



Faculté des  
Sciences de Bizerte

République Tunisienne  
Ministère de l'Enseignement Supérieur,  
de la Recherche Scientifique et de la Technologie  
Université de Carthage



Département  
Informatique

Code: SI3

## Rapport de Projet de fin d'études

Pour l'obtention de la Licence Fondamentale en Sciences de l'Informatique

Parcours: Sciences de l'Informatique

*Intitulé :*

**Mise en place d'un site web qui gère les fonctions des services de l'entreprise : les fonctions de production, les fonctions d'expédition et les fonctions des services financières.**

*Réalisé par:*

Nouha Ben Gara Ali  
Nour Beyrem

*Au sein de:*

Manufacturing International Company



*Encadré par:*

Mr. Jed Soussi (MIC)  
Mr. Slim Kharbech (MIC)  
Mme. Sabra Mabrouk (FSB)

Année Universitaire 2017-2018



## **Remerciements**

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous adressons nos sincères remerciements à Madame Mabrouk Sabra, notre encadrante au sein de la FSB pour son aide.

Nous voudrions également lui témoigner notre gratitude pour sa patience et son soutien qui nous a été précieux afin de mener notre travail à bon port.

Nous tenons à remercier également, l'ensemble du personnel de la société MIC de nous avoir accueilli au sein de leur organisme. Qu'il nous soit permis d'exprimer notre respect et notre profonde gratitude à Monsieur Jed Soussi et Monsieur Slim Kharbech, nos encadrants à MIC, pour leur bienveillance et leur disponibilité et pour leur confiance qu'ils ont su nous accorder.

Enfin, nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner.

# **Table des matières**

|  |    |
|--|----|
| Introduction générale .....  | 1  |
| Chapitre I. Présentation générale & Etat de l'art.....                 | 2  |
| I.1 Contexte générale du projet.....                                   | 2  |
| I.1.1 Cadre du projet.....   | 2  |
| I.1.2 Présentation de l'organisme d'accueil.....                       | 2  |
| I.2 Présentation du projet.....  | 3  |
| I.2.1 Problématique .....  | 3  |
| I.2.2 Objectif .....   | 4  |
| I.3 Etude de l'existant .....  | 4  |
| I.3.1 L'existant dans l'organisme d'accueil.....                       | 4  |
| I.3.2 Autres solutions présentes sur le marché.....                    | 5  |
| I.3.2.1 Les Entreprises Ressources Planning .....                      | 5  |
| I.3.2.2 Les caractéristiques des entreprises ressources planning ..... | 5  |
| I.3.2.3 Les avantages liés à la mise en place d'un ERP .....           | 5  |
| I.3.2.4 Les inconvénients liés à la mise en place d'un ERP .....       | 6  |
| I.3.2.5 Exemple de ERP .....   | 6  |
| I.3.2.5.1 L'application TexAMS ERP .....                               | 6  |
| I.3.2.5.2 Analyse fonctionnelle .....                                  | 7  |
| I.3.2.6 Critique de la solution TexAMS ERP .....                       | 8  |
| I.3.3 Solution proposée.....   | 8  |
| I.4 Méthodologie de travail .....                                      | 8  |
| Chapitre II. Analyse & Spécification des besoins.....                  | 11 |
| II.1 Analyse des besoins .....   | 11 |
| II.1.1 Identification des acteurs du système .....                     | 11 |
| II.1.2 Analyse des besoins fonctionnels.....                           | 12 |
| II.1.2.1 Les besoins du client .....                                   | 12 |
| II.1.2.2 Les besoins des utilisateurs de l'entreprise.....             | 12 |
| II.1.3 Analyse des besoins non fonctionnels .....                      | 13 |
| II.2 Spécification des besoins.....                                    | 13 |
| II.2.1 Identification des cas d'utilisation .....                      | 13 |
| II.2.1.1 Le diagramme des cas d'utilisation global .....               | 14 |
| II.2.2 Spécification des besoins du client.....                        | 15 |
| II.2.2.1 Cas d'utilisation « Créer un compte » .....                   | 16 |
| II.2.2.2 Cas d'utilisation « Passer un bon de commande ».....          | 17 |

|               |  |    |
|---------------|--|----|
| II.2.2.3      | Cas d'utilisation « Suivre la commande » .....                                     | 18 |
| II.2.3        | Spécification des besoins de l'administrateur.....                                 | 19 |
| II.2.3.1      | Le diagramme de cas d'utilisation Gérer les utilisateurs.....                      | 20 |
| II.2.3.1.1    | Cas d'utilisation « Ajouter un nouvel utilisateur » .....                          | 21 |
| II.2.3.1.2    | Cas d'utilisation « Modifier un utilisateur » .....                                | 22 |
| II.2.3.1.3    | Cas d'utilisation « Supprimer un utilisateur» .....                                | 23 |
| II.2.3.1.4    | Cas d'utilisation «Consulter un utilisateur».....                                  | 24 |
| II.2.3.1.5    | Cas d'utilisation «Chercher un utilisateur ».....                                  | 25 |
| II.2.4        | Spécification des besoins du responsable Key-Account .....                         | 26 |
| II.2.5        | Spécification des besoins du responsable de planning-export.....                   | 26 |
| II.2.6        | Diagramme de séquence système.....   | 27 |
| II.2.6.1      | Diagramme séquence système du cas d'utilisation s'authentifier .....               | 27 |
| II.2.6.2      | La description textuelle du cas d'utilisation «S'authentifier» .....               | 28 |
| II.2.6.3      | Diagramme séquence système du cas d'utilisation Ajouter une commande<br>29         |    |
| II.2.6.4      | La description textuelle du cas d'utilisation « Ajouter une commande » ...         | 30 |
| II.2.6.5      | Diagramme séquence système du cas d'utilisation Modifier une commande<br>31        |    |
| II.2.6.6      | La description textuelle du cas d'utilisation « Modifier une commande » .          | 32 |
| II.2.6.7      | Diagramme séquence système du cas d'utilisation « Consulter une<br>commande »..... | 33 |
| II.2.6.8      | Description textuelle du cas d'utilisation « Consulter une commande » ....         | 34 |
| II.2.6.9      | Diagramme séquence système du cas d'utilisation « Supprimer une<br>commande »..... | 35 |
|               | .....  | 35 |
| II.2.6.10     | La description textuelle du cas d'utilisation « Supprimer une commande »<br>36     |    |
| II.2.7        | Etude du domaine d'un point de vue statique .....                                  | 37 |
| II.3          | Les maquettes d'interfaces .....   | 40 |
| II.3.1        | La maquette d'identification.....  | 40 |
| II.3.2        | La maquette de gestion des utilisateurs .....                                      | 40 |
| II.3.3        | La maquette de gestion des bons de commande .....                                  | 41 |
| II.3.4        | La maquette de gestion de l'expédition.....  | 42 |
| Chapitre III. | Conception .....   | 43 |
| III.1         | Conception global .....  | 43 |
| III.1.1       | Architecture physique.....   | 43 |
| III.1.2       | Architecture logique .....   | 45 |
| III.1.3       | Le Patron de conception MVC .....  | 46 |

|              |   |    |
|--------------|---|----|
| III.2        | Modélisation du système .....   | 47 |
| III.2.1      | Le diagramme de package .....   | 47 |
| III.3        | Conception détaillée .....  | 49 |
| III.3.1      | Diagramme de classe de conception du cas d'utilisation « S'identifier » .....               | 50 |
| III.3.2      | Diagramme de classe de conception du cas d'utilisation « Gérer les utilisateurs»            |    |
|              | 51  |    |
| III.3.3      | Diagramme de classe de conception du cas d'utilisation « Créer un compte »..                | 52 |
| III.3.4      | Diagramme de classe de conception du cas d'utilisation « Approuver un client »              |    |
|              | 53  |    |
| III.3.5      | Diagramme de classe de conception du cas d'utilisation « Gérer les bons de commande » ..... | 54 |
| III.3.6      | Diagramme de classe de conception du cas d'utilisation « Gérer la production »              |    |
|              | 55  |    |
| III.3.7      | Diagramme de classe de conception du cas d'utilisation « Gérer le lavage »....              | 56 |
| III.3.8      | Diagramme de classe de conception du cas d'utilisation « Gérer la finition »....            | 57 |
| III.3.9      | Diagramme de classe de conception du cas d'utilisation « Gérer les produits »               | 58 |
| III.3.10     | Diagramme de classe de conception du cas d'utilisation « Gérer l'expédition »               |    |
|              | 59  |    |
| III.3.11     | Diagramme de classe de conception du cas d'utilisation « Gérer les articles »               |    |
|              | 60  |    |
| III.3.12     | Diagramme de classe de conception du cas d'utilisation « Gérer les achats »                 | 61 |
| III.3.13     | Diagramme de séquence objet.....  | 62 |
| III.3.13.1   | Diagramme de séquence objet du cas d'utilisation « Ajouter un utilisateur»                  |    |
|              | 62  |    |
| III.3.13.2   | Diagramme de séquence objet du cas d'utilisation « Modifier un utilisateur»                 |    |
|              | 64  |    |
| III.3.14     | Diagramme de composant.....   | 65 |
| III.3.14.1   | Diagramme de composants de la Gestion des utilisateurs .....                                | 65 |
| Chapitre IV. | Réalisation.....  | 67 |
| IV.1         | Environnement de travail .....  | 67 |
| IV.1.1       | Environnement matériel .....  | 67 |
| IV.1.2       | Environnement logiciel .....  | 68 |
| IV.1.2.1     | Eclipse Oxygen .....  | 68 |
| IV.1.2.2     | WebStorm.....   | 68 |
| IV.1.2.3     | MongoDB .....   | 68 |
| IV.1.2.4     | NodeJS.....   | 71 |
| IV.1.2.5     | Apache Tomcat .....   | 71 |
| IV.1.2.6     | PowerAMC .....  | 71 |
| IV.1.2.7     | Visual Paradigm.....  | 71 |

|            |   |    |
|------------|---|----|
| IV.1.2.8   | Balsamiq Mockups.....   | 71 |
| IV.2       | Choix technique .....   | 72 |
| IV.2.1     | Framework de développement .....  | 72 |
| IV.2.1.1   | Angular 4 .....   | 72 |
| IV.2.1.2   | Bootstrap.....  | 73 |
| IV.2.1.3   | Spring Boot.....  | 73 |
| IV.2.1.3.1 | Etude de cas : Création du Backend Spring Boot (Exemple : Gestion des commandes)..... | 75 |
| IV.2.2     | Langage de programmation .....  | 75 |
| IV.2.2.1   | JAVA Entreprise Edition.....  | 75 |
| IV.2.2.2   | HTML : HypertextMarkup Langage .....  | 76 |
| IV.2.2.3   | CSS : Cascading Style Sheet .....   | 76 |
| IV.2.2.4   | JavaScript.....   | 76 |
| IV.2.2.5   | TypeScript.....   | 76 |
| IV.2.2.6   | UML : Langage de modélisation unifié .....  | 77 |
| IV.3       | Les interfaces homme/machine .....  | 77 |
| IV.3.1     | Interface de l'identification.....  | 77 |
| IV.3.2     | Interface de gestion des utilisateurs .....   | 78 |
| IV.3.3     | Interface de gestion des commandes .....  | 78 |
| IV.3.4     | Interface de gestion des exports .....  | 79 |
| IV.4       | Test et validation.....   | 81 |
| IV.4.1     | Critères ergonomiques.....  | 81 |
| IV.4.1.1   | Guidage.....  | 81 |
| IV.4.1.1.1 | Incitation.....   | 81 |
| IV.4.1.1.2 | Groupement/Distinction entre Items : .....  | 83 |
| IV.4.1.1.3 | Lisibilité : .....  | 83 |
| IV.4.1.2   | Charge du travail : .....   | 84 |
| IV.4.1.2.1 | Brièveté .....  | 84 |
| IV.4.1.2.2 | Action minimale .....   | 84 |
| IV.4.1.3   | Gestion des erreurs.....  | 85 |
| IV.4.1.3.1 | Qualité des messages d'erreurs : .....  | 85 |
| IV.4.1.4   | Homogénéité/Cohérence : .....   | 86 |
| IV.4.1.5   | Contrôle explicite .....  | 87 |
| IV.4.1.5.1 | Contrôle utilisateur.....   | 87 |
| IV.4.1.6   | Rapidité et performance de l'application : .....                                      | 88 |
| IV.4.1.6.1 | Le temps de réponse de quelques services de notre application :.....                  | 88 |
|            | Conclusion générale et perspectives.....  | 93 |

## **Tables des figures**

|   |    |
|---|----|
| <i>Figure I.1 - Logo de société MIC .....</i>   | 2  |
| <i>Figure I.2 - Organigramme de la société .....</i>  | 3  |
| <i>Figure I.3 - Page de l'application TexAMS ERP .....</i>  | 7  |
| <i>Figure I.4 - Le modèle en Cascade .....</i>  | 9  |
| <i>Figure II.1 - Diagramme de cas d'utilisation global .....</i>  | 14 |
| <i>Figure II.2- Diagramme de cas d'utilisation du client .....</i>  | 15 |
| <i>Figure II.3 - Diagramme de cas d'utilisation de l'administrateur .....</i>                                 | 19 |
| <i>Figure II.4 - Diagramme de cas d'utilisation de Gérer les utilisateurs .....</i>                           | 20 |
| <i>Figure II.5 - Diagramme de cas d'utilisation du responsable Key-Account .....</i>                          | 26 |
| <i>Figure II.6 - Diagramme de cas d'utilisation du responsable d'export.....</i>                              | 26 |
| <i>Figure II.7 - Diagramme séquence système du cas d'utilisation s'authentifier.....</i>                      | 27 |
| <i>Figure II.8 - Description textuelle du cas d'utilisation Ajouter une commande.....</i>                     | 29 |
| <i>Figure II.9 - Description textuelle du cas d'utilisation Modifier une commande .....</i>                   | 31 |
| <i>Figure II.10 - Description textuelle du cas d'utilisation Consulter une commande .....</i>                 | 33 |
| <i>Figure II.11 - Description textuelle du cas d'utilisation Supprimer une commande .....</i>                 | 35 |
| <i>Figure II.12 - Diagramme de classe d'analyse .....</i>   | 38 |
| <i>Figure II.13 - Prototype d'identification .....</i>  | 40 |
| <i>Figure II.14 - Prototype de Gérer les utilisateurs .....</i>   | 41 |
| <i>Figure II.15 - Prototype de Gérer les bons de commande .....</i>   | 41 |
| <i>Figure II.16 Prototype de Gérer l'expédition .....</i>   | 42 |
| <i>Figure III.1 - Architecture 3-tiers de notre application .....</i>   | 44 |
| <i>Figure III.2 - Architecture logique de l'application .....</i>   | 45 |
| <i>Figure III.3 - Le patron de conception MVC .....</i>   | 46 |
| <i>Figure III.4 - Diagramme de packages .....</i>   | 48 |
| <i>Figure III.5 - Diagramme de classe de conception du cas d'utilisation « S'identifier » .....</i>           | 50 |
| <i>Figure III.6 - Diagramme de classe de conception du cas d'utilisation Gérer les utilisateurs</i>           | 51 |
| <i>Figure III.7 - Diagramme de classe de conception du cas d'utilisation Créeer un compte (Client) .....</i>  | 52 |
| <i>Figure III.8 - Diagramme de classe de conception du cas d'utilisation Approuver un client .</i>            | 53 |
| <i>Figure III.9 - Diagramme de classe de conception du cas d'utilisation Gérer les bons de commande .....</i> | 54 |
| <i>Figure III.10 - Diagramme de classe de conception du cas d'utilisation Gérer la production .....</i>       | 55 |
| <i>Figure III.11 - Diagramme de classe de conception du cas d'utilisation Gérer le lavage .....</i>           | 56 |
| <i>Figure III.12 - Diagramme de classe de conception du cas d'utilisation Gérer la finition .....</i>         | 57 |
| <i>Figure III.13 - Diagramme de classe de conception du cas d'utilisation Gérer les produits ..</i>           | 58 |
| <i>Figure III.14 - Diagramme de classe de conception du cas d'utilisation Gérer l'expédition ..</i>           | 59 |
| <i>Figure III.15 - Diagramme de classe de conception du cas d'utilisation Gérer les articles ...</i>          | 60 |
| <i>Figure III.16 - Diagramme de classe de conception du cas d'utilisation Gérer les achats....</i>            | 61 |
| <i>Figure III.17 - Diagramme de séquence objet du cas d'utilisation Ajouter un utilisateur.....</i>           | 63 |
| <i>Figure III.18 - Diagramme de séquence objet du cas d'utilisation Modifier un utilisateur.....</i>          | 64 |

|   |    |
|---|----|
| <i>Figure III.19- Diagramme de composant de la partie FrontEnd de la Gestion des utilisateurs</i>           | 66 |
| <i>Figure IV.1 - Les caractéristiques du PC HP</i>  | 67 |
| <i>Figure IV.2 Les caractéristiques du PC LENOVO</i>  | 67 |
| <i>Figure IV.3 - Logo du MongoDB</i>  | 68 |
| <i>Figure IV.4 – Le modèle de la base de données</i>  | 70 |
| <i>Figure IV.5 - Logo du NodeJS</i>   | 71 |
| <i>Figure IV.6 - Logo de l'Angular 4</i>  | 72 |
| <i>Figure IV.7 - Logo du Bootstrap</i>  | 73 |
| <i>Figure IV.8 - Logo du Spring Boot</i>  | 73 |
| <i>Figure IV.9 - Architecture du Spring Boot</i>  | 73 |
| <i>Figure IV.10 - Logo Java Entreprise Edition</i>  | 76 |
| <i>Figure IV.11 - Interface d'identification</i>  | 77 |
| <i>Figure IV.12 - Interface de gestion des utilisateurs</i>   | 78 |
| <i>Figure IV.13 - Interface de gestion des commandes</i>  | 79 |
| <i>Figure IV.14 - Interface de gestion des exports</i>  | 79 |
| <i>Figure IV.15 - Interface d'ajout d'une facture</i>   | 80 |
| <i>Figure IV.16 - Interface de téléchargement d'une facture</i>   | 80 |
| <i>Figure IV.17 - Exemple de l'incitation respectée dans l'application : Format de Date</i>                 | 82 |
| <i>Figure IV.18 - Exemple de l'incitation respectée dans l'application : Fournir un label clair</i>         | 82 |
| <i>Figure IV.19 - Exemple de Groupement entre items respecté dans l'application</i>                         | 83 |
| <i>Figure IV.20 - Exemple qui montre la lisibilité respectée dans l'application</i>                         | 83 |
| <i>Figure IV.21 - Exemple qui montre la brièveté respectée dans l'application</i>                           | 84 |
| <i>Figure IV.22 - Exemple qui illustre l'utilisation des actions minimales respectée dans l'application</i> | 85 |
| <i>Figure IV.23 - Exemple qui montre la qualité des messages générés par l'application</i>                  | 86 |
| <i>Figure IV.24 - Homogénéité et cohérence : Gestion de production</i>                                      | 86 |
| <i>Figure IV.25 - Homogénéité et cohérence : Gestion de finition</i>  | 87 |
| <i>Figure IV.26 - Exemple de Contrôle utilisateur utilisé dans l'application</i>                            | 87 |
| <i>Figure IV.27 - Exemple de Contrôle utilisateur utilisé dans l'application</i>                            | 88 |
| <i>Figure IV.28 - Calcul de temps de réponse pour l'identification</i>                                      | 88 |
| <i>Figure IV.29 - Calcul de temps de réponse pour le cas d'affichage de la liste des commandes</i>          | 89 |
| <i>Figure IV.30 - Calcul de temps de réponse pour le cas d'ajouter une commande</i>                         | 89 |
| <i>Figure IV.31 - Calcul de temps de réponse pour le cas de modifier une commande</i>                       | 90 |
| <i>Figure IV.32 - Calcul de temps de réponse pour le cas de supprimer une commande</i>                      | 90 |
| <i>Figure IV.33 - Calcul de temps de réponse pour le cas de chercher une commande</i>                       | 91 |
| <i>Figure IV.34 - Calcul de temps de réponse pour le cas de chercher une commande</i>                       | 91 |

## **Table des tableaux**

|   |    |
|---|----|
| <i>Tableau II-1 - La description textuelle de cas d'utilisation Créer un compte .....</i>           | 16 |
| <i>Tableau II-2 - La description textuelle de cas d'utilisation Passer un bon de commande .....</i> | 17 |
| <i>Tableau II-3 - Description textuelle du cas d'utilisation Suivre la commande.....</i>            | 18 |
| <i>Tableau II-4 - Description textuelle du cas d'utilisation Ajouter un nouvel utilisateur.....</i> | 21 |
| <i>Tableau II-5 - Description textuelle du cas d'utilisation Modifier un utilisateur.....</i>       | 22 |
| <i>Tableau II-6 - Description textuelle du cas d'utilisation Supprimer un utilisateur .....</i>     | 23 |
| <i>Tableau II-7 - Description textuelle du cas d'utilisation Consulter un utilisateur.....</i>      | 24 |
| <i>Tableau II-8 - Description textuelle du cas d'utilisation Chercher un utilisateur .....</i>      | 25 |
| <i>Tableau II-9 - Description textuelle du cas d'utilisation S'authentifier .....</i>               | 28 |
| <i>Tableau II-10 - Description textuelle du cas d'utilisation Ajouter une commande .....</i>        | 30 |
| <i>Tableau II-11 - Description textuelle du cas d'utilisation Modifier une commande.....</i>        | 32 |
| <i>Tableau II-12 - Description textuelle du cas d'utilisation Consulter une commande .....</i>      | 34 |
| <i>Tableau II-13 - Description textuelle du cas d'utilisation Supprimer une commande .....</i>      | 36 |

# Introduction générale

L'entreprise d'aujourd'hui est plus consciente de l'impact de la gestion efficiente des ressources internes sur l'amélioration de sa performance et par la suite de sa compétitivité dans le marché. Or, la tâche de gérer s'avère de plus en plus difficile et complexe. En effet, la croissance des activités engendre un énorme flux de données, alors que la diversité des processus fonctionnels nécessite une gestion adéquate des compétences humaines. Pour surpasser ces difficultés, toute entreprise est prête à investir des sommes considérables dans l'implémentation de technologies logicielles afin d'améliorer ses services et d'accroître son agilité vis-à-vis de ses clients tout en optimisant la communication au sein de ses équipes.

Pour la société MIC, la gestion de l'ensemble de ses activités se fait manuellement ou par le biais des applications informatiques. La société souhaite gérer la totalité de ses activités autour d'un même système d'information.

C'est dans ce cadre que s'inscrit notre projet de fin d'études qui a pour objectif de mettre en place un site web qui gère les fonctions des services de l'entreprise MIC. Pour cela, l'analyse des besoins a permis d'identifier les fonctionnalités à mettre en place à savoir la gestion des articles, la gestion des achats, la gestion des bons de commandes, la gestion de fabrication, la gestion de finition, la gestion de lavage, la gestion des produits, la gestion d'expédition ainsi que la gestion des paiements.

Pour ce faire, Nous allons utiliser le modèle en Cascade comme méthodologie de gestion de projet. Il nous permet d'élaborer un projet qui se déroule séquentiellement à partir de sa conception, chaque phase conduisant à la suivante. Nous allons utiliser JAVA EE comme plateforme de développement qui est aujourd'hui la meilleure plate-forme de développement pour les entreprises. Pour les Framework, nous allons utiliser Spring Boot et Angular 4.

Ce présent rapport se compose de cinq chapitres. Dans le premier chapitre « Présentation générale & Etat de l'art », nous présentons l'organisme d'accueil, l'étude de l'existant ainsi que la problématique qui a donné naissance à ce projet et la méthodologie utilisée pour sa réalisation. Le deuxième chapitre, intitulé « Analyse et spécifications des besoins », est consacré à l'identification et l'analyse des besoins fonctionnels et les besoins non fonctionnels, et ensuite la présentation des diagrammes des cas d'utilisations, des diagrammes séquence système et le diagramme de classe d'analyse. Dans le troisième chapitre « Conception », nous présentons l'architecture physique et logique de notre application et le patron de conception que nous avons adopté, puis la modélisation du système par le diagramme de package. Ensuite nous identifions la conception détaillée dans laquelle les diagrammes conceptuels seront présentés : les diagrammes de classe de conception, les diagrammes séquence objet et le diagramme de composant. Dans le quatrième et dernier chapitre, nous présentons les outils matérielle et logicielle adoptées pour le développement de ce projet. Ensuite, nous présenterons une démonstration de quelques scénarios du fonctionnement de l'application à travers des captures d'écrans et un ensemble de test de validation.

La conclusion dresse le bilan de ce travail et présente quelques perspectives.

# Chapitre I. Présentation générale & Etat de l'art

---

La construction d'un état de l'art constitue une étape préliminaire pour la réalisation d'une application. En effet, elle permet d'analyser, d'évaluer et de critiquer le fonctionnement habituel, tout en élaborant la liste des solutions possibles.

Ce chapitre sera réservé pour présenter l'état de l'art de notre projet. Nous commençons par la définition du contexte général du projet puis de la problématique. Ensuite nous étudions l'existant et nous analysons quelques solutions existantes sur le marché en discutant leurs avantages et leurs inconvénients. L'analyse et le critique de l'existant permettent de cerner nos objectifs afin de développer un système de qualité dans le futur. Enfin, nous proposons les différentes solutions aux problèmes soulevés.

## I.1 Contexte générale du projet

### I.1.1 Cadre du projet

Dans le cadre de notre projet de fin d'études à la Faculté des sciences de Bizerte (FSB) et en vue de l'obtention du titre Technicien Supérieur en informatique, nous avons effectué un stage de quatre mois à Manufacturing International Company (MIC).

Durant ce stage, nous avons eu pour mission dans un premier temps de concevoir et de réaliser une application qui gère les fonctions de services, les services d'ordonnancement, les fonctions d'achat, les fonctions de production et les fonctions d'expédition en suivant un cycle de vie qui commence à l'étape de l'analyse et la spécification des besoins, puis la conception générale, la conception détaillée, le codage jusqu'à la publication finale en passant par les étapes de la vérification, de la validation et de l'approbation. Dans un deuxième temps, nous étions amenées à intégrer notre application dans le portail interne de la société et assurer son opérabilité avec leur système d'information.

### I.1.2 Présentation de l'organisme d'accueil

Manufacturing International Company est une entreprise industrielle dans le secteur d'industrie Textiles et habillement à Bizerte. Crée en 2005, son directeur générale est Eric Lancouvski et son gérant est Philippe Benat. C'est une société anonyme, son objectif social est la confection de jeans et son domaine est le textile.



Figure I.1 - Logo de société MIC

L'organigramme de la figure I.2 permet de visualiser la répartition des rôles et de l'autorité au sein de l'entreprise, il positionne les leaders, les détenteurs de pouvoir et de responsabilité. Nous sommes accueillies dans le service informatique pour développer une application qui gère le fonctionnement de tous les services de l'entreprise.

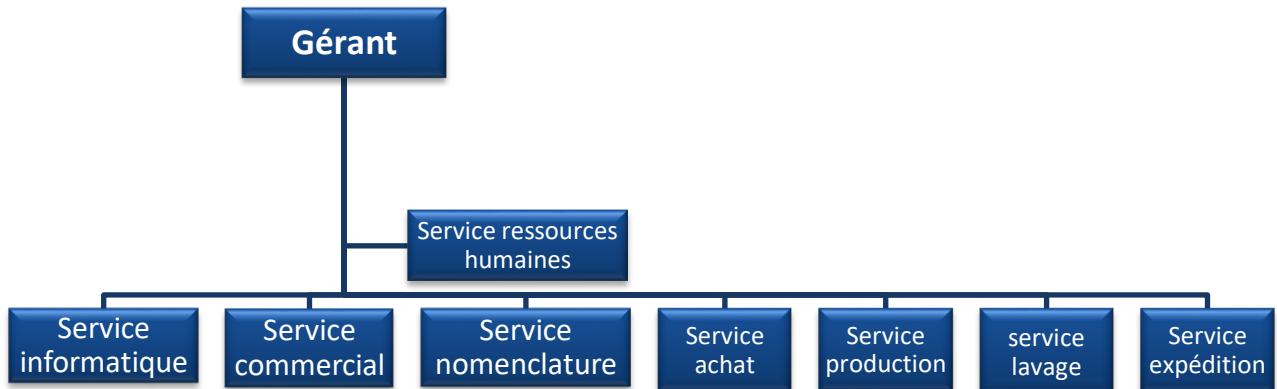


Figure I.2 - Organigramme de la société

## I.2 Présentation du projet

### I.2.1 Problématique

Pour gérer l'activité globale de la société, chaque département utilise indépendamment des applications informatiques. Ceci génère non seulement une difficulté d'échange de données entre les différents départements, mais également une incohérence de l'information interne et externe.

Certes, il existe des solutions sur le marché, mais elles sont payantes, en plus du fait qu'il est difficile de communiquer entre ces applications avec d'autres solutions externes et leur support est bien limité vu qu'elles ont été codées à la demande, à noter aussi qu'elles ne sont pas flexibles.

Avec ces outils, un gérant se retrouve obligé à effectuer lui-même un bon nombre de tâches qui, normalement, avec ces solutions doivent être automatisées, nous citons comme exemple l'envoi des bons de commande vers l'entreprise : le client crée un bon de commande avec une liste des produits demandés ainsi que leur quantités et l'envoie sous forme d'un dossier électronique vers l'entreprise par mail, mais il est obligé de suivre sa commande par la méthode classique via l'appel téléphonique.

Les bons de commande sont donc temporaires, et se trouvent dans la plupart des cas supprimés après avoir effectué la réception des produits.

Ainsi, il s'avère important de coordonner l'ensemble des activités de la société (comme la gestion d'ordonnancement, la gestion de production, la gestion de lavage et la gestion des services d'expédition...) autour d'un même système d'information.

## I.2.2 Objectif

Notre projet de fin d'études a pour objectif de coordonner l'ensemble des activités de la société (comme la gestion d'ordonnancement, la gestion de production, la gestion de lavage et la gestion des services d'expédition...), tout en travaillant sur une base de données unique et homogène afin de gagner en productivité et réduire les travaux redondants.

## I.3 Etude de l'existant

Avant de spécifier les besoins, il est nécessaire de réaliser une étude de l'existant. L'étude de l'existant est une étape clé dans la réalisation de n'importe quelle application informatique, quel que soit le domaine concerné. Il s'agit d'une étude permettant de comprendre la problématique du projet et de détecter les avantages et inconvénients des solutions proposées actuellement sur le marché afin d'en profiter pour la réalisation de notre projet.

### I.3.1 L'existant dans l'organisme d'accueil

La société MIC repose principalement sur quatre divisions qui assurent le fonctionnement normal d'une chaîne de planification des ressources d'entreprise qui sont : le service d'ordonnancement, le service d'achat, le service de production et enfin le service d'expédition.

La chaîne de planification des ressources d'entreprise est présentée comme un ensemble d'acteurs, qui sont l'administrateur, les différents responsables de la société et les clients, interconnectés par des liens physiques et informationnels.

Cette chaîne est toujours considérée comme un réseau qui procure de la matière première et la transforme en des produits finis livrés aux clients. L'objectif étant une satisfaction de ce client au plus tôt et à moindre coût.

En effet, le service d'ordonnancement couvre la planification de l'approvisionnement avec la gestion de stock de la matière première.

Le service d'achat contrôle les rapports prix/coûts. Le cœur de ce service consiste en premier lieu à satisfaire les besoins des clients internes, en obtenant la meilleure performance des fournisseurs (Qualité / Coût / Délais / Services) et en optimisant les coûts complets.

Le service de production qui consiste en la planification de la production et la gestion des ressources industrielles internes.

Le service d'expédition concernant la planification de la distribution, le transport et la livraison du produit fini. Il délimite l'acheminement des produits finis de l'entreprise jusqu'aux clients.

Ces services sont gérés par une base de données enregistrée sous forme de fichiers Excel. L'utilisation de ces fichiers cause une grande perte de temps lors de la recherche d'une commande ainsi que la difficulté de faire des mises à jour. De plus, l'inexistence d'une plateforme unique qui réunit toutes les informations relatives à l'entreprise pour gérer les fonctions de services offertes par celle-ci.

On remarque que les solutions existantes ne sont pas adaptées aux besoins exacts de l'entreprise.

### I.3.2 Autres solutions présentes sur le marché

#### I.3.2.1 Les Entreprises Ressources Planning

Un ERP est un progiciel qui peut être utilisé par tous les salariés d'une entreprise et qui permet de centraliser et de mutualiser les informations de chaque service sur une base unique de données. Il est capable d'automatiser la plupart des tâches financières, logistiques, ou commerciales de l'activité d'une entreprise, de faire communiquer ces différents services entre eux et de fournir une vue globale aux dirigeants. En effet, les ERP couvrent généralement les fonctions de la gestion financière, la gestion de la chaîne logistique (approvisionnement, production, distribution), la gestion commerciale et de la relation client et la gestion de ressources humaines.

#### I.3.2.2 Les caractéristiques des entreprises ressources planning

Afin d'assurer le bon fonctionnement d'une organisation, différentes solutions logicielles e-business ont tendance à fonctionner en arrière-plan. Il existe différentes solutions logicielles de gestion disponibles sur le marché et qui peuvent aider à la gestion d'un certain nombre d'activités, mais malgré cette vérité, les entreprises ont recours de plus en plus à l'ERP grâce à un ensemble de caractéristiques qui le distinguent des autres solutions de gestion :

- Architecturé autour d'un système de base de données relationnelle assurant une gestion unique, cohérente, sécurisée et ouverte des données de l'entreprise : c'est-à-dire qu'un ERP permet une logique et une ergonomie unique à travers sa base de données, elle aussi unique au sens "logique". Ceci se traduit par le fait qu'il peut exister plusieurs bases de données "physiques".
- Permet de couvrir et modéliser l'ensemble des besoins de gestion de production, de distribution (achats, ventes, stocks), gestion de la chaîne logistique, de finance (comptabilité générale, analytique, budgétaire, tiers), gestion de la relation client, des services commerciaux.
- Constitué de modules fonctionnels : Un ERP est subdivisé en modules qui répondent chacun à un des domaines de gestion, chacun couvrant un périmètre de gestion de l'entreprise.
- Gère le plus souvent un contexte international : multi sociétés, multidevises, multi-langues, multi-législation : cela se traduit par l'adaptation de l'ERP au marché mondial, en particulier aux multinationales.

#### I.3.2.3 Les avantages liés à la mise en place d'un ERP

Avant de mettre en place l'ERP, chaque service avait son propre système d'information, aujourd'hui et grâce aux caractéristiques citées ci-dessus, un ensemble de fait sont achevés:

- ✓ Meilleure intégration des applications
- ✓ Meilleure traçabilité des informations.
- ✓ Meilleure transversalité du business.

- ✓ L'existence d'un référentiel unique de données.
- ✓ Gagner en temps et en efficacité
- ✓ L'adaptation rapide aux règles de gestion.
- ✓ Travail d'équipe optimisé.
- ✓ Une gestion effective de plusieurs domaines de l'entreprise, susceptibles de permettre une collaboration de processus.
- ✓ Contribuer à réduire les coûts et les risques, améliorer leur productivité et accroître leur activité.

#### **I.3.2.4 Les inconvénients liés à la mise en place d'un ERP**

Comme toute solution, les ERP ont aussi des frontières ; il est inévitable qu'elles ne coïncident pas avec ce que l'entreprise aurait souhaité. Ils peuvent rencontrer des nombreux risques au cours et après l'implantation :

- ✓ La mise en place d'un ERP est difficile et présente de nombreux risques. L'ERP peut avoir un périmètre fonctionnel limité (quelques fonctions, toute l'entreprise mais pas l'entreprise étendue - CRM, SCM, e-commerce ...).
- ✓ Sa mise en place est lourde aux plans organisationnel et financier.
- ✓ Retour en arrière ou le changement d'ERP est difficile (coût, délai, effet sur l'organisation).
- ✓ Impact sur l'organisation (il faut s'adapter à l'ERP).
- ✓ Dépendance d'un éditeur, de sa capacité de suivi, de sa vision de la gestion.
- ✓ les ERP ne peuvent s'adapter à toutes les entreprises, car chacune d'entre elles est unique et présentent des particularités de gestion. Par conséquent, des développements spécifiques peuvent être envisageables mais ils coûtent cher et sont risqués.
- ✓ Important délai de prise en main dû à sa complexité.

#### **I.3.2.5 Exemple de ERP**

##### **I.3.2.5.1 L'application TexAMS ERP**

TexAMS ERP aide une entreprise à gérer des fonctions clés telles que la planification des produits, l'achat de pièces, le maintien des stocks, l'interaction avec les fournisseurs, le service à la clientèle et le suivi des commandes. Une fois que l'usine est équipée de TexAMS ERP, tout est à portée de main. Avec cela, la direction est au courant des développements dans différentes unités, ce qui aide à mieux utiliser les ressources. [1]

## SALES ORDER APPROVAL

| ORDER NO     | PARTY CODE | PARTY NAME                              | PA NO          | ARTICAL CODE | ARTICAL DESC                            | SHADE CODE     | ORD QTY | View CR                 | BOM         | COST        | PROCESS ROOT | FLAG |
|--------------|------------|---|----------------|--------------|---|----------------|---------|-------------------------|-------------|-------------|--------------|------|
| YSSWDE000001 | 000197     | NEW LUXMI ELECTRIC CO.                  | YSSWDE00000101 | C120CHD100   | Ne 20/1 100% cotton combed hosiery yarn | 10897          | 676     | <a href="#">View CR</a> | Approved    | Approved    | Approved     |      |
| YSSWDE000002 | 000199     | KRISHNA TUBES                           | YSSWDE00000201 | C130KHD100   | Ne 30/1 100% cotton carded hosiery yarn | BLUE           | 10000   | <a href="#">View CR</a> | Approved    | Approved    | Approved     |      |
| YSSWDE000003 | 000198     | SHIV HARDWARE & PAINT STORE             | YSSWDE00000301 | C130KHD100   | Ne 30/1 100% cotton carded hosiery yarn | BLUE           | 4545    | <a href="#">View CR</a> | Approved    | Approved    | Approved     |      |
| YSSWDE000004 | ST001      | SAINATH TEXPORT LTD. - (LUDHIANA DEPOT) | YSSWDE00000401 | C130KHD100   | Ne 30/1 100% cotton carded hosiery yarn | BLUE           | 1234    | <a href="#">View CR</a> | Approved    | Un-Approved | Approved     |      |
| YSSWDE000005 | 000198     | SHIV HARDWARE & PAINT STORE             | YSSWDE00000501 | C124CHD100   | Ne 24/1 100% Cotton Combed Hosiery Yarn | BROWNISH BLACK | 361     | <a href="#">View CR</a> | Approved    | Approved    | Approved     |      |
| YSSWDE000006 | 000197     | NEW LUXMI ELECTRIC CO.                  | YSSWDE00000601 | C120CHD100   | Ne 20/1 100% cotton combed hosiery yarn | 11011          | 676     | <a href="#">View CR</a> | Approved    | Un-Approved | Un-Approved  |      |
| YSSWDE000007 | 000197     | NEW LUXMI ELECTRIC CO.                  | YSSWDE00000701 | C024RMD100   |   |                | 11011   | <a href="#">View CR</a> | Approved    | Approved    | Approved     |      |
| YSSWDE000008 | 000199     | KRISHNA TUBES                           | YSSWDE00000801 | C130KHD100   | Ne 30/1 100% cotton carded hosiery yarn | 19-0303        | 32      | <a href="#">View CR</a> | Un-Approved | Un-Approved | Un-Approved  |      |

Figure I.3 - Page de l'application TexAMS ERP

### I.3.2.5.2 Analyse fonctionnelle

- Définir plan de production
  - ✓ Définir les articles et leurs propriétés détaillées
- Définir les achats
  - ✓ Définir les termes de paiement et les conditions d'achat standard
  - ✓ Créer une demande d'achat en ligne en fonction des demandes
  - ✓ Suspendre, annuler ou fermer une demande d'achat
  - ✓ Suivre tout le matériel par numéro
- Définir la planification de distribution
  - ✓ Définir chaque produit et ses détails tels que pack taille, poids etc.
- Le contrôle des stocks
  - ✓ Générer des livres de stock instantanément
  - ✓ Suivre tout le matériel par numéro
- Le contrôle de la qualité
  - ✓ Définir les spécifications de contrôle de la qualité globale pour chaque article
- Définir les ventes
  - ✓ Maintenir produits finis prix listes
  - ✓ Traiter les commandes clients
  - ✓ Créer des commandes d'expédition à partir des commandes clients
  - ✓ Créer les factures de vente

### I.3.2.6 Critique de la solution TexAMS ERP

Après l'étude que nous avons faite concernant cette solution web TexAMS, nous avons remarqué qu'il existe des points manquants au niveau de cette application :

- Elle peut coûter cher à l'achat (entre US \$30,000 et \$50,000 selon la taille de l'entreprise). Cependant, il faut également prévoir le coût de la formation des employés à ce nouvel outil.
- La nécessité d'une maintenance régulière.
- L'application peut être non compréhensible pour l'utilisateur.
- TexAMS ne peut pas s'adapter à cette entreprise, car elle est unique et présente des particularités de gestion. Par conséquent, des développements spécifiques peuvent être envisageables mais ils coûtent cher et sont risqués (peuvent impacter les autres modules).

### I.3.3 Solution proposée

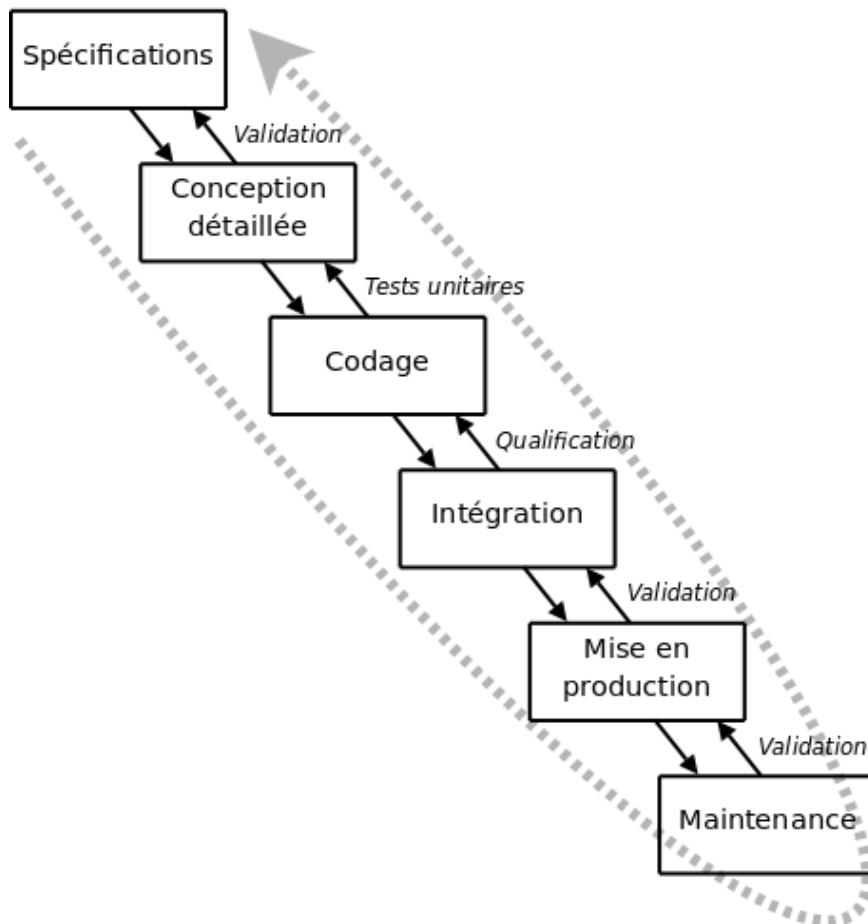
L'application que nous envisageons de mettre en place permettra d'atteindre les objectifs suivants :

- ✓ Gérer les gestions de base
  - Gérer des utilisateurs
  - Gérer des produits
  - Consulter la liste des clients
  - Gérer les bons de commandes
- ✓ Gérer l'expédition
  - Gérer les paiements
- ✓ Gérer des achats
  - Gérer les bons de commandes d'achat
  - Vérification des stocks
- ✓ Gérer les rapports
  - Imprimer les rapports (bon de commande, facture)
  - Imprimer listes des produits
- ✓ Gérer les services de messageries
  - Gestion des emails des utilisateurs
  - Envoi des commandes et factures par mail
- ✓ Gérer un planning de production
- ✓ Suivre le cycle de vie des transactions de la chaîne de production du produit

## I.4 Méthodologie de travail

En génie logiciel, le modèle en cascade est une version répandue du modèle de gestion du cycle de développement des systèmes et des applications. Souvent considéré comme l'approche classique du développement, ce modèle décrit un cycle linéaire et séquentiel.

Cette méthode constitue six ou sept étapes :



*Figure I.4 - Le modèle en Cascade*

1. L'analyse des besoins : on étudie la faisabilité du projet pour définir les besoins et les exigences. Seuls les processus majeurs sont concernés. En estimant les différents coûts du projet et les matériaux nécessaires. Il faut chiffrer les charges et envisager des délais.
2. L'analyse du système : cette étape permet de peaufiner les points vus dans l'étape précédente. On peut commencer à traiter les points mineurs.
3. Conception : ici on effectue les différents choix techniques.
4. Implémentation et tests unitaires : dans cette étape on programme. En fonction du langage choisi, les développeurs vont programmer. Une fois terminé, on effectue des tests unitaires. Ces tests vont permettre de vérifier si tous fonctionnent correctement.
5. Validation et tests d'intégration : le client à son tour effectue des tests pour valider le système. Les tests d'intégration se font souvent avec les utilisateurs.
6. Exploitation et maintenance : dans cette dernière étape, il ne reste plus qu'à déployer le système. Et la maintenance permet de corriger les problèmes et les anomalies. [2]

Il n'est possible que de revenir une étape en arrière s'il y a un problème.

## Conclusion

Dans ce chapitre nous avons décrit le contexte général dans lequel s'inscrit notre projet de fin d'études. Nous avons déterminé la problématique et nous avons analysé l'existant au sein de l'entreprise. Ensuite, nous avons présenté la solution proposée du projet qui se résume à la mise en place d'un système de suivi de la chaîne de planification des ressources d'entreprise. Dans la suite nous commençons l'analyse et la spécification des besoins dans le chapitre suivant.

## Chapitre II. Analyse & Spécification des besoins

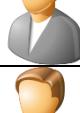
---

La phase d'analyse et spécification des besoins présente une étape primordiale dans le cycle de développement d'un projet. En effet, elle permet de mieux comprendre le travail demandé en dégageant les besoins des différents utilisateurs que doit le système accomplir. Pour ce faire, dans le présent chapitre nous allons commencer par l'analyse des besoins en définissant les différents acteurs du système et en détaillant les besoins fonctionnels et non fonctionnels ensuite nous allons présenter les différents cas d'utilisation de l'application et enfin nous exposons quelques scénarios.

### II.1 Analyse des besoins

#### II.1.1 Identification des acteurs du système

Nous avons identifié, essentiellement, huit rôles à savoir le client et l'administrateur et les différents responsables de la société:

| Utilisateur   | Rôle   |
|---|--|
|  Client                       | <ul style="list-style-type: none"><li>- Passer un bon de commande</li><li>- Suivre un bon de commande</li><li>- Créer un compte</li></ul>  |
|  Administrateur              | <ul style="list-style-type: none"><li>- Gérer les utilisateurs</li><li>- Suivre un bon de commande</li><li>- Approuver un compte</li></ul> |
|  Responsable Key-Account     | <ul style="list-style-type: none"><li>- Gérer les commandes</li><li>- Gérer les produits</li></ul>   |
|  Responsable de production   | <ul style="list-style-type: none"><li>- Gérer la production</li><li>- Gérer la finition</li></ul>  |
|  Responsable de lavage       | <ul style="list-style-type: none"><li>- Gérer lavage</li></ul>   |
|  Responsable de nomenclature | <ul style="list-style-type: none"><li>- Gérer les articles</li></ul>   |
|  Responsable d'export        | <ul style="list-style-type: none"><li>- Gérer l'expédition</li><li>- Gérer les paiements</li></ul>   |
|  Responsable d'achat         | <ul style="list-style-type: none"><li>- Gérer les achats</li></ul>   |

## II.1.2 Analyse des besoins fonctionnels

Notre application web doit offrir aux différents utilisateurs de la société différentes fonctionnalités selon les priviléges de chacun d'entre eux. Ces besoins peuvent être résumés dans les points suivants.

- Le système doit permettre à tous les utilisateurs de s'authentifier

### II.1.2.1 Les besoins du client

- Le système doit donner la possibilité au client de saisir les informations nécessaires pour la création d'un compte sous la forme d'un formulaire à remplir.
- Le système doit donner la possibilité au client de passer un bon de commande.
- Le système doit permettre au client de pouvoir suivre sa commande.

### II.1.2.2 Les besoins des utilisateurs de l'entreprise

**L'administrateur :** Le système doit permettre à l'administrateur de gérer toutes les rubriques de la société.

- Le système doit permettre à l'administrateur de gérer les différents utilisateurs.
- Le système doit permettre à l'administrateur de superviser l'ensemble des activités et il possède les mêmes droits d'utilisation que les autres responsables.
- Le système doit permettre à l'administrateur d'approuver les clients en les attribuant des login et des mots de passe.

**Responsable Key-account :** Le système doit permettre au responsable Key-account de gérer les produits finis et les commandes.

- Le système doit permettre au responsable Key-account de gérer les produits finis.
- Le système doit permettre au responsable Key-account de gérer les commandes.

**Responsable nomenclature :** Le système doit permettre au responsable nomenclature de gérer les articles.

- Le système doit permettre au responsable nomenclature de gérer les articles. En effet, il permet de créer des articles de type service ou bien les mettre à jour ou les supprimer. Aussi, il permet de visualiser la variation des stocks et de signaler les besoins vis-à-vis les produits matériels non disponibles.

**Responsable d'achat :** Le système doit permettre au responsable d'achat de gérer les achats et les factures d'achats.

- Le système doit permettre au responsable d'achat de saisir la quantité à commander afin de passer une commande d'achat.

**Responsable de production :** Le système doit permettre au responsable de production de gérer les productions.

- Le système doit permettre au responsable de production de planifier la production en indiquant les différentes dates de la chaîne de la production.
- Le système doit permettre au responsable de production de suivre le cycle de vie de la chaîne de production et de finition.

**Responsable de lavage :** Le système doit permettre au responsable de lavage de gérer les lavages.

- Le système doit permettre au responsable de lavage d'indiquer les différentes dates.
- Le système doit permettre au responsable de lavage de fixer les opérations de lavage selon les standards de qualité requis d'après le client.

**Responsable d'export :** Le système doit permettre au responsable de planning export de gérer les factures et l'expédition.

- Le système doit permettre au responsable de planning export de planifier l'expédition et confirmer la date réelle d'expédition.
- Le système doit permettre au responsable de planning export de préciser la disponibilité des différents articles, la date de livraison et le prix total de la commande.
- Le système doit permettre au responsable de planning export de gérer les factures.

### II.1.3 Analyse des besoins non fonctionnels

Notre système doit répondre à certains besoins non fonctionnels que nous devons prendre en considération pour mettre en place une solution adéquate aux attentes telles que :

- ✓ L'ergonomie : La présentation doit respecter les règles de bases de l'ergonomie des interfaces. Le thème adopté par l'application doit être inspiré des couleurs et du logo de l'entreprise.
- ✓ L'extensibilité : le code doit être clair pour permettre les nouvelles améliorations ou évolutions. Il pourra avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.
- ✓ La convivialité : l'application doit offrir une interface conviviale et facile à utiliser
- ✓ La confidentialité : consistant à s'assurer que seules les personnes autorisées aient accès aux ressources échangées
- ✓ Rapidité et flexibilité du traitement : Optimisation du temps de la réponse pour les utilisateurs.
- ✓ Fiabilité : L'application doit fonctionner de façon cohérente sans erreurs.

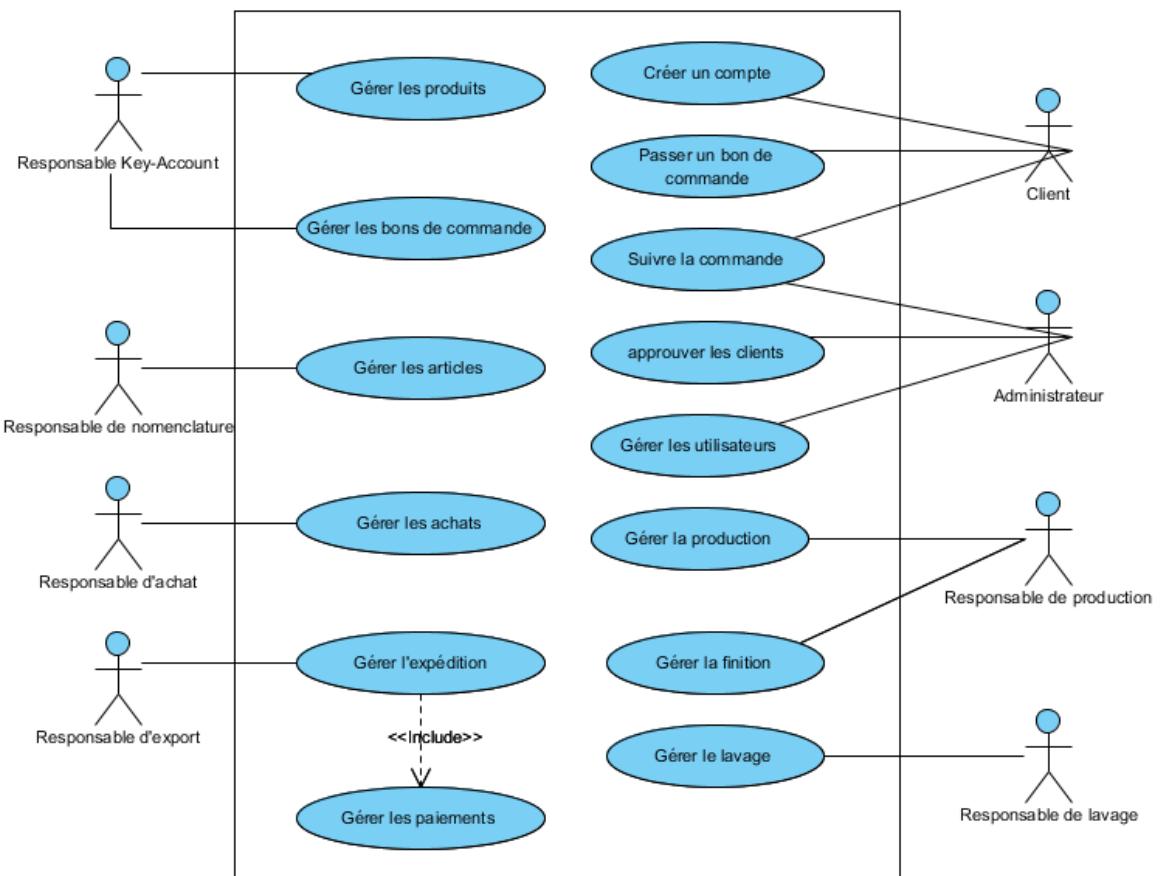
## II.2 Spécification des besoins

### II.2.1 Identification des cas d'utilisation

Cette phase a pour objectif de décrire le comportement attendu de l'application. Pour cela, nous nous basons sur le diagramme des cas d'utilisations qui présente un élément essentiel de la modélisation orientée objet. Elle permet de modéliser les fonctionnalités de l'application de point de vue besoin utilisateur. Elle sert aussi à définir le produit à développer, et le modéliser.

### II.2.1.1 Le diagramme des cas d'utilisation global

Le diagramme des cas d'utilisation global de *la figure II.1* permet de regrouper tous les acteurs du système afin de représenter une exigence fonctionnelle dans son ensemble. Chaque cas d'utilisation représente le besoin de l'acteur en question vis à vis de notre système.



*Figure II.1 - Diagramme de cas d'utilisation global*

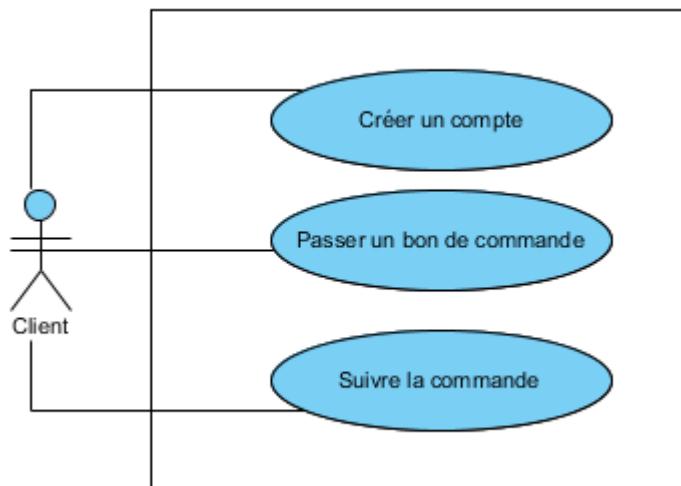
Notre application présente les cas d'utilisations suivants :

- **S'authentifier** : ce cas d'utilisation concerne la sécurité des données de la société. Chaque acteur s'identifie avec son login et son mot de passe qui vont être vérifiés par le système.
- **Gérer les commandes** : Cette fonctionnalité permet d'ajouter, de supprimer, de modifier, de consulter et de chercher une commande. Elle est réalisée par le responsable Key-Account.
- **Gérer les produits** : Elle permet d'ajouter, de supprimer, de modifier, de consulter et de chercher un produit. Elle est réalisée par le responsable Key-Account.
- **Gérer les utilisateurs** : Cette fonction permet à l'administrateur d'ajouter, de modifier, de supprimer, de consulter et de chercher les utilisateurs et leurs droits d'accès (par responsabilité).
- **Approuver les clients** : Cette fonction permet à l'administrateur d'approuver les nouveaux clients.

- **Suivre la commande** : Cette fonctionnalité permet à l'administrateur et au client de consulter l'état de la commande. Elle permet de visualiser son évolution.
- **Passer un bon de commande** : Cette fonctionnalité permet client de passer un bon de commande à la société.
- **Créer un compte** : Cette fonctionnalité permet au client de créer un compte qui sera vérifié par la suite par l'administrateur.
- **Gérer les articles** : Elle permet d'ajouter, de supprimer, de modifier, de consulter et de chercher un article. Elle est réalisée par le responsable de nomenclature.
- **Gérer les achats** : Elle permet d'ajouter, de supprimer, de modifier, de consulter et de chercher une demande d'achat. Elle est réalisée par le responsable d'achat.
- **Gérer la production** : Elle permet d'ajouter, de modifier, de consulter et de chercher une production d'une commande. Elle est réalisée par le responsable de production.
- **Gérer la finition** : Elle permet d'ajouter, de modifier, de consulter et de chercher une finition d'une commande. Elle est réalisée par le responsable de production.
- **Gérer le lavage** : Elle permet d'ajouter, de modifier, de consulter et de chercher un lavage d'une commande. Elle est réalisée par le responsable de lavage.
- **Gérer l'expédition** : Elle permet d'ajouter, de modifier, de consulter et de chercher une expédition d'une commande. Elle est réalisée par le responsable d'export. Cette fonction ne peut être réalisée que si cette fonction est effectuée :
- **Gérer les paiements** : Elle permet d'ajouter, de modifier, de consulter et de chercher une facture d'une commande. Elle est réalisée par le responsable d'export.

### II.2.2 Spécification des besoins du client

Le diagramme de cas d'utilisation du client de *la figure II.2* permet de regrouper tous les cas d'utilisation relatifs au client afin de représenter les fonctions de celui-ci.



*Figure II.2- Diagramme de cas d'utilisation du client*

### II.2.2.1 Cas d'utilisation « Créeer un compte »

*Le tableau II-1 montre la description textuelle du cas d'utilisation «Créer un compte »*

|                       |  |
|-----------------------|--|
| Titre                 | Créer un compte  |
| Acteur                | Client   |
| Résumé                | Chaque client doit créer un compte.  |
| Précondition          | <ul style="list-style-type: none"><li>- Le système doit être ouvert.</li><li>- La connexion doit être effectuée.</li></ul>   |
| Scénario Nominal      | <ol style="list-style-type: none"><li>1. Le client clique sur le bouton s'inscrire.</li><li>2. Le système affiche le formulaire de création d'un compte</li><li>3. Le client doit remplir le formulaire.</li><li>4. Le système vérifie les champs.</li><li>5. Le client clique sur le bouton envoyer pour confirmer l'envoie.</li><li>6. Le système affiche « Votre demande a été envoyé ».</li><li>7. Ouverture d'Outlook. Le client saisit l'email de l'administrateur et clique sur envoyer.</li><li>8. Retour au système : Le client clique sur le bouton Enregistrer.</li><li>9. Le Système affiche « Votre demande a été enregistré ».</li></ol> |
| Scénarios alternatifs | <ul style="list-style-type: none"><li>- Le client peut cliquer sur le bouton annuler pour annuler l'envoi.</li></ul>   |
| Scénarios d'exception | <ol style="list-style-type: none"><li>1. Retour à l'étape 3 du scénario nominal : Si le client a saisi des données invalides : un message d'erreur sera affiché sur chaque champ.</li></ol>  |
| Postcondition         | <ul style="list-style-type: none"><li>- Scénario nominal : Le message « Votre demande a été enregistré » est affiché.</li><li>- Scénario nominal : Le système se redirige vers Outlook.</li><li>- Scénario d'exception : Message d'erreur sur chaque champ.</li></ul>  |

*Tableau II-1 - La description textuelle de cas d'utilisation Créeer un compte*

### II.2.2.2 Cas d'utilisation « Passer un bon de commande »

*Le tableau II-2 montre la description textuelle du cas d'utilisation « Passer un bon de commande »*

|                       |   |
|-----------------------|---|
| Titre                 | Passer un bon de commande   |
| Acteur                | Client  |
| Résumé                | Chaque client est demandé de passer un bon de commande.   |
| Précondition          | <ul style="list-style-type: none"> <li>- Le système doit être ouvert.</li> <li>- La connexion doit être effectuée.</li> <li>- Le client doit s'identifier.</li> </ul>   |
| Scénario Nominal      | <ol style="list-style-type: none"> <li>1. Le client demande de passer un bon de commande.</li> <li>2. Le système affiche le formulaire de passer une commande.</li> <li>3. Le client doit remplir le formulaire d'une commande.</li> <li>4. Le client peut écrire un commentaire.</li> <li>5. Le système vérifie les champs et la validité des données.</li> <li>6. Le client clique sur le bouton envoyer pour confirmer l'envoie.</li> <li>7. Ouverture de l'Outlook. Le client doit choisir son fichier technique qui correspond à la commande.</li> <li>8. Le client clique sur envoyer.</li> </ol> |
| Scénarios alternatifs | <ul style="list-style-type: none"> <li>- Le client peut cliquer sur le bouton annuler pour annuler l'envoi.</li> </ul>  |
| Scénarios d'exception | <ol style="list-style-type: none"> <li>1. Retour à l'étape 3 du scénario nominal : Si le client n'a pas rempli le formulaire: un message d'erreur sera affiché.</li> <li>2. Retour à l'étape 2 du scénario nominale.</li> </ol>   |
| Postcondition         | <ul style="list-style-type: none"> <li>- Scénario alternatifs : Le système redirige l'utilisateur vers le menu principal.</li> <li>- Scénario d'exception : Message d'erreur sur chaque champ.</li> </ul>   |

*Tableau II-2 - La description textuelle de cas d'utilisation **Passer un bon de commande***

### II.2.2.3 Cas d'utilisation « Suivre la commande »

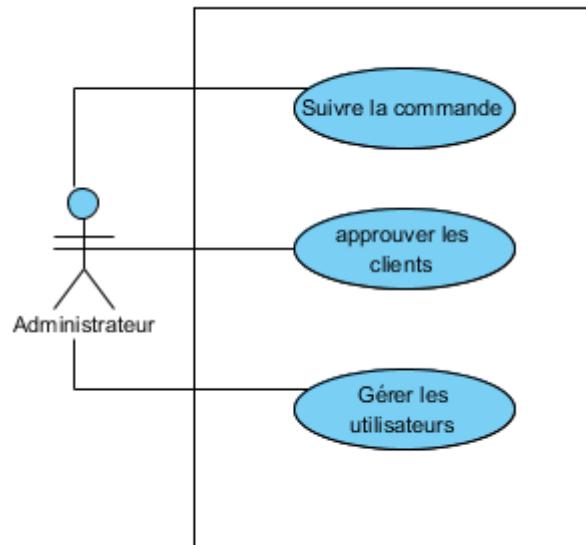
*Le tableau II-3* montre la description textuelle du cas d'utilisation « Suivre la commande »

|                      |   |
|----------------------|---|
| Titre                | Suivre la commande  |
| Acteur               | Client, Administrateur  |
| Résumé               | Le client ou l'administrateur peut suivre l'évolution de la commande.   |
| Précondition         | <ul style="list-style-type: none"> <li>- Le système doit être ouvert.</li> <li>- La connexion doit être effectuée.</li> <li>- Le client ou l'administrateur doit être connecté à son compte.</li> </ul>   |
| Scénario Nominal     | <ol style="list-style-type: none"> <li>1. L'utilisateur saisit l'identifiant de la commande et clique sur le bouton Chercher.</li> <li>2. Le système exécute la recherche et affiche l'état de la commande cherchée.</li> </ol>   |
| Scénario d'exception | <ol style="list-style-type: none"> <li>1. Si le client ou l'administrateur vient de saisir un identifiant incorrect : un message d'erreur sera affiché : « Vérifier le code de la commande »</li> </ol>   |
| Postcondition        | <ul style="list-style-type: none"> <li>- Scénario nominal : Le client ou l'administrateur voit l'évolution de la commande demandée.</li> <li>- Scénario d'exception : Un message d'erreur sera affiché si l'identifiant est incorrect: « Vérifier le code de la commande »</li> </ul> |

*Tableau II-3 - Description textuelle du cas d'utilisation Suivre la commande*

### II.2.3 Spécification des besoins de l'administrateur

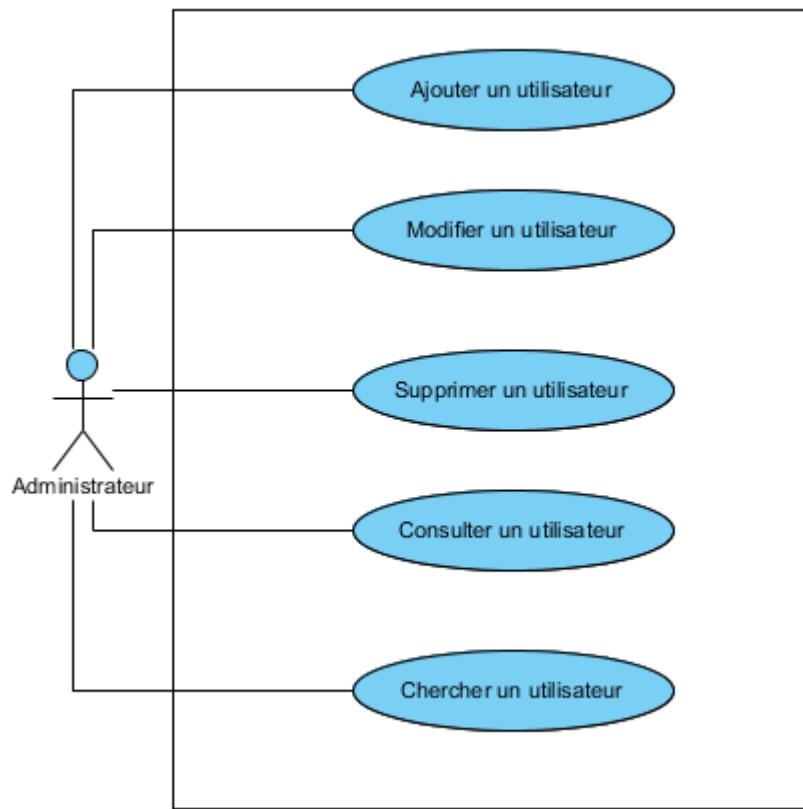
Le diagramme de cas d'utilisation de l'administrateur de *la figure II.3* permet de regrouper tous les cas d'utilisation relatifs à l'administrateur afin de représenter les fonctions de celui-ci.



*Figure II.3 - Diagramme de cas d'utilisation de l'administrateur*

### II.2.3.1 Le diagramme de cas d'utilisation Gérer les utilisateurs

Dans cette partie, nous détaillons les cas d'utilisation relatifs à l'administrateur. Le diagramme de cas d'utilisation de *la figure II.4* permet de détailler le cas d'utilisation «Gérer les utilisateurs» relatifs à l'administrateur.



*Figure II.4 - Diagramme de cas d'utilisation de Gérer les utilisateurs*

### **II.2.3.1.1 Cas d'utilisation « Ajouter un nouvel utilisateur »**

*Le tableau II-4 montre la description textuelle du cas d'utilisation « Ajouter un nouvel utilisateur»*

|                      |  |
|----------------------|--|
| Titre                | Ajouter un nouvel utilisateur  |
| Acteur               | Administrateur   |
| Résumé               | L'administrateur peut ajouter un nouvel utilisateur.   |
| Précondition         | <ul style="list-style-type: none"> <li>- L'administrateur doit s'identifier.</li> </ul>  |
| Scénario Nominal     | <ol style="list-style-type: none"> <li>1. L'administrateur demande d'ajouter un nouvel utilisateur.</li> <li>2. Le système affiche le formulaire d'ajout d'un nouvel utilisateur.</li> <li>3. L'administrateur saisit les informations relatives à chaque utilisateur.</li> <li>4. Le système vérifie les champs.</li> <li>5. L'administrateur confirme l'ajout de l'utilisateur.</li> <li>6. Si les champs sont valides, le système enregistre les informations saisies et affiche un message de confirmation : « Utilisateur ajouté avec succès ».</li> <li>7. Redirection vers la liste des utilisateurs</li> </ol> |
| Scénario Alternatif  | <ul style="list-style-type: none"> <li>- L'administrateur peut cliquer sur le bouton annuler pour annuler l'ajout.</li> </ul>  |
| Scénario d'exception | <ol style="list-style-type: none"> <li>1. Retour à l'étape 5 du scénario nominal : Si l'administrateur a saisi les informations relatives à un autre utilisateur déjà existant, un message d'erreur sera afficher « cette identifiant est déjà existant ».</li> <li>2. Retour à l'étape 6 du scénario nominal : si les champs sont invalides : un message d'erreur sera affiché.</li> </ol>  |
| Postcondition        | <ul style="list-style-type: none"> <li>- Scénario nominal : Un message de confirmation : « Utilisateur ajouté avec succès » est affiché.</li> <li>- Scénario d'exception : un message d'erreur est affiché sur chaque champ.</li> <li>- Scénario alternatifs : Le système redirige l'utilisateur le menu principal « La liste des utilisateurs ».</li> </ul>   |

*Tableau II-4 - Description textuelle du cas d'utilisation **Ajouter un nouvel utilisateur***

### II.2.3.1.2 Cas d'utilisation « Modifier un utilisateur »

*Le tableau II-5 montre la description textuelle du cas d'utilisation « Modifier un utilisateur»*

|                      |   |
|----------------------|---|
| Titre                | Modifier un utilisateur   |
| Acteur               | Administrateur  |
| Résumé               | L'administrateur peut modifier les informations relatives à un utilisateur.   |
| Précondition         | - L'administrateur doit s'identifier.   |
| Scénario Nominal     | <ol style="list-style-type: none"> <li>1. Le système affiche la liste des utilisateurs.</li> <li>2. L'administrateur saisit l'identifiant de l'utilisateur à modifier.</li> <li>3. Le système affiche les informations de l'utilisateur cherché.</li> <li>4. L'administrateur clique sur le bouton « Modifier »</li> <li>5. Le système affiche l'interface de modification de l'utilisateur.</li> <li>6. L'administrateur modifie les informations qu'il désire.</li> <li>7. Le système vérifie les informations entrées.</li> <li>8. L'administrateur confirme la modification des informations de l'utilisateur choisi.</li> <li>9. Si la saisie est valide, le système enregistre les informations saisies et affiche un message de confirmation « Les modifications sont appliquées avec succès ».</li> </ol> |
| Scénario Alternatif  | <ul style="list-style-type: none"> <li>- L'administrateur peut cliquer sur le bouton annuler pour annuler la modification.</li> </ul>   |
| Scénario d'exception | <ol style="list-style-type: none"> <li>1. Retour à l'étape 9 du scénario nominal : Si l'administrateur a rempli des champs invalides : un message d'erreur sera affiché.</li> <li>2. Retour à l'étape 2 du scénario nominal : L'administrateur a saisi un identifiant invalide : Le système affiche « Vérifier la CIN de l'utilisateur »</li> <li>3. Si la connexion a échoué : le système affiche « Un problème de connexion ».</li> </ol>   |
| Postcondition        | <ul style="list-style-type: none"> <li>- Scénario nominal : Un message de confirmation « Les modifications sont appliquées avec succès ».</li> <li>- Scénario d'exception : Un message d'erreur est affiché sur chaque champ.</li> <li>- Scénario alternatif : Le système redirige l'utilisateur le menu principal « La liste des utilisateurs ».</li> </ul>  |

*Tableau II-5 - Description textuelle du cas d'utilisation **Modifier un utilisateur***

### **II.2.3.1.3 Cas d'utilisation « Supprimer un utilisateur»**

*Le tableau II-6 montre la description textuelle du cas d'utilisation « Supprimer un utilisateur»*

|                      |  |
|----------------------|--|
| Titre                | Supprimer un utilisateur   |
| Acteur               | Administrateur   |
| Résumé               | L'administrateur peut supprimer un utilisateur.  |
| Précondition         | - L'administrateur doit s'identifier.  |
| Scénario Nominal     | <ol style="list-style-type: none"> <li>1. Le système affiche la liste des utilisateurs.</li> <li>2. L'administrateur saisit l'identifiant de l'utilisateur à supprimer.</li> <li>3. L'administrateur clique sur le bouton « Supprimer ».</li> <li>4. Le système demande à l'administrateur de confirmer sa demande.</li> <li>5. L'administrateur confirme sa demande de suppression.</li> <li>6. Le système supprime l'utilisateur et affiche un message de confirmation : « Un utilisateur est supprimé avec succès ».</li> </ol> |
| Scénario Alternatif  | <ul style="list-style-type: none"> <li>- L'administrateur peut cliquer sur le bouton annuler pour annuler la suppression.</li> </ul>   |
| Scénario d'exception | <ul style="list-style-type: none"> <li>- Retour à l'étape 2 du scénario nominal : L'administrateur a saisi un identifiant invalide : Le système affiche « Vérifier la CIN de l'utilisateur »</li> </ul>  |
| Postcondition        | <ul style="list-style-type: none"> <li>- Scénario nominal : Un utilisateur est supprimé avec succès.</li> <li>- Scénario alternatifs : Le système redirige l'utilisateur le menu principal « La liste des utilisateurs ».</li> </ul>   |

*Tableau II-6 - Description textuelle du cas d'utilisation **Supprimer un utilisateur***

#### II.2.3.1.4 Cas d'utilisation «Consulter un utilisateur»

*Le tableau II-7 montre la description textuelle du cas d'utilisation « Consulter un utilisateur»*

|                      |   |
|----------------------|---|
| Titre                | Consulter un utilisateur  |
| Acteur               | Administrateur  |
| Résumé               | L'administrateur peut consulter un utilisateur.   |
| Précondition         | <ul style="list-style-type: none"><li>- L'administrateur doit s'authentifier.</li><li>- La liste des clients est affichée.</li></ul>  |
| Scénario Nominal     | <ol style="list-style-type: none"><li>1. L'administrateur saisit l'identifiant de l'utilisateur (Ou bien il choisit directement un utilisateur de la liste).</li><li>2. Le système affiche l'utilisateur recherché.</li><li>3. L'administrateur clique sur le bouton consulter.</li><li>4. Le système affiche l'interface contenant les informations relatives à l'utilisateur cherché.</li></ol> |
| Scénario d'exception | <ul style="list-style-type: none"><li>- Retour à l'étape 1 du scénario nominal : L'administrateur a saisi un identifiant invalide : Le système affiche « Vérifier la CIN de l'utilisateur »</li></ul>   |
| Postcondition        | <ul style="list-style-type: none"><li>- Scénario nominal : L'utilisateur est affiché.</li></ul>   |

*Tableau II-7 - Description textuelle du cas d'utilisation **Consulter un utilisateur***

### II.2.3.1.5 Cas d'utilisation «Chercher un utilisateur»

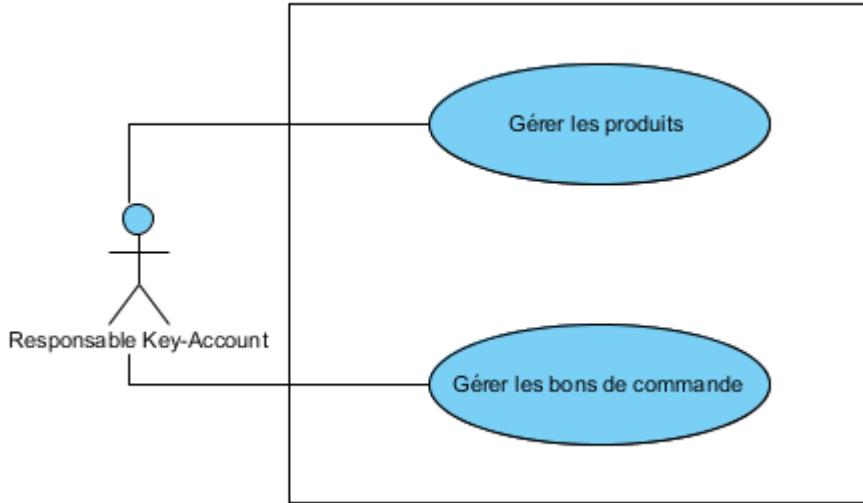
Le tableau II-8 montre la description textuelle du cas d'utilisation «Chercher un utilisateur»

|                      |  |
|----------------------|--|
| Titre                | Chercher un utilisateur  |
| Acteur               | Administrateur   |
| Résumé               | L'administrateur peut chercher un utilisateur.   |
| Précondition         | <ul style="list-style-type: none"><li>- L'administrateur doit s'authentifier.</li><li>- La liste des clients est affichée.</li></ul>   |
| Scénario Nominal     | <ol style="list-style-type: none"><li>1. L'administrateur saisit l'identifiant de l'utilisateur.</li><li>2. L'administrateur clique sur le bouton chercher.</li><li>3. Le système affiche l'utilisateur cherché.</li></ol>   |
| Scénario d'exception | <ol style="list-style-type: none"><li>1. Retour à l'étape 1 du scénario nominal : L'administrateur a saisi un identifiant invalide : Le système affiche « Vérifier la CIN de l'utilisateur »</li><li>2. Si la connexion a échoué : le système informe l'administrateur que la procédure de connexion a échoué.</li></ol> |
| Postcondition        | <ul style="list-style-type: none"><li>- Scénario nominal : L'utilisateur recherché est affiché.</li><li>- Scénario d'exception : Un message d'erreur sera affiché « Vérifier la CIN de l'utilisateur ».</li></ul>  |

*Tableau II-8 - Description textuelle du cas d'utilisation Chercher un utilisateur*

### II.2.4 Spécification des besoins du responsable Key-Account

Le diagramme de cas d'utilisation de *la figure II.5* permet de détailler les cas d'utilisation relatifs au responsable Key-Account.

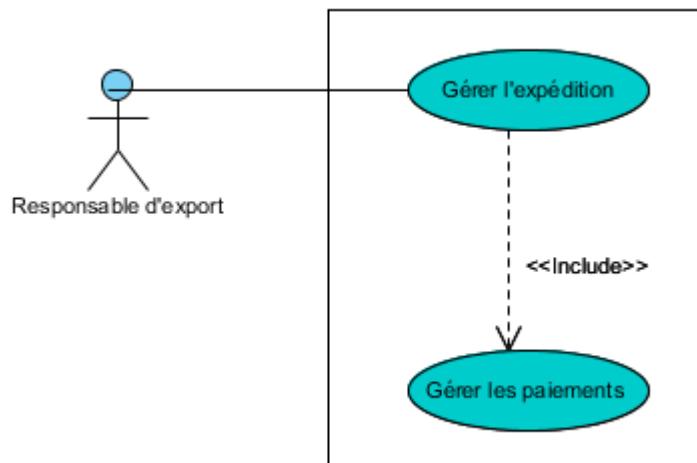


*Figure II.5 - Diagramme de cas d'utilisation du responsable Key-Account*

Les cas d'utilisation « Gérer les produits » et « Gérer les bons de commande » sont composés des sous cas d'utilisation : Ajouter, Modifier, consulter et supprimer et qui sont traités de la même manière que gérer utilisateur illustré par *la figure II.4*.

### II.2.5 Spécification des besoins du responsable de planning-export

Le diagramme de cas d'utilisation de *la figure II.6* permet de détailler les cas d'utilisation relatifs au responsable d'export.



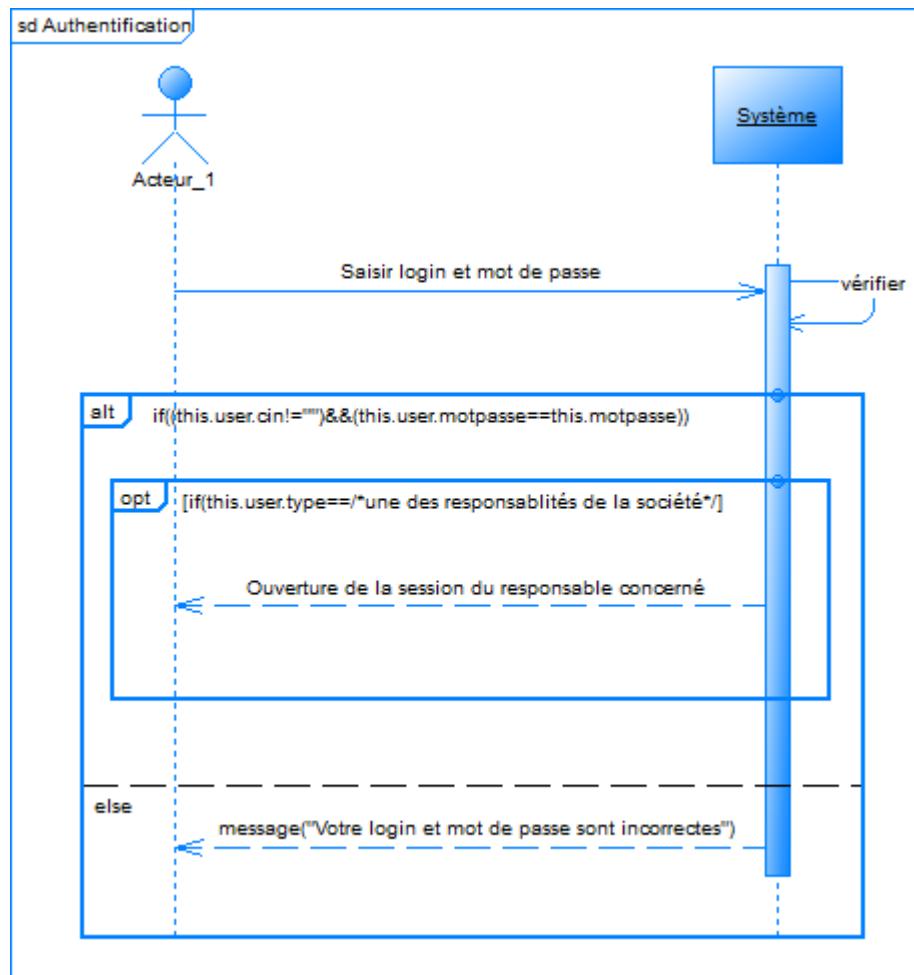
*Figure II.6 - Diagramme de cas d'utilisation du responsable d'export*

## II.2.6 Diagramme de séquence système

Pour décrire les scenarios les plus importants, nous nous sommes basées sur une vue dynamique d'UML étant le diagramme de séquence système.

#### II.2.6.1 Diagramme séquence système du cas d'utilisation s'authentifier

*La figure II-7 montre le diagramme de séquence système du cas d'utilisation «S‘authentifier».*



*Figure II.7 - Diagramme séquence système du cas d'utilisation s'authentifier*

### II.2.6.2 La description textuelle du cas d'utilisation «S'authentifier»

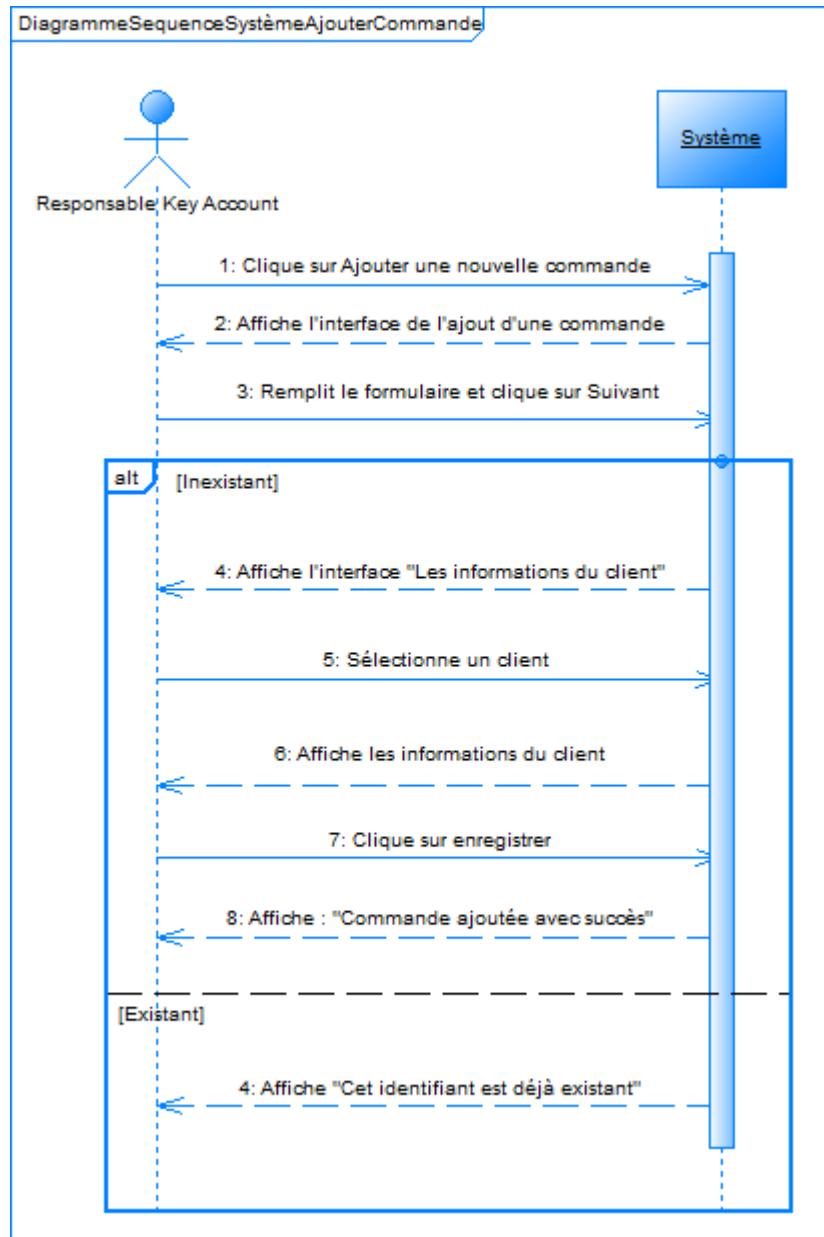
Le tableau II-9 montre la description textuelle du cas d'utilisation «S'authentifier».

|                      |   |
|----------------------|---|
| Titre                | S'authentifier  |
| Acteur               | Tous les utilisateurs de l'application.   |
| Résumé               | Chaque utilisateur doit s'authentifier afin d'accéder à l'application.  |
| Précondition         | - Chaque utilisateur doit avoir un login et un mot de passe.  |
| Scénario Nominal     | <ol style="list-style-type: none"><li>1. Chaque utilisateur doit saisir son login et son mot de passe pour accéder à l'application.</li><li>2. Si l'utilisateur existe : Le système ouvre la session de l'utilisateur concerné.</li><li>3. L'utilisateur accède à sa session.</li></ol> |
| Scénario d'exception | <ol style="list-style-type: none"><li>1. Retour à l'étape 2 du scénario nominal : si l'administrateur a saisi un login ou un mot de passe invalide : un message d'erreur sera affiché.</li></ol>  |
| Postcondition        | - Scénario nominal : Une session est ouverte.   |

Tableau II-9 - Description textuelle du cas d'utilisation S'authentifier

### II.2.6.3 Diagramme séquence système du cas d'utilisation Ajouter une commande

*La figure II-8 montre le diagramme de séquence système du cas d'utilisation «Ajouter une commande».*



*Figure II.8 - Description textuelle du cas d'utilisation Ajouter une commande*

#### II.2.6.4 La description textuelle du cas d'utilisation « Ajouter une commande »

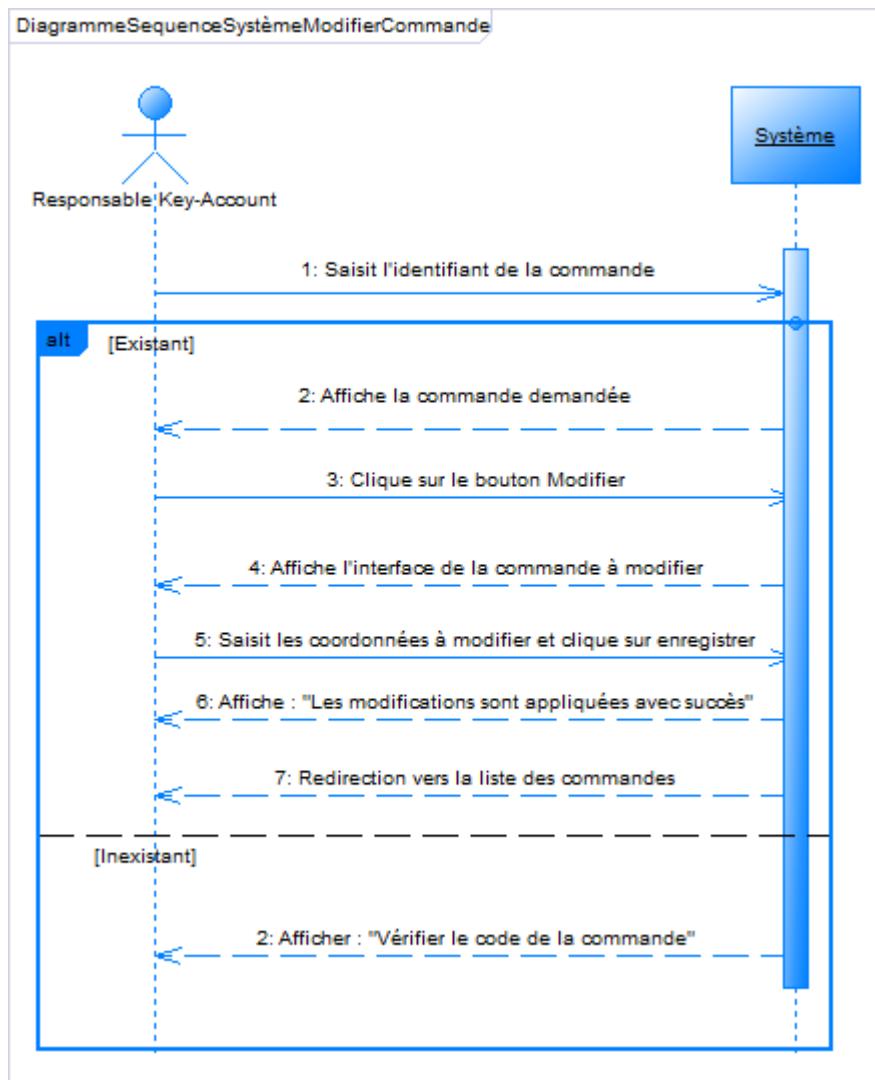
Le tableau II-10 montre la description textuelle du cas d'utilisation «Ajouter une commande».

|                      |   |
|----------------------|---|
| Titre                | Ajouter une commande  |
| Acteur               | Responsable Key-Account, Administrateur.  |
| Résumé               | Le responsable Key-Account ajoute un bon de commande.   |
| Précondition         | - Chaque utilisateur doit s'identifier.   |
| Scénario Nominal     | <ol style="list-style-type: none"><li>1. Le responsable demande d'ajouter une commande.</li><li>2. Le système affiche l'interface de l'ajout d'une commande.</li><li>3. Le responsable remplit le formulaire.</li><li>4. Le système vérifie les champs.</li><li>5. Le responsable clique sur suivant.</li><li>6. Le système affiche l'interface «Les informations du client ».</li><li>7. Le responsable choisit un client.</li><li>8. Le système affiche les informations du client.</li><li>9. Le responsable clique sur le bouton Enregistrer.</li><li>10. Le Système affiche « La commande est ajoutée avec succès ».</li></ol> |
| Scénario d'exception | <ol style="list-style-type: none"><li>1. Retour à l'étape 4 du scénario nominal : si le responsable a rempli des champs invalides : un message d'erreur sera affiché sur chaque champ.</li><li>2. Retour à l'étape 6 du scénario nominal : si le responsable a saisi les informations relatives à une autre commande déjà existante, un message d'erreur sera afficher « cette identifiant est déjà existant » et redirige l'utilisateur vers l'interface contenant la liste des commandes.</li></ol>   |
| Postcondition        | - Scénario nominal : Afficher « La commande est ajoutée avec succès ».  |

Tableau II-10 - Description textuelle du cas d'utilisation **Ajouter une commande**

### II.2.6.5 Diagramme séquence système du cas d'utilisation **Modifier une commande**

*La figure II-9 montre le diagramme de séquence système du cas d'utilisation «Ajouter une commande».*



*Figure II.9 - Description textuelle du cas d'utilisation **Modifier une commande***

### II.2.6.6 La description textuelle du cas d'utilisation «Modifier une commande»

Le tableau II-11 montre la description textuelle du cas d'utilisation «Modifier une commande».

|                      |  |
|----------------------|--|
| Titre                | Modifier une commande  |
| Acteur               | Responsable Key-Account, Administrateur.   |
| Résumé               | Le responsable Key-Account modifie un bon de commande.   |
| Précondition         | <ul style="list-style-type: none"><li>- Chaque utilisateur doit s'identifier.</li><li>- La liste des commandes est affichée.</li></ul>   |
| Scénario Nominal     | <ol style="list-style-type: none"><li>1. Le responsable saisit le code de la commande.</li><li>2. Le système affiche la commande.</li><li>3. Le responsable clique sur le bouton Modifier.</li><li>4. Le système affiche l'interface de modification.</li><li>5. Le responsable fait les modifications et clique sur le bouton Enregistrer.</li><li>6. Le système affiche « Les modifications sont appliquées avec succès ».</li><li>7. Le Système redirige l'utilisateur vers la liste des commandes.</li></ol> |
| Scénario d'exception | <ol style="list-style-type: none"><li>1. Retour à l'étape 1 du scénario nominal : si le responsable a saisi un code invalide : un message d'erreur sera affiché : « Vérifier le code de la commande ».</li></ol>   |
| Postcondition        | <ul style="list-style-type: none"><li>- Scénario nominal : Afficher « Les modifications sont appliquées avec succès ».</li></ul>   |

Tableau II-11 - Description textuelle du cas d'utilisation **Modifier une commande**

### II.2.6.7 Diagramme séquence système du cas d'utilisation « Consulter une commande »

La figure II-10 montre le diagramme de séquence système du cas d'utilisation «Consulter une commande».

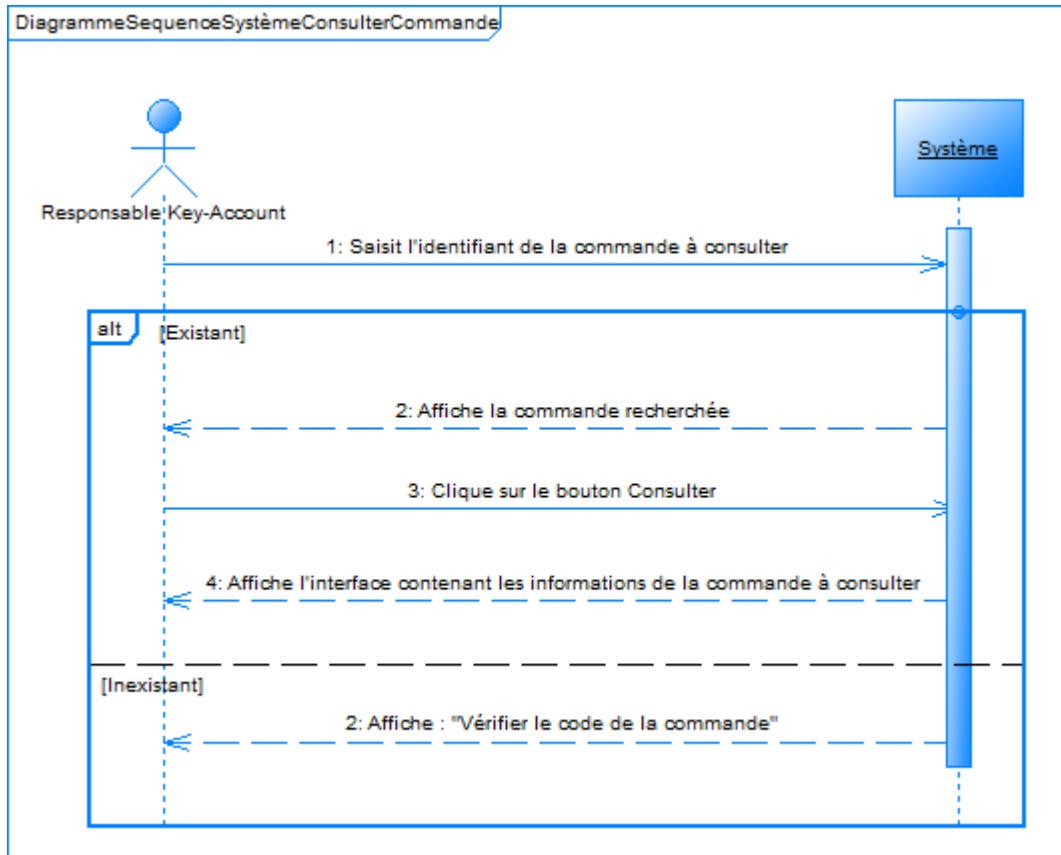


Figure II.10 - Description textuelle du cas d'utilisation **Consulter une commande**

### II.2.6.8 Description textuelle du cas d'utilisation « Consulter une commande »

Le tableau II-12 montre la description textuelle du cas d'utilisation « Consulter une commande ».

|                      |   |
|----------------------|---|
| Titre                | Consulter une commande  |
| Acteur               | Responsable Key-Account, Administrateur.  |
| Résumé               | Le responsable Key-Account consulte un bon de commande.   |
| Précondition         | <ul style="list-style-type: none"><li>- Chaque utilisateur doit s'identifier.</li><li>- La liste des commandes est affichée.</li></ul>  |
| Scénario Nominal     | <ol style="list-style-type: none"><li>1. Le responsable saisit le code de la commande.</li><li>2. Le système affiche la commande.</li><li>3. Le responsable clique sur le bouton Consulter.</li><li>4. Le système affiche l'interface de consultation, contenant les informations de la commande.</li></ol> |
| Scénario d'exception | <ol style="list-style-type: none"><li>1. Retour à l'étape 1 du scénario nominal : si le responsable a saisi un code invalide : un message d'erreur sera affiché : « Vérifier le code de la commande ».</li></ol>  |
| Postcondition        | <ul style="list-style-type: none"><li>- Scénario nominal : Afficher les informations de la commande.</li></ul>  |

Tableau II-12 - Description textuelle du cas d'utilisation **Consulter une commande**

### II.2.6.9 Diagramme séquence système du cas d'utilisation «Supprimer une commande»

*La figure II-11 montre le diagramme de séquence système du cas d'utilisation «Consulter une commande».*

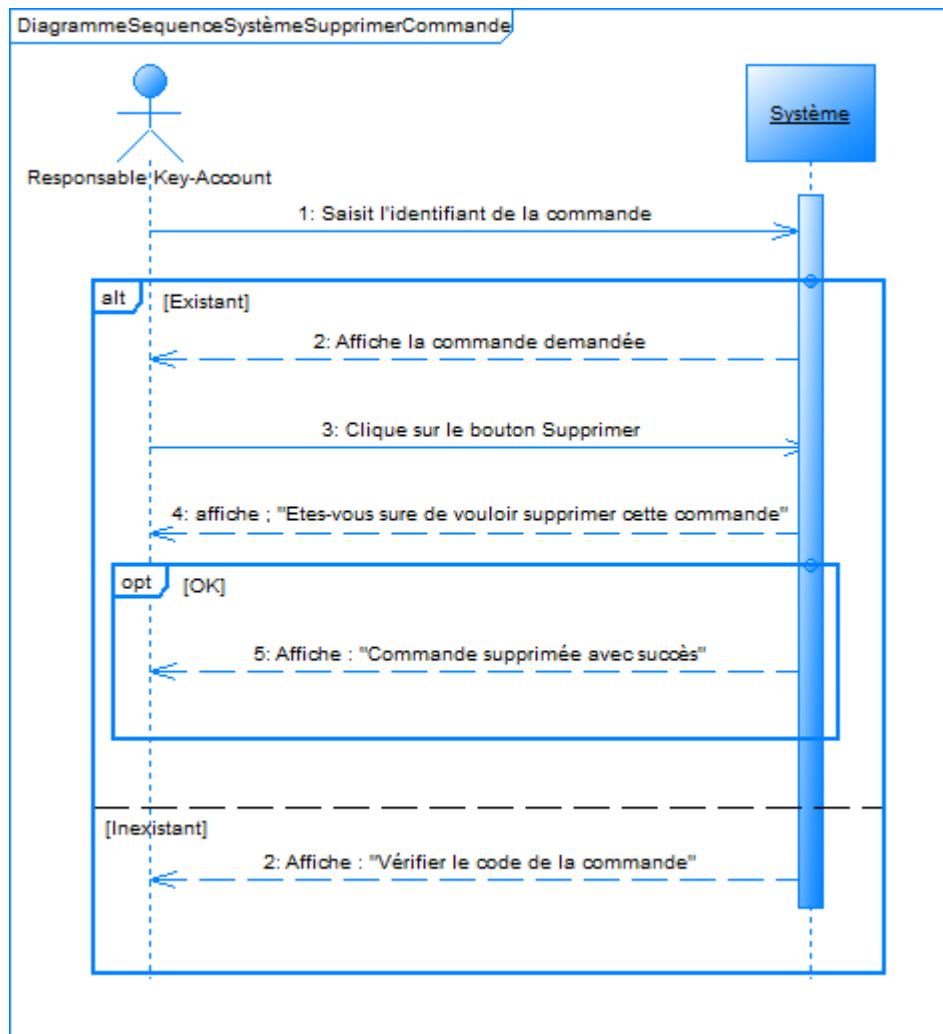


Figure II.11 - Description textuelle du cas d'utilisation **Supprimer une commande**

### II.2.6.10 La description textuelle du cas d'utilisation « Supprimer une commande »

Le tableau II-13 montre la description textuelle du cas d'utilisation «Supprimer une commande».

|                      |  |
|----------------------|--|
| Titre                | Supprimer une commande   |
| Acteur               | Responsable Key-Account, Administrateur.   |
| Résumé               | Le responsable Key-Account supprime un bon de commande.  |
| Précondition         | <ul style="list-style-type: none"><li>- Chaque utilisateur doit s'identifier.</li><li>- La liste des commandes est affichée.</li></ul>   |
| Scénario Nominal     | <ol style="list-style-type: none"><li>1. Le responsable saisit le code de la commande.</li><li>2. Le système affiche la commande.</li><li>3. Le responsable clique sur le bouton Supprimer.</li><li>4. Le système affiche un message de confirmation : « «Etes-vous sûre de vouloir supprimer cette commande ?»</li><li>5. Le responsable clique sur Ok.</li><li>6. Le système affiche : « Commande supprimée avec succès ».</li></ol> |
| Scénario d'exception | <ol style="list-style-type: none"><li>1. Retour à l'étape 1 du scénario nominal : si le responsable a saisi un code invalide : un message d'erreur sera affiché : « Vérifier le code de la commande ».</li></ol>   |
| Scénario Alternatif  | <ul style="list-style-type: none"><li>- L'administrateur peut cliquer sur le bouton annuler pour annuler la suppression.</li></ul>   |
| Postcondition        | <ul style="list-style-type: none"><li>- Scénario nominal : Affiche : « Commande supprimée avec succès ».</li></ul>   |

Tableau II-13 - Description textuelle du cas d'utilisation **Supprimer une commande**

### **II.2.7 Etude du domaine d'un point de vue statique**

Dans cette partie nous allons présenter le diagramme de classe d'analyse de notre système comportant les classes utilisées et les attributs qui les décrivent.

*La figure II.12 présente le diagramme de classe d'analyse de notre système.*

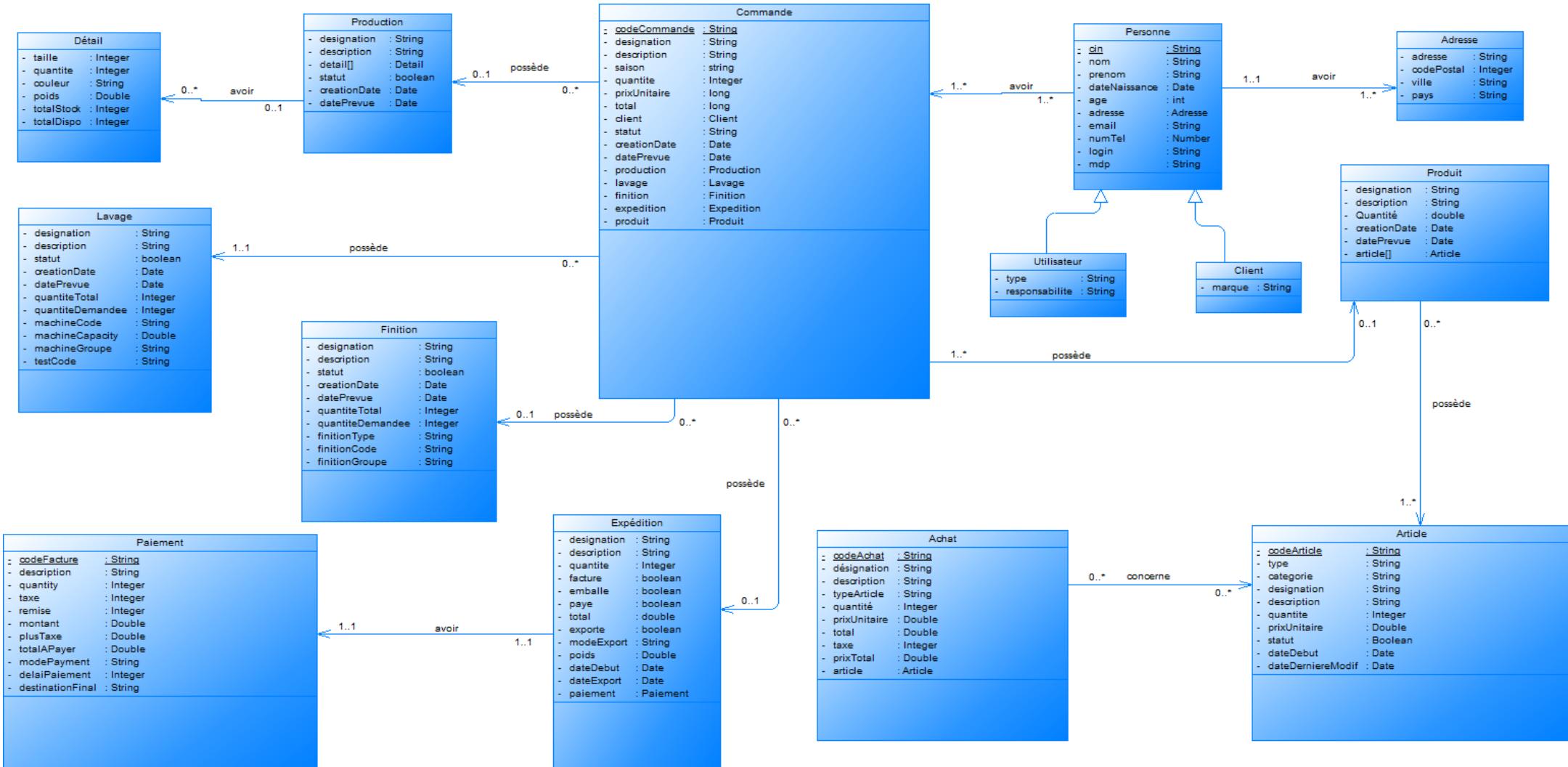


Figure II.12 - Diagramme de classe d'analyse

Dans le diagramme de classe d'analyse représenté ci-dessus, nous avons représenté seulement les classes et les attributs nécessaires pour la mise en place de notre application et son bon fonctionnement

Nous allons reprendre ci-dessous les différentes classes dans le diagramme avec une brève description.

- **Personne** : Regroupe les différentes caractéristiques communes aux différents acteurs humains tels que, le nom, le prénom, le login et le mot de passe.
- **Utilisateur** : Classe qui hérite de la classe personne et qui comprend deux attributs type et responsabilité pour distinguer entre les différents acteurs de la société.
- **Client** : Classe qui hérite de la classe personne et qui comprend l'attribut marque pour caractériser les différents clients.
- **Adresse** : Associer à la classe personne pour décrire les adresses des différentes personnes.
- **Commande** : Cette classe comporte toutes les caractéristiques des différentes commandes.
- **Produit** : Regroupe l'ensemble des produits finis. Comporte les caractéristiques des besoins matériels d'un produit d'une commande.
- **Article** : Tous les articles et ses informations sont enregistrés au niveau de cette classe. Elle sert essentiellement à vérifier l'appartenance des articles à des commandes précises. Classe comportant toutes les informations relatives au Stock : transfert, mouvement et inventaire.
- **Achat**: Regroupe l'ensemble des demandes d'achat. Comporte toutes les commandes d'achat passées. Classe contenant toutes les informations relatives aux commandes d'achat passées (article, fournisseur..).
- **Production** : Comporte les informations relatives à la production des commandes (désignation, Date début de la production, date prévue,...).
- **Détail** : Regroupe les informations quantité-taille d'une production de la commande passée pour faciliter les tâches de colisage.
- **Lavage** : Comporte les caractéristiques de lavage : chaque commande doit passer par un lavage bien déterminé. Ainsi que d'autres caractéristiques (désignation, quantité, les différentes date,...).
- **Finition** : Comporte les caractéristiques de finition : chaque commande possède une finition bien déterminée. Ainsi que d'autres caractéristiques (désignation, quantité, les différentes date,...).
- **Expédition** : Classe regroupant toutes les expéditions (code commande, code facture...). Cette classe contient toutes les informations relatives aux commandes de vente passées. Elle comporte ainsi tous les articles expédiés.
- **Paiements** : Classe regroupant toutes les factures (code facture, montant, mode paiement...). Cette classe contient les informations relatives aux factures des différentes commandes de vente passées.

## II.3 Les maquettes d'interfaces

Le maquettage nous permet d'avoir une idée sur les futures interfaces à créer et les fonctionnalités à y implémenter.

### II.3.1 La maquette d'identification

*La figure II.13* est la première page qui sera affiché à l'utilisateur est l'interface de l'authentification. Cette interface permet à l'utilisateur de s'identifier par la saisie du CIN et du mot de passe.

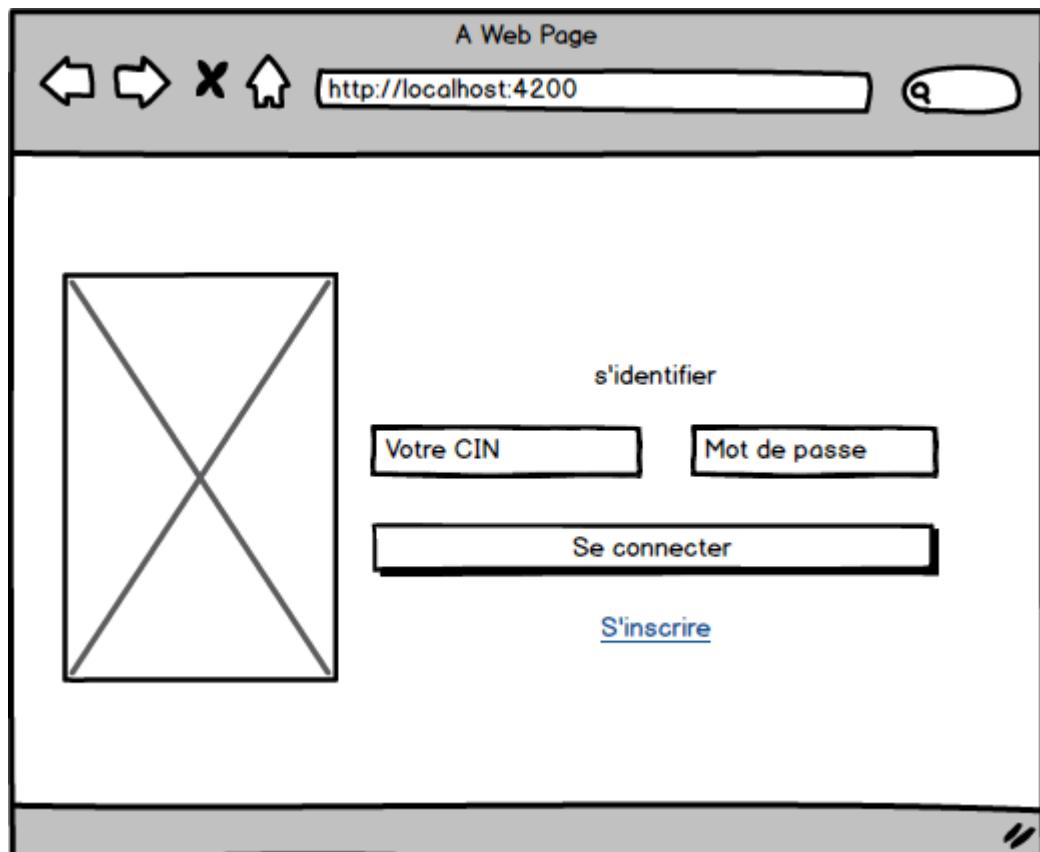


Figure II.13 - Prototype d'identification

### II.3.2 La maquette de gestion des utilisateurs

*La figure II.14* présente l'interface de la gestion des utilisateurs, elle permet à l'administrateur de visualiser la liste des utilisateurs, les modifier, les consulter et les supprimer.

Figure II.14 - Prototype de Gérer les utilisateurs

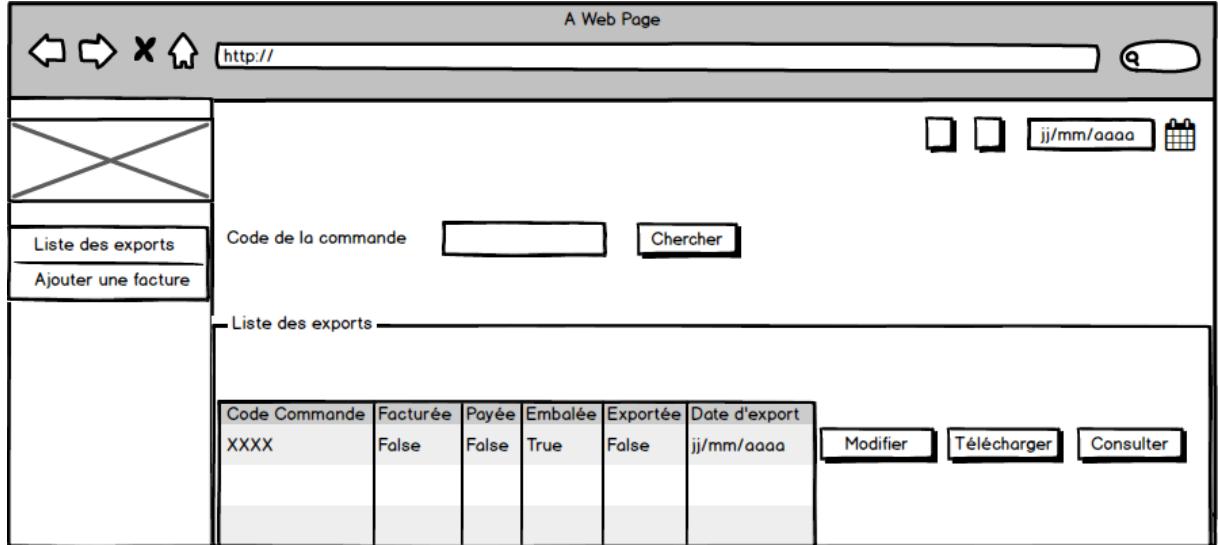
### II.3.3 La maquette de gestion des bons de commande

*La figure II.15 présente l’interface de la gestion des bons de commande, elle permet à l’administrateur en premier lieu de visualiser la liste des commandes, les modifier, les consulter, les valider, et les supprimer.*

Figure II.15 - Prototype de Gérer les bons de commande

### II.3.4 La maquette de gestion de l'expédition

*La figure II.16* présente l'interface de la gestion des exports, elle permet à l'administrateur en premier lieu de visualiser la liste des exports, les modifier, les consulter et les télécharger. De plus elle permet d'ajouter une nouvelle facture relative à un export.



Le prototype de l'interface de gestion des exports est présenté dans une fenêtre de navigateur intitulée "A Web Page". Le menu de gauche contient "Liste des exports" et "Ajouter une facture". Le corps de la page affiche une liste des exports avec les colonnes suivantes : Code Commande, Facturée, Payée, Embalée, Exportée, Date d'export. Pour l'enregistrement XXXX, les valeurs sont : Facturée = False, Payée = False, Embalée = True, Exportée = False, Date d'export = jj/mm/aaaa. Des boutons "Modifier", "Télécharger" et "Consulter" sont disponibles pour chaque ligne. En haut à droite, il y a des boutons pour imprimer, enregistrer et une barre de recherche avec un champ "jj/mm/aaaa" et un calendrier.

Figure II.16 Prototype de Gérer l'expédition

## Conclusion

Les différentes fonctionnalités de notre système sont ainsi spécifiées. Ce qui nous a permis de distinguer le rôle de chaque acteur et de présenter les interactions de chacun avec notre système. Dans le prochain chapitre, nous pouvons passer à la phase de conception de notre application.

# Chapitre III. Conception

---

Après une analyse et spécification des besoins, nous effectuons une conception globale et détaillée de notre projet. Cette phase nous permet de tracer une meilleure stratégie d'implémentation des besoins. De ce fait, ce chapitre présente une étape intermédiaire entre le résultat de la spécification des besoins et l'implémentations.

La première partie de ce chapitre sera consacrée au choix de l'architecture logicielle de l'application. Ensuite, et afin d'établir une description détaillée nous présenterons les diagrammes de classes de conception et des diagrammes de communication.

## III.1 Conception global

### III.1.1 Architecture physique

Notre application se repose sur une architecture 3-tiers, elle vise à modéliser l'application comme étant un assemblage de trois couches :

- **Couche1** : Le navigateur web : Conteneur d'application et demandeur de ressources. Il représente la couche présentation associé au client léger qui est un navigateur Web. C'est la partie la plus immédiatement visible pour l'utilisateur. Ce navigateur permettra aux demandeurs de ressources et à tout utilisateur de la plateforme d'accéder à l'application via internet.
- **Couche 2** : Le serveur de l'application : Il représente la couche fonctionnelle qui symbolise la logique applicative qui constitue les traitements métier nécessaires afin de le rendre exploitable par chaque utilisateur.
- **Couche 3** : Le serveur de base de données : Il représente la couche métier et permet le stockage et l'accès aux données. Il a pour but de conserver une quantité plus ou moins importante de données de façon structurée.

Le principe d'une architecture trois-tiers est relativement simple: il consiste à séparer la réalisation des trois parties vues précédemment (stockage des données, logique applicative, présentation). Cette séparation signifie qu'il est possible de déployer chaque partie sur un serveur indépendant, toutefois cela n'est pas obligatoire. [3]

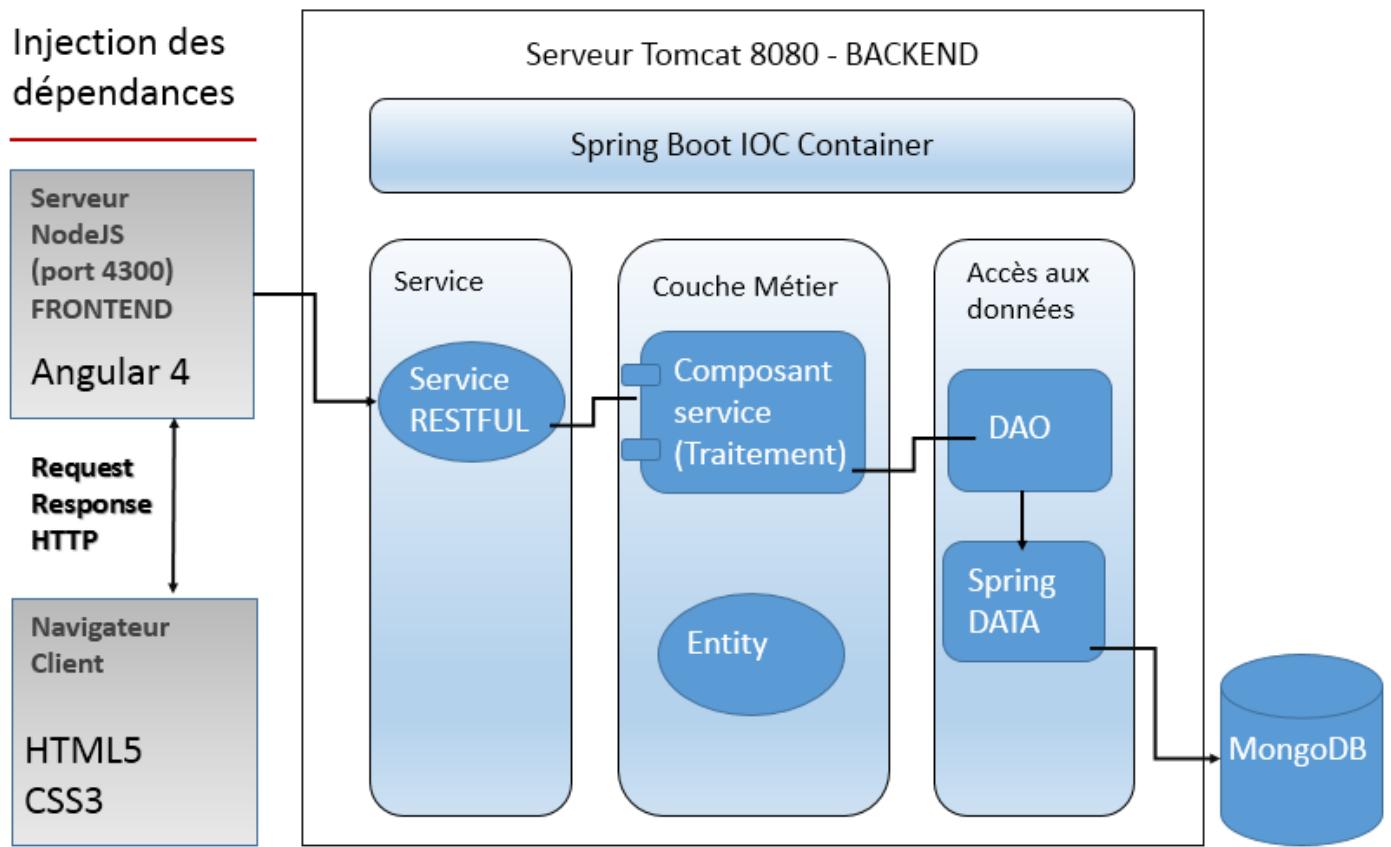


Figure III.1 - Architecture 3-tiers de notre application

### III.1.2 Architecture logique

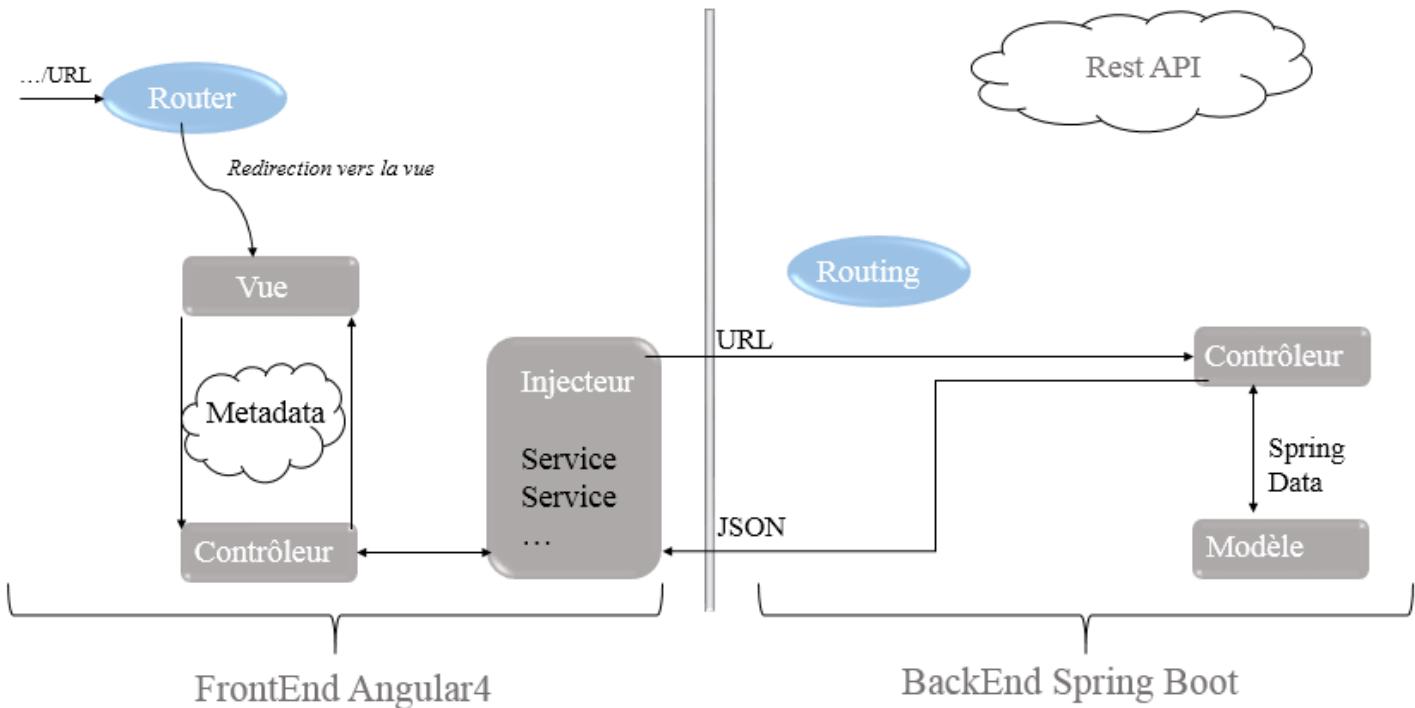


Figure III.2 - Architecture logique de l'application

Notre application web est composée de deux projets qui sont, la partie FrontEnd et la partie Backend. La première partie est réalisée avec le Framework Angular4 qui consomme les Api Rest préparées par la deuxième partie qui est réalisée avec Spring Boot.

La figure III.2 illustre l'architecture logique de ces deux parties et la relation entre elles, où :

- **Router** : est un composant Angular responsable de la redirection et la pagination entre les pages.
- **Vue, contrôleur et service** : Sont des composants Angular que nous aurons expliqués dans le chapitre IV.
- **Metadata** : Indique le comportement d'une classe (contrôleur, directive ...)
- **Routing** : C'est un mécanisme de routage qui définit quel contrôleur à invoquer à partir d'URI.
- **Contrôleur** : C'est une classe contenant le logique métier, qui communique avec le modèle via Spring Data que nous aurons expliqués dans le chapitre IV.

### III.1.3 Le Patron de conception MVC

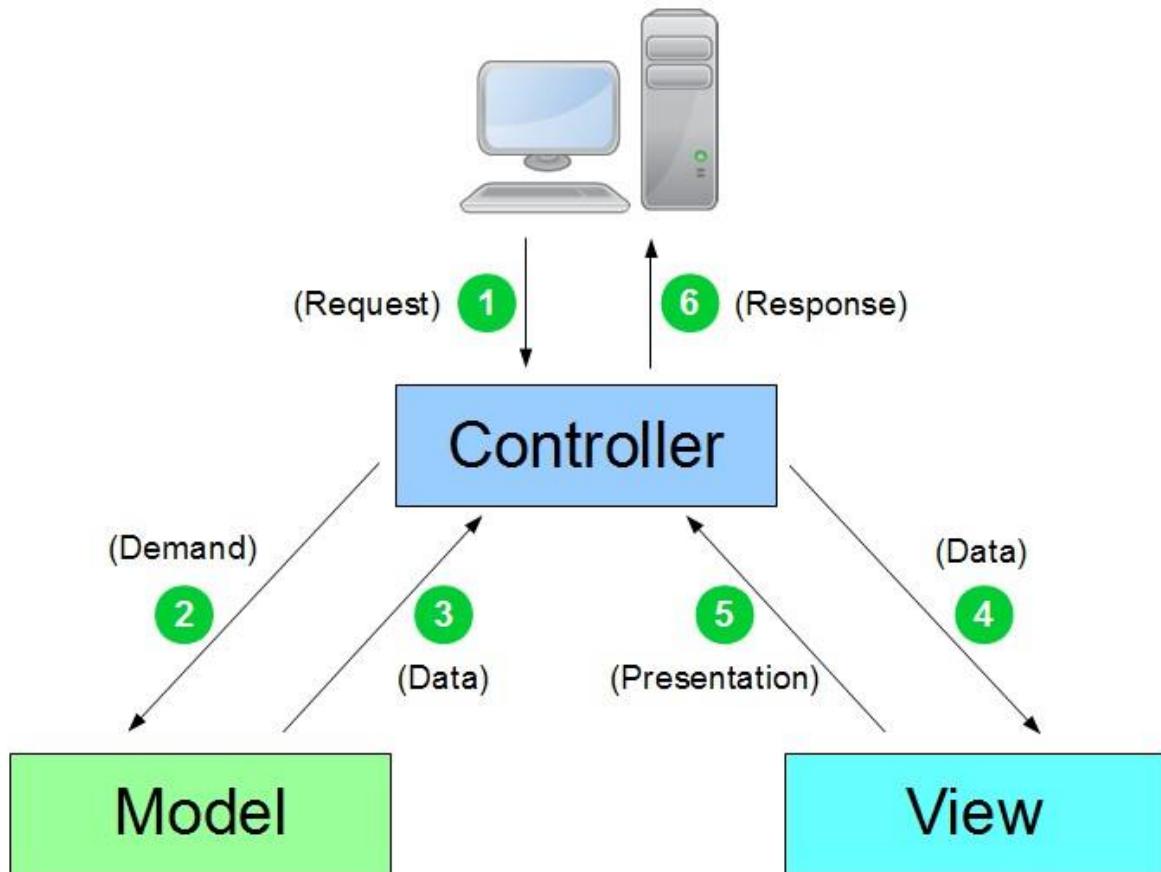


Figure III.3 - Le patron de conception MVC

Les développeurs ont aujourd’hui tendance à utiliser les bonnes pratiques de développement qu’on appelle les Patrons de conception ou Design Pattern (en anglais). Un design pattern ou un modèle de conception est en quelque sorte une ligne de conduite qui permet de décrire les grandes lignes d’une solution. Le modèle MVC découpe littéralement l’application en couches distinctes et de ce fait impacte très fortement l’organisation du code. MVC signifie Modèle – Vue – Contrôleur, en gros quand on développe avec le MVC on segmente son code en trois parties ou couches, chaque couche ayant une fonction bien précise. [4]

- **La couche View :** Notre partie présentation est un projet Angular4 totalement séparé du modèle et contrôleur. Cette partie communique avec le contrôleur en échangeant des données JSON.
- **La couche Controller :** Classe JAVA contenant la logique de l’application, il envoie le flux de données de la vue au modèle et vice versa.
- **La couche Model :** classe JAVA contentant les attributs, les getters, les setters et les constructeurs d’un objet à manipuler.

## III.2 Modélisation du système

### III.2.1 Le diagramme de package

Nous proposerons dans ce qui suit le diagramme de package de notre application présenté par *la figure III.4* :

- ✓ La couche Model regroupe toutes les classes qui accèdent aux services et récupère les informations nécessaires.
- ✓ La couche View (présentation) : regroupe toutes les interfaces graphiques relatives à notre application.
- ✓ La couche Control : regroupe l'ensemble des contrôleurs utilisé pour réaliser les traitements.

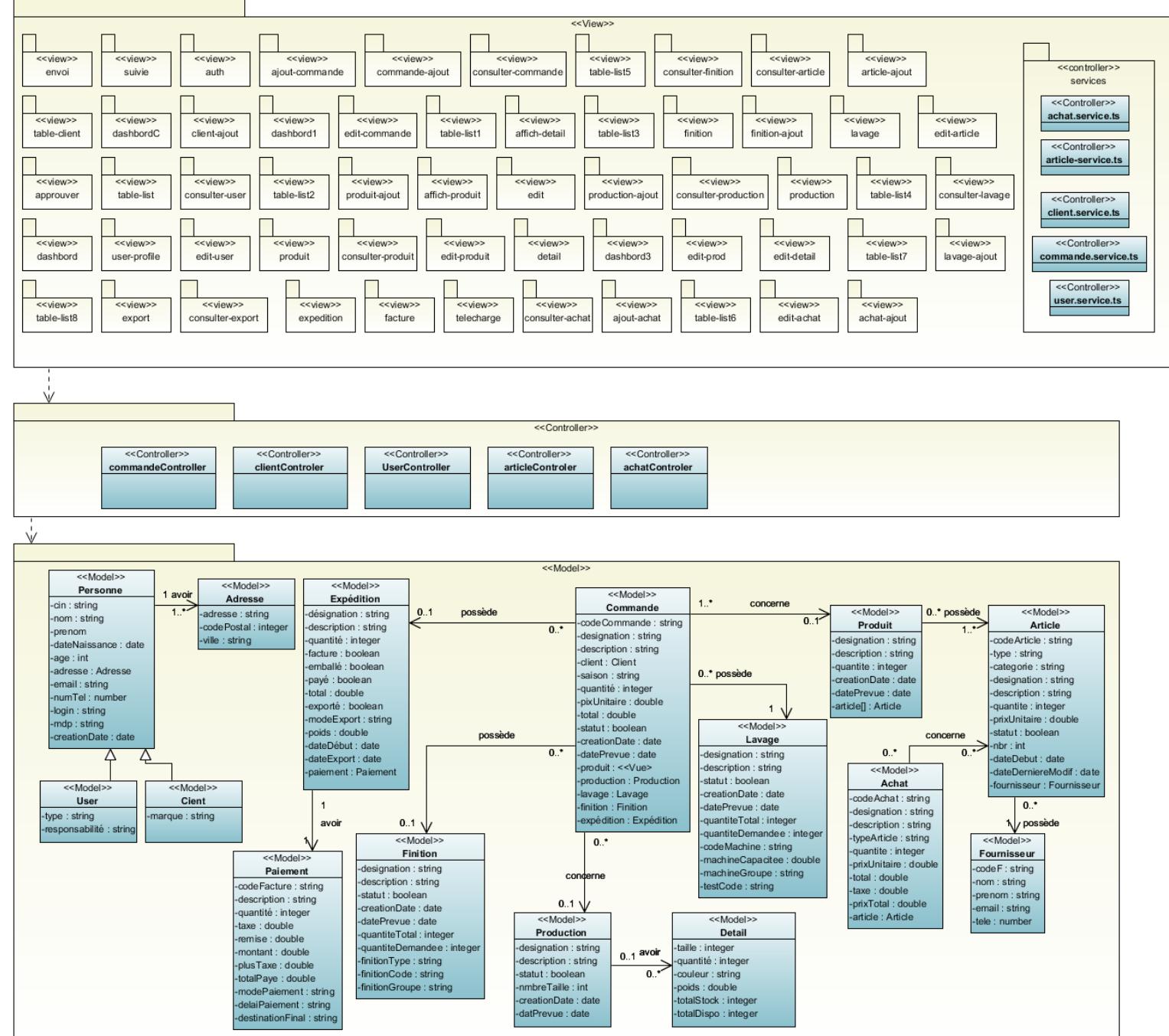


Figure III.4 - Diagramme de packages

### III.3 Conception détaillée

Le diagramme de classes est important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, les modèles de la couche métiers et l'interface avec l'utilisateur. Nous utiliserons alors une implémentation de l'architecture 3-tiers, le pattern Modèle-Vue-Contrôleur (MVC).

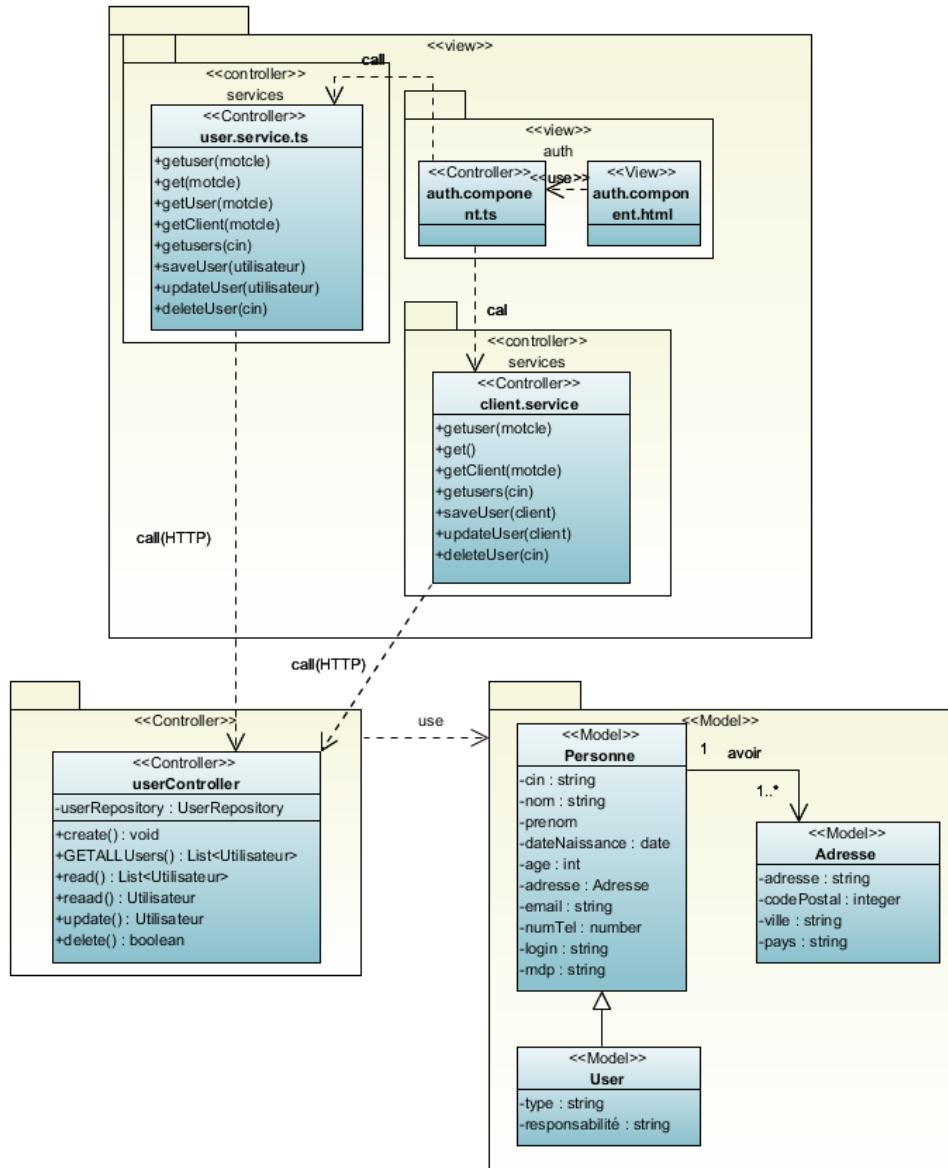
Nous allons élaborer dans cette partie les diagrammes de classes de conception de chaque cas d'utilisation du système. Les indications suivantes sont valables pour tous les diagrammes de classes de conception :

- Les paquetages « Model » et « Controller » sont aux niveaux du Backend (Projet Spring Boot) et le paquetage « View » désigne la partie FrontEnd (Projet Angular 4).
- Dans le paquetage « View », la classe « Controller » dont le nom se termine par « .component.ts » désigne une classe contrôleur d'Angular, et la classe dont le nom se termine par « .service.ts » désigne une classe service d'Angular.
- La relation « use » indique qu'une première classe utilise des données fournis par une deuxième classe.
- La relation « call » entre le paquetage « View » et le paquetage « Controller » désigne un appel à un service Web.
- Au niveau du FrontEnd, le stéréotype « View » est réservé pour les pages de présentation (vues), « Controller » indique qu'il s'agit d'un contrôleur (classe TypeScript dans la partie vue)
- Au niveau du Backend, le stéréotype « Controller » indique qu'il s'agit d'un contrôleur (classe JAVA dans la partie contrôleur), et « Model » présente les classes modèles qui sont générés à partir de la base de données. Chaque classe modèle contient les attributs et les méthodes : Getters & Setters de chaque attribut.

### III.3.1 Diagramme de classe de conception du cas d'utilisation

#### « S'identifier »

Le diagramme de *la figure III.5* représente le diagramme des classes détaillées du cas d'utilisation « S'identifier ».



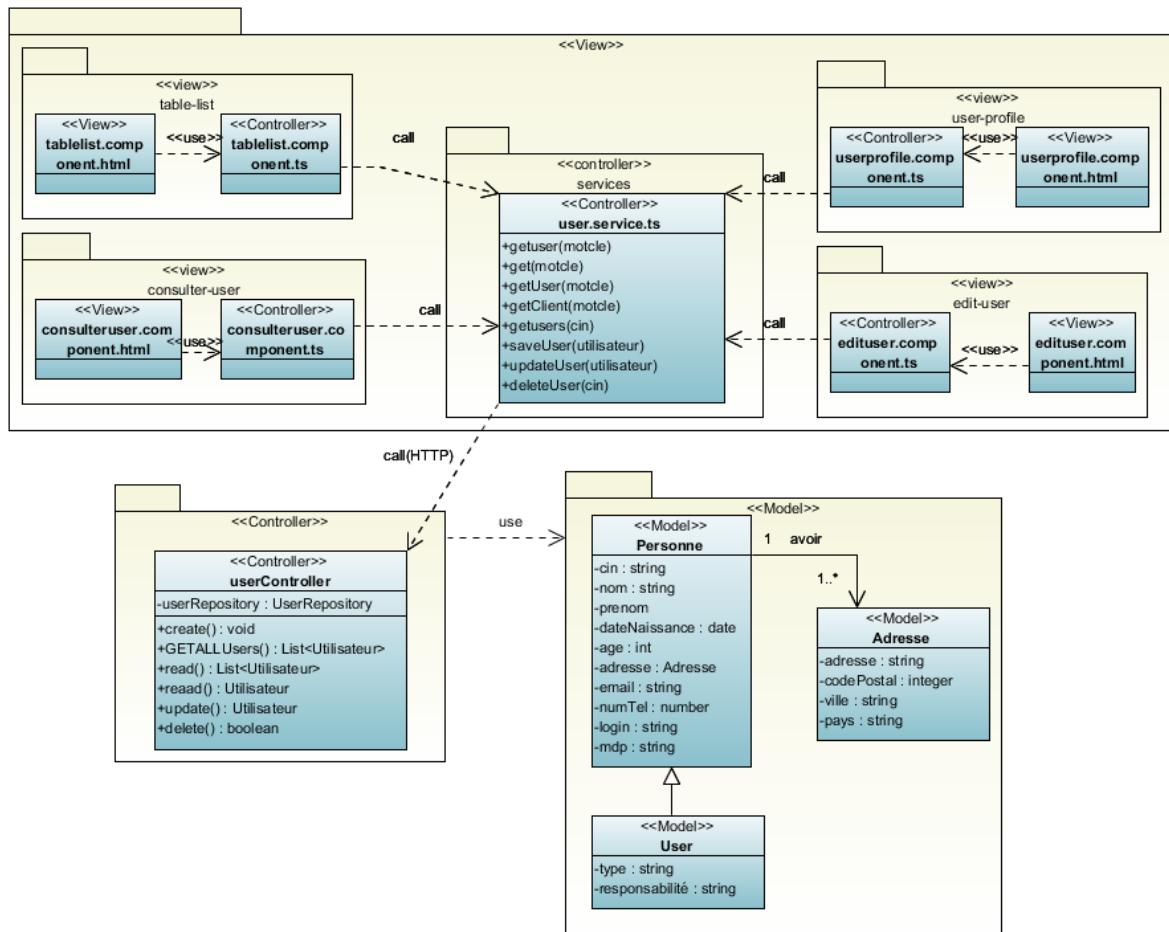
*Figure III.5 - Diagramme de classe de conception du cas d'utilisation « S'identifier »*

Chaque utilisateur de l'application doit s'identifier en saisissant son CIN et son mot de passe afin d'y accéder. Cette opération est effectuée via la vue principale « auth ».

Ces opérations sont effectuées via le service du FrontEnd `user.service` et `client.service` pour vérifier s'il s'agit d'un responsable de la société ou bien d'un client, qui fait un appel au contrôleur du Backend `userController` et `clientController` via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.2 Diagramme de classe de conception du cas d'utilisation « Gérer les utilisateurs»

Le diagramme de *la figure III.6* représente le diagramme des classes détaillées du cas d'utilisation « Gérer les utilisateurs».



*Figure III.6 - Diagramme de classe de conception du cas d'utilisation Gérer les utilisateurs*

Les utilisateurs disponibles sont affichés sur la vue principale table-list et à partir de cette vue l'administrateur peut rechercher ou supprimer un utilisateur. L'affichage des détails des utilisateurs s'effectue à partir de la vue consulter-user. L'ajout d'un nouvel utilisateur s'effectue à partir de la vue user-profile. La modification d'un utilisateur s'effectue à partir de la vue edit-user.

Ces opérations sont effectuées via le service du FrontEnd user.service qui fait un appel au contrôleur du Backend UserController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.3 Diagramme de classe de conception du cas d'utilisation «Créer un compte»

Le diagramme de *la figure III.7* représente le diagramme des classes détaillées du cas d'utilisation «Créer un compte (Client)».

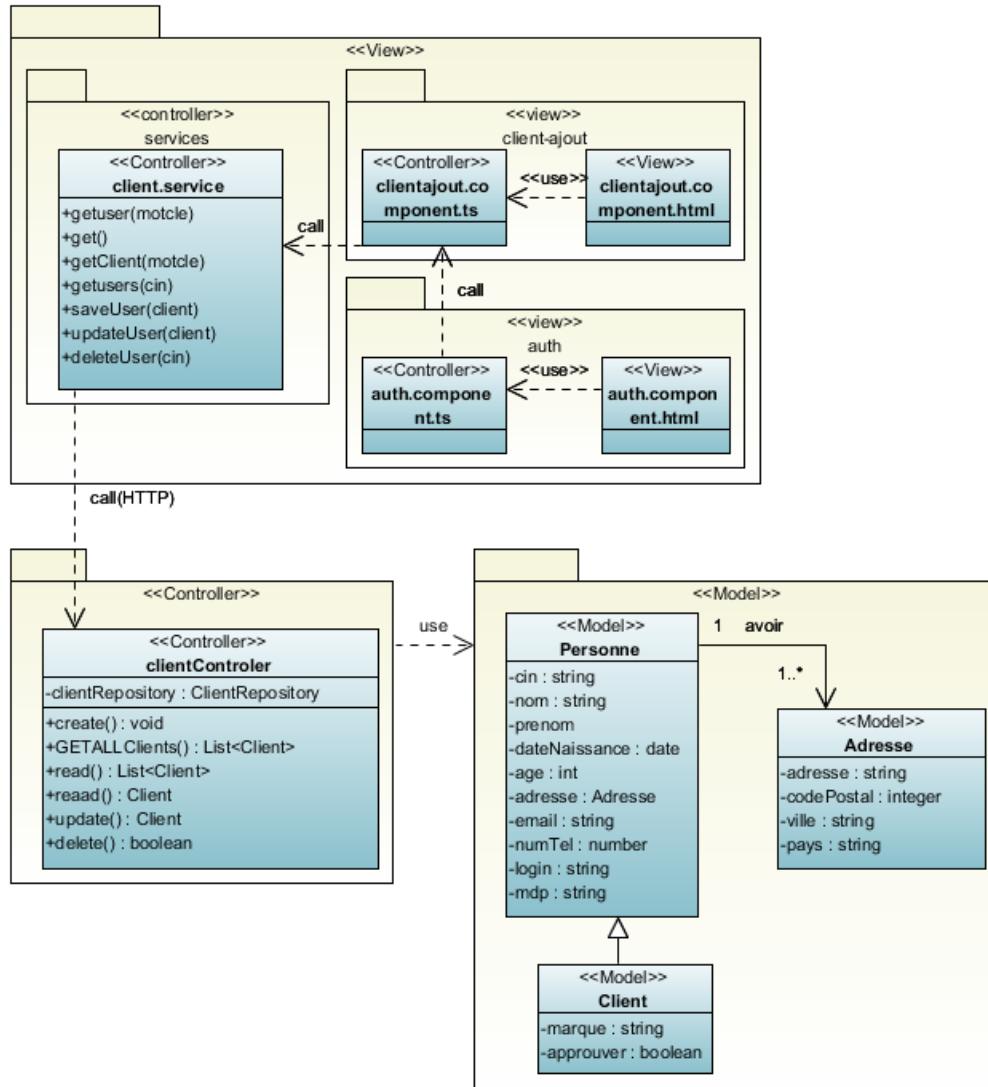


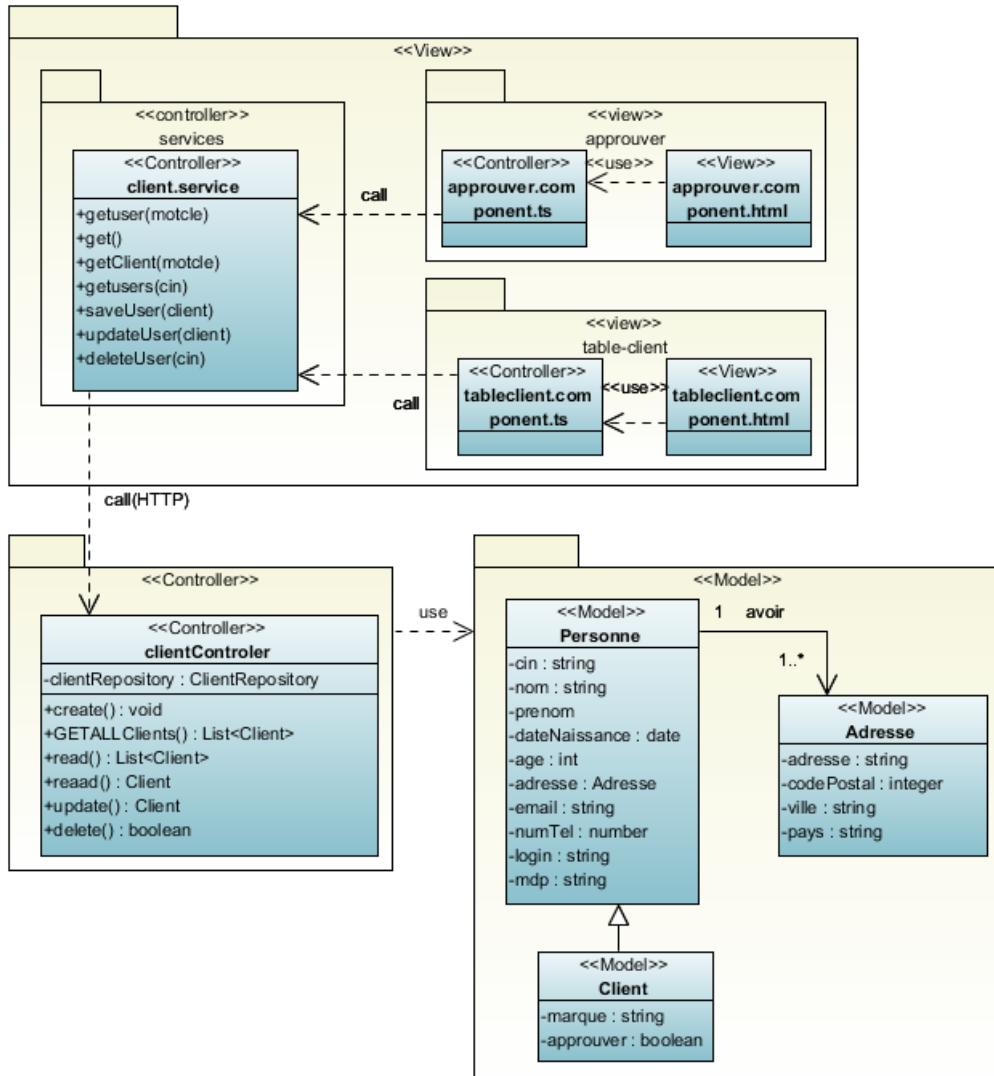
Figure III.7 - - Diagramme de classe de conception du cas d'utilisation Crer un compte (Client)

A partir de la vue principale auth, le client demande de créer un compte (s'inscrire) en remplissant le formulaire qui sera affiché sur la vue client-ajout. La création des comptes s'effectue via le contrôleur qui utilise une instance de la classe.

Cette opération est effectuée via le service du FrontEnd `client.service` qui fait un appel au contrôleur du Backend `client` via le protocole HTTP.

### III.3.4 Diagramme de classe de conception du cas d'utilisation « Approuver un client »

Le diagramme de *la figure III.8* représente le diagramme des classes détaillées du cas d'utilisation « Approuver un client »

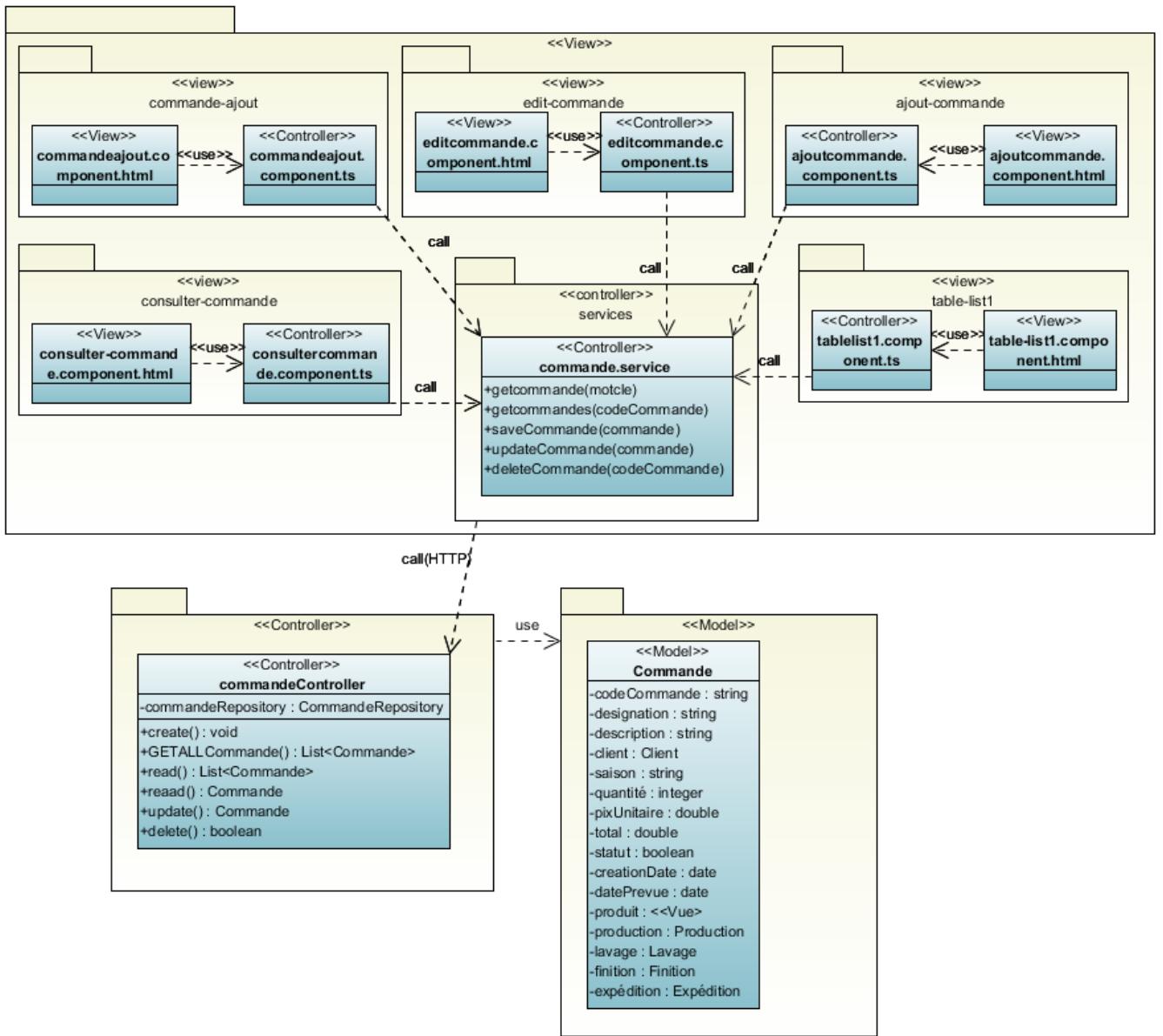


*Figure III.8 - Diagramme de classe de conception du cas d'utilisation Approuver un client*

A partir de la vue principale table-client, l'administrateur peut approuver ou rechercher un client. Cette opération est effectuée via le service du FrontEnd client.service qui fait un appel au contrôleur du Backend client via le protocole HTTP.

### III.3.5 Diagramme de classe de conception du cas d'utilisation « Gérer les bons de commande »

Le diagramme de *la figure III.9* représente le diagramme des classes détaillées du cas d'utilisation « Gérer les bons de commande».



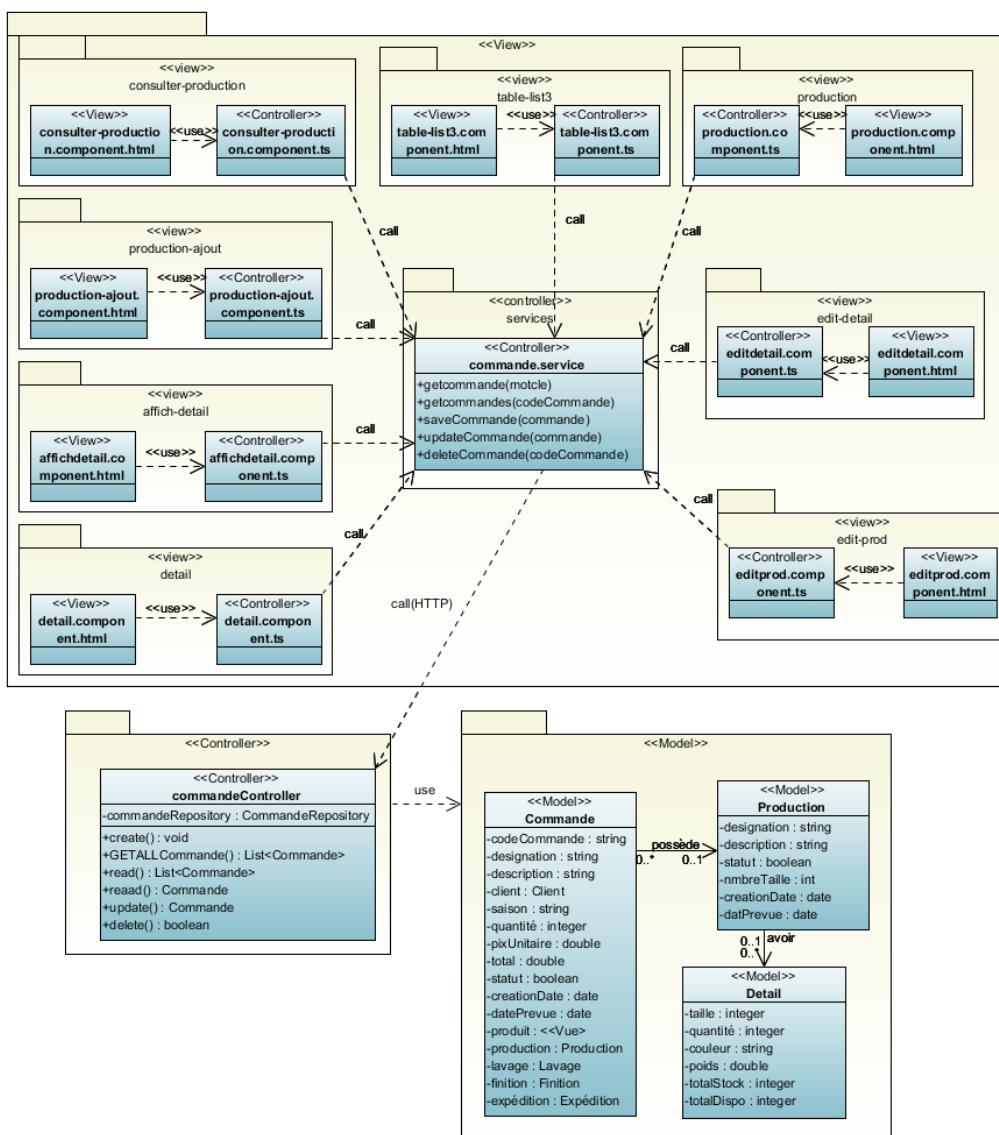
*Figure III.9 - Diagramme de classe de conception du cas d'utilisation Gérer les bons de commande*

A partir de la vue principale de gestion des commandes tablelist1 le responsable Key-Account peut supprimer ou rechercher une commande. L'ajout, la modification et l'affichage des commandes s'effectue respectivement à partir de la vue commande-ajout / ajout-commande, edit-commande et consulter-commande.

Ces opérations sont effectuées via le service du FrontEnd commande.service qui fait un appel au contrôleur du Backend commandeController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.6 Diagramme de classe de conception du cas d'utilisation « Gérer la production »

Le diagramme de *la figure III.10* représente le diagramme des classes détaillées du cas d'utilisation « Gérer la production ».



*Figure III.10 - Diagramme de classe de conception du cas d'utilisation Gérer la production*

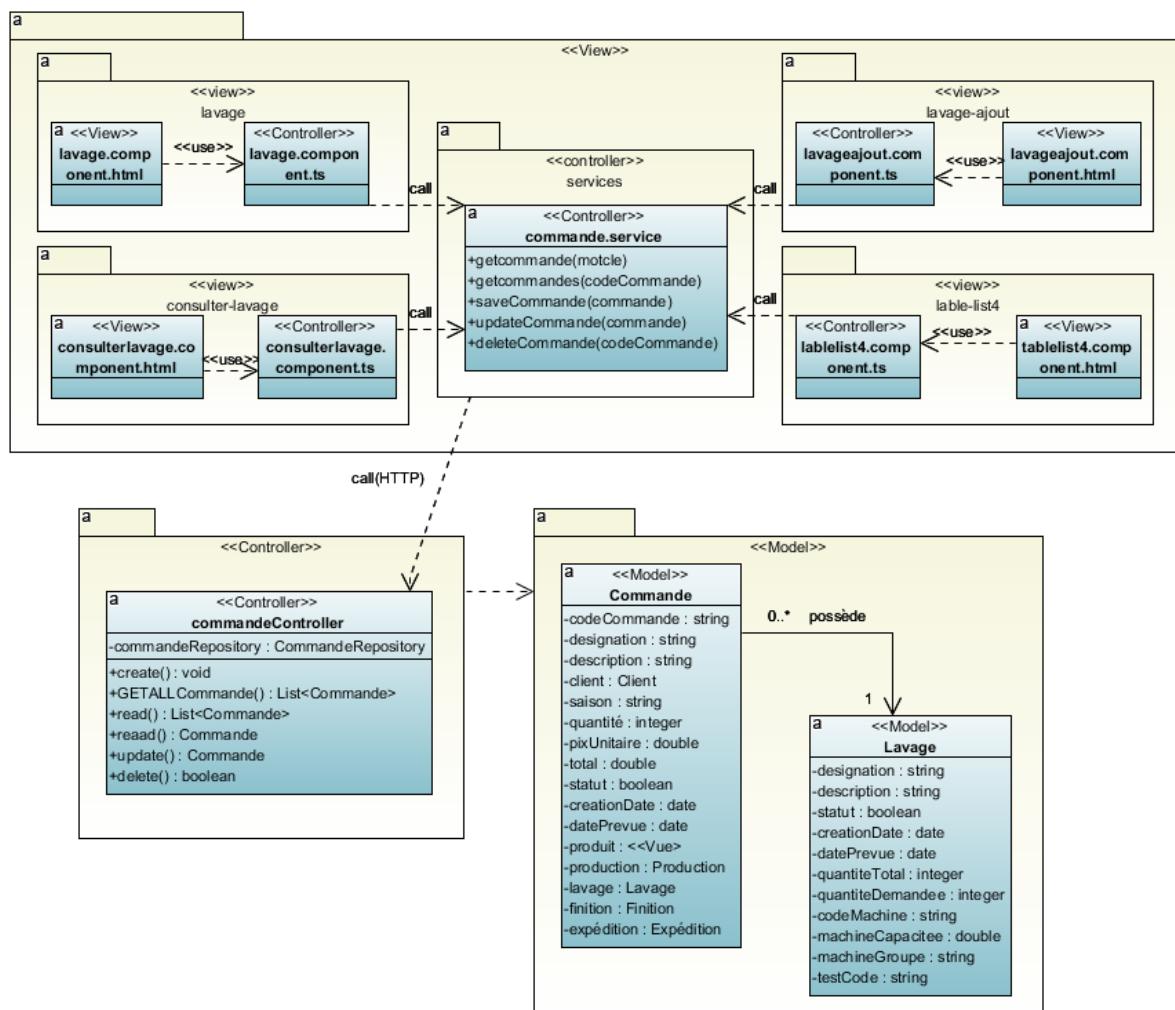
A partir de la vue principale de gestion des commandes table-list3 qui affiche la liste des productions, le responsable peut exécuter la validation et recherche d'une production. L'affichage et la modification des productions s'effectuent respectivement à partir de la vue consulter-production / affich-detail et edit-prod / edit-detail.

A partir de la vue Production qui affiche la liste des commandes le responsable de production peut rechercher ou peut ajouter une production à partir de la vue production-ajout / detail.

Ces opérations sont effectuées via le service du FrontEnd commande.service qui fait un appel au contrôleur du Backend commandeController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.7 Diagramme de classe de conception du cas d'utilisation « Gérer le lavage »

Le diagramme de *la figure III.11* représente le diagramme des classes détaillées du cas d'utilisation « Gérer le lavage ».



*Figure III.11 - Diagramme de classe de conception du cas d'utilisation Gérer le lavage*

A partir de la vue principale de gestion des lavages table-list4, le responsable peut exécuter la validation et recherche d'un lavage. L'affichage et la modification des lavages s'effectuent respectivement à partir de la vue consulter-lavage et lavage-ajout.

A partir de la vue lavage qui affiche la liste des commandes le responsable de lavage peut rechercher ou peut ajouter un lavage à partir de la vue lavage-ajout (nous pouvons faire modifier/ajouter lavage à partir de lavage-ajout).

Ces opérations sont effectuées via le service du FrontEnd commande.service qui fait un appel au contrôleur du Backend commandeController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.8 Diagramme de classe de conception du cas d'utilisation « Gérer la finition »

Le diagramme de *la figure III.12* représente le diagramme des classes détaillées du cas d'utilisation « Gérer la finition ».

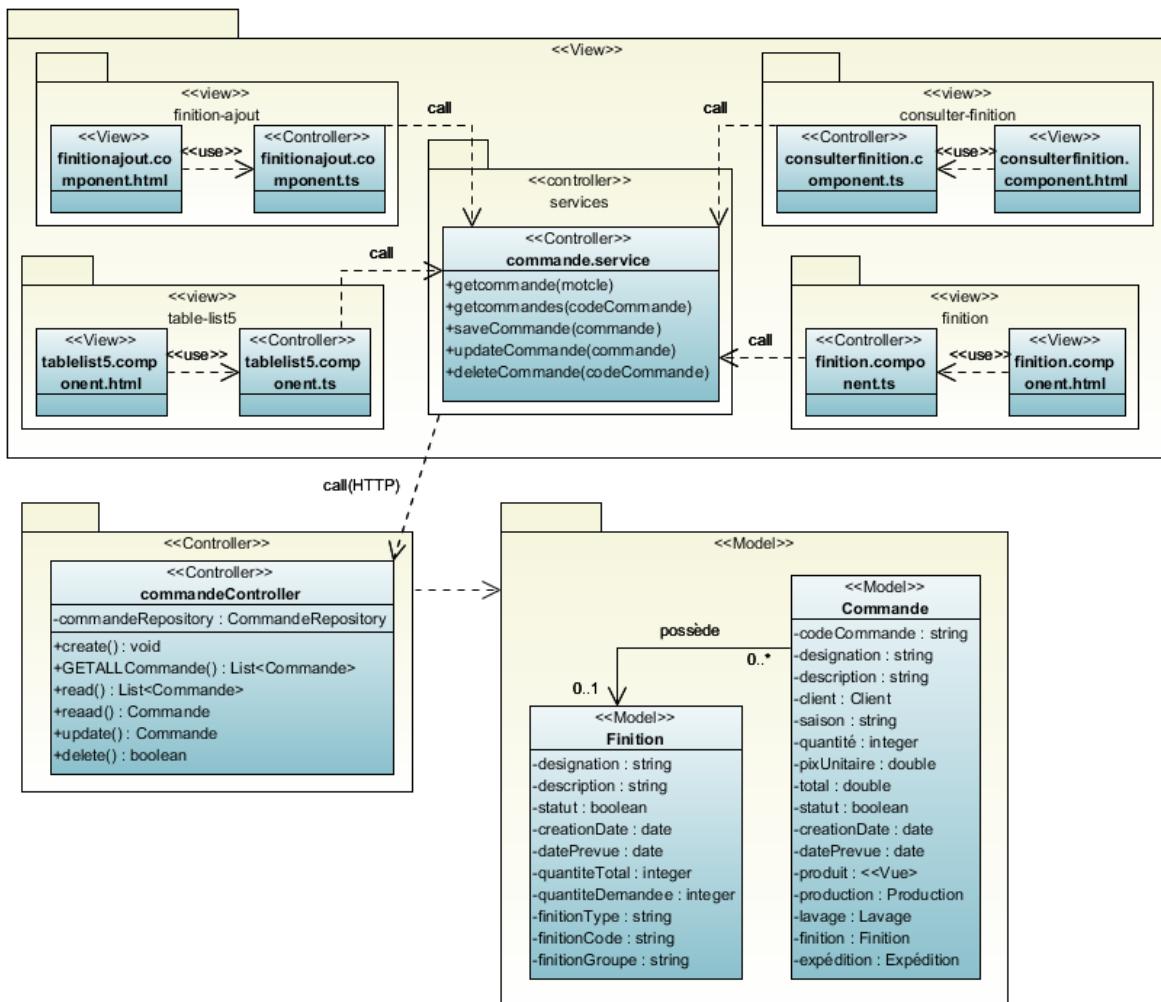


Figure III.12 - Diagramme de classe de conception du cas d'utilisation **Gérer la finition**

A partir de la vue principale de gestion des finitions table-list5, le responsable peut exécuter la validation et recherche d'une finition. L'affichage et la modification des finitions s'effectuent respectivement à partir de la vue consulter-finition et finition-ajout.

A partir de la vue finition qui affiche la liste des commandes le responsable de finition peut rechercher ou peut ajouter une finition à partir de la vue finition-ajout (nous pouvons faire modifier/ajouter finition à partir de finition-ajout).

Ces opérations sont effectuées via le service du FrontEnd commande.service qui fait un appel au contrôleur du Backend commandeController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.9 Diagramme de classe de conception du cas d'utilisation « Gérer les produits »

Le diagramme de *la figure III.13* représente le diagramme des classes détaillées du cas d'utilisation « Gérer les produits ».

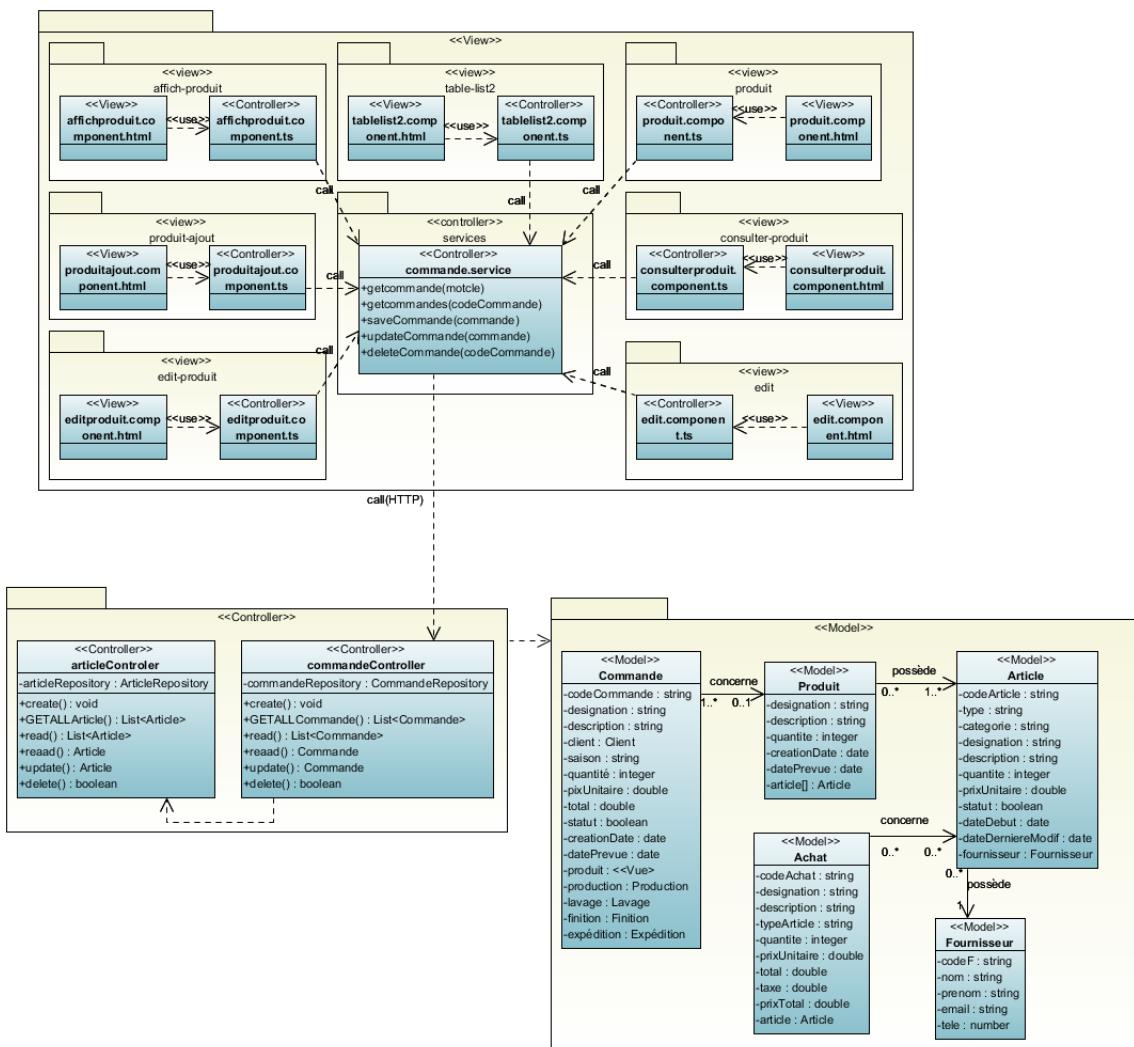


Figure III.13 - Diagramme de classe de conception du cas d'utilisation **Gérer les produits**

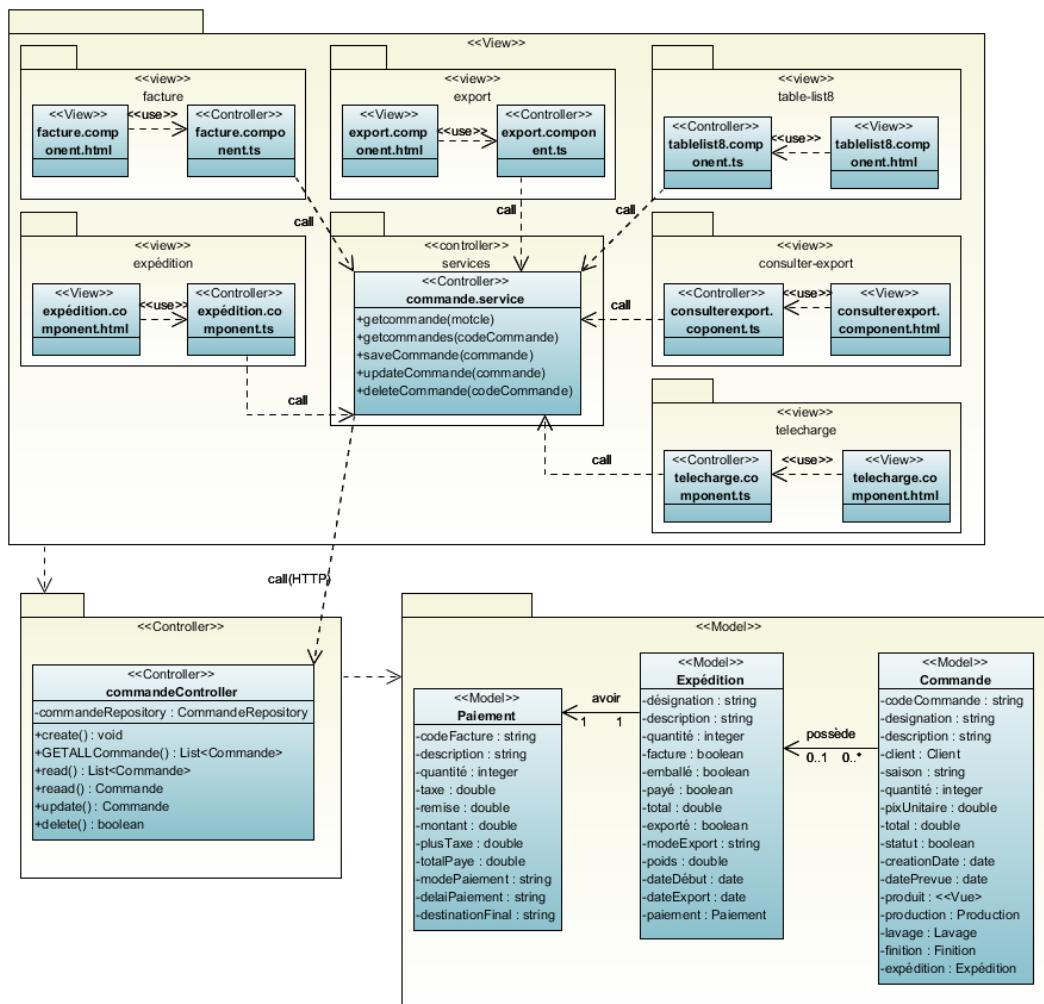
A partir de la vue principale de gestion des produits table-list2 qui affiche la liste des produits, le responsable peut exécuter la validation et recherche d'un produit. L'affichage et la modification des produits s'effectuent respectivement à partir de la vue consulter-produit / affich-produit et edit-produit / edit.

A partir de la vue Expédition qui affiche la liste des commandes le responsable d'export peut rechercher ou peut ajouter une production à partir de la vue produit-ajout.

Ces opérations sont effectuées via le service du FrontEnd commande.service qui fait un appel au contrôleur du Backend commandeController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.10 Diagramme de classe de conception du cas d'utilisation « Gérer l'expédition »

Le diagramme de *la figure III.14* représente le diagramme des classes détaillées du cas d'utilisation « Gérer l'expédition ».



*Figure III.14 - Diagramme de classe de conception du cas d'utilisation Gérer l'expédition*

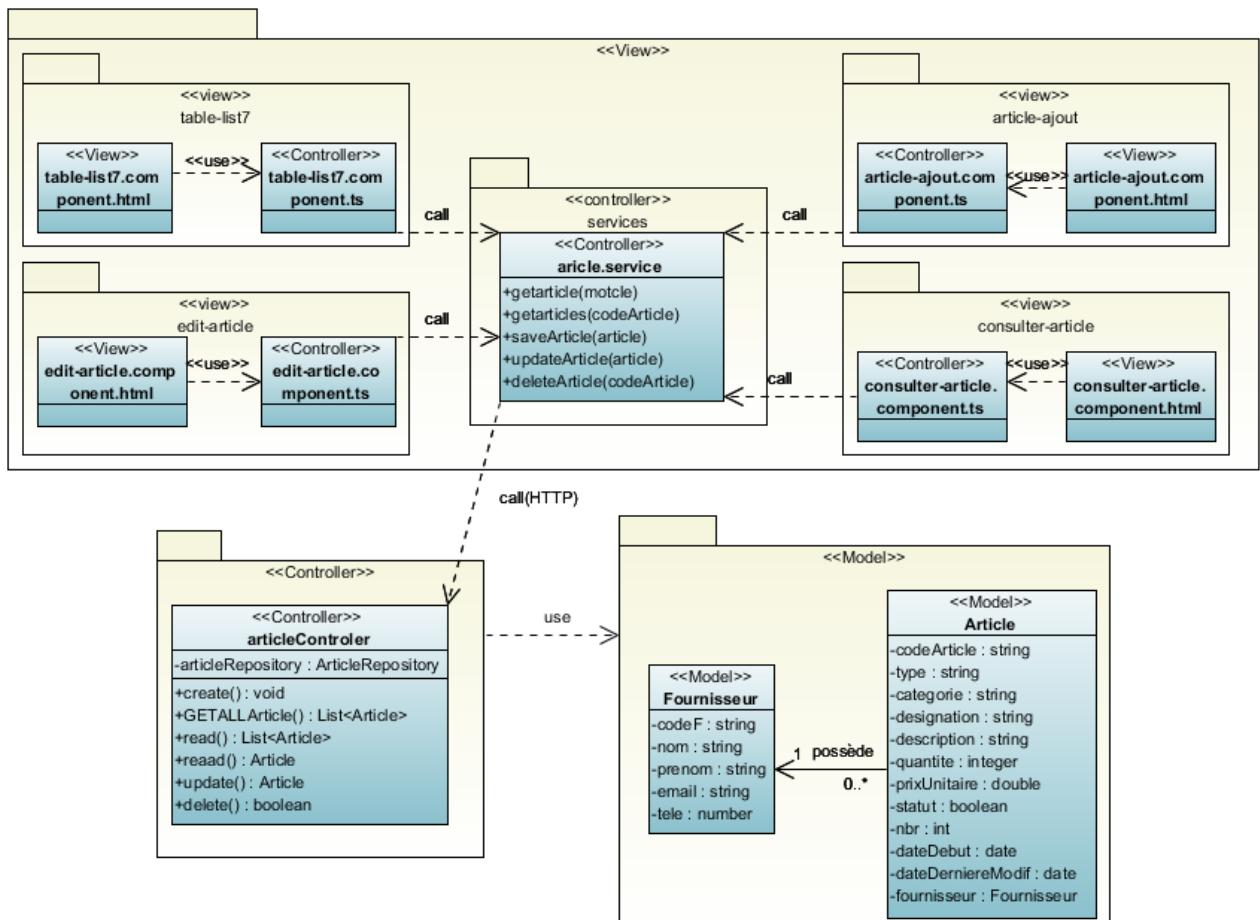
A partir de la vue principale de gestion d'expédition table-list8 qui affiche la liste des exports, le responsable peut exécuter la recherche d'un export. L'affichage et la modification des exports s'effectuent respectivement à partir de la vue consulter-export / telecharge et facture / export. Pour ajouter un export le responsable doit remplir le formulaire de la facture après celui de l'export.

A partir de la vue Expédition qui affiche la liste des commandes le responsable d'expédition peut rechercher ou peut ajouter un export à partir de la vue facture / export. (Nous pouvons faire modifier/ajouter export à partir des vues facture/export).

Ces opérations sont effectuées via le service du FrontEnd commande.service qui fait un appel au contrôleur du Backend commandeController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.11 Diagramme de classe de conception du cas d'utilisation « Gérer les articles »

Le diagramme de *la figure III.15* représente le diagramme des classes détaillées du cas d'utilisation « Gérer les articles ».



*Figure III.15 - Diagramme de classe de conception du cas d'utilisation Gérer les articles*

A partir de la vue principale de gestion des articles table-list7, le responsable peut supprimer ou rechercher un article. L'ajout, la modification et l'affichage des articles s'effectuent respectivement à partir de la vue article-ajout, edit-article et consulter-article.

Ces opérations sont effectuées via le service du FrontEnd article.service qui fait un appel au contrôleur du Backend artilceController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### III.3.12 Diagramme de classe de conception du cas d'utilisation « Gérer les achats »

Le diagramme de *la figure III.16* représente le diagramme des classes détaillées du cas d'utilisation « Gérer les achats ».

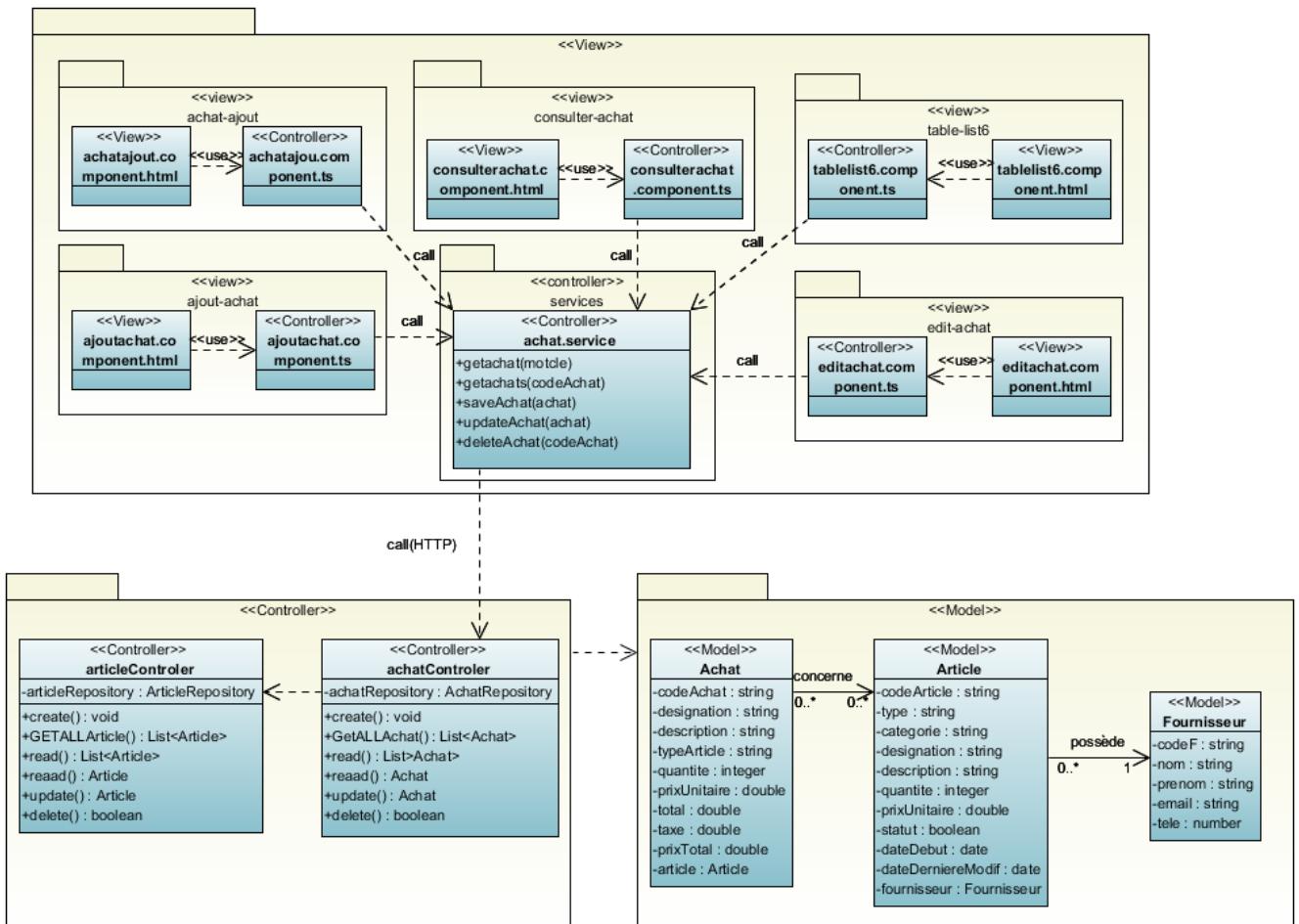


Figure III.16 - Diagramme de classe de conception du cas d'utilisation **Gérer les achats**

A partir de la vue principale de gestion des articles table-list6, le responsable peut supprimer ou rechercher un achat. L'ajout, la modification et l'affichage des achats s'effectuent respectivement à partir de la vue achat-ajout/ajout-achat, edit-achat et consulter-achat.

Ces opérations sont effectuées via le service du FrontEnd achat.service qui fait un appel au contrôleur du Backend achatController et articleController via le protocole HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.

### **III.3.13      Diagramme de séquence objet**

Un diagramme de séquence objet décrit les interactions en montrant, de façon séquentielle, les envois de message qui interviennent entre les objets. Une interaction est une séquence de messages entre des instances habituelles de classes, de composants, de sous-systèmes ou encore d'acteurs. On présentera ci-après les diagrammes de séquences, que nous avons jugées importants pour la réalisation de notre application.

#### **III.3.13.1    Diagramme de séquence objet du cas d'utilisation « Ajouter un utilisateur»**

Le diagramme de *la figure III.17* représente le diagramme de séquence objet du cas d'utilisation « Ajouter un utilisateur».

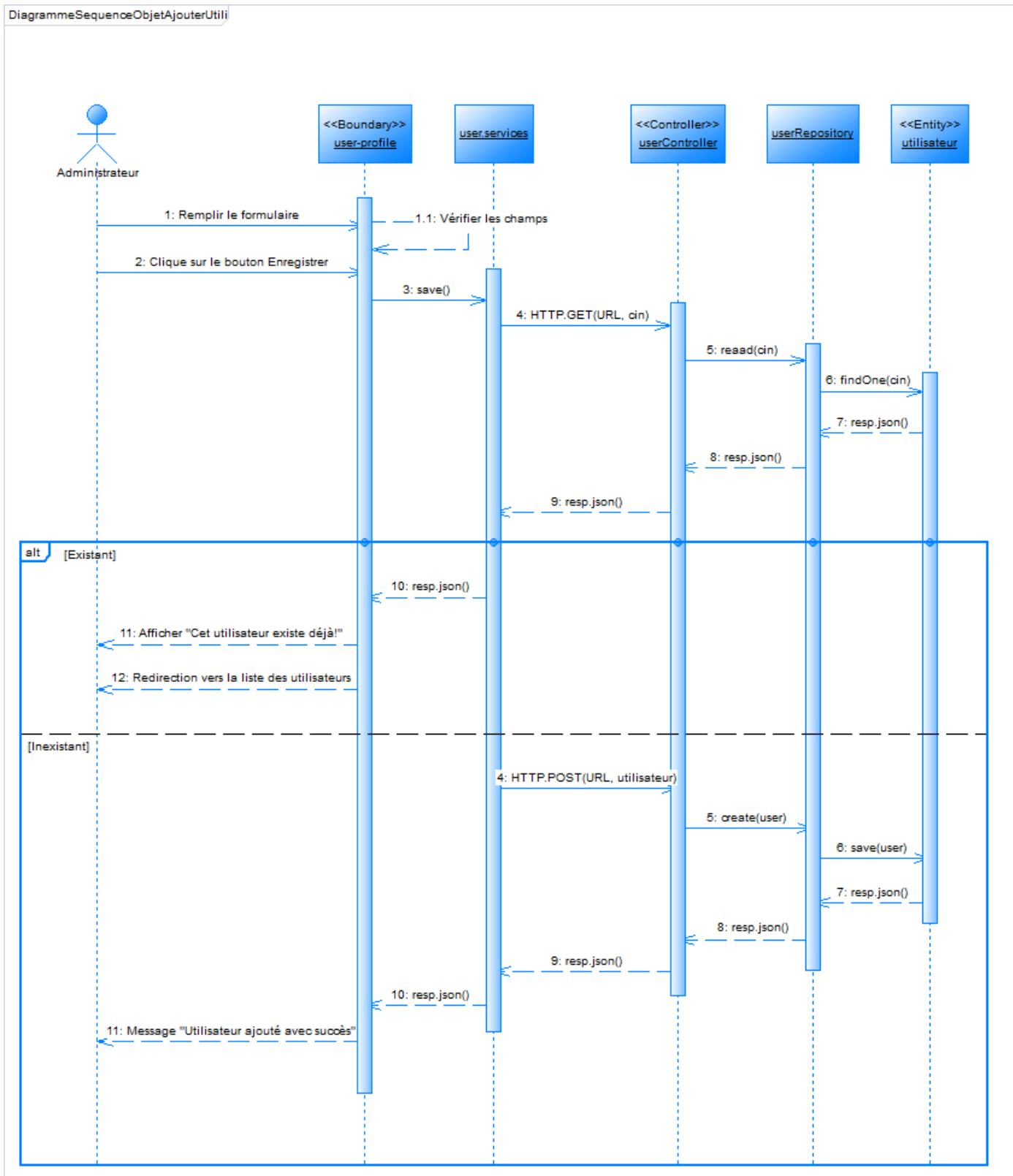


Figure III.17 - Diagramme de séquence objet du cas d'utilisation **Ajouter un utilisateur**

### III.3.13.2 Diagramme de séquence objet du cas d'utilisation «Modifier un utilisateur»

Le diagramme de *la figure III.18* représente le diagramme de séquence objet du cas d'utilisation «Modifier un utilisateur».

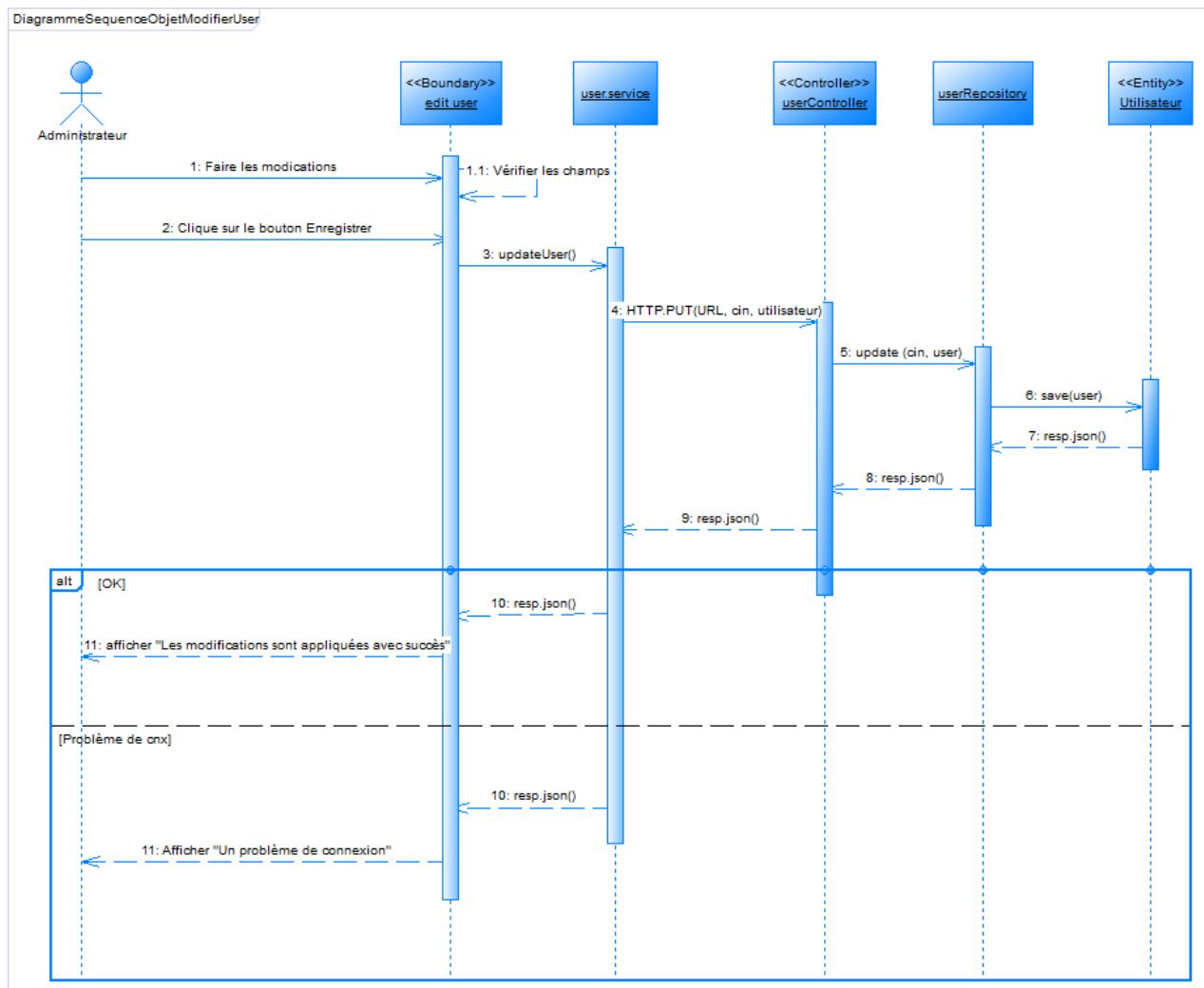


Figure III.18 - Diagramme de séquence objet du cas d'utilisation **Modifier un utilisateur**

Le système affiche la liste des utilisateurs existants et permet au responsable de consulter les détails de chacun. Le responsable peut modifier les coordonnées d'un utilisateur via un formulaire de modification affiché par le système, qui vérifie les modifications puis si elles sont valides il les enregistre sinon il demande au responsable de vérifier ses données saisies. Quand le responsable demande au système d'ajouter un responsable, le système affiche un formulaire d'ajout. Le responsable remplit les coordonnées du nouvel utilisateur qui seront vérifiées par le système : si l'utilisateur existe déjà, un message d'erreur sera affiché « Cet utilisateur existe déjà » sinon, s'il n'existe pas et si les données sont valides le système crée un nouvel utilisateur, sinon il rejette ce qui a été introduit et demande de vérifier les données. Aussi le responsable a la faculté de supprimer un client. Si cette demande est confirmée, le système supprime l'utilisateur.

### **III.3.14      Diagramme de composant**

Les diagrammes de composants permettent de décrire l'architecture physique et statique d'une application en termes de modules. Les dépendances entre composants permettent notamment d'identifier les contraintes de compilation et de mettre en évidence la réutilisation de composants. Les composants peuvent être organisés en paquetages, qui définissent des sous-systèmes. Les sous-systèmes organisent la vue des composants (de réalisation) d'un système [5].

#### **III.3.14.1    Diagramme de composants de la Gestion des utilisateurs**

Ce diagramme décrit l'architecture d'une application en termes des éléments logiciels (composants) réutilisables, en identifiant les dépendances entre eux. Le diagramme de composants montre la mise en œuvre des composants de la vue logique avec l'environnement de développement.

Comme il est indiqué dans l'architecture logique, notre application est composée d'une partie Front et une partie Back, nous allons présenter les diagrammes de composants de la partie FrontEnd de la Gestion des utilisateurs dans *la figure III.19*.

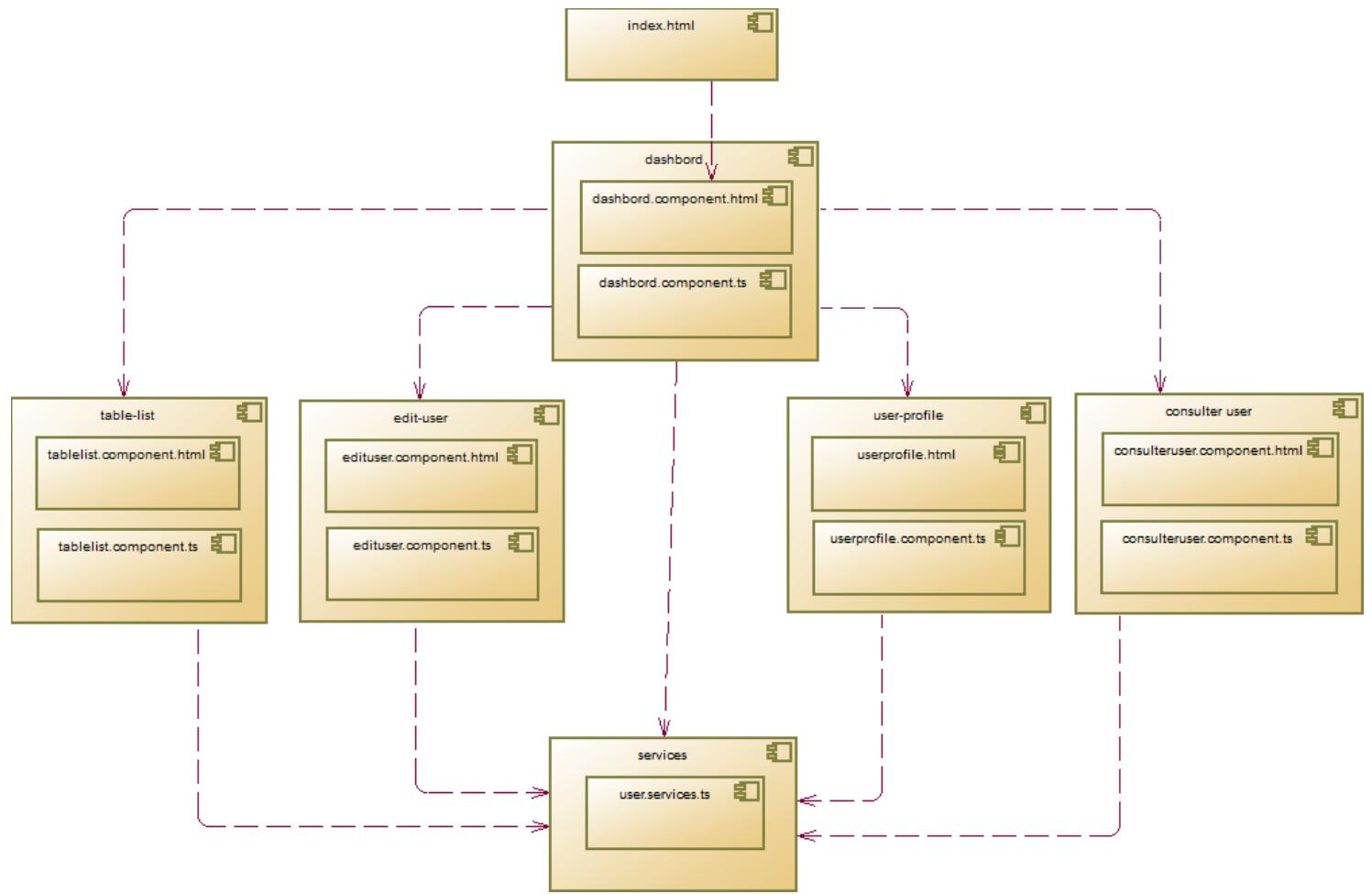


Figure III.19- Diagramme de composant de la partie FrontEnd de la **Gestion des utilisateurs**

## Conclusion

Tout au long de ce chapitre, nous avons essayé d'élaborer un modèle de conception afin de donner une image «prête à coder» de notre solution et d'aboutir à un système fonctionnel reflétant une réalité physique.

## Chapitre IV. Réalisation

---

Cette partie représente le dernier volet de ce rapport. Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Après avoir terminé la phase de spécification et conception, la solution étant déjà choisie et étudiée, il ne nous reste que de donner un bref aperçu des technologies et langages de programmation utilisés et présenter l'état de réalisation.

### IV.1 Environnement de travail

#### IV.1.1 Environnement matériel

Pour la réalisation de ce projet, nous avons utilisé deux ordinateurs portables :

*Les figures IV.1 et VI.2 illustrent les caractéristiques des deux machines utilisés lors du développement de ce Projet.*

| Système                     |   |
|-----------------------------|---|
| Fabricant:                  | Hewlett-Packard   |
| Évaluation:                 |  Indice de performance Windows |
| Processeur:                 | Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz 1.70 GHz   |
| Mémoire installée (RAM):    | 4,00 Go   |
| Type du système:            | Système d'exploitation 64 bits  |
| Stylet et fonction tactile: | La fonctionnalité de saisie tactile ou avec un stylet n'est pas disponible sur cet écran                          |

*Figure IV.1 - Les caractéristiques du PC HP*

| Système  |  |
|--|--|
| Processeur:  | Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz 2.59 GHz  |
| Mémoire installée (RAM):   | 8,00 Go (7,86 Go utilisable)   |
| Type du système:   | Système d'exploitation 64 bits, processeur x64   |
| Stylet et fonction tactile:  | La fonctionnalité d'entrée tactile ou avec un stylet n'est pas disponible sur cet écran. |
| Paramètres de nom d'ordinateur, de domaine et de groupe de travail |  |
| Nom de l'ordinateur:   | fsb-hb   |
| Nom complet:   | fsb-hb   |
| Description de l'ordinateur:                                       |  |
| Groupe de travail:   | DEP-INFO   |

*Figure IV.2 Les caractéristiques du PC LENOVO*

## IV.1.2 Environnement logiciel

### IV.1.2.1 Eclipse Oxygen

Nous avons utilisé pour le développement de la partie Backend de notre application une nouvelle version de l'IDE Eclipse, la version Eclipse Oxygen qui a apporté des nouveautés et améliorations intéressantes. Nous avons ajouté le plugin SpringTools, afin d'utiliser le Framework SPRING.

### IV.1.2.2 WebStorm

Nous avons utilisé pour le développement de la partie FrontEnd de notre application, l'IDE WebStorm qui est développé par l'entreprise JetBrains et basé sur la plateforme IntelliJ IDEA. Il améliore la productivité et offre une expérience de développement agréable grâce un éditeur intelligent avec l'autocomplétion, la détection d'erreur à la volée, les refactorisations et le formatage de code, etc. Il offre un support avancé pour Node.js et les Framework populaires comme Angular que nous avons également utilisé.

### IV.1.2.3 MongoDB



Figure IV.3 - Logo du MongoDB

Nous avons utilisé MongoDB comme un système de gestion de base de données qui n'est pas comme les autres systèmes car il s'agit d'une base de données NoSQL.

MongoDB est un système de gestion de base de données orientée documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédefini des données. En tant que base de données NoSQL, MongoDB évite la structure basée sur la table de la base de données relationnelle pour adapter les documents de type JSON qui ont des schémas dynamiques appelés BSON. Cela rend l'intégration de données pour certains types d'applications plus rapide et plus facile. MongoDB est conçu pour l'évolutivité, la haute disponibilité et les performances d'un déploiement de serveur unique vers des infrastructures multi-sites complexes et de grande taille. [6]

## Modèle de la base de données

Le modèle de *la figure IV.4* représente le modèle de la base de données de notre application. Nous avons 5 collections : Commande, User, Client, Article et Achat.

Les collections User et Client possèdent l'enregistrement Adresse. La collection Article possède l'enregistrement Fournisseur. Dans tous les enregistrements Article, nous avons l'enregistrement Fournisseur. La collection Commande possède les enregistrements suivants : Client, Production, Lavage, Finition, Expédition et Produit. A l'intérieur de l'enregistrement Produit, nous avons l'enregistrement Article. A l'intérieur de l'enregistrement Expédition, nous avons l'enregistrement Paiement et à l'intérieur de l'enregistrement Production, nous avons l'enregistrement Détail. La collection Achat possède l'enregistrement Article.

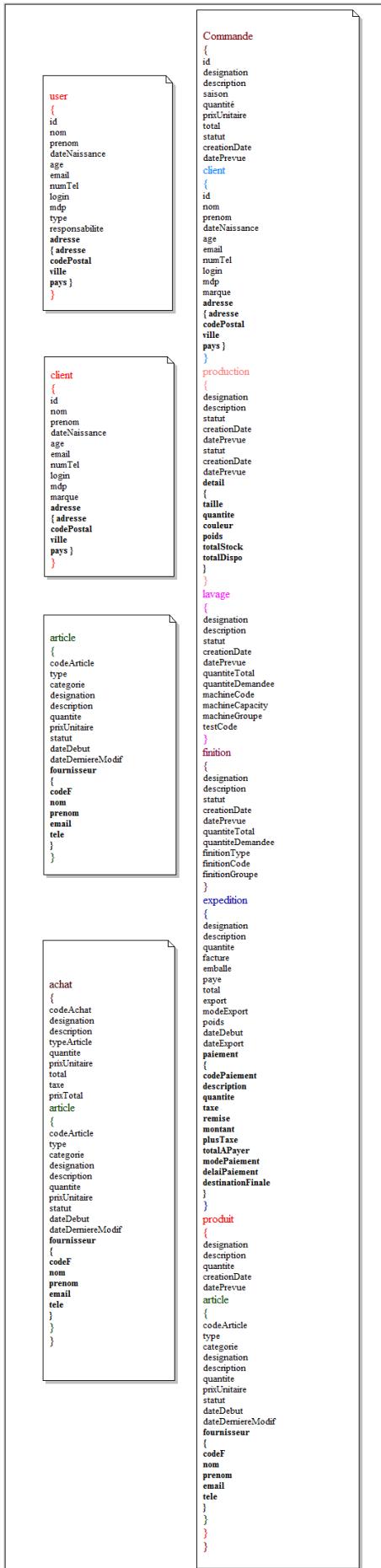


Figure IV.4 – Le modèle de la base de données

#### IV.1.2.4 NodeJS

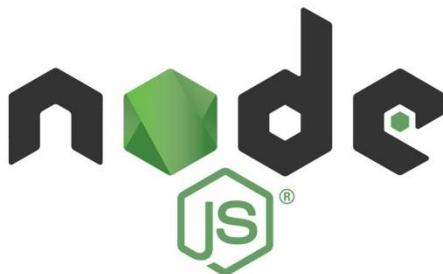


Figure IV.5 - Logo du NodeJS

Node.js est un environnement JavaScript basé sur la machine virtuelle V8 et implémente sous licence MIT les spécifications CommonJS [7].

Node.js nous permet d'exécuter du code JavaScript dans un navigateur web et notamment pour la création de serveur web puisqu'il utilise un modèle d'entrée/sortie non bloquant ce qui lui permet de gérer plusieurs tâches de façon asynchrone. Le package Node.js inclut également l'outil NPM (Node Package Manager) qui nous a permis de télécharger et d'accéder à de très nombreuses librairies pour tourner du JavaScript sur notre machine.

#### IV.1.2.5 Apache Tomcat

Apache Tomcat est une implémentation open source d'un conteneur web qui permet donc d'exécuter des applications web reposant sur les technologies servlets et JSP.

Il est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

#### IV.1.2.6 PowerAMC

PowerAMC est un logiciel de modélisation. Il nous a permis de réaliser les diagrammes de séquence système, les diagrammes de séquence objet et le diagramme de classe d'analyse. Nous avons réalisé aussi avec PowerAMC le diagramme de composant de la « Gestion des utilisateurs ».

#### IV.1.2.7 Visual Paradigm

Visual Paradigm est un logiciel de création de diagrammes dans le cadre d'une programmation. Tout en un, il possède plusieurs options qui nous a permis de modéliser en UML. Nous avons réalisé avec cet outil, les diagrammes des cas d'utilisation de notre système et le diagramme de package ainsi que les diagrammes de classe de conception des différents cas d'utilisation.

#### IV.1.2.8 Balsamiq Mockups

Balsamiq est l'éditeur du produit Balsamiq Mockups, un outil permettant de créer facilement des prototypes d'IHM électronique. Avec Balsamiq Mockups, nous avons réalisé les différentes maquettes du chapitre II.

## IV.2 Choix technique

### IV.2.1 Framework de développement

#### IV.2.1.1 Angular 4



Figure IV.6 - Logo de l'Angular 4

Angular est un Framework **orienté composant**. Angular est une plate-forme d'applications Web open-source basée sur TypeScript et dirigée par l'équipe Angular de Google et par une communauté de particuliers et d'entreprises. Angular est une réécriture complète de la même équipe qui a construit AngularJS [8].

**Anguar4 est orienté composants, un composant est une entité réutilisable, il peut être :**

- **Vue + Contrôleur** : où la vue est un fichier HTML et un fichier CSS, et le contrôleur est une classe TypeScript qui offre les traitements à faire, c'est le lien entre la vue et les services.

- **Service** : c'est un composant qui gère les requêtes http, afin d'assurer l'échange des données JSON entre les services Web et le contrôleur.

- **Pipe** : c'est un composant qui peut être utilisé dans les balises html pour personnaliser l'affichage des données, il prend en entrée des données, les transforme et les retourne en sortie dans un format désiré.

Nous avons choisi cette nouvelle technologie comme Framework FrontEnd car il nous fournit une application très rapide par rapport aux autres Framework JavaScript et plus sécurisé parce qu'on ne peut pas injecter la balise <script> n'importe où. De plus, il apporte une façon de coder plus claire grâce à TypeScript, en utilisant les classes, les constantes, les pipes, etc. Ce qui améliore la maintenabilité du code. Angular nous fournit une application modulaire, c'est-à-dire une application divisée en components. Chaque component est affecté à une partie de l'application spécifique, et doit pouvoir être réutilisable dans une autre application.

#### IV.2.1.2 Bootstrap



Figure IV.7 - Logo du Bootstrap

Bootstrap est un Framework frontal gratuit pour un développement Web. Il apporte du style pour les éléments HTML de façon simplifiée et rapide, car le code de style est déjà prêt, nous allons juste faire appel à lui et l'utiliser.

#### IV.2.1.3 Spring Boot



Figure IV.8 - Logo du Spring Boot

Nous avons développés nos services Web REST avec ce Framework. En effet, Spring est un Framework libre pour construire et définir l'infrastructure d'une application java, dont il facilite le développement et les tests. Spring Boot est un projet qui se trouve sur la couche d'IO Execution la couche de Spring IO Framework [9].

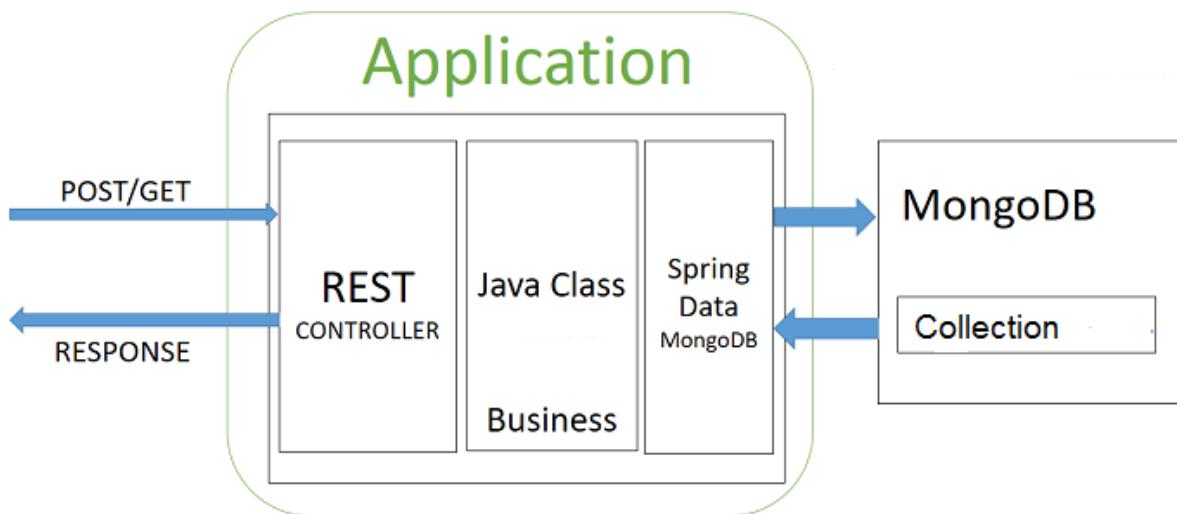


Figure IV.9 - Architecture du Spring Boot

Nous avons choisi cette nouvelle technologie comme Framework Backend car il diminue énormément du temps et augmente la productivité et suit l'approche “Configuration par défaut” afin de diminuer le temps et l'effort de développement. De plus il nous fournit des serveurs intégrés (Embedded HTTP servers) comme Tomcat, Jetty... afin de développer et de tester des applications web à la manière la plus facilement. Il fournit aussi beaucoup de plugins afin de développer et de tester des applications Spring Boot rapidement en utilisant les outils de Build comme Maven et Gradle.

**Pour développer avec le Framework Spring, il faut savoir d'abord certaines notations tels que :**

➤ **JSON**

JavaScript Object Notation (JSON) est un standard ouvert, humain et lisible par une machine qui facilite l'échange de données. JSON prend en charge tous les types de données de base: les nombres, les chaînes et les valeurs booléennes, ainsi que les tableaux et les hachages [10].

Les bases de documents telles que MongoDB que nous avons utilisé, utilisent des documents JSON pour stocker des enregistrements, tout comme les tables et les lignes stockent des enregistrements dans une base de données relationnelle. En effet, MongoDB représente les documents JSON au format binaire appelé BSON dans les coulisses. BSON étend le modèle JSON pour fournir des types de données supplémentaires, des champs ordonnés et pour être efficace pour l'encodage et le décodage dans différentes langues.

➤ **Service RESTFUL**

REST (REpresentational State Transfer) n'est pas un protocole mais un style architectural. Il décrit comment un système (serveur) peut communiquer avec un client en utilisant des méthodes HTTP pour échanger des données au format JSON, XML, texte ou tout autre format. Il n'y a pas de règles strictes décrivant comment construire et comment le contrôleur REST devrait ressembler, mais il y a quelques caractéristiques et directives que nous devons garder à l'esprit :

- Le client envoie une requête HTTP. Notre API REST (serveur) interroge la base de données pour savoir ce dont elle a besoin et renvoie une réponse HTTP en tant que donnée.
- La demande et la réponse doivent être dans un format que le client et le serveur peuvent comprendre.
- Les données sont presque toujours échangées via HTTP en utilisant différentes méthodes. Ces méthodes sont POST, GET, PUT et DELETE et elles correspondent à créer, lire, mettre à jour et supprimer (ou CRUD) les opérations de base de données.

➤ **Spring Data**

- ✓ Spring Data offre une API qui fait abstraction de ces API « bas niveau » (tout en prenant en compte les spécificités de chacune d'elles). Par exemple, pour bénéficier d'opérations CRUD de base, il suffit tout simplement d'étendre une interface de Spring Data pour y avoir accès.

- ✓ Spring Data s'interface avec plusieurs sources de données parmi lesquelles JPA, Neo4j, MongoDB, GemFire, Hadoop et quelques autres.
- ✓ Spring Data est découpé en une couche commune à toutes les sources de données sous-jacentes (Neo4j, MongoDB, JPA...), appelée Spring Data Commons, à laquelle s'ajoute une couche propre à la source de données. Donc si vous utilisez Spring Data MongoDB, vous aurez Spring Data Commons et Spring Data MongoDB.
- ✓ Spring Data Commons contient les classes pour fonctionner, mais surtout les interfaces que l'utilisateur aura à étendre (directement ou pas) : Repository, CrudRepository et PagingAndSortingRepository.

#### **IV.2.1.3.1 Etude de cas : Création du Backend Spring Boot (Exemple : Gestion des commandes)**

##### **1. Configuration de la base de données MongoDB**

Comme nous avons ajouté la dépendance Spring-Boot-Starter-MongoDB, Spring Boot tente de créer une connexion avec MongoDB en lisant la configuration de la base de données à partir du fichier application.properties.

##### **2. Crédation du modèle Commande**

Nous commençons par créer un modèle Commande qui sera mappé à un document dans la base de données « manufacturing » MongoDB.

##### **3. Crédation d'commandeRepository pour accéder à la base de données**

Ensuite, nous devons créer commandeRepository pour accéder aux données de la base de données.

Nous avons étendu commandeRepository avec l'interface MongoRepository fournie par Spring-Data-MongoDB. L'interface MongoRepository définit des méthodes pour toutes les opérations CRUD sur le document comme findAll (), fineOne (), save (), delete () etc.

Spring Boot branche automatiquement une implémentation de l'interface MongoRepository appelée SimpleMongoRepository lors de l'exécution. Ainsi, toutes les méthodes CRUD définies par MongoRepository sont facilement accessibles.

##### **4. Crédation des API - commandeController**

Enfin, créons les API qui seront exposées aux clients. Créez un nouveau paquet de contrôleurs et ajoutez un fichier commandeController.java.

#### **IV.2.2 Langage de programmation**

##### **IV.2.2.1 JAVA Entreprise Edition**



*Figure IV.10 - Logo Java Entreprise Edition*

Pour le développement de notre application, nous avons utilisé Java EE qui est aujourd’hui la meilleure plate-forme de développement pour les entreprises. Java Enterprise Edition, est un ensemble de spécifications destinées aux applications d’entreprise. Java EE peut être vu comme une extension du langage Java afin de faciliter la création d’applications réparties, robustes, performantes et à haute disponibilité. [11]

#### **IV.2.2.2 HTML : HypertextMarkup Langage**

L'HTML un langage de balisage, utilisé pour concevoir les sites Internet. Il offre la possibilité de définir le contenu principal d'une page Web, d'ajouter une introduction en header, d'insérer un sous-titre à un contenu multimédia de type vidéo, etc [12].

#### **IV.2.2.3 CSS : Cascading Style Sheet**

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML. [13]

#### **IV.2.2.4 JavaScript**

Le JavaScript est un langage de script incorporé dans un document HTML. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web. Ainsi le langage JavaScript est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu [14].

#### **IV.2.2.5 TypeScript**

TypeScript est un langage de script structuré et orienté objet qui permet de simplifier le développement d'applications JavaScript.

TypeScript fait partie de la ribambelle de langages alternatifs compilant en JavaScript. Ses caractéristiques principales sont le typage statique à la compilation et le fait qu'il soit un super-ensemble de JavaScript [15].

#### IV.2.2.6 UML : Langage de modélisation unifié

UML (Unified Modeling Language, que l'on peut traduire par "langage de modélisation unifié) est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en termes de modélisation objet [16].

### IV.3 Les interfaces homme/machine

Les interfaces hommes/machine de l'application sont l'ensemble des moyens utilisés par l'homme pour communiquer avec une machine. Elles permettent de faciliter le dialogue entre l'homme et la machine ainsi que d'améliorer les performances de l'application .Dans la conception des interfaces de notre application nous avons respecté un ensemble des choix ergonomiques comme la lisibilité, la compréhensibilité, etc.

#### IV.3.1 Interface de l'identification

L'interface présentée par *la figure IV.11* permet à l'utilisateur d'accéder à son espace personnel en s'identifiant avec la saisie de son CIN et son mot de passe. Cette étape met en valeur l'aspect sécurité : nous vérifions la disponibilité du compte utilisateur et nous lui attribuons les droits et les priviléges nécessaires.

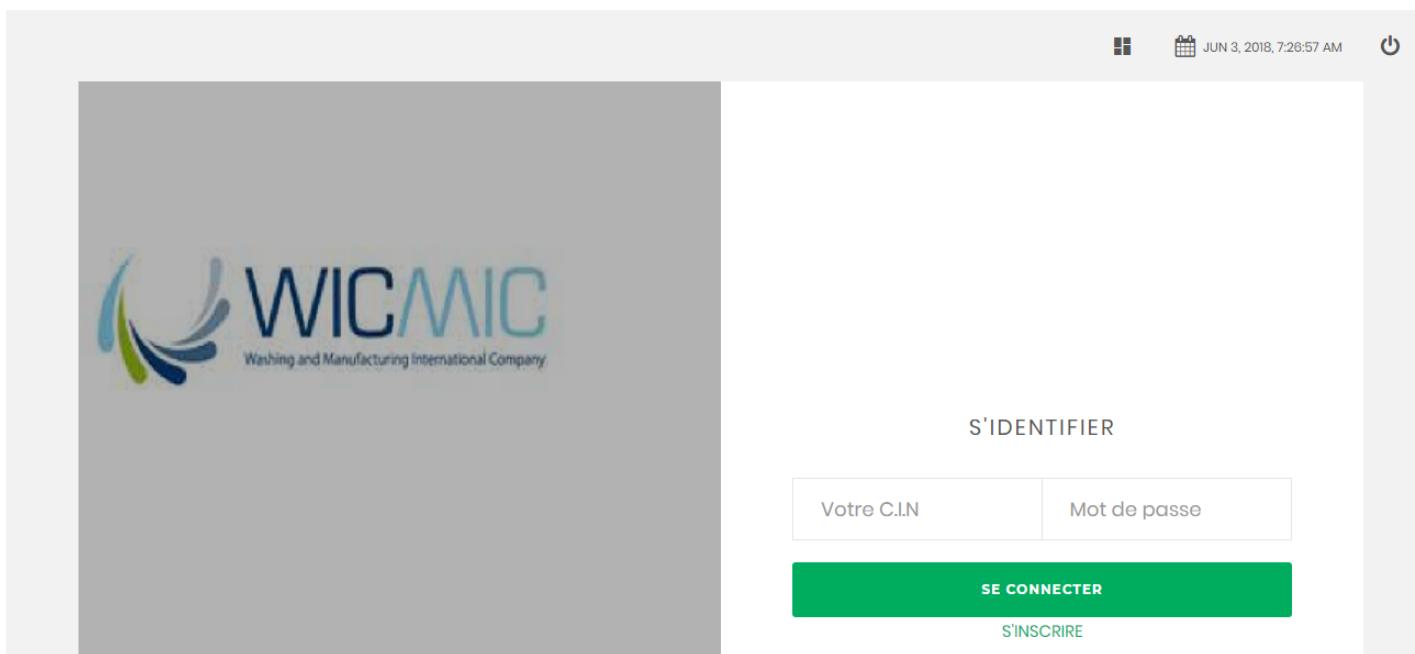


Figure IV.11 - Interface d'identification

### IV.3.2 Interface de gestion des utilisateurs

L'interface présentée par *la figure IV.12* permet à l'administrateur de supprimer et rechercher un utilisateur et on peut aussi accéder aux interfaces d'ajout, de modification et de consultation en cliquant sur l'icône qui convient.

The screenshot shows a web-based application interface for user management. At the top left is the WICMIC logo. On the right are icons for refresh, date (JUN 1, 2018, 8:01:36 PM), and power. Below the logo is a search bar with 'Code d'utilisateur' placeholder and a magnifying glass icon. A sidebar on the left has 'Gestion Utilisateur ▾' and 'Autre ▾'. The main area has a teal header 'Liste des utilisateurs'. A table lists four users:

| C.I.N    | Nom    | Prenom   | Type               | Responsabilité           | Action |
|----------|--------|----------|--------------------|--------------------------|--------|
| 98412356 | beyrem | hhhhhhhg | Administrateur     | Responsable Lavage       |        |
| 11395269 | beyrem | nour     | Administrateur     | Responsable Nomenclature |        |
| 62147896 | nour   | beyrem   | Administrateur     | Responsable Finition     |        |
| 45217888 | zzz    | zzee     | Simple utilisateur | Responsable Lavage       |        |

Figure IV.12 - Interface de gestion des utilisateurs

### IV.3.3 Interface de gestion des commandes

L'interface présentée par *la figure IV.13* permet au responsable Key-Account de supprimer, rechercher et valider un bon de commande et on peut accéder aux interfaces d'ajout, de modification et de consultation en cliquant sur l'icône qui convient.

On peut aussi accéder aux interfaces « Liste des produits » et « Ajouter un produit » à partir de cette interface en cliquant sur l'icône qui convient.

The screenshot shows the 'List des commandes' (List of Orders) section of the WICMIC application. At the top, there is a search bar labeled 'code du commande' with a magnifying glass icon. To the right of the search bar are three small icons: a grid, a calendar, and a power button. On the left side, there is a sidebar with four items: 'Liste Des Commande', 'Ajouter Une Nouvelle Commande', 'Liste Des Produit', and 'Ajouter Un Produit'. The main content area has a blue header 'List des commandes'. Below it is a table with columns: 'Code Commande', 'Designation', 'Description', 'Statut', and 'Action'. There are three rows of data:

| Code Commande | Designation | Description | Statut      | Action |
|---------------|-------------|-------------|-------------|--------|
| e01           | dd          | bb          | production  |        |
| aze           | erf         | hh          | production  |        |
| e021          | e78         | e45         | Préparation |        |

Figure IV.13 - Interface de gestion des commandes

#### IV.3.4 Interface de gestion des exports

L’interface présentée par la figure IV.14 permet au responsable d’export de rechercher et valider un export et on peut accéder aux interfaces de modification et de consultation en cliquant sur l’icône qui convient.

The screenshot shows the 'List des Exports' (List of Exports) section of the WICMIC application. At the top, there is a search bar labeled 'Code du commande' with a magnifying glass icon. To the right of the search bar are three small icons: a grid, a calendar, and a power button. On the left side, there is a sidebar with two items: 'Liste Des Exports' and 'Ajouter Une Facture'. The main content area has a blue header 'List des Exports'. Below it is a table with columns: 'Code Commande', 'Facturé', 'payée', 'emballé', 'exporté', 'Date d'export', and 'Action'. There are three rows of data:

| Code Commande | Facturé | payée | emballé | exporté | Date d'export            | Action |
|---------------|---------|-------|---------|---------|--------------------------|--------|
| e01           | true    | true  | true    | true    | May 26, 2018, 2:00:00 AM |        |
| aze           | true    | true  | true    | true    | May 13, 2018, 2:00:00 AM |        |
| e021          | false   | false | false   | false   |                          |        |

Figure IV.14 - Interface de gestion des exports

On peut aussi accéder à l'interface d'ajout d'une facture en cliquant sur l'icône qui convient. L'interface présentée par *la figure IV.15* permet au responsable d'export d'ajouter une facture.

Ajouter une facture

Code Facture:

Description:

Quantité:  
0

Taxe:  
0

Remise:  
0

montant:

montant plus taxe:

Liste Des Exports

Ajouter Une Facture

JUN 1, 2018, 8:11:06 PM

*Figure IV.15 - Interface d'ajout d'une facture*

L'interface de téléchargement d'une facture est accessible en cliquant sur l'icône qui convient. L'interface présentée par *la figure IV.16* permet au responsable d'export de télécharger une facture

Télécharger la facture

Facture: Code Facture: zz  
Description: zzz

Prix et mode de paiement:  
Quantité: 77  
Taxe: 78  
Remise: 77  
montant: 77  
montant plus taxe: 77  
Total a payer: 77  
mode de paiement: yyyy  
délai de paiement: 78  
destination final: ttt

Client:  
code client 12345678  
Nom et prénom louerr  
Adresse hh  
Pays r

EXPORTER ANNULER

Liste Des Exports

Ajouter Une Facture

JUN 2, 2018, 2:52:40 AM

*Figure IV.16 - Interface de téléchargement d'une facture*

## IV.4 Test et validation

Cette phase consiste à effectuer des jeux de tests pour les principaux cas d'utilisation du système pour assurer le bon fonctionnement de l'application et afin de respecter plusieurs critères pris en charge comme l'ergonomie, la sécurité, la rapidité etc.

Pour ce faire, nous avons choisi la gestion des productions, la gestion des finitions et la gestion des commandes pour exécuter les tests cités dans la suite. Les résultats des tests restent valides pour toutes les autres gestions.

### IV.4.1 Critères ergonomiques

Les critères ergonomiques sont des caractéristiques de l'interface qui vont déterminer son utilisabilité (ou ergonomie). Ils permettent d'identifier les problèmes d'ergonomie d'une interface. Ils sont généralement utilisés avant d'engager les tests d'utilisabilité [17].

Le thème que nous avons adopté par l'application est inspiré des couleurs et du logo de l'entreprise comme le montre les captures d'écran représentés par les figures qui suivent.

#### IV.4.1.1 Guidage

L'ensemble des moyens mis en œuvre pour orienter et conduire l'utilisateur lors de ses interactions avec l'ordinateur. En effet, le guidage rend le logiciel plus convivial.

##### IV.4.1.1.1 Incitation

L'ensemble les moyens mis en œuvre pour amener les utilisateurs à effectuer des actions spécifiques en lui fournissant des indices.

Par exemple :

- ✓ Guider les entrées de données en indiquant le format adéquat et les valeurs acceptables : Date : jj/mm/aaaa, comme il est présenté par *la figure IV.17*.

## Chapitre IV. Réalisation

---

The screenshot shows a mobile application interface for WICMIC. On the left, there is a sidebar with icons and labels: 'Liste Des Commande', 'Ajouter Une Nouvelle Commande', 'Liste Des Produit', and 'Ajouter Un Produit'. The main area contains several input fields with placeholder text: 'Description:', 'Saison:', 'Quantité:', 'Prix Unitaire:', 'Prix Total:', and 'Date Prévue:'. The 'Date Prévue:' field is highlighted with a red border, indicating it is the current focus. Below this field is a date input field containing 'jj/mm/aaaa' and two buttons: a blue one with a right arrow icon and a red one with a white 'X' and the word 'ANNULER'.

Figure IV.17 - Exemple de l'incitation respectée dans l'application : Format de Date

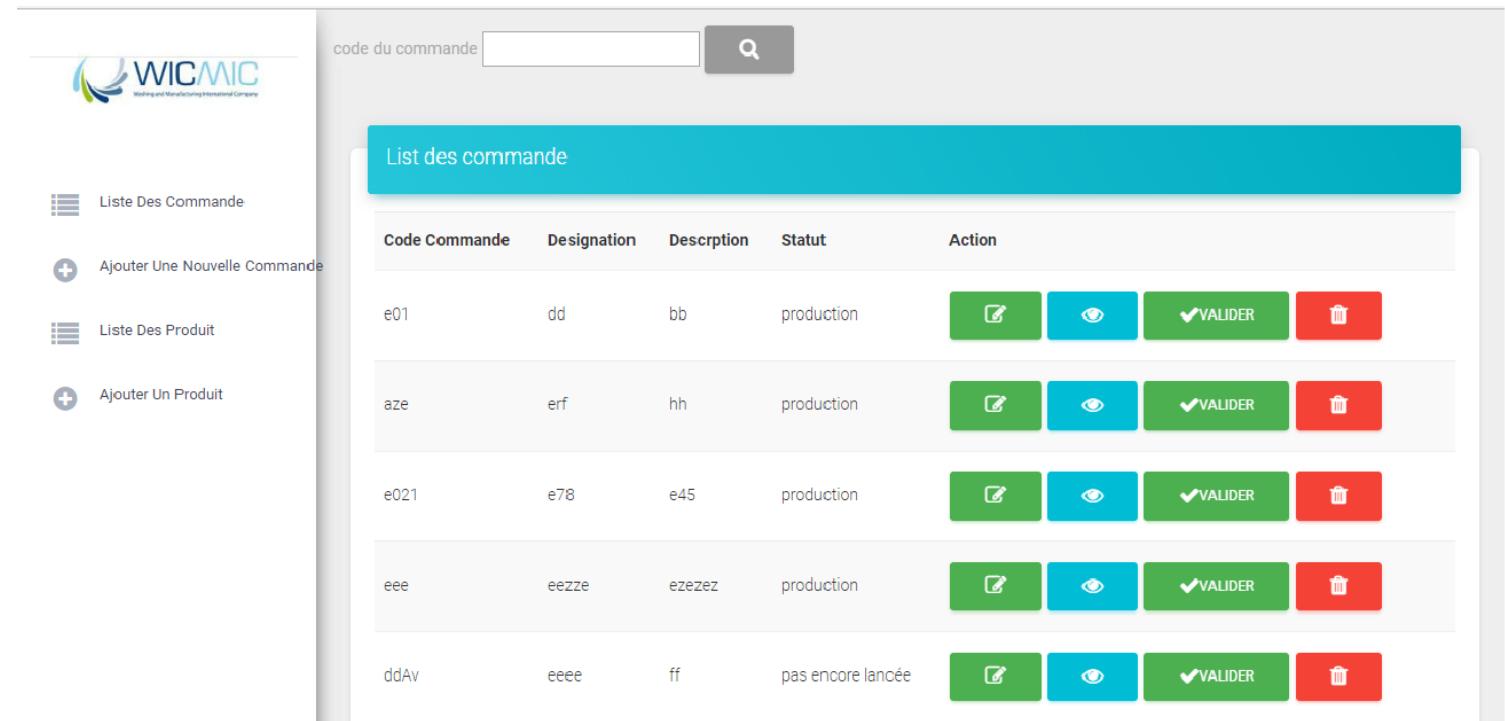
- ✓ Pour chaque champ de données, fournir un label illustré par la figure IV.18.

The screenshot shows a mobile application interface for WICMIC. On the left, there is a sidebar with icons and labels: 'Liste Des Commande', 'Ajouter Une Nouvelle Commande', 'Liste Des Produit', and 'Ajouter Un Produit'. The main area contains several input fields with clear labels above them: 'Numero de téléphone', 'Email:', 'Marque:', 'Adresse:', 'Ville:', 'Pays:', 'Code Postal:', and 'Rue:'.

Figure IV.18 - Exemple de l'incitation respectée dans l'application : Fournir un label clair

#### IV.4.1.1.2 Groupement/Distinction entre Items :

Consiste à s'assurer que l'interface regroupe ce qui est de même nature de façon cohérent et ordonner comme indique le tableau des articles illustré par *la figure IV.19*.



The screenshot shows a user interface for managing commands. On the left, there's a sidebar with icons for 'Liste Des Commande', 'Ajouter Une Nouvelle Commande', 'Liste Des Produit', and 'Ajouter Un Produit'. The main area has a search bar at the top with a placeholder 'code du commande' and a magnifying glass icon. Below it is a table titled 'List des commande' with columns: 'Code Commande', 'Designation', 'Description', 'Statut', and 'Action'. The table contains five rows of data:

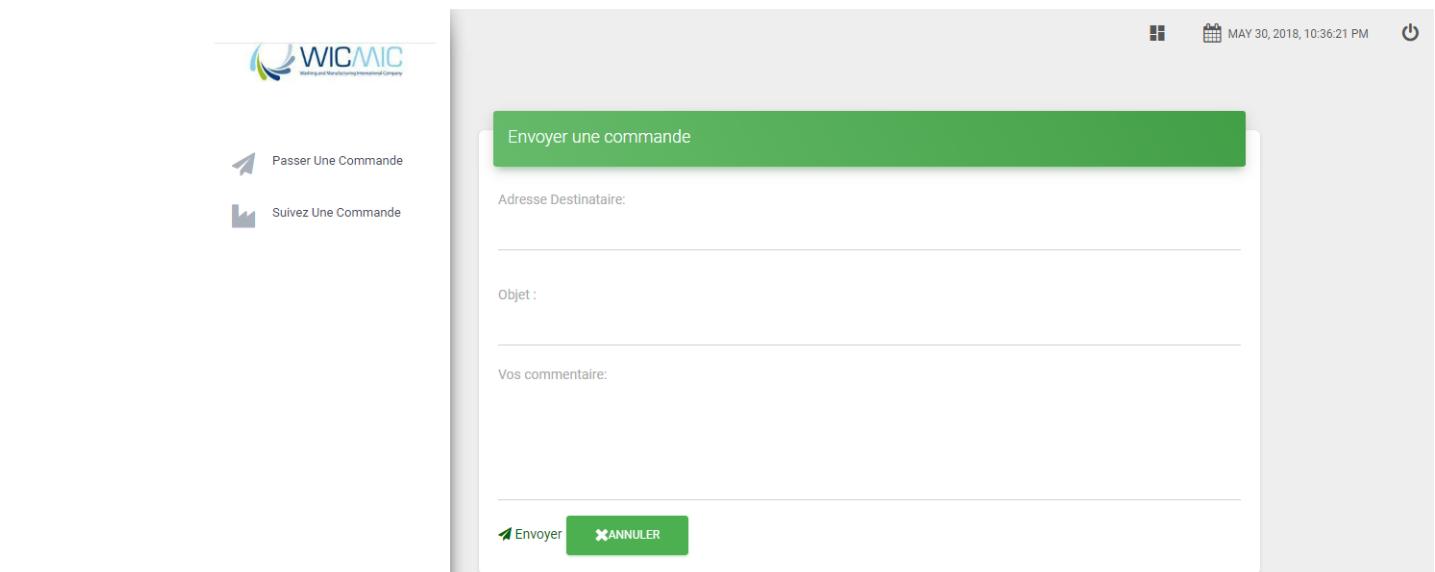
| Code Commande | Designation | Description | Statut            | Action |
|---------------|-------------|-------------|-------------------|--------|
| e01           | dd          | bb          | production        |        |
| aze           | erf         | hh          | production        |        |
| e021          | e78         | e45         | production        |        |
| eee           | eezze       | ezevez      | production        |        |
| ddAv          | eeee        | ff          | pas encore lancée |        |

Figure IV.19 - Exemple de Groupement entre items respecté dans l'application

#### IV.4.1.1.3 Lisibilité :

Le critère Lisibilité concerne la présentation des informations sur l'écran de façon claire et lisible aux utilisateurs de l'application.

Par exemple : Les champs du client sont clairs et faciles à comprendre comme le montre *la figure IV.20*.



The screenshot shows a 'Passer Une Commande' (Place Order) form. At the top right, there are icons for a grid, a calendar, and a power button, with the date 'MAY 30, 2018, 10:36:21 PM' displayed. The form has a green header bar with the text 'Envoyer une commande'. Below it are three input fields: 'Adresse Destinataire:', 'Objet:', and 'Vos commentaire:'. At the bottom are two buttons: a green 'Envoyer' (Send) button and a red 'ANNULER' (Cancel) button.

Figure IV.20 - Exemple qui montre la lisibilité respectée dans l'application

#### IV.4.1.2 Charge du travail :

L'ensemble d'éléments qui permettent de réduire la charge perceptive des utilisateurs, aussi d'augmenter l'efficacité du dialogue.

##### IV.4.1.2.1 Brièveté

Il s'agit ici de limiter autant que possible le travail de lecture, d'entrée et les étapes par lesquelles doivent passer les utilisateurs.

Par exemple : Proposer des valeurs par défaut comme le montre *la figure IV.21* de l'ajout d'une finition.

The screenshot shows a user interface for managing production or finishing touches. On the left, there's a sidebar with icons for 'Liste Des Productions', 'Ajouter Une Production', 'Liste Des Finitions', and 'Ajouter Une Finition'. The main area contains several input fields:

- Quantité total:** 0 (highlighted with a red box)
- Quantité demandée:** 0
- Type de finition:** (empty field)
- Groupe de finition:** (empty field)
- Code de finition:** (empty field)
- Statut:** false (highlighted with a red box)

*Figure IV.21 - Exemple qui montre la brièveté respectée dans l'application*

##### IV.4.1.2.2 Action minimale

Dans ce cadre nous avons limité les étapes par lesquelles doivent passer un utilisateur.

Par exemple : Lors de la modification d'un utilisateur les anciennes informations sont déjà affichée comme le montre *la figure IV.22*.

The screenshot shows a user interface for modifying a command. On the left, there is a sidebar with the WICMIC logo and four menu items: "Liste Des Commande", "Ajouter Une Nouvelle Commande", "Liste Des Produit", and "Ajouter Un Produit". The main area has a teal header bar with the text "Modifier la commande". Below the header, there are several input fields:

- Code Commande: e01
- Désignation: dd
- Description: bb
- Saison: bbb
- Quantité: 80
- Prix Unitaire: 7
- Prévu Total: (This field is partially visible at the bottom)

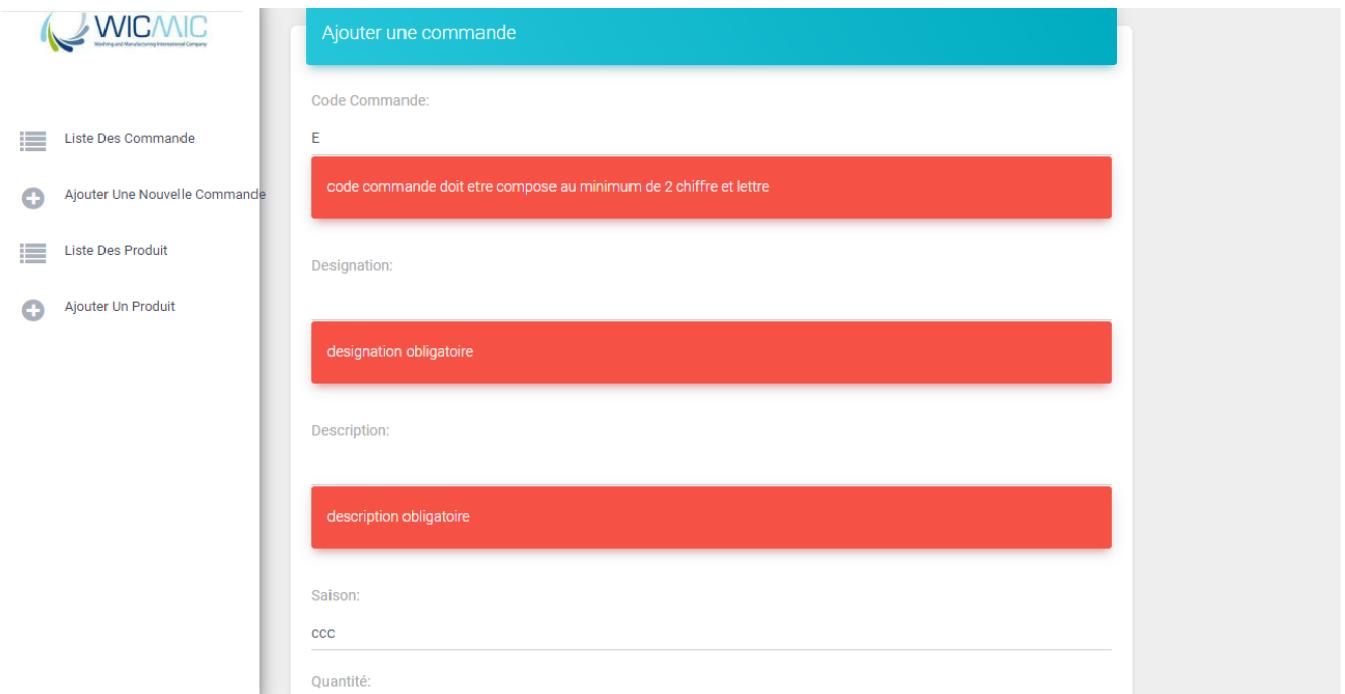
Figure IV.22 - Exemple qui illustre l'utilisation des actions minimales respectée dans l'application

### IV.4.1.3 Gestion des erreurs

C'est l'ensemble d'élément qui permet d'éviter et de réduire les erreurs ainsi de les corriger lorsqu'ils surviennent

#### IV.4.1.3.1 Qualité des messages d'erreurs :

Dans ce cadre nous avons essayé de mettre des messages d'erreur indicative et compréhensible par les utilisateurs pour qu'il puisse comprendre l'erreur et la manière de la corriger Comme indique la figure IV.23:

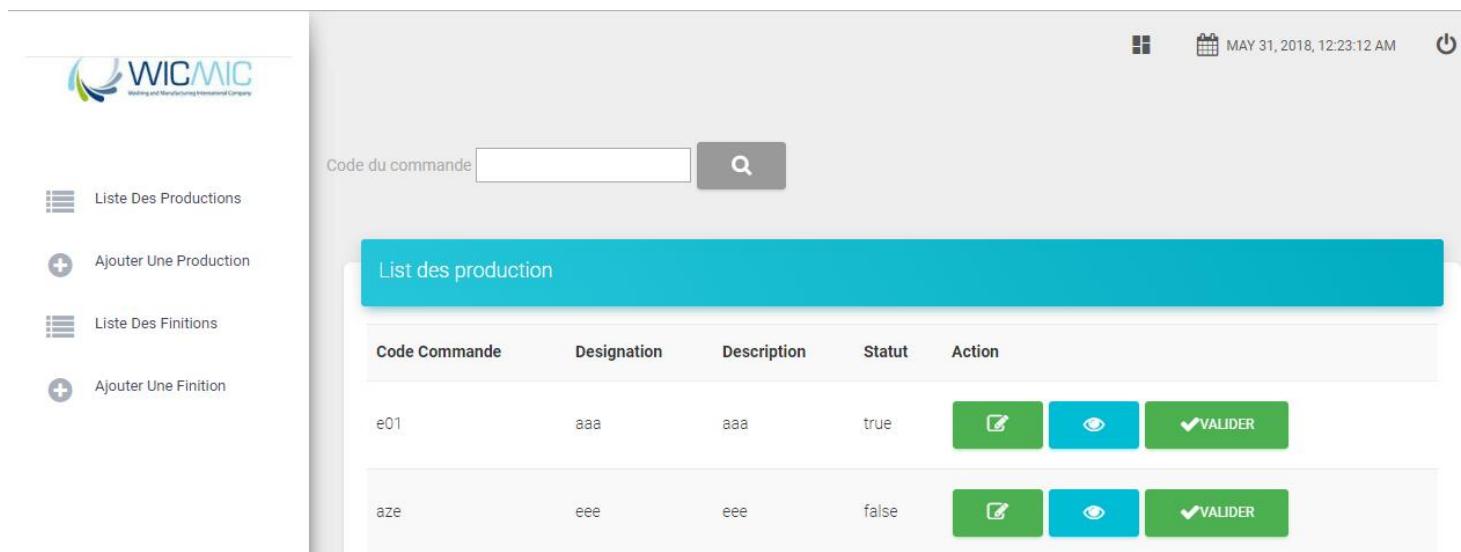


The screenshot shows a user interface for adding a new command. The top bar is teal with the title "Ajouter une commande". On the left, there's a sidebar with icons for "Liste Des Commande", "Ajouter Une Nouvelle Commande", "Liste Des Produit", and "Ajouter Un Produit". The main area has fields for "Code Commande" (containing "E"), "Designation" (containing "designation obligatoire"), "Description" (containing "description obligatoire"), and "Saison" (containing "ccc"). Below these is a "Quantité:" field. A red validation message at the top right says "code commande doit etre compose au minimum de 2 chiffre et lettre".

Figure IV.23 - Exemple qui montre la qualité des messages générés par l'application

#### IV.4.1.4 Homogénéité/Cohérence :

Dans notre application nous nous sommes assuré que les interfaces conservent le même modèle d'affichages mais des contextes différents. Par exemple, le même modèle d'affichage pour tous les services de gestion comme il est illustré par *les figures IV.24 et IV.25*.



The screenshot shows a production management screen. The top bar includes the WICMIC logo, a search bar with "Code du commande" and a magnifying glass icon, and a date/time stamp "MAY 31, 2018, 12:23:12 AM". The sidebar on the left has icons for "Liste Des Productions", "Ajouter Une Production", "Liste Des Finitions", and "Ajouter Une Finition". The main area has a teal header "List des production". Below it is a table with columns: "Code Commande", "Designation", "Description", "Statut", and "Action". Two rows of data are shown:

| Code Commande | Designation | Description | Statut | Action |
|---------------|-------------|-------------|--------|--------|
| e01           | aaa         | aaa         | true   |        |
| aze           | eee         | eee         | false  |        |

Figure IV.24 - Homogénéité et cohérence : Gestion de production

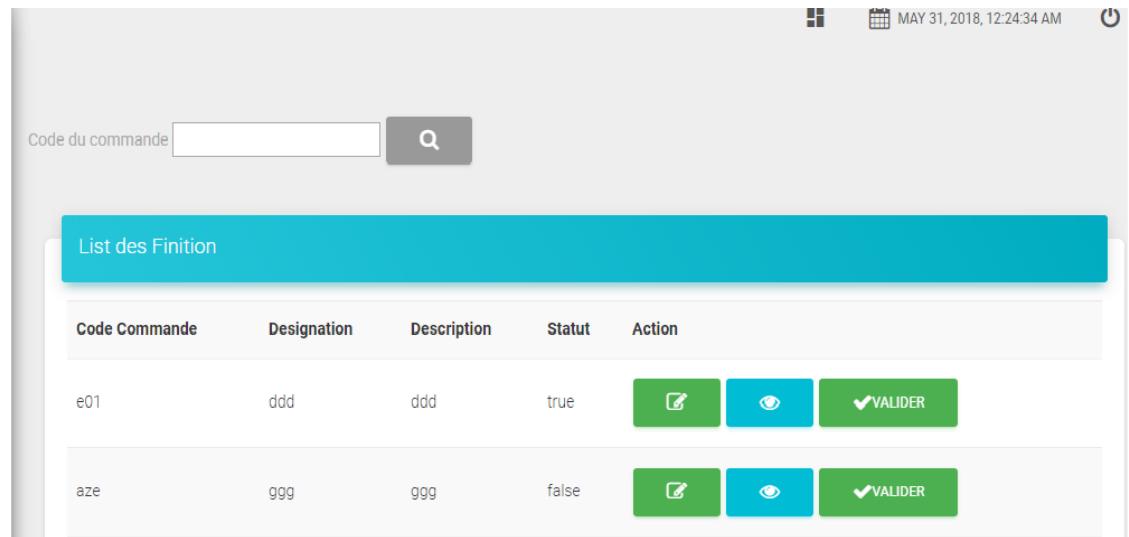


Figure IV.25 - Homogénéité et cohérence : Gestion de finition

#### IV.4.1.5 Contrôle explicite

C'est la prise en compte par le système des actions explicites des utilisateurs et le contrôle qu'ont les utilisateurs sur le traitement de leur actions (autrement dit c'est l'utilisateur qui doit décider).

##### IV.4.1.5.1 Contrôle utilisateur

On doit laisser l'initiative à l'utilisateur de décider quoi faire par exemple dans la suppression comme montre *la figure IV.26* et *la figure IV.27*.

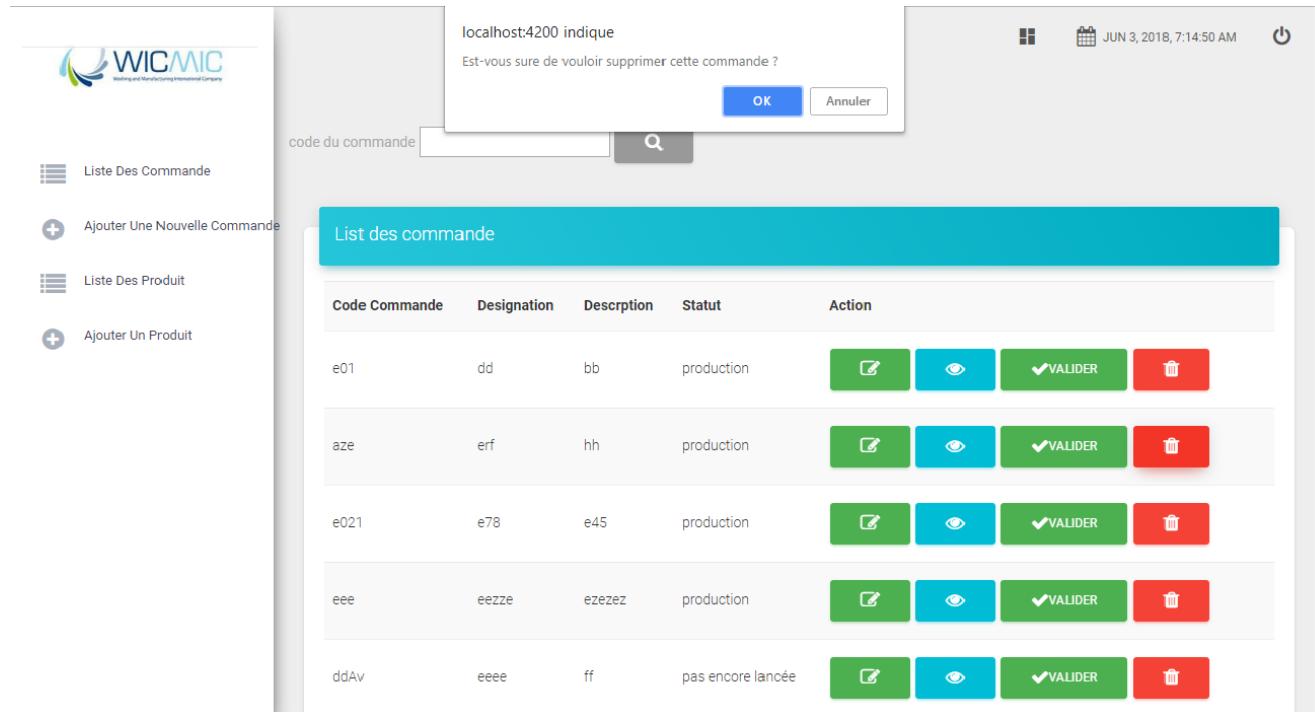
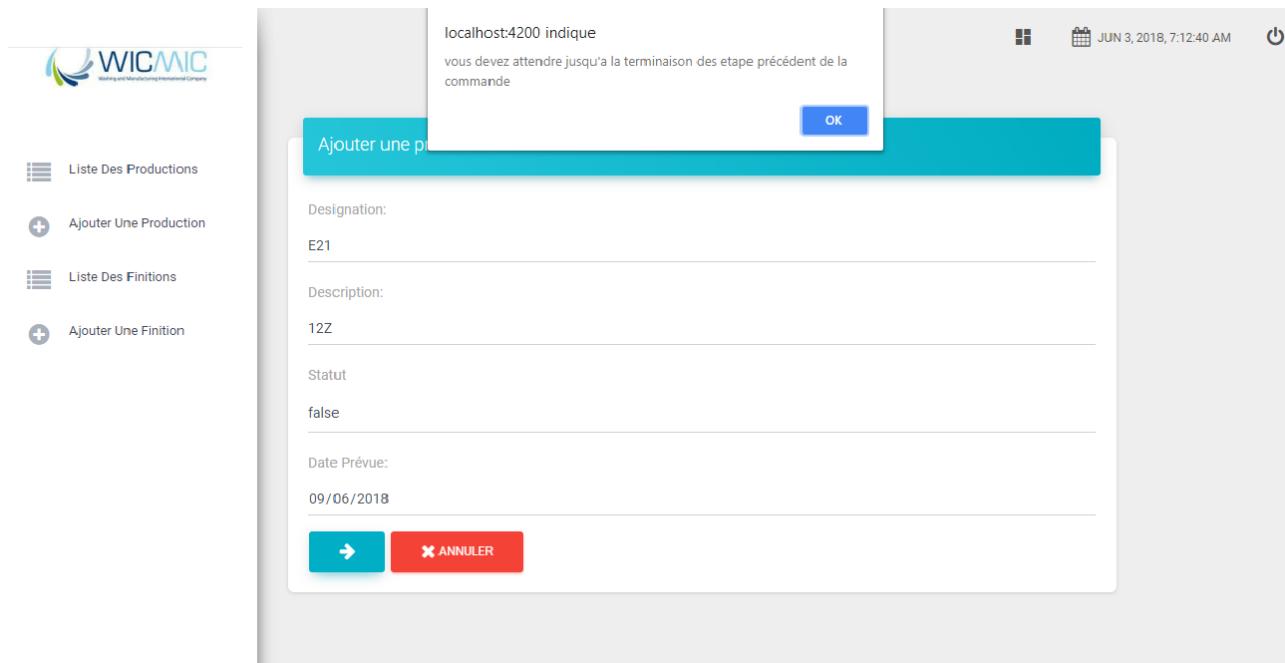


Figure IV.26 - Exemple de Contrôle utilisateur utilisé dans l'application



*Figure IV.27 - Exemple de Contrôle utilisateur utilisé dans l'application*

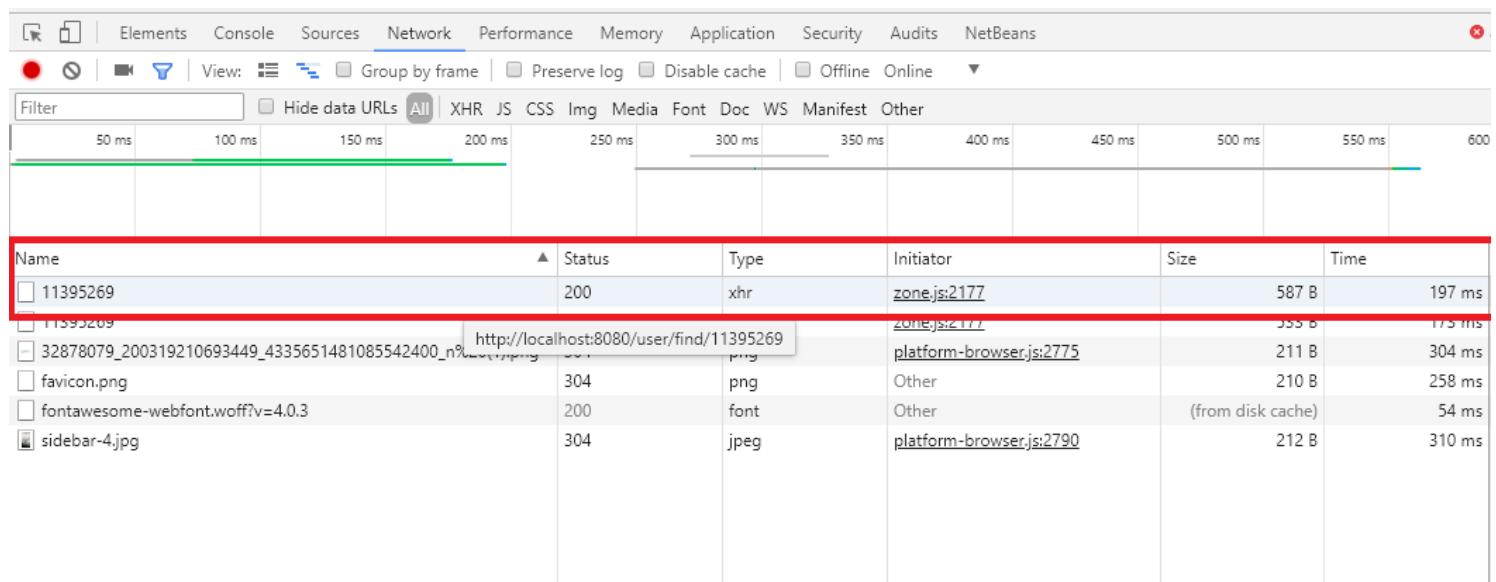
#### **IV.4.1.6 Rapidité et performance de l‘application :**

La rapidité d'une application web est l'un des critères les plus importants pour que l'application réussisse dans le marché. Pour ce test nous avons utilisé le PC LENOVO dont les caractéristiques sont décrites dans la section 13.

#### **IV.4.1.6.1 Le temps de réponse de quelques services de notre application :**

**Identification :** L'état initial : 0ms.

L'application a pris 197ms pour se connecter comme indique *la figure IV.28* :



*Figure IV.28 - Calcul de temps de réponse pour l'identification*

L'affichage de la liste des commandes prend 42 ms ce qui montre la rapidité de temps de réponse d'où la fiabilité de notre système comme illustré par la figure IV.29 :

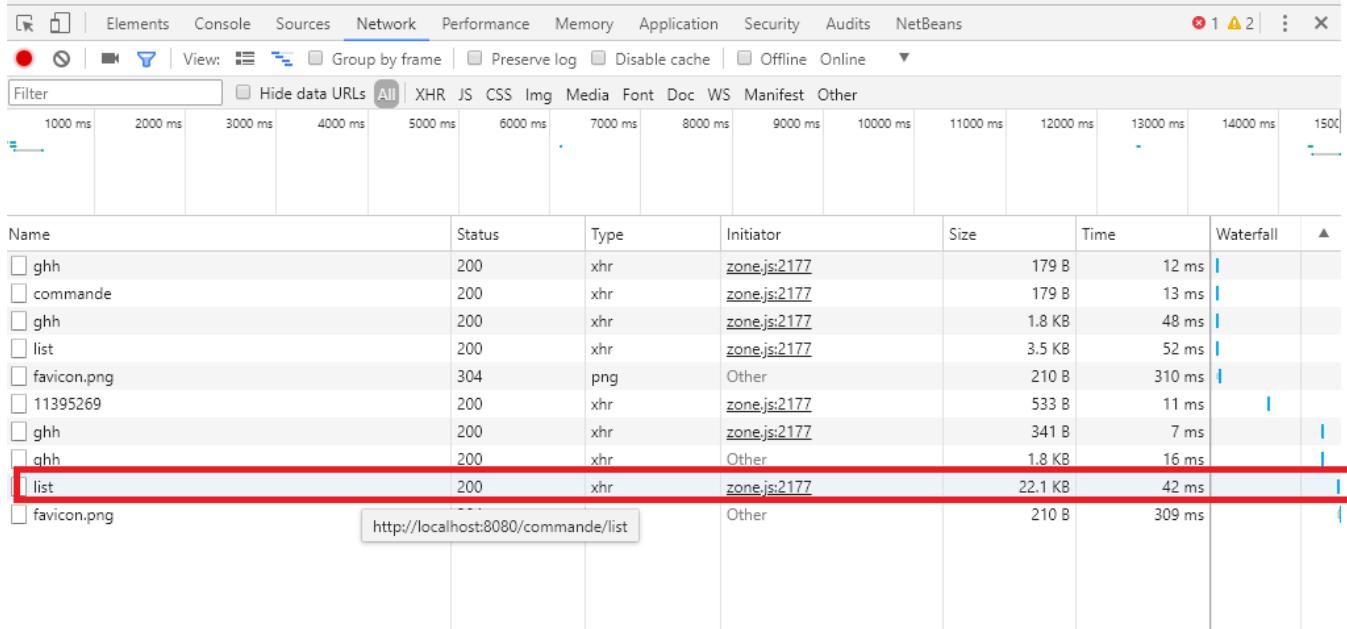


Figure IV.29 - Calcul de temps de réponse pour le cas d'affichage de la liste des commandes

Le temps de réponse d'ajout d'une commande est 11 ms ce qui montre encore la rapidité du temps de réponse de notre application comme illustré par la figure IV.30 :

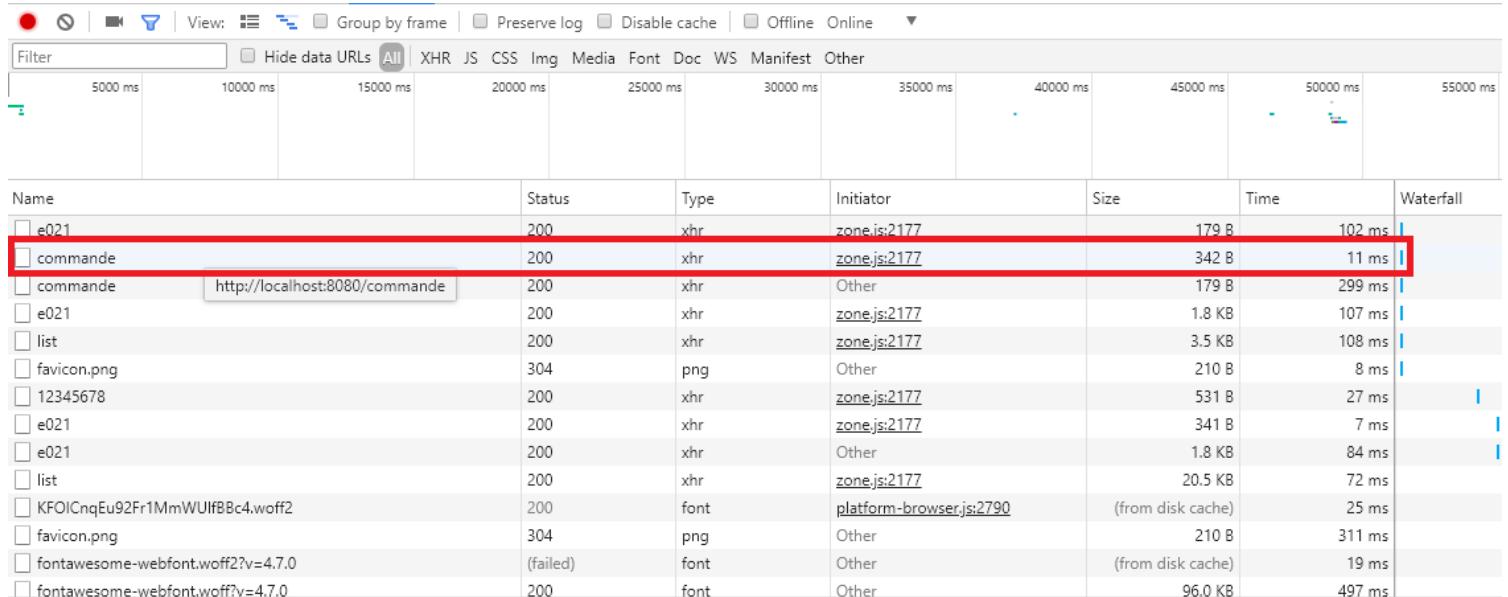
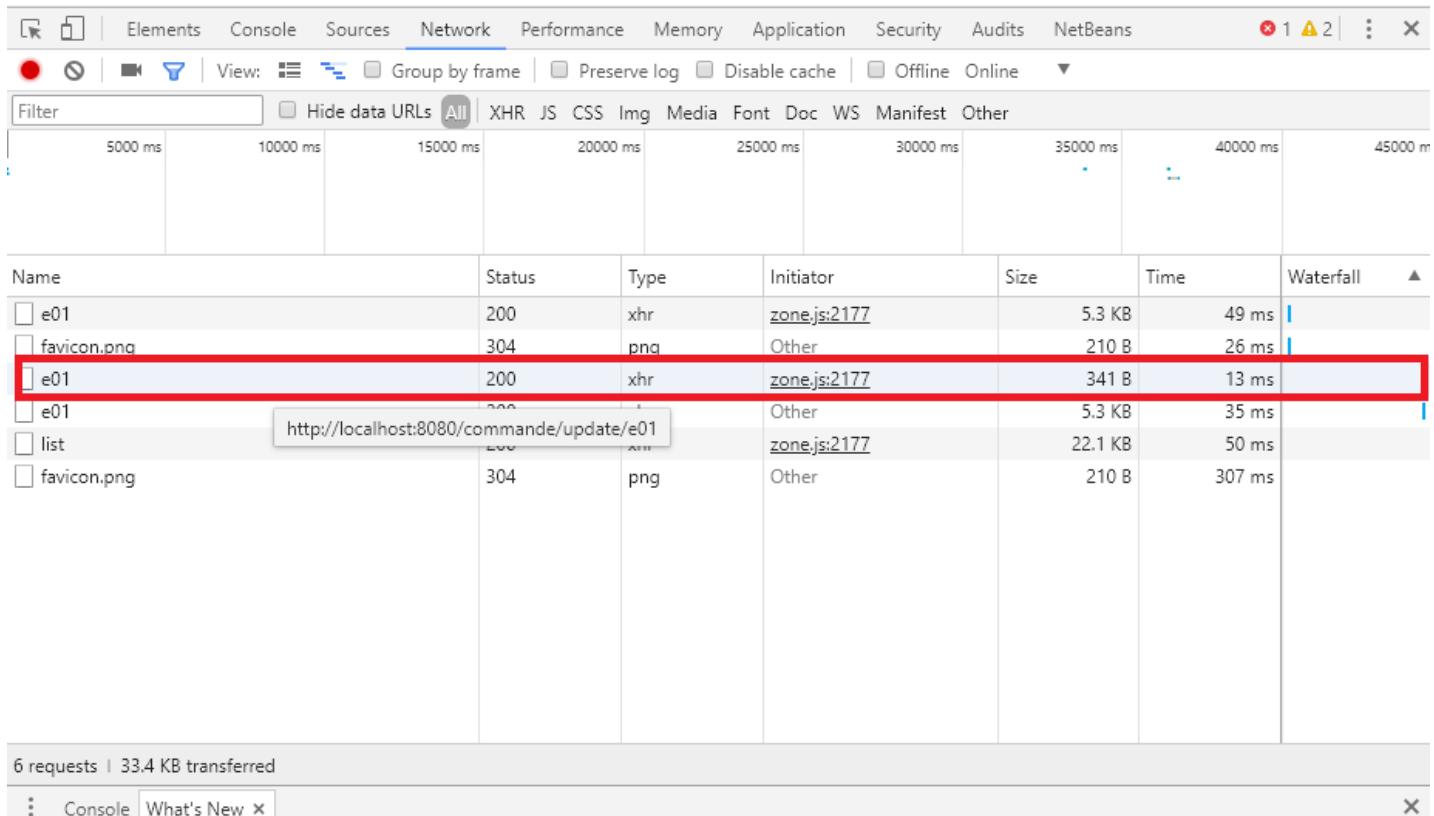


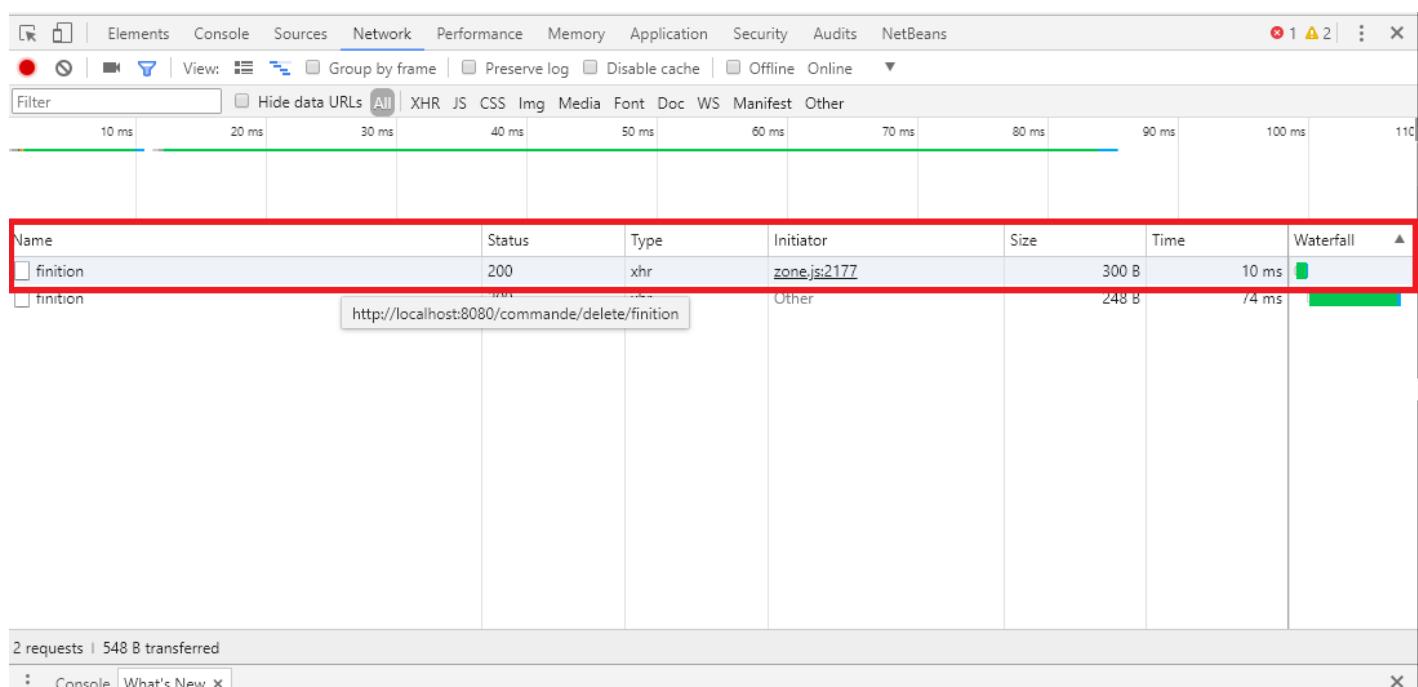
Figure IV.30 - Calcul de temps de réponse pour le cas d'ajouter une commande

Le temps de réponse de la modification d'une commande prend 13 ms ce qui montre encore la rapidité du temps de réponse de notre application comme illustré par *la figure IV.31* :



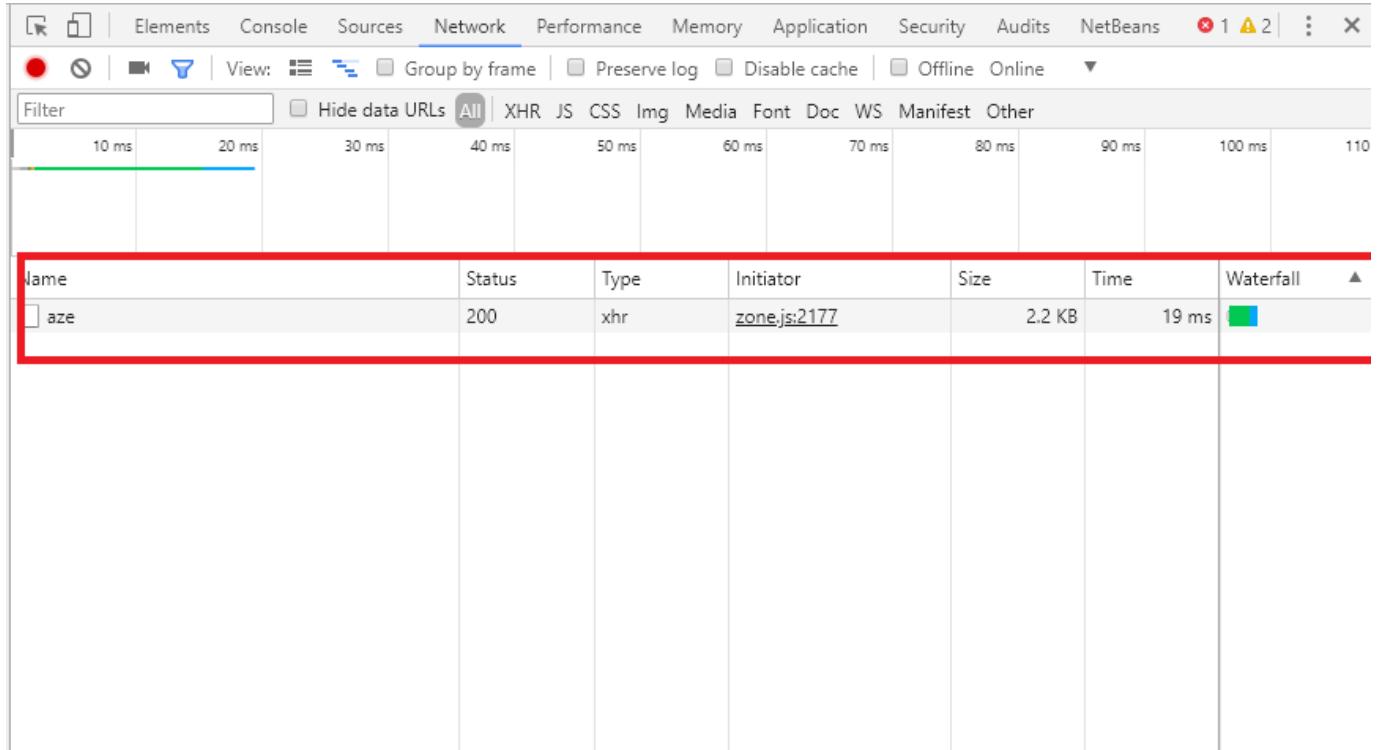
*Figure IV.31 - Calcul de temps de réponse pour le cas de modifier une commande*

Le temps de réponse de la suppression égal à 10ms ce qui montre la rapidité de notre application comme indique *la figure IV.32* :



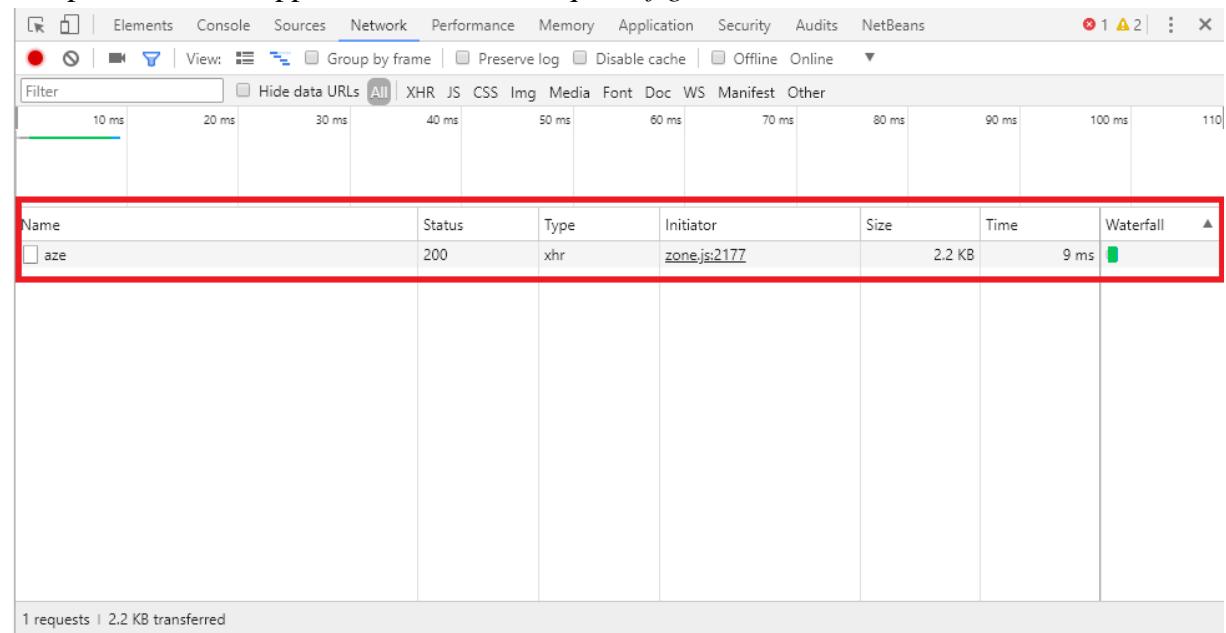
*Figure IV.32 - Calcul de temps de réponse pour le cas de supprimer une commande*

La recherche d'une commande dans une List de 8 documents prend 19 ms, ce qui montre la rapidité de notre application comme indique *la figure IV.33* :



*Figure IV.33 - Calcul de temps de réponse pour le cas de chercher une commande*

La recherche de la même commande dans une liste de 3 documents prend 9 ms ce qui montre la rapidité de notre application comme indique *la figure IV.34* :



*Figure IV.34 - Calcul de temps de réponse pour le cas de chercher une commande*

## Conclusion

Durant ce chapitre, nous avons présenté les technologies, les Framework et les logiciels que nous avons utilisés durant ce projet. Ensuite, nous avons présenté une description de quelques interfaces graphiques de notre application. Enfin, nous avons présenté quelques tests effectués sur notre application.

## Conclusion générale et perspectives

Dans notre projet de fin d'étude, le travail a été réalisé au profit de la société MIC. Dans ce cadre, nous avons mis en place un système qui gère les fonctions des services au sein de la société. Pour ce faire, nous avons commencé par la description des besoins des futurs utilisateurs du système. Par la suite, nous nous sommes basés sur une étude de quelques solutions existantes et à comprendre la présentation générale de notre application afin de dégager les différents besoins dont l'application est chargé d'y répondre. Ces besoins ont été bien traités et analysés dans la phase d'analyse des besoins fonctionnels à l'aide du langage de modélisation UML. La phase suivante, englobant une présentation globale de l'architecture utilisée par le système a été réalisée pour aboutir à une meilleure solution ainsi qu'une conception détaillée des besoins de la société. Ensuite, les technologies mise en œuvre sont détaillées dans la partie réalisation avec quelques interfaces graphiques du système.

Malgré toutes les difficultés rencontrées au niveau du processus de développement et les contraintes de temps, nous avons réussi à réaliser la totalité de notre application.

Ce stage de fin d'études nous a été une occasion pour côtoyer le monde des entreprises, mais avec plus de responsabilités. Certes, il nous a permis d'enrichir nos connaissances académiques acquises tout au long de nos études supérieures à la faculté des sciences de Bizerte. L'expérience auprès de MIC était très enrichissante et bénéfique pour notre formation. Nous avons eu l'opportunité de travailler au sein d'une entreprise et d'acquérir un savoir non négligeable, une expérience en gestion des projets avec l'amélioration en termes de communication, collaboration et adaptation avec l'environnement professionnel

Pour ce qui est de perspectives, notre application peut évoluer en un progiciel de gestion intégrée. Nous proposons d'intégrer quelques fonctionnalités, tels que les services de notifications pour les bons de commandes reçus et les factures à envoyer. Il est envisageable aussi de rajouter la gestion des statistiques et un tableau de bord pour consulter les différents produits du stock et la distribution mensuelles des ventes par client, et. Aussi, nous proposons une application mobile pour suivre l'état de la commande par des smartphones quel que soit le lieu où on se trouve.

## Webographie

- [1]: Alibaba.com : <https://french.alibaba.com/product-detail/erp-for-textiles-industries-texams-erp-textiles-application-management-system-50007646101.html?spm=a2700.details.toavopna6.14.75e5327eSVzBPU>, Date de la dernière consultation le : 07/05/2018
- [2] : <https://www.supinfo.com/articles/single/3099-waterfall-model-modele-cascade>, Date de la dernière consultation le 29/04/2018
- [3]: cedric.babault : <http://cedric.babault.free.fr/rapport/node4.html>, Date de la dernière consultation le : 14/05/2018
- [4] : supinfo : <https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>, Date de la dernière consultation le : 14/05/2018
- [5] : <http://uml.free.fr/cours/i-p17.html>, Date de la dernière consultation : 29/05/2018
- [6] : <https://fr.wikipedia.org/wiki/MongoDB>, Date de la dernière consultation le 21/05/2018
- [7] : <https://www.supinfo.com/articles/single/1178-creer-serveur-web-avec-nodejs>, Date de la dernière consultation le 21/05/2018
- [8] : Edureka : <https://www.edureka.co/blog/angular-tutorial/>, Date de la dernière consultation le 20/05/2018
- [9] : <https://o7planning.org/fr/11267/le-tutoriel-de-spring-boot-pour-les-debutants>, Date de la dernière consultation le 21/05/2018
- [10] : <https://www.mongodb.com/json-and-bson>, Date de la dernière consultation le 26/05/2018
- [11] : [https://www.eyrolles.com/Chapitres/9782212126587/Chap-2\\_Goncalves.pdf](https://www.eyrolles.com/Chapitres/9782212126587/Chap-2_Goncalves.pdf), Date de la dernière consultation le 21/05/2018
- [12] : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203257-html5-hypertext-markup-langage5-definition-traduction/>, Date de la dernière consultation le 23/05/2018
- [13] : <http://glossaire.infowebmaster.fr/css/>, Date de la dernière consultation le 23/05/2018
- [14] : <https://www.commentcamarche.com/contents/577-javascript-introduction-au-langage-javascript>, Date de la dernière consultation le 23/05/2018
- [15] : <https://blog.xebia.fr/2014/03/12/angular-et-typescript-un-mariage-heureux/>, Date de la dernière consultation le 26/05/2018
- [16] : <https://www.commentcamarche.com/contents/1141-introduction-a-uml>, Date de la dernière consultation le 25/05/2018
- [17] : <http://www.usabilis.com/criteres-ergonomiques-bastien-et-schapin/>, Date de la dernière consultation le 01/06/2018