

USE CASE: TRYTON FOR WAREHOUSE MANAGEMENT

SALUC & PRECIBALL



- Billiard and Precision Balls manufacture.
- Belgian company

OUR CONTEXT

- Developing in Tryton since 2016 exclusively for our company
- Migrate our legacy ERP to Tryton.
- Our transition needs a lot of customisation and new functionalities in Tryton.
- First step of our migration concerns PRECIBALL and precision balls.

PRECIBALL'S WAREHOUSE

3 areas of storage.



PRECIBALL'S WAREHOUSE

A lot of locations to manage : +/- 2000.



PRECIBALL'S WAREHOUSE

Detail of location :

Codification : SEN1-A-01-1

- SEN1 = code of area
- A = Row identification
- 01 = Column identification
- 1 = Floor number

Content :

- Multi Product
- Multi Lot



THE TRYTON PART...

WAREHOUSE STRUCTURE CREATION

- Use of option “Flat Children” of Tryton for optimisation.
- Creation of labels with ReportLab open-source PDF Toolkit.

CREATION OF LOCATIONS WITH PROTEUS.

```
Location = Model.get('stock.location')
entrepot_preciball, = Location.find([('type', '=', 'storage'),
                                     ('code', '=', 'SEN1')])

allee = ['A', 'B', 'C', 'D', 'E']
locations = []

for a in allee:
    for r in range(1, 34):
        for n in range(0, 6):
            location = Location()
            location.code = 'SEN1-%s-%02d-%d' % (a,r,n)
            location.name = location.code
            location.type = 'storage'
            location.parent = entrepot_preciball
            locations.append(location)

Location.save(locations)
```

LOCATIONS : ISSUES AND SOLUTIONS

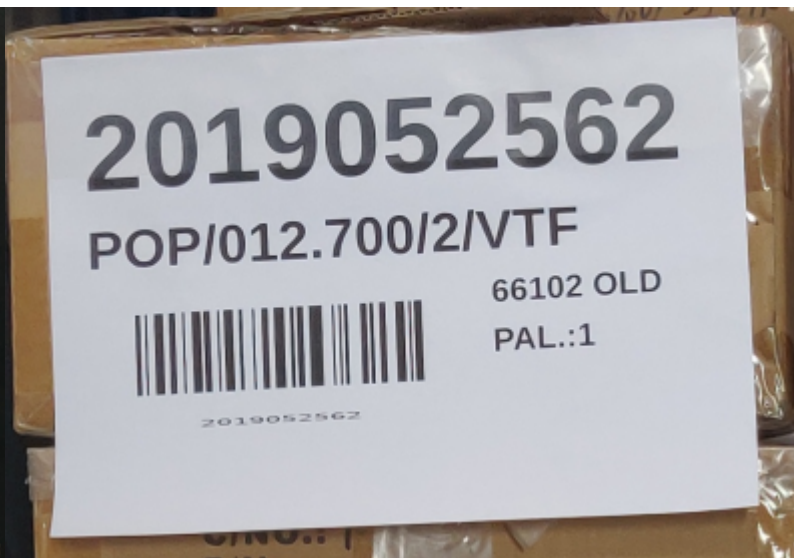
- Location display and processing based on Location's name alphabetical order.
 - to be careful with number in the name : 01 and 1 or 10
 - the assign_try browse the location by alphabetical order.
- Location Code is not mandatory
- Traduction du nom et donc modification de l'ordre !!!
Code is not use in the processes

LOT MANAGEMENT

- Automatic sequence
- Metadata on the lot (supplier tracking id, pallet id, packaging, ...)
- View Product/Lot/Quantity
 - https://bitbucket.org/zikzakmedia/trytond-product_quantity_by_location/src/default/
 - adding lot number on this view

PRINTING LOT LABEL

- Relatorio Model → A4 Paper
- Need to print small labels through a ZEBRA label printer



NEW REPORT ENGINE : RENDER_TEMPLATE WITH JINJA2

```
@classmethod
def get_environment(cls):
    """
    Create and return a jinja environment to render templates
    Downstream modules can override this method to easily make c
    to environment
    """
    env = Environment(
        autoescape=True,
        trim_blocks=False,
        extensions=['jinja2.ext.loopcontrols'],
    )
    # env.filters.update(cls.get_jinja_filters())
    return env

@classmethod
def render_template(cls, template, localcontext, translator):
    """
    Render the template using Jinja2
    """
    env = cls.get_environment()
```

```
report_template = env.from_string(template.decode('utf-8'))  
return report_template.render(**localcontext)
```

MODIFY EXECUTE OF REPORT

```
@classmethod
def execute(cls, ids, data):
    _, content, _, _ = super().execute(ids, data)
    cls.print_xml(content)
    return None

def print_xml(xml_document, timeout=10):
    """
    Send XML formatted text to a network label printer
    :param xml_document: Document object, fully build for label.
    :param timeout: Socket timeout for printer connection, default 10
    """

    printer_hostname = Transaction().context.get('printer_hostname')
    if not printer_hostname:
        printer_hostname = ZEBRA_HOST

    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.settimeout(timeout)
        s.connect((printer_hostname, ZEBRA_PORT))
        s.send(bytes(xml_document, 'utf-8'))
```


USE OF NEW REPORT

```
<record model="ir.action.report" id="report_label_zebra">
    <field name="name">Zebra Label</field>
    <field name="model">stock.lot</field>
    <field name="report_name">stock.lot.zebra.xml</field>
    <field name="report">preciball_stock/zebra/stock_lot.xml
</record>
<record model="ir.action.keyword" id="report_label_stock_lot">
    <field name="keyword">form_print</field>
    <field name="model">stock.lot,-1</field>
    <field name="action" ref="report_label_zebra"/>
</record>
```

```
from trytond.modules.zebra_print.zebra import ZebraReport

class ZebraLabelStockLot(ZebraReport):
    'Zebra Label'
    __name__ = 'stock.lot.zebra.xml'

    @classmethod
    def execute(cls, ids, data):
        with Transaction().set_context(
            printer_hostname='imp_label_4.saluc.com'):
            return super().execute(ids, data)
```

TEMPLATE JINJA2 LABEL ZEBRA

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE labels SYSTEM "label.dtd">
<labels _FORMAT="STOCKLOT.ZPL" _QUANTITY="1" _PRINTERNAME="Printer 1"
  {% for lot in records %}
  <label>
    <variable name="no_lot">{{ lot.number }}</variable>
    <variable name="code_barre">{{ lot.number }}</variable>
    <variable name="code_article">{{ lot.product.code }}</variable>
    <variable name="bcnum">{{ lot.id_preciball_po }}</variable>
    <variable name="nopalette">PAL.: {{ lot.id_supplier }}</variable>
  </label>
  {% endfor %}
</labels>
```

MOBILE APP ANDROID

- Test with vue.js → too difficult to realize an ergonomic app
- Switch to react native to use Android native components
- Difficulties : javascript stack and REACT development: nodejs, packages dependencies, fast framework evolution, asynchronous call ...
- Use of user app from Tryton (routes, authentication with key,...)

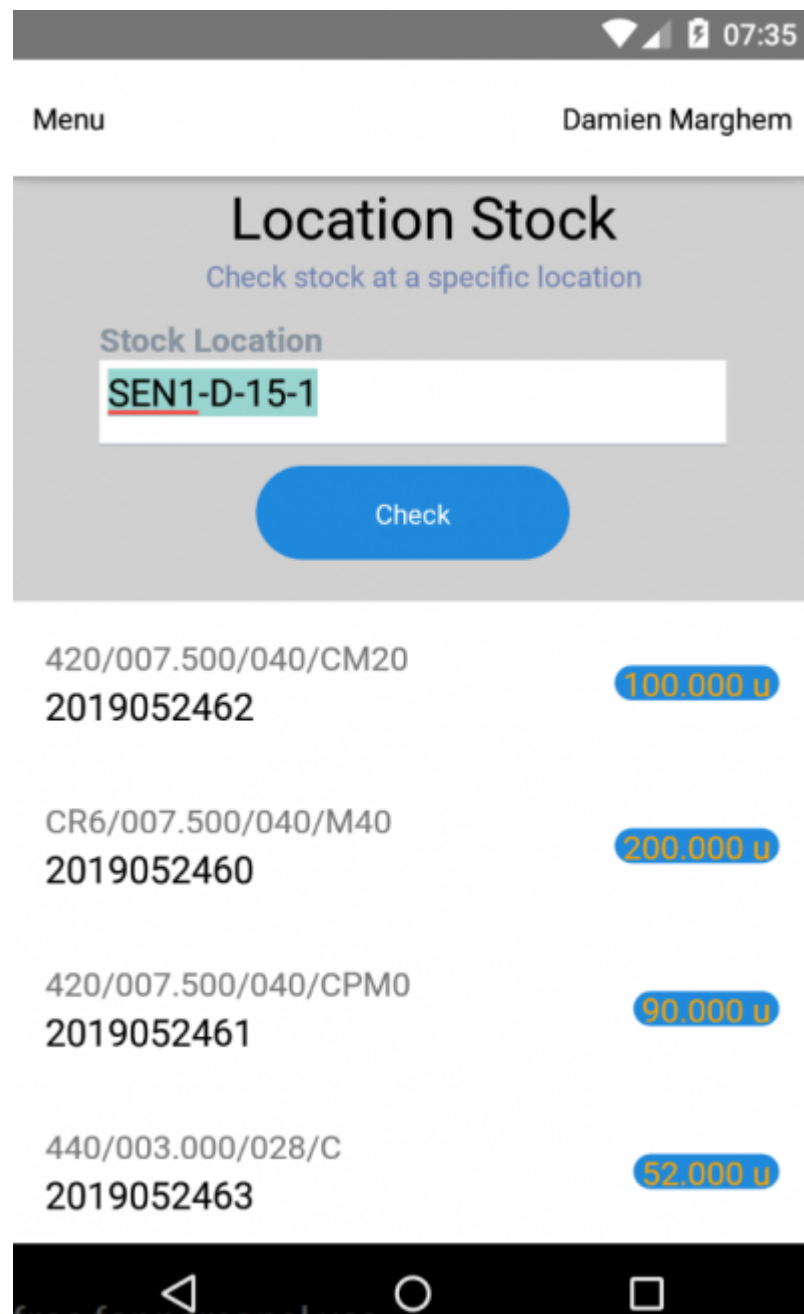
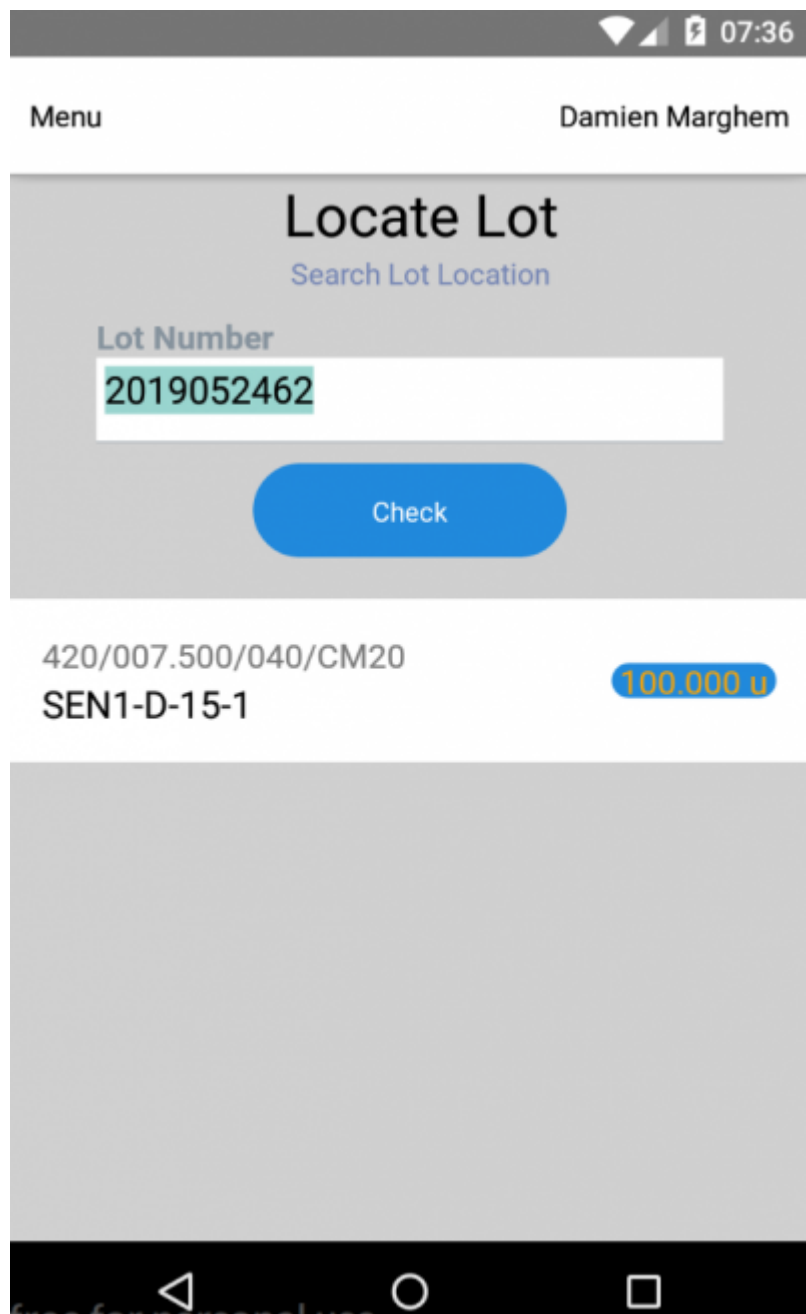
HARDWARE

Use of ZEBRA TC52 android 8.0 touch computer with
barcode scanner/imager



MOBILE APP FUNCTIONALITIES

- Define storage location
- Check quantities on a location
- Moves between locations
- Stock locations



MOBILE APP FUNCTIONALITIES

07:15

MenuDamien Marghem

Déplacement de Stock Interne

Déplacement d'un lot d'un emplacement vers un autre

Emplacement de Stock Départ

SEN1-B-27-5

Numéro de Lot

2018090446

Article

VER/003.000/PM300

Quantité

55 kg

Emplacement de Stock Arrivée

Emplacement de Stock Arrivée

Validation

07:37

MenuDamien Marghem

Stock Entry

Define where lot will be stored

Lot Number

2019052462

Article

420/007.500/040/CM20

Stock Location

SEN1-D-15-2

Validate

STOCK MOVE AND LOT

- Issues and solutions → CLOSED
 - Assignment
- Issues and solutions → OPEN
 - Use of a query for a view Product/Lot/Quantity
 - Entry moves can only be done when supplier shipment is globally finished

QUESTIONS ?

maxime.richez@saluc.com

thierry.bruyere@saluc.com