**服务器运维管理系统**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**系统设计说明书**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式发布  [ ] 正在修改 | 文件标识： | Company-Project-SD-MODULE |
| 当前版本： | X.Y |
| 作 者： |  |
| 完成日期： |  |

版 本 历 史

目 录

[1. 文档介绍 - 4 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308521)

[1.1 文档目的 - 4 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308522)

[1.2 参考文献 - 4 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308523)

[1.3 术语与缩写解释 - 4 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308524)

[2.系统技术框架 - 5 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308525)

[3.此系统模块命名规则 - 6 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308526)

[4.模块汇总 - 6 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308527)

[5.静态模型精化图 - 7 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308528)

[6.功能时序设计 - 8 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308529)

[7.项目文件结构设计 - 9 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308530)

[8.表现层设计 - 10 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308531)

[9.数据存储设计 - 10 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308532)

[9.1 XML文件格式设计 - 10 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308533)

[9.2普通文件(含Excell等)存储格式设计 - 10 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308534)

[9.3数据库设计与优化 - 11 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308535)

[9.3.1表设计 - 11 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308536)

[9.3.2安全性设计 - 14 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308537)

[9.3.2.1 防止用户直接操作数据库的方法 - 14 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308538)

[9.3.2.2 用户帐号密码的加密方法 - 14 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308539)

[9.3.2.3 角色与权限 - 14 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308540)

[9.3.3优化 - 15 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308541)

[9.3.4数据库对象设计 - 15 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308542)

[9.4外部数据来源描述 - 15 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308543)

[11.变量配置字典 - 15 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308544)

[12整体设计要点 - 16 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308545)

[附录1模块设计规则 - 16 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308546)

[附录2 开发命名规范 - 16 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308547)

[代码开发规范 - 16 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308548)

[必须遵守的规范（Rules） - 16 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308549)

[指导性规范（Guidelines） - 17 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308550)

[最佳实践（Best Practices） - 18 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308551)

[数据库设计规范 - 18 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308552)

[1设计规范 - 18 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308553)

[2 命名规则 - 19 -](file:///C:\Users\Administrator\Desktop\项目文档模板\系统设计模板.doc#_Toc239308554)

# 1. 文档介绍

服务器运维管理系统，是一个用来统计服务器等设备使用情况的信息系统。本文档是在需求分析的基础上，详细设计结果的文档化表示。……

## 1.1 文档目的

本文档作为系统设计的结果，是程序员进行开发时必须遵守的规范。

## 1.2 参考文献

[需求分析规格说明书]

[Java项目开发规范]

## 1.3 术语与缩写解释

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| 总体结构 | 软件系统的总体逻辑结构。按照不同的设计方法，有不同的总体逻辑结构。可以采用树形的功能模块结构图或者部件（构件）的组装图 |
| 外部接口 | 本软件系统的与其他系统之间的接口，接口设施可以是中间件。 |
| 模块 | 具有功能独立、能被调用的信息单元叫做模块。 |
| 内部接口 | 软件系统内部各系统之间、各部件之间、各模板之间的接口叫做内部接口。 |
| 相关文件 | 相关文件是指当本文件内容变更后，可能引起变更的其他文件。如需求分析报告、详细设计说明书、测试计划、用户手册等。 |
| 参考资料 | 参考资料是指本文档书写时用的其他材料，如各种有关规范、模板、标准、准则等。 |

# 2.系统技术框架

此系统是一个相对独立的系统与其他系统关系不大,我们在此使用BS技术来进行实现。

整个系统共分为视图表现层、控制层、业务逻辑层和数据持久层。

* 视图表现层：为用户提供表示支持，用户通过界面层来执行相应的业务逻辑；
* 控制层：控制业务逻辑的组合和页面的转向；
* 业务逻辑层：集合业务流程、事务的处理层，由多个业务构件组成，不受数据形式，类型变化的约束
* 数据持久层：在数据库和对象之间建立映射关系，服务层本身不直接访问数据库，由数据持久层负责对数据库的访问。

## 2.1技术选型

系统实现中采用数据持久层、服务层和表现层的三层结构，灵活运行SSM（Spring、SpringMVC、Mybatis）开发框架，充分考虑系统执行效率，使系统开发简单，易于维护。

数据持久层采用Mybatis实现，为每个持久化对象建立数据操作Dao，实现对数据的增、删、查、改操作，同时还提供执行SHQL语句的方法，开发时只需在业务层封装业务处理逻辑并调用Dao层的方法进行处理，降低了Dao层的复杂度，使数据持久层变得简单、易维护。

服务层完成对数据访问层的封装，并为控制层提供调用接口。

控制层采用SpringMVC实现，在Action中定义用于与页面传递数据的属性，并采用Spring依赖注入的方式导入对应的Service。

## 2.2核心组件

* Spring
* SpringMVc
* Mybatis

# 3.此系统模块命名规则

***提示：****模块设计人员确定本软件的模块命名规则（例如类，函数，变量等），确保模块设计文档的风格与代码的风格保持一致。可以从机构的编程规范中摘取或引用（如果存在的话）。*

### 3.1数据库命名

* 表名称

表名称采用具有含义的英文单词（或缩写）进行命名，需要多个英文单词（或缩写）时以下划线“\_”进行分隔。

当表间有父子关系时，对表名进行编码，父表采用两位编码，子表采用四位编码，依次递推。这样可以从表名看出表间关系，增强可读性。

当表名没有进行编码时，说明时公用表或独立的表。

* 字段名称

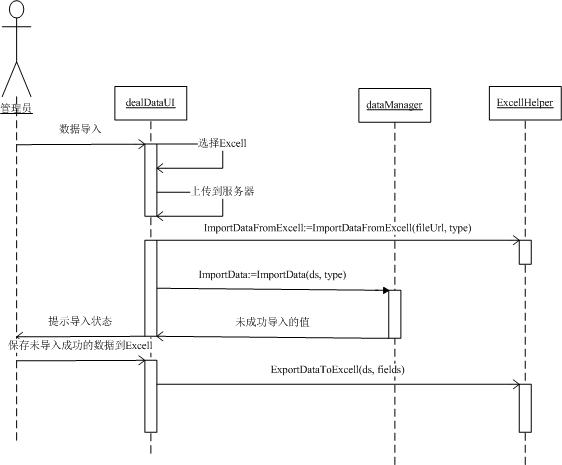
字段名称采用具有含义的英文单词（或缩写）进行命名，需要多个英文单词（或缩写）时以下划线“\_”进行分隔。

### 3.2数据库命名

# 4.模块汇总

# 5.功能时序设计

**5.1数据导入**

**

1. **通用部分设计**

### 6.1业务主键

在系统中各数据表的主键由系统提供的主键服务类的自动增加，不采用数据库自身的自动增长方式。

com.yanyuan.util BuildPrimaryKey 主键服务类

亦或采用Uuid生成

1. **数据库设计与优化**

**7.1数据库设计规范**

**7.1.1采用有意义的字段名**

每个单词要求全部用小写，用下划线“\_”来连接每个单词，例如cus\_address。

尽可能地把字段描述得清楚些。

**7.1.2 遵守3NF标准3NF规范**

(1) 表内的每一个值都只能被表达一次。

(2)表内的每一行都应该被唯一地标识（有惟一键）。

（3）表内不应该存储依赖于其他键的非键信息。

表设计应符合第三范式的规范。但如果过分追求第三范式，会造成过度规范化，即使用了大量的、小的、相互关联的表达定义数据库。在数据库对这些表中的数据库进行处理时，必须执行大量的额外工作以组合相关的数据。这种额外处理可能降低数据库是性能。在这种情况下，适当降低数据库的规范化程度以复杂处理，同时可以提高性能。

**7.1.3 字段设计规定**

（1） 具有序号含义的列尽量采用提供的列，可以有效地避免重号及跳号。

（2）数值型的列都要有默认值。

（3）明显不能为空的列，必须禁止为空。

（4）可以为空的列，应根据实际情况，设定适当的默认值。

（5） 在命名字段并为其指定数据库类型时，一定要在数据库中使其保持一致性。

**7.1.4 视图设计规则**

（1）不推荐在视图之上再建立视图。

（2）定义视图的查询不可以包含ORDER BY,COMPUTE或COMPUTE BY子句或INTO关键字。

(3) 不能创建临时视图，也不能在临时表上创建视图。

**7.1.5 索引设计规则**

（1）在考虑是否为一个列创建索引时，应考虑被索引的列是否以及如何用于查询中。

一个表如果建有大量索引会影响INSERT,UPDATE和DELETE语句的性能，因为在表中的数据库更改时，所有索引都必须进行适合的调整。另一方面，对于不需要修改数据的查询(SELECT语句)，大量索引有助于提高性能，因为SQL Server有更多的索引可供选择，以便确定以最快的速度访问数据的最佳方法。

（2）在查询经常用到的所有列上创建非聚集索引，比如：

日期列。

数量列。

在FOREIGN KEY列上建立索引。

在Order By的列上建立索引。

在范围查询的列上建立索引。

在精确匹配查询的列上建立索引。

（3）对小型表进行索引可能不会长生优化效果，因为在遍历索引以搜索数据时，花费是时间可能会比简单的表扫描还长。

**7.2 数据库（DataBase）的命名**

数据库名称=数据库内容标识。

**7.2.1表和表字段的命名**

1)表（Table）的命名

(1) 表名称=表名前缀+下划线（-）+表内容标识。例如：系统用户信息表sya-user-info.

(2) 命名应尽量反映表所存储的数据内容。例如：系统用户信息表sys-user-info.

(3) 表名前缀：以该表及与该表相关联的一系列表的内容而到一个代表统一的标识，一般采用模块名的缩词。例如：系统模块sys。

2）字段（Field）的定义

字段名称=表名缩写=下划线“-”+字段名称标识。

**7.2.2视图名**

视图的名称=“v”+下划线“-”+视图内容标识。例如：用户信息视图v-user-info。

**7.2.3存储过程名**

存储过程名称=（存储过程前缀）+下划线“-”+存储过程类型（增删改查）+存储过程内容标识。例如：校验用户密码sp-seclet-chk-user-pass。

**7.2.4命名中其他注意事项**

（1）命名都采用英文字符，禁止使用中文命名。

（2）命名都不得超过20个字符，变量名的长度限制为20个（不包括标识字符@）

**7.2.5 SQL语句的编写规范**

关键字大写：在SQL语句的编写中，凡是SQL语句的关键字一律大写，如：SELECT,ORDERBY,GROUP BY,FROM,WHERE,UPDATE,INSERT INTO,SET,BEGIN,END等。

**7.2.6索引**

idx+下划线“-”+字段标识。

**7.2.7外键**

fk+下划线“-”+字段标识。

**7.3 数据库设计**

参考《服务器运维管理数据库设计》

1. **安全性设计**

每次访问jsp、do文件时，由系统的过滤器对访问用户进行安全控制，如果用户未登陆系统直接访问系统其他界面，则是非法不能访问系统将会直接跳转到登录页面。

**8.1性能设计**

对数据库进行性能优化，建立索引，提高数据库访问效率；采用高性能数据库链接池技术，提高每次访问数据库的效率。

**8.2维护性设计**

系统采用集中式配置降低后续运行维护成本。系统硬件设备放置于统一网络机房，便于集中管理。系统软件采用友好的用户界面，整体布局符合人工设计，简捷直观的操作按钮，便于易学，易用。

**8.3 角色与权限**

提示：确定每个角色对数据库表的操作权限，如创建、检索、更新、删除等。每个角色拥有刚好能够完成任务的权限，不多也不少。在应用时再为用户分配角色，则每个用户的权限等于他所兼角色的权限之和。

|  |  |  |
| --- | --- | --- |
| 角色 | 可以访问的表与列 | 操作权限 |
| 角色A |  |  |
|  |  |
|  |  |
| 角色B |  |  |
|  |  |
|  |  |

1. **整体设计要点说明**

# 附录1模块设计规则

|  |  |
| --- | --- |
| 模块名称 |  |
| 功能描述 |  |
| 接口与属性 | **提示：**用专业的设计（开发）工具来设计本模块的接口与属性，说明函数功能、输入参数、输出参数、返回值等 |
| 数据结构与算法 | ***提示：****不论是采用经典的还是专用的数据结构与算法，都应该作必要的描述。不仅用于指导程序的实现，还可以让人们清楚的了解对象类是如何是实现的* |
| 补充说明 |  |

# 附录2 开发命名规范

## 代码开发规范

**9.1必须遵守的规范（Rules）**

1. 局部变量的名称要有意义，尽量用对应的英文命名，比如“用户姓名”变量，不要用aa bb cc等来命名，而要使用userName.
2. 不要使用单个字母的变量，如I,n,x等。而要使用index,temp等。用于循环迭代的变量例外。
3. 应该使用命名法命名方法名，属性名，类名和命名空间，且方法的命名一般将其命名为动宾短语，如

ShowDialog（）

CreateFile()

GetPath()

1. 用命名法命名局部变量和方法的参数。

以上几条综合举例如下。

publio class Person

{

private string firstName;//使用Camel方式命名

public string FirstName()//使用Pascal方式命名

-{

get{}

set{}

}

pulic void DoSomething() //使用动宾短语作为方法名

{

}

public void Buy (string productName)

{

}

}

1. 仅使用匈牙利命名法对WinForms和Web控件命名，禁止使用textboxl,gridviewl等默认命名。
2. 接口名称以大写1作为前缀，如ISortable，以便直观地区分接口和类。
3. 使用命名空间的层次性结构来组织各种类和接口。

正确：

Using Myoffice.DAL

Namespace Myoffice..BLL

{

……

}

错误:

using MyofficeDAL

namespace MyofficeBLL

{

……

}

1. 所有代码必须保证有足够的注释，注释量要求在15%以上。需要注释的内容包括以下几种。
2. 关键性语句（如关键性的变量声明，条件判断等）。
3. 类名前的文档注释（开头的注释）包括类的简单功能说明、使用方法等。
4. 具有复杂参数列表的方法。

（9） 代码缩进用Tab键，不要用空格键。

**9.2指导性规范（Guidelines）**

（1） 对于超过三屏以上的代码，建议使用region折叠。

（2） Web窗体建议以DIV方式布局。

（3） 代码中DataSet,DataTable,DataView等对象，定义时建议将ds,dt,dv作为前缀。

（4） 在一个自定义的类中，建议将类中各成员，有规律地组织起来。从上至下的顺序为常量，变量，属性，公有方法，私有方法法。可以将每个部分用region划分开。

（5） If/foreach/switch语句的嵌套不宜太多，建议不超过三层。

**9.3最佳实践（Best Practices）**

（1） 避免写太长的方法。一个典型的方法代码应该控制在100行以内（包括注释、空行、括号占用的独立一行等）。如果一个方法代码超过100行，应该考虑将其分解为不同的方法。

（2） 一个方法只完成一个任务。不要把多个任务组合到一个方法中，即使那些任务非常小。

备注：

编程的命名方式主要有Pascal和Camel两种。

Pascal: 每个单词的首字母大写，例如ProductType。

Camel: 首个单词的首字母小写，其余单词的首字母大写，例如productType。