Cahier des Charges

1. Contexte et Objectifs

Contexte :
L'IFRI, l'Institut de Formation et de Recherche en Informatique de l'Université d'Abomey-Calavi, rencontre des difficultés dans la gestion des demandes de documents académiques (bulletins de notes, attestations, certificats, relevés, etc.). Le processus actuel est lent, peu transparent et dépend de manipulations manuelles. De plus, le traitement des paiements liés à ces services nécessite une vérification rigoureuse, souvent via des plateformes de paiement en ligne (ex. trésor public).

Objectifs :

- **Automatiser** le processus de demande de documents académiques.
- **Centraliser** la gestion des demandes via une plateforme en ligne.
- **Améliorer** la traçabilité et la transparence du suivi des demandes.
- **Intégrer** la vérification du paiement (via API ou extraction d'informations à partir d'un reçu/identifiant unique).
- **Intégrer** des fonctionnalités d'extraction automatique (OCR) pour limiter le volume de fichiers uploadés et en simplifier le monitoring.

2. Périmètre Fonctionnel

2.1. Acteurs

- **Étudiants :**
    - Accès à la plateforme pour initier des demandes de documents ou de services.
    - Soumission d'un formulaire comprenant :
        - Informations personnelles (identifiant unique de l'étudiant, infos de la carte étudiant, etc.).
        - Type de service demandé.
        - Preuve de paiement (upload de fichier ou données d'un paiement en ligne via un identifiant unique).
- **Administration / Personnel IFRI :**
    - Réception et traitement des demandes via :
        - Un tableau de bord intégré à leur système interne (via API).
        - Ou un module de gestion intégré à la plateforme proposée.
    - Vérification et validation du paiement ainsi que des pièces justificatives.
    - Possibilité de revoir manuellement les informations extraites par l'outil OCR et valider le rendu final.

2.2. Processus de Demande (Workflow)

1. **Initiation de la demande :**
    - L'étudiant se connecte et remplit un formulaire de demande.
    - Avant la demande, le paiement doit être effectué via le canal officiel (banque/trésor public).
    - L'étudiant fournit la preuve de paiement (sous forme de fichier scanné [PDF, PNG] ou en renseignant un identifiant unique de paiement).
2. **Vérification du paiement et des pièces justificatives :**
    - **Approche 1 :** Vérification manuelle via tableau de bord par l'administration.
    - **Approche 2 :** Intégration d'API pour interroger automatiquement les systèmes de paiement afin de valider le paiement.
    - **Approche complémentaire :** Utilisation d'un module d'OCR pour extraire automatiquement les informations des pièces uploadées.
        - Un tableau de bord intermédiaire affichera les informations extraites pour vérification et validation (prototype).
3. **Traitement et Suivi de la Demande :**
    - Le statut de la demande évolue (ex. : en attente, en cours de traitement, validée, livrée).
    - Envoi de notifications par e-mail à l'étudiant à chaque changement d'état.
    - Une fois validée, le document généré (bulletin, attestation, etc.) est mis à disposition ou envoyé directement à l'étudiant.

3. Spécifications Fonctionnelles

3.1. Gestion des Demandes

- **Formulaire en ligne :**
    - Saisie des informations nécessaires (type de service, identifiant étudiant, etc.).
    - Possibilité de renseigner l'identifiant unique de paiement ou d'uploader un fichier de preuve de paiement.
- **Validation et Suivi :**
    - Suivi en temps réel de l'état de la demande via un tableau de bord pour l'étudiant.
    - Notifications par e-mail à chaque mise à jour (soumission, validation, refus, livraison).
- **Traitement des Pièces Justificatives :**
    - Intégration d'un module d'OCR pour extraire automatiquement les informations des documents scannés.
    - Possibilité d'afficher un récapitulatif (sous forme de tableau) des données extraites pour vérification avant validation définitive (fonctionnalité prototype).

3.2. Vérification de Paiement

- **Possibilité d'implémenter le paiement sur la plateforme:**
- **API d'Agrégation de Paiement :**
    - Intégration d'API permettant de vérifier automatiquement le paiement (montant, correspondance avec le compte IFRI, etc.).
    - Possibilité de croiser l'information fournie par l'étudiant (identifiant ou reçu) avec les données reçues via l'API.
- **Alternatives en cas d'accès limité :**
    - Prévoir un module de vérification manuelle par l'administration si l'accès direct aux systèmes de paiement n'est pas possible.

4. Spécifications Techniques

4.1. Architecture et Technologies

- **Backend :**
    - Langages / Frameworks : Python avec Django ou Flask (pour le prototype), possibilité d'évolution vers NodeJS.
    - Mise en place d'API REST pour la communication avec le système interne d'IFRI et les agrégateurs de paiement.
- **Frontend :**
    - Technologies : React JS ou HTML/CSS classique associé à Tailwind CSS (selon les compétences de l'équipe).
    - Interface responsive pour une utilisation sur desktop et mobile.
- **Base de Données :**
    - Options envisagées : MongoDB, SQLite ou autre solution relationnelle/NoSQL en fonction des exigences d'intégration et de scalabilité.
- **Modules complémentaires :**
    - API pour extraction d'informations via OCR.
    - Intégration de modules de scan et d'analyse des documents pour générer un rendu JSON structuré des données extraites.

4.2. Intégration et Déploiement

- **Interfaçage avec les systèmes internes :**
    - Prévoir des endpoints sécurisés et des échanges via API pour permettre la communication avec le système de l'IFRI.
- **Sécurité et Conteneurisation :**
    - Mise en place des bonnes pratiques en termes de sécurité (gestion des données sensibles, conformité RGPD).
    - Authentification basée sur une identification unique pour les utilisateurs (possibilité d'évolution vers une authentification multi-facteurs ultérieurement).
    - Déploiement via conteneurisation (Docker) avec gestion des tests unitaires et d'intégration pour assurer la robustesse de la solution.

5. Exigences Non Fonctionnelles

- **Performance & Scalabilité :**
    - Bien que le prototype vise un déploiement de petite échelle, la solution doit être conçue pour supporter potentiellement des milliers d'étudiants (actuels et anciens).
    - Disponibilité élevée et temps de réponse optimisé, surtout pour les vérifications de paiement et l'extraction des informations.
- **Sécurité :**
    - Gestion stricte des accès et protection des données sensibles.
    - Respect des normes de sécurité applicables aux institutions d'État.
- **Ergonomie & Accessibilité :**
    - Interface simple et intuitive pour les étudiants et le personnel administratif.
    - Site responsive et compatible avec divers navigateurs.

6. Planning et Livrables (Prototype Initial)

- **Livrable Prototype :**
    - Mise en place d'une plateforme fonctionnelle permettant :
        - La soumission des demandes avec vérification du paiement.
        - Un tableau de suivi pour l'étudiant.
        - Une démonstration du module d'OCR avec extraction et affichage des informations extraites (fonctionnalité démonstrative).
- **Documentation :**
    - Requirement Document (spécifications détaillées).
    - Design Document (schémas d'architecture et de processus).
    - Technical Document (détails d'implémentation, choix technologiques, protocoles d'intégration).
- **Tests :**
    - Tests unitaires et tests d'intégration pour assurer le bon fonctionnement des modules critiques.

7. Points Ouverts et Évolutions Futures

- **Accès aux systèmes internes de l'IFRI :**
    - La faisabilité des intégrations dépendra de l'acceptation du projet par l'administration et de la possibilité d'accéder aux endpoints requis.
- **Évolution des méthodes d'authentification et sécurité :**
    - Possibilité d'ajouter des couches supplémentaires (authentification multi-facteurs, single sign-on, etc.) après validation du prototype.
- **Optimisation de l'extraction d'informations :**
    - Affiner les algorithmes OCR et le traitement des données extraites pour chaque type de document, en fonction des retours des utilisateurs

8. Prochaine étape: Valider ces points avec l’ensemble des parties prenantes.
Noter que ce ne sont que des spécifications simplifiées il aura bientôt le Requirements Doc, le Design Doc et le Technical Doc.