库存管理系统

软件构架文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2016-05-30 | 1.0 | 初始创建 | 徐元明 |
| 2016-06-05 | 1.0 | 补充 | 刘洪通 |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 构架表示方式 4

3. 构架目标和约束 4

4. 用例视图 4

4.1 用例实现 4

5. 逻辑视图 4

5.1 概述 5

5.2 在构架方面具有重要意义的设计包 5

6. 进程视图 5

7. 部署视图 5

8. 实施视图 5

8.1 概述 5

8.2 层 5

9. 数据视图（可选） 5

10. 大小和性能 5

11. 质量 5

软件构架文档

# 简介

MVC它是一种设计模式，它把应用程序分成三个核心模块：模型、视图、控制器，它们各自处理自己的任务。

MVC（Model-View-Controller）应用程序结构被用来分析分布式应用程序的特征。这种抽象结构能有助于将应用程序分割成若干逻辑部件，使程序设计变得更加容易。   
  
MVC结构提供了一种按功能对各种对象进行分割的方法（这些对象是用来维护和表现数据的），其目的是为了将各对象间的耦合程度减至最小。MVC结构本来是为了将传统的输入（input）、处理（processing）、输出（output）任务运用到图形化用户交互模型中而设计的。但是，将这些概念运用于基于Web的企业级多层应用领域也是很适合的。   
  
在MVC结构中，模型（Model）代表应用程序的数据（data）和用于控制访问和修改这些数据的业务规则（business rule）。通常模型被用来作为对现实世界中一个处理过程的软件近似，当定义一个模型时，可以采用一般的简单的建模技术。   
  
当模型发生改变时，它会通知视图（View），并且为视提供查询模型相关状态的能力。同时，它也为控制器（Controller）提供访问封装在模型内部的应用程序功能的能力。   
  
一个视图（View）用来组织模型的内容。它从模型那里获得数据并指定这些数据如何表现。当模型变化时，视负责维持数据表现的一致性。视同时将用户要求告知控制器（Controller）。   
  
控制器（Controller）定义了应用程序的行为；它负责对来自视的用户要求进行解释，并把这些要求映射成相应的行为，这些行为由模型负责实现。在独立运行的GUI客户端，用户要求可能是一些鼠标单击或是菜单选择操作。在一个Web应用程序中，它们的表现形式可能是一些来自客户端的GET或POST的HTTP请求。模型所实现的行为包括处理业务和修改模型的状态。根据用户要求和模型行为的结果，控制器选择一个视作为对用户请求的应答。通常一组相关功能集对应一个控制器。

## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

本文的目的为使软件开发人员更好的理解系统，并指导软件开发工作。

## 范围

**本文适用于架构师，软件设计师**

## 定义、首字母缩写词和缩略语

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

## 参考资料

参考SRS文档

## 概述

本文包含了架构建模，用例图，ACME组件连接器视图，部署图，数据库设计图

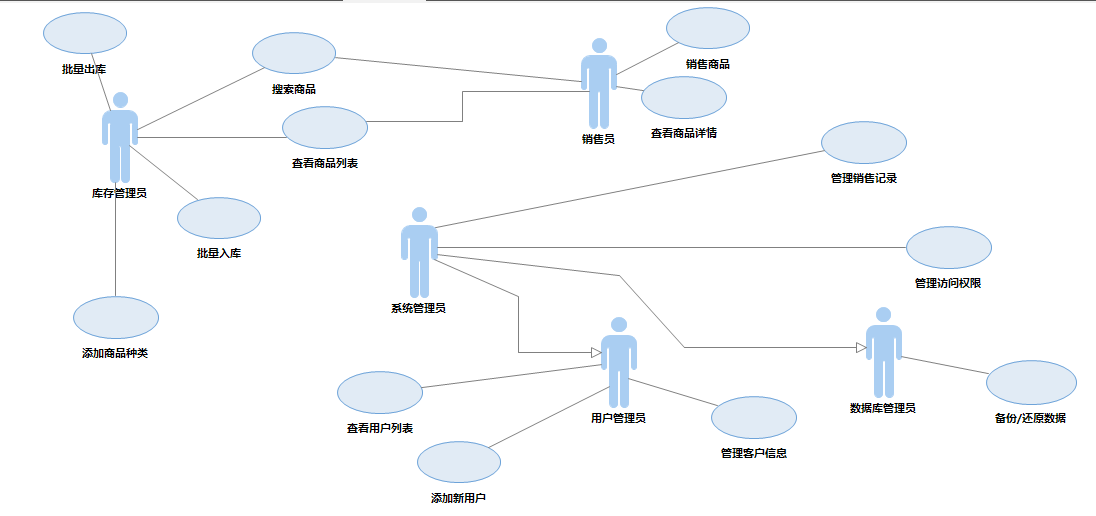
# 构架表示方式

当前系统使用用例图，ACME组件连接器视图，部署图，数据库建模图。

# 构架目标和约束

本架构使用MVC结构

# 用例视图



## 用例实现

库存管理系统需实现以下功能：

1. 查看商品列表，优先高亮显示将缺货的商品

2. 添加商品种类

3. 按商品名称、商品别名搜索商品

4. 查看商品详情

5. 批量出库

6. 批量入库

7. 查看、修改、删除销售记录

8. 标记销售单已收条/已开发票/已收款/已审计

9. 按时间/商品名称排序显示商品销售记录

10.按时间段（当天、本周、本月、本季度、本年、自定义）查询销售记录

11. 添加新客户

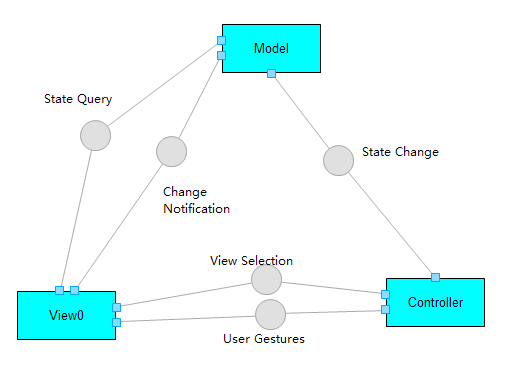
12. 查看客户列表

13.查询、修改、删除客户信息

14.访问控制

15.数据库导入导出

# 逻辑视图

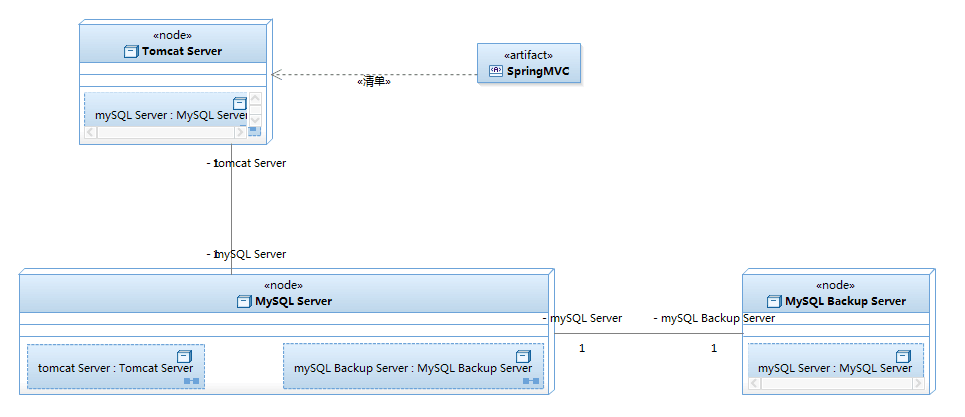


## 概述

[逻辑视图](http://www.baike.com/sowiki/%E9%80%BB%E8%BE%91%E8%A7%86%E5%9B%BE?prd=content_doc_search)（[Logical View](javascript:linkredwin('Logical%20View');)，又叫[设计视图](http://www.baike.com/sowiki/%E8%AE%BE%E8%AE%A1%E8%A7%86%E5%9B%BE?prd=content_doc_search)（[Design View](javascript:linkredwin('Design%20View');)））是为了便于理解系统设计的结构与组织，在“分析设计”[工作流](http://www.baike.com/sowiki/%E5%B7%A5%E4%BD%9C%E6%B5%81?prd=content_doc_search)程中使用了名为逻辑[视图](http://www.baike.com/sowiki/%E8%A7%86%E5%9B%BE?prd=content_doc_search)的构架视图。系统只有一个逻辑视图，该视图以图形方式说明关键的[用例](http://www.baike.com/sowiki/%E7%94%A8%E4%BE%8B?prd=content_doc_search)实现、[子系统](http://www.baike.com/sowiki/%E5%AD%90%E7%B3%BB%E7%BB%9F?prd=content_doc_search)、包和[类](http://www.baike.com/sowiki/%E7%B1%BB?prd=content_doc_search)，它们包含了在构架方面具有重要意义的行为。逻辑视图在每次迭代过程中都会加以改进。

# 进程视图

# *D:\在职研究生\SoftwareArchitectureTeamC\时序图.png*部署视图



# 实施视图

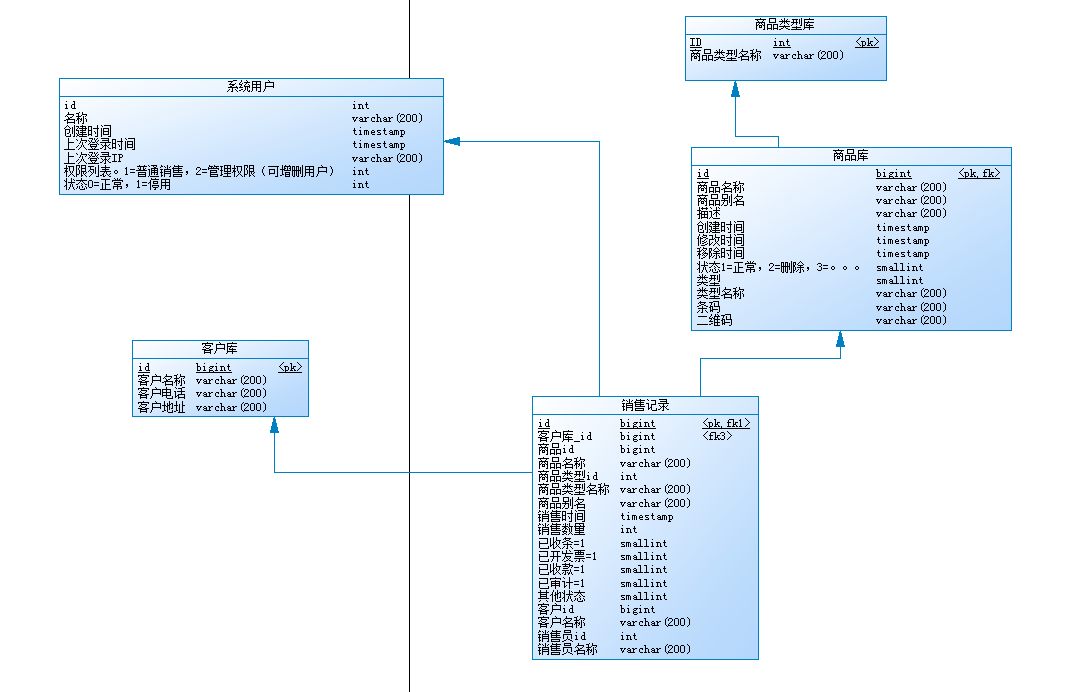
## 概述

实施视图的作用是获取为实施制定的构架决策。实施视图通常包括以下内容：

列举实施模型中的所有子系统。

说明子系统如何组织为层次和分层结构的构件图。

说明子系统间的导入依赖关系的图解。



# 大小和性能

当数据库容量达到一定规模，初始打开软件过程需要1S内完成；

每次操作须在0.1S内完成；

# 质量

可运行程序要兼容主流的Windows操作系统和主流的浏览器（如IE9+、Chrome、FireFox等）；

提供数据库自动备份功能，预防重要数据丢失；

采用加密的方式保存敏感信息，尽可能降低用户数据泄漏的可能性。