Technische Hochschule
Brandenburg
University of
Applied Sciences

Implementation of processes
- Checkout system -

Authors:
Thuy Phuong Nguyen, Stefan Balluneit, Hanna Schönrock

Examiner: Prof. Dr. rer. nat. Vera G. Meister

Brandenburg/H., den 07. Juli 2019

Content

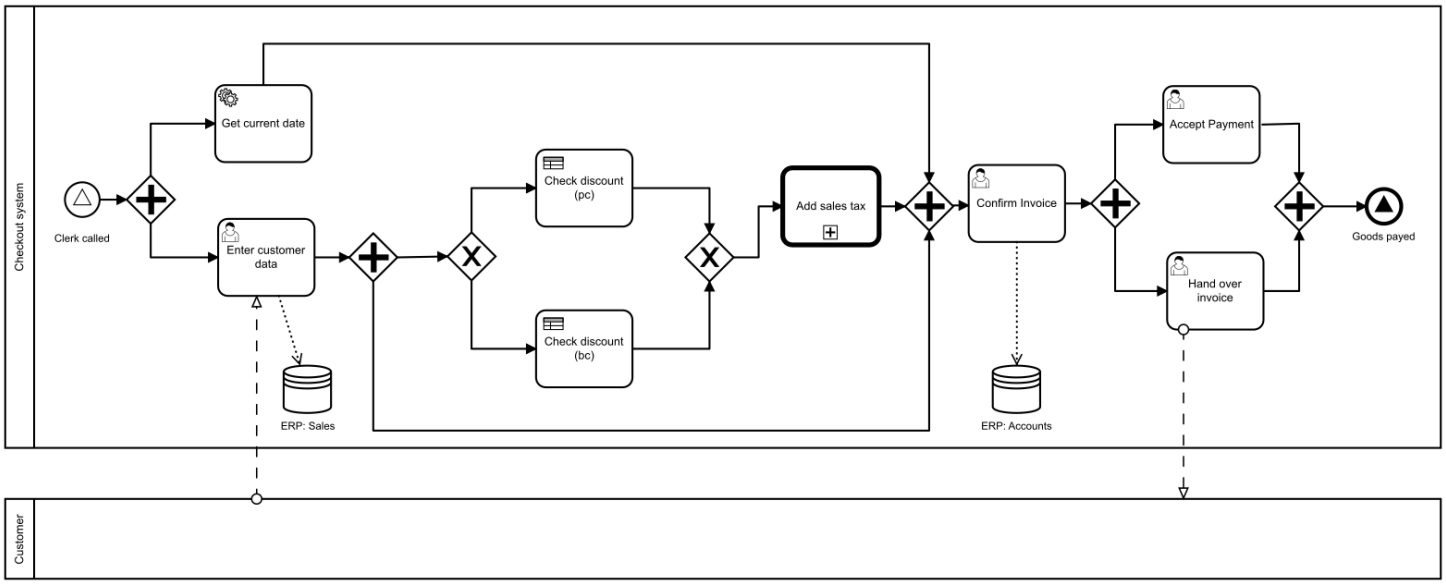
1 Arrangement of the topic

The main topic of our project is implementing Checkout system in the field for B2B. Because the company can have different discounts for each VIP customers, we decided to create a simple system that can simplify the counting for cashier in the future.

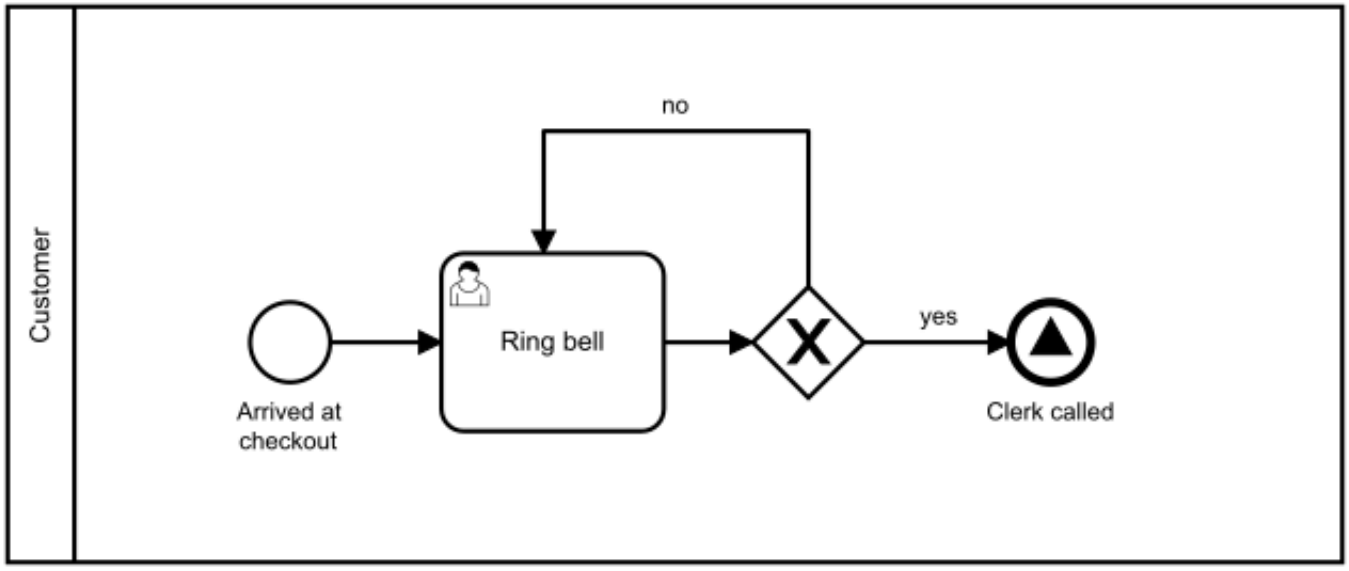
At first, we will describe how does the process work, which elements we used, how we implemented the system and, in the end, we will write down the vulnerabilities of this system.

2 Delimitation and Deskription

2.1 BPMN description



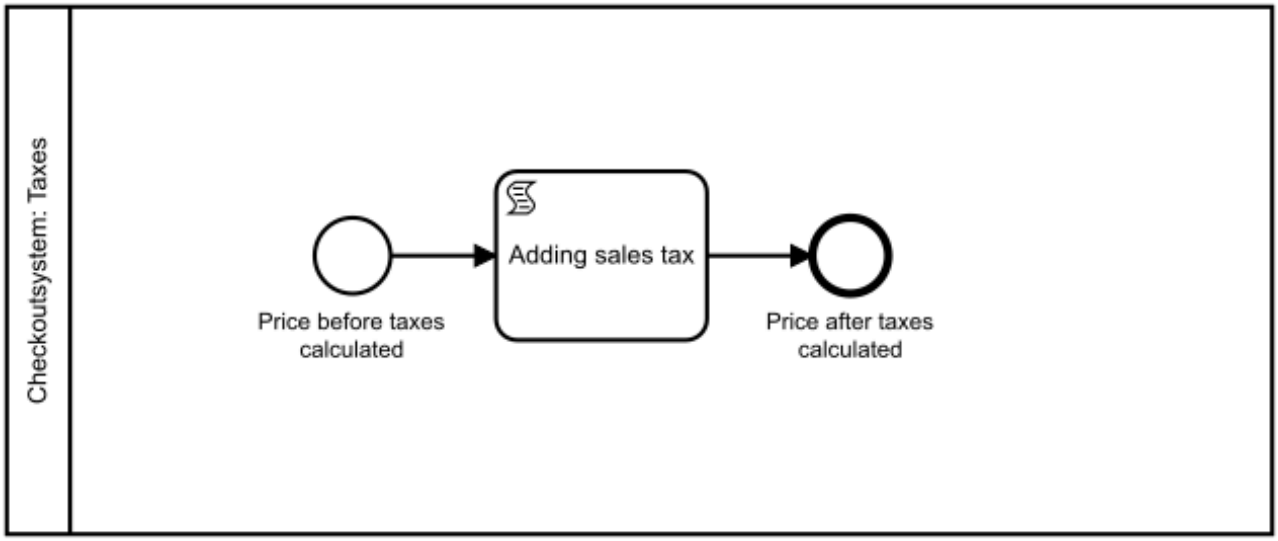
The process starts with a signal event: Clerk called. When a customer decides to pay, the person will have to ring a bell in order to checkout. This event will repeat until the seller comes. Signal event is connected to our checkout bell.



With User Task - seller must manually get and put information about the customer into the system. Filled information will be stored in the company's ERP Sales system. At the same time the system will try to get current date automatically with an API, with Service Task.

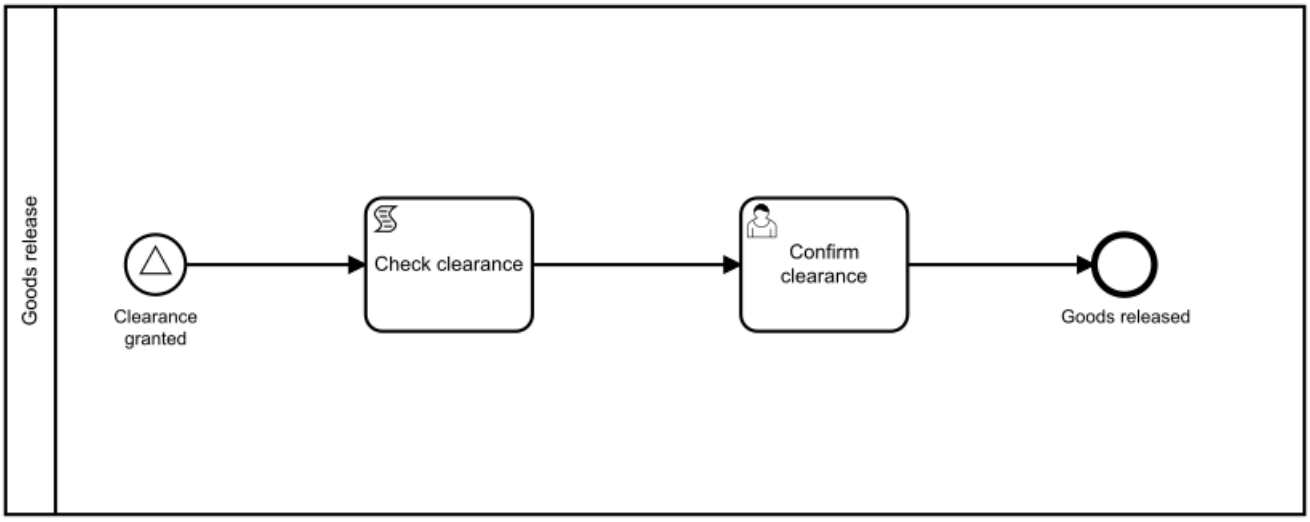
After getting information from company's database, the system will have to decide, whether she/he is private customer or is the person business customer. For both segments are different discount measures. That is why we created two Business Rule Task in which we implemented DMN. We will talk about DMN later.

Next step is adding sales tax with corresponding discounts. In this part we created Collapsed Sub-Process called "Add sales tax". This sub-process is connected to our Checkout system: Taxes. We used Script Task. Our idea is to multiply the discount price with 19 % taxes.

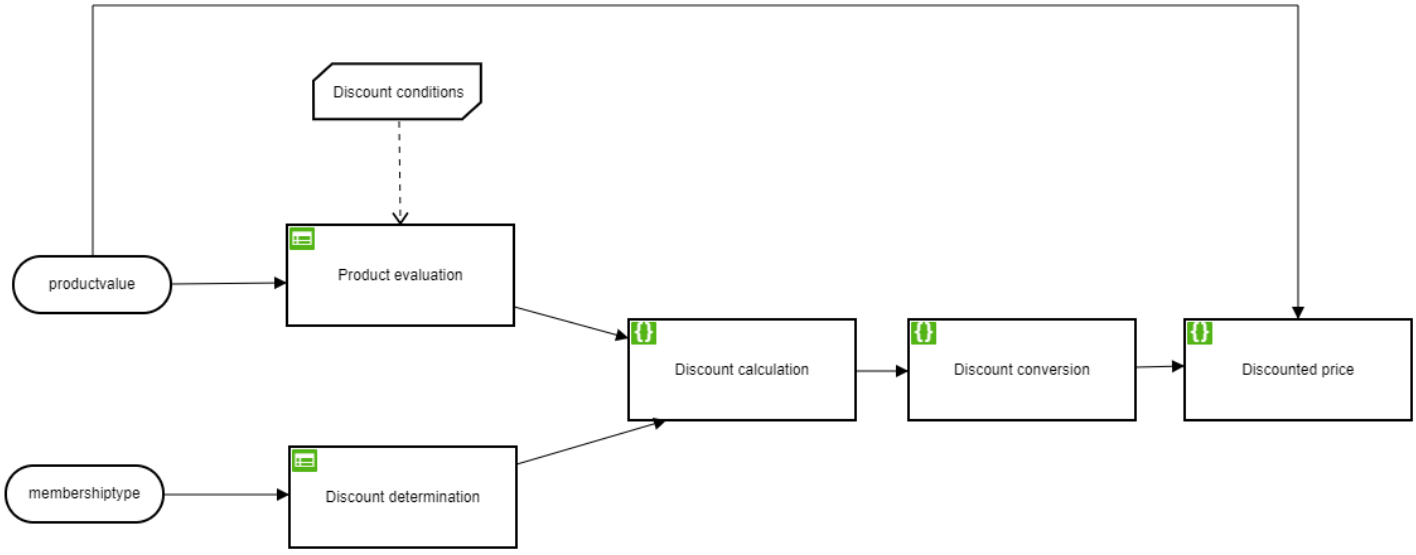


When everything is counted and the invoice is being created, the seller still needs to confirm it – User Task. He/she must control all information. Seller accepts payment, hand the invoice over to the customer and the goods are paid.

When the goods are paid then people in warehouse will get a signal: Clearance granted. They confirm it and ship the products right after confirming.



2.2 DMN description



In this project we created two Decision tables for each two main segments of customers – business customer and private customer. As we mentioned before, each of them has different type of discounts.

Discount depends on product value and membership type of the customer. The company has 3 membership types of private/business customer – none, silver, gold.

Product evaluation

View DRD

dqc1

C+	Input +	Output +	Annotation
	Value of shopped product	Discount qualification	
	double	double	
1	>= 10.00	0.25	-
2	> 50.00	0.25	-
3	> 150.00	0.25	-
4	> 500.00	0.25	-
5	< 10.00	0.0	-
6	>= 0	0.0	-
7	< 0	0.0	-
+	-	-	-

Discount determination

View DRD

dd1

U	Input +	Output +	Annotation
	Membership type	Percentual discount	
	string	double	
1	"none"	0.0	-
2	"silver"	0.05	-
3	"gold"	0.15	-
4	not("none","silver","gold")	0.0	-
+	-	-	-

Both have the same DMN structure, the only difference is output Percentual Discount in Discount Determination.

Discount calculation is Literal Expression. This is where we multiply the Percentual Discount with Discount Qualify.

The last Discounted price consist of also multiplying actual product value with final discount.

3 Explanation of technical modeling decisions

3.1 BPMN

The first Signal Events are connected with Signal Name – *ring*. It will stop after seller arrives.

When checkout process starts, the seller will have to fill out a form that consist of these Form Fields:

- Please enter the total value of all goods to be purchased – String type.
- Is it a private customer? – Boolean type.
- Which membership type does the customer hold (none, silver, gold)? – String type.

After getting information and finding discount price, the company still need to add sales taxes. This Collapsed Sub-process is connected to our Checkout system: Taxes. We used Script Task, more specifically JavaScript as Script Format. The Script looks like below:

```
var pricebt = execution.getVariable("pricebt");
var pricebt =parseFloat(pricebt);
var priceat = pricebt*1.19;
pricebt;
priceat;
```

Next step is confirmation of the user – User Task: Confirm Invoice. ID of these Form Fields should be identical to the first ones that the seller first put in. The seller needs to control:

- Current date
- This is NOT a business customer: - Boolean type of output from the first User Task
- The membership currently being held by the customer is: - String type, from the first User Task

- The value of goods to be purchased is: - String type, from the first User Task
- The discounted price before taxes is: - String type, from Business Rule Task
- The final price after taxes and to be payed is: - String type, from Add Sale Tax Sub-Process.

User Task: Accept Payment has also form. Questions are:

- Price payed? – Boolean type
- This price was payed: - String type

User Task: Hand invoice is to confirm:

- Handed over invoice? – Boolean type

3.2 DMN

The most important points to consider when creating a DMN decision tables are uniqueness and completeness. This means that all possible combinations must be used for all variables and that there is only one output value for each possibility, since the model we have created uses the Unique Hit Policy. The data formats of the variables correspond to those of the inputs. Dates are about membership type of customer and shopped value.

4 Reflection of vulnerabilities and options for improvements

Although the system worked well on our computer, there are still some weak points that we have not repaired yet. The DMN Decision tables are not the most elegant solution. If we tested more times, there would occur an error in some point.

In the actual state, all specified tasks were implemented technically. However, some tasks remain open. The BPMN model had to be revised several times and until the end, since it is the technical implementation to run again.

We used mainly Camunda Modeller for modelling BPMN and DMN processes. Then tested BPMN model in Camunda Cockpit and DMN model in DMN Simulators.

Because of our short knowledge about technical implementing of the process (specifically JavaScript know-how), we believe that if we had a professional in our team, everything would work smoothly.