**学 士 学 位 论 文**

**题 目 博林特电梯系统**

**学 生 XXX**

**指导教师 XX讲师**

**年 级 2012级**

**专 业 计算机科学与技术**

**系 别 计算机科学与技术**

**学 院 计算机科学与信息工程**

哈尔滨师范大学

2016年5月

**摘要：**高层的快速崛起。使电梯成为人们生活中不可或缺的一部分，每天我们都要乘坐这些电梯，无论是客梯、货梯、或者是扶梯。出了危险都不是小事情。都是能威胁我们的人身安全的。即便是不出现大的事故，总是维修也会给我们的生活带来不便。为了解决以上的问题，我从企业管理的角度出发。为电梯企业定制开发EAM设备管理系统，让企业更好的管理电梯的生命周期，及时的处理电梯事故，定期的维修保养电梯。使用市面上比较常用的java语言，基于B/S的web应用。为人们带来方便的同时也提高企业对员工的管理能力。提高生产力，创造更多的价值。

**关键字：**电梯、B/S、EAM、java

**目 录**

**[第一章 绪论 6](#_Toc16598)**

[1.1选题背景 6](#_Toc31668)

[1.2选题意义 6](#_Toc11978)

[1.3系统实现的目标 7](#_Toc24233)

**[第二章 总体设计](#_Toc22293)** [7](#_Toc22293)

[1.1 可行性分析 7](#_Toc27492)

[1.1.1 技术可行性 7](#_Toc13486)

[1.1.2经济可行性 7](#_Toc2396)

[1.1.3社会可行性 7](#_Toc28705)

[1.2架构设计 8](#_Toc27156)

[1.2.1 框架的选择 8](#_Toc7013)

[2.2.2自我框架封装 12](#_Toc27682)

**[第三章UI设计](#_Toc17740)** [12](#_Toc17740)

[3.1社会ui分析 12](#_Toc11099)

[3.2Easyui的选择 13](#_Toc26999)

**[第四章 数据库设计](#_Toc20340)** [15](#_Toc20340)

[4.1数据库设计范式 15](#_Toc10492)

[4.2Er模型图 16](#_Toc29848)

[4.3数据库表 16](#_Toc22809)

**[第五章 功能设计](#_Toc9667)** [24](#_Toc9667)

[5.1功能总述 24](#_Toc8366)

[5.2核心功能讲解 25](#_Toc24291)

**[第六章 各模块详细设计](#_Toc25170)** [26](#_Toc25170)

[6.1 权限管理 26](#_Toc32012)

[6.2设备管理主数据 26](#_Toc24635)

[6.3设备保养计划管理流程 26](#_Toc30020)

[6.4维修工单执行流程 27](#_Toc12380)

[6.5知识库管理 28](#_Toc11949)

[6.6统计分析 29](#_Toc2650)

[6.7工单管理 29](#_Toc802)

[6.8设备台账管理 29](#_Toc1613)

**[第七章 测试](#_Toc16759)** [29](#_Toc16759)

[7.1搜索功能 29](#_Toc23143)

[7.2添加、修改功能 30](#_Toc9982)

[7.3删除功能 30](#_Toc22263)

[7.4注册、登陆模块 31](#_Toc12804)

[7.4.1注册功能 31](#_Toc7011)

[7.4.2登陆 功能 31](#_Toc32097)

**[第八章 所遇问题与解决](#_Toc5132)** [32](#_Toc5132)

[8.1 问题一hibernate不发出更新语句 32](#_Toc5451)

[8.2 hibernate查询问题 32](#_Toc4155)

**[第九章 总结 34](#_Toc30738)**

**[致 谢 35](#_Toc30854)**

**[参考文献 36](#_Toc18801)**

**[Abstract 37](#_Toc18801)**

**[附 录 38](#_Toc12621)**

# 

# 第一章 绪论

## 1.1选题背景

随着社会的发展，电梯已经变成了生活中的常见事物，在各大高层，商场都有不少的电梯。随之电梯数量的增加，电梯的检修维护等问题，也应运而生。

如三湘都市报中报道的一样，本报7月22日讯 “没有电梯，我们住高楼层的人每天爬楼，实在受不了了。” 7月份以来，家住长沙市天心丽城小区的业主们苦不堪言，因为高达16层的房子，两个电梯都坏了，大家只能每天爬楼梯出门回家。7月21日，居民联系上本报，希望能解决这个难题。

还有，本人也自己遇到了这个问题，我实习锁住的公寓中，一共有2部电梯，坏了以后，其中一部用了1周左右的时间修理好了。这一周中很多人都是爬15-20层楼，对于年轻人还好，对于老年人实在有点吃不消。在其中一部电梯修好后，另一部电梯，2个月了，一直都没有人来修。

还有很多诸如此类的新闻，可见电梯的及时维修，电梯的及时报修，电梯的维修人员分配都成了一个大的问题。

为了应对和解决这些问题，我们采访了柏林特电梯有限公司。了解他们的日常管理办法，从中发现不足进行弥补，已达到能实时为广大电梯用户解决电梯问题。

在采访中了解到，他们不是不进行电梯的修理，而是电梯的报修是有物业来报修的，物业报上来之后，就会安排人员进行去检修维护。所有报修都是录入员进行手动录入的。然后用电话，或者微信的方式通知维修人员。维修人员进行检修。在报告给管理人员。管理人员审查在报给录入人员。录入人员进行维修完成的登记。

以上是电梯发展的现状，下面我们再来说说EAM系统，设备管理系统，在市场上产生和流行已经很多年，尤其是国家的电力行业，电力设备的宕机对人员和经济都有非常的影响。所以我国电力部门，在很早之前就已经开始使用了EAM系统，这些年取得的很好的效果。现在其他设备的价值都在提升。EAM系统的发展也是随着智能家居等等硬件系统的提升而提升着。

## 1.2选题意义

如上文了解到的柏林特电梯有限公司的情况，他们存在很多潜在的问题。

通知流程繁琐，每个人通知下个人都需要一个时间，这个时间对于一个来说还好，加在一起就很长，增加的维修的时间。

人员录入问题，纸质文件，不容易修改。存放有大的问题，在用户想要查询自己报修的记录是很麻烦，常常出错。

还有如果维修人员不能一次维修好，后续工作很是繁琐。维修人员一直在反馈，等待管理者消息，不能把心思放在维修上。

多点包维修的时候，不能合理的分配维修人员的路线，人员一直在各个地方奔跑，不能实现就近维修提高效率。

针对以上问题，我们可以用电梯管理软件来解决此类问题。

据了解，在西方发达国家，[软件开发平台](http://baike.baidu.com/view/2788316.htm" \t "http://baike.baidu.com/_blank)是被大、中、小公司所广泛采用的一种软件开发形式，在企业管理软件占有40%左右的[市场份额](http://baike.baidu.com/view/386620.htm" \t "http://baike.baidu.com/_blank)。而在我国，由于经济环境和传统管理方式的不同，还没有被广大客户所认同。

很多公司还在使用传统的纸质人工处理方案，在面对社会上的大而杂的数据的处理上会有很大的问题。

此软件不只在处理柏林特一家公司的问题。而是以点变面，开发一套EAM管理系统，

EAM管理系统。旨在在商业竞争日益激烈的今天，对于拥有高价值资产的企业来说，[设备维护](http://baike.baidu.com/view/1228435.htm" \t "http://baike.baidu.com/_blank)已不再局限于成本范畴，更成为获取利润的战略工具，EAM系列产品使这一目标得以实现。

## 1.3系统实现的目标

EAM概念说明。EAM系统是一套针对资产密集型企业的信息化解决方案的资产[管理系统](http://www.qykh2009.com/" \t "http://www.qykh2009.com/_blank)，是一套完整的解决方案的总称。对于企业管理者来说，企业生产就是追求资产可利用率不断提高、企业运行维护成本日趋减少，从而优化企业维修资源为核心，通过信息化途径，合理安排维修计划及相关资源与活动。因此，企业引进或投资EAM体系是自身发展必然结果。

但EAM系统是也一种特殊的投资，它的投入与其他信息化项目一样，企业也需要谨慎投入资金，要实实在在取得相应的收获、等比例的回报就要建立EAM评价体系。

组成EAM评价体系主要部分是：评价指标体系、评价制度体系和评价组织体系。EAM评价体系的核心内容是评价指标体系，该体系反映的是从需求、到实施再到应用的全过程，包括与之紧密关联的管理思想、管理模式、管理基础、管理方法、管理机制、业务流程、组织结构、工薪、工种、组织机构、培训管理、工时管理、绩效管理及反映企业综合能力和过程控制状况的相关经济指标等评价内容，并考虑企业的战略效益、经济效益、技术指标、社会效益等四方面的普遍因素。

其实，EAM体系除了其可以给企业创造应有价值外，也有其评价体系存在的必然原因。EAM体系属于综合性的项目，涉及到企业的方方面面，很难用一个绝对化的指标来表明两三年内其产生的回报，会给企业管理经营者的一种错觉，即在EAM项目只有投入、没有产出，使得管理者不能明确要害，很容易盲目加大投入，亦或直接拒绝不再追加投入。

EAM项目的投资也并非一次性的过程，具有很高的隐性成本，除了购买硬件、软件费用和实施费用以外，还要投入维护和服务费用，特别是由资产管理信息化而引发的对企业整体人员基本素质的要求更是需要长期的培训。

因此，建议EAM评价体系就能明确企业项目驱动因素和企业需求；可优化企业资产管理；可降低库存成本和零件采购成本；量化可考核的业务处理指标，实现实施EAM全过程的绩效管理；可预防性设备日常维护和维修数据库，降低实施风险，提高EAM实施的成功率。

EAM以资产模型、设备台帐为基础，强化[成本核算](http://baike.baidu.com/view/45392.htm" \t "http://baike.baidu.com/_blank)的[管理思想](http://baike.baidu.com/view/141611.htm" \t "http://baike.baidu.com/_blank)，以工单的创建、审批、执行、关闭为主线，合理、优化地安排相关的人、财、物资源，将传统的被动检修转变为积极主动的预防性维修，与实时的数据采集系统集成，可以实现[预防性维护](http://baike.baidu.com/view/1435861.htm" \t "http://baike.baidu.com/_blank)。通过跟踪记录企业全过程的维护历史活动，将维修人员的个人[知识转化](http://baike.baidu.com/view/2926100.htm" \t "http://baike.baidu.com/_blank)为企业范围的智力资本。集成的工业流程与[业务流程](http://baike.baidu.com/view/1368133.htm" \t "http://baike.baidu.com/_blank)配置功能，使得用户可以方便地进行系统的授权管理和应用的客户化改造工作。

实现目标：

1.量化[设备资产管理](http://baike.baidu.com/view/3613282.htm" \t "http://baike.baidu.com/_blank)，规范其[工作流程](http://baike.baidu.com/view/3992935.htm" \t "http://baike.baidu.com/_blank)

2.实现[资产管理信息化](http://baike.baidu.com/view/3977931.htm" \t "http://baike.baidu.com/_blank)，更有效地配置生产设备、人员及其他资源

3.借助于系统的帮助，每位维修管理人员可以管理更多的设备

4.改善工人的安全保障，促进规程的执行，减少停产时间

5.提高[有效工作时间](http://baike.baidu.com/view/1588784.htm" \t "http://baike.baidu.com/_blank)

6.减少设备停机时间

7.增加设备使用效率

# 

# 第二章 总体设计

## 1.1 可行性分析

### 1.1.1 技术可行性

由于EAM系统的开发已有一定的时期，而且有很多成功的实例，技术基础也已经非常雄厚，因而技术上的准备应该不成问题。

EAM系统需要有严谨的规划，注重每一个细小的环节，这样才能使得在用户使用时避免不必要错误发生。我们将使用HTML、JSP等技术来编辑网页，并运用JDBC技术把数据库和动态网页相关联。

本系统采用springMVC+ hibernate 的组合来发这个网站，这两门技术已经在市场存在很长时间，是时间考验后得到认可的技术。同时这两门技术意识我使用比较熟练的技术，在论文这种时间相对短期的情况，使用此两种技术是一个非常合适的场景，即可减少开发时间，又是代码有一定的扩展性，在以后开发更多的功能，提供一个基本的架构。

### 1.1.2经济可行性

本产品的技术上相对比较简单，所需开发周期较短，几乎一两个人就可以完成。因为是论文设计，此产品是由主体的台账，设备管理，功能，权限管理，工单管理为核心，辅助知识库，保养管理，报表等业务。可以采用迭代开发的模式，先开发主体功能，在用户使用后再开发，额外的功能。美工方面，才用easyui。节省了很大的美工人员成本。所以经济成本不大。后期功能扩展多了以后，就可进行集群，负载均衡，分库分表等。成本是一个递增的过程，正符合一个企业的发展。

### 1.1.3社会可行性

EAM系统，对于一些以设备为核心的企业来说，是必须不可缺少的，它所能带来的管理优化，经济效益，是隐形而又不可忽视。也是许多成熟企业所必须的。如国家电网，中国水利等部门都在使用。而博林特这种电梯为主体业务的公司，也正需要此产品。

## 1.2架构设计

### 1.2.1 框架的选择

本系统选择现在比较流行的spring+springMVC+hibernate的框架

1 spring

Spring是一个开源框架，Spring是于2003 年兴起的一个轻量级的Java 开发框架，由[Rod Johnson](http://baike.baidu.com/view/2192255.htm" \t "http://baike.baidu.com/_blank) 在其著作Expert One-On-One J2EE Development and Design中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许使用者选择使用哪一个组件，同时为 [J2EE](http://baike.baidu.com/view/1507.htm" \t "http://baike.baidu.com/_blank) 应用程序开发提供集成的框架。Spring使用基本的[JavaBean](http://baike.baidu.com/view/28155.htm" \t "http://baike.baidu.com/_blank)来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于[服务器](http://baike.baidu.com/view/899.htm" \t "http://baike.baidu.com/_blank)端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。Spring的核心是[控制反转](http://baike.baidu.com/view/1486379.htm" \t "http://baike.baidu.com/_blank)（[IoC](http://baike.baidu.com/view/146665.htm" \t "http://baike.baidu.com/_blank)）和面向切面（[AOP](http://baike.baidu.com/view/73626.htm" \t "http://baike.baidu.com/_blank)）。简单来说，Spring是一个分层的JavaSE/EEfull-stack(一站式) [轻量级](http://baike.baidu.com/subview/1318763/16205192.htm" \t "http://baike.baidu.com/_blank)开源框架。

轻量——从大小与开销两方面而言Spring都是轻量的。完整的Spring框架可以在一个大小只有1MB多的JAR文件里发布。并且Spring所需的处理开销也是微不足道的。此外，Spring是非侵入式的：典型地，Spring应用中的对象不依赖于Spring的特定类。

[控制反转](http://baike.baidu.com/view/1486379.htm" \t "http://baike.baidu.com/_blank)——Spring通过一种称作控制反转（[IoC](http://baike.baidu.com/subview/146665/146665.htm" \t "http://baike.baidu.com/_blank)）的技术促进了低耦合。当应用了IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。你可以认为IoC与JNDI相反——不是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它。

面向切面——Spring提供了[面向切面编程](http://baike.baidu.com/view/1865230.htm" \t "http://baike.baidu.com/_blank)的丰富支持，允许通过分离应用的业务逻辑与系统级服务（例如审计（auditing）和[事务](http://baike.baidu.com/view/121511.htm" \t "http://baike.baidu.com/_blank)（[transaction](http://baike.baidu.com/view/2368136.htm" \t "http://baike.baidu.com/_blank)）管理）进行[内聚性](http://baike.baidu.com/view/2424328.htm" \t "http://baike.baidu.com/_blank)的开发。[应用对象](http://baike.baidu.com/view/5314991.htm" \t "http://baike.baidu.com/_blank)只实现它们应该做的——完成业务逻辑——仅此而已。它们并不负责（甚至是意识）其它的系统级关注点，例如日志或事务支持。

容器——Spring包含并管理应用对象的配置和生命周期，在这个意义上它是一种容器，你可以配置你的每个bean如何被创建——基于一个可配置原型（[prototype](http://baike.baidu.com/subview/1217697/14146221.htm" \t "http://baike.baidu.com/_blank)），你的bean可以创建一个单独的实例或者每次需要时都生成一个新的实例——以及它们是如何相互关联的。然而，Spring不应该被混同于传统的重量级的EJB容器，它们经常是庞大与笨重的，难以使用。

框架——Spring可以将简单的[组件](http://baike.baidu.com/view/379950.htm" \t "http://baike.baidu.com/_blank)配置、组合成为复杂的应用。在Spring中，[应用对象](http://baike.baidu.com/view/5314991.htm" \t "http://baike.baidu.com/_blank)被声明式地组合，典型地是在一个XML文件里。Spring也提供了很多基础功能（[事务管理](http://baike.baidu.com/view/3871410.htm" \t "http://baike.baidu.com/_blank)、持久化框架集成等等），将应用逻辑的开发留给了你。

MVC——Spring的作用是整合，但不仅仅限于整合，Spring 框架可以被看做是一个企业解决方案级别的框架。客户端发送请求，服务器控制器（由DispatcherServlet实现的)完成请求的转发，控制器调用一个用于映射的类HandlerMapping，该类用于将请求映射到对应的处理器来处理请求。HandlerMapping 将请求映射到对应的处理器Controller（相当于Action）在Spring 当中如果写一些处理器组件，一般实现Controller 接口，在Controller 中就可以调用一些Service 或DAO 来进行数据操作 ModelAndView 用于存放从DAO 中取出的数据，还可以存放响应视图的一些数据。 如果想将处理结果返回给用户，那么在Spring 框架中还提供一个视图组件ViewResolver，该组件根据Controller 返回的标示，找到对应的视图，将响应response 返回给用户。

所有Spring的这些特征使你能够编写更干净、更可管理、并且更易于测试的代码。它们也为Spring中的各种模块提供了基础支持。

2.springMVC

Spring MVC属于SpringFrameWork的后续产品，已经融合在Spring Web Flow里面。Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。使用 Spring 可插入的 [MVC](http://baike.baidu.com/view/31.htm" \t "http://baike.baidu.com/_blank) 架构，从而在使用Spring进行WEB开发时，可以选择使用Spring的SpringMVC框架或集成其他MVC开发框架，如Struts1，Struts2等。

springMVC的代替品有很多，我们选择他主要是有一下的优点。Lifecycle for overriding binding, validation, etc，易于同其它View框架（Tiles等）无缝集成，采用[IOC](http://baike.baidu.com/subview/146665/146665.htm" \t "http://baike.baidu.com/_blank)便于测试。

它是一个典型的教科书式的mvc构架，而不像[struts](http://baike.baidu.com/view/25603.htm" \t "http://baike.baidu.com/_blank)等都是变种或者不是完全基于mvc系统的框架，对于初学者或者想了解mvc的人来说我觉得 spring是最好的，它的实现就是教科书！第二它和[tapestry](http://baike.baidu.com/view/66879.htm" \t "http://baike.baidu.com/_blank)一样是一个纯正的servlet系统，这也是它和tapestry相比 struts所没有的优势。而且框架本身有代码，看起来容易理解。

2hibernate分析

hibernate 简介：hibernate是一个开源框架，它是对象关联关系映射的框架，它对JDBC做了轻量级的封装，而我们java程序员可以使用面向对象的思想来操纵数据库。  
 hibernate核心接口:session：负责被持久化对象CRUD操作。sessionFactory:负责初始化hibernate，创建session对象。configuration:负责配置并启动hibernate，创建SessionFactory  
Transaction:负责事物相关的操作。Query和Criteria接口：负责执行各种数据库查询。  
 hibernate工作原理：

（1）通过Configuration config = new Configuration().configure();//读取并解析hibernate.cfg.xml配置文件  
 （2）由hibernate.cfg.xml中的<mapping resource="com/xx/User.hbm.xml"/>读取并解析映射信息  
 （3）通过SessionFactory sf = config.buildSessionFactory();//创建SessionFactory  
 （4）Session session = sf.openSession();//打开Sesssion  
 （5）Transaction tx = session.beginTransaction();//创建并启动事务Transation  
 （6）persistent operate操作数据，持久化操作  
 （7）tx.commit();//提交事务  
 （8）关闭Session  
 （9）关闭SesstionFactory

为什么要用hibernate：  
 （1）对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。  
 （2） Hibernate是一个基于JDBC的主流持久化框架，是一个优秀的ORM实现。他很大程度的简化DAO层的编码工作  
 （3）hibernate使用Java反射机制，而不是字节码增强程序来实现透明性。  
 （4）hibernate的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持各种关系数据库，从一对一到多对多的各种复杂关系。

Hibernate中怎样实现类之间的关系。类与类之间的关系主要体现在表与表之间的关系进行操作，它们都是对对象进行操作，我们程序中把所有的表与类都映射在一起，它们通过配置文件中的many-to-one、one-to-many、many-to-many、

Hibernate缓存的作用。Hibernate是一个持久层框架，经常访问物理数据库，为了降低应用程序对物理数据源访问的频次，从而提高应用程序的运行性能。缓存内的数据是对物理数据源中的数据的复制，应用程序在运行时从缓存读写数据，在特定的时刻或事件会同步缓存和物理数据源的数据

Hibernate缓存分类。Hibernate缓存包括两大类：Hibernate一级缓存和Hibernate二级缓存，Hibernate一级缓存又称为“Session的缓存”，它是内置的，意思就是说，只要你使用hibernate就必须使用session缓存。由于Session对象的生命周期通常对应一个数据库事务或者一个应用事务，因此它的缓存是事务范围的缓存。在第一级缓存中，持久化类的每个实例都具有唯一的OID。 Hibernate二级缓存又称为“SessionFactory的缓存”，由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。第二级缓存是可选的，是一个可配置的插件，在默认情况下，SessionFactory不会启用这个插件。

hibernate的开发步骤：搭建好环境、引入hibernate最小的jar包准备Hibernate.cfg.xml启动配置文件、写实体类(pojo)、为实体类写映射文件"User.hbm.xml"在hibernate.cfg.xml添加映射的实体、创建库表、写测试类、获得Configuration、创建SessionFactory、打开Session

开启事务、 使用session操作数据、提交事务、关闭资源。

### 2.2.2自我框架封装

在本系统中，我们为了更好的进行以后的扩展，我进行了对spring+springMVC和hibernate进行了进一步的封装，下面我对自己封装的代码进行一步解释与说明，以便读者能够更好的理解这个系统。详细代码见附录。

由于hibernate是采用一种面向对象思想完成的一种全自动的orm框架，自然已经封装好了我们常用的增加，修改，删除和查询功能。但是每次我们都需要调用hibernate的session。这样对性能和使用上都是一种损失。根据这点我们包装了一个baseDao 其中最终要的一个方法就是getSession

public Session getSession (){

return this.sessionFactory.getCurrentSession();

}

封装了getSession后，我们有发现我们对每个实体进行保存的时候，我们都需要获得他的实体对象。这时候我们想到我们可以利用反射，在创建Dao对象的时候，你传递进来一个Class对象，我们用反射进行封装与重组，在返回给你。这就好像模板模式。都是印刷dao的工具，根据不同材料生产出不同质地的东西。这时候我创建了public abstract Class<T> getEntityClass ();的这个抽象方法，继承者，重写此方法。

完成了上两步，几乎主题已经搭建出来。根据我的开发经验，我发现一些我们常用的功能。根据id查询一个对象、分页查询对象、分页查询根据条件、分页查询根据多个条件、删除根据条件、删除根据多个条件、更新根据条件、查询根据条件、更新一个实体、根据id进行删除。这些抽象成了一个借口，在附录中有良好的注解。

# 

# 第三章UI设计

## 3.1社会ui分析

首先分析一下本系统对ui的要求：

前后端分离，通过类似restFul接口访问后端服务   
 2、前端布局及样式可灵活配置，可更换皮肤   
 3、有常用组件如form组件,tab,tree,grid等   
 4、浏览器兼容性好，支持IE9、Chrome等主流浏览器（IE8能支持更好，可怜的XP）

社会上主流的ui有easyui，Bootstrap，extjs

下面来分析一下：

Bootstrap，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言[Less](http://baike.baidu.com/view/1229034.htm" \t "http://baike.baidu.com/_blank)写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC（微软全国广播公司）的Breaking News都使用了该项目。国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。

Foundation 在国内市场上并不多见，但是他的特点很是突出。他是一种自力更生的框架，他更像自己把自己代码进行封装复用。这样的优点很明显，就是让每个网站都有自己的特色，与其他人不同。缺点也很明显，就是前期开发很慢，以后的开发速度还可以。同时它没有更多的插件可以使用，就像我们喜爱的jquery 那么多的插件，使用起来非常爽。

Semantic 是一个用来帮助创建漂亮、响应化、人性化的开发框架。这是他的标语。我十分想写在我的论文中。它的优点最大在于人性开发。这个框架看起来代码十分好理解，可读性非常好。对于学习来说，也是十分的简单。他摒弃了那种高深的各种嵌套、内部类的设计。就是让你简单的学会，没有那么多的语法规则，就像list语言一样。七个语法创建世界。

Ext JS是构建互动的跨平台的Web应用程序使用的技术，如Ajax纯JavaScript应用程序框架、DHTML和DOM脚本。原本作为一个附加的库扩展 YUI的杰克61 2007年4月15日， Ext JS包括jQuery原型的互操作性。从1.1版本开始，Ext JS保留不依赖于外部，而使其使用可选的。

以上可以看出他们的设计者都是有一定水平和高度的，都是web前端的专家，但是从使用难以程度，学习成本上，extjs较难，所以放弃了。bootstrap主要是为响应式而生，在我们所有是的web管理系统上没有太大的优势。如果以后做手机端可能会采用，下面我们再来说说我们选用的easyui框架。

## 3.2Easyui的选择

社会上流行的ui框架有很多，很多框架也是有很多的使用人群，维护等等都很好，为什么选择easyui。

因为他有以下的优点：

easyui是一种基于jQuery的用户界面插件集合。

easyui为创建现代化，互动，JavaScript应用程序，提供必要的功能。

使用easyui你不需要写很多代码，你只需要通过编写一些简单HTML标记，就可以定义用户界面。

easyui是个完美支持HTML5网页的完整框架。

easyui节省您网页开发的时间和规模。

easyui很简单但功能强大的。

下面在介绍两种常用的方式，你在本系统中会经常的看到

jQuery EasyUI 提供易于使用的组件，它使 Web 开发人员能快速地在流行的 jQuery 核心和 HTML5 上建立程序页面。 这些功能使您的应用适合今天的网络。 有两个方法声明的 UI 组件:

1. 直接在 HTML 声明组件。

<div class="easyui-dialog" style="width:400px;height:200px"

data-options="title:'My Dialog',collapsible:true,iconCls:'icon-ok',onOpen:function(){}">

dialog content.

</div>

2. 编写 JavaScript 代码来创建组件。

<input id="cc" style="width:200px" />

$('#cc').combobox({

url: ...,

required: true,

valueField: 'id',

textField: 'text'

});

# 

# 第四章 数据库设计

## 4.1数据库设计范式

设计关系[数据库](http://baike.baidu.com/view/1088.htm" \t "http://baike.baidu.com/_blank)时，遵从不同的规范[要求](http://baike.baidu.com/view/198817.htm" \t "http://baike.baidu.com/_blank)，设计出合理的关系型数据库，这些不同的规范要求被称为不同的范式，各种范式呈递次规范，越高的范式数据库冗余越小。

目前[关系数据库](http://baike.baidu.com/view/68348.htm" \t "http://baike.baidu.com/_blank)有六种范式：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、巴斯-科德范式（BCNF）、[第四范式](http://baike.baidu.com/view/176748.htm" \t "http://baike.baidu.com/_blank)(4NF）和[第五范式](http://baike.baidu.com/view/7170192.htm" \t "http://baike.baidu.com/_blank)（5NF，还又称完美范式）。

参考文献中的三大范式：

第一范式（1NF）

所谓第一范式（1NF）是指在[关系模型](http://baike.baidu.com/view/176484.htm" \t "http://baike.baidu.com/_blank)中，对域添加的一个规范要求，所有的域都应该是原子性的，即数据库表的每一列都是不可分割的原子数据项，而不能是集合，数组，记录等非原子数据项。即实体中的某个属性有多个值时，必须拆分为不同的属性。在符合第一范式（1NF）表中的每个域值只能是实体的一个属性或一个属性的一部分。简而言之，第一范式就是无重复的域。

说明：在任何一个[关系数据库](http://baike.baidu.com/view/68348.htm" \t "http://baike.baidu.com/_blank)中，第一范式（1NF）是对[关系模式](http://baike.baidu.com/view/68347.htm" \t "http://baike.baidu.com/_blank)的设计基本要求，一般设计中都必须满足第一范式（1NF）。不过有些关系模型中突破了1NF的限制，这种称为非1NF的关系模型。换句话说，是否必须满足1NF的最低要求，主要依赖于所使用的[关系模型](http://baike.baidu.com/view/176484.htm" \t "http://baike.baidu.com/_blank)。

第二范式（2NF）

在1NF的基础上，非码属性必须完全依赖于码[在1NF基础上消除非主属性对主码的部分函数依赖]

第二范式（2NF）是在第一范式（1NF）的基础上建立起来的，即满足第二范式（2NF）必须先满足第一范式（1NF）。第二范式（2NF）要求数据库表中的每个实例或记录必须可以被唯一地区分。选取一个能区分每个实体的属性或属性组，作为实体的唯一标识。例如在员工表中的身份证号码即可实现每个一员工的区分，该身份证号码即为候选键，任何一个候选键都可以被选作主键。在找不到候选键时，可额外增加属性以实现区分，如果在员工关系中，没有对其身份证号进行存储，而姓名可能会在数据库运行的某个时间重复，无法区分出实体时，设计辟如ID等不重复的编号以实现区分，被添加的编号或ID选作主键。（该主键的添加是在ER设计时添加，不是建库时随意添加）

第二范式（2NF）要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列，以存储各个实例的唯一标识。简而言之，第二范式就是在第一范式的基础上属性完全依赖于主键。

下面就社会实践和数据库设计上讲一下我理解的三大范式

第一范式:确保每列的原子性.  
 如果每列(或者每个属性)都是不可再分的最小数据单元(也称为最小的原子单元),则满足第一范式.  
 例如:顾客表(姓名、编号、地址、……)其中"地址"列还可以细分为国家、省、市、区等。  
 第二范式:在第一范式的基础上更进一层,目标是确保表中的每列都和主键相关.  
 如果一个关系满足第一范式,并且除了主键以外的其它列,都依赖于该主键,则满足第二范式.  
 例如:订单表([订单编号](https://www.baidu.com/s?wd=%E8%AE%A2%E5%8D%95%E7%BC%96%E5%8F%B7&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4mW6kmh7bnj-BmyN-mHKb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErjbvPWT4rHD" \t "http://zhidao.baidu.com/_blank)、产品编号、定购日期、价格、……)，"[订单编号](https://www.baidu.com/s?wd=%E8%AE%A2%E5%8D%95%E7%BC%96%E5%8F%B7&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4mW6kmh7bnj-BmyN-mHKb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErjbvPWT4rHD" \t "http://zhidao.baidu.com/_blank)"为主键，"产品编号"和主键列没有直接的关系，即"产品编号"列不依赖于主键列，应删除该列。  
 第三范式:在第二范式的基础上更进一层,目标是确保每列都和主键列直接相关,而不是间接相关.  
 如果一个关系满足第二范式,并且除了主键以外的其它列都不依赖于主键列,则满足第三范式.  
 为了理解第三范式，需要根据Armstrong公里之一定义传递依赖。假设A、B和C是关系R的三个属性，如果A-〉B且B-〉C，则从这些函数依赖中，可以得出A-〉C，如上所述，依赖A-〉C是传递依赖。

根据自我所学的知识，和市场上的行情，我们大多数程序员都采用第三范式（3NF）下面对第三范式进行详细解说一下

在1NF基础上，任何非主[属性](http://baike.baidu.com/view/77730.htm" \t "http://baike.baidu.com/_blank)不依赖于其它非主属性[在2NF基础上消除传递依赖]。第三范式（3NF）是第二范式（2NF）的一个子集，即满足第三范式（3NF）必须满足第二范式（2NF）。简而言之，第三范式（3NF）要求一个关系中不包含已在其它关系已包含的非主关键字信息。例如，存在一个部门信息表，其中每个部门有部门编号（dept\_id）、部门名称、部门简介等信息。那么在员工信息表中列出部门编号后就不能再将部门名称、部门简介等与部门有关的信息再加入员工信息表中。如果不存在部门信息表，则根据第三范式（3NF）也应该构建它，否则就会有大量的数据冗余。简而言之，第三范式就是属性不依赖于其它非主属性，也就是在满足2NF的基础上，任何非主属性不得传递依赖于主属性。

## 4.2Er模型图



图4-2ER图

## 4.3数据库表

表1维修记录 t\_maintainReword

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | id | int |  | auto |  |
| 记录编号 | num | varchar2 | 30 |  |  |
| 设备名称 | mname | varchar2 | 50 |  |  |
| 设备编号 | dnum | varcahr2 | 40 |  |  |
| 规格型号 | unit | varchar2 | 20 |  |  |
| 使用部门 | dname | varchar2 | 50 |  |  |
| 维修级别 | mlevel | varchar2 |  |  |  |
| 维修人员 | man | varchar2 |  |  |  |
| 计划维修时间 | dotime | date |  |  |  |
| 维修项目 | project | varcahr2 | 50 |  |  |
| 维修标准 | standard | varhcar2 | 100 |  |  |
| 工作描述 | info | varchar2 | 200 |  |  |
| 设备id | eid | int |  |  |  |
| 开始时间 | starttime | datetime |  |  |  |
| 完成时间 | endtime | datetime |  |  |  |
| 维修备注 | remark | varchar2 |  |  |  |

表2保养计划t\_maintainPlan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | id | int |  | auto |  |
| 设备名称 | mname | varchar2 | 50 |  |  |
| 设备编号 | num | varcahr2 | 40 |  |  |
| 规格型号 | unit | varchar2 | 20 |  |  |
| 使用部门 | dname | varchar2 | 50 |  |  |
| 维修级别 | mlevel | varchar2 |  |  |  |
| 循环方式 | method | varchar2 |  |  | 有固定的几种 |
| 维修人员 | man | varchar2 |  |  |  |
| 计划维修时间 | dotime | date |  |  |  |
| 状态 | state | varchar2 | 10 |  | 有效，结束 |
| 维修项目 | project | varcahr2 | 50 |  |  |
| 维修标准 | standard | varhcar2 | 100 |  |  |
| 工作描述 | info | varchar2 | 200 |  |  |
| 设备id | eid | int |  |  |  |

表3保养记录t\_maintainReword

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | id | int |  | auto |  |
| 记录编号 | num | varchar2 | 30 |  |  |
| 设备名称 | mname | varchar2 | 50 |  |  |
| 设备编号 | dnum | varcahr2 | 40 |  |  |
| 规格型号 | unit | varchar2 | 20 |  |  |
| 使用部门 | dname | varchar2 | 50 |  |  |
| 维修级别 | mlevel | varchar2 |  |  |  |
| 维修人员 | man | varchar2 |  |  |  |
| 计划维修时间 | dotime | date |  |  |  |
| 维修项目 | project | varcahr2 | 50 |  |  |
| 维修标准 | standard | varhcar2 | 100 |  |  |
| 工作描述 | info | varchar2 | 200 |  |  |
| 设备id | eid | int |  |  |  |
| 开始时间 | starttime | datetime |  |  |  |
| 完成时间 | endtime | datetime |  |  |  |
| 维修备注 | remark | varchar2 |  |  |  |

表4设备表 t\_equipment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | id | int |  | auto |  |
| 设备名称 | name | varcahr2 | 50 |  |  |
| 设备编号 | num | varcahr2 | 40 |  |  |
| 规格型号 | unit | varchar2 | 20 |  |  |
| 设备类别 | category | int |  |  | 可能需要表 |
| 生产厂商 | creater | int |  |  | 可能需要表 |
| 供应商 | supplier | int |  |  | 可能需要表 |
| 设备标示 | flag | char | 1 |  | 重型设备、一般设备、大型设备、待改造设备 |
| 购置时间 | intime | date |  |  |  |
| 购置方式 | inmethod | char | 1 |  |  |
| 资产证明人 | certifier | varchar2 | 50 |  |  |
| 折旧方式 | deprection | char | 1 |  |  |
| 资产原值 | prePrice | double | 10,2 |  |  |
| 净残率 | rate | double | 2,2 |  | 百分数 |
| 检验周期 | checkRound | varchar2 | 8 |  |  |
| 上次检验 | preCheck | date |  |  |  |
| 下次检验 | preCheck | date |  |  |  |
| 上次维修 | preMaintain | date |  |  |  |
| 使用情况 | useStatus | char | 1 |  | 索引（使用情况）启用，在用，出租，停用 |
| 使用部门 | did | int |  |  | 部门表 |
| 部门名称 | dname | varchar2 | 50 |  | 查询索引（考虑中）0办公室；1质检部；2维修部；3生产部 |
| 操作员 | operator | varchar2 | 50 |  | 操作员编号 |
| 安装地点 | place | varchar2 | 100 |  |  |
| 检查情况 | checkStatus | char | 1 |  | 计划是外键，还是什么 |
| 冗余1-10 | ext1 | varchar2 | 200 |  |  |
| 删除标记 | del | char | 1 |  | 0，删除;1,不是删除；默认是1 |
| 设备类型 | cid |  |  |  |  |
| 供应商外键 | sid |  |  |  |  |

表6设备分类表 t\_equipment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | id | int |  | auto |  |
| 说明 | info | varchar2 | 200 |  |  |
| 设备分类编码 | num | varchar2 | 40 |  |  |
| 设备分类名称 | name | varchar2 | 50 |  |  |
| 有效性 | del | char | 1 |  | 0，删除;1,不是删除；默认是1 |
| 父设备分类编码 | sid | int |  |  |  |

表7用户表 t\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | uid | int |  | auto |  |
| 用户编号 | num | varchar2 | 40 |  |  |
| 用户名 | uname | varhcar2 | 50 |  |  |
| 密码 | pwd | varhcar2 |  |  |  |
| 真是姓名 | truename | varchar2 |  |  |  |
| 性别 | sex | varhcar2 |  |  |  |
| 手机号 | phone | varhcar2 |  |  |  |
| 邮箱 | email | varhcar2 |  |  |  |
| 现居地址 | address | varchar2 |  |  |  |
| 人员状态 | flag | char | 1 |  | 在职，等等 |
| 角色 | rid | int |  |  | 外键n-1 |
| 所属部门 | did | int |  |  | 外键n-1 |
| 冗余 | ext1 | varchar2 | 200 |  | 角色名称 |
| 删除标记 | del | char | 1 |  | 0，删除;1,不是删除；默认是1 |
| 冗余 | ext2 | varchar2 | 200 |  | 部门名称 |

表8角色表t\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | uid | int |  | auto |  |
| 用户编号 | num | varchar2 | 40 |  |  |
| 用户名 | uname | varhcar2 | 50 |  |  |
| 密码 | pwd | varhcar2 |  |  |  |
| 真是姓名 | truename | varchar2 |  |  |  |
| 性别 | sex | varhcar2 |  |  |  |
| 手机号 | phone | varhcar2 |  |  |  |
| 邮箱 | email | varhcar2 |  |  |  |
| 现居地址 | address | varchar2 |  |  |  |
| 人员状态 | flag | char | 1 |  | 在职，等等 |
| 角色 | rid | int |  |  | 外键n-1 |
| 所属部门 | did | int |  |  | 外键n-1 |
| 冗余 | ext1 | varchar2 | 200 |  | 角色名称 |
| 删除标记 | del | char | 1 |  | 0，删除;1,不是删除；默认是1 |
| 冗余 | ext2 | varchar2 | 200 |  | 部门名称 |

表9部门表 t\_department

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | id | int |  | auto |  |
| 部门编号 | num | varhcar2 |  |  |  |
| 部门名称 | name | varhcar2 | 50 |  |  |
| 部门父类 | fid | int |  |  | 本表树父id |
| 简称 | jc | varcahr2 |  |  |  |
| 主负责人 | man | varchar2 | 50 |  |  |
| 电话 | phone | varchar2 | 11 |  |  |
| 传真 | email | varchar2 | 22 |  |  |
| 说明 | info | varchar2 | 200 |  |  |
| 冗余1-10 | ext1 | varchar2 | 200 |  |  |
| 删除标记 | del | char | 1 |  | 0，删除;1,不是删除；默认是1 |
| 级别 | level | char | 2 |  |  |

表10功能表 t\_function

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | id | int |  | auto |  |
| 名子 | name | varchar2 |  |  |  |
| 连接 | href | varchar | 255 |  |  |
| 父id | fid | int | 11 |  |  |
| 冗余1-10 | ext1 | varchar2 | 255 |  |  |

表11生产商表 t\_manufacture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| manufactureid | id | Integer |  | 主 | 非 |
| address | 地址 | Varchar2 | 100 |  |  |
| contact | 联系人 | Varchar2 | 10 |  |  |
| telphone | 联系电话 | Varchar2 | 20 |  |  |
| remark | 备注 | Varchar2 | 100 |  |  |

表12供应商t\_supplier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| supplierid | 供应商ID | Integer |  | 主 | 非 |
| name | 供应商名称 | vahcar2 | 50 |  |  |
| address | 地址 | Varchar2 | 100 |  |  |
| contact | 联系人 | Varchar2 | 10 |  |  |
| telphone | 联系电话 | Varchar2 | 20 |  |  |
| remark | 备注 | Varchar2 | 100 |  |  |

表13工单记录 t\_maintainOrder

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 类型 | 长度 | 约束 | 注解 |
| 主键 | id | int |  |  |  |
| 工单编号 | num | varchar2 | 30 |  |  |
| 设备名称 | mname | varchar2 | 50 |  |  |
| 设备编号 | dnum | varcahr2 | 40 |  |  |
| 规格型号 | unit | varchar2 | 20 |  |  |
| 使用部门 | dname | varchar2 | 50 |  |  |
| 维修级别 | mlevel | varchar2 |  |  |  |
| 维修人员 | man | varchar2 |  |  |  |
| 计划维修时间 | dotime | date |  |  |  |
| 维修项目 | project | varcahr2 | 50 |  |  |
| 维修标准 | standard | varhcar2 | 100 |  |  |
| 工作描述 | info | varchar2 | 200 |  |  |
| 设备id | eid | int |  |  |  |
| 工单状态 | flag | varchar2 |  |  |  |

# 

# 第五章 功能设计

## 5.1功能总述

设备管理系统是以企业设备资产为主要对象，从设备资产采购、安装调试、运行维护到报废的全生命周期进行管理的应用信息系统，以实现企业设备利用最大化，提高效益及投资回报率为总体目标。

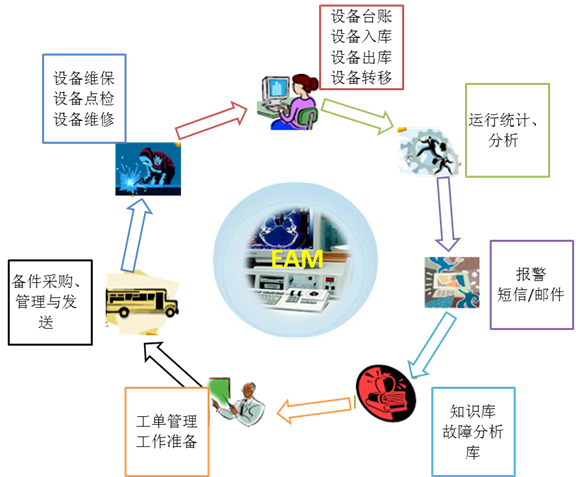


图5.1EAM图

本系统的主要功能有：主数据管理、权限管理、设备台账管理、报表管理、统计分析、维修管理、保养管理、知识库管理。

下面附上一张功能图：

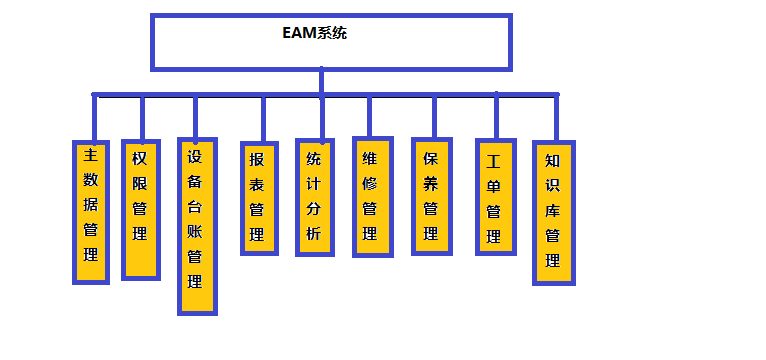


图5.1.1结构图

## 5.2核心功能讲解

本系统主要是一个管理设备的EAM系统，由于电梯设备的特殊性，我们通常是会有电梯监控这个功能，但是这个功能由于需要购买硬件设备，与硬件设备对接，暂时没有开发，属于资金问题，在实际的生产中，电梯的厂商会提供这种设备，就不存在这种问题了。

虽然没有做我也参考了大量的资料学习，这里讲解一下。在硬件提供设备接口的时候，很多硬件都是已经继承好了的。通常硬件会提供两种接口一种，直接返回高低电平，你可以通过c语言。另一种，提供了prompt的接口，这类接口是一种标准，可以与java对接（比较少）。就以主流的c为例，虽然是c语言，我们可以用c类库中的，socket建立一个客户端，遵循http协议链接，java的server端。这样我们就实现了跨语言的沟通。

Server端在进行数据解析和数据持久化， 其他用户登录后可以用get请求的到相应的数据。进行查询观察。

# 

# 第六章 各模块详细设计

## 6.1 权限管理

权限管理主要分为3部分。

第一部分是功能，每个系统都有很多的功能，我们把功能抽象一个类的概念，他包括功能的名称，功能的描述，功能的访问地址，功能的父级，这些东西构建成功能的类，同时也是一个功能树。这个功能树也对应着一个菜单树。

第二部分是角色，角色是每个系统中不可或缺的一部分，最基本的系统也是包含管理员和普通用户两个角色，此系统是一个多角色的系统，我们默认给与管理员和员工两个角色，管理员具有所有的功能。普通用户只有查询的功能。企业还可以使用管理进行添加新的角色， 并为角色进行设置新的功能，让他具有新的意义。

第三部用户，现在都是多用户时代很少出现单用户的系统。多用户，用户权限才是使用者具体感觉到的，因为此系统是3级划分，所以，我们只能为用户设计角色，并不能设计功能。为每一个用户设计角色，这就好像为每个同学分配专业一样，让他们可以学习不同的课程。

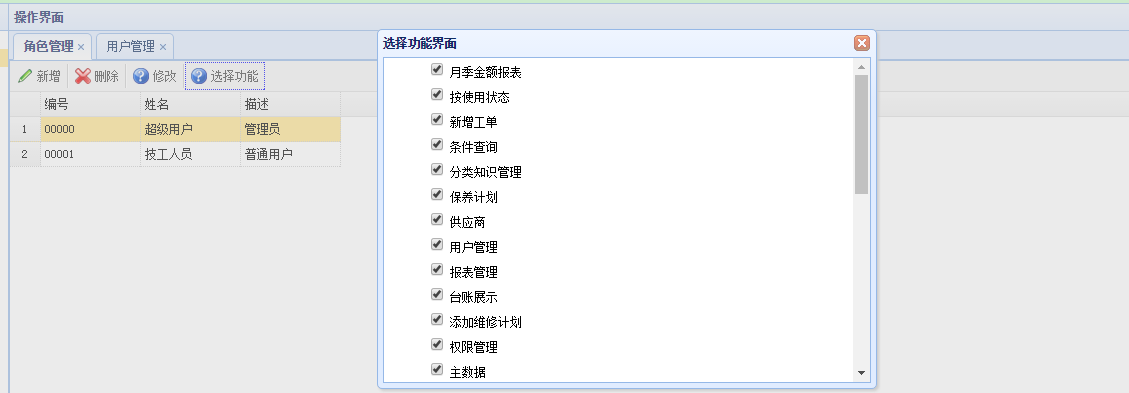


图6.1选择功能示意图

## 6.2设备管理主数据

设备管理的核心是设备主数据的管理，本项目采用国际标准化的设备主数据结构，以便系统有良好的功能可扩展性及二次开发能力，适应企业未来业务需求变更的要求。

设备是独立的维护单元，被维修、测试、检查的技术对象。设备损坏后首先是修理，而不是更换，配件则相反。当需要管理单台设备的技术参数、失效、维护、移动、成本、评估和统计运行时间等信息的时候，就需要在系统内维护设备主数据。设备和功能位置是分配和被分配，也就是安装和拆卸的关系。如在某变电站内，1号主变压器安装在变电站的功能位置上。在设备记录信息里面，可以查看到该设备目前的安装地点，并可查询到历史安装和拆卸记录。

图6.2设备查询

## 6.3设备保养计划管理流程

根据设备运行状态、维修周期、运行状态等自动提醒建立预防性维修保养计划（需人工审核后下达，避免系统自动产生大量实际不执行的工单）、同时支持人工制定维修保养计划。



图6.3新建维修计划

图6.4 流程图

## 6.4维修工单执行流程



适用于设备部维修工执行设备故障诊断、领料出库、工单信息完善、维修记录、系统数据录入等。

业务流程图

图6.5 流程图



## 6.5知识库管理

知识库管理把企业内部的设备维修的方法、总结、经验等等形成企业内部的知识库，最终推动企业内部设备维修制度和过程的逐步完善，达到降低设备维护成本、减少设备停机时间和提高设备运行效率的效果。

现场工作人员是知识体系的载体，其技术水平和素质直接决定了设备维保水平的高低。系统知识库就是把有关设备的技术资料，关键参数、图纸等静态信息以及维修经验、检修路线、维修履历、故障案例、改善提案等动态知识全部进行汇编。通过文字、表格或图片等直观的方式精确地展现。它直接面向现场的技术人员和操作员工，提供直观生动的培训和技术手册，并可以进行交互式工作指导。

知识库通过相关管理人员、现场维修人员的日常工作，不断的增加知识和完善企业知识库体系。本项目通过设备档案、维修记录的维护和多重组合查询，实现设备维保知识的沉淀与共享。

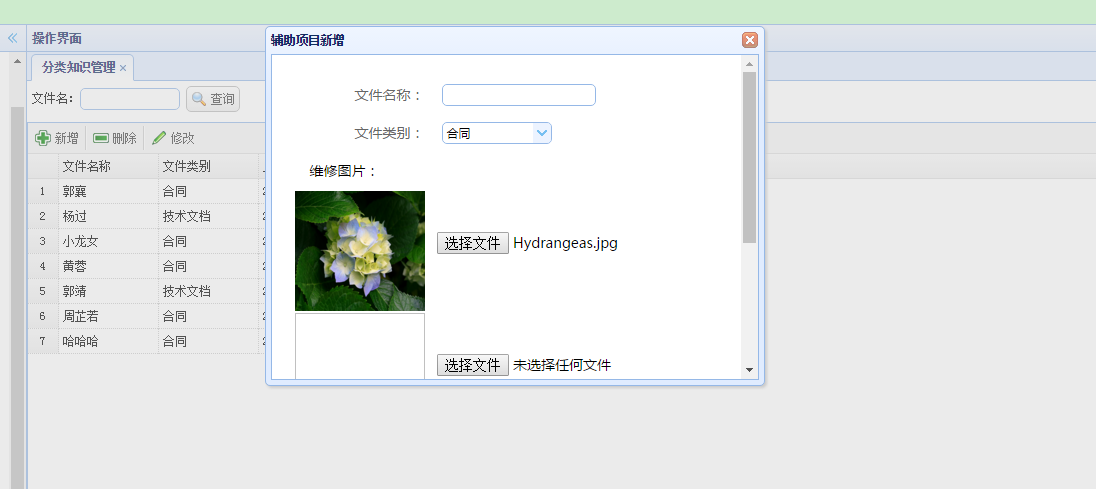


图6.5.2知识库上传

## 6.6统计分析

系统提供符合用户习惯的报表模板，来设计报表的模型

能提供灵活的数据查询

对设备数据可以由用户自定义各种可用条件和模糊方式进行查询，同时还可以把查询到的信息转入到EXCEL等常用格式文件中，让用户进行处理。

能根据各类数据，生成各类图表，协助进行趋势分析。

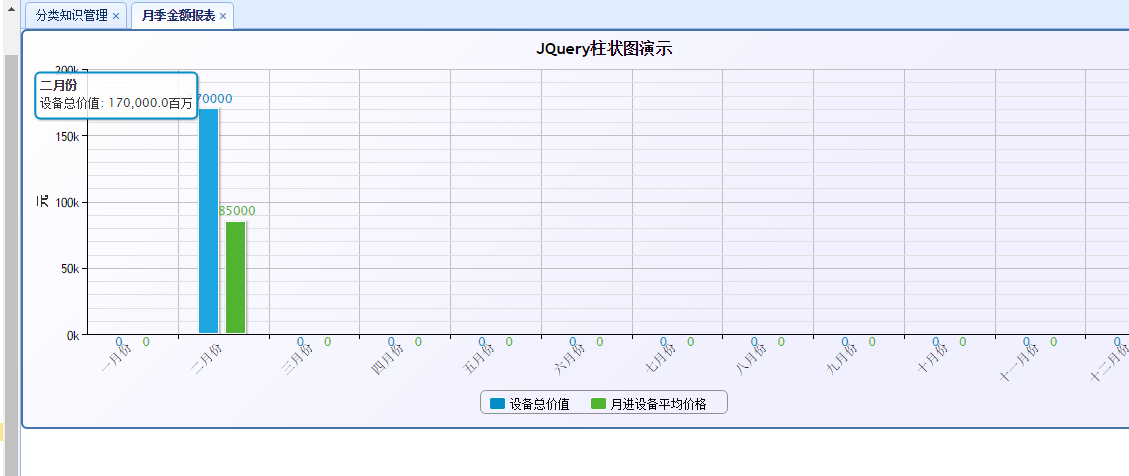


图6.6报表

## 6.7工单管理

工单也是EAM系统的核心功能之一，设备发生了错误，设备需要维修，设备设计了维修计划，知识库的新增。这些等等最终都是落实道工单上面，因为工单才是一个系统的关键点。在维修和保养界面都添加生成工单的功能。每天维修保养得计划人员，定好计划以后，点击指定工单，系统会自动判断当天的日期，进行工单的生成。

工单的执行人员也就是用户，登录后可以根据个人信息，在工单的查询界面，看到自己今天的工单，以后还会连接打印机，或者推出手机端的系统，让维修员工，根据工单执行自己的维修内容。



图6.7工单

## 6.8设备台账管理

台账原是指摆放在台上供人翻阅的账簿，故名台账。久而久之，这个名词就固定下来，实际上就是[流水账](http://baike.baidu.com/view/827244.htm" \t "http://baike.baidu.com/_blank)。设备台账，顾名思义就是设备的流水账，我们在银行、邮局都会有这种流水账，这种流水账只能增加，是不能修改和删除的，他相当于一种日志，记录着设备的在系统内部的生命周期中的每个变动。如果以后发生了任何的纠纷，设备台账将是解决问题最好的依据。



图6.8新增台账

# 

# 第七章 测试

## 7.1搜索功能

若查询条件为输入框，则参考输入框对应类型的[测试](http://www.cnblogs.com/Jessy/admin/javascript.:;)方法

1、功能实现：

（1）如果支持模糊查询，搜索名称中任意一个字符是否能搜索到

（2）比较长的名称是否能查到

（3）输入系统中不存在的与之匹配的条件

（4）用户进行查询操作时，一般情况是不进行查询条件的清空，除非需求特殊说明。

2、组合测试：

（1）不同查询条件之间来回选择，是否出现页面错误（单选框和多选框最容易出错）

（2）测试多个查询条件时，要注意查询条件的组合测试，可能不同组合的测试会报错。

## 7.2添加、修改功能

1、特殊键：（1）是否支持Tab键 （2）是否支持回车键

2、提示信息：（1）不符合要求的地方是否有错误提示

3、唯一性：（1）字段唯一的，是否可以重复添加，添加后是否能修改为已存在的字段（字段包括区分大小写以及在输入的内容前后输入空格，保存后，数据是否真的插入到数据库中，注意保存后数据的正确性）

4、数据 正确性：

（1）对编辑页的每个编辑项进行修改，点击保存，是否可以保存成功，检查想关联的数据是否得到更新。

（2）进行必填项检查（即是否给出提示以及提示后是否依然把数据存到数据库中；是否提示后出现页码错乱等）

（3）是否能够连续添加（针对特殊情况）

（4）在编辑的时候，注意编辑项的长度限制，有时在添加的时候有，在编辑的时候却没有（注意要添加和修改规则是否一致）

（5）对于有图片上传功能的编辑框，若不上传图片，查看编辑页面时是否显示有默认的图片，若上传图片，查看是否显示为上传图片

（6）修改后增加数据后，特别要注意查询页面的数据是否及时更新，特别是在首页时要注意数据的更新。

（7）提交数据时，连续多次点击，查看系统会不会连续增加几条相同的数据或报错。

（8）若结果列表中没有记录或者没选择某条记录，点击修改按钮，系统会抛异常。

## 7.3删除功能

1、特殊键：（1）是否支持Tab键 （2）是否支持回车键

2、提示信息：（1）不选择任何信息，直接点击删除按钮，是否有提示（2）删除某条信息时，应该有确认提示

3、数据 实现：（1）是否能连续删除多个产品（2）当只有一条数据时，是否可以删除成功 （3）删除一条数据后，是否可以添加相同的数据（4）如系统支持批量删除，注意删除的信息是否正确 （5）如有全选，注意是否把所有的数据删除（6）删除数据时，要注意相应查询页面的数据是否及时更新 （7）如删除的数据与其他业务数据关联，要注意其关联性（如删除部门信息时，部门下游员工，则应该给出提示）（8）如果结果列表中没有记录或没有选择任何一条记录，点击删除按钮系统会报错。

如：某一功能模块具有最基本的增删改查功能，则需要进行以下测试

单项功能测试（增加、修改、查询、删除）

增加——>增加——>增加 （连续增加测试）

增加——>删除

增加——>删除——>增加 （新增加的内容与删除内容一致）

增加——>修改——>删除

修改——>修改——>修改 （连续修改测试）

修改——>增加（新增加的内容与修改前内容一致）

修改——>删除

修改——>删除——>增加 （新增加的内容与删除内容一致）

删除——>删除——>删除 （连续删除测试）

## 7.4注册、登陆模块

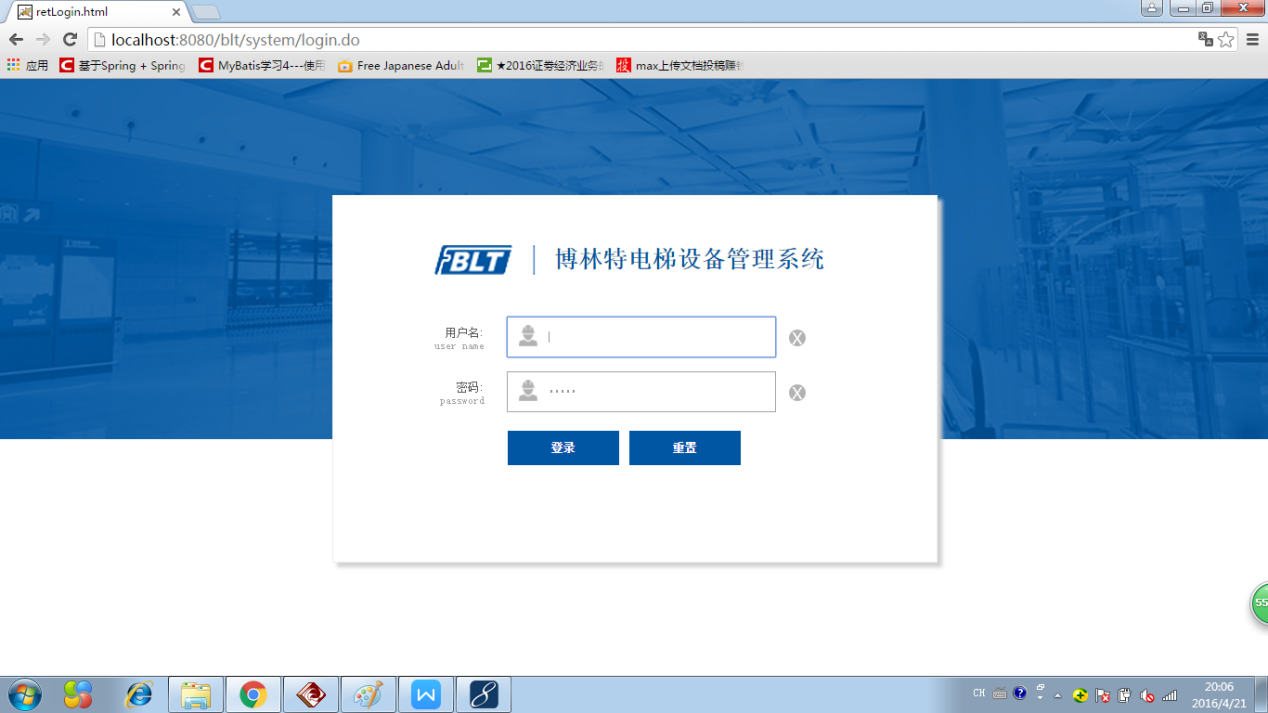


图7.4登录页

### 7.4.1注册功能

（1）注册时，设置密码为特殊版本号，检查登录时是否会报错

（2）注册成功后，页面应该以登陆状态跳转到首页或指定页面

（3）在注册信息中删除已输入的信息，检查是否可以注册成功。

### 7.4.2登陆 功能

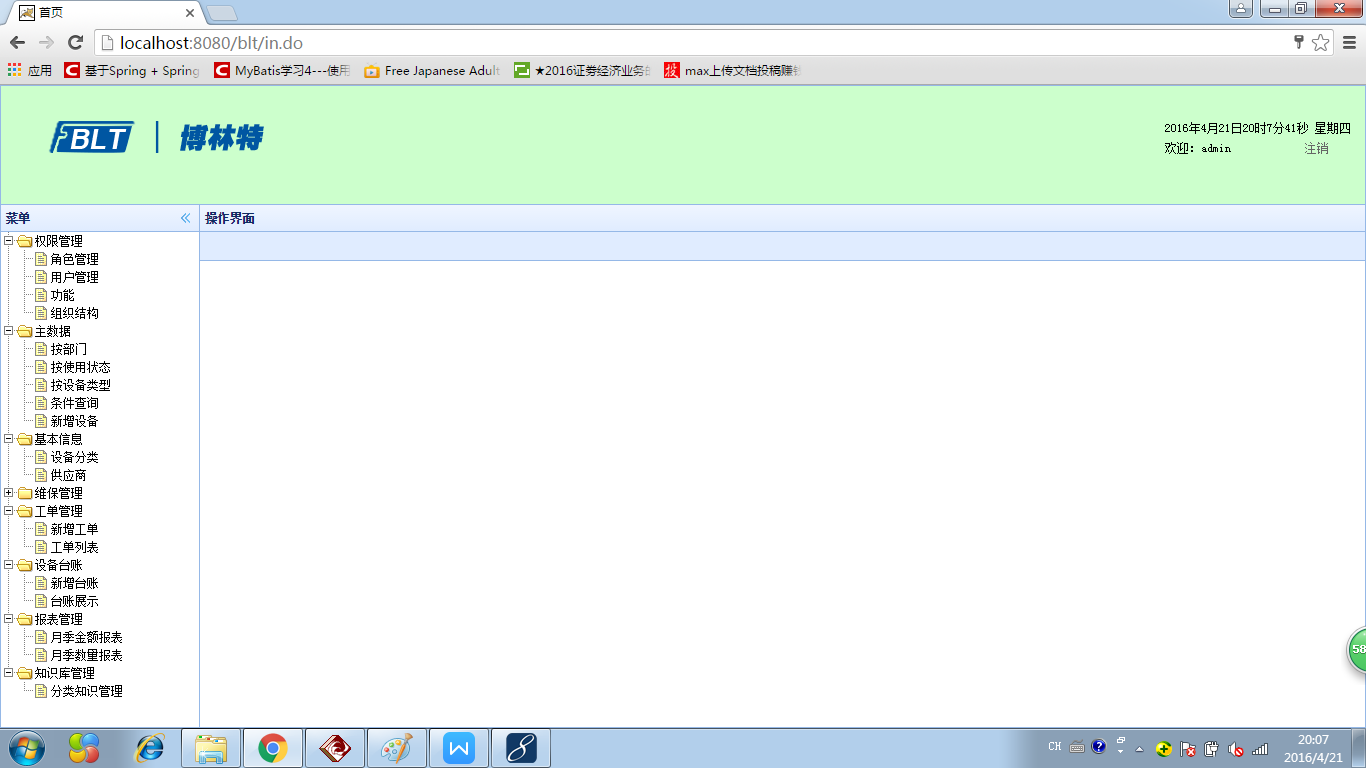


图7.4.2登录成功后后台首页

（1）输入正确的用户名和正确的密码

（2）输入正确的用户名和错误的密码

（3）输入错误的用户名和正确的密码

（4）输入错误的用户名和错误的密码

（5）不输入用户名和密码（均为空格）

（6）只输入用户名，密码为空

（7）用户名为空，只输入密码

（8）输入正确的用户名和密码，但是不区分大小写

（9）用户名和密码包括特殊字符

（10）用户名和密码输入超长值

（11）已删除的用户名和密码

（12）登录时，当页面刷新或重新输入数据时，验证码是否更新

## 7.5 测试小结

对于项目来说， 手动测试和自动化测试同等重要，都是保障软件质量的方法。 目前大部分的项目组都是手动测试和自动化测试相结合。因为很多测试无法做成自动化，很多复杂的业务逻辑也很难自动化， 所以自动化测试无法取代手动测试。

对于软件测试人员个人发展来说， 做自动化测试是个挑战，也是测试人员发展的一个方向，  需要测试人员学习大量的开发知识（开发的知识真是学无止境啊）。 从长远角度来看，自动化测试肯定是越来越吃香的。

而手动测试比较适合刚工作不久的人，手动测试最大的缺点就是技术含量低，单调乏味，容易废人。

总的来说，手工测试胜在测试业务逻辑，而自动化测试胜在测试底层架构。

非功能测试

一个软件除了基本功能之外，还有很多功能之外的特性，这些叫“Quality of Service requirement”服务质量需求。没有软件的功能，这些特性都无从表现出来，因此，我们要在软件开发的适当阶段-基本功能完成后做这些测试。

对于一个项目来说，他的质量就是有测试来保证的，没有一个程序员，可以不出错，没有bug，哪怕一行代码，也可能会出现bug。所以我们java程序员在完成一部分功能后，就要不断的改变输入量，检测输出量，来完成单元测试。在后面我们还有集成后完成集成测试。然后才能上线。好的项目必须加重测试环节。

# 

# 第八章 所遇问题与解决

## 8.1 问题一hibernate不发出更新语句

此次论文设计编写中第一个问题就是没有抛出异常就是不发出sql语句，导致数据库没有进行修改。我查找网站个方面资料。采用如下方法解决了问题。

首先在后台hibernate的配置文件，或者spring中配置如下信息。

<property name="hibernate.show\_sql">true</property>

<property name="hibernate.format\_sql">true</property>

在hibernate中进行配置这条信息的时候，我们就会看到hibernate发出的语句。这是一种常见的调试方法。

关于不发出sql语句我对我的程序进行了一下的猜想。

1.myeclipse和浏览器和tomcat都会存在一些缓存，你修改的代码是没有及时发布的。导致操作失效。

我尝试着将tomcat中localhost下，和webapp的项目都删除。重新发布。重启myeclipse。同时结束所有myeclipse的线程。ctrl+f5刷新浏览器。不过我不是这个问题没有解决。

2.事务没有配置成功。

<tx:advice id="txAdvice" transaction-manager="txManager">

    <tx:attributes>

        <tx:method name="save\*" propagation="REQUIRED"

            rollback-for="java.lang.Exception" />

        <tx:method name="delete\*" propagation="REQUIRED"

            no-rollback-for="java.lang.Exception" />

        <tx:method name="update\*" propagation="REQUIRED"

            rollback-for="java.lang.SQLException" />

        <tx:method name="\*" read-only="true" />

    </tx:attributes>

</tx:advice>    
如果是这么配置的事务。那么你就要注意的你的方法，是不是与配置的事务方法相同。如果你是采用“<!-- 使用注解，在Service层的类上添加@Transcational() -->  <tx:annotation-driven transaction-manager="trManager"   proxy-target-class="true" />  ”

这么配置的事务，你要注意，是否在指定的类上，放上@transcational注解。同时注意你的注解扫描包。也不是这个原因我配置了如下参数。

3.更新的时候，你的数据没有发生变化。

hibernate在更新的时候，我们很多时候采取，根据id进行查询的到持久化对象，对对象进行修改，进行保存。如果你的对象没有发生修改，那么就不会发生update语句。比如你setNum（num++）；这就没有进行修改。我发生的就是这个原因。基础不够扎实。Num++是先赋值后加1.所以数据是没有进行修改的。

## 6.2hibernate查询问题

在论文的编写中我又遇到了我第二个难题。就是查询出一个po对象。然后在集合中判断他是否存在。对象的判断需要重写hashcode 和 equal 方法。这里采用的是eclipse的快捷键重写，选择的是以id 作为依据。写完之后里面有一句代码是这个“if (getClass() != obj.getClass())   return false;  ”就是这句代码产生了错误。

我再次网上查找相关信息。仔细检查我的程序。终于发现了问题。下面记录一下这个问题，这个问题涉及到了hibernate的原理，和java，Object的默认方法重写问题，还是很值得我们进行深入研究的。

hibernate 在设置一对多属性的时候，我们默认在onetomany的时候fetch为lazy ，就是懒加载many端。

所以你在list()查询的时候的得到的不是一个po对象，而是一个po的代理对象。“class com.qykh.core.domain.TFunctionEnhancerByCGLIBce339e51”。“class com.qykh.core.domain.TFunction”

代理对象和实体对象的class类型是这样的一个是代理，一个是原对象不相等。所以indexOf进行匹配的时候会一直出错。

解决办法：自己来写equal 方法，

@Override

    public boolean equals(Object obj) {

        TFunction t = (TFunction)obj;

        if(t.getId()==this.getId()){

            return true;

        }

        return false ;

    }

以上是在编写论文项目中的第二问题的解决办法。

8.3 Hibernate懒加载导致json 问题

这是我在论文中的第三个问题，也是最有深度的一个问题。这个问题的表现，主要是你在点击一个查询页面的时候，页面没有任何数据，后台抛出一个json异常。

这时候很多人都无法解决这个问题，以为你json 使用的是springMVC的responsebody

@RequestMapping("/list.do")

@ResponseBody

public Map<String, Object> List(int page, int rows) {

PageTion data = service.listByPage(page, rows, null);

Map<String, Object> map = new HashMap<String, Object>();

map.put("total", data.getNum());

map.put("rows", data.getList());

return map;

}

这样的代码。你的json是有springMVC框架进行转化的，代码的Exception抛出的位置，也springMVC的源码中。让人感觉有些棘手，难道我要修改springMVC的源码。这是不可能的，首先我们是没办法修改源码的。其次这是社会的最流行的框架，已经经过无数人的测试，不会出现这么大的bug。所以一定是我们的使用发生了问题。

我就开始在网上查询springMVC的资料。然后阅读源码。我发现springMVC是jackson进行的json转化。不是我们原来使用jsonlib。而hibernate是懒加载。我们解析jackson的时候，都要进行遍历。

进行遍历后，会碰到hibernate设计的one-to-many。或者many-to-many。都是这些在设置为懒加载后，都会发出sql进行加载。你发出sql是有延时。For循环遍历较快。这是会发生null，又变成了有数据。Json转化抛出了异常。

经过代码的研究。我找到了一个解决办法。就是在实体中。你不想加载的属性上加上JsonIgnore。具体事例如下:

@OneToMany(fetch = FetchType.LAZY)

@JoinColumn(name = "fid")

@JsonIgnore

public Set<TFunction> getSet() {

return set;

}

## 8.4 本章小结

本章内容虽然不是论文的核心要点，但这是最能展现一个人的编程能力高低第一章。软件我们都知道是一个迭代开发的过程。为什么很多软件开发一阵子之后就不行了。为什么阿里一定要找基础好的程序员。这些全在于bug的修改。开发程序很容易，但是修改就很难。你没有扎实的基础丰富的经验，你很难判断出是哪里出的问题。即便你知道了问题的位置，你不理解java的底层怎么编写的。你也很难修改。此章节让我感受良多，也进步了许多。大学的基础课都是很有用的。不要好高骛远，安心的学好老师所向你传授的知识。脚踏实地才能走的更远。

# 总结

本系统开始设计到完成，其中改了很多次，由于个人的技术水平和社会经验不足还发生过，数据库缺少字段发生大改动。使得自己开发周期加长了不少，在以后的开发中我会设计更多的冗余和更加仔细分析自己的业务，想好了，在开工减少后面的更改量。不过到此还是实现了系统的主要功能，在此次的毕业设计中也使得自己的技术水平有了一定的提升。

其实还有很多的不足。对于一个网站来讲，不仅要拥有强大和先进的功能，更要兼顾其稳定性和容错性。在网站功能实现后，仍然存在很多问题需要经过大量的测试才能发现和改进。测试主要分三个方面进行：网站的易操作性；网站输入的容错性；网站功能的稳定性。一般采用功能测试和验收测试。功能测试是自己根据用户的要求进行测试，测试过程尽量发现并改正错误；验收测试是让客户或其他人员根据自己对网站功能的理解进行测试，从而查找一些小的或者想不到的错误。

由于这是一个原型网站，所以只进行了功能测试，但发现了不少错误，并进行了改正。在功能测试的过程中，一些问题很显而易见，有些错误却比较隐藏。每个小模块做好的时候，先运行，看是否成功，若有错误，则根据网页的错误提示及时进行更正。

通过对网站的测试，由于时间的原因，本网站中还存在以下一些问题与不足之处：用户界面不够友好，一些界面操作不够灵活、方便；由于时间原因，在网站美化上，尚且不够完善；由于这是一个原型网站，在网站安全性方面还不够完善。

# 

# 致 谢

本文从选题、构思、提纲拟订、资料收集、论文写作及修改至最后定稿，历时将近半年，在这段时间内， 常晓娟老师为我的课题倾入了大量的心血，给予了我丰富的建议和悉心的指导。毕业设计临近尾声，我对老师表示衷心的感谢。他是一位非常优秀的年轻老师，知识渊博，严谨负责，做事态度非常认真，他优良的风范深深地影响着我，让我拥有了正确的学习态度和对科学的严谨，顺利地完成了毕业设计。

另外，我要感谢学院使我学到了很多知识，让我顺利完成本科学业。同时也非常感谢所有关心过我、帮助过我的领导、老师和同学们。我会再接再厉，更上一层楼。

最后向所有支持和曾经帮助过我的人表示衷心的感谢！

# 

# 参考文献

[1] 冯燕奎, 赵德奎. JSP实用案例教程[M] 清华大学出版社, 2014, 5: 70-100

[2] 王家华 软件工程[M] 东北大学出版社2011年3月303页

[3] 王宜贵 软件工程[M] 北京：机械工业出版社，2012：20-79

[4] 孙卫琴 精通struts[M]电子工业出版社 2014年8月 50-70

[5] 孙卫琴 精通hibernate[M]电子工业出版社 2015年4月 100-120

[6] 张洪斌 java2高级程序设计[M]中科多媒体出版社 2014年11月 85-90

[7] Jim Arlow UML2.and the Unified Process[M]机械工业出版社 2006年6月 30-40

[8] 林信良 spring2.0技术手册[M]电子工业出版社 2015年6月 50-100

[9] 熊节、林仪明、张凯峰、陈玉泉等主编[《CSDN社区电子杂志 Java杂志》创刊号]

[10]《程序员》杂志 2014 年第4期

[11] 耿祥义编著.JSP基础编程[M].清华大学出版社，2014.55-162

[12]徐建波，周新莲.Web设计原理于编程技术[M].中南大学出版社，2015.185-193

[13]孙鑫编著.Java Web开发详解[M].电子工业出版社，2011.189-274

[14]林上杰，林康司编著.JSP2.0技术手册[M].电子工业出版社，2013.3-6

[15]陈强.精通Java开发技术 .清华大学出版社.2014   
 [16] Buell.Datastructures using Java. Jones&Bartlett Learning.2013   
 [17] Martin NgobyeComputing Static Slice for Java Programs.2014

[18]Kathy.Head FirstJava,2nd Edition.O’Reilly Media.2015  
 [19] Bruce Eckel.Thinkingin Java.Addison-Wesley Professional.2013   
 [20] JoshuaBloch.Effective Java Programming LanguageGuide.Addison-Wesley  2015

# 

**Brilliant elevator system**

**LIU Dong-He**

**Abstract：**Rapid rise of high rise. The elevator has become a part of people life indispensable, every day we are going to be riding the elevator, both passenger and goods ladder, or escalator. Out of danger is not a small thing. Is a threat to our personal safety. Even if there is no big accident, always maintenance will bring inconvenience to our lives. In order to solve the above problems, I start from the perspective of enterprise management. For elevator enterprises to customize the development of EAM equipment management system, so that enterprises better manage the life cycle of the elevator, the timely treatment of elevator accidents, regular maintenance of the elevator. Use the market more commonly used java language, web based B/S application. To bring convenience to people, but also to improve the management ability of the enterprise to the staff. Improve productivity, create more value.

**Key words: elevator, B/S, EAM, Java**

# 附 录

package com.qykh.frame.dao;

import java.io.Serializable;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import org.hibernate.Criteria;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.criterion.Restrictions;

import org.hibernate.transform.Transformers;

import org.springframework.beans.factory.annotation.Autowired;

import com.qykh.frame.page.Order;

import com.qykh.frame.page.PageTion;

import com.qykh.frame.page.Paramter;

public abstract class HibernateBaseDao<T> implements IBaseDao<T> {

@Autowired

private SessionFactory sessionFactory;

public abstract Class<T> getEntityClass ();

public SessionFactory getSessionFactory() {

return sessionFactory;

}

public void setSessionFactory(SessionFactory sessionFactory) {

this.sessionFactory = sessionFactory;

}

public Session getSession (){

return this.sessionFactory.getCurrentSession();

}

/\*\*

\* 查询分页

\* @param pageNo

\* @param pageSize

\* @param par

\* @return

\*/

@SuppressWarnings("unchecked")

public PageTion listByPage(int pageNo,int pageSize ,Paramter par){

//获取session

Session session = getSession();

//得到表名称

Class<? extends Object> class1 = getEntityClass();

String simpleName = class1.getSimpleName();

String hql = "from "+simpleName+" where 1=1 ";

//插入参数集合

Map<String,Object> map = new HashMap<String, Object>();

if(par!=null){

map = par.getMap();

//设置参数

for(Map.Entry<String, Object> en:map.entrySet()){

hql+=" and "+en.getKey()+"=:"+en.getKey();

}

//是否判断日期

if(par.getBtweenAnd()!=null){

hql+=par.getBtweenAnd();

}

//是否排序

List<Order> orders = par.getOrders();

if(orders.size()>0){

hql += " order by ";

for(Order order :orders){

if(order.getFalg()== 1){

hql+=" "+order.getClumn()+" desc,";

}else{

hql+=" "+order.getClumn()+",";

}

}

hql = hql.substring(0, hql.lastIndexOf(","));

}

}

//创建查询

String count = "select count(\*) "+hql;

Query query = session.createQuery(hql);

Query countQuery = session.createQuery(count);

//插叙结果

for(Map.Entry<String, Object> en:map.entrySet()){

query.setParameter(en.getKey(), en.getValue());

countQuery.setParameter(en.getKey(), en.getValue());

}

//分页

query.setFirstResult((pageNo-1)\*pageSize).setMaxResults(pageSize);

//结果总数

long lon = (Long) countQuery.uniqueResult();

int total =(int) lon;

List<T> list = query.list();

return new PageTion(pageNo, pageSize, list, total);

}

@Override

public void delete(T t) {

getSession().delete(t);

}

@SuppressWarnings("unchecked")

@Override

public List<T> list() {

Class<? extends Object> class1 = getEntityClass();

String simpleName = class1.getSimpleName();

String hql = "from "+simpleName ;

List list = getSession().createQuery(hql).list();

return list;

}

@Override

public List<T> queryByCondition(T t) {

return null;

}

@SuppressWarnings("unchecked")

@Override

public List<T> queryByCondition(String key ,Object obj) {

Criteria cri = getSession().createCriteria(getEntityClass());

cri.add(Restrictions.eq(key, obj));

return cri.list();

}

@SuppressWarnings("unchecked")

@Override

public T queryById(Serializable id) {

Class<? extends Object> class1 = getEntityClass();

T t= (T) getSession().get(class1, id);

return t;

}

@Override

public void save(T t) {

getSession().save(t);

}

@Override

public void update(T t) {

getSession().update(t);

}

@SuppressWarnings("unchecked")

public List<Map<String,Object>> queryBySql(String sql){

List<Map<String,Object>> list = getSession().createSQLQuery(sql).setResultTransformer(Transformers.ALIAS\_TO\_ENTITY\_MAP).list();

return list;

}

@SuppressWarnings("unchecked")

@Override

public T queryOneByCondition(String key, Object value) {

T retVal = (T) getSession().createCriteria(getEntityClass()).

add(Restrictions.eq(key, value)).setMaxResults(1).uniqueResult();

return retVal;

}

public void saveOrUpdate(T t) {

Session session = getSession();

session.saveOrUpdate(t);

};

/\*\*

\* 按条件删除

\*/

public void deleteByCondition(String key,Object value){

String hql = "delete from "+getEntityClass().getSimpleName()+" where 1=1 and "+key+"=?";

getSession().createQuery(hql).setParameter(0, value).executeUpdate();

}

/\*\*

\* 安多条件删除

\*/

public void deleteByConditions(Map<String,Object> map){

}

}

package com.qykh.frame.dao;

import java.io.Serializable;

import java.util.List;

import java.util.Map;

import com.qykh.frame.page.PageTion;

import com.qykh.frame.page.Paramter;

public interface IBaseDao<T> {

/\*\*

\* 分页查询

\* @param pageNo

\* @param pageSize

\* @param par

\*/

public PageTion listByPage(int pageNo,int pageSize ,Paramter par);

/\*\*

\* 查询所有

\*/

public List<T> list();

/\*\*

\* 保存

\* @param t

\*/

public void save(T t);

/\*\*

\* 更新

\* @param t

\*/

public void update(T t);

/\*\*

\* 删除

\* @param t

\*/

public void delete(T t);

/\*\*

\* 根据id进行查询

\* @param id

\*/

public T queryById(Serializable id);

/\*\*

\* 按条件查询，如果条件不为null，那么就根据条件查询，但是不包括id字段

\* @param t

\*/

public List<T> queryByCondition(T t);

/\*\*

\* 根据一个条件进行查询

\* @param key

\* @param value

\* @return

\*/

public List<T> queryByCondition(String key ,Object value);

/\*\*

\* 由sql进行查询

\* @param sql

\* @return

\*/

public List<Map<String,Object>> queryBySql(String sql);

/\*\*

\* 根据条件查询出一条

\*/

public T queryOneByCondition(String key,Object value);

/\*\*

\* 保存或更新

\*/

public void saveOrUpdate(T t);

/\*\*

\* 按条件删除

\*/

public void deleteByCondition(String key,Object value);

/\*\*

\* 安多条件删除

\*/

public void deleteByConditions(Map<String,Object> map);

}

package com.qykh.frame.service;

import java.io.Serializable;

import java.util.List;

import com.qykh.frame.page.PageTion;

import com.qykh.frame.page.Paramter;

public interface IBaseService<T> {

/\*\*

\* 分页查询

\* @param pageNo

\* @param pageSize

\* @param par

\*/

public PageTion listByPage(int pageNo,int pageSize ,Paramter par);

/\*\*

\* 查询所有

\*/

public List<T> list();

/\*\*

\* 保存

\* @param t

\*/

public void save(T t);

/\*\*

\* 更新

\* @param t

\*/

public void update(T t);

/\*\*

\* 删除

\* @param t

\*/

public void delete(T t);

/\*\*

\* 根据id进行查询

\* @param id

\*/

public T queryById(Serializable id);

/\*\*

\* 按条件查询，如果条件不为null，那么就根据条件查询，但是不包括id字段

\* @param t

\*/

public List<T> queryByCondition(T t);

}

论文评阅人意见

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **论文（设计）题目** |  | | | | |
| **作 者** |  | | | | |
| **指导教师** |  | **指导教师职称** | | |  |
| **评阅人** |  | **评阅人职称** | | |  |
| **意 见** |  | | | | |
| **评阅人**  **签字** |  | | **评阅意见** |  | |

论文评阅人意见

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **论文（设计）题目** |  | | | | |
| **作 者** |  | | | | |
| **指导教师** |  | **指导教师职称** | | |  |
| **评阅人** |  | **评阅人职称** | | |  |
| **意 见** |  | | | | |
| **评阅人**  **签字** |  | | **评阅意见** |  | |

指导教师评语页

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **论文（设计）题目** |  | | | | |
| **作 者** |  | | | | |
| **指导教师** |  | **职称** | |  | |
| **评 语** |  | | | | |
| **指导教师签字** |  | | **论文等级** | |  |

评语页范例（评语应客观，反映学生实际情况，字数不得少于300字）

本科毕业论文（设计）答辩过程记录

院系 专业 年级

答辩人姓名 学号

毕业论文（设计）题目

毕业论文（设计）答辩过程记录：

答辩是否通过：通过（ ） 未通过（ ）

记录员 答辩小组组长签字

年 月 日 年 月 日

本科毕业论文（设计）答辩登记表

院（系）： 专业： 年级：

|  |  |
| --- | --- |
| 论文（设计）题目： | |
| 答辩人： | 学号： |
| 评阅人： | |
| 指导教师： 论文（设计）等级： | |
| 答辩小组成员： | |
| 答辩小组意见：  秘书签名：  年 月 日 | |
| 论文（设计）答辩是否通过：通过（ ） 未通过（ ） | |
| 论文（设计）最终等级： | |
| 答辩小组组长签名： | 答辩委员会主席签名： |

校级优秀毕业论文（设计）推荐表

所属院(系)： 填表日期： 年 月 日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 论文作者 | | | | 毕业论文（设计）总周数 |  |
| 姓名 | 性别 | 专业 | |
|  |  |  | |
| 指导教师 | | | |
| 姓名 | 年龄 | 专业技术职务 | 所在单位 |
|  |  |  |  |
| 毕业论文（设计）  题目 |  | | | | |
| 毕业论文（设计）主要涉及研究方向 |  | | | | |
| 毕业论文（设计）选题依据及背景 |  | | | | |
| 院系中期  检查情况 |  | | | | |
| 毕业论文（设计）的水平与  特色 |  | | | | |

|  |  |  |
| --- | --- | --- |
| 毕业论文（设计）有何实验、实践或  实习基础 | |  |
| 毕业论文（设计）期间研读书目 | |  |
| 指导教师  评语及推荐  意见 | 指导教师签字：  年 月 日 | |
| 指导教师对申报材料真实性的意见 | 指导教师签字：  年 月 日 | |
| 院系推荐  意见 | | （公章）  年 月 日 |
| 学校主管  部门意见 | | （公章）  年 月 日 |