

M2 DataScale

TP - Architectures Orientées Service

Spécification du Service Web Composite : Evaluation de
Demande de Prêt Immobilier

Thivagini SUGUMAR, Marine CONOR

3 décembre 2023

Table Des Matières

1	Contexte général du TP	3
2	Objectif	3
3	Fonctionnement du système	3
3.1	Architecture	3
3.2	Étapes	4
4	Identification des services	5
5	Modélisation des services	6
5.1	Modélisation	6
5.2	Fonctionnement interne des services	7
6	Implémentation et déploiement avec Flask avec SOAP	10
6.1	Implémentation	10
6.2	Interface	15
6.3	Cas à tester	15
7	Implémentation et déploiement avec Flask avec REST	15
7.1	Implémentation	15
7.2	Interface	15
7.3	Cas à tester	15
8	Liens des dépôts du projet	16

1 Contexte général du TP

Le service Web Composite d'Évaluation de Demande de Prêt Immobilier est conçu pour automatiser le processus d'évaluation des demandes de prêt immobilier en utilisant des services Web spécialisés. Il permet aux clients de soumettre des demandes de prêt immobilier exprimées en langage naturel. Le service intègre des composants d'extraction des informations métiers de texte de la demande, de vérification de solvabilité, d'évaluation de la propriété et de décision d'approbation pour fournir une évaluation complète et précise des demandes de prêt.

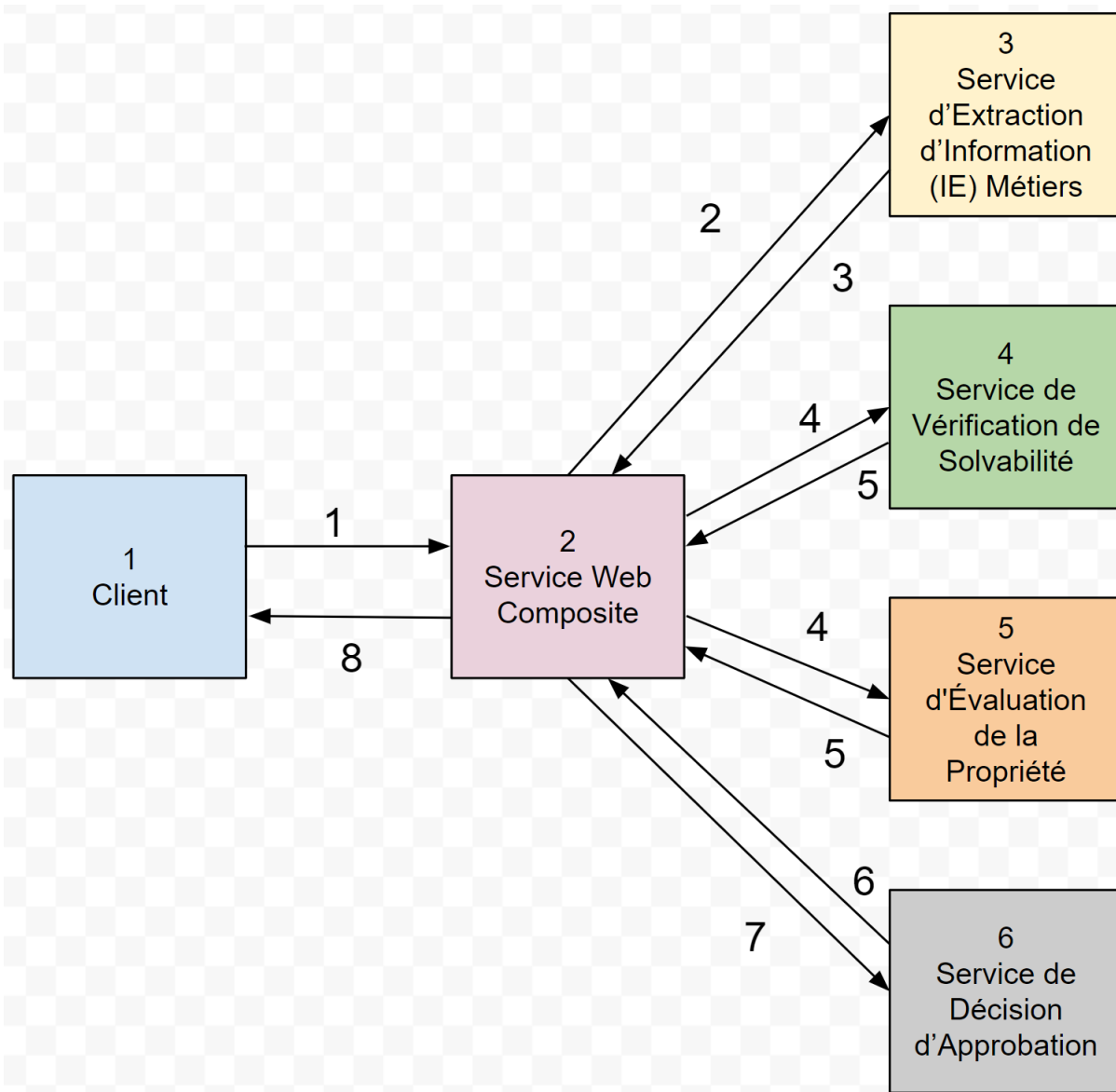
2 Objectif

L'objectif de ce service est de fournir une interface unique pour les clients qui souhaitent demander un prêt immobilier. Le service coordonne les différents services Web nécessaires pour évaluer la demande du client, garantissant ainsi un processus fluide et automatisé.

3 Fonctionnement du système

3.1 Architecture

Nous représentons l'architecture du fonctionnement du système par le diagramme suivant :



3.2 Étapes

- Sur l'interface, le client va effectuer une demande de prêt. Pour cela, le client doit remplir un formulaire et le soumettre. Cette action va permettre de créer un fichier sous format `txt` dans le répertoire `demandeTxt`
- Le service **Composite** va détecter l'arrivée du fichier `txt` dans le répertoire `demandeTxt` grâce à une routine qui se lance automatiquement. Ainsi le service composite va récupérer ce fichier `txt`
- Le service composite va effectuer une requête au service **Extraction** afin de récupérer le fichier WSDL
- Le service composite va alors transmettre le fichier `txt` au service d'**Extraction** via sa méthode web service encapsulé dans une enveloppe SOAP
- Le service d'**Extraction** va effectuer le prétraitement, identifier des entités et extraire les entités avec les expressions régulières. Cette extraction sera sauvegardée dans un fichier XML dans `demandeXML` sous le format : `numeroDossier.xml`
- Le service d'**Extraction** renvoie ce fichier XML obtenu au service **Composite**
- Le service **Composite** donne le fichier XML au service de **Vérification de Solvabilité**
- Le service de **Vérification de Solvabilité** va utiliser une fonction de scoring pour donner un score. Il ajoute les balises `<scoring></scoring>` et `<decisionScoring></decisionScoring>` dans le fichier XML
- Le service de **Vérification de Solvabilité** renvoie le fichier XML modifié au service **Composite**
- Le service **Composite** donne le fichier XML modifié au service d'**Évaluation de la Propriété**
- Le service d'**Évaluation de la Propriété** va utiliser une fonction pour évaluer le marché immobilier afin de donner une moyenne et évaluer les normes légales et réglementaires du bâtiment. Il peut aussi décider de faire une visite virtuelle et une visite sur place. Il ajoute les balises :
`<estimation_valeur></estimation_valeur>` et
`<decisionConformite></decisionConformite>`, si la décision est "Non admissible".
Il ajoute également aussi une balise `<raisons></raisons>` dans le fichier XML donné en entrée
- Le service d'**Évaluation de la Propriété** renvoie le fichier XML modifié au service **Composite**
- Le service **Composite** donne le fichier XML modifié au service de **Décision d'Approbation**
- Le service de **Décision d'Approbation** va utiliser une fonction comparant les résultats et analyses précédents. Il va ensuite créer un nouveau fichier `txt` dans le dossier `reponseTxt` sous le format `numeroDossier.txt`
- Le service de **Décision d'Approbation** renvoie le fichier `txt` au service **Composite**
- Le client peut retrouver sa réponse sur l'interface en consultant les résultats

4 Identification des services

Les services identifiés sont les suivants :

- Le service **Composite** :

Permet l'interaction entre le Client et les différents Services. Il s'occupe notamment de la réception de la demande, l'extraction d'information métier, la vérification de solvabilité, l'évaluation de la propriété et la décision d'approbation.

- Le service d'**Extraction** :

Reçoit le texte de la demande de prêt soumise par le client, via le service **Composite**. Il s'occupe principalement du prétraitement, analyse, identification des entités et l'extraction de ces informations. De plus, il permet le stockage des informations extraites dans la base de données du service **Composite**

- Le service de **Vérification de Solvabilité** :

Permet d'évaluer la capacité financière du client à rembourser le prêt. Il s'occupe donc de l'intégration des bureaux de crédit pour se connecter à leurs bureaux qui stockent l'historique financier des clients. Il récupère des informations telles que les dettes en cours, les paiements en retard et les antécédents de faillite pour évaluer la crédibilité du client. Ensuite, le service utilise des algorithmes de scoring de crédit pour attribuer un score au client en fonction de son historique financier. Ce score représente le niveau de risque associé à l'accord d'un prêt au client. Les clients avec des scores élevés ont une solvabilité plus élevée. Il analyse aussi les revenus et les dépenses mensuelles.

- Le service d'**Évaluation de la Propriété** :

Permet d'estimer la valeur marchande de la propriété pour laquelle le prêt est demandé. Il peut utiliser des données immobilières, des expertises locales et des critères légaux pour effectuer cette évaluation. Il s'occupe donc d'analyser les données du marché immobilier, peut effectuer une inspection virtuelle de la propriété ou même des visites sur place et vérifier si la propriété est conforme aux normes légales et réglementaires en vigueur.

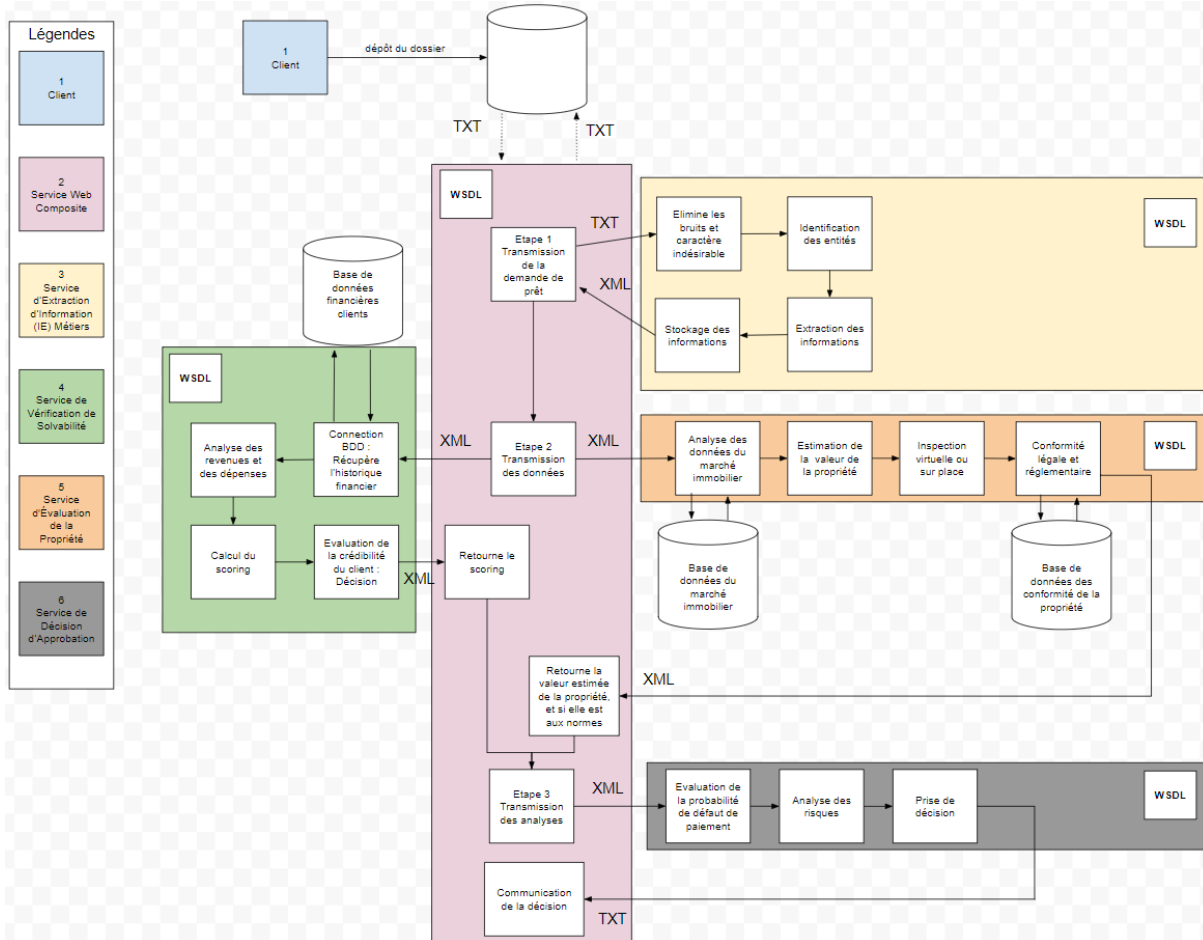
- Le service de **Décision d'Approbation** :

Analyse les données recueillies lors des étapes précédentes (Vérification de Solvabilité et Évaluation de la Propriété) pour déterminer si le prêt immobilier peut être approuvé. Il s'occupe d'analyser les risques associés à l'accord du prêt, de comparer les données de la demande de prêt avec les politiques internes de l'institution financière pour évaluer la probabilité de défaut de paiement du client. En fonction des résultats, le service prend une décision d'approbation ou de refus du prêt puis communique celle-ci au service **Composite** (en indiquant les raisons de l'approbation ou du refus du prêt) qui va retourner cette réponse au client.

5 Modélisation des services

5.1 Modélisation

Une architecture à base de service pour mettre en œuvre le processus d'évaluation de demande de prêt immobilier :



5.2 Fonctionnement interne des services

Le service **Composite** permet les échanges entre les services et les échanges avec le client et organise le SOAP.

Le service d'**Extraction** prend en paramètre le fichier **txt** de la demande du client.

Ce service va récupérer les informations indiquées dans le fichier **txt**, supprime les caractères indésirables et extrait les données après le caractère " : ".

Une fois extraites, les données sont écrites dans des balises correspondantes dans un fichier **XML**.

Le fichier XML sera nommé selon le numéro de dossier attribué au client.

Le service de **Vérification de Solvabilité** prend en paramètre le fichier XML concernant la demande du client.

Ce service va récupérer les informations concernant le montant et la durée du prêt ainsi que les revenus et les dépenses mensuelles à l'aide du numéro de dossier du client.

Il va également récupérer le fichier **json** banque :

(**idBanque**, **age**, **enfants**, **emploi**, **nbCreditsEnCours**, **antecedents**, **tauxEndettement**) où :

- **idBanque** est l'identifiant de la banque du client
- **age** est l'âge du client
- **enfants** est le nombre d'enfants à sa charge
- **emploi** est un booléen avec 0 si l'emploi n'est pas stable, 1 sinon
- **nbCreditsEnCours** est le nombre de crédits totaux toujours en cours
- **antecedents** est le nombre d'antécédents à son actif (dettes non payées, retards...)
- **tauxEndettement** est le taux d'endettement actuel

Le service va ensuite extraire les données en prenant le tuple correspondant à l'**idBanque** du fichier XML et faire un scoring en fonction des informations.

La décision sera immédiatement "Non Admissible" si le client a moins de 18 ans ou si sa capacité d'emprunt ajouté à ses dépenses est supérieure à son revenu ou encore si son taux d'endettement est supérieur à 33%.

Une fois l'algorithme de scoring effectué, le service va déterminer en fonction du score, une décision :

- entre 0 et 10 : peu probable
- entre 10 et 20 : à défendre
- entre 20 et 30 : sous conditions
- entre 30 et 40 : très favorable

Finalement, le service ajoute au fichier XML reçu, les balises **<scoring>Valeur du score</scoring>** et **<decisionScoring>Decision en fonction du score</decisionScoring>**.

Le service d'**Évaluation de la Propriété** prend en paramètre le fichier XML selon la demande du client.

Le service vérifie si la propriété est conforme aux normes légales et réglementaires en vigueur. Il va pour cela, récupérer l'adresse de la propriété puis récupérer le fichier `json` immobilier : (`idImmobilier`, `adresse`, `age`, `normeLegal`, `normeReglementaire`, `litigesEnCours`, `normeElectricite`, `normeGaz`) où :

- `idImmobilier` est l'identifiant de la propriété
- `adresse` est l'adresse de la propriété
- `age` est l'âge de la propriété
- `normeLegal` est un booléen avec 0 si la propriété est aux normes, 1 sinon
- `normeReglementaire` est un booléen avec 0 si la propriété est aux normes, 1 sinon
- `litigesEnCours` est le nombre de litiges en cours
- `normeElectricite` est la norme électrique ("NFC 15-100", "NF C 15-100" ou "C 15-100")
- `normeGaz` est la norme pour le gaz ('NF P 45-500' ou vide s'il n'y a pas de gaz dans la propriété)

Le service va ensuite extraire les données en prenant le tuple correspondant à l'adresse du fichier XML et vérifier que tout est conforme : soit 0, soit dans les normes définies.

Si l'un des attributs n'est pas aux normes, la décision est "Non Admissible à un prêt" et liste les raisons pour lesquelles la propriété pose problème.

Il ajoute au fichier XML reçu la balise :

```
<decisionConformite>
Decision en fonction du respect ou non des normes
</decisionConformite>
```

Si la décision est "Non admissible", il ajoute aussi une balise :

```
<raisons>Liste des raisons du refus</raisons>.
```

Le service peut demander à faire une visite virtuelle ou sur place, on considère que si le service demande une visite, il effectue d'abord une visite virtuelle, si celle-ci n'est pas concluante, il demande alors un expert pour une visite sur place.

Le service va choisir un entier aléatoire entre 0 et 1 pour savoir s'il fait une visite.

Si c'est 0, il ne fait aucune demande de visite, sinon il enclenche une visite virtuelle.

Par la suite, on suppose qu'il va faire une visite sur place uniquement si l'âge de la propriété est inférieur à 10 ans (il n'est plus nécessaire d'avoir un certificat de conformité si la propriété a plus de 10 ans).

En fonction des résultats précédents vis-à-vis de la conformité, la visite sur place est déterminée concluante ou non.

Le service va aussi estimer la valeur marchande de la propriété pour laquelle le prêt est demandé.

Il va récupérer le code postal de l'adresse puis récupérer le type de bâtiment (maison/appartement) et, s'il est donné, le nombre d'étages, avec la `descriptionPropriete` du fichier XML.

Ensuite il récupère le fichier `json` `marcheImmobilier` :

(`adresse`, `codePostal`, `batiment`, `nbEtage`, `valeur`) où :

- `adresse` est l'adresse de la propriété
- `codePostal` est le code postal
- `batiment` est le type de bâtiment
- `nbEtage` est le nombre d'étages
- `valeur` est la valeur marchande de la propriété

Le service va récupérer tous les tuples qui correspondent au même code postal que celui du fichier XML ainsi que le même type de bâtiment, le nombre d'étages s'il est donné et faire une moyenne de leur valeur. Finalement, il ajoute au fichier XML reçu, la balise :

`<estimation_valeur>`

Valeur moyenne des bâtiments du même type et dans le même secteur

`</estimation_valeur>`

Le service de **Décision d'Approbation** prend en paramètre le fichier XML concernant la demande du client.

Ce service va récupérer les informations concernant le montant du prêt, le score, la décision du score, la décision de conformité (les raisons pour lesquelles c'est un refus si c'en est un) ainsi que l'estimation de la valeur.

- Si la valeur du score est -1, le score n'a pas été calculé à cause de sa situation financière, le prêt est refusé.
- Si la décision de conformité est "Non admissible à un prêt immobilier", le service émet le refus du prêt avec une liste de raisons.
- Si le montant est supérieur à l'estimation de la valeur en plus d'une valeur choisie arbitrairement, le service émet un refus avec les raisons que le montant demandé est trop important pour la propriété.

Par la suite en fonction de la décision du score, s'il s'agit de "Très favorable", "Sous conditions" ou "À défendre" et que la décision de conformité est aussi "Admissible à un prêt immobilier" alors le service accepte la demande, sinon il la refuse.

En fonction des résultats précédents, le service a pris une décision d'approbation ou de refus du prêt puis communique celle-ci au service **Composite** (en indiquant les raisons du refus du prêt) sous la forme d'un fichier `txt` qui sera stocké dans le répertoire `reponseTxt`.

6 Implémentation et déploiement avec Flask avec SOAP

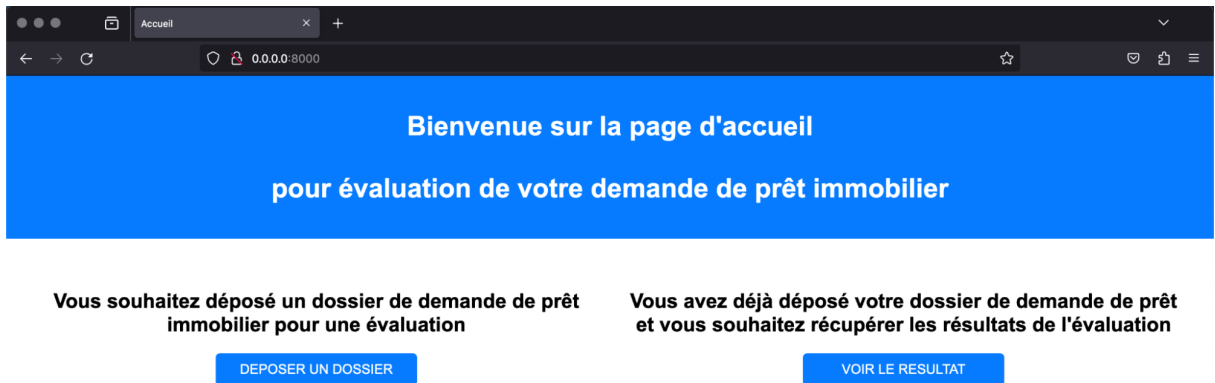
6.1 Implémentation

- Lancement du serveur

```
[(base) thivani@MacBook-de-Thiva ~ % cd Desktop/Projet_SOA  
(base) thivani@MacBook-de-Thiva Projet_SOA % ./run
```

```
Démarrage du serveur...
```

- Interface Web accueil :



- Si l'utilisateur clique sur "Déposer un dossier", il est redirigé vers la page de formulaire :

The screenshot shows a web browser window with the address bar displaying '0.0.0.0:8000/formulaire.html'. The page has a blue background and a white form titled 'Formulaire de demande de prêt'. The form contains the following fields:

- Nom complet : John Doe
- Adresse : 123 Rue de la Liberté, 75001 Paris, France
- E-mail : john.doe@email.com
- Numéro de téléphone : 0123456789
- Montant du prêt (en euros) : 200000
- Durée du prêt (en années) : 20
- Description de la propriété : Maison a deux étages avec jardin, située dans un quartier résidentiel calme
- Revenu mensuel (en euros) : 5000
- Dépense mensuelle (en euros) : 3000
- Numéro de compte bancaire : 11
- Référence de la propriété : 1

At the bottom of the form is a blue button labeled 'SOUMETTRE'.

- Après avoir cliqué sur le bouton "Soumettre", la page confirmation s'ouvre :

The screenshot shows a web browser window with the address bar displaying '0.0.0.0:8000/enregistrer'. The page has a blue background and a white confirmation box titled 'Confirmation de la soumission du dossier de demande de prêt'. The box contains the following text:

Votre dossier a bien été soumis à notre service d'évaluation sous le numéro de dossier suivant :

439971

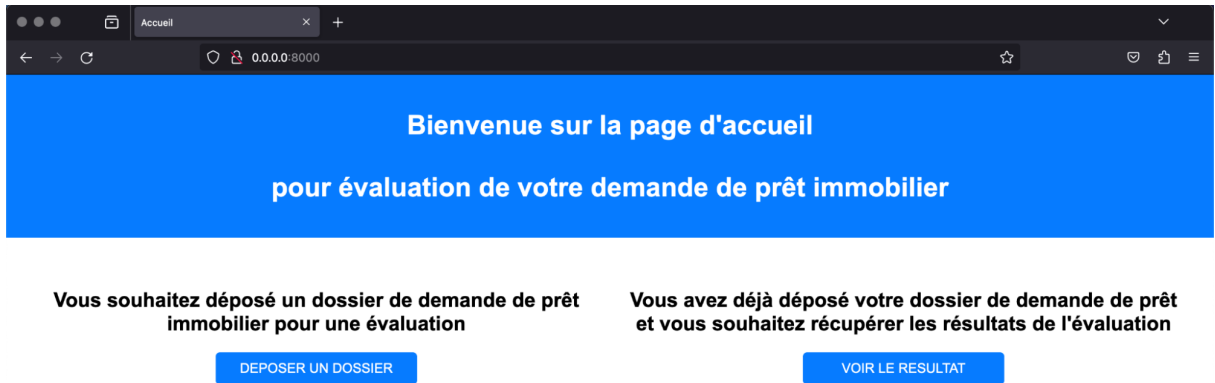
Gardez bien ce numéro pour vous connecter la prochaine fois pour pouvoir récupérer vos résultats.

At the bottom of the box is a blue button labeled 'RETOUR'.

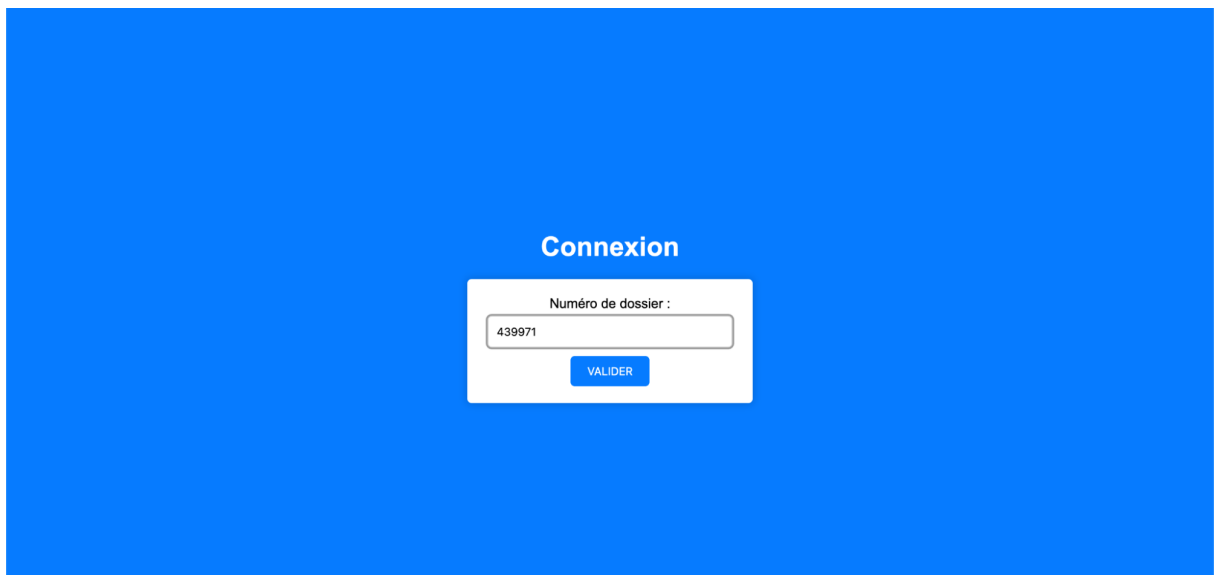
Elle nous indique le numéro de dossier de la demande qui est à conserver pour pouvoir consul-

ter les résultats par la suite

- Si l'utilisateur clique sur "Retour", il revient à la page d'accueil



- Si l'utilisateur clique sur le bouton "Voir le résultat", la page de connexion s'affiche



L'utilisateur saisit donc son numéro de dossier

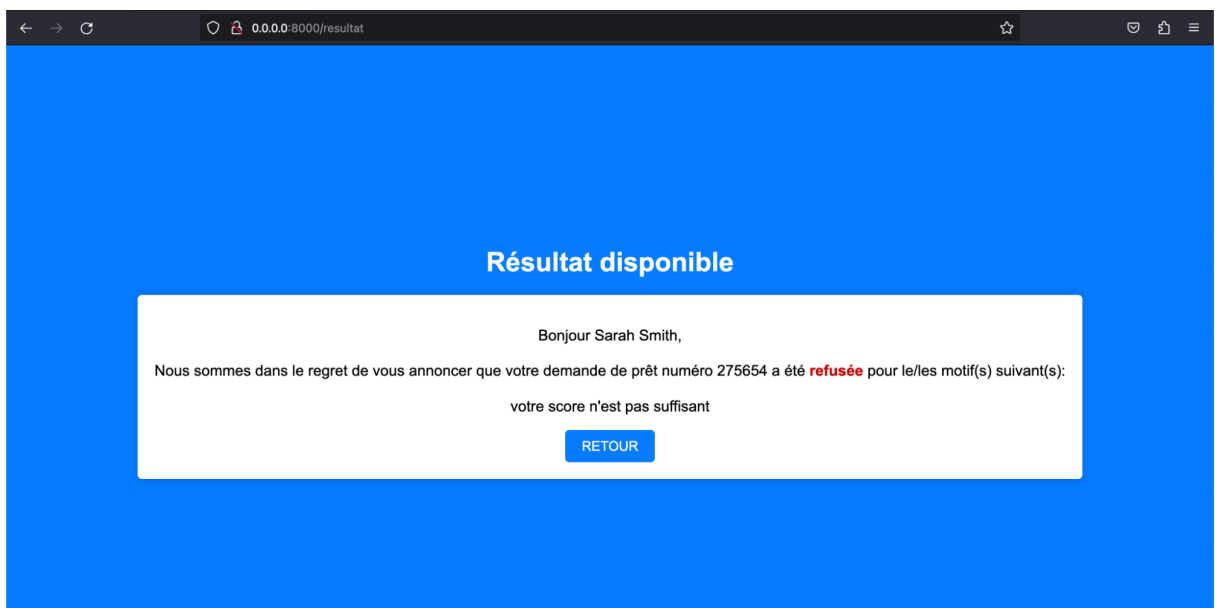
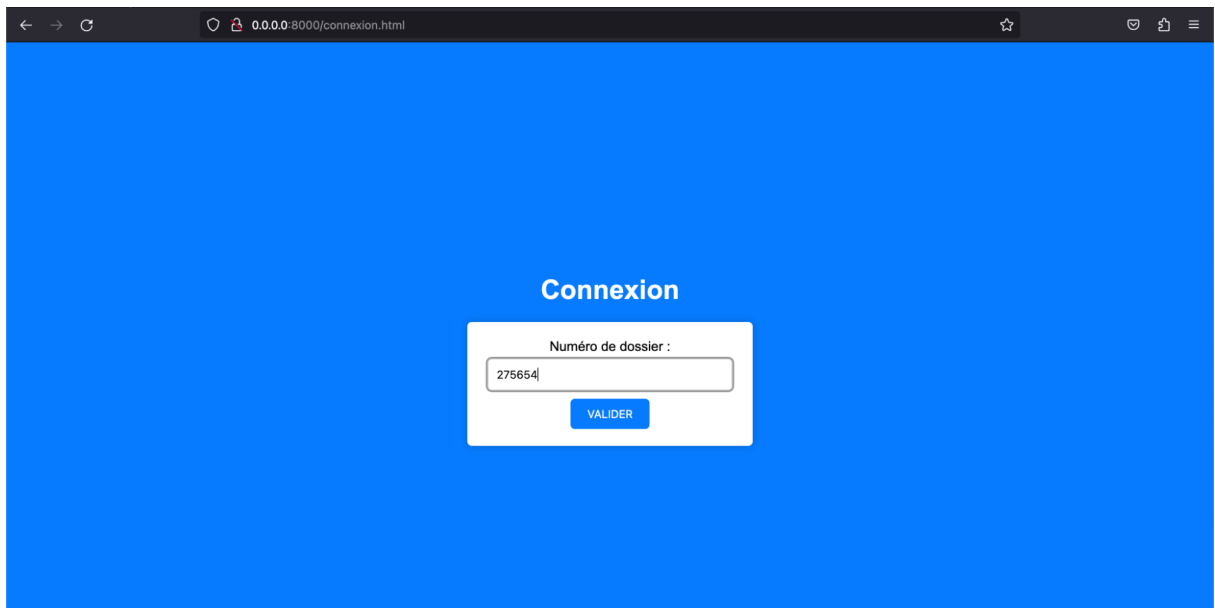
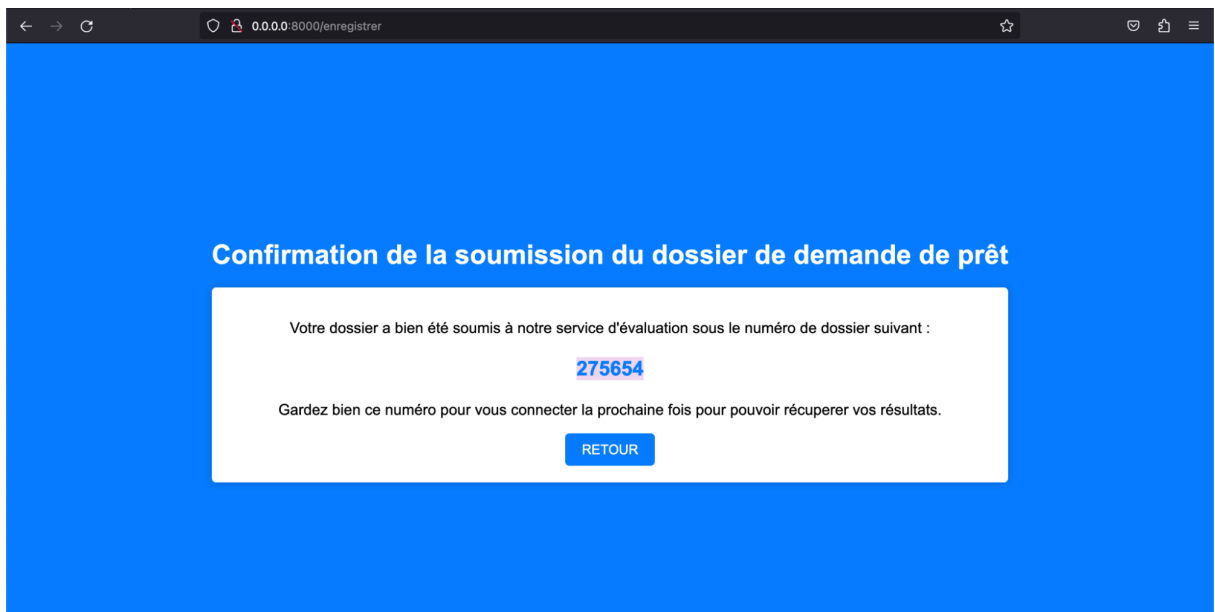
- L'utilisateur accède à la page de confirmation de la demande de prêt :



Sur celle-ci y figure les modalités à respecter

- Voici la même demande avec un dossier refusé :

A screenshot of a web browser displaying a loan application form titled "Formulaire de demande de prêt". The browser's address bar shows "0.0.0.0:8000/formulaire.html". The form is centered on a blue background and contains the following fields: "Nom complet :" with the value "Sarah Smith"; "Adresse :" with the value "45 avenue des Etats-Unis, 78000 Versailles, France"; "E-mail :" with the value "sarahsmith@email.com"; "Numéro de téléphone :" with the value "0123456788"; "Montant du prêt (en euros) :" with a spinner box set to "150000"; "Durée du prêt (en années) :" with a spinner box set to "30"; "Description de la propriété :" with the text "Appartement situé dans un quartier d'affaire" (where "situé" is underlined in red); "Revenu mensuel (en euros) :" with a spinner box set to "2000"; "Dépense mensuelle (en euros) :" with a spinner box set to "1500"; "Numéro de compte bancaire :" with a spinner box set to "22"; and "Référence de la propriété :" with a spinner box set to "2". At the bottom of the form is a blue button labeled "SOUMETTRE".



6.2 Interface

6.3 Cas à tester

7 Implémentation et déploiement avec Flask avec REST

7.1 Implémentation

7.2 Interface

7.3 Cas à tester

8 Liens des dépôts du projet

- [SOAP](https://github.com/uvsq21704755/ProjetSOAP.git) : `https://github.com/uvsq21704755/ProjetSOAP.git`
- [REST](https://github.com/uvsq21704755/Projet_REST.git) : `https://github.com/uvsq21704755/Projet_REST.git`