

Contexte

DIGICHEESE est une fromagerie familiale disposant d'un programme de fidélité via de la collecte de vignette et de points clients. Ces derniers peuvent être utilisés par le client pour commander des articles.

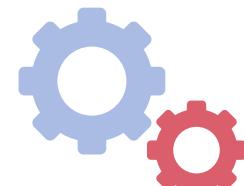
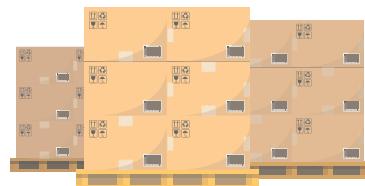
Actuellement DIGICHEESE, dispose d'un interface développée via ACCESS 2000. Elle est très peu ergonomique, beaucoup de bugs sont présent .

L'idée, tout en gardant ces fonctionnalité de gestion serait de créer une solution plus moderne, ergonomique et synthétique.

Architecture système

Les acteurs

- Administrateur système
- Opérateur de colis
- Opérateur de stock
- Client



Objectif global

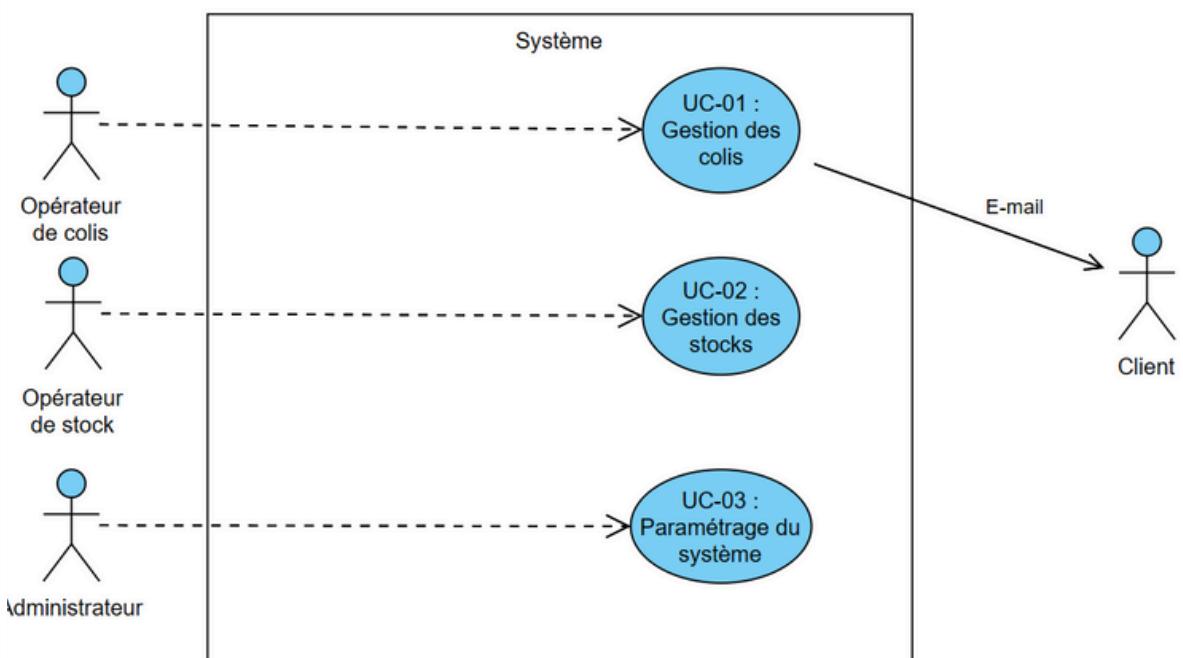
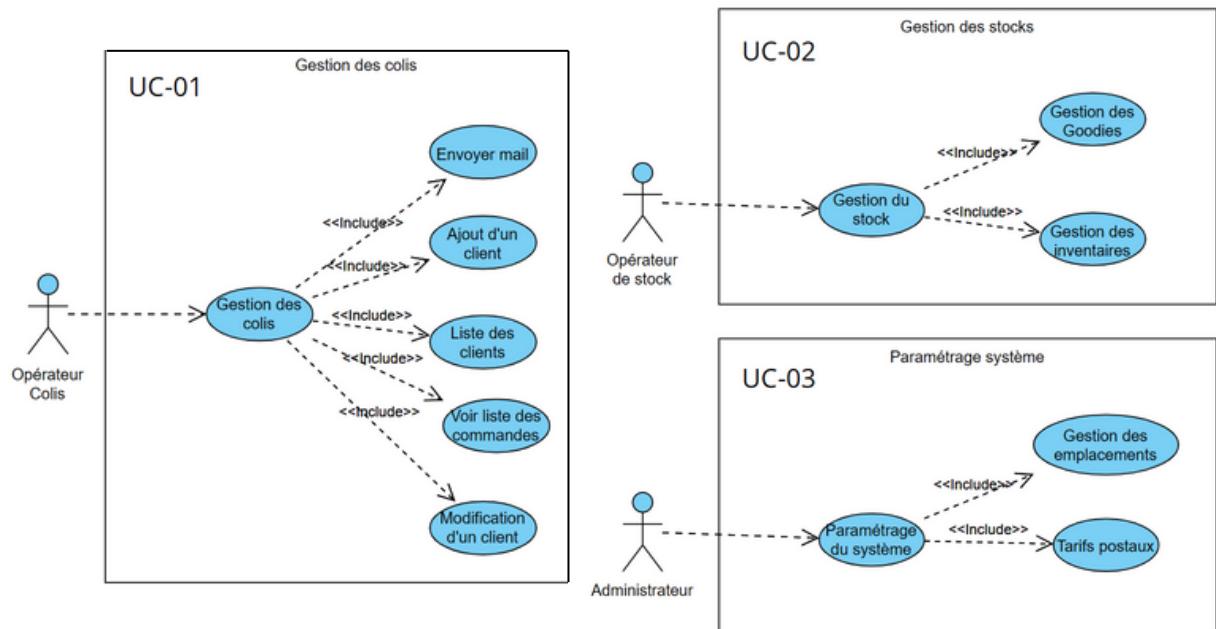
Prendre en compte ces différents acteurs dans le but de créer une interface multi-utilisateur

Pourquoi ?

Une fois les acteurs précisément définis, Le fait de dissocier chaque rôle pour un avoir un compte avec un affichage différent sera plus efficace pour chacun d'entre eux.

Une application moderne, est plus agréable à utiliser au vu de sa rapidité, mais également des modules de développement actuels permettant aux développeur de rendre l'interface plus fluide qu'il y a 20 ans.

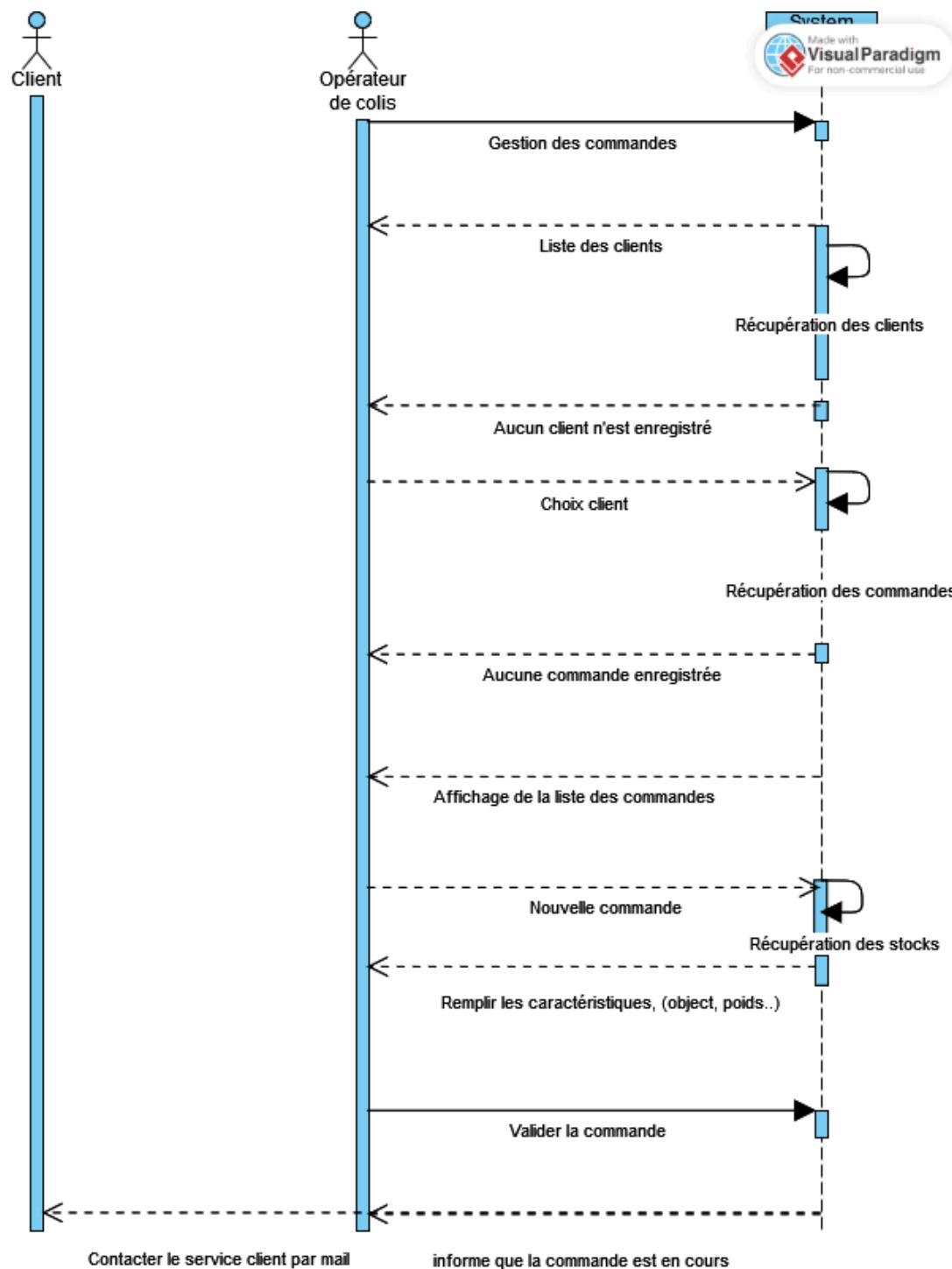
Cas d'utilisation



Ces 3 sous-Uses Case, montre pour chaque acteur les détails des actions possibles et les conditions (<<includes>>).

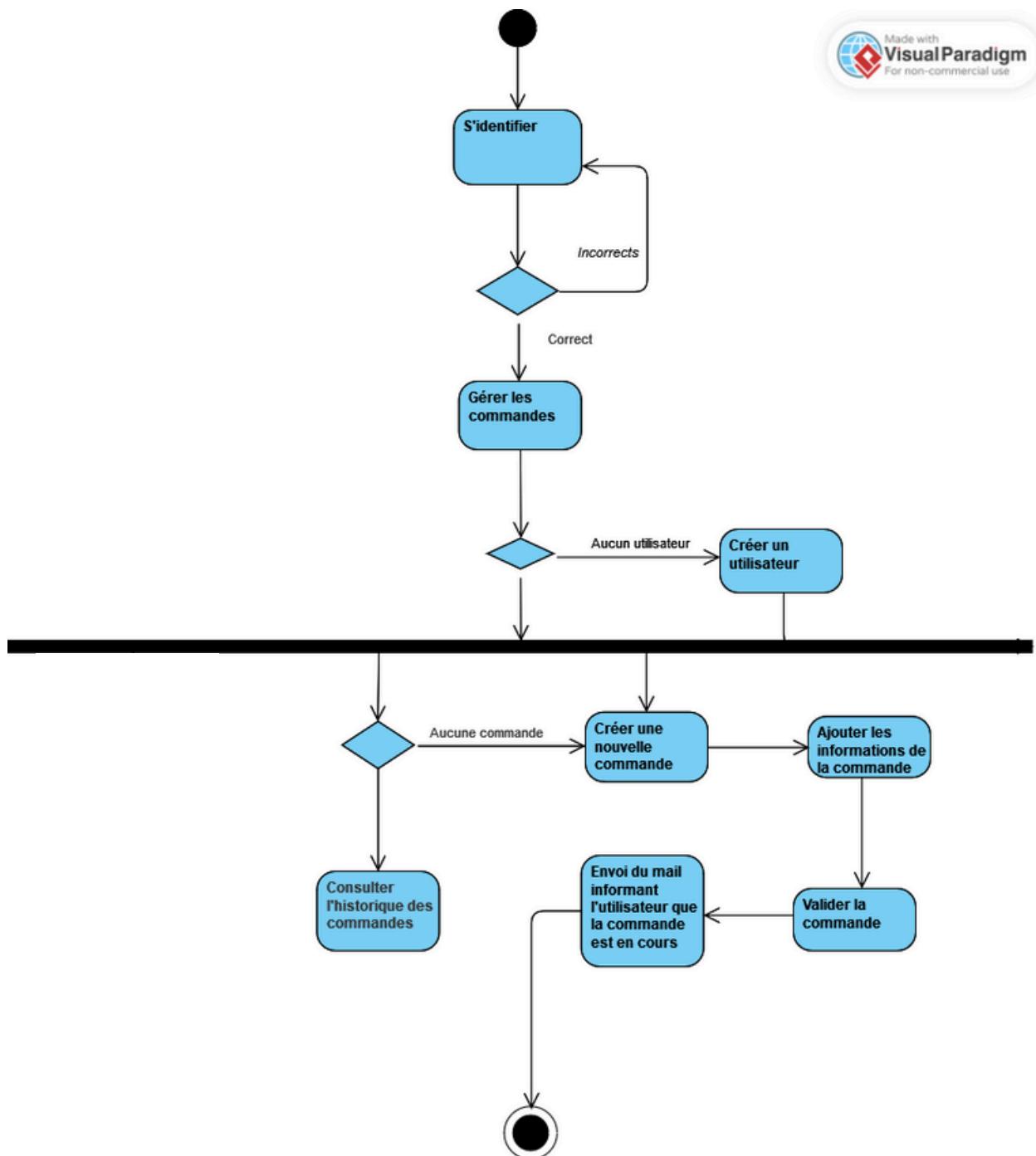
Le grand, est le use case du système réunissant les trois autres donnant une vue globale.

Diagramme de séquences



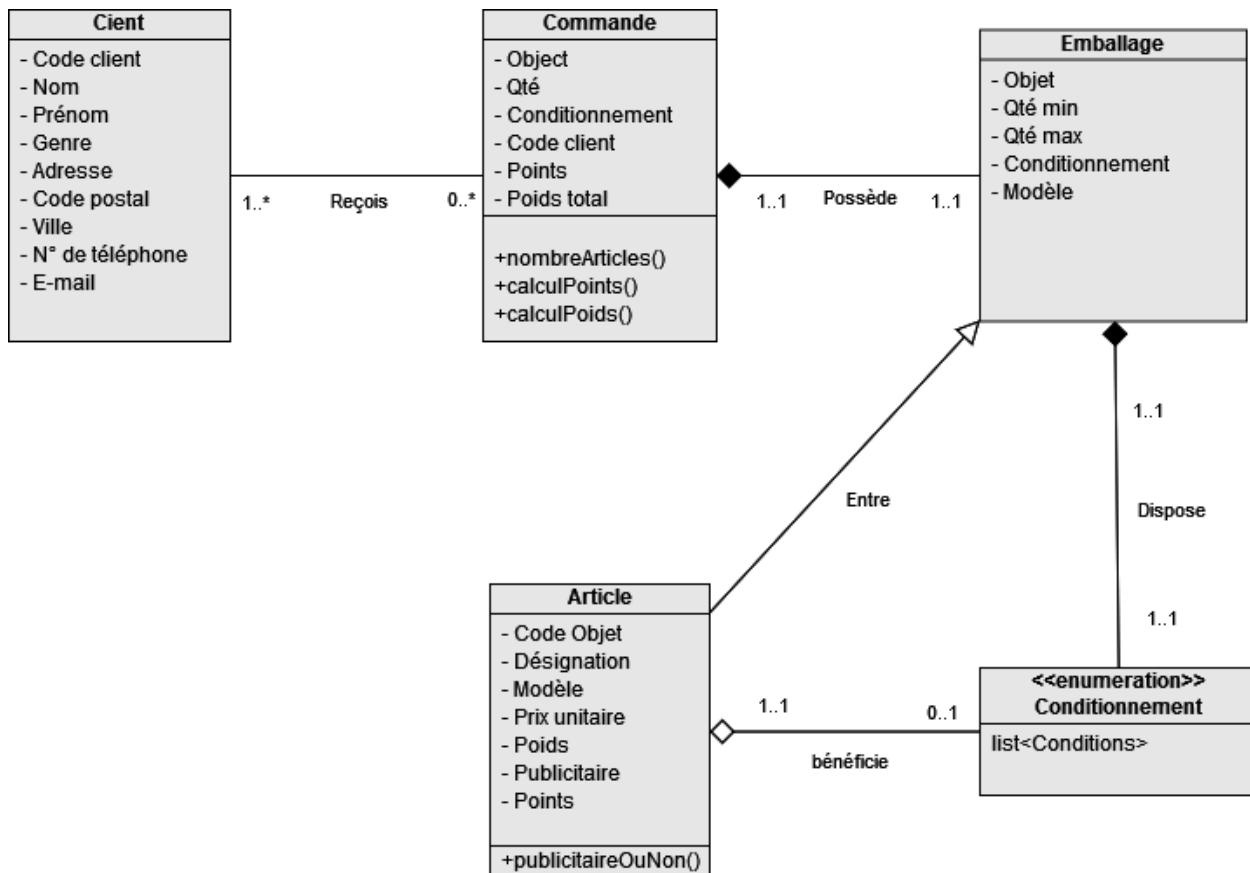
Après avoir eu une vue globale du système avec les *Uses cases*, ce diagramme de séquence nous donne des détails plus précis quant aux fonctionnements de l'application et les réponses du système de manière chronologique concernant chaque actions lié à la gestion des commandes.

Diagramme d'activité

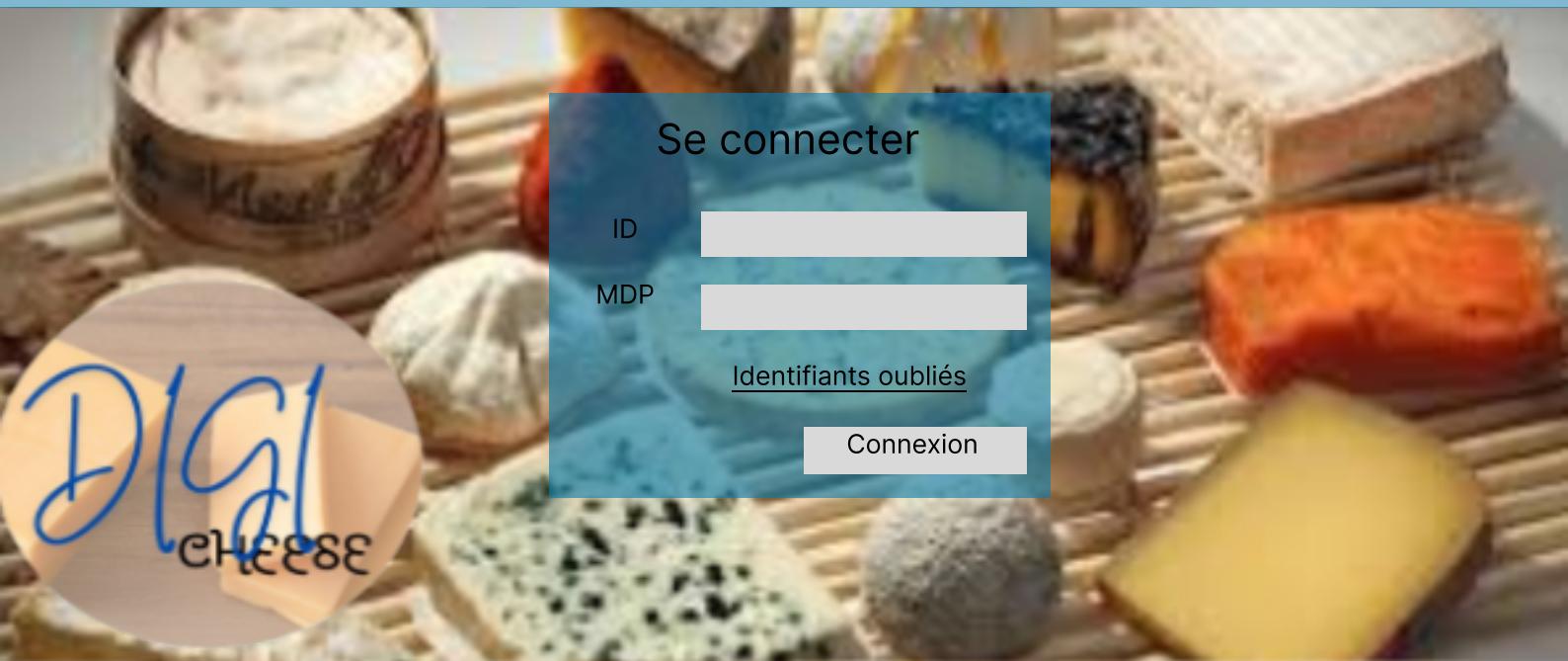


Ce diagramme d'activité précise les actions de l'opérateur de colis selon ses choix, et les conditions du système (ex : utilisateur existant ou non) la barre de synchronisation converge les actions de l'ensemble des étapes de la première activité (client) pour enchaîner sur la deuxième (colis)

Diagramme de classes



Un client peut avoir une, plusieurs ou aucunes commandes. Une commande possède un emballage. Un Emballage peut contenir plusieurs articles mais minimum un, nous retrouvons donc une notion d'héritage. Un emballage est conditionné. Les conditionnements sont dissociés dans une classe d'énumération permettant de lister les différents types de conditonnements existants. Un article en lui même peut être conditionné ou non, (à dissocié de celui de l'emballage). Ce dernier peut contenir un nombre indéterminé d'articles.



Nouvelle commande	<input checked="" type="checkbox"/>
Object	<input type="text"/>
Quantité	- <input type="button"/> +
Conditionnement	<input type="button"/>
Code client	<input type="text"/>
Points	- <input type="button"/> +
Poids total	- <input type="button"/> +



Gestion des commandes	
<input type="text"/>	<input type="button"/> Tout <input type="checkbox"/>
<input type="button"/> Trier par :	Plus récent (par défaut) Plus ancien Avis Ordre Alphabetique
Commande 1	<input type="button"/> <input type="checkbox"/>
Commande 2	<input type="button"/> <input type="checkbox"/>
Commande 3	<input type="button"/> <input type="checkbox"/>
Commande 4	<input type="button"/> <input type="checkbox"/>
Commande 5	<input type="button"/> <input type="checkbox"/>
Commande 6	<input type="button"/> <input type="checkbox"/>



Client concerné par cette commande



Caractéristiques de la commande
Object <input checked="" type="checkbox"/>
Quantité <input checked="" type="checkbox"/>
Conditionnement <input checked="" type="checkbox"/>
Code client <input checked="" type="checkbox"/>
Points <input checked="" type="checkbox"/>
Poids total <input checked="" type="checkbox"/>
État



Choisissez un Client	Tout <input type="checkbox"/> <input checked="" type="checkbox"/>
<<Nom + Prénom>>	<input type="checkbox"/>
<<Nom + Prénom>>	<input type="checkbox"/>
<<Nom + Prénom>>	<input type="checkbox"/>
<<Nom + Prénom>>	<input type="checkbox"/>
<<Nom + Prénom>>	<input type="checkbox"/>
<<Nom + Prénom>>	<input type="checkbox"/>

Nouveau <input type="button"/> Tout <input type="checkbox"/> <input type="button"/>
<<Nom + Prénom>>
<<Code client>>
<<Adresse>>
<<N° de téléphone>>
<<E-mail>>

Gestion des clients

Tout

<<Nom + Prénom>>			<input type="checkbox"/>
<<Nom + Prénom>>			<input type="checkbox"/>
<<Nom + Prénom>>			<input type="checkbox"/>
<<Nom + Prénom>>			<input type="checkbox"/>
<<Nom + Prénom>>			<input type="checkbox"/>
<<Nom + Prénom>>			<input type="checkbox"/>
<<Nom + Prénom>>			<input type="checkbox"/>

Nouveau mail

À _____
Objet _____

Contenu _____

1/400

Envoyer

Liste des commandes pour le client : <<Nom + Prénom>>

Tout

Trier par : **Plus récent (par défaut)** Plus ancien Ordre Alphabetique

Commande 1	État:			<input type="checkbox"/>
Commande 2	État:			<input type="checkbox"/>
Commande 3	État:			<input type="checkbox"/>
Commande 4	État:			<input type="checkbox"/>
Commande 5	État:			<input type="checkbox"/>
Commande 6	État:			<input type="checkbox"/>

Détails de la commande

Object
Code client
Date
Conditionnement
Adresse
Ville
Code postal
Poids total
Points
Quantité

Cette maquette donnera des détails plus visuels . Elle sera également plus cohérente au vu du fait d'avoir conçu les diagrammes dans le bon ordre. Sur cette maquette, je détaille avec les flèches les actions de chaque bouton. Cela nous donne une vue plus concrète permettant d'assurer une bonne continuité au projet sans y revenir par la suite.

Nathan DEVOISE