

**企业增值税数据发票系统**

|  |  |
| --- | --- |
| 团队名称 | HaloTeam |
| 指导老师 | 苗巍 |
| 队长 | 李常 |
| 队员1 | 孙晓聪 |
| 队员2 | 牛明鑫 |

目录

[1. 概述 4](#_Toc8783)

[1.1 引言 4](#_Toc7212)

[2. 方案论证 5](#_Toc29805)

[2.1 开发方案 5](#_Toc30607)

[2.1.1 前端技术 5](#_Toc29224)

[2.1.2 后台技术 5](#_Toc16213)

[2.1.3 开发工具 6](#_Toc1559)

[2.2 可行性分析 6](#_Toc14179)

[2.2.1 技术可行性 6](#_Toc8563)

[2.2.2 操作可行性 7](#_Toc32087)

[2.2.3 经济可行性 7](#_Toc8098)

[2.2.4 法律可行性分析 7](#_Toc19021)

[3. 需求分析 8](#_Toc15189)

[3.1 系统描述 8](#_Toc25623)

[3.1.1 需求描述 8](#_Toc18799)

[3.1.2 业务流程 8](#_Toc16634)

[3.2 系统用例 11](#_Toc20559)

[3.3 E-R图 12](#_Toc14945)

[3.4 功能性需求分析 13](#_Toc1582)

[3.4.1 题目需求 13](#_Toc20011)

[3.4.2 系统配置文件依赖图 13](#_Toc22453)

[3.4.3 时序图 13](#_Toc11143)

[3.4.4 类图 15](#_Toc24564)

[3.4.5 活动图 16](#_Toc28529)

[3.5 非功能性需求分析 16](#_Toc17761)

[4. 概要设计 18](#_Toc22091)

[4.1 系统架构 18](#_Toc19341)

[4.2 逻辑架构 18](#_Toc4916)

[4.3 部署结构 19](#_Toc10152)

[4.4 功能架构 20](#_Toc17151)

[4.5 数据库表结构 20](#_Toc22035)

[5. 详细设计及实现 23](#_Toc24570)

[5.1 数据管理模块 23](#_Toc24973)

[5.1.1 数据导入功能 23](#_Toc1259)

[5.1.2 数据操作功能 24](#_Toc20671)

[5.2 数据分析模块 26](#_Toc230)

[5.2.1 月度统计功能 26](#_Toc21813)

[5.2.2 年度统计功能 27](#_Toc2799)

[5.2.3 分类统计功能 27](#_Toc28416)

[5.2.4 比率统计功能 29](#_Toc7039)

[5.3 趋势预测模块 29](#_Toc4763)

[5.3.1 线性预测功能 29](#_Toc25450)

[5.4 员工管理模块 31](#_Toc2762)

[5.4.1 查看员工功能 31](#_Toc21172)

[5.4.2 员工管理功能 31](#_Toc24197)

[5.5 个人中心模块 32](#_Toc30554)

[5.5.1 个人资料功能 32](#_Toc817)

[5.5.2 修改信息功能 33](#_Toc31516)

[5.5.3 安全中心功能 35](#_Toc9655)

[5.6 消息管理模块 35](#_Toc196)

[5.6.1 我的消息功能 35](#_Toc26934)

[5.6.2 发送消息功能 36](#_Toc25730)

[5.7 系统安全模块 37](#_Toc12390)

[5.7.1 系统日志功能 37](#_Toc16470)

[6. 测试 39](#_Toc28385)

[6.1 黑盒测试 39](#_Toc23712)

[6.1.1 登录测试 39](#_Toc10735)

[6.1.2 手机、邮箱验证和重置密码 40](#_Toc13899)

[6.1.3 数据管理验证 41](#_Toc19600)

[6.1.4 数据分析 41](#_Toc20267)

[6.1.5 趋势预测 42](#_Toc25231)

[6.1.6 员工管理 42](#_Toc4238)

[6.1.7 个人中心 43](#_Toc10000)

[6.1.8 消息管理 44](#_Toc25272)

[6.1.9 系统安全 44](#_Toc20515)

[6.1.10 关于 45](#_Toc3303)

[6.2 白盒测试 45](#_Toc21758)

[6.2.1 数据访问层单元测试 45](#_Toc6468)

[6.2.2 业务逻辑层集成测试 47](#_Toc28612)

# 概述

## 1.1 引言

企业增值税发票数据分析系统是由Aisino航天信息有限公司出题，航天信息股份有限公司是集技、工、贸于一体的具有现代化企业管理机制的高新技术企业。由中国航天科工集团公司等十二家中国航天领域的知名企业和科研院所在原航天金穗高技术有限公司、原北京航天金卡电子工程公司和北京航天斯大电子有限公司的基础上于2000年11月1日共同发起成立。

赛题业务场景：描述赛题相关的真实企业业务背景。从真实场景中，适当简化或者提炼出适合比赛的赛题场景。

赛题要求：设计并研发企业数据分析系统，企业数据分析系统主要使用者是企业财务管理人员或者运营管理层，对企业的销项发票数据、进项发票数据进行管理，并根据进销项数据比对，分析企业的经营情况，出具分析报表。

# 方案论证

## 2.1 开发方案

B/S架构的应用也叫Web应用，可以通过浏览器直接访问的应用程序。与传统的C/S架构的应用相比，Web应用有其独特的优势，在企业级应用方面，Web应用能够做到无需线下部署，大量节省成本，同时支持在线演化，跨平台，高并发，服务集群和高可扩展性等。

### 2.1.1 前端技术

Web前端技术是从网页演变而来，HTML是网页的核心，HTML是制作万维网页面的标准语言，他消除了不同计算机信息交流的障碍，CSS用于实现网页的美观，JavaScript实现页面的渲染。因此HTML，CSS，JavaScript是必不可少的，除此之外，也应用到了许多插件，比如：datatables，fontawesome，highcharts等。当前流行的Web前端开发框架有很多，比较新的是HTML5 规范，BootStrap 响应式布局开发框架和JQuery开发库，这些一起提供了一个一套完整的前端流行开发框架，极大地简化了我们的编码。

由于JQuery封装了JavaScript，提供了一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互，同时具有高效灵活的CSS选择器，并且对原生CSS选择器进行扩展，提供短小清晰的多功能接口，并且具有链式语法，提供丰富的插件，支持主流的浏览器，还有对表单的校验上有很大的优势和简单可用行，对于部分数据的渲染和局部刷新，我们不得不采用JQuery的AJAX请求方式，所以JQuery对本项目是非常有用的。

Thymeleaf 的亮点在于他能将后端与前端分离。Thymeleaf 有效的利用了HTML5 规范，将渲染的标签定义为合法的HTML5 标签，使其看起来就是合法的 HTML5 页面，这时我们的前端开发人员就可以只关心如何写HTML5。它和很好的让我们前台开发人员和后端开发人员一起搭配工作。

因为各种浏览器客户端标准不统一，造成了很多的兼容性问题。所以我们采用了HTML5，因为HTML5 增加了很多新的语法特性，同时定义了各种非法文档的规则，使得因为不同浏览器而造成的兼容性问题得到很大的改善。我们也引用了Response.js，Response.js是一个快速、轻量的profill，用于一些不支持CSS3 Media Queries 的浏览器提供媒体查询的min-width和max-width特性，实现响应式网页设计。

### 2.1.2 后台技术

SSM框架是 Spring MVC,Spring 和 MyBatis 的缩写，同时SSM是基于Java的 Web应用后端开发技术，同时也是最流行的Web后端开发技术。Java语言以其优越的特性得到了企业和开发者的一致认可，例如面向对象，跨平台，开放源代码等。所以基于Java语言的SSM框架就成了非常流行的JavaWeb开发框架，其中Spring 可以与很多其他的框架实现无缝集成，已达到丰富应用的工能。

在Spring MVC 中，配置文件有很多备选方案，可以使基于XML的配置文件对Spring MVC 进行配置，也可以是基于Java 的配置文件，当然也可以是两种方案混搭，在本次开发中我们采用xml的配置形式。

基于Spring的一些特性：

1.面向切面（AOP）

2. 控制反转(IOC)

3. 声明式事物管理

4. 丰富的Spring 生态圈

由于以上的特性，我们没有理由不选择它，使用它能给我们带来以下的好处：

1.低耦合，降低了开发的复杂性

2.面向AOP编程，将业务逻辑无关的从业务逻辑剥离

3.提供声明式事务，以便灵活的进行事务管理

4.高可扩展性，在Spring的生态圈中有很多的优秀框架都可以和Spring完成无缝集成，这给应用的可扩展性带来了很大的提高

MyBatis同时支持XML和注解配置(我们采用XML形式)，来将普通的Java对象(POJO)映射为关系数据库中的记录。采用Mybatis的原因：

1.简单灵活，mybatis设计灵活，无第三方依赖。

2.降低程序与Sql的耦合度，在mybatis中是提供接口与程序交互，与具体的Sql解耦。

3.提供编写动态Sql的功能

4.支持映射标签，对象关系标签和对象关系组件维护

### 2.1.3 开发工具

Jdk:1.8

架包管理工具：apache-maven-3.3.9

开发工具：MyEclipse2015

静态资源服务器：nginx

Web服务器：apache-tomcat-9.0.0.M8

建模工具：StarUML

缓存数据库：redis

数据库：达梦数据库免费试用版

## 2.2 可行性分析

### 2.2.1 技术可行性

在开发方案中列举出来了B/S和C/S技术架构，相比之下B/S架构更有优势。同时比较了SSH开发框架和SSM开发框架，SSH更比较适合大型老旧的项目，SSM技术更先进，所以SSM后端开发技术更具优势。相比与原生的前端开发，前端开发框架能够提供更好的技术支持，同时降低开发难度。

以上的技术都是开源的技术，所以不用担心版权授权的问题。同时因为是开放源代码所以可以随时更改源代码已达到自己想要的技术实现。而且以上都是最流行的开发框架，整合了开源社区很多人的贡献，所以能够提供稳定的技术支持，遇到难题可以 很容易找到解决方案。

### 2.2.2 操作可行性

企业增值税发票数据分析系统主要使用者是企业财务管理人员或者运营管理层,以数据分析为主体,另加一些辅助的数据的管理和员工的管理。整个系统具备完整独立的数据流程，系统结构简洁明了，图形简单美观，用户容易使用，操作简单。使用者并不需要拥有太多的计算机知识，也无需处理复杂使用逻辑，即可使用。

### 2.2.3 经济可行性

本项目主要用于比赛，相关技术都是开源免费版本，不涉及商业利益，基本没有经济成本，在经济可行性分析上都是可行的。

### 2.2.4 法律可行性分析

从以上的分析可以看出，本项目主要用于参加中国软件杯第六届比赛，无商业利益。并且采用的相关技术都是开源免费版本，无需涉及授权问题，所以不存在法律不可行问题。

# 需求分析

## 3.1 系统描述

### 3.1.1 需求描述

企业数据分析系统是以企业的进、销项发票信息为基础，对企业的经营数据进行比对分析，有利于企业根据分析报表适时进行税收筹划，调整经营策略。

提供企业12个月增值税发票数据信息，对企业经营情况进行分析。

### 3.1.2 业务流程

系统分为三个角色：管理员、操作员、决策员，不同的角色有着不同的功能操作。由于权限的不同，管理员有最高权限，操作的功能最多，所以我们以管理员的角色进行业务流程的描述。

对管理员业务流程进行说明：所有角色的操作必须在登录状态下，管理员进行登录，如果登录不成功，可以通过电话号码或者邮箱重置密码，但电话号码或邮箱必须进行验证码的校验，校验成功之后便可以设置新的密码，然后重新登录。

登录成功之后管理员可以查看数据图表，数据图表里包含进销项数据的年度统计和月度统计，还可以进行日志记录的查看，这主要是为了保护数据修改的课恢复性。

还可以查看个人信息，个人信息里可以修改个人的相关信息，包括头像等，还可通过邮箱或则手机号来修改自己的登录密码，之后就是数据管理这模块，管理员可以修改进销项数据，也可以删除数据，但删除的时候是假删除，记录信息还存在于数据库，同时消息模块也有相应的记录。之后的就是员工模块，所有的员工都是有管理员分配，管理员可以给已经分配的角色修改它的权限，或者也可以删除员工。

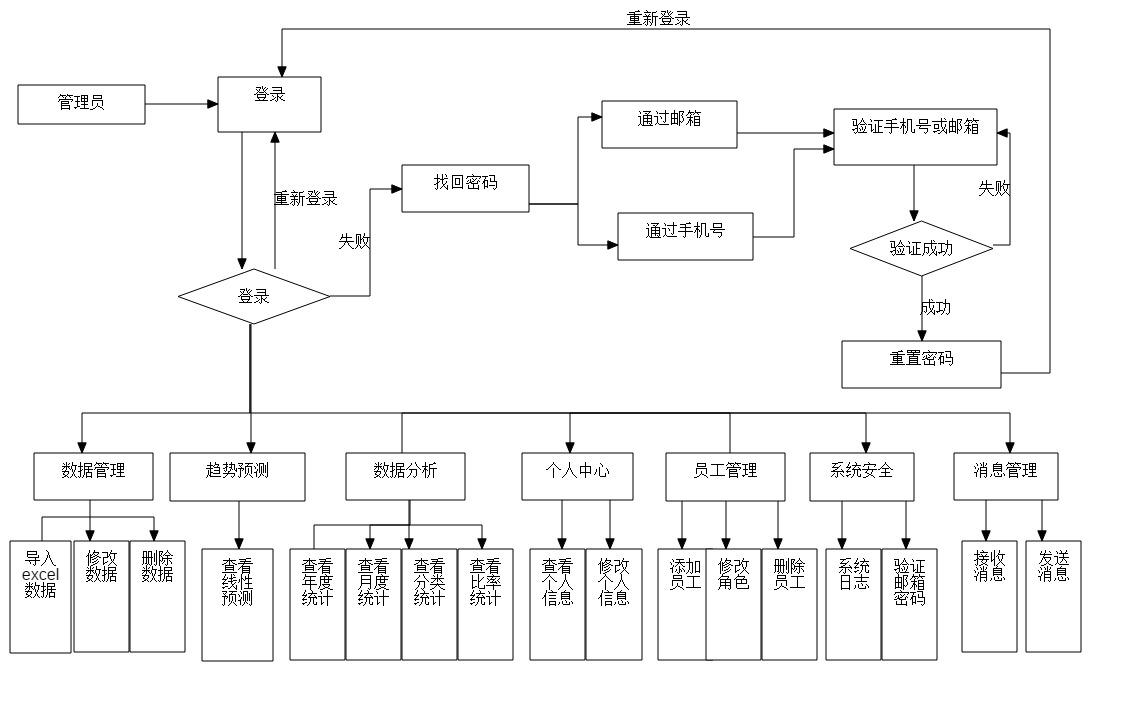


图3-1管理员流程图

对操作员进行业务流程的说明：操作员与管理员相比，权限少了很多，唯一的不同就是操作员可以添加excel进销项数据，其它的安全模块的操作都一样。

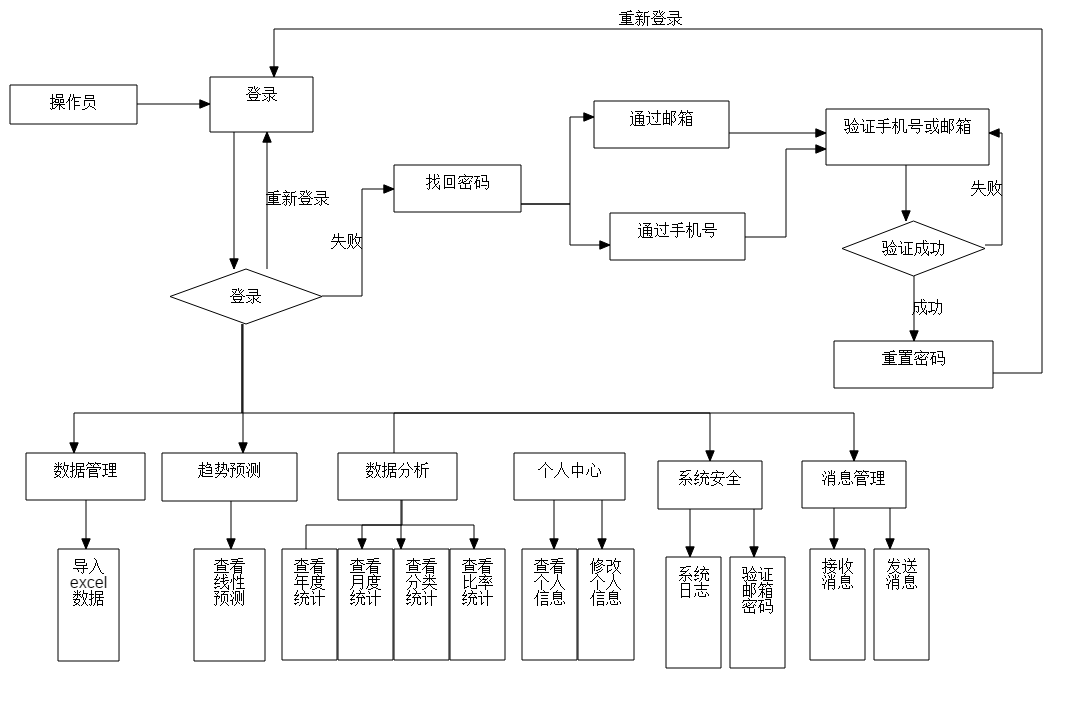


图3-2操作员业务流程图

对决策员业务流程图进行说明：相比之下决策员又少了很多功能，决策员就是来查看的信息的，主要就是数据图表这块。也就是说决策员只能查看所有的数据消息，对数据的操作权限（数据的添加、数据的修改、数据的删除）是没有的。

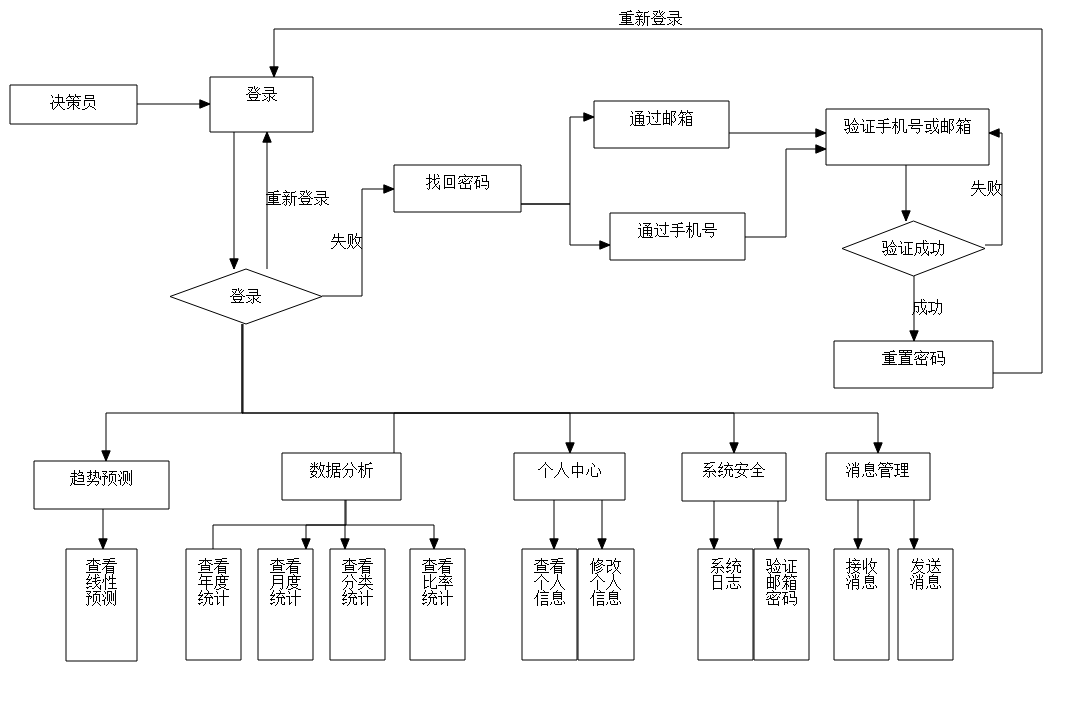


图3-3决策员业务流程图

## 3.2 系统用例

管理员用例

管理员的功能比较多，相比之下它的主要功能是员工管理模块和进销项数据管理模块上，员工的添加、权限的修改、以及员工的删除，进销项数据的修改和删除的操作。包括查看日志消息、查看数据图表（年度统计、月度统计、分类统计、比率统计）、修改个人信息、员工的管理已经数据源的管理。

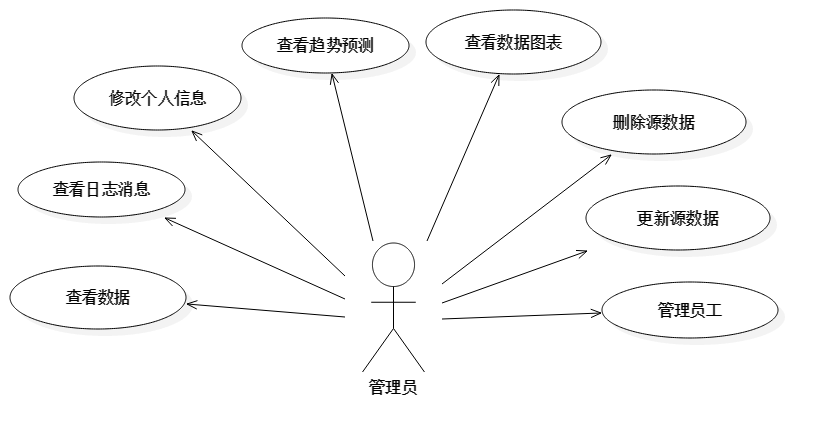


图3-4管理员用例图

决策员

决策员主要就是对数据图的查看，通过进销项数据的年度统计、月度统计以及给出的数据趋势预测进行数据的调整。包括查看数据图表、查看日志消息、修改个人信息等。

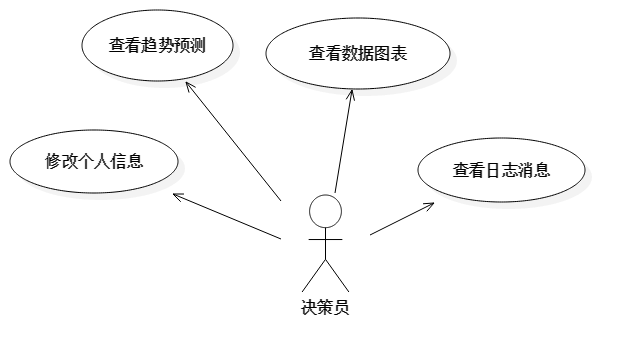


图3-5决策员用例图

操作员

操作员的主要功能为添加进销项excel源数据，也包括查看数据图表、查看数据源、修改个人信息、查看趋势预测等操作。

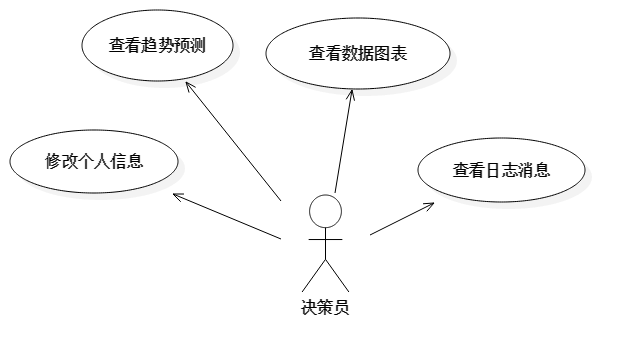


图3-6决策员用例图

## 3.3 E-R图

后台可以由六张表来表示：员工表、角色表、权限表、消息表、数据表、日志表。

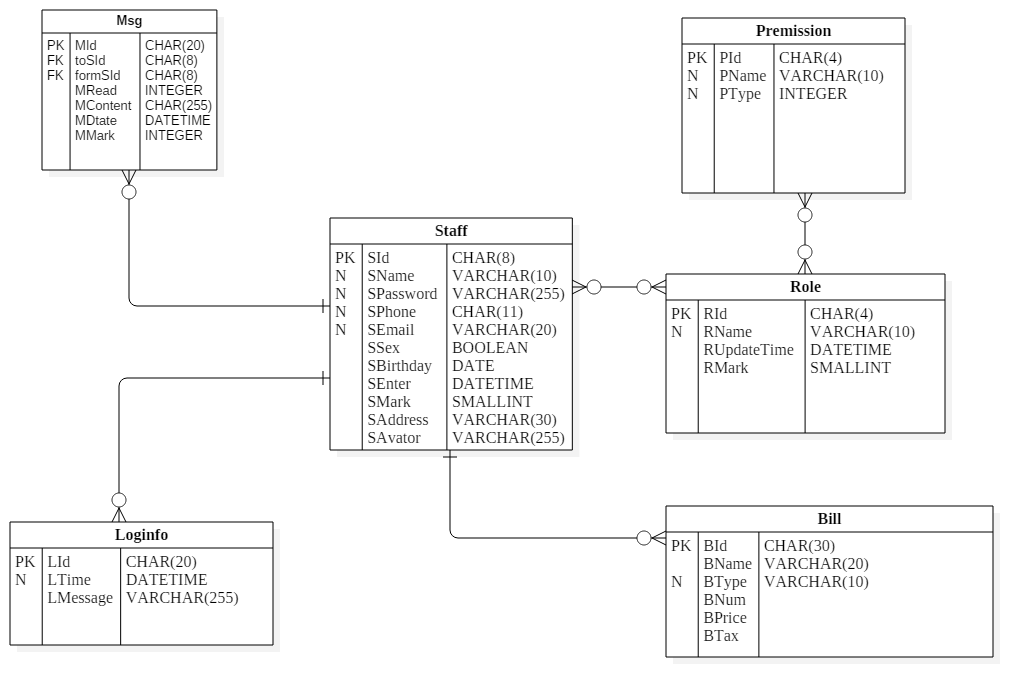


图3-7 e-r图

对表与表的关系做如下说明：

Staff表跟premission表：由于一个员工可以有N个角色，一个角色可以有N个权限，一个权限对应N个角色，所以角色为员工和权限的中间表。

Staff表跟Loginfo表：由于一个员工可以有N操作的日志消息记录，所以员工和日志消息为一对N的关系。

Staff表跟Bill表：一个员工可以操作N个数据，所以员工和数据表为一对N的关系。

Staff表跟Msg表：一个员工可以发多个消息，一个消息属于一个一个员工，所以员工和消息为一对N的关系

## 3.4 功能性需求分析

### 3.4.1 题目需求

根据提供的增值税发票数据（进项数据和销项数据），设计一套数据分析系统，以分析图的形式，对企业的进项数据和销项数据进行月度统计、年度统计，并对进、销项数据进行比对分析，例如，哪月的进项明显高于销项，哪月的销项明显高于进项，以便企业可以根据分析的结果，进行税收筹划和经营状况分析。

### 3.4.2 系统配置文件依赖图

根据功能性要求的需求分析，对整个赛题的整体分析，我们可以得出以下的配置文件的依赖图，系统在技术上采用ssm的框架结构，我们很清晰明了的得出了它所需要的配置文件信息。以及在安全模块上我们采用了shiro权限的技术，在数据缓存上采用redis，所以它们的配置文件我们也可以得出。

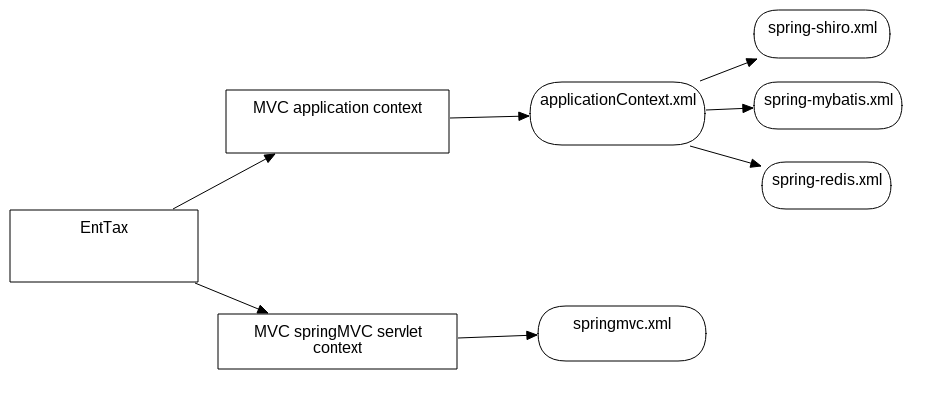


图3-8配置文件依赖图

### 3.4.3 时序图

登录时序图说明：因为系统有shiro权限管理，如果在没有登录的情况下直接访问后台的url则会被直接弹回到登录页面，登录的形式为通过用户提交的用户编号或者手机号或者邮箱加密码，先到后台数据库查看是否有提交的用户编号或者手机号或者邮箱，如果没有直接返回不存在该用户的消息，如果有则验证密码是否正确，用户提交的密码通过shiro配置文件加密后和找到的用户的密码进行匹配，若果相同则登录成功，不同返回登录密码错误消息。

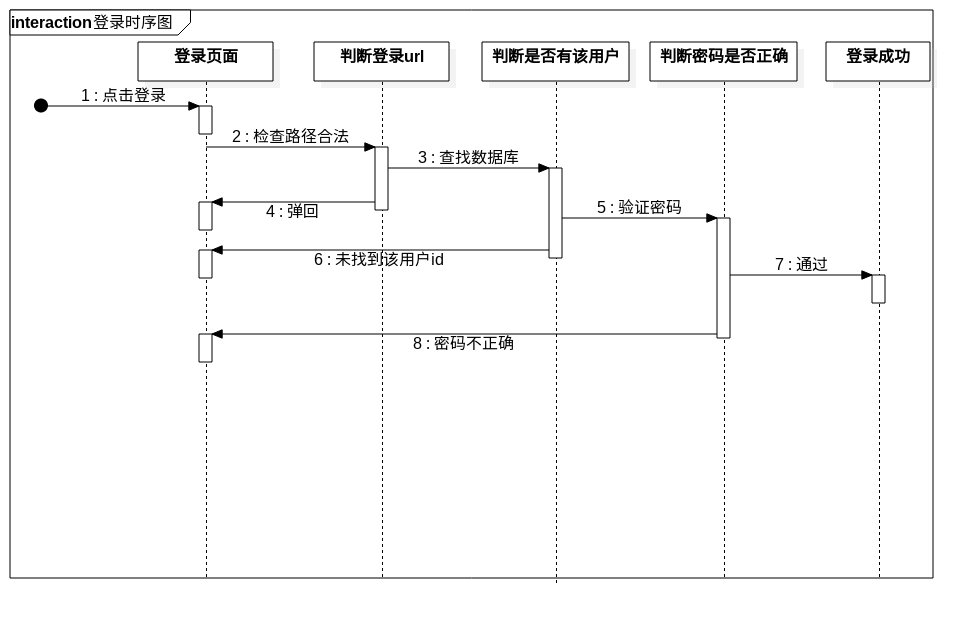


图3-9登录时序图

趋势预测时序图说明：用户点击趋势预测按钮时先去数据库里找到所有的销

数据的项目名称，然后第一次默认选择一个项目名称进行趋势预测，当然用户也可以自己选择想要查看的项目名称，这里主要功能实现为AIServiceImpl里的getLinearLine的方法，通过三层架构的思想，数据通过控制层（AIController）到业务处理层（AIServiceImpl）在数据访问层（BillMapper）再到数据库的传递的一个过程，用户可以根据直接的需求按前2,3,4,5,6个月的选择进行预测。

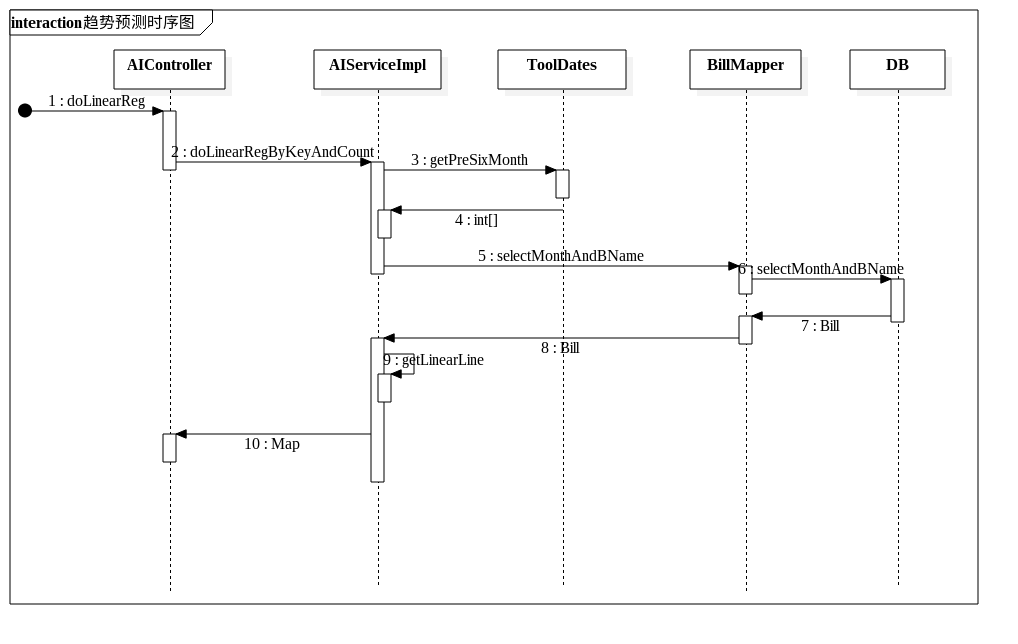


图3-10趋势预测时序图

### 3.4.4 类图

类图之间的关系在整个项目中是非常重要的，下面我们列出了最为常见的几种类图关系，由于篇幅的原因我们没有列出所有的类图关系。通过百度百科上我们能够了解到类图显示的是模型的静态结构，特别是模型中存在类与类直接的调用、类的内部结构，这样更加让我们清晰的了解系统的结构化设计，通常类图泛化、实现、关联、依赖等关系。

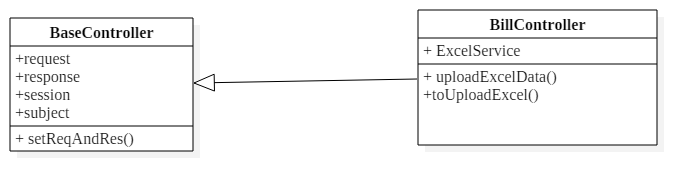


图3-11继承类图



图3-12关联类图

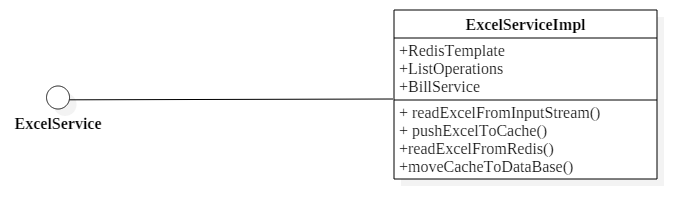


图3-13接口与实现类图

### 3.4.5 活动图

针对管理画的活动图，进行必要的说明，管理员进入系统后，重要的活动就是个人信息模块、消息日志模块、员工管理模块、数据管理模块，在这些地方进行操作，个人信息里可有查看自己的信息，也可对自己的信息进行修改。日志消息就能查看有没有一些最新的消息记录。由于不能自己注册，所以管理员最重要的一个功能是分配一个新的员工，而且对已经分配的员工的权限进行修改。数据模块就是对可能出现错的数据的修改和删除。

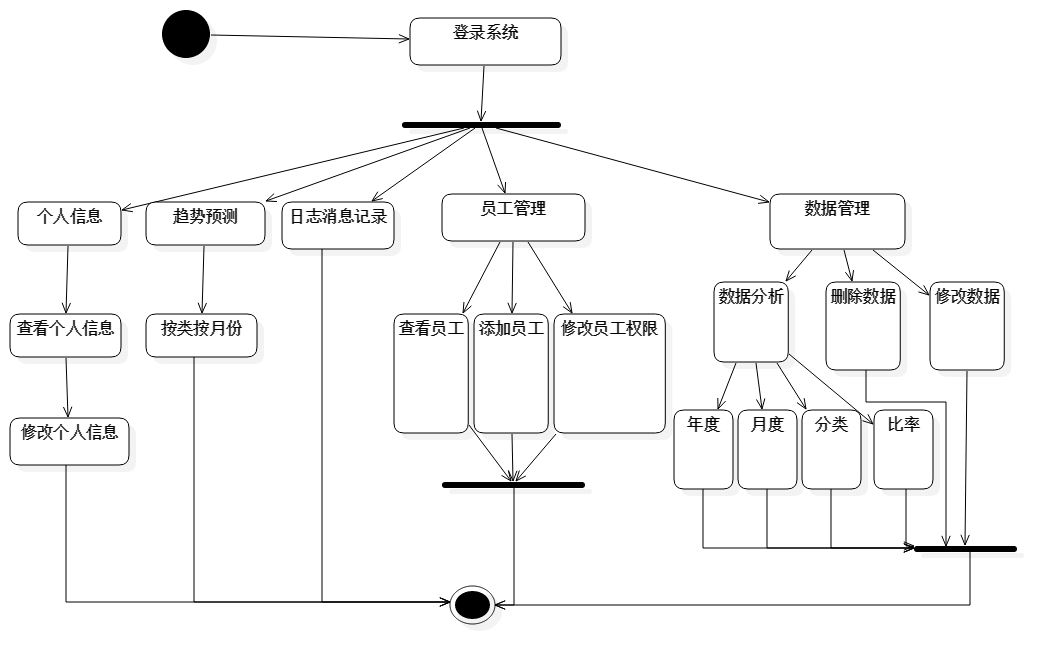


图3-14活动图

## 3.5 非功能性需求分析

安全模块 当用户关掉cookie时，系统生成session id鉴别用户的合法性，当用户在没有登录的情况下通过url路径访问页面是自动被shiro的权限认证弹回到登录页面要求重新登录，当系统的数据被操作时（添加、修改、删除）时，系统对这些操作有记录。

上传数据是数据先存在缓存数据库redis里，如果员工确认无误后上传将缓存数据保存到数据里。

对于员工的登录密码采用MD5动态随机加密，忘记密码后只能通过电话号码或邮箱验证来重置密码，但邮箱和电话号码后台数据库为唯一不可重复。

# 概要设计

## 4.1 系统架构

本系统主要采用B/S框架结构，用户从浏览器里发出HTTP请求,服务器接收相应的请求然后做出处理，将数据通过SQL的方式存入数据库，再从数据库里返回数据或结果到服务器，服务器做出响应，最后返回相应的HTML界面给浏览器或者数据给页面渲染。

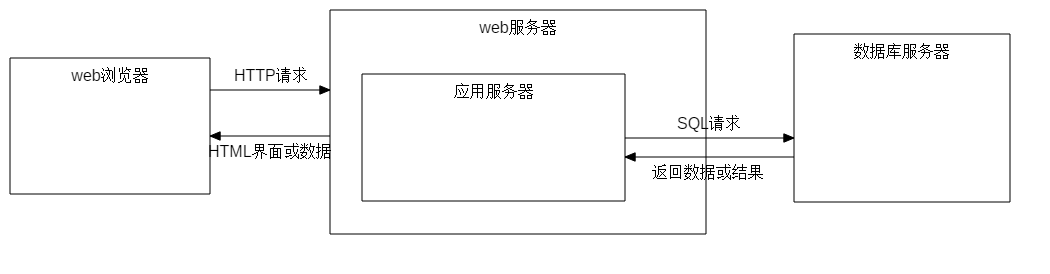


图4-1系统架构图

## 4.2 逻辑架构

系统主要采用三层架构模式，表示层（UI）、业务逻辑处理层（BLL）、数据访问层（DAL）。其中业务实体（Model）和通用类库（Common）贯穿整个项目。



图4-2逻辑架构图

表现层简单的说就是我们的前台界面，用户一眼就能看到的东西。业务逻辑层主要就是针对不同的业务功能进行数据的操作，对数据业务逻辑的处理。数据访问层主要就是通过事物操作数据库，对数据进行增、删、改、查等操作。从而就可以得到一个清晰后台的逻辑结构。系统包图如下:

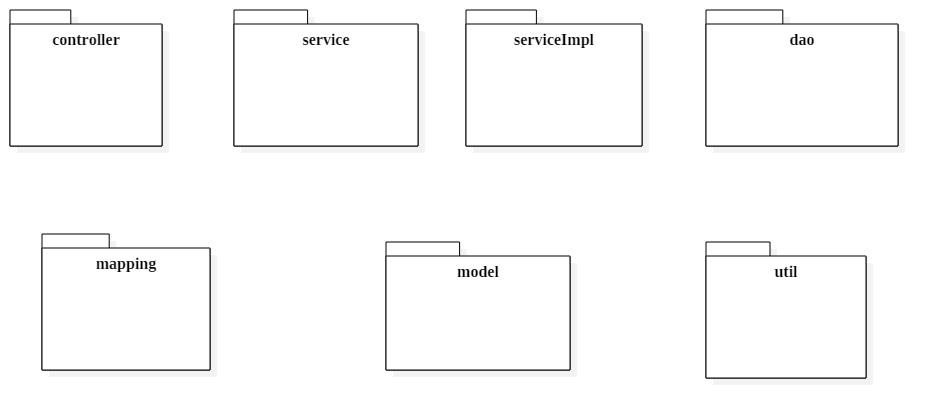


图4-3系统包图

## 4.3 部署结构

考虑到静态资源的访问（图片）部署一个分布式的nginx，可以作为静态资源的服务器，考虑到数据的缓存部署一个redis数据库做缓存用，后台数据存放的地方达梦数据库部署的中间件tomcat服务器。

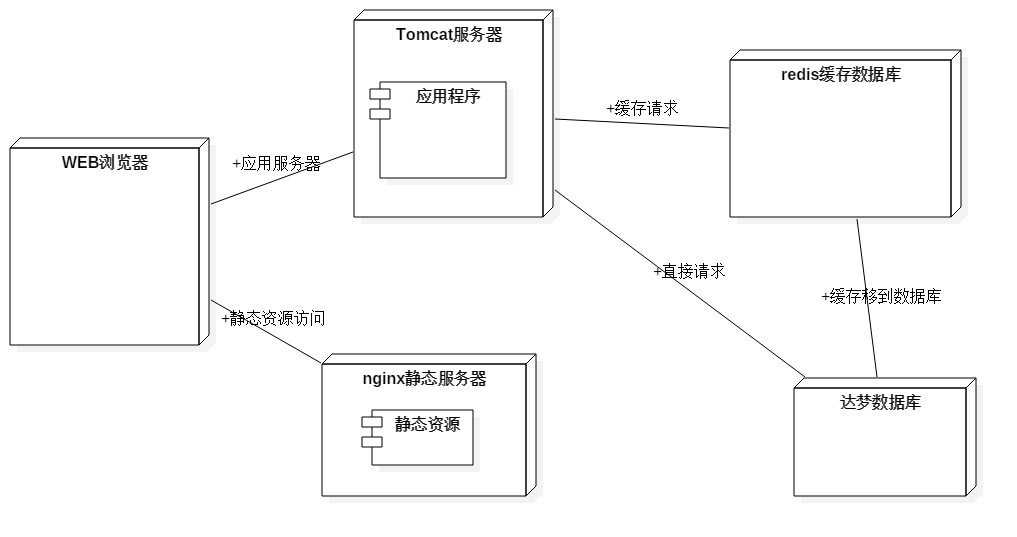


图4-4部署图

## 4.4 功能架构

系统主要分为员工管理模块、数据模块（数据管理、趋势预测、数据分析）、安全模块、个人信息模块、消息管理模块。其中数据模块为核心模块，包括进销项数据通过excel表上传、修改数据、删除数据，线性回归预测为重点之重，数据分析通过年度分析、月度分析、分类统计、比率统计等。

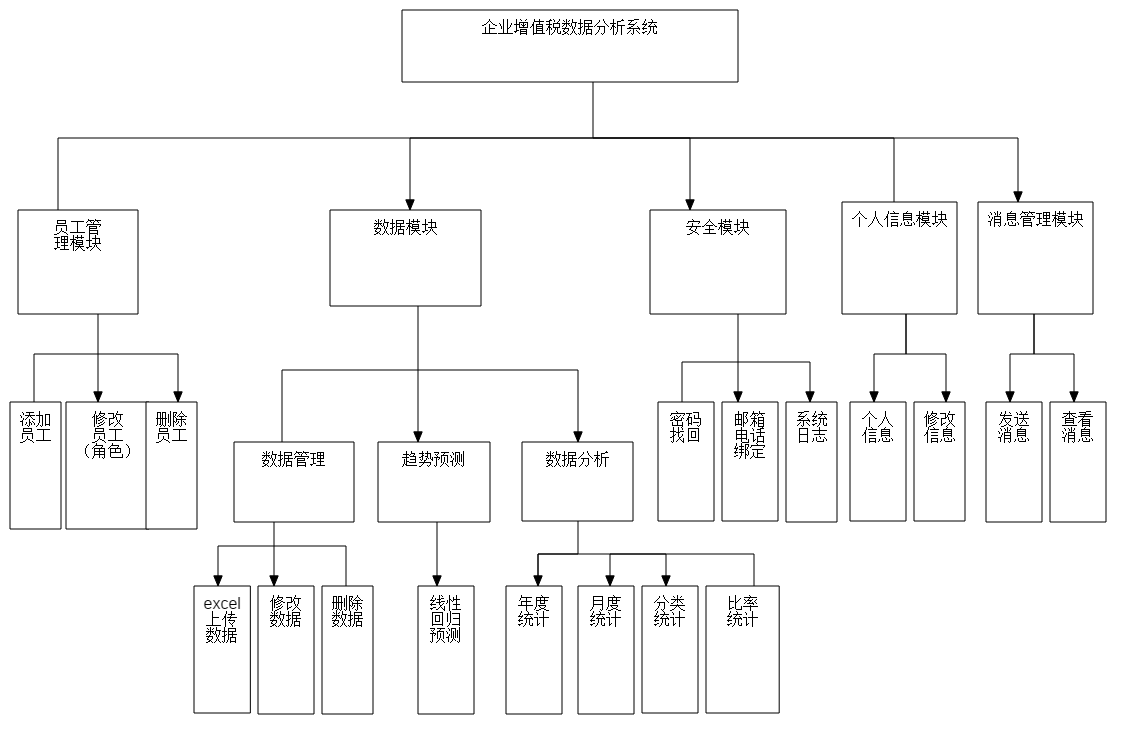


图4-5功能模块图

## 4.5 数据库表结构

基于赛题的要求以及前面的需求分析和功能模块的分析，得出系统的主要表有员工表、角色表、权限表、消息表、数据表、日志表。其中表的各种信息如下表。

图4-6员工表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **字段长度** | **主外键** | **是否为空** | **说明** |
| SId | CHAR | 8 | PK | Not Null | 主键，唯一标识 |
| SName | VARCHAR | 10 |  | Not Null | 用户名 |
| SPassword | VARCHAR | 255 |  | Not Null | 密码 |
| SPhone | CHAR | 11 |  | Not Null | 电话号码 |
| SEmail | VARCHAR | 20 |  | Not Null | 邮箱 |
| SSex | BOOLEAN |  |  |  | 性别 |
| SBirthday | DATE |  |  |  | 生日 |
| SEnter | DATETIME |  |  |  | 入职时间 |
| SMark | SMALLINT |  |  |  | 标记字段 |
| SAddress | VARCHAR | 30 |  |  | 地址 |
| SAvator | VARCHAR | 255 |  |  | 头像地址 |

图4-7角色表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **字段长度** | **主外键** | **是否为空** | **说明** |
| RId | CHAR | 4 | PK | Not Null | 主键，唯一标识 |
| RName | VARCHAR | 10 |  | Not Null | 角色名 |
| RUpdateTie | DATETIME |  |  |  | 更新名称 |
| RMark | SMALLINT |  |  |  | 标记字段 |

图4-8权限表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **字段长度** | **主外键** | **是否为空** | **说明** |
| PId | CHAR | 4 | PK | Not Null | 主键，唯一标识 |
| PName | VARCHAR | 10 |  | Not Null | 权限名称 |
| PType | INTEGER |  |  | Not Null | 类型 |

图4-9数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **字段长度** | **主外键** | **是否为空** | **说明** |
| BId | CHAR | 30 | PK | Not Null | 主键，唯一标识 |
| BName | VARCHAR | 20 |  | Not Null | 项目名称 |
| BType | VARCHAR | 10 |  | Not Null | 数据类型 |
| BNum | DATETIME |  |  |  | 月份 |
| BPrice | DOUBLE |  |  |  | 金额 |
| RMark | SMALLINT |  |  |  | 标记字段 |

图4-10日志表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **字段长度** | **主外键** | **是否为空** | **说明** |
| LId | CHAR | 20 | PK | Not Null | 主键，唯一标识 |
| LTime | DATETIME |  |  |  | 操作时间 |
| LMessage | VARCHAR | 255 |  |  | 消息描述 |
| RMark | SMALLINT |  |  |  | 标记字段 |

图4-11消息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **字段长度** | **主外键** | **是否为空** | **说明** |
| MId | CHAR | 20 | PK | Not Null | 主键，唯一标识 |
| toSId | CHAR | 8 | FK | Not Null | 外键，发送者 |
| formSId | CHAR | 8 | FK | Not Null | 外键，接收者 |
| MRead | INTEGER |  |  |  | 标记是否已读 |
| MContent | VARCHAR | 255 |  |  | 消息内容 |
| MDtate | DATETIME |  |  |  | 发送时间 |
| RMark | SMALLINT |  |  |  | 标记字段 |

# 5. 详细设计及实现

## 5.1 数据管理模块

### 5.1.1 数据导入功能

由于数据导入是通过excel模板形式，模板可以在本系统的首页下载，数据必须按模板的模拟数据的格式填写，导入数据分为两种，一种为销项数据，另一种为进项数据，上传的类型用户可以根据自己的需求选择销项或进项或则两个都选，上传数据选存入redis缓存中，用户可以预览上传的数据，如果觉得上传的数据没有问题，就可以点确认上传，确认上传是数据存入数据库里，并返回上传的结果。

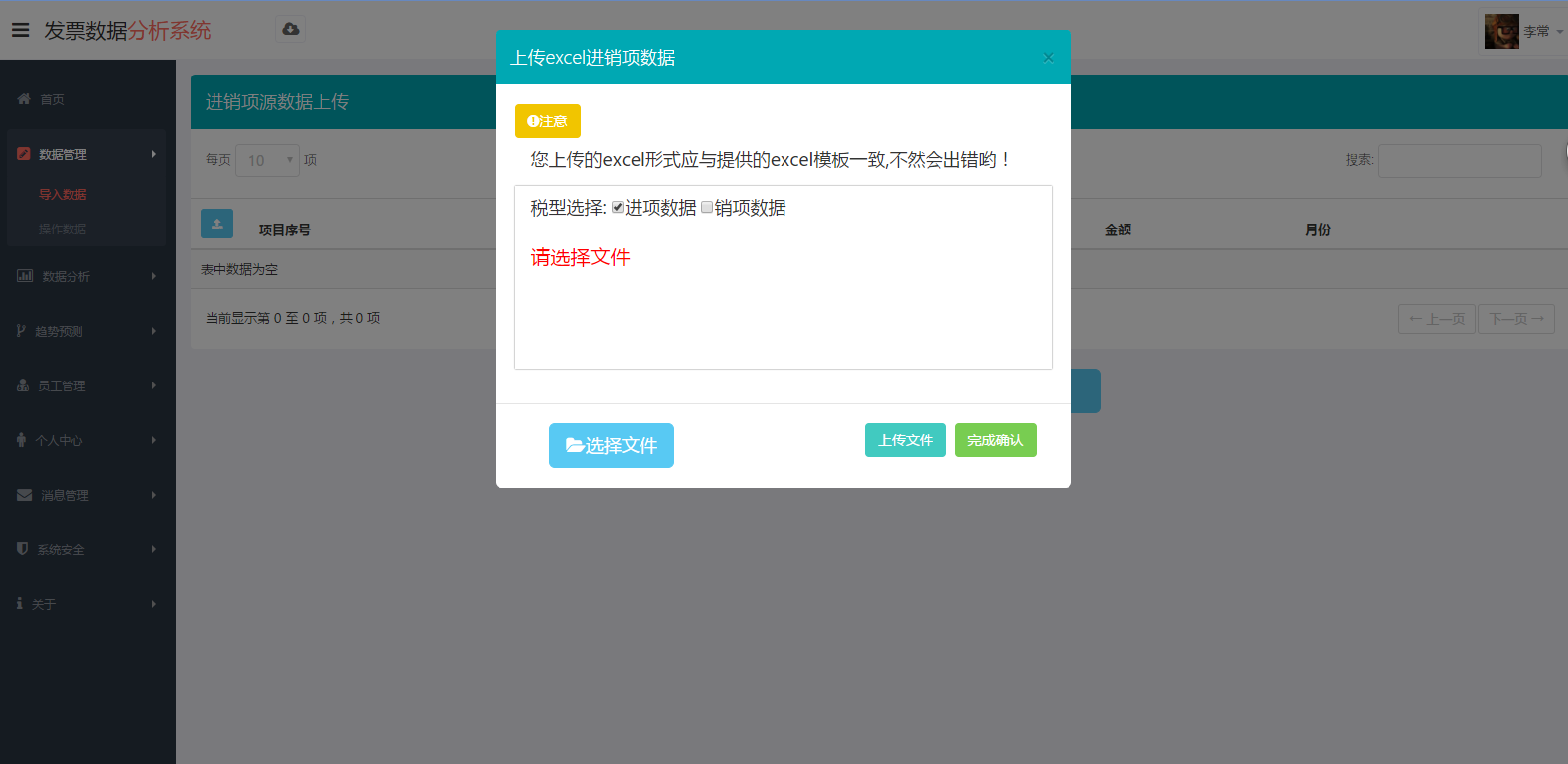


图5-1数据导入图

伪代码实现

//读数据

public List<Bill> readExcelFromInputStream(int bMark, int sheetAt, InputStream is, String extName) throws Exception {

//判断excel的类型（xls、xlsx、或者其他）

if (ConstantStr.EXCELTYPEXLS.equals(extName))

wb = new HSSFWorkbook(is);

Sheet sheet = wb.getSheetAt(sheetAt);

//将数据放入list集合里

//遍历sheet将单元格的数据存入Bill的属性值里

for (int i = 1; i <= numRows; i++)

//当前行的集合

List rowList = new ArrayList();

//获取当前行

Row row = sheet.getRow(i);

//获取当前行的单元格数量

int cellCount = row.getLastCellNum();

//获取元素

Cell cell = row.getCell(j);

//填值

bill.setBType(bMark == 1 ? TAX\_OUT : TAX\_IN);

//为了模拟数据，折算月份成年份

ToolDates.fixDateBaseMonth(bill);

}

//redis生成key值

public long pushExcelToCache(String key, List<Bill> bills) {

return listOps.leftPush(key, bills);

}

//移除redis的key值

public List<Bill> readExcelFromRedis(String key) {

return listOps.rightPopAndLeftPush(key, key);

}

//存入数据库

public int moveCacheToDataBase(String key ,Session session) {

//先从缓存取出

List<Bill> bills = listOps.leftPop(key);

//生成系统日志

logMapper.insert(CommonLog.createLogMessage(message,session);

//把记录值置空

//存入MySql数据库

}

### 5.1.2 数据操作功能

数据操作功能主要有两个操作，对数据的修改和删除，由于数据的项目的编号为唯一主键，是允许修改的，对数据的项目名称、项目类型、金额以及月份是可以修改的，但权限仅限于管理员，删除是根据项目编号进行删除，并且这些操作完成之后在安全模块可以查看系统日志，系统日志里记录了数据的变更记录。

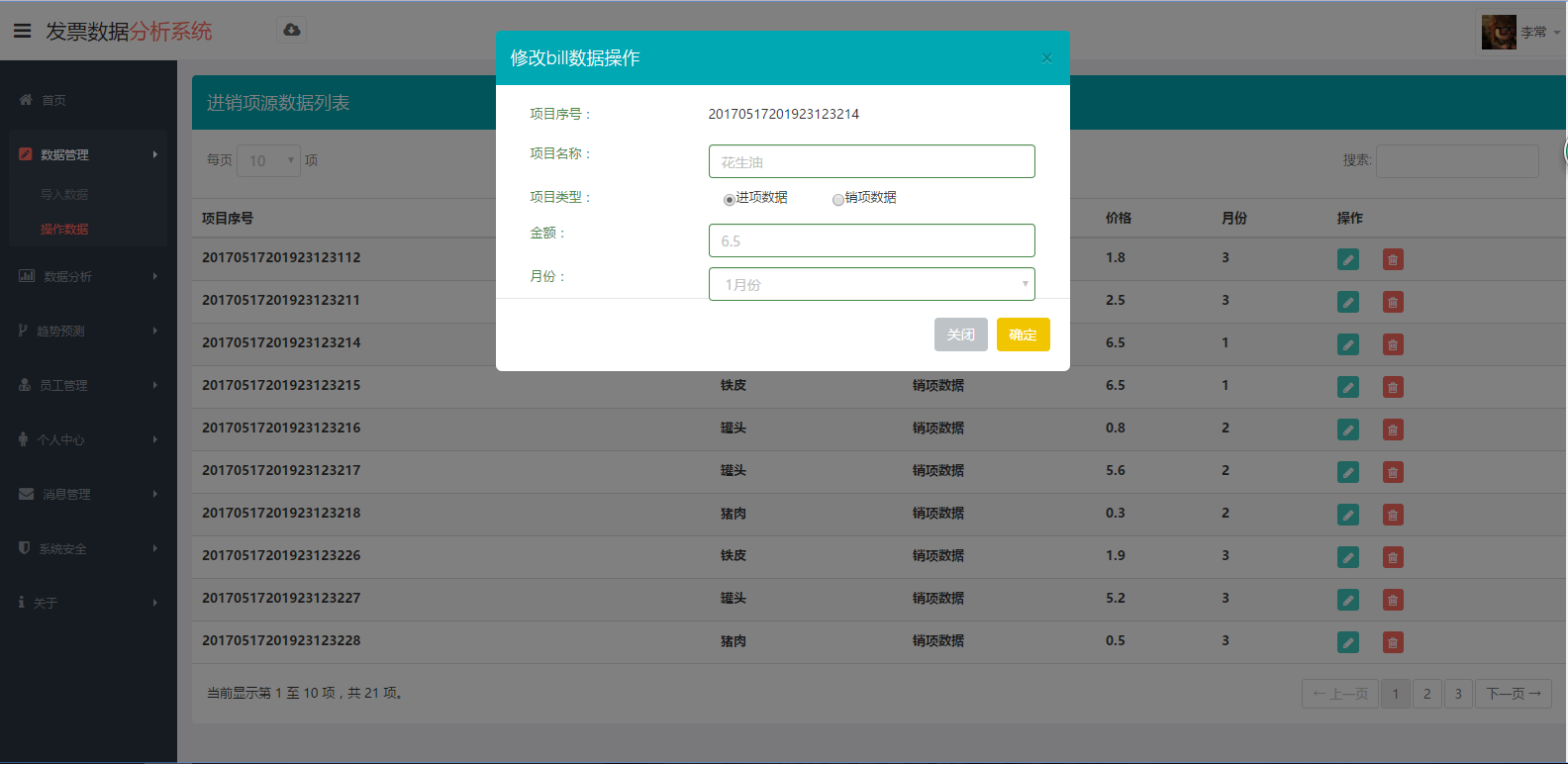


图5-2数据操作图

伪代码实现

//修改数据

public int updateBill(Bill bill,Session session) {

Bill preBill= billMapper.selectByPrimaryKey(bill.getBId());

String message=":将项目编号为"+bill.getBId()+"的";

//对比数据库里的数据的和传入进来的那些不一样

//把不一样的记录在消息里

//生成系统日志

logMapper.insert(CommonLog.createLogMessage(message,session);

bill.setBMark(0);

return billMapper.updateByPrimaryKey(bill);

}

//删除数据

public int deleteBillById(String bId,Session session) {

Bill bill = billMapper.selectByPrimaryKey(bId);

//假删除，将Mark置为-1

bill.setBMark(-1);

//生成系统日志

String message=":将项目编号为"+bId+"的数据删除";

logMapper.insert(CommonLog.createLogMessage(message, session));

return billMapper.updateByPrimaryKey(bill);

}

## 5.2 数据分析模块

### 5.2.1 月度统计功能

月度统计默认为当前时间的年份，也可根据用户自己的需求选择年份，月度统计里主要由每一个月总体的进项数据和销项数据组成，是一个总体上的宏观描述的展示，主要用于决策层的整体上的掌控。

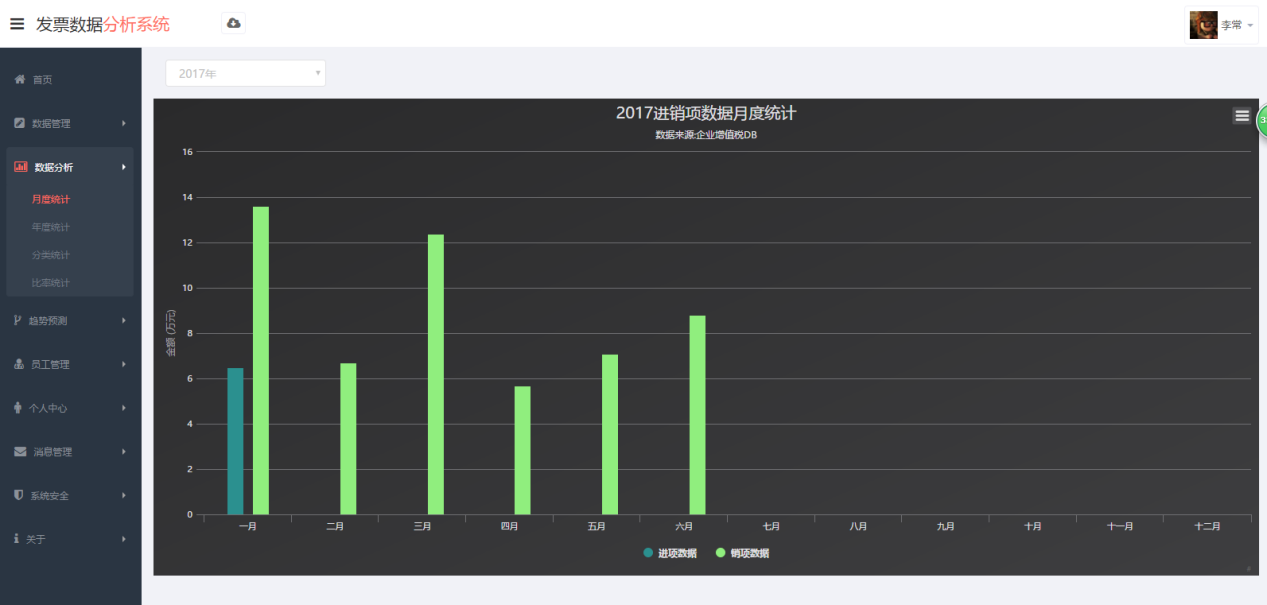


图5-3月度统计图

伪代码实现

public Map showMonthBill(String year) {

List<BillInfo> billInfos = billMapper.selectMonthBill(year);

for (BillInfo billInfo : billInfos) {

//将查出的进销项数据分开

if (billInfo.getbType().equals(ConstantStr.INPUTDATA))

List inputList = dataConver(inputMap);

map.put(OUTPUTDATA, outputList);

return map;

}

//转换成页面需要的格式数据————monthcountbill

private List dataConver(Map<Integer, Double> map) {

//将进项list填充月份的金额值，没有月份的填充0

for (Integer key : map.keySet())

if (key == index)

//保留小数点两位

arrayList.add(map.get(key));

//将进项数据遍历完的其他月份填充0

for (; index <= 12; index++)

arrayList.add(0);

return arrayList;

}

### 5.2.2 年度统计功能

年度统计为展示数据库里每一年的总体上的数据统计，数据也是为销项数据和进项数据组成，数据为数据库里已有的年份数据。

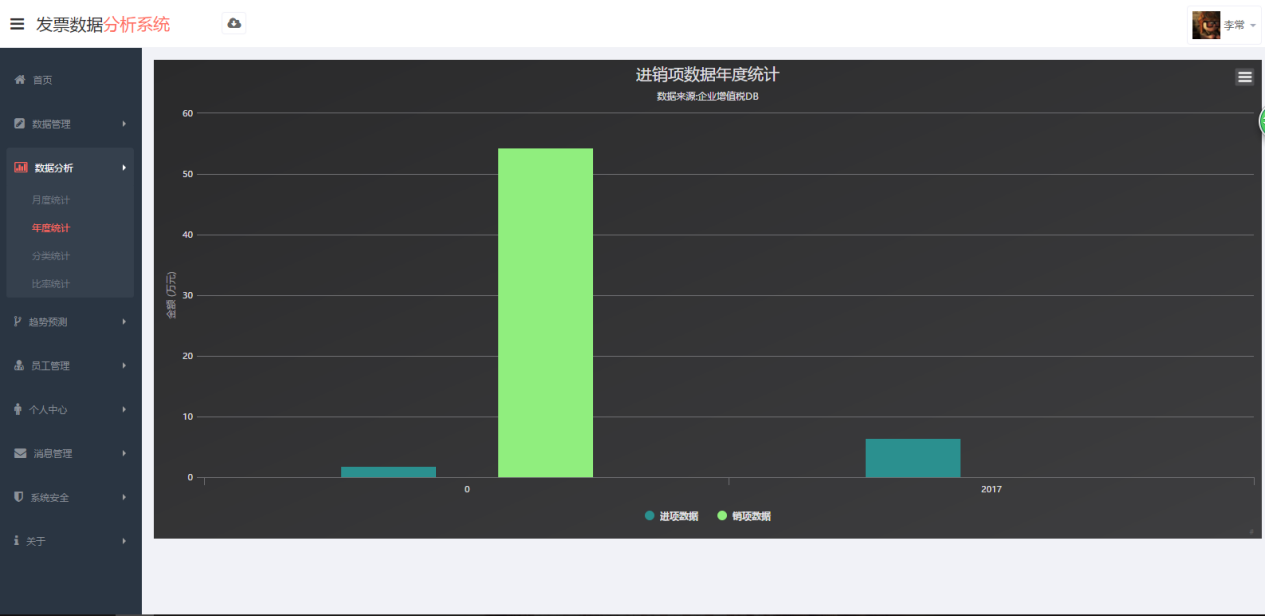


图5-4年度统计图

伪代码实现

public Map showYearBill() {

List<BillInfo> billInfos = billMapper.selectYearBill();

for (BillInfo billInfo : billInfos) {

//通过月份填充数据

//把查出来的进销项数据分开

if (billInfo.getbType().equals(ConstantStr.INPUTDATA)) {

inputList.add(billInfo.getTotalPrice());

year.add(billInfo.getbYear());

outputList.add(billInfo.getTotalPrice());

return map;

}

### 5.2.3 分类统计功能

由于年度统计和月度统计只是大体上对总数据的展示，没有区分到具体的哪一类，由于题目是对某生产加工型企业使用，所以用户想要看效果应该选择进项数据项目名称和销项数据项目名称的搭配，去观察它们每一个月的数据变从而得出数据变化上的一些信息，当然，进销项数据名称为数据库已经存在数据的项目名称，用户还可以查看上其他年份的相关数据变化形式。



5-5分类统计图

伪代码实现

public Map showCategoryName() {

//拿到进销项数据的说有项目名称

List inputlist = billMapper.selectAllbName(ConstantStr.INPUTDATA);

List outputlist = billMapper.selectAllbName(ConstantStr.OUTPUTDATA);

return map;

}

public Map showCategoryBill(String year, String inputbName, String outputbName) {

//拿到进销项数据

//数据整理与转换

List transfInput = dataTransf(inputlist);

return map;

}

//转换成页面需要格式的数据————categorycountbill

private List dataTransf(List<BillInfo> list) {

for (BillInfo billInfo : list)

//保留小数点2位

//将月份的金额数据填充到list集合里

if (billInfo.getbMonth() == index)

//没有月份的数据填充0

arrayList.add(0);

//其它月份的用0填充

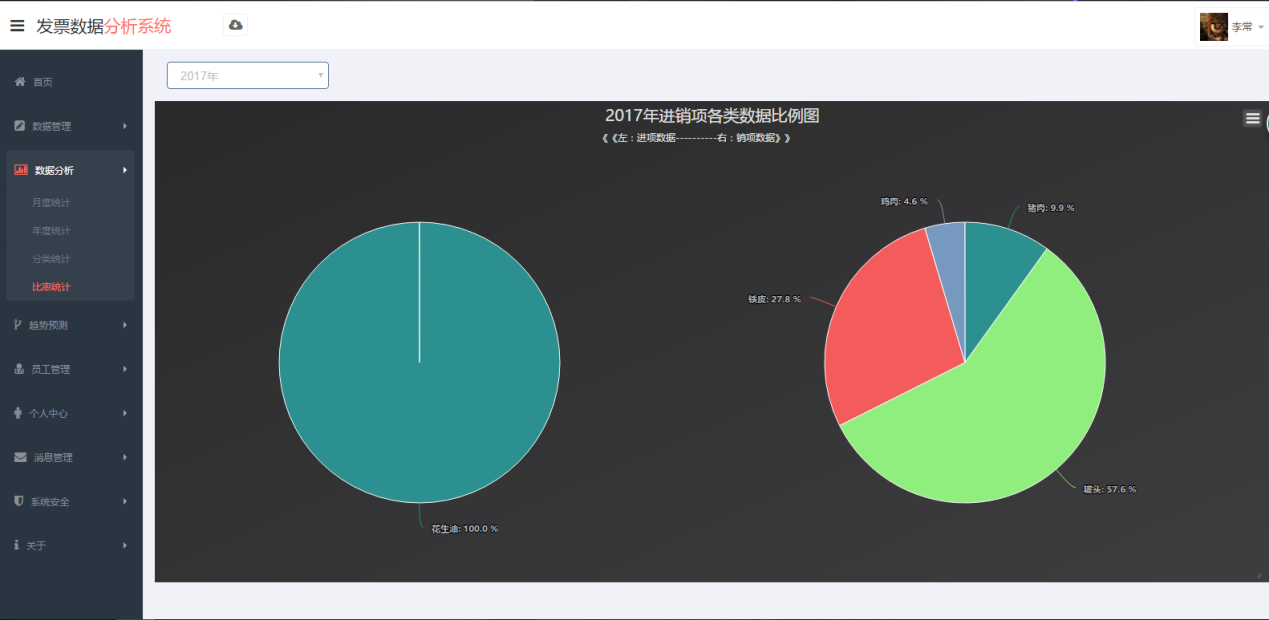
arrayList.add(0);

return arrayList;

}

### 5.2.4 比率统计功能

比率统计默认展示的是这一年我们有什么样的项目和每一个项目在整体上占的百分比，也可以通过选择年份查看其他年份项目的整体分布情况。



5-6比率统计图

伪代码实现

public Map showRateCountBill(String year) {

//从数据库里拿到数据

billMapper.selectRateCountBill(year, ConstantStr.INPUTDATA);

billMapper.selectRateCountBill(year, ConstantStr.OUTPUTDATA);

map.put(INPUTDATA, dataToList(inputlist));

return map;

}

## 5.3 趋势预测模块

### 5.3.1 线性预测功能

对于数据的趋势预测，我们采用的是线性回归的算法，通过分类分月份的预测方法，由于数据的多少对预测月份数据有误差影响，所以我们给出可以按前2、3、4、5、6个月的销项数据来预测下一个月的销项数据，用户从而可以根据预测的销项数据来调整下一个月的进项数据，对于每一类项目都可以进行这样的预测。



图5-7趋势预测图

伪代码实现

public Map getLinearLine(double[][] series)

//先准备两个矩阵

Matrix X = DenseMatrix.Factory.ones(series.length, 2);

Matrix Y = DenseMatrix.Factory.zeros(series.length, 1);

//赋值

Y.setAsDouble(series[i][1], i, 0);

//转置矩阵

Matrix transX = X.transpose();

//矩阵相乘

Matrix mtimesX = transX.mtimes(X);

//逆矩阵

Matrix invX = mtimesX.inv();

//再运算

Matrix theta = invX.mtimes(transX).mtimes(Y);

//预测数据 核心代码

double[][] point = new double[1][2];

point[0][0] = 6;

point[0][1] = theta.getAsDouble(1, 0) \* 6 + theta.getAsDouble(0, 0);

double[][] line = new double[2][2];

line[1][0] = 0;

line[1][1] = theta.getAsDouble(1, 0) \* 0 + theta.getAsDouble(0, 0);

line[0][0] = 6;

line[0][1] = theta.getAsDouble(1, 0) \* 6 + theta.getAsDouble(0, 0);

## 5.4 员工管理模块

### 5.4.1 查看员工功能

只能看到用户的头像、员工号、用户名称、手机号、邮箱和角色等消息，这个主要用于操作员和决策员使用，没有修改和删除的权限。

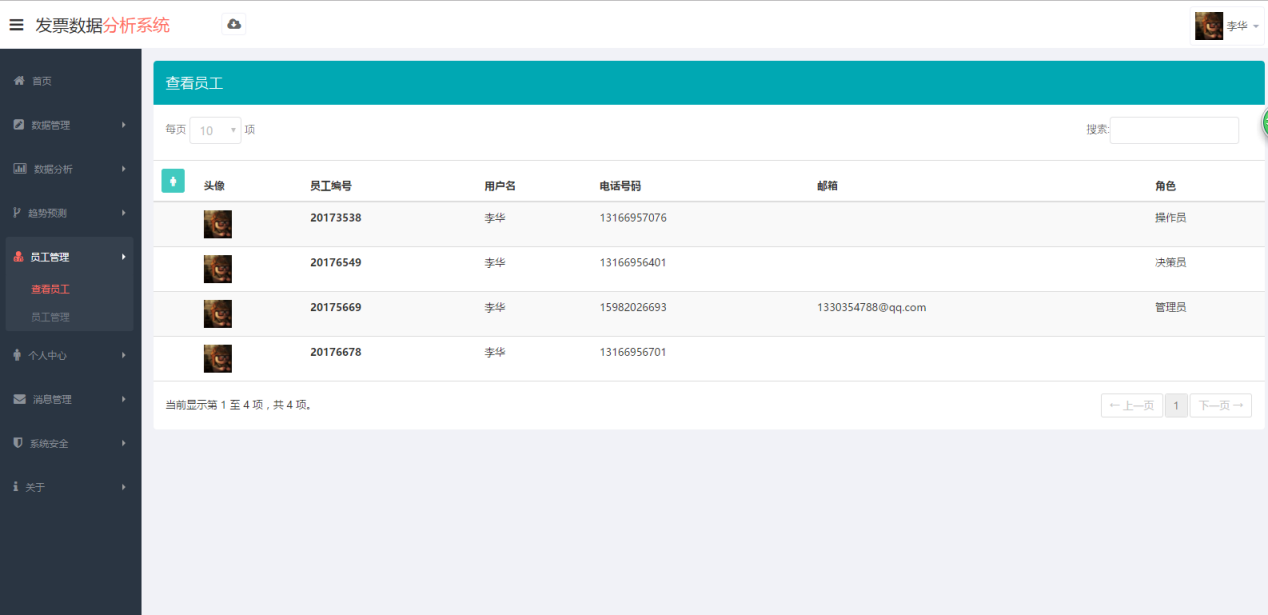


图5-8查看员工图

### 5.4.2 员工管理功能

员工管理模块分为员工的分配，员工角色的更改和员工的删除，当然这些操作都只有管理员才有的权限，员工的分配，管理员只会给用户分配一个角色和电话号码，电话号码在数据库为唯一存在，不可重复。管理员只能修改员工的角色，不能修改员工的个人信息，员管理也可以通过员工号删除用户，这一切的操作在安全模块的日志系统里都有记录。

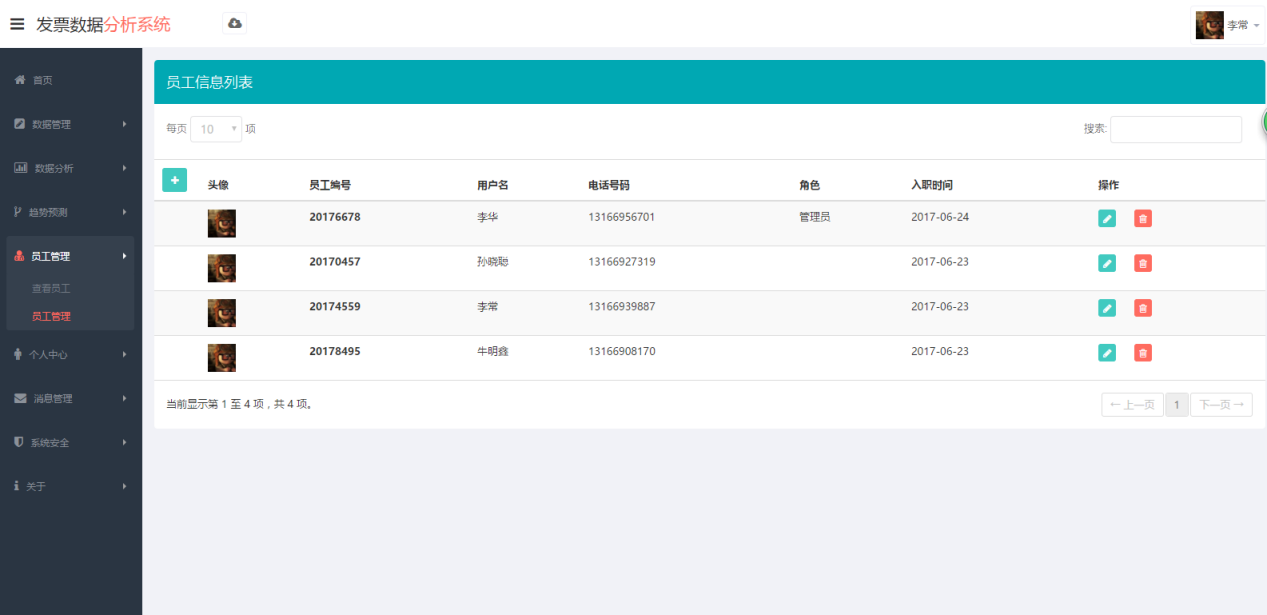


图5-9员工管理图

伪代码实现

//添加员工

public int addStaff(String sPhone, String role,Session session) {

//添加员工

Staff staff = new Staff();

//拿到加密后的密码和摘要

String sPassword = (String) map.get(SecurityService.ENCODE\_RESULT\_KEY\_PASSWORD);

String Ssalt = (String) map.get(SecurityService.ENCODE\_RESULT\_KEY\_SALT);

//电话号码为为唯一键，捕捉异常

//初始化一个员工

result = staffMapper.insert(staff);

//添加角色

result = roleMapper.insert(role1);

//添加员工与角之间的关系

result = staffMapper.insertStaffAndRoleRelation(sId, rId);

//生成系统日志

logMapper.insert(CommonLog.createLogMessage(message,session);

}

//随机声生成后半 摘要

public Map<String, String> encodePassword(String password, String preSalt) {

//后半 摘要

String subSalt = randomNumberGenerator.nextBytes().toHex();

//生成

SimpleHash simpleHash = new SimpleHash(ENCODE\_ALGORITHM\_NAME,

password, salt, ENCODE\_HASH\_ITERATION);

String encodePassword = simpleHash.toHex();

//返回 生成的 和 摘要

map.put(ENCODE\_RESULT\_KEY\_PASSWORD, encodePassword);

}

## 5.5 个人中心模块

### 5.5.1 个人资料功能

用户拿着管理员角色登录系统是用户是没有激活的，用户可以通过登录后修改自己的个人信息，包括验证邮箱或者电话号码修改密码，用户可以查看自己的个人信息，对自己的个人信息可以在修改个人信息页面修改。



图5-10个人信息图

伪代码实现

public Staff selectByPrimaryKey(String sId) {

return staffMapper.selectByPrimaryKey(sId);

}

### 5.5.2 修改信息功能

修改个人信息分为两个方面，一是头像的修改，用户可以选择一张图片，截取自己的头像大小和尺寸，然后提交头像路径到后台数据库，在一个是基本信息的修改，用户可以修改自己的用户和生日、地址等其它个人信息，但用户个人信息的改变不记录安全日志记录里。

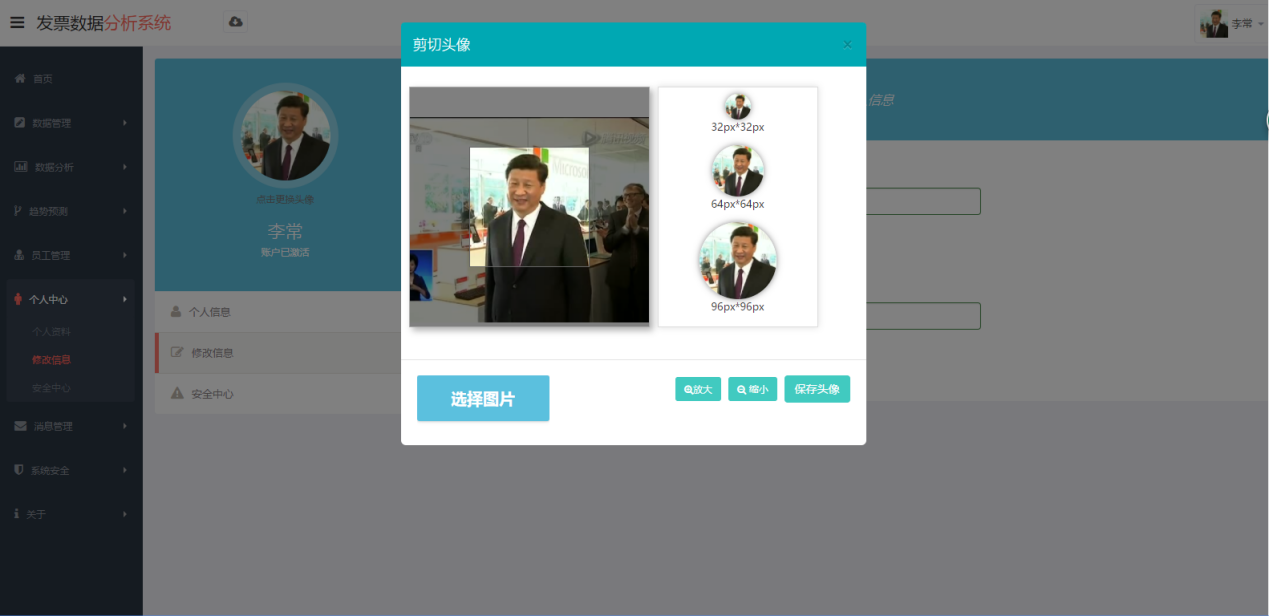


图5-11个人信息修改图

伪代码实现

//头像照片选择

$(window).load(function () {

//默认头像图片

var options ={

thumbBox: '.thumbBox',

spinner: '.spinner',

imgSrc: '../../img/avatar.jpg'

}

cropper = $('.imageBox').cropbox(options);

//选择图片文件

$('#upload-file').on('change', function () {

var reader = new FileReader();

reader.onload = function (e) {

options.imgSrc = e.target.result;

cropper = $('.imageBox').cropbox(options);

changeFontAvatar();

}

reader.readAsDataURL(this.files[0]);

})

//图片放大

$('#btnZoomIn').on('click', function () {

cropper.zoomIn();

})

//图片缩小

$('#btnZoomOut').on('click', function () {

cropper.zoomOut();

})

$("#cropAvatar").on('mouseup', changeFontAvatar);

});

//个人信息修改

public int updateStaffInfo(Profile profile, Session session) {

//拿到session的staff对象

Staff staff = (Staff) session.getAttribute(ConstantStr.STAFFINFO);

//激活账户

if (staff.getSMark()==0)

// 将profile要更改的信息填充到staff对象里

//将字符串的“男” “女” 转换为 true和false

staff.setSSex(EnumSex.ToSex(profile.getSSex()));

//将字符串的日期转换为date类型

Date birthday = ToolDates.parseDateStr(sbirthday);

staff.setSBirthday(birthday);

staffMapper.updateByPrimaryKey(staff)

//放入session缓存

session.setAttribute(ConstantStr.STAFFINFO, staff);

}

### 5.5.3 安全中心功能

安全中心这主要分为两个方面，一个邮箱，用户可以绑定自己的邮箱，但是邮箱必须验证，当用户输入了自己的合法邮箱后会收到一条邮箱验证码，成功提交邮箱验证码后下次就可以通过邮箱加密码的形式登录了，还有就是手机号，可能用户会更换手机号，同样会通过短信验证的形式验证合法正确了之后行进行修改，还有就是用户可能要修改密码，修改密码的形式也是基于电话号码和邮箱来进行安全的修改。

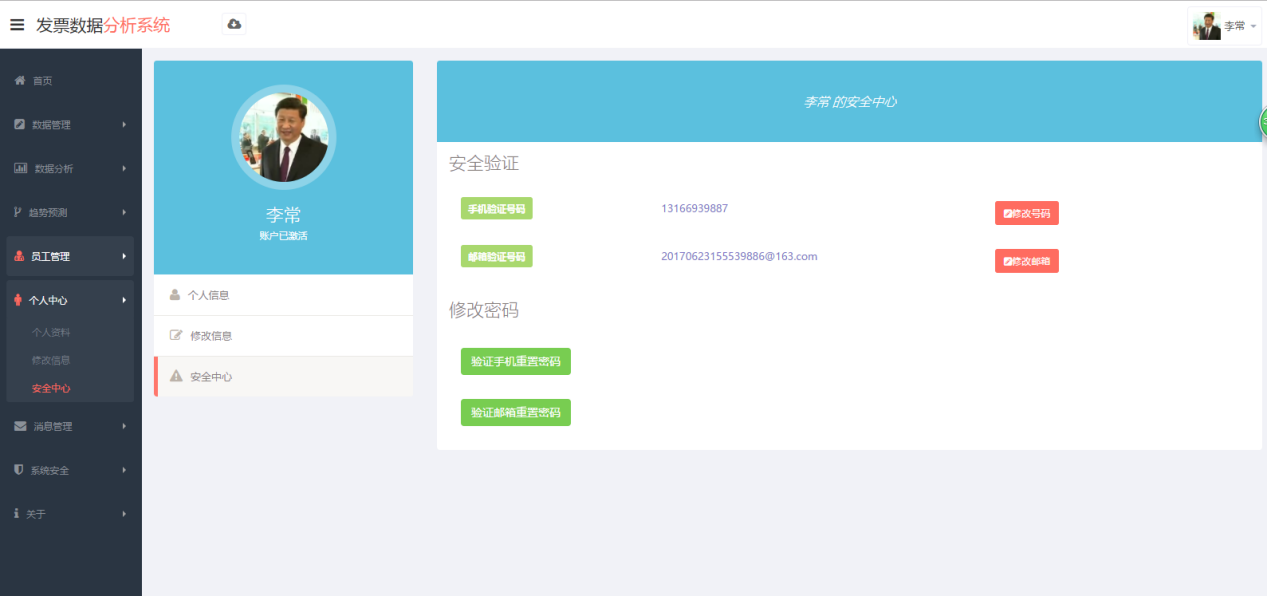


图5-12安全中心图

## 5.6 消息管理模块

### 5.6.1 我的消息功能

因为系统内部可以相互通信，所以当有新的消息时候可以查看自己的新消息，当然在用户刚进入系统首页时也有新消息提示，当用户把消息选为已读时小事提示就回消失，用户也可以把读过的消息删除，好比qq邮箱的邮件形式。

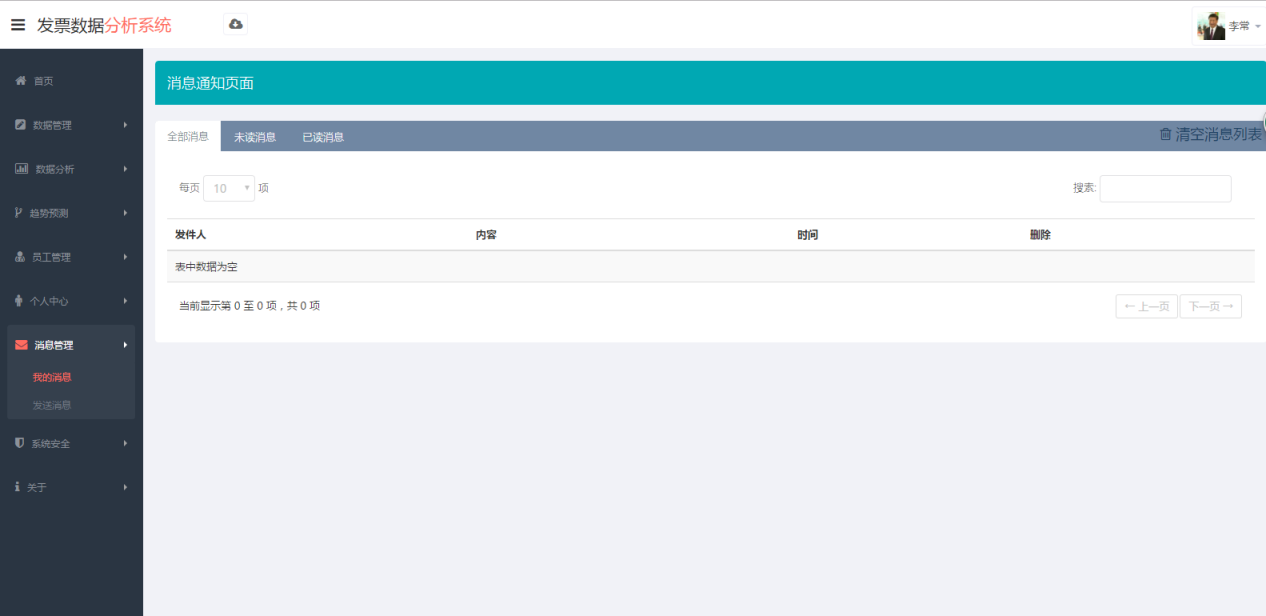


图5-13消息模块图

伪代码实现

public List<MsgInfo> getMsgByToSId(String toSId) {

return msgMapper.selectByToSId(toSId);

}

public int markReadByMIds(String[] mIds) {

//将所有的消息 都不标记位已读

msgMapper.markReadByMId(mId);

return mIds.length;

}

public int deleteByMIds(String[] mIds) {

//挨个删除

msgMapper.phonyDeleteByPrimaryKey(mId);

return mIds.length;

}

### 5.6.2 发送消息功能

有些时候系统内部员工之间可能要沟通交流，需要相互发送消息，你只需要知道对方员工的员工编号就能够相互发送消息了。

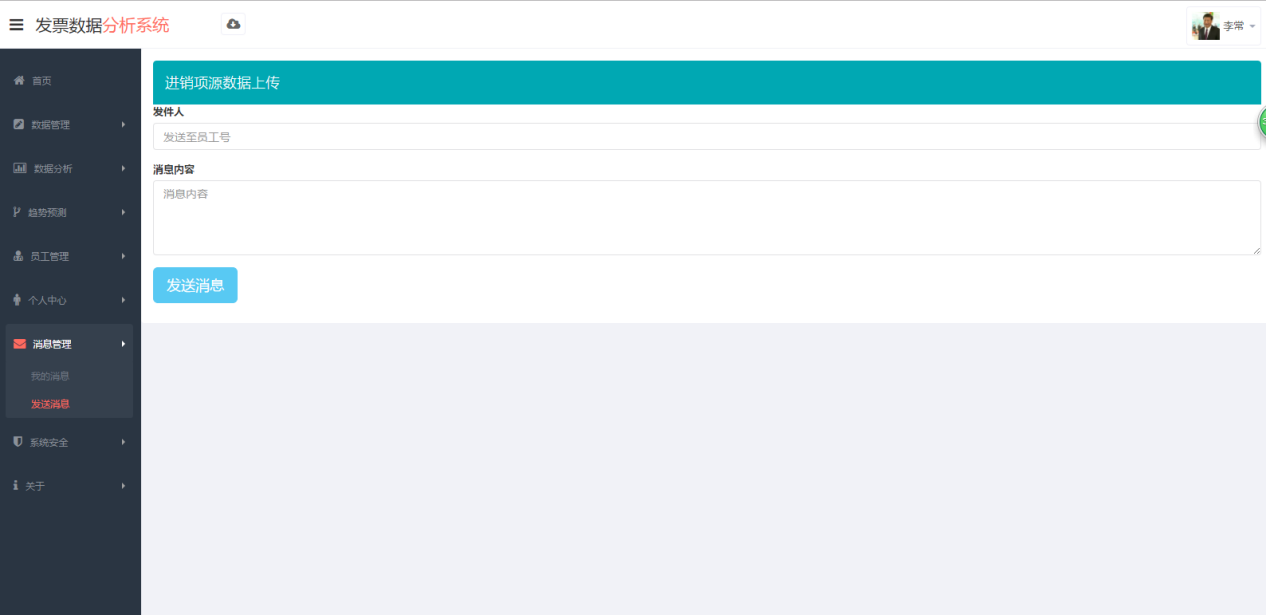


图5-13发送消息图

伪代码实现

public int sendMsg(String toSId, String mContent, String fromSId) {

//初始化数据

Msg msg = new Msg();

return msgMapper.insert(msg);

}

## 5.7 系统安全模块

### 5.7.1 系统日志功能

由于系统为数据分析系统，对数据的安全和监控特别重要，所以对说有用的数据的操作都应该有记录，方便数据的恢复性，从而也可以达到一定程度上的安全性上的要求，记录的形式为谁什么时候干什么事这样的形式。



图5-14系统日志图

伪代码实现

public List<Log> showLogInfo() {

return logMapper.selectAll();

}

# 测试

## 6.1 黑盒测试

### 6.1.1 登录测试

在未填写内容的时候点击立即登录，会有相应的提示信息，提示需要输入的内容，输入完后，才可以进行登录。



图6-1

员工号输入后，并且没有输入其他的内容，会有错误的提示，在输入完整以后，才可以进行登录。没有验证码的输入的时候点击登录，会有相同的提示信息出现。



图6-2

### 手机、邮箱验证和重置密码

手机验证的时候内容输入不完整，进行内容提示。如下图：在两次密码输入不一致的时候，会提示（密码不一致），必须两次密码一致的时候，才可以进行提交修改。



图6-3

邮箱验证和手机验证会有相同的提示效果，来让用户有很好的体验。如下图：两次密码一致的情况。



图6-4

### 数据管理验证

在进行数据上传的时候，会提示上传的excel形式应与提供的excel模板一致和税务类型的选择可单独上传一个进项或者销项也可以同时上传进项和销项，然后选择文件上传文件。



图6-5

文件上传成功后，就可以对数据进行一系列的操作，如果你在修改的时候忘记输入某个数据，它会返回数据不为空，才可以进行修改。如下图所示

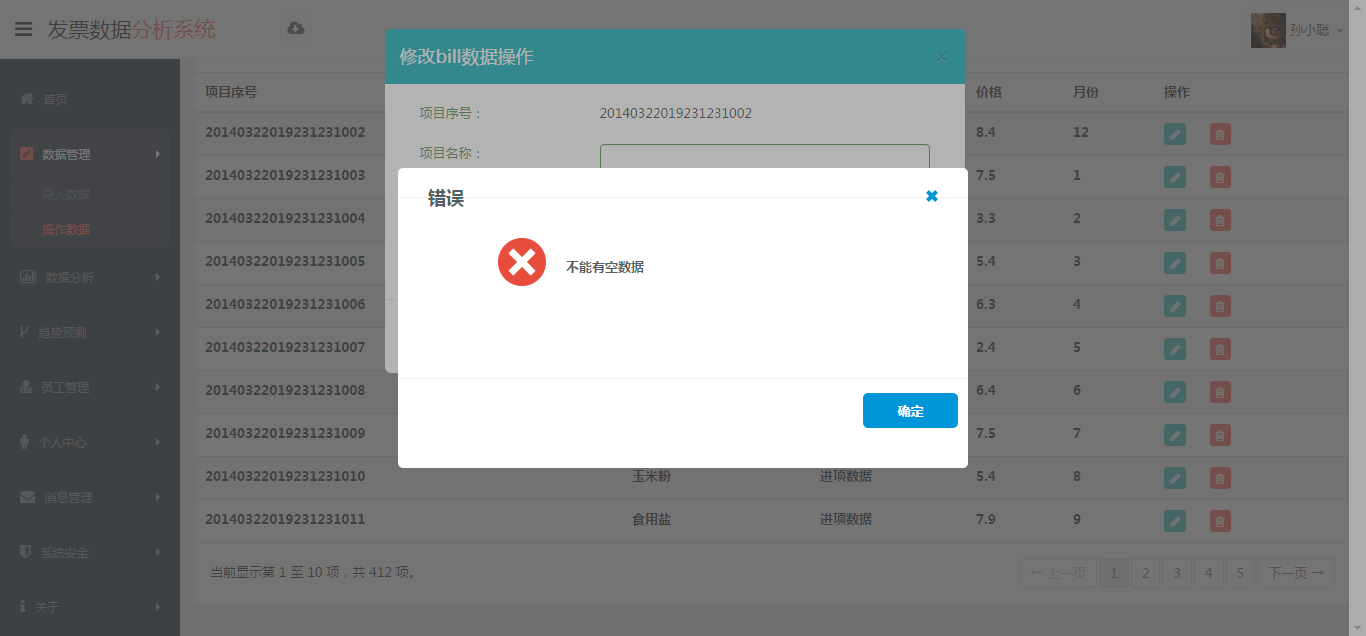


图6-6

### 数据分析

在图表显示的时候在有数据的年份，会进行相应的数据显示，如果该年份并没有数据，则会不显示任何内容。如下两图所示：



图6-7

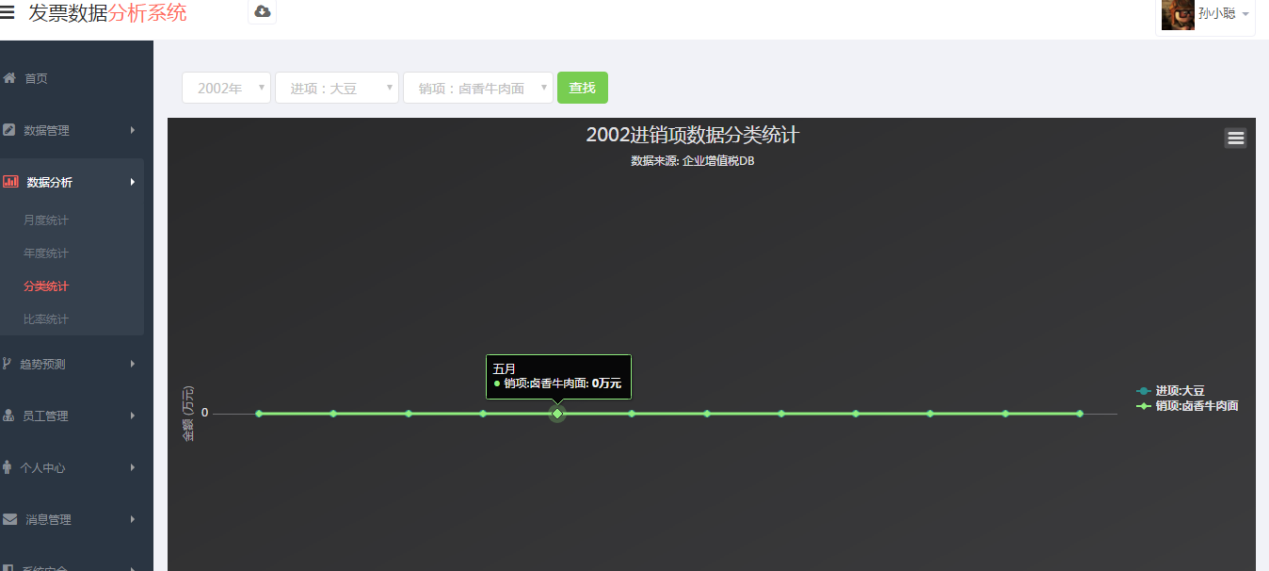


图6-8

### 趋势预测

主要是实现的线性预测可以展示某个项目的所选择个阶段和预测月份。

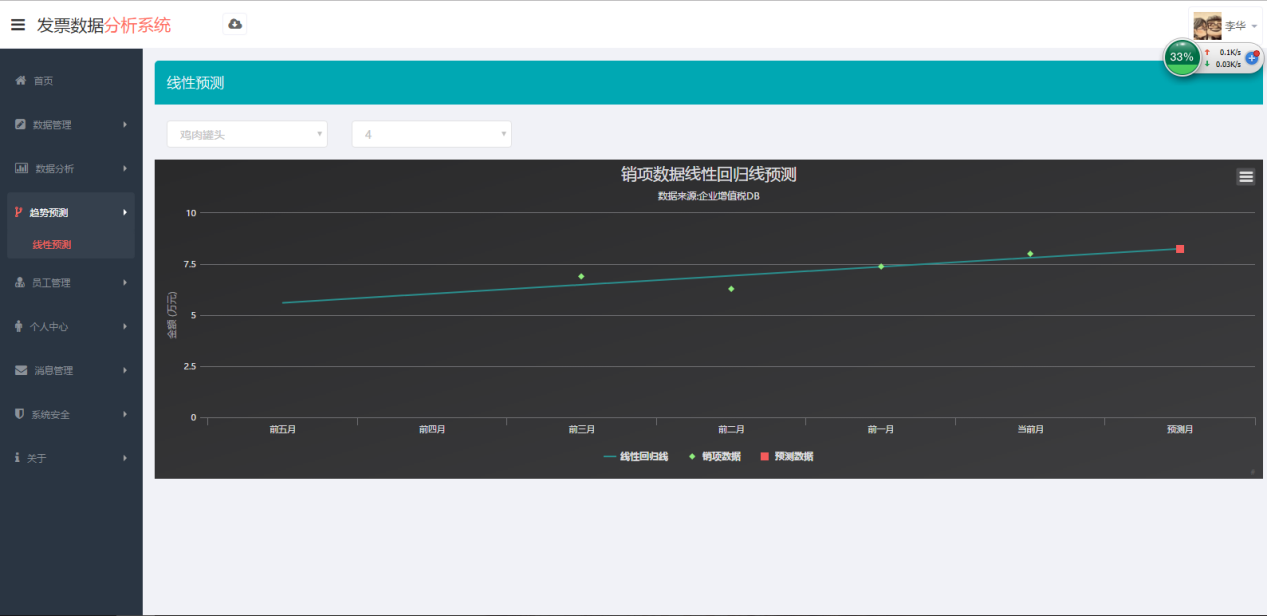


图6-9

### 员工管理

员工管理进行修改员工角色，修改成功后会提示修改成功。在点击删除的时候，会显示删除失败（三种管理人权限不同）。

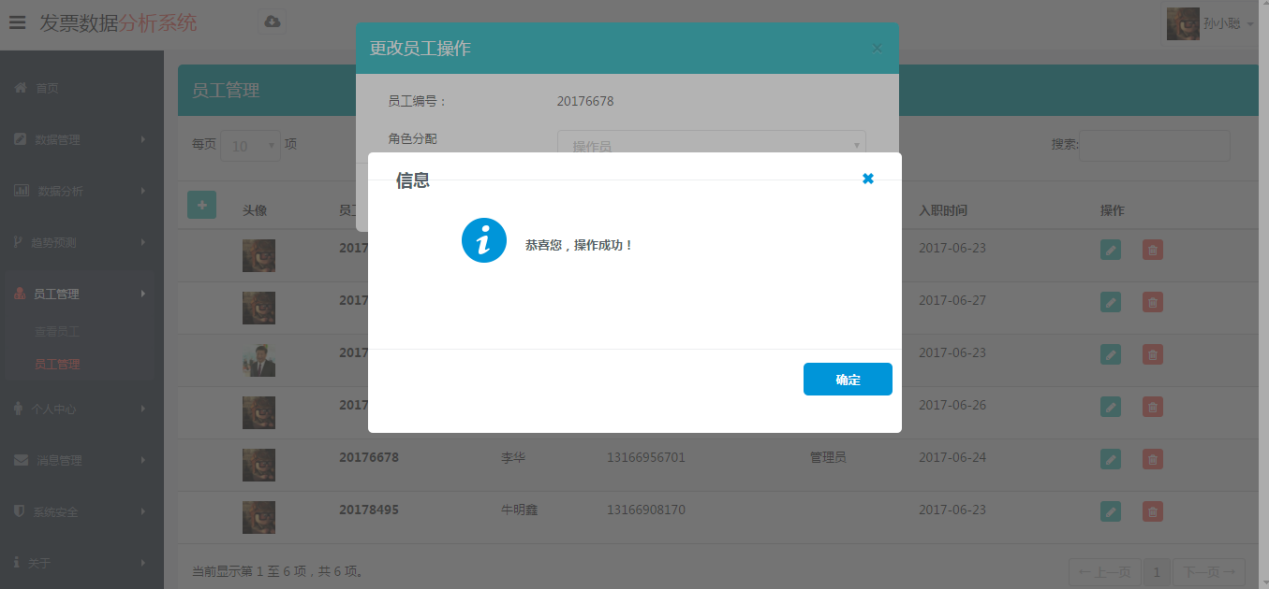


图6-10

### 个人中心

进行修改个人信息的时候，信息输入完整后点击保存修改会显示，更改成功。



图6-11

在进行修改手机号的时候手机号输入错误，会提示输入正确的手机号。

邮箱验证，也可以和手机验证有相同的提示功能。



图6-12

### 消息管理

进行员工之间发送消息的时候，员工号输入错误，会有收件员工的错误，且消息发送失败。需要正确的员工号，才可以发送消息，消息发送完成后可以在我的消息页面进行查看。



图6-13

### 系统安全

在对系统的敏感信息进行修改后，会生成系统日志，并且对以前的修改进行记录。



图6-14

### 关于

可以正常的显示关于我们和关于系统

## 6.2 白盒测试

### 6.2.1 数据访问层单元测试

a)BillMapperTest 测试

BillMapper主要负责数据访问层与Bill实体进行数据交互，主要方法如下：

int insert(Bill record);

List<Bill> selectAll();

测试的主要代码为下：

@RunWith(SpringJUnit4ClassRunner.class)

@WebAppConfiguration

@ContextConfiguration(locations = {"classpath:spring/spring-mybatis.xml"})

public class BillMapperTest {

@Autowired

private BillMapper billMapper;

/\*\*

\* 测试查询数据

\*/

@Test

public void selectAllTest() {

//查询 Bill

List<Bill> bills = billMapper.selectAll();

//断言不为空

Assert.assertNotNull(bills);

}

/\*\*

\* 测试插入一条 数据

\*/

@Test

public void insertTest() {

//生成数据

Bill bill = new Bill();

bill.setBId(ToolRandoms.randomBillId());

bill.setBType("进项数据");

bill.setBName("猪肉");

bill.setBMonth(1);

bill.setBPrice(ToolRandoms.random.nextDouble());

bill.setBUpdateTime(new Date());

bill.setBMark(1);

//插入数据

int resultCode = billMapper.insert(bill);

//断言 影响一行记录

Assert.assertEquals(resultCode, 1);

}

}

b)StaffMapperTest 测试

StaffMapper主要负责数据访问层与Staff实体的交互，方法签名如下：

int deleteByPrimaryKey(String sId);

/\*\*

\* 用于删除中间表staff\_role

\*/

int deleteStaffRole(String sId);

Staff selectByPrimaryKey(String sId);

List<Staff> selectAll();

/\*\*

\* 显示指定字段的员工信息

\* @return

\*/

List<StaffInfo> selectAllStaffInfo();

int updateByPrimaryKey(Staff record);

/\*\*

\* 添加用户

\*/

int insert(Staff record);

/\*\*

\* 添加用户和角色之间的关系

\*/

int insertStaffAndRoleRelation(@Param("sId") String sId, @Param("rId") String rId);

/\*\*

\* 找回密码（也就是重置密码）

\*/

int updatePassword(@Param("sId") String sId, @Param("newPassword") String newPassword, @Param("sSalt") String sSalt);

### 6.2.2 业务逻辑层集成测试

a)BillServiceTest

BillService主要负责与BillMapper的交互，测试主要保证Bill各方法业务逻辑没有错误。

b)ExcelServiceTest

ExcelService主要负责Excel相关的业务逻辑，ExcelServiceTest主要对ExcelService进行测试。

c)PermsServiceTest

PermsService主要负责与PermsMapper交互同时处理自己的业务逻辑，PermsServiceTest 主要对PermsService 进行测试。

d)RoleServiceTest

RoleService 主要负责与RoleMapper进行交互同时处理自己的业务逻辑，RoleServiceTest主要对RoleServie进行测试。

e)SecurityServiceTest

SecurityService主要负责安全相关的业务逻辑，SecurityServiceTest主要对SecurityService进行测试。

f)StaffServiceTest

StaffService主要负责与StaffMapper进行交互同时处理自己的业务逻辑，StaffServiceTest主要对StaffService进行测试。