

**INSTITUT TECCART**



## **RAPPORT DE STAGE DE FIN D'ETUDES**

Présenté en vue de l'obtention du

Diplôme d'études collégiales (DEC)

**TECHNIQUE EN INFORMATIQUE DE GESTION.**

**PAR**

Nom de l'étudiant

Intégration et Innovation en Environnement Cloud : Exploration de Streamlit, Mongo Atlas et AWS lors d'un  
Stage en Développement de Solutions Interactives

**Entreprise d'accueil**

ICE-CURRENCY EXCHANGE

**Encadrant à l'entreprise :** M. Glenn Osei

**Encadrant à l'institut:** M. Safouen Bani

## **Remerciement**

Je tiens à exprimer ma profonde gratitude envers ceux qui ont rendu mon stage chez Ice Currency à la fois enrichissant et instructif.

Tout d'abord, je souhaite remercier M. Safouen Bani, mon encadrant de stage, pour son accompagnement, ses conseils précieux et son soutien constant tout au long de cette expérience. Son expertise et sa pédagogie m'ont permis de développer mes compétences de manière significative.

Je tiens également à remercier chaleureusement M. Wassim Boualamallah, le Directeur Général de l'entreprise, pour m'avoir donné l'opportunité de rejoindre leur équipe. Sa vision inspirante a grandement enrichi mon expérience professionnelle.

Enfin, un merci spécial à M. Glenn Osei, mon superviseur, pour sa patience et sa disponibilité. Grâce à son encadrement au quotidien, j'ai pu m'immerger pleinement dans le milieu professionnel et surmonter les défis rencontrés.

Cette expérience a été une étape déterminante de mon parcours professionnel et les leçons apprises seront un atout pour ma future carrière.

## Table des matières

<b>CHAPITRE 1: Etude de l'existant et solution proposée .....</b>	<b>7</b>
1.1.Problématique.....	7
1.2.Solution.....	8
<b>CHAPITRE 2 : Etude conceptuelle .....</b>	<b>10</b>
2.1.Description générale .....	10
2.2.Conception de classes .....	11
2.2.1.Classes et Interfaces .....	11
2.2.2.Relations entre Classes .....	13
2.2.3.Glossaire .....	13
2.3.Les cas d'utilisation.....	13
2.3.1.Acteurs.....	14
2.3.2.Cas d'utilisation:.....	14
2.3.3.Relations:.....	15
2.4.Diagramme de séquence.....	15
2.4.1.Entités impliquées.....	16
2.4.2.Processus d'Authentification .....	16
2.5.Spot.....	17
2.5.1.Acteurs : .....	18
2.5.2.Étapes du Processus .....	18
2.6.Spreads .....	22
2.6.1.Acteurs du système :.....	23
2.6.2.Processus Décrit par le Diagramme .....	23
2.7.Rates .....	25
2.7.1.Acteurs impliqués .....	25
2.7.2.Fonctionnalités clés .....	26
<b>CHAPITE 3 : UI/UX de l'application Streamlit.....</b>	<b>27</b>
3.1.Structure de l'application.....	27
3.1.1.Authentification et Accueil.....	27
3.1.2.Navigation .....	28
3.1.3.Interaction et visualisation des données .....	29
3.1.4.Validation de toutes les fonctionnalités : .....	29
3.1.5.Éléments d'Interface (UI) .....	38
3.1.6.Accessibilité.....	39
3.2.Conclusion .....	39
<b>CHAPITRE 4 : Conception de la base de données.....</b>	<b>40</b>
4.1. Structure de la Base de Données .....	40
4.1.1. Table Spot.....	40
4.1.2. Table Spreads .....	40
4.1.3. Table Rates .....	41
4.2. Relations entre les Tables .....	41

4.3. Considérations Techniques.....	41
4.4. Hébergement sur MongoDB Atlas .....	41
4.5. Conclusion .....	42
<b>CHAPITRE 5 : Déploiement de l'application Streamlit sur Heroku.....</b>	<b>43</b>
5.1.URL de notre application :.....	43
5.2.Prérequis.....	43
5.3.Fichiers du projet.....	43
5.4.Étapes de déploiement .....	44
5.5.Conclusion .....	45
<b>CHAPITRE 6 : Utilisation de REST API avec lambda de AWS.....</b>	<b>46</b>
6.1.Configuration Préliminaire .....	46
6.1.1.Environnement MongoDB Atlas .....	46
6.1.2.AWS Lambda .....	46
6.1.3.Dépendances .....	46
6.1.4.API Endpoints.....	47
6.1.5.Format de la Réponse.....	47
6.1.6.Gestion des Erreurs .....	48
6.1.7.Sécurité .....	48
6.2.Conclusion .....	48
<b>7- Conclusion générale et perspectives .....</b>	<b>49</b>
7.1.Affrontements et Surmontements des Problèmes .....	49
7.2.Contributions et Impacts .....	49
7.3.Perspectives d'Amélioration.....	50
7.4.Visions Futures .....	50

## Liste des Diagrammes

<a href="#"><u>Diagramme 1: Diagramme de classe (UML)</u></a> .....	11
<a href="#"><u>Diagramme 2 : Diagramme de cas d'utilisation</u></a> .....	13
<a href="#"><u>Diagramme 3 : Diagramme de séquence (authentification)</u></a> .....	15
<a href="#"><u>Diagramme 4 : Diagramme de séquence (Spot)</u></a> .....	17
<a href="#"><u>Diagramme 5 : Diagramme de séquence ( Option insérer une devise manuellement)</u></a> .....	19
<a href="#"><u>Diagramme 6 : Diagramme de séquence ( Option corriger une valeur )</u></a> .....	20
<a href="#"><u>Diagramme 7 : Diagramme de séquence (Option Visualiser les données)</u></a> .....	21
<a href="#"><u>Diagramme 8 : Diagramme de séquence (Spreads)</u></a> .....	22
<a href="#"><u>Diagramme 9 : Diagramme de séquence (Rates)</u></a> .....	25



## **CHAPITRE 1: Étude de l'existant et solution proposée**

### **I. Introduction**

Durant mon stage au sein de Ice Currency, une compagnie renommée dans le domaine des échanges monétaires, j'ai eu l'opportunité privilégiée de collaborer étroitement avec le Directeur Général de la succursale de Québec. Cette expérience m'a permis non seulement de me familiariser avec le fonctionnement interne de l'entreprise mais aussi de participer à des discussions stratégiques cruciales. Au cours d'un entretien approfondi avec le Directeur, nous avons examiné les défis significatifs auxquels l'entreprise est confrontée, notamment en termes de gestion des taux de change du marché (spot) et des marges de vente et d'achat (spreads). Il est rapidement devenu évident que la mise en place d'un système de gestion de données performant et agile était indispensable pour assurer la précision et la rapidité requises dans le calcul quotidien des taux de vente et d'achat offerts aux clients. Cette prise de conscience a enrichi ma compréhension des enjeux financiers et opérationnels et a souligné l'importance de l'innovation technologique dans le maintien de la compétitivité de l'entreprise. Ce stage a été une occasion inestimable de mettre en pratique mes connaissances académiques en finance et en gestion, tout en contribuant à des projets qui ont des répercussions tangibles sur le succès de l'entreprise.

#### **I.1. Problématique**

L'analyse de la situation actuelle chez Ice Currency révèle une problématique majeure liée à la gestion des données financières, notamment celles concernant les taux de change du marché (spot) et les marges de vente et d'achat (spreads). À l'heure actuelle, ces données essentielles sont collectées, stockées et traitées via des méthodes manuelles ou semi-automatisées. Cette approche présente plusieurs inconvénients significatifs :

**Risques d'Erreur Accrus** : L'intervention manuelle et les processus semi-automatisés augmentent le risque d'erreurs dans les données, ce qui peut conduire à des décisions commerciales inappropriées ou à des informations inexactes fournies aux clients.

**Retards Opérationnels** : Le manque d'automatisation entraîne des délais dans le traitement des données, retardant ainsi la mise à jour des taux de change et des spreads. Ce retard peut nuire à la capacité de l'entreprise à réagir rapidement aux fluctuations du marché, une compétence essentielle dans l'industrie des changes.

**Complexité dans le Calcul des Taux** : Les difficultés inhérentes au traitement manuel ou semi-automatisé des données compliquent le calcul précis des taux de vente et d'achat, affectant directement la compétitivité des offres de l'entreprise.

Ces défis compromettent non seulement la compétitivité d'Ice Currency sur le marché des changes, mais menacent également la satisfaction et la fidélité de ses clients. Il est donc impératif d'envisager des solutions pour optimiser la gestion des données financières afin d'améliorer les performances opérationnelles et de maintenir un avantage concurrentiel dans un environnement de marché dynamique.

## **I.2. Solution**

Pour répondre à cette problématique, il est proposé de développer un système de gestion de données dédié aux taux de change du marché et aux marges de vente et d'achat. Ce système sera conçu pour automatiser la collecte, le stockage et le traitement des données, en utilisant une infrastructure moderne et évolutive.

La solution envisagée consistera en la mise en place d'une base de données MongoDB Atlas, qui offrira la flexibilité nécessaire pour gérer efficacement les données complexes liées aux taux de change et aux marges. Cette base de données sera alimentée par des flux de données provenant de diverses sources, telles que les marchés financiers et les plateformes de trading.

Parallèlement, un site web sera développé à l'aide de Streamlit, offrant une interface conviviale permettant aux utilisateurs d'accéder facilement aux données et aux



fonctionnalités du système. Enfin, une API AWS sera utilisée pour faciliter l'accès et l'intégration des données avec d'autres applications et systèmes internes de l'entreprise.

En mettant en œuvre cette solution, Ice Currency pourra améliorer la précision, la rapidité et la fiabilité dans le calcul des taux de vente et d'achat, renforçant ainsi sa compétitivité sur le marché tout en garantissant la satisfaction de sa clientèle.

## **CHAPITRE 2 : Etude conceptuelle**

### **II. La conception de diagramme de classe :**

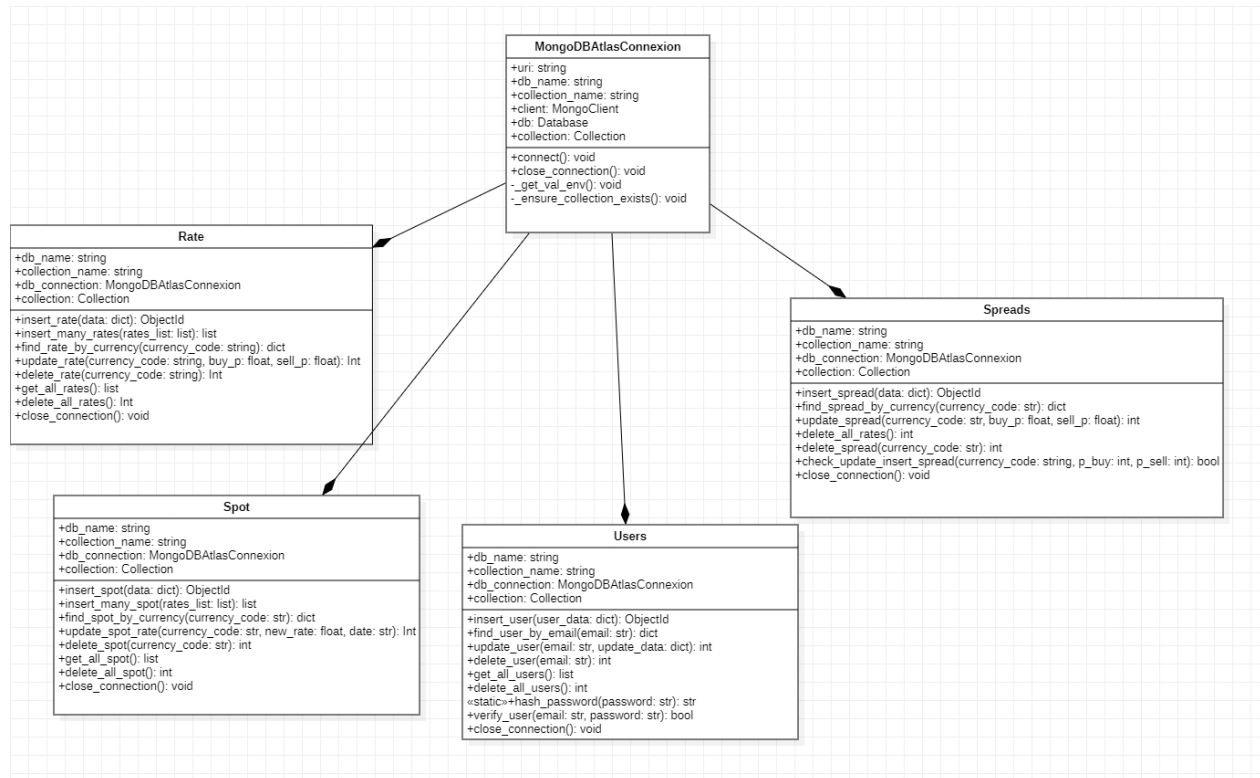
Dans cette section, nous explorerons la conception et la structure des classes déployées pour gérer les connexions à MongoDB Atlas et effectuer des opérations sur des données cruciales pour notre application. Nous plongerons dans les détails des mécanismes sous-jacents qui facilitent l'insertion, la récupération, la mise à jour et la suppression de données vitales telles que les taux, les spots, les spreads et les informations utilisateur. Ce chapitre fournira un aperçu approfondi des fondements techniques nécessaires à la manipulation efficace de ces données au sein de notre environnement applicatif.

#### **II.1. Description générale**

- **Contexte :** Application de gestion de données financières.
- **Acteurs principaux :** Administrateurs de base de données, utilisateurs finaux accédant aux données financières.

## II.2. Conception de classes

Diagramme 1: Diagramme de classe (UML)



### II.2.1. Classes et Interfaces

#### MongoDBAtlasConnexion

- **Description** : Gère la connexion à une base de données MongoDB Atlas, en permettant de sélectionner une collection spécifique pour les opérations de données.
- **Attributs** :
  - **uri : string** - L'URI de connexion à MongoDB Atlas.
  - **db\_name : string** - Le nom de la base de données cible.
  - **collection\_name : string** - Le nom de la collection cible.
  - **client : MongoClient** - L'instance du client de connexion.

- **db : Database** - L'instance de la base de données sélectionnée.
- **collection : Collection** - L'instance de la collection sélectionnée.
- **Méthodes :**
  - **connect(): void** - Établit la connexion à la base de données.
  - **close\_connection(): void** - Ferme la connexion établie.
  - **\_ensure\_collection\_exists(): void** - Assure que la collection spécifiée existe.
  - **\_get\_val\_env(): string** - Récupère l'URI de connexion depuis les variables d'environnement.

## Spot

- **Description :** Permet la manipulation des données de spots dans la base de données.
- **Attributs et Méthodes similaires à ceux décrits pour MongoDBAtlasConnexion** avec des opérations spécifiques aux spots.

## Rate

- **Description :** Gère les opérations sur les taux de change.
- **Attributs et Méthodes similaires à ceux décrits pour MongoDBAtlasConnexion** avec des opérations spécifiques aux taux.

## Spreads

- **Description :** Permet la manipulation des spreads dans la base de données.
- **Attributs et Méthodes similaires à ceux décrits pour MongoDBAtlasConnexion** avec des opérations spécifiques aux spreads.

## Users

- **Description :** Gère les opérations sur les utilisateurs de l'application.

- **Attributs et Méthodes similaires à ceux décrits pour MongoDBAtlasConnexion** avec des opérations spécifiques aux utilisateurs, y compris le hachage et la vérification des mots de passe.

### II.2.2. Relations entre Classes

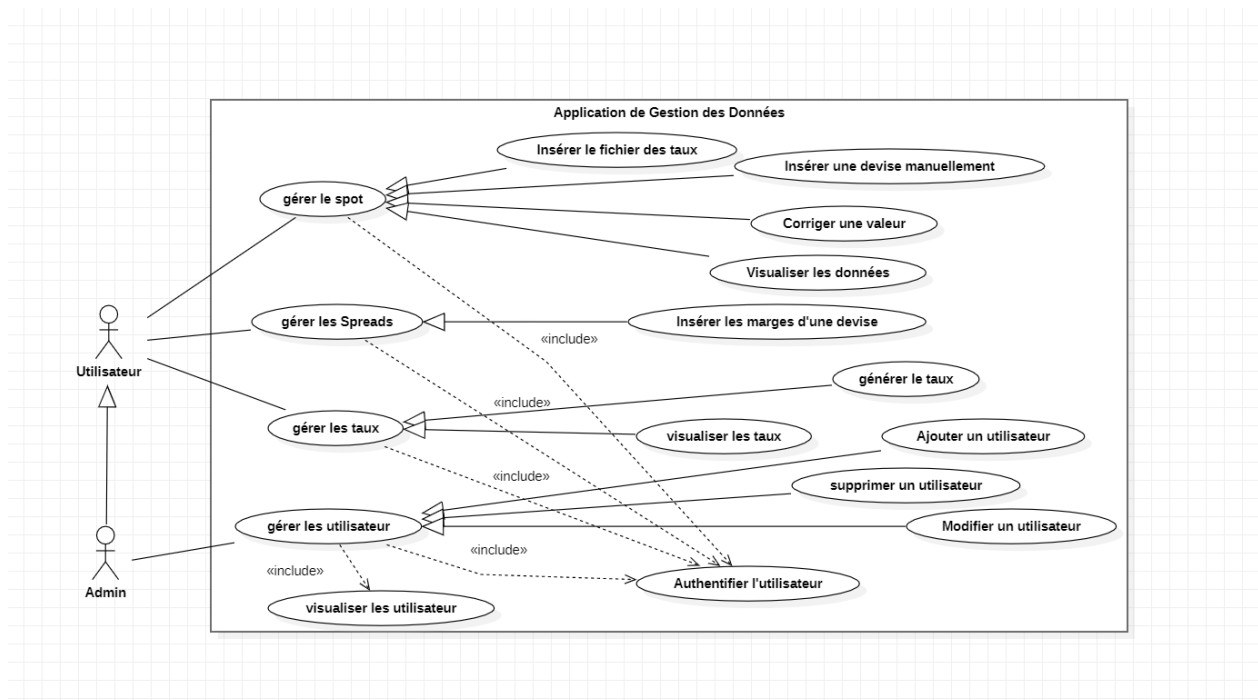
Toutes les classes (**Spot**, **Rate**, **Spreads**, **Users**) ont une **composition** avec **MongoDBAtlasConnexion**, utilisant une instance de cette classe pour établir des connexions à la base de données et exécuter des opérations spécifiques.

### II.2.3. Glossaire

- **MongoClient** : Classe du package **pymongo** pour interagir avec une base de données MongoDB.
- **SHA256** : Algorithme de hachage utilisé pour sécuriser les mots de passe.

## II.3. Les cas d'utilisation

**Diagramme 2 : Diagramme de cas d'utilisation**



### II.3.1. Acteurs

- **Utilisateur:** Peut gérer le spot, les spreads, et les taux. Il peut aussi visualiser les taux et les données, insérer les marges d'une devise, générer le taux, corriger une valeur, insérer une devise manuellement et insérer le fichier des taux.
- **Admin:** Peut gérer les utilisateurs (ajouter, supprimer, modifier), visualiser les utilisateurs, et authentifier l'utilisateur.

### II.3.2. Cas d'utilisation:

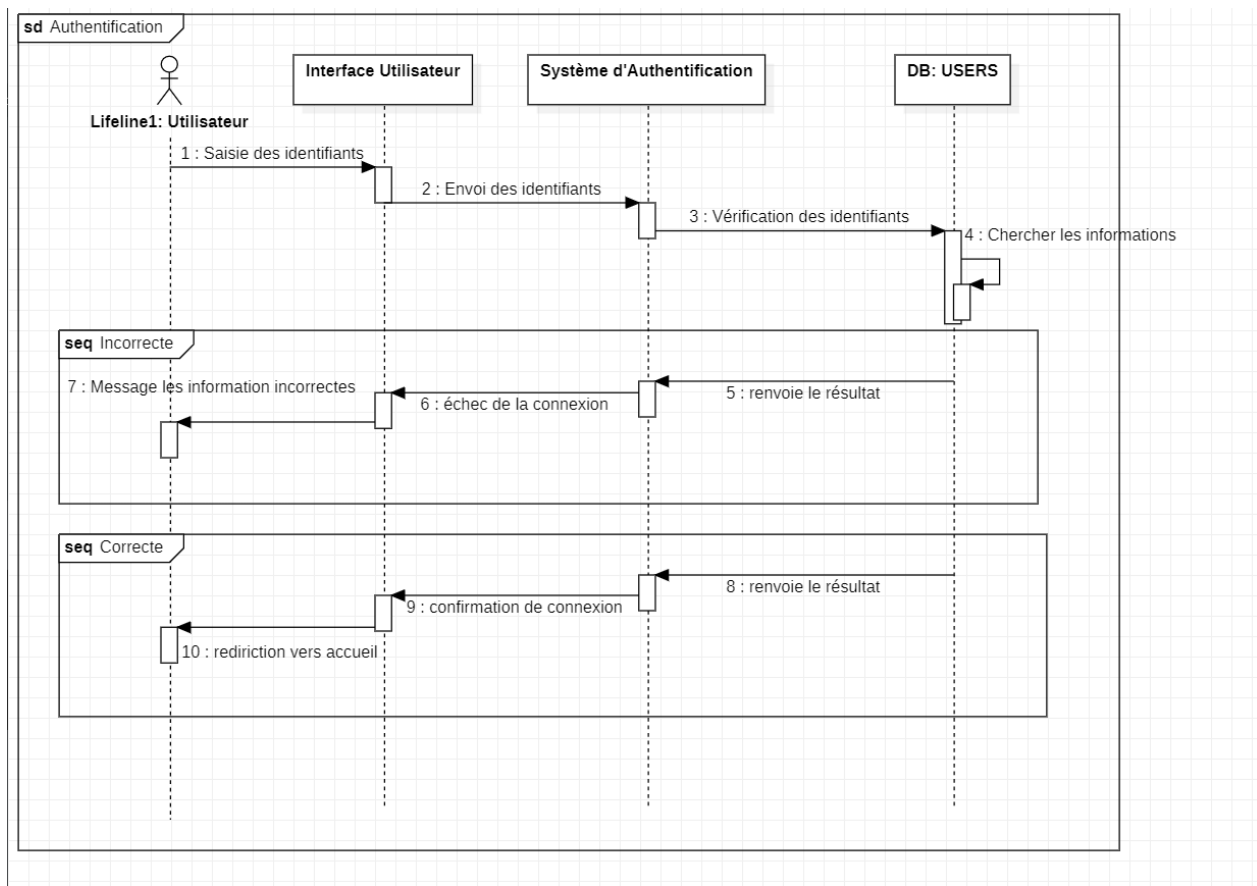
- **Gérer le spot :** Inclut les tâches relatives à la gestion des taux de change au comptant.
- **Gérer les spreads :** Inclut les opérations de gestion des écarts de prix ou des commissions.
- **Gérer les taux :** Comprend l'insertion, la visualisation, et la correction des taux de change.
- **Visualiser les taux :** Permet à l'utilisateur de voir les taux de change actuels ou historiques.
- **Insérer les marges d'une devise :** Inclut l'ajout de données spécifiques relatives aux marges pour une devise donnée.
- **Générer le taux :** Inclut le calcul ou la récupération d'un taux de change.
- **Visualiser les données :** Permet de consulter l'ensemble des données de l'application.
- **Insérer une devise manuellement :** Permet d'ajouter des informations sur une nouvelle devise qui n'est pas actuellement dans le système.
- **Corriger une valeur :** Permet de modifier des données existantes en cas d'erreur ou de changement.
- **Insérer le fichier des taux :** Permet l'importation en masse des taux de change via un fichier.
- **Authentifier l'utilisateur :** Processus de vérification de l'identité de l'utilisateur avant de lui permettre d'accéder aux fonctionnalités.
- **Gérer les utilisateurs :** Ensemble des fonctions liées à la création, la suppression, et la modification des comptes d'utilisateurs.

### II.3.3. Relations:

- Les relations « include » (inclure) indiquent que la tâche inclut automatiquement la tâche connectée par la flèche. Par exemple, "générer le taux" inclut "visualiser les taux".
- Les relations avec les lignes pointillées montrent que l'action est optionnelle ou une extension de la fonctionnalité principale. Par exemple, "gérer les taux" peut ou non inclure "authentifier l'utilisateur" selon le contexte.

### II.4. Diagramme de séquence

**Diagramme 3 :** Diagramme de séquence (authentification)



Ce diagramme de séquence présenté illustre le processus d'authentification d'un utilisateur dans un système informatique. Ce processus est une interaction entre quatre entités principales : l'Utilisateur, l'Interface Utilisateur, le Système d'Authentification et la base de

données des utilisateurs (DB: USERS). Le diagramme décrit deux scénarios possibles lors de l'authentification : l'échec de la connexion avec des identifiants incorrects et la réussite de la connexion avec des identifiants corrects.

#### **II.4.1. Entités impliquées**

- **Utilisateur (Lifeline1):** La personne qui tente de se connecter au système.
- **Interface Utilisateur:** Le frontend par lequel l'Utilisateur interagit avec le système.
- **Système d'Authentification:** Le backend qui traite les informations d'identification et valide l'accès.
- **DB: USERS:** La base de données contenant les informations d'identification des utilisateurs enregistrés.

#### **II.4.2. Processus d'Authentification**

##### 1. Saisie des Identifiants

- L'Utilisateur commence le processus en saisissant ses identifiants dans l'Interface Utilisateur.

##### 2. Envoi des Identifiants

- L'Interface Utilisateur soumet les identifiants saisis au Système d'Authentification.

##### 3. Vérification des Identifiants

- Le Système d'Authentification reçoit et vérifie les identifiants.

##### 4. Recherche des Informations

- Le Système d'Authentification interroge la DB: USERS pour valider les identifiants.

- **Scénario d'Échec (seq Incorrecte)**

##### 5. Renvoi du Résultat (Échec)

Si les identifiants sont incorrects, le Système d'Authentification retourne un résultat négatif.



## 6.Échec de la Connexion

L'Interface Utilisateur reçoit la notification d'échec.

## 7.Message d'Erreur

L'Utilisateur est informé que les informations saisies sont incorrectes via un message d'erreur.

- **Scénario de Réussite (seq Correcte)**

## 8.Renvoi du Résultat (Succès)

Si les identifiants sont corrects, le Système d'Authentification retourne un résultat positif.

## 9.Confirmation de Connexion

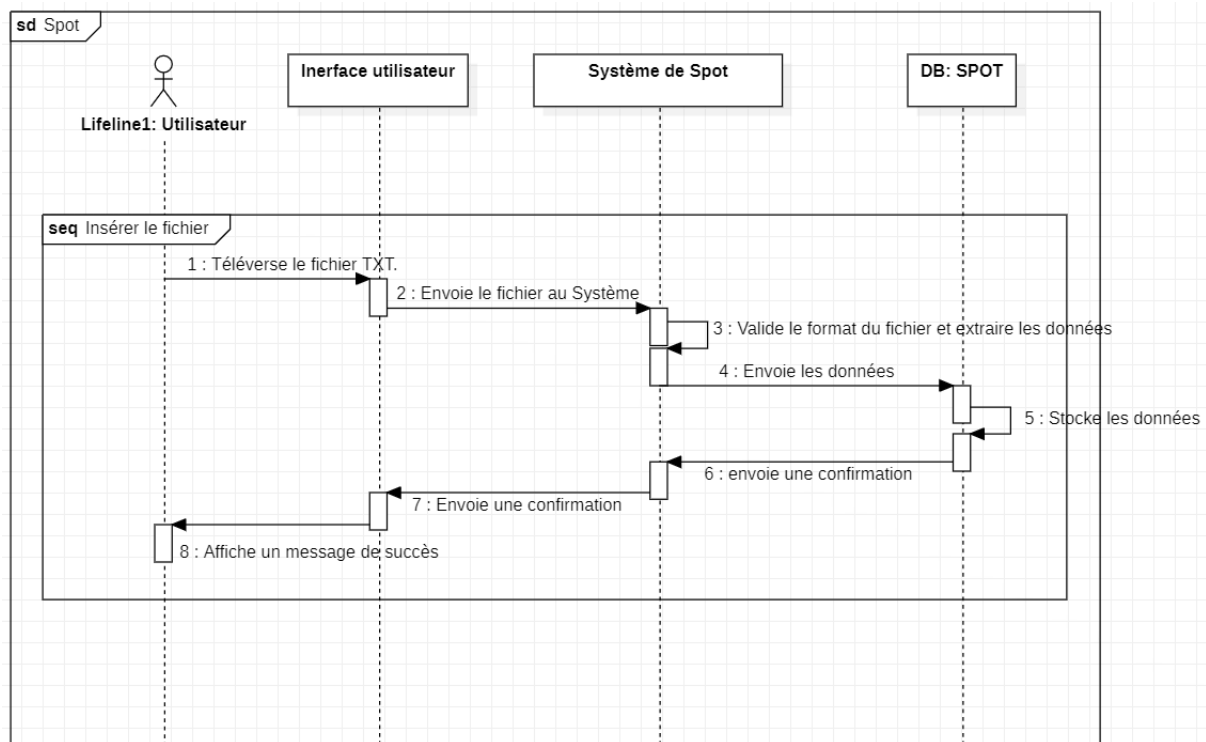
L'Interface Utilisateur reçoit la confirmation de connexion réussie.

## 10.Redirection vers l'Accueil

L'Utilisateur est redirigé vers la page d'accueil ou l'espace utilisateur du système.

## II.5. Spot

**Diagramme 4 : Diagramme de séquence (Spot)**



Ce diagramme de séquence décrit le processus par lequel un utilisateur charge un fichier texte dans le système de Spot et comment ce fichier est traité par le système jusqu'à son stockage dans la base de données SPOT. Le diagramme décrit les interactions entre l'Utilisateur, l'Interface Utilisateur, le Système de Spot et la base de données.

#### **II.5.1. Acteurs:**

- **Utilisateur (Lifeline1)** : La personne qui effectue l'action de téléchargement du fichier.
- **Interface Utilisateur** : La façade par laquelle l'Utilisateur interagit pour charger le fichier.
- **Système de Spot** : Le service en backend qui traite le fichier chargé.
- **(DB : SPOT)** : La base de données où les données extraites du fichier sont finalement stockées.

#### **II.5.2. Étapes du Processus :**

##### **1. Téléversement du Fichier**

- L'Utilisateur sélectionne et téléverse un fichier texte (.TXT) via l'Interface Utilisateur.

##### **2. Envoi du Fichier au Système**

- Après le téléversement, l'Interface Utilisateur envoie le fichier au Système de Spot.

##### **3. Validation et Extraction des Données**

- Le Système de Spot vérifie le format du fichier pour s'assurer qu'il correspond aux exigences et extrait ensuite les données nécessaires du fichier.

##### **4. Envoi des Données**

- Après l'extraction, les données sont envoyées à la DB: SPOT pour être stockées.

##### **5. Stockage des Données**

- La DB: SPOT reçoit et stocke les données extraites du fichier.

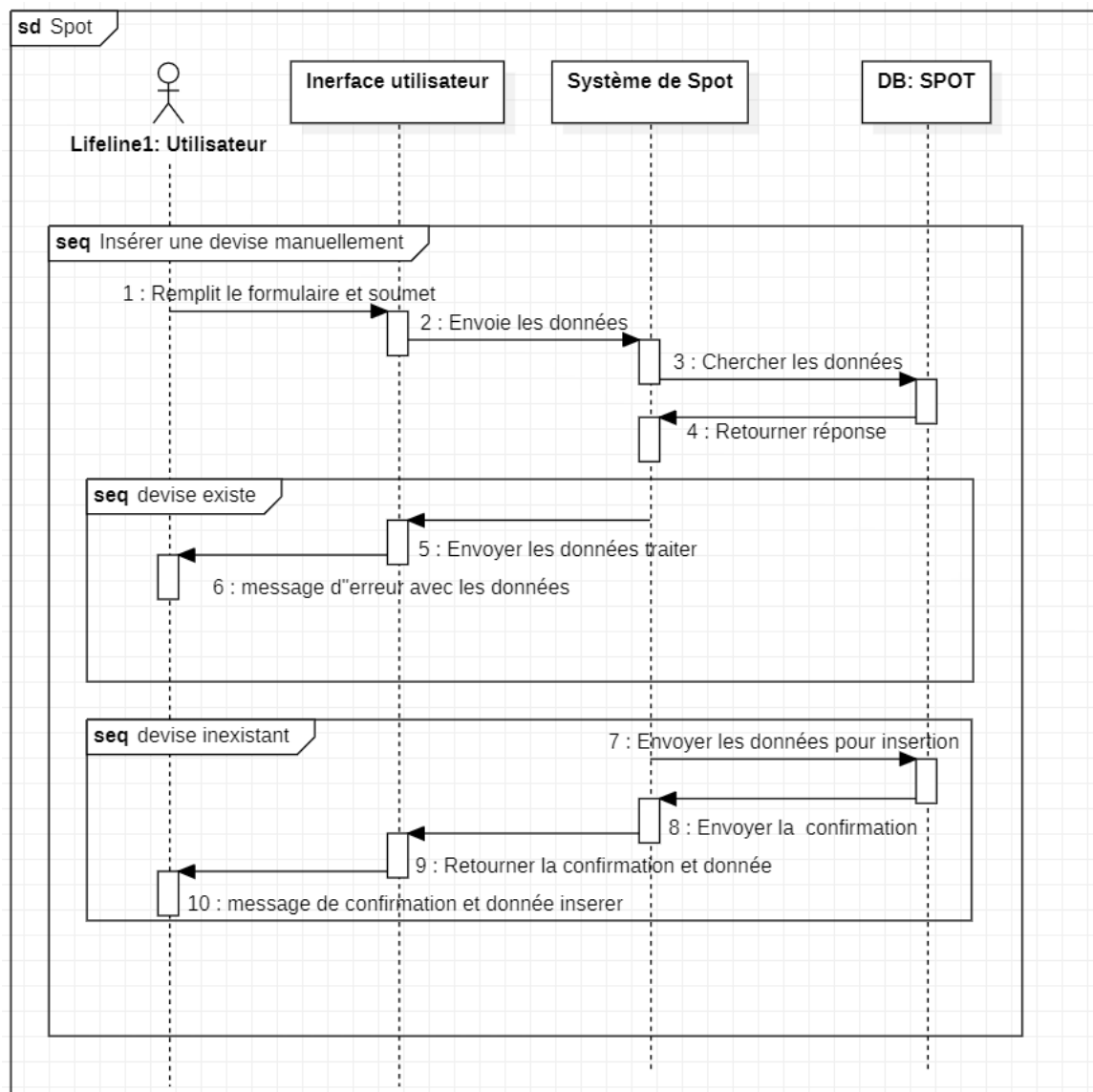
##### **6. Envoi de la Confirmation**

- Une fois les données stockées avec succès, le Système de Spot envoie une confirmation de succès à l'Interface Utilisateur.

## 7. Affichage du Message de Succès

- L'Interface Utilisateur reçoit la confirmation et affiche un message de succès à l'Utilisateur, indiquant que le fichier a été chargé et traité avec succès.

**Diagramme 5 :** Diagramme de séquence ( Option insérer une devise manuellement)



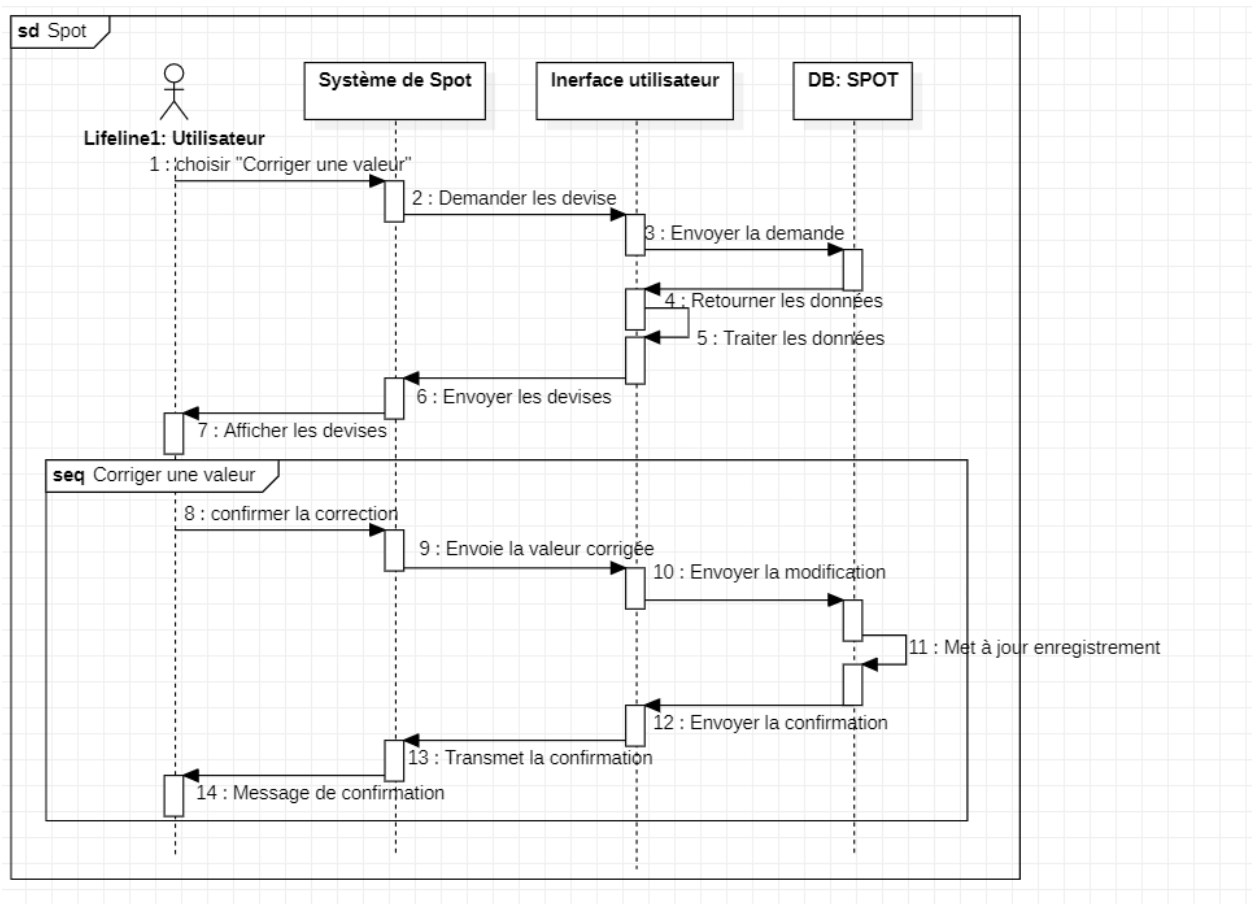
Ce diagramme de séquence UML détaillant les interactions entre un utilisateur, une interface utilisateur, un système appelé "Système de Spot" et une base de données (DB) étiquetée "SPOT".

Le processus commence lorsque l'utilisateur insère manuellement une devise dans le système via l'interface utilisateur, en remplissant et soumettant un formulaire. Les données sont alors envoyées au système de Spot pour vérification, où une recherche est effectuée dans la base de données.

Si la devise existe déjà (comme indiqué par la séquence "devise existe"), le système envoie un message d'erreur à l'interface utilisateur, informant l'utilisateur que la devise ne peut être ajoutée car elle est déjà présente.

Dans le cas contraire, pour une "devise inexistante", les données sont envoyées à la base de données pour insertion. Une fois l'insertion réussie, le système de Spot renvoie une confirmation à l'interface utilisateur, qui à son tour informe l'utilisateur de la réussite de l'opération par un message de confirmation.

**Diagramme 6 : Diagramme de séquence ( Option corriger une valeur )**

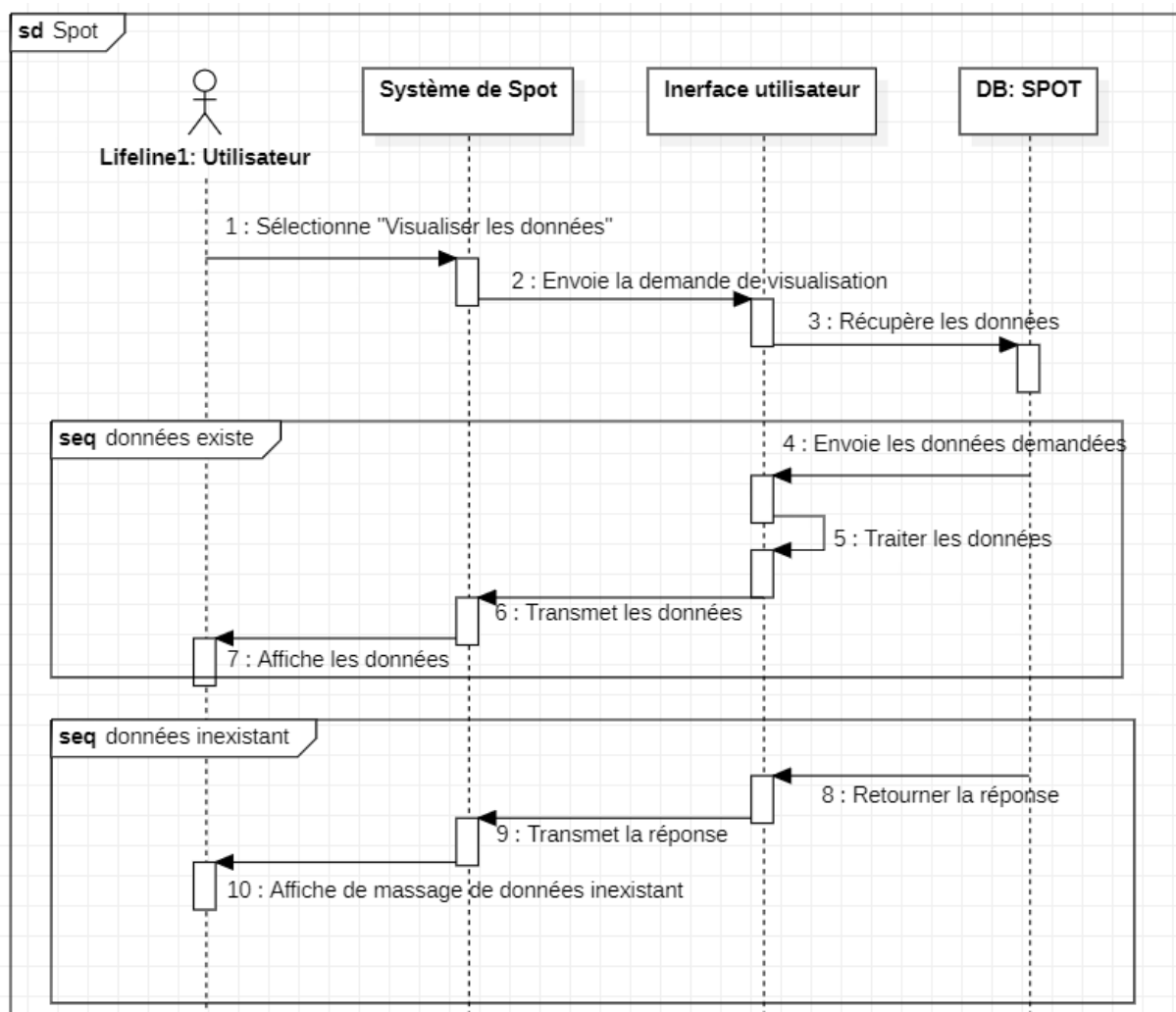


L'utilisateur fait une demande au système de Spot pour obtenir la devise à corriger. Le système de Spot envoie ensuite cette demande à l'interface utilisateur, qui à son tour interagit avec la base de

données pour récupérer les données nécessaires. Les données sont retournées au système de Spot, qui les traite et les envoie à l'interface utilisateur pour affichage.

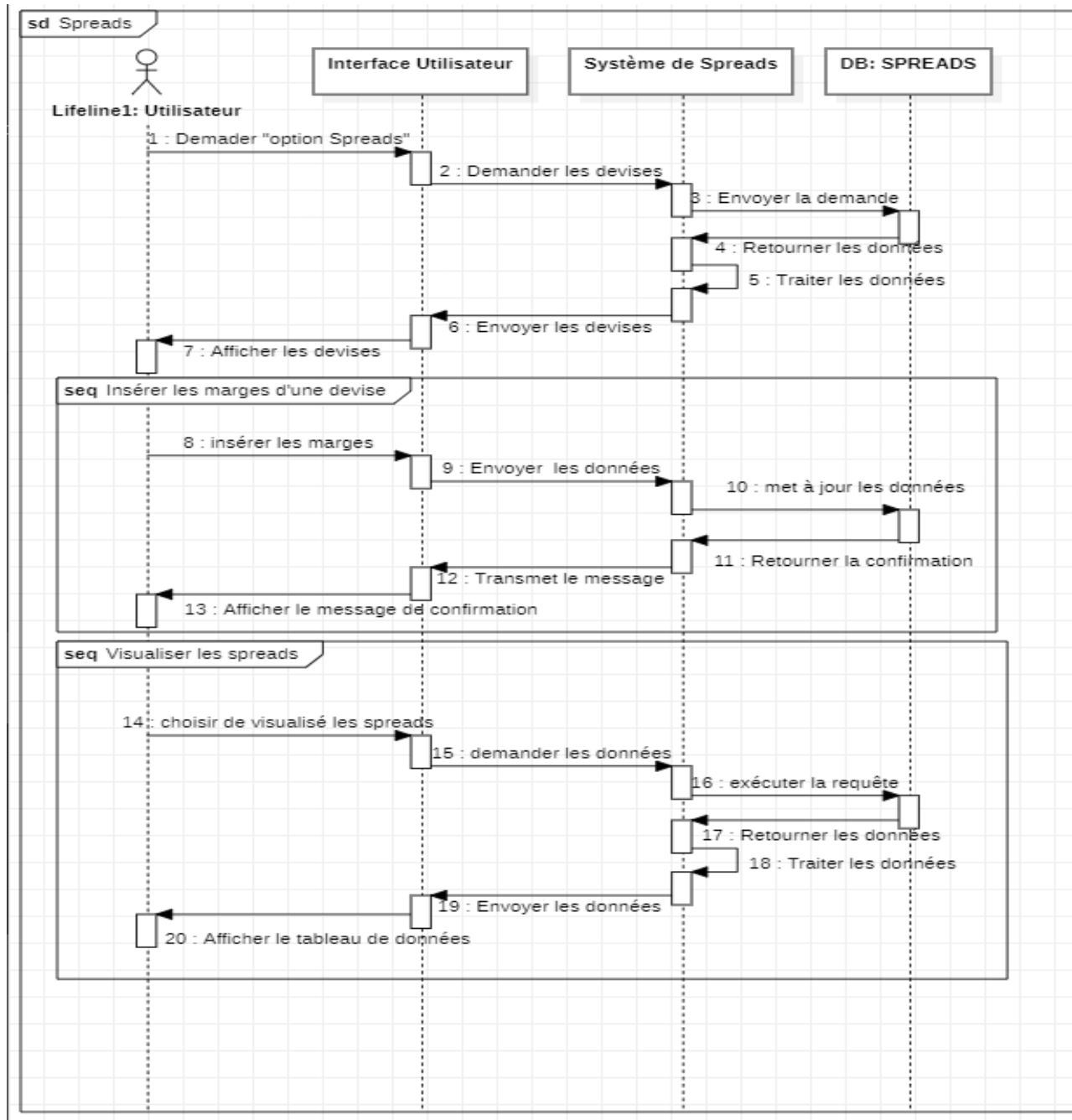
Après avoir consulté les devises, l'utilisateur confirme la correction d'une valeur spécifique, ce qui déclenche l'envoi de la valeur corrigée au système de Spot. Le système procède ensuite à l'envoi de cette modification pour mise à jour dans la base de données. Une fois l'enregistrement mis à jour, le système de Spot envoie une confirmation de la mise à jour à l'interface utilisateur, qui affiche finalement un message de confirmation à l'utilisateur.

**Diagramme 7 :** Diagramme de séquence (Option Visualiser les données)



## II.6. Spreads

Diagramme 8 : Diagramme de séquence (Spreads)



Ce diagramme de séquence illustre le flux d'interactions entre un utilisateur et un système de gestion des spreads pour réaliser trois opérations distinctes : afficher les devises, insérer les marges d'une devise, et visualiser les spreads. Le processus implique des communications entre

l'Utilisateur, l'Interface Utilisateur, le Système de Spreads et la base de données des spreads (DB: SPREADS).

#### **II.6.1. Acteurs du système :**

- **Utilisateur (Lifeline1)** : Personne réalisant des opérations sur les spreads.
- **Interface Utilisateur** : Le point de contact pour l'entrée et la sortie des données pour l'Utilisateur.
- **Système de Spreads** : Le backend qui traite les demandes de l'Interface Utilisateur et interagit avec la base de données.
- **DB : SPREADS** : La base de données contenant les données relatives aux spreads.

#### **II.6.2. Processus Décrit par le Diagramme :**

Afficher les Devises

1. **Demande 'Option Spreads'**: L'Utilisateur initie une demande pour voir les options de spreads via l'Interface Utilisateur.
2. **Demander les Devises** : L'Interface Utilisateur transmet la demande au Système de Spreads.
3. **Envoyer la Demande**: Le Système de Spreads interroge la DB: SPREADS pour les devises.
4. **Retourner les Données**: Les informations sont récupérées et renvoyées par la DB: SPREADS.
5. **Traiter les Données**: Les données sont traitées par le Système de Spreads.
6. **Envoyer les Devises**: Les devises traitées sont envoyées à l'Interface Utilisateur.
7. **Afficher les Devises**: L'Interface Utilisateur présente les devises à l'Utilisateur.

#### **Insérer les Marges d'une Devise :**

8. **Insérer les Marges**: L'Utilisateur entre les marges pour une devise spécifique.

9. **Envoyer les Données** : Ces nouvelles données de marge sont envoyées au Système de Spreads.
10. **Mettre à jour les Données** : Le Système de Spreads met à jour les informations dans la DB : SPREADS.
11. **Retourner la Confirmation** : Une confirmation de mise à jour est envoyée au Système de Spreads.
12. **Transmet le Message** : Le Système de Spreads transmet le message de confirmation à l'Interface Utilisateur.
13. **Afficher le Message de Confirmation** : Un message de confirmation est affiché à l'Utilisateur par l'Interface Utilisateur.

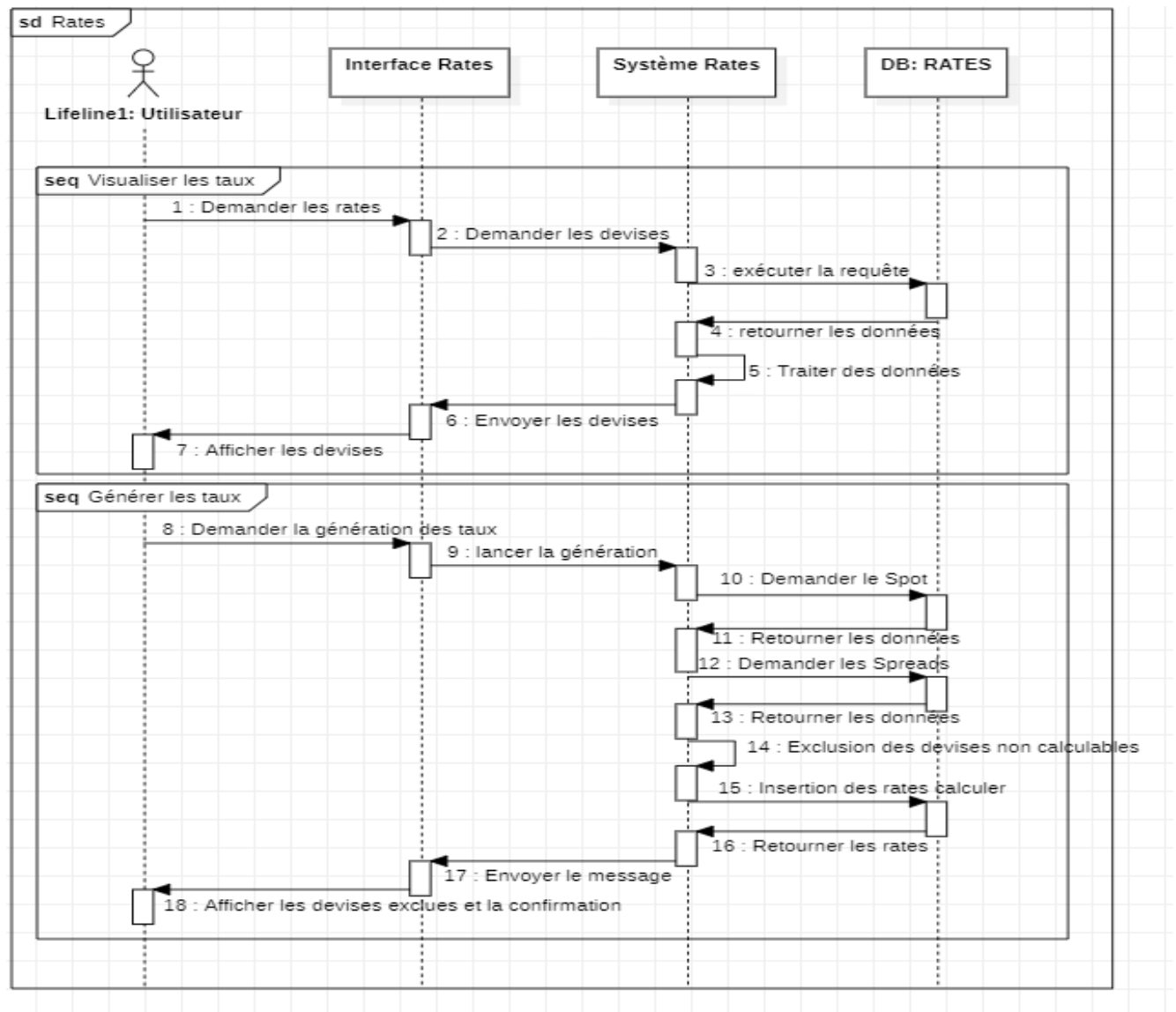
#### **Visualiser les Spreads:**

14. **Choisir de Visualiser les Spreads** : L'Utilisateur sélectionne l'option pour visualiser les spreads.
15. **Demander les Données** : L'Interface Utilisateur demande les données de spreads au Système de Spreads.
16. **Exécuter la Requête** : Le Système de Spreads exécute la requête dans la DB : SPREADS.
17. **Retourner les Données** : Les données des spreads sont renvoyées par la DB : SPREADS.
18. **Traiter les Données** : Le Système de Spreads traite les données des spreads.
19. **Envoyer les Données** : Les données traitées sont envoyées à l'Interface Utilisateur.
20. **Afficher le Tableau de Données** : L'Interface Utilisateur affiche les spreads à l'Utilisateur sous forme de tableau.



## II.7. Rates

Diagramme 9 : Diagramme de séquence (Rates)



### II.7.1. Acteurs impliqués :

- **Utilisateur** : La personne qui initie la requête de visualisation ou de génération des taux.
- **Interface Rates** : Le portail par lequel l'utilisateur soumet des requêtes.
- **Système Rates** : Le backend qui traite les requêtes et communique avec la base de données.

- **DB: RATES :** La base de données où les informations sur les taux de change sont stockées et mises à jour.

## **II.7.2. Fonctionnalités clés :**

### **1. Visualiser les Taux :**

- L'utilisateur initie une demande pour visualiser les taux de change actuels.
- L'Interface Rates transmet cette demande au Système Rates, qui exécute une requête dans la DB : RATES.
- Les données sont renvoyées à l'interface et affichées à l'utilisateur.

### **2. Générer les Taux :**

- Pour générer de nouveaux taux, l'utilisateur soumet une demande via l'Interface Rates.
- Le Système Rates lance la génération en sollicitant des informations de marché supplémentaires, telles que les prix spot et les spreads.
- Après le calcul, le système met à jour les enregistrements dans la DB: RATES et confirme l'opération à l'interface.
- L'utilisateur reçoit une confirmation que les nouveaux taux ont été générés et visualise les taux exclus et confirmés par l'interface.

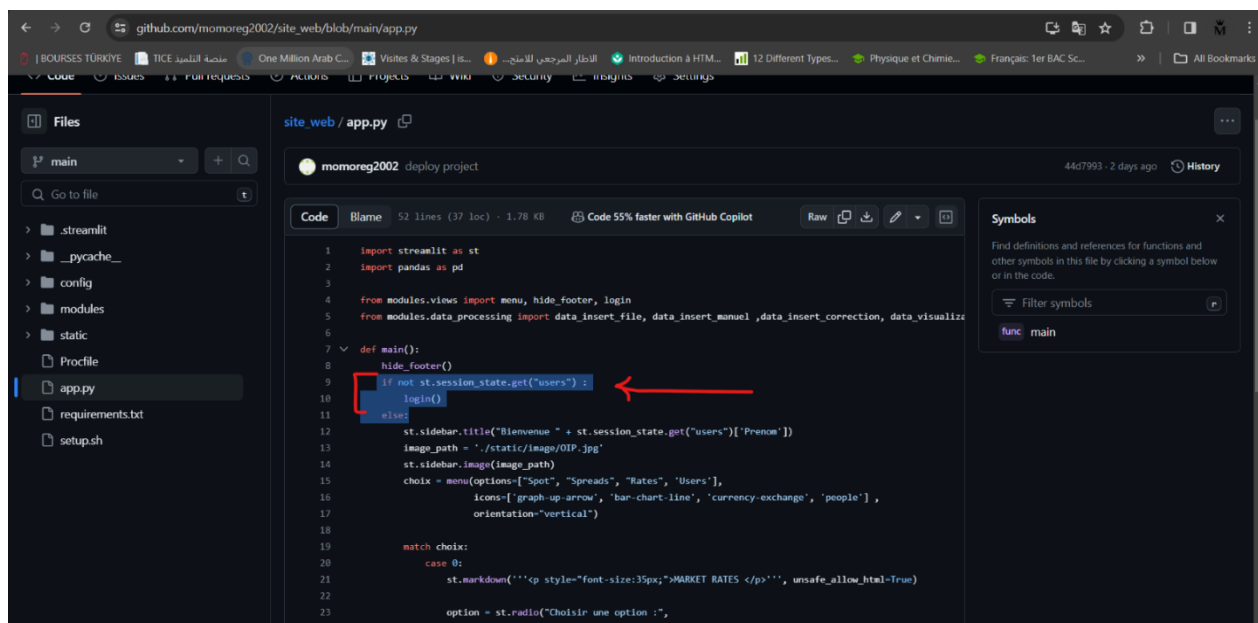
## CHAPITE 3 : UI/UX de l'Application Streamlit

### III. UI/UX de l'Application Streamlit

Cette partie détaille la conception UI/UX de l'application Streamlit, en mettant l'accent sur la navigation, l'interactivité, et l'esthétique visuelle. Le code source, notamment **app.py**, utilise des pratiques avancées pour offrir une expérience utilisateur optimale, intégrant des composants Streamlit pour une interface utilisateur intuitive et réactive.

#### III.1. Structure de l'application

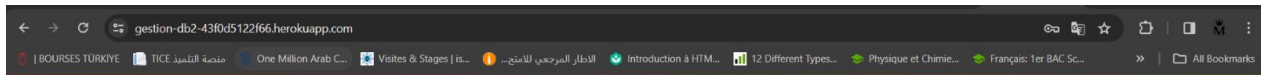
L'application s'articule autour d'un modèle de navigation basé sur l'état de session pour distinguer les utilisateurs non authentifiés des utilisateurs authentifiés, avec une barre latérale personnalisée servant de point central pour la navigation.



```
1 import streamlit as st
2 import pandas as pd
3
4 from modules.views import menu, hide_footer, login
5 from modules.data_processing import data_insert_file, data_insert_manuel, data_insert_correction, data_visualization
6
7 def main():
8     hide_footer()
9     if not st.session_state.get("users"):
10         login()
11     else:
12         st.sidebar.title("Bienvenue " + st.session_state.get("users")["Prenom"])
13         image_path = './static/image/OIP.jpg'
14         st.sidebar.image(image_path)
15         choix = menu(options=["Spot", "Spreads", "Rates", "Users"],
16                     icons=["graph-up-arrow", "bar-chart-line", "currency-exchange", "people"],
17                     orientation="vertical")
18
19         match choix:
20             case 0:
21                 st.markdown('`cp` style="font-size:35px;">MARKET RATES </p`', unsafe_allow_html=True)
22
23         option = st.radio("Choisir une option :",
24                          ("Insérer le fichier", "Insérer une donnée manuellement", "Filtrer une valeur"))
```

##### III.1.1. Authentification et Accueil

- **Authentification** : Les utilisateurs débutent sur une page d'authentification. Le système d'authentification est essentiel pour personnaliser l'expérience utilisateur.



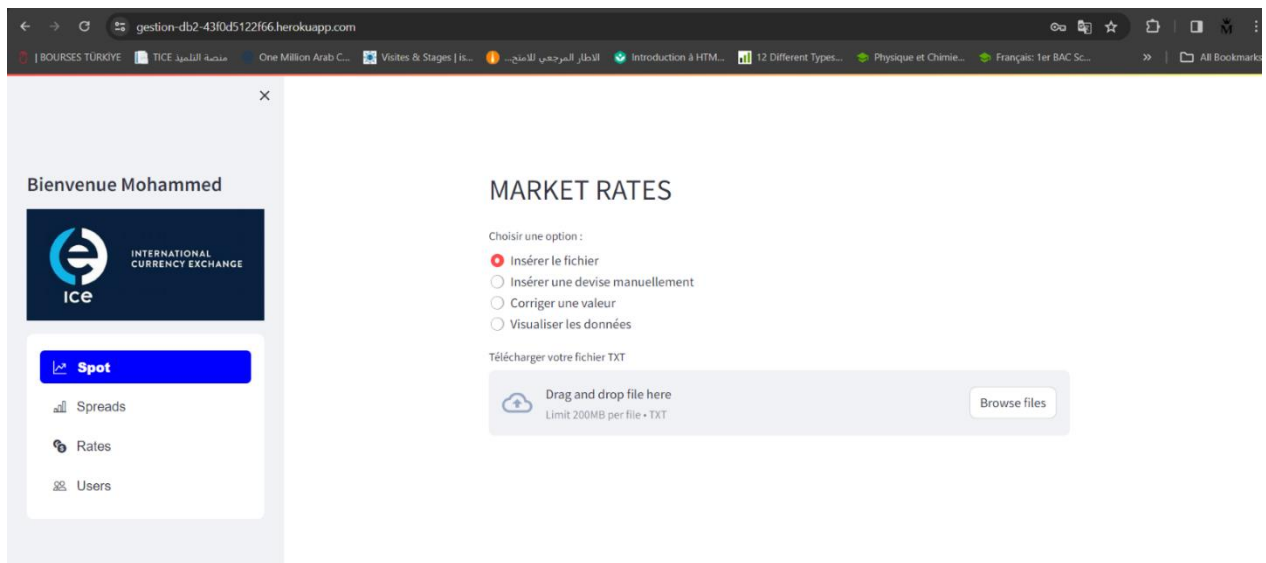
Email

0/50

Password

Login

- **Accueil** : Après connexion, un message de bienvenue personnalisé s'affiche, renforçant l'aspect personnalisé de l'UI.



### III.1.2. Navigation

- **Barre latérale** : La barre latérale agit comme le principal mécanisme de navigation, permettant aux utilisateurs de choisir entre différents modules tels que la gestion des taux, des spreads, et des utilisateurs. Cela simplifie l'architecture de navigation et améliore l'expérience utilisateur en rendant l'application facile à naviguer.

III.1.3.Interaction et visualisation des données

- **Sélection des fonctionnalités** : Les utilisateurs peuvent choisir parmi diverses options telles que l'insertion de données, la visualisation, et la gestion des utilisateurs. Cela implique une interaction directe, permettant une gestion efficace des données.
- **Visualisation des données** : La capacité à visualiser les données directement dans l'application renforce l'UX en fournissant un feedback instantané et pertinent.

- ☐ Insérer le fichier
- ☐ Insérer une devise manuellement
- ☐ Corriger une valeur
- ☒ Visualiser les données

Spot :

	Currency_Code	Rate	UpdateDate
0	DOP	0.023	2024-01-03
1	JPY	0.0093	2024-01-03
2	CHF	1.5646	2024-01-03
3	THB	0.0387	2024-01-03
4	BRL	0.2716	2024-01-03
5	GBP	1.6871	2024-01-03
6	USD	1.3359	2024-01-03
7	MXN	0.0783	2024-01-03
8	EUR	1.4585	2024-01-03
9	CRC	0.0026	2024-01-03

III.1.4.Validation de toutes les fonctionnalités :


III.1.4.1. Validation de Spot.

1<sup>er</sup> option (insérer le fichier) :

## Structure de fichier

Fichier	Modifier	Affichage
AED	AED	0.363755 01/03/2024 09:45
ALL	ALL	0.014097 01/03/2024 09:45
ANG	ANG	0.745477 01/03/2024 09:45
AOA	AOA	0.001587 01/03/2024 09:45
ARS	ARS	0.001647 01/03/2024 09:45
AUD	AUD	0.896767 01/03/2024 09:45
AWG	AWG	0.746307 01/03/2024 09:45
AZN	AZN	0.785410 01/03/2024 09:45
BAM	BAM	0.745727 01/03/2024 09:45
BBD	BBD	0.667945 01/03/2024 09:45
BDT	BDT	0.012184 01/03/2024 09:45
BGN	BGN	0.745727 01/03/2024 09:45
BHD	BHD	3.552898 01/03/2024 09:45
BIF	BIF	0.000469 01/03/2024 09:45
BMD	BMD	1.335890 01/03/2024 09:45
BND	BND	1.005404 01/03/2024 09:45
BOB	BOB	0.193394 01/03/2024 09:45
BRL	BRL	0.271604 01/03/2024 09:45
BSD	BSD	1.335890 01/03/2024 09:45
BWP	BWP	0.099345 01/03/2024 09:45
BZD	BZD	0.662894 01/03/2024 09:45
CAD	CAD	1.000000 01/03/2024 09:45
CHF	CHF	1.564560 01/03/2024 09:45
CLP	CLP	0.001507 01/03/2024 09:45
CNY	CNY	0.187446 01/03/2024 09:45
COP	COP	0.000341 01/03/2024 09:45
CRC	CRC	0.002561 01/03/2024 09:45
CVE	CVE	0.013227 01/03/2024 09:45
CZK	CZK	0.059082 01/03/2024 09:45
DJF	DJF	0.007511 01/03/2024 09:45
DKK	DKK	0.195565 01/03/2024 09:45
DOP	DOP	0.023021 01/03/2024 09:45
DZD	DZD	0.009918 01/03/2024 09:45
EGP	EGP	0.043234 01/03/2024 09:45
EUR	EUR	1.458516 01/03/2024 09:45
FJD	FJD	0.599402 01/03/2024 09:45
GBP	GBP	1.687107 01/03/2024 09:45
GEL	GEL	0.497542 01/03/2024 09:45
GHS	GHS	0.111795 01/03/2024 09:45
GIP	GIP	1.687107 01/03/2024 09:45

Bienvenue Wassim



Spot

Spreads

Rates

Users

## MARKET RATES

Choisir une option :

- ☒ Insérer le fichier
- ☐ Insérer une devise manuellement
- ☐ Corriger une valeur
- ☐ Visualiser les données

Télécharger votre fichier TXT

Drag and drop file here

Limit 200MB per file • TXT

Browse files

CADCCRATES.TXT

3.7KB

X


document inséré.

	Currency_Code	Rate	UpdateDate
0	BRL	0.2716	2024-01-03
1	CHF	1.5646	2024-01-03
2	COP	0.0003	2024-01-03
3	CRC	0.0026	2024-01-03
4	DOP	0.023	2024-01-03
5	EUR	1.4585	2024-01-03
6	GBP	1.6871	2024-01-03
7	JPY	0.0093	2024-01-03

2ème option (Insérer une devise manuellement):

Teste avec une devise qui existe déjà :

Bienvenue Wassim



Spot

Spreads

Rates

Users

## MARKET RATES

Choisir une option :

- ☐ Insérer le fichier
- ☒ Insérer une devise manuellement
- ☐ Corriger une valeur
- ☐ Visualiser les données

Le code de devise doit être composé des deux premières lettres qui réfèrent au nom du pays, suivies d'une lettre représentant la devise.

Entrez le code de devise (Exemple : USD pour le dollar américain)

jpy

Entrez le taux de change

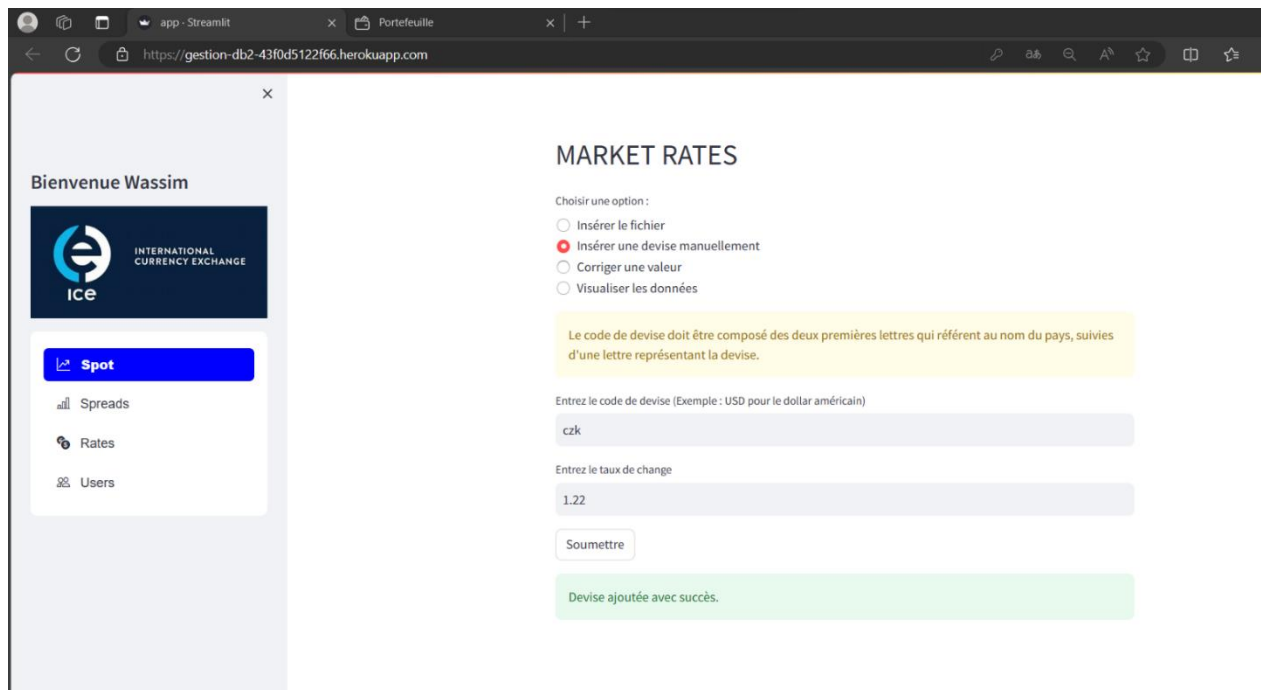
Soumettre

Cette devise existe déjà dans la base de données.

Voici les détails :

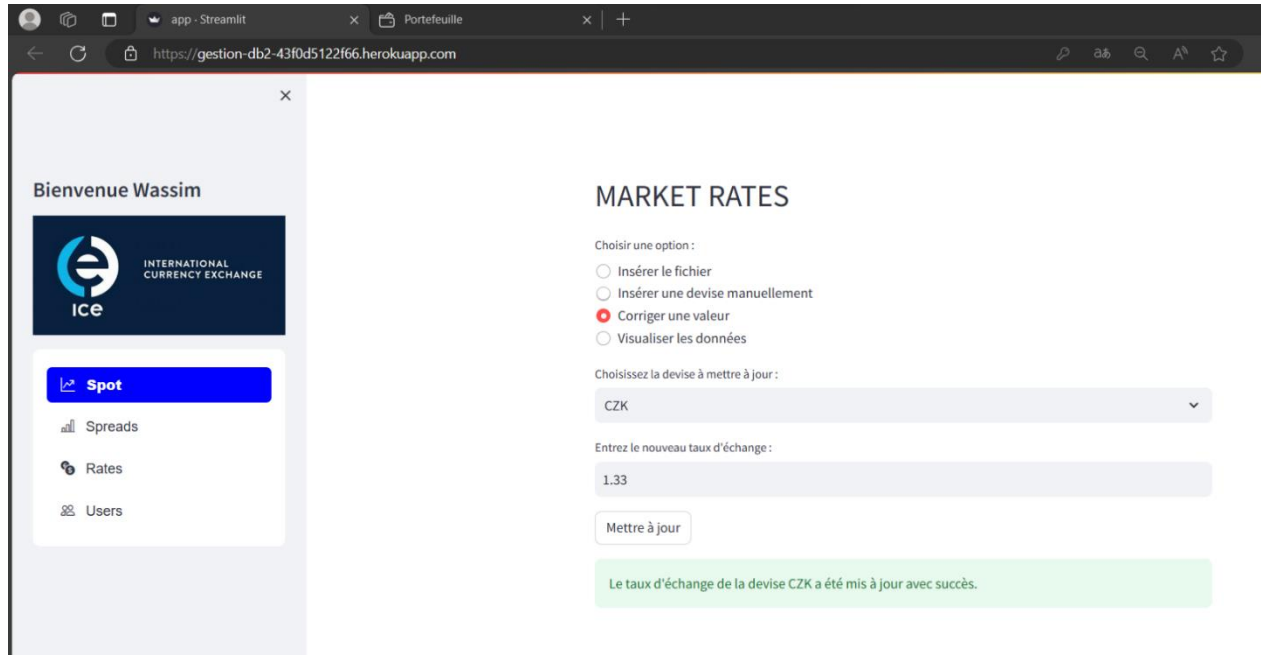
	Currency_Code	Rate	UpdateDate
0	JPY	0.0093	2024-01-03

### Insérer une devise qui n'existe pas :



The screenshot shows a web application interface for 'ICE INTERNATIONAL CURRENCY EXCHANGE'. On the left, a sidebar contains a welcome message 'Bienvenue Wassim', the ICE logo, and a menu with 'Spot', 'Spreads', 'Rates', and 'Users'. The main area is titled 'MARKET RATES' and contains a form. Under the heading 'Choisir une option :', the radio button for 'Insérer une devise manuellement' is selected. A yellow warning box states: 'Le code de devise doit être composé des deux premières lettres qui réfèrent au nom du pays, suivies d'une lettre représentant la devise.' Below this, the 'Entrez le code de devise (Exemple : USD pour le dollar américain)' field contains 'czk'. The 'Entrez le taux de change' field contains '1.22'. A 'Soumettre' button is present. A green success message at the bottom reads: 'Devise ajoutée avec succès.'

### 3ème option (Corriger une valeur) :



The screenshot shows the same web application interface. In the 'MARKET RATES' section, the radio button for 'Corriger une valeur' is now selected. The 'Choisissez la devise à mettre à jour :' dropdown menu is set to 'CZK'. The 'Entrez le nouveau taux d'échange :' field contains '1.33'. A 'Mettre à jour' button is visible. A green success message at the bottom reads: 'Le taux d'échange de la devise CZK a été mis à jour avec succès.'



Confirmation de changement est validation de la 4<sup>ème</sup> option :

### III.1.4.2. Validation de Spreads.

#### 1) Insérer les marges d'une devise :

The screenshot shows a web browser window with the URL `https://gestion-db2-43f0d5122f66.herokuapp.com`. The application is titled "app - Streamlit" and has a sidebar with the "Ice" logo and navigation links for "Spot", "Spreads", "Rates", and "Users". The main content area is titled "Insérer les marges d'une devise". It contains a dropdown menu for "Choisissez la devise à mettre à jour:" with "CZK" selected. Below this are two input fields for "Entrez le pourcentage de Buy:" and "Entrez le pourcentage de Sell:", both containing the value "12". A "Soumettre" button is present. A green success message states: "Les marges d'échange de la devise CZK ont été mises à jour avec succès." Below this is a "Visualiser les marges" button with a bar chart icon.

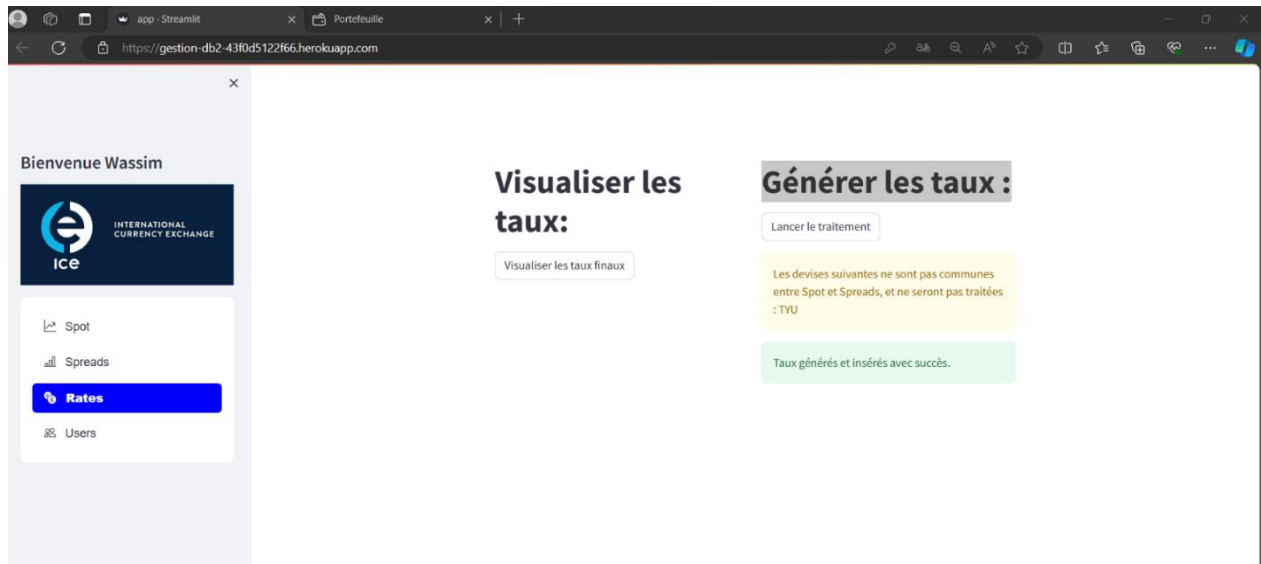
#### 2 Visualiser les marges :

The screenshot shows the same web browser window, but the "Visualiser les marges" button has been clicked. The form fields are now disabled. Below the "Soumettre" button, a table displays the exchange rates for various currencies. The table has four columns: "Currency\_Code", "Buy\_P", and "Sell\_P". The data is as follows:

	Currency_Code	Buy_P	Sell_P
5	CHF	12	12
6	CRC	20	25
7	COP	20	25
8	EUR	14.5	14
9	GBP	14.5	14
10	MXN	16	20
11	JPY	15	15
12	MAD	25	35
13	TYU	11	11
14	CZK	12	12

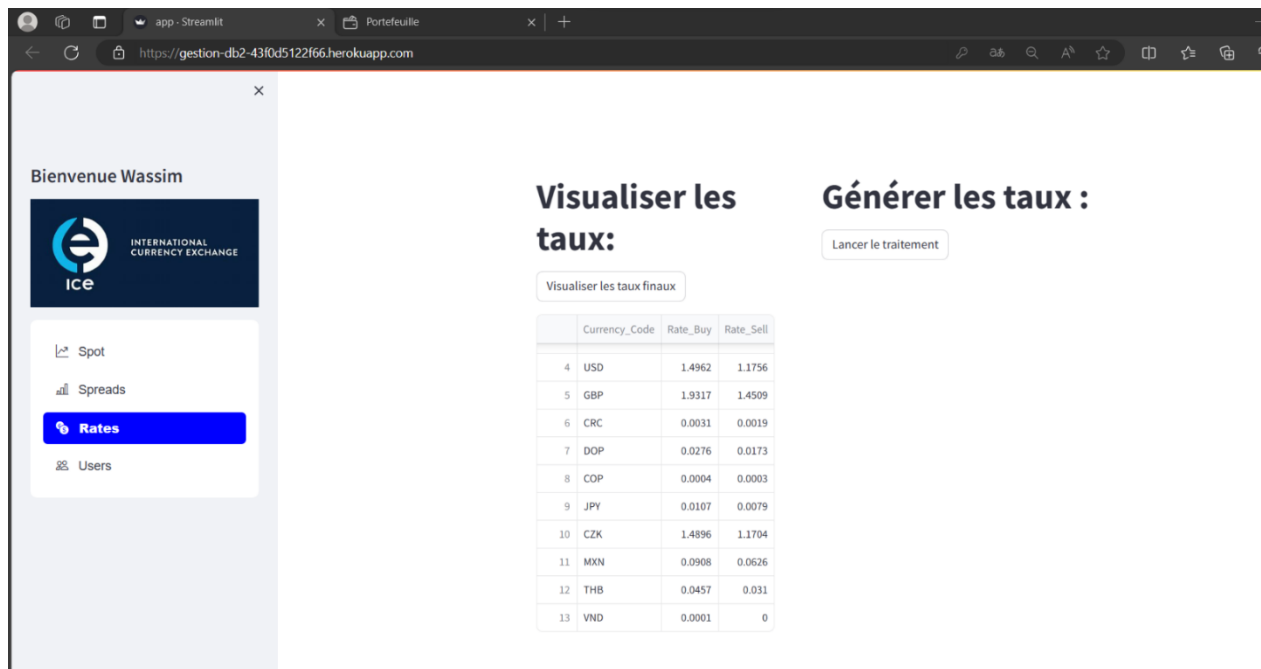
### III.1.4.3. Validation de Rates

#### 1) Générer les taux :



- Ici on remarque que la devise TYU n'existe pas dans les deux table donc il sera exclue de traitement.


#### 2) Visualiser les taux :



III.1.4.4. Validation de Users.

1) Ajouter un nouvel utilisateur :

Bienvenue Wassim



Spot

Spreads

Rates

**Users**

0	Wassim	Boualamallah	Wassim.Boualamallah@icecurrency.com
1	Mohammed	Regragui	Mohammed.Regragui@icecurrency.com
2	Safouen	Bani	Safouen.Bani@icecurrency.com
3	Mehamed	Regragui	mohamedregragui@gmail.com

Ajouter un nouvel utilisateur

Prénom

test

4/50

Nom

test

4/50

Email

test.test@icecurrency.com

25/50

Mot de passe


\*\*\*\*

👁

Ajouter

Utilisateur test test ajouté avec succès.

Bienvenue Wassim



Spot

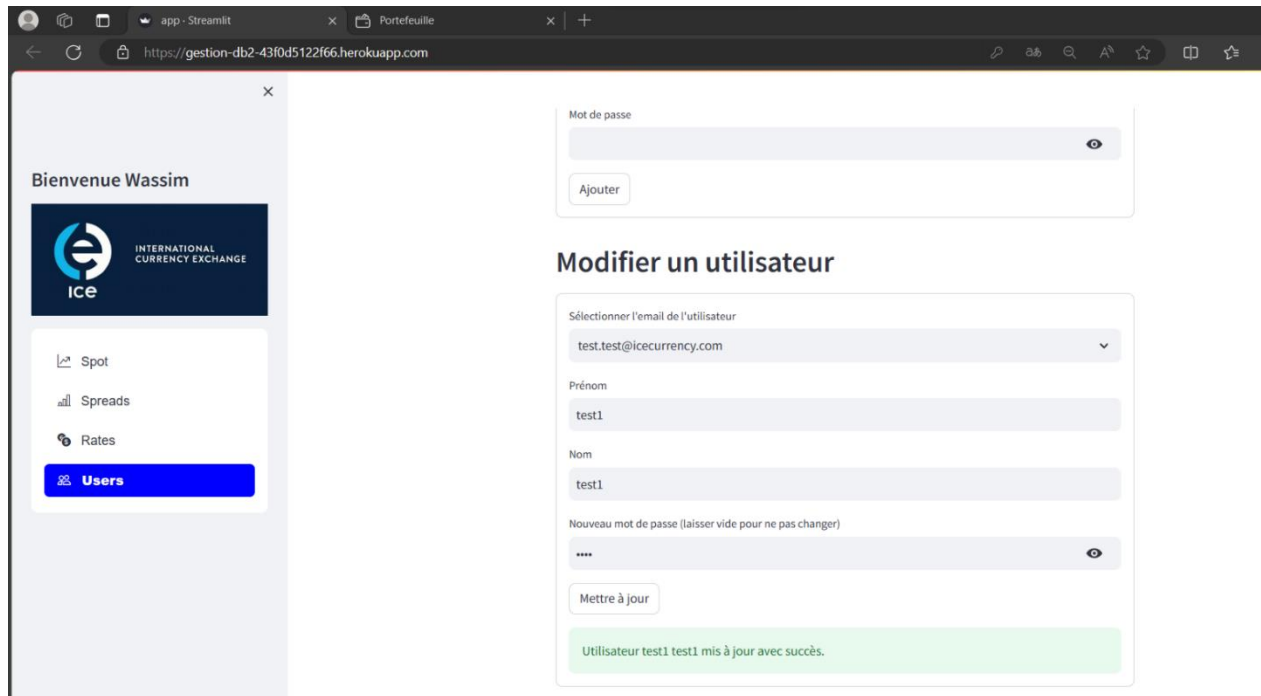
Spreads

Rates

Gestion des utilisateurs

	Prenom	Nom	Email
0	Wassim	Boualamallah	Wassim.Boualamallah@icecurrency.com
1	Mohammed	Regragui	Mohammed.Regragui@icecurrency.com
2	Safouen	Bani	Safouen.Bani@icecurrency.com
3	Mehamed	Regragui	mohamedregragui@gmail.com
4	test	test	test.test@icecurrency.com

## 2) Modifier un utilisateur :



Bienvenue Wassim

INTERNATIONAL CURRENCY EXCHANGE  
ICE

Spot  
Spreads  
Rates  
**Users**

Mot de passe

Ajouter

### Modifier un utilisateur

Sélectionner l'email de l'utilisateur  
test.test@icecurrency.com

Prénom  
test1

Nom  
test1

Nouveau mot de passe (laisser vide pour ne pas changer)  
\*\*\*\*

Mettre à jour

Utilisateur test1 test1 mis à jour avec succès.

# Gestion des utilisateurs

	Prenom	Nom	Email
0	Wassim	Boualamallah	Wassim.Boualamallah@icecurrency.com
1	Mohammed	Regragui	Mohammed.Regragui@icecurrency.com
2	Safouen	Bani	Safouen.Bani@icecurrency.com
3	Mehamed	Regragui	mohamedregrgui@gmail.com
4	test1	test1	test.test@icecurrency.com

### 3) Supprimer un utilisateur :

## Supprimer un utilisateur

Sélectionner l'email de l'utilisateur à supprimer

test.test@icecurrency.com



Supprimer

Utilisateur [test.test@icecurrency.com](#) supprimé avec succès.

## Gestion des utilisateurs

	Prenom	Nom	Email
0	Wassim	Boualamallah	Wassim.Boualamallah@icecurrency.com
1	Mohammed	Regragui	Mohammed.Regragui@icecurrency.com
2	Safouen	Bani	Safouen.Bani@icecurrency.com
3	Mehamed	Regragui	mohamedregrgui@gmail.com

### III.1.5.Éléments d'Interface (UI)

- **Composants Streamlit** : L'application tire parti des composants Streamlit pour créer une interface riche et interactive, incluant des boutons radio, des images, et des titres personnalisés.
- **Personnalisation** : Les éléments visuels comme les images dans la barre latérale contribuent à une expérience utilisateur plus engageante et personnalisée.
- **Responsive Design** : Bien que non explicitement mentionné dans le segment analysé, l'utilisation de Streamlit suggère une interface adaptative aux différents dispositifs d'affichage.

### **III.1.6. Accessibilité**

- **Simplicité et Clarté** : L'application maintient une interface épurée et intuitive, essentielle pour l'accessibilité. La documentation n'a pas révélé de fonctionnalités spécifiques d'accessibilité, mais la clarté de l'UI favorise une expérience utilisateur inclusive.

### **III.2. Conclusion**

L'application Streamlit a été conçue avec un soin particulier pour l'UI/UX, en utilisant des pratiques de développement avancées pour assurer une expérience utilisateur fluide et personnalisée. Les choix de conception, de la navigation personnalisée à la visualisation des données, reflètent un engagement envers une interface utilisateur intuitive et une expérience utilisateur enrichissante.

## CHAPITRE 4 : Conception de la base de données

Dans le cadre de la gestion des opérations de change, il est crucial de disposer d'une base de données bien conçue qui permet de stocker et de calculer efficacement les taux de change du marché ainsi que les écarts (spreads) d'achat et de vente. Cette base de données doit être en mesure de refléter les fluctuations des taux de change en temps réel et d'ajuster les taux d'achat et de vente en conséquence.

### 4.1. Structure de la Base de Données

La base de données est conçue autour de trois tables principales : **Spot**, **Spreads**, et **Rates**, chacune ayant un rôle spécifique dans le calcul et le stockage des taux de change et des écarts.

#### 4.1.1. Table Spot

- **Currency\_Code** : Identifie la devise concernée. C'est une clé primaire qui garantit l'unicité de chaque enregistrement.
- **Rate** : Représente le taux de change actuel de la devise sur le marché.
- **UpdateDate** : Indique la date et l'heure de la dernière mise à jour du taux de change, permettant de suivre l'évolution des taux en temps réel.

#### 4.1.2. Table Spreads

- **Currency\_Code** : Code de la devise, qui sert de clé étrangère pour référencer la table **Spot**.
- **Buy\_P** : Pourcentage de l'écart appliqué au taux de change du marché pour calculer le taux d'achat.
- **Sell\_P** : Pourcentage de l'écart appliqué au taux de change du marché pour calculer le taux de vente.

#### 4.1.3. Table Rates



- **Currency\_Code** : Code de la devise, faisant référence à la table **Spot** et assurant la cohérence des données.
- **Buy** : Taux d'achat calculé, basé sur le taux de change du marché et l'écart d'achat.
- **Sell** : Taux de vente calculé, de même, dérivé du taux de change du marché et de l'écart de vente.

#### 4.2. Relations entre les Tables

La relation entre les tables **Spot** et **Spreads** est essentielle pour le calcul des taux dans la table **Rates**. Chaque enregistrement dans **Rates** dépend des données présentes dans **Spot** et **Spreads**, à travers le **Currency\_Code** qui agit comme une clé étrangère, garantissant l'intégrité référentielle et la précision des calculs.

#### 4.3. Considérations Techniques

- **Calcul des Taux** : Les taux d'achat et de vente dans la table **Rates** sont calculés en appliquant les écarts (spreads) spécifiés dans la table **Spreads** aux taux de change du marché de la table **Spot**.
- **Intégrité des Données** : L'utilisation des clés étrangères assure que les taux calculés correspondent toujours aux devises valides et à jour.
- **Optimisation des Performances** : Des index peuvent être ajoutés sur les colonnes fréquemment accédées ou recherchées pour améliorer les performances des requêtes mais, Mongo Atlas à l'avantage de l'indexation automatique (`_id`).

#### 4.4. Hébergement sur MongoDB Atlas

L'ensemble de la base de données ICE est hébergé sur MongoDB Atlas, bénéficiant ainsi d'une infrastructure de cloud sécurisée, performante et facilement scalable. Cette plateforme DBaaS (Database as a Service) est idéalement adaptée aux exigences du projet, offrant une gestion sans souci des données volumineuses et une disponibilité globale pour les opérations de change.

## **4.5. Conclusion**

La conception et le déploiement de ICE sur MongoDB Atlas illustrent l'application réussie de principes de conception de base de données avancés pour répondre aux besoins complexes du trading de devises. ICE représente une solution robuste, flexible et évolutive pour la gestion des taux de change et des écarts, mettant en lumière la capacité de la technologie moderne à soutenir les exigences rigoureuses du secteur financier.

## CHAPITRE 5 : Déploiement d'une application Streamlit sur Heroku

Cet aspect, explique comment déployer une application Streamlit sur Heroku. Le processus utilise Git pour le déploiement et nécessite des fichiers spécifiques dans le projet pour sa configuration sur Heroku.

### V.1. URL de notre application :

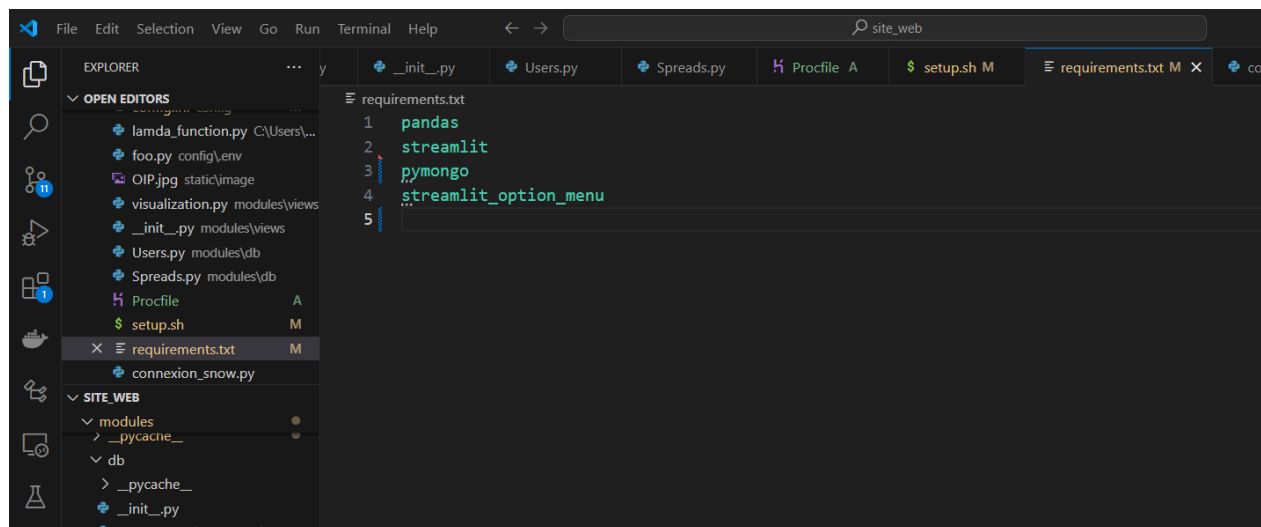
<https://gestion-db2-43f0d5122f66.herokuapp.com/>

### V.2. Prérequis

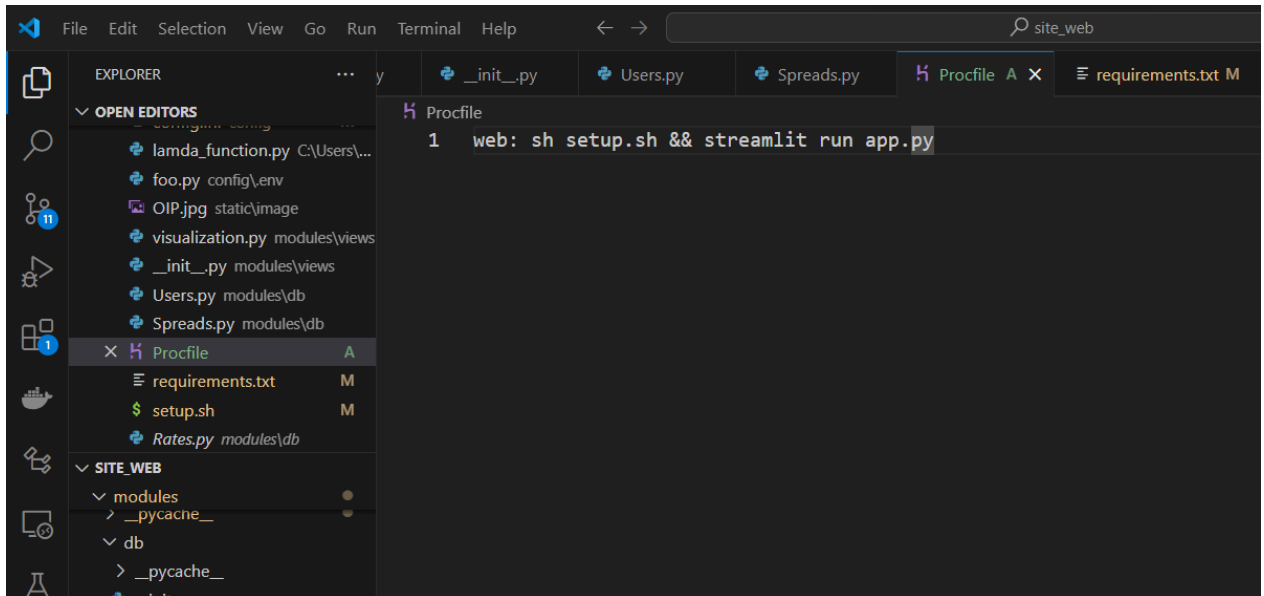
- Compte Heroku
- CLI Heroku installé localement
- Git installé localement

### V.3. Fichiers du projet

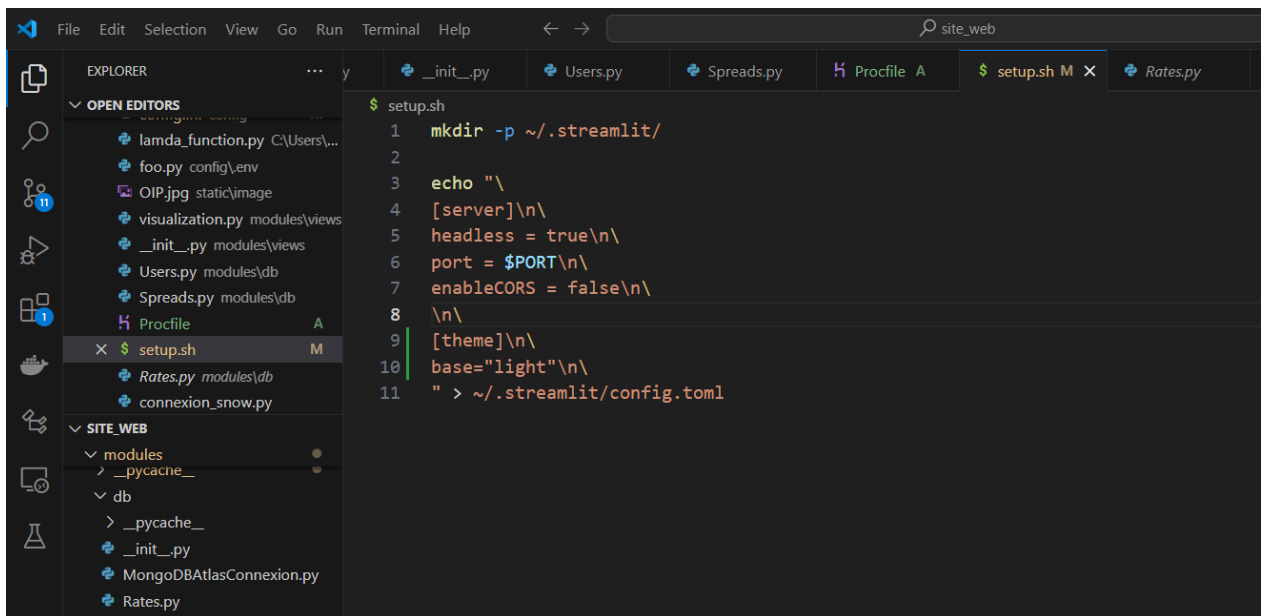
- **app.py** : Le script principal de l'application Streamlit.
- **requirements.txt** : Liste des dépendances Python nécessaires à l'application.



- **Procfile** : Fichier de configuration pour Heroku indiquant comment démarrer l'application.



- **setup.sh** : Script de configuration du serveur pour Streamlit sur Heroku.



## V.4. Étapes de déploiement

### 1. Configuration initiale

Assurez-vous d'avoir installé le CLI Heroku et connectez-vous à votre compte :

La commande => **heroku login**

## 2. Préparation de l'application

Dans le répertoire de notre projet, initialisez un dépôt Git si ce n'est pas déjà fait :

La commande =>

```
git init
```

```
git add .
```

```
git commit -m "Initial commit"
```

## 3. Création de l'application Heroku

Créez une application Heroku depuis la ligne de commande :

La commande =>

```
heroku create applicationGestion
```

## 4. Déploiement sur Heroku

Associez votre dépôt Git à l'application Heroku et déployez :

La commande =>

```
heroku git:remote -a applicationGestion
```

```
git push heroku master
```

## 5. Vérification

Accédez à l'application déployée en utilisant l'URL fournie par Heroku ou en exécutant :

La commande =>

```
heroku open
```

### Surveillance

**N.B :** Pour surveiller l'application et diagnostiquer les problèmes, utilisez les logs Heroku :

```
heroku logs -tail
```

## V.5. Conclusion

Ce guide couvre le déploiement de base d'une application Streamlit sur Heroku. Pour des configurations ou des intégrations spécifiques, référez-vous à la documentation de Heroku ou Streamlit.

## CHAPITRE 6 : Utilisation de REST API avec lambada de AWS

### VI. REST API avec lambada de AWS :

Cette section détaille l'utilisation et le fonctionnement de l'API REST développée dans le cadre de ma synthèse de stage. L'API est conçue pour interagir avec une base de données MongoDB hébergée sur Atlas, permettant de récupérer les taux de change actuels (Rates) sous forme de JSON. Le service est hébergé sur AWS Lambda, offrant une exécution sans serveur et une grande scalabilité.

#### VI.1. Configuration Préliminaire

##### VI.1.1. Environnement MongoDB Atlas

- **Base de données:** ICE
- **Collection:** RATES
- **Authentification :** Configurez `ATLAS_URI` dans les variables d'environnement de votre fonction Lambda pour inclure votre chaîne de connexion à MongoDB Atlas.

##### VI.1.2. AWS Lambda

- **Nom de la fonction :** `fetchRatesFunction`
- **Runtime :** Python 3.8 ou supérieur
- **Variables d'environnement :**
  - `ATLAS_URI` : Votre chaîne de connexion à MongoDB Atlas.

##### VI.1.3. Dépendances

- **pymongo :** Ce module est requis pour interagir avec MongoDB depuis Python. Assurez-vous qu'il est inclus dans votre environnement d'exécution Lambda.

## VI.1.4. API Endpoints

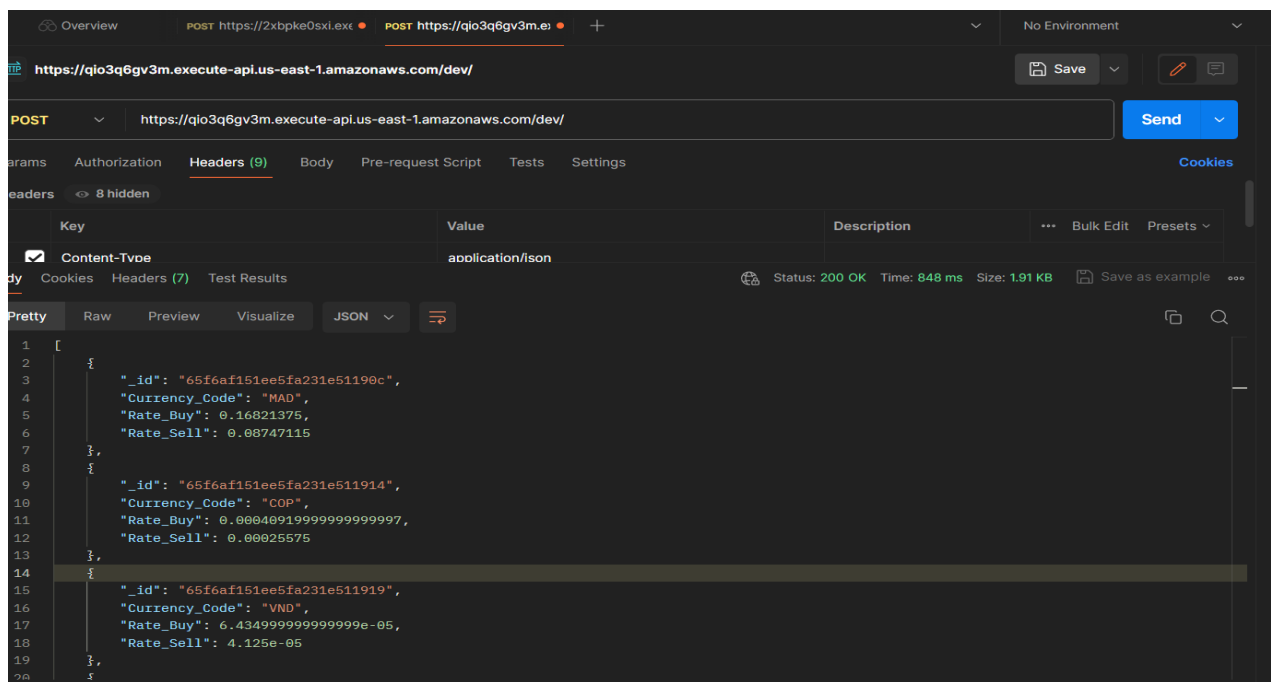
### Récupération des Taux de Change (Rates)

- **URL** : <https://qio3q6gv3m.execute-api.us-east-1.amazonaws.com/dev/>
- **Méthode** : GET
- **Authentification** : Aucune (selon la configuration, cela peut varier)
- **Réponses** :
  - **200 OK** : Succès de la requête. Le corps de la réponse contient les taux de change actuels sous forme de JSON.
  - **500 Internal Server Error** : Un problème est survenu côté serveur. Le corps de la réponse contient le message d'erreur.

### VI.1.5. Format de la Réponse

Le service renvoie les données sous forme d'un tableau JSON contenant les documents de la collection **RATES**. Chaque document inclut les champs stockés dans la base de données MongoDB, convertis en chaîne de caractères.

### Exemple de Réponse



The screenshot shows a REST client interface with the following details:

- Method:** POST (though the URL is a GET endpoint)
- URL:** <https://qio3q6gv3m.execute-api.us-east-1.amazonaws.com/dev/>
- Status:** 200 OK
- Time:** 848 ms
- Size:** 1.91 KB
- Content-Type:** application/json

The response body is a JSON array of three documents:

```
[
  {
    "_id": "65f6af151ee5fa231e51190c",
    "Currency_Code": "MAD",
    "Rate_Buy": 0.16821375,
    "Rate_Sell": 0.08747115
  },
  {
    "_id": "65f6af151ee5fa231e511914",
    "Currency_Code": "COP",
    "Rate_Buy": 0.00040919999999999997,
    "Rate_Sell": 0.00025575
  },
  {
    "_id": "65f6af151ee5fa231e511919",
    "Currency_Code": "VND",
    "Rate_Buy": 6.434999999999999e-05,
    "Rate_Sell": 4.125e-05
  }
]
```

### VI.1.6. Gestion des Erreurs

Les erreurs côté serveur sont signalées par le statut HTTP 500. Le corps de la réponse contiendra un objet JSON avec un champ **error** décrivant l'erreur.

#### Exemple de Réponse d'Erreur

```
{  
  "error": "Erreur de connexion à la base de données."  
}
```

### VI.1.7. Sécurité

La sécurité de l'API doit être prise en considération, notamment l'authentification et la gestion des accès. Selon la configuration de votre API Gateway AWS et de MongoDB Atlas, vous pourriez avoir besoin de mettre en place des clés API, des tokens d'authentification, ou d'autres mécanismes de sécurité.

## VI.2. Conclusion

Cette API REST simplifie l'accès aux taux de change actuels stockés dans MongoDB Atlas, en offrant une interface HTTP légère et facile à utiliser. Parfaitement intégrée à AWS Lambda, elle assure une exécution efficace et sans serveur, adaptée à un large éventail d'applications nécessitant l'accès en temps réel aux données financières.



## **CHAPITRE 7 : Conclusion générale et perspective**

### **VII. Conclusion**

Alors que mon stage touche à sa fin, je prends un moment pour considérer le chemin parcouru, les défis relevés et les connaissances emmagasinées. Au travers de ce stage, j'ai eu l'opportunité de travailler sur des technologies de pointe telles que Mongo Atlas pour la gestion de bases de données, Streamlit pour la création d'interfaces utilisateurs interactives, et AWS Lambda avec API Gateway pour développer des architectures serveurless évolutives.

#### **VII.1. Affrontements et Surmontements des Problèmes**

J'ai été confronté à plusieurs défis techniques tout au long de mon stage. L'intégration des services de Mongo Atlas a exigé une compréhension approfondie des principes NoSQL, qui contrastaient avec ma familiarité antérieure avec les bases de données relationnelles. L'apprentissage de Streamlit a également présenté une courbe d'apprentissage abrupte, m'obligeant à repenser la manière dont les interfaces utilisateur peuvent être rapidement prototypées et déployées pour un feedback immédiat. En utilisant AWS Lambda et API Gateway, j'ai dû naviguer à travers la configuration parfois complexe de la sécurité et des permissions, ce qui a renforcé ma capacité à diagnostiquer et résoudre les problèmes de connectivité entre les services cloud.

Ces obstacles, bien que décourageants à l'occasion, ont été des vecteurs d'apprentissage précieux. Chaque problème résolu m'a rapproché d'une compréhension plus nuancée du développement de logiciels dans un environnement cloud moderne. Les solutions que j'ai élaborées en réponse à ces problématiques ne se sont pas seulement traduites par des avancées techniques, mais aussi par une croissance personnelle significative.

#### **VII.2. Contributions et Impacts**

Ma contribution au projet a été diversifiée : non seulement j'ai écrit du code fonctionnel, mais j'ai également conçu une interface utilisateur visant à enrichir l'expérience client, et j'ai développé une base de données à la fois flexible et évolutive. Ces éléments, essentiels à la réussite du projet, ont également eu un impact sur la performance et la satisfaction de l'équipe, ce qui a été une source de grande fierté pour moi.

### **VII.3. Perspectives d'Amélioration**

En regardant en arrière, je vois plusieurs domaines dans lesquels des améliorations pourraient être apportées. Par exemple, la gestion de l'état global dans Streamlit pourrait être optimisée pour des interactions plus fluides, et les fonctions AWS Lambda pourraient être affinées pour réduire encore plus les temps de réponse.

### **VII.4. Visions Futures**

Ce stage m'a non seulement équipé de compétences techniques avancées, mais m'a aussi donné une direction pour ma future carrière. Je suis particulièrement attiré par le potentiel des architectures serveurless et par l'impact positif que des interfaces utilisateur intuitives peuvent avoir sur l'engagement des utilisateurs.