

Gestion de processus pour E-Commerce

Elaboré par:

Fatma REBEI

Maram ZRIBI

Maram ABAIDI

Sami MASMOUDI

Encadré par:

Mme.ZOUALI Meriem



Plan



- 01 Introduction
- 02 Contexte du Projet
- 03 Diagramme BPMN
- 04 Implémentation
- 05 Conclusion

Introduction



Le commerce en ligne nécessite une efficacité opérationnelle maximale pour maintenir la rapidité d'exécution et la satisfaction client.

Ce projet vise à automatiser le business process en utilisant Camunda Zeebe, réduisant ainsi les délais et les erreurs, et améliorant l'expérience utilisateur de manière significative.

3 Acteurs



Livreur



**Admin
(vendeur)**



Utilisateur

Diagramme BPMN

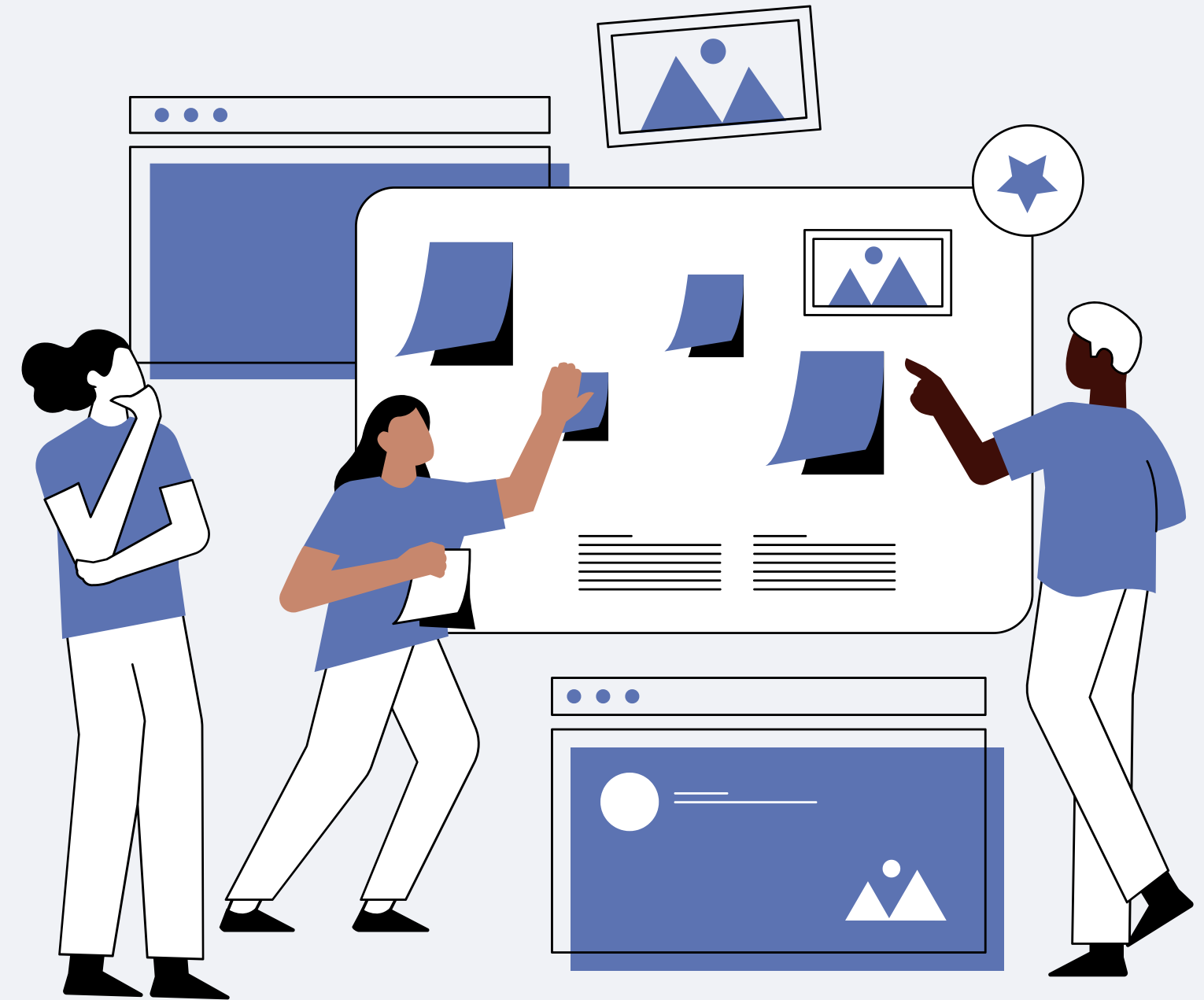


Diagramme BPMN:

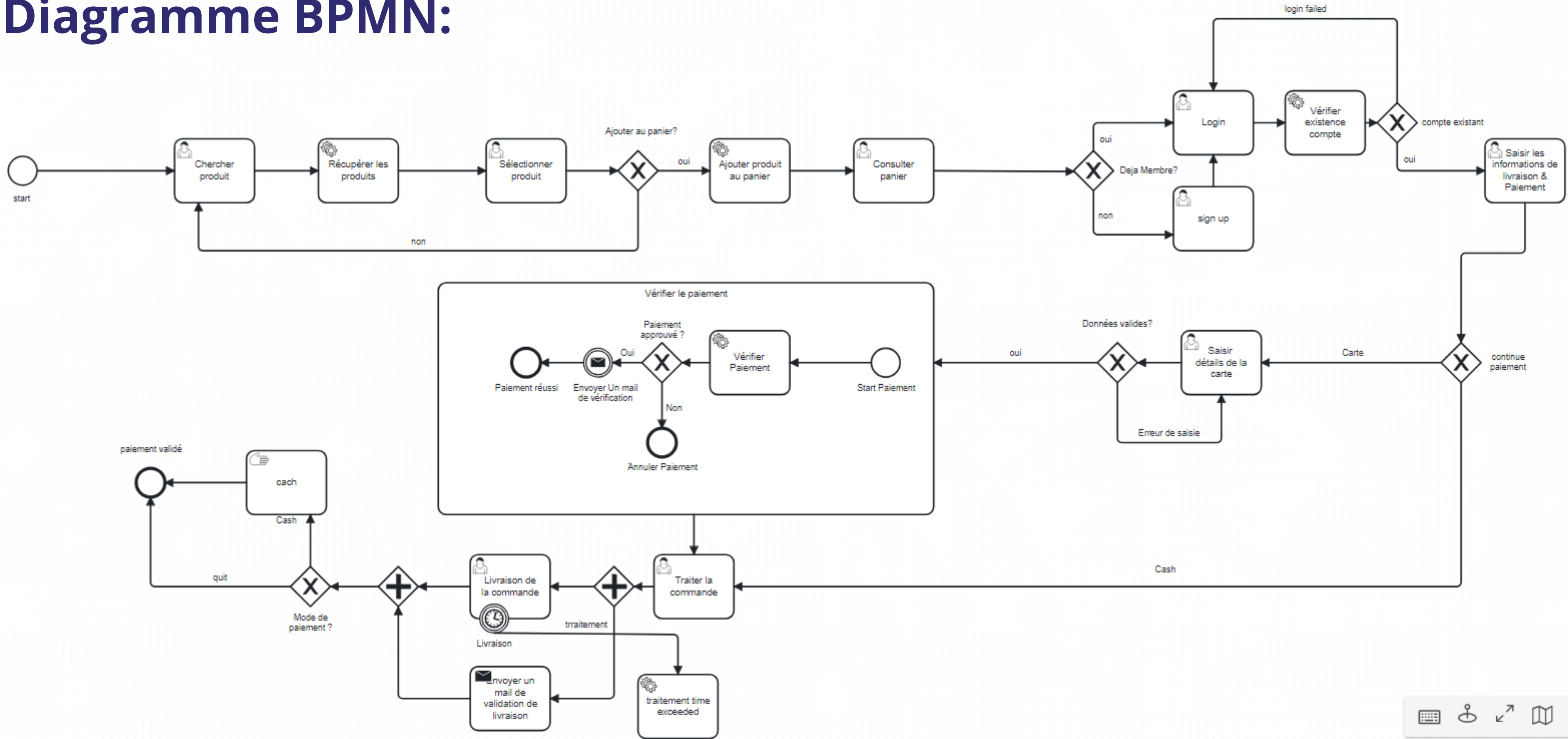


Diagramme BPMN:

-10 User Task

-6 Service Task

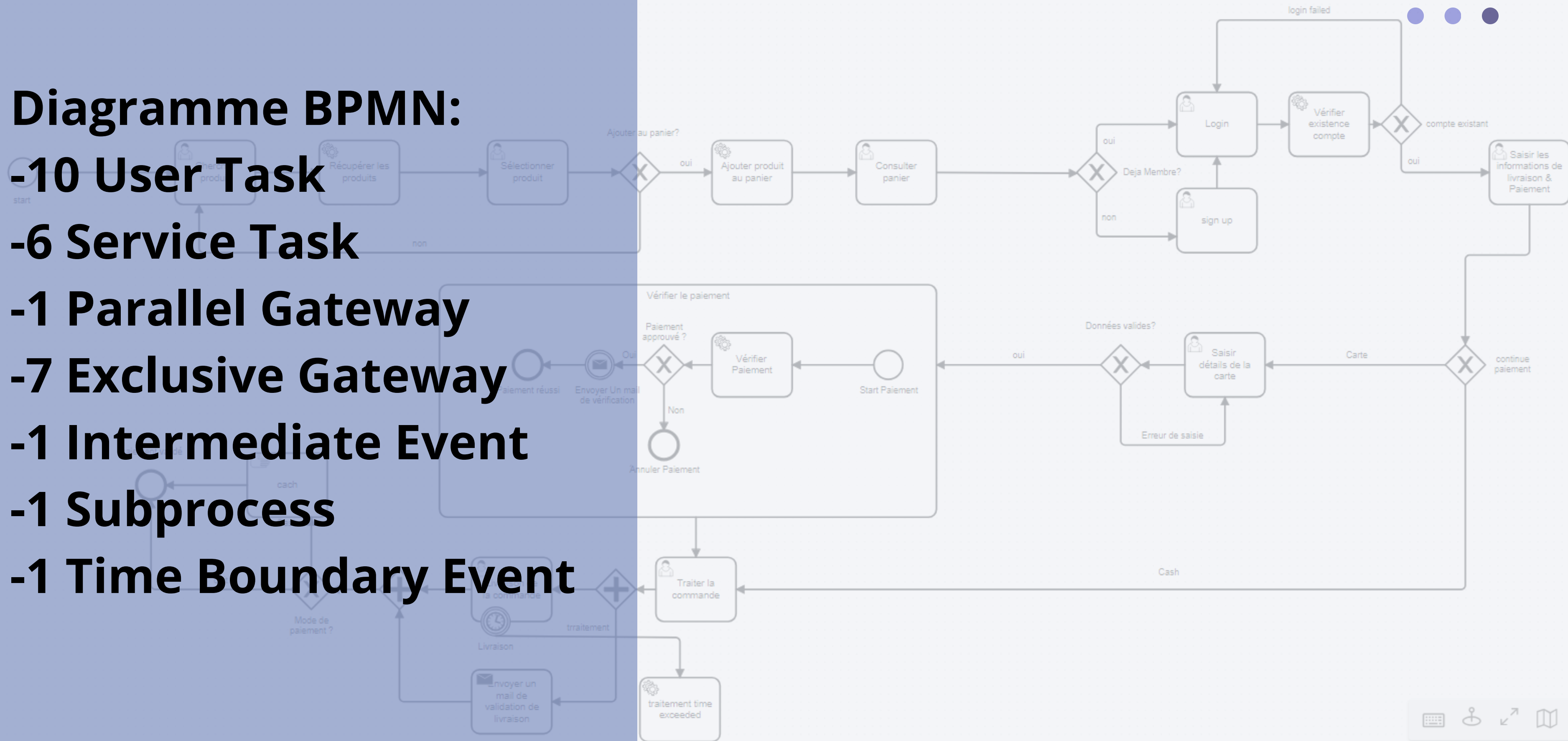
-1 Parallel Gateway

-7 Exclusive Gateway

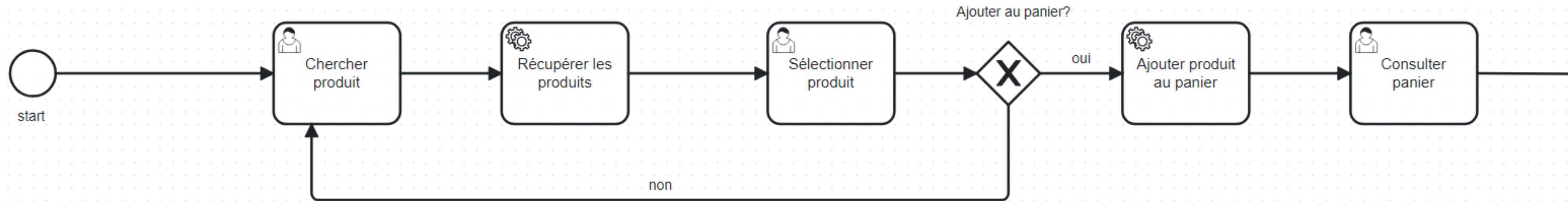
-1 Intermediate Event

-1 Subprocess

-1 Time Boundary Event






Sequence Navigation



Recherche Produit



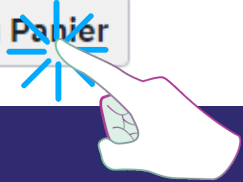
	Product Name Dada	In stock	<input type="text" value="1"/>	124,90 €
	Product Name Toto	In stock	<input type="text" value="1"/>	33,90 €
	Product Name Titl	In stock	<input type="text" value="1"/>	70,00 €

Laptop ASUS
Caracteristiques :
Stockage : 1TB HDD
RAM : 16Go
Ports : 3 X Usb3.2
Prix : 500

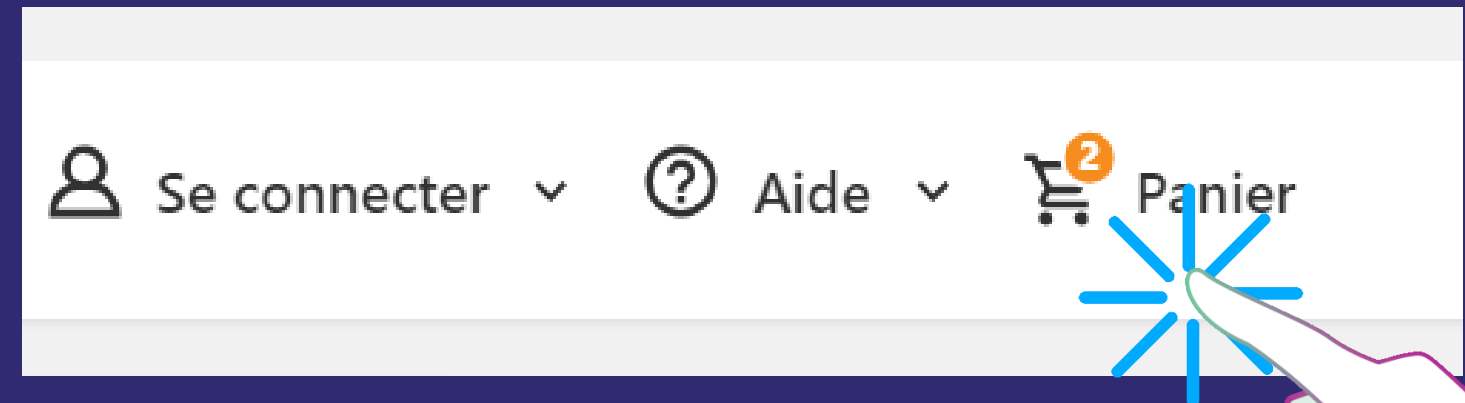
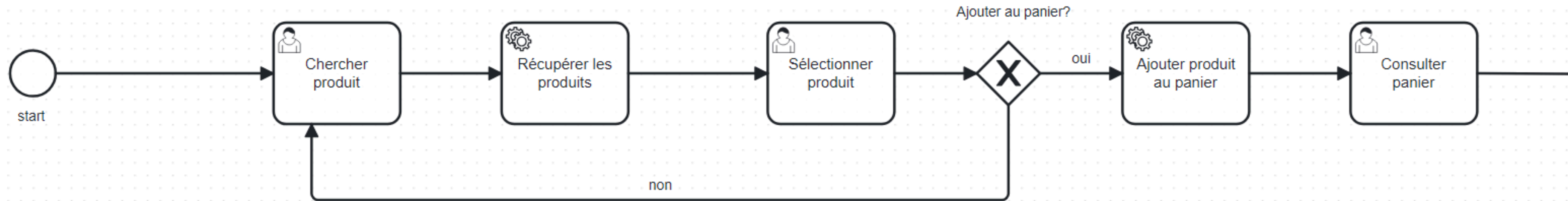


Retour

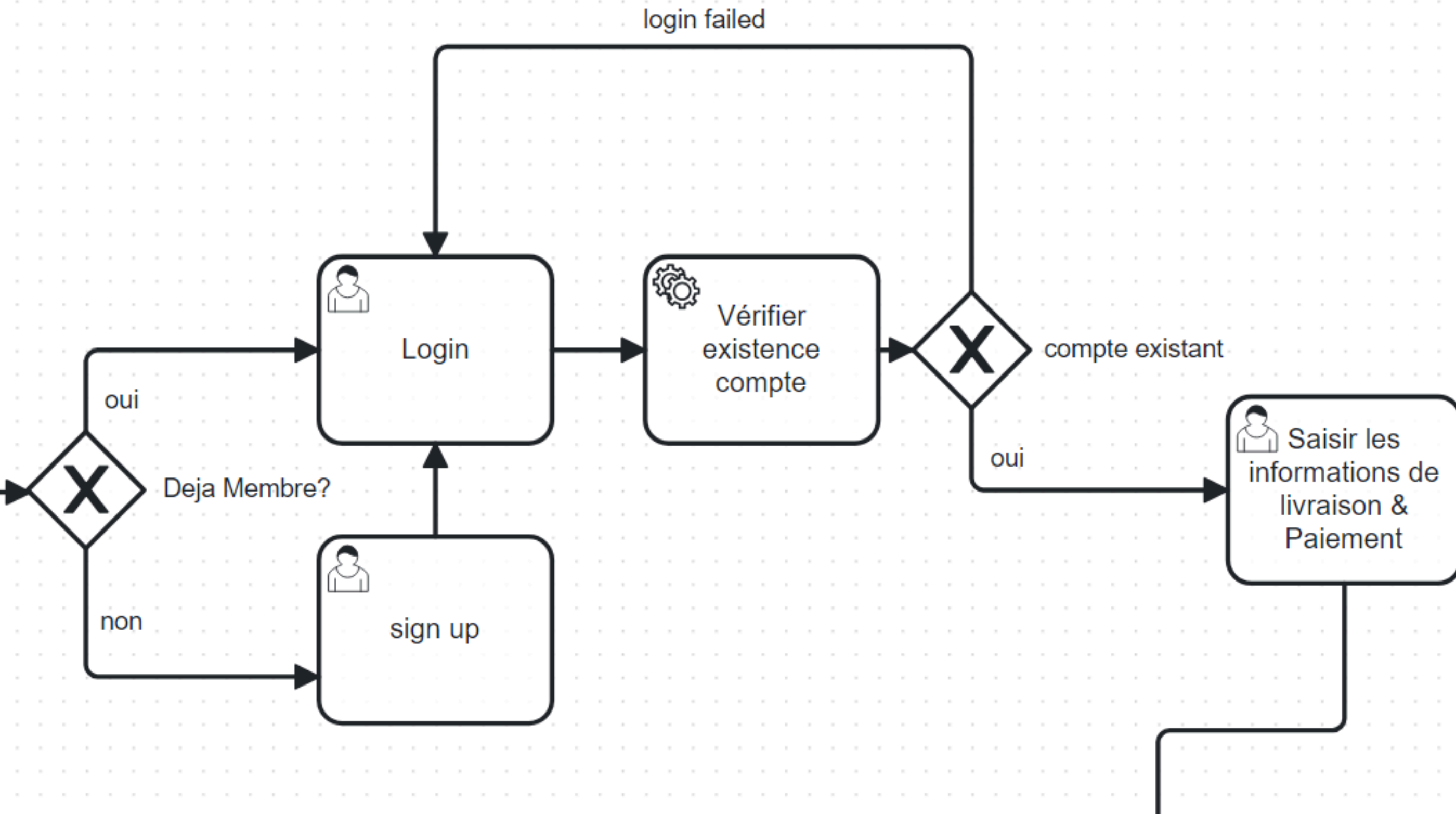
Ajouter au Panier



Sequence Navigation



Sequence Authentication



Form Definition

Nom

Telephone

Adresse

Code postal

Mode paiement

☐ Espèces

☐ Carte

Passer au paiement

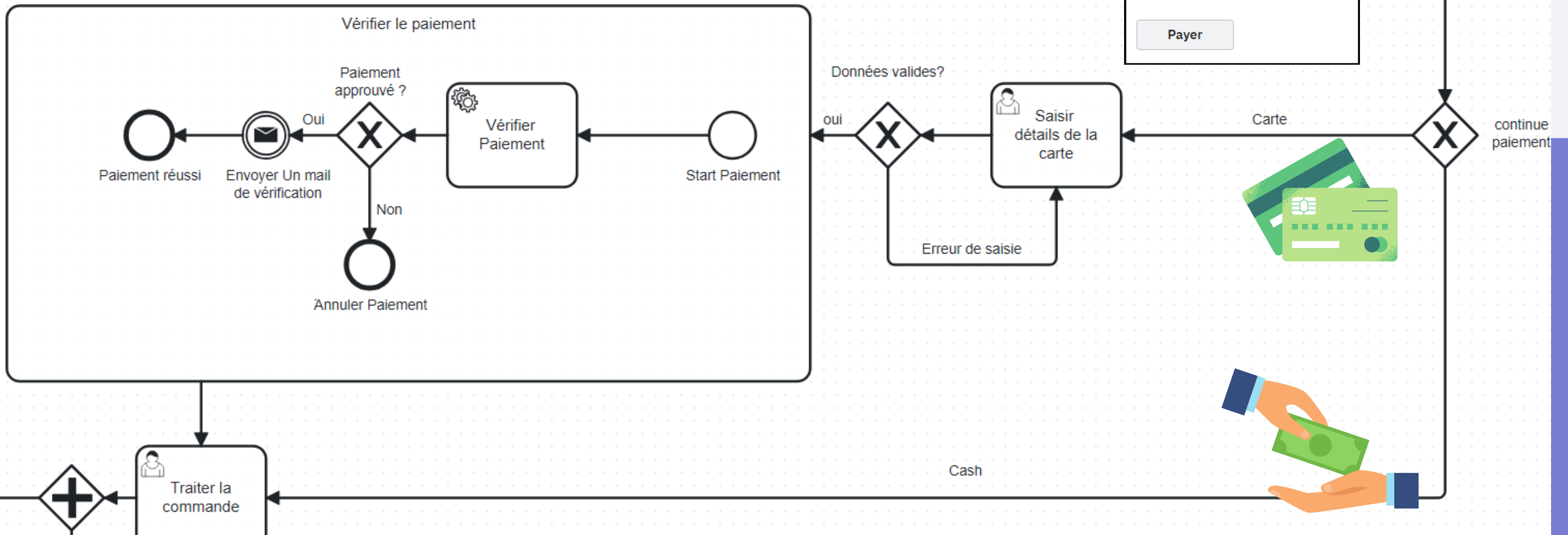
Sequence Paiement

Nom Complet

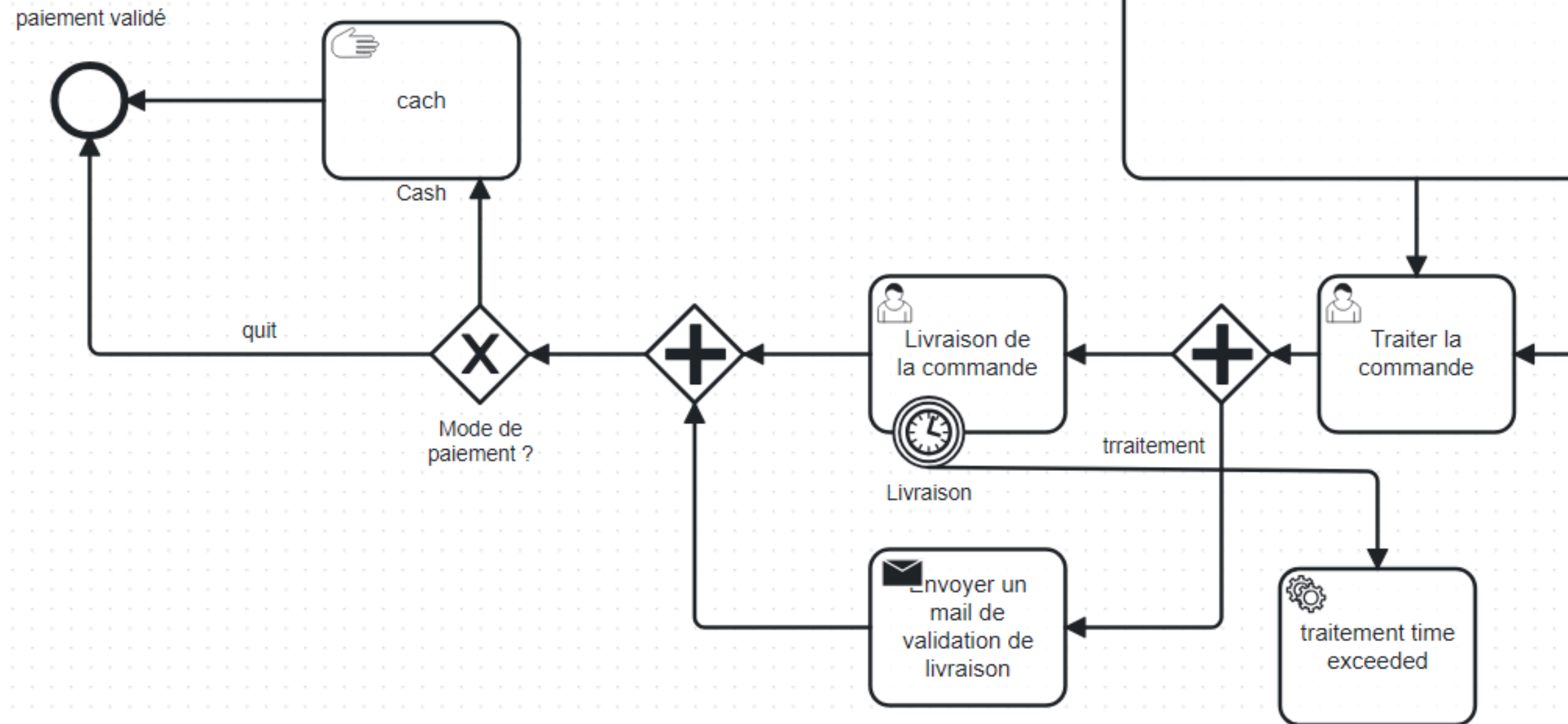
Numero de la carte

Code 4 chiffres

Email



Sequence Livraison



Nom Produit

Prix

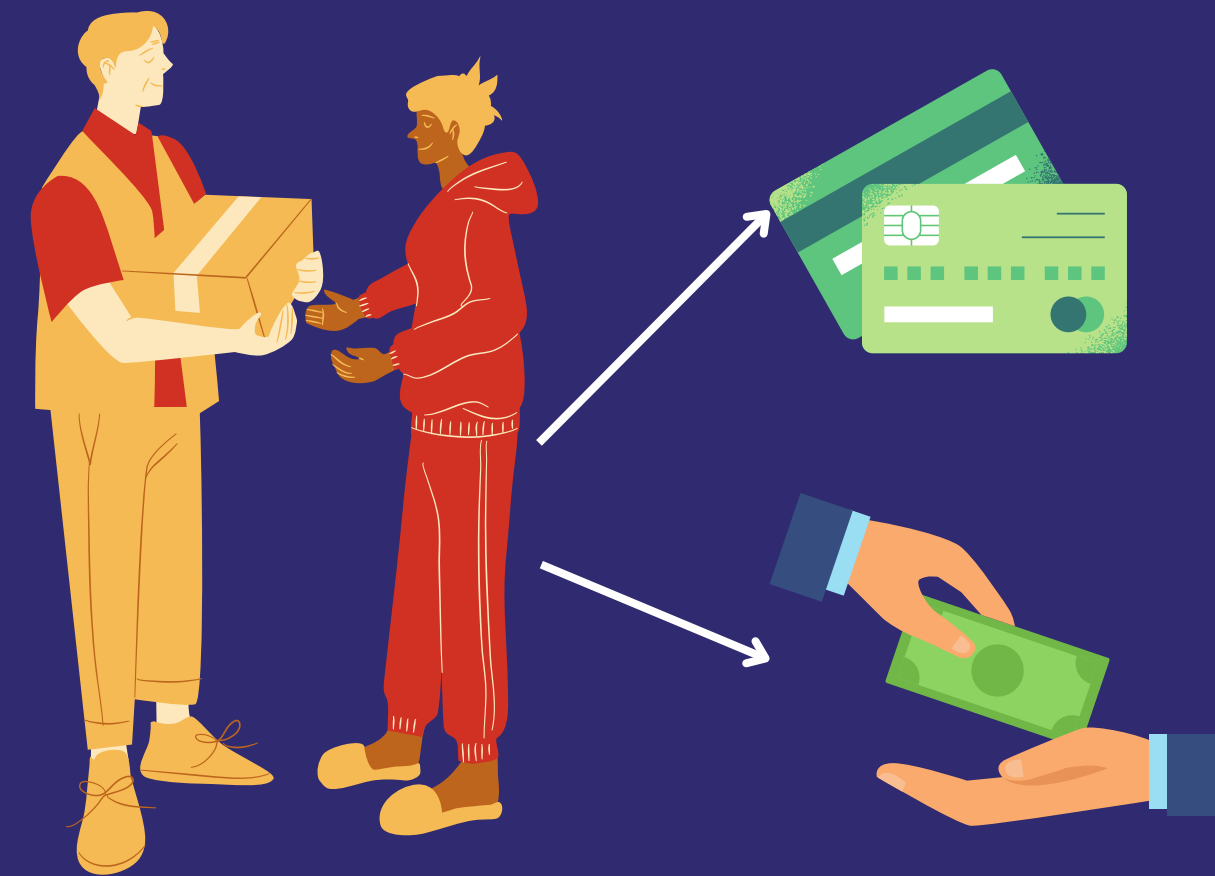
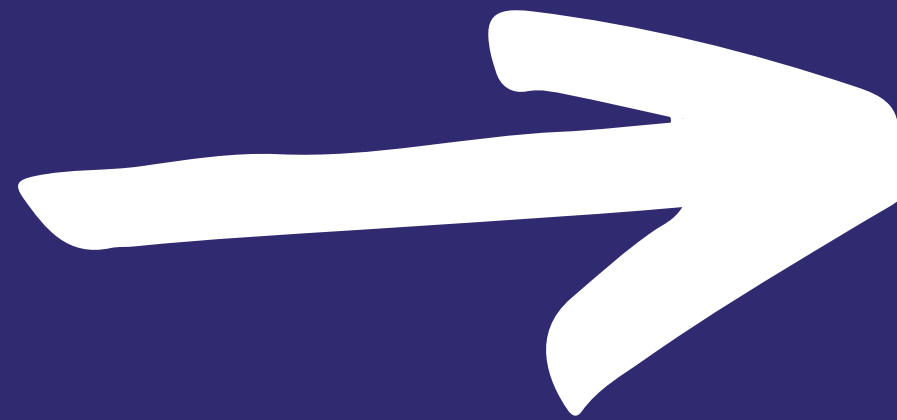
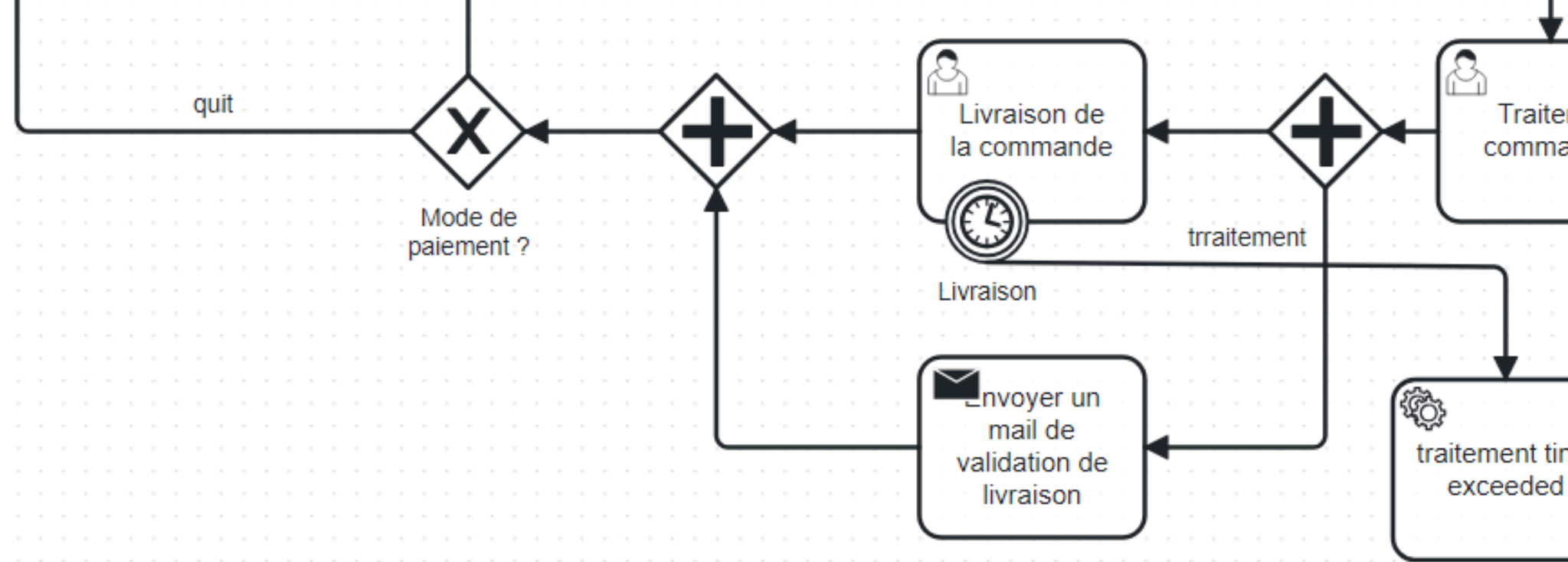
Valider Commande

- ☐ Accepter
☐ Refuser

Nom Du livreur



Sequence Livraison



Sequence Livraison



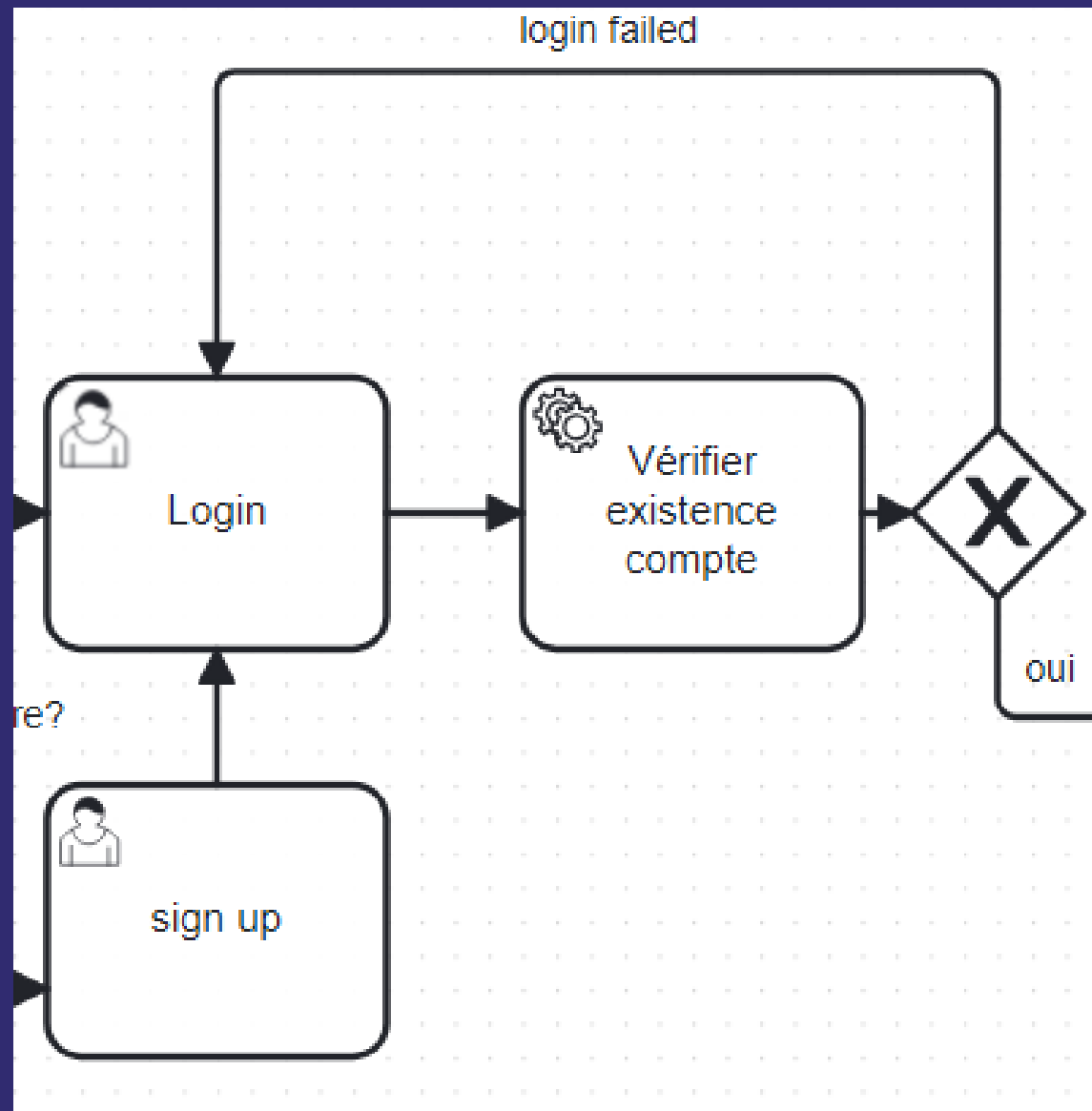
Montant Payé

Valider

Implémentation



Authentication



Login / Signup

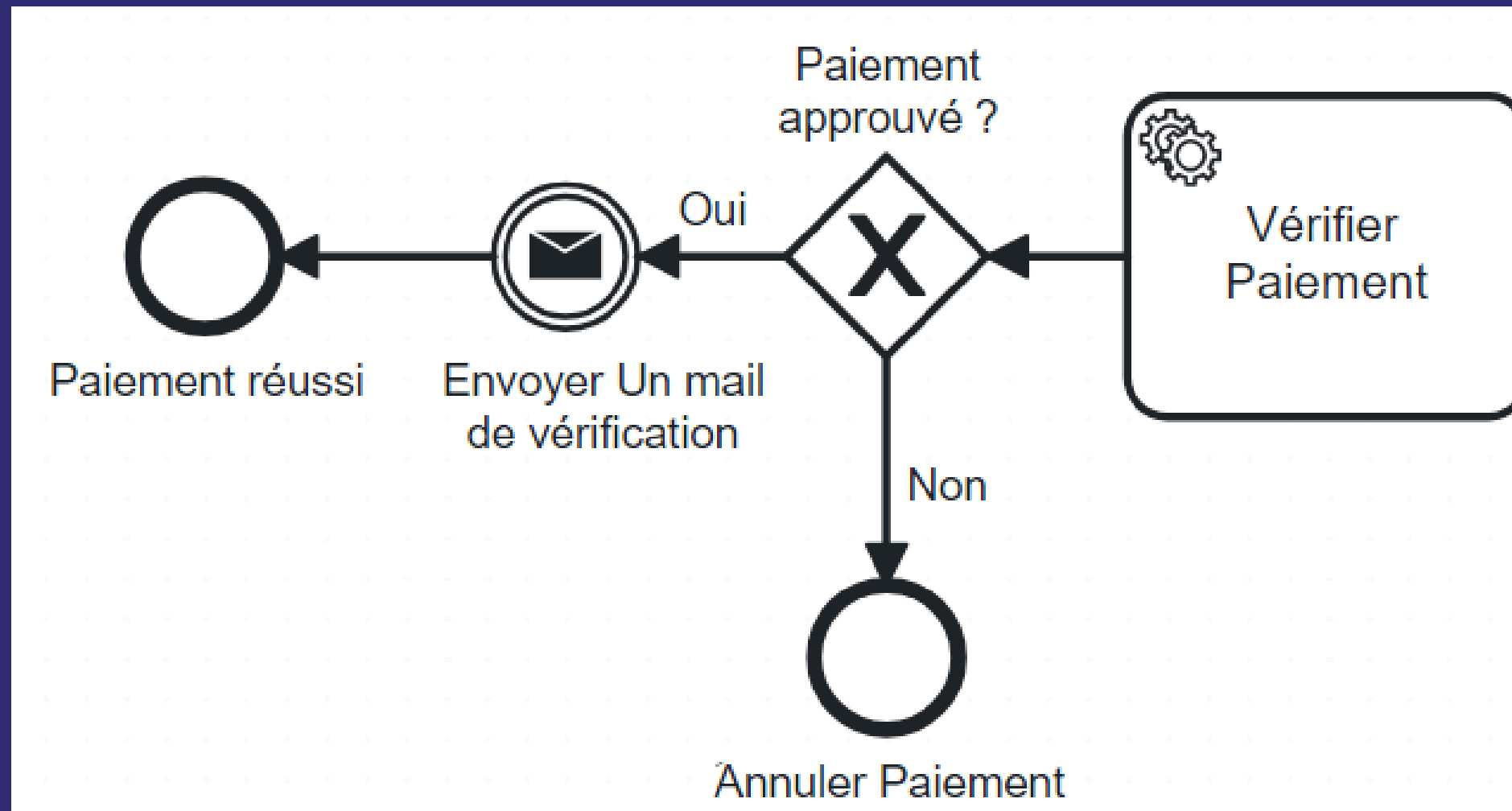
```
import io.camunda.zeebe.client.api.response.ActivatedJob;
import io.camunda.zeebe.client.api.worker.JobClient;
import io.camunda.zeebe.client.api.worker.JobHandler;

public class LoginHandler implements JobHandler { 2 usages new *

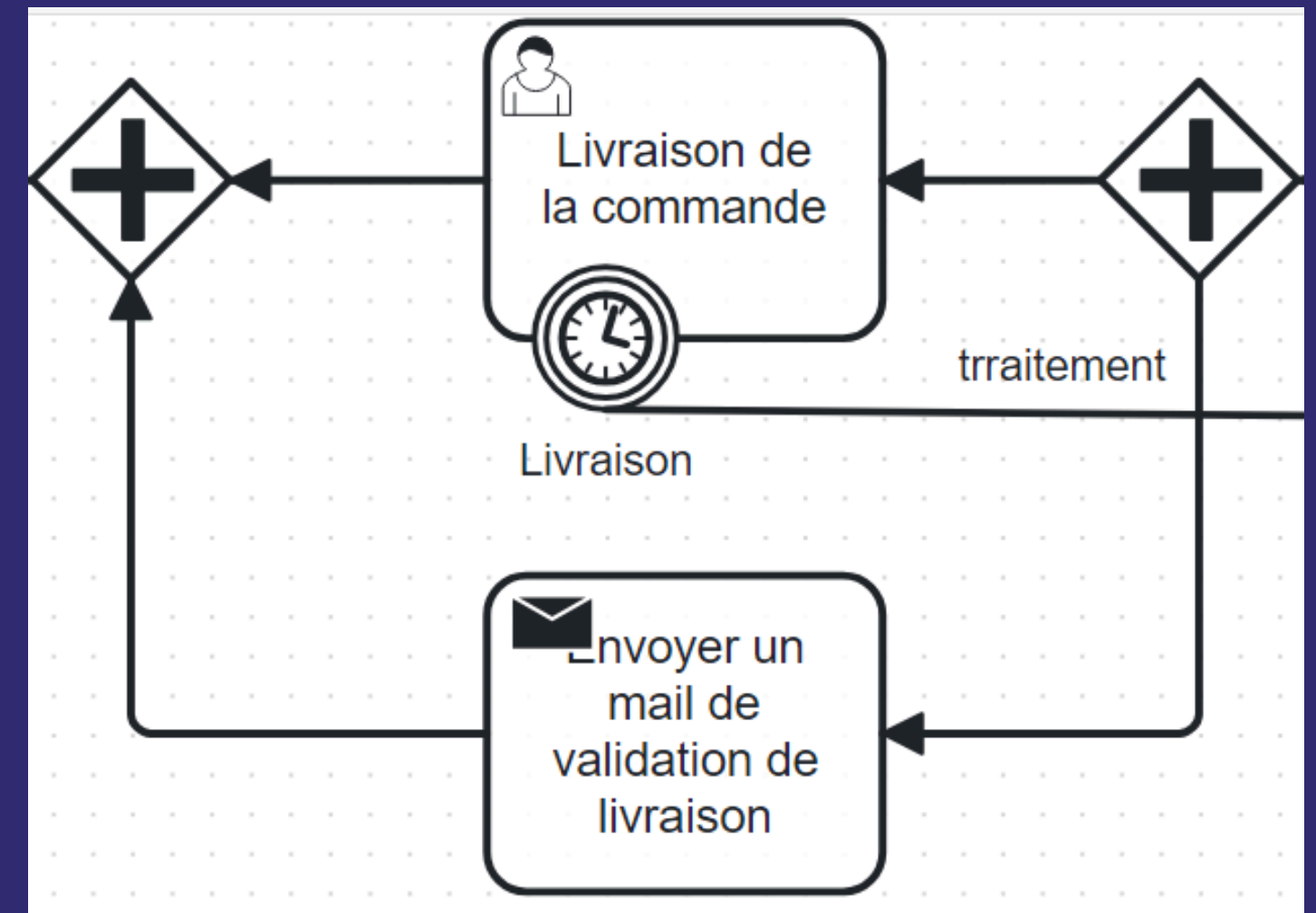
    @Override new *
    public void handle(JobClient client, ActivatedJob job) throws Exception {
        String username = (String) job.getVariablesAsMap().get("email");
        String password = (String) job.getVariablesAsMap().get("password");

        if (authenticate(username, password)) {
            client.newCompleteCommand(job.getKey()).send().join();
        } else {
            client.newFailCommand(job.getKey()).retries(remainingRetries: 0).errorMessage(errorMessage: );
        }
    }
}
```


Notifications



Paieement



Début de livraison

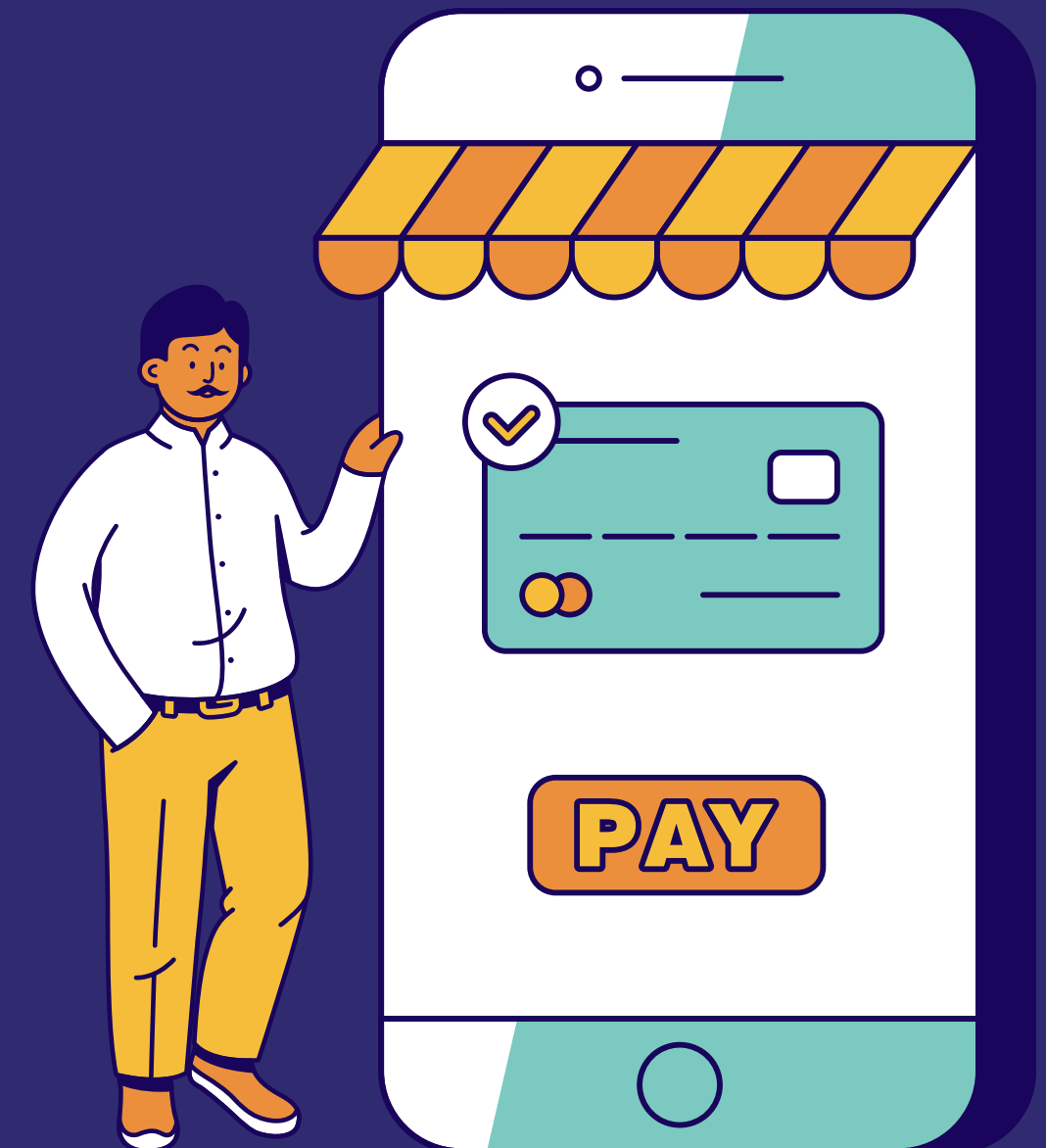
Notifications

```
5
6      try (final ZeebeClient travelAgencyClient = ZeebeClient.newClientBuilder()
7          .gatewayAddress(ZEEBE_ADDRESS)
8          .credentialsProvider(credentialsProvider)
9          .build()) {
10
11
12      //Build the Message Variables
13      final Map<String, Object> messageVariables = new HashMap<String, Object>();
14
15      messageVariables.put("payement", true);
16      System.out.println(articlesToSend + "Confirmation de votre commande");
17
18      //Complete the Job
19      client.newCompleteCommand(job.getKey()) CompleteJobCommandStep1
20          .variables(messageVariables)
21          .send() ZeebeFuture<CompleteJobResponse>
22          .join();
23      }
24  }
25  }
26
27
```

Conclusion

En conclusion, la combinaison du diagramme BPMN pour un système d'e-commerce et de son implémentation Java a été cruciale pour développer un processus d'achat en ligne, de paiement et de livraison efficace.

Cette méthodologie permet de créer une solution robuste qui répond aux exigences du commerce électronique modern



Thank you!

SellFLY