燕山大学****

软件工程课程设计报告

学 院 信息科学与工程学院(软件学院)

年级专业 22级计算机

学生姓名 蔡远 周彧函

摆轶轩 刘聪瑞

指导教师 于洪俊

实习日期 2024年6月17日--

2024年7月5日

目录

[摘 要 II](#_Toc170925926)

[第1章 绪论 1](#_Toc170925927)

[**1.1 项目背景** 1](#_Toc170925928)

[**1.2 项目意义** 1](#_Toc170925929)

[第2章 开发工具及相关技术简介 3](#_Toc170925930)

[**2.1 开发工具介绍** 3](#_Toc170925931)

[**2.1.1 前端使用工具及简介** 3](#_Toc170925932)

[**2.1.2 后端使用工具及简介** 3](#_Toc170925933)

[**2.1.3 数据库使用工具及简介** 4](#_Toc170925934)

[**2.2 相关技术及有关应用** 4](#_Toc170925935)

[**2.2.1 前端开发技术** 4](#_Toc170925936)

[**2.2.2 后端使用技术** 6](#_Toc170925937)

[**2.2.3 数据库** 7](#_Toc170925938)

[**2.3 本章小结** 8](#_Toc170925939)

[第3章 软件开发工作 8](#_Toc170925940)

[**3.1 需求分析** 8](#_Toc170925941)

[**3.1.1概述** 8](#_Toc170925942)

[**3.1.2角色与权限** 8](#_Toc170925943)

[**3.2 用例图** 10](#_Toc170925944)

[**3.2.1总用例图** 10](#_Toc170925945)

[第4章 原型设计 13](#_Toc170925946)

[**4.1业务流程图** 13](#_Toc170925947)

[**4.2页面原型图** 14](#_Toc170925948)

[**4.3需求规约** 27](#_Toc170925949)

[**4.3.1 系统概述** 27](#_Toc170925950)

[**4.3.2 功能需求** 27](#_Toc170925951)

[**4.3.3 非功能性需求** 28](#_Toc170925952)

[第5章 系统设计 29](#_Toc170925953)

[**5.1数据库设计** 29](#_Toc170925954)

[**5.1.1ER图** 29](#_Toc170925955)

[**5.1.2表关系图** 30](#_Toc170925956)

[**5.2状态图** 31](#_Toc170925957)

[**5.3类图** 34](#_Toc170925958)

[**5.4 订单处理时序图** 34](#_Toc170925959)

[第6章 重要用例：动态显示当前工厂的情况 34](#_Toc170925960)

[小组成员分工及自评贡献度级别 48](#_Toc170925961)

# **摘 要**

东软智能制造云平台旨在通过现代化的工业4.0智能制造生态链，优化工厂生产流程，提升生产效率。系统主要功能模块包括注册、登录、首页展示、产品管理、设备管理、订单管理、生产计划管理、生产调度管理以及生产跟踪等。用户可以实现对生产流程的全面在线处理和业务需求的规范化管理。

**主要功能**：

* **注册**：用户注册新工厂，默认管理员权限。
* **登录**：用户登录后获取工厂信息及菜单权限。
* **首页**：动态显示设备运行、生产效益及订单统计。
* **产品管理**：支持产品查询、新增、编辑及删除操作。
* **设备管理**：支持设备查询、录入、编辑及删除操作。
* **订单管理**：新建、接单、拒单、转计划及完成订单操作。
* **生产计划管理**：新建、启动及删除生产计划。
* **生产调度管理**：新建、启动及删除工单。
* **生产跟踪**：员工报工并更新生产进度。

本系统按照软件工程标准流程开发，采用前后端分离的架构，前端使用Vue，后端使用SpringBoot。本文档为用户全面了解平台设计、实施和应用提供重要参考。

**关键词**：智能制造云平台 前后端分离 SpringBoot Vue

# **第1章 绪论**

## **1.1 项目背景**

随着全球制造业向智能化、数字化方向的快速发展，工业4.0时代已经到来。传统制造业面临生产效率低下、资源浪费严重、生产管理不规范等诸多问题，亟需通过信息化手段进行转型升级。东软智能制造云平台正是基于这一背景应运而生。

在当前市场竞争日益激烈的环境下，制造企业迫切需要一种高效、灵活且智能化的生产管理系统，来提升生产效率、降低成本、优化资源配置，实现生产全过程的实时监控和管理。传统的生产管理模式已经无法满足现代制造业的需求，企业需要通过引入先进的管理系统来应对挑战。

东软智能制造云平台旨在解决制造企业在生产管理中遇到的实际问题，帮助企业实现生产流程的全面数字化和智能化。通过引入云计算、大数据和物联网等先进技术，平台能够为企业提供从生产订单处理、排产到生产过程监控的一体化解决方案，显著提升企业的生产管理水平和市场竞争力。

在平台的开发过程中，充分考虑了工业4.0的核心理念和智能制造的实际需求。系统设计以用户体验为中心，通过前后端分离的架构，实现了高效、灵活的系统交互。前端采用Vue技术，提供友好且易操作的用户界面；后端采用SpringBoot框架，确保系统的高性能和可扩展性。

东软智能制造云平台的推出，不仅帮助企业解决了生产管理中的痛点问题，也为制造业的智能化转型提供了强有力的技术支持。通过该平台，制造企业能够更好地适应市场变化，提升生产效率，实现可持续发展。

## **1.2 项目意义**

东软智能制造云平台的开发与实施具有重要的战略意义和实际价值，对于制造企业及整个工业4.0领域的发展都具有深远影响。

1. **提升生产效率**： 平台通过对生产流程的全面数字化管理，优化了订单处理、排产和生产调度等环节，显著提升了生产效率。企业能够更快速地响应市场需求，缩短产品交付周期，从而提升市场竞争力。
2. **实现精益生产**： 通过实时监控生产过程和设备运行情况，平台能够有效减少资源浪费，降低生产成本，提升产品质量。企业可以实现精益生产，最大化利用生产资源，提升整体效益。
3. **增强管理透明度**： 平台提供的动态数据展示和详细报表功能，使企业管理层能够随时掌握生产状况和运营情况，提升管理透明度和决策效率。实时数据分析和预警机制帮助企业及时发现并解决潜在问题，保障生产顺利进行。
4. **推动智能制造转型**： 东软智能制造云平台引入了云计算、大数据和物联网等先进技术，推动了制造业向智能化、数字化方向转型。平台实现了生产管理的智能化，提高了企业自动化程度，为未来进一步的技术创新和应用奠定了基础。
5. **提升市场竞争力**： 在市场竞争日益激烈的环境下，平台帮助企业提高了生产管理的灵活性和适应性，能够更快速地应对市场变化，满足客户需求。通过优化生产流程和提升产品质量，企业能够在市场中获得更多的竞争优势。
6. **实现可持续发展**： 平台通过优化资源配置和提高生产效率，减少了能源消耗和环境污染，推动了绿色制造和可持续发展。企业不仅能够获得经济效益，还能履行社会责任，实现经济效益和社会效益的双赢。

总之，东软智能制造云平台不仅是企业提高生产管理水平的有力工具，更是推动制造业智能化转型的重要引擎。通过平台的应用，制造企业能够实现生产流程的全面优化，提升市场竞争力，推动产业升级，为实现工业4.0的愿景提供坚实保障

# **第2章 开发工具及相关技术简介**

## **2.1 开发工具介绍**

在东软智能制造云平台的开发过程中，我们采用了现代化的开发工具和技术，以确保系统的高效性、稳定性和易维护性。具体来说，前端开发使用了WebStorm，后端开发使用了IntelliJ IDEA，数据库管理使用了Navicat。以下是对这些工具的详细介绍：

### **2.1.1 前端使用工具及简介**

WebStorm是一款由JetBrains开发的专业JavaScript集成开发环境（IDE），在前端开发中具有以下优势：

1. **智能编码辅助**： WebStorm提供了智能代码补全、错误检测和快速修复等功能，使开发者能够更快速、更准确地编写代码，提升开发效率。
2. **丰富的框架支持**： WebStorm对Vue.js有着良好的支持，包括Vue文件的语法高亮、代码补全和模板提示等功能，方便开发者进行Vue.js应用开发。
3. **强大的调试工具**： WebStorm集成了强大的调试工具，支持JavaScript、HTML和CSS的调试，使开发者能够快速定位和修复前端问题。
4. **版本控制集成**： WebStorm支持Git等主流版本控制系统，开发者可以方便地进行代码版本管理和团队协作，提高项目管理效率。

### **2.1.2 后端使用工具及简介**

IntelliJ IDEA是另一款由JetBrains开发的综合性Java集成开发环境（IDE），在后端开发中具有以下优势：

1. **智能代码提示和导航**： IntelliJ IDEA提供了对Java和SpringBoot的智能提示功能，包括代码补全、重构和导航，帮助开发者提高编码效率。
2. **内置调试工具**： IntelliJ IDEA集成了强大的调试工具，支持断点调试、变量监视和线程管理，使开发者能够快速定位和解决后端问题。
3. **插件丰富**： IntelliJ IDEA拥有丰富的插件生态，开发者可以根据需要安装各种插件来扩展IDE的功能，满足不同的开发需求。
4. **版本控制支持**： IntelliJ IDEA同样支持Git等版本控制系统，方便开发者进行代码管理和团队协作。

### **2.1.3 数据库使用工具及简介**

Navicat是一款专业的数据库管理工具，支持多种数据库系统，包括MySQL、PostgreSQL、SQLite等，在数据库管理中具有以下优势：

1. **用户友好的界面**： Navicat提供了直观的图形用户界面，使开发者能够方便地进行数据库连接、查询和管理操作。
2. **强大的数据管理功能**： Navicat支持数据库的导入导出、数据同步和备份恢复等功能，帮助开发者高效管理数据库数据。
3. **高级查询编辑器**： Navicat提供了功能强大的SQL编辑器，支持代码补全、语法高亮和查询优化，使开发者能够更高效地编写和调试SQL语句。
4. **安全连接**： Navicat支持SSH和SSL等安全连接方式，确保数据库连接的安全性。

## **2.2 相关技术及有关应用**

### **2.2.1 前端开发技术**

#### **2.2.1.1 vue3**

* **特性**：Vue 3 是一款现代化的渐进式JavaScript框架，专注于构建用户界面。
* **优势**：通过虚拟DOM和响应式数据绑定，实现高效的页面渲染和组件化开发。
* **详细介绍**：Vue 3 提供了更快的渲染速度和更小的包大小，支持Composition API和Teleport等新特性，使得开发者能够更加灵活地组织和重用代码。

#### **2.2.1.2 Axios**

* **特性**：Axios 是基于Promise的现代化的JavaScript HTTP客户端库，用于在浏览器和Node.js环境中发送HTTP请求。
* **优势**：提供简洁强大的API，支持请求和响应的拦截、取消、并发管理等功能。
* **详细介绍**：Axios 可以处理各种HTTP请求和响应，支持自动转换响应数据和错误处理。它能够设置请求头、处理文件上传和下载，并与常见的身份验证机制和RESTful API进行交互，是开发者首选的HTTP客户端库之一。

#### **2.2.1.3 Element Plus**

* **特性**：Element Plus 是基于Vue 3的组件库，提供了一套丰富的UI组件和设计语言。
* **优势**：支持快速构建美观的用户界面，具备灵活的布局和丰富的交互功能。
* **详细介绍**：Element Plus 继承了Element UI的优秀特性，但在Vue 3的基础上进行了优化和升级，包括性能提升、组件的可访问性和开发体验的改进，为开发者提供了更好的用户界面解决方案。

#### **2.2.1.4 ECharts**

* **特性**：ECharts 是一个基于JavaScript的开源可视化库，支持多种图表类型和数据展示。
* **优势**：提供了丰富的交互方式和灵活的配置选项，适用于复杂的数据可视化需求。
* **详细介绍**：ECharts 可以实现动态数据更新和复杂图表的展示，支持实时数据的呈现和用户交互，是开发者在数据展示方面的强大工具。

### **2.2.2 后端使用技术**

#### **2.2.2.1 Spring Boot 框架**

* **特性**：Spring Boot 是一款简化Spring应用开发的框架，提供了快速配置和快速部署的特性。
* **优势**：通过自动配置和约定大于配置的原则，减少了开发者的工作量，提高了开发效率。
* **详细介绍**：Spring Boot 可以集成各种Spring生态系统的组件，如Spring MVC、Spring Data等，同时支持内嵌服务器和微服务架构，适用于构建高效、稳定的后端服务。

#### **2.2.2.2** **Spring MVC**

* **特性**：Spring MVC 是Spring框架的Web模块，用于实现基于MVC模式的Web应用程序。
* **优势**：通过注解驱动的方式实现请求的处理和响应，支持RESTful风格的API开发。
* **详细介绍**：Spring MVC 提供了清晰的分层架构，将业务逻辑、数据模型和用户界面有效地分离，是开发Web应用的强大框架之一。

#### **2.2.2.3 MyBatis-Plus**

* **特性**：MyBatis-Plus 是 MyBatis 的增强工具包，简化了数据访问层的开发。
* **优势**：提供了CRUD操作的便捷方法、代码生成器和分页插件等功能，加速了开发周期。
* **详细介绍**：MyBatis-Plus 集成了MyBatis的优点，并在此基础上进一步封装，提高了SQL查询的灵活性和性能，是开发持久层的首选工具之一。

#### **2.2.2.4** **Druid**

* **特性**：Druid 是一个高性能的开源数据库连接池，支持监控和扩展。
* **优势**：提供了稳定可靠的数据库连接管理，有效地管理和优化数据库连接，提升了系统的性能。
* **详细介绍**：Druid 不仅支持连接池功能，还提供了SQL执行监控、防火墙等功能，是处理数据库连接的理想选择。

#### **2.2.2.5** **JWT（JSON Web Token）**

* **特性**：JWT 是一种基于JSON的开放标准（RFC 7519），用于在各方之间安全地传输信息。
* **优势**：轻量、紧凑，能够安全地传输数据和进行身份验证，适用于分布式环境下的认证授权。
* **详细介绍**：JWT 可以生成和验证Token，支持无状态的认证机制，避免了服务端存储Session和Token信息的开销，是现代应用中常用的身份验证解决方案之一。

#### **2.2.2.6** **Jackson**

* **特性**：Jackson 是一个用于处理JSON数据的Java库，提供了高效的数据绑定和转换功能。
* **优势**：支持将Java对象序列化为JSON格式数据，以及将JSON数据反序列化为Java对象，便于前后端数据的传输和处理。
* **详细介绍**：Jackson 提供了丰富的API和配置选项，可以处理复杂对象和嵌套结构，是Java开发中处理JSON数据的首选库之一。

### **2.2.3 数据库**

#### **2.2.3.1 Mysql**

 **特性**：MySQL 是一种关系型数据库管理系统，广泛应用于Web开发和数据存储领域。

 **优势**：支持多用户、多线程和多表查询，提供了高性能和稳定性的数据存储解决方案。

 **详细介绍**：MySQL 提供了完善的SQL语法和事务支持，能够处理大规模数据和复杂查询，是许多企业和开发者选择的数据库系统之一。

## **2.3 本章小结**

本次项目采用前后端分离的架构，前后端均采用市面上主流的开发工具以及开发技术来进行开发。通过对多种开发技术的综合应用，来简化前后端的开发。

# **第3章 软件开发工作**

## **3.1 需求分析**

### **3.1.1概述**

智能制造云平台用于管理云工厂在实际生产中的各项工作流程，包括用户下单、订单跟踪修改、平台接单转为生产计划、工厂启动生产计划并拆分为工单、以及对工单的报工与生产进度管理。本系统包括：用户订单管理、平台生产计划管理、工厂工单管理三个功能模块。另包括权限管理模块用于系统的用户、角色和相关权限管理。该平台将作为连接客户与工厂、实现订单与生产无缝对接的桥梁，通过实时监控和数据分析，为企业提供决策支持，推动企业实现数字化转型和智能化升级。

### **3.1.2角色与权限**

与本系统相关的角色包括：客户、工厂、平台管理员

**客户**

* 负责在平台上下单、跟踪订单以及修改订单等功能。
* 可以查看订单状态和详情。

**工厂**

* 负责接收和管理生产计划，执行生产任务。
* 可以查看并启动生产计划，将生产计划拆分为工单，并进行报工操作。
* 拥有生产计划管理、工单管理和生产进度更新的权限。

**平台管理员**

* 负责平台整体的订单管理和转发生产计划给工厂。
* 可以查看、接单并将订单转为生产计划。
* 拥有订单管理和生产计划转发的权限。

## **3.2 用例图**

### **3.2.1总用例图**

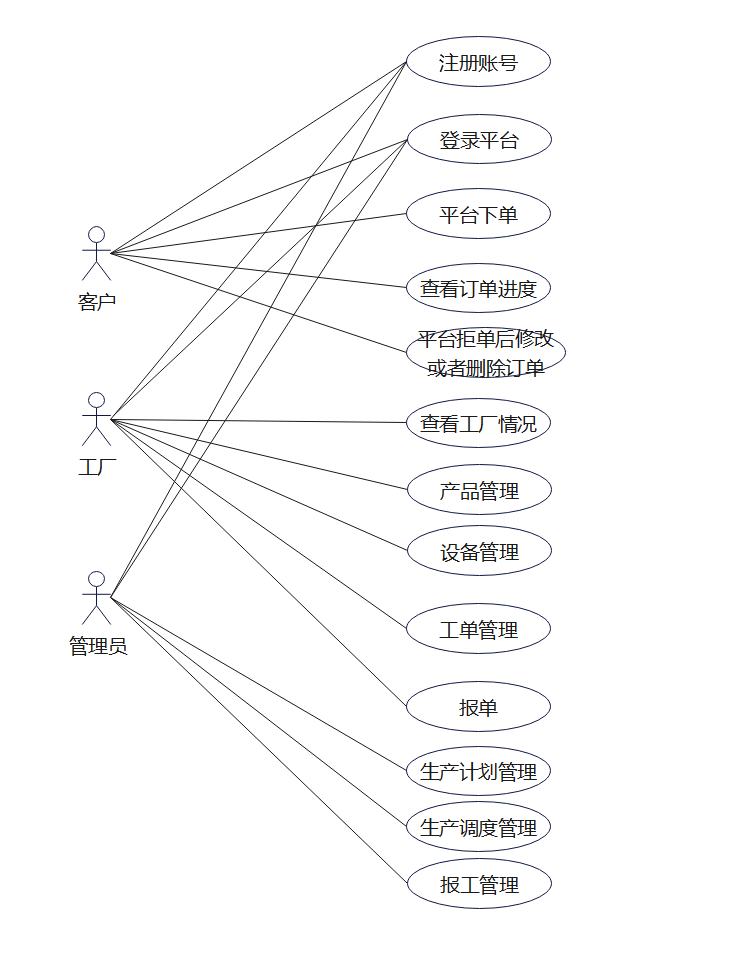


图3-3-1 总用例图

总用例图用例描述，如表3-3-1所示。

表3-3-1 总用例图用例描述

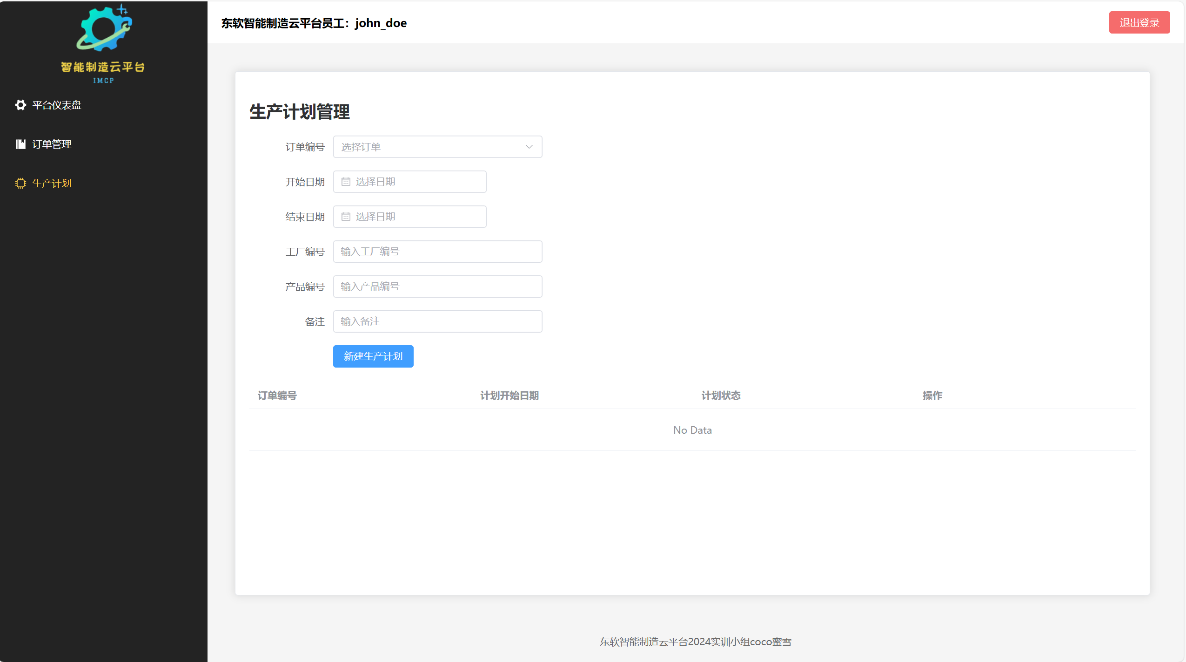
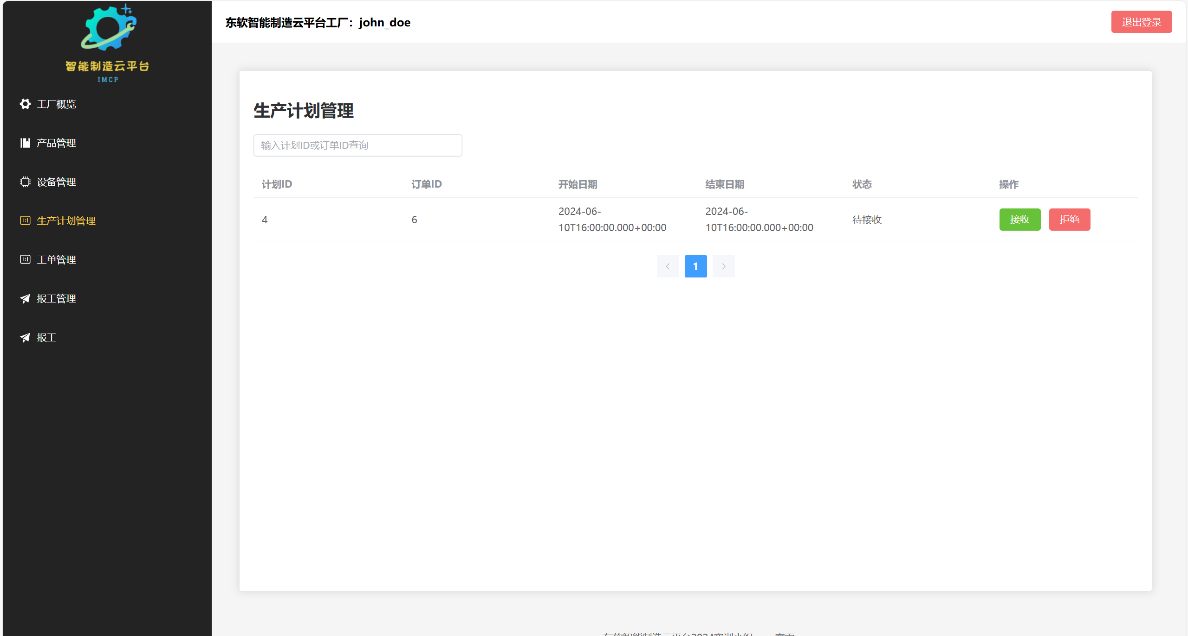
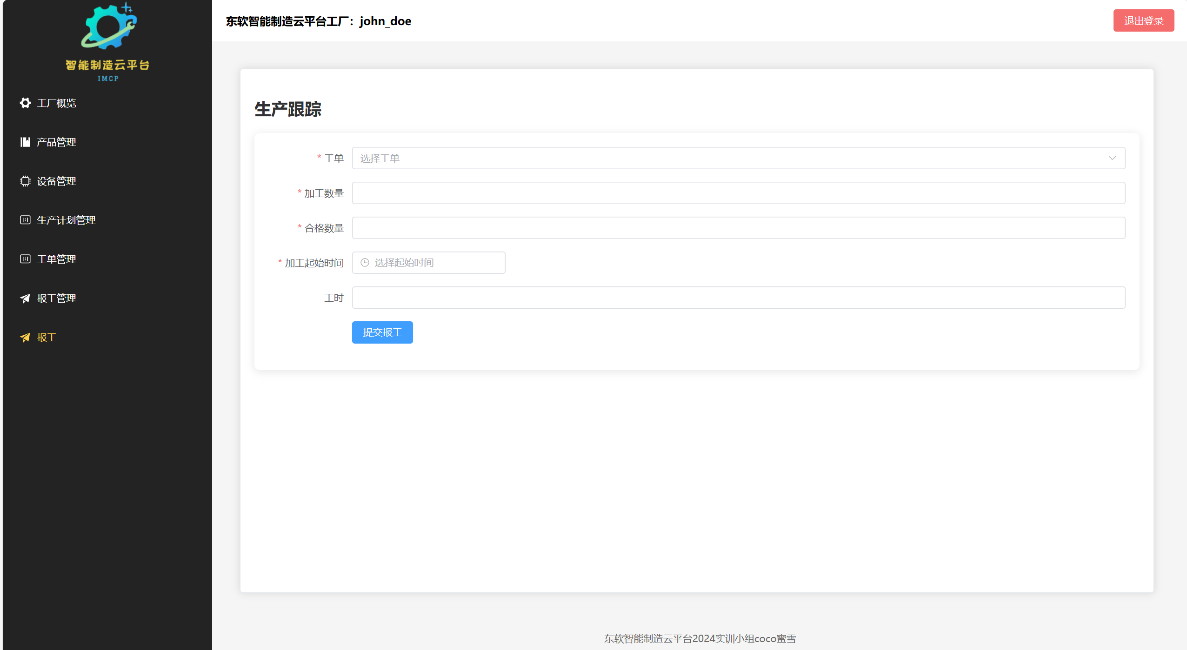
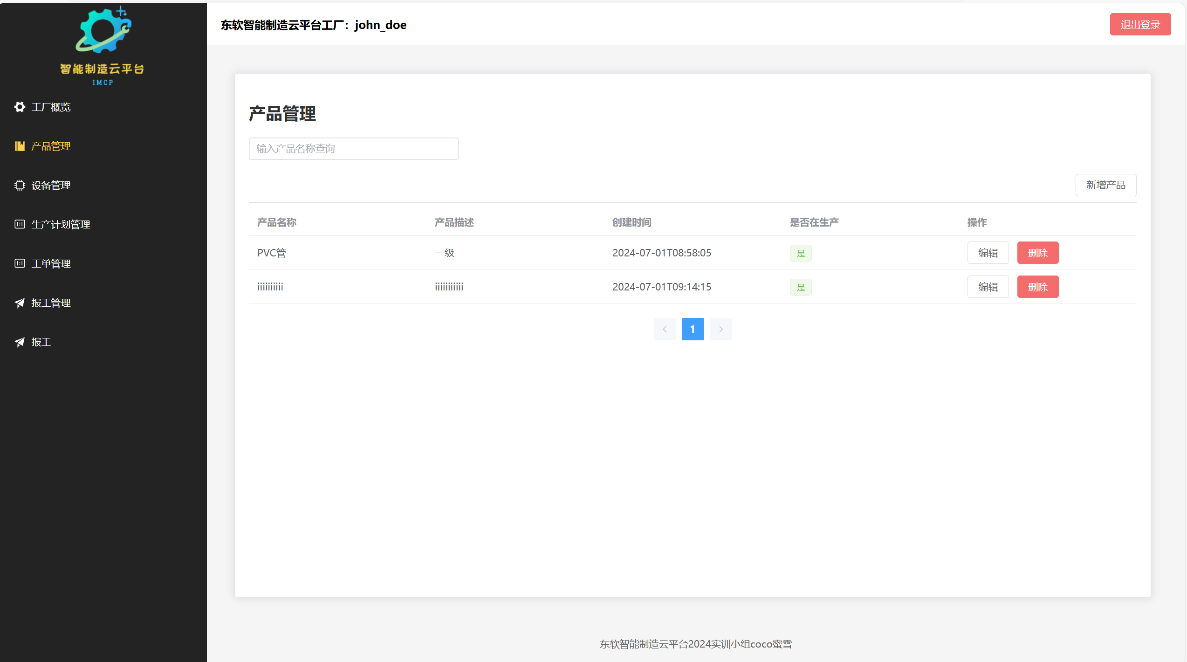
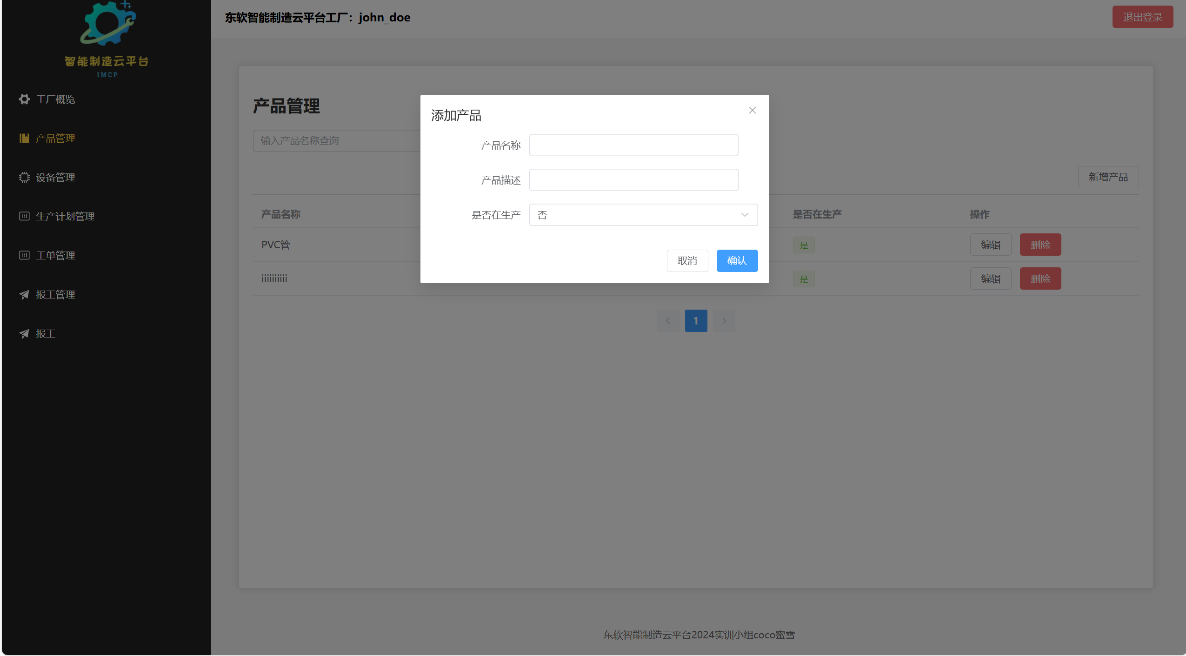
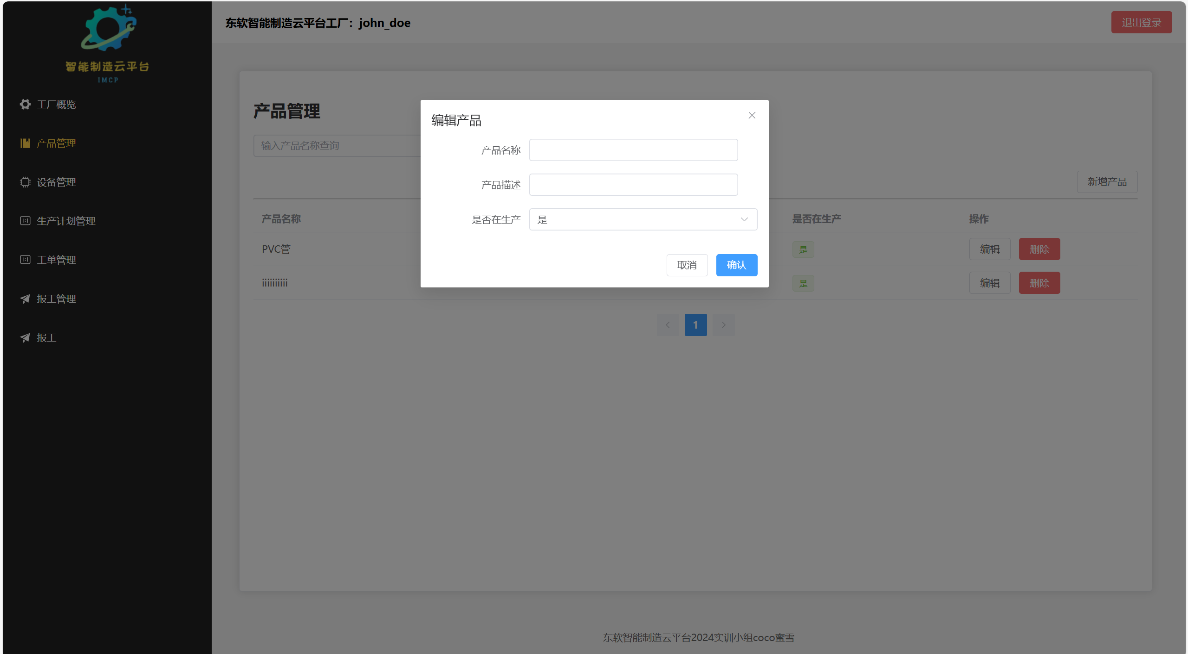
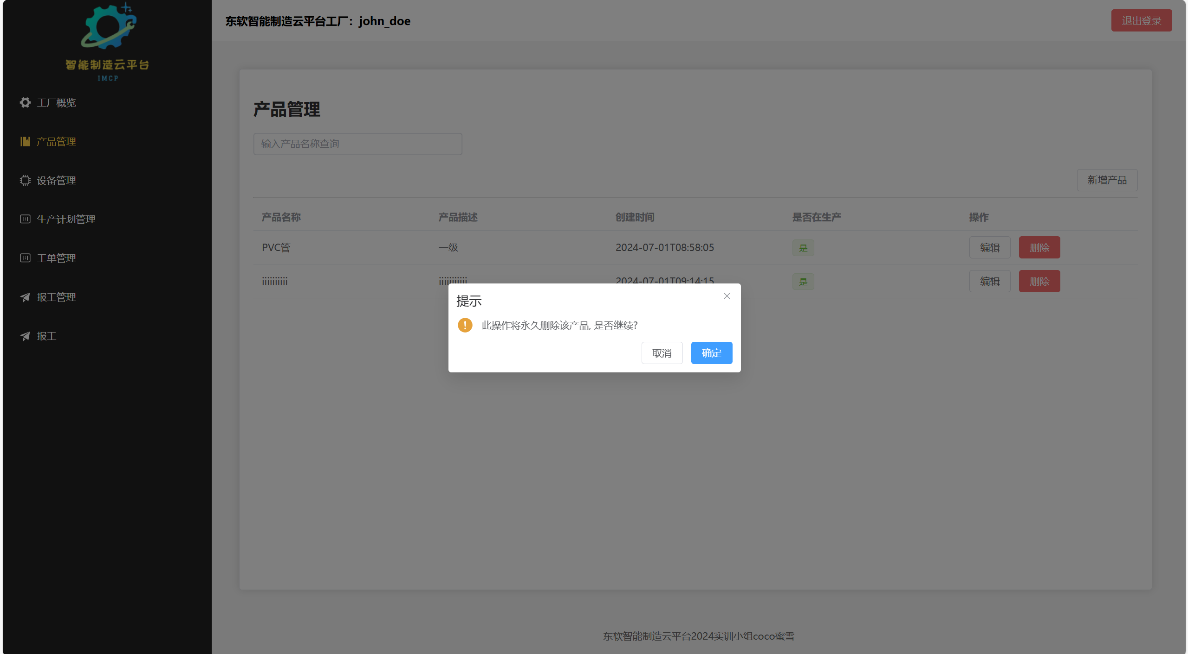
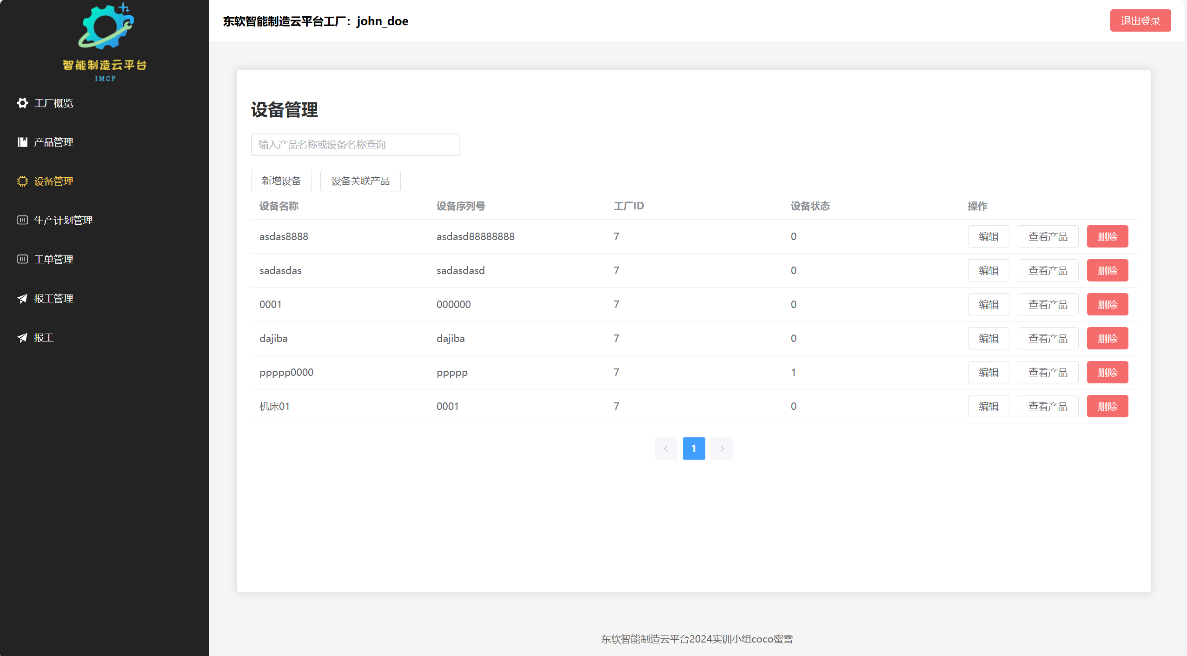
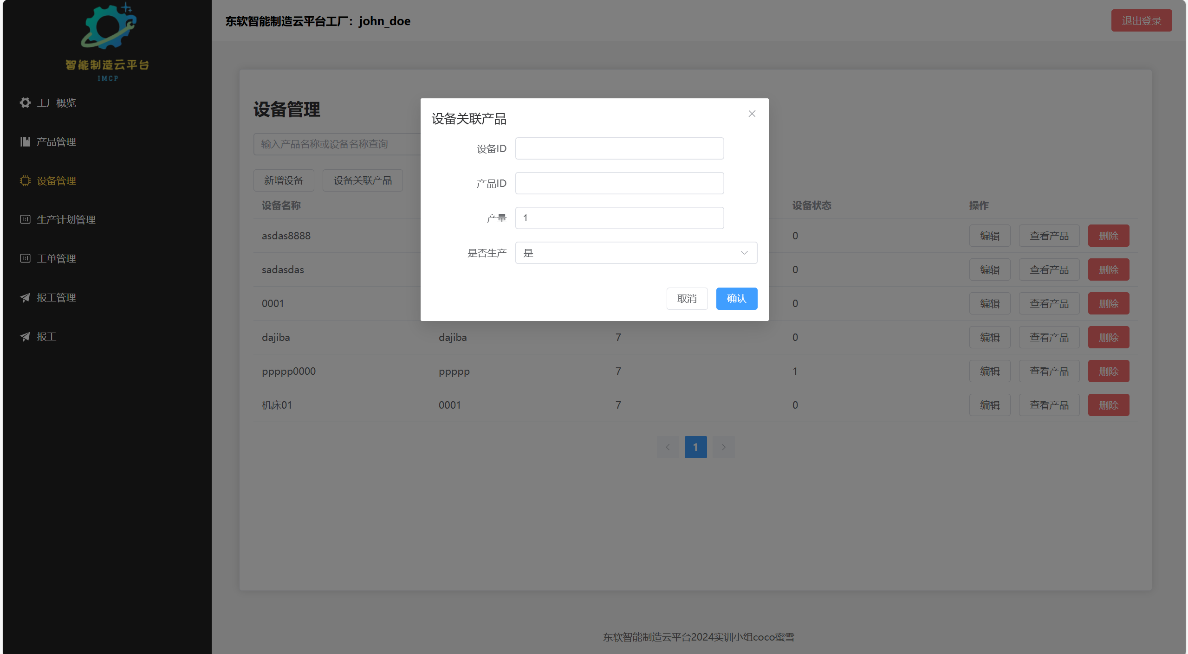
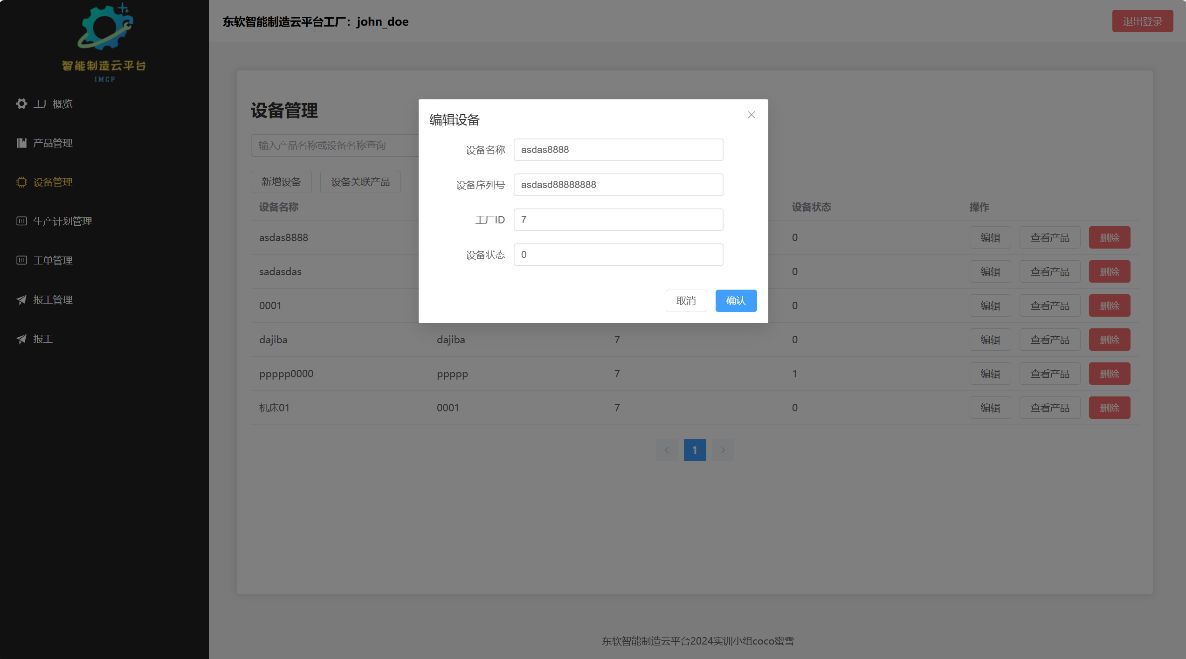
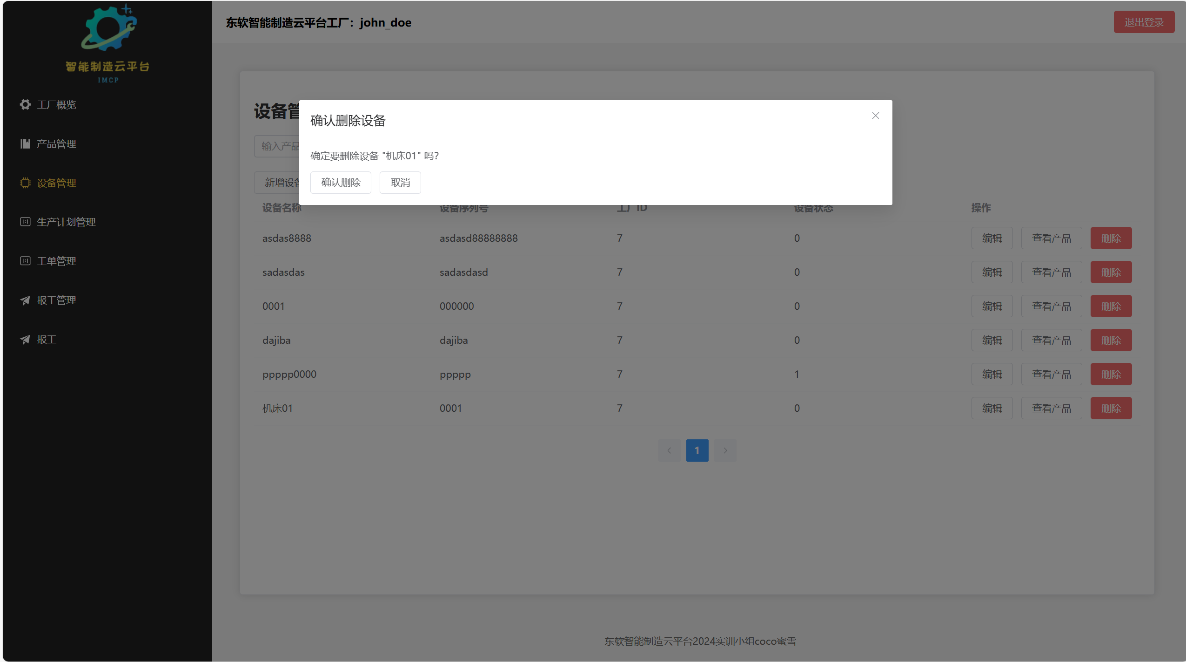
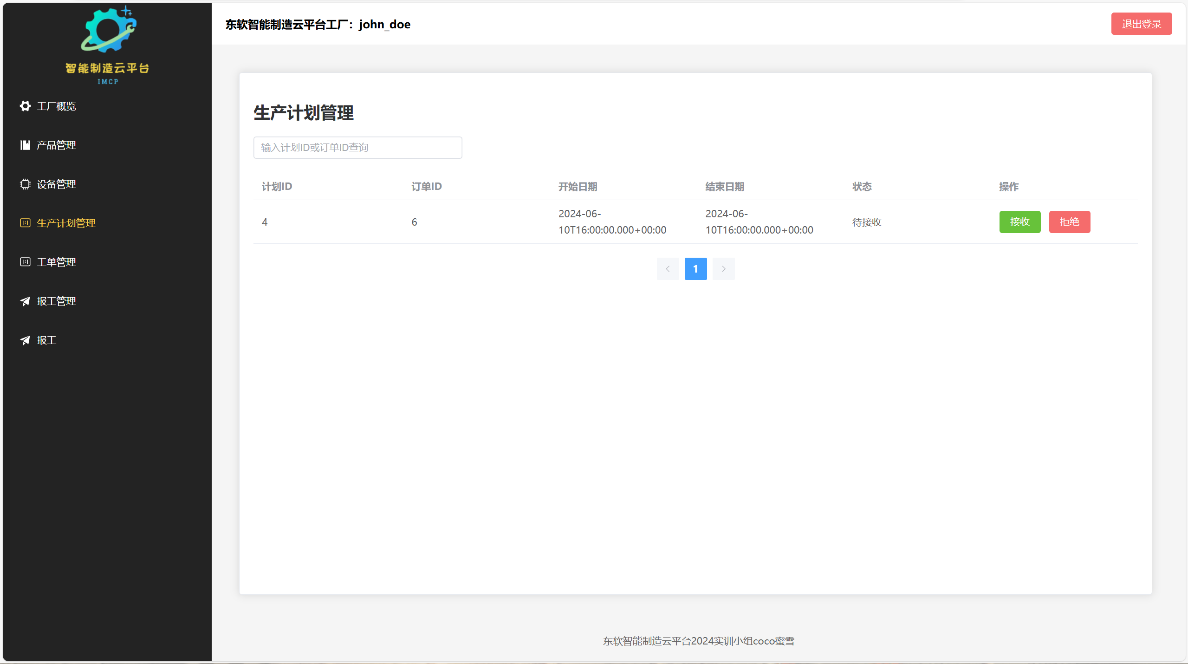
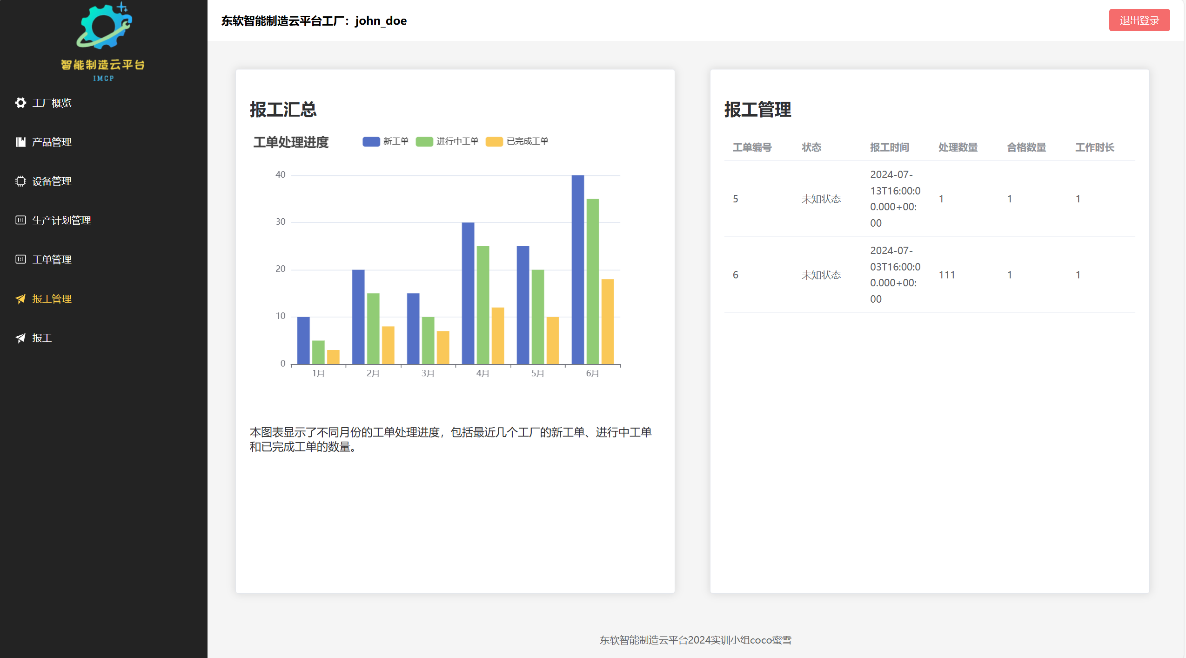
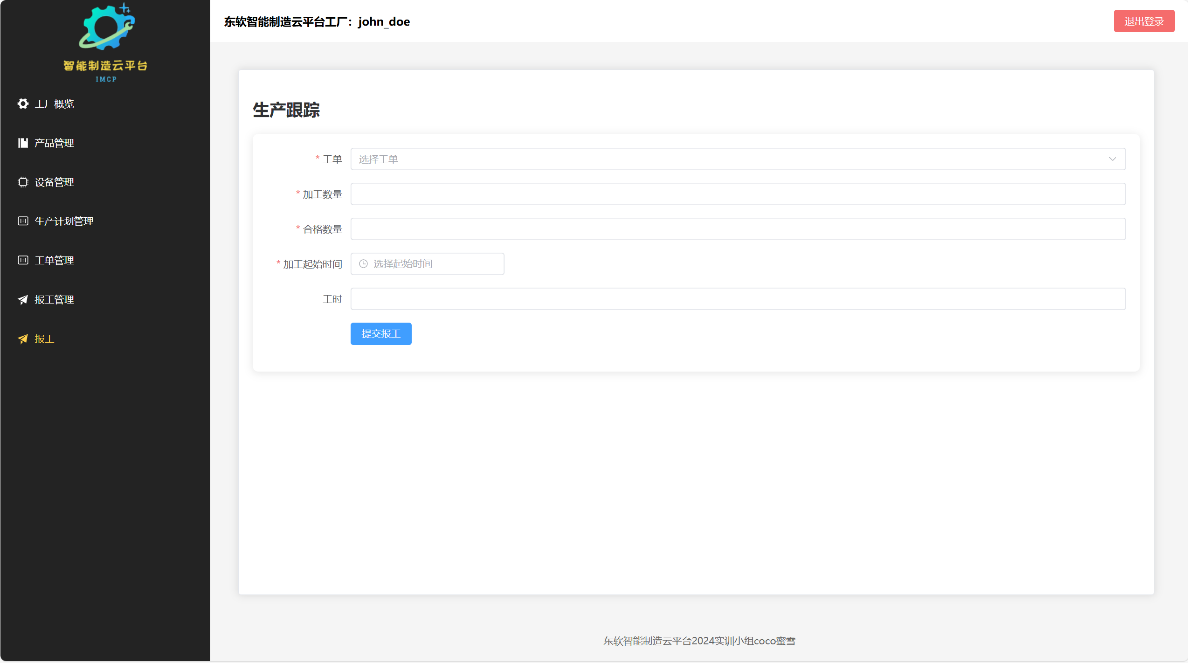
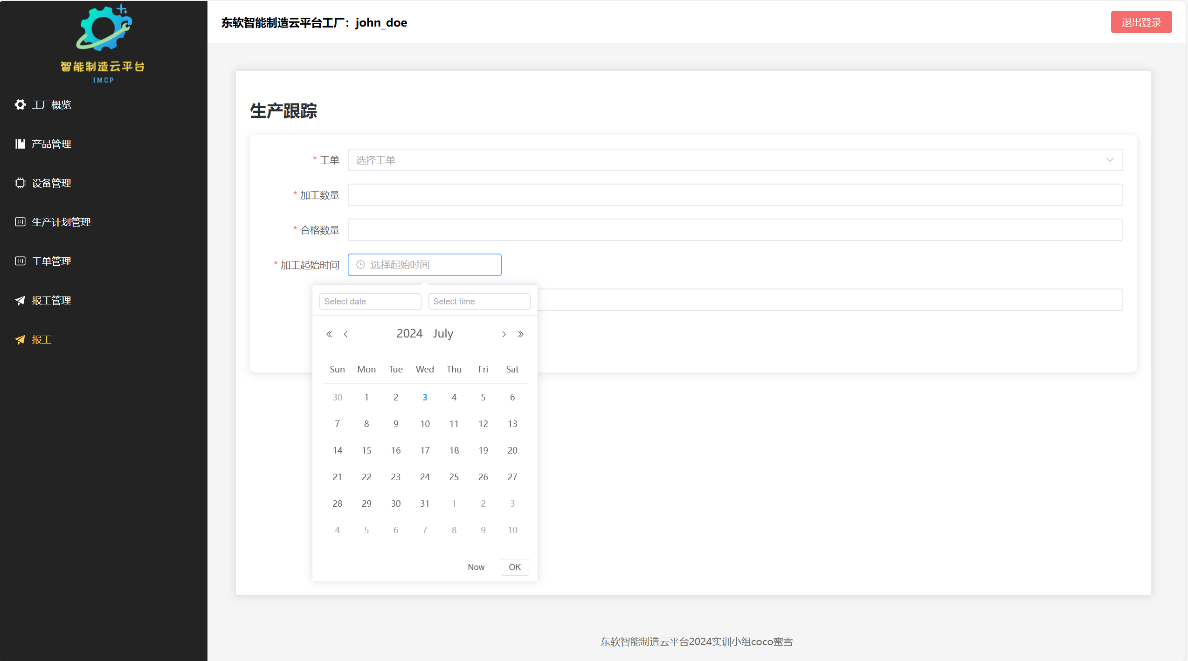
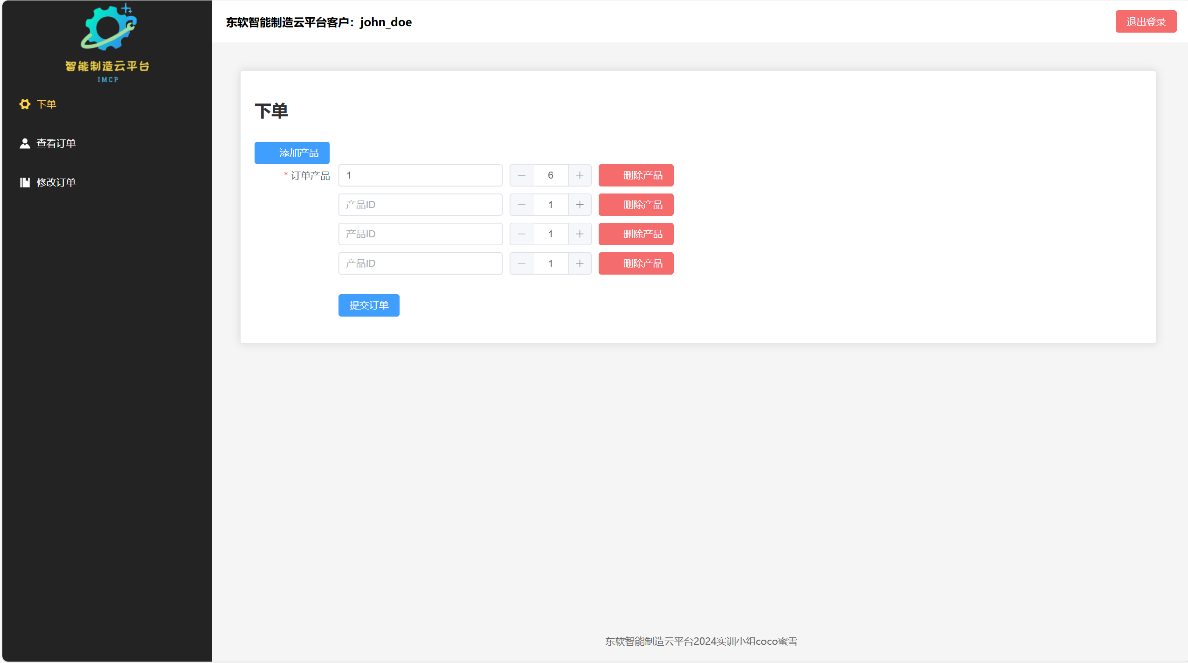
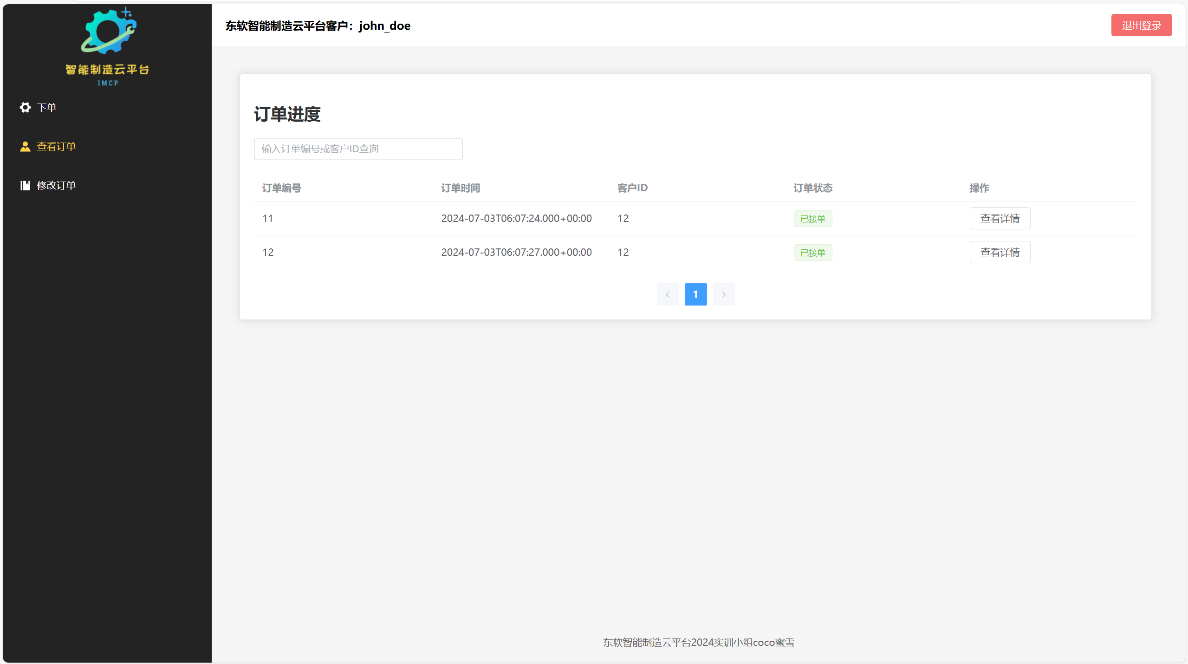
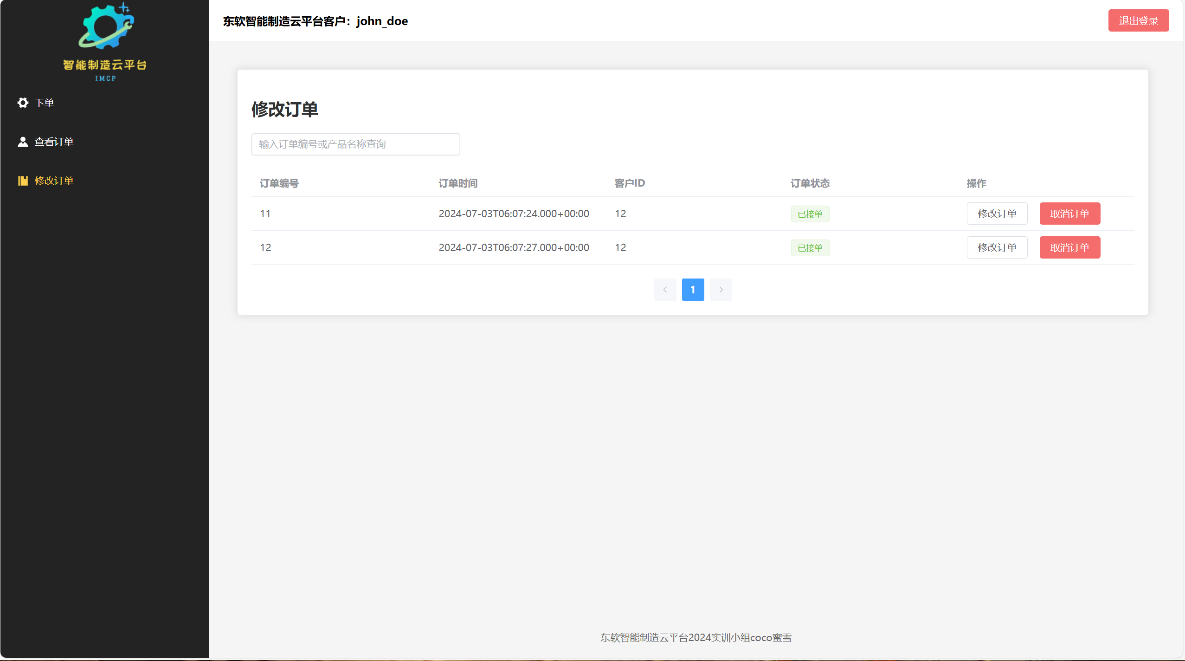
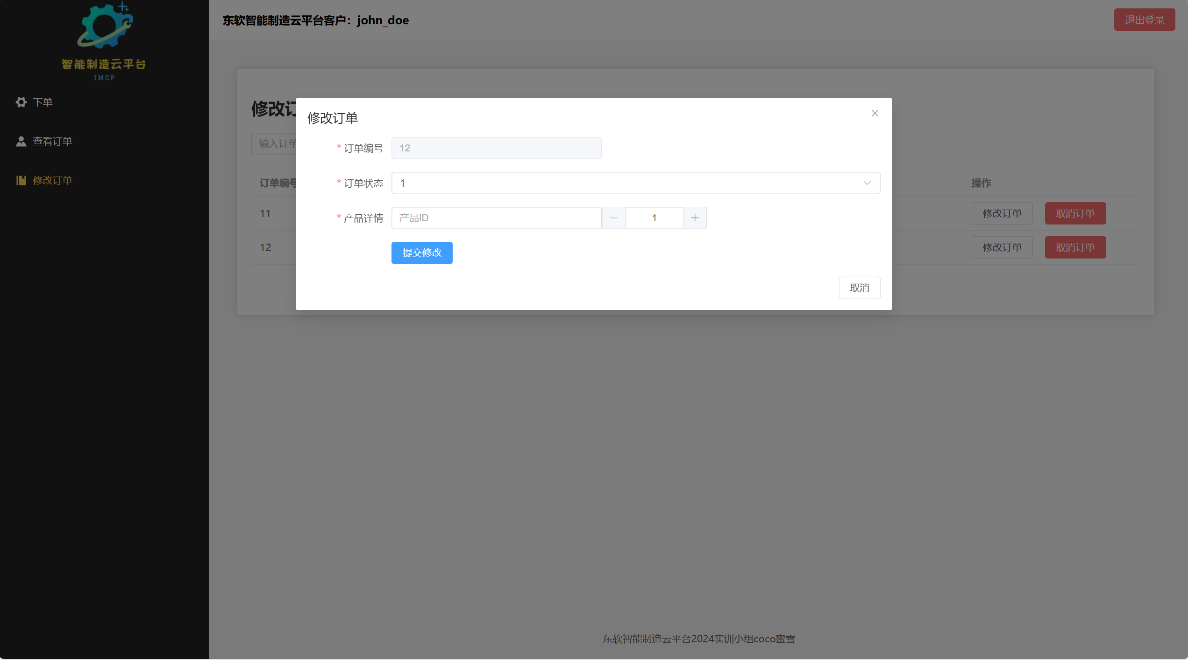
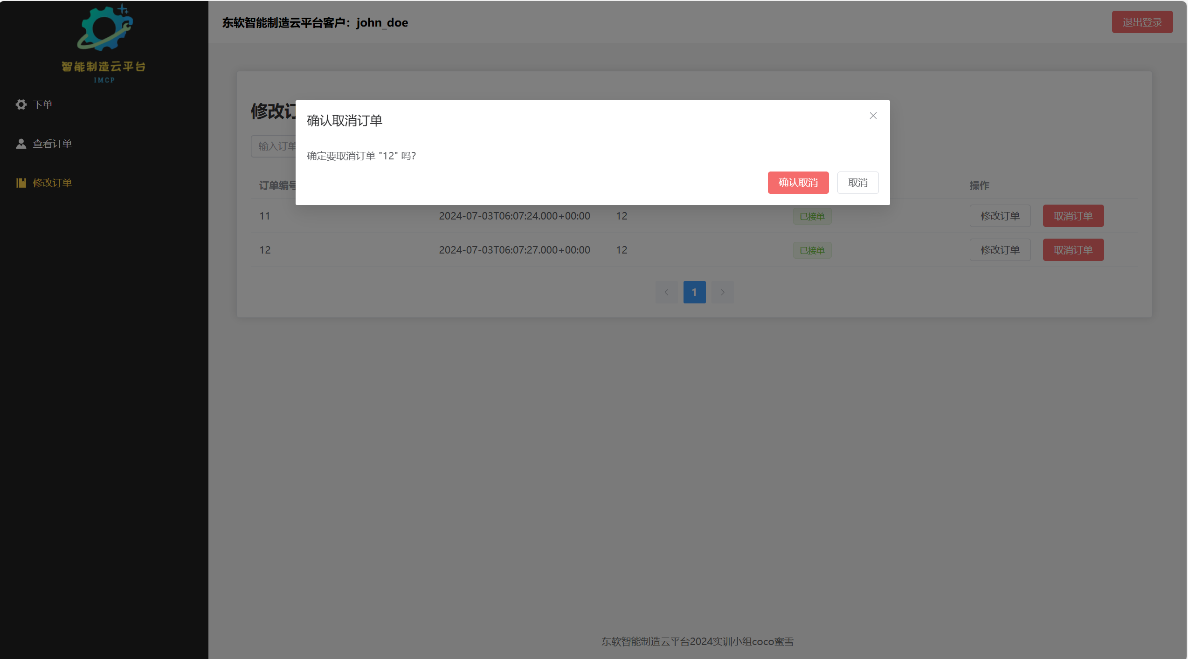
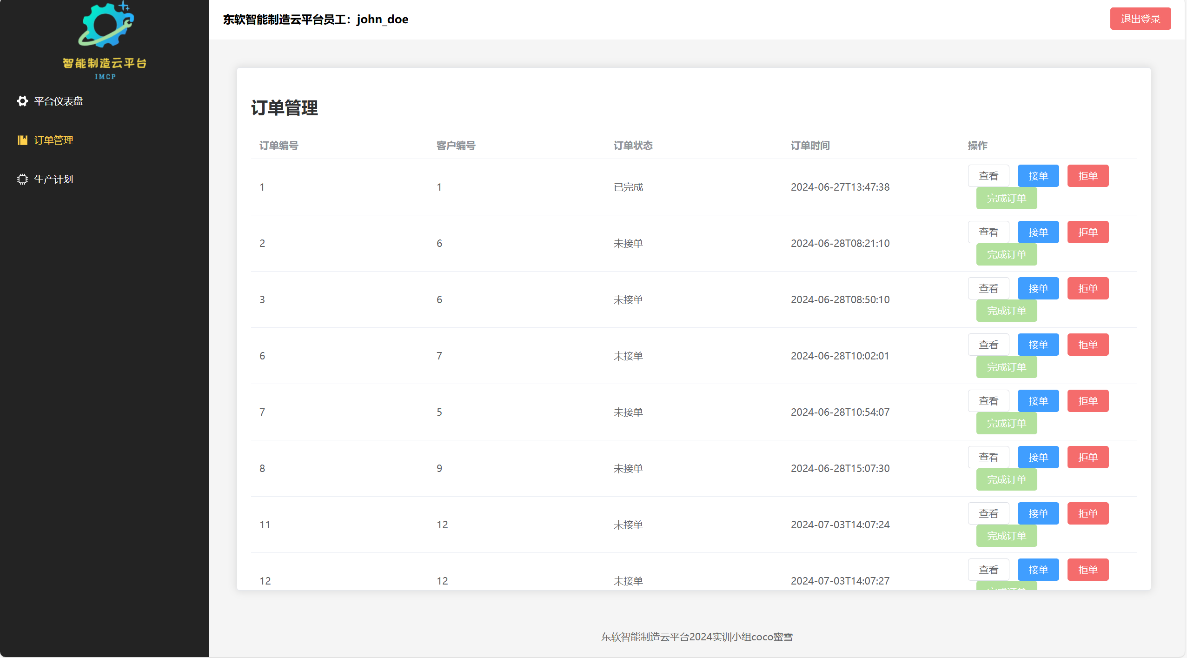
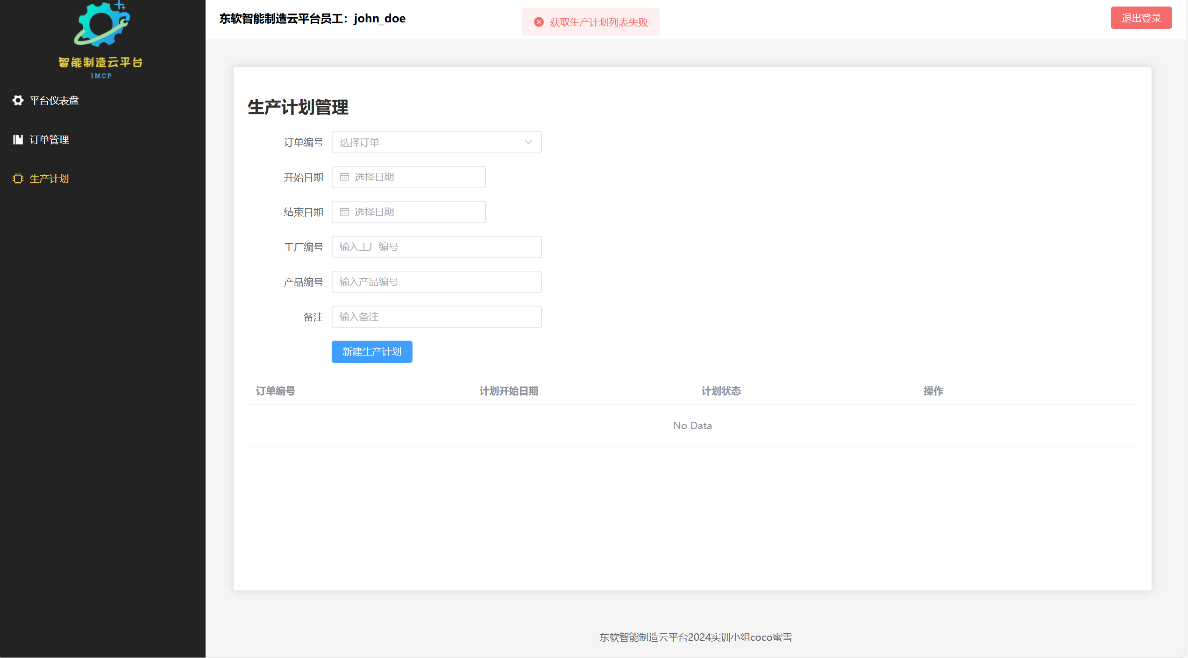
|  |  |
| --- | --- |
| 用例名称 | 东软智能制造云平台 |
| 参与者 | 客户、工厂、管理员 |
| 简要描述 | 该用例描述了不同角色进入此东软智能制造云平台所具有的功能。 |
| 前置条件 | 各角色系统进入东软智能制造云平台，并使用。 |
| 后置条件 | 各角色利用此系统达到了预期目标。 |
| 主要场景 | 客户：  ①注册账号  ②平台下单  ③查看订单情况  ④修改订单  ⑤删除订单  工厂：  ①注册账号  ②接受生产计划  ③将计划划分为订单  ④工厂报告工作进度  ⑤完成订单  平台管理员：  ①注册账号  ②进行产品管理  ③进行设备管理  ④接受报单 |
| 异常场景 | ①各用户无法使用各自的模块功能。  ②各用户使用了其他角色的功能。 |
| 备注 | 无 |

# **第4章 原型设计**

## **4.1业务流程图**



## **4.2页面原型图**



## **4.3需求规约**

## **4.3.1 系统概述**

**系统名称**：东软智能制造云平台

**系统目标**：为客户、工厂、管理员提供一站式智能制造解决方案，实现订单管理、生产计划、设备监控等功能的高效集成。

**参与者**：客户、工厂、管理员

## **4.3.2 功能需求**

##### **4.3.2.1 客户功能需求**

1. 注册账号：客户能够通过平台提供的注册页面完成账号注册.
2. 平台下单：客户能够提交订单信息，包括产品、数量等。
3. 查看订单情况：客户能够查看自己的订单列表，包括订单状态、订单详情等信息。
4. 修改订单：在订单未接受前，客户能够修改订单信息，如数量、收货地址等。
5. 删除订单：在订单未接受前，客户能够删除已提交的订单。

##### **4.3.2.2 工厂功能需求**

1. 注册账号：工厂能够通过平台提供的注册页面完成账号注册。
2. 接受生产计划：工厂能够接收平台分配的生产计划，并查看计划详情。
3. 将计划划分为订单：工厂能够根据生产计划，将任务细化为具体的订单，并分配资源。
4. 工厂报告工作进度：工厂能够定期或按需更新订单的工作进度，并提交给平台。
5. 完成订单：工厂在订单生产完成后，能够在平台上标记订单为已完成，并通知客户和管理员。

##### **4.3.2.3 管理员功能需求**

1. 注册账号：管理员通过特殊流程完成账号注册，并进行权限审核。
2. 进行产品管理：管理员能够管理产品目录，包括添加、编辑、删除产品信息，设置价格、库存等属性。
3. 进行设备管理：管理员能够管理工厂上报的设备信息，监控设备状态，处理设备故障报修。
4. 接受报单：管理员能够接收来自工厂或客户的各类报单，分析报单内容，并协调相关部门处理

## **4.3.3 非功能性需求**

##### **4.3.3.1 性能需求**

* 系统响应时间：关键操作（如订单提交、生产计划接收）响应时间不超过2秒。
* 系统并发能力：支持至少1000个并发用户同时在线操作。
* 数据处理能力：能够高效处理大规模数据，保证数据实时性和准确性。

##### **4.3.3.2 安全性需求**

* 用户认证与授权：采用多因素认证机制，确保用户身份安全；严格控制用户权限，防止越权操作。
* 数据加密：对敏感数据（如用户密码、交易信息等）进行加密存储和传输。
* 访问控制：实施细粒度的访问控制策略，确保数据资源的安全访问。

##### **4.3.3.3 可用性需求**

* 系统可用性：系统全年无故障运行时间不低于99.9%。
* 易用性：界面友好，操作直观，提供详细的使用说明和帮助文档。
* 兼容性：支持多种主流浏览器和操作系统平台

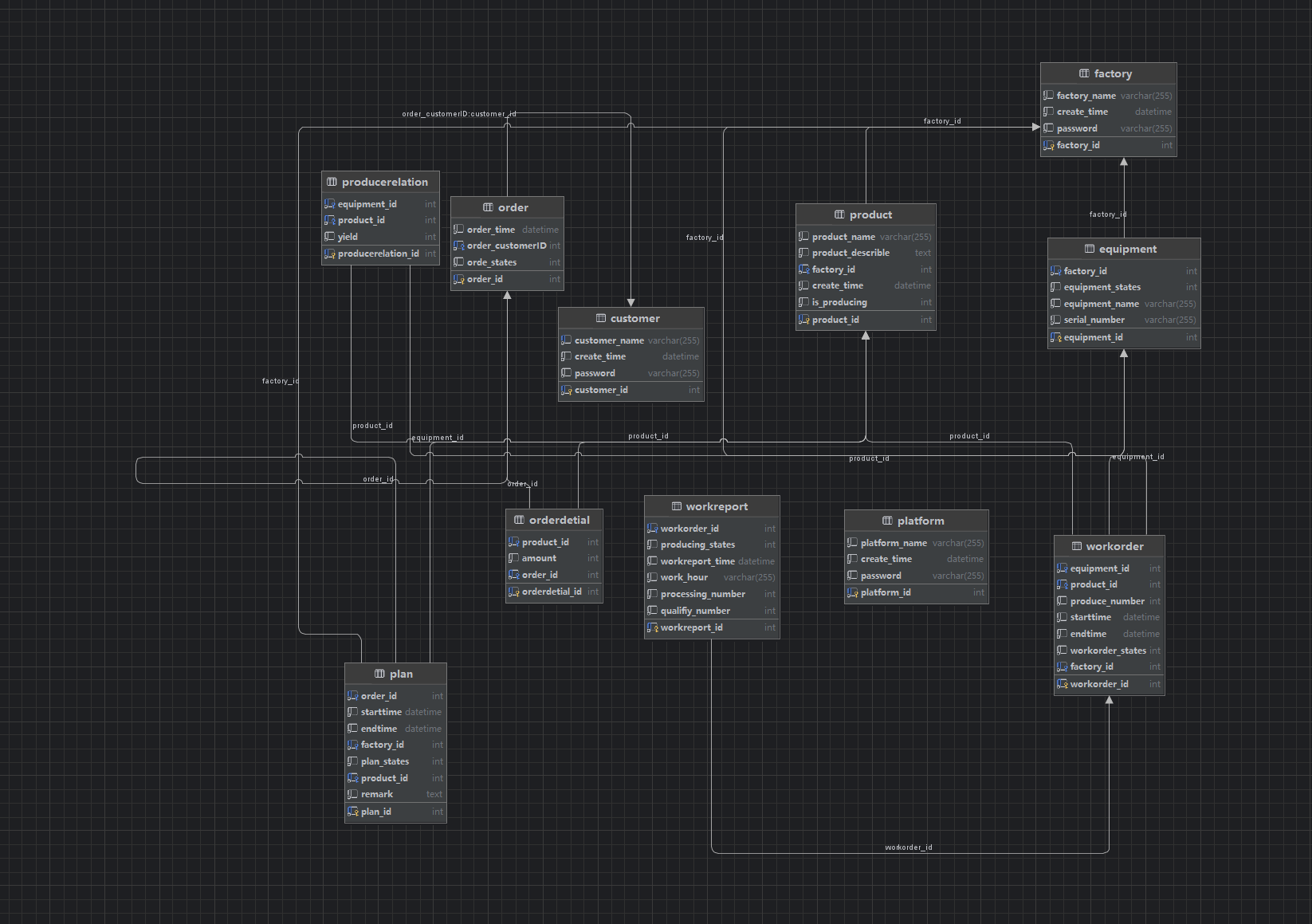
# **第5章 系统设计**

## **5.1数据库设计**

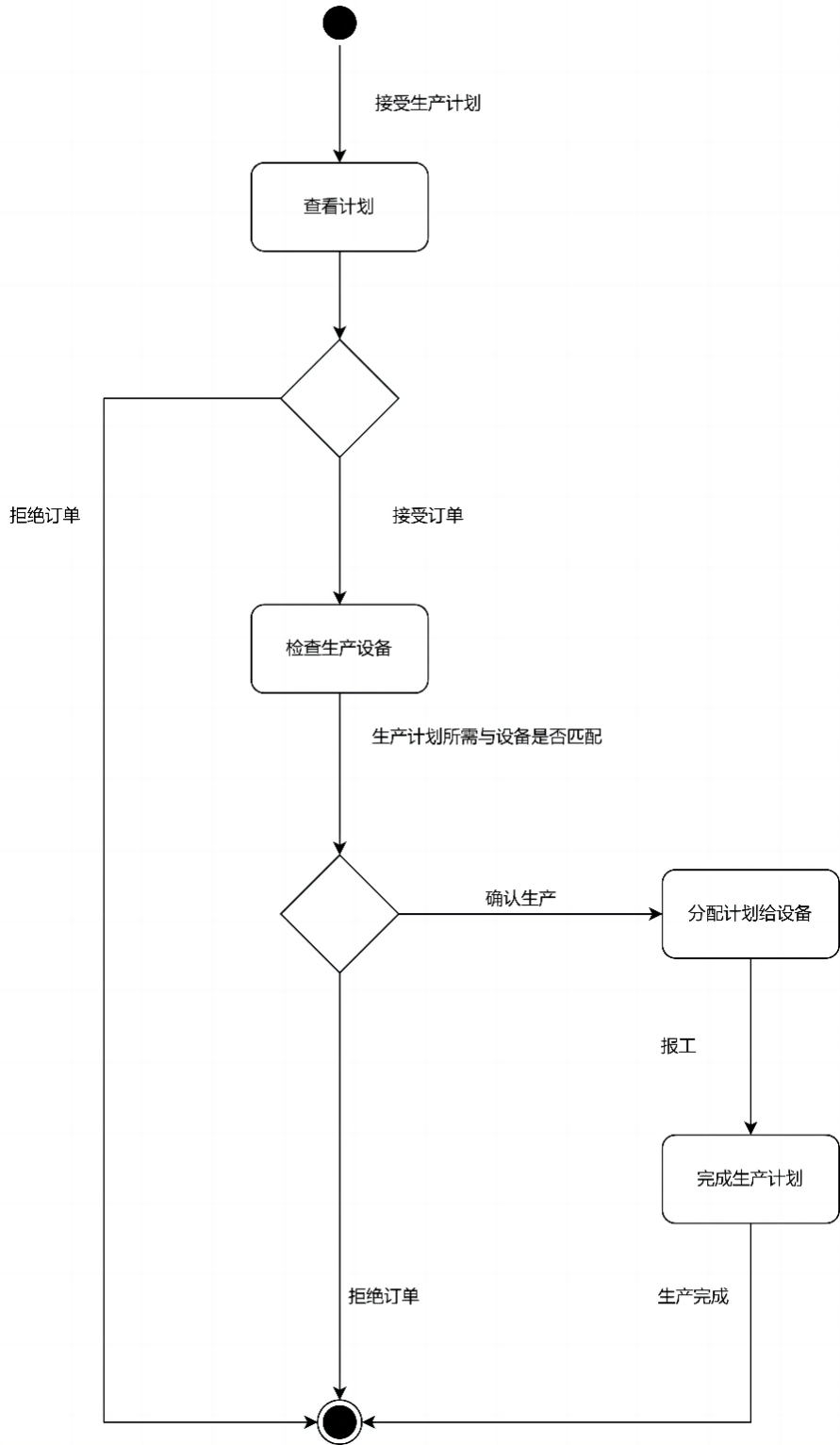
### **5.1.1ER图**



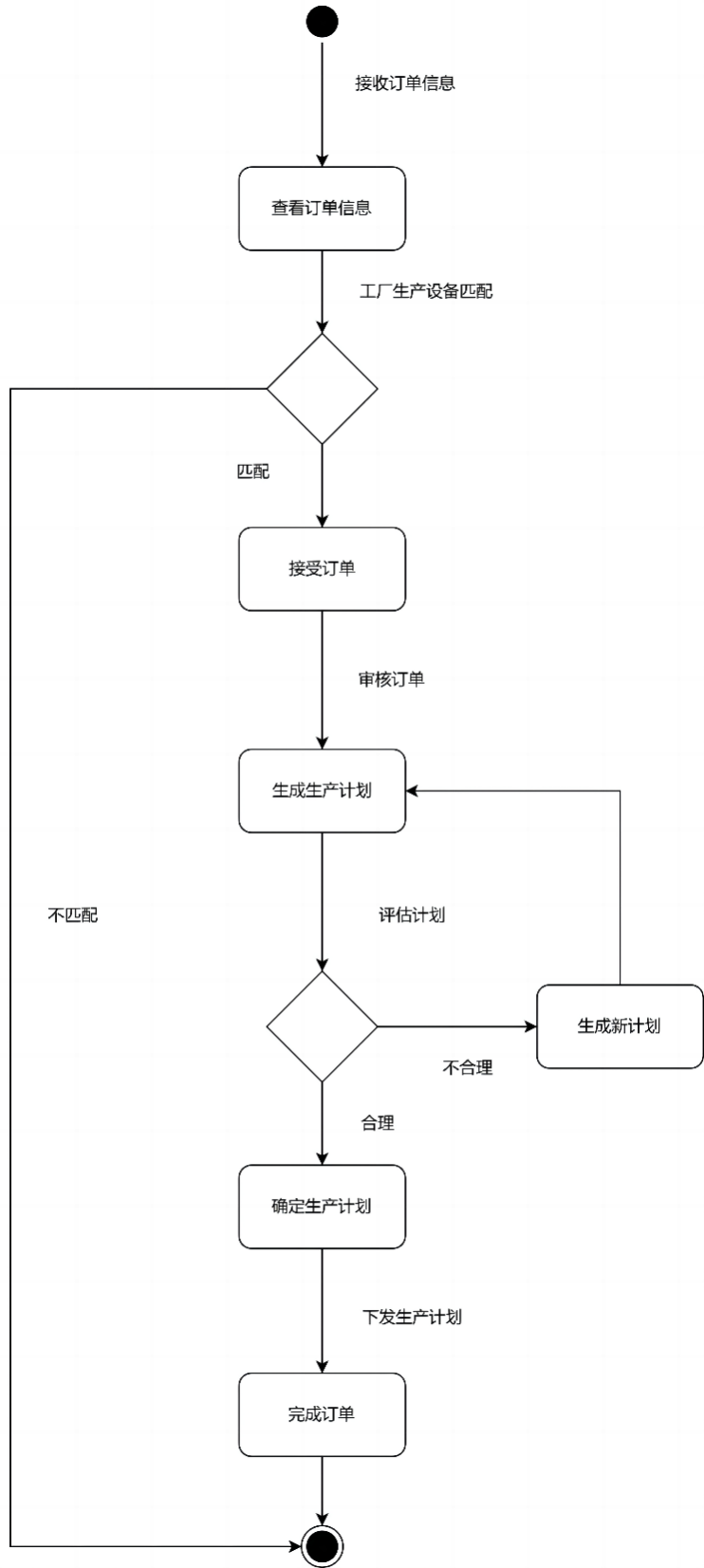
### **5.1.2表关系图**



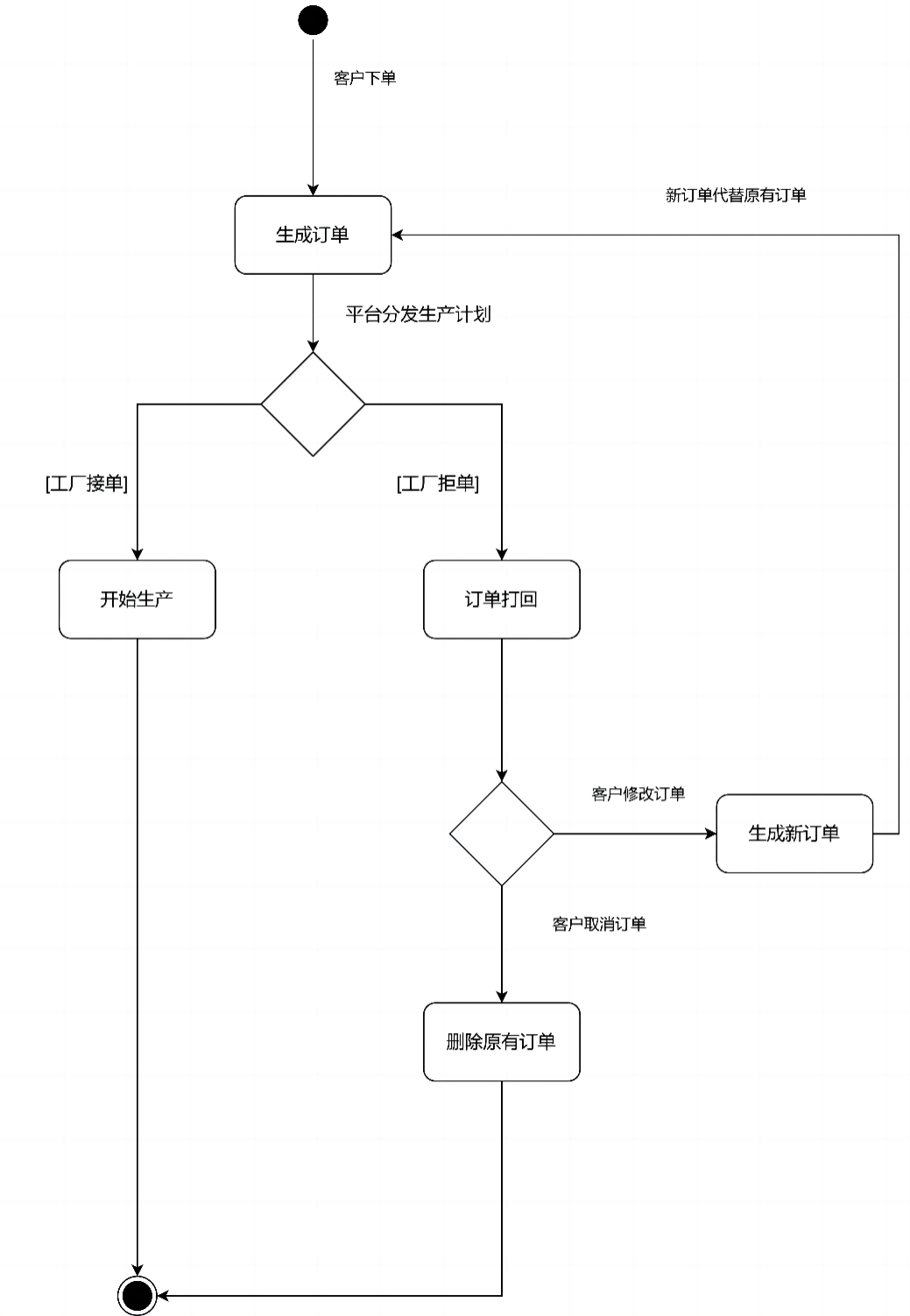
## **5.2状态图**



图一 工厂生产计划实体状态图

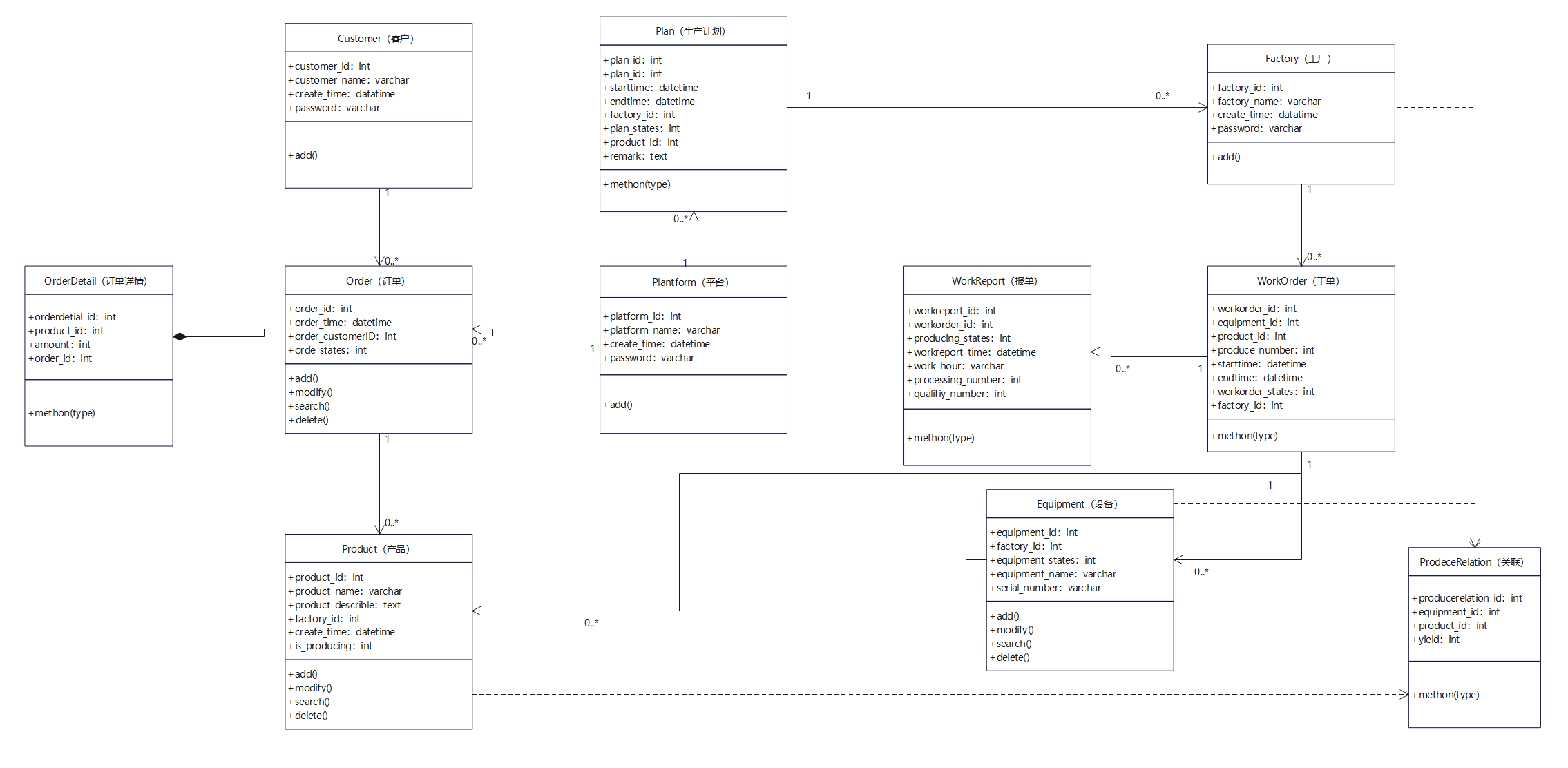


图二 平台创建生产计划实体状态图

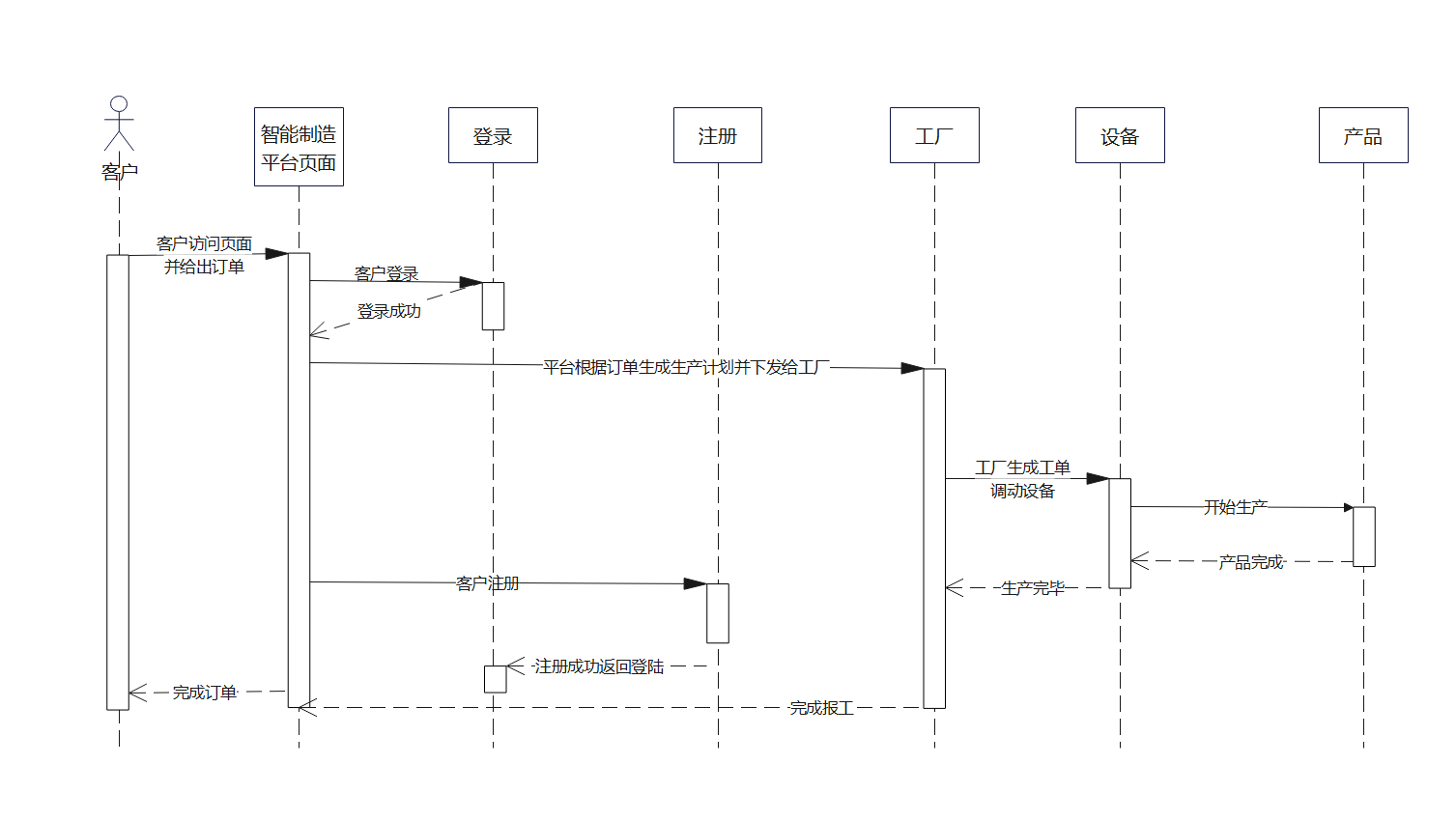


图三 客户下单实体状态图

## **5.3类图**



## **5.4 订单处理时序图**



# **第6章 重要用例：动态显示当前工厂的情况**

该功能旨在实时动态地展示当前登录工厂的设备运行状态、生产效益情况及订单统计信息。系统通过定时向后端服务发送请求，获取最新的工厂数据，包括但不限于设备运行效率、产量、合格率等生产效益指标，以及订单量、完成率等订单统计信息。获取的数据随后被用于ECharts图表绘制，以直观、易理解的方式展示给用户。这种方式不仅提升了信息获取的效率，还增强了数据监控的实时性和准确性，有助于工厂管理者快速响应生产变化，优化资源配置，提升整体运营效率。



**前端代码：**

|  |
| --- |
| <template> |
| <el-container class="dashboard-container"> |
| <el-header> |
| <h2>工厂概览</h2> |
| </el-header> |
| <el-main> |
| <el-row :gutter="20"> |
| <el-col :span="8"> |
| <el-card shadow="hover"> |
| <div slot="header" class="clearfix"> |
| <span>设备运行情况</span> |
| <el-button @click="refreshDeviceChart" type="primary" size="mini" style="float: right;">刷新</el-button> |
| </div> |
| <div ref="deviceChart" class="chart"></div> |
| </el-card> |
| </el-col> |
| <el-col :span="8"> |
| <el-card shadow="hover"> |
| <div slot="header" class="clearfix"> |
| <span>工厂生产效益情况</span> |
| <el-button @click="switchEfficiencyChartType" type="primary" size="mini" style="float: right;">切换图表</el-button> |
| </div> |
| <div ref="efficiencyChart" class="chart"></div> |
| </el-card> |
| </el-col> |
| <el-col :span="8"> |
| <el-card shadow="hover"> |
| <div slot="header" class="clearfix"> |
| <span>订单统计情况</span> |
| </div> |
| <div ref="orderChart" class="chart"></div> |
| </el-card> |
| </el-col> |
| </el-row> |
| <el-row :gutter="20" style="margin-top: 20px;"> |
| <el-col :span="24"> |
| <el-card shadow="hover"> |
| <div slot="header" class="clearfix"> |
| <span>工厂概览信息</span> |
| </div> |
| <div class="overview-text"> |
| <p>工厂概览展示了当前设备的运行状态、生产效益和订单统计情况。</p> |
| <p>通过这些信息，您可以更好地了解工厂的整体运作情况，及时进行调整和优化。</p> |
| <p>定期刷新设备运行情况，切换生产效益图表类型以查看不同的展示效果。</p> |
| </div> |
| </el-card> |
| </el-col> |
| </el-row> |
| </el-main> |
| </el-container> |
| </template> |
|  |
| <script setup> |
| import { ref, onMounted } from 'vue'; |
| import \* as echarts from 'echarts'; |
| import { getFactoryOverviewData } from '@/api/factory/factoryview.js'; |
|  |
| const deviceChart = ref(null); |
| const efficiencyChart = ref(null); |
| const orderChart = ref(null); |
| let efficiencyChartType = ref('line'); |
|  |
| const orderStatusMap = { |
| 1: '未接单', |
| 2: '已接单', |
| 3: '生产中', |
| 4: '已完成' |
| }; |
|  |
| onMounted(() => { |
| initCharts(); |
| setInterval(refreshDeviceChart, 300000); // 每分钟刷新一次设备运行情况图表 |
| }); |
|  |
| const initCharts = async () => { |
| // 获取工厂概览数据 |
| const factoryOverviewData = await getFactoryOverviewData().then(res => res.data); |
|  |
| // 初始化设备运行情况图表 |
| const deviceChartInstance = echarts.init(deviceChart.value); |
| const deviceChartOption = { |
| title: { text: '设备运行情况' }, |
| tooltip: {}, |
| legend: { data: ['设备状态'] }, |
| xAxis: { data: factoryOverviewData.equipmentStatus.map((\_, index) => `设备${index + 1}`) }, |
| yAxis: {}, |
| series: [{ |
| name: '设备状态', |
| type: 'bar', |
| data: factoryOverviewData.equipmentStatus |
| }] |
| }; |
| deviceChartInstance.setOption(deviceChartOption); |
|  |
| // 初始化工厂生产效益图表 |
| const efficiencyChartInstance = echarts.init(efficiencyChart.value); |
| const efficiencyChartOption = { |
| title: { text: '工厂生产效益情况' }, |
| tooltip: {}, |
| legend: { data: ['生产效益'] }, |
| xAxis: { data: factoryOverviewData.productionEfficiency.map((\_, index) => `${index + 1}月`) }, |
| yAxis: {}, |
| series: [{ |
| name: '生产效益', |
| type: efficiencyChartType.value, |
| data: factoryOverviewData.productionEfficiency |
| }] |
| }; |
| efficiencyChartInstance.setOption(efficiencyChartOption); |
|  |
| // 初始化订单统计图表 |
| const orderChartInstance = echarts.init(orderChart.value); |
| const orderChartOption = { |
| title: { text: '订单统计情况' }, |
| tooltip: {}, |
| legend: { data: ['订单数量'] }, |
| xAxis: { data: factoryOverviewData.orderStatistics.map(stat => orderStatusMap[stat.orderstatus]) }, |
| yAxis: {}, |
| series: [{ |
| name: '订单数量', |
| type: 'bar', |
| data: factoryOverviewData.orderStatistics.map(stat => stat.doneordernumber) |
| }] |
| }; |
| orderChartInstance.setOption(orderChartOption); |
| }; |
|  |
| const refreshDeviceChart = async () => { |
| // 刷新设备运行情况图表数据 |
| const factoryOverviewData = await getFactoryOverviewData().then(res => res.data.data); |
| const deviceChartInstance = echarts.init(deviceChart.value); |
| deviceChartInstance.setOption({ |
| series: [{ |
| data: factoryOverviewData.equipmentStatus |
| }] |
| }); |
| }; |
|  |
| const switchEfficiencyChartType = () => { |
| efficiencyChartType.value = efficiencyChartType.value === 'line' ? 'bar' : 'line'; |
| initCharts(); |
| }; |
| </script> |
|  |
| <style scoped lang="scss"> |
| .dashboard-container { |
| width: 1300px; |
| height: 800px; |
| margin: 0 auto; |
| } |
|  |
| .chart { |
| width: 100%; |
| height: 400px; |
| } |
|  |
| .overview-text { |
| padding: 20px; |
| font-size: 16px; |
| line-height: 1.5; |
|  |
| p { |
| margin: 10px 0; |
| } |
| } |
| </style> |

**后端代码：**

|  |
| --- |
| @Override |
| public Result getView(String token) { |
|  |
| /\* 根据token获取用户数据 |
| 1.token是否在有效期 |
| 2.根据token解析userId |
| 3.根据用户id查询用数据 |
| 4.去掉密码，封装result结果返回即可\*/ |
|  |
| //是否过期 true 过期 |
| boolean expiration = jwtHelper.isExpiration(token); |
|  |
| if (expiration) { |
| //失效 未登录看待 |
| return Result.build(null, ResultCodeEnum.NOTLOGIN); |
| } |
|  |
| //根据token获取对应的用户id |
| int factoryId = jwtHelper.getUserId(token).intValue(); |
| int produceNumber = 0; |
| //获取多个pequipmentid |
| List<Integer> equipmentids = equipmentMapper.selectequipmentids(factoryId); |
|  |
| List<Map<String, Object>> equipmentMaps = new ArrayList<>(); |
| List<Map<String, Object>> productionEfficiencies = new ArrayList<>(); |
| List<Map<String, Object>> orderstatisticses = new ArrayList<>(); |
| boolean flag = true; |
| for (int equipmentId : equipmentids) { |
| //设备运行状况 工单中的设备ID和对应订单中产品的产量 |
| //LinkedHashMap是HashMap的子类，但是内部还有一个双向链表维护键值对的顺序， |
| // 每个键值对既位于哈希表中，也位于双向链表中。LinkedHashMap支持两种顺序插入顺序 、 访问顺序 |
| LinkedHashMap<String,Object> equipmentMap = new LinkedHashMap<>(); |
| List<Integer> addnumbers = producerelationMapper.selectProduceNumber(equipmentId); |
| for(int addnumber : addnumbers){ |
| produceNumber+=addnumber; |
| } |
| int States = equipmentMapper.selectStates(equipmentId); |
| equipmentMap.put("equipmentId", equipmentId); |
| equipmentMap.put("States", States); |
| equipmentMap.put("produceNumber", produceNumber); |
| equipmentMaps.add(equipmentMap); |
| } |
|  |
| List<Integer> workorderIds = workorderMapper.queryWorkorderIds(factoryId); |
| for(int workorderId : workorderIds){ |
| List<Integer> workreportIds = workreportMapper.queryWorkreportIds(workorderId); |
| for(int workreportid : workreportIds){ |
| //根据factoryid查一堆workorderid，再根据这一堆workorderid查出一堆workreportid |
| //再根据workreportid查出对应的报工日期与加工数量和合格数量 |
| //工厂生产效益 工单中的报工日期和加工数量 |
| LinkedHashMap<String,Object> productionEfficiency = new LinkedHashMap<>(); |
| Date reportdate = workreportMapper.queryreportdate(workreportid); |
| int processingnumber = workreportMapper.queryprocessingnumber(workreportid); |
| int qualifiynumber = workreportMapper.queryqualifiynumber(workreportid); |
| productionEfficiency.put("reportdate", reportdate); |
| productionEfficiency.put("processingnumber", processingnumber); |
| productionEfficiency.put("qualifiynumber", qualifiynumber); |
| productionEfficiencies.add(productionEfficiency); |
| } |
| } |
|  |
|  |
| //订单统计情况 订单状态和每个状态对应订单数量 |
| OrderStatus[] orderstatuses = OrderStatus.values(); |
| for(int i = 0; i < orderstatuses.length; i++) { |
| LinkedHashMap<String,Object> orderStatistics = new LinkedHashMap<>(); |
| Integer orderstatus = orderstatuses[i].getDescription(); |
| int ordernumber = orderMapper.queryordersnumber(orderstatus); |
| orderStatistics.put("orderstatus", orderstatus); |
| orderStatistics.put("doneordernumber", ordernumber); |
| orderstatisticses.add(orderStatistics); |
| } |
|  |
| LinkedHashMap<String,Object> data = new LinkedHashMap<>(); |
| data.put("equipmentStatus", equipmentMaps); |
| data.put("productionEfficiency", productionEfficiencies); |
| data.put("orderStatistics", orderstatisticses); |
| return Result.ok(data); |
| } |

小组成员分工及自评贡献度级别

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号** | **姓名** | **班级** | **主要任务** | **自评级别（贡献度）** |
| 202211040042 | 蔡远 | 计算机22-2 | 页面原型设计，数据库设计，前端代码，ppt制作，报告编写 | A |
| 202211040047 | 周彧函 | 计算机22-2 | 需求分析部分，状态图，时序图设计，测试 | B |
| 202211040046 | 刘聪瑞 | 计算机22-2 | 需求分析，用例图，类图设计，ppt制作，报告编写 | A |
| 202211040056 | 摆轶轩 | 计算机22-2 | 后端代码，ppt制作 | B |

自评成绩等级分为A、B、C；组内成绩不能全部一致，A不能多于2人，B不能多于2人，如果B多于2人，需在报告中注明每个组员的贡献率，并在答辩中说明。