浙 江 理 工 大 学

毕业设计 (论文)诚信声明

我谨在此保证：本人所写的毕业设计 (论文)，凡引用他人的研究成果均已在参考文献或注释中列出。 设计(论文)主体均由本人独立完成，没有抄袭、剽窃他人已经发表或未发表的研究成果行为。如出现以上违反知识产权的情况，本人愿意承担相应的责任。

声明人(签名)：

年 月 日

摘 要

食品安全是重要的民生问题之一,将直接决定着人们的生活水平和生活质量。近年来,随着我国经济的飞速发展,也带来了各类食品安全问题层出不穷,使得食品安全问题已经成为公众关注的焦点问题之一,这不仅严重危害了我国人们的身体健康,更是严重损害了我国的形象,不利于我国长远发展[1]。食品安全事件频发，三聚氰胺、染色馒头、塑化剂、地沟油屡屡发生的食品安全事件不断削弱人民对食品安全状况的信心，对我国经济发展产生了一定程度的消极影响[2]。

随着我国经济的快速发展，食品安全问题层出不穷，已经成为公众关注的焦点问题之一。食品安全问题是重要的民生问题，将直接决定着人们的生活质量和生活水平，这不仅严重危害到我国人们的身体健康，更是严重损害到我国的大国形象。

本文在对现有票据市场进行详细的需求分析之后，设计了汇票线手机APP并完成了其中的核心功能模块，包括了手机APP的首页模块，用户模块，报价模块，票据模块以及汇票助手模块，实现了从持票方出票到收票方接收票据完成交易并评价的整个业务流程模块，规范了整个票据交易流程，同时通过监管账户实现交易的安全性保证。此外，为了机构收票的便捷性，增加机构报价这一模块，针对不同类型不同交易方式的票据进行详尽的报价，保证报价信息的准确传递，同时促进了票据的流通。

汇票线手机APP前端采用Ionic和Cordova的组合，保证了APP的跨平台使用以及兼容性问题，同时采用SpringBoot和Mybatis编写后台符合RestfulAPI的接口，采用了微服务的架构，利用Docker容器对微服务进行封装并部署于阿里云之上，保障了平台的安全性以及高效性。

关键词：汇票交易，机构报价，Ionic，Cordova，阿里云，Docker，微服务，Restful API

ABSTRACT

Internet has become the most popular word, its exuberant development brings convenience for people. “Internet Finance” emerged at this moment has also become the hot topic of Financial Service Industry. At the information-expansion era of Internet Finance, the convenience and rapidity of information transmission by Internet enhances the propagation of off-line traditional financial business, the mode of Internet Inclusive Finance is developing rapidly. At the same time, as the continuous extension of note market and implementation of electronic bills, the current bill transaction system only support single transaction of bank bill, leading to the obstacle of bill circulating.

This thesis analyses existing note market detailedly, designing Draft Line APP and achieving its core modules, including homepage module, user module, offer module, bill module and draft assistant module, realizing the whole business process, namely from draft holders draw bill to draft receivers receive bill and completing evaluating each other as well as strengthening security through supervising account. In addition, to reinforce the convenience of receiving draft for organizations, adding the module of organization offering and designing distinguished offering methods in accordance with different types and transaction ways, ensuring the accurate transmission of offering information, boosting the circulating of bills.

Draft Line APP employing Ionic and Cordova in front-end, assuring cross-platform and compatibility. In addition, using SpringBoot and MyBatis to complete back-end interfaces confirming to Restful API criterion. Employing micro-service architecture and Docker to packaging these services then deploying on Aliyun, guaranteeing the security and efficiency of this platform.

Keywords: draft transaction, organization offering, Ionic, Cordova, Aliyun, Docker, Micro-service, Restful API

目 录

摘 要

Abstract

第1章 绪论 1

1.1 研究背景 1

1.2 研究目的与意义 1

1.3 研究现状 2

1.3.1 国外发展 2

1.3.2 国内市场 2

1.3.3 主要问题 3

1.4 论文主要工作 4

1.5 论文组织结构 4

1.6 本章小结 5

第2章 基础理论与相关技术 6

2.1 云计算 6

2.1.1 IaaS服务模式 6

2.1.2 PaaS服务模式 6

2.1.3 SaaS服务模式 6

2.2 容器技术 6

2.2.1 虚拟机 7

2.2.2 Docker容器 7

2.3 Hybrid App技术 7

2.3.1 Cordova 8

2.3.2 Ionic 8

2.4 数据库 8

2.5 存储 8

2.6 消息推送 9

2.6.1 JPush极光推送 9

2.6.2 SMS短信服务 9

2.7 开发模式 10

2.7.1 C/S模式 10

2.7.2 微服务系统架构 10

2.7.3 MVVM+RESTful API 10

2.8 前端技术 11

2.9 后端技术 12

2.10 本章小结 12

第3章 需求分析 13

3.1 系统整体功能概览 13

3.2 基本概念 13

3.3 首页模块 15

3.3.1 报价信息 15

3.3.2 票源信息 15

3.3.3 热门推荐 15

3.4 用户模块 16

3.4.1 资料完善 16

3.4.2 我的竞价 17

3.4.3 我的发布 17

3.4.4 机构认证 17

3.4.5 消息中心 17

3.4.6 我要充值 17

3.5 报价模块 18

3.5.1 新建报价 18

3.5.2 修改报价 18

3.5.3 关闭报价 19

3.6 票据模块 19

3.6.1 我要收票 19

3.6.2 我要出票 20

3.7 汇票助手 20

3.7.1 贴现计算器 20

3.7.2 工商查询 21

3.7.3 挂失查询 21

3.7.4 行号查询 21

3.7.5 开票日历 21

3.7.6 票面涂抹 21

3.8 本章小结 21

第4章 系统设计 22

4.1 架构设计 22

4.1.1 基于阿里云的物理架构 22

4.1.2 基于前后端分离的逻辑架构 25

4.2 数据字典 26

4.3 本章小结 45

第5章 系统实现 46

5.1 RESTful API规范 46

5.1.1 基本规范 46

5.1.2 状态码 46

5.2 前后端分离 47

5.2.1 前端层次 47

5.2.2 后端逻辑 52

5.3 基于token的安全认证 54

5.4 Restangular简化API请求 56

5.5 票据交易流程合理 58

5.6 机构报价完善 58

5.7 汇票助手便捷 59

5.8 本章小结 62

第6章 系统部署与测试 63

6.1 系统部署 63

6.1.1 APP打包 63

6.1.2 后端部署 64

6.2 系统测试 66

6.2.1 后端接口测试 66

6.2.2 黑盒测试 67

6.3 本章小结 68

第7章 成果与展望 69

7.1 研究总结 69

7.2 不足之处 69

7.3 后续展望 70

参考文献 71

致 谢 73

1. 绪论

线下电票交易具有交易对手的信用风险，交易的安全问题，汇票线APP通过建立全新的模式，利用“移动互联网+票据”的方式减少风险的产生。通过信息直接对接，减少中间环节，有效地降低了票据贴现的成本；另外的一个重要问题就是如何解决原来工具碎片化的问题，例如贴现计算器，挂失查询，工商查询等。汇票线APP通过一揽子汇票工具可以有效解决上述问题。汇票线APP所要极力提供的是一个提供票据信息发布的平台，提供了票据贴现及电票交易的平台。我们有理由相信在未来的某一天，平必将受到行业内各大平台及的关注及推荐。汇票线必将实现票据交易触手可及的愿景，低成本有效的解决企业票据融资难题。

## 研究背景

随着我国经济的快速发展，食品安全问题层出不穷，三聚氰胺、染色馒头、塑化剂、地沟油屡屡发生的食品安全事件不断剥削人民对食品安全状况的信心，已经成为公众关注的焦点问题之一。食品安全问题是重要的民生问题，将直接决定着人们的生活质量和生活水平，这不仅严重危害到我国人们的身体健康，更是严重损害到我国的大国形象。国家鼓励食品生产经营者采用信息化手段采集、留存生产经营信息, 建立食品安全追溯体系。但是, 我国在食品安全追溯体系建设方面, 还处于起步阶段。我国急需要建立一个完整有效的食品安全追溯系统，实现食品可追溯，从源头上保障消费者的合法权益。

## 研究目的与意义

现有的汇票交易系统仍存在一些问题，主要体现在市场较为集中，交易过程中风险较大，交易制度不规范以及流通性较差，汇票信息不能即时获取，时效性差，技术陈旧等。

通过现在成熟的云计算服务，底成本的跨平台Hybrid App技术以及加密保护技术建立一个汇票交易App有望解决以上问题，针对现在汇票系统局限较大的需求，通过建立一个具有规范化交易流程且支持所有类型汇票交易的App，真正做到汇票的移动化即时化交易，使汇票市场的更加健康发展。

## 研究现状

### 国外发展

从20世纪90年代开始，许多国家为了推动食品安全管理水平，陆续建立了追溯制度，美国、欧盟和日本是较早开展食品追溯标准化工作的国家和地区，有健全的法律法规、配套的组织执行机构，已经建立起了以预防、控制和追溯为特征的食品安全追溯体系，推动食品安全得到全程监控。

1. 美国食品安全追溯体系

美国食品安全追溯体系包括多个政府部门和民间机构，部门和机构在解决食品安全问题上各司其职，进行制定食品质量安全标准、实施食品质量安全监管、进行食品质量安全教育等，形成一个对食品质量安全实行“从生产到消费”的全程监管体系。上有严格的监管，下有激烈的竞争，生产商和销售商如果被查出有食品安全问题，都会被处罚，并且需要花费巨额费用来召回有问题的食品。

1. 欧盟食品安全追溯体系

欧盟食品安全追溯体系的最大特点是，通过法律强制要求所有食品生产经营者建立可追溯系统。体系对食品的生产、加工、储存、运输、销售过程所有环节相关的单位规定了具体的责任，对来源进行追溯和追查，以便快速准确地找到食品安全的问题环节。

1. 日本食品安全追溯体系

日本自2003年开始在全国推行食品可追溯制度，从日本农林水产省决定建设牛的追溯体系，要求肉牛业实施强制性的可追溯制度，到日本农林水产省宣布要建立大米的可追溯体系，形成农林水产省、厚生劳动省和食品安全委员会三方协同制衡的食品安全监管体系。

### 国内发展

我国食品安全可追溯系统起步较晚，但也取得了长足的发展。近年来，我国已经制定了一些食品安全有关的法律法规，2015年修订的《食品安全法》[9]第四十二条明确规定了：“国家建立食品安全全程追溯制度。食品生产经营者应建立食品安全追溯体系，保证食品安全可追溯[10]。”国家鼓励食品生产和经营者采用互联网信息化手段采集、留存食品生产经营信息，建立食品安全追溯体系[11]。

我国关于食品溯源体系的研究始于2002年[12]。在研究和实施过程中，逐步制定了一些相关的标准指南以及研究成果。

2002 年，北京市首先建立了食品安全追溯制度。2004 年 2 月上海市建立了“上海食用农副产品质量安全信息平台”，该平台能通过网络查询和条码技术对农副产品的生产过程进行监控。2005 年 8 月福建省建立了省内首个肉类食品追溯系统，消费者通过该系统可以参与监督产品的生产企业的各个环节，让企业的日常生产在食品安全相关的法律法规监管之下[13]。

2010 年，国务院食品安全委员会成立[14]，对国务院各个部门的职责进行了具体的确认。农业部门对初级农产品的生产进行监管，质检部门对产品生产企业和产品加工质量进行监管，商务部门负责产品在市场上的流通，食品药品监管部门负责综合监督。

2015 年中国食品信息追溯大会在江苏南京召开，大会对贵州茅台集团进行了表彰。茅台酒股份公司率先对集团旗下酒类产品进行了全过程的追溯。茅台集团利用二维识别码对酒的溯源信息进行标记，扫码即可查看该瓶酒的溯源信息。同时还有 RFID 识别，可用根据扫码数据，对产品地域分布，串货情况进行多种分析。

### 主要问题

1. 管理对象单一

现在的汇票交易市场，尽管从数据上分析，电子汇票从09年央行投产后，参与者已达近400家[11]，但从拥有电子票据交易系统的机构来看，主要都是集中在中大型企业[12]，例如大型企业集团财务公司、国有和股份制商业银行和外资金融机构，并没有完全渗透到所有金融机构或者所有企业中去。

当前已有的可追溯系统是针对某一具体类别食品

1. 交易风险

电子票据信息在网络上的交易公开化程度较高，加上使用的是公共的通信通道，汇票交易面临诸多安全隐患[13]。但众所周知，安全性在金融市场的要求极高。我们要解决以下问题：

1. 如何做到票据信息的安全性，即只让票据的双方以及监管人员看到票据信息；
2. 如何保证在票据交易过程中的消息的精准传递问题，防止票据交易双方接收到不正确的交易信息；
3. 信息的不可否认性，即发出后不可否认。

这三点必须要求本身交易平台十分安全，并且票据中介等相关机构足够可信。

除此以外，电子票据缺乏规范也是造成高风险的原因之一，由于我国各个银行间未有统一的电子票据样式[14]，交易十分不规范，缺乏同一的标准，认证方式，也降低了交易的效率。

1. 流通局限

与传统的纸质票据相比，电票流通性明显提高，由于电票的基本性质即决定了它仅仅是以电子数据的形式在计算机内存储，在网络中传输的。但是现在基于行内网上银行的电子汇票系统决定了电子票据仅仅停留在本银行内，该银行的汇票的信息流通很难传播到外界，各个银行的电子票据信息不能互通，针对上述问题，中国外汇中心也开通了“中国票据网”，但因种种原因现只有一人在从事日常维护工作，因为整个汇票市场的信息流通局限性很差。

但是针对汇票本身对于高流通性的要求，需要有一个支持所有类型电子票据交易的应用产生，不断促进企业银行，货币市场，资本市场不断良性互动[15]。

## 论文主要工作

食安云企业端是对食品生产和食品经营企业进行食品生产、食品流通管理的平台。具体功能包括企业信息管理、产品管理、一票通管理、报备管理等功能。企业可以通过企业信息管理增添企业，实现一个账号管理多家企业；在产品管理中添加产品，在交易对象管理中添加供货企业和购货企业；在电子一票通中创建采购票据和销售票据；在报备管理中，生产企业需要生产报备，流通企业需要电子一票通报备，电子一票通报备将数据串成一条数据链，上报给监管端，实现食品供应链的追溯。

食安云企业端前端至少包括以下功能：

1. 用户注册、登录、信息完善；
2. 企业信息管理，产品管理，交易对象管理，从业人员管理；
3. 电子一票通的采购录入和销售录入；
4. 企业状态、产品生产和一票通报备；
5. 销售台账和生产台账；
6. 充值，消费记录，开具发票等费用功能；
7. 用户手册和意见反馈；

## 论文组织结构

本文将按如下章节依次展开，见表 3‑1

表 3‑1论文组织结构表

|  |  |  |
| --- | --- | --- |
| 章节 | 名称 | 内容简介 |
| 1 | 绪论 | 介绍食安云企业端研究现状及本文研究目的与意义 |
| 2 | 基础理论与相关技术 | 介绍本文涉及到的基础理论与相关的开发技术 |
| 3 | 需求分析 | 介绍食安云企业端的需求分析，主要展示各功能模块用例图 |
| 4 | 系统概要设计 | 介绍食安云企业端的概要设计，给出架构设计 |
| 5 | 系统详细设计 | 介绍食安云企业端的详细设计，包含流程图与几个关键设计 |
| 6 | 系统实现 | 介绍食安云企业端的实现，以若干个重难点为例 |
| 7 | 系统部署与测试 | 介绍食安云企业端的云端部署和相应的测试方法 |
| 8 | 结论与展望 | 对食安云企业端的总结和分析 |

## 本章小结

本章先简要描述了课题研究的背景、目的与意义，然后通过查阅文献、实际试用等方法对目前国内外的一些相似系统进行了对比，得出了其中的主要矛盾并加以评述，最后说明了本论文的主要工作，及各章节的组织顺序。

1. 基础理论与相关技术

## AngularJS数据双向绑定

AngularJS是近些年流行的前端语言，通过新的属性和表达式扩展了HTML，克服了HTML在构建应用上的不足。Angular声明式的模板功能强大，自带了丰富的指令，是一个完善的前端MVVM框架，实现了数据双向绑定、过滤器、依赖注入、模块化、服务等功能。AngularJS实现数据双向绑定分别表现在与浏览器交互和与用户交互。

### AugularJS与浏览器交互

当浏览器访问到AngularJS脚本时，浏览器依次进行了如下操作：

1. 加载html，对DOM进行解析；
2. AngularJS等待DOMContentLoaded事件的触发；
3. AngularJS寻找ng-app指令，找到应用程序边界;
4. 在ng-app中指定的模块配置$injector;
5. 使用$injector创建$rootScope和$complie服务；
6. 使用$complie服务编译DOM并把它链接到$rootScope上；
7. ng-init指令对$scope的变量进行赋值；
8. 对表达式{{变量}}进行替换。

整个流程图如图2-1所示。

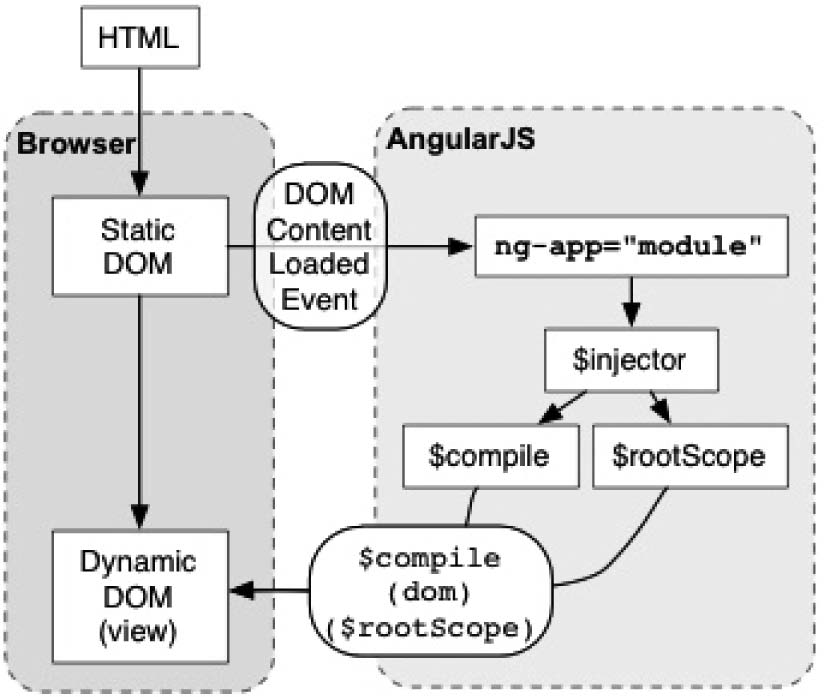


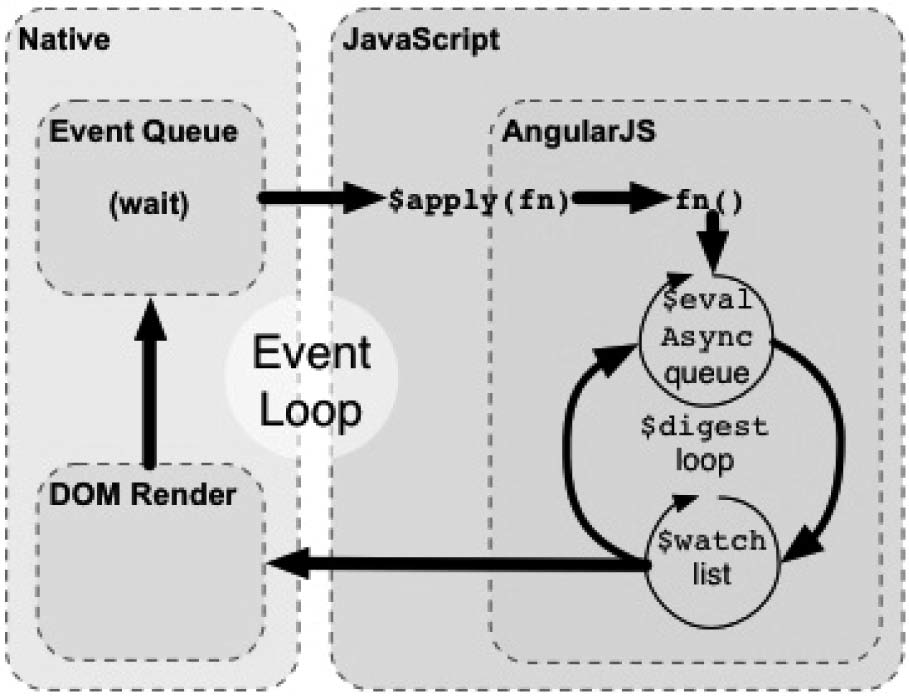
图 2‑1 AngularJS与浏览器的交互图

### AugularJS与用户交互

AngularJS与用户交互过程，整体上分为三个阶段：

1. 浏览器的事件回路一直处于等待事件触发的状态，这些事件包括用户的交互操作、网络事件和定时事件。
2. 一旦有事件触发浏览器的事件回路，就会进入Javascript的context中，使用回调函数对DOM进行修改。
3. 回调函数执行完后，浏览器根据新的DOM进行重新渲染。

AngularJS与用户的交互流程如图2-2所示。



2‑2 AngularJS与用户的交互图

AngularJS适用于编程大型应用，声明式模板，扩展了HTML语法，更清晰、快捷构建应用组件，利用依赖注入和数据绑定，减小了开发过程中的代码冗余。

## MVVM设计模式

MVVM设计模式，将软件的架构分离为Model、View和ViewModel，将数据层、业务逻辑层以及表现层分离开来，达到“高内聚，低耦合”的效果，有效地减少了代码工作量，易于理解，条理清晰，提高了软件开发和维护的效率。MVVM分为3个模块，Model模块、View模块和ViewModel模块。

### Model模块

Model模块，是MVVM设计模式中用于存放数据和业务逻辑的类，以数据模型的方式将数据呈现出来，这是一个项目中最有价值的部分，记录了整个项目的数据以及数据之间的联系，但是用户无法直接对其进行操作。

### View模块

View模块，是MVVM设计模式中用来定义用户界面和存放界面逻辑的部分，通过美观的界面以及交互式响应提高使用者的人机交互体验。用户在界面上的操作会对相应的数据进行修改，将用户需要的数据展现出来。

### ViewModel模块

ViewModel模块是MVVM设计模式的核心，它是View的抽象，负责View和Model之间的信息交换。ViewModel起到了承上启下的作用，对于Model层，他提供了数据交互的接口，对于View层，他可以对View进行同步更新，对View的操作进行响应。

MVVM设计模式原理如图2-3所示。



图 2‑3 MVVM设计模式

## 又拍云存储

又拍云存储是面向互联网应用的对象存储服务，具有使用简单、高可靠、高安全等特性，又拍云存储单存储集群支持 PB 级容量，单数据中心支持 EB 级容量，全网轻松应对 ZB 级的异构化数据，实现海量云存储。食安云企业端在完善企业信息、产品管理、生产报备、一票通报备等多个功能模块中都需要图片上传，使用又拍云上传图片，可以做到灵活上传，简单易用，提高工作效率，降低开发成本。

## 短信服务推送

SMS短信（Short Messaging Service）即短消息业务，是现在最被广泛使用，且被第三方厂商支持更多的短消息业务。在本项目中我们采用阿里云作为我们的短信云通信服务平台，支持验证码，短信通知等服务，保证了账户的安全性以及消息通知的即时性。

## MetronicUI库

Metronic是一个基于Bootstrap 3.\*设计的多用途的响应式后台管理模板，被广泛应用于各种web应用。Metronic拥有干净、直观、平滑的地铁风格设计，让项目看起来简洁且友好。Metronic有着丰富的UI组件和插件，而且能够和谐的工作在Chrome、FireFox、IE等主流浏览器、手机和平板上。

## 本章小结

本章主要介绍在食安云企业端的开发过程中所使用的技术：使用AngularJS实现前端MVVM设计模式，使用又拍云上传图片，同时采用阿里云短信服务做短信服务推送，结合基于Bootstrap的前端MetronicUI框架进行开发，提高开发效率，提升用户体验。

、

# 需求分析

## 系统整体功能概览

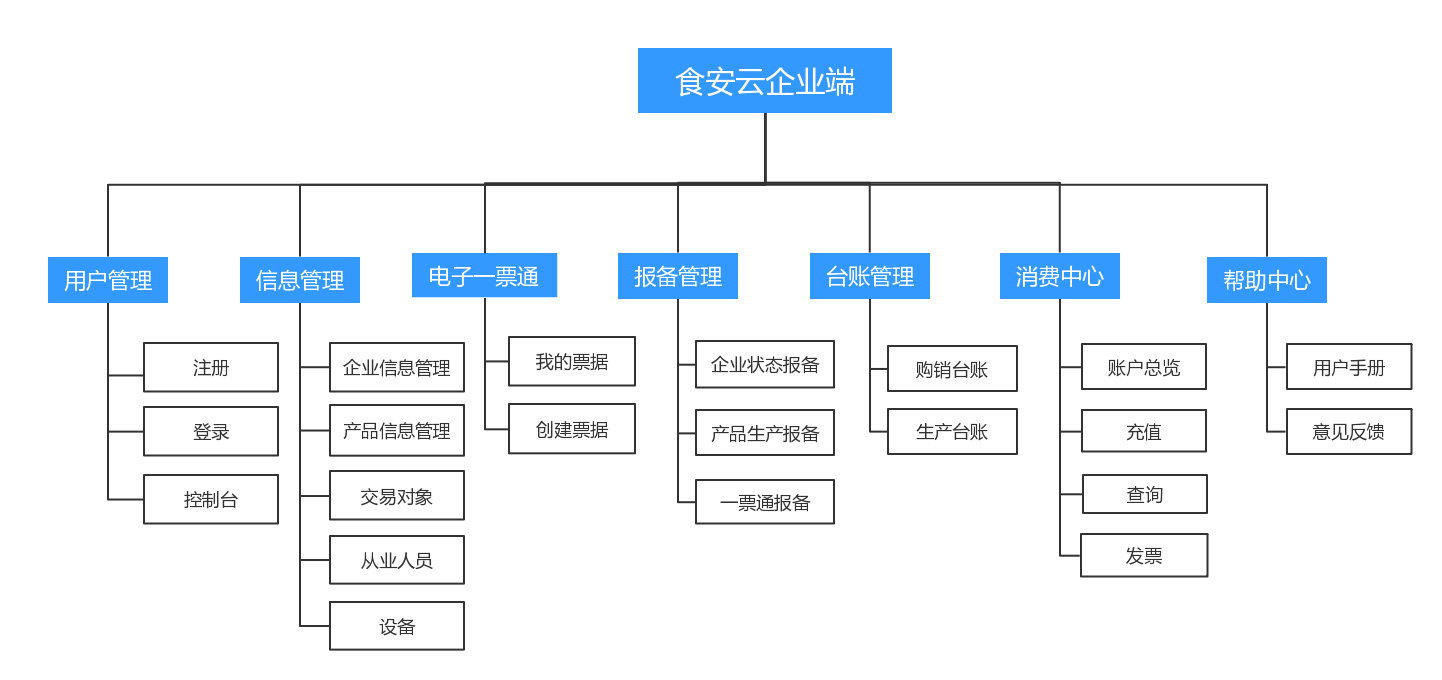


图 3‑1 系统总体功能模块图

本项目的目标为，提供一个安全、可靠、便捷的食品安全企业报备平台，实现企业信息管理、电子一票通管理、报备管理、台账管理等业务功 能，从而为监管端提供报备数据实现食品可追溯。

根据严谨的需求分析和整理工作得出本项目的总体功能模块图如图3-1所示，系统分为七大功能模块，以下我们将围绕这几个功能模块进行分析。

## 基本概念

1. 营业执照：是工商行政管理机关发给工商企业、个体经营者的准许从事某项生产经营活动的凭证。其格式由国家工商行政管理局统一规定。
2. 生产许可证：国家对于具备某种产品的生产条件并能保证产品质量的企业,依法授予的许可生产该项产品的凭证。
3. 经营许可证：国家对于具备某种产品的经营条件并能保证产品质量的企业,依法授予的许可经营该项产品的凭证。
4. 子账号：用户分配给他的多家企业中一家企业的账号，用于管理该企业。
5. 预包装产品：预先定量包装或者制作在包装材料和容器中，有标准条形码的产品。
6. 散装产品：食物成品没有进行包装即进行销售的产品。
7. 交易对象：跟用户的企业发生交易关系的企业，交易对象分为两类，一类为供应商，一类为客户。
8. 时间戳：是自1970年1月1日至当前时间的总秒数。
9. 中国物品编码中心：统一组织、协调、管理我国商品条码、物品编码与自动识别技术的专门机构。
10. 产品标准条形码：由中国物品编码中心规定的用于唯一标识预包装产品的条码。
11. 产品自编条形码：企业自定义唯一标识散装产品的条码，由企业营业执照号和创建产品时间的时间戳组成。
12. 标准企业库：入驻平台的企业以及其交易对象组成的企业信息库。
13. 索证：向交易对象索要营业执照和许可证，用户的交易对象中没有证照的需要进行索证，以证明该交易对象是合法的。
14. 从业人员：从业人员是指在本单位工作并取得劳动报酬的全部人员数，包括在岗职工、再就业的离退休人员以及在各单位中工作的外方人员和港澳台方人员、兼职人员、借用的外单位人员和第二职业者等，但不包括离开本单位仍保留劳动关系的职工。
15. 一票通：食品销售单位的销货凭证与食品零售单位的进货凭证统一格式和内容，合二为一的票据。
16. 待领取票据：是指用户的交易对象比用户先制作的一票通，系统会将此票据发送给用户。
17. 退回的票据：是指用户创建的票据在交易对象领取后，发现有错误，被对方退回的票据。
18. 拍照票据：是指将票据以拍照的形式报备的数据。
19. 采购报备：生产企业对原材料、防腐剂的采购信息进行报备；食品经营企业对采购产品的报备。
20. 销售报备：生产企业、食品经营企业对产品销售信息的报备。
21. 拍照报备：对采购单或者销售单拍照上传报备。
22. 台账：流水账。台账包括购销台账和生产台账。其中购销台账是指进货台账和销货台账，记录了食品名称、条形码、规格、数量、供货（购货）者名称及联系方式、交易类型、交易日期等信息。生产台账记录了食品的名称、条形码、报备时间、生产批次等信息。

## 用户模块



图 3‑2 用户模块用例图

### 注册

用户在平台上没有账号的时候可以自行注册，注册项有手机号、短信验证码、密码、确认密码、协议许可。企业用户需要准确填写营业执照信息和企业许可证信息，包括执照编号、执照图片、企业名称、主体类型、法人姓名、联系方式、许可证类型、生产地址、许可证编号、证照图片等，系统可以根据执照编号查询标准企业库填充部分信息内容。

### 登录

输入注册时绑定的手机号码和密码，如果有多家企业，则需要选择进入哪家企业，进入控制台。

### 控制台

控制台显示企业名称、用户余额、剩余天数；显示最后报备时间、台账总数、未报备电子票据数和待领取票据数以及可以查看其详情；另外有电子一票通、领取交易票据、电子一票通报备和产品生产报备的快捷入口；显示消息，消息分为个人消息和系统消息，包括已读和未读消息，可以进行查看详情；可以进行修改密码、修改绑定手机号、切换企业和退出系统。

## 信息管理模块

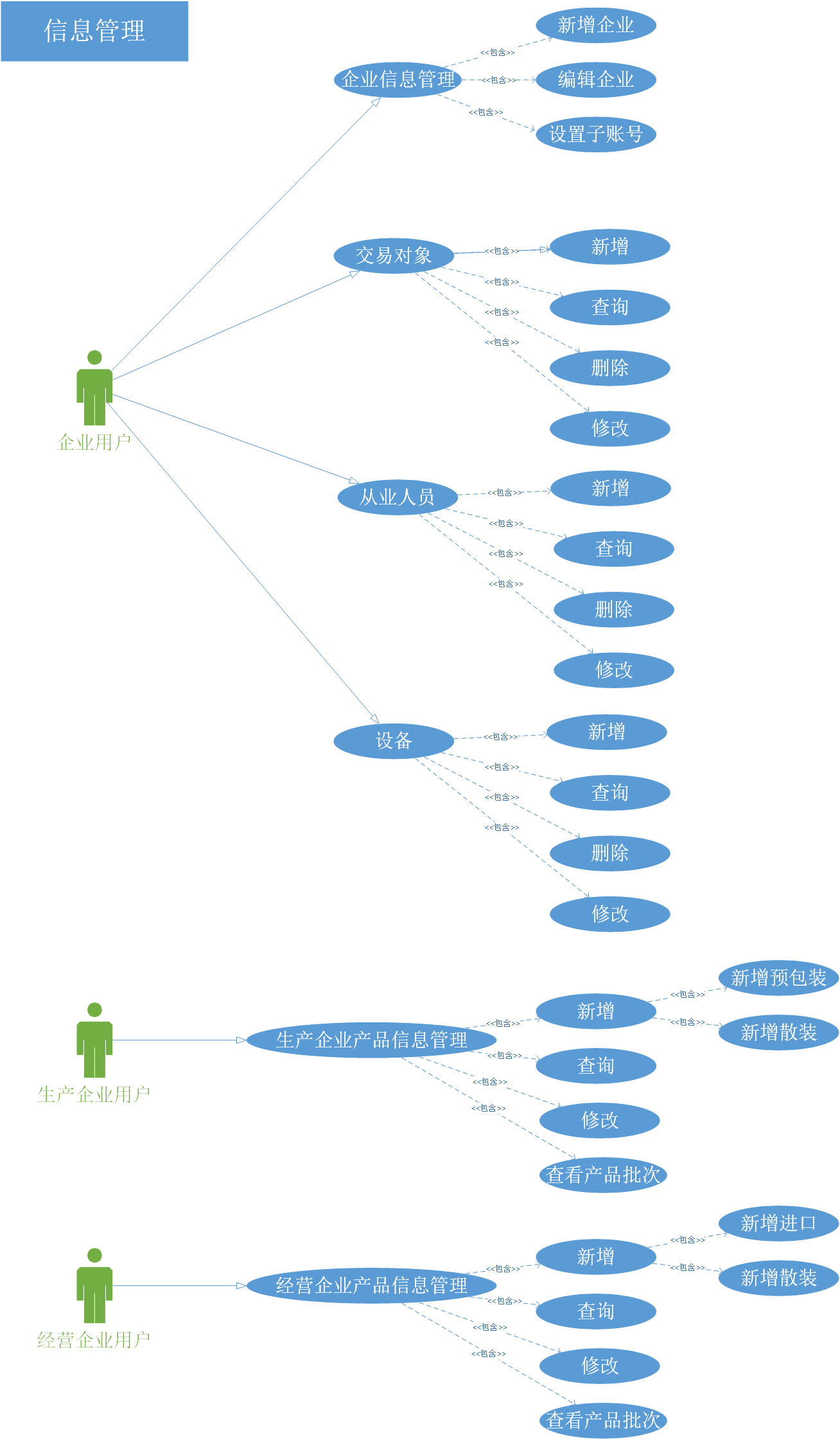


图 3‑3 信息管理模块用例图

### 企业信息管理

企业用户可以通过企业信息管理对企业信息进行新增、编辑、设置子账号，来实现一个企业管理多家企业的工作。新增企业之后会进入审核状态，在未审核通过之前不能对该企业信息做任何业务操作，并且不能切换到该企业。

### 产品信息管理

企业用户可以通过产品信息管理进行查询和管理本企业生产或经营的食品，通过产品名称、条形码、产品类型和产品分类精确查询产品，生产企业可以新增预包装产品和散装产品，经营企业可以新增进口产品和散装产品。预包装产品内容包括条形码、食品分类、产品名称、规格型号、净含量、保质期等信息，散装产品内容包括产品名称、产品自编码、规格型号、保质期、食品分类等。新增预包装产品时，可以通过条形码从物码中心中找出对应的产品信息自动填补在表单中。

### 交易对象管理

企业用户可以对交易对象进行增、删、查、改，新增的交易对象优先从标准企业库中选择，当标准企业库中查询不到用户准备添加的交易对象的时候，用户可新增一个企业，作为自己的交易对象。用户通过单位名称、法人姓名和联系号码可以精确查询交易对象。用户可以对已有的交易对象进行删除，如果在报备数据里出现过，该交易对象不可删除。用户的交易对象中没有证照的需要对其进行索证，以证明该交易对象是合法的，才可以进行交易。

### 从业人员管理

用户需要管理本企业需要执证上岗的人员，如执有健康证人员，安全管理人员等。对从业人员进行增删改查，增添需要上传证件照片。从业人员信息包括姓名、身份证、身份证照片、证件类型、证件照片和联系电话等。用户通过姓名和到期日期精确查询从业人员。

### 设备管理

对企业生产加工所需的设备进行管理，设备信息包括设备名称、设备编号、设别状态和设备用途等，可以进行增添、更改状态。

## 电子一票通模块\*

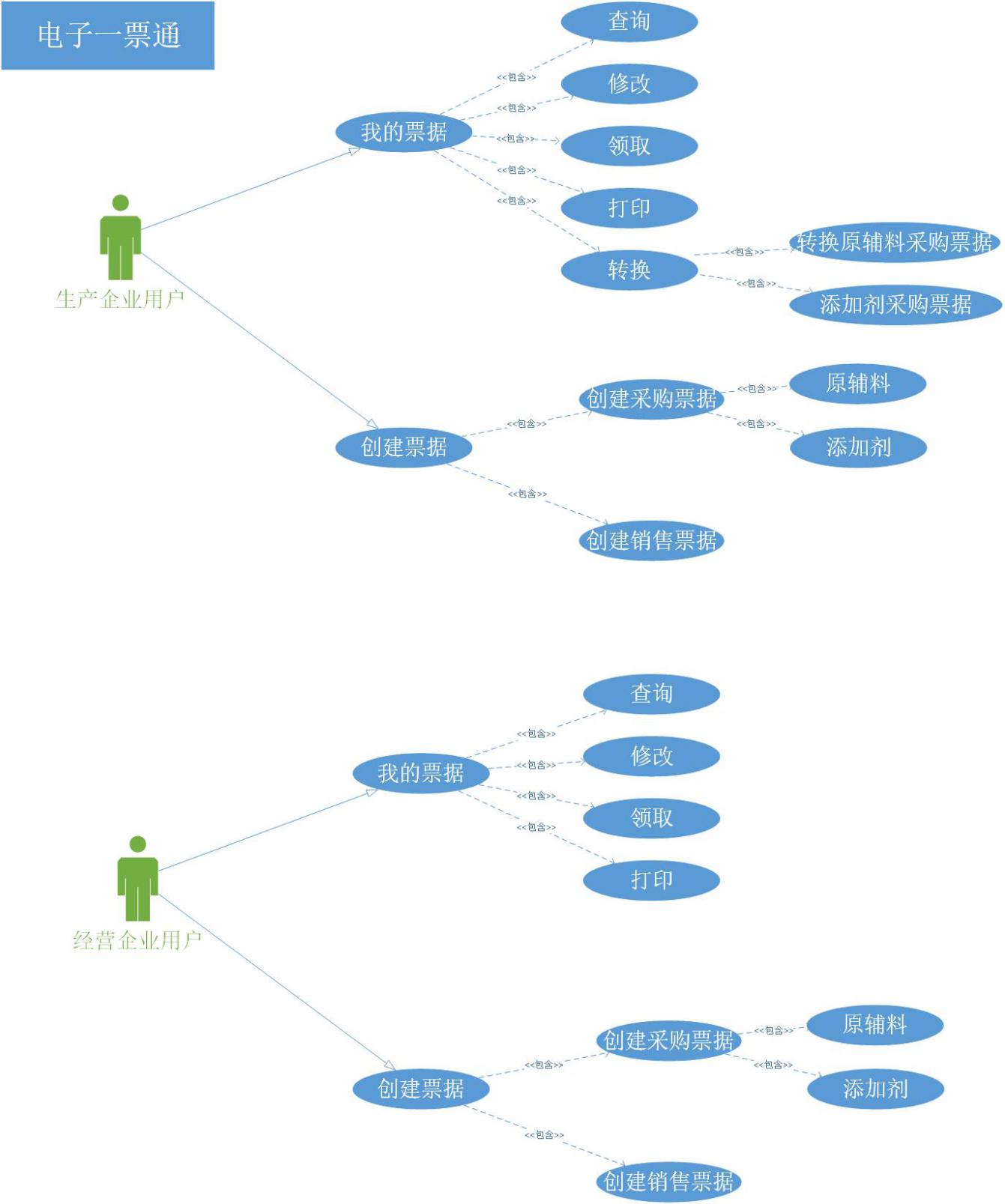


图 3‑4 电子一票通模块用例图

### 我的票据

我的票据主要是查询与本企业相关的所有“一票通”，并可按交易行为分列， 如原辅料采购、添加剂采购、食品采购和食品销售。票据分为全部票据、待领取票据、退回票据和拍照票据。票据信息包括票据号、票据类别、交易时间、交易单位、票据来源和状态等。

### 创建票据

票据分为采购票据和销售票据，生产企业根据交易行为和采购物质选择原辅料采购票据、添加剂采购票据、食品销售票据。经营企业根据交易行为选择食品采购票据和食品销售票据。票据的创建流程是：选择供货单位，从交易对象中选择企业，营业执照号、联系方式等企业信息系统自动填写；选择采购或销售日期，送货人和收货人为选填内容，企业根据需要进行输入；填写商品信息，商品信息包括商品类型、商品条码、名称、规格、保质期、生产日期、批号、数量等，填写预包装商品条形码后系统将从中国物品编码中心进行模糊查询，找出所有符合条码条件的商品供以选择，选择后系统将自动填写部分商品信息，可以进行清空和删除操作，可以填写多项商品。

## 报备管理模块

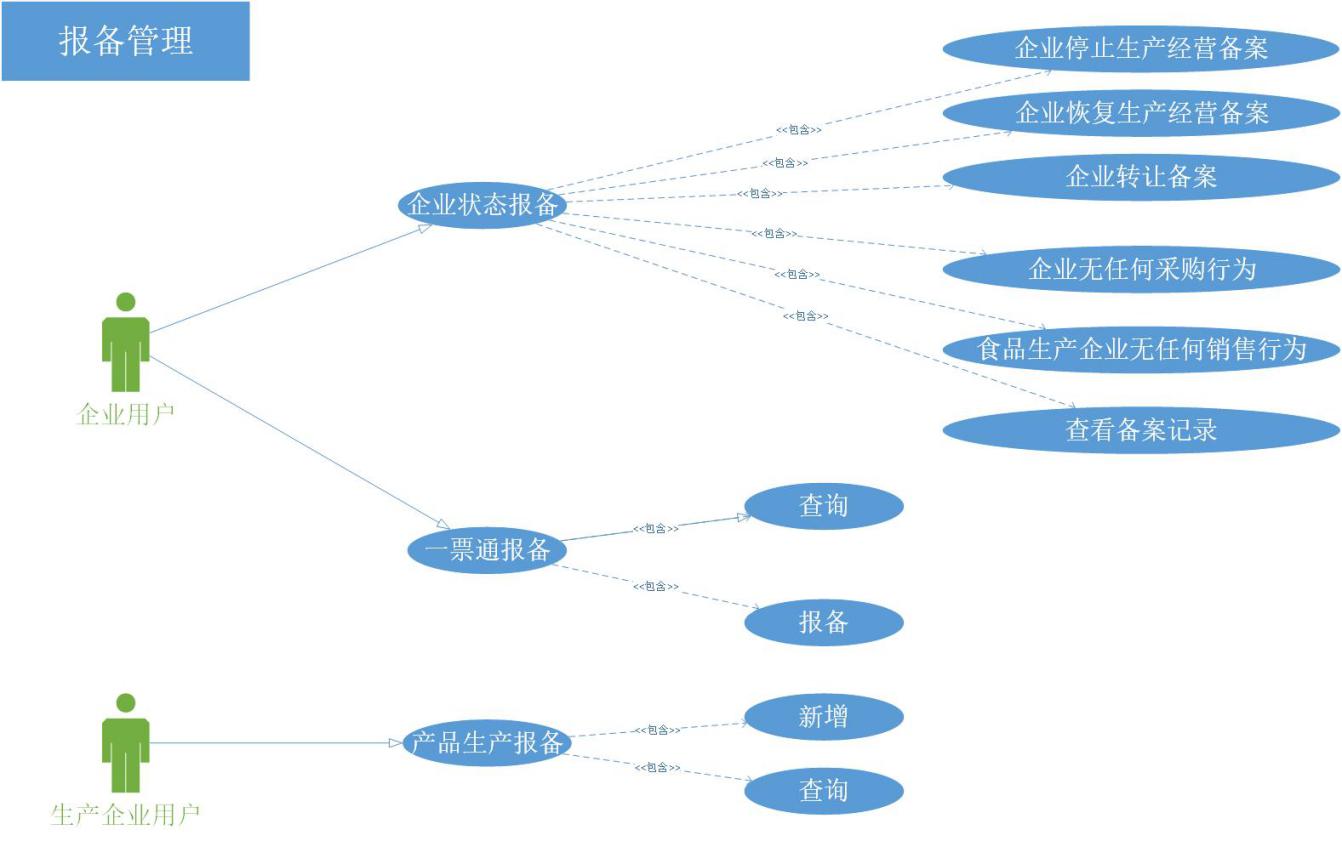


图 3‑5 报备管理模块用例图

### 企业状态报备

用户可以根据自己的实际情况，向监管部门进行备案报备，以便于监管部门了解企业最新的情况，一个月一个备案类型只能报备一次，可作废（必须要写明作废原因），只有作废后才能再次新增同一类型。企业状态报备分为企业停止生产经营备案、企业恢复生产经营备案、企业转让备案、企业无任何采购行为备案和食品生产企业无任何销售行为备案。企业转让备案需要填写转入转让日期和用户的姓名、身份证号、联系电话，其他备案需要填写日期。同时可以查询备案记录，可以对备案进行作废操作。

### 产品生产报备

生产企业需要进行生产报备，报备流程是：

1. 填写产品信息。需要填写产品名称、生产日期、加工工艺、生产批次、本批次产品数量、填报日期、出厂检验报告、标签图片、配料内容以及原始检测记录，其中产品名称将从自产品管理记录的产品中进行选择，生产批次是由生产日期生成的。
2. 填写原辅料使用列表。原辅料使用列表将从原辅料采购库中选择，需要填写原料使用量。
3. 填写添加剂使用列表。添加剂使用列表将从添加剂采购库中选择，需要填写添加剂使用量。
4. 填写关键控制点列表。需要填写控制点名称、关键控制点参数要求、加工起止时间、关键控制点参数实际情况、设备、操作人。第一次由企业自行填写，以后将作为模板不再需要填写。设备将从设备管理记录的设备中进行选择。

### 一票通报备

企业选择创建的票据进行报备。用户可以通过交易对象、交易日期和报备状态精确查询，对未报备的票据可以进行报备，对已报备的票据可以进行查看详细。

## 台账管理模块

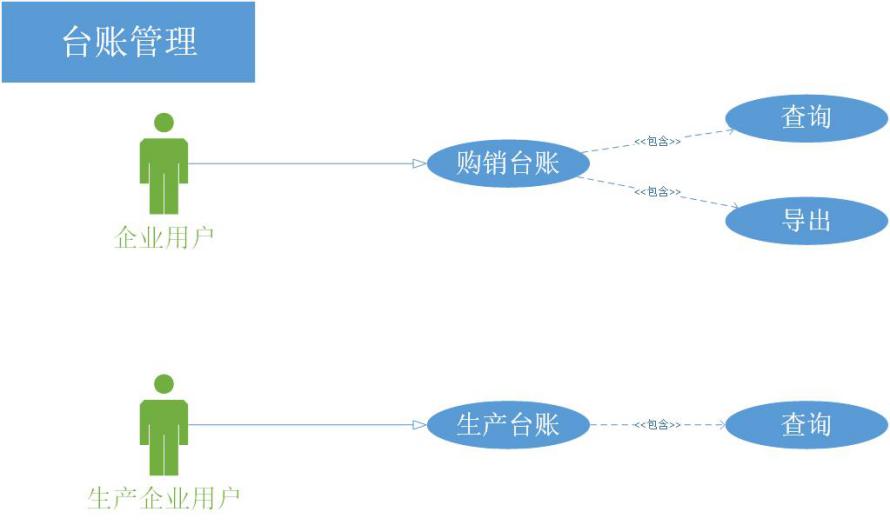


图 3‑6 台账管理模块用例图

### 销售台账

销售台账可以通过商品名称、交易单位、交易类型和交易时间进行精确查询并可以导出Excel表格，显示数据包括所属票据号、商品条码、商品名称、规格、数量、生产批次、保质期、交易类型、交易日期、交易单位名称、两证一报告、联系淡化和台账创建日期等。

### 生产台账

生产台账可以通过产品名称、报备时间和生产批次进行精确查询，显示数据包括产品名称、产品标准条形码、产品自编条形码、报备时间和生产批次等，并且可以查看产品详情。

## 费用中心模块

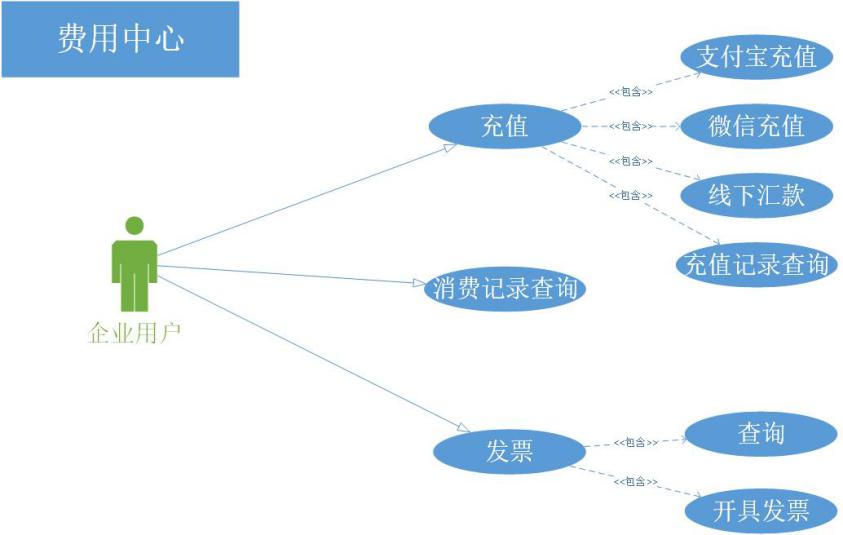


图 3‑7 费用中心模块用例图

### 账户总览

显示账户余额。用户充值后，金额显示在现金金额，再用此金额购买不同类型的软件服务。

### 充值

选择支付方式，如支付宝、微信或线下汇款，点击支付宝或微信充值，会跳出付款二维码，扫码即可。

### 消费记录

可以根据起止时间查询该段时间的消费记录。

### 发票

需要开具发票的企业，需提供发票抬头、税号、邮寄地址、联系方式、收件人和邮箱。可以根据起止时间查询该段时间的消费记录，对消费记录进行选择，开具发票。

## 帮助中心模块

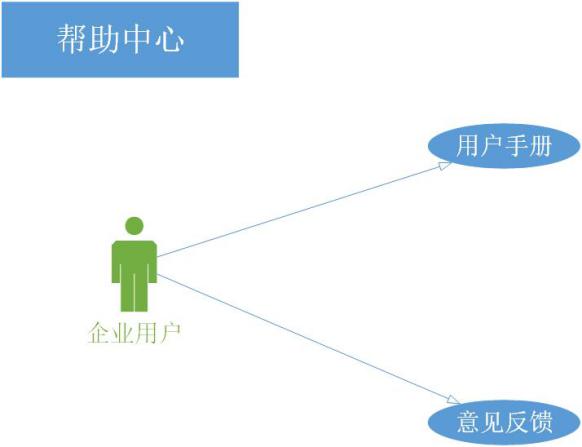


图 3‑8 帮助中心模块用例图

### 用户手册

用户手册包括pc端操作手册、票据录入指南和商户接口接入API。

### 意见反馈

意见包括对本公司意见和对食安云意见，填写意见内容和联系方式进行提交。

## 本章小结

本章简要地介绍了系统的需求分析，主要有七大功能模块：用户模块、信息管理模块、电子一票通模块、报备管理模块、台账管理模块、费用中心模块和帮助中心模块，然后通过用例图的方式介绍了各模块所拥有的子功能，并配以相关的文字对其进行描述。

# 系统设计

## 系统整体逻辑架构设计



图4-1 逻辑架构图

系统采用前后端分离的模式，整体逻辑架构图如图4-3所示，前端使用AngularJS框架来实现，访问基于Restful API的后端接口，并以JSON的数据格式进行交互。

整个后台被封装成多个微服务，在每个微服务中有多个接口，每个接口都是Resource，后台通过Mybatis框架对数据进行持久化操作。

值得注意的是，在访问API接口时，我们需要做以下判断：

1. 判断用户是否登录：通过Json WEB Token验证，用户在登录后，后台系统生成一个有时间限制的token，每次访问都需要携带token访问接口。
2. 判断资源是否存在：若资源不存在则返回404。
3. 判断请求参数是否合法：若请求参数不合法，则进行拦截。防止后台接口被破坏。

## 前端架构设计



图 4‑1前端架构设计图

前端的架构设计如图4-2所示，总体基于AngularJS框架，系统主要分为：View，Controller/Model和Service层，利用分层的思想将用户的显示页面，流程控制和模型层分离开来，在开发过程中更加高效、便捷，使系统具有良好的可扩展性和可维护性。

其中View层主要负责页面的展示功能以及交互功能，可以获取用户的输入，并将处理结果返回给用户。视图层主要由HTML、CSS、Bootstrap来实现，主页面index.html搭建完成后，其他各个页面作为模板插入到主页面中。

Controller/Model层主要负责将获取到的View层的数据进行一定处理后，将数据传给Service层进行与服务端的数据交互，然后将从Service获取到的数据进行处理后，通过给上下文域中的Model赋值来改变View层的内容展示。

为了避免Controller层的代码过于臃肿，不利于项目的可扩展性，分离出了Service层，主要用于通过HTTP协议与服务端的接口进行通讯。一个Controller可以注入多个Service服务。

## RESTful API规范

本项目访问的后端接口应符合RESTful API规范。

### 基本规范

1. 用户与API通信协议，总是基于HTTP/1.1协议。
2. 应将版本号放入URL中去，例如<http://ip:port/v1>
3. url都用小写，单词与单词连接都用“-”。
4. 每个URL都代表唯一的网络资源，因此URL中不应该存在动词，只能有名词，一般操作的资源与数据库的表明一一对应，例如<http://ip:port/v1/account>
5. 资源的具体操作类型，由HTTP动词GET，POST，PUT，DELETE，分别对应资源的查，增，改，删。
6. 过滤信信息，利用p和n来处理分页，p表示页数，n表示每页数量；利用orderby来处理排序。例如：<http://ip:port/v1/account?p=1&n=10&orderBy=-announce_time>
7. 默认格式：JSON；
8. JSON响应结构：{“meta”:{“code”:xxx,”message”:””},”data”:{}}，主要分为两段，meta段封装操作成功或失败信息，用code和message指明，data段封装具体的业务逻辑顺序。

### 基本状态码

使用的状态码主要如下：

1. 200 —OK：服务器请求成功，成功返回用户请求数据。
2. 201 —Created：用户新建或修改数据成功。
3. 204 —Accepted：用户删除数据成功。
4. 400 —Invalid Request：用户发出的请求有错误，服务器没有进行新建或修改数据的操作。
5. 401 —Unauthorized：即用户的JSON Web Token失效，或JSON Web Token不合法。
6. 403 -Forbidden：即用户的相应请求被阻止。
7. 404 -NOT FOUND：即用户请求的资源不存在。
8. 406 -Not Acceptable：用户请求的格式有误，标准格式为JSON格式。
9. 413 -Payload Too Large：请求消息实体过大
10. 500 -INTERNAL SERVER ERROR ：服务器发生内部错误。

## 本章小结

本章介绍了系统的逻辑架构设计，采用前后端分离的模式，随后介绍了系统的前端架构设计，包括View，Controller/Model和Service层，利用了分层的思想，使系统具有良好的可扩展性和可维护性。

# 系统实现

## 框架搭建及环境配置的实现

### AngularJS框架搭建

1. 在页面引入AngularJS库：

<script src="../../assets/global/plugins/angularjs/angular.min.js" type="text/javascript"></script>

1. 创建WebApp

AngularJS的每个页面都是一个独立的WebApp，属于单页面应用。AngularJS使用ng-app指令来定义AngularJS应用的作用域，通常ng-app在<html>标签中加载。

<!doctype html>

<html lang="en" ng-app="MetronicApp">

</html>

在main.js中创建MetronicApp模块对象，并注入依赖：

var MetronicApp = angular.module("MetronicApp", [

"ui.router",

"ui.bootstrap",

"ngMessages",

"oc.lazyLoad",

"ngSanitize",

"restangular",

'ui.calendar',

'ngFileUpload',

'ngCookies',

'ngAnimate',

'ui.bootstrap.datetimepicker',

'monospaced.qrcode'

]);

1. 在模板中添加控制器

接下来，需要为希望让AngularJS模块控制的HTML元素添加控制器，同时需要在模块代码中定义控制器。下面以loginController为例进行介绍。

首先在login.html中将loginController赋给<div>元素：

<div class="login" ng-controller="loginController">

然后在JavaScript中将loginController添加到MetronicApp模块中：

MetronicApp.controller('loginController', function($rootScope, $scope, $http, $state,$timeout, $filter, $cookieStore, customerService, $interval) {});

1. 路由设置

使用AngualrJS路由我们可以通过不同的URL访问不同的页面。

MetronicApp.config(['$stateProvider','$urlRouterProvider',function($stateProvider,$urlRouterProvider) {

// Redirect any unmatched url

$urlRouterProvider.otherwise("/dashboard");

$stateProvider

.state('app', {

url: '/app',

templateUrl: 'app.html',

abstract: true

})

.state('app/login',{

url: "/login",

data: { pageTitle: '登录注册' },

templateUrl: "views/login.html",

})

}]);

### Gulp自动化工具的配置

食安云企业端使用gulp作为前端的自动化构建工具。将所有的控制器代码合并为一个文件angular-controller.js，所有的service代码合并为一个文件angular-service.js，合并的好处是在页面访问时，只需要发起一个http请求，能减少加载时间。同时能对文件进行监听，在修改代码后能自动刷新浏览器，提高开发效率。使用gulp还能在本地开启一个轻量级的web服务，下面主要介绍gulp的配置方法。

1. 首先安装nodejs

gulp是基于nodejs的，所以需要先安装nodejs。

1. 新建package.json文件

package.json是用来管理本地安装的npm包的一个重要配置文件，放在项目的根目录中。它描述了项目的基本信息，以及所用到的依赖包和版本，在命令行中输入npm init，可以生成一个默认的package.json文件。

{

"name": "shianyun-web-en",

"version": "1.0.0",

"description": "",

"main": "gulpfile.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC",

"devDependencies": {

"gulp": "^3.9.1",

"gulp-concat": "^2.6.1",

"gulp-connect": "^4.2.0",

"gulp-ignore": "^2.0.1",

"gulp-minify-css": "^1.2.4",

"gulp-prettify": "^0.5.0",

"gulp-rename": "^1.2.2",

"gulp-rtlcss": "^0.1.4",

"gulp-sass": "^3.1.0",

"gulp-sourcemaps": "^1.5.2",

"gulp-uglify": "^1.5.4",

"node-sass": "^4.5.3"

}

}

1. 全局和本地安装gulp

全局安装gulp： npm install gulp –g

本地安装gulp插件（以gulp-less为例）：npm install gulp-less --save-dev

1. 新建gulpfile.js文件

gulpfile.js是gulp的配置文件，放于项目的根目录中。下面列出gulpfile.js的部分内容：

var gulp = require('gulp');

var sass = require('gulp-sass');

var prettify = require('gulp-prettify');

var minifyCss = require("gulp-minify-css");

var rename = require("gulp-rename");

var uglify = require("gulp-uglify");

var rtlcss = require("gulp-rtlcss");

var connect = require('gulp-connect');

var concat = require('gulp-concat');

//\*\*\* Localhost server tast

gulp.task('localhost', function() {

connect.server();

});

gulp.task('localhost-live', function() {

connect.server({

livereload: true,

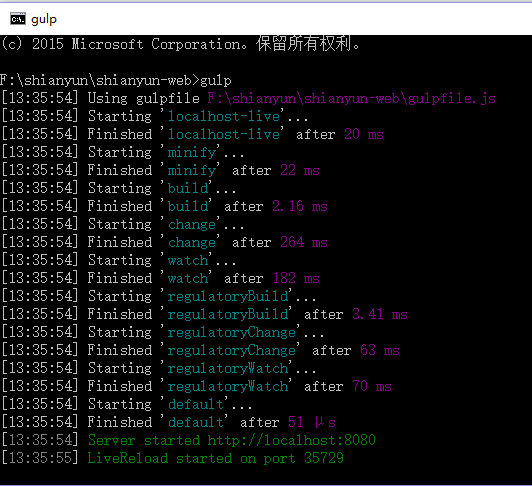
});

});

// 默认任务

gulp.task('default',['localhost-live','minify','build','change','watch','regulatoryBuild','regulatoryChange','regulatoryWatch']);

1. 通过命令行运行gulp，开启本地web服务,如图5-1



5‑1 gulp运行效果图

### Restangular简化API请求

在本项目中采用Restangular来进行前后台交互，Restangular 是AngularJS 服务，简化常规的 GET, POST, DELETE 和 UPDATE 请求，使用更简短的代码[Restangular 适用于所有WebApp，从RESTful API中消耗数据。

1. 利用setBaseUrl方法设置API请求路径

Restangular.setBaseUrl(API\_URL);

利用setResponseInterceptor方法添加拦截回复请求：

Restangular.setResponseInterceptor(function (data, operation, what, url, response, deferred) {

if (data.status == -1) {

deferred.reject(data);

swal("登录已过期，请重新登录！");

window.location.href="app.html#/login";

}else if (data.status == 200) {

if(data.paramcentcount){

data.param.paramcentCount = data.paramcentcount;

}

return data.param;

}else if (data.status == 204){

return data.param;

}else {

deferred.reject(data);

hideWait();

swal(data.info);

}

});

利用setErrorInterceptor方法拦截错误请求：

Restangular.setErrorInterceptor(function(response, deferred, responseHandler) {

console.log("http status="+response.status);

if(response.status === 500) {

swal("http 500状态！");

hideWait();

return false; // error handled

}else if(response.status === -1){

swal("请求超时");

hideWait();

return false;// error handled

}

else{

swal("其他http错误");

return false;// error handled

}

});

## 用户模块功能实现

用户模块包括注册、登录以及控制台功能，使用ng-pattern实现正则表达式，对注册登录的手机号以及密码输入格式进行判断是否正确；使用ng-message实现表单验证，在用户输入有误情况下提供友好的提示信息；使用$interval定时器实现发送验证码倒计时，限制用户一分钟内操作一次；使用modal模态框实现控制台的切换企业、消息查看详情功能；使用ui-sref路由绑定实现控制台的快捷入口页面的跳转。



图5-2注册登录页面



图5-3完善信息页面

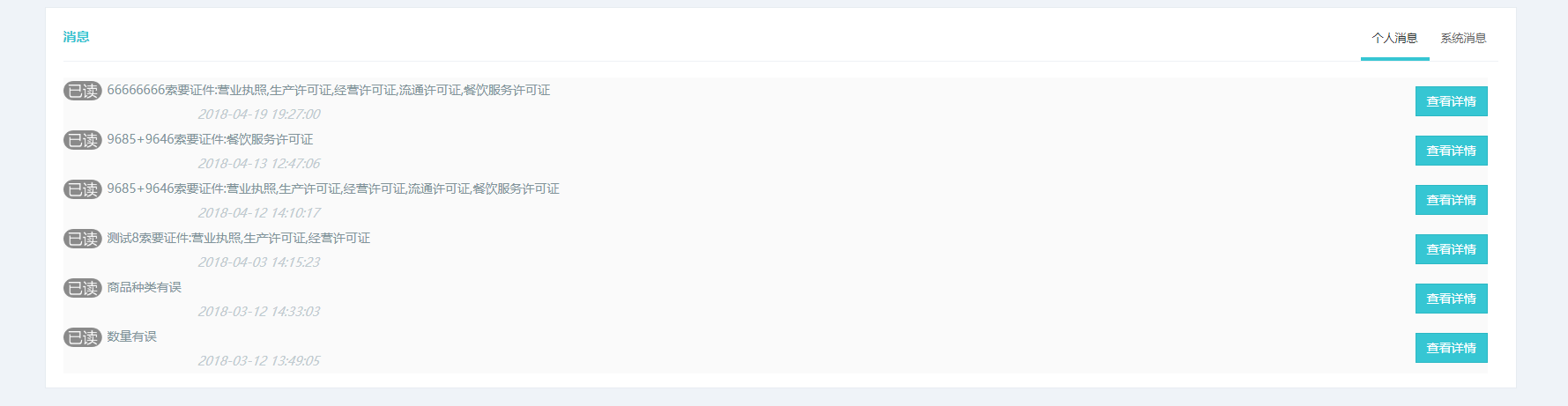


图5-4控制台页面



图5-5切换企业页面

## 信息管理模块功能实现

信息管理模块是对企业信息、产品信息、交易对象、从业人员、设备等进行管理，为企业生产报备和一票通报备做充分的准备。使用bootstrap-pagination控件实现分页查询与显示；使用bootstrap-datepicker控件实现日期选择；使用ng-file-upload和调用又拍云存储接口实现图片上传；使用jQuery的Zoom插件实现图片放大功能；使用jstree实现选择许可范围、产品分类的树状图显示；使用modal模态框实现子账号设置与查看等功能。

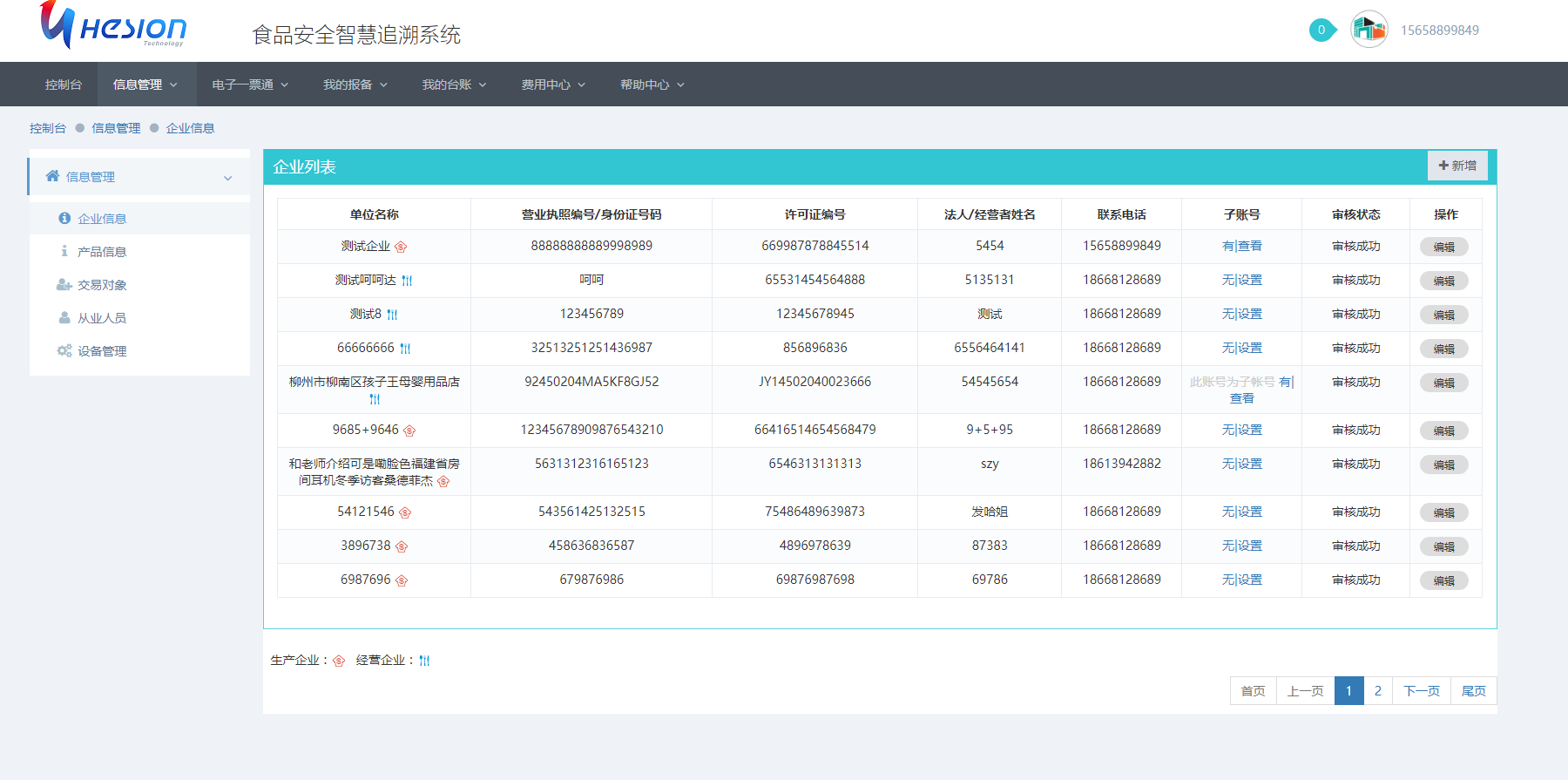


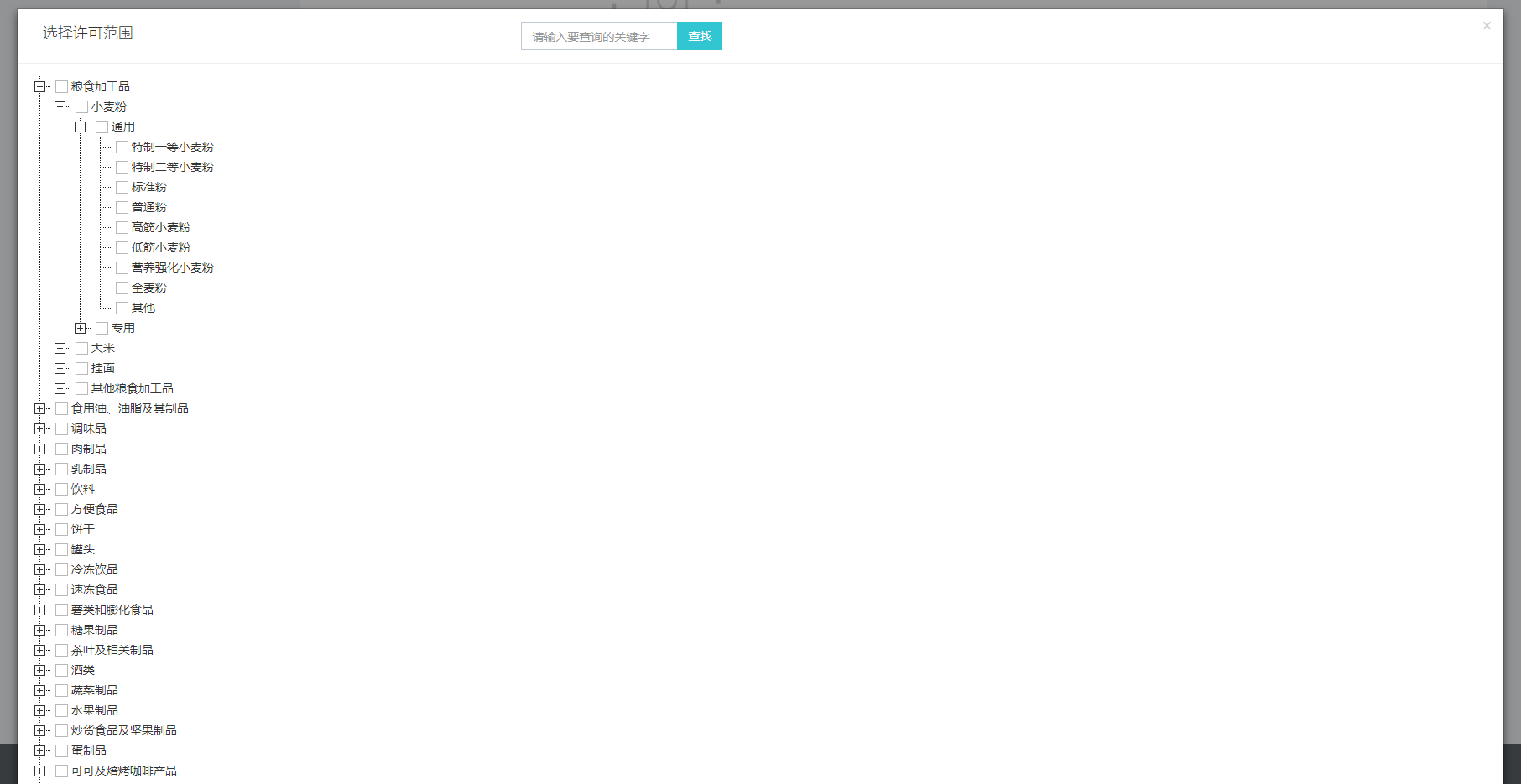
图5-6企业信息页面



图5-7新增企业页面



图5-8编辑企业页面



5-9许可范围页面



图5-10产品信息页面



图5-11交易对象页面

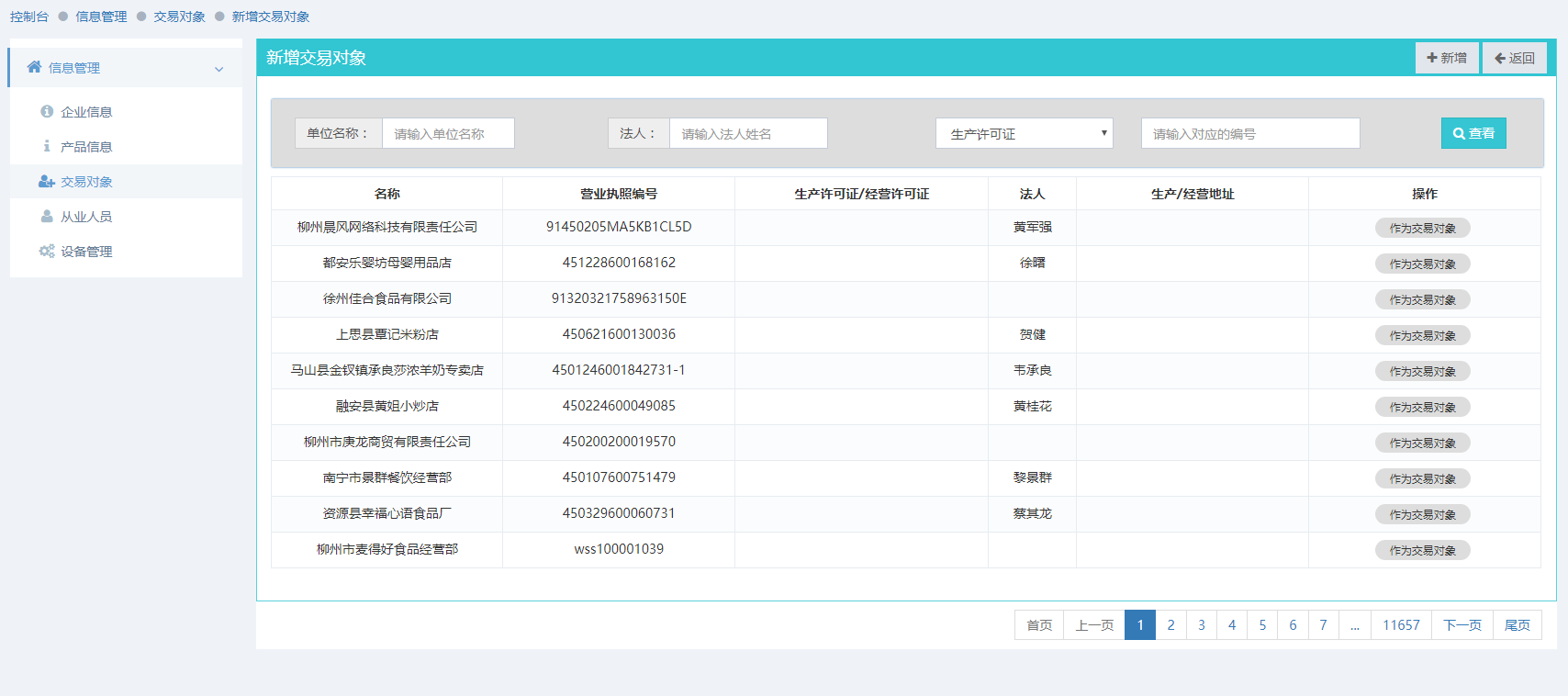


图5-12新增交易对象页面



图5-13从业人员页面

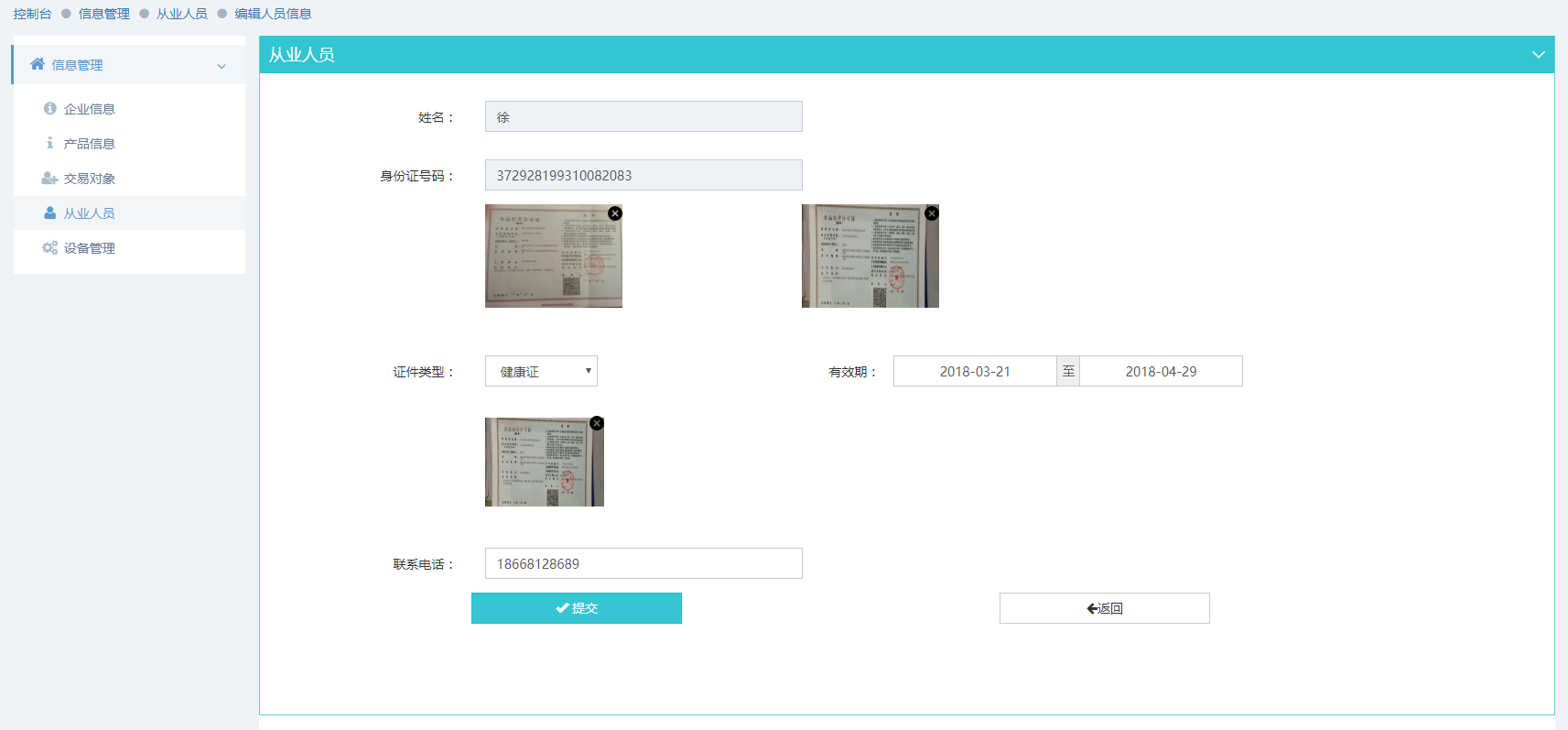


图5-14编辑从业人员页面



图5-15设备管理页面

## 电子一票通模块功能实现

电子一票通模块包括我的票据和创建票据两部分，生产企业可以创建原辅料采购票据、添加剂采购票据、食品销售票据，经营企业可以创建食品采购票据和食品销售票据，根据不同类型的企业限制其创建票据权限；创建票据需要生成票据号，其算法是“cg”/“xs”(采购票据/销售票据)+当前时间的时间戳+0到1之间的31进制随机数的第3位到第7位的字符；创建票据是从交易对象中选择供货单位，将其营业执照号、商户地址、联系方式、证照图片等信息绑定到显示框中，实现自动填写；创建采购预包装票据填写商品条码，从中国物品编码中心进行查找，并将其名称、规格、保质期等信息绑定到显示框中；创建采购散装票据，输入商品名称，根据供货单位的产品信息管理进行查找，如果没有查找到的话可以进行创建该商品，其产品条码为企业营业执照号+时间戳；创建销售票据，产品名称是从产品信息管理中选择产品以及其产品批次，并对该产品的商品类型、商品号码、规格、批号等信息进行绑定；使用window.print()方法可以实现对票据的打印。

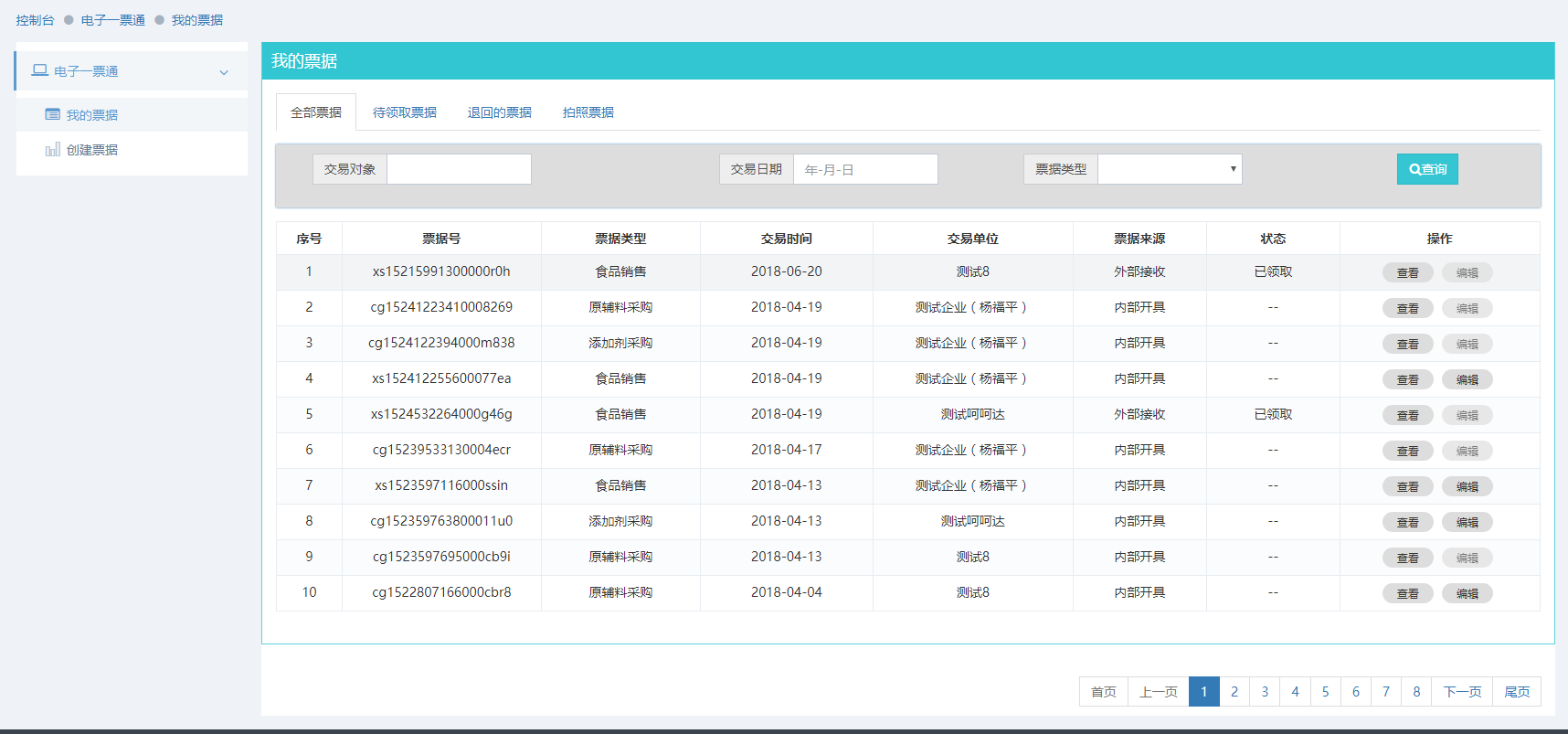


图5-16我的票据页面

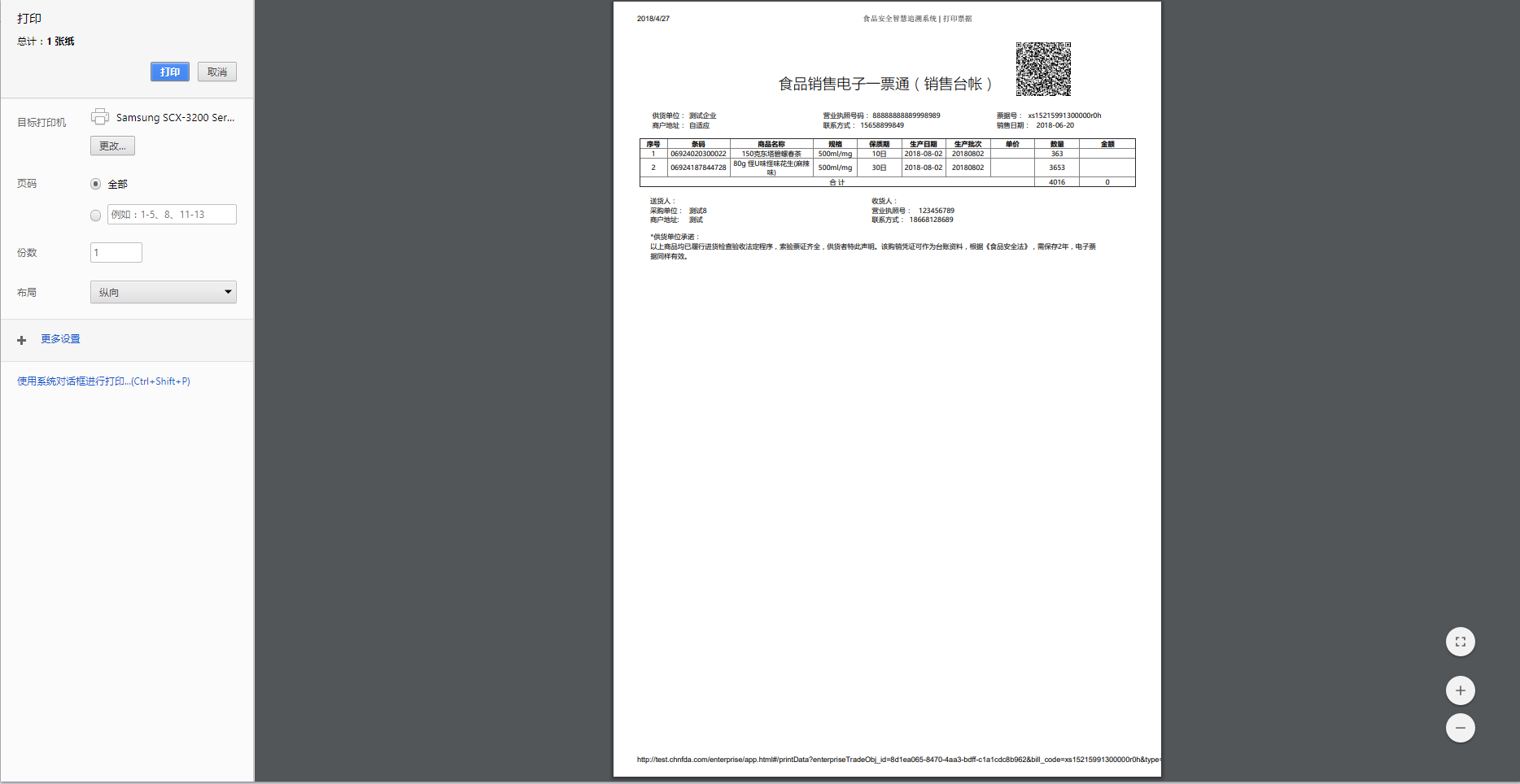
图5-17票据详细页面

图5-18票据打印页面

图5-19创建票据页面

图5-20原辅料采购电子一票通页面

图5-21食品销售电子一票通页面

## 报备管理模块功能实现

报备管理包括企业状态报备、产品生产报备和一票通报备，产品生产报备的产品从产品信息管理中进行选择，将其标签图片、配料内容和关键控制点列表进行绑定，填写生产日期，根据生产日期生成生产批次，原辅料实用列表和添加剂使用列表分别是从其对应的电子一票通报备中的商品进行选择；一票通报备是对我的票据进行报备和查看，传递票据号、交易对象id和报备状态id这些参数跳转到票据详细页面，使用$stateParams获得传值，根据传值进行显示。



图5-22企业状态报备页面

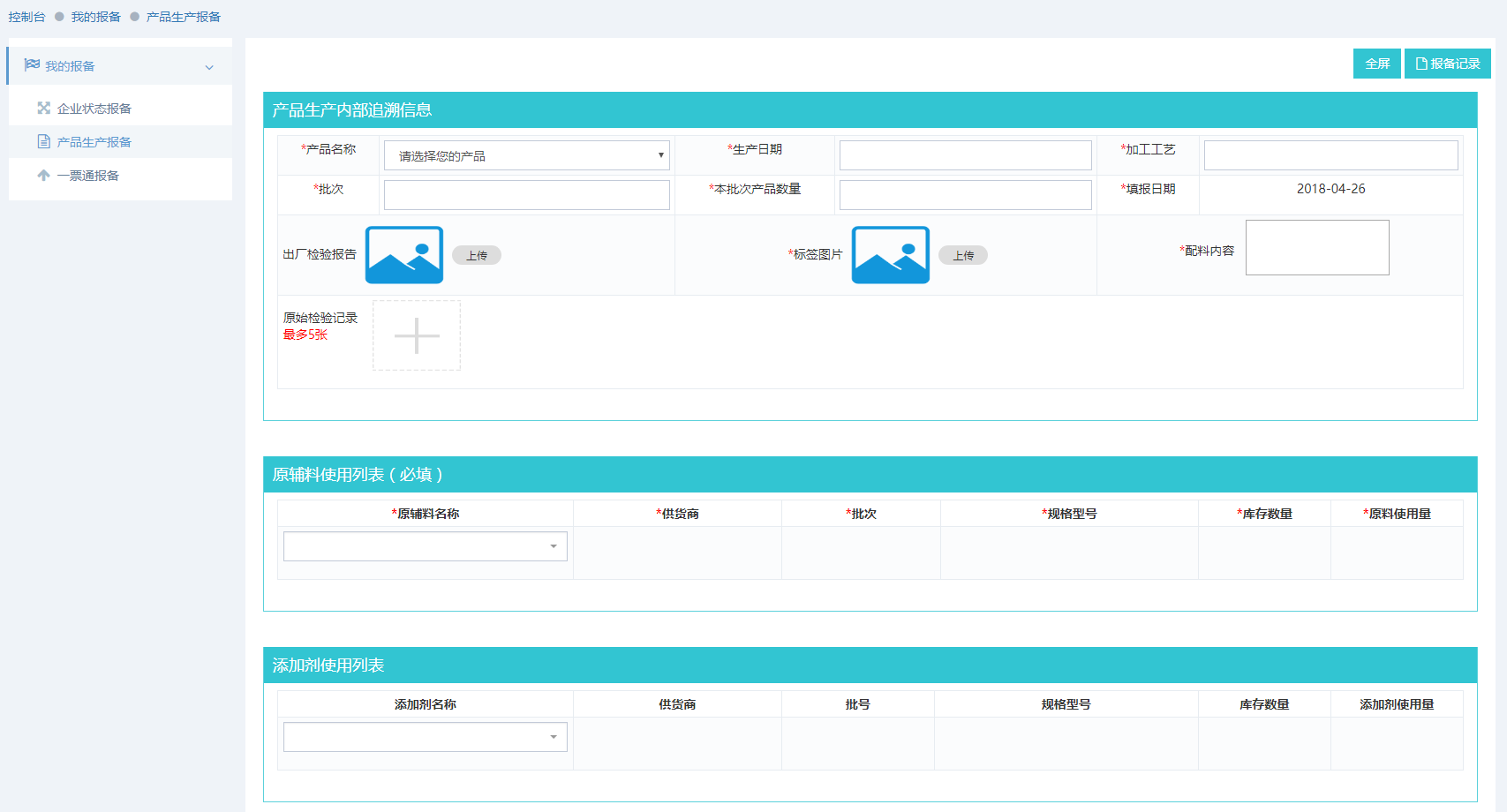




图5-23产品生产报备页面

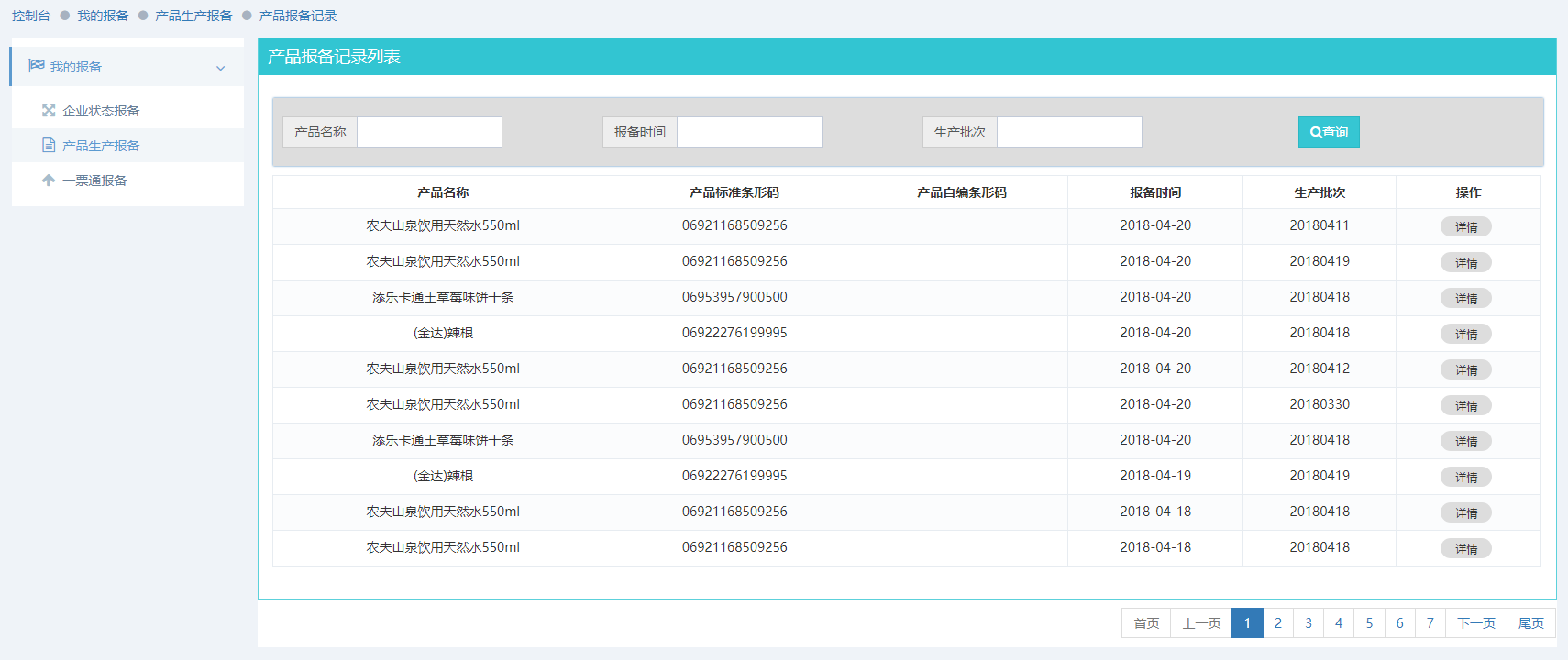


图5-24产品报备记录页面

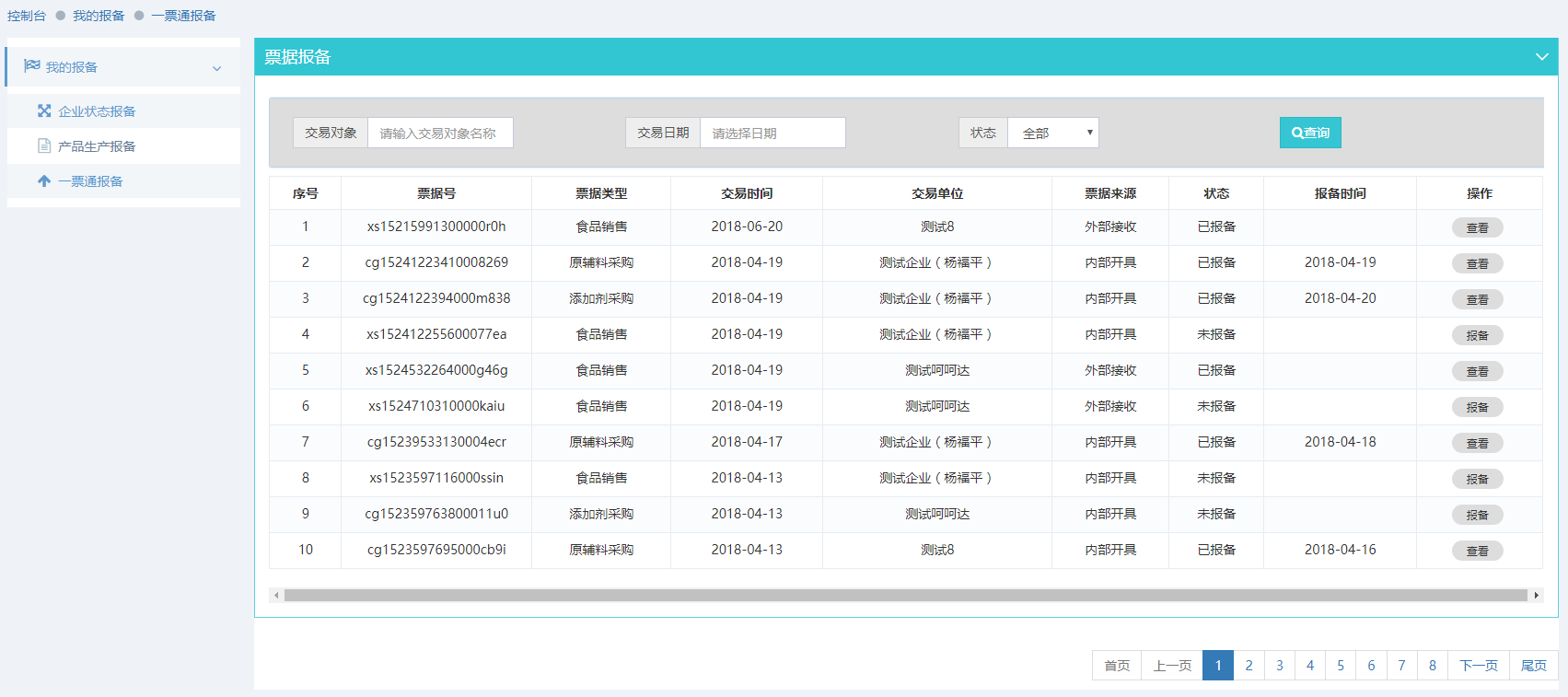


图5-25一票通报备页面



图5-26一票通报备详细页面

## 台账管理模块功能实现

台账管理模块包括生产台账和销售台账，使用Blob类文件对象实现数据excel表格导出。

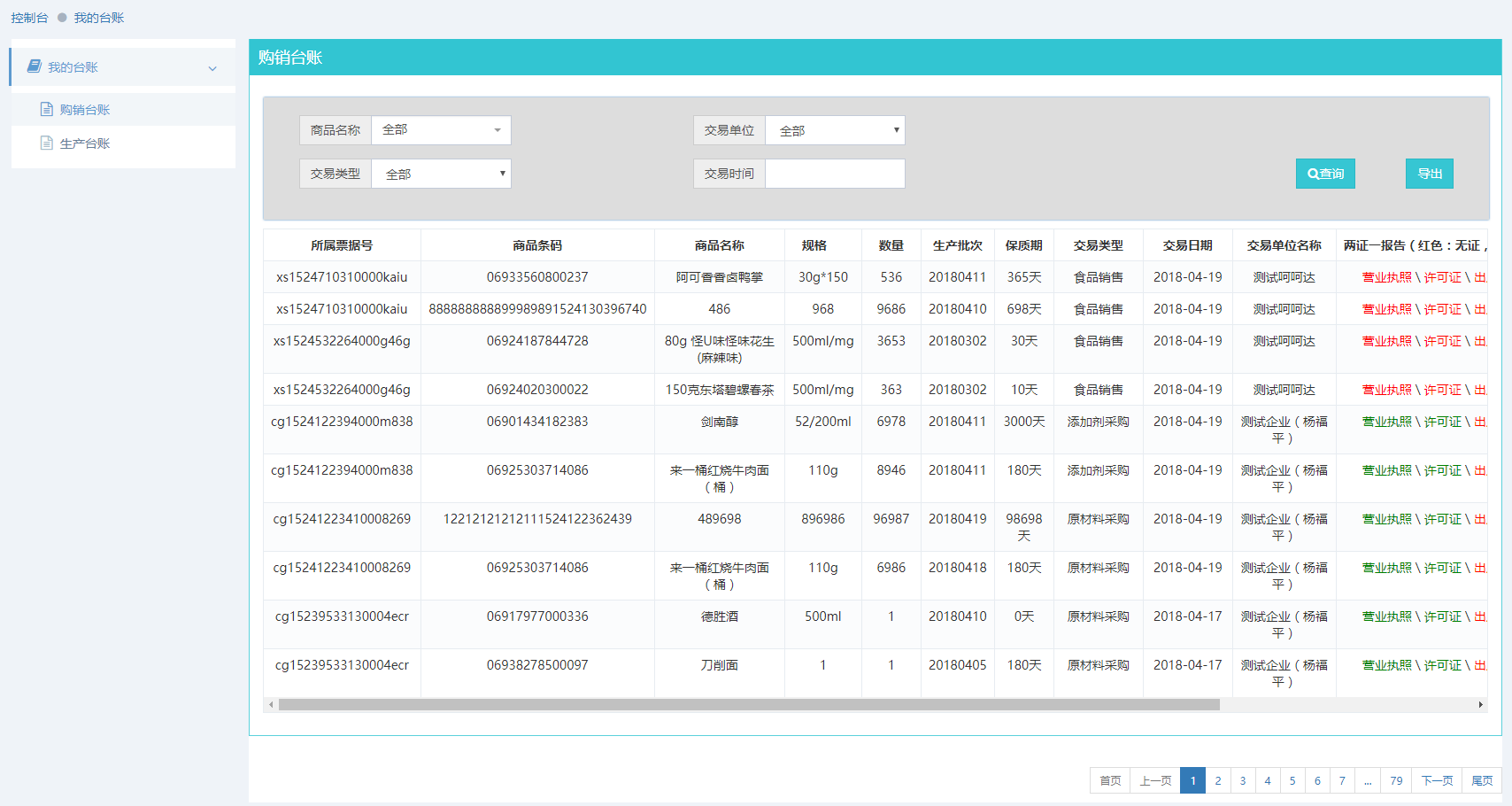


图5-27购销台账页面

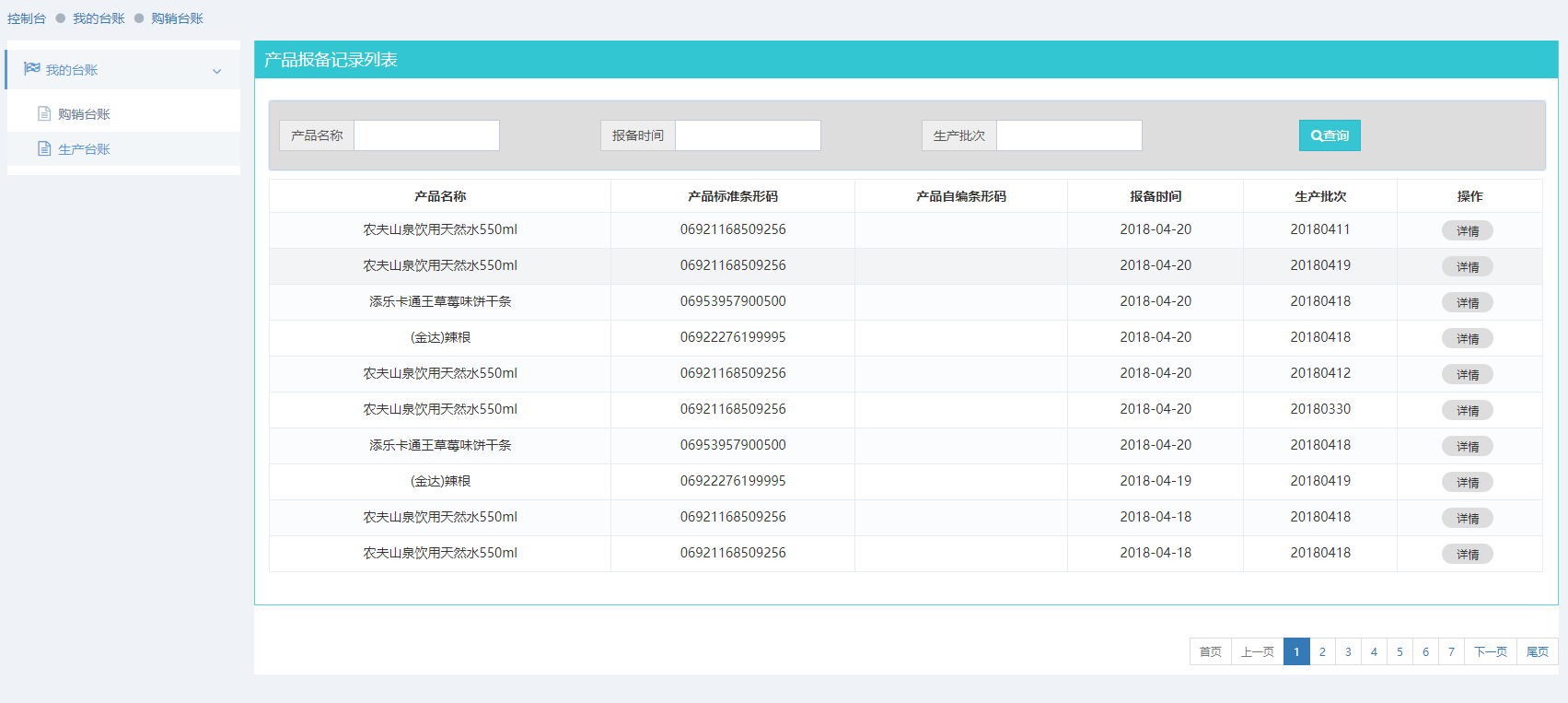


图5-28生产台账页面

## 费用中心模块功能实现

费用中心模块包括账户总览、充值、消费记录和发票，充值调用支付宝和微信的支付接口，获得支付URL和订单号，通过qrcodeJS库实现二维码显示，并使用$interval定时器每10秒循环调用一次查询接口监听该订单是否支付，直到支付成功或失败为止。

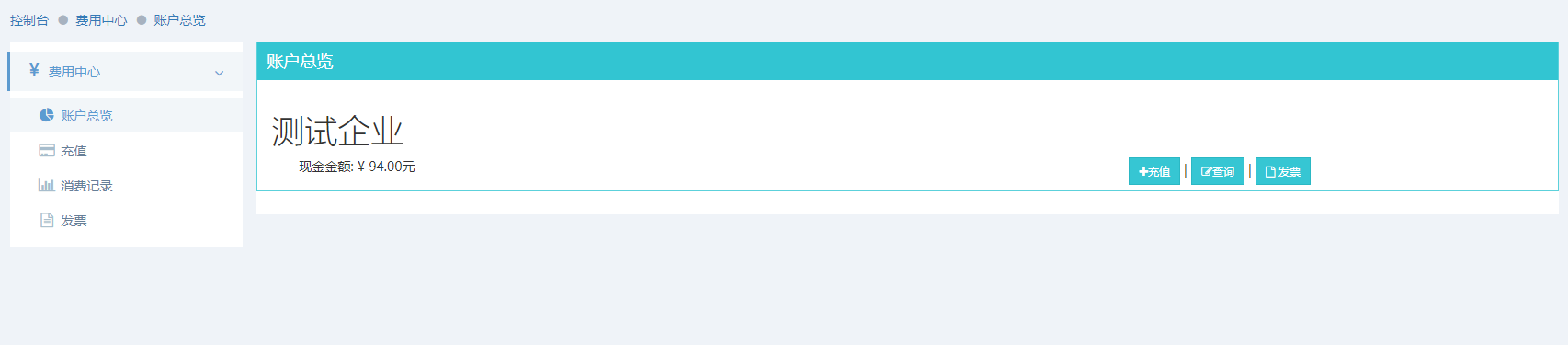


图5-29账户总览页面

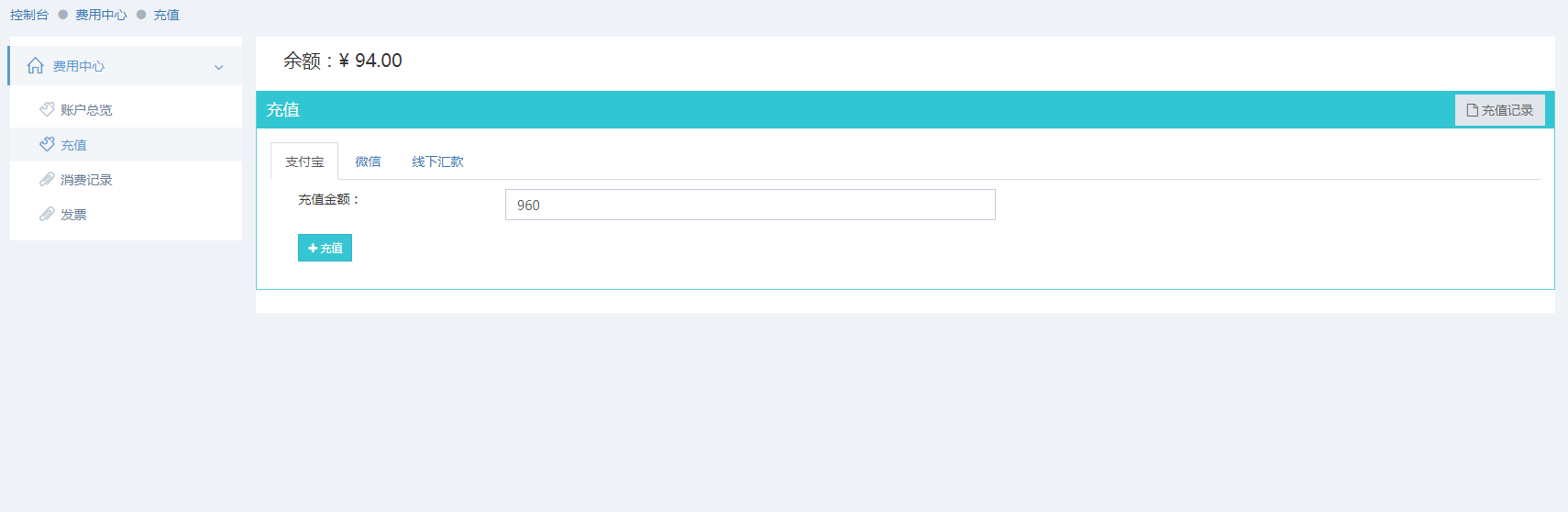


图5-30充值页面



图5-31充值二维码页面

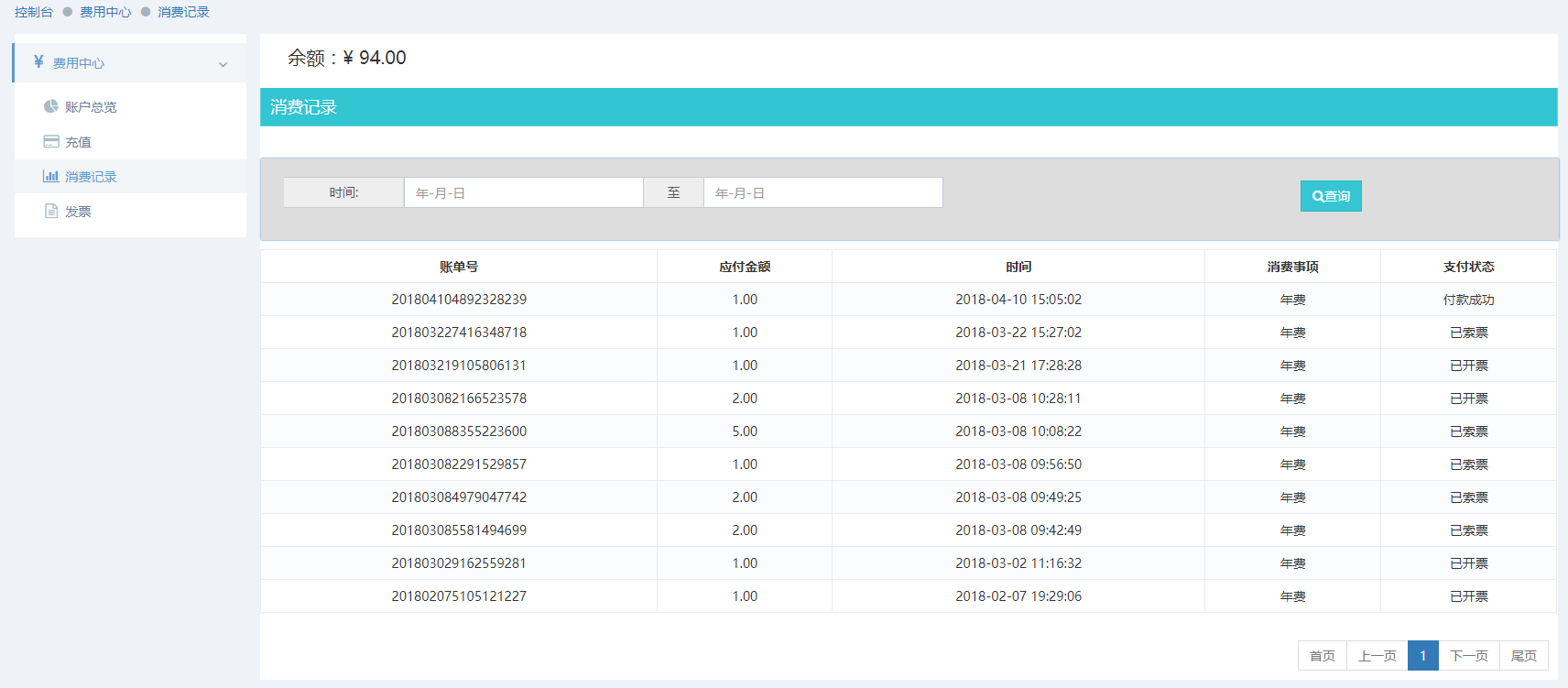


图5-32消费记录页面

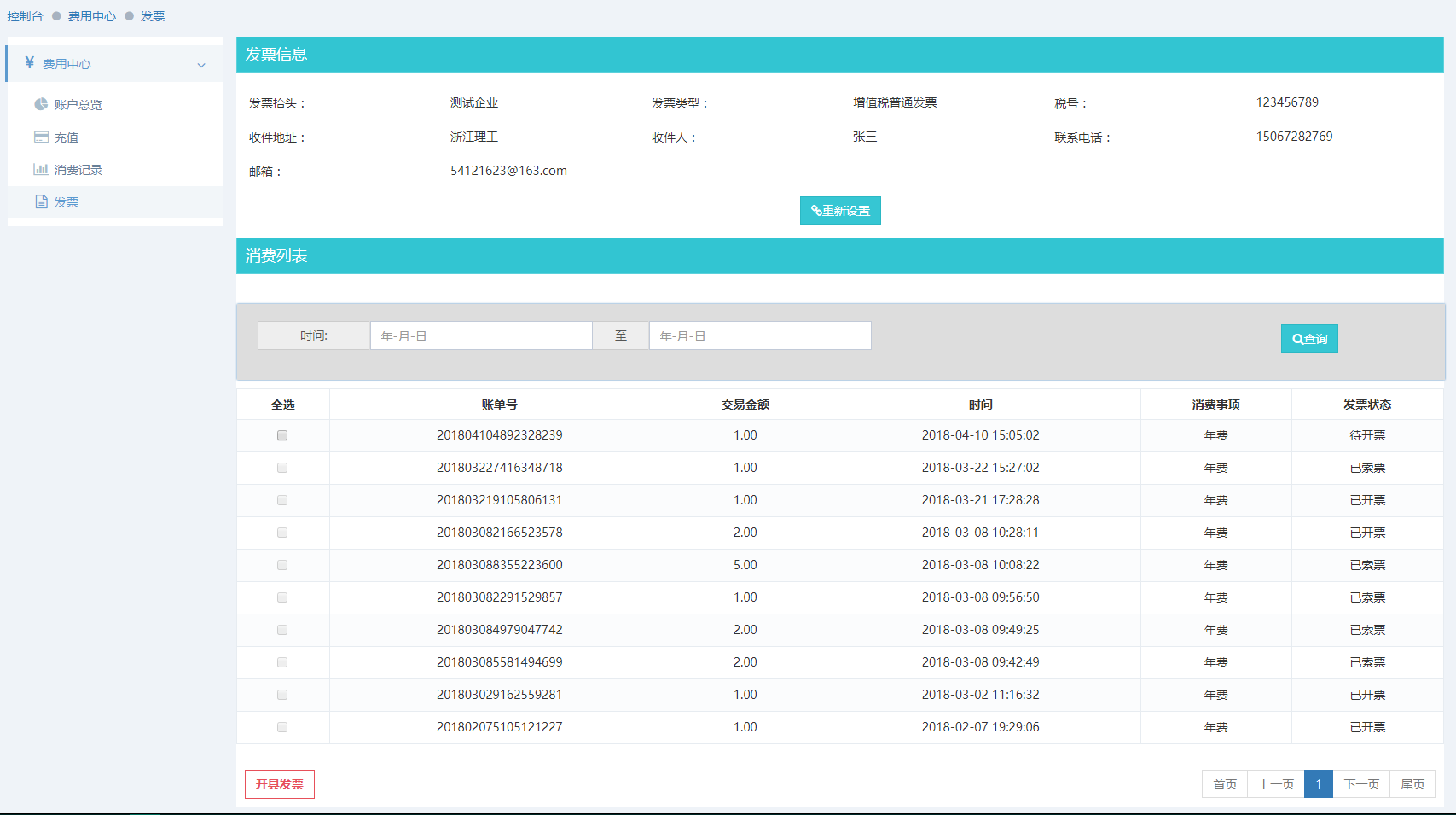


图5-33发票页面

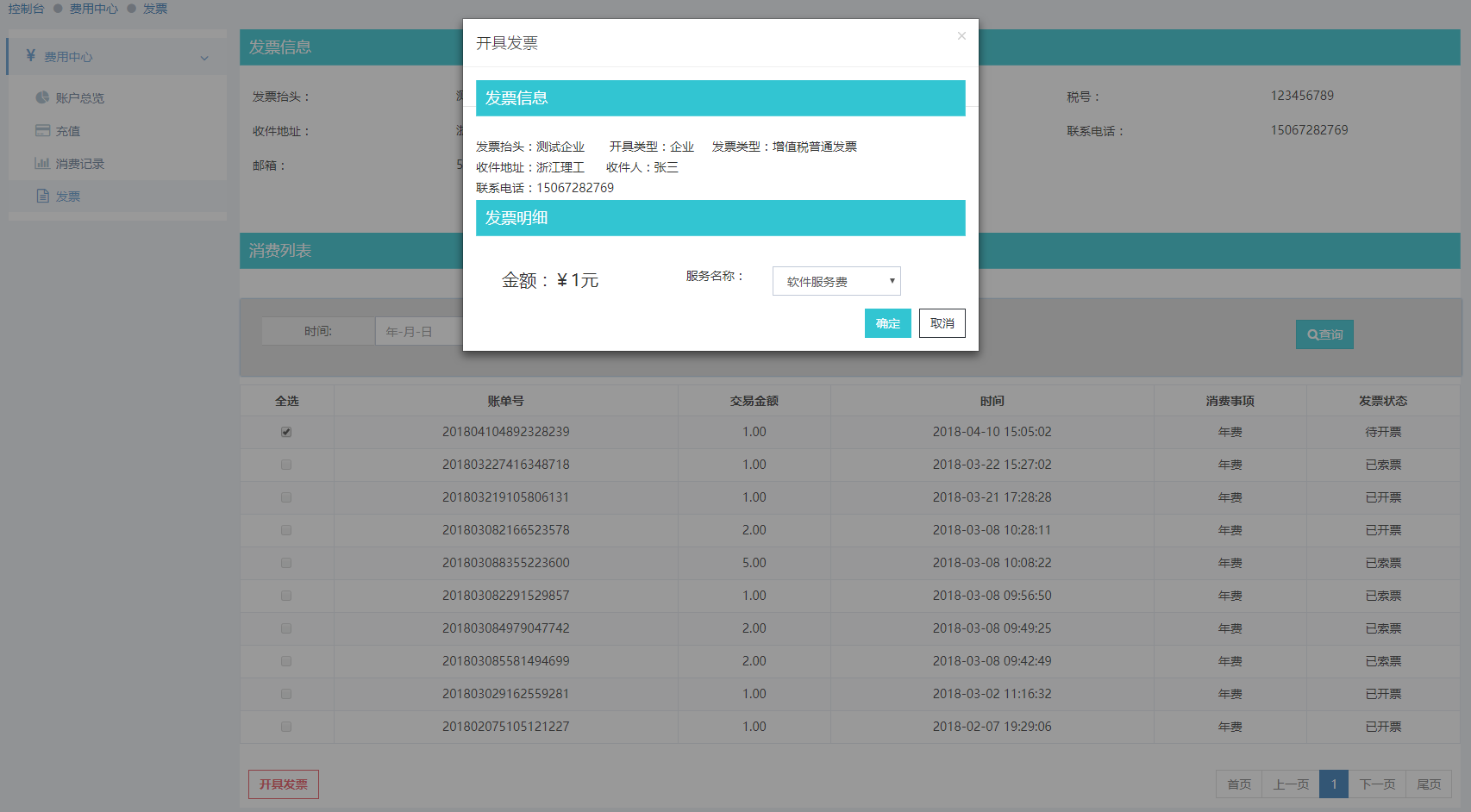


图5-34开具发票页面

## 本章小结

本章简要介绍了系统的框架搭建以及环境配置的实现过程，介绍了AngularJS的框架搭建，gulp自动化工具的配置，以及Restangular在简化请求API中起到的作用。最后介绍了系统六个主要功能模块的具体实现，包括用户模块、信息管理模块、电子一票通模块、报备管理模块、台账管理模块、费用中心模块。

# 关键技术及具体实现

## 又拍云上传图片

食安云企业端在完善企业信息、产品管理、生产报备、一票通报备等多个功能模块中都需要图片上传，使用又拍云上传图片，可以做到灵活上传，提高工作效率，降低开发成本，实现效果如下图所示。



图6-1上传图片

具体实现核心代码如下：

<div class="form-group">

<label class="col-md-3 control-label" for="title">主体图片</label>

<label class="col-md-6 control-label" for="title" style="text-align:left;"><span style="color:red;">最多5张，请上传企业门面照片、招牌照片、证件悬挂位置照片</span></label>

<div class="col-md-9 col-md-offset-3">

<div class="col-md-2 text-center" ng-repeat="e in imgObjs" style="width:165px;height:135px;margin-right: 10px;margin-top: 20px;float: left;" >

<div style="width: 135px;height:135px;">

<img id="cpImgId{{$index}}" class="jqzoom\_main" ng-src="{{e.url | uploadFile}}" style="width:{{e.Width}}px;height:{{e.Height}}px;margin-right:10px;margin-top: 10px;" rel="{{e.url | uploadFile}}" >

</div>

<div style="position: relative; float: right;margin-top: -136px; z-index:1000;" >

<img ng-click="remove($index)" src="../../assets/pages/img/remove .png">

</div>

</div>

<div style="margin-top:20px;display:inline-block" ng-if="enclosure.length < 5" id="fileUpdate5">

<img src="../assets/pages/img/fujian.png" ngf-select="uploadFiles($files, $invalidFiles, add,'fileUpdate5')" ngf-pattern="'image/\*'" accept="image/jpg,image/jpeg,image/png" ngf-max-size="10MB" style="width:135px;height:135px;margin-right: 10px;margin-top: 10px;">

</div>

</div>

</div>

//上传图片

$scope.uploadFiles = function(files, errFiles, successFunc, fileid) {

$scope.file.fileid = fileid;

App.blockUI({ //加载等待提示图片

target: '#' + fileid,

iconOnly: true

});

//检查不符合要求的文件

if (errFiles.length > 0) {

swal('有文件不符合要求，无法上传！');

App.unblockUI('#' + $scope.file.fileid); //加载等待提示图片

return;

}

//没有选择文件取消等待操作

if (!files) {

App.unblockUI('#' + $scope.file.fileid); //加载等待提示图片

}

//选择文件后继续多个上传操作

angular.forEach(files, function (file) {

$scope.submit(file, successFunc);

})

};

$scope.filter = {

files: []

};

//提交

$scope.submit = function (file, successFunc) {

//操作员定义

var operator = 'hesion';

var pwd = hex\_md5('\*\*\*\*\*\*\*\*\*\*');//md5加密

//从html获取file

$scope.filter.files.push(file.name);

//计算policy

var date = new Date();

var content\_md5 = hex\_md5(file);//md5加密

var policy\_before\_md5 = {

"bucket": 'hesion',

"save-key": '/' + date.getFullYear() + '/' + (date.getMonth() + 1) + '/' + date.getDate() + '/' + content\_md5 + file.name,

"expiration": Date.parse(date) / 1000 + 6000,

};

var policy = JSON.stringify(policy\_before\_md5);//将序列化成一个JSON字符串

policy = BASE64.encoder(policy);//Base64加密

//计算signature

var encrypt\_string = 'POST' + '&/hesion' + '&' + policy;

var signature = b64\_hmac\_sha1(pwd, encrypt\_string);//b64\_hmac\_sha1签名加密

var sign = 'UPYUN hesion:' + signature + '=';

//利用ng-file-upload上传

$scope.upload(policy, sign, file, successFunc);

};

//上传服务器

$scope.upload = function (policy, signature, file, successFunc) {

$scope.fileInfo = file;

Upload.upload({

//服务端接收

url: FILE\_URL,

withCredentials: false,

//上传的同时带的参数

data: {

'policy': policy,

'file': file,

'authorization': signature

},

}).success(successFunc, function (data, status, headers, config) {

//上传成功 图片所处的URL

$scope.url = data.url;

}).error(function (data, status, headers, config) {

//上传失败

swal("图片上传失败，" + 'error status: ' + status)、

});

};

## JsTree实现树形结构

jsTree是基于JQuery的一个跨浏览器树控件，功能强大，而且是免费的。它支持HTML和JSON数据源，容易改写和扩展，可进行编辑和配置。项目中在选择产品类别时，使用了jsTree控件，帮助用户一目了然的看到产品类别的多级树形结构，实现效果如下图：



图 6‑2 jsTree实现许可范围树形结构

核心代码如下：

goodsService.queryAllGoodslevel(87, $scope.filter.hashData.time, $scope.filter.hashData.rannum, $scope.filter.hashData.signature).then(function (result) {

for (var i = 0; i < result.length; i++) {

$scope.filter.data.push(result[i]);

}

$('#goodsTree').jstree({

'core': {

'data': $scope.filter.data,

"themes": {

"theme": "classic",

"dots": true,

"icons": false

},

},

"plugins": ["themes", "json\_data", "search", "checkbox"]

});

}).finally(function () {

App.unblockUI('#goodsTree'); //取消加载提示图片

});

$('#goodModal').modal('show');

}

//搜索功能

$scope.searchGoods = function () {

if ($scope.model.serarText) {

showWait();

var serartext = {

'key\_word': $scope.model.serarText,

};

var hashData = hashSignature(serartext, $rootScope.identity.token);

enterprisesService.searchKeyWord(121, hashData.time, hashData.rannum, hashData.signature, $scope.model.serarText).then(function (result) {

if (result[0].goodsLevel) {

for (var i = 0; i < result[0].goodsLevel.length; i++)

$("#goodsTree").jstree('check\_node', result[0].goodsLevel[i].code);

} else {

swal("未搜索到相关许可范围!");

}

}).finally(function () {

$scope.model.serarText = '';

hideWait();

});

}

};

## 支付宝、微信支付

支付宝、微信支付已经是人们流行的在线支付方式，极大便捷了人们的生活，调用支付宝、微信支付API，需要监听支付情况，以提示用户是否支付成功，为用户提供良好的支付体验，实现效果如下图所示。



图6-3微信支付

具体实现代码如下：

//提交支付信息

$scope.commitOrder = function(isValid) {

//判断表单是否合法

if(!$scope.model.money || !isValid) {

return;

}

//判断是否正在监听支付

if($scope.timer) {

$interval.cancel($scope.timer);

}

var model = {

"c\_id": $scope.model.c\_id,

"user\_id": $scope.model.user\_id,

"money": $scope.model.money,

"pay\_mode": $scope.model.pay\_mode

};

console.log(JSON.stringify(model));

var hashData = hashSignature(model, $rootScope.identity.token);

feeService.commitOrder(88, hashData.time, hashData.rannum, hashData.signature, model).then(function(result) {

$scope.qrcodeModel = result[0];

$scope.qrcode.makeCode($scope.qrcodeModel.qr\_code);//生成支付二维码

$("#pay").modal('show');//显示二维码

//启动支付监听

$scope.timer = $interval(function(){

$scope.queryFeeRechargeOrderStatus($scope.qrcodeModel.order\_num);

console.log($scope.qrcodeModel.order\_num);

}, 10000);

});

};

//监听支付

$scope.queryFeeRechargeOrderStatus = function(order\_num) {

var model = {

order\_num: order\_num

};

console.log(JSON.stringify(model));

var hashData = hashSignature(model, $rootScope.identity.token);

feeService.queryFeeRechargeOrderStatus(93, hashData.time, hashData.rannum, hashData.signature, model).then(function(result) {

$scope.qrcodeModel.order\_status = result[0].order\_status;

switch($scope.qrcodeModel.order\_status) {

//支付成功

case '1':

case 1:

$("#pay").modal('hide');

swal("支付成功");

$interval.cancel($scope.timer);

$scope.checkBalance();

break;

//支付失败

case 2:

case '2':

$("#pay").modal('hide');

swal("支付失败");

$interval.cancel($scope.timer);

break;

}

});

};

## Blob对象实现导出Excel

导出台账数据Excel，极大地方便了用户获取账单数据，为公司对流水账的统计提供了快捷的方法。Blob 对象表示一个不可变、原始数据的类文件对象，可以实现数据excel表格导出，实现效果如下图说所示。

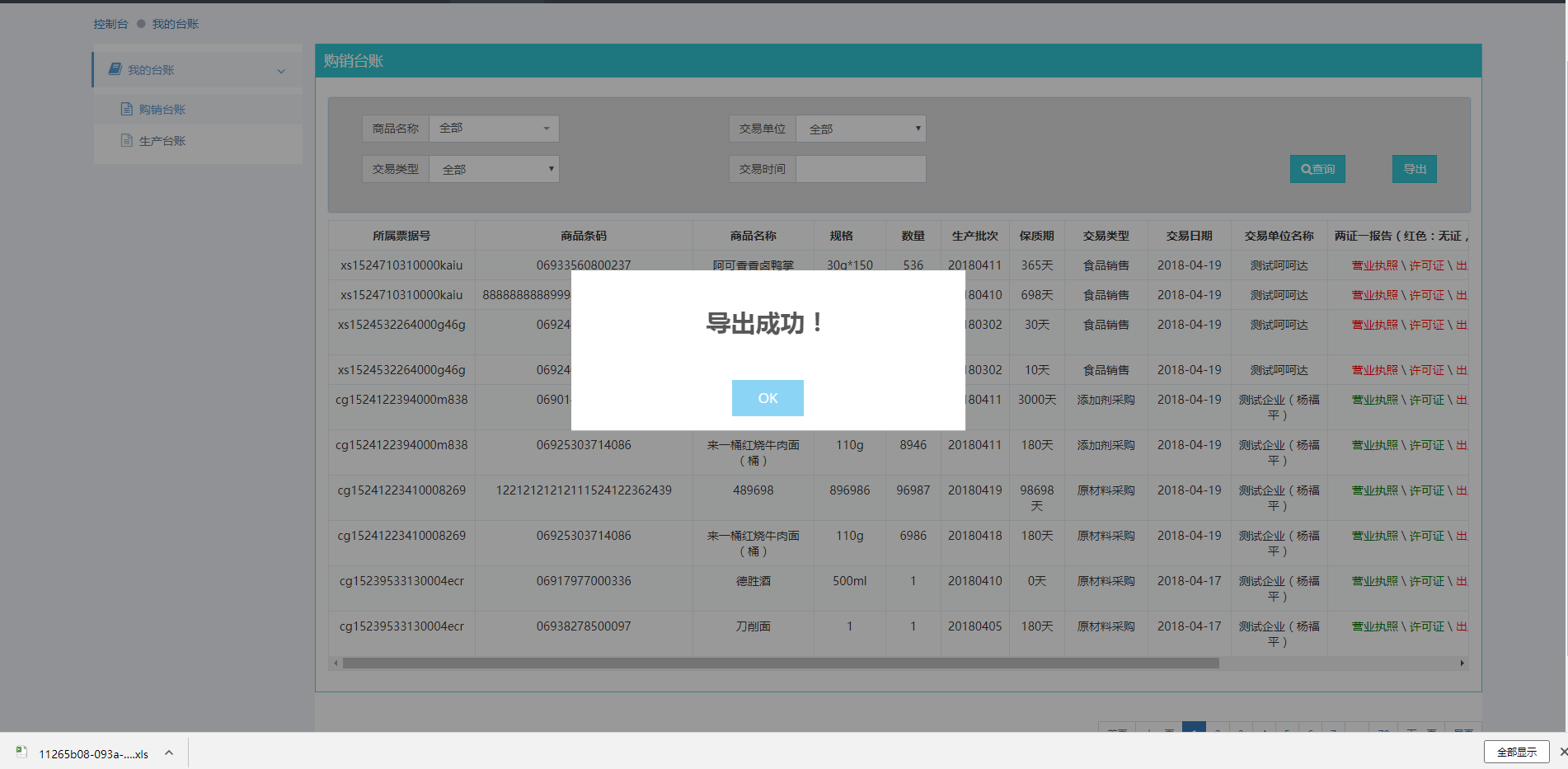


图6-4导出Excel表



图6-5Excel数据

实现代码如下所示

$http.post(url,queryParam,{responseType: 'arraybuffer'}).success(function (res) {

//begin

var blob = new Blob([res], {type: "application/vnd.ms-excel"});

var objectUrl = URL.createObjectURL(blob);

var aForExcel = $("<a><span class='forExcel'>下载excel</span></a>").attr("href",objectUrl);

$("body").append(aForExcel);

$(".forExcel").click();

aForExcel.remove();

//end\_date

hideWait();

swal('导出成功！');

})

## 数据合理性和完整性验证

前端对于用户输入的数据需要做合理性和完整性的验证，一方面是验证用户输入的内容是否符合规范，另一方面是检验必填项是否填入，对于不合法的输入给出提示，这样做不仅能提高用户体验，还能减轻服务器的压力。本项目的前端验证使用AngualrJS的指令来完成。下面介绍几个常用指令：

1. required 必填项
2. ng-minlength 输入的最小长度，ng-minlength=”{number}”
3. ng-maxlength 输入的最大长度，ng-maxlength=“{number}”
4. ng-pattern 正则表达式验证，ng-pattern="/PATTERN/"

例如对手机号的验证代码如下：

<input class="form-control " type="text" autocomplete="off" placeholder="手机号" name="username" id="loginUsername" ng-minlength="11" ng-maxlength="11" ng-model="model.user\_tel" ng-pattern="/^1[3|4|5|8|6|7][0-9]\d{8}$/" required />



图 6‑3登录验证

## 提示框插件SweetAlert

SweetAlert是一个漂亮的、响应式的能代替传统Javascript Alert的弹出框。它不需要JQuery的支持，使用原生的JavaScript编写。SweetAlert允许自定义，能设置弹框的类型，包括"warning","error", "success" and "info"四种内置的类型，还能设置弹窗的标题、内容、按钮样式、点击后的回调函数等。以退出的提示框为例，效果图如下：



实现代码如下：

swal({

title: "确认退出？",

type: "warning",

showCancelButton: true,

confirmButtonText: "是",

cancelButtonText: "否",

closeOnConfirm: true

}, function () {

var param=[];

var hashData = hashSignature('', $rootScope.identity.token);

customerService.loginOut(2,hashData.time,hashData.rannum,hashData.signature,roleid, param).then(function(){

$cookieStore.put('customer', null);

$cookieStore.put('menus', null);

$rootScope.identity = null;

$rootScope.systemMenus = null;

Restangular.setDefaultHeaders({});

window.location.href = "app.html#/login";

})

})

## 本章小结

本章主要介绍了系统的关键技术及具体实现方法，首先介绍了Echarts实现统计图表展示，接着介绍了使用百度地图API显示商品分布，利用jsTree插件实现树形结构的展示，最后介绍了漂亮美观的SweetAlert提示框，为用户提供一个美观大方、可操作性强的系统。

# 成果与展望

本文通过较为完善的需求分析与可行性缝隙，开发汇票线手机APP，目前基础功能已经基本成型，但是任何人都知道一个好的产品需要不断的测试以及不断的改进，需要继续吸收其他不同的意见，接下来还会有很多工作需要开展。

## 研究总结

本文的主要工作如何：

1. 结合当前国内的票据市场，并且分析部分国外经济发达国家的票据市场，使用了现有的国内的票据交易系统，通过整理分析比较，设计了本项目具体实现需求以及方案。

2. 对本APP的技术方案进行了详细的介绍，进行了可行性分析，验证其可行性。

3. 根据需求分析进行数据库的设计与优化，给出了详细的数据字典，然后依托于阿里云设计了合适的系统物理架构，设计了前后端缝隙的逻辑结构并依此进行系统的实现。

4. 实现了汇票线APP的主要功能，并对部分主要的核心功能模块以及关键技术进行代码分析，如票据交易流程，以及前后端分离代码的实现等。

5. 对APP的打包以及后台的部署进行了介绍，并且进行相应的测试工作，保证了APP运行的正确性以及稳定性。

## 不足之处

虽然前期进行较为完善的需求分析也进行充分的理论论证，但是难免仍然会出现一部分缺漏以及错误，在测试时候仍需汲取各方面意见进行改正。

在技术方面主要从这几方面进行改正：

1. 后台微服务架构中加入API Gateway中间件，对所有的API调用进行统一的认证以及接口转发，保证接口的安全性。
2. 后台微服务中间的调用改为rpc方式，数据序列化采用protobuf3。
3. 修改部分CSS页面，利用Ionic原生的标签库以及CSS组件库，保证页面针对不同屏幕大小的兼容性。
4. 增加数据库索引，并且优化数据库读写策略，实现数据库读写分离等工作，保证数据量扩大时的读写优化。

## 后续展望

随着现在票据市场的不断扩大，票据流通也会愈发频繁，并且如今互联网技术不断的提高，票据交易移动化也是十分迫切存在的需求，所以在保证票据交易安全性的条件下，促进票据流通更加便捷显得十分重要。因此本APP也是顺应“互联网+”的一种体现，愿意为促进票据交易信息化做出一份尝试。

参考文献

1. 候威, 银行电子商业汇票系统的设计和实现. 2010, 吉林大学
2. 郭建华 ,应收票据的核算 2001,中国会计电算化(04): 62-63
3. 舒雄 ,我国电子票据市场的现状、制约因素和发展路径, 广西金融研究,2008,(6):37-39
4. Stijn Claessens,Thomas Glaessner and Daniela Klingebid.Electronic Finance:Reshaping the Financial Landscape Around the World[J].Journal 0f Financial Services Research,2002,(8):29-61
5. Commercial Paper.British Bankers' Association. www.bba.org.uk.2007,6
6. 李震山 (2010). 电子商业汇票交易系统的研究与设计, 湖南大学.
7. Asset Purchase Facility Commercial Paper.Bank of england . www.bankofengland.co.uk/markets/apf/cp/index.htm.2005.9
8. 吴丽薇 (2011). 我国票据市场发展现状及对策研究, 海南大学.
9. http://money.163.com/16/0905/16/C07A6J73002580S6.html
10. 舒雄 (2008). "我国电子票据市场的现状、制约因素和发展路径." 广西金融研究(06): 37-39.
11. 孟培 网络银行票据对账系统的开发应用，中国金融电脑，2011,05:35-37
12. 中国工商银行票据营业部课题组（2015），“商业银行创新“互联网+票据”业务模式的思考”，上海金融学院学报（06）
13. 竺佳蕾（2009），银行电子票据业务发展存在问题分析及对策研究，金融会计
14. 王姝，“电子票据法律制度研究”，《鞍山科技大学学报》，2006 年第 l 期。
15. 厉群 (2006). 基于J2EE平台的商业银行票据管理系统的设计与实现, 山东大学.
16. 朱凯南, et al. (2016). "基于Ionic和Cordova的跨平台移动APP的研究与应用." 电脑知识与技术(01): 119-121.
17. 陈培培 (2011) “基于信誉度的云环境下资源管理的研究” 青岛大学
18. 马艳霞 (2015)“基于云计算的教育信息化应用探讨” 课程教育研究(25)
19. 廖坚 (2014) “虚拟机技术在《计算机系统与维护》教学中的应用” 计算机光盘软件与应用（11）
20. 徐锦韬 (2016) “虚拟机、容器与Docker技术对比” 科学与财富(06)
21. 梁合功 (2016) “胜利油田异构数据迁移系统设计与实现” 上海交通大学
22. 肖莉 (2012) “网站建设中如何选择数据库的标准” 品牌与标准(02)
23. <http://www.xue163.com/2377/1/23775925.html>
24. <http://blog.csdn.net/zhangxing52077/article/details/52864132>
25. 杨建辉 江标荣 康震 曾维 (2009) “基于SMS的多协议交通信息发布系统研究与设计” 第五届中国智能交通年会暨第六届国际节能与新能源汽车创新发展论坛
26. 李恒 (2012) “基层连队人员管理系统设计与实现” 合肥工业大学
27. 江萍 (2008) “浅谈票据业务的发展与风险控制” 现代经济（现代物业下半月刊）

致 谢

光阴飞逝，转眼间大学四年的本科生涯即将结束。这是一段我未曾预料到如此精彩纷呈的旅途，感谢一路上家人、老师、同学、朋友的关心和照顾。这一路上，感恩母校“厚德致远，博学敦行”的谆谆教诲，感恩家人、老师、朋友的陪伴与支持，感恩挫折与失败带来的磨砺，也感恩自己的不气馁和韧性。

感谢默默付出的父母，他们是我最坚实的后盾，不仅为我提供良好的物质条件，让我能专心的学习、追求自己的目标，还在精神上给予我最大的信任与鼓励，帮助我在面临挑战时勇敢应对。

然后感谢各位老师多年来的悉心教导，成为我前行的指路明灯。特别感谢我的导师沈炜教授，本论文从选题到完成也倾注了大量的心血，为我提供有益的指导。沈老师渊博的计算机专业知识，严谨的教学态度，精益求精的工作作风，对我影响深远。在实验室中我得到了足够多的机会来锻炼自己的能力，也得到了很多师兄师姐的帮助。在此，向沈老师以及实验室的所有伙伴表示衷心的感谢！

生命短暂而脆弱，希望我们都能用这唯一的生的机会，追求梦想、实现自我，无悔岁月。