

【软考达人】

# 软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



**微信扫一扫，立马获取**



**6W+ 免费题库**



**免费备考资料**

PC版题库: [ruankaodaren.com](http://ruankaodaren.com)

### 摘要：

本人于 2018 年 1 月参与了中石化 XX 油田 XX 采油厂“用电管理系统”的项目建设，该系统建设目标是实现分单位、分线路、分系统评价、优化、考核，全面提升采油厂用电管理水平。在该项目组中我担任系统架构师一职，主要负责系统整体架构设计。本文以该项目为例，讨论系统三层架构的设计过程，重点讨论系统在设计过程和在设计实施过程中碰到的一些问题和解决方法，最后说明采用三层架构所带来的效果，以及可以改进的地方。在我的带领下，项目实施非常顺利，并获得采油厂领导及厂、区多级用户的一致好评，该项目也成功获得了当年度分公司科技进步创新一等奖。

### 正文：

“用电管理系统”项目是采油厂能源管控中心系统的一个子系统。能源管控中心是中石化集团公司十三五规划中的“能效倍增”计划在胜利油田分公司的示范应用项目，该示范项目能够在实现企业节能目标管理、能源计量统计、节能潜力识别、能效分析优化的同时，有效支撑企业实施节能技术改造、促进企业用能水平不断提升

“用电管理系统”的建设目标是建立覆盖厂、区两级用电管理一体化体系，实现分单位、分线路、分系统评价、优化、考核，达到电网运行质量实时监控、异常情况精准管控、能耗总量全面受控，按照运行产量的方式运行电量，全面提升采油厂用电管理水平。该项目功能设计参考 PDCA 闭环管理的理念，共设计包括用电计划、用电分析、用电优化、用电考核、设备管理等五大功能模块。

我作为单位技术骨干之一，主持并参与了项目计划制定、需求分析、设计、编码、设计、测试等阶段的工作。由于油田系统对安全性、可靠性、可用性和扩展性要求很高，我选择了三层 B/S 架构作为该系统的软件体系架构，下面，我将分层次详细介绍三层 B/S 软件体系结构的设计过程。

(1) 表示层采用了 HTML5+Vue +BootStrap 等前端技术。

由于用户提出全厂的电力系统拓扑图能够自由定制，我们综合考虑多种方案后选择了使用 HTML5 Canvas 技术来实现电力系统拓扑图的自由定制。HTML5 是互联网的下一代标准，是构建以及呈现互联网内容的一种语言方式，被认为是互联网的核心技术之一。最终采用 HTML5 Canvas 技术实现的拓扑图的非常轻巧，性能出色，能够支持上万图元，操作流畅，并且支持矢量图形，无极缩放等。Vue 是一套用于构建用户界面的渐进式 JavaScript 框架。Vue 与其他 JS 库不同，它的核心库只关注视图层，非常容易学习，非常容易且方便与第三方库或既有项目整合。项目组的两个新接触 vue 的开发人员在极短时间内就掌握了相关开发技巧，为项目前端的快速开发打下了良好的基础。Bootstrap 是最受欢迎的基于 HTML、CSS、JS 的框架之一，是一个用于快速开发 Web 应用程序和网站的前端框架。我们采用基于 Bootstrap 的系统界面样式及响应式的展现形式得到了客

户的称赞，为后续项目工作的开展打下了良好的基础。

(2) 中间业务逻辑层采用 SpringMVC+Apache Shiro，并使用 ehcache 缓存技术以提高访问效率。

我们选择了当前最优秀的 MVC 框架之一的 SpringMVC 来简化开发，使开发人员可以专注于表示逻辑和业务逻辑的开发工作，SpringMVC 是一种实现了 Web MVC 设计模式的请求驱动类型的轻量级 Web 框架。通过框架的使用，缩短了开发周期，减少了开发费用和维护费用，提高了开发的成功率。基于安全性方面考虑，我们选择 Apache Shiro，它是一个功能强大且易于使用的 Java 安全框架，为开发人员提供了一个直观而全面的解决方案，用于身份验证、授权、加密和会话管理。通过使用 SpringMVC+Apache Shiro 的分层设计，实现了各层次低耦合，高内聚，并严格遵循了 web 安全的相关规范，通过前后台双重验证，参数编码传输，密码 SHA-256 加密存储，shiro 权限验证等手段，从根本上避免了 SQL 注入，XSS 攻击，CSRF 攻击等常见的 web 攻击手段。为防止频繁读取数据库导致系统性能下降问题，我们选择采用开源的 J2Cache 两级缓存框架，即一级采用 Ehcache，二级采用 Redis 缓存技术，通过该框架的使用，提高了系统性能和访问效率。

(3) 数据层选择 MyBatis，数据库选择 Oracle 11g。

数据层方面我们选择了 MyBatis 作为持久层框架，它可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs(Plain Ordinary Java Object, 普通的 Java 对象)映射成数据库中的记录。MyBatis 简单易学、灵活并且解除了 sql 与程序代码的耦合，通过提供 DAO 层，将业务逻辑和数据访问逻辑分离，使系统的设计更清晰，更易维护，更易单元测试。这种 sql 和代码分离的方式，极大提高了系统的可维护性。另外基于安全性和稳定性方面考虑，我们选择了 Oracle 11g 作为数据库。

但是，在设计的过程中也遇到了一些困难，我们主要采取了以下的办法来解决：

(1) 电力拓扑库设计。项目启动伊始所面临最大的难题是厘清全厂一百多条供电线路及线路上所带的提升、注入、集输、转供、后辅等多类用电数据，实现线路用电数据的闭合。经过对在用系统及数据库的多次调研分析发现，该库共需涉及 EPBP 电力库，源头库，PCS 数据库，厂用电管理库，设备档案库以及分线计量库等 6 类不同数据源，数据关系极其复杂。由于项目组比较缺乏实际的现场生产经验，对这些复杂数据源毫无头绪，无从下手。但通过项目组主动与甲方多次采用用户访谈、发放调查问卷、联合需求计划会议及现场观摩的方式进行沟通，对多类业务数据有了较为清晰的认识，经过项目组的共同努力与甲方的积极配合，中间又经历了数次返工、改版，最终成功完成了电网拓扑数据库设计。完成后的拓扑库共设计六十余张数据表，该设计逻辑关系清晰、具



有良好的可扩展性，为后续软件开发工作的顺利进行打下了坚实的基础。系统上线后，全厂每天的实际用电约在 110 万 kWh 左右，通过数据闭合后自动采集的用电数据达到了 100 万 kWh 左右，线损约在 10% 左右，基本满足线路设计的线损要求，实际运行效果良好，达到了厂领导提出的“像运行产量一样运行电量”的要求。

(2) 多系统数据集成。采油厂用电管理系统的涉及的业务数据分散在不同系统库中。目前油田油水井日度生产数据主要使用源头数据库存储，实时数据采用现有 PCS 系统数据库存储，集输相关数据使用采油厂自建的分队计量系统库存储，注水相关数据采用油田统一搭建的 EPBP 系统库存储，电力相关数据由电力分公司管理和存储。为了解决不同库之间的数据差异问题，我们对相关业务数据进行统一归集，通过 ETL 的方式，每天自动将不同业务系统的数据进行抽取、清洗转换，然后加载到用电管理系统数据库，从而将目前分散、零乱、标准不统一的数据整合到一起，为系统的决策提供分析依据。

由于软件三层架构设计得当，并采取了有效的措施去解决设计中遇到的问题，用电管理系统最后按计划完成并顺利投入运行，不但保证了系统的开放性、可用性和互用性，取得了良好的经济效益和管理效益，而且我的软件三层架构设计得到了采油厂领导和厂、区两级用户的一致认同与称赞，为系统的试运行打下了良好的基础，该项目也获得了当年度分公司科技进步创新一等奖。

在总结经验的同时，我也看到了我在软件三层架构设计中的不足之处：

首先：找标杆算法过于简单。由于油田现有数据质量限制，存在很多空数据、假数据，我们的找标杆算法虽然在理论上经过验证，但在实际应用过程中发现，四项以上的参数组合经常无法返回正确结果，这样对需要参考对比结果的技术人员非常不友好。后续我们经过多次与相关关键客户分析讨论，终于确定了采用多项单参数最优的方式进行标杆筛选，获得了相关客户的认可和领导的肯定。

其次，数据库层面存在很多无效的存储过程与定时任务，对数据库性能造成了影响。由于是在原系统基础上开发的子系统，原系统中存在大量的存储过程与定时任务，我们在用电管理系统开发过程中遇到过多次数据库 session 过多造成的问题，如存储过程无法编译、数据库无法访问等。后续经过项目组的不懈努力，通过将大部分无效存储过程进行重构整合，对定时任务进行重新分配的方式，解决这个困扰大家已久的难题。

对于第一种情况，通过及时的分析沟通和对算法的优化，保证了中间层的业务逻辑处理的准确性和表现层结果视图显示的完整性，提高了用户体验质量，而第二种情况，对这种高水平高价值的遗留系统问题，我们采用了重构整合的方式进行了改造，保证了系统的延续性。为了更好地开展系统分析设计工作，我也将在以后的工作实践中不断地学习，提高自身素质和能力，为我国软件事业贡献自己微薄的力量。