**服装销售管理系统**

盈利统计详细设计说明书

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式发布  [ ] 正在修改 | 文件标识： |  |
| 当前版本： | 1.0 |
| 作 者： | 刘光月 |
| 完成日期： | 2018/7 |

**目录**

[1 引言 5](#_Toc390869734)

[1.1 编写目的 5](#_Toc390869735)

[1.2 背景 5](#_Toc390869736)

[1.3 参考资料 5](#_Toc390869737)

[1.4 术语定义及说明 5](#_Toc390869738)

[2 设计概述 5](#_Toc390869739)

[2.1 任务和目标 5](#_Toc390869740)

[2.1.1 需求概述 5](#_Toc390869741)

[2.1.2 运行环境概述 5](#_Toc390869742)

[2.1.3 条件与限制 6](#_Toc390869743)

[2.1.4 详细设计方法和工具 6](#_Toc390869744)

[3 系统详细需求分析 6](#_Toc390869745)

[3.1 详细需求分析 6](#_Toc390869746)

[3.2 详细系统运行环境及限制条件分析接口需求分析 6](#_Toc390869747)

[4 总体方案确认 6](#_Toc390869748)

[4.1 系统总体结构确认 6](#_Toc390869749)

[4.2 系统详细界面划分 7](#_Toc390869750)

[4.2.1 应用系统与支撑系统的详细界面划分 7](#_Toc390869751)

[4.2.2 系统内部详细界面划分 7](#_Toc390869752)

[5 系统详细设计 7](#_Toc390869753)

[5.1 系统程序代码架构设计 7](#_Toc390869754)

[5.1.1 UI(User Interface)用户界面表示层 7](#_Toc390869755)

[5.1.2 BLL(Business Logic Layer)业务逻辑层 8](#_Toc390869756)

[5.1.3 DAL(Data Access Layer)数据访问层 8](#_Toc390869757)

[5.1.4 Common类库 8](#_Toc390869758)

[5.1.5 Entity Class实体类 8](#_Toc390869759)

[5.2 系统结构设计及子系统划分 8](#_Toc390869760)

[5.3 系统功能模块详细设计 9](#_Toc390869761)

[5.3.1 XX子系统 9](#_Toc390869762)

[.1 XX模块 9](#_Toc390869763)

[列表和分页 9](#_Toc390869764)

[创建XX 9](#_Toc390869765)

[.2 XX模块 9](#_Toc390869766)

[XX列表 9](#_Toc390869767)

[XX修改 9](#_Toc390869768)

[5.3.2 XX子系统 9](#_Toc390869769)

[5.3.6.1 用户管理模块 9](#_Toc390869770)

[5.3.6.2 角色管理模块 14](#_Toc390869771)

[5.3.6.3 系统设置模块 14](#_Toc390869772)

[5.3.6.4 系统登录注销模块 14](#_Toc390869773)

[5.4 系统界面详细设计 14](#_Toc390869774)

[5.4.1 外部界面设计 14](#_Toc390869775)

[5.4.2 内部界面设计 14](#_Toc390869776)

[5.4.3 用户界面设计 14](#_Toc390869777)

[6 数据库系统设计 14](#_Toc390869778)

[6.1 设计要求 14](#_Toc390869779)

[6.2 信息模型设计 14](#_Toc390869780)

[6.3 数据库设计 14](#_Toc390869781)

[6.3.1 设计依据 14](#_Toc390869782)

[6.3.2 数据库种类及特点 15](#_Toc390869783)

[6.3.3 数据库逻辑结构 15](#_Toc390869784)

[6.3.4 物理结构设计 15](#_Toc390869785)

[6.3.5 数据库安全 15](#_Toc390869786)

[6.3.6 数据字典 15](#_Toc390869787)

[7 信息编码设计 15](#_Toc390869788)

[7.1 代码结构设计 15](#_Toc390869789)

[7.2 代码编制 15](#_Toc390869790)

# 引言

## 编写目的

## 背景

## 参考资料

## 术语定义及说明

# 设计概述

## 任务和目标

### 需求概述

### 运行环境概述

### 条件与限制

*详细描述系统所受的内部和外部条件的约束和限制说明。包括业务和技术方面的条件与限制以及进度、管理等方面的限制。*

### 详细设计方法和工具

*简要说明详细设计所采用的方法和使用的工具。如HIPO图方法、IDEF（I2DEF）方法、E－R图，数据流程图、业务流程图、选用的CASE工具等，尽量采用标准规范和辅助工具。*

# 系统详细需求分析

*主要对系统级的需求进行分析。首先应对需求分析提出的企业需求进一步确认，并对由于情况变化而带来的需求变化进行较为详细的分析。*

## 详细需求分析

*包括：*

* *详细功能需求分析*
* *详细性能需求分析*
* *详细资源需求分析*
* *详细系统运行环境及限制条件分析*

## 详细系统运行环境及限制条件分析接口需求分析

*包括：*

* *系统接口需求分析*
* *现有硬、软件资源接口需求分析*
* *引进硬、软件资源接口需求分析*

# 总体方案确认

*着重解决系统总体结构确认及界面划分问题。*

## 系统总体结构确认

*对系统组成、逻辑结构及层次进行确认，对应用系统、支撑系统及各自实现的功能进行确认，细化集成设计及系统工作流程，特别要注意因软件的引进造成的系统本身结构和公司其他系统的结构变化。包括：*

1. *系统组成、逻辑结构及层次确认*
2. *应用系统结构确认*
3. *支撑系统结构确认*
4. *系统集成确认*
5. *系统工作流程确认*

## 系统详细界面划分

### 应用系统与支撑系统的详细界面划分

*应用系统与支撑系统之间的界面包括系统主服务器与其他服务器的服务范围及访问方式，网络及数据库对应用系统的支撑方式，全局数据的管理与存取方式等。*

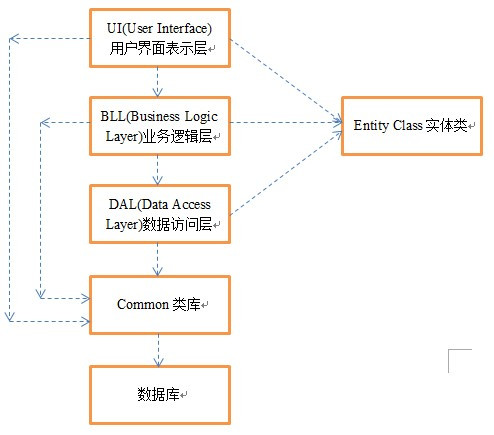
### 系统内部详细界面划分

*系统各功能之间的界面包括覆盖范围，模块间功能调用涉及到的系统模块及方法，全局数据格式，系统性能要求等。*

# 系统详细设计

## 系统程序代码架构设计

系统采用三层架构模型，将应用系统划分为用户界面表示层、业务逻辑层、数据访问层，以及Entity Class实体类、Common类库组成，各层的关系如下图所示：



### UI(User Interface)用户界面表示层

负责与用户进行交互，显示、接受数据，与此同时，做一些简单逻辑处理，如：输入数据有效性判断、显示各种异常、处理Dataset记录集数据。它只与BLL(Business Logic Layer)业务逻辑层、Entity Class实体类两个项目发生关联，可能与Common类库发生关联。

### BLL(Business Logic Layer)业务逻辑层

是整个系统的核心，它承担了所有的逻辑判断，实现了程序的功能，它是灵活的。BLL层既是调用者，又是被调用者，因此，要适当的进行设计达到解耦的效果。BLL层只关联DAL层和实体类，可能关联Common类库。虽然BLL层被U层调用，但是BLL层无需关心UI层的情况。数据库中每个表都对应一个BLL类，为了达到解耦效果，BLL类不能直接调用其他表的DAL类，可以BLL类之间相互调用。

### DAL(Data Access Layer)数据访问层

提供数据访问的接口，没有任何逻辑。在接口中对数据库操作语句进行组合装配。DAL层一般关联Common类库中的最底层，最基础的数据库类（比如：链接数据库），必须关联Entity Class实体类项目。DAL层只是数据库的管理者，但不是访问者，不直接与数据库发生关联。数据库中每个表都对应一个DAL层的接口(访问控制)类。

### Common类库

用于存放公用的类。最常用的就是数据库访问类，比如：链接字符串，数据库引擎类。它直接与数据库进行机械式的交换，无任何逻辑。

### Entity Class实体类

相当于加强的数据结构，实现了对数据的封装。数据库中每个表都对应一个实体类，表的字段就是实体类的属性，类型一一对应。UI、BLL、DAL这三层的交互主要就是通过实体类作为参数，并Return回信息。

## 系统结构设计及子系统划分

根据业务和功能，将系统的逻辑结构划分为XX子系统、XX子系等6个子系统，如下图所示：

服装销售管理系统

客服管理

子系统

搜索管理

子系统

账号注册

子系统

盈利统计

子系统

数据分析

子系统

信息录入

子系统

各个子系统按照功能角度分解，划分出若干不同的功能模块，如下面各图所示：

## 系统功能模块详细设计

*按结构化设计方法，在系统功能逐层分解的基础上，对系统各功能模块或子系统进行设计。此为详细设计的主要部分之一。*

*用层次图描述系统的总体结构、功能分解及各个模块之间的相互调用关系和信息交互，用IPO图或其他方法描述各模块完成的功能。 以上建议采用HIPO图进行功能分解与模块描述，更高的要求建议采用IDEF0方法进行功能模型设计。*

*详细设计应用系统的各个构成模块完成的功能及其相互之间的关系，用IPO或结构图描述各模块的组成结构、算法、模块间的接口关系，以及需求、功能和模块三者之间的交叉参照关系。*

*每个模块的描述说明可参照以下格式：*

***模块编号：***

***模块名称：***

***输入：***

***处理：***

***算法描述：***

***输出：***

*其中处理和算法描述部分主要采用伪码或具体的程序语言完成。*

*对详细设计更高的要求建议用IDEF0图进行各功能模块的设计。*

*如果对软件需进行二次开发（包括功能扩展、功能改造、用户界面改造等），则相应的设计工作应该设立子课题完成。*

**注意：**算法描述主要是说明BLL层的代码逻辑，UI层只做简单的输入验证和界面显示功能。

### XX子系统

### XX模块

### 列表和分页

### 创建XX

### XX模块

### XX列表

### XX修改

### XX子系统

#### 用户管理模块

**模块描述：**管理系统用户，包括添加用户并赋予角色、修改用户资料和角色、删除用户。

**主要功能：**添加用户、修改用户、删除用户、列表和分页。

##### 列表和分页

1. **流程图**

省略

1. **输入项**

分页索引 页面传参数 页面加载时为1

1. **输出项**
2. 列表详细记录；
3. 当前页码；
4. 分页总数；
5. 记录总数。
6. **算法描述**

系统管理用户数据量不大，而且该功能使用频率不高，可以不用优化分页，直接获取用户表全部记录，建议UI层使用gridview控件，调用ExamSys.BLL.System\_admin\_info GetAllList()绑定控件，利用gridview自带分页功能，显示分页按钮、当前页、分页总数、记录总数、

##### 添加用户

1. **流程图**

输入用户资料

选择角色

加密密码

填写合规

用户信息保存至表System\_admin\_info

验证必填项

验证用户名

是否存在

是否成功

分拆角色ID字符串

循环字符串数组

信息保存至表Dict\_admin\_vs\_roles

返回成功信息

是

否

是

返回失败信息

否

是

否

操作记录日志

1. **输入项**

用户ID string 必填

姓名 string 必填

用户密码 string 必填

确认密码 string 必填

联系电话 string

E-mail string

状态 char

用户角色ID string

1. **输出项**
2. 成功，UI提示添加用户完成；
3. 失败，UI提示具体信息。
4. **算法描述**
5. 前端js判断必填项的输入和二次密码是否相同，角色可以不选；
6. BLL须验证必填项，用户名是唯一的不能重复；
7. 加密密码使用MD5.Encrypt(string String, string Key)，Key = "XXXXXXXX"；
8. 用户信息保存至表System\_admin\_info，ExamSys.DAL.System\_admin\_info Add(ExamSys.Model.System\_admin\_info model) ；
9. 分拆角色ID字符串并循环字符串数组，信息保存至表Dict\_admin\_vs\_roles，ExamSys.BLL.Dict\_admin\_vs\_roles Add(ExamSys.Model.Dict\_admin\_vs\_roles model)。

##### 修改用户

1. **流程图**

输入用户资料

选择角色

填写合规

用户信息保存至表System\_admin\_info

验证必填项

加密密码

是否成功

分拆角色ID字符串

循环字符串数组

信息保存至表Dict\_admin\_vs\_roles

返回成功信息

是

是

返回失败信息

否

否

根据用户ID删除表Dict\_admin\_vs\_roles记录

是否成功

是

否

操作记录日志

1. **输入项**

用户ID string 页面传参数

姓名 string 必填

用户密码 string 必填

确认密码 string 必填

联系电话 string

E-mail string

状态 char

用户角色ID string

1. **输出项**
2. 成功，UI提示修改用户完成；
3. 失败，UI提示具体信息。
4. **算法描述**
5. 前端js判断必填项的输入和二次密码是否相同，角色可以不选；
6. 加密密码使用Chukeh.Security.MD5. Encrypt(string String, string Key)，Key = "LinLiYan"；
7. 根据用户ID删除表Dict\_admin\_vs\_roles的记录ExamSys.BLL.Dict\_admin\_vs\_roles DeleteByAdminID(string admin\_id) ；
8. 分拆角色ID字符串并循环字符串数组，信息保存至该表ExamSys.BLL.Dict\_admin\_vs\_roles Add(ExamSys.Model.Dict\_admin\_vs\_roles model)。

##### 删除用户

1. **流程图**

列表中

点击删除按钮

是否成功

返回成功信息

是

返回失败信息

否

根据用户ID删除表Dict\_admin\_vs\_roles记录

是否成功

是

否

根据用户ID删除表System\_admin\_info记录

操作记录日志

1. **输入项**

用户ID string 页面传参数

1. **输出项**
2. 成功，UI提示删除用户完成，刷新列表；
3. 失败，UI提示具体信息。
4. **算法描述**
5. 根据用户ID删除表System\_admin\_info的记录ExamSys.DAL.System\_admin\_info Delete(string admin\_id) ；
6. 根据用户ID删除表Dict\_admin\_vs\_roles的记录ExamSys.BLL.Dict\_admin\_vs\_roles DeleteByAdminID(string admin\_id) 。

#### 角色管理模块

#### 系统设置模块

#### 系统登录注销模块

## 系统界面详细设计

*系统界面说明应用系统软件的各种接口。整个系统的其他接口（如系统硬件接口、通讯接口等）在相应的部分说明。*

### 外部界面设计

*根据系统界面划分进行系统外部界面设计，对系统的所有外部接口（包括功能和数据接口）进行设计。*

### 内部界面设计

*设计系统内部各功能模块间的调用关系和数据接口。*

### 用户界面设计

*规定人机界面的内容、界面风格、调用方式等，包括所谓的表单设计、报表设计和用户需要的打印输出等设计。此部分内容可能比较多。*

# 数据库系统设计

*此数据库设计可单独成册，尤其对大型的数据库应用系统，即有一个单独的《数据库设计说明书》。*

## 设计要求

## 信息模型设计

*确定系统信息的类型（实体或视图），确定系统信息实体的属性、关键字及实体之间的联系， 详细描述数据库和结构设计，数据元素及属性定义，数据关系模式，数据约束和限制。*

## 数据库设计

### 设计依据

*说明数据被访问的频度和流量，最大数据存储量，数据增长量，存储时间等数据库设计依据。*

### 数据库种类及特点

*说明系统内应用的数据库种类、各自的特点、数量及如何实现互联，数据如何传递。*

### 数据库逻辑结构

*说明数据库概念模式向逻辑模式转换所采用的方法论及工具，完成数据库概念模式向逻辑模式的转换。 详细列出所使用的数据结构中每个数据项、记录和文件的标识、定义、长度及它们之间的相互关系。此节内容为数据库设计的主要部分。*

### 物理结构设计

*列出所使用的数据结构中每个数据项的存储要求、访问方法、存取单位和存取物理关系等。建立系统程序员视图，包括：*

1. *数据在内存中的安排，包括对索引区、缓冲区的设计；*
2. *所使用的外存设备及外存空间的组织，包括索引区、数据块的组织与划分；*
3. *访问数据的方式方法。*

### 数据库安全

*说明数据的共享方式，如何保证数据的安全性及保密性。*

### 数据字典

*编写详细的数据字典。 对数据库设计中涉及到的各种项目，如数据项、记录、系、文卷模式、子模式等一般要建立起数据字典，以说明它们的标识符、同义名及有关信息。*

# 信息编码设计

## 代码结构设计

*确认信息分类编码总体方案，进行分类代码结构设计。*

## 代码编制

*按代码结构编制信息代码*