

# production\_ssm

生产管理

ERP

SSM

[production\\_ssm](#)

[项目简介](#)

[项目技术架构\(Spring+SpringMVC+Mybatis \)](#)

[系统架构](#)

[数据库设计 \( 详见sql文件 \)](#)

[软件运行说明](#)

[登录界面](#)

[运行界面](#)

[图片上传](#)

[文件上传](#)

[富文本编辑](#)

[关联信息](#)

[search-box查找](#)

[session过期跳转登录](#)

[动态权限控制](#)

[备注](#)

[导入项目到STS](#)

[文件上传配置](#)

[idea classpath配置](#)

[Mybatis逆向工程](#)

## 项目简介

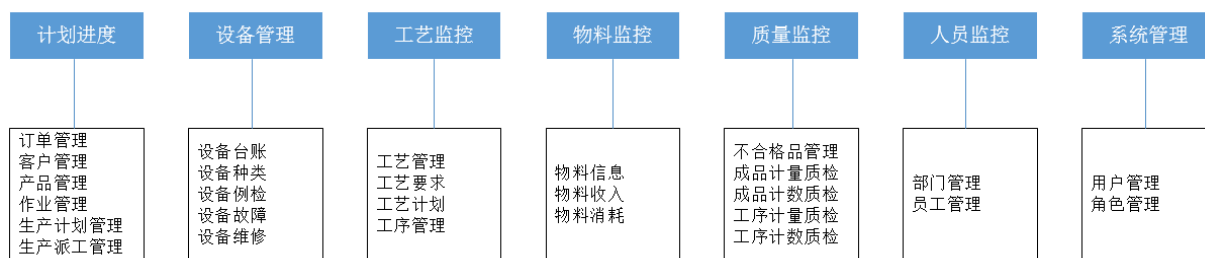
production\_ssm是一个生产管理ERP系统。依托科技计划重点项目“制造装备物联及生产管理系统研发”，主要包括：计划进度、设备管理、工艺监控、物料监控、人员监控、质量监控、系统管理7大模块。

## 项目技术架构 (Spring+SpringMVC+Mybatis )

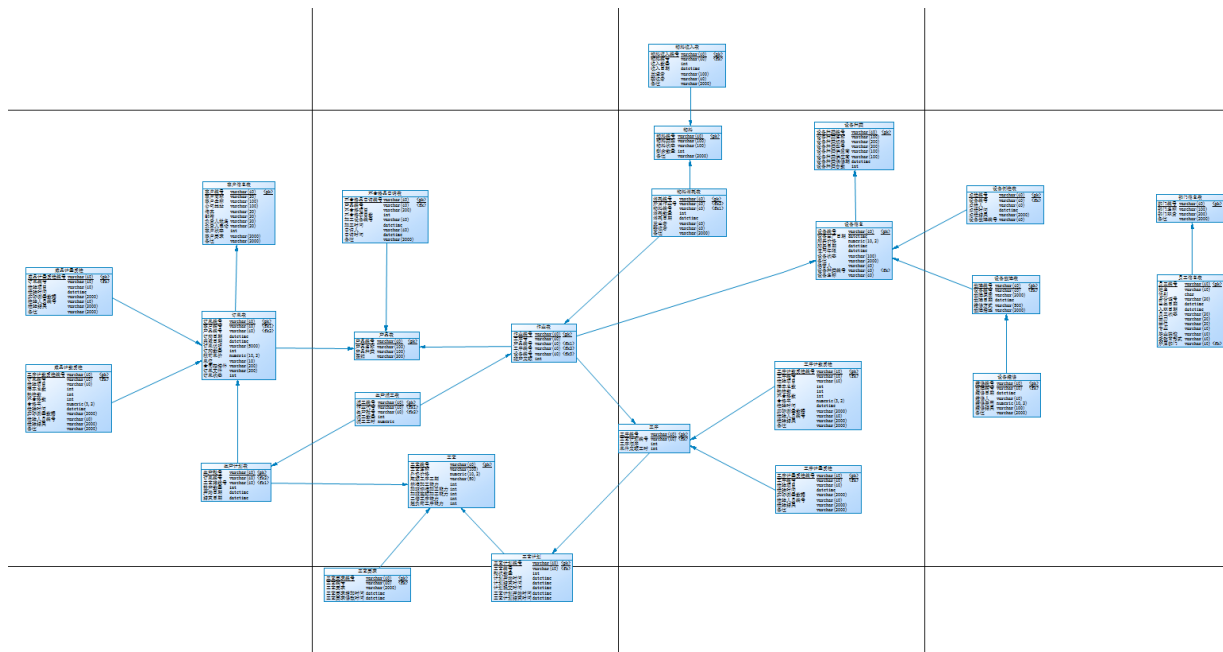
- Maven
- Spring ( IOC DI AOP 声明式事务处理 )
- SpringMVC ( 支持Restful风格 )
- Hibernate Validator ( 参数校验 )

- Mybatis ( 最少配置方案 )
- shiro权限控制,结合ajax实现了异步认证与异步授权，同时实现了细粒度的权限动态分配（到按钮级别）；添加了shiro session过期的登录跳转
- jQuery EasyUI开发前端页面，利用jQuery文件上传插件实现拖拽上传的效果并对文件类型、大小、数量进行控制；利用search-box实现查找功能
- Druid ( 数据源配置 sql防注入 sql性能监控)
- 统一的异常处理
- JSP JSTL JavaScript
- kindeditor富文本编辑器，处理图片上传和富文本编辑

## 系统架构



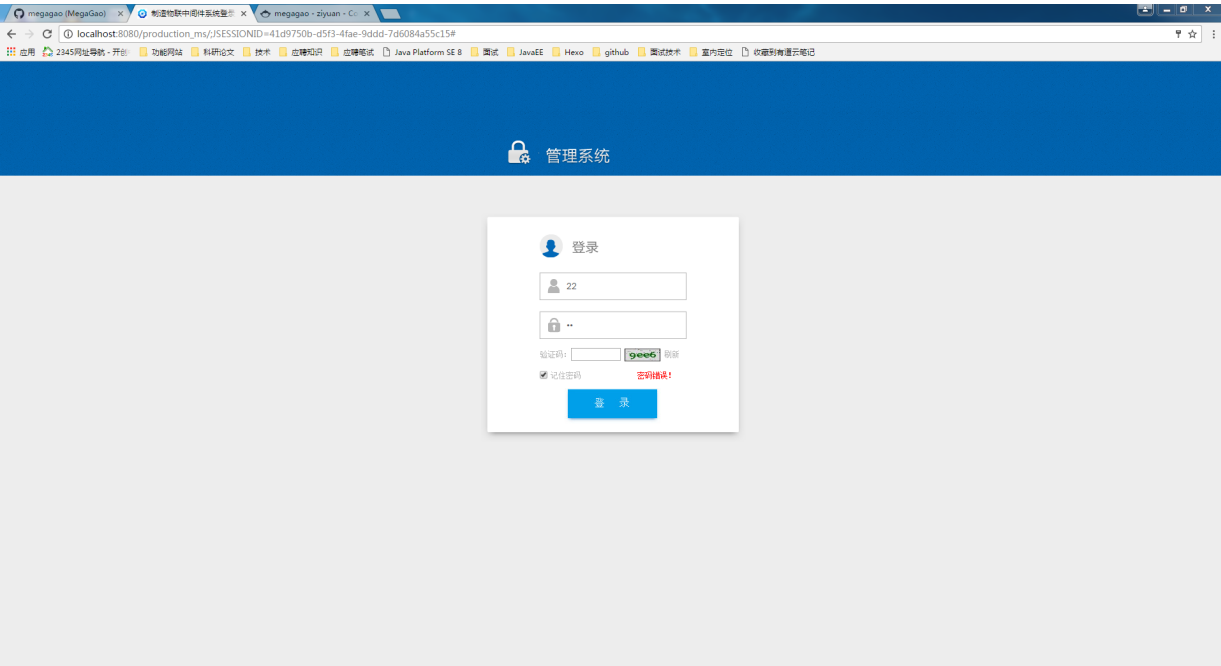
## 数据库设计 ( 详见sql文件 )



## 软件运行说明

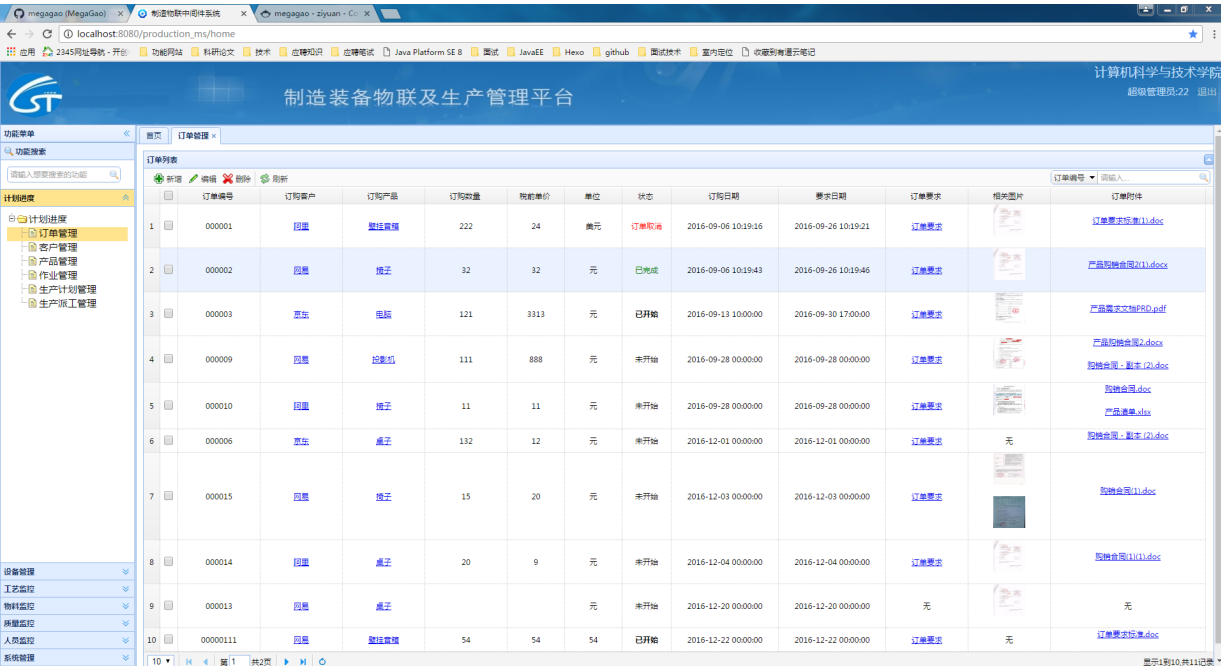
### 登录界面

登录可使用账号：22，密码：22的超级管理员登录，若密码输错，下次登录需输入验证码。



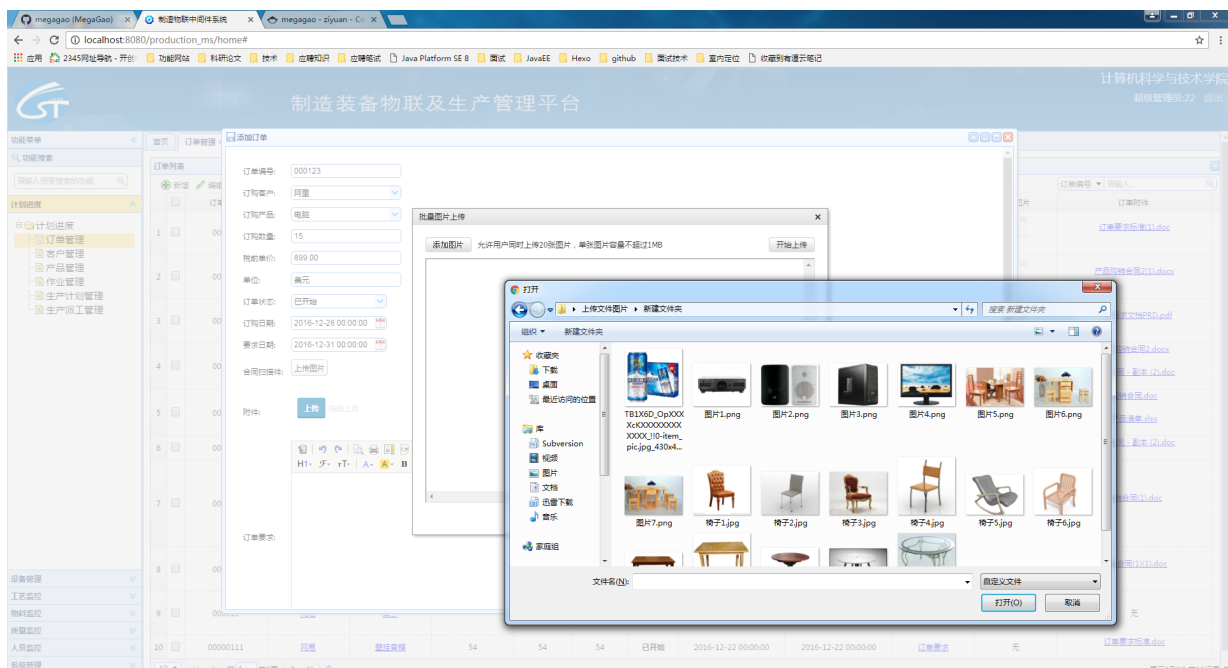
## 运行界面

超级管理员可显示系统管理模块进行系统权限分配与管理，其他角色可查看除系统管理外的剩余模块的信息（包括下载附件、查看图片等），但是只能维护该角色对应权限内的信息。左边功能搜索栏可进行功能模糊查找。



## 图片上传

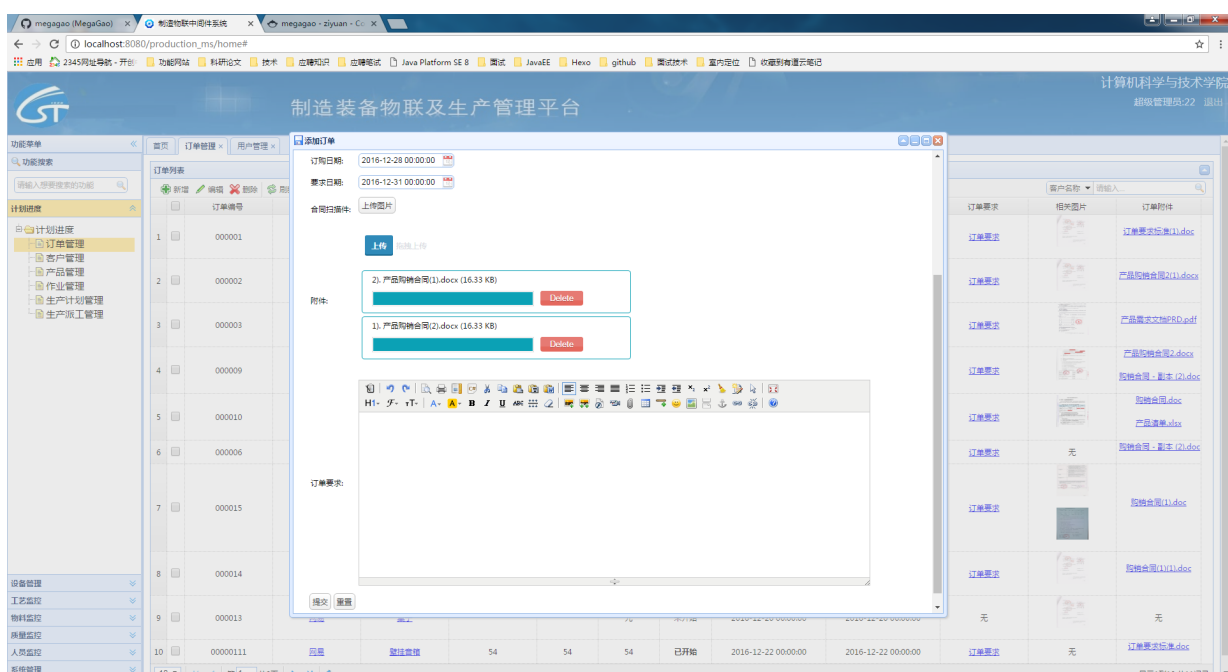
图片上传的配置请查看文档尾部的注释，图片大小要求不能超过1M，支持jpg、png等多种格式的图片，上传成功后可在相应的展示栏进行回显。



## 文件上传

文件上传使用了一个开源的jQuery文件上传插件，可以在common.js里面修改上传文件的参数，包括上传个数，支持的文件类型等，配置信息如下：

```
url:"file/upload",
maxFileCount: 5, //上传文件个数（多个时修改此处）
returnType: 'json', //服务返回数据
allowedTypes: 'doc,docx,excel,sql,txt,ppt,pdf', //允许上传的文件式
showDone: false, //是否显示"Done"(完成)按钮
showDelete: true, //是否显示"Delete"(删除)按钮
```

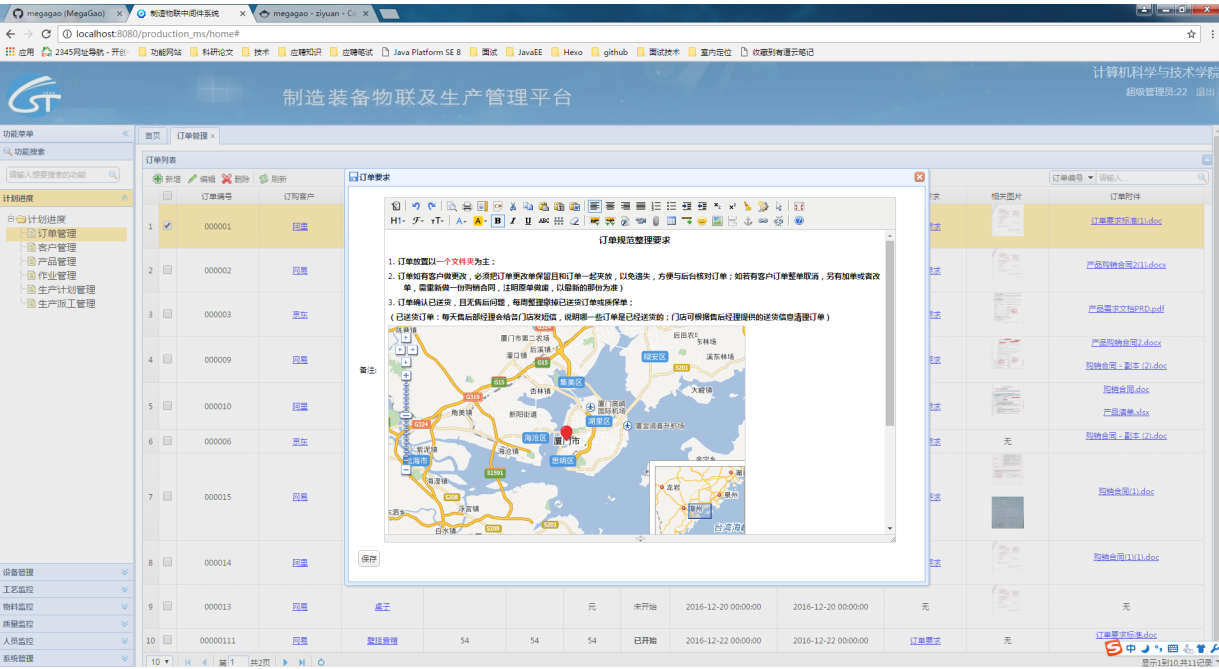


# 富文本编辑

本系统采用了开源的KindEditor富文本编辑器，它是一套在线HTML编辑器，主要用于让用户在网站上获得所见即所得编辑效果。KindEditor把传统的多行文本输入框(textarea)替换为可视化的富文本输入框。

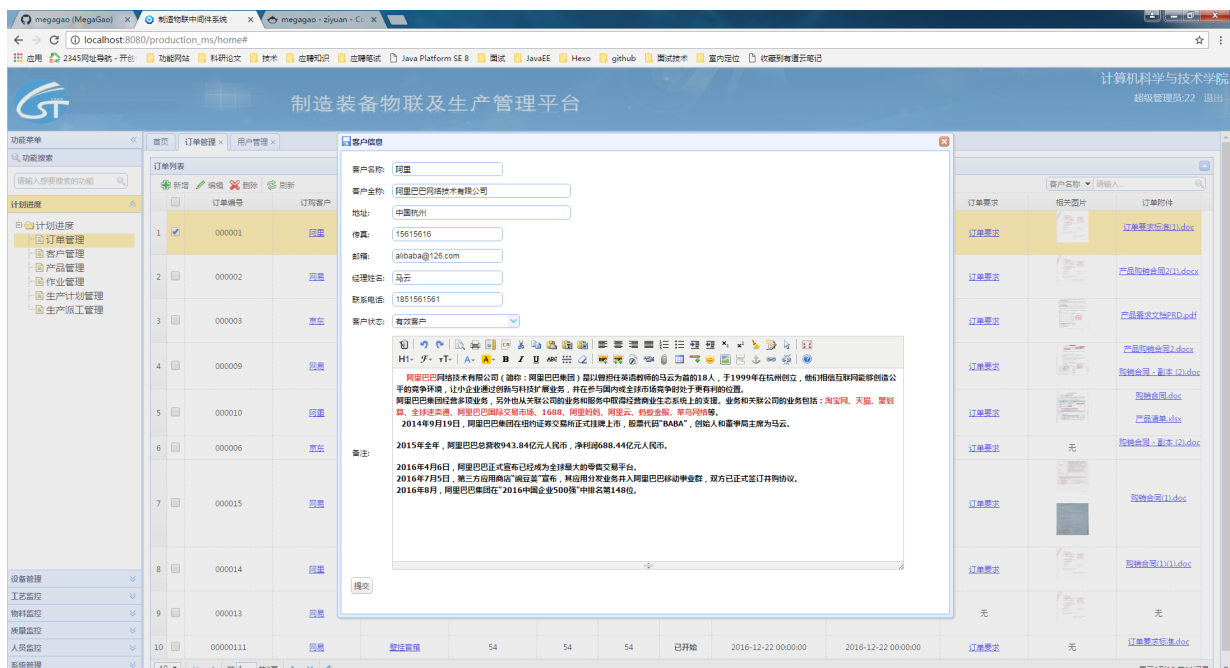
## KindEditor主要特点

- 快速：体积小，加载速度快
- 开源：开放源代码，高水平，高品质
- 底层：内置自定义 DOM 类库，精确操作 DOM
- 扩展：基于插件的设计，所有功能都是插件，可根据需求增减功能
- 风格：修改编辑器风格非常容易，只需修改一个 CSS 文件
- 兼容：支持大部分主流浏览器，比如 IE、Firefox、Safari、Chrome、Opera



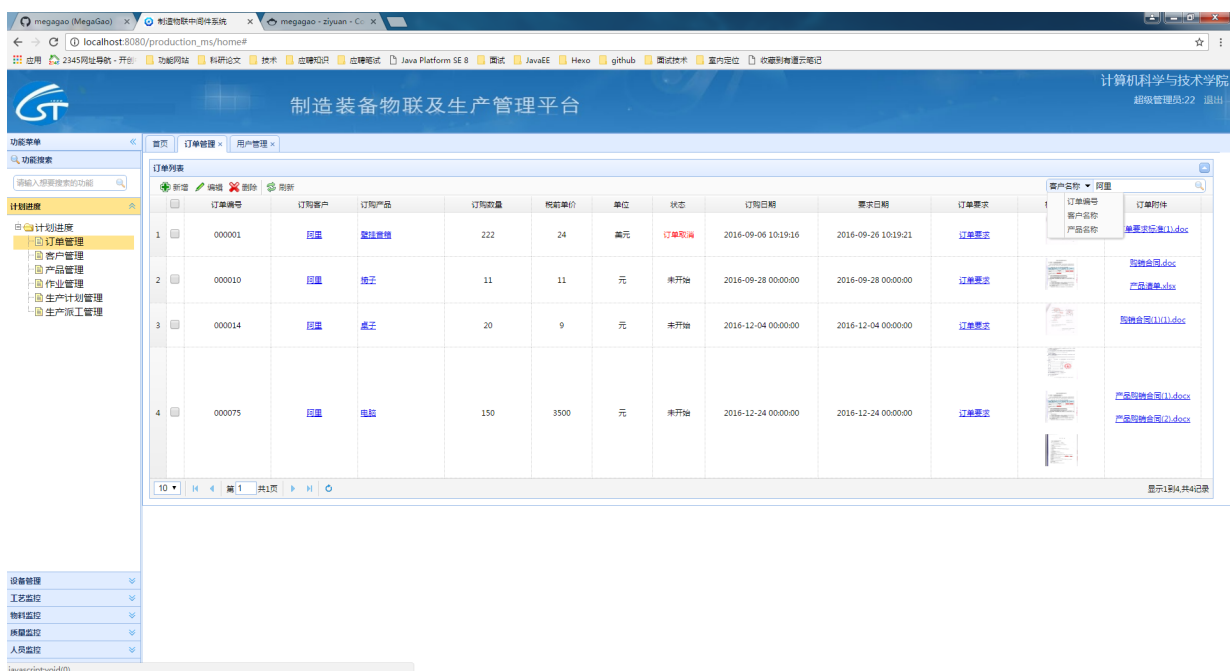
# 关联信息

关联对象的信息，点击以弹窗的形式显示，若具有该模块对应的修改权限，则也可在此进行信息维护。



## search-box查找

可以在右上角的search-box选择查询条件，输入关键字进行对应信息的模糊查找。



## session过期跳转登录

用户登录后，会建立相应的session，系统默认过期时间为10分钟，若需更改，可在applicationContext-shiro.xml配置文件中，更改如下配置。

```
<!-- 会话管理器 -->
<bean id="sessionManager" class="org.apache.shiro.web.session.mgt.DefaultWebSessionManager">
    <!-- session的失效时长，单位毫秒，这里设置为10分钟-->
    <property name="globalSessionTimeout" value="600000"/>
    <!-- 删除失效的session -->
    <property name="deleteInvalidSessions" value="true"/>
    <!-- 指定本系统sessionId，默认为：JSESSIONID 问题：与Servlet容器名冲突，如Jetty，Tomcat等默认JSESSIONID，
        当跳出shiro Servlet时如Error-page容器会为JSESSIONID重新分配值导致登录会话丢失！ -->
    <property name="sessionIdCookie" ref="sessionIdCookie"/>
</bean>
```

若session过期，则应跳转登录界面重新登录。系统采用的方式是设置一个sessionfilter，如下：

```

public class SessionFilter implements Filter {

    public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {

        HttpServletRequest request = (HttpServletRequest) servletRequest;
        HttpServletResponse response = (HttpServletResponse) servletResponse;
        if (!SecurityUtils.getSubject().isAuthenticated()) {
            //判断session里是否有用户信息,且是否为ajax请求, 如果是ajax请求响应头会有, x-requested-with
            if (request.getHeader("x-requested-with") != null
                && request.getHeader("x-requested-with").equalsIgnoreCase("XMLHttpRequest")) {
                response.setHeader("session-status", "timeout");//在响应头设置session状态
            }
        }
        filterChain.doFilter(request, servletResponse);
    }

    @Override
    public void destroy() {
        // TODO Auto-generated method stub
    }

    @Override
    public void init(FilterConfig arg0) throws ServletException {
        // TODO Auto-generated method stub
    }
}

```

并在web.xml中配置该filter。当session过期时，在响应头设置session状态为timeout，然后采用全局的ajax事件对session状态进行判断，跳转到登录页面。该事件定义在common.js中，如下：



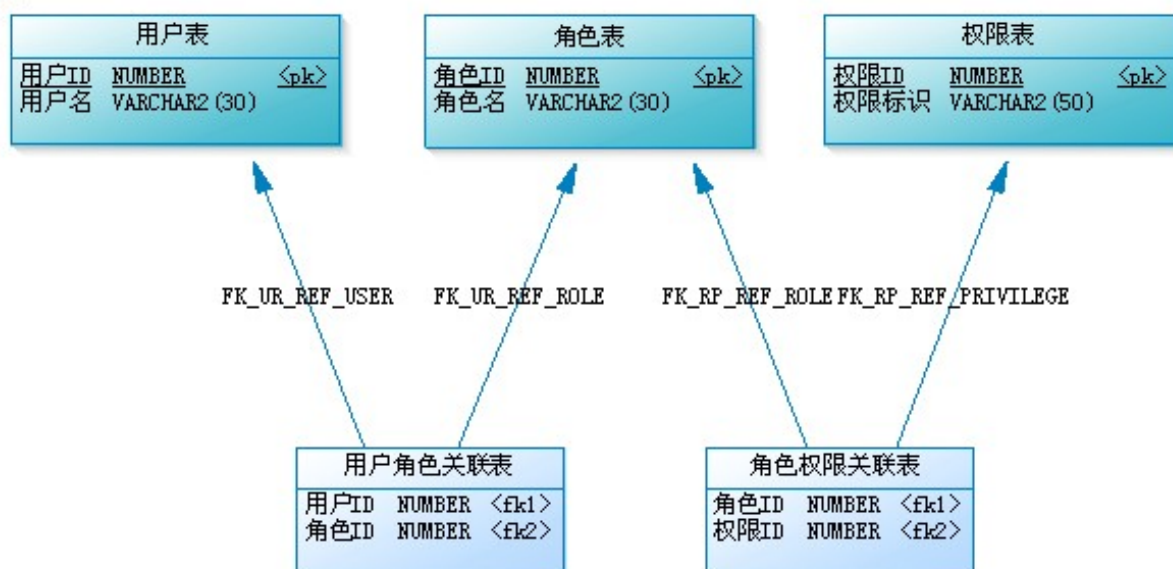
```
//全局ajax事件，处理session过期跳转登录
$.ajaxSetup({
    complete:function(XMLHttpRequest,sessionStatus){
        var sessionstatus = XMLHttpRequest.getResponseHeader("sessi
on-status");
        if(sessionstatus=="timeout"){
            $.messager.alert('提示信息', "登录超时！请重新登录！", 'inf
o',function(){
                window.location.href = 'login';
            });
        }
    }
});
```

关于这部分的详细内容请参考我的一篇博客：[利用filter和全局ajax事件实现shiro session过期登录跳转](#)

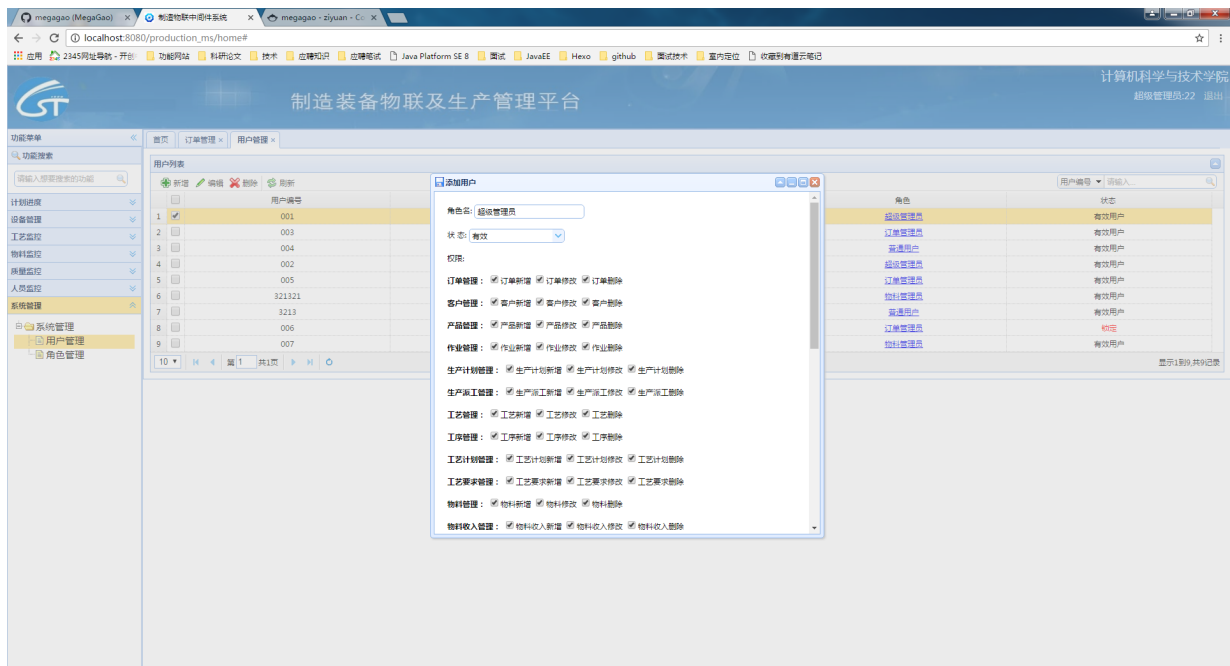
## 动态权限控制

本系统采用经典的权限模型，即RBAC（Role-Based Access Control）基于角色的访问控制，即用户通过角色与权限进行关联。模型用到5张表：用户表、角色表、权限表、用户角色表、角色权限表。简单地说，一个用户拥有若干角色，每一个角色拥有若干权限。这样，就构造成“用户-角色-权限”的授权模型。在这种模型中，用户与角色之间，角色与权限之间，一般者是多对多的关系。

该模型可简化表示为下图：



本系统基于RBAC权限模型，采用shiro框架进行权限控制。只有角色为超级管理员的用户才能进行系统的权限管理，权限级别细化到菜单选项。



## 备注

## 导入项目到STS

感谢一位韩国道友给我发的邮件，介绍了将production\_ssm导入到STS中，并将详细过程与注意事项完整记录了下来。非常感谢他的文档，也为STS的使用提供了参考。

参见：[导入production\\_ssm到STS](#)

## 文件上传配置

在本地建立上传图片 and 文件的文件夹，如我的存放路径是在D:\upload\temp\img，D:\upload\temp\file文件夹下，然后修改tomcat的配置文件server.xml，添加虚拟路径，将对图片和文件的请求url映射到本机硬盘的相应路径，如下：

```

<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">
  <!-- SingleSignOn valve, share authentication between web applications
      Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

  <!-- Access log processes all example.
      Documentation at: /docs/config/valve.html
      Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t \"%r\" %s %b" />

  <!-- 在Host标签下添加下面两行，配置虚拟路径到你本机的文件夹 -->
  <Context path="/pic" docBase="D:\upload\temp\img" crossContext="true" trusted="true" reloadable="true"/>
  <Context path="/file" docBase="D:\upload\temp\file" crossContext="true" trusted="true" reloadable="true"/>
</Host>

```

## idea classpath配置

idea引入项目后，resources目录在eclipse中是在classpath下的，而在idea中变成在classpath外，导致项目无法识别配置文件。解决办法是把resources文件夹加入到classpath中，请参照此博文操作：[eclipse与idea中classpath配置路径不同导致迁移项目时的FileNotFoundException问题](#)

## Mybatis逆向工程

系统使用了Mybatis的逆向工程，依据数据库表自动生成domain和mapper（注：自动生成的代码都是针对单表，若需多表整合，则要手动修改实现），其中，针对每个数据库表，都会生成两个封装对象，可以认为example对象是对其相应的ORM映射对象查询条件的封装。逆向工程的实现—Mybatis Generator的代码托管在<https://github.com/mybatis/generator>，代码down下来后，配置generatorConfig.xml文件，如下：

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration
  PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration
  1.0//EN"
  "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">

<generatorConfiguration>
  <context id="testTables" targetRuntime="MyBatis3">

    <commentGenerator>
      <!-- 是否去除自动生成的注释 true: 是 : false:否 -->
      <property name="suppressAllComments" value="true" />
    </commentGenerator>

    <!--数据库连接的信息：驱动类、连接地址、用户名、密码 -->
    <jdbcConnection driverClass="com.mysql.jdbc.Driver"
      connectionURL="jdbc:mysql://localhost:3306/production_ms"
      userId="root"
      password="root">
    </jdbcConnection>

    <!-- 默认false，把JDBC DECIMAL 和 NUMERIC 类型解析为 Integer，为 true时把JDBC DECIMAL 和
      NUMERIC 类型解析为java.math.BigDecimal -->
    <javaTypeResolver>
      <property name="forceBigDecimals" value="false" />
    </javaTypeResolver>

    <!-- targetProject:生成PO类的位置 -->
    <javaModelGenerator targetPackage="org.hqu.production_ms.domain"
      main"
      targetProject=".\\src">
      <!-- enableSubPackages:是否让schema作为包的后缀 -->
      <property name="enableSubPackages" value="false" />
      <!-- 从数据库返回的值被清理前后的空格 -->
      <property name="trimStrings" value="true" />
    </javaModelGenerator>

    <!-- targetProject:mapper映射文件生成的位置 -->
    <sqlMapGenerator targetPackage="org.hqu.production_ms.mapper"
      r"
      targetProject=".\\src">
      <!-- enableSubPackages:是否让schema作为包的后缀 -->
      <property name="enableSubPackages" value="false" />
    </sqlMapGenerator>

    <!-- targetPackage: mapper接口生成的位置 -->
    <javaClientGenerator type="XMLMAPPER"

```

```
        targetPackage="org.hqu.production_ms.mapper"
        targetProject=".\\src">
        <!-- enableSubPackages:是否让schema作为包的后缀 -->
        <property name="enableSubPackages" value="false" />
    </javaClientGenerator>

    <!-- 指定数据库表 -->
    <table schema="" tableName="unqualify_apply"></table>

</context>
</generatorConfiguration>
```

修改完成后，执行main方法即可。需要注意的是，若要重新生成，则需把已经生成的文件删除，因为它不会自己覆盖，导致文件混乱。关于Mybatis逆向工程的更多详细信息，读者可以自行上网查阅。