
项目：流水线二维码产品管理系统
流水线终端子系统

2030416018 高歌

（同组：2013401036 陈阳 仓储管理子系统）

终稿

Version 5.0

2022.12.20

目录

1	子系统简介.....	4
2	功能概述.....	5
2.1	仓库管理员	5
2.1.1	打印入库标签.....	5
2.2	标签粘贴员	5
2.2.1	添加产品并打印标签.....	5
2.2.2	跳过产品并上报产线故障.....	6
2.3	质检员	6
2.3.1	标记产品缺陷.....	6
2.3.2	确认检验记录单.....	7
2.3.3	上报产线故障.....	7
2.4	生产组长.....	7
2.4.1	统计跟踪流水线情况.....	7
2.4.2	缺陷追溯.....	7
2.4.3	异常报警.....	7
2.4.4	生成生产报告	8
2.5	生产主管.....	8
2.5.1	暂停流水线.....	8
2.5.2	确认生产报告	8
3	用例建模.....	9
3.1	顶层用例图	9
3.2	仓库管理员细化用例图.....	9
3.2.1	打印入库标签.....	10
3.3	标签粘贴员与质检员细化用例图	11
3.3.1	录入工件（中间件）	11
3.3.2	缺陷标记.....	12
3.3.3	确认检验记录单.....	13
3.4	生产组长与生产主管细化用例图.....	14
3.4.1	缺陷追踪.....	14
3.4.2	查询参数异常报警.....	15
3.4.3	生成生产报告	16
3.4.4	暂停流水线.....	16
4	静态建模.....	18
4.1	类图	18
4.2	数据字典.....	18
4.2.1	产品（Product）	18
4.2.2	批次（Batch）	19
4.2.3	产品模板（ProductModel）	19
4.2.4	故障信息（FaultInfo）	19

4.2.5	缺陷信息 (DefectInfo)	20
4.2.6	生产报告 (ProductionReport)	20
4.3	方法说明	20
5	概要与设计	22
5.1	包图	22
5.2	产品缺陷信息模块	22
5.2.1	产品状态变化	22
5.2.2	缺陷上报流程	23
5.3	产品追溯图预览模块	23
5.3.1	追溯图预览流程	24
5.4	产线故障模块	24
5.4.1	故障报警流程	24
5.5	生产报告模块	25
5.5.1	生产报告创建与确认流程	25
6	数据库设计	26
6.1	E-R 图	26
6.2	数据表描述	26
6.2.1	账户 (account)	26
6.2.2	账户角色 (role)	26
6.2.3	账户权限 (permission)	27
6.2.4	缺陷信息 (defect_info)	27
6.2.5	故障信息 (fault_info)	27
6.2.6	机器 (machine)	27
6.2.7	产线 (assembly_line)	28
6.2.8	批次 (batch)	28
6.2.9	生产报告 (production_report)	28
6.2.10	检验记录单 (inspection_record_sheet)	28
6.2.11	产品 (product)	29
6.2.12	前置产品 (product_previous)	29
6.2.13	产品可选信息 (product_previous)	29
6.2.14	前置产品模板 (product_model_previous)	29
6.2.15	产品模板可选信息 (product_previous)	30
7	系统实现	31
7.1	入库 (对应仓库管理员终端)	31
7.2	产品模板 (对应生产组长和生产总管)	32
7.3	生产 (对应生产组长与生产主管)	33
7.4	质检 (对应质检员)	36
8	华为云辅助开发情况	38
8.1	工作项功能使用	38
8.2	代码托管和代码检查功能使用	38
8.3	流水线 (自动化构建+自动化部署)	39
8.4	自动化测试	40
9	UML 工具小结	41

9.1	Proccesson 思维导图	41
9.2	DRAW.IO.....	41
9.3	Enterprise Architect	41
9.4	PLANTUML	41

1 子系统简介

本子系统用于流水线终端生产（不包含仓储管理），涵盖标签打印、质检、缺陷追踪、故障报警等多个流程。

本系统的主要用户有：（1）入库管理员（2）标签打印员（3）质检员（4）生产组长（5）生产总管。

2 功能概述

2.1 仓库管理员

2.1.1 打印入库标签

仓库管理员可以在录入一批次原料后（需与仓储管理子系统结合使用）于子系统终端上选中数据库中已录入的某条记录，然后点击“打印标签”打印该批次的二维码标签，标签上应包含原料的唯一 ID、批次入库日期、标签打印日期、批次号、该批次中原料的数量、该原料在批次中的编号，以及其他可选配置字段。



用户可选择标签的大小（如仅打印二维码、ID 号和标签打印日期的小标签，或如图中包含二维码、条码以及其他详细信息的大标签），并可在标签上隐藏除 ID 外的所有字段，也可自由调整字段排序。

用户也可配置检测到新一批次原料录入后自动打印标签，或通过使用后端程序接口编写脚本调用标签打印功能。

2.2 标签粘贴员

2.2.1 添加产品并打印标签

在每个生产步骤后都应为中间品（或成品）重新打印并粘贴新标签。在每个中间批次产品生产开始前，由负责人（可以是生产组长或标签粘贴员自身）在终端上填入该批次中间品/成品的必要信息（如类别、批次号等）。



在这里将每一生产步骤抽象为一个接收若干原料并输出唯一产品的机器。每当原料进入机器时，终端系统从扫码设备读入原料上的二维码标签并将原料 ID 记录于终端缓存中，在产品生产完成后，标签粘贴员点击“打印标签”（或设置为自动打印、通过程序脚本调用接口完成打印等），终端将根据原料 ID 以及之前录入的批次信息为产品分配唯一 ID 号及其在该批次中的批号（递增），生成相应的二维码标签并打印（并上传产品信息到数据库）。上图展示了一个示例标签。

2.2.2 跳过产品并上报产线故障

机器可能因为异常情况导致标签无法正常粘贴，如机器并未输出产品、机器卡死等。或产品可能出现明显缺陷，以至于能够被标签粘贴员轻易发现。此时标签粘贴员可以选择不打印标签并在终端上直接点击“跳过产品”，在填写故障信息后上报故障。终端将跳过当前产品，直接进入下一个产品的标签打印流程。

如可供标签粘贴员填写异常信息的时间较短，以至于来不及即时填写详细信息，可选择暂时跳过填写详细信息的流程，留待该批次产品生产结束后再回头补充故障信息。

2.3 质检员

2.3.1 标记产品缺陷

在每个生产步骤中，粘贴标签后应有质检员负责检查产品是否存在缺陷。每当产品从流水线上经过时，扫码设备都会扫描产品上的二维码标签，同步查询得到当前产品的详细信息并在终端上显示，每一产品信息可能持续显示数秒。

质检员通过比对终端上显示的信息与检查产品的实际状态，判断产品是否存在缺陷，如存在缺陷则点击“标记缺陷”按钮将产品状态标记为“有缺陷”。同时质检员也可根据工厂内部约定将产品标记为其他自定义状态，如“表面划痕缺陷”、“产品不完整缺陷”等具体的缺陷状态，并添加可选备注。

如可供质检员标记缺陷的时间较短，以至于来不及即时填写缺陷信息，可选择暂时跳过填写详细信息的流程，留待该批次产品质检结束后再回头补充缺陷品的详细缺陷信息。

产品状态将决定产品下一步流向，如轻微缺陷产品进入缺陷仓库，严重缺陷产品销毁，无缺陷产品进入下一生产步骤等。这一步骤可由质检员进行，如手动将缺陷产品扫入缺陷篮，留待后续收集到缺陷仓库；也可依赖于自动化设备，如流水线上后续存在其他机器扫描产品二维码并查询该产品是否存在缺陷，然后将产品分流到后续的不同流水线上。

2.3.2 确认检验记录单

每一批次产品检验完成后，终端将显示该批次产品的检验信息，以及缺陷发生率等统计信息。质检员可根据生成的检验记录单确认该批次产品是否存在异常（如缺陷率过高等），并点击“上报异常”，填写可选备注信息后上传至数据库。在确认无误后，质检员点击“确认检验记录单”，填写可选备注信息后将完整信息上传至数据库。

2.3.3 上报产线故障

当一批次产品连续发生严重缺陷时，质检员应上报产线故障，并备注故障信息，待生产组长处理相关信息。

2.4 生产组长

2.4.1 统计跟踪流水线情况

生产组长可查看所负责产线（或部分产线）上当前正在生产的产品信息，如批次信息、产线各部分缺陷率/故障率、某一具体产品的当前生产进度（生产至第几个中间件以及当前位于产线哪一部分）等。

此外，生产组长还能查看当前产线上标签粘贴员以及质检员上报的异常及故障情况，并作确认处理。如生产组长无法直接处理问题，可将异常继续上报至生产主管。

2.4.2 缺陷追溯

当质检员上报缺陷后，生产组长可具体缺陷产品的生产流程（通过 `previous` 字段回溯该产品的上一个（或几个）中间产品，然后依次回溯至最初加工原料），并显示至终端上。生产组长可根据产品生产追溯图确定缺陷发生源头。

2.4.3 异常报警

生产组长可设置当产线某一部分的缺陷率/故障率达到某一阈值时发出报警，并配置报警方式。生产组长可在终端上查看当前触发的全部报警以及报警历史记录。

2.4.4 生成生产报告

当生产组长当前负责的批次生产完成后，将生成生产报告。报告将显示产品数量、生产大致流程（追溯图）、缺陷率、故障率等相关信息。生产组长确认生产报告无误（或标记为异常并添加备注信息）后将生产报告上传至生产主管，并由仓储管理子系统统计分析。

2.5 生产主管

2.5.1 暂停流水线

当生产组长向生产主管上报严重故障时，生产主管可考虑暂停部分或全部流水线，并将流水线当前状态上报至仓储管理子系统。

2.5.2 确认生产报告

生产主管应确认生产组长上传的生产报告，并在各产线均完成生产后生成总生产报告并将生产报告上传至数据库。

3 用例建模

3.1 顶层用例图

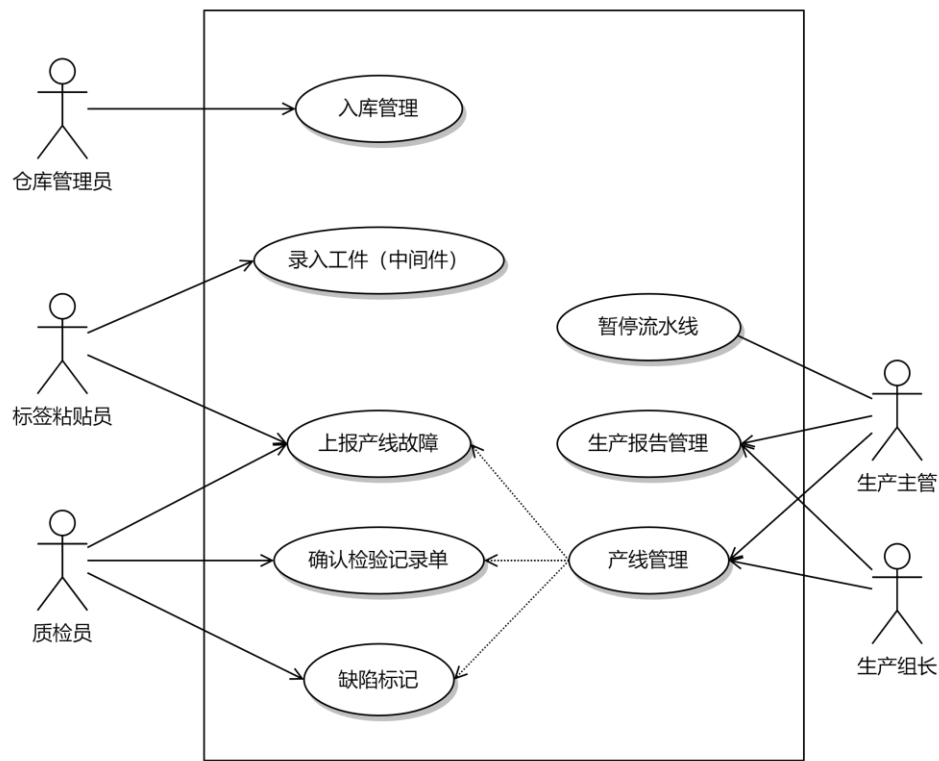


图 3-1 流水线终端子系统顶层用例图

3.2 仓库管理员细化用例图

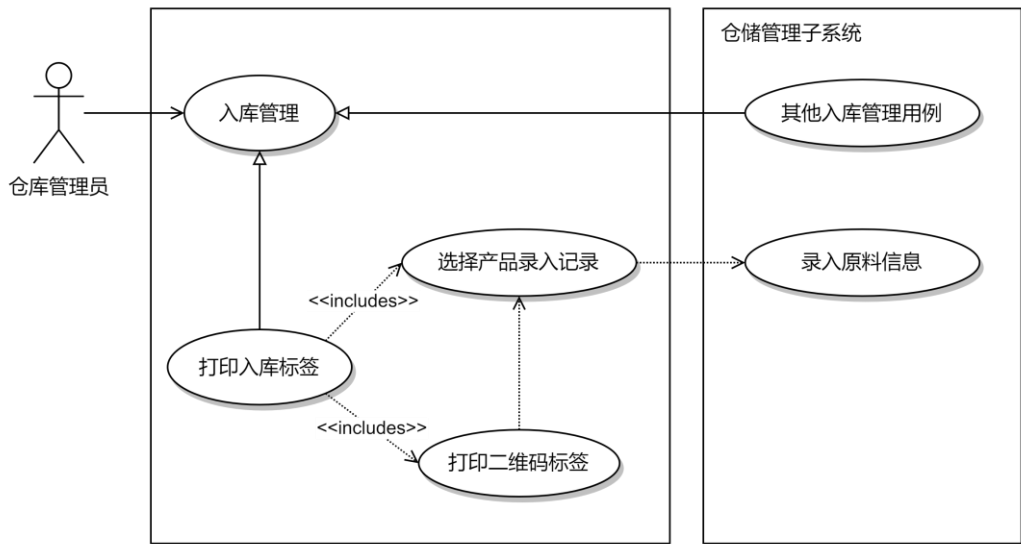


图 3-2 仓库管理员细化用例图

3.2.1 打印入库标签

用例简述：仓库管理员在一批次原料进入仓库时，为所有原料登记入库信息后（由仓储管理子系统负责），在仓储终端上选择录入信息并打印用于粘贴于原料上的入库标签。

行为者：仓库管理员

前置条件：

- ① 原料入库且入库信息登记完毕（由仓储管理子系统负责）

后置条件：

- ① 入库标签打印完毕，将产品状态更新为待出库

基本事件流：

- ① 用户在终端上选择已经登记（且未打印标签）的产品批次
- ② 用户选择批次中需要打印标签的产品编号范围
- ③ 用户调整标签样式，选择需要显示在标签上的相关信息
- ④ 用户预览标签样式
- ⑤ 用户点击打印，打印机将打印入库标签
- ⑥ 打印完成后，系统将所选产品的状态更新为待出库

备选事件流：

- ① 输入的产品编号范围中包含已经打印标签的产品，系统将提示用户重新输入编号
- ② 预览标签不符合用户需求，用户返回上一步重新调整标签样式
- ③ 打印完成后，用户认为打印标签质量不符合预期，取消相关产品的状态更新并重新打印

异常事件流：

- ① 打印机故障，无法打印全部或部分标签，系统将提示用户发生异常。若已打印部分标签，用户可选择保留已经打印的标签并更新这部分产品的状态为待出库，或放弃已打印成功的标签

3.3 标签粘贴员与质检员细化用例图

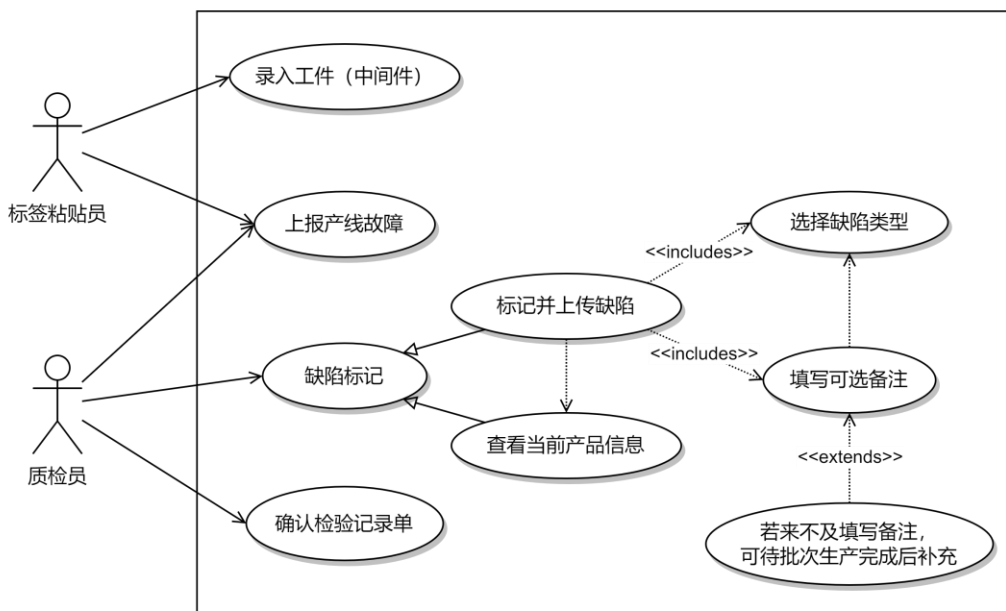


图 3-3 标签粘贴员与质检员细化用例图

3.3.1 录入工件（中间件）

用例简述：每个生产步骤后的标签粘贴员在工件（中间件）生产完成后，扫描工件上已有的标签（可能是上一个生产步骤打印的标签或原料标签）并在终端上查看工件信息，打印用于粘贴的新标签。

行为者：标签粘贴员

前置条件：

- ① 工件在当前生产步骤生产完成，且产品信息已通过已有的标签录入终端

后置条件：

- ① 待粘贴的新标签打印完毕，更新产品的实时加工信息

基本事件流：

- ① 用户查看通过已有标签二维码自动录入的当前产品信息
- ② 用户点击打印，打印机将打印新标签
- ③ 打印完成后，系统将所选产品的状态更新为待出库

备选事件流：

-
- ① 自动录入出现问题，用户使用扫码枪手动重新录入当前产品信息
 - ② 打印完成后，用户认为打印标签质量不符合预期，取消相关产品的状态更新并重新打印

异常事件流：

- ① 终端上显示的当前产品信息有误，用户将异常信息上报至生产组长
- ② 打印机故障，无法打印标签，系统将提示用户发生异常，且不再更新产品状态，并将异常上报至生产组长

3.3.2 缺陷标记

用例简述：每个生产步骤后的质检员在标签粘贴员录入工件（中间件）后，扫描工件上的标签并在终端上查看工件信息，并与工件实际状态进行核对，根据实际情况选择是否将工件标记为缺陷品，并选择缺陷类别与填写可选缺陷备注。

行为者：质检员

前置条件：

- ① 工件已由标签粘贴员粘贴当前生产步骤的新标签

后置条件：

- ① 质检完毕后，更新产品状态为当前生产步骤已检验

基本事件流：

- ① 用户查看通过已有标签二维码自动录入的当前产品信息
- ② 用户比对终端上显示的产品信息并检查产品的实际状态
- ③ 若用户发现产品存在缺陷，通过下面的步骤标记缺陷，否则不进行任何操作，等待当前产品正常经过后产品状态将自动更新为已检验且无缺陷
- ④ 用户选择产品缺陷类别
- ⑤ 用户添加可选的缺陷备注信息
- ⑥ 用户点击上报当前产品缺陷
- ⑦ 缺陷上报完成后，当前产品的状态更新为已检验且包含缺陷，待下一步于分流装置处分流至缺陷仓库

备选事件流：

-
- ① 自动录入出现问题，用户使用扫码枪手动重新录入当前产品信息
 - ② 用户来不及填写缺陷备注信息，留待当前批次产品生产完毕后将当前产品的缺陷备注信息与检验记录单一同提交

异常事件流：

- ① 终端上显示的当前产品信息有误，用户将异常信息上报至生产组长

3.3.3 确认检验记录单

用例简述：每一批次产品质检完毕后，终端将自动生成检验记录单，质检员在确认后（可填写备注信息）提交检验记录单。

行为者：质检员

前置条件：

- ① 当前批次工件全部质检完成

后置条件：

- ① 检验记录单提交后，产品所在批次状态更新为已全部检验

基本事件流：

- ① 用户查看自动生成的检验记录单
- ② 用户根据实际情况填写可选备注信息，若之前批次中某一缺陷产品未来得及填写缺陷备注，则在此处填写
- ③ 用户点击确认，上传检验记录单

备选事件流：

- ① 检验记录单存在异常情况（如缺陷率过高），用户在提交检验记录单的同时上报异常（并填写可选备注信息）

异常事件流：

- ① 检验记录单有误，用户将错误上报至生产组长

3.4 生产组长与生产主管细化用例图

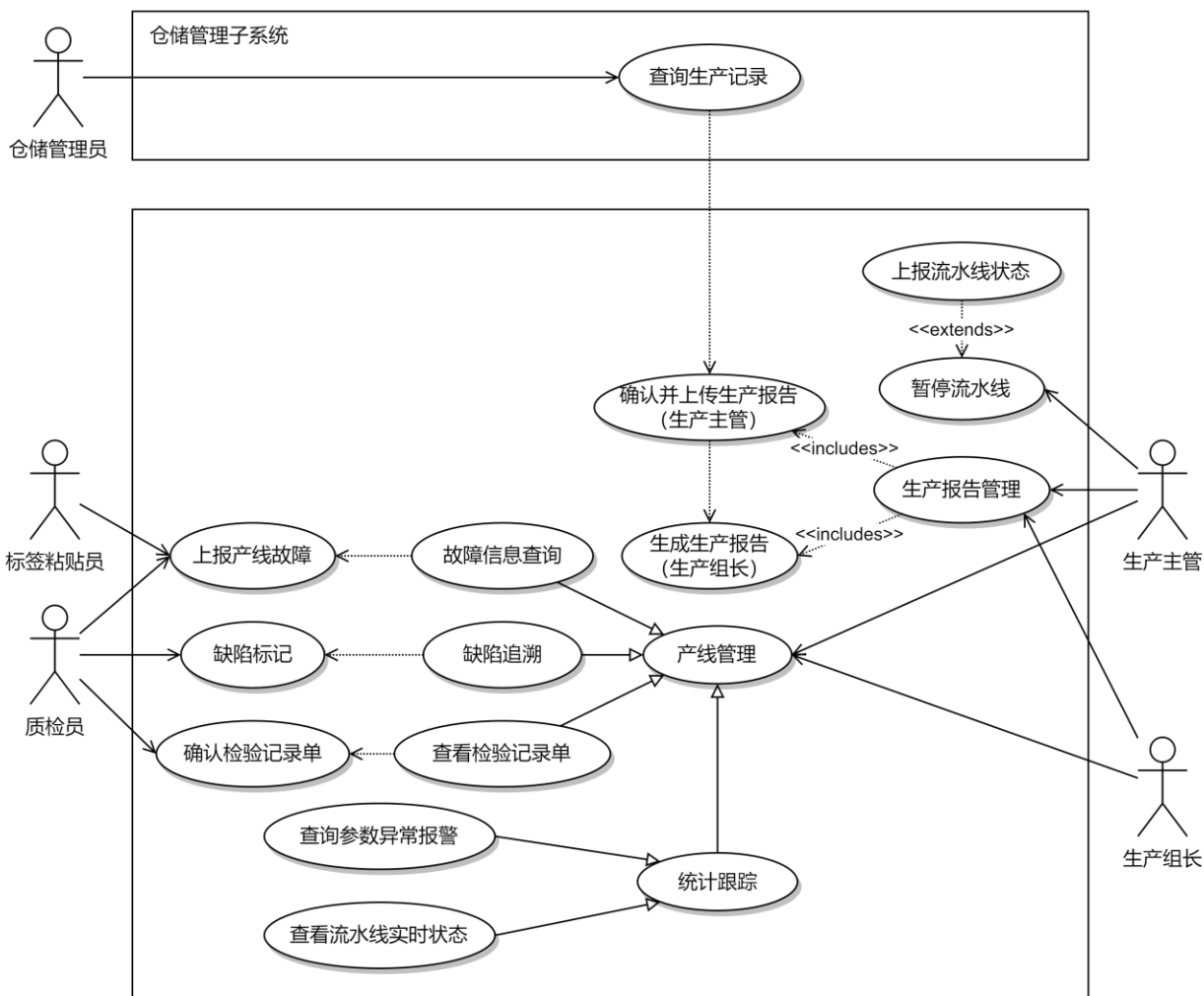


图 3-4 生产组长与生产主管细化用例图

3.4.1 缺陷追踪

用例简述：生产组长或生产主管可实时查看产线上缺陷品的当前状态，并跟踪其生产流程（以流程图显示），以确认缺陷来源。

行为者：生产组长、生产主管

前置条件：

- ① 产品在某一生产步骤中被质检员上报存在缺陷

后置条件：

- ① 终端上显示产品的生产流程图及实时信息

基本事件流：

-
- ① 用户查看当前产线上存在的缺陷品及其简要信息
 - ② 用户选择某一缺陷品查看其生产流程图及详细信息
 - ③ 用户查看相关信息并根据实际情况确认缺陷来源，并确定相关负责人（生产工人、质检员等）

备选事件流：

无

异常事件流：

- ① 产品信息存储出错，系统无法正确读取并显示生产流程图，提示用户存在异常

3.4.2 查询参数异常报警

用例简述：生产组长或生产主管可设置当流水线参数（如故障率、缺陷率、传送带速率等）满足某一条时在系统中发出报警，并在报警产生后查询报警信息。

行为者：生产组长、生产主管

前置条件：

- ① 用户已设置合理的报警条件（当某一参数满足某一条件时触发报警，当条件满足重复 n 次/周期内重复多次时触发报警等）并被触发，在终端上弹出提示泡显示报警触发

后置条件：

- ① 终端显示报警详细信息

基本事件流：

- ① 用户查看历史报警记录
- ② 用户选择某一具体报警信息并查看其详细信息
- ③ 用户查看相关信息并根据实际情况决定是否及如何调整生产线状态

备选事件流：

无

异常事件流：

- ① 报警条件设置不合理，发生循环触发，系统将提示用户报警条件存在问题

3.4.3 生成生产报告

用例简述：产线当前批次产品生产完毕后，终端将自动生成生产报告，由生产组长补充无法自动生成的相关信息后确认后生成生产报告无误（或填写备注信息），并将其上传至生产主管。

行为者：生产组长

前置条件：

- ① 当前批次产品生产完毕，且质检员已全部提交检验记录单

后置条件：

- ① 当前批次状态更新为生产完成，待生产主管确认

基本事件流：

- ① 用户查看并核对终端自动生成的生产报告
- ② 用户补充填写相关信息
- ③ 用户点击确认，上传生产报告至生产组长

备选事件流：

- ① 用户发现生产报告存在问题，修改部分自动生成结果

异常事件流：

- ① 用户发现生产报告有误且难以修正，将错误上报至生产主管

3.4.4 暂停流水线

用例简述：当流水线出现严重故障时，生产主管可选择暂停流水线，以派遣人员处理故障。

行为者：生产主管

前置条件：

- ① 流水线存在严重故障

后置条件：

- ① 流水线成功暂停，系统将流水线状态上传至仓储管理子系统

基本事件流：

- ① 用户查看当前所有流水线列表

-
- ② 用户选择某一具体流水线并将其暂停（或选择多个或全部流水线并将其暂停）
 - ③ 流水线成功暂停后，系统中断所选流水线的生产进程，并将流水线状态更新为停止

备选事件流：

无

异常事件流：

- ① 流水线无法正常暂停，系统取消已经修改的流水线状态并提示用户控制系统发生故障

4 静态建模

4.1 类图

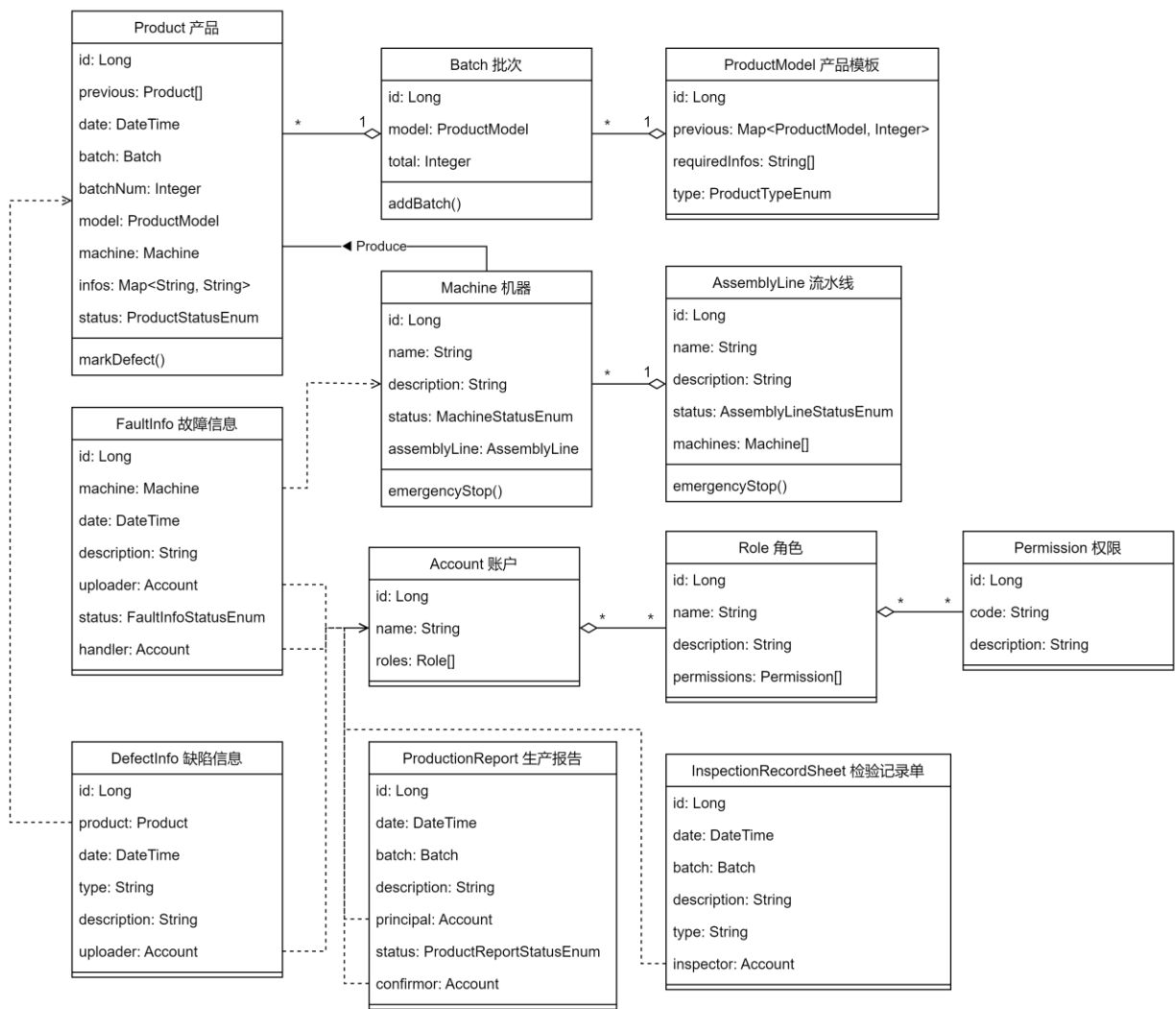


图 4-1 类图（静态建模）

4.2 数据字典

4.2.1 产品（Product）

字段	类型	描述
id	Long	产品 ID
previous	Product[]	生产该产品的前置产品，用于追溯
date	DateTime	产品生产时间

batch	Batch	产品所在批次
batchNum	Long	产品在批次中的编号
model	ProductModel	产品所属产品模板（见下）
machine	Machine	加工产品使用的机器
infos	Map<String, String>	产品自定义可选信息
status	ProductStatusEnum	产品状态（待生产/正在生产/已生产/检验合格/检验不合格/已消耗）

4.2.2 批次（Batch）

字段	类型	描述
id	Long	批次 ID
model	ProductModel	该批次生产的产品模板（必须是成品模板，不能选择中间品模板或原料模板）
total	Integer	批次生产的产品数量

4.2.3 产品模板（ProductModel）

字段	类型	描述
id	Long	模板 ID
previous	Map<ProductModel[], Integer>	生产该产品（模板）的前置产品（模板），用于在创建批次时决定生产计划。之所以不同于 Product ，不是个数组而是个 Map ，是因为一个产品可能由多件其他产品生产而来，比如中间品 1 需要 2 个原料 A 和 3 个原料 B 才能生产，成品 1 需要 2 个中间品 1 才能生产，所以采用 Map 表示
requiredInfos	String[]	可选信息，用于约束模板的实例（即 Product ）中的可选信息取值
type	ProductTypeEnum	模板类型（原料/中间品/成品）

4.2.4 故障信息（FaultInfo）

字段	类型	描述
id	Long	故障信息 ID
machine	Machine	发生故障的机器
date	DateTime	故障发生时间

description	String	故障描述
uploader	Account	故障信息上传者（标签粘贴员/质检员）
status	FaultInfoStatusEnum	故障处理状态（待处理/正在处理/已处理/已忽略）
handler	Account	故障处理者（生产组长）

4.2.5 缺陷信息 (DefectInfo)

字段	类型	描述
id	Long	缺陷信息 ID
product	Product	与缺陷相关的产品
date	DateTime	缺陷上传时间
type	String	缺陷类型
description	String	缺陷描述
uploader	Account	缺陷信息上传者（质检员）

4.2.6 生产报告 (ProductionReport)

字段	类型	描述
id	Long	生产报告 ID
batch	Long	生产报告所关联的批次号
date	DateTime	报告生成时间
description	String	生产报告描述
principal	Account	报告生成者（生产组长）
status	String	报告处理状态（待确认/已确认）
confirmer	Account	报告确认者（生产总管）

4.3 方法说明

上面类图中出现的所有类均为实体类，且均具有增删改查方法，为避免冗余在这里全部省略，仅保留几个逻辑稍微特殊的方法留以说明。

1. 机器与流水线均具有 emergencyStop() 方法，用于生产总管在紧急情况下停止流水线。
2. 新建批次时（addBatch），需要选择一个产品模板，且该产品模板的类型必须是“成品”，并输入期望生产的成品数量。发送请求后，后台会根据 productModel 的

`previous` 字段计算生产这些成品需要多少中间品、原料，并创建相应的 **Product** 实体，将它们的状态标记为“待生产”，然后返回。

3. 将某已生产完毕但尚未质检的产品标记为“缺陷品”（`markDefect`）时，后台会将该产品视为生产失败，并且会根据当前批次的生产计划补上新的待生产产品，这些产品的信息同样会返回。

5 概要与设计

5.1 包图

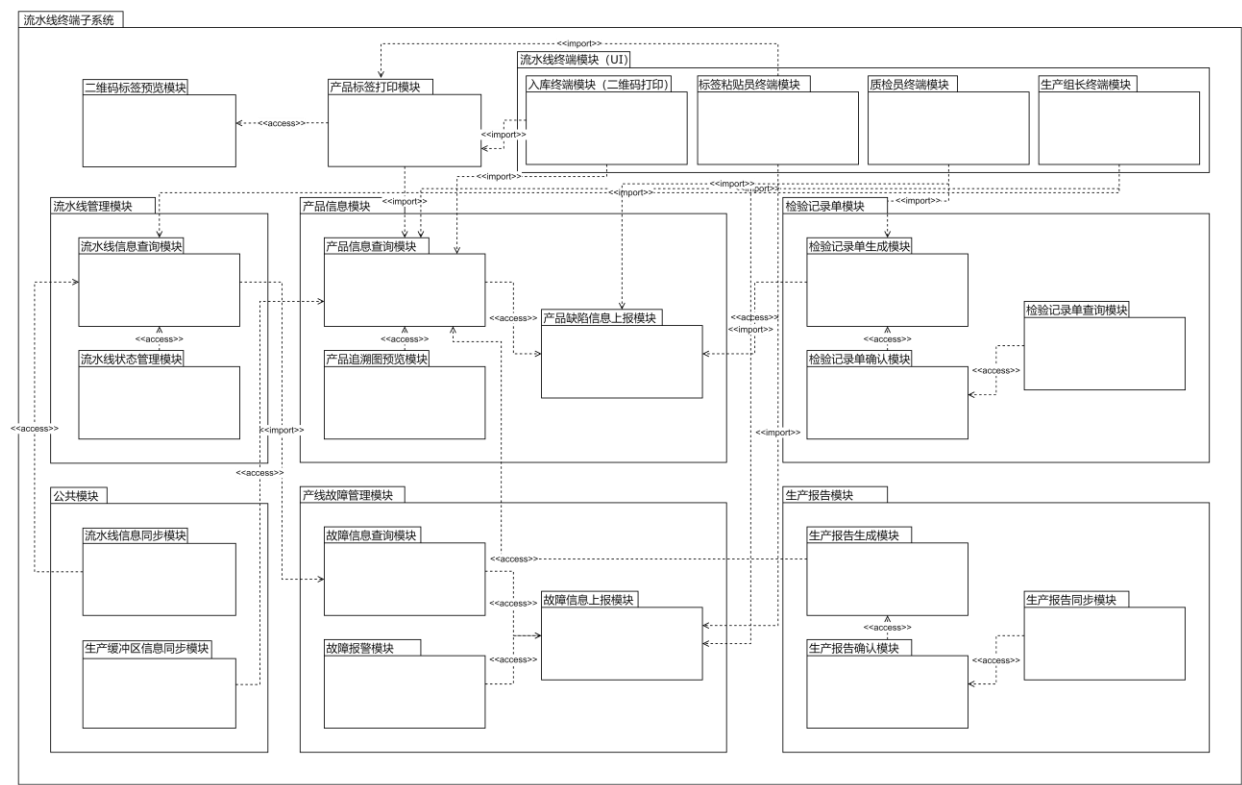


图 5-1 包图（概要设计）

5.2 产品缺陷信息模块

模块描述：产品缺陷信息模块负责管理产品的缺陷信息，用于质检步骤。

5.2.1 产品状态变化

产品信息模块中对产品状态的更新流程如下面的状态图所示。

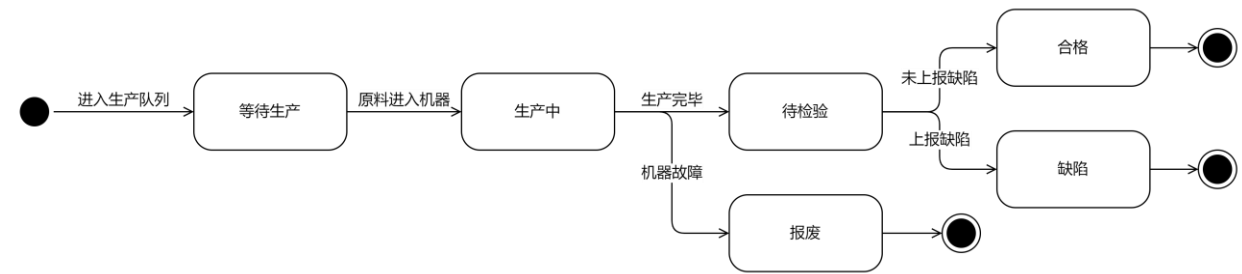


图 5-2 产品状态变化（状态图）

5.2.2 缺陷上报流程

上报产品缺陷信息的流程如以下顺序图所示。

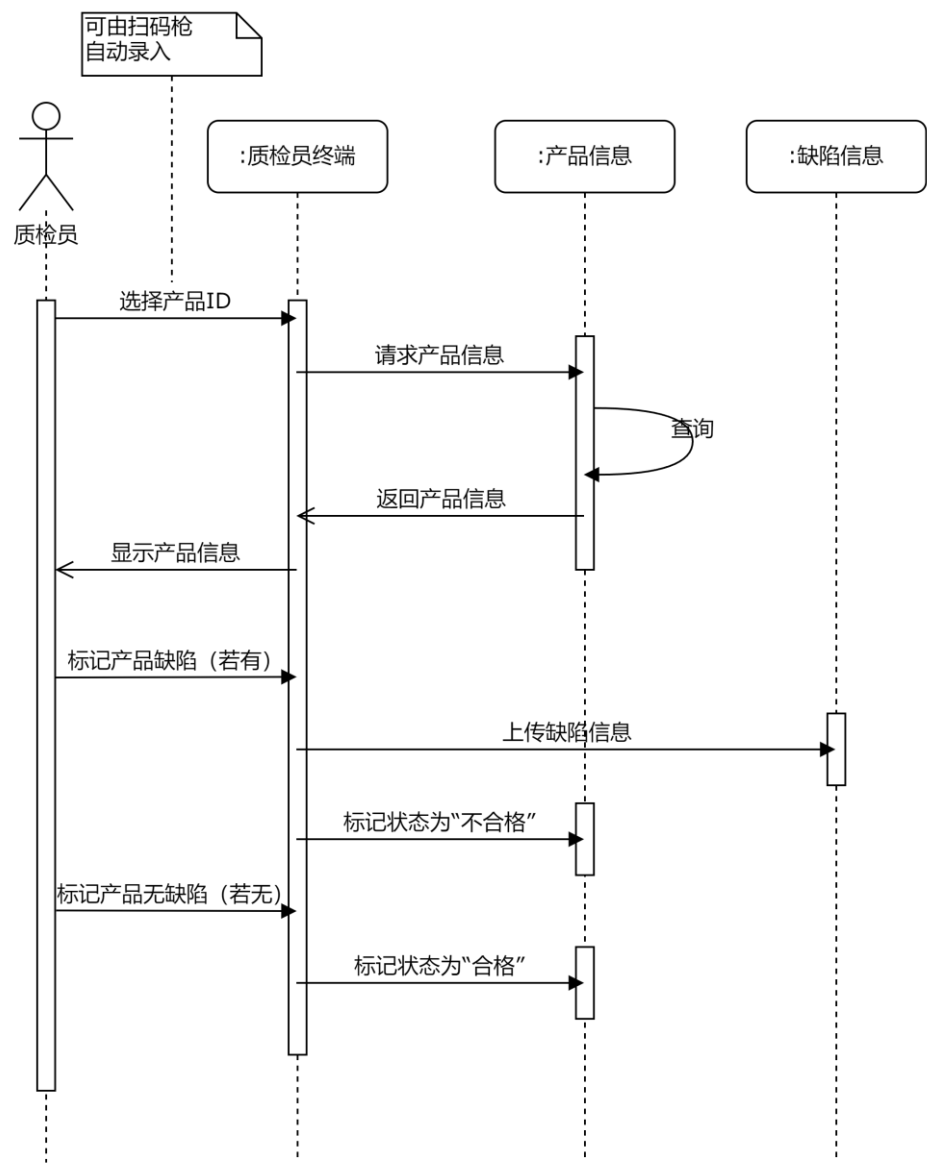


图 5-3 质检流程（顺序图）

质检过程中如要标记缺陷，会先上传缺陷信息，然后将产品状态标记为“不合格”。

5.3 产品追溯图预览模块

模块描述：产品追溯图预览模块负责预览产品追溯情况。

5.3.1 追溯图预览流程

预览追溯图的流程如以下活动图所示。

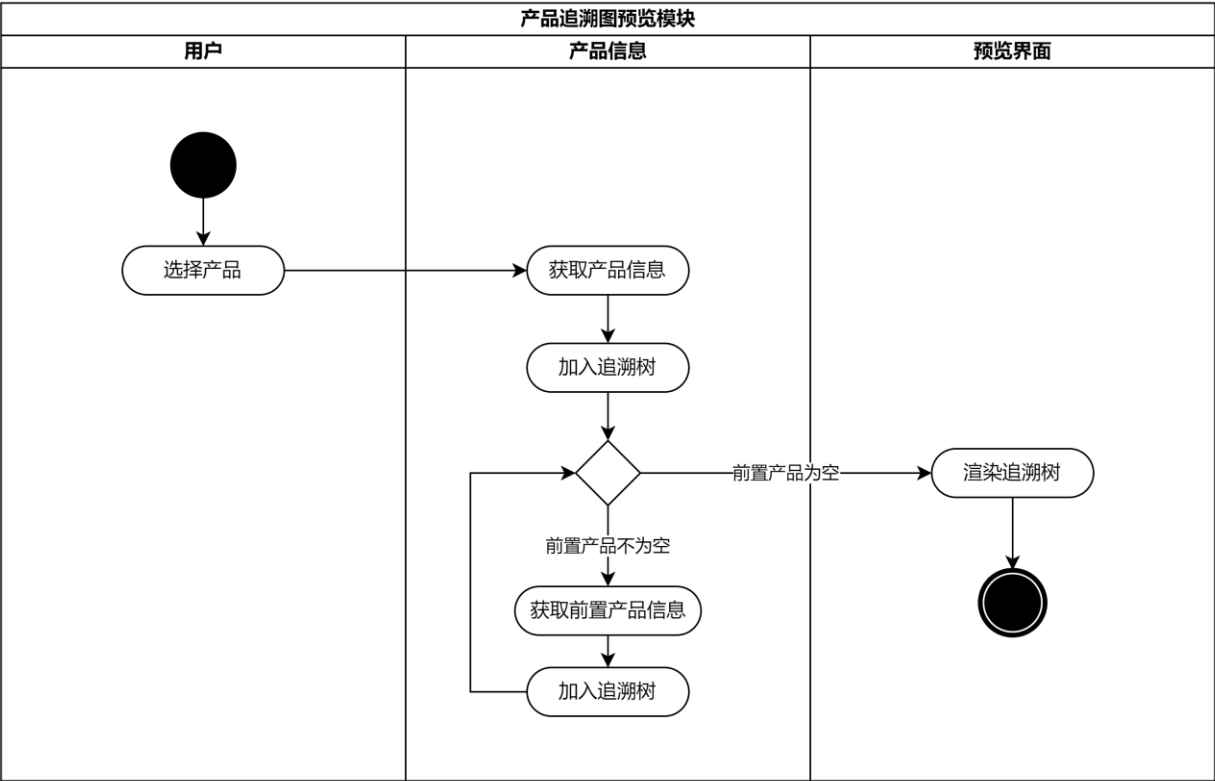


图 5-4 追溯图预览流程（活动图）

预览追溯图时，会先获取所选产品信息，并将必要信息加入树形数据结构追溯树中。若有前置产品，则递归地获取前置产品信息与其前置产品并加入追溯树，直至前置产品数组为空。追溯树构建完成后，预览界面通过追溯树渲染追溯图。

5.4 产线故障模块

模块描述：产品故障模块负责流水线与机器的故障及报警信息。

5.4.1 故障报警流程

下图使用协作图展示了故障报警的流程。

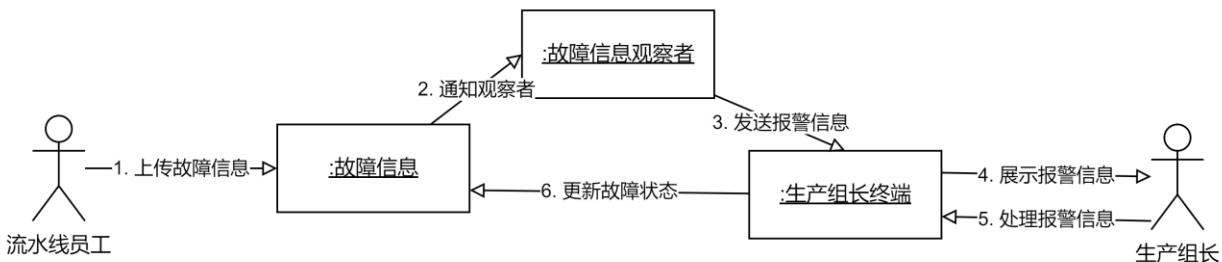


图 5-5 故障报警流程（协作图）

当流水线员工（标签粘贴员或质检员）发现机器故障时，上报故障信息，然后生产组长终端显示故障报警，生产组长确认后，处理报警信息并变更故障信息状态为“已处理”。

此处使用了观察者模式，主体为故障信息，观察者负责观察当前产线的故障信息，处理成报警信息后发送到生产组长终端。

5.5 生产报告模块

模块描述：生产报告模块负责管理生产报告（每一批次生产完成后由生产组长填写并由生产总管确认）

5.5.1 生产报告创建与确认流程

下图使用协作图展示了生产报告的创建与确认流程。

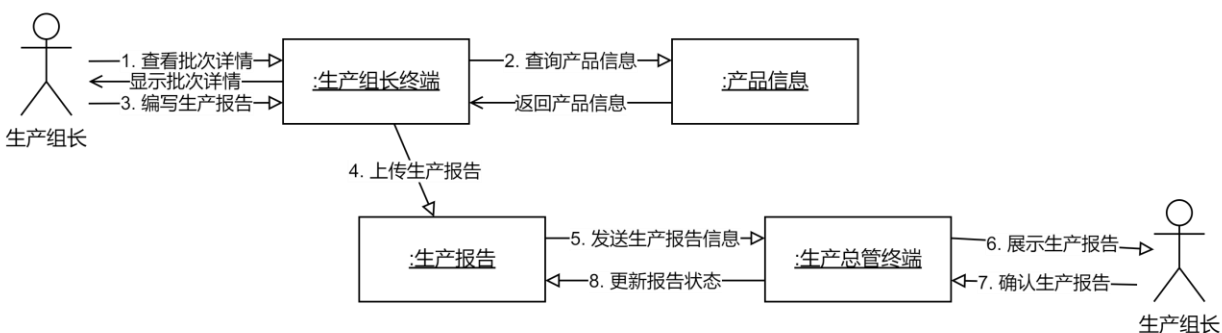


图 5-6 生产报告创建与确认流程（协作图）

当一批次的产品生产完成后，生产组长查看该批次产品的生产详情，然后编写上传生产报告。生产主管接收到生产报告后，确认生产报告并将生产报告状态更新为“已确认”。

6 数据库设计

6.1 E-R 图

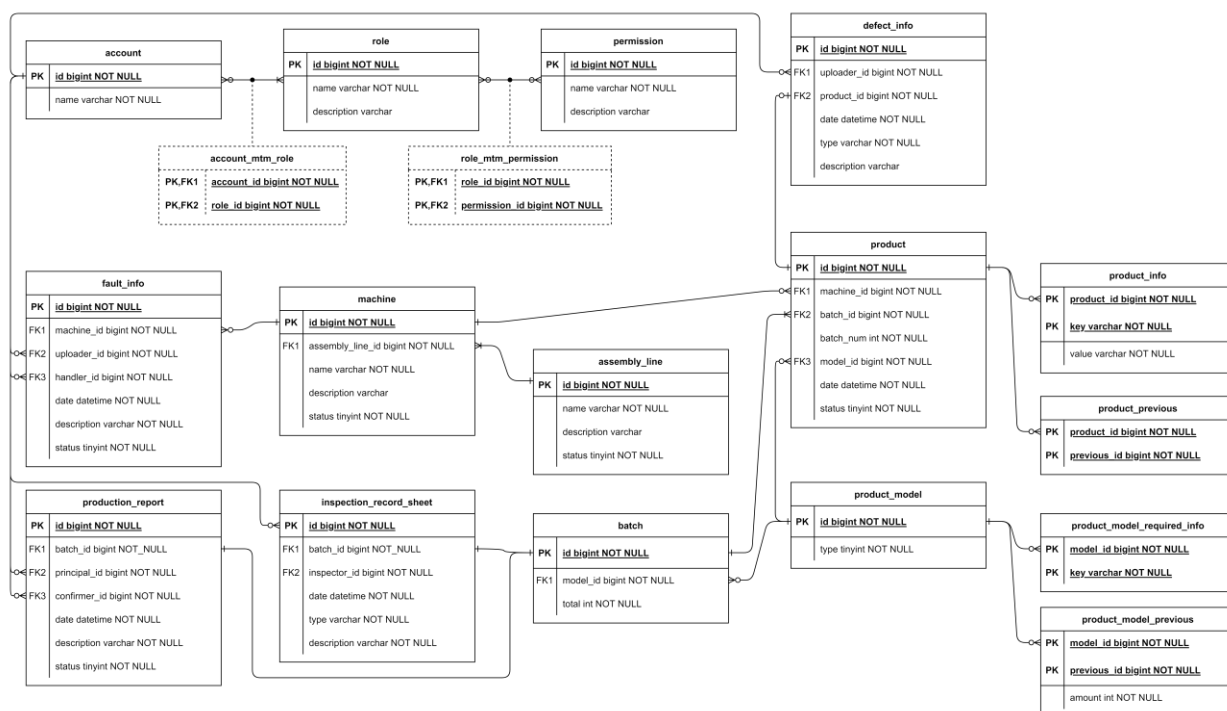


图 6-1 数据库 E-R 图

数据库大致上与类图（数据类）呈现对应关系，不同之处仅在于类图中实例属性的类型不受限，因此很多地方可以直接将类型标为 Enum、数组、Map 甚至另一个实体类，而转换成 E-R 图时，需遵从数据库设计范式，数组、Map、其他实体类等作为类型时需转换为相应的关联表。其余均同类图，因此不再详述。主外键关系均已标注在 E-R 图上。

6.2 数据表描述

6.2.1 账户（account）

字段	类型	非空	主键	外键	描述
id	bigint	√	√		账户 ID
name	varchar	√			账户名

6.2.2 账户角色（role）

*注：可为管理员、标签粘贴员、质检员、生产组长或生产总管，用于区分权限

字段	类型	非空	主键	外键	描述
id	bigint	√	√		角色 ID

name	varchar	√			角色名
description	varchar				角色描述

6.2.3 账户权限 (permission)

*注：归属于角色，用于细化权限

字段	类型	非空	主键	外键	描述
id	bigint	√	√		权限 ID
name	varchar	√			权限名
description	varchar				权限描述

6.2.4 缺陷信息 (defect_info)

字段	类型	非空	主键	外键	描述
id	bigint	√	√		缺陷信息 ID
uploader_id	bigint	√		√	上传者账户 ID
product_id	bigint	√		√	产品 ID
date	datetime	√			上传日期
type	varchar	√			缺陷类型
description	varchar				缺陷描述

6.2.5 故障信息 (fault_info)

字段	类型	非空	主键	外键	描述
id	bigint	√	√		故障信息 ID
machine_id	bigint	√		√	机器 ID
uploader_id	bigint	√		√	上传者账户 ID
handler_id	bigint	√		√	处理者账户 ID
date	datetime	√			上传日期
status	tinyint	√			处理状态
description	varchar	√			故障描述（必填）

6.2.6 机器 (machine)

字段	类型	非空	主键	外键	描述
id	bigint	√	√		机器 ID

assembly_line_id	bigint	√		√	产线 ID
name	varchar	√			机器名
status	tinyint	√			机器状态
description	varchar				备注

6.2.7 产线 (assembly_line)

字段	类型	非空	主键	外键	描述
id	bigint	√	√		产线 ID
name	varchar	√			产线名
status	tinyint	√			产线状态
description	varchar				备注

6.2.8 批次 (batch)

字段	类型	非空	主键	外键	描述
id	bigint	√	√		批次号
model_id	bigint	√		√	产品模型 ID
total	int	√			生产数量

6.2.9 生产报告 (production_report)

字段	类型	非空	主键	外键	描述
id	bigint	√	√		报告 ID
batch_id	bigint	√		√	批次 ID
principal_id	bigint	√		√	责任人 ID (生产组长)
confirmer_id	bigint	√		√	确认人 ID (生产总管)
date	datetime	√			上传日期
status	tinyint	√			确认状态
description	varchar	√			描述

6.2.10 检验记录单 (inspection_record_sheet)

字段	类型	非空	主键	外键	描述
id	bigint	√	√		报告 ID
batch_id	bigint	√		√	批次 ID
inspector_id	bigint	√		√	质检员 ID

date	datetime	√			上传日期
type	varchar	√			记录单类型
description	varchar	√			描述

6.2.11 产品 (product)

字段	类型	非空	主键	外键	描述
id	bigint	√	√		报告 ID
machine_id	bigint	√		√	机器 ID
batch_id	bigint	√		√	批次 ID
batch_num	int	√			在批次中的编号
model_id	bigint	√		√	产品模板 ID
date	datetime	√			上传日期
status	tinyint	√			产品状态

6.2.12 前置产品 (product_previous)

*注：该表用于追溯产品生产流程

字段	类型	非空	主键	外键	描述
product_id	bigint	√	√	√	产品 ID
previous_id	bigint	√	√	√	前置产品 ID

6.2.13 产品可选信息 (product_previous)

*注：该表用于保存产品附带的可选信息

字段	类型	非空	主键	外键	描述
product_id	bigint	√	√	√	产品 ID
key	varchar	√	√		可选信息键
value	varchar	√			可选信息值

6.2.14 前置产品模板 (product_model_previous)

*注：该表用于新建批次时生成待生产产品

字段	类型	非空	主键	外键	描述
model_id	bigint	√	√	√	模板 ID
previous_id	bigint	√	√	√	前置模板 ID

6.2.15 产品模板可选信息 (product_previous)

*注：该表用于约束产品信息（因为仅起约束作用，所以只需要 key）

字段	类型	非空	主键	外键	描述
model_id	bigint	√	√	√	产品 ID
key	varchar	√	√		可选信息键

7 系统实现

7.1 入库（对应仓库管理员终端）

在“原料入库”界面，用户可向系统中录入原料、删除录入信息。

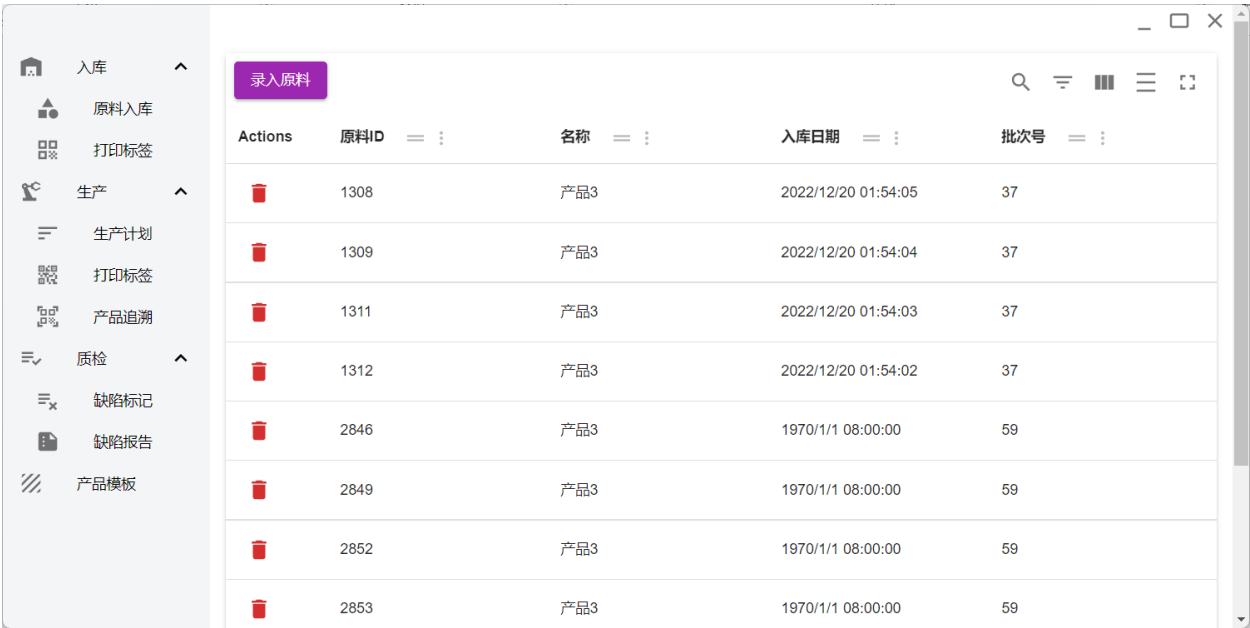


图 7-1 原料入库界面

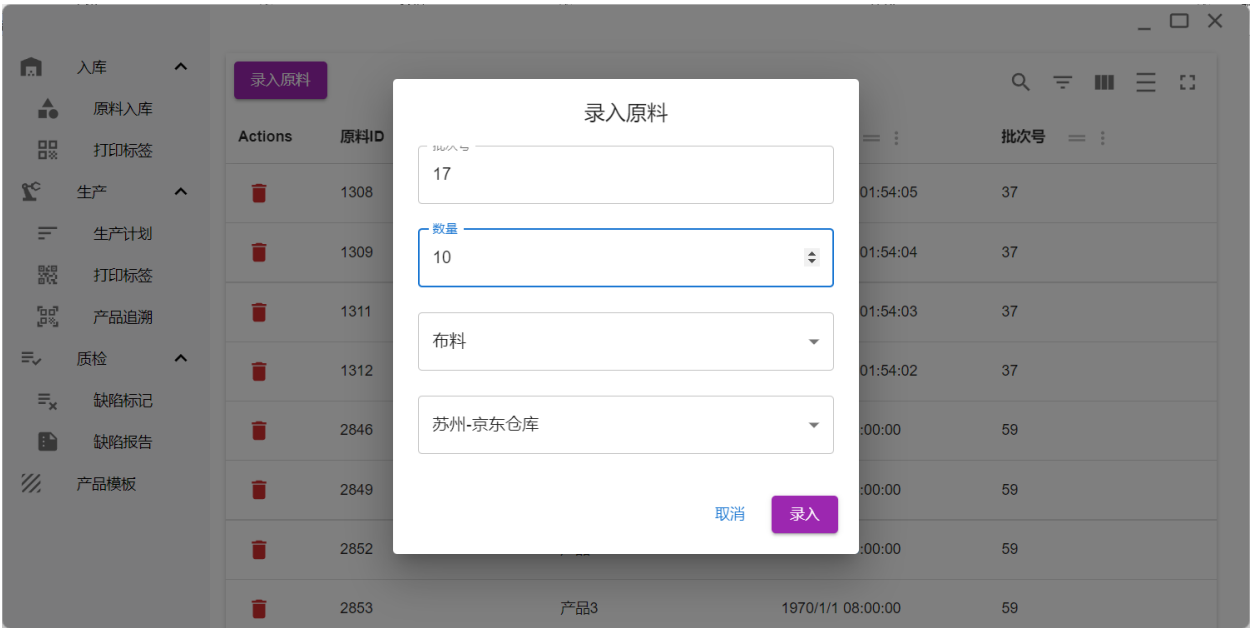


图 7-2 录入原料

在“打印标签”界面，用户可选中原料并打印标签。

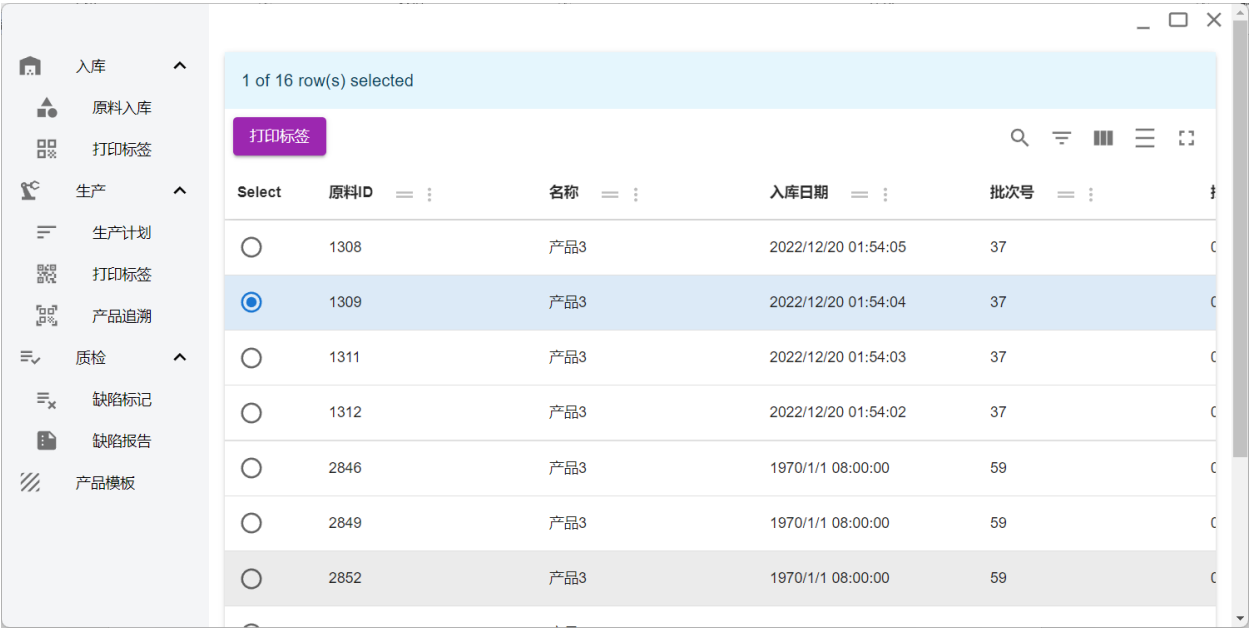


图 7-3 打印标签界面



图 7-4 打印标签预览

7.2 产品模板（对应生产组长和生产总管）

在产品模板界面，用户可创建和删除产品模板，并查看前置产品信息。

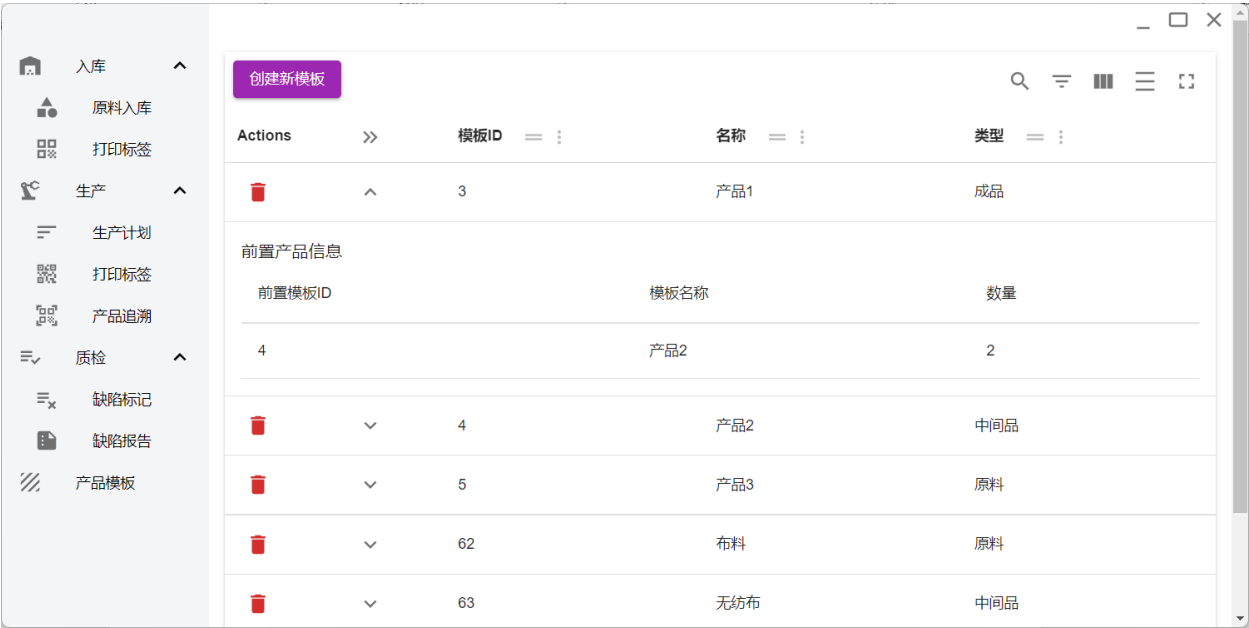


图 7-5 产品模板界面



图 7-6 创建新模板

7.3 生产（对应生产组长与生产主管）

在“生产计划”界面，用户可新建批次，选择成品模板，填写生产数量，系统将会递归地根据前置产品模板创建对应的待生产产品，放入待生产列表中。

Actions	产品ID	名称	状态	加工日期
	1307	产品2	检验不合格	2022/12/19 19:08:08
	1308	产品3	检验合格	2022/12/20 01:54:05
	1309	产品3	检验合格	2022/12/20 01:54:04
	1310	产品2	待生产	1970/1/1 08:00:00
	1311	产品3	检验合格	2022/12/20 01:54:03
	1312	产品3	检验合格	2022/12/20 01:54:02
	1455	产品N	检验不合格	2022/12/18 04:57:26
	2846	产品3	待生产	1970/1/1 08:00:00

图 7-7 生产计划界面

新建批次

口罩

数量1000

取消新建

图 7-8 新建批次

在“打印标签”界面，用户可以选择已经生产好但未进行质检的产品并打印标签。打印标签时，用户可填写其前置产品信息（或是自动填写）。该信息用于生成产品追溯图。

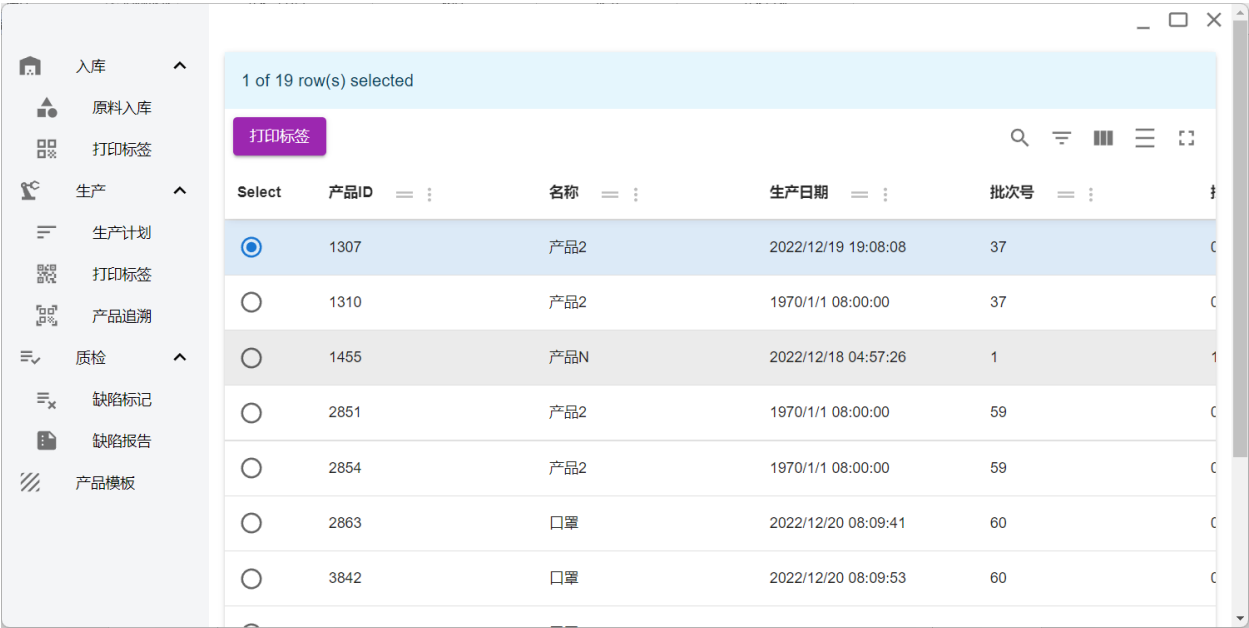


图 7-9 打印标签界面

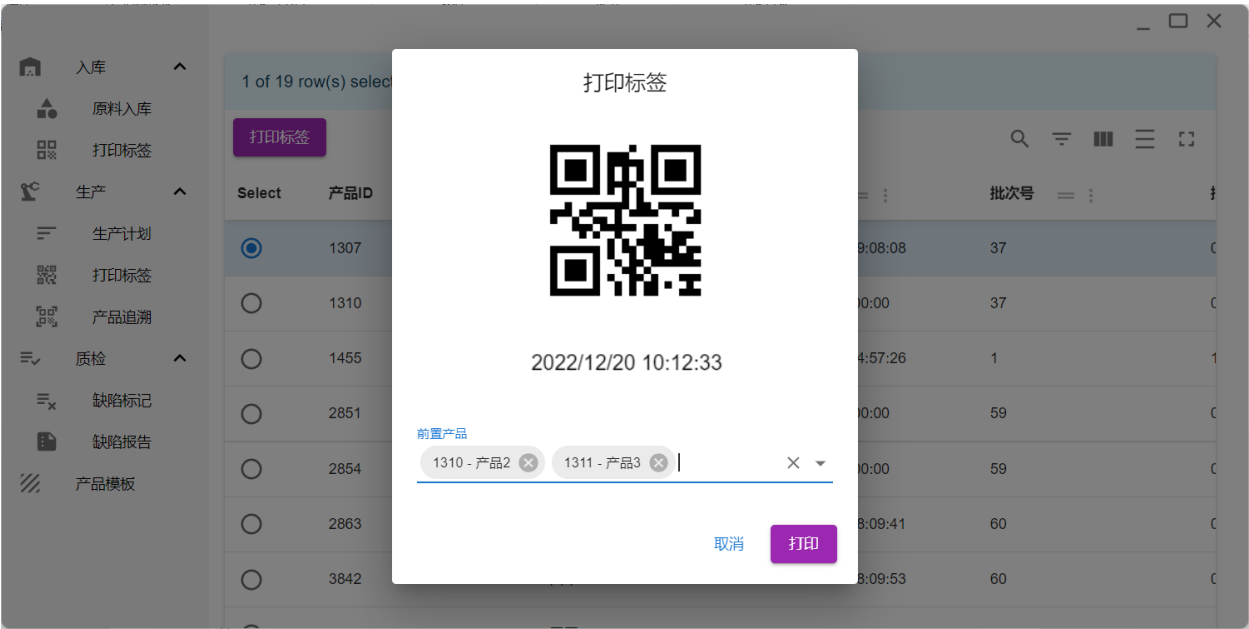


图 7-10 打印标签预览

在“产品追溯”界面，用户可选择产品，查看追溯图。追溯图会根据产品信息的前置产品自动生成。

入库

原料入库

打印标签

生产

生产计划

打印标签

产品追溯

质检

缺陷标记

缺陷报告

产品模板

1 of 19 row(s) selected

查看追溯图

Select	产品ID	名称	生产日期	批次号
<input checked="" type="radio"/>	1307	产品2	2022/12/19 19:08:08	37
<input type="radio"/>	1310	产品2	1970/1/1 08:00:00	37
<input type="radio"/>	1455	产品N	2022/12/18 04:57:26	1
<input type="radio"/>	2851	产品2	1970/1/1 08:00:00	59
<input type="radio"/>	2854	产品2	1970/1/1 08:00:00	59
<input type="radio"/>	2863	口罩	2022/12/20 08:09:41	60
<input type="radio"/>	3842	口罩	2022/12/20 08:09:53	60

图 7-11 产品追溯界面



图 7-12 产品追溯图预览

7.4 质检（对应质检员）

在“缺陷标记”界面，用户可将一件已生产的产品标记为缺陷（需要上传缺陷信息）或无缺陷。

入库

原料入库

打印标签

生产

生产计划

打印标签

产品追溯

质检

缺陷标记

缺陷报告

产品模板

缺陷品标记

1455 - 产品N

缺陷类型

划痕

缺陷描述

标记缺陷

标记非缺陷

图 7-13 缺陷标记界面

在“缺陷报告”界面，用户可查看缺陷报告。

入库

原料入库

打印标签

生产

生产计划

打印标签

产品追溯

质检

缺陷标记

缺陷报告

产品模板

>>	缺陷信息ID	产品ID	上传时间	缺陷类型
▼	1	1	2022-12-17T00:00:00	否
▲	20	1455	2022-12-20T00:00:00	划痕
1234				

Rows per page 10 1-2 of 2

图 7-14 缺陷报告界面

8 华为云辅助开发情况

本次文档撰写及开发过程中均使用了华为的 DevOps 平台 DevCloud，以下为使用情况及小结。

8.1 工作项功能使用

通过华为云的工作项功能，和另一名小组队友完成了计划的制定和时间表的确定，划分了彼此的具体任务，一定程度上提高了合作效率。

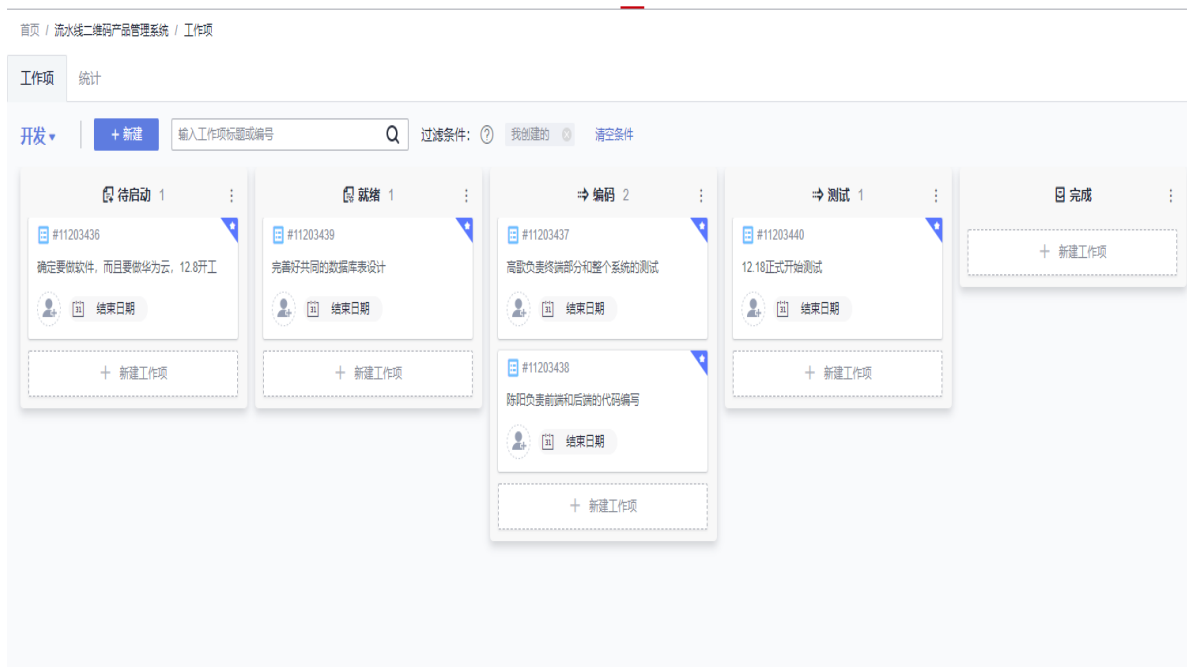


图 8-1 工作项功能使用

8.2 代码托管和代码检查功能使用

通过 DevCloud 平台的代码托管和代码检查功能，我们两名小组成员可以直接提交、推送和拉取、合并代码，能够非常方便地进行代码修改和保存，极大程度上提升了开发效率，代码检查功能更是有力地减轻了成员的开发负担。

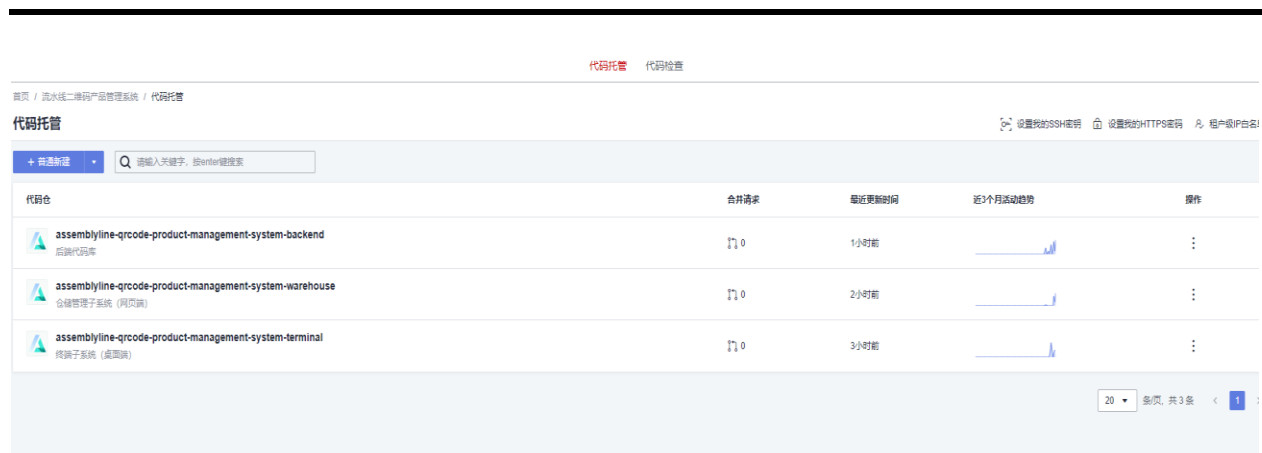


图 8-2 代码托管功能使用

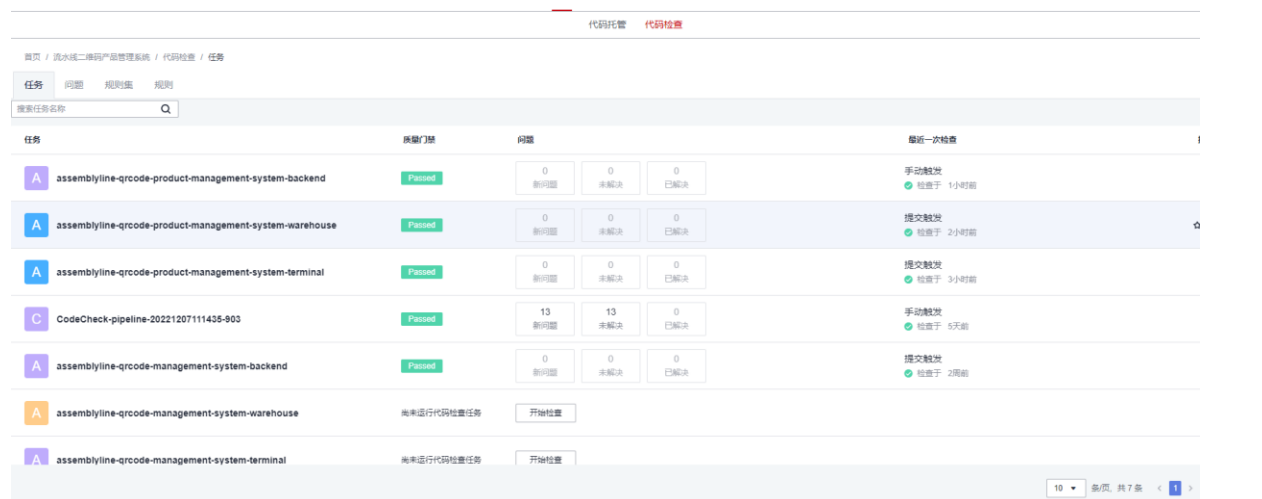


图 8-3 代码检查功能使用

8.3 流水线（自动化构建+自动化部署）

我们使用了 DevCloud 平台的流水线功能，此功能能够将我们推送到仓库中的代码进行自动化构建和自动化部署，直接挂载到服务器上。我们的系统实现采用了前后端分离的架构，流水线极大程度上减轻了沟通成本和负担。

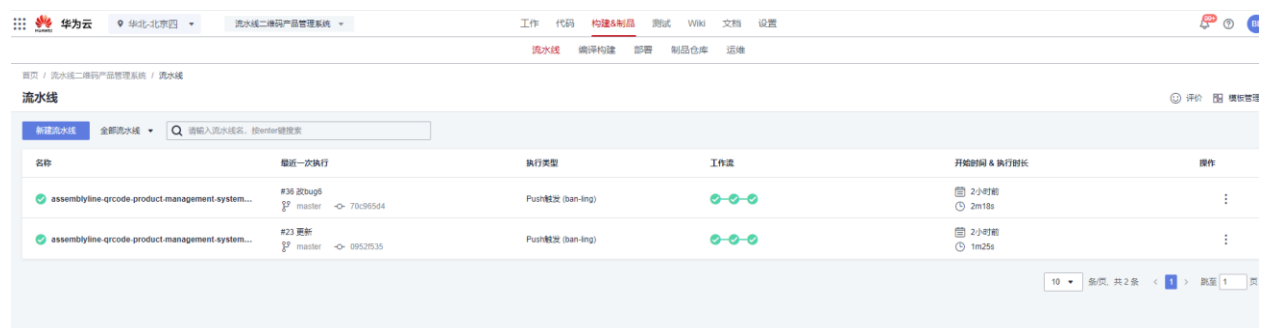


图 8-4 流水线

8.4 自动化测试

我们使用了华为云的自动化测试功能，此功能能够将我们提交到华为云上的代码进行自动化测试，分离了开发与测试的环节，有助于划分职责和提高效率，提升了团队协作的效率，也节省了手动运行测试脚本的时间。

水镜二维码产品管理系统 / 测试 / 测试执行

库

接口自动化性能自动化

输入关键字高级过滤

新建测试套件

名称	编号	状态	模块	缺陷	结果	处理器	进度	执行时间	执行次数	执行者	版本号	操作
整体功能测试	002	完成	-	-	成功	snwmyjt		2022/12/18 21:0...	1	ban-ling	-	

9 UML 工具小结

9.1 PROCESSION 思维导图

一款相当良心的国产在线工具，支持绘制 UML 图，功能也比较齐全。比较适合在非重度情况下使用。

由于是在线工具，提供了较多模板，也可克隆他人做好的模板，很多时候比较省心。缺点是免费使用对画图数量有限制，而且毕竟不是专业的 UML 绘图工具，有些图还是不能很好地实现。

9.2 DRAW.IO

一款完全开源的图表绘制工具，支持网页端直接使用和客户端安装使用（网页端可保存至 Google Drive 或 OneDrive 等云存储空间，两端均可保存到本地并导入）。是一款很轻量级的应用，但五脏俱全，功能很齐全。会用的话能够遇到的大部分图都能画。

缺点是软件本地化做得比较一般，很缺少中文资料，即使英文资料也不是很多。此外画图手感稍差，要把图画得好看不如 ProcessOn 容易。同时 Draw.io 也并非专精于 UML 绘图，因此专业性稍差。

本文档中几乎所有 UML 图都由 Draw.io 绘制。

9.3 ENTERPRISE ARCHITECT

专业 UML 绘图工具，比起 Rose 更轻巧更现代化。缺点是本身收费且不支持 Linux 平台。另外个人感觉还是有点笨重了。

9.4 PLANTUML

通过写代码的方式生成 UML 图，无需任何拖拽，自动排版。使用体验很好，但仅适用于对图细节要求不高的情况，如果需要微调图中元素位置，会比那些采用拖拽方案的工具更折腾。此外 PlantUML 画出来的图美观度也比较一般，如需微调配色会很困难。