



RAPPORT DE STAGE

BTS Services Informatiques aux Organisations
1re année – 2024-2025

ROCHER Maxime
15 mai 2025 au 27 juin 2025

Entreprise d'accueil : SNEF

Tuteur de stage : CUSSONNEAU Christian

Établissement : Lycée La Colinière

Enseignant référent : L. CHARLOT

Sommaire

1. Remerciements.....	3
2. Introduction.....	4
2.1. Présentation du groupe SNEF.....	4
2.2. Contexte et objectifs du projet.....	4
2.3. Domaine métier de mon intervention.....	5
2.4. Intégration dans l'équipe et la vie de l'entreprise.....	5
3. Architecture et organisation de l'application.....	5
3.1. Organisation du développement et méthodologie de travail.....	5
Développement par itérations, backlog et identification des tâches.....	5
Approche partiel Agile.....	5
Perspectives d'amélioration.....	6
3.2. Architecture technique de l'application.....	6
Hébergement.....	6
Architecture technique.....	6
Organisation du site et gestion des utilisateurs.....	7
Base de données et modélisation.....	7
Conséquence de l'historique du projet.....	8
Organisation du code.....	8
Gestion des utilisateurs et authentification.....	8
4. Mon intervention.....	9
4.1. Suivi de l'avancement du tirage de câbles.....	9
Mise en œuvre technique.....	9
Phase de test et ajustements.....	10
Débogage et amélioration continue.....	10
4.2. Comparaison des câbles entre SNEF et Chantiers de l'Atlantique.....	10
Objectifs fonctionnels.....	11
Problèmes rencontrés et ajustements.....	11
Système de logs.....	11
4.3. Optimisation des performances.....	12
Requêtes SQL.....	12
Traitements côté PHP et JavaScript.....	12
Résolution des problèmes liés à la pagination.....	12
Améliorations complémentaires.....	12
Adaptation pour les équipements SNEF.....	12
4.4. Validation et suivi des problèmes.....	13
5. Conclusion.....	13

1. Remerciements

Je remercie toute l'équipe de SNEF pour m'avoir accueilli dans de très bonnes conditions.

Un merci particulier à Christian CUSSONNEAU pour m'avoir fait confiance dès le début, pour ses explications claires, sa patience, et le temps qu'il a pris pour m'accompagner dans les tâches qui m'ont été données.

Merci aussi à tous ceux avec qui j'ai pu échanger, que ce soit pour m'aider sur le projet ou simplement pour discuter dans une bonne ambiance. J'ai vraiment apprécié cette immersion dans le monde professionnel, et je repars de ce stage avec une expérience concrète et utile pour la suite de mon parcours.

Enfin, merci à mes enseignants pour leur accompagnement tout au long de l'année et pour m'avoir bien préparé à ce stage.

2. Introduction

2.1. Présentation du groupe SNEF

L'agence SNEF de Saint-Nazaire est une entité du groupe SNEF, une **entreprise française** indépendante spécialisée de **plus 10000 salariés** dans le domaine de l'électricité. Son siège social est basé à Marseille. Présent sur l'ensemble du territoire français, le groupe intervient dans divers secteurs d'activités.

À **Saint-Nazaire**, où j'effectue mon stage, SNEF travaille principalement dans le secteur naval et EMR, en **partenariat avec les Chantiers de l'Atlantique**. L'entreprise se concentre essentiellement sur la distribution électrique des courants forts (CFO) et des courants faibles (CFA).

SNEF réalise notamment l'installation de câbles, d'appareils, d'armoires et de supports à bord des navires et des sous-stations électriques pour les champs éoliens offshore sur lesquels elle intervient.



Quelques références :

- [SNEF France](https://www.snef.fr/)¹
- [Linkedin SNEF](https://www.linkedin.com/company/groupe-snef)²
- [PAPPERS](https://www.pappers.fr/entreprise/snef-056800659)³
- [societe.com](https://www.societe.com/societe/snef-056800659.html)⁴

2.2. Contexte et objectifs du projet

Dans le cadre de ses activités à Saint-Nazaire, SNEF a mis en place une application informatique interne. Cette solution nommée SNEF Naval gère les différents aspects de l'activité de l'entreprise, notamment :

- **Étude et planification du câblage** à bord des navires (plans d'implantation, notes de calcul, cheminements, schémas électriques).
- **Saisie et suivi** de la pose des câbles et des équipements électriques dans l'outil de production (avancement, non-conformités, mètres).
- **Installation et raccordement** des câbles de puissance, de contrôle-commande et réseaux de communication.
- **Montage d'armoires et coffrets électriques** (installation et raccordement).
- **Contrôle du montage**.

D'autres activités, dans le secteur naval sont possibles :

- **Intégration d'automates et systèmes de contrôle-commande** (PLC, I/O, supervision).
- **Essais, mise en service et validation** des équipements à quai puis en mer.

1 <https://www.snef.fr/>

2 <https://www.linkedin.com/company/groupe-snef>

3 <https://www.pappers.fr/entreprise/snef-056800659>

4 <https://www.societe.com/societe/snef-056800659.html>

- **Maintenance corrective et préventive** des installations électriques navales et offshore.
- **Câblage pour l'éolien offshore** : interconnexions des sous-stations marines, pose de systèmes d'éclairage et de sécurité.

Elle constitue ainsi une réponse efficace aux besoins opérationnels de l'entreprise. Accessible à l'encadrement de chantier, au bureau d'études, aux chargés d'affaires, cette application est déployée via un intranet interne.

2.3. Domaine métier de mon intervention

✂ Je suis intervenu plus spécifiquement sur le domaine lié au traitement des câbles, à la réalisation de calculs d'optimisation et au suivi de la consommation.

2.4. Intégration dans l'équipe et la vie de l'entreprise

Dès mon arrivée, l'équipe m'a rapidement intégré dans une ambiance à la fois conviviale et professionnelle. Les échanges ont été enrichissants, aussi bien sur le plan technique que sur des aspects plus généraux liés à la vie en entreprise. J'ai particulièrement apprécié la confiance qui m'a été accordée, en tant qu'étudiant de première année de BTS, pour participer à des sujets concrets.

Ma première journée a été consacrée à la découverte des différents métiers exercés chez SNEF, à la présentation des locaux et à une introduction au site web interne, à ses fonctionnalités et à ses composants.

Cela m'a permis de m'impliquer pleinement dans le projet et de mieux comprendre les enjeux liés au démarrage d'un projet de développement applicatif, en particulier la nécessité d'obtenir des résultats visibles rapidement pour justifier l'investissement engagé.

3. Architecture et organisation de l'application

3.1. Organisation du développement et méthodologie de travail

Développement par itérations, backlog et identification des tâches

Les développements sont organisés par itérations successives, dont chacune a pour objectif d'intégrer de nouvelles fonctionnalités ou d'enrichir une fonctionnalité existante.

Ces itérations sont pilotées à partir d'un backlog⁵ de tickets dont chacun des tickets décrit l'ensemble des tâches à réaliser.

Approche partiel Agile

Ce backlog est géré via l'outil Microsoft Tasks. Les tickets s'apparentent à des user stories de la méthode Agile⁶.

5 Backlog : <https://en.wikipedia.org/wiki/Wikipedia:Backlog>

6 Manifeste Agile : <https://agilemanifesto.org/iso/fr/manifesto.html>

📌 Je définis les tâches à réaliser conjointement avec mon maître de stage.

On distingue deux types de tickets :

- Fonctionnelles, qui répondent à des besoins concrets des utilisateurs.
- Techniques : Ce sont les solutions techniques à mettre en place. Quand elles sont nécessaires avant de pouvoir développer une fonctionnalité, on les appelle alors des « Enablers »⁷ (ou facilitateurs).

Perspectives d'amélioration

- **Intégrer plus de cérémonies Agile** : Même si on s'inspire de la méthode Agile, le travail n'est pas découpé en périodes fixes de 2 à 3 semaines, appelées sprints⁸. Comme les personnes qui utilisent l'application sont aussi celles qui la développent, cela ne serait pas vraiment utile. De la même façon, faire des démonstrations à la fin de chaque cycle n'apporterait pas grand-chose pour l'instant. Par la suite, dans une perspective d'amélioration continue du projet, on pourrait évoluer en intégrant progressivement la méthode agile, et ajouter des points réguliers (rétrospectives) pour évaluer, dans les méthodes de travail, ce qui a bien fonctionné ou non.
- **Intégrer une gestion de configuration et version** : Actuellement, le projet est développé en interne sans système de gestion de version. L'introduction d'un outil comme GitLab pourrait permettre, à terme, de structurer davantage l'historique des modifications.
- **Ajouter un serveur de développement et automatiser les déploiements** : On pourrait aussi envisager d'installer un serveur dédié au développement. Une fois les tests terminés et la solution jugée satisfaisante, les fichiers sources seraient déployés sur le serveur de production. Ce processus pourrait être automatisé grâce à une démarche d'intégration continue, appelée CI/CD (intégration et déploiement continus), qui est le sujet de ma veille technologique et s'appuyant sur des pipelines.

3.2. Architecture technique de l'application

Hébergement

SNEF a opté pour une solution "on-premise"⁹ (hébergée en interne), contrairement à une solution cloud comme Google Cloud Platform (GCP), sur laquelle j'héberge par exemple mon portfolio à l'aide de services managés¹⁰

Architecture technique

La solution repose sur une architecture de type **AMP**, composée de :

- 🌐 **Un serveur HTTP Apache** qui est le serveur web en charge de la gestion des requêtes HTTP.

7 Enabler : https://framework.scaledagile.com/blog/glossary_term/enablers/

8 Sprint : <https://www.atlassian.com/agile/scrum/sprints>

9 On Premise : hébergés dans les locaux de l'entreprise (<https://stackoverflow.com/questions/2617456/on-premise-off-premise-applications-what-are-they>)

10 Mon portfolio : <https://portfolio-513482020117.us-central1.run.app/>

-  **Un serveur PHP¹¹** : langage côté serveur intégré à Apache pour le traitement dynamique des pages.
-  **Une base de données MySQL** pour une gestion de la base de données relationnelle pour stocker les données applicatives.

Ce choix d'architecture permet une mise en œuvre rapide, avec des technologies largement répandues et faciles à maintenir.

Organisation du site et gestion des utilisateurs

Le site est structuré en deux grandes interfaces distinctes, chacune répondant à des besoins spécifiques :

- **Interface d'administration** : réservée aux développeurs et administrateurs, elle intègre des outils avancés permettant d'interagir directement avec la base de données. Cette interface reprend des fonctionnalités similaires à celles que l'on trouve dans des outils professionnels comme DBeaver¹² ou MySQL Workbench¹³, facilitant ainsi le requêtage, la modification et la consultation des données.
- **Interface utilisateur** : destinée aux utilisateurs de l'entreprise, cette interface propose une expérience simplifiée avec des fonctionnalités limitées, en fonction du rôle de chacun. Elle permet d'accéder uniquement aux opérations autorisées selon les droits attribués.

Les utilisateurs sont catégorisés par profils ou rôles :

- Administrateur
- Chargés d'affaires
- BE Montage
- Utilisateur standard, lui-même réparti en plusieurs sous-rôles, en fonction de la profession ou des responsabilités de chacun dans l'entreprise (par exemple : technicien, responsable chantier, etc.).

Cette organisation permet d'assurer la sécurité des données, tout en adaptant l'interface et les fonctionnalités aux besoins de chaque profil.

Base de données et modélisation

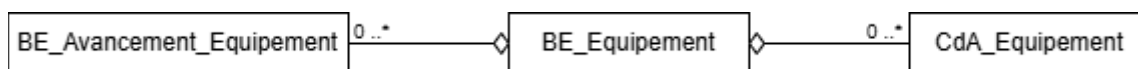
La base de données, de type relationnel, contient plusieurs milliers d'enregistrements. À titre d'exemple, pour un seul navire, cela embarque environ 200 000 câbles actifs et non actifs qui sont tous référencés en base de données.

 De mon côté la modélisation conceptuelle des données a été réalisée de façon informelle sur papier. Pour rédiger mes requêtes SQL, je me suis appuyé directement sur la modélisation physique des données, autrement dit sur les tables elles-mêmes. Cela m'a permis de prendre du recul et d'avoir une vue d'ensemble claire des liens entre les différentes tables, ce qui est essentiel lorsqu'on intervient sur des structures complexes.

11 PHP : <https://www.php.net/>

12 Dbeaver : <https://dbeaver.io/>

13 MySQL Workbench : <https://www.mysql.com/products/workbench/>



Exemple d'UML : Pour la partie comparaison des données d'équipement

A noter : Le choix a été fait de centraliser la logique métier dans le code PHP, ce qui facilite sa maîtrise et son évolution par les développeurs internes.

Conséquence de l'historique du projet

Le développement par itérations successives, comme expliqué précédemment, a conduit à des tables avec un nombre de champs important.

À terme, un refactoring pourrait être envisagé pour mieux découper ces tables et les recentrer sur des domaines fonctionnels distincts.

Organisation du code

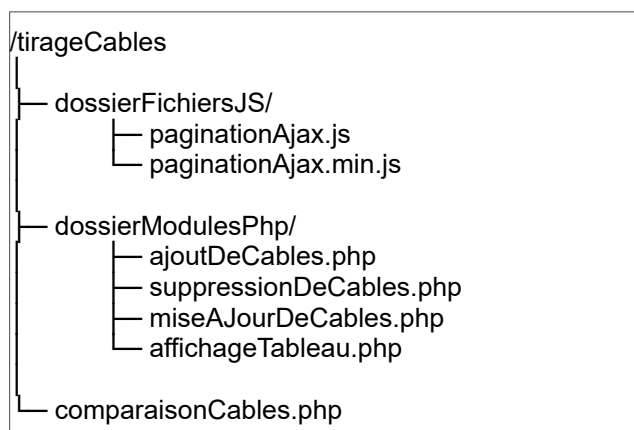
L'architecture initiale du projet privilégiait une intégration rapide des fonctionnalités. Durant mon intervention, j'ai pu proposer une séparation plus marquée entre les couches (inspirée du modèle MVC : "Modèle¹⁴, Vue¹⁵, Contrôleur¹⁶"), afin d'en faciliter la lisibilité et la maintenance dans la durée.

J'ai notamment:

- Séparé la vue (HTML/CSS/JS) de la logique métier (PHP) et minifié les fichiers JavaScript et CSS pour optimiser les performances ;

main.js	25/06/2025 16:13	Fichier de JavaScript	20 Ko
main.min.js	24/06/2025 15:14	Fichier de JavaScript	16 Ko
style.css	24/06/2025 14:08	Document de feui...	6 Ko
style.min.css	24/06/2025 14:09	Document de feui...	4 Ko

- Rendu le code plus lisible et réutilisable ;
- Introduit une synchronisation entre le front-end et le back-end via des requêtes XHR pour une communication plus fluide, sans rechargement de page à chaque requête.



Exemple de structure de fichier

Cela va permettre de faciliter la prise en main par d'autres développeurs de ma programmation.

¹⁴ Gestion des données

¹⁵ Partie affichée dans le navigateur

¹⁶ Traitement des actions utilisateur et liaison entre modèle et vue

Gestion des utilisateurs et authentification

La gestion des utilisateurs et des habilitations associées repose sur des solutions internes à l'entreprise, qui ne relèvent pas de mon périmètre d'intervention. On peut supposer que l'authentification s'appuie sur des protocoles standards tels qu'OAuth2¹⁷ ou SAML¹⁸, bien que cela ne m'ait pas été confirmé.

L'accès à l'application semble également restreint à un réseau interne, avec une filtration par adresses IP fixes, ce qui renforce la sécurité en limitant l'accès aux seules machines de l'entreprise.

4. Mon intervention

4.1. Suivi de l'avancement du tirage de câbles

Au cours de la première partie de mon stage, il m'a été demandé de développer une courbe permettant de suivre l'avancement du tirage de câbles sur les différents projets en cours, en comparaison avec des objectifs définis à l'avance. Cette courbe devait répondre à plusieurs critères fonctionnels :

- Générer une courbe pour chaque lot d'un navire sur lequel intervient la SNEF ;
- Afficher une courbe globale de l'avancement total du tirage des câbles ;
- Intégrer les objectifs de tirage pour chaque lot ainsi qu'un objectif global ;
- Permettre une analyse selon plusieurs axes :
 - le nombre de câbles,
 - la longueur en m,
 - le temps passé en h ;
- Afficher les annotations suivantes sur le graphique :
 - Date de début de tirage prévue (OB) et date réelle de début (D.T) ;
 - Date de fin de tirage prévue (OD) et date réelle de fin (F.T) ;
 - Date du jour, pour situer l'avancement dans le temps ;
 - Ratios objectifs et calculés pour atteindre la cible à la date OD :
 - un ratio fixe, correspondant à l'objectif initial,
 - un ratio corrigé, évolutif selon l'avancement réel ;
- Tracer dans les logs les actions et modifications réalisées par chaque utilisateur.

17 Informations complémentaire : <https://fr.wikipedia.org/wiki/OAuth>

18 Security Assertion Markup Language

Pour initier ce projet, un code existant m'a été transmis. Celui-ci comprenait l'utilisation d'une bibliothèque de graphiques ainsi qu'une base en PHP, que j'ai adaptée pour répondre aux besoins du projet.

Mise en œuvre technique

J'ai commencé par établir les requêtes SQL permettant de récupérer les données d'objectif ainsi que les dates clés pour les annotations. Une fois cette étape validée, j'ai modifié le code PHP et JavaScript pour générer la courbe globale.

Ensuite, j'ai intégré une liste déroulante regroupant l'ensemble des lots d'un navire. Pour chaque sélection, une courbe individuelle du tirage de câbles est affichée. J'ai répliqué cette logique pour afficher les courbes d'avancement réel.

Une fois les données et les courbes fonctionnelles, j'ai ajouté l'ensemble des annotations dynamiques sur les graphiques.

Phase de test et ajustements

Les tests ont été réalisés à partir de fichiers Excel de référence, enrichis avec des filtres et des formules comme RECHERCHEX pour croiser les données. Cela m'a permis de vérifier que les résultats produits étaient conformes aux attentes.

Après plusieurs itérations et corrections (notamment sur certaines requêtes SQL ou traitements PHP), la courbe a pu être intégrée dans l'application. Par la suite, quelques retours utilisateurs ont nécessité des ajustements, notamment concernant les annotations : dans certains cas, les courbes affichaient un taux de réalisation supérieur à 100 % alors que le tirage n'était pas terminé.

Débogage et amélioration continue

En cas d'incohérences, j'ai suivi une démarche pour isoler et corriger les erreurs :

- Analyse des requêtes SQL : je vérifiais d'abord les données brutes renvoyées. Si nécessaire, je modifiais la requête, ce qui n'était pas toujours évident compte tenu de leur complexité croissante. Pour rendre mes requêtes plus lisibles, j'ai eu recours aux CTE¹⁹ (Common Table Expressions), ce qui m'a permis de mieux identifier les sources d'erreur.
- Vérification du code PHP : si les requêtes étaient correctes, je m'orientais vers le code PHP, où s'effectuait une partie importante des calculs transmis ensuite au JavaScript.
- Affichages de débogage : pour mieux comprendre le comportement du code, j'utilisais des affichages temporaires directement dans la page, afin de visualiser les valeurs à chaque étape et remonter plus efficacement aux erreurs.

19 CTE : <https://learn.microsoft.com/fr-fr/sql/t-sql/queries/with-common-table-expression-transact-sql>

```

if (formData.get('changement') === 0) {
  // Mise à jour des données existantes
  myChart.data.labels = jours;
  myChart.data.datasets[0].data = ratios;
  myChart.data.datasets[1].data = nb_cable_tire;

  const annotations = myChart.options.plugins.annotation.annotations;

  annotations[0].value = data.date_ob;
  annotations[0].label.content = 'OB: ' + data.date_ob;

  annotations[1].value = data.date_od;
  annotations[1].label.content = 'OD: ' + data.date_od;

  annotations[2].value = data.date_debut_tirage;
  annotations[2].label.content = 'DT: ' + data.date_debut_tirage;

  annotations[3].value = data.date_fin_tirage;
  annotations[3].label.content = 'FT: ' + data.date_fin_tirage;
  annotations[3].label.yAdjust = adjustValue;

  annotations[4].value = Newtoday;
  annotations[4].label.content = 'DJ: ' + Newtoday;
  annotations[4].label.yAdjust = adjustValue;

  annotations[5].xValue = data.date_od_planning;
  annotations[5].yValue = data.total_cables * 0.75;
  annotations[5].content = ['Ratio Initial: ', ratio_moyen + unite_cbl];

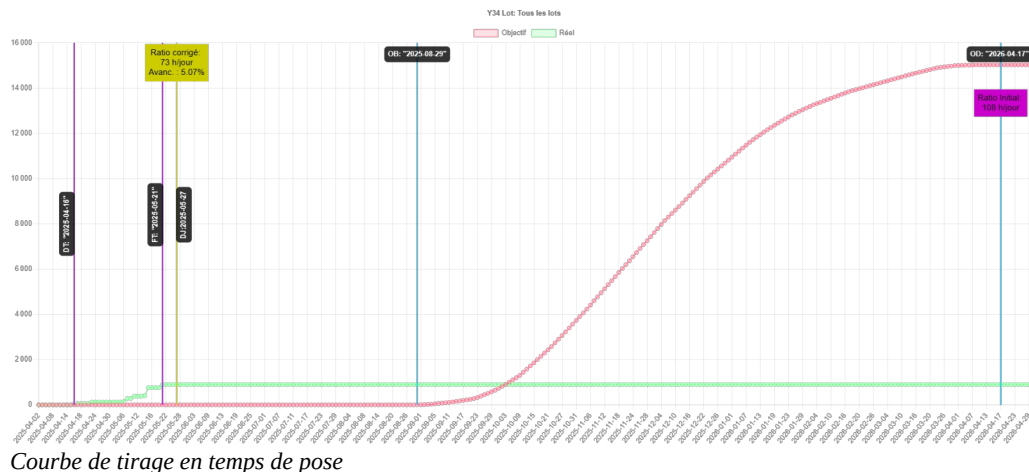
  annotations[6].xValue = Newtoday;
  annotations[6].yValue = adjustValue1;
  annotations[6].content = ['Ratio corrigé: ', ratio_corrige + unite_cbl, 'Avanc. : ' + avancement + '%'];

  myChart.update();
} else {

  if (window.myChart instanceof Chart) {
    window.myChart.destroy();
  }

  window.myChart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: jours,
      datasets: [
        {
          label: 'Objectif',
          data: ratios,
          borderColor: 'rgba(255, 99, 132, 1)',
          backgroundColor: 'rgba(255, 99, 132, 0.2)',
          borderWidth: 1,
          tension: 0.8
        },
        {
          label: 'Réel',
          data: nb_cable_tire,
          borderColor: 'rgba(99, 255, 132, 1)',
          backgroundColor: 'rgba(99, 255, 132, 0.2)',
          borderWidth: 1
        }
      ]
    },
    options: {
      maintainAspectRatio: false,
      responsive: true,
      scales: {
        y: { beginAtZero: true }
      }
    }
  });
}

```



4.2. Comparaison des câbles entre SNEF et Chantiers de l'Atlantique

Au cours de la deuxième partie de mon stage, j'ai été chargé de développer une interface permettant de comparer les données de câbles présentes dans la base de données de SNEF avec celles fournies par les Chantiers de l'Atlantique. L'objectif était de mettre en évidence toutes les incohérences entre les deux sources afin de faciliter la mise à jour de la base interne.

Objectifs fonctionnels

Le projet devait répondre aux exigences suivantes :

- Générer un tableau répertoriant les différences constatées entre les câbles des deux bases ;
- Produire un tableau listant les câbles présents uniquement dans les données Chantiers, à ajouter à la base SNEF ;
- Générer un tableau pour les câbles présents uniquement côté SNEF, à considérer comme supprimés ;
- Intégrer des boutons d'action (mise à jour, suppression, ajout) en fonction du tableau sélectionné ;
- Mettre en place un système de filtres pour isoler les différences spécifiques.

Problèmes rencontrés et ajustements

Dès le début du développement, je me suis heurté à des problèmes liés au volume important de données. Les requêtes SQL renvoyaient un grand nombre de résultats, ce qui ralentissait considérablement le chargement du tableau dans l'interface web.

Afin d'y remédier, j'ai mis en place une pagination dynamique, combinée à AJAX¹ (JavaScript et XML asynchrone).

AJAX a été utilisé pour les appels REST : j'ai principalement recours à la méthode **POST**²⁰ pour créer mes tableaux et interagir avec des fichiers PHP externes.

²⁰ Documentation de la méthode POST : <https://api.jquery.com/jQuery.post/>

Il m'aurait été possible d'utiliser la méthode **PUT**²¹ afin de mettre à jour les données de manière plus efficace, mais je n'en ai pris connaissance qu'après avoir terminé mon programme.

Quant à la méthode **GET**²², bien que je ne l'aie pas utilisée, elle reste utile pour la lecture de données.

Cela m'a permis d'avoir un chargement par blocs en Javascript et ainsi avoir un tableau plus fluide, des filtres permettant de charger les données par lots, tout en conservant la logique des filtres et des types de tableaux. Cependant, la mise en place de cette pagination a rapidement soulevé de nouveaux défis :

- Les filtres appliqués n'étaient pas pris en compte lors du changement de page ;
- Le système de pagination devenait incohérent lors de l'affichage de tableaux multiples (ajouts, suppressions, différences) ;
- Le temps de chargement de la page initiale était trop élevé (environ une minute) ;
- Le changement de page renvoyait parfois au mauvais tableau.

Système de logs

Comme mentionné dans les objectifs, un système de journalisation (logs) devait être mis en place. Une table dédiée m'avait été fournie, dans laquelle chaque modification effectuée sur la base de données via le site devait être enregistrée. Ces enregistrements incluent le nom et le prénom de l'utilisateur ayant réalisé la modification.

Pour cela, j'ai ajouté une requête SQL permettant d'insérer automatiquement une entrée dans la table des logs à chaque opération sur la base de données. J'ai également personnalisé ces logs afin qu'ils précisent la nature de la modification : ajout, suppression ou mise à jour.

```
$logStmt = $mysqli navire->prepare("
    INSERT INTO [REDACTED]
    (Navire, Prestation, Utilisateur, ID_CBL, CBL, Champ, Valeur)
    VALUES (?, ?, ?, ?, ?, ?, ?)
");
```

4.3. Optimisation des performances

Requêtes SQL

J'ai revu et optimisé les requêtes SQL, notamment en réduisant le nombre de jointures et en filtrant dès l'interrogation de la base. L'utilisation de CTE (Common Table Expressions) a amélioré la lisibilité et la maintenance des requêtes. Ces optimisations ont permis de réduire le temps de chargement à moins de trois secondes.

Traitements côté PHP et JavaScript

Initialement, de nombreux calculs étaient réalisés en PHP. J'ai progressivement déplacé certains traitements côté JavaScript pour améliorer la fluidité des calculs client. J'ai également utilisé des requêtes XHR (XMLHttpRequest) pour exécuter les traitements dans des fichiers PHP distincts, avec un retour en JSON, afin d'accélérer l'affichage des tableaux.

21 Documentation utilisé pour connaître la méthode PUT : https://docs.dhtmlx.com/suite/ajax/api/ajax_put_method/

22 Documentation de la méthode GET: <https://api.jquery.com/jQuery.get/>

Résolution des problèmes liés à la pagination

Pour résoudre les incohérences de pagination, j'ai intégré les filtres directement dans les requêtes paginées, garantissant une continuité logique entre les pages. J'ai également séparé clairement le traitement des trois tableaux dans le code pour éviter les interférences entre eux.

Améliorations complémentaires

Pendant certaines périodes d'attente liées à des retours ou à la validation de données, j'ai profité du temps disponible pour effectuer diverses améliorations secondaires :

- Amélioration de l'interface utilisateur (UI/UX) pour une navigation plus claire et agréable ;
- Mise en place d'un filtre spécifique sur la colonne "fonction du câble", permettant un tri plus pertinent ;
- Cette évolution a nécessité une refactorisation de la pagination, afin de l'adapter aux nouveaux filtres ;
- Optimisation du changement d'affichage entre les différents tableaux, pour gagner en réactivité.

Tout afficher		Câbles à Ajouter		Câbles à Supprimer		Tous les codes									
Page précédente		Page suivante													
Mettre à jour		1 / 15		Aller		Modifications de câbles									
☐	N°	Câble	Tirage	Ind. Câble	Ind. Câble actif	Gamme	Ind. gamme	Fonction	RSO	Section	Tenant	MAP Tenant	Abouissant	MAP Abouissant	Code erreur
☐	1	Y34-C-BSAB6VA		0	O			EPC	B	PX 3G 2.50	Y34-LB6162		Y34-BSAB6VA		Ecart PT_CBL
☐				0	O			EPC	B	PX 3X 6.00	Y34-LB6162	N	Y34-BSAB6VA	N	
☐	2	Y34-C-E1AB6HA		0	O			EPC	B	PX 3G 2.50	Y34-LB6191		Y34-E1AB6HA		Ecart PT_CBL
☐				0	O			EPC	B	PX 3X 6.00	Y34-LB6191	N	Y34-E1AB6HA	N	

Modification de câbles interface

Tout afficher		Câbles à Ajouter		Câbles à Supprimer		Tous les codes									
Page précédente		Page suivante													
Ajouter câbles		1 / 4		Aller		Ajout de câbles									
☐	N°	Câble	Tirage	Ind. Câble	Ind. Câble actif	Gamme	Ind. gamme	Fonction	RSO	Section	Tenant	MAP Tenant	Abouissant	MAP Abouissant	Code erreur
☐	1	Y34-C-5191DAAC		0				ELP	B	TX 10 0.75	Y34-5191DAAB		Y34-5191DAAC		Nouveau câble
☐	2	Y34-C-E1AB6HAA		0				EPC	B	PX 3G 2.50	Y34-E1AB6HA		Y34-E1AB6HAA		Nouveau câble

Ajout de câbles interface

Tout afficher		Câbles à Ajouter		Câbles à Supprimer		Tous les codes									
Page précédente		Page suivante													
Supprimer câbles		1 / 1		Aller		Suppression de câbles									
☐	N°	Câble	Tirage	Ind. Câble	Ind. Câble actif	Gamme	Ind. gamme	Fonction	RSO	Section	Tenant	MAP Tenant	Abouissant	MAP Abouissant	Code erreur
☐	1	Y34-C-H1AB6HAE		0	N					PX 3G 2.50					A supprimer
☐	2	Y34-C-H1AB6HAF		0	N					PX 3G 2.50					A supprimer

Dernière ligne !!!

Suppression de câbles interface

Adaptation pour les équipements SNEF

Une fois la comparaison des câbles finalisée et stabilisée, une nouvelle tâche m'a été confiée : réaliser un système similaire, cette fois pour les équipements SNEF, tels que les luminaires, prises de courant ou armoires électriques, également en lien avec les données des Chantiers de l'Atlantique.

Bien que l'objectif global reste identique (identifier les écarts entre deux bases de données), les conditions de comparaison, les filtres à appliquer et les règles de traitement étaient sensiblement différents. Par exemple, les critères de correspondance ne portaient plus uniquement sur des références

ou des longueurs, mais également sur des affectations fonctionnelles, des localisations ou des typologies d'équipements.

J'ai donc dû adapter les requêtes SQL et revoir certains éléments de l'interface pour prendre en compte ces nouvelles spécificités. La structure générale du projet m'a toutefois permis de réutiliser une partie du code (notamment les composants de pagination et d'affichage), ce qui a favorisé un développement plus rapide et plus robuste.

4.4. Validation et suivi des problèmes

La phase de test repose sur la comparaison entre les données attendues, généralement transmises sous forme de fichiers Excel, et les données obtenues après l'exécution de scénarios de test. Ces scénarios ne sont pas formalisés dans un outil spécifique : ils sont présentés oralement par mon maître de stage, étape par étape, et je les note afin de pouvoir les reproduire.

Chaque scénario comprend plusieurs étapes de validation, dont les résultats attendus sont détaillés dans les fichiers Excel fournis en amont. De son côté, mon maître de stage effectue également ses propres tests pour valider les fonctionnalités que j'ai développées.

Lorsqu'aucune anomalie n'est détectée, la fonctionnalité est ouverte aux utilisateurs. Si des anomalies apparaissent après la mise en service, elles sont évaluées en fonction de leur niveau de gravité (bloquante, majeure, moyenne ou mineure). Le traitement des priorités est défini par mon maître de stage : certaines anomalies sont corrigées immédiatement, tandis que d'autres peuvent être reportées si une nouvelle fonctionnalité est jugée prioritaire.

À titre personnel, je m'efforce de corriger les anomalies dans les meilleurs délais, afin d'éviter qu'elles ne s'accumulent. Il arrive également que certaines évolutions fonctionnelles soient demandées après coup, lorsque la solution ne répond pas entièrement aux attentes des utilisateurs. Ces ajustements sont alors intégrés au fil de l'eau.

5. Conclusion

Ce stage a constitué une première véritable immersion dans le monde de l'entreprise, avec ses exigences, ses rythmes et ses attentes concrètes. Bien qu'orienté avant tout vers la productivité et la contribution directe à des projets en cours, il m'a offert une occasion précieuse de mettre en pratique mes compétences techniques en développement web (notamment en PHP et JavaScript) et en gestion de bases de données via le langage SQL.

J'ai particulièrement apprécié la confiance qui m'a été accordée dès les premiers jours, malgré mon statut d'étudiant de première année de BTS. L'équipe m'a rapidement intégré dans une ambiance à la fois bienveillante et professionnelle, où l'entraide, la rigueur et la bonne humeur coexistent naturellement.

Travailler sur des cas concrets tels que le suivi de l'avancement de tirage de câbles ou la comparaison de données techniques entre plusieurs systèmes m'a permis de mesurer la complexité d'un projet informatique réel, tout en développant ma capacité à m'adapter, à trouver des solutions et à travailler en autonomie, tout en respectant un cadre collectif.

Au-delà des compétences techniques, cette expérience m'a permis de mieux comprendre les enjeux d'un service informatique dans une entreprise industrielle, et de me projeter plus concrètement dans mon parcours professionnel.

Je garderai un excellent souvenir de ce stage, autant pour les aspects techniques que pour la qualité humaine des personnes qui m'ont accompagné.