



RAPPORT DE MISSION 2

Mission effectuée du 27/03/2022 au 12/05/2023



*Tuteur Académique :*

Christian DESVALOIS

*Tuteur Professionnel :*

HAMZA BEN JEMAA

*Rédigé par : MONKAM TOWO Jonathan*

TABLE DES FIGURES

[Figure 1 : BNPPARIBAS DANS LE MONDE 5](#_Toc134995811)

[Figure 2 : BNP PARIBAS CIB 6](#_Toc134995812)

[Figure 3 : CYCLE DE VIE D'UNE RELEASE 7](#_Toc134995813)

[Figure 4 : PRESENTATION D'OCTANE 8](#_Toc134995814)

[Figure 5 : GESTION DES DEFAUTS 11](#_Toc134995815)

[Figure 6 : ARCHITECTURE DE L'APPLICATION 13](#_Toc134995816)

[Figure 7 : Page de connexion 14](#_Toc134995817)

[Figure 8 : Page de configuration 14](#_Toc134995818)

[Figure 9 : Page de lancement des tests 15](#_Toc134995819)

[Figure 10 : REPORT CALYPSO 16](#_Toc134995820)

TABLE DES TABLEAUX

[Tableau 1 : TEST UNITAIRE 9](#_Toc134995779)

[Tableau 2 : TEST D'INTEGRATION 10](#_Toc134995780)

[Tableau 3 : TEST FONCTIONNEL 10](#_Toc134995781)

[Tableau 4 : TEST DE REGRESSION GLOBAL 10](#_Toc134995782)

Table des matières

[INTRODUCTION 3](#_Toc134997182)

[I. PRESENTATION DE L’ENTREPRISE D’ACCUEIL 4](#_Toc134997183)

[1. Historique de BNP PARIBAS 4](#_Toc134997184)

[2. Quelques chiffres clés 4](#_Toc134997185)

[3. BNP Paribas Corporate and Institutional Banking 5](#_Toc134997186)

[II. STRATEGIE DE TEST 5](#_Toc134997187)

[1. Rappel Présentation du Progiciel CALYPSO 5](#_Toc134997188)

[2. PRESENTAION DE L’Organisation de projet 6](#_Toc134997189)

[3. Test et Séquence d’intégration 7](#_Toc134997190)

[4. Objectifs et critère DE TESTS. 7](#_Toc134997191)

[5. gestion des defauts 10](#_Toc134997192)

[III. DESCRIPTION DU TRAVAIL REALISE 11](#_Toc134997193)

[1. Projet d’automatisation des NRT 11](#_Toc134997194)

[2. Creation d’un report calypso 14](#_Toc134997195)

[CONCLUSION 15](#_Toc134997196)

GLOSSAIRE

ST (Schedule Task) : est une tache automatisée sur calypso pouvant être exécuté avec des paramètre de date bien précise.

TRADE : Un Trade est une transaction effectuée entre deux parties qui caractérise l’aboutissement de la négociation entre ces deux parties.

# INTRODUCTION

Dans mon cursus d’ingénieur en informatique à l’institut d’ingénierie informatique de Limoges, j’ai eu l’opportunité de réaliser ma dernière année d’étude sous contrat de professionnalisation à BNP PARIBAS Securities Services, situé à PANTIN en tant qu’Architecte Technique (TA) au sein de l’équipe MFS Calypso.

BNP Paribas Securities Services est une filiale de BNP Paribas, l’un des plus grands groupes bancaires au monde. La société fournit une large gamme de services aux client institutionnels et aux entreprises, notamment des services de prêt de titres, d’administration de fonds et de garde.

Ce contrat qui a pour principal objectif de parfaire la formation commencée à l’école tout en facilitant mes rapports avec le monde professionnel, s’articule selon un découpage particulier bâti autour de 3 grandes périodes en entreprise. La description de la seconde période comprise entre le 27 mars 2023 et le 12 mai 2023 fera l’objet de ce rapport de mission. Tout au long de ce rapport, je ferais une brève présentation de l’entreprise qui m’accueille dans le cadre de mon contrat de professionnalisation, puis j’enchainerais avec la stratégie de test mis en place dans mon équipe, et enfin je présenterais les différentes taches réalisées durant ladite période.

# PRESENTATION DE L’ENTREPRISE D’ACCUEIL

### Historique de BNP PARIBAS

BNP Paribas est la première banque de l’Union européenne et acteur bancaire international de premier plan. Elle est présente dans 65 pays avec près de 190 000 collaborateurs dont près de 145 000 en Europe. Le Groupe accompagne tous ses clients particuliers, associations, entrepreneurs, PME-ETI, grandes entreprises et institutions dans la réussite de leurs projets grâce à ses solutions de financement, d’investissement, d’épargne et de protection.

BNP Paribas occupe des positions clés dans ses trois pôles opérationnels :

* **Commercial, Personal Banking & Services** fédère les réseaux des banques de détail du Groupe et plusieurs métiers spécialisés.
* **Investment & Protection Services** regroupe des métiers spécialisés offrant un large éventail de solutions d’épargne, d’investissement et de protection.
* **Corporate & Institutional Banking** propose des solutions financières sur mesure pour les clients, entreprises et institutions.

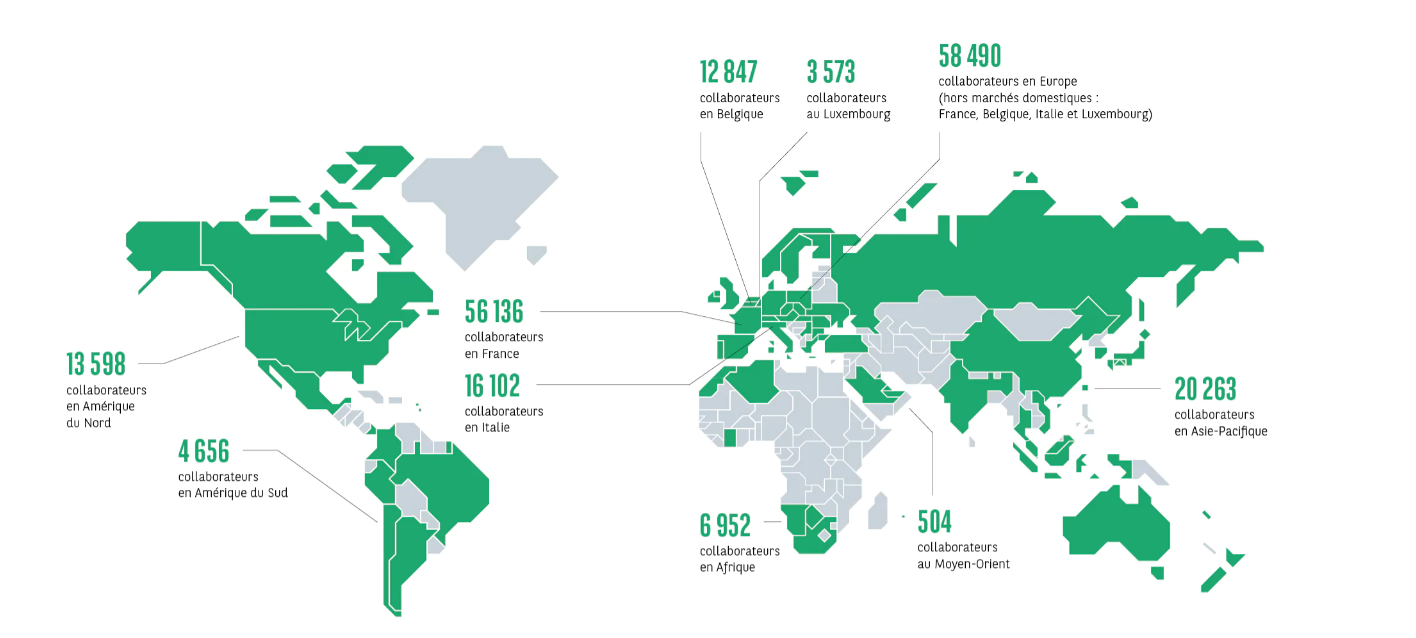


Figure 1 : BNPPARIBAS DANS LE MONDE

### Quelques chiffres clés

* 50,4 Md€ Produit net bancaire
* 10,2 Md€ Résultat net part du Groupe
* 12,3 % Ratio CET1
* 461 Md€ Réserve de liquidité immédiatement disponible
* 949,6 Md€ de crédits à la clientèle
* 1 189 Md€ d’actifs gérés par les équipes de l’Asset Management, de BNP Paribas Wealth Management, de BNP Paribas Cardif et de BNP Paribas Real Estate
* N° 1 mondial avec 19,5 Md€ d’obligations vertes (Bloomberg), meilleure banque dans le monde en matière de finance en 2022 (Euromoney).

### BNP Paribas Corporate and Institutional Banking

BNP PARIBAS CORPORATE AND INSTITUTIONAL BANKING propose des solutions financières sur mesure à ses clients institutionnels et entreprises.

Ci-dessous BNP PARIBAS CIB en quelques mots.

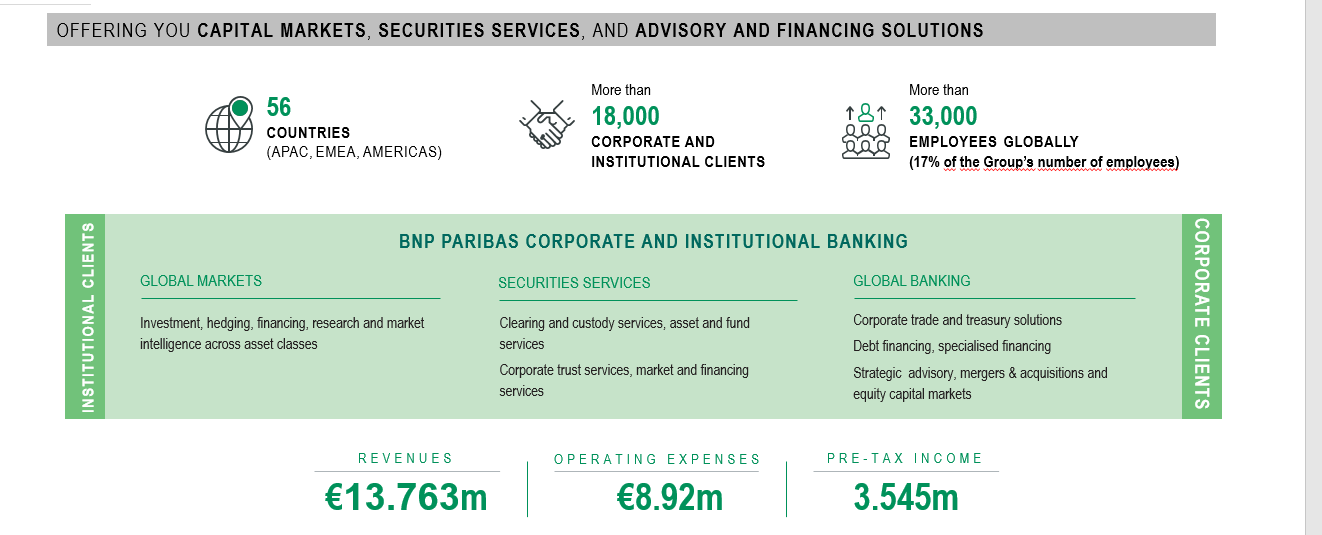


Figure 2 : BNP PARIBAS CIB

Les activités de se pole opérationnel sont :

* **GLOBAL MARKETS** : Investissement, couverture, financement, recherche et renseignements sur les marchés dans toutes les classes d’actifs.
* **SECURITIES SERVICES** : Services de compensation et de garde, services d’actifs et de fonds, services fiduciaires d’entreprises, services de marché et de financement.
* **GLOBAL BANKING**: Corporate Clients Group, Capital Markets, Transaction Banking and COO Office.

# STRATEGIE DE TEST

### Rappel Présentation du Progiciel CALYPSO

Calypso est un progiciel développé par Adenza, Inc. en langage Java. C’est une suite d’applications intégrant des fonctionnalités liées au trading et à la gestion des risques notamment pour les entreprises ayant des intérêts sur les marchés financiers.

Il faut bien comprendre que l’application Calypso fournit des fonctionnalités très étendues pour toutes les entités d’une banque, à savoir :

* Front Office (ou « FO »): c’est la partie la plus proche du marché, en charge de la négociation et du traitement des opérations financières avec les contreparties.
* Middle Office (ou « MO »): il s’agit des services financiers en charge du contrôle des positions bookées par le front office et de suivi des litiges remontés par le Back Office
* Back Office (ou « BO »): c’est le département qui est en charge de contrôle des échanges, du règlement et de la livraison ou encore des activités de régulation.

D’autres grandes banques françaises ont fait le choix de l’utiliser comme la Société Générale, la Banque Postale, Natixis ou encore la Banque de France.

Chez BNP Paribas, Calypso est utilisé comme outil du BackOffice, alimenté grâce aux données fournies par le FrontOffice.

Au sein de notre équipe, nous comptons au total 5 environnement (serveur d’application et serveur de base de données) de DEV Calypso, 8 environnement de Test, un environnement de PREPROD et la PROD.

### PRESENTAION DE L’Organisation de projet

Les projets sont menés en méthodologie Agile : Nous travaillons sur des cycles itératifs appelés "Sprint" d'une durée de deux semaines et des Releases d’un mois et demi.

La méthodologie agile que l’équipe MFS Calypso a adopté, même si elle s’inspire de la méthode Scrum, n’est identique à aucun type de méthode agile, et s’avère être propre à notre équipe.

Le cycle de vie standard d’une Release est la suivante :

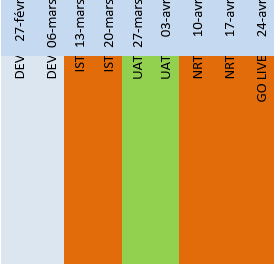


Figure 3 : CYCLE DE VIE D'UNE RELEASE

* Dev : phase de développement des différents sujets.
* IST : phase de test par l’équipe IT.
* UAT (USER ACCEPTANCE TEST) : phase de test effectué par les utilisateurs.
* NRT : phase de test de régression globale.
* Go Live : phase de déploiement en PROD des nouveaux développements.

### Test et Séquence d’intégration

Dans notre équipe, des tests interviennent dans différentes phases du cycle d’une Release.

* Analyse : Les différents cas de test (technique et fonctionnel) sont rédigés pendant cette phase par l’équipe IT (BA, TA, ou Testeur). Ils sont rédigés sur la plateforme d’automatisation de test OCTANE.

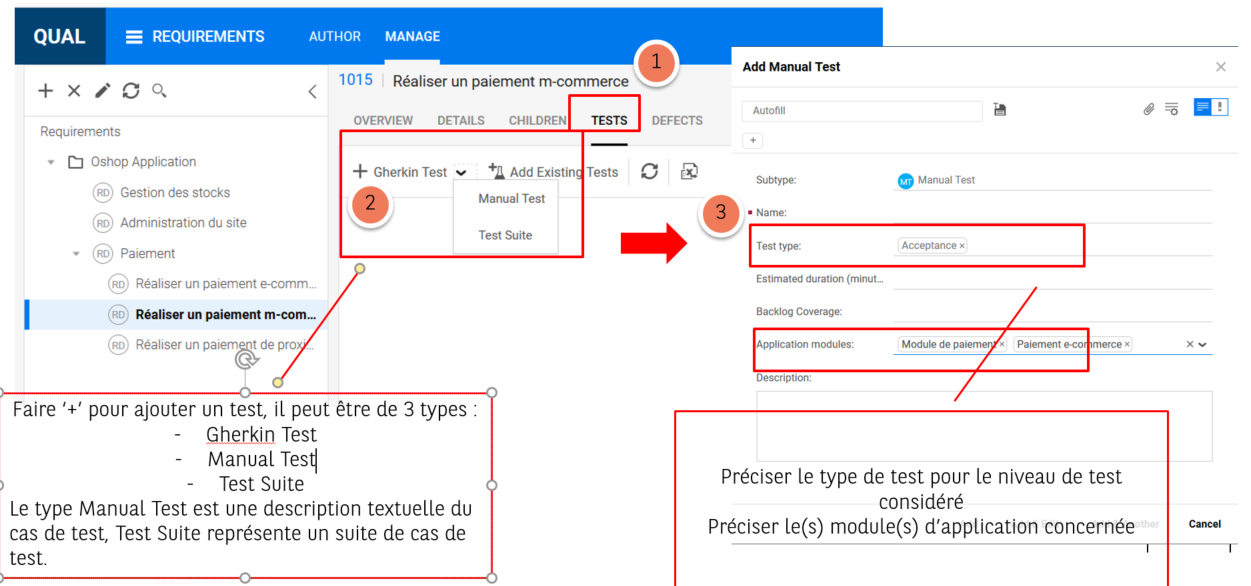


Figure 4 : PRESENTATION D'OCTANE

* Développement : Les tests unitaires sont effectués par les développeurs pour valider que les nouveaux développements fonctionnent correctement sans bugs techniques.
* IST : Deux types de test sont réalisés en parallèle, les tests de régression et les tests d’intégrations.
* UAT : Trois types de tests sont effectué durant cette phase :
* Les tests fonctionnels de régression : effectué par les utilisateurs
* Les tests d’acception de l’utilisateur : effectué par les utilisateurs sur de nouvelles fonctionnalités.
* Les tests d’acceptation technique : effectué par l’équipe IT sur les changements techniques.
* Régression GLOBAL : Ici nous effectuons un test global de non régression effectué sur un environnement dédié si possible sur une fonctionnalité critique. Cette phase se déroule pendant 5 jours et le jeu de test est partagé avec l’équipe business.

### Objectifs et critère DE TESTS.

#### Tests unitaires

Tableau 1 : TEST UNITAIRE

|  |  |
| --- | --- |
| Test Unitaire |  |
| Objectif du test | A la responsabilité de l’équipe de développement, ces tests ont pour objectif de vérifier que les développements fonctionnent correctement sans bugs techniques, et de s’assurer que le code développé est conforme aux règle sonar Bp2s. |
| Effectué par | Le développeur dont est assigné la modification. Il est en charge de :   * La préparation et l’exécution de ses tests. * L’analyse et la correction des défauts de code. |
| Prérequis | * Environnement de développement soit up * Le schéma de la base de donnée dédié soit up * Plugin Sonar soit up |
| Données de tests | Les données sont extraites de la production deux fois par an et insérées partiellement et manuellement (en raison de la taille limite de la base de données) dans la base de données de développement.  Ensuite, les tests de données sont créés/mis à jour en fonction des besoins du développeur. |

#### Analyse du code statique

L'analyse statique se concentre sur l'identification des faiblesses et des vulnérabilités de sécurité introduites dans la conception par le code source. L'examen des codes de sécurité peut être effectué manuellement et à l'aide d'un outil automatisé d'analyse des codes statiques.

En plus de l'analyse statique, un responsable de la sécurité peut également effectuer un balayage manuel des zones très sensibles du code. Ces domaines d'intérêt comprennent le code qui met en œuvre le trading de faible latence, permettent d’effectuer la validation des entrées ou réalise des opérations cryptographiques. Cette analyse est hebdomadaire et est effectuée à l'aide de l'outil Sonar Bp2s fourni par l'équipe Architecture (Sonarqube). Sonarqube peut être utilisé pour effectuer l'analyse du code, et les rapports produits doivent être évalués pour identifier les mesures d'atténuation appropriées, pour éliminer les vulnérabilités dans la mesure du possible.

Dans notre équipe, la fréquence d’analyse est d’une fois par semaine et elle est effectuée uniquement sur la branche d’intégration sur GIT. Elle suit également plusieurs métriques d’analyse dont :

* Les dettes techniques.
* Les bugs prioritaires de code.
* Les vulnérabilités prioritaires du code.
* Les duplications de code.
* L’ordonnancement du code.
* La couverture des tests unitaires.

#### Les tests d’intégration

Tableau 2 : TEST D'INTEGRATION

|  |  |
| --- | --- |
| IST |  |
| Objectif des tests | Vérifier si toutes les **Schedule Task** et les services Calypso sont toujours en cours d’exécution. |
| Effectué par | L’équipe de développeurs et l’équipe de Business Analystes |
| Prérequis | * Développement a été poussé dans la branche d’intégration * Le Build calypso est réussi * L’environnement d’intégration est UP * Plugin Sonar est UP |
| Données de tests | Donnée crée et mise à jour en accord avec le cas de test |

#### Les tests fonctionnels

Tableau 3 : TEST FONCTIONNEL

|  |  |
| --- | --- |
| UAT |  |
| Objectif du test | C'est la responsabilité des utilisateurs. Le but de ces tests est de valider que les améliorations produisent des résultats en fonction des exigences opérationnelles et des spécifications fonctionnelles. |
| Effectué par | L’équipe projet et les opérationnel qui sont chargés de :   * L’organisation de la phase UAT * La préparation et l’exécution des cas de tests * La validation de la release a déployer en production |
| Prérequis | * Disponibilité de l’environnement UAT (tous les composants fonctionne correctement) * Autorisation d’accès au utilisateur a l’environnement UAT |
| Donnée de test | La base de données UAT est chargée une fois avec les données de production au début de la phase UAT. Il peut être rechargé exceptionnellement lorsque les données ont été corrompues. |

#### Les tests de régressions Globaux

Tableau 4 : TEST DE REGRESSION GLOBAL

|  |  |
| --- | --- |
| Global Régression test |  |
| Objectif des tests | C'est à la responsabilité de l'équipe IT.  L'objectif de ces tests est de s'assurer que certaines améliorations pour une activité n'impactent aucune fonction critique à Calypso pour une autre activité |
| Effectué par | IT qui sont en charge de :   * Créer le package technique * La préparation et l’exécution des cas de test contenu dans OCTANE * D’uploader les réglages * D’analyser les défauts avec l’équipe business. |

#### Les tests de stress ou test de performance

L'objectif du test de stress est de s'assurer que les fonctions ou les processus sont évaluables avec des volumes élevés. En téléchargeant avec un outil spécifique, des volumes massifs de scénarios de transactions peuvent être insérés dans le système, simulant une forte utilisation de Calypso et de ses processus quotidiens. Ces tests devraient être effectués au moins une fois par an et leurs résultats présenter à l'entreprise. Les tests sont conservés sur OCTANE.

### gestion des defauts

Les défauts de test survenant au cours de la séquence de test doivent être consignés dans OCTANE, ou tous les commentaires d’analyse et de résolution sont enregistrés.

Ci-dessus, le flux de travail pour la gestion des défauts de test.

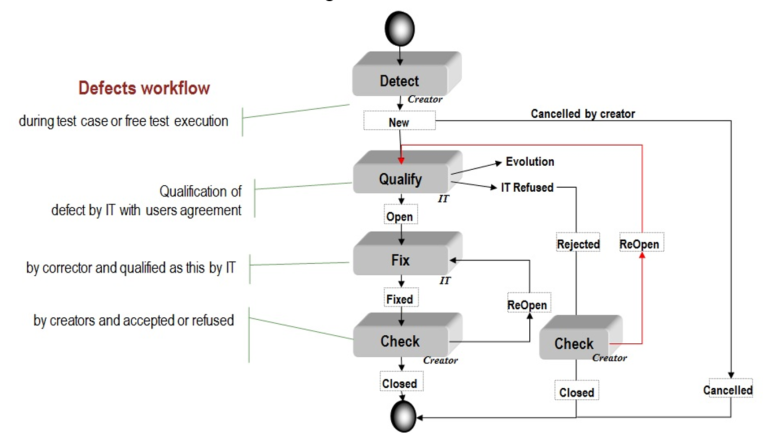


Figure 5 : GESTION DES DEFAUTS

# DESCRIPTION DU TRAVAIL REALISE

### Projet d’automatisation des NRT

Ce projet a pour objectif de créer une application web en utilisant JAVA Spring pour automatiser les tests de non-régression des personnalisations apportées au progiciel Calypso. Cette application web a pour but de vérifier si les modifications apportées au logiciel n'ont pas affecté les fonctionnalités existantes. L'objectif est de garantir que le système fonctionne correctement après chaque mise à jour ou modification.

##### Organisation

Ce projet est effectué par deux développeur à Paris, et un développeur à Chennai. Nous avons mis en place un point d’échange en anglais organisé en Daily meeting pour discuter sur l’avancé des taches de chacun, et chaque jeudi de la semaine, nous avons un meeting avec le Project Manager basé à Chennai en INDE pour lui présenter l’avancée du projet.

##### Architecture

Pour cette Application, nous nous sommes basé sur l’architecture MASTER-SLAVE, avec un maitre, et deux slaves. Le maitre est déployé dans l’environnement de PREPROD et c’est lui qui est chargé de :

* Orchestrer la procédure de test.
* Etablir la communication entre les slaves à travers l’API REST.
* Recevoir les réponses des exécutions.
* Lancer le script de test de non régression.
* Enregistrer les résultats.

Les slaves quant à eux sont déployé l’un dans l’environnement UAT\_REF et l’autre dans UAT\_NRT. Ils sont chargés de :

* Vérifier si tous les servers et les moteurs de calypso sont UP (Dataserver, Engineserver, …)
* Nettoyer le dossier de la ST avant son exécution.
* Télécharger des fichiers d’entré s’il y’en a.

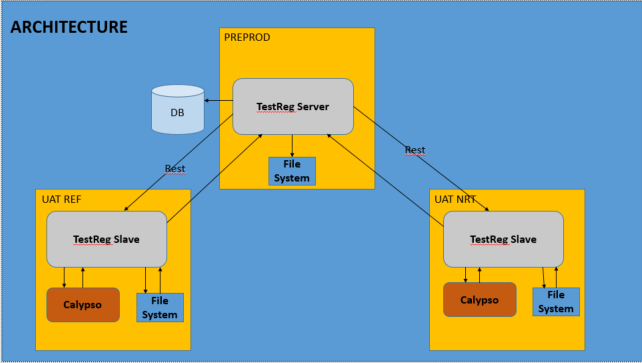


Figure 6 : ARCHITECTURE DE L'APPLICATION

##### Création des IHM (interface homme machine)

Pendant la première phase du projet, ma principale tâche était de réaliser les différentes interface homme machine pour les utilisateurs de l’application, et de les relier au données enregistré en base. Pour ce faire je suis partie sur architecture simple en utilisant un moteur de Template Spring appelé THYMELEAF, et comme Framework j’ai utilisé BOOTSTRAP 5.

Je vais vous présenter quelques pages créées.

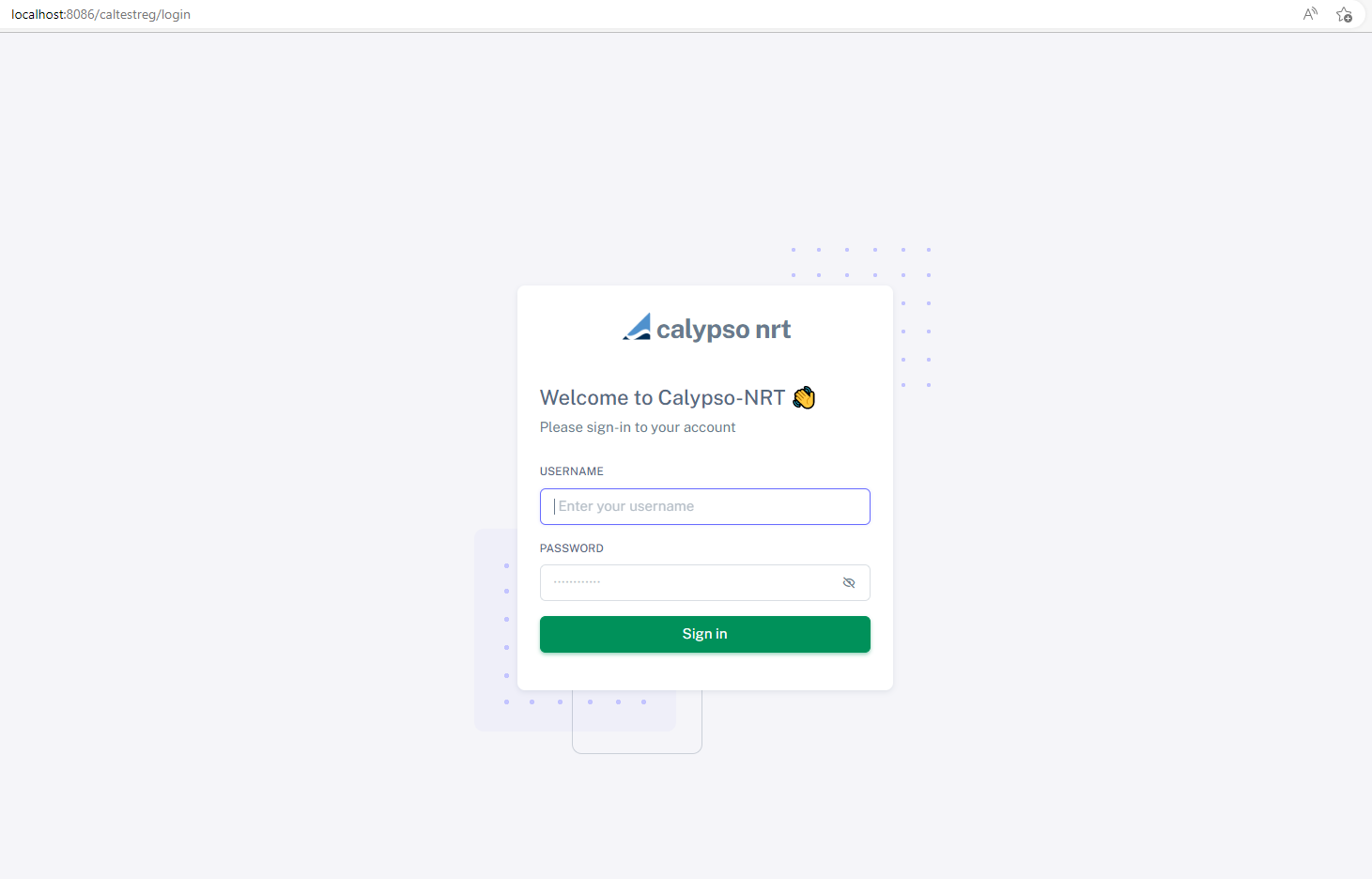


Figure 7 : Page de connexion

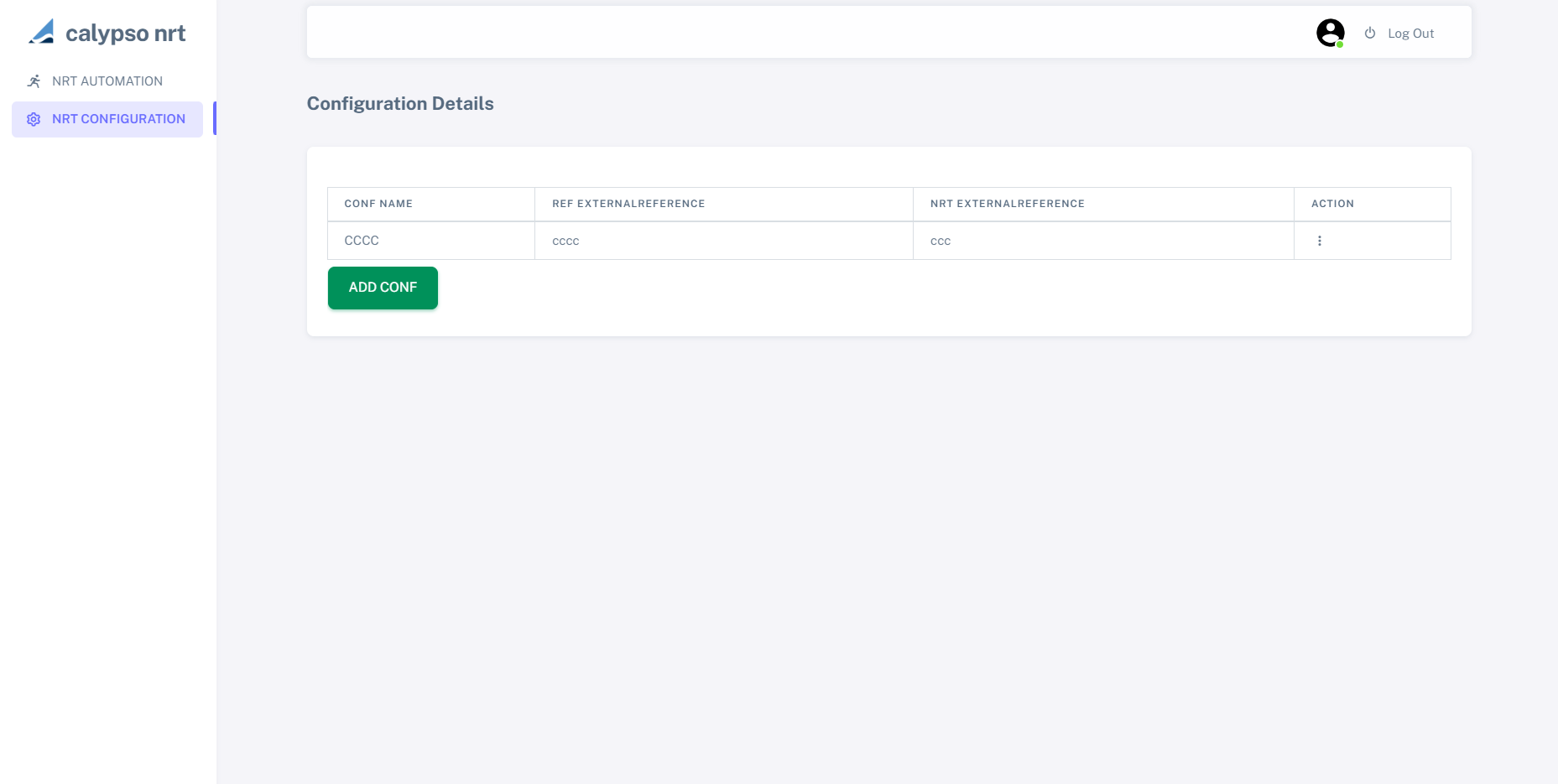


Figure 8 : Page de configuration

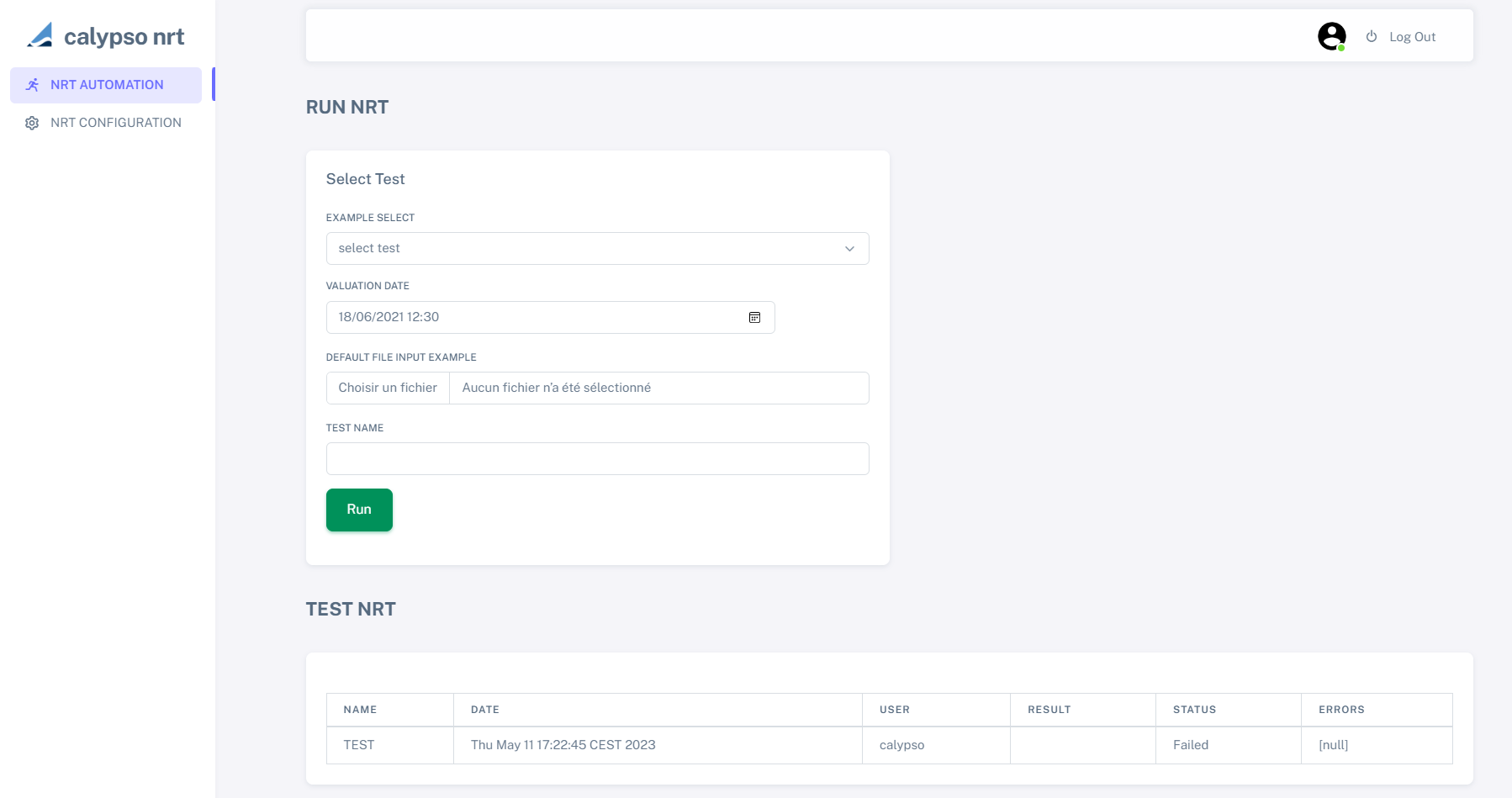


Figure 9 : Page de lancement des tests

##### Continuité du projet

Les différentes taches à réaliser pour la suite de l’application sont :

* La migration des interfaces en ANGULAR.
* L’exécution de façons automatisé de plusieurs cas de test.
* La Mise en place des test unitaire, d’intégration et fonctionnel.

### Creation d’un report calypso

Cette tache portais sur la création d’un report calypso permettant de visualiser les utilisateurs ayant un accès temporaire d’urgence a un environnement sur une période de date bien précise.

L’objectif final de ce report est de créer une ST qui va l’appeler et l’envoyer de façon journalière a l’Application Manager pour qu’il est un contrôle permanent sur les accès de chaque utilisateur.

Pour la création de celui-ci, j’ai mis en place une méthode SQL permettant de recevoir les données de la base, puis créer trois classes afin d’implémenter mon report.

* **EmergencyUserAccessTemplate** : Le Template et ses sous-classes fournissent les moyens de définir les paramètres de recherche et les données que le report devrait contenir. Il permet de stocker ces renseignements dans la base de données pour les utiliser dans les fenêtres du report et/ou dans l’exécution des taches prévues dans le report. Par exemple, pour mon report, je les ai personnalisés en y insérant deux paramètres de date. (StartDate et EndDate).
* **EmergencyUserAccessReport** : La classe de Report est ou une requête est construite en fonction des paramètres d’entrée dans le modèle de report. La requête est construite ; un appel est fait au serveur de données Calypso (via l’API distante) et les données retournées sont un ReportOutput avec un ensemble de ReportRows. Un ReportRow identifie les objets récupérés du système par rangée. La raison d’avoir les objets ReportRow (par opposition au simple passage des objets Vector lui-même) est qu’il y’a des moments où il faut associer plusieurs objets différents à une rangée. Par exemple, un report de message aura parfois un message, un transfert et un Trade tous liés au même report.
* **EmergencyUserAccessStyle** : ReportStyle est une classe d'aide qui fournit la fonctionnalité pour extraire les valeurs des colonnes des objets ReportRow.

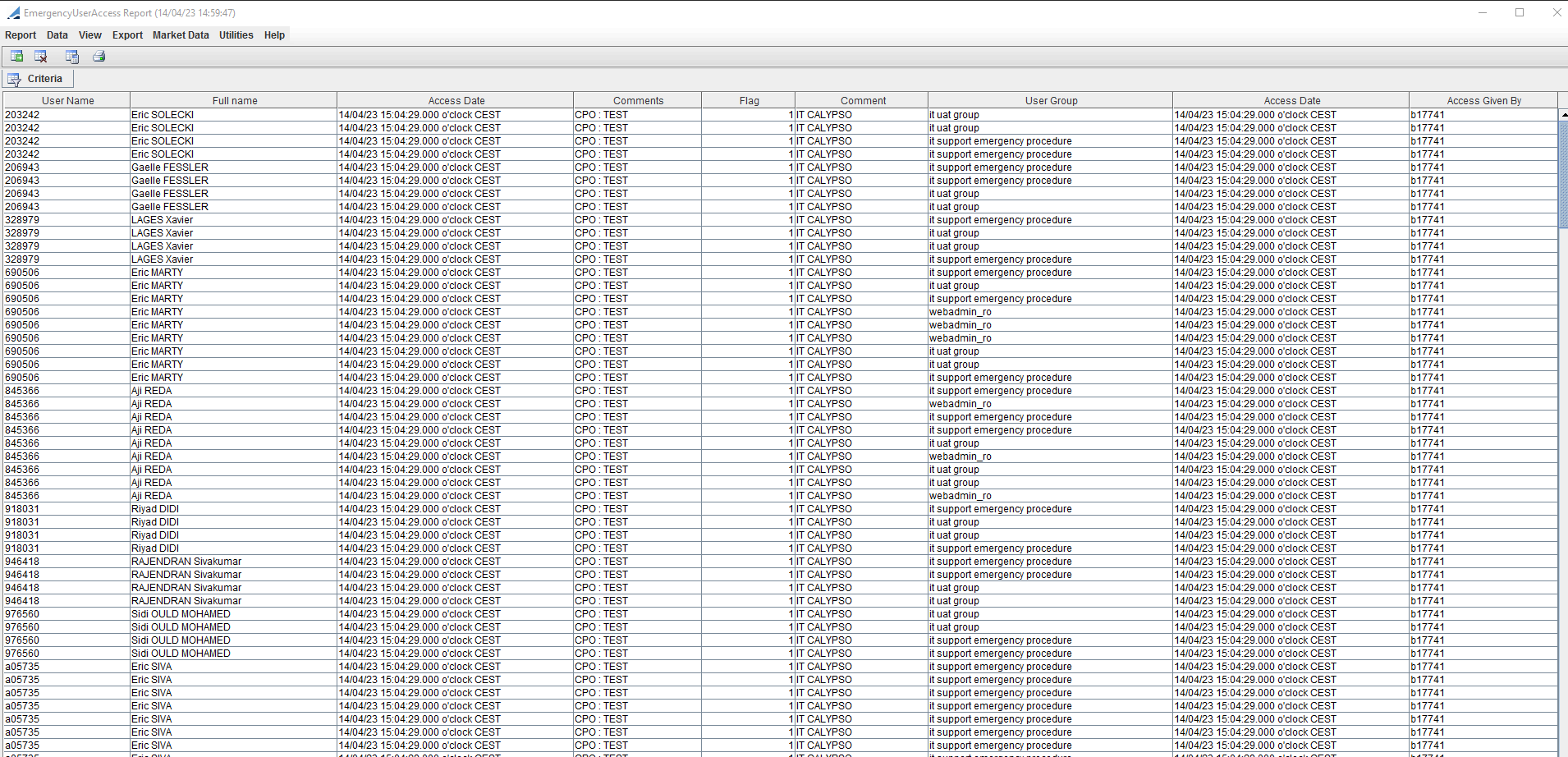


Figure 10 : REPORT CALYPSO

# CONCLUSION

Durant cette période chez BNPPARIBAS SECURITIES SERVICES, j'ai eu l'opportunité d'élargir mes compétences techniques en développement informatique, en particulier dans le domaine de la gestion des activités financières avec la solution Calypso. J'ai acquis des connaissances en matière de langages de programmation tels que Java, ainsi que des compétences en développement de logiciels et en test de qualité.

Cette période m'a également permis de me familiariser avec le monde professionnel et les méthodes de travail en entreprise. J'ai pu observer les processus de gestion de projet, de planification et de coordination en équipe, ainsi que les normes de qualité et les protocoles de sécurité en place chez BNPPARIBAS SECUTIES SERVICE.

Travailler au sein d'une équipe de cette taille m'a également permis d'expérimenter le travail collaboratif et la communication en milieu professionnel. J'ai pu observer comment les membres de l'équipe se sont réparti les tâches et se sont coordonnés pour atteindre les objectifs de projet fixés.

Dans l'ensemble, cette période a été une expérience très enrichissante pour moi. J'ai acquis de nouvelles compétences techniques, j'ai appris à travailler en équipe et j'ai pu observer le fonctionnement d'une entreprise de premier plan dans le secteur de la finance.