



UNIVERSITÉ  
DE LORRAINE



nancy Charlemagne  
Département Informatique

IUT Nancy Charlemagne  
Université de Lorraine  
2 ter boulevard Charlemagne  
BP 55227  
54052 Nancy Cedex

Département informatique

## **Développement d'un logiciel type ERP**

Rapport de stage de BUT informatique  
SYG Software

Mirac DEMIRCI  
Fanny BINET  
BUT2 Informatique 2024/2025





UNIVERSITÉ  
DE LORRAINE



nancy Charlemagne  
Département Informatique

IUT Nancy Charlemagne  
Université de Lorraine  
2 ter boulevard Charlemagne  
BP 55227  
54052 Nancy Cedex

Département informatique

## **Développement d'un logiciel type ERP**

Rapport de stage de BUT informatique  
SYG Software

Mirac DEMIRCI  
Fanny BINET  
BUT2 Informatique 2024/2025

## Remerciement

Je souhaite sincèrement remercier à l'ensemble de l'équipe de **SYG Software** pour m'avoir accueilli durant mon stage de dix semaines. Ce fut une expérience aussi enrichissante que inspirante, tant sur le plan professionnel que personnel.

Je remercie tout particulièrement **Monsieur Baptiste MINCK** pour m'avoir donné l'opportunité d'intégrer l'entreprise et d'évoluer au sein d'un environnement stimulant et bienveillant. Sa confiance m'a permis de participer activement à des projets concrets et formateurs.

Je souhaite également adresser un grand merci à **Benoît Bourgeois** et **Mireille Vignal**, qui m'ont accompagné tout au long de cette aventure. Leur patience, leurs conseils avisés et leur disponibilité ont joué un rôle essentiel dans ma progression. Grâce à eux, j'ai pu non seulement améliorer mes compétences en développement, mais aussi découvrir les exigences, les codes et la dynamique du monde professionnel.

Enfin, je remercie toute l'équipe de SYG pour leur accueil, leur bonne humeur et leur esprit d'entraide. Travailler à leurs côtés m'a permis de me sentir rapidement intégré et motivé à donner le meilleur de moi-même.

Ce stage a été une étape précieuse dans mon parcours, et je quitte cette expérience avec de nouvelles connaissances, une vision plus claire de mon avenir professionnel, et des souvenirs très positifs.

# Table des matières

<b>Introduction.....</b>	<b>1</b>
Présentation de l'entreprise.....	3
1. Le service informatique rattaché.....	4
2. Mission.....	4
<b>Organisation du Stage.....</b>	<b>5</b>
Chronologie du stage.....	5
<b>Formation OpenClassroom.....</b>	<b>6</b>
1. Module 1 « Débutez avec Angular ».....	6
2. Module 2 « Complétez vos connaissances sur Angular ».....	7
3. Module 3 « Perfectionnez-vous sur Angular ».....	7
<b>Développement de l'édition d'étiquettes code barre.....</b>	<b>9</b>
1 Introduction au projet.....	9
2. Développement du projet d'édition et d'impression d'étiquettes de codes-barres.....	9
2.1. Objectifs et spécifications du projet.....	9
2.2. Défis rencontrés et solutions mises en place.....	11
3. Optimisation du Code et Amélioration des Performances.....	12
4. Maquette Attendu :.....	13
5. Résultat final.....	13
6. Implémentation.....	14
7. Gestion des erreurs lors de l'importation de données.....	15
7.1. Identification des données non conformes.....	15
7.2. Gestion des doublons et des données manquantes.....	15
8. Gestion des erreurs sur le nombre d'impressions.....	16
Conclusion : Développement de l'édition d'étiquettes code-barres.....	16
Compétences techniques développées :.....	16
Évolution personnelle et ressentis.....	16
<b>Développement de la gestion des promotions.....</b>	<b>17</b>
Introduction au projet.....	17
Objectifs et fonctionnalités principales.....	18
1. Affichage et gestion des promotions existantes.....	18
2. Création d'une nouvelle promotion.....	18
3. Ajout des articles concernés.....	19
3.1. Recherche manuelle.....	19
3.2. Recherche multiple avec un second pop-up.....	19
3.3. Import de fichier CSV.....	20
4. Définition des réductions et remises.....	20
5. Modification et mise à jour d'une promotion existante.....	20
Gestion de la cohérence des données.....	21

1. Vérification des champs obligatoires.....	22
2. Validation des dates de la promotion.....	22
3. Contrôle du type de remise appliquée.....	22
4. Vérification de la présence d'articles dans la promotion.....	22
Difficulté rencontrée – Intégration avec le backend et cohérence des données.....	23
Compétences développées.....	23
1. Maîtrise de la logique front-end / back-end.....	23
2. Manipulation de données complexes.....	24
3. Validation des formulaires et retours utilisateurs.....	24
4. Expérience utilisateur (UX).....	24
5. Gestion des cas particuliers.....	24
6. Communication avec l'équipe et compréhension métier.....	24
Explication du Code pour l'Ajout d'une Promotion.....	25
Conclusion du développement de la gestion des promotions.....	27
<b>Développement de nouvelles pages : montée en compétence et gain de productivité.....</b>	<b>28</b>
Page de gestion des mouvements de caisse et de stock :	28
Page de gestion de la carte de fidélité :	29
Page de gestion des vendeurs et des commissions :	31
<b>Exportation des données en fichier Excel.....</b>	<b>33</b>
Contexte :	33
Difficultés rencontrées :	33
Résultat obtenu.....	34
<b>Conclusion Générale.....</b>	<b>35</b>
<b>Bibliographie.....</b>	<b>36</b>
<b>Glossaire.....</b>	<b>37</b>
<b>Organigramme SYG Software.....</b>	<b>39</b>

# Introduction

Dans le cadre de mon parcours en BUT informatique, j'ai eu l'opportunité de réaliser un stage de 10 semaines au sein de l'entreprise **SYG Software**, spécialisée dans la création et le développement de solutions logicielles. Ce stage m'a permis de mettre en pratique mes connaissances théoriques et d'acquérir de nouvelles compétences, notamment dans le domaine du développement d'**ERP**<sup>[\*]</sup>. En tant que stagiaire, j'ai été encadré par **Benoit Bourgeois**, mon maître de stage, qui m'a guidé tout au long.

Parmi les missions principales qui m'ont été confiées, j'ai participé au développement de l'interface utilisateur (**frontend**<sup>[\*]</sup>) de l'ERP, ce qui m'a permis d'approfondir les fondamentaux d'**Angular**<sup>[\*]</sup>. J'ai ainsi pu approfondir mes compétences en développement web, notamment en maîtrisant la gestion des **composants**<sup>[\*]</sup>, des **services**<sup>[\*]</sup>, ainsi que la manipulation des données au sein d'une architecture. Mon travail a consisté à résoudre de nombreux problèmes techniques relatifs à l'interface utilisateur de l'ERP, et grâce aux conseils de Benoît Bourgeois, j'ai pu surmonter ces défis et améliorer mes compétences techniques.

Au-delà des aspects techniques, ce stage a été une véritable source d'apprentissage sur le plan personnel et professionnel. Il m'a permis de mieux comprendre le fonctionnement d'une entreprise de développement logiciel, la gestion de projets en équipe, ainsi que les attentes et les enjeux liés à la création d'un ERP. Une des leçons majeures que j'ai tirées de cette expérience est liée à ma perception de l'intelligence artificielle (IA). Avant ce stage, je pensais que l'IA pouvait potentiellement remplacer l'humain dans de nombreux domaines, y compris dans le développement logiciel. Cependant, au cours de mes 10 semaines d'immersion, j'ai compris que l'IA ne peut pas remplacer la créativité, la réflexion, ni la capacité à résoudre des problèmes complexes de manière autonome, des compétences essentielles dans le développement logiciel. L'IA, bien que extrêmement utile dans certains aspects du processus de développement (comme l'automatisation de tâches répétitives ou la génération de suggestions de code), reste un outil au service de l'humain, et non un remplaçant.

Ce stage a aussi été l'occasion de travailler sur des projets concrets, ce qui m'a permis de voir l'impact direct de mes actions et de mes choix techniques sur le logiciel. Ces expériences pratiques, associées à l'accompagnement de mon maître de stage, ont été cruciales pour ma montée en compétences et m'ont donné une meilleure vision de mon futur. En résumé, ce stage chez **SYG Software** a été une expérience formative, tant sur le plan technique que sur le plan

humain. J'y ai acquis des compétences pratiques en développement frontend, notamment avec **Angular**, et j'ai également enrichi ma compréhension de l'impact des nouvelles technologies, telles que l'intelligence artificielle, sur le secteur du développement logiciel. Ce stage marque une étape importante dans ma formation, et je suis désormais plus confiant dans mes capacités à aborder des projets complexes dans le domaine de l'informatique.

## Présentation de l'entreprise

**SYG Software** est une entreprise fondée en 1984, spécialisée dans le développement de logiciels de gestion commerciale et comptable. Depuis sa création, l'entreprise a su se diversifier et proposer des solutions adaptées aux besoins spécifiques de différents secteurs d'activité. Dirigée par **MINCK Baptiste**, l'entreprise emploie actuellement **8** employés. Ces derniers sont répartis entre plusieurs équipes, chacune dédiée à un domaine clé : le **développement logiciel**, la **gestion de projets**, et le **support client**. Grâce à cette organisation, SYG Software a réussi à maintenir leur service, aussi bien pour ses projets internes que pour les clients externes.

L'entreprise se distingue par la diversité des secteurs qu'elle accompagne en proposant des solutions logicielles de gestion commerciale adaptées à des secteurs très variés, comme le négoce en matériaux de construction, la papeterie, les pièces détachées automobiles, ainsi que l'emballage alimentaire. Ces logiciels permettent de gérer l'ensemble des flux commerciaux des entreprises, depuis la saisie des commandes jusqu'à la facturation, tout en optimisant la gestion des stocks et des livraisons. Une des solutions développées par l'entreprise est **Eurocom**, un logiciel complet permettant aux entreprises de mieux gérer leurs opérations commerciales et d'améliorer leur productivité.

L'entreprise accorde une grande importance à la formation et à l'accompagnement de ses clients. Elle propose des formations détaillées pour garantir une prise en main optimale des outils. **SYG Software** est d'ailleurs agréée **DATADOCK**<sup>[\*]</sup> et certifiée **QUALIOP**<sup>[\*]</sup>, ce qui atteste de la qualité de ses formations professionnelles. Ces **certifications** assurent aux clients que les prestations proposées respectent des **critères** rigoureux de **qualité** et de **professionnalisme**.

Bien que l'entreprise soit basée à **Velaine-en-Haye**, elle bénéficie d'une implantation géographique étendue, réalisant des projets pour des clients partout en **France**, mais aussi en **Guadeloupe**. **SYG Software** a su adapter ses solutions aux besoins spécifiques des entreprises, peu importe leur **localisation**.

## 1. Le service informatique rattaché

Au sein de SYG Software, j'ai intégré le service informatique spécialisé dans le développement de logiciels de type ERP. Cette équipe est dédiée à la création et à la maintenance de solutions logicielles destinées à optimiser la gestion des entreprises, en intégrant des fonctionnalités telles que la gestion commerciale, la comptabilité et la gestion des stocks. Pour ce faire, le service utilise le **framework**<sup>[\*]</sup> **Angular** et les langages **TypeScript**<sup>[\*]</sup>, **HTML**<sup>[\*]</sup> et **CSS**<sup>[\*]</sup>, afin de développer des interfaces utilisateur **dynamiques** et **réactives**.

## 2. Mission

Lors de mon stage, on m'a attribué comme mission principale de développer des **pages web** du logiciel ERP afin de répondre aux **besoins** spécifiques des clients. Cette tâche impliquait une **collaboration** avec les **équipes** et les clients pour comprendre leurs **exigences** et concevoir des **solutions** adaptées.

La **problématique** résidait dans la nécessité de se mettre à la place du client afin de concevoir des interfaces **intuitives** et **fonctionnelles**. Il était essentiel de comprendre les **processus** métiers des clients pour créer des **outils numériques** qui **optimisent** leur efficacité et **facilitent** leur travail quotidien.

Les **objectifs** visés étaient de concevoir des interfaces utilisées pour améliorer l'expérience utilisateur, de développer des outils numériques adaptés aux besoins spécifiques des clients afin d'optimiser leurs processus métiers, de faciliter l'intégration des nouvelles pages avec les systèmes existants des clients. En somme, l'objectif était de fournir aux clients des solutions web sur mesure qui répondent précisément à leurs besoins, tout en garantissant une expérience utilisateur optimale et fluide.

# Organisation du Stage

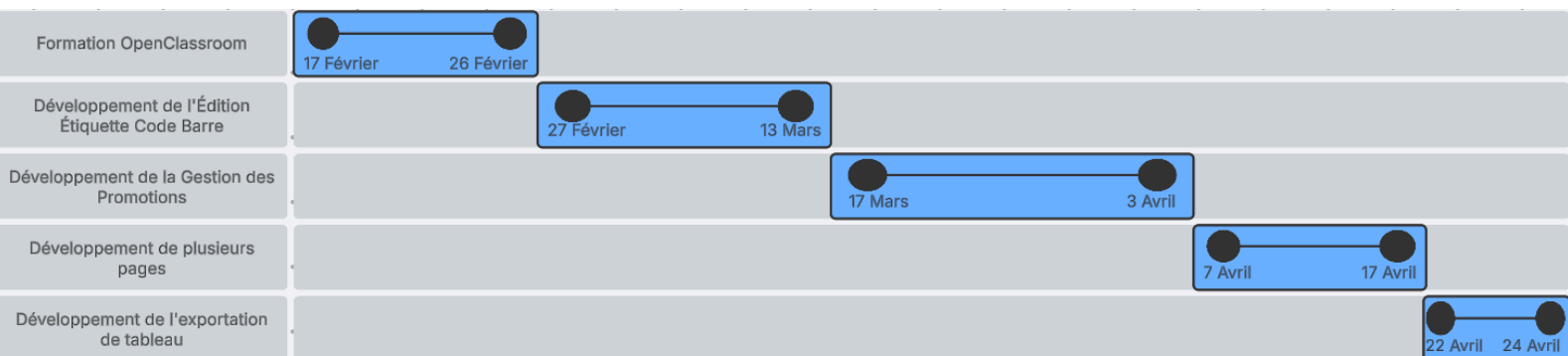


Figure 1 : Chronologie du Stage

## Chronologie du stage

Afin d'avoir une vision claire et structurée de l'évolution de mon travail durant ces dix semaines de stage retraçant les principales tâches effectuées et leur période de réalisation. Ce planning illustre les différentes phases de mon stage, depuis ma formation initiale jusqu'au développement de fonctionnalités concrètes au sein de l'ERP de SYG Software.

Parmi les étapes clés figurent :

- La formation sur Angular via la plateforme OpenClassrooms, pour renforcer mes bases en développement frontend.
- Le développement de la page d'édition d'étiquettes codes-barres
- La mise en place d'un module de gestion des promotions.
- Le développement de plusieurs pages fonctionnelles de l'application, intégrant des interfaces utilisateurs variées.
- Enfin, la fonctionnalité d'exportation des données vers un fichier Excel sur différentes pages du logiciel.

Ce découpage temporel met en lumière la montée en compétence progressive ainsi que la diversité des tâches techniques réalisées au fil du stage.

# Formation OpenClassroom

Au cours de ma formation sur **OpenClassrooms**<sup>[\*]</sup>, j'ai suivi trois modules dédiés à l'apprentissage du **framework**<sup>[\*]</sup> **Angular**, visant à renforcer mes compétences en développement web.

## 1. Module 1 « Débutez avec Angular »

Ce premier module m'a permis d'acquérir les bases essentielles d'Angular. J'ai appris à créer une application Angular, à construire des **composants** et à **structurer** un **document** avec les directives. J'ai également appris à **modifier** l'affichage des données et à gérer les **événements** utilisateur. Pour mettre en pratique ces compétences, j'ai développé un site web fictif nommé « **Snapface** », inspiré des réseaux sociaux, où j'ai intégré ces concepts fondamentaux.

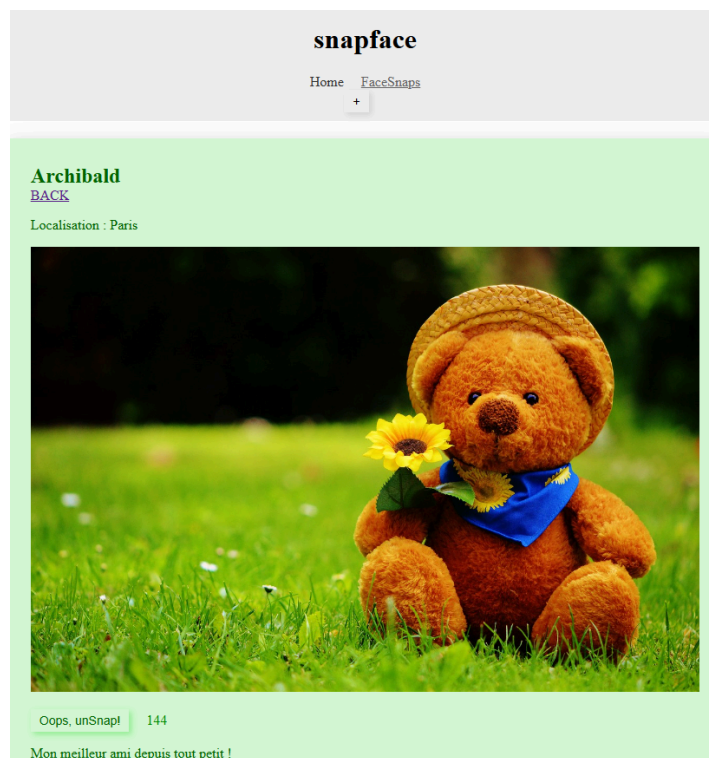


Figure 2 : Démonstration du site web « Snapface »

## 2. Module 2 « Complétez vos connaissances sur Angular »

Le deuxième module a approfondi mes connaissances en Angular, en explorant des fonctionnalités avancées telles que les **services** (des blocs de code réutilisables qui permettent de centraliser certaines fonctions, comme la récupération de données ou le partage d'informations entre différentes parties de l'application), le **routing** (qui permet de naviguer entre différentes pages de l'application sans recharger le site), les **formulaires réactifs** (des formulaires intelligents qui réagissent en temps réel aux actions de l'utilisateur, comme afficher une erreur si un champ est mal rempli), et la **gestion des observables** (un mécanisme qui permet de suivre en direct les changements de données, un peu comme une alerte automatique lorsqu'une information est mise à jour dans l'application). J'ai également appris à **sécuriser** les requêtes HTTP et à optimiser les performances de l'application. Pour appliquer ces nouvelles compétences, j'ai travaillé sur un projet appelé « **Awesome-Component** », un réseau social fictif, où j'ai intégré ces fonctionnalités avancées pour améliorer l'interactivité et la performance du site.

## 3. Module 3 « Perfectionnez-vous sur Angular »

Le dernier module a exploré des sujets avancés du framework Angular, tels que l'architecture, les animations, les **formulaires complexes** et la gestion d'état. Contrairement aux **formulaires réactifs** qui permettent déjà de gérer dynamiquement les champs d'un formulaire en fonction des actions de l'utilisateur, les **formulaires complexes** vont plus loin. Ils permettent de construire des structures de formulaire imbriquées, avec des validations personnalisées et des comportements conditionnels plus poussés, adaptés aux applications de grande envergure. J'ai appris à créer ce type de formulaires, à réagir aux changements de données en temps réel et à mettre en place une gestion d'état réactif (c'est-à-dire une façon d'organiser et de suivre l'évolution des données dans l'application pour qu'elle reste fluide et cohérente). Ces compétences ont été appliquées dans le projet « **Awesome-Component** », où j'ai intégré des animations et une gestion d'état réactif pour améliorer l'expérience utilisateur et la réactivité de l'application.

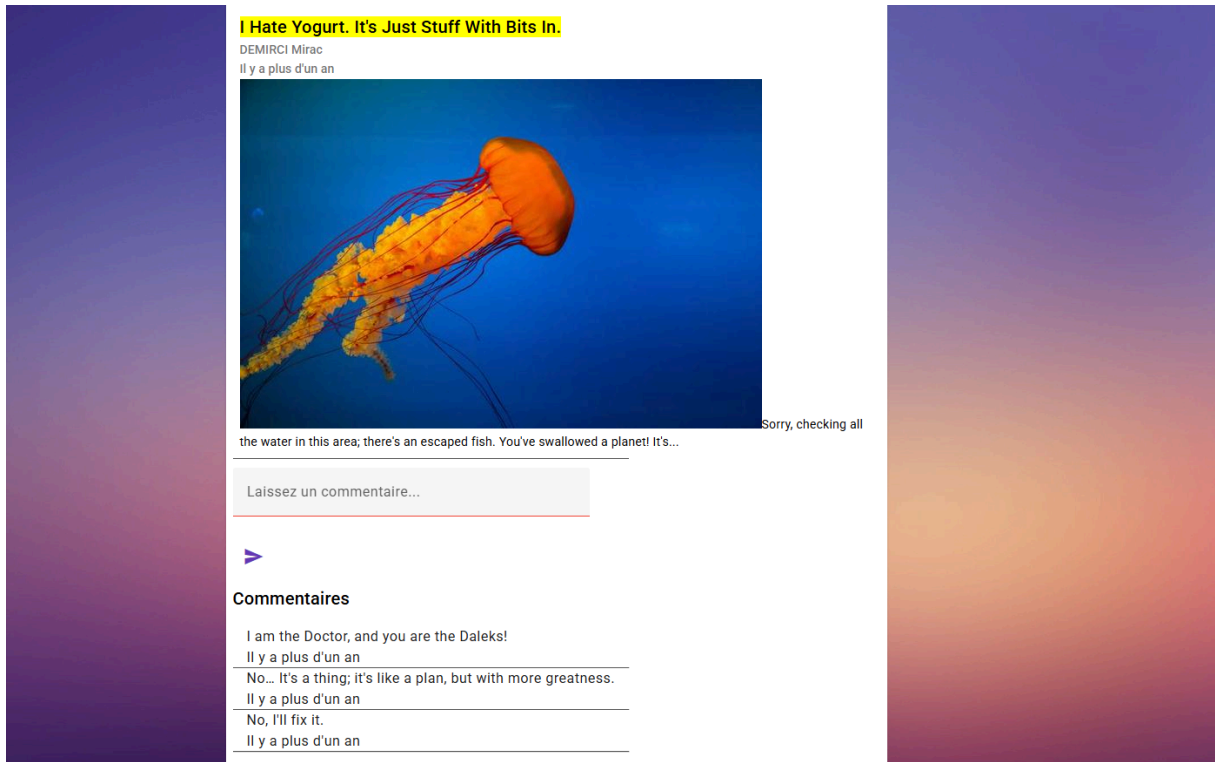


Figure 3 : Développement du site web du projet « Awesome-Component »

Ces formations m'ont permis de développer une compréhension approfondie d'Angular et de ses applications pratiques dans le développement de sites web dynamiques et interactifs.

# Développement de l'édition d'étiquettes code barre

## 1 Introduction au projet

Dans le cadre de mon stage, on m'a attribué le développement d'une fonctionnalité dédiée à **l'édition et à l'impression d'étiquettes de codes-barres**. Cette fonctionnalité permet aux utilisateurs de **rechercher** des codes-barres existants, de sélectionner ceux qu'ils souhaitent imprimer et de spécifier le nombre d'exemplaires nécessaires. Une fois ces paramètres définis, l'utilisateur peut procéder à **l'impression** des étiquettes correspondantes.

Cette solution vise à **simplifier et à automatiser le processus d'étiquetage**, en offrant une interface intuitive pour la gestion des codes-barres et en garantissant une impression précise et conforme aux besoins de l'utilisateur.

Pour mener à bien le développement de la fonctionnalité d'édition et d'impression d'étiquettes de codes-barres, j'ai consulté plusieurs **documentations techniques essentielles** :

### **Angular Material**<sup>[2]</sup> :

Cette documentation m'a permis, dans le cadre de ce projet d'intégrer des éléments tels que des boutons, des formulaires et des pop-up dans l'application

### **Bootstrap**<sup>[2]</sup> :

Bootstrap m'a permis d'intégrer des éléments de mise en page et de style, garantissant une présentation uniforme et responsive de l'application.

## 2. Développement du projet d'édition et d'impression d'étiquettes de codes-barres

### 2.1. Objectifs et spécifications du projet

Le projet d'édition et d'impression d'étiquettes de codes-barres a pour objectif de fournir une solution simple et efficace aux utilisateurs. En tant que fonctionnalité clé, il permet aux entreprises de **gérer leurs stocks de manière optimisée** en imprimant des étiquettes de codes-barres qui seront attachées à des produits ou équipements.

## Fonctionnalités principales

- **Recherche des codes-barres existants**

L'utilisateur peut rechercher et sélectionner des codes-barres déjà enregistrés dans le système via un moteur de recherche intégré, facilitant ainsi l'accès rapide aux articles concernés.

- **Sélection et paramétrage des impressions**

L'utilisateur peut spécifier la quantité d'étiquettes à imprimer pour chaque code-barres sélectionné. Une validation est effectuée pour garantir que les données saisies sont correctes.

- **Affichage en temps réel des données**

L'interface affiche les données de manière dynamique. Toute modification (ajout, suppression, mise à jour) est immédiatement visible, permettant un meilleur suivi et une détection rapide des anomalies éventuelles.

## Fonctionnalités optionnelles

- **Importation de fichiers CSV**

L'utilisateur a la possibilité d'importer un fichier **.csv** contenant une liste d'articles à imprimer. Le système traite ces données en vérifiant leur conformité, en gérant les doublons et en rejetant les lignes incorrectes avec une alerte détaillée.

- **Puits de saisie (importation assistée)**

Le puits de saisie permet également l'importation de données via fichier CSV, mais avec une validation en temps réel. Cette méthode offre une meilleure fiabilité avant insertion dans la base de données et optimise la fluidité du processus pour des saisies en masse.

### Processus détaillé :

1. **Recherche des codes-barres existants :**

- L'utilisateur peut rechercher et sélectionner des codes-barres déjà enregistrés dans le système.
- Mise en place d'un moteur de recherche.

2. **Importation de données via un fichier CSV (optionnelle) :**

- Les données sont insérées automatiquement dans le système.
  - Gestion des doublons pour éviter l'importation redondante.
  - Validation des données pour s'assurer qu'elles correspondent aux colonnes du tableau.
  - Notification en cas d'erreurs ou d'incohérences pour une correction rapide.
3. **Puit de saisie et affichage des données** (optionnelle) :
- Permet l'importation manuelle et via un fichier CSV.
  - Interface de saisie optimisée pour une insertion rapide et précise des données.
  - Affichage des données en temps réel pour un suivi instantané.
4. **Sélection et paramétrage des impressions** :
- L'utilisateur spécifie les quantités d'étiquettes à imprimer pour chaque article.
  - Impression selon des spécifications prédéfinies.

L'objectif final est de **simplifier le processus d'étiquetage**, de **réduire les erreurs humaines**, et d'**améliorer l'efficacité du personnel** dans la gestion des stocks, en fournissant une interface claire et intuitive qui permet de générer des étiquettes de manière fluide et rapide.

## 2.2. Défis rencontrés et solutions mises en place

Pendant le développement de cette fonctionnalité, plusieurs défis techniques sont apparus, notamment :

- **Adaptation à la base de données** : comprendre la structure et les relations entre les données pour savoir comment extraire et utiliser efficacement les informations essentielles aux codes-barres et aux articles.
- **Recherche de codes-barres et d'articles** : La recherche de codes-barres et d'articles n'a pas été évidente au début, notamment à cause de la complexité des requêtes. Swagger UI a été utilisé pour mieux comprendre comment les données étaient structurées et repérer celles qui étaient utiles, ce qui a permis de mettre en place un système de recherche plus clair et efficace..
- **Validation des paramètres avant l'impression** : Un autre défi a été la gestion des paramètres d'impression, notamment s'assurer que les valeurs entrées par l'utilisateur (quantité d'étiquettes) étaient **valides** avant de lancer l'impression. J'ai désactivé le

bouton d'impression lorsque les valeurs saisies étaient incorrectes afin d'éviter tout problème lors de l'impression

### 3. Optimisation du Code et Amélioration des Performances

Au début de mon développement de la fonctionnalité d'édition des étiquettes de codes-barres, j'ai réalisé que certaines parties de mon code n'étaient pas **optimisées** en termes de **performances et de lisibilité**. En effet, certaines solutions que j'avais implémentées étaient fonctionnelles, mais elles ne respectaient pas toujours les **bonnes pratiques en matière d'optimisation du code**. Par exemple, certaines requêtes à la base de données étaient réalisées de manière inefficace, entraînant des délais de réponse plus longs que prévu. De plus, j'avais parfois tendance à utiliser des **structures** de code qui n'étaient pas les plus appropriées, ce qui pouvait engendrer un code plus lourd et moins performant.

Mon maître de stage, Benoit Bourgeois, a pris le temps de revoir mon code et m'a fait part de remarques concernant ces points en soulignant les problèmes de **performance** liés à certaines de mes **implémentations**.

Grâce à ses retours, j'ai pu **réajuster mon code** et appliquer des méthodes plus performantes, telles que l'utilisation de la réduction du nombre d'appels redondants à l'**API**<sup>[44]</sup>, et la révision de certaines parties de mon **algorithme** de recherche pour rendre la fonctionnalité plus rapide et plus fluide.

Cette expérience m'a permis non seulement de corriger ces problèmes immédiats, mais aussi de **mieux comprendre l'importance de l'optimisation du code** pour améliorer la réactivité et l'évolutivité des applications, tout en réduisant les coûts en termes de ressources système. Ces corrections m'ont également aidé à acquérir une **meilleure discipline en matière de développement** et à prendre conscience des bonnes pratiques.

## 4. Maquette Attendu :

Voici la maquette initiale pour la page d'édition de code-barres :

Figure 4 : Maquette de la page d'édition étiquettes code barre

## 5. Résultat final

Le résultat final correspond aux attentes avec l'intégration fluide des fonctionnalités, la gestion optimisée des données et une interface conviviale pour l'utilisateur.

N°	Code Article	Désignation	NB impression
1	04L1C5	JTTTPX7C56	2
2	072CRW	EAPCQNHWQE	3
3	09M7AJ	EENVCT3CUB	4
4	06M2VA	M6CSW2CKSP	5
5	07YMGZ	DDYNFNFDH	6
6	0883LC	2WWMBJZ1QO	7
7	09FLVT	2SUZK0ZKBQ	8
8	0AL8B1	DBJBKEVFKN	9
9	0AZSSX	QHFTBWIIFO	10
10	0DU8AD	03UZRQLPOT	11

Figure 5 : Résultat de la page d'édition étiquettes code barre

## 6. Implémentation

### Contexte :

L'implémentation concerne l'ajout d'un code article pour l'impression. L'objectif était de vérifier si l'article existait déjà dans le tableau, dans ce cas je devais mettre à jour uniquement le nombre d'impressions sinon d'ajouter l'élément dans le tableau.

```
ajouterImpression(): void {
  const existingIndex = this.tableCodeBarre.findIndex(index: EditCodeBarreInterface => index.codearticle === this.genCodeTAB.idarticle);
  if (existingIndex === -1) {
    if (this.nbImpression >= 1) {
      this.tablecode = {
        codearticle: this.genCodeTAB.idarticle,
        designation: this.genCodeTAB.designation,
        nbimpression: this.nbImpression,
      };
      this.tableCodeBarre.push(this.tablecode);
    } else {
      console.log("Le nombre d'impressions doit être supérieur ou égal à 1");
    }
  } else {
    this.tableCodeBarre[existingIndex].nbimpression = this.nbImpression;
    console.log("Mis à jour:", this.tableCodeBarre[existingIndex]);
  }
  this.ListeCodeBarreSelectionne = new MatTableDataSource(this.tableCodeBarre);
  console.log("TEST : ", this.ListeCodeBarreSelectionne.data)
}
```

Figure 6 : Implémentation de l'ajout d'un article à l'impression

### Difficultés rencontrées

J'ai rencontré une difficulté pour identifier un article déjà présent dans le tableau, afin de mettre à jour son nombre d'impressions sans créer un doublon. Pour résoudre ce problème, j'ai exploré différentes solutions, comme l'utilisation de `map`<sup>[2]</sup> et la manipulation d'un tableau à l'aide de `Array`<sup>[\*]</sup>. Concrètement, j'ai tenté d'utiliser ces outils pour parcourir et comparer les éléments existants dans le tableau. L'idée était de détecter automatiquement si un article était déjà présent, puis d'augmenter simplement son compteur d'impressions. Malgré ces essais, les solutions ne s'adaptaient pas totalement à mon besoin spécifique, car elles ne permettaient pas toujours une mise à jour efficace des données existantes.

### Solution trouvée

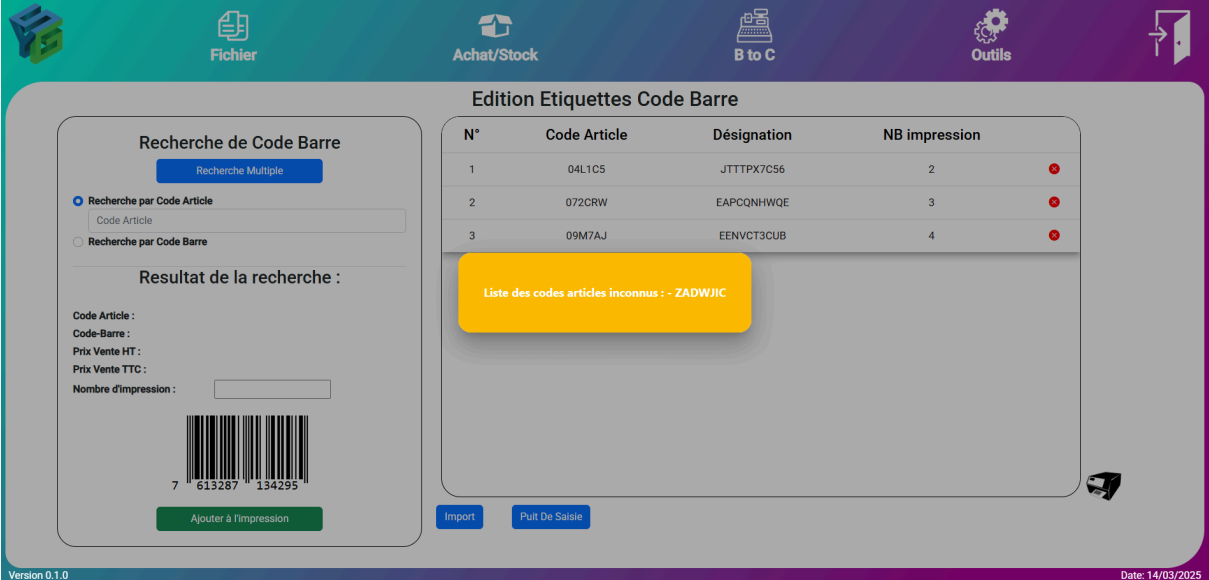
La fonction `findIndex` a été la clé pour résoudre ce problème. Elle m'a permis d'identifier l'élément et de modifier le nombre d'impressions sans avoir à recréer l'entrée. Cette approche a simplifié mon code et l'a rendu plus efficace et ainsi pu respecter les exigences du cahier des charges tout en améliorant la performance et la lisibilité du code.

## 7. Gestion des erreurs lors de l'importation de données

Lors de l'**importation** d'un fichier **CSV** contenant une liste de codes-barres et de désignations, j'ai mis en place un système de **gestion des erreurs** permettant de garantir l'intégrité des données insérées dans le tableau.

### 7.1. Identification des données non conformes

Lorsque certaines lignes du fichier **CSV** contiennent des données **incorrectes** ou des désignations **invalides**, une **alerte** est générée pour avertir l'utilisateur. Par exemple, si un **code article** n'existe pas dans la base ou si la **désignation** ne correspond pas aux standards définis, une **alerte** apparaît sous la forme suivante :



The screenshot shows a software interface with a navigation bar at the top containing icons for 'Fichier', 'Achat/Stock', 'B to C', and 'Outils'. The main content area is titled 'Edition Etiquettes Code Barre'. On the left, there is a 'Recherche de Code Barre' section with a 'Recherche Multiple' button and radio buttons for 'Recherche par Code Article' (selected) and 'Recherche par Code Barre'. Below this is a 'Resultat de la recherche' section with fields for 'Code Article', 'Code-Barre', 'Prix Vente HT', 'Prix Vente TTC', and 'Nombre d'impression', along with a barcode and an 'Ajouter à l'impression' button. On the right, there is a table with the following data:

N°	Code Article	Désignation	NB impression
1	04L1CS	JTTTPX7C56	2
2	072CRW	EAPCQHWQE	3
3	09M7AJ	EENVCT3CUB	4

Below the table, a yellow alert box displays the message: 'Liste des codes articles inconnus : - ZADWJIC'. At the bottom of the interface, there are 'Import' and 'Puit De Saisie' buttons, and a footer with 'Version 0.1.0' and 'Date: 14/03/2025'.

Figure 7 : Alerte d'erreur d'importation

Cela permet à l'utilisateur de visualiser précisément les lignes rejetées et de **corriger** manuellement les erreurs dans le fichier **CSV** avant de réessayer l'importation.

### 7.2. Gestion des doublons et des données manquantes

- Si un code article est déjà présent dans le tableau, le système **détecte** automatiquement le **doublon** et met à jour uniquement le nombre d'impressions.
- Si la colonne du nombre d'impressions est vide ou contient une valeur inférieure à 1 dans le fichier **.csv**, le nombre d'impression est automatiquement mis à 1.

## 8. Gestion des erreurs sur le nombre d'impressions

Lors de l'ajout manuel d'un code article pour l'impression, j'ai également intégré une validation avec des conditions strictes sur la saisie du nombre d'impressions.

### Nombre d'impressions inférieur à 1 ou champ vide :

Si l'utilisateur tente de saisir une valeur inférieure à 1 le bouton d'ajout apparaîtra désactivé et donc l'ajout ne sera pas possible.

### Conclusion : Développement de l'édition d'étiquettes code-barres

Cette réalisation m'a permis d'acquérir des compétences techniques et pratiques essentielles dans le domaine du développement web et de l'intégration d'API.

### Compétences techniques développées :

1. **Maîtrise d'Angular Material et Bootstrap** : En travaillant avec Angular Material et Bootstrap, j'ai appris à créer une interface utilisateur fluide, moderne et responsive. J'ai pu intégrer des **composants**<sup>[4]</sup> interactifs et des mises en page efficaces pour améliorer l'expérience utilisateur.
2. **Gestion de la base de données et optimisation du code** : Des compétences renforcées en gestion de base de données ont été acquises, en particulier dans la recherche et la manipulation des données liées aux codes-barres. L'optimisation des requêtes et la réduction des appels redondants à l'API, grâce aux retours du maître de stage, ont permis d'améliorer significativement les performances de l'application.
3. **Gestion des erreurs et validation des données** : Les compétences en détection et prévention des erreurs ont été renforcées, grâce à l'ajout de validations et de contrôles rigoureux pour garantir la fiabilité des données saisies.

### Évolution personnelle et ressentis

Au début du projet, je craignais d'avoir des difficultés à développer une fonctionnalité aussi complexe. Mais en programmant et en faisant mes **recherches** dans des **documentations**, j'ai gagné en **compétences**. Les défis techniques rencontrés, tels que l'utilisation de la fonction **findIndex** pour mettre à jour le nombre d'impressions sans dupliquer les articles m'ont permis de renforcer ma logique de programmation et ma capacité à résoudre des problèmes.

# Développement de la gestion des promotions

## Introduction au projet

Après le développement des étiquettes de code-barres, j'ai été chargé du développement d'une nouvelle interface de gestion des promotions. Cette fonctionnalité a pour objectif d'optimiser la gestion des périodes promotionnelles, comme les rentrées scolaires, les soldes d'été, ou encore les offres spéciales ponctuelles.

Grâce à cette nouvelle interface, les utilisateurs peuvent :

- Créer et configurer différentes promotions, avec des dates et types de réductions variés (remise fixe, pourcentage, 1+1, 2+1...).
- Gérer facilement les articles concernés par chaque promotion grâce à un système de recherche avancé ou via l'importation d'un fichier CSV.
- Appliquer automatiquement des réductions sous forme de montant fixe ou de pourcentage, et également ajouter un prix d'achat promotionnel pour chaque article.
- Modifier les promotions existantes pour ajuster leurs paramètres en fonction des besoins commerciaux.

N°	Code	Désignation	Prix Achat HT	Prix Vente HT	Prix Remise HT	Prix Remise TTC	Remise %
1	09M7AJ	EENVCT3CUB	40.29197	325.65308	275,65	294,52	15,35
2	0AL8B1	DBJBKEVFKN	194.89107	403.3606	353,36	377,54	12,40
3	0KFAOG	1R2OSMWSAX	229.40225	21.044525	-28,96	-30,94	237,59
4	0883LC	2WWMBJZ1QO	360.79218	744.66595	694,67	742,20	6,71
5	06MZVA	M6CSW2CKSP	151.40851	534.6908	484,69	540,01	9,35
6	09FLVT	2SUZK0ZKBQ	361.8842	228.15073	178,15	190,34	21,92
7	0AZSSX	QHFTBWIIFO	264.1169	986.30707	936,31	1 043,16	5,07
8	0DUBAD	03UZRLPOT	225.28279	244.03702	194,04	208,92	20,49
9	0X65B5	0VGKYHEZBV	172.83636	603.09375	553,09	616,22	8,29
10	0EJYJH	8NULXQKH0I	208.53069	781.6833	731,68	781,75	6,40

Figure 8 : Interface principale de gestion des promotions

Cette page permet ainsi une gestion fluide, intuitive et centralisée des promotions, réduisant le temps et les efforts nécessaires pour appliquer des offres commerciales aux articles.

## **Objectifs et fonctionnalités principales**

### **1. Affichage et gestion des promotions existantes**

Dès l'ouverture de la page, l'utilisateur accède à un tableau listant toutes les promotions existantes. Chaque promotion est caractérisée par plusieurs informations essentielles :

- Nom de la promotion (ex : Soldes d'été, Rentrée scolaire 2024, Black Friday...).
- Type de promotion (remise fixe, remise en pourcentage, 1+1, 2+1...).
- Dates de début et de fin, afin de déterminer la durée d'application de la promotion.

Lorsqu'un utilisateur clique sur une promotion spécifique, un second tableau s'affiche sur la même page. Ce tableau contient la liste des articles concernés par cette promotion, avec leurs détails tels que :

- Le nom et la référence de l'article.
- La remise appliquée (montant ou pourcentage, selon le type de promotion).
- Le prix avant remise et après remise (HT et TTC).
- Le prix d'achat fournisseur s'il a été saisi.

Les colonnes affichées dans le tableau dépendent du **type de promotion** sélectionné : par exemple, les remises ne sont pas affichées pour les promotions de type 1+1 ou 2+1.

### **2. Création d'une nouvelle promotion**

L'utilisateur peut ajouter une nouvelle promotion grâce à un bouton dédié.

En cliquant dessus, une fenêtre pop-up s'ouvre et permet de renseigner les informations suivantes :

- Nom de la promotion (ex : Soldes d'été, Remise spéciale rentrée...).
- Type de promotion (ex : réduction fixe de 5€, réduction de 20%, 2+1...).
- Date de début et de fin pour définir la période de validité.
- Articles concernés par cette promotion via une interface d'ajout simplifiée.

⚠ Une fois le type de promotion sélectionné à la création, celui-ci **ne peut plus être modifié** par la suite pour garantir la cohérence des données.

### 3. Ajout des articles concernés

Une des parties essentielles du projet consiste à rendre l'ajout des articles rapide et intuitif. Trois méthodes sont mises à disposition de l'utilisateur :

#### 3.1. Recherche manuelle

L'utilisateur peut saisir directement la référence d'un article via un champ de recherche.

- Une erreur s'affiche lorsque la référence de l'article n'a pas pu être retrouvée ou n'existe pas.
- Si l'article est déjà présent dans la promotion, il ne sera pas ajouté une seconde fois et une alerte est affichée.
- Un effet visuel (fond vert) permet de repérer les articles ajoutés récemment.

The screenshot shows the 'Gestion des promotions' interface. At the top, there are fields for 'Solde d'été' (Le nom est obligatoire), 'Type de la promotion\*' (Listing), 'Début de la promotion\*' (01/04/2025), and 'Fin de la promotion\*' (10/04/2025). Below these are buttons for 'Recherche d'Article', 'Ajouter', and 'Recherche Multiple'. The main area contains a table of articles with columns for 'Code', 'Désignation', 'Prix Achat HT', 'Prix Achat Promo', and 'Prix Vente HT'. A yellow alert box is overlaid on the table, stating 'Le code article : 0883LC est déjà inséré'. The table shows several rows, with the last two rows (0AZSSX and 0DU8AD) highlighted in green. To the right of the table, there is a 'Remise Appliquée' section with radio buttons for 'Remise en un Montant' (set to 50) and 'Remise en Pourcentage' (set to 0). At the bottom, there are buttons for 'Fermer', 'Enregistrer les modifications', 'Supprimer', 'Reinitialiser Promo Fournisseur', and 'Import'.

	Code	Désignation	Prix Achat HT	Prix Achat Promo	Prix Vente HT
<input type="checkbox"/>	0883LC	2WWMBJZ1QO			
<input type="checkbox"/>	06MZVA	M6CSW2CKSP			
<input type="checkbox"/>	09FLVT	2SUZK0ZKBQ	361.8842	0	228.15073
<input type="checkbox"/>	0AZSSX	QHFTBWIIFO	264.1169	0	986.30707
<input type="checkbox"/>	0DU8AD	03UZRQLPOT	225.28279	0	244.03702

Figure 9 : Alerte lors d'un ajout d'article déjà inséré

#### 3.2. Recherche multiple avec un second pop-up

Pour simplifier l'ajout d'un grand nombre d'articles, un bouton permet d'ouvrir une fenêtre listant tous les articles disponibles.

- L'utilisateur peut ajouter un article d'un simple clic gauche, sans avoir à les rechercher un par un.
- Il peut filtrer la recherche grâce aux différents champs de recherche proposés (désignation, code article, fournisseur, etc.).
- L'article est ajouté automatiquement à la liste, en évitant les doublons.

### 3.3. Import de fichier CSV

Pour les promotions massives, l'utilisateur peut importer un fichier contenant plusieurs références.

- Les références invalides ou déjà présentes sont ignorées et un message d'alerte informe l'utilisateur.
- Les références valides sont automatiquement ajoutées à la promotion.

### 4. Définition des réductions et remises

Une fois les articles ajoutés à la promotion, l'utilisateur peut définir la réduction à appliquer. Deux options sont disponibles :

- **Montant fixe** (ex : 5€, 10€...).
- **Réduction en pourcentage** (ex : 10%, 25%...).

Cette remise est automatiquement appliquée à tous les articles sélectionnés et sera prise en compte lors des ventes pendant la période promotionnelle.

En plus de la remise, l'utilisateur peut également définir un **prix d'achat promotionnel** spécifique à chaque article, utile pour les calculs de marges ou suivi comptable.

### 5. Modification et mise à jour d'une promotion existante

L'application permet également de modifier facilement une promotion déjà créée.

En sélectionnant une promotion dans le tableau principal, l'utilisateur peut cliquer sur un bouton « Modifier », qui ouvre une fenêtre pop-up similaire à celle de création. Il peut alors modifier tous les paramètres de la promotion, à l'exception du type. Cela inclut notamment :

- Le nom de la promotion.
- Les dates de validité.

- Ajouter ou supprimer des articles concernés.
- Modifier la réduction appliquée aux articles (remise ou pourcentage).
- Ajouter/Modifier le prix d'achat du fournisseur.

Une fois les modifications validées, la promotion est mise à jour en temps réel et les changements sont immédiatement visibles pour les utilisateurs.

**Gestion des promotions :**

Solde d'été | Type de la promotion\* Listing | Début de la promotion\* 01/04/2025 | Fin de la promotion\* 10/04/2025

Le nom est obligatoire | Recherche d'Article | Ajouter | Recherche Multiple

ID	Code	Désignation	Prix Achat HT	Prix Achat Promo	Prix Vente HT
2115	0883LC	2WWMBJZ1Q0	360.79218	5	744.66595
2116	06MZVA	M6CSW2CKSP	151.40851	5	534.6908
	09FLVT	2SUZK0ZKBQ	361.8842	0	228.15073
	0AZSSX	QHFTBWIF0	264.1169	0	986.30707
	0DUBAD	03UZRQLPOT	225.28279	0	244.03702

**Remise Appliquée :**

Remise en un Montant : 50

Remise en Pourcentage : 0

Remise Prix Fournisseur : 5

Modifier la promo Fournisseur

Fermer | Enregistrer les modifications | Supprimer | Reinitialiser Promo Fournisseur | Import

Version 0.1.0 | Date: 04/04/2025

Figure 10 : Modification d'une promotion

Les articles affichés en **vert** correspondent à ceux qui viennent d'être ajoutés, ce qui permet de les distinguer facilement des articles déjà présents dans la promotion.

Les articles sélectionnés peuvent ensuite :

- Se voir appliquer un **prix d'achat promotionnel fournisseur** (ou prix achat promo),
- Être **supprimés** de la promotion,
- Ou bien une **réinitialisation du prix fournisseur**, c'est-à-dire remettre la remise à zéro.

### Gestion de la cohérence des données

Pour garantir l'intégrité et la cohérence des données saisies, plusieurs règles de validation ont été mises en place lors de la création et de la modification d'une promotion. Ces contrôles

permettent d'éviter des erreurs susceptibles d'impacter l'application des remises et d'assurer une expérience utilisateur fluide et fiable.

### 1. Vérification des champs obligatoires

Une promotion doit obligatoirement contenir un **nom** et un **type de promotion**.

- Le champ « Nom de la promotion » est requis. Sans un nom, il est impossible de l'identifier correctement dans la liste des promotions.
- Le champ « Type de promotion » est obligatoire. Il définit la nature de la promotion (ex : 2+1, 1+1, Listing).  
Si ces informations ne sont pas renseignées, un message d'erreur est affiché et le bouton d'ajout/modification reste désactivé.

### 2. Validation des dates de la promotion

Lorsqu'un utilisateur définit une nouvelle promotion, il doit renseigner une date de début et une date de fin.

- La date de début ne peut pas être postérieure à la date de fin.
- Si une date invalide est saisie, un message d'erreur s'affiche clairement pour informer l'utilisateur.
- Tant que la correction n'est pas effectuée, le bouton « Enregistrer les données » reste désactivé, empêchant l'enregistrement de données erronées.

### 3. Contrôle du type de remise appliquée

L'utilisateur peut appliquer **soit** une remise en montant, **soit** en pourcentage.

- Si une remise en montant est saisie, le champ pourcentage est désactivé automatiquement, et inversement.
- Il est impossible d'appliquer les deux types de remise en même temps.

### 4. Vérification de la présence d'articles dans la promotion

Une promotion doit obligatoirement contenir **au moins un article** concerné.

- Une promotion vide n'a aucun effet, et son enregistrement est donc interdit.

- Le bouton d'ajout/modification reste désactivé tant qu'aucun article n'a été ajouté à la liste.

### **Difficulté rencontrée – Intégration avec le backend et cohérence des données**

L'un des défis majeurs de ce projet a été la gestion de la cohérence des données lors de l'échange avec le **backend**<sup>[\*]</sup>, notamment pendant l'ajout ou la modification d'une promotion.

En effet, chaque promotion pouvait contenir plusieurs articles, chacun ayant un **prix d'achat promotionnel fournisseur spécifique**. Cela signifie que le même article pouvait avoir un prix d'achat différent d'une promotion à l'autre. Il fallait éviter toute confusion ou perte d'information entre les différentes promotions contenant le même article.

Voici les principales difficultés rencontrées :

- **Synchronisation en temps réel** : Lorsqu'un article était ajouté ou modifié, les données devaient être instantanément renvoyées au backend et reflétées dans l'interface, sans délai ni incohérence.
- **Validation conditionnelle des actions** : Il a fallu être très vigilant sur la logique d'enregistrement. Par exemple :
  - Par exemple : Si un **prix d'achat fournisseur (prix d'achat promotionnel)** était modifié, mais sans enregistrement final, **la valeur initiale devait être conservée**.

### **Compétences développées**

Ce projet de gestion des promotions m'a permis de renforcer et d'approfondir plusieurs compétences techniques et professionnelles, essentielles dans le cadre d'un développement applicatif complet.

#### **1. Maîtrise de la logique front-end / back-end**

J'ai consolidé ma compréhension de l'architecture client-serveur, en veillant à ce que toutes les actions effectuées côté interface (ajout, suppression, modification) soient **parfaitement synchronisées avec les opérations backend**. Cela m'a appris à mieux gérer :

- Les appels API asynchrones,
- Le traitement des réponses serveur,

- Et la gestion des états complexes côté interface.

## 2. Manipulation de données complexes

J'ai appris à manipuler des objets de données dynamiques contenant des informations imbriquées (articles, remises, prix d'achat fournisseurs), tout en assurant leur **intégrité et leur cohérence**. Cela m'a poussé à structurer correctement les données avant envoi au **backend**, et à gérer les retours de manière fiable.

## 3. Validation des formulaires et retours utilisateurs

J'ai développé une meilleure maîtrise des **mécanismes de validation conditionnelle**, afin de garantir que l'utilisateur ne puisse pas enregistrer une promotion incomplète ou incohérente. Ce travail m'a permis de mettre en place des règles claires, des messages d'erreurs pertinents et une logique d'activation/désactivation intelligente des boutons.

## 4. Expérience utilisateur (UX)

En pensant comme un utilisateur final, j'ai amélioré la **fluidité de l'expérience** :

- Ajout rapide d'articles,
- Filtres de recherche multi-critères,
- Feedbacks visuels (messages, couleurs, champs désactivés),
- Mise à jour en temps réel des tableaux après chaque action.

## 5. Gestion des cas particuliers

Ce projet m'a confronté à des cas réels complexes, comme :

- Le traitement d'un même article dans plusieurs promotions avec des contextes différents,
- L'annulation partielle ou complète d'une action avant validation,

## 6. Communication avec l'équipe et compréhension métier

Enfin, ce projet m'a aussi permis de mieux comprendre les besoins du client (logique commerciale autour des promotions), et d'échanger efficacement avec les membres de l'équipe pour résoudre ensemble les problématiques techniques rencontrées.

## Explication du Code pour l'Ajout d'une Promotion

Le programme présenté permet d'ajouter une nouvelle promotion pour des articles. Il prend en compte plusieurs éléments, comme les dates de début et de fin de la promotion, le type de remise, ainsi que les articles concernés. Voici ce que chaque partie du code fait :

```
async ajouterPromo(): Promise<void> {
  if (this.promo.value.remisePourcentage == null) {
    this.promo.value.remisePourcentage = 0
  }
  if (this.promo.value.remiseMontant == null) {
    this.promo.value.remiseMontant = 0
  }

  for (let i = 0; i < this.promoCodeTAB.length; i++) {
    this.allCodevalue.push(this.promoCodeTAB[i].idarticle)
  }
  this.promoArticle = {
    libelle: this.promo.value.nom,
    dateDebutPromotion: new Date(this.promo.value.dateDebut),
    dateFinPromotion: new Date(this.promo.value.dateFin),
    typePromo: this.promo.value.type,
    promotionMontant: this.promo.value.remiseMontant,
    promotionPourcentage: this.promo.value.remisePourcentage,
    articles: this.allCodevalue
  }
  this.promoArticle.dateDebutPromotion.setHours(12, 0, 0, 0)
  this.promoArticle.dateFinPromotion.setHours(12, 0, 0, 0)

  this.promoService.addPromotionArticle(this.promoArticle).subscribe({
    next: data: number => {
      this.promotion = {
        idpromotion: data,
        libelle: this.promo.value.nom,
        dateDebutPromotion: new Date(this.promo.value.dateDebut),
        dateFinPromotion: new Date(this.promo.value.dateFin),
        typePromo: this.promoArticle.typePromo,
        promotionMontant: this.promo.value.remiseMontant,
        promotionPourcentage: this.promo.value.remisePourcentage,
        articles: this.promoCodeTAB
      }

      this.promotion.dateDebutPromotion.setHours(12, 0, 0, 0)
      this.promotion.dateFinPromotion.setHours(12, 0, 0, 0)

      this.notificationService.notify('La promotion ' + this.promoArticle.libelle + " a été ajoutée avec succès", 'create')
      this.dialogResult = [this.promotion, true]
      this.closeDialog()
    },
    error: err: any => {
      this.dialogResult = [this.promotion, false]
      alert(err.message);
    }
  })
}
```

Figure 11 : Code permettant l'ajout d'une promotion

### 1. Vérification des champs de remise :

Avant toute chose, le programme vérifie si les champs de remise en pourcentage (remisePourcentage) ou en montant (remiseMontant) sont vides (ou égaux à null). Si c'est le cas, ils sont automatiquement remplacés par 0 pour éviter d'envoyer des valeurs non valides à la base de données.

## **2. Récupération des articles concernés :**

Le programme parcourt le tableau promoCodeTAB qui contient tous les articles sélectionnés pour la promotion. Il en extrait les identifiants (idarticle) et les stocke dans un tableau allCodevalue, utilisé ensuite pour associer les articles à la promotion.

## **3. Construction de l'objet promotion :**

Une fois les données prêtes, un objet promoArticle est créé. Il contient les informations nécessaires : nom de la promotion, dates de début et de fin, type de remise, valeur de la remise et les articles concernés. Cela permet de structurer les données avant leur enregistrement.

## **4. Normalisation des heures :**

Pour éviter les problèmes liés aux fuseaux horaires ou aux décalages horaires (comme l'heure d'été), les heures de début et de fin sont fixées à 12h. Cela garantit une gestion plus fiable des dates de promotion dans l'application.

## **5. Enregistrement dans la base de données :**

La méthode promoService.addPromotionArticle() est appelée pour envoyer les données vers le serveur et enregistrer la promotion en base. Cette opération est asynchrone et gérée via subscribe().

## **6. Traitement du retour serveur :**

Si l'ajout est un succès (next), les données renvoyées par le serveur sont récupérées pour reconstruire un objet promotion complet. Ce dernier contient également les dates normalisées, les informations sur la remise et les articles. Le tout est ensuite stocké dans dialogResult.

## **7. Notification et fermeture :**

Une notification est affichée pour informer l'utilisateur que la promotion a bien été ajoutée. Enfin, la fenêtre de dialogue est fermée grâce à la méthode closeDialog().

## **8. Gestion des erreurs :**

Si une erreur survient lors de l'ajout (dans la partie error), un message d'erreur est affiché à l'utilisateur, et le résultat est enregistré comme un échec dans dialogResult.

## **Conclusion du développement de la gestion des promotions**

La mise en place de cette fonctionnalité de gestion des promotions m'a permis de concevoir une interface complète, fiable et intuitive, répondant aux besoins concrets des utilisateurs. Ce projet m'a confronté à des problématiques techniques réelles, notamment la validation des données, la gestion des interactions front-end / back-end, ainsi que l'amélioration de l'expérience utilisateur. Grâce à cette interface, la création et le suivi des offres promotionnelles sont désormais centralisés et rapides, ce qui représente un véritable gain de temps pour les utilisateurs tout en assurant la cohérence des données en base.

## **Développement de nouvelles pages : montée en compétence et gain de productivité**

Grâce aux deux premières pages que j'ai développées au sein de l'ERP, j'ai considérablement progressé dans ma manière de coder. Cette expérience m'a permis de mieux comprendre le fonctionnement du framework Angular, de gagner en fluidité dans l'utilisation des requêtes API, mais surtout d'apprendre à manipuler Bootstrap de manière bien plus efficace. Là où il me fallait auparavant deux jours pour concevoir une interface complète, je suis désormais capable de développer une page similaire en une à deux heures, tout en garantissant une meilleure qualité de code et une plus grande clarté visuelle.

Ce progrès s'est également ressenti dans ma capacité à structurer des formulaires plus rapidement et plus proprement, en tenant compte à la fois de l'expérience utilisateur et des contraintes fonctionnelles. Ces compétences techniques nouvellement acquises m'ont permis de développer plusieurs pages de manière plus autonome et confiante. Je sais désormais aborder une nouvelle interface avec sérénité, en ayant une vision plus claire des étapes de développement et des outils à utiliser.

### **Page de gestion des mouvements de caisse et de stock :**

La première page développée dans ce cadre a été celle de la gestion des mouvements de caisse et de stock. Elle permet aux utilisateurs d'ajouter de nouveaux mouvements, qui seront ensuite utilisés dans d'autres modules de l'ERP pour assurer le suivi des flux de trésorerie et de marchandises.

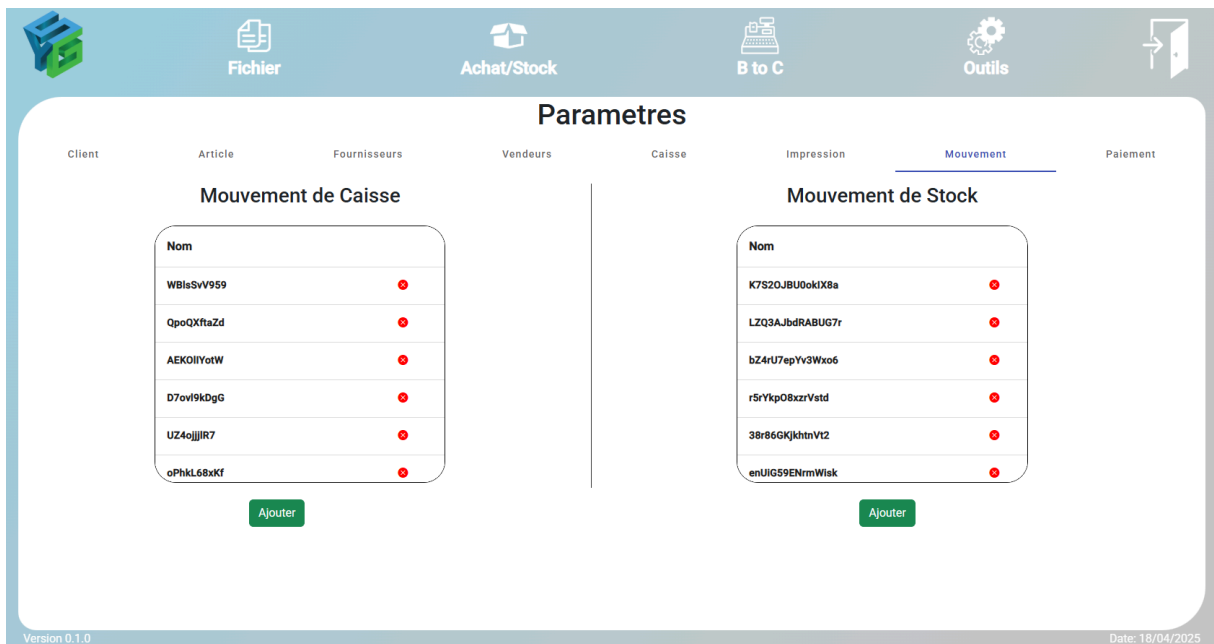


Figure 12 : Page de la gestion des mouvements

Cette page répond à un besoin précis : offrir un moyen de renseigner manuellement un mouvement, qu'il s'agisse d'un vol, d'une casse, d'une utilisation interne, ou d'un ajustement ponctuel du stock. L'utilisateur peut y définir un nom de mouvement, qui sera ensuite exploité dans différents écrans tels que :

- la page des fonds et mouvements de caisse ;
- le journal de caisse journalier ;

L'apparition de ces données dans les autres modules est gérée par la partie backend. Mon rôle a été de créer l'interface d'ajout des mouvements, en utilisant Angular et Bootstrap, tout en assurant l'envoi correct des données via une requête API.

### Page de gestion de la carte de fidélité :

La seconde page que j'ai développée dans cette nouvelle phase du projet est dédiée à la **gestion des cartes de fidélité client**. Cette interface permet d'assurer à la fois un **suivi précis des points de fidélité** et une **personnalisation des règles de fonctionnement** du programme.

J'ai commencé par créer une **page de recherche** permettant d'identifier un client à partir de différents critères comme :

- son **ID**, **nom**, **prénom**, ou **adresse complète** (rue, code postal, ville) ;

- son **numéro de téléphone** ou son **adresse mail** ;
- ou encore son **nombre de points de fidélité**.

Les résultats sont affichés dans un **tableau clair et interactif**, facilitant la consultation et la sélection d'un client.

Figure 13 : Page de recherche d'un client

Sur cette même page, l'utilisateur peut aussi **modifier les paramètres du programme de fidélité**, notamment :

- le **nombre de points gagnés** pour un montant dépensé (ex. : 1 point pour 10 €) ;
- la **valeur monétaire d'un point** (ex. : 1 point = 0,05 €) ;
- le **seuil de déblocage** (ex. : la fidélité devient utilisable à partir de 500 points, soit 25 € accumulés).

J'ai également mis en place la gestion des **règles d'échéance** de la fidélité. Trois modes sont disponibles :

- un **renouvellement automatique à partir de la date d'achat**, avec une saisie du **nombre de jours** de validité ;
- une **expiration après une période sans achat**, également paramétrée en **nombre de jours** ;

- ou une **remise à zéro à une date fixe**, avec une saisie de la date au format **JJ/MM/AAAA**.

Lorsqu'on clique sur un client dans le tableau, on est redirigé vers une **page de détail** affichant toutes les informations relatives à sa carte de fidélité. Cette page permet :

- de **consulter les données du client** et son nombre de points actuels ;
- **d'ajouter ou retirer un nombre précis de points** ;
- ou encore de **réinitialiser son solde à zéro** si nécessaire.

Cette fonctionnalité n'a pas pu être finalisée entièrement, car certaines **requêtes côté backend** n'étaient pas encore disponibles. J'ai néanmoins intégré toutes les parties réalisables à partir des **requêtes déjà existantes**, tout en préparant le reste du code pour faciliter la suite du développement.

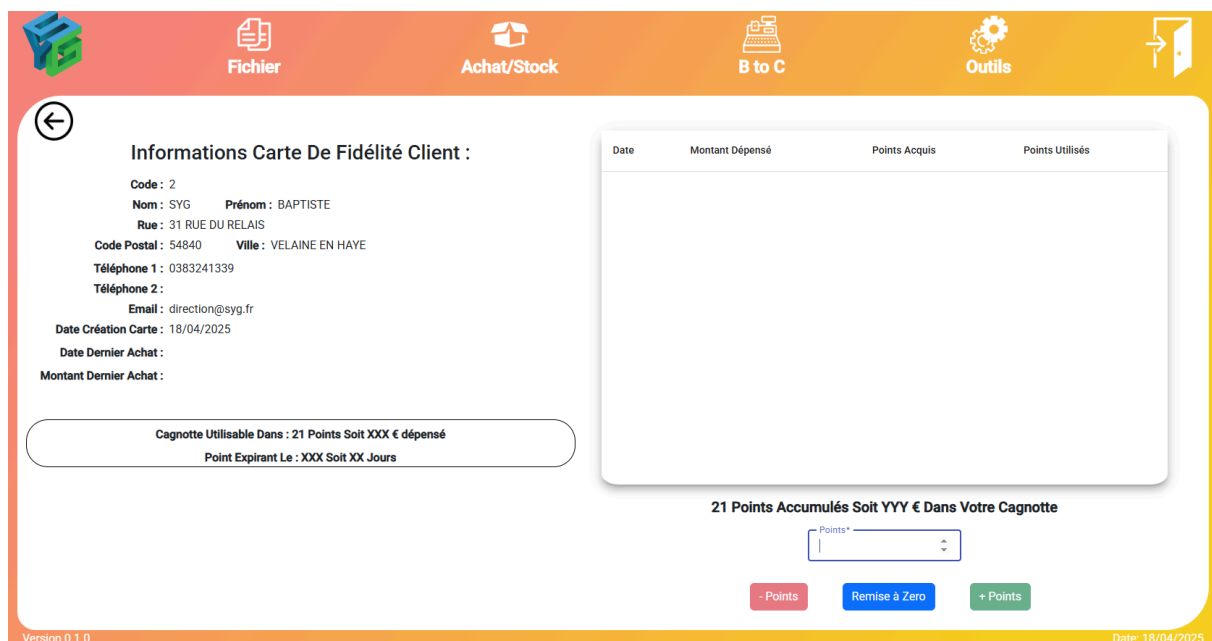
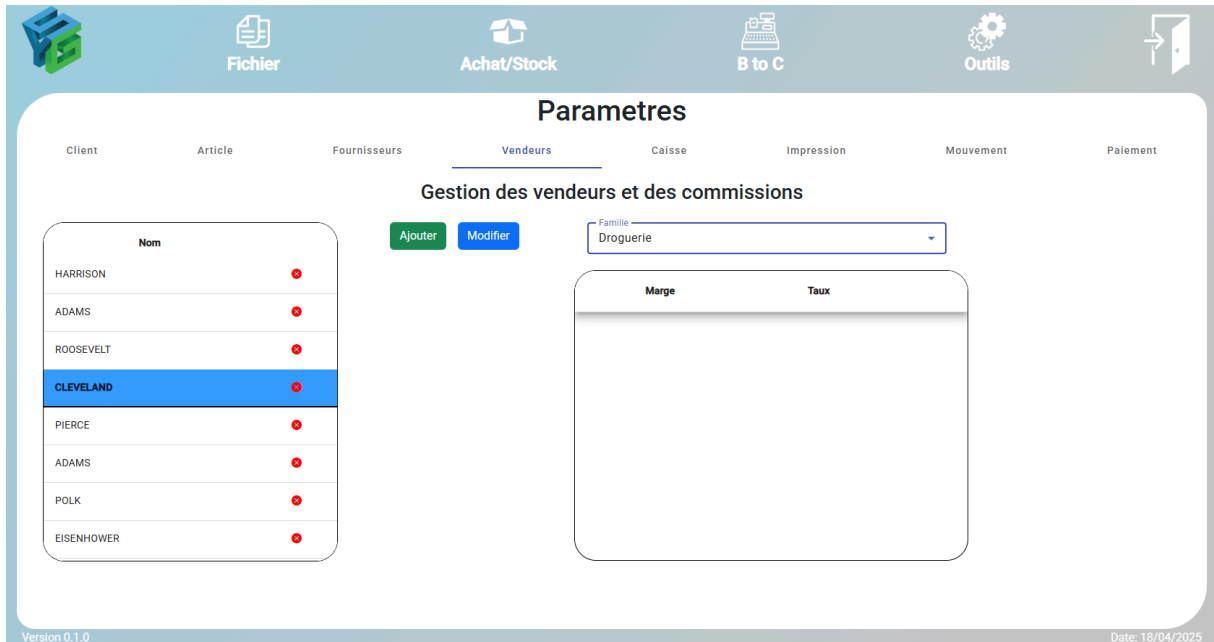


Figure 14 : Page d'information de Carte de Fidélité d'un client

### Page de gestion des vendeurs et des commissions :

Cette page permet **d'ajouter, de modifier ou de supprimer un vendeur**, ainsi que de **consulter les différentes familles de produits** comme *Ordinateurs*, *Matériels*, etc. Ces fonctionnalités facilitent la gestion quotidienne des équipes commerciales.

La partie permettant d'enregistrer les **marges** et les **taux de commission** n'était pas encore développée côté technique (back-end). Le travail s'est donc concentré sur **la récupération des vendeurs et des familles**, ainsi que sur **les actions liées à leur gestion** (ajout, modification, suppression).



*Figure 15 :Page de la Gestion des Vendeurs et des Commissions*

Pour aider les utilisateurs à mieux se repérer, **l'interface visuelle change selon l'action** : un encadré vert s'affiche lorsqu'on ajoute un vendeur, et un encadré bleu lors d'une modification. Ce système de couleurs permet de **différencier clairement les étapes**, et rend l'interface plus intuitive.

Même si cette partie ne comportait pas de logique très complexe, elle m'a permis de **mieux structurer une interface claire** et de **renforcer ma capacité à adapter l'affichage selon le contexte**, ce qui est essentiel dans le développement d'outils accessibles et faciles à prendre en main.

## Exportation des données en fichier Excel

Au cours de mon stage chez SYG Software, l'une des tâches qui m'a été confiée a consisté à intégrer une **fonctionnalité d'exportation des données affichées dans les tableaux** vers des fichiers Excel. Cette fonctionnalité est particulièrement utile pour les utilisateurs, car elle leur permet de récupérer facilement les informations affichées dans l'ERP, afin de les analyser, les partager ou les archiver à l'aide d'outils bureautiques comme Microsoft Excel.

### Contexte :

Cette mission s'inscrit dans un besoin de **modernisation et d'amélioration de l'expérience utilisateur** au sein de l'interface de l'ERP. L'exportation vers Excel devait être ajoutée sur plusieurs pages existantes, permettant aux utilisateurs de récupérer instantanément les données visibles à l'écran, dans un format structuré et réutilisable.

Cependant, ces pages **n'avaient pas été développées par moi**, ce qui a nécessité un temps d'adaptation pour bien comprendre la logique et l'organisation du code existant.

L'export s'appuyait sur un **service déjà développé par un autre stagiaire**, que j'ai dû étudier et comprendre afin de le réutiliser efficacement et de l'adapter à de nouveaux cas d'usage. Ce service offrait une base solide, mais chaque page ayant ses spécificités, des ajustements ont été nécessaires pour garantir la cohérence et l'exactitude des données exportées.

### Difficultés rencontrées :

La première difficulté a été de **bien comprendre le fonctionnement du service d'exportation** existant. Il fallait identifier la manipulation des données, le format et le déclenchement du téléchargement. Ce travail m'a permis de mieux appréhender **la gestion des services partagés dans Angular**, ainsi que les particularités liées au formatage des fichiers Excel.

Une autre difficulté majeure a été de travailler sur des pages que je n'avais pas conçues, ce qui signifie que **les données à exporter étaient parfois réparties sur plusieurs composants**, avec une logique d'affichage fragmentée. Dans certains cas, les données provenaient de services différents ou étaient chargées de manière dynamique, ce qui m'a contraint à **faire une exploration minutieuse du code**, souvent en naviguant entre plusieurs fichiers, pour identifier les bonnes sources d'information.

Il fallait également veiller à **ne pas perturber le comportement existant de la page**, et à intégrer le bouton d'export de manière fluide dans l'interface utilisateur, avec un rendu cohérent et une utilisation intuitive.

### **Résultat obtenu**

Malgré les obstacles, j'ai pu **mettre en place avec succès l'exportation Excel sur plusieurs pages importantes de l'ERP**, parmi lesquelles :

- **La page d'édition – Liste des articles fournisseurs et clients**
- **La page Détail Client – Onglet Ventes**
- **La page Détail Article – Onglet Achats/Ventes**
- **La page de la proposition automatique de commande**

Dans chacune de ces pages, un **bouton dédié permet désormais d'exporter les données affichées** en un clic, sous la forme d'un fichier Excel bien structuré, avec les bons intitulés de colonnes et des formats adaptés à une lecture claire. Cette fonctionnalité a été testée et validée, répondant ainsi à un besoin concret des utilisateurs.

Ce travail m'a permis de **développer mes compétences en intégration de fonctionnalités sur du code existant**, en collaboration avec les développeurs déjà en place, mais aussi d'approfondir ma maîtrise des outils d'export et de traitement de données côté frontend. Il m'a surtout appris à m'adapter à un environnement déjà en place, à respecter une architecture de projet complexe, et à **analyser des composants Angular dans un cadre réel d'entreprise**.

## Conclusion Générale

Mon stage de dix semaines chez **SYG Software** a été une expérience enrichissante qui m'a permis d'approfondir mes connaissances techniques tout en découvrant un environnement professionnel dynamique. Travailler sur le développement d'applications web, notamment avec **Angular**, a été une véritable révélation. J'ai pu explorer de nouvelles technologies et améliorer mes compétences en frontend, ce qui m'a permis de voir sous un autre angle les enjeux du développement d'applications modernes.

Ce stage m'a aussi offert l'opportunité de travailler au sein d'une équipe agile, ce qui m'a appris l'importance de la collaboration et de la communication dans un projet de développement. Les défis techniques rencontrés, notamment en **TypeScript**, ont renforcé mes capacités à résoudre des problèmes et à m'adapter aux exigences du développement en entreprise.

## **Bibliographie**

[Angular] Angular Material 2010 : <https://material.angular.io/>

[Bootstrap] Bootstrap Doc 2010 : <https://getbootstrap.com/>

[SYG Software] SYG Software 2015 : <https://syg.fr>

# Glossaire

**API (Application Programming Interface)** : Un ensemble de règles et de protocoles qui permet à différentes applications de communiquer entre elles et d'échanger des données ou des fonctionnalités de manière simplifiée.

**Angular** : Un outil qui permet de créer des sites web interactifs en organisant le code de manière claire et réutilisable.

**Angular Material** : Un ensemble d'éléments prêts à l'emploi (comme des boutons, menus, formulaires, etc.) qu'on peut utiliser pour construire facilement une interface de site web qui soit claire, agréable à regarder et bien organisée. Ces éléments suivent un style visuel conçu par Google, ce qui permet d'avoir un rendu moderne et harmonieux, sans devoir tout créer de zéro.

**Array** : Un tableau qui permet de ranger plusieurs données ensemble (comme une liste), pour pouvoir y accéder et les modifier facilement.

**Backend** : La partie invisible d'une application qui gère la logique, le traitement des données et les interactions avec la base de données, permettant le bon fonctionnement de l'application.

**Bootstrap** : Un framework CSS populaire qui fournit des outils pour la création de sites web réactifs et mobiles. Il propose une grille flexible, des composants préconçus et des plugins JavaScript, permettant de développer rapidement des interfaces attrayantes.

**CSS** : Un langage utilisé pour styliser et mettre en forme les pages web, en contrôlant l'apparence des éléments HTML, comme les couleurs, les polices et les dispositions.

**Composant** : Élément visuel d'une application (comme un bouton ou une page), qui gère l'affichage et le comportement associé.

**Datadock** : Un référentiel de qualité pour les organismes de formation en France, attestant de leur conformité aux critères exigés par les financeurs publics et paritaires afin d'assurer la qualité des formations dispensées.

**ERP (Enterprise Resource Planning)** : Système qui aide une entreprise à gérer et automatiser ses activités (comme la production, les ventes ou la comptabilité) en les regroupant dans un seul outil.

**Frontend** : La partie visible et interactive d'une application avec laquelle les utilisateurs interagissent, englobant l'interface, le design et la logique côté client.

**Framework** : Un ensemble d'outils, de bibliothèques et de conventions qui facilite le développement d'applications en fournissant une structure préétablie.

**HTML** : Un langage utilisé pour structurer le contenu d'une page web, en définissant des éléments comme les titres, les paragraphes et les images.

**map** : Une méthode utilisée sur les tableaux (arrays) qui permet de créer un nouveau tableau en appliquant une fonction à chaque élément du tableau d'origine, sans modifier ce dernier.

**OpenClassrooms** : Une plateforme d'apprentissage en ligne qui propose des cours et des formations dans divers domaines, avec des parcours certifiants, permettant aux utilisateurs d'acquérir des compétences pratiques et professionnelles.

**Qualiopi** : Une certification délivrée aux prestataires de formation en France, attestant de la qualité de leurs processus et de leurs actions de formation, conformément aux exigences définies par la loi pour accéder aux financements publics ou paritaires.

**Service** : Bloc de code qui centralise des fonctions réutilisables, comme les échanges avec une base de données ou une API.

**Swagger UI** : Un outil open-source qui génère une interface interactive pour explorer et tester des API. Il permet d'envoyer des requêtes et de recevoir des réponses, facilitant ainsi le développement et la documentation des API.

**TypeScript** : Un langage utilisé pour développer des applications à grande échelle, en rendant le code plus prévisible et compréhensible.

# Organigramme SYG Software

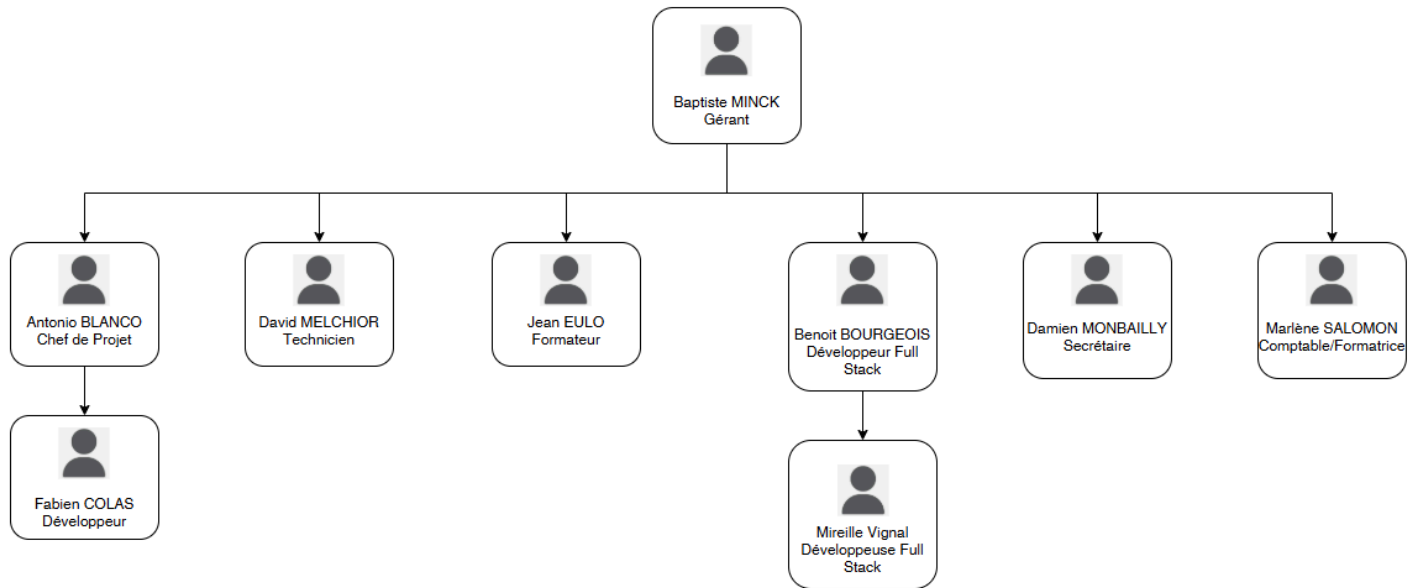


Figure 15 : Organigramme de SYG Software