# 国电煤炭交易平台详细设计说明书

## 需求概述

本系统主要功能包括：用户注册登录、创建用户、买卖方挂牌摘牌、操作资讯（包含浏览、编辑、审核等功能）、资金预存、缴纳保证金、交易审核功能。

## 总体方案设计

### 2.1编写目的

本文件阐述了国煤交易平台系统的软件总体设计、软件运行配置与应用方式以及使用的关键技术等。

### 2.2设计依据

依据需求分析中输出的相关文档中阐述的产品功能，进行对应技术方案输出。

参考教学介绍的WEB系统架构方案，结合产品实际情况、功能演进规划，进行技术架构和演进规划。

### 2.3术语、定义和缩略语

#### 2.3.1.术语、定义

| **名词** | **解释** |
| --- | --- |
| SpringBoot | Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。 |
| MyBatis-Plus | 一个 MyBatis 的增强工具，在 MyBatis 的基础上只做增强不做改变，为简化开发、提高效率而生。 |
| Vue.js | 一套用于构建用户界面的渐进式框架。 |
| MySql | 一个关系型数据库管理系统。 |
| Nginx | 一个高性能的HTTP和反向代理web服务器。 |
| Docker | 一个开源的应用容器引擎。 |

#### 2.3.2缩略语

### 2.4概述

#### 2.4.1系统目标

**用户体验**：尽量贴近用户习惯，化繁为简。

#### 2.4.2设计原则

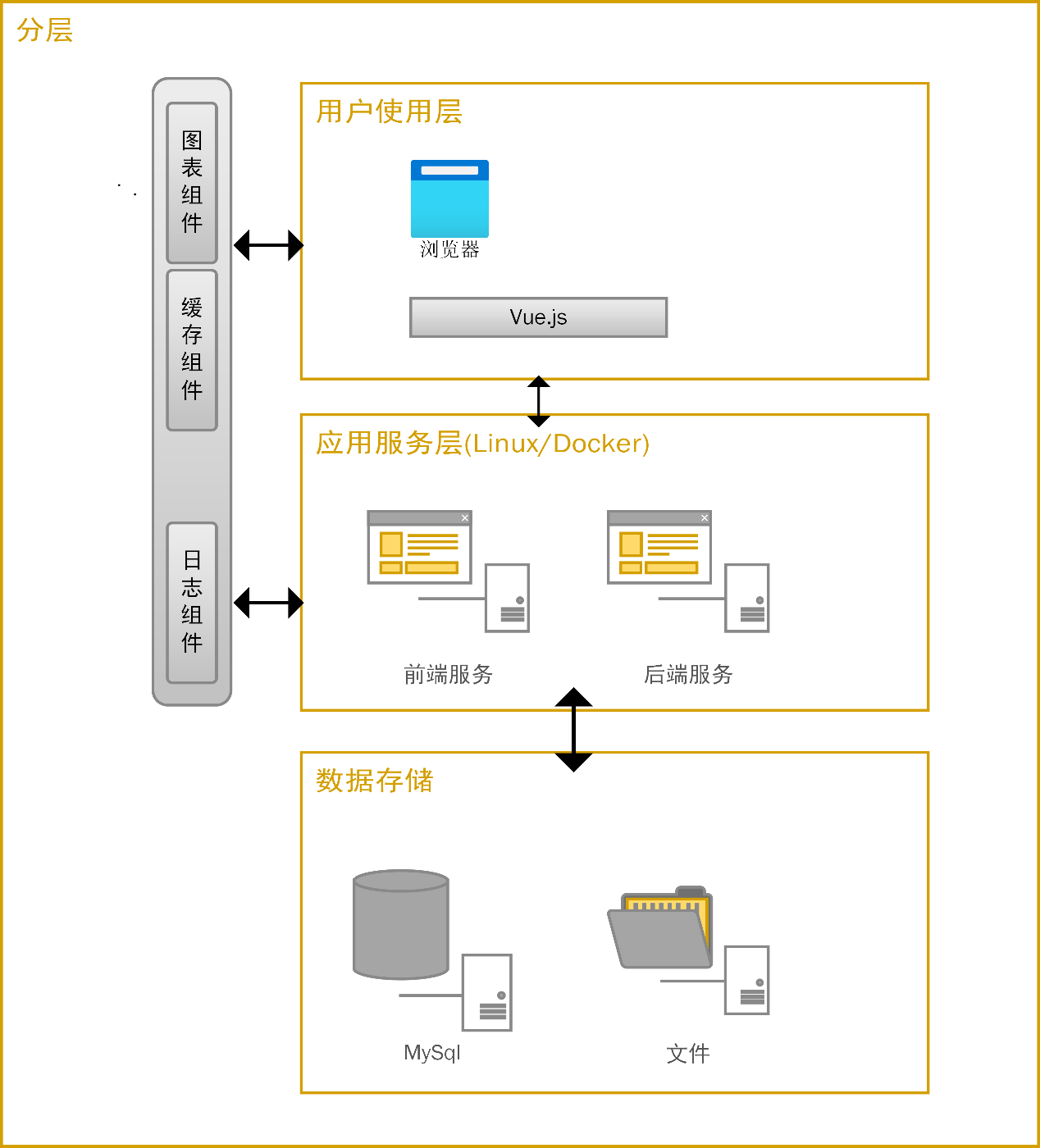
**高效运维**：提高运维效率、减少运维成本

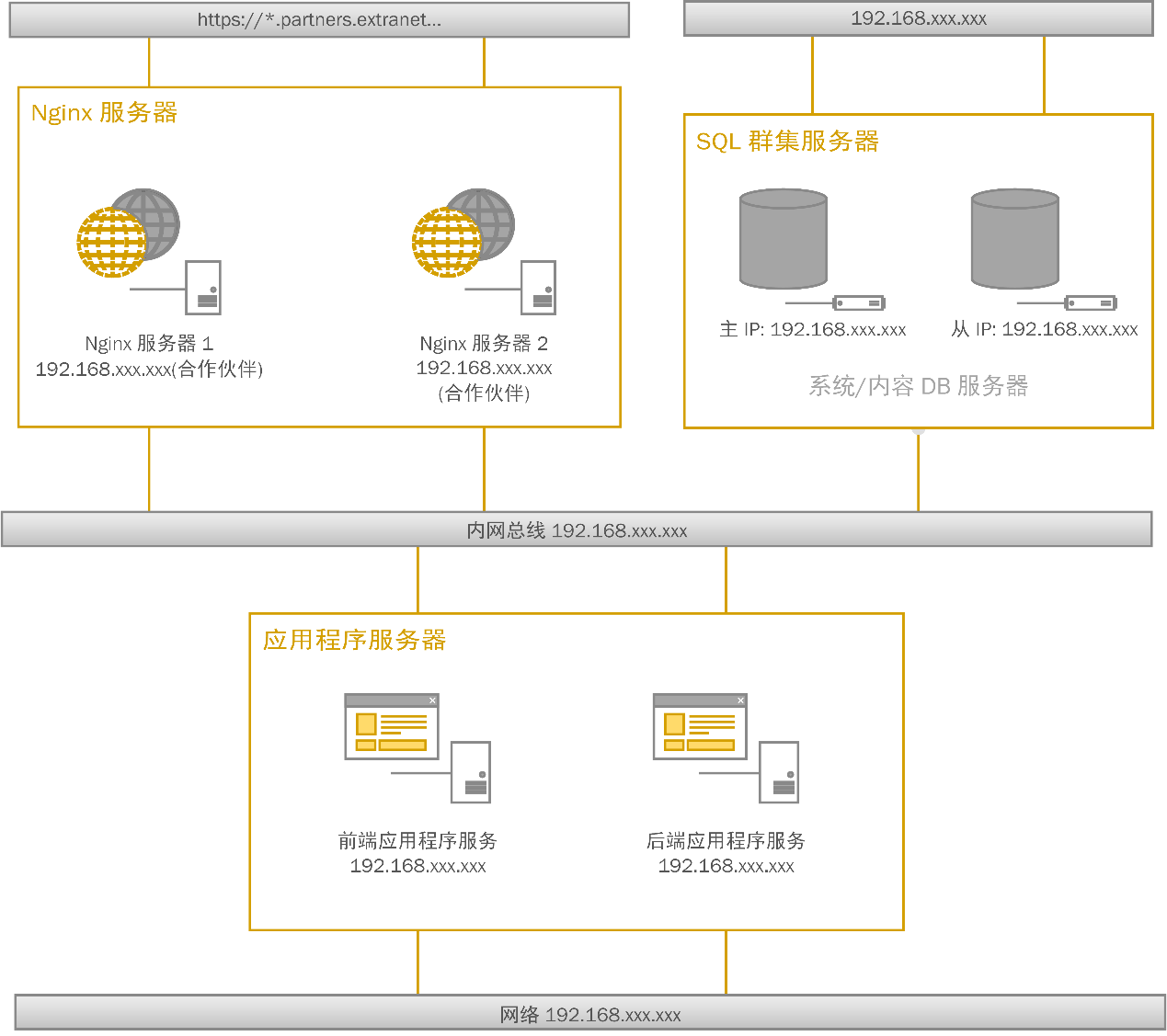
#### 2.4.3演进规划

暂无

### 2.5整体方案

#### 2.5.1技术架构





##### 2.5.1.1用户使用层

支持在PC电脑浏览器上使用

##### 2.5.1.2应用服务层

系统后台基于业内主流WebService框架Springboot进行应用开发。

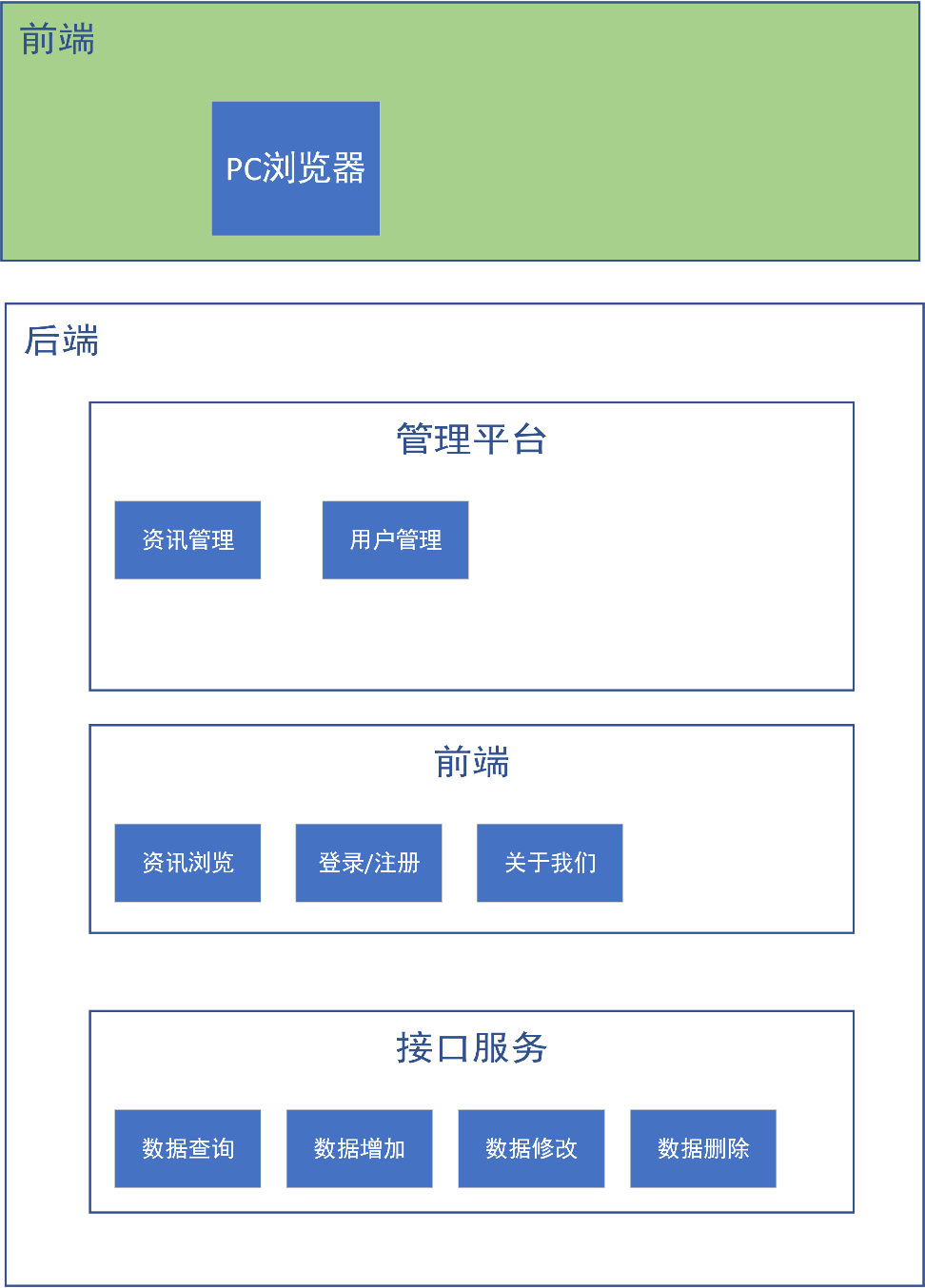
系统WEB前端使用主流渐进式框架Vue.js.

##### 2.5.1.3数据存储层

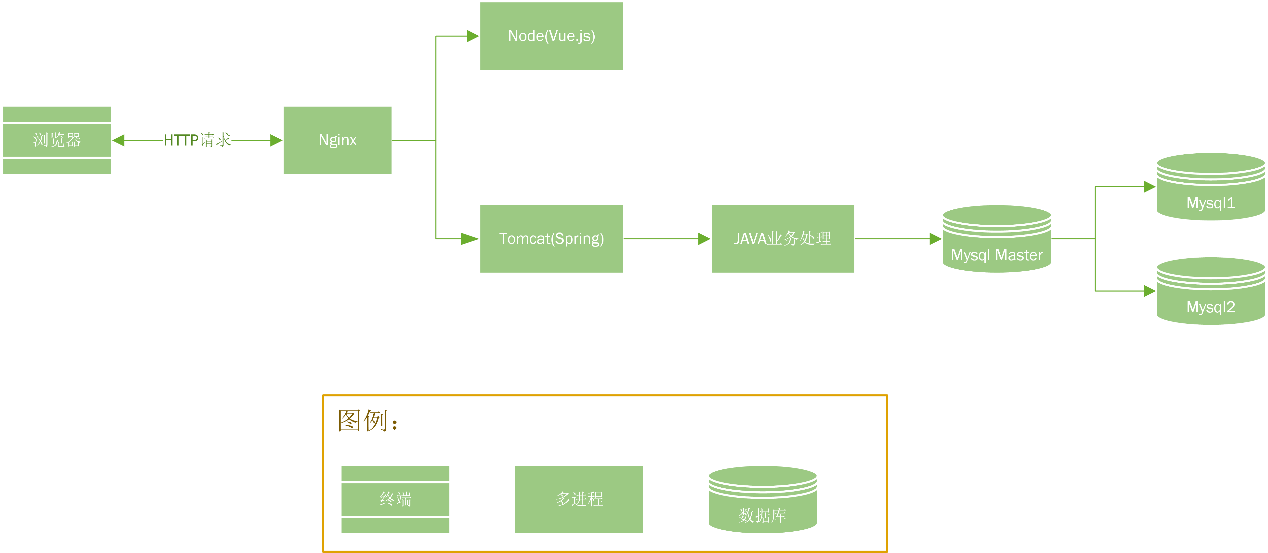
1. 关系型数据存放与：MySql数据库
2. 非关系型数据库存放于：文件

#### 2.5.2.功能架构

TODO: 有待完善



#### 2.5.3.运行流程



运行步骤：

1. PC浏览器发起HTTP请求
2. Nginx处理请求分发至对应前后端服务器
3. 对应服务器处理数据后，返回给Nginx输出

#### 2.5.4.部署架构

docker-compose

nginx

#### 2.5.5.性能设计

##### 性能指标

1. 整体用户量 20000
2. 并发数 > 15000

##### Web服务器要求

| **序号** | **参数** | **指标** | **备注** |
| --- | --- | --- | --- |
| 01 | 平均每次请求响应时间 | ≤3s | 每次从发送请求到获得响应数据的时间 |
| 02 | 成功的请求率 | ≥98% | 发送的所有请求成功率 |
| 03 | 失败的请求率 | <2% | 发送的所有请求失败率 |
| 04 | 点击成功率 | ≥99% | 前端点击得到响应的概率 |
| 05 | 点击失败率 | <1% | 前端点击未得到响应的概率 |
| 06 | 每秒点击成功率 | ≥95% | 1s时间内的请求成功率 |
| 07 | 每秒点击失败率 | <5% | 1s时间内的请求失败率 |

### 2.6.功能详述

#### 2.6.1资讯模块

##### 2.6.1.1模块说明

资讯标题列表由若干资讯详细信息页面的链接组成，点击相应标题可打开一个新页面用于浏览详细资讯  
  
咨询内容由资讯编辑人员编写，由资讯审核人员审核通过后发布到该页面。资讯有效期结束后由资讯编辑人员进行撤下操作

资讯列表上方有一个搜索框，用于根据资讯标题搜索包含有该关键字的所有资讯

资讯编辑完成后可存草稿，也可提交给审核人员进行审核。

审核人员可驳回资讯，也可同意发布资讯

##### 2.6.1.2实现方案

Controller层：  
  
1、public ResponseData showNews(@RequestParam(value="size",defaultValue = "20") int size,@RequestParam(value = "current",defaultValue = "1") int current)加载时获取咨询标题  
  
2、public ResponseData detailNews(@PathVariable("newsId") Long id)：点击资讯查看详细内容  
  
3、public ResponseData moreNews(@PathVariable("newsTitle") String title)：查询更多咨询  
  
4、public ResponseData pushNews(@RequestBody News news)：发布咨询（提交审核）  
  
5、public ResponseData saveDraft(@RequestBody News news)：存草稿  
  
6、public ResponseData auditNews(@PathVariable String type,@PathVariable String newsId)审核咨询

7、public ResponseData showAuditingNews(@RequestParam(value="size",defaultValue = "20") int size,@RequestParam(value = "current",defaultValue = "1") int current) 获取待审核的咨询标题

Service层：  
  
1、Page<News> getAllNews(int current, int size)刷新页面展示所有资讯  
  
2、Page<News> getAuditingNews(int current,int size)展示所有待审核资讯  
  
3、News getNewsById(Long id)点击单条资讯查看详细信息  
   
4、List<News> getNewsByTitle(String title)查询咨询  
  
5、News setOneNews(News content, Long authorId)发布资讯  
   
6、News setOneDraft(News content, Long authorId)存草稿  
   
7、String newsAudit(String type,Long newsId)资讯审核

#### 2.6.2.订单模块

##### 2.6.2.1模块说明

该模块包含获取订单列表、获取订单详情、生成订单三个功能

##### 2.6.2.2实现方案

Controller层：

1、public ResponseData orderList(@RequestParam(defaultValue = "1", required = false) int page, @RequestParam(defaultValue = "1", required = false) int limit) 获取订单列表

2、public ResponseData newOrder(@PathVariable Long requestId) 创建新订单

3、public ResponseData getDetail(@PathVariable String orderId) 获取订单详情

Service层：

1、Map<String, Object> list(Order order, int page, int limit)

2、Order addNewOrder(Long userId,Long requestId)

3、Order getOrderDetail(Long id)

#### 2.6.3.需求挂牌模块

##### 2.6.3.1模块说明

除申请单位等特殊信息不可编辑、由系统自动获取之外，用户需要手动填写煤炭质量等挂牌信息

##### 2.6.3.2实现方案

Controller层：  
  
1、public ResponseData getList(@RequestParam(defaultValue = "", required = false) Long userId, @RequestParam(defaultValue = "1", required = false) int page, @RequestParam(defaultValue = "10", required = false) int limit)：获取公共可用的挂牌需求列表  
  
2、public ResponseData getPublicDetail(@RequestParam int id)：获取某一个可用的公共挂牌需求详细信息  
  
3、public ResponseData myList(@RequestParam(defaultValue = "1", required = false) int page, @RequestParam(defaultValue = "10", required = false) int limit)：获取用户所有挂牌需求  
  
4、public ResponseData myDetail(@PathVariable long id)：获取用户自己的挂牌需求详细信息  
  
5、public ResponseData publish(@RequestBody Request req)：发布新挂牌需求或保存新挂牌需求草稿  
  
6、public ResponseData edit(@RequestBody Request req)：编辑用户自己的挂牌信息  
  
7、public ResponseData auditPending(@RequestParam(defaultValue = "1", required = false) int page, @RequestParam(defaultValue = "10", required = false) int limit)：审核员：获取待审核的挂牌列表  
  
8、public ResponseData auditDetail(@PathVariable long request\_id)：获取待审核挂牌信息列表中的某一详细挂牌信息  
  
9、public ResponseData auditDo(@RequestBody Map<String, Object> request, @PathVariable long req\_id)：审核员：对挂牌信息进行审核操作  
  
10、public ResponseData contractUpload(@RequestParam MultipartFile contract, @PathVariable("req\_id") long requestId)：交易用户进行交易合同的上传  
  
11、public ResponseData contractFile(HttpServletResponse response, @PathVariable("req\_id") long requestId, @RequestParam String path)：获取上传的交易合同  
  
12、public ResponseData getComName()：获取公司名称  
  
Service层：  
  
1、Map<String,Object> getPublicList(Long userId, int page, int limit)：获取需求列表  
  
2、Request getPublicDetail(int id)：获取详细需求  
  
3、long newReq(Request request)：新建需求  
  
4、int edit(Request request)：编辑需求  
  
5、Map<String,Object> myList(Long userId, int page, int limit)：用户的需求  
  
6、Request myDetail(Long userId, long req\_id)：用户的需求详情  
  
7、Map<String, Object> auditPending(int page, int limit)：待审核需求列表  
  
8、Request auditDetail(long req\_id)：未审核需求的详情  
  
9、boolean doAudit(Request request)：审核需求  
  
10、boolean updateContract(long reqId, long userId, String contractPath)：更新合同文件路径  
  
11、String getComName(long id)：获取公司名称

#### 2.6.4.需求摘牌模块

##### 2.6.4.1模块说明

摘牌界面所有数据不可编辑，数据由卖方挂牌信息提供

##### 2.6.4.2实现方案

Controller层：  
  
1、public ResponseData delistRequest(@RequestParam long request\_id)：摘牌功能  
  
2、public ResponseData getDelistList( @RequestParam(defaultValue = "1", required = false) int page, @RequestParam(defaultValue = "10", required = false) int limit)：交易审核员获取所有摘牌列表  
  
3、public ResponseData getDetailInfo(@RequestParam long delistId)：交易审核员获取指定的摘牌详细信息及摘牌信息  
  
4、public ResponseData getDetailInfoForUser(@RequestParam long delistId)：公司账户获取指定的摘牌详细信息及对应的挂牌信息  
  
5、public ResponseData getDetailInfoForUser2(@RequestParam long reqId)：公司账户获取指定的挂牌信息及对应的摘牌详细信息  
  
6、public ResponseData examineTransaction(@RequestParam int zpId, @RequestBody AuditOpinion opinion)：交易审核操作  
  
7、public ResponseData getDelistListFinance( @RequestParam(defaultValue = "1", required = false) int page, @RequestParam(defaultValue = "10", required = false) int limit)：获取财务用户的公司所有摘牌信息  
  
Service层：  
  
1、ResponseData delist(Delisting delisting)：摘牌功能  
  
2、Map<String,Object> listDelist(int page, int limit)：获取摘牌列表  
  
3、ResponseData getDetailInfoByZPId(long delistId)：根据摘牌ID获取挂牌摘牌的详细信息  
  
4、ResponseData getDetailInfo2(long gpId, long userId)：根据用户ID和挂牌ID 获取挂牌摘牌的详细信息  
  
5、ResponseData examine(long delistId, AuditOpinion opinion)：审核功能  
  
6、Map<String,Object> listDelistFinance(TokenProfile profile, int page, int limit)：获取摘牌列表  
  
7、ResponseData getDetailInfoForUser(TokenProfile profile,long delistId)：公司账户获取指定的摘牌详细信息  
  
8、ResponseData getDetailInfoForUser2(TokenProfile profile, long reqId)

#### 2.6.5.站内信模块

##### 2.6.5.1模块说明

该模块包含用户查看系统发布的消息以及针对收到的消息进行的一些操作

##### 2.6.5.2实现方案

Controller层：

1、public Map<String, Object> getMyMsgList(@RequestParam(defaultValue = "0", required = false) Integer page, @RequestParam(defaultValue = "10", required = false) Integer limit) 获取个人消息列表

2、public ResponseData msgDetail(@PathVariable("msg\_id") Integer msgId) 获取详细消息

3、public ResponseData markAsRead(@RequestBody HashSet<Long> ids) 将消息设为已读

Service层：

1、boolean send(Msg msg) 发送消息

2、Map<String, Object> getMsgByToId(long userId, int page, int limit) 根据目标用户获取消息

3、List<Msg> getMsgByFromId(Long userId) 根据来源用户获取消息

4、Msg getMsgDetail(long msgId, long userId) 根据消息ID与用户ID查询消息具体内容

5、boolean markAsRead(Set<Long> ids, long userId) 将消息标记为已读

#### 2.6.6保证金支付模块

##### 2.6.6.1模块说明

该模块界面内信息除了投标量可编辑之外，其余所有数据不可编辑

##### 2.6.6.2实现方案

Controller层：

1、public ResponseData showInformation(  
 @RequestParam(required = false) Long gpid,  
 @RequestParam(required = false) Long zpid) 展示用户财务信息

2、public ResponseData SecurityPayment(@RequestBody double number, @PathVariable String requestId) 缴纳保证金

Service层：

1、Map<String,Object> getFinanceById(Long userId, long id, String type) 获取用户财产信息

2、Boolean setDeposit(Long userId,String requestId,double number) 缴纳保证金

#### 2.6.7用户信息模块

##### 2.6.7.1模块说明

该模块包含用户注册、登录、修改信息、企业资质文件上传等功能

##### 2.6.7.2实现方案

Controller层：  
  
1、public ResponseData register(@RequestBody User user)注册新用户（平台外）  
  
2、public ResponseData login(@RequestBody User user)用户登录  
  
3、public ResponseData complete(@RequestBody CompanyInformation info)用户完善企业信息  
  
4、public ResponseData uploadFile(@RequestPart("file") MultipartFile file, @RequestParam CertType type)企业上传资质文件  
  
5、public ResponseData openFinancialAccount(@RequestBody FinanceProperty finance)生成企业财务账户表和财务用户功能  
  
6、public ResponseData getUserInfo()获取登录用户的个人信息  
  
7、public ResponseData getUserFullInfo()获取企业所有注册信息

8、public ResponseData newUser(@RequestBody User user)新增用户

Service层：  
  
1、public boolean isUserExist(String login, String email)检测用户是否已经存在  
  
2、public String register(User user)注册新用户  
  
3、ResponseData complete(CompanyInformation companyInformation)完善企业信息  
  
4、List<Role> getRoleList(String type)获取指定角色类型的角色列表  
  
5、ResponseData finance(FinanceProperty financeProperty)生成企业财务账务表  
  
6、Map<String,String> financeAccount()生成财务用户  
  
7、ResponseData getInfo(long id)获取登录用户的基本信息  
  
8、User getFullInfo(long id)获取企业所有注册信息

9、public String newUser(User user)创建用户

#### 2.6.8资金预存模块

##### 2.6.8.1模块说明

该模块包含展示买方信息、提交数量以及转账凭证文件上传功能

##### 2.6.8.2实现方案

Controller层：

1、public ResponseData basicInfo()获取企业信息

2、public ResponseData setQuantity(@PathVariable Double quantity) 设置提交数量

3、public ResponseData upLoad(@PathVariable Double quantity, @PathVariable String path) 提交文件

Service层：

1、List<FinanceProperty> getFinInfo(int id)获取财产信息

2、boolean TransInfo(FinanceStore financeStore)输入汇款金额

3、boolean updateF(String path,Long id)提交凭证文件

### 2.7接口说明

#### 2.7.1公共分类

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [获取角色列表](https://yapi.jysafe.cn/project/12/interface/api/4) | GET/user/getRoleList/{type} | 公共分类 | 已完成 | 未设置 |

#### 2.7.2注册用户审核

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [显示公司信息](https://yapi.jysafe.cn/project/12/interface/api/50) | GET/info/{id} | 注册用户审核 | 已完成 | 未设置 |
| [提交审核意见](https://yapi.jysafe.cn/project/12/interface/api/51) | GET/info/{id}/{opinion} | 注册用户审核 | 已完成 | 未设置 |
| [通过按钮](https://yapi.jysafe.cn/project/12/interface/api/52) | GET/info/verify/{id} | 注册用户审核 | 已完成 | 未设置 |
| [未通过按钮](https://yapi.jysafe.cn/project/12/interface/api/53) | GET/info/reject/{id} | 注册用户审核 | 已完成 | 未设置 |
| [文件下载](https://yapi.jysafe.cn/project/12/interface/api/54) | GET/servlet/pic | 注册用户审核 | 已完成 | 未设置 |
| [审核信息](https://yapi.jysafe.cn/project/12/interface/api/58) | GET/info/list | 注册用户审核 | 已完成 | 未设置 |

#### 2.7.3资金预存

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [买方信息](https://yapi.jysafe.cn/project/12/interface/api/55) | GET/fin/info | 资金预存 | 已完成 | 未设置 |
| [提交数量(废弃)](https://yapi.jysafe.cn/project/12/interface/api/56) | GET/fin/updateQ/{quantity} | 资金预存 | 已完成 | 未设置 |
| [文件上传(废弃)](https://yapi.jysafe.cn/project/12/interface/api/57) | GET/fin/updateF/{path} | 资金预存 | 已完成 | 未设置 |
| [提交预存资金](https://yapi.jysafe.cn/project/12/interface/api/61) | POST/fin/store | 资金预存 | 已完成 | 未设置 |

#### 2.7.4用户

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [注册新用户](https://yapi.jysafe.cn/project/12/interface/api/7) | POST/user/register | 用户 | 已完成 | 未设置 |
| [用户登录](https://yapi.jysafe.cn/project/12/interface/api/8) | POST/user/login | 用户 | 已完成 | 未设置 |
| [用户完善企业信息](https://yapi.jysafe.cn/project/12/interface/api/9) | POST/user/complete | 用户 | 已完成 | 未设置 |
| [企业资质文件上传](https://yapi.jysafe.cn/project/12/interface/api/21) | POST/user/uploadFile | 用户 | 已完成 | 未设置 |
| [用户完善财务信息（废弃）](https://yapi.jysafe.cn/project/12/interface/api/23) | POST/user/finance | 用户 | 已完成 | 未设置 |
| [用户修改基本信息](https://yapi.jysafe.cn/project/12/interface/api/27) | PATCH/update/userinfo | 用户 | 已完成 | 未设置 |
| [获取用户基本信息](https://yapi.jysafe.cn/project/12/interface/api/28) | GET/user/info | 用户 | 已完成 | 未设置 |

#### 2.7.5管理

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [增加用户](https://yapi.jysafe.cn/project/12/interface/api/5) | POST/user/new | 管理 | 已完成 | 未设置 |

#### 2.7.6资讯

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [展示简要资讯](https://yapi.jysafe.cn/project/12/interface/api/10) | GET/news/show | 资讯 | 已完成 | 未设置 |
| [查看详细资讯](https://yapi.jysafe.cn/project/12/interface/api/24) | GET/news/detail/{newsId} | 资讯 | 已完成 | 未设置 |
| [查询资讯](https://yapi.jysafe.cn/project/12/interface/api/25) | GET/news/more/{newsTitle} | 资讯 | 已完成 | 未设置 |
| [发布资讯](https://yapi.jysafe.cn/project/12/interface/api/26) | POST/news/publish | 资讯 | 已完成 | 未设置 |
| [资讯审核](https://yapi.jysafe.cn/project/12/interface/api/33) | POST /news/audit/{type}/{newsId} | 资讯 | 已完成 | 未设置 |
| [存草稿](https://yapi.jysafe.cn/project/12/interface/api/39) | POST/news/draft | 资讯 | 已完成 | 未设置 |
| [展示待审核资讯列表](https://yapi.jysafe.cn/project/12/interface/api/63) | GET/news/showAudit | 资讯 | 已完成 | 未设置 |

#### 2.7.7订单

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [获取订单列表](https://yapi.jysafe.cn/project/12/interface/api/11) | GET/order/list | 订单 | 已完成 | 未设置 |
| [获取订单详情](https://yapi.jysafe.cn/project/12/interface/api/12) | GET/order/detail/{OrderId} | 订单 | 已完成 | 未设置 |
| [生成订单](https://yapi.jysafe.cn/project/12/interface/api/46) | POST/order/new | 订单 | 已完成 | 未设置 |

#### 2.7.8需求挂牌

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [公共：首页查询需求列表（全部，指定用户）](https://yapi.jysafe.cn/project/12/interface/api/22) | GET/request/public/list | 需求挂牌 | 已完成 | 未设置 |
| [公共：获取挂牌详细信息](https://yapi.jysafe.cn/project/12/interface/api/37) | POST/request/public/detail | 需求挂牌 | 已完成 | 未设置 |
| [用户：需求详细信息](https://yapi.jysafe.cn/project/12/interface/api/47) | GET/request/my/detail/{request\_id} | 需求挂牌 | 已完成 | 未设置 |
| [用户：新建需求（挂牌）](https://yapi.jysafe.cn/project/12/interface/api/14) | POST/request/publish | 需求挂牌 | 已完成 | 未设置 |
| [用户：修改需求信息](https://yapi.jysafe.cn/project/12/interface/api/16) | POST/request/edit | 需求挂牌 | 已完成 | 未设置 |
| [用户：查询用户自己的所有挂牌](https://yapi.jysafe.cn/project/12/interface/api/29) | GET/request/my/list | 需求挂牌 | 已完成 | 未设置 |
| [审核：待审核需求列表](https://yapi.jysafe.cn/project/12/interface/api/30) | GET/request/audit/pending | 需求挂牌 | 已完成 | 未设置 |
| [审核：获取未审核需求详情](https://yapi.jysafe.cn/project/12/interface/api/32) | GET/request/audit/detail/{request\_id} | 需求挂牌 | 已完成 | 未设置 |
| [审核：对需求进行审核](https://yapi.jysafe.cn/project/12/interface/api/31) | POST/request/audit/do/{req\_id} | 需求挂牌 | 已完成 | 未设置 |

#### 2.7.9需求摘牌

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [客户摘牌](https://yapi.jysafe.cn/project/12/interface/api/38) | POST/delist/delist | 需求摘牌 | 已完成 | 未设置 |
| [获取所有摘牌信息](https://yapi.jysafe.cn/project/12/interface/api/42) | GET/delist/all/listDelist | 需求摘牌 | 已完成 | 未设置 |
| [交易审核人员获取指定的挂牌和摘牌信息](https://yapi.jysafe.cn/project/12/interface/api/43) | GET/delist/detailInfo | 需求摘牌 | 已完成 | 未设置 |
| [交易审核功能](https://yapi.jysafe.cn/project/12/interface/api/44) | POST/delist/examine | 需求摘牌 | 已完成 | 未设置 |
| [获取公司所有的摘牌信息(公司账号和财务账号均可使用)](https://yapi.jysafe.cn/project/12/interface/api/45) | GET/delist/my/list | 需求摘牌 | 已完成 | 未设置 |

#### 2.7.10需求剩余操作

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [用户：上传需求对应合同](https://yapi.jysafe.cn/project/12/interface/api/17) | POST/request/contract/upload/{request\_id} | 需求剩余操作 | 已完成 | 未设置 |
| [用户：查看需求对应合同](https://yapi.jysafe.cn/project/12/interface/api/18) | GET/request/contract/{request\_id} | 需求剩余操作 | 已完成 | 未设置 |
| [用户：合同文件下载](https://yapi.jysafe.cn/project/12/interface/api/34) | GET/request/contract/file/{req\_id} | 需求剩余操作 | 已完成 | 未设置 |

#### 2.7.11站内信

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [用户个人消息列表](https://yapi.jysafe.cn/project/12/interface/api/6) | GET/message/myMsg | 站内信 | 已完成 | 未设置 |
| [查看消息内容](https://yapi.jysafe.cn/project/12/interface/api/19) | GET/message/detail/{msg\_id} | 站内信 | 已完成 | 未设置 |
| [批量标记已读](https://yapi.jysafe.cn/project/12/interface/api/20) | POST/message/markAsRead | 站内信 | 已完成 | 未设置 |

#### 2.7.12保证金支付

| 接口名称 | 接口路径 | 接口分类 | 状态 | tag |
| --- | --- | --- | --- | --- |
| [展示用户财产信息](https://yapi.jysafe.cn/project/12/interface/api/40) | GET/payment/show | 保证金支付 | 已完成 | 未设置 |
| [输入投标量](https://yapi.jysafe.cn/project/12/interface/api/41) | POST/payment/number/{requestId} | 保证金支付 | 已完成 | 未设置 |

### 2.8.开发与运行环境

#### 2.8.1.硬件环境

x86服务器、Lighthouse

#### 2.8.2.软件环境

* 开发工具

Linux, Docker, MySql, Tomcat, NodeJs, Spring, HBuilderX, Intellij IDEA 等

* 基础版本 无

## 功能设计

### 3.0整体界面展示

图形用户界面, 应用程序

描述已自动生成

### 3.1账户管理：

#### 3.1.1登录

图形用户界面, 应用程序

描述已自动生成

功能点：

* 输入用户名和密码后点击登录即可登录。
* 点击注册后跳转至注册界面

#### 3.1.2注册

图形用户界面, 文本, 应用程序, Teams

描述已自动生成

功能点：

* 该注册页面为第一次注册，需要填写注册人简要内容；
* 用户名、密码、确认密码、邮箱、公司均为字符串格式
* 身份选择为单选框，仅能选择供应商或采购商
* 不勾选“使用条款和隐私策略”无法提交信息

#### 3.1.3完善信息

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

中度可信度描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

功能点：

* 该次属于第二次“注册”：当第一次注册通过审核之后会要求进行信息完善
* 信息完善界面不可更改企业类型、企业邮箱
* 信息完善要求上传营业执照、税务登记证等资质证件图片，每张图片大小不得超过500kb，仅支持jpg/png格式
* 完善财务信息需要填入财务邮箱、汇款单位名称、开户银行、银行账号以及上传开户许可证的图片。图片大小不得超过500kb，仅支持jpg/png格式

#### 3.1.4新增用户

图形用户界面, 文本, 应用程序

描述已自动生成

功能点：

* 限定超级管理员权限查看、编辑
* 可编辑用户名、密码、真实姓名、邮箱、联系方式以及角色
* 角色使用单选框从：超级管理员、咨询编辑员、咨询审核员、咨询维护员、注册用户审核员和交易审核员中选择

### 3.2资讯

#### 3.2.1资讯列表

图形用户界面

中度可信度描述已自动生成

图形用户界面, 应用程序

描述已自动生成

功能点：

* 自定义每页资讯条数
* 前往下一页、回到上一页
* 跳转指定页面
* 根据输入内容查找标题内包含该内容的资讯
* 点击列表内资讯查看详细内容

#### 3.2.2资讯编辑

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 应用程序, Teams

描述已自动生成

功能点：

* 该界面可以自由输入资讯标题和咨询内容
* 点击“保存“不会将资讯提交给审核，而是存进草稿箱
* 点击发布则会将当前资讯提交审核。待审核通过之后才会发布在资讯列表
* 点击清楚则会清空当前正在编辑的资讯
* 限定咨询编辑员权限查看、编辑

### 3.3审核中心

#### 3.3.1注册审核

图形用户界面, 应用程序

描述已自动生成

功能点：

* 该列表会展示当前请求的注册审核
* 点击审核请求则会跳转至审核具体请求的页面
* 限定注册审核员权限查看、编辑

#### 3.3.1咨询审核

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

功能点：

* 咨询审核人员可以在待审核资讯列表里选取资讯进行审核
* 对于带审核咨询可以进行“驳回”操作或“通过”操作
* 若进行驳回操作必须写审核意见
* 通过的资讯将在资讯列表里展示
* 驳回的资讯将返回至资讯编辑人员的草稿箱，并标注“驳回”和附带审核意见，同时系统将会向该资讯编辑人员发送消息通知

### 3.4交易中心

#### 3.4.1公共挂牌列表

图形用户界面, 应用程序

描述已自动生成

功能点：

* 该列表会展示通过审核的挂牌信息
* 点击某一条挂牌信息则会跳转至该挂牌信息的详细内容界面查看内容或进行摘牌操作
* 可自由选择分页信息条数
* 可跳转至指定页码

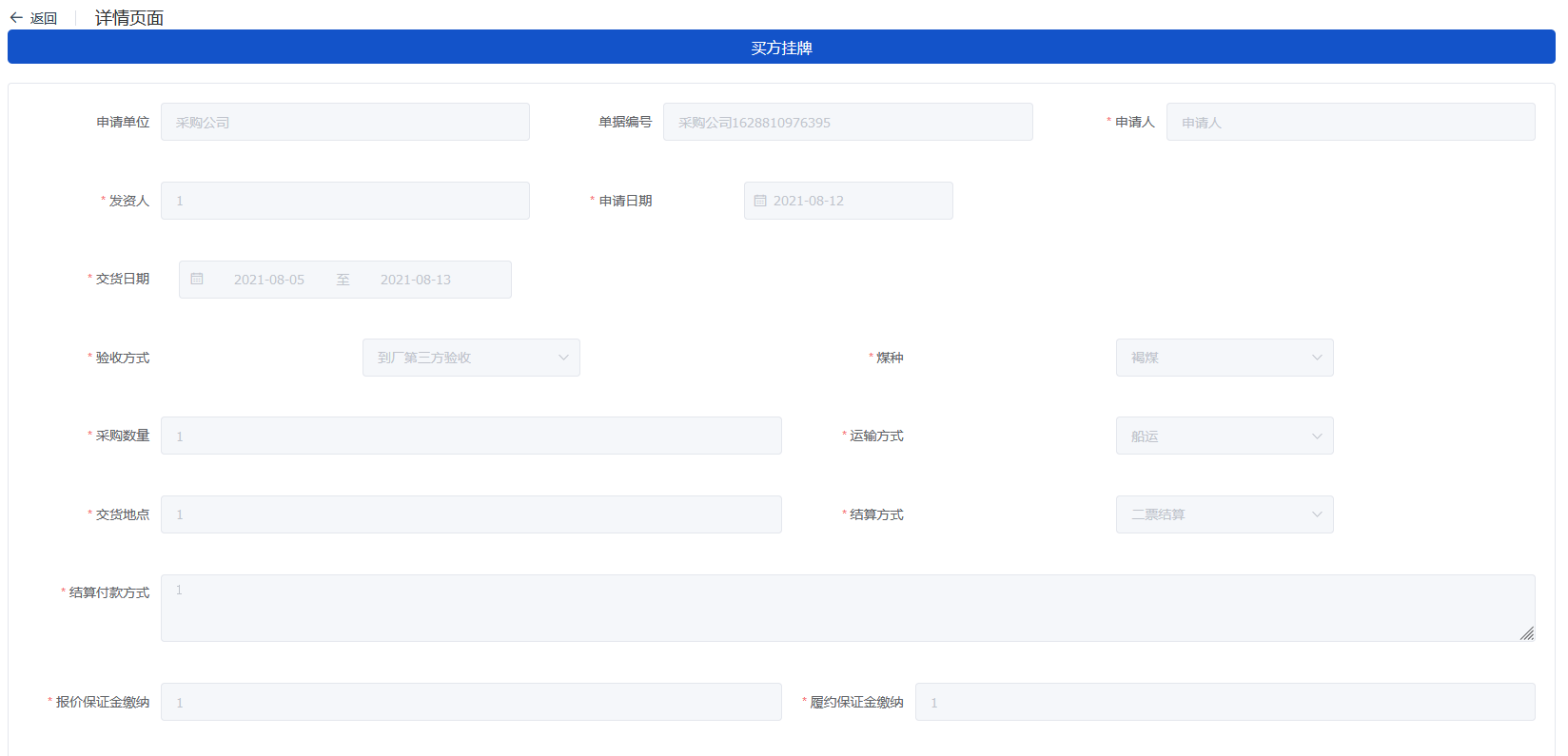
#### 3.4.2本公司挂牌列表



功能点：

* 供应商或采购商登陆人员可选择新建挂牌信息
* 限定供应商或采购商用户权限查看或编辑
* 列表基本功能同公共挂牌列表
* 查看内容仅限本公司挂牌信息方便查阅

#### 3.4.3编辑挂牌信息



图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

功能点：

* 该界面仅限挂牌用户编辑，摘牌用户不可编辑
* 该挂牌项目被摘牌成功后不可再次编辑

#### 3.4.4本公司摘牌列表



功能点：

* 列表基本功能同公共挂牌列表
* 限定供应商或采购商用户权限查看或编辑
* 查看内容仅限本公司摘牌信息，方便查阅

#### 3.4.5供应商新建挂牌

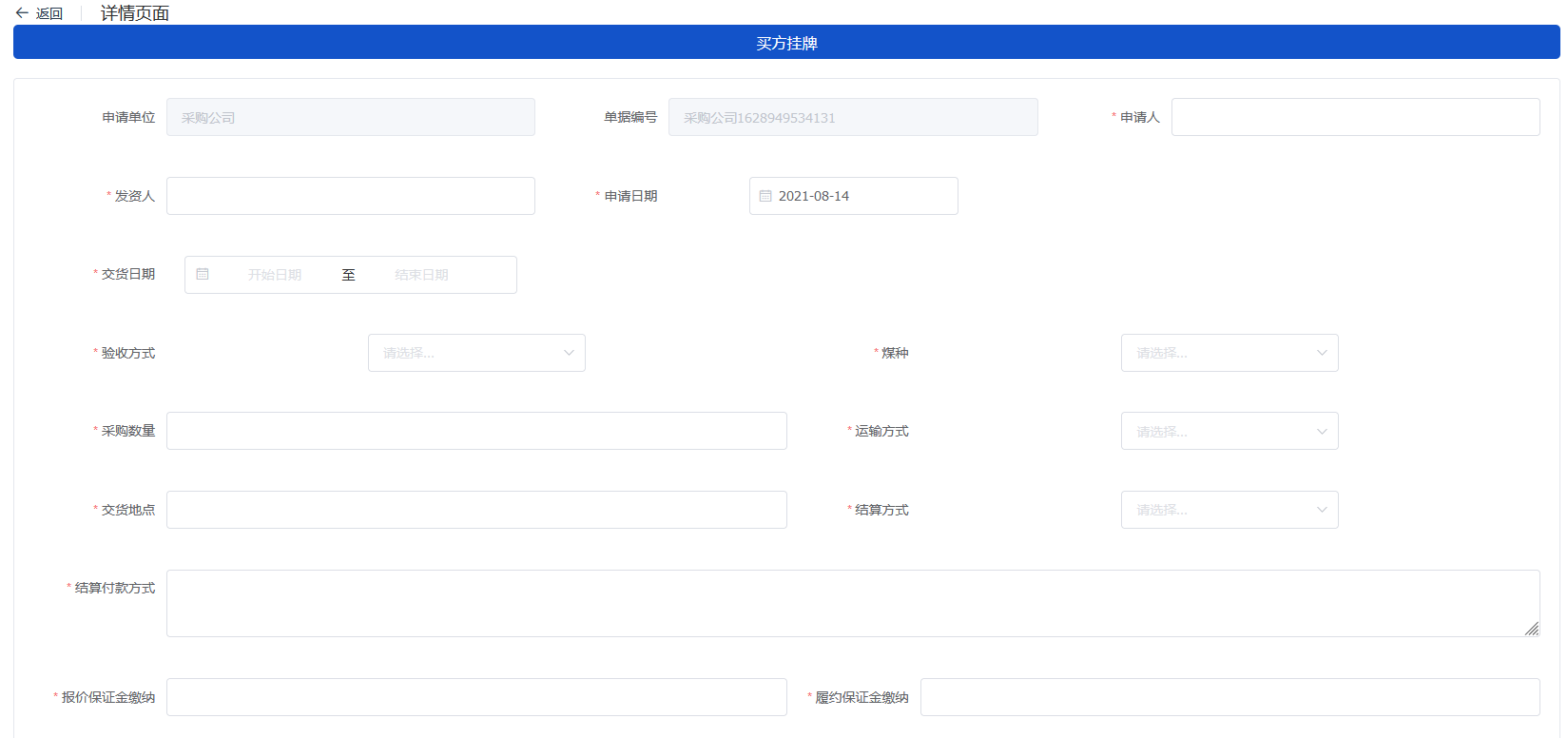
图形用户界面, 应用程序

描述已自动生成

功能点：

* 点击提交将挂牌信息送至审核
* 点击保存将挂牌信息存入草稿箱
* 仅限供应商操作人员权限进行查看、编辑

#### 3.4.6采购商新建挂牌



图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

功能点：

* “申请单位”框自动填写登录用户所属公司名称，且不允许用户修改
* “申请日期”框自动获取当前日期，且不允许用户修改
* 煤种信息中，煤炭分类可选值为：烟煤、无烟煤、褐煤、贫瘦煤、其他；商业分类可选值为：原煤、混煤、中煤、煤泥、其他
* 运输方式可选值为：火车、汽车、船运、其他
* 结算方式可选值为：一票结算、二票结算、铁路大票、煤款税票、其他
* 验收方式可选值为：到厂验收、到厂第三方验收、港口第三方验收
* 报价保证金缴纳：若为“不要求”，则后续价格框为不可编辑
* 履约保证金缴纳：要求同上，但在当前版本中不使用
* 煤质要求：收到基低位发热量、收到基全硫、全水分、空干基全硫、干燥无灰基挥发分均为必填项，其他各项可选
* 仅限采购商操作人员权限进行查看编辑

### 3.5财务中心

#### 3.5.1预存资金

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

功能点：

该界面除汇款金额和汇款凭证外均不可编辑

汇款凭证可上传所有图片格式的文件

## 数据库设计

## 接口设计



## 6.DFX设计

2天

### 6.1可制造性（DFM）

为了减少制造成本，提升制造流程，在完成产品的初期就对数据库，结构，方法进行设计，这样在日后开发的时候能够更加方便的完成产品

### 6.2可装配性（DFA）

保证以模块化完成程序，这样在日后测试或者是组装起来都会非常的方便。

### 6.3可测试性（DFT）

为了提高产品能观能控、故障检测与定位隔离的能力，尽量多写批注，这样在测试出问题的时候好找错

### 6.4可靠性（DFR）

降低故障发生的影响，故障发生后能尽快恢复，代码逻辑不要太复杂，不易于维护。每次都测试完了以后再上传。

### 6.5可服务性（DFS）

提高系统安装调测与维护管理能力，提高服务效率。功能设计时简化操作逻辑，使得用户对操作容易上手

### 6.6面向环保的设计（DFE）

减少毒害性和资源消耗，保护生态环境，如减少对数据库的访问次数，耗电量少，自然就更加环保了。

## 7.依赖

3天

#### 7.1升级依赖

* 升级到较新版本的Boot：最稳妥的办法
* 分别升级依赖项：可能会导致自动配置在启动时不起作用，您可以尝试通过明确排除自动配置并编写手动配置来替换自动配置。
* 注意Boot版本，并定期尝试将项目干式升级到较新的版本，但不一定要将升级版本交付生产。
* 这样就可以评估升级的风险，并估计需要进行的工作量。

#### 7.2内部依赖

* spring-boot-dependencies
* spring-boot-starter-parent
* spring-boot-autoconfigure
* spring boot官方包装的start组件
* spring-boot-starter-web
* spring-boot-starter-test
* spring-boot-configuration-processor
* spring-boot-devtools
* spring-boot-starter-security
* lombok

#### 7.3外部依赖

* mybatis-spring-boot-starter
* knife4j-spring-boot-starter
* jackson-databind
* jsch
* commons-lang3
* shaun-spring-boot-starter
* mysql-connector-java