



## **Projet OC Pizza**

Linocent Timothée  
Développeur Junior  
IT Consulting & Development



1 -Objet du document.....	3
1.2 -Besoin du client.....	3
2 -Description générale de la solution.....	4
2.1 -Cycle de vie d'une commande.....	4
2.2 -Les acteurs.....	5
2.3 Cas du client .....	6
2.4 Cas de l'administrateur .....	11
2.5 Cas du pizzaïolo .....	16
2.6 Cas du livreur .....	19
2.7 Cas du caissier .....	22
3. Les solutions techniques envisagées.....	25
3.1 Le langage.....	25
3.2 Le Framework.....	25
3.3 La database.....	26
3.4 L'hébergement.....	26
3.5 Support de l'application.....	28
3.6 Schéma solution technique.....	28
4. Annexe.....	29



## 1. Introduction

### 1. Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application OC Pizza.

L'objectif du document est de présenter le cahier des charges, les fonctionnalités et les solutions trouvées.

### 1.2. Besoin du client

Contexte :

"OC Pizza" est un jeune groupe de pizzeria en plein essor. Créé par Franck et Lola, le groupe est spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d'en ouvrir au moins 3 d'ici 6 mois.

Le système informatique actuel ne répond plus aux besoins du groupe car il ne permet pas une gestion centralisée de toutes les pizzerias.

De plus, il est très difficile pour les responsables de suivre ce qui se passe dans les points de ventes.

Enfin, les livreurs ne peuvent pas indiquer "en live" que la livraison est effectuée.

Enjeux et objectifs :

Le client souhaite :

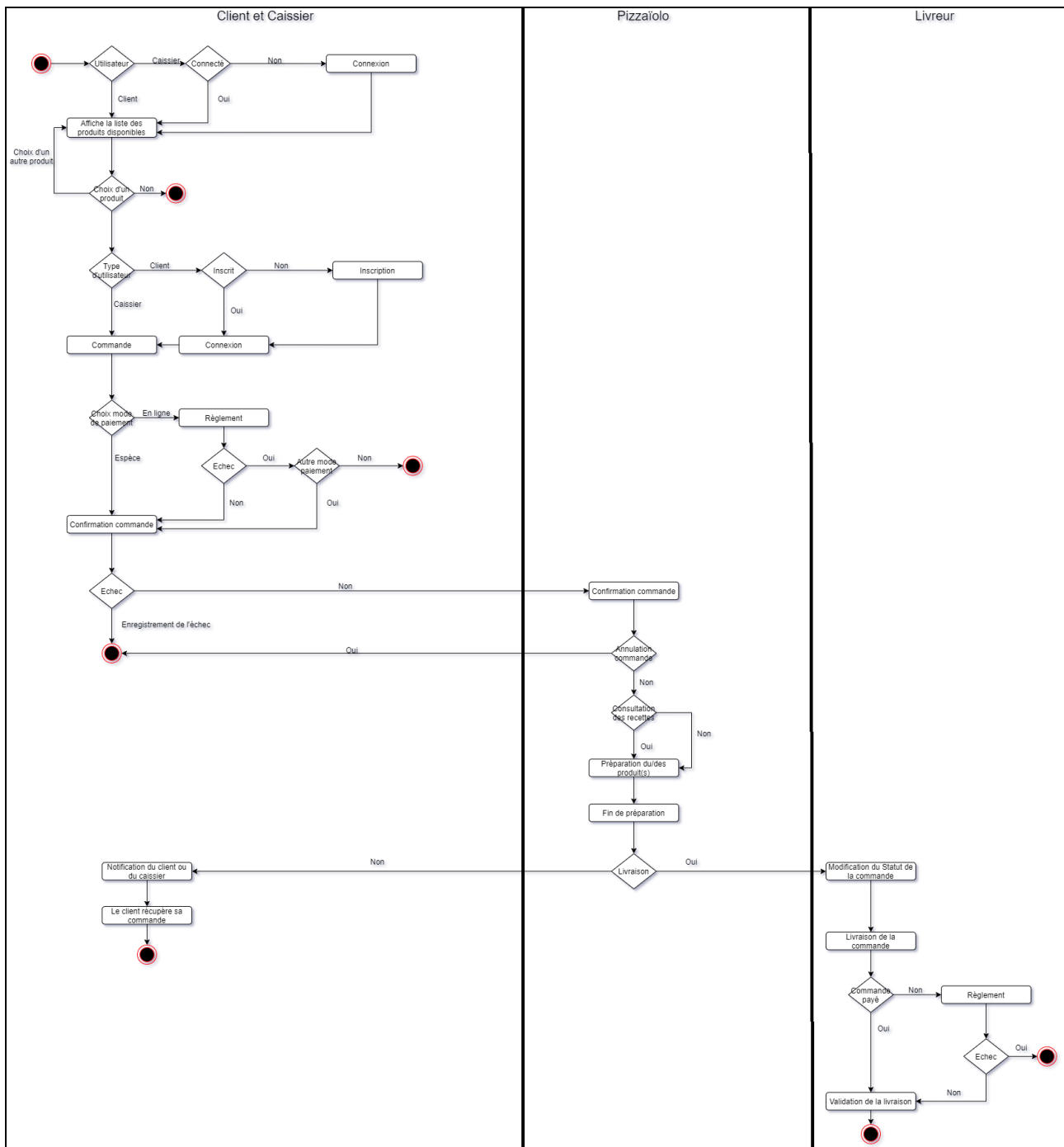
- Être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
- Suivre en temps réel le stock d'ingrédients restant pour savoir quelles pizzas peuvent encore être réalisées ;
- Proposer un site internet pour que les clients puissent :
  - Passer leurs commandes, en plus de la prise de commande par téléphone ou sur place ;
  - Payer en ligne leur commande s'ils le souhaitent - sinon, ils paieront directement à la livraison ;
  - Modifier ou annuler leur commande tant que celle-ci n'a pas été préparée.
- Proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza.

Le client souhaite obtenir le système informatique sous 6 mois, pour l'ouverture des trois nouvelles pizzérias.



## 2. Description générale de la solution.

### 2.1. Cycle de vie d'une commande



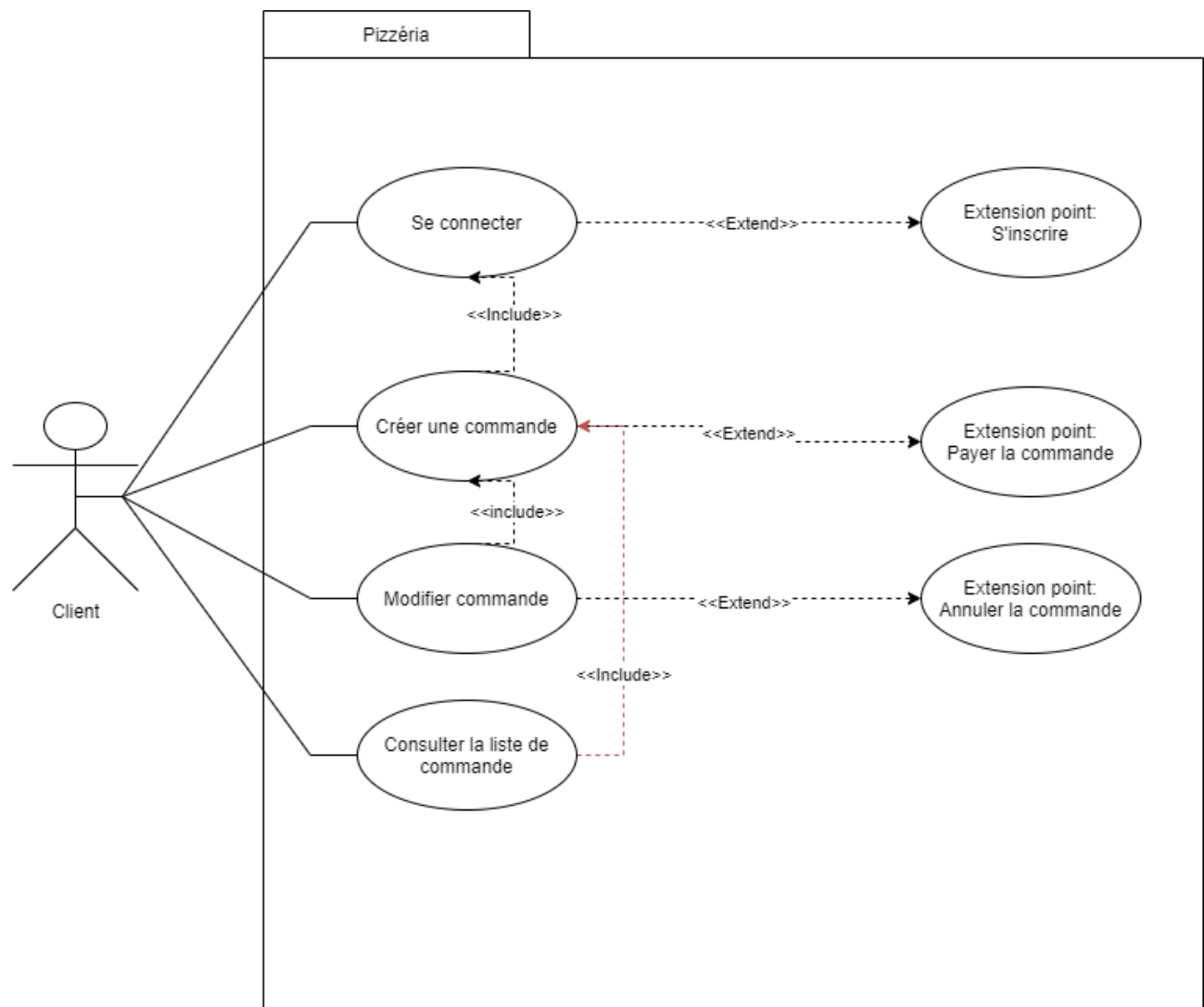


## 2.2. Les acteurs

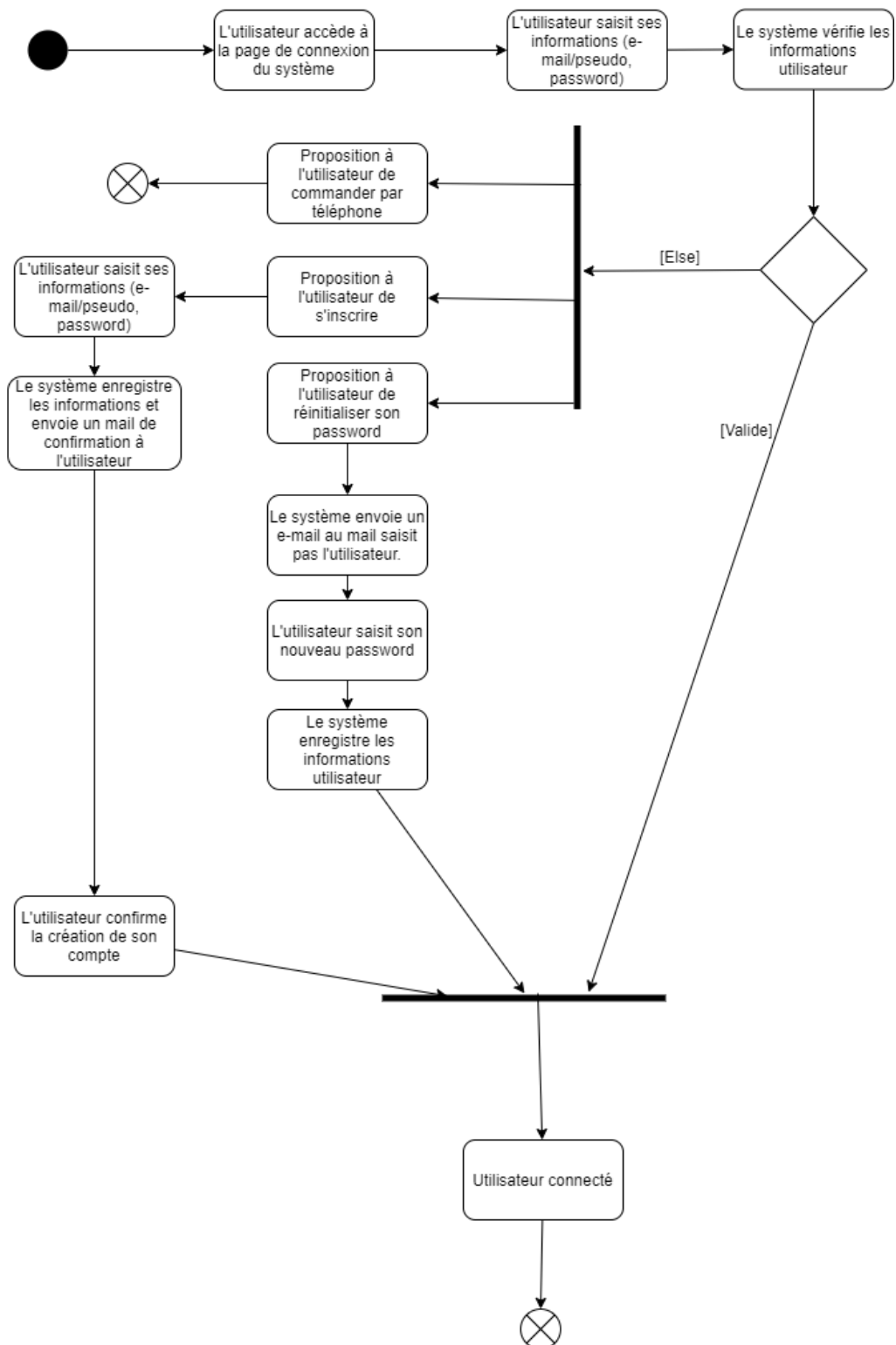
Les acteurs dénombrés sont au nombre de cinq. Nous trouvons :

- Les clients qui sont des utilisateurs souhaitant commander sur place, par téléphone ou via Internet parmi les produits disponibles, souhaitant annuler ou modifier une commande ;
- Les Pizzaiolos qui sont des utilisateurs consultant la liste des commandes en attente, les recettes des différents produits, validant la mise en préparation des commandes ;
- Les livreurs qui sont des utilisateurs consultant la liste des commandes en attente de livraison, les informations clients relatives à la commande, validant le début et la fin de livraison de la commande ;
- Les caissiers qui sont des utilisateurs utilisant le système pour insérer les commandes sur place et par téléphone. Ils saisisent si le paiement a été réglé sur place ou non.
- L'administrateur qui est un utilisateur possédant tous les droits sur le système. Il peut consulter l'état des stocks d'ingrédients, l'historique et la liste des commandes en cours, créer ou modifier un compte employé.

## 2.3. Cas du client



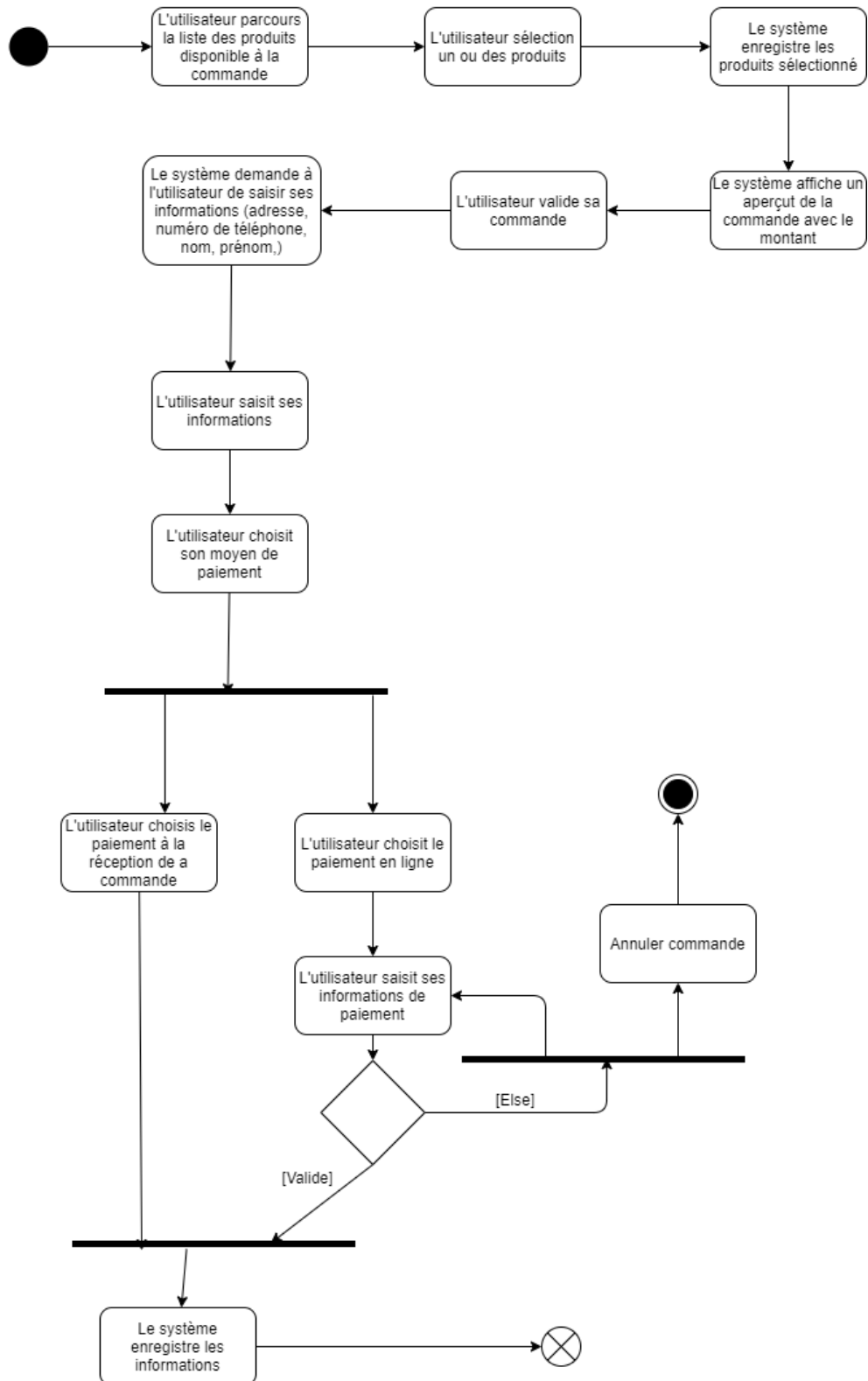
## Cas n° 1.1: Se connecter





# OC Pizza

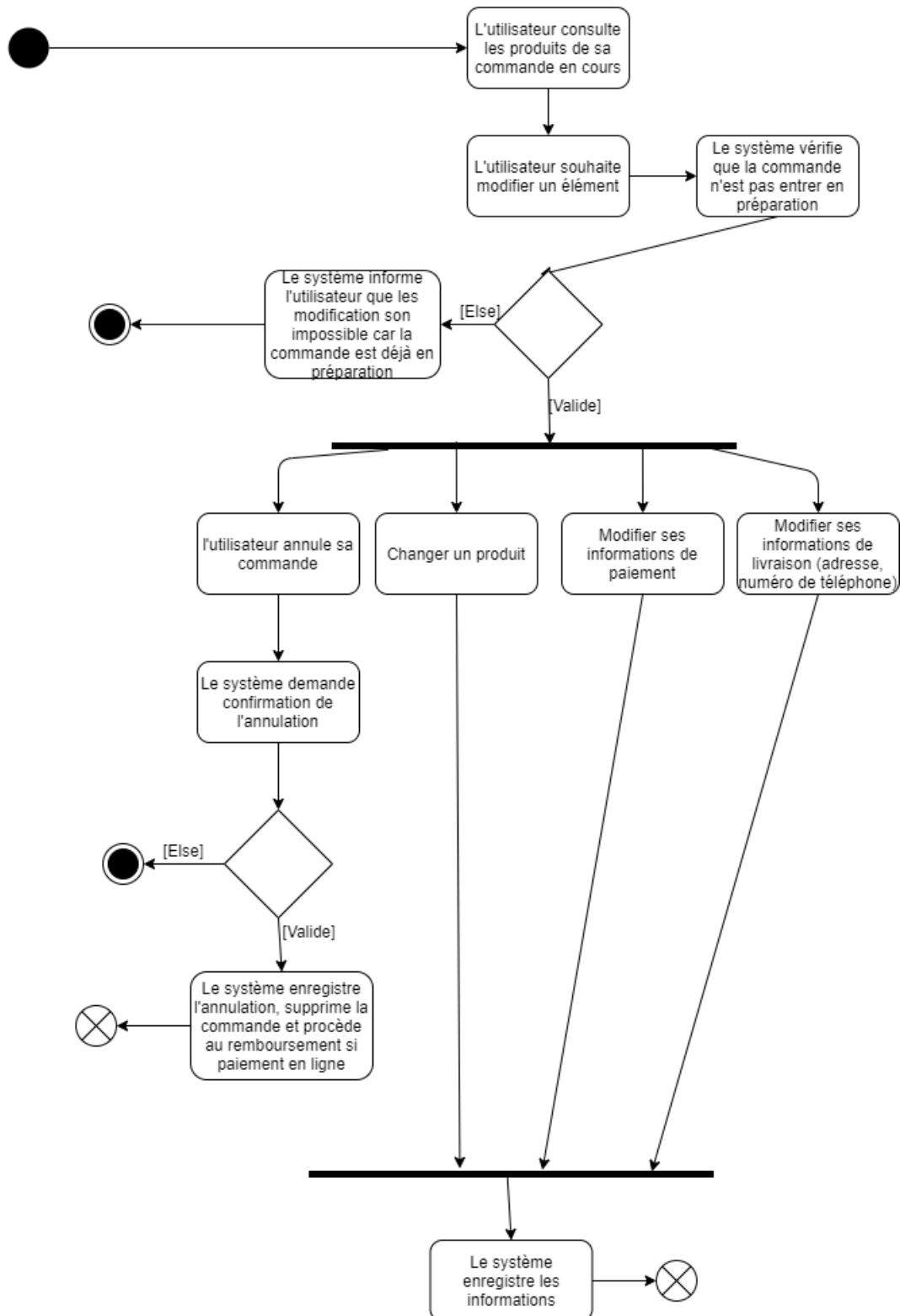
## Cas n° 1.2: Créer une commande.







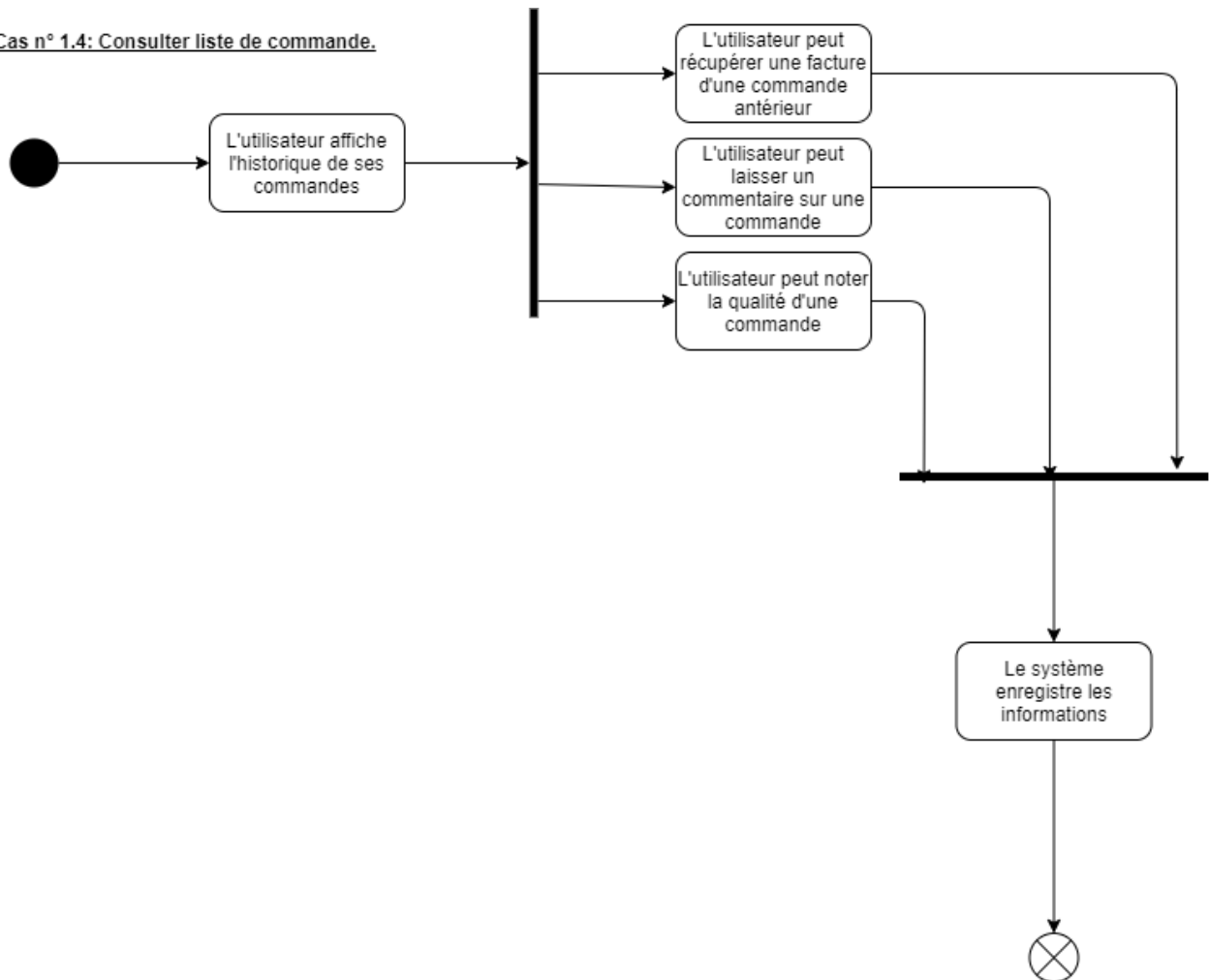
## Cas n°1.3 : Modifier commande.



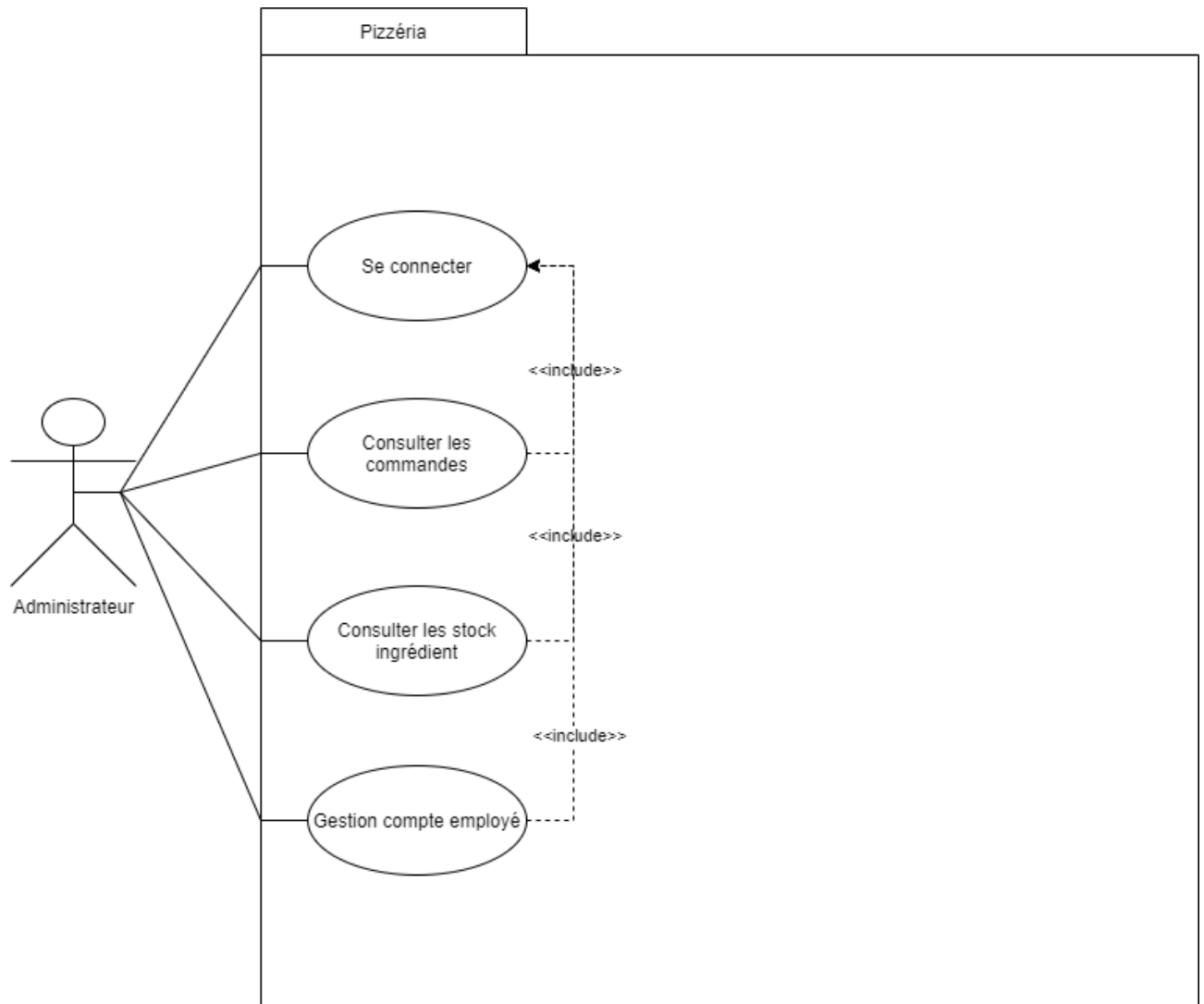


# OC Pizza

## Cas n° 1.4: Consulter liste de commande.

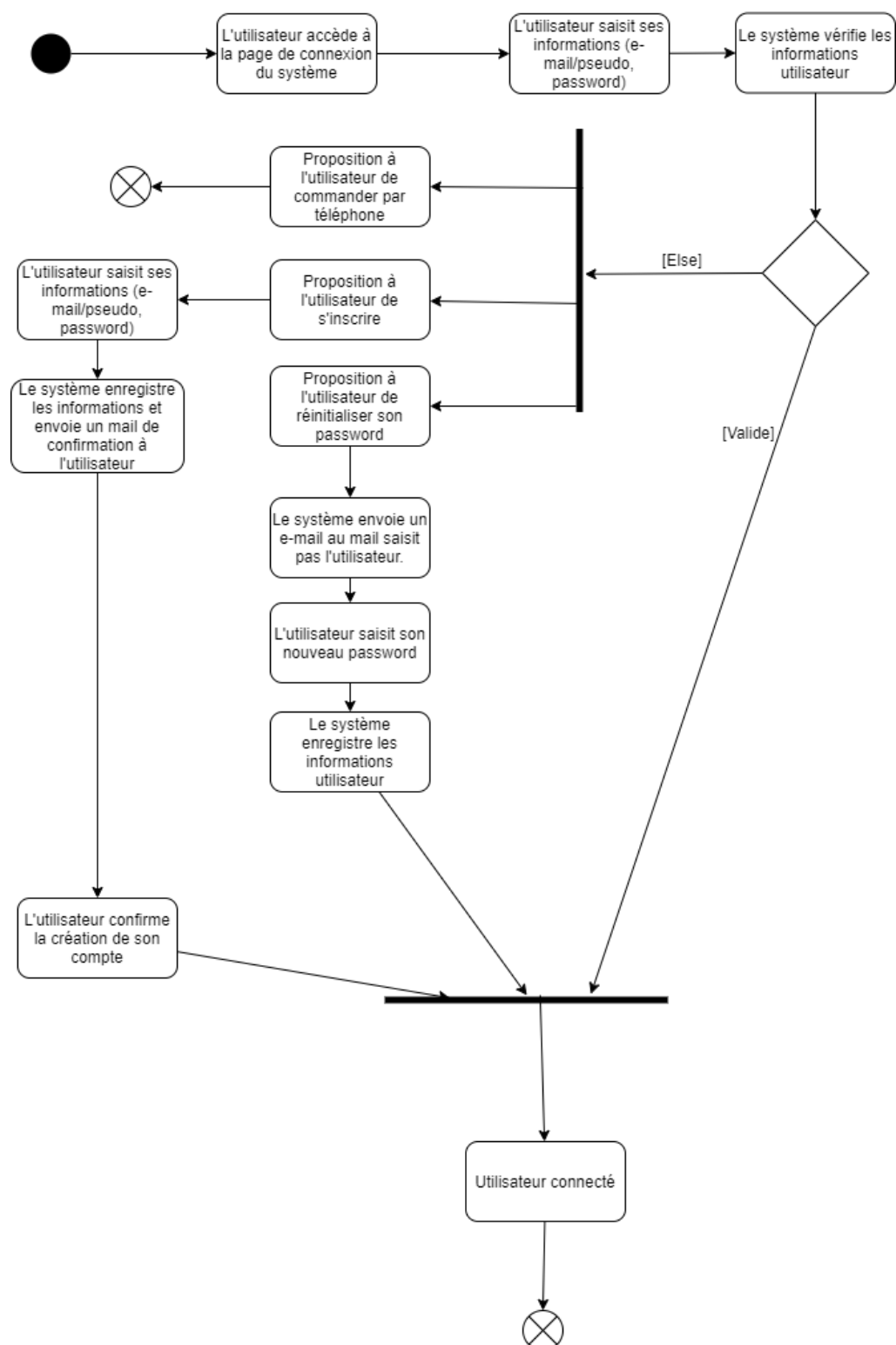


## 2.4. Cas de l'administrateur

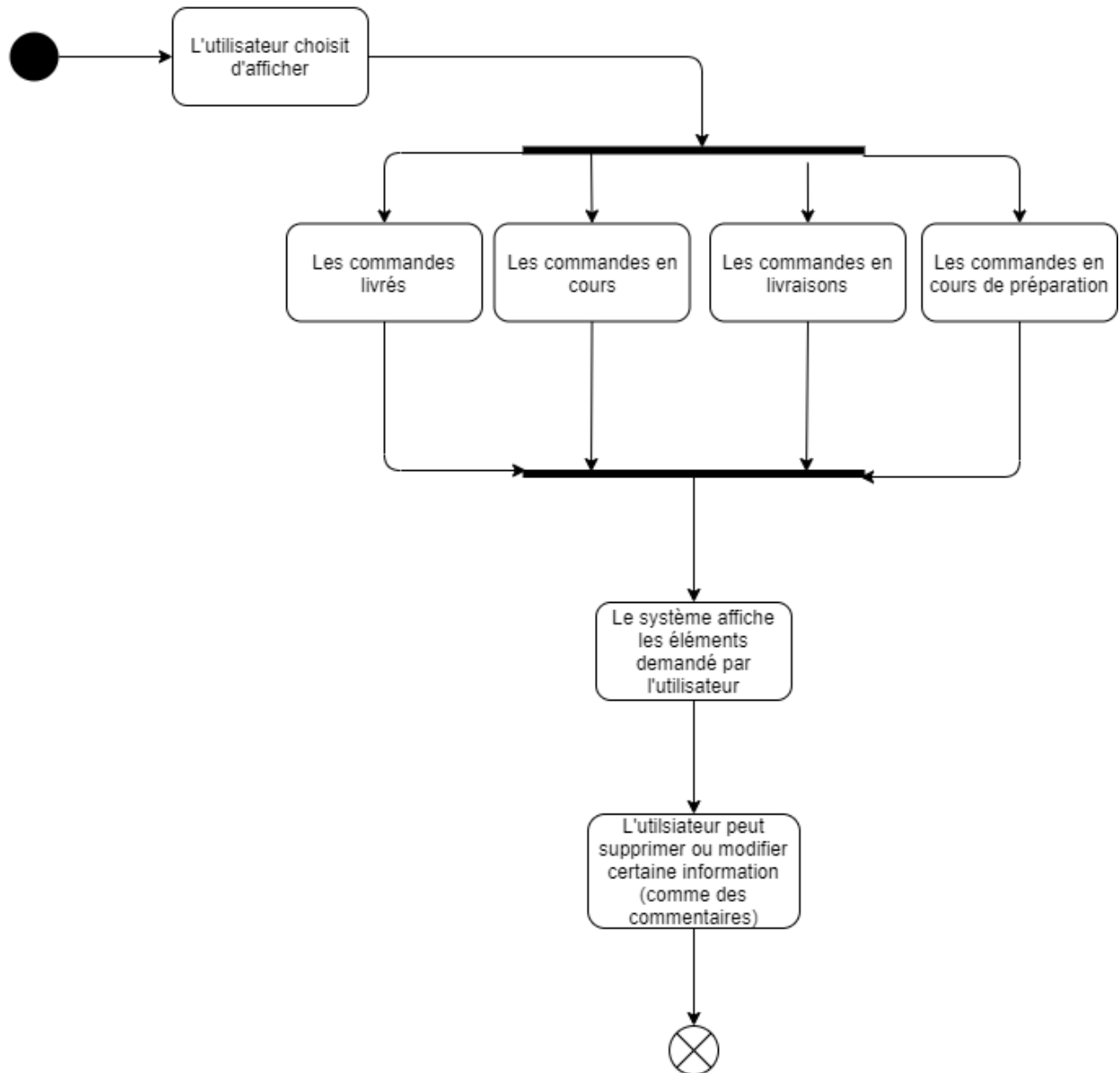




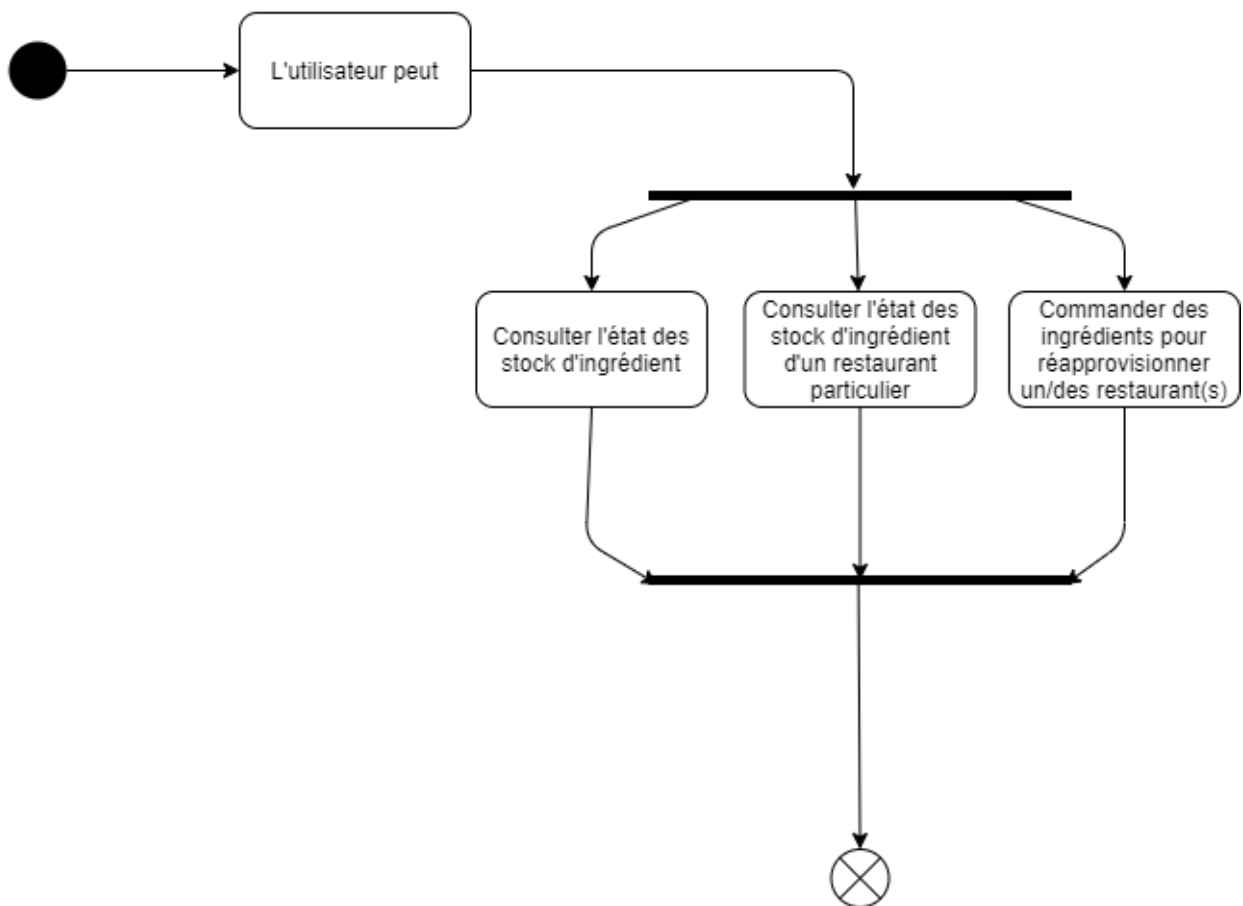
Cas n° 2.1: Se connecter



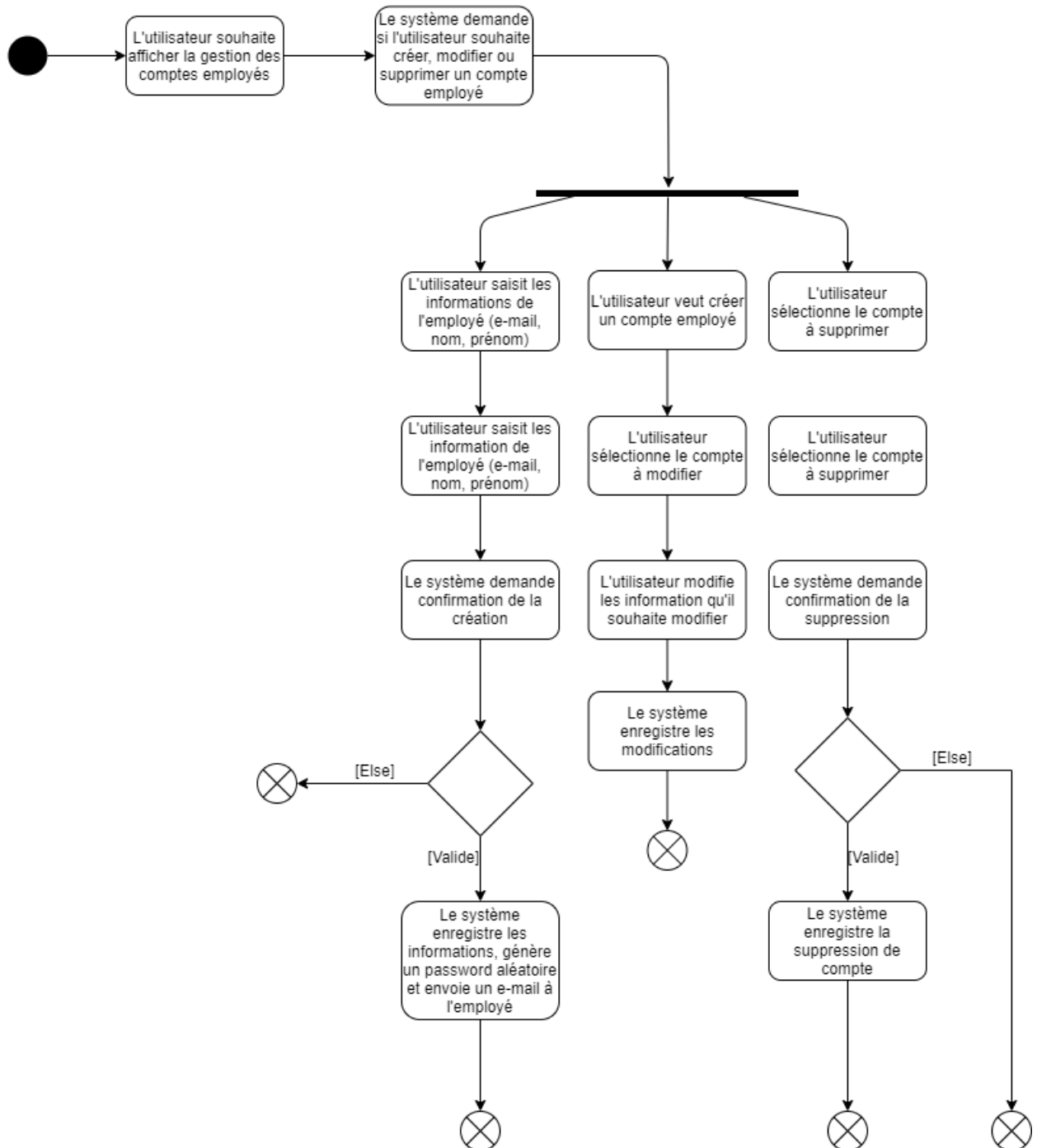
Cas n° 2.2 : Consulter les commandes.



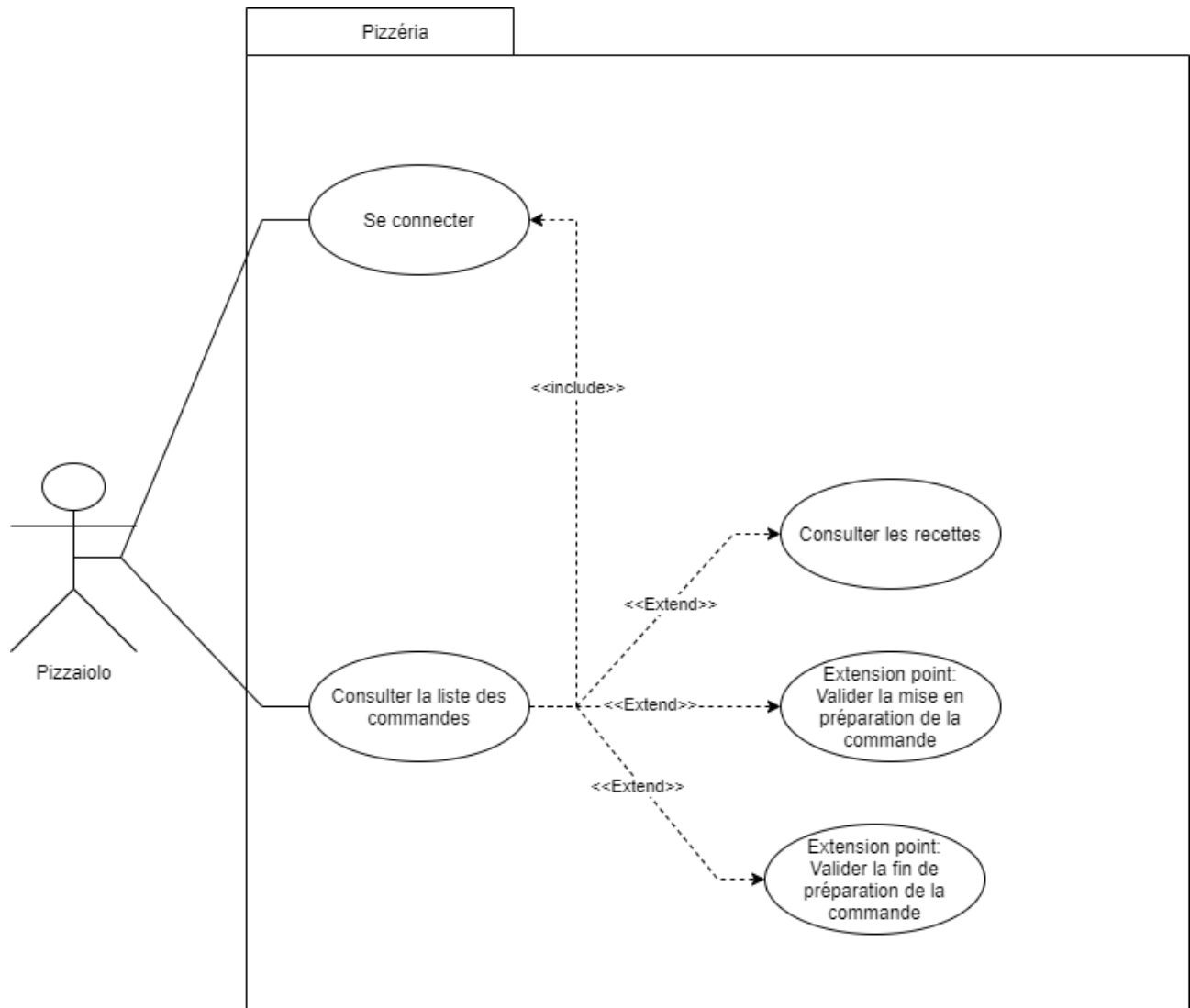
Cas n° 2.3: Consulter les stocks ingrédients



## Cas n° 2.4: Gestion compte employé.

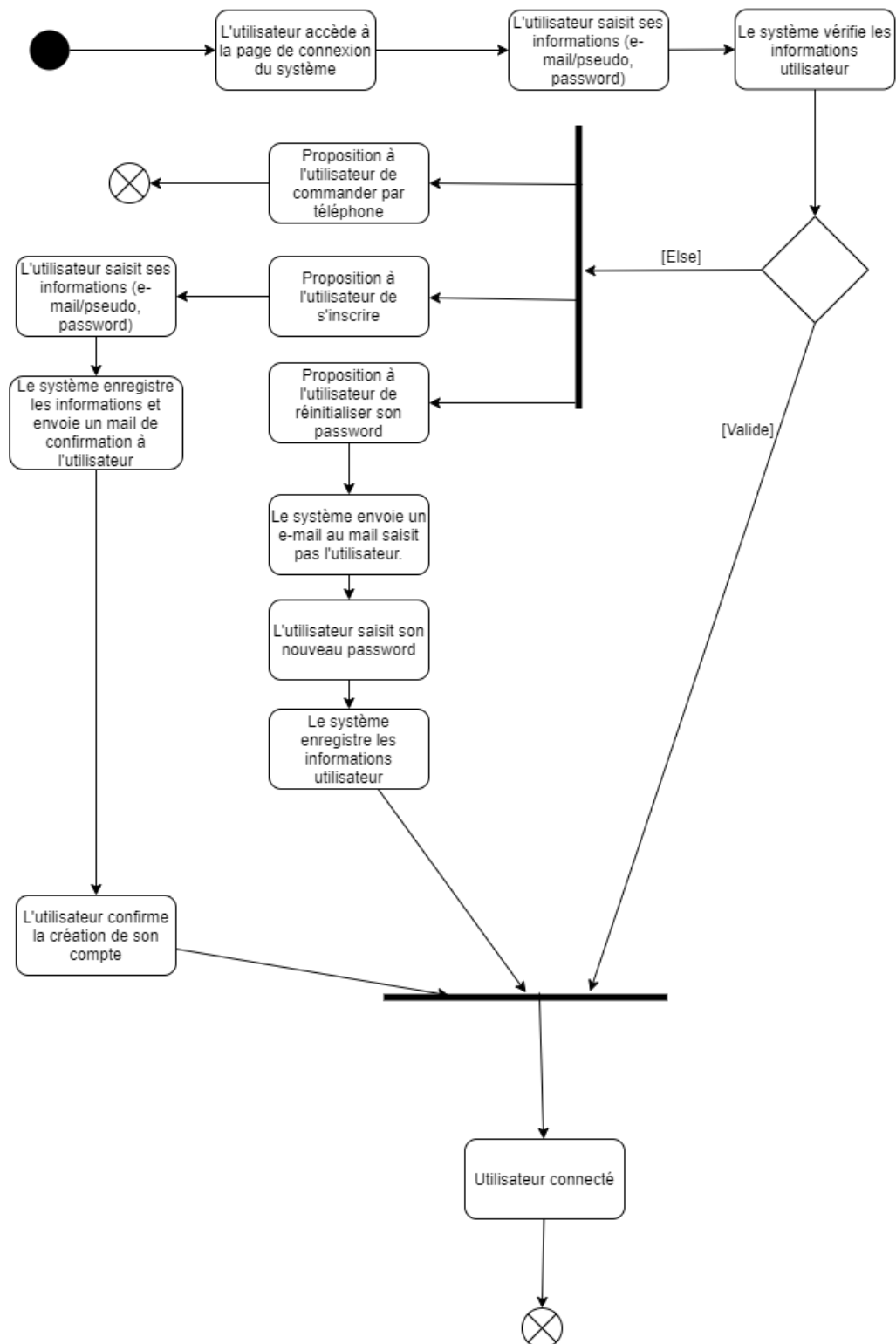


## 2.5. Cas du Pizzaiolo



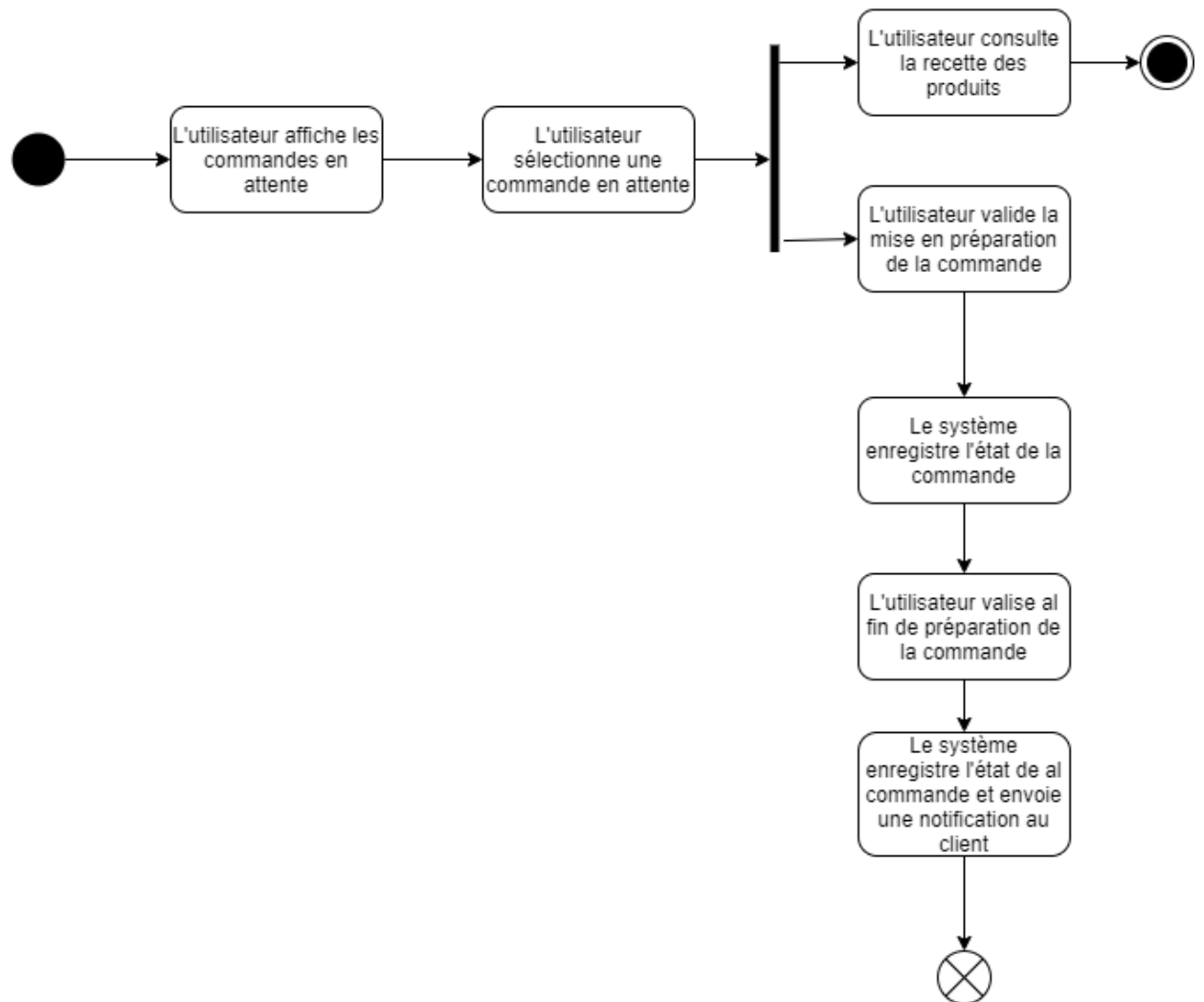


## Cas n° 3.1: Se connecter

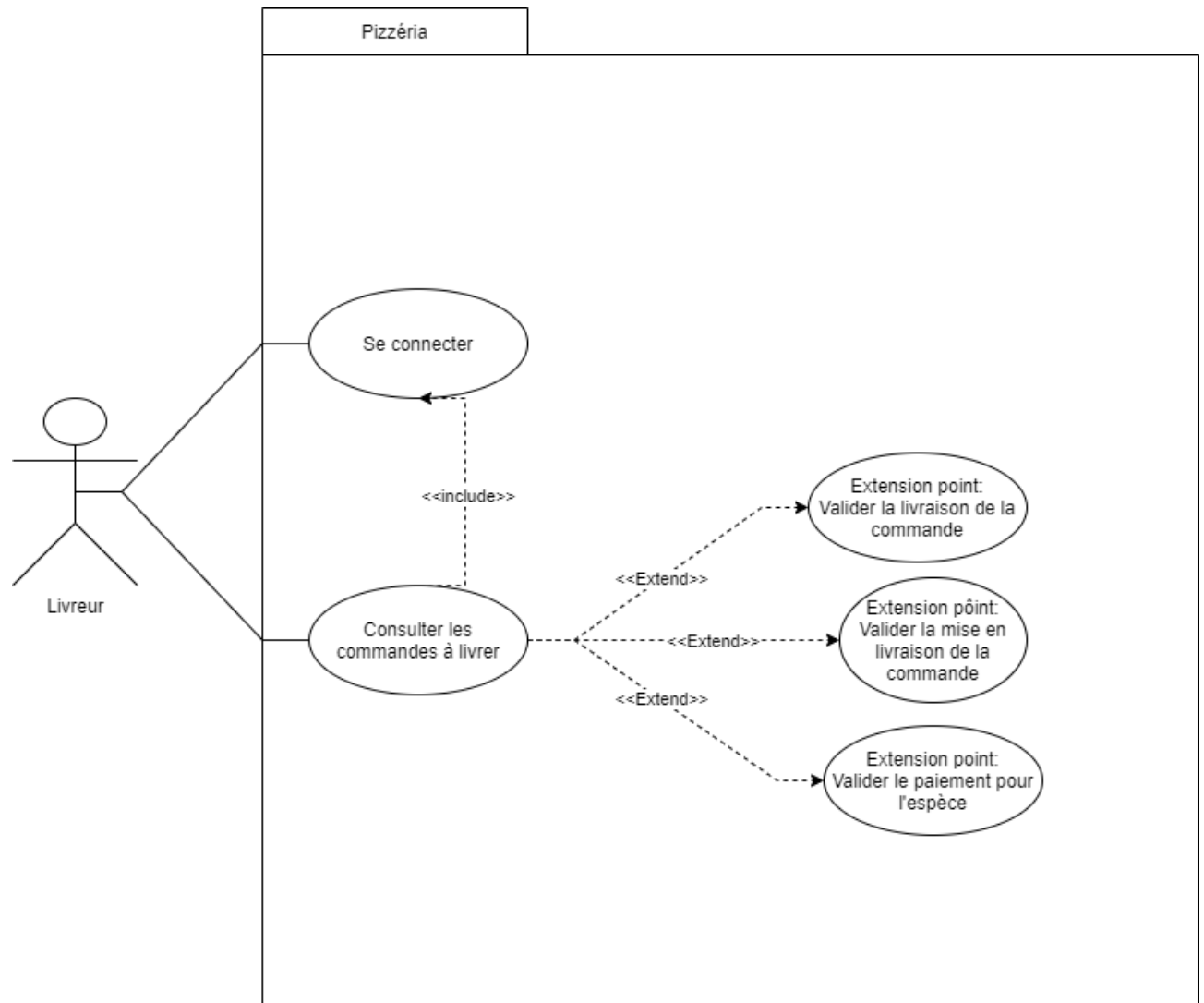




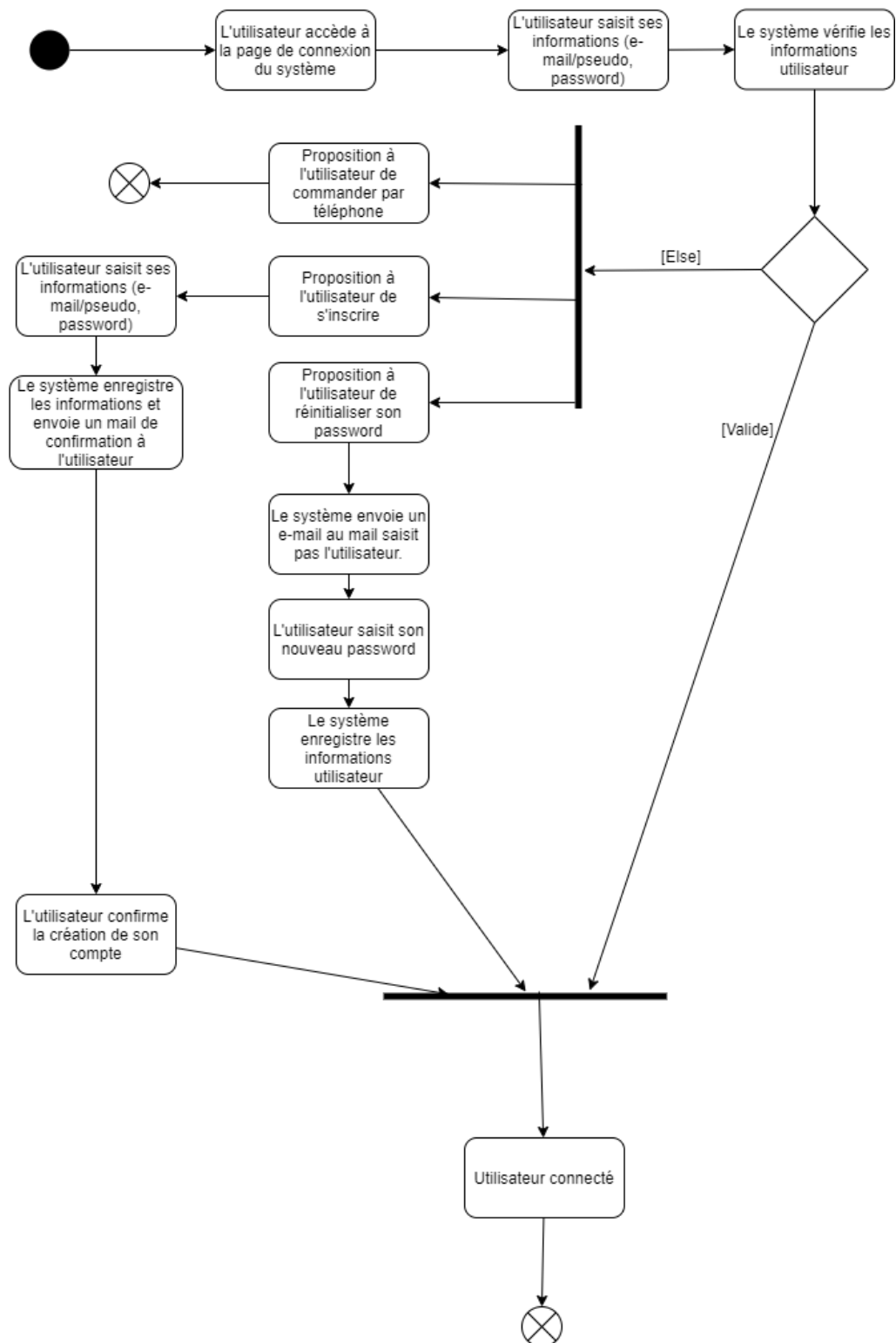
## Cas n° 3.2 : Consulter la liste des commandes.



## 2.6. Cas du Livreur

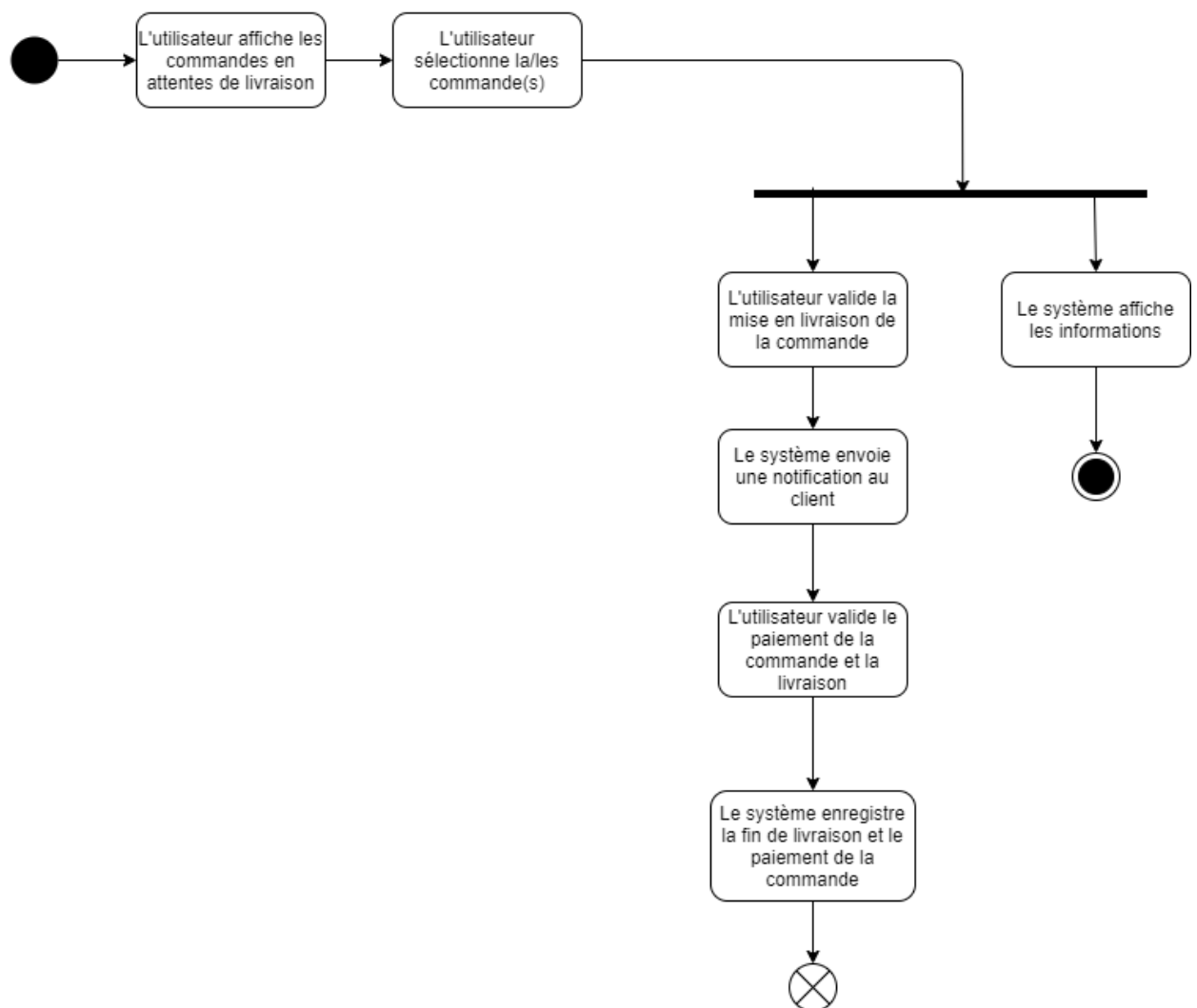


## Cas n° 4.1: Se connecter



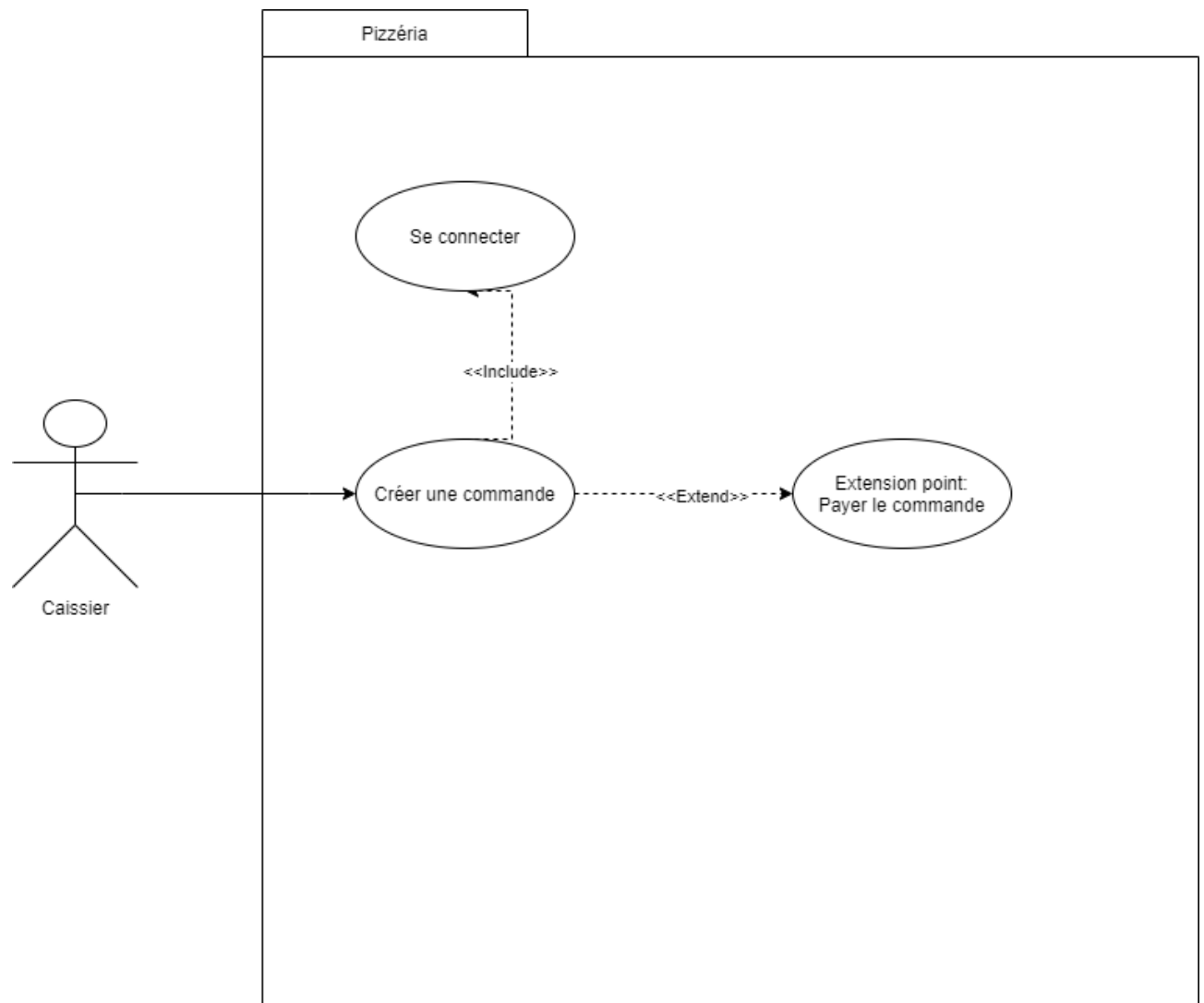


Cas n° 4.2: Consulter les commandes à livrer.

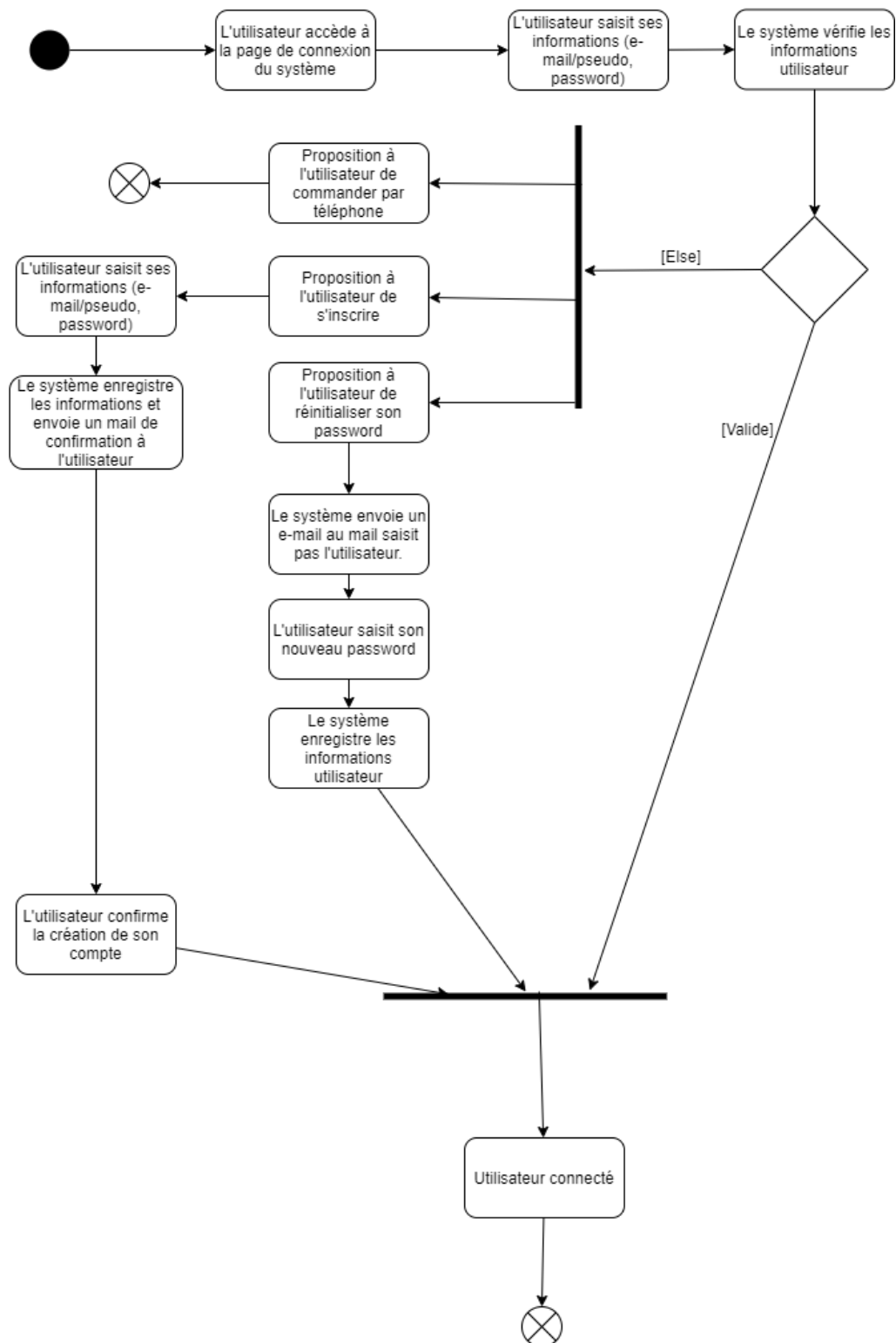




## 2.7. Cas du caissier

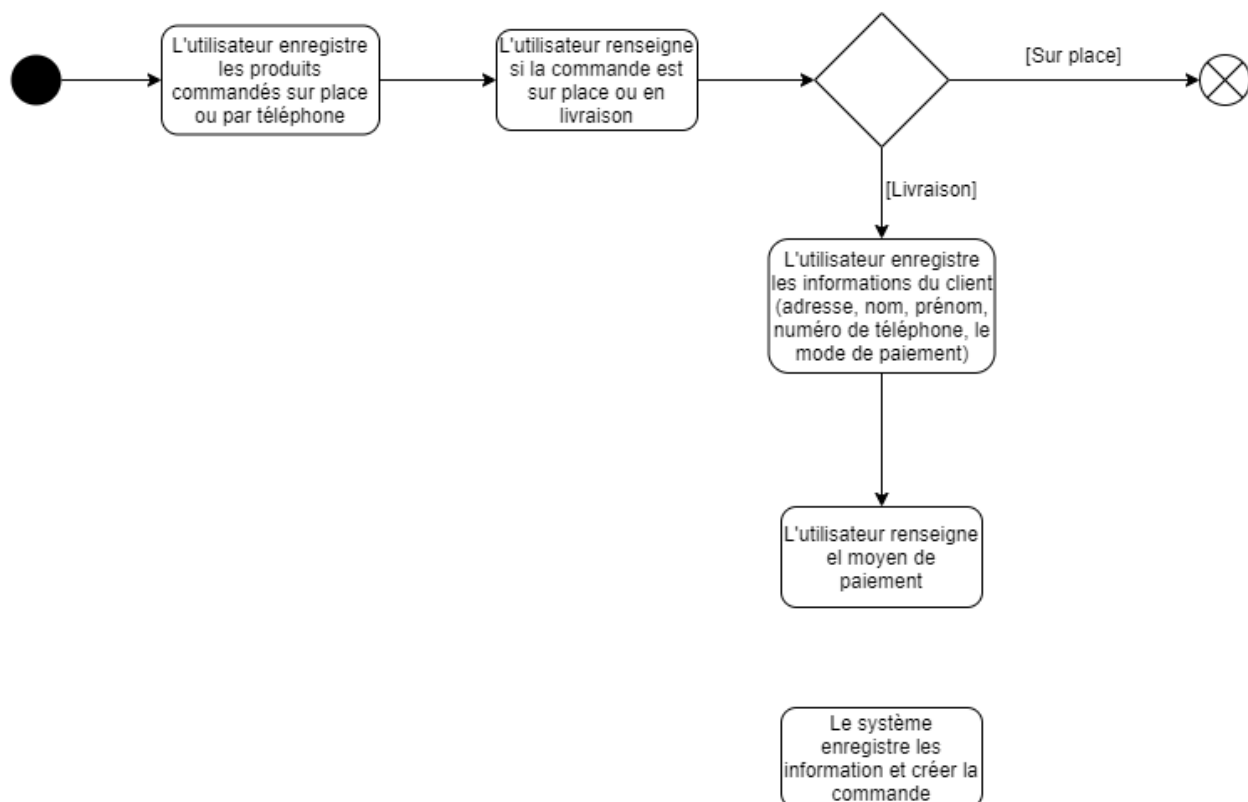


## Cas n° 5.1: Se connecter





Cas n° 5.2: Créer une commande







### 3. La solution technique envisagé

#### 3.1. Le langage

Selon plusieurs sources, voir annexe, le langage de programmation web le plus utilisé, le plus simple à mettre en pratique est le python. Le python est langage libre doté d'une syntaxe simple et facile à comprendre. Le python est un langage polyvalent qui admet beaucoup d'application.

Le langage Python fut utilisé dans la création d'Instagram, de Youtube et de Spotify, et est un des langages de programmation officiels de Google.

Il y a également le Java qui fait partie des langages les plus connu. Il est utilisé pour faire fonctionner des applications pour ordinateur. Il est également utilisé sur la plupart des supports numérique.

Cependant le langage Java est beaucoup plus gourmand en mémoire et ses simulations sont assez lentes. De plus, le langage est en constante évolution ce qui impliquera probablement une maintenance régulière et assez récurrente.

Nous allons choisir le Python comme langage de programmation pour ce projet pour sa simplicité de mise en œuvre et sa large bibliothèque de module. En outre, le python monte terme d'utilisation. Il vient de dépasser le Java et continue son ascension.

#### 3.2. Le Framework

Les différents frameworks les plus utilisé dans la programmation web en python sont Django, Flask et Pyramide.

Django est l'un des framework le plus mature pour Python. Ses règles de conception se base sur la réduction du temps de développement d'application web. Les fonctionnalités fournies par Django permettent aux développeurs de créer rapidement des applications Web personnalisées en fonction des différentes exigences commerciales. Django fournit également des fonctionnalités de sécurités robuste. Enfin, Django a été conçu de manière à pouvoir gérer tout type de matériel. Cet avantage est la principale raison pour laquelle les sites les plus fréquenté du monde utilise le framework comme Django.

Cependant, Django impact grandement les performances des petites applications web et pose des problèmes quand au processus qui ne peuvent gérer plusieurs requêtes en simultanés.

Flask offre une certaine simplicité, flexibilité et contrôle sur notre projet. Il est recommandé pour les projets de moyenne envergure, mais il est moins utilisé que Django. Flask aider à la



construction d'une base solide pour des application web. Il s'agit du choix par défaut pour tout projet web qui ne correspond pas à Django.

Enfin, Pyramid est une application web basé sur Python, et le second framework le plus populaire. Son but est de faire le plus possible avec le minimum de complexité. La caractéristique frappante de Pyramid est sa capacité à bien fonctionner avec de petites et grandes applications.

Pour ce projet nous allons utiliser Django grâce à la diversité de ses fonctionnalités.

### 3.3. La DataBase

Les bases de données les plus populaire sont MySQL, PostgreSQL et Microsoft SQL server. Cependant la base de données la plus apprécié par les développeurs parmi ces trois-ci est PostgreSQL.

MySQL est gratuit et très puissant. Il possède la double licence GPL et possédé par Oracle. Il répond à une logique client/serveur, c'est-à-dire que plusieurs clients peuvent se connecter sur un seul serveur qui héberge les données.

PostgreSQL est la base de données à utiliser pour les projets. Stable et puissant, il permet de gérer des Gigabytes des données sans problème.

Mycrosoft SQL Server est un produit Mycrosoft qui ne tourne que sur un OS Windows. Il est payant et n'apporte rien de plus que les logiciels concurrents libre de droit. Il sera donc à écarter pour sa portabilité limitée à un seul OS.

Concernant MySQL et PostgreSQL, on préféra MySQL pour des projet plus modeste où le nombre d'utilisateur est faible avec un petit volume de données. Par contre pour des projets plus importants avec des données dépassant la dizaine de téraoctets, c'est lui le meilleur. Cependant, il faut optimiser la configuration, ce qui demande un certain travail.

Nous allons travailler avec PostgreSQL en prévision du possible essor du client qui souhaite s'étendre à l'avenir.

### 3.4. L'hébergement

Concernant l'hébergement, un serveur chez un grand hébergeur comme OVH ou IBM nous garantira de bonne performance et une bonne qualité d'exploitation.

OVH est l'un des plus grands hébergeurs en France, ce qui pourrait faciliter la communication avec l'équipe technique en cas de soucis.

IBM est une entreprise reconnu dans le monde. Cependant le prix est fonction du temps d'utilisation et pourrait revenir plus chère.



Nous choisirons OVH comme hébergeur qui, pour une même configuration machine, reviendra moins chère qu'IBM.

### 3.5. Support de l'application

Concernant les supports, nous devons faire en sorte que l'application soit responsive. Cela signifie qu'elle devra être disponible sur tous les supports possibles afin de permettre à chacun de pouvoir y accéder.

Les caissiers auront besoin d'une tablette afin de prendre les commandes des clients en salle ou au téléphone.

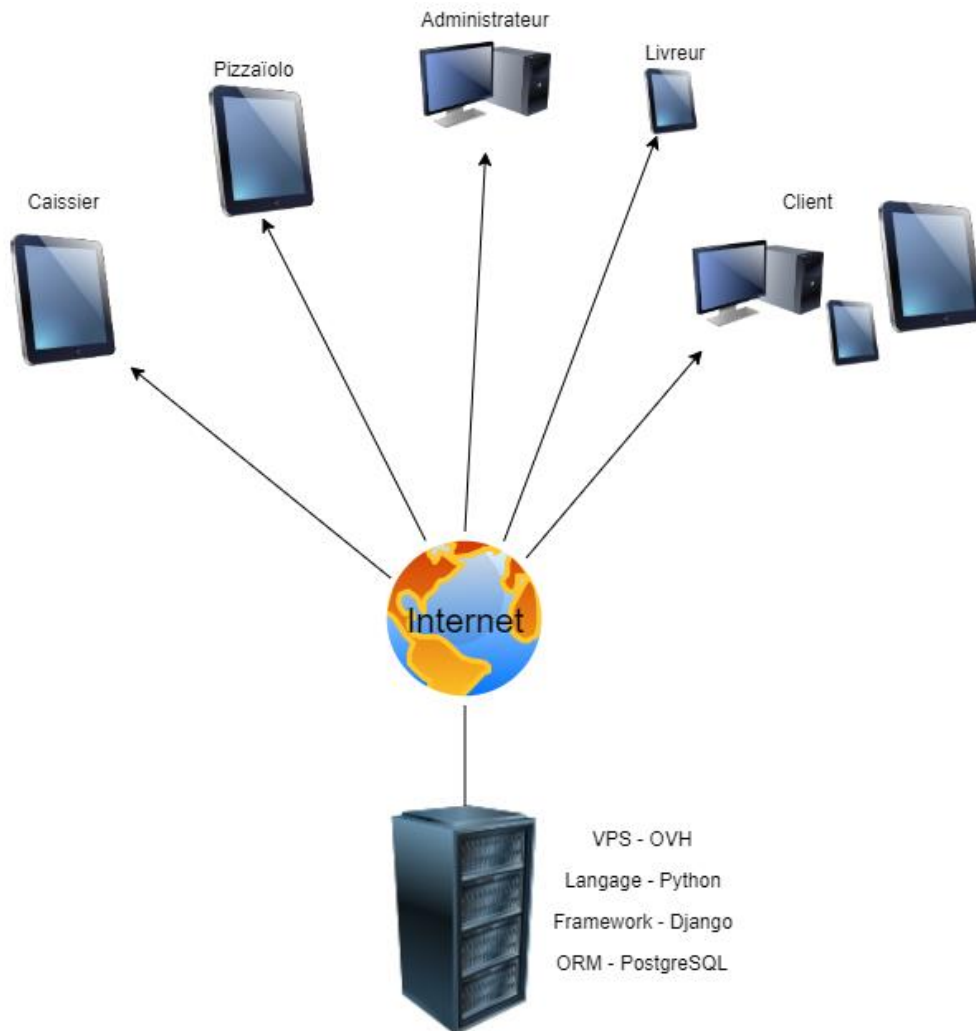
Les clients devront pouvoir utiliser leurs ordinateurs, leurs téléphones ou leurs tablettes afin de passer commande et de suivre le statut de leurs commandes.

Le pizzaïolo devra pouvoir accéder à la liste des commandes et des recettes depuis une tablette.

L'administrateur doit pouvoir y accéder avec un ordinateur et le livreur depuis son téléphone afin d'accéder aux commandes à livrer et les informations de livraison et de paiement.



### 3.6. Schéma solution technique





#### 4. Annexe

##### 1. Explication textuelle des cas d'utilisation

###### Cas des Clients

###### Cas n° 1.1

Nom : Se connecter.

Acteurs(s) : Client.

Pré-conditions : L'utilisateur a déjà un compte.

Scénario nominal (main success scénario) :

1. L'utilisateur accède à la page de connexion du système ;
2. L'utilisateur saisit ses informations (e-mail/pseudo, password) ;
3. Le système vérifie les informations de l'utilisateur ;
  1. Le système valide les informations de l'utilisateur ;
  2. Le système invalide les informations. Voir point d'extension ;

Extensions : S'inscrire

4. Le système invalide les informations de l'utilisateur ;
5. Le système propose à l'utilisateur de s'inscrire ou de commander par téléphone ;
6. L'utilisateur saisit ses informations (e-mail, pseudo, password) ;
7. Le système enregistre les informations de l'utilisateur ;
8. Le système envoie un e-mail de confirmation à l'utilisateur ;
9. L'utilisateur confirme son adresse mail.

Fins : L'utilisateur est connecté.

###### Cas n° 1.2

Nom : Créer une commande.

Acteurs(s) : Client

Pré-conditions : L'utilisateur est connecté

Scénario nominal (main success scénario) :

1. L'utilisateur parcourt la liste de produits disponible à la commande ;
2. L'utilisateur sélectionne un ou des produits ;
3. Le système enregistre les produits sélectionnés ;
4. Le système ajoute un aperçu de la commande avec le montant ;
5. L'utilisateur valide sa commande.

Extensions :

6. Le système demande à l'utilisateur de saisir ses informations (adresse, numéro de tel, nom, prénom, moyen de paiement) ;
7. L'utilisateur saisit ses informations ;
8. L'utilisateur choisit le moyen de paiement :
  1. L'utilisateur choisit le paiement en espèce ;
    1. Le système enregistre le paiement en espèce à la livraison ;
  2. L'utilisateur choisit le paiement en CB ;
    1. L'utilisateur saisit ses informations bancaires ;
    2. Le système valide les informations et le paiement.



Fins :

Le système valide le paiement et l'enregistrement de la commande.

Cas n° 1.3

Nom : Modifier commande.

Acteurs(s) : Client.

Pré-conditions : L'utilisateur est connecté, l'utilisateur à passer une commande, la commande n'est pas en préparation.

Scénario nominal (main success scénario) :

1. L'utilisateur consulte les produits de sa commande en cours ;
2. L'utilisateur souhaite :
  1. Modifier ses informations de livraison (adresse, numéro de téléphone) ;
  2. Modifier ses informations de paiement (espèce <=> CB) ;
  3. Changer un produit ;
3. Le système enregistre les modifications.

Extensions :

4. L'utilisateur annule sa commande ;
5. Le système demande confirmation de l'annulation ;
6. L'utilisateur confirme la demande d'annulation ;
7. Le système enregistre l'annulation.
8. Le système procède au remboursement si le paiement fut en CB.

Fins :

La commande est annulée.

Cas n° 1.4

Nom : Consulter liste de commande.

Acteurs(s) : client.

Pré-conditions : L'utilisateur est connecté, l'utilisateur a déjà commandé.

Démarrage :

Scénario nominal (main success scénario) :

1. L'utilisateur affiche l'historique de ses commandes ;
2. L'utilisateur peut noter la qualité des produits d'une commande antérieur ;
  1. Le système enregistre la note attribuée par l'utilisateur ;
3. L'utilisateur peut récupérer une facture d'une commande antérieure ;

L'utilisateur peut laisser un commentaire sur une commande.

Cas de l'administrateur

Cas n° 2.1

Nom : Se connecter

Acteurs(s) : Administrateur.

Pré-conditions :

Scénario nominal (main success scénario) :

1. L'utilisateur saisit ses informations ;
2. Le système vérifie les informations saisit ;



3. Le système valide les informations saisit ;

4. L'utilisateur est connecté.

Fins : L'utilisateur est connecté.

#### Cas n° 2.2

Nom : Consulter les commandes

Acteurs(s) : Administrateur.

Pré-conditions :

Démarrage :

Scénario nominal (main success scénario) :

1. L'utilisateur choisit d'afficher les commandes en cours, les commande en préparation, les commande en livraison ou les commandes livré ;
2. Le système affiche les éléments demandés par l'utilisateur ;
3. L'utilisateur peut supprimer les commentaires laisser par des clients ;

Fins :

#### Cas n° 2.3

Nom : Consulter les stocks d'ingrédients.

Acteurs(s) : Administrateur.

Pré-conditions : L'utilisateur est connecté.

Scénario nominal (main success scénario) :

1. L'utilisateur consulte l'état des stocks d'ingrédients ;
2. L'utilisateur peut cibler les stocks d'un restaurant en particulier ;
3. L'utilisateur peut commander des ingrédients pour réapprovisionner un/des restaurant(s).

Fins :

#### Cas n° 2.4

Nom : Gestion compte employé.

Acteurs(s) : Administrateur.

Pré-conditions : L'utilisateur est connecté.

Scénario nominal (main success scénario) :

1. Le système demande si l'utilisateur souhaite créer, supprimer ou modifier un compte employé ;
  1. L'utilisateur veut créer un compte employé :
    1. L'utilisateur saisit les informations de l'employé (e-mail, nom, prénom) ;
    2. Le système demande confirmation de la création ;
    3. L'utilisateur confirme la création ;
    4. Le système enregistre les informations ;
    5. Le système génère un password aléatoire ;
    6. Le système envoie un mail à l'e-mail renseigné.
  2. L'utilisateur veut modifier un compte employé :
    1. L'utilisateur sélectionne le compte à modifier ;
    2. L'utilisateur sélectionne le(s) information(s) à modifier ;
    3. L'utilisateur modifie le(s) information(s) ;



4. Le système enregistre les informations.
3. L'utilisateur veut supprimer un compte employé :
  1. L'utilisateur sélectionne le compte à supprimer ;
  2. Le système demande confirmation de la suppression du compte ou de passer le compte en compte client ;
  3. L'utilisateur confirme la suppression du compte ou le passage en compte client ;
  4. Le système enregistre les informations.

Fins : Le compte est créé, modifié ou supprimé.

Cas Pizzaïolo

Cas n° 3.1

Nom : Se connecter.

Acteurs(s) : Pizzaïolo

Scénario nominal (main success scénario) :

1. L'utilisateur saisit ses informations ;
2. Le système vérifie les informations saisit ;
3. Le système valide les informations saisit ;
4. L'utilisateur est connecté.

Fins : L'utilisateur est connecté.

Cas n° 3.2

Nom : Consulter la liste des commandes.

Acteurs(s) : Pizzaïolo.

Pré-conditions : L'utilisateur doit être connecté.

Démarrage :

Scénario nominal (main success scénario) :

1. L'utilisateur affiche les commandes en attente ;
2. L'utilisateur sélectionne une commande en attente
3. L'utilisateur vérifie les commentaires laissés par le client.

Extensions :

4. Point d'extension : Valider la mise en préparation de la commande :
  1. Le système enregistre la mise en préparation de la commande.
  2. Le système envoie une notification au client de la commande.
5. Point d'extension : Consulter les recettes :
  1. L'utilisateur sélectionne la recette qu'il souhaite consulter ;
  2. Le système affiche la recette.
6. Point d'extension : Valider la fin de préparation de la commande :
  1. Le système enregistre la fin de la préparation de la commande.

Fins:

La commande est préparée et en attente de livraison.

Cas du Livreur

Cas n° 4.1

Nom: Se connecter.





Acteurs(s): Livreur

Démarrage:

Scénario nominal (main success scénario):

1. L'utilisateur saisit ses informations;
2. Le système vérifie les informations saisit;
3. Le système valide les informations saisit;
4. L'utilisateur est connecté.

Fins:

L'utilisateur est connecté.

Cas n° 4.2

Nom : Consulter les commandes à livrer.

Acteurs(s) : Livreur.

Pré-conditions : L'utilisateur est connecté.

Démarrage :

Scénario nominal (main success scénario) :

1. L'utilisateur affiche les commandes en attentes de livraison ;
2. L'utilisateur sélectionne les commandes ;
3. Le système affiche les informations de livraison de la commande.

Extensions :

4. Valider la mise en livraison de la commande :
  1. Le système enregistre la mise en livraison ;
  2. Le système envoie une notification au client de la commande.
5. Valider le paiement en espèce :
6. Le système enregistre le paiement de la commande ;
7. Valide la livraison de la commande :
8. Le système enregistre la livraison.

Fins :

La commande est livrée.

Cas du caissier

Cas n° 5.1

Nom : Se connecter.

Acteurs(s) : Caissier.

Pré-conditions :

Scénario nominal (main success scénario) :

1. L'utilisateur saisit ses informations ;
2. Le système vérifie les informations saisit ;
3. Le système valide les informations saisit ;
4. L'utilisateur est connecté.

Fins : L'utilisateur est connecté.

Cas n° 4.1

Nom : Créer commande

Acteurs(s) : Caissier.



Auteur : Linocent Timothée

Pré-conditions : L'utilisateur est connecté.

Scénario nominal (main success scénario) :

1. L'utilisateur enregistre les produits commandés sur place ou par téléphone ;
2. L'utilisateur crée la commande ;
3. L'utilisateur enregistre les informations du client :
  1. L'adresse, nom, numéro de téléphone pour les livraisons ;
  2. Le mode de paiement espèce ou carte bleue
4. Le système enregistre la commande ;
5. Le système ajoute la commande à la liste de commande.

Extensions :

6. Payer la commande :
  1. L'utilisateur renseigne le si le paiement a été effectué et si c'est en espèce ou carte CB.
  2. Le système enregistre les informations.

Fins: La commande est créée.

## 2. Solution envisagées

Source pour le langage :

<https://fr.yeePLY.com/blog/langages-programmation-type-developpement/>

<https://www.genie-inc.com/langage-de-programmation-choisir/>

<https://espresso-jobs.com/conseils-carriere/quel-langage-de-programmation-choisir/>

<https://www.apollo-formation.com/developpement/langages-programmation-frameworks-base-de-donnees-2019.html>

Source pour le framework :

<https://waytolearnx.com/2020/02/10-meilleurs-frameworks-de-developpement-web.html#:~:text=Django%20est%20sans%20doute%20le,Python%20pour%20le%20d%C3%A9veloppement%20web.>

<https://www.codeur.com/blog/meilleurs-frameworks-python/>

<https://www.codeur.com/blog/meilleurs-frameworks-python/>

Source pour la database :

<https://www.apollo-formation.com/developpement/langages-programmation-frameworks-base-de-donnees-2019.html>