Advice Report

Introduction

This is the final report for Onderwijs in Beeld to provide coherent and realistic advice for the automation of their invoicing processes. In this document, there will be the reasoning behind the decisions, based on the provided requirements and analysis.
The goal of this report is to give a clear starting point and direction to start your own projects and apply them as you please.

Considerations

Criteria
1) The design must include:
- An emailing service, for sending automated emails to customers as well as employees;
- An invoice managing service, for the upkeep of invoices as well as uploading them to MoneyBird;
- An invoice production service, which creates invoices and sends them to the invoice managing service.

2) The design must decrease the time spent on the invoicing process by at least 90%
3) The design should not take away from the customer experience
4) The design must be highly scalable, since the customer base will be growing in the coming years

Requirements
There need to be new diagrams visualizing the new processes for:
- Sending instruction emails to the customer;
- Creating an invoice for a customer;
- Uploading an invoice to MoneyBird;

These diagrams should include a description in which is explained what kind of data will be passed between services. The reason these processes need to be updated is because these are the processes that require manual input from the back office manager and will therefore sharply decrease the time spent on invoicing.

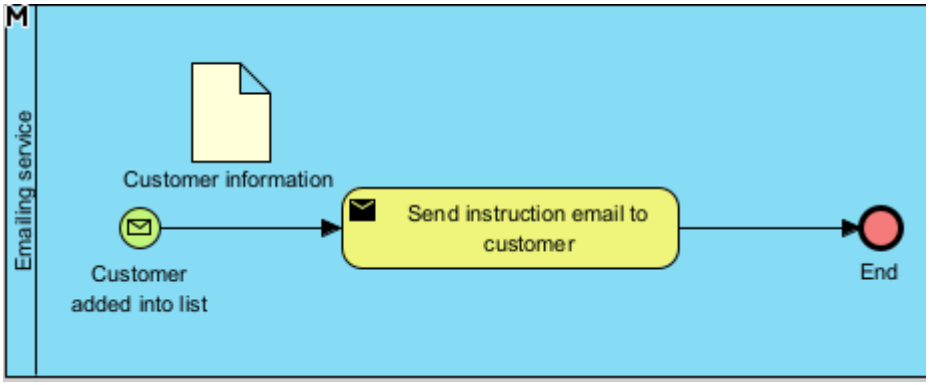
Because Tom said that most customers are sending emails that they want to get started instead of filling in the dedicated form on the website, it is important to work with a list of customers that is both automatically filled but can also be manually altered. This means that when customers use the form, they will be added into this list automatically, but when they send an email they will also be able to be put in the list.

Scope
The processes of the creation, sending and updating of invoices will be focused on. For new customers as well as existing customers.

The time spent on packaging and sending out the packages will not be taken into account in this design, because it is stated earlier that the packages will be out of scope. Therefore, any steps that include a reference to the packages will not be fleshed out in detail.

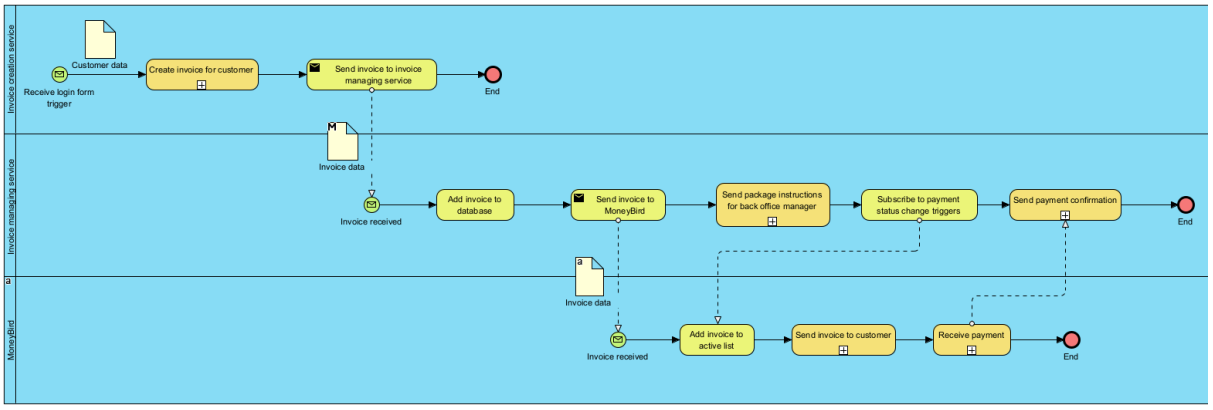
Advice

Sending instruction emails to the customer



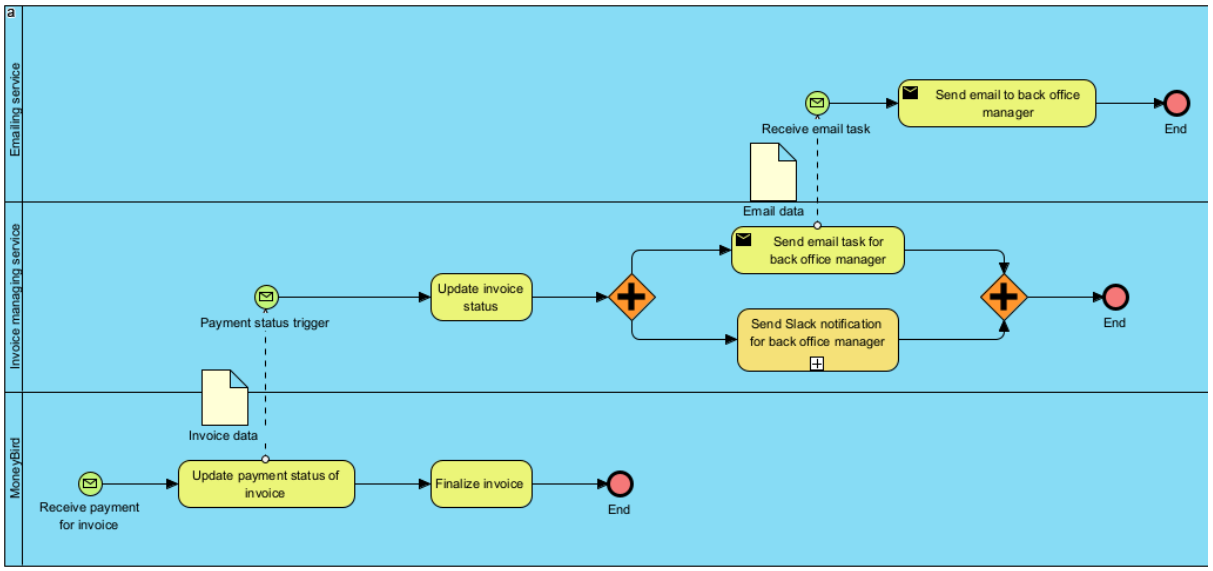
With an emailing service, the instruction email can be formed based on the customer information that is received from the trigger when a customer is added into the customer list. There is no need for the back office manager to send any instruction mails anymore.

Creating an invoice for a new customer / Uploading an invoice to MoneyBird



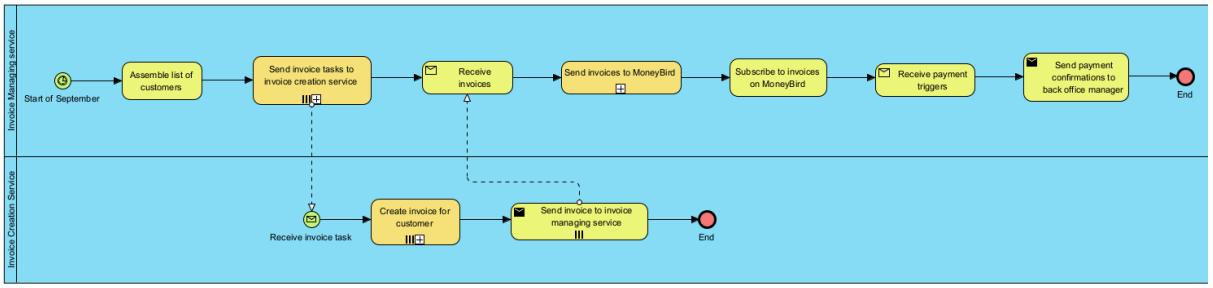
This process starts upon receiving a trigger that the login form has been submitted by the customer. It is debatable whether you want the invoice managing service or the invoice creation service to listen to these events. After the invoice is created in the invoice creation service, it is sent to the invoice managing service, and this is where the process ends for the creation service.
The invoice managing service starts by adding the invoice received into the database, possibly by a status indicating that it has not yet been sent out. Then it sends the invoice to MoneyBird. It makes sense to send the starter package instructions after sending the invoice to MoneyBird, because this will make sure that the payment process is pending before you send out any materials.
MoneyBird starts by adding the invoice into the active list on its system. What's important here is that there is a unique id from the invoice received from the invoice managing service, in order to track the payment status. Because after the invoice has been added into

MoneyBird's system, the invoice managing service will subscribe to any payment status change triggers in MoneyBird. MoneyBird now sends out the invoice to the customer and awaits payment completion. After MoneyBird has received the payment, the payment status will change and therefore the payment confirmation process will start, which is explained below.



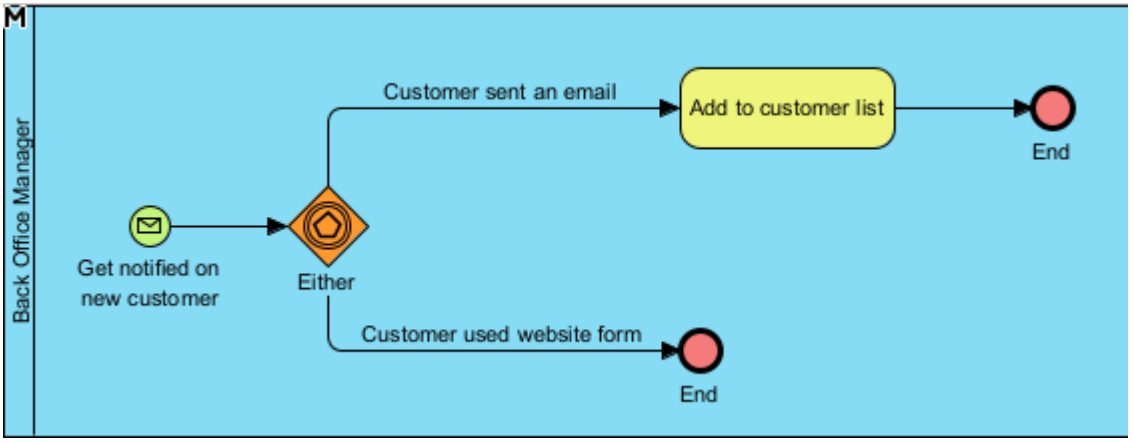
After the payment is received from the customer and the payment status has been changed, MoneyBird will finalize the invoice and end its process here. Now the invoice managing service receives the payment status change trigger and updates the invoice in its own database accordingly. As a suggestion, you can choose to both send an email and a Slack notification of the invoice completion, which is handled by the emailing service as well as the Slack service accordingly.

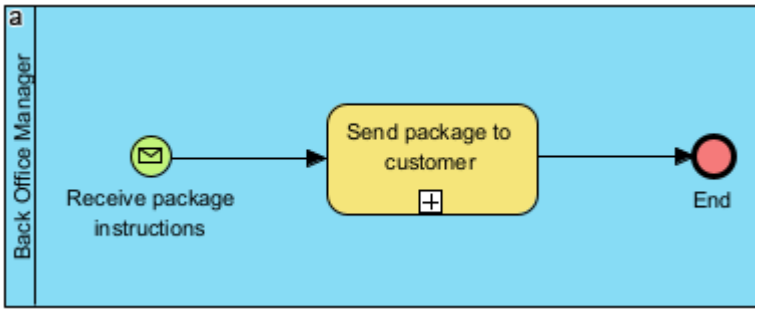
Yearly invoicing for existing customers



At the start of September each year, the invoice managing service will assemble its list of existing customers and send out an invoice task to the invoice creation service for each customer respectively. What's important here, is that there should be a unique id for each customer invoice, so that the invoice managing service can track that each invoice is received. These invoices will be sent to MoneyBird and should be tracked based on the unique ids. You can choose to either send a payment confirmation for each school, or you could choose to periodically send a notification on how many schools have paid and which ones still need to pay. This is a valid consideration since the customer base will be growing in the next few years and you might avoid spam for the back office manager this way.

Tasks for back office manager
Besides calling with a customer to help them with the login process, there should only be two tasks left for the invoicing process.
That is, adding the customer to the customer list, which should be the list that the invoice managing service (or invoice creation service) is listening to. But also sending out the starter package for the customer based on the instructions received. These instructions could be received via Slack or Email, and you could make this process the same for when someone orders a package from the webshop.





Costs

According to the requirements from Onderwijs in Beeld before starting this project, there were estimations on the time it would take on creating the services necessary to fulfill the desired situation. These estimations are:

- Emailing service: 2 two-weekly sprints
- Invoice managing service: 2 working days
- Invoice creation service: no estimation given

According to the responsibilities and complexity of the services according to the advisory processes, I would estimate the development of the services differently:

- Emailing service: ~3-5 working days, as the only objective for the emailing service is to send out an email based on the instructions given. Consistency and security might impose more difficulty, but applying the right amount of automated tests could make this feasible within this time frame.
- Invoice Managing service: ~2-3 sprints, because there are many interactions for the invoice managing service:

    - it sends out tasks to other services such as the invoice creation service, Slack service and emailing service;
    - it sends instructions to the back office manager;
    - it listens to triggers from the invoice creation service and from MoneyBird;
    - it has a yearly event to trigger a very large invoicing task;
    - it keeps the list of invoices up to date;

  This service has the most responsibilities and integrations and therefore requires a lot of testing before it can be implemented. You might consider splitting the invoice managing service into separate services, so that the invoicing responsibilities don't solely rely on the invoice managing service. The estimated time on this should be the same.
- Invoice creation service: ~2-3 working days. The creation of invoices is quite streamlined, but there might be some edge cases which could make this a more lengthy development process. The only responsibilities of this service are to create invoices and send them to the invoice managing service.

In total this would take 25 to 38 working days to develop. But these estimations might be high considering the experience from OIB in developing stable software, as well as the amount of employees that have changed over time.

The new invoicing workflow for the OIB manager would only require them to occasionally call a customer to help them with the login process and to add customers to the customer list manually.
This would be a yearly estimate of:

- (average amount of new customers yearly / 2) * time spent on adding customer to list = (372 / 2) * 2 = 372 minutes = >6.2 hours yearly;
- chance of having to call customer * 372 * time spent on calling = 0.1 * 372 * (10 to 30 minutes) = 372 to 1116 minutes = 6.2 to 18.6 hours yearly;

Total = 12.4 to 24.8 hours spent on invoicing annually.

Conclusion

Results
Considering the KPI's formula on achieving the goal of decreasing time spent on the invoicing process by 90%:
Tnw(days) = time spent annually on new customers
Tex(days) = time spent annually on existing customers
Tim(days) = time spent implementing the automation services
Tnp(days) = time spent by the new processes annually.
Invoice Automation Profit (days) = Tnw + Tex - Tim - Tnp

If we fill in the formula:
Minimum estimations: 12.13 + 0.97 - 25 - 0.51 = -12.41 days
Maximum estimations: 14.73 + 0.97 - 38 - 1.03 = -23.33 days

What this shows is that with these estimations and formulas, it is not achievable to reduce the time spent on the invoicing process by 90% within one year, since the time spent on the development takes more time than the time saved.
However, the development of the services is a one-time investment, and this will be negligible overtime. So if we alter the formula and ignore the time spent on implementing the automation services, the result would be:
Minimum estimation: 12.13 + 0.97 - 0.51 = 12.59 days, which is a 96% improvement.
Maximum estimation: 14.73 + 0.97 - 1.03 = 14.67 days, which is a 93.4% improvement.

Since the customer base will be growing a lot, the profit in time saved will only improve overtime. So it is highly recommendable to implement these new processes sooner than later, because it will result in more time saved.

Future implementations
Since this project only focused on saving time in the invoicing process, future projects might focus on the automation of sending packages.
Since customers currently mainly choose to send an email rather than fill in the form on the website, it might be a solution to improve in this process by making the application instructions more clear, to reduce time spent on replying to emails and adding customers to the customer list manually.