

摘 要

固定资产管理系统是一个企事业单位不可缺少的部分,它的内容对于企事业单位的决策者和管理者来说都至关重要,所以固定资产管理系统应该能够为用户提供充足的信息和快捷的查询手段。但一直以来人们使用传统人工的方式管理固定资产的信息,这种管理方式存在着许多缺点,如:效率低、保密性差,另外时间一长,将产生大量的文件和数据,这对于查找、更新和维护都带来了不少的困难。随着计算机技术的不断发展,计算机应用于各大领域,并给人们的生活带来了极大的便利,在固定管理系统亦是如此。以往固定资产员由于缺乏适当的软件而给其工作带来了很大不便。本论文所介绍的便是一个固定资产管理系统,以方便在固定资产安排和固定资产管理上的工作任务。本系统的目的是为用户提供便捷、高效的管理服务,同时,通过使用该系统对固定资产进行管理可以对固定资产的新购,变动,领用,退还,转交等信息一目了然,也方便用户对其固定资产进行维护更新。

本论文主要介绍的是固定资产管理系统的整个设计实现过程。前言包括选题的背景,目的和意义;第1章系统开发工具简介包括 Microsoft Visual Studio 2005 和 SQL Server 数据库的介绍;第2章系统分析,可行性分析,需求分析;第3章系统设计与实现部分包括系统的功能结构设计,系统的数据库结构设计,系统功能的实现;第4章是系统测试部分包括测试目的与环境,测试内容;论文最后为结论、致谢、参考文献。

关键词: 固定资产管理, SQL Server 数据库, B/S 结构

Abstract

Fixed Assets Management System is an enterprise units indispensable part. It is an enterprise units for the policy makers and managers are of great importance. Therefore, fixed asset management system should be able to provide users with sufficient information and efficient means of inquiry. But there has been the use of traditional artificial approach in the management of fixed assets information. This mode of administration, there are many drawbacks, such as : low efficiency, poor security, the other over the long term. will have a lot of documents and data, which identify, update and maintenance caused a lot of difficulties. As computer technology continues to develop, computers for all major areas, and to people's lives brought great convenience, In fixed management system is reflected. Members of the previous fixed assets due to the lack of appropriate software, which bring to their work a lot of inconvenience.

This paper is presented in a fixed asset management system, to facilitate the arrangements in fixed assets and fixed assets management of the task. This paper introduces the fixed-asset management system throughout the design process. Introduction of topics including the background, purpose and significance of the domestic; Chapter 1 Introduction including Microsoft Visual Studio 2005 platform and SQL Server database introduced; Chapter 2 analysis, possibility analysis demand analysis, ; Chapter 3 of system design and process to achieve, including functional construction design, database design, systematic process to achieve; Chapter 4 is include system test, the environment of testing, the content of system testing; Finally, the paper of the conclusions, thank references.

Keywords : fixed assets management, SQL Server database, B/S Construction

目 录

| | |
|-------------------------|----|
| 前 言 | 3 |
| 第一章 系统开发工具简介..... | 5 |
| 1.1 数据库平台的选择..... | 5 |
| 1.2 开发工具的选择..... | 7 |
| 1.2.1 .NET 框架简介 | 7 |
| 技术简介..... | 9 |
| 1.2.3 ADO.NET 技术简介..... | 10 |
| 第二章 系统分析..... | 12 |
| 2.1 系统目标..... | 12 |
| 2.2 可行性分析..... | 13 |
| 2.3 需求分析..... | 14 |
| 2.3.1 管理者的信息需求..... | 15 |
| 2.3.2 企业财务人员的工作需求..... | 15 |
| 2.3.3 员工财务管理需求..... | 15 |
| 2.3.4 公司业务需求..... | 16 |
| 2.3.5 运行需求..... | 16 |
| 2.3.6 安全性需求..... | 16 |
| 第三章 系统设计..... | 18 |
| 3.1 系统的功能结构设计..... | 18 |
| 3.2 系统的数据库结构设计..... | 20 |
| 3.2.1 系统数据流程分析..... | 20 |
| 3.2.2 系统数据库设计..... | 21 |
| 第四章 系统功能实现..... | 27 |
| 4.1 系统页面设计..... | 27 |
| 4.2 资产类别管理页面设计..... | 28 |
| 4.3 库存管理页面设计..... | 29 |
| 4.4 资产登记页面设计..... | 30 |
| 4.5 电脑管理页面设计..... | 31 |
| 4.6 办公设备管理页面..... | 32 |
| 4.7 车辆管理页面设计..... | 33 |
| 4.8 车辆维修计划页面设计..... | 34 |
| 4.9 配件管理页面设计..... | 35 |
| 4.10 小额用品管理页面设计..... | 36 |
| 4.11 软件管理页面设计..... | 36 |
| 4.12 移库处理页面设计..... | 37 |
| 4.13 原材料交付页面设计..... | 38 |
| 4.14 资产照会页面设计..... | 39 |
| 4.15 月资产变动页面设计..... | 39 |

| | |
|----------------------|----|
| 4.16 资产报废页面设计..... | 40 |
| 4.17 资产盘点页面设计..... | 41 |
| 4.18 维修申请登录页面设计..... | 42 |
| 第五章 系统测试..... | 43 |
| 测试目的与运行环境..... | 43 |
| 5.2 测试内容..... | 43 |
| 5.2.1 登录页面测试..... | 43 |
| 5.2.2 部分关键功能测试..... | 44 |
| 第六章 总 结..... | 51 |
| 致 谢..... | 52 |
| 参考文献..... | 53 |

前 言

固定资产管理系统是企业管理中的一个重要组成部分，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。许多企业在固定资产管理中也许都遇到过以下所列的问题：企业资产管理中帐、卡、物不相符合；不清楚每个资产所在的位置，也无法得知某个位置上究竟有多少资产；资产管理缺乏基础数据以及相对应的管理手段；当前状态无法跟踪，如调拨、借用、维修等，没有 IT 系统支持相关的工作流程；资产的报废无法及时处理，财务上无法及时销账，无法形成报废清单，实物拆下来后，无法与资产上的实物卡片进行核对；折旧计算繁复，准确性差，导致固定资产流失；资产缺乏中间跟踪管理没有资产的历史记录，如安装、移动、调拨、报废、维修等，没有和资产一一对应的设备编码；资产的保修无法进行管理；缺乏对非在线资产的有效管理。

一个企业的良性发展，避免不了的要涉及到企业资产的有效管理。对于那些技术装备密集型企业，固定资产的管理显得尤为重要。传统的固定资产管理模式无论从质量上还是效率上，都难以适应经营管理新形式的需要。因此，寻找一种简便、高效的管理手段成为必然。在充分研究分析企业固定资产管理的业务需求的基础上，固定资产管理系统，改变了固定资产清查数据的采集方式，解决固定资产实物清查的瓶颈问题，大大提高清查效率，同时也增加了固定资产的形态方面的管理，有效解决企业资产的管理难题，使企业更轻松、更有效地管理固定资产。如能在公司内部建立固定资产管理，将使各级相关管理人员及有关领导快速查询、统计固定资产情况，实现合理配置资源、决策，提供依据，提高工作效率。

固定资产管理系统可以有效地对固定资产进行科学管理，防止虚增资产和国家或者企业资产流失。

随着改革开放的深入特别是中国已经加入 WTO，中国经济将融入世界经济大潮，而这种大潮必将把中国企业抛入更激烈的竞争。

企业竞争的优势主要体现在生产效率上，为提高生产效率，企业纷纷进行信息化改造，实现管理的专业化和规范化。企业固定资产管理信息系统正以一种特别的方式在我国正呈现新的发展态势。

固定资产管理系统能为企业提供一个专业管理企业资产的模式。系统的目标是，以规范的内部业务流程为基础，建立一种用户同信息系统之间的关系，建立和维护一个高质量的物流管理信息系统。以便帮助用户方便人员资产管理能力更加强大，形式更加灵活的多层资产管理应用系统。

固定资产是企业重要的经济资源。很多企业固定资产的手段一直还处于手工，或者借助专业软件系统手动修改登记的管理方式。由于管理方式落后，导致了管理不善、使用效率低下，甚至在一些企业出现了固定资产管理混乱，给企业带来了巨大的浪费和损失。使用固定资产管理系统则改善了传统管理中对固定资产仅从财务角度的管理，记录了固定资产使用的流程及相关责任人；任何时间点管理者可以随时查询数据；可按多个查询条件，及时多角度了解固定资产的情况。

通过对固定资产的采购，借用赔偿等活动的录入，记录固定资产原始信息，以资产信息为中心可进行申领，发放，转交，调拨，退还，收回，变更等活动，对资产的跟踪式管理，避免了资产的丢失，损坏。

固定资产管理模块用于提供与固定资产、租赁资产和其他资产（如现场维护设备）等有关的财务信息。描述数据、位置数据、原值、折旧和维修信息或校准要求等应得到完善地维护。除此之外，用户可能希望保存保险或财产税等数据于相应的文件中，以便提供相应的报告及满足查询要求。

每种资产在系统中应有其对应的适用折旧方法及建立之时间表，以便系统在每个会计期间内能自动计算折旧额。

固定资产管理模块也应能提供一套完整的有关资产的取得和处置的交易审计轨迹报告。

第一章 系统开发工具简介

固定资产管理系统将 Internet 网络技术与现代管理观念相融合, 针对信息技术的特点对固定资产管理进行规划和重构, 对公司管理进行优化。

设计时使用了 WEB 的服务方式, 系统采用 B/S 结构, 数据库采用 SQL Server 前台选用了 ASP.NET。

1.1 数据库平台的选择

SQL Server 2005 是一个全面的、集成的、端到端的数据解决方案, 它为企业中的用户提供了一个安全、可靠和高效的平台用于企业数据管理和商业智能应用。SQL Server 2005 为 IT 专家和信息工作者带来了强大的、熟悉的工具, 同时减少了在从移动设备到企业数据系统的多平台上创建、部署、管理及使用企业数据和分析应用程序的复杂度。通过全面的功能集、和现有系统的集成性、以及对日常任务的自动化管理能力, SQL Server 2005 为不同规模的企业提供了一个完整的数据解决方案。图 1.1 显示了 SQL Server 2005 数据平台的组成架构。



图 1.1 SQL Server 2005 数据平台

SQL Server 2005 是一个全面的数据库平台与 Microsoft Visual Studio、Microsoft Office System 以及新的开发工具包（包括 Business Intelligence Development Studio）的紧密集成使 SQL Server 2005 与众不同。无论您是开发人员、数据库管理员、信息工作者还是决策者，SQL Server 2005 都可以为您提供创新的解决方案，帮助您从数据中更多地获益。

SQL SERVER 2005 主要提供了如下的特点：

- 全面重新设计的 DTS 体系结构和工具
- 引入了由管理工具和管理应用编程接口（API）组成的集成化套件用以降低操作的复杂度
- 主要改进包括表分区、增强复制功能和 64 位支持特性
- 允许使用任何 Microsoft .NET 语言开发数据库对象的能力
- 增添新的 XML 数据类型
- 新增查询类型和在事务中进行错误处理的能力

SQL Server 2005 包含许多可以显著提高开发人员生产效率的新技术包括以下方面：

- .NET 框架主机

使用 SQL Server 2005，开发人员通过使用相似的语言，例如微软的 Visual C# .NET 和微软的 Visual Basic，将能够创立数据库对象。开发人员还将能够建立两个新的对象——用户定义的和集合。

- XML 技术

在使用本地网络和互联网的情况下，在不同应用软件之间散步数据的时候，可扩展标记语言（XML）是一个重要的标准。SQL Server 2005 将会自身支持存储和查询可扩展标记语言文件。

- ADO.NET 2.0 版本

SQL 类的新的支持，到多活动结果集（MARS），SQL Server 2005 中的 ADO.NET 将推动数据集的存取和操纵，实现更大的可升级性和灵活性。

- 增强的安全性

SQL Server 2005 中的新安全模式将用户和对象分开，提供 fine-grain access 存取、并允许对数据存取进行更大的控制。另外，所有系统表格将作

为视图得到实施，对数据库系统对象进行了更大程度的控制。

- Transact-SQL 的增强性能

SQL Server 2005 为开发可升级的数据库应用软件，提供了新的语言功能。这些增强的性能包括处理错误、递归查询功能、关系运算符 PIVOT, APPLY, ROW_NUMBER 和其他数据列排行功能，等等。

- SQL 服务中介

SQL 服务中介将为大型、营业范围内的应用软件，提供一个分布式的、异步应用框架。

- 通告服务

通告服务使得业务可以建立丰富的通知应用软件，向任何设备，提供个人化的和及时的信息，例如股市警报、新闻订阅、包裹递送警报、航空公司票价等。在 SQL Server 2005 中，通告服务和其他技术更加紧密地融合在了一起，这些技术包括分析服务、SQL Server Management Studio 。

- Web 服务

使用 SQL Server 2005，开发人员将能够在数据库层开发 Web 服务，将 SQL Server 当作一个超文本传输协议（HTTP）侦听器，并且为网络服务中心应用软件提供一个新型的数据存取功能。

- 报表服务

利用 SQL Server 2005，报表服务可以提供报表控制，可以通过 Visual Studio 2005 发行。

- 全文搜索功能的增强

SQL SERVER 2005 将支持丰富的全文应用软件。服务器的编目功能将得到增强，对编目的对象提供更大的灵活性。查询性能和可升级性将大幅得到改进，同时新的管理工具将为有关全文功能的运行，提供更深入的了解。

1.2 开发工具的选择

1 .NET 框架简介

Microsoft.NET 是微软的新一代基于互联网的软件开发平台，它是一种全新

的开发平台。它能够使开发人员更容易建立各种网络环境下的应用系统。包括了
可以适合多种语言编程类库。它正在对现有的软件开发产生巨大影响。

框架(.NET Framework), 具有两个主要组件:公共语言运行(时) (CLR) 和
基础类库。公共语言运行库是.NET Framework的基础。它可以被看成是一个在执
行时管理代码的代理, 它提供核心服务(如内存管理、线程管理和远程处理), 而
而且还强制实施严格的类型安全以及可确保安全性和可靠性的其他形式的代码准
确性。.NET Framework的另一个主要组件是基础类库, 它是一个综合性的面向对
象的可重用类型集合, 它可以用来开发多种应用程序, 这些应用程序包括传统的
命令行或图形用户界面(GUI)应用程序, 也包括基于所提供的最新的应用程序(如
Web窗体和XML Web Services)。

.NET Framework 是生成、运行下一代应用程序和 XML Web Services 的
内部 Windows 组件。它简化了高度分布式 Internet 环境中的应用程序开发,.NET
Framework 旨在实现如下目标。

- 提供一个面向对象的、一致的编程环境, 无论对象代码是在本地存储执行,
还是在本地执行, 在 Internet 上分布, 或者是在远程执行的。

- 提供一个将软件部署和版本控制冲突最小化的代码执行环境

- 提供一个可提高代码(包括由未知的或不完全受信任的第三方创建的代
码)执行安全性的执行环境

- 提供一个可消除脚本环境或解决环境性能问题的代码执行环境, 是开发人
员的经验在面对不同类型的应用程序(如基于 Windows 的应用程序和基于 Web
的应用程序)时保持一致。

- 按照工业标准生成所有通信, 以确保基于.NET Framework 的代码可与任
何其他代码集成

也提供了多种编程语言的开发工具, 不同的编程语言除了语法上的区别, 功
能上的区别很小。公共语言运行库和XML Web Services(Web服务是.NET
Framework提供的两个核心的集成技术。对于最终用户来说, .NET技术将使计算
机的功能得到大幅度的提升, 而用户对Web的访问变得更加简单:对于程序开发人
员来说, 可以访问VS.NET (VisualSt的简写)所提供的内容丰富的类库, 直接使
用各种各样来自Internet的模块来构建自己的应用程序, 能够快速高效地开发应

用程序，同时，借助于.NET技术所提供的基于XML的Web服务技术，可以真正地将应用程序逻辑分发到网络上，不必顾及服务器与客户端是否属于同一种结构，执行机制是否相同，接口是否对应。能够提供加速开发过程的高效工具，提供对各种各样的网络应用程序的快速设计能力，利用XML和Web服务来简化分布式计算，快速构建中间层商务组件，以及构建可靠的可伸缩的解决方案。

下图是.Net 平台结构：

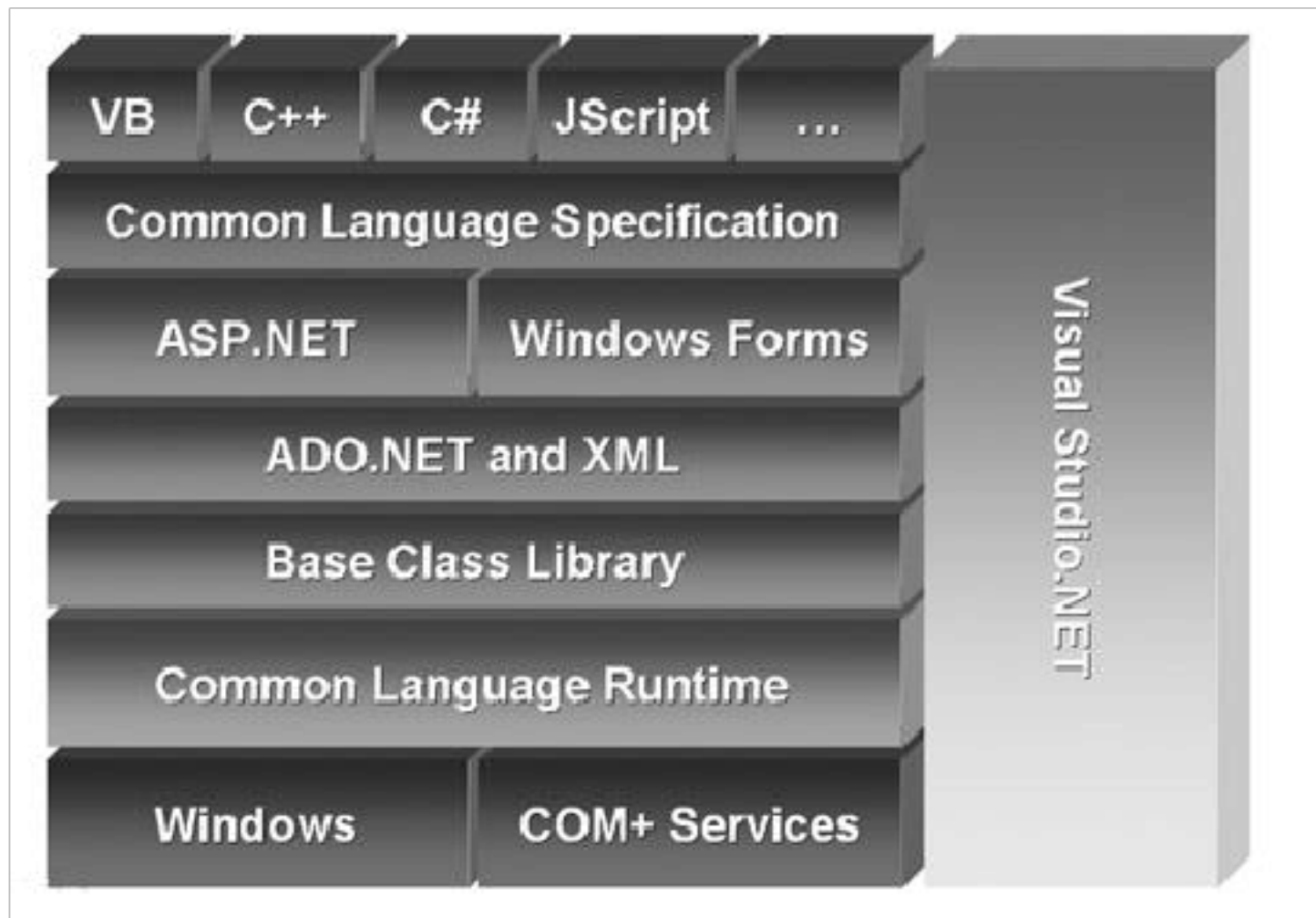


图 1.2 .Net 平台结构

1.2.2 A

ASP.NET又称为ASP+，它不是ASP的简单升级，而是微软开发的一种基于.NET平台的新技术，是建立、管理、部署Web应用程序最佳的平台。ASP.NET构建在CLR之上，可以使用CLR来编辑代码，管理执行，创造运行更快、表现更好的Web应用程序。ASP.NET可以无缝地与其他编程工具一起工作。这不仅使得Web开发更加方便，而且还能提供这些工具必须提供的所有优点，包括开发人员可以用来将服务器控件拖放到Web页的GUI和完全集成的调试支持。当创建应用程序时，开发人员可以使用Web窗体或XML Web services，或其他合适的任何方式进行组合。每个功能都能得到同样结构的支持。

此系统采用了ASP.NET技术，提供了一个统一的WEB开发模型。它提供了一

种新的编程模型和结构，可生成伸缩性和稳定性更好的应用程序，并提供更好的安全保护。可以通过在现有 ASP 应用程序中逐渐添加 ASP.NET 功能，随时增强 ASP 应用程序的功能。ASP.NET 是一个已编译的、基于 .NET 的环境，可以用任何与 .NET 兼容的语言（包括 Visual Basic .NET、C# 和 JScript .NET.）创作应用程序。另外，任何 ASP.NET 应用程序都可以使用整个 .NET Framework。开发人员可以方便地获得这些技术的优点，其中包括托管的公共语言运行库环境、类型安全、继承等。

微软为 ASP.NET 设计了这样一些策略：易于写出结构清晰的代码、代码易于重用和共享、可用编译类语言编写等等，目的是让程序员更容易开发出 WEB 应用，满足计算向 WEB 转移的战略需要。

- :ASP.NET 提供了稳定的性能、优秀的升级性、更快速的开发、更简便的管理、全新的语言以及网络服务。贯穿整个 ASP.NET 的主题就是系统帮客户做了大部分不重要的琐碎的工作。
- 全新的构造:新的 ASP.NET 引入受管代码 (Managed Code) 这样一个全新概念，横贯整个视窗开发平台。受管代码在 NGWS Runtime 下运行，而 NGWS Runtime 是一个时间运行环境，它管理代码的执行，使程序设计更简便。
- 高效率:对于一个程序，速度是一件非常令人渴望的东西。一旦代码开始工作，接下来你就得尽可能地让它运作得快些、再快些。在 ASP 中你只有尽可能精简你的代码，以至于不得不将它们移植到一个仅有很少一点性能的部件中。而现在，ASP.NET 会妥善地解决这一问题。
- 易控制:在 ASP.NET 里，你将会拥有一个“Data-Bounds”（数据约束），这意味着它会与数据源连接，并会自动装入数据，使控制工作简单易行。
- 语言支持:ASP.NET 支持多种语言，支持编译类语言，支持比如 VB、VC++、C# 等，它比这些编译类语言运行速度快，更适合编写大型应用。

1.2.3 ADO.NET 技术简介

ADO.NET 对 Microsoft SQL Server 和 XML 等数据源以及通过 OLE DB 和 XML 公开的数据源提供一致的访问。数据共享使用者应用程序可以使用 ADO.NET 来连接到这些数据源，并检索、处理和更新所包含的数据。

ADO.NET 通过数据处理将数据访问分解为多个可以单独使用或一前一后使用的非连续组件。ADO.NET 包含用于连接到数据库、执行命令和检索结果的 .NET Framework 数据提供程序。您可以直接处理检索到的结果，或将其放入 ADO.NET DataSet 对象，以便与来自多个源的数据或在层之间进行远程处理的数据组合在一起，以特殊方式向用户公开。ADO.NET DataSet 对象也可以独立于 .NET Framework 数据提供程序使用，以管理应用程序本地的数据或源自 XML 的数据。

ADO.NET 利用 XML 的力量来提供对数据的断开式访问。ADO.NET 的设计与 .NET Framework 中 XML 类的设计是并进的，它们都是同一个结构的组件。ADO.NET 和 .NET Framework 中的 XML 类集中于 DataSet 对象。无论 XML 源是文件还是 XML 流，都可以用来填充 DataSet。无论 DataSet 中数据的数据源是什么，DataSet 都可以作为符合万维网联合会 (W3C) 标准的 XML 进行编写，并且将其架构包含为 XML 架构定义语言 (XSD) 架构。由于 DataSet 固有的序列化格式为 XML，因此是在层间移动数据出色的媒介，这使 DataSet 成为在远程向 XML Web 服务发送数据和架构上下文以及从 XML Web 服务接收数据和架构上下文的最佳选择。如下：

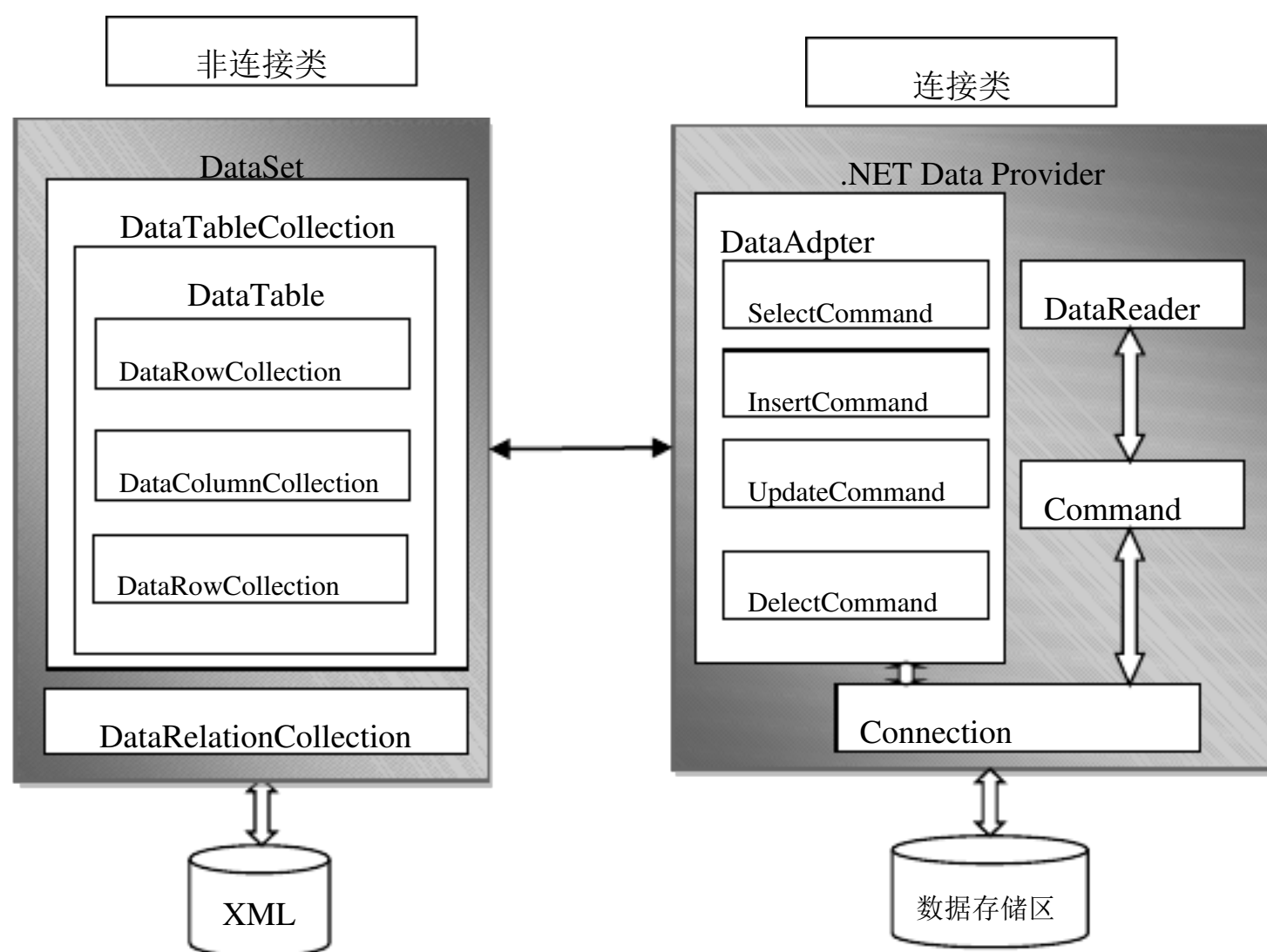


图 1.3

第二章 系统分析

软件需求分析是软件开发期的第一个阶段，也是关系到软件开发的关键步骤。在开发时期之前还有一个计划时期，用以对问题定义和可行性研究。可行性研究涉及的范围较广，除了需求分析还包括其他内容。不言而喻，在实际工作中，总是计划在先，开发在后的。本章主要说明了系统的需求分析、可行性研究，给出了系统的数据流图。

2.1 系统目标

固定资产管理系统是企业管理中的一个重要组成部分，由于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。所以该课题设计的固定资产管理信息系统应具有以下特点：

- 能快捷地进行资产的日常管理
- 能高效地进行资产以及相关人员的查询
- 能简单地进行系统的各项操作

根据以上要求，设计系统的主要功能如下：

- ◇ 资产类别管理：可以对资产添加新的资产类别和资产型号。
- ◇ 库存查询：检索资产类别，一览信息，在资产类别一览中选择一条记录，显示该资产类别下所有资产型号。
- ◇ 资产维护菜单：此菜单中又包括资产登录、临时保存、电脑管理、办公设备管理、车辆管理、配件管理、小额用品管理、软件管理，在这些小项中可以对其进行新建、编辑、删除等操作，允许重复登录。
- ◇ 移库处理：对企业资产和原材料资产进行移库处理。
- ◇ 原材料交付处理：进行原材料资产交付和反交付处理，对现有软件可以执行编辑和删除操作，也可添加新的软件信息，也可根据购入日期期间，

所属类别及购买价格模糊查询。

- ◇ 资产照会：根据条件进行查询；一览中选中一条记录，对现有资产可以执行删除操作；在编辑链接中，可以进入资产更新界面进行资产更新；可以进入资产查看界面；可以进行明细查看。
- ◇ 月资产变动一览：根据购入日期期间，所属类别及购买价格模糊查询，一览中选中一条记录，对现有软件可以执行编辑和删除操作，还可以添加新的软件。
- ◇ 资产报废：查询需要报废的资产，进行报废处理；可以查询已经报废的资产；可以查看报废品维修履历。
- ◇ 资产盘点：根据购入日期期间，所属类别及购买价格模糊查询；一览中选中一条记录，对现有软件可以执行编辑和删除操作；可以添加新购入的软件信息。
- ◇ 设备维修管理：申请维修信息的输入；以请求期间，处理状态做模糊查询；在请求信息一览中，选中一条已对应的信息，给该对应活动填写反馈意见。

2.2 可行性分析

可行性分析也称为可行性研究，是在系统调查的基础上，针对新系统的开发是否具备必要性和可能性，对新系统的开发从技术、经济、社会等方面进行分析和研究，以避免投资失误，保证新系统的开发成功。可行性分析的目的就是用最小的代价在尽可能短的时间内确定问题是否能够解决。该系统的可行性分析包括以下几个方面的内容：

- (1) 经济可行性：主要是对项目的经济效益进行评价，本系统作为公司一个实训项目，具有经济可行性是显而易见的。
- (2) 技术上的可行性：技术上的可行性分析主要分析技术条件能否顺利完成开发工作，软硬件能否满足开发者的需求等。该系统采用 B/S 模式进行开发。结合 Web 技术，数据库服务器选用微软公司的 SQL

Server 数据库，它能够处理大量数据，同时保持数据的完整性并提供许多高级管理功能。它的灵活性、安全性和易用性为数据库编程提供了良好的条件。ASP 技术的应用，使用户可以通过网页向服务器传递特定的信息，服务器可根据用户信息对网页做动态更新或记录用户信息[3]，这也使得网上预约成为可能。选择 MVC2 结构，后台数据库采用 SQL Server，WEB 服务器运行微软的 IIS，以 ASP 作为开发工具，客户端用 IE 浏览器。因此，系统的软件开发平台已经成熟可行。硬件方面，科技日益发展的今天，硬件更新的速度越来越快，容量越来越大，可靠性越来越高，价格越来越低，其硬件平台完全能满足此系统的需求。

- (3) 环境可行性：公司内部都有局域网，员工和管理者均可以在网上处理相关事宜。
- (4) 法律可行性：固定资产管理系统的开发作为校外毕业设计的项目，不会引起任何侵权或其他责任问题，在法律上是可行的。

综上所述，此系统开发目标已明确，在技术和经济等方面都可行。因此，系统的开发是完全可行的。

2.3 需求分析

一个企业的良性发展，避免不了的要涉及到企业资产的有效管理。对于那些技术装备密集型企业，固定资产的管理显得尤为重要。传统的固定资产管理模式无论从质量上还是效率上，都难以适应经营管理新形式的需要。因此，寻找一种简便、高效的管理手段成为必然。在充分研究分析企业固定资产管理的业务需求的基础上，北京首码伟业科技的固定资产管理条码解决方案把条码引入固定资产管理中，开发了一套首码固定资产管理条码管理系统，改变了固定资产清查数据的采集方式，解决固定资产实物清查的瓶颈问题，大大提高清查效率，同时也增加了固定资产的形态方面的管理，有效解决企业资产的管理难题，使企业更轻松、更有效地管理固定资产。如能在公司内部建立固定资产管理，将使各级相关管理人员及有关领导快速查询、统计固定资产情况，实现合理配置资源、决策，提供依据，提高工作效率。

根据无锡华夏计算机公司的实际情况开发的基于 B/S(Browser/Server)的总务管理系统,可以方便管理者对公司的管理,也方便公司员工对与其相关的信息进行查询,在该系统的固定资产模块中,员工对固定资产的使用申请也可通过该系统在局域网内进行申请,相对的提高了工作的效率,也方便了员工之间的沟通。

要设计一个性能良好的管理系统,明确用户的应用环境对系统的要求是首要的和基本的。本系统从以下几个方面对用户要求进行了分析。

2.3.1 管理者的信息需求

改善了传统管理中对固定资产仅从财务角度的管理,记录了固定资产使用的流程及相关责任人;

固定资产流转管理:固定资产的新购,变动,领用,退还,转交等信息一目了然;

及时性:任何时间点管理者可以随时查询数据,不必等待结果型的报表;

多角度的管理:可按日期,作者,部门,状态等多个查询条件,及时多角度了固定资产情况;

异地管理:提高纵向管理的深度和方便性,对异地办事处或者分公司固定资产有效管理。

2.3.2 企业财务人员的工作需求

准确性:所有的固定资产信息,都可以在当前界面查询到关联信息,及时了解每个固定资产使用状况;

降低劳动工作量:根据电子流程单快速检索到凭证所属的时间区段,快速查询凭证,提高员工对财务工作的满意度;

通过异步办公的方式:财务人员随时可查询到异地办事处的资产信息;

内部审计方便:无需翻查纸质凭证,就能够查询全部固定资产流转过程。

2.3.3 员工财务管理需求

自助查询:员工可以随时到系统中查询与自己相关的固定资产的资产清单,

办理，办结；

办理相关申领手续方便：自己可以随时申请领用固定资产，及查询审批流程，无需反复问询财务人员；

流程规范化：降低对业务人员的要求，避免其工作的随意性，责任明晰。

2.3.4 公司业务需求

及时性：随时了解最新的固定资产使用情况；

提高业务流程规范性：通过标准的表单和流程的流转达到规范的业务效果；

方便公司管理者介入跨地域跨部门的业务流程；

开放式系统带来的方便性：便于相关业务人员及时了解有关资产信息。

2.3.5 运行需求

固定资产管理系统的建设，运行是一个长期的过程。网站的运行与维护主要有以下几个方面：

- ✧ WEB 网络应用程序平台的运行与维护。
- ✧ WEB 网络应用程序使用的数据库的添加与维护。
- ✧ Web 网络提供的各类服务实时管理。
- ✧ SQL SERVER 的维护管理。
- ✧ IIS 管理软件的日常维护，ASP.NET 是 Microsoft 提出的以 C#语言创建网页的技术。在以 IIS 作为系统平台的系统中，ASP.NET 是目前首选的技术。
- ✧ 在 WEB 网络应用程序平台上增加新的应用。

为了做好固定资产管理系统的运行与维护，一方面网络应用技术及数据库管理软件必须可靠，设计文档和源代码完整，另一方面，本管理系统的管理员需要有一定的系统软件维护能力。

2.3.6 安全性需求

互联网是一个基于 TCP/IP 协议的对全世界各个国家都开放的网络平台，所

以对各种与互联网联接的网络系统需要较高的安全性。

要防止一些人为的破坏，主要有以下几方面：

- ◇ 数据库服务器与 WEB 服务器分离，更有效的保护了数据库的数据安全，防止来自互联网的恶意攻击对数据库的数据的更改；
- ◇ 防病毒 Web 应用服务器和数据库服务器中的防病毒系统一定要全面考虑，制定多层次、全方位的防毒策略，保证服务器具备预防、检查和清除病毒的能力。
- ◇ WEB 服务器与数据库服务器的通讯全部封装，保护通讯过程中的数据安全。

第三章 系统设计

3.1 系统的功能结构设计

系统按照软件工程的思想方法进行分析和设计，以及技术文档的编写。在力求系统功能尽可能强大的同时做到界面美观友好，结构合理、容错性好。

系统对数据库设计要求是熟练和 SQL Server 数据库开发技术。

由于该系统界面较多，所以在此仅挑选几个界面的基本流程关联图进行介绍。根据详细设计的要求，总务管理系统的界面基本流程关联图如下：

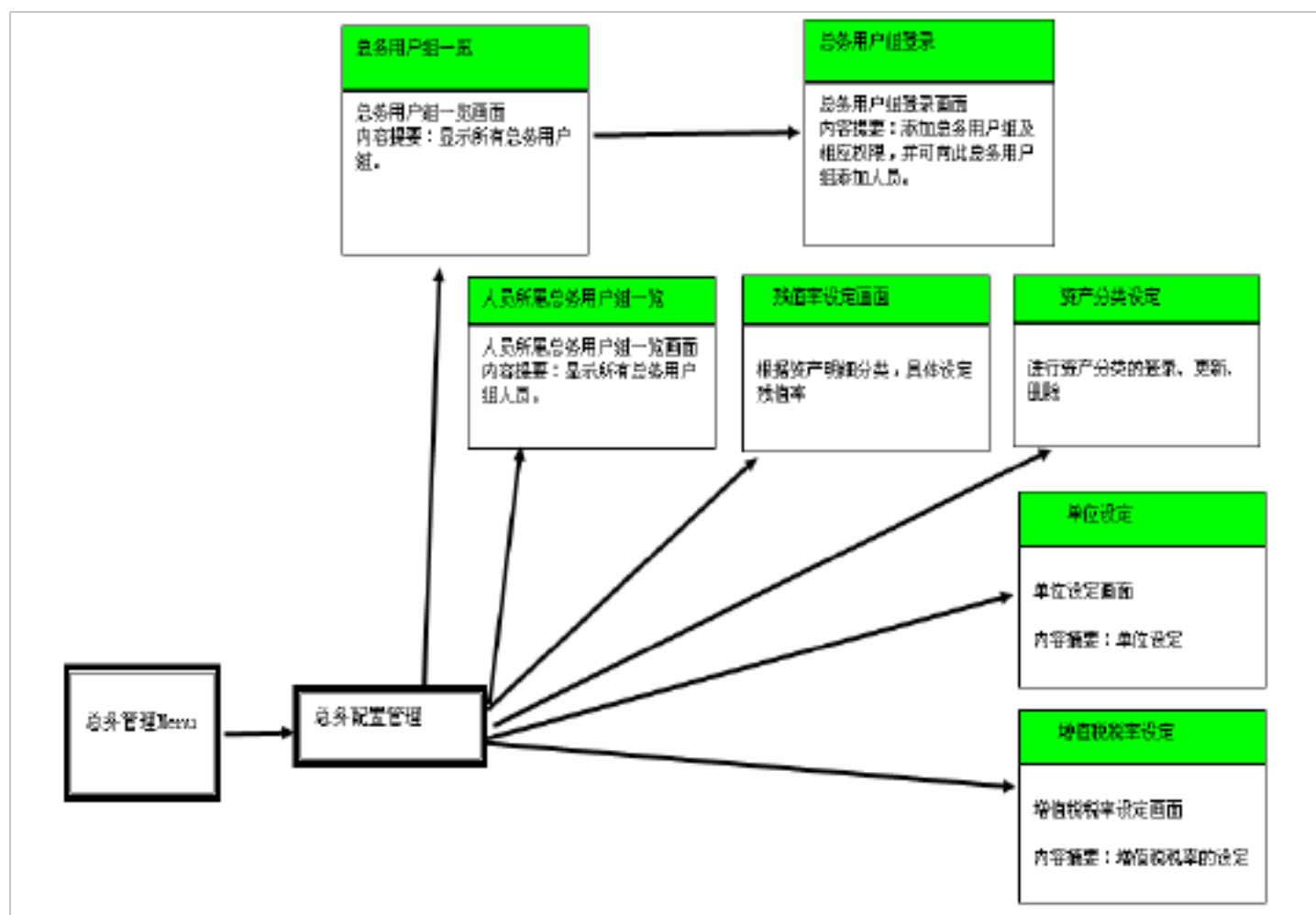


图 3.1 总务管理系统界面基本流程关联图

固定资产管理子系统的界面基本流程关联图如下：

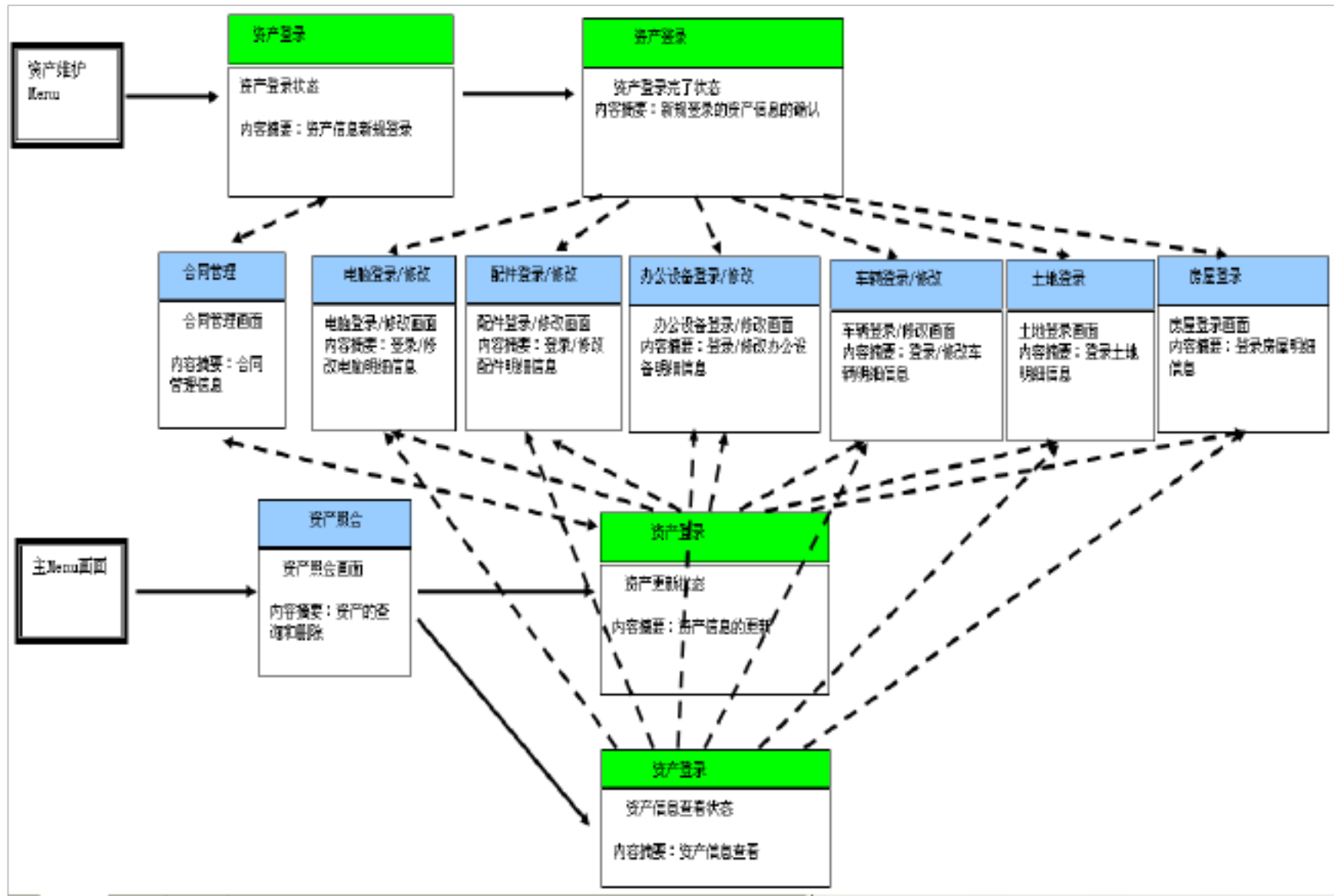


图 3.2 固定资产管理系统界面基本流程关联图

电脑登录界面基本流程关联图如下：

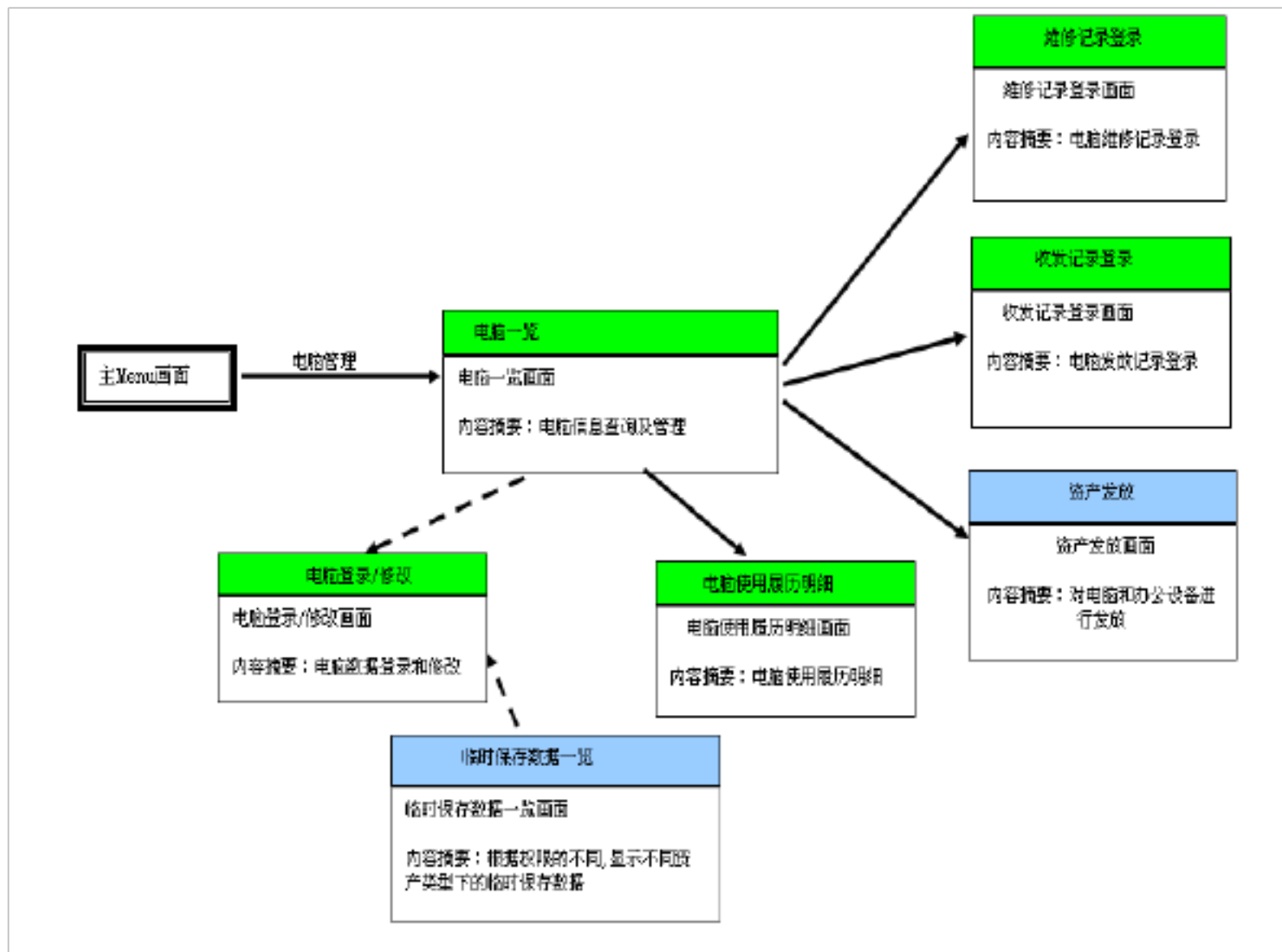


图 3.3 电脑登录管理界面基本流程关联图

3.2 系统的数据库结构设计

3 系统数据流程分析

数据库结构设计的好坏将直接影响系统的运行效率和系统开发进度，合理的数据库结构不仅可以提高系统运行效率，也有利于保证数据的完整性和一致性，有利于系统的实现。

根据系统功能模块结构图和管理流程，绘制的数据流程图。

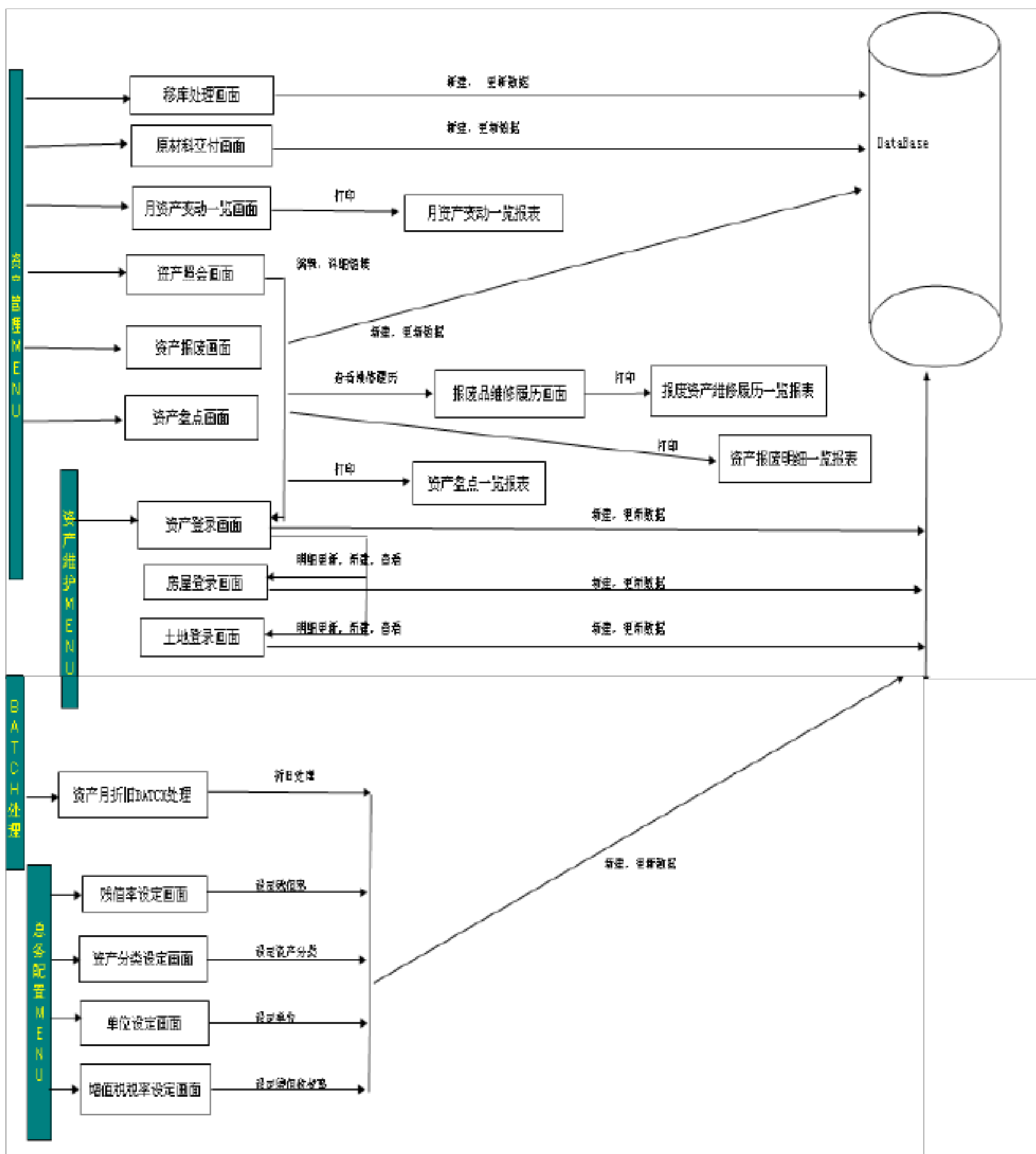


图 3.4 固定资产管理系统流程图

3 系统数据库设计

系统的数据需求包括以下几点：

1) 数据录入和处理的准确性和实时性

数据的输入是否准确是数据处理的前提，错误的输入会导致系统输出的不正确和不可用，从而使系统的工作失去意义。数据的输入来源是手工输入。手工输入要通过系统界面上的安排系统具有容错性，要求对输入人员进行培训。

在系统中，数据的输入往往是大量的，因此系统要有一定的处理能力，以保证迅速的处理数据。

2) 数据的一致性与完整性

由于系统的数据是共享的，在不同的环境中，有好多共享数据，所以如何保证这些数据的一致性，是系统必须解决的问题。要解决这一问题，要有一定的人员维护数据的一致性，在数据录入处控制数据的去向，并且要求对数据库的数据完整性进行严格的约束。对于输入的数据，要为其定义完整性规则，如果不能符合完整性约束，系统应该拒绝该数据。

3) 数据的共享与独立性

整个系统的数据是共享的。然而，从系统开发的角度上看，共享会给设计和调试带来困难。因此，应该提供灵活的配置，使各个分系统能够独立运行，而通过人工干预的手段进行系统数据的交换。这样，也能提供系统的强壮性。

数据库逻辑设计：根据固定资产管理系统的需求，得出以下数据表，由于该系统所用的数据库比较庞大，所以在这里就列出这几张表：

表 3.1 电脑表

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NU11 |
|-----|------|---------------|---------|-----|-----|----------|
| 1 | 合同编号 | ContractID | varchar | 50 | | |
| 2 | 电脑编号 | ComputerID | varchar | 25 | PK | ✓ |
| 3 | 电脑名称 | ComputerName | varchar | 100 | | |
| 4 | 电脑品牌 | ComputerBrand | varchar | 20 | | |
| 5 | 员工编号 | EmployeeID | varchar | 20 | | |
| 6 | 硬盘型号 | DiscID | varchar | 20 | | |
| 7 | 硬盘大小 | DiscSize | varchar | | | |
| 8 | 内存型号 | RAMID | varchar | | | |
| 9 | 内存大小 | RAMSize | varchar | | | |
| 10 | 主板型号 | BoardID | varchar | 20 | | |

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NULL |
|-----|---------|-----------------|-----------|-----|-----|----------|
| 11 | CPU 型号 | CPUID | varchar | 20 | | |
| 12 | CPU 频率 | CPUHZ | varchar | | | |
| 13 | 显卡型号 | DisplayCardID | varchar | 20 | | |
| 14 | 显示器型号 | MonitorID | varchar | 20 | | |
| 15 | 网卡型号 | NetCardID | varchar | 20 | | |
| 16 | 电源型号 | PowerID | varchar | 20 | | |
| 17 | 操作系统 | OS | varchar | 100 | | |
| 18 | 应用软件 | App | varchar | 50 | | |
| 19 | 杀毒软件 | AntiVirus | varchar | 50 | | |
| 20 | OA 软件 | OA | varchar | 50 | | |
| 21 | 其他软件 | OtherSoftware | varchar | 50 | | |
| 22 | 电脑来源 | ComputerOrigin | varchar | 1 | | |
| 23 | 电脑状态 | ComputerStatus | varchar | 1 | | |
| 24 | 购买价格 | ComputerPrice | money | | | |
| 25 | 购买日期 | PurchaseDate | datetime | | | |
| 26 | 入库日期 | WarehouseDate | datetime | | | |
| 27 | 最新收发时间 | GrantDate | datetime | | | |
| 28 | 报废时间 | ScrapDate | datetime | | | |
| 29 | 类别标志 | KindFlg | varchar | 2 | | |
| 30 | 删除标志 | DeleteFlg | varchar | 1 | | |
| 31 | 备注 | Note | varchar | | | |
| 32 | 预备字段 1 | ReservedNumber1 | numeric | 20 | | |
| 33 | 预备字段 2 | ReservedNumber2 | numeric | 20 | | |
| 34 | 预备字段 3 | ReservedNumber3 | numeric | 20 | | |
| 35 | 预备字段 4 | ReservedNumber4 | numeric | 20 | | |
| 36 | 预备字段 5 | ReservedNumber5 | numeric | 20 | | |
| 37 | 预备字符串 1 | ReservedString1 | varchar | 20 | | |
| 38 | 预备字符串 2 | ReservedString2 | varchar | 20 | | |
| 39 | 预备字符串 3 | ReservedString3 | varchar | 20 | | |
| 40 | 预备字符串 4 | ReservedString4 | varchar | 20 | | |
| 41 | 预备字符串 5 | ReservedString5 | varchar | 20 | | |
| 42 | 创建者 | CreatedUserID | varchar | | | √ |
| 43 | 创建日期 | CreatedDate | datetime | | | √ |
| 44 | 更新者 | UpdateUserID | varchar | | | √ |
| 45 | 更新日期 | UpdatedDate | datetime | | | √ |
| 46 | 行版本 | RowVer | timestamp | | | |
| 47 | 其他 | Other | varchar | 100 | | |

表 3.2 车辆表

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NULL |
|-----|-----|-----------|---------|----|-----|----------|
| 1 | 车牌号 | VehicleID | varchar | 20 | | √ |
| 2 | 编号 | ID | int | | | √ |

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NULL |
|-----|---------|-----------------|-----------|----|-----|----------|
| 3 | 车辆名称 | VehicleName | varchar | 20 | | |
| 4 | 发动机编号 | EngineID | varchar | 25 | | |
| 5 | 车架编号 | VehicleFrameID | varchar | 25 | | |
| 6 | 购买日期 | BuyDate | datetime | | | |
| 7 | 购买价格 | BuyPrice | money | | | |
| 8 | 车辆状态 | VehicleStatus | varchar | 2 | | |
| 9 | 报废时间 | ScrapDate | datetime | | | |
| 10 | 备注 | Note | varchar | 50 | | |
| 11 | 预备字段 1 | ReservedNumber1 | numeric | 20 | | |
| 12 | 预备字段 2 | ReservedNumber2 | numeric | 20 | | |
| 13 | 预备字段 3 | ReservedNumber3 | numeric | 20 | | |
| 14 | 预备字段 4 | ReservedNumber4 | numeric | 20 | | |
| 15 | 预备字段 5 | ReservedNumber5 | numeric | 20 | | |
| 16 | 预备字符串 1 | ReservedString1 | varchar | 20 | | |
| 17 | 预备字符串 2 | ReservedString2 | varchar | 20 | | |
| 18 | 预备字符串 3 | ReservedString3 | varchar | 20 | | |
| 19 | 预备字符串 4 | ReservedString4 | varchar | 20 | | |
| 20 | 预备字符串 5 | ReservedString5 | varchar | 20 | | |
| 21 | 删除标志 | DeleteFlg | varchar | 1 | | |
| 22 | 创建者 | CreatedUserID | varchar | 10 | | √ |
| 23 | 创建日期 | CreatedDate | datetime | | | √ |
| 24 | 更新者 | UpdateUserID | varchar | 10 | | √ |
| 25 | 更新日期 | UpdatedDate | datetime | | | √ |
| 26 | 行版本 | RowVer | timestamp | | | |

表 3.3 办公设备表

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NULL |
|-----|--------|------------------|----------|-----|-----|----------|
| 1 | 设备编号 | OeID | varchar | 25 | PK | √ |
| 2 | 设备名称 | OeName | varchar | 20 | | |
| 3 | 合同编号 | ContractID | varchar | 50 | | |
| 4 | 资产类别编号 | AssetsKindID | int | | | |
| 5 | 资产型号编号 | AssetsKindTypeID | int | | | |
| 6 | 设备来源 | OeSource | varchar | 20 | | |
| 7 | 购买日期 | PurchaseDate | datetime | | | |
| 8 | 购买价格 | PurchasePrice | money | | | |
| 9 | 资产状态 | OeStatus | varchar | 2 | | |
| 10 | 物理位置 | OePosition | varchar | 20 | | |
| 11 | 报废时间 | ScrapDate | datetime | | | |
| 12 | 备注 | Note | varchar | 200 | | |
| 13 | 预备字段 1 | ReservedNumber1 | numeric | 20 | | |
| 14 | 预备字段 2 | ReservedNumber2 | numeric | 20 | | |
| 15 | 预备字段 3 | ReservedNumber3 | numeric | 20 | | |

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NULL |
|-----|---------|-----------------|----------|----|-----|----------|
| 16 | 预备字段 4 | ReservedNumber4 | numeric | 20 | | |
| 17 | 预备字段 5 | ReservedNumber5 | numeric | 20 | | |
| 18 | 预备字符串 1 | ReservedString1 | varchar | 20 | | |
| 19 | 预备字符串 2 | ReservedString2 | varchar | 20 | | |
| 20 | 预备字符串 3 | ReservedString3 | varchar | 20 | | |
| 21 | 预备字符串 4 | ReservedString4 | varchar | 20 | | |
| 22 | 预备字符串 5 | ReservedString5 | varchar | 20 | | |
| 23 | 删除标志 | DeleteFlg | varchar | 1 | | |
| 24 | 创建者 | CreatedUserID | varchar | 10 | | √ |
| 25 | 创建日期 | CreatedDate | datetime | | | √ |
| 26 | 更新者 | UpdateUserID | varchar | 10 | | √ |

表 3.4 资产类别表

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT Null |
|-----|------------|------------------|-----------|----|-----|----------|
| 1 | 资产类别编号 | AssetsKindID | int | | | |
| 2 | 资产类别 | CategoryName | varchar | 20 | | |
| 3 | 资产类型 | AssetsCategoryID | varchar | 20 | | |
| 4 | 类别编码前缀文字 | PrefixText | varchar | 20 | PK | √ |
| 5 | 类别编码后缀数字个数 | SuffixNumber | int | | | |
| 6 | 备注 | Note | varchar | 50 | | |
| 7 | 预备字段 1 | ReservedNumber1 | numeric | 20 | | |
| 8 | 预备字段 2 | ReservedNumber2 | numeric | 20 | | |
| 9 | 预备字段 3 | ReservedNumber3 | numeric | 20 | | |
| 10 | 预备字段 4 | ReservedNumber4 | numeric | 20 | | |
| 11 | 预备字段 5 | ReservedNumber5 | numeric | 20 | | |
| 12 | 预备字符串 1 | ReservedString1 | varchar | 20 | | |
| 13 | 预备字符串 2 | ReservedString2 | varchar | 20 | | |
| 14 | 预备字符串 3 | ReservedString3 | varchar | 20 | | |
| 15 | 预备字符串 4 | ReservedString4 | varchar | 20 | | |
| 16 | 预备字符串 5 | ReservedString5 | varchar | 20 | | |
| 17 | 删除标志 | DeleteFlg | varchar | 1 | | |
| 18 | 创建者 | CreatedUserID | varchar | 10 | | √ |
| 19 | 创建日期 | CreatedDate | datetime | | | √ |
| 20 | 更新者 | UpdateUserID | varchar | 10 | | √ |
| 21 | 更新日期 | UpdatedDate | datetime | | | √ |
| 22 | 行版本 | RowVwe | timestamp | | | |

表 3.5 房屋表

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NULL |
|-----|------|------------|---------|----|-----|----------|
| 1 | 房屋代码 | BuidID | varchar | 2 | | √ |
| 2 | 编号 | ID | int | | PK | √ |
| 3 | 地目代码 | TerraID | varchar | 2 | | |
| 4 | 房屋状态 | BuidStatus | varchar | 2 | | |

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NULL |
|-----|---------|------------------|-----------|----|-----|----------|
| 5 | 使用者 | UserName | varchar | 20 | | |
| 6 | 单位 | Units | varchar | 2 | | |
| 7 | 地上层数 | UpgroundLayer | int | | | |
| 8 | 地下层数 | UndergroundLayer | int | | | |
| 9 | 房屋门牌号码 | BuidDoorplate | varchar | 50 | | |
| 10 | 登记日期 | RegisterDate | datetime | | | |
| 11 | 房产证编号 | BuidpropertyID | varchar | 50 | | |
| 12 | 建筑面积 | ConstructionArea | float | | | |
| 13 | 删除标识 | DeleteFlg | varchar | 1 | | |
| 14 | 备注 | Note | varchar | 50 | | |
| 15 | 预备数字 1 | ReservedNumber1 | numeric | 10 | | |
| 16 | 预备数字 2 | ReservedNumber2 | numeric | 10 | | |
| 17 | 预备数字 3 | ReservedNumber3 | numeric | 10 | | |
| 18 | 预备数字 4 | ReservedNumber4 | numeric | 10 | | |
| 19 | 预备数字 5 | ReservedNumber5 | numeric | 10 | | |
| 20 | 预备字符串 1 | ReservedString1 | varchar | 50 | | |
| 21 | 预备字符串 2 | ReservedString2 | varchar | 50 | | |
| 22 | 预备字符串 3 | ReservedString3 | varchar | 50 | | |
| 23 | 预备字符串 4 | ReservedString4 | varchar | 50 | | |
| 24 | 预备字符串 5 | ReservedString5 | varchar | 50 | | |
| 25 | 创建者 | CreatedUserID | varchar | 10 | | √ |
| 26 | 创建日期 | CreatedDate | datetime | 10 | | √ |
| 27 | 最终更新者 | UpdateUserID | varchar | 10 | | √ |
| 28 | 最终更新日 | UpdatedDate | datetime | 10 | | √ |
| 29 | 行版本 | RowVer | timestamp | 8 | | |

表 3.6 资产分类设定表

| No. | 项目名 | 字段名 | 类型 | 位数 | Key | NOT NULL |
|-----|---------|-----------------|---------|----|-----|----------|
| 1 | 资产分类 ID | DeviceKindID | int | | PK | √ |
| 2 | 资产分类名称 | DeviceKindName | varchar | 50 | | |
| 3 | 删除标识 | DeleteFlg | varchar | 1 | | |
| 4 | 备注 | Note | varchar | 50 | | |
| 5 | 预备数字 1 | ReservedNumber1 | numeric | 10 | | |
| 6 | 预备数字 2 | ReservedNumber2 | numeric | 10 | | |
| 7 | 预备数字 3 | ReservedNumber3 | numeric | 10 | | |
| 8 | 预备数字 4 | ReservedNumber4 | numeric | 10 | | |
| 9 | 预备数字 5 | ReservedNumber5 | numeric | 10 | | |
| 10 | 预备字符串 1 | ReservedString1 | varchar | 50 | | |
| 11 | 预备字符串 2 | ReservedString2 | varchar | 50 | | |
| 12 | 预备字符串 3 | ReservedString3 | varchar | 50 | | |
| 13 | 预备字符串 4 | ReservedString4 | varchar | 50 | | |
| 14 | 预备字符串 5 | ReservedString5 | varchar | 50 | | |

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/168100040040006060>