基于JavaWeb航天科工财务管理分析系统实训手册

2018

**融创软通科技有限公司**

|  |
| --- |
|  |

**目录**

[1.系统开发要求 1](#_Toc515443085)

[1.1 软件环境： 1](#_Toc515443086)

[1.2 开发环境： 1](#_Toc515443087)

[1.3 数据库设计 2](#_Toc515443088)

[1.4 SpringMVC+MyBatis 框架集成 8](#_Toc515443089)

[2系统功能概述 11](#_Toc515443090)

[2.1登陆功能 11](#_Toc515443091)

[业务逻辑 11](#_Toc515443092)

[页面设计 12](#_Toc515443093)

[后台代码 19](#_Toc515443094)

[2.2登录验证码功能 27](#_Toc515443095)

[业务逻辑 27](#_Toc515443096)

[前台页面 27](#_Toc515443097)

[2.3 Session失效处理 27](#_Toc515443098)

[业务逻辑 27](#_Toc515443099)

[后台代码 27](#_Toc515443100)

[2.4 用户注册功能 28](#_Toc515443101)

[业务逻辑 29](#_Toc515443102)

[页面设计 29](#_Toc515443103)

[后台代码 35](#_Toc515443104)

[2.5数据字典管理 35](#_Toc515443105)

[2.5.1 数据加载 35](#_Toc515443106)

[业务逻辑 35](#_Toc515443107)

[页面设计 36](#_Toc515443108)

[2.5.2添加修改数据字典 49](#_Toc515443109)

[业务逻辑 49](#_Toc515443110)

[页面设计 49](#_Toc515443111)

[后台代码 49](#_Toc515443112)

[2.5.3删除数据字典 50](#_Toc515443113)

[业务逻辑 50](#_Toc515443114)

[页面设计 50](#_Toc515443115)

[后台代码 50](#_Toc515443116)

[2.5.4查询数据字典 51](#_Toc515443117)

[业务逻辑 51](#_Toc515443118)

[页面设计 51](#_Toc515443119)

[后台代码 51](#_Toc515443120)

[2.6个人中心 51](#_Toc515443121)

[业务逻辑 52](#_Toc515443122)

[页面设计 52](#_Toc515443123)

[后台代码 67](#_Toc515443124)

[2.7头像上传 72](#_Toc515443125)

[业务逻辑 72](#_Toc515443126)

[页面设计 72](#_Toc515443127)

[后台代码 72](#_Toc515443128)

[2.8个人密码修改 81](#_Toc515443129)

[业务逻辑 81](#_Toc515443130)

[页面设计 81](#_Toc515443131)

[后台代码 86](#_Toc515443132)

[2.9系统退出功能 87](#_Toc515443133)

[业务逻辑 87](#_Toc515443134)

[页面设计 87](#_Toc515443135)

[后台代码 88](#_Toc515443136)

[2.10系统用户管理 88](#_Toc515443137)

[2.10.1数据加载 88](#_Toc515443138)

[业务逻辑 88](#_Toc515443139)

[页面设计 88](#_Toc515443140)

[后台代码 109](#_Toc515443141)

[2.10.2用户审核功能 118](#_Toc515443142)

[业务逻辑 118](#_Toc515443143)

[页面设计 118](#_Toc515443144)

[后台代码 119](#_Toc515443145)

[2.10.3用户密码重置功能 119](#_Toc515443146)

[业务逻辑 119](#_Toc515443147)

[页面设计 119](#_Toc515443148)

[后台代码 120](#_Toc515443149)

[2.10.4删除用户功能 120](#_Toc515443150)

[业务逻辑 120](#_Toc515443151)

[页面设计 120](#_Toc515443152)

[后台代码 121](#_Toc515443153)

[2.10.5用户查询功能 122](#_Toc515443154)

[业务逻辑 122](#_Toc515443155)

[页面设计 122](#_Toc515443156)

[后台代码 122](#_Toc515443157)

[2.10.6数据列表分页功能 122](#_Toc515443158)

[业务逻辑 122](#_Toc515443159)

[页面设计 123](#_Toc515443160)

[后台代码 123](#_Toc515443161)

[2.11公司信息管理 124](#_Toc515443162)

[2.11.1 数据加载 124](#_Toc515443163)

[业务逻辑 124](#_Toc515443164)

[页面设计 125](#_Toc515443165)

[后台代码 139](#_Toc515443166)

[2.11.2添加公司信息 162](#_Toc515443167)

[业务逻辑 162](#_Toc515443168)

[页面设计 163](#_Toc515443169)

[后台代码 164](#_Toc515443170)

[2.11.3删除公司信息 165](#_Toc515443171)

[业务逻辑 165](#_Toc515443172)

[页面设计 165](#_Toc515443173)

[后台代码 165](#_Toc515443174)

[2.11.4公司信息查询 166](#_Toc515443175)

[业务逻辑 166](#_Toc515443176)

[页面设计 167](#_Toc515443177)

[后台代码 167](#_Toc515443178)

[2.11.5 公司信息修改 168](#_Toc515443179)

[业务逻辑 168](#_Toc515443180)

[页面设计 169](#_Toc515443181)

[后台代码 169](#_Toc515443182)

[2.12公司详细信息管理 170](#_Toc515443183)

[业务逻辑 170](#_Toc515443184)

[页面设计 170](#_Toc515443185)

[后台代码 175](#_Toc515443186)

[2.13公司资产负债表 175](#_Toc515443187)

[2.13.1数据加载 175](#_Toc515443188)

[业务逻辑 175](#_Toc515443189)

[页面设计 176](#_Toc515443190)

[后台代码 181](#_Toc515443191)

[2.14资金借款分析 182](#_Toc515443192)

[2.14.1数据加载 182](#_Toc515443193)

[业务逻辑 182](#_Toc515443194)

[页面设计 183](#_Toc515443195)

[后台代码 193](#_Toc515443196)

[2.14.2借款申请填写 209](#_Toc515443197)

[业务逻辑 209](#_Toc515443198)

[页面设计 210](#_Toc515443199)

[后台代码 210](#_Toc515443200)

[2.15资金放款分析 210](#_Toc515443201)

[业务逻辑 210](#_Toc515443202)

[页面设计 211](#_Toc515443203)

[后台代码 217](#_Toc515443204)

[2.16 借款历史记录 217](#_Toc515443205)

[业务逻辑 217](#_Toc515443206)

[页面设计 217](#_Toc515443207)

[后台代码 227](#_Toc515443208)

[2.16.1 借款历史记录查询 228](#_Toc515443209)

[业务逻辑 228](#_Toc515443210)

[页面设计 228](#_Toc515443211)

[后台代码 229](#_Toc515443212)

[2.16.2 确认借款 229](#_Toc515443213)

[业务逻辑 230](#_Toc515443214)

[页面设计 230](#_Toc515443215)

[后台代码 231](#_Toc515443216)

[2.16.3 债务方还款 232](#_Toc515443217)

[业务逻辑 232](#_Toc515443218)

[页面设计 232](#_Toc515443219)

[后台代码 232](#_Toc515443220)

[2.17 放款历史记录 233](#_Toc515443221)

[2.17.1 数据加载 233](#_Toc515443222)

[业务逻辑 233](#_Toc515443223)

[页面设计 233](#_Toc515443224)

[后台代码 238](#_Toc515443225)

# 1.系统开发要求

## 1.1 软件环境：

Eclipse Java EE IDE for Web Developers.

Version: Neon Release (4.6.0)

JDK 8

Tomcat 8.0

Platform: Windows 10 Enterprise

## 1.2 开发环境：

前端：JQuery V2.1.1

Bootstrap v3.3.7 (<http://getbootstrap.com>)

Gson gson-2.2.4.jar

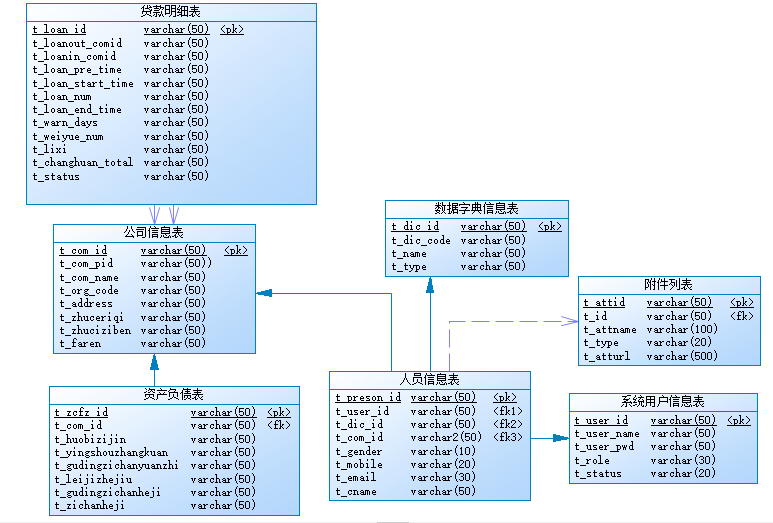
JSTL <http://java.sun.com/jsp/jstl/core>

Jsp

后台: Spring mvc spring-webmvc-4.1.2

MyBastis mybatis-3.1.1mybatis-generator-core-1.3.2

## 1.3 数据库设计



-- ----------------------------

-- Table structure for `t\_attenchment`

-- ----------------------------

DROP TABLE IF EXISTS `t\_attenchment`;

CREATE TABLE `t\_attenchment` (

`t\_attid` varchar(50) NOT NULL,

`t\_id` varchar(50) DEFAULT NULL,

`t\_attname` varchar(100) DEFAULT NULL,

`t\_type` varchar(20) DEFAULT NULL,

`t\_atturl` varchar(500) DEFAULT NULL,

PRIMARY KEY (`t\_attid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **表名** | | t\_attenchment | | **实体名称** |  | | |
| **主键** | | t\_attid | | | | | |
| **估计记录数（条/月）** | | | 0 | **估计数据量（MB/月）** | | | 0 |
| **序号** | **字段名称** | | **字段说明** | **类型** | **位数** | **属性** | **备注** |
| 1 | t\_attid | | 附件主键 | Varchar | 50 |  |  |
| 2 | t\_id | | 附件ID | Varchar | 50 |  |  |
| 3 | t\_attname | | 附件名称 | Varchar | 100 |  |  |
| 4 | t\_type | | 附件类型 | Varchar | 20 |  |  |
| 5 | t\_atturl | | 附件下载URL | Varchar | 500 |  |  |

-- ----------------------------

-- Table structure for `t\_company`

-- ----------------------------

DROP TABLE IF EXISTS `t\_company`;

CREATE TABLE `t\_company` (

`t\_com\_id` varchar(50) NOT NULL,

`t\_com\_pid` varchar(50) DEFAULT NULL,

`t\_com\_name` varchar(50) DEFAULT NULL,

`t\_org\_code` varchar(50) DEFAULT NULL,

`t\_address` varchar(50) DEFAULT NULL,

`t\_zhuceriqi` varchar(50) DEFAULT NULL,

`t\_zhuciziben` varchar(50) DEFAULT NULL,

`t\_faren` varchar(50) DEFAULT NULL,

PRIMARY KEY (`t\_com\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **表名** | | t\_company | | **实体名称** |  | | |
| **主键** | | t\_agency\_id | | | | | |
| **估计记录数（条/月）** | | | 0 | **估计数据量（MB/月）** | | | 0 |
| **序号** | **字段名称** | | **字段说明** | **类型** | **位数** | **属性** | **备注** |
| 1 | t\_com\_id | | 公司ID | Varchar | 50 |  |  |
| 2 | t\_com\_pid | | 公司父ID | Varchar | 50 |  |  |
| 3 | t\_com\_name | | 公司名称 | Varchar | 50 |  |  |
| 4 | t\_org\_code | | 组织机构代码 | Varchar | 50 |  |  |
| 5 | t\_address | | 公司地址 | Varchar | 50 |  |  |
| 6 | t\_zhuceriqi | | 注册日期 | Varchar | 50 |  |  |
| 7 | t\_zhuciziben | | 注册资本 | Varchar | 50 |  |  |
| 8 | t\_faren | | 法人代表 | Varchar | 50 |  |  |

-- ----------------------------

-- Table structure for `t\_dic\_code`

-- ----------------------------

DROP TABLE IF EXISTS `t\_dic\_code`;

CREATE TABLE `t\_dic\_code` (

`t\_dic\_id` varchar(50) NOT NULL,

`t\_dic\_code` varchar(50) DEFAULT NULL,

`t\_name` varchar(50) DEFAULT NULL,

`t\_type` varchar(50) DEFAULT NULL,

PRIMARY KEY (`t\_dic\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **表名** | | t\_dic\_code | | **实体名称** |  | | |
| **主键** | | t\_dic\_id | | | | | |
| **估计记录数（条/月）** | | | 0 | **估计数据量（MB/月）** | | | 0 |
| **序号** | **字段名称** | | **字段说明** | **类型** | **位数** | **属性** | **备注** |
| 1 | t\_dic\_id | | 数据字典ID | Varchar | 50 |  |  |
| 2 | t\_dic\_code | | 数据字典代码 | Varchar | 20 |  |  |
| 3 | t\_name | | 数据字典名称 | Varchar | 50 |  |  |
| 4 | t\_type | | 数据字典类型 | Varchar | 50 |  |  |

-- ----------------------------

-- Table structure for `t\_loan\_details`

-- ----------------------------

DROP TABLE IF EXISTS `t\_loan\_details`;

CREATE TABLE `t\_loan\_details` (

`t\_loan\_id` varchar(50) NOT NULL,

`t\_loanout\_comid` varchar(50) DEFAULT NULL,

`t\_loanin\_comid` varchar(50) DEFAULT NULL,

`t\_loan\_pre\_time` varchar(50) DEFAULT NULL,

`t\_loan\_start\_time` varchar(50) DEFAULT NULL,

`t\_loan\_num` varchar(50) DEFAULT NULL,

`t\_loan\_end\_time` varchar(50) DEFAULT NULL,

`t\_warn\_days` varchar(50) DEFAULT NULL,

`t\_weiyue\_num` varchar(50) DEFAULT NULL,

`t\_lixi` varchar(50) DEFAULT NULL,

`t\_changhuan\_total` varchar(50) DEFAULT NULL,

`t\_status` varchar(50) DEFAULT NULL,

PRIMARY KEY (`t\_loan\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **表名** | | t\_loan\_details | | **实体名称** | |  | | |
| **主键** | | t\_loan\_id | | | | | | |
| **估计记录数（条/月）** | | | 0 | | **估计数据量（MB/月）** | | | 0 |
| **序号** | **字段名称** | | **字段说明** | | **类型** | **位数** | **属性** | **备注** |
| 1 | t\_loan\_id | | 贷款ID | | Varchar | 50 |  |  |
| 2 | t\_loanout\_comid | | 债权方公司ID | | Varchar | 20 |  |  |
| 3 | t\_loanin\_comid | | 债务方公司ID | | Varchar | 50 |  |  |
| 4 | t\_loan\_pre\_time | | 距离还款日天数 | | Varchar | 50 |  |  |
| 5 | t\_loan\_start\_time | | 贷款开始时间 | | Varchar | 50 |  |  |
| 6 | t\_loan\_num | | 贷款总额 | | Varchar | 50 |  |  |
| 7 | t\_loan\_end\_time | | 贷款结束时间 | | Varchar | 50 |  |  |
| 8 | t\_warn\_days | | 提前预警天数 | | Varchar | 50 |  |  |
| 9 | t\_weiyue\_num | | 违约金 | | Varchar | 50 |  |  |
| 10 | t\_lixi | | 贷款利率 | | Varchar | 50 |  |  |
| 11 | t\_changhuan\_total | | 还款总额 | | Varchar | 50 |  |  |
| 12 | t\_status | | 贷款状态 | | Varchar | 50 |  |  |

- ----------------------------

-- Table structure for `t\_person`

-- ----------------------------

DROP TABLE IF EXISTS `t\_person`;

CREATE TABLE `t\_person` (

`t\_preson\_id` varchar(50) NOT NULL,

`t\_user\_id` varchar(50) DEFAULT NULL,

`t\_com\_id` varchar(50) DEFAULT NULL,

`t\_dic\_id` varchar(50) DEFAULT NULL,

`t\_gender` varchar(10) DEFAULT NULL,

`t\_mobile` varchar(20) DEFAULT NULL,

`t\_email` varchar(30) DEFAULT NULL,

`t\_cname` varchar(50) DEFAULT NULL,

PRIMARY KEY (`t\_preson\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **表名** | | t\_person | | **实体名称** | 人员信息表 | | |
| **主键** | | t\_preson\_id | | | | | |
| **估计记录数（条/月）** | | | 0 | **估计数据量（MB/月）** | | | 0 |
| **序号** | **字段名称** | | **字段说明** | **类型** | **位数** | **属性** | **备注** |
| 1 | t\_preson\_id | | 人员表主键 | Varchar | 50 |  |  |
| 2 | t\_user\_id | | 用户表外键 | Varchar | 20 |  |  |
| 3 | t\_com\_id | | 公司表外键 | Varchar | 50 |  |  |
| 4 | t\_dic\_id | | 数据字典ID | Varchar | 50 |  |  |
| 5 | t\_gender | | 性别 | Varchar | 50 |  |  |
| 6 | t\_mobile | | 手机号 | Varchar | 50 |  |  |
| 7 | t\_email | | 邮箱 | Varchar | 50 |  |  |
| 8 | t\_cname | | 中文姓名 | Varchar | 50 |  |  |

-- ----------------------------

-- Table structure for `t\_sys\_user`

-- ----------------------------

DROP TABLE IF EXISTS `t\_sys\_user`;

CREATE TABLE `t\_sys\_user` (

`t\_user\_id` varchar(50) NOT NULL,

`t\_user\_name` varchar(50) DEFAULT NULL,

`t\_user\_pwd` varchar(50) DEFAULT NULL,

`t\_role` varchar(50) DEFAULT NULL,

`t\_status` varchar(20) DEFAULT NULL,

PRIMARY KEY (`t\_user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **表名** | | t\_sys\_user | | **实体名称** |  | | |
| **主键** | | t\_user\_id | | | | | |
| **估计记录数（条/月）** | | | 0 | **估计数据量（MB/月）** | | | 0 |
| **序号** | **字段名称** | | **字段说明** | **类型** | **位数** | **属性** | **备注** |
| 1 | t\_user\_id | | 用户ID | Varchar | 50 |  |  |
| 2 | t\_user\_name | | 登录用户名 | Varchar | 20 |  |  |
| 3 | t\_user\_pwd | | 登录密码 | Varchar | 50 |  |  |
| 4 | t\_role | | 角色 | Varchar | 50 |  |  |
| 5 | t\_status | | 用户状态 | Varchar | 20 |  |  |

-- ----------------------------

-- Table structure for `t\_zichanfuzhai`

-- ----------------------------

DROP TABLE IF EXISTS `t\_zichanfuzhai`;

CREATE TABLE `t\_zichanfuzhai` (

`t\_zcfz\_id` varchar(50) NOT NULL,

`t\_com\_id` varchar(50) DEFAULT NULL,

`t\_huobizijin` varchar(50) DEFAULT NULL,

`t\_yingshouzhangkuan` varchar(50) DEFAULT NULL,

`t\_gudingzichanyuanzhi` varchar(50) DEFAULT NULL,

`t\_leijizhejiu` varchar(50) DEFAULT NULL,

`t\_gudingzichanheji` varchar(50) DEFAULT NULL,

`t\_zichanheji` varchar(50) DEFAULT NULL,

PRIMARY KEY (`t\_zcfz\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **表名** | | t\_zichanfuzhai | | **实体名称** |  | | |
| **主键** | | t\_zcfz\_id | | | | | |
| **估计记录数（条/月）** | | | 0 | **估计数据量（MB/月）** | | | 0 |
| **序号** | **字段名称** | | **字段说明** | **类型** | **位数** | **属性** | **备注** |
| 1 | t\_zcfz\_id | | 资产负债表ID | Varchar | 50 |  |  |
| 2 | t\_com\_id | | 外键公司ID | Varchar | 50 |  |  |
| 3 | t\_huobizijin | | 货币资金 | Varchar | 50 |  |  |
| 4 | t\_yingshouzhangkuan | | 应收账款 | Varchar | 50 |  |  |
| 5 | t\_gudingzichanyuanzhi | | 固定资产原值 | Varchar | 50 |  |  |
| 6 | t\_leijizhejiu | | 累计折旧 | Varchar | 50 |  |  |
| 7 | t\_gudingzichanheji | | 固定资产合计 | Varchar | 50 |  |  |
| 8 | t\_zichanheji | | 资产合计 | Varchar | 50 |  |  |

## 1.4 SpringMVC+MyBatis 框架集成

a. web.xml

<servlet>

<servlet-name>myhos</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>myhos</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

b.myhos-servlet.xml (servlet-name-servlet.xml 必须和 servlet-name 完全一样)

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beansxmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd"*

default-autowire=*"byName"*>

<!-- 扫描 注解 -->

<context:component-scanbase-package=*"com.equip"*/>

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<propertyname=*"prefix"*value=*"/WEB-INF/jsp/"*/>

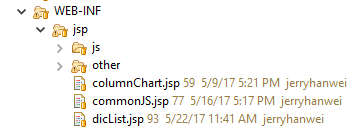
<propertyname=*"suffix"*value=*".jsp"*/>

</bean>

<context:annotation-config/>

<mvc:annotation-driven/>

</beans>



c. applicationContext.xml

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beansxmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd"*>

<context:property-placeholderlocation=*"classpath:/db.properties"*/>

<beanid=*"dataSource"*class=*"com.alibaba.druid.pool.DruidDataSource"*init-method=*"init"*destroy-method=*"close"*>

<propertyname=*"driverClassName"*value=*"${db.driverClassName}"*/>

<propertyname=*"url"*value=*"${db.url}"*/>

<propertyname=*"username"*value=*"${db.username}"*/>

<propertyname=*"password"*value=*"${db.password}"*/>

<propertyname=*"maxActive"*value=*"${db.maxActive}"*/>

<propertyname=*"maxIdle"*value=*"${db.maxIdle}"*/>

<propertyname=*"minIdle"*value=*"${db.minIdle}"*/>

<propertyname=*"maxWait"*value=*"${db.maxWait}"*/>

</bean>

<!-- spring和MyBatis完美整合，不需要mybatis的配置映射文件 -->

<beanid=*"sqlSessionFactory"*class=*"org.mybatis.spring.SqlSessionFactoryBean"*>

<propertyname=*"dataSource"*ref=*"dataSource"*/>

<!-- 自动扫描mapping.xml文件 -->

<propertyname=*"mapperLocations"*value=*"classpath:com/equip/dao/mapping/\*.xml"*></property>

</bean>

<!-- DAO接口所在包名，Spring会自动查找其下的类 -->

<beanclass=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>

<propertyname=*"basePackage"*value=*"com.equip.dao"*/>

<propertyname=*"sqlSessionFactoryBeanName"*value=*"sqlSessionFactory"*></property>

</bean>

<!-- (事务管理)transaction manager, use JtaTransactionManager for global tx -->

<beanid=*"transactionManager"*

class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>

<propertyname=*"dataSource"*ref=*"dataSource"*/>

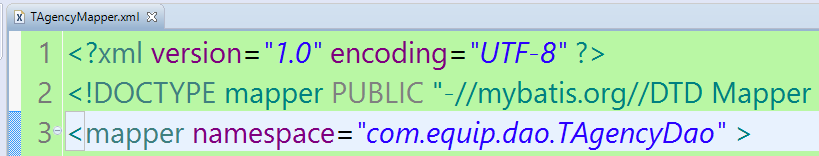
</bean>

<!-- 支持上传文件 -->

<beanid=*"multipartResolver"*class=*"org.springframework.web.multipart.commons.CommonsMultipartResolver"*/>

</beans>

d. 注意Mybatis mapper映射文件的DAO接口的路径

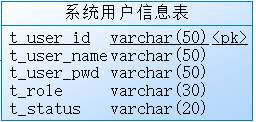


# 2系统功能概述

## 2.1登录功能



### 业务逻辑



用户如果没有账号，那么需要先注册账号，注册信息只有 用户名 密码。用户注册成功之后 等待管理员审核。管理员审核通过之后，用户才可以登录，在个人中心功能维护用户个人信息. 首次登录要求用户必须先完善个人信息。

用户默认状态为(t\_status)：-1，当管理员审核通过时把用户状态设置为：1。所以，登录系统或者查询用户信息必须设置用户状态条件为：1

用户登录成功之后，把用户名密码、用户角色寄存在session里面

### 页面设计

newLogin.jsp

### 后台代码

LoginController.java

## 2.2登录验证码功能

### 业务逻辑

页面加载生成验证码 在当前页面js校验所输入的验证码和生成的验证码是否一致

### 前台页面

参考登录功能页面设计

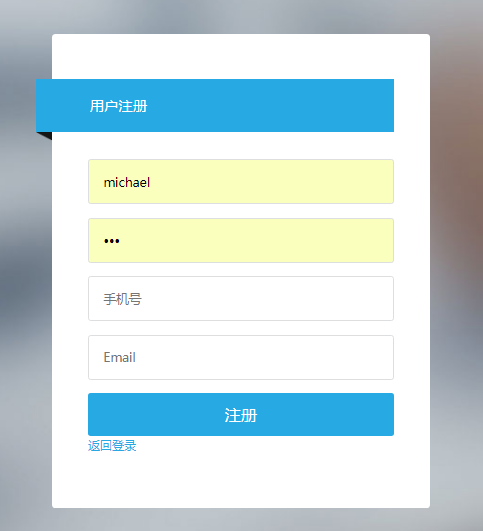
## 2.3 Session失效处理

### 业务逻辑

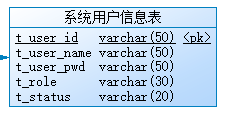
需要在web.xml中 注册过滤器，当session失效的时候 跳转到登录页

### 后台代码

## 2.4 用户注册功能



### 业务逻辑



注册功能，用户名输入时通过Ajax校验用户名是否重复，如果用户名已经重复，提示用户选择其他用户名。用户默认状态为(t\_status)：-1，当管理员审核通过时把用户状态设置为：1

### 页面设计

newReg.jsp

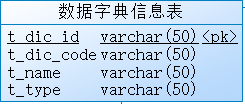
### 后台代码

参考 LoginController.java

## 2.5数据字典管理

### 2.5.1 数据加载

#### 业务逻辑





数据字典表目前提供的数据类型如下：

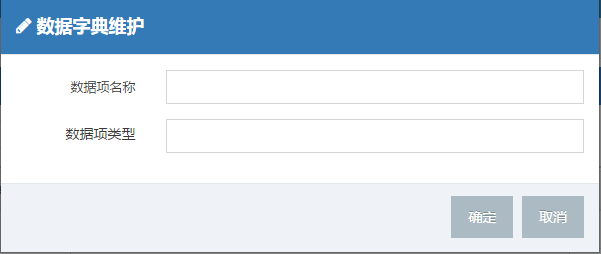
1. 人员性别下拉框数据
2. 用户权限下拉框数据

#### 页面设计

dicList.jsp

### 2.5.2添加修改数据字典

#### 业务逻辑



添加修改功能共用同一个页面，添加数据时，把所填数据之间插入到数据库;修改数据时,点击修改按钮通过某一条数据的id查询数据，然后把所查询到的数据展现在页面。点击确定按钮更新数据。

#### 页面设计

添加数据

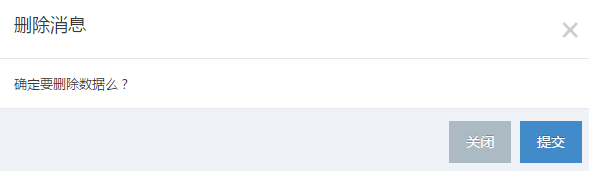
**function** openAddDicWin()

更新数据

**function** openUpdateWin(dic\_ic)

#### 后台代码

### 2.5.3删除数据字典



#### 业务逻辑

点击删除按钮，从页面获取此条数据的id，传递到后台进行删除操作.需要编写删除确认对话框。

#### 页面设计

#### 后台代码

### 2.5.4查询数据字典

#### 业务逻辑



把数据项名称 传递到后台通过sql语句查询

#### 页面设计

参考加载代码

#### 后台代码

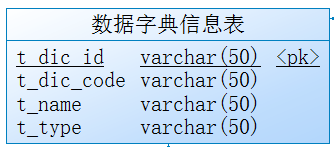
参考加载代码

## 2.6个人中心



### 业务逻辑





个人中心：通过personId 查询用户个人信息。头像信息作为附件信息处理，查询时一起展现在前台页面。数据库附件列表属性”t\_id” 存储每个功能数据库表的主键id 。属性“t\_type”存储功能模块名称,比如“headicon”头像。点击“个人中心”如果查询到数据就展现，如果无数据则为空，然后填充数据，点击提交按钮进行数据更新.



科室信息需要从数据字典表查询

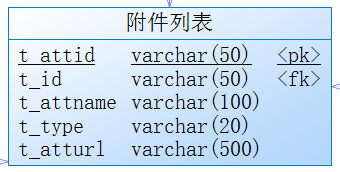
### 页面设计

personInfo.jsp

### 后台代码

TPersonMapper.xml

## 2.7头像上传



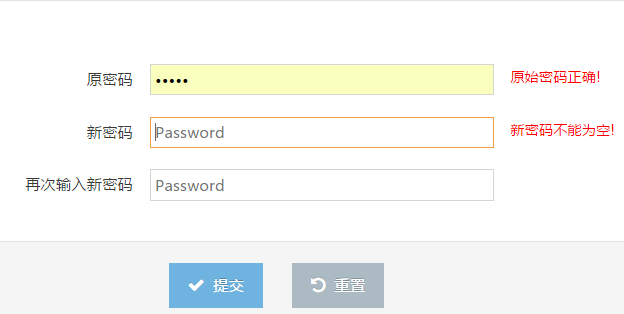
### 业务逻辑

头像上传功能和个人信息更新功能是分开进行的,头像作为附件上传我们只需要personid保存在附件表里面.fileToUpload 此id 页面和js 必须一致 需要引入 ajaxupload插件

### 页面设计

### 后台代码

## 2.8个人密码修改



### 业务逻辑

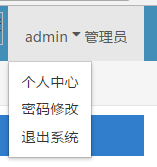
当用户输入原始密码之后，应用ajax去数据库查询校验是否匹配，如果匹配那么输入新密码进行密码修改;如果不匹配 建议联系管理员。

### 页面设计

password.jsp

### 后台代码

## 2.9系统退出功能



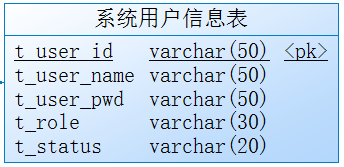
### 业务逻辑

当用户退出系统后，清理session中的用户信息，然后跳转到登录页

### 页面设计

### 后台代码

## 2.10系统用户管理



### 2.10.1数据加载

#### 业务逻辑

页面加载时向后台发起Ajax请求加载所有数据

#### 页面设计

sysUser.jsp

#### 后台代码

TSysUserMapper.xml

sysUserService可查阅其他功能，不再重复赘述

### 2.10.2用户审核功能



#### 业务逻辑

新用户注册之后，数据显示“未审核”,管理员点击之后显示 已审核 表示同意用户使用本系统，然后用户可以登录系统。用户首次登录系统需要强制维护”个人中心”中的用户信息。

当点击删除用户按钮时，把用户的状态置为”-1”，刷新页面用户状态为”未审核”

#### 页面设计

#### 后台代码

### 2.10.3用户密码重置功能

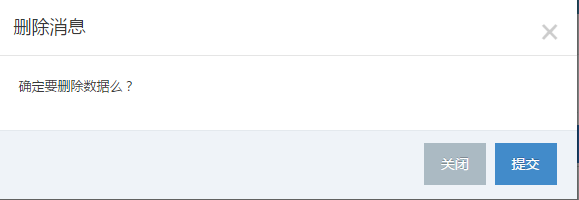
#### 业务逻辑

用户可以自己修改密码，但前期是用户自己记得原始密码。如果用户忘记了原始密码，那么只能通过系统管理员人为重置密码为：8个0

#### 页面设计

#### 后台代码

### 2.10.4删除用户功能



#### 业务逻辑

删除用户 把用户状态设置为 “未审核”从而保留用户历史数据

#### 页面设计

#### 后台代码

### 2.10.5用户查询功能

#### 业务逻辑

把输入的用户名通过Ajax方式传值到后台进行sql查询,把返回的json结果集在页面展现

#### 页面设计

#### 后台代码

详见数据加载代码

### 2.10.6数据列表分页功能

每个数据列表都需要一个分页插件进行数据分页展示，使用JQuery pagination 分页插件



#### 业务逻辑

每个功能跳转到功能首页之前 先在jumpController里面查询到数据总数，保存在request对象里面

@RequestMapping(value="goToDicList", method = RequestMethod.***GET***)

**public** String goToDicList(HttpServletRequest request,HttpServletResponse response) **throws** Exception

{

request.setAttribute("allCodeNum", codeService.selectCountDicCode());

**return**"dicList";

}

然后在页面 通过EL表达式展现数据总数

每次点击 上一页 下一页按钮 都会触发回调函数

**function** PageCallback(index) {

loadPage(index,pageSize);

}

#### 页面设计

#### 后台代码

## 2.11公司信息管理

### 2.11.1 数据加载

#### 业务逻辑



此功能只能由系统管理员来操作，为各个公司开立系统账户。公司信息管理采用树形结构，根节点为“公司管理”,然后分别添加总公司及其全国各地分公司。添加公司时一定要选择此公司的父公司是哪家，这样公司之间才能有一个父子关系。

公司信息维护的展现形式为行状表格，当添加、修改公司信息时弹出二级页面。二级页面左侧为公司树形结构信息，右侧为公司基本信息（不包括详细信息）。

#### 页面设计

comsInfo.jsp

#### 后台代码

1.CompanyController.java

TCompanyMapper.xml

ZiChanFuZhaiService.java

TZiChanFuZhaiDao.java

TZiChanFuZhaiMapper.xml

### 2.11.2添加公司信息



#### 业务逻辑

当添加公司信息时：弹出页面左侧为公司树形结构信息，选择某一公司为父公司，右侧父公司名称显示为当前所选父公司名称。然后填写右侧公司名称。

在页面当中嵌入了 IFrame 引入了 一棵公司信息树形结构



treeID 为了获取元素id，treeText显示所点击节点名称

#### 页面设计

1.参考页面加载 弹出框

2.以下是树形结构的页面代码 companyTree.jsp

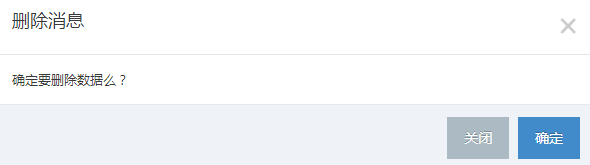
#### 后台代码

### 2.11.3删除公司信息

#### 业务逻辑

从页面获取某公司id，传递到后台执行删除操作。但是，如果被删除公司下面有子公司，那么不能删除。

#### 页面设计



#### 后台代码

### 2.11.4公司信息查询



#### 业务逻辑

把页面输入的公司名称通过ajax参数传递到后台，进行查询，返回的结果重新加载loadpage()函数

#### 页面设计

#### 后台代码

### 2.11.5 公司信息修改



### 业务逻辑

通过公司ID修改公司信息，主要是修改本公司名称以及本公司属于哪家父公司。

提交更新时要提交本公司ID以及父公司ID。原理就是使用HTML 隐藏元素赋值提交数据。

### 页面设计

### 后台代码

## 2.12公司详细信息管理



### 业务逻辑

当各个公司负责人登录系统之后，可以维护本公司的详细信息。公司信息的添加和修改使用的同一个页面

### 页面设计

companyInfo.jsp

### 后台代码

## 2.13公司资产负债表

### 2.13.1数据加载



#### 业务逻辑

各个公司的财务总监填写本公司的资产负债表。固定资产合计=固定资产原值 减去 累计折旧。资产合计=货币资金+应收账款+固定资产合计。要求学生自己能通过JavaScript进行自动计算。

#### 页面设计

#### 后台代码

## 2.14资金借款分析

### 2.14.1数据加载







#### 业务逻辑

债务公司负责人使用此功能可以看见所有公司的资金财务情况，并向某债权公司发起借款请求（状态为：债务方已提交），请求之后等待债权方的放款条件（还款预警天数、违约金百分比等）。当债权公司提出放款条件之后（状态为：债权方已审核），债务公司是否同意对方的放款条件，如果同意那么此次借款交易开始生效（状态为：债务方已确认），否则失败。

当债务方还款成功之后（状态为：债务方已还款），债务方货币资金项减少，债权方货币资金项增加。

#### 页面设计

loanInAnalysis.jsp

#### 后台代码

1.LoanDetailsController.java

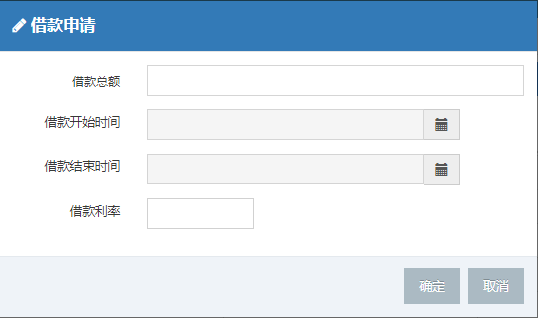
2. ILoanDetailsService.java

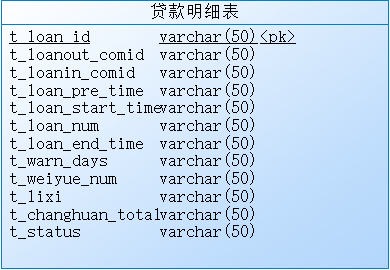
3. LoanDetailsService.java

4. TLoadDetailsDao.java

5. TLoadDetailsMapper.xml

### 2.14.2借款申请填写





#### 业务逻辑

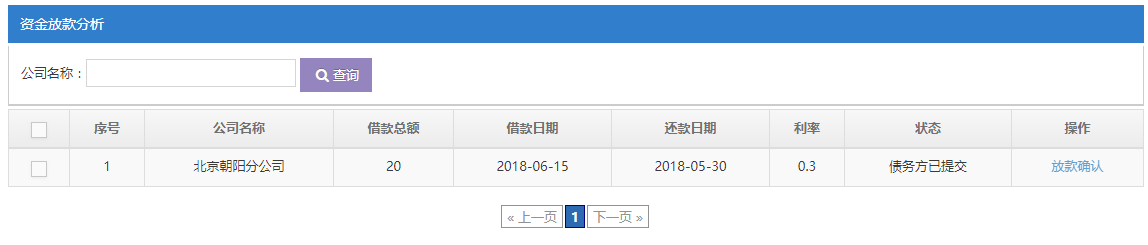
债务方向债权方发起借款请求，声明借款总额、借款还款日期、利率（利率越高、债权方获取利润越多）这是吸引债权方放款的一项重要指标。新增贷款明细数据状态为：债务方已提交

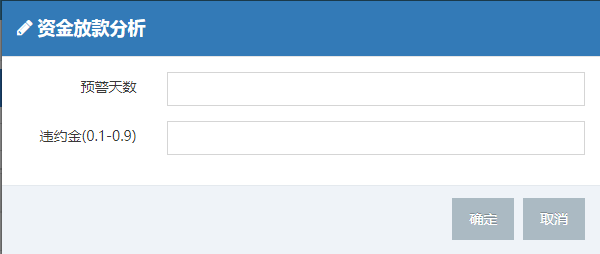
#### 页面设计

#### 后台代码

## 2.15资金放款分析

#### 业务逻辑





债权公司通过此功能同意放款，同时可以向债务方提出违约金赔偿比例，也就是总贷款额度乘以违约金(0.1-0.9)的一次性赔偿。债权方可以填写资金偿还预警天数，比如：提前10天预警，也就是现在时间距离还款日期不能小于等于10天，否则红色字体标注开始预警。

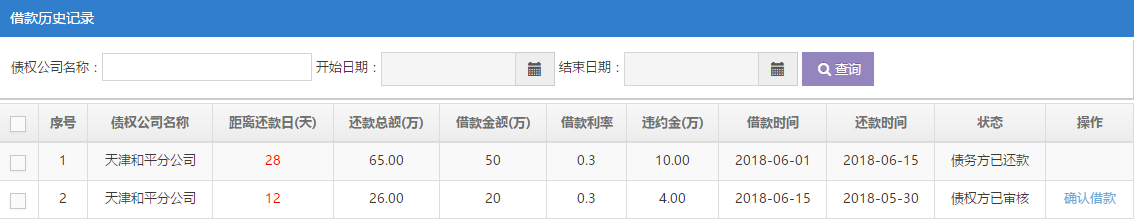


#### 页面设计

loanOutAnalysis.jsp

#### 后台代码

### 2.16 借款历史记录



#### 业务逻辑

当债务方向债权方发起借款之后，可以在此功能看到所发起借款的审批情况。债务方可以确认借款，但是一定要考虑违约金的数额，因为债务方如果没能在还款时间之前还款，那么必须按照 本金+利息+违约金的形式还款给债权方。

此功能后台sql语句自动计算还款总额度，如果按时还款那么还款总额度=本金+利息。如果没能按时还款，那么还款总额度=本金+利息+违约金。

#### 页面设计

loanInHistory.jsp

#### 后台代码

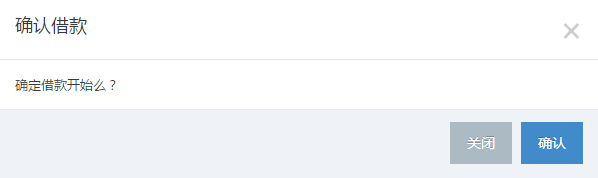
### 2.16.1 借款历史记录查询

#### 业务逻辑

通过公司名称、借款开始日期、结束日期等参数进行查询。

#### 页面设计

### 2.16.2 确认借款



#### 业务逻辑

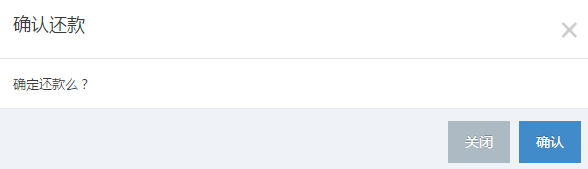
债务方收到债权方的审核信息，如果债务方同意债权方所提出的违约金数额，那么点击“确认借款”。此时，信息状态变为“债务方已确认”，借款流程正式开始。债务方与债权方资金往来需要在资产负债表中的“货币资金”项进行加减。

#### 页面设计

#### 后台代码

### 2.16.3 债务方还款





#### 业务逻辑

如果债务方在规定期限内还款，还款总额=本金+利息;如果逾期还款，那么还款总额=本金+利息+违约金。资金增加或者减少需要更新资产负债表中的“货币资金”一项。

#### 页面设计

loanInHistory.jsp

#### 后台代码

LoanDetailsController.java

## 2.17 放款历史记录

### 2.17.1 数据加载

****

#### 业务逻辑

当债权方填写完“预警天数”、“违约金”之后同意放款，然后就可以在此功能查询借款的进度，主要是查看债务方是否接受我方提出的违约金赔偿金额。

#### 页面设计

loanOutHistory.jsp

#### 后台代码

LoanDetailsController.java

建议上架：考试 培训类

作 者：IsoftStone

定 价：50.00 元