口传来的 RFID 信号，并将其转成可识别的十六进制字符串，通过 HTTP 协议将读取到的信息发送到 web 端树莓派访问接口。然后 web 服务器在处理完成之后，会向树莓派发送操作成功的手环 ID，树莓派在收到信息后会执行 shell 脚本，从而触发语音系统的指令，完成本系统与语音系统的对接。实现代码如图 4.11 所示：

```
public class SerialRead {
    public String data;
    public static String oldDeviceId = null;

    * 串口初始化
    public void init() {}
    * 串口读取的监听事件
    class Read implements SerialPortEventListener {
        InputStream inputStream;
        SerialPort serialPort;
        Thread readThread;

        * 构造方法，用来初始化串口监听事件
        public Read(CommPortIdentifier portId) throws InterruptedException {}

        * BI -> 通讯中断, CD -> 载波检测, CTS -> 清除发送, DATA_AVAILABLE -> 有数据到达, DSR -> 数据设备准备好.
        public void serialEvent(SerialPortEvent event) {}
    }

    * 把字节数组转换成16进制字符串
    public static String bytesToHexString(byte[] bArray) {}

    * 校验和计算
    private byte CheckSum(byte[] readBuffer, int count) {}
}
```

图 4.11RFID 串口读取程序

4.2.2 web 端树莓派访问接口程序

(1) 接收树莓派发送过来的 HTTP POST 请求，从而获取 RFID 读卡器和手环的请求数据。

(2) 根据请求数据，从数据库查询手环 ID 对应的嫌疑人信息，然后根据 RFID 读卡器的编号来决定调用 websocket 向前台推送与该嫌疑人相关的业务页面。

(3) 根据业务流程来决定是开启新一路录像，还是切换录像源，然后调用第三方录播服务器的接口，实现相关业务功能。

(4) 向树莓派请求返回该嫌疑人对应的编号。

(5) 树莓派在收到 HTTP 响应之后，触发物控子系统的 shell 脚本，从而播放触

发音频信号的 shell，播放该业务房间对应的音频信息。树莓派程序的代码如图 4.12 所示：

```
@Controller
@Scope("prototype")
public class RFID_ReadAction {
    private static WebSocketUtils ws = new WebSocketUtils();
    protected HttpServletRequest request;
    protected HttpServletResponse response;
    protected ServletContext application;
    @Autowired
    private SuspectService suspectService;
    @Autowired
    private RoomService roomService; // 查询房间号的dao
    @Autowired
    private BandService bandService;

    * public String readRFID(HttpServletRequest request) throws IOException {}

    * private void triggerWebsocket(PHCSMP_Room room, String suspectID) {}

    * * 根据新需求更改，根据房间号和Record_Status switch和RoomID改，录像状态的切换，(0/1入区时，1->2RFID中，3结束LeaveAction)
    * private void VedioCommandAndUpdateMessage(PHCSMP_Suspect suspect, PHCSMP_Room room, int bandId) thro

    private void repairSwitch(int cardReader_Switch, String suspect_ID) {
        suspectService.updateSwitch(cardReader_Switch, suspect_ID);
    }
}
```

图 4.12 web 端树莓派访问接口程序

5 软件测试

5.1 测试的目的及技术

软件测试作为软件质量保证的重要手段，贯穿整个软件生命周期，从程序测试扩展到需求和设计的评审，涵盖静态测试和动态测试，并依据质量标准和测试规范，主要采用白盒测试方法和黑盒测试方法，完成各项具体的测试任务，以保证软件产品的质量^[8]。

（1）测试的目的

软件测试是软件开发过程中的一个重要环节，软件测试的目的就是对本系统进行验证，发现在本系统中所存在的各种不可预知的问题。

（2）测试的技术

① 黑盒测试技术

黑盒测试：也称功能测试，是测试人员使用测试用例来检测系统中每个功能是否可以按照预期正常工作。

② 白盒测试技术

也称结构测试，是测试人员按照系统代码内部的结构进行测试，通过测试工具来检测系统内部功能的实现是否按照软件规格说明书的规定正常运行，检验程序中的每条通路是否都能按预期正确的完成工作^[8]。

5.2 测试的步骤

（1）单元测试

单元测试是指系统开发过程中，对系统的最小可测试单元在工具的帮助下进行检查和验证。

（2）集成测试

集成测试是单元测试的逻辑扩展。集成测试是在单元测试的基础上，采用的方法是测试软件单元的组合能否正常工作，以及与其他组的模块能否集成起来工作。最后，还要测试系统所有模块组合在一起能否正常工作^[8]。

5.3 测试的过程

5.3.1 系统架构测试

表 5.2 系统架构测试用例

测试目的	验证系统各层之间数据传递的正确性				
测试数据	登录方法	用户名=admin	密码=123		
操作步骤	操作描述	数据	期望结果	实际结果	测试状态
1	表现层调用业务逻辑层	用户名=admin, 密码=admin	登录成功	与预期结果相同	良好
2	业务逻辑层调用数据访问层	用户名=admin, 密码=admin	登录成功	与预期结果相同	良好
3	数据访问层与数据库交互	用户名=admin	得到该用户的数据信息	与预期结果相同	良好

5.3.2 WEB 端树莓派接口测试

表 5.3 树莓派接口测试用例

测试目的	验证系统各层之间接口实现的正确性				
测试数据			手环 ID		
操作步骤	操作描述	数据	期望结果	实际结果	测试状态
1	刷卡操作读取手环信息, 通过 HTTP 发送到 WEB 端	手环信息跟 RFID 读卡器信息	录播服务器正常工作, 树莓派播放声音	与预期结果相同	良好
2	刷卡之后, 查看 web 端日志	打印日志	日志与预期一致	与预期结果相同	良好
3	刷卡操作在业务房间完成之后	录播服务器生成的视频文件	录播服务器生成的视频文件下载到本地	与预期结果相同	良好

5.3.3 系统主要业务测试

表 5.4 系统主要业务测试用例

测试目的	验证系统各层之间接口实现的正确性				
测试数据			入区嫌疑人所有信息		
操作步骤	操作描述	数据	期望结果	实际结果	测试状态
1	嫌疑人信息登记	嫌疑人基本信息, 绑定手环根据手环 ID,	成功提交, 刷卡不出错	与预期结果相同	良好
2	刷卡, 采集嫌疑人人身安全信息	调出嫌疑人基本信息, 在页面添加嫌疑人人身安全信息	成功提交, 刷卡不出错	与预期结果相同	良好

3	刷卡, 嫌疑人 办案区活动信 息记录	根据手环 ID, 调出嫌疑人基 本信息, 在页面 添加嫌疑人活 动记录	成功提交, 刷 卡不出错	与预期结果相同	良好
4	刷卡, 嫌疑人 出区登记, 解 除手环绑定	根据手环 ID, 调出嫌疑人基 本信息, 添加出 区信息	成功提交, 刷 卡不出错, 手 环解除绑定	与预期结果相同	良好
5	嫌疑人办案区 内信息报告	嫌疑人 ID	嫌疑人对应 的信息报告	与预期结果相同	良好

5.3.4 树莓派读取串口数据测试

表 5.5 树莓派读取串口数据测试用例

测试目的	验证树莓派采集数据，数据传输以及数据反馈实现的正确性				
测试数据			手环 ID		
操作步骤	操作描述	数据	期望结果	实际结果	测试状态
1	使用手环在读卡器上刷卡	手环信息跟RFID 读卡器信息	录播设备开始录像	与预期结果相同	良好
2	使用手环在读卡器上刷卡	返回操作成功的标志	树莓派连接的喇叭发出相应的声音	与预期结果相同	良好

5.4 测试结论

通过测试, 基本上可以检测出整个项目存在的 bug 和性能问题。另外, 整个系统在整体上设计合理, 但是在一些功能设计上, 有待进一步的优化和完善, 不过基本满足客户现阶段的需求。通过测试后, 该项目已经部署上线使用。

5.5 项目部署

(1) 智能化派出所系统部署

① 使用 IDE 导出本系统 war 包, 将本系统的 war 包放到 Tomcat 服务器的 webapps 目录下。

② 修改配置文件, 配置项目的数据库的用户名, 密码, 配置 FTP 服务器的 IP, 配置第三方录播系统服务器 IP。

③ 在第三方录播系统的 web 管理系统中配置需要连接的网络摄像头的 IP，用户名，密码，配置录播系统的 FTP 输出源的 IP。

(2) 智能化终端程序部署

① 将树莓派程序编译成字节码文件上传到树莓派。

② 修改配置文件，配置 RFID 的串口号。

③ 通过 `Java MainApp` 命令启动串口读取程序，看到打印出串口号，说明程序启动成功，否则检查串口是否被占用。

6 结论

本系统采用 MVC 加三层架构的设计模式，满足软件工程中“高内聚，低耦合”的要求，在前期开发和后期维护的过程中在效率上有很大的提升。

优秀的项目架构，可以使得开发人员的分工明确，开发效率高。在定义好各层次之间的接口，负责不同逻辑设计的开发人员就可以集中精力开发自己的任务了，保证项目的同步开发。

6.1 完成的功能模块

- (1) 角色管理。完成系统角色的设置及权限分配。
- (2) 采集设备管理。完成采集设备和 RIFD 手环的注册及开通、注销。
- (3) 涉案人员信息管理。通过设备读取涉案人员二代身份证信息，完成涉案人员信息的采集、维护。
- (4) 采集信息填报。在不同区域，按照相应功能和用户角色权限，完成采集数据的填报和上传保存。
- (5) 系统日志管理。完成系统操作及通讯日志的保存。
- (6) 系统初始化。系统的初始化配置。

6.2 待优化的功能

- (1) 嫌疑人入区之后的整个业务逻辑的控制性能不稳定有待优化，根据客户提出的新的需求，开发新的功能。
- (2) 第三方录播系统的级联，同时在本系统中实现录播回路的扩展，提供系统的可重用性和业务处理量。

参考文献

- [1] 张海藩. 软件工程导论 [M]. 第五版 .清华大学出版社, 2008.
- [2] 牛丽平. UML 面向对象设计与分析 [M]. 清华大学出版社, 2008.
- [3] 陈松乔. 现代软件工程 [M]. 第五版 .清华大学出版社, 2004.
- [4] 王占军;王伟; 赵志刚. 基于 FTP 协议的文件传输组件设计与实现 [J/OL]. 沈阳师范大学学报(自然科学版), 2012-02, 中国知网 <http://epub.cnki.net/>.
- [5] 刘继元. 基于树莓派的物联网应用 [J/OL]. 探索与观察 (2016 年 08 期), 2016-04, 中国知网 <http://epub.cnki.net/>.
- [6] 毕翼. 基于 Struts、XML 和 Ajax 整合框架的接处警信息管理系统设计及实现 [D/OL]. 东北大学, 2010-06 中国知网 <http://epub.cnki.net/>.
- [7] 蔡沁舟. 政法信息共享平台公安局刑事案件协同系统设计与实现[D/OL]. 电子科技大学, 2015-09, 中国知网 <http://epub.cnki.net/>.
- [8] 朱少民. 软件测试方法和技术 [M]. 清华大学出版社, 2010.
- [9] Miracle He: 理解 MVC 底层运行机制 [EB/OL]. 博客园, 2012-05-25, <http://www.cnblogs.com/hmiinyu/archive/2012/05/25/2517730.html>
- [10] 王珊. 萨师煊. 数据库系统概论(第 4 版) [M]. 高等教育出版社, 2006.
- [11] Singh. Information System Management[M]. S.B. Nangia APH publishing Corporation .2007.
- [12] John W. Satzinger. Systems Analysis and Design in a Changing World[M]. Cengage Learning EMEA, 2008.
- [13] Motor Industry Software Reliability Association. MISRA-C: 2004: guidelines for the use of the C language in critical systems. [M]. MIRA, 2008.
- [14] Peter Rob, Carlos Coronel .Database Systems[M]. Cengage Learning EMEA, 2008.
- [15] Toby J. Teorey, Stephen Buxton, Lowell Fryman, Terry Halpin. Database design: know it all [M]. Morgan Kaufmann, 2009.

致谢

在指导老师的帮助与同学的支持和帮助下，我能够在规定的时间内顺利地完成本次毕业设计，实属不易。在毕业设计的设计与实现的这段时间里，我明白了很多道理，并获得了丰富的项目经验和团队合作的经历，感受到大学的魅力和真谛。

在本次毕业设计实现的过程中，杨慧炯老师以孜孜不倦，认真负责的态度深深的影响着我和我的团队，让我们在完成毕业设计的道路上信念更加坚定。杨老师注重理论上与实践的结合，不断地教导我们“项目一定要站在用户的角度上思考问题”，使我们逐步认识到做项目不是简单的小 demo，不能随随便便，马马虎虎，而是认真负责，以用户优先。同时，杨老师在我们遇到问题的时候不断的帮助我们想办法，找解决方案，给我们提想法，开阔了我们的眼见，培养了我们独立解决问题的能力。在此，我对杨老师在毕业设计期间给予的指导和帮助表示由衷的感谢。同时还感谢其他老师在此期间的帮助和指导，以及学院为我提供良好的毕业设计环境。

最后感谢在毕业设计答辩中为我们操劳的各位老师。