合同管理系统设计说明书

[1 Introduction 简介 5](#_Toc18487)

[1.1 Purpose 目的 5](#_Toc4851)

[1.2 Scope 范围 5](#_Toc28841)

[1.2.1 Name 软件名称 5](#_Toc21237)

[1.2.2 Functions 软件功能 5](#_Toc23098)

[1.2.3 Applications软件应用 5](#_Toc6307)

[1.3 软件系统上下文定义 5](#_Toc1197)

[1.4 Design Considerations设计思路 6](#_Toc7913)

[2 Level 1 Design Description 第1层设计 8](#_Toc30872)

[2.1 System Architecture系统结构 8](#_Toc26645)

[2.2 Representation of the Business Flow业务流程说明 9](#_Toc749)

[2.3 Decomposition Description分解描述 9](#_Toc6568)

[2.3.1 注册 9](#_Toc30238)

[2.3.2 登录/登出 10](#_Toc15223)

[2.3.3 合同管理 10](#_Toc25903)

[2.3.3.1 起草合同 11](#_Toc7408)

[2.3.3.2 会签合同 12](#_Toc12389)

[2.3.3.3 定稿合同 12](#_Toc31113)

[2.3.3.4 审批合同 13](#_Toc14466)

[2.3.3.5 签订合同 14](#_Toc32159)

[2.3.4 查询统计 14](#_Toc29853)

[2.3.4.1 合同信息查询 14](#_Toc5969)

[2.3.4.2 合同流程查询 14](#_Toc18467)

[2.3.5 基础数据管理 15](#_Toc6929)

[2.3.5.1 合同信息管理 15](#_Toc8584)

[2.3.5.2 客户信息管理 15](#_Toc16911)

[2.3.6 系统管理 15](#_Toc28577)

[2.3.6.1 分配合同 15](#_Toc30888)

[2.3.6.2 权限管理 16](#_Toc10953)

[2.3.6.3 日志管理 17](#_Toc25971)

[2.4 Interface Description接口描述 17](#_Toc22592)

[3 Database Design 数据库设计 18](#_Toc5548)

[3.1 概念模型 18](#_Toc24927)

[3.2 物理模型 18](#_Toc25102)

[3.3 数据库表设计 18](#_Toc9627)

[3.4 基础数据配置 21](#_Toc30550)

[4 UI Design 界面设计 24](#_Toc9581)

[4.1 登录页面 24](#_Toc2055)

[4.2 管理页面 25](#_Toc21902)

[4.3 合同起草页面 25](#_Toc6439)

[4.4 合同列表页面 26](#_Toc25916)

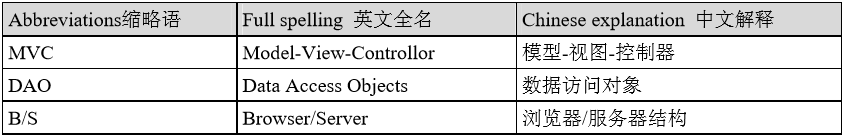
[5 Detailed Design 详细设计 27](#_Toc10767)

[6 Error Design 出错处理设计 32](#_Toc22355)

[6.1 概述 32](#_Toc14206)

[6.2 自定义异常类 32](#_Toc3458)

Keywords 关键词： 合同 流程 权限 Abstract 摘 要： 本系需求规格说明书，说明了合同管理系统的功能结构，详细功能介绍、数据字典和性 能、接口需求等。通过系统用例图、活动图和功能结构图等描述，讲解了合同管理系统的的 详细需求，系统包含用户注册，登录，合同管理，查询统计，基础信息管理和系统管理等功 能。 List of abbreviations 缩略语清单：



1 Introduction 简介

1.1 Purpose 目的

该需求规格说明书是关于合同管理系统的功能需求和性能需求的描述，该说明书的预期 读者为： 用户； 项目管理人员； 测试人员； 设计人员； 开发人员。 重点描述了流程管理和权限管理两个模块的功能需求，明确所要开发的软件应具有的功 能、性能与界面，使系统分析人员及软件开发人员能清楚地了解用户的需求。

1.2 Scope 范围

1.2.1 Name 软件名称

合同管理系统

1.2.2 Functions 软件功能

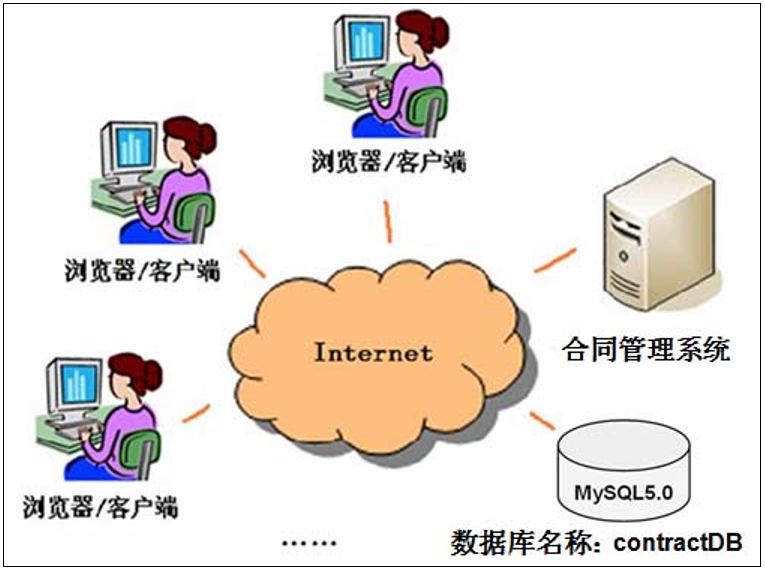
参考《合同管理系统需求规格说明书》

1.2.3 Applications软件应用

合同管理系统，从收集合同基础参数开始，进行合同起草，与用户进行谈判、会签并修 订合同，签订合同，记录合同执行过程，进行合同执行跟踪管理直到合同终止，贯穿合同管 理全过程。

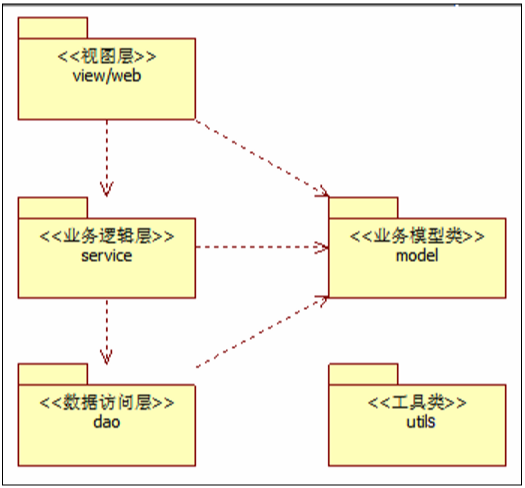
1.3 软件系统上下文定义

合同管理系统为一个B/S结构的系统，系统的访问结构如下：



1.4 Design Considerations设计思路

1. 系统的三层架构，包的结构图如下



系统分为表示层，业务逻辑层，数据访问层，分别对应包(com.ruanko.web)，

(com.ruanko.service)，(com.ruanko.dao)，他们通过调用业务模型类(即com.ruanko.model包中 的类)来完成数据的传递，其中有一部分类会被多处公用，就像工具一样,，这时把他们统一 放在com.ruanko.utils包中。

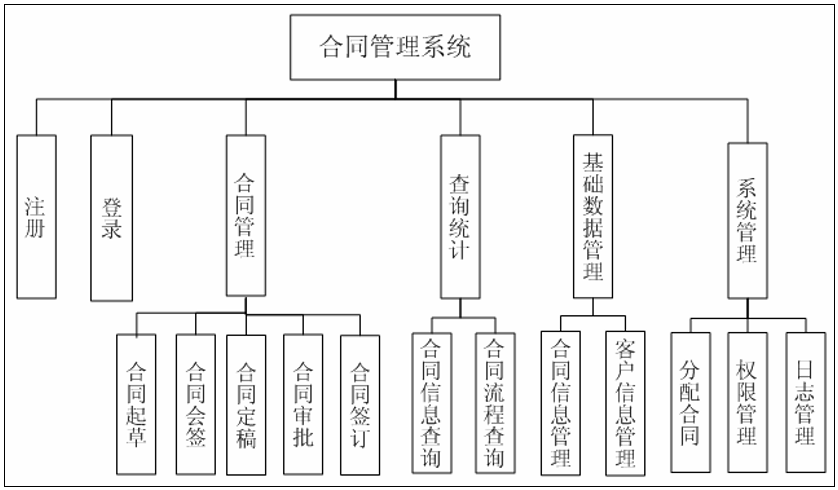
2、程序的开发模式和三层结构 系统使用Model 2开发模式：JSP+Servlet+JavaBean(MVC)，它与三层架构的结合中， Servlet(控制器)和JSP页面(视图)充当表示层，JavaBean即模型(包含业务逻辑，数据访问和业 务模型类等)充当业务逻辑层和数据访问层。

1. 三层结构与Model 2结合的包层次结构



2 Level 1 Design Description 第1层设计

2.1 System Architecture系统结构



1、合同管理 合同操作员登录后，可以进行起草操作；点击其中的待会签合同，可以对分配给自己的 合同进行会签操作；点击待定稿合同，可以对合同进行定稿操作；点击待审批合同，可以对 合同进行审批操作；相应的，也可以对合同进行签订操作。

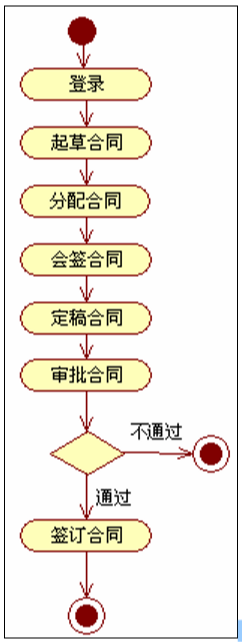
2、查询统计 管理员登录系统后，可以对合同基本信息进行查询，如根据名称等进行查询，除此之外， 也可以跟踪合同流程的操作步骤，查询合同所处的状态。

3、基础数据管理 管理员进入系统后，能够对客户信息进行新增、查询、修改、删除。也可以对合同进行 新增、查询、修改、删除。

4、系统管理 管理员登录系统后，可以对起草的合同进行人员分配，对系统的功能模块进行权限分配， 除此之外，对于系统中数据库的操作，应建立操作日志，必要时可返回单次或批量操作以前 的状态，日志操作是伴随用户操作功能是自动发生并入库。

2.2 Representation of the Business Flow业务流程说明

1. 合同管理流程：



2.3 Decomposition Description分解描述

2.3.1 注册

1、功能设计描述 在注册页面填写注册信息，要求填写用户名（必填，4~20个汉字、字母、数字或下划线 组成的字符串，且不能有重名注册），密码（必填，6~12个字符），确认密码（和密码保持 一致），进行注册。 提交注册信息时，有必填项未填写或格式不正确时，提示用户正确填写相应项，当注册 失败时在注册页面显示“注册失败！”。当所有信息填写正确，提交表单，注册成功跳转到登 录页面，如果程序出现异常则跳转到异常页面。

2、 功能实现 用户提交注册请求，RegisterServlet从注册页面中取得用户注册信息，调用业务逻辑层 UserService的register()方法处理注册业务。

UserService调用数据访问层UserDaoImpl的isExist()处理同名验证逻辑，不存在同名用户 则调用add()方法保存用户的注册信息。 各层之间通过实体类User对象传递参数。

2.3.2 登录/登出

1、功能设计描述 在登录页面填写用户名、密码，点击“登录”按钮进行登录。要求用户名和密码不能为 空，为空则在登录页面给出错误提示。登录失败时在登录页面提示 “用户名或密码错误！”； 系统异常，跳转到异常页面；登录成功跳转到新用户页面，并显示当前登录用户名。 已登录的用户，点击新用户页面的“注销登录”链接，清除用户的登录状态。返回到登 录页面。

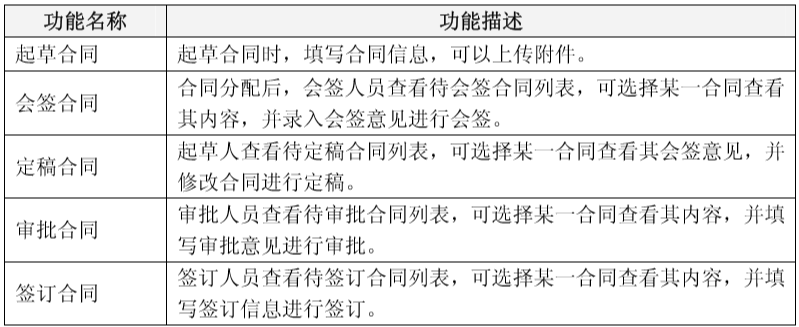
2、功能实现 （1）登录 用户提交登录请求，LoginServlet从登录页面中取得用户登录信息，调用业务逻辑层 UserService的login()方法处理登录业务。 UserService调用数据访问层UserDaoImpl.login()方法查询匹配用户信息，返回用户编号。

（2）登出 LogoutServlet接收新用户页面头部的“注销登录”请求，移除session中的用户信息，跳 转到登录页面。

2.3.3 合同管理

1、简介 合同操作员登录后，如果拥有起草权限，可以进行起草操作；点击其中的待会签合同， 可以对分配给自己的合同进行会签操作；点击待定稿合同，可以对合同进行定稿操作；点击 待审批合同，可以对合同进行审批操作；相应的，也可以对合同进行签订操作。

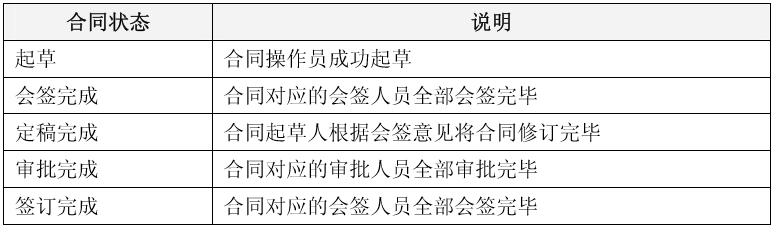
2、功能列表



2.3.3.1 起草合同

1. 功能设计描述 合同操作员在起草合同页面填写合同名称，开始时间，结束时间，合同内容，客户，并 可以上传合同附件。 成功起草合同后，合同状态为“起草”，等待管理员对该合同进行人员分配。

合同的状态：



2、 功能实现 用户起草合同，DraftServlet使用doPost()处理用户的起草合同请求，该方法调用 ContractService中的draft()方法完成起草合同逻辑。 在draft()方法中，首先生成合同编号，并设置到Contract中，调用ContractDao中add()保 存合同信息，如果起草成功，则调用ConStateDao中的add()保存合同状态信息。

2.3.3.2 会签合同

1、功能设计描述 会签人员查看待会签合同列表，可选择某一合同查看其内容，并录入会签意见进行会签。

2、 功能实现 (1) 查看“待会签”合同列表 涉及的会签人员点击管理页面的“待会签合同”超链接，ToDhqhtListServlet获取查询请 求，由doPost()方法调用ContractService中的getDhqhtList()处理查询待会签合同请求。 (2) 查看合同详情 会签人员针对某一个合同，点击合同名称超链接，ContractDetailServlet获取查询请求， 由doPost()方法调用ContractService中的getContractDetail()方法查询合同详情。 (3) 会签合同 会签人员针对某一个合同，点击“会签”超链接，ToAddHQOpinionServlet获取查询请 求，由doPost()方法调用ContractService中的getContract()方法查询合同基本信息。 当会签人员在会签合同页面，输入会签意见并提交时，AddHQOpinionServlet获取会签 合同请求，调用ContractService中的counterSign()方法处理会签合同业务逻辑。

2.3.3.3 定稿合同

1. 功能设计描述
2. 起草人查看待定稿合同列表，可选择某一合同查看其定稿意见，并修改合同进行定稿。

5. 2、 功能实现 (1) 查看“待定稿”合同列表 起草人点击管理页面的“待定稿合同”超链接，ToDdghtListServlet获取查询请求，由 doPost()方法调用ContractService中的getDdghtList()处理查询待定稿合同请求。 (2) 查看会签意见 起草人针对某一个合同，点击“会签意见”超链接，ShowHQOpinionServlet获取查询请 求，由doPost()方法调用ContractService中的showHQOpinion()方法查询会签意见。 (3) 定稿合同 起草人针对某一个合同，点击“定稿”超链接，ToDgContractServlet获取查询请求，由 doPost()方法调用ContractService中的getContract()方法查询合同基本信息。 当起草人在定稿合同页面，输入合同内容并提交时，DgContractServlet获取定稿合同请 求，调用ContractService中的finalize()方法处理定稿合同业务逻辑。

2.3.3.4 审批合同

1、功能设计描述 审批人员查看待审批合同列表，可选择某一合同查看其内容，并填写审批意见进行审批。

1. 功能实现 (1) 查看“待审批”合同列表 涉及的审批人员点击管理页面的“待审批合同”超链接，ToDshphtListServlet获取查询 请求，由doPost()方法调用ContractService中的getDshphtList()处理查询待审批合同请求。 (2) 审批合同 审批人员针对某一个合同，点击“审批”超链接，ToAddSHPOpinionServlet获取查询请 求，由doPost()方法调用ContractService中的getContract()方法查询合同基本信息。 当审批人员在审批合同页面，输入审批意见并提交时，AddSHPOpinionServlet获取审批 合同请求，调用ContractService中的approve()方法处理审批合同业务逻辑。

2.3.3.5 签订合同

1、功能设计描述 签订人员查看待签订合同列表，可选择某一合同查看其内容，并填写签订信息进行签订。

1. 功能实现 (1) 查看“待签订”合同列表 涉及的签订人员点击管理页面的“待签订合同”超链接，ToDqdhtListServlet获取查询请 求，由doPost()方法调用ContractService中的getDqdhtList()处理查询待签订合同请求。 (2) 签订合同 签订人员针对某一个合同，点击“签订”超链接，ToAddQDOpinionServlet获取查询请 求，由doPost()方法调用ContractService中的getContract()方法查询合同基本信息。 当签订人员在签订合同页面，输入签订意见并提交时，AddQDOpinionServlet获取签订

2.3.4 查询统计

2.3.4.1 合同信息查询

1、功能设计描述 根据合同名称分页查询合同信息。

2、 功能实现 管理员点击管理页面的“合同信息”超链接，QueryConInfoServlet获取查询请求，由 doPost()方法调用ContractService中的queryContract()处理查询合同信息请求。

2.3.4.2 合同流程查询

1、功能设计描述 根据合同名称分页查询合同流程。

1. 功能实现 管理员点击管理页面的“流程查询”超链接，QueryProcessServlet获取查询请求，由doPost() 方法调用ContractService中的queryProcess()处理查询合同流程请求。

2.3.5 基础数据管理

2.3.5.1 合同信息管理

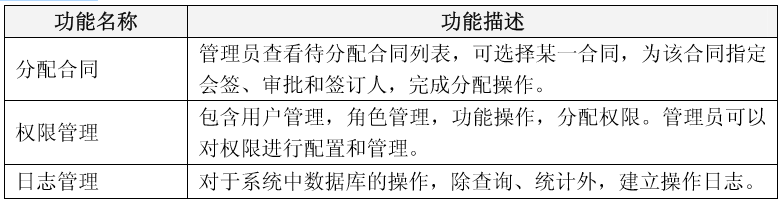
对合同信息进行新增，查询,修改和删除操作。 根据实际需要进行该功能设计。

2.3.5.2 客户信息管理

对客户信息进行新增，查询，修改和删除操作。 根据实际需要进行该功能设计。

2.3.6 系统管理

1. 简介 管理员可以分配合同，进行权限管理和日志管理等。
2. 功能列表



2.3.6.1 分配合同

1、功能设计描述 管理员选择状态为“起草”的待分配的合同，指定参与会签、审批、签订的人员，完成 分配。

1. 功能实现 (1) 查看“待分配”合同列表 管理员点击管理页面的“待分配合同”超链接，ToDfphtListServlet获取查询请求，由 doPost()方法调用ContractService中的getDfphtList()处理查询待分配合同请求。 (2) 分配合同 管理员针对某一个合同，点击“分配”超链接，ToAssignOperServlet获取查询请求，由 doPost()方法调用ContractService中的getContract()方法查询合同基本信息。调用UserService 中的getUserListByRoleId()方法查询用户基本信息。 当管理员在分配合同页面，分配完相关的操作人员并提交时，AssignOperServlet获取分 配合同请求，调用ContractService中的distribute()方法处理分配合同业务逻辑。

2.3.6.2 权限管理

权限管理包含用户管理，角色管理，功能操作，分配权限。管理员可以对权限进行配置 和管理。 本次针对“分配权限”功能进行设计，其他功能根据实际需要进行该功能设计。 1、功能设计描述 管理员登录系统后，查看待分配权限的用户列表，然后，针对某一用户，点击“授权” 超链接，转到分配用户权限页面，进行分配操作。

1. 功能实现 (1) 查看用户权限列表 管理员点击管理页面的“配置权限”超链接，ToYhqxListServlet获取查询请求，由doPost() 方法调用UserService中的getYhqxList()处理查询待分配权限的用户请求。 (2) 权限分配 管理员针对某一个用户，点击“授权”超链接，ToAssignPermServlet获取查询请求，由 doPost()方法调用UserService中的getRoleList()方法查询角色列表信息。 当管理员在分配权限页面，分配完用户角色并提交时，AssignPermServlet获取分配权限 请求，调用UserService中的assignPermission()方法处理分配权限业务逻辑。

2.3.6.3 日志管理

对于系统中数据库的操作，建立操作日志，必要时可返回单次或批量操作以前的状态。

根据实际需要进行该功能设计。

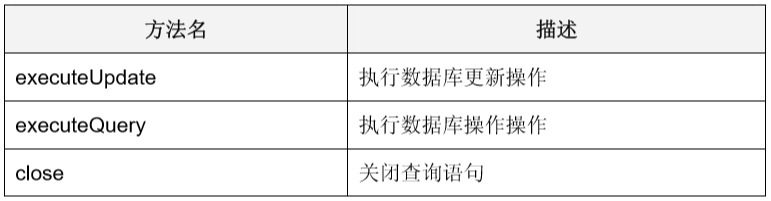
2.4 Interface Description接口描述

在使用JDBC技术进行数据库访问时，主要涉及Connection、PreparedStatement和ResultSet 接口。

1. JDBC Connection对象接口



1. JDBC PreparedStatement对象接口



1. JDBC ResultSet对象接口





3 Database Design 数据库设计

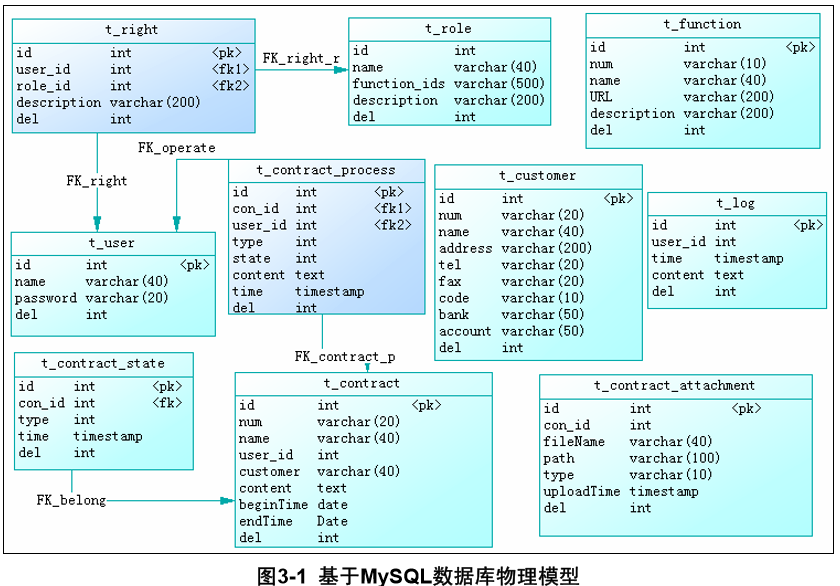
根据合同管理系统的需求和业务流程，系统数据信息可能来自于界面操作，流程处理， 数据存储等过程，对这些过程中涉及到的数据信息进行抽取和分析，进行项目数据库设计。

3.1 概念模型

根据抽样出的数据字典，识别实体，实体的属性和实体间的关系，通过E-R模型进行设 计。

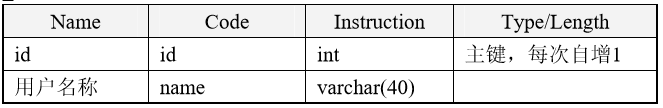
3.2 物理模型

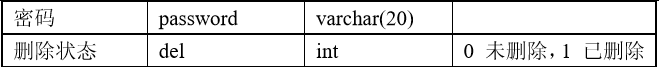
以下是基于MySQL的物理模型。



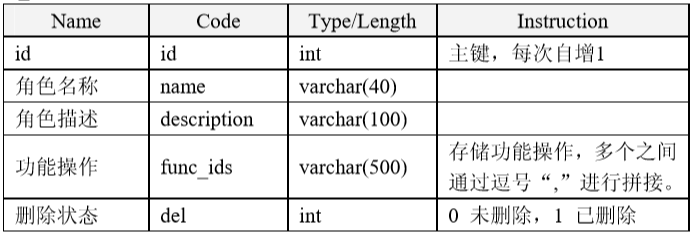
3.3 数据库表设计

1. 用户(t\_user)表

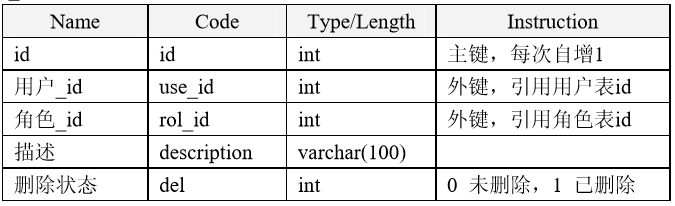




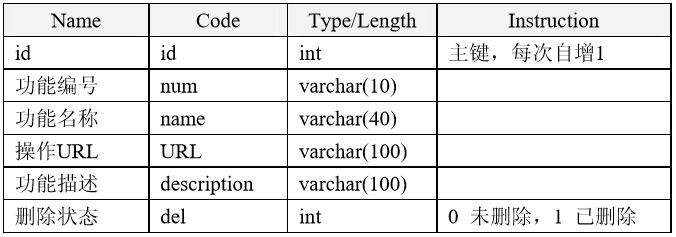
1. 角色(t\_role)表



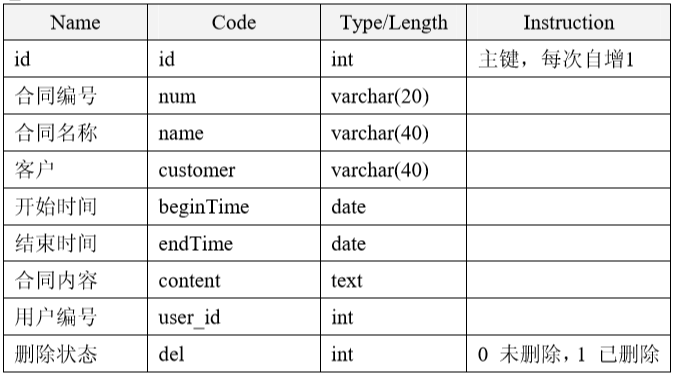
1. 权限(t\_right)表



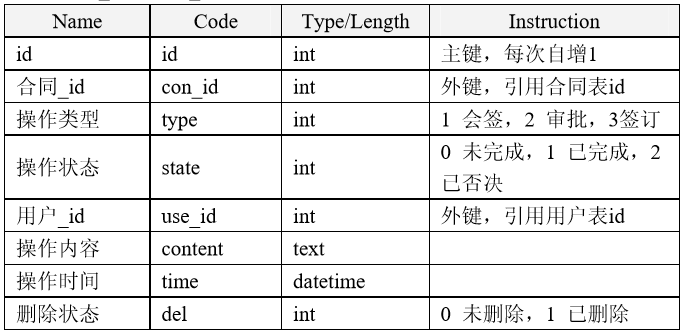
1. 功能(t\_function)表



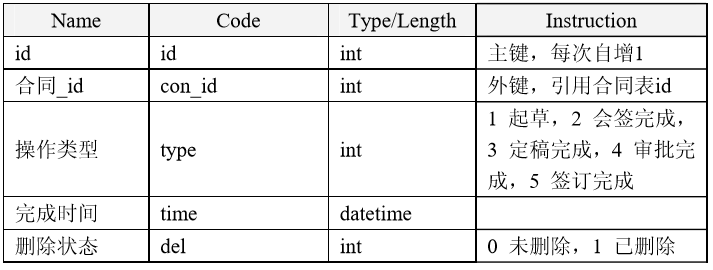
1. 合同(t\_contract)表



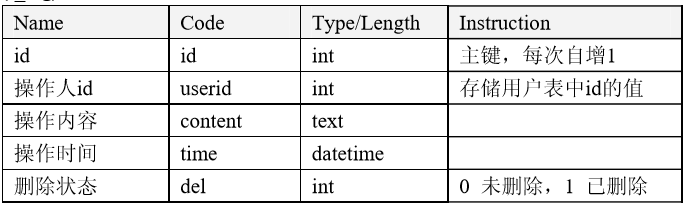
1. 合同操作流程(t\_contract \_process)表



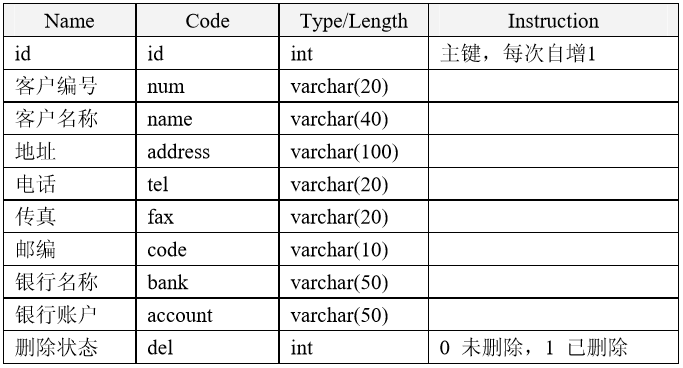
1. 合同操作状态(t\_contract \_state)表



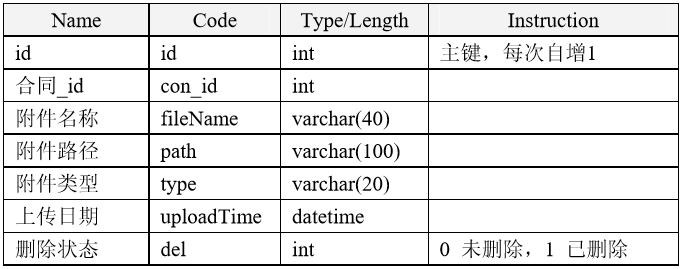
1. 日志(t\_log)表



1. 客户(t\_customer)表

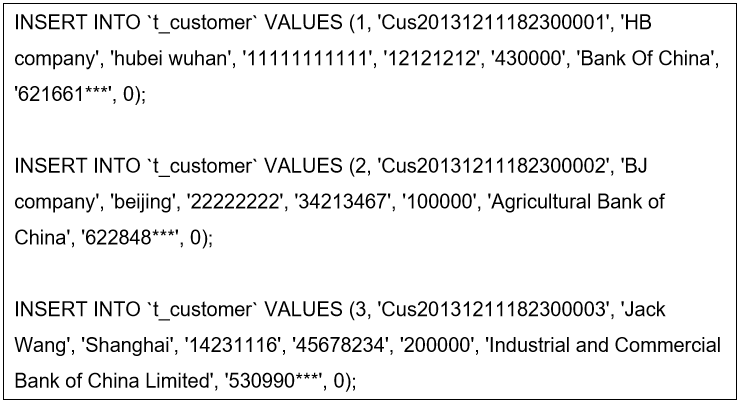


1. 合同附件(t\_contract \_attachment)表

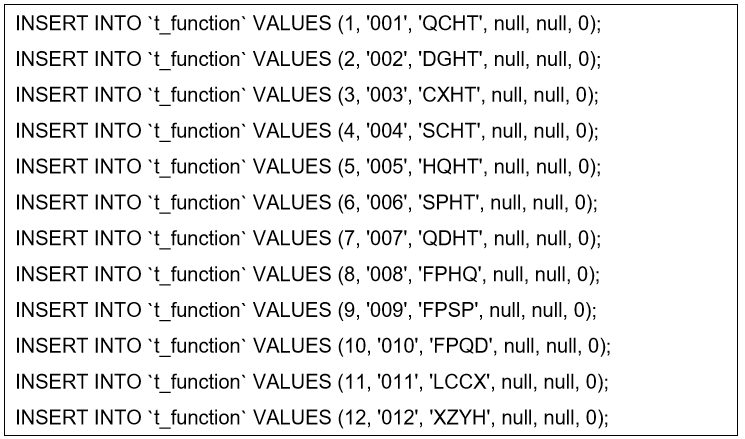


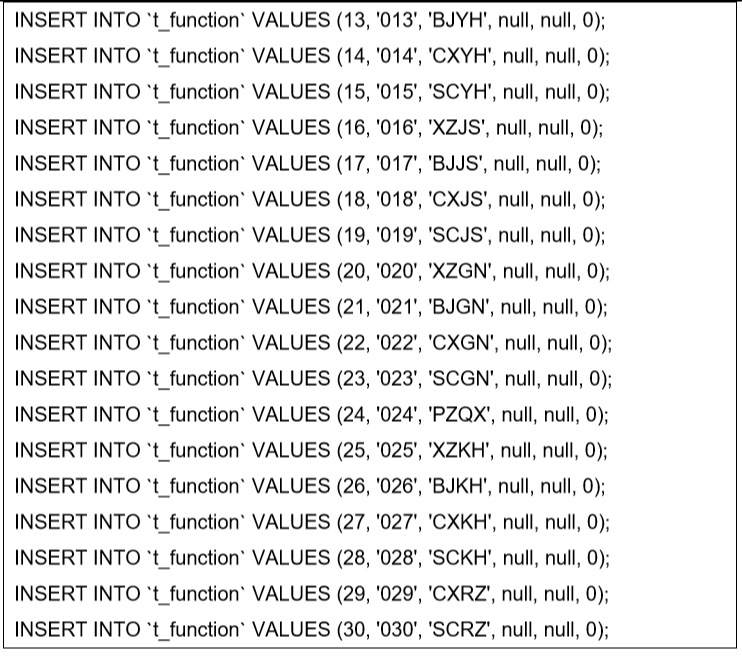
3.4 基础数据配置

初始化客户信息

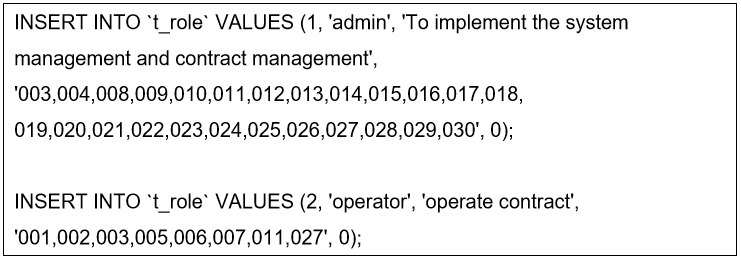


初始化功能信息。

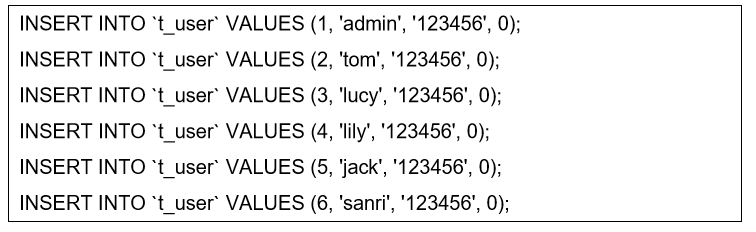




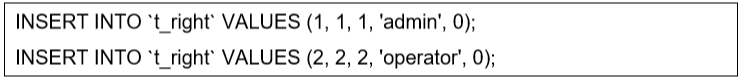
初始化角色信息。

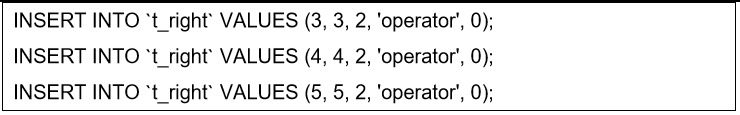


初始化用户信息。



初始化权限信息。





4 UI Design 界面设计

根据需求设计合同管理系统的如下页面：

1. 注册页面和登录页面。
2. 合同管理页面。

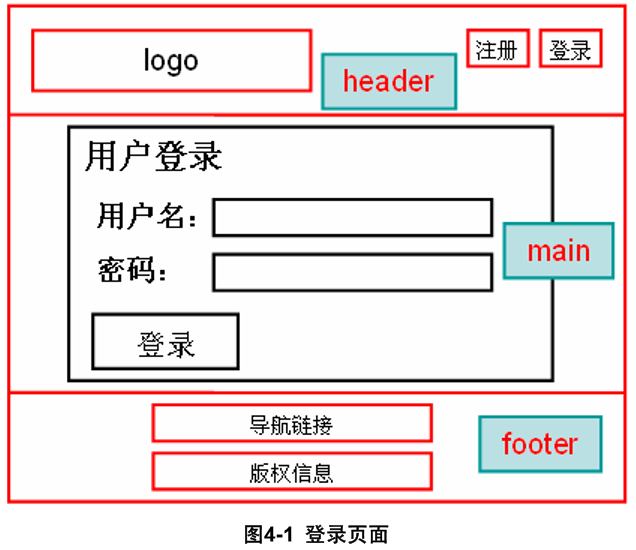
起草，会签，定稿，审批，签订等页面，合同详情页面，合同列表页面，查看会签 意见页面。

1. 系统管理页面。

分配合同页面，权限管理页面，日志管理页面。

4.1 登录页面

1. 界面原图



1. 界面说明 主体部分为相应表单，以接收登录信息。

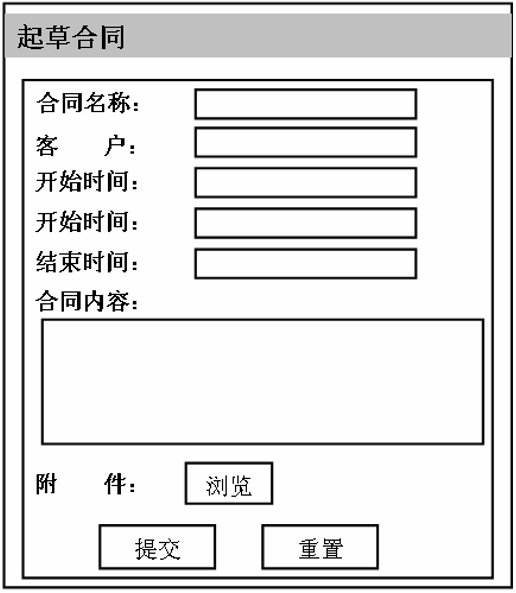
4.2 管理页面

1. 界面原图
2. 界面说明 管理页面采用HTML框架网页Frame进行布局实现。 主体分为左边和右侧。

点击左侧的管理超链接，在右边加载对应功能页面。

4.3 合同起草页面

1. 界面原图



1. 界面说明 主体为合同表单，使用<table>进行布局。

4.4 合同列表页面

1. 界面原图



1. 界面说明 以列表显示合同信息，并能按条件查询合同。

5 Detailed Design 详细设计

5.1 实体类设计

实体类用于层之间传递数据，实体类有一部分与数据库表相对应，有一部分根据实际业 务需要,建立业务实体类。 与数据库表对应的实体类：User，Contract，ConAttach，ConProcess，ConState，Customer， Function，Log，Right和Role。 根据业务需要，建立业务实体类：ConBusiModel，ConDetailBusiModel，CsignatureOpinion 和PermissionBusiModel等。 在设计实体类时，类主要包括三部分内容： (1) 每一个类包括私有成员变量。 (2) getXX()和setXX()方法：为public，用于访问类中私有成员变量(属性)，并分别建立 与私有变量一一对应的getXX()和setXX()方法。 (3) 无参构造方法：用于初始化私有成员变量，给成员变量赋初值。 实体类存放在com.ruanko.model包中。

5.1.1 Contract

5.1.1.1 简介

合同实体类：用于建立和数据表（合同表）之间的映射关系，声明私有属性和数据库表 中的字段相对应，提供初始化构造方法，setter和getter方法供外部调用，在程序中实体类被 用作数据传递的载体。

5.1.1.2 类图

5.1.1.3 属性

以上类图和下列表格中，只列出了合同实体类中的部分关键属性。

5.1.1.4 方法

在实体类中会为这些私有属性提供setter和getter方法供外部调用，而不通过属性名称直 接调用。通过setter方法给实体的属性赋值，通过getter方法取实体的属性值。下面我们以合 同的id为例来进行说明，其他属性同理。 1、setId() (1) 方法描述

(2) 实现描述 现在给合同实体类的id赋值，假设contract为其一个实例：contract.setId(1)表示将合同的 id属性设置为1。

1. getId() (1) 方法描述
2. (2) 实现描述 现在取出合同实体类的id的值，假设contract为其一个实例：contract.getId()表示取出合 同的id的值。
4. 3、Contract() (1) 方法描述
5. 实现描述 定义一个实体的构造方法，方法名称和类名相同：public Contract()，构造方法中可以根 据需要对属性赋指定值，如下为Contract实体的构造方法。

5.1.2 User

用户实体类：用于建立和数据表（用户表）之间的映射关系，其结构与Contract一致。 属性如下：

5.1.3 Role

角色实体类：用于建立和数据表（角色表）之间的映射关系。属性如下：

5.1.4 Function

功能实体类：用于建立和数据表（功能表）之间的映射关系。属性如下：

5.1.5 Right

权限实体类：用于建立和数据表（权限表）之间的映射关系。属性如下：

5.1.6 ConAttach

附件实体类：用于建立和数据表（附件表）之间的映射关系。属性如下：

5.1.7 ConProcess

合同流程实体类：用于建立和数据表（合同流程表）之间的映射关系。属性如下：

5.1.8 ConState

合同状态实体类：用于建立和数据表（合同状态表）之间的映射关系。属性如下：

5.1.9 Customer

客户实体类：用于建立和数据表（客户表）之间的映射关系。属性如下：

5.1.10 Log

日志实体类：用于建立和数据表（日志表）之间的映射关系。属性如下：

5.1.11 ConBusiModel

合同业务模型类：在程序实现过程中，通常会封装业务实体模型，将不同实体中的相关 属性组装到一起以方便数据的传递和显示。 在本工程中，获取合同名称、合同id、起草时间 等合同信息时，需要查询合同表t\_contract 以获取合同名称、合同id，并级联查询合同状态表t\_contract\_state以获取起草时间，从合同 实体类和合同流程实体类中抽取对应的属性重新组合成业务实体类ConBusiModel，以接收 并传递合同信息。 属性如下：

5.1.12 ConDetailBusiModel

合同详情业务模型类：封装原理同合同业务模型，在查看合同详情时，需要获取合同名 称、合同编号、合同内容、客户、起草人等信息，需要查询合同表t\_contract以获取合同名称、 合同编号、客户、合同内容，并级联查询客户表t\_user以获取起草人名称，从合同实体类和 用户实体类中抽取对应的属性重新组合成业务实体类ConDetailBusiModel，以接收并传递合 同详情。 属性如下：

5.1.13 CSignatureOpinion

会签意见业务模型类：封装原理同合同业务模型，在查看会签意见时，需要获取合同编 号、会签意见和会签人姓名等信息，需要查询合同流程表t\_contract\_process以获取合同编号、 会签意见，并级联查询用户表t\_user以获取会签人名称，从合同流程实体类和用户实体类中 抽取对应的属性重新组合成业务实体类CSignatureOpinion，以接收并传递会签意见。 属性如下：

5.1.14 PermissionBusiModel

权限业务模型类：封装原理同合同业务模型，在用户权限列表信息时，需要获取用户编 号、用户名、角色编号和角色名称等信息，需要查询用户表t\_user以获取用户编号、用户名， 并级联查询角色表t\_role以获取角色编号、角色名称，从用户实体类和角色实体类中抽取对 应的属性重新组合成业务实体类PermissionBusiModel，以接收并传递用户权限信息。 属性如下：

5.1.15 PageModel

5.1.15.1 简介

分页模型：封装了分页查询的基本属性，上一页、下一页、当前页、页大小、总记录数、 总页数和当前页集合，并为属性提供setter和getter方法，供分页查询时使用。

5.1.15.2 类图

5.1.15.3 属性

5.1.15.4 方法

在分页模型中，会有有参数和无参数构造方法，他们都是给属性赋初值的。一些属性的 getter方法会有简单的逻辑运算，如总页数，可以通过总记录数和页大小计算得到；上一页 和下一页，可以通过当前页计算获得。 基本的setter和getter方法同合同实体，这里主要描述分页模型的两种构造方法，总页数、 上一页和下一页的getter方法，以及判断是否有上一页和下一页的业务方法。

1. PageModel(int page, int size, int totalCount, List<T> list)

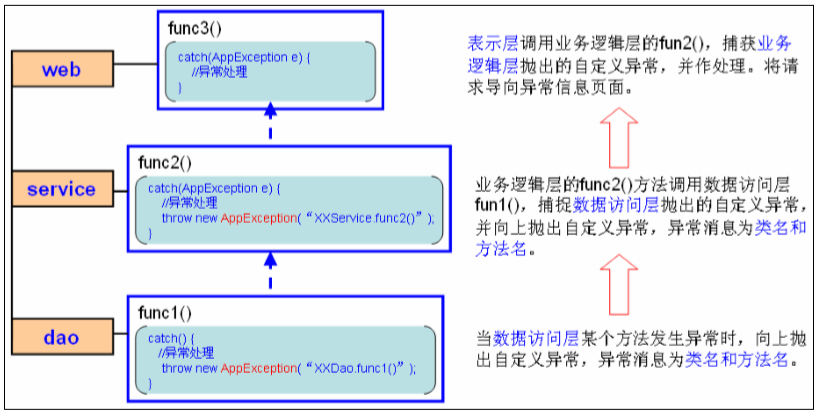
(1) 方法描述

1. 实现描述 定义一个实体的构造方法，方法名称和类名相同：public PageModel()，构造方法中可以 根据需要对属性赋指定值。 定义分页模型的有参构造方法，设置当前页、页大小、总记录数和数据集合，调用对应 属性的setter方法完成设置。
2. PageModel(int page, int size)
3. (1) 方法描述
4. 实现描述 定义分页模型的有参构造方法，设置当前页和页大小，并设置总记录数为0。

6 Error Design 出错处理设计

6.1 概述

为了统一处理表示层、业务逻辑层、数据访问层的异常，定义 AppException 类作为异 常处理的工具类。采用 try，catch，throw 进行异常处理。当底层出现异常，将自定义异常 上抛，并包含异常信息，异常信息格式为：出现异常的类.出现异常的方法名称。直至表示 层统一处理。 异常结构如下图所示：



6.2 自定义异常类

程序中统一的异常类为 AppException，存放在工程的 com.ruanko.utils 包中。 类的定义如下所示：

