

Avrile Floro (n° étudiant : 22000086)
Aude Hennino (n° étudiant : 19011803)
Maxime Bronny (n° étudiant : 19009314)

Réalisation de programme

Le 2 juin 2024



Table des matières

1	Documentation technique	3
1.1	Introduction	3
1.1.1	Fonctionnalités principales	3
1.2	Architecture du système	3
1.2.1	Les variables globales	5
1.3	Détails des composants de l'interface utilisateur	6
1.3.1	Onglet 1 : Affichage de l'image	6
1.3.2	Onglet 2 : Vérification	11
1.3.3	Onglet 3 : Traduction	16
1.3.4	Onglet 4 : Comptabilité	17
1.3.5	Onglet 5 : Statistiques	19
1.4	Technologies utilisées et limitations	21
1.5	Packaging	22
1.6	Configurations minimales requises	23
1.6.1	Langage de Programmation	23
1.6.2	Bibliothèques et dépendances	23
1.7	Construction du packaging	23
1.8	Procédure d'installation et de configuration	24
1.8.1	Contenu du dossier projet	24
1.8.2	Installations préalables à l'exécution du projet	24
1.8.3	Lancer le programme	24
1.8.4	Consulter le code source	25

Réalisation de programme

Documentation technique

Avrile Floro
Étudiante n°22000086

Aude Hennino
Étudiante n°19011803

Maxime Bronny
Étudiant n°19009314

1 Documentation technique

1.1 Introduction

FactureFacile est une application permettant d'optimiser la gestion des factures. Elle intègre des fonctionnalités d'extraction, de vérification, d'enregistrement et de traduction du contenu des factures. Grâce à la technologie OCR (Optical Character Recognition) fournie par Tesseract, le logiciel est capable d'extraire le texte des images ou des fichiers PDF, facilitant ainsi la validation et la correction des informations par les utilisateurs avant leur enregistrement définitif dans une base de données. En outre, FactureFacile offre des options de traduction du texte en anglais, espagnol ou allemand et peut effectuer la conversion des montants non-euro en euros de manière automatique.

1.1.1 Fonctionnalités principales

Importation et visualisation de factures : Permet aux utilisateurs de charger des factures sous forme d'images ou de fichiers PDF et de les visualiser sur un canvas interactif. Les utilisateurs peuvent dessiner des rectangles pour délimiter les zones de texte spécifiques sur les factures afin de faciliter l'extraction textuelle.

Extraction de texte : Utilise l'OCR pour détecter et extraire le texte des zones spécifiées, avec possibilité d'ajustement manuel des zones d'extraction.

Vérification des données : Offre une interface facilitant la vérification et la correction des données extraites comme l'émetteur, la date, le montant, la devise et la catégorie de la facture.

Enregistrement des données : Enregistre les informations validées dans une base de données MySQL, favorisant la gestion et le suivi des transactions.

Traduction : Intègre des fonctionnalités de traduction du français vers l'anglais, l'espagnol et l'allemand.

Conversion de devises : Convertit automatiquement (pour les devises prises en charge) les montants d'une devise autre que l'euro en euros.

Analyse comptable : Fournit des outils pour analyser les dépenses par catégories et comparer les dépenses sur différentes périodes.

1.2 Architecture du système

L'architecture de notre système se divise en sept modules.

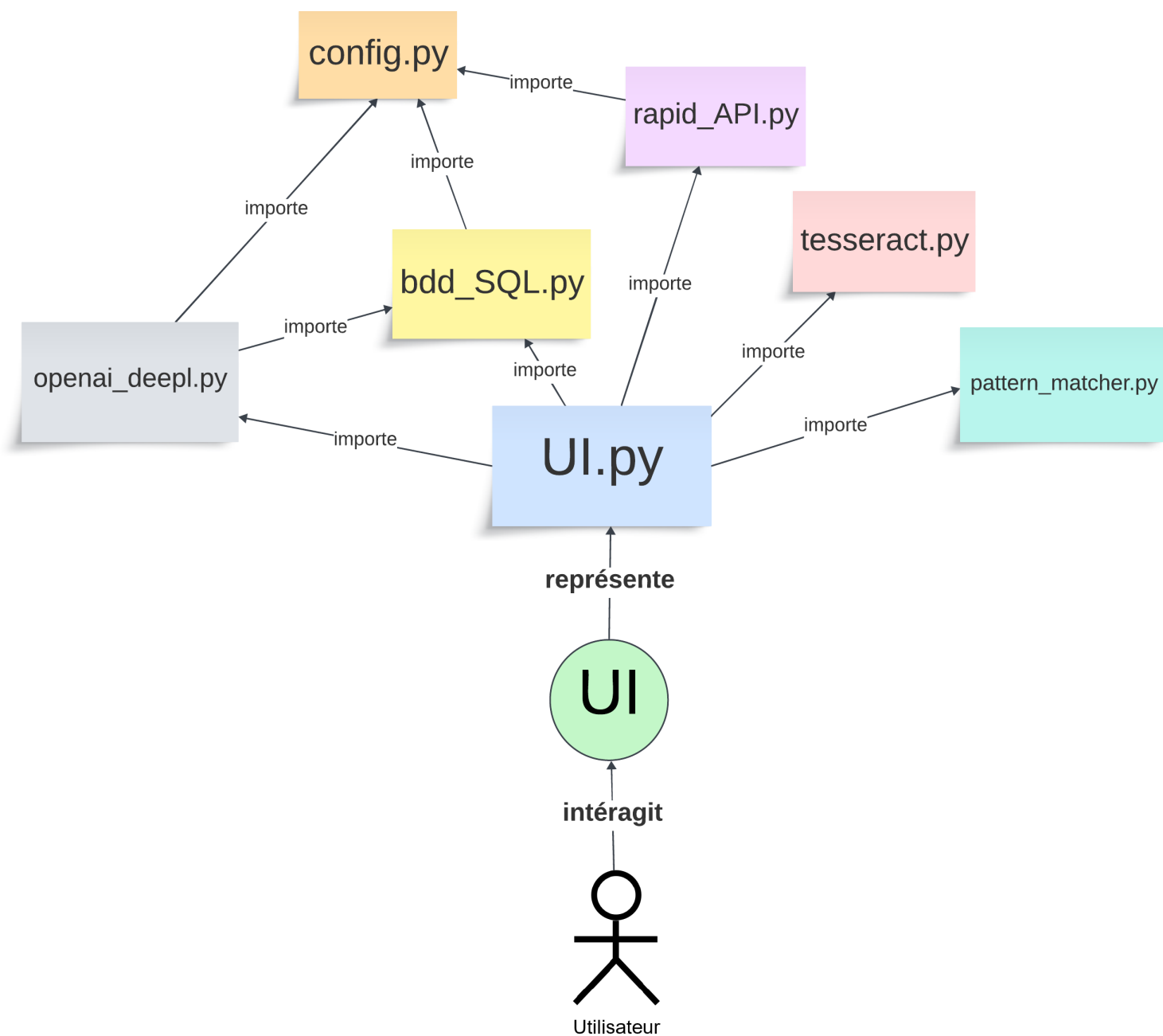


FIGURE 1 – Schéma de l'architecture du système

L'architecture de FactureFacile repose sur une division modulaire. Chaque module est conçu pour gérer un aspect spécifique du traitement des factures :

UI.py : Module principal gérant l'interface utilisateur. C'est le point d'interaction principal pour les utilisateurs, permettant la navigation et l'utilisation des différentes fonctionnalités du logiciel.

tesseract.py : Ce module utilise les bibliothèques OpenCV et Tesseract pour l'extraction de texte à partir des images de factures, transformant les données visuelles en texte manipulable.

openai_deepl.py : S'occupe du post-traitement des textes extraits et de leur traduction. Le post-traitement est effectué grâce à l'API de OpenAI et la traduction est gérée par l'API de DeepL.

config.py : Contient les clés API pour l'accès aux services externes (utilisés pour le post-traitement, la traduction et la conversion de devises) ainsi que les informations de connexion à la base de données.

pattern_matcher.py : Utilise un pattern matcher pour l'identification des informations pertinentes des textes extraits, telles que la date, le montant, la devise d'origine et la catégorie de la facture.

rapid_API.py : Gère la conversion des devises, utilisant les taux de change actuels pour convertir les montants des factures en euros, si la devise d'origine n'est pas celle-ci.

bdd_SQL.py : Module responsable de toutes les interactions avec la base de données MySQL. Il permet le stockage des données des factures, facilitant leur récupération et leur analyse ultérieure. La plupart des variables globales sont définies dans le module UI.py.

1.2.1 Les variables globales

Les variables globales utilisées tout au long de l'UI sont les suivantes :

- **temp_folder** : le dossier temporaire utilisé pour sauvegarder les factures à chaque étape de la lecture, de la correction et de la traduction.
- **Variables liées au design de l'UI** : telles que la taille des fenêtres, la taille des canvas, ainsi que le padding en x et en y.
- **Variables déterminant l'état de l'onglet 3** : ces variables contrôlent l'état spécifique de l'onglet 3.
- **rectangles** : la liste des rectangles dessinés par l'utilisateur.
- **img_cv** : une variable image.
- **liste_categorie_depense** : la liste des types de dépenses. Toute modification de cette liste nécessite une mise à jour de la liste **categories** de la fonction **get_categorie(content)** dans le module **pattern_matcher.py**, et vice-versa.
- **liste_devises** : la liste des devises utilisées dans le menu déroulant de l'onglet 2.

Des variables globales supplémentaires sont utilisées spécifiquement pour les onglets 1 à 3 et sont réinitialisées à chaque chargement d'une nouvelle facture via la fonction **initialisation()**. Ces variables incluent, par exemple, la taille de la facture, les éléments clés de la facture (émetteur, date, montant, devise, montant en euros, type de dépense) ainsi que la langue de traduction souhaitée.

En outre, d'autres variables globales sont spécifiques à l'onglet 4 (Comptabilité) et sont réinitialisées par les fonctions **initialisation_acc()** (partiellement) ou **initialisation_acc_full()**. Ces variables comprennent le mois et l'année sélectionnés par l'utilisateur ainsi que les montants affichés pour une période donnée.

Enfin, certaines variables globales sont définies dans le module **pattern_matcher.py** :

- **month_names** : un dictionnaire pour convertir les numéros de mois en noms français.
- **sort_folder** : le dossier utilisé pour classer les factures et leurs traductions (le cas échéant) par année, mois et type de dépense.

1.3 Détails des composants de l'interface utilisateur

Afin de détailler les différents composants de notre programme, nous procéderons à l'analyse de cinq fenêtres principales de l'interface utilisateur (UI), chacune étant abordée individuellement afin de suivre la logique et le flux d'opérations du programme. Cette approche nous permettra de comprendre comment chaque onglet contribue à la gestion et au traitement des factures, de l'importation d'images à la traduction des textes extraits.

1.3.1 Onglet 1 : Affichage de l'image

La première fenêtre de l'UI est dédiée au chargement et à la préparation initiale de l'image de la facture pour le traitement. Dans cette interface, l'utilisateur peut importer une image de facture depuis le système de fichiers. Une fois l'image importée, elle est affichée sur un canvas où l'utilisateur peut interagir directement avec l'image pour diviser l'image en zones de texte. Ces zones, représentées par des rectangles dessinés par l'utilisateur, délimitent les sections du texte à extraire. Cette division est nécessaire pour assurer une qualité optimale d'extraction de texte.

Les principales fonctionnalités de cet onglet comprennent :

Importation d'image : Permet à l'utilisateur de naviguer dans son système de fichiers et de charger une image ou un document PDF. Les fichiers PDF sont automatiquement convertis en images pour le traitement. Les formats acceptés sont : .jpg, .jpeg, .png et .pdf.

Affichage et manipulation d'image : L'image chargée est affichée sur un canvas. L'utilisateur est invité à y dessiner des rectangles pour indiquer les zones de texte pertinentes. Ces zones sont ensuite utilisées pour extraire le texte via OCR. La définition des zones de texte est primordiale afin d'assurer la qualité de l'extraction de texte.

Validation et extraction de texte : Après avoir marqué les zones nécessaires, l'utilisateur peut valider ses sélections, ce qui déclenche le processus d'extraction de texte. Le texte initialement extrait ainsi que le texte post-traité sont temporairement enregistrés et prêts pour les étapes de correction dans l'onglet suivant.

Cette fenêtre de l'UI est essentielle car elle initie le traitement des factures en préparant les données pour les étapes suivantes.

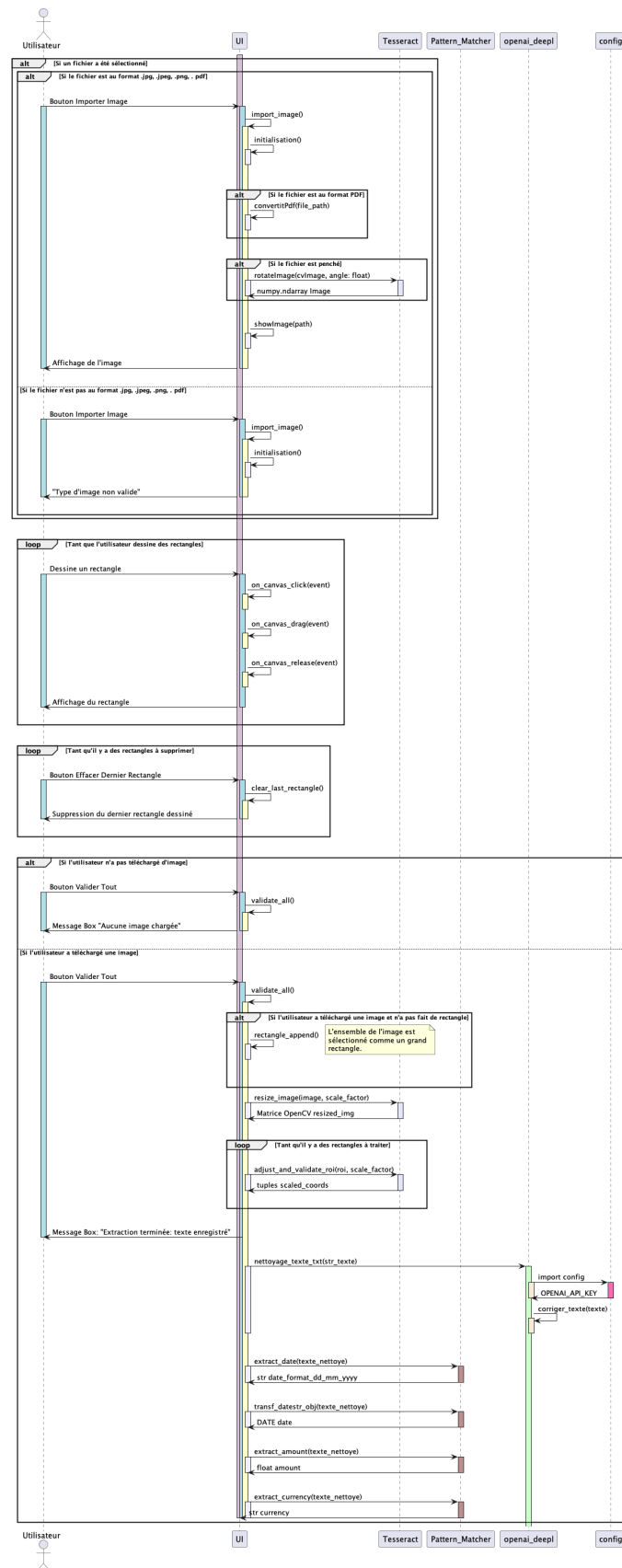


FIGURE 2 – Schéma détaillé du fonctionnement d’affichage de l’image (permet le zoom).

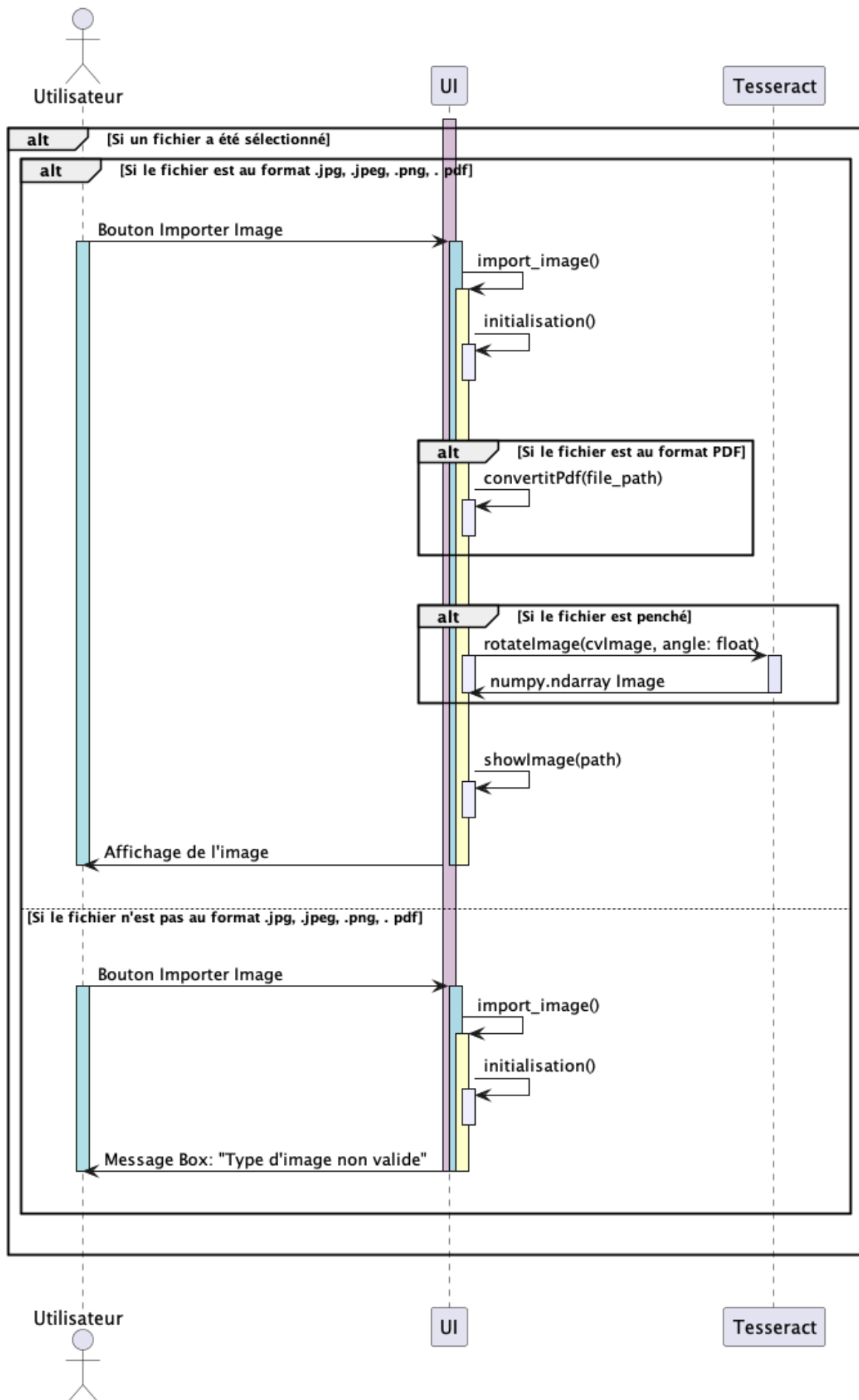


FIGURE 3 – Schéma détaillé du fonctionnement du chargement de l'image.

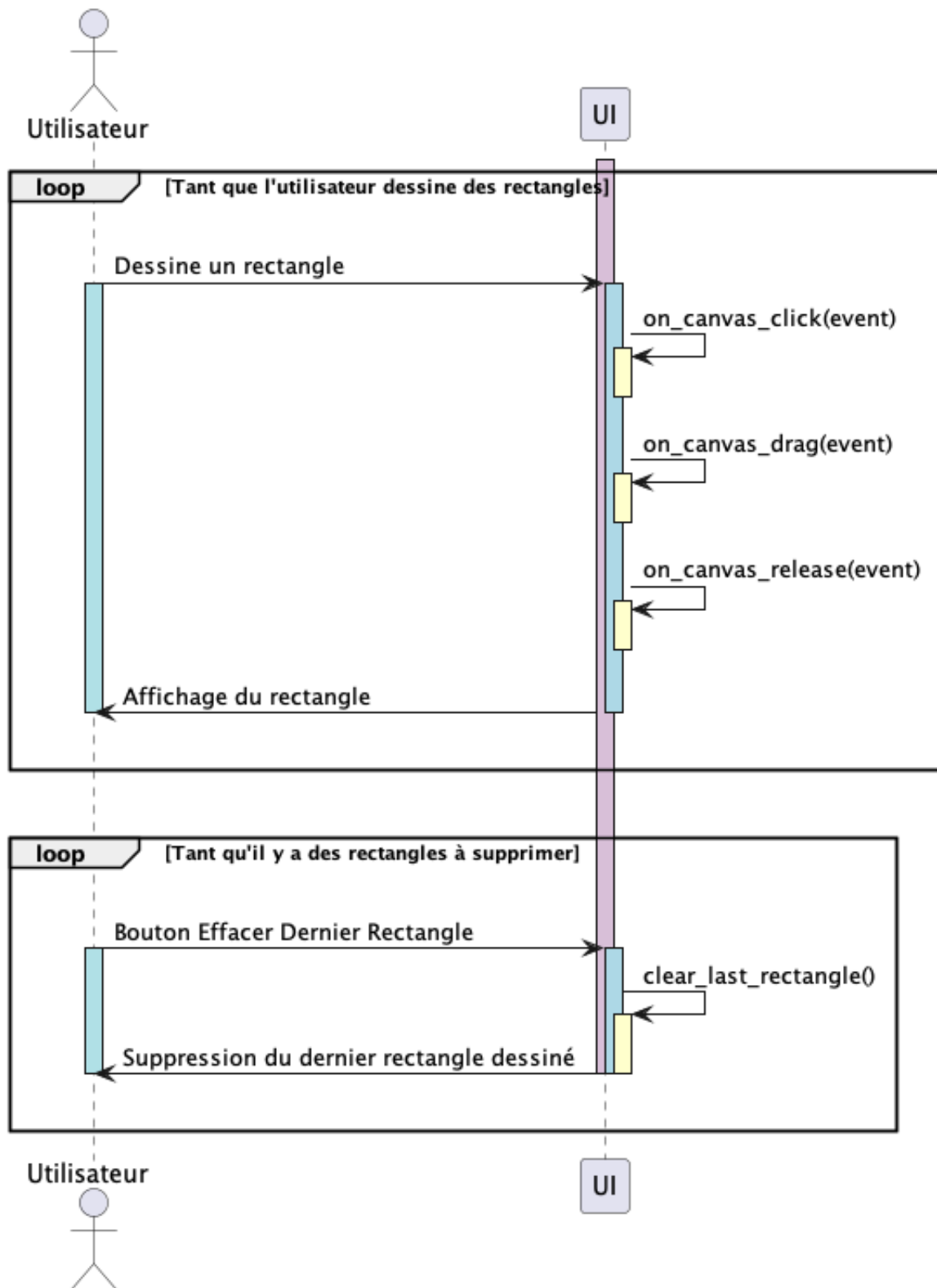


FIGURE 4 – Schéma détaillé du fonctionnement de la gestion des rectangles (création et suppression).

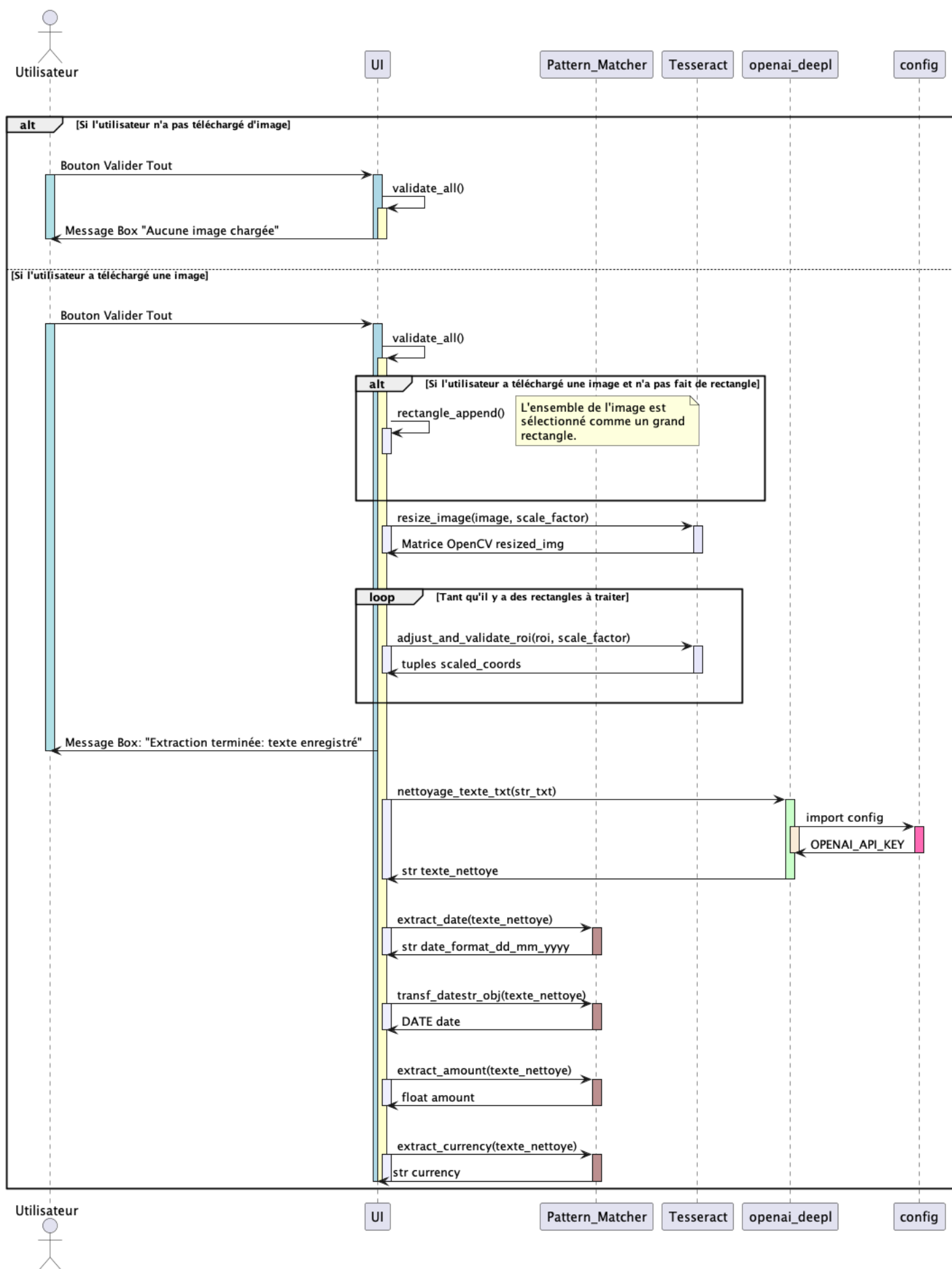


FIGURE 5 – Schéma détaillé du fonctionnement du bouton "Valider tout".

1.3.2 Onglet 2 : Vérification

L'onglet de vérification permet à l'utilisateur de vérifier et de corriger les données extraites de la facture. Cet onglet vérifie l'exactitude des informations avant leur enregistrement dans la base de données et leur traduction éventuelle.

Les principales fonctionnalités de cet onglet comprennent :

Renseignement, vérification et correction des données : L'utilisateur peut vérifier les informations extraites de la facture, telles que la date, le montant, la devise et la catégorie de dépense. Si nécessaire, il peut corriger ces informations directement dans les champs de saisie correspondants. Par ailleurs, l'utilisateur doit renseigner l'émetteur de la facture.

Validation des données : Une fois les données complétées, vérifiées et corrigées, l'utilisateur peut cliquer sur le bouton de validation. Cette action déclenche une série de vérifications pour s'assurer que tous les champs obligatoires sont remplis. Si des champs obligatoires sont manquants, une boîte de dialogue informe l'utilisateur des éléments manquants à compléter.

Validation du texte avant traduction : Si l'utilisateur souhaite obtenir une traduction du texte extrait, il peut cliquer sur le bouton permettant l'affichage du texte avant la traduction. Le texte nettoyé est alors affiché dans une zone de texte où l'utilisateur peut apporter des corrections avant de valider le texte.

Sélection de la langue : L'utilisateur peut choisir la langue cible pour la traduction parmi une liste de langues disponibles (anglais, espagnol et allemand). La langue sélectionnée est ensuite utilisée pour traduire le texte corrigé.

Traduction du texte : En cliquant sur le bouton de traduction, le texte corrigé est envoyé pour traduction. Les données concernant le texte traduit (son nombre de caractères et sa langue cible) sont ensuite enregistrées dans la base de données. Si la langue n'est pas renseignée, une boîte de dialogue informe l'utilisateur de la nécessité de spécifier la langue cible.

Cet onglet garantit que les données extraites sont exactes et complètes avant leur enregistrement dans la base de données et leur traduction. Il assure la qualité des informations traitées.

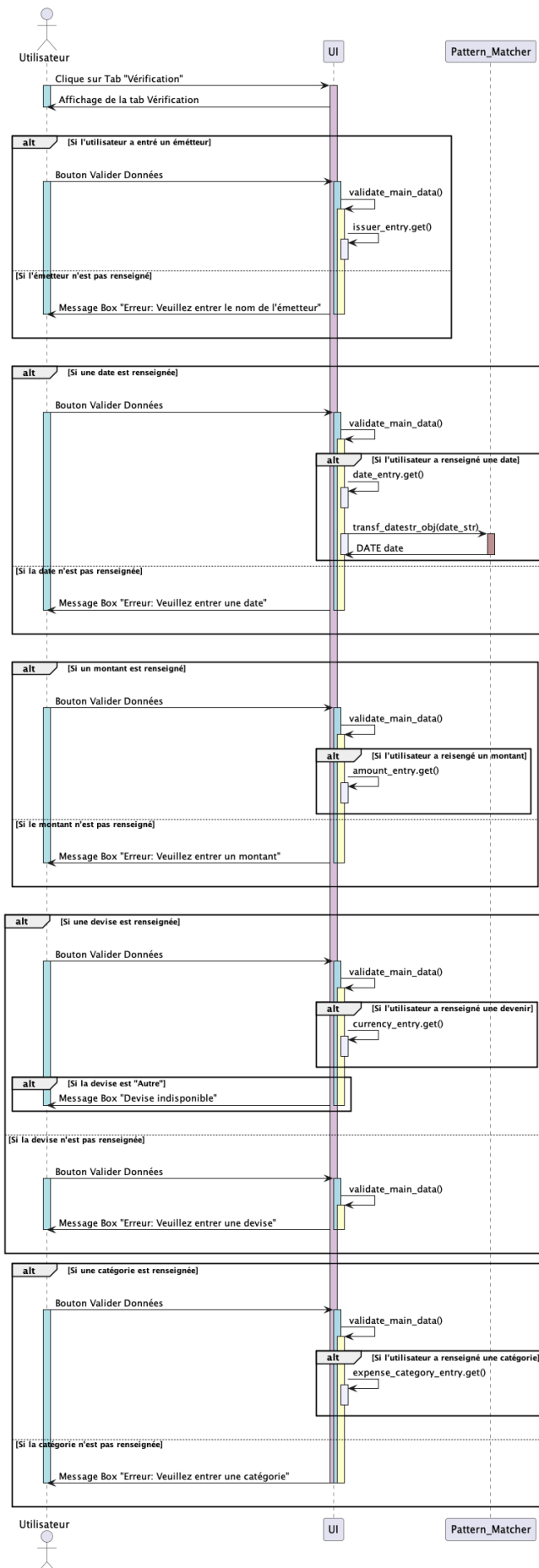


FIGURE 7 – Schéma détaillé du fonctionnement de la vérification des champs.

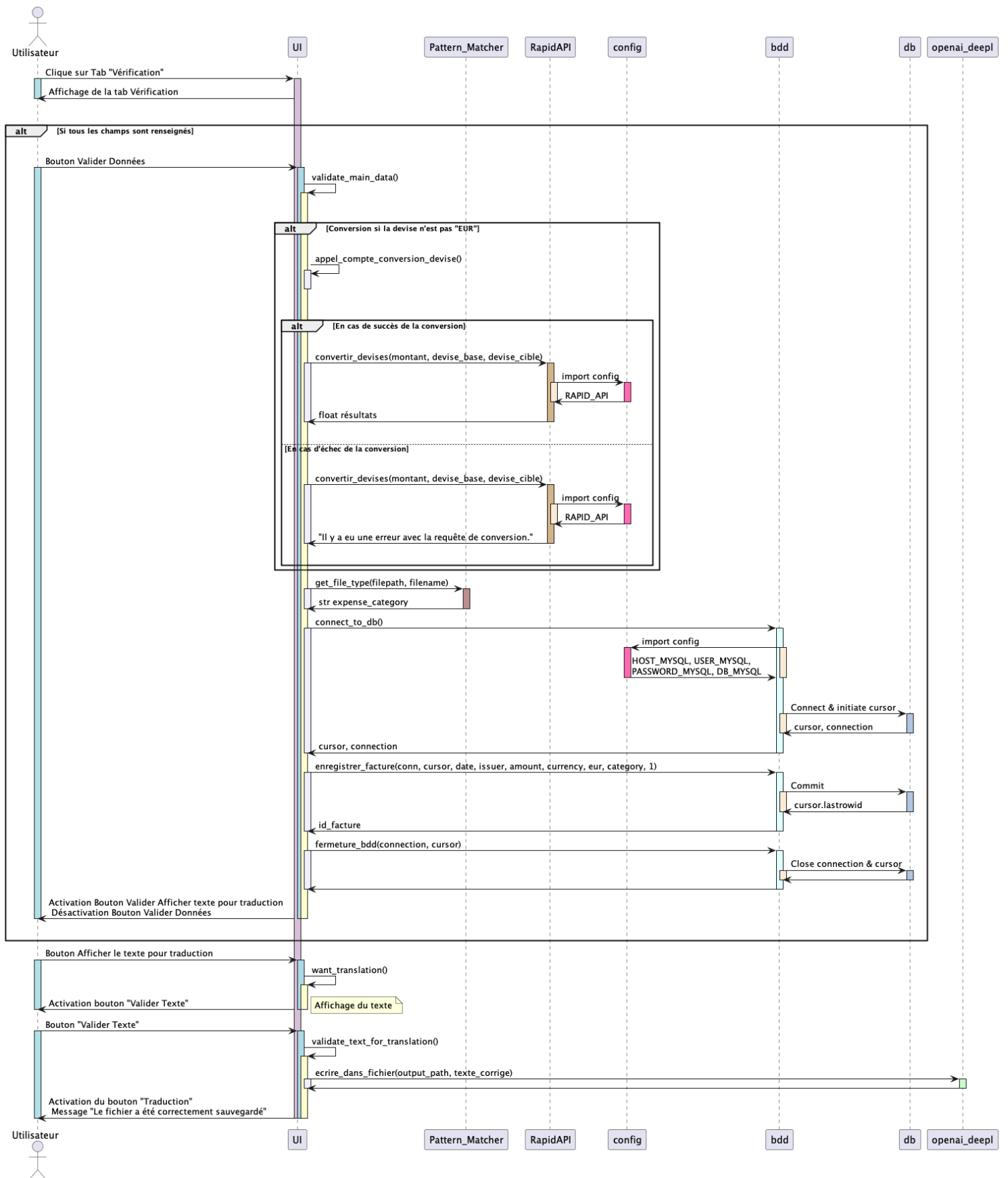


FIGURE 8 – Schéma détaillé du fonctionnement de la validation des données.

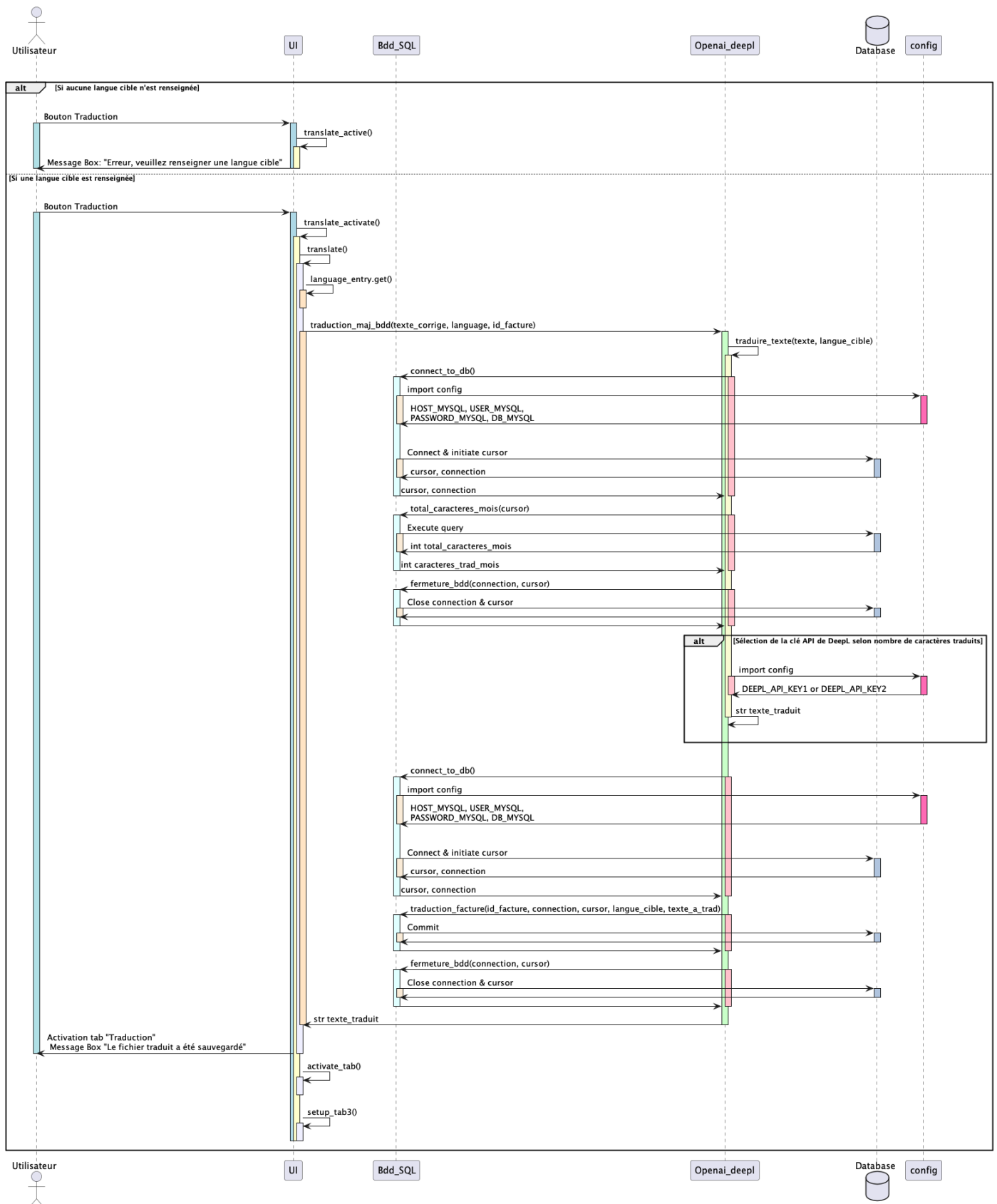


FIGURE 9 – Schéma détaillé du fonctionnement de la correction et de la validation du texte pour traduction.

1.3.3 Onglet 3 : Traduction

L'onglet de traduction permet à l'utilisateur de visualiser et de corriger le texte extrait de la facture puis traduit. Cet onglet permet aussi d'apporter des modifications au texte traduit avant d'enregistrer la version finale.

Les principales fonctionnalités de cet onglet comprennent :

Affichage du texte traduit : Après avoir effectué la traduction dans l'onglet de vérification, le texte traduit est automatiquement affiché dans une zone de texte dans cet onglet. L'utilisateur peut visualiser le texte traduit pour s'assurer qu'il est correct.

Édition du texte traduit : L'utilisateur peut apporter des corrections directement dans la zone de texte si des ajustements sont nécessaires.

Sauvegarde de la traduction : La traduction initialement proposée est automatiquement sauvegardée. Néanmoins, l'utilisateur peut apporter des modifications à la traduction proposée. Lorsqu'il cliquera sur le bouton de sauvegarde, cela écrasera la version enregistrée précédemment.

Cet onglet permet de gérer la correction et l'enregistrement des textes traduits, assurant une gestion multilingue des documents.

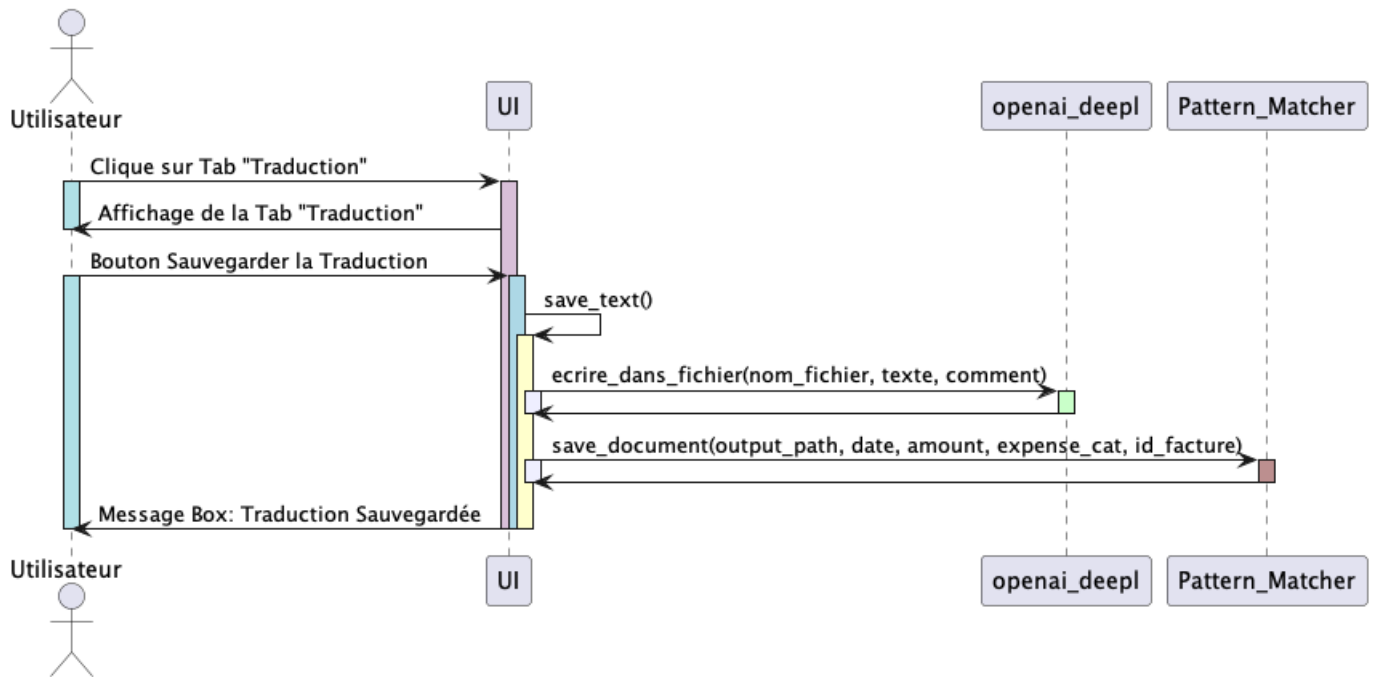


FIGURE 10 – Schéma détaillé du fonctionnement de l'onglet de traduction.

1.3.4 Onglet 4 : Comptabilité

L'onglet de comptabilité permet à l'utilisateur de visualiser les montants des dépenses par catégorie pour une période donnée.

Les principales fonctionnalités de cet onglet comprennent :

Sélection de la période : L'utilisateur peut sélectionner un mois et une année spécifiques, ou choisir d'afficher les dépenses pour une année entière. Si l'utilisateur ne sélectionne pas d'année, un message d'erreur est affiché pour indiquer que la sélection de l'année est obligatoire. Les années prises en charge sont : 2018, 2019, 2020, 2021, 2022, 2023 et 2024.

Affichage des montants par catégorie : Lorsque l'utilisateur clique sur le bouton "Afficher les montants", l'application présente les totaux des dépenses pour chaque catégorie durant la période sélectionnée. Si un mois d'une année spécifique est choisi, les montants pour ce mois seront affichés, ainsi que les montants pour la même période de l'année précédente. Les données sont extraites de la base de données et affichées.

Détails des factures par catégorie : L'utilisateur peut sélectionner une catégorie spécifique et cliquer sur le bouton "Liste des factures" pour afficher les détails des factures correspondant à cette catégorie pour la période sélectionnée. Si aucune catégorie n'est sélectionnée, un message est affiché.

Détails d'une facture spécifique et actions associées : L'utilisateur peut entrer l'ID d'une facture spécifique et cliquer sur le bouton "Détails de la facture" pour afficher les détails de cette facture. L'utilisateur peut également afficher l'image correspondant à la facture ainsi que sa traduction (si la facture a été traduite). Si l'utilisateur clique le bouton "Effacer la facture", la facture sera effacée de la base de données. Si l'ID de la facture n'est pas renseigné, un message est affiché. Si l'ID ne correspond à aucune facture dans la base de données, un message s'affiche également.

Réinitialisation de l'onglet : Si l'utilisateur clique sur le bouton "Réinitialiser la fenêtre", le contenu de l'onglet est réinitialisé.

Cet onglet permet d'analyser les dépenses et d'obtenir une vue détaillée des factures par catégorie. Par ailleurs, cet onglet permet également d'interagir directement avec la base de données en recherchant des informations précises ou en supprimant des factures.

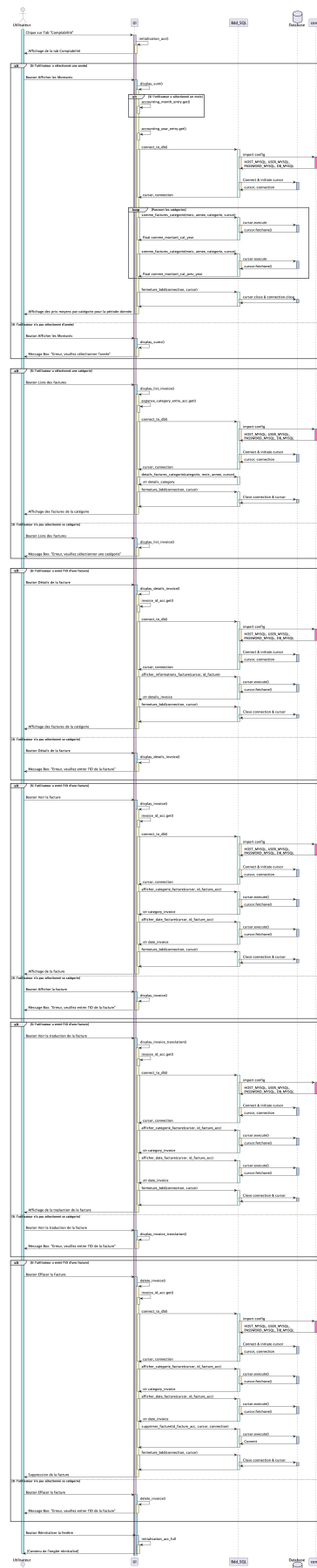


FIGURE 11 – Schéma détaillé du fonctionnement de l'onglet de comptabilité (permet le zoom).

1.3.5 Onglet 5 : Statistiques

L'onglet de statistiques offre à l'utilisateur la possibilité d'obtenir diverses informations statistiques sur les factures traitées.

Les principales fonctionnalités de cet onglet comprennent :

Sélection de la période : L'utilisateur peut sélectionner un mois et une année spécifiques, ou choisir d'afficher les statistiques depuis toujours. Si l'utilisateur choisit un mois sans sélectionner une année, un message s'affiche pour indiquer que la sélection de l'année est obligatoire. Lors de la sélection d'une nouvelle année, le mois est réinitialisé à "choisir le mois", obligeant l'utilisateur à sélectionner un nouveau mois s'il veut des statistiques pour un mois précis de l'année.

Affichage des statistiques globales : Les utilisateurs peuvent obtenir des statistiques globales telles que le nombre de factures traitées, la catégorie la plus fréquente, le nombre de factures traduites, le prix moyen des factures, le prix moyen des factures pour une catégorie spécifique, les langues de traduction les plus fréquentes, le nombre de caractères traduits et le nombre de conversions de devise effectuées pour le mois en cours.

Résultats par catégorie spécifique : Pour certaines catégories de statistiques, l'utilisateur peut spécifier une catégorie de dépense particulière afin d'obtenir des résultats plus précis, notamment pour le prix moyen des factures d'une catégorie précise.

Cet onglet permet une analyse approfondie des données qui est surtout utile pour les développeurs de l'application.

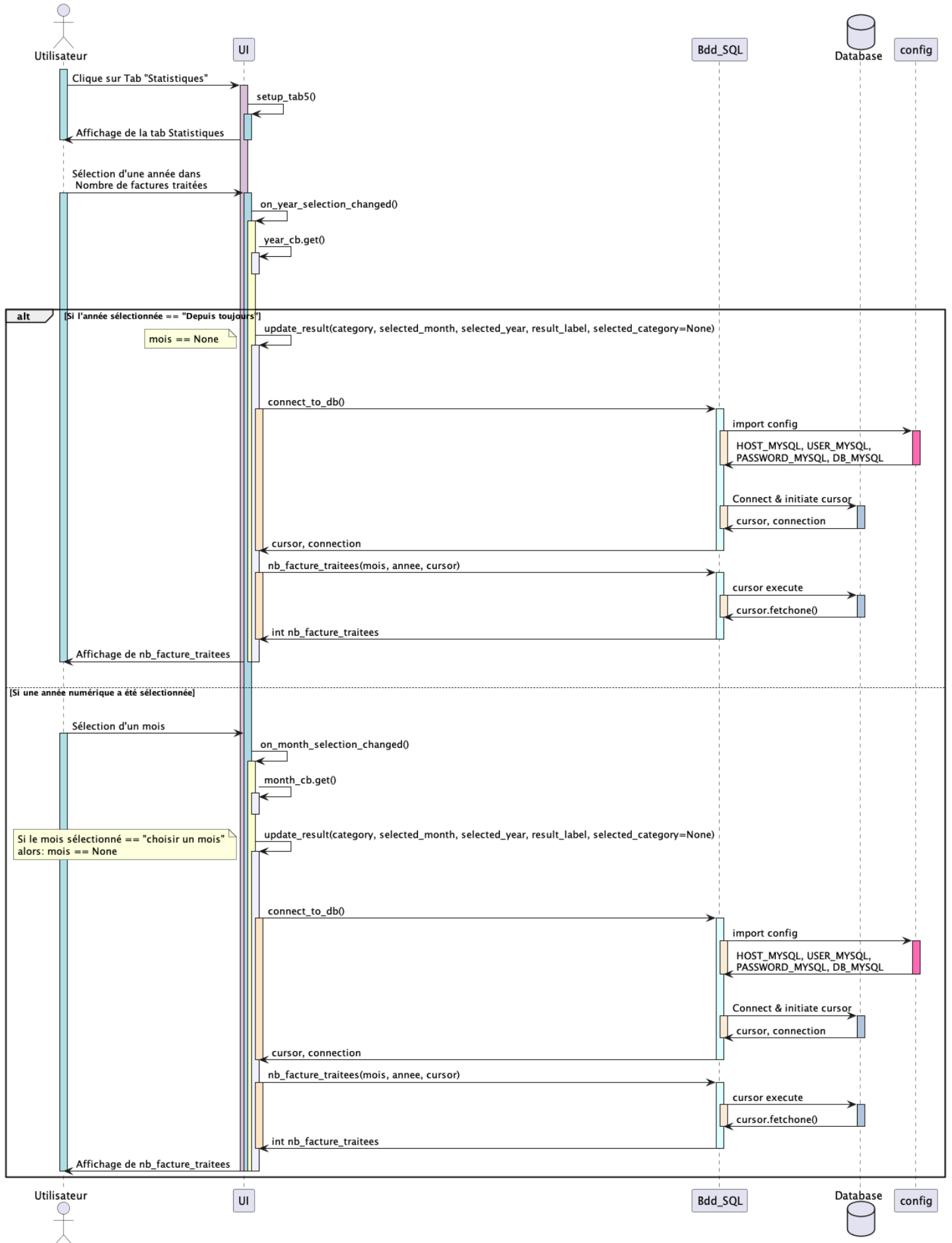


FIGURE 12 – Schéma détaillé du fonctionnement de l'onglet statistiques (permet le zoom).

1.4 Technologies utilisées et limitations

Le programme est développé en utilisant plusieurs technologies et outils. Voici une liste des principales technologies utilisées :

- **Langage de programmation** : Le programme est entièrement écrit en Python.

- **Base de données** :

MySQL est utilisé pour stocker et gérer les informations des factures.

Notre base de données est hébergée sur Azure (URL : <https://azure.microsoft.com/>).

Nous disposons d'un accès gratuit (via Azure for Students) à Azure Database for MySQL - Flexible Server pendant 12 mois avec une limitation à 32 GB de stockage et 32 GB de stockage de sauvegarde.

Voici les informations techniques relatives à la base de données : **Burstable, B1ms, 1 vCores, 2 GiB RAM, 32 GiB s**

Les identifiants et informations utiles pour la connexion à la base de données se trouvent dans le fichier `config.py`.

- **API** :

- **DeepL** : Utilisé pour les traductions vers l'espagnol, l'anglais et l'allemand.

URL : <https://developers.deepl.com/docs>

La version gratuite de l'API de DeepL est limitée à 500 000 caractères traduits par mois. Nous disposons de deux clés qui sont utilisées selon le nombre de caractères déjà traduits. Ainsi, nous sommes limités à 1 000 000 de caractères traduits par mois.

- **OpenAI** : Utilisé pour le post-traitement du texte extrait.

URL : <https://openai.com/api/>

Nous avons souscrit à 15\$ de crédit auprès de l'API de OpenAI. Cela correspond à environ 7 millions de tokens en entrée et 7 millions de tokens en sortie. Ce qui équivaut à environ 28 millions de caractères traités dans le cadre du post-traitement.

- **Currency Conversion and Exchange Rates API** : Fournit les taux de change et les conversions de devises.

URL : <https://rapidapi.com/principalapis/api/currency-conversion-and-exchange-rates/details>

La version gratuite de l'API permet d'effectuer 1 000 conversions par mois.

- **Bibliothèques et frameworks Python** :

- **tkinter** : Pour la création de l'interface utilisateur.
- **tkcalendar** : Pour la gestion des dates dans l'interface.
- **Pillow** : Pour le traitement d'images.
- **OpenCV** : Pour la manipulation avancée d'images.
- **pytesseract** : Pour l'extraction de texte des images (OCR).
- **mysql-connector-python** : Pour la connexion à la base de données MySQL.
- **deepl** : Pour l'intégration avec l'API DeepL.
- **requests** : Pour les requêtes HTTP.
- **dateparser** : Pour la gestion des dates.
- **openai** : Pour l'intégration avec l'API OpenAI.
- **PyMuPDF** : Pour la manipulation des fichiers PDF.

- **Composants logiciels supplémentaires** :

Les logiciels suivants sont installés via le script `install.sh` pour le bon fonctionnement du programme :

- **python3-pip** : Utilisé pour la gestion des packages Python.
- **tesseract-ocr** : Utilisé pour l'extraction de texte à partir des images (OCR).

- **tesseract-ocr-fra** : Module linguistique français pour Tesseract OCR.
 - **python3-tk** : Utilisé pour les interfaces graphiques avec Tkinter.
- **Outils de développement** :
 - **Hatchling** : Utilisé pour le packaging et la gestion des dépendances du projet.

1.5 Packaging

Le projet utilise Hatchling comme système de build pour gérer le packaging et les dépendances. Voici le fichier de configuration du packaging `pyproject.toml` :

```

1 [build-system]
2 requires = ["hatchling"]
3 build-backend = "hatchling.build"
4
5 [project]
6 name = "FactureFacile"
7 version = "0.1.2"
8 authors = [
9   { name = "Avrile Floro", studentID = "22000086" },
10  { name = "Aude Hennino", studentID = "19011803" },
11  { name = "Maxime Bronny", studentID = "19009314" }
12 ]
13
14 requires-python = ">=3.10"
15 dependencies = [
16   "mysql-connector-python>=8.4.0",
17   "Pillow>=10.3.0",
18   "opencv-python>=4.9.0.80",
19   "pytesseract>=0.3.10",
20   "deepl>=1.18.0",
21   "requests>=2.31.0",
22   "dateparser>=1.2.0",
23   "tkcalendar>=1.6.1",
24   "openai>=1.25.1",
25   "PyMuPDF>=1.24.2",
26 ]
27
28 [tool.hatch.build.targets.wheel]
29 packages = ["src"]
30
31 classifiers = [
32   "Programming Language :: Python :: 3",
33   "License :: OSI Approved :: MIT License",
34   "Operating System :: OS Independent",
35 ]

```

Cette configuration inclut toutes les dépendances nécessaires pour le projet, permettant une gestion facilitée des environnements de développement.

1.6 Configurations minimales requises

D'après le fichier `pyproject.toml` et les bibliothèques utilisées, voici les configurations minimales requises pour ce projet :

1.6.1 Langage de Programmation

- **Python** : version `>= 3.10`

1.6.2 Bibliothèques et dépendances

- `mysql-connector-python` : version `>= 8.4.0`
- `Pillow` : version `>= 10.3.0`
- `opencv-python` : version `>= 4.9.0.80`
- `pytesseract` : version `>= 0.3.10`
- `deepl` : version `>= 1.18.0`
- `requests` : version `>= 2.31.0`
- `dateparser` : version `>= 1.2.0`
- `tkcalendar` : version `>= 1.6.1`
- `openai` : version `>= 1.25.1`
- `PyMuPDF` : version `>= 1.24.2`

1.7 Construction du packaging

Pour construire le paquet, les étapes suivantes ont été nécessaires sur un système d'exploitation Linux :

1. Installation de pip : `sudo apt install python3-pip`
2. Installation de build : `sudo apt install build`
3. Installation de Hatchling : `sudo apt install hatchling`
4. Installation de l'environnement virtuel : `sudo apt install python3.10-venv`
5. Création du fichier wheel (depuis le répertoire parent de `src`) : `python3 -m build`

1.8 Procédure d'installation et de configuration

1.8.1 Contenu du dossier projet

Voici les étapes à suivre pour installer les dépendances nécessaires à l'exécution du programme :

1. Télécharger le fichier `.tar.gz` sur GitHub au lien suivant : <https://github.com/CryZiiiX/PROJECTL2>
2. Extraire le contenu de l'archive téléchargée. Un dossier **projet** qui se décompose comme suit sera obtenu :
 - Un script `install.sh` qui permet de procéder aux installations nécessaires à l'exécution du programme.
 - Un fichier `.wheel` qui permet de télécharger les dépendances python.
 - Un répertoire `src` qui contient le code source.
 - Un répertoire `src_avec_tests` qui contient le code source ainsi que les fichiers nécessaires pour l'exécution des tests.
 - Un répertoire `factures` proposant des factures à utiliser avec le programme.

1.8.2 Installations préalables à l'exécution du projet

Le script `install.sh` lance l'installation des composants logiciels nécessaires (`tesseract`, `pip`, `tkinter`) et va ensuite procéder à l'installation des dépendances via l'installation du paquet `.wheel`. Voici le contenu du script :

```
1 #!/bin/bash
2
3 # mettre à jour les dépôts et installer Python3-pip, Tesseract OCR et Tkinter
4 sudo apt-get update
5 sudo apt-get install -y python3-pip tesseract-ocr tesseract-ocr-fra python3-tk
6
7 # installer les dépendances Python
8 pip install facturefacile-0.1.2-py3-none-any.whl
9
10 echo "Installation terminée avec succès."
```

Afin de procéder aux installations nécessaires, via le script `install.sh`, il convient de se placer dans le répertoire où se trouve le script. La commande suivante permet d'attribuer au script des droits d'exécution :

```
1 chmod +x install.sh
```

Afin d'exécuter le script, il faut se placer dans le répertoire adéquat (le répertoire où se trouve le script) et lancer la commande :

```
1 ./install.sh
```

1.8.3 Lancer le programme

Pour lancer le projet, il faut exécuter le fichier `UI.py` qui se trouve dans le répertoire `src`.

1.8.4 Consulter le code source

Le code source se trouve dans le répertoire `projet/src`.