归 档 号：

**武汉工商学院毕业论文(设计)**

**学院： 信息工程学院**

**专业：计算机科学与技术 年级： 2015级**

**学生：贺英杰 学号：15401032**

**指导教师： 朱捷 职称: 讲师**

**题目: 基于C#的公司资产管理系统设计与实现**

**2019年5月20日**

**武汉工商学院**

**本科毕业论文（设计）原创性声明**

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

年 月 日

目 录

摘 要 I

关键词 I

Abstract II

Key words II

1 绪论 1

1.1 选题背景 1

1.2 选题目的 1

1.3 选题意义 1

2 可行性分析 2

3 系统需求分析 3

3.1 系统功能概述 3

3.2 功能需求 3

3.3 界面需求 3

3.4 运行环境需求 3

3.5 其他需求（若有） 3

4 系统总体设计 4

4.1 系统架构设计 4

4.2 数据库设计 4

4.2.1 数据库概念结构设计 4

4.2.2 逻辑结构设计 4

4.2.3 数据库表的设计 4

5 系统详细设计与实现 5

5.1 XXX功能模块 5

5.1.1 XXX功能模块概述 5

5.1.2 XXX功能模块流程设计 5

5.1.3 XXX功能模块核心代码 5

5.2 XXX功能模块 5

5.2.1 XXX功能模块概述 5

5.2.2 XXX功能模块流程设计 5

5.2.3 XXX功能模块核心代码 5

6.1 XXX功能测试 6

参考文献 7

致谢 8

基于C#的公司资产管理系统设计与实现

摘 要

如今，每个企业或者事业单位基本上都有自己的资产管理系统，一个完善且安全的资产管理系统是任意企业或者事业单位必须具备的标准，它的内容对于企业或事业单位的财产部门管理者来说非常的重要。

本文具体的介绍了公司资产管理系统开发设计的整个过程，包括：前期的系统需求分析、开发平台及环境搭建、中期的数据库数据表的制作、系统主要框架的设计、后台代码的编写以及后期的系统所有功能的测试。本系统中主要有资产信息的查询及浏览、员工信息管理、部门信息管理、资产新增、资产异动、资产报废、资产维修、资产分类管理、打印资产、导出资产到Excel、用户的权限管理以及个人信息管理等功能。

项目中主要使用了VisualStudio2017作为开发工具，Apache Tomcat 6.0作为应用服务器，SQL Server 2008R2作为后台数据库，采取了三层架构（表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL））模式进行系统架构的设计，运用了WPF实现客户端美化及特效处理。

关键词**：**资产管理；固定资产；c#；wpf；sqlserver

Design and Implementation of Company Asset Management System Based on C#

Abstract

Nowadays, every enterprise or institution basically has its own asset management system. A perfect and secure asset management system is a standard that any enterprise or institution must have. Its content is for the property department manager of the enterprise or institution. It is very important.

This paper introduces the whole process of the development and design of the company's asset management system, including: early system requirements analysis, development platform and environment construction, medium-term database data table production, system main framework design, background code writing and later Testing of all features of the system. The system mainly includes asset information inquiry and browsing, employee information management, department information management, asset addition, asset transaction, asset retirement, asset maintenance, asset classification management, printing assets, exporting assets to Excel, user rights management, and Personal information management and other functions.

The project mainly uses VisualStudio2017 as the development tool, Apache Tomcat 6.0 as the application server, and SQL Server 2008R2 as the background database. It adopts a three-tier architecture (presentation layer (UI), business logic layer (BLL), data access layer (DAL)). The model is designed for system architecture, and WPF is used to implement client beautification and special effects processing.

**Key words:** Asset management; fixed assets; c#; wpf; sqlserver

# 绪论

## 1.1 选题背景

随着网络科技的发展，计算机的应用已经渗透到各个领域，企业占有、使用的固定资产规模急剧膨胀，其构成日趋复杂，管理难度越来越大。尤其是随着学校、企业内部推行的后勤、财务、人事、分配等改革的深化，对固定资产管理工作不断提出新要求。但是，多年来固定资产管理工作一直是高校、企业管理的一个薄弱环节，管理基础工作不够规范，资产安全控制体系尚不完善，家底不清、帐实不符、资产流失的现象依然存在。因此，必须从有力支撑发展改革的高度出发，对固定资产管理工作进行系统的规划和建设，运用现代信息技术真正实现管理水平上的飞跃

随着互联网的普及和固定资产管理信息系统的更趋向于成熟化，会有越来越多的企业需要资产管理系统。资产管理系统是一个企业管理者在管理过程中不可缺少的一部分，对于企业的决策者和管理者来说都至关重要，使用资产管理系统对各类资产进行管理具有手工管理无法比拟的优点，固定资产管理的操作自动化和信息的电子化，全面提高的固定资产管理水平的可行性

采用资产管理系统，它给企业管理带来了明显的经济效益和社会效益。极大提高了管理者的工作效率，大大减少了以往管理流程的繁琐，杂乱，周期长的弊端。基于固定资产管理的全面自动化，可以减少设备购入、使用、异动以及处理时的漏洞，可以节约不少管理开支，增加企业收入。

## 1.2 选题目的

公司每年都投入大量的金钱来购置各种资产，但是随着时间的推移，很难及时、清楚地知道单位及下属机构拥有多少资产，它们分布在那些部门以及存放在何处、谁在使用、状况如何。并且固定资产的利用效率低，重复购置现象严重。员工离职或工作变动时资产交接不完整，经常存在无法快速完整的查清此人手中保管的资产，从而造成资产的流失。责任人不明确，核查和盘点工作量大，出错率较高，给资产管理部门带来了很多工作难题和错误。经常存在帐物不符、帐面价值与实际价值有很大出入等现象。

传统的手工管理方式无法解决以上所有的问题。我在充分研究分析国有企业、私营企业、行政单位等对资产管理业务需求的基础上，有针对性的开发出了资产管理系统，把编码技术引入固定资产管理，有效解决了固定资产的管理难题，使单位更轻松、有效地管理固定资产，不仅为单位节约了大量的资金，而且为提高单位的生产经营效益提供了重要的后勤保障。

目前高校和企业的固定资产具有价值高、使用周期长、使用地点分散、管理难度大等特点。企事业、机关等的固定资产如果使用人工手动管理，将面临管理起来工作量大、固定资产的增加、减少、转移、维修、闲置、报废和调拨等不易管理或者监管。因此，使用计算机通过资产管理系统对固定资产进行合理管理，提高了管理水平，提高了工作效率，提高了企业事业单位的经济效益。因此我才选择资产管理系统作为毕业论文的研究课题

## 1.3 选题意义

随着科学技术的不断提高，计算机科学技术日渐成熟，其强大的功能对人们留下的深刻的认识，它已经进入人类社会的各个领域并发挥着越来越重要的作用。

作为计算机应用的一部分，使用计算机对固定资产进行管理，具有手工管理所无法比拟的优点。例如：检索迅速、查找方便、可靠性高、保密性好、成本低等。这些优点能够极大提高固定资产管理的效率，也是正规的科学的管理的重要条件。资产管理系统是通过对固定资产日常管理所产生的各种数据，根据资产管理要求，对数据进行分类、统计、分析、汇总，建立的程序化规范的资产管理管理系统，包括了一般的固定资产特性、各种状态和行为的管理，为固定资产管理提供了一个现代化的管理手段，这一个手段可以强化规范对固定资产的管理。而且可以方便的让财务、审计等管理部门的介入，形成严格的监督、约束机制。

# 2 可行性分析

现在，如果要开发一个资产管理系统工具非常多，目前我国使用最多、最流行的开发工具有：微软公司的VisualStudio适用于C++、Python、C#等语言的开发工具，微软还有一个比较强大的工具VisualStudioCode适用于几乎任何语言的开发工具，被网友称为全宇宙最强代码编辑器，可以根据用户需求安装和定制各种插件。我经常用的一个代码编辑器Sublime,既小巧又强大。Orwell 公司的DevC++是适用于C语言和C++的代码编译器。JetBrains公司出的PyCharm是适用于Python语言开发的编译器，这个公司还有一个比较有名的Java编译器IDEA在业界公认为最好用的Java开发工具之一。这些开发工具都非常的好用，都是独具匠心的产品。从我的开发工具需求出发，首先我要做的是客户端（CS）程序,选择的语言是C#，那么我选择VisualStudio作为我开发的核心工具，这个开发工具满足我的开发工具需求。

（1）VisualStudio开发软件技术介绍

我的资产管理系统是在VisualStuadio开发软件中设计出来的。VisualStuadio是美国微软公司的开发工具包的系列产品。用VisualStuadio所写的.NET项目支持微软的Windows环境，资产管理系统是在Windows环境下运行的，VisualStuadio是目前最流行的Windows平台应用程序的集成开发环境。

我使用的是VisualStuadio2017版本，根据官网介绍VisualStuadio2017是一个面向任何开发者的开发工具，用户在安装的时候可以根据需求选择安装比如.NET桌面开发、C++桌面开发、ASP.NET和Web开发、Python开发、Node.js开发、Unity游戏开发、C++的linux开发、移动开发等。对于初学者也能快速上手使用。

（2）SQL Server数据库技术

SQL Server是由微软开发的关系数据库管理系统，具有使用方便可伸缩性好与相关软件集成程度高等优点，可跨越从运行Microsoft Windows 98 的膝上型电脑到运行Microsoft Windows 2012 的大型多处理器的服务器等多种平台使用。Microsoft SQL Server 是一个全面的数据库平台，使用集成的商业智能 (BI)工具提供了企业级的数据管理。Microsoft SQL Server 数据库引擎为关系型数据和结构化数据提供了更安全可靠的存储功能，使您可以构建和管理用于业务的高可用和高性能的数据应用程序。

SQL语句可以用来执行各种各样的操作，例如更新数据库中的数据，从数据库中提取数据等。目前，绝大多数流行的关系型[数据库管理系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F)，如Oracle,Sybase,Microsoft SQL Server,Access等都采用了SQL语言标准。虽然很多数据库都对SQL语句进行了再开发和扩展功能等修改，但是包括Select,Insert,Update,Delete,Create,以及Drop在内的标准的SQL命令仍然可以被用来完成几乎所有的数据库操作。

（3）WPF框架技术

WPF是微软开发的用于Windows程序的界面框架，运行环境基于.NET Framework3.0及以上版本。它提供了一套用于编写界面的语言XAML，是一种全新的多媒体交互用户图形界面，可以开发各种交互特效，真正做到了分离界面设计人员和业务开发的工作。

使用 WPF，你可以创建适用于 Windows 且具有非凡视觉效果的桌面客户端应用程序。WPF 的核心是一个与分辨率无关且基于矢量的呈现引擎，旨在充分利用现代图形硬件。WPF 通过一套完善的应用程序开发功能对该核心进行了扩展，这些功能包括可扩展应用程序标记语言 (XAML)、控件、数据绑定、布局、二维和三维图形、动画、样式、模板、文档、媒体、文本和版式。 WPF 包含在 .NET Framework 中，因此你可以生成整合其他 .NET Framework 类库元素的应用程序。

（4）C/S技术介绍

CS即Client/Server（客户机/服务器）结构，C/S结构在技术上很成熟，它的主要特点是交互性强、具有安全的存取模式、网络通信量低、响应速度快、利于处理大量数据。但是该结构的程序是针对性开发，变更不够灵活，维护和管理的难度较大。通常只局限于小型局域网，不利于扩展。并且，由于该结构的每台客户机都需要安装相应的客户端程序，分布功能弱且兼容性差，不能实现快速部署安装和配置，因此缺少通用性，具有较大的局限性。客户端程序是独立设计，所以可以实现个性化，安全性也比一般的BS结构程序要好，BS不能对本地文件进行特殊操作，比如删除本地文件，所以所有杀毒软件都是CS架构的。

（5）三层架构技术

三层架构是指逻辑上的三层，是一般在客户端和数据库之间的数据处理加入的一个中间层，也叫组件层。UI层就是表现层，主要指用户交互的界面。用于接收用户输入的数据和显示处理后用户需要的数据。BLL层是业务逻辑层，UI层和DAL层之间的桥梁。实现业务逻辑。业务逻辑具体包含：验证、计算、业务规则等等。DAL层是数据访问层， 与数据库打交道。主要实现对数据的增、删、改、查。将存储在数据库中的数据提交给业务层，同时将业务层处理的数据保存到数据库。当然这些操作都是基于UI层的。用户的需求反映给界面（UI），UI反映给BLL，BLL反映给DAL，DAL进行数据的操作，操作后再一一返回，直到将用户所需数据反馈给用户。

每一层都各负其责，那怎么把三层联系起来，其实还有一个实体层，它不属于三层架构中的任何一层，但是它是必不可少的一层。它实现了面向对象思想中的“封装”，并且贯穿于三层，在三层之间传递数据，准确的说是实体层贯穿于三层之间，来连接三层。在数据库中每张数据表对应一个实体，每一层之间的数据传递（单向）是靠变量或实体作为参数来传递的，这样就构造了三层之间的联系，完成了功能的实现。

（6）.NET技术

NET是微软推出的一种新的软件运行平台，它包括一整套应用范围广泛的技术，旨在迎接Java所带来的挑战。.NET给网站管理系统开发带来了很多新的技术，尤其是它的“混合语言”开发方式，是第一个支持多种语言混合开发的软件运行平台，下一代Windows--LongHorn的核心将会看到.NET的身影。如果是学过Java，那么您会发现，.NET与Java实在是太象了。尤其是C#，如果给你一段代码，有的时候还真难分辨出是用Java还是C#写的。.NET框架包括C++, VB.NET和C#等一系列语言；与JAVA虚拟机类似的一套运行时环境；以及一套倾向与WINDOWS体系的API接口。其中的运行时环境可能存在于一个浏览器、或是一个WEB SERVER、或是在操作系统中。将来也许在SQL SERVER中也可能存在这样的运行时环境。另外需要提及的是微软的SOAP协议，它在继承了DCOM的一些特性的基础上发展起来，基于XML格式通过HTTP进行传输。

# 3 系统需求分析

## 3.1 系统功能概述

公司资产管理系统包括资产管理、审核资产、部门管理、类别管理和用户管理五大模块，其中资产管理里面又包括查找、导出、打印、增加、修改、异动、维修、报废，审核资产里面能审核资产新增、修改、异动、维修、报废，部门管理里面能增、删、改部门信息，类别管理可以增删改资产类别，用户管理里面包括查询、修改、权限、禁用或启用等。具体功能图如下：

图3-1 系统功能结构图

## 3.2 功能需求

（1）用户登录注册:对于第一次登录的用户，首先需要注册才能使用，进入系统的注册页面，注册成功后返回登录界面。登录之后才能使用系统。

（2）资产查询:用户登录成功进入系统主页面，就可以看到所有固定资产，流动资产等详细信息。并且还可以根据条件进行筛选、查找固定资产。

（3）资产导出Excel:可以把查询到的资产直接导出到Excel进行处理

（4）资产打印:对于打印功能我需要一张数据库表来添加需要打印的资产，用户可以把资产添加到打印表，然后在系统里可以查看这张打印表，把不需要的打印的资产从打印表删除。点击打印就能打印资产标签。

（5）资产新增:对资产实现添加操作，资产的添加方式有很多种，比如购买、初盘、盘盈、转入、融资租赁、直接投资等，然后填写资产详细信息。为了区分是否固定资产，在填写资产详细信息时需要选择是否低值易耗品，低值易耗品属于流动资产。

（6）资产修改:修改资产可以对除了资产类别和资产编码以外的任何资产信息进行修改，但是需要管理人员确认修改申请。

（7）资产异动：对资产进行异动，异动到另外一个部门或者地点。填写异动单详细信息，就可以申请异动资产。

（8）资产维修:对于损坏的资产，可以对资产建立报修单，填写报修单的详细信息，比如故障的原因，申请报修的报修人，资产的编号名称。还可以看到该资产的维修记录。

（9）资产报废：对于不能使用的资产可以申请资产报废，报废原因有很多种，比如盘亏，调出，非常损失等。填写报废理由等信息后就能申请报废。

（10）资产类别管理：用户可以对资产的一级类别和二级类别进行增删改查。

（11）审核资产：审核资产可以对资产新增、修改、维修、异动、报废等操作进行确认审核。

（12）管理部门：管理部门可以对公司部门进行管理，对事业部和部门进行增删改查

（13）用户管理：管理用户可以对该系统的所有用户进行查找、修改、禁用、权限管理等操作。修改权限可以修改用户可以访问的功能，对于权限可以建立权限菜单

（14）个人信息管理：个人信息管理可以对当前使用用户的信息进行修改和查看。

## 3.3 界面需求

界面需要用户WPF来进行设计，可以切换不同颜色的主题，交互行为流畅。采用左边功能菜单，右边功能界面的布局方式。

## 3.4 运行环境需求

需要.net 4.0。Vista和Win7有自带，[xp](https://www.baidu.com/s?wd=xp&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao) [SP2](https://www.baidu.com/s?wd=SP2&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao) 以上只需安装.net 4.0就可以运行。

WinXP SP2及以上版本系统只需要安装.net4.0就能运行。内存1G以上，Atom级以上CPU。

# 4 系统总体设计

## 4.1 系统架构设计

此部分描述系统的架构设计，给出系统架构图。

公司资产管理系统中架构主要分为五层：

1. 业务逻辑层：主要用来表示数据的逻辑处理。
2. 数据访问层：主要用于控制数据的流程问题，采用ADO.NET来实现的。
3. 数据访问帮助层：主要用于安全访问数据库和格式化数据库字段，构建数据库条件语句。

（4）表现层：主要用于显示系统的界面，便于用户进行交互操作，主要用WPF来实现的。

（5）公共类层：加密解密类，配置文件读写类，生成二维码类等。

主要的框架结构如下图4-1所示。



图4-1系统总体架构图

## 4.2 数据库设计

### 4.2.1 数据库概念结构设计

概念模型是数据库系统的核心和基础。它能够抽象的表现出每一个实体与属性之间的相互依赖关系，通过数据库的这种概念结构，用户可以很方便合理的设计出自己满意的数据库表[2]。概念模型的描述工具通常是使用E-R 模型图。

（1）资产实体属性图

资产信息里面包括有资产ID,编号，名称，一级类别，二级类别，规格型号，品牌，生产厂家，计量单位，使用状态，事业部，部门，保管人员，增加方式，原值，出厂日期，购置日期，入账日期，折旧方法，有限年份，存放地点，备注，使用人员，残值率，是否耗材，资产状态，审核状态等属性，其具体的实体属性图如图4-2所示。



图4-2 资产实体属性图

（2）添加方式实体属性图

添加方式里面包括ID,排序ID,名称，添加时间，其具体的实体属性图如图4-3所示。



图4-3 添加方式实体属性图

（3）资产异动实体属性图

资产异动里面包括异动ID，资产ID,异动人账户，异动人，异动日期，转移人，批准人，异动事业部，异动部门，异动地点，增加方式，备注，异动状态，添加时间，其具体的实体属性图如图4-4所示。



图4-4 资产异动实体属性图

（4）资产报废实体属性图

资产报废里面包括报废ID，资产ID，申请人账户，申请人，减少方式，减少日期，报废理由，报废状态，添加时间，其具体的实体属性图如图4-5所示。



图4-5 资产报废属性图

（5）部门实体属性图

事业部里包括部门ID,排序ID,名称，添加时间，所属事业部ID，其具体的实体属性图如图4-6所示。



图4-6 事业部属性图

（6）折旧方法实体属性图

折旧方法里包括方法ID,排序ID,名称，添加时间，其具体的实体属性图如图4-7所示。



图4-7 折旧方法实体属性图

（7）事业部实体属性图

事业部里包括事业部ID,排序ID，名称，添加时间，其具体的实体属性图如图4-8所示。



图4-8 事业部实体属性图

（8）资产一级类别实体属性图

资产一级类别里包括一级类别ID，排序ID，名称，添加时间，其具体的实体属性图如图4-9所示。



图4-9 资产一级类别实体属性图

（9）打印列表实体属性图

打印列表里包括打印ID，资产ID，添加人，状态，添加时间，其具体的实体属性图如图4-10所示。



图4-10 打印列表实体属性图

（10）资产二级类别实体属性图

资产二级类别里包括二级类别ID，排序ID，名称，所属一级类别ID，单位ID，单位，有限年份，剩余价值率，添加时间，其具体的实体属性图如图4-9所示。



图4-11 资产二级类别实体属性图

（11）资产维修实体属性图

资产维修里面包括维修ID，资产ID，资产编码，维修人账户，维修人名称，维修电话，维修地址，维修理由，维修资产名，规格型号，事业部ID,部门ID,维修时间，维修状态，处理时间，添加时间，其具体的实体属性图如图4-11所示。



图4-12 资产维修实体属性图

（12）单位实体属性图

单位里面包括单位ID，排序ID，名称，添加时间，其具体的实体属性图如图4-12所示。



图4-13 单位实体属性图

（13）用户实体属性图

用户里面包括用户ID，账户，密码，事业部ID，部门ID，部门名称，姓名，职称，电话，邮箱，地址，是否启用，添加时间，修改时间，能否管理资产，能否添加资产，能否审核新增，能否审核异动，能否审核维修，能否审核报废，能否批量异动，能否管理我的资产，能否管理事业部，能否管理部门，能否管理一级类别，能否管理二级类别，能否管理用户，其具体的实体属性图如图4-13所示。



图4-14 用户实体属性图

（15）系统整体E-R图

公司资产管理系统整体E-R 图具体分析如下： 一个用户可以管理多个资产信息，所以管理员用户与资产之间是1:m 的关系，一门课程又可以被多个学员所选，所以课程与学员之间是1:n 的关系，又由于1个课程可以划分1个班级，所以类型与文件之间是一对一的关系，一个专业里面包含了多门课程信息，所以专业与课程之间是一对多的关系，一个管理员可以查看多个学员信息和管理多个教师信息，一个教师也可以代课多门课程。具体的E－R 图请看下图4-12所示。



图4-12 系统整体E-R图

### 4.2.2 逻辑结构设计

将上面的数据库概念模型转化为 SQL Server数据库所能支持的实际数据模型，就是数据库的逻辑结构。在实体以及实体之间关系的基础上，形成数据库中的表格属性之间的关系以及各个表格属性之间的关系。

资产（资产ID,资产编码,资产名称,资产一级类别ID,资产一级类别名称,资产二级类别ID,资产二级类别名称,规格型号,品牌,生产厂家,单位ID,单位名称,使用情况ID,事业部ID,事业部名称,部门ID,部门名称,保管者账户,保管者,增加方式ID,增加方式名称,原值,出厂日期,购置日期,入账日期,折旧方法ID,折旧方法名称,累计折旧,月折旧,残值率,剩余月份,净值,残值,有限年份,存放地点,备注,使用者账户,使用者,是否低值易耗品,资产状态,是否新增,申请状态,申请人账户,申请人,申请时间,添加时间）

增加方式（增加方式ID，排序ID，增加方式，添加时间）

资产异动（异动ID，异动编号，资产ID，异动后事业部ID，异动后事业部，异动后部门ID，异动后部门，异动后保管人账户，异动后保管人，异动后存放地点，异动日期，转移人账户，转移人，审核人账户，审核人，备注，异动状态，添加时间）

资产报废（报废ID，报废编号，资产ID，申请人账户，申请人，审核人账户，审核人，减少日期，减少方式ID，减少方式，报废原因，报废状态，添加时间）

部门（部门ID，排序ID，所属事业部ID，部门，添加时间）

折旧方法（方法ID，排序ID，方法，添加时间）

事业部（事业部ID，排序ID，事业部，添加时间）

资产一级类别（一级类别ID，排序ID，一级类别，添加时间）

打印列表（专业ID，班级ID，专业名称，班级名称）

资产维修（打印ID，打印资产ID，打印人账户，打印状态，添加时间）

资产二级类别（二级类别ID 主键，排序ID，所属一级类别ID，二级类别，单位ID，单位，使用年限，折旧率，添加时间）

单位（单位ID，排序ID，单位，添加时间）

用户（用户ID，账户，密码，事业部ID，部门ID，部门，用户，职称，电话，邮件，地址，级别，是否启用，是否网络维护人员，用户说明，谁添加的，登录时间，登录IP，修改时间，添加时间，能否管理资产，能否新增资产，能否审核新增，能否审核异动，能否审核维修，能否审核报废，能否批量异动，能否管理我的资产，能否管理事业部，能否管理部门，能否管理一级类别，能否管理二姐类别，能否管理用户，能否修改资料）。

（以上下划线位置即为主键）。

### 4.2.3 数据库表的设计

（1）资产信息表

表4-1 资产信息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| FixedAssetsID | bigint | 资产ID 主键 |
| AssetsCoding | varchar(25) | 资产编码 |
| AssetName | nvarchar(100) | 资产名称 |
| MajorID | int | 资产一级类别ID |
| MajorName | nvarchar(50) | 资产一级类别名称 |
| SubID | int | 资产二级类别ID |
| SubName | nvarchar(50) | 资产二级类别名称 |
| SpecificationsModel | nvarchar(50) | 规格型号 |
| Brand | nvarchar(50) | 品牌 |
| Manufacturer | nvarchar(50) | 生产厂家 |
| UnitsID | int | 单位ID |
| UnitsName | nvarchar(50) | 单位名称 |
| UseSituationID | int | 使用情况ID,1：使用中；2：报废；3：闲置；4：维修中。 |
| DivisionID | int | 事业部ID |
| DivisionName | nvarchar(50) | 事业部名称 |
| DepartmentID | int | 部门ID |
| DepartmentName | nvarchar(50) | 部门名称 |
| UserAccount | nvarchar(50) | 保管者账户 |
| Contactor | nvarchar(50) | 保管者 |
| AddWaysID | int | 增加方式ID |
| AddwaysName | nvarchar(50) | 增加方式名称 |
| OrginalValue | numeric(80,2) | 原值 |
| ExFactoryDate | datetime | 出厂日期 |
| PurchaseDate | datetime | 购置日期 |
| RecordedDate | datetime | 入账日期 |
| MethodID | int | 折旧方法ID |
| MethodName | nvarchar(50) | 折旧方法名称 |
| AccumulatedDepreciation | numeric(18,2) | 累计折旧 |
| MonthDepreciation | numeric(18,2) | 月折旧 |
| ResidualValueRate | numeric(18,2) | 残值率 |
| RemainderMonth | numeric(18,2) | 剩余月份 |
| NetValue | numeric(18,2) | 净值 |
| Residuals | numeric(18,2) | 残值 |
| LimitedYear | numeric(18,2) | 有限年份 |
| StorageSites | nvarchar(255) | 存放地点 |
| AssetsBackup | ntext | 备注 |
| UseUserAccount | nvarchar(50) | 使用者账户 |
| UseContactor | nvarchar(50) | 使用者 |
| LowConsumables | int | 是否低值易耗品 |
| AssetStatus | int | 资产状态 |
| isNewAdd | int | 是否新增 |
| ApplyStatus | int | 申请状态 |
| ApplyUserAccount | nvarchar(50) | 申请人账户 |
| ApplyContactor | nvarchar(50) | 申请人 |
| ApplyDate | datetime | 申请时间 |
| AddTime | datetime | 添加时间 |

（2）增加方式表

表4-2 增加方式表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| AddWaysID | int | 增加方式ID 主键 |
| AddWaysSortID | int | 排序ID |
| AddWaysName | nvarchar(50) | 增加方式 |
| AddTime | datetime | 添加时间 |

（3）资产异动表

表4-3 资产异动表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| AssetsChangesID | bigint | 异动ID 主键 |
| AssetsChangesNum | varchar(25) | 异动编号 |
| FixedAssetsID | bigint | 资产ID |
| CDivisionID | int | 异动后事业部ID |
| CDivisionName | nvarchar(50) | 异动后事业部 |
| CDepartmentID | int | 异动后部门ID |
| CDepartmentName | nvarchar(50) | 异动后部门 |
| CUserAccount | nvarchar(50) | 异动后保管人账户 |
| CContactor | nvarchar(50) | 异动后保管人 |
| CStorageSites | nvarchar(50) | 异动后存放地点 |
| CChangesDate | datetime | 异动日期 |
| TransferUserAccount | nvarchar(50) | 转移人账户 |
| TransferPeople | nvarchar(50) | 转移人 |
| ApprovedUserAccount | nvarchar(50) | 审核人账户 |
| ApprovedPerson | nvarchar(50) | 审核人 |
| CBackup | ntest | 备注 |
| ChangesStatus | int | 异动状态 |
| AddTime | datetime | 添加时间 |

（4）资产报废表

表4-4 资产报废表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| AssetsScrappedID | bigint | 报废ID 主键 |
| AssetsScrappedNum | varchar(25) | 报废编号 |
| FixedAssetsID | bigint | 资产ID |
| ApplyUserAccount | nvarchar(50) | 申请人账户 |
| Applicant | nvarchar(50) | 申请人 |
| ApprovedUserAccount | nvarchar(50) | 审核人账户 |
| ApprovedPerson | nvarchar(50) | 审核人 |
| ReduceDate | datetime | 减少日期 |
| ReduceWaysID | int | 减少方式ID，1：报废；2：盘亏；3：调出；4：非常损失； |
| ReduceWays | nvarchar(50) | 减少方式 |
| ScrappedReason | ntext | 报废原因 |
| ScrappedStatus | int | 报废状态 |
| AddTime | datetime | 添加时间 |

（5）部门表

表4-5 部门表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| DepartmentID | int | 部门ID 主键 |
| DepartmentSortID | int | 排序ID |
| DivisionID | int | 所属事业部ID |
| DepartmentName | nvarchar(50) | 部门 |
| AddTime | datetime | 添加时间 |

（6）折旧方法表

表4-6 折旧方法表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| MethodID | int | 方法ID 主键 |
| MethodSortID | int | 排序ID |
| MethodName | nvarchar(50) | 方法 |
| AddTime | datetime | 添加时间 |

（7）事业部表

表4-7 事业部表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| DivisionID | int | 事业部ID 主键 |
| DivisionSortID | int | 排序ID |
| DivisionName | nvarchar(50) | 事业部 |
| AddTime | datetime | 添加时间 |

（8）资产一级类别表

表4-8 资产一级类别表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| MajorID | int | 一级类别ID 主键 |
| MajorSortID | int | 排序ID |
| MajorName | nvarchar(50) | 一级类别 |
| AddTime | datetime | 添加时间 |

（9）打印列表表

表4-9 打印列表表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| PrintListID | bigint | 打印ID 主键 |
| PFixedAssetsID | bigint | 打印资产ID |
| PUserAccount | nvarchar(50) | 打印人账户 |
| PrintStatus | int | 打印状态 |
| AddTime | datetime | 添加时间 |

（10）资产维修表

表**4-10**资产维修表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| RepairListID | bigint | 维修ID 主键 |
| FixedAssetsID | bigint | 资产ID |
| AssetsCoding | varchar(25) | 资产编码 |
| AssetName | nvarchar(100) | 资产名称 |
| SpecificationsModel | nvarchar(50) | 规格型号 |
| RepairUserAccount | nvarchar(50) | 维修人账户 |
| RepairContactor | nvarchar(50) | 维修人 |
| RepairDivisionID | int | 维修事业部ID |
| RepairDepartmentID | int | 维修部门ID |
| RepairDepartmentName | nvarchar(255) | 维修部门 |
| RepairTel | nvarchar(50) | 维修电话 |
| RepairAddress | nvarchar(255) | 维修地址 |
| RepairTitle | nvarchar(255) | 维修标题 |
| RepairContent | ntext | 维修原因 |
| RepairIP | nvarchar(50) | 维修IP |
| RepairTime | datetime | 维修时间 |
| DealUserAccount | nvarchar(50) | 处理人账户 |
| DealContactor | nvarchar(50) | 处理人 |
| DealTime | datetime | 处理时间 |
| DealState | int | 处理状态 |
| DealContent | ntext | 处理内容 |

（11）资产二级类别表

表4-5 资产二级类别表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| SubID | int | 二级类别ID 主键 |
| SubSortID | int | 排序ID |
| MajorID | int | 所属一级类别ID |
| SubName | nvarchar(50) | 二级类别 |
| UnitsID | int | 单位ID |
| Units | nvarchar(50) | 单位 |
| UsefulLife | numeric(18,0) | 使用年限 |
| DepreciationRate | numeric(18,2) | 折旧率 |
| AddTime | datetime | 添加时间 |

（12）单位表

表4-5 单位表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| UnitsID | int | 单位ID 主键 |
| UnitsSortID | int | 排序ID |
| UnitsName | nvarchar(50) | 单位 |
| AddTime | datetime | 添加时间 |

（13）用户表

表4-5 用户表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| UserListID | bigint | 用户ID 主键 |
| UserAccount | nvarchar(50) | 账户 |
| UserPassword | nvarchar(50) | 密码 |
| DivisionID | int | 事业部ID |
| DepartmentID | int | 部门ID |
| DepartmentName | nvarchar(255) | 部门 |
| Contactor | nvarchar(50) | 用户 |
| JobPosition | nvarchar(50) | 职称 |
| Tel | nvarchar(50) | 电话 |
| Email | nvarchar(50) | 邮件 |
| Address | nvarchar(50) | 地址 |
| UserLevel | int | 级别 |
| IsAllowed | int | 是否启用 |
| IsMaintenance | int | 是否网络维护人员 |
| UserDescription | nvarchar(255) | 用户说明 |
| AddWho | nvarchar(50) | 谁添加的 |
| LoginTimes | int | 登录时间 |
| LoginIP | nvarchar(50) | 登录IP |
| UpdateTime | datetime | 修改时间 |
| AddTime | datetime | 添加时间 |
| HasDuty\_ManageFixedAssets | int | 能否管理资产 |
| HasDuty\_AddFixedAssets | int | 能否新增资产 |
| HasDuty\_CheckNewFixedAssets | int | 能否审核新增 |
| HasDuty\_CheckChangeFixedAssets | int | 能否审核异动 |
| HasDuty\_CheckRepairFixedAssets | int | 能否审核维修 |
| HasDuty\_CheckScrappedFixedAssets | int | 能否审核报废 |
| HasDuty\_VolumeChanges | int | 能否批量异动 |
| HasDuty\_MyFixedAssets | int | 能否管理我的资产 |
| HasDuty\_ManageDivision | int | 能否管理事业部 |
| HasDuty\_ManageDepartment | int | 能否管理部门 |
| HasDuty\_ManageMajorclass | int | 能否管理一级类别 |
| HasDuty\_ManageSubclass | int | 能否管理二姐类别 |
| HasDuty\_ManageUser | int | 能否管理用户 |
| HasDuty\_ModifyProfile | int | 能否修改资料 |

# 5 系统详细设计与实现

## 5.1 登录功能模块

### 5.1.1 登录功能模块概述

用户在登录公司资产管理系统前，要确认自己是否已经注册过并拥有自己的工号和密码，这些工号和密码可以由有管理权限的用户管理。用户输入工号和密码后，系统会在后台去数据库验证用户名和密码，登录成功后会有相应的权限菜单给用户操作，登录失败会在登录页面显示失败信息，其具体的流程图如图5-1所示。

### 5.1.2 登录功能模块流程设计



图5-1用户登录系统流程图

### 5.1.3 登录功能模块核心代码

public void LoadData(string userAccount)

{

Database db = new Database();

string sql = "";

sql = "Select \* From UserLists where IsAllowed=0 and UserAccount = "

+ SqlStringConstructor.GetQuotedString(userAccount);

DataRow dr = db.GetDataRow(sql); //利用Database类的GetDataRow方法查询用户数据

//根据查询得到的数据，对成员赋值

if (dr != null)

{

this.\_userListID = GetSafeData.ValidateDataRow\_N(dr, "UserListID");

this.\_userAccount = GetSafeData.ValidateDataRow\_S(dr, "UserAccount");

this.\_userPassword = GetSafeData.ValidateDataRow\_S(dr, "UserPassword");

this.\_divisionID = GetSafeData.ValidateDataRow\_N(dr, "DivisionID");

this.\_departmentID = GetSafeData.ValidateDataRow\_N(dr, "DepartmentID");

this.\_departmentName = GetSafeData.ValidateDataRow\_S(dr, "DepartmentName");

this.\_contactor = GetSafeData.ValidateDataRow\_S(dr, "Contactor");

this.\_jobPosition = GetSafeData.ValidateDataRow\_S(dr, "JobPosition");

this.\_tel = GetSafeData.ValidateDataRow\_S(dr, "Tel");

this.\_email = GetSafeData.ValidateDataRow\_S(dr, "Email");

this.\_address = GetSafeData.ValidateDataRow\_S(dr, "Address");

this.\_userLevel = GetSafeData.ValidateDataRow\_N(dr, "UserLevel");

this.\_isMaintenance = GetSafeData.ValidateDataRow\_N(dr, "IsMaintenance");

//获取权限集合

string colName = "";

for (int i = 0; i < dr.ItemArray.Length; i++)

{

colName = dr.Table.Columns[i].ColumnName;

if (colName.StartsWith("HasDuty\_") && GetSafeData.ValidateDataRow\_N(dr, colName) == 1)

{

this.\_duties.Add(dr.Table.Columns[i].ColumnName.Substring(8)); //去掉前缀“HasDuty\_”

}

}

this.\_exist = true;

}

else

{

this.\_exist = false;

}

}

## 5.2 注册功能模块

### 5.2.1 注册功能模块概述

用户如果没有注册是不能进入系统的，在注册页面填写个人信息进行注册，后台会验证个人信息和工号的重复性，如果注册失败页面上会有相应的提示信息。

### 5.2.2 注册功能模块流程设计



图5-2用户注册流程图

### 5.2.3 注册功能模块核心代码

private void BtnRegistered\_Click(object sender, RoutedEventArgs e)

{

UserList userlist = new UserList();

userlist.LoadData(mtbUserAccount.Text);

if (userlist.Exist)

{

txbErr.Text = "您登记的用户名已经存在!";

txbErr.Visibility = Visibility.Visible;

return;

}

else

{

Hashtable ht = new Hashtable();

ht.Add("UserAccount", SqlStringConstructor.GetQuotedString(mtbUserAccount.Text));

ht.Add("UserPassword", SqlStringConstructor.GetQuotedString(mtbUserPassword.Text));

ht.Add("DivisionID", SqlStringConstructor.GetQuotedString(cbxDivisionID.SelectedValue.ToString()));

ht.Add("DepartmentID", SqlStringConstructor.GetQuotedString(cbxDepartmentID.SelectedValue.ToString()));

ht.Add("Contactor", SqlStringConstructor.GetQuotedString(mtbContactor.Text));

ht.Add("JobPosition", SqlStringConstructor.GetQuotedString(mtbJobPosition.Text));

ht.Add("Tel", SqlStringConstructor.GetQuotedString(mtbTel.Text));

UserList adduserlist = new UserList();

adduserlist.Add(ht);

MessageBox.Show("用户登记成功！", "成功", MessageBoxButton.OK, MessageBoxImage.Information);

tabiLogin.IsSelected = true;

}

}

## 5.3 新增资产功能模块

### 5.23.1 新增资产功能模块概述

用户需要新增资产，可以进入新增资产页面，输入资产信息确认新增后，后台会检测新增资产是否规范，如果不能信息会有提示信息。规范的资产新增成功。流程图如图5-3所示。

### 5.3.2 新增资产功能模块流程设计



图5-3新增资产流程图

### 5.3.3 新增资产功能模块核心代码

private void BtnEnter\_Click(object sender, RoutedEventArgs e)

{

//2.获取用户在页面上的输入

string assetsCoding = txtAssetsCoding.Text; //资产编码

int showAssetsCoding = Convert.ToInt32(txtAssetsCoding.Text); //资产编码

int showAssetNum = Convert.ToInt32(txtAssetNum.Text); //资产数目

for (int i = 0; i < showAssetNum; i++)

{

showAssetsCoding = showAssetsCoding + 1;

FixedAsset fixedAsset = new FixedAsset();

fixedAsset.LoadData(assetsCoding);

if (fixedAsset.Exist)

{

MessageBox.Show("您输入的固定资产编码已经存在！", "失败", MessageBoxButton.OK, MessageBoxImage.Stop);

return;

}

else

{

Hashtable ht = new Hashtable();

ht.Add("AssetsCoding", showAssetsCoding);

}

}

MessageBox.Show("添加固定资产成功！", "成功", MessageBoxButton.OK, MessageBoxImage.Information);

InitialData();

}

## 5.4 查询资产功能模块

### 5.4.1 查询资产功能模块概述

如果要查询资产，就要在进入资产查询页面，输入关键字等查询信息，如果查询到信息就会给出查询结果。具体的数据流程图如下图5-4所示。

### 5.4.2 查询资产功能模块流程设计



图5-4查询资产流程图

### 5.4.3 查询资产功能模块核心代码

private void BtQuery\_Click(object sender, RoutedEventArgs e)

{

string strSql = "";

int divisionID, departmentID;

showAllUseSituationID = showAllUseSituationID.TrimEnd(',');

DataView dvlist = new DataView();

if (cbxDivisionID.SelectedValue == null & cbxDepartmentID.SelectedValue == null)

{

strSql = "select \* from Web\_V\_FixedAssets where UseSituationID in(" + showAllUseSituationID + ") and (AssetsCoding like \'%" + keywords + "%\' or AssetName like \'%" + keywords + "%\') order by FixedAssetsID desc";

}

else if (cbxDivisionID.SelectedValue != null & cbxDepartmentID.SelectedValue == null)

{

divisionID = Convert.ToInt32(cbxDivisionID.SelectedValue.ToString());

strSql = "select \* from Web\_V\_FixedAssets where UseSituationID in(" + showAllUseSituationID + ") and DivisionID=" + divisionID + " and (AssetsCoding like \'%" + keywords + "%\' or AssetName like \'%" + keywords + "%\') order by FixedAssetsID desc";

}

else if (cbxDivisionID.SelectedValue != null & cbxDepartmentID.SelectedValue != null)

{

divisionID = Convert.ToInt32(cbxDivisionID.SelectedValue.ToString());

departmentID = Convert.ToInt32(cbxDepartmentID.SelectedValue.ToString());

strSql = "select \* from Web\_V\_FixedAssets where UseSituationID in(" + showAllUseSituationID + ") and DivisionID=" + divisionID + " and DepartmentID=" + departmentID + " and (AssetsCoding like \'%" + keywords + "%\' or AssetName like \'%" + keywords + "%\') order by FixedAssetsID desc";

}

dvlist = FixedAsset.QueryFixedAssets1(strSql);

dtgShow.ItemsSource = dvlist;

//导出excel按钮启用

btExport.IsEnabled = true;

}

## 5.5 导出Excel功能模块

### 5.5.1 导出Excel功能模块概述

如果想要导出资产到Excel中，首先要查询资产，然后再对查询到的资产做导出操作。具体的流程图如下图5-5所示。

### 5.5.2 导出Excel功能模块流程设计



图5-5导出Excel流程图

### 5.5.3 导出Excel功能模块核心代码

//cmd = new OleDbCommand("create table [sheet4]([姓名] Text,[年龄] Text,[电话] int)", conn);

//cmd.ExecuteNonQuery();

string strSQL = "INSERT INTO [Sheet1$] ([资产编码],[资产名称],[资产大类ID],[资产大类],[资产小类ID],[资产小类],[购置日期],[建立日期],[型号规格],[生产厂家],[品牌],[保管员工号],[保管员姓名],[存放地点],[事业部ID],[事业部名称],[部门ID],[部门名称],[使用情况ID],[使用情况],[申请状态ID],[增加方式ID],[增加方式],[计量单位ID],[计量单位],[原值],[有限年份],[残值率],[剩余月份],[残值],[月折旧],[累计折旧],[净值],[备注],[低值易耗品]) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

cmd = new OleDbCommand(strSQL, conn);

for (int i = 0; i < 35; i++)

{

cmd.Parameters.Add(i.ToString(), OleDbType.VarChar);

}

DataView dv = dvlist;

foreach (DataRowView row in dv)

{

cmd.Parameters[0].Value = row["AssetsCoding"].ToString();

cmd.ExecuteNonQuery();

}

bRet = true;

## 5.6 资产管理功能模块

### 5.6.1 资产管理功能模块概述

当用户要管理资产，先要进入资产管理页面，选择操作后，系统检测操作是否规范。规范就操作成功，否则给出提示。流程图如下图5-6所示。

### 5.6.2 资产管理功能模块流程设计



图5-6资产管理流程图

### 5.6.3 资产管理功能模块核心代码

public bool Update(String TableName, Hashtable Cols, String Where)

{

//更新表

int Count = 0;

if (Cols.Count <= 0)

{

return true;

}

String Fields = " ";

foreach (DictionaryEntry item in Cols)

{

if (Count != 0)

{

Fields += ",";

}

Fields += "[" + item.Key.ToString() + "]";

Fields += "=";

Fields += item.Value.ToString();

Count++;

}

Fields += " ";

String SqlString = "Update " + TableName + " Set " + Fields + Where;

String[] Sqls = { SqlString };

return ExecuteSQL(Sqls);

}

## 5.7 审核资产功能模块

### 5.7.1 审核资产功能模块概述

用户对资产的操作比如新增、异动、维修、报废，会进入新增待审核状态，进入审核页面确认审核，系统判断审核信息无误，就能审核成功，否则给出提示信息。详细的数据流程图如下图5-7所示。

### 5.7.2 审核资产功能模块流程设计



图5-7审核新增流程图

### 5.7.3 审核资产功能模块核心代码

public static DataView QueryFixedAssets(int applyStatus)

{

string strSql = "";

strSql = "select FixedAssetsID,AssetsCoding,AssetName,SpecificationsModel,Manufacturer,UserAccount,Contactor,PurchaseDate,UseSituationID,AssetStatus,IsNewAdd,ApplyStatus From FixedAssets where ApplyStatus="+ applyStatus +" order by FixedAssetsID desc";

Database db = new Database();

return db.GetDataView(strSql);

}

## 5.8 部门管理功能模块

### 5.8.1 部门管理功能模块概述

当用户要管理部门，先要进入部门管理页面，选择并进行操作后，系统检测操作是否规范。规范就操作成功，否则给出提示。流程图如下图5-8所示。

### 5.8.2 部门管理功能模块流程设计



图5-8部门管理流程图

### 5.8.3 部门管理功能模块核心代码

public static DataView QueryDivision()

{

string strSql = "";

strSql = "select DivisionID,DivisionSortID,DivisionName,AddTime From Divisions order by DivisionSortID,DivisionID desc";

Database db = new Database();

return db.GetDataView(strSql);

}

## 5.9 类别管理功能模块

### 5.9.1 类别管理功能模块概述

当用户要管理资产类别，先要进入资产类别管理页面，选择并进行操作后，系统检测操作是否规范。规范就操作成功，否则给出提示。流程图如下图5-9所示。

### 5.9.2 类别管理功能模块流程设计



图5-9类别管理流程图

### 5.9.3 类别管理功能模块核心代码

public static DataView QueryMajorClass()

{

string strSql = "";

strSql = "select MajorID,MajorSortID,MajorName,AddTime From MajorClass order by MajorSortID,MajorID desc";

Database db = new Database();

return db.GetDataView(strSql);

}

## 5.10 用户管理功能模块

### 5.10.1 用户管理功能模块概述

当用户要管理资产，先要进入资产管理页面，选择操作后，系统检测操作是否规范。规范就操作成功，否则给出提示。流程图如下图5-10所示。

### 5.10.2 用户管理功能模块流程设计



图5-10用户管理流程图

### 5.10.3 用户管理功能模块核心代码

public static DataView QueryAllUserLists()

{

string strSql = "";

strSql = "Select a.UserListID,a.UserAccount,a.UserPassword,a.Contactor,a.DivisionID,a.DepartmentID,a.JobPosition,a.Tel,a.Email,a.Address,a.UserLevel,a.IsAllowed,a.IsMaintenance,b.DepartmentName,c.DivisionName From UserLists a left outer join Departments b on a.DepartmentID=b.DepartmentID left outer join Divisions c on b.DivisionID=c.DivisionID order by a.UserListID desc";

Database db = new Database();

return db.GetDataView(strSql);

}

## 5.11 打印功能模块

### 5.11.1 打印功能模块概述

当用户要管理资产，先要进入资产管理页面，选择操作后，系统检测操作是否规范。规范就操作成功，否则给出提示。流程图如下图5-11所示。

### 5.11.2 打印功能模块流程设计



图5-11打印流程图

### 5.11.3 打印功能模块核心代码

private void BtPrintMark\_Click(object sender, RoutedEventArgs e)

{

string userAccount = string.Empty;

if (ini.ExistINIFile())

{

userAccount = ini.IniReadValue("登录详细", "UserAccount");

}

DataView dvlist = PrintList.Query\_V\_PrintList(userAccount);

QRcode qr = new QRcode();

foreach (DataRowView printList in dvlist)

{

printList.Row["StorageSites"]= System.AppDomain.CurrentDomain.BaseDirectory+"~"+qr.Create(printList.Row["AssetsCoding"].ToString(), 2, "");

}

myItemsControl.ItemsSource = dvlist;

tabpPrintCard.IsSelected = true;

}

6 系统功能测试

## 6.1 登录功能测试

登录页面如图6-1所示，我们输入用户名和密码，然后点击登录按钮。



图6-1登录测试图

## 6.2 注册功能测试

注册页面如图6-2所示，我们输入用户名、密码、职称、真实姓名、电话和部门信息，然后点击注册按钮。



图6-2注册测试图

## 6.3 新增资产功能测试

新增资产页面如图6-3所示，我们输入资产名称，类别，编码，部门等必填信息，然后点击确认新增按钮。



图6-3新增资产测试图

## 6.4 查询资产功能测试

查询资产页面如图6-4所示，我们输入关键字、使用情况、和部门信息，然后点击开始查询按钮。

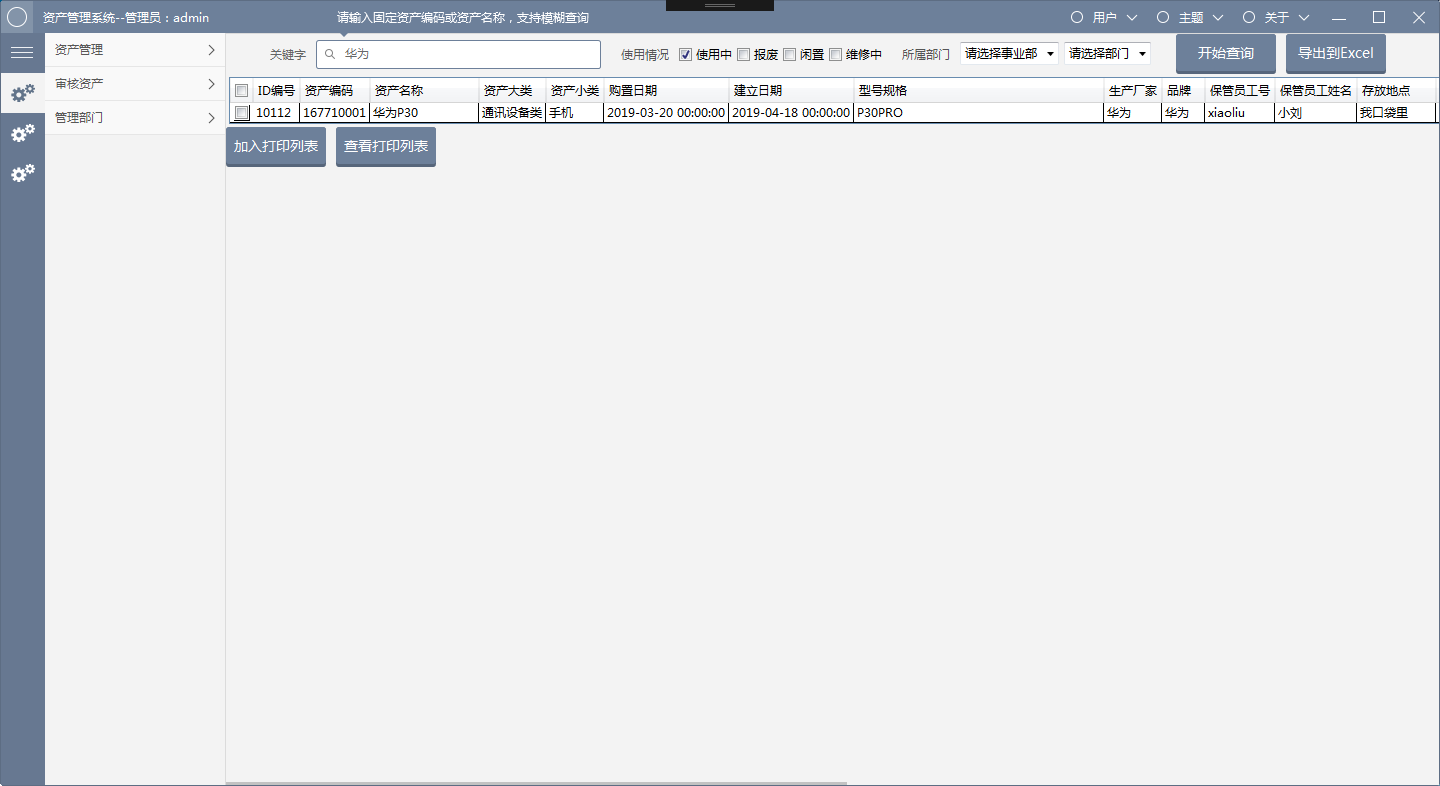


图6-4查询资产测试图

## 6.5 导出到Excel功能测试

在查询资产后我们可以点击导出到Excel按钮，把当前查询的资产导出到Excel再用Excel打开如图6-5所示。

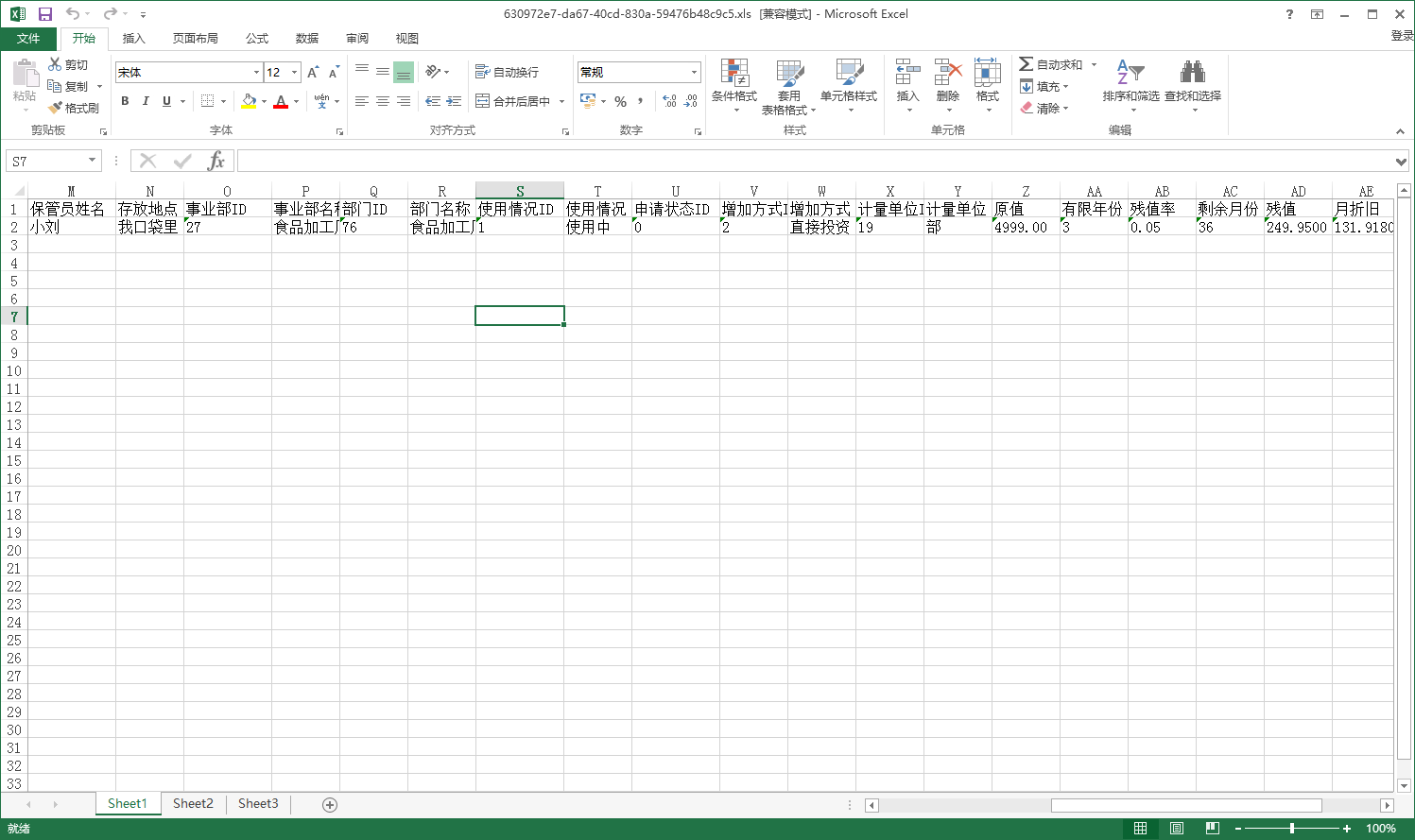


图6-5导出到Excel测试图

## 6.5 资产管理功能测试

查询资产页面如图6-4所示，查看可以进行的操作，可以对已有资产进行修改、异动、维修、报废、打印等操作。

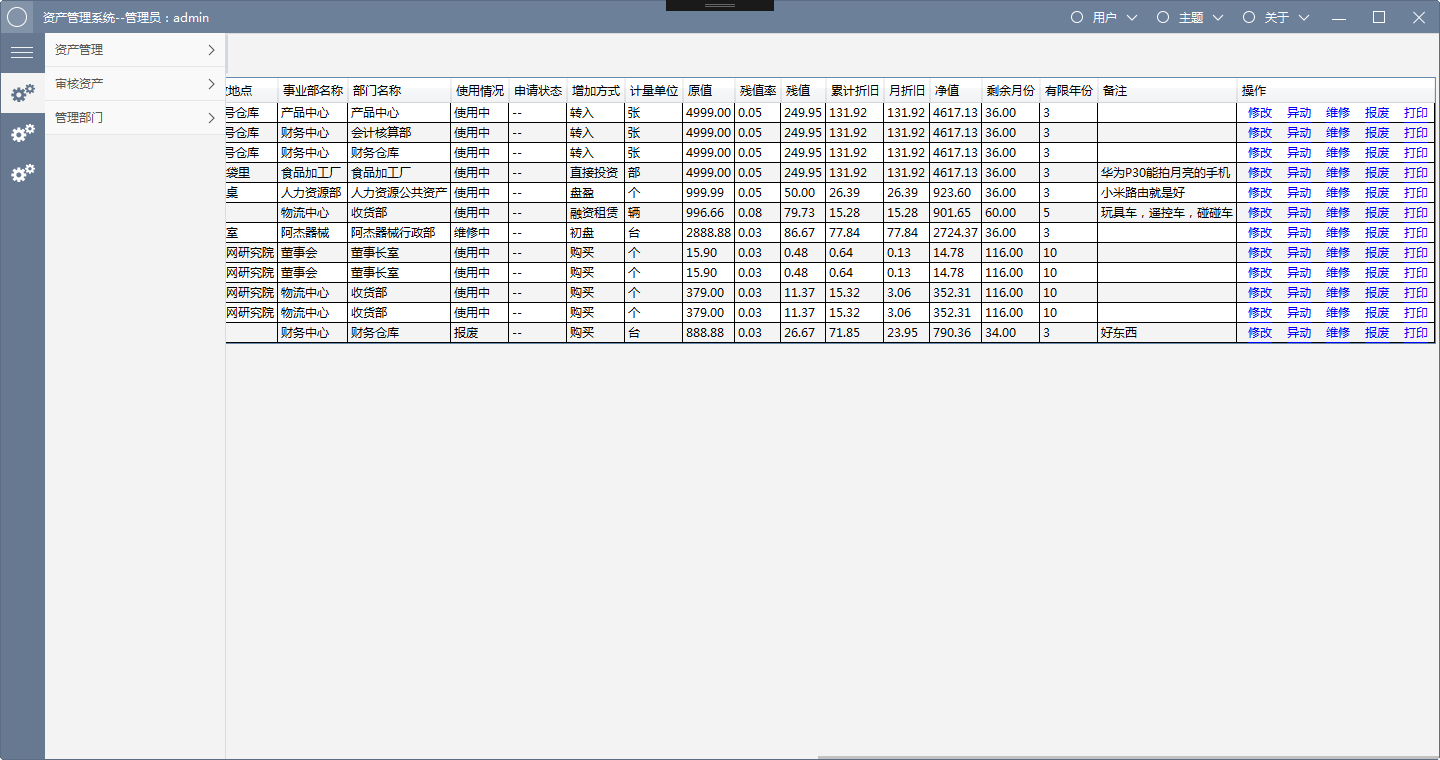


图6-6资产管理测试图

（1）修改资产

在资产管理页面点击修改，进入修改资产页面如图6-7所示，输入需要修改的信息，然后点击确认操作。



图6-7资产管理测试图

（2）异动资产

在资产管理页面点击异动，进入修改资产页面如图6-8所示，输入需要异动的信息，然后点击确认操作。



图6-8异动资产录入界面

# 参考文献

[1]黄卿,梁柳宁,黄真珍,程光煜,许江超.基于C#.NET编程的固定资产智能管理系统设计与开发[J].视听,2018(04):8-13.

[2]蔡凯.二维码固定资产管理系统[J].条码与信息系统,2018(05):18-19.

[3]李坤.中小企业运用科技信息手段实现固定资产动态化管理[J].财经界(学术版),2014(03):115.

[4]赵文平.关于多校区高校的资产综合化管理分析[J].现代营销(经营版),2018(10):219.

[5]李蕾. H高校固定资产内部控制优化研究[D].西安石油大学,2018.

[[6]贾鸣华. 基于JavaEE的电子类资产管理系统的设计与实现[D].南京大学,2018.](http://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CMFD&filename=1018150493.nh&v=MjQ4NDc5SHRYRnJKRWJQSVFLREg4NHZSNFQ2ajU0TzN6cXFCdEdGckNVUkxPZlplUm5GeTNoV3JyTFZGMjZGcks=&uid=WEEvREcwSlJHSldRa1FhdkJkVG1COG9TMU5nY1pRMUhKelFwTS9ReHd1ND0=$9A4hF_YAuvQ5obgVAqNKPCYcEjKensW4IQMovwHtwkF4VYPoHbKxJw!!" \t "_blank)

[7]陈晓瑜.高校资产管理信息化平台建设研究——以GDGM职业技术学院为例[J].现代营销(创富信息版),2019(01):154-155.

[8]李平. 经营性资产管理系统设计与实现[D].西安电子科技大学,2016.

[9]罗琼.中小型制造企业固定资产核算问题研究[J].科技经济导刊,2018(02):174.

[10]李坤.中小企业运用科技信息手段实现固定资产动态化管理[J].财经界(学术版),2014(03):115.

[11]赵艺鹏.中小企业固定资产动态化管理研究[J].现代经济信息,2017(10):252.

# 致谢

首先我要感谢学校的老师和同学们，一起学习了这么久，马上就要毕业了，感谢老师对我的悉心指导，感谢同学们对我的陪伴。感谢你们在生活上给予我关心和帮助，让我读过充实的大学生活。我祝愿同学们毕业后能找到理想的工作，也希望老师身体健康，万事如意。

然后我要感谢我的论文指导老师朱捷老师，在百忙之中挤出时间来审查，修改我的论文，提出了很多的宝贵意见和建议，您严谨细致、一丝不苟的作风是我学习的榜样，在老师的精心指导和大力支持下我才完成这篇论文