

Déterminer les acteurs et décrire leurs rôles respectifs.

Acteurs internes (utilisateurs du système)

- **Opérateurs colis** : Ils traitent manuellement les courriers reçus, gèrent les clients, les commandes, le conditionnement, l'envoi d'emails personnalisés et les statistiques
- **Opérateurs stock** : Ils sont chargés de la gestion des stocks de « goodies » (cadeaux) et de la réalisation des inventaires
- **Administrateur du système** : Il est responsable du paramétrage de l'application. Ses tâches incluent la gestion des listes d'emballages, la configuration des données pour le calcul des frais d'affranchissement et la mise à jour des poids des articles

Acteurs externes (utilisateurs du système)

- **Client final (particulier)** : Il initie le processus en collectant les points de fidélité sur les emballages, en consultant le site vitrine pour choisir ses cadeaux et en envoyant sa demande (points, chèque, choix des cadeaux) par courrier à la fromagerie
- **La Poste** : Il s'agit du partenaire externe exclusif pour l'envoi des colis, il permet la gestion des tarifs.
- **Enseignes / Grands distributeurs** : Dans le futur, il pourront être utilisés pour les statistiques des ventes et le nombre de clients collectant des points.

Réaliser un schéma global de l'architecture du système (acteurs et grandes fonctionnalités) avec les interactions des acteurs.

grandes fonctionnalités

1. Gestion des clients
2. Gestion des commandes & colis
3. Gestion du conditionnement & affranchissement
4. Gestion des stocks
5. Administration du système
6. Communication & reporting

schéma

graph LR

```

subgraph SYS["Gestion globale des colis"]
    direction LR

    Cl(Client final)
    OC(Opérateurs colis)
    OS(Opérateurs stock)
    Ad(Administrateur système)
    LP(La Poste)
    Di(Enseignes / Distributeurs)

    GCC[**Gestion des clients et des commandes**
    - Fiche client et listes
    - Suivi des commandes
    - Lettre suivie]
    CL[**Conditionnement et affranchissement**
    - Calcul d'affranchissement
    - Algorithme de conditionnement
    - Gestion des emballages]
    CM[**Communication et marketing**
    - Mailing et emails
    - Newsletter]
    StPi[**Statistiques et pilotage**
    - Statistiques
    - Impressions]
    SuPa[**Fonctionnalités de support et paramétrage**
    - Emplacements
    - Administration]

    Cl --> GCC
    Cl --> CM

    OC --> GCC
    OC --> CL
    OC --> CM
    OC --> StPi

    OS --> CL
    OS --> SuPa

    Ad --> SuPa

    CL --> LP
    Di --> SuPa
end

```

```

classDef acteur fill:#E3F2FD,stroke:#1E88E5,stroke-width:2px;
classDef fonction fill:#E8F5E9,stroke:#2E7D32,stroke-width:2px;

class Cl,OC,OS,Ad,LP,Di acteur;
class GCC,CL,CM,StPi,SuPa fonction;

subgraph LEG["Légende"]
  A[Acteur]:::acteur
  F[Fonctionnalité du système]:::fonction
end

```

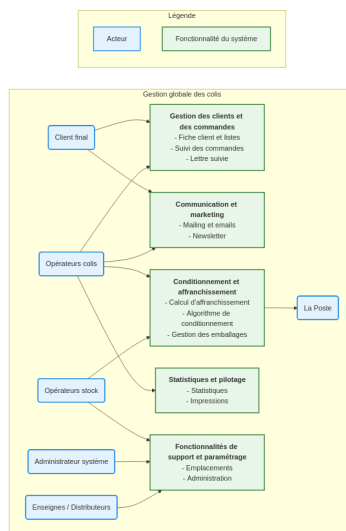


Figure 1: diag1

Représenter les acteurs et le système sous forme de diagrammes de UCs

```

@startuml
left to right direction
skinparam packageStyle rectangle

actor "Client final" as Client
actor "Opérateur colis" as OpColis
actor "Opérateur stock" as OpStock
actor "Administrateur" as Admin
actor "La Poste" as Poste

Client --> UC1[Gestion des clients et des commandes]
OpColis --> UC1
OpColis --> UC2[Communication et marketing]
OpColis --> UC3[Conditionnement et affranchissement]
OpStock --> UC3
OpStock --> UC4[Statistiques et pilotage]
Admin --> UC5[Fonctionnalités de support et paramétrage]
Enseignes --> UC5
UC3 --> Poste

```

```

rectangle "Système de gestion de colis fidélité" {

    ' --- Client ---
    usecase "Commander des cadeaux\npar courrier" as UC_Commande
    usecase "Recevoir des informations\n(client)" as UC_Info

    ' --- Commandes ---
    usecase "Traiter le courrier client" as UC_Courrier
    usecase "Gérer les clients" as UC_Client
    usecase "Gérer les commandes" as UC_GCommande
    usecase "Conditionner les colis" as UC_Cond
    usecase "Calculer l'affranchissement" as UC_Aff
    usecase "Envoyer le colis" as UC_Envoi
    usecase "Suivre le colis" as UC_Suivi

    ' --- Détails conditionnement ---
    usecase "Calculer le poids total\n(colis + objets)" as UC_Poids
    usecase "Choisir l'emballage optimal" as UC_Emballage

    ' --- Stocks ---
    usecase "Gérer le stock de goodies" as UC_Stock
    usecase "Réaliser les inventaires" as UC_Inventaire

    ' --- Communication ---
    usecase "Gérer la communication client" as UC_Com
    usecase "Envoyer des emails\npersonnalisés" as UC_Email
    usecase "Générer mailing papier" as UC_Mail

    ' --- Statistiques ---
    usecase "Consulter les statistiques" as UC_Stats
    usecase "Imprimer documents" as UC_Print
    usecase "Imprimer liste clients" as UC_PrintClient
    usecase "Imprimer liste goodies" as UC_PrintGoodies
    usecase "Imprimer commandes client" as UC_PrintCmd
    usecase "Imprimer rapports statistiques" as UC_PrintStats

    ' --- Paramétrage ---
    usecase "Paramétrer le système" as UC_Admin
}

Client --> UC_Commande
Client --> UC_Info

OpColis --> UC_Courrier
OpColis --> UC_Client
OpColis --> UC_GCommande

```

```

OpColis --> UC_Com
OpColis --> UC_Stats
OpColis --> UC_Print

OpStock --> UC_Stock
OpStock --> UC_Inventaire

Admin --> UC_Admin

Poste --> UC_Aff
Poste --> UC_Envoi
Poste --> UC_Suivi

UC_GCommande --> UC_Cond : <<include>>
UC_GCommande --> UC_Aff : <<include>>
UC_GCommande --> UC_Envoi : <<include>>
UC_GCommande --> UC_Suivi : <<include>>

UC_Cond --> UC_Poids : <<include>>
UC_Cond --> UC_Emballage : <<include>>

UC_Com --> UC_Email : <<extend>>
UC_Com --> UC_Mail : <<extend>>

UC_Print --> UC_PrintClient : <<extend>>
UC_Print --> UC_PrintGoodies : <<extend>>
UC_Print --> UC_PrintCmd : <<extend>>
UC_Print --> UC_PrintStats : <<extend>>

@enduml

```

Décrire (textuellement) le scénario de la gestion des colis et le représenter à l'aide des diagrammes de séquence et d'activité.

Description

Le scénario de la gestion des colis au sein de la fromagerie DIGICHEESE suit un processus précis, allant de la réception de la demande du client jusqu'au suivi de l'expédition. Voici les étapes textuelles de ce scénario : 1. Initialisation par le client Le processus débute hors système : le client final collecte ses points de fidélité, choisit ses cadeaux (« goodies ») sur le site vitrine et envoie par courrier postal son dossier comprenant les points, le chèque de règlement des frais de port et sa liste de cadeaux . 2. Réception et Saisie de la commande À la

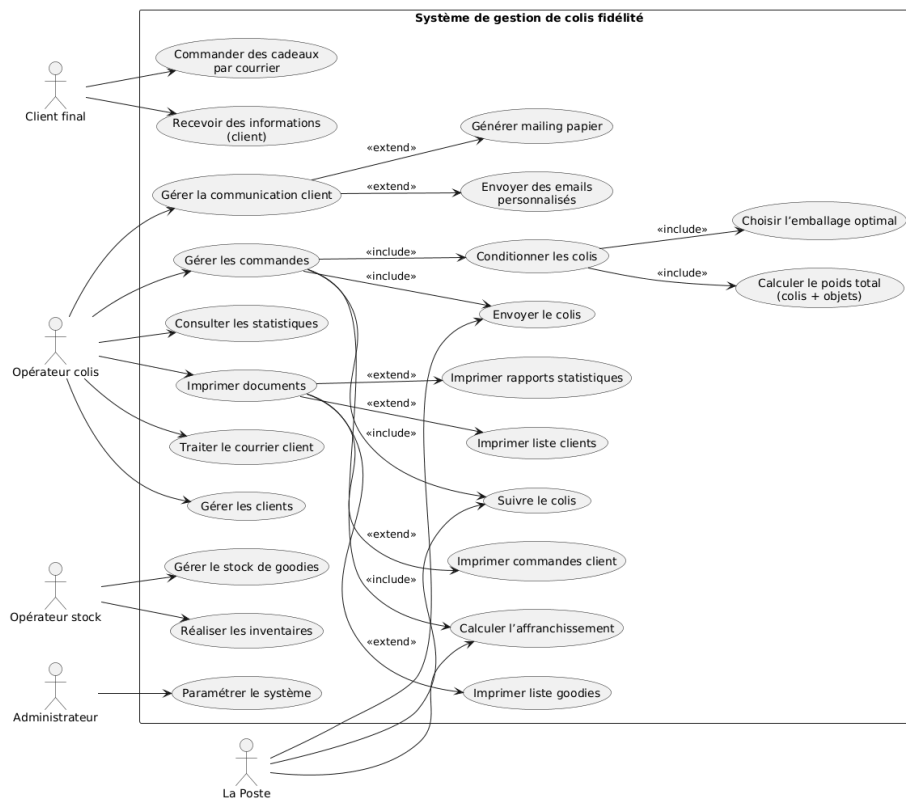


Figure 2: diag2

réception du courrier, un opérateur colis traite la demande manuellement . Il utilise l'application pour : • Rechercher ou créer la fiche client (nom, adresse, code postal et ville sont obligatoires) . • Saisir la commande en sélectionnant les objets demandés par le client . 3. Détermination du conditionnement et calcul des frais Une fois les articles saisis, le système intervient pour la partie logistique : • Algorithme de conditionnement : L'application s'appuie sur un calcul complexe (utilisant les quantités minimales et maximales paramétrées par l'administrateur) pour identifier automatiquement le meilleur emballage adapté à la commande . • Calcul d'affranchissement : Le système calcule automatiquement le montant des frais de port en combinant le poids total (articles + emballage choisi) et les tarifs postaux de La Poste mis à jour annuellement . 4. Expédition et Suivi (Lettre Suivie) Le colis est ensuite confié à La Poste pour l'envoi physique . • Numéro de suivi : Au moment du dépôt, La Poste fournit un numéro de suivi . • Mise à jour du système : L'opérateur reporte manuellement ce numéro dans la zone « commentaires » de la commande dans l'application . 5. Communication et Clôture • Notification client : Si le client a renseigné une adresse électronique, l'opérateur déclenche manuellement l'envoi d'un email personnalisé . • Mailing papier : Pour les clients sans email, le système peut générer un fichier texte pour une procédure de mailing papier . • Historisation : Le mouvement du colis est enregistré dans l'historique des commandes, ce qui permet ultérieurement à la direction de consulter des statistiques d'activité .

Diagramme de séquence

sequenceDiagram

actor Client

participant OpColis as Opérateur colis

participant Sys as Système

participant Stock as Stock goodies

participant Poste as La Poste

Client ->> OpColis: Envoi courrier (points + choix + chèque)

OpColis ->> Sys: Saisir client et commande

Sys ->> Sys: Vérifier points fidélité

alt Points suffisants

Sys ->> Stock: Vérifier disponibilité des goodies

alt Stock disponible

Sys ->> Sys: Calcul conditionnement

Sys ->> Sys: Calcul affranchissement

OpColis ->> Poste: Dépôt du colis

Poste -->> OpColis: Numéro de suivi

OpColis ->> Sys: Saisir numéro de suivi

opt Client avec email

OpColis ->> System: Déclencher email

```

        System -->> Client: Notification
    end

    Poste ->> Client: Livraison

else Stock insuffisant
    Sys ->> OpColis: Alerte stock insuffisant
    OpColis ->> Sys: Mise en attente / action corrective
end
else Points insuffisants
    Sys ->> OpColis: Alerte points insuffisants
    OpColis ->> Sys: Rejet ou demande complément
end
end

```

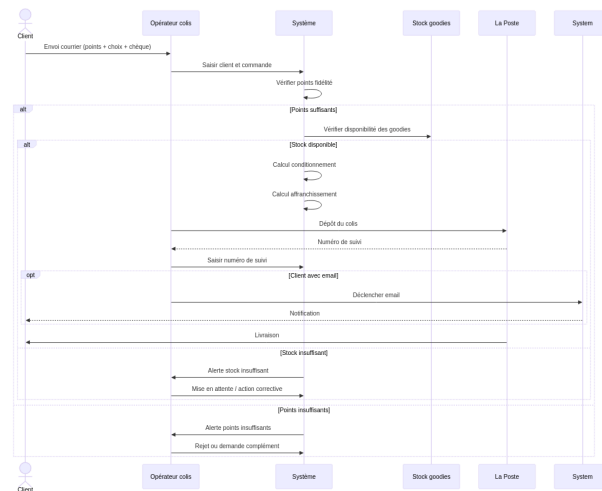


Figure 3: diag3

Diagramme d'activité

stateDiagram-v2

[*] --> EnvoiCourrier

EnvoiCourrier --> ReceptionCourrier

ReceptionCourrier --> SaisieCommande

SaisieCommande --> VerifPoints

VerifPoints --> PointsOK: points suffisants

VerifPoints --> PointsKO: points insuffisants


```

PointsKO --> ActionPoints
ActionPoints --> [*]

PointsOK --> VerifStock

VerifStock --> StockOK: stock disponible
VerifStock --> StockKO: stock insuffisant

StockKO --> ActionStock
ActionStock --> [*]

StockOK --> Conditionnement
Conditionnement --> CalculAffranchissement
CalculAffranchissement --> DepotColis

DepotColis --> NumeroSuivi
NumeroSuivi --> SaisieSuivi

SaisieSuivi --> DecisionEmail

DecisionEmail --> EnvoiEmail: client avec email
DecisionEmail --> FinTraitement : client sans email

EnvoiEmail --> FinTraitement
FinTraitement --> [*]

state ActionPoints {
    [*] --> DemandeComplementPoints
    DemandeComplementPoints --> [*]
}

state ActionStock {
    [*] --> MiseEnAttente
    MiseEnAttente --> [*]
}

```

Créer le diagramme de classe permettant de réaliser la partie gestion des colis (recueillir les data nécessaires pour le réaliser).

```

classDiagram

class Client {
    idClient

```

Figure 4: diag4

```

nom
prenom
adresse1
adresse2
adresse3
codePostal
ville
email
tel
newsletter
}

```

```
class Commande {
    idCommande
    dateCommande
    commentaire
    archive
}
```

```
class Colis {
    idColis
    poidsTotal
}
```

```
class LigneCommande {
```

```

    quantite
}

class Objet {
    idObjet
    libelle
    poidsUnitaire
    points
    indisponible
}

class Conditionnement {
    idConditionnement
    libelle
    poidsEmballage
}

class RegleConditionnement {
    quantiteMin
    quantiteMax
}

class TarifPostal {
    poidsMin
    montant
}

class Suivi {
    numeroSuivi
}

class Enseigne {
    idEnseigne
    nom
    ville
}

Client "1" --> "0..*" Commande : passe
Commande "1" --> "1" Colis : génère
Commande "1" --> "1..*" LigneCommande
LigneCommande "1" --> "1" Objet

Colis "1" --> "1" Conditionnement : utilise
Colis "1" --> "1" TarifPostal : applique
Colis "1" --> "0..1" Suivi : possède

```

Objet "1" --> "0..*" RegleConditionnement
Conditionnement "1" --> "0..*" RegleConditionnement

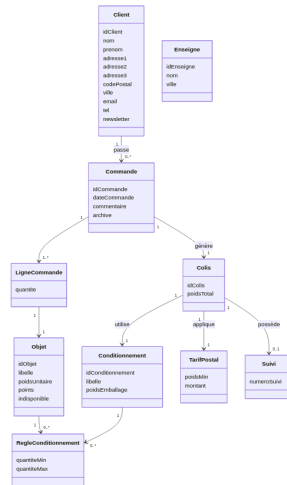


Figure 5: diag5