

Immeuble Apogée – 500 avenue Galilée
13100 Aix-en-Provence
04-42-90-03-97
Site: <https://www.smart-trade.net>

Rapport de stage d'Ingénieur de 3ème année

Automatisation des Tests de non-régression du logiciel (Java)

Présenté par :
Jiquan ZHONG

Tuteur académique : Nicole DOUMIT

Date de la soutenance : 29/09/2023

Tuteur d'entreprise : Nathan ORSONI

Durée du stage : 900 h (6 mois)

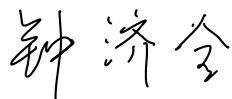
Institut supérieur d'électronique de Paris - ISEP

Campus Notre-Dame-des-Champs (NDC)
28, rue Notre-Dame-des-Champs, 75006 Paris
Campus Notre-Dame-de-Lorette (NDL)
10, rue de Vanves, 92130 Issy-les-Moulineaux

Je, soussigné Nathan Orsoni, maître de stage de l’élève Isep, Jiquan Zhong, atteste avoir pris connaissance du cahier des charges du rapport de stage Isep, avoir lu et évalué le présent rapport au regard du cahier des charges et des pratiques de mon employeur, et autorisé sa transmission à l’Isep.

Date: 22/09/2023

Signature de l’élève:



Signature du maître de stage:

22/09/2023

Orsoni Nathan



Remerciements

Je tiens tout d'abord à exprimer ma gratitude envers Josepha Besnard, qui a été responsable du recrutement et m'a donné l'opportunité de réaliser ce stage.

Ensuite, je souhaite remercier chaleureusement Nathan Orsoni, mon tuteur pendant toute la durée de mon stage. Sa patience et ses compétences m'ont permis d'apprendre de nombreux aspects du travail, depuis la langue française jusqu'aux logiciels utilisés dans l'entreprise, en passant par les techniques de développement et les cadres de projets. Nathan a toujours été là pour me guider et m'expliquer avec patience, même lorsque je rencontrais des difficultés à cause de ma maîtrise limitée du français.

Je tiens également à exprimer ma gratitude envers Julien Aymard, le responsable de l'équipe, ainsi qu'envers Christophe Hemon et Rene Somphanethabanesouk, membres de l'équipe d'automatisation des tests. Ils ont toujours été patients et prêts à me fournir des conseils lorsque j'avais besoin d'aide.

Je souhaite également remercier l'équipe voisine, l'équipe SEBI, qui développe les modules que j'ai testés. Chaque fois que j'avais des questions concernant les fonctionnalités, ils ont toujours été rapides à m'apporter leur aide.

Je tiens à exprimer ma gratitude envers Dang Dinh Nguyen, le stagiaire de l'équipe Sebi, avec qui j'ai travaillé en étroite collaboration sur de nombreux projets d'automatisation des tests. Je tiens également à remercier les autres stagiaires de SmartTrade avec qui j'ai partagé des moments joyeux et agréables.

Un grand merci à mes partenaires de tennis de table, Romain Colonna d'Istria, Elise Barnabé et encore Nathan Orsoni, avec qui j'ai passé d'agréables moments de détente pendant nos pauses.

Enfin, je tiens à exprimer ma gratitude envers tous les membres de SmartTrade qui ont contribué à créer une ambiance de travail exceptionnelle et qui ont travaillé ensemble pour développer ce logiciel exceptionnel.

Je suis extrêmement reconnaissant envers toutes les personnes mentionnées ci-dessus pour leur soutien, leur patience et leur collaboration précieuse tout au long de mon stage. Grâce à eux, mon expérience de stage a été enrichissante et inoubliable.

Table des matières

Remerciements	3
Table des matières	5
Résumé	7
Lexique	8
I. Introduction	9
II. Contexte du stage	10
1. Introduction au Marché FOREX	10
a. Fonctionnement du FOREX	10
b. Les acteurs du FOREX	10
c. L'évolution technologique et le FOREX	11
2. Historique, Activités et Structure de smartTrade Technologies	11
a. Historique et Activités de smartTrade Technologies	11
b. Clients et Reconnaissance	12
c. Structure Organisationnelle	12
d. Les Squads	13
e. Les Chapters	14
3. Le cadre technique SMS et les modules fonctionnels, SmartCC et SmartCC2	15
a. Les connecteurs	16
b. Le LALO	16
c. Le LOLD	17
d. Le cockpit	18
e. Les Modules	18
f. SmartCC et SmartCC2	18
g. Market Simulator	19
4. Framework d'automatisation	19
a. L'option classique: Selenium	19
b. Automation Framework: Une solution sur mesure	20
c. Cucumber: Une exploration initiale	21
5. Prérequis et Outils du Stage	21
a. Exigences de la pile technologique	21
b. Compétences linguistiques	22
c. Exigences en matière de logiciels de bureau	23
III. Présentation et analyse du travail effectué	25
1. Accueil et Intégration chez smartTrade	25
a. Les Formations	25
c. Les Défis de l'Intégration	27
2. Méthodologie du développement	28
a. Les fonctionnalités de SSI	28
3. Développement d'un Test d'automation	30
a. Préparation et Compréhension du Contexte	30

b. Compréhension du flow de Test	30
c. Configuration de la Fonctionnalité	31
d. Configuration de la Base de Données	31
e. Développement des Modules UI	31
f. Développement du Test Junit	32
g. Exécution et Analyse des Résultats	32
4. Conception du Flow de Test pour SSI	32
a. Classification de l'Interface Utilisateur de SSI	32
b. Approche de Conception du Flow	33
5. Le plan du stage	34
IV. Analyse du travail effectué	36
1. Rappel du framework d'automation	36
2. Analyse du flow pour SSI	37
a. Flow de SSI dans SmartCC2	37
b. Flow de SSI dans Cockpit	40
3. Développement de Test de JUnit	43
a. Importance des Test Suites :	43
b. Processus de création	43
c. Exemple concret	44
d. Exécution de Test Suite sur Bamboo	45
4. Développement de Test de FIX	47
a. Rappel de FIX et l'intégration de FIX chez smartTrade	47
b. Processus de développement des tests FIX	48
c. Analyse et retours d'expérience	50
V. Analyses managériales	52
1. RSE	52
a. Introduction	52
b. Engagement envers les Employés	52
c. Actions Caritatives	52
d. Protection de l'Environnement	53
e. Initiatives Communautaires	54
2. Management de l'innovation	55
a. Définition et importance du management de l'innovation	55
b. Stratégie et objectifs de smartTrade dans le domaine de l'innovation	55
c. Principales mesures et pratiques en matière d'innovation chez smartTrade	55
e. Analyse du management de l'innovation chez smartTrade	56
f. Points clés et conclusions des entretiens avec les cadres supérieurs et les départements concernés	56
g. Recommandations pour améliorer la stratégie et les pratiques d'innovation	57
VI. Conclusion	58
VII. Référence	59

Résumé

Ce résumé porte sur mon stage de fin d'études au sein de l'entreprise smartTrade Technologies à Aix-En-Provence, au sein du département de recherche et développement. L'entreprise se spécialise dans la création, la fourniture et la maintenance de solutions logicielles dans le domaine financier à l'échelle mondiale. Mon rôle au sein de l'entreprise consiste à travailler sur le framework interne d'automatisation de tests basé sur Selenium. Initié il y a quatre ans, ce framework vise à tester automatiquement le logiciel du point de vue de l'utilisateur afin de minimiser les risques de régression.

Ma mission a été centrée sur les tests liés aux Instructions de Règlement Standard (SSI), qui sont des instructions de règlement pour les entités légales, où les informations clés restent identiques d'un règlement en espèces à un autre (c'est-à-dire la banque, le numéro de compte et le nom du compte), seuls le montant et la date de valeur sont modifiés. J'ai également été responsable des tests concernant les SSP avec le protocole FIX, où les SSP sont utilisés dans le SMS pour envoyer plusieurs ordres en une seule fois pour des transactions impliquant la même devise. J'ai ainsi enrichi le framework en y ajoutant les ressources et le code nécessaires pour manipuler les SSI et les SSP. Ces ajouts m'ont permis de concevoir des scénarios de test garantissant l'absence de régression pour ces nouvelles fonctionnalités.

En conclusion, ce stage m'a offert l'opportunité de découvrir la vie en entreprise, à la fois en présentiel et à distance, ainsi que l'environnement dynamique qu'une entreprise de taille raisonnable mais ambitieuse comme smartTrade peut offrir.

Mots-Clés:

- Finance
- Selenium
- Automatisation
- Tests
- Java

Lexique

DOM	Connecte les pages web aux scripts et aux codes en représentant la structure du document. Le DOM représente le document sous la forme d'arbre logique.
GUI	Graphical user interface
standalone	Chez smartTrade c'est un produit SMS complet, lancé localement sur une machine dans le but d'effectuer des tests en simulant le marché
SMS	Un fournisseur mondial de premier plan de plates-formes de trading électronique multi-actifs et de paiements
JMX	Permet d'appeler des méthodes à distance sans interface graphique
LALO	Liquidity Aggregator Liquidity Orchestrator
LOLD	Liquidity Orchestrator Liquidity Distributor
MS	Market Simulator
RFQ	Request For Quote
VT	Voice Trading (module pour les vendeurs)
SQL	Structured Query Language
SSP	Single Spot Portfolio
SSI	donne le sigle Des instructions de règlement pour les entités légales, avec clés inchangées, utilisées dans les transactions financières
Smart Analytics	La plateforme d'analyse commerciale de SmartTrade

I. Introduction

Au cours des derniers mois, j'ai eu l'opportunité d'effectuer un stage au sein de smartTrade Technologies, une entreprise reconnue dans le domaine du logiciel de finance. Mon rôle au sein de cette entreprise était celui d'un développeur de tests automatisés en Java. Cette position m'a permis de travailler étroitement avec des outils et des technologies avancés, notamment Selenium, pour développer des tests automatisés pour certaines parties spécifiques du logiciel, en particulier les sections SSP (Single Spot Portfolio) et SSI (Standard Settlement Instructions).

À ce stade de mon parcours professionnel, je me trouve dans une phase d'apprentissage et d'acquisition de compétences pratiques. Mon statut de stagiaire m'a offert une perspective unique, me permettant de comprendre non seulement les aspects techniques de mon rôle, mais aussi comment ces aspects s'intègrent dans le fonctionnement global de l'entreprise.

La rédaction de ce rapport est une étape essentielle pour documenter et réfléchir sur mon expérience. Il vise à présenter les tâches que j'ai accomplies, les défis que j'ai rencontrés et les compétences que j'ai acquises. De plus, ce rapport offre une opportunité de mettre en lumière la valeur ajoutée de mon stage, tant pour mon développement personnel que pour la contribution à smartTrade Technologies.

II. Contexte du stage

1. Introduction au Marché FOREX

Le marché des changes, plus communément appelé FOREX (pour "foreign exchange"), est l'endroit où les devises sont échangées. C'est un marché mondial, décentralisé, où s'échangent toutes les monnaies du monde. Avec un volume de transactions quotidien dépassant les 6 000 milliards de dollars, le FOREX est le plus grand marché financier au monde, surpassant largement les marchés boursiers mondiaux.

a. Fonctionnement du FOREX

Sur le FOREX, les devises sont négociées par paires. Par exemple, EUR/USD représente le taux de change entre l'euro et le dollar américain. Si cette paire est cotée à 1,20, cela signifie qu'1 euro vaut 1,20 dollar.

Prenons un exemple simple pour illustrer le fonctionnement du FOREX. Supposons que vous prévoyez de voyager aux États-Unis et que vous alliez à votre banque pour échanger 1000 euros contre des dollars. Si le taux de change est de 1,20, cela signifie que pour 1 euro, vous obtiendrez 1,20 dollar. Ainsi, pour vos 1000 euros, vous recevrez 1200 dollars. Si, à votre retour, le taux de change a changé et est maintenant de 1,25, lorsque vous échangerez vos dollars restants contre des euros, vous obtiendrez plus d'euros en retour que ce que vous aviez initialement échangé. C'est ainsi que les traders sur le FOREX cherchent à réaliser des profits, en anticipant les mouvements des taux de change.

b. Les acteurs du FOREX

Le marché FOREX est composé d'une variété d'acteurs, allant des banques centrales aux traders individuels. Les banques centrales, comme la Banque centrale européenne ou la Réserve fédérale américaine, jouent un rôle majeur dans la détermination des taux de change. Les grandes banques internationales servent également de principaux intermédiaires pour les transactions FOREX et déterminent le taux de change en fonction de l'offre et de la demande.

Les entreprises multinationales utilisent le FOREX pour payer les biens et services achetés à l'étranger. Les fonds d'investissement, les courtiers et même les particuliers participent également activement au trading sur le FOREX.

c. L'évolution technologique et le FOREX

Avec l'avènement de l'ère numérique, le marché FOREX a connu une transformation radicale. Les plateformes de trading en ligne ont rendu le marché accessible à un plus grand nombre de personnes, permettant même aux traders individuels de participer activement au trading de devises depuis le confort de leur domicile. Ces plateformes offrent une multitude d'outils pour aider les traders à prendre des décisions éclairées, allant des actualités en temps réel aux analyses techniques avancées.

Des entreprises spécialisées, comme smartTrade Technologies, ont vu le jour pour répondre à la demande croissante de solutions technologiques innovantes dans le domaine du trading FOREX. Ces entreprises développent des logiciels qui non seulement facilitent le trading, mais offrent également une multitude d'outils pour aider les traders à analyser le marché, gérer leurs risques et exécuter leurs stratégies de trading.

2. Historique, Activités et Structure de smartTrade Technologies

a. Historique et Activités de smartTrade Technologies

Fondée en 1999 par David Vincent, actuel Président Directeur Général, et Harry Gozlan, ancien trader, smartTrade Technologies s'est imposée comme une référence dans le secteur de l'Asset Liquidity Management. Spécialisée dans le développement de logiciels pour le trading, la finance de marché et la gestion du risque, l'entreprise a su évoluer avec les besoins du marché.

À ses débuts, smartTrade se concentrat sur la fourniture de briques logicielles pour le domaine financier, destinées à s'intégrer dans divers systèmes. Cependant, en

2003, une réorientation stratégique a conduit l'entreprise à développer une solution globale hébergée sur sa propre infrastructure.

Aujourd'hui, smartTrade adapte ses logiciels aux besoins spécifiques de chaque client, ciblant principalement les acteurs financiers tels que les banques, les courtiers et les gestionnaires d'actifs. Ses produits sont des solutions de négociation flexibles, couvrant un large éventail d'actifs financiers, de la monnaie aux crypto-monnaies en passant par les produits dérivés. Ces solutions offrent notamment l'accès à plus de 130 fournisseurs de liquidité, l'agrégation des données du marché, la gestion des ordres, la tarification et bien plus encore.

b. Clients et Reconnaissance

La réputation de smartTrade n'est plus à faire. Parmi ses clients les plus prestigieux figurent des banques de renom telles que HSBC, Bank Of China et Leumi. Cette excellence a été reconnue à plusieurs reprises, comme en témoigne le prix de "meilleure solution de gestion de services dans le domaine du trading FX" obtenu lors des TradingTech Insight Awards 2021.

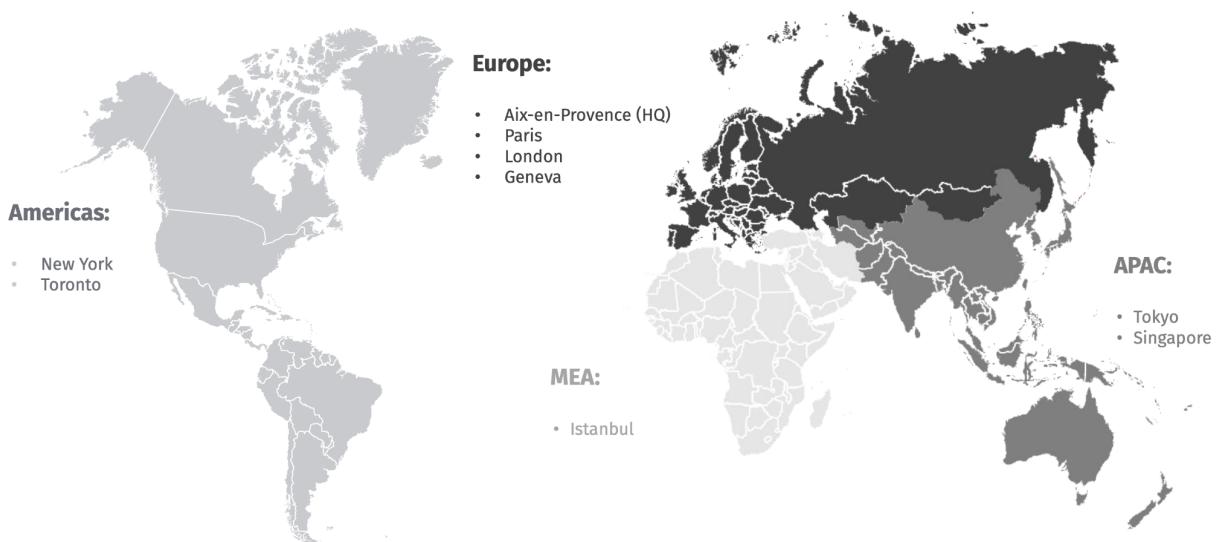


Clients de smartTrade Technology

c. Structure Organisationnelle

L'organisation de smartTrade est à la fois dynamique et adaptative. Bien que l'entreprise compte plusieurs centaines de collaborateurs, principalement basés à Aix-en-Provence, elle a su conserver une taille humaine. Les équipes sont réparties entre le développement, les tests, l'administratif, la stratégie d'entreprise et le support

client. Pour répondre aux besoins de ses clients internationaux, smartTrade a également établi des succursales à Tokyo, Londres, New York et Singapour.



bureaux de smartTrade Technology

L'organisation interne est structurée autour de plusieurs équipes. Ces équipes couvrent un large éventail de fonctions, de la recherche et développement à la gestion des infrastructures, en passant par le recrutement et la sécurité des locaux.

d. Les Squads

Au sein de SmartTrade, la structure des Squads joue un rôle essentiel dans le développement et l'innovation. Chaque Squad est responsable d'une partie spécifique du carnet de produits. Voici quelques points clés concernant leur fonctionnement :

Autonomie : Chaque Squad possède et gère les fonctionnalités qu'il développe. L'équipe travaille de manière autonome, et les décisions sont prises localement, minimisant ainsi les interventions externes superflues.

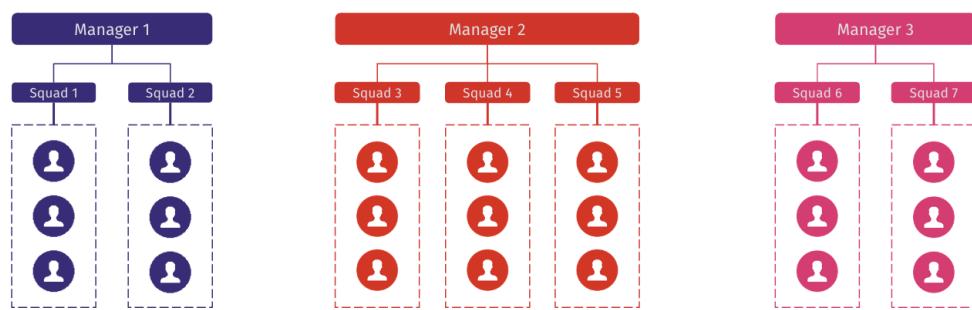
Expertise : Les Squads fournissent une expertise et des orientations sur leur ensemble spécifique de fonctionnalités, s'occupant des demandes d'évolution futures et assurant le support des fonctionnalités qu'ils ont mises en œuvre.

Contrôle : Bien que les Squads bénéficient d'une grande autonomie, celle-ci est équilibrée par des processus de contrôle rigoureux, tels que la revue par

les architectes, les tests automatiques, la validation des performances et la création de maquettes UX.

Compétences : Les membres de chaque Squad peuvent posséder diverses compétences, allant du développement backend et frontend à l'assurance qualité, en passant par des outils spécifiques tels que SmartCC et Smart Analytics.

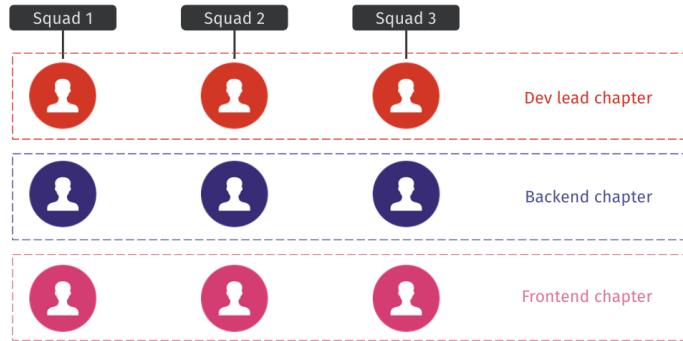
Coordination : Les managers d'équipe assurent une coordination interéquipe, abordant des sujets ad-hoc et servant de point de référence pour l'écosystème de l'équipe SmartTrade.



organigramme de squad

e. Les Chapters

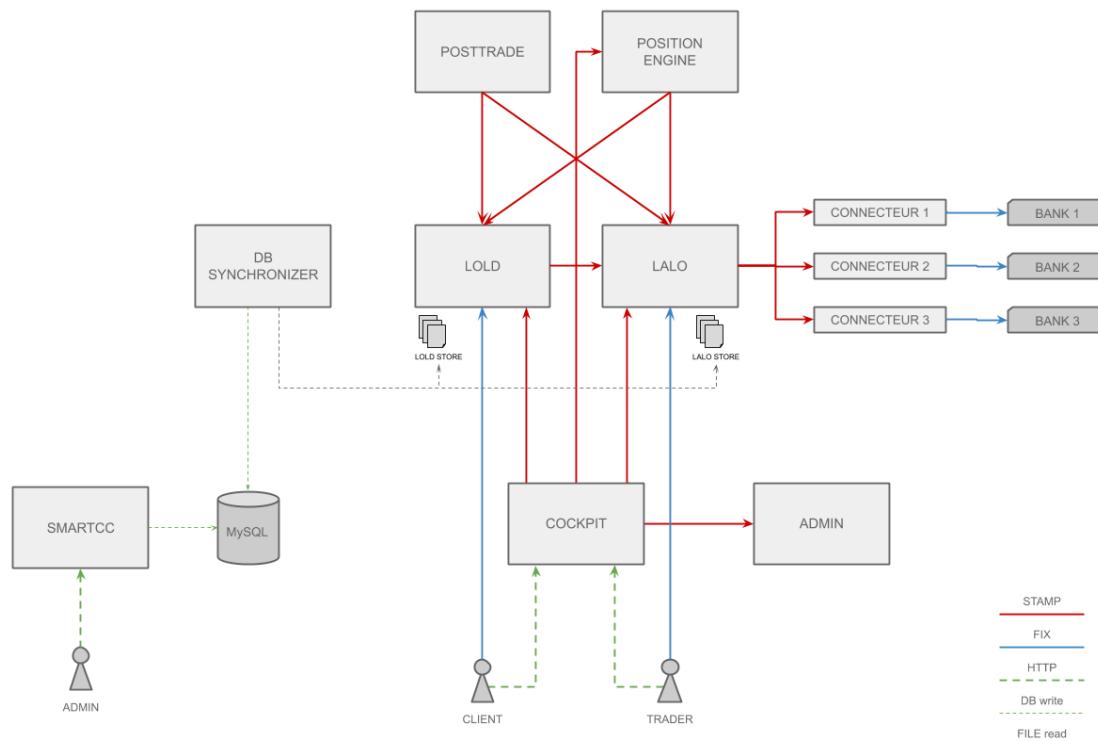
Les Chapters représentent les rôles spécialisés, les personnes et les services nécessaires au succès du Squad. Ils couvrent des domaines tels que l'architecture, la conception et la performance. Leur objectif principal est de fournir un leadership éclairé dans leur domaine d'expertise, d'examiner les métriques, d'identifier les points chauds, etc. Ils offrent également une plateforme pour partager les défis, évoluer les meilleures pratiques et demander l'assistance ou les conseils de pairs.



structure d'équipe

3. Le cadre technique SMS et les modules fonctionnels, SmartCC et SmartCC2

La suite logicielle LFX, ou SMS, est le cœur battant de la technologie smartTrade. Elle représente une fusion de l'expertise technique et de la connaissance approfondie du marché financier. Dans un monde où la rapidité, la stabilité et la précision sont essentielles, cette suite logicielle est conçue pour offrir une performance inégalée.



a. Les connecteurs

La communication est le pilier de toute infrastructure technologique. Dans le monde du trading électronique, elle est d'autant plus cruciale. Les connecteurs de smartTrade sont conçus pour assurer une communication fluide et sans faille.

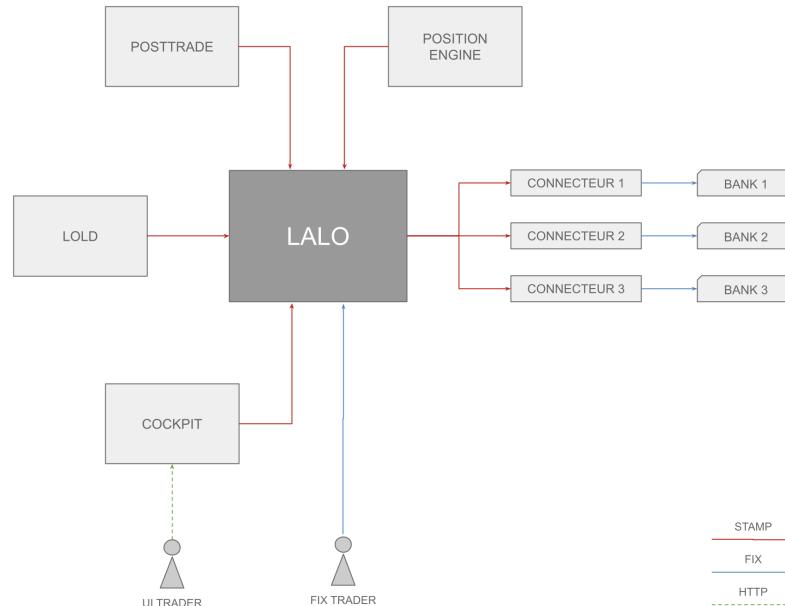
- Protocole FIX : Adopté comme une norme internationale, le protocole FIX est le langage universel du trading électronique. Il garantit que les transactions sont effectuées rapidement, efficacement et sans erreur. Il est essentiel pour assurer une communication standardisée entre différentes plateformes de trading. Le protocole FIX a été développé dans les années 1990 et est devenu le standard de facto pour le trading électronique. Il est utilisé par les bourses, les courtiers, les institutions financières et les traders du monde entier.
- Protocole STAMP : En reconnaissant la nécessité d'une personnalisation, smartTrade a développé le protocole STAMP. Bien qu'il s'inspire du protocole FIX, il offre une flexibilité supplémentaire, permettant une adaptation aux besoins spécifiques des clients de smartTrade grâce à l'ajout d'informations supplémentaires dans les messages. C'est une innovation qui distingue smartTrade de ses concurrents. Le protocole STAMP a été développé pour répondre aux besoins spécifiques des clients de smartTrade, offrant une personnalisation qui ne sont pas possibles avec le protocole FIX standard.

b. Le LALO

L'Aggregateur et Orchestrateur de Liquidités (LALO) est bien plus qu'un simple intermédiaire. Il est le cerveau qui analyse, trie et distribue les informations.

- Flux Direct ESP : Ce flux en streaming garantit que tous les abonnés voient les mêmes prix en temps réel, créant ainsi un environnement de trading équitable. C'est une technologie essentielle pour garantir la transparence et l'équité sur le marché. Le flux ESP est essentiel pour les traders qui ont besoin d'accéder aux prix en temps réel, garantissant qu'ils peuvent prendre des décisions éclairées basées sur les informations les plus récentes.
- Flux RFQ : Offrant une approche plus personnalisée, le flux RFQ permet une négociation directe, où les prix peuvent être discutés et convenus. C'est une

fonctionnalité qui offre une flexibilité supplémentaire aux traders, leur permettant d'obtenir le meilleur prix possible. Le flux RFQ est particulièrement utile pour les traders qui négocient de gros volumes ou qui ont des besoins spécifiques qui ne peuvent pas être satisfaits par le flux ESP standard.



Architecture LALO simplifiée

c. Le LOLD

Le LOLD est le “gardien” des prix. Il prend les informations brutes et les transforme en données précieuses pour les clients. Chaque étape, qu'il s'agisse de l'ajout de marges ou de la conversion de devises, est effectuée avec une précision chirurgicale. Dans un monde où chaque centime compte, le LOLD garantit que les clients obtiennent toujours le meilleur prix.

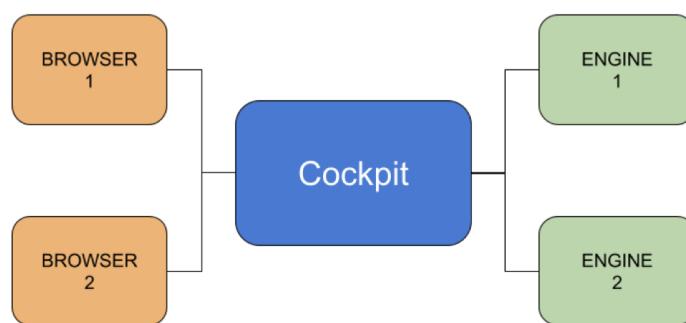
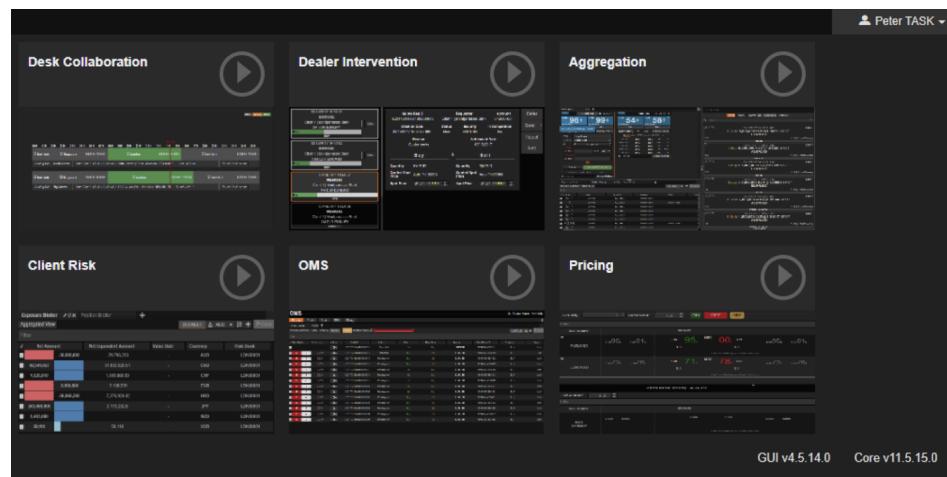


Schéma simplifié du cockpit

d. Le cockpit

Le cockpit est la vitrine de LFX. C'est par cette interface que les utilisateurs interagissent avec le système.

- **Pony** : Ce framework maison, opensource est le résultat d'années de développement et d'optimisation. Il garantit que l'interface utilisateur est non seulement esthétiquement agréable, mais aussi fonctionnellement robuste. C'est une preuve supplémentaire de l'engagement de smartTrade envers l'innovation.



UI modules de Cockpit

e. Les Modules

Chaque module est conçu pour répondre à un besoin spécifique du marché. Que ce soit pour le trading institutionnel ou pour la négociation vocale, chaque module est le fruit d'une recherche approfondie, de nombreux échanges avec les clients et d'un développement méticuleux. Ces modules garantissent que chaque utilisateur, quel que soit son rôle ou ses besoins, dispose des outils nécessaires pour réussir.

f. SmartCC et SmartCC2

Ces deux logiciels sont responsables des configurations. Les administrateurs ont la possibilité de configurer directement le logiciel et de surveiller l'état du système sur ces deux plates-formes.

- **JMX** : Cette technologie permet une mise à jour en temps réel, garantissant que les modifications sont immédiatement reflétées. Dans un monde où les

Marchés évoluent rapidement, cette capacité à s'adapter en temps réel est essentielle.

g. Market Simulator

Dans le monde du trading, la préparation est la clé. Le simulateur de marché permet aux clients et aux développeurs de tester leurs stratégies, de s'entraîner et de se préparer pour le monde réel du trading. C'est un outil essentiel pour garantir que les traders sont toujours prêts à affronter les défis du marché.

4. Framework d'automatisation

L'automatisation des tests est devenue une pierre angulaire du développement logiciel moderne. Dans un monde où les cycles de développement sont de plus en plus courts et où la demande de qualité est toujours élevée, l'automatisation des tests est essentielle. Pendant mon stage, j'ai été chargé d'automatiser les tests de non régression pour les SSP et SSI, deux composants essentiels de notre infrastructure.

a. L'option classique: Selenium

Selenium, introduit en 2004, est devenu le standard de facto pour l'automatisation des tests web. Il offre une interface pour interagir avec les navigateurs, simulant les actions des utilisateurs réels.

La force de Selenium réside dans sa polyvalence. Il prend en charge de nombreux navigateurs, des plus populaires aux moins connus. De plus, sa compatibilité avec Java, le langage de programmation largement utilisé, le rendait idéal pour notre projet.



Cependant, Selenium n'est pas sans ses défis. L'un des plus grands est la localisation des éléments sur une page web. XPath, un langage utilisé pour naviguer à travers les éléments et attributs dans un document XML, est souvent utilisé à cette fin. Bien que puissant, XPath peut être complexe, nécessitant une expertise pour être utilisé efficacement.

b. Automation Framework: Une solution sur mesure

Face aux défis rencontrés avec Selenium, l'équipe a décidé de développer un framework d'automatisation sur mesure. En avril 2020, face aux défis croissants posés par l'automatisation des tests, l'équipe a pris la décision audacieuse de créer une surcouche pour Selenium. Cette surcouche, baptisée Automation Framework, avait pour objectif de simplifier le processus d'écriture et de maintenance des tests.

L'Automation Framework a été conçu avec plusieurs principes clés en tête:

- Clarté: Chaque méthode dans le framework a un nom explicite, rendant son utilisation intuitive.
- Simplicité: Les utilisateurs n'ont pas besoin de fournir des adresses ou des identifiants complexes pour utiliser le framework.
- Modularité: Le framework est organisé en modules, ce qui facilite la maintenance et l'extension.

Pour utiliser le framework, les testeurs doivent exécuter le script "buildDev.bat". Ce script prépare l'environnement de test en récupérant une version standalone de l'application depuis Artifactory. Une fois l'environnement prêt, les tests peuvent être exécutés. Le framework génère deux applications Java, qui gèrent respectivement la configuration et l'exécution des tests.

Au cœur de l'Automation Framework se trouvent les Flows, des classes qui définissent des séquences d'actions pour tester différentes parties de l'application. Les Annotations, quant à elles, permettent d'ajouter des métadonnées aux tests, offrant ainsi une flexibilité supplémentaire lors de l'exécution des tests.

L'Automation Framework n'est pas seulement un outil pour interagir avec l'interface utilisateur. Grâce à l'intégration avec JMX, le framework peut également interagir avec des éléments non graphiques, offrant ainsi une couverture de test complète. Cela signifie que non seulement nous pouvons tester l'interface utilisateur, mais aussi les services et les processus en arrière-plan, garantissant ainsi une qualité de bout en bout.

c. Cucumber: Une exploration initiale

Cucumber est un outil de test qui prend en charge la programmation pilotée par le comportement (BDD). Il permet aux équipes de définir des scénarios de test dans un langage naturel, rendant les tests compréhensibles même pour les non-développeurs. L'attrait de Cucumber réside dans sa capacité à encourager la collaboration entre les développeurs, les testeurs et les parties prenantes non techniques. Avec son langage Gherkin, il est possible de définir des scénarios de test dans un langage simple et compréhensible.

Cependant, malgré ses avantages, Cucumber a présenté des défis pour notre équipe. La principale difficulté résidait dans la connexion des scénarios définis dans Gherkin aux actions réelles à effectuer lors des tests. Cette complexité a finalement conduit à l'abandon de Cucumber en faveur de l'Automation Framework.

Avec l'Automation Framework en place, l'équipe est bien équipée pour relever les défis futurs de l'automatisation des tests. Les tests de non régression pour SSP et SSI peuvent maintenant être écrits et exécutés rapidement et efficacement, garantissant que toute modification du code n'introduit pas de nouveaux bugs.

5. Prérequis et Outils du Stage

a. Exigences de la pile technologique

- Java: Java, un langage de programmation orienté objet, est au cœur de nombreux projets technologiques. Sa polyvalence et sa robustesse en font un choix incontournable pour le développement d'applications d'entreprise. Dans le cadre de mon stage, Java est omniprésent, servant de fondement à la majorité des développements et des tests.

La maîtrise de Java est donc essentielle pour naviguer dans le code source, écrire des tests automatisés, interagir avec des bases de données et communiquer avec d'autres services. Les bibliothèques et frameworks associés à Java, tels que JUnit pour les tests, sont également des outils clés pour accomplir efficacement les tâches du stage.



- Git: Git est plus qu'un simple système de contrôle de version : c'est un outil essentiel pour collaborer sur des projets de développement. Dans un environnement où plusieurs développeurs apportent des modifications simultanées, Git facilite la gestion des versions, le suivi des modifications et la résolution des conflits.

La maîtrise de Git est donc cruciale pour s'assurer que le code est toujours à jour, que les modifications sont correctement documentées et que la collaboration entre les membres de l'équipe est fluide et efficace.

- Selenium: Selenium est un outil d'automatisation des actions sur un navigateur web, permettant de simuler des actions utilisateur pour tester des applications web. Dans le cadre des tests automatisés, Selenium joue un rôle central pour garantir que l'application fonctionne comme prévu, en simulant des interactions réelles d'utilisateurs.

La connaissance approfondie de Selenium est donc essentielle pour créer, exécuter et maintenir des tests automatisés, garantissant ainsi la qualité et la fiabilité de l'application.

b. Compétences linguistiques

La maîtrise des langues est essentielle dans un environnement professionnel international.

- Français: Le français est la langue principale de communication au sein de l'entreprise. Que ce soit pour des réunions, des discussions avec des collègues ou pour comprendre les exigences du projet, une excellente maîtrise du français est indispensable.

- Anglais: L'anglais est essentiel pour accéder à la documentation technique, aux guides et aux spécifications. La capacité à lire, comprendre et éventuellement rédiger en anglais est donc cruciale pour garantir que les exigences techniques et fonctionnelles sont correctement interprétées et mises en œuvre.

c. Exigences en matière de logiciels de bureau

- Suite Google: La suite Google, comprenant Google Docs, Sheets, Slides, et d'autres outils, est un ensemble d'outils collaboratifs en ligne. Ces outils permettent de collaborer en temps réel, de suivre les modifications et de partager des documents avec des équipes ou des clients.



- **Suite logicielle Atlassian**

JIRA : Cet outil est au cœur de la gestion de projet. Il permet de suivre les tâches, les bugs, les demandes d'amélioration et bien plus encore. Chaque ticket dans JIRA peut être assigné, suivi et mis à jour avec des commentaires, des pièces jointes et des liens vers d'autres tickets ou documentation. La capacité à utiliser JIRA efficacement est essentielle pour suivre l'avancement du projet, communiquer avec l'équipe et s'assurer que toutes les tâches sont correctement documentées et suivies.



Confluence : Confluence sert de référentiel central pour toute la documentation liée au projet. Il offre un espace pour stocker, partager et collaborer sur des documents. Contrairement à d'autres plateformes, Confluence est conçu pour la collaboration,

permettant aux équipes de créer des pages, d'ajouter des commentaires, de suivre les modifications et de lier des documents pertinents. La capacité à naviguer, contribuer et utiliser efficacement Confluence est essentielle pour accéder à l'information, documenter les processus et collaborer avec l'équipe.

Bamboo: Au cœur de notre processus d'intégration et de déploiement continu chez smartTrade, nous utilisons Atlassian Bamboo. Cet outil, conçu par Atlassian, est essentiel pour garantir une livraison de logiciel rapide et fiable.



Caractéristiques principales de Bamboo :

- Intégration Continue (CI) : Bamboo automatise la compilation, le test et la validation du code, assurant ainsi une qualité constante tout au long du cycle de développement.
- Déploiement Continu (CD) : Après validation, Bamboo facilite le déploiement du code dans un environnement de production ou de pré-production, garantissant une mise en production rapide et sans faille.
- Intégration avec d'autres outils Atlassian : La force de Bamboo réside également dans son intégration transparente avec d'autres produits Atlassian, tels que JIRA pour la gestion des tâches et Bitbucket pour la gestion du code source.

Chez smartTrade, nous avons mis en place un serveur Bamboo pour répondre à nos besoins de tests intensifs. Cependant, il est important de noter que notre expansion ne se limite pas uniquement à l'augmentation du nombre de machines de test. Cette infrastructure robuste, combinée à l'écosystème Atlassian, nous permet d'assurer la fiabilité et la performance de nos solutions logicielles.

III.Présentation et analyse du travail effectué

1. Accueil et Intégration chez smartTrade

Dès mon arrivée chez smartTrade, j'ai été enveloppé dans une atmosphère à la fois professionnelle et accueillante. La société avait minutieusement organisé une série de formations pour faciliter l'intégration des nouveaux stagiaires, dont j'étais l'un des membres.

a. Les Formations

L'approche de formation adoptée par smartTrade était bifurquée, se divisant en deux volets principaux : la formation par conférence et la formation en ligne.

Formation par Conférence

Les premières semaines ont été marquées par des sessions de formation en conférence. Ces sessions, animées par des collègues expérimentés, couvraient une gamme de sujets essentiels pour comprendre l'écosystème de smartTrade.

- Test Framework : Une introduction approfondie au cadre de test utilisé par smartTrade, essentiel pour comprendre comment l'entreprise garantit la qualité de ses produits.
- Présentation du Produit et des Affaires : Une vue d'ensemble des solutions offertes par smartTrade, leur positionnement sur le marché et leur valeur ajoutée pour les clients.
- Conception des Services : Une plongée dans la manière dont les services sont conçus, mettant en lumière la philosophie de conception de l'entreprise.
- Modèle de Données et Flux de Trading : Comprendre comment les données sont structurées et comment elles circulent à travers les différents systèmes était crucial.
- Connecteurs, IBus + ScenarioRunner : Ces outils et plateformes sont au cœur de l'infrastructure de smartTrade, et une compréhension approfondie de leur fonctionnement était essentielle.
- Moteurs de Trading et de Données de Marché : Une exploration des moteurs qui alimentent les plateformes de smartTrade, offrant une vue d'ensemble de la manière dont les transactions sont traitées.

- Aperçu de SMS + LALO, Pony et Cockpit : Ces systèmes spécifiques sont des éléments clés de l'offre de smartTrade, et leur compréhension était essentielle pour mon rôle.

Formation en Ligne

En complément des sessions de formation en conférence, j'ai été orienté vers des modules de formation en ligne. Ces modules étaient conçus pour offrir une flexibilité dans l'apprentissage, permettant d'approfondir certains sujets à mon propre rythme.

- Basics of ST platform : Une introduction complète à la plateforme phare de smartTrade, couvrant ses fonctionnalités clés et ses avantages.
- LFX and FIX : Une plongée dans la structure de LFX et le protocole FIX, essentielle pour comprendre comment les transactions sont traitées.
- Sensibilisation à la Sécurité : Dans le monde numérique d'aujourd'hui, la sécurité est primordiale. Ces modules m'ont formé aux meilleures pratiques pour garantir la sécurité des données et des transactions.

b. La Cohésion d'Équipe

L'intégration ne se limitait pas seulement à la formation technique. Pour renforcer l'esprit d'équipe et faciliter l'intégration sociale des nouveaux stagiaires, smartTrade a organisé une journée de team building dans un parc forestier. C'était une expérience mémorable. Divisés en deux équipes, nous avons participé à des jeux compétitifs, cherchant à collaborer avec nos coéquipiers pour surpasser l'équipe adverse. La participation active des cadres supérieurs de l'entreprise a renforcé le sentiment d'appartenance et a montré l'importance que smartTrade accorde à chaque membre de son équipe.



La photo de groupe de la sortie au parc

c. Les Défis de l'Intégration

Barrière Linguistique

L'un des défis majeurs que j'ai rencontrés a été la barrière linguistique. Bien que la plupart des employés parlent anglais, la communication quotidienne se faisait principalement en français. C'était à la fois un défi et une opportunité pour moi. J'ai dû m'immerger complètement dans la langue, non seulement pour suivre les discussions techniques, mais aussi pour participer aux conversations informelles.

Adaptation au Clavier Français

Un autre défi inattendu a été l'adaptation au clavier français. Après avoir utilisé un clavier américain pendant plus de vingt ans, le passage au clavier français a été une expérience déroutante. Cependant, avec le temps et la pratique, j'ai réussi à m'adapter.

Transition vers de Nouveaux Outils et Systèmes

En tant que développeur habitué à utiliser IntelliJ IDEA sur MacOS, la transition vers Eclipse sur Windows a été un autre défi. J'ai dû réapprendre plusieurs fonctionnalités et m'adapter à un nouvel environnement de développement.

2. Méthodologie du développement

Au cours de mon stage chez smartTrade, mes principales responsabilités étaient axées sur les tests. Plus précisément, j'étais chargé de mener à bien les tests des messages FIX du SSP et de développer la majeure partie des tests pour le SSI. Il est important de souligner que, bien que ces deux tâches aient été cruciales, elles présentaient des niveaux de complexité différents. Pour le SSP, la majorité du code avait déjà été rédigée, ne nécessitant de ma part que des ajustements mineurs. En revanche, pour le SSI, j'ai été le principal acteur de la conception et du développement des tests, ce qui m'a offert un défi bien plus conséquent.

C'est pour cette raison que je souhaite me concentrer principalement sur la présentation du processus de développement des tests pour le SSI dans les sections suivantes. Cette tâche, étant entièrement sous ma responsabilité, a été à la fois représentative de mon expérience et riche en défis. Elle m'a permis d'appliquer une méthodologie rigoureuse et de réfléchir en profondeur à chaque étape du processus.

Avant de plonger dans les détails techniques et méthodologiques du développement des tests pour SSI, il est essentiel de comprendre pleinement les scénarios d'utilisation et les spécificités fonctionnelles de cette fonctionnalité.

a. Les fonctionnalités de SSI

Qu'est-ce que le SSI ?

SSI, ou Standard Settlement Instructions, est un ensemble d'instructions destiné à guider la banque sur la manière de régler un commerce. En d'autres termes, il s'agit d'une feuille de route détaillée qui indique à la banque comment et où transférer les fonds une fois qu'une transaction a été effectuée. Chaque SSI est unique et est défini par un identifiant spécifique pour chaque compte, pour une monnaie donnée et pour une direction donnée (Delivery ou Funding).

Comment fonctionne le SSI ?

Les SSI sont transmis au système back office de la banque via l'API FIX PostTrade. Leur rôle principal est de faciliter le transfert de fonds. Il est à noter que ces instructions peuvent être définies de deux manières principales:

- Via smartCC2 : SmartCC2, étant le module administrateur du logiciel SMS, offre la possibilité de définir manuellement les SSI. Cela donne aux administrateurs un contrôle direct et une flexibilité pour définir ou modifier les instructions selon les besoins.

CREATE SSI	
ID	*
Enter an ID	
Description	*
Enter a Description	
Account	*
Select an Account	Select a Currency
Direction	*
<input type="checkbox"/> FUNDING	<input type="checkbox"/> DELIVERY
<input checked="" type="radio"/> FALSE	<input type="radio"/> TRUE
Reset Save	

GUI de création SSI sur SmartCC2

- Via l'API FIX : Pour les organisations qui disposent déjà d'un ensemble d'instructions ou qui souhaitent automatiser le processus, les SSI peuvent être fournies par une source externe via l'API FIX.

Intégration avec le module "Institutional Trading"

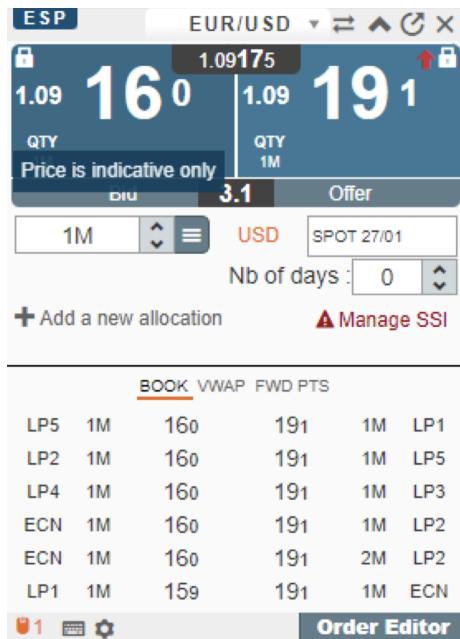
Selon "Institutional Trading User Guide", une fois qu'un compte a été défini avec un SSI, il est impératif de spécifier ce SSI lors de la réalisation d'une transaction.

STANDARD SETTLEMENT INSTRUCTIONS					
Summary Workflow ESP	CCY Pair EUR/USD	Traded CCY USD	Tenor/Value Date SPOT 27/01	Direction BOTH	
Instructions					
Account Name 001001-001	Quantity 1M	Pay CCY EUR	Pay SSI KEY 001001-001 EUR B...	Rec. CCY USD	Receive SSI KEY 001001-001 USD D...
001001-001	1M	USD		EUR	KEY 001001-001 EUR B...
Close Save Changes					

GUI de configuration de SSI sur Box

Si ce n'est pas le cas, la soumission de la commande est bloquée. Cela garantit que chaque transaction est clairement associée à un ensemble d'instructions de

règlement, évitant ainsi toute confusion ou erreur lors du processus de transfert de fonds.



GUI de box

En résumé, le SSI joue un rôle crucial dans le processus de transaction, en servant de pont entre la réalisation d'une transaction et son règlement final. La compréhension approfondie de cette fonctionnalité est donc essentielle pour garantir un développement de test efficace et pertinent.

3. Développement d'un Test d'automation

a. Préparation et Compréhension du Contexte

Avant de commencer le développement du test, il est crucial de comprendre en profondeur la fonctionnalité à tester. Pour cela, je m'immerge dans les guides d'utilisation de chaque module concerné. Cette étape me permet d'acquérir une connaissance approfondie de la fonctionnalité, essentielle pour garantir la pertinence du test.

b. Compréhension du flow de Test

La plupart du temps, je n'ai pas à créer les étapes de test moi-même. Des ingénieurs plus expérimentés définissent ces étapes du test d'automation dans JIRA. Mon rôle est de comprendre l'objectif du test et les besoins métier associés. Avant de me

rendre dans le code, je réalise manuellement un test, exécutant chaque étape pour m'assurer de sa validité.

c. Configuration de la Fonctionnalité

La première étape du développement consiste à configurer le logiciel. SMS offre une multitude de fonctionnalités, mais la plupart sont désactivées par défaut. Je dois donc activer les fonctionnalités pertinentes pour le test.

d. Configuration de la Base de Données

Outre la configuration du logiciel, je prépare également la base de données. J'ajoute des instructions SQL dans le fichier de configuration pour préparer les enregistrements nécessaires au test, comme les SSI dans cet exemple.

```

76 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('ZAR_SSI_Account1_Funding_1', 'ZAR', 'aAcc01-1', 'ZAR_SSI_Account1_Funding_1', 'FUNDING', 'F');
77 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('ZAR_SSI_Account1_Funding_2', 'ZAR', 'aAcc01-1', 'ZAR_SSI_Account1_Funding_2', 'FUNDING', 'F');
78 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('ZAR_SSI_Account1_Delivery_1', 'ZAR', 'aAcc01-1', 'ZAR_SSI_Account1_Delivery_1', 'DELIVERY', 'F');
79 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('ZAR_SSI_Account1_Delivery_2', 'ZAR', 'aAcc01-1', 'ZAR_SSI_Account1_Delivery_2', 'DELIVERY', 'F');
80 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('ZAR_SSI_Account2_Funding_1', 'ZAR', 'aAcc01-2', 'ZAR_SSI_Account2_Funding_1', 'FUNDING', 'F');
81 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('ZAR_SSI_Account2_Funding_2', 'ZAR', 'aAcc01-2', 'ZAR_SSI_Account2_Funding_2', 'FUNDING', 'F');
82
83 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('EUR_Account1_Funding_1', 'EUR', 'aAcc02-1', 'EUR_Account1_Funding_1', 'FUNDING', 'F');
84 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('EUR_Account1_Funding_2', 'EUR', 'aAcc02-1', 'EUR_Account1_Funding_2', 'FUNDING', 'F');
85 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('EUR_Account1_Delivery_1', 'EUR', 'aAcc02-1', 'EUR_Account1_Delivery_1', 'DELIVERY', 'F');
86 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('EUR_Account1_Delivery_2', 'EUR', 'aAcc02-1', 'EUR_Account1_Delivery_2', 'DELIVERY', 'F');
87
88 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('USD_Account1_Funding_1', 'USD', 'aAcc02-1', 'USD_Account1_Funding_1', 'FUNDING', 'F');
89 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('USD_Account1_Funding_2', 'USD', 'aAcc02-1', 'USD_Account1_Funding_2', 'FUNDING', 'F');
90 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('USD_Account1_Delivery_1', 'USD', 'aAcc02-1', 'USD_Account1_Delivery_1', 'DELIVERY', 'F');
91 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('USD_Account1_Delivery_2', 'USD', 'aAcc02-1', 'USD_Account1_Delivery_2', 'DELIVERY', 'F');
92
93 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('AUD_Account1_Funding_1', 'AUD', 'aAcc02-1', 'AUD_Account1_Funding_1', 'FUNDING', 'F');
94 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('AUD_Account1_Funding_2', 'AUD', 'aAcc02-1', 'AUD_Account1_Funding_2', 'FUNDING', 'F');
95 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('AUD_Account1_Delivery_1', 'AUD', 'aAcc02-1', 'AUD_Account1_Delivery_1', 'DELIVERY', 'F');
96 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('AUD_Account1_Delivery_2', 'AUD', 'aAcc02-1', 'AUD_Account1_Delivery_2', 'DELIVERY', 'F');
97
98 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('GBP_Account1_Funding_1', 'GBP', 'aAcc02-1', 'GBP_Account1_Funding_1', 'FUNDING', 'F');
99 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('GBP_Account1_Funding_2', 'GBP', 'aAcc02-1', 'GBP_Account1_Funding_2', 'FUNDING', 'F');
100 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('GBP_Account1_Delivery_1', 'GBP', 'aAcc02-1', 'GBP_Account1_Delivery_1', 'DELIVERY', 'F');
101 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('GBP_Account1_Delivery_2', 'GBP', 'aAcc02-1', 'GBP_Account1_Delivery_2', 'DELIVERY', 'F');
102
103 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('EUR_Account2_Both', 'EUR', 'aAcc02-2', 'EUR_Account2_Both', 'BOTH', 'F');
104 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('EUR_Account2_Funding_1', 'EUR', 'aAcc02-2', 'EUR_Account2_Funding_1', 'FUNDING', 'F');
105 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('EUR_Account2_Funding_2', 'EUR', 'aAcc02-2', 'EUR_Account2_Funding_2', 'FUNDING', 'F');
106 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('EUR_Account2_Delivery_1', 'EUR', 'aAcc02-2', 'EUR_Account2_Delivery_1', 'DELIVERY', 'F');
107
108 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('USD_Account2_Both', 'USD', 'aAcc02-2', 'USD_Account2_Both', 'BOTH', 'F');
109 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('USD_Account2_Funding_1', 'USD', 'aAcc02-2', 'USD_Account2_Funding_1', 'FUNDING', 'F');
110 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('USD_Account2_Funding_2', 'USD', 'aAcc02-2', 'USD_Account2_Funding_2', 'FUNDING', 'F');
111
112 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('AUD_Account2_Funding_1', 'AUD', 'aAcc02-2', 'AUD_Account2_Funding_1', 'FUNDING', 'F');
113
114 INSERT IGNORE INTO STANDARD_SETLEMENT_INSTRUCTION(SSI_KEY, SSI_CURRENCY, SSI_ACCOUNT_ID, SSI_DESCRIPTION, SSI_DIRECTION, SSI_DEFAULT, SSI_DELETED) VALUES ('GBP_Account2_Delivery_1', 'GBP', 'aAcc02-2', 'GBP_Account2_Delivery_1', 'DELIVERY', 'F');
115
116

```

dossier de SQL configuration

e. Développement des Modules UI

L'étape suivante consiste à développer les modules UI. Mon objectif est de créer des abstractions pour chaque élément de l'interface utilisateur et d'encapsuler toutes les méthodes associées. Par exemple, pour le SSI, je crée une classe SSI pour représenter la fenêtre UI. Cette approche modulaire facilite grandement le développement de tests par d'autres ingénieurs qui n'ont pas besoin de connaître les détails de Selenium. Je vais parler de cette partie en détail dans la section suivante.

f. Développement du Test Junit

Une fois les modules UI en place, je commence à écrire le test Junit. En utilisant les méthodes que j'ai développées, je suis les étapes définies dans JIRA pour construire le test. Cette étape peut également révéler des lacunes ou des besoins d'amélioration dans les modules UI, que je rectifie au fur et à mesure.

g. Exécution et Analyse des Résultats

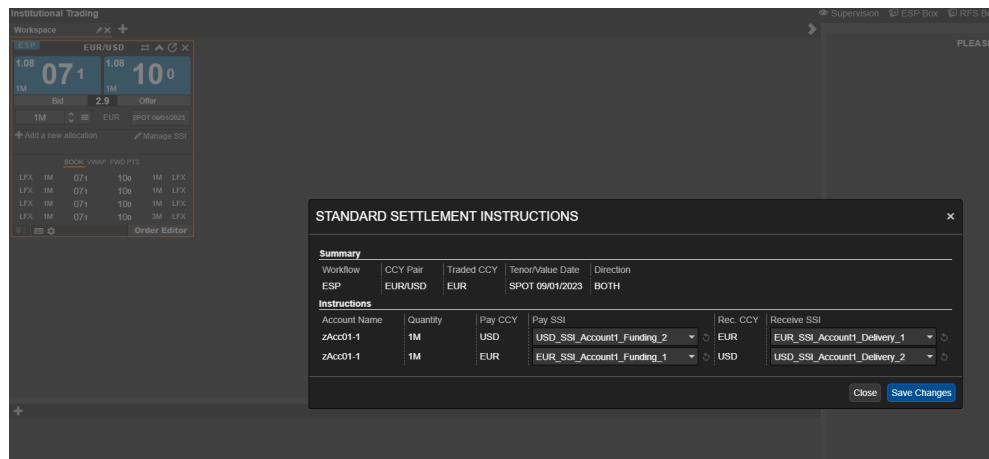
Lorsque le test est prêt, je l'exécute. Si le test échoue, il y a généralement deux raisons possibles : soit il y a un bug dans le flow de test que j'ai développé, soit il y a un bug dans la fonctionnalité elle-même. Dans le premier cas, je corrige le bug dans le flow de test. Dans le second cas, je crée un ticket JIRA pour signaler le bug à l'équipe de développement. C'est aussi le but des tests : le développeur de tests n'a pas besoin de comprendre les détails de développement de la fonction, mais effectue des tests en boîte noire. Si le résultat attendu n'est pas obtenu, il semble que la fonctionnalité a un souci.

4. Conception du Flow de Test pour SSI

Lors de la mise en œuvre des tests automatisés pour la fonctionnalité SSI, il est essentiel de comprendre la nature de l'interface utilisateur de SSI pour concevoir un flow de test efficace. Voici comment j'ai abordé la conception du flow pour SSI :

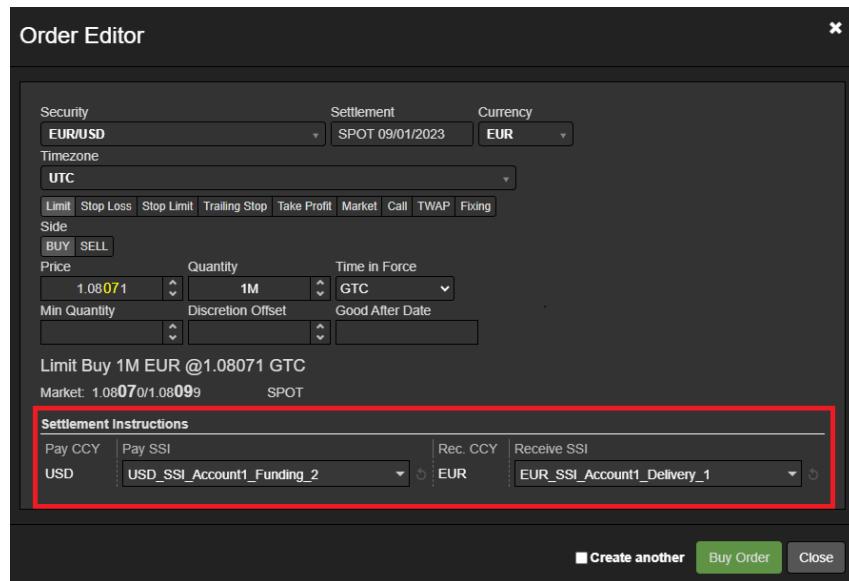
a. Classification de l'Interface Utilisateur de SSI

- Type Pop-up [1]: Comme son nom l'indique, cette interface est une fenêtre contextuelle qui s'affiche lorsque l'utilisateur clique sur un bouton spécifique. Ce bouton n'est visible que si l'option SSI est activée dans le fichier de configuration de démarrage de SMS. Dans cette fenêtre pop-up, l'utilisateur peut définir, confirmer et sélectionner diverses informations liées à SSI, telles que les détails du compte, les informations sur la monnaie et l'ID SSI.



GUI de SSI de type Pop-up

- Type Intégré [2]: Lorsque la fonctionnalité SSI est activée, une table SSI est ajoutée à l'interface utilisateur existante. Cette table contient des informations similaires à celles de la fenêtre pop-up et permet également de définir, confirmer et sélectionner des informations liées à SSI.



GUI de SSI de type intégré

b. Approche de Conception du Flow

La conception du flow de test pour ces deux types d'interfaces nécessite des approches différentes :

Pour le Type Pop-up :

- Un nouveau flow est créé spécifiquement pour gérer les interactions avec la fenêtre pop-up.

- Une instance de ce flow est ensuite ajoutée à JunitRunner, qui sert de classe parente pour tous les tests. JunitRunner contient des instances de tous les objets UI utilisés dans les tests, et ces instances sont créées en tant que singletons pour garantir la cohérence et éviter les duplications.

Pour le Type Intégré:

- Comme l'interface utilisateur cible (par exemple, Order Editor) a déjà un flow associé dans JunitRunner, il n'est pas nécessaire de créer un nouveau flow.
- Au lieu de cela, le flow existant est étendu pour inclure les opérations liées à SSI. Cela garantit que les tests peuvent interagir avec les éléments SSI intégrés de manière transparente.

La conception du flow de test pour SSI nécessite une compréhension approfondie de l'interface utilisateur de SSI et une approche modulaire pour garantir que les tests sont à la fois efficaces et maintenables. En outre, il est essentiel de souligner que ces tests jouent un rôle essentiel pour nos clients, car ils garantissent la qualité et la fiabilité de notre produit, renforçant ainsi notre réputation sur le marché. Néanmoins, cette tâche n'a pas été sans défis pour notre équipe. Plus loin dans ce rapport, nous explorerons plus en détail les implications de ces tests pour l'entreprise et les difficultés que nous avons rencontrées lors de leur mise en œuvre.

5. Le plan du stage

Mon stage, d'une durée de 22 semaines avant la soutenance, a été structuré comme suit :

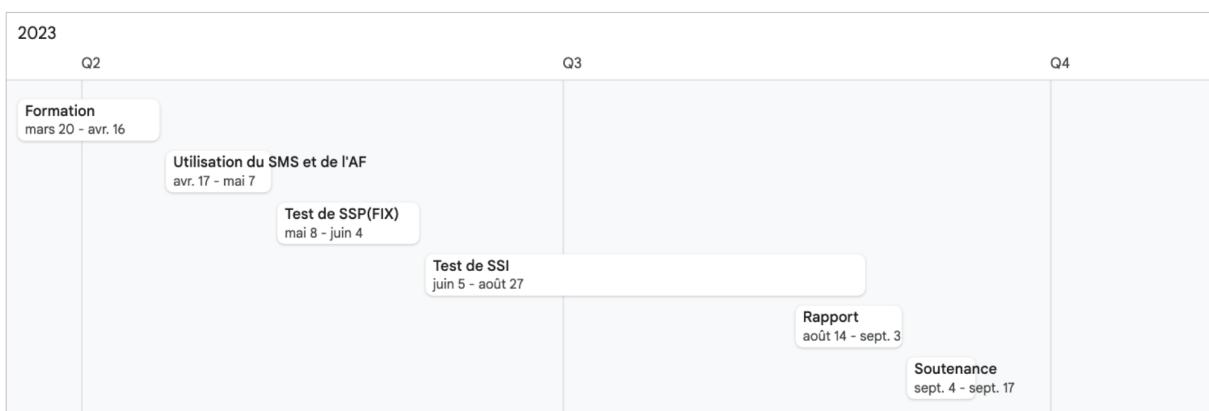


Diagramme de Gantt de stage

Dans le but d'améliorer la qualité de son logiciel, smartTrade envisage d'automatiser certains tests pour les intégrer au processus d'intégration continue et de livraison

(CI/CD). Parmi les tests envisagés, ce sont les tests utilisateurs qui sont principalement ciblés. Ces tests consistent à démarrer le logiciel et à simuler des comportements typiques, appelés scénarios. Il est important de noter que ces tests sont effectués en mode "boîte noire", c'est-à-dire sans accès direct au code source ou aux méthodes internes du logiciel. Au lieu de cela, ils se concentrent sur l'évaluation des comportements observables depuis l'interface utilisateur du logiciel.

IV. Analyse du travail effectué

1. Rappel du framework d'automation

Au cours de mon stage, j'ai été entièrement responsable du développement des tests UI, ce qui rend ce travail particulièrement représentatif de mes contributions. Tous mes développements de tests ont été basés sur le cadre de test existant nommé "Automation". Ce projet a déjà défini presque tous les outils nécessaires pour le développement de tests courants. Le projet "Automation" est structuré en plusieurs packages :

- fix : contient des classes utilitaires liées aux messages FIX.
- rest : contient des classes utilitaires pour développer des tests pour les interfaces restful.
- standalone : encapsule les outils de configuration pour le démarrage de SMS et les outils de configuration pour télécharger les dépendances avant le déploiement (chaque démarrage de SMS peut obtenir les dernières dépendances à partir du téléchargement).
- web : contient des classes liées au développement de tests UI, telles que la configuration des extensions Chrome, la génération de rapports de tests, le pilote junit et les actions. J'ai principalement utilisé les classes sous le package "action", en particulier UIActions et KeyboardActions, qui sont basées sur Selenium et permettent d'effectuer des opérations sur les éléments UI dans le navigateur et des entrées au clavier.
- util : contient des méthodes utilitaires courantes, telles que la formatisation des chaînes et la prise de captures d'écran lors de l'échec d'un test.

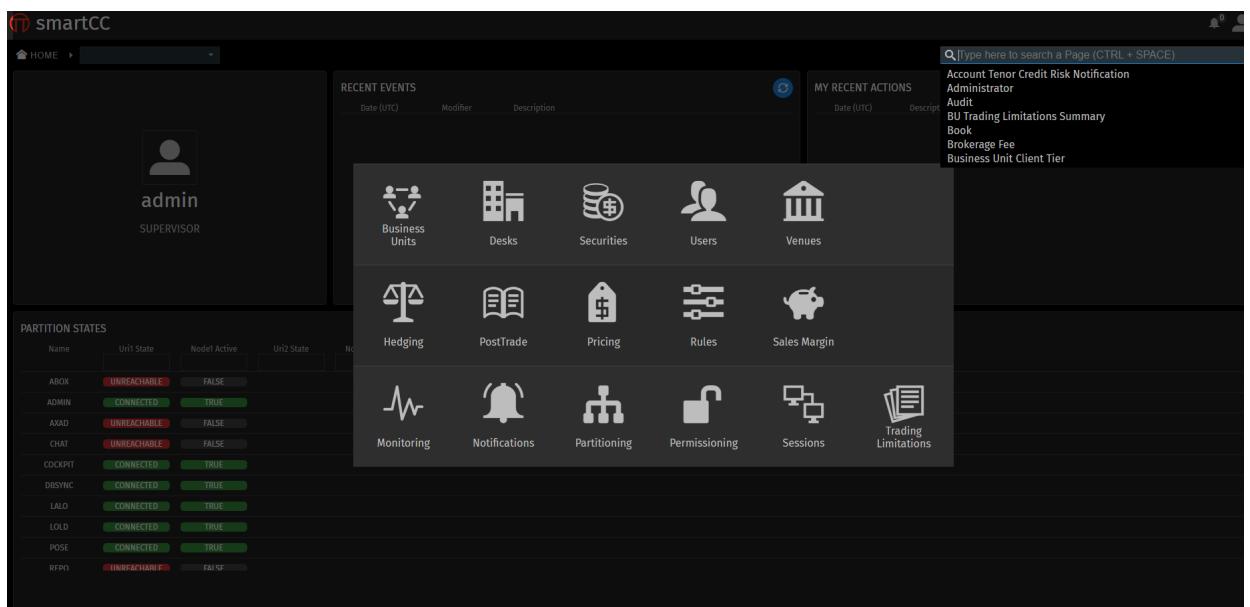
Tous les développements de flows que j'ai effectués par la suite devaient utiliser les classes de ce projet. Ce cadre a grandement simplifié le développement des tests, me permettant de rédiger des flows appropriés sans avoir à interagir directement avec Selenium.

2. Analyse du flow pour SSI

a. Flow de SSI dans SmartCC2

Mon premier objectif était de développer des tests pour valider la création, la modification et la suppression de SSI sur SmartCC2. SmartCC2 est le projet de configuration et de gestion de Cockpit, généralement réservé aux administrateurs. Dans SmartCC2, nous pouvons configurer et vérifier diverses fonctionnalités de Cockpit, y compris SSI. Selon la définition de SSI, les SSI dans Cockpit ne peuvent être définis que dans SmartCC2 ou par un fournisseur tiers. J'avais donc pour tâche de mettre en œuvre des tests automatisés pour ajouter, modifier et supprimer des SSI sur SmartCC2.

C'était un défi car SSI était une fonctionnalité entièrement nouvelle. Avant cela, il n'y avait aucune implémentation de SSI dans le projet de test. Par conséquent, j'ai dû d'abord implémenter l'interface utilisateur de SSI dans SmartCC2.



GUI de SmartCC2

Heureusement, le style de code du cadre de test automatisé est très bien structuré. Les développeurs du cadre ont abstrait les interfaces communes de SmartCC2 et ont encapsulé de nombreuses méthodes communes pour manipuler des lignes et des colonnes spécifiques dans des tableaux. Par exemple, la classe SmartCC2Flow abstrait toute l'architecture UI de [SmartCC2](#). Elle définit deux instances à l'intérieur, [SmartCC2Grid](#) et [SmartCC2Form](#). [SmartCC2Grid](#) représente le tableau affichant plusieurs lignes de données, tandis que [SmartCC2Form](#) représente le formulaire utilisé pour éditer une ligne de données spécifique. Ces deux classes fonctionnent avec des numéros de lignes et de colonnes. J'ai donc simplement créé une nouvelle

classe, SmartCC2SSIFlow, héritant de SmartCC2, et après avoir correctement encapsulé les noms de colonnes en tant qu'énumérations, j'ai à nouveau refactorisé les méthodes à l'intérieur de SmartCC2Flow.

ID	Description	Account	Currency	Direction	Default
AUD_SSI_Account1_Delivery_1	AUD_SSI_Account1_Delivery_1	zAcc01-1	AUD	DELIVERY	FALSE
AUD_SSI_Account1_Delivery_2	AUD_SSI_Account1_Delivery_2	zAcc01-1	AUD	DELIVERY	FALSE
AUD_SSI_Account1_Funding_1	AUD_SSI_Account1_Funding_1	zAcc01-1	AUD	FUNDING	FALSE
AUD_SSI_Account1_Funding_2	AUD_SSI_Account1_Funding_2	zAcc01-1	AUD	FUNDING	FALSE
CAD_SSI_Account1_Delivery_1	CAD_SSI_Account1_Delivery_1	zAcc01-1	CAD	DELIVERY	FALSE
CAD_SSI_Account1_Delivery_2	CAD_SSI_Account1_Delivery_2	zAcc01-1	CAD	DELIVERY	FALSE
CAD_SSI_Account1_Funding_1	CAD_SSI_Account1_Funding_1	zAcc01-1	CAD	FUNDING	FALSE
CAD_SSI_Account1_Funding_2	CAD_SSI_Account1_Funding_2	zAcc01-1	CAD	FUNDING	FALSE
EUR_SSI_Account1_Delivery_1	EUR_SSI_Account1_Delivery_1	zAcc01-1	EUR	DELIVERY	TRUE
EUR_SSI_Account1_Delivery_2	EUR_SSI_Account1_Delivery_2	zAcc01-1	EUR	DELIVERY	FALSE
EUR_SSI_Account1_Funding_1	EUR_SSI_Account1_Funding_1	zAcc01-1	EUR	FUNDING	TRUE
EUR_SSI_Account1_Funding_2	EUR_SSI_Account1_Funding_2	zAcc01-1	EUR	FUNDING	FALSE
JPY_SSI_Account1_Both_1	JPY_SSI_Account1_Both_1	zAcc01-1	JPY	FUNDING, DELIVERY	TRUE
JPY_SSI_Account1_Both_2	JPY_SSI_Account1_Both_2	zAcc01-1	JPY	FUNDING, DELIVERY	FALSE
JPY_SSI_Account1_Delivery_1	JPY_SSI_Account1_Delivery_1	zAcc01-1	JPY	DELIVERY	FALSE
JPY_SSI_Account1_Delivery_2	JPY_SSI_Account1_Delivery_2	zAcc01-1	JPY	DELIVERY	FALSE
JPY_SSI_Account1_Funding_1	JPY_SSI_Account1_Funding_1	zAcc01-1	JPY	FUNDING	FALSE
JPY_SSI_Account1_Funding_2	JPY_SSI_Account1_Funding_2	zAcc01-1	JPY	FUNDING	FALSE
USD_SSI_Account1_Delivery_1	USD_SSI_Account1_Delivery_1	zAcc01-1	USD	DELIVERY	FALSE
USD_SSI_Account1_Delivery_2	USD_SSI_Account1_Delivery_2	zAcc01-1	USD	DELIVERY	TRUE

GUI abstraite SmartCC2Grid

ID	Description	Account	Currency	Direction	Default
AUD_SSI_Account1_Delivery_1	AUD_SSI_Account1_Delivery_1	zAcc01-1	AUD	DELIVERY	FALSE
AUD_SSI_Account1_Delivery_2	AUD_SSI_Account1_Delivery_2	zAcc01-1	AUD	DELIVERY	FALSE
AUD_SSI_Account1_Funding_1	AUD_SSI_Account1_Funding_1	zAcc01-1	AUD	FUNDING	FALSE
AUD_SSI_Account1_Funding_2	AUD_SSI_Account1_Funding_2	zAcc01-1	AUD	FUNDING	FALSE
CAD_SSI_Account1_Delivery_1	CAD_SSI_Account1_Delivery_1	zAcc01-1	CAD	DELIVERY	FALSE
CAD_SSI_Account1_Delivery_2	CAD_SSI_Account1_Delivery_2	zAcc01-1	CAD	DELIVERY	FALSE
CAD_SSI_Account1_Funding_1	CAD_SSI_Account1_Funding_1	zAcc01-1	CAD	FUNDING	FALSE
CAD_SSI_Account1_Funding_2	CAD_SSI_Account1_Funding_2	zAcc01-1	CAD	FUNDING	FALSE
EUR_SSI_Account1_Delivery_1	EUR_SSI_Account1_Delivery_1	zAcc01-1	EUR	DELIVERY	TRUE
EUR_SSI_Account1_Delivery_2	EUR_SSI_Account1_Delivery_2	zAcc01-1	EUR	DELIVERY	FALSE
EUR_SSI_Account1_Funding_1	EUR_SSI_Account1_Funding_1	zAcc01-1	EUR	FUNDING	TRUE
EUR_SSI_Account1_Funding_2	EUR_SSI_Account1_Funding_2	zAcc01-1	EUR	FUNDING	FALSE
JPY_SSI_Account1_Both_1	JPY_SSI_Account1_Both_1	zAcc01-1	JPY	FUNDING, DELIVERY	TRUE
JPY_SSI_Account1_Both_2	JPY_SSI_Account1_Both_2	zAcc01-1	JPY	FUNDING, DELIVERY	FALSE
JPY_SSI_Account1_Delivery_1	JPY_SSI_Account1_Delivery_1	zAcc01-1	JPY	DELIVERY	FALSE
JPY_SSI_Account1_Delivery_2	JPY_SSI_Account1_Delivery_2	zAcc01-1	JPY	DELIVERY	FALSE
JPY_SSI_Account1_Funding_1	JPY_SSI_Account1_Funding_1	zAcc01-1	JPY	FUNDING	FALSE
JPY_SSI_Account1_Funding_2	JPY_SSI_Account1_Funding_2	zAcc01-1	JPY	FUNDING	FALSE
USD_SSI_Account1_Delivery_1	USD_SSI_Account1_Delivery_1	zAcc01-1	USD	DELIVERY	FALSE
USD_SSI_Account1_Delivery_2	USD_SSI_Account1_Delivery_2	zAcc01-1	USD	DELIVERY	TRUE

GUI abstraite SmartCC2Form (la partie gauche)

Pour illustrer, lors de la création d'un formulaire SSI, si je souhaite définir l'ID d'un nouveau SSI comme "[ssiID](#)", je n'ai qu'à appeler la méthode `fillInput` de SmartCC2Form, fournir le numéro de colonne approprié et la donnée. Cette action

remplira la ligne ID du formulaire avec "ssiID". Cette conception offre une grande extensibilité et commodité pour le développement de nouveaux flows comme on peut le voir dans l'image suivante:

```
/*
 * Fill the id input in creating SSI form; Before call this method, createNewSSI() should be called, otherwise throw an
 *
 * @param ssiID
 */
public void editID(String ssiID) {
    try {
        form.fillInput(Scc2FormFieldsLabels.SSI_ID, ssiID);
    } catch (final Exception e) {
        final String msg = TestAutoUtils.formatString("Cannot editing the id input", Scc2FormFieldsLabels.SSI_ID.name());
        TestAutoUtils.fail(msg, e, LOG);
    }
}
```

Méthode utilisée pour modifier l'ID lors de la création de SSI

De la même manière, avec SmartCC2Grid, si je veux supprimer une ligne spécifique, je n'ai qu'à fournir l'ID de la ligne à supprimer. L'action déclenchera le bouton "supprimer" pour cette ligne. Si aucune ligne avec l'ID spécifié n'est trouvée, une erreur est générée et le test échoue. Cette action est réalisée en utilisant des critères de sélection XPath complexes pour identifier la ligne contenant l'ID spécifié, puis en utilisant XPath pour cliquer sur le bouton de suppression de cette ligne. La méthode de clic, est encapsulée dans UIAction du projet Automation. Elle prend une chaîne XPath comme argument pour effectuer l'action de clic sur l'élément cible.

```
public void deleteSSIWithID(final String ID) {
    grid.deleteEntry(ID);
}
```

Méthode utilisée pour supprimer le SSI sur SmartCC2

Cependant, de nombreuses méthodes n'avaient pas été implémentées dans SmartCC2Flow car cette classe avait été conçue pour être universelle et n'incluait pas de méthodes pour des usages spécifiques. Pour ces méthodes spécifiques, j'ai dû les implémenter directement dans SmartCC2SSIFlow.

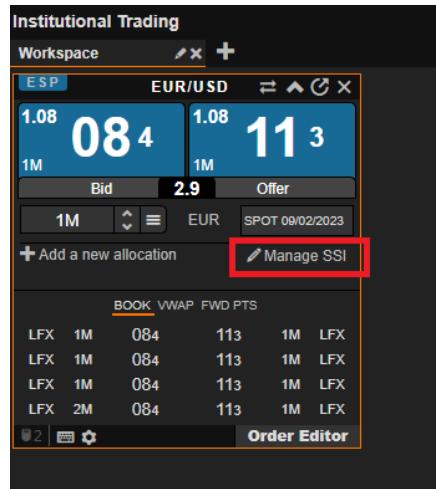
Un autre défi était que toutes les interfaces utilisateur ne suivaient pas un standard de développement uniforme, étant donné que c'est un logiciel avec une histoire de 20 ans. Cela signifiait que certains éléments UI, bien qu'ils semblent identiques en apparence, ne pouvaient pas être sélectionnés avec les XPath précédents. En tant qu'ingénieur de développement de tests, je ne pouvais pas directement modifier le code du logiciel. Ma solution a été de créer des XPath personnalisés et d'implémenter de nouvelles méthodes pour ces éléments. Cette expérience m'a fait comprendre l'importance de la standardisation du code. Un code de mauvaise qualité peut rendre la refonte très difficile, rendant le projet lourd et complexe. Dans certains cas, 99 scénarios pourraient utiliser une méthode définie à l'avance, mais un seul

scénario, développé sans suivre les normes, ne pourrait pas. Le résultat est que nous devons créer de nombreuses méthodes, voire de nombreuses classes, pour ce scénario particulier. C'est pour ces situations de plus en plus fréquentes qu'une bibliothèque de composants est en développement chez smartTrade.

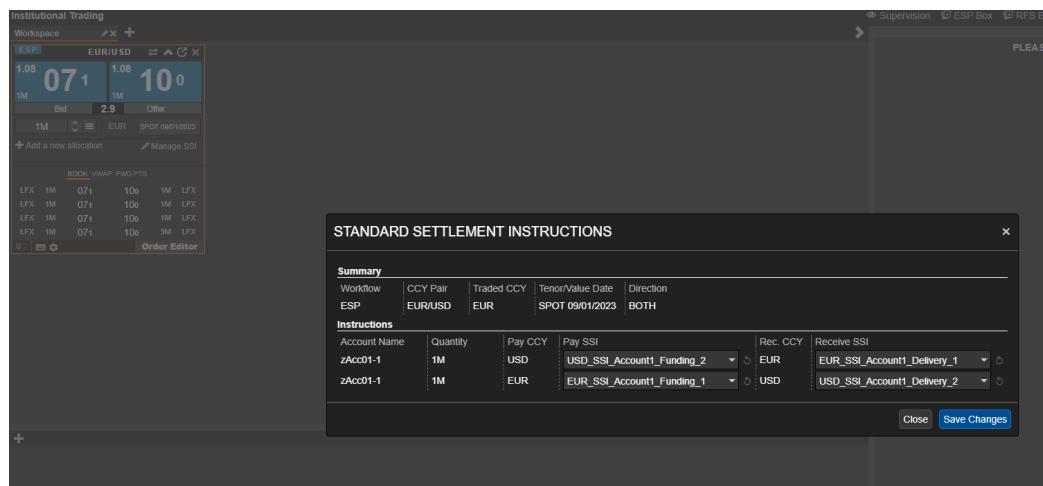
b. Flow de SSI dans Cockpit

Le développement de SSI sur Cockpit se distingue nettement de celui sur SmartCC2. En effet, il n'existe pas de méthodes communes, comme SmartCC2Flow, qui permettent de décrire de manière abstraite la majorité des interfaces utilisateur de Cockpit. La raison en est la richesse et la complexité des fonctionnalités de Cockpit, dont la plupart des interfaces utilisateur sont distinctes et certaines ont même été conçues sur mesure pour des clients spécifiques. Par conséquent, j'ai dû développer moi-même le flow pour SSI dans Cockpit. Comme mentionné dans le chapitre précédent, l'interface utilisateur de SSI dans Cockpit peut être grossièrement divisée en deux types : Pop-up et intégrée. J'ai donc basé mon développement de flow dans Cockpit sur cette distinction.

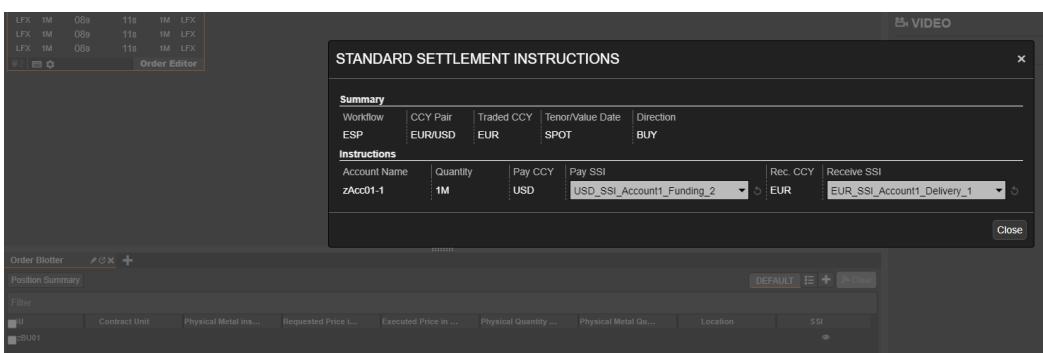
- Type Pop-up: dans les tests actuels de SSI, il existe deux modules associés au type modale. Le premier est accessible via le bouton "[Manager SSI](#)" dans le module "Box". Lorsqu'on clique dessus, [une fenêtre pop-up](#) apparaît pour gérer les SSI. Le second est visible dans le widget "Order Blotter", qui affiche les informations des commandes historiques. Une colonne spécifique, nommée "SSI", contient une icône en forme d'œil. En cliquant dessus, les informations SSI associées à cette commande s'affichent dans une fenêtre pop-up. Alors que le premier est utilisé pour modifier et visualiser les SSI, le second ne sert qu'à la visualisation. J'ai donc pu créer une classe parente commune pour ces deux types, intégrant les méthodes communes telles que "cliquer sur le bouton de fermeture" ou "utiliser la méthode assert pour vérifier le contenu du tableau". Cependant, des méthodes comme "cliquer sur le bouton save" ou "sélectionner un SSI spécifique" n'ont pas été définies car elles ne sont pas applicables à [la fenêtre pop-up](#) de "Order Blotter".



Après avoir cliqué sur « Manager SSI » sur la Box, un SSI Pop-up apparaîtra



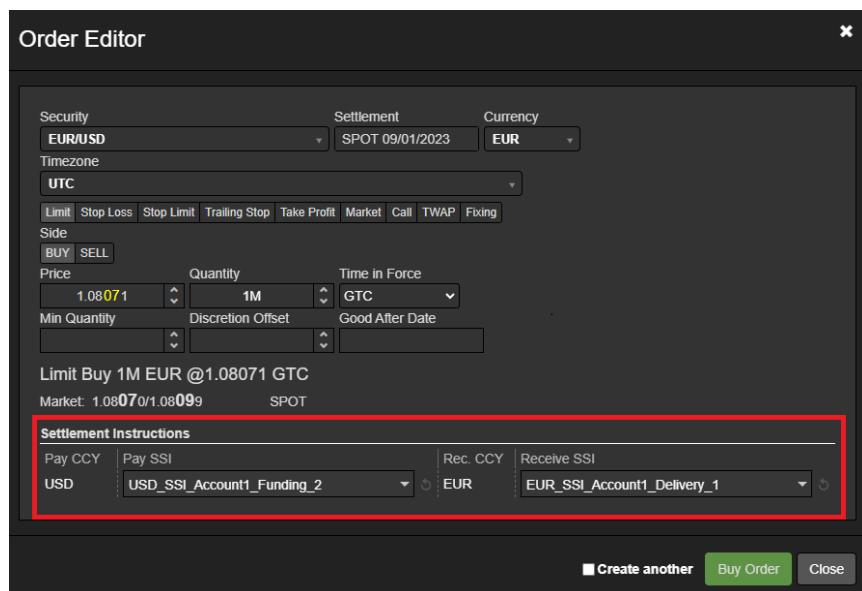
Pop-up de SSI en Box



Pop-up de SSI en Order Blotter

- Type Intégré: Pour le type intégré de SSI, une nouvelle section est ajoutée à l'interface utilisateur existante pour gérer les SSI. Par exemple, comme le montre l'image, le module "Order Editor", utilisé pour éditer et envoyer des commandes, affiche une nouvelle section "[Settlement Instructions](#)" lorsque la fonctionnalité SSI est activée. Cette section permet de visualiser et de

sélectionner les SSI. Le développement de ce type de flow est le plus complexe car il ne nécessite pas la création d'un nouveau flow, mais l'ajout de méthodes liées à SSI dans le flow existant de "Order Editor". Cela m'a obligé à comprendre en profondeur les fonctionnalités du module "Order Editor" et à maîtriser les méthodes de son flow associé. Par exemple, j'ai créé deux [méthodes](#), `selectFundingSSI` et `selectDeliverySSI`, pour sélectionner respectivement les SSI de financement et de livraison. Ces méthodes reposent sur une méthode parent [selectSSI](#) qui, en fonction du type de SSI, choisit le XPath approprié. En réalité, la rédaction du code est souvent la partie la plus simple du processus. Le véritable défi et ce qui prend le plus de temps est d'examiner attentivement le code écrit par d'autres. Avant de créer une nouvelle méthode ou classe, je dois m'assurer qu'elle n'a pas déjà été implémentée, afin d'éviter la redondance et de maintenir le projet épuré.



GUI de SSI en Order Editor

```
public void selectFundingSSI(final String selectedSSI) {
    selectSSI(SSI.FUNDING, selectedSSI);
}

public void selectDeliverySSI(final String selectedSSI) {
    selectSSI(SSI.DELIVERY, selectedSSI);
}
```

Méthodes définies dans OrderEditorFlow pour sélectionner le SSI

```

private void selectSSI(SSI ssiType, final String selectedSSI) {
    final String ssiSelectionDivPath = ssiType == SSI.FUNDING ? SmsKeys.RETAIL_FUNDING_SSI_DIV.getLocator() : SmsKeys.RETAIL_DELIVERY_SSI_DIV.getLocator();
    final String SSILlistBox = SmsKeys.SSI_BOX_LIST.getLocator();
    try {
        UIActions.get().click(ssiSelectionDivPath);
        UIActions.get().waitForElementVisible(SSILlistBox);
        final String targetSSI = SmsKeys.SSI_BOX_SINGLE_IN_LIST_BY_NAME.getLocator(selectedSSI);
        UIActions.get().assertElementVisible(targetSSI);
        UIActions.get().click(targetSSI);
    } catch (final Exception e) {
        final String msg = ErrorMessage.getInstance().getErrorMessage("Funding SSI list box");
        TestAutoUtils.fail(msg, e, LOG);
    }
}

```

Méthode parent de *selectFundingSSI* et *selectDeliverySSI*

3. Développement de Test de JUnit

Une fois les flows nécessaires développés, j'ai entamé la création des Test Suites avec JUnit. Ces suites de tests sont essentielles pour garantir que les fonctionnalités existantes demeurent opérationnelles après le lancement du logiciel ou après l'intégration de nouveaux codes. Dans un environnement CI/CD, les développeurs peuvent soumettre leurs codes par portions, sans attendre la finalisation complète d'un projet. Cependant, un code partiel peut potentiellement perturber le code existant, causant des dysfonctionnements ou même des erreurs de compilation. Si GitLab peut vérifier la compilabilité du code, il ne peut pas déterminer si les fonctionnalités du projet fonctionnent correctement. C'est là que les tests automatisés entrent en jeu.

a. Importance des Test Suites :

Les Test Suites jouent un rôle crucial dans la stabilité du développement d'un projet. Si un test échoue, le code soumis doit être retiré. Une Test Suite JUnit est composée de nombreux tests unitaires et l'échec de l'un n'affecte pas le fonctionnement des autres.

b. Processus de création

La mise en place d'une Test Suite JUnit consiste principalement à transcrire en code les étapes de test rédigées par des ingénieurs qualités (QA), dans des tickets JIRA. Lors de la conception des tests, le rédacteur doit prendre en compte la couverture des tests et envisager divers scénarios extrêmes pour assurer le bon fonctionnement de la majorité des fonctionnalités du logiciel. De plus, chaque étape du test doit préciser le résultat attendu pour que le développeur de tests puisse traiter correctement les résultats. Si un test, bien développé et exécuté, ne produit pas le résultat attendu, cela peut indiquer un bug dans le logiciel. Cependant, de nombreux échecs de tests que j'ai rencontrés étaient dus à des imperfections dans le développement des tests eux-mêmes. Un changement mineur, comme une pression

supplémentaire sur la touche Tab, peut entraîner un échec du test. Cela nécessite une vérification minutieuse de chaque étape du test pour identifier et corriger le segment de code fautif.

c. Exemple concret

Prenons un exemple illustratif. La [figure](#) montre une étiquette de conception d'étape de test sur JIRA.

Test Details			
		Test Step	
1	In smartcc2, create following SSIs:	-Key1 Desc1 EUR Account1 Funding false -Key2 Desc2 EUR Account1 Funding true -Key21 Desc21 EUR Account2 Delivery true -Key3 Desc3 EUR Account1 Delivery true -Key4 Desc4 USD Account2 Funding false -Key5 Desc5 USD Account2 Funding&Delivery true -Key51 Desc51 USD Account2 Delivery false	SSIs are created
2	Create another SSI: -Key6 Desc6 EUR Account1 Funding true	Funding side default replaced by Funding	SSI is created SSI Key2 has default property updated to false Other SSIs are unchanged

Etape de Test sur JIRA

À l'étape 1, je dois créer six nouveaux SSI dans SmartCC2, et à l'étape 2, un autre SSI. Les "Résultats de Test" indiquent les résultats attendus : les opérations de l'étape 1 devraient toutes réussir, avec six nouveaux SSI listés, tandis que la création à l'étape 2 ne devrait pas affecter les enregistrements SSI existants. Le code correspondant dans la Test Suite JUnit est illustré dans la [figure](#).

```

@test
@Order(10)
@DisplayName("STQA-12605 Configuring new SSI in SmartCC2")
public void addNewSSIsTest() {
    BrowserManager.select(AutomationUsers.ADMIN.toString());
    flowDashboard.loginSmartCC2(AutomationUsers.ADMIN);
    stableflowSmartCC2.selectPage(Sco2Pages.SSI);
    smartCC2SSIFlow.filterByAccount(ZACC00_1);

    // 1
    smartCC2SSIFlow.createNewSSI("Key1", "Desc1", ZACC00_1, EUR, false, FUNDING);
    smartCC2SSIFlow.assertGridSSIInfo(1, "Key1", "Desc1", ZACC00_1, EUR, false, STR_FUNDING);
    smartCC2SSIFlow.createNewSSI("Key2", "Desc2", ZACC00_1, EUR, true, FUNDING);
    smartCC2SSIFlow.assertGridSSIInfo(2, "Key2", "Desc2", ZACC00_1, EUR, true, STR_FUNDING);
    smartCC2SSIFlow.createNewSSI("Key3", "Desc3", ZACC00_1, EUR, true, DELIVERY);
    smartCC2SSIFlow.assertGridSSIInfo(3, "Key3", "Desc3", ZACC00_1, EUR, true, STR_DELIVERY);
    smartCC2SSIFlow.filterByAccount(ZACC00_2);
    smartCC2SSIFlow.createNewSSI("Key21", "Desc21", ZACC00_2, EUR, true, DELIVERY);
    smartCC2SSIFlow.assertGridSSIInfo(1, "Key21", "Desc21", ZACC00_2, EUR, true, STR_DELIVERY);
    smartCC2SSIFlow.createNewSSI("Key4", "Desc4", ZACC00_2, USD, false, FUNDING);
    smartCC2SSIFlow.assertGridSSIInfo(2, "Key4", "Desc4", ZACC00_2, USD, false, STR_FUNDING);
    smartCC2SSIFlow.createNewSSI("Key5", "Desc5", ZACC00_2, USD, true, BOTH);
    smartCC2SSIFlow.assertGridSSIInfo(3, "Key5", "Desc5", ZACC00_2, USD, true, STR_BOTH);
    smartCC2SSIFlow.createNewSSI("Key51", "Desc51", ZACC00_2, USD, false, DELIVERY);
    smartCC2SSIFlow.assertGridSSIInfo(4, "Key51", "Desc51", ZACC00_2, USD, false, STR_DELIVERY);

    // 2
    smartCC2SSIFlow.assertGridIsDefault(1);
    smartCC2SSIFlow.assertGridIsNotDefault(2);
    smartCC2SSIFlow.assertGridIsDefault(3);
    smartCC2SSIFlow.assertGridIsNotDefault(4);
    smartCC2SSIFlow.filterByAccount(ZACC00_1);
    smartCC2SSIFlow.createNewSSI("Key6", "Desc6", ZACC00_1, EUR, true, FUNDING);
    smartCC2SSIFlow.assertGridIsNotDefault(1);
    smartCC2SSIFlow.assertGridIsNotDefault(2);
    smartCC2SSIFlow.assertGridIsDefault(3);
    smartCC2SSIFlow.assertGridIsDefault(4);
}

```

Implémentation de code des étapes de test ci-dessus

J'utilise les flows précédemment créés et j'appelle les méthodes appropriées. Il est important de noter que les étapes 1 et 2 nécessitent d'abord de se connecter à SmartCC2 et d'accéder au module approprié. Après chaque exécution, il est essentiel de vérifier que les résultats correspondent aux attentes. Si tous les tests unitaires d'une Test Suite passent, alors la Test Suite est considérée comme réussie.

d. Exécution de Test Suite sur Bamboo

Après avoir achevé le développement d'une suite de tests et validé son bon fonctionnement en environnement local, la démarche ne s'arrête pas là. Chez smartTrade, nous utilisons [Bamboo](#) pour notre intégration et déploiement continu. Ainsi, pour garantir la stabilité de la nouvelle suite de tests, il est impératif de l'exécuter sur Bamboo.

Build dashboard

Building

7 of 12 online agents building R Si

Test Pipeline > SMS QUICK TAF Develop and support JDK17 > master_sebi > #29 > Test QUICK TAF building on smartSolution > QT J17 > LFX_7_8_GA_BUGFIX > #8 > checkout_Build_lic building on	Building for 11 mins. 3 mins remaining
Test Pipeline > SMS QUICK TAF Develop and support JDK17 > LFX_7_8_GA_BUGFIX > #7 > Test QUICK TAF building on	Building for 8 mins. 6 mins remaining
smartSolution > Automation Sonar > #787 > checkout_Build_lic building on	Building for 8 mins. 8 mins remaining
smartSolution > QT J17 > release > #62 > checkout_Build_lic building on	Building for 4 mins. 190 mins remaining
Product Strategy > ST Process Checker > #7749 > Default Job building on	Building for 4 mins. 11 mins remaining
Product Strategy > Documentation Generation Confluence > #1550 > Generate, export to Confluence building on	Building for 52 secs. 8 mins remaining
	Building for 52 secs. 85 mins remaining

Queue

Test Pipeline > SMS TAF Develop and Support JDK17 > release > #39 >	Waiting to be built...
Test Pipeline > SMS QUICK TAF Develop and support JDK17 > release > #31 >	Waiting to be built...

Tâches de test en cours d'exécution sur Bamboo

Dans notre configuration Bamboo chez smartTrade, l'outil répartit automatiquement les tâches de test entre tous les serveurs disponibles. Dès qu'un serveur est libre, il reçoit une tâche, lance SMS (SmartSolution) dans un environnement Linux selon les configurations et fichiers de base de données que j'ai définis, puis exécute la tâche de test. De plus, chaque fois qu'un collaborateur soumet du code à la branche principale, cela déclenche automatiquement l'exécution des suites de tests existantes. Seul un passage réussi de ces tests garantit le bon fonctionnement de SMS.

En cas d'échec d'un test, une notification par e-mail est envoyée à toutes les parties concernées. Comme illustré, l'interface de Bamboo offre une [vue détaillée](#) des tests : nombre de tests réussis, échoués, corrigés, les journaux d'exécution, et même les commits associés aux tests échoués. Ces informations permettent aux développeurs de rapidement identifier et résoudre les problèmes. L'utilisation de Bamboo optimise considérablement notre efficacité de développement et la stabilité du code, tout en évitant l'intégration de code erroné dans la branche principale.

The screenshot shows a GitLab build dashboard for a project named 'smartSolution' under the 'QT' plan. The build number is #356, which failed. The summary section displays the following test results:

Category	Value
New failures	0
Existing failures	1
Fixed	1

Détails des tâches de test sur Bamboo

Une fois que mes tests passent avec succès sur Bamboo, je peux initier une "Merge Request" sur GitLab. Mon code sera alors examiné par le responsable de la branche. Une fois approuvé, il sera fusionné avec la branche principale. Dès lors, les tests que j'ai développés seront intégrés au processus de vérification de chaque soumission de code par d'autres membres de l'équipe.

4. Développement de Test de FIX

a. Rappel de FIX et l'intégration de FIX chez smartTrade

Le protocole FIX (Financial Information eXchange) est devenu la norme de facto pour l'échange électronique d'informations financières. Conçu à l'origine pour le trading d'actions, il s'est rapidement étendu à d'autres domaines financiers, offrant une communication rapide, fiable et sécurisée entre les différentes parties prenantes du marché financier.

Au sein de smartTrade, la communication entre les différents composants internes est assurée par le protocole Stamp, une innovation propre à l'entreprise. Ce protocole, développé en interne, a été conçu pour offrir une efficacité maximale et répondre aux exigences spécifiques de smartTrade en matière de communication interne entre ses composants. Sa conception sur mesure le rend particulièrement adapté aux besoins uniques de l'entreprise, garantissant rapidité et fiabilité.

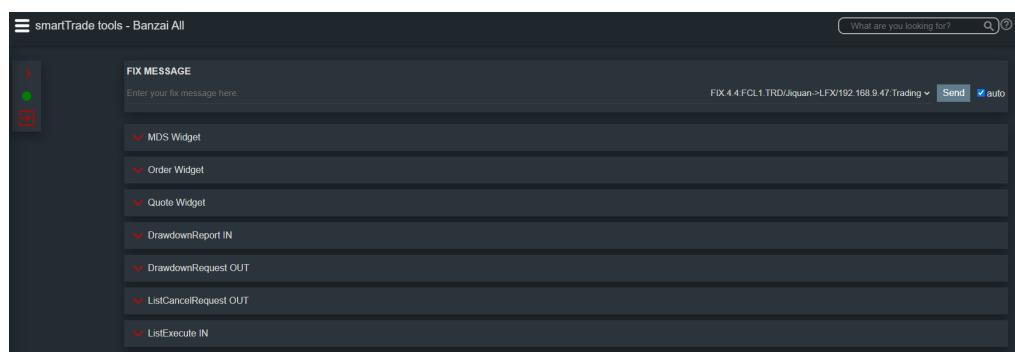
Cependant, pour les interactions avec les entités externes, smartTrade a choisi d'adopter le protocole FIX. Cette décision permet d'assurer une compatibilité et une interopérabilité optimales avec les partenaires et clients de l'entreprise. Ainsi, ces derniers peuvent envoyer des ordres à SMS via FIX, et une fois ces ordres reçus et validés par SMS, ils sont traités de manière efficace. Cette dualité de protocoles garantit que smartTrade peut bénéficier à la fois de la spécificité de Stamp pour ses opérations internes et de l'universalité de FIX pour ses communications externes.

b. Processus de développement des tests FIX

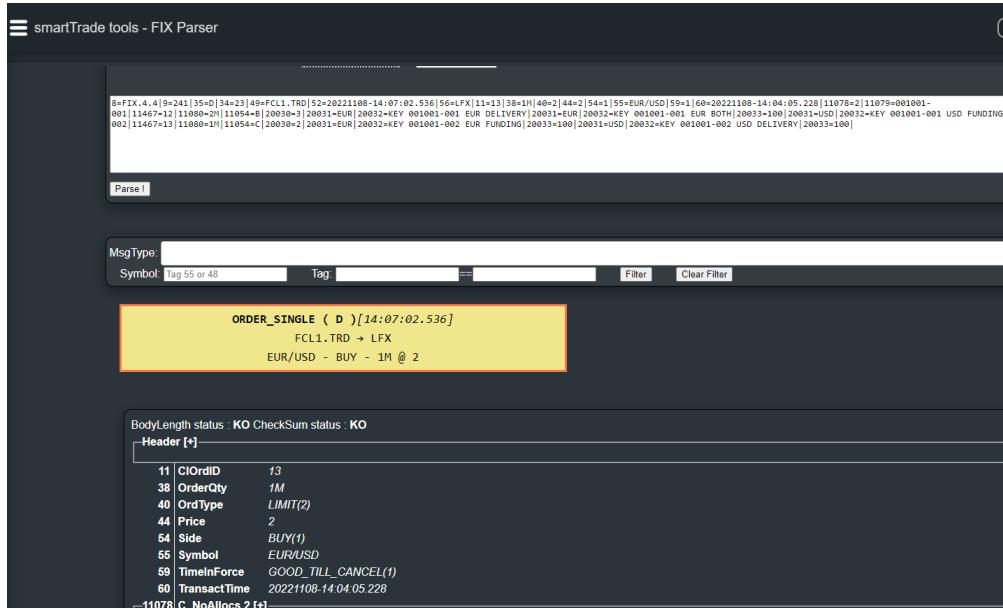
Le développement des tests pour le protocole FIX est une tâche complexe et exigeante. Contrairement aux tests d'interface utilisateur, les tests FIX sont axés sur l'API, nécessitant une compréhension approfondie du protocole.

Pour faciliter cette tâche, j'ai utilisé deux outils clés de smartTrade :

- FIX Parser : C'est un outil essentiel qui décode et interprète les messages FIX. Il offre une visualisation claire des données échangées, permettant une analyse rapide et efficace. Grâce à cet outil, j'ai pu comprendre les nuances des messages, identifier les erreurs et les corriger.
- Banzai : C'est un simulateur d'envoi de messages FIX. Il est crucial pour tester la transmission d'ordres et d'autres communications basées sur ce protocole. Avec [Banzai](#), j'ai pu simuler des scénarios réels, tester les réponses du système et affiner mes tests.



GUI de Banzai



GUI de FIX Parser

L'élaboration des messages FIX a été un défi majeur. Souvent, j'ai dû me référer à la documentation du protocole FIX 4.4, analysant chaque tag et chaque message pour garantir leur exactitude. Cette étape a été rendue encore plus complexe par l'absence de messages prédéfinis dans les spécifications de test sur JIRA. Chaque message devait être construit à partir de zéro, nécessitant une attention méticuleuse aux détails.

Tout d'abord, il est essentiel pour moi de vérifier sur le "[FIX Parser](#)" que le format des messages FIX que j'ai rédigés est correct. Une fois que je suis certain que mon message FIX peut être correctement interprété (indépendamment des éventuelles erreurs de logique métier), j'utilise "Banzai" pour envoyer une requête FIX à SMS. En général, je reçois presque immédiatement une réponse FIX à ma commande. Selon le type de réponse reçue (déterminé par les champs 35 et 150 du corps du message), je peux évaluer si la fonctionnalité fonctionne comme prévu. Plus précisément, nous nous attendons à ce qu'un message FIX correct reçoive une exécution appropriée, tandis qu'une demande FIX incorrecte devrait être refusée. Enfin, je dois transcrire ces opérations en code, finalisant ainsi le [développement d'un test FIX](#).

```
@Test  
 @Order(120)  
 @DisplayName("STQA-12541 With a fix client send an order limit GTC on  
 public void OrderSSIWithFIXTest12() throws NoSuchFieldException {  
     activateFIXTradeSession();  
     final Message fixMessage = FillerUtility.stringToMessage(  
         "8=FIX.4.4|9=241|35=D|34=23|49=FCL1.TRD|52=20221108-14:07:  
     sendFixMessageAndOrderRejected(fixMessage);  
     stopFIXTradeSession();  
 }
```

Un JUnit test FIX

En règle générale, pour chaque type de commande, nous disposons d'un exemple de FIX correctement exécuté ainsi que de nombreux cas de FIX incorrects. Ces demandes de messages FIX erronés devraient toutes être rejetées. Nous souhaitons tester autant de scénarios d'erreur que possible pour assurer que le système reste sécurisé et stable, même en l'absence de protections d'interface utilisateur. Cependant, à la suite de mes tests, j'ai découvert que de nombreux messages FIX incorrects étaient acceptés par le système. Ces acceptations représentent des risques potentiels pour la fonctionnalité SSI. C'est pour cette raison que la fonctionnalité SSI est toujours en phase bêta.

c. Analyse et retours d'expérience

Les tests FIX ont été une source riche d'enseignements. Ils ont mis en lumière de nombreux bugs, comme la possibilité d'ajouter deux SSI de financement pour une seule commande. Ces découvertes ont été soigneusement documentées pour action.

Mais au-delà des bugs, le processus m'a offert des leçons inestimables. La plus marquante est l'importance de la communication. En discutant avec mes collègues, j'ai pu surmonter de nombreux obstacles. Cette expérience m'a rappelé que la collaboration est aussi cruciale que la compétence technique. De plus, la nécessité de comprendre en profondeur le protocole FIX m'a permis d'acquérir une expertise précieuse, renforçant ma capacité à développer des tests robustes et fiables.

Au cours de mon stage, j'ai travaillé intensivement sur plusieurs aspects clés du développement et de l'automatisation des tests pour le système smartTrade. Voici un aperçu des points :

- Framework d'automation : Je me suis familiarisé avec le framework d'automation existant, posant ainsi les bases de mon travail ultérieur.

- Analyse du flow pour SSI :

Dans SmartCC2 : J'ai étudié en profondeur l'interface utilisateur de SmartCC2 et comment elle a été abstraite pour faciliter les tests. J'ai notamment exploré comment les SSI sont créés, modifiés et supprimés.

Dans Cockpit : J'ai analysé comment les SSI sont gérés dans Cockpit, en me concentrant sur les pop-ups et les interfaces intégrées, comme celle de l'Order Editor.

- Développement de Test de JUnit : J'ai souligné l'importance des suites de tests pour assurer la qualité du logiciel. J'ai détaillé le processus de création des tests, en donnant un exemple concret de la manière dont les étapes de test sur JIRA sont transformées en code.

J'ai également abordé l'exécution des suites de tests sur Bamboo, montrant comment les tests sont exécutés et comment les détails des tests sont affichés.

- Développement de Test de FIX : Après avoir rappelé les principes du protocole FIX et comment il est intégré chez smartTrade, j'ai décrit le processus de développement des tests FIX.

J'ai utilisé des outils comme Banzai et FIX Parser pour aider à la création et à la validation des messages FIX.

Enfin, j'ai partagé une analyse et des retours d'expérience sur l'ensemble du processus, soulignant les défis rencontrés et les leçons apprises.

En somme, j'ai démontré une compréhension approfondie des outils et des processus utilisés chez smartTrade, et j'ai contribué de manière significative à l'amélioration de la qualité du logiciel grâce à mes efforts en matière de développement et d'automatisation des tests.

V. Analyses managériales

1. RSE

a. Introduction

La Responsabilité Sociale et Environnementale (RSE) chez smartTrade Technologies n'est pas simplement une obligation, mais une conviction. C'est une approche pour montrer notre engagement envers la société et l'environnement, pour contribuer de manière positive aux communautés dans lesquelles nous opérons et pour maintenir un environnement de travail favorable pour nos employés.

b. Engagement envers les Employés

Événements pour Employés

A smartTrade, la création d'un environnement de travail positif est cruciale. Nous organisons régulièrement des événements pour renforcer l'esprit d'équipe, encourager la collaboration et garantir le bien-être mental et physique des employés. Par exemple, nous avons une ligue de football à cinq à Londres qui rassemble des personnes de l'industrie financière, et des tournois de padel après le travail au sein de l'entreprise.

c. Actions Caritatives

Soutien aux Organisations Caritatives



Nous croyons au pouvoir de la communauté. Nous sommes fiers de soutenir et d'investir notre temps et nos ressources dans plusieurs organisations caritatives. Par

exemple, nous avons un programme de bénévolat corporatif avec 'Brands for Canada' pour donner des vêtements et des articles de soins personnels aux personnes vivant dans la pauvreté. Ici à Aix-en-Provence, nous avons également participé à une course sur route pour collecter des fonds pour "La Maison", un centre de soins palliatifs en France.

d. Protection de l'Environnement

Nettoyage de Plage

smartTrade est fier de soutenir l'association française "Les Perles Côte Bleue". Cette association de 10 ans agit pour la préservation et la protection de notre environnement, la biodiversité maritime, terrestre, la faune et la flore. Nos employés ont participé à leurs côtés à une opération de nettoyage de la plage et des littoraux.



Partenariat Tree Nation

En renouvelant notre engagement en tant qu'entreprise respectueuse de l'environnement, nous avons mis en place un nouveau projet avec Sociabble et Tree-Nation. Depuis juillet 2019, smartTrade et ses employés soutiennent divers projets de reforestation à travers le monde, comme au Burkina Faso et au Brésil, pour lutter contre la désertification et l'épuisement des sols tout en aidant à la conservation de la biodiversité.



e. Initiatives Communautaires

Sourire à la vie et Défi Full Ace

Nous lançons des défis sportifs pour soutenir des causes charitables. Par exemple, le 'smartTrade Challenge' pendant l'Open Féminin de Marseille et le "smartTrade Full Ace Challenge" pendant l'"Open du Pays d'Aix" collectent des fonds pour des associations comme 'Sourire à la Vie' et 'La chaîne de l'espoir', respectivement. Ces défis sportifs permettent de collecter des fonds pour aider les enfants atteints de cancer et fournir des soins médicaux et de l'éducation aux enfants dans les pays en développement.



**La chaîne
de l'espoir**

2. Management de l'innovation

a. Définition et importance du management de l'innovation

Le management de l'innovation désigne l'ensemble des démarches, techniques et outils mis en place par une entreprise pour piloter et soutenir ses activités d'innovation. Dans un environnement de marché de plus en plus compétitif, l'innovation est essentielle pour garantir la pérennité et la croissance d'une entreprise. Elle permet de répondre aux besoins changeants des clients, d'explorer de nouveaux marchés et d'obtenir un avantage concurrentiel.

b. Stratégie et objectifs de smartTrade dans le domaine de l'innovation

Chez smartTrade, l'innovation est au cœur de la stratégie d'entreprise. La société cherche constamment à anticiper les besoins du marché et à développer des solutions avant-gardistes pour le trading électronique. L'objectif principal est de renforcer sa position de leader sur le marché tout en explorant de nouvelles opportunités, en particulier dans le domaine de l'intelligence artificielle.

c. Principales mesures et pratiques en matière d'innovation chez smartTrade

smartTrade ne se contente pas seulement d'encourager les idées innovantes, mais va plus loin en instituant une culture d'innovation systématique :

Plateforme de propositions innovantes : Une plateforme est mise à la disposition des employés où ils peuvent soumettre et discuter des idées innovantes. Cette démarche démocratise l'innovation et permet une réflexion collective.

Ateliers interdisciplinaires : Au-delà des réunions hebdomadaires, des ateliers interdisciplinaires sont organisés pour permettre aux équipes de différents départements de collaborer et de fusionner leurs compétences.

Expansion vers l'intelligence artificielle : smartTrade a lancé un programme d'exploration de l'IA visant à intégrer des solutions d'IA dans ses systèmes existants pour améliorer l'efficacité des interactions entre le client et son trader.

e. Analyse du management de l'innovation chez smartTrade

L'approche de smartTrade en matière de gestion de l'innovation est à la fois proactive et réactive :

Analyse de l'écosystème technologique : L'entreprise analyse en permanence l'écosystème technologique pour identifier les opportunités d'innovation qui peuvent être exploitées pour améliorer ses services et solutions.

Évaluation de l'impact de l'IA : L'entreprise a entrepris une évaluation approfondie pour comprendre comment l'IA peut révolutionner ses opérations, en témoigne son initiative d'intégrer l'IA dans les interactions entre le client et son trader.

Retour sur investissement de l'innovation : Les retours des initiatives innovantes sont évalués pour comprendre leur impact et pour garantir que l'innovation chez smartTrade est alignée avec ses objectifs stratégiques à long terme.

f. Points clés et conclusions des entretiens avec les cadres supérieurs et les départements concernés

Lors de mes discussions avec le DRH et le Directeur de R&D, ils ont tous deux souligné que l'innovation est portée par le haut management, mais qu'il est essentiel que chaque employé se sente impliqué et encouragé à contribuer. L'ouverture des responsables de développement aux nouvelles idées et l'importance accordée aux feedbacks des équipes reflètent cette culture d'innovation.

g. Recommandations pour améliorer la stratégie et les pratiques d'innovation

Mise en place d'un laboratoire d'innovation: smartTrade a créé un espace dédié, nommé laboratoire d'innovation, où les équipes ont la liberté de travailler sur des projets pilotes et de tester de nouvelles idées. Cet environnement stimulant favorise l'exploration et l'émergence d'innovations qui peuvent ensuite être intégrées dans nos opérations courantes.

Établissement de partenariats solides avec des institutions académiques: Afin de rester à la pointe des nouvelles technologies et des approches émergentes, smartTrade a établi des partenariats avec des universités et des centres de recherche renommés. Ces collaborations permettent un échange de connaissances bénéfique et assurent que nous sommes toujours au courant des dernières avancées dans notre domaine.

Instauration d'un système de reconnaissance pour l'innovation : La reconnaissance et la récompense des employés qui contribuent de manière significative à l'innovation sont des priorités chez smartTrade. Nous avons mis en place un système de reconnaissance qui valorise les efforts et les contributions de nos employés en matière d'innovation, encourageant ainsi une culture d'innovation continue au sein de l'entreprise.

VI. Conclusion

Au cours de mon stage chez smartTrade, j'ai été principalement chargé de développer des tests automatisés pour la fonctionnalité SSI (Standing Settlement Instructions) et SSP (Single Spot Portfolio) en utilisant le protocole FIX. L'objectif principal était d'assurer que ces fonctionnalités, essentielles pour le logiciel de trading SMS, fonctionnent de manière optimale et conforme aux attentes.

Travaillant en présentiel au sein de l'entreprise, j'ai eu l'opportunité d'utiliser quotidiennement des outils professionnels tels que JIRA et Eclipse. Ces outils m'ont aidé à structurer et à suivre mon travail de manière efficace. De plus, j'ai souvent dû travailler avec des chemins html, les convertissant au format XPath pour localiser précisément les éléments sur lesquels j'intervenais.

Cette immersion complète m'a permis de me familiariser avec les méthodes agiles employées chez smartTrade et de comprendre les routines et les attentes des employés. C'était une expérience professionnelle enrichissante, me permettant de me confronter directement aux défis du monde réel de l'ingénierie logicielle.

L'une de mes réalisations majeures a été la maîtrise du framework utilisé chez smartTrade. J'ai appris à comprendre sa structure, son fonctionnement et j'ai même eu l'opportunité d'y ajouter de nouvelles fonctionnalités tout en préservant son intégrité. Grâce à mes efforts, une centaine de méthodes ont été intégrées au framework, renforçant ainsi sa capacité et sa flexibilité.

Le feedback que j'ai reçu de l'équipe et de l'entreprise a été positif. Mon travail sur le développement des tests, en particulier pour les fonctionnalités SSI et SSP via FIX, a répondu aux besoins de l'entreprise. Je suis confiant que les tests que j'ai développés continueront d'être utilisés pour garantir la qualité et la fiabilité de ces fonctionnalités essentielles du logiciel SMS.

En somme, ce stage chez smartTrade a été une étape cruciale dans mon parcours professionnel, me fournissant des compétences pratiques et une compréhension approfondie des défis et des opportunités dans le domaine du développement de logiciels de trading.

VII. Référence

1. <https://www.swift.com/fr/node/47026>
2. <https://www.atlassian.com/fr>
3. <https://www.ecologie.gouv.fr/responsabilite-societale-des-entreprises>
4. <https://www.atlassian.com/fr/software/bamboo>
5. <https://www.atlassian.com/fr/software/confluence>
6. <https://fr.wikipedia.org/wiki/Forex>
7. <https://www.selenium.dev/documentation/>
8. <https://www.eclipse.org/documentation/>
9. <https://www.kms.fr/v5/public/courses/4938>
10. <https://smart-trade.net/esg/>
11. https://help.sap.com/docs/SAP_PAYMENT_ENGINE/d38a93482a2b492694c2acaac3372566/f6dfb545822d4383a47844ccba619607.html
12. <https://www.youtube.com/watch?v=RU9rDfa2UqQ>