**Packing Pizza**

**Use Case**

**修订历史**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 章节号 | 章节名称 | 变更原因 | 变更内容描述 | 变更日期 | 版本 |
| All | All | 新建 | 初版 | 2011-10-4 | 0.01a |
| 2.4 | Packing Pizza Label | 补充 | 增加打印参数说明 | 2012-1-6 | 0.03a |
| All | All | 修改 | 根据新表结构修改：  Pizza\_part.Value ->PartSn | 2012-2-15 | 0.03a |
| 2.1.5 | 业务规则 | 补充 | 明确本站是否允许没有待检料的情况发生 | 2010-2-27 | 0.03a |
| 2.1.5 | 业务规则 | 新需求 – Mantis #0000947 | 检料的时候，进行Check Battery | 2012-6-1 | 0.03a |

**目录**

[1. 前言 4](#_Toc305506575)

[1.1. Introduce 4](#_Toc305506576)

[1.2. References 4](#_Toc305506577)

[2. Use Cases 5](#_Toc305506578)

[2.1. UC-Packing Pizza 5](#_Toc305506579)

[2.1.1. 功能及目标 5](#_Toc305506580)

[2.1.2. 前置条件 5](#_Toc305506581)

[2.1.3. 后置条件 5](#_Toc305506582)

[2.1.4. 过程描述 5](#_Toc305506583)

[2.1.5. 业务规则 6](#_Toc305506584)

[2.2. UC-Packing Pizza Label Reprint 7](#_Toc305506585)

[2.2.1. 功能及目标 7](#_Toc305506586)

[2.2.2. 前置条件 8](#_Toc305506587)

[2.2.3. 后置条件 8](#_Toc305506588)

[2.2.4. 过程描述 8](#_Toc305506589)

[2.2.5. 业务规则 8](#_Toc305506590)

[2.3. UC-Packing Pizza UnPack 8](#_Toc305506591)

[2.3.1. 功能及目标 8](#_Toc305506592)

[2.3.2. 前置条件 9](#_Toc305506593)

[2.3.3. 后置条件 9](#_Toc305506594)

[2.3.4. 过程描述 9](#_Toc305506595)

[2.3.5. 业务规则 9](#_Toc305506596)

[2.4. UC-Packing Pizza Label 10](#_Toc305506597)

[Appendix 11](#_Toc305506598)

[Question 11](#_Toc305506599)

# 前言

## Introduce

本文档用于定义[Packing Pizza] 部分的业务需求，作为规格设计与程序设计的依据；读者为iMES 2012项目的用户，设计人员，开发人员和质检人员。

## References

# Use Cases

## UC-Packing Pizza

### 功能及目标

### 前置条件

### 后置条件

### 过程描述

|  |  |
| --- | --- |
| **UI** | **System** |
| 1.刷入Model |  |
|  | 2.check model  异常情况：   1. 若刷入的不是model时，则提示错误”请先刷入Model.”，焦点置于Data Entry输入框 |
|  | 3..展BOM，得到对应的BOM填入UI的table中  界面的Qty是需要刷入的part总数  异常情况：   1. 若刷入model不能得到BOM时，则提示错误”该model不能得到对应的BOM”，焦点置于Data Entry输入框 |
| 4.刷入part |  |
|  | 5.Part Match and check |
|  | 6.全部刷入后产生Kit ID，并保存 |
|  | 7.打印Packing Pizza Label  新系统用模板方式打印 |
| 8.刷7777 | 界面恢复到初始状态 |
|  |  |

### 业务规则

|  |  |
| --- | --- |
| **Function** | **Rule** |
| 本站是否允许没有待检料的情况发生 | 不允许 |
| 2A. 刷入的不是model | 在Model表里不能得到对应的记录 |
| 3.展BOM | 这里先描述现有的业务，后期有可能整合到GetBom中  1.根据Model展下阶得到BomNodeType=CN或C5的part。(Pn前5为是DUMMY取其part的PN属性值作为pn)  2.根据Model展3阶，得到第一阶是V2的part其下下阶(注意Qty需要相乘) [ V2一定会有VC下阶: V2->P1->KP->VC]  界面上显示Part No和Description是P1对应的Part；不同的P1作为多条记录显示在界面上，即相同的P1只显示一条，其VC合并，相同的P1对应的数量认为是一样的，数量只取第一条记录的Qty即可。  3.删掉PN前3位是MMI的Part ((GYJ）前三位只是代号，可能有DIB,MMI等)  ~~在展bom中下面这段是否还需要，即在C5,CN,V2下还会存在BM吗？(GYJ)不会~~  ~~declare @bm varchar(20),@mpnop1 varchar(20),@spnobm varchar(20)~~  ~~select @bm=MPno from #tt2 a,BomParts b (nolock) where b.Pno=MPno and b.Tp='BM'~~  ~~if @bm is not null and @bm<>''~~  ~~begin~~  ~~select @spnobm=left(SPno,6) from #tt2 where MPno=@bm~~  ~~select @mpnop1=MPno from Bom a(nolock),BomParts b (nolock) where a.SPno=@spnobm and MPno=Pno and b.Tp='P1'~~  ~~update #tt2 set MPno=@mpnop1 where MPno=@bm~~  ~~end~~  ~~update #t set SPno=@mpnop1 where SPno=@bm~~ |
| 5.Part Match and check | 先根据现有业务总结如下：   |  |  |  | | --- | --- | --- | | Type | Remark | match and check rule | | V2 |  | 14码，前5位等于VC 没有与其他kit ID绑定  如果用户录入的数据第一位为6，并且 14位长，则可以认定用户刷入的是Battery Part S/N，需要进行Battery Check:  Battery Part S/N 的前5位 = OlymBattery.HPPN 为条件查询OlymBattery 表，如果没有记录，或者查询到的记录的HSSN 字段值为'', 则报告错误：“请联系IE Maintain ”+ LEFT(@BatteryPartSN, 5) + ' 的HSTNN 号码!' - @BatteryPartSN 为Battery Part S/N | | CN |  | 12码前2位是CN，或12码前2位是7E 没有与其他Kit绑定，不存在于Pizza\_Part.PartSn  在Product.CUSTSN中存在  且  Product.Model等于CN阶下阶的pn  现在的逻辑：  在SnoDet中不存在Tp='KIT' and Sno=@sn 在SnoDet中存在Tp='CPQSNO' 且"select c.SPno from Bom a (NOLOCK),BomParts b (nolock), Bom c (nolock) where a.SPno=b.Pno and b.Tp='CN'and a.MPno='"&trim(lblModel.value)&"' and a.SPno=c.MPno and c.SPno='"&Pno&"'" | | C5 |  | 输入的code等于PN | |
| 6.产生Kit ID | Kit ID格式：  P Y S M XXXXX --9位  P：固定字符  Y：一位年  S：SiteCode，SysSetting.Value where Name=’SITECODE’  M：一位月，123456789ABC  XXXXX：包括0123456789的sequence number，从00001开始 |
| 6.保存 | Insert Pizza  Insert PizzaStatus: Station=’MP’,Line=’’  Insert Pizza\_Part  原先的逻辑：  insert KitMas values (@id,@model,'KIT','','MP',@user,@dt,@dt)  insert KitDet values(@id,'KIT',@kp,@user,@dt,@dt) |
|  |  |
|  |  |
|  |  |
|  |  |

## UC-Packing Pizza Label Reprint

### 功能及目标

重印Packing Pizza Label

### 前置条件

### 后置条件

### 过程描述

|  |  |
| --- | --- |
| **UI** | **System** |
| 1.刷入Kit ID |  |
| 2.输入Reason |  |
| 3.点击Reprint Button |  |
|  | 4.检查是否可以重印  异常情况：   1. 若输入的Kit ID没有找到数据时，则提示错误信息” 此KIT ID不存在！” |
|  | 5.记录打印重印log |
|  | 6.打印Packing Pizza Label |
|  |  |

### 业务规则

|  |  |
| --- | --- |
| **Function** | **Rule** |
| 4A. 输入的KIT ID没有找到数据 | Pizza中不存在 |
|  |  |
|  |  |

## UC-Packing Pizza UnPack

### 功能及目标

解绑

### 前置条件

### 后置条件

### 过程描述

|  |  |
| --- | --- |
| **UI** | **System** |
| 1.刷Kit Id或IEC CT或Vendor CT |  |
| 2.输入Reason |  |
| 3.点击UnPack Button | 弹出对话框，要求User做Confirm: “是否解掉此PIZZA?”  选择是才进行后续处理 |
|  | 4.检查是否可以解绑  异常情况：   1. 若输入的code包含”-”字符时，则提示错误信息” 6-3碼不能唯一確認PIZZA，請刷IEC CT或Vendor CT或Pizza ID” 2. 若不能找到记录时，则提示错误信息”Wrong Code.” 3. 若能找到多条记录时，则提示错误信息” 此號碼不唯一，不能唯一確認PIZZA，請刷IEC CT或Vendor CT或Pizza ID” 4. 若已与机器绑定时，则提示错误信息” 此Pizza已經與機器結合，不能unpack” |
|  | 5.删除绑定数据~~，保存unpack log?~~ |
|  |  |
|  |  |

### 业务规则

|  |  |
| --- | --- |
| **Function** | **Rule** |
| 4B. 不能找到记录 | 在Pizza\_Part按照PizzaID和PartSn都不能找到记录 |
| 4C. 能找到多条记录 | Pizza\_Part按照Value能找到多条记录  Pizza\_Part.PartSn=刷入的code |
| 4D. 已与机器绑定 | 存在ProductInfo.InfoType=’KIT2’ and InfoValue=@kitid 或 Product.PizzaID =@kitid |
| 5.删除绑定数据~~，保存unpack log~~ | 删除以下表的相关记录：  Pizza  PizzaStatus  Pizza\_Part |
|  |  |

## UC-Packing Pizza Label

打印时传递的参数是Kit ID

# Appendix

## Question