密级：公开

高校固定资产管理系统的

设计与实现

**Design and Implementation of University Fixed Assets Management System**

|  |  |
| --- | --- |
| **学 院：** | 软件学院 |
| **学 号：** | 161203726 |
| **专业班级：** | 软件工程1607班 |
| **学生姓名：** | 刘靖诗 |
| **指导教师：** | 邵中 |

2020年06月

摘　要

在我国经济不断发展和高校扩招的背景下，我国高等教育已从精英教育转向大众教育，高校的办学规模也在不断扩大，办学规模的扩大必然会导致高校的固定资产的种类和数量不断增多，固定资产管理越来越复杂，传统的固定资产模式已不能适应新时代下的高校固定资产管理要求。在信息化时代背景下，高校管理已从传统模式转变为信息化管理模式，固定资产管理信息化也是其中重要的组成部分。

本文将二维码加密算法、资产核对算法和资产折旧算法三部分作为高校固定资产管理系统的核心内容。对资产管理、资产盘点、审批管理、系统管理等功能模块作为主要功能。还对资产进行GPS定位，对每个资产生成一个二维码，该二维码只可以在本文所说的移动端进行扫描查看。此外，通过资产信息提醒实现对高校资产的动态变化进行实时的把握。高校固定资产管理系统基于ThinkPHP5以及Tornado框架开发，移动端采用WebSocket进行通讯，以MySQL作为数据库，以B/S架构对软件进行设计。在开发完成后对本系统进行全面的测试，测试结果表明，系统能够完成预先设定的各项功能，并且运行稳定，可以投入实际的使用当中。

通过本文设计出来的高校固定资产管理系统能够实现高校固定资产管理的信息化，简化原有的固定资产管理的繁琐性，改善固定资产使用效率和管理效率，进而提高高校管理水平，为高校数字化、智能化管理添砖加瓦。

**关键词：**高校固定资产管理；管理系统；二维码管理

**Abstract**

In our country economy development and under the background of enrollment expansion of colleges and universities, Chinese higher education has shifted from elite education to mass education, college scale has been enlarged and the expansion of scale will inevitably lead to the amount and type of the fixed assets in colleges and universities, fixed assets management is more and more complex, the traditional pattern of fixed assets can not meet the needs of management of fixed assets in institutions of higher education under the new era. Under the background of the information age, university management has changed from the traditional mode to the information management mode, and the information of fixed assets management is also an important part of it.

In this paper, the qr code encryption algorithm, asset verification algorithm and asset depreciation algorithm as the core of the university fixed assets management system. The asset management, asset inventory, approval management, system management and other functional modules as the main function. Assets are also GPS located and each asset is generated with a qr code that can only be scanned on the mobile end of this article. In addition, the dynamic changes of university assets can be grasped in real time through the reminder of asset information. The university fixed asset management system is developed based on ThinkPHP5 and Tornado frameworks. The mobile terminal uses WebSocket for communication, MySQL as database, and B/S architecture for software design. After the completion of the development of the system for a comprehensive test, the test results show that the system can complete the preset functions, and stable operation, can be put into actual use.

The university fixed assets management system designed in this paper can realize the informatization of the university fixed assets management, simplify the complexity of the original fixed assets management, improve the use and management efficiency of fixed assets, and then improve the university management level, contributing to the university digital and intelligent management.

**Keywords:** University fixed assets management; Management system; Qr code management;

目　录

[摘　要 I](#_Toc41621393)

[Abstract II](#_Toc41621394)

[第 1 章 引　言 1](#_Toc41621395)

[1.1 研究背景与意义 1](#_Toc41621396)

[1.2 国内外研究现状 1](#_Toc41621397)

[1.2.1 应用现状 1](#_Toc41621398)

[1.2.2 研究现状 2](#_Toc41621399)

[1.3 论文主要技术 2](#_Toc41621400)

[1.4 论文框架结构 3](#_Toc41621401)

[第 2 章 相关技术 4](#_Toc41621402)

[2.1 ThinkPHP5框架 4](#_Toc41621403)

[2.1.1 ThinkPHP5的生命周期 4](#_Toc41621404)

[2.1.2 ThinkPHP5框架特性 5](#_Toc41621405)

[2.2 Tornado框架 5](#_Toc41621406)

[2.2.1 Tornado框架简介 5](#_Toc41621407)

[2.2.2 tornado.websocket.WebSocketHandler生命周期 6](#_Toc41621408)

[2.3 uni-app 6](#_Toc41621409)

[2.3.1 uni-app简介 6](#_Toc41621410)

[2.3.2 uni-app应用生命周期 6](#_Toc41621411)

[2.3.3 uni-app页面生命周期 7](#_Toc41621412)

[2.4 colorUI 7](#_Toc41621413)

[第 3 章 系统需求分析 8](#_Toc41621414)

[3.1 高校业务分析 8](#_Toc41621415)

[3.1.1 高校背景及业务需求 8](#_Toc41621416)

[3.1.2 固定资产管理的必要性 9](#_Toc41621417)

[3.2 系统功能描述 9](#_Toc41621418)

[3.2.1 固定资产管理的业务描述 9](#_Toc41621419)

[3.2.2 人员用户及权限管理业务描述 11](#_Toc41621420)

[3.3 系统非功能性业务描述 12](#_Toc41621421)

[3.3.1 性能需求 12](#_Toc41621422)

[3.3.2 操作需求 13](#_Toc41621423)

[3.3.3 安全需求 13](#_Toc41621424)

[第 4 章 固定资产管理系统设计 14](#_Toc41621425)

[4.1 系统设计原则与目标 14](#_Toc41621426)

[4.1.1 设计原则 14](#_Toc41621427)

[4.1.2 设计目标 15](#_Toc41621428)

[4.2 高校固定资产管理系统的功能设计 15](#_Toc41621429)

[4.2.1 资产管理模块的设计 15](#_Toc41621430)

[4.2.2 资产盘点模块的设计 16](#_Toc41621431)

[4.2.3 审批管理模块的设计 17](#_Toc41621432)

[4.2.4 系统管理模块的设计 18](#_Toc41621433)

[4.3 高校固定资产管理系统的数据库设计 19](#_Toc41621434)

[4.3.1 系统数据库设计框架概述 19](#_Toc41621435)

[4.3.2 数据库概念框架设计 20](#_Toc41621436)

[4.3.3 数据库物理结构设计 21](#_Toc41621437)

[第 5 章 固定资产管理系统的实现 27](#_Toc41621438)

[5.1 系统的实现环境 27](#_Toc41621439)

[5.2 资产管理模块的实现 27](#_Toc41621440)

[5.2.1 功能概述 27](#_Toc41621441)

[5.2.2 模块实现 27](#_Toc41621442)

[5.3 资产盘点模块的实现 33](#_Toc41621443)

[5.3.1 功能概述 33](#_Toc41621444)

[5.3.2 模块实现 33](#_Toc41621445)

[5.4 审批管理模块的实现 35](#_Toc41621446)

[5.4.1 功能概述 35](#_Toc41621447)

[5.4.2 模块实现 35](#_Toc41621448)

[5.5 系统管理模块的实现 37](#_Toc41621449)

[5.5.1 功能概述 37](#_Toc41621450)

[5.5.2 模块实现 37](#_Toc41621451)

[第 6 章 系统实验与结果分析 39](#_Toc41621452)

[6.1 系统测试的工具与环境 39](#_Toc41621453)

[6.1.1 系统测试工具 39](#_Toc41621454)

[6.1.2 系统测试环境 39](#_Toc41621455)

[6.2 系统测试流程 39](#_Toc41621456)

[6.3 系统测试用例 40](#_Toc41621457)

[6.3.1 资产管理功能测试用例 40](#_Toc41621458)

[6.3.2 资产盘点功能测试用例 40](#_Toc41621459)

[6.3.3 审批管理功能测试用例 41](#_Toc41621460)

[6.3.4 系统管理功能测试用例 41](#_Toc41621461)

[6.3.5 系统压力测试 42](#_Toc41621462)

[6.4 系统测试结果分析 42](#_Toc41621463)

[第 7 章 结　论 43](#_Toc41621464)

[参考文献 44](#_Toc41621465)

[致　谢 45](#_Toc41621466)

1. 引　言

## 研究背景与意义

近几年随着我国经济的突飞猛进，高等教育事业得到了蓬勃发展。高校固定资产规模随之扩大，资产构成日趋复杂，导致管理难度也越来越大。加强高校固定资产管理不仅是为了确保国有资产的安全完整，也是高校自身发展的内在需要。因此，进一步提高高校固定资产管理水平和能力显得至关重要。

当前信息化技术已经被逐步引入到高校的固定资产管理工作中，在台账建档、资产调拨、折损计算等业务工作中取得了明显的效益，大幅度地提高了工作效率和质量。但是多数现有系统所提供的业务支持不够全面，偏重于资产登记、调拨、使用和折损处置等环节，而在清查、盘点、实物核对这一类需要大量人员参与的业务工作中却缺乏有效的工作模式和功能支撑，导致上述工作依然需要通过少数人员的繁重劳动方可完成，未能从根本上解决高校固定资产管理所面临的痛点问题。尽管目前以RFID智能标签结合移动射频采集设备的解决方案可以在一定程度上缓解资产清查盘点工作的困境，但相对高昂的应用成本则成为其应用推广所面临的主要障碍。

本课题以图形二维码和智能手机取代RFID标签及专用射频采集设备，给出一种低成本且便于广泛参与的高校固定资产管理解决方案及配套软件系统。课题的研究及实践进一步提升固定资产管理水平和质量，提高资产清查盘点的准确性，降低应用成本和工作人员的劳动强度，具有明确的实用价值和现实意义。

## 国内外研究现状

分别从同类产品的应用现状和相关技术研究进展两个方面对与课题相关的现存工作进行调研分析，分类归纳如下。

### 应用现状

固定资产管理信息系统是以固定资产为管理对象，以资产台帐为基础，以统筹使用，防止流失，加强管理监督并提升使用效率为目的的管理系信息系统。当前市场上存在大量的同类产品，本文重点对其中与移动端应用功能支持有关的3款产品进行分析。

“易点固定资产管理系统”由易点易动公司开发，主要功能分为资产管理、耗材管理、财务管理以及审批管理四大模块。该系统支持移动端APP接入，可以使用手机扫码完成资产盘点，但无法对固定资产地理位置进行GPS定位；后台可提供对RFID标签的功能支持，但需要单独购买相应设备。

“联想百应资产管理系统”是由联想公司开发的一款同类应用系统。功能分为资产管理和耗材管理两大模块，无法使用手机扫码进行资产盘点，但可以申请工作人员代盘点，需要支付代盘点费用，也可以直接使用联想商城购买耗材。

“好客EAM资产管理软件”是一款由ASP/C#开发的开源固定资产管理软件，其功能分为个人办公、资产管理和耗材管理三大模块，无法使用手机扫码进行资产盘点，界面简单，但支持支持盘点任务的短信通知。

### 研究现状

大部分固定资产管理相关系统采用PC端B/S架构模式以及RFID自动识别技术，在2013年以前，多数为Java开发的C/S架构模式，目前，已有小部分系统采用了移动端的形式。

在2019年康健明和张旭[1]采用微信小程序中的微信Web开发者工具进行开发，PC端基于ASP.NET平台和SQLServer数据库实现浏览器与服务器(Browser/Server，B/S)架构的数据库系统，通过扫描二维码可以查看资产、房间、部门等信息。利用微信用户面广的特点，选择微信小程序进行移动端的开发，且微信小程序拥有应用程序的体验效果，不需要安装，即开即用。通过移动端与PC端结合应用，实现移动端扫码即可确定资产信息并上报，PC端资产管理部门核对，更方便快捷地进行资产管理。

杨伟元[2]基于RFID自动识别技术采用C/S客户端模式，管理员通过校园网对系统进行应用维护。并使用并通过手持式移动终端对固定资产设备进行现场高务盘点和网络在线设备对比统计查询。采用RFID技术，在未来高校固定资产资产管理工作中，实现进一步降低管理成本、提高管理效率及准确性，为高校的正常教学及科研工作提供基础保障。

## 论文主要技术

本文研究在PC端上采用ThinkPHP5的MVC三层架构设计开发高校固定资产管理系统，前端采用jQuery、Echarts、layui、Vue.js，而移动端采用Tornado框架作为后端开发框架，前端采用uni-app开发，在数据库上采用MySQL数据库，在开发语言上使用PHP、Python、JavaScript作为开发语言。

## 论文框架结构

本论文分为 7 部分进行概述

第 1 章为绪论部分，着重介绍了论文研究的背景及理论与实践价值，对国内外关于固定资产管理系统建设的研究与应用现状进行了梳理、归纳，明确了本论文研究所牵涉到的主要技术。

第 2 章是相关技术介绍，在前后端开发框架、关键算法技术等方面进行介绍。

第 3 章是系统需求分析，首先对系统的可行性、固定资产关系系统发展趋势、高校固定资产管理现状等问题进行了分析，然后详细论述系统业务需求、业务流程、功能需求和非功能性需求。

第 4 章是系统设计，在整体系统设计方案的基础上，对系统的具体功能实现做详细的设计阐述，对系统的资产信息管理模块、资产盘点模块、审批管理模块、系统管理以及日志管理模块等具体功能模块的设计，同时还有对系统界面的设置，保证界面的简洁美观。

第 5 章是系统实现，介绍系统资产信息管理模块、资产盘点模块、审批管理模块、系统管理以及日志管理模块等具体功能等模块的实现。

第 6 章是系统测试，对初步完成开发的系统进行测试检验，该环节包括对系统单元模块的测试还有系统整体架构测试。在每一个具体功能模块开发完成后先对其进行测试，确保该功能可以满足系统需求后，再进行另一模块部分的开发测试，在所有单元模块功能测试完成后，再对整体系统功能测试，确保单元模块功能的无缝衔接，最终实现系统整体功能可以满足完成前期的需求分析，系统中输入相关数据指令从而导出需求信息。

第 7 章是结论与展望，概述了论文研究得到的结论，指出了研究的不足之处及未来的研究方向。

1. 相关技术

高校固定资产管理系统分为PC端和移动端。PC端采用PHP语言ThinkPHP5框架；移动端前端采用uni-app、colorUI，后端采用Python语言Tornado框架。

## ThinkPHP5框架

### ThinkPHP5的生命周期

(1) 入口文件

用户发起的请求都会经过应用的入口文件，即public/index.php，通过define('APP\_PATH', \_\_DIR\_\_ . '/../application/');加载application里面的Controller，然后根据Controller中具体内容加载相应的View和Model。

(2) 引导文件

ThinkPHP5中的引导文件在thinkphp/start.php. 在引导文件中，会依次执行的操作：加载系统常量定义；加载环境变量定义文件；注册自动加载机制；注册错误和异常处理机制；加载惯例配置文件；执行应用。

(3) 注册自动加载

系统会调用 Loader::register()方法注册自动加载，在这一步完成后，所有符合规范的类库（包括Composer依赖加载的第三方类库）都将自动加载。

(4) 注册错误和异常机制

执行Error::register()注册错误和异常处理机制。

(5) 应用初始化

执行应用的第一步操作就是对应用进行初始化，包括：加载应用（公共）配置；加载扩展配置文件（由extra\_config\_list定义）；加载应用状态配置；加载别名定义；加载行为定义；加载公共（函数）文件；注册应用命名空间；加载扩展函数文件（由extra\_file\_list定义）；设置默认时区；加载系统语言包。

(6) URL访问检测

应用初始化完成后，就会进行URL的访问检测，包括PATH\_INFO检测和URL后缀检测。

(7) 路由检测

如果开启了url\_route\_on参数的话，会首先进行URL的路由检测。如果一旦检测到匹配的路由，根据定义的路由地址会注册到相应的URL调度。

(8) 分发请求

在完成了URL检测和路由检测之后，路由器会分发请求到对应的路由地址，这也是应用请求的生命周期中最重要的一个环节。在这一步骤中，完成应用的业务逻辑及数据返回。统一使用return返回数据，而不是echo输出。

(9) 响应输出

控制器的所有操作方法都是return返回而不是直接输出，系统会调用Response::send方法将最终的应用返回的数据输出到页面或者客户端，并自动转换成default\_return\_type参数配置的格式。所以，应用执行的数据输出只需要返回一个正常的PHP数据即可。

(10) 应用结束

在应用的数据响应输出之后，应用并没真正的结束，系统会在应用输出或者中断后进行日志保存写入操作。

### ThinkPHP5框架特性

(1) 规范

遵循PSR-2、PSR-4规范，Composer及单元测试支持。

(2) 严谨

异常严谨的错误检测和安全机制，详细的日志信息，为你的开发保驾护航。

(3) 灵活

减少核心依赖，扩展更灵活、方便，支持命令行指令扩展。

(4) API友好

出色的性能和REST支持、远程调试，更好的支持API开发。

(5) 高效

惰性加载，及路由、配置和自动加载的缓存机制。

(6) ORM

重构的数据库、模型及关联，MongoDb支持。

## Tornado框架

### Tornado框架简介

Tornado是一个Python web框架和异步网络库，起初由FriendFeed开发。 通过使用非阻塞网络I/O，Tornado可以支撑上万级的连接，处理长连接，WebSockets，和其他需要与每个用户保持长久连接的应用。而本项目就是使用了tornado.websocket与客户端进行通讯。

### tornado.websocket.WebSocketHandler生命周期

(1) WebSocketHandler.open(\*args, \*\*kwargs)

当打开一个新的 WebSocket 时调用。open 的参数是从 tornado.web.URLSpec 通过正则表达式获取的，就像获取 tornado.web.RequestHandler.get 的参数一样。

(2) WebSocketHandler.on\_close()

当关闭该 WebSocket 时调用。当连接被彻底关闭并且支持 status code 或 reason phtase 的时候，可以通过 self.close\_code 和 self.close\_reason 这两个属性来获取它们。

(3) WebSocketHandler.on\_message(message)

处理在 WebSocket 中收到的消息。

(4) WebSocketHandler.select\_subprotocol(subprotocols)

当一个新的 WebSocket 请求特定子协议(subprotocols)时调用

## uni-app

### uni-app简介

uni-app 是一个使用 Vue.js 开发所有前端应用的框架，开发者编写一套代码，可发布到iOS、Android、H5、以及各种小程序（微信/支付宝/百度/头条/QQ/钉钉/淘宝）、快应用等多个平台。

### uni-app应用生命周期

(1) onLaunch

当uni-app 初始化完成时触发（全局只触发一次）

(2) onShow

当 uni-app 启动，或从后台进入前台显示

(3) onHide

当 uni-app 从前台进入后台

(4) onError

当 uni-app 报错时触发

(5) onUniNViewMessage

对 nvue 页面发送的数据进行监听

### uni-app页面生命周期

(1) onload

监听页面加载，其参数为上个页面传递的数据，参数类型为Object。

(2) onShow

监听页面显示。页面每次出现在屏幕上都触发，包括从下级页面点返回露出当前页面。

(3) onReady

监听页面初次渲染完成。如果渲染速度快，会在页面进入动画完成前触发。

(4) onHide

监听页面隐藏。

(5) onUnload

监听页面卸载。

(6) onResize

监听窗口尺寸变化。

(7) onPullDownRefresh

监听用户下拉动作，一般用于下拉刷新。

(8) onReachBottom

页面上拉触底事件的处理函数。

(9) onShareAppMessage(不支持APP)

用户点击右上角分享。

(10) onPageScroll

监听页面滚动，参数为Object。

## colorUI

colorUI是在移动端用于前端的一款高颜值，高度自定义的Css组件库。

1. 系统需求分析

## 高校业务分析

### 高校背景及业务需求

(1) 高校背景

随着我国高校的快速发展，高校对固定资产的管理和使用越来越规范。手工固定资产管理的流程图如图3-1所示。



图3-1- 1 手工固定资产管理流程图

通过上述流程图可以看出，在采用手工固定资产管理的时候，高校都是通过账簿来对固定资产进行统计，首先对各部门固定资产定期施行统计，对所属归口部门的固定资产进行编码，通常所采用的方法就是贴标签，然后由各部门对贴好标签的固定资产进行造册统计，最后分别上报学校财务和后勤部门。

(2) 业务需求

固定资产管理是高校运行管理一个重要环节，对高校的教学、科研起着服务保障作用。高效率的固定资产管理，不仅可以为高校发展提供坚实的物资保障，还能有效杜绝腐败问题。同时，固定资产经营管理的好，也可以借助固定资产管理对各部门进行很好的监管和管理。如果对固定资产管理不善，则会导致固定资产利用率低，阻碍高校良性发展，甚至还会导致国有资产流失。由于国内大多数高校都是一个牵涉到部门多、校区多的单位，包括学校附属机构，中学和分机构。高校固定资产数量大、更新速度快但分布相对分散，因此对高校固定资产的管理，必须及时、准确地对资产信息进行修改和维护，这既是一项重要工作，又是一项困难任务。可喜的是，国内还是有许多在固定资产管理工作方面做的比较好的高校。[3]

### 固定资产管理的必要性

高校固定资产是保障学校的教学、科研、管理等各项活动顺利开展的物质基础条件。近年来，随着我国高等学校教育规模的迅速发展壮大，固定资产数量也越来越多。同时，资产构成也日趋复杂，管理难度也越来越大。如何做好资产精细化管理现已成为高校普遍面临的共同课题。

许多学校为了满足扩招、创建示范性院校等需要，加大了对教学设备、实验室、宿舍等地方固定资产的投入，随着学校的不断发展壮大，占用、拥有的资产规模日益庞大，种类越来越丰富，管理难度越来越大，管理要求也越来越高。随着计算机与信息技术不断地发展，使用过去的人工管理或单一计算机的方式来管理资产，消耗了大量的人力、物力，各种管理信息也不能共享，不能适应高校日益增长的信息化需要。目前，信息化的工作已在各个高校开展得如火如荼，特别在日益庞大的高校固定资产管理方面，如何利用信息化将高校固定资产管理资源信息共享，达到节约人力、物力，提炼管理效果、提高资产利用率、盘活资产，使资产增值最大化是重要的课题[4]。

## 系统功能描述

### 固定资产管理的业务描述

从目前高校发展状况以及高校对固定资产的需求来看，其对固定资产管理系统的要求具体体现在通用和实用上，主要功能包括资产管理、资产盘点、审批管理、系统管理等功能。

(1) 资产管理

资产管理是固定资产管理的核心功能，在资产管理中涉及到普通用户和管理员用户，管理员用户拥有全部权限，主要包括：资产添加、资产信息变更、资产删除、借用归还等，而普通用户只可以进行借用归还，如图3-2-1所示：



图3-2- 1 资产管理用例图

(2) 资产盘点

资产盘点时固定资产管理的特色内容，需要管理员用户方可创建盘点单，指定的用户方可进行资产盘点，如图3-2-2所示：



图3-2- 2 资产盘点用例图

(3) 审批管理

审批管理只可以管理员用户进入，普通用户无权限进入，如图3-2-3所示：



图3-2- 3 审批管理用例图

(4) 系统管理

系统管理只允许系统管理员用户进入，主要包括部门管理、职位管理、人员管理、用户管理。如图3-2-4所示：



图3-2- 4 系统管理用例图

### 人员用户及权限管理业务描述

本文开发研究的高校固定资产管理系统对系统登录的角色管理，涉及了4个用户层面，按照系统权限设置的不同，其用户角色也有所偏差。系统管理员可以设置系统的用户操作权限与用户信息资料的变更，便于系统信息的管理。如图3-2-5所示：



图3-2- 5 用户权限图

## 系统非功能性业务描述

### 性能需求

(1) 系统运行的响应能力：录入数据和进行简单查询的平均响应时间不超过一分钟；相对较复杂的查询及对报表统计平均响应时间不能超过 70 秒。

(2) 系统处理业务的能力：可以同时支撑在线用户大约 600 户，并发访问用户达到230 户。

(3) 系统运行的稳定性：在进行系统操作时要保证系统能够正常稳定的持续运行，实现系统资源的合理利用。[5]

### 操作需求

(1) 易用性：经常使用的按键可以设置为快捷键，TAB 键可以进行自由切换操作页面，默认按钮应可支持 ENTER 键操作，系统需使用标准术语构造显示界面。

(2) 规范性：基础菜单需要具备快捷栏，快捷栏内的图标可以清楚的表达操作需求。

(3) 帮助设施原则：在帮助界面应该及时更新帮助文档，有助于使用者能及时了解当前版本系统操作，并在使用者操作期间遇到疑难问题时可以及时提供帮助，文档以当前主流的 HTML 和 CHM 格式为最优方式。

(4) 合理性原则：应合理布置窗口构造，以对角线为中心线，将系统主界面合理分布在焦点中心。

(5) 独特性原则：在安装页面和操作页面的显著位置上安放产品介绍及标识。[6]

### 安全需求

针对高校固定资产管理系统的安全方面的需求，我们必须做到以下几个方面：

(1) 要有合理完善的权限管理机制。用来确保用户在合法及其具备的系统操作权限内进行系统操作。

(2) 关于系统中数据保密的安全性要求。系统管理中的资产数据通常都属于高校机密信息，其数据量也十分庞大。因此在对本系统的网络安全方面进行设计时，需要切实加强对网络病毒的防控，同时加强系统的防火墙功能，严格控制系统用户的操作权限，从各个方面保证固定资产管理系统中数据的完整性和安全性。

(3) 系统数据的存储。对本系统中的所有数据都应做好备份工作，以防发生意外时导致重要数据的遗失。[5]

(4)系统容错性的要求。目的在于保证系统数据的完整性和统一性。

1. 固定资产管理系统设计

## 系统设计原则与目标

### 设计原则

高校固定资产管理系统建立的原则主要从系统设计和系统实施两个角度来进行分析。要想达到一个有效、实用的系统，需要遵循如下原则[3]：

(1) 安全性原则

显然一个安全的系统，才是一个有效的系统，而系统的安全主要包括硬件的安全性、网络的安全性、软件的安全性和数据的安全性。而确保有效的安全需要单位制定相应安全的管理机制，在硬件上确保操作员的规范操作；网络上的安全采用防火墙策略；软件的安全在于做好软件的及时更新，安排专人维护；而在数据上的安全要定期做好数据的备份，如果数据量大的话可以考虑按天甚至按时来进行备份。

(2) 实用性原则

实用性原则主要是指系统能够满足单位需求，能够真正实施起来。一个不符合市场、不能满足高校所需要的系统，显然不能算一个有效地高校固定资产管理系统，同时确保系统能够保持跟上高校发展的需求。这样才能保证实用、连贯。

(3) 可靠性原则

可靠性是指系统的故障发生率、可恢复性等，作为一个网络系统，高校固定资产管理系统是用以收集数据和进行自动化处理的。如果系统发生故障，并且不可恢复的话，那将导致用户巨大的损失。从这个意义上说，固定资产管理系统对系统的可靠性要求非常高。所以需要对系统中的数据做定期的备份，提供“热备份”的硬件设备的配置，确保计算机能智能处理系统故障，并能快速恢复使用，确保系统运行的可靠性和数据的连续性。

(4) 经济性原则

在符合成本效益、设备的建设成本的前提下，确保系统能迅速和全面地保护客户的投资选择的系统要求。

(5)可优化原则

固定资产管理系统各模块设计要符合可优化原则，即系统界面、系统操作、运行等方面在系统建立后是可以不断完善、优化的。

### 设计目标

高校固定资产管理系统主要是针对高校目前固定资产管理的现状和不足而提出的一种解决方案。从设计初来看，要想到达目前高校所需要的功能，设计高校固定资产管理系统需要达到以下目标[3]：

(1) 固定资产管理系统的建立，确保行政主管部门可以方便的对其管辖范围内固定资产的价值、分布以及使用情况进行评估，和其他的查询和分析。

(2) 学员或老师借用资产后可以一键创建盘点单，用户自己进行盘点，后台会自动识别盘点结果供管理员审核。

(3) 高校可以跟的类型，数量，价值分析，研究，以及各种统计数据的汇总系统设备的实验报告按照规定，采用现代化的设备由实施科学管理的手段。

## 高校固定资产管理系统的功能设计

### 资产管理模块的设计

资产管理是对固定资产进行管理，由系统管理员登录在PC端后台根据已有固定资产的数量批量生产二维码，再讲二维码打印贴在固定资产上，然后管理员通过移动端操作，先扫码再填写资产信息最后拍照保存，这样资产录入工作就完成了。管理员也可以在PC端后台直接对固定资产进行增删改查操作，也可在移动端扫码进行固定资产信息的修改。普通用户可以通过移动端扫码的方式查看资产信息，以及借用或者归还资产。管理员可以在PC端后台创建盘点单，这样资产的使用者就会在移动端接收到盘点单，然后扫码、拍照操作进行盘点，管理员也可以去亲自盘点。普通用户和管理员可以申请资产报废或者维修，管理员在后台可以进行资产调拨。活动图如图4-2-1所示。



图4-2-1 资产管理活动图

### 资产盘点模块的设计

资产盘点是用户对自己借用的资产或管理员分配的盘点任务中的资产进行盘点。管理员创建盘点单之后，指定盘点人员和管理员可以查看盘点任务，用户将盘点单对应的资产进行扫码拍张以及GPS定位。盘点结束后将盘点单转至审批模块。活动图如图4-2-2所示。



图4-2-2 资产盘点活动图

### 审批管理模块的设计

在普通用户进行申请操作时，管理员需要审批，例如固定资产的借用，在管理员审核通过的时候会告知用户去哪里取或者派人统一派送。在普通用户进行资产的盘点之后，可以对盘点结果的异常情况进行审批。活动图如图4-2-3所示。



图4-2-3 审批管理活动图

### 系统管理模块的设计

系统管理只有系统管理员可以进行操作，组织架构是对各高校各部门进行管理，方便分部门管理固定资产，也可以对单独部门设置管理员，普通用户和系统用户管理是对用户的统一管理，只是两者的操作权限不同。活动图如图4-2-4所示。



图4-2-4 系统管理活动图

## 高校固定资产管理系统的数据库设计

### 系统数据库设计框架概述

为了应对系统的需求和建立一个可以储存数据的系统，必须进行数据库的设计。数据库的设计是系统设计中的关键一步，数据库设计的成功与否决定着系统实施的成败。在完成对数据库的逻辑分析后，需要进行数据库的物理分析，这主要包括：使用什么型号的硬件设备；在访问数据库时使用何种搜索引擎更加快捷方便等。

对系统数据库进行设计时，应至少建立一个表空间，并完成对象的数据库存储。系统数据库中的数据将直接参与到象结构的逻辑结构中，其包含了表、图、结构、序列、检索、关键词、存储过程等链集。而数据库则利用其包括的时间、空间、范围对相应的物理空间与象之间的关系进行表述。

段：通常由一个或多个范围组成，是一种属于表空间的逻辑存储结构，具体包括数据段、索引、中期、回滚段等。

范围：属于逻辑单元，由多个数据块共同连接组成，主要用来对数据库存储空间的使用情况进行说明。范围可以分为初始范围和增量分为两方面，初始范围是根据段所划分的首个范围，初始范围之外的所有范围则都属于增量范围。

数据块：数据块不同于操作系统中的块，它是数据库中用来组成范围的最小操作单位。

下面将对本文固定资产管理系统的数据库选型及其逻辑结构设计进行详细介绍。[5]

### 数据库概念框架设计

系统数据库的设计主要是数据库在处理来自应用服务器的请求后将相关的数据反馈给应用服务器，应用服务器将最后的结论发送到客户端。在实体的固定资产管理系统中，一个部门可以管理多个职员。一个职员只能从属一个部门；多个资产和多个职员相互之间可以交叉对应，可以一对一也可以一对多，而二维码与固定资产是互相唯一的[5]。如图4-3-1所示。



图4-3-1 数据库框架设计图

### 数据库物理结构设计

本文系统中所包括的数据表主要有资产信息表、借用归还表、资产类别表、资产盘点表、资产盘点单表、资产盘点状态表、部门信息表、地点信息表、日志信息表、日志操作表、日志分类表、人员信息表、职位信息表、资产维修表、资产状态表、用户信息表、用户权限表等。

(1) 资产信息表如表4-3-1所示：

表4-3-1 资产信息表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | as\_id | int | 10 | 主键 | 主键 |
| 2 | as\_no | varchar | 64 | 资产编号 | 唯一 |
| 3 | as\_name | varchar | 64 | 资产名称 | 非空 |
| 4 | as\_price | double | 10,2 | 资产价格 | 非空 |
| 5 | cate\_id | int | 11 | 类别id | 非空 |
| 6 | sta\_no | varchar | 255 | 状态编号 | 非空 |
| 7 | as\_import\_time | datetime | 0 | 资产入库时间 | 非空 |
| 8 | as\_image | varchar | 255 | 资产照片 | 非空 |
| 9 | as\_local\_id | int | 11 | 资产地点 | 非空 |
| 10 | as\_qrcode | varchar | 255 | 资产二维码 | 非空 |
| 11 | as\_exist | int | 11 | 记录是否存在 | 非空 |

(2) 借用归还表如表4-3-2所示：

表4-3-2 借用归还表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | bl\_id | int | 10 | 主键 | 主键 |
| 2 | as\_no | varchar | 64 | 资产编号 | 非空 |
| 3 | u\_id | int | 11 | 用户id | 非空 |
| 4 | b\_time | datetime | 0 | 领用时间 | 非空 |
| 5 | l\_time | datetime | 0 | 归还时间 | 非空 |
| 6 | bl\_ok | int | 11 | 该记录已完成 | 非空 |
| 7 | bl\_exist | int | 11 | 记录是否存在 | 非空 |

(3) 资产类别表如表4-3-3所示：

表4-3-3 资产类别表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | cate\_id | int | 10 | 主键 | 主键 |
| 2 | cate\_no | varchar | 32 | 类别编号 | 唯一 |
| 3 | cate\_name | varchar | 64 | 类别名称 | 非空 |
| 4 | cate\_exist | varchar | 11 | 记录是否存在 | 非空 |

(4) 资产盘点表如表4-3-4所示：

表4-3-4 资产盘点表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | c\_id | int | 10 | 主键 | 主键 |
| 2 | c\_title | varchar | 255 | 盘点单名称 | 非空 |
| 3 | u\_id | int | 11 | 指定盘点人 | 非空 |
| 4 | c\_time | datetime | 0 | 盘点单创建日期 | 非空 |
| 5 | c\_e\_time | datetime | 0 | 盘点限期 | 非空 |
| 6 | c\_sta\_no | varchar | 16 | 盘点单状态 | 非空 |
| 7 | c\_exist | int | 11 | 记录是否存在 | 非空 |

(5) 资产盘点单表如表4-3-5所示：

表4-3-5 资产盘点单表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | c\_c\_id | int | 10 | 主键 | 主键 |
| 2 | c\_id | int | 11 | 所属盘点单id | 非空 |
| 3 | as\_no | varchar | 64 | 资产编号 | 非空 |
| 4 | c\_c\_image | varchar | 255 | 盘点图片 | 非空 |
| 5 | c\_c\_exist | int | 11 | 记录是否存在 | 非空 |

(6) 资产盘点状态表如表4-3-6所示：

表4-3-6 资产盘点状态表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | c\_sta\_id | int | 10 | 主键 | 主键 |
| 2 | c\_sta\_no | varchar | 16 | 盘点单状态编号 | 唯一 |
| 3 | c\_sta\_name | varchar | 255 | 盘点单状态名 | 非空 |

(7) 部门信息表如表4-3-7所示：

表4-3-7 部门信息表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | dep\_id | int | 10 | 主键 | 主键 |
| 2 | dep\_no | varchar | 32 | 部门编号 | 唯一 |
| 3 | dep\_name | varchar | 40 | 部门名称 | 非空 |
| 4 | up\_dep\_id | int | 11 | 上级部门id | 非空 |
| 5 | dep\_remark | varchar | 255 | 备注 | 可空 |
| 6 | dep\_exist | int | 11 | 记录是否存在 | 非空 |

(8) 地点信息表如表4-3-8所示：

表4-3-8 地点信息表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | local\_id | int | 10 | 主键 | 主键 |
| 2 | local\_name | varchar | 255 | 地点名称 | 非空 |
| 3 | local\_data | varchar | 255 | GPS相关数据 | 非空 |
| 4 | local\_exist | int | 11 | 记录是否存在 | 非空 |

(9) 日志信息表如表4-3-9所示：

表4-3-9 日志信息表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | log\_id | int | 10 | 主键 | 主键 |
| 2 | log\_category\_no | varchar | 32 | 日志类别编号 | 非空 |
| 3 | log\_action\_no | varchar | 32 | 日志操作编号 | 非空 |
| 4 | u\_id | int | 11 | 用户id | 非空 |
| 5 | log\_datetime | datetime | 0 | 操作时间 | 非空 |
| 6 | common\_id | int | 11 | 公用id | 非空 |
| 7 | table\_name | varchar | 64 | 表名 | 非空 |
| 8 | table\_main\_key | varchar | 32 | 关联的表的主键 | 非空 |
| 9 | log\_exist | int | 11 | 记录是否存在 | 非空 |

(10) 日志操作表如表4-3-10所示：

表4-3-10 日志操作表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | log\_action\_id | int | 10 | 主键 | 主键 |
| 2 | log\_action\_no | varchar | 32 | 日志操作编号 | 唯一 |
| 3 | log\_action\_name | varchar | 128 | 日志操作名称 | 非空 |

(11) 日志分类表如表4-3-11所示：

表4-3-11 日志分类表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | log\_category\_id | int | 10 | 主键 | 主键 |
| 2 | log\_category\_no | varchar | 32 | 日志类别编号 | 唯一 |
| 3 | log\_category\_name | varchar | 64 | 日志类别名称 | 非空 |

(12) 人员信息表如表4-3-12所示：

表4-3-12 人员信息表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | p\_id | int | 10 | 主键 | 主键 |
| 2 | p\_no | varchar | 32 | 工号 | 唯一 |
| 3 | dep\_id | int | 11 | 部门主键 | 非空 |
| 4 | pos\_id | int | 11 | 职位主键 | 非空 |
| 5 | p\_name | varchar | 40 | 姓名 | 非空 |
| 6 | p\_sex | char | 2 | 性别 | 非空 |
| 7 | p\_ic | varchar | 20 | 身份证号 | 非空 |
| 8 | p\_email | varchar | 40 | 邮箱 | 非空 |
| 9 | p\_exist | int | 11 | 记录是否存在 | 非空 |

(13) 职位信息表如表4-3-13所示：

表4-3-13 职位信息表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | pos\_id | int | 10 | 主键 | 主键 |
| 2 | pos\_no | varchar | 32 | 职位编号 | 唯一 |
| 3 | pos\_name | varchar | 40 | 职位名称 | 非空 |
| 4 | pos\_remark | varchar | 255 | 备注 | 可空 |
| 5 | pos\_exist | int | 11 | 记录是否存在 | 非空 |

(14) 资产维修表如表4-3-14所示：

表4-3-14 资产维修表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | rep\_id | int | 10 | 主键 | 主键 |
| 2 | as\_no | varchar | 64 | 资产编号 | 非空 |
| 3 | rep\_price | double | 10,2 | 维修费用 | 非空 |
| 4 | rep\_time | datetime | 0 | 维修时间 | 非空 |
| 5 | rep\_exist | int | 11 | 记录是否存在 | 非空 |

(15) 资产状态表如表4-3-15所示：

表4-3-15 资产状态表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | sta\_id | int | 10 | 主键 | 主键 |
| 2 | sta\_no | varchar | 32 | 状态编号 | 唯一 |
| 3 | sta\_name | varchar | 64 | 状态名称 | 非空 |

(16) 用户信息表如表4-3-16所示：

表4-3-16 用户信息表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | u\_id | int | 10 | 主键 | 主键 |
| 2 | p\_id | int | 11 | 人员表主键 | 非空 |
| 3 | u\_phone | varchar | 16 | 用户电话 | 非空 |
| 4 | u\_openid | varchar | 64 | 微信openid | 非空 |
| 5 | power\_no | varchar | 16 | 用户权限编号 | 非空 |
| 6 | u\_head | varchar | 255 | 用户头像 | 非空 |
| 7 | u\_money | int | 11 | 用户积分 | 非空 |
| 8 | u\_login\_code | varchar | 255 | 用户登录码 | 非空 |
| 9 | u\_exist | int | 11 | 记录是否存在 | 非空 |

(17) 用户权限表如表4-3-17所示：

表4-3-17 用户权限表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | power\_id | int | 10 | 主键 | 主键 |
| 2 | power\_no | varchar | 32 | 权限编号 | 唯一 |
| 3 | power\_name | varchar | 64 | 权限名 | 唯一 |

1. 固定资产管理系统的实现

## 系统的实现环境

本课题实验环境为1台普通PC机，硬件配置为1核的CPU和2G内存。软件环境配置为CentOS7.4的操作系统，安装Apache2.4作为Web服务器，MySQL5.7.25作为数据库管理系统。

开发与实验在同设备上进行，开发环境选择的PHP版本为PHP7.4，以phpmyadmin 4.0.10.20作为数据库可视化管理工具，以VisualStudioCode作为代码编辑器，微信开发者工具和HbuilderX作为项目集成开发工具。

## 资产管理模块的实现

### 功能概述

资产管理是对固定资产进行管理，由系统管理员登录在PC端后台根据已有固定资产的数量批量生产二维码，再讲二维码打印贴在固定资产上，然后管理员通过移动端操作，先扫码再填写资产信息最后拍照保存，这样资产录入工作就完成了。管理员也可以在PC端后台直接对固定资产进行增删改查操作，也可在移动端扫码进行固定资产信息的修改。普通用户可以通过移动端扫码的方式查看资产信息，以及借用或者归还资产。管理员可以在PC端后台创建盘点单，这样资产的使用者就会在移动端接收到盘点单，然后扫码、拍照操作进行盘点，管理员也可以去亲自盘点。普通用户和管理员可以申请资产报废或者维修，管理员在后台可以进行资产调拨。

### 模块实现

从上节可以看出资产管理功能在PC端和移动端都可以进行，在PC端可以进行资产入库功能，该功能需要进行资产的增删改查，PC端界面如图5-1-1所示：

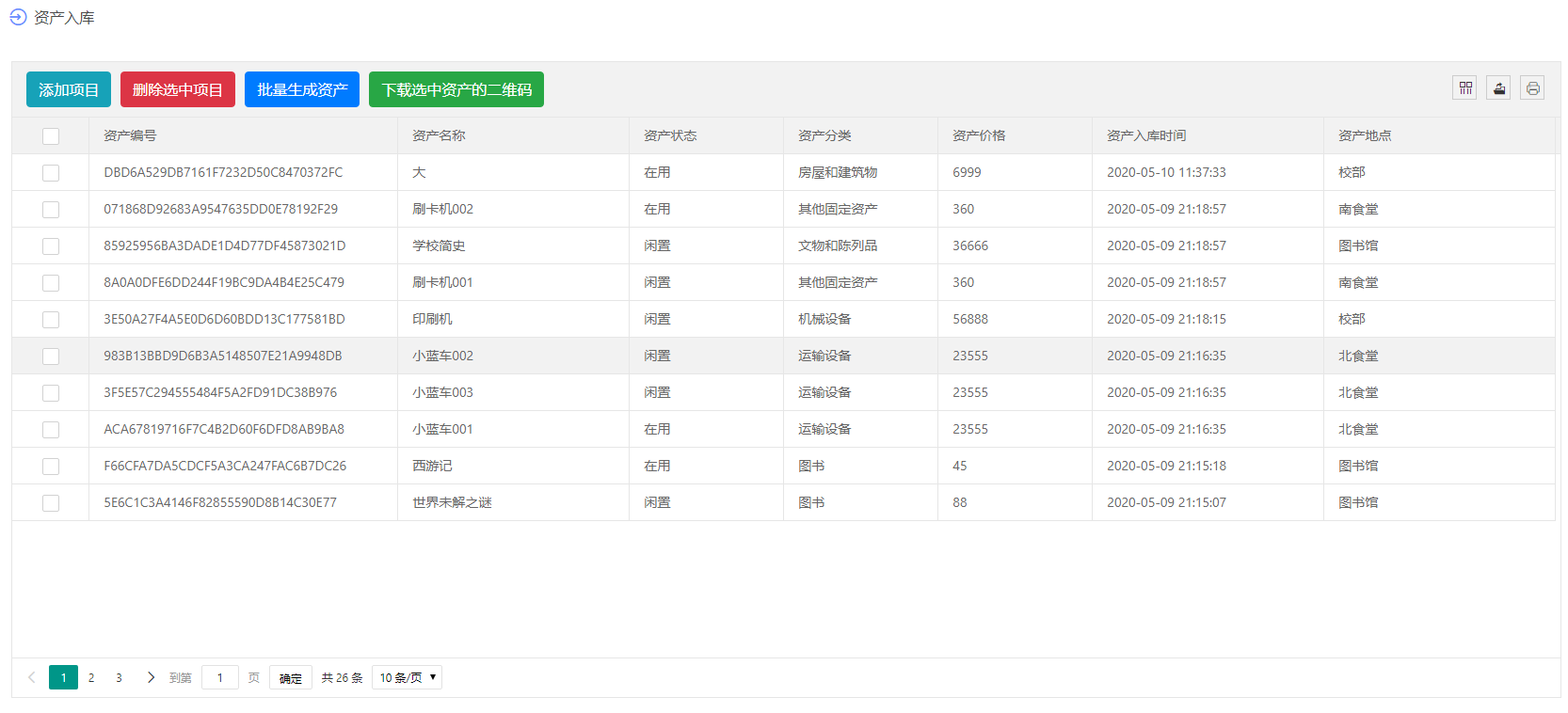


图5-1- 1 资产入库PC端界面

添加项目也就是资产登记，需要填写资产的相关信息，如图5-1-2所示：



图5-1- 2 添加资产界面

后端收到前端传递过来的数据后先对用户权限以及数据的唯一性与正确性进行判断，再用含有时间戳的随机数字字母组合并转换为全大写的32位md5字符串，实现代码如下：

private function createNo() {

$no = "";

$pattern = '1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLOMNOPQRSTUVWXYZ';

$returnStr = '';

for($i = 0; $i < rand(10,100); $i ++) {

$returnStr .= $pattern[mt\_rand(0, 61)];

}

$no = $returnStr . time();

$returnStr = '';

for($i = 0; $i < rand(10,100); $i ++) {

$returnStr .= $pattern[mt\_rand(0, 61)];

}

$no .= $returnStr;

return strtoupper(md5($no));

}

得到一个全新的资产编号后进行二维码的生成，实现代码如下：

public static function createQrcode($value) {

vendor('phpqrcode');

$errorCorrectionLevel = 'L';

$matrixPointSize = 5;

$filename = 'qrcode/'.$value.'.png';

\QRcode::png($value, $filename, $errorCorrectionLevel, $matrixPointSize, 2);

}

最后利用已经定义好的数据库模型将数据存储到数据库中。

资产变更也就是对资产信息进行修改，步骤和添加资产类似，首先也需要判断用户权限和数据的唯一性和正确性，然后利用已经定义好的数据库模型将数据进行修改，实现代码如下：

public function update() {

if (Utils::userAlreadyLogin() && $this->powerTrue()) {

$co = $this->checkOnlyAndNull();

if ($co['code'] == 0) {

return json\_encode($co);

}

$obj = new Asset(Param::get("asId"));

$obj->update($this->getFields());

$commonId = Param::get("asId");

Logging::updateAs($commonId);

return json\_encode(Utils::returnCode(1));

} else {

$this->error(Constant::PLEASELOGIN, Constant::LOGINPATH);

}

}

删除资产以及批量删除资产只需要向资产id发送给后台，后台验证权限后利用已经定义好的数据库模型将数据进行删除。

批量生成资产是批量向数据库写入未命名资产，后利用移动端进行录入。

批量下载二维码需要先对选择的资产的二维码进行打包，也就是生成zip文件，实现代码如下：

private function createZip($files) {

$filename = "download/".md5(time()).".zip";

$zip = new \ZipArchive();

$zip->open($filename, \ZIPARCHIVE::CREATE);

foreach($files as $file){

$zip->addFile($file, basename($file));

}

$zip->close();

return '\/'.$filename;

}

然后将打包好的zip文件路径返回给前端，前端在新标签页进行跳转。

资产的借用与归还，借用的时候需要选择借用人以及借用的资产，并设置借用的时间，关于不同权限会有所不同，普通用户不可以选择借用人，只能是自己借用，并且需要管理员审核，如图5-1-3所示：



图5-1- 3 普通用户借用资产页面

管理员用户可以选择借用的人员，并直接借用成功，不需要管理员审核，如图5-1-4所示：

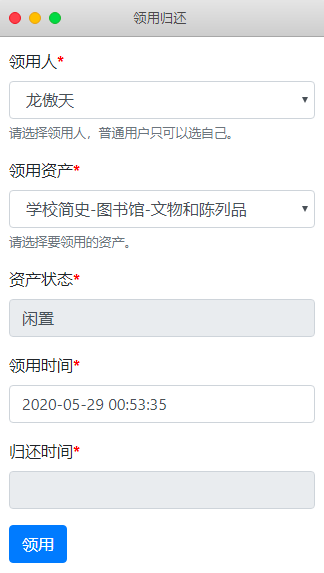


图5-1- 4 管理员借用资产页面

借用与归还资产只需要对数据库进行添加和修改操作即可。

移动端不可以直接添加资产，只可以先扫码再对资产信息进行填写，需要配合PC端批量生成资产使用，实现代码如下：

def assetImport(tornadoSelf, obj):

Ljsmysql.table("fams\_asset").where("as\_id", obj['as\_id']).update({

"as\_name": obj['as\_name'],

"as\_price": obj['as\_price'],

"cate\_id": obj['cate\_id'],

"as\_image": obj['as\_image'],

"as\_local\_id": obj['as\_local\_id']

})

sendMsg(tornadoSelf, "assetImportSuccess")

## 资产盘点模块的实现

### 功能概述

资产盘点是用户对自己借用的资产或管理员分配的盘点任务中的资产进行盘点。管理员创建盘点单之后，指定盘点人员和管理员可以查看盘点任务，用户将盘点单对应的资产进行扫码拍张以及GPS定位。盘点结束后将盘点单转至审批模块。

### 模块实现

资产盘点功能需要PC端和移动端进行配合，由于移动端界面大小有限，进行资产选取会不方便，所以盘点单的创建需要在PC端进行，而且只有管理员身份可以创建盘点单，盘点单创建界面如图5-1-5所示：



图5-1- 5 创建盘点单界面

可以指定盘点人员，也可以不指定盘点人员，不指定就是所以管理员进行盘点，勾选‘添加盘点人员领用的资产到盘点单’会自动将盘点人员所借用的资产添加到盘点单中，实现代码如下：

if ($data['u\_id'] > 0 && Param::get("checked")) {

$assets = Borrowlend::getAllbyUId($data['u\_id']);

foreach ($assets as $asset) {

Check\_content::insert([

"c\_id" => $commonId,

"as\_no" => $asset['as\_no']

]);

}

}

创建盘点单之后可以向盘点单中添加待盘点的资产，前端将选中的资产id发送给后端，后端将其写入到数据库中，实现代码如下：

public function inserts($id) {

if (Utils::userAlreadyLogin() && $this->powerTrue()) {

$datas = json\_decode(Param::get("data"), true);

foreach ($datas as $data) {

Check\_content::insert([

"c\_id" => $id,

"as\_no" => $data['as\_no']

]);

}

return json\_encode(Utils::returnCode(1));

} else {

$this->error(Constant::PLEASELOGIN, Constant::LOGINPATH);

}

}

创建后的盘点单可以在移动端中查看，移动端中执行盘点任务，先扫描待盘点的资产二维码，然后获取手机GPS定位，查看样图，拍照上传图片，后台进行图片审核，实现代码如下：

def calculate(image1, image2):

# 灰度直方图算法

# 计算单通道的直方图的相似值

hist1 = cv2.calcHist([image1], [0], None, [256], [0.0, 255.0])

hist2 = cv2.calcHist([image2], [0], None, [256], [0.0, 255.0])

# 计算直方图的重合度

degree = 0

for i in range(len(hist1)):

if hist1[i] != hist2[i]:

degree = degree + \

(1 - abs(hist1[i] - hist2[i]) / max(hist1[i], hist2[i]))

else:

degree = degree + 1

degree = degree / len(hist1)

return degree

def classify\_hist\_with\_split(image1, image2, size=(256, 256)):

# RGB每个通道的直方图相似度

# 将图像resize后，分离为RGB三个通道，再计算每个通道的相似值

image1 = cv2.resize(image1, size)

image2 = cv2.resize(image2, size)

sub\_image1 = cv2.split(image1)

sub\_image2 = cv2.split(image2)

sub\_data = 0

for im1, im2 in zip(sub\_image1, sub\_image2):

sub\_data += calculate(im1, im2)

sub\_data = sub\_data / 3

最后交于管理员进行审核通过。

## 审批管理模块的实现

### 功能概述

在普通用户进行申请操作时，管理员需要审批，例如固定资产的借用，在管理员审核通过的时候会告知用户去哪里取或者派人统一派送。在普通用户进行资产的盘点之后，可以对盘点结果的异常情况进行审批。

### 模块实现

从上文可以看出审批管理主要分为借用归还资产的审批以及盘点的审批。这个模块需要管理员身份才能进入，借用归还审批的界面如图5-1-6所示：

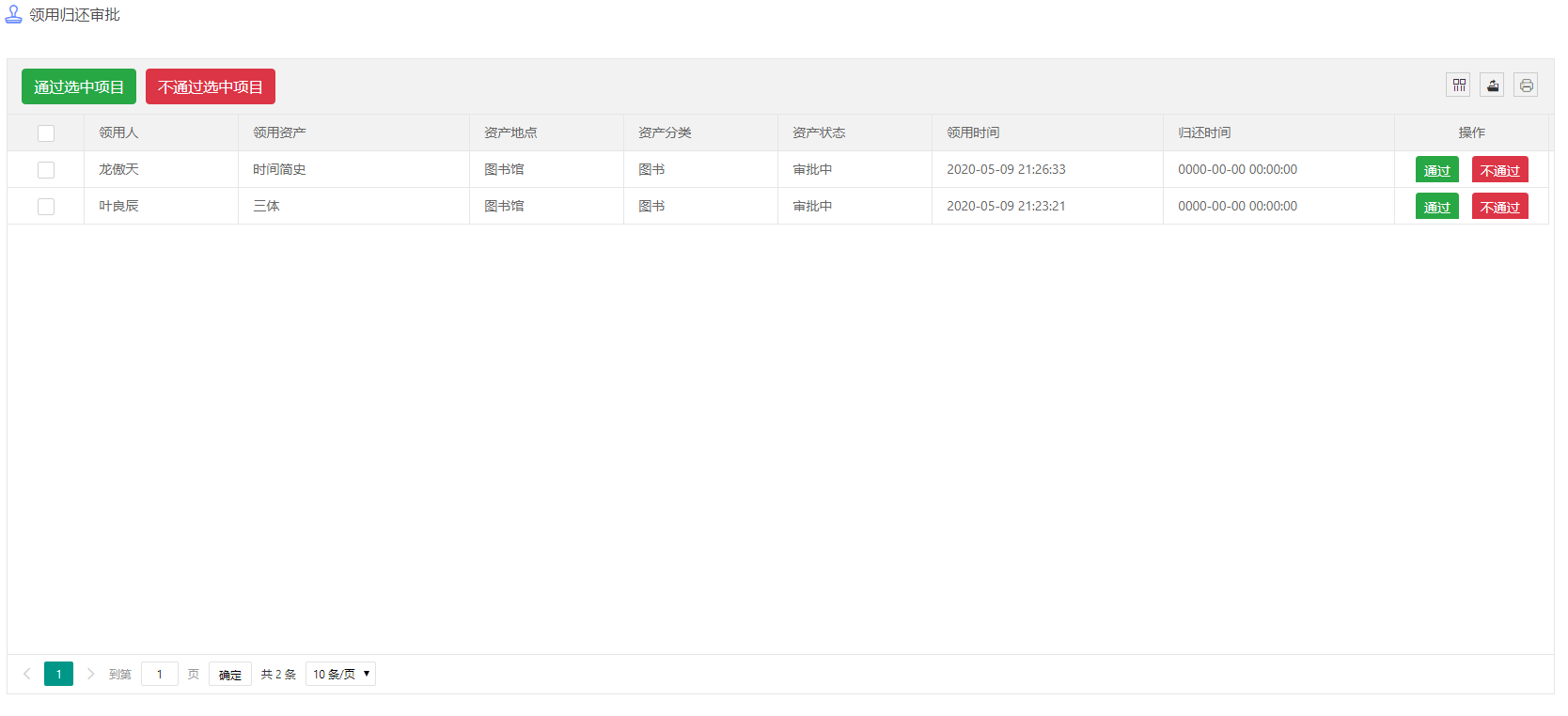


图5-1- 6 借用归还审批界面

这里的审批分为两种情况，一种为借用资产的审批，一种为归还资产的审批。借用资产的审批，通过的情况下需要将资产的状态设置为在用，借用归还表不变，实现代码如下：

public function pass() {

if (Utils::userAlreadyLogin() && $this->powerTrue()) {

$obj = new Borrowlend(Param::get("bl\_id"));

$sta = "XZ";

if ($obj->getL\_time() == "0000-00-00 00:00:00") {

$sta = "ZY";

} else {

$obj->setBl\_ok(1);

}

$obj = Asset::newByAs\_no($obj->getAs\_no());

$obj->setSta\_no($sta);

return json\_encode(Utils::returnCode(1));

} else {

$this->error(Constant::PLEASELOGIN, Constant::LOGINPATH);

}

}

不通过的情况下需要将资产状态设置为闲置，借用归还表相应记录设置为已完成。归还资产的审批，在通过的情况下需要设置资产状态为闲置，借用归还表相应记录设置为完成，实现代码如下：

public function notpass() {

if (Utils::userAlreadyLogin() && $this->powerTrue()) {

$obj = new Borrowlend(Param::get("bl\_id"));

$sta = "ZY";

if ($obj->getL\_time() == "0000-00-00 00:00:00") {

$sta = "XZ";

$obj->setBl\_ok(1);

}

$obj = Asset::newByAs\_no($obj->getAs\_no());

$obj->setSta\_no($sta);

return json\_encode(Utils::returnCode(1));

} else {

$this->error(Constant::PLEASELOGIN, Constant::LOGINPATH);

}

}

不通过的情况下，需要设置资产状态为在用，借用归还表不变。

盘点审批通过只需将盘点单状态设置为已完成即可，不通过则需提醒用户或者自己亲自去查看不通过的资产状态，判断是否可以通过。

## 系统管理模块的实现

### 功能概述

系统管理只有系统管理员可以进行操作，组织架构是对各高校各部门进行管理，方便分部门管理固定资产，也可以对单独部门设置管理员，普通用户和系统用户管理是对用户的统一管理，只是两者的操作权限不同。

### 模块实现

系统管理分为部门管理、职位管理、人员管理、用户管理。只有系统管理员才可以访问各个页面以及接口。其中部门管理采用无限级treeview来显示数据，如图5-1-7所示：



图5-1- 7 部门管理界面

除了根部门不可删除，其余部门都可删除，可以向每一个部门添加子部门，因为以自身的主键做外键，所以不可以选择自己作为上级部门。

由于职位管理中没有外键，只需要进行最基本的增删改查即可。

人员管理中有部门id和职位id两个外键，所以需要前端向后端请求部门表以及职位表的数据显示在下拉列表框中，前端Vue部分实现代码如下：

mounted() { // ajax请求数据 用于select框

axios

.post("/app/departmentManage/departmentsData")

.then(response => (this.deps = response.data))

.catch(function (error) {

console.log(error);

});

axios

.post("/app/positionManage/positionsData")

.then(response => (this.poses = response.data))

.catch(function (error) {

console.log(error);

});

}

用户管理需要对手机号码的准确性进行验证，实现代码如下：

if (!preg\_match("/^1[3456789]\d{9}$/", $uPhone)) {

return Utils::returnMsg(0, "uPhoneError");

}

其余部分只需要进行最基本的增删改查即可。

1. 系统实验与结果分析

## 系统测试的工具与环境

### 系统测试工具

对本文中高校固定资产管理系统性能方面的测试，主要采用了 Load Runner 测试工具来完成对该系统的测试。

### 系统测试环境

对系统进行环境的要求主要有以下几部分：

(1) Web Server(1台)：负责实现应用系统的负载均衡及管理。

硬件要求：1vCPU、2G 内存；

软件要求：CentOS7.4、Apache2.4。

(2) MySQL Server(1台)：作为系统的数据库服务器。

硬件要求： 1vCPU、2G 内存；

软件要求：CentOS7.4、MySQL5.7。

(3) 测试服务器(1台)：负责对系统平台进行测试。

## 系统测试流程

对本文中高校固定资产管理系统的测试流程依次是：设计并制定测试计划、完成测试准备工作、按照计划进行系统测试、分析并评价测试结果。

制定测试计划：根据高校固定资产管理系统的需求分析及系统功能进行测试用例的编写，同时按照测试目的制定出科学、合理、严格的测试计划。

测试准备工作：按要求完成对测试环境的软硬件配置，及其它相关准备工作等。

系统测试的实施：严格按照测试计划对高校固定资产管理系统进行系统测试工作，并随时对测试结果进行完整记录。

系统测试结果的分析和评价：将测试用例输入的预期结果和实际输出结果进行对比和分析，给出该系统的最终测试评价。

## 系统测试用例

### 资产管理功能测试用例

(1) 用例名称：资产管理。

(2) 测试环境：硬件：测试服务器；软件：操作系统、数据库。

(3) 测试过程：

① 用PC机登录PC端后台，进行添加资产操作，预期结果：table里出现新添加的记录；实际结果：和预期结果相符。

② 双击记录对资产信息进行修改，预期结果：table中对应记录随之改变；实际结果：和预期结果相符。

③ 双击后删除单条记录，预期结果：table中该记录消失；实际结果：和预期结果相符。

④ 多选后批量删除记录，预期结果：table中选择记录消失；实际结果：和预期结果相符。

⑤ 点击批量生成资产，输入10，预期结果：table中出现10条未命名资产；实际结果：和预期结果相符。

⑥ 多选后下载选中二维码，预期结果：弹出下载提示框；实际结果：和预期结果相符。

(4) 测试结果评估：和预期的结果完全一致，测试通过。

### 资产盘点功能测试用例

(1) 用例名称：资产盘点。

(2) 测试环境：硬件：测试服务器；软件：操作系统、数据库。

(3) 测试过程：

① 用PC机登录PC端后台，进行创建盘点单操作，预期结果：table里出现新添加的记录；实际结果：和预期结果相符。

② 双击记录对查看盘点单详情，添加待盘点资产，预期结果：盘点单详情table中多出选中资产；实际结果：和预期结果相符。

③ 多选后批量删除记录，预期结果：table中选择记录消失；实际结果：和预期结果相符。

④ 用手机登录移动端，查看盘点单，预期结果：资产盘点栏目中出现盘点单；实际结果：和预期结果相符。

⑤ 在移动端资产盘点中扫描对应资产二维码，拍照上传，预期结果：盘点单进入审核状态；实际结果：和预期结果相符。

(4) 测试结果评估：和预期的结果完全一致，测试通过。

### 审批管理功能测试用例

(1) 用例名称：审批管理。

(2) 测试环境：硬件：测试服务器；软件：操作系统、数据库。

(3) 测试过程：

① 用PC机登录PC端后台，进入借用归还审批，通过审批，预期结果：table对应记录消失；实际结果：和预期结果相符。

② 在借用归还审批中，不通过审批，预期结果：table对应记录消失；实际结果：和预期结果相符。

③ 盘点审批中，通过盘点，预期结果：盘点单状态变为已完成；实际结果：和预期结果相符。

④ 盘点审批中，部门通过盘点，预期结果：盘点单状态变为进行中；实际结果：和预期结果相符。

(4) 测试结果评估：和预期的结果完全一致，测试通过。

### 系统管理功能测试用例

(1) 用例名称：系统管理。

(2) 测试环境：硬件：测试服务器；软件：操作系统、数据库。

(3) 测试过程：

① 用PC机使用系统管理员用户登录PC端后台，在部门管理中添加子部门，预期结果：treeview中对应分支出现新添加的部门；实际结果：和预期结果相符。

② 在部门管理中删除部门，预期结果：treeview中对应部门消失；实际结果：和预期结果相符。

③ 在部门管理中修改部门，预期结果：再次进入页面对应部门信息发生改变；实际结果：和预期结果相符。

④ 在人员管理中添加人员，预期结果：table中出现添加的人员；实际结果：和预期结果相符。

⑤ 在人员管理中修改人员，预期结果：双击对应记录发生改变；实际结果：和预期结果相符。

⑥ 在人员管理中删除人员，预期结果：table中对应记录消失；实际结果：和预期结果相符。

⑦ 在职位管理中添加职位，预期结果：table中出现添加的职位；实际结果：和预期结果相符。

⑧ 在职位管理中修改职位，预期结果：双击对应记录发生改变；实际结果：和预期结果相符。

⑨ 在职位管理中删除职位，预期结果：table中对应记录消失；实际结果：和预期结果相符。

(4) 测试结果评估：和预期的结果完全一致，测试通过。

### 系统压力测试

由 200 个用户端同时登录高校固定资产管理系统并对其进行操作，得出系统此时的响应时间已经控制在 3 秒以内。

## 系统测试结果分析

对本次高校固定资产管理系统的测试主要是针对该系统 4个功能模块进行的。重点完成了对系统 UI、功能、应用、以及性能的测试。

(1)系统 UI 测试：该系统的界面设计简洁明了，操作流程清晰简单，界面丰富多彩，满足了系统用户对系统操作的基本要求。

(2)系统功能测试：本文中的高校固定资产管理系统主要包含了 4 个功能模块，本次测试对这 4 个功能及其子功能都分别进行了测试，确保了系统各个功能的正确性和完整性。

(3)系统应用测试：该系统可以在最短时间内读取到高校固定资产相关业务工作需求的汇总数据和明细数据，能够减轻用户对数据的统计工作。

(4)系统性能测试：通过对本文高校固定资产管理系统的测试，保证了该系统在实际应用过程中能够安全稳定的运行。依据对该系统运行情况所进行的长时间监测情况来看，该系统对业务处理的均速保持在每秒钟 2000条左右，失误率为 0%，完全符合设计初期对系统性能的预期效果。此外，该系统在最大并发负载情况下仍然能够保证系统运行保持良好性能，系统响应时间和 CPU 都保持在最佳状态。系统整体性能都完全符合预期要求。

1. 结　论

本文设计的高校固定资产管理系统项目，从系统分析、系统架构到最终的系统实现耗时较长，在开发过程中，主要采用了软件工程中的相关理论知识，本文的研究过程是一个典型的理实一体化的过程，并最终将理论成果转化为实践产物的过程。

通过与指导教师的共同研究和设计，最终开发出了符合高校对于固定资产管理实际需求的软件系统，帮助高校实现了利用扫描条码进行固定资产管理和信息登记，同时对高校固定资产发生变化时能够及时完成相关信息的变更和记录工作。此外，利用本系统还可以帮助财务部门及时得到高校固定资产的所有折旧信息，并对固定资产的利用情况进行统计和分析，进而保证高校固定资产的利用率，最大程度的减少对固定资产的浪费。

系统开发完成后，根据系统常用的测试方法及理论原理，对高校固定资产管理系统进行功能测试和性能测试。该结果充分说明了所开发的资产系统安全可靠，能够保证学校资产的高效管理。

在固定资产管理工作进入正轨之后，还可以跟据高校的实际情况进一步去完善和拓展系统的相关功能，如服务端智能监控系统，可以实时监控各资产以及系统运行的情况，并通知管理员采取相关措施，一旦大型设备发生异常，管理人员能够第一时间获知，并及时处理，大大降低资产风险。

参考文献

1. 康健明,张旭.基于移动端的高校资产管理系统的设计与应用[J].中国医学装备,2019,16(06):132-137.
2. 杨伟元.高校固定资产管理难点及解决模式研究——基于RFID自动识别技术[J].经贸实践,2018,(01):81.
3. 陈瀚. 高校固定资产管理系统设计与开发[D]. 江西: 东华理工大学, 2017.
4. 刘雯萱, 彭琼. 基于高等学校固定资产管理系统的研究[J]. 无线互联科技, 2020, 17(07): 57-58.
5. 刘晶. 高校固定资产管理系统的设计与实现[D]. 大连: 大连理工大学, 2018.
6. 张慕然. 高校固定资产管理系统设计与实现[D]. 长江: 长江大学, 2019.
7. 姜宇飞. 设备资产管理的设计与实现[D]. 青岛: 青岛大学, 2019.
8. 王宁邦, 徐博, 夏百川, 夏娜, 邰永航, 李琼. 一种颜色直方图计算相似度的资产图片核对算法[J]. 智能计算机与应用, 2018, 8(02): 59-62, 67.
9. 张华. 基于非对称加密算法的QR二维码[J]. 电子技术与软件工程, 2018(05): 29-29.
10. 张子杰. 固定资产动态折旧模型研究[J]. 企业改革与管理, 2019(21): 116-116, 126.

# 致　谢

感谢我的导师邵中老师。在老师的悉心指导和严格要求下完成了本论文，导师的心血和汗水凝聚在本论文的各个方面：课题选择、方案论证到具体设计和实现。在四年的大学学习和生活期间，导师的精心教导让我更好的掌握和运用专业知识，并在论文中得以体现，不仅如此，日常的学习中也始终能够感受到导师无私的关怀。在此向邵中老师表示深深的感谢和崇高的敬意。