UVSQ université PARIS-SACLAY

ISTY
Institut des Sciences et Techniques des Yvelines
CAMPUS DE MANTES EN YVELINES
CAMPUS DE SAINT-QUENTIN-EN-YVELINES

Automatisation de l'évaluation des demandes de prêt immobilier FastAPI

Préparé par : Rochdi DARDOR

IATIC5

2024/2025

TABLE DES MATIÈRES :

1. INTRODUCTION :

Contexte

Dans le secteur bancaire, les processus d'évaluation de demandes de prêt immobilier sont souvent longs et nécessitent une analyse approfondie de plusieurs facteurs. L'automatisation de ces processus permettrait d'augmenter la rapidité de traitement tout en assurant une évaluation précise et fiable. Le présent travail pratique (TP) vise à concevoir et développer un système capable de traiter les demandes de prêt immobilier de manière automatisée, en intégrant une série de services web spécialisés.
Le projet repose sur le développement d'un Service Web Composite, c'est-à-dire une architecture composée de plusieurs services individuels coordonnés. Dans le TP précédent, cette architecture avait été implémentée en utilisant **Spyne** et des services SOAP. Dans ce TP, l'objectif est de réaliser une solution similaire en exploitant **FastAPI**, une technologie moderne pour le développement d'APIs REST.

Problématique

Les institutions financières doivent analyser plusieurs aspects pour chaque demande de prêt, y compris :

- Les informations personnelles et financières du client.
- La solvabilité du client, basée sur ses revenus et dépenses mensuels.
- La valeur de la propriété pour laquelle le prêt est demandé, ainsi que son adéquation avec le marché immobilier.

Ce projet permet d'automatiser chacune de ces étapes en développant des services web autonomes, chacun d'eux se concentrant sur un aspect spécifique de l'évaluation.

Objectifs du Projet

L'objectif principal du TP est de développer un système d'évaluation automatisée des demandes de prêt immobilier en exploitant des services RESTful sous FastAPI. Ce système devra :

1. **Fournir une extraction automatique des informations** contenues dans la demande de prêt soumise par le client.
2. **Évaluer la solvabilité du client** en fonction de ses revenus, de ses dépenses et du montant du prêt demandé.
3. **Estimer la valeur de la propriété** basée sur des critères d'évaluation du marché immobilier.
4. **Prendre une décision d'approbation ou de refus** de la demande de prêt en intégrant les résultats des analyses précédentes.

Solution Proposée

La solution adoptée s'appuie sur une architecture orientée services (SOA) avec FastAPI. Cette architecture présente plusieurs avantages :

- **Modularité** : Chaque fonction (extraction, solvabilité, évaluation, décision) est isolée dans un service spécifique, facilitant ainsi la maintenance et les mises à jour futures.
- **Interopérabilité** : En utilisant le protocole HTTP et des données au format JSON, les services peuvent facilement être intégrés dans d'autres systèmes ou plateformes.
- **Scalabilité** : Les services peuvent être adaptés ou redéployés indépendamment selon les besoins opérationnels, sans impact sur l'ensemble du système.

2. CONCEPTION DU PROJET :

Présentation du Scénario :

Dans le scénario ciblé par ce projet, un utilisateur (client) souhaite soumettre une demande de prêt immobilier. La demande inclut plusieurs informations exprimées en langage naturel, telles que le nom et l'adresse du client, le montant du prêt souhaité, des détails sur la propriété ainsi que des données financières sur le revenu et les dépenses du client.
Une fois la demande soumise, le système déclenche un processus automatisé qui passe par les étapes suivantes :

1. **Extraction des informations** : Le système utilise un service d'extraction qui analyse le contenu de la demande, en identifiant et en structurant les informations essentielles pour l'évaluation du prêt (nom, montant, revenu, dépenses, etc.).
2. **Vérification de la solvabilité** : Le système détermine si le client dispose de suffisamment de ressources financières pour rembourser le prêt demandé. Cette évaluation repose sur l'analyse de données telles que les revenus et les dépenses du client, en calculant notamment le ratio d'endettement.
3. **Évaluation de la propriété** : La demande est ensuite soumise à un service d'évaluation de la propriété qui utilise des critères prédéfinis pour estimer la valeur marchande de la propriété. Ce service garantit que la propriété répond aux critères financiers et légaux nécessaires pour l'approbation du prêt.
4. **Décision d'approbation** : Finalement, une décision d'approbation ou de refus du prêt est prise en fonction des résultats des étapes précédentes. La décision peut également inclure des recommandations ou conditions particulières selon le niveau de risque.

Objectifs

Les objectifs du TP sont les suivants :

- **Automatiser le processus d'évaluation des demandes de prêt** en intégrant les services nécessaires dans une architecture unique, rapide et fiable.
- **Améliorer la rapidité de traitement** des demandes en évitant les processus manuels, qui sont chronophages et sujets aux erreurs.
- **Garantir la précision dans l'évaluation des prêts** grâce à des critères stricts appliqués à chaque étape du processus.
- **Assurer la traçabilité et le stockage structuré des données** en sauvegardant chaque demande et son évaluation dans une base de données dédiée.

Solution Adoptée

La solution repose sur une architecture microservices, chaque étape du processus étant gérée par un service distinct. La coordination des services est assurée par un orchestrateur, qui suit les étapes d'évaluation dans un ordre prédéfini. Voici une description succincte de chaque service :

- **Service d'Extraction** : Enregistre les informations de la demande de prêt en les structurant pour les étapes suivantes.
- **Service de Solvabilité** : Analyse la capacité de remboursement du client.
- **Service d'Évaluation** : Assure la conformité de la propriété et évalue sa valeur en fonction des normes du marché.
- **Service de Décision** : Prend une décision d'approbation ou de refus en fonction des résultats cumulés des services précédents.

L'architecture permet une interopérabilité entre les différents services et garantit un traitement rapide et fiable des demandes.

3. ARCHITECTURE DU SYSTÈME

Principes de l'Architecture SOA (Service-Oriented Architecture)

L'architecture de ce projet repose sur une approche orientée services (SOA). Cette architecture est particulièrement adaptée aux applications complexes nécessitant plusieurs étapes de traitement de données, telles que l'évaluation de demandes de prêt. Les principaux principes SOA adoptés dans cette architecture sont :

- **Modularité** : Chaque service est conçu pour effectuer une tâche spécifique, facilitant ainsi le développement, la maintenance et les mises à jour.

- **Interopérabilité** : En utilisant HTTP et des échanges de données au format JSON, chaque service peut communiquer facilement avec les autres, ce qui permet leur intégration dans d'autres systèmes si nécessaire.
- **Réutilisabilité** : Les services développés dans ce projet peuvent être réutilisés dans d'autres applications ou projets nécessitant des fonctionnalités similaires, comme l'évaluation de la solvabilité ou l'extraction de données.
- **Scalabilité** : Les services peuvent être adaptés indépendamment pour répondre à des exigences de charge plus élevées, par exemple en ajoutant des instances de services sans impacter l'ensemble de l'application.

Composants de l'architecture :

Le système est composé de plusieurs composants principaux, chacun ayant un rôle bien défini :

1. **Client** : Il s'agit de l'utilisateur final ou de l'application frontale qui soumet les demandes de prêt. Les demandes peuvent être introduites sous la forme de fichiers texte contenant les informations nécessaires.
2. **Orchestrateur** : Ce composant est responsable de la coordination des services. Il surveille un dossier pour détecter les nouvelles demandes et déclenche les appels aux services dans un ordre prédéfini. Chaque résultat est transmis au service suivant jusqu'à la prise de décision finale.
3. Services Web :
    - **Service d'Extraction d'Informations** : Ce service analyse le texte brut de la demande pour en extraire des informations structurées.
    - **Service de Vérification de Solvabilité** : Ce service évalue la capacité de remboursement du client en calculant un ratio d'endettement basé sur les revenus et les dépenses du client.
    - **Service d'Évaluation de la Propriété** : Ce service estime la valeur de la propriété décrite dans la demande et vérifie sa conformité aux normes du marché.
    - **Service de Décision d'Approbation** : Ce service compile les résultats des services précédents et prend une décision d'approbation ou de refus.
4. **Base de Données** : Toutes les demandes de prêt, y compris les résultats de chaque service et la décision finale, sont enregistrées dans une base de données SQLite. Cette base de données permet de conserver un historique des demandes et d'assurer la traçabilité des décisions.
    -

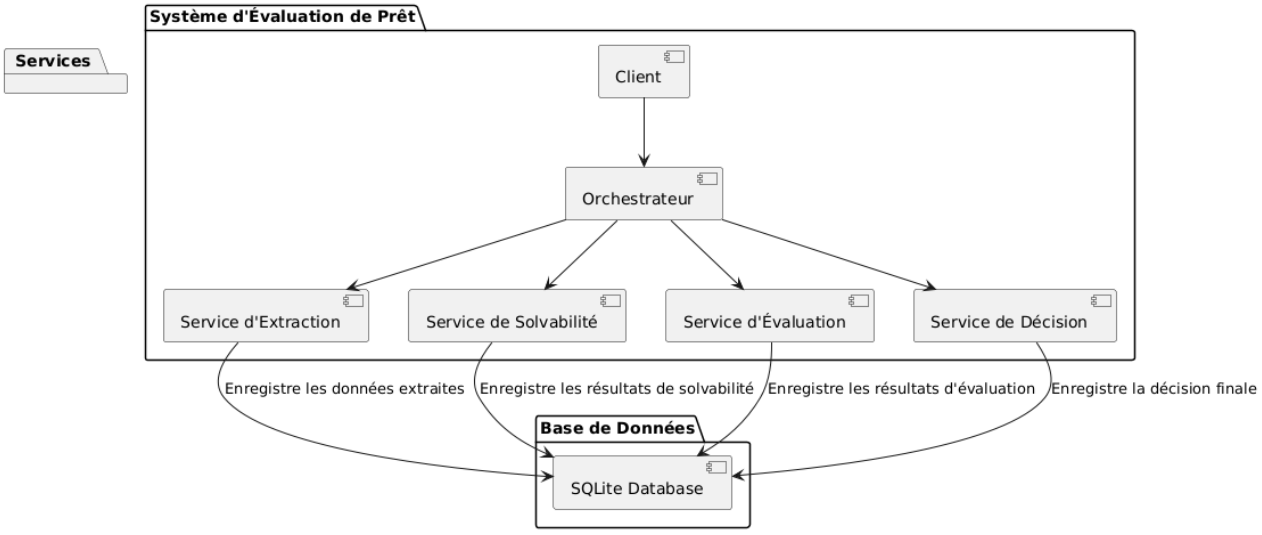
Flux de travail :

Soumission de la demande : Un client soumet une demande de prêt sous forme de fichier texte dans un dossier surveillé par l'orchestrateur.
Détection de la demande : L'orchestrateur détecte la nouvelle demande et initie le traitement.
Service d'extraction d'informations : L'orchestrateur envoie la demande au service d'extraction pour récupérer et structurer les données importantes.

Service de solvabilité : Les informations extraites (revenu, dépenses, montant du prêt) sont envoyées au service de solvabilité pour évaluer la capacité de remboursement du client.
Service d'évaluation de la propriété : La description de la propriété et le montant estimé du prêt sont envoyés au service d'évaluation pour estimer la valeur de la propriété et vérifier sa conformité.
Service de décision : Les résultats de la solvabilité et de l'évaluation de la propriété sont transmis au service de décision, qui approuve ou refuse la demande.
Enregistrement en base de données : Tous les résultats sont stockés dans la base de données pour un suivi et une analyse futurs.

Diagramme d'architecture :

Le diagramme d'architecture met en évidence les relations entre les composants du système, en montrant comment les services sont coordonnés par l'orchestrateur pour garantir une évaluation complète et précise des demandes de prêt.



4. DÉVELOPPEMENT DES SERVICES WEB (FASTAPI) :

Les services ont été développés en utilisant FastAPI, un framework rapide et moderne pour la création d'APIs REST en Python. FastAPI offre des performances élevées et une syntaxe simple, facilitant le développement et le déploiement des services. Chaque service dispose de son propre endpoint pour recevoir et traiter les demandes, puis renvoyer les résultats au format JSON.

4.1 Service d'Extraction d'Informations

Le **service d'extraction d'informations** est le premier composant de traitement de la demande de prêt. Ce service prend en entrée un texte brut (contenu de la demande) et utilise des techniques de traitement de texte pour identifier les informations clés telles que le nom du client, l'adresse, le revenu mensuel, les dépenses, et le montant du prêt demandé. Les données extraites sont ensuite structurées en JSON pour être utilisées par les services suivants.

- **Fonction principale** : Extraire et structurer les données contenues dans la demande de prêt.
- **Entrée** : Texte brut de la demande.
- **Sortie** : Données structurées (nom, montant du prêt, revenu mensuel, dépenses mensuelles, description de la propriété).

4.2 Service de Vérification de Solvabilité

Le **service de vérification de solvabilité** reçoit les informations financières extraites (revenu mensuel, dépenses mensuelles, montant du prêt) et calcule un ratio d'endettement. Ce ratio d'endettement permet d'évaluer la capacité de remboursement du client en fonction de ses revenus et de ses charges. Si le ratio d'endettement est jugé trop élevé, la demande de prêt sera considérée comme présentant un risque élevé, influençant ainsi la décision finale.

- **Fonction principale** : Calculer le ratio d'endettement pour évaluer la capacité de remboursement du client.
- **Entrée** : Revenu mensuel, dépenses mensuelles, montant du prêt demandé.
- **Sortie** : Ratio d'endettement, statut de solvabilité (solvable ou non), reste à vivre du client.

4.3 Service d'Évaluation de la Propriété

Le **service d'évaluation de la propriété** analyse les données relatives à la propriété mentionnée dans la demande. Il évalue la valeur de la propriété en fonction de critères standards (comme la surface en m^2, la localisation, le type de bien, etc.) et compare cette évaluation avec le montant du prêt demandé. Ce service détermine également si la propriété respecte les normes et règlements immobiliers.

- **Fonction principale** : Estimer la valeur de la propriété et évaluer sa conformité aux critères de prêt.
- **Entrée** : Description de la propriété (surface, emplacement, type de bien).
- **Sortie** : Valeur estimée de la propriété, niveau de risque associé (faible, moyen, élevé).

4.4 Service de Décision d'Approbation

Le **service de décision d'approbation** reçoit les résultats des services de solvabilité et d'évaluation de la propriété, ainsi que le montant du prêt demandé. Il compile ces informations et applique les critères d'approbation de l'institution financière. Ce service renvoie ensuite une décision d'approbation ou de refus, et peut également proposer des conditions particulières si nécessaire (par exemple, ajout de garanties ou ajustement du montant).

- **Fonction principale** : Prendre une décision finale d'approbation ou de refus du prêt.
- **Entrée** : Résultats de solvabilité, résultat de l'évaluation de la propriété, montant du prêt.

- **Sortie** : Décision finale (approuvé ou refusé), niveau de risque, raisons de la décision, éventuelles conditions supplémentaires.

Avantages de FastAPI pour le Développement des Services

Le choix de FastAPI pour le développement des services a permis d'obtenir plusieurs avantages :

- **Rapidité** : FastAPI est conçu pour offrir des performances élevées, ce qui est particulièrement utile pour les systèmes nécessitant une faible latence.
- **Validation des Données** : Grâce à la bibliothèque Pydantic intégrée à FastAPI, les entrées et sorties de chaque endpoint sont validées automatiquement, réduisant ainsi le risque d'erreurs liées aux formats de données.
- **Documentation Automatique** : FastAPI génère automatiquement une documentation interactive pour chaque service via Swagger, facilitant les tests et l'intégration des APIs.

5. L'ORCHESTRATEUR :

L'orchestrateur joue un rôle central dans le système d'évaluation des demandes de prêt. Il agit comme chef d'orchestre, assurant la coordination entre les différents services (extraction, solvabilité, évaluation, décision) afin que chaque étape du traitement de la demande se déroule dans le bon ordre. De plus, l'orchestrateur gère les erreurs et garantit que chaque service est bien appelé avec les données nécessaires pour effectuer son traitement.
En surveillant en permanence le répertoire où les nouvelles demandes sont déposées, l'orchestrateur détecte les nouveaux fichiers de demande de prêt et les traite immédiatement. Ce processus de surveillance et de déclenchement automatique permet une réactivité et une autonomie dans le traitement des demandes.
Fonctionnalités de l'Orchestrateur

1. **Surveillance des Demandes de Prêt**
   L'orchestrateur utilise une fonctionnalité de surveillance de fichiers pour détecter en temps réel les nouvelles demandes ajoutées dans un dossier dédié. Dès qu'un fichier texte contenant une demande de prêt est détecté, il est automatiquement pris en charge par le système pour passer à l'étape de traitement.
2. **Coordination des Services**
   Une fois la demande détectée, l'orchestrateur suit un enchaînement d'appels successifs aux services :
   - **Extraction des informations** : Envoie la demande brute au service d'extraction pour en récupérer les informations clés (revenus, dépenses, montant du prêt, etc.).
   - **Vérification de la solvabilité** : Passe les informations financières extraites au service de solvabilité pour déterminer la capacité de remboursement du client.
   - **Évaluation de la propriété** : Transmet la description de la propriété au service d'évaluation pour estimer sa valeur marchande.

- **Décision d’approbation** : Compile les résultats de solvabilité et d’évaluation pour prendre une décision d'approbation ou de refus de la demande.

3. **Gestion des Erreurs**
   L'orchestrateur inclut une gestion d’erreurs pour gérer les exceptions et les anomalies possibles lors de l’appel à chaque service. Si un service échoue, l’orchestrateur peut générer un message d'erreur et enregistrer la demande en tant que cas nécessitant une intervention manuelle. Cela garantit que le processus continue à fonctionner même en cas de défaillances ponctuelles.
4. **Stockage des Résultats**
   Après avoir obtenu la décision finale d’approbation ou de refus, l'orchestrateur compile toutes les informations de la demande et les résultats de chaque service. Les données sont ensuite sauvegardées dans une base de données SQLite, permettant ainsi de conserver un historique complet des demandes traitées, incluant les informations extraites, le calcul de solvabilité, l’évaluation de la propriété, et la décision finale.
5. **Génération de Fichiers de Résultats**
   En plus de stocker les données dans la base de données, l’orchestrateur crée un fichier JSON contenant le détail de la demande traitée, les résultats intermédiaires, et la décision finale. Ce fichier est stocké dans le même dossier que la demande originale, permettant une consultation facile des résultats par un utilisateur ou une interface externe.

Interaction avec les Autres Composants :

L’orchestrateur interagit avec chacun des services via des requêtes HTTP vers leurs endpoints respectifs, en suivant le processus d’appel séquentiel décrit ci-dessus. Chaque appel envoie les données nécessaires au service en question et récupère les résultats pour les transmettre au service suivant, jusqu'à la prise de décision finale.
Avantages de l'Orchestrateur

L'orchestrateur apporte plusieurs avantages au système :

- **Automatisation du traitement** : Toutes les étapes du processus sont déclenchées automatiquement dès qu'une demande est déposée.
- **Fiabilité et Résilience** : L’orchestrateur gère les erreurs pour garantir que le système continue à fonctionner même en cas de problème avec un service individuel.
- **Traçabilité** : Le stockage des résultats et la génération de fichiers de résultats permettent de conserver une trace détaillée de chaque demande, facilitant ainsi les analyses ultérieures et l’audit des décisions prises.

6. GESTION DES DONNÉES :

Stockage des Demandes de Prêt :

Chaque demande de prêt traitée par le système génère une quantité importante de données, incluant les informations extraites de la demande, les résultats de solvabilité, l'évaluation de la propriété, et la décision finale d'approbation ou de refus. Afin d'assurer la traçabilité et de pouvoir consulter ou analyser ces informations ultérieurement, toutes les données sont stockées dans une base de données relationnelle.
La base de données est utilisée pour :

- **Enregistrer chaque demande de prêt** avec toutes les étapes de son traitement.
- **Garder un historique des décisions** prises, permettant ainsi de consulter les demandes passées et leurs résultats.
- **Assurer la traçabilité** pour permettre d'auditer les décisions ou de répondre à des questions sur les critères de refus.

Choix de la Base de Données : SQLite :

Pour ce projet, **SQLite** a été choisie comme solution de stockage. SQLite est un système de gestion de base de données léger, facile à intégrer dans une application Python, et parfaitement adapté pour les besoins d'un projet de TP. Ce choix permet de stocker localement les données de manière simple et rapide, tout en offrant des fonctionnalités relationnelles essentielles pour structurer les données.
Avantages d'utiliser SQLite pour ce TP :

- **Simplicité d'Intégration** : SQLite fonctionne directement avec Python sans nécessiter de configuration serveur, ce qui en fait un choix idéal pour les environnements de développement.
- **Autonomie** : La base de données étant stockée sous forme de fichier, elle est portable et facilement manipulable.
- **Performances** : Pour le volume de données généré par le projet, SQLite offre des performances suffisantes.

Structure de la Base de Données

La base de données est structurée autour d'une table principale, `loan_requests`, qui centralise toutes les informations relatives aux demandes de prêt. Chaque enregistrement dans cette table représente une demande unique et contient les résultats des différentes étapes de traitement.
Table : loan_requests

- **id** : Identifiant unique pour chaque demande de prêt (clé primaire).
- **created_at** : Date et heure de la soumission de la demande, pour un suivi chronologique des traitements.
- **request_data** : Contenu brut de la demande de prêt tel qu'extrait du fichier texte initial.

- **extraction_result** : Résultat de l'extraction d'informations sous forme de données structurées (nom, montant du prêt, etc.).
- **solvability_result** : Résultats de l'évaluation de la solvabilité, incluant le ratio d'endettement et le statut de solvabilité.
- **evaluation_result** : Résultats de l'évaluation de la propriété, comme la valeur estimée et le niveau de risque.
- **decision** : Décision finale d'approbation ou de refus, ainsi que les éventuelles conditions ou recommandations.

Cette structure relationnelle permet de regrouper toutes les informations pertinentes pour chaque demande en un seul enregistrement, facilitant ainsi l'accès et la manipulation des données.

Exemple de Flux de Données

1. Lorsqu'une demande est soumise, elle est d'abord sauvegardée sous forme brute dans la base de données.
2. Une fois traitée par le service d'extraction, les informations structurées (comme les revenus, les dépenses, et le montant du prêt) sont ajoutées dans la colonne `extraction_result`.
3. Après l'évaluation de la solvabilité et de la propriété, les résultats sont également ajoutés dans les colonnes respectives, enrichissant l'enregistrement initial.
4. La décision finale est enregistrée dans la colonne `decision` une fois que toutes les étapes sont complétées, permettant ainsi une vue d'ensemble complète de la demande.

TESTS ET RÉSULTATS :

Pour garantir le bon fonctionnement du système et valider les interactions entre les différents services, plusieurs scénarios de test ont été définis. Ces tests permettent de vérifier la fiabilité, la précision et la rapidité du système dans divers cas de demande de prêt. Voici les principaux scénarios de test et leurs objectifs :

1. Test de Solvabilité avec Ratio d'Endettement Élevé
    - **Objectif** : Vérifier que le service de solvabilité identifie correctement les demandes présentant un risque financier élevé.
    - **Scénario** : Un client soumet une demande de prêt avec un montant élevé par rapport à ses revenus et dépenses.
    - **Résultat attendu** : Le service de solvabilité doit calculer un ratio d'endettement supérieur au seuil acceptable et indiquer que le client n'est pas solvable, influençant ainsi la décision de refus dans le service de décision.

2. Test de Solvabilité avec Ratio d'Endettement Bas
    - **Objectif** : Valider que le système approuve la demande lorsque le client présente un ratio d'endettement acceptable.
    - **Scénario** : Le client présente un revenu mensuel élevé et des dépenses modérées.
    - **Résultat attendu** : Le système identifie un faible risque financier, considérant le client comme solvable, et la demande est approuvée.
3. Test d'Évaluation de Propriété avec Valeur Anormale
    - **Objectif** : Tester la capacité du service d'évaluation à détecter les propriétés avec des valeurs suspectes (trop élevées ou trop basses).
    - **Scénario** : Le montant de la propriété est largement en dessous du prix de marché pour une région donnée.
    - **Résultat attendu** : Le service d'évaluation détecte la valeur comme suspecte, ce qui entraîne un refus de la demande en raison du risque élevé associé à la propriété.
4. Test d'Intégration Complète
    - **Objectif** : Valider l'enchaînement des appels entre services pour une demande complète de prêt, assurant que toutes les étapes fonctionnent ensemble sans erreur.
    - **Scénario** : Une demande de prêt avec toutes les informations nécessaires est soumise.
    - **Résultat attendu** : L'orchestrateur appelle successivement chaque service, qui renvoie des résultats cohérents. La décision finale est conforme aux données et critères fournis.

Résultats des Tests :

Les tests ont confirmé que le système atteint les objectifs de précision, de rapidité et de résilience dans tous les scénarios suivants :

1. **Précision des Résultats** : Chaque service a renvoyé les résultats attendus selon les différents scénarios testés. Le service de solvabilité a correctement évalué les capacités de remboursement en fonction des revenus et des dépenses, tandis que le service d'évaluation a identifié les biens dont les valeurs étaient anormales.
2. **Efficacité de l'Orchestrateur** : L'orchestrateur a géré les appels aux services dans le bon ordre et a correctement consolidé les résultats à chaque étape. Les tests ont démontré que même lors de l'exécution de plusieurs demandes en série, l'orchestrateur maintient un flux de travail fluide sans ralentissement.
3. **Gestion des Erreurs** : Dans les cas de données manquantes ou de valeurs incorrectes, le système a renvoyé des messages d'erreur pertinents et enregistré des notifications d'erreurs pour un suivi manuel. Cela garantit une résilience du système, assurant que les traitements se poursuivent pour les autres demandes sans interruption.
4. **Temps de Réponse** : L'utilisation de FastAPI a permis d'obtenir des temps de réponse optimisés pour chaque service, réduisant considérablement le délai de traitement d'une demande de prêt par rapport à l'implémentation précédente. Le traitement complet d'une demande, incluant les quatre services, a été mesuré comme étant suffisamment rapide pour une utilisation en environnement réel.

Conclusion des Tests

Les tests démontrent que le système remplit ses fonctions de manière cohérente et précise, offrant une automatisation fiable du processus d'évaluation des demandes de prêt immobilier. La transition vers FastAPI a permis d'améliorer les performances globales et la rapidité des échanges, rendant le système plus performant et capable de traiter les demandes en temps quasi-réel.
L'efficacité du système dans les différents scénarios testés confirme sa capacité à prendre des décisions de manière autonome et à répondre à des critères de solvabilité et d'évaluation rigoureux. Grâce à cette solution, les demandes de prêt peuvent désormais être traitées de manière rapide et automatisée, avec des résultats fiables à chaque étape du processus.

Conclusion :

Le projet de création d'un Service Web Composite d'Évaluation de Demande de Prêt Immobilier a atteint ses objectifs en développant un système robuste, automatisé et efficace pour le traitement des demandes de prêt. En s'appuyant sur une architecture microservices orchestrée par FastAPI, chaque étape du processus a été rationalisée, allant de l'extraction des informations à la décision finale d'approbation.
Le choix de SQLite comme solution de stockage a permis de garantir une gestion efficace et autonome des données, facilitant ainsi la traçabilité et l'analyse des décisions. La structure relationnelle de la base de données a permis de centraliser toutes les informations pertinentes pour chaque demande, rendant l'accès et la manipulation des données simples et rapides.
Les tests effectués ont confirmé la fiabilité et la précision du système. Les scénarios de test, allant de la vérification de la solvabilité à l'évaluation de la propriété, ont démontré que le système est capable de prendre des décisions éclairées, même dans des cas complexes. La gestion des erreurs a également été efficace, assurant une continuité de service malgré des données manquantes ou incorrectes.
Les résultats des tests ont prouvé que le système est non seulement rapide, mais également résilient, capable de gérer plusieurs demandes simultanément sans perte de performance. L'utilisation de FastAPI a été déterminante pour atteindre ces niveaux de performance, réduisant considérablement les délais de traitement des demandes de prêt.
En conclusion, cette solution répond aux exigences croissantes du secteur bancaire en matière d'automatisation et d'efficacité. En intégrant des critères de solvabilité rigoureux et des évaluations précises des propriétés, notre système constitue un atout précieux pour les institutions financières, leur permettant d'optimiser leurs processus d'évaluation de prêts. Les perspectives d'évolution incluent l'ajout de fonctionnalités avancées et l'adaptation du système à de nouveaux contextes réglementaires, garantissant ainsi sa pérennité et sa pertinence dans un paysage financier en constante mutation.

Annexes: Captures d'écran des tests